

# Package: openintro (via r-universe)

October 13, 2024

**Title** Datasets and Supplemental Functions from 'OpenIntro' Textbooks and Labs

**Version** 2.5.0

**Description** Supplemental functions and data for 'OpenIntro' resources, which includes open-source textbooks and resources for introductory statistics (<<https://www.openintro.org/>>). The package contains datasets used in our open-source textbooks along with custom plotting functions for reproducing book figures. Note that many functions and examples include color transparency; some plotting elements may not show up properly (or at all) when run in some versions of Windows operating system.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.3.1

**Suggests** broom, dplyr, forcats, knitr, lubridate, scales, testthat (>= 3.0.0), tidyr, tidytext, stringr, maps

**Imports** ggplot2 (>= 2.2.1), graphics, readr, rmarkdown, tibble

**Depends** R (>= 2.10), airports, cherryblossom, usdata

**URL** <http://openintrostat.github.io/openintro/>,  
<https://github.com/OpenIntroStat/openintro/>

**BugReports** <https://github.com/OpenIntroStat/openintro/issues>

**VignetteBuilder** knitr

**Roxygen** list(markdown = TRUE)

**Config/testthat/edition** 3

**Repository** <https://openintrostat.r-universe.dev>

**RemoteUrl** <https://github.com/openintrostat/openintro>

**RemoteRef** HEAD

**RemoteSha** 9b858c9f865951e1990131acbd401836568818e5

## Contents

absenteeism . . . . .	8
acs12 . . . . .	9
age_at_mar . . . . .	10
ames . . . . .	11
ami_occurrences . . . . .	13
antibiotics . . . . .	14
arbutnot . . . . .	15
ArrowLines . . . . .	16
ask . . . . .	17
association . . . . .	18
assortive_mating . . . . .	19
avandia . . . . .	20
AxisInDollars . . . . .	21
AxisInPercent . . . . .	22
babies . . . . .	23
babies_crawl . . . . .	23
bac . . . . .	24
ball_bearing . . . . .	25
bdims . . . . .	26
BG . . . . .	28
biontech_adolescents . . . . .	29
birds . . . . .	30
births . . . . .	31
births14 . . . . .	32
blizzard_salary . . . . .	33
books . . . . .	34
boxPlot . . . . .	35
Braces . . . . .	37
buildAxis . . . . .	38
burger . . . . .	41
calc_streak . . . . .	41
cancer_in_dogs . . . . .	42
cards . . . . .	42
cars04 . . . . .	43
cars93 . . . . .	44
cchousing . . . . .	45
CCP . . . . .	46
census . . . . .	47
cherry . . . . .	48
children_gender_stereo . . . . .	49
china . . . . .	50
ChiSquareTail . . . . .	51
cia_factbook . . . . .	52
classdata . . . . .	53
cle_sac . . . . .	54
climate70 . . . . .	54

climber_drugs . . . . .	55
coast_starlight . . . . .	56
COL . . . . .	57
comics . . . . .	58
contTable . . . . .	59
corr_match . . . . .	60
country_iso . . . . .	61
cpr . . . . .	62
cpu . . . . .	62
credits . . . . .	64
CT2DF . . . . .	64
daycare_fines . . . . .	65
densityPlot . . . . .	67
diabetes2 . . . . .	69
dlsegments . . . . .	70
dotPlot . . . . .	72
dotPlotStack . . . . .	74
dream . . . . .	75
drone_blades . . . . .	75
drug_use . . . . .	76
duke_forest . . . . .	77
earthquakes . . . . .	78
ebola_survey . . . . .	79
edaPlot . . . . .	79
elmhurst . . . . .	80
email . . . . .	81
email50 . . . . .	83
env_regulation . . . . .	85
epa2012 . . . . .	86
epa2021 . . . . .	87
esi . . . . .	89
ethanol . . . . .	91
evals . . . . .	92
exams . . . . .	93
exam_grades . . . . .	93
exclusive_relationship . . . . .	94
fact_opinion . . . . .	95
fadeColor . . . . .	96
family_college . . . . .	98
fastfood . . . . .	99
fcid . . . . .	100
fheights . . . . .	100
fish_age . . . . .	101
fish_oil_18 . . . . .	102
flow_rates . . . . .	103
friday . . . . .	104
full_body_scan . . . . .	105
gdp_countries . . . . .	106

gear_company . . . . .	107
gender_discrimination . . . . .	107
get_it_dunn_run . . . . .	108
gifted . . . . .	109
global_warming_pew . . . . .	110
goog . . . . .	111
gov_poll . . . . .	111
gpa . . . . .	112
gpa_iq . . . . .	113
gpa_study_hours . . . . .	113
gradestv . . . . .	114
gsearch . . . . .	115
gss2010 . . . . .	115
gss_wordsum_class . . . . .	116
healthcare_law_survey . . . . .	117
health_coverage . . . . .	117
heart_transplant . . . . .	118
helium . . . . .	119
helmet . . . . .	120
hfi . . . . .	121
histPlot . . . . .	125
house . . . . .	127
housing . . . . .	129
hsb2 . . . . .	129
husbands_wives . . . . .	130
immigration . . . . .	131
IMSCOL . . . . .	132
infmortrate . . . . .	132
iowa . . . . .	133
ipo . . . . .	134
ipod . . . . .	135
iran . . . . .	136
jury . . . . .	137
kobe_basket . . . . .	138
labor_market_discriminiation . . . . .	138
lab_report . . . . .	141
LAhomes . . . . .	142
law_resume . . . . .	143
lecture_learning . . . . .	144
lego_population . . . . .	145
lego_sample . . . . .	147
leg_mari . . . . .	148
life_exp . . . . .	149
linResPlot . . . . .	149
lizard_habitat . . . . .	151
lizard_run . . . . .	152
lmPlot . . . . .	153
loans_full_schema . . . . .	155

london_boroughs . . . . .	157
london_murders . . . . .	158
loop . . . . .	160
lsegments . . . . .	160
mail_me . . . . .	162
major_survey . . . . .	163
makeTube . . . . .	163
malaria . . . . .	165
male_heights . . . . .	166
male_heights_fcid . . . . .	167
mammals . . . . .	167
mammogram . . . . .	169
manhattan . . . . .	169
marathon . . . . .	170
mariokart . . . . .	171
mcu_films . . . . .	173
midterms_house . . . . .	174
migraine . . . . .	175
military . . . . .	175
mlb . . . . .	177
mlbbat10 . . . . .	178
mlb_players_18 . . . . .	179
mlb_teams . . . . .	181
mn_police_use_of_force . . . . .	183
MosaicPlot . . . . .	184
movies . . . . .	185
mtl . . . . .	186
murders . . . . .	187
myPDF . . . . .	188
nba_finals . . . . .	189
nba_finals_teams . . . . .	191
nba_heights . . . . .	192
nba_players_19 . . . . .	193
ncbirths . . . . .	193
normTail . . . . .	195
nuclear_survey . . . . .	197
nyc . . . . .	197
nycflights . . . . .	198
nyc_marathon . . . . .	199
offshore_drilling . . . . .	200
openintro_colors . . . . .	201
openintro_cols . . . . .	201
openintro_pal . . . . .	202
openintro_palettes . . . . .	202
opportunity_cost . . . . .	203
orings . . . . .	204
oscars . . . . .	205
outliers . . . . .	206

paralympic_1500 . . . . .	207
penelope . . . . .	208
penetrating_oil . . . . .	209
penny_ages . . . . .	210
pew_energy_2018 . . . . .	211
photo_classify . . . . .	212
piracy . . . . .	212
playing_cards . . . . .	214
PlotWLine . . . . .	215
pm25_2011_durham . . . . .	216
pm25_2022_durham . . . . .	217
poker . . . . .	219
possum . . . . .	219
ppp_201503 . . . . .	220
present . . . . .	221
president . . . . .	222
prison . . . . .	222
prius_mpg . . . . .	223
qqnormsim . . . . .	224
race_justice . . . . .	224
reddit_finance . . . . .	226
resume . . . . .	229
res_demo_1 . . . . .	232
res_demo_2 . . . . .	233
rosling_responses . . . . .	233
russian_influence_on_us_election_2016 . . . . .	234
salinity . . . . .	235
satgpa . . . . .	236
sat_improve . . . . .	237
sa_gdp_elec . . . . .	238
scale_color_openintro . . . . .	239
scale_fill_openintro . . . . .	240
scotus_healthcare . . . . .	241
seattlepets . . . . .	242
sex_discrimination . . . . .	242
simpsons_paradox_covid . . . . .	243
simulated_dist . . . . .	244
simulated_normal . . . . .	245
simulated_scatter . . . . .	245
sinusitis . . . . .	246
sleep_deprivation . . . . .	246
smallpox . . . . .	247
smoking . . . . .	248
snowfall . . . . .	249
socialexp . . . . .	250
soda . . . . .	251
solar . . . . .	251
sowc_child_mortality . . . . .	252

sowc_demographics . . . . .	253
sowc_maternal_newborn . . . . .	255
sp500 . . . . .	256
sp500_1950_2018 . . . . .	258
sp500_seq . . . . .	259
speed_gender_height . . . . .	259
ssd_speed . . . . .	260
starbucks . . . . .	261
stats_scores . . . . .	261
stem_cell . . . . .	262
stent30 . . . . .	263
stocks_18 . . . . .	263
student_housing . . . . .	264
student_sleep . . . . .	265
sulphinpyrazone . . . . .	265
supreme_court . . . . .	266
teacher . . . . .	267
textbooks . . . . .	268
thanksgiving_spend . . . . .	269
tips . . . . .	269
toohey . . . . .	270
tourism . . . . .	271
toy_anova . . . . .	272
transplant . . . . .	272
treeDiag . . . . .	273
twins . . . . .	274
ucla_f18 . . . . .	275
ucla_textbooks_f18 . . . . .	276
ukdemo . . . . .	278
unempl . . . . .	279
unemploy_pres . . . . .	280
usb_admit . . . . .	281
us_temperature . . . . .	282
winery_cars . . . . .	283
world_pop . . . . .	284
write_pkg_data . . . . .	287
xom . . . . .	288
yawn . . . . .	289
yrbss . . . . .	289
yrbss_samp . . . . .	290

---

absenteeism

*Absenteeism from school in New South Wales*

---

## Description

Researchers interested in the relationship between absenteeism from school and certain demographic characteristics of children collected data from 146 randomly sampled students in rural New South Wales, Australia, in a particular school year.

## Usage

absenteeism

## Format

A data frame with 146 observations on the following 5 variables.

**eth** Ethnicity, representing Aboriginal (A) or not (N).

**sex** Gender.

**age** Age bucket.

**lrn** Learner status, with average learner (AL) and slow learner (SL).

**days** Number of days absent.

## Source

Venables WN, Ripley BD. 2002. Modern Applied Statistics with S. Fourth Edition. New York: Springer.

Data can also be found in the R MASS package under the dataset name quine.

## Examples

```
library(ggplot2)

ggplot(absenteeism, aes(x = eth, y = days)) +
  geom_boxplot() +
  coord_flip()
```



---

acs12*American Community Survey, 2012*

---

**Description**

Results from the US Census American Community Survey, 2012.

**Usage**

acs12

**Format**

A data frame with 2000 observations on the following 13 variables.

**income** Annual income.

**employment** Employment status.

**hrs\_work** Hours worked per week.

**race** Race.

**age** Age, in years.

**gender** Gender.

**citizen** Whether the person is a U.S. citizen.

**time\_to\_work** Travel time to work, in minutes.

**lang** Language spoken at home.

**married** Whether the person is married.

**edu** Education level.

**disability** Whether the person is disabled.

**birth\_qtr** The quarter of the year that the person was born, e.g. Jan thru Mar.

**Source**

<https://www.census.gov/programs-surveys/acs>

**Examples**

```
library(dplyr)
library(ggplot2)
library(broom)

# employed only
acs12_emp <- acs12 |>
  filter(
    age >= 30, age <= 60,
    employment == "employed",
    income > 0
```

```

)

# linear model
ggplot(acs12_emp, mapping = aes(x = age, y = income)) +
  geom_point() +
  geom_smooth(method = "lm")

lm(income ~ age, data = acs12_emp) |>
  tidy()

# log-transformed model
ggplot(acs12_emp, mapping = aes(x = age, y = log(income))) +
  geom_point() +
  geom_smooth(method = "lm")

lm(log(income) ~ age, data = acs12_emp) |>
  tidy()

```

---

age_at_mar	<i>Age at first marriage of 5,534 US women.</i>
------------	---

---

## Description

Age at first marriage of 5,534 US women who responded to the National Survey of Family Growth (NSFG) conducted by the CDC in the 2006 and 2010 cycle.

## Usage

```
age_at_mar
```

## Format

A data frame with 5,534 observations and 1 variable.

**age** Age a first marriage.

## Source

National Survey of Family Growth, 2006-2010 cycle, [https://www.cdc.gov/nchs/nsfg/nsfg\\_2006\\_2010\\_puf.htm](https://www.cdc.gov/nchs/nsfg/nsfg_2006_2010_puf.htm).

## Examples

```

library(ggplot2)

ggplot(age_at_mar, mapping = aes(x = age)) +
  geom_histogram(binwidth = 3) +
  labs(
    x = "Age", y = "Count", title = "Age at first marriage, US Women",
    subtitle = "Source: National Survey of Family Growth Survey, 2006 - 2010"
  )

```

ames

*Housing prices in Ames, Iowa***Description**

Data set contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. See [here](#) for detailed variable descriptions.

**Usage**

ames

**Format**

A `tbl_df` with 2930 rows and 82 variables:

**Order** Observation number.

**PID** Parcel identification number - can be used with city web site for parcel review.

**area** Above grade (ground) living area square feet.

**price** Sale price in USD.

**MS.SubClass** Identifies the type of dwelling involved in the sale.

**MS.Zoning** Identifies the general zoning classification of the sale.

**Lot.Frontage** Linear feet of street connected to property.

**Lot.Area** Lot size in square feet.

**Street** Type of road access to property.

**Alley** Type of alley access to property.

**Lot.Shape** General shape of property.

**Land.Contour** Flatness of the property.

**Utilities** Type of utilities available.

**Lot.Config** Lot configuration.

**Land.Slope** Slope of property.

**Neighborhood** Physical locations within Ames city limits (map available).

**Condition.1** Proximity to various conditions.

**Condition.2** Proximity to various conditions (if more than one is present).

**Bldg.Type** Type of dwelling.

**House.Style** Style of dwelling.

**Overall.Qual** Rates the overall material and finish of the house.

**Overall.Cond** Rates the overall condition of the house.

**Year.Built** Original construction date.

**Year.Remod.Add** Remodel date (same as construction date if no remodeling or additions).

**Roof.Style** Type of roof.

**Roof.Matl** Roof material.

**Exterior.1st** Exterior covering on house.

**Exterior.2nd** Exterior covering on house (if more than one material).

**Mas.Vnr.Type** Masonry veneer type.

**Mas.Vnr.Area** Masonry veneer area in square feet.

**Exter.Qual** Evaluates the quality of the material on the exterior.

**Exter.Cond** Evaluates the present condition of the material on the exterior.

**Foundation** Type of foundation.

**Bsmt.Qual** Evaluates the height of the basement.

**Bsmt.Cond** Evaluates the general condition of the basement.

**Bsmt.Exposure** Refers to walkout or garden level walls.

**BsmtFin.Type.1** Rating of basement finished area.

**BsmtFin.SF.1** Type 1 finished square feet.

**BsmtFin.Type.2** Rating of basement finished area (if multiple types).

**BsmtFin.SF.2** Type 2 finished square feet.

**Bsmt.Unf.SF** Unfinished square feet of basement area.

**Total.Bsmt.SF** Total square feet of basement area.

**Heating** Type of heating.

**Heating.QC** Heating quality and condition.

**Central.Air** Central air conditioning.

**Electrical** Electrical system.

**X1st.Flr.SF** First Floor square feet.

**X2nd.Flr.SF** Second floor square feet.

**Low.Qual.Fin.SF** Low quality finished square feet (all floors).

**Bsmt.Full.Bath** Basement full bathrooms.

**Bsmt.Half.Bath** Basement half bathrooms.

**Full.Bath** Full bathrooms above grade.

**Half.Bath** Half baths above grade.

**Bedroom.AbvGr** Bedrooms above grade (does NOT include basement bedrooms).

**Kitchen.AbvGr** Kitchens above grade.

**Kitchen.Qual** Kitchen quality.

**TotRms.AbvGrd** Total rooms above grade (does not include bathrooms).

**Functional** Home functionality (Assume typical unless deductions are warranted).

**Fireplaces** Number of fireplaces.

**Fireplace.Qu** Fireplace quality.

**Garage.Type** Garage location.  
**Garage.Yr.Blt** Year garage was built.  
**Garage.Finish** Interior finish of the garage.  
**Garage.Cars** Size of garage in car capacity.  
**Garage.Area** Size of garage in square feet.  
**Garage.Qual** Garage quality.  
**Garage.Cond** Garage condition.  
**Paved.Drive** Paved driveway.  
**Wood.Deck.SF** Wood deck area in square feet.  
**Open.Porch.SF** Open porch area in square feet.  
**Enclosed.Porch** Enclosed porch area in square feet.  
**X3Ssn.Porch** Three season porch area in square feet.  
**Screen.Porch** Screen porch area in square feet.  
**Pool.Area** Pool area in square feet.  
**Pool.QC** Pool quality.  
**Fence** Fence quality.  
**Misc.Feature** Miscellaneous feature not covered in other categories.  
**Misc.Val** Dollar value of miscellaneous feature.  
**Mo.Sold** Month Sold (MM).  
**Yr.Sold** Year Sold (YYYY).  
**Sale.Type** Type of sale.  
**Sale.Condition** Condition of sale.

### Source

De Cock, Dean. "Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project." *Journal of Statistics Education* 19.3 (2011).

---

 ami\_occurrences

---

*Acute Myocardial Infarction (Heart Attack) Events*


---

### Description

This dataset is simulated but contains realistic occurrences of AMI in NY City.

### Usage

ami\_occurrences

**Format**

A data frame with 365 observations on the following variable.

**ami** Number of daily occurrences of heart attacks in NY City.

**Examples**

```
library(ggplot2)

ggplot(ami_occurrences, mapping = aes(x = ami)) +
  geom_bar() +
  labs(
    x = "Acute Myocardial Infarction events",
    y = "Count",
    title = "Acute Myocardial Infarction events in NYC"
  )
```

---

antibiotics

---

*Pre-existing conditions in 92 children*


---

**Description**

Pre-existing medical conditions of 92 children involved in a study on the optimal duration of antibiotic use in treatment of tracheitis, which is an upper respiratory infection.

**Usage**

```
antibiotics
```

**Format**

A data frame with 92 observations, each representing a child, on the following variable.

**condition** Pre-existing medical condition.

**Examples**

```
library(ggplot2)

ggplot(antibiotics, aes(x = condition)) +
  geom_bar() +
  labs(
    x = "Conidition", y = "Count",
    title = "Pre-existing coniditions of children",
    subtitle = "in antibiotic use study"
  ) +
  coord_flip()
```

---

arbuthnot*Male and female births in London*

---

**Description**

Arbuthnot's data describes male and female christenings (births) for London from 1629-1710.

**Usage**

arbuthnot

**Format**

A `tbl_df` with 82 rows and 3 variables:

**year** year, ranging from 1629 to 1710

**boys** number of male christenings (births)

**girls** number of female christenings (births)

**Details**

John Arbuthnot (1710) used these time series data to carry out the first known significance test. During every one of the 82 years, there were more male christenings than female christenings. As Arbuthnot wondered, we might also wonder if this could be due to chance, or whether it meant the birth ratio was not actually 1:1.

**Source**

These data are excerpted from the Arbuthnot dataset in the [HistData](#) package.

**Examples**

```
library(ggplot2)
library(tidyr)

# All births
ggplot(arbuthnot, aes(x = year, y = boys + girls, group = 1)) +
  geom_line()

# Boys and girls
arbuthnot |>
  pivot_longer(cols = -year, names_to = "sex", values_to = "n") |>
  ggplot(aes(x = year, y = n, color = sex, group = sex)) +
  geom_line()
```

## ArrowLines

*Create a Line That may have Arrows on the Ends***Description**

Similar to [lines](#), this function will include endpoints that are solid points, open points, or arrows (mix-and-match ready).

**Usage**

```
ArrowLines(
  x,
  y,
  lty = 1,
  lwd = 2.5,
  col = 1,
  length = 0.1,
  af = 3,
  cex.pch = 1.2,
  ends = c("a", "a"),
  ...
)
```

**Arguments**

x	A vector of the x-coordinates of the line to be drawn.
y	A vector of the y-coordinates of the line to be drawn. This vector should have the same length as that of x.
lty	The line type.
lwd	The line width.
col	The line and endpoint color.
length	If an end point is an arrow, then this specifies the sizing of the arrow. See the <a href="#">length</a> argument in the <a href="#">arrows</a> help file for additional details.
af	A tuning parameter for creating the arrow. Usually the default (3) will work. If no arrow is shown, make this value larger. If the arrow appears to extend off of the line, then specify a smaller value.
cex.pch	Plotting character size (if open or closed point at the end).
ends	A character vector of length 2, where the first value corresponds to the start of the line and the second to the end of the line. A value of "a" corresponds to an arrow being shown, "o" to an open circle, and "c" for a closed point.
...	All additional arguments are passed to the <a href="#">lines</a> function.

**Author(s)**

David Diez



**See Also**

[lsegments](#), [dlsegments](#), [CCP](#)

**Examples**

```
CCP(xlim = c(-6, 6), ylim = c(-6, 6), ticklabs = 2)
x <- c(-2, 0, 2, 4)
y <- c(0, 3, 0, 3)
ArrowLines(x, y, col = COL[1], ends = c("c", "c"))
points(x, y, col = COL[1], pch = 19, cex = 1.2)
```

```
CCP(xlim = c(-6, 6), ylim = c(-6, 6), ticklabs = 2)
x <- c(-3, 0, 1, 3)
y <- c(2, 1, -2, 1)
ArrowLines(x, y, col = COL[1], ends = c("c", "c"))
points(x, y, col = COL[1], pch = 19, cex = 1.2)
```

```
CCP(xlim = c(-6, 6), ylim = c(-6, 6), ticklabs = 2)
x <- seq(-2, 2, 0.01)
y <- x^2 - 3
ArrowLines(x, y, col = COL[1], ends = c("c", "c"))
x <- seq(-2, 2, 1)
y <- x^2 - 3
points(x, y, col = COL[1], pch = 19, cex = 1.2)
```

---

ask

---

*How important is it to ask pointed questions?*


---

**Description**

In this experiment, each individual was asked to be a seller of an iPod (a product commonly used to store music on before smart phones...). They participant received \$10 + 5% of the sale price for participating. The iPod they were selling had frozen twice in the past inexplicably but otherwise worked fine. The prospective buyer starts off and then asks one of three final questions, depending on the seller's treatment group.

**Usage**

```
ask
```

**Format**

A data frame with 219 observations on the following 3 variables.

**question\_class** The type of question: general, pos\_assumption, and neg\_assumption.

**question** The question corresponding to the question.class

**response** The classified response from the seller, either disclose or hide.

## Details

The three possible questions:

- General: What can you tell me about it?
- Positive Assumption: It doesn't have any problems, does it?
- Negative Assumption: What problems does it have?

The outcome variable is whether or not the participant discloses or hides the problem with the iPod.

## Source

Minson JA, Ruedy NE, Schweitzer ME. There *is* such a thing as a stupid question: Question disclosure in strategic communication.

## Examples

```
library(dplyr)
library(ggplot2)

# Distribution of responses based on question type
ask |>
  count(question_class, response)

# Visualize relative frequencies of responses based on question type
ggplot(ask, aes(x = question_class, fill = response)) +
  geom_bar(position = "fill")

# Perform chi-square test
(test <- chisq.test(table(ask$question_class, ask$response)))

# Check the test's assumption around sufficient expected observations
# per table cell.
test$expected
```

---

association

*Simulated data for association plots*

---

## Description

Simulated dataset.

## Usage

```
association
```

**Format**

A data frame with 121 observations on the following 4 variables.

**x1** a numeric vector  
**x2** a numeric vector  
**x3** a numeric vector  
**y1** a numeric vector  
**y2** a numeric vector  
**y3** a numeric vector  
**y4** a numeric vector  
**y5** a numeric vector  
**y6** a numeric vector  
**y7** a numeric vector  
**y8** a numeric vector  
**y9** a numeric vector  
**y10** a numeric vector  
**y11** a numeric vector  
**y12** a numeric vector

**Examples**

```
library(ggplot2)

ggplot(association, aes(x = x1, y = y1)) +
  geom_point()

ggplot(association, aes(x = x2, y = y4)) +
  geom_point()

ggplot(association, aes(x = x3, y = y7)) +
  geom_point()
```

---

assortive\_mating

*Eye color of couples*

---

**Description**

Colors of the eye colors of male and female partners.

**Usage**

assortative\_mating

**Format**

A data frame with 204 observations on the following 2 variables.

**self\_male** a factor with levels blue, brown, and green

**partner\_female** a factor with blue, brown, and green

**Source**

B. Laeng et al. Why do blue-eyed men prefer women with the same eye color? In: Behavioral Ecology and Sociobiology 61.3 (2007), pp. 371-384.

**Examples**

```
data(assortive_mating)
table(assortive_mating)
```

---

avandia

*Cardiovascular problems for two types of Diabetes medicines*

---

**Description**

A comparison of cardiovascular problems for Rosiglitazone and Pioglitazone.

**Usage**

```
avandia
```

**Format**

A data frame with 227571 observations on the following 2 variables.

**treatment** a factor with levels Pioglitazone and Rosiglitazone

**cardiovascular\_problems** a factor with levels no and yes

**Source**

D.J. Graham et al. Risk of acute myocardial infarction, stroke, heart failure, and death in elderly Medicare patients treated with rosiglitazone or pioglitazone. In: JAMA 304.4 (2010), p. 411. issn: 0098-7484.

**Examples**

```
table(avandia)
```

**Description**

Convert and simplify axis labels that are in US Dollars.

**Usage**

```
AxisInDollars(side, at, include.symbol = TRUE, simplify = TRUE, ...)
```

**Arguments**

side	An integer specifying which side of the plot the axis is to be drawn on. The axis is placed as follows: 1 = below, 2 = left, 3 = above and 4 = right.
at	The points at which tick-marks are to be drawn.
include.symbol	Whether to include a dollar or percent symbol, where the symbol chosen depends on the function.
simplify	For dollars, simplify the amount to use abbreviations of "k", "m", "b", or "t" when numbers tend to be in the thousands, millions, billions, or trillions, respectively.
...	Arguments passed to <a href="#">axis</a>

**Value**

The numeric locations on the axis scale at which tick marks were drawn when the plot was first drawn.

**Author(s)**

David Diez

**See Also**

[buildAxis](#) [AxisInDollars](#) [AxisInPercent](#)

**Examples**

```
x <- sample(50e6, 100)
hist(x, axes = FALSE)
AxisInDollars(1, pretty(x))
```

AxisInPercent

*Build Better Looking Axis Labels for Percentages*

---

**Description**

Convert and simplify axis labels that are in percentages.

**Usage**

```
AxisInPercent(side, at, include.symbol = TRUE, simplify = TRUE, ...)
```

**Arguments**

side	An integer specifying which side of the plot the axis is to be drawn on. The axis is placed as follows: 1 = below, 2 = left, 3 = above and 4 = right.
at	The points at which tick-marks are to be drawn.
include.symbol	Whether to include a dollar or percent symbol, where the symbol chosen depends on the function.
simplify	For dollars, simplify the amount to use abbreviations of "k", "m", "b", or "t" when numbers tend to be in the thousands, millions, billions, or trillions, respectively.
...	Arguments passed to <a href="#">axis</a>

**Value**

The numeric locations on the axis scale at which tick marks were drawn when the plot was first drawn.

**Author(s)**

David Diez

**See Also**

[buildAxis](#) [AxisInDollars](#) [AxisInDollars](#)

**Examples**

```
x <- sample(50e6, 100)
hist(x, axes = FALSE)
AxisInDollars(1, pretty(x))
```

babies

*The Child Health and Development Studies***Description**

The Child Health and Development Studies investigate a range of topics. One study, in particular, considered all pregnancies between 1960 and 1967 among women in the Kaiser Foundation Health Plan in the San Francisco East Bay area. We do not have ideal provenance for these data. For a better documented and more recent dataset on a similar topic with similar variables, see [births14](#). Additionally, Gestation dataset in the [mosaicData](#) package also contains similar data.

**Usage**

babies

**Format**

A data frame with 1236 rows and 8 variables:

**case** id number

**bwt** birthweight, in ounces

**gestation** length of gestation, in days

**parity** binary indicator for a first pregnancy (0 = first pregnancy)

**age** mother's age in years

**height** mother's height in inches

**weight** mother's weight in pounds

**smoke** binary indicator for whether the mother smokes

**Source**

These data come from Child Health and Development Studies.

babies\_crawl

*Crawling age***Description**

Crawling age of babies along with the average outdoor temperature at 6 months of age.

**Usage**

babies\_crawl

**Format**

A data frame with 12 observations on the following 5 variables.

**birth\_month** A factor with levels corresponding to months

**avg\_crawling\_age** a numeric vector

**sd** a numeric vector

**n** a numeric vector

**temperature** a numeric vector

**Source**

J.B. Benson. Season of birth and onset of locomotion: Theoretical and methodological implications. In: Infant behavior and development 16.1 (1993), pp. 69-81. issn: 0163-6383.

**Examples**

```
library(ggplot2)

ggplot(babies_crawl, aes(x = temperature, y = avg_crawling_age)) +
  geom_point() +
  labs(x = "Temperature", y = "Average crawling age")
```

---

 bac

---

*Beer and blood alcohol content*


---

**Description**

Here we examine data from sixteen student volunteers at Ohio State University who each drank a randomly assigned number of cans of beer.

**Usage**

bac

**Format**

A data frame with 16 observations on the following 3 variables.

**student** a numeric vector

**beers** a numeric vector

**bac** a numeric vector

**Source**

J. Malkevitch and L.M. Lesser. For All Practical Purposes: Mathematical Literacy in Today's World. WH Freeman & Co, 2008. The data origin is given in the [Electronic Encyclopedia of Statistical Examples and Exercises](#), 1992.



**Examples**

```
library(ggplot2)

ggplot(bac, aes(x = beers, y = bac)) +
  geom_point() +
  labs(x = "Number of beers", y = "Blood alcohol content")
```

---

ball_bearing	<i>Lifespan of ball bearings</i>
--------------	----------------------------------

---

**Description**

A simulated dataset on lifespan of ball bearings.

**Usage**

```
ball_bearing
```

**Format**

A data frame with 75 observations on the following variable.

**life\_span** Lifespan of ball bearings (in hours).

**Source**

Simulated data.

**Examples**

```
library(ggplot2)

ggplot(ball_bearing, aes(x = life_span)) +
  geom_histogram(binwidth = 1)

qqnorm(ball_bearing$life_span)
```

bdims

*Body measurements of 507 physically active individuals.***Description**

Body girth measurements and skeletal diameter measurements, as well as age, weight, height and gender, are given for 507 physically active individuals - 247 men and 260 women. These data can be used to provide statistics students practice in the art of data analysis. Such analyses range from simple descriptive displays to more complicated multivariate analyses such as multiple regression and discriminant analysis.

**Usage**

bdims

**Format**

A data frame with 507 observations on the following 25 variables.

**bia\_di** A numerical vector, respondent's biacromial diameter in centimeters.

**bii\_di** A numerical vector, respondent's biiliac diameter (pelvic breadth) in centimeters.

**bit\_di** A numerical vector, respondent's bitrochanteric diameter in centimeters.

**che\_de** A numerical vector, respondent's chest depth in centimeters, measured between spine and sternum at nipple level, mid-expiration.

**che\_di** A numerical vector, respondent's chest diameter in centimeters, measured at nipple level, mid-expiration.

**elb\_di** A numerical vector, respondent's elbow diameter in centimeters, measured as sum of two elbows.

**wri\_di** A numerical vector, respondent's wrist diameter in centimeters, measured as sum of two wrists.

**kne\_di** A numerical vector, respondent's knee diameter in centimeters, measured as sum of two knees.

**ank\_di** A numerical vector, respondent's ankle diameter in centimeters, measured as sum of two ankles.

**sho\_gi** A numerical vector, respondent's shoulder girth in centimeters, measured over deltoid muscles.

**che\_gi** A numerical vector, respondent's chest girth in centimeters, measured at nipple line in males and just above breast tissue in females, mid-expiration.

**wai\_gi** A numerical vector, respondent's waist girth in centimeters, measured at the narrowest part of torso below the rib cage as average of contracted and relaxed position.

**nav\_gi** A numerical vector, respondent's navel (abdominal) girth in centimeters, measured at umbilicus and iliac crest using iliac crest as a landmark.

**hip\_gi** A numerical vector, respondent's hip girth in centimeters, measured at at level of bitrochanteric diameter.

**thi\_gi** A numerical vector, respondent's thigh girth in centimeters, measured below gluteal fold as the average of right and left girths.

**bic\_gi** A numerical vector, respondent's bicep girth in centimeters, measured when flexed as the average of right and left girths.

**for\_gi** A numerical vector, respondent's forearm girth in centimeters, measured when extended, palm up as the average of right and left girths.

**kne\_gi** A numerical vector, respondent's knee diameter in centimeters, measured as sum of two knees.

**cal\_gi** A numerical vector, respondent's calf maximum girth in centimeters, measured as average of right and left girths.

**ank\_gi** A numerical vector, respondent's ankle minimum girth in centimeters, measured as average of right and left girths.

**wri\_gi** A numerical vector, respondent's wrist minimum girth in centimeters, measured as average of right and left girths.

**age** A numerical vector, respondent's age in years.

**wgt** A numerical vector, respondent's weight in kilograms.

**hgt** A numerical vector, respondent's height in centimeters.

**sex** A categorical vector, 1 if the respondent is male, 0 if female.

## Source

Heinz G, Peterson LJ, Johnson RW, Kerk CJ. 2003. Exploring Relationships in Body Dimensions. Journal of Statistics Education 11(2).

## Examples

```
library(ggplot2)
ggplot(bdims, aes(x = hgt)) +
  geom_histogram(binwidth = 5)

ggplot(bdims, aes(x = hgt, y = wgt)) +
  geom_point() +
  labs(x = "Height", y = "Weight")

ggplot(bdims, aes(x = hgt, y = sho_gi)) +
  geom_point() +
  labs(x = "Height", y = "Shoulder girth")

ggplot(bdims, aes(x = hgt, y = hip_gi)) +
  geom_point() +
  labs(x = "Height", y = "Hip girth")
```

---

BG*Add background color to a plot*

---

**Description**

Overlays a colored rectangle over the entire plotting region.

**Usage**

```
BG(col = openintro::COL[5, 9])
```

**Arguments**

col                      Color to overlay.

**See Also**

[COL](#)

**Examples**

```
Test <- function(col) {  
  plot(1:7,  
    col = COL[1:7], pch = 19, cex = 5,  
    xlim = c(0, 8),  
    ylim = c(0, 9)  
  )  
  BG(col)  
  points(2:8, col = COL[1:7], pch = 19, cex = 5)  
  text(2, 6, "Correct Color")  
  text(6, 2, "Affected Color")  
}  
  
# Works well since black color almost fully transparent  
Test(COL[5, 9])  
  
# Works less well since transparency isn't as significant  
Test(COL[5, 6])  
  
# Pretty ugly due to overlay  
Test(COL[5, 3])  
  
# Basically useless due to heavy color gradient  
Test(COL[4, 2])
```

---

biontech\_adolescents    *Efficacy of Pfizer-BioNTech COVID-19 vaccine on adolescents*

---

## Description

On March 31, 2021, Pfizer and BioNTech announced that "in a Phase 3 trial in adolescents 12 to 15 years of age with or without prior evidence of SARS-CoV-2 infection, the Pfizer-BioNTech COVID-19 vaccine BNT162b2 demonstrated 100% efficacy and robust antibody responses, exceeding those recorded earlier in vaccinated participants aged 16 to 25 years old, and was well tolerated." These results are from a Phase 3 trial in 2,260 adolescents 12 to 15 years of age in the United States. In the trial, 18 cases of COVID-19 were observed in the placebo group (n = 1,129) versus none in the vaccinated group (n = 1,131).

## Usage

biontech\_adolescents

## Format

A data frame with 2260 observations on the following 2 variables.

**group** Study group: vaccine (Pfizer-BioNTech COVID-19 vaccine administered) or placebo.

**outcome** Study outcome: COVID-19 or no COVID-19.

## Source

"Pfizer-Biontech Announce Positive Topline Results Of Pivotal Covid-19 Vaccine Study In Adolescents". March 21, 2021. (Retrieved April 25, 2021.)

## Examples

```
library(dplyr)
library(ggplot2)

biontech_adolescents |>
  count(group, outcome)

ggplot(biontech_adolescents, aes(y = group, fill = outcome)) +
  geom_bar()
```

birds

*Aircraft-Wildlife Collisions***Description**

A collection of all collisions between aircraft in wildlife that were reported to the US Federal Aviation Administration between 1990 and 1997, with details on the circumstances of the collision.

**Usage**

birds

**Format**

A data frame with 19302 observations on the following 17 variables.

**opid** Three letter identification code for the operator (carrier) of the aircraft.

**operator** Name of the aircraft operator.

**atype** Make and model of aircraft.

**remarks** Verbal remarks regarding the collision.

**phase\_of\_flight** Phase of the flight during which the collision occurred: Approach, Climb, Descent, En Route, Landing Roll, Parked, Take-off run, Taxi.

**ac\_mass** Mass of the aircraft classified as 2250 kg or less (1), 2251-5700 kg (2), 5701-27000 kg (3), 27001-272000 kg (4), above 272000 kg (5).

**num\_engs** Number of engines on the aircraft.

**date** Date of the collision (MM/DD/YYYY).

**time\_of\_day** Light conditions: Dawn, Day, Dusk, Night.

**state** Two letter abbreviation of the US state in which the collision occurred.

**height** Feet above ground level.

**speed** Knots (indicated air speed).

**effect** Effect on flight: Aborted Take-off, Engine Shut Down, None, Other, Precautionary Landing.

**sky** Type of cloud cover, if any: No Cloud, Overcast, Some Cloud.

**species** Common name for bird or other wildlife.

**birds\_seen** Number of birds/wildlife seen by pilot: 1, 2-10, 11-100, Over 100.

**birds\_struck** Number of birds/wildlife struck: 0, 1, 2-10, 11-100, Over 100.

**Details**

The FAA National Wildlife Strike Database contains strike reports that are voluntarily reported to the FAA by pilots, airlines, airports and others. Current research indicates that only about 20% Wildlife strike reporting is not uniform as some organizations have more robust voluntary reporting procedures. Because of variations in reporting, users are cautioned that the comparisons between individual airports or airlines may be misleading.

**Source**

Aircraft Wildlife Strike Data: Search Tool - FAA Wildlife Strike Database. Available at <https://datahub.transportation.gov/Aviation/Aircraft-Wildlife-Strike-Data-Search-Tool-FAA-Wild/jhay-dgxy>. Retrieval date: Feb 4, 2012.

**Examples**

```
library(dplyr)
library(ggplot2)
library(forcats)
library(tidyr)

# Phase of the flight during which the collision occurred, tabular
birds |>
  count(phase_of_flt, sort = TRUE)

# Phase of the flight during which the collision occurred, barplot
ggplot(birds, aes(y = fct_infreq(phase_of_flt))) +
  geom_bar() +
  labs(x = "Phase of flight")

# Height summary statistics
summary(birds$height)

# Phase of flight vs. effect of crash
birds |>
  drop_na(phase_of_flt, effect) |>
  ggplot(aes(y = phase_of_flt, fill = effect)) +
  geom_bar(position = "fill") +
  labs(x = "Proportion", y = "Phase of flight", fill = "Effect")
```

---

births

*North Carolina births, 100 cases*


---

**Description**

Data on a random sample of 100 births for babies in North Carolina where the mother was not a smoker and another 50 where the mother was a smoker.

**Usage**

```
births
```

**Format**

A data frame with 150 observations on the following 14 variables.

**f\_age** Father's age.

**m\_age** Mother's age.

**weeks** Weeks at which the mother gave birth.  
**premature** Indicates whether the baby was premature or not.  
**visits** Number of hospital visits.  
**gained** Weight gained by mother.  
**weight** Birth weight of the baby.  
**sex\_baby** Gender of the baby.  
**smoke** Whether or not the mother was a smoker.

### Source

Birth records released by North Carolina in 2004.

### See Also

We do not have ideal provenance for these data. For a better documented and more recent dataset on a similar topic with similar variables, see [births14](#). Additionally, [ncbirths](#) also contains similar data.

### Examples

```
library(ggplot2)

ggplot(births, aes(x = smoke, y = weight)) +
  geom_boxplot()
```

---

births14

*US births*

---

### Description

Every year, the US releases to the public a large dataset containing information on births recorded in the country. This dataset has been of interest to medical researchers who are studying the relation between habits and practices of expectant mothers and the birth of their children. This is a random sample of 1,000 cases from the dataset released in 2014.

### Usage

```
births14
```

### Format

A data frame with 1,000 observations on the following 13 variables.

**fage** Father's age in years.  
**mage** Mother's age in years.  
**mature** Maturity status of mother.



**weeks** Length of pregnancy in weeks.

**premie** Whether the birth was classified as premature (premie) or full-term.

**visits** Number of hospital visits during pregnancy.

**gained** Weight gained by mother during pregnancy in pounds.

**weight** Weight of the baby at birth in pounds.

**lowbirthweight** Whether baby was classified as low birthweight (low) or not (not low).

**sex** Sex of the baby, female or male.

**habit** Status of the mother as a nonsmoker or a smoker.

**marital** Whether mother is married or not married at birth.

**whitemom** Whether mom is white or not white.

### Source

United States Department of Health and Human Services. Centers for Disease Control and Prevention. National Center for Health Statistics. Natality Detail File, 2014 United States. Inter-university Consortium for Political and Social Research, 2016-10-07. [doi:10.3886/ICPSR36461.v1](https://doi.org/10.3886/ICPSR36461.v1).

### Examples

```
library(ggplot2)

ggplot(births14, aes(x = habit, y = weight)) +
  geom_boxplot() +
  labs(x = "Smoking status of mother", y = "Birth weight of baby (in lbs)")

ggplot(births14, aes(x = whitemom, y = visits)) +
  geom_boxplot() +
  labs(x = "Mother's race", y = "Number of doctor visits during pregnancy")

ggplot(births14, aes(x = mature, y = gained)) +
  geom_boxplot() +
  labs(x = "Mother's age category", y = "Weight gained during pregnancy")
```

---

blizzard\_salary

*Blizzard Employee Voluntary Salary Info.*


---

### Description

Employee generated anonymous survey of salary information.

### Usage

```
blizzard_salary
```

**Format**

A data frame with 466 rows and 9 variables.

**timestamp** Time data was entered

**status** Specifies employment status.

**current\_title** Current job title.

**current\_salary** Current salary (in USD).

**salary\_type** Frequency with levels year, hour, week.

**percent\_incr** Raise given July 2020.

**other\_info** Other information submitted by employee.

**location** Current office of employment.

**performance\_rating** Most recent review performance rating.

**Source**

**Bloomberg - Blizzard workers share salaries in revolt over wage disparities.**

**Examples**

```
library(ggplot2)
library(dplyr)

plot_data <- blizzard_salary |>
  mutate(annual_salary = case_when(
    salary_type == "week" ~ current_salary * 52,
    salary_type == "hour" ~ current_salary * 40 * 52,
    TRUE ~ current_salary
  ))

ggplot(plot_data, aes(annual_salary)) +
  geom_histogram(binwidth = 25000, color = "white") +
  labs(
    title = "Current Salary of Blizzard Employees",
    x = "Salary",
    y = "Number of Employees"
  )
```

---

books

*Sample of books on a shelf*

---

**Description**

Simulated dataset.

**Usage**

```
books
```

**Format**

A data frame with 95 observations on the following 2 variables.

**type** a factor with levels fiction and nonfiction

**format** a factor with levels hardcover and paperback

**Examples**

```
table(books)
```

---

boxPlot

*Box plot*


---

**Description**

An alternative to boxplot. Equations are not accepted. Instead, the second argument, `fact`, is used to split the data.

**Usage**

```
boxPlot(
  x,
  fact = NULL,
  horiz = FALSE,
  width = 2/3,
  lwd = 1,
  lcol = "black",
  medianLwd = 2,
  pch = 20,
  pchCex = 1.8,
  col = grDevices::rgb(0, 0, 0, 0.25),
  add = FALSE,
  key = NULL,
  axes = TRUE,
  xlab = "",
  ylab = "",
  xlim = NULL,
  ylim = NULL,
  na.rm = TRUE,
  ...
)
```

**Arguments**

<code>x</code>	A numerical vector.
<code>fact</code>	A character or factor vector defining the grouping for side-by-side box plots.
<code>horiz</code>	If TRUE, the box plot is oriented horizontally.
<code>width</code>	The width of the boxes in the plot. Value between 0 and 1.
<code>lwd</code>	Width of lines used in box and whiskers.
<code>lcol</code>	Color of the box, median, and whiskers.
<code>medianLwd</code>	Width of the line marking the median.
<code>pch</code>	Plotting character of outliers.
<code>pchCex</code>	Size of outlier character.
<code>col</code>	Color of outliers.
<code>add</code>	If FALSE, a new plot is created. Otherwise, the boxplots are added to the current plot for values of TRUE or a numerical vector specifying the locations of the boxes.
<code>key</code>	The order in which to display the side-by-side boxplots. If locations are specified in add, then the elements of add will correspond to the elements of key.
<code>axes</code>	Whether to plot the axes.
<code>xlab</code>	Label for the x axis.
<code>ylab</code>	Label for the y axis.
<code>xlim</code>	Limits for the x axis.
<code>ylim</code>	Limits for the y axis.
<code>na.rm</code>	Indicate whether NA values should be removed.
<code>...</code>	Additional arguments to plot.

**Author(s)**

David Diez

**See Also**[histPlot](#), [dotPlot](#), [densityPlot](#)**Examples**

```
# univariate
boxPlot(email$num_char, ylab = "Number of characters in emails")

# bivariate
boxPlot(email$num_char, email$spam,
  xlab = "Spam",
  ylab = "Number of characters in emails"
)

# faded outliers
```

```

boxPlot(email$num_char, email$spam,
  xlab = "Spam",
  ylab = "Number of characters in emails",
  col = fadeColor("black", 18)
)

# horizontal plots
boxPlot(email$num_char, email$spam,
  horiz = TRUE,
  xlab = "Spam",
  ylab = "Number of characters in emails",
  col = fadeColor("black", 18)
)

# bivariate relationships where categorical data have more than 2 levels
boxPlot(email$num_char, email$image,
  horiz = TRUE,
  xlab = "Number of attached images",
  ylab = "Number of characters in emails",
  col = fadeColor("black", 18)
)

# key can be used to restrict to only the desired groups
boxPlot(email$num_char, email$image,
  horiz = TRUE, key = c(0, 1, 2),
  xlab = "Number of attached images (limited to 0, 1, 2)",
  ylab = "Number of characters in emails",
  col = fadeColor("black", 18)
)

# combine boxPlot and dotPlot
boxPlot(tips$tip, tips$day,
  horiz = TRUE, key = c("Tuesday", "Friday")
)
dotPlot(tips$tip, tips$day,
  add = TRUE, at = 1:2 + 0.05,
  key = c("Tuesday", "Friday")
)

# adding a box
boxPlot(email$num_char[email$spam == 0], xlim = c(0, 3))
boxPlot(email$num_char[email$spam == 1], add = 2, axes = FALSE)
axis(1, at = 1:2, labels = c(0, 1))
boxPlot(email$num_char[email$spam == 0], ylim = c(0, 3), horiz = TRUE)
boxPlot(email$num_char[email$spam == 1], add = 2, horiz = TRUE, axes = FALSE)
axis(2, at = 1:2, labels = c(0, 1))

```

**Description**

This function is not yet very flexible.

**Usage**

```
Braces(x, y, face.radians = 0, long = 1, short = 0.2, ...)
```

**Arguments**

<code>x</code>	x-coordinate of the center of the braces.
<code>y</code>	y-coordinate of the center of the braces.
<code>face.radians</code>	Radians of where the braces should face. For example, the default with <code>face.radians = 0</code> has the braces facing right. Setting to $\pi / 2$ would result in the braces facing up.
<code>long</code>	The units for the long dimension of the braces.
<code>short</code>	The units for the short dimension of the braces. This must be less than or equal to half of the long dimension.
<code>...</code>	Arguments passed to <a href="#">lines</a> .

**Author(s)**

David Diez

**See Also**

[dlsegments](#)

**Examples**

```
plot(0:1, 0:1, type = "n")
Braces(0.5, 0.5, face.radians = 3 * pi / 2)
```

---

buildAxis

*Axis function substitute*

---

**Description**

The function `buildAxis` is built to provide more control of the number of labels on the axis. This function is still under development.

**Usage**

```
buildAxis(side, limits, n, nMin = 2, nMax = 10, extend = 2, eps = 10^-12, ...)
```

**Arguments**

side	The side of the plot where to add the axis.
limits	Either lower and upper limits on the axis or a dataset.
n	The preferred number of axis labels.
nMin	The minimum number of axis labels.
nMax	The maximum number of axis labels.
extend	How far the axis may extend beyond <code>range(limits)</code> .
eps	The smallest increment allowed.
...	Arguments passed to <code>axis</code>

**Details**

The primary reason behind building this function was to allow a plot to be created with similar features but with different datasets. For instance, if a set of code was written for one dataset and the function `axis` had been utilized with pre-specified values, the axis may not match the plot of a new set of data. The function `buildAxis` addresses this problem by allowing the number of axis labels to be specified and controlled.

The axis is built by assigning penalties to a variety of potential axis setups, ranking them based on these penalties and then selecting the axis with the best score.

**Value**

A vector of the axis plotted.

**Author(s)**

David Diez

**See Also**

[histPlot](#), [dotPlot](#), [boxPlot](#), [densityPlot](#)

**Examples**

```
# ==> 0 <==#
limits <- rnorm(100, 605490, 10)
hist(limits, axes = FALSE)
buildAxis(1, limits, 2, nMax = 4)

# ==> 1 <==#
x <- seq(0, 500, 10)
y <- 8 * x + rnorm(length(x), mean = 6000, sd = 200)
plot(x, y, axes = FALSE)
buildAxis(1, limits = x, n = 5)
buildAxis(2, limits = y, n = 3)

# ==> 2 <==#
x <- 9528412 + seq(0, 200, 10)
```

```

y <- 8 * x + rnorm(length(x), mean = 6000, sd = 200)
plot(x, y, axes = FALSE)
temp <- buildAxis(1, limits = x, n = 4)
buildAxis(2, y, 3)

# ==> 3 <==#
x <- seq(367, 1251, 10)
y <- 7.5 * x + rnorm(length(x), mean = 6000, sd = 800)
plot(x, y, axes = FALSE)
buildAxis(1, limits = x, n = 4, nMin = 3, nMax = 3)
buildAxis(2, limits = y, n = 4, nMin = 3, nMax = 5)

# ==> 4 <==#
x <- seq(367, 367.1, 0.001)
y <- 7.5 * x + rnorm(length(x), mean = 6000, sd = 0.01)
plot(x, y, axes = FALSE)
buildAxis(1, limits = x, n = 4, nMin = 5, nMax = 6)
buildAxis(2, limits = y, n = 2, nMin = 3, nMax = 4)

# ==> 5 <==#
x <- seq(-0.05, -0.003, 0.0001)
y <- 50 + 20 * x + rnorm(length(x), sd = 0.1)
plot(x, y, axes = FALSE)
buildAxis(1, limits = x, n = 4, nMin = 5, nMax = 6)
buildAxis(2, limits = y, n = 4, nMax = 5)
abline(lm(y ~ x))

# ==> 6 <==#
x <- seq(-0.0097, -0.008, 0.0001)
y <- 50 + 20 * x + rnorm(length(x), sd = 0.1)
plot(x, y, axes = FALSE)
buildAxis(1, limits = x, n = 4, nMin = 2, nMax = 5)
buildAxis(2, limits = y, n = 4, nMax = 5)
abline(lm(y ~ x))

# ==> 7 <==#
x <- seq(0.03, -0.003099, -0.00001)
y <- 50 + 20 * x + rnorm(length(x), sd = 0.1)
plot(x, y, axes = FALSE)
buildAxis(1, limits = x, n = 4, nMin = 2, nMax = 5)
buildAxis(2, limits = y, n = 4, nMax = 6)
abline(lm(y ~ x))

# ==> 8 - repeat <==#
m <- runif(1) / runif(1) +
  rgamma(1, runif(1) / runif(1), runif(1) / runif(1))
s <- rgamma(1, runif(1) / runif(1), runif(1) / runif(1))
x <- rnorm(50, m, s)
hist(x, axes = FALSE)
buildAxis(1, limits = x, n = 5, nMin = 4, nMax = 6, eps = 10^-12)
if (diff(range(x)) < 10^-12) {
  cat("too small\n")
}

```



---

burger	<i>Burger preferences</i>
--------	---------------------------

---

**Description**

Sample burger place preferences versus gender.

**Usage**

```
burger
```

**Format**

A data frame with 500 observations on the following 2 variables.

**best\_burger\_place** Burger place.

**gender** a factor with levels Female and Male

**Source**

SurveyUSA, Results of SurveyUSA News Poll #17718, data collected on December 2, 2010.

**Examples**

```
table(burger)
```

---

calc_streak	<i>Calculate hit streaks</i>
-------------	------------------------------

---

**Description**

Calculate hit streaks

**Usage**

```
calc_streak(x)
```

**Arguments**

x A character vector of hits ("H") and misses ("M").

**Value**

A data frame with one column, length, containing the length of each hit streak.

**Examples**

```
data(kobe_basket)
calc_streak(kobe_basket$shot)
```

---

cancer_in_dogs	<i>Cancer in dogs</i>
----------------	-----------------------

---

**Description**

A study in 1994 examined 491 dogs that had developed cancer and 945 dogs as a control group to determine whether there is an increased risk of cancer in dogs that are exposed to the herbicide 2,4-Dichlorophenoxyacetic acid (2,4-D).

**Usage**

```
cancer_in_dogs
```

**Format**

A data frame with 1436 observations on the following 2 variables.

**order** a factor with levels 2, 4-D and no 2, 4-D

**response** a factor with levels cancer and no cancer

**Source**

Hayes HM, Tarone RE, Cantor KP, Jessen CR, McCurnin DM, and Richardson RC. 1991. Case-Control Study of Canine Malignant Lymphoma: Positive Association With Dog Owner's Use of 2, 4- Dichlorophenoxyacetic Acid Herbicides. Journal of the National Cancer Institute 83(17):1226-1231.

**Examples**

```
table(cancer_in_dogs)
```

---

cards	<i>Deck of cards</i>
-------	----------------------

---

**Description**

All the cards in a standard deck.

**Usage**

```
cards
```

**Format**

A data frame with 52 observations on the following 4 variables.

**value** a factor with levels 10 2 3 4 5 6 7 8 9 A J K Q

**color** a factor with levels black red

**suit** a factor with levels Club Diamond Heart Spade

**face** a logical vector

**Examples**

```
table(cards$value)
table(cards$color)
table(cards$suit)
table(cards$face)
table(cards$suit, cards$face)
```

---

cars04

cars04

---

**Description**

A data frame with 428 rows and 19 columns. This is a record of characteristics on all of the new models of cars for sale in the US in the year 2004.

**Usage**

```
cars04
```

**Format**

A data frame with 428 observations on the following 19 variables.

**name** The name of the vehicle including manufacturer and model.

**sports\_car** Logical variable indicating if the vehicle is a sports car.

**suv** Logical variable indicating if the vehicle is an suv.

**wagon** Logical variable indicating if the vehicle is a wagon.

**minivan** Logical variable indicating if the vehicle is a minivan.

**pickup** Logical variable indicating if the vehicle is a pickup.

**all\_wheel** Logical variable indicating if the vehicle is all-wheel drive.

**rear\_wheel** Logical variable indicating if the vehicle is rear-wheel drive.

**msrp** Manufacturer suggested retail price of the vehicle.

**dealer\_cost** Amount of money the dealer paid for the vehicle.

**eng\_size** Displacement of the engine - the total volume of all the cylinders, measured in liters.

**ncyl** Number of cylinders in the engine.

- horsepwr** Amount of horsepower produced by the engine.
- city\_mpg** Gas mileage for city driving, measured in miles per gallon.
- hwy\_mpg** Gas mileage for highway driving, measured in miles per gallon.
- weight** Total weight of the vehicle, measured in pounds.
- wheel\_base** Distance between the center of the front wheels and the center of the rear wheels, measured in inches.
- length** Total length of the vehicle, measured in inches.
- width** Total width of the vehicle, measured in inches.

**Examples**

```
library(ggplot2)

# Highway gas mileage
ggplot(cars04, aes(x = hwy_mpg)) +
  geom_histogram(
    bins = 15, color = "white",
    fill = openintro::IMSCOL["green", "full"]
  ) +
  theme_minimal() +
  labs(
    title = "Highway gas milage for cars from 2004",
    x = "Gas Mileage (miles per gallon)",
    y = "Number of cars"
  )
```

---

<code>cars93</code>	<i>cars93</i>
---------------------	---------------

---

**Description**

A data frame with 54 rows and 6 columns. This data is a subset of the Cars93 dataset from the MASS package.

**Usage**

```
cars93
```

**Format**

A data frame with 54 observations on the following 6 variables.

- type** The vehicle type with levels large, midsize, and small.
- price** Vehicle price (USD).
- mpg\_city** Vehicle mileage in city (miles per gallon).
- drive\_train** Vehicle drive train with levels 4WD, front, and rear.
- passengers** The vehicle passenger capacity.
- weight** Vehicle weight (lbs).

## Details

These cars represent a random sample for 1993 models that were in both *Consumer Reports* and *PACE Buying Guide*. Only vehicles of type small, midsize, and large were include.

Further description can be found in Lock (1993). Use the URL <http://jse.amstat.org/v1n1/datasets.lock.html>.

## Source

Lock, R. H. (1993) 1993 New Car Data. *Journal of Statistics Education* 1(1).

## Examples

```
library(ggplot2)

# Vehicle price by type
ggplot(cars93, aes(x = price)) +
  geom_histogram(binwidth = 5) +
  facet_wrap(~type)

# Vehicle price vs. weight
ggplot(cars93, aes(x = weight, y = price)) +
  geom_point()

# Milleage vs. weight
ggplot(cars93, aes(x = weight, y = mpg_city)) +
  geom_point() +
  geom_smooth()
```

---

cchousing

---

*Community college housing (simulated data)*


---

## Description

These are simulated data and intended to represent housing prices of students at a community college.

## Usage

```
cchousing
```

## Format

A data frame with 75 observations on the following variable.

**price** Monthly housing price, simulated.

## Examples

```
hist(cchousing$price)
```

CCP

*Plot a Cartesian Coordinate Plane***Description**

Create a Cartesian Coordinate Plane.

**Usage**

```
CCP(
  xlim = c(-4, 4),
  ylim = c(-4, 4),
  mar = rep(0, 4),
  length = 0.1,
  tcl = 0.007,
  xylab = FALSE,
  ticks = 1,
  ticklabs = 1,
  xpos = 1,
  ypos = 2,
  cex.coord = 1,
  cex.xylab = 1.5,
  add = FALSE
)
```

**Arguments**

<code>xlim</code>	The x-limits for the plane (vector of length 2).
<code>ylim</code>	The y-limits for the plane (vector of length 2).
<code>mar</code>	Plotting margins.
<code>length</code>	The <code>length</code> argument is passed to the <a href="#">arrows</a> function and is used to control the size of the arrow.
<code>tcl</code>	Tick size.
<code>xylab</code>	Whether x and y should be shown next to the labels.
<code>ticks</code>	How frequently tick marks should be shown on the axes. If a vector of length 2, the first argument will correspond to the x-axis and the second to the y-axis.
<code>ticklabs</code>	How frequently tick labels should be shown on the axes. If a vector of length 2, the first argument will correspond to the x-axis and the second to the y-axis.
<code>xpos</code>	The position of the labels on the x-axis. See the <code>pos</code> argument in the <a href="#">text</a> function for additional details.
<code>ypos</code>	The position of the labels on the y-axis. See the <code>pos</code> argument in the <a href="#">text</a> function for additional details.
<code>cex.coord</code>	Inflation factor for font size of the coordinates, where any value larger than zero is acceptable and 1 corresponds to the default.

**cex.xylab** Inflation factor for font size of the x and y labels, where any value larger than zero is acceptable and 1 corresponds to the default.

**add** Indicate whether a new plot should be created (FALSE, the default) or if the Cartesian Coordinate Plane should be added to the existing plot.

**Author(s)**

David Diez

**See Also**

[lsegments](#), [dlsegments](#), [ArrowLines](#)

**Examples**

```
CCP()

CCP(xylab = TRUE, ylim = c(-3.5, 2), xpos = 3, cex.coord = 1)

CCP(xlim = c(-8, 8), ylim = c(-10, 6), ticklabs = c(2, 2), cex.xylab = 0.8)
```

---

census

*Random sample of 2000 U.S. Census Data*


---

**Description**

A random sample of 500 observations from the 2000 U.S. Census Data.

**Usage**

```
census
```

**Format**

A data frame with 500 observations on the following 8 variables.

**census\_year** Census Year.

**state\_fips\_code** Name of state.

**total\_family\_income** Total family income (in U.S. dollars).

**age** Age.

**sex** Sex with levels Female and Male.

**race\_general** Race with levels American Indian or Alaska Native, Black, Chinese, Japanese, Other Asian or Pacific Islander, Two major races, White and Other.

**marital\_status** Marital status with levels Divorced, Married/spouse absent, Married/spouse present, Never married/single, Separated and Widowed.

**total\_personal\_income** Total personal income (in U.S. dollars).

**Source**

<https://data.census.gov/cedsci>

**Examples**

```
library(dplyr)
library(ggplot2)

census |>
  filter(total_family_income > 0) |>
  ggplot(aes(x = total_family_income)) +
  geom_histogram(binwidth = 25000)
```

---

cherry

---

*Summary information for 31 cherry trees*


---

**Description**

Researchers wanting to understand the relationship between these variables for black cherry trees collected data from 31 trees in the Allegheny National Forest, Pennsylvania.

**Usage**

```
cherry
```

**Format**

A data frame with 31 observations on the following 3 variables.

**diam** diameter in inches (at 54 inches above ground)

**height** height is measured in feet

**volume** volume in cubic feet

**Source**

D.J. Hand. A handbook of small data sets. Chapman & Hall/CRC, 1994.

**Examples**

```
library(ggplot2)
library(broom)

ggplot(cherry, aes(x = diam, y = volume)) +
  geom_point() +
  geom_smooth(method = "lm")

mod <- lm(volume ~ diam + height, cherry)
tidy(mod)
```



---

`children_gender_stereo`*Gender Stereotypes in 5-7 year old Children*

---

## Description

Stereotypes are common, but at what age do they start? This study investigates stereotypes in young children aged 5-7 years old. There are four studies reported in the paper, and all four datasets are provided here.

## Usage

`children_gender_stereo`

## Format

This data object is more unusual than most. It is a list of 4 data frames. The four data frames correspond to the data used in Studies 1-4 of the referenced paper, and these data frames each have variables (columns) that are among the following:

**subject** Subject ID. Note that Subject 1 in the first data frame (dataset) does **not** correspond to Subject 1 in the second data frame.

**gender** Gender of the subject.

**age** Age of the subject, in years.

**trait** The trait that the children were making a judgement about, which was either nice or smart.

**target** The age group of the people the children were making judgements about (as being either nice or smart): children or adults.

**stereotype** The proportion of trials where the child picked a gender target that matched the trait that was the same as the gender of the child. For example, suppose we had 18 pictures, where each picture showed 2 men and 2 women (and a different set of people in each photo). Then if we asked a boy to pick the person in each picture who they believed to be really smart, this stereotype variable would report the fraction of pictures where the boy picked a man. When a girl reviews the photos, then this stereotype variable reports the fraction of photos where she picked a woman. That is, this variable differs in meaning depending on the gender of the child. (This variable design is a little confusing, but it is useful when analyzing the data.)

**high\_achieve\_caution** The proportion of trials where the child said that children of their own gender were high-achieving in school.

**interest** Average score that measured the interest of the child in the game.

**difference** A difference score between the interest of the child in the “smart” game and their interest in the “try-hard” game.

## Details

The structure of the data object is a little unusual, so we recommend reviewing the Examples section before starting your analysis.

Thank you to Nicholas Horton for pointing us to this study and the data!

Most of the results in the paper can be reproduced using the data provided here.

% TODO(David) - Add short descriptions of each study.

## Source

Bian L, Leslie SJ, Cimpian A. 2017. "Gender stereotypes about intellectual ability emerge early and influence children's interests". *Science* 355:6323 (389-391). <https://www.science.org/doi/10.1126/science.aah6524>.

The original data may be found [here](#).

## Examples

```
# This dataset is a little funny to work with.
# If wanting to review the data for a study, we
# recommend first assigning the corresponding
# data frame to a new variable. For instance,
# below we assign the second study's data to an
# object called `d` (d is for data!).
d <- children_gender_stereo[[2]]
```

---

china

*Child care hours*

---

## Description

The China Health and Nutrition Survey aims to examine the effects of the health, nutrition, and family planning policies and programs implemented by national and local governments.

## Usage

china

## Format

A data frame with 9788 observations on the following 3 variables.

**gender** a numeric vector

**edu** a numeric vector

**child\_care** a numeric vector

## Source

UNC Carolina Population Center, China Health and Nutrition Survey, 2006.

**Examples**

```
summary(china)
```

---

**ChiSquareTail***Plot upper tail in chi-square distribution*

---

**Description**

Plot a chi-square distribution and shade the upper tail.

**Usage**

```
ChiSquareTail(  
  U,  
  df,  
  xlim = c(0, 10),  
  col = fadeColor("black", "22"),  
  axes = TRUE,  
  ...  
)
```

**Arguments**

U	Cut off for the upper tail.
df	Degrees of freedom.
xlim	Limits for the plot.
col	Color of the shading.
axes	Whether to plot an x-axis.
...	Currently ignored.

**Value**

Nothing is returned from the function.

**Author(s)**

David Diez

**See Also**

[normTail](#)

**Examples**

```
data(COL)
ChiSquareTail(11.7,
  7,
  c(0, 25),
  col = COL[1]
)
```

---

cia\_factbook

*CIA Factbook Details on Countries*


---

**Description**

Country-level statistics from the US Central Intelligence Agency (CIA).

**Usage**

```
cia_factbook
```

**Format**

A data frame with 259 observations on the following 11 variables.

**country** Country name.

**area** Land area, in square kilometers. (1 square kilometer is 0.386 square miles)

**birth\_rate** Birth rate, in births per 1,000 people.

**death\_rate** Death rate, in deaths per 1,000 people.

**infant\_mortality\_rate** Infant mortality, in deaths per 1,000 live births.

**internet\_users** Total number of internet users.

**life\_exp\_at\_birth** Live expectancy at birth, in years.

**maternal\_mortality\_rate** Number of female deaths per 100,000 live births where the death is related to pregnancy or birth.

**net\_migration\_rate** Net migration rate.

**population** Total population.

**population\_growth\_rate** Population growth rate.

**Source**

CIA Factbook, Country Comparisons, 2014. <https://www.cia.gov/the-world-factbook/references/guide-to-country-comparisons/>

**Examples**

```
library(dplyr)
library(ggplot2)

cia_factbook_iup <- cia_factbook |>
  mutate(internet_users_percent = 100 * internet_users / population)

ggplot(cia_factbook_iup, aes(x = internet_users_percent, y = life_exp_at_birth)) +
  geom_point() +
  labs(x = "Percentage of internet users", y = "Life expectancy at birth")
```

---

classdata

---

*Simulated class data*


---

**Description**

This data is simulated and is meant to represent students scores from three different lectures who were all given the same exam.

**Usage**

```
classdata
```

**Format**

A data frame with 164 observations on the following 2 variables.

**m1** Represents a first midterm score.

**lecture** Three classes: a, b, and c.

**References**

OpenIntro Statistics, Chapter 8.

**Examples**

```
anova(lm(m1 ~ lecture, classdata))
```

---

cle\_sac

---

Cleveland and Sacramento

---

### Description

Data on a sample of 500 people from the Cleveland, OH and Sacramento, CA metro areas.

### Usage

```
cle_sac
```

### Format

A data frame with 500 observations representing people on the following 8 variables.

**year** Year the data was collected.

**state** State where person resides.

**city** City.

**age** Age.

**sex** Sex.

**race** Race.

**marital\_status** Marital status.

**personal\_income** Personal income.

### Examples

```
library(ggplot2)

ggplot(cle_sac, aes(x = personal_income)) +
  geom_histogram(binwidth = 20000) +
  facet_wrap(~city)
```

---

climate70

---

Temperature Summary Data, Geography Limited

---

### Description

A random set of monitoring locations were taken from NOAA data that had both years of interest (1948 and 2018) as well as data for both summary metrics of interest (dx70 and dx90, which are described below).

### Usage

```
climate70
```

**Format**

A data frame with 197 observations on the following 7 variables.

**station** Station ID.

**latitude** Latitude of the station.

**longitude** Longitude of the station.

**dx70\_1948** Number of days above 70 degrees in 1948.

**dx70\_2018** Number of days above 70 degrees in 2018.

**dx90\_1948** Number of days above 90 degrees in 1948.

**dx90\_2018** Number of days above 90 degrees in 2018.

**Details**

Please keep in mind that these are two annual snapshots, and a complete analysis would consider much more than two years of data and much additional information for those years.

**Source**

<https://www.ncdc.noaa.gov/cdo-web>, retrieved 2019-04-24.

**Examples**

```
# Data sampled are from the US, Europe, and Australia.
# This geographic limitation may be due to the particular
# years considered, since locations without both 1948 and
# 2018 were discarded for this (simple) dataset.
plot(climate70$longitude, climate70$latitude)

plot(climate70$dx70_1948, climate70$dx70_2018)
abline(0, 1, lty = 2)
plot(climate70$dx90_1948, climate70$dx90_2018)
abline(0, 1, lty = 2)
hist(climate70$dx70_2018 - climate70$dx70_1948)
hist(climate70$dx90_2018 - climate70$dx90_1948)

t.test(climate70$dx70_2018 - climate70$dx70_1948)
t.test(climate70$dx90_2018 - climate70$dx90_1948)
```

---

climber\_drugs

*Climber Drugs Data.*


---

**Description**

Anonymous data was collected from urine samples at huts along the climb of Mont Blanc. Several types of drugs were tested, and proportions were reported.

**Usage**

```
climber_drugs
```

**Format**

A data frame with 211 rows and 6 variables.

**positive\_sample** Identification number of a specific urine sample.

**hut** Location where the sample was taken.

**substance** Substance detected to be present in the urine sample.

**concentration** Amount of substance found measured in ng/ml.

**screening\_analysis** Indicates that the concentration was determined by screening analysis.

**concomitant** Indicates that this substance was always detected concomitantly with the previous one, within the same urine sample.

**Source**

[PLOS One - Drug Use on Mont Blanc: A Study Using Automated Urine Collection](#)

**Examples**

```
library(dplyr)

# Calculate the average concentration of each substance and number of occurrences.
climber_drugs |>
  group_by(substance) |>
  summarize(count = n(), mean_con = mean(concentration))

# Proportion samples in which each substance was detected.
climber_drugs |>
  group_by(substance) |>
  summarize(prop = n() / 154)
```

---

 coast\_starlight

---

*Coast Starlight Amtrak train*


---

**Description**

Travel times and distances.

**Usage**

```
coast_starlight
```



**Format**

A data frame with 16 observations on the following 3 variables.

**station** Station.

**dist** Distance.

**travel\_time** Travel time.

**Examples**

```
library(ggplot2)

ggplot(coast_starlight, aes(x = dist, y = travel_time)) +
  geom_point()
```

---

 COL

*OpenIntro Statistics colors*


---

**Description**

These are the core colors used for the OpenIntro Statistics textbook. The blue, green, yellow, and red colors are also gray-scaled, meaning no changes are required when printing black and white copies.

**Usage**

```
COL
```

**Format**

A 7-by-13 matrix of 7 colors with thirteen fading scales: blue, green, yellow, red, black, gray, and light gray.

**Source**

Colors selected by OpenIntro's in-house graphic designer, [Meenal Patel](#).

**Examples**

```
plot(1:7, 7:1,
     col = COL, pch = 19, cex = 6, xlab = "", ylab = "",
     xlim = c(0.5, 7.5), ylim = c(-2.5, 8), axes = FALSE
)
text(1:7, 7:1 + 0.7, paste("COL[", 1:7, "]", sep = ""), cex = 0.9)
points(1:7, 7:1 - 0.7, col = COL[, 2], pch = 19, cex = 6)
points(1:7, 7:1 - 1.4, col = COL[, 3], pch = 19, cex = 6)
points(1:7, 7:1 - 2.1, col = COL[, 4], pch = 19, cex = 6)
```

comics

*comics***Description**

A data frame containing information about comic book characters from Marvel Comics and DC Comics.

**Usage**

comics

**Format**

A data frame with 21821 observations on the following 11 variables.

**name** Name of the character. May include: Real name, hero or villain name, alias(es) and/or which universe they live in (i.e. Earth-616 in Marvel's multiverse).

**id** Status of the characters identity with levels Secret, Public, No Dual and Unknown.

**align** Character's alignment with levels Good, Bad, Neutral and Reformed Criminals.

**eye** Character's eye color.

**hair** Character's hair color.

**gender** Character's gender.

**gsm** Character's classification as a gender or sexual minority.

**alive** Is the character dead or alive?

**appearances** Number of comic books the character appears in.

**first\_appear** Date of publication for the comic book the character first appeared in.

**publisher** Publisher of the comic with levels Marvel and DC.

**Examples**

```
library(ggplot2)
library(dplyr)

# Good v Bad

plot_data <- comics |>
  filter(align == "Good" | align == "Bad")

ggplot(plot_data, aes(x = align, fill = align)) +
  geom_bar() +
  facet_wrap(~publisher) +
  scale_fill_manual(values = c(IMSCOL["red", "full"], IMSCOL["blue", "full"])) +
  theme_minimal() +
  labs(
    title = "Is there a balance of power",
```

```

    x = "",
    y = "Number of characters",
    fill = ""
  )

```

---

contTable

---

*Generate Contingency Tables for LaTeX*


---

### Description

Input a data frame or a table, and the LaTeX output will be returned. Options exist for row and column proportions as well as for showing work.

### Usage

```

contTable(
  x,
  prop = c("none", "row", "col"),
  show = FALSE,
  digits = 3,
  caption = NULL,
  label = NULL
)

```

### Arguments

x	A data frame (with two columns) or a table.
prop	Indicate whether row ("r", "R", "row") or column ("c", "C", "col") proportions should be used. The default is to simply print the contingency table.
show	If row or column proportions are specified, indicate whether work should be shown.
digits	The number of digits after the decimal that should be shown for row or column proportions.
caption	A string that contains the table caption. The default value is NULL. If x is a data frame and caption=NULL, then contTable creates a sensible caption from the data frame's column names. If x is a table and caption=NULL, then the caption is an empty string.
label	The latex table label. The default value is NULL. If x is a data frame and label=NULL, then contTable creates a sensible label from the data frame's column names. If x is a table and label=NULL, then the label is an empty string.

### Details

The contTable function makes substantial use of the [cat](#) function.

**Author(s)**

David Diez

**See Also**

[email](#), [cars93](#), [possum](#), [mariokart](#)

**Examples**

```
data(email)
table(email[, c("spam", "sent_email")])
contTable(email[, c("spam", "sent_email")])
```

---

corr\_match

*Sample datasets for correlation problems*

---

**Description**

Simulated data.

**Usage**

```
corr_match
```

**Format**

A data frame with 121 observations on the following 9 variables.

**x** a numeric vector

**y1** a numeric vector

**y2** a numeric vector

**y3** a numeric vector

**y4** a numeric vector

**y5** a numeric vector

**y6** a numeric vector

**y7** a numeric vector

**y8** a numeric vector

**Source**

Simulated dataset.

### Examples

```
library(ggplot2)

ggplot(corr_match, aes(x = x, y = y1)) +
  geom_point()

cor(corr_match$x, corr_match$y1)
```

---

country_iso	<i>Country ISO information</i>
-------------	--------------------------------

---

### Description

Country International Organization for Standardization (ISO) information.

### Usage

```
country_iso
```

### Format

A data frame with 249 observations on the following 4 variables.

**country\_code** Two-letter ISO country code.

**country\_name** Country name.

**year** Year the two-letter ISO country code was assigned.

**top\_level\_domain** op-level domain name.

### Source

Wikipedia, retrieved 2018-11-18. [https://en.wikipedia.org/wiki/ISO\\_3166-1\\_alpha-2](https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2)

### Examples

```
country_iso
```

---

cpr

---

*CPR dataset*


---

### Description

These patients were randomly divided into a treatment group where they received a blood thinner or the control group where they did not receive a blood thinner. The outcome variable of interest was whether the patients survived for at least 24 hours.

### Usage

```
cpr
```

### Format

A data frame with 90 observations on the following 2 variables.

**group** a factor with levels control and treatment

**outcome** a factor with levels died and survived

### Source

Efficacy and safety of thrombolytic therapy after initially unsuccessful cardiopulmonary resuscitation: a prospective clinical trial, by Bottiger et al., The Lancet, 2001.

### Examples

```
table(cpr)
```

---

cpu

---

*CPU's Released between 2010 and 2020.*


---

### Description

Data on computer processors released between 2010 and 2020.

### Usage

```
cpu
```

## Format

A data frame with 875 rows and 12 variables.

**company** Manufacturer of the CPU.

**name** Model name of the processor.

**codename** Name given by manufacturer to all chips with this architecture.

**cores** Number of compute cores per processor.

**threads** The number of *threads* represents the number of simultaneous calculations that can be ongoing in the processor.

**base\_clock** Base speed for the CPU in GHz.

**boost\_clock** Single-core max speed for the CPU in GHz.

**socket** Specifies the type of connection to the motherboard.

**process** Size of the process node used in production in nm.

**l3\_cache** Size of the level 3 cache on the processor in MB.

**tdp** Total draw power of the processor.

**released** Date which the processor was released to the public.

## Source

[TechPowerUp CPU Database](#).

## Examples

```
library(ggplot2)

# CPU base speed
ggplot(cpu, aes(x = company, y = base_clock)) +
  geom_boxplot() +
  labs(
    x = "Company",
    y = "Base Clock (GHz)",
    title = "CPU base speed"
  )

# Process node size vs. boost speed
ggplot(cpu, aes(x = process, y = boost_clock)) +
  geom_point() +
  labs(
    x = "Process node size (nm)",
    y = "Boost Clock (GHz)",
    title = "Process node size vs. boost speed"
  )
```

---

credits	<i>College credits.</i>
---------	-------------------------

---

**Description**

A simulated dataset of number of credits taken by college students each semester.

**Usage**

```
credits
```

**Format**

A data frame with 100 observations on the following variable.

**credits** Number of credits.

**Source**

Simulated data.

**Examples**

```
library(ggplot2)

ggplot(credits, aes(x = credits)) +
  geom_histogram(binwidth = 1)
```

---

CT2DF	<i>Contingency Table to Data Frame</i>
-------	--

---

**Description**

Take a 2D contingency table and create a data frame representing the individual cases.

**Usage**

```
CT2DF(x, rn = row.names(x), cn = colnames(x), dfn = c("row.var", "col.var"))
```

**Arguments**

x	Contingency table as a matrix.
rn	Character vector of the row names.
cn	Character vector of the column names.
dfn	Character vector with 2 values for the variable representing the rows and columns.



**Value**

A data frame with two columns.

**Author(s)**

David Diez

**See Also**

[MosaicPlot](#)

**Examples**

```
a <- matrix(
  c(459, 727, 854, 385, 99, 4198, 6245, 4821, 1634, 578),
  2,
  byrow = TRUE
)
b <-
  CT2DF(
    a,
    c("No", "Yes"),
    c("Excellent", "Very good", "Good", "Fair", "Poor"),
    c("coverage", "health_status")
  )
table(b)
```

---

daycare_fines	<i>Daycare fines</i>
---------------	----------------------

---

**Description**

Researchers tested the deterrence hypothesis which predicts that the introduction of a penalty will reduce the occurrence of the behavior subject to the fine, with the condition that the fine leaves everything else unchanged by instituting a fine for late pickup at daycare centers. For this study, they worked with 10 volunteer daycare centers that did not originally impose a fine to parents for picking up their kids late. They randomly selected 6 of these daycare centers and instituted a monetary fine (of a considerable amount) for picking up children late and then removed it. In the remaining 4 daycare centers no fine was introduced. The study period was divided into four: before the fine (weeks 1–4), the first 4 weeks with the fine (weeks 5-8), the entire period with the fine (weeks 5–16), and the after fine period (weeks 17-20). Throughout the study, the number of kids who were picked up late was recorded each week for each daycare. The study found that the number of late-coming parents increased significantly when the fine was introduced, and no reduction occurred after the fine was removed.

**Usage**

daycare\_fines

## Format

A data frame with 200 observations on the following 7 variables.

**center** Daycare center id.

**group** Study group: test (fine instituted) or control (no fine).

**children** Number of children at daycare center.

**week** Week of study.

**late\_pickups** Number of late pickups for a given week and daycare center.

**study\_period\_4** Period of study, divided into 4 periods: before fine, first 4 weeks with fine, last 8 weeks with fine, after fine

**study\_period\_3** Period of study, divided into 4 periods: before fine, with fine, after fine

## Source

Gneezy, Uri, and Aldo Rustichini. "A fine is a price." *The Journal of Legal Studies* 29, no. 1 (2000): 1-17.

## Examples

```
library(dplyr)
library(tidyr)
library(ggplot2)

# The following tables roughly match results presented in Table 2 of the source article
# The results are only off by rounding for some of the weeks
daycare_fines |>
  group_by(center, study_period_4) |>
  summarise(avg_late_pickups = mean(late_pickups), .groups = "drop") |>
  pivot_wider(names_from = study_period_4, values_from = avg_late_pickups)

daycare_fines |>
  group_by(center, study_period_3) |>
  summarise(avg_late_pickups = mean(late_pickups), .groups = "drop") |>
  pivot_wider(names_from = study_period_3, values_from = avg_late_pickups)

# The following plot matches Figure 1 of the source article
daycare_fines |>
  group_by(week, group) |>
  summarise(avg_late_pickups = mean(late_pickups), .groups = "drop") |>
  ggplot(aes(x = week, y = avg_late_pickups, group = group, color = group)) +
  geom_point() +
  geom_line()
```

densityPlot

*Density plot***Description**

Compute kernel density plots, written in the same structure as [boxPlot](#). Histograms can be automatically added for teaching purposes.

**Usage**

```
densityPlot(
  x,
  fact = NULL,
  bw = "nrd0",
  histo = c("none", "faded", "hollow"),
  breaks = "Sturges",
  fading = "0E",
  fadingBorder = "25",
  lty = NULL,
  lwd = 1,
  col = c("black", "red", "blue"),
  key = NULL,
  add = FALSE,
  adjust = 1,
  kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine"),
  weights = NULL,
  n = 512,
  from,
  to,
  na.rm = FALSE,
  xlim = NULL,
  ylim = NULL,
  main = "",
  ...
)
```

**Arguments**

x	A numerical vector.
fact	A character or factor vector defining the grouping for data in x.
bw	Bandwidth. See <a href="#">density</a> .
histo	Whether to plot a faded histogram ('faded') or hollow histogram ('hollow') in the background. By default, no histogram will be plotted.
breaks	The breaks argument for <a href="#">histPlot</a> if histo is 'faded' or 'hollow'.

fading	Character value of hexadecimal, e.g. '22' or '5D', describing the amount of fading inside the rectangles of the histogram if histo='faded'.
fadingBorder	Character value of hexadecimal, e.g. '22' or '5D', describing the amount of fading of the rectangle borders of the histogram if histo is 'faded' or 'hollow'.
lty	Numerical vector describing the line type for the density curve(s). Each element corresponds to a different level of the argumentfact.
lwd	Numerical vector describing the line width for the density curve(s). Each element corresponds to a different level of the argumentfact.
col	Numerical vector describing the line color for the density curve(s). Each element corresponds to a different level of the argumentfact.
key	An argument to specify ordering of the factor levels.
add	If TRUE, the density curve is added to the plot.
adjust	Argument passed to density to adjust the bandwidth.
kernel	Argument passed to density to select the kernel used.
weights	Argument passed to density to weight observations.
n	Argument passed to density to specify the detail in the density estimate.
from	Argument passed to density specifying the lowest value to include in the density estimate.
to	Argument passed to density specifying the largest value to include in the density estimate.
na.rm	Argument passed to density specifying handling of NA values.
xlim	x-axis limits.
ylim	y-axis limits.
main	Title for the plot.
...	If add=FALSE, then additional arguments to plot.

**Author(s)**

David Diez

**See Also**[histPlot](#), [dotPlot](#), [boxPlot](#)**Examples**

```
# hollow histograms
histPlot(tips$tip[tips$day == "Tuesday"],
  hollow = TRUE, xlim = c(0, 30),
  lty = 1, main = "Tips by day"
)
histPlot(tips$tip[tips$day == "Friday"],
  hollow = TRUE, border = "red",
  add = TRUE, main = "Tips by day"
```

```

)
legend("topright",
      col = c("black", "red"),
      lty = 1:2, legend = c("Tuesday", "Friday"))
)

# density plots
densityPlot(tips$tip, tips$day,
            col = c("black", "red"), main = "Tips by day"
)
legend("topright",
      col = c("black", "red"),
      lty = 1:2, legend = c("Tuesday", "Friday"))
)

densityPlot(tips$tip,
            histo = "faded",
            breaks = 15, main = "Tips by day"
)

densityPlot(tips$tip,
            histo = "hollow",
            breaks = 30, fadingBorder = "66",
            lty = 1, main = "Tips by day"
)

```

---

diabetes2

---

*Type 2 Diabetes Clinical Trial for Patients 10-17 Years Old*


---

## Description

Three treatments were compared to test their relative efficacy (effectiveness) in treating Type 2 Diabetes in patients aged 10-17 who were being treated with metformin. The primary outcome was lack of glycemic control (or not); lacking glycemic control means the patient still needed insulin, which is not the preferred outcome for a patient.

## Usage

```
diabetes2
```

## Format

A data frame with 699 observations on the following 2 variables.

**treatment** The treatment the patient received.

**outcome** Whether there patient still needs insulin (failure) or met a basic positive outcome bar (success).

## Details

Each of the 699 patients in the experiment were randomized to one of the following treatments: (1) continued treatment with metformin (coded as met), (2) formin combined with rosiglitazone (coded as rosi), or or (3) a lifestyle-intervention program (coded as lifestyle).

## Source

Zeitler P, et al. 2012. A Clinical Trial to Maintain Glycemic Control in Youth with Type 2 Diabetes. N Engl J Med.

## Examples

```
lapply(diabetes2, table)
(cont.table <- table(diabetes2))
(m <- chisq.test(cont.table))
m$expected
```

---

dlsegments

---

*Create a Double Line Segment Plot*


---

## Description

Creae a plot showing two line segments. The union or intersection of those line segments can also be generated by utilizing the type argument.

## Usage

```
dlsegments(
  x1 = c(3, 7),
  x2 = c(5, 9),
  l = c("o", "o"),
  r = c("c", "c"),
  type = c("n", "u", "i"),
  COL = 2,
  lwd = 2.224,
  ylim = c(-0.35, 2),
  mar = rep(0, 4),
  hideOrig = FALSE
)
```

## Arguments

x1	The endpoints of the first interval. Values larger (smaller) than 999 (-999) will be interpreted as (negative) infinity.
x2	The endpoints of the second interval. Values larger (smaller) than 999 (-999) will be interpreted as (negative) infinity.

<code>l</code>	A vector of length 2, where the values correspond to the left end point of each interval. A value of "o" indicates the interval is open at the left and "c" indicates the interval is closed at this end.
<code>r</code>	A vector of length 2, where the values correspond to the right end point of each interval. A value of "o" indicates the interval is open at the right and "c" indicates the interval is closed at this end.
<code>type</code>	By default, no intersection or union of the two lines will be shown (value of "n"). To show the union of the line segments, specify "u". To indicate that the intersection be shown, specify "i".
<code>COL</code>	If the union or intersection is to be shown (see the <code>type</code> argument), then this parameter controls the color that will be shown.
<code>lwd</code>	If the union or intersection is to be shown (see the <code>type</code> argument), then this parameter controls the width of any corresponding lines or open points in the union or intersection.
<code>ylim</code>	A vector of length 2 specifying the vertical plotting limits, which may be useful for fine-tuning plots. The default is <code>c(-0.35, 2)</code> .
<code>mar</code>	A vector of length 4 that represent the plotting margins.
<code>hideOrig</code>	An optional argument that to specify that the two line segments should be shown ( <code>hideOrig</code> takes value <code>FALSE</code> , the default) or that they should be hidden ( <code>hideOrig</code> takes value <code>TRUE</code> ).

**Author(s)**

David Diez

**See Also**[lsegments](#), [CCP](#), [ArrowLines](#)**Examples**

```
dlsegments(c(-3, 3), c(1, 1000),
  r = c("o", "o"), l = c("c", "o"), COL = COL[4]
)

dlsegments(c(-3, 3), c(1, 1000),
  r = c("o", "o"), l = c("c", "o"), type = "un", COL = COL[4]
)

dlsegments(c(-3, 3), c(1, 1000),
  r = c("o", "o"), l = c("c", "o"), type = "in", COL = COL[4]
)
```

dotPlot

*Dot plot***Description**

Plot observations as dots.

**Usage**

```
dotPlot(
  x,
  fact = NULL,
  vertical = FALSE,
  at = 1,
  key = NULL,
  pch = 20,
  col = fadeColor("black", "66"),
  cex = 1.5,
  add = FALSE,
  axes = TRUE,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

**Arguments**

<code>x</code>	A numerical vector.
<code>fact</code>	A character or factor vector defining the grouping for data in <code>x</code> .
<code>vertical</code>	If TRUE, the plot will be oriented vertically.
<code>at</code>	The vertical coordinate of the points, or the horizontal coordinate if <code>vertical=TRUE</code> . If <code>fact</code> is provided, then locations can be specified for each group.
<code>key</code>	The factor levels corresponding to <code>at</code> , <code>pch</code> , <code>col</code> , and <code>cex</code> .
<code>pch</code>	Plotting character. If <code>fact</code> is given, then different plotting characters can be specified for each factor level. If <code>key</code> is specified, the elements of <code>pch</code> will correspond to the elements of <code>key</code> .
<code>col</code>	Plotting character color. If <code>fact</code> is given, then different colors can be specified for each factor level. If <code>key</code> is specified, the elements of <code>col</code> will correspond to the elements of <code>key</code> .
<code>cex</code>	Plotting character size. If <code>fact</code> is given, then different character sizes can be specified for each factor level. If <code>key</code> is specified, the elements of <code>cex</code> will correspond to the elements of <code>key</code> .
<code>add</code>	If TRUE, then the points are added to the plot.
<code>axes</code>	If FALSE, no axes are plotted.



xlim	Limits for the x axis.
ylim	Limits for the y axis.
...	Additional arguments to be passed to plot if add=FALSE or points if add=TRUE.

**Author(s)**

David Diez

**See Also**

[histPlot](#), [densityPlot](#), [boxPlot](#)

**Examples**

```
library(dplyr)

# Price by type
dotPlot(cars93$price,
  cars93$type,
  key = c("large", "midsize", "small"),
  cex = 1:3
)

# Hours worked by educational attainment or degree
gss2010_nona <- gss2010 |>
  filter(!is.na(hrs1) & !is.na(degree))

dotPlot(gss2010_nona$hrs1,
  gss2010_nona$degree,
  col = fadeColor("black", "11")
)

# levels reordered
dotPlot(gss2010_nona$hrs1,
  gss2010_nona$degree,
  col = fadeColor("black", "11"),
  key = c("LT HIGH SCHOOL", "HIGH SCHOOL", "BACHELOR", "JUNIOR COLLEGE", "GRADUATE")
)

# with boxPlot() overlaid
dotPlot(mariokart$total_pr,
  mariokart$cond,
  ylim = c(0.5, 2.5), xlim = c(25, 80), cex = 1
)
boxPlot(mariokart$total_pr,
  mariokart$cond,
  add = 1:2 + 0.1,
  key = c("new", "used"), horiz = TRUE, axes = FALSE
)
```

---

`dotPlotStack`*Add a Stacked Dot Plot to an Existing Plot*

---

### Description

Add a stacked dot plot to an existing plot. The locations for the points in the dot plot are returned from the function in a list.

### Usage

```
dotPlotStack(x, radius = 1, seed = 1, addDots = TRUE, ...)
```

### Arguments

<code>x</code>	A vector of numerical observations for the dot plot.
<code>radius</code>	The approximate distance that should separate each point.
<code>seed</code>	A random seed (integer). Different values will produce different variations.
<code>addDots</code>	Indicate whether the points should be added to the plot.
<code>...</code>	Additional arguments are passed to <a href="#">points</a> .

### Value

Returns a list with a height that can be used as the upper bound of `ylim` for a plot, then also the x- and y-coordinates of the points in the stacked dot plot.

### Author(s)

David Diez

### See Also

[dotPlot](#), [histPlot](#)

### Examples

```
#
```

---

dream	<i>Survey on views of the DREAM Act</i>
-------	---

---

**Description**

A SurveyUSA poll.

**Usage**

```
dream
```

**Format**

A data frame with 910 observations on the following 2 variables.

**ideology** a factor with levels Conservative Liberal Moderate

**stance** a factor with levels No Not sure Yes

**Source**

SurveyUSA, News Poll #18927, data collected Jan 27-29, 2012.

**Examples**

```
table(dream)
```

---

drone_blades	<i>Quadcopter Drone Blades</i>
--------------	--------------------------------

---

**Description**

Quality control dataset for quadcopter drone blades, where this data has been made up for an example.

**Usage**

```
drone_blades
```

**Format**

A data frame with 2000 observations on the following 2 variables.

**supplier** The supplier for the blade.

**inspection** The inspection conclusion.

## References

OpenIntro Statistics, Third Edition and Fourth Edition.

## Examples

```
library(dplyr)

drone_blades |>
  count(supplier, inspection)
```

---

drug_use	<i>Drug use of students and parents</i>
----------	---

---

## Description

Summary of 445 student-parent pairs.

## Usage

```
drug_use
```

## Format

A data frame with 445 observations on the following 2 variables.

**student** a factor with levels not uses

**parents** a factor with levels not used

## Source

Ellis GJ and Stone LH. 1979. Marijuana Use in College: An Evaluation of a Modeling Explanation. Youth and Society 10:323-334.

## Examples

```
table(drug_use)
```

---

duke\_forest*Sale prices of houses in Duke Forest, Durham, NC*

---

**Description**

Data on houses that were recently sold in the Duke Forest neighborhood of Durham, NC in November 2020.

**Usage**

```
duke_forest
```

**Format**

A data frame with 98 rows and 13 variables.

**address** Address of house.

**price** Sale price, in USD.

**bed** Number of bedrooms.

**bath** Number of bathrooms.

**area** Area of home, in square feet.

**type** Type of home (all are Single Family).

**year\_built** Year the home was built.

**heating** Heating system.

**cooling** Cooling system (other or central).

**parking** Type of parking available and number of parking spaces.

**lot** Area of the entire property, in acres.

**hoa** If the home belongs to an Home Owners Association, the associated fee (NA otherwise).

**url** URL of the listing.

**Source**

Data were collected from Zillow in November 2020.

**Examples**

```
library(ggplot2)

# Number of bedrooms and price
ggplot(duke_forest, aes(x = as.factor(bed), y = price)) +
  geom_boxplot() +
  labs(
    x = "Number of bedrooms",
    y = "Sale price (USD)",
    title = "Homes for sale in Duke Forest, Durham, NC",
```

```

    subtitle = "Data are from November 2020"
  )

# Area and price
ggplot(duke_forest, aes(x = area, y = price)) +
  geom_point() +
  labs(
    x = "Area (square feet)",
    y = "Sale price (USD)",
    title = "Homes for sale in Duke Forest, Durham, NC",
    subtitle = "Data are from November 2020"
  )

```

---

earthquakes

*Earthquakes*


---

### Description

Select set of notable earthquakes from 1900 to 1999.

### Usage

```
earthquakes
```

### Format

A data frame with 123 rows and 7 variables.

**year** Year the earthquake took place.

**month** Month the earthquake took place.

**day** Day the earthquake took place

**richter** Magnitude of earthquake using the Richter Scale.

**area** City or geographic location of earthquakes.

**region** Country or countries if the earthquake occurred on a border.

**deaths** Approximate number of deaths caused by earthquake

### Source

World Almanac and Book of Facts: 2011.

### Examples

```

library(ggplot2)

ggplot(earthquakes, aes(x = richter, y = deaths)) +
  geom_point()

ggplot(earthquakes, aes(x = log(deaths))) +
  geom_histogram()

```

---

`ebola_survey`*Survey on Ebola quarantine*

---

**Description**

In New York City on October 23rd, 2014, a doctor who had recently been treating Ebola patients in Guinea went to the hospital with a slight fever and was subsequently diagnosed with Ebola. Soon thereafter, an NBC 4 New York/The Wall Street Journal/Marist Poll asked New Yorkers whether they favored a "mandatory 21-day quarantine for anyone who has come in contact with an Ebola patient". This poll included responses of 1,042 New York adults between October 26th and 28th, 2014.

**Usage**`ebola_survey`**Format**

A data frame with 1042 observations on the following variable.

**quarantine** Indicates whether the respondent is in favor or against the mandatory quarantine.

**Source**

Poll ID NY141026 on [maristpoll.marist.edu](http://maristpoll.marist.edu).

**Examples**

```
table(ebola_survey)
```

---

`edaPlot`*Exploratory data analysis plot*

---

**Description**

Explore different plotting methods using a click interface.

**Usage**

```
edaPlot(  
  dataFrame,  
  Col = c("#888888", "#FF0000", "#222222", "#FFFFFF", "#CCCCC", "#3377AA")  
)
```

**Arguments**

**dataFrame** A data frame.  
**Col** A vector containing six colors. The colors may be given in any form.

**Author(s)**

David Diez

**See Also**

[histPlot](#), [densityPlot](#), [boxPlot](#), [dotPlot](#)

**Examples**

```
data(mlbbat10)
bat <- mlbbat10[mlbbat10$at_bat > 200, ]
# edaPlot(bat)

data(mariokart)
mk <- mariokart[mariokart$total_pr < 100, ]
# edaPlot(mk)
```

---

elmhurst

*Elmhurst College gift aid*

---

**Description**

A random sample of 50 students gift aid for students at Elmhurst College.

**Usage**

elmhurst

**Format**

A data frame with 50 observations on the following 3 variables.

**family\_income** Family income of the student.

**gift\_aid** Gift aid, in \$1000s.

**price\_paid** Price paid by the student (tuition - gift aid).

**Source**

These data were sampled from a table of data for all freshman from the 2011 class at Elmhurst College that accompanied an article titled What Students Really Pay to Go to College published online by The Chronicle of Higher Education: <https://www.chronicle.com/article/what-students-really-pay-to-go-to?sra=true>.



**Examples**

```
library(ggplot2)
library(broom)

ggplot(elmhurst, aes(x = family_income, y = gift_aid)) +
  geom_point() +
  geom_smooth(method = "lm")

mod <- lm(gift_aid ~ family_income, data = elmhurst)
tidy(mod)
```

email

*Data frame representing information about a collection of emails***Description**

These data represent incoming emails for the first three months of 2012 for an email account (see Source).

**Usage**

```
email
```

**Format**

A email (email\_sent) data frame has 3921 (1252) observations on the following 21 variables.

**spam** Indicator for whether the email was spam.

**to\_multiple** Indicator for whether the email was addressed to more than one recipient.

**from** Whether the message was listed as from anyone (this is usually set by default for regular outgoing email).

**cc** Number of people cc'ed.

**sent\_email** Indicator for whether the sender had been sent an email in the last 30 days.

**time** Time at which email was sent.

**image** The number of images attached.

**attach** The number of attached files.

**dollar** The number of times a dollar sign or the word “dollar” appeared in the email.

**winner** Indicates whether “winner” appeared in the email.

**inherit** The number of times “inherit” (or an extension, such as “inheritance”) appeared in the email.

**viagra** The number of times “viagra” appeared in the email.

**password** The number of times “password” appeared in the email.

**num\_char** The number of characters in the email, in thousands.

**line\_breaks** The number of line breaks in the email (does not count text wrapping).

**format** Indicates whether the email was written using HTML (e.g. may have included bolding or active links).

**re\_subj** Whether the subject started with “Re:”, “RE:”, “re:”, or “rE:”

**exclaim\_subj** Whether there was an exclamation point in the subject.

**urgent\_subj** Whether the word “urgent” was in the email subject.

**exclaim\_mess** The number of exclamation points in the email message.

**number** Factor variable saying whether there was no number, a small number (under 1 million), or a big number.

### Source

David Diez’s Gmail Account, early months of 2012. All personally identifiable information has been removed.

### See Also

[email50](#)

### Examples

```
e <- email

# ----- Variables For Logistic Regression -----#
# Variables are modified to match
# OpenIntro Statistics, Second Edition
# As Is (7): spam, to_multiple, winner, format,
#           re_subj, exclaim_subj
# Omitted (6): from, sent_email, time, image,
#           viagra, urgent_subj, number
# Become Indicators (5): cc, attach, dollar,
#           inherit, password
e$cc <- ifelse(email$cc > 0, 1, 0)
e$attach <- ifelse(email$attach > 0, 1, 0)
e$dollar <- ifelse(email$dollar > 0, 1, 0)
e$inherit <- ifelse(email$inherit > 0, 1, 0)
e$password <- ifelse(email$password > 0, 1, 0)
# Transform (3): num_char, line_breaks, exclaim_mess
# e$num_char <- cut(email$num_char, c(0,1,5,10,20,1000))
# e$line_breaks <- cut(email$line_breaks, c(0,10,100,500,10000))
# e$exclaim_mess <- cut(email$exclaim_mess, c(-1,0,1,5,10000))
g <- glm(
  spam ~ to_multiple + winner + format +
    re_subj + exclaim_subj +
    cc + attach + dollar +
    inherit + password, # +
  # num_char + line_breaks + exclaim_mess,
  data = e, family = binomial
)
summary(g)
```

```
# ----- Variable Selection Via AIC -----#
g. <- step(g)
plot(predict(g., type = "response"), e$spam)

# ----- Splitting num_char by html -----#
x <- log(email$num_char)
bw <- 0.004
R <- range(x) + c(-1, 1)
wt <- sum(email$format == 1) / nrow(email)
htmlAll <- density(x, bw = 0.4, from = R[1], to = R[2])
htmlNo <- density(x[email$format != 1],
  bw = 0.4,
  from = R[1], to = R[2]
)
htmlYes <- density(x[email$format == 1],
  bw = 0.4,
  from = R[1], to = R[2]
)
htmlNo$y <- htmlNo$y #* (1-wt)
htmlYes$y <- htmlYes$y #* wt + htmlNo$y
plot(htmlAll, xlim = c(-4, 6), ylim = c(0, 0.4))
lines(htmlNo, col = 4)
lines(htmlYes, lwd = 2, col = 2)
```

---

email50

*Sample of 50 emails*


---

## Description

This is a subsample of the [email](#) dataset.

## Usage

```
email50
```

## Format

A data frame with 50 observations on the following 21 variables.

**spam** Indicator for whether the email was spam.

**to\_multiple** Indicator for whether the email was addressed to more than one recipient.

**from** Whether the message was listed as from anyone (this is usually set by default for regular outgoing email).

**cc** Number of people cc'ed.

**sent\_email** Indicator for whether the sender had been sent an email in the last 30 days.

**time** Time at which email was sent.

**image** The number of images attached.

**attach** The number of attached files.

**dollar** The number of times a dollar sign or the word “dollar” appeared in the email.

**winner** Indicates whether “winner” appeared in the email.

**inherit** The number of times “inherit” (or an extension, such as “inheritance”) appeared in the email.

**viagra** The number of times “viagra” appeared in the email.

**password** The number of times “password” appeared in the email.

**num\_char** The number of characters in the email, in thousands.

**line\_breaks** The number of line breaks in the email (does not count text wrapping).

**format** Indicates whether the email was written using HTML (e.g. may have included bolding or active links).

**re\_subj** Whether the subject started with “Re:”, “RE:”, “re:”, or “rE:”

**exclaim\_subj** Whether there was an exclamation point in the subject.

**urgent\_subj** Whether the word “urgent” was in the email subject.

**exclaim\_mess** The number of exclamation points in the email message.

**number** Factor variable saying whether there was no number, a small number (under 1 million), or a big number.

### Source

David Diez’s Gmail Account, early months of 2012. All personally identifiable information has been removed.

### See Also

[email](#)

### Examples

```
index <- c(
  101, 105, 116, 162, 194, 211, 263, 308, 361, 374,
  375, 465, 509, 513, 571, 691, 785, 842, 966, 968,
  1051, 1201, 1251, 1433, 1519, 1727, 1760, 1777, 1899, 1920,
  1943, 2013, 2052, 2252, 2515, 2629, 2634, 2710, 2823, 2835,
  2944, 3098, 3227, 3360, 3452, 3496, 3530, 3665, 3786, 3877
)
order <- c(
  3, 33, 12, 1, 21, 15, 43, 49, 8, 6,
  34, 25, 24, 35, 41, 9, 22, 50, 4, 48,
  7, 14, 46, 10, 38, 32, 26, 18, 23, 45,
  30, 16, 17, 20, 40, 47, 31, 37, 27, 11,
  5, 44, 29, 19, 13, 36, 39, 42, 28, 2
)
d <- email[index, ][order, ]
identical(d, email50)
```

---

env_regulation	<i>American Adults on Regulation and Renewable Energy</i>
----------------	---

---

## Description

Pew Research conducted a poll to find whether American adults support regulation or believe the private market will move the American economy towards renewable energy.

## Usage

```
env_regulation
```

## Format

A data frame with 705 observations on the following variable.

**statement** There were three possible outcomes for each person: "Regulations necessary", "Private marketplace will ensure", and "Don't know".

## Details

The exact statements being selected were: (1) Government regulations are necessary to encourage businesses and consumers to rely more on renewable energy sources. (2) The private marketplace will ensure that businesses and consumers rely more on renewable energy sources, even without government regulations.

The actual sample size was 1012. However, the original data were not from a simple random sample; after accounting for the design, the equivalent sample size was about 705, which was what was used for the dataset here to keep things simpler for intro stat analyses.

## Source

<https://www.pewresearch.org/science/2017/05/16/public-divides-over-environmental-regulation-and-energy/>

## Examples

```
table(env_regulation)
```

---

epa2012

---

*Vehicle info from the EPA for 2012*


---

**Description**

Details from the EPA.

**Usage**

epa2012

**Format**

A data frame with 1129 observations on the following 28 variables.

**model\_yr** a numeric vector  
**mfr\_name** Manufacturer name.  
**division** Vehicle division.  
**carline** Vehicle line.  
**mfr\_code** Manufacturer code.  
**model\_type\_index** Model type index.  
**engine\_displacement** Engine displacement.  
**no\_cylinders** Number of cylinders.  
**transmission\_speed** Transmission speed.  
**city\_mpg** City mileage.  
**hwy\_mpg** Highway mileage.  
**comb\_mpg** Combined mileage.  
**guzzler** Whether the car is considered a "guzzler" or not, a factor with levels N and Y.  
**air\_aspir\_method** Air aspiration method.  
**air\_aspir\_method\_desc** Air aspiration method description.  
**transmission** Transmission type.  
**transmission\_desc** Transmission type description.  
**no\_gears** Number of gears.  
**trans\_lockup** Whether transmission locks up, a factor with levels N and Y.  
**trans\_creeper\_gear** A factor with level N only.  
**drive\_sys** Drive system, a factor with levels.  
**drive\_desc** Drive system description.  
**fuel\_usage** Fuel usage, a factor with levels.  
**fuel\_usage\_desc** Fuel usage description.  
**class** Class of car.  
**car\_truck** Car or truck, a factor with levels car, 1, 2.  
**release\_date** Date of vehicle release.  
**fuel\_cell** Whether the car has a fuel cell or not, a factor with levels N, Y.

**Source**

Fueleconomy.gov, Shared MPG Estimates: Toyota Prius 2012.

**See Also**

epa2021

**Examples**

```
library(ggplot2)
library(dplyr)

# Variable descriptions
distinct(epa2012, air_aspir_method_desc, air_aspir_method)
distinct(epa2012, transmission_desc, transmission)
distinct(epa2012, drive_desc, drive_sys)
distinct(epa2012, fuel_usage_desc, fuel_usage)

# Guzzlers and their mileages
ggplot(epa2012, aes(x = city_mpg, y = hwy_mpg, color = guzzler)) +
  geom_point() +
  facet_wrap(~guzzler, ncol = 1)
```

---

epa2021

---

*Vehicle info from the EPA for 2021*


---

**Description**

Details from the EPA.

**Usage**

epa2021

**Format**

A data frame with 1108 observations on the following 28 variables.

**model\_yr** a numeric vector  
**mfr\_name** Manufacturer name.  
**division** Vehicle division.  
**carline** Vehicle line.  
**mfr\_code** Manufacturer code.  
**model\_type\_index** Model type index.  
**engine\_displacement** Engine displacement.  
**no\_cylinders** Number of cylinders.

**transmission\_speed** Transmission speed.  
**city\_mpg** City mileage.  
**hwy\_mpg** Highway mileage.  
**comb\_mpg** Combined mileage.  
**guzzler** Whether the car is considered a "guzzler" or not, a factor with levels N and Y.  
**air\_aspir\_method** Air aspiration method.  
**air\_aspir\_method\_desc** Air aspiration method description.  
**transmission** Transmission type.  
**transmission\_desc** Transmission type description.  
**no\_gears** Number of gears.  
**trans\_lockup** Whether transmission locks up, a factor with levels N and Y.  
**trans\_creeper\_gear** A factor with level N only.  
**drive\_sys** Drive system, a factor with levels.  
**drive\_desc** Drive system description.  
**fuel\_usage** Fuel usage, a factor with levels.  
**fuel\_usage\_desc** Fuel usage description.  
**class** Class of car.  
**car\_truck** Car or truck, a factor with levels car, 1, ??, 1.  
**release\_date** Date of vehicle release.  
**fuel\_cell** Whether the car has a fuel cell or not, a factor with levels N, NA.

### Source

Fuel Economy Data from [fueleconomy.gov](https://www.fueleconomy.gov). Retrieved 6 May, 2021.

### See Also

epa2012

### Examples

```
library(ggplot2)
library(dplyr)

# Variable descriptions
distinct(epa2021, air_aspir_method_desc, air_aspir_method)
distinct(epa2021, transmission_desc, transmission)
distinct(epa2021, drive_desc, drive_sys)
distinct(epa2021, fuel_usage_desc, fuel_usage)

# Guzzlers and their mileages
ggplot(epa2021, aes(x = city_mpg, y = hwy_mpg, color = guzzler)) +
  geom_point() +
  facet_wrap(~guzzler, ncol = 1)
```



```
# Compare to 2012
epa2021 |>
  bind_rows(epa2012) |>
  group_by(model_yr) |>
  summarise(
    mean_city = mean(city_mpg),
    mean_hwy = mean(hwy_mpg)
  )
```

---

esi

*Environmental Sustainability Index 2005*


---

## Description

This dataset comes from the 2005 Environmental Sustainability Index: Benchmarking National Environmental Stewardship. Countries are given an overall sustainability score as well as scores in each of several different environmental areas.

## Usage

```
esi
```

## Format

A data frame with 146 observations on the following 29 variables.

**code** ISO3 country code.

**country** Country.

**esi** Environmental Sustainability Index.

**system** ESI core component: systems

**stress** ESI core component: stresses

**vulner** ESI core component: vulnerability

**cap** ESI core component: capacity

**global** ESI core component: global stewardship

**sys\_air** Air quality.

**sys\_bio** Biodiversity.

**sys\_lan** Land.

**sys\_wql** Water quality.

**sys\_wqn** Water quantity.

**str\_air** Reducing air pollution.

**str\_eco** Reducing ecosystem stress.

**str\_pop** Reducing population pressure.

**str\_was** Reducing waste and consumption pressures.  
**str\_wat** Reducing water stress.  
**str\_nrm** Natural resource management.  
**vul\_hea** Environmental health.  
**vul\_sus** Basic human sustenance.  
**vul\_dis** Exposure to natural disasters.  
**cap\_gov** Environmental governance.  
**cap\_eff** Eco-efficiency.  
**cap\_pri** Private sector responsiveness.  
**cap\_st** Science and technology.  
**glo\_col** Participation in international collaboration efforts.  
**glo\_ghg** Greenhouse gas emissions.  
**glo\_tbp** Reducing transboundary environmental pressures.

## Details

ESI and Component scores are presented as standard normal percentiles. Indicator scores are in the form of z-scores. See Appendix A of the report for information on the methodology and Appendix C for more detail on original data sources.

For more information on how each of the indices were calculated, see the documentation linked below.

## Source

ESI Component Indicators. *2005 Environmental Sustainability Index: Benchmarking National Environmental Stewardship*, Yale Center for Environmental Law and Policy, Yale University & Center for International Earth Science Information Network (CIESIN), Columbia University

In collaboration with: World Economic Forum, Geneva, Switzerland Joint Research Centre of the European Commission, Ispra, Italy.

Available at [https://www.earth.columbia.edu/news/2005/images/ESI2005\\_policysummary.pdf](https://www.earth.columbia.edu/news/2005/images/ESI2005_policysummary.pdf).

## References

Esty, Daniel C., Marc Levy, Tanja Srebotnjak, and Alexander de Sherbinin (2005). *2005 Environmental Sustainability Index: Benchmarking National Environmental Stewardship*. New Haven: Yale Center for Environmental Law and Policy

## Examples

```
library(ggplot2)

ggplot(esi, aes(x = cap_st, y = glo_col)) +
  geom_point(color = ifelse(esi$code == "USA", "red", "black")) +
  geom_text(
```

```

    aes(label = ifelse(code == "USA", as.character(code), "")),
    hjust = 1.2, color = "red"
  ) +
  labs(x = "Science and technology", y = "Participation in international collaboration efforts")

ggplot(esi, aes(x = vulner, y = cap)) +
  geom_point(color = ifelse(esi$code == "USA", "red", "black")) +
  geom_text(
    aes(label = ifelse(code == "USA", as.character(code), "")),
    hjust = 1.2, color = "red"
  ) +
  labs(x = "Vulnerability", y = "Capacity")

```

ethanol

*Ethanol Treatment for Tumors Experiment***Description**

Experiment where 3 different treatments of ethanol were tested on the treatment of oral cancer tumors in hamsters.

**Usage**

ethanol

**Format**

A data frame with 24 observations, each representing one hamster, on the following 2 variables.

**treatment** Treatment the hamster received.

**regress** a factor with levels no yes

**Details**

The ethyl\_cellulose and pure\_ethanol treatments consisted of about a quarter of the volume of the tumors, while the pure\_ethanol\_16x treatment was 16x that, so about 4 times the size of the tumors.

**Source**

Morhard R, et al. 2017. Development of enhanced ethanol ablation as an alternative to surgery in treatment of superficial solid tumors. Scientific Reports 7:8750.

**Examples**

```

table(ethanol)
fisher.test(table(ethanol))

```

evals

*Professor evaluations and beauty***Description**

The data are gathered from end of semester student evaluations for 463 courses taught by a sample of 94 professors from the University of Texas at Austin. In addition, six students rate the professors' physical appearance. The result is a data frame where each row contains a different course and each column has information on the course and the professor who taught that course.

**Usage**

evals

**Format**

A data frame with 463 observations on the following 23 variables.

**course\_id** Variable identifying the course (out of 463 courses).

**prof\_id** Variable identifying the professor who taught the course (out of 94 professors).

**score** Average professor evaluation score: (1) very unsatisfactory - (5) excellent.

**rank** Rank of professor: teaching, tenure track, tenured.

**ethnicity** Ethnicity of professor: not minority, minority.

**gender** Gender of professor: female, male.

**language** Language of school where professor received education: English or non-English.

**age** Age of professor.

**cls\_perc\_eval** Percent of students in class who completed evaluation.

**cls\_did\_eval** Number of students in class who completed evaluation.

**cls\_students** Total number of students in class.

**cls\_level** Class level: lower, upper.

**cls\_profs** Number of professors teaching sections in course in sample: single, multiple.

**cls\_credits** Number of credits of class: one credit (lab, PE, etc.), multi credit.

**bty\_flower** Beauty rating of professor from lower level female: (1) lowest - (10) highest.

**bty\_flupper** Beauty rating of professor from upper level female: (1) lowest - (10) highest.

**bty\_f2upper** Beauty rating of professor from second level female: (1) lowest - (10) highest.

**bty\_m1lower** Beauty rating of professor from lower level male: (1) lowest - (10) highest.

**bty\_m1upper** Beauty rating of professor from upper level male: (1) lowest - (10) highest.

**bty\_m2upper** Beauty rating of professor from second upper level male: (1) lowest - (10) highest.

**bty\_avg** Average beauty rating of professor.

**pic\_outfit** Outfit of professor in picture: not formal, formal.

**pic\_color** Color of professor's picture: color, black & white.

**Source**

Daniel S. Hamermesh, Amy Parker, Beauty in the classroom: instructors' pulchritude and putative pedagogical productivity, *Economics of Education Review*, Volume 24, Issue 4, 2005. doi:[10.1016/j.econedurev.2004.07.013](https://doi.org/10.1016/j.econedurev.2004.07.013).

**Examples**

```
evals
```

---

exams	<i>Exam scores</i>
-------	--------------------

---

**Description**

Exam scores from a class of 19 students.

**Usage**

```
exams
```

**Format**

A data frame with 19 observations on the following variable.

**scores** a numeric vector

**Examples**

```
hist(exams$scores)
```

---

exam_grades	<i>Exam and course grades for statistics students</i>
-------------	---

---

**Description**

Grades on three exams and overall course grade for 233 students during several years for a statistics course at a university.

**Usage**

```
exam_grades
```

**Format**

A data frame with 233 observations, each representing a student.

**semester** Semester when grades were recorded.

**sex** Sex of the student as recorded on the university registration system: Man or Woman.

**exam1** Exam 1 grade.

**exam2** Exam 2 grade.

**exam3** Exam 3 grade.

**course\_grade** Overall course grade.

**Examples**

```
library(ggplot2)
library(dplyr)

# Course grade vs. each exam
ggplot(exam_grades, aes(x = exam1, y = course_grade)) +
  geom_point()

ggplot(exam_grades, aes(x = exam2, y = course_grade)) +
  geom_point()

ggplot(exam_grades, aes(x = exam2, y = course_grade)) +
  geom_point()

# Semester averages
exam_grades |>
  group_by(semester) |>
  summarise(across(exam1:course_grade, mean, na.rm = TRUE))
```

---

exclusive\_relationship

*Number of Exclusive Relationships*

---

**Description**

A survey conducted on a reasonably random sample of 203 undergraduates asked, among many other questions, about the number of exclusive relationships these students have been in.

**Usage**

exclusive\_relationship

**Format**

A data frame with 218 observations on the following variable.

**num** Number of exclusive relationships.

**Examples**

```
summary(exclusive_relationship$num)
table(exclusive_relationship$num)
hist(exclusive_relationship$num)
```

---

fact\_opinion

---

*Can Americans categorize facts and opinions?*


---

**Description**

Pew Research Center conducted a survey in 2018, asking a sample of U.S. adults to categorize five factual and five opinion statements. This dataset provides data from this survey, with information on the age group of the participant as well as the number of factual and opinion statements they classified correctly (out of 5).

**Usage**

```
fact_opinion
```

**Format**

A data frame with 5,035 rows and 3 variables.

**age\_group** Age group of survey participant.

**fact\_correct** Number of factual statements classified correctly (out of 5).

**opinion\_correct** Number of opinion statements classified correctly (out of 5).

**Source**

Younger Americans are better than older Americans at telling factual news statements from opinions, Pew Research Center, October 23, 2018.

**Examples**

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(forcats)

# Distribution of fact_correct by age group
ggplot(fact_opinion, aes(x = age_group, y = fact_correct)) +
  geom_boxplot() +
  labs(
    x = "Age group",
    y = "Number correct (factual)",
    title = "Number of factual statements classified correctly by age group"
  )
```

```

# Distribution of opinion_correct by age group
ggplot(fact_opinion, aes(x = age_group, y = opinion_correct)) +
  geom_boxplot() +
  labs(
    x = "Age group",
    y = "Number correct (opinion)",
    title = "Number of opinion statements classified correctly by age group"
  )

# Replicating the figure from Pew report (see source for link)
fact_opinion |>
  mutate(
    facts = case_when(
      fact_correct <= 2 ~ "Two or fewer",
      fact_correct %in% c(3, 4) ~ "Three or four",
      fact_correct == 5 ~ "All five"
    ),
    facts = fct_relevel(facts, "Two or fewer", "Three or four", "All five"),
    opinions = case_when(
      opinion_correct <= 2 ~ "Two or fewer",
      opinion_correct %in% c(3, 4) ~ "Three or four",
      opinion_correct == 5 ~ "All five"
    ),
    opinions = fct_relevel(opinions, "Two or fewer", "Three or four", "All five")
  ) |>
  select(-fact_correct, -opinion_correct) |>
  pivot_longer(cols = -age_group, names_to = "question_type", values_to = "n_correct") |>
  ggplot(aes(y = fct_rev(age_group), fill = n_correct)) +
  geom_bar(position = "fill") +
  facet_wrap(~question_type, ncol = 1) +
  scale_fill_viridis_d(guide = guide_legend(reverse = TRUE)) +
  labs(
    x = "Proportion",
    y = "Age group",
    fill = "Number of\nincorrect\n classifications"
  )

```

---

fadeColor

*Fade colors*


---

## Description

Fade colors so they are transparent.

## Usage

```
fadeColor(col, fade = "FF")
```



**Arguments**

col	An integer, color name, or RGB hexadecimal.
fade	The amount to fade col. This value should be a character in hexadecimal from '00' to 'FF'. The smaller the value, the greater the fading.

**Author(s)**

David Diez

**Examples**

```
data(mariokart)
new <- mariokart$cond == "new"
used <- mariokart$cond == "used"

# ==> color numbers <==#
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80), pch = 20,
  col = 2, cex = 2, main = "using regular colors"
)
dotPlot(mariokart$total_pr[used], at = 2, add = TRUE, col = 4, pch = 20, cex = 2)
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80),
  col = fadeColor(2, "22"), pch = 20, cex = 2,
  main = "fading the colors first"
)
dotPlot(mariokart$total_pr[used],
  at = 2, add = TRUE,
  col = fadeColor(4, "22"), pch = 20, cex = 2
)

# ==> color names <==#
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80), pch = 20,
  col = "red", cex = 2, main = "using regular colors"
)
dotPlot(mariokart$total_pr[used], at = 2, add = TRUE, col = "blue", pch = 20, cex = 2)
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80),
  col = fadeColor("red", "22"), pch = 20, cex = 2,
  main = "fading the colors first"
)
dotPlot(mariokart$total_pr[used],
  at = 2, add = TRUE,
  col = fadeColor("blue", "22"), pch = 20, cex = 2
)

# ==> hexadecimal <==#
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80), pch = 20,
  col = "#FF0000", cex = 2, main = "using regular colors"
```

```

)
dotPlot(mariokart$total_pr[used],
  at = 2, add = TRUE, col = "#0000FF", pch = 20,
  cex = 2
)
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80),
  col = fadeColor("#FF0000", "22"), pch = 20, cex = 2,
  main = "fading the colors first"
)
dotPlot(mariokart$total_pr[used],
  at = 2, add = TRUE,
  col = fadeColor("#0000FF", "22"), pch = 20, cex = 2
)

# ==> alternative: rgb function <==#
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80), pch = 20,
  col = rgb(1, 0, 0), cex = 2, main = "using regular colors"
)
dotPlot(mariokart$total_pr[used],
  at = 2, add = TRUE, col = rgb(0, 0, 1),
  pch = 20, cex = 2
)
dotPlot(mariokart$total_pr[new],
  ylim = c(0, 3), xlim = c(25, 80),
  col = rgb(1, 0, 0, 1 / 8), pch = 20, cex = 2,
  main = "fading the colors first"
)
dotPlot(mariokart$total_pr[used],
  at = 2, add = TRUE,
  col = rgb(0, 0, 1, 1 / 8), pch = 20, cex = 2
)

```

---

family\_college

*Simulated sample of parent / teen college attendance*


---

## Description

A simulated dataset based on real population summaries.

## Usage

```
family_college
```

## Format

A data frame with 792 observations on the following 2 variables.

**teen** Whether the teen goes to college or not.

**parents** Whether the parent holds a college degree or not.

**Source**

Simulation based off of summary information provided at <https://eric.ed.gov/?id=ED460660>.

**Examples**

```
library(dplyr)

family_college |>
  count(teen, parents)
```

---

fastfood

*Nutrition in fast food*

---

**Description**

Nutrition amounts in 515 fast food items. The author of the data scraped only entrees (not sides, drinks, desserts, etc.).

**Usage**

```
fastfood
```

**Format**

A data frame with 515 observations on the following 17 variables.

**restaurant** Name of restaurant

**item** Name of item

**calories** Number of calories

**cal\_fat** Calories from fat

**total\_fat** Total fat

**sat\_fat** Saturated fat

**trans\_fat** Trans fat

**cholesterol** Cholesterol

**sodium** Sodium

**total\_carb** Total carbs

**fiber** Fiber

**sugar** Suger

**protein** Protein

**vit\_a** Vitamin A

**vit\_c** Vitamin C

**calcium** Calcium

**salad** Salad or not

**Source**

Retrieved from [Tidy Tuesday Fast food entree data](#).

---

fcid	<i>Summary of male heights from USDA Food Commodity Intake Database</i>
------	---

---

**Description**

Sample of heights based on the weighted sample in the survey.

**Usage**

fcid

**Format**

A data frame with 100 observations on the following 2 variables.

**height** a numeric vector

**num\_of\_adults** a numeric vector

**Examples**

fcid

---

fheights	<i>Female college student heights, in inches</i>
----------	--

---

**Description**

24 sample observations.

**Usage**

fheights

**Format**

A data frame with 24 observations on the following variable.

**heights** height, in inches

**Examples**

hist(fheights\$heights)

---

fish_age	<i>Young fish in the North Sea.</i>
----------	-------------------------------------

---

## Description

Samples of 50 Tobis fish, or Sand Eels, were collected at three different locations in the North Sea and the number of one-year-old fish were counted.

## Usage

fish\_age

## Format

A data frame with 300 rows and 3 variables:

**year** Year the fish was caught with levels 1997 and 1998.

**location** Site the fish was caught with levels A, B and C.

**one\_year\_old** Is the fish one-year-old, yes or no?

## Source

Henrik Madsen, Paul Thyregod. 2011. Introduction to General and Generalized Linear Models  
CRC Press. Boca Raton, FL. ISBN: 978-1-4200-9155-7 [Website](#)

## Examples

```
library(dplyr)
library(tidyr)

# Count the number of one-year-old fish at each location.

fish_age |>
  filter(one_year_old == "yes") |>
  count(year, location) |>
  pivot_wider(names_from = location, values_from = n)
```

fish\_oil\_18

*Findings on n-3 Fatty Acid Supplement Health Benefits***Description**

The results summarize each of the health outcomes for an experiment where 12,933 subjects received a 1g fish oil supplement daily and 12,938 received a placebo daily. The experiment's duration was 5-years.

**Usage**

fish\_oil\_18

**Format**

The format is a list of 24 matrices. Each matrix is a 2x2 table, and below are the named items in the list, which also represent the outcomes.

**major\_cardio\_event** Major cardiovascular event. (Primary end point.)

**cardio\_event\_expanded** Cardiovascular event in expanded composite endpoint.

**myocardial\_infarction** Total myocardial infarction. (Heart attack.)

**stroke** Total stroke.

**cardio\_death** Death from cardiovascular causes.

**PCI** Percutaneous coronary intervention.

**CABG** Coronary artery bypass graft.

**total\_coronary\_heart\_disease** Total coronary heart disease.

**ischemic\_stroke** Ischemic stroke.

**hemorrhagic\_stroke** Hemorrhagic stroke.

**chd\_death** Death from coronary heart disease.

**myocardial\_infarction\_death** Death from myocardial infarction.

**stroke\_death** Death from stroke.

**invasive\_cancer** Invasive cancer of any type. (Primary end point.)

**breast\_cancer** Breast cancer.

**prostate\_cancer** Prostate cancer.

**colorectal\_cancer** Colorectal cancer.

**cancer\_death** Death from cancer.

**death** Death from any cause.

**major\_cardio\_event\_after\_2y** Major cardiovascular event, excluding the first 2 years of follow-up.

**myocardial\_infarction\_after\_2y** Total myocardial infarction, excluding the first 2 years of follow-up.

**invasive\_cancer\_after\_2y** Invasive cancer of any type, excluding the first 2 years of follow-up.

**cancer\_death\_after\_2y** Death from cancer, excluding the first 2 years of follow-up.

**death\_after\_2y** Death from any cause, excluding the first 2 years of follow-up.

**Source**

Manson JE, et al. 2018. Marine n-3 Fatty Acids and Prevention of Cardiovascular Disease and Cancer. NEJMoa1811403. doi:[10.1056/NEJMoa1811403](https://doi.org/10.1056/NEJMoa1811403).

**Examples**

```
names(fish_oil_18)
(tab <- fish_oil_18[["major_cardio_event"]])
chisq.test(tab)
fisher.test(tab)

(tab <- fish_oil_18[["myocardial_infarction"]])
chisq.test(tab)
fisher.test(tab)
```

---

flow\_rates

*River flow data*


---

**Description**

Flow rates (measured in cubic feet per second) of Clarks Creek, Leach Creek, Silver Creek, and Wildwood Creek Spring collected by volunteers of the Pierce Conservation District in the State of Washington in the US.

**Usage**

```
flow_rates
```

**Format**

A data frame with 31 rows and 3 variables.

**site** Location where measurements were taken.

**date** Date measurements were taken.

**flow** Flow rate of the river in cubic feet per second.

**Source**

[Pierce County Water Data Viewer](#).

**Examples**

```
library(ggplot2)

# River flow rates by site
ggplot(flow_rates, aes(x = site, y = flow)) +
  geom_boxplot() +
  labs(
```

```

    title = "River flow rates by site",
    x = "Site",
    y = expression(paste("Flow (ft"^3 * "/s)"))
  )

# River flow rates over time
ggplot(flow_rates, aes(x = date, y = flow, color = site, shape = site)) +
  geom_point(size = 2) +
  labs(
    title = "River flow rates over time",
    x = "Date",
    y = expression(paste("Flow (ft"^3 * "/s)")),
    color = "Site", shape = "Site"
  )

```

friday

*Friday the 13th*

## Description

This dataset addresses issues of how superstitions regarding Friday the 13th affect human behavior, and whether Friday the 13th is an unlucky day. Scanlon, et al. collected data on traffic and shopping patterns and accident frequency for Fridays the 6th and 13th between October of 1989 and November of 1992.

## Usage

friday

## Format

A data frame with 61 observations and 6 variables.

**type** Type of observation, traffic, shopping, or accident.

**date** Year and month of observation.

**sixth** Counts on the 6th of the month.

**thirteenth** Counts on the 13th of the month.

**diff** Difference between the sixth and the thirteenth.

**location** Location where data is collected.

## Details

There are three types of observations: traffic, shopping, and accident. For traffic, the researchers obtained information from the British Department of Transport regarding the traffic flows between junctions 7 to 8 and junctions 9 to 10 of the M25 motorway. For shopping, they collected the numbers of shoppers in nine different supermarkets in southeast England. For accidents, they collected numbers of emergency admissions to hospitals due to transport accidents.



**Source**

Scanlon, T.J., Luben, R.N., Scanlon, F.L., Singleton, N. (1993), "Is Friday the 13th Bad For Your Health?," BMJ, 307, 1584-1586. <https://das1.datadescription.com/datafile/friday-the-13th-traffic> and <https://das1.datadescription.com/datafile/friday-the-13th-accidents>.

**Examples**

```
library(dplyr)
library(ggplot2)

friday |>
  filter(type == "traffic") |>
  ggplot(aes(x = sixth)) +
  geom_histogram(binwidth = 2000) +
  xlim(110000, 140000)

friday |>
  filter(type == "traffic") |>
  ggplot(aes(x = thirteenth)) +
  geom_histogram(binwidth = 2000) +
  xlim(110000, 140000)
```

full\_body\_scan

*Poll about use of full-body airport scanners***Description**

Poll about use of full-body airport scanners, where about 4-in-5 people supported the use of the scanners.

**Usage**

```
full_body_scan
```

**Format**

A data frame with 1137 observations on the following 2 variables.

**answer** a factor with levels do not know / no answer should should not

**party.affiliation** a factor with levels Democrat Independent Republican

**Source**

S. Condon. Poll: 4 in 5 Support Full-Body Airport Scanners. In: CBS News (2010).

**Examples**

```
full_body_scan
```

---

gdp_countries	<i>GDP Countries Data.</i>
---------------	----------------------------

---

### Description

From World Bank, GDP in current U.S. dollars 1960-2020 by decade

### Usage

```
gdp_countries
```

### Format

A data frame with 659 rows and 9 variables.

**country** Name of country.

**description** description of data: GDP (in current US\$), GDP growth (annual %), GDP per capita (in current US\$)

**year\_1960** value in 1960

**year\_1970** value in 1970

**year\_1980** value in 1980

**year\_1990** value in 1990

**year\_2000** value in 2000

**year\_2010** value in 2010

**year\_2020** value in 2020

### Source

[World Bank](#)

### Examples

```
library(dplyr)
# don't use scientific notation
options(scipen = 999)
# List the top 10 countries by GDP (There is a row for World)
gdp_countries |>
  filter(description == "GDP") |>
  mutate(year2020 = format(year_2020, big.mark = ",")) |>
  select(country, year2020) |>
  arrange(desc(year2020)) |>
  top_n(n = 11)

# List the 10 countries with the biggest GDP per capita change from 1960 to 2020
gdp_countries |>
  filter(description == "GDP per capita") |>
```

```
mutate(change = format(round(year_2020 - year_1960, 0), big.mark = ",")) |>
select(country, change, year_1960, year_2020) |>
na.omit() |>
arrange(desc(change)) |>
top_n(n = 10)
```

---

gear\_company

*Fake data for a gear company example*


---

### Description

Made-up data for whether a sample of two gear companies' parts pass inspection.

### Usage

```
gear_company
```

### Format

A data frame with 2000 observations on the following 2 variables.

**company** a factor with levels current prospective

**outcome** a factor with levels not pass

### Examples

```
gear_company
```

---

gender\_discrimination

*Bank manager recommendations based on gender*


---

### Description

Study from the 1970s about whether gender influences hiring recommendations.

### Usage

```
gender_discrimination
```

### Format

A data frame with 48 observations on the following 2 variables.

**gender** a factor with levels female and male

**decision** a factor with levels not promoted and promoted

### Source

Rosen B and Jerdee T. 1974. Influence of sex role stereotypes on personnel decisions. Journal of Applied Psychology 59(1):9-14.

### Examples

```
library(ggplot2)

table(gender_discrimination)

ggplot(gender_discrimination, aes(y = gender, fill = decision)) +
  geom_bar(position = "fill")
```

---

get_it_dunn_run	<i>Get it Dunn Run, Race Times</i>
-----------------	------------------------------------

---

### Description

Get it Dunn is a small regional run that got extra attention when a runner, Nichole Porath, made the Guinness Book of World Records for the fastest time pushing a double stroller in a half marathon. This dataset contains results from the 2017 and 2018 races.

### Usage

```
get_it_dunn_run
```

### Format

A data frame with 978 observations on the following 10 variables.

**date** Date of the run.  
**race** Run distance.  
**bib\_num** Bib number of the runner.  
**first\_name** First name of the runner.  
**last\_initial** Initial of the runner's last name.  
**sex** Sex of the runner.  
**age** Age of the runner.  
**city** City of residence.  
**state** State of residence.  
**run\_time\_minutes** Run time, in minutes.

### Source

Data were collected from GSE Timing: [2018 data](#), [2017 race data](#).

### Examples

```
d <- subset(
  get_it_dunn_run,
  race == "5k" & date == "2018-05-12" &
  !is.na(age) & state %in% c("MN", "WI")
)
head(d)
m <- lm(run_time_minutes ~ sex + age + state, d)
summary(m)
plot(m$fitted, m$residuals)
boxplot(m$residuals ~ d$sex)
plot(m$residuals ~ d$age)
hist(m$residuals)
```

---

gifted

*Analytical skills of young gifted children*

---

### Description

An investigator is interested in understanding the relationship, if any, between the analytical skills of young gifted children and the following variables: father's IQ, mother's IQ, age in month when the child first said "mummy" or "daddy", age in month when the child first counted to 10 successfully, average number of hours per week the child's mother or father reads to the child, average number of hours per week the child watched an educational program on TV during the past three months, average number of hours per week the child watched cartoons on TV during the past three months. The analytical skills are evaluated using a standard testing procedure, and the score on this test is used as the response variable.

### Usage

gifted

### Format

A data frame with 36 observations and 8 variables.

**score** Score in test of analytical skills.

**fatheriq** Father's IQ.

**motheriq** Mother's IQ.

**speak** Age in months when the child first said "mummy" or "daddy".

**count** Age in months when the child first counted to 10 successfully.

**read** Average number of hours per week the child's mother or father reads to the child.

**edutv** Average number of hours per week the child watched an educational program on TV during the past three months.

**cartoons** Average number of hours per week the child watched cartoons on TV during the past three months.

**Details**

Data were collected from schools in a large city on a set of thirty-six children who were identified as gifted children soon after they reached the age of four.

**Source**

Graybill, F.A. & Iyer, H.K., (1994) Regression Analysis: Concepts and Applications, Duxbury, p. 511-6.

**Examples**

gifted

---

global_warming_pew	<i>Pew survey on global warming</i>
--------------------	-------------------------------------

---

**Description**

A 2010 Pew Research poll asked 1,306 Americans, "From what you've read and heard, is there solid evidence that the average temperature on earth has been getting warmer over the past few decades, or not?"

**Usage**

global\_warming\_pew

**Format**

A data frame with 2253 observations on the following 2 variables.

**party\_or\_ideology** a factor with levels Conservative Republican Liberal Democrat Mod/Cons Democrat Mod/Lib Republican

**response** Response.

**Source**

Pew Research Center, Majority of Republicans No Longer See Evidence of Global Warming, data collected on October 27, 2010.

**Examples**

global\_warming\_pew

---

`goog`*Google stock data*

---

**Description**

Google stock data from 2006 to early 2014, where data from the first day each month was collected.

**Usage**`goog`**Format**

A data frame with 98 observations on the following 7 variables.

**date** a factor with levels 2006-01-03, 2006-02-01, and so on

**open** a numeric vector

**high** a numeric vector

**low** a numeric vector

**close** a numeric vector

**volume** a numeric vector

**adj\_close** a numeric vector

**Source**

Yahoo! Finance.

**Examples**`goog`

---

`gov_poll`*Pew Research poll on government approval ratings*

---

**Description**

The poll's focus is on Obama and then Democrats and Republicans in Congress.

**Usage**`gov_poll`

**Format**

A data frame with 4223 observations on the following 2 variables.

- poll** a factor with levels approve disapprove
- eval** a factor with levels Democrats Obama Republicans

**Source**

See the Pew Research website: [www.people-press.org/2012/03/14/romney-leads-gop-contest-trails-in-matchup-with-obama](http://www.people-press.org/2012/03/14/romney-leads-gop-contest-trails-in-matchup-with-obama). The counts in Table 6.19 are approximate.

**Examples**

gov\_poll

---

gpa	<i>Survey of Duke students on GPA, studying, and more</i>
-----	---

---

**Description**

A survey of 55 Duke University students asked about their GPA, number of hours they study at night, number of nights they go out, and their gender.

**Usage**

gpa

**Format**

A data frame with 55 observations on the following 5 variables.

- gpa** a numeric vector
- studyweek** a numeric vector
- sleepnight** a numeric vector
- out** a numeric vector
- gender** a factor with levels female male

**Examples**

gpa



---

gpa\_iq*Sample of students and their GPA and IQ*

---

**Description**

Data on 78 students including GPA, IQ, and gender.

**Usage**

gpa\_iq

**Format**

A data frame with 78 observations representing students on the following 5 variables.

**obs** a numeric vector

**gpa** Grade point average (GPA).

**iq** IQ.

**gender** Gender.

**concept** a numeric vector

**Examples**

gpa\_iq

---

gpa\_study\_hours*gpa\_study\_hours*

---

**Description**

A data frame with 193 rows and 2 columns. The columns represent the variables gpa and study\_hours for a sample of 193 undergraduate students who took an introductory statistics course in 2012 at a private US university.

**Usage**

gpa\_study\_hours

**Format**

A data frame with 193 observations on the following 2 variables.

**gpa** Grade point average (GPA) of student.

**study\_hours** Number of hours students study per week.

### Details

GPA ranges from 0 to 4 points, however one student reported a GPA > 4. This is a data error but this observation has been left in the dataset as it is used to illustrate issues with real survey data. Both variables are self reported, hence may not be accurate.

### Source

Collected at a private US university as part of an anonymous survey in an introductory statistics course.

### Examples

```
library(ggplot2)

ggplot(gpa_study_hours, aes(x = study_hours, y = gpa)) +
  geom_point(alpha = 0.5) +
  labs(x = "Study hours/week", y = "GPA")
```

---

gradestv

---

*Simulated data for analyzing the relationship between watching TV and grades*


---

### Description

This is a simulated dataset to be used to estimate the relationship between number of hours per week students watch TV and the grade they got in a statistics class.

### Usage

```
gradestv
```

### Format

A data frame with 25 observations on the following 2 variables.

**tv** Number of hours per week students watch TV.

**grades** Grades students got in a statistics class (out of 100).

### Details

There are a few potential outliers in this dataset. When analyzing the data one should consider how (if at all) these outliers may affect the estimates of correlation coefficient and regression parameters.

### Source

Simulated data

**Examples**

```
library(ggplot2)

ggplot(gradestv, aes(x = tv, y = grades)) +
  geom_point() +
  geom_smooth(method = "lm")
```

gsearch

*Simulated Google search experiment***Description**

The data were simulated to look like sample results from a Google search experiment.

**Usage**

```
gsearch
```

**Format**

A data frame with 10000 observations on the following 2 variables.

**type** a factor with levels new search no new search

**outcome** a factor with levels current test 1 test 2

**Examples**

```
library(ggplot2)

table(gsearch$type, gsearch$outcome)

ggplot(gsearch, aes(x = type, fill = outcome)) +
  geom_bar(position = "fill") +
  labs(y = "proportion")
```

gss2010

*2010 General Social Survey***Description**

Data from the 2010 General Social Survey.

**Usage**

```
gss2010
```

**Format**

A data frame with 2044 observations on the following 5 variables.

**hrsrelax** After an average work day, about how many hours do you have to relax or pursue activities that you enjoy

**mntlhlth** For how many days during the past 30 days was your mental health, which includes stress, depression, and problems with emotions, not good?

**hrs1** Hours worked each week.

**degree** Educational attainment or degree.

**grass** Do you think the use of marijuana should be made legal, or not?

**Source**

US 2010 General Social Survey.

**Examples**

```
gss2010
```

---

```
gss_wordsum_class      gss_wordsum_class
```

---

**Description**

A data frame containing data from the General Social Survey.

**Usage**

```
gss_wordsum_class
```

**Format**

A data frame with 795 observations on the following 2 variables.

**wordsum** A vocabulary score calculated based on a ten question vocabulary test, where a higher score means better vocabulary. Scores range from 1 to 10.

**class** Self-identified social class has 4 levels: lower, working, middle, and upper class.

**Examples**

```
library(dplyr)

gss_wordsum_class |>
  group_by(class) |>
  summarize(mean_wordsum = mean(wordsum))
```

---

healthcare\_law\_survey *Pew Research Center poll on health care, including question variants*

---

### Description

For example, Pew Research Center conducted a survey with the following question: "As you may know, by 2014 nearly all Americans will be required to have health insurance. People who do not buy insurance will pay a penalty while people who cannot afford it will receive financial help from the government. Do you approve or disapprove of this policy?" For each randomly sampled respondent, the statements in brackets were randomized: either they were kept in the order given above, or the two statements were reversed.

### Usage

```
healthcare_law_survey
```

### Format

A data frame with 1503 observations on the following 2 variables.

**order** a factor with levels cannot\_afford\_second penalty\_second

**response** a factor with levels approve disapprove other

### Source

[www.people-press.org/2012/03/26/public-remains-split-on-health-care-bill-opposed-to-mandate/](http://www.people-press.org/2012/03/26/public-remains-split-on-health-care-bill-opposed-to-mandate/). Sample sizes for each polling group are approximate.

### Examples

```
healthcare_law_survey
```

---

health_coverage	<i>Health Coverage and Health Status</i>
-----------------	--

---

### Description

Survey responses for 20,000 responses to the Behavioral Risk Factor Surveillance System.

### Usage

```
health_coverage
```

**Format**

A data frame with 20000 observations on the following 2 variables.

**coverage** Whether the person had health coverage or not.

**health\_status** The person's health status.

**Source**

Office of Surveillance, Epidemiology, and Laboratory Services Behavioral Risk Factor Surveillance System, BRFSS 2010 Survey Data.

**Examples**

```
table(health_coverage)
```

---

heart_transplant	<i>Heart Transplant Data</i>
------------------	------------------------------

---

**Description**

The Stanford University Heart Transplant Study was conducted to determine whether an experimental heart transplant program increased lifespan. Each patient entering the program was designated officially a heart transplant candidate, meaning that he was gravely ill and would most likely benefit from a new heart. Then the actual heart transplant occurs between a few weeks to several months depending on the availability of a donor. Very few candidates during this waiting period show improvement and get *deselected* as a heart transplant candidate, but for the purposes of this experiment those patients were kept in the data as continuing candidates.

**Usage**

```
heart_transplant
```

**Format**

A data frame with 103 observations on the following 8 variables.

**id** ID number of the patient.

**acceptyear** Year of acceptance as a heart transplant candidate.

**age** Age of the patient at the beginning of the study.

**survived** Survival status with levels alive and dead.

**survtime** Number of days patients were alive after the date they were determined to be a candidate for a heart transplant until the termination date of the study

**prior** Whether or not the patient had prior surgery with levels yes and no.

**transplant** Transplant status with levels control (did not receive a transplant) and treatment (received a transplant).

**wait** Waiting Time for Transplant

**Source**

<http://www.stat.ucla.edu/~jsanchez/data/stanford.txt>

**References**

Turnbull B, Brown B, and Hu M (1974). "Survivorship of heart transplant data." Journal of the American Statistical Association, vol. 69, pp. 74-80.

**Examples**

```
library(ggplot2)

ggplot(heart_transplant, aes(x = transplant, y = survtime)) +
  geom_boxplot() +
  labs(x = "Transplant", y = "Survival time (days)")

ggplot(heart_transplant, aes(x = transplant, fill = survived)) +
  geom_bar(position = "fill") +
  labs(x = "Transplant", y = "Proportion", fill = "Outcome")
```

---

helium

*Helium football*


---

**Description**

At the 1976 Pro Bowl, Ray Guy, a punter for the Oakland Raiders, punted a ball that hung mid-air long enough for officials to question whether the pigskin was filled with helium. The ball was found to be filled with air, but since then many have tossed around the idea that a helium-filled football would outdistance an air-filled one. Students at Ohio State University conducted an experiment to test this myth. They used two identical footballs, one air filled with air and one filled with helium. Each football was kicked 39 times and the two footballs were alternated with each kick.

**Usage**

```
helium
```

**Format**

A data frame with 39 observations on the following 3 variables.

**trial** Trial number.

**air** Distance in years for air-filled football.

**helium** Distance in years for helium-filled football.

**Details**

Lafferty, M. B. (1993), "OSU scientists get a kick out of sports controversy, "The Columbus Dispatch (November, 21, 1993), B7.

## Source

Previously part of the Data and Story Library, <https://dasl.datadescription.com>. Removed as of 2020.

## Examples

```
boxPlot(helium$air, xlab = "air")
boxPlot(helium$helium, xlab = "helium")
```

---

helmet

*Socioeconomic status and reduced-fee school lunches*

---

## Description

Examining the relationship between socioeconomic status measured as the percentage of children in a neighborhood receiving reduced-fee lunches at school (lunch) and the percentage of bike riders in the neighborhood wearing helmets (helmet).

## Usage

helmet

## Format

A data frame with 12 observations representing neighborhoods on the following 2 variables.

**lunch** Percent of students receiving reduced-fee school lunches.

**helmet** Percent of bike riders wearing helmets.

## Examples

```
library(ggplot2)

ggplot(helmet, aes(x = lunch, y = helmet)) +
  geom_point()
```



hfi

*Human Freedom Index***Description**

The Human Freedom Index is a report that attempts to summarize the idea of "freedom" through a bunch of different variables for many countries around the globe. It serves as a rough objective measure for the relationships between the different types of freedom - whether it's political, religious, economical or personal freedom - and other social and economic circumstances. The Human Freedom Index is an annually co-published report by the Cato Institute, the Fraser Institute, and the Liberales Institut at the Friedrich Naumann Foundation for Freedom.

**Usage**

hfi

**Format**

A data frame with 1458 observations on the following 123 variables.

**year** Year

**ISO\_code** ISO code of country

**countries** Name of country

**region** Region where country is located

**pf\_rol\_procedural** Procedural justice

**pf\_rol\_civil** Civil justice

**pf\_rol\_criminal** Criminal justice

**pf\_rol** Rule of law

**pf\_ss\_homicide** Homicide

**pf\_ss\_disappearances\_disap** Disappearances

**pf\_ss\_disappearances\_violent** Violent conflicts

**pf\_ss\_disappearances\_organized** Violent conflicts

**pf\_ss\_disappearances\_fatalities** Terrorism fatalities

**pf\_ss\_disappearances\_injuries** Terrorism injuries

**pf\_ss\_disappearances** Disappearances, conflict, and terrorism

**pf\_ss\_women\_fgm** Female genital mutilation

**pf\_ss\_women\_missing** Missing women

**pf\_ss\_women\_inheritance\_widows** Inheritance rights for widows

**pf\_ss\_women\_inheritance\_daughters** Inheritance rights for daughters

**pf\_ss\_women\_inheritance** Inheritance

**pf\_ss\_women** Women's security

**pf\_ss** Security and safety  
**pf\_movement\_domestic** Freedom of domestic movement  
**pf\_movement\_foreign** Freedom of foreign movement  
**pf\_movement\_women** Women's movement  
**pf\_movement** Freedom of movement  
**pf\_religion\_estop\_establish** Freedom to establish religious organizations  
**pf\_religion\_estop\_operate** Freedom to operate religious organizations  
**pf\_religion\_estop** Freedom to establish and operate religious organizations  
**pf\_religion\_harassment** Harassment and physical hostilities  
**pf\_religion\_restrictions** Legal and regulatory restrictions  
**pf\_religion** Religious freedom  
**pf\_association\_association** Freedom of association  
**pf\_association\_assembly** Freedom of assembly  
**pf\_association\_political\_establish** Freedom to establish political parties  
**pf\_association\_political\_operate** Freedom to operate political parties  
**pf\_association\_political** Freedom to establish and operate political parties  
**pf\_association\_prof\_establish** Freedom to establish professional organizations  
**pf\_association\_prof\_operate** Freedom to operate professional organizations  
**pf\_association\_prof** Freedom to establish and operate professional organizations  
**pf\_association\_sport\_establish** Freedom to establish educational, sporting, and cultural organizations  
**pf\_association\_sport\_operate** Freedom to operate educational, sporting, and cultural organizations  
**pf\_association\_sport** Freedom to establish and operate educational, sporting, and cultural organizations  
**pf\_association** Freedom to associate and assemble with peaceful individuals or organizations  
**pf\_expression\_killed** Press killed  
**pf\_expression\_jailed** Press jailed  
**pf\_expression\_influence** Laws and regulations that influence media content  
**pf\_expression\_control** Political pressures and controls on media content  
**pf\_expression\_cable** Access to cable/satellite  
**pf\_expression\_newspapers** Access to foreign newspapers  
**pf\_expression\_internet** State control over internet access  
**pf\_expression** Freedom of expression  
**pf\_identity\_legal** Legal gender  
**pf\_identity\_parental\_marriage** Parental rights in marriage  
**pf\_identity\_parental\_divorce** Parental rights after divorce  
**pf\_identity\_parental** Parental rights

**pf\_identity\_sex\_male** Male-to-male relationships  
**pf\_identity\_sex\_female** Female-to-female relationships  
**pf\_identity\_sex** Same-sex relationships  
**pf\_identity\_divorce** Divor  
**pf\_identity** Identity and relationships  
**pf\_score** Personal Freedom (score)  
**pf\_rank** Personal Freedom (rank)  
**ef\_government\_consumption** Government consumption  
**ef\_government\_transfers** Transfers and subsidies  
**ef\_government\_enterprises** Government enterprises and investments  
**ef\_government\_tax\_income** Top marginal income tax rate - Top marginal income tax rates  
**ef\_government\_tax\_payroll** Top marginal income tax rate - Top marginal income and payroll tax rate  
**ef\_government\_tax** Top marginal tax rate  
**ef\_government** Size of government  
**ef\_legal\_judicial** Judicial independence  
**ef\_legal\_courts** Impartial courts  
**ef\_legal\_protection** Protection of property rights  
**ef\_legal\_military** Military interference in rule of law and politics  
**ef\_legal\_integrity** Integrity of the legal system  
**ef\_legal\_enforcement** Legal enforcement of contracts  
**ef\_legal\_restrictions** Regulatory restrictions on the sale of real property  
**ef\_legal\_police** Reliability of police  
**ef\_legal\_crime** Business costs of crime  
**ef\_legal\_gender** Gender adjustment  
**ef\_legal** Legal system and property rights  
**ef\_money\_growth** Money growth  
**ef\_money\_sd** Standard deviation of inflation  
**ef\_money\_inflation** Inflation - most recent year  
**ef\_money\_currency** Freedom to own foreign currency bank account  
**ef\_money** Sound money  
**ef\_trade\_tariffs\_revenue** Tariffs - Revenue from trade taxes (percentage of trade sector)  
**ef\_trade\_tariffs\_mean** Tariffs - Mean tariff rate  
**ef\_trade\_tariffs\_sd** Tariffs - Standard deviation of tariffs rates  
**ef\_trade\_tariffs** Tariffs  
**ef\_trade\_regulatory\_nontariff** Regulatory trade barriers - Nontariff trade barriers  
**ef\_trade\_regulatory\_compliance** Regulatory trade barriers - Compliance costs of importing and exporting

**ef\_trade\_regulatory** Regulatory trade barriers  
**ef\_trade\_black** Black-market exchange rates  
**ef\_trade\_movement\_foreign** Controls of the movement of capital and people - Foreign ownership/investment restrictions  
**ef\_trade\_movement\_capital** Controls of the movement of capital and people - Capital controls  
**ef\_trade\_movement\_visit** Controls of the movement of capital and people - Freedom of foreigners to visit  
**ef\_trade\_movement** Controls of the movement of capital and people  
**ef\_trade** Freedom to trade internationally  
**ef\_regulation\_credit\_ownership** Credit market regulations - Ownership of banks  
**ef\_regulation\_credit\_private** Credit market regulations - Private sector credit  
**ef\_regulation\_credit\_interest** Credit market regulations - Interest rate controls/negative real interest rates  
**ef\_regulation\_credit** Credit market regulation  
**ef\_regulation\_labor\_minwage** Labor market regulations - Hiring regulations and minimum wage  
**ef\_regulation\_labor\_firing** Labor market regulations - Hiring and firing regulations  
**ef\_regulation\_labor\_bargain** Labor market regulations - Centralized collective bargaining  
**ef\_regulation\_labor\_hours** Labor market regulations - Hours regulations  
**ef\_regulation\_labor\_dismissal** Labor market regulations - Dismissal regulations  
**ef\_regulation\_labor\_conscription** Labor market regulations - Conscription  
**ef\_regulation\_labor** Labor market regulation  
**ef\_regulation\_business\_adm** Business regulations - Administrative requirements  
**ef\_regulation\_business\_bureaucracy** Business regulations - Bureaucracy costs  
**ef\_regulation\_business\_start** Business regulations - Starting a business  
**ef\_regulation\_business\_bribes** Business regulations - Extra payments/bribes/favoritism  
**ef\_regulation\_business\_licensing** Business regulations - Licensing restrictions  
**ef\_regulation\_business\_compliance** Business regulations - Cost of tax compliance  
**ef\_regulation\_business** Business regulation  
**ef\_regulation** Economic freedom regulation score  
**ef\_score** Economic freedom score  
**ef\_rank** Economic freedom rank  
**hf\_score** Human freedom score  
**hf\_rank** Human freedom rank  
**hf\_quartile** Human freedom quartile

## Details

This dataset contains information from Human Freedom Index reports from 2008-2016.

## Source

Ian Vasquez and Tanja Porcnik, The Human Freedom Index 2018: A Global Measurement of Personal, Civil, and Economic Freedom (Washington: Cato Institute, Fraser Institute, and the Friedrich Naumann Foundation for Freedom, 2018). <https://www.cato.org/sites/cato.org/files/human-freedom-index-files/human-freedom-index-2016.pdf>. <https://www.kaggle.com/gsutters/the-human-freedom-index>.

---

histPlot

*Histogram or hollow histogram*


---

## Description

Create histograms and hollow histograms. This function permits easy color and appearance customization.

## Usage

```
histPlot(
  x,
  col = fadeColor("black", "22"),
  border = "black",
  breaks = "default",
  probability = FALSE,
  hollow = FALSE,
  add = FALSE,
  lty = 2,
  lwd = 1,
  freqTable = FALSE,
  right = TRUE,
  axes = TRUE,
  xlab = NULL,
  ylab = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

## Arguments

x	Numerical vector or a frequency table (matrix) where the first column represents the observed values and the second column the frequencies. See also <code>freqTable</code> argument.
col	Shading of the histogram bins.
border	Color of histogram bin borders.
breaks	A vector for the bin boundaries or an approximate number of bins.

probability	If FALSE, the frequency is plotted. If TRUE, then a probability density.
hollow	If TRUE, a hollow histogram will be created.
add	If TRUE, the histogram is added to the plot.
lty	Line type. Applies only if hollow=TRUE.
lwd	Line width. Applies only if hollow=TRUE.
freqTable	Set to TRUE if x is a frequency table.
right	Set to FALSE to assign values of x that fall on a bin margin to the left bin. Otherwise the ties default to the right bin.
axes	If FALSE, the axes are not plotted.
xlab	Label for the x axis.
ylab	Label for the y axis.
xlim	Limits for the x axis.
ylim	Limits for the y axis.
...	Additional arguments to plot. If add is TRUE, these arguments are ignored.

**Author(s)**

David Diez

**See Also**[boxPlot](#), [dotPlot](#), [densityPlot](#)**Examples**

```

histPlot(tips$tip, main = "Tips")

# overlaid hollow histograms
histPlot(tips$tip[tips$day == "Tuesday"],
  probability = TRUE,
  hollow = TRUE,
  main = "Tips by day"
)
histPlot(tips$tip[tips$day == "Friday"],
  probability = TRUE,
  hollow = TRUE,
  add = TRUE,
  lty = 3,
  border = "red"
)
legend("topright",
  col = c("black", "red"),
  lty = 1:2,
  legend = c("Tuesday", "Friday")
)

```

```
# breaks and colors
histPlot(tips$tip,
  col = fadeColor("yellow", "33"),
  border = "darkblue",
  probability = TRUE,
  breaks = 30,
  lwd = 3
)

# custom breaks
brks <- c(-1, 0, 1, 2, 3, 4, seq(5, 20, 5), 22, 24, 26)
histPlot(tips$tip,
  probability = TRUE,
  breaks = brks,
  col = fadeColor("darkgoldenrod4", "33"),
  xlim = c(0, 26)
)
```

house

*United States House of Representatives historical make-up***Description**

The make-up of the United States House of Representatives every two years since 1789. The last Congress included is the 112th Congress, which completed its term in 2013.

**Usage**

```
house
```

**Format**

A data frame with 112 observations on the following 12 variables.

**congress** The number of that year's Congress  
**year\_start** Starting year  
**year\_end** Ending year  
**seats** Total number of seats  
**p1** Name of the first political party  
**np1** Number of seats held by the first political party  
**p2** Name of the second political party  
**np2** Number of seats held by the second political party  
**other** Other  
**vac** Vacancy  
**del** Delegate  
**res** Resident commissioner

## Source

Party Divisions of the House of Representatives, 1789 to Present. <https://history.house.gov/Institution/Party-Divisions/Party-Divisions>.

## Examples

```
library(dplyr)
library(ggplot2)
library(forcats)

# Examine two-party relationship since 1855
house_since_1855 <- house |>
  filter(year_start >= 1855) |>
  mutate(
    p1_perc = 100 * np1 / seats,
    p2_perc = 100 * np2 / seats,
    era = case_when(
      between(year_start, 1861, 1865) ~ "Civil War",
      between(year_start, 1914, 1918) ~ "World War I",
      between(year_start, 1929, 1939) ~ "Great Depression",
      between(year_start, 1940, 1945) ~ "World War II",
      between(year_start, 1960, 1965) ~ "Vietnam War Start",
      between(year_start, 1965, 1975) ~ "Vietnam War Escalated",
      TRUE ~ NA_character_
    ),
    era = fct_relevel(
      era, "Civil War", "World War I",
      "Great Depression", "World War II",
      "Vietnam War Start", "Vietnam War Escalated"
    )
  )

ggplot(house_since_1855, aes(x = year_start)) +
  geom_rect(aes(
    xmin = year_start, xmax = lead(year_start),
    ymin = -Inf, ymax = Inf, fill = era
  )) +
  geom_line(aes(y = p1_perc, color = "Democrats")) + # Democrats
  geom_line(aes(y = p2_perc, color = "Republicans")) + # Republicans
  scale_fill_brewer(palette = "Pastel1", na.translate = FALSE) +
  scale_color_manual(
    name = "Party",
    values = c("Democrats" = "blue", "Republicans" = "red"),
    labels = c("Democrats", "Republicans")
  ) +
  theme_minimal() +
  ylim(0, 100) +
  labs(x = "Year", y = "Percentage of seats", fill = "Era")
```



---

housing	<i>Simulated dataset on student housing</i>
---------	---

---

**Description**

Each observation represents a simulated rent price for a student.

**Usage**

housing

**Format**

A data frame with 75 observations on the following variable.

**cost** a numeric vector

**Examples**

housing

---

hsb2	<i>High School and Beyond survey</i>
------	--------------------------------------

---

**Description**

Two hundred observations were randomly sampled from the High School and Beyond survey, a survey conducted on high school seniors by the National Center of Education Statistics.

**Usage**

hsb2

**Format**

A data frame with 200 observations and 11 variables.

**id** Student ID.

**gender** Student's gender, with levels female and male.

**race** Student's race, with levels african american, asian, hispanic, and white.

**ses** Socio economic status of student's family, with levels low, middle, and high.

**schtyp** Type of school, with levels public and private.

**prog** Type of program, with levels general, academic, and vocational.

**read** Standardized reading score.

**write** Standardized writing score.  
**math** Standardized math score.  
**science** Standardized science score.  
**socst** Standardized social studies score.

### Source

UCLA Institute for Digital Research & Education - Statistical Consulting.

### Examples

```
library(ggplot2)

ggplot(hsb2, aes(x = read - write, y = ses)) +
  geom_boxplot() +
  labs(
    x = "Difference between reading and writing scores",
    y = "Socio-economic status"
  )
```

---

husbands\_wives

*Great Britain: husband and wife pairs*

---

### Description

The Great Britain Office of Population Census and Surveys once collected data on a random sample of 170 married couples in Britain, recording the age (in years) and heights of the husbands and wives.

### Usage

husbands\_wives

### Format

A data frame with 199 observations on the following 8 variables.

**age\_husband** Age of husband.  
**age\_wife** Age of wife.  
**ht\_husband** Height of husband (mm).  
**ht\_wife** Height of wife (mm).  
**age\_husb\_at\_marriage** Age of husband at the time they married.  
**age\_wife\_at\_marriage** Age of wife at the time they married.  
**years\_married** Number of years married.

**Source**

Hand DJ. 1994. A handbook of small data sets. Chapman & Hall/CRC.

**Examples**

```
library(ggplot2)

ggplot(husbands_wives, aes(x = ht_husband, y = ht_wife)) +
  geom_point()
```

---

immigration

---

*Poll on illegal workers in the US*


---

**Description**

910 randomly sampled registered voters in Tampa, FL were asked if they thought workers who have illegally entered the US should be (i) allowed to keep their jobs and apply for US citizenship, (ii) allowed to keep their jobs as temporary guest workers but not allowed to apply for US citizenship, or (iii) lose their jobs and have to leave the country as well as their political ideology.

**Usage**

```
immigration
```

**Format**

A data frame with 910 observations on the following 2 variables.

**response** a factor with levels Apply for citizenship Guest worker Leave the country Not sure

**political** a factor with levels conservative liberal moderate

**Source**

SurveyUSA, News Poll #18927, data collected Jan 27-29, 2012.

**Examples**

```
immigration
```

---

IMSCOL	<i>Introduction to Modern Statistics (IMS) Colors</i>
--------	---

---

**Description**

These are the core colors used for the Introduction to Modern Statistics textbook. The blue, green, pink, yellow, and red colors are also gray-scaled, meaning no changes are required when printing black and white copies.

**Usage**

IMSCOL

**Format**

A 8-by-13 matrix of 7 colors with four fading scales: blue, green, pink, yellow, red, black, gray, and light gray.

**Examples**

```
plot(1:7, 7:1,
     col = IMSCOL, pch = 19, cex = 6, xlab = "", ylab = "",
     xlim = c(0.5, 7.5), ylim = c(-2.5, 8), axes = FALSE
)
text(1:7, 7:1 + 0.7, paste("IMSCOL[", 1:7, "]", sep = ""), cex = 0.9)
points(1:7, 7:1 - 0.7, col = IMSCOL[, 2], pch = 19, cex = 6)
points(1:7, 7:1 - 1.4, col = IMSCOL[, 3], pch = 19, cex = 6)
points(1:7, 7:1 - 2.1, col = IMSCOL[, 4], pch = 19, cex = 6)
```

---

infmortrate	<i>Infant Mortality Rates, 2012</i>
-------------	-------------------------------------

---

**Description**

This entry gives the number of deaths of infants under one year old in 2012 per 1,000 live births in the same year. This rate is often used as an indicator of the level of health in a country.

**Usage**

infmortrate

**Format**

A data frame with 222 observations on the following 2 variables.

- country** Name of country.
- inf\_mort\_rate** Infant mortality rate per 1,000 live births.

## Details

The data is given in decreasing order of infant mortality rates. There are a few potential outliers.

## Source

CIA World Factbook, <https://www.cia.gov/the-world-factbook/field/infant-mortality-rate/country-comparison>.

## Examples

```
library(ggplot2)

ggplot(infmortrate, aes(x = inf_mort_rate)) +
  geom_histogram(binwidth = 10)

ggplot(infmortrate, aes(x = inf_mort_rate)) +
  geom_density()
```

---

iowa

*iowa*


---

## Description

A data frame containing information about the 2016 US Presidential Election for the state of Iowa.

## Usage

```
iowa
```

## Format

A data frame with 1386 observations on the following 5 variables.

**office** The office that the candidates were running for.

**candidate** President/Vice President pairs who were running for office.

**party** Political part of the candidate.

**county** County in Iowa where the votes were cast.

**votes** Number of votes received by the candidate.

## Examples

```
library(ggplot2)
library(dplyr)

plot_data <- iowa |>
  filter(candidate != "Total") |>
  group_by(candidate) |>
  summarize(total_votes = sum(votes) / 1000)
```

```
ggplot(plot_data, aes(total_votes, candidate)) +
  geom_col() +
  theme_minimal() +
  labs(
    title = "2016 Presidential Election in Iowa",
    subtitle = "Popular vote",
    y = "",
    x = "Number of Votes (in thousands)"
  )
```

---

 ipo

---

*Facebook, Google, and LinkedIn IPO filings*


---

## Description

On Feb 1st, 2011, Facebook Inc. filed an S-1 form with the Securities and Exchange Commission as part of their initial public offering (IPO). This dataset includes the text of that document as well as text from the IPOs of two competing companies: Google and LinkedIn.

## Usage

ipo

## Format

The format is a list of three character vectors. Each vector contains the line-by-line text of the IPO Prospectus of Facebook, Google, and LinkedIn, respectively.

## Details

Each of the three prospectuses is encoded in UTF-8 format and contains some non-word characters related to the layout of the original documents. For analysis on the words, it is recommended that the data be processed with packages such as [tidytext](#). See examples below.

## Source

All IPO prospectuses are available from the U.S. Securities and Exchange Commission: [Facebook](#), [Google](#), [LinkedIn](#).

## References

Zweig, J., 2020. Mark Zuckerberg: CEO For Life?. WSJ.

**Examples**

```

library(tidytext)
library(tibble)
library(dplyr)
library(ggplot2)
library(forcats)

# Analyzing Facebook IPO text

facebook <- tibble(text = ipo$facebook, company = "Facebook")

facebook |>
  unnest_tokens(word, text) |>
  anti_join(stop_words) |>
  count(word, sort = TRUE) |>
  slice_head(n = 20) |>
  ggplot(aes(y = fct_reorder(word, n), x = n, fill = n)) +
  geom_col() +
  labs(
    title = "Top 20 most common words in Facebook IPO",
    x = "Frequency",
    y = "Word"
  )

# Comparisons to Google and LinkedIn IPO texts

google <- tibble(text = ipo$google, company = "Google")
linkedin <- tibble(text = ipo$linkedin, company = "LinkedIn")

ipo_texts <- bind_rows(facebook, google, linkedin)

ipo_texts |>
  unnest_tokens(word, text) |>
  count(company, word, sort = TRUE) |>
  bind_tf_idf(word, company, n) |>
  arrange(desc(tf_idf)) |>
  group_by(company) |>
  slice_max(tf_idf, n = 15) |>
  ungroup() |>
  ggplot(aes(tf_idf, fct_reorder(word, tf_idf), fill = company)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~company, ncol = 3, scales = "free") +
  labs(x = "tf-idf", y = NULL)

```

ipod

*Length of songs on an iPod***Description**

A simulated dataset on lengths of songs on an iPod.

**Usage**

```
ipod
```

**Format**

A data frame with 3000 observations on the following variable.

**song\_length** Length of song (in minutes).

**Source**

Simulated data.

**Examples**

```
library(ggplot2)

ggplot(ipod, aes(x = song_length)) +
  geom_histogram(binwidth = 0.5)
```

---

iran

*iran*


---

**Description**

A data frame containing information about the 2009 Presidential Election in Iran. There were widespread claims of election fraud in this election both internationally and within Iran.

**Usage**

```
iran
```

**Format**

A data frame with 366 observations on the following 9 variables.

**province** Iranian province where votes were cast.

**city** City within province where votes were cast.

**ahmadinejad** Number of votes received by Ahmadinejad.

**rezai** Number of votes received by Rezai.

**karrubi** Number of votes received by Karrubi.

**mousavi** Number of votes received by Mousavi.

**total\_votes\_cast** Total number of votes cast.

**voided\_votes** Number of votes that were not counted.

**legitimate\_votes** Number of votes that were counted.



**Examples**

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(stringr)

plot_data <- iran |>
  summarize(
    ahmadinejad = sum(ahmadinejad) / 1000,
    rezai = sum(rezai) / 1000,
    karrubi = sum(karrubi) / 1000,
    mousavi = sum(mousavi) / 1000
  ) |>
  pivot_longer(
    cols = c(ahmadinejad, rezai, karrubi, mousavi),
    names_to = "candidate",
    values_to = "votes"
  ) |>
  mutate(candidate = str_to_title(candidate))

ggplot(plot_data, aes(votes, candidate)) +
  geom_col() +
  theme_minimal() +
  labs(
    title = "2009 Iranian Presidential Election",
    x = "Number of votes (in thousands)",
    y = ""
  )
```

jury

*Simulated juror dataset***Description**

Simulated dataset of registered voters proportions and representation on juries.

**Usage**

```
jury
```

**Format**

A data frame with 275 observations on the following variable.

**race** a factor with levels black hispanic other white

**Examples**

```
jury
```

---

kobe\_basket

*Kobe Bryant basketball performance*


---

### Description

Data from the five games the Los Angeles Lakers played against the Orlando Magic in the 2009 NBA finals.

### Usage

```
kobe_basket
```

### Format

A data frame with 133 rows and 6 variables:

**vs** A categorical vector, ORL if the Los Angeles Lakers played against Orlando

**game** A numerical vector, game in the 2009 NBA finals

**quarter** A categorical vector, quarter in the game, OT stands for overtime

**time** A character vector, time at which Kobe took a shot

**description** A character vector, description of the shot

**shot** A categorical vector, H if the shot was a hit, M if the shot was a miss

### Details

Each row represents a shot Kobe Bryant took during the five games of the 2009 NBA finals. Kobe Bryant's performance earned him the title of Most Valuable Player and many spectators commented on how he appeared to show a hot hand.

---

labor\_market\_discrimination

*Are Emily and Greg More Employable Than Lakisha and Jamal?*


---

### Description

Original data from the experiment run by Bertrand and Mullainathan (2004).

### Usage

```
labor_market_discrimination
```

**Format**

A tibble with 4870 observations of 63 variables.

**education** Highest education, with levels of 0 = not reported; 1 = high school diploma; 2 = high school graduate; 3 = some college; 4 = college or more.

**n\_jobs** Number of jobs listed on resume.

**years\_exp** Number of years of work experience on the resume.

**honors** Indicator variable for which 1 = resume mentions some honors.

**volunteer** Indicator variable for which 1 = resume mentions some volunteering experience.

**military** Indicator variable for which 1 = resume mentions some military experience.

**emp\_holes** Indicator variable for which 1 = resume mentions some employment holes.

**occup\_specific** 1990 Census Occupation Code. See sources for a key.

**occup\_broad** Occupation broad with levels 1 = executives and managerial occupations, 2 = administrative supervisors, 3 = sales representatives, 4 = sales workers, 5 = secretaries and legal assistants, 6 = clerical occupations

**work\_in\_school** Indicator variable for which 1 = resume mentions some work experience while at school

**email** Indicator variable for which 1 = email address on applicant's resume.

**computer\_skills** Indicator variable for which 1 = resume mentions some computer skills.

**special\_skills** Indicator variable for which 1 = resume mentions some special skills.

**first\_name** Applicant's first name.

**sex** Sex, with levels of 'f' = female; 'm' = male.

**race** Race, with levels of 'b' = black; 'w' = white.

**h** Indicator variable for which 1 = high quality resume.

**l** Indicator variable for which 1 = low quality resume.

**call** Indicator variable for which 1 = applicant was called back.

**city** City, with levels of 'c' = chicago; 'b' = boston.

**kind** Kind, with levels of 'a' = administrative; 's' = sales.

**ad\_id** Employment ad identifier.

**frac\_black** Fraction of blacks in applicant's zip.

**frac\_white** Fraction of whites in applicant's zip.

**l\_med\_hh\_inc** Log median household income in applicant's zip.

**frac\_dropout** Fraction of high-school dropouts in applicant's zip.

**frac\_colp** Fraction of college degree or more in applicant's zip

**l\_inc** Log per capita income in applicant's zip.

**col** Indicator variable for which 1 = applicant has college degree or more.

**exminreq** Minimum experience required, if any (in years when numeric).

**school\_req** Specific education requirement, if any. 'hsg' = high school graduate, 'somcol' = some college, 'colp' = four year degree or higher

**oe** Indicator variable for which 1 = ad mentions employer is 'Equal Opportunity Employer'.

**parent\_sales** Sales of parent company (in millions of US \$).

**parent\_emp** Number of parent company employees.

**branch\_sales** Sales of branch (in millions of US \$).

**branch\_emp** Number of branch employees.

**fed** Indicator variable for which 1 = employer is a federal contractor.

**frac\_black\_emp\_zip** Fraction of blacks in employers's zipcode.

**frac\_white\_emp\_zip** Fraction of whites in employer's zipcode.

**l\_med\_hh\_inc\_emp\_zip** Log median household income in employer's zipcode.

**frac\_dropout\_emp\_zip** Fraction of high-school dropouts in employer's zipcode.

**frac\_colp\_emp\_zip** Fraction of college degree or more in employer's zipcode.

**l\_inc\_emp\_zip** Log per capita income in employer's zipcode.

**manager** Indicator variable for which 1 = executives or managers wanted.

**supervisor** Indicator variable for which 1 = administrative supervisors wanted.

**secretary** Indicator variable for which 1 = secretaries or legal assistants wanted.

**off\_support** Indicator variable for which 1 = clerical workers wanted.

**sales\_rep** Indicator variable for which 1 = sales representative wanted.

**retail\_sales** Indicator variable for which 1 = retail sales worker wanted.

**req** Indicator variable for which 1 = ad mentions any requirement for job.

**exp\_req** Indicator variable for which 1 = ad mentions some experience requirement.

**com\_req** Indicator variable for which 1 = ad mentions some communication skills requirement.

**educ\_req** Indicator variable for which 1 = ad mentions some educational requirement.

**comp\_req** Indicator variable for which 1 = ad mentions some computer skill requirement.

**org\_req** Indicator variable for which 1 = ad mentions some organizational skills requirement.

**manuf** Indicator variable for which 1 = employer industry is manufacturing.

**trans\_com** Indicator variable for which 1 = employer industry is transport or communication.

**bank\_real** Indicator variable for which 1 = employer industry is finance, insurance or real estate.

**trade** Indicator variable for which 1 = employer industry is wholesale or retail trade.

**bus\_service** Indicator variable for which 1 = employer industry is business or personal services.

**oth\_service** Indicator variable for which 1 = employer industry is health, education or social services.

**miss\_ind** Indicator variable for which 1 = employer industry is other or unknown.

**ownership** Ownership status of employer, with levels of 'non-profit'; 'private'; 'public'

## Details

From the summary: "We study race in the labor market by sending fictitious resumes to help-wanted ads in Boston and Chicago newspapers. To manipulate perceived race, resumes are randomly assigned African-American- or White-sounding names. White names receive 50 percent more callbacks for interviews. Callbacks are also more responsive to resume quality for White names than for African-American ones. The racial gap is uniform across occupation, industry, and employer size. We also find little evidence that employers are inferring social class from the names. Differential treatment by race still appears to be prominent in the U. S. labor market."

## Source

Bertrand, Marianne, and Mullainathan, Sendhil. Replication data for: Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. Nashville, TN: American Economic Association [publisher], 2004. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 2019-12-06. doi:10.3886/E116023V1.

[NBER Working Papers](#)

[1990 Census Occupation Codes](#)

Note: The description of the variables follows closely the labels provided in the original dataset, with small edits for clarity.

## Examples

```
library(dplyr)

# Percent callback for typical White names and typical African-American names (table 1, p. 997)

labor_market_discrimination |>
  group_by(race) |>
  summarise(call_back = mean(call))
```

---

lab\_report

*lab\_report*


---

## Description

Acts as a simplified template to common parameters passed to `rmarkdown::html_document()`.

## Usage

```
lab_report(
  highlight = "pygments",
  theme = "spacelab",
  toc = TRUE,
  toc_float = TRUE,
  code_download = TRUE,
  code_folding = "show"
)
```

## Arguments

highlight	Syntax highlighting style. Supported styles include "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", "haddock", and "textmate". Pass NULL to prevent syntax highlighting.
theme	Visual theme ("default", "cerulean", "journal", "flatly", "readable", "spacelab", "united", "cosmo", "lumen", "paper", "sandstone", "simplex", or "yeti"). Pass NULL for no theme (in this case you can use the css parameter to add your own styles).

<code>toc</code>	TRUE to include a table of contents in the output
<code>toc_float</code>	TRUE to float the table of contents to the left of the main document content. Rather than TRUE you may also pass a list of options that control the behavior of the floating table of contents. See the <i>Floating Table of Contents</i> section below for details.
<code>code_download</code>	Embed the Rmd source code within the document and provide a link that can be used by readers to download the code.
<code>code_folding</code>	Enable document readers to toggle the display of R code chunks. Specify "none" to display all code chunks (assuming they were knit with <code>echo = TRUE</code> ). Specify "hide" to hide all R code chunks by default (users can show hidden code chunks either individually or document-wide). Specify "show" to show all R code chunks by default.

---

LAhomes

---

LAhomes

---

## Description

Data collected by Andrew Bray at Reed College on characteristics of LA Homes in 2010.

## Usage

LAhomes

## Format

A data frame with 1594 observations on the following 8 variables.

**city** City where the home is located.

**type** Type of home with levels Condo/Twh - condo or townhouse, SFR - single family residence, and NA

**bed** Number of bedrooms in the home.

**bath** Number of bathrooms in the home.

**garage** Number of cars that can be parked in the garage. Note that a value of 4 refers to 4 or more garage spaces.

**sqft** Squarefootage of the home.

**pool** Indicates if the home has a pool.

**price** Listing price of the home.

## Examples

```
library(ggplot2)

ggplot(LAhomes, aes(sqft, price)) +
  geom_point(alpha = 0.2) +
  theme_minimal() +
  labs(
    title = "Can we predict list price from squarefootage?",
    subtitle = "Homes in the Los Angeles area",
    x = "Square feet",
    y = "List price"
  )
```

---

law\_resume

*Gender, Socioeconomic Class, and Interview Invites*

---

## Description

Resumes were sent out to 316 top law firms in the United States, and there were two randomized characteristics of each resume. First, the gender associated with the resume was randomized by assigning a first name of either James or Julia. Second, the socioeconomic class of the candidate was randomly assigned and represented through five minor changes associated with personal interests and other other minor details (e.g. an extracurricular activity of sailing team vs track and field). The outcome variable was whether the candidate was received an interview.

## Usage

```
law_resume
```

## Format

A data frame with 316 observations on the following 3 variables. Each row represents a resume sent a top law firm for this experiment.

**class** The resume represented irrelevant details suggesting either "low" or "high" socioeconomic class.

**gender** The resume implied the candidate was either "male" or "female".

**outcome** If the candidate received an invitation for an "interview" or "not".

## Source

For a casual overview, see <https://hbr.org/2016/12/research-how-subtle-class-cues-can-backfire-on-your-re>

For the academic paper, see Tilcsik A, Rivera LA. 2016. Class Advantage, Commitment Penalty. The Gendered Effect of Social Class Signals in an Elite Labor Market. American Sociological Review 81:6 p1097-1131. doi:10.1177/0003122416668154.

## Examples

```
tapply(law_resume$outcome == "interview", law_resume[, c("class", "gender")], mean)
m <- glm(I(outcome == "interview") ~ gender * class, data = law_resume, family = binomial)
summary(m)
predict(m, type = "response")
```

---

lecture\_learning

*Lecture Delivery Method and Learning Outcomes*

---

## Description

Data was collected from 276 students in a university psychology course to determine the effect of lecture delivery method on learning. Students were presented a live lecture by the professor on one day and a pre-recorded lecture on a different topic by the same professor on a different day. Survey data was collected during the lectures to determine mind wandering, interest, and motivation. Students were also ultimately asked about the preferred lecture delivery method. Finally, students completed an assessment at the end of the lecture to determine memory recall.

## Usage

lecture\_learning

## Format

A data frame with 552 rows and 8 variables.

**student** Identification number of a specific student. Each identification appears twice because same student heard both lecture delivery methods.

**gender** Gender of student. Recored a binary variable with levels Male and Female in the study.

**method** Delivery method of lecture was either in-person(Live) or pre-recorded(Video).

**mindwander** An indicator of distraction during the lecture. It is a proportion of six mind wandering probes during the lecture when a student answered yes that mind wandering had just occurred.

**memory** An indicator of recall of information provided during the lecture. It is the proportion of correct answers in a six question assessment given at the end of the lecture presentation.

**interest** A Likert scale that gauged student interest level concerning the lecture.

**motivation\_both** After experiencing both lecture delivery methods, students were asked about which method they were most motivated to remain attentive.

**motivation\_single** After a single lecture delivery experience, this Likert scale was used to gauge motivation to remain attentive during the lecture.

## Source

[PLOS One](#)



**Examples**

```

library(dplyr)
library(ggplot2)

# Calculate the average memory test proportion by lecture delivery method
# and gender.
lecture_learning |>
  group_by(method, gender) |>
  summarize(average_memory = mean(memory), count = n(), .groups = "drop")

# Compare visually the differences in memory test proportions by delivery
# method and gender.
ggplot(lecture_learning, aes(x = method, y = memory, fill = gender)) +
  geom_boxplot() +
  theme_minimal() +
  labs(
    title = "Difference in memory test proportions",
    x = "Method",
    y = "Memory",
    fill = "Gender"
  )

# Use a paired t-test to determine whether memory test proportion score
# differed by delivery method. Note that paired t-tests are identical
# to one sample t-test on the difference between the Live and Video methods.
learning_diff <- lecture_learning |>
  tidyr::pivot_wider(id_cols = student, names_from = method, values_from = memory) |>
  mutate(time_diff = Live - Video)
t.test(time_diff ~ 1, data = learning_diff)

# Calculating the proportion of students who were most motivated to remain
# attentive in each delivery method.
lecture_learning |>
  count(motivation_both) |>
  mutate(proportion = n / sum(n))

```

---

lego_population	<i>Population of Lego Sets for Sale between Jan. 1, 2018 and Sept. 11, 2020.</i>
-----------------	--

---

**Description**

Data about Lego Sets for sale. Based on JSDSE article by Anna Peterson and Laura Ziegler Data from their article was scrapped from multiple sources including brickset.com

**Usage**

```
lego_population
```

## Format

A data frame with 1304 rows and 14 variables.

**item\_number** Set Item number

**set\_name** Name of the set.

**theme** Set theme: Duplo, City or Friends.

**pieces** Number of pieces in the set.

**price** Recommended retail price from LEGO.

**amazon\_price** Price of the set at Amazon.

**year** Year that it was produced.

**ages** LEGO's recommended ages of children for the set

**pages** Pages in the instruction booklet.

**minifigures** Number of LEGO people in the data, if unknown "NA" was recorded.

**packaging** Type of packaging: bag, box, etc.

**weight** Weight of the set of LEGOS in pounds and kilograms.

**unique\_pieces** Number of pieces classified as unique in the instruction manual.

**size** Size of the lego pieces: Large if safe for small children and Small for older children.

## Source

Peterson, A. D., & Ziegler, L. (2021). Building a multiple linear regression model with LEGO brick data. *Journal of Statistics and Data Science Education*, 29(3), 1-7. doi:10.1080/26939169.2021.1946450

BrickInstructions.com. (n.d.). Retrieved February 2, 2021 from

Brickset. (n.d.). BRICKSET: Your LEGO® set guide. Retrieved February 2, 2021 from

## Examples

```
library(ggplot2)
library(dplyr)

lego_population |>
  filter(theme == "Friends" | theme == "City") |>
  ggplot(aes(x = pieces, y = amazon_price)) +
  geom_point(alpha = 0.3) +
  labs(
    x = "Pieces in the Set",
    y = "Amazon Price",
    title = "Amazon Price vs Number of Pieces in Lego Sets",
    subtitle = "Friends and City Themes"
  )
```

lego\_sample

*Sample of Lego Sets***Description**

Data about Lego Sets for sale. Based on JSDSE article by Anna Peterson and Laura Ziegler Data from their article was scrapped from multiple sources including brickset.com

**Usage**

lego\_sample

**Format**

A data frame with 75 rows and 15 variables.

**item\_number** Set Item number

**set\_name** Name of the set.

**theme** Set theme: Duplo, City or Friends.

**pieces** Number of pieces in the set.

**price** Recommended retail price from LEGO.

**amazon\_price** Price of the set at Amazon.

**year** Year that it was produced.

**ages** LEGO's recommended ages of children for the set

**pages** Pages in the instruction booklet.

**minifigures** Number of LEGO people in the data, if unknown "NA" was recorded.

**packaging** Type of packaging: bag, box, etc.

**weight** Weight of the set of LEGOS in pounds and kilograms.

**unique\_pieces** Number of pieces classified as unique in the instruction manual.

**size** Size of the lego pieces: Large if safe for small children and Small for older children.

**Source**

Peterson, A. D., & Ziegler, L. (2021). Building a multiple linear regression model with LEGO brick data. *Journal of Statistics and Data Science Education*, 29(3), 1-7. doi:10.1080/26939169.2021.1946450

BrickInstructions.com. (n.d.). Retrieved February 2, 2021 from

Brickset. (n.d.). BRICKSET: Your LEGO® set guide. Retrieved February 2, 2021 from

### Examples

```
library(ggplot2)
library(dplyr)

lego_sample |>
  filter(theme == "Friends" | theme == "City") |>
  ggplot(aes(x = pieces, y = amazon_price)) +
  geom_point(alpha = 0.3) +
  labs(
    x = "Pieces in the Set",
    y = "Amazon Price",
    title = "Amazon Price vs Number of Pieces in Lego Sets",
    subtitle = "Friends and City Themes"
  )
```

---

leg\_mari

*Legalization of Marijuana Support in 2010 California Survey*

---

### Description

In a 2010 Survey USA poll, 70% of the 119 respondents between the ages of 18 and 34 said they would vote in the 2010 general election for Prop 19, which would change California law to legalize marijuana and allow it to be regulated and taxed.

### Usage

```
leg_mari
```

### Format

A data frame with 119 observations on the following variable.

**response** One of two values: oppose or support.

### Source

Survey USA, Election Poll #16804, data collected July 8-11, 2010.

### Examples

```
table(leg_mari)
```

---

life_exp	<i>life_exp</i>
----------	-----------------

---

**Description**

A data frame with 3142 rows and 4 columns. County level data for life expectancy and median income in the United States.

**Usage**

life\_exp

**Format**

A data frame with 3142 observations on the following 4 variables.

- state** Name of the state.
- county** Name of the county.
- expectancy** Life expectancy in the county.
- income** Median income in the county, measured in US \$.

**Examples**

```
library(ggplot2)

# Income V Expectancy
ggplot(life_exp, aes(x = income, y = expectancy)) +
  geom_point(color = openintro::IMSCOL["green", "full"], alpha = 0.2) +
  theme_minimal() +
  labs(
    title = "Is there a relationship between median income and life expectancy?",
    x = "Median income (US $)",
    y = "Life Expectancy (year)"
  )
```

---

linResPlot	<i>Create simple regression plot with residual plot</i>
------------	---

---

**Description**

Create a simple regression plot with residual plot.

**Usage**

```
linResPlot(
  x,
  y,
  axes = FALSE,
  wBox = TRUE,
  wLine = TRUE,
  lCol = "#00000088",
  lty = 1,
  lwd = 1,
  main = "",
  xlab = "",
  ylab = "",
  marRes = NULL,
  col = fadeColor(4, "88"),
  pch = 20,
  cex = 1.5,
  yR = 0.1,
  ylim = NULL,
  subset = NULL,
  ...
)
```

**Arguments**

<code>x</code>	Predictor variable.
<code>y</code>	Outcome variable.
<code>axes</code>	Whether to plot axis labels.
<code>wBox</code>	Whether to plot boxes around each plot.
<code>wLine</code>	Add a regression line.
<code>lCol</code>	Line color.
<code>lty</code>	Line type.
<code>lwd</code>	Line width.
<code>main</code>	Title for the top plot.
<code>xlab</code>	x-label.
<code>ylab</code>	y-label.
<code>marRes</code>	Margin for the residuals plot.
<code>col</code>	Color of the points.
<code>pch</code>	Plotting character of points.
<code>cex</code>	Size of points.
<code>yR</code>	An additional vertical stretch factor on the plot.
<code>ylim</code>	y-limits.
<code>subset</code>	Boolean vector, if wanting a subset of the data.
<code>...</code>	Additional arguments passed to both plots.

**See Also**[makeTube](#)**Examples**

```
# Currently seems broken for this example.
n <- 25
x <- runif(n)
y <- 5 * x + rnorm(n)
myMat <- rbind(matrix(1:2, 2))
myW <- 1
myH <- c(1, 0.45)
par(mar = c(0.35, 0.654, 0.35, 0.654))
layout(myMat, myW, myH)
linResPlot(x, y, col = COL[1, 2])
```

lizard\_habitat

*Field data on lizards observed in their natural habitat***Description**

Data on here lizard was observed and the level of sunlight. The data are collected on *Sceloporus occidentalis* (western fence lizards) by Stephen C. Adolph in 1983 (in desert and mountain sites) and by Dee Asbury in 2002-3 (in valley site).

**Usage**

```
lizard_habitat
```

**Format**

A data frame with 332 observations on the following 2 variables.

**site** Site of lizard observation: desert, mountain, or valley.

**sunlight** Sunlight level at time of observation: sun (lizard was observed perching in full sunlight), partial (lizard was observed perching with part of its body in the sun, part in the shade), shade (lizard was observed perching in the shade).

**Source**

Adolph, S. C. 1990. Influence of behavioral thermoregulation on microhabitat use by two *Sceloporus* lizards. *Ecology* 71: 315-327. Asbury, D.A., and S. C. Adolph. 2007. Behavioral plasticity in an ecological generalist: microhabitat use by western fence lizards. *Evolutionary Ecology Research* 9:801-815.

**Examples**

```
library(ggplot2)

# Frequencies
table(lizard_habitat)

# Stacked bar plots
ggplot(lizard_habitat, aes(y = site, fill = sunlight)) +
  geom_bar(position = "fill") +
  labs(x = "Proportion")
```

lizard\_run

*Lizard speeds***Description**

Data on top speeds measured on a laboratory race track for two species of lizards: Western fence lizard (*Sceloporus occidentalis*) and Sagebrush lizard (*Sceloporus graciosus*).

**Usage**

```
lizard_run
```

**Format**

A data frame with 48 observations on the following 3 variables.

**top\_speed** Top speed of lizard, meters per second.

**common\_name** Common name: Western fence lizard and Sagebrush lizard.

**scientific\_name** Scientific name (Genus and species): *Sceloporus occidentalis* and *Sceloporus graciosus*.

**Source**

Adolph, S. C. 1987. Physiological and behavioral ecology of the lizards *Sceloporus occidentalis* and *Sceloporus graciosus*. Dissertation. University of Washington, Seattle, Washington, USA.

**Examples**

```
library(ggplot2)
library(dplyr)

# Top speed by species
ggplot(lizard_run, aes(x = top_speed, color = common_name, fill = common_name)) +
  geom_density(alpha = 0.5)

# Top speed summary statistics by species
lizard_run |>
  group_by(common_name) |>
```



```
summarise(  
  n      = n(),  
  mean   = mean(top_speed),  
  sd     = sd(top_speed)  
)
```

---

**lmPlot***Linear regression plot with residual plot*

---

**Description**

Plot data, the linear model, and a residual plot simultaneously.

**Usage**

```
lmPlot(  
  x,  
  y,  
  xAxis = 0,  
  yAxis = 4,  
  resAxis = 3,  
  resSymm = TRUE,  
  wBox = TRUE,  
  wLine = TRUE,  
  lCol = "#00000088",  
  lty = 1,  
  lwd = 1,  
  xlab = "",  
  ylab = "",  
  marRes = NULL,  
  col = "#22558888",  
  pch = 20,  
  cex = 1.5,  
  xR = 0.02,  
  yR = 0.1,  
  xlim = NULL,  
  ylim = NULL,  
  subset = NULL,  
  parCustom = FALSE,  
  myHeight = c(1, 0.45),  
  plots = c("both", "mainOnly", "resOnly"),  
  highlight = NULL,  
  hlCol = NULL,  
  hlCex = 1.5,  
  hlPch = 20,  
  na.rm = TRUE,  
  ...  
)
```

**Arguments**

<code>x</code>	The x coordinates of points in the plot.
<code>y</code>	The y coordinates of points in the plot.
<code>xAxis</code>	The maximum number of x axis labels.
<code>yAxis</code>	The maximum number of y axis labels.
<code>resAxis</code>	The maximum number of y axis labels in the residual plot.
<code>resSymm</code>	Boolean determining whether the range of the residual plot should be symmetric about zero.
<code>wBox</code>	Boolean determining whether a box should be added around each plot.
<code>wLine</code>	Boolean determining whether to add a regression line to the plot.
<code>lCol</code>	The color of the regression line to be added.
<code>lty</code>	The line type of the regression line to be added.
<code>lwd</code>	The line width of the regression line to be added.
<code>xlab</code>	A label for the x axis.
<code>ylab</code>	A label for the y axis
<code>marRes</code>	Margin specified for the residuals.
<code>col</code>	Color of points.
<code>pch</code>	Plotting character.
<code>cex</code>	Plotting character size.
<code>xR</code>	Scaling the limits of the x axis. Ignored if <code>xlim</code> specified.
<code>yR</code>	Scaling the limits of the y axis. Ignored if <code>ylim</code> specified.
<code>xlim</code>	Limits for the x axis.
<code>ylim</code>	Limits for the y axis.
<code>subset</code>	A subset of the data to be used for the linear model.
<code>parCustom</code>	If TRUE, then the plotting margins are not modified automatically. This value should also be TRUE if the plots are being placed within a plot of multiple panels.
<code>myHeight</code>	A numerical vector of length 2 representing the ratio of the primary plot to the residual plot, in height.
<code>plots</code>	Not currently utilized.
<code>highlight</code>	Numerical vector specifying particular points to highlight.
<code>hlCol</code>	Color of highlighted points.
<code>hlCex</code>	Size of highlighted points.
<code>hlPch</code>	Plotting characters of highlighted points.
<code>na.rm</code>	Remove cases with NA values.
<code>...</code>	Additional arguments to plot.

**Author(s)**

David Diez

**See Also**[makeTube](#)**Examples**

```
lmPlot(satgpa$sat_sum, satgpa$fy_gpa)

lmPlot(gradestv$tv, gradestv$grades,
  xAxis = 4,
  xlab = "time watching TV", yR = 0.2, highlight = c(1, 15, 20)
)
```

---

loans_full_schema	<i>Loan data from Lending Club</i>
-------------------	------------------------------------

---

**Description**

This dataset represents thousands of loans made through the Lending Club platform, which is a platform that allows individuals to lend to other individuals. Of course, not all loans are created equal. Someone who is a essentially a sure bet to pay back a loan will have an easier time getting a loan with a low interest rate than someone who appears to be riskier. And for people who are very risky? They may not even get a loan offer, or they may not have accepted the loan offer due to a high interest rate. It is important to keep that last part in mind, since this dataset only represents loans actually made, i.e. do not mistake this data for loan applications!

**Usage**

```
loans_full_schema
```

**Format**

A data frame with 10,000 observations on the following 55 variables.

**emp\_title** Job title.

**emp\_length** Number of years in the job, rounded down. If longer than 10 years, then this is represented by the value 10.

**state** Two-letter state code.

**homeownership** The ownership status of the applicant's residence.

**annual\_income** Annual income.

**verified\_income** Type of verification of the applicant's income.

**debt\_to\_income** Debt-to-income ratio.

**annual\_income\_joint** If this is a joint application, then the annual income of the two parties applying.

**verification\_income\_joint** Type of verification of the joint income.

**debt\_to\_income\_joint** Debt-to-income ratio for the two parties.

**delinq\_2y** Delinquencies on lines of credit in the last 2 years.

**months\_since\_last\_delinq** Months since the last delinquency.

**earliest\_credit\_line** Year of the applicant's earliest line of credit

**inquiries\_last\_12m** Inquiries into the applicant's credit during the last 12 months.

**total\_credit\_lines** Total number of credit lines in this applicant's credit history.

**open\_credit\_lines** Number of currently open lines of credit.

**total\_credit\_limit** Total available credit, e.g. if only credit cards, then the total of all the credit limits. This excludes a mortgage.

**total\_credit\_utilized** Total credit balance, excluding a mortgage.

**num\_collections\_last\_12m** Number of collections in the last 12 months. This excludes medical collections.

**num\_historical\_failed\_to\_pay** The number of derogatory public records, which roughly means the number of times the applicant failed to pay.

**months\_since\_90d\_late** Months since the last time the applicant was 90 days late on a payment.

**current\_accounts\_delinq** Number of accounts where the applicant is currently delinquent.

**total\_collection\_amount\_ever** The total amount that the applicant has had against them in collections.

**current\_installment\_accounts** Number of installment accounts, which are (roughly) accounts with a fixed payment amount and period. A typical example might be a 36-month car loan.

**accounts\_opened\_24m** Number of new lines of credit opened in the last 24 months.

**months\_since\_last\_credit\_inquiry** Number of months since the last credit inquiry on this applicant.

**num\_satisfactory\_accounts** Number of satisfactory accounts.

**num\_accounts\_120d\_past\_due** Number of current accounts that are 120 days past due.

**num\_accounts\_30d\_past\_due** Number of current accounts that are 30 days past due.

**num\_active\_debit\_accounts** Number of currently active bank cards.

**total\_debit\_limit** Total of all bank card limits.

**num\_total\_cc\_accounts** Total number of credit card accounts in the applicant's history.

**num\_open\_cc\_accounts** Total number of currently open credit card accounts.

**num\_cc\_carrying\_balance** Number of credit cards that are carrying a balance.

**num\_mort\_accounts** Number of mortgage accounts.

**account\_never\_delinq\_percent** Percent of all lines of credit where the applicant was never delinquent.

**tax\_liens** a numeric vector

**public\_record\_bankrupt** Number of bankruptcies listed in the public record for this applicant.

**loan\_purpose** The category for the purpose of the loan.

**application\_type** The type of application: either individual or joint.

**loan\_amount** The amount of the loan the applicant received.

**term** The number of months of the loan the applicant received.

**interest\_rate** Interest rate of the loan the applicant received.

**installment** Monthly payment for the loan the applicant received.

**grade** Grade associated with the loan.

**sub\_grade** Detailed grade associated with the loan.

**issue\_month** Month the loan was issued.

**loan\_status** Status of the loan.

**initial\_listing\_status** Initial listing status of the loan. (I think this has to do with whether the lender provided the entire loan or if the loan is across multiple lenders.)

**disbursement\_method** Disbursement method of the loan.

**balance** Current balance on the loan.

**paid\_total** Total that has been paid on the loan by the applicant.

**paid\_principal** The difference between the original loan amount and the current balance on the loan.

**paid\_interest** The amount of interest paid so far by the applicant.

**paid\_late\_fees** Late fees paid by the applicant.

### Source

This data comes from Lending Club (<https://www.lendingclub.com/info/statistics.action>), which provides a very large, open set of data on the people who received loans through their platform.

### Examples

```
loans_full_schema
```

---

london_boroughs	<i>London Borough Boundaries</i>
-----------------	----------------------------------

---

### Description

This dataset contains the coordinates of the boundaries of all 32 boroughs of the Greater London area.

### Usage

```
london_boroughs
```

### Format

A data frame with 45341 observations on the following 3 variables.

**borough** Name of the borough.

**x** The "easting" component of the coordinate, see details.

**y** The "northing" component of the coordinate, see details.

## Details

Map data was made available through the Ordnance Survey Open Data initiative. The data use the **National Grid** coordinate system, based upon eastings (x) and northings (y) instead of longitude and latitude.

The name variable covers all 32 boroughs in Greater London: Barking & Dagenham, Barnet, Bexley, Brent, Bromley, Camden, Croydon, Ealing, Enfield, Greenwich, Hackney, Hammersmith & Fulham, Haringey, Harrow, Havering, Hillingdon, Hounslow, Islington, Kensington & Chelsea, Kingston, Lambeth, Lewisham, Merton, Newham, Redbridge, Richmond, Southwark, Sutton, Tower Hamlets, Waltham Forest, Wandsworth, Westminster

## Source

<https://data.london.gov.uk/dataset/ordnance-survey-code-point>

Contains Ordnance Survey data released under the **Open Government License, OGL v2**.

## See Also

`london_murders`

## Examples

```
library(dplyr)
library(ggplot2)

# Calculate number of murders by borough
london_murders_counts <- london_murders |>
  group_by(borough) |>
  add_tally()

london_murders_counts
## Not run:
# Add number of murders to geographic boundary data
london_boroughs_murders <- inner_join(london_boroughs, london_murders_counts, by = "borough")

# Map murders
ggplot(london_boroughs_murders) +
  geom_polygon(aes(x = x, y = y, group = borough, fill = n), colour = "white") +
  scale_fill_distiller(direction = 1) +
  labs(x = "Easting", y = "Northing", fill = "Number of murders")

## End(Not run)
```

---

`london_murders`

*London Murders, 2006-2011*

---

## Description

This dataset contains the victim name, age, and location of every murder recorded in the Greater London area by the Metropolitan Police from January 1, 2006 to September 7, 2011.

## Usage

london\_murders

## Format

A data frame with 838 observations on the following 5 variables.

**forename** First name(s) of the victim.

**age** Age of the victim.

**date** Date of the murder (YYYY-MM-DD).

**year** Year of the murder.

**borough** The London borough in which the murder took place. See the Details section for a list of all the boroughs.

## Details

To visualize this dataset using a map, see the [london\\_boroughs](#) dataset, which contains the latitude and longitude of polygons that define the boundaries of the 32 boroughs of Greater London.

The borough variable covers all 32 boroughs in Greater London: Barking & Dagenham, Barnet, Bexley, Brent, Bromley, Camden, Croydon, Ealing, Enfield, Greenwich, Hackney, Hammersmith & Fulham, Haringey, Harrow, Havering, Hillingdon, Hounslow, Islington, Kensington & Chelsea, Kingston, Lambeth, Lewisham, Merton, Newham, Redbridge, Richmond, Southwark, Sutton, Tower Hamlets, Waltham Forest, Wandsworth, Westminster

## Source

<https://www.theguardian.com/news/datablog/2011/oct/05/murder-london-list#data>

## References

Inspired by [The Guardian Datablog](#).

## Examples

```
library(dplyr)
library(ggplot2)
library(lubridate)

london_murders |>
  mutate(
    day_count = as.numeric(date - ymd("2006-01-01")),
    date_cut = cut(day_count, seq(0, 2160, 90))
  ) |>
  group_by(date_cut) |>
  add_tally() |>
  ggplot(aes(x = date_cut, y = n)) +
  geom_col() +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  labs(x = "Date from 01/2006 - 09/2011", y = "Number of deaths per 90 days")
```

---

loop	<i>Output a message while inside a loop</i>
------	---

---

**Description**

NOTE: `utils::txtProgressBar()` and `utils::setTxtProgressBar()` are better. Output a message while inside a for loop to update the user on progress. This function is useful in tracking progress when the number of iterations is large or the procedures in each iteration take a long time.

**Usage**

```
loop(i, n = NULL, every = 1, extra = NULL)
```

**Arguments**

- `i`                    The index value used in the loop.
- `n`                    The last entry in the loop.
- `every`                The number of loops between messages.
- `extra`                Additional information to print.

**Author(s)**

David Diez

**See Also**

[myPDF](#)

**Examples**

```
for (i in 1:160) {  
  loop(i, 160, 20, paste("iter", i))  
}
```

---

<code>lsegments</code>	<i>Create a Line Segment Plot</i>
------------------------	-----------------------------------

---

**Description**

Creae a simple plot showing a line segment.



**Usage**

```
lsegments(
  x = c(3, 7),
  l = "o",
  r = "c",
  ticks = TRUE,
  labs = 1,
  add = 0,
  ylim = c(-0.75, 0.25)
)
```

**Arguments**

<code>x</code>	The endpoints of the interval. Values larger (smaller) than 999 (-999) will be interpreted as (negative) infinity.
<code>l</code>	Indicate whether the left end point should be open ("o") or closed ("c").
<code>r</code>	Indicate whether the right end point should be open ("o") or closed ("c").
<code>ticks</code>	Indicate whether to show tick marks (TRUE) or not (FALSE).
<code>labs</code>	The position for the point labels. Set to 0 if no labels should be shown.
<code>add</code>	Indicate whether the line segment should be added to an existing plot (TRUE) or a new plot should be created (FALSE).
<code>ylim</code>	A vector of length 2 specifying the vertical plotting limits, which may be useful for fine-tuning plots. The default is <code>c(-0.75, 0.25)</code> .

**Author(s)**

David Diez

**See Also**

[dlsegments](#), [CCP](#), [ArrowLines](#)

**Examples**

```
lsegments(c(2, 7), "o", "c", ylim = c(-0.3, 0.2))

lsegments(c(5, 7), "c", "c", ylim = c(-0.3, 0.2))

lsegments(c(4, 1000), "o", "o", ylim = c(-0.3, 0.2))
```

---

mail\_me*Influence of a Good Mood on Helpfulness*

---

**Description**

This study investigated whether finding a coin influenced a person's likelihood of mailing a sealed but addressed letter that appeared to have been accidentally left in a conspicuous place. Several variables were collected during the experiment, including two randomized variables of whether there was a coin to be found and whether the letter already had a stamp on it.

**Usage**

```
mail_me
```

**Format**

A data frame with 42 observations on the following 4 variables.

**stamped** a factor with levels no yes

**found\_coin** a factor with levels coin no\_coin

**gender** a factor with levels female male

**mailed\_letter** a factor with levels no yes

**Details**

The precise context was in a phone booth (this study is from the 1970s!), where a person who entered a phone booth would find a dime in the phone tray, which would be sufficient to pay for their phone call. There was also a letter next to the phone, which sometimes had a stamp on it.

**Source**

Levin PF, Isen AM. 1975. Studies on the Effect of Feeling Good on Helping. *Sociometry* 31(1), p141-147.

**Examples**

```
table(mail_me)
(x <- table(mail_me[, c("mailed_letter", "found_coin")]))
chisq.test(x)

(x <- table(mail_me[, c("mailed_letter", "stamped")]))
chisq.test(x)

m <- glm(mailed_letter ~ stamped + found_coin + gender,
  data = mail_me,
  family = binomial
)
summary(m)
```

---

major_survey	<i>Survey of Duke students and the area of their major</i>
--------------	--

---

**Description**

Survey of 218 students, collecting information on their GPAs and their academic major.

**Usage**

```
major_survey
```

**Format**

A data frame with 218 observations on the following 2 variables.

**gpa** Grade point average (GPA).

**major** Area of academic major.

**Examples**

```
library(ggplot2)

ggplot(major_survey, aes(x = major, y = gpa)) +
  geom_boxplot()
```

---

makeTube	<i>Regression tube</i>
----------	------------------------

---

**Description**

Produce a linear, quadratic, or nonparametric tube for regression data.

**Usage**

```
makeTube(
  x,
  y,
  Z = 2,
  R = 1,
  col = "#00000022",
  border = "#00000000",
  type = c("lin", "quad", "robust"),
  stDev = c("constant", "linear", "other"),
  length.out = 99,
  bw = "default",
  plotTube = TRUE,
```

```

    addLine = TRUE,
    ...
)

```

### Arguments

x	x coordinates.
y	y coordinates.
Z	Number of standard deviations out from the regression line to extend the tube.
R	Control of how far the tube extends to the left and right.
col	Fill color of the tube.
border	Border color of the tube.
type	The type of model fit to the data. Here 'robust' results in a nonparametric estimate.
stDev	Choices are constant variance ('constant'), the standard deviation of the errors changes linearly ('linear'), or the standard deviation of the errors should be estimated using nonparametric methods ('other').
length.out	The number of observations used to build the regression model. This argument may be increased to increase the smoothing of a quadratic or nonparametric curve.
bw	Bandwidth used if type='robust' or homosk=FALSE.
plotTube	Whether the tube should be plotted.
addLine	Whether the linear model should be plotted.
...	Additional arguments passed to the lines function if addLine=TRUE.

### Value

X	x coordinates for the regression model.
Y	y coordinates for the regression model.
tubeX	x coordinates for the boundary of the tube.
tubeY	y coordinates for the boundary of the tube.

### Author(s)

David Diez

### See Also

[lmPlot](#)

**Examples**

```

# possum example
plot(possum$total_l, possum$head_l)
makeTube(possum$total_l, possum$head_l, 1)
makeTube(possum$total_l, possum$head_l, 2)
makeTube(possum$total_l, possum$head_l, 3)

# grades and TV example
plot(gradestv)
makeTube(gradestv$tv, gradestv$grades, 1.5)
plot(gradestv)
makeTube(gradestv$tv, gradestv$grades, 1.5, stDev = "o")
plot(gradestv)
makeTube(gradestv$tv, gradestv$grades, 1.5, type = "robust")
plot(gradestv)
makeTube(gradestv$tv, gradestv$grades, 1.5, type = "robust", stDev = "o")

# what can go wrong with a basic least squares model
# 1
x <- runif(100)
y <- 25 * x - 20 * x^2 + rnorm(length(x), sd = 1.5)
plot(x, y)
makeTube(x, y, type = "q")
# 2
x <- c(-0.6, -0.46, -0.091, runif(97))
y <- 25 * x + rnorm(length(x))
y[2] <- y[2] + 8
y[1] <- y[1] + 1
plot(x, y, ylim = range(y) + c(-10, 5))
makeTube(x, y)
# 3
x <- runif(100)
y <- 5 * x + rnorm(length(x), sd = x)
plot(x, y)
makeTube(x, y, stDev = "l", bw = 0.03)

```

---

malaria

---

*Malaria Vaccine Trial*


---

**Description**

Volunteer patients were randomized into one of two experiment groups where they would receive an experimental vaccine or a placebo. They were subsequently exposed to a drug-sensitive strain of malaria and observed to see whether they came down with an infection.

**Usage**

```
malaria
```

**Format**

A data frame with 20 observations on the following 2 variables.

**treatment** Whether a person was given the experimental vaccine or a placebo.

**outcome** Whether the person got an infection or no infection.

**Details**

In this study, volunteer patients were randomized into one of two experiment groups: 14 patients received an experimental vaccine or 6 patients received a placebo vaccine. Nineteen weeks later, all 20 patients were exposed to a drug-sensitive malaria virus strain; the motivation of using a drug-sensitive strain of virus here is for ethical considerations, allowing any infections to be treated effectively.

**Source**

Lyke et al. 2017. PfSPZ vaccine induces strain-transcending T cells and durable protection against heterologous controlled human malaria infection. PNAS 114(10):2711-2716. doi:[10.1073/pnas.1615324114](https://doi.org/10.1073/pnas.1615324114).

**Examples**

```
library(dplyr)

# Calculate conditional probabilities of infection after vaccine/placebo
malaria |>
  count(treatment, outcome) |>
  group_by(treatment) |>
  mutate(prop = n / sum(n))

# Fisher's exact test
fisher.test(table(malaria))
```

---

male\_heights

*Sample of 100 male heights*

---

**Description**

Random sample based on Food Commodity Intake Database distribution

**Usage**

```
male_heights
```

**Format**

A data frame with 100 observations on the following variable.

**heights** a numeric vector

## References

What We Eat In America - Food Commodity Intake Database. Available at <https://fcid.foodrisk.org/>.

## Examples

```
male_heights
```

---

male_heights_fcid	<i>Random sample of adult male heights</i>
-------------------	--

---

## Description

This sample is based on data from the USDA Food Commodity Intake Database.

## Usage

```
male_heights_fcid
```

## Format

A data frame with 100 observations on the following variable.

**height\_inch** Height, in inches.

## Source

Simulated based on data from USDA.

## Examples

```
data(male_heights_fcid)
histPlot(male_heights_fcid$height_inch)
```

---

mammals	<i>Sleep in Mammals</i>
---------	-------------------------

---

## Description

This dataset includes data for 39 species of mammals distributed over 13 orders. The data were used for analyzing the relationship between constitutional and ecological factors and sleeping in mammals. Two qualitatively different sleep variables (dreaming and non dreaming) were recorded. Constitutional variables such as life span, body weight, brain weight and gestation time were evaluated. Ecological variables such as severity of predation, safety of sleeping place and overall danger were inferred from field observations in the literature.

**Usage**

mammals

**Format**

A data frame with 62 observations on the following 11 variables.

**species** Species of mammals

**body\_wt** Total body weight of the mammal (in kg)

**brain\_wt** Brain weight of the mammal (in kg)

**non\_dreaming** Number of hours of non dreaming sleep

**dreaming** Number of hours of dreaming sleep

**total\_sleep** Total number of hours of sleep

**life\_span** Life span (in years)

**gestation** Gestation time (in days)

**predation** An index of how likely the mammal is to be preyed upon. 1 = least likely to be preyed upon. 5 = most likely to be preyed upon.

**exposure** An index of the how exposed the mammal is during sleep. 1 = least exposed (e.g., sleeps in a well-protected den). 5 = most exposed.

**danger** An index of how much danger the mammal faces from other animals. This index is based upon Predation and Exposure. 1 = least danger from other animals. 5 = most danger from other animals.

**Source**

<http://www.statsci.org/data/general/sleep.txt>

**References**

T. Allison and D. Cicchetti, "Sleep in mammals: ecological and constitutional correlates," Arch. Hydrobiol, vol. 75, p. 442, 1975.

**Examples**

```
library(ggplot2)

ggplot(mammals, aes(x = log(body_wt), y = log(brain_wt))) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(x = "Log of body weight", y = "Log of brain weight")
```



---

mammogram

*Experiment with Mammogram Randomized*


---

**Description**

An experiment where 89,835 women were randomized to either get a mammogram or a non-mammogram breast screening. The response measured was whether they had died from breast cancer within 25 years.

**Usage**

```
mammogram
```

**Format**

A data frame with 89835 observations on the following 2 variables.

**treatment** a factor with levels control mammogram

**breast\_cancer\_death** a factor with levels no yes

**Source**

Miller AB. 2014. Twenty five year follow-up for breast cancer incidence and mortality of the Canadian National Breast Screening Study: randomised screening trial. *BMJ* 2014;348:g366.

**Examples**

```
table(mammogram)
chisq.test(table(mammogram))
```

---

manhattan

*manhattan*


---

**Description**

A data frame containing data on apartment rentals in Manhattan.

**Usage**

```
manhattan
```

**Format**

A data frame with 20 observations on the following 1 variable.

**rent** Monthly rent for a 1 bedroom apartment listed as "For rent by owner".

**Examples**

```
library(ggplot2)

ggplot(manhattan, aes(rent)) +
  geom_histogram(color = "white", binwidth = 300) +
  theme_minimal() +
  labs(
    title = "Rent in Manhattan",
    subtitle = "1 Bedroom Apartments",
    x = "Rent (in US$)",
    caption = "Source: Craigslist"
  )
```

marathon

*New York City Marathon Times (outdated)***Description**

Marathon times of male and female winners of the New York City Marathon 1970-1999. See [nyc\\_marathon](#) for a more updated dataset. We recommend not using this dataset since the data source has been taken off the web.

**Usage**

```
marathon
```

**Format**

A data frame with 60 observations on the following 3 variables.

**year** Year

**gender** Gender

**time** Running time (in hours)

**Source**

Data source has been removed.

**Examples**

```
library(ggplot2)

ggplot(marathon, aes(x = time)) +
  geom_histogram(binwidth = 0.15)

ggplot(marathon, aes(y = time, x = gender)) +
  geom_boxplot()
```

---

mariokart*Wii Mario Kart auctions from Ebay*

---

**Description**

Auction data from Ebay for the game Mario Kart for the Nintendo Wii. This data was collected in early October 2009.

**Usage**

mariokart

**Format**

A data frame with 143 observations on the following 12 variables. All prices are in US dollars.

**id** Auction ID assigned by Ebay.

**duration** Auction length, in days.

**n\_bids** Number of bids.

**cond** Game condition, either new or used.

**start\_pr** Start price of the auction.

**ship\_pr** Shipping price.

**total\_pr** Total price, which equals the auction price plus the shipping price.

**ship\_sp** Shipping speed or method.

**seller\_rate** The seller's rating on Ebay. This is the number of positive ratings minus the number of negative ratings for the seller.

**stock\_photo** Whether the auction feature photo was a stock photo or not. If the picture was used in many auctions, then it was called a stock photo.

**wheels** Number of Wii wheels included in the auction. These are steering wheel attachments to make it seem as though you are actually driving in the game. When used with the controller, turning the wheel actually causes the character on screen to turn.

**title** The title of the auctions.

**Details**

There are several interesting features in the data. First off, note that there are two outliers in the data. These serve as a nice example of what one should do when encountering an outlier: examine the data point and remove it only if there is a good reason. In these two cases, we can see from the auction titles that they included other items in their auctions besides the game, which justifies removing them from the dataset.

This dataset includes all auctions for a full week in October 2009. Auctions were included in the dataset if they satisfied a number of conditions. (1) They were included in a search for "wii mario kart" on ebay.com, (2) items were in the Video Games > Games > Nintendo Wii section of Ebay, (3) the listing was an auction and not exclusively a "Buy it Now" listing (sellers sometimes offer

an optional higher price for a buyer to end bidding and win the auction immediately, which is an *optional* Buy it Now auction), (4) the item listed was the actual game, (5) the item was being sold from the US, (6) the item had at least one bidder, (7) there were no other items included in the auction with the exception of racing wheels, either generic or brand-name being acceptable, and (8) the auction did not end with a Buy It Now option.

## Source

Ebay.

## Examples

```
library(ggplot2)
library(broom)
library(dplyr)

# Identify outliers
ggplot(mariokart, aes(x = total_pr, y = cond)) +
  geom_boxplot()

# Replot without the outliers
mariokart |>
  filter(total_pr < 80) |>
  ggplot(aes(x = total_pr, y = cond)) +
  geom_boxplot()

# Fit a multiple regression models
mariokart_no <- mariokart |> filter(total_pr < 80)
m1 <- lm(total_pr ~ cond + stock_photo + duration + wheels, data = mariokart_no)
tidy(m1)
m2 <- lm(total_pr ~ cond + stock_photo + wheels, data = mariokart_no)
tidy(m2)
m3 <- lm(total_pr ~ cond + wheels, data = mariokart_no)
tidy(m3)

# Fit diagnostics
aug_m3 <- augment(m3)

ggplot(aug_m3, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted values", y = "Residuals")

ggplot(aug_m3, aes(x = .fitted, y = abs(.resid))) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted values", y = "Absolute value of residuals")

ggplot(aug_m3, aes(x = 1:nrow(aug_m3), y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Order of data collection", y = "Residuals")
```

```

ggplot(aug_m3, aes(x = cond, y = .resid)) +
  geom_boxplot() +
  labs(x = "Condition", y = "Residuals")

ggplot(aug_m3, aes(x = wheels, y = .resid)) +
  geom_point() +
  labs(
    x = "Number of wheels", y = "Residuals",
    title = "Notice curvature"
  )

```

mcu\_films

*Marvel Cinematic Universe films***Description**

A list of Marvel Cinematic Universe films through the Infinity saga. The Infinity saga is a 23 movie storyline spanning from Ironman in 2008 to Endgame in 2019.

**Usage**

mcu\_films

**Format**

A data frame with 23 rows and 7 variables.

**movie** Title of the movie.

**length\_hrs** Length of the movie: hours portion.

**length\_min** Length of the movie: minutes portion.

**release\_date** Date the movie was released in the US.

**opening\_weekend\_us** Box office totals for opening weekend in the US.

**gross\_us** All box office totals in US.

**gross\_world** All box office totals world wide.

**Details**

Box office figures are not adjusted to a specific year. They are from the year the film was released.

**Source**

[Internet Movie Database.](#)

**Examples**

```
library(ggplot2)
library(scales)

ggplot(mcu_films, aes(x = opening_weekend_us, y = gross_us)) +
  geom_point() +
  labs(
    title = "MCU Box Office Totals: Opening weekend vs. all-time",
    x = "Opening weekend totals (USD in millions)",
    y = "All-time totals (USD)"
  ) +
  scale_x_continuous(labels = label_dollar(scale = 1 / 1000000)) +
  scale_y_continuous(labels = label_dollar(scale = 1 / 1000000))
```

midterms\_house

*President's party performance and unemployment rate***Description**

Covers midterm elections.

**Usage**

```
midterms_house
```

**Format**

A data frame with 29 observations on the following 5 variables.

**year** Year.

**potus** The president in office.

**party** President's party: Democrat or Republican.

**unemp** Unemployment rate.

**house\_change** Change in House seats for the President's party.

**Details**

An older version of this data is at [unemploy\\_pres](#).

**Source**

Wikipedia.

**Examples**

```
library(ggplot2)

ggplot(midterms_house, aes(x = unemp, y = house_change)) +
  geom_point()
```

---

migraine

*Migraines and acupuncture*

---

### Description

Experiment involving acupuncture and sham acupuncture (as placebo) in the treatment of migraines.

### Usage

migraine

### Format

A data frame with 89 observations on the following 2 variables.

**group** a factor with levels control treatment

**pain\_free** a factor with levels no yes

### Source

G. Allais et al. Ear acupuncture in the treatment of migraine attacks: a randomized trial on the efficacy of appropriate versus inappropriate acupoints. In: Neurological Sci. 32.1 (2011), pp. 173-175.

### Examples

migraine

---

military

*US Military Demographics*

---

### Description

This dataset contains demographic information on every member of the US armed forces including gender, race, and rank.

### Usage

military

## Format

A data frame with 1,414,593 observations on the following 6 variables.

**grade** The status of the service member as enlisted officer or warrant officer.

**branch** The branch of the armed forces: air force, army, marine corps, navy.

**gender** Whether the service member is female or male.

**race** The race identified by the service member: ami/aln (american indian/alaskan native), asian, black, multi (multi-ethnic), p/i (pacific islander), unk (unknown), or white.

**hisp** Whether a service member identifies with being hispanic (TRUE) or not (FALSE).

**rank** The numeric rank of the service member (higher number indicates higher rank).

## Details

The branches covered by this dataset include the Army, Navy, Air Force, and Marine Corps. Demographic information on the Coast Guard is contained in the original dataset but has not been included here.

## Source

Data provided by the Department of Defense and made available at <https://catalog.data.gov/dataset/personnel-trends-by-gender-race>, retrieved 2012-02-20.

## Examples

```
## Not run:
library(dplyr)
library(ggplot2)
library(forcats)

# Proportion of females in military branches
military |>
  ggplot(aes(x = branch, fill = gender)) +
  geom_bar(position = "fill") +
  labs(
    x = "Branch", y = "Proportion", fill = "Gender",
    title = "Proportion of females in military branches"
  )

# Proportion of army officer females across ranks
military |>
  filter(
    grade == "officer",
    branch == "army"
  ) |>
  ggplot(aes(x = factor(rank), fill = fct_rev(gender))) +
  geom_bar(position = "fill") +
  labs(
    x = "Rank", y = "Proportion", fill = "Gender",
    title = "Proportion of army officer females across ranks"
  )
```



```
## End(Not run)
```

---

mlb

*Salary data for Major League Baseball (2010)*


---

## Description

Salary data for Major League Baseball players in the year 2010.

## Usage

```
mlb
```

## Format

A data frame with 828 observations on the following 4 variables.

**player** Player name

**team** Team

**position** Field position

**salary** Salary (in \$1000s)

## Source

<https://databases.usatoday.com/mlb-salaries/>, retrieved 2011-02-23.

## Examples

```
# ----- Basic Histogram ----- #
hist(mlb$salary / 1000,
     breaks = 15,
     main = "", xlab = "Salary (millions of dollars)", ylab = "",
     axes = FALSE,
     col = "#22558844"
)
axis(1, seq(0, 40, 10))
axis(2, c(0, 500))
axis(2, seq(100, 400, 100), rep("", 4), tcl = -0.2)

# ----- Histogram on Log Scale ----- #
hist(log(mlb$salary / 1000),
     breaks = 15,
     main = "", xlab = "log(Salary)", ylab = "",
     axes = FALSE, col = "#22558844"
)
axis(1) # , seq(0, 40, 10))
```

```
axis(2, seq(0, 300, 100))

# ----- Box plot of log(salary) against position ----- #
boxPlot(log(mlb$salary / 1000), mlb$position, horiz = TRUE, ylab = "")
```

mlbbat10

*Major League Baseball Player Hitting Statistics for 2010***Description**

Major League Baseball Player Hitting Statistics for 2010.

**Usage**

```
mlbbat10
```

**Format**

A data frame with 1199 observations on the following 19 variables.

**name** Player name  
**team** Team abbreviation  
**position** Player position  
**game** Number of games  
**at\_bat** Number of at bats  
**run** Number of runs  
**hit** Number of hits  
**double** Number of doubles  
**triple** Number of triples  
**home\_run** Number of home runs  
**rbi** Number of runs batted in  
**total\_base** Total bases, computed as  $3HR + 23B + 1*2B + H$   
**walk** Number of walks  
**strike\_out** Number of strikeouts  
**stolen\_base** Number of stolen bases  
**caught\_stealing** Number of times caught stealing  
**obp** On base percentage  
**slg** Slugging percentage ( $\text{total\_base} / \text{at\_bat}$ )  
**bat\_avg** Batting average

**Source**

<https://www.mlb.com>, retrieved 2011-04-22.

**Examples**

```

library(ggplot2)
library(dplyr)
library(scales)

mlbbat10_200 <- mlbbat10 |>
  filter(mlbbat10$at_bat > 200)

# On-base percentage across positions
ggplot(mlbbat10_200, aes(x = position, y = obp, fill = position)) +
  geom_boxplot(show.legend = FALSE) +
  scale_y_continuous(labels = label_number(suffix = "%", accuracy = 0.01)) +
  labs(
    title = "On-base percentage across positions",
    y = "On-base percentage across positions",
    x = "Position"
  )

# Batting average across positions
ggplot(mlbbat10_200, aes(x = bat_avg, fill = position)) +
  geom_density(alpha = 0.5) +
  labs(
    title = "Batting average across positions",
    fill = NULL,
    y = "Batting average",
    x = "Position"
  )

# Mean number of home runs across positions
mlbbat10_200 |>
  group_by(position) |>
  summarise(mean_home_run = mean(home_run)) |>
  ggplot(aes(x = position, y = mean_home_run, fill = position)) +
  geom_col(show.legend = FALSE) +
  labs(
    title = "Mean number of home runs across positions",
    y = "Home runs",
    x = "Position"
  )

# Runs batted in across positions
ggplot(mlbbat10_200, aes(x = run, y = obp, fill = position)) +
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Runs batted in across positions",
    y = "Runs",
    x = "Position"
  )

```

**Description**

Batter statistics for 2018 Major League Baseball season.

**Usage**

```
mlb_players_18
```

**Format**

A data frame with 1270 observations on the following 19 variables.

**name** Player name

**team** Team abbreviation

**position** Position abbreviation: 1B = first base, 2B = second base, 3B = third base, C = catcher, CF = center field (outfield), DH = designated hitter, LF = left field (outfield), P = pitcher, RF = right field (outfield), SS = shortstop.

**games** Number of games played.

**AB** At bats.

**R** Runs.

**H** Hits.

**doubles** Doubles.

**triples** Triples.

**HR** Home runs.

**RBI** Runs batted in.

**walks** Walks.

**strike\_outs** Strike outs.

**stolen\_bases** Stolen bases.

**caught\_stealing\_base** Number of times caught stealing a base.

**AVG** Batting average.

**OBP** On-base percentage.

**SLG** Slugging percentage.

**OPS** On-base percentage plus slugging percentage.

**Source**

<https://www.mlb.com/stats>

**See Also**

[mlbbat10](#), [mlb](#)

**Examples**

```
d <- subset(mlb_players_18, !position %in% c("P", "DH") & AB >= 100)
dim(d)

# ----- Per Position, No Further Grouping ----- #
plot(d$OBP ~ as.factor(d$position))
model <- lm(OBP ~ as.factor(position), d)
summary(model)
anova(model)

# ----- Simplified Analysis, Fewer Positions ----- #
pos <- list(
  c("LF", "CF", "RF"),
  c("1B", "2B", "3B", "SS"),
  "C"
)
POS <- c("OF", "IF", "C")
table(d$position)

# ----- On-Base Percentage Across Positions ----- #
out <- c()
gp <- c()
for (i in 1:length(pos)) {
  these <- which(d$position %in% pos[[i]])
  out <- c(out, d$OBP[these])
  gp <- c(gp, rep(POS[i], length(these)))
}
plot(out ~ as.factor(gp))
summary(lm(out ~ as.factor(gp)))
anova(lm(out ~ as.factor(gp)))
```

mlb\_teams

*Major League Baseball Teams Data.***Description**

A subset of data on Major League Baseball teams from Lahman's Baseball Database. The full dataset is available in the [Lahman R package](#).

**Usage**

```
mlb_teams
```

**Format**

A data frame with 2784 rows and 41 variables.

**year** Year of play.

**league\_id** League the team plays in with levels AL (American League) and NL (National League).

**division\_id** Division the team plays in with levels W (west), E (east) and C (central).  
**rank** Team's rank in their division at the end of the regular season.  
**games\_played** Games played.  
**home\_games** Games played at home.  
**wins** Number of games won.  
**losses** Number of games lost.  
**division\_winner** Did the team win their division? Levels of Y (yes) and N (no).  
**wild\_card\_winner** Was the team a wild card winner. Levels of Y (yes) and N (no).  
**league\_winner** Did the team win their league? Levels of Y (yes) and N (no).  
**world\_series\_winner** Did the team win the World Series? Levels of Y (yes) and N (no).  
**runs\_scored** Number of runs scored during the season.  
**at\_bats** Number of at bats during the season.  
**hits** Number of hits during the season. Includes singles, doubles, triples and homeruns.  
**doubles** Number of doubles hit.  
**triples** Number of triples hit.  
**homeruns** Homeruns by batters.  
**walks** Number of walks.  
**strikeouts\_by\_batters** Number of batters struckout.  
**stolen\_bases** Number of stolen bases.  
**caught\_stealing** Number of base runners caught stealing.  
**batters\_hit\_by\_pitch** Number of batters hit by a pitch.  
**sacrifice\_flies** Number of sacrifice flies.  
**opponents\_runs\_scored** Number of runs scored by opponents.  
**earned\_runs\_allowed** Number of earned runs allowed.  
**earned\_run\_average** Earned run average.  
**complete\_games** Number of games where a single pitcher played the entire game.  
**shutouts** Number of shutouts.  
**saves** Number of saves.  
**outs\_pitches** Number of outs pitched for the season (number of innings pitched times 3).  
**hits\_allowed** Number of hits made by opponents.  
**homeruns\_allowed** Number of homeruns hit by opponents.  
**walks\_allowed** Number of opponents who were walked.  
**strikeouts\_by\_pitchers** Number of opponents who were struckout.  
**errors** Number of errors.  
**double\_plays** Number of double plays.  
**fielding\_percentage** Teams fielding percentage.  
**team\_name** Full name of team.  
**ball\_park** Home ballpark name.  
**home\_attendance** Home attendance total.

**Source**

Lahmans Baseball Database

**Examples**

```
library(dplyr)

# List the World Series winning teams for each year
mlb_teams |>
  filter(world_series_winner == "Y") |>
  select(year, team_name, ball_park)

# List the teams with their average number of wins and losses
mlb_teams |>
  group_by(team_name) |>
  summarize(mean_wins = mean(wins), mean_losses = mean(losses)) |>
  arrange((team_name))
```

---

mn\_police\_use\_of\_force

*Minneapolis police use of force data.*

---

**Description**

From Minneapolis, data from 2016 through August 2021

**Usage**

```
mn_police_use_of_force
```

**Format**

A data frame with 12925 rows and 13 variables.

**response\_datetime** DateTime of police response.

**problem** Problem that required police response.

**is\_911\_call** Whether response was initiated by call to 911.

**primary\_offense** Offense of subject.

**subject\_injury** Whether subject was injured Yes/No/null.

**force\_type** Type of police force used.

**force\_type\_action** Detail of police force used.

**race** Race of subject.

**sex** Gender of subject.

**age** Age of subject.

**type\_resistance** Resistance to police by subject.

**precinct** Precinct where response occurred.

**neighborhood** Neighborhood where response occurred.

**Source**[Minneapolis](#)**Examples**

```
library(dplyr)
library(ggplot2)

# List percent of total for each race
mn_police_use_of_force |>
  count(race) |>
  mutate(percent = round(n / sum(n) * 100, 2)) |>
  arrange(desc(percent))

# Display use of force count by three races
race_sub <- c("Asian", "White", "Black")
ggplot(
  mn_police_use_of_force |> filter(race %in% race_sub),
  aes(force_type, ..count..)
) +
  geom_point(stat = "count", size = 4) +
  coord_flip() +
  facet_grid(race ~ .) +
  labs(
    x = "Force Type",
    y = "Number of Incidents"
  )
```

MosaicPlot

*Custom Mosaic Plot***Description**

Plot a mosaic plot custom built for a particular figure.

**Usage**

```
MosaicPlot(
  formula,
  data,
  col = "#00000022",
  border = 1,
  dir = c("v", "h"),
  off = 0.01,
  cex.axis = 0.7,
  col.dir = "v",
  flip = c("v"),
  ...
)
```



**Arguments**

<code>formula</code>	Formula describing the variable relationship.
<code>data</code>	Data frame for the variables, optional.
<code>col</code>	Colors for plotting.
<code>border</code>	Ignored.
<code>dir</code>	Ignored.
<code>off</code>	Fraction of white space between each box in the plot.
<code>cex.axis</code>	Axis label size.
<code>col.dir</code>	Direction to lay out colors.
<code>flip</code>	Whether to flip the ordering of the vertical ("v") and/or horizontal ("h") ordering in the plot.
<code>...</code>	Ignored.

**Author(s)**

David Diez

**Examples**

```
data(email)
data(COL)
email$spam <- ifelse(email$spam == 0, "not\nspam", "spam")
MosaicPlot(number ~ spam, email, col = COL[1:3], off = 0.02)
```

---

*movies*

*movies*

---

**Description**

A dataset with information about movies released in 2003.

**Usage**

```
movies
```

**Format**

A data frame with 140 observations on the following 5 variables.

**movie** Title of the movie.

**genre** Genre of the movie.

**score** Critics score of the movie on a 0 to 100 scale.

**rating** MPAA rating of the film.

**box\_office** Millions of dollars earned at the box office in the US and Canada.

**Source**

Investigating Statistical Concepts, Applications and Methods

**Examples**

```
library(ggplot2)

ggplot(movies, aes(score, box_office, color = genre)) +
  geom_point() +
  theme_minimal() +
  labs(
    title = "Does a critic score predict box office earnings?",
    x = "Critic rating",
    y = "Box office earnings (millions US$",
    color = "Genre"
  )
```

---

mtl

---

*Medial temporal lobe (MTL) and other data for 26 participants*


---

**Description**

The data are from a convenience sample of 25 women and 10 men who were middle-aged or older. The purpose of the study was to understand the relationship between sedentary behavior and thickness of the medial temporal lobe (MTL) in the brain.

**Usage**

```
mtl
```

**Format**

A data frame with 35 observations on the following 23 variables.

**subject** ID for the individual.

**sex** Gender, which takes values F (female) or M (male).

**ethnic** Ethnicity, simplified to Caucasian and Other.

**educ** Years of educational.

**e4grp** APOE-4 status, taking a value of E4 or Non-E4.

**age** Age, in years.

**mmse** Score from the Mini-Mental State Examination, which is a global cognition evaluation.

**ham\_a** Score on the Hamilton Rating Scale for anxiety.

**ham\_d** Score on the Hamilton Rating Scale for depression.

**dig\_sym** We (the authors of this R package) are unsure as to the meaning of this variable.

**delay\_vp** We (the authors of this R package) are unsure as to the meaning of this variable.

**bfr\_selective\_reminding\_delayed** We (the authors of this R package) are unsure as to the meaning of this variable.

**sitting** Self-reported time sitting per day, averaged to the nearest hour.

**met\_minwk** Metabolic equivalent units score (activity level). A score of 0 means "no activity" while 3000 is considered "high activity".

**ipa\_qgrp** Classification of METminwk into Low or High.

**aca1** Thickness of the CA1 subregion of the MTL.

**aca23dg** Thickness of the CA23DG subregion of the MTL.

**ae\_cort** Thickness of a subregion of the MTL.

**a\_fusi\_cort** Thickness of the fusiform gyrus subregion of the MTL.

**a\_ph\_cort** Thickness of the perirhinal cortex subregion of the MTL.

**a\_pe\_cort** Thickness of the entorhinal cortex subregion of the MTL.

**asubic** Thickness of the subiculum subregion of the MTL.

**total** Total MTL thickness.

## Source

Siddarth P, Burggren AC, Eyre HA, Small GW, Merrill DA. 2018. Sedentary behavior associated with reduced medial temporal lobe thickness in middle-aged and older adults. PLoS ONE 13(4): e0195549. doi:10.1371/journal.pone.0195549.

Thank you to Professor Silas Bergen of Winona State University for pointing us to this dataset!

## References

A New York Times article references this study. <https://www.nytimes.com/2018/04/19/opinion/standing-up-at-your-desk-could-make-you-smarter.html>

## Examples

```
library(ggplot2)

ggplot(mtl, aes(x = ipa_qgrp, y = met_minwk)) +
  geom_boxplot()
```

---

murders

*Data for 20 metropolitan areas*


---

## Description

Population, percent in poverty, percent unemployment, and murder rate.

## Usage

```
murders
```

**Format**

A data frame with 20 metropolitan areas on the following 4 variables.

**population** Population.

**perc\_pov** Percent in poverty.

**perc\_unemp** Percent unemployed.

**annual\_murders\_per\_mil** Number of murders per year per million people.

**Source**

We do not have provenance for these data hence recommend not using them for analysis.

**Examples**

```
library(ggplot2)

ggplot(murders, aes(x = perc_pov, y = annual_murders_per_mil)) +
  geom_point() +
  labs(
    x = "Percent in poverty",
    y = "Number of murders per year per million people"
  )
```

---

myPDF

*Custom PDF function*

---

**Description**

A similar function to pdf and png, except that different defaults are provided, including for the plotting parameters.

**Usage**

```
myPDF(
  fileName,
  width = 5,
  height = 3,
  mar = c(3.9, 3.9, 1, 1),
  mgp = c(2.8, 0.55, 0),
  las = 1,
  tcl = -0.3,
  ...
)
```

**Arguments**

fileName	File name for the image to be output. The name should end in .pdf.
width	The width of the image file (inches). Default: 5.
height	The height of the image file (inches). Default: 3.
mar	Plotting margins. To change, input a numerical vector of length 4.
mgp	Margin graphing parameters. To change, input a numerical vector of length 3. The first argument specifies where x and y labels are placed; the second specifies the axis labels are placed; and the third specifies how far to pull the entire axis from the plot.
las	Orientation of axis labels. Input 0 for the default.
tcl	The tick mark length as a proportion of text height. The default is -0.5.
...	Additional arguments to par.

**Author(s)**

David Diez

**See Also**[edaPlot](#)**Examples**

```
# save a plot to a PDF
# myPDF("myPlot.pdf")
histPlot(mariokart$total_pr)
# dev.off()

# save a plot to a PNG
# myPNG("myPlot.png")
histPlot(mariokart$total_pr)
# dev.off()
```

nba\_finals

*NBA Finals History***Description**

This dataset contains information about the teams who played in the NBA Finals from 1950 - 2022.

**Usage**

```
nba_finals
```

**Format**

A data frame with 73 rows and 9 variables:

**year** The year in which the Finals took place.

**winner** The team who won the series.

**western\_wins** Number of series wins by the Western Conference Champions.

**eastern\_wins** Number of series wins by the Eastern Conference Champions.

**western\_champions** Team that won the Western Conference title and played in the Finals.

**eastern\_champions** Team that won the Eastern Conference title and played in the Finals.

**western\_coach** Coach of the Western Conference champions.

**eastern\_coach** Coach of the Eastern Conference champions.

**home\_court** Which conference held home court advantage for the series.

**Source**

[Wikipedia: List of NBA Champions](#)

**Examples**

```
library(dplyr)
library(ggplot2)
library(tidyr)

# Top 5 Appearing Coaches
nba_finals |>
  pivot_longer(
    cols = c("western_coach", "eastern_coach"),
    names_to = "conference", values_to = "coach"
  ) |>
  count(coach, sort = TRUE) |>
  slice_head(n = 5)

# Top 5 Winning Coaches
nba_finals |>
  mutate(
    winning_coach = case_when(
      western_wins == 4 ~ western_coach,
      eastern_wins == 4 ~ eastern_coach
    )
  ) |>
  count(winning_coach, sort = TRUE) |>
  slice_head(n = 5)
```

---

nba_finals_teams	<i>NBA Finals Team Summary</i>
------------------	--------------------------------

---

**Description**

A dataset with individual team summaries for the NBA Finals series from 1950 to 2022. To win the Finals, a team must win 4 games. The maximum number of games in a series is 7.

**Usage**

```
nba_finals_teams
```

**Format**

A data frame with 33 rows and 7 variables:

**team** Team name.  
**win** Number of NBA Championships won.  
**loss** Number of NBA Championships lost.  
**apps** Number of NBA Finals appearances.  
**pct** Win percentage.  
**years\_won** Years in which the team won a Championship.  
**years\_lost** Years in which the team lost a Championship.

**Details**

Notes:

1. The Chicago Stags folded in 1950, the Washington Capitols in 1951 and the Baltimore Bullets in 1954.
2. This list uses current team names. For example, the Seattle SuperSonics are not on the list as that team moved and became the Oklahoma City Thunder.

**Source**

[List of NBA Champions.](#)

**Examples**

```
library(ggplot2)
library(dplyr)
library(openintro)

teams_with_apps <- nba_finals_teams |>
  filter(apps != 0)

ggplot(teams_with_apps, aes(x = win)) +
```

```
geom_histogram(binwidth = 2) +  
labs(  
  title = "Number of NBA Finals series wins",  
  x = "Number of wins",  
  y = "Number of teams"  
)  
  
ggplot(teams_with_apps, aes(x = apps, y = win)) +  
  geom_point(alpha = 0.3) +  
  labs(  
    title = "Can we predict how many NBA Championships a  
team has based on the number of appearances?",  
    x = "Number of NBA Finals appearances",  
    y = "Number of NBA Finals series wins"  
  )
```

---

nba\_heights

*NBA Player heights from 2008-9*

---

### Description

Heights of all NBA players from the 2008-9 season.

### Usage

```
nba_heights
```

### Format

A data frame with 435 observations (players) on the following 4 variables.

**last\_name** Last name.

**first\_name** First name.

**h\_meters** Height, in meters.

**h\_in** Height, in inches.

### Source

Collected from **NBA**.

### Examples

```
qqnorm(nba_heights$h_meters)
```



---

nba_players_19	<i>NBA Players for the 2018-2019 season</i>
----------------	---

---

**Description**

Summary information from the NBA players for the 2018-2019 season.

**Usage**

```
nba_players_19
```

**Format**

A data frame with 494 observations on the following 7 variables.

**first\_name** First name.

**last\_name** Last name.

**team** Team name

**team\_abbr** 3-letter team abbreviation.

**position** Player position.

**number** Jersey number.

**height** Height, in inches.

**Source**

<https://www.nba.com/players>

**Examples**

```
hist(nba_players_19$height, 20)
table(nba_players_19$team)
```

---

ncbirths	<i>North Carolina births, 1000 cases</i>
----------	--

---

**Description**

In 2004, the state of North Carolina released to the public a large dataset containing information on births recorded in this state. This dataset has been of interest to medical researchers who are studying the relation between habits and practices of expectant mothers and the birth of their children. This is a random sample of 1,000 cases from this dataset.

**Usage**

```
ncbirths
```

**Format**

A data frame with 1000 observations on the following 13 variables.

**fage** Father's age in years.

**mage** Mother's age in years.

**mature** Maturity status of mother.

**weeks** Length of pregnancy in weeks.

**premie** Whether the birth was classified as premature (premie) or full-term.

**visits** Number of hospital visits during pregnancy.

**gained** Weight gained by mother during pregnancy in pounds.

**weight** Weight of the baby at birth in pounds.

**lowbirthweight** Whether baby was classified as low birthweight (low) or not (not low).

**gender** Gender of the baby, female or male.

**habit** Status of the mother as a nonsmoker or a smoker.

**marital** Whether mother is married or not married at birth.

**whitemom** Whether mom is white or not white.

**See Also**

We do not have ideal provenance for these data. For a better documented and more recent dataset on a similar topic with similar variables, see [births14](#).

**Examples**

```
library(ggplot2)

ggplot(ncbirths, aes(x = habit, y = weight)) +
  geom_boxplot() +
  labs(x = "Smoking status of mother", y = "Birth weight of baby (in lbs)")

ggplot(ncbirths, aes(x = whitemom, y = visits)) +
  geom_boxplot() +
  labs(x = "Mother's race", y = "Number of doctor visits during pregnancy")

ggplot(ncbirths, aes(x = mature, y = gained)) +
  geom_boxplot() +
  labs(x = "Mother's age category", y = "Weight gained during pregnancy")
```

---

normTail	<i>Normal distribution tails</i>
----------	----------------------------------

---

**Description**

Produce a normal (or t) distribution and shaded tail.

**Usage**

```
normTail(  
  m = 0,  
  s = 1,  
  L = NULL,  
  U = NULL,  
  M = NULL,  
  df = 1000,  
  curveColor = 1,  
  border = 1,  
  col = "#CCCCCC",  
  xlim = NULL,  
  ylim = NULL,  
  xlab = "",  
  ylab = "",  
  digits = 2,  
  axes = 1,  
  detail = 999,  
  xLab = c("number", "symbol"),  
  cex.axis = 1,  
  xAxisIncr = 1,  
  add = FALSE,  
  ...  
)
```

**Arguments**

m	Numerical value for the distribution mean.
s	Numerical value for the distribution standard deviation.
L	Numerical value representing the cutoff for a shaded lower tail.
U	Numerical value representing the cutoff for a shaded upper tail.
M	Numerical value representing the cutoff for a shaded central region.
df	Numerical value describing the degrees of freedom. Default is 1000, which results in a nearly normal distribution. Small values may be useful to emphasize small tails.
curveColor	The color for the distribution curve.
border	The color for the border of the shaded area.

col	The color for filling the shaded area.
xlim	Limits for the x axis.
ylim	Limits for the y axis.
xlab	A title for the x axis.
ylab	A title for the y axis.
digits	The maximum number of digits past the decimal to use in axes values.
axes	A numeric value denoting whether to draw both axes (3), only the vertical axes (2), only the horizontal axes (1, the default), or no axes (0).
detail	A number describing the number of points to use in drawing the normal curve. Smaller values correspond to a less smooth curve but reduced memory usage in the final file.
xLab	If "number", then the axis is drawn at the mean, and every standard deviation out until the third standard deviation. If "symbol", then Greek letters are used for standard deviations from three standard deviations from the mean.
cex.axis	Numerical value controlling the size of the axis labels.
xAxisIncr	A number describing how often axis labels are placed, scaled by standard deviations. This argument is ignored if xLab = "symbol".
add	Boolean indicating whether to add this normal curve to the existing plot.
...	Additional arguments to plot.

**Author(s)**

David Diez

**See Also**[buildAxis](#)**Examples**

```

normTail(3, 2, 5)
normTail(3, 2, 1, xLab = "symbol")
normTail(3, 2, M = 1:2, xLab = "symbol", cex.axis = 0.8)
normTail(3, 2, U = 5, axes = FALSE)
normTail(L = -1, U = 2, M = c(0, 1), axes = 3, xAxisIncr = 2)
normTail(
  L = -1, U = 2, M = c(0, 1),
  xLab = "symbol", cex.axis = 0.8, xAxisIncr = 2
)

```

---

nuclear_survey	<i>Nuclear Arms Reduction Survey</i>
----------------	--------------------------------------

---

**Description**

A simple random sample of 1,028 US adults in March 2013 found that 56\ support nuclear arms reduction.

**Usage**

```
nuclear_survey
```

**Format**

A data frame with 1028 observations on the following variable.

**arms\_reduction** Responses of favor or against.

**Source**

Gallup report: In U.S., 56 percent Favor U.S.-Russian Nuclear Arms Reductions. Available at <https://news.gallup.com/poll/161198/favor-russian-nuclear-arms-reductions.aspx>.

**Examples**

```
table(nuclear_survey)
```

---

nyc	<i>nyc</i>
-----	------------

---

**Description**

Zagat is a public survey where anyone can provide scores to a restaurant. The scores from the general public are then gathered to produce ratings. This dataset contains a list of 168 NYC restaurants and their Zagat Ratings.

**Usage**

```
nyc
```

**Format**

A data frame with 168 observations on the following 6 variables.

**restaurant** Name of the restaurant.

**price** Price of a meal for two, with drinks, in US \$.

**food** Zagat rating for food.

**decor** Zagat rating for decor.

**service** Zagat rating for service.

**east** Indicator variable for location of the restaurant. 0 = west of 5th Avenue, 1 = east of 5th Avenue

**Details**

For each category the scales are as follows:

0 - 9: poor to fair 10 - 15: fair to good 16 - 19: good to very good 20 - 25: very good to excellent

25 - 30: extraordinary to perfection

**Examples**

```
library(dplyr)
library(ggplot2)

location_labs <- c("West", "East")
names(location_labs) <- c(0, 1)

ggplot(nyc, mapping = aes(x = price, group = east, fill = east)) +
  geom_boxplot(alpha = 0.5) +
  facet_grid(east ~ ., labeller = labeller(east = location_labs)) +
  labs(
    title = "Is food more expensive east of 5th Avenue?",
    x = "Price (US$)"
  ) +
  guides(fill = "none") +
  theme_minimal() +
  theme(axis.text.y = element_blank())
```

---

nycflights

*Flights data*


---

**Description**

On-time data for a random sample of flights that departed NYC (i.e. JFK, LGA or EWR) in 2013.

**Usage**

```
nycflights
```

**Format**

A `tbl_df` with 32,735 rows and 16 variables:

**year,month,day** Date of departure.

**dep\_time,arr\_time** Departure and arrival times, local tz.

**dep\_delay,arr\_delay** Departure and arrival delays, in minutes. Negative times represent early departures/arrivals.

**hour,minute** Time of departure broken in to hour and minutes.

**carrier** Two letter carrier abbreviation. See `airlines` in the `nycflights13` package for more information or google the airline code.

**tailnum** Plane tail number.

**flight** Flight number.

**origin,dest** Origin and destination. See `airports` in the `nycflights13` package for more information or google airport the code.

**air\_time** Amount of time spent in the air.

**distance** Distance flown.

**Source**

Hadley Wickham (2014). `nycflights13`: Data about flights departing NYC in 2013. R package version 0.1.

**Examples**

```
library(dplyr)

# Longest departure delays
nycflights |>
  select(flight, origin, dest, dep_delay, arr_delay) |>
  arrange(desc(dep_delay))

# Longest arrival delays
nycflights |>
  select(flight, origin, dest, dep_delay, arr_delay) |>
  arrange(desc(arr_delay))
```

---

nyc\_marathon

*New York City Marathon Times*


---

**Description**

Marathon times of runners in the Men and Women divisions of the New York City Marathon, 1970 - 2023.

**Usage**

```
nyc_marathon
```

**Format**

A data frame with 108 observations on the following 7 variables.

**year** Year of marathom.

**name** Name of winner.

**country** Country of winner.

**time** Running time (HH:MM:SS).

**time\_hrs** Running time (in hours).

**division** Division: Men or Women.

**note** Note about the race or the winning time.

**Source**

Wikipedia, [List of winners of the New York City Marathon](#). Retrieved 6 November, 2023.

**Examples**

```
library(ggplot2)

ggplot(nyc_marathon, aes(x = year, y = time_hrs, color = division, shape = division)) +
  geom_point()
```

---

offshore_drilling	<i>California poll on drilling off the California coast</i>
-------------------	---

---

**Description**

A 2010 survey asking a randomly sample of registered voters in California for their position on drilling for oil and natural gas off the Coast of California.

**Usage**

```
offshore_drilling
```

**Format**

A data frame with 827 observations on the following 2 variables.

**position** a factor with levels do not know oppose support

**college\_grad** a factor with levels no yes



**Source**

Survey USA, Election Poll #16804, data collected July 8-11, 2010.

**Examples**

```
offshore_drilling
```

---

openintro_colors	<i>OpenIntro colors</i>
------------------	-------------------------

---

**Description**

A character string of full colors from `IMSCOL[,1]`

**Usage**

```
openintro_colors
```

**Format**

A named character string with 9 elements: "blue", "green", "pink", "yellow", "red", "black", "gray", "lgray"

**Examples**

```
openintro_colors
openintro_colors["blue"]
```

---

openintro_cols	<i>Function to extract OpenIntro IMS colors as hex codes</i>
----------------	--

---

**Description**

Uses full colors from [IMSCOL](#)

**Usage**

```
openintro_cols(...)
```

**Arguments**

... Character names of [openintro\\_colors](#)

**Examples**

```
openintro_cols("blue")
openintro_cols("red")
```

---

openintro_pal	<i>Return function to interpolate an OpenIntro IMS color palette</i>
---------------	--

---

**Description**

Not exported

**Usage**

```
openintro_pal(palette = "main", reverse = FALSE, ...)
```

**Arguments**

palette	Character name of palette in <a href="#">openintro_palettes</a>
reverse	Boolean indicating whether the palette should be reversed
...	Additional arguments to pass to <a href="#">grDevices::colorRampPalette()</a>

---

openintro_palettes	<i>OpenIntro palettes</i>
--------------------	---------------------------

---

**Description**

A list with OpenIntro color palettes

**Usage**

```
openintro_palettes
```

**Format**

A list with 8 color palettes: main, two, three, four, five, cool, hot, gray

**Examples**

```
openintro_palettes  
  
openintro_palettes$main  
openintro_palettes$three  
openintro_palettes$cool  
openintro_palettes$hot
```

---

opportunity_cost	<i>Opportunity cost of purchases</i>
------------------	--------------------------------------

---

### Description

In a study on opportunity cost, 150 students were given the following statement: "Imagine that you have been saving some extra money on the side to make some purchases, and on your most recent visit to the video store you come across a special sale on a new video. This video is one with your favorite actor or actress, and your favorite type of movie (such as a comedy, drama, thriller, etc.). This particular video that you are considering is one you have been thinking about buying for a long time. It is available for a special sale price of \$14.99. What would you do in this situation? Please circle one of the options below." Half of the students were given the following two options: (A) Buy this entertaining video. (B) Not buy this entertaining video. The other half were given the following two options (note the modified option B): (A) Buy this entertaining video. (B) Not buy this entertaining video. Keep the \$14.99 for other purchases. The results of this study are in this dataset.

### Usage

```
opportunity_cost
```

### Format

A data frame with 150 observations on the following 2 variables.

**group** a factor with levels control and treatment

**decision** a factor with levels buy video and not buy video

### Source

Frederick S, Novemsky N, Wang J, Dhar R, Nowlis S. 2009. Opportunity Cost Neglect. *Journal of Consumer Research* 36: 553-561.

### Examples

```
library(ggplot2)

table(opportunity_cost)

ggplot(opportunity_cost, aes(y = group, fill = decision)) +
  geom_bar(position = "fill")
```

orings

*1986 Challenger disaster and O-rings***Description**

On January 28, 1986, a routine launch was anticipated for the Challenger space shuttle. Seventy-three seconds into the flight, disaster happened: the shuttle broke apart, killing all seven crew members on board. An investigation into the cause of the disaster focused on a critical seal called an O-ring, and it is believed that damage to these O-rings during a shuttle launch may be related to the ambient temperature during the launch. The table below summarizes observational data on O-rings for 23 shuttle missions, where the mission order is based on the temperature at the time of the launch.

**Usage**

orings

**Format**

A data frame with 23 observations on the following 4 variables.

**mission** Shuttle mission number.

**temperature** Temperature, in Fahrenheit.

**damaged** Number of damaged O-rings (out of 6).

**undamaged** Number of undamaged O-rings (out of 6).

**Source**

<https://archive.ics.uci.edu/dataset/92/challenger+usa+space+shuttle+o+ring>

**Examples**

```
library(dplyr)
library(forcats)
library(tidyr)
library(broom)

# This is a wide data frame. You can convert it to a long
# data frame to predict probability of O-ring damage based
# on temperature using logistic regression.

orings_long <- orings |>
  pivot_longer(cols = c(damaged, undamaged), names_to = "outcome", values_to = "n") |>
  uncount(n) |>
  mutate(outcome = fct_relevel(outcome, "undamaged", "damaged"))

orings_mod <- glm(outcome ~ temperature, data = orings_long, family = "binomial")
tidy(orings_mod)
```

---

oscars*Oscar winners, 1929 to 2018*

---

**Description**

Best actor and actress Oscar winners from 1929 to 2018

**Usage**

oscars

**Format**

A data frame with 182 observations on the following 10 variables.

**oscar\_no** Oscar ceremony number.

**oscar\_yr** Year the Oscar ceremony was held.

**award** Best actress or Best actor.

**name** Name of winning actor or actress.

**movie** Name of movie actor or actress got the Oscar for.

**age** Age at which the actor or actress won the Oscar.

**birth\_pl** US State where the actor or actress was born, country if foreign.

**birth\_date** Birth date of actor or actress.

**birth\_mo** Birth month of actor or actress.

**birth\_d** Birth day of actor or actress.

**birth\_y** Birth year of actor or actress.

**Details**

Although there have been only 84 Oscar ceremonies until 2012, there are 85 male winners and 85 female winners because ties happened on two occasions (1933 for the best actor and 1969 for the best actress).

**Source**

Journal of Statistical Education, <http://jse.amstat.org/datasets/oscars.dat.txt>, updated through 2019 using information from Oscars.org and Wikipedia.org.

### Examples

```
library(ggplot2)
library(dplyr)

ggplot(oscars, aes(x = award, y = age)) +
  geom_boxplot()

ggplot(oscars, aes(x = factor(birth_mo))) +
  geom_bar()

oscars |>
  count(birth_pl, sort = TRUE)
```

---

outliers

*Simulated datasets for different types of outliers*

---

### Description

Data sets for showing different types of outliers

### Usage

```
outliers
```

### Format

A data frame with 50 observations on the following 5 variables.

**x** a numeric vector

**y** a numeric vector

**x\_inf** a numeric vector

**y\_lev** a numeric vector

**y\_out** a numeric vector

### Examples

```
outliers
```

---

paralympic_1500	<i>Race time for Olympic and Paralympic 1500m.</i>
-----------------	--

---

## Description

Compiled gold medal times for the 1500m race in the Olympic Games and the Paralympic Games. The times given for contestants competing in the Paralympic Games are for athletes with different visual impairments; T11 indicates fully blind (with an option to race with a guide-runner) with T12 and T13 as lower levels of visual impairment.

## Usage

```
paralympic_1500
```

## Format

A data frame with 83 rows and 10 variables.

**year** Year the games took place.

**city** City of the games.

**country\_of\_games** Country of the games.

**division** Division: Men or Women.

**type** Type.

**name** Name of the athlete.

**country\_of\_athlete** Country of athlete.

**time** Time of gold medal race, in m:s.

**time\_min** Time of gold medal race, in decimal minutes (min + sec/60).

## Source

<https://www.paralympic.org/> and [https://en.wikipedia.org/wiki/1500\\_metres\\_at\\_the\\_Olympics](https://en.wikipedia.org/wiki/1500_metres_at_the_Olympics).

## Examples

```
library(ggplot2)
library(dplyr)

paralympic_1500 |>
  mutate(
    sight_level = case_when(
      type == "T11" ~ "total impairment",
      type == "T12" ~ "some impairment",
      type == "T13" ~ "some impairment",
      type == "Olympic" ~ "no impairment"
    )
  )
```

```

) |>
filter(division == "Men", year > 1920) |>
filter(type == "Olympic" | type == "T11") |>
ggplot(aes(x = year, y = time_min, color = sight_level, shape = sight_level)) +
  geom_point() +
  scale_x_continuous(breaks = seq(1924, 2020, by = 8)) +
  labs(
    title = "Men's Olympic and Paralympic 1500m race times",
    x = "Year",
    y = "Time of Race (minutes)",
    color = "Sight level",
    shape = "Sight level"
  )

```

---

penelope

*Guesses at the weight of Penelope (a cow)*


---

## Description

The data was collected by the Planet Money podcast to test a theory about crowd-sourcing. Penelope's actual weight was 1,355 pounds.

## Usage

```
penelope
```

## Format

A data frame with 17,184 observations on the following variable.

**weight** Guesses of Penelope's weight, in pounds.

## Source

<https://www.npr.org/sections/money/2015/08/07/429720443/17-205-people-guessed-the-weight-of-a-cow-1>

## Examples

```

library(ggplot2)

ggplot(penelope, aes(x = weight)) +
  geom_histogram(binwidth = 250)

summary(penelope$weight)

```



penetrating\_oil

*What's the best way to loosen a rusty bolt?***Description**

The channel Project Farm on YouTube investigated penetrating oils and other options for loosening rusty bolts. Eight options were evaluated, including a control group, to determine which was most effective.

**Usage**

penetrating\_oil

**Format**

A data frame with 30 observations on the following 2 variables.

**treatment** The different treatments tried: none (control), Heat (via blow torch), Acetone/ATF, AeroKroil, Liquid Wrench, PB Blaster, Royal Purple, and WD-40.

**torque** Torque required to loosen the rusty bolt, which was measured in foot-pounds.

**Source**

<https://www.youtube.com/watch?v=xUEob2oAKVs>

**Examples**

```
m <- lm(torque ~ treatment, data = penetrating_oil)
anova(m)

# There are 28 pairwise comparisons to be made.
xbar <- tapply(penetrating_oil$torque, penetrating_oil$treatment, mean)
n <- tapply(penetrating_oil$torque, penetrating_oil$treatment, length)
s <- summary(m)$sigma
df <- summary(m)$df[1]

diff <- c()
se <- c()
k <- 0
N <- length(n)
K <- N * (N - 1) / 2
for (i in 1:(N - 1)) {
  for (j in (i + 1):N) {
    k <- k + 1
    diff[k] <- xbar[i] - xbar[j]
    se[k] <- s * sqrt(1 / n[i] + 1 / n[j])
    if (2 * K * pt(-abs(diff[k] / se[k]), df) < 0.05) {
      cat("0.05 - ", names(n)[c(i, j)], "\n")
    } else if (2 * K * pt(-abs(diff[k] / se[k]), df) < 0.1) {
```

```

      cat("0.1 - ", names(n)[c(i, j)], "\n")
    } else if (2 * K * pt(-abs(diff[k] / se[k]), df) < 0.2) {
      cat("0.2 - ", names(n)[c(i, j)], "\n")
    } else if (2 * K * pt(-abs(diff[k] / se[k]), df) < 0.3) {
      cat("0.3 - ", names(n)[c(i, j)], "\n")
    }
  }
}

# Smallest p-value using Bonferroni
min(2 * K * pt(-abs(diff / se), df))

# Better pairwise comparison method.
anova(m1 <- aov(torque ~ treatment, data = penetrating_oil))
TukeyHSD(m1)

```

---

penny\_ages

*Penny Ages*


---

## Description

Sample of pennies and their ages. Taken in 2004.

## Usage

```
penny_ages
```

## Format

A data frame with 648 observations on the following 2 variables.

**year** Penny's year.

**age** Age as of 2004.

## Examples

```
hist(penny_ages$year)
```

---

pew\_energy\_2018*Pew Survey on Energy Sources in 2018*

---

**Description**

US-based survey on support for expanding six different sources of energy, including solar, wind, offshore drilling, hydrolic fracturing ("fracking"), coal, and nuclear.

**Usage**

```
pew_energy_2018
```

**Format**

The format is: List of 6 \$ solar\_panel\_farms : List of responses on solar farms. \$ wind\_turbine\_farms : List of responses on wind turbine farms. \$ offshore\_drilling : List of responses on offshore drilling. \$ hydrolic\_fracturing : List of responses on hydrolic fracturing. \$ coal\_mining : List of responses on coal mining. \$ nuclear\_power\_plants: List of responses on nuclear.

**Details**

We did not have access to individual responses in original dataset, so we took the published percentages and backed out the breakdown

**Source**

<https://www.pewresearch.org/science/2018/05/14/majorities-see-government-efforts-to-protect-the-environment/>

**Examples**

```
data(pew_energy_2018)
lapply(pew_energy_2018, head)
lapply(pew_energy_2018, length)
lapply(pew_energy_2018, table)
Prop <- function(x) {
  table(x) / length(x)
}
lapply(pew_energy_2018, Prop)
```

---

photo_classify	<i>Photo classifications: fashion or not</i>
----------------	--

---

### Description

This is a simulated dataset for photo classifications based on a machine learning algorithm versus what the true classification is for those photos. While the data are not real, they resemble performance that would be reasonable to expect in a well-built classifier.

### Usage

```
photo_classify
```

### Format

A data frame with 1822 observations on the following 2 variables.

**mach\_learn** The prediction by the machine learning system as to whether the photo is about fashion or not.

**truth** The actual classification of the photo by a team of humans.

### Details

The hypothetical ML algorithm has a precision of 90\ photos it claims are fashion, about 90\ The recall of the ML algorithm is about 64\ about fashion, it correctly predicts that they are about fashion about 64\ of the time.

### Source

The data are simulated / hypothetical.

### Examples

```
data(photo_classify)
table(photo_classify)
```

---

piracy	<i>Piracy and PIPA/SOPA</i>
--------	-----------------------------

---

### Description

This dataset contains observations on all 100 US Senators and 434 of the 325 US Congressional Representatives related to their support of anti-piracy legislation that was introduced at the end of 2011.

**Usage**

piracy

**Format**

A data frame with 534 observations on the following 8 variables.

**name** Name of legislator.

**party** Party affiliation as democrat (D), Republican (R), or Independent (I).

**state** Two letter state abbreviation.

**money\_pro** Amount of money in dollars contributed to the legislator's campaign in 2010 by groups generally thought to be supportive of PIPA/SOPA: movie and TV studios, record labels.

**money\_con** Amount of money in dollars contributed to the legislator's campaign in 2010 by groups generally thought to be opposed to PIPA/SOPA: computer and internet companies.

**years** Number of years of service in Congress.

**stance** Degree of support for PIPA/SOPA with levels Leaning No, No, Undecided, Unknown, Yes

**chamber** Whether the legislator is a member of either the house or senate.

**Details**

The Stop Online Piracy Act (SOPA) and the Protect Intellectual Property Act (PIPA) were two bills introduced in the US House of Representatives and the US Senate, respectively, to curtail copyright infringement. The bill was controversial because there were concerns the bill limited free speech rights. ProPublica, the independent and non-profit news organization, compiled this dataset to compare the stance of legislators towards the bills with the amount of campaign funds that they received from groups considered to be supportive of or in opposition to the legislation.

For more background on the legislation and the formulation of money\_pro and money\_con, read the documentation on ProPublica, linked below.

**Source**

<https://projects.propublica.org/sopa> The list may be slightly out of date since many politician's perspectives on the legislation were in flux at the time of data collection.

**Examples**

```
library(dplyr)
library(ggplot2)

pipa <- filter(piracy, chamber == "senate")

pipa |>
  group_by(stance) |>
  summarise(money_pro_mean = mean(money_pro, na.rm = TRUE)) |>
  ggplot(aes(x = stance, y = money_pro_mean)) +
  geom_col() +
  labs(
    x = "Stance", y = "Average contribution, in $",
```

```

    title = "Average contribution to the legislator's campaign in 2010",
    subtitle = "by groups supportive of PIPA/SOPA (movie and TV studios, record labels)"
  )

ggplot(pipa, aes(x = stance, y = money_pro)) +
  geom_boxplot() +
  labs(
    x = "Stance", y = "Contribution, in $",
    title = "Contribution by groups supportive of PIPA/SOPA",
    subtitle = "Movie and TV studios, record labels"
  )

ggplot(pipa, aes(x = stance, y = money_con)) +
  geom_boxplot() +
  labs(
    x = "Stance", y = "Contribution, in $",
    title = "Contribution by groups opposed to PIPA/SOPA",
    subtitle = "Computer and internet companies"
  )

pipa |>
  filter(
    money_pro > 0,
    money_con > 0
  ) |>
  mutate(for_pipa = ifelse(stance == "yes", "yes", "no")) |>
  ggplot(aes(x = money_pro, y = money_con, color = for_pipa)) +
  geom_point() +
  scale_color_manual(values = c("gray", "red")) +
  scale_y_log10() +
  scale_x_log10() +
  labs(
    x = "Contribution by pro-PIPA groups",
    y = "Contribution by anti-PIPA groups",
    color = "For PIPA"
  )

```

---

playing\_cards

*Table of Playing Cards in 52-Card Deck*

---

### Description

A table describing each of the 52 cards in a deck.

### Usage

playing\_cards

**Format**

A data frame with 52 observations on the following 2 variables.

**number** The number or card type.

**suit** Card suit, which takes one of four values: Club, Diamond, Heart, or Spade.

**face\_card** Whether the card counts as a face card.

**Source**

This extremely complex dataset was generated from scratch.

**Examples**

```
playing_cards <- data.frame(
  number = rep(c(2:10, "J", "Q", "K", "A"), 4),
  suit = rep(c("Spade", "Diamond", "Club", "Heart"), rep(13, 4))
)
playing_cards$face_card <-
  ifelse(playing_cards$number %in% c(2:10, "A"), "no", "yes")
```

---

PlotWLine

*Plot data and add a regression line*


---

**Description**

Plot data and add a regression line.

**Usage**

```
PlotWLine(
  x,
  y,
  xlab = "",
  ylab = "",
  col = fadeColor(4, "88"),
  cex = 1.2,
  pch = 20,
  n = 4,
  nMax = 4,
  yR = 0.1,
  axes = TRUE,
  ...
)
```

Arguments

x	Predictor variable.
y	Outcome variable.
xlab	x-axis label.
ylab	y-axis label.
col	Color of points.
cex	Size of points.
pch	Plotting character.
n	The preferred number of axis labels.
nMax	The maximum number of axis labels.
yR	y-limit buffer factor.
axes	Boolean to indicate whether or not to include axes.
...	Passed to plot.

See Also

[makeTube](#)

Examples

```
PlotWLine(1:10, seq(-5, -2, length.out = 10) + rnorm(10))
```

---

pm25_2011_durham	<i>Air quality for Durham, NC</i>
------------------	-----------------------------------

---

Description

Daily air quality is measured by the air quality index (AQI) reported by the Environmental Protection Agency in 2011.

Usage

```
pm25_2011_durham
```

Format

A data frame with 449 observations on the following 20 variables.

- date** Date
- aqs\_site\_id** The numeric site ID.
- poc** A numeric vector, the Parameter Occurance Code.
- daily\_mean\_pm2\_5\_concentration** A numeric vector with the average daily concentration of fine particulates, or particulate matter 2.5.



**units** A character vector with value ug/m3 LC.

**daily\_aqi\_value** A numeric vector with the daily air quality index.

**daily\_obs\_count** A numeric vector.

**percent\_complete** A numeric vector.

**aqs\_parameter\_code** A numeric vector.

**aqs\_parameter\_desc** A factor with levels PM2.5 - Local Conditions and Acceptable PM2.5 AQI & Speciation Mass.

**cbsa\_code** A numeric vector.

**cbsa\_name** A character vector with value Durham, NC.

**state\_code** A numeric vector.

**state** A character vector with value North Carolina.

**county\_code** A numeric vector.

**county** A character vector with value Durham.

**site\_latitude** A numeric vector of the latitude.

**site\_longitude** A numeric vector of the longitude.

**csa\_code** a numeric vector

**csa\_name** a factor with levels Raleigh-Durham-Cary, NC

### Source

US Environmental Protection Agency, AirData, 2011. [http://www3.epa.gov/airdata/ad\\_data\\_daily.html](http://www3.epa.gov/airdata/ad_data_daily.html)

### Examples

```
library(ggplot2)

ggplot(pm25_2011_durham, aes(x = date, y = daily_mean_pm2_5_concentration, group = 1)) +
  geom_line()
```

---

pm25_2022_durham	<i>Air quality for Durham, NC</i>
------------------	-----------------------------------

---

### Description

Daily air quality is measured by the air quality index (AQI) reported by the Environmental Protection Agency in 2022.

### Usage

```
pm25_2022_durham
```

**Format**

A data frame with 356 observations on the following 20 variables.

**date** Date.

**aqs\_site\_id** The numeric site ID.

**poc** A numeric vector, the Parameter Occurance Code.

**daily\_mean\_pm2\_5\_concentration** A numeric vector with the average daily concentration of fine particulates, or particulate matter 2.5.

**units** A character vector with value ug/m3 LC.

**daily\_aqi\_value** A numeric vector with the daily air quality index.

**daily\_obs\_count** A numeric vector.

**percent\_complete** A numeric vector.

**aqs\_parameter\_code** A numeric vector.

**aqs\_parameter\_desc** A factor vector with level PM2.5 – Local Conditions.

**cbsa\_code** A numeric vector.

**cbsa\_name** A character vector with value Durham-Chapel Hill, NC.

**state\_code** A numeric vector.

**state** A character vector with value North Carolina.

**county\_code** A numeric vector.

**county** A character vector with value Durham.

**site\_latitude** A numeric vector of the latitude.

**site\_longitude** A numeric vector of the longitude.

**site\_name** A character vector with value Durham Armory.

**Source**

US Environmental Protection Agency, AirData, 2022. [http://www3.epa.gov/airdata/ad\\_data\\_daily.html](http://www3.epa.gov/airdata/ad_data_daily.html)

**Examples**

```
library(ggplot2)

ggplot(pm25_2022_durham, aes(x = date, y = daily_mean_pm2_5_concentration, group = 1)) +
  geom_line()
```

---

poker	<i>Poker winnings during 50 sessions</i>
-------	--

---

**Description**

Poker winnings (and losses) for 50 days by a professional poker player.

**Usage**

```
poker
```

**Format**

A data frame with 49 observations on the following variable.

**winnings** Poker winnings and losses, in US dollars.

**Source**

Anonymity has been requested by the player.

**Examples**

```
library(ggplot2)

ggplot(poker, aes(x = winnings)) +
  geom_histogram(binwidth = 250)
```

---

possum	<i>Possums in Australia and New Guinea</i>
--------	--

---

**Description**

Data representing possums in Australia and New Guinea. This is a copy of the dataset by the same name in the DAAG package, however, the dataset included here includes fewer variables.

**Usage**

```
possum
```

**Format**

A data frame with 104 observations on the following 8 variables.

**site** The site number where the possum was trapped.

**pop** Population, either Vic (Victoria) or other (New South Wales or Queensland).

**sex** Gender, either m (male) or f (female).

**age** Age.

**head\_l** Head length, in mm.

**skull\_w** Skull width, in mm.

**total\_l** Total length, in cm.

**tail\_l** Tail length, in cm.

**Source**

Lindenmayer, D. B., Viggers, K. L., Cunningham, R. B., and Donnelly, C. F. 1995. Morphological variation among columns of the mountain brushtail possum, *Trichosurus caninus* Ogilby (Phalangeridae: Marsupiala). *Australian Journal of Zoology* 43: 449-458.

**Examples**

```
library(ggplot2)

# Skull width vs. head length
ggplot(possum, aes(x = head_l, y = skull_w)) +
  geom_point()

# Total length vs. sex
ggplot(possum, aes(x = total_l, fill = sex)) +
  geom_density(alpha = 0.5)
```

---

ppp\_201503

---

*US Poll on who it is better to raise taxes on*


---

**Description**

A poll of 691 people, with party affiliation collected, asked whether they think it's better to raise taxes on the rich or raise taxes on the poor.

**Usage**

```
ppp_201503
```

**Format**

A data frame with 691 observations on the following 2 variables.

**party** Political party affiliation.

**taxes** Support for who to raise taxes on.

**Source**

Public Policy Polling, Americans on College Degrees, Classic Literature, the Seasons, and More, data collected Feb 20-22, 2015.

**Examples**

```
library(ggplot2)

ggplot(ppp_201503, aes(x = party, fill = taxes)) +
  geom_bar(position = "fill") +
  labs(x = "Party", x = "Proportion", fill = "Taxes")
```

---

present	<i>Birth counts</i>
---------	---------------------

---

**Description**

An updated version of the historical Arbuthnot dataset. Numbers of boys and girls born in the United States between 1940 and 2002.

**Usage**

```
present
```

**Format**

A data frame with 63 observations on the following 3 variables.

**year** Year.

**boys** Number of boys born.

**girls** Number of girls born.

**Source**

Mathews, T. J., and Brady E. Hamilton. "Trend analysis of the sex ratio at birth in the United States." National vital statistics reports 53.20 (2005): 1-17.

**Examples**

```
library(ggplot2)

ggplot(present, mapping = aes(x = year, y = boys / girls)) +
  geom_line()
```

---

president	<i>United States Presidential History</i>
-----------	---

---

### Description

Summary of the changes in the president and vice president for the United States of America.

### Usage

president

### Format

A data frame with 67 observations on the following 5 variables.

**potus** President of the United States

**party** Political party of the president

**start** Start year

**end** End year

**vpotus** Vice President of the United States

### Source

Presidents of the United States (table) – infoplease.com (visited: Nov 2nd, 2010)

<https://www.infoplease.com/us/government/executive-branch/presidents> and <https://www.infoplease.com/us/government/executive-branch/vice-presidents>

### Examples

president

---

prison	<i>Prison isolation experiment</i>
--------	------------------------------------

---

### Description

Subjects from Central Prison in Raleigh, NC, volunteered for an experiment involving an "isolation" experience. The goal of the experiment was to find a treatment that reduces subjects' psychopathic deviant T scores. This score measures a person's need for control or their rebellion against control, and it is part of a commonly used mental health test called the Minnesota Multiphasic Personality Inventory (MMPI) test.

**Usage**

```
prison
```

**Format**

A data frame with 14 observations on the following 6 variables.

**pre\_trt1** Pre-treatment 1.

**post\_trt1** Post-treatment 1.

**pre\_trt2** Pre-treatment 2.

**post\_trt2** Post-treatment 2.

**pre\_trt3** Pre-treatment 3.

**post\_trt3** Post-treatment 3.

**Source**

<https://stat.duke.edu/datasets/prison-isolation>

**Examples**

```
prison
```

---

```
prius_mpg
```

*User reported fuel efficiency for 2017 Toyota Prius Prime*

---

**Description**

Fueleconomy.gov, the official US government source for fuel economy information, allows users to share gas mileage information on their vehicles. These data come from 19 users sharing gas mileage on their 2017 Toyota Prius Prime. Note that these data are user estimates and since the sources data cannot be verified, the accuracy of these estimates are not guaranteed.

**Usage**

```
prius_mpg
```

**Format**

A data frame with 19 observations on the following 10 variables.

**average\_mpg** Average mileage as estimated by the user.

**state** US State the user lives in.

**stop\_and\_go** Proportion of stop and go driving.

**highway** Proportion of highway driving.

**last\_updated** Date estimate was last updated.

**Source**

Fueleconomy.gov, <https://www.fueleconomy.gov/mpg/MPG.do?action=mpgData&vehicleID=38531&browser=true&details=on>, retrieved 2019-04-14.

**Examples**

```
library(ggplot2)
library(dplyr)

ggplot(prius_mpg, aes(x = average_mpg)) +
  geom_histogram(binwidth = 25)
```

---

qnormsim

*Generate simulated QQ plots*


---

**Description**

Create a 3 x 3 grid of quantile-quantile plots, the first of which corresponds to the input data. The other eight plots arise from simulating random normal data with the same mean, standard deviation, and length as the data. For use in comparing known-normal qqplots to an observed qqplot to assess normality.

**Usage**

```
qnormsim(sample, data)
```

**Arguments**

sample	the variable to be plotted.
data	data frame to use.

**Value**

A 3 x 3 grid of qqplots.

---

race\_justice

*Yahoo! News Race and Justice poll results*


---

**Description**

Results from a Yahoo! News poll conducted by YouGov on May 29-31, 2020. In total 1060 U.S. adults were asked a series of questions regarding race and justice in the wake of the killing of George Floyd by a police officer. Results in this dataset are percentages for the question, "Do you think Blacks and Whites receive equal treatment from the police?" For this particular question there were 1059 respondents.



**Usage**

```
race_justice
```

**Format**

A data frame with 1,059 rows and 2 variables.

**race\_eth** Race/ethnicity of respondent, with levels White, Black, Hispanic, and Other.

**response** Response to the question "Do you think Black and White people receive equal treatment from the police?", with levels Yes, No, and Not sure.

**Source**

[Yahoo! News Race and Justice - May 31, 2020.](#)

**Examples**

```
library(ggplot2)
library(dplyr)

# Conditional probabilities of response for each race/ethnicity
race_justice |>
  count(race_eth, response) |>
  group_by(race_eth) |>
  mutate(prop = n / sum(n))

# Stacked bar plot of counts
ggplot(race_justice, aes(x = race_eth, fill = response)) +
  geom_bar() +
  labs(
    x = "Race / ethnicity",
    y = "Count",
    title = "Do you think Black and White people receive
equal treatment from the police?",
    fill = "Response"
  )

# Stacked bar plot of proportions
ggplot(race_justice, aes(x = race_eth, fill = response)) +
  geom_bar(position = "fill") +
  labs(
    x = "Race / ethnicity",
    y = "Proportion",
    title = "Do you think Black and White people receive
equal treatment from the police?",
    fill = "Response"
  )
```

reddit\_finance

*Reddit Survey on Financial Independence.***Description**

A reduced set of the official results of the 2020 FI Survey from Reddit (r/financialindependence). Only responses that represent the respondent (not other contributors in the household) are listed. Does not include retired individuals. As per instructed, respondents give dollar values in their native currency.

**Usage**

reddit\_finance

**Format**

A data frame with 1998 rows and 65 variables.

**num\_incomes** How many individuals contribute to your household income?

**pan\_inc\_chg** As a result of the pandemic, did your earned income increase, decrease, or remain the same?

**pan\_inc\_chg\_pct** By how much did your earned income change?

**pan\_exp\_chg** As a result of the pandemic, did your expenses increase, decrease, or remain the same?

**pan\_exp\_chg\_pct** By how much did your expenses change?

**pan\_fi\_chg** As a result of the pandemic, did your FI (financially independent) number...

**pan\_ret\_date\_chg** As a result of the pandemic, did your planned RE (retirement) date...

**pan\_financial\_impact** Overall, how would you characterize the pandemic's impact on your finances?

**political** With which political party do you most closely identify? You do not need to be registered with a party to select it, answer based on your personal views.

**race\_eth** What is your race/ethnicity? Select all that apply.

**gender** What is your gender?

**age** What is your age?

**edu** What is the highest level of education you have completed?

**rel\_status** What is your relationship status?

**children** Do you have children?

**country** What country are you in?

**fin\_indy** Are you financially independent? Meaning you do not need to work for money, regardless of whether you work for money.

**fin\_indy\_num** At what amount invested will you consider yourself Financially Independent? (What is your FI number?)

**fin\_indy\_pct** What percent FI are you? (What percent of your FI number do you currently have?)

**retire\_invst\_num** At what amount invested do you intend to retire? (What is your RE number)

**tgt\_sf\_wthdrw\_rt** What is your target safe withdrawal rate? (If your answer is 3.5%, enter it as 3.5)

**max\_retire\_sup** How much annual income do you expect to have from the sources you selected in question T5 at the point where you are utilizing all of them (or a majority if you do not intend to use all at the same time)? Enter your answer as a dollar amount.

**retire\_exp** How much money (from your savings and other sources) do you intend to spend each year once you are retired? Enter your answer as a dollar amount.

**whn\_fin\_indy\_num** At what amount invested did you consider yourself Financially Independent? (AKA what was your "FI number")

**fin\_indy\_lvl** Which of the following would you have considered yourself at the time you reached Financial Independence:

**retire\_age** At what age do you intend to retire?

**stp\_whn\_fin\_indy** Do you intend to stop working for money when you reach financial independence?

**industry** Which of the following best describes the industry in which you currently or most recently work(ed)?

**employer** Which of the following best describes your current or most recent employer?

**role** Which of the following best describes your current or most recent job role?

**ft\_status** What is your current employment status? - Full Time

**pt\_status** What is your current employment status? - Part Time, Regular

**gig\_status** What is your current employment status? -Side Gig, Intermittent

**ne\_status** What is your current employment status? -Not Employed

**edu\_status** What is your current educational status?

**housing** What is your current housing situation?

**home\_value** Primary residence value.

**brokerage\_accts\_tax** Brokerage accounts (Taxable).

**retirement\_accts\_tax** Retirement accounts (Tax Advantaged).

**cash** Cash / cash equivalents (Savings, Checking, C.D.s, Money Market).

**invst\_accts** Dedicated Savings/Investment Accounts (Healthcare, Education).

**spec\_crypto** Speculation (Crypto, P2P Lending, Gold, etc.).

**invst\_prop\_bus\_own** investment properties / owned business(es).

**other\_val** Other assets.

**student\_loans** Outstanding student loans.

**mortgage** Outstanding mortgage / HELOC.

**auto\_loan** Outstanding auto loans.

**credit\_personal\_loan** Outstanding credit cards / personal loans.

**medical\_debt** Outstanding medical debt.

**invst\_prop\_bus\_own\_debt** Debt from investment properties / owned business.

**other\_debt** Debt from other sources.

**2020\_gross\_inc** What was your 2020 gross (pre-tax, pre-deductions) annual household income?

**2020\_housing\_exp** Housing expenses(rent, mortgage, insurance, taxes, upkeep).

**2020\_utilities\_exp** Utilities expenses(phone, internet, gas, electric, water, sewer).

**2020\_transp\_exp** Transportation expenses(car payment, bus / subway tickets, gas, insurance, maintenance).

**2020\_necessities\_exp** Necessities expenses(Groceries, Clothing, Personal Care, Household Supplies).

**2020\_lux\_exp** Luxury expenses (Restaurants/Dining, Entertainment, Hobbies, Travel, Pets, Gifts).

**2020\_child\_exp** Children expenses(child care, soccer team, etc.).

**2020\_debt\_repay** Debt repayment (excluding mortgage/auto).

**2020\_invst\_save** Investments / savings.

**2020\_charity** Charity / Tithing.

**2020\_healthcare\_exp** Healthcare expenses(direct costs, co-pays, insurance you pay).

**2020\_taxes** Taxes (the sum of all taxes paid, including amounts deducted from paychecks).

**2020\_edu\_exp** Education expenses.

**2020\_other\_exp** Other expenses.

## Source

Reddit Official 2020 FI Survey Results, [https://www.reddit.com/r/financialindependence/comments/m1q8ia/official\\_2020\\_fi\\_survey\\_results/](https://www.reddit.com/r/financialindependence/comments/m1q8ia/official_2020_fi_survey_results/)

## Examples

```
library(ggplot2)

# Histogram of Expected Retirement Age.
ggplot(reddit_finance, aes(retire_age)) +
  geom_bar(na.rm = TRUE) +
  labs(
    title = "At what age do you expect to retire?",
    x = "Age Bracket",
    y = "Number of Respondents"
  )

# Histogram of Dollar Amount at Which FI was reached.
ggplot(reddit_finance, aes(whn_fin_indy_num)) +
  geom_histogram(na.rm = TRUE, bins = 20) +
  labs(
    title = "At what amount invested did you consider\nyourself Financially Independent?",
    x = "Dollar Amount (in local currency)",
    y = "Number of Respondents"
  )
```

resume

*Which resume attributes drive job callbacks?***Description**

This experiment data comes from a study that sought to understand the influence of race and gender on job application callback rates. The study monitored job postings in Boston and Chicago for several months during 2001 and 2002 and used this to build up a set of test cases. Over this time period, the researchers randomly generating resumes to go out to a job posting, such as years of experience and education details, to create a realistic-looking resume. They then randomly assigned a name to the resume that would communicate the applicant's gender and race. The first names chosen for the study were selected so that the names would predominantly be recognized as belonging to black or white individuals. For example, Lakisha was a name that their survey indicated would be interpreted as a black woman, while Greg was a name that would generally be interpreted to be associated with a white male.

**Usage**

resume

**Format**

A data frame with 4870 observations, representing 4870 resumes, over 30 different variables that describe the job details, the outcome (received\_callback), and attributes of the resume.

**job\_ad\_id** Unique ID associated with the advertisement.

**job\_city** City where the job was located.

**job\_industry** Industry of the job.

**job\_type** Type of role.

**job\_fed\_contractor** Indicator for if the employer is a federal contractor.

**job\_equal\_opp\_employer** Indicator for if the employer is an Equal Opportunity Employer.

**job\_ownership** The type of company, e.g. a nonprofit or a private company.

**job\_req\_any** Indicator for if any job requirements are listed. If so, the other job\_req\_\* fields give more detail.

**job\_req\_communication** Indicator for if communication skills are required.

**job\_req\_education** Indicator for if some level of education is required.

**job\_req\_min\_experience** Amount of experience required.

**job\_req\_computer** Indicator for if computer skills are required.

**job\_req\_organization** Indicator for if organization skills are required.

**job\_req\_school** Level of education required.

**received\_callback** Indicator for if there was a callback from the job posting for the person listed on this resume.

**firstname** The first name used on the resume.

- race** Inferred race associated with the first name on the resume.
- gender** Inferred gender associated with the first name on the resume.
- years\_college** Years of college education listed on the resume.
- college\_degree** Indicator for if the resume listed a college degree.
- honors** Indicator for if the resume listed that the candidate has been awarded some honors.
- worked\_during\_school** Indicator for if the resume listed working while in school.
- years\_experience** Years of experience listed on the resume.
- computer\_skills** Indicator for if computer skills were listed on the resume. These skills were adapted for listings, though the skills were assigned independently of other details on the resume.
- special\_skills** Indicator for if any special skills were listed on the resume.
- volunteer** Indicator for if volunteering was listed on the resume.
- military** Indicator for if military experience was listed on the resume.
- employment\_holes** Indicator for if there were holes in the person's employment history.
- has\_email\_address** Indicator for if the resume lists an email address.
- resume\_quality** Each resume was generally classified as either lower or higher quality.

### Details

Because this is an experiment, where the race and gender attributes are being randomly assigned to the resumes, we can conclude that any statistically significant difference in callback rates is causally linked to these attributes.

Do you think it's reasonable to make a causal conclusion? You may have some health skepticism. However, do take care to appreciate that this was an experiment: the first name (and so the inferred race and gender) were randomly assigned to the resumes, and the quality and attributes of a resume were assigned independent of the race and gender. This means that any effects we observe are in fact causal, and the effects related to race are both statistically significant and very large: white applicants had about a 50\

Do you still have doubts lingering in the back of your mind about the validity of this study? Maybe a counterargument about why the standard conclusions from this study may not apply? The article summarizing the results was exceptionally well-written, and it addresses many potential concerns about the study's approach. So if you're feeling skeptical about the conclusions, please find the link below and explore!

### Source

Bertrand M, Mullainathan S. 2004. "Are Emily and Greg More Employable than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination". The American Economic Review 94:4 (991-1013). [doi:10.3386/w9873](https://doi.org/10.3386/w9873).

### See Also

[resume](#)

**Examples**

```

head(resume, 5)

# Some checks to confirm balance between race and
# other attributes of a resume. There should be
# some minor differences due to randomness, but
# each variable should be (and is) generally
# well-balanced.
table(resume$race, resume$years_college)
table(resume$race, resume$college_degree)
table(resume$race, resume$honors)
table(resume$race, resume$worked_during_school)
table(resume$race, resume$years_experience)
table(resume$race, resume$computer_skills)
table(resume$race, resume$special_skills)
table(resume$race, resume$volunteer)
table(resume$race, resume$military)
table(resume$race, resume$employment_holes)
table(resume$race, resume$has_email_address)
table(resume$race, resume$resume_quality)

# Regarding the callback outcome for race,
# we observe a very large difference.
tapply(
  resume$received_callback,
  resume[c("race", "gender")],
  mean
)

# Natural question: is this statistically significant?
# A proper analysis would take into account the
# paired nature of the data. For each ad, let's
# compute the following statistic:
#   <callback rate for white candidates>
#   - <callback rate for black candidates>
# First construct the callbacks for white and
# black candidates by ad ID:
table(resume$race)
cb_white <- with(
  subset(resume, race == "white"),
  tapply(received_callback, job_ad_id, mean)
)
cb_black <- with(
  subset(resume, race == "black"),
  tapply(received_callback, job_ad_id, mean)
)
# Next, compute the differences, where the
# names(cb_white) part ensures we matched up the
# job ad IDs.
diff <- cb_white - cb_black[names(cb_white)]
# Finally, we can apply a t-test on the differences:
t.test(diff)

```

```

# There is very strong evidence of an effect.

# Here's a similar check with gender. There are
# more female-inferred candidates used on the resumes.
table(resume$gender)
cb_male <- with(
  subset(resume, gender == "m"),
  tapply(received_callback, job_ad_id, mean)
)
cb_female <- with(
  subset(resume, gender == "f"),
  tapply(received_callback, job_ad_id, mean)
)
diff <- cb_female - cb_male[names(cb_female)]
# The `na.rm = TRUE` part ensures we limit to jobs
# where both a male and female resume were sent.
t.test(diff, na.rm = TRUE)
# There is no statistically significant difference.

# Was that the best analysis? Absolutely not!
# However, the analysis was unbiased. To get more
# precision on the estimates, we could build a
# multivariate model that includes many characteristics
# of the resumes sent, e.g. years of experience.
# Since those other characteristics were assigned
# independently of the race characteristics, this
# means the race finding will almost certainly will
# hold. However, it is possible that we'll find
# more interesting results with the gender investigation.

```

---

res\_demo\_1

*Simulated data for regression*


---

## Description

Simulated data for regression

## Usage

```
res_demo_1
```

## Format

A data frame with 100 observations on the following 3 variables.

**x** a numeric vector

**y\_lin** a numeric vector

**y\_fan\_back** a numeric vector



**Examples**

res\_demo\_1

res\_demo\_2

*Simulated data for regression***Description**

Simulated data for regression

**Usage**

res\_demo\_2

**Format**

A data frame with 300 observations on the following 3 variables.

**x** a numeric vector**y\_fan** a numeric vector**y\_log** a numeric vector**Examples**

res\_demo\_2

rosling\_responses

*Sample Responses to Two Public Health Questions***Description**

Public health has improved and evolved, but has the public's knowledge changed with it? This dataset explores sample responses for two survey questions posed by Hans Rosling during lectures to a wide array of well-educated audiences.

**Usage**

rosling\_responses

**Format**

A data frame with 278 rows and 3 variables:

**question** ID for the question being posed.**response** Noting whether the response was correct or incorrect.**prob\_random\_correct** The probability the person would have guessed the answer correctly if they were guessing completely randomly.

**Source**

The samples we describe are plausible based on the exact rates observed in larger samples. For more info on the actual rates observed, visit <https://www.gapminder.org>.

Another relevant reference is a book by Hans Rosling, Anna Rosling Ronnlund, and Ola Rosling called [Factfulness](#).

**Examples**

```
frac_correct <- tapply(
  rosling_responses$response == "correct",
  rosling_responses$question,
  mean
)
frac_correct
n <- table(rosling_responses$question)
n
expected <- tapply(
  rosling_responses$prob_random_correct,
  rosling_responses$question,
  mean
)

# Construct confidence intervals.
se <- sqrt(frac_correct * (1 - frac_correct) / n)
# Lower bounds.
frac_correct - 1.96 * se
# Upper bounds.
frac_correct + 1.96 * se

# Construct Z-scores and p-values.
z <- (frac_correct - expected) / se
pt(z, df = n - 1)
```

---

russian\_influence\_on\_us\_election\_2016

*Russians' Opinions on US Election Influence in 2016*

---

**Description**

Survey of Russian citizens on whether they believed their government tried to influence the 2016 US election. The survey was taken in Spring 2018 by Pew Research.

**Usage**

```
russian_influence_on_us_election_2016
```

**Format**

A data frame with 506 observations on the following variable.

**influence\_2016** Response of the Russian survey participant to the question of whether their government tried to influence the 2016 election in the United States.

**Details**

The actual sample size was 1000. However, the original data were not from a simple random sample; after accounting for the design, the equivalent sample size was 506, which was what was used for the dataset here to keep things simpler for intro stat analyses.

**Source**

<https://www.pewresearch.org/global/2018/08/21/russians-say-their-government-did-not-try-to-influence-the-2016-election/>

**Examples**

```
table(russian_influence_on_us_election_2016)
```

---

salinity	<i>Salinity in Bimini Lagoon, Bahamas</i>
----------	---

---

**Description**

Data collected at three different water masses in the Bimini Lagoon, Bahamas.

**Usage**

```
salinity
```

**Format**

A data frame with 30 rows and 2 variables.

**site\_number** Location where measurements were taken.

**salinity\_ppt** Salinity value in parts per thousand.

**Source**

Till, R. (1974) Statistical Methods for the Earth Scientist: An Introduction. London: Macmillan, 104.

**Examples**

```
library(ggplot2)
library(broom)

ggplot(salinity, aes(x = salinity_ppt)) +
  geom_dotplot() +
  facet_wrap(~site_number, ncol = 1)

tidy(aov(salinity_ppt ~ site_number, data = salinity))
```

---

 satgpa

*SAT and GPA data*


---

**Description**

SAT and GPA data for 1000 students at an unnamed college.

**Usage**

```
satgpa
```

**Format**

A data frame with 1000 observations on the following 6 variables.

**sex** Gender of the student.

**sat\_v** Verbal SAT percentile.

**sat\_m** Math SAT percentile.

**sat\_sum** Total of verbal and math SAT percentiles.

**hs\_gpa** High school grade point average.

**fy\_gpa** First year (college) grade point average.

**Source**

Educational Testing Service originally collected the data.

**References**

<https://chance.dartmouth.edu/course/Syllabi/Princeton96/ETSValidation.html>

**Examples**

```
library(ggplot2)
library(broom)

# Verbal scores
ggplot(satgpa, aes(x = sat_v, fy_gpa)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(
    x = "Verbal SAT percentile",
    y = "First year (college) grade point average"
  )

mod <- lm(fy_gpa ~ sat_v, data = satgpa)
tidy(mod)

# Math scores
ggplot(satgpa, aes(x = sat_m, fy_gpa)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(
    x = "Math SAT percentile",
    y = "First year (college) grade point average"
  )

mod <- lm(fy_gpa ~ sat_m, data = satgpa)
tidy(mod)
```

---

sat\_improve

---

*Simulated data for SAT score improvement*


---

**Description**

Fake data for score improvements from students who took a course from an SAT score improvement company.

**Usage**

```
sat_improve
```

**Format**

A data frame with 30 observations on the following variable.

**sat\_improve** a numeric vector

**Examples**

```
sat_improve
```

sa\_gdp\_elec

*Sustainability and Economic Indicators for South Africa.***Description**

Includes yearly data on gdp, gni, co2 emissions, start up costs.

**Usage**

```
sa_gdp_elec
```

**Format**

A data frame with 16 rows and 7 variables.

**year** Year data collected.

**access\_elec** Access to electricity as a percentage of the population.

**startup** cost of business startup procedures as a percent of GNI.

**co2** CO2 emission in kt (kiloton).

**gdp** GDP per capita, PPP in constant 2017 international dollars.

**gni** GNI per capita, PPP in constant 2017 international dollars.

**co2\_kg\_ppp** kg per 2017 PPP dollars of GDP.

**Source**

- [World Bank I](#)
- [World Bank II](#)
- [Carbon Dioxide Information Analysis Center, Environmental Sciences Division, Oak Ridge National Laboratory](#)

**Examples**

```
library(ggplot2)

ggplot(sa_gdp_elec, aes(year, access_elec)) +
  geom_point(alpha = 0.3) +
  labs(
    x = "Year",
    y = "Percent of Population",
    title = "Access to Electricity in South Africa 2003 - 2018"
  )
```

---

scale\_color\_openintro *Color scale constructor for OpenIntro IMS colors*

---

## Description

Color scale constructor for OpenIntro IMS colors

## Usage

```
scale_color_openintro(palette = "main", discrete = TRUE, reverse = FALSE, ...)
```

## Arguments

palette	Character name of palette in <a href="#">openintro_palettes</a>
discrete	Boolean indicating whether color aesthetic is discrete or not
reverse	Boolean indicating whether the palette should be reversed
...	Additional arguments passed to <a href="#">ggplot2::discrete_scale()</a> or <a href="#">ggplot2::scale_color_gradientn()</a> used respectively when discrete is TRUE or FALSE

## Examples

```
library(ggplot2)

# Categorical variable with three levels
ggplot(evals, aes(
  x = bty_avg, y = score,
  color = rank, shape = rank
)) +
  geom_jitter(size = 2, alpha = 0.6) +
  scale_color_openintro("three")

# Categorical variable with two levels
ggplot(evals, aes(
  x = bty_avg, y = score,
  color = language, shape = language
)) +
  geom_jitter(size = 2, alpha = 0.6) +
  scale_color_openintro("two")

# Continuous variable
# Generates a palette, but not recommended
ggplot(evals, aes(
  x = bty_avg, y = score,
  color = score
)) +
  geom_jitter(size = 2, alpha = 0.8) +
  scale_color_openintro(discrete = FALSE)
```

```
# For continous palettes
# use scale_color_gradient instead
ggplot(evals, aes(
  x = bty_avg, y = score,
  color = score
)) +
  geom_jitter(size = 2) +
  scale_color_gradient(low = IMSCOL["blue", "full"], high = IMSCOL["blue", "f6"])

ggplot(evals, aes(
  x = bty_avg, y = score,
  color = cls_perc_eval
)) +
  geom_jitter(size = 2) +
  scale_color_gradient(low = COL["red", "full"], high = COL["red", "f8"])
```

---

scale\_fill\_openintro    *Fill scale constructor for OpenIntro IMS colors*

---

## Description

Fill scale constructor for OpenIntro IMS colors

## Usage

```
scale_fill_openintro(palette = "main", discrete = TRUE, reverse = FALSE, ...)
```

## Arguments

palette	Character name of palette in <a href="#">openintro_palettes</a>
discrete	Boolean indicating whether color aesthetic is discrete or not
reverse	Boolean indicating whether the palette should be reversed
...	Additional arguments passed to <a href="#">ggplot2::discrete_scale()</a> or <a href="#">ggplot2::scale_fill_gradientn()</a> used respectively when discrete is TRUE or FALSE

## Examples

```
library(ggplot2)
library(dplyr)

# Categorical variable with two levels
ggplot(evals, aes(x = ethnicity, fill = ethnicity)) +
  geom_bar() +
  scale_fill_openintro("two")

# Categorical variable with three levels
ggplot(evals, aes(x = rank, fill = rank)) +
  geom_bar() +
  scale_fill_openintro("three")
```



```

# Continuous variable with levels
# Generates a palette, but may not be the best palette
# in terms of color-blind and grayscale friendliness
ggplot(diamonds, aes(x = clarity, fill = clarity)) +
  geom_bar() +
  scale_fill_openintro()

# For continuous palettes
# use scale_color_gradient instead
ggplot(evals, aes(
  x = bty_avg, y = score,
  color = score
)) +
  geom_jitter(size = 2) +
  scale_color_gradient(low = IMSCOL["blue", "full"], high = IMSCOL["blue", "f6"])

ggplot(evals, aes(
  x = bty_avg, y = score,
  color = cls_perc_eval
)) +
  geom_jitter(size = 2) +
  scale_color_gradient(low = IMSCOL["green", "full"], high = IMSCOL["green", "f6"])

```

---

scotus\_healthcare

*Public Opinion with SCOTUS ruling on American Healthcare Act*


---

## Description

On June 28, 2012 the U.S. Supreme Court upheld the much debated 2010 healthcare law, declaring it constitutional. A Gallup poll released the day after this decision indicates that 46% of 1,012 Americans agree with this decision.

## Usage

```
scotus_healthcare
```

## Format

A data frame with 1012 observations on the following variable.

**response** Response values reported are agree and other.

## Source

Gallup, Americans Issue Split Decision on Healthcare Ruling, retrieved 2012-06-28.

## Examples

```
table(scotus_healthcare)
```

---

seattlepets	<i>Names of pets in Seattle</i>
-------------	---------------------------------

---

**Description**

Names of registered pets in Seattle, WA, between 2003 and 2018, provided by the city's Open Data Portal.

**Usage**

```
seattlepets
```

**Format**

A data frame with 52,519 rows and 7 variables:

**license\_issue\_date** Date the animal was registered with Seattle

**license\_number** Unique license number

**animal\_name** Animal's name

**species** Animal's species (dog, cat, goat, etc.)

**primary\_breed** Primary breed of the animal

**secondary\_breed** Secondary breed if mixed

**zip\_code** Zip code animal is registered in

**Source**

These data come from Seattle's Open Data Portal, <https://data.seattle.gov/Community/Seattle-Pet-Licenses/jguv-t9rb>

---

sex_discrimination	<i>Bank manager recommendations based on sex</i>
--------------------	--

---

**Description**

Study from the 1970s about whether sex influences hiring recommendations.

**Usage**

```
sex_discrimination
```

**Format**

A data frame with 48 observations on the following 2 variables.

**sex** a factor with levels female and male

**decision** a factor with levels not promoted and promoted

**Source**

Rosen B and Jerdee T. 1974. Influence of sex role stereotypes on personnel decisions. *Journal of Applied Psychology* 59(1):9-14.

**Examples**

```
library(ggplot2)

table(sex_discrimination)

ggplot(sex_discrimination, aes(y = sex, fill = decision)) +
  geom_bar(position = "fill")
```

---

simpsons\_paradox\_covid

*Simpson's Paradox: Covid*

---

**Description**

A dataset on Delta Variant Covid-19 cases in the UK. This dataset gives a great example of Simpson's Paradox. When aggregating results without regard to age group, the death rate for vaccinated individuals is higher – but they have a much higher risk population. Once we look at populations with more comparable risks (breakout age groups), we see that the vaccinated group tends to be lower risk in each risk-bucketed group and that many of the higher risk patients had gotten vaccinated. The dataset was brought to OpenIntro's attention by Matthew T. Brenneman of Embry-Riddle Aeronautical University. Note: some totals in the original source differ as there were some cases that did not have ages associated with them.

**Usage**

```
simpsons_paradox_covid
```

**Format**

A data frame with 286,166 rows and 3 variables:

**age\_group** Age of the person. Levels: under 50, 50 +.

**vaccine\_status** Vaccination status of the person. Note: the vaccinated group includes those who were only partially vaccinated. Levels: vaccinated, unvaccinated

**outcome** Did the person die from the Delta variant? Levels: death and survived.

**Source**

Public Health England: Technical briefing 20

**Examples**

```
library(dplyr)
library(scales)
# Calculate the mortality rate for all cases by vaccination status
simpsons_paradox_covid |>
  group_by(vaccine_status, outcome) |>
  summarize(count = n()) |>
  ungroup() |>
  group_by(vaccine_status) |>
  mutate(total = sum(count)) |>
  filter(outcome == "death") |>
  select(c(vaccine_status, count, total)) |>
  mutate(mortality_rate = label_percent(accuracy = 0.01)(round(count / total, 4))) |>
  select(-c(count, total))

# Calculate mortality rate by age group and vaccination status
simpsons_paradox_covid |>
  group_by(age_group, vaccine_status, outcome) |>
  summarize(count = n()) |>
  ungroup() |>
  group_by(age_group, vaccine_status) |>
  mutate(total = sum(count)) |>
  filter(outcome == "death") |>
  select(c(age_group, vaccine_status, count, total)) |>
  mutate(mortality_rate = label_percent(accuracy = 0.01)(round(count / total, 4))) |>
  select(-c(count, total))
```

---

simulated\_dist

*Simulated datasets, not necessarily drawn from a normal distribution.*


---

**Description**

Data were simulated in R, and some of the simulations do not represent data from actual normal distributions.

**Usage**

```
simulated_dist
```

**Format**

The format is: List of 4 \$ d1: dataset of 100 observations. \$ d2: dataset of 50 observations. \$ d3: num dataset of 500 observations. \$ d4: dataset of 15 observations. \$ d5: num dataset of 25 observations. \$ d6: dataset of 50 observations.

**Examples**

```
data(simulated_dist)
lapply(simulated_dist, qqnorm)
```

---

simulated_normal	<i>Simulated datasets, drawn from a normal distribution.</i>
------------------	--

---

**Description**

Data were simulated using `rnorm`.

**Usage**

```
simulated_normal
```

**Format**

The format is: List of 3 \$ n40 : 40 observations from a standard normal distribution. \$ n100: 100 observations from a standard normal distribution. \$ n400: 400 observations from a standard normal distribution.

**Examples**

```
data(simulated_normal)
lapply(simulated_normal, qqnorm)
```

---

simulated_scatter	<i>Simulated data for sample scatterplots</i>
-------------------	---

---

**Description**

Fake data.

**Usage**

```
simulated_scatter
```

**Format**

A data frame with 500 observations on the following 3 variables.

**group** Group, representing data for a specific plot.

**x** x-value.

**y** y-value.

**Examples**

```
library(ggplot2)

ggplot(simulated_scatter, aes(x = x, y = y)) +
  geom_point() +
  facet_wrap(~group)
```

---

sinusitis	<i>Sinusitis and antibiotic experiment</i>
-----------	--

---

**Description**

Researchers studying the effect of antibiotic treatment for acute sinusitis to one of two groups: treatment or control.

**Usage**

```
sinusitis
```

**Format**

A data frame with 166 observations on the following 2 variables.

**group** a factor with levels control and treatment

**self\_reported\_improvement** a factor with levels no and yes

**Source**

J.M. Garbutt et al. Amoxicillin for Acute Rhinosinusitis: A Randomized Controlled Trial. In: JAMA: The Journal of the American Medical Association 307.7 (2012), pp. 685-692.

**Examples**

```
sinusitis
```

---

sleep_deprivation	<i>Survey on sleep deprivation and transportation workers</i>
-------------------	---

---

**Description**

The National Sleep Foundation conducted a survey on the sleep habits of randomly sampled transportation workers and a control sample of non-transportation workers.

**Usage**

```
sleep_deprivation
```

**Format**

A data frame with 1087 observations on the following 2 variables.

**sleep** a factor with levels <6, 6-8, and >8

**profession** a factor with levels bus / taxi / limo drivers, control, pilots, train operators, truck drivers

**Source**

National Sleep Foundation, 2012 Sleep in America Poll: Transportation Workers' Sleep, 2012.

<https://www.sleepfoundation.org/professionals/sleep-american-polls/2012-sleep-america-poll-transportation-workers-sleep>

**Examples**

```
sleep_deprivation
```

---

smallpox	<i>Smallpox vaccine results</i>
----------	---------------------------------

---

**Description**

A sample of 6,224 individuals from the year 1721 who were exposed to smallpox in Boston. Some of them had received a vaccine (inoculated) while others had not. Doctors at the time believed that inoculation, which involves exposing a person to the disease in a controlled form, could reduce the likelihood of death.

**Usage**

```
smallpox
```

**Format**

A data frame with 6224 observations on the following 2 variables.

**result** Whether the person died or lived.

**inoculated** Whether the person received inoculated.

**Source**

Fenner F. 1988. Smallpox and Its Eradication (History of International Public Health, No. 6). Geneva: World Health Organization. ISBN 92-4-156110-6.

**Examples**

```
data(smallpox)
table(smallpox)
```

smoking

*UK Smoking Data***Description**

Survey data on smoking habits from the UK. The dataset can be used for analyzing the demographic characteristics of smokers and types of tobacco consumed.

**Usage**

smoking

**Format**

A data frame with 1691 observations on the following 12 variables.

**gender** Gender with levels Female and Male.

**age** Age.

**marital\_status** Marital status with levels Divorced, Married, Separated, Single and Widowed.

**highest\_qualification** Highest education level with levels A Levels, Degree, GCSE/CSE, GCSE/O Level, Higher/Sub Degree, No Qualification, ONC/BTEC and Other/Sub Degree

**nationality** Nationality with levels British, English, Irish, Scottish, Welsh, Other, Refused and Unknown.

**ethnicity** Ethnicity with levels Asian, Black, Chinese, Mixed, White and Refused Unknown.

**gross\_income** Gross income with levels Under 2,600, 2,600 to 5,200, 5,200 to 10,400, 10,400 to 15,600, 15,600 to 20,800, 20,800 to 28,600, 28,600 to 36,400, Above 36,400, Refused and Unknown.

**region** Region with levels London, Midlands & East Anglia, Scotland, South East, South West, The North and Wales

**smoke** Smoking status with levels No and Yes

**amt\_weekends** Number of cigarettes smoked per day on weekends.

**amt\_weekdays** Number of cigarettes smoked per day on weekdays.

**type** Type of cigarettes smoked with levels Packets, Hand-Rolled, Both/Mainly Packets and Both/Mainly Hand-Rolled

**Source**

National STEM Centre, Large Datasets from stats4schools, <https://www.stem.org.uk/resources/elibrary/resource/28452/large-datasets-stats4schools>.



### Examples

```
library(ggplot2)

ggplot(smoking, aes(x = amt_weekends)) +
  geom_histogram(binwidth = 5)

ggplot(smoking, aes(x = amt_weekdays)) +
  geom_histogram(binwidth = 5)

ggplot(smoking, aes(x = gender, fill = smoke)) +
  geom_bar(position = "fill")

ggplot(smoking, aes(x = marital_status, fill = smoke)) +
  geom_bar(position = "fill")
```

---

snowfall

*Snowfall at Paradise, Mt. Rainier National Park*

---

### Description

Annual snowfall data for Paradise, Mt. Rainier National Park. To include a full winter season, snowfall is recorded from July 1 to June 30. Data from 1943-1946 not available due to road closure during World War II. Records also unavailable from 1948-1954.

### Usage

```
snowfall
```

### Format

A data frame with 100 rows and 3 variables.

**year\_start** The year snowfall measurement began on July 1.

**year\_end** The year snowfall measurement ended on June 30.

**total\_snow** Snowfall measured in inches.

### Source

[National Parks Services.](#)

### Examples

```
library(ggplot2)

ggplot(snowfall, aes(x = total_snow)) +
  geom_histogram(binwidth = 50) +
  labs(
    title = "Annual Snowfall",
```

```

    subtitle = "Paradise, Mt. Rainier National Park",
    x = "Snowfall (in.)",
    y = "Number of Years",
    caption = "Source: National Parks Services"
  )

ggplot(snowfall, aes(x = year_start, y = total_snow, group = 1)) +
  geom_line() +
  labs(
    title = "Annual Snowfall",
    subtitle = "Paradise, Mt. Rainier National Park",
    y = "Snowfall (in.)",
    x = "Year",
    caption = "Source: National Parks Services"
  )

```

---

socialexp

*Social experiment*


---

## Description

A "social experiment" conducted by a TV program questioned what people do when they see a very obviously bruised woman getting picked on by her boyfriend. On two different occasions at the same restaurant, the same couple was depicted. In one scenario the woman was dressed "provocatively" and in the other scenario the woman was dressed "conservatively". The table below shows how many restaurant diners were present under each scenario, and whether or not they intervened.

## Usage

```
socialexp
```

## Format

A data frame with 45 observations on the following 2 variables.

**intervene** Whether other diners intervened or not.

**scenario** How the woman was dressed.

## Examples

```
table(socialexp)
```

---

`soda`

---

*soda***Description**

A randomly generated dataset of soda preference (cola or orange) based on location.

**Usage**

```
soda
```

**Format**

A data frame with 60 observations on the following 2 variables.

**drink** Soda preference, cola or orange.

**location** Is the person from the West coast or East coast?

**Examples**

```
library(dplyr)

soda |>
  count(location, drink)
```

---

`solar`

---

*Energy Output From Two Solar Arrays in San Francisco***Description**

The data provide the energy output for several months from two roof-top solar arrays in San Francisco. This city is known for having highly variable weather, so while these two arrays are only about 1 mile apart from each other, the Inner Sunset location tends to have more fog.

**Usage**

```
solar
```

**Format**

A data frame with 284 observations on the following 3 variables. Each row represents a single day for one of the arrays.

**location** Location for the array.

**date** Date.

**kwh** Number of kWh

## Details

The Haight-Ashbury array is a 10.4 kWh array, while the Inner Sunset array is a 2.8 kWh array. The kWh units represents kilowatt-hours, which is the unit of energy that typically is used for electricity bills. The cost per kWh in San Francisco was about \$0.25 in 2016.

## Source

These data were provided by Larry Rosenfeld, a resident in San Francisco.

## Examples

```
solar.is <- subset(solar, location == "Inner_Sunset")
solar.ha <- subset(solar, location == "Haight_Ashbury")
plot(solar.is$date, solar.is$kwh, type = "l", ylim = c(0, max(solar$kwh)))
lines(solar.ha$date, solar.ha$kwh, col = 4)

d <- merge(solar.ha, solar.is, by = "date")
plot(d$date, d$kwh.x / d$kwh.y, type = "l")
```

---

sowc\_child\_mortality    *SOWC Child Mortality Data.*

---

## Description

Child mortality data from UNICEF's State of the World's Children 2019 Statistical Tables.

## Usage

```
sowc_child_mortality
```

## Format

A data frame with 195 rows and 19 variables.

**countries\_and\_areas** Country or area name.

**under5\_mortality\_1990** Under-5 mortality rate (deaths per 1,000 live births) in 1990.

**under5\_mortality\_2000** Under-5 mortality rate (deaths per 1,000 live births) in 2000.

**under5\_mortality\_2018** Under-5 mortality rate (deaths per 1,000 live births) in 2018.

**under5\_reduction** Annual rate of reduction in under-5 mortality rate (%)2000–2018.

**under5\_mortality\_2018\_male** Under-5 mortality rate male (deaths per 1,000 live births) 2018.

**under5\_mortality\_2018\_female** Under-5 mortality rate female (deaths per 1,000 live births) 2018.

**infant\_mortality\_1990** Infant mortality rate (deaths per 1,000 live births) 1990

**infant\_mortality\_2018** Infant mortality rate (deaths per 1,000 live births) 2018

**neonatal\_mortality\_1990** Neonatal mortality rate (deaths per 1,000 live births) 1990.

**neonatal\_mortality\_2000** Neonatal mortality rate (deaths per 1,000 live births) 2000.

**neonatal\_mortality\_2018** Neonatal mortality rate (deaths per 1,000 live births) 2018.

**prob\_dying\_age5to14\_1990** Probability of dying among children aged 5–14 (deaths per 1,000 children aged 5) 1990.

**prob\_dying\_age5to14\_2018** Probability of dying among children aged 5–14 (deaths per 1,000 children aged 5) 2018.

**under5\_deaths\_2018** Annual number of under-5 deaths (thousands) 2018.

**neonatal\_deaths\_2018** Annual number of neonatal deaths (thousands) 2018.

**neonatal\_deaths\_percent\_under5** Neonatal deaths as proportion of all under-5 deaths (%) 2018.

**age5to14\_deaths\_2018** Number of deaths among children aged 5–14 (thousands) 2018.

## Source

United Nations Children’s Emergency Fund (UNICEF)

## Examples

```
library(dplyr)
library(ggplot2)

# List countries and areas whose children aged 5 and under have a higher probability of dying in
# 2018 than they did in 1990
sowc_child_mortality |>
  mutate(decrease_prob_dying = prob_dying_age5to14_1990 - prob_dying_age5to14_2018) |>
  select(countries_and_areas, decrease_prob_dying) |>
  filter(decrease_prob_dying < 0) |>
  arrange(decrease_prob_dying)

# List countries and areas and their relative rank for neonatal mortality in 2018
sowc_child_mortality |>
  mutate(rank = round(rank(-neonatal_mortality_2018))) |>
  select(countries_and_areas, rank, neonatal_mortality_2018) |>
  arrange(rank)
```

---

sowc_demographics	<i>SOWC Demographics Data.</i>
-------------------	--------------------------------

---

## Description

Demographic data from UNICEF’s State of the World’s Children 2019 Statistical Tables.

## Usage

```
sowc_demographics
```

## Format

A data frame with 202 rows and 18 variables.

**countries\_and\_areas** Country or area name.

**total\_pop\_2018** Population in 2018 in thousands.

**under18\_pop\_2018** Population under age 18 in 2018 in thousands.

**under5\_pop\_2018** Population under age 5 in 2018 in thousands.

**pop\_growth\_rate\_2018** Rate at which population is growing in 2018.

**pop\_growth\_rate\_2030** Rate at which population is estimated to grow in 2030.

**births\_2018** Number of births in 2018 in thousands.

**fertility\_2018** Number of live births per woman in 2018. A total fertility level of 2.1 is called replacement level and represents a level at which the population would remain the same size.

**life\_expectancy\_1970** Life expectancy at birth in 1970.

**life\_expectancy\_2000** Life expectancy at birth in 2000.

**life\_expectancy\_2018** Life expectancy at birth in 2018.

**dependency\_ratio\_total** The ratio of the not-working-age population to the working-age population of 15 - 64 years.

**dependency\_ratio\_child** The ratio of the under 15 population to the working-age population of 15 - 64 years.

**dependency\_ratio\_oldage** The ratio of the over 64 population to the working-age population of 15 - 64 years.

**percent\_urban\_2018** Percent of population living in urban areas.

**pop\_urban\_growth\_rate\_2018** Annual urban population growth rate from 2000 to 2018.

**pop\_urban\_growth\_rate\_2030** Estimated annual urban population growth rate from 2018 to 2030.

**migration\_rate** Net migration rate per 1000 population from 2015 to 2020.

## Source

United Nations Children's Emergency Fund (UNICEF)

## Examples

```
library(dplyr)
library(ggplot2)

# List countries and areas' life expectancy, ordered by rank of life expectancy in 2018
sowc_demographics |>
  mutate(life_expectancy_change = life_expectancy_2018 - life_expectancy_1970) |>
  mutate(rank_life_expectancy = round(rank(-life_expectancy_2018), 0)) |>
  select(
    countries_and_areas, rank_life_expectancy, life_expectancy_2018,
    life_expectancy_change
  ) |>
  arrange(rank_life_expectancy)
```

```
# List countries and areas' migration rate and population, ordered by rank of migration rate
sowc_demographics |>
  mutate(rank = round(rank(migration_rate))) |>
  mutate(population_millions = total_pop_2018 / 1000) |>
  select(countries_and_areas, rank, migration_rate, population_millions) |>
  arrange(rank)

# Scatterplot of life expectancy v population in 2018
ggplot(sowc_demographics, aes(life_expectancy_1970, life_expectancy_2018, size = total_pop_2018)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Life Expectancy",
    subtitle = "1970 v. 2018",
    x = "Life Expectancy in 1970",
    y = "Life Expectancy in 2018",
    size = "2018 Total Population"
  )
```

---

sowc\_maternal\_newborn *SOWC Maternal and Newborn Health Data.*

---

## Description

Data from UNICEF's State of the World's Children 2019 Statistical Tables.

## Usage

```
sowc_maternal_newborn
```

## Format

A data frame with 202 rows and 18 variables.

**countries\_and\_areas** Country or area name.

**life\_expectancy\_female** Life expectancy: female in 2018.

**family\_planning\_1549** Demand for family planning satisfied with modern methods (%) 2013–2018  
Women aged 15 to 49.

**family\_planning\_1519** Demand for family planning satisfied with modern methods (%) 2013–2018  
Women aged 15 to 19.

**adolescent\_birth\_rate** Adolescent birth rate 2013 to 2018.

**births\_age\_18** Births by age 18 (%) 2013 to 2018.

**antenatal\_care\_1** Antenatal care (%) 2013 to 2018 At least one visit.

**antenatal\_care\_4\_1549** Antenatal care (%) 2013 to 2018 At least four visits Women aged 15 to 49.

**antenatal\_care\_4\_1519** Antenatal care (%) 2013 to 2018 At least four visits Women aged 15 to 19.

**delivery\_care\_attendant\_1549** Delivery care (%) 2013 to 2018 Skilled birth attendant Women aged 15 to 49.

**delivery\_care\_attendant\_1519** Delivery care (%) 2013 to 2018 Skilled birth attendant Women aged 15 to 19.

**delivery\_care\_institutional** Delivery care (%) 2013 to 2018 Institutional delivery.

**c\_section** Delivery care (%) 2013–2018 C-section.

**postnatal\_health\_newborns** Postnatal health check(%) 2013 to 2018 For newborns.

**postnatal\_health\_mothers** Postnatal health check(%) 2013 to 2018 For mothers.

**maternal\_deaths\_2017** Maternal mortality 2017 Number of maternal deaths.

**maternal\_mortality\_ratio\_2017** Maternal mortality 2017 Maternal Mortality Ratio.

**risk\_maternal\_death\_2017** Maternal mortality 2017 Lifetime risk of maternal death (1 in X).

### Source

United Nations Children's Emergency Fund (UNICEF)

### Examples

```
library(dplyr)
library(ggplot2)

# List countries and lifetime risk of maternal death (1 in X), ranked
sowc_maternal_newborn |>
  mutate(rank = round(rank(risk_maternal_death_2017), 0)) |>
  select(countries_and_areas, rank, risk_maternal_death_2017) |>
  arrange(rank)

# Graph scatterplot of Maternal Mortality Ratio 2017 and Antenatal Care 4+ Visits %
sowc_maternal_newborn |>
  select(antenatal_care_4_1549, maternal_mortality_ratio_2017) |>
  remove_missing(na.rm = TRUE) |>
  ggplot(aes(antenatal_care_4_1549, maternal_mortality_ratio_2017)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Antenatal Care and Mortality",
    x = "Antenatal Care 4+ visits %",
    y = "Maternal Mortality Ratio"
  )
```

### Description

Fifty companies were randomly sampled from the 500 companies in the S&P 500, and their financial information was collected on March 8, 2012.



**Usage**

sp500

**Format**

A data frame with 50 observations on the following 12 variables.

**market\_cap** Total value of all company shares, in millions of dollars.

**stock** The name of the stock (e.g. AAPL for Apple).

**ent\_value** Enterprise value, which is an alternative to market cap that also accounts for things like cash and debt, in millions of dollars.

**trail\_pe** The market cap divided by the earnings (profits) over the last year.

**forward\_pe** The market cap divided by the forecasted earnings (profits) over the next year.

**ev\_over\_rev** Enterprise value divided by the company's revenue.

**profit\_margin** Percent of earnings that are profits.

**revenue** Revenue, in millions of dollars.

**growth** Quarterly revenue growth (year over year), in millions of dollars.

**earn\_before** Earnings before interest, taxes, depreciation, and amortization, in millions of dollars.

**cash** Total cash, in millions of dollars.

**debt** Total debt, in millions of dollars.

**Source**

Yahoo! Finance, retrieved 2012-03-08.

**Examples**

```
library(ggplot2)

ggplot(sp500, aes(x = ent_value, y = earn_before)) +
  geom_point() +
  labs(x = "Enterprise value", y = "Earnings")

ggplot(sp500, aes(x = ev_over_rev, y = forward_pe)) +
  geom_point() +
  labs(
    x = "Enterprise value / revenue, logged",
    y = "Market cap / forecasted earnings, logged"
  )

ggplot(sp500, aes(x = ent_value, y = earn_before)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Enterprise value", y = "Earnings")

ggplot(sp500, aes(x = ev_over_rev, y = forward_pe)) +
```

```

geom_point() +
scale_x_log10() +
scale_y_log10() +
labs(
  x = "Enterprise value / revenue, logged",
  y = "Market cap / forecasted earnings, logged"
)

```

---

sp500\_1950\_2018

*Daily observations for the S&P 500*


---

### Description

Data runs from 1950 to near the end of 2018.

### Usage

```
sp500_1950_2018
```

### Format

A data frame with 17346 observations on the following 7 variables.

**Date** Date of the form "YYYY-MM-DD".

**Open** Opening price.

**High** Highest price of the day.

**Low** Lowest price of the day.

**Close** Closing price of the day.

**Adj.Close** Adjusted price at close after accounting for dividends paid out.

**Volume** Trading volume.

### Source

Yahoo! Finance

### Examples

```

data(sp500_1950_2018)
sp500.ten.years <- subset(
  sp500_1950_2018,
  "2009-01-01" <= as.Date(Date) & as.Date(Date) <= "2018-12-31"
)
d <- diff(sp500.ten.years$Adj.Close)
mean(d > 0)

```

---

sp500\_seq

*S&P 500 stock data*


---

**Description**

Daily stock returns from the S&P500 for 1990-2011 can be used to assess whether stock activity each day is independent of the stock's behavior on previous days. We label each day as Up or Down (D) depending on whether the market was up or down that day. For example, consider the following changes in price, their new labels of up and down, and then the number of days that must be observed before each Up day.

**Usage**

```
sp500_seq
```

**Format**

A data frame with 2948 observations on the following variable.

**race** a factor with levels 1, 2, 3, 4, 5, 6, and 7+

**Source**

[Google Finance](#).

**Examples**

```
sp500_seq
```

---

speed\_gender\_height

*Speed, gender, and height of 1325 students*


---

**Description**

1,325 UCLA students were asked to fill out a survey where they were asked about their height, fastest speed they have ever driven, and gender.

**Usage**

```
speed_gender_height
```

**Format**

A data frame with 1325 observations on the following 3 variables.

**speed** a numeric vector

**gender** a factor with levels female and male

**height** a numeric vector

**Examples**

```
speed_gender_height
```

---

ssd_speed	<i>SSD read and write speeds</i>
-----------	----------------------------------

---

**Description**

User submitted data on 1TB solid state drives (SSD).

**Usage**

```
ssd_speed
```

**Format**

A data frame with 54 rows and 7 variables.

**brand** Brand name of the drive.

**model** Model name of the drive.

**samples** Number of user submitted benchmarks.

**form\_factor** Physical form of the drive with levels 2.5, m.2, and mSATA.

**nvme** If a drive uses the *nvme* protocol this value is 1, 0 if it does not.

**read** Average read speed from user benchmarks in MB/s.

**write** Average write speed from user benchmarks in MB/s.

**Source**

[UserBenchmark](#), retrieved September 1, 2020.

**Examples**

```
library(ggplot2)
library(dplyr)

ssd_speed |>
  count(form_factor)

ssd_speed |>
  filter(form_factor != "mSATA") |>
  ggplot(aes(x = read, y = write, color = form_factor)) +
  geom_point() +
  labs(
    title = "Average read vs. write speed of SSDs",
    x = "Read speed (MB/s)",
    y = "Write speed (MB/s)"
  ) +
  facet_wrap(~form_factor, ncol = 1, scales = "free") +
  guides(color = FALSE)
```

---

starbucks*Starbucks nutrition*

---

**Description**

Nutrition facts for several Starbucks food items

**Usage**

```
starbucks
```

**Format**

A data frame with 77 observations on the following 7 variables.

**item** Food item.

**calories** Calories.

**fat** a numeric vector

**carb** a numeric vector

**fiber** a numeric vector

**protein** a numeric vector

**type** a factor with levels bakery, bistro box, hot breakfast, parfait, petite, salad, and sandwich

**Source**

<https://www.starbucks.com/menu>, retrieved 2011-03-10.

**Examples**

```
starbucks
```

---

stats\_scores*Final exam scores for twenty students*

---

**Description**

Scores range from 57 to 94.

**Usage**

```
stats_scores
```

**Format**

A data frame with 20 observations on the following variable.

**scores** a numeric vector

**Examples**

```
stats_scores
```

---

```
stem_cell
```

---

*Embryonic stem cells to treat heart attack (in sheep)*

---

**Description**

Does treatment using embryonic stem cells (ESCs) help improve heart function following a heart attack? Each sheep in the study was randomly assigned to the ESC or control group, and the change in their hearts' pumping capacity was measured in the study. A positive value corresponds to increased pumping capacity, which generally suggests a stronger recovery.

**Usage**

```
stem_cell
```

**Format**

A data frame with 18 observations on the following 3 variables.

**trmt** a factor with levels ctrl esc

**before** a numeric vector

**after** a numeric vector

**Source**

[doi:10.1016/S01406736\(05\)673801](https://doi.org/10.1016/S01406736(05)673801)

**Examples**

```
stem_cell
```

---

stent30*Stents for the treatment of stroke*

---

**Description**

An experiment that studies effectiveness of stents in treating patients at risk of stroke with some unexpected results. `stent30` represents the results 30 days after stroke and `stent365` represents the results 365 days after stroke.

**Usage**

```
stent30
```

**Format**

A data frame with 451 observations on the following 2 variables.

**group** a factor with levels control and treatment

**outcome** a factor with levels no event and stroke

**Source**

Chimowitz MI, Lynn MJ, Derdeyn CP, et al. 2011. Stenting versus Aggressive Medical Therapy for Intracranial Arterial Stenosis. *New England Journal of Medicine* 365:993- 1003. doi:10.1056/NEJMoal105335. NY Times article reporting on the study: <https://www.nytimes.com/2011/09/08/health/research/08stent.html>.

**Examples**

```
# 30-day results
table(stent30)

# 365-day results
table(stent365)
```

---

stocks\_18*Monthly Returns for a few stocks*

---

**Description**

Monthly return data for a few stocks, which covers stock prices from November 2015 through October 2018.

**Usage**

```
stocks_18
```

**Format**

A data frame with 36 observations on the following 3 variables.

**date** First day of the month corresponding to the returns.

**goog** Google stock price change.

**cat** Caterpillar stock price change.

**xom** Exxon Mobil stock price change.

**Source**

Yahoo! Finance, direct download.

**Examples**

```
d <- stocks_18
dim(d)
apply(d[, 2:3], 2, mean)
apply(d[, 2:3], 2, sd)
```

---

student\_housing

---

*Community college housing (simulated data, 2015)*


---

**Description**

These are simulated data and intended to represent housing prices of students at a college.

**Usage**

```
student_housing
```

**Format**

A data frame with 175 observations on the following variable.

**price** Monthly housing price, simulated.

**Examples**

```
set.seed(5)
generate_student_housing <- data.frame(
  price = round(rnorm(175, 515, 65) + exp(rnorm(175, 4.2, 1)))
)
hist(student_housing$price, 20)
t.test(student_housing$price)
mean(student_housing$price)
sd(student_housing$price)
identical(student_housing, generate_student_housing)
```



---

student_sleep	<i>Sleep for 110 students (simulated)</i>
---------------	---

---

**Description**

A simulated dataset for how much 110 college students each slept in a single night.

**Usage**

```
student_sleep
```

**Format**

A data frame with 110 observations on the following variable.

**hours** Number of hours slept by this student (simulated).

**Source**

Simulated data.

**Examples**

```
set.seed(2)
x <- exp(c(
  rnorm(100, log(7.5), 0.15),
  rnorm(10, log(10), 0.196)
))
x <- round(x - mean(x) + 7.42, 2)

identical(x, student_sleep$hours)
```

---

sulphinpyrazone	<i>Treating heart attacks</i>
-----------------	-------------------------------

---

**Description**

Experiment data for studying the efficacy of treating patients who have had a heart attack with Sulphinpyrazone.

**Usage**

```
sulphinpyrazone
```

**Format**

A data frame with 1475 observations on the following 2 variables.

**group** a factor with levels control treatment

**outcome** a factor with levels died lived

**Source**

Anturane Reinfarction Trial Research Group. 1980. Sulfipyrazone in the prevention of sudden death after myocardial infarction. New England Journal of Medicine 302(5):250-256.

**Examples**

sulphinpyrazone

---

supreme_court	<i>Supreme Court approval rating</i>
---------------	--------------------------------------

---

**Description**

Summary of a random survey of 976 people.

**Usage**

supreme\_court

**Format**

A data frame with 976 observations on the following variable.

**answer** a factor with levels approve and not

**Source**

<https://www.nytimes.com/2012/06/08/us/politics/44-percent-of-americans-approve-of-supreme-court-in.html>

**Examples**

supreme\_court

teacher

*Teacher Salaries in St. Louis, Michigan***Description**

This dataset contains teacher salaries from 2009-2010 for 71 teachers employed by the St. Louis Public School in Michigan, as well as several covariates.

**Usage**

```
teacher
```

**Format**

A data frame with 71 observations on the following 8 variables.

**id** Identification code for each teacher, assigned randomly.

**degree** Highest educational degree attained: BA (bachelor's degree) or MA (master's degree).

**fte** Full-time enrollment status: full-time 1 or part-time 0.5.

**years** Number of years employed by the school district.

**base** Base annual salary, in dollars.

**fica** Amount paid into Social Security and Medicare per year through the Federal Insurance Contribution Act (FICA), in dollars.

**retirement** Amount paid into the retirement fund of the teacher per year, in dollars.

**total** Total annual salary of the teacher, resulting from the sum of base salary + fica + retirement, in dollars.

**Source**

Originally posted on SODA Developers ([dev.socrata.com/data](http://dev.socrata.com/data)), removed in 2020.

**Examples**

```
library(ggplot2)

# Salary and education level
ggplot(teacher, aes(x = degree, y = base)) +
  geom_boxplot() +
  labs(
    x = "Highest educational degree attained",
    y = "Base annual salary, in $",
    color = "Degree",
    title = "Salary and education level"
  )

# Salary and years of employment
ggplot(teacher, aes(x = years, y = base, color = degree)) +
```

```
geom_point() +
labs(
  x = "Number of years employed by the school district",
  y = "Base annual salary, in $",
  color = "Degree",
  title = "Salary and years of employment"
)
```

---

textbooks

*Textbook data for UCLA Bookstore and Amazon*


---

### Description

A random sample was taken of nearly 10\ textbook for each course was identified, and its new price at the UCLA Bookstore and on Amazon.com were recorded.

### Usage

```
textbooks
```

### Format

A data frame with 73 observations on the following 7 variables.

**dept\_abbr** Course department (abbreviated).

**course** Course number.

**isbn** Book ISBN.

**ucla\_new** New price at the UCLA Bookstore.

**amaz\_new** New price on Amazon.com.

**more** Whether additional books were required for the course (Y means "yes, additional books were required").

**diff** The UCLA Bookstore price minus the Amazon.com price for each book.

### Details

The sample represents only courses where textbooks were listed online through UCLA Bookstore's website. The most expensive textbook was selected based on the UCLA Bookstore price, which may insert bias into the data; for this reason, it may be beneficial to analyze only the data where more is "N".

### Source

Collected by David Diez.

**Examples**

```
library(ggplot2)

ggplot(textbooks, aes(x = diff)) +
  geom_histogram(binwidth = 5)

t.test(textbooks$diff)
```

---

thanksgiving_spend	<i>Thanksgiving spending, simulated based on Gallup poll.</i>
--------------------	---

---

**Description**

This entry gives simulated spending data for Americans during Thanksgiving in 2009 based on findings of a Gallup poll.

**Usage**

```
thanksgiving_spend
```

**Format**

A data frame with 436 observations on the following 1 variable.

**spending** Amount of spending, in US dollars.

**Examples**

```
library(ggplot2)

ggplot(thanksgiving_spend, aes(x = spending)) +
  geom_histogram(binwidth = 20)
```

---

tips	<i>Tip data</i>
------	-----------------

---

**Description**

A simulated dataset of tips over a few weeks on a couple days per week. Each tip is associated with a single group, which may include several bills and tables (i.e. groups paid in one lump sum in simulations).

**Usage**

```
tips
```

**Format**

A data frame with 95 observations on the following 5 variables.

**week** Week number.

**day** Day, either Friday or Tuesday.

**n\_peop** Number of people associated with the group.

**bill** Total bill for the group.

**tip** Total tip from the group.

**Details**

This dataset was built using simulations of tables, then bills, then tips based on the bills. Large groups were assumed to only pay the gratuity, which is evident in the data. Tips were set to be plausible round values; they were often (but not always) rounded to dollars, quarters, etc.

**Source**

Simulated dataset.

**Examples**

```
library(ggplot2)

ggplot(tips, aes(x = day, y = tip)) +
  geom_boxplot()

ggplot(tips, aes(x = tip, fill = factor(week))) +
  geom_density(alpha = 0.5) +
  labs(x = "Tip", y = "Density", fill = "Week")

ggplot(tips, aes(x = tip)) +
  geom_dotplot()

ggplot(tips, aes(x = tip, fill = factor(day))) +
  geom_density(alpha = 0.5) +
  labs(x = "Tip", y = "Density", fill = "Day")
```

---

toohey

*Simulated polling dataset*

---

**Description**

Simulated data for a fake political candidate.

**Usage**

toohey

**Format**

A data frame with 500 observations on the following variable.

**vote\_for** a factor with levels no yes

**Examples**

toohey

---

tourism	<i>Turkey tourism</i>
---------	-----------------------

---

**Description**

Summary of tourism in Turkey.

**Usage**

tourism

**Format**

A data frame with 47 observations on the following 3 variables.

**year** a numeric vector

**visitor\_count\_tho** a numeric vector

**tourist\_spending** a numeric vector

**Source**

Association of Turkish Travel Agencies, Foreign Visitors Figure & Tourist Spendings By Years.

[http://www.tursab.org.tr/en/statistics/foreign-visitors-figure-tourist-spendings-by-years\\_1083.html](http://www.tursab.org.tr/en/statistics/foreign-visitors-figure-tourist-spendings-by-years_1083.html)

**Examples**

tourism

---

toy_anova	<i>Simulated dataset for ANOVA</i>
-----------	------------------------------------

---

**Description**

Simulated dataset for getting a better understanding of intuition that ANOVA is based off of.

**Usage**

toy\_anova

**Format**

A data frame with 70 observations on the following 3 variables.

**group** a factor with levels I II III

**outcome** a numeric vector

**Examples**

toy\_anova

---

transplant	<i>Transplant consultant success rate (fake data)</i>
------------	---

---

**Description**

Summarizing whether there was or was not a complication for 62 patients who used a particular medical consultant.

**Usage**

transplant

**Format**

A data frame with 62 observations on the following variable.

**outcome** a factor with levels complications okay

**Examples**

transplant



---

treeDiag	<i>Construct tree diagrams</i>
----------	--------------------------------

---

## Description

Construct beautiful tree diagrams

## Usage

```
treeDiag(
  main,
  p1,
  p2,
  out1 = c("Yes", "No"),
  out2 = c("Yes", "No"),
  textwd = 0.15,
  solwd = 0.2,
  SBS = c(TRUE, TRUE),
  showSol = TRUE,
  solSub = NULL,
  digits = 4,
  textadj = 0.015,
  cex.main = 1.3,
  col.main = "#999999",
  showWork = FALSE
)
```

## Arguments

main	Character vector with two variable names, descriptions, or questions
p1	Vector of probabilities for the primary branches
p2	List for the secondary branches, where each list item should be a numerical vector of probabilities corresponding to the primary branches of p1
out1	Character vector of the outcomes corresponding to the primary branches
out2	Character vector of the outcomes corresponding to the secondary branches
textwd	The width provided for text with a default of 0.15
solwd	The width provided for the solution with a default of 0.2
SBS	A boolean vector indicating whether to place text and probability side-by-side for the primary and secondary branches
showSol	Boolean indicating whether to show the solution in the tree diagram
solSub	An optional list of vectors corresponding to p2 to list alternative text or solutions
digits	The number of digits to show in the solution
textadj	Vertical adjustment of text

cex.main	Size of main in the plot
col.main	Color of main in the plot
showWork	Whether work should be shown for the solutions

**Author(s)**

David Diez, Christopher Barr

**Examples**

```
treeDiag(
  c("Flight on time?", "Luggage on time?"),
  c(0.8, 0.2), list(c(0.97, 0.03), c(0.15, 0.85))
)
treeDiag(c("Breakfast?", "Go to class"), c(.4, .6),
  list(c(0.4, 0.36, 0.34), c(0.6, 0.3, 0.1)), c("Yes", "No"),
  c("Statistics", "English", "Sociology"),
  showWork = TRUE
)
treeDiag(
  c("Breakfast?", "Go to class"), c(0.4, 0.11, 0.49),
  list(c(0.4, 0.36, 0.24), c(0.6, 0.3, 0.1), c(0.1, 0.4, 0.5)),
  c("one", "two", "three"), c("Statistics", "English", "Sociology")
)
treeDiag(c("Dow Jones rise?", "NASDAQ rise?"),
  c(0.53, 0.47), list(c(0.75, 0.25), c(0.72, 0.28)),
  solSub = list(c("(a)", "(b)"), c("(c)", "(d)")), solwd = 0.08
)
```

---

twins

---

*twins*


---

**Description**

A data frame containing data collected in the mid 20th century by Cyril Burt from a study tracked down identical twins who were separated at birth: one child was raised in the home of their biological parents and the other in a foster home. In an attempt to answer the question of whether intelligence is the result of nature or nurture, both children were given IQ tests.

**Usage**

```
twins
```

**Format**

A data frame with 27 observations on the following 2 variables.

**foster** IQ score of the twin raised by Foster parents.

**biological** IQ score of the twin raised by Biological parents.

**Examples**

```
library(ggplot2)
library(dplyr)
library(tidyr)

plot_data <- twins |>
  pivot_longer(cols = c(foster, biological), names_to = "twin", values_to = "iq")

ggplot(plot_data, aes(iq, fill = twin)) +
  geom_histogram(color = "white", binwidth = 5) +
  facet_wrap(~twin) +
  theme_minimal() +
  labs(
    title = "IQ of identical twins",
    subtitle = "Separated at birth",
    x = "IQ",
    y = "Count",
    fill = ""
  )
```

ucla\_f18

*UCLA courses in Fall 2018***Description**

List of all courses at UCLA during Fall 2018.

**Usage**

```
ucla_f18
```

**Format**

A data frame with 3950 observations on the following 14 variables.

**year** Year the course was offered

**term** Term the course was offered

**subject** Subject

**subject\_abbr** Subject abbreviation, if any

**course** Course name

**course\_num** Course number, complete

**course\_numeric** Course number, numeric only

**seminar** Boolean for if this is a seminar course

**ind\_study** Boolean for if this is some form of independent study

**apprenticeship** Boolean for if this is an apprenticeship

**internship** Boolean for if this is an internship  
**honors\_contracts** Boolean for if this is an honors contracts course  
**laboratory** Boolean for if this is a lab  
**special\_topic** Boolean for if this is any of the special types of courses listed

### Source

<https://sa.ucla.edu/ro/public/soc>, retrieved 2018-11-22.

### Examples

```
nrow(ucla_f18)
table(ucla_f18$special_topic)
subset(ucla_f18, is.na(course_numeric))
table(subset(ucla_f18, !special_topic)$course_numeric < 100)
elig_courses <-
  subset(ucla_f18, !special_topic & course_numeric < 100)
set.seed(1)
ucla_textbooks_f18 <-
  elig_courses[sample(nrow(elig_courses), 100), ]
tmp <- order(
  ucla_textbooks_f18$subject,
  ucla_textbooks_f18$course_numeric
)
ucla_textbooks_f18 <- ucla_textbooks_f18[tmp, ]
rownames(ucla_textbooks_f18) <- NULL
head(ucla_textbooks_f18)
```

---

ucla_textbooks_f18	<i>Sample of UCLA course textbooks for Fall 2018</i>
--------------------	--

---

### Description

A sample of courses were collected from UCLA from Fall 2018, and the corresponding textbook prices were collected from the UCLA bookstore and also from Amazon.

### Usage

```
ucla_textbooks_f18
```

### Format

A data frame with 201 observations on the following 20 variables.

**year** Year the course was offered  
**term** Term the course was offered  
**subject** Subject

**subject\_abbr** Subject abbreviation, if any  
**course** Course name  
**course\_num** Course number, complete  
**course\_numeric** Course number, numeric only  
**seminar** Boolean for if this is a seminar course.  
**ind\_study** Boolean for if this is some form of independent study  
**apprenticeship** Boolean for if this is an apprenticeship  
**internship** Boolean for if this is an internship  
**honors\_contracts** Boolean for if this is an honors contracts course  
**laboratory** Boolean for if this is a lab  
**special\_topic** Boolean for if this is any of the special types of courses listed  
**textbook\_isbn** Textbook ISBN  
**bookstore\_new** New price at the UCLA bookstore  
**bookstore\_used** Used price at the UCLA bookstore  
**amazon\_new** New price sold by Amazon  
**amazon\_used** Used price sold by Amazon  
**notes** Any relevant notes

## Details

A past dataset was collected from UCLA courses in Spring 2010, and Amazon at that time was found to be almost uniformly lower than those of the UCLA bookstore's. Now in 2018, the UCLA bookstore is about even with Amazon on the vast majority of titles, and there is no statistical difference in the sample data.

The most expensive book required for the course was generally used.

The reason why we advocate for using raw amount differences instead of percent differences is that a 20\ to a 20\ price difference on low-priced books would balance numerically (but not in a practical sense) a moderate but important price difference on more expensive books. So while this tends to result in a bit less sensitivity in detecting *some* effect, we believe the absolute difference compares prices in a more meaningful way.

Used prices contain the shipping cost but do not contain tax. The used prices are a more nuanced comparison, since these are all 3rd party sellers. Amazon is often more a marketplace than a retail site at this point, and many people buy from 3rd party sellers on Amazon now without realizing it. The relationship Amazon has with 3rd party sellers is also challenging. Given the frequently changing dynamics in this space, we don't think any analysis here will be very reliable for long term insights since products from these sellers changes frequently in quantity and price. For this reason, we focus only on new books sold directly by Amazon in our comparison. In a future round of data collection, it may be interesting to explore whether the dynamics have changed in the used market.

## Source

<https://sa.ucla.edu/ro/public/soc>

<https://ucla.verbacompare.com>

<https://www.amazon.com>

**See Also**

[textbooks](#), [ucla\\_f18](#)

**Examples**

```
library(ggplot2)
library(dplyr)

ggplot(ucla_textbooks_f18, aes(x = bookstore_new, y = amazon_new)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "orange") +
  labs(
    x = "UCLA Bookstore price", y = "Amazon price",
    title = "Amazon vs. UCLA Bookstore prices of new textbooks",
    subtitle = "Orange line represents y = x"
  )

# The following outliers were double checked for accuracy
ucla_textbooks_f18_with_diff <- ucla_textbooks_f18 |>
  mutate(diff = bookstore_new - amazon_new)

ucla_textbooks_f18_with_diff |>
  filter(diff > 20 | diff < -20)

# Distribution of price differences
ggplot(ucla_textbooks_f18_with_diff, aes(x = diff)) +
  geom_histogram(binwidth = 5)

# t-test of price differences
t.test(ucla_textbooks_f18_with_diff$diff)
```

---

 ukdemo

*United Kingdom Demographic Data*


---

**Description**

This dataset comes from the Guardian's Data Blog and includes five financial demographic variables.

**Usage**

```
ukdemo
```

**Format**

A data frame with 12 observations on the following 6 variables.

**region** Region in the United Kingdom

**debt** Average regional debt, not including mortgages, in pounds

**unemployment** Percent unemployment

**house** Average house price, in pounds

**pay** Average hourly pay, in pounds

**rpi** Retail price index, which is standardized to 100 for the entire UK, and lower index scores correspond to lower prices

## Source

The data was described in the Guardian Data Blog: <https://www.theguardian.com/news/datablog/interactive/2011/oct/27/debt-money-expert-facts>, retrieved 2011-11-01.

## References

Guardian Data Blog

## Examples

```
library(ggplot2)

ggplot(ukdemo, aes(x = pay, y = rpi)) +
  geom_point() +
  labs(x = "Average hourly pay", y = "Retail price index")
```

---

unempl

*Annual unemployment since 1890*

---

## Description

A compilation of two datasets that provides an estimate of unemployment from 1890 to 2010.

## Usage

```
unempl
```

## Format

A data frame with 121 observations on the following 3 variables.

**year** Year

**unemp** Unemployment rate, in percent

**us\_data** 1 if from the Bureau of Labor Statistics, 0 otherwise

## Source

The data are from Wikipedia at the following URL accessed on November 1st, 2010:

[https://en.wikipedia.org/wiki/File:US\\_Unemployment\\_1890-2009.gif](https://en.wikipedia.org/wiki/File:US_Unemployment_1890-2009.gif)

Below is a direct quote from Wikipedia describing the sources of the data:

Own work by Peace01234 Complete raw data are on Peace01234. 1930-2009 data are from Bureau of Labor Statistics (BLS), Employment status of the civilian noninstitutional population, 1940 to date retrieved on March 6, 2009 and February 12, 2010 from the BLS' FTP server. Data prior to 1948 are for persons age 14 and over. Data beginning in 1948 are for persons age 16 and over. See also "Historical Comparability" under the Household Data section of the Explanatory Notes at [https://www.bls.gov/cps/eetech\\_methods.pdf](https://www.bls.gov/cps/eetech_methods.pdf). 1890-1930 data are from Christina Romer (1986). "Spurious Volatility in Historical Unemployment Data", The Journal of Political Economy, 94(1): 1-37. 1930-1940 data are from Robert M. Coen (1973). "Labor Force and Unemployment in the 1920's and 1930's: A Re-Examination Based on Postwar Experience", The Review of Economics and Statistics, 55(1): 46-55. Unemployment data was only surveyed once each decade until 1940 when yearly surveys were begun. The yearly data estimates before 1940 are based on the decade surveys combined with other relevant surveys that were collected during those years. The methods are described in detail by Coen and Romer.

## Examples

```
# =====> Time Series Plot of Data <=====#
COL <- c("#DDEEBB", "#EEDDBB", "#BBDDEE", "#FFD5DD", "#FFC5CC")
plot(unempl$year, unempl$unemp, type = "n")
rect(0, -50, 3000, 100, col = "#E2E2E2")
rect(1914.5, -1000, 1918.9, 1000, col = COL[1], border = "#E2E2E2")
rect(1929, -1000, 1939, 1000, col = COL[2], border = "#E2E2E2")
rect(1939.7, -1000, 1945.6, 1000, col = COL[3], border = "#E2E2E2")
rect(1955.8, -1000, 1965.3, 1000, col = COL[4], border = "#E2E2E2")
rect(1965.3, -1000, 1975.4, 1000, col = COL[5], border = "#E2E2E2")
abline(h = seq(0, 50, 5), col = "#F8F8F8", lwd = 2)
abline(v = seq(1900, 2000, 20), col = "#FFFFFF", lwd = 1.3)
lines(unempl$year, unempl$unemp)
points(unempl$year, unempl$unemp, pch = 20)
legend("topright",
      fill = COL,
      c(
        "World War I", "Great Depression", "World War II",
        "Vietnam War Start", "Vietnam War Escalated"
      ),
      bg = "#FFFFFF", border = "#FFFFFF"
    )
```

---

unemploy\_pres

*President's party performance and unemployment rate*

---

## Description

Covers midterm elections.



**Usage**

```
unemploy_pres
```

**Format**

A data frame with 29 observations on the following 5 variables.

**year** Year.

**potus** The president in office.

**party** President's party.

**unemp** Unemployment rate.

**change** Change in House seats for the president's party.

**Source**

Wikipedia.

**Examples**

```
unemploy_pres
```

---

```
usb_admit
```

```
ucb_admit
```

---

**Description**

Data from a study carried out by the graduate Division of the University of California, Berkeley in the early 1970's to evaluate whether there was a sex bias in graduate admissions.

**Usage**

```
ucb_admit
```

**Format**

A data frame with 4526 observations on the following 3 variables.

**admit** Was the applicant admitted to the university?

**gender** Whether the applicant identified as male or female.

**department** What department did the applicant apply to, noted as A through F for confidentiality.

**Examples**

```
library(ggplot2)
library(dplyr)

plot_data <- ucb_admit |>
  count(dept, gender, admit)

ggplot(plot_data, aes(dept, n, fill = gender)) +
  geom_col(position = "dodge") +
  facet_wrap(~admit) +
  theme_minimal() +
  labs(
    title = "Does gender discrimination play a role in college admittance?",
    x = "Department",
    y = "Number of Students",
    fill = "Gender",
    caption = "Source: UC Berkeley, 1970's"
  )
```

---

us\_temperature

*US temperatures in 1950 and 2022*


---

**Description**

A representative set of monitoring locations were taken from NOAA data in 1950 and 2022 such that the locations are sampled roughly geographically across the continental US (the observations do not represent a random sample of geographical locations).

**Usage**

```
us_temperature
```

**Format**

A data frame with 18759 observations on the following 9 variables.

**location** Location of the NOAA weather station.

**station** Formal ID of the NOAA weather station.

**latitude** Latitude of the NOAA weather station.

**longitude** Longitude of the NOAA weather station.

**elevation** Elevation of the NOAA weather station.

**date** Date the measurement was taken (Y-m-d).

**tmax** Maximum daily temperature (Fahrenheit).

**tmin** Minimum daily temperature (Fahrenheit).

**year** Year of the measurement.

## Details

Please keep in mind that the data represent two annual snapshots, and a complete analysis would consider more than two years of data and a random or more complete sampling of weather stations across the US.

## Source

[NOAA Climate Data Online](#). Retrieved 23 September, 2023.

## Examples

```
library(dplyr)
library(ggplot2)
library(maps)

summarized_temp <- us_temperature |>
  group_by(station, year, latitude, longitude) |>
  summarize(tmax_med = median(tmax, na.rm = TRUE)) |>
  mutate(plot_shift = ifelse(year == "1950", 0, 1)) |>
  mutate(year = as.factor(year))

usa <- map_data("state")

ggplot(data = usa, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group), color = "black", fill = "white") +
  geom_point(
    data = summarized_temp,
    aes(
      x = longitude + plot_shift, y = latitude,
      color = tmax_med, shape = year
    )
  ) +
  scale_color_gradient(high = IMSCOL["red", 1], low = IMSCOL["yellow", 1]) +
  ggtitle("Median of the daily high temp, 1950 & 2022") +
  labs(
    x = "longitude",
    color = "median high temp"
  ) +
  guides(shape = guide_legend(override.aes = list(color = "black")))
```

---

winery\_cars

*Time Between Gondola Cars at Sterling Winery*


---

## Description

These times represent times between gondolas at Sterling Winery. The main take-away: there are 7 cars, as evidenced by the somewhat regular increases in splits between every 7 cars. The reason the times are slightly non-constant is that the gondolas come off the tracks, so times will change a little between each period.

**Usage**

```
winery_cars
```

**Format**

A data frame with 52 observations on the following 2 variables.

**obs\_number** The observation number, e.g. observation 3 was immediately preceded by observation 2.

**time\_until\_next** Time until this gondola car arrived since the last car had left.

**Details**

Important context: there was a sufficient line that people were leaving the winery.

So why is this data valuable? It indicates that the winery should add one more car since it has a lot of time wasted every 7th car. By adding another car, fewer visitors are likely to be turned away, resulting in increased revenue.

**Source**

In-person data collection by David Diez (OpenIntro) on 2013-07-04.

**Examples**

```
winery_cars$car_number <- rep(1:7, 10)[1:nrow(winery_cars)]
col <- COL[ifelse(winery_cars$car_number == 3, 4, 1)]
plot(winery_cars[, c("obs_number", "time_until_next")],
     col = col, pch = 19
)
plot(winery_cars$car_number, winery_cars$time_until_next,
     col = fadeColor(col, "88"), pch = 19
)
```

---

world\_pop

*World Population Data.*

---

**Description**

From World Bank, population 1960-2020

**Usage**

```
world_pop
```

**Format**

A data frame with 216 rows and 62 variables.

**country** Name of country.

**year\_1960** population in 1960.

**year\_1961** population in 1961.

**year\_1962** population in 1962.

**year\_1963** population in 1963.

**year\_1964** population in 1964.

**year\_1965** population in 1965.

**year\_1966** population in 1966.

**year\_1967** population in 1967.

**year\_1968** population in 1968.

**year\_1969** population in 1969.

**year\_1970** population in 1970.

**year\_1971** population in 1971.

**year\_1972** population in 1972.

**year\_1973** population in 1973.

**year\_1974** population in 1974.

**year\_1975** population in 1975.

**year\_1976** population in 1976.

**year\_1977** population in 1977.

**year\_1978** population in 1978.

**year\_1979** population in 1979.

**year\_1980** population in 1980.

**year\_1981** population in 1981.

**year\_1982** population in 1982.

**year\_1983** population in 1983.

**year\_1984** population in 1984.

**year\_1985** population in 1985.

**year\_1986** population in 1986.

**year\_1987** population in 1987.

**year\_1988** population in 1988.

**year\_1989** population in 1989.

**year\_1990** population in 1990.

**year\_1991** population in 1991.

**year\_1992** population in 1992.

**year\_1993** population in 1993.

**year\_1994** population in 1994.  
**year\_1995** population in 1995.  
**year\_1996** population in 1996.  
**year\_1997** population in 1997.  
**year\_1998** population in 1998.  
**year\_1999** population in 1999.  
**year\_2000** population in 2000.  
**year\_2001** population in 2001.  
**year\_2002** population in 2002.  
**year\_2003** population in 2003.  
**year\_2004** population in 2004.  
**year\_2005** population in 2005.  
**year\_2006** population in 2006.  
**year\_2007** population in 2007.  
**year\_2008** population in 2008.  
**year\_2009** population in 2009.  
**year\_2010** population in 2010.  
**year\_2011** population in 2011.  
**year\_2012** population in 2012.  
**year\_2013** population in 2013.  
**year\_2014** population in 2014.  
**year\_2015** population in 2015.  
**year\_2016** population in 2016.  
**year\_2017** population in 2017.  
**year\_2018** population in 2018.  
**year\_2019** population in 2019.  
**year\_2020** population in 2020.

### Source

[World Bank](#)

### Examples

```
library(dplyr)
library(ggplot2)
library(tidyr)

# List percentage of population change from 1960 to 2020
world_pop |>
  mutate(percent_change = round((year_2020 - year_1960) / year_2020 * 100, 2)) |>
  mutate(rank_pop_change = round(rank(-percent_change)), 0) |>
```

```

select(rank_pop_change, country, percent_change) |>
  arrange(rank_pop_change)

# Graph population in millions by decade for specified countries
world_pop |>
  select(
    country, year_1960, year_1970, year_1980, year_1990,
    year_2000, year_2010, year_2020
  ) |>
  filter(country %in% c("China", "India", "United States")) |>
  pivot_longer(
    cols = c(year_1960, year_1970, year_1980, year_1990, year_2000, year_2010, year_2020),
    names_to = "year",
    values_to = "population"
  ) |>
  mutate(year = as.numeric(gsub("year_", "", year))) |>
  ggplot(aes(year, population, color = country)) +
  geom_point() +
  geom_smooth(method = "loess", formula = "y ~ x") +
  labs(
    title = "Population",
    subtitle = "by Decade",
    x = "Year",
    y = "Population (in millions)",
    color = "Country"
  )

```

---

write_pkg_data	Create a CSV variant of .rda files
----------------	------------------------------------

---

## Description

The function should be run with a path to a package directory. It will then look through the data directory of the package, and for all datasets that are data frames, create CSV variants in a data-csv directory.

## Usage

```

write_pkg_data(
  pkg,
  dir = paste0("data-", out_type),
  overwrite = FALSE,
  out_type = c("csv", "tab", "R")
)

```

## Arguments

pkg	The R package where we'd like to generate CSVs of any data frames.
-----	--

<code>dir</code>	A character string representing the path to the folder. where the CSV files should be written. If no such directory exists, one will be created (recursively).
<code>overwrite</code>	Boolean to indicate if to overwrite any existing files that have conflicting names in the directory specified.
<code>out_type</code>	Format for the type of output as a CSV (" <code>csv</code> "), tab-delimited text file (" <code>tab</code> "), or the R code to generate the object (" <code>R</code> ").

### Examples

```
## Not run:
write_pkg_data("openintro")
list.files("data-csv")

## End(Not run)
```

---

xom

*Exxon Mobile stock data*


---

### Description

Monthly data covering 2006 through early 2014.

### Usage

```
xom
```

### Format

A data frame with 98 observations on the following 7 variables.

**date** Date.

**open** a numeric vector

**high** a numeric vector

**low** a numeric vector

**close** a numeric vector

**volume** a numeric vector

**adj\_close** a numeric vector

### Source

Yahoo! Finance.

### Examples

```
xom
```



---

yawn*Contagiousness of yawning*

---

**Description**

An experiment conducted by the MythBusters, a science entertainment TV program on the Discovery Channel, tested if a person can be subconsciously influenced into yawning if another person near them yawns. 50 people were randomly assigned to two groups: 34 to a group where a person near them yawned (treatment) and 16 to a group where there wasn't a person yawning near them (control).

**Usage**

yawn

**Format**

A data frame with 50 observations on the following 2 variables.

**result** a factor with levels not yawn yawn

**group** a factor with levels ctrl trmt

**Source**

MythBusters, Season 3, Episode 28.

**Examples**

yawn

---

yrbss*Youth Risk Behavior Surveillance System (YRBSS)*

---

**Description**

Select variables from YRBSS.

**Usage**

yrbss

**Format**

A data frame with 13583 observations on the following 13 variables.

**age** Age, in years.

**gender** Gender.

**grade** School grade.

**hispanic** Hispanic or not.

**race** Race / ethnicity.

**height** Height, in meters (3.28 feet per meter).

**weight** Weight, in kilograms (2.2 pounds per kilogram).

**helmet\_12m** How often did you wear a helmet when biking in the last 12 months?

**text\_while\_driving\_30d** How many days did you text while driving in the last 30 days?

**physically\_active\_7d** How many days were you physically active for 60+ minutes in the last 7 days?

**hours\_tv\_per\_school\_day** How many hours of TV do you typically watch on a school night?

**strength\_training\_7d** How many days did you do strength training (e.g. lift weights) in the last 7 days?

**school\_night\_hours\_sleep** How many hours of sleep do you typically get on a school night?

**Source**

CDC's Youth Risk Behavior Surveillance System (YRBSS)

**Examples**

```
table(yrbss$physically_active_7d)
```

---

yrbss\_samp

---

*Sample of Youth Risk Behavior Surveillance System (YRBSS)*


---

**Description**

A sample of the [yrbss](#) dataset.

**Usage**

```
yrbss_samp
```

**Format**

A data frame with 100 observations on the following 13 variables.

**age** Age, in years.

**gender** Gender.

**grade** School grade.

**hispanic** Hispanic or not.

**race** Race / ethnicity.

**height** Height, in meters (3.28 feet per meter).

**weight** Weight, in kilograms (2.2 pounds per kilogram).

**helmet\_12m** How often did you wear a helmet when biking in the last 12 months?

**text\_while\_driving\_30d** How many days did you text while driving in the last 30 days?

**physically\_active\_7d** How many days were you physically active for 60+ minutes in the last 7 days?

**hours\_tv\_per\_school\_day** How many hours of TV do you typically watch on a school night?

**strength\_training\_7d** How many days did you do strength training (e.g. lift weights) in the last 7 days?

**school\_night\_hours\_sleep** How many hours of sleep do you typically get on a school night?

**Source**

CDC's Youth Risk Behavior Surveillance System (YRBSS)

**Examples**

```
table(yrbss_samp$physically_active_7d)
```

# Index

- \* **500**
  - sp500, 256
- \* **Algebra**
  - ArrowLines, 16
  - CCP, 46
  - dlsegments, 70
  - lsegments, 160
- \* **Bayes**
  - treeDiag, 273
- \* **Cartesian**
  - CCP, 46
- \* **Conditional**
  - treeDiag, 273
- \* **Congress**
  - piracy, 212
- \* **Coordinate**
  - CCP, 46
- \* **Data**
  - makeTube, 163
- \* **Graphics**
  - myPDF, 188
- \* **Kernel**
  - makeTube, 163
- \* **LaTeX**
  - contTable, 59
- \* **Least**
  - makeTube, 163
- \* **Line**
  - ArrowLines, 16
  - dlsegments, 70
  - lsegments, 160
- \* **London**
  - london\_boroughs, 157
  - london\_murders, 158
- \* **Michigan**
  - teacher, 267
- \* **PDF**
  - myPDF, 188
- \* **PIPA**
  - piracy, 212
- \* **Plane**
  - CCP, 46
- \* **Plotting**
  - myPDF, 188
- \* **Regression**
  - makeTube, 163
- \* **SOPA**
  - piracy, 212
- \* **SP**
  - sp500, 256
- \* **Save**
  - myPDF, 188
- \* **Segment**
  - ArrowLines, 16
  - dlsegments, 70
  - lsegments, 160
- \* **Theorem**
  - treeDiag, 273
- \* **Tree**
  - treeDiag, 273
- \* **airplane**
  - birds, 30
- \* **axis**
  - buildAxis, 38
- \* **bird**
  - birds, 30
- \* **borough**
  - london\_boroughs, 157
- \* **categorical**
  - heart\_transplant, 118
- \* **college**
  - credits, 64
- \* **contingency**
  - heart\_transplant, 118
- \* **control**
  - buildAxis, 38
- \* **copyright**
  - piracy, 212

- \* **corpus**
  - ipo, 134
- \* **correlation**
  - gradestv, 114
- \* **country**
  - esi, 89
- \* **credits**
  - credits, 64
- \* **crime**
  - london\_murders, 158
- \* **customize**
  - buildAxis, 38
- \* **datasets**
  - absenteeism, 8
  - acs12, 9
  - age\_at\_mar, 10
  - ames, 11
  - ami\_occurrences, 13
  - antibiotics, 14
  - arbuthnot, 15
  - ask, 17
  - association, 18
  - assortive\_mating, 19
  - avandia, 20
  - babies, 23
  - babies\_crawl, 23
  - bac, 24
  - ball\_bearing, 25
  - bdims, 26
  - biontech\_adolescents, 29
  - birds, 30
  - births, 31
  - births14, 32
  - blizzard\_salary, 33
  - books, 34
  - burger, 41
  - cancer\_in\_dogs, 42
  - cards, 42
  - cars04, 43
  - cars93, 44
  - cchousing, 45
  - census, 47
  - cherry, 48
  - children\_gender\_stereo, 49
  - china, 50
  - cia\_factbook, 52
  - classdata, 53
  - cle\_sac, 54
  - climate70, 54
  - climber\_drugs, 55
  - coast\_starlight, 56
  - COL, 57
  - comics, 58
  - corr\_match, 60
  - country\_iso, 61
  - cpr, 62
  - cpu, 62
  - credits, 64
  - daycare\_fines, 65
  - diabetes2, 69
  - dream, 75
  - drone\_blades, 75
  - drug\_use, 76
  - duke\_forest, 77
  - earthquakes, 78
  - ebola\_survey, 79
  - elmhurst, 80
  - email, 81
  - email50, 83
  - env\_regulation, 85
  - epa2012, 86
  - epa2021, 87
  - esi, 89
  - ethanol, 91
  - evals, 92
  - exam\_grades, 93
  - exams, 93
  - exclusive\_relationship, 94
  - fact\_opinion, 95
  - family\_college, 98
  - fastfood, 99
  - fcid, 100
  - fheights, 100
  - fish\_age, 101
  - fish\_oil\_18, 102
  - flow\_rates, 103
  - friday, 104
  - full\_body\_scan, 105
  - gdp\_countries, 106
  - gear\_company, 107
  - gender\_discrimination, 107
  - get\_it\_dunn\_run, 108
  - gifted, 109
  - global\_warming\_pew, 110
  - goog, 111
  - gov\_poll, 111

gpa, 112  
gpa\_iq, 113  
gpa\_study\_hours, 113  
gradestv, 114  
gsearch, 115  
gss2010, 115  
gss\_wordsum\_class, 116  
health\_coverage, 117  
healthcare\_law\_survey, 117  
heart\_transplant, 118  
helium, 119  
helmet, 120  
hfi, 121  
house, 127  
housing, 129  
hsb2, 129  
husbands\_wives, 130  
immigration, 131  
IMSCOL, 132  
infmortrate, 132  
iowa, 133  
ipo, 134  
ipod, 135  
iran, 136  
jury, 137  
kobe\_basket, 138  
labor\_market\_discrimination, 138  
LAhomes, 142  
law\_resume, 143  
lecture\_learning, 144  
leg\_mari, 148  
lego\_population, 145  
lego\_sample, 147  
life\_exp, 149  
lizard\_habitat, 151  
lizard\_run, 152  
loans\_full\_schema, 155  
london\_boroughs, 157  
london\_murders, 158  
mail\_me, 162  
major\_survey, 163  
malaria, 165  
male\_heights, 166  
male\_heights\_fcid, 167  
mammals, 167  
mammogram, 169  
manhattan, 169  
marathon, 170  
mariokart, 171  
mcu\_films, 173  
midterms\_house, 174  
migraine, 175  
military, 175  
mlb, 177  
mlb\_players\_18, 179  
mlb\_teams, 181  
mlbbat10, 178  
mn\_police\_use\_of\_force, 183  
movies, 185  
mtl, 186  
murders, 187  
nba\_finals, 189  
nba\_finals\_teams, 191  
nba\_heights, 192  
nba\_players\_19, 193  
ncbirths, 193  
nuclear\_survey, 197  
nyc, 197  
nyc\_marathon, 199  
nycflights, 198  
offshore\_drilling, 200  
openintro\_colors, 201  
openintro\_palettes, 202  
opportunity\_cost, 203  
orings, 204  
oscars, 205  
outliers, 206  
paralympic\_1500, 207  
penelope, 208  
penetrating\_oil, 209  
penny\_ages, 210  
pew\_energy\_2018, 211  
photo\_classify, 212  
piracy, 212  
playing\_cards, 214  
pm25\_2011\_durham, 216  
pm25\_2022\_durham, 217  
poker, 219  
possum, 219  
ppp\_201503, 220  
present, 221  
president, 222  
prison, 222  
prius\_mpg, 223  
race\_justice, 224  
reddit\_finance, 226

- res\_demo\_1, 232
- res\_demo\_2, 233
- resume, 229
- rosling\_responses, 233
- russian\_influence\_on\_us\_election\_2016, 234
- sa\_gdp\_elec, 238
- salinity, 235
- sat\_improve, 237
- satgpa, 236
- scotus\_healthcare, 241
- seattlepets, 242
- sex\_discrimination, 242
- simpsons\_paradox\_covid, 243
- simulated\_dist, 244
- simulated\_normal, 245
- simulated\_scatter, 245
- sinusitis, 246
- sleep\_deprivation, 246
- smallpox, 247
- smoking, 248
- snowfall, 249
- socialexp, 250
- soda, 251
- solar, 251
- sowc\_child\_mortality, 252
- sowc\_demographics, 253
- sowc\_maternal\_newborn, 255
- sp500, 256
- sp500\_1950\_2018, 258
- sp500\_seq, 259
- speed\_gender\_height, 259
- ssd\_speed, 260
- starbucks, 261
- stats\_scores, 261
- stem\_cell, 262
- stent30, 263
- stocks\_18, 263
- student\_housing, 264
- student\_sleep, 265
- sulphinpyrazone, 265
- supreme\_court, 266
- teacher, 267
- textbooks, 268
- thanksgiving\_spend, 269
- tips, 269
- toohey, 270
- tourism, 271
- toy\_anova, 272
- transplant, 272
- twins, 274
- ucla\_f18, 275
- ucla\_textbooks\_f18, 276
- ukdemo, 278
- unempl, 279
- unemploy\_pres, 280
- us\_temperature, 282
- usb\_admit, 281
- winery\_cars, 283
- world\_pop, 284
- xom, 288
- yawn, 289
- yrbss, 289
- yrbss\_samp, 290
- \* data**
  - heart\_transplant, 118
- \* degree**
  - teacher, 267
- \* demographics**
  - military, 175
- \* diagram**
  - treeDiag, 273
- \* distribution**
  - infmortrate, 132
  - thanksgiving\_spend, 269
- \* dot**
  - dotPlotStack, 74
- \* education**
  - teacher, 267
- \* efficiency**
  - esi, 89
- \* energy**
  - esi, 89
- \* environment**
  - esi, 89
- \* financial**
  - sp500, 256
- \* flight**
  - birds, 30
- \* for**
  - loop, 160
- \* heart**
  - heart\_transplant, 118
- \* histogram**
  - infmortrate, 132
  - thanksgiving\_spend, 269

- \* **index**
    - loop, 160
  - \* **infringement**
    - piracy, 212
  - \* **ipo**
    - ipo, 134
  - \* **legislation**
    - piracy, 212
  - \* **linear**
    - lmPlot, 153
  - \* **looping**
    - loop, 160
  - \* **loop**
    - loop, 160
  - \* **map**
    - london\_boroughs, 157
    - london\_murders, 158
  - \* **message**
    - loop, 160
  - \* **military**
    - military, 175
  - \* **mining**
    - ipo, 134
  - \* **model**
    - lmPlot, 153
  - \* **money**
    - sp500, 256
  - \* **murder**
    - london\_murders, 158
  - \* **music**
    - ipod, 135
  - \* **myPDF**
    - myPDF, 188
  - \* **plot**
    - dotPlotStack, 74
  - \* **probability**
    - treeDiag, 273
  - \* **randomization**
    - heart\_transplant, 118
  - \* **regression**
    - gifted, 109
    - gradestv, 114
  - \* **residuals**
    - lmPlot, 153
  - \* **salary**
    - teacher, 267
  - \* **smoking**
    - smoking, 248
  - \* **smoothing**
    - makeTube, 163
  - \* **squares**
    - makeTube, 163
  - \* **stacked**
    - dotPlotStack, 74
  - \* **stocks**
    - sp500, 256
  - \* **sustainability**
    - esi, 89
  - \* **tables**
    - heart\_transplant, 118
  - \* **table**
    - contTable, 59
  - \* **teacher**
    - teacher, 267
  - \* **tests**
    - heart\_transplant, 118
  - \* **text**
    - ipo, 134
  - \* **transplant**
    - heart\_transplant, 118
  - \* **tube**
    - makeTube, 163
  - \* **wildlife**
    - birds, 30
- 
- absenteeism, 8
  - acs12, 9
  - age.at.mar (age\_at\_mar), 10
  - age\_at\_mar, 10
  - ageAtMar (age\_at\_mar), 10
  - ames, 11
  - ami\_occurrences, 13
  - antibiotics, 14
  - antibiotics\_in\_children (antibiotics), 14
  - arbuthnot, 15
  - ArrowLines, 16, 47, 71, 161
  - arrows, 16, 46
  - ask, 17
  - association, 18
  - association\_1\_3, (association), 18
  - association\_4\_6, (association), 18
  - association\_7\_12 (association), 18
  - assortative\_mating (assortive\_mating), 19
  - assortive.mating (assortive\_mating), 19
  - assortive\_mating, 19



- avandia, 20
- axis, 21, 22
- AxisInDollars, 21, 21, 22
- AxisInPercent, 21, 22
- babies, 23
- babies\_crawl, 23
- bac, 24
- ball.bearing (ball\_bearing), 25
- ball\_bearing, 25
- ballBearing (ball\_bearing), 25
- bdims, 26
- BG, 28
- biontech\_adolescents, 29
- birds, 30
- births, 31
- births14, 23, 32, 32, 194
- blizzard\_salary, 33
- books, 34
- boxPlot, 35, 39, 67, 68, 73, 80, 126
- Braces, 37
- buildAxis, 21, 22, 38, 196
- burger, 41
- calc\_streak, 41
- cancer\_in\_dogs, 42
- cards, 42
- cars04, 43
- cars93, 44, 60
- cat, 59
- cchousing, 45
- CCP, 17, 46, 71, 161
- census, 47
- cherry, 48
- children\_gender\_stereo, 49
- china, 50
- ChiSquareTail, 51
- cia.factbook (cia\_factbook), 52
- cia\_factbook, 52
- classdata, 53
- cle\_sac, 54
- climate70, 54
- climber\_drugs, 55
- coast.starlight (coast\_starlight), 56
- coast\_starlight, 56
- COL, 28, 57
- comics, 58
- contTable, 59
- corr.match (corr\_match), 60
- corr\_match, 60
- country\_iso, 61
- cpr, 62
- cpu, 62
- createEdaOptions (edaPlot), 79
- credits, 64
- CT2DF, 64
- daycare\_fines, 65
- densityPlot, 36, 39, 67, 73, 80, 126
- diabetes2, 69
- dlsegments, 17, 38, 47, 70, 161
- dotPlot, 36, 39, 68, 72, 74, 80, 126
- dotPlotStack, 74
- dream, 75
- drone\_blades, 75
- drug\_use, 76
- duke\_forest, 77
- earthquakes, 78
- ebola\_survey, 79
- edaPlot, 79, 189
- elmhurst, 80
- email, 60, 81, 83, 84
- email50, 82, 83
- email\_test (email), 81
- env\_regulation, 85
- epa2012, 86
- epa2021, 87
- esi, 89
- ethanol, 91
- evals, 92
- exam\_grades, 93
- exams, 93
- exclusive.relationship  
(exclusive\_relationship), 94
- exclusive\_relationship, 94
- fact\_opinion, 95
- fadeColor, 96
- family\_college, 98
- fastfood, 99
- fcid, 100
- fheights, 100
- fish\_age, 101
- fish\_oil\_18, 102
- fitNormal (edaPlot), 79
- flow\_rates, 103
- friday, 104

- full.body.scan(full\_body\_scan), 105
- full\_body\_scan, 105
- gdp\_countries, 106
- gear\_company, 107
- gender\_discrimination, 107
- get\_it\_dunn\_run, 108
- ggplot2::discrete\_scale(), 239, 240
- ggplot2::scale\_color\_gradientn(), 239
- ggplot2::scale\_fill\_gradientn(), 240
- gifted, 109
- global.warming.pew
  - (global\_warming\_pew), 110
- global\_warming\_pew, 110
- goog, 111
- gov\_poll, 111
- gpa, 112
- gpa.iq(gpa\_iq), 113
- gpa\_iq, 113
- gpa\_study\_hours, 113
- gradestv, 114
- grDevices::colorRampPalette(), 202
- gsearch, 115
- gss2010, 115
- gss\_wordsum\_class, 116
- guessMethod(edaPlot), 79
- health.coverage(health\_coverage), 117
- health\_coverage, 117
- healthcare\_law\_survey, 117
- heart\_transplant, 118
- heartTr(heart\_transplant), 118
- helium, 119
- helmet, 120
- hfi, 121
- histPlot, 36, 39, 68, 73, 74, 80, 125
- house, 127
- housing, 129
- hsb2, 129
- husbands.wives(husbands\_wives), 130
- husbands\_wives, 130
- immigration, 131
- IMSCOL, 132, 201
- infmortrate, 132
- iowa, 133
- ipo, 134
- ipod, 135
- iran, 136
- jury, 137
- kobe\_basket, 138
- lab\_report, 141
- labor\_market\_discrimination
  - (labor\_market\_discrimination), 138
- labor\_market\_discrimination, 138
- LAhomes, 142
- law\_resume, 143
- lecture\_learning, 144
- leg\_mari, 148
- lego\_population, 145
- lego\_sample, 147
- life\_exp, 149
- lines, 16, 38
- linResPlot, 149
- lizard\_habitat, 151
- lizard\_run, 152
- lmPlot, 153, 164
- loan50(loans\_full\_schema), 155
- loans\_full\_schema, 155
- london\_boroughs, 157, 159
- london\_murders, 158
- loop, 160
- lsegments, 17, 47, 71, 160
- mail\_me, 162
- major.survey(major\_survey), 163
- major\_survey, 163
- makePlotIcon(edaPlot), 79
- makeTube, 151, 155, 163, 216
- malaria, 165
- male\_heights, 166
- male\_heights\_fcid, 167
- mammals, 167
- mammogram, 169
- manhattan, 169
- marathon, 170
- mariokart, 60, 171
- mcu\_films, 173
- midterms\_house, 174
- migraine, 175
- military, 175
- mlb, 177, 180
- mlb\_players\_18, 179
- mlb\_teams, 181
- mlbbat10, 178, 180

- mn\_police\_use\_of\_force, 183
- MosaicPlot, 65, 184
- movies, 185
- mtl, 186
- murders, 187
- myPDF, 160, 188
- myPNG (myPDF), 188
  
- nba\_finals, 189
- nba\_finals\_teams, 191
- nba\_heights, 192
- nba\_players\_19, 193
- ncbirths, 32, 193
- normTail, 51, 195
- nuclear\_survey, 197
- nyc, 197
- nyc\_marathon, 170, 199
- nycflights, 198
  
- offshore.drilling (offshore\_drilling), 200
- offshore\_drilling, 200
- openintro\_colors, 201, 201
- openintro\_cols, 201
- openintro\_pal, 202
- openintro\_palettes, 202, 202, 239, 240
- opportunity\_cost, 203
- orings, 204
- oscars, 205
- outliers, 206
  
- paralympic\_1500, 207
- penelope, 208
- penetrating\_oil, 209
- penny.ages (penny\_ages), 210
- penny\_ages, 210
- pew\_energy\_2018, 211
- photo\_classify, 212
- piracy, 212
- playing\_cards, 214
- plotNothing (edaPlot), 79
- PlotWLine, 215
- pm25.2011.durham (pm25\_2011\_durham), 216
- pm25\_2011\_durham, 216
- pm25\_2022\_durham, 217
- points, 74
- poker, 219
- possum, 60, 219
- ppp.201503 (ppp\_201503), 220
- ppp\_201503, 220
- present, 221
- president, 222
- prison, 222
- prius\_mpg, 223
  
- qqnormsim, 224
  
- race\_justice, 224
- reddit\_finance, 226
- res\_demo\_1, 232
- res\_demo\_2, 233
- resume, 229, 230
- rnorm, 245
- rosling\_responses, 233
- russian\_influence\_on\_us\_election\_2016, 234
  
- sa\_gdp\_elec, 238
- salinity, 235
- sat\_improve, 237
- satgpa, 236
- scale\_color\_openintro, 239
- scale\_fill\_openintro, 240
- scotus\_healthcare, 241
- seattlepets, 242
- sex\_discrimination, 242
- simpsons\_paradox\_covid, 243
- simulated\_dist, 244
- simulated\_normal, 245
- simulated\_scatter, 245
- sinusitis, 246
- sleep\_deprivation, 246
- smallpox, 247
- smoking, 248
- snowfall, 249
- socialexp, 250
- soda, 251
- solar, 251
- sowc\_child\_mortality, 252
- sowc\_demographics, 253
- sowc\_maternal\_newborn, 255
- sp500, 256
- sp500\_1950\_2018, 258
- sp500\_seq, 259
- speed\_gender\_height, 259
- ssd\_speed, 260
- starbucks, 261
- stats\_scores, 261

stem\_cell, 262  
stent30, 263  
stent365 (stent30), 263  
stocks\_18, 263  
student\_housing, 264  
student\_sleep, 265  
sulphinpyrazone, 265  
supreme\_court, 266  
  
teacher, 267  
text, 46  
textbooks, 268, 278  
tgSpending (thanksgiving\_spend), 269  
thanksgiving.spend  
    (thanksgiving\_spend), 269  
thanksgiving\_spend, 269  
tips, 269  
toohey, 270  
tourism, 271  
toy\_anova, 272  
transplant, 272  
treeDiag, 273  
twins, 274  
  
ucb\_admit (usb\_admit), 281  
ucla\_f18, 275, 278  
ucla\_textbooks\_f18, 276  
ukdemo, 278  
unempl, 279  
unemploy\_pres, 174, 280  
us\_temperature, 282  
usb\_admit, 281  
  
winery\_cars, 283  
world\_pop, 284  
write\_pkg\_data, 287  
  
xom, 288  
  
yawn, 289  
yrbss, 289, 290  
yrbss\_samp, 290