Package ‘qacBase’

Title Functions to Facilitate Exploratory Data Analysis
Version 1.0.3
Description Functions for descriptive statistics, data management, and data visualization.
Depends R (>= 3.5.0)
Encoding UTF-8
LazyData true
RoxygenNote 7.1.2
License MIT + file LICENSE
VignetteBuilder knitr
BugReports https://github.com/rkabacoff/qacBase/issues
URL https://github.com/rkabacoff/qacBase
Suggests rmarkdown, knitr, kableExtra
Imports ggplot2, dplyr, tidyr, ggcormplot, multcompView, PMCMRplus, crayon, purrr, haven, rlang, ggExtra, patchwork
NeedsCompilation no
Author Kabacoff Robert [aut, cre],
Barich Griffen [ctb],
Jamrog Kelly [ctb],
Kravchenko Elizaveta [ctb],
Kuruviла Jacob [ctb],
Liu Lex [ctb],
Nakamura Shota [ctb],
Pham Kim [ctb],
Rodriguez Belen [ctb],
Ross Shane [ctb],
Russo Chris [ctb],
Corpuz Frederick [ctb],
Juradat Nurah [ctb],
Karp Harrison [ctb],
Koech Kevin [ctb],
Peters Anna [ctb],
Shah Dhhyey [ctb],
Stevenson Kenneth [ctb],
Thomas-Franz Kaitlyn [ctb],
Zheng Jiner [ctb],
Aldarmaki Ahmed [ctb],
Alneyadi Mohammed [ctb],
Altai Chossis [ctb],
Colorado Sofia [ctb],
Northrop Blake [ctb],
Peretz Shea [ctb],
Qin Cher [ctb],
Tuhabonye Emma [ctb],
Wong Phillip [ctb]

Maintainer Kabacoff Robert <rkabacoff@wesleyan.edu>
Repository CRAN
Date/Publication 2022-02-09 22:20:02 UTC

R topics documented:

barcharts .................................................. 3
cardata ...................................................... 4
cars74 ....................................................... 5
contents .................................................... 6
cor_plot ..................................................... 7
crosstab ..................................................... 8
densities ................................................... 9
df_plot ...................................................... 10
groupdiff .................................................. 11
histograms ................................................. 12
iso .......................................................... 13
normalize .................................................. 14
phelp ......................................................... 15
plot.crosstab .............................................. 16
plot.tab .................................................... 16
print.contents ............................................ 17
print.crosstab ........................................... 18
print.tab .................................................. 19
qstats ....................................................... 19
rcolors ..................................................... 20
recodes .................................................... 21
scatter ..................................................... 22
skewness ................................................... 24
standardize ............................................... 25
tab .......................................................... 25
tv ............................................................ 26
univariate_plot ......................................... 28
Create barcharts for all categorical variables in a data frame.

Usage

```r
barcharts(
  data,
  fill = "deepskyblue2",
  color = "grey30",
  labels = TRUE,
  sort = TRUE,
  maxcat = 20,
  abbrev = 20
)
```

Arguments

data: data frame
fill: fill color for bars
color: color for bar labels
labels: if TRUE, bars are labeled with percents
sort: if TRUE, bars are sorted by frequency
maxcat: numeric. barcharts with more than this number of bars will not be plotted.
abbrev: numeric. abbreviate bar labels to at most, this character length.

Value

a ggplot graph

Examples

```r
barcharts(cars74)
```
Description

Cars dataset with features including make, model, year, engine, and other properties of the car used to predict its price.

Usage

cardata

Format

A data frame with 11914 rows and 16 variables. The variables are as follows:

- **make**  car brand
- **model**  model given by its brand
- **year**  year of manufacture
- **engine_fuel_type**  type of fuel required by its manufacturer
- **engine_hp**  engine horse power
- **engine_cylinders**  number of cylinders
- **transmission_type**  automatic vs. manual
- **driven_wheels**  AWD, FWD, AWD
- **number_of_doors**  Number of Doors
- **market_category**  Luxury, Performance, Hatchback, etc.
- **vehicle_size**  Compact, Midsize, Large
- **vehicle_style**  Type of Vehicle: Sedan, SUV, Coupe, etc.
- **highway_mpg**  highway miles per gallon
- **city_mpg**  city miles per gallon
- **popularity**  Popularity index
- **msrp**  manufacturer’s suggested retail price

Details

This package contains a detailed car dataset.

Source

Taken from Kaggle [https://www.kaggle.com/CooperUnion/cardataset](https://www.kaggle.com/CooperUnion/cardataset).

Examples

summary(cardata)
Description

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models).

Usage
cars74

Format

A data frame with 32 rows and 11 variables. The variables are as follows:

- **auto**: highway miles per gallon
- **mpg**: Miles/(US) gallon
- **cyl**: Number of cylinders
- **disp**: Displacement (cu.in.)
- **hp**: Gross horsepower
- **drat**: Rear axle ratio
- **wt**: Weight (1000 lbs)
- **qsec**: 1/4 mile time
- **vs**: Engine cylinder configuration
- **am**: Transmission type
- **gear**: Number of forward gears
- **carb**: Number of carburetors

Details

This dataset is the mtcars dataset that comes with base R. However, cyl, vs, am, gear and carb have been converted to factors and rownames have been converted to the variable auto. A description of the variables by Soren Heitmann can be found here.

Source


Examples

summary(cars74)
**contents**

*Detailed description of a data frame*

---

**Description**

`contents` provides a comprehensive description of a data frame, including summary statistics for both quantitative and categorical variables.

**Usage**

```r
contents(data, digits = 2, maxcat = 10, label_length = 20)
```

**Arguments**

- `data` a data frame
- `digits` number of decimal digits for statistics.
- `maxcat` maximum number of levels of a character/factor variable to print.
- `label_length` maximum length of factor level label to print. Longer labels will be truncated.

**Details**

Prints a comprehensive description of a data frame via several tables, a general summary table and tables that provide a breakdown of quantitative and categorical variables.

**Value**

a list with 6 components:

- `dfname` name of data frame
- `nrow` number of rows
- `ncol` number of columns
- `overall` data frame of overall dataset characteristics
- `qvars` data frame with summary statistics for quantitative variables
- `cvars` data frame with summary statistics for categorical variables

**Examples**

```r
contents(cars74)
```
Description

Create a correlation matrix for all quantitative variables in a data frame.

Usage

```r
cor_plot(
data, 
method = c("pearson", "kendall", "spearman"), 
sort = FALSE, 
axis_text_size = 12, 
number_text_size = 3, 
legend = FALSE
)
```

Arguments

- `data` : data frame
- `method` : a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman".
- `sort` : logical. If TRUE, reorder variables to place variables with similar correlation patterns together.
- `axis_text_size` : size for axis labels (default=12).
- `number_text_size` : size for correlation coefficient labels (default=3).
- `legend` : logical, if TRUE the legend is displayed. (default=FALSE)

Details

The `cor_plot` function will only select quantitative variables from a data frame. Categorical variables are ignored. The correlation matrix is presented as a lower triangle matrix. Missing values are deleted in listwise fashion.

Value

a ggplot graph

Note

This function is a wrapper for the `ggcorrplot` function.

Examples

```r
cor_plot(cars74)
cor_plot(cars74, sort=TRUE)
```
Description

This function creates a two way frequency table.

Usage

crosstab(
  data,  
  rowvar,  
  colvar,  
  type = c("freq", "percent", "rowpercent", "colpercent"),
  total = TRUE,
  na.rm = TRUE,
  digits = 2,
  chisquare = FALSE,
  plot = FALSE
)

Arguments

data          data frame
rowvar        row factor (unquoted)
colvar        column factor (unquoted)
type          statistics to print. Options are "freq", "percent", "rowpercent", or "colpercent" for frequencies, cell percents, row percents, or column percents).
total         logical. if TRUE, includes total percents.
na.rm         logical. if TRUE, deletes cases with missing values.
digits        number of decimal digits to report for percents.
chisquare     logical. If TRUE perform a chi-square test of independence
plot          logical. If TRUE generate stacked bar chart.

Details

Given a data frame, a row factor, a column factor, and a type (frequencies, cell percents, row percents, or column percents) the function provides the requested cross-tabulation.

If na.rm = FALSE, a level labeled <NA> added. If total = TRUE, a level labeled Total is added. If chisquare = TRUE, a chi-square test of independence is performed.
densities

Value
If plot=TRUE, return a ggplot2 graph. Otherwise the function return a list with 6 components:

• table (table). Table of frequencies or percents
• type (character). Type of table to print
• total (logical). If TRUE, print row and or column totals
• digits (numeric). number of digits to print
• rowname (character). Row variable name
• colname (character). Column variable name
• chisquare (character). If chisquare=TRUE, contains the results of the Chi-square test. NULL otherwise.

See Also
print.crosstab, plot.crosstab

Examples

# print frequencies
crosstab(mtcars, cyl, gear)
# print cell percents
crosstab(cardata, vehicle_size, driven_wheels)
crosstab(cardata, vehicle_size, driven_wheels, plot=TRUE)
crosstab(cardata, driven_wheels, vehicle_size, type="colpercent", plot=TRUE, chisquare=TRUE)

densities (Density plots)

Description
Create density plots for all quantitative variables in a data frame.

Usage
densities(data, fill = "deepskyblue2", adjust = 1)

Arguments
data data frame
fill fill color for density plots
adjust a factor multiplied by the smoothing bandwidth. See details.
Details

The densities function will only plot quantitative variables from a data frame. Categorical variables are ignored.

The adjust parameter multiplies the smoothing parameter. For example adjust = 2 will make the density plots twice as smooth. The adjust = 1/2 will make the density plots half as smooth (i.e., twice as spiky).

Value

a ggplot graph

Examples

densities(cars74)
densities(cars74, adjust=2)
densities(cars74, adjust=1/2)

df_plot

Visualize a data frame

Description

df_plot visualizes the variables in a data frame.

Usage

df_plot(data)

Arguments

data a data frame.

Details

For each variable, the plot displays

• type (numeric, integer, factor, ordered factor, logical, or date)
• percent of available (and missing) cases

Variables are sorted by type and the total number of variables and cases are printed in the caption.

Value

a ggplot2 graph
**groupdiff**

**See Also**

For more descriptive statistics on a data frame see [contents](#).

**Examples**

```r
df_plot(cars74)
```

---

**groupdiff**  
*Test of group differences*

**Description**

One-way analysis (ANOVA or Kruskal-Wallis Test) with post-hoc comparisons and plots

**Usage**

```r
library(groupdiff)

# example usage

groupdiff(  
data,  
y,  
x,  
method = c("anova", "kw"),  
digits = 2,  
horizontal = FALSE,  
posthoc = FALSE  
)
```

**Arguments**

- `data` a data frame.
- `y` a numeric response variable
- `x` a categorical explanatory variable. It will coerced to be a factor.
- `method` character. Either "anova", or "kw" (see details).
- `digits` Number of significant digits to print.
- `horizontal` logical. If TRUE, boxplots are plotted horizontally.
- `posthoc` logical. If TRUE, the default, perform pairwise post-hoc comparisons (TukeyHSD for ANOVA and Conover Test for Kruskal Wallis). This test will only be performed if there are 3 or more levels for X.

**Details**

The `groupdiff` function performs one of two analyses:

- **anova** A one-way analysis of variance, with TukeyHSD post-hoc comparisons.
- **kw** A Kruskal Wallis Rank Sum Test, with Conover Test post-hoc comparisons.
In each case, summary statistics and a grouped boxplots are provided. In the parametric case, the statistics are n, mean, and standard deviation. In the nonparametric case the statistics are n, median, and median absolute deviation. If posthoc = TRUE, pairwise comparisons of superimposed on the boxplots. Groups that share a letter are not significantly different (p < .05), controlling for multiple comparisons.

Value

a list with 3 components:

result omnibus test
summarystats summary statistics
plot ggplot2 graph

See Also

kwAllPairsConoverTest, multcompLetters.

Examples

# parametric analysis
groupdiff(cars74, hp, gear)

# nonparametric analysis
groupdiff(cardata, popularity, vehicle_style, posthoc=TRUE,
  method="kw", horizontal=TRUE)

histograms

Description

Create histograms for all quantitative variables in a data frame.

Usage

histograms(data, fill = "deepskyblue2", color = "white", bins = 30)

Arguments

data data frame
fill fill color for histogram bars
color border color for histogram bars
bins number of bins (bars) for the histograms
Details

The histograms function will only plot quantitative variables from a data frame. Categorical variables are ignored.

Value

a ggplot graph

Examples

histograms(cars74)
histograms(cars74, bins=15, fill="darkred")

lso

List object sizes and types

Description

lso lists object sizes and types.

Usage

lso(
    pos = 1,
    pattern,
    order.by = "Size",
    decreasing = TRUE,
    head = TRUE,
    n = 10
)

Arguments

pos       a number specifying the environment as a position in the search list.
pattern   an optional regular expression. Only names matching pattern are returned. glob2rx can be used to convert wildcard patterns to regular expressions.
order.by  column to sort the list by. Values are "Type", "Size", "Rows", and "Columns".
decreasing logical. If FALSE, the list is sorted in ascending order.
head      logical. Should output be limited to n lines?
n         if head=TRUE, number of rows should be displayed?

Details

This function list the sizes and types of all objects in an environment. By default, the list describes the objects in the current environment, presented in descending order by object size and reported in megabytes (Mb).
**normalize**

**Value**

a data.frame with four columns (Type, Size, Rows, Columns) and object names as row names.

**Author(s)**

Based on postings by Petr Pikal and David Hinds to the r-help list in 2004 and modified
Dirk Eddelbuettel, Patrick McCann, and Rob Kabacoff.

**References**


**Examples**

data(cardata)
data(cars74)
ls()

---

normalize Normalize numeric variables

**Description**

Normalize the numeric variables in a data frame

**Usage**

normalize(data, new_min = 0, new_max = 1)

**Arguments**

data a data frame.
new_min minimum for the transformed variables.
new_max maximum for the transformed variables.

**Details**

normalize transforms all the numeric variables in a data frame to have the same minimum and maximum values. By default, this will be a minimum of 0 and maximum of 1. Character variables and factors are left unchanged.

**Value**

a data frame
**Note**

Use this function to transform variables into a given range. The default is [0, 1], but [-1, 1], [0, 100], or any other range is permissible.

**Examples**

```r
head(cars74)

cars74_st <- normalize(cars74)
head(cars74_st)
```

---

**Description**

`phelp` provides help on an installed package.

**Usage**

```r
phelp(pckg)
```

**Arguments**

- `pckg` The name of a package

**Details**

This function provides help on an installed package. The package does not have to be loaded. The package name does not need to be entered with quotes.

**Value**

No return value, called for side effects.

**Examples**

```r
phelp(stats)
```
plot.crosstab  

Plot a crosstab object

Description

This function plots the results of a calculated two-way frequency table.

Usage

```r
## S3 method for class 'crosstab'
plot(x, size = 3.5, ...)
```

Arguments

- `x`: An object of class `crosstab`
- `size`: numeric. Size of bar text labels.
- `...`: no currently used.

Value

A ggplot2 graph

Examples

```r
tbl <- crosstab(cars74, cyl, gear, type = "freq")
plot(tbl)

tbl <- crosstab(cars74, cyl, gear, type = "colpercent")
plot(tbl)
```

plot.tab  

Plot a tab object

Description

Plot a frequency or cumulative frequency table

Usage

```r
## S3 method for class 'tab'
plot(x, fill = "deepskyblue2", size = 3.5, ...)
```
print.contents

Arguments

x  An object of class `tab`
fill  Fill color for bars
size  numeric. Size of bar text labels.
...  Parameters passed to a function

Value

a ggplot2 graph

Examples

tbl1 <- tab(cars74, carb)
plot(tbl1)
tbl2 <- tab(cars74, carb, sort = TRUE)
plot(tbl2)
tbl3 <- tab(cars74, carb, cum=TRUE)
plot(tbl3)

print.contents  

Description

print.contents prints the results of the content function.

Usage

## S3 method for class 'contents'
print(x, ...)

Arguments

x  a object of class contents
...  not used.

Value

No return value, called for side effects.
### print.crosstab

**Print a crosstab object**

This function prints the results of a calculated two-way frequency table.

### Description

This function prints the results of a calculated two-way frequency table.

### Usage

```r
## S3 method for class 'crosstab'
print(x, ...)
```

### Arguments

- **x**
  - An object of class crosstab
- **...**
  - not currently used.

### Value

No return value, called for side effects

### Examples

```r
mycrosstab <- crosstab(mtcars, cyl, gear, type = "freq", digits = 2)
print(mycrosstab)

mycrosstab <- crosstab(mtcars, cyl, gear, type = "rowpercent", digits = 3)
print(mycrosstab)
```
### print.tab

**Print a tab object**

#### Description

Print the results of calculating a frequency table

#### Usage

```r
## S3 method for class 'tab'
print(x, ...)
```

#### Arguments

- `x`: An object of class `tab`
- `...`: Parameters passed to the print function

#### Value

No return value, called for side effects

#### Examples

```r
frequency <- tab(cardata, make, sort = TRUE, na.rm = FALSE)
print(frequency)
```

### qstats

**Summary statistics for a quantitative variable**

#### Description

This function provides descriptive statistics for a quantitative variable alone or separately by groups. Any function that returns a single numeric value can be used.

#### Usage

```r
qstats(data, x, ..., stats = c("n", "mean", "sd"), na.rm = TRUE, digits = 2)
```
Arguments

data data frame
x numeric variable in data (unquoted)
... list of grouping variables
stats statistics to calculate (any function that produces a numeric value), Default: c("n", "mean", "sd")
na.rm if TRUE, delete cases with missing values on x and or grouping variables, Default: TRUE
digits number of decimal digits to print, Default: 2

Value

a data frame, where columns are grouping variables (optional) and statistics

Examples

# If no keyword arguments are provided, default values are used
qstats(mtcars, mpg, am, gear)

# You can supply as many (or no) grouping variables as needed
qstats(mtcars, mpg)
qstats(mtcars, mpg, am, cyl)

# You can specify your own functions (e.g., median, median absolute deviation, minimum, maximum)
qstats(mtcars, mpg, am, gear,
  stats = c("median", "mad", "min", "max"))

rcolors

R Colors

Description

Plot a grid of R colors and their associated names

Usage

rcolors(color = NULL, cex = 0.6)

Arguments

color character. A text string used to search for specific color variations (see examples.)
cex numeric. text size for color labels.
**Details**

By default `rcolors` plots the basic 502 distinct colors provided by the `colors` function. If a color name or part of a name is provided, only colors with matching names are plotted.

**Value**

No return value, called for side effects

**References**

This function is adapted from code published by Karl W. Broman.

**See Also**

`colors`

**Examples**

```r
rcolors()
rcolors("blue")
rcolors("red")
rcolors("dark")
```

---

**recode**

Recode one or more variables

**Description**

`recode` recodes the values of one or more variables in a data frame

**Usage**

`recode(data, vars, from, to)`

**Arguments**

- `data`: a data frame.
- `vars`: character vector of variable names.
- `from`: a vector of values or conditions (see Details).
- `to`: a vector of replacement values.
Details

- For each variable in the `vars` parameter, values are checked against the list of values in the `from` vector. If a value matches, it is replaced with the corresponding entry in the `to` vector.
- Once a given observation's value matches a `from` value, it is recoded. That particular observation will not be recoded again by that `recodes()` statement (i.e., no chaining).
- One or more values in the `from` vector can be an expression, using the dollar sign ($) to represent the variable being recoded. If the expression evaluates to `TRUE`, the corresponding `to` value is returned.
- If the number of values in the `to` vector is less than the `from` vector, the values are recycled. This lets you convert several values to a single outcome value (e.g., NA).
- If the `to` values are numeric, the resulting recoded variable will be numeric. If the variable being recoded is a factor and the `to` values are character values, the resulting variable will remain a factor. If the variable being recoded is a character variable and the `to` values are character values, the resulting variable will remain a character variable.

Value

- a data frame

Note

See the vignette for detailed examples.

Examples

```r
df <- data.frame(x = c(1, 5, 7, 3, 0),
                 y = c(9, 0, 5, 9, 2),
                 z = c(1, 1, 2, 2, 1))
df <- recodes(df,
              vars = c("x", "y"),
              from = 0, to = NA)
df <- recodes(df,
              vars = "z",
              from = c(1, 2), to = c("pass", "fail"))
```

Description

Create a scatter plot between two quantitative variables.
Usage

scatter(
    data,
    x,
    y,
    outlier = 3,
    alpha = 1,
    digits = 3,
    title,
    margin = "none",
    stats = TRUE,
    point_color = "deepskyblue2",
    outlier_color = "violetred1",
    line_color = "grey30",
    margin_color = "deepskyblue2"
)

Arguments

data [data frame] quantitative predictor variable
x quantitative response variable
y outlier [number. Observations with studentized residuals larger than this value are flagged. If set to 0, observations are not flagged.]
alpha Transparency of data points. A numeric value between 0 (completely transparent) and 1 (completely opaque).
digits Number of significant digits in displayed statistics.
title Optional title.
margin Marginal plots. If specified, parameter can be histogram, boxplot, violin, or density. Will add these features to the top and right margin of the graph.
stats logical. If TRUE, the slope, correlation, and correlation squared (expressed as a percentage) for the regression line are printed on the subtitle line.
point_color Color used for points.
outlier_color Color used to identify outliers (see the outlier parameter).
line_color Color for regression line.
margin_color Fill color for margin boxplots, density plots, or histograms.

Details

The scatter function generates a scatterplot between two quantitative variables, along with a line of best fit and a 95% confidence interval. By default, regression statistics (b, r, r2, p) are printed and outliers (observations with studentized residuals > 3) are flagged. Optionally, variable distributions (histograms, boxplots, violin plots, density plots) can be added to the plot margins.
Value

a ggplot2 graph

Note

Variable names do not have to be quoted.

Examples

scatter(cars74, hp, mpg)
scatter(cars74, wt, hp)
p <- scatter(ggplot2::mpg, displ, hwy,
  margin="histogram",
  title="Engine Displacement vs. Highway Mileage")
plot(p)

skewness

<table>
<thead>
<tr>
<th>skewness</th>
<th>Skewness</th>
</tr>
</thead>
</table>

Description

Calculate the skewness of a numeric variable

Usage

skewness(x, na.rm = TRUE)

Arguments

x
  numeric vector.

na.rm
  if TRUE, delete missing values.

Value

a number

Examples

skewness(mtcars$mpg)
standardize

Standardize numeric variables

Description

Standardize the numeric variables in a data frame

Usage

standardize(data, mean = 0, sd = 1, include_dummy = FALSE)

Arguments

data: a data frame.
mean: mean of the transformed variables.
sd: standard deviation of the transformed variables.
include_dummy: logical. If TRUE, transform dummy coded (0,1) variables.

Details

standardize transforms all the numeric variables in a data frame to have the same mean and standard deviation. By default, this will be a mean of 0 and standard deviation of 1. Character variables and factors are left unchanged. By default, dummy coded variables are also left unchanged. Use include_dummy=TRUE to transform these variables as well.

Value

a data frame

Examples

head(cars74)
cars74_st <- standardize(cars74)
head(cars74_st)
Description

Function to calculate frequency distributions for categorical variables

Usage

```r
tab(
data,  
x,  
sort = FALSE,  
maxcat = NULL,  
minp = NULL,  
na.rm = FALSE,  
total = FALSE,  
digits = 2,  
cum = FALSE,  
plot = FALSE
)
```

Arguments

data A dataframe
x A factor variable in the data frame.
sort logical. Sort levels from high to low.
maxcat Maximum number of categories to be included. Smaller categories will be combined into an "Other" category.
minp Minimum proportion for a category to be included. Categories representing smaller proportions will be combined into an "Other" category. maxcat and minp cannot both be specified.
na.rm logical. Removes missing values when TRUE.
total logical. Include a total category when TRUE.
digits Number of digits the percents should be rounded to.
cum logical. If TRUE, include cumulative counts and percents. In this case total will be set to FALSE.
plot logical. If TRUE, generate bar chart rather than a frequency table.

Details

The function `tab` will calculate the frequency distribution for a categorical variable and output a data frame with three columns: level, n, percent.
Value

If `plot` = TRUE return a ggplot2 bar chart. Otherwise return a data frame.

Examples

```r
tab(cars74, carb)
tab(cars74, carb, plot=TRUE)
tab(cars74, carb, sort=TRUE)
tab(cars74, carb, sort=TRUE, plot=TRUE)
tab(cars74, carb, cum=TRUE)
tab(cars74, carb, cum=TRUE, plot=TRUE)
```

Description

This is a data set detailing TV usage on days surveyed as determined by the 2017 American Time
Use Survey. The data set includes demographic information, as well as details regarding employment
and family makeup, where applicable. Information on days surveyed, as well as whether the
day is a holiday, is also included.

Usage

tv

Format

A data frame with 10,223 rows and 21 variables. The variables are as follows:

- **id**: ID of respondent
- **weight**: ATUS final weight
- **youngest_child**: Age of the youngest child in the household that is less than 18 years old (if applicable). Range: 1-17; if no child in household: NA
- **age**: Age of respondent
- **sex**: Sex of respondent
- **job**: Status of employment of the respondent. Direct transcription from original codebook: 1 = Employed, at work, 2 = Employed, absent, 3 = Unemployed, on layoff, 4 = Unemployed, looking, 5 = Not in the labor force.
- **m_job**: The response to question, “in the last seven days did you have more than one job?” Returns NA if no job.
- **f_job**: Does the respondent have a full time job or a part time job? (NA if no job)
- **educ**: Are you enrolled in high school, college, or university? (NA if not currently enrolled)
- **educ2**: If yes to educ, are you enrolled in high school or upper schooling? (NA if not currently enrolled)
**partner**  Presence of the respondent’s spouse or unmarried partner in the household with 1 = Spouse present 2 = Unmarried partner present 3 = No spouse/unmarried partner present

**pr_job**  Answer to the question, “does your partner have a job?” (NA if not applicable)

**salary**  Weekly earnings at the respondent’s main job, two decimals implied

**children**  Number of children under 18 in the household

**pr_job_f**  Part time/full time job status of partner, if applicable (NA if partner unemployed or no partner)

**job_hours**  Total hours usually worked per week (-4: Hours vary)

**day**  Day of the week about which the respondent was interviewed (Monday through Friday)

**holiday**  Notes if the respondent was interviewed on a holiday

**elder_care**  Total time spent providing elder care that day by the respondent, in minutes

**child_time**  Total time spent during diary day providing secondary childcare for household children younger than 13, in minutes

**tv**  Minutes spent watching TV

### Details


### Examples

```r
summary(tv)

hist(tv$tv, col="skyblue")
```

### Description

Generates a descriptive graph for a quantitative variable.

### Usage

```r
univariate_plot(
  data,
  x,
  bins = 30,
  fill = "deepskyblue",
  pointcolor = "black",
  density = TRUE,
  densitycolor = "grey",
  alpha = 0.2,
  seed = 1234
)
```
univariate_plot

Arguments

- **data**: a data frame.
- **x**: a variable name (without quotes).
- **bins**: number of histogram bins.
- **fill**: fill color for the histogram and boxplot.
- **pointcolor**: point color for the jitter plot.
- **density**: logical. Plot a filled density curve over the histogram. (default=TRUE)
- **densitycolor**: fill color for density curve.
- **alpha**: Alpha transparency (0-1) for the density curve and jittered points.
- **seed**: pseudorandom number seed for jittered plot.

Details

univariate_plot generates a plot containing three graphs: a histogram (with an optional density curve), a horizontal jittered point plot, and a horizontal box plot. The subtitle contains descriptive statistics, including the mean, standard deviation, median, minimum, maximum, and skew.

Value

- a ggplot2 graph

Note

The graphs are created with ggplot2 and then assembled into a single plot through the patchwork package. Missing values are deleted.

Examples

univariate_plot(mtcars, mpg)
univariate_plot(cardata, city_mpg, fill="lightsteelblue",
    pointcolor="lightsteelblue", densitycolor="lightpink",
    alpha=.6)
Index

* datasets
  - cardata, 4
  - cars74, 5
  - tv, 27

- barcharts, 3
- cardata, 4
- cars74, 5
- colors, 21
- contents, 6, 11
- cor_plot, 7
- crosstab, 8

- densities, 9
- df_plot, 10
- ggcorrplot, 7
- ggplot2, 29
- glob2rx, 13
- groupdiff, 11

- histograms, 12
- kwAllPairsConoverTest, 12

- lso, 13
- mtcars, 5
- multcompLetters, 12

- normalize, 14
- patchwork, 29
- phelp, 15
- plot.crosstab, 9, 16
- plot.tab, 16
- print.contents, 17
- print.crosstab, 9, 18
- print.tab, 19

- qstats, 19
- rcolors, 20
- recodes, 21
- regular expression, 13

- scatter, 22
- skewness, 24
- standardize, 25

- tab, 26
- tv, 27
- univariate_plot, 28