Package 'anscombiser'

October 12, 2020

Title Create Datasets with Identical Summary Statistics

Version 1.0.0

Date 2020-10-10

Description Anscombe's quartet are a set of four two-variable datasets that have several common summary statistics but which have very different joint distributions. This becomes apparent when the data are plotted, which illustrates the importance of using graphical displays in Statistics. This package enables the creation of datasets that have identical marginal sample means and sample variances, sample correlation, least squares regression coefficients and coefficient of determination. The user supplies an initial dataset, which is shifted, scaled and rotated in order to achieve target summary statistics. The general shape of the initial dataset is retained. The target statistics can be supplied directly or calculated based on a user-supplied dataset. The 'datasauRus' package <https://cran.r-project.org/package=datasauRus> provides further examples of datasets that have markedly different scatter plots but share many sample summary statistics.

Imports datasets, graphics, stats

License GPL (>= 2)

LazyData TRUE

Encoding UTF-8

Depends R (>= 3.3.0)

RoxygenNote 7.1.0

Suggests datasauRus, maps, testthat, knitr, rmarkdown

VignetteBuilder knitr

URL https://paulnorthrop.github.io/anscombiser/, https://github.com/paulnorthrop/anscombiser

BugReports https://github.com/paulnorthrop/anscombiser/issues

NeedsCompilation no

Author Paul J. Northrop [aut, cre, cph]

Maintainer Paul J. Northrop <p.northrop@ucl.ac.uk>

Repository CRAN

Date/Publication 2020-10-11 23:40:05 UTC

R topics documented:

`anscombise` *Create new versions of Anscombe's quartet*

Description

Modifies a dataset `x` so that it shares sample summary statistics with Anscombe's quartet.

Usage

```
anscombise(x, which = 1)
```

Arguments

`x`	A numeric matrix or data frame. Each column contains observations on a different variable. Missing observations are not allowed.
`which`	An integer in {1, 2, 3, 4}. Which of Anscombe's dataset to use. Obviously, this makes very little difference.

Details

The input dataset `x` is modified by shifting, scaling and rotating it so that its sample mean and covariance matrix match those of the Anscombe quartet.

Value

An object of class `c("anscombe",class(x))`. A dataset with the same format as `x`. The returned dataset has the following summary statistics in common with Anscombe's quartet.

- The sample means of each variable.
- The sample variances of each variable.
- The sample correlation matrix.
- The estimated regression coefficients from least squares linear regressions of each variable on each other variable. The target and new summary statistics are returned as attributes `old_stats` and `new_stats` and the chosen Anscombe's quartet dataset as an attribute `old_data`.

See Also

`mimic` to modify a dataset to share sample summary statistics with another dataset.

Examples

```
# Old faithful to new faithful
new_faithful <- anscombise(datasets::faithful, which = 4)
plot(new_faithful)
# Then check that the sample summary statistics are the same
plot(new_faithful, input = TRUE)

# Map of Italy
got_maps <- requireNamespace("maps", quietly = TRUE)
if (got_maps) {
  italy <- mapdata("Italy")
  new_italy <- anscombise(italy, which = 4)
  plot(new_italy)
}
```

`anscombiser` *anscombiser: Create Datasets with Identical Summary Statistics*

Description

Anscombe's quartet (Anscombe, 1973) are a set of four two-variable datasets that have several common summary statistics but which have very different joint distributions. This becomes apparent when the data are plotted, which illustrates the importance of using graphical displays in Statistics. This package enables the creation of datasets that have identical marginal sample means and sample variances, sample correlation, least squares regression coefficients and coefficient of determination. The user supplies an initial dataset, which is shifted, scaled and rotated in order to achieve target summary statistics. The general shape of the initial dataset is retained. The target statistics can be supplied directly or calculated based on a user-supplied dataset.

Details

The main functions in anscombiser are

- anscombise, which modifies a user-supplied dataset so that it shares sample summary statistics with Anscombe's quartet.
- mimic, which modified a user-supplied dataset so that is shares sample summary statistics with another user-supplied dataset.

See vignette("intro-to-anscombiser",package = "anscombiser") for an overview of the package.

References

Anscombe, F. J. (1973). Graphs in Statistical Analysis. The American Statistician 27 (1): 17–21. https://doi.org/10.1080/00031305.1973.10478966.

See Also

anscombise and mimic

---

get_stats *Calculate Anscombe's summary statistics*

Description

Calculates a particular set of summary statistics for a dataset.

Usage

```
get_stats(x)
```

Arguments

x	a numeric matrix or data frame with at least 2 columns/variables. Each column contains observations on a different variable. Missing observations are not allowed.

Value

A named list of summary statistics containing

- n The sample size.
- means The sample means of each variable.
- variances The sample means of each variable.
- correlation The sample correlation matrix.
- intercepts,slopes,rsquared Matrices whose (i,j)th entries are the estimated regression coefficients in a regression of x[,i] on x[,j] and the resulting coefficient of determination R^2.

Examples

```
get_stats(anscombe[, c(1, 5)])
```

`mapdata` *Extract longitude and latitude values*

Description

Extracts longitude and latitude values for a particular region from the world map supplied by the maps package.

Usage

```
mapdata(region = ".", map = "world", exact = FALSE, ...)
```

Arguments

`region`	Passed to `map` as the argument `regions`.
`map`	Passed to `map` as the argument `database`
`exact`	The argument `exact` passed to the `map` function.
`...`	Additional arguments to be passed to `map`.

Value

A dataframe with two columns: `long` and `lat` for longitude and latitude.

Examples

See the examples in `mimic`.

`mimic` *Modify a dataset to mimic another dataset*

Description

Modifies a dataset `x` so that it shares sample summary statistics with another dataset `x2`. ‘

Usage

```
mimic(x, x2, ...)
```

Arguments

x, x2	Numeric matrices or data frames. Each column contains observations on a different variable. Missing observations are not allowed. get_stats(x2) sets the target summary statistics. If x2 is missing then set_stats is called with d = ncol(x) and any additional arguments supplied via ....
...	Additional arguments to be passed to set_stats.

Details

The input dataset x is modified by shifting, scaling and rotating it so that its sample mean and covariance matrix match those of x2.

Value

A dataset with the same format as x. The returned dataset has the following summary statistics in common with x2.

- The sample means of each variable.
- The sample variances of each variable.
- The sample correlation matrix.
- The estimated regression coefficients from least squares linear regressions of each variable on each other variable. The target and new summary statistics are returned as attributes old_stats and new_stats. If x2 is supplied then it is returned as a attribute old_data.

See Also

anscombise modifies a dataset so that it shares sample summary statistics with Anscombe's quartet.

Examples

```
### 2D examples

# The UK and a dinosaur
got_maps <- requireNamespace("maps", quietly = TRUE)
got_datasauRus <- requireNamespace("datasauRus", quietly = TRUE)
if (got_maps && got_datasauRus) {
  library(maps)
  library(datasauRus)
  dino <- datasaurus_dozen_wide[, c("dino_x", "dino_y")]
  UK <- mapdata("UK")
  new_UK <- mimic(UK, dino)
  plot(new_UK)
}

# Trump and a dinosaur
if (got_datasauRus) {
  library(datasauRus)
  dino <- datasaurus_dozen_wide[, c("dino_x", "dino_y")]
  new_dino <- mimic(dino, trump)
  plot(new_dino)
```

```
}

## Examples of passing summary statistics

# The default is zero mean, unit variance and no correlation
new_faithful <- mimic(faithful)
plot(new_faithful)

# Change the correlation
mat <- matrix(c(1, -0.9, -0.9, 1), 2, 2)
new_faithful <- mimic(faithful, correlation = mat)
plot(new_faithful)

### A 3D example

new_randu <- mimic(datasets::randu, datasets::trees)
# The samples summary statistics are equal
get_stats(new_randu)
get_stats(datasets::trees)
```

`plot.anscombe` *Plot method for objects of class "anscombe"*

Description

`plot` method for objects inheriting from class `"anscombe"`.

Usage

```
## S3 method for class 'anscombe'
plot(x, input = FALSE, stats = TRUE, digits = 3, legend_args = list(), ...)
```

Arguments

`x`	an object of class `'anscombe'`, a result of a call to `anscombise` or `mimic`.
`input`	A logical scalar. Should the old, input data, that is, the Anscombe's dataset chosen for `anscombise` or the argument `x2` to `mimic`, be plotted? If `old = FALSE` then the new, output data are plotted. If `old = TRUE` then the old data are plotted.
`stats`	A logical scalar. Should the sample summary statistics `n`, `means`, `variances` and `correlation` be added to the plot?
`digits`	An integer. The argument `digits` passed to `signif` to round the values of the statistics before adding them to the plot.
`legend_args`	A list of arguments to be passed to `legend` when `stats = TRUE`, especially `legend_args$x` to control the position of the legend.
`...`	Further arguments to be passed to `plot`

Details

This function is only applicable in 2 dimensions, that is, when `length(attr(x,"new_stats")$means)` = 2.

Value

Nothing is returned.

Examples

See the examples in `anscombise` and `mimic`.

See Also

`anscombise` and `mimic`.

`print.anscombe` *Print method for objects of class "anscombe"*

Description

`print` method for class "anscombe".

Usage

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

Arguments

`x`	an object of class "anscombe", a result of a call to `anscombise` or `mimic`.
`...`	Additional optional arguments to be passed to `print`.

Details

Just extracts the new dataset from `x` and prints it using `print`.

Value

The argument `x`, invisibly.

See Also

`anscombise` and `mimic`

set_stats *Create a list of summary statistics*

Description

Creates a list of summary statistics to pass to mimic.

Usage

```
set_stats(d = 2, means = 0, variances = 1, correlation = diag(2))
```

Arguments

d	An integer that is no smaller than 2.
means	A numeric vector of sample means.
variances	A numeric vector of positive sample variances.
correlation	A numeric correlation matrix. None of the off-diagonal entries in correlation are allowed to be equal to 1 in absolute value.

Details

The vectors means and variances are recycled using rep_len to have length d.

Value

A list containing the following components.

- means a d-vector of sample means.
- variances a d-vector sample variances.
- correlation a d by d correlation matrix.

Examples

```
# Uncorrelated with zero means and unit variances
set_stats()
# Sample correlation = 0.9
set_stats(correlation = matrix(c(1, 0.9, 0.9, 1), 2, 2))
```

trump *Donald Trump*

Description

A dataset that provides an image of Donald Trump's face.

Usage

```
trump
```

Format

A matrix with 4885 rows and 2 columns: `x` and `y`.

Source

This image was created by Accentaur from the Noun Project. https://thenounproject.com/term/donald-trump/727774/

Index