Package 'gsse401'

December 9, 2001

Version 0.0-8

Date 2001/12/08

Title GSSE-401 teaching supporting functions

Author Paulo J. Ribeiro Jr <p.ribeiro@lancaster.ac.uk> Peter J. Diggle <p.diggle@lancaster.ac.uk>

Maintainer Paulo J. Ribeiro Jr <p.ribeiro@lancaster.ac.uk>

Depends R (>= 1.0)

Description Teaching support functions for the postgraduate course GSSE401 tought at Lancaster University

License GPL Version 2 or later

URL http://www.maths.lancs.ac.uk/~diggle/gsse401

R topics documented:

`ansc` *GSSE 401 - Anscombe quartet data*

Description

The `ansc` data frame has 11 rows and 6 columns.

This synthetic data-set was constructed by the statistican Frank Anscombe to illustrate how residuals can be used to discriminate between data-sets which give identical fitted regression models.

Usage

```
data(ansc)
```

Format

This data frame contains the following columns:

x1 a numeric vector with explanatory variable for each of responses 1, 2 or 3.

y1 a numeric vector with response 1.

y2 a numeric vector with response 2.

y3 a numeric vector with response 3.

x4 a numeric vector with explanatory variable for response 4

y4 a numeric vector with response 4.

Source

Anscombe, Francis J. (1973) Graphs in statistical analysis. American Statistician, 27, 17-21.

Examples

```
data(ansc)
summary(ansc)
attach(ansc)
parmf <- par()$mfrow
par(mfrow=c(2,2))
plot(x1, y1)
plot(x1, y2)
plot(x1, y3)
plot(x4, y4)
par(parmf)
detach()
```

`campy` *GSSE 401 - Campylobacter data*

Description

The `campy` data frame has 60 rows and 4 columns.

This data-set records monthly numbers of incident campylobacter cases in the Morecambe Bay area, from January 1990 to December 1994 inclusive.

Usage

```
data(campy)
```

Format

This data frame contains the following columns:

month a numeric vector with the number of month.

cases a numeric vector with the number of incident cases.

nstrend a numeric vector with non-seasonal trend estimated by moving average smoothing as explained in lecture notes.

strend a numeric vector with seasonal trend estimated by moving average smoothing as explained in lecture notes.

Source

Supplied by Dr K Jones (Lancaster University, UK).

Examples

```
data(campy)
attach(campy)
plot(month, cases)
lines(month, nstrend, lty=2)
lines(month, strend, lty=3)
```

`class96` *GSSE 401 - Students weight and height data*

Description

The `class96` data frame has 19 rows and 3 columns. The `class97` data frame has 44 rows and 3 columns. The `class98` data frame has 52 rows and 3 columns.

These data-sets records self-reported heights and weights of students who attended a GSSE401 class in the academic years 1996–97, 1997–98 and 1998–99. The symbol `NA` denotes missing values.

Usage

```
data(class96)

data(class97)

data(class98)
```

Format

This data frame contains the following columns:

height a numeric vector with height of the students (m).

weight a numeric vector with weight of the students (kg).

sex a factor with two levels:

`F` female student

`M` male student

Examples

```
data(class98)
by(class98$weight, class98$sex, summary)
##
data(class97)
coplot(weight ~ height | sex, data=class97)
```

---

clt *GSSE 401 - Illustrates the Central Limit Theorem*

Description

Take samples of size n from a vector of data of size N and computes the empirical distribution of the sample mean, illustrating the central limit theorem.

Usage

```
clt(x, n, nsim, plot = TRUE, ncols = 2)

plot(x)
```

Arguments

`x`	a numeric vector with the data
`n`	an integer defining the sample size
`nsim`	an integer defining number of samples to be taken
`plot`	logical. If `TRUE` histograms are produced in the graphical device.
`ncols`	numerical. The number of columns in the graphical device. Only valid if `plot = TRUE`.

Value

Returns a list which is an object of the class `clt`. The list components are:

`data`	a vector with the data passed to the function
`sizeN`	a list with vectors of averages (`xbar`) and standard deviations (`sd`) of each sample

For each sample size N provided there will be one component as the latter.

The function `hist.clt` plots histograms of the sample means on the current graphics device.

Author(s)

Peter J. Diggle ⟨p.diggle@lancaster.ac.uk⟩
Paulo Justiniano Ribeiro Jr. ⟨p.ribeiro@lancaster.ac.uk⟩.

Examples

```
clt(rexp(1000), c(2,4,8,16,32), 1000)
#
par.now <- par(no.readonly=TRUE)
par(mfrow=c(3,2))
data.clt <- clt(exp(rnorm(2000)), c(2,4,8,16,32), 1000, plot=F)
plot(data.clt)
par(par.now)
#
# For an interactive input type:

clt()
```

`crossover` *GSSE 401 - Asthma data*

Description

The `crossover` data frame has 13 rows and 4 columns.

Usage

```
data(crossover)
```

Format

This data frame contains the following columns:

sequence a factor with levels indicating the sequence of administration of drugs:

- `FS` Formoterol first and Salbutamol second
- `SF` Salbutamol first and Formotemol second

patient a factor with patient identifier

PEF1 a numeric vector with measurements of the lung function in period one

PEF2 a numeric vector with measurements of the lung function in period two

Details

This data-set gives the results of a crossover trial into the effectiveness of two proprietary medicines used to treat chronic asthma. The medicines are Formoterol (F) and Salbutamol (S). Each patient receives both treatments in sequence, with an intervening "washout" period intended to ensure that the drug received in period 1 has no residual effect during period 2. The resonse is a measure of lung function, peak expiratory flow (PEF), 8 hours after treatment.

Source

Supplied by Prof S. Senn (University College, London, UK).

---

glyp *GSSE 401 - Glyphosate data*

Description

The `glyp` data frame has 54 rows and 4 columns.

This data-set records the results of an experiment to investigate the relationship between root length of safflower plants and the concentration of glyphosate (a weed-killer) introduced into the water-supply.

Usage

```
data(glyp)
```

Format

This data frame contains the following columns:

conc normal-bracket11bracket-normala numeric vector with glyphosate concentration in water supply (*ppm*)

water a factor with the type of water to which glyphosate was added

`1` distilled water

`2` tap water

replicate a factor with levels corresponding to replicate experiment started on dates respectively. Note that control runs with zero glyphosate were also duplicated on each of the three dates.

length a numeric vector with total root length of 15 safflower plants

Source

Supplid by Prof P. Diggle (Lancaster University, UK) from an experiment conducted by CSIRO Centre for Irrigation Research, Griffith, New South wales, Australia

`gravity` *GSSE 401 - Gravity data*

Description

The `gravity` data frame has 22 rows and 2 columns.

Usage

```
data(gravity)
```

Format

This data frame contains the following columns:

time a numeric vector with time taken to fall (seconds)

distance a numeric vector with vertical distance fallen (cm)

Details

These data are from an undergraduate experiment to estimate the value of the gravitational constant, g (the acceleration of a free-falling body at the earth's surface). For each replicate of the experiment (row of data) a ball-bearing was released from a pre-determined height above the workbench and the time taken before the ball-bearing hit the workbench was recorded, as described in the lecture notes.

Examples

```
##
## Loading the data
##
data(gravity)
##
## Attaching the data frame
##
attach(gravity)
##
## Ploting original data
##
plot(distance, time)
##
## Now including the intercept
##
plot(distance, time, xlim=c(0,max(distance)), ylim=c(0,max(time)))
##
## transforming the distances vector
##
dt <- sqrt(distance)
plot(dt, time, xlim=c(0,max(dt)), ylim=c(0,max(time)))
##
## plotting fitted models without and with intercept
##
abline(lm(time~dt-1))
abline(lm(time~dt), lty=2)
```

```
##
detach()
```

`gsse401.data` *GSSE 401 - Lists data and functions in the package*

Description

These functions lists the data-sets and functions included in the package `gsse401`.

Usage

```
gsse401.data()

gsse401.functions()
```

`lh` *GSSE 401 - Cow luteinising hormone data*

Description

The `lh` data frame has 100 rows and 8 columns.

This data-set records the concentrations of luteinising hormone in blood samples taken at 15-minute intervals from each of 8 cows.

The eight columns correspond to the 8 cows, the rows to the time-sequence of measurements on each cow.

Usage

```
data(lh)
```

Format

This data frame contains the following columns:

cow1 ... cow8 a numeric vector with hormone measurements in *cow1* to *cow8*

Source

Diggle, P.J. and Zeger, S.L. (1984) A non-Gaussian model for time series with pulses. *JASA* 354-9.

Examples

```
pmf <- par()$mfrow
##
data(lh)
par(mfrow=c(4,2), mar=c(3,3,3,1))
for(i in 1:length(lh))
  plot(lh[,i], type="l", ylim=range(lh), main=paste("cow", i))
##
par(mfrow=pmf)
```

`mandible` *GSSE 401 - Golden Jackals Mandible Data*

Description

The `mandible` data frame has 10 rows and 2 columns.

The data are mandible lengths in (mm) for male and female golden jackals (*Canis aureus*) for 10 of each sex in the collection in the British Museum (Natural History).

Usage

```
data(mandible)
```

Format

This data frame contains the following columns:

females a numeric vector with mandible lengths for females (*mm*).

male a numeric vector with mandible lengths for males (*mm*).

Source

Higham, C.F, Kijngam, A. and Manly, B.F.J. (1980) An analysis of prehistoric canid remains from Thailand. Journal of Archeologycal Science, 7, 149-165.

References

Manly, B.F.J. (1991) Randomization, Bootstrap and Monte Carlo methods in Biology. Second Edition. Chapman and Hall.

Examples

```
data(mandible)
summary(mandible)
boxplot(mandible)
```

`maxtemp` *GSSE 401 - Daily maximum temperatures in Lancaster*

Description

The `maxtemp` data frame has 366 rows and 4 columns.

This data-set records daily maximum temperatures at the Hazelrigg field station, near the Lancaster University campus, from 1 September 1995 to 31 August 1996.

Usage

```
data(maxtemp)
```

Format

This data frame contains the following columns:

year a numeric vector corresponding to the year

month a numeric vector corresponding to the mounth

day a numeric vector corresponding to the day

maxtemp a numeric vector with the maximum temperature

Source

Hazelrigg field station, Lancaster, UK.

Examples

```
data(maxtemp)
summary(maxtemp$maxtemp)
plot(maxtemp$maxtemp)
```

---

mctest *GSS 401 - Paired and two-sample Monte Carlo tests*

Description

Performs Monte Carlo tests for paired or two-sample analysis.
A graphical display of the results is produced by `plot.mctest`.

Usage

```
mctest(x, y, paired = TRUE, nsim, plot = TRUE)

plot(x)

print(x)
```

Arguments

`x`	a numeric vector with data on first variable
`y`	a numeric vector with data on the second variable
`paired`	logical indicating whether a paired test should be performed. Defualts to `FALSE` which implies the two-sample analysis.
`nsim`	an integer with the number of Monte Carlo samples. Defaults to 1000.
`plot`	logical. If `TRUE` plot is automaticaly produced.

Details

Paired data

1 For the n pairs of data (x_i, y_i) compute the differences $d_i = x_i - y_i$ and then the test statistic:

$$\frac{\overline{d}}{\sqrt{\frac{Var(d)}{n}}}$$

2 For each pair re-allocate the two data to the two groups randomly. Then re-compute the test statistics above.

3 repeat the previous step *nsim* times. This generates an empirical distribution for the test statistics.

4 compare the statistics computed for the data with the empirical distribution.

Two-sample

For two-sample test a randomization test is performed as follows:

1 For two independent samples x and y of sizes n_x and n_y define

$$S^2 = \frac{(n_x - 1)Var(x) + (n_y - 1)Var(y)}{n_x + n_y - 2}$$

and compute the following statistics:

$$\frac{\overline{x} - \overline{y}}{\sqrt{S^2(\frac{1}{n_x} + \frac{1}{n_y})}}$$

2 Pool $n_x + n_y$ data together and re-allocate to the two groups of sizes n_x and n_y randomly.

3 repeat the previous step *nsim* times. This generates an empirical distribution for the test statistics.

4 compare the statistics computed for the data with the empirical distribution.

For both cases the test statistic for the original data is compared against the empirical distribution in order to produce *P-values*. Upper and lower tail probabilities are computed by counting how many values of the statistic computed for the simulated data are above and below the value obtained for the original data.

Additionally, *P-values* based on the `t` distribution are also reported.

Value

The function `mctest` returns an object of the class `mctest` which is a list with components

`p`	a numerical vector with upper and lower tail probabilities based on the empirical distribution
`pt`	a numerical vector with upper and lower tail probabilities based on the `t` distribution
`data.statistic`	the statistics above computed for the original data
`sim.statistic`	the statistics above computed for each simulations

The function `plot.mctest` produces a histogram of the empirical distribution with an indication to the value of the data statistics. A `t` distribution can be added to the plot.

Author(s)

Peter J. Diggle ⟨p.diggle@lancaster.ac.uk⟩
Paulo Justiniano Ribeiro Jr. ⟨p.ribeiro@lancaster.ac.uk⟩.

Examples

```
##
## A two-sample test
##
data(mandible)
mctest(mandible$female, mandible$male, paired = FALSE)
#
# For an interactive input type:

mctest()
```

`queue` *GSSE 401 - Simulation a Queue*

Description

This function simulates a stochastic process describing number of subjects in a queue.

Usage

```
queue(lambda, rho, n, plot = TRUE)

plot(queue)
```

Arguments

`lambda`	a numeric values for the arrival parameter. If not provided, a prompt is issue.
`rho`	a numeric value for the services parameter. If not provided, a prompt is issue.
`n`	an integer value of the number of arrivals. If not provided, a prompt is issue.
`plot`	logical. If `TRUE` plot is automaticaly produced.

Value

The function `queue` returns a $(2n+1)\times 2$ matrix with columns corresponding to cumulative times and number in the queue respectively. The class `queue` is assigned.

The function `plot.queue` takes an object of the class `queue` and produces a plot of numbers *vs times.*

Author(s)

Peter J. Diggle ⟨p.diggle@lancaster.ac.uk⟩
Paulo Justiniano Ribeiro Jr. ⟨p.ribeiro@lancaster.ac.uk⟩.

Examples

```
queue(20, 20, 200)
#
q1 <- queue(.6, .5, 50, plot = FALSE)
q1
plot(q1)
#
# For an interactive input type:

queue()
```

`reg` *GSSE 401 - Illustrates regression models*

Description

These functions takes arguments defining a regression model with one or two explanatory variables and plots the data, regression model and residuals.

Usage

```
reg(n.expl, ...)

reg1(true.model, n.points, range.x, regular, x,  ...)

reg2(true.model, n.points, range.x1, range.x2,
     regular, x1, x2, ...)
```

Arguments

`n.expl`	the number of explanatory variables. The only values allowed are 1 or 2.
`true.model`	a equation defining the true regression model. Examples of specifications are: $2 + 3 * x$ for one explanatory variable and $3 + 2 * x1 - 3 * x2$ for two explanatory variables
`range.x, range.x1, range.x2`	a two elements numerical vector with minimum and maximum values for the explanatory variable
`n.points`	number of data points
`regular`	logical. `TRUE` (the default) indicates that values of the explanatory variable will be regularly spaced between the range of x. `FALSE` means that values of x will be sampled in the range of x
`xvec, xvec1, xvec2`	objects or keybord input with values of the explanatory variables. Only used if `range.x` and `range.y` are not provided
`ask`	logical. If `TRUE` the user is prompted before the next plot is shown
`...`	further arguments to be passed to the plot function

Details

The main function `reg` asks for the number of explanatory variables and then calls `reg1` or `reg2` for the case of one or two respectively.

The function takes a "true" model and values for the explanatory variable(s) and build the response variable from this, adding to the response variable an error with variance equivalent to 20% of its variance.

A linear regression model is fitted and results are displayed in the graphical device.

Value

The function `reg1` produces a 2D plot with data against the explanatory variable and adds a line with the fitted regression model.

If the packages `scatterplot3d` is available, the function `reg2` produces a 3D plot with data against the explanatory variables and adds a plane with the regression model fitted to the data.

In both cases a second plot with residuals against fitted value is produced.

Author(s)

Peter J. Diggle ⟨p.diggle@lancaster.ac.uk⟩
Paulo Justiniano Ribeiro Jr. ⟨p.ribeiro@lancaster.ac.uk⟩.

See Also

`lm`, `scatterplot3d`

Examples

```
reg(1, "1 + 3*x", n.p = 21, range.x = c(0,20), reg = TRUE)
reg(1, "1 + 3*log(x^2)", n.p=25, range.x = c(1, 100), reg=TRUE)
reg(1, "10 - 2*exp(-x/10)", x = 1:30)
##
reg(2, "10 - 5*x1 + 3*x2" , x1 = runif(25), x2 =runif(25))
reg(2, "1+ 3*sqrt(x1) -2*x2^2" , n.p=50, range.x1=c(10, 30),
    range.x2 =c(1, 5))
#
# For an interactive input type:

reg()
```

---

rubber *GSSE 401 - Rubber abrasion experiment data*

Description

The `rubber` data frame has 10 rows and 3 columns.

Usage

```
data(rubber)
```

Format

This data frame contains the following columns:

piece a numeric vector with the number of the test piece

untreated a numeric vector with measured abrasion resistance for untreated half

treated a numeric vector with measured abrasion resistance for treated half

Details

These data are from a paired comparison experiment to assess the effectiveness of a chemical treatment in increasing the abrasion resistance of rubber. A more detailed description of the experiment is given in the lecture notes.

Source

Davies, O.L. (1954). Design and Analysis of Industrial Experiments. Oliver and Boyd.

`screen` *GSSE 401 - Screening storm water data*

Description

The `screen` data frame has 8 rows and 5 columns. Give a concise description here

Usage

```
data(screen)
```

Format

This data frame contains the following columns:

run a numeric vector with the run number

collar a numeric vector with collar screen mesh sizes

horizontal a numeric vector with horizontal screen mesh size

flow a numeric vector with flow rate of water

solids a numeric vector with percent solids removed from water

Details

These data are from a factorial experiment to test the effectiveness of a screening facility for storm water overflow, as described in the lecture notes.

Source

Box, G.E.P, Hunter, W.G, and Hunter, J.S. (1978) Statistics for experimenters. Wiley

---

`ugclass` *GSSE 401 - Undergraduate classroom data*

Description

The `ugclass` data frame has 63 rows and 5 columns.

Usage

```
data(ugclass)
```

Format

This data frame contains the following columns:

sex a factor with levels of sex:
- `F` for female
- `M` for male

hand a factor with levels handed-ness:
- `L` for left-handed
- `R` for right handed

height a numeric vector with height of the students (cm)

weight a numeric vector with weight of the students (kg)

digit a numeric vector with a random digit assign to to each student

Details

This data-set records self-reported sex, handed-ness, height and weight of students who attended an undergraduate class in the academic year 1898/98. The students were also asked to write down a "random digit" (with no detailed explanation of what this means!).

`warping` *GSSE 401 - Warping of Cu material data*

Description

The `warping` data frame has 8 rows and 5 columns.

Usage

```
data(warping)
```

Format

This data frame contains the following columns:

temperature a numeric vector with temperature (degrees centrigrade)

Cu40 a numeric vector with extent of warping with Cu content 40%

Cu60 a numeric vector with extent of warping with Cu content 60%

Cu80 a numeric vector with extent of warping with Cu content 80%

Cu100 a numeric vector with extent of warping with Cu content 100%

Details

These data are from a factorial experiment to investigate how metal plates with different levels of copper content warp when subjected to different temperatures. Warping is undesirable.

Examples

```
data(warping)
matplot(x=warping[,1], y=warping[,-1], ty="l", xlab="Temperature", ylab="warping")
legend(50, 15, c("40%", "60%", "80%", "100%"), lty=1:4, col=1:4)
```

Index