Package ‘MSG’

October 12, 2022

Type Package

Title Data and Functions for the Book Modern Statistical Graphics

Version 0.8

Description A companion to the Chinese book "Modern Statistical Graphics".

License GPL

Depends R (>= 3.5.0)

Imports RColorBrewer (>= 1.1.2)

Suggests animation (>= 2.6), igraph (>= 1.2.4.1), KernSmooth (>= 2.23.15), maps (>= 3.3.0), mvtnorm (>= 1.0.11), rgl (>= 0.100.30), RgoogleMaps (>= 1.4.4), plotrix (>= 3.7.6), ggplot2 (>= 3.2.1), grid (>= 3.6.0), sna (>= 2.4)

URL https://github.com/yihui/MSG

BugReports https://github.com/yihui/MSG/issues

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Yihui Xie [aut, cre] (<https://orcid.org/0000-0003-0645-5666>), Peng Zhao [aut], Lijia Yu [ctb], Xiangyun Huang [ctb]

Maintainer Yihui Xie <xie@yihui.name>

Repository CRAN

Date/Publication 2021-07-21 19:50:05 UTC

R topics documented:

MSG-package

andrews_curve

assists

BinormCircle

andrews_curve

assists

BinormCircle
Index

MSG-package  Modern Statistical Graphics

Description

Datasets and functions for the Chinese book “Modern Statistical Graphics”.

Author(s)

Yihui Xie <https://yihui.org>

andrews_curve  Draw Andrew’s Curve

Description

This function evaluates the transformation of the original data matrix for $t$ from $-\pi$ to $\pi$, and uses `matplot` to draw the curves.
andrews_curve

Usage

```r
andrews_curve(
  x,
  n = 101,
  type = "l",
  lty = 1,
  lwd = 1,
  pch = NA,
  xlab = "t",
  ylab = "f(t)",
  ...
)
```

Arguments

- `x`: a data frame or matrix
- `n`: number of x-axis values at which f(t) is evaluated
- `type, lty, lwd, pch, xlab, ylab, ...`: passed to `matplot`

Value

a matrix of coefficients for each observation at different t values

Author(s)

Yihui Xie <https://yihui.org>

References

https://en.wikipedia.org/wiki/Andrews_plot

See Also

`matplot`

Examples

```r
andrews_curve(iris[, -5], col = as.integer(iris[, 5]))
```
**BinormCircle**

**Assists**

Assists between players in CLE and LAL

**Description**

The players in the rows assisted the ones in the columns.

**References**

http://www.basketballgeek.com/data/

**Examples**

data(assists)

```r
if (require("sna")) {
  set.seed(2011)
  gplot(assists, displaylabels = TRUE, label.cex = 0.7)
}
```

---

**BinormCircle**

Random numbers containing a “circle”

**Description**

The data was generated from two independent random variables (standard Normal distribution) and further points on a circle were added to the data. The order of the data was randomized.

**Format**

A data frame with 20000 observations on the following 2 variables.

- **V1** the first random variable with the x-axis coordinate of the circle
- **V2** the second random variable with the y-axis coordinate of the circle

**Details**

See the example section for the code to generate the data.

**Source**

Examples

data(BinormCircle)

## original plot: cannot see anything
plot(BinormCircle)

## transparent colors (alpha = 0.1)
plot(BinormCircle, col = rgb(0, 0, 0, 0.1))

## set axes limits
plot(BinormCircle, xlim = c(-1, 1), ylim = c(-1, 1))

## small symbols
plot(BinormCircle, pch = ".")

## subset
plot(BinormCircle[sample(nrow(BinormCircle), 1000), ])

## 2D density estimation
library(KernSmooth)
fit = bkde2D(as.matrix(BinormCircle), dpik(as.matrix(BinormCircle)))
# perspective plot by persp()
persp(fit$x1, fit$x2, fit$fhat)

if (interactive() && require("rgl")) {
  # perspective plot by OpenGL
  rgl.surface(fit$x1, fit$x2, fit$fhat)
  # animation
  M = par3d("userMatrix")
  play3d(par3dinterp(userMatrix = list(M, rotate3d(M, pi/2, 1, 0, 0), rotate3d(M, pi/2, 0, 1, 0), rotate3d(M, pi, 0, 0, 1))), duration = 20)
}

## data generation
x1 = rnorm(10000)
y1 = rnorm(10000)
x2 = rep(0.5 * cos(seq(0, 2 * pi, length = 500)), 20)
y2 = rep(0.5 * sin(seq(0, 2 * pi, length = 500)), 20)
x = cbind(c(x1, x2), c(y1, y2))
BinormCircle = as.data.frame(round(x[sample(20000), ], 3))

canabalt

The scores of the game Canabalt from Twitter

description

The scores of the game Canabalt from Twitter
References

(the URL is not longer accessible)

Examples

```r
library(ggplot2)
data(canabalt)
print(qplot(device, score, data = canabalt))
print(qplot(reorder(death, score, median), score, data = canabalt, geom = "boxplot") +
    coord_flip())
```

---

char_gen

*Generate a matrix of similar characters*

Description

This function prints a matrix of characters which are very similar to each other.

Usage

```r
char_gen(x = c("V", "W"), n = 300, nrow = 10)
```

Arguments

- `x` a character vector of length 2 (usually two similar characters)
- `n` the total number of characters in the matrix
- `nrow` the number of rows

Value

a character matrix on the screen

Author(s)

Yihui Xie <https://yihui.org>

Examples

```r
char_gen()
char_gen(c("0", "Q"))
```
ChinaLifeEdu

Life Expectancy and the Number of People with Higher Education in China (2005)

Description

This data contains the life expectancy and number of people with higher education in the 31 provinces and districts in China (2005).

Format

A data frame with 31 observations on the following 2 variables.

- **Life.Expectancy**  Life expectancy
- **High.Edu.NO**  Number of people with higher education

Source


Examples

```r
data(ChinaLifeEdu)
x = ChinaLifeEdu
plot(x, type = "n", xlim = range(x[, 1]), ylim = range(x[, 2]))
u = par("usr")
rect(u[1], u[3], u[2], u[4], col = "antiquewhite", border = "red")
library(KernSmooth)
est = bkde2D(x, apply(x, 2, dpik))
contour(est$x1, est$x2, est$fhat, nlevels = 15, col = "darkgreen", add = TRUE,
vfont = c("sans serif", "plain"))
```

---

cut_plot

Cut the points in a scatter plot into groups according to x-axis

Description

This function can categorize the variable on the x-axis into groups and plot the mean values of y. The purpose is to show the arbitrariness of the discretization of data.

Usage

```r
cut_plot(x, y, breaks, ..., pch.cut = 20)
```
Arguments

- `x` the x variable
- `y` the y variable
- `breaks` the breaks to cut the x variable
- `...` other arguments to be passed to `plot.default`
- `pch.cut` the point symbol to denote the mean values of y

Author(s)

Yihui Xie [https://yihui.org](https://yihui.org)

Examples

```r
x = rnorm(100)
y = rnorm(100)
cut_plot(x, y, seq(min(x), max(x), length = 5))
```

Description

Export of US and China from 1999 to 2004 in US dollars

Format

A data frame with 13 observations on the following 3 variables.

- **Export** amount of export
- **Year** year from 1999 to 2004
- **Country** country: US or China

Source

[https://www.wto.org/english/res_e/statis_e/statis_e.htm](https://www.wto.org/english/res_e/statis_e/statis_e.htm)

Examples

```r
data(Export.USCN)
par(mar = c(4, 4.5, 1, 4.5))
plot(1:13, Export.USCN$Export, xlab = "Year / Country", ylab = "US Dollars ($10^16)", axes = FALSE, type = "h", lwd = 10, col = c(rep(2, 6), NA, rep(4, 6)), lend = 1, panel.first = grid())
xlabel = paste(Export.USCN$Year, "\n", Export.USCN$Country)
xlabel[7] = ""
xlabel
abline(v = 7, lty = 2)
```
gov.cn.pct

Percentage data in some government websites

description

This data was collected from Google by searching for percentages in some government websites.

format

A data frame with 10000 observations on the following 4 variables.

- **percentage**: a numeric vector: the percentages
- **count**: a numeric vector: the number of webpages corresponding to a certain percentage
- **round0**: a logical vector: rounded to integers?
- **round1**: a logical vector: rounded to the 1st decimal place?

details

We can specify the domain when searching in Google. For this data, we used `site:gov.cn`, e.g. to search for `87.53% site:gov.cn`.

source

Google (date: 2009/12/17)

examples

data(gov.cn.pct)
pct.lowess = function(cond) {
  with(gov.cn.pct, {
    plot(count ~ percentage, pch = ifelse(cond, 4, 20), col = rgb(0:1, 
      0, 0, c(0.04, 0.5))[cond + 1], log = "y")
    lines(lowess(gov.cn.pct[cond, 1:2], f = 1/3), col = 2, lwd = 2)
    lines(lowess(gov.cn.pct[!cond, 1:2], f = 1/3), col = 1, lwd = 2)
  })
}
par(mar = c(3.5, 3.5, 1, 0.2), mfrow = c(2, 2))
with(gov.cn.pct, {
  plot(percentage, count, type = "l", panel.first = grid())
  plot(percentage, count, type = "l", xlim = c(10, 11), panel.first = grid())
  pct.lowess(round0)
  pct.lowess(round1)
heart_curve

Draw a heart curve

Description

Calculate the coordinates of a heart shape and draw it with a polygon.

Usage

heart_curve(n = 101, ...)

Arguments

n

the number of points to use when calculating the coordinates of the heart shape

... other arguments to be passed to polygon, e.g. the color of the polygon (usually red)
msg

Author(s)
Yihui Xie <https://yihui.org>

Examples

heart_curve()
heart_curve(col = "red")
heart_curve(col = "pink", border = "red")

Description
Plot a graph with a pre-installed R script

Usage
msg(fig = "3.6", show_code = TRUE, print_plot = TRUE, filter = 0)

Arguments

fig Character. The figure number or the R script name, which is given in the book.
show_code Logical. TRUE means the codes are shown in the console.
print_plot Logical. TRUE means the graph is printed.
filter Integer. The line numbers indicating which lines in the code are displayed (when positive) or hidden (when negative).

Value
A graph and the source code

Examples
# msg('3.6') msg('ChinaPop')
murcia  Composition of Soil from Murcia Province, Spain

Description

The proportions of sand, silt and clay in soil samples are given for 8 contiguous sites. The sites extended over the crest and flank of a low rise in a valley underlain by marl near Albudeite in the province of Murcia, Spain. The sites were small areas of ground surface of uniform shape internally and delimited by relative discontinuities externally. Soil samples were obtained for each site at 11 random points within a 10m by 10m area centred on the mid-point of the site. All samples were taken from the same depth. The data give the sand, silt and clay content of each sample, expressed as a percentage of the total sand, silt and clay content.

References

http://www.statsci.org/data/general/murcia.html

Examples

data(murcia)
boxplot(sand ~ site, data = murcia)

music  Attributes of some music clips

Description

Attributes of some music clips

References


Examples

data(music)
### PlantCounts

**Number of plants corresponding to altitude**

**Description**

For each altitude, the number of plants is recorded.

**Format**

A data frame with 600 observations on the following 2 variables.

- **altitude** altitude of the area
- **counts** number of plants

**Source**


**Examples**

```r
## different span for LOWESS
data(PlantCounts)
par(las = 1, mar = c(4, 4, 0.1, 0.1), mgp = c(2.2, 0.9, 0))
with(PlantCounts, {
  plot(altitude, counts, pch = 20, col = rgb(0, 0, 0, 0.5), panel.first = grid())
  for (i in seq(0.01, 1, length = 70)) {
    lines(lowess(altitude, counts, f = i), col = rgb(i, 0, 0, 0.5), lwd = 1.5)
  }
})
```

---

### quake6

**Earth quakes from 1973 to 2010**

**Description**

The time, location and magnitude of all the earth quakes with magnitude being greater than 6 since 1973.

**References**

https://d.cosx.org/d/101510

**Examples**

```r
data(quake6)
library(ggplot2)
qplot(year, month, data = quake6) + stat_sum(aes(size = ..n..)) + scale_size(range = c(1, 10))
```
The differences of P-values in t test assuming equal or unequal variances

Description

Given that the variances of two groups are unequal, we compute the difference of P-values assuming equal or unequal variances respectively by simulation.

Format

A data frame with 1000 rows and 99 columns.

Details

See the Examples section for the generation of this data.

Source

By simulation.

References


Examples

data(t.diff)
boxplot(t.diff, axes = FALSE, xlab = expression(n[1]))
axis(1)
axis(2)
box()

## reproducing the data
if (interactive()) {
  set.seed(123)
  t.diff = NULL
  for (n1 in 2:100) {
    t.diff = rbind(t.diff, replicate(1000, {
      x1 = rnorm(n1, mean = 0, sd = runif(1, 0.5, 1))
      x2 = rnorm(30, mean = 1, sd = runif(1, 2, 5))
      t.test(x1, x2, var.equal = TRUE)$p.value - t.test(x1, x2, var.equal = FALSE)$p.value
    )))
  }
  t.diff = as.data.frame(t(t.diff))
  colnames(t.diff) = 2:100
}
Results of a Simulation to Tukey’s Fast Test

Description

For the test of means of two samples, we calculated the P-values and recorded the counts of Tukey’s rule of thumb.

Format

A data frame with 10000 observations on the following 3 variables.

- `pvalue.t` P-values of t test
- `pvalue.w` P-values of Wilcoxon test
- `count` Tukey’s counts

Details

See the reference for details.

Source

Simulation; see the Examples section below.

References


Examples

data(tukeyCount)

```r
## does Tukey’s rule of thumb agree with t test and Wilcoxon test?
with(tukeyCount, {
      ucount = unique(count)
      stripchart(pvalue.t ~ count, method = "jitter", jitter = 0.2, pch = 19,
                  cex = 0.7, vertical = TRUE, at = ucount - 0.2, col = rgb(1, 0, 0, 0.2),
                  xlim = c(min(count) - 1, max(count) + 1), xaxt = "n", xlab = "Tukey Count",
                  ylab = "P-values")
      stripchart(pvalue.w ~ count, method = "jitter", jitter = 0.2, pch = 21,
                  cex = 0.7, vertical = TRUE, at = ucount + 0.2, add = TRUE, col = rgb(0, 0, 1, 0.2),
                  xaxt = "n")
      axis(1, unique(count))
      lines(sort(ucount), tapply(pvalue.t, count, median), type = "o", pch = 19,
            cex = 1.3, col = "red")
      lines(sort(ucount), tapply(pvalue.w, count, median), type = "o", pch = 21,
            cex = 1.3, col = "blue", lty = 2)
      legend("topright", c("t test", "Wilcoxon test"), col = c("red", "blue"),
```

pch = c(19, 21), lty = 1:2, bty = "n", cex = 0.8)
})

if (interactive()) {

## this is how the data was generated
set.seed(402)
n = 30
tukeyCount = data.frame(t(replicate(10000, {
x1 = rweibull(n, runif(1, 0.5, 4))
x2 = rweibull(n, runif(1, 1, 5))
c(t.test(x1, x2)$p.value, wilcox.test(x1, x2)$p.value, with(rle(rep(0:1, each = n)[order(c(x1, x2))]), ifelse(head(values, 1) == tail(values, 1), 0, sum(lengths[c(1, length(lengths)]))))
}))
colnames(tukeyCount) = c("pvalue.t", "pvalue.w", "count")
}

tvearn

<table>
<thead>
<tr>
<th>Top TV earners</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>The pay per episode for actors as well as other information.</td>
</tr>
<tr>
<td><strong>References</strong></td>
</tr>
<tr>
<td><a href="https://flowingdata.com/2011/02/15/visualize-this-tvs-top-earners/">https://flowingdata.com/2011/02/15/visualize-this-tvs-top-earners/</a></td>
</tr>
<tr>
<td><strong>Examples</strong></td>
</tr>
</tbody>
</table>
| data(tvearn)
plot(pay ~ rating, data = tvearn)
library(ggplot2)
qplot(pay, data = tvearn, geom = "histogram", facets = gender ~ ., binwidth = 20000)
qplot(rating, pay, data = tvearn, geom = c("jitter", "smooth"), color = type) |

vec2col

| Generate colors from a vector |
| **Description** |
| This function generates a color vector from an input vector, which can be of the class numeric or factor. |
Usage

vec2col(vec, n, name)

## Default S3 method:
vec2col(vec, n, name)

## S3 method for class 'factor'
vec2col(vec, n, name)

Arguments

vec       the numeric or factor vector
n         the number of colors to be generated from the palette
name      the name of the palette

Value

a vector of colors corresponding to the input vector

Author(s)

Yihui Xie <https://yihui.org>

Examples

## convert factor to colors
with(iris, plot(Petal.Length, Petal.Width, col = vec2col(Species), pch = 19))

# another palette
with(iris, plot(Petal.Length, Petal.Width, col = vec2col(Species, name = "Dark2"), pch = 19))

## turn numeric values to colors
with(iris, plot(Petal.Length, Petal.Width, col = vec2col(Petal.Width), pch = 19))
Index

* package
  MSG-package, 2
andrews_curve, 2
assists, 4
BinormCircle, 4
canabalt, 5
char_gen, 6
ChinaLifeEdu, 7
cut_plot, 7
Export.USCN, 8
gov.cn.pct, 9
heart_curve, 10
matplot, 3
MSG (MSG-package), 2
msg, 11
MSG-package, 2
murcia, 12
music, 12
PlantCounts, 13
plot.default, 8
polygon, 10
quake6, 13
t.diff, 14
tukeyCount, 15
tvearn, 16
vec2col, 16