Package 'visualizationTools'

July 21, 2011

Type Package

Title Package contains a few functions to visualize statistical circumstances.

Suggests utils, stats, grDevices, graphics,base64

Depends R (>= 2.12.2),utils, stats, grDevices, graphics, base64

Version 0.2.02

Date 2011-07-21

Author Thomas Roth Etienne Stockhausen

Maintainer T. Roth <thomas.roth@tu-berlin.de>
E. Stockhausen <stocdarf@mailbox.tu-berlin.de>

Description Package contains function to visualize a t-test, the power of a t-test,control charts and the influence of regulating them, Oc-curves, the Law of large Numbers and confidence intervals.

License GPL version 2 or newer

LazyLoad yes

R topics documented:

`visualizationTools-package`
Package contains a few functions to visualize statistical circumstances.

Description

Package contains function to visualize a t-test, the power of a t-test, control charts and the influence of regulating them, Oc-curves, the Law of large Numbers and confidence intervals.

Details

Package:	visualizationTools
Type:	Package
Version:	0.2.02
Date:	2011-07-21
License:	GPL version 2 or newer
LazyLoad:	yes

Author(s)

Thomas Roth
Etienne Stockhausen

Maintainer:
T. Roth <thomas.roth@tu-berlin.de>
E. Stockhausen <stocdarf@mailbox.tu-berlin.de>

References

- Vardeman, Stephen B. & Jobe, J. Marcus (1999) Statistical Quality Assurance Methods for Engineers, 1st ed. New York: John Wiley & Sons.
- Montgomery, D.C. (2005) Introduction to Statistical Quality Control, 5th ed. New York: John Wiley & Sons.
- Blume, J.D.; Royall, R.M. (February 2003) Illustrating the Law of Large Numbers (and Confidence Intervals), The American Statistical Association Vol. 57 No. 1, p. 51-55

Examples

```
par(ask=FALSE)
LLN(n=100,distr="unif",fun=median,param=list(min=2,max=7),col="orange")
par(ask=TRUE)
ocCurve(n=c(2,3,4,5,6,7,8,10,15,25,50,100),
        sig.level=0.05,col=rep(2,12),main="OC-Curves")
var=c(29.60, 28.26,30.50,31.79,29.24,29.75,27.24,30.31,29.27,27.63,31.79,
      27.43,30.18,28.65,28.30,29.28)
to=t.test(var,mu=30,alternative="two.sided",conf.level=0.95)
plot(to)
po=power.t.test(n=30,delta=7,sd=12,sig.level=0.05,type="one.sample",
```

```
                alternative="one.sided")
plot(po,main="one.sample one.sided")
plot(0,0,col="white",axes=FALSE,ylab="",xlab="")
par(ask=FALSE)
qrk(type="o",pch=20,n=5,FUN=mean,N=250,rest=0)
par(ask=TRUE)
plot(0,0,col="white",axes=FALSE,ylab="",xlab="")
par(ask=FALSE)
qrkSD(pch=20,n=1,FUN=mean,N=250,rest=0,numPlot=25,drift=TRUE,
      spread=TRUE,start=0.05,ENDdr=1.5,ENDsp=1.5,speed=2)
par(ask=TRUE)
plot(0,0,col="white",axes=FALSE,ylab="",xlab="")
par(ask=FALSE)
CLT(fun=mean,times=100,distribution=c("normal","weibull",
"gamma","normal","beta"),
param=list(list(mean=0,sd=0.01,n=100),list(shape=1,scale=3,n=100),
list(n=100,shape=0.1),list(mean=2,sd=0.1,n=100),
list(n=100,shape1=1,shape2=2)),
seed=123,col=c(rep("grey",5),"green"))
par(ask=TRUE)
plot(0,0,col="white",axes=FALSE,ylab="",xlab="")
par(ask=FALSE)
confSim()
par(ask=TRUE)
```

`CLT` *Function to visualize the Central Limit Theorem*

Description

Function visualize the Central Limit Theorem by apply a chosen function to a population of random values out of different distributions and plotting the result step by step in a histogram.

Usage

```
CLT(fun, type, times, distribution, param, sleep, col, line.col, seed,
    main)
```

Arguments

`fun`	gives the function that is applied to the population of distributions given by distribution.
`type`	gives the type of the histogram. It can be “density” or “counts”.
`times`	give the number of runs for creating a population and plotting the histograms.
`distribution`	gives a vector of used distributions for the population. For more information see Details!
`param`	a list of list, which contain the paramater for the single distributions given by distribution. Even if distribution has length = 1, param has to be a list of a single list!
`sleep`	a single value giving the time in seconds between the single plots of the histogram. Variable for the command `Sys.sleep`.

col	a vector giving the colors of all plotted histograms.
line.col	color of the plotted density line (only if type="density").
seed	value for the command set.seed.
main	vector for the mains of every single histogram.

Details

CLT will accept the following distributions:

- "beta"
- "cauchy"
- "chi-squared"
- "exponential"
- "f"
- "log-normal"
- "logistic"
- "negative binomial"
- "normal"
- "poisson"
- "t"
- "weibull"
- "gamma"
- "unif"
- "binomial"

Value

CLT returns a invisble list containing a matrix including all randomized values of the different distributions and a vector containing the results of executing fun on every row of that matrix. This vector represents the values plotted in the histogram.

Author(s)

Thomas Roth thomas.roth@tu-berlin.de
Etienne Stockhausen stocdarf@mailbox.tu-berlin.de

Examples

```
CLT(fun=mean,times=100,distribution=c("normal","weibull",
    "gamma","normal","beta"),
    param=list(list(mean=0,sd=0.01,n=100),list(shape=1,scale=3,n=100),
    list(n=100,shape=0.1),list(mean=2,sd=0.1,n=100),
    list(n=100,shape1=1,shape2=2)),
    seed=123,col=c(rep("grey",5),"green"))
```

`confSim` *Visualization/Simulation of the Position of Confidence Intervals*

Description

Function visualize the positions of several confidence intervals of normal distributed values in dependence of a few parameters as sample size, confidence level, etc.

Usage

```
confSim(fun = mean, conf.level = 0.95, mu = 0, stdev = 1, sleep = 0.2, trials =
```

Arguments

`fun`	gives the type of which confidence interval is regarded. Possibilities for fun are in this version `mean` and `sd`. `mean` is the default and will lead to a visualization of confidence intervals for the mean of the values of each trial. `sd` will lead to the visualization of the confidence intervals for the standard deviation.
`conf.level`	gives the confidence level for the confidence intervals. It must be set between 0 and 1. The default value for `conf.level` is 0.95.
`mu`	gives the mean of the random normal distributed values dragged for each trial. By default 0.
`stdev`	gives the standard deviation of the random normal distributed values dragged for each trial. By default 1.
`sleep`	gives the time in seconds between each step of the visualization in which a single line is drawn.
`trials`	gives the number of trials, i.e. the number of samples that are drawn. The default value is 100.
`n`	gives the sample size for each trial. By default there will be drawn 8 values for each trial.
`N`	gives the number of trials that will be displayed at the y axis at the same time. This parameter loses his function if `sim` ist set to FALSE. By default `N` will be 20
`xlim`	gives the range of the x-axis and can be used to compare the visualization of different settings. If this value is missing the function will adjust the range automatically.
`sim`	a logigal value specifying whether the visualization should be animated or not. By default `sim` is set to TRUE.

Details

Value

The function `confSim()` returns an invisible list with 2 entries, which contain the x-coordinates for the end points of every single confidence interval.

Note

This function is not yet finished, but can be used as dexcribed above.
It is planned to integrate further functions.

Author(s)

Thomas Roth thomas.roth@tu-berlin.de
Etienne Stockhausen stocdarf@mailbox.tu-berlin.de

See Also

`Sys.sleep`

Examples

```
confSim()                                                    #function
```

LLN *Function to visualize the "Law of Large Numbers".*

Description

LLN visualize the “Law of Large Nimbers” by apply a chosen function to a population of random values out of a chosen distribution. The results of the function are plotted as a graph.

Usage

```
LLN(n, fun = mean, distr, param, sleep, main, type, xlab, ylab, ...)
```

Arguments

`n`	is a single value that gives the number of random values.
`fun`	gives the function that is applied to the population given by distribution.
`distr`	is a string that gives the type of distribution for the random population. For further information see ‘Details’.
`param`	is a list that gives the parameters needed to specify the distribution. Notice that the number of observations ‘n’ is the same as the above and must not be specified in that list anymore!
`sleep`	is a single value and gives the time in seconds the system will pause between plotting two values. Variable for the internal usage of `Sys.sleep`.
`main`	is an optional string and gives an overall title for the plot: see ‘title’

type	is an optional string and gives the type for the plot: see plot
xlab	is an optional string and gives a title for the x axis: see 'title'
ylab	is an optional string and gives a title for the y axis: see 'title'
...	Arguments to be passed to methods, such as graphical parameters (see 'par').

Details

LLN will accept the following distributions:

- "beta"
- "cauchy"
- "chi-squared"
- "exponential"
- "f"
- "log-normal"
- "logistic"
- "negative binomial"
- "normal"
- "poisson"
- "t"
- "weibull"
- "gamma"
- "unif"
- "binomial"

Value

LLN returns a list. The first element gives the n single random values of the chosen distribution. The second object gives the values on which the function 'fun' is applied and which are plotted.

Note

A line for the expected value is in this version only be drawn for the functions 'mean' and 'sd'. The idea of visualize the Law of Large Numbers is mentioned by Blume and Royall (2003).

Author(s)

Thomas Roth: thomas.roth@tu-berlin.de
Etienne Stockhausen: stocdarf@mailbox.tu-berlin.de

References

- Montgomery, D.C. (2005) Introduction to Statistical Quality Control, 5th ed. New York: John Wiley & Sons.
- Blume, J.D.; Royall, R.M. (February 2003) Illustrating the Law of Large Numbers (and Confidence Intervals), The American Statistical Association Vol. 57 No. 1, p. 51-55

See Also

plot
Sys.sleep

Examples

```
par(ask=FALSE)
 LLN(n=100,distr="normal",fun=mean,param=list(mean=2,sd=0.5),col=2)
par(ask=TRUE)
```

`ocCurve` *Function to plot oc-Curves.*

Description

This function plots oc-Curves by using the parameter out of power.t.test

Usage

```
ocCurve(n, sig.level, alternative, type, distribution, col, xlim,
        ylim, ...)
```

Arguments

n	can be a single value or a vector of vealues giving the number of observations per group: see power.t.test
sig.level	a single value giving the significance level (Type I error probability) for the test: see power.t.test
alternative	is a char value giving the alternative (one- or two sided) for the test: see power.t.test
type	is a char value giving the type (one- or two sample,paired) for the test: see power.t.test
distribution	is a char value indicating the test-procedure used to get the plotting-values. Further information see Note.
col	'col' a vector of colors for the plotted oc-curves.
xlim	giving the x limits (x1, x2) of the plot. Note that x1 > x2 is allowed and leads to a 'reversed axis'.
ylim	giving the y limits of the plot.
...	other graphical parameters (see par).

Details

Value

ocCurves() returns the values for beta listed in a matrix.

Note

This function is not yet fully finished. This Version will only work for t-test. The parameter distribution can therfore only be "t". In further Versions the visualization of oc-curves of other tests, like the chi-squared-test, will be implemented.

Author(s)

Thomas Roth: thomas.roth@tu-berlin.de
Etienne Stockhausen: stocdarf@mailbox.tu-berlin.de

References

Montgomery, D.C. (2005) Introduction to Statistical Quality Control, 5th ed. New York: John Wiley & Sons.

See Also

power.t.test
par

Examples

```
ocCurve(n=c(2,3,4,5,6,7,8,10,15,25,50,100),sig.level=0.05,col=rep(2,12),
        main="OC-Curves")
```

---

plot.htest *Function to visualize results of a t-test.*

Description

This function visualize graphically the results of a t-test by using the values provided by t.test.

Usage

```
## S3 method for class 'htest'
plot(x, col.region, conf.int, col.conf, border, lwd, pch, ...)
```

Arguments

x	is an object of class "htest"
col.region	is a color string, a numeric value or a function giving the color for the visualization of the significance niveau of the test (type I error).Default value is 'lightblue'
conf.int	is a logical value appropriating whether the confidence intervall calculated by t.test is to be drawn. Default value is 'TRUE'
col.conf	is a color string, a numeric value or a function giving the color for the shading lines, which are showing the confidence intervall calculated by t.test.This variable has no effect if 'conf.int' is set as 'FALSE'. Default value is 'gray'
border	is a color string, a numeric value or a function giving the color for the border of the plotted curve. Default value is '1'
lwd	is giving the line width of the plotted curve, a _positive_ number, defaulting to '1'. Default value is '2'
pch	is either an integer specifying a symbol or a single character to be used as marking the fitted t-value given by t.test
...	Other aguments that can be passed to plot

Details

Value

plot.htest does not return any value. Values used are returned by t.test.

Note

The development of this function is fully completed for the usage of t.test.
Enhancements for further test methods are planned to be implemented in future.

Author(s)

Thomas Roth: thomas.roth@tu-berlin.de
Etienne Stockhausen: stocdarf@mailbox.tu-berlin.de

See Also

t.test
plot
par

Examples

```
# to --> TestObject
var=c(29.60, 28.26, 30.50, 31.79, 29.24, 29.75, 27.24, 30.31, 29.27,
      27.63, 31.79, 27.43, 30.18, 28.65, 28.30, 29.28)
to=t.test(var,mu=30,alternative="two.sided",conf.level=0.95)
plot(to)
###
var=c(37.86, 39.52, 40.39, 38.63, 40.39, 39.16, 37.31, 40.38, 40.22,
      39.72, 40.29, 39.27, 38.68, 40.00, 39.66, 40.04, 39.34, 39.94)
```

```
to=t.test(var,mu=40,alternative="less",conf.level=0.95)
plot(to,col.region="yellow")
###
var=c(503.08, 499.48, 505.47, 508.91, 502.10, 503.47, 496.76, 504.95,
      502.18, 497.81, 508.91, 497.29, 504.61, 500.53, 499.60, 502.22,
      508.83, 497.62, 500.84, 510.83, 504.07, 496.38, 510.06, 497.25)
to=t.test(var,mu=500,alternative="greater",conf.level=0.90)
plot(to,col.region="orange",conf.int=FALSE)
###
MachineA=c(20.46, 19.92, 20.82, 21.34, 20.31, 20.52, 19.51, 20.74,
           20.33, 19.67, 21.34, 19.59, 20.69, 20.08, 19.94, 20.33)
MachineB=c(19.33, 19.90, 21.65, 20.46, 19.12, 21.51, 19.27, 20.23,
           19.28, 20.07, 21.75, 21.08, 19.71, 19.63, 20.44, 19.27)
to=t.test(MachineA,MachineB,alternative="two.sided",conf.level=0.95)
plot(to,col.region="purple",col.conf=rgb(0.2,0.8,0,0.6))
```

`plot.power.htest` *Function to visualize the power of a t-test.*

Description

This function visualize graphically the power, the type I error and the type II error of a t-test by using the values provided by `power.t.test`.

Usage

```
## S3 method for class 'power.htest'
plot(x, col, col.line, lwd, main, xlab, ylab, ...)
```

Arguments

`x`	is an object of class "power.htest"
`col`	is a vector giving the colors for visualize the type I error, the type II error and the power
`col.line`	is a vector giving the color of the lines of the distribution curves
`lwd`	is a vector giving the width of the lines of the distribution curves
`main`	giving an overall title for the plot: see `title`
`xlab`	giving a title for the x axis : see `title`
`ylab`	giving a title for the y axis : see `title`
`...`	Other aguments that can be passed to `plot`

Details

Value

plot.power.htest does not return any value.

Note

The development of this function is not yet fully completed.
There are minor deviations between the calculated value and the plotted value of the power which are insignificant.
The function can be used with very small deviations to visualize the power of t-tests with sample numbers n greater 10. For smaller sample numbers this function does not deliver right visualizations, especially for type one.sample! Please consider this, if you want to visualize the power of a t-test.

Author(s)

Thomas Roth: thomas.roth@tu-berlin.de
Etienne Stockhausen: stocdarf@mailbox.tu-berlin.de

See Also

`title`
`power.t.test`

Examples

```
#po --> PowerObject
par(mfrow=c(2,2))
po=power.t.test(n=30,delta=7,sd=12,sig.level=0.05,
                type = "one.sample", alternative="one.sided")
plot(po,main="one.sample one.sided")
po=power.t.test(n=30,delta=7,sd=12,sig.level=0.05,
                type = "one.sample", alternative="two.sided")
plot(po,main="one.sample two.sided")
po=power.t.test(n=30,delta=7,sd=12,sig.level=0.05,
                type = "two.sample", alternative="one.sided")
plot(po,main="two.sample one.sided")
po=power.t.test(n=30,delta=7,sd=12,sig.level=0.05,
                type = "two.sample", alternative="two.sided")
plot(po,main="two.sample two.sided")
```

---

qrk *Simulation of wrong permanent regulation in quality-control-charts.*

Description

Simulate different types of control charts and shows the influence of a wrong permanent regulation of the considered values.

Usage

```
qrk(FUN, mu, stdev, n, N, cl, wl, sl, numPlot, seed, rest, ...)
```

Arguments

FUN	gives the type of the control chart. The implemented possibilities are the functions mean, median and sd. For an individual observation chart mean and median can be used but n must be set to '1'. Default value is 'mean'.
mu	is the expected value of the randomised single values (normal distribution). Default value is '1'.
stdev	is the standard deviation of the randomised single values (normal distribution). Default value is '0.1'.
n	gives the sample size of the regarded groups. Default value is '1'.
N	is the number of points that will be plotted after the limits are calculated and drawn. Default value is '1000'.
cl	A vector giving the quantiles of the control lines. By default 'c(0.00135,0.99865)'.
wl	A vector giving the quantiles of the warning lines. By default 'c(0.0225,0.9775)'.
sl	A vector optional giving the position of the tolerances. By default 'FALSE' (not drawn)
numPlot	gives the number of points visualised on the screen before the points is moving. Default value is '20'.
seed	Variable for set.seed to generate repeatable results. Default value is '125879'.
rest	Value for Sys.sleep i.e. break time in seconds between the plotting of the points. Default value is '0.1'.
...	Further graphical parameters see par

Details

The adjustment of the next value is be shown by the blue arrow on the right side of the plot.

Value

'qrk' returns a data.frame in which plotted regulated and not-regulated values are listed, as well as the single values and for both , which are given in groups to 'FUN' to obtain the plotted values.

Author(s)

Thomas Roth: thomas.roth@tu-berlin.de
Etienne Stockhausen: stocdarf@mailbox.tu-berlin.de

References

- Vardeman, Stephen B. & Jobe, J. Marcus (1999) Statistical Quality Assurance Methods for Engineers, 1st ed. New York: John Wiley & Sons.
- Montgomery, D.C. (2005) Introduction to Statistical Quality Control, 5th ed. New York: John Wiley & Sons.

See Also

par
set.seed
Sys.sleep

Examples

```
par(ask=FALSE)
qrk(type="o",pch=20,n=5,FUN=mean,N=500,rest=0)
par(ask=TRUE)
```

`qrkSD` *Simulation of a spread or a drift in control charts.*

Description

'qrkSD' shows the influence of a spread or a drift in control charts. Therefore the breaks of the 4 main Western Electric alarm rules are be counted and visualized. The spread or/ and the drift of the normal distribution represented in the control chart is visualized too.

Usage

```
qrkSD(FUN, mu, stdev, n, N, cl, wl, sl, numPlot, seed, rest,
      drift, spread, start, ENDdr, ENDsp, speed, ...)
```

Arguments

`FUN`	gives the type of the control chart. The implemented possibilities are the functions mean, median and sd. For an individual observation chart mean and median can be used but n must be set to '1'. Default value is 'mean'.
`mu`	is the expected value of the randomised single values (normal distribution). Default value is '1'.
`stdev`	is the standard deviation of the randomised single values (normal distribution). Default value is '0.1'.
`n`	gives the sample size of the regarded groups. Default value is '1'.
`N`	is the number of points that will be plotted after the limits are calculated and drawn. Default value is '1000'.
`cl`	A vector giving the quantiles of the control lines. By default 'c(0.00135,0.99865)'.
`wl`	A vector giving the quantiles of the warning lines. By default 'c(0.0225,0.9775)'.
`sl`	A vector optional giving the position of the tolerances. By default 'FALSE' (not drawn).
`numPlot`	gives the number of points visualised on the screen before the points are moving. Default value is '20'.
`seed`	Variable for `set.seed` to generate repeatable results. Default value is '125879'.
`rest`	Value for `Sys.sleep` i.e. break time in seconds between the plotting of the points. Default value is '0.1'.
`drift`	A logical value indicating whether a drift should be initialized or not. Default value is 'FALSE'.
`spread`	A logical value indicating whether a spread should be initialized or not. Default value is 'FALSE'.
`start`	A numeric value between 0 and 1. 'start' gives the percentage of passed plotted points after which a spread and/ or drift should be initialized. 'start' has no effect if both parameter 'spread' and 'drift' are set as 'FALSE'. Default value is '0'.

ENDdr	A positive numeric value. 'ENDdr' factorised with 'stdev' gives the value on which the drift will be end if the mean reaches there. Default value is '0'.
ENDsp	A positive numeric value. 'ENDdr' factorised with 'stdev' gives the value on which the spread will be end if the standard deviation reaches there. Default value is '0'.
speed	A postive numeric value. Controls as a factor the speed of the spread and/ or the drift.
...	Further graphical parameters see par

Details

On the left side of the output window the control chart is shown in which each violation of one of the 4 main Western Electric alarm rules ist spotted out with red pointmarks.
On the right upper side the violations for each rule are counted.

The 4 main Western Electric alarm rules are:

- Rule 1: One point outside of the control limits.
- Rule 2: Two of three consecutive points outside the warning limits but inside the control limits.
- Rule 3: Four of five consecutive points beyond the one-sigma limits.
- Rule 4: Eight consecutive points on one side of the center.

Value

'qrkSD' returns a data.frame in which the plotted regulated values are listed, as well as the single values, which are given in groups to 'FUN' to obtain the plotted values.

Author(s)

Thomas Roth: thomas.roth@tu-berlin.de
Etienne Stockhausen: stocdarf@mailbox.tu-berlin.de

References

- Vardeman, Stephen B. & Jobe, J. Marcus (1999) Statistical Quality Assurance Methods for Engineers, 1st ed. New York: John Wiley & Sons.
- Montgomery, D.C. (2005) Introduction to Statistical Quality Control, 5th ed. New York: John Wiley & Sons.

See Also

par
set.seed
Sys.sleep

Examples

```
par(ask=FALSE)
qrkSD(pch=20,n=1,FUN=mean,N=1000,rest=0,numPlot=25,drift=TRUE,spread=TRUE,
      start=0.05,ENDdr=1.5,ENDsp=1.5,speed=2)
par(ask=TRUE)
```

Index