

Package ‘smovie’

November 5, 2020

Type Package

Title Some Movies to Illustrate Concepts in Statistics

Version 1.1.3

Date 2020-11-05

Description Provides movies to help students to understand statistical concepts. The 'rpanel' package <<https://cran.r-project.org/package=rpanel>> is used to create interactive plots that move to illustrate key statistical ideas and methods. There are movies to: visualise probability distributions (including user-supplied ones); illustrate sampling distributions of the sample mean (central limit theorem), the median, the sample maximum (extremal types theorem) and (the Fisher transformation of the) product moment correlation coefficient; examine the influence of an individual observation in simple linear regression; illustrate key concepts in statistical hypothesis testing. Also provided are dpqr functions for the distribution of the Fisher transformation of the correlation coefficient under sampling from a bivariate normal distribution.

Depends R (>= 3.3.0), rpanel (>= 1.1-3)

License GPL (>= 2)

LazyData TRUE

Encoding UTF-8

RoxygenNote 7.1.0

Imports graphics, stats

Suggests knitr, numDeriv, revdbayes (>= 1.1.0), tkrplot, testthat, rmarkdown, SuppDists

VignetteBuilder knitr

URL <https://paulnorthrop.github.io/smovie/>,
<https://github.com/paulnorthrop/smovie/>

BugReports <https://github.com/paulnorthrop/smovie/issues>

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2020-11-05 13:10:03 UTC

R topics documented:

clt	2
cltq	5
continuous	7
correlation	10
discrete	11
ett	14
Fcorr	17
lev_inf	18
mean_vs_median	19
movies	21
shypo	22
smovie	24
wws	25
Index	29

clt	<i>Central Limit Theorem (CLT)</i>
-----	------------------------------------

Description

A movie to illustrate the ideas of the sampling distribution of a mean and the central limit theorem.

Usage

```
clt(
  n = 20,
  distn,
  params = list(),
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale,
  n_add = 1,
  delta_n = 1,
  arrow = TRUE,
  leg_cex = 1.25,
  ...
)
```

Arguments

n	An integer scalar. The size of the samples drawn from the distribution chosen using <code>distn</code> .
distn	A character scalar specifying the distribution from which observations are sampled. Distributions "beta", "binomial", "chisq", "chi-squared", "exponential", "f", "gamma", "geometric", "gev", "gp", "hypergeometric", "lognormal", "log-normal", "negative binomial", "normal", "poisson", "t", "uniform" and "weibull" are recognised, case being ignored. If <code>distn</code> is not supplied then <code>distn = "exponential"</code> is used. The "gev" and "gp" cases use the gev and gp distributional functions in the revdbayes package. The other cases use the distributional functions in the stats-package . If <code>distn = "gamma"</code> then the (shape,rate) parameterisation is used. If scale is supplied via <code>params</code> then rate is inferred from this. If <code>distn = "negative binomial"</code> then the (size,prob) parameterisation is used. If <code>mu</code> is supplied via <code>params</code> then <code>prob</code> is inferred from this (and size). If <code>distn = "beta"</code> then <code>ncp</code> is forced to be zero.
params	A named list of additional arguments to be passed to the density function associated with distribution <code>distn</code> . The (shape,rate) parameterisation is used for the gamma distribution (see GammaDist) even if the value of the scale parameter is set using <code>params</code> . If a parameter value is not supplied then the default values in the relevant distributional function set using <code>distn</code> are used, except for "beta" (shape1 = 2, shape2 = 2), "chisq" (df = 4), "f" (df1 = 4, df2 = 8), "gev" (shape = 0.2), "gamma" (shape = 2, "gp" (shape = 0.1), "poisson" (lambda = 5) and "t" (df = 4) and "weibull" (shape = 2).
panel_plot	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
hscale, vscale	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
n_add	An integer scalar. The number of simulated datasets to add to each new frame of the movie.
delta_n	A numeric scalar. The amount by which <code>n</code> is increased (or decreased) after one click of the + (or -) button in the parameter window.
arrow	A logical scalar. Should an arrow be included to show the simulated sample mean from the top plot being placed into the bottom plot?
leg_cex	The argument <code>cex</code> to legend . Allows the size of the legend to be controlled manually.
...	Additional arguments to the <code>rpanel</code> functions rp.button and rp.doublebutton , not including <code>panel</code> , <code>variable</code> , <code>title</code> , <code>step</code> , <code>action</code> , <code>initval</code> , <code>range</code> .

Details

Loosely speaking, a consequence of the [Central Limit Theorem](#) is that the mean of a **large number** of independent and identically distributed random variables, each with mean μ and finite standard

deviation σ , has **approximately** a normal distribution, even if these original variables are not normally distributed.

This movie considers examples where this limiting result holds and illustrates graphically the closeness of the limiting approximation provided by the relevant normal limit to the true finite- n distribution. Of course, when `distn = "normal"` this result is exact.

Samples of size n are repeatedly simulated from the distribution chosen using `distn`. These samples are summarized using a plot that appears at the top of the movie screen. For each sample the mean of these n values is calculated, stored and added to another plot, situated below the first plot. This plot is either a histogram or an empirical c.d.f., chosen using a radio button. A `rug` is added to a histogram provided that it contains no more than 1000 points.

The p.d.f. (for a continuous variable) or p.m.f. (for a discrete variable) of the original variables is added to the top plot.

Once it starts, four aspects of this movie are controlled by the user.

- There are buttons to increase (+) or decrease (-) the sample size, that is, the number of values over which a mean is calculated.
- Each time the button labelled "simulate another `n_add` samples of size `n`" is clicked `n_add` new samples are simulated and their sample mean are added to the bottom histogram.
- There is a button to switch the bottom plot from displaying a histogram of the simulated means and the limiting normal p.d.f. to the empirical c.d.f. of the simulated data and the limiting normal c.d.f.
- There is a checkbox to add to the bottom plot the approximate (large n) normal p.d.f./c.d.f. (with mean μ and standard deviation σ/\sqrt{n}), implied by the CLT.

Value

Nothing is returned, only the animation is produced.

See Also

[movies](#): a user-friendly menu panel.

[smovie](#): general information about smovie.

[cltq](#): Central Limit Theorem for sample quantiles.

Examples

```
# Exponential data
clt()

# Uniform data
clt(distn = "uniform")

# Poisson data
clt(distn = "poisson")
```

cltq

Central Limit Theorem (CLT) for sample quantiles

Description

A movie to illustrate the ideas of the sampling distribution of the sample 100*p*% quantile and the central limit theorem for sample quantiles.

Usage

```
cltq(
  n = 20,
  p = 0.5,
  distn,
  params = list(),
  type = 7,
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale,
  n_add = 1,
  delta_n = 1,
  arrow = TRUE,
  leg_cex = 1.25,
  ...
)
```

Arguments

<code>n</code>	An integer scalar. The size of the samples drawn from the distribution chosen using <code>distn</code> .
<code>p</code>	A numeric scalar in (0, 1). The value of <i>p</i> .
<code>distn</code>	A character scalar specifying the (continuous) distribution from which observations are sampled. Distributions "beta", "chisq", "chi-squared", "exponential", "f", "gamma", "gev", "gp", "lognormal", "log-normal", "normal", "t", "uniform" and "weibull" are recognised, case being ignored. If <code>distn</code> is not supplied then <code>distn = "exponential"</code> is used. The "gev" and "gp" cases use the gev and gp distributional functions in the revdbayes package. The other cases use the distributional functions in the stats-package . If <code>distn = "gamma"</code> then the (shape, rate) parameterisation is used. If scale is supplied via <code>params</code> then rate is inferred from this. If <code>distn = "beta"</code> then <code>ncp</code> is forced to be zero.
<code>params</code>	A named list of additional arguments to be passed to the density function associated with distribution <code>distn</code> . The (shape, rate) parameterisation is used for the gamma distribution (see GammaDist) even if the value of the scale parameter is set using <code>params</code> .

If a parameter value is not supplied then the default values in the relevant distributional function set using `distn` are used, except for "beta" (shape1 = 2, shape2 = 2), "chisq" (df = 4), "f" (df1 = 4, df2 = 8), "gev" (shape = 0.2), "gamma" (shape = 2), "gp" (shape = 0.1), "t" (df = 4) and "weibull" (shape = 2).

type	An integer between 1 and 9. The value of the argument type to be passed to quantile to when calculating a sample quantile.
panel_plot	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
hscale, vscale	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
n_add	An integer scalar. The number of simulated datasets to add to each new frame of the movie.
delta_n	A numeric scalar. The amount by which <code>n</code> is increased (or decreased) after one click of the + (or -) button in the parameter window.
arrow	A logical scalar. Should an arrow be included to show the simulated sample quantile from the top plot being placed into the bottom plot?
leg_cex	The argument <code>cex</code> to legend . Allows the size of the legend to be controlled manually.
...	Additional arguments to the <code>rpanel</code> functions <code>rp.button</code> and <code>rp.doublebutton</code> , not including <code>panel</code> , <code>variable</code> , <code>title</code> , <code>step</code> , <code>action</code> , <code>initval</code> , <code>range</code> .

Details

Loosely speaking, a consequence of the CLT for sample quantiles is that the $100p\%$ sample quantile of a **large number** of identically distributed random variables, each with probability density function f and $100p\%$ quantile $\xi(p)$, has **approximately** a normal distribution. See, for example, [Lehmann \(1999\)](#) for a precise statement and conditions.

This movie considers examples where this limiting result holds and illustrates graphically the closeness of the limiting approximation provided by the relevant normal limit to the true finite- n distribution.

Samples of size n are repeatedly simulated from the distribution chosen using `distn`. These samples are summarized using a plot that appears at the top of the movie screen. For each sample the $100p\%$ sample quantile of these n values is calculated, stored and added to another plot, situated below the first plot. This plot is either a histogram or an empirical c.d.f., chosen using a radio button. A [rug](#) is added to a histogram provided that it contains no more than 1000 points.

The p.d.f. of the original variables is added to the top plot.

Once it starts, four aspects of this movie are controlled by the user.

- There are buttons to increase (+) or decrease (-) the sample size, that is, the number of values for which a sample quantile is calculated.
- Each time the button labelled "simulate another `n_add` samples of size `n`" is clicked `n_add` new samples are simulated and their sample quantile are added to the bottom histogram.

- There is a button to switch the bottom plot from displaying a histogram of the simulated sample quantiles and the limiting normal p.d.f. to the empirical c.d.f. of the simulated data and the limiting normal c.d.f.
- There is a checkbox to add to the bottom plot the approximate (large n) normal p.d.f./c.d.f. implied by the CLT for sample quantiles: the mean is equal to $\xi(p)$ and standard deviation is equal to $\sqrt{p} \sqrt{q} / n f(\xi(p))$, where $q = 1 - p$.

Value

Nothing is returned, only the animation is produced.

References

Lehman, E. L. (1999) *Elements of Large-Sample Theory*, Springer-Verlag, London. <https://doi.org/10.1007/b98855>

See Also

[movies](#): a user-friendly menu panel.
[smovie](#): general information about smovie.
[clt](#): Central Limit Theorem.

Examples

```
# Exponential data
cltq()

# Uniform data
cltq(distn = "t", params = list(df = 2))
```

continuous

Univariate Continuous Distributions: p.d.f and c.d.f.

Description

A movie to illustrate how the probability density function (p.d.f.) and cumulative distribution function (c.d.f.) of a continuous random variable depend on the values of its parameters.

Usage

```
continuous(
  distn,
  var_range = NULL,
  params = list(),
  param_step = list(),
  param_range = list(),
  p_vec = NULL,
```

```

smallest = 0.01,
plot_par = list(),
panel_plot = TRUE,
hscale = NA,
vscale = hscale,
...
)

```

Arguments

- distn** Either a character string or a function to choose the continuous random variable. Strings "beta", "cauchy", "chisq" "chi-squared", "exponential", "f", "gamma", "gev", "gp", "lognormal", "log-normal", "normal", "t", "uniform" and "weibull" are recognised, case being ignored. The relevant distributional functions dxxx and pxxx in the [stats-package](#) are used. The abbreviations xxx are also recognised. The "gev" and "gp" cases use the [gev](#) and [gp](#) distributional functions in the [revdbayes](#) package. If distn = "gamma" then the (shape,rate) parameterisation is used, unless a value for scale is provided via the argument params when the (shape, scale) parameterisation is used. Valid functions are set up like a standard distributional function dxxx, with first argument x, last argument log and with arguments to set the parameters of the distribution in between. See the [CRAN task view on distributions](#). If distn is not supplied then distn = "normal" is used.
- var_range** A numeric vector of length 2. Can be used to set a fixed range of values over which to plot the p.d.f. and c.d.f., in order better to see the effects of changing the parameter values. If var_range is set then it overrides p_vec (see below).
- params** A named list of initial parameter values with which to start the movie. If distn is a string and a particular parameter value is not supplied then the following values are used. "beta": shape1 = 2, shape2 = 2, ncp = 0; "cauchy": location = 0, scale = 1; "chi-squared": df = 4, ncp = 0; "exponential": rate = 1; "f": df1 = 4, df2 = 8, ncp = 0; "gamma": shape = 2, rate = 1; "gev": loc = 0, scale = 1, shape = 0.1; "gp": loc = 0, scale = 1, shape = 0.1; "lognormal": meanlog = 0, sdlog = 1; "normal": mean = 0, sd = 1; "t": df = 4, ncp = 0; "uniform": min = 0, max = 1; "weibull": shape = 2, scale = 1. If distn is a function then params must set any required parameters. If parameter value is outside the corresponding range specified by param_range then it is set to the closest limit of the range.
- param_step** A named list of the amounts by which the respective parameters in params are increased/decreased after one click of the +/- button. If distn is a function then the default is 0.1 for all parameters. If distn is a string then a sensible distribution-specific default is set internally.
- param_range** A named list of the ranges over which the respective parameters in params are allowed to vary. Each element of the list should be a vector of length 2: the first element gives the lower limit of the range, the second element the upper limit. Use NA to impose no limit. If distn is a function then all parameters are unconstrained.

<code>p_vec</code>	A numeric vector of length 2. The p.d.f. and c.d.f. are plotted between the $100p_vec[1]\%$ and $100p_vec[2]\%$ quantiles of the distribution. If <code>p_vec</code> is not supplied then a sensible distribution-specific default is used. If <code>distn</code> is a function then the default is <code>p_vec = c(0.001, 0.999)</code> .
<code>smallest</code>	A positive numeric scalar. The smallest value to be used for any strictly positive parameters when <code>distn</code> is a string.
<code>plot_par</code>	A named list of graphical parameters (see par) to be passed to plot . This may be used to alter the appearance of the plots of the p.m.f. and c.d.f.
<code>panel_plot</code>	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
<code>hscale, vscale</code>	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
<code>...</code>	Additional arguments to be passed to rp.doublebutton , not including <code>panel</code> , <code>variable</code> , <code>title</code> , <code>step</code> , <code>action</code> , <code>initval</code> , <code>range</code> .

Details

The movie starts with a plot of the p.d.f. of the distribution for the initial values of the parameters. Buttons increase (+) or decrease (-) each parameter. There are radio buttons to switch the plot from the p.d.f. to the c.d.f. and back.

Value

Nothing is returned, only the animation is produced.

See Also

[movies](#): a user-friendly menu panel.
[smovie](#): general information about smovie.

Examples

```
# Normal example
continuous()
# Fix the range of values over which to plot
continuous(var_range = c(-10, 10))

# The same example, but using a user-supplied function and setting manually
# the initial parameters, parameter step size and range
continuous(distn = dnorm, params = list(mean = 0, sd = 1),
           param_step = list(mean = 1, sd = 1),
           param_range = list(sd = c(0, NA)))

# Gamma distribution. Show the use of var_range
continuous(distn = "gamma", var_range = c(0, 15))
```

correlation

*Sampling distribution of the correlation coefficient movie***Description**

A movie to illustrate how the sampling distribution of the product moment sample correlation coefficient r depends on the sample size n and on the true correlation ρ .

Usage

```
correlation(
  n = 30,
  rho = 0,
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale,
  delta_n = 1,
  delta_rho = 0.1,
  ...
)
```

Arguments

<code>n</code>	An integer scalar. The initial value of the sample size. Must not be less than 2.
<code>rho</code>	A numeric scalar. The initial value of the true correlation ρ . Must be in $[-1, 1]$.
<code>panel_plot</code>	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
<code>hscale, vscale</code>	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
<code>delta_n</code>	An integer scalar. The amount by which the value of the sample size is increased/decreased after one click of the +/- button.
<code>delta_rho</code>	A numeric scalar. The amount by which the value of rho is increased/decreased after one click of the +/- button.
<code>...</code>	Additional arguments to the <code>rpanel</code> functions <code>rp.button</code> and <code>rp.doublebutton</code> , not including <code>panel</code> , <code>variable</code> , <code>title</code> , <code>step</code> , <code>action</code> , <code>initval</code> , <code>range</code> .

Details

Random samples of size n are simulated from a bivariate normal distribution **bivariate normal distribution** in which each of the variables has a mean of 0 and a variance of 1 and the correlation ρ between the variables is chosen by the user.

The movie contains two plots. On the top is a scatter plot of the simulated sample, illustrating the strength of the association between the simulated values of the variables. A new sample is

produced by clicking "simulate another sample". For each simulated sample the sample (product moment) correlation coefficient r is calculated and displayed in the title of the top plot.

The values of the sample correlation coefficients are stored and are plotted in a histogram in the bottom plot. A rug displays the individual values, with the most recent value coloured red. As we accumulate a large number of values in this histogram the shape of the sampling distribution of r emerges. The exact p.d.f. of r is superimposed on this histogram, as is the value of ρ .

The bottom plot can be changed in two ways: (i) a radio button can be pressed to replace the histogram and pdf with a plot of the empirical c.d.f. and exact cdf; (ii) the variable can be changed from ρ to Fisher's z-transformation $F(\rho) = \operatorname{arctanh}(\rho) = [\ln(1+\rho) - \ln(1-\rho)]/2$. For sufficiently large values of n , $F(\rho)$ has approximately a normal distribution with mean ρ and variance $1/(n-3)$.

The values of the sample size n or true correlation coefficient ρ can be changed using the respective +/- buttons. If one of these is changed then the bottom plot is reset using the sample correlation coefficient of the first sample simulated using the new combination of n and ρ .

Value

Nothing is returned, only the animation is produced.

See Also

[movies](#): a user-friendly menu panel.

[smovie](#): general information about smovie.

Examples

```
correlation(rho = 0.8)
correlation(n = 10)
```

discrete

Univariate Discrete Distributions: p.m.f and c.d.f.

Description

A movie to illustrate how the probability mass function (p.m.f.) and cumulative distribution function (c.d.f.) of a discrete random variable depend on the values of its parameters.

Usage

```
discrete(
  distn,
  var_support = NULL,
  params = list(),
  param_step = list(),
  param_range = list(),
  p_vec = NULL,
  smallest = 0.01,
```

```

plot_par = list(),
panel_plot = TRUE,
hscale = NA,
vscale = hscale,
observed_value = NA,
...
)

```

Arguments

distn	<p>Either a character string or a function to choose the discrete random variable. Strings "binomial", "geometric", "hypergeometric", "negative binomial" and "poisson" are recognised, case being ignored. The relevant distributional functions dxxx and pxxx in the stats-package are used. The abbreviations xxx are also recognised. If distn = "negative binomial" then the (size,prob) parameterisation is used, unless a value for mu is provided via the argument params when the (size,mu) parameterisation is used.</p> <p>Valid functions are set up like a standard distributional function dxxx, with first argument x, last argument log and with arguments to set the parameters of the distribution in between. See the CRAN task view on distributions. It is assumed that the support of the random variable is a subset of the integers, unless var_support is set to the contrary.</p> <p>If distn is not supplied then distn = "binomial" is used.</p>
var_support	<p>A numeric vector. Can be used to set a fixed set of values for which to plot the p.m.f. and c.d.f., in order better to see the effects of changing the parameter values or to set a support that isn't a subset of the integers. If var_support is set then it overrides p_vec (see below).</p>
params	<p>A named list of initial parameter values with which to start the movie. If distn is a string and a particular parameter value is not supplied then the following values are used. "binomial": size = 10,prob = 0.5; "geometric": prob = 0.5; "hypergeometric": m = 10,n = 7,k = 8; "negative binomial": size = 10,prob = 0.5; "poisson": lambda = 5.</p> <p>If distn is a function then params must set any required parameters.</p> <p>If parameter value is outside the corresponding range specified by param_range then it is set to the closest limit of the range.</p>
param_step	<p>A named list of the amounts by which the respective parameters in params are increased/decreased after one click of the +/- button. If distn is a function then the default is 0.1 for all parameters. If distn is a string then a sensible distribution-specific default is set internally.</p>
param_range	<p>A named list of the ranges over which the respective parameters in params are allowed to vary. Each element of the list should be a vector of length 2: the first element gives the lower limit of the range, the second element the upper limit. Use NA to impose no limit. If distn is a function then all parameters are unconstrained.</p>
p_vec	<p>A numeric vector of length 2. The p.d.f. and c.d.f. are plotted between the 100p_vec[1]% and 100p_vec[2]% quantiles of the distribution. If p_vec is</p>

	not supplied then a sensible distribution-specific default is used. If <code>distn</code> is a function then the default is <code>p_vec = c(0.001, 0.999)</code> .
<code>smallest</code>	A positive numeric scalar. The smallest value to be used for any strictly positive parameters when <code>distn</code> is a string.
<code>plot_par</code>	A named list of graphical parameters (see par) to be passed to <code>plot</code> . This may be used to alter the appearance of the plots of the p.m.f. and c.d.f.
<code>panel_plot</code>	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
<code>hscale, vscale</code>	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
<code>observed_value</code>	A non-negative integer. If <code>observed_value</code> is supplied then the corresponding line in the plot of the p.m.f. is coloured in red.
<code>...</code>	Additional arguments to be passed to <code>rp.doublebutton</code> , not including <code>panel</code> , <code>variable</code> , <code>title</code> , <code>step</code> , <code>action</code> , <code>initval</code> , <code>range</code> .

Details

The movie starts with a plot of the p.m.f. of the distribution for the initial values of the parameters. Buttons increase (+) or decrease (-) each parameter. There are radio buttons to switch the plot from the p.m.f. to the c.d.f. and back.

If `distn == "geometric"` then there are radio buttons to switch between the version of the the geometric distribution based on the number of trials up to including the first success and the number of failures until the first success.

Owing to a conflict with the argument `size` of the function `rp.control` the parameter `size of`, for example, the binomial and negative binomial distributions, is labelled as `n`.

Value

Nothing is returned, only the animation is produced.

See Also

[movies](#): a user-friendly menu panel.

[smovie](#): general information about smovie.

Examples

```
# Binomial example
discrete()

# The same example, but using a user-supplied function and setting manually
# the initial parameters, parameter step size and range
discrete(distn = dbinom, params = list(size = 10, prob = 0.5),
         param_step = list(size = 1),
         param_range = list(size = c(1, NA), prob = c(0, 1)))
```

```
# Poisson distribution. Show the use of var_support
discrete(distn = "poisson", var_support = 0:20)
```

ett

Extremal Types Theorem (ETT)

Description

A movie to illustrate the extremal types theorem, that is, convergence of the distribution of the maximum of a random sample of size n from certain distributions to a member of the Generalized Extreme Value (GEV) family, as n tends to infinity. Samples of size n are simulated repeatedly from the chosen distribution. The distributions (simulated empirical and true) of the sample maxima are compared to the relevant GEV limit.

Usage

```
ett(
  n = 20,
  distn,
  params = list(),
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale,
  n_add = 1,
  delta_n = 1,
  arrow = TRUE,
  leg_cex = 1.25,
  ...
)
```

Arguments

n	An integer scalar. The size of the samples drawn from the distribution chosen using distn. n must be no smaller than 2.
distn	A character scalar specifying the distribution from which observations are sampled. Distributions "beta", "cauchy", "chisq", "chi-squared", "exponential", "f", "gamma", "gp", "lognormal", "log-normal", "ngev", "normal", "t", "uniform" and "weibull" are recognised, case being ignored. If distn is not supplied then distn = "exponential" is used. The "gp" case uses the gp distributional functions in the revdbayes package. The "ngev" case is a negated GEV($1/\xi$, 1, ξ) distribution, for $\xi > 0$, and uses the gev distributional functions in the revdbayes package. If $\xi = 1$ then this coincides with Example 1.7.5 in Leadbetter, Lindgren and Rootzen (1983). The other cases use the distributional functions in the stats-package . If distn = "gamma" then the (shape, rate) parameterisation is used. If scale is supplied via params then rate is inferred from this. If distn = "beta" then ncp is forced to be zero.

params	<p>A named list of additional arguments to be passed to the density function associated with distribution <code>distn</code>. The <code>(shape, rate)</code> parameterisation is used for the gamma distribution (see GammaDist) even if the value of the scale parameter is set using <code>params</code>.</p> <p>If a parameter value is not supplied then the default values in the relevant distributional function set using <code>distn</code> are used, except for "beta" (<code>shape1 = 2, shape2 = 2</code>), "chisq" (<code>df = 4</code>), "f" (<code>df1 = 4, df2 = 8</code>), "ngev" (<code>shape = 0.2</code>), "gamma" (<code>shape = 2</code>), "gp" (<code>shape = 0.1</code>), "t" (<code>df = 4</code>) and "weibull" (<code>shape = 2</code>).</p>
panel_plot	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
hscale, vscale	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
n_add	An integer scalar. The number of simulated datasets to add to each new frame of the movie.
delta_n	A numeric scalar. The amount by which <code>n</code> is increased (or decreased) after one click of the + (or -) button in the parameter window.
arrow	A logical scalar. Should an arrow be included to show the simulated sample maximum from the top plot being placed into the bottom plot?
leg_cex	The argument <code>cex</code> to legend . Allows the size of the legend to be controlled manually.
...	Additional arguments to the <code>rpanel</code> functions <code>rp.button</code> and <code>rp.doublebutton</code> , not including <code>panel</code> , <code>variable</code> , <code>title</code> , <code>step</code> , <code>action</code> , <code>initval</code> , <code>range</code> .

Details

Loosely speaking, a consequence of the **Extremal Types Theorem** is that, in many situations, the maximum of a *large number* n of independent random variables has *approximately* a $GEV(\mu, \sigma, \xi)$ distribution, where μ is a location parameter, σ is a scale parameter and ξ is a shape parameter. See Coles (2001) for an introductory account and Leadbetter et al (1983) for greater detail and more examples. The Extremal Types Theorem is an asymptotic result that considers the possible limiting distribution of linearly normalised maxima as n tends to infinity. This movie considers examples where this limiting result holds and illustrates graphically the closeness of the limiting approximation provided by the relevant GEV limit to the true finite- n distribution.

Samples of size n are repeatedly simulated from the distribution chosen using `distn`. These samples are summarized using a histogram that appears at the top of the movie screen. For each sample the maximum of these n values is calculated, stored and added to another plot, situated below the first plot. A `rug` is added to a histogram provided that it contains no more than 1000 points. This plot is either a histogram or an empirical c.d.f., chosen using a radio button.

The probability density function (p.d.f.) of the original variables is superimposed on the top histogram. There is a checkbox to add to the bottom plot the exact p.d.f./c.d.f. of the sample maxima and an approximate (large n) GEV p.d.f./c.d.f. implied by the ETT. The GEV shape parameter ξ that applies in the limiting case is used. The GEV location μ and scale σ are set based on constants used to normalise the maxima to achieve the GEV limit. Specifically, μ is set at the $100(1-1/n)\%$

quantile of the distribution `distn` and σ at $(1/n) / f(\mu)$, where f is the density function of the distribution `distn`.

Once it starts, four aspects of this movie are controlled by the user.

- There are buttons to increase (+) or decrease (-) the sample size, that is, the number of values over which a maximum is calculated.
- Each time the button labelled "simulate another `n_add` samples of size `n`" is clicked `n_add` new samples are simulated and their sample maxima are added to the bottom histogram.
- There is a button to switch the bottom plot from displaying a histogram of the simulated maxima, the exact p.d.f. and the limiting GEV p.d.f. to the empirical c.d.f. of the simulated data, the exact c.d.f. and the limiting GEV c.d.f.
- There is a box that can be used to display only the bottom plot. This option is selected automatically if the sample size n exceeds 100000.
- There is a box that can be used to display only the bottom plot. This option is selected automatically if the sample size n exceeds 100000.

For further detail about the examples specified by `distn` see Chapter 1 of Leadbetter et al. (1983) and Chapter 3 of Coles (2001). In many of these examples ("exponential", "normal", "gamma", "lognormal", "chi-squared", "weibull", "ngev") the limiting GEV distribution has a shape parameter that is equal to 0. In the "uniform" case the limiting shape parameter is -1 and in the "beta" case it is $-1 / \text{shape2}$, where `shape2` is the second parameter of the Beta distribution. In the other cases the limiting shape parameter is positive, with respective values `shape` ("gp", see `gp`), $1 / \text{df}$ ("t", see `TDist`), 1 ("cauchy", see `Cauchy`), $2 / \text{df2}$ ("f", see `FDist`).

Value

Nothing is returned, only the animation is produced.

References

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London. https://doi.org/10.1007/978-1-4471-3675-0_3

Leadbetter, M., Lindgren, G. and Rootzen, H. (1983) *Extremes and Related Properties of Random Sequences and Processes*. Springer-Verlag, New York. <https://doi.org/10.1007/978-1-4612-5449-2>

See Also

`movies`: a user-friendly menu panel.

`smovie`: general information about `smovie`.

Examples

```
# Exponential data: xi = 0
ett()
```

```
# Uniform data: xi = -1
ett(distn = "uniform")
```



```
# Student t data: xi = 1 / df
ett(distn = "t", params = list(df = 5))
```

Fcorr	<i>Fisher's transformation of the product moment correlation coefficient</i>
-------	--

Description

Density, distribution function, quantile function and random generator for the distribution of Fisher's transformation of product moment correlation, based on a random sample from a bivariate normal distribution

Usage

```
dFcorr(x, N, rho = 0, log = FALSE)

pFcorr(q, N, rho = 0, lower.tail = TRUE, log.p = FALSE)

qFcorr(p, N, rho = 0, lower.tail = TRUE, log.p = FALSE)

rFcorr(n, N, rho = 0, lower.tail = TRUE, log.p = FALSE)
```

Arguments

x, q	Numeric vectors of quantiles.
N	Numeric vector. Number of observations, (N > 3).
rho	Numeric vector. Population correlations, (-1 < rho < 1).
log, log.p	A logical scalar; if TRUE, probabilities p are given as log(p).
lower.tail	A logical scalar. If TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	A numeric vector of probabilities in [0,1].
n	Numeric scalar. The number of observations to be simulated. If length(n) > 1 then length(n) is taken to be the number required.

Details

These functions rely on the [correlation coefficient](#) functions in the SuppDists package. SuppDists must be installed in order for these functions to work.

References

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples of an indefinitely large population. *Biometrika*, **10**(4), 507-521.

Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron*, **1**, 3-32. <https://digital.library.adelaide.edu.au/dspace/bitstream/2440/15169/1/14.pdf>

See Also

[correlation coefficient](#) in the SuppDists package for dpqr functions for the untransformed product moment correlation coefficient.

[correlation](#): correlation sampling distribution movie.

Examples

```
dFcorr(-1:1, N = 10)
dFcorr(0, N = 11:20)

pFcorr(0.5, N = 10)
pFcorr(0.5, N = 10, rho = c(0, 0.3))

qFcorr((1:9)/10, N = 10, rho = 0.2)
qFcorr(0.5, N = c(10, 20), rho = c(0, 0.3))

rFcorr(6, N = 10, rho = 0.6)
```

lev_inf

Leverage and influence in simple linear regression movie

Description

A movie to examine the influence of a single outlying observation on a least squares regression line.

Usage

```
lev_inf(
  association = c("positive", "negative", "none"),
  n = 25,
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale
)
```

Arguments

association	A character scalar. Determines the type of association between (not-outlying) observations: "positive" for positive linear association; "negative" negative linear association; "none" for no association.
n	An integer scalar. The size of the sample of (non-outlying) observations.
panel_plot	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the tkrplot library is required.
hscale, vscale	Numeric scalars. Scaling parameters for the size of the plot when panel_plot = TRUE. The default values are 1.4 on Unix platforms and 2 on Windows platforms.

Details

n pairs of observations are simulated with the property that the mean of response variable y is a linear function of the values of the explanatory variable x . These pairs of observations are plotted using filled black circles. An extra observation is plotted using a filled red circle. Initially this observation is placed in the middle of the plot.

Superimposed on the plot are two least squares regression lines: one based on all the data ('with observation') and one in which the 'red' observation has been removed ('without observation'). Initially these lines coincide.

The location of the 'red' observation can be changed using the +/- buttons so that the effect of the position of this observation on the 'with observation' line can be seen.

We see that if the red observation is outlying, that is, it is far from the least squares line fitted to the other observations, then its **influence** on the least squares regression line depends on its x -coordinate. If its x -coordinate is much larger or smaller than the x -coordinate of the other observations (**high leverage**) then the influence is higher than if it has a similar x -coordinate to the other observations (**low leverage**). An observation with high leverage does not necessarily have high influence: if its y -coordinate falls very close to the regression line fitted to the other observations then its influence will be low.

Value

Nothing is returned, only the animation is produced.

See Also

[movies](#): a user-friendly menu panel.

[smovie](#): general information about smovie.

Examples

```
# Positive association
lev_inf()

# No association
lev_inf(association = "none")
```

mean_vs_median

Sample mean vs sample median

Description

A movie to compare the sampling distributions of the sample mean and sample median based on a random sample of size n from either a standard normal distribution or a standard Student's t distribution. An interesting comparison is between the normal and Student t with 2 degrees of freedom cases (see **Examples**).

Usage

```
mean_vs_median(
  n = 10,
  t_df = NULL,
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale,
  n_add = 1,
  delta_n = 1,
  arrow = TRUE,
  leg_cex = 1.75,
  ...
)
```

Arguments

<code>n</code>	An integer scalar. The size of the samples drawn from a standard normal distribution.
<code>t_df</code>	A positive scalar. The degrees of freedom <code>df</code> of a Student t distribution, as in TDist . If <code>t_df</code> is not supplied then data are simulated from a standard normal distribution.
<code>panel_plot</code>	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
<code>hscale</code> , <code>vscale</code>	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
<code>n_add</code>	An integer scalar. The number of simulated datasets to add to each new frame of the movie.
<code>delta_n</code>	A numeric scalar. The amount by which <code>n</code> is increased (or decreased) after one click of the + (or -) button in the parameter window.
<code>arrow</code>	A logical scalar. Should an arrow be included to show the simulated sample maximum from the top plot being placed into the bottom plot?
<code>leg_cex</code>	The argument <code>cex</code> to legend . Allows the size of the legend to be controlled manually.
<code>...</code>	Additional arguments to the <code>rpanel</code> functions rp.button and rp.doublebutton , not including <code>panel</code> , <code>variable</code> , <code>title</code> , <code>step</code> , <code>action</code> , <code>initval</code> , <code>range</code> .

Details

The movie is based on simulating repeatedly samples of size `n` from either a standard normal $N(0,1)$ distribution or a standard Student t distribution. The latter is selected by supplying the degrees of freedom of this distribution, using `t_df`. The movie contains three plots. The top plot contains a histogram of the most recently simulated dataset, with the relevant probability density function (p.d.f.) superimposed. A [rug](#) is added to a histogram provided that it contains no more than 1000 points.

Each time a sample is simulated the sample mean and sample median are calculated. These values are indicated on the top plot using an arrow (if `arrow = TRUE`) or a vertical (rug) line on the horizontal axis (`arrow = FALSE`), coloured red for the sample mean and blue for the sample median. If `arrow = TRUE` then the arrows show the positionings of most recent mean and median in the two plots below. If `arrow = FALSE` then the rug lines are replicated in these plots.

The plot in the middle contains a histogram of the sample means of *all* the simulated samples. The plot on the bottom contains a histogram of the sample medians of *all* the simulated samples. A `rug` is added to these histograms provided that they contains no more than 1000 points.

Once it starts, three aspects of this movie are controlled by the user.

- There are buttons to increase (+) or decrease (-) the sample size, that is, the number of values over which a maximum is calculated.
- Each time the button labelled "simulate another `n_add` samples of size `n`" is clicked `n_add` new samples are simulated and their sample mean are added to the bottom histogram.
- For the $N(0,1)$ case only, there is a checkbox to add to the bottom plot the p.d.f.s of the distribution of the sample mean and the (approximate, large `n`) distribution of the sample median.

Value

Nothing is returned, only the animation is produced.

See Also

`movies`: a user-friendly menu panel.

`smovie`: general information about `smovie`.

Examples

```
# Sampling from a standard normal distribution
mean_vs_median()

# Sampling from a standard t(2) distribution
mean_vs_median(t_df = 2)
```

`movies`

Main menu for `smovie` movies

Description

Uses the template `rp.cartoons` function to produce a menu panel from which any of the movies in `smovie` package can be launched. For greater control of an individual example call the relevant function directly.

Usage

```
movies(fixed_range = TRUE, hscale = NA, vscale = hscale)
```

Arguments

- `fixed_range` A logical scalar. Only relevant to the **Discrete** and **Continuous** menus. If TRUE then in the call to `discrete` or `continuous` the argument `var_support` (`discrete`) or `var_range` (`continuous`) is set so that the values on the horizontal axes are fixed at values that enable the movie to show the effects of changing the parameters of the distribution, at least locally to the default initial values for the parameters. For greater control call `discrete` or `continuous` directly.
- `hscale`, `vscale` Numeric scalars. Scaling parameters for the size of the plot when `panel_plot = TRUE`. The default values are 1.4 on Unix platforms and 2 on Windows platforms.

See Also

`discrete`, `continuous`, `clt`, `cltq`, `ett`, `mean_vs_median`, `correlation`, `lev_inf`, `wvs`, `shypo`.
`smovie`: general information about `smovie`.

Examples

```
movies()
```

shypo

Testing simple hypotheses

Description

A movie to illustrate statistical concepts involved in the testing of one simple hypothesis against another. The example used is a random sample from a normal distribution whose variance is assumed to be known. The simple hypotheses relate to the value of the mean μ .

Usage

```
shypo(
  mu0 = 0,
  sd = 6,
  eff = sd,
  n = 10,
  a = mu0 + eff/2,
  target_alpha = 0.05,
  target_beta = 0.1,
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale,
  delta_n = 1,
  delta_a = sd/(10 * sqrt(n)),
  delta_eff = sd,
  delta_mu0 = 1,
  delta_sd = 1
)
```

Arguments

<code>mu0</code>	A numeric scalar. The value of μ under the null hypothesis H_0 with which to start the movie.
<code>sd</code>	A positive numeric scalar. The (common) standard deviation σ of the normal distributions of the data under the two hypotheses.
<code>eff</code>	A numeric scalar. The <i>effect size</i> . The amount by which the value of μ under the alternative hypothesis is greater than the value <code>mu0</code> under the null hypothesis. That is, $\mu_1 = \text{eff} + \text{mu0}$. <code>eff</code> must be non-negative.
<code>n</code>	A positive integer scalar. The sample size with which to start the movie.
<code>a</code>	A numeric scalar. The critical value of the test with which to start the movie. H_0 is rejected if the sample mean is greater than <code>a</code> .
<code>target_alpha</code>	A numeric scalar in (0,1). The target value of the type I error to be achieved by setting <code>a</code> and/or <code>n</code> if the user asks for this using a radio button.
<code>target_beta</code>	A numeric scalar in (0,1). The target value of the type II error to be achieved by setting <code>a</code> and/or <code>n</code> if the user asks for this using a radio button.
<code>panel_plot</code>	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
<code>hscale, vscale</code>	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
<code>delta_mu0, delta_eff, delta_a, delta_n, delta_sd</code>	Numeric scalars. The respective amounts by which the values of <code>mu0</code> , <code>eff</code> , <code>a</code> , <code>n</code> and <code>sd</code> are increased (or decreased) after one click of the + (or -) button in the parameter window.

Details

The movie is based on two plots.

The top plot shows the (normal) probability density functions of the sample mean under the null hypothesis H_0 (mean `mu0`) and the alternative hypothesis H_1 (mean `mu1`, where $\mu_1 > \mu_0$), with the values of `mu0` and `mu1` indicated by vertical dashed lines. H_0 is rejected if the sample mean exceeds the critical value `a`, which is indicated by a vertical black line.

The bottom plot shows how the probabilities of making a type I or type II error (alpha and beta respectively) depend on the value of `a`, by plotting these probabilities against `a`.

A parameter window enables the user to change the values of `n`, `a`, `mu0`, `eff = mu1 - mu0` or `sd` by clicking the +/- buttons.

Radio buttons can be used either to:

- set `a` to achieve the target type I error probability `target_alpha`, based on the current value of `n`;
- set `a` and (integer) `n` to achieve (or better) the respective target type I and type II error probabilities of `target_alpha` and `target_beta`.

If `eff = 0` then a plot will be produced even though this case is not practically meaningful. In the "set `a` and `n` to achieve target alpha and beta" case, the plot will be the same as the case "set `a` and `n` by hand" case.

Value

Nothing is returned, only the animation is produced.

See Also

[movies](#): a user-friendly menu panel.
[smovie](#): general information about smovie.

Examples

```
# 1. Change a (for fixed n) to achieve alpha = 0.05
# 2. Change a and n to achieve alpha <= 0.05 and beta <= 0.1
shypo(mu0 = 0, eff = 5, n = 16, a = 2.3, delta_a = 0.01)
```

smovie

smovie: some movies to illustrate concepts in statistics

Description

These movies are animations used to illustrate key statistical ideas. They are produced using the [rpanel](#) package, which has [BWidget](#) as a system requirement. [BWidget](#) is included in the R installers for Windows and macOS. For other platforms please see [Section 1.1.7 of Writing R Extensions](#) for installation advice.

Details

When one of these functions is called R opens up a small *parameter window* containing clickable buttons that can be used to change parameters underlying the plot. For the effects of these buttons see the documentation of the individual functions.

See `vignette("smovie-vignette", package = "smovie")` for an overview of the package and the [user-friendly menu panel](#).

There are movies on the following topics.

Probability distributions

- [Discrete distributions](#)
- [Continuous distributions](#)

Sampling distributions

- [Central Limit Theorem: sampling distribution of a sample mean](#)
- [Central Limit Theorem for sample quantiles: sampling distribution of the 100p% sample quantile](#)
- [Sampling distributions of the mean and median](#)
- [Extremal Types Theorem: sampling distribution of a sample maximum](#)
- [Correlation coefficient](#)

Regression

- [Leverage and influence in simple linear regression](#)

Hypothesis testing

- [Wald, Wilks and Score tests](#)
- [Testing simple hypotheses](#)

References

Bowman, A., Crawford, E., Alexander, G. and Bowman, R. W. (2007). rpanel: Simple Interactive Controls for R Functions Using the tcltk Package. *Journal of Statistical Software*, **17(9)**, 1-18. <https://www.jstatsoft.org/v17/i09/>.

wws

Wald, Wilks and Score tests

Description

A movie to illustrate the nature of the Wald, Wilks and score likelihood-based test statistics, for a model with a scalar unknown parameter θ . The user can change the value of the parameter under a simple null hypothesis and observe the effect on the test statistics and (approximate) p-values associated with the tests of this hypothesis against the general alternative. The user can specify their own log-likelihood or use one of two in-built examples.

Usage

```
wws(
  model = c("norm", "binom"),
  theta_range = NULL,
  ...,
  mult = 3,
  theta0 = if (!is.null(theta_range)) sum(c(0.25, 0.75) * theta_range) else NULL,
  panel_plot = TRUE,
  hscale = NA,
  vscale = hscale,
  delta_theta0 = if (!is.null(theta_range)) abs(diff(theta_range))/20 else NULL,
  theta_mle = NULL,
  loglik = NULL,
  alg_score = NULL,
  alg_obs_info = NULL,
  digits = 3
)
```

Arguments

<code>model</code>	<p>A character scalar. Name of the the distribution on which one of two in-built examples are based.</p> <p>If <code>model = "norm"</code> then the setting is a random sample of size n from a normal distribution with unknown mean $\mu = \theta$ and known standard deviation σ.</p> <p>If <code>model = "binom"</code> then the setting is a random sample from a Bernoulli distribution with unknown success probability θ.</p> <p>The behaviour of these examples can be controlled using arguments supplied via <code>...</code>. In particular, the data can be supplied using <code>data</code>. If <code>model = "norm"</code> then n, μ, and σ can also be chosen. The default cases for these examples are:</p> <ul style="list-style-type: none"> • <code>model = "norm"</code>: $n = 10$, $\mu = 0$, $\sigma = 1$ and <code>data</code> contains a sample of a sample of size n simulated, using Normal, from a normal distribution with mean μ and standard deviation σ. • <code>model = "binom"</code>: <code>data = c(7, 13)</code>, that is, 7 successes and 13 failures observed in 20 trials. For the purposes of this movie there must be at least one success and at least one failure.
<code>theta_range</code>	A numeric vector of length 2. The range of values of θ over which to plot the log-likelihood. If <code>theta_range</code> is not supplied then the argument <code>mult</code> is used to set the range automatically.
<code>...</code>	Additional arguments to be passed to <code>loglik</code> , <code>alg_score</code> and <code>alg_obs_info</code> if <code>loglik</code> is supplied, or to functions functions relating to the in-built examples otherwise. See the description of <code>model</code> above for details.
<code>mult</code>	A positive numeric scalar. If <code>theta_range</code> is not supplied then an interval of width $2 \times \text{mult}$ standard errors centred on <code>theta_mle</code> is used. If <code>model = "binom"</code> then <code>theta_range</code> is truncated to $(0,1)$ if necessary.
<code>theta0</code>	A numeric scalar. The value of θ under the null hypothesis to use at the start of the movie.
<code>panel_plot</code>	A logical parameter that determines whether the plot is placed inside the panel (TRUE) or in the standard graphics window (FALSE). If the plot is to be placed inside the panel then the <code>tkrplot</code> library is required.
<code>hscale, vscale</code>	Numeric scalars. Scaling parameters for the size of the plot when <code>panel_plot = TRUE</code> . The default values are 1.4 on Unix platforms and 2 on Windows platforms.
<code>delta_theta0</code>	A numeric scalar. The amount by which the value of <code>theta0</code> is increased (or decreased) after one click of the + (or -) button in the parameter window.
<code>theta_mle</code>	A numeric scalar. The user may use this to supply the value of the maximum likelihood estimate (MLE) of θ . Otherwise, <code>optim</code> is used to search for the MLE, using <code>theta0</code> as the initial value and <code>theta_range</code> as bounds within which to search.
<code>loglik</code>	An R function, vectorised with respect to its first argument, that returns the value of the log-likelihood (up to an additive constant). The movie will not work if the observed information is not finite at the maximum likelihood estimate.
<code>alg_score</code>	A R function that returns the score function, that is, the derivative of <code>loglik</code> with respect to θ .

<code>alg_obs_info</code>	A R function that returns the observed information that is, the negated second derivative of <code>loglik</code> with respect to θ .
<code>digits</code>	An integer indicating the number of significant digits to be used in the displayed values of the test statistics and p-values. See signif .

Details

The **Wald**, **Wilks** (or likelihood ratio) and **Score** tests are asymptotically equivalent tests of a simple hypothesis that a parameter of interest θ is equal to a particular value θ_0 . The test statistics are all based on the log-likelihood $l(\theta)$ for θ but they differ in the way that they measure the distance between the maximum likelihood estimate (MLE) of θ and θ_0 . The Wilks statistic is the amount by which the log-likelihood evaluated θ_0 is smaller than the log-likelihood evaluated at the MLE. The Wald statistic is based on the absolute difference between the MLE and θ_0 . The score test is based on the gradient of the log-likelihood (the score function) at θ_0 . For details see Azzalini (1996).

This movie illustrates the differences between the test statistics for simple models with a single scalar parameter. In the (default) normal example the three test statistics coincide. This is not true in general, as shown by the other in-built example (`distn = "binom"`).

A user-supplied log-likelihood can be provided via `loglik`.

Value

Nothing is returned, only the animation is produced.

References

Azzalini, A. (1996) *Statistical Inference Based on the Likelihood*, Chapman & Hall / CRC, London.

See Also

[movies](#): a user-friendly menu panel.
[smovie](#): general information about smovie.

Examples

```
# N(theta, 1) example, test statistics equivalent
wvs(theta0 = 0.8)

# binomial(20, theta) example, test statistics similar
wvs(theta0 = 0.5, model = "binom")

# binomial(20, theta) example, test statistic rather different
# for theta0 distant from theta_mle
wvs(theta0 = 0.9, model = "binom", data = c(19, 1), theta_range = c(0.1, 0.99))

# binomial(2000, theta) example, test statistics very similar
wvs(theta0 = 0.5, model = "binom", data = c(1000, 1000))

set.seed(47)
x <- rnorm(10)
wvs(theta0 = 0.2, model = "norm", theta_range = c(-1, 1))
```

```
# Log-likelihood for a binomial experiment (up to an additive constant)
bin_loglik <- function(p, n_success, n_failure) {
  return(n_success * log(p) + n_failure * log(1 - p))
}

wws(loglik = bin_loglik, theta0 = 0.5, theta_range = c(0.1, 0.7),
    theta_mle = 7 / 20, n_success = 7, n_failure = 13)

bin_alg_score <- function(p, n_success, n_failure) {
  return(n_success / p - n_failure / (1 - p))
}
bin_alg_obs_info <- function(p, n_success, n_failure) {
  return(n_success / p ^ 2 + n_failure / (1 - p) ^ 2)
}
wws(loglik = bin_loglik, theta0 = 0.5, theta_range = c(0.1, 0.7),
    theta_mle = 7 / 20, n_success = 7, n_failure = 13,
    alg_score = bin_alg_score, alg_obs_info = bin_alg_obs_info)
```

Index

- Beta, [16](#)
- Cauchy, [16](#)
- Central Limit Theorem for sample quantiles: sampling distribution of the 100p% sample quantile, [24](#)
- Central Limit Theorem: sampling distribution of a sample mean, [24](#)
- clt, [2](#), [7](#), [22](#)
- cltq, [4](#), [5](#), [22](#)
- continuous, [7](#), [22](#)
- Continuous distributions, [24](#)
- correlation, [10](#), [18](#), [22](#)
- Correlation coefficient, [24](#)
- correlation coefficient, [17](#), [18](#)

- dFcorr (Fcorr), [17](#)
- discrete, [11](#), [22](#)
- Discrete distributions, [24](#)

- ett, [14](#), [22](#)
- Extremal Types Theorem: sampling distribution of a sample maximum, [24](#)

- Fcorr, [17](#)
- FDist, [16](#)

- GammaDist, [3](#), [5](#), [15](#)
- gev, [3](#), [5](#), [8](#), [14](#)
- gp, [3](#), [5](#), [8](#), [14](#), [16](#)

- legend, [3](#), [6](#), [15](#), [20](#)
- lev_inf, [18](#), [22](#)
- Leverage and influence in simple linear regression, [25](#)

- mean_vs_median, [19](#), [22](#)
- movies, [4](#), [7](#), [9](#), [11](#), [13](#), [16](#), [19](#), [21](#), [21](#), [24](#), [27](#)

- Normal, [26](#)
- optim, [26](#)

- par, [9](#), [13](#)
- pFcorr (Fcorr), [17](#)
- plot, [9](#), [13](#)

- qFcorr (Fcorr), [17](#)
- quantile, [6](#)

- revdbayes, [3](#), [5](#), [8](#), [14](#)
- rFcorr (Fcorr), [17](#)
- rp.button, [3](#), [6](#), [10](#), [15](#), [20](#)
- rp.cartoons, [21](#)
- rp.control, [13](#)
- rp.doublebutton, [3](#), [6](#), [9](#), [10](#), [13](#), [15](#), [20](#)
- rpanel, [24](#)
- rug, [4](#), [6](#), [15](#), [20](#), [21](#)

- Sampling distributions of the mean and median, [24](#)
- shypo, [22](#), [22](#)
- signif, [27](#)
- smovie, [4](#), [7](#), [9](#), [11](#), [13](#), [16](#), [19](#), [21](#), [22](#), [24](#), [24](#), [27](#)

- TDist, [16](#), [20](#)
- Testing simple hypotheses, [25](#)

- user-friendly menu panel, [24](#)

- Wald, Wilks and Score tests, [25](#)
- wws, [22](#), [25](#)