

# Package ‘simitation’

September 22, 2023

**Title** Simplified Simulations

**Version** 0.0.7

**Depends** R (>= 3.1.0)

**Description** Provides tools for generating and analyzing simulation studies. Users may easily specify all terms of a simulation study, often in a single line of code. Common univariate and bivariate methods, such as t tests, proportions tests, and chi squared tests, are integrated. Multivariate studies involving linear or logistic regression may also be specified with symbolic inputs. The simulation studies generate data for n observations in each of B experiments. Analyses of each experiment are integrated, and empirical results across the experiments are also provided.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Suggests** knitr, rmarkdown, covr, devtools, markdown, formatR, testthat  
(>= 3.0.0)

**VignetteBuilder** knitr

**Imports** data.table

**NeedsCompilation** no

**Author** David Shilane [aut],  
Srivastav Budugutta [ctb, cre],  
Mayur Bansal [ctb]

**Maintainer** Srivastav Budugutta <sb4788@columbia.edu>

**Repository** CRAN

**Config/testthat/edition** 3

**Date/Publication** 2023-09-22 16:00:07 UTC

## R topics documented:

analyze.simstudy.chisq.test.gf . . . . .	3
analyze.simstudy.chisq.test.ind . . . . .	4
analyze.simstudy.lm . . . . .	5

analyze.simstudy.logistic . . . . .	6
analyze.simstudy.prop . . . . .	8
analyze.simstudy.prop2 . . . . .	9
analyze.simstudy.t . . . . .	10
analyze.simstudy.t2 . . . . .	12
buildsim.binary . . . . .	13
buildsim.binomial . . . . .	14
buildsim.lm . . . . .	14
buildsim.logistic . . . . .	15
buildsim.normal . . . . .	15
buildsim.poisson . . . . .	16
buildsim.sample . . . . .	17
buildsim.uniform . . . . .	18
identify.distribution . . . . .	18
internal.chisq.test.gf . . . . .	19
internal.chisq.test.ind . . . . .	20
internal.prop.test . . . . .	20
internal.prop2.test . . . . .	21
internal.quantiles.mean.sd . . . . .	22
internal.statistics.one.lm . . . . .	22
internal.statistics.one.logistic . . . . .	23
internal.statistics.onelm . . . . .	23
internal.t.test . . . . .	24
internal.t2.test . . . . .	25
quantile.dt . . . . .	26
sim.chisq.gf . . . . .	26
sim.chisq.ind . . . . .	27
sim.chisq.test.gf . . . . .	29
sim.chisq.test.ind . . . . .	30
sim.norm . . . . .	31
sim.prop . . . . .	32
sim.prop.test . . . . .	33
sim.prop2 . . . . .	34
sim.prop2.test . . . . .	35
sim.statistics.lm . . . . .	36
sim.statistics.logistic . . . . .	37
sim.t . . . . .	38
sim.t.test . . . . .	39
sim.t2 . . . . .	41
sim.t2.test . . . . .	42
simstudy.chisq.test.gf . . . . .	43
simstudy.chisq.test.ind . . . . .	45
simstudy.lm . . . . .	46
simstudy.logistic . . . . .	49
simstudy.prop . . . . .	51
simstudy.prop2 . . . . .	52
simstudy.t . . . . .	54
simstudy.t2 . . . . .	55

**Index****59**

analyze.simstudy.chisq.test.gf

*analyze.simstudy.chisq.test.gf***Description**

This function analyzes the results of a simulated chi-squared test of goodness of fit.

**Usage**

```
analyze.simstudy.chisq.test.gf(
  test.statistics.chisq.test.gf,
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)
)
```

**Arguments**

`test.statistics.chisq.test.gf` A list containing summary information for fitting chi squared tests of goodness of fit. The structure is in the form returned by the function `simulation::sim.chisq.test.gf()`.

`conf.level` A numeric value between 0 and 1 representing the confidence level (1 - significance level). Default is 0.95.

`the.quantiles` A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles. Default values are `c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)`.

**Value**

A list containing the following elements:

- `stat.summary`: Summary statistics for the test statistics.
- `p.value.summary`: Proportions of tests that rejected and did not reject the null hypothesis.

**Examples**

```
simdat.chisq.gf <- sim.chisq.gf(n = 100, values = LETTERS[1:4],
  prob = c(0.4, 0.3, 0.2, 0.1), num.experiments = 2, experiment.name =
  "experiment_id", value.name = "classification", seed = 31)
test.statistics.chisq.test.gf <- sim.chisq.test.gf(simdat.chisq.gf =
  simdat.chisq.gf, hypothesized.probs = c(0.25, 0.3, 0.15, 0.3),
  correct = FALSE, experiment.name = "experiment_id",
  value.name = "classification")
```

```
analysis.chisq.gf <-
analyze.simstudy.chisq.test.gf(test.statistics.chisq.test.gf =
test.statistics.chisq.test.gf, conf.level = 0.95,
the.quantiles = c(0.25, 0.75))
```

---

```
analyze.simstudy.chisq.test.ind
      analyze.simstudy.chisq.test.ind
```

---

## Description

This function analyzes the results of a simulated chi-squared test of independence.

## Usage

```
analyze.simstudy.chisq.test.ind(
  test.statistics.chisq.test.ind,
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)
)
```

## Arguments

test.statistics.chisq.test.ind	A list containing summary information for fitting chi squared tests of independence. The structure is in the form returned by the function <code>simulation::sim.chisq.test.ind()</code> .
conf.level	A numeric value between 0 and 1 representing the confidence level (1 - significance level). Default is 0.95.
the.quantiles	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles. Default values are <code>c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)</code> .

## Value

A list containing the following elements:

- `stat.summary`: Summary statistics for the test statistics.
- `p.value.summary`: Proportions of tests that rejected and did not reject the null hypothesis.

## Examples

```
n <- c(50, 75, 100)
values <- LETTERS[1:4]
group.names <- paste0("group_", 1:3)
probs <- matrix(data = c(0.25, 0.25, 0.25, 0.25, 0.4, 0.3, 0.2,
0.1, 0.2, 0.4, 0.2, 0.2), nrow = length(n), byrow = TRUE)
```

```
simdat.chisq.ind <- sim.chisq.ind(n = c(50, 75, 100),
  values = LETTERS[1:4], probs = probs, num.experiments = 2,
  experiment.name = "exp_id", group.name = "treatment_group",
  group.values = paste0("group_", 1:3), value.name = "category", seed = 31)
test.statistics.chisq.test.ind <- sim.chisq.test.ind(simdat.chisq.ind =
  simdat.chisq.ind, correct = TRUE, experiment.name = "exp_id", group.name =
  "treatment_group", value.name = "category")
```

```
analysis.chisq.ind <- analyze.simstudy.chisq.test.ind(
  test.statistics.chisq.test.ind = test.statistics.chisq.test.ind, conf.level =
  0.95, the.quantiles = c(0.025, 0.975))
```

---

analyze.simstudy.lm    *Analyze Simulated Linear Regression Models*

---

## Description

This function analyzes the results of simulated linear regression models, providing various summary statistics about the model coefficients, fit, and other aspects.

## Usage

```
analyze.simstudy.lm(
  the.coefs,
  summary.stats,
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  coef.name = "Coefficient",
  estimate.name = "Estimate",
  lm.p.name = "Pr(>|t|)",
  f.p.name = "f.pvalue"
)
```

## Arguments

the.coefs	A data frame or data.table containing the summary table of estimated coefficients from repeated linear regression models. It should be structured like the output of <code>simulation::sim.statistics.lm\$the.coefs()</code> .
summary.stats	A data frame or data.table containing the summary statistics from repeated linear regression models, similar to <code>simulation::sim.statistics.lm\$summary.stats()</code> .
conf.level	A numeric value for the confidence level (1 - significance level). Default is 0.95.
the.quantiles	Numeric vector of quantile values for which statistics are required.
coef.name	Column name in 'the.coefs' that has input variable names of the regression model.
estimate.name	Column name in 'the.coefs' for estimated coefficients of the regression model.
lm.p.name	Column name in 'the.coefs' for p-values of coefficient tests.
f.p.name	Column name in 'summary.stats' for the F-test p-value.

**Value**

A list with several summary statistics for the linear regression model.

**Examples**

```

step.age <- "Age ~ N(45, 10)"
step.female <- "Female ~ binary(0.53)"
step.health.percentile <- "Health.Percentile ~ U(0,100)"
step.exercise.sessions <- "Exercise.Sessions ~ Poisson(2)"
step.diet <- "Diet ~ sample(('Light', 'Moderate', 'Heavy'),
(0.2, 0.45, 0.35))"
step.healthy.lifestyle <- "Healthy.Lifestyle ~
logistic(log(0.45) - 0.1 * (Age -45) + 0.05 * Female +
0.01 * Health.Percentile + 0.5 * Exercise.Sessions - 0.1 *
(Diet == 'Moderate') - 0.4 * (Diet == 'Heavy'))"

step.weight <- "Weight ~ lm(150 - 15 * Female + 0.5 * Age - 0.1 *
Health.Percentile - 0.2 * Exercise.Sessions + 5 * (Diet == 'Moderate') +
15 * (Diet == 'Heavy') - 2 * Healthy.Lifestyle + N(0, 10))"

the.steps <- c(step.age, step.female, step.health.percentile,
step.exercise.sessions, step.diet, step.healthy.lifestyle, step.weight)

simdat.multivariate <- simulation.steps(the.steps = the.steps,
n = 50, num.experiments = 2, experiment.name = "sim", seed = 41)

stats.lm <- sim.statistics.lm(simdat = simdat.multivariate, the.formula =
Weight ~ Age + Female + Health.Percentile + Exercise.Sessions +
Healthy.Lifestyle, grouping.variables = "sim")

analysis.lm <- analyze.simstudy.lm(the.coefs = stats.lm$the.coefs,
summary.stats = stats.lm$summary.stats, conf.level = 0.95,
the.quantiles = c(0.25, 0.75), coef.name = "Coefficient",
estimate.name = "Estimate", lm.p.name = "Pr(>|t|)", f.p.name = "f.pvalue")

```

---

analyze.simstudy.logistic

*Analyze Simulated Logistic Regression Models*

---

**Description**

This function analyzes the results of simulated logistic regression models, providing various summary statistics about the model coefficients, fit, and other aspects.

**Usage**

```
analyze.simstudy.logistic(
  the.coefs,
  summary.stats,
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  coef.name = "Coefficient",
  estimate.name = "Estimate",
  logistic.p.name = "Pr(>|z|)"
)
```

**Arguments**

the.coefs	A data frame or data.table containing the summary table of estimated coefficients from repeated logistic regression models. It should be structured like the output of <code>simulation::sim.statistics.logistic\$the.coefs()</code> .
summary.stats	A data.frame or data.table object of the summary statistics of repeated logistic regression models. Structure is in the form returned by the function <code>simulation::sim.statistics.logistic\$summary.stats()</code> .
conf.level	A numeric value between 0 and 1 representing the confidence level (1 - significance level).
the.quantiles	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
coef.name	A character value specifying the column of the.coefs that contains the names of the input variables of the logistic regression model.
estimate.name	A character value specifying the column of the.coefs that contains the estimated coefficients of the logistic regression model.
logistic.p.name	A character value specifying the column of the.coefs that contains the p-values for the tests of the estimated coefficients of the logistic regression model.

**Value**

A list with several summary statistics for the logistic regression model.

**Examples**

```
step.age <- "Age ~ N(45, 10)"
step.female <- "Female ~ binary(0.53)"
step.health.percentile <- "Health.Percentile ~ U(0,100)"
step.exercise.sessions <- "Exercise.Sessions ~ Poisson(2)"
step.diet <- "Diet ~ sample('Light', 'Moderate', 'Heavy'),
(0.2, 0.45, 0.35)"
step.healthy.lifestyle <- "Healthy.Lifestyle ~
logistic(log(0.45) - 0.1 * (Age -45) + 0.05 * Female +
0.01 * Health.Percentile + 0.5 * Exercise.Sessions - 0.1 *
(Diet == 'Moderate') - 0.4 * (Diet == 'Heavy'))"
```

```

step.weight <- "Weight ~ lm(150 - 15 * Female + 0.5 * Age - 0.1 *
Health.Percentile - 0.2 * Exercise.Sessions + 5 * (Diet == 'Moderate') +
15 * (Diet == 'Heavy') - 2 * Healthy.Lifestyle + N(0, 10))"

the.steps <- c(step.age, step.female, step.health.percentile,
step.exercise.sessions, step.diet, step.healthy.lifestyle, step.weight)

simdat.multivariate <- simulation.steps(the.steps = the.steps,
n = 50, num.experiments = 2, experiment.name = "sim", seed = 41)

stats.logistic <- sim.statistics.logistic(simdat =
simdat.multivariate, the.formula =
Healthy.Lifestyle ~ Age + Female + Health.Percentile + Exercise.Sessions,
grouping.variables = "sim")

analysis.logistic <- analyze.simstudy.logistic(the.coefs =
stats.logistic$the.coefs, summary.stats =
stats.logistic$summary.stats,
conf.level = 0.95, the.quantiles = c(0.1, 0.9))

```

---

analyze.simstudy.prop *Analyze Simulated Proportion Tests*

---

## Description

This function analyzes the results of simulated tests for proportions, providing various summary statistics about the test statistics, estimates, and confidence intervals.

## Usage

```

analyze.simstudy.prop(
  test.statistics.prop,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)
)

```

## Arguments

test.statistics.prop	A data frame or data.table containing the summary table of estimated coefficients from repeated proportion tests. Expected structure is similar to the output of <code>simulation::sim.prop.test()</code> .
alternative	A character string specifying the alternative hypothesis. Must be one of "two.sided", "less", or "greater". Default is "two.sided".



- conf.level      A numeric value between 0 and 1 representing the confidence level. Default is 0.95.
- the.quantiles    A numeric vector of values between 0 and 1. The function will return the specified quantiles for summary statistics.

**Value**

A list containing various summary statistics for the proportion test.

**Examples**

```
simdat.prop <- sim.prop(n = 30, p = 0.45,
  num.experiments = 2, experiment.name = "simulation_id",
  value.name = "success", seed = 104)

test.statistics.prop <- sim.prop.test(simdat.prop =
  simdat.prop, p = 0.5, alternative = "two.sided",
  conf.level = 0.99, correct = TRUE,
  experiment.name = "simulation_id", value.name = "success")

analysis.prop <- analyze.simstudy.prop(test.statistics.prop =
  test.statistics.prop, alternative = "two.sided", conf.level = 0.99,
  the.quantiles = c(0.005, 0.995))
```

---

```
analyze.simstudy.prop2
      analyze.simstudy.prop2
```

---

**Description**

analyze.simstudy.prop2

**Usage**

```
analyze.simstudy.prop2(
  test.statistics.prop2,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)
)
```

**Arguments**

test.statistics.prop2      Summary information for fitting two-sample tests of proportions. Structure is in the form returned by the function `simulation::sim.prop2.test()`.

alternative See help(prop.test).  
 conf.level See help(prop.test).  
 the.quantiles A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.

### Value

A list containing the following elements:

- estimate.summary: Summary statistics for the estimate.
- stat.summary: Summary statistics for the test statistic.
- p.value.summary: Data frame with proportions of tests that rejected and did not reject the null hypothesis.
- df.summary: Summary statistics for the degrees of freedom.
- ci.range.summary: Summary statistics for the confidence interval range.
- ci.proportion.above.null.summary: Summary statistics for the proportion of the upper confidence limit above the null value.
- ci.proportion.below.null.summary: Summary statistics for the proportion of the lower confidence limit below the null value.
- ci.limit.summary: Summary statistics for either the lower or upper confidence limit, depending on the specified alternative.

### Examples

```
simdat.prop2 <- sim.prop2(nx = 30, ny = 40, px = 0.5,
  py = 0.55, num.experiments = 2, experiment.name = "sim",
  group.name = "treatment", x.value = "group_1", y.value = "group_2",
  value.name = "correct_answer", seed = 3)

test.statistics.prop2 <- sim.prop2.test(simdat.prop2 = simdat.prop2,
  p = NULL, alternative = "less", conf.level = 0.95, correct = TRUE,
  experiment.name = "sim", group.name = "treatment", x.value = "group_1",
  y.value = "group_2", value.name = "correct_answer")

analysis.prop2 <- analyze.simstudy.prop2(test.statistics.prop2 =
  test.statistics.prop2, alternative = "less", conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.9, 0.975))
```

---

analyze.simstudy.t      *analyze.simstudy.t*

---

### Description

analyze.simstudy.t

**Usage**

```
analyze.simstudy.t(
  test.statistics.t,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)
)
```

**Arguments**

<code>test.statistics.t</code>	Summary information for fitting one-sample t tests. Structure is in the form returned by the function <code>simulation::sim.t.test()</code> .
<code>alternative</code>	See <code>help(t.test)</code> .
<code>conf.level</code>	See <code>help(t.test)</code> .
<code>the.quantiles</code>	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.

**Value**

A list containing the following elements:

- `estimate.summary`: Summary statistics for the estimate.
- `stat.summary`: Summary statistics for the t-test statistic.
- `p.value.summary`: Data frame with proportions of tests that rejected and did not reject the null hypothesis.
- `ci.range.summary`: Summary statistics for the confidence interval range.
- `ci.proportion.above.null.summary`: Summary statistics for the proportion of the upper confidence limit above the null value.
- `ci.proportion.below.null.summary`: Summary statistics for the proportion of the lower confidence limit below the null value.
- `ci.limit.summary`: (Only if 'alternative' is "greater" or "less") Summary statistics for either the lower or upper confidence limit, depending on the specified alternative.

**Examples**

```
simdat.t <- sim.t(n = 25, mean = 0.3, sd = 1, num.experiments = 2,
  experiment.name = "experiment", value.name = "x", seed = 2187)
```

```
test.statistics.t <- sim.t.test(simdat.t = simdat.t, alternative = "greater",
  mu = 0, conf.level = 0.95, experiment.name = "experiment", value.name = "x")
```

```
analysis.t <- analyze.simstudy.t(test.statistics.t = test.statistics.t,
  conf.level = 0.95, alternative = "greater", the.quantiles =
  c(0.025, 0.25, 0.25, 0.5, 0.75, 0.975))
```

---

analyze.simstudy.t2    *analyze.simstudy.t2*

---

## Description

analyze.simstudy.t2

## Usage

```
analyze.simstudy.t2(
  test.statistics.t2,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975)
)
```

## Arguments

test.statistics.t2	Summary information for fitting two-sample t tests. Structure is in the form returned by the function <code>simulation::sim.t2.test()</code> .
alternative	See <code>help(t.test)</code> .
conf.level	See <code>help(t.test)</code> .
the.quantiles	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.

## Value

A list containing the following elements:

- `estimate.summary`: Summary statistics for the estimate.
- `stat.summary`: Summary statistics for the t-test statistic.
- `df.summary`: Summary statistics for the degrees of freedom.
- `p.value.summary`: Data frame with proportions of tests that rejected and did not reject the null hypothesis.
- `ci.range.summary`: Summary statistics for the confidence interval range.
- `ci.proportion.above.null.summary`: Summary statistics for the proportion of the upper confidence limit above the null value.
- `ci.proportion.below.null.summary`: Summary statistics for the proportion of the lower confidence limit below the null value.
- `ci.limit.summary`: Summary statistics for either the lower or upper confidence limit, depending on the specified alternative.

## Examples

```
simdat.t2 <- sim.t2(nx = 30, ny = 40, meanx = 0, meany = 0.2, sdx = 1,
  sdy = 1, num.experiments = 2, experiment.name = "experiment",
  group.name = "group", x.value = "x", y.value = "y", value.name = "value",
  seed = 17)
```

```
test.statistics.t2 <- sim.t2.test(simdat.t2 = simdat.t2,
  alternative = "less", mu = 0, conf.level = 0.9,
  experiment.name = "experiment", group.name = "group",
  x.value = "x", y.value = "y", value.name = "value")
```

```
analysis.t2 <- analyze.simstudy.t2(test.statistics.t2 =
  test.statistics.t2, alternative = "less", conf.level = 0.9,
  the.quantiles = c(0.25, 0.5, 0.75))
```

---

buildsim.binary

*Internal function for Simulation for Binary Data*

---

## Description

This function is designed to generate binary data based on the provided formula. It is an internal function and is not meant for end-users.

## Usage

```
buildsim.binary(the.formula, the.variable, n, num.experiments = 1)
```

## Arguments

<code>the.formula</code>	A character string specifying the distribution function, e.g., "binary(0.5)".
<code>the.variable</code>	A character string naming the variable in the generated data.
<code>n</code>	An integer specifying the number of data points to generate for each experiment.
<code>num.experiments</code>	An integer specifying the number of experiments to simulate. Default is 1.

## Value

A data frame with simulated binary values based on the given formula.

---

buildsim.binomial      *Internal function for Simulation for Binomial Data*

---

### Description

This internal function is designed to generate binomial data based on the provided formula. It is not intended for direct usage by end-users.

### Usage

```
buildsim.binomial(the.formula, the.variable, n, num.experiments = 1)
```

### Arguments

the.formula	A character string specifying the distribution function, e.g., "Bin(10, 0.5)".
the.variable	A character string naming the variable in the generated data.
n	An integer specifying the number of data points to generate for each experiment.
num.experiments	An integer specifying the number of experiments to simulate. Default is 1.

### Value

A data frame with simulated binomial values based on the given formula.

---

buildsim.lm      *Internal function for Simulation for Linear Regression Data*

---

### Description

This internal function is designed to generate data based on a linear regression model specified by the provided formula. It is not intended for direct usage by end-users.

### Usage

```
buildsim.lm(dat, the.formula, the.variable, n, num.experiments = 1)
```

### Arguments

dat	A data.frame or data.table containing the variables referenced in the.formula.
the.formula	A character string specifying the linear regression function, e.g., "lm(0.5 * X + 1.2 * Y + N(0,2))".
the.variable	A character string naming the variable in the generated data.
n	An integer specifying the number of data points to generate for each experiment.
num.experiments	An integer specifying the number of experiments to simulate. Default is 1.

**Value**

A data frame with simulated linear regression values based on the given formula.

---

`buildsim.logistic`      *Internal function for Simulation for Logistic Regression Data*

---

**Description**

This internal function is designed to generate data based on a logistic regression model specified by the provided formula. It is not intended for direct usage by end-users.

**Usage**

```
buildsim.logistic(dat, the.formula, the.variable, n, num.experiments = 1)
```

**Arguments**

<code>dat</code>	A data.frame or data.table containing the variables referenced in the.formula.
<code>the.formula</code>	A character string specifying the logistic regression function, e.g., "logistic(0.5 * X + 1.2 * Y)".
<code>the.variable</code>	A character string naming the variable in the generated data.
<code>n</code>	An integer specifying the number of data points to generate for each experiment.
<code>num.experiments</code>	An integer specifying the number of experiments to simulate. Default is 1.

**Value**

A data frame with simulated logistic regression values based on the given formula.

---

`buildsim.normal`      *Internal Simulation for Normally Distributed Data*

---

**Description**

This internal function generates data based on a normal distribution specified by the provided formula. It is not intended for direct usage by end-users.

**Usage**

```
buildsim.normal(the.formula, the.variable, n, num.experiments)
```

**Arguments**

<code>the.formula</code>	A character string specifying the normal distribution function, e.g., "N(0,1)".
<code>the.variable</code>	A character string naming the variable in the generated data.
<code>n</code>	An integer specifying the number of data points to generate for each experiment.
<code>num.experiments</code>	An integer specifying the number of experiments to simulate.

**Value**

A data frame with simulated normally distributed values based on the given formula.

---

<code>buildsim.poisson</code>	<i>Internal function for Simulation for Poisson Distributed Data</i>
-------------------------------	--

---

**Description**

This internal function generates data based on a Poisson distribution specified by the provided formula. It is not intended for direct usage by end-users.

**Usage**

```
buildsim.poisson(the.formula, the.variable, n, num.experiments)
```

**Arguments**

<code>the.formula</code>	A character string specifying the Poisson distribution function, e.g., "poisson(3)".
<code>the.variable</code>	A character string naming the variable in the generated data.
<code>n</code>	An integer specifying the number of data points to generate for each experiment.
<code>num.experiments</code>	An integer specifying the number of experiments to simulate.

**Value**

A data frame with simulated Poisson distributed values based on the given formula.



---

buildsim.sample	<i>Internal function for Sampling Function</i>
-----------------	--

---

### Description

This internal function generates samples based on the specified distributions and probabilities. It is not intended for direct usage by end-users.

### Usage

```
buildsim.sample(  
  the.formula,  
  the.variable,  
  n,  
  num.experiments,  
  value.split = ",",  
  symbol.open.paren = "(",  
  symbol.close.paren = ")"  
)
```

### Arguments

<code>the.formula</code>	A character string specifying the sampling formula, e.g., "sample(('Red', 'Green', 'Blue'), (0.5, 0.3, 0.2))".
<code>the.variable</code>	A character string naming the variable in the generated data.
<code>n</code>	An integer specifying the number of data points to generate for each experiment.
<code>num.experiments</code>	An integer specifying the number of experiments to simulate.
<code>value.split</code>	A character used to split the values in the sample.
<code>symbol.open.paren</code>	A character specifying the opening parenthesis.
<code>symbol.close.paren</code>	A character specifying the closing parenthesis.

### Value

A data frame with sampled values based on the given formula.

---

buildsim.uniform	<i>Internal function for Simulation for Uniform Distributed Data</i>
------------------	--

---

### Description

This internal function generates data based on a Uniform distribution specified by the provided formula. It is not intended for direct usage by end-users.

### Usage

```
buildsim.uniform(the.formula, the.variable, n, num.experiments = 1)
```

### Arguments

the.formula	A character string specifying the Uniform distribution function, e.g., "U(0,1)".
the.variable	A character string naming the variable in the generated data.
n	An integer specifying the number of data points to generate for each experiment.
num.experiments	An integer specifying the number of experiments to simulate.

### Value

A data frame with simulated Uniform distributed values based on the given formula.

---

identify.distribution	<i>Internal function for Distribution Identification</i>
-----------------------	--

---

### Description

This internal function identifies the type of distribution based on a given formula and simulates data accordingly. It is not intended for direct usage by end-users.

### Usage

```
## S3 method for class 'distribution'
identify(
  dat = NULL,
  the.step,
  n,
  num.experiments,
  step.split = "~",
  value.split = ",",
)
```

**Arguments**

dat	Optional data table for generating data.
the.step	A character string specifying the formula for simulation.
n	An integer specifying the number of data points to generate.
num.experiments	An integer specifying the number of experiments to simulate.
step.split	A character indicating the delimiter for splitting the step formula.
value.split	A character used to split values in certain distributions.

**Value**

A data table with simulated values based on the identified distribution.

---

internal.chisq.test.gf

*Internal function for Chi-Squared Test of Goodness of Fit*

---

**Description**

Computes the chi-squared test for the given categorical data.

**Usage**

```
internal.chisq.test.gf(x, hypothesized.probs = NULL, correct = TRUE)
```

**Arguments**

x	A categorical variable.
hypothesized.probs	Hypothesized probabilities for each level of x.
correct	A logical indicating if a continuity correction should be applied.

**Value**

A data frame containing the test statistic, degrees of freedom, and p-value.

---

`internal.chisq.test.ind`*Internal function for Chi-Squared Test of Independence*

---

**Description**

Computes the chi-squared test for the given data.

**Usage**

```
internal.chisq.test.ind(the.data, group.name, value.name, correct = TRUE)
```

**Arguments**

<code>the.data</code>	The data table.
<code>group.name</code>	Group variable name.
<code>value.name</code>	Value variable name.
<code>correct</code>	A logical indicating if a continuity correction should be applied.

**Value**

A data frame containing the test statistic, degrees of freedom, and p-value.

---

`internal.prop.test`*Internal function for One-sample Proportions Test*

---

**Description**

Computes the test for proportion for the given binary data.

**Usage**

```
internal.prop.test(  
  x,  
  p = NULL,  
  alternative = c("two.sided", "less", "greater"),  
  conf.level = 0.95,  
  correct = TRUE  
)
```

**Arguments**

x	A binary variable.
p	Null hypothesis value for proportion. Default is NULL.
alternative	A character string specifying the alternative hypothesis. One of "two.sided", "less", or "greater". Default is "two.sided".
conf.level	A numeric value between 0 and 1 indicating the confidence level for the interval estimate of the proportion. Default is 0.95.
correct	A logical indicating if Yates' continuity correction should be applied for the test. Default is TRUE.

**Value**

A data frame with test results.

---

internal.prop2.test    *Internal function for Two-sample Proportions Test*

---

**Description**

Computes the test for proportions for two given binary variables.

**Usage**

```
internal.prop2.test(
  x,
  y,
  p = NULL,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  correct = TRUE,
  na.rm = TRUE
)
```

**Arguments**

x	First binary variable.
y	Second binary variable.
p	Null hypothesis value for proportion. Default is NULL.
alternative	A character string specifying the alternative hypothesis. One of "two.sided", "less", or "greater". Default is "two.sided".
conf.level	A numeric value between 0 and 1 indicating the confidence level for the interval estimate of the proportion. Default is 0.95.
correct	A logical indicating if Yates' continuity correction should be applied for the test. Default is TRUE.
na.rm	A logical indicating if NA values should be removed. Default is TRUE.

**Value**

A data frame with test results.

---

```
internal.quantiles.mean.sd
```

*Internal function for Quantile, Mean, and Standard Deviation Calculation*

---

**Description**

Computes the specified quantiles, mean, and standard deviation for the given data.

**Usage**

```
internal.quantiles.mean.sd(x, the.quantiles, na.rm = TRUE)
```

**Arguments**

x	A numeric vector.
the.quantiles	A numeric vector of quantile values.
na.rm	A logical indicating if missing values should be removed.

**Value**

A data table with summary statistics.

---

```
internal.statistics.one.lm
```

*Internal function for Summary Statistics of Linear Model*

---

**Description**

Computes the summary statistics for the linear model fit on the given data.

**Usage**

```
internal.statistics.one.lm(the.data, the.formula)
```

**Arguments**

the.data	The data table.
the.formula	A formula specifying the linear model.

**Value**

A list containing the following components:

- `coef.table`A data table containing the coefficients, their estimates, standard error, t-values, and associated p-values.
- `summary.stats`A data table containing summary statistics for the linear model. This includes the sigma, degrees of freedom, root mean square error (rse), R-squared, adjusted R-squared, F-statistic, F-statistic degrees of freedom, and the p-value for the F-statistic.

---

`internal.statistics.one.logistic`

*Internal function for Summary Statistics of Logistic Regression*

---

**Description**

Computes the summary statistics for the logistic regression fit on the given data.

**Usage**

```
internal.statistics.one.logistic(the.data, the.formula)
```

**Arguments**

<code>the.data</code>	The data table.
<code>the.formula</code>	A formula specifying the logistic regression model.

**Value**

A list containing the coefficient table and summary statistics.

---

`internal.statistics.onelm`

*Internal function for Summary Statistics of Linear Model*

---

**Description**

Computes the summary statistics for the linear model fit on the given data.

**Usage**

```
internal.statistics.onelm(the.data, the.formula)
```

**Arguments**

<code>the.data</code>	The data table.
<code>the.formula</code>	A formula specifying the linear model.

**Value**

A list containing the coefficient table and summary statistics.

---

internal.t.test	<i>Internal function for One-sample t-test</i>
-----------------	--

---

**Description**

Computes the one-sample t-test for the given data.

**Usage**

```
internal.t.test(  
  x,  
  alternative = c("two.sided", "less", "greater"),  
  mu = 0,  
  paired = FALSE,  
  var.equal = FALSE,  
  conf.level = 0.95  
)
```

**Arguments**

x	A numeric vector.
alternative	A character string specifying the alternative hypothesis. One of "two.sided", "less", or "greater". Default is "two.sided".
mu	A number indicating the true value of the mean (or difference in means if you are performing a two-sample test). Default is 0.
paired	A logical indicating whether you want a paired t-test. Default is FALSE.
var.equal	A logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used. Default is FALSE.
conf.level	A numeric value between 0 and 1 indicating the confidence level for the interval estimate of the mean. Default is 0.95.

**Value**

A data frame with test results.



---

internal.t2.test      *Internal function for Two-sample t-test*

---

### Description

Computes the two-sample t-test for the given data.

### Usage

```
internal.t2.test(  
  x,  
  y,  
  alternative = c("two.sided", "less", "greater"),  
  mu = 0,  
  paired = FALSE,  
  var.equal = FALSE,  
  conf.level = 0.95  
)
```

### Arguments

x	First numeric vector.
y	Second numeric vector.
alternative	A character string specifying the alternative hypothesis. One of "two.sided", "less", or "greater". Default is "two.sided".
mu	A number indicating the true value of the mean difference (relevant if paired = TRUE). Default is 0.
paired	A logical indicating whether you want a paired t-test. Default is FALSE.
var.equal	A logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used. Default is FALSE.
conf.level	A numeric value between 0 and 1 indicating the confidence level for the interval estimate of the mean difference. Default is 0.95.

### Value

A data frame with test results.

---

`quantile.dt`*Internal function for Quantiles Calculation*

---

**Description**

Computes the specified quantiles for the given data.

**Usage**

```
## S3 method for class 'dt'  
quantile(x, probs, na.rm = TRUE)
```

**Arguments**

<code>x</code>	A numeric vector.
<code>probs</code>	A numeric vector of quantile values.
<code>na.rm</code>	A logical indicating if missing values should be removed.

**Value**

A data table with quantile values.

---

`sim.chisq.gf`*sim.chisq.gf*

---

**Description**

Simulate data for chi-squared tests of goodness of fit across experiments.

**Usage**

```
sim.chisq.gf(  
  n,  
  values,  
  prob = NULL,  
  num.experiments = 1,  
  experiment.name = "experiment",  
  value.name = "x",  
  seed = 91,  
  vstr = "3.6"  
)
```

**Arguments**

n	A numeric value indicating the number of observations in each experiment.
values	A numeric vector specifying the possible values (sample space).
prob	A numeric vector of probabilities corresponding to the values for simulation. If not provided, equal probabilities are assumed for all values.
num.experiments	An integer indicating the number of simulated experiments to conduct.
experiment.name	A character string specifying the column name for identifying each experiment in the output.
value.name	A character string specifying the column name for the simulated values in the output.
seed	An integer specifying the seed for reproducibility. Default is 91.
vstr	A numeric or character string specifying the seed for random number generation to ensure reproducibility. Default is "3.6". For more details, refer to <a href="#">set.seed</a> .

**Value**

A 'data.table' containing the simulated experiments with specified column names.

**Examples**

```
simdat.chisq.gf <- sim.chisq.gf(n = 100, values = LETTERS[1:4],
  prob = c(0.4, 0.3, 0.2, 0.1), num.experiments = 2, experiment.name =
  "experiment_id", value.name = "classification", seed = 31)
```

---

sim.chisq.ind	<i>sim.chisq.ind</i>
---------------	----------------------

---

**Description**

sim.chisq.ind

**Usage**

```
sim.chisq.ind(
  n,
  values,
  probs,
  num.experiments = 2,
  experiment.name = "experiment",
  group.name = "group",
  group.values = NULL,
  value.name = "value",
  seed = 8272,
  vstr = 3.6
)
```

**Arguments**

<code>n</code>	A vector of sample sizes for the different groups.
<code>values</code>	A vector of values specifying the sample space.
<code>probs</code>	A matrix of probabilities used to simulate the values in each group. The rows of the probs matrix correspond to the groups, while the columns correspond to the values.
<code>num.experiments</code>	A numeric value representing the number of simulated experiments.
<code>experiment.name</code>	A character value providing the name for the column identifying the experiment.
<code>group.name</code>	A character value providing the name of the column of the group labels.
<code>group.values</code>	A vector of unique values that identify the different groups, e.g. <code>c("x", "y", "z")</code> . If <code>NULL</code> , then values "x1", "x2", ..., "xk" will be assigned for the k groups specified.
<code>value.name</code>	A character value providing the name for the simulated values.
<code>seed</code>	A single numeric value, interpreted as an integer, or <code>NULL</code> . See <code>help(set.seed)</code> .
<code>vstr</code>	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if <code>vstr</code> is greater than the current version. See <code>help(set.seed)</code> .

**Value**

A data table with the following columns:

- `experiment.name`: The name of the experiment, repeated according to the number of experiments and group sample sizes.
- `group.name`: The name of the group for each observation.
- `value.name`: The simulated value for each observation.

**Examples**

```
n <- c(50, 75, 100)
values <- LETTERS[1:4]
group.names <- paste0("group_", 1:3)
probs <- matrix(data =
c(0.25, 0.25, 0.25, 0.25, 0.4, 0.3, 0.2, 0.1, 0.2, 0.4, 0.2, 0.2),
nrow = length(n), byrow = TRUE)
simdat.chisq.ind <- sim.chisq.ind(n = c(50, 75, 100),
values = LETTERS[1:4], probs = probs, num.experiments = 2,
experiment.name = "exp_id", group.name = "treatment_group",
group.values = paste0("group_", 1:3), value.name = "category", seed = 31)
```

---

sim.chisq.test.gf      *sim.chisq.test.gf*

---

## Description

Perform a chi-squared test of goodness of fit across one or more experiments.

## Usage

```
sim.chisq.test.gf(
  simdat.chisq.gf,
  hypothesized.probs = NULL,
  correct = TRUE,
  experiment.name = "experiment",
  value.name = "x"
)
```

## Arguments

`simdat.chisq.gf` Data for use in chi squared tests of goodness of fit across one or more experiments. The structure should be in the form returned by the function `simitation::sim.chisq.gf()`.

`hypothesized.probs` A vector of hypothesized probabilities corresponding to the values in the column specified by `value.name`. If the values include `c("B", "A", "C")`, then a probability vector of `c(0.5, 0.3, 0.2)` would associate a value of 0.5 with "A", 0.3 with "B", and 0.2 with "C".

`correct` Logical. For details, refer to the `chisq.test` documentation.

`experiment.name` A character value providing the name of the column identifying the experiment.

`value.name` A character value providing the name of the column identifying the values.

## Value

A `data.table` or `data.frame` with the results of the chi-squared tests.

## Examples

```
simdat.chisq.gf <- sim.chisq.gf(n = 100, values = LETTERS[1:4], prob =
c(0.4, 0.3, 0.2, 0.1), num.experiments = 2,
experiment.name = "experiment_id", value.name = "classification", seed = 31)

test.statistics.chisq.test.gf <- sim.chisq.test.gf(simdat.chisq.gf =
simdat.chisq.gf, hypothesized.probs = c(0.25, 0.3, 0.15, 0.3),
correct = FALSE, experiment.name = "experiment_id", value.name =
"classification")
```

---

sim.chisq.test.ind     *sim.chisq.test.ind*

---

## Description

sim.chisq.test.ind

## Usage

```
sim.chisq.test.ind(
  simdat.chisq.ind,
  correct = TRUE,
  experiment.name = "experiment",
  group.name = "variable",
  value.name = "value"
)
```

## Arguments

simdat.chisq.ind	Data for use in chi squared tests of independence across one or more experiments. Structure is in the form returned by the function <code>simulation::sim.chisq.ind()</code> .
correct	See <code>help(chisq.test)</code> .
experiment.name	A character value providing the name of the column identifying the experiment.
group.name	A character value providing the name of the column of the group labels.
value.name	A character value providing the name of the column identifying the values.

## Value

A data frame with results of the chi-squared tests for each experiment.

## Examples

```
n <- c(50, 75, 100)
values <- LETTERS[1:4]
group.names <- paste0("group_", 1:3)
probs <- matrix(data =
  c(0.25, 0.25, 0.25, 0.25, 0.4, 0.3, 0.2, 0.1, 0.2, 0.4, 0.2, 0.2),
  nrow = length(n), byrow = TRUE)
simdat.chisq.ind <- sim.chisq.ind(n = c(50, 75, 100),
  values = LETTERS[1:4], probs = probs, num.experiments = 2,
  experiment.name = "exp_id", group.name = "treatment_group",
  group.values = paste0("group_", 1:3), value.name = "category", seed = 31)
```

```
test.statistics.chisq.test.ind <- sim.chisq.test.ind(simdat.chisq.ind =
simdat.chisq.ind, correct = TRUE, experiment.name = "exp_id",
group.name = "treatment_group", value.name = "category")
```

---

sim.norm

*Internal function for Normal Distribution Simulation*


---

## Description

Simulates data from normal distributions given specified parameters.

## Usage

```
sim.norm(
  n.values,
  mean.values,
  sd.values,
  num.experiments = 1,
  variable.names = NULL,
  seed = 1978,
  vstr = 3.6
)
```

## Arguments

n.values	A numeric vector indicating the number of values to be simulated for each normal distribution.
mean.values	A numeric vector indicating the mean values for each normal distribution.
sd.values	A numeric vector indicating the standard deviation values for each normal distribution.
num.experiments	A single integer indicating the number of experiments to simulate. Default is 1.
variable.names	A character vector with names for the variables. If NULL, default names "x1", "x2", ... will be used.
seed	An integer to set as the seed for reproducibility. Default is 1978.
vstr	A character string specifying the RNG version. Default is "3.6".

## Value

A data.table containing the simulated data.

---

 sim.prop
 

---

*sim.prop***Description**

sim.prop

**Usage**

```
sim.prop(
  n,
  p = 0.5,
  num.experiments = 1,
  experiment.name = "experiment",
  value.name = "x",
  seed = 2470,
  vstr = 3.6
)
```

**Arguments**

n	A numeric value for the number of observations in each experiment.
p	A numeric value for the probability of success.
num.experiments	A numeric value representing the number of simulated experiments.
experiment.name	A character value providing the name for the column identifying the experiment.
value.name	A character value providing the name for the simulated values.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

**Value**

A data.table containing simulated values based on the probability of success for each experiment. The output will have two columns: one indicating the experiment and another containing the simulated values.

**Examples**

```
simdat.prop <- sim.prop(n = 30, p = 0.45, num.experiments = 2,
  experiment.name = "simulation_id", value.name = "success", seed = 104)
```



---

sim.prop.test	<i>sim.prop.test</i>
---------------	----------------------

---

## Description

sim.prop.test

## Usage

```
sim.prop.test(
  simdat.prop,
  p = NULL,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  correct = TRUE,
  experiment.name = "experiment",
  value.name = "x"
)
```

## Arguments

simdat.prop	Data for use in one-sample proportions tests across one or more experiments. Structure is in the form returned by the function <code>simulation::sim.prop()</code> .
p	See <code>help(prop.test)</code> .
alternative	See <code>help(prop.test)</code> .
conf.level	See <code>help(prop.test)</code> .
correct	See <code>help(prop.test)</code> .
experiment.name	A character value providing the name of the column identifying the experiment.
value.name	A character value providing the name of the column identifying the values.

## Value

A `data.table` containing test statistics for each experiment based on the one-sample proportions test. Columns typically include experiment name, statistic, p-value, and other relevant metrics as returned by the internal testing function.

## Examples

```
simdat.prop <- sim.prop(n = 30, p = 0.45, num.experiments = 2,
  experiment.name = "simulation_id", value.name = "success", seed = 104)

test.statistics.prop <- sim.prop.test(simdat.prop = simdat.prop, p = 0.5,
  alternative = "two.sided", conf.level = 0.99, correct = TRUE,
  experiment.name = "simulation_id", value.name = "success")
```

---

 sim.prop2

*sim.prop2*


---

## Description

sim.prop2

## Usage

```
sim.prop2(
  nx,
  ny,
  px = 0.5,
  py = 0.5,
  num.experiments = 1,
  experiment.name = "experiment",
  group.name = "group",
  x.value = "x",
  y.value = "y",
  value.name = "value",
  seed = 3471,
  vstr = 3.6
)
```

## Arguments

nx	A numeric value for the number of observations in the x group for each experiment.
ny	A numeric value for the number of observations in the y group for each experiment.
px	A numeric value for the probability of success in the x group.
py	A numeric value for the probability of success in the y group.
num.experiments	A numeric value representing the number of simulated experiments.
experiment.name	A character value providing the name for the column identifying the experiment.
group.name	A character value providing the name of the column of the group labels.
x.value	A character value specifying the label used for data in the x group (in the column labeled by the group.name parameter).
y.value	A character value specifying the label used for data in the y group (in the column labeled by the group.name parameter).
value.name	A character value specifying the name of the column that contains the value of the simulated data.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).

`vstr` A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if `vstr` is greater than the current version. See `help(set.seed)`.

### Value

A 'data.table' with simulated data for two groups ('x' and 'y') based on binomial distributions. The table contains columns corresponding to the experiment identifier, group labels ('x' or 'y').

### Examples

```
simdat.prop2 <- sim.prop2(nx = 30, ny = 40, px = 0.5,
  py = 0.55, num.experiments = 2, experiment.name = "sim",
  group.name = "treatment", x.value = "group_1", y.value = "group_2",
  value.name = "correct_answer", seed = 3)
```

---

<code>sim.prop2.test</code>	<i>sim.prop2.test</i>
-----------------------------	-----------------------

---

### Description

`sim.prop2.test`

### Usage

```
sim.prop2.test(
  simdat.prop2,
  p = NULL,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  correct = TRUE,
  experiment.name = "experiment",
  group.name = "group",
  x.value = "x",
  y.value = "y",
  value.name = "value"
)
```

### Arguments

<code>simdat.prop2</code>	Data for use in two-sample proportions tests across one or more experiments. Structure is in the form returned by the function <code>simulation::sim.prop2()</code> .
<code>p</code>	See <code>help(prop.test)</code> .
<code>alternative</code>	See <code>help(prop.test)</code> .
<code>conf.level</code>	See <code>help(prop.test)</code> .
<code>correct</code>	See <code>help(prop.test)</code> .

experiment.name	A character value providing the name of the column identifying the experiment.
group.name	A character value providing the name of the column of the group labels.
x.value	A character value providing a label for the first group in the two-sample t test in the column of data named by group.name.
y.value	A character value providing a label for the second group in the two-sample t test in the column of data named by group.name.
value.name	A character value providing the name of the column identifying the values.

**Value**

A data.table containing test statistics for each experiment based on the two-sample proportions test.

**Examples**

```
simdat.prop2 <- sim.prop2(nx = 30, ny = 40, px = 0.5, py = 0.55,
  num.experiments = 2, experiment.name = "sim", group.name = "treatment",
  x.value = "group_1", y.value = "group_2", value.name = "correct_answer",
  seed = 3)
```

```
test.statistics.prop2 <- sim.prop2.test(simdat.prop2 = simdat.prop2,
  p = NULL, alternative = "less", conf.level = 0.95, correct = TRUE,
  experiment.name = "sim", group.name = "treatment", x.value = "group_1",
  y.value = "group_2", value.name = "correct_answer")
```

---

sim.statistics.lm      *sim.statistic.lm*

---

**Description**

sim.statistic.lm

**Usage**

```
sim.statistics.lm(simdat, the.formula, grouping.variables)
```

**Arguments**

simdat	Data for use in multivariable regression models across one or more experiments. Structure is in the form returned by the function <code>simulation::simulation.steps()</code> .
the.formula	A formula object or character value specifying the formula for the regression model.
grouping.variables	A character vector of column names from <code>simdat</code> on which to group the data. The intended regression model will be fit in groups based on this selection.

**Value**

A list containing: 'the.coefs': A data.table of regression coefficients for each group and predictor variable. 'summary.stats': A data.table of summary statistics for each group, typically including metrics such as R-squared, residuals, etc.

**Examples**

```

step.age <- "Age ~ N(45, 10)"
step.female <- "Female ~ binary(0.53)"
step.health.percentile <- "Health.Percentile ~ U(0,100)"
step.exercise.sessions <- "Exercise.Sessions ~ Poisson(2)"
step.diet <- "Diet ~ sample(('Light', 'Moderate', 'Heavy'),
(0.2, 0.45, 0.35))"

step.healthy.lifestyle <- "Healthy.Lifestyle ~
logistic(log(0.45) - 0.1 * (Age -45) + 0.05 * Female +
0.01 * Health.Percentile + 0.5 * Exercise.Sessions - 0.1 *
(Diet == 'Moderate') - 0.4 * (Diet == 'Heavy'))"

step.weight <- "Weight ~ lm(
150 - 15 * Female +
0.5 * Age - 0.1 * Health.Percentile - 0.2 * Exercise.Sessions +
5 * (Diet == 'Moderate') + 15 * (Diet == 'Heavy') - 2 *
Healthy.Lifestyle + N(0, 10))"

the.steps <- c(step.age, step.female, step.health.percentile,
step.exercise.sessions, step.diet, step.healthy.lifestyle, step.weight)

simdat.multivariate <- simulation.steps(the.steps = the.steps,
n = 50, num.experiments = 2, experiment.name = "sim", seed = 41)

stats.lm <- sim.statistics.lm(simdat = simdat.multivariate,
the.formula = Weight ~ Age + Female + Health.Percentile + Exercise.Sessions +
Healthy.Lifestyle, grouping.variables = "sim")

```

---

sim.statistics.logistic

*sim.statistics.logistic*

---

**Description**

sim.statistics.logistic

**Usage**

```
sim.statistics.logistic(simdat, the.formula, grouping.variables)
```

**Arguments**

simdat	Data for use in multivariable regression models across one or more experiments. Structure is in the form returned by the function <code>simulation::simulation.steps()</code> .
the.formula	A formula object or character value specifying the formula for the regression model.
grouping.variables	A character vector of column names from <code>simdat</code> on which to group the data. The intended regression model will be fit in groups based on this selection.

**Value**

A list containing: \* `'the.coefs'`: A data frame of logistic regression coefficients for each group and predictor variable. \* `'summary.stats'`: A data frame of summary statistics for each group, such as the Akaike Information Criterion (AIC), deviance, residuals, and other metrics.

**Examples**

```

step.age <- "Age ~ N(45, 10)"
step.female <- "Female ~ binary(0.53)"
step.health.percentile <- "Health.Percentile ~ U(0,100)"
step.exercise.sessions <- "Exercise.Sessions ~ Poisson(2)"
step.diet <- "Diet ~ sample(('Light', 'Moderate', 'Heavy'),
(0.2, 0.45, 0.35))"

step.healthy.lifestyle <- "Healthy.Lifestyle ~ logistic(log(0.45) - 0.1 * (Age -45)
+ 0.05 * Female + 0.01 * Health.Percentile + 0.5 * Exercise.Sessions - 0.1 * (Diet
== 'Moderate') - 0.4 * (Diet == 'Heavy'))"

step.weight <- "Weight ~ lm(150 - 15 * Female + 0.5 * Age - 0.1 *
Health.Percentile - 0.2 * Exercise.Sessions + 5 * (Diet == 'Moderate') +
15 * (Diet == 'Heavy') - 2 * Healthy.Lifestyle + N(0, 10))"

the.steps <- c(step.age, step.female, step.health.percentile,
step.exercise.sessions, step.diet, step.healthy.lifestyle, step.weight)

simdat.multivariate <- simulation.steps(the.steps = the.steps, n = 50,
num.experiments = 2, experiment.name = "sim", seed = 41)

stats.logistic <- sim.statistics.logistic(simdat = simdat.multivariate,
the.formula = Healthy.Lifestyle ~ Age + Female + Health.Percentile +
Exercise.Sessions, grouping.variables = "sim")

```

---

sim.t

*sim.t*


---

**Description**

sim.t

**Usage**

```
sim.t(
  n,
  mean = 0,
  sd = 1,
  num.experiments = 1,
  experiment.name = "experiment",
  value.name = "x",
  seed = 7261,
  vstr = 3.6
)
```

**Arguments**

n	A numeric value for the number of observations in each experiment.
mean	A numeric value for the expected value of the data to be simulated.
sd	A numeric value for the standard deviation of the data to be simulated.
num.experiments	A numeric value representing the number of simulated experiments.
experiment.name	A character value providing the name for the column identifying the experiment.
value.name	A character value providing the name for the simulated values.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

**Value**

A data frame with simulated data for a t-test. The returned table will have columns corresponding to 'experiment.name' and 'value.name'.

**Examples**

```
simdat.t <- sim.t(n = 25, mean = 0.3, sd = 1, num.experiments = 2,
  experiment.name = "experiment", value.name = "x", seed = 2187)
```

---

 sim.t.test

---

 sim.t.test

---

**Description**

sim.t.test

**Usage**

```
sim.t.test(
  simdat.t,
  alternative = c("two.sided", "less", "greater"),
  mu = 0,
  paired = FALSE,
  var.equal = FALSE,
  conf.level = 0.95,
  experiment.name = "experiment",
  value.name = "x"
)
```

**Arguments**

simdat.t	Data for use in one-sample t tests across one or more experiments. Structure is in the form returned by the function <code>simulation::sim.t()</code> .
alternative	See <code>help(t.test)</code> .
mu	See <code>help(t.test)</code>
paired	See <code>help(t.test)</code>
var.equal	See <code>help(t.test)</code>
conf.level	See <code>help(t.test)</code>
experiment.name	A character value providing the name of the column identifying the experiment.
value.name	A character value providing the name of the column identifying the values.

**Value**

A data frame or `data.table` containing the results of one-sample t-tests performed on the data generated by the 'sim.t' function.

**Examples**

```
simdat.t <- sim.t(n = 25, mean = 0.3, sd = 1, num.experiments = 2,
  experiment.name = "experiment", value.name = "x", seed = 2187)
```

```
test.statistics.t <- sim.t.test(simdat.t = simdat.t, alternative = "greater",
  mu = 0, conf.level = 0.95, experiment.name = "experiment", value.name = "x")
```



---

sim.t2	<i>sim.t2</i>
--------	---------------

---

## Description

sim.t2

## Usage

```
sim.t2(
  nx,
  ny,
  meanx = 0,
  meany = 1,
  sdx = 1,
  sdy = 1,
  num.experiments = 1,
  experiment.name = "experiment",
  group.name = "group",
  x.value = "x",
  y.value = "y",
  value.name = "value",
  seed = 3471,
  vstr = 3.6
)
```

## Arguments

nx	A numeric value for the number of observations in the x group for each experiment.
ny	A numeric value for the number of observations in the y group for each experiment.
meanx	A numeric value for the expected value of the x group used in the simulation.
meany	A numeric value for the expected value of the y group used in the simulation.
sdx	A numeric value for the standard deviation of the x group used in the simulation.
sdy	A numeric value for the standard deviation of the y group used in the simulation.
num.experiments	A numeric value representing the number of simulated experiments.
experiment.name	A character value providing the name for the column identifying the experiment.
group.name	A character value providing the name of the column of the group labels.
x.value	A character value specifying the label used for data in the x group (in the column labeled by the group.name parameter).
y.value	A character value specifying the label used for data in the y group (in the column labeled by the group.name parameter).

value.name	A character value specifying the name of the column that contains the value of the simulated data.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

### Value

A data frame or data.table with simulated data for two independent groups. Columns in the returned table correspond to the experiment identifier, group labels ('x' or 'y'), and the simulated values for each group. @examples `simdat.t2 <- sim.t2(nx = 30, ny = 40, meanx = 0, meany = 0.2, sdx = 1, sdy = 1, num.experiments = 2, experiment.name = "experiment", group.name = "group", x.value = "x", y.value = "y", value.name = "value", seed = 17)`

---

sim.t2.test	<i>sim.t2.test</i>
-------------	--------------------

---

### Description

sim.t2.test

### Usage

```
sim.t2.test(
  simdat.t2,
  alternative = c("two.sided", "less", "greater"),
  mu = 0,
  paired = FALSE,
  var.equal = FALSE,
  conf.level = 0.95,
  experiment.name = "experiment",
  group.name = "group",
  x.value = "x",
  y.value = "y",
  value.name = "value"
)
```

### Arguments

simdat.t2	Data for use in two-sample t tests across one or more experiments. Structure is in the form returned by the function <code>simulation::sim.t2()</code> .
alternative	See help(t.test).
mu	See help(t.test)
paired	See help(t.test)
var.equal	See help(t.test)

conf.level	See help(t.test)
experiment.name	A character value providing the name for the column identifying the experiment.
group.name	A character value providing the name of the column of the group labels.
x.value	A character value providing a label for the first group in the two-sample t test in the column of data named by group.name.
y.value	A character value providing a label for the second group in the two-sample t test in the column of data named by group.name.
value.name	A character value providing the name of the column of the values.

**Value**

A data frame or data.table containing the results of two-sample t-tests performed on the data generated by the 'sim.t2' function.

**Examples**

```
simdat.t2 <- sim.t2(nx = 30, ny = 40, meanx = 0, meany = 0.2, sdx = 1,
  sdy = 1, num.experiments = 2, experiment.name = "experiment",
  group.name = "group", x.value = "x", y.value = "y", value.name = "value",
  seed = 17)

test.statistics.t2 <- sim.t2.test(simdat.t2 = simdat.t2,
  alternative = "less", mu = 0, conf.level = 0.9,
  experiment.name = "experiment", group.name = "group", x.value = "x",
  y.value = "y", value.name = "value")
```

---

```
simstudy.chisq.test.gf
      simstudy.chisq.test.gf
```

---

**Description**

simstudy.chisq.test.gf

**Usage**

```
simstudy.chisq.test.gf(
  n,
  values,
  actual.probs,
  hypothesized.probs = NULL,
  num.experiments = 1,
  conf.level = 0.95,
  correct = TRUE,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
```

```

  experiment.name = "experiment",
  value.name = "x",
  seed = 7261,
  vstr = 3.6
)

```

## Arguments

<code>n</code>	A numeric value for the number of observations in each experiment.
<code>values</code>	A vector of values specifying the sample space.
<code>actual.probs</code>	A vector of probabilities used to simulate the values.
<code>hypothesized.probs</code>	A vector of hypothesized probabilities for the values.
<code>num.experiments</code>	A numeric value representing the number of simulated experiments.
<code>conf.level</code>	A numeric value between 0 and 1 representing the confidence level (1 - significance level).
<code>correct</code>	See <code>help(chisq.test)</code> .
<code>the.quantiles</code>	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
<code>experiment.name</code>	A character value providing the name for the column identifying the experiment.
<code>value.name</code>	A character value providing the name for the simulated values.
<code>seed</code>	A single numeric value, interpreted as an integer, or NULL. See <code>help(set.seed)</code> .
<code>vstr</code>	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if <code>vstr</code> is greater than the current version. See <code>help(set.seed)</code> .

## Value

A list containing three elements: `'simdat'`: A data frame of the simulated data based on chi-squared distributions. `'test.statistics'`: A data frame or `data.table` containing the results of chi-squared tests performed on the simulated data. `'sim.analysis'`: Summary statistics or analysis results from the simulated chi-squared tests.

## Examples

```

study.chisq.gf <- simstudy.chisq.test.gf(n = 75, values = LETTERS[1:4],
actual.probs = c(0.3, 0.3, 0.2, 0.2),
hypothesized.probs = rep.int(x = 0.25, times = 4),
num.experiments = 2, conf.level = 0.95, correct = FALSE,
the.quantiles = c(0.25, 0.75), experiment.name = "experiment_id",
value.name = "classification", seed = 61)

```

---

 simstudy.chisq.test.ind

*simstudy.chisq.test.ind*


---

## Description

simstudy.chisq.test.ind

## Usage

```
simstudy.chisq.test.ind(
  n,
  values,
  probs,
  num.experiments = 1,
  conf.level = 0.95,
  correct = TRUE,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  experiment.name = "experiment",
  group.name = "group",
  group.values = NULL,
  value.name = "value",
  seed = 403,
  vstr = 3.6
)
```

## Arguments

n	A vector of sample sizes for the different groups.
values	A vector of values specifying the sample space.
probs	A matrix of probabilities used to simulate the values in each group. The rows of the probs matrix correspond to the groups, while the columns correspond to the values.
num.experiments	A numeric value representing the number of simulated experiments.
conf.level	A numeric value between 0 and 1 representing the confidence level (1 - significance level).
correct	See help(chisq.test).
the.quantiles	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
experiment.name	A character value providing the name for the column identifying the experiment.
group.name	A character value providing the name of the column of the group labels.

group.values	A vector of unique values that identify the different groups, e.g. c("x", "y", "z"). If NULL, then values "x1", "x2", ..., "xk" will be assigned for the k groups specified.
value.name	A character value providing the name for the simulated values.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

### Value

A list containing three elements: `simdat.chisq.ind`: A data frame of the simulated data based on chi-squared distributions for independent groups. `test.statistics.chisq.test.ind`: A data frame containing the results of chi-squared tests for independence performed on the simulated data. `sim.analysis`: Summary statistics or analysis results from the chi-squared tests for independence.

### Examples

```
n <- c(50, 75, 100)
values <- LETTERS[1:4]
group.names <- paste0("group_", 1:3)
probs <- matrix(data = c(0.25, 0.25, 0.25, 0.25, 0.4, 0.3,
0.2, 0.1, 0.2, 0.4, 0.2, 0.2),
nrow = length(n), byrow = TRUE)

study.chisq.ind <- simstudy.chisq.test.ind(n = c(30, 35, 40),
values = LETTERS[1:4], probs = probs, num.experiments = 2,
conf.level = 0.95, correct = TRUE, the.quantiles = c(0.025, 0.975),
experiment.name = "exp_id", group.name = "treatment_group",
group.values = paste0("group_", 1:3), value.name = "category",
seed = 77)
```

---

simstudy.lm

*simstudy.lm*

---

### Description

simstudy.lm

### Usage

```
simstudy.lm(
  the.steps,
  n,
  num.experiments,
  the.formula,
  conf.level = 0.95,
```

```

the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
experiment.name = "experiment",
step.split = "~",
coef.name = "Coefficient",
estimate.name = "Estimate",
lm.p.name = "Pr(>|t|)",
f.p.name = "f.pvalue",
seed = 41,
vstr = 3.6
)

```

## Arguments

the.steps	A character vector of variables to simulate. The variables are simulated in the order specified. Later variables can be generated to depend on earlier variables. The possible specifications include: Normal "X ~ N(100, 5)" with the mean and SD. Uniform "X ~ U(0, 100)" with the minimum and maximum. Poisson "X ~ Poisson(3)" with the mean. Binary "X ~ Binary(0.5)" with the probability of success. Binomial "X ~ Bin(10, 0.2)" with the number of trials and probability of success. Categorical "Diet ~ sample(c('Light', 'Moderate', 'Heavy'), (0.2, 0.45, 0.35))" with the values in the first set of parentheses and their respective probabilities in the second. Regression "Healthy.Lifestyle ~ logistic(log(0.45) - 0.1 * (Age - 45) + 0.05 * Female + 0.01 * Health.Percentile + 0.5 * Exercise.Sessions - 0.1 * (Diet == 'Moderate') - 0.4 * (Diet == 'Heavy'))" Linear Regression "Weight ~ lm(150 - 15 * Female + 0.5 * Age - 0.1 * Health.Percentile - 0.2 * Exercise.Sessions + 5 * (Diet == 'Moderate') + 15 * (Diet == 'Heavy') - 2 * Healthy.Lifestyle + N(0, 10))". Note that the error term may be specified symbolically with any of the above distributions.
n	A numeric value for the number of observations in each experiment.
num.experiments	A numeric value representing the number of simulated experiments.
the.formula	A formula object or character value specifying the formula for the regression model.
conf.level	A numeric value between 0 and 1 representing the confidence level (1 - significance level).
the.quantiles	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
experiment.name	A character value providing the name for the column identifying the experiment.
step.split	A character value that separates the name of the variable to be simulated (left side) from its distribution (right side). Using the.steps = "X ~ N(0,1)" with step.split = "~" will generate a variable named X from a standard Normal distribution.
coef.name	A character value specifying the column of the.coefs that contains the names of the input variables of the linear regression model.
estimate.name	A character value specifying the column of the.coefs that contains the estimated coefficients of the linear regression model.

lm.p.name	A character value specifying the column of the.coefs that contains the p-values for the tests of the estimated coefficients of the linear regression model.
f.p.name	A character value specifying the column of summary.stats that contains the p-value for the linear regression model's F test.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

## Value

A list containing four elements: 'the.steps': The steps or processes defined for the simulation, detailing the structure and relationships among the variables. 'simdat': A data frame of the simulated data based on linear regression. 'statistics': A data frame containing regression statistics, coefficient estimates, and related results. 'sim.analysis': Summary statistics or analysis results from the simulated linear regression.

## Examples

```

step.age <- "Age ~ N(45, 10)"
step.female <- "Female ~ binary(0.53)"
step.health.percentile <- "Health.Percentile ~ U(0,100)"
step.exercise.sessions <- "Exercise.Sessions ~ Poisson(2)"
step.diet <- "Diet ~ sample(('Light', 'Moderate', 'Heavy'),
(0.2, 0.45, 0.35))"

step.healthy.lifestyle <- "Healthy.Lifestyle ~ logistic(log(0.45) - 0.1 *
(Age -45) + 0.05 * Female + 0.01 * Health.Percentile + 0.5 *
Exercise.Sessions - 0.1 * (Diet == 'Moderate') - 0.4 *
(Diet == 'Heavy'))"

step.weight <- "Weight ~ lm(150 - 15 * Female + 0.5 * Age - 0.1 *
Health.Percentile - 0.2 * Exercise.Sessions + 5 * (Diet == 'Moderate') +
15 * (Diet == 'Heavy') - 2 * Healthy.Lifestyle + N(0, 10))"

the.steps <- c(step.age, step.female, step.health.percentile,
step.exercise.sessions, step.diet, step.healthy.lifestyle,
step.weight)

study.lm <- simstudy.lm(the.steps = the.steps, n = 100,
num.experiments = 2,
the.formula = Weight ~ Age + Female + Health.Percentile +
Exercise.Sessions + Healthy.Lifestyle,
conf.level = 0.95, the.quantiles = c(0.25, 0.75),
experiment.name = "sim", seed = 11)

```



---

 simstudy.logistic      *simstudy.logistic*


---

## Description

simstudy.logistic

## Usage

```
simstudy.logistic(
  the.steps,
  n,
  num.experiments,
  the.formula,
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  experiment.name = "experiment",
  step.split = "~",
  coef.name = "Coefficient",
  estimate.name = "Estimate",
  logistic.p.name = "Pr(>|z|)",
  seed = 39,
  vstr = 3.6
)
```

## Arguments

the.steps	A character vector of variables to simulate. The variables are simulated in the order specified. Later variables can be generated to depend on earlier variables. The possible specifications include: Normal "X ~ N(100, 5)" with the mean and SD. Uniform "X ~ U(0, 100)" with the minimum and maximum. Poisson "X ~ Poisson(3)" with the mean. Binary "X ~ Binary(0.5)" with the probability of success. Binomial "X ~ Bin(10, 0.2)" with the number of trials and probability of success. Categorical "Diet ~ sample(('Light', 'Moderate', 'Heavy'), (0.2, 0.45, 0.35))" with the values in the first set of parentheses and their respective probabilities in the second. Logistic Regression "Healthy.Lifestyle ~ logistic(log(0.45) - 0.1 * (Age - 45) + 0.05 * Female + 0.01 * Health.Percentile + 0.5 * Exercise.Sessions - 0.1 * (Diet == 'Moderate') - 0.4 * (Diet == 'Heavy'))" Linear Regression "Weight ~ lm(150 - 15 * Female + 0.5 * Age - 0.1 * Health.Percentile - 0.2 * Exercise.Sessions + 5 * (Diet == 'Moderate') + 15 * (Diet == 'Heavy') - 2 * Healthy.Lifestyle + N(0, 10))". Note that the error term may be specified symbolically with any of the above distributions.
n	A numeric value for the number of observations in each experiment.
num.experiments	A numeric value representing the number of simulated experiments.
the.formula	A formula object or character value specifying the formula for the regression model.

<code>conf.level</code>	A numeric value between 0 and 1 representing the confidence level (1 - significance level).
<code>the.quantiles</code>	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
<code>experiment.name</code>	A character value providing the name for the column identifying the experiment.
<code>step.split</code>	A character value that separates the name of the variable to be simulated (left side) from its distribution (right side). Using <code>the.steps = "X ~ N(0,1)"</code> with <code>step.split = "~"</code> will generate a variable named X from a standard Normal distribution.
<code>coef.name</code>	A character value specifying the column of the <code>the.coefs</code> that contains the names of the input variables of the linear regression model.
<code>estimate.name</code>	A character value specifying the column of the <code>the.coefs</code> that contains the estimated coefficients of the linear regression model.
<code>logistic.p.name</code>	A character value specifying the column of the <code>the.coefs</code> that contains the p-values for the tests of the estimated coefficients of the logistic regression model.
<code>seed</code>	A single numeric value, interpreted as an integer, or NULL. See <code>help(set.seed)</code> .
<code>vstr</code>	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if <code>vstr</code> is greater than the current version. See <code>help(set.seed)</code> .

## Value

A list containing the following components:

- `the.steps`A character vector detailing the steps of the simulation.
- `simdat`A data frame of the simulated data based on the steps provided.
- `statistics`A data frame containing the statistical analysis of the simulated data.
- `sim.analysis`A data frame containing the analysis of the simulated study for the logistic regression.

## Examples

```
step.age <- "Age ~ N(45, 10)"
step.female <- "Female ~ binary(0.53)"
step.health.percentile <- "Health.Percentile ~ U(0,100)"
step.exercise.sessions <- "Exercise.Sessions ~ Poisson(2)"
step.diet <- "Diet ~ sample(('Light', 'Moderate', 'Heavy'),
(0.2, 0.45, 0.35))"

step.healthy.lifestyle <- "Healthy.Lifestyle ~ logistic(log(0.45) - 0.1 *
(Age -45) + 0.05 * Female + 0.01 * Health.Percentile + 0.5 *
Exercise.Sessions - 0.1 * (Diet == 'Moderate') - 0.4 *
(Diet == 'Heavy'))"
```

```

step.weight <- "Weight ~ lm(150 - 15 * Female + 0.5 * Age - 0.1 *
Health.Percentile - 0.2 * Exercise.Sessions + 5 *
(Diet == 'Moderate') + 15 * (Diet == 'Heavy') - 2 *
Healthy.Lifestyle + N(0, 10))"

the.steps <- c(step.age, step.female, step.health.percentile,
step.exercise.sessions, step.diet,
step.healthy.lifestyle, step.weight)

study.logistic <- simstudy.logistic(the.steps = the.steps,
n = 100, num.experiments = 2,
the.formula = Healthy.Lifestyle ~ Age + Female + Health.Percentile +
Exercise.Sessions, conf.level = 0.95,
the.quantiles = c(0.025, 0.1, 0.5, 0.9, 0.975),
experiment.name = "sim", seed = 222)

```

---

simstudy.prop

*simstudy.prop*


---

## Description

simstudy.prop

## Usage

```

simstudy.prop(
  n,
  p.actual = 0.5,
  p.hypothesized = 0.5,
  num.experiments = 1,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  correct = TRUE,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  experiment.name = "experiment",
  value.name = "x",
  seed = 7261,
  vstr = 3.6
)

```

## Arguments

**n** A numeric value for the number of observations in each experiment.

**p.actual** A numeric value for the actual probability of success.

**p.hypothesized** A numeric value for the hypothesized probability of success.

num.experiments	A numeric value representing the number of simulated experiments.
alternative	See help(prop.test).
conf.level	See help(prop.test).
correct	See help(prop.test).
the.quantiles	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
experiment.name	A character value providing the name for the column identifying the experiment.
value.name	A character value providing the name for the simulated values.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

### Value

A list containing the following components:

- `simdat.propA` data frame of the simulated proportions.
- `test.statistics.propA` data frame containing the results of the proportion tests.
- `sim.analysis.propA` data frame containing the analysis of the simulated study for proportions.

### Examples

```
study.prop <- simstudy.prop(n = 30, p.actual = 0.42,
  p.hypothesized = 0.5, num.experiments = 2, alternative = "less",
  conf.level = 0.92, correct = TRUE, the.quantiles = c(0.04, 0.5, 0.96),
  experiment.name = "simulation_id", value.name = "success", seed = 8001)
```

---

simstudy.prop2

*simstudy.prop2*

---

### Description

simstudy.prop2

**Usage**

```

simstudy.prop2(
  nx,
  ny,
  px,
  py,
  num.experiments,
  p = NULL,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  correct = TRUE,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  experiment.name = "experiment",
  group.name = "group",
  x.value = "x",
  y.value = "y",
  value.name = "value",
  seed = 920173,
  vstr = 3.6
)

```

**Arguments**

<code>nx</code>	A numeric value for the number of observations in the x group for each experiment.
<code>ny</code>	A numeric value for the number of observations in the y group for each experiment.
<code>px</code>	A numeric value for the probability of success in the x group.
<code>py</code>	A numeric value for the probability of success in the y group.
<code>num.experiments</code>	A numeric value representing the number of simulated experiments.
<code>p</code>	See <code>help(prop.test)</code> .
<code>alternative</code>	See <code>help(prop.test)</code> .
<code>conf.level</code>	See <code>help(prop.test)</code> .
<code>correct</code>	See <code>help(prop.test)</code> .
<code>the.quantiles</code>	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
<code>experiment.name</code>	A character value providing the name for the column identifying the experiment.
<code>group.name</code>	A character value providing the name of the column of the group labels.
<code>x.value</code>	A character value specifying the label used for data in the x group (in the column labeled by the <code>group.name</code> parameter).
<code>y.value</code>	A character value specifying the label used for data in the y group (in the column labeled by the <code>group.name</code> parameter).

value.name	A character value specifying the name of the column that contains the value of the simulated data.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

### Value

A list containing the following components:

- `simdat.prop2A` data frame of the simulated data for two proportions.
- `test.statistics.prop2A` data frame containing the results of the tests for the two proportions.
- `sim.analysis.prop2A` data frame containing the analysis of the simulated study for the two proportions.

### Examples

```
study.noeffect.prop2 <- simstudy.prop2(nx = 40,
  ny = 40, px = 0.4, py = 0.4, num.experiments = 2,
  alternative = "two.sided", conf.level = 0.95, seed = 71)
```

---

simstudy.t	<i>simstudy.t</i>
------------	-------------------

---

### Description

simstudy.t

### Usage

```
simstudy.t(
  n,
  mean = 0,
  sd = 1,
  num.experiments = 1,
  alternative = c("two.sided", "less", "greater"),
  mu = 0,
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  experiment.name = "experiment",
  value.name = "x",
  seed = 7261,
  vstr = 3.6
)
```

**Arguments**

<code>n</code>	A numeric value for the number of observations in each experiment.
<code>mean</code>	A numeric value for the expected value of the data to be simulated.
<code>sd</code>	A numeric value for the standard deviation of the data to be simulated.
<code>num.experiments</code>	A numeric value representing the number of simulated experiments.
<code>alternative</code>	See <code>help(t.test)</code> .
<code>mu</code>	See <code>help(t.test)</code>
<code>conf.level</code>	See <code>help(t.test)</code>
<code>the.quantiles</code>	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
<code>experiment.name</code>	A character value providing the name for the column identifying the experiment.
<code>value.name</code>	A character value providing the name for the simulated values.
<code>seed</code>	A single numeric value, interpreted as an integer, or NULL. See <code>help(set.seed)</code> .
<code>vstr</code>	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if <code>vstr</code> is greater than the current version. See <code>help(set.seed)</code> .

**Value**

A list containing the following components:

- `simdat.t` A data frame of the simulated t-distributed data.
- `test.statistics.t` A data frame containing the results of the t-tests.
- `sim.analysis.t` A data frame containing the analysis of the simulated study for t-tests.

**Examples**

```
study.t <- simstudy.t(n = 25, mean = 0.3, sd = 1,
  num.experiments = 2, alternative = "greater", mu = 0,
  conf.level = 0.95, the.quantiles = c(0.025, 0.975),
  experiment.name = "experiment", value.name = "x", seed = 817)
```

---

`simstudy.t2`

*simstudy.t2*

---

**Description**

`simstudy.t2`

**Usage**

```

simstudy.t2(
  nx,
  ny,
  meanx = 0,
  meany = 1,
  sdx = 1,
  sdy = 1,
  num.experiments = 1,
  alternative = c("two.sided", "less", "greater"),
  mu = 0,
  var.equal = FALSE,
  conf.level = 0.95,
  the.quantiles = c(0.025, 0.1, 0.25, 0.5, 0.75, 0.9, 0.975),
  experiment.name = "experiment",
  group.name = "group",
  x.value = "x",
  y.value = "y",
  value.name = "value",
  seed = 3471,
  vstr = 3.6
)

```

**Arguments**

<code>nx</code>	A numeric value for the number of observations in the x group for each experiment.
<code>ny</code>	A numeric value for the number of observations in the y group for each experiment.
<code>meanx</code>	A numeric value for the expected value of the x group used in the simulation.
<code>meany</code>	A numeric value for the expected value of the y group used in the simulation.
<code>sdx</code>	A numeric value for the standard deviation of the x group used in the simulation.
<code>sdy</code>	A numeric value for the standard deviation of the y group used in the simulation.
<code>num.experiments</code>	A numeric value representing the number of simulated experiments.
<code>alternative</code>	See <code>help(t.test)</code> .
<code>mu</code>	See <code>help(t.test)</code>
<code>var.equal</code>	A logical indicating whether to treat the two variances as being equal. If TRUE, then a pooled variance is used to estimate the variance, otherwise the variances are estimated separately. See <code>help(t.test)</code> .
<code>conf.level</code>	See <code>help(t.test)</code>
<code>the.quantiles</code>	A numeric vector of values between 0 and 1. Summary statistics to analyze the tests will return the specified quantiles.
<code>experiment.name</code>	A character value providing the name for the column identifying the experiment.



group.name	A character value providing the name of the column of the group labels.
x.value	A character value specifying the label used for data in the x group (in the column labled by the group.name parameter).
y.value	A character value specifying the label used for data in the y group (in the column labled by the group.name parameter).
value.name	A character value specifying the name of the column that contains the value of the simulated data.
seed	A single numeric value, interpreted as an integer, or NULL. See help(set.seed).
vstr	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if vstr is greater than the current version. See help(set.seed).

### Value

A list containing the following components:

- `simdat.t2A` data frame of the simulated t-distributed data for two groups.
- `test.statistics.t2A` data frame containing the results of the t-tests for two groups.
- `sim.analysis.t2A` data frame containing the analysis of the simulated study for t-tests for two groups.

### Examples

```
study.t2 <- simstudy.t2(nx = 30, ny = 40,
  meanx = 0, meany = 0.2, sdX = 1, sdY = 1, num.experiments = 2,
  alternative = "less", mu = 0, conf.level = 0.9,
  the.quantiles = c(0.1, 0.5, 0.9), experiment.name = "experiment_id",
  group.name = "category", x.value = "a",
  y.value = "b", value.name = "measurement", seed = 41)
```

---

simulation.steps      *simulation.steps*

---

### Description

simulation.steps

### Usage

```
simulation.steps(
  the.steps,
  n,
  num.experiments = 1,
  experiment.name = "experiment",
  step.split = "~",
  seed = 62,
  vstr = 3.6
)
```

**Arguments**

<code>the.steps</code>	A character vector of variables to simulate. The variables are simulated in the order specified. Later variables can be generated to depend on earlier variables. The possible specifications include:
<code>n</code>	A numeric value for the number of observations in each experiment.
<code>num.experiments</code>	A numeric value representing the number of simulated experiments.
<code>experiment.name</code>	A character value providing the name for the column identifying the experiment.
<code>step.split</code>	A character value that separates the name of the variable to be simulated (left side) from its distribution (right side). Using <code>the.steps = "X ~ N(0,1)"</code> with <code>step.split = "~"</code> will generate a variable named X from a standard Normal distribution.
<code>seed</code>	A single numeric value, interpreted as an integer, or NULL. See <code>help(set.seed)</code> .
<code>vstr</code>	A character string containing a version number, e.g., "1.6.2". The default RNG configuration of the current R version is used if <code>vstr</code> is greater than the current version. See <code>help(set.seed)</code> .

**Value**

A data table containing the simulated data based on the sequence of steps specified in `the.steps`. The data table also includes an additional column identifying the experiment as specified by the `experiment.name` parameter.

**Examples**

```
step.age <- "Age ~ N(45, 10)"
step.female <- "Female ~ binary(0.53)"
step.health.percentile <- "Health.Percentile ~ U(0,100)"
step.exercise.sessions <- "Exercise.Sessions ~ Poisson(2)"
step.diet <- "Diet ~ sample(('Light', 'Moderate', 'Heavy'),
(0.2, 0.45, 0.35))"

step.healthy.lifestyle <- "Healthy.Lifestyle ~
logistic(log(0.45) - 0.1 * (Age -45) + 0.05 * Female + 0.01 *
Health.Percentile + 0.5 * Exercise.Sessions - 0.1 *
(Diet == 'Moderate') - 0.4 * (Diet == 'Heavy'))"

step.weight <- "Weight ~ lm(150 - 15 * Female + 0.5 *
Age - 0.1 * Health.Percentile - 0.2 * Exercise.Sessions +
5 * (Diet == 'Moderate') + 15 * (Diet == 'Heavy') -
2 * Healthy.Lifestyle + N(0, 10))"

the.steps <- c(step.age, step.female, step.health.percentile,
step.exercise.sessions, step.diet, step.healthy.lifestyle, step.weight)

simdat.multivariate <- simulation.steps(the.steps = the.steps,
n = 50, num.experiments = 2, experiment.name = "sim", seed = 41)
```

# Index

analyze.simstudy.chisq.test.gf, 3  
analyze.simstudy.chisq.test.ind, 4  
analyze.simstudy.lm, 5  
analyze.simstudy.logistic, 6  
analyze.simstudy.prop, 8  
analyze.simstudy.prop2, 9  
analyze.simstudy.t, 10  
analyze.simstudy.t2, 12

buildsim.binary, 13  
buildsim.binomial, 14  
buildsim.lm, 14  
buildsim.logistic, 15  
buildsim.normal, 15  
buildsim.poisson, 16  
buildsim.sample, 17  
buildsim.uniform, 18

identify.distribution, 18  
internal.chisq.test.gf, 19  
internal.chisq.test.ind, 20  
internal.prop.test, 20  
internal.prop2.test, 21  
internal.quantiles.mean.sd, 22  
internal.statistics.one.lm, 22  
internal.statistics.one.logistic, 23  
internal.statistics.onelm, 23  
internal.t.test, 24  
internal.t2.test, 25

quantile.dt, 26

set.seed, 27  
sim.chisq.gf, 26  
sim.chisq.ind, 27  
sim.chisq.test.gf, 29  
sim.chisq.test.ind, 30  
sim.norm, 31  
sim.prop, 32  
sim.prop.test, 33  
sim.prop2, 34  
sim.prop2.test, 35  
sim.statistics.lm, 36  
sim.statistics.logistic, 37  
sim.t, 38  
sim.t.test, 39  
sim.t2, 41  
sim.t2.test, 42  
simstudy.chisq.test.gf, 43  
simstudy.chisq.test.ind, 45  
simstudy.lm, 46  
simstudy.logistic, 49  
simstudy.prop, 51  
simstudy.prop2, 52  
simstudy.t, 54  
simstudy.t2, 55  
simulation.steps, 57