Package 'sigr'

February 16, 2017

Type Package

Title Format Significance Summaries for Reports

Version 0.1.4

Date 2017-02-16

Author John Mount, Nina Zumel

Maintainer John Mount <jmount@win-vector.com>

URL https://github.com/WinVector/sigr

BugReports https://github.com/WinVector/sigr/issues

Description Succinctly format significance summaries of
various models and tests. The main purpose is unified reporting and planning
of experimental results, working around issue such as the difficulty of
extracting model summary facts (such as with 'lm'/'glm'). This package also
includes empirical tests, such as bootstrap estimates.

License GPL-3

LazyData TRUE

RoxygenNote 5.0.1

Suggests parallel, knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-16 17:57:13

R topics documented:

`calcAUC` *calculate AUC.*

Description

Based on: http://blog.revolutionanalytics.com/2016/08/roc-curves-in-two-lines-of-code.html

Usage

```
calcAUC(modelPredictions, yValues)
```

Arguments

modelPredictions	numeric predictions (not empty)
yValues	logical truth (not empty, same lenght as model predictions)

Value

area under curve

Examples

```
sigr::calcAUC(1:4,c(TRUE,FALSE,TRUE,TRUE)) # should be 2/3
```

---

calcDeviance *Calculate deviance.*

Description

Calculate deviance.

Usage

```
calcDeviance(pred, y)
```

Arguments

pred	numeric predictions
y	logical truth

Value

deviance

Examples

```
sigr::calcDeviance(1:4,c(TRUE,FALSE,TRUE,TRUE))
```

`calcSSE` *Calculate sum of squared error.*

Description

Calculate sum of squared error.

Usage

```
calcSSE(pred, y)
```

Arguments

`pred`	numeric predictions
`y`	numeric truth

Value

sum of squared error

Examples

```
sigr::calcSSE(1:4,c(TRUE,FALSE,TRUE,TRUE))
```

`estimateDifferenceZeroCrossing` *Studentized estimate of how often a difference is below zero.*

Description

Studentized estimate of how often a difference is below zero.

Usage

```
estimateDifferenceZeroCrossing(resampledDiffs)
```

Arguments

`resampledDiffs`	numeric vector resampled observations

Value

estimated probbility of seeing a re-sampled difference below zero.

Examples

```
set.seed(2352)
resampledDiffs <- rnorm(10)+1
estimateDifferenceZeroCrossing(resampledDiffs)
```

`getRenderingFormat` *Detect rendering format (using knitr).*

Description

Detect rendering format (using knitr).

Usage

```
getRenderingFormat()
```

Value

rendering format

Examples

```
getRenderingFormat()
```

`permTestAUC` *Perform AUC permutation test.*

Description

Estimate signficance of AUC by permuation test.

Usage

```
permTestAUC(d, modelName, yName, yTarget, ..., returnScores = FALSE,
  nrep = 100, parallelCluster = NULL)
```

Arguments

d	data.frame
modelName	character model column name
yName	character outcome column name
yTarget	target to match to y
...	extra arguments (not used)
returnScores	logical if TRUE return detailed permutedScores
nrep	number of permutation repititions to estimate p values.
parallelCluster	(optional) a cluster object created by package parallel or package snow

Value

AUC statistic

Examples

```
set.seed(25325)
d <- data.frame(x1=c(1,2,3,4,5,6,7,7),
                y=c(FALSE,TRUE,FALSE,FALSE,
                    TRUE,TRUE,FALSE,TRUE))
permTestAUC(d,'x1','y',TRUE)
```

---

permutationScoreModel *Empirical permuation test of significance of scoreFn(modelValues,yValues) >= scoreFn(modelValues,perm(yValues)).*

Description

Treat permutaiton re-samples as similar to bootstrap replications.

Usage

```
permutationScoreModel(modelValues, yValues, scoreFn, ...,
  returnScores = FALSE, nRep = 100, parallelCluster = NULL)
```

Arguments

modelValues	numeric array of predictions.
yValues	numeric/logical array of outcomes, depedendent, or truth values
scoreFn	function with signature scoreFn(modelValues,yValues) returning scalar numeric score.
...	not used, forces later arguments to be bound by name
returnScores	logical if TRUE return detailed permutedScores
nRep	integer number of repititions to perform
parallelCluster	optional snow-style parallel cluster.

Value

summaries

Examples

```
set.seed(25325)
y <- 1:5
m <- c(1,1,2,2,2)
cor.test(m,y,alternative='greater')
f <- function(modelValues,yValues) cor(modelValues,yValues)
permutationScoreModel(m,y,f)
```

---

print.sigr_statistic *Print*

Description

Print

Usage

```
## S3 method for class 'sigr_statistic'
print(x, ...)
```

Arguments

x	sigr wrapper to print
...	extra arguments

Value

formatted string

Examples

```
print(wrapSignificance(1/300))
```

`render` *Format summary roughly in "APA Style" (American Psychological Association).*

Description

Format summary roughly in "APA Style" (American Psychological Association).

Usage

```
render(statistic, ..., format, sigDigits = 2, pLargeCutoff = 0.05,
  pSmallCutoff = 1e-05)
```

Arguments

`statistic`	sigr summary statistic
`...`	extra arguments
`format`	if set the format to return ("html", "latex", "markdown", "ascii")
`sigDigits`	integer number of digits to show
`pLargeCutoff`	value to declare non-significance at or above.
`pSmallCutoff`	smallest value to print

Value

formatted string

See Also

`render.sigr_significance`, `render.sigr_ftest`

`render.sigr_aucpairtest` *Format an AUC-test (quality of a probability score)*

Description

Format an AUC-test (quality of a probability score)

Usage

```
## S3 method for class 'sigr_aucpairtest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

`statistic`	wrapped AUC test
`...`	not used, force use of named binding for later arguments
`format`	if set the format to return ("html", "latex", "markdown", "ascii")
`sigDigits`	integer number of digits to show
`pLargeCutoff`	value to declare non-significance at or above.
`pSmallCutoff`	smallest value to print

Value

formatted string

`render.sigr_aucpermtest` *Format an AUC-test (quality of a probability score)*

Description

Format an AUC-test (quality of a probability score)

Usage

```
## S3 method for class 'sigr_aucpermtest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped AUC test
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

---

render.sigr_aucresamptest

Format an AUC-test (quality of a probability score)

Description

Format an AUC-test (quality of a probability score)

Usage

```
## S3 method for class 'sigr_aucresamptest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped AUC test
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

render.sigr_chisqtest *Format a chi-square test (quality of categorical prediction)*

Description

Format a chi-square test (quality of categorical prediction)

Usage

```
## S3 method for class 'sigr_chisqtest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped T-test
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

render.sigr_cortest *Format cor.test (test of liner correlation).*

Description

Format cor.test (test of liner correlation).

Usage

```
## S3 method for class 'sigr_cortest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped cor.test.
...	extra arguments (not used)
format	if set the format to return ("html", "latex", "markdown", "ascii", "docx", ...)
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
ct <- cor.test(d$x,d$y)
wrapCorTest(ct)
```

---

render.sigr_emptest *Format an empirical test (quality of categorical prediction)*

Description

Format an empirical test (quality of categorical prediction)

Usage

```
## S3 method for class 'sigr_emptest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped T-test
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

`render.sigr_fishertest` *Format fisher.test (test of categorial indendence).*

Description

Format fisher.test (test of categorial indendence).

Usage

```
## S3 method for class 'sigr_fishertest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

`statistic`	wrapped Fisher test
`...`	extra arguments (not used)
`format`	if set the format to return ("html", "latex", "markdown", "ascii", "docx", ...)
`sigDigits`	integer number of digits to show
`pLargeCutoff`	value to declare non-significance at or above.
`pSmallCutoff`	smallest value to print

Value

formatted string and fields

Examples

```
d <- data.frame(x=c('b','a','a','a','b','b','b'),
                y=c('1','1','1','2','2','2','2'))
ft <- fisher.test(table(d))
wrapFisherTest(ft)
```

render.sigr_ftest *Format an F-test*

Description

Format an F-test

Usage

```
## S3 method for class 'sigr_ftest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped test
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

render.sigr_permtest *Format an empirical test (quality of categorical prediction)*

Description

Format an empirical test (quality of categorical prediction)

Usage

```
## S3 method for class 'sigr_permtest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped T-test
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

---

render.sigr_significance

Format a significance

Description

Format a significance

Usage

```
## S3 method for class 'sigr_significance'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped significance
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

Examples

```
cat(render(wrapSignificance(1/300),format='html'))
```

render.sigr_ttest *Format a T-test (difference in means by group)*

Description

Format a T-test (difference in means by group)

Usage

```
## S3 method for class 'sigr_ttest'
render(statistic, ..., format, sigDigits = 2,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

statistic	wrapped T-test
...	not used, force use of named binding for later arguments
format	if set the format to return ("html", "latex", "markdown", "ascii")
sigDigits	integer number of digits to show
pLargeCutoff	value to declare non-significance at or above.
pSmallCutoff	smallest value to print

Value

formatted string

resampleScoreModel *Studentized bootstrap variance estimate for scoreFn(yValues,modelValues).*

Description

Studentized bootstrap variance estimate for scoreFn(yValues,modelValues).

Usage

```
resampleScoreModel(modelValues, yValues, scoreFn, ..., returnScores = FALSE,
  nRep = 100, parallelCluster = NULL)
```

Arguments

modelValues	numeric array of predictions (model to test).
yValues	numeric/logical array of outcomes, depedendent, or truth values
scoreFn	function with signature scoreFn(modelValues,yValues) returning scalar numeric score.
...	not used, forces later arguments to be bound by name
returnScores	logical if TRUE return detailed resampledScores
nRep	integer number of repititions to perform
parallelCluster	optional snow-style parallel cluster.

Value

summaries

Examples

```
set.seed(25325)
y <- 1:5
m1 <- c(1,1,2,2,2)
cor.test(m1,y,alternative='greater')
f <- function(modelValues,yValues) {
 if((sd(modelValues)<=0)||(sd(yValues)<=0)) {
   return(0)
 }
 cor(modelValues,yValues)
}
s <- sigr::resampleScoreModel(m1,y,f)
print(s)
z <- (s$observedScore-0)/s$sd # should check size of z relative to bias!
pValue <- pt(z,df=length(y)-2,lower.tail=FALSE)
pValue
```

---

resampleScoreModelPair

Studentized bootstrap test of strength of scoreFn(yValues,model1Values) > scoreFn(yValues,model1Values).

Description

True confidence intervals are harder to get right (see "An Introduction to the Bootstrap", Bradely Efron, and Robert J. Tibshirani, Chapman & Hall/CRC, 1993.), but we will settle for simple p-value estimates.

Usage

```
resampleScoreModelPair(model1Values, model2Values, yValues, scoreFn, ...,
  returnScores = FALSE, nRep = 100, parallelCluster = NULL,
  sameSample = FALSE)
```

Arguments

model1Values	numeric array of predictions (model to test).
model2Values	numeric array of predictions (reference model).
yValues	numeric/logical array of outcomes, depedendent, or truth values
scoreFn	function with signature scoreFn(modelValues,yValues) returning scalar numeric score.
...	not used, forces later arguments to be bound by name.
returnScores	logical if TRUE return detailed resampledScores.
nRep	integer number of repititions to perform.
parallelCluster	optional snow-style parallel cluster.
sameSample	logical if TRUE use the same sample in computing both scores during bootstrap replication (else use independent samples).

Value

summaries

Examples

```
set.seed(25325)
y <- 1:5
m1 <- c(1,1,2,2,2)
m2 <- c(1,1,1,1,2)
cor(m1,y)
cor(m2,y)
f <- function(modelValues,yValues) {
  if((sd(modelValues)<=0)||(sd(yValues)<=0)) {
    return(0)
  }
  cor(modelValues,yValues)
}
resampleScoreModelPair(m1,m2,y,f)
```

resampleTestAUC *Wrap AUC resampling test results.*

Description

Estimate signficance of AUC by resampling test.

Usage

```
resampleTestAUC(d, modelName, yName, yTarget, ..., returnScores = FALSE,
  nrep = 100, parallelCluster = NULL)
```

Arguments

d	data.frame
modelName	character model column name
yName	character outcome column name
yTarget	target to match to y
...	extra arguments (not used)
returnScores	logical if TRUE return detailed resampledScores.
nrep	number of permutation repititions to estimate p values.
parallelCluster	(optional) a cluster object created by package parallel or package snow.

Value

AUC statistic

Examples

```
set.seed(25325)
d <- data.frame(x1=c(1,2,3,4,5,6,7,7),
                y=c(FALSE,TRUE,FALSE,FALSE,
                    TRUE,TRUE,FALSE,TRUE))
resampleTestAUC(d,'x1','y',TRUE)
```

sigr *sigr: Format Significance Summaries for Reports*

Description

Succinctly format significance summaries of various models and tests. The main purpose is unified reporting and planning of experimental results, working around issue such as the difficulty of extracting model summary facts (such as with 'lm'/'glm'). This package also includes empirical tests, such as bootstrap estimates.

Details

To learn more about replyr, please start with the vignette: `vignette('sigrFormatting','sigr')`

testAUCpair *Test AUC pair results.*

Description

Estimate signficance of difference in two AUCs by resampling.

Usage

```
testAUCpair(d, model1Name, model2Name, yName, yTarget, ...,
  returnScores = FALSE, nrep = 100, parallelCluster = NULL)
```

Arguments

d	data.frame
model1Name	character model 1 column name
model2Name	character model 2 column name
yName	character outcome column name
yTarget	target to match to y
...	extra arguments (not used)
returnScores	logical if TRUE return detailed resampledScores
nrep	number of re-sample repetition to estimate p value.
parallelCluster	(optional) a cluster object created by package parallel or package snow

Value

AUC pair test

Examples

```
set.seed(25325)
d <- data.frame(x1=c(1,2,3,4,5,6,7,7),
                x2=1,
                y=c(FALSE,TRUE,FALSE,FALSE,
                    TRUE,TRUE,FALSE,TRUE))
testAUCpair(d,'x1','x2','y',TRUE)
```

`wrapChiSqTest` *Wrap quality of a categorical prediction roughly in "APA Style" (American Psychological Association).*

Description

Wrap quality of a categorical prediction roughly in "APA Style" (American Psychological Association).

Usage

```
wrapChiSqTest(x, ...)
```

Arguments

`x`	numeric, data.frame or lm where to get model or data to score.
`...`	extra arguments

See Also

`wrapChiSqTestImpl`, `wrapChiSqTest.glm`, and `wrapChiSqTest.data.frame`

`wrapChiSqTest.data.frame` *Format ChiSqTest from data.*

Description

Format ChiSqTest from data.

Usage

```
## S3 method for class 'data.frame'
wrapChiSqTest(x, predictionColumnName, yColumnName,
  nParameters = 1, meany = mean(x[[yColumnName]]), ...)
```

Arguments

x	data frame containing columns to compare
predictionColumnName	character name of prediction column
yColumnName	character name of column containing dependent variable
nParameters	number of variables in model
meany	(optional) mean of y
...	extra arguments (not used)

Value

wrapped test

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
      y=c(TRUE,FALSE,FALSE,FALSE,TRUE,TRUE,TRUE,FALSE))
model <- glm(y~x,data=d,family=binomial)
summary(model)
d$pred <- predict(model,type='response',newdata=d)
render(wrapChiSqTest(d,'pred','y'),pLargeCutoff=1)
```

---

wrapChiSqTest.glm *Format ChiSqTest from model.*

Description

Format ChiSqTest from model.

Usage

```
## S3 method for class 'glm'
wrapChiSqTest(x, ...)
```

Arguments

x	glm logistic regression model
...	extra arguments (not used)

Value

wrapped test

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
      y=c(TRUE,FALSE,FALSE,FALSE,TRUE,TRUE,TRUE,FALSE))
model <- glm(y~x,data=d,family=binomial)
summary(model)
render(wrapChiSqTest(model),pLargeCutoff=1,format='ascii')
```

`wrapChiSqTestImpl` *Format quality of a logistic regression roughly in "APA Style" (American Psychological Association).*

Description

Format quality of a logistic regression roughly in "APA Style" (American Psychological Association).

Usage

```
wrapChiSqTestImpl(df.null, df.residual, null.deviance, deviance)
```

Arguments

`df.null`	null degrees of freedom.
`df.residual`	residual degrees of freedom.
`null.deviance`	null deviance
`deviance`	residual deviance

Value

wrapped statistic

Examples

```
wrapChiSqTestImpl(df.null=7,df.residual=6,
    null.deviance=11.09035,deviance=10.83726)
```

`wrapCorTest` *Wrap cor.test (test of liner correlation).*

Description

Wrap cor.test (test of liner correlation).

Usage

```
wrapCorTest(x, ...)
```

Arguments

`x`	numeric, data.frame or test.
`...`	extra arguments

See Also

`wrapCorTest.htest`, and `wrapCorTest.data.frame`

`wrapCorTest.data.frame` *Wrap cor.test (test of liner correlation).*

Description

Wrap cor.test (test of liner correlation).

Usage

```
## S3 method for class 'data.frame'
wrapCorTest(x, Column1Name, Column2Name, ...)
```

Arguments

`x`	data.frame
`Column1Name`	character column 1 name
`Column2Name`	character column 2 name
`...`	extra arguments

Value

wrapped stat

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
wrapCorTest(d,'x','y')
```

`wrapCorTest.htest` *Wrap cor.test (test of liner correlation).*

Description

Wrap cor.test (test of liner correlation).

Usage

```
## S3 method for class 'htest'
wrapCorTest(x, ...)
```

Arguments

`x`	cor.test result
`...`	extra arguments (not used)

Value

wrapped stat

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
ct <- cor.test(d$x,d$y)
wrapCorTest(ct)
```

`wrapFisherTest` *Wrap fisher.test (test of categorial indendence).*

Description

Wrap fisher.test (test of categorial indendence).

Usage

```
wrapFisherTest(x, ...)
```

Arguments

`x`	numeric, data.frame or test.
`...`	extra arguments

See Also

`wrapFisherTest.htest`, and `wrapFisherTest.data.frame`

`wrapFisherTest.data.frame`
Wrap fisher.test (test of categorial indendence).

Description

Wrap fisher.test (test of categorial indendence).

Usage

```
## S3 method for class 'data.frame'
wrapFisherTest(x, Column1Name, Column2Name, ...)
```

Arguments

`x`	data.frame
`Column1Name`	character column 1 name
`Column2Name`	character column 2 name
`...`	extra arguments

Value

wrapped test.

Examples

```
d <- data.frame(x=c('b','a','a','a','b','b','b'),
                y=c('1','1','1','2','2','2','2'))
wrapFisherTest(d,'x','y')
```

`wrapFisherTest.htest` *Wrap fisher.test (test of categorial indendence).*

Description

Wrap fisher.test (test of categorial indendence).

Usage

```
## S3 method for class 'htest'
wrapFisherTest(x, ...)
```

Arguments

`x`	fisher.test result
`...`	extra arguments (not used)

Value

wrapped test.

Examples

```
d <- data.frame(x=c('b','a','a','a','b','b','b'),
                y=c('1','1','1','2','2','2','2'))
ft <- fisher.test(table(d))
wrapFisherTest(ft)
```

`wrapFTest` *Wrap F-test (significane of a linear relation).*

Description

Wrap F-test (significane of a linear relation).

Usage

```
wrapFTest(x, ...)
```

Arguments

`x`	numeric, data.frame or lm where to get model or data to score.
`...`	extra arguments

See Also

`wrapFTestImpl`, `wrapFTest.lm`, and `wrapFTest.data.frame`

`wrapFTest.data.frame` *Wrap quality statistic of a linear relation from data.*

Description

Wrap quality statistic of a linear relation from data.

Usage

```
## S3 method for class 'data.frame'
wrapFTest(x, predictionColumnName, yColumnName,
  nParameters = 1, meany = mean(x[[yColumnName]]), ..., format,
  pLargeCutoff = 0.05, pSmallCutoff = 1e-05)
```

Arguments

`x`	data frame containing columns to compare
`predictionColumnName`	character name of prediction column
`yColumnName`	character name of column containing dependent variable
`nParameters`	number of variables in model
`meany`	(optional) mean of y
`...`	extra arguments (not used)
`format`	if set the format to return ("html", "latex", "markdown", "ascii", "docx")
`pLargeCutoff`	value to declare non-significance at or above.
`pSmallCutoff`	smallest value to print

Value

formatted string and fields

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
model <- lm(y~x,data=d)
summary(model)
d$pred <- predict(model,newdata=d)
sigr::wrapFTest(d,'pred','y')
```

`wrapFTest.lm` *Wrap quality statistic of a linear regression.*

Description

Wrap quality statistic of a linear regression.

Usage

```
## S3 method for class 'lm'
wrapFTest(x, ..., format, pLargeCutoff = 0.05,
  pSmallCutoff = 1e-05)
```

Arguments

`x`	lm model
`...`	extra arguments (not used)
`format`	if set the format to return ("html", "latex", "markdown", "ascii", "docx", ...)
`pLargeCutoff`	value to declare non-significance at or above.
`pSmallCutoff`	smallest value to print

Value

formatted string

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
model <- lm(y~x,data=d)
summary(model)
sigr::wrapFTest(model)
```

wrapFTestImpl *Wrap F-test (significane of a linear relation).*

Description

Wrap F-test (significane of a linear relation).

Usage

```
wrapFTestImpl(numdf, dendf, FValue)
```

Arguments

numdf	degrees of freedom 1.
dendf	degrees of freedom 2.
FValue	observed F test statistic

Value

wrapped statistic

Examples

```
wrapFTestImpl(numdf=2,dendf=55,FValue=5.56)
```

wrapSignificance *Wrap a significance*

Description

Wrap a significance

Usage

```
wrapSignificance(significance, symbol = "p")
```

Arguments

significance	numeric the significance value.
symbol	the name of the value (e.g. "p", "t", ...).

Value

wrapped significance

Examples

```
wrapSignificance(1/300)
```

`wrapTTest` *Wrap t.test (difference in means by group).*

Description

Wrap t.test (difference in means by group).

Usage

```
wrapTTest(x, ...)
```

Arguments

`x`	numeric, data.frame or test.
`...`	extra arguments

See Also

`wrapTTest.htest`, and `wrapTTest.data.frame`

`wrapTTest.data.frame` *Wrap t.test (difference in means by group).*

Description

Wrap t.test (difference in means by group).

Usage

```
## S3 method for class 'data.frame'
wrapTTest(x, Column1Name, Column2Name, ...)
```

Arguments

`x`	data.frame
`Column1Name`	character column 1 name
`Column2Name`	character column 2 name
`...`	extra arguments

Value

formatted string and fields

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
render(wrapTTest(d,'x','y'),pLargeCutoff=1)
# confirm p not order depedent
render(wrapTTest(d,'y','x'),pLargeCutoff=1)
```

`wrapTTest.htest` *Wrap t.test (difference in means by group).*

Description

Wrap t.test (difference in means by group).

Usage

```
## S3 method for class 'htest'
wrapTTest(x, ...)
```

Arguments

`x`	t.test result
`...`	extra arguments (not used)

Value

formatted string and fields

Examples

```
d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
tt <- t.test(d$x,d$y)
render(wrapTTest(tt),pLargeCutoff=1)
# confirm not rescaling, as a correlation test would
render(wrapTTest(t.test(d$x,2*d$y)),pLargeCutoff=1)
```

Index