

Package: metricsUW (via r-universe)

September 22, 2026

Version 0.2-0

Date 2026-09-17

Title Tools for Teaching Econometrics

Author Pierre Chausse [aut, cre]

Maintainer Pierre Chausse <pchausse@uwaterloo.ca>

Description Functions used in my econometrics courses to generate solutions, print results and generate data.

Depends R (>= 4.0.0)

Imports stats, momentfit, texreg, methods, xtable

Suggests knitr, AER, sandwich, wooldridge, stargazer

URL <https://codeberg.org/pchausse/metricsUW>,
<https://pchausse.r-universe.dev/metricsUW>

BugReports <https://codeberg.org/pchausse/metricsUW/issues>

License GPL (>= 2)

NeedsCompilation no

VignetteBuilder knitr

Config/pak/sysreqs texlive libssl-dev

Repository <https://pchausse.r-universe.dev>

Date/Publication 2026-09-22 14:12:55 UTC

RemoteUrl <https://codeberg.org/pchausse/metricsUW>

RemoteRef HEAD

RemoteSha 83d5580f650072a44ca3765184ed3df88e42d39c

Contents

distTable	2
genDemSup	3
genTS	4

inference	5
inferenceReg	6
print	8
printReg	9
probFromTable	10
Index	12

distTable	<i>Distribution Table</i>
-----------	---------------------------

Description

The function create a distribution table to be printed in Latex format.q

Usage

```
distTable(df=NULL, df2=NULL, digits=4, dist=c("t", "chisq", "f", "normal"),
          size=0.05, CDF=FALSE, float=TRUE, label=NULL)
```

Arguments

- df A vector of degrees of freedom.
- df2 A second vector of degrees of freedom for the f-distribution.
- digits How many digits to print?
- dist The distribution table to be printed.
- size The size for F-tests.
- CDF Should the normal table show CDF values? By default, it shows the probability of being between 0 and z.
- float Should the Latex table be a floating table?
- label The label of referencing the table in Latex. By default, it is dist followed by "Table".

Value

It prints a latex table.

Examples

```
## Not run: distTable()
```

genDemSup	<i>Demand and supply generator</i>
-----------	------------------------------------

Description

Functions to simulate data from a supply and demand system with one exogenous variable in each function, plot the system and print the equations.

Usage

```
genDemSup(n, a=c(20, 1, -2), d=c(100,-1,2,0), sige=16,
          sigeta=16, sigX=c(5,5), sigZ=5)

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

## S3 method for class 'DemSup'
plot(x, y=NULL, nCurves=NULL, nPoints=NULL, fac=c(0.9,1.1), ...)
```

Arguments

x	An object of class DemSup.
n	The sample size.
a	The coefficient of the supply function. The first is the intercept, the second is the slope of the supply and the third is the coefficient of the exogenous variable Z.
d	The coefficient of the demand function. The first is the intercept, the second is the slope of the demand and the third is the coefficient of the exogenous variable X.
sige, sigeta, sigX, sigZ	The variance of the error in the supply, the error in the demand, the exogenous variable X and the exogenous variable Z.
nCurves	How many supply and demand curves to plot. By default, all curves are plotted.
nPoints	How many data points to plot. By default, they are all plotted.
fac	It determine the range for each demand and supply function. The function goes from fac[1] times the observed price to fac[2] times the observed price.
y	Not used at the moment.
...	Arguments to pass to other methods. Currently not used.

Value

genDemSup returns an object of class DemSup. The object is a list that contains the following:

dat	A data.frame with the quantity Q, the price P, the exogenous variable from the demand X, the exogenous variable from the supply Z, the error from the demand eta and the error from the supply e.
-----	---

a	The parameters of the supply.
d	The parameters of the demand.
sig	The variance of the error terms.
sigExo	The variance of the exogenous variables.

Examples

```
set.seed(112233)
sys <- genDemSup(200)
print(sys)
plot(sys, nCurves=10)
```

genTS	<i>Functions to generate datasets</i>
-------	---------------------------------------

Description

The function generate datasets with known properties. Students can try different techniques to identify the processes.

Usage

```
genTS(n=200, frequency=12, start=c(1960,1), seed)
```

Arguments

n	The sample size
frequency, start	The frequency and starting data. See ts .
seed	An optional seed for the random generators

Value

It returns an `mts` object with two time series, X and Y . One of them is a unit root process and the other is a trend stationary series. The true processes are saved in the attribute "which".

Examples

```
dat <- genTS()
```

inference

*Object generators for inference solution***Description**

The following are all solution generators for different types of inference questions. They return an object that can be converted into Latex through its print method.

Usage

```
testMean(x, h0, size=.05, alter=c("diff","greater","less"),
        assume=c("Normal","nonNormal"), digits=4, dround=NULL,
        xbar=NULL, se=NULL, n=NULL)

ciMean(x, size=.05, assume=c("Normal","nonNormal"), digits=4, dround=NULL,
        xbar=NULL, se=NULL, n=NULL)

testDiffMeans(x1, x2, h0, size=.05, alter=c("diff","greater","less"),
              assume=c("Normal","nonNormal"), assumev=c("same", "diff"),
              digits=4, dround=NULL, xbar=NULL, se=NULL, n=NULL)

ciDiffMeans(x1, x2, size=.05, assume=c("Normal","nonNormal"),
            assumev=c("same", "diff"),
            digits=4, dround=NULL, xbar=NULL, se=NULL, n=NULL)

testVar(x, h0, size=.05, alter=c("diff","greater","less"),
        assume=c("Normal","nonNormal"), digits=4, dround=NULL,
        se=NULL, n=NULL)

ciVar(x, size=.05, assume=c("Normal","nonNormal"), digits=4, dround=NULL,
        se=NULL, n=NULL)

testDiffVar(x1, x2, size=.05,
            assume=c("Normal","nonNormal"), digits=4, dround=NULL,
            se=NULL, n=NULL)
```

Arguments

x, x1, x2	Numeric vectors
h0	The null hypothesis.
size	The size of the test or one minus the confidence level when confidence intervals are computed.
alter	The alternative hypothesis.
assume	What is assumed about the distribution of the data. It affects how the critical values are obtained. The t or F distribution is used under normality, and the asymptotic N(0,1) or chi-square is used otherwise.

assumev	Do we assume that the variance of the two samples are different or that they are same?
digits	How many digits to print?
dround	If we want to round the numbers before doing inference, dround is the number of decimals. This is particularly useful to generate solutions for exams in which coefficients, means and standard errors are rounded.
xbar, se, n	Alternatively, the sample mean(s), standard deviation(s) and sample size(s) can be provided, instead of numeric vector(s).

Value

It returns an object of class `metricsSol`.

Examples

```
## Test on the mean from the data

set.seed(112233)
x1 <- rnorm(30)
obj <- testMean(x1, 0)

## Test on the mean without data

obj <- testMean(h0=0, xbar=0.1, se=0.6, n=35)

## See vignette for more examples
```

inferenceReg

Object generators for inference solution in regression models

Description

The following are all solution generators for different types of inference questions based on linear regression. They return an object that can be converted into Latex though its print method.

Usage

```
testRegCoef(object, h0, which, size=.05, alter=c("diff","greater","less"),
            digits=4, assume=c("Normal","nonNormal"),
            beta=NULL, se = NULL, n=NULL, ncoef=NULL, dround=NULL)

ciRegCoef(object, size=0.05, which, digits=4,
            assume=c("Normal", "nonNormal"),
            beta=NULL, se=NULL, n=NULL, ncoef=NULL, dround=NULL)

testLinRegCoefs(object, R, q, size=.05, alter=c("diff","greater","less"),
```

```

      digits=4, assume=c("Normal","nonNormal"),
      beta=NULL, vcoef=NULL, n=NULL, dround=NULL)

testRegF(object, objectr=NULL, size=.05, digits=4,
      assume = c("Normal", "nonNormal"), dround=NULL,
      Rsq=NULL, SSRu=NULL, SSRr=NULL, nrest=NULL, n=NULL, ncoef=NULL)

```

Arguments

<code>object, objectr</code>	Object of class <code>lm</code> . For F-tests, <code>objectr</code> is the <code>lm</code> object of the restricted model.
<code>h0</code>	The null hypothesis.
<code>which</code>	An integer specifying which coefficient is being tested.
<code>size</code>	The size of the test or one minus the confidence level when confidence intervals are computed.
<code>alter</code>	The alternative hypothesis.
<code>assume</code>	What is assumed about the distribution of the data. It affects how the critical values are obtained. The t or F distribution is used under normality, and the asymptotic $N(0,1)$ or chi-square is used otherwise.
<code>digits</code>	How many digits to print?
<code>dround</code>	If we want to round the numbers before doing inference, <code>dround</code> is the number of decimals. This is particularly useful to generate solutions for exams in which coefficients, means and standard errors are rounded.
<code>beta, se, n, ncoef</code>	Alternatively, the coefficient estimate(s), the standard error(s), the sample size(s) and the number of coefficients in the regression model can be provided, instead of the <code>lm</code> object. The argument <code>se</code> can also be used to pass a vector of robust standard errors. By default, the non-robust one is used.
<code>R, q</code>	For test of linear combinations of coefficients, <code>R</code> is the hypothesis matrix with one row and <code>q</code> is the right-hand side.
<code>vcoef</code>	The alternative estimated covariance matrix. It is needed for robust testings
<code>Rsq</code>	The R-squared of the unrestricted model. If provided, an F-test for the significance of all coefficients but the intercept is performed.
<code>SSRu, SSRr, nrest</code>	The unrestricted SSR, the restricted SSR and the number of restrictions being tested. These values are needed if not <code>lm</code> objects are provided.

Value

It returns an object of class `metricsSol`.

Examples

```

## Test on one coefficient

data(wage1, package="wooldridge")
fit <- lm(wage~educ+exper, wage1)

```

```
testRegCoef(fit, h0=0.5, which=2, size=0.10, alter="greater")

## See vignette for more examples
```

print

Print method for objects of class metricsSol

Description

It prints the solution from the object using Latex format. It is mainly used inside markdown documents to generate nicely formatted solution.

Usage

```
## S3 method for class 'metricsSol'
print(x, addMess=NULL, ...)
```

Arguments

x	An object of class metricsSol.
addMess	Message to print at the end of the solution. I use it to add the mark distribution when I generate solutions for exams.
...	Arguments to pass to other methods. Currently not used.

Value

It print the type of inference and the detailed solution in Latex format.

Examples

```
set.seed(112233)
x1 <- rnorm(30)
obj <- testMean(x1, 0)
print(obj, addMess="Not point if no justification")
```

printReg

*Printing Estimated Models***Description**

The function generates a Latex equation from a model estimated by [lm](#) or [glm](#).

Usage

```
## S3 method for class 'lm'
printReg(obj, digits=4, format="f", maxpl=5,
         adjrsq=FALSE, se=NULL, stars=FALSE, dist=c("t","n"), label=NULL,
         omit=NULL, ...)

## S3 method for class 'tsls'
printReg(obj,digits=4, format='f', maxpl=5,
         se=NULL, stars=FALSE, label=NULL, omit=NULL,
         strength=FALSE, Jtest=FALSE, Exo=FALSE, ...)

## S3 method for class 'formula'
printReg(obj, maxpl=5, label=NULL, coef="\\beta",
         error="\\epsilon", ...)
```

Arguments

<code>obj</code>	An object used to print the regression equation.
<code>digits</code>	The number of digits to be printed (See formatC).
<code>format</code>	Printing format for coefficient estimates and their standard errors.(See formatC)
<code>maxpl</code>	The maximum number of variables to be printed per line.
<code>adjrsq</code>	Should we print the adjusted R-square?
<code>se</code>	Alternative vector of standard errors. By default, the one from vcov is used.
<code>stars</code>	Should we add stars next to significant coefficients?
<code>dist</code>	Distribution used for p-values. By default, the t-distribution is used. Alternatively, we can replace it by the asymptotic N(0,1).
<code>label</code>	Label for latex equations in case a reference is needed. By default, it is null and the equation is not numbered.
<code>omit</code>	This argument is to specify variables to omit from the printed equation. It is a character vector, and the function omits any variables that contain the characters detected by grep . When specified, the equation ends by specifying how many variables are omitted.
<code>strength, Jtest, Exo</code>	Should the first stage F-test, J-test and the list of exogenous variables be printed?
<code>coef, error</code>	What Greek letters to use for coefficients and error term? Note that two backslashes is needed for Greek letter. It is also possible to use normal letter and omit the backslashes.
<code>...</code>	Other argument to pass to format

Value

It returns an equation in Latex format.

Examples

```
## Estimated linear model

set.seed(112233)
x1 <- rnorm(30)
x2 <- rnorm(30)
y <- x1+x2+rnorm(30)
fit <- lm(y~x1+x2)
printReg(fit)

## formula to equation

printReg(y~x1*x2+z+I(z^2))
```

probFromTable	<i>Probability from normal tables</i>
---------------	---------------------------------------

Description

The function generates a Latex equation with the detailed steps to calculate probability of areas using a printed normal table.

Usage

```
probFromTable(a, b=NULL, type=c("less","greater"), digits=3,
              mu=0, sig2=1, varName="X", approx=FALSE, print=TRUE,
              CDF=FALSE)
```

Arguments

a	One of the boundary of the interval. See details.
b	One of the boundary of the interval. See details
digits	The number of digits shown in the normal table
type	When b is NULL, do we want the probability of being less or greater than a
mu	The expected value of the normal distribution for the unscaled variable.
sig2	The variance of the normal distribution for the unscaled variable.
varName	The name of the variable to be printed in the solution.
approx	Is the normal distribution an approximation?
print	Should the table be printed? If FALSE, the probabilities are returned.
CDF	Is the normal table showing the CDF? The alternative is a table that shows the probability of being between 0 and z.

Details

When *b* is NULL, it returns the probability that *X* is less than *a* when *type*="less" and the probability of being greater than *a* when *type*="greater". When *b* is not NULL, it returns the probability of being between *a* and *b*.

Value

It returns an equation in Latex format or a numeric probability.

Examples

```
## Not run: probFromTable(-10, 8, mu=6, sig2=25, digits=3)
```

Index

`ciDiffMeans (inference)`, [5](#)
`ciMean (inference)`, [5](#)
`ciRegCoef (inferenceReg)`, [6](#)
`ciVar (inference)`, [5](#)

`distTable`, [2](#)

`format`, [9](#)
`formatC`, [9](#)

`genDemSup`, [3](#)
`genTS`, [4](#)
`glm`, [9](#)
`grep`, [9](#)

`inference`, [5](#)
`inferenceReg`, [6](#)

`lm`, [7](#), [9](#)

`plot.DemSup (genDemSup)`, [3](#)
`print`, [8](#)
`print.DemSup (genDemSup)`, [3](#)
`printReg`, [9](#)
`probFromTable`, [10](#)

`testDiffMeans (inference)`, [5](#)
`testDiffVar (inference)`, [5](#)
`testLinRegCoefs (inferenceReg)`, [6](#)
`testMean (inference)`, [5](#)
`testRegCoef (inferenceReg)`, [6](#)
`testRegF (inferenceReg)`, [6](#)
`testVar (inference)`, [5](#)
`ts`, [4](#)

`vcov`, [9](#)