Package ‘StatEngine’

June 22, 2020

Type Package

Title An R Package for the UofSC Course STAT 509 ``Statistics for Engineers''

Version 1.0.0

Author Dewei Wang

Maintainer Dewei Wang <deweiwang@stat.sc.edu>

Description This R package provides tools for methods covered in the course STAT 509 ``Statistics for Engineers'' at the University of South Carolina

License MIT

Encoding UTF-8

LazyData true

R topics documented:

`Continuous Distributions` *Continuous distributions*

Description

Plot pdf/cdf, calculate mean/variance/standard deviation, and compute probabilities for various continuous distributions

Usage

```
# continuous Uniform Distribution
uniform.summary(a,b,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
uniform.prob(a,b,lb,ub)
uniform.quantile(a,b,q)

# Normal distribution
normal.summary(mu,sigma,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
normal.prob(mu,sigma,lb,ub)
normal.quantile(mu,sigma,q)

# Exponential distribution
exponential.summary(lambda,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
exponential.prob(lambda,lb,ub)
exponential.quantile(lambda,q)

# Gamma distribution
gamma.summary(r,lambda,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
gamma.prob(r,lambda,lb,ub)
gamma.quantile(r,lambda,p)

# Weibull distribution
weibull.summary(beta,delta,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
weibull.prob(beta,delta,lb,ub)
weibull.quantile(beta,delta,p)

# Lognormal distribution
lognormal.summary(theta,omega,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
lognormal.prob(theta,omega,lb,ub)
lognormal.quantile(theta,omega,p)

#Beta distribution
beta.summary(alpha,beta,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
beta.prob(alpha,beta,lb,ub)
beta.quantile(alpha,beta,p)
```

Arguments

`plotpdf`	TRUE or FALSE, if TRUE, it plots the pmf
`plotcdf`	TRUE or FALSE, if TRUE, it polts the cdf
`lb,ub`	lower bound (lb) and upper bound (ub) in a probability statement; lb could be -Inf; ub could be Inf; ub cannot be less than lb
`a,b`	lower bound and upper bound of the support of a continuous uniform distribution
`mu,sigma`	mean and standard deviation of a normal distribution
`lambda`	parameter of an exponential distribution
`r,lambda`	shape and scale parameters of a gamma distribution
`beta,delta`	shape and scale parameters of a Weibull distribution
`theta,omega`	shape and scale parameters of a lognormal distribution
`alpba,beta`	parameters of a beta distribution

Details

Plot the probability density function (pdf) and the cumulative distribution function (cdf) and calculate probability, mean, variable and standard deviation, and compute probabilities of various discrete distributions (continuous uniform distribution, normal distribution, exponential distribution, gamma distribution, Weibull distribution, lognormal distribution, beta distribution).

Value

`uniform.summary` a list of the mean, variance, and standard deviation of a continuous uniform distribution, plot of the pdf/cdf or not

`uniform.prob` probability of X between lb and ub based on a continuous uniform distribution

`uniform.quantile` quantile of a continuous uniform distribution

`normal.summary` a list of the mean, variance, and standard deviation of a normal distribution, plot of the pdf/cdf or not

`normal.prob` probability of X between lb and ub based on a normal distribution

`normal.quantile` quantile of a normal distribution

`exponential.summary` a list of the mean, variance, and standard deviation of an exponential distribution, plot of the pdf/cdf or not

`exponential.prob` probability of X between lb and ub based on an exponential distribution

`exponential.quantile` quantile of an exponential distribution

`gamma.summary` a list of the mean, variance, and standard deviation of a gamma distribution, plot of the pdf/cdf or not

`gamma.prob` probability of X between lb and ub based on a gamma distribution

`gamma.quantile` quantile of a gamma distribution

`weibull.summary` a list of the mean, variance, and standard deviation of a Weibull distribution, plot of the pdf/cdf or not

`weibull.prob` probability of X between lb and ub based on a Weibull distribution

`weibull.quantile` quantile of a Weibull distribution

`lognormal.summary` a list of the mean, variance, and standard deviation of a lognormal distribution, plot of the pdf/cdf or not

`lognormal.prob` probability of X between lb and ub based on a lognormal distribution

`lognormal.quantile` quantile of a lognormal distribution

`beta.summary` a list of the mean, variance, and standard deviation of a beta distribution, plot of the pdf/cdf or not

`beta.prob` probability of X between lb and ub based on a beta distribution

`beta.quantile` quantile of a beta distribution

Note

deweiwang@stat.sc.edu

Author(s)

Dewei Wang

References

Chapter 4 of the textbook "Applied Statistics and Probability for Engineers" 7th edition

Examples

```
# Continuous uniform distribution
a=4.9;b=5.1;uniform.summary(a,b)
uniform.prob(a,b,4.95,5) # P(4.95<X<5)
uniform.quantile(a,b,0.9) # x such that P(X>x)=0.1

#Normal distribution
normal.summary(10,2) #mu=10,sigma=2,variance=4
normal.prob(10,2,9,11)
normal.quantile(10,2,0.98)

#Exponential distribution
exponential.summary(25) #lambda=25
exponential.prob(25,0.1,Inf) #P(X>0.1)
exponential.quantile(25,0.1) # x such that P(X>x)=0.9

#Gamma distribution
gamma.summary(10,.5) #r=10,lambda=0.5
gamma.prob(10,0.5,25,Inf) #P(X>25)
gamma.quantile(10,0.5,0.95) # x such that P(X<x)=0.95

#Weibull distribution
weibull.summary(2,5000) #beta=2,delta=5000
weibull.prob(2,5000,6000,Inf) #P(X>6000)
weibull.quantile(2,5000,1-0.05) # x such that P(X>x)=0.05

#Lognormal distribution
lognormal.summary(10,1.5) #theta=10,omega=1.5
lognormal.prob(10,1.5,10000,Inf) #P(X>10000)
lognormal.quantile(10,1.5,1-0.99) # x such that P(X>x)=0.99

#Beta distribution
beta.summary(2.5,1) #alpha=2.5, beta=1
beta.prob(2.5,1,0.7,Inf) # P(X>0.7)
beta.quantile(2.5,1,0.99) # x such that P(X<x)=0.99
```

`Counting Techniques` *Sample Space Characterization*

Description

Use permutations and combinations to count outcomes

Usage

```
nPr(n,r): permute r items from a set of n distinct items

nCr(n,r): select r items from a set of n items where order does not matter

SimPerm(n_vec): permute n items when they are not totally distinct
```

Arguments

`n`	total n itmes
`r`	select r
`n_vec`	a vector of (n_1, n_2,...,n_r)

Details

Permutation, Combination, and Permutation of similar items

Value

nPr(n,r) provides the number of different ways to permute r items from a set of n distinct items.

nCr(n,r) provides the number of different ways to select r items from a set of n items where order does not matter.

SimPerm(n_vec) provides the number of different ways to permute n items when they are not totally distinct.

Note

deweiwang@stat.sc.edu

Author(s)

Dewei Wang

References

Section 2.2 of the textbook "Applied Statistics and Probability for Engineers" 7th edition

Examples

```
nPr(8,4)
nCr(3,2)
nCr(47,4)
SimPerm(c(2,3,2))
```

`Descriptive Statistics` *Descriptive Statistics*

Description

Calculate and plot descriptive statistics

Usage

```
data.summary(x)
```

Arguments

`x`	the sample
`plot`	True of False; plot or not the stem and leaf diagram, scatterplot, histogram, boxplot, QQ-plot

Details

Calculate various numerical descpritive statistics and plot the stem and leaf diagram, scatterplot, histogram, boxplot, QQ-plot

Value

`data.summary`	a table of numerical descpritive statistics (std stands for standard deviation), a stem and leaf diagram, and a figure consisting of scatterplot, histogram, boxplot, QQ-plot of the sample

Note

deweiwang@stat.sc.edu

Author(s)

Dewei Wang

References

Chapter 6 of the textbook "Applied Statistics and Probability for Engineers" 7th edition

Examples

```
x=scan("https://raw.githubusercontent.com/Harrindy/StatEngine/master/Data/CompressiveStrength.csv")
data.summary(x)
```

Discrete Distributions *Discrete distributions*

Description

Plot pmf/cdf, calculate mean/variance/standard deviation, and compute probabilities for various discrete distributions

Usage

```
# For use-defined discrete distribution
discrete.plotpdf(x,fx)
discrete.plotcdf(x,fx)
discrete.summary(x,fx,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE"))
discrete.prob(x,fx,lb)
discrete.prob(x,fx,lb,ub,inclusive=c("none","left","right","both"))

# Discrete Uniform Distribution
duniform.summary(range,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
duniform.prob(range,lb)
duniform.prob(range,lb,ub,inclusive=c("none","left","right","both"))

# Binomial distribution
binomial.summary(n,p,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
binomial.prob(n,p,lb)
binomial.prob(n,p,lb,ub,inclusive=c("none","left","right","both"))

# Geometric distribution
geometric.summary(p,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
geometric.prob(p,lb)
geometric.prob(p,lb,ub,inclusive=c("none","left","right","both"))

# Negative Binomial distribution
negbinom.summary(r,p,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
negbinom.prob(r,p,lb)
negbinom.prob(r,p,lb,ub,inclusive=c("none","left","right","both"))

# Hypergeometric distribution
hypergeo.summary(N,K,n,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
hypergeo.prob(N,K,n,lb)
hypergeo.prob(N,K,n,lb,ub,inclusive=c("none","left","right","both"))

# Poisson distribution
poisson.summary(lambda,L,plotpdf=c("TRUE","FALSE"), plotcdf=c("TRUE","FALSE")))
poisson.prob(lambda,L,lb)
poisson.prob(lambda,L,lb,ub,inclusive=c("none","left","right","both"))
```

Arguments

x	possible values of a user defined discrete random variable

fx probabilities of X=x, the order of entries in fx must matches the order of entries in x.

plotpdf TRUE or FALSE, if TRUE, it plots the pmf

plotcdf TRUE or FALSE, if TRUE, it polts the cdf

lb,ub lower bound (lb) and upper bound (ub) in a probability statement; lb could be -Inf; ub could be Inf; ub cannot be less than lb

inclusive "none": lb<X<ub; "left": lb<=X<ub; "right": lb<X<=ub; "both": lb<=X<=ub

Range contains all possible values of a discrete uniform random variable

p parameter p of a Binomial distribution/Geometric distribution/Negtive Binomial distribution

n parameter n of a Binomial distribution

r parameter r of a negative Binomial distribution

N,K,n parameters N, K, and n of a hypergeometric distribution

lambda,L parameters lambda and L of a Poisson distribution. Default: L=1

Details

Plot the probability mass function (pmf) and the cumulative distribution function (cdf) and calculate probability, mean, variable and standard deviation, and compute probabilities of various discrete distributions (a self-defined discrete distribution, discrete uniform distribution, binomial distribution, geometric distribution, negative binomial distribution, hypergeometric distribution, and Poisson distribution).

Value

discrete.plotpdf
a figure of the pmf of the user-defined discrete distribution

discrete.plotcdf
a figure of the cdf of the user-defined discrete distribution

discrete.summary
a list of the mean, variance, and standard deviation of the user-defined discrete distribution, plot of the pmf/cdf or not

discrete.prob probability of X between lb and ub based on the user-defined discrete distribution

duniform.summary
a list of the mean, variance, and standard deviation of a discrete uniform distribution, plot of the pmf/cdf or not

duniform.prob probability of X between lb and ub based on a uniform distribution

binomial.summary
a list of the mean, variance, and standard deviation of a binomial distribution, plot of the pmf/cdf or not

binomial.prob probability of X between lb and ub based on a binomial distribution

geometric.summary
a list of the mean, variance, and standard deviation of a geometric distribution, plot of the pmf/cdf or not

geometric.prob probability of X between lb and ub based on a geometric distribution

negbinom.summary
a list of the mean, variance, and standard deviation of a negative binomial distribution, plot of the pmf/cdf or not

`negbinom.prob` probability of X between lb and ub based on a negative binomial distribution

`hypergeo.summary` a list of the mean, variance, and standard deviation of a hypergeometric distribution, plot of the pmf/cdf or not

`hypergeo.prob` probability of X between lb and ub based on a hypergeometric distribution

`poisson.summary` a list of the mean, variance, and standard deviation of a Poisson distribution, plot of the pmf/cdf or not

`poisson.prob` probability of X between lb and ub based on a Poisson distribution

Note

deweiwang@stat.sc.edu

Author(s)

Dewei Wang

References

Chapter 3 of the textbook "Applied Statistics and Probability for Engineers" 7th edition

Examples

```
x=c(0,1,2,3,4);fx=c(0.6561,0.2916,0.0486,0.0036,0.0001);

discrete.summary(x,fx)
discrete.summary(x,fx,plotpdf=FALSE,plotcdf=TRUE)
discrete.summary(x,fx,plotpdf=TRUE,plotcdf=FALSE)

discrete.prob(x,fx,2) #P(X=2)

discrete.prob(x,fx,2,4,inclusive="none") #P(2<X<4)
discrete.prob(x,fx,2,4,inclusive="left") #P(2<=X<4)
discrete.prob(x,fx,2,4,inclusive="right") #P(2<X<=4)
discrete.prob(x,fx,2,4,inclusive="both") #P(2<=X<=4)

discrete.prob(x,fx,2,Inf,inclusive="none") #P(2<X)
discrete.prob(x,fx,2,Inf,inclusive="left") #P(2<=X)
discrete.prob(x,fx,2,Inf,inclusive="right") #P(2<X)
discrete.prob(x,fx,2,Inf,inclusive="both") #P(2<=X)

discrete.prob(x,fx,-Inf,4,inclusive="none") #P(X<4)
discrete.prob(x,fx,-Inf,4,inclusive="left") #P(X<4)
discrete.prob(x,fx,-Inf,4,inclusive="right") #P(X<=4)
discrete.prob(x,fx,-Inf,4,inclusive="both") #P(X<=4)

#X~Discrete Uniform(1,2,4,5,8,10)
range=c(1,2,4,5,8,10)
duniform.summary(range)
duniform.prob(range,2) #P(X=2)
duniform.prob(range,2,4,inclusive="left") #P(2<=X<4)

#X~Binomial(n=5,p=0.3)
binomial.summary(5,0.3)
binomial.prob(5,0.3,2) #P(X=2)
binomial.prob(5,0.3,2,4,inclusive="left") #P(2<=X<4)

#X~Geometric(p=0.3)
geometric.summary(0.3)
geometric.prob(0.3,2) #P(X=2)
geometric.prob(0.3,2,4,inclusive="left") #P(2<=X<4)

#X~Negative Binomial(r=3,p=0.3)
negbinom.summary(3,0.3)
negbinom.prob(3,0.3,5) #P(X=5)
negbinom.prob(3,0.3,5,7,inclusive="left") #P(5<=X<7)

#X~Hypergeometric(N=300,K=100,n=4)
hypergeo.summary(300,100,4)
hypergeo.prob(300,100,4,2) #P(X=2)
hypergeo.prob(300,100,4,2,4,inclusive="left") #P(2<=X<4)

#X~Poisson(lambda=2.3,L=5)
poisson.summary(2.3,5)
poisson.prob(2.3,5,10) #P(X=10)
poisson.prob(2.3,5,1,Inf,inclusive="left") #P(1<=X)
```

`One-Sample Inference` *One-Sample Confidence Intervals and Hypothesis Tests*

Description

Compute confidence intervals and conduct hypothesis tests and on the population mean, population variance, and population proportaion. In addition, it computes prediction interval for a single future observation from a normal distribution.

Usage

```
#CI for pupulation mean of a normal distribution when the population variance is known:
Zinterval.data(data,sigma,level)
Zinterval.data(n,barx,sigma,level)
# Choice of sample size for estimating the population mean when error is specified
sample.size.Zinterval(E,sigma,level)

#CI for pupulation mean when the population variance is unknown and the distribution is normal
#or when the sample size is smaller than 25:
Tinterval.data(data,level)
Tinterval.data(n,barx,s,level)

#Large-sample CI for pupulation mean:
AZinterval.data(data,level)
AZinterval.data(n,barx,s,level)

#CI for pupulation variance (or standard deviation) of a normal distribution:
Chi2interval.data(data,level)
Chi2interval.data(n,s,level)
```

```
#Large-sample CI for a pupulation proportion:
Propinterval(n,x,level)
# Choice of sample size for estimating a population proportion when error is specified
sample.size.Propinterval(E,ini.p,level)
sample.size.Propinterval(E,ini.p=0.5,level) # using the conservative apporach

# Prediction interval of a single future observation form a normal distribution:
Predinterval.data(data,level)
Predinterval.stat(n,barx,s,level)
```

Arguments

`level`	the confidence level
`data`	a vector of the observed sample
`sigma`	the known population standard deviation
`s`	the observed sample standard deviation
`barx`	the observed sample mean
`n`	the sample size
`E`	specified error in sample size calculation
`ini.p`	A initial estimate of the populatin proportion. Default is 0.5 which corresponds to the conservative approach
`df`	the degrees of freedom of a t or chi.square distribution
`q`	a quantile value

Details

Compute CIs for the population mean, population variance, and population proportaion and PI for a single future observation from a normal distribution.

Value

`interval`	As long as the function has "interval", the outcome contains a two-sided CI (or PI) and the two one-sided confidence bounds.
`sample.size`	As long as the function has "sample.size", the outcome is the minimum sample size required to control the error to be no larger than E.
`t.quantile`	quantile of a t distribution
`Chi2.quantile`	quantile of a chi.square distribution

Note

deweiwang@stat.sc.edu

Author(s)

Dewei Wang

References

Chapter 8 of the textbook "Applied Statistics and Probability for Engineers" 7th edition

Examples

```
data=c(64.1, 64.7, 64.5, 64.6, 64.5, 64.3, 64.6, 64.8, 64.2, 64.3)
Zinterval.data(data,sigma=1,level=0.95)
Zinterval.data(data,sigma=1,level=0.99)
sample.size.Zinterval(E=0.5,sigma=1,level=0.95)
Zinterval.stat(n=9,barx=98,sigma=2,level=0.95)

Tinterval.stat(n=10,barx=1000,s=20,level=0.95)
Tinterval.stat(n=25,barx=1000,s=20,level=0.95)
Tinterval.stat(n=10,barx=1000,s=20,level=0.99)
Tinterval.stat(n=25,barx=1000,s=20,level=0.99)

data=scan("https://raw.githubusercontent.com/Harrindy/StatEngine/master/Data/Mercury.csv")
AZinterval.data(data,level=0.95)

Chi2interval.stat(n=20,s=0.01532,level=0.95)

Propinterval(n=85,x=10,level=0.95)
sample.size.Propinterval(E=0.05,ini.p=0.12,level=0.95)
sample.size.Propinterval(E=0.05,ini.p=0.5,level=0.95)

data=c(19.8, 10.1, 14.9, 7.5, 15.4, 15.4, 15.4, 18.5, 7.9, 12.7, 11.9, 11.4, 11.4,
14.1, 17.6, 16.7, 15.8, 19.5, 8.8, 13.6, 11.9, 11.4)
Tinterval.data(data,level=0.95)
Predinterval.data(data,level=0.95)
```

Index