

Package ‘Bolstad’

January 2, 2012

Version 0.2-21

Date 2011-11-17

Title Bolstad functions

Author James Curran <curran@stat.auckland.ac.nz>

Maintainer James M. Curran <curran@stat.auckland.ac.nz>

Description A set of R functions and data sets for the book Introduction to Bayesian Statistics, Bolstad, W.M. (2007), John Wiley & Sons ISBN 0-471-27020-2

License GPL (>= 2)

URL <http://www.waikato.ac.nz/Staff/bolstad>

Repository CRAN

Date/Publication 2011-11-18 07:10:37

R topics documented:

bayes.lin.reg	2
binobp	4
binodp	5
binogcp	7
binomixp	8
normdp	10
normgcp	11
normmixp	13
normnp	14
nvaricp	16
poisd	17
poisgamp	18
poisgcp	19
sintegral	21
sscsample	23
sscsample.data	24
xdesign	24

bayes.lin.reg	<i>Bayesian inference for simple linear regression</i>
---------------	--

Description

This function is used to find the posterior distribution of the simple linear regression slope variable β when we have a random sample of ordered pairs (x_i, y_i) from the simple linear regression model:

$$y_i = \alpha_{\bar{x}} + \beta x_i + \epsilon_i$$

where the observation errors are, ϵ_i , independent $normal(0, \sigma^2)$ with known variance.

Usage

```
bayes.lin.reg(y, x, slope.prior = "flat", intcpt.prior = "flat",
             mb0 = 0, sb0 = 0, ma0 = 0, sa0 = 0, sigma = NULL,
             alpha = 0.05, plot.data = FALSE, pred.x = NULL,
             ret = FALSE)
```

Arguments

y	the vector of responses.
x	the value of the explanatory variable associated with each response.
slope.prior	use a “flat” prior or a “normal” prior. for β
intcpt.prior	use a “flat” prior or a “normal” prior. for $\alpha_{[\bar{x}]}$
mb0	the prior mean of the simple linear regression slope variable β . This argument is ignored for a flat prior.
sb0	the prior std. deviation of the simple linear regression slope variable β - must be greater than zero. This argument is ignored for a flat prior.
ma0	the prior mean of the simple linear regression intercept variable $\alpha_{\bar{x}}$. This argument is ignored for a flat prior.
sa0	the prior std. deviation of the simple linear regression variable $\alpha_{\bar{x}}$ - must be greater than zero. This argument is ignored for a flat prior.
sigma	the value of the std. deviation of the residuals. By default, this is assumed to be unknown and the sample value is used instead. This affects the prediction intervals.
alpha	controls the width of the credible interval.
plot.data	if true the data are plotted, and the posterior regression line superimposed on the data.
pred.x	a vector of x values for which the predicted y values are obtained and the std. errors of prediction
ret	this argument is deprecated.

Value

A list will be returned with the following components:

post.coef	the posterior mean of the intercept and the slope
post.sd	the posterior standard deviation of the intercept the slope
pred.x	the vector of values for which predictions have been requested. If pred.x is NULL then this is not returned
pred.y	the vector predicted values corresponding to pred.x. If pred.x is NULL then this is not returned
pred.se	The standard errors of the predicted values in pred.y. If pred.x is NULL then this is not returned

Examples

```
## generate some data from a known model, where the true value of the
## intercept alpha is 2, the true value of the slope beta is 3, and the
## errors come from a normal(0,1) distribution
x = rnorm(50)
y = 22+3*x+rnorm(50)

## use the function with a flat prior for the slope beta and a
## flat prior for the intercept, alpha_xbar.

bayes.lin.reg(y,x)

## use the function with a normal(0,3) prior for the slope beta and a
## normal(30,10) prior for the intercept, alpha_xbar.

bayes.lin.reg(y,x,"n","n",0,3,30,10)

## use the same data but plot it and the credible interval

bayes.lin.reg(y,x,"n","n",0,3,30,10,plot.data = TRUE)

## The heart rate vs. O2 uptake example 14.1
O2 = c(0.47,0.75,0.83,0.98,1.18,1.29,1.40,1.60,1.75,1.90,2.23)
HR = c(94,96,94,95,104,106,108,113,115,121,131)
plot(HR,O2,xlab="Heart Rate",ylab="Oxygen uptake (Percent)")

bayes.lin.reg(y,x,"n","f",0,1,sigma=0.13)

## Repeat the example but obtain predictions for HR = 100 and 110

bayes.lin.reg(y,x,"n","f",0,1,sigma=0.13,pred.x=c(100,110))
```

binobp

Binomial sampling with a beta prior

Description

Evaluates and plots the posterior density for π , the probability of a success in a Bernoulli trial, with binomial sampling and a continuous $beta(a, b)$ prior.

Usage

```
binobp(x, n, a = 1, b = 1, ret = FALSE)
```

Arguments

x	the number of observed successes in the binomial experiment.
n	the number of trials in the binomial experiment.
a	parameter for the beta prior - must be greater than zero
b	parameter for the beta prior - must be greater than zero
ret	this argument is deprecated.

Value

A list will be returned with the following components:

posterior	the posterior density of π given x and n - i.e. the $beta(a + x, b + n - x)$ density
likelihood	the likelihood of x given π and n , i.e. the $binomial(n, \pi)$ density
prior	the prior density of π , i.e. the $beta(a, b)$ density
pi	the values of π for which the posterior density was evaluated
mean	the posterior mean
var	the posterior variance
sd	the posterior std. deviation
quantiles	a set of quantiles from the posterior

See Also

[binodp](#) [binogcp](#)

Examples

```
## simplest call with 6 successes observed in 8 trials and a beta(1,1) uniform
## prior
binodp(6,8)

## 6 successes observed in 8 trials and a non-uniform beta(0.5,6) prior
binodp(6,8,0.5,6)

## 4 successes observed in 12 trials with a non uniform beta(3,3) prior
## plot the stored prior, likelihood and posterior
results = binodp(4,12,3,3)

par(mfrow=c(3,1))
y.lims = c(0,1.1*max(results$posterior,results$prior))

plot(results$pi,results$prior,ylim=y.lims,type="l"
, xlab=expression(pi),ylab="Density",main="Prior")
polygon(results$pi,results$prior,col="red")

plot(results$pi,results$likelihood,ylim=c(0,0.25),type="l"
, xlab=expression(pi),ylab="Density",main="Likelihood")
polygon(results$pi,results$likelihood,col="green")

plot(results$pi,results$posterior,ylim=y.lims,type="l"
, xlab=expression(pi),ylab="Density",main="Posterior")
polygon(results$pi,results$posterior,col="blue")
```

binodp

Binomial sampling with a discrete prior

Description

Evaluates and plots the posterior density for π , the probability of a success in a Bernoulli trial, with binomial sampling and a discrete prior on π

Usage

```
binodp(x,n, pi = NULL, pi.prior = NULL, n.pi = 10, ret = FALSE)
```

Arguments

x the number of observed successes in the binomial experiment.

n the number of trials in the binomial experiment.

pi a vector of possibilities for the probability of success in a single trial. if pi is NULL then a discrete uniform prior for π will be used.

<code>pi.prior</code>	the associated prior probability mass.
<code>n.pi</code>	the number of possible π values in the prior
<code>ret</code>	this argument is deprecated.

Value

A list will be returned with the following components:

<code>pi</code>	the vector of possible π values used in the prior
<code>pi.prior</code>	the associated probability mass for the values in π
<code>likelihood</code>	the scaled likelihood function for π given x and n
<code>posterior</code>	the posterior probability of π given x and n
<code>f.cond</code>	the conditional distribution of x given π and n
<code>f.joint</code>	the joint distribution of x and π given n
<code>f.marg</code>	the marginal distribution of x

See Also

[binobp](#) [binogcp](#)

Examples

```
## simplest call with 6 successes observed in 8 trials and a uniform prior
binodp(6,8)

## same as previous example but with more possibilities for pi
binodp(6,8,n.pi=100)

## 6 successes, 8 trials and a non-uniform discrete prior
pi = seq(0,1,by=0.01)
pi.prior = runif(101)
pi.prior = sort(pi.prior/sum(pi.prior))
binodp(6,8,pi,pi.prior)

## 5 successes, 6 trials, non-uniform prior
pi = c(0.3,0.4,0.5)
pi.prior = c(0.2,0.3,0.5)
results = binodp(5,6,pi,pi.prior)

## plot the results from the previous example using a side-by-side barplot
results.matrix = rbind(results$pi.prior,results$posterior)
colnames(results.matrix) = pi
barplot(results.matrix,col=c("red","blue"),beside=TRUE
,xlab=expression(pi),ylab=expression(Probability(pin)))
box()
legend(1,0.65,legend=c("Prior","Posterior"),fill=c("red","blue"))
```

binogcp

*Binomial sampling with a general continuous prior***Description**

Evaluates and plots the posterior density for π , the probability of a success in a Bernoulli trial, with binomial sampling and a general continuous prior on π

Usage

```
binogcp(x, n, density = "uniform", params = c(0,1), n.pi = 1000,
pi = NULL, pi.prior = NULL, ret = FALSE)
```

Arguments

x	the number of observed successes in the binomial experiment.
n	the number of trials in the binomial experiment.
density	may be one of "beta", "exp", "normal", "student", "uniform" or "user"
params	if density is one of the parameteric forms then then a vector of parameters must be supplied. beta: a,b exp: rate normal: mean,sd uniform: min,max
n.pi	the number of possible π values in the prior
pi	a vector of possibilities for the probability of success in a single trial. This must be set if density="user".
pi.prior	the associated prior probability mass. This must be set if density="user".
ret	this argument is deprecated.

Value

A list will be returned with the following components:

likelihood	the scaled likelihood function for π given x and n
posterior	the posterior probability of π given x and n
pi	the vector of possible π values used in the prior
pi.prior	the associated probability mass for the values in π

See Also

[binobp](#) [binodp](#)

Examples

```

## simplest call with 6 successes observed in 8 trials and a continuous
## uniform prior
binogcp(6,8)

## 6 successes, 8 trials and a Beta(2,2) prior
binogcp(6,8,density="beta",params=c(2,2))

## 5 successes, 10 trials and a N(0.5,0.25) prior
binogcp(5,10,density="normal",params=c(0.5,0.25))

## 4 successes, 12 trials with a user specified triangular continuous prior
pi = seq(0,1,by=0.001)
pi.prior = rep(0,length(pi))
pi.prior[pi<=0.5] = 4*pi[pi<=0.5]
pi.prior[pi>0.5] = 4-4*pi[pi>0.5]
results = binogcp(4,12,"user",pi=pi,pi.prior=pi.prior)

## find the posterior CDF using the previous example and Simpson's rule
## Please note that the syntax of sintegral has changed since the
## second edition of the book
cdf = sintegral(pi,results$posterior,n.pts=length(pi))$cdf
plot(cdf,type="l",xlab=expression(pi[0])
,ylab=expression(Pr(pi<=pi[0])))

## use the cdf to find the 95% credible region.
## Thanks to John Wilkinson for this simplified code.
lcb = cdf$x[with(cdf,which.max(x[y<=0.025]))]
ucb = cdf$x[with(cdf,which.max(x[y<=0.975]))]
cat(paste("Approximate 95% credible interval : ["
,round(lcb,4)," ",round(ucb,4),"]\n",sep=""))

## find the posterior mean, variance and std. deviation
## using Simpson's rule and the output from the previous example
dens = pi*results$posterior # calculate pi*f(pi | x, n)
post.mean = sintegral(pi,dens)$value

dens = (pi-post.mean)^2*results$posterior
post.var = sintegral(pi,dens)$value
post.sd = sqrt(post.var)

# calculate an approximate 95% credible region using the posterior mean and
# std. deviation
lb = post.mean-qnorm(0.975)*post.sd
ub = post.mean+qnorm(0.975)*post.sd

cat(paste("Approximate 95% credible interval : ["
,round(lb,4)," ",round(ub,4),"]\n",sep=""))

```

Description

Evaluates and plots the posterior density for π , the probability of a success in a Bernoulli trial, with binomial sampling when the prior density for π is a mixture of two beta distributions, $beta(a_0, b_0)$ and $beta(a_1, b_1)$.

Usage

```
binomixp(x, n, alpha0=c(1,1), alpha1=c(1,1), p=0.5, ret = FALSE)
```

Arguments

x	the number of observed successes in the binomial experiment.
n	the number of trials in the binomial experiment.
alpha0	a vector of length two containing the parameters, a_0 and b_0 , for the first component beta prior - must be greater than zero. By default the elements of alpha0 are set to 1.
alpha1	a vector of length two containing the parameters, a_1 and b_1 , for the second component beta prior - must be greater than zero. By default the elements of alpha1 are set to 1.
p	The prior mixing proportion for the two component beta priors. That is the prior is $p \times beta(a_0, b_0) + (1 - p) \times beta(a_1, b_1)$. p is set to 0.5 by default
ret	this argument is deprecated.

Value

A list will be returned with the following components:

pi	the values of π for which the posterior density was evaluated
posterior	the posterior density of π given n and x
likelihood	the likelihood function for π given x and n , i.e. the $binomial(n, \pi)$ density
prior	the prior density of π density

See Also

[binodp](#) [binogcp](#) [normmixp](#)

Examples

```
## simplest call with 6 successes observed in 8 trials and a 50:50 mix
## of two beta(1,1) uniform priors
binomixp(6,8)

## 6 successes observed in 8 trials and a 20:80 mix of a non-uniform
## beta(0.5,6) prior and a uniform beta(1,1) prior
binomixp(6,8,alpha0=c(0.5,6),alpha1=c(1,1),p=0.2)

## 4 successes observed in 12 trials with a 90:10 non uniform beta(3,3) prior
## and a non uniform beta(4,12).
```

```
## Plot the stored prior, likelihood and posterior
results = binomixp(4,12,c(3,3),c(4,12),0.9)

par(mfrow=c(3,1))
y.lims = c(0,1.1*max(results$posterior,results$prior))

plot(results$pi,results$prior,ylim=y.lims,type="l"
,xlab=expression(pi),ylab="Density",main="Prior")
polygon(results$pi,results$prior,col="red")

plot(results$pi,results$likelihood,type="l"
,xlab=expression(pi),ylab="Density",main="Likelihood")
polygon(results$pi,results$likelihood,col="green")

plot(results$pi,results$posterior,ylim=y.lims,type="l"
,xlab=expression(pi),ylab="Density",main="Posterior")
polygon(results$pi,results$posterior,col="blue")
```

normdp

Bayesian inference on a normal mean with a discrete prior

Description

Evaluates and plots the posterior density for μ , the mean of a normal distribution, with a discrete prior on μ

Usage

```
normdp(x, sigma.x = NULL, mu = NULL, mu.prior = NULL, n.mu = 50,
ret = FALSE)
```

Arguments

x	a vector of observations from a normal distribution with unknown mean and known std. deviation.
sigma.x	the population std. deviation of the normal distribution
mu	a vector of possibilities for the probability of success in a single trial. If mu is NULL then a uniform prior is used.
mu.prior	the associated prior probability mass.
n.mu	the number of possible μ values in the prior
ret	this argument is deprecated.

Value

A list will be returned with the following components:

mu	the vector of possible μ values used in the prior
mu.prior	the associated probability mass for the values in μ
likelihood	the scaled likelihood function for μ given x and σ_x
posterior	the posterior probability of μ given x and σ_x

See Also

[normnp](#) [normgcp](#)

Examples

```
## generate a sample of 20 observations from a N(-0.5,1) population
x = rnorm(20,-0.5,1)

## find the posterior density with a uniform prior on mu
normdp(x,1)

## find the posterior density with a non-uniform prior on mu
mu = seq(-3,3,by=0.1)
mu.prior = runif(length(mu))
mu.prior = sort(mu.prior/sum(mu.prior))
normdp(x,1,mu,mu.prior)

## Let mu have the discrete distribution with 5 possible
## values, 2, 2.5, 3, 3.5 and 4, and associated prior probability of
## 0.1, 0.2, 0.4, 0.2, 0.1 respectively. Find the posterior
## distribution after a drawing random sample of n = 5 observations
## from a N(mu,1) distribution y = [1.52, 0.02, 3.35, 3.49, 1.82]
mu = seq(2,4,by=0.5)
mu.prior = c(0.1,0.2,0.4,0.2,0.1)
y = c(1.52,0.02,3.35,3.49,1.82)
normdp(y,1,mu,mu.prior)
```

normgcp

Bayesian inference on a normal mean with a general continuous prior

Description

Evaluates and plots the posterior density for μ , the mean of a normal distribution, with a general continuous prior on μ

Usage

```
normgcp(x, sigma.x, density = "uniform" , params = NULL, n.mu = 50, mu =
NULL,mu.prior = NULL, ret = FALSE)
```

Arguments

x	a vector of observations from a normal distribution with unknown mean and known std. deviation.
sigma.x	the population std. deviation of the normal distribution
density	distributional form of the prior density can be one of: "normal", "uniform", or "user".
params	if density = "normal" then params must contain at least a mean and possible a std. deviation. If a std. deviation is not specified then sigma.x will be used as the std. deviation of the prior. If density = "uniform" then params must contain a minimum and a maximum value for the uniform prior. If a maximum and minimum are not specified then a $U[0, 1]$ prior is used
n.mu	the number of possible μ values in the prior
mu	a vector of possibilities for the probability of success in a single trial. Must be set if density="user"
mu.prior	the associated prior probability mass. Must be set if density="user"
ret	this argument is deprecated.

Value

A list will be returned with the following components:

likelihood	the scaled likelihood function for μ given x and σ_x
posterior	the posterior probability of μ given x and σ
mu	the vector of possible μ values used in the prior
mu.prior	the associated probability mass for the values in μ

See Also

[normdp](#) [normnp](#)

Examples

```
## generate a sample of 20 observations from a N(-0.5,1) population
x = rnorm(20,-0.5,1)

## find the posterior density with a uniform U[-3,3] prior on mu
normgcp(x,1,params=c(-3,3))

## find the posterior density with a non-uniform prior on mu
mu = seq(-3,3,by=0.1)
mu.prior = rep(0,length(mu))
mu.prior[mu<=0] = 1/3+mu[mu<=0]/9
mu.prior[mu>0] = 1/3-mu[mu>0]/9
normgcp(x,1,density="user",mu=mu,mu.prior=mu.prior)

## find the CDF for the previous example and plot it
## Note the syntax for sintegral has changed
```

```

results = normgcp(x,1,density="user",mu=mu,mu.prior=mu.prior)
cdf = sintegral(mu,results$posterior,n.pts=length(mu))$cdf
plot(cdf,type="l",xlab=expression(mu[0])
      ,ylab=expression(Pr(mu<=mu[0])))

## use the CDF for the previous example to find a 95%
## credible interval for mu. Thanks to John Wilkinson for this simplified code

lcb = cdf$x[with(cdf,which.max(x[y<=0.025]))]
ucb = cdf$x[with(cdf,which.max(x[y<=0.975]))]
cat(paste("Approximate 95% credible interval : ["
          ,round(lcb,4)," ",round(ucb,4),"]\n",sep=""))

## use the CDF from the previous example to find the posterior mean
## and std. deviation
dens = mu*results$posterior
post.mean = sintegral(mu,dens)$value

dens = (mu-post.mean)^2*results$posterior
post.var = sintegral(mu,dens)$value
post.sd = sqrt(post.var)

## use the mean and std. deviation from the previous example to find
## an approximate 95% credible interval
lb = post.mean-qnorm(0.975)*post.sd
ub = post.mean+qnorm(0.975)*post.sd

cat(paste("Approximate 95% credible interval : ["
          ,round(lb,4)," ",round(ub,4),"]\n",sep=""))

```

normmixp

Bayesian inference on a normal mean with a mixture of normal priors

Description

Evaluates and plots the posterior density for μ , the mean of a normal distribution, with a mixture of normal priors on μ

Usage

```
normmixp(x, sigma.x, prior0, prior1, p = 0.5, n.mu = 100, ret = FALSE)
```

Arguments

x a vector of observations from a normal distribution with unknown mean and known std. deviation.

sigma.x the population std. deviation of the observations

prior0	the vector of length 2 which contains the means and standard deviation of your precise prior
prior1	the vector of length 2 which contains the means and standard deviation of your vague prior
n.mu	the number of possible μ values in the prior
p	the mixing proportion for the two component normal priors
ret	this argument is deprecated.

Value

A list will be returned with the following components:

mu	the vector of possible μ values used in the prior
prior	the associated probability mass for the values in μ
likelihood	the scaled likelihood function for μ given x and σ_x
posterior	the posterior probability of μ given x and σ_x

See Also

[binomixp](#) [normdp](#) [normgcp](#)

Examples

```
## generate a sample of 20 observations from a N(-0.5,1) population
x = rnorm(20,-0.5,1)

## find the posterior density with a N(0,1) prior on mu - a 50:50 mix of
## two N(0,1) densities
normmixp(x,1,c(0,1),c(0,1))

## find the posterior density with 50:50 mix of a N(0.5,3) prior and a
## N(0,1) prior on mu
normmixp(x,1,c(0.5,3),c(0,1))

## Find the posterior density for mu, given a random sample of 4
## observations from N(mu,1), y = [2.99, 5.56, 2.83, 3.47],
## and a 80:20 mix of a N(3,2) prior and a N(0,100) prior for mu
x = c(2.99,5.56,2.83,3.47)
normmixp(x,1,c(3,2),c(0,100),0.8)
```

normnp

Bayesian inference on a normal mean with a normal prior

Description

Evaluates and plots the posterior density for μ , the mean of a normal distribution, with a normal prior on μ

Usage

```
normnp(x, m.x = 0, s.x = 1, sigma.x = NULL, n.mu = 100, ret = FALSE)
```

Arguments

x	a vector of observations from a normal distribution with unknown mean and known std. deviation.
m.x	the mean of the normal prior
s.x	the standard deviation of the normal prior
sigma.x	the population std. deviation of the normal distribution. If this value is NULL, which is by default, then a flat prior is used and m.x and s.x are ignored
n.mu	the number of possible μ values in the prior
ret	this argument is deprecated.

Value

A list will be returned with the following components:

mu	the vector of possible μ values used in the prior
mu.prior	the associated probability mass for the values in μ
likelihood	the scaled likelihood function for μ given x and σ_x
posterior	the posterior probability of μ given x and σ_x
mean	the posterior mean
sd	the posterior standard deviation
qtls	a selection of quantiles from the posterior density

See Also

[normdp](#) [normgcp](#)

Examples

```
## generate a sample of 20 observations from a N(-0.5,1) population
x = rnorm(20,-0.5,1)

## find the posterior density with a N(0,1) prior on mu
normnp(x,sigma=1)

## find the posterior density with N(0.5,3) prior on mu
normnp(x,0.5,3,1)

## Find the posterior density for mu, given a random sample of 4
## observations from N(mu,sigma^2=1), y = [2.99, 5.56, 2.83, 3.47],
## and a N(3,sd=2)$ prior for mu
y = c(2.99,5.56,2.83,3.47)
normnp(y,3,2,1)
```

nvaricp	<i>Bayesian inference for a normal standard deviation with a scaled inverse chi-squared distribution</i>
---------	--

Description

Evaluates and plots the posterior density for σ , the standard deviation of a Normal distribution where the mean μ is known

Usage

```
nvaricp(y, mu, S0, kappa, cred.int = FALSE, alpha = 0.05, ret = FALSE)
```

Arguments

y	a random sample from a $normal(\mu, \sigma^2)$ distribution.
mu	the known population mean of the random sample.
S0	the prior scaling factor.
kappa	the degrees of freedom of the prior.
cred.int	if TRUE then a 100(1-alpha) percent credible interval will be calculated for σ
alpha	controls the width of the credible interval. Ignored if cred.int is FALSE
ret	this argument is deprecated.

Value

A list will be returned with the following components:

sigma	the values of σ for which the prior, likelihood and posterior have been calculated
prior	the prior density for σ
likelihood	the likelihood function for σ given y
posterior	the posterior density of μ given y
S1	the posterior scaling constant
kappa1	the posterior degrees of freedom

Examples

```
## Suppose we have five observations from a normal(mu, sigma^2)
## distribution mu = 200 which are 206.4, 197.4, 212.7, 208.5.
y = c(206.4, 197.4, 212.7, 208.5, 203.4)

## We wish to choose a prior that has a median of 8. This happens when
## S0 = 29.11 and kappa = 1
nvaricp(y,200,29.11,1)

## Same as the previous example but a calculate a 95% credible
```

```
## interval for sigma
nvaricp(y,200,29.11,1,cred.int=TRUE)

## Same as the previous example but a calculate a 95% credible
## interval for sigma by hand. Note that the syntax of sintegral has
## changed
results = nvaricp(y,200,29.11,1,cred.int=TRUE)
attach(results)
cdf = sintegral(sigma,posterior)$cdf
Finv = approxfun(cdf$y,cdf$x)
lb = Finv(0.025)
ub = Finv(0.975)
cat(paste("95% credible interval for sigma: [",
         signif(lb,4),", ", " ", signif(ub,4),"]\n",sep=""))
```

poisdp

Poisson sampling with a discrete prior

Description

Evaluates and plots the posterior density for μ , the mean rate of occurrence in a Poisson process and a discrete prior on μ

Usage

```
poisdp(y.obs, mu, mu.prior, ret = FALSE)
```

Arguments

y.obs	a random sample from a Poisson distribution.
mu	a vector of possibilities for the mean rate of occurrence of an event over a finite period of space or time.
mu.prior	the associated prior probability mass.
ret	this argument is deprecated.

Value

A list will be returned with the following components:

likelihood	the scaled likelihood function for μ given y_{obs}
posterior	the posterior probability of μ given y_{obs}
mu	the vector of possible μ values used in the prior
mu.prior	the associated probability mass for the values in μ

See Also

[poisgamp](#) [poisgcp](#)

Examples

```
## simplest call with an observation of 4 and a uniform prior on the
## values mu = 1,2,3
poisdp(4,1:3,c(1,1,1)/3)

## Same as the previous example but a non-uniform discrete prior
mu = 1:3
mu.prior = c(0.3,0.4,0.3)
poisdp(4,mu=mu,mu.prior=mu.prior)

## Same as the previous example but a non-uniform discrete prior
mu = seq(0.5,9.5,by=0.05)
mu.prior = runif(length(mu))
mu.prior = sort(mu.prior/sum(mu.prior))
poisdp(4,mu=mu,mu.prior=mu.prior)

## A random sample of 50 observations from a Poisson distribution with
## parameter mu = 3 and non-uniform prior
y.obs = rpois(50,3)
mu = c(1:5)
mu.prior = c(0.1,0.1,0.05,0.25,0.5)
results = poisdp(y.obs, mu, mu.prior)

## Same as the previous example but a non-uniform discrete prior
mu = seq(0.5,5.5,by=0.05)
mu.prior = runif(length(mu))
mu.prior = sort(mu.prior/sum(mu.prior))
y.obs = rpois(50,3)
poisdp(y.obs,mu=mu,mu.prior=mu.prior)
```

poisgamp

Poisson sampling with a gamma prior

Description

Evaluates and plots the posterior density for μ , the mean rate of occurrence in a Poisson process and a *gamma* prior on μ

Usage

```
poisgamp(y, r, v, ret = FALSE)
```

Arguments

y	a random sample from a Poisson distribution.
r	the shape parameter of the <i>gamma</i> prior.
v	the rate parameter of the <i>gamma</i> prior. Note that the scale is $1/v$
ret	This argument is deprecated and has no effect

Value

A list will be returned with the following components:

prior	the prior density assigned to μ
likelihood	the scaled likelihood function for μ given y
posterior	the posterior probability of μ given y
r	the shape parameter for the <i>gamma</i> posterior
v	the rate parameter for the <i>gamma</i> posterior

See Also

[poisdpoisgcp](#)

Examples

```
## simplest call with an observation of 4 and a gamma(1,1), i.e. an exponential prior on the
## mu
poisgamp(4,1,1)

## Same as the previous example but a gamma(10,1) prior
poisgamp(4,10,1)

## Same as the previous example but an improper gamma(1,0) prior
poisgamp(4,1,0)

## A random sample of 50 observations from a Poisson distribution with
## parameter mu = 3 and gamma(6,3) prior
y = rpois(50,3)
poisgamp(y,6,3)

## In this example we have a random sample from a Poisson distribution
## with an unknown mean. We will use a gamma(6,3) prior to obtain the
## posterior gamma distribution, and use the R function qgamma to get a
## 95% credible interval for mu
y = c(3,4,4,3,3,4,2,3,1,7)
results = poisgamp(y,6,3)
ci = qgamma(c(0.025,0.975),results$r, results$v)
cat(paste("95% credible interval for mu: [",round(ci[1],3), ",", round(ci[2],3)),"]\n")
```

poisgcp

Poisson sampling with a general continuous prior

Description

Evaluates and plots the posterior density for μ , the mean rate of occurrence of an event or objects, with Poisson sampling and a general continuous prior on μ

Usage

```
poisgcp(y, density = "normal", params=c(0,1), n.mu = 100
        ,mu = NULL, mu.prior = NULL,
        print.sum.stat=FALSE, alpha=0.05, ret=FALSE)
```

Arguments

y	A random sample of one or more observations from a Poisson distribution
density	may be one of "gamma", "normal", or "user"
params	if density is one of the parameteric forms then then a vector of parameters must be supplied. gamma: a0,b0 normal: mean,sd
n.mu	the number of possible μ values in the prior. This number must be greater than or equal to 100. It is ignored when density="user".
mu	a vector of possibilities for the mean of a Poisson distribution. This must be set if density="user".
mu.prior	the associated prior density. This must be set if density="user".
print.sum.stat	if set to TRUE then the posterior mean, posterior variance, and a credible interval for the mean are printed. The width of the credible interval is controlled by the parameter alpha.
alpha	The width of the credible interval is controlled by the parameter alpha.
ret	this arguement is deprecated.

Value

A list will be returned with the following components:

mu	the vector of possible μ values used in the prior
mu.prior	the associated probability mass for the values in μ
likelihood	the scaled likelihood function for μ given y
posterior	the posterior probability of μ given y

See Also

[poisd](#) [poisgamp](#)

Examples

```
## Our data is random sample is 3, 4, 3, 0, 1. We will try a normal
## prior with a mean of 2 and a standard deviation of 0.5.
y = c(3,4,3,0,1)
poisgcp(y,density="normal",params=c(2,0.5))

## The same data as above, but with a gamma(6,8) prior
y = c(3,4,3,0,1)
poisgcp(y,density="gamma",params=c(6,8))

## The same data as above, but a user specified continuous prior.
```

```

## We will use print.sum.stat to get a 99% credible interval for mu.
y = c(3,4,3,0,1)
mu = seq(0,8,by=0.001)
mu.prior = c(seq(0,2,by=0.001),rep(2,1999),seq(2,0,by=-0.0005))/10
poisgcp(y,"user",mu=mu,mu.prior=mu.prior,print.sum.stat=TRUE,alpha=0.01)

## find the posterior CDF using the results from the previous example
## and Simpson's rule. Note that the syntax of sintegral has changed.
results = poisgcp(y,"user",mu=mu,mu.prior=mu.prior)
cdf = sintegral(mu,results$posterior,n.pts=length(mu))$cdf
plot(cdf,type="l",xlab=expression(mu[0])
,ylab=expression(Pr(mu<=mu[0])))

## use the cdf to find the 95% credible region.
lcb = cdf$x[with(cdf,which.max(x[y<=0.025]))]
ucb = cdf$x[with(cdf,which.max(x[y<=0.975]))]
cat(paste("Approximate 95% credible interval : ["
,round(lcb,4)," ",round(ucb,4),"]\n",sep=""))

## find the posterior mean, variance and std. deviation
## using Simpson's rule and the output from the previous example
dens = mu*results$posterior # calculate mu*f(mu | x, n)
post.mean = sintegral(mu,dens)$value

dens = (mu-post.mean)^2*results$posterior
post.var = sintegral(mu,dens)$value
post.sd = sqrt(post.var)

# calculate an approximate 95% credible region using the posterior mean and
# std. deviation
lb = post.mean-qnorm(0.975)*post.sd
ub = post.mean+qnorm(0.975)*post.sd

cat(paste("Approximate 95% credible interval : ["
,round(lb,4)," ",round(ub,4),"]\n",sep=""))

```

sintegral

Numerical integration using Simpson's Rule

Description

Takes a vector of x values and a corresponding set of positive $f(x) = y$ values, or a function, and evaluates the area under the curve:

$$\int f(x)dx$$

Usage

```
sintegral(x, fx, n.pts = 256, ret = FALSE)
```

Arguments

x	a sequence of x values.
fx	the value of the function to be integrated at x or a function
n.pts	the number of points to be used in the integration.
ret	this argument is deprecated (and ignored)

Value

A list containing two elements, value - the value of the intergral, and cdf - a list containing elements x and y which give a numeric specification of the cdf.

Examples

```
## integrate the normal density from -3 to 3
x = seq(-3, 3, length = 100)
fx = dnorm(x)
estimate = sintegral(x,fx)$value
true.val = diff(pnorm(c(-3,3)))
abs.error = abs(estimate-true.val)
rel.pct.error = 100*abs(estimate-true.val)/true.val
cat(paste("Absolute error :",round(abs.error,7),"\n"))
cat(paste("Relative percentage error :",round(rel.pct.error,6),"percent\n"))

## repeat the example above using dnorm as function
x = seq(-3, 3, length = 100)
estimate = sintegral(x,dnorm)$value
true.val = diff(pnorm(c(-3,3)))
abs.error = abs(estimate-true.val)
rel.pct.error = 100*abs(estimate-true.val)/true.val
cat(paste("Absolute error :",round(abs.error,7),"\n"))
cat(paste("Relative percentage error :",round(rel.pct.error,6)," percent\n"))

## use the cdf

cdf = sintegral(x,dnorm)$cdf
plot(cdf, type = 'l', col = "black")
lines(x, pnorm(x), col = "red", lty = 2)

## integrate the function x^2-1 over the range 1-2
x = seq(1,2,length = 100)
sintegral(x,function(x){x^2-1})$value

## compare to integrate
integrate(function(x){x^2-1},1,2)
```

 sscsample

Simple, Stratified and Cluster Sampling

Description

Samples from a fixed population using either simple random sampling, stratified sampling or cluster sampling.

Usage

```
sscsample(size, n.samples, sample.type="simple", x = NULL,
          strata = NULL, cluster = NULL, ret=FALSE, print = TRUE)
```

Arguments

size	the desired size of the sample
n.samples	the number of repeat samples to take
sample.type	the sampling method. Can be one of "simple", "stratified", "cluser" or 1, 2, 3 where 1 corresponds to "simple", 2 to "stratified" and 3 to "cluster" Do not set the following values unless you know what you are doing!
x	a vector of measurements for each unit in the population. By default x is not used, and the builtin data set sscsample.data is used
strata	a corresponding vector for each unit in the population indicating membership to a stratum
cluster	a corresponding vector for each unit in the population indicating membership to a cluster
ret	this argument is deprecated
print	if false then the screen output will be suppressed.

Value

A list will be returned with the following components:

samples	a matrix with the number of rows equal to size and the number of columns equal to n.samples. Each column corresponds to a sample drawn from the population
s.strata	a matrix showing how many units from each stratum were included in the sample
means	a vector containing the mean of each sample drawn

Author(s)

James M. Curran, Dept. of Statistics, University of Waikato, Hamilton, New Zealand. Janko Dietzsch, Proteomics Algorithm and Simulation, Zentrum f. Bioinformatik Tuebingen Fakultät f. Informations- und Kognitionswissenschaften, Universität Tuebingen

See Also

[sscsample.data](#)

Examples

```
## Draw 200 samples of size 20 using simple random sampling
sscsample(20,200)

## Draw 200 samples of size 20 using simple random sampling and store the
## results. Extract the means of all 200 samples, and the 50th sample
res = sscsample(20,200)
res$means
res$samples[,50]
```

sscsample.data	<i>A stratified and clustered data set</i>
----------------	--

Description

A data set with predetermined strata and clusters. This data set is mostly for internal use with the sscsample function

Usage

```
data(sscsample.data)
```

Format

value: the measurement taken on each sample unit

stratum: the stratum the sample unit belongs to

cluster: the cluster the sample unit belongs to

xdesign	<i>Monte Carlo study of randomized and blocked designs</i>
---------	--

Description

Simulates completely randomized design and randomized block designs from a population of experimental units with underlying response values y and underlying other variable values x (possibly lurking)

Usage

```
xdesign(x = NULL, y = NULL, corr = 0.8, size = 20, n.treatments = 4
, n.rep = 500)
```

Arguments

x	a set of lurking values which are correlated with the response
y	a set of response values
corr	the correlation between the response and lurking variable
size	the size of the treatment groups
n.treatments	the number of treatments
n.rep	the number of Monte Carlo replicates

Value

If the output of `xdesign` is assigned to a variable, then a list is returned with the following components:

block.means	a vector of the means of the lurking variable from each replicate of the simulation stored by treatment number within replicate number
treat.means	a vector of the means of the response variable from each replicate of the simulation stored by treatment number within replicate number
ind	a vector containing the treatment group numbers. Note that there will be twice as many group numbers as there are treatments corresponding to the simulations done using a completely randomized design and the simulations done using a randomized block design

Examples

```
# Carry out simulations using the default parameters
xdesign()
# Carry out simulations using a simulated response with 5 treatments, groups of size 25, and a correlation of -0.6 bet
xdesign(corr=-0.6,size=25,n.treatments=5)
```

Index

*Topic **datasets**

sscsample.data, [24](#)

*Topic **misc**

bayes.lin.reg, [2](#)

binobp, [4](#)

binodp, [5](#)

binogcp, [7](#)

binomixp, [8](#)

normdp, [10](#)

normgcp, [11](#)

normmixp, [13](#)

normnp, [14](#)

nvaricp, [16](#)

poisdp, [17](#)

poisgamp, [18](#)

poisgcp, [19](#)

sintegral, [21](#)

sscsample, [23](#)

xdesign, [24](#)

bayes.lin.reg, [2](#)

binobp, [4](#), [6](#), [7](#)

binodp, [4](#), [5](#), [7](#), [9](#)

binogcp, [4](#), [6](#), [7](#), [9](#)

binomixp, [8](#), [14](#)

normdp, [10](#), [12](#), [14](#), [15](#)

normgcp, [11](#), [11](#), [14](#), [15](#)

normmixp, [9](#), [13](#)

normnp, [11](#), [12](#), [14](#)

nvaricp, [16](#)

poisdp, [17](#), [19](#), [20](#)

poisgamp, [17](#), [18](#), [20](#)

poisgcp, [17](#), [19](#), [19](#)

sintegral, [21](#)

sscsample, [23](#)

sscsample.data, [24](#), [24](#)

xdesign, [24](#)