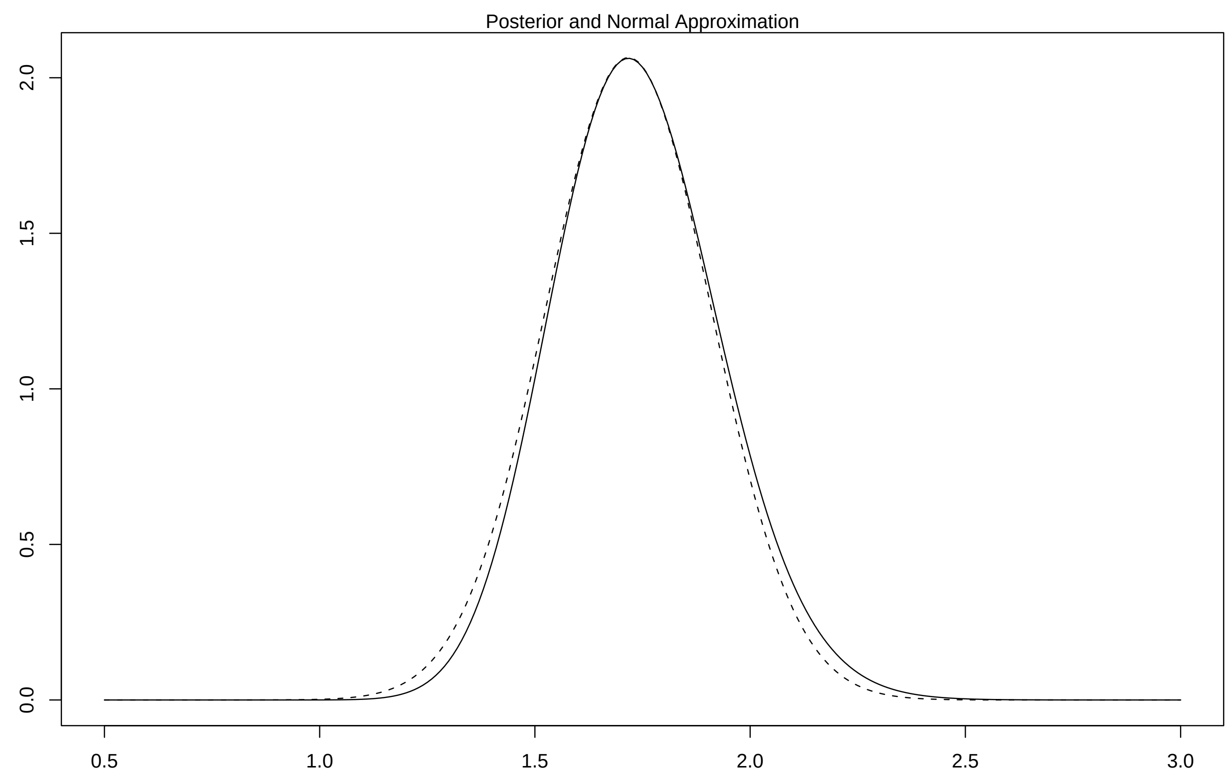
S/R code for Poisson example, Normal and Laplace approximations

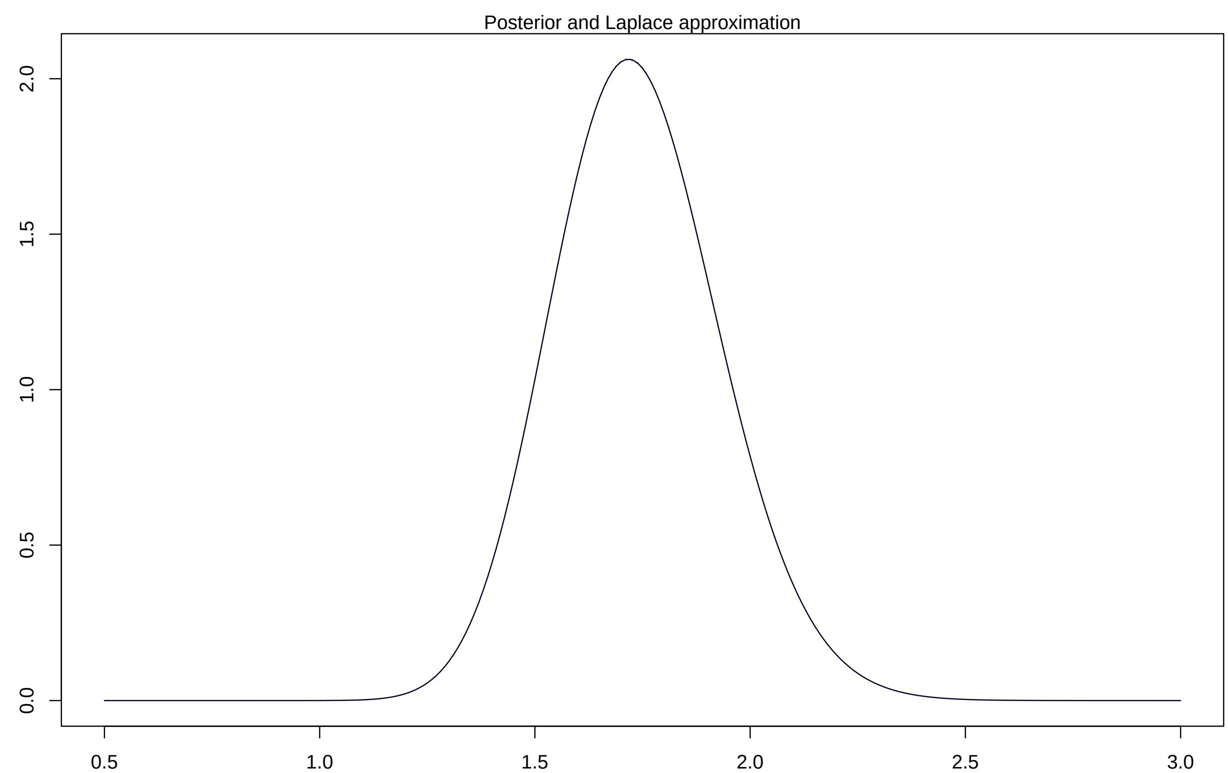
Data is taken from Ex. 8 of Tanner's book Tools for Stat. Inference. Prior is Gamma(2,1)

```
# Data
x <- c(rep(0,6), rep(1,18),rep(2,9),rep(3,7),rep(4,4),5)
n <- length(x)
# Hyperparameters prior
a <-2
b <-1
# Hyperparameters posterior
ast <- a + sum(x)
bst <- b + n
z <- seq(0.5,3.0,.01)
y <- dgamma(z,shape=ast,rate=bst)
plot(z,y,type='l',xlab=" ",ylab=" ")
mtext("Posterior and Normal Approximation")
# Normal approximation
mu <- (a + sum(x)-1)/(n+b)
sg <- (a + sum(x)-1)/(n+b)^2
yy <- dnorm(z,mu,sqrt(sg))
lines(z,yy,type='l',lty=2)
```


Posterior and Normal Approximation
0.0
0.5
1.0
1.5
2.0
0.5
1.0
1.5
2.0
2.5
3.0

Code for Laplace's approximation to the posterior.

```
# Laplace's method. function f=1 and
# exp(-nh(theta))=p(theta)*L(theta|x)
lamhat <- mu
# lamhat is mode of posterior
sig <- ( (-1/n)*(-1/sg))^{-1/2}
ihat <- sqrt(2*pi/n)*sig*((lamhat)^{ast-1})
*exp(-lamhat*bst)
lprox <- (z^{ast-1}*exp(-z*bst))/ihat
plot(z,y,type='l',xlab=" ",ylab=" ")
lines(z,lprox,lty=3,col=4)
mtext("Posterior and Laplace approximation")
```


Posterior and Laplace approximation
0.0
0.5
1.0
1.5
2.0
0.5
1.0
1.5
2.0
2.5
3.0

Computations for first moment $E(\lambda|data)$

```
m <- ast/bst
#  1st order Laplace approximation
m1 <- lamhat
#  2nd order Laplace approximation
m3 <- (ast/bst)* ( (ast/(ast-1))^{ast-0.5})*(exp(-1))
c(m,m1,m3)
[1] 1.739130 1.717391 1.739153
# Relative error
m3/m
[1] 1.000013
```

Stochastic Simulation

- Generate "random" numbers using the computer and that resemble the probabilitic properties of $f(x)$.
- The starting point of stochastic simulation is a random generator that produces uniform values on the $[0,1]$ interval or $X \sim U(0,1)$ (congruential algorithms).

```
u <- runif(1000)
```

Generation of discrete random variables

- Say X is a random variable that takes the values $\{x_1, x_2, \ldots, x_k\}$ with probabilities $\{p_1, p_2, \ldots, p_k\}$
- Split the interval $[0,1]$ into k intervals, $I_1, I_2, \ldots, I_k$ with $I_i = (F_{i-1}, F_i]$ where $F_0 = 0$ and $F_i = p_1 + p_2 + \ldots + p_i; i = 1, 2, \ldots, k$
- Generate u from a $U(0,1)$ and if it checks that u belongs to I_i, then the generated value for x is x_i.

Example: $X \sim Bernoulli(p = 0.4)$

- X may only take two values 1 or 0.
- The probabilities are $p_1 = 0.4$ and $p_2 = 0.6$.
- Then, $I_1 = (0, 0.4]$ and $I_2 = (0.4, 1]$.
- Generate $u \sim U(0, 1)$.
- u falls in I_1 then $x = 1$, if u falls in I_2 then $x = 0$.

In Splus/R we can generate from several discrete distributions

```
rbinom(100,size=1,prob=0.4)
rgeom(100,prob=0.7)
rnbinom(100, size=20,prob=0.3)
rdpois(100,lambda=2.2)
```

Generation of continuous random variables

Probability integral transform

- If X is a continuous random variable with pdf $f(x)$ and cdf $F(x)$, then $U = F(X)$ is a $U[0,1]$ random variable.
- Then we generate $U \sim U[0,1]$ and then make $X = F^{-}(U)$

Example X has an exponential distribution of parameter λ, $F(x) = 1 - exp(-\lambda x)$.

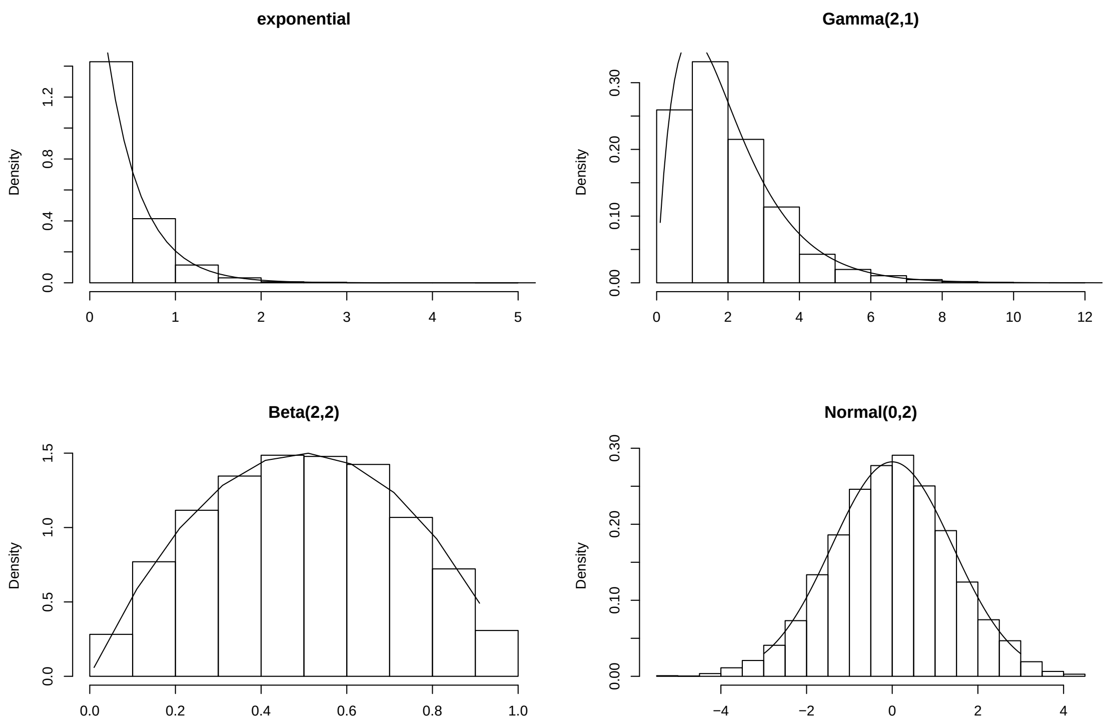
If we set $U = 1 - exp(-\lambda X)$ and solve for X ,we get $X = -(1/\lambda)log(1 - U)$. However, we have ways of simulating from several continuous distributions:

```
rexp(1000,rate=2.5)
rgamma(1000,shape=2,rate=1)
rbeta(1000,shape1=2,shape2=2)
rnorm(1000, mean=0, sd=sqrt(2))
```

Summarizing a Simulation

Histogram A histogram of several (100's or 1000's) simulated values allows us to picture a pdf or pmf

```
par(mfrow=c(2,2))
hist(rexp(5000,rate=2.5),prob=T,xlab=" ",
main="exponential")
x <- seq(0.1,20,by=0.1)
lines(x,dexp(x,rate=2.5))
hist(rgamma(5000,shape=2,rate=1),prob=T,xlab=" ",
main="Gamma(2,1)")
x <- seq(0.1,20,by=0.1)
lines(x,dgamma(x,shape=2,rate=1))
hist(rbeta(5000,shape1=2,shape2=2),prob=T,xlab=" ",
main="Beta(2,2)")
x <- seq(0.01,1.0,by=0.1)
lines(x,dbeta(x,shape1=2,shape2=2))
hist(rnorm(5000,mean=0,sd=sqrt(2)),prob=T,xlab=" ",
main="Normal(0,2)")
x <- seq(-3,3,by=0.01)
lines(x,dnorm(x,mean=0,sd=sqrt(2)))
```


exponential
Gamma(2,1)
Beta(2,2)
Normal(0,2)
Density
0.0
0.4
0.8
1.2
0
1
2
3
4
5
0.00
0.10
0.20
0.30
6
8
10
12
0.5
1.0
1.5
0.2
0.6
−4
−2

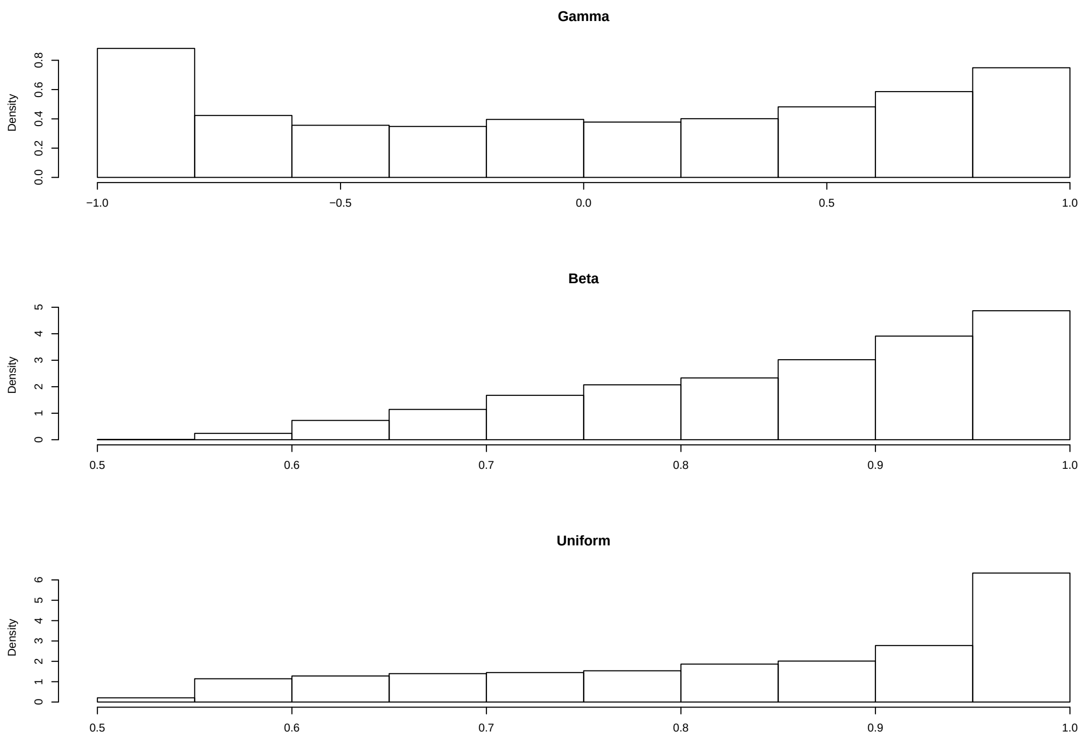
Approximation to the pdf of $Y = g(X)$ This problem utilizes the inverse function of g and the pdf of X. However if g^{-1} or $f(x)$ are difficult to handle analytically,

- simulate values $x_1, x_2, x_3, \ldots, x_m$ from $f(x)$
- transform values to $y_1 = g(x_1), y_2 = g(x_2), y_3 = g(x_3), \ldots, y_m = g(x_m)$

The sample $y_1, y_2, \ldots, y_m$ comes from the pdf of Y!

Example Let X follow a $Gamma(2,1)$ ($Beta(2,2)$) pdf and suppose $Y = cos(X)$. Find (at least approximately) the pdf of Y.

```
par(mfrow=c(3,1))
hist(cos(rgamma(5000,2,1)),prob=T,xlab=" ",
main="Gamma")
hist(cos(rbeta(5000,2,2)),prob=T,xlab=" ",
main="Beta")
hist(cos(runif(5000,0,1)),prob=T,xlab=" ",
main="Uniform")
```



Gamma
Density
0.0
0.2
0.4
0.6
0.8
−1.0
−0.5
0.0
0.5
1.0
Beta
Density
0
1
2
3
4
5
0.5
0.6
0.7
0.8
0.9
1.0
Uniform
Density
0
1
2
3
4
5
6
0.5
0.6
0.7
0.8
0.9
1.0

Approximation of means, SD, and quantiles

```
mean(rgamma(5000,2,1))
[1] 1.957874
sqrt(var(rgamma(10000,2,1)))
[1] 1.382089
quantile(rgamma(10000,2,1),prob=0.35)
     35%
1.230220
qgamma(0.35,2,1)
[1] 1.235044
quantile(rgamma(10000,2,1),prob=0.75)
     75%
2.688899
qgamma(0.75,2,1)
[1] 2.692635
> quantile(rnorm(1000,0,1),prob=0.95)
     95%
1.652202
 qnorm(0.95,0,1)
[1] 1.644854
```

Conclusion: Based on our simulated values we can approximate anything we want to know about the pdf (or pmf) $f(x)$.

Monte Carlo Integration

Suppose we have a continuous random variable X with pdf $f(x)$. Our problem is to compute the integral

$$J = \int g(x)f(x)dx = E(g(x))$$

A way to solve this is by

- Simulation of m values from $f(x)$, $x_1, x_2, \ldots, x_m$
- Approximate the integral through the average

$$\hat{J} = \frac{1}{m}\sum_{i=1}^{m} g(x_i)$$

Aditionally, the standard error of $\hat{J}$ is

$$\frac{1}{\sqrt{m}}\sqrt{\frac{\sum_{i=1}^{m}(g(x_i) - \hat{J})^2}{m-1}}$$

By the large law of number $\hat{J}$ converges A.S. to J.

Example (Tanner): Evaluate $\int_0^1 \sqrt{1-x^2}dx$. We know the result is $\pi/4 = 0.7854\ldots$.

Make $g(x) = \sqrt{1-x^2}$, $f(x) = U(0,1)$. With $m = 5000$ simulations, $\hat{J} = 0.7865$ with a standard error of 0.0032.

```
x <- runif(5000,0,1)
jhat <- mean(sqrt(1-x^2))
se <- sqrt((1/5000)*var( x-jhat))
jhat
[1] 0.7818319
se
[1] 0.00403632
```

Example 2×2 table

	Population 1	Population 2
Success	x	y
Failure	m-x	n-y
Total	m	n

where m and n are fixed by design. Suppose X and Y are independent and

$$X \sim Binomial(m, \pi); \quad Y \sim Binomial(n, \rho)$$

Lets adopt Beta priors for π and ρ or $\pi \sim Beta(\alpha_0, \beta_0)$ and $\rho \sim Beta(\gamma_0, \delta_0)$.

Posterior distributions:

- $\pi \sim Beta(\alpha_0 + x, \beta_0 + m - x)$
- $\rho \sim Beta(\gamma_0 + y, \delta_0 + n - y)$

Suppose the interest lies on the difference $\pi - \rho$. We need to use the posterior distribution $p(\pi - \rho | data)$.

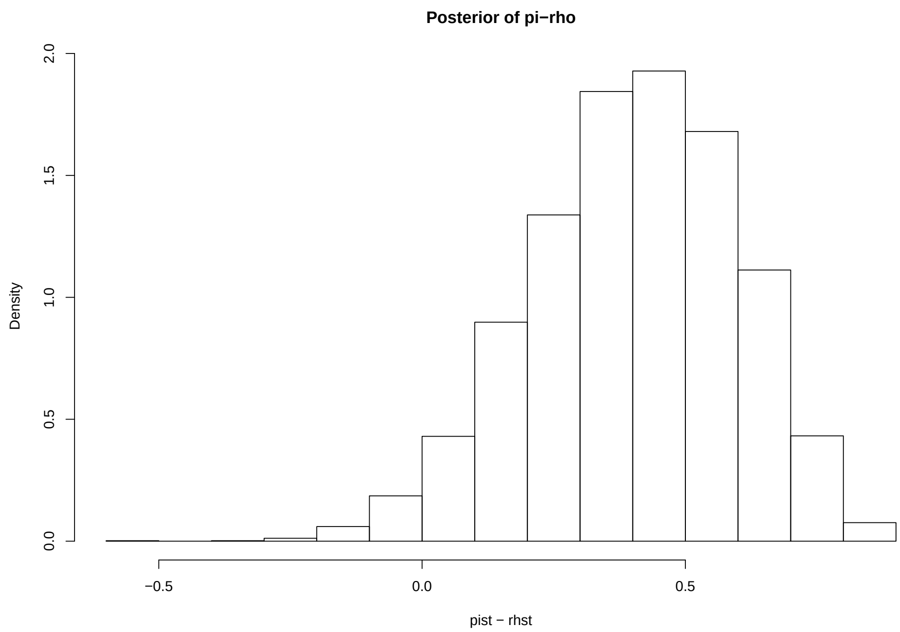
Via simulation it is straightforward to generate values for the posterior distribution of $\pi - \rho$. To obtain an observation from $p(\pi - \rho|data)$

- Simulate $\pi^* \sim Beta(\alpha_0 + x, \beta_0 + m - x)$
- Simulate $\rho^* \sim Beta(\beta_0 + y, \beta_0 + n - y)$
- Compute $\pi^* - \rho^*$

For that matter, the posterior of any function of (π, ρ) can be obtain on the same way. For example, the well known *log-odds ratio*.

Suppose $m = 18$, $x = 14$, $n = 6$, $y = 2$ and that we adopt Jeffrey's prior (Beta(0.5,0.5)) for both π and ρ $(\alpha_0 = \beta_0 = \gamma_0 = \delta_0 = 0.5)$

```
pist <- rbeta(5000,14.5,4.5)
rhst <- rbeta(5000,2.5,4.5)
hist(pist-rhst,prob=T,main=''Posterior of pi-rho'')
mean(pist-rhst)
[1] 0.4019447
var(pist-rhst)
[1] 0.03788451
```



What is the $P(\pi - \rho > 0|data)$?

$$P(\pi - \rho > 0|data) = \int_{\pi-\rho>0} p(\pi - \rho|data)$$

By Monte Carlo integration,

```
mean ( pist-rhst >0)
[1] 0.9738
```