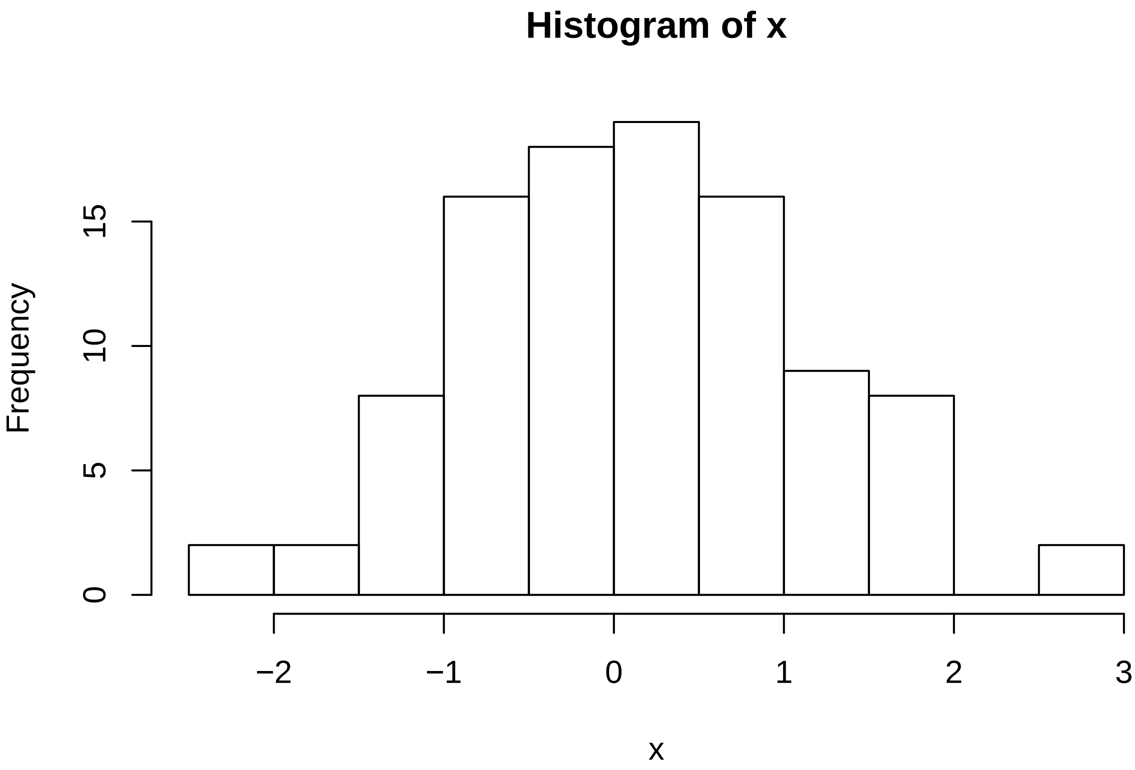
From r-exercises

Hacking statistics Exercises Part 1

Exercise 1

Use rnorm() to generate 100 points, then plot those points in an histogram.

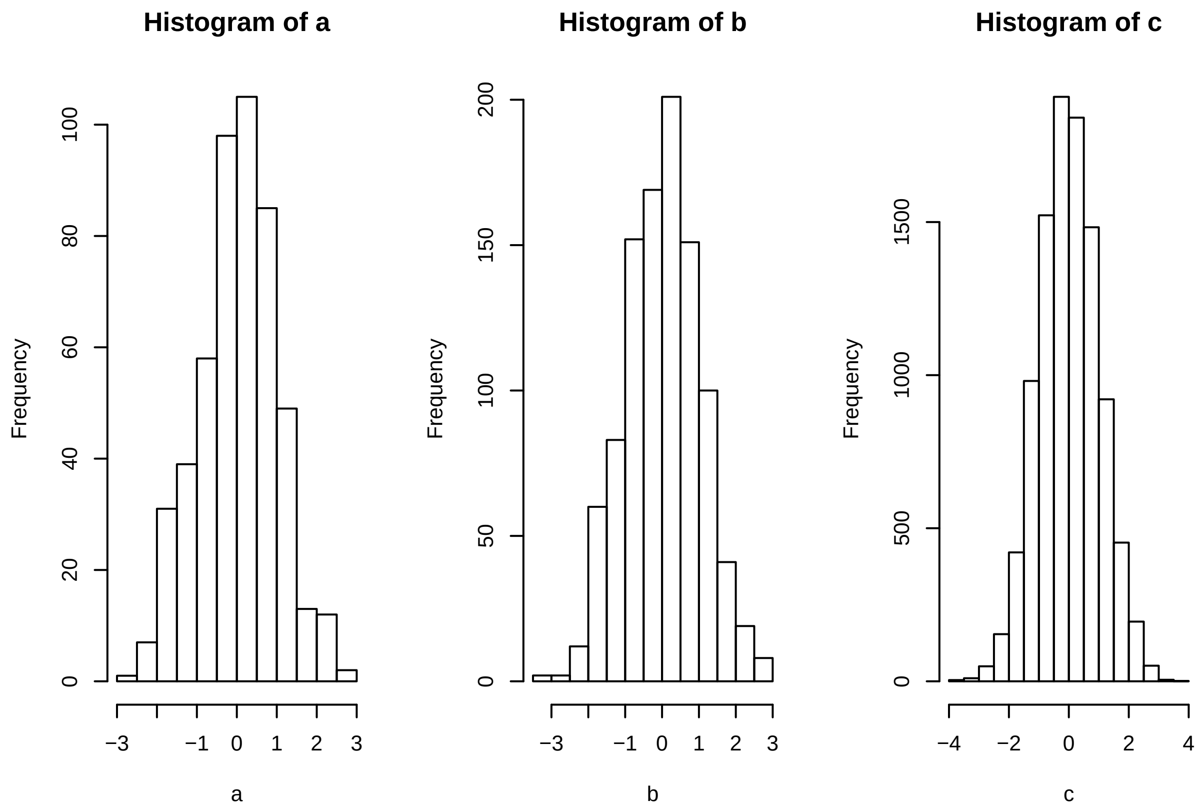
```
x <- rnorm(100)
hist(x)
```



Exercise 2

Repeat exercise 1, but this time with, 500, 1000 and 10000 points.

```
a <- rnorm(500)
b <- rnorm(1000)
c <- rnorm(10000)
par(mfrow =c(1,3))
hist(a)
hist(b)
hist(c)
```

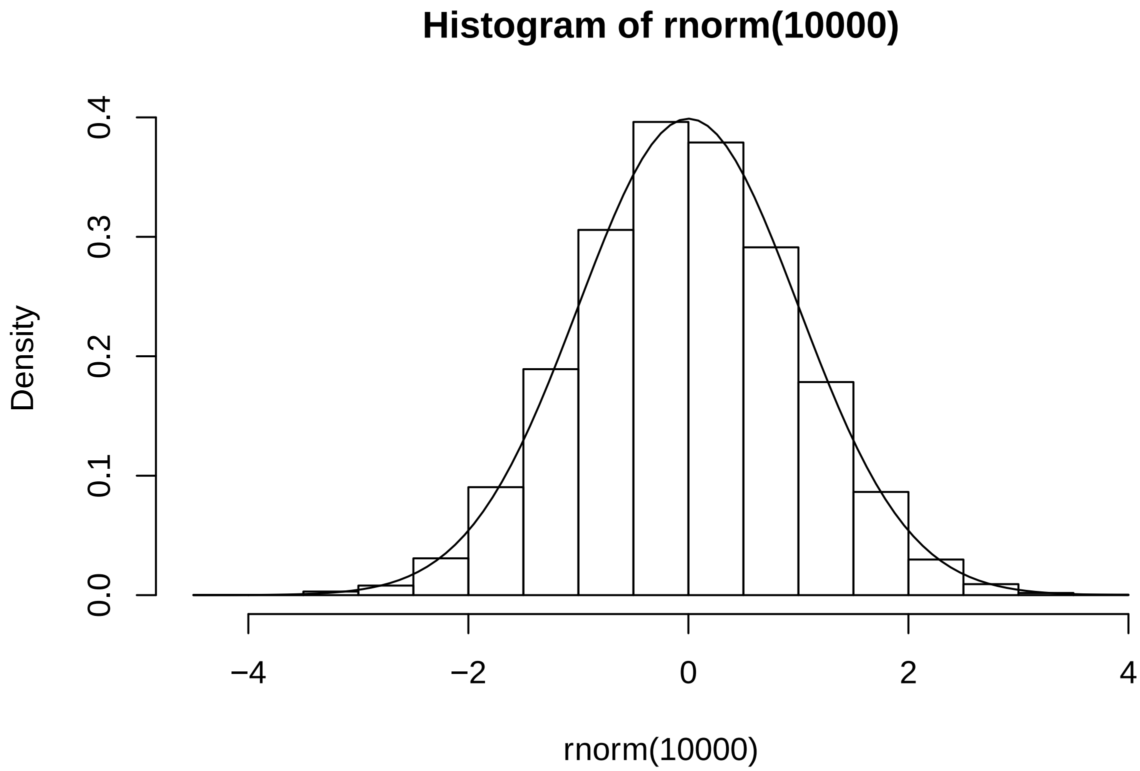


We see that the more points, the more the histgram become symmetric and certered around 0.

Exercise 3

Use the dnorm() function to plot the density function of a normal distribution of mean 0 and standard deviation of 1 and add it to the last histogram you plot.

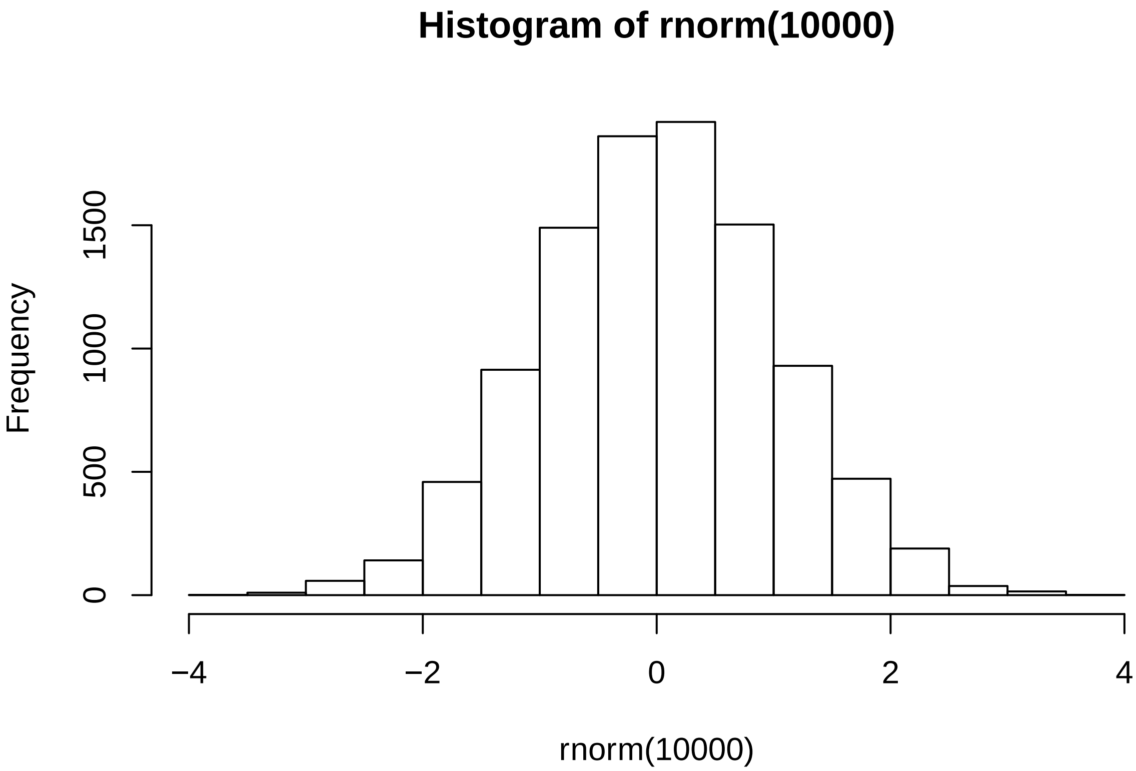
```
x <- seq(-4, 4, 0.1)
hist(rnorm(10000), prob=TRUE)
curve(dnorm(x, mean=0, sd=1), add = TRUE)
```



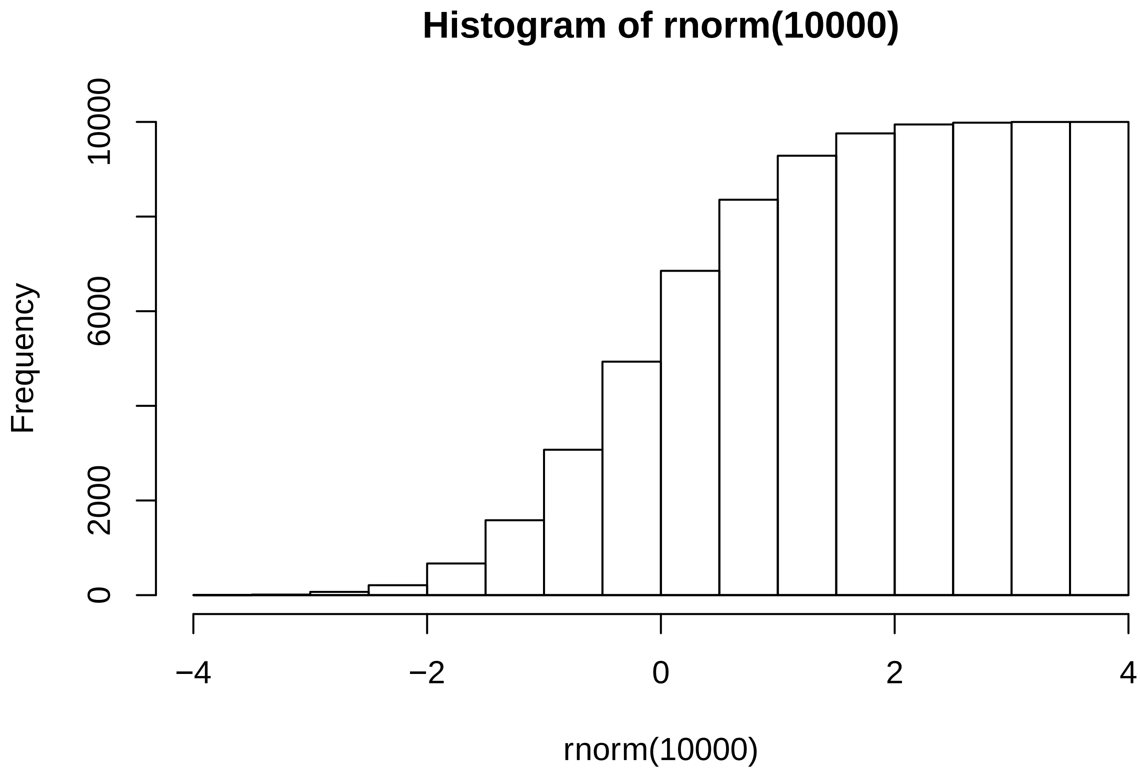
Exercise 4

We use the shape to verify if a random process is a normal process. use pnorm() to plot empirical cumulative distribution function (ECDF) curve.

```
hist.cum.sum <- hist(rnorm(10000))
```



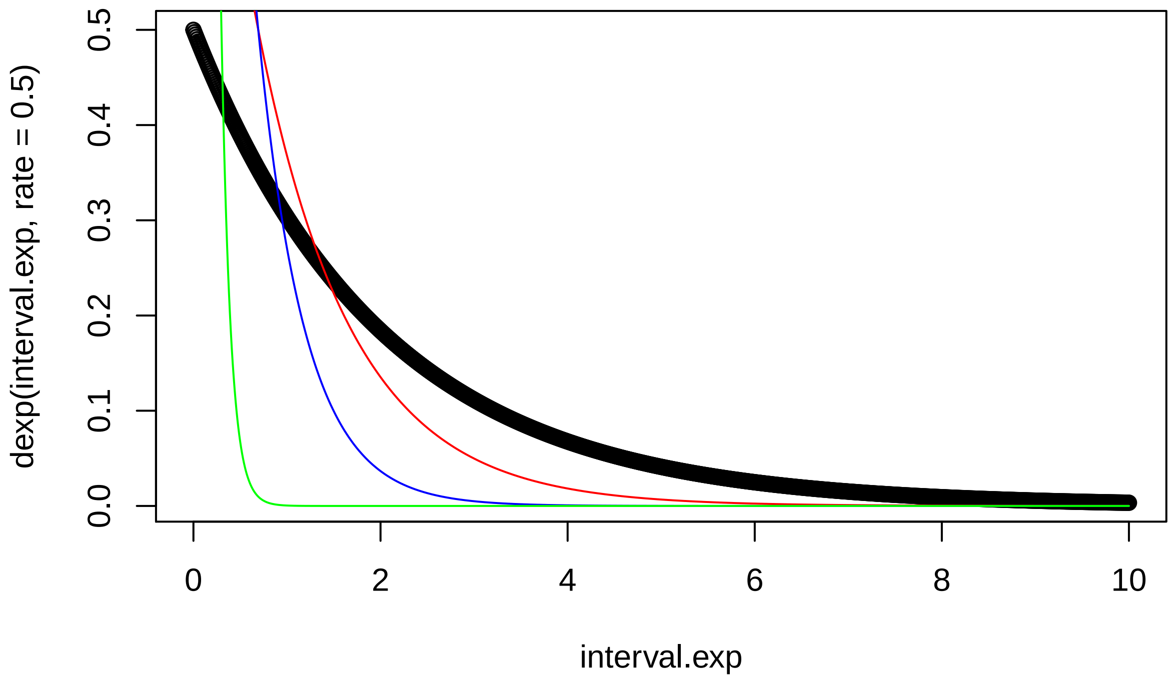
```r
hist.cum.sum$counts <- cumsum(hist.cum.sum$counts)
plot(hist.cum.sum)
curve(pnorm(x, mean=0, sd =1), add=TRUE)
```



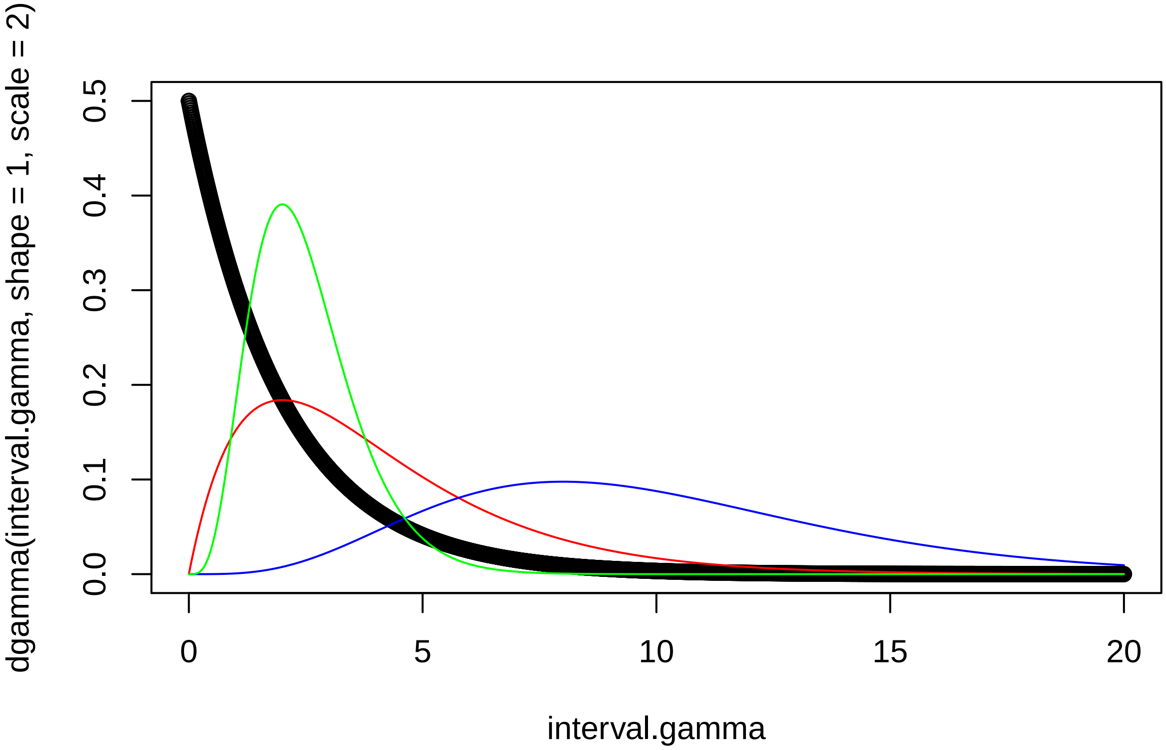
Exercise 5

Let's see other distribution, not the normal.

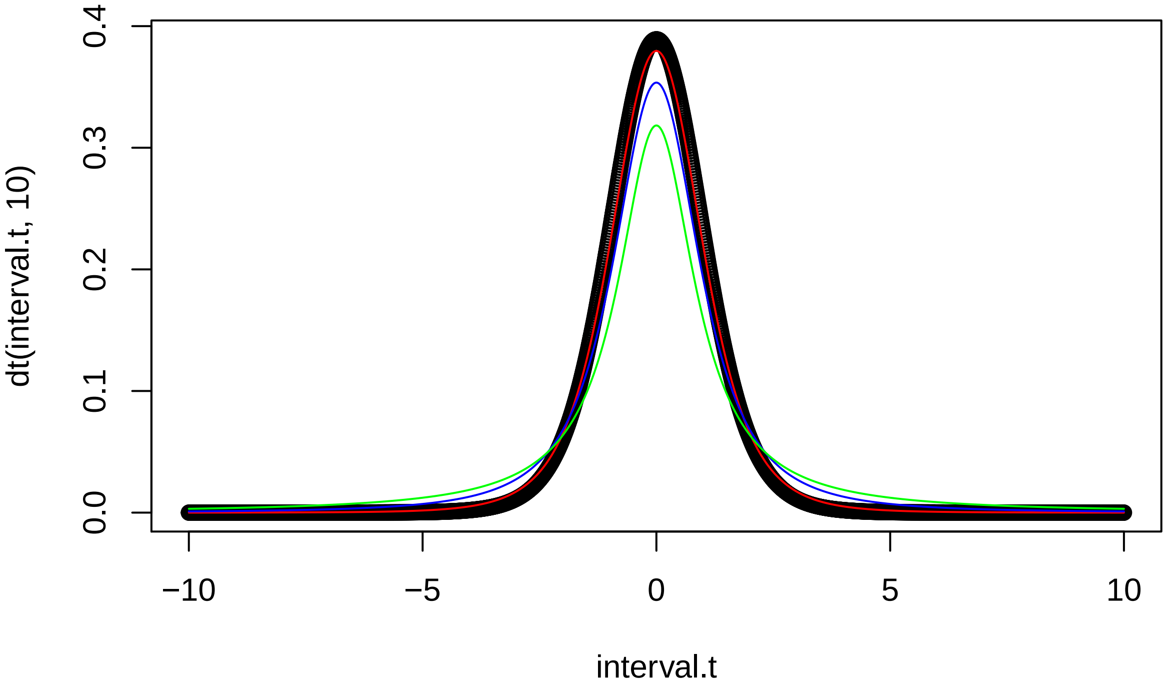
```
#Exponential

interval.exp <- seq(0,10,0.01)
plot(interval.exp, dexp(interval.exp, rate=0.5))
lines(interval.exp,dexp(interval.exp,rate=1),col="red")
lines(interval.exp,dexp(interval.exp,rate=2),col="blue")
lines(interval.exp,dexp(interval.exp,rate=10),col="green")
```



```
#Gamma
interval.gamma<-seq(0, 20, 0.01)
plot(interval.gamma,dgamma(interval.gamma,shape=1,scale=2))
lines(interval.gamma,dgamma(interval.gamma,shape=2,scale=2),col="red")
lines(interval.gamma,dgamma(interval.gamma,shape=5,scale=2),col="blue")
lines(interval.gamma,dgamma(interval.gamma,shape=5,scale=0.5),col="green")
```



```
#Student
interval.t<-seq(-10,10, 0.01)
plot(interval.t,dt(interval.t,10))
lines(interval.t,dt(interval.t,5),col="red")
lines(interval.t,dt(interval.t,2),col="blue")
lines(interval.t,dt(interval.t,1),col="green")
```


0.0
0.1
0.2
0.3
0.4
dt(interval.t, 10)
−10
−5
0
5
10
interval.t