lab4.R

sathvik

2021-03-27

```
# 171EC146
# Sathvik S Prabhu

# 1. 6.1. Distribution Functions in R, Owen-The R Guide:
#   • Learn to get the distribution (d), cdf (p), quantile (q), and generate samples of the distributio
# Try for as many distributions as you wish.
# x <- rnorm(100)
# w <- rexp(1000,rate=.1)
# dbinom(3,size=10,prob=.25) # P(X=3) for X ~ Bin(n=10, p=.25)
# dpois(0:2, lambda=4)
# pbinom(3,size=10,prob=.25)
# dnorm(12,mean=10,sd=2)
# qnorm(.75,mean=10,sd=2)
# qchisq(.10,df=8)
# qt(.95,df=20)

# • Report for any three distributions
# Dist 1: norm
dnorm(-1:1,mean=0.5,sd=1)
```

```
## [1] 0.1295176 0.3520653 0.3520653
```

```
pnorm(-1:1,mean=0.5,sd=1)
```

```
## [1] 0.0668072 0.3085375 0.6914625
```

```
qnorm(0.9985,mean=0.5,sd=1)
```

```
## [1] 3.467738
```

```
rnorm(5,mean=0.5,sd=1)
```

```
## [1]  0.8049424  0.7495055 -0.8895891  0.5196809  1.4821425
```

```
# Dist 2: exp
dexp(0.5:3,rate=0.5)
```

```
## [1] 0.3894004 0.2361833 0.1432524
```

```
pexp(0.5:3,rate=0.5)
```

```
## [1] 0.2211992 0.5276334 0.7134952
```

```
qexp(0.92,rate=0.5)
```

```
## [1] 5.051457
```

```r
rexp(4,rate=0.5)
```

```
## [1] 2.5736500 2.5695583 1.5882616 0.5828665
```

```r
# Dist 3: binom
dbinom(1:10,10,prob=0.5)
```

```
##  [1] 0.0097656250 0.0439453125 0.1171875000 0.2050781250 0.2460937500
##  [6] 0.2050781250 0.1171875000 0.0439453125 0.0097656250 0.0009765625
```

```r
pbinom(1:10,10,prob=0.5)
```

```
##  [1] 0.01074219 0.05468750 0.17187500 0.37695313 0.62304687 0.82812500
##  [7] 0.94531250 0.98925781 0.99902344 1.00000000
```

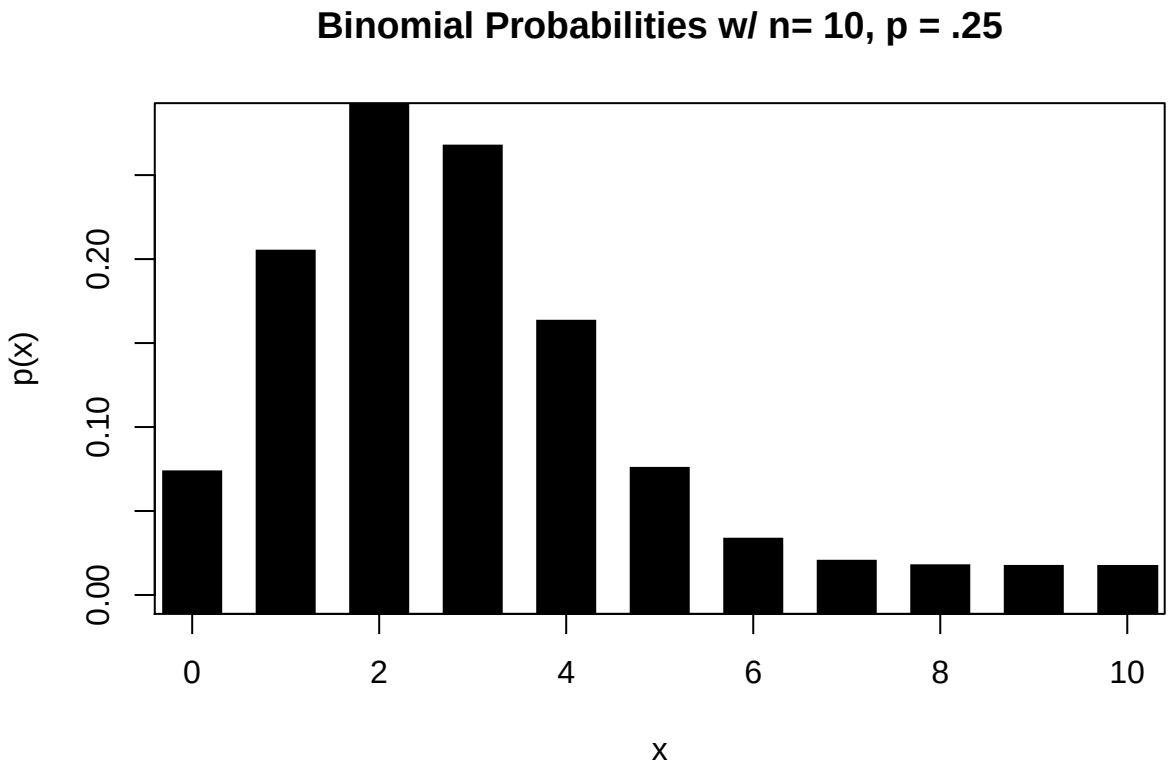
```r
qbinom(0.5,10,prob=0.5)
```

```
## [1] 5
```

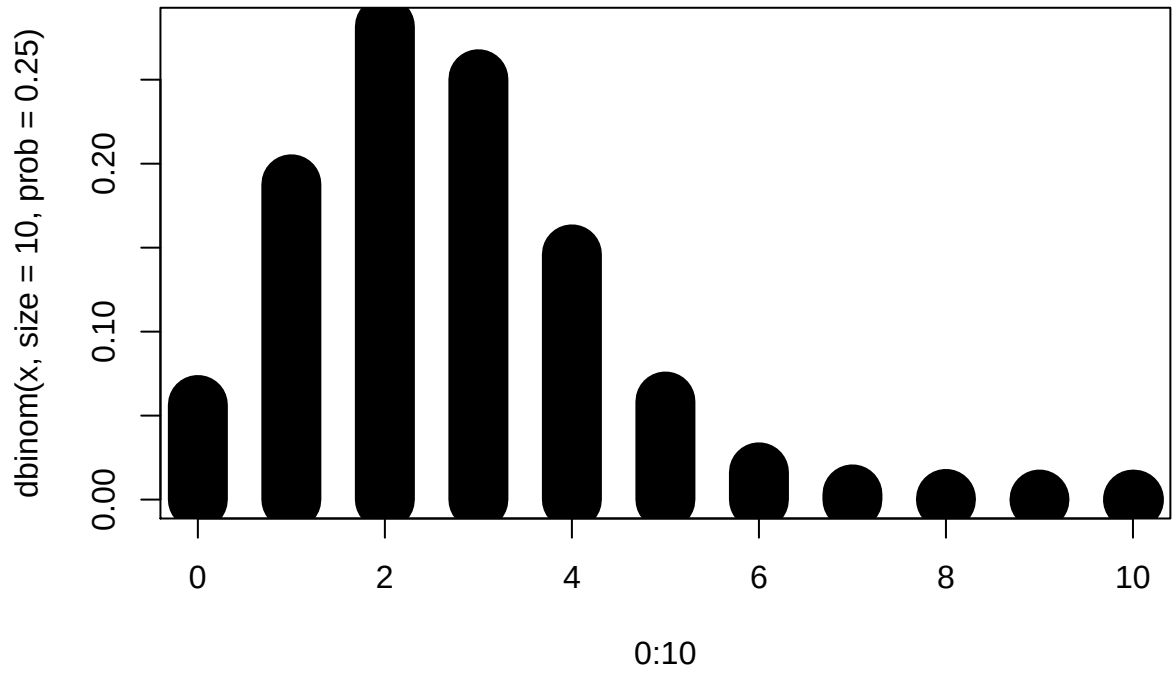
```r
rbinom(2,10,prob=0.5)
```

```
## [1] 7 5
```

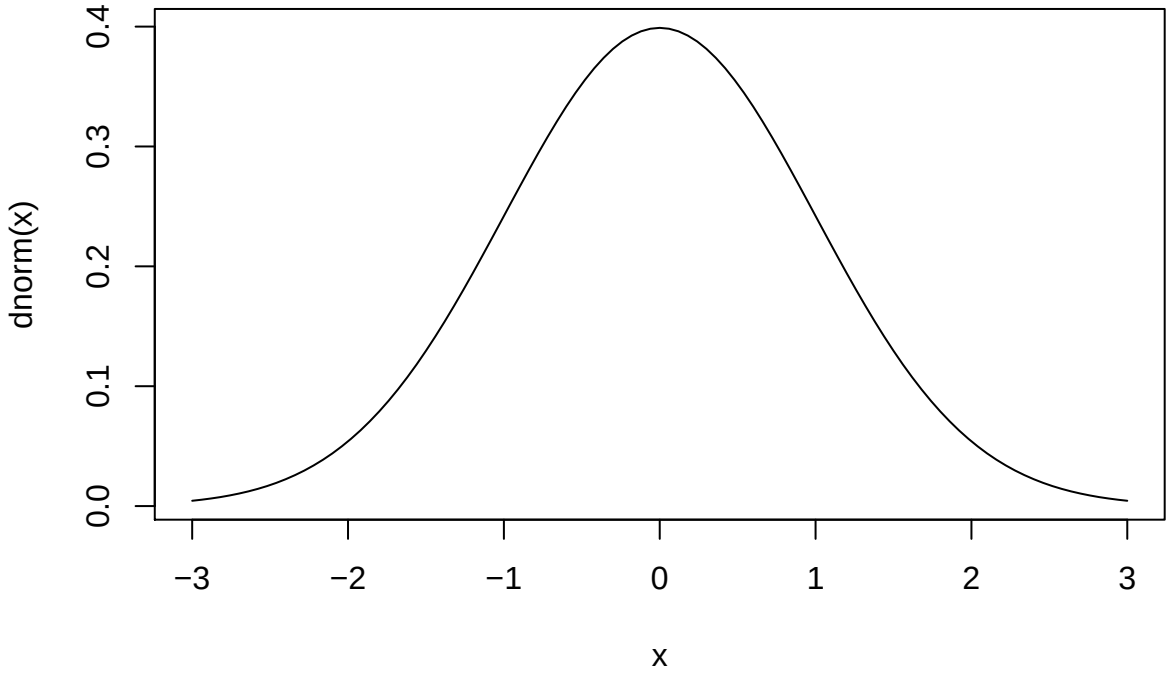
```r
# 2. 6.3 Graphing Distributions
# Discrete
x <- 0:10
y <- dbinom(x, size=10, prob=.25)
plot(x, y, type = "h", lwd = 30, main = "Binomial Probabilities w/ n= 10, p = .25", ylab = "p(x)", lend
```



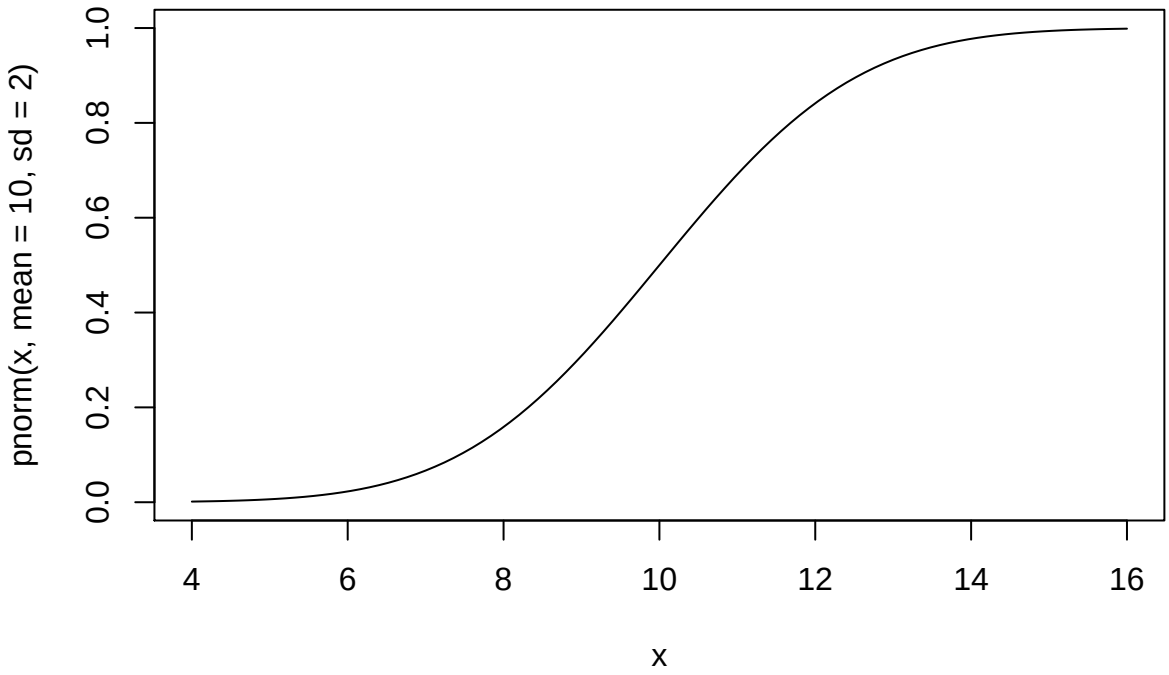

```r
plot(0:10, dbinom(x, size=10, prob=.25), type = "h", lwd = 30) # without other embellishments
```




```
# Continuous
curve(dnorm(x), from = -3, to = 3) # normal pdf
```




```
curve(pnorm(x, mean=10, sd=2), from = 4, to = 16) # normal cdf
```



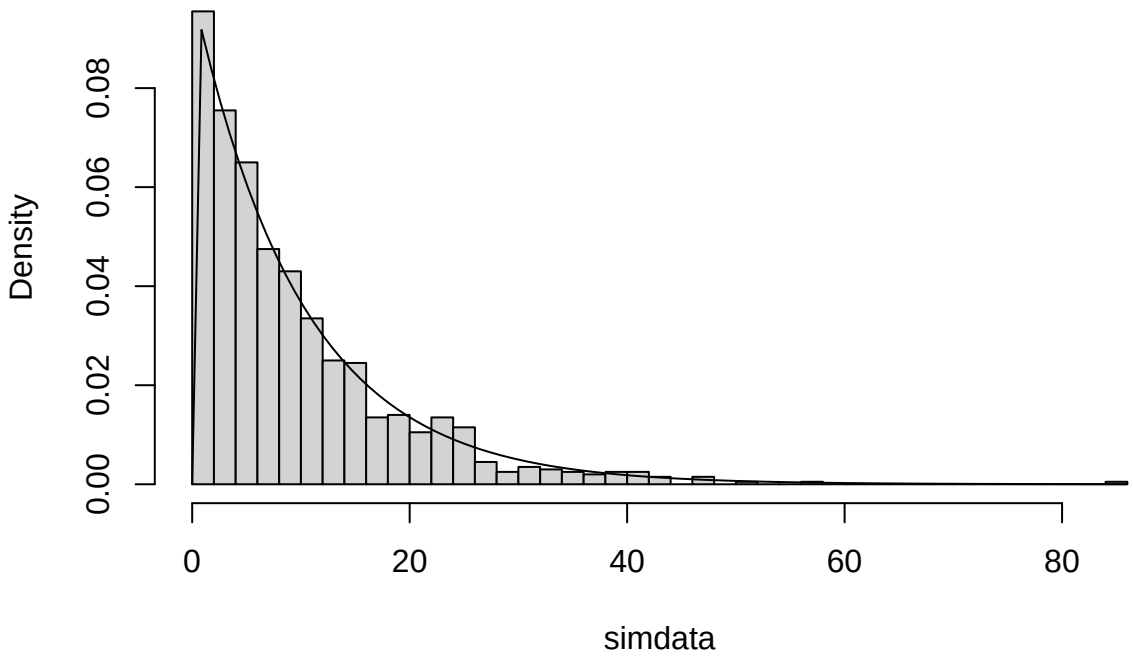

```
qnorm(0.9985) #68 95 99.7 rule
```

```
## [1] 2.967738
```

```
qnorm(0.975)
```

```
## [1] 1.959964
```

```
simdata <- rexp(1000, rate=.1)
hist(simdata, prob = T, breaks = "FD", main="Exp(theta = 10) RVs")
curve(dexp(x, rate=.1), add = T)
```




```
# 3. 6.4 Random Sampling
# sample(x, size, replace = FALSE, prob = NULL)
sample(1:100, 1)
```

```
## [1] 13
```

```
sample(1:6, 10, replace = T) # fair die
```

```
##  [1] 4 3 5 3 4 5 4 4 2 4
```

```
sample(1:6, 10, T, c(.6,.2,.1,.05,.03,.02)) # biased die
```

```
##  [1] 1 2 2 1 1 1 2 1 1 1
```

```
urn <- c(rep("red",8),rep("blue",4),rep("yellow",3))
sample(urn, 6, replace = F)
```

```
## [1] "red"    "red"    "red"    "red"    "blue"   "yellow"
```

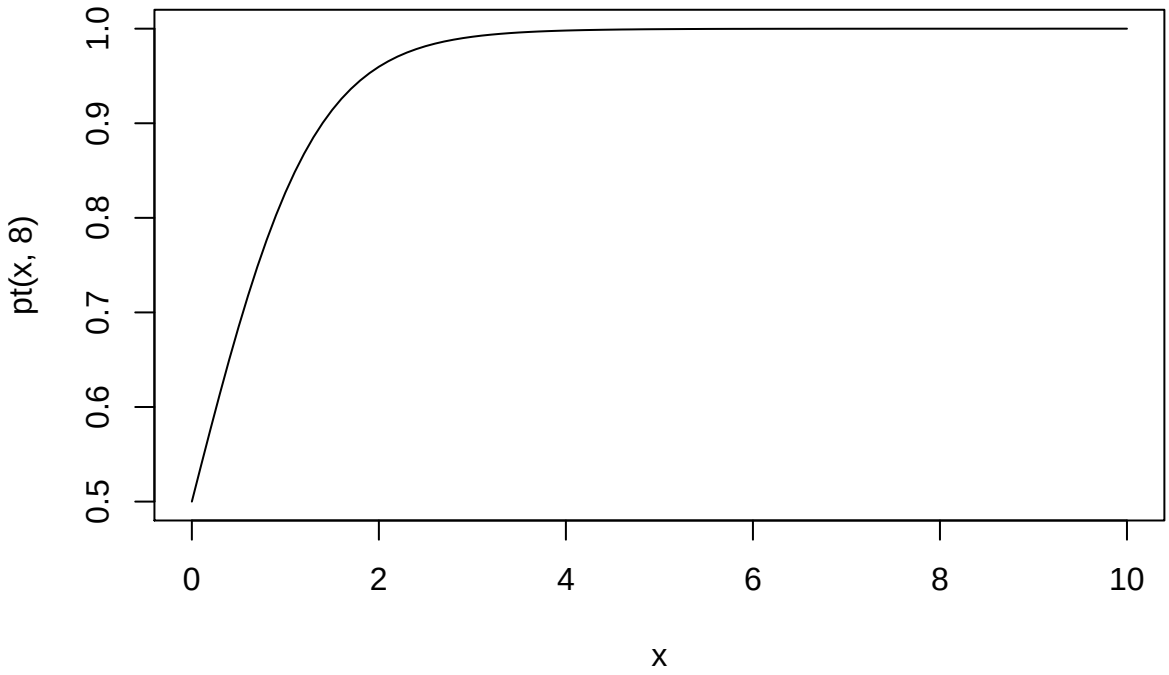
```
# 4. 6.5 Exercises covered here (6.2, 6.4, and 6.5).
#1
simdata<-rbinom(20,15,0.2)
simdata
```

```
##  [1] 5 3 4 4 3 4 3 3 4 1 3 4 4 2 3 3 0 4 3 3
```

```
#2
qgamma(0.2,shape=2,scale=10)
```

```
## [1] 8.243883
```

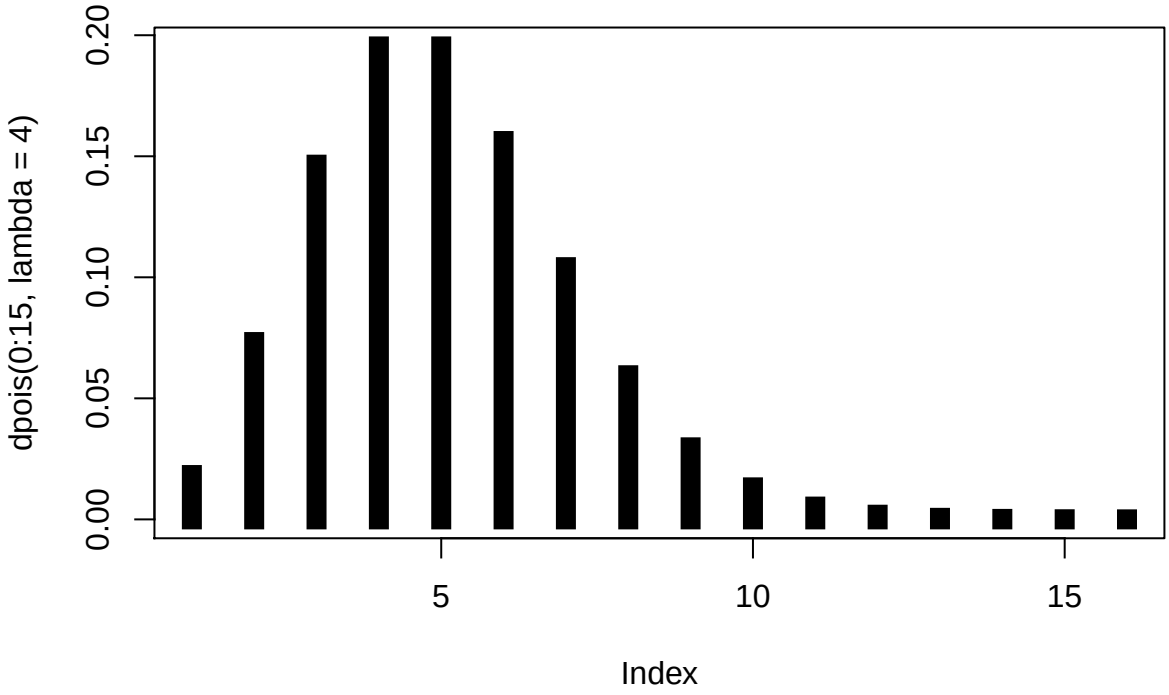
```
#3
curve(pt(x,8),from = 0,to=10)
```




```r
1-pt(2,8)
```

```
## [1] 0.04025812
```

```r
#4
plot(dpois(0:15,lambda=4),type="h",lwd=10,lend="square")
```



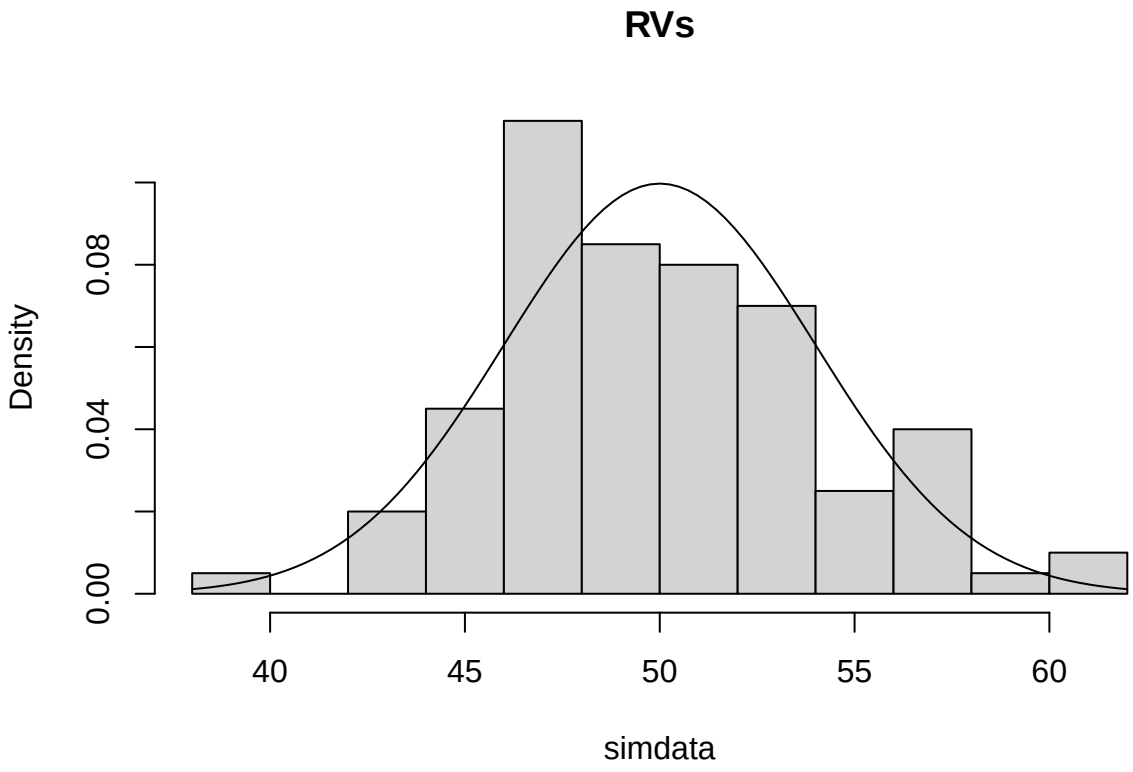

```r
#5
x <- runif(1000000, min=0, max=2)
a=2*mean(exp(x^3))
a
```

```
## [1] 277.5068
```

```
b=integrate(f = function(x) exp(x^3), lower = 0, upper = 2)
b$value
```

```
## [1] 276.8529
```

```
#6
simdata<-rnorm(100,mean=50,sd=4)
hist(simdata, prob = T, breaks = "FD", main="RVs")
curve(dnorm(x,mean=50,sd=4), add = T)
```




```
#7
fair_coin <- c("heads","tails")
sample(fair_coin,25,replace=T)
```

```
##  [1] "tails" "tails" "tails" "tails" "heads" "heads" "tails" "heads" "tails"
## [10] "heads" "tails" "heads" "tails" "tails" "heads" "heads" "tails" "heads"
## [19] "tails" "heads" "tails" "heads" "tails" "tails" "tails"
```