Homework 5

Alexander Ollerton

10/4/2019

1.a)

```r
sapply(SFH,class)
```

```
     cities     housing
"character" "character"
```

```r
sapply(cities, class)
```

```
  longitude    latitude      county medianPrice  medianSize   numHouses
    "array"     "array"    "factor"     "array"     "array"     "array"
   medianBR
    "array"
```

```r
sapply(housing, class)
```

```
$county
[1] "factor"

$city
[1] "factor"

$zip
[1] "factor"

$street
[1] "character"

$price
[1] "numeric"

$br
[1] "integer"

$lsqft
[1] "numeric"

$bsqft
[1] "integer"

$year
[1] "integer"

$date
[1] "POSIXt"  "POSIXct"

$long
[1] "numeric"
```

```
$lat
[1] "numeric"

$quality
[1] "factor"

$match
[1] "factor"

$wk
[1] "Date"
```

1.b)

```r
sapply(housing, function(x) sum(is.na(x)))
```

```
 county    city     zip  street   price      br   lsqft   bsqft    year
      0       0       5       0       0       0   21687     426    9202
   date    long     lat quality   match      wk
      0   23316   23316   23316   23316       0
```

1.c)

```r
tapply(housing$price, housing$county, median)
```

```
      Alameda County  Contra Costa County         Marin County
              510000               466000               739000
         Napa County San Francisco County     San Mateo County
              505000               702000               700000
  Santa Clara County        Solano County        Sonoma County
              582000               380000               476500
```

1.d)

```r
head(sort(tapply(housing$price, housing$city, mean),decreasing = T), n=10)
```

```
  Los Altos Hills          Atherton      Hillsborough Belvedere/Tiburon
          2393311           2379174           2354199           2217681
        Belvedere              Ross            Diablo Belvedere/tiburon
          2170088           2135883           1973025           1776572
     Monte Sereno     Stinson Beach
          1656639           1640469
```

1.e) When looking at the data set it looks as though there was some potential duplication in the data that was returned, however, this is not the case. It looks like they could be representing separate areas or the person who did the data imput may have not noticed their mistake, but there is a Belvedere, Belvedere/Tiburon and a Belvedere/tiburon.

1.f)

```r
czip <- as.character(housing$zip)
izip <- as.integer(czip)
SFZip <- (izip >= 94102) & (izip <= 94134)
inSanFran <- subset(housing,SFZip == T)
nrow(inSanFran)
```

```
[1] 8134
```

The number of zipcodes that fall into SanFrancisco is 8134

1.g)

```
mean(inSanFran$br)
```

```
## [1] 2.36956
```

```
outSanFran <- subset(housing, SFZip==F)
mean(outSanFran$br)
```

```
## [1] 3.043085
```

The average number of bedrooms in SanFran is 2.37 and the average number outside of SanFran is 3.04. This means that there are more oppprtunities to find a living situation outside of San Francisco where you have more bedrooms. Also, comparing this with other data from the previous questions we can see that the living situation in SanFrancisco is round 700,000 dollars compared to other places like Napa County of 500,000 dollars so you could live outside of San Francisco for cheapter and you can get more for your dollars spent.

2.a)

```
myFactorial1 <- function(n){factorial(n)}
myFactorial2 <- function(n){
  if(n==0){
    return(1)
  }
  else
    prod(1:n)}
myFactorial3 <- function(n) {
  if(n==0){
    return(1)
  }
  else {
  y <- 1
  for(i in 1:n){
y <-y*((1:n)[i])
print(y)
}
}}
myFactorial4 <- function(n){
   if(n==0){
    return(1)
  }
  else {
  f <- 1
  while (n > 0){
      f = f * n
      n = n - 1}
  print(f)
}}
myFactorial5 <-function(n) {
if (n == 0) {
return(1)
} else {
return(n * myFactorial5(n - 1))
}
}
```

```r
myFactorial6 <- function(n) {
if (n == 0) {
return(1)
} else {
return(myFactorial6(n - 1)*n)
}
}
```

2.b)

```r
mbench
```

```
Unit: nanoseconds
            expr    min         lq         mean     median         uq
 myFactorial1(n)    423      790.5     5290.890     2533.5     3848.0
 myFactorial2(n)    595     1098.5     6450.206     2020.5     4853.5
 myFactorial3(n) 655884 11298676.5 22777755.760 23321188.5 33242219.5
 myFactorial4(n)  14921   103960.5   212649.680   177799.0   260893.5
 myFactorial5(n)  26854    29192.5    43988.408    32766.0    45225.0
 myFactorial6(n)  27528    29714.0    44439.004    37401.5    44506.0
      max neval
  1341834   500
  1661288   500
 89868711   500
 11880346   500
  2252300   500
  1945922   500
```

2.c)

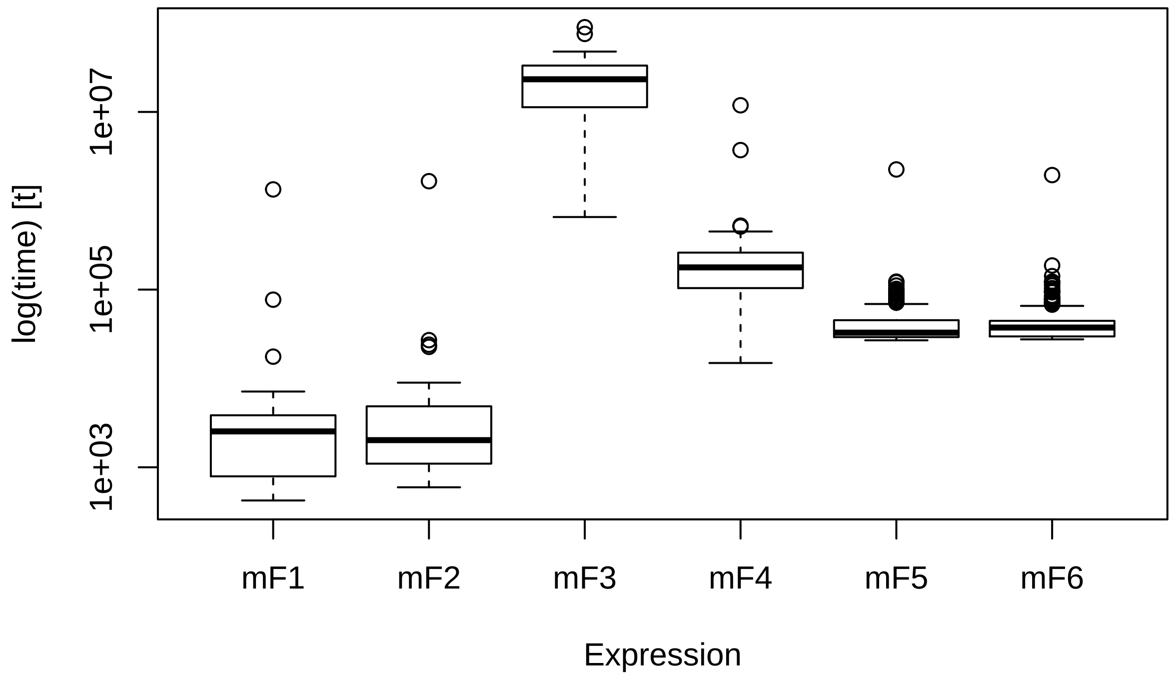
```r
class(mbench)
```

```
[1] "microbenchmark" "data.frame"
```

The class of mbench is a “microbenchmark” “data.frame” and we have seen the data.frame class before, but not the benchmark.

2.d)

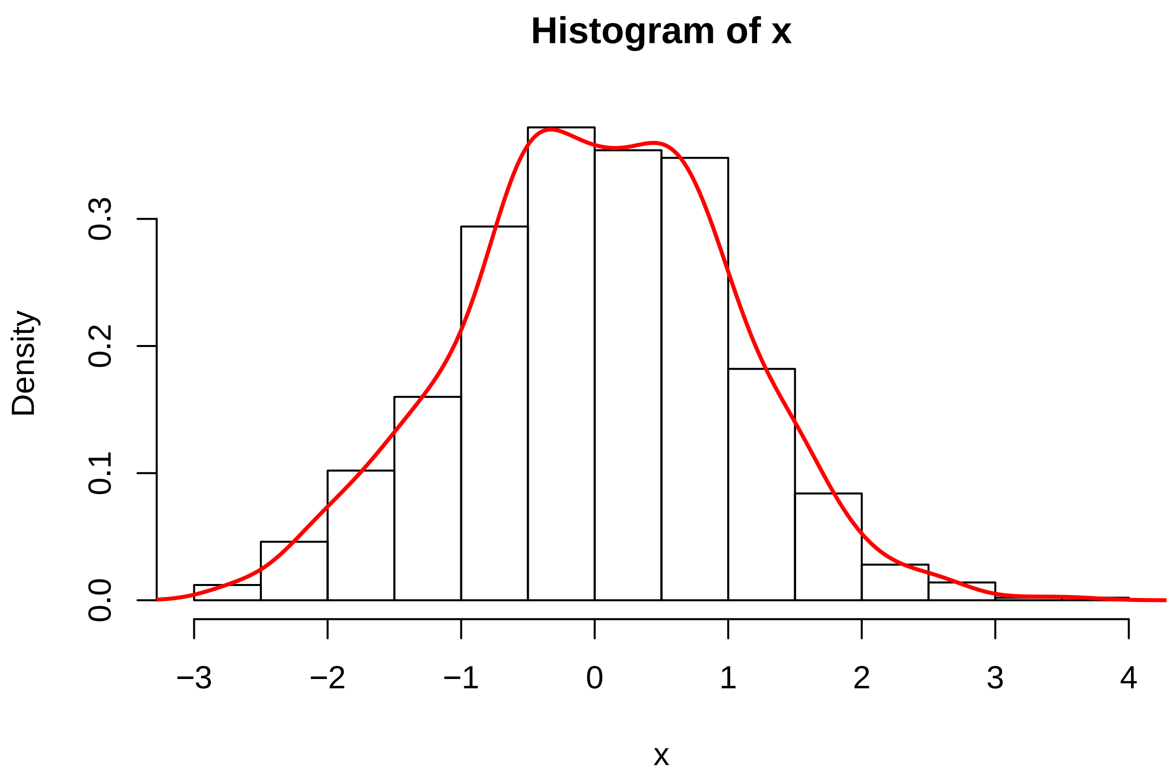
```r
boxplot(mbench, names = c('mF1','mF2','mF3', 'mF4','mF5','mF6'))
```



2.e) Looking at the medians for these functions, the fastest function for this exercise is myFactorial2 (vecorization). The slowest of these is myFactorial3. The functions that are about the same are myFactorial1 and myFactorial2. The other functions that are similar are myFactorial5 and myFactorial6.

3.a)

```
x <- rnorm(1000)
hist(x, prob=TRUE)
lines(density(x), col="red", lwd=2)
```



The lwd function within this code changes the thickness of the line.

3.b)

```
class(density(x))
```

```
[1] "density"
```

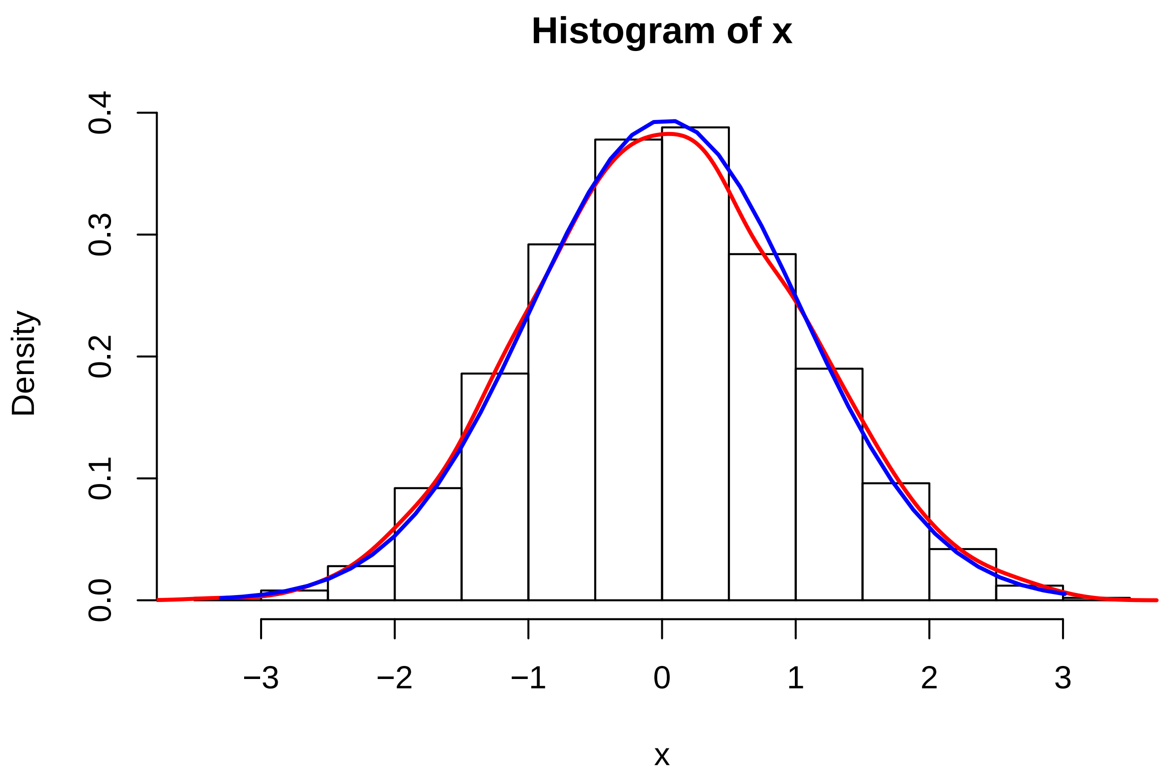
The class of density(x) is 'density'.

3.c)

The argument x is a double vector and this is a vector of numbers coming from the rnorm function and a vector of numbers from 1:1000.

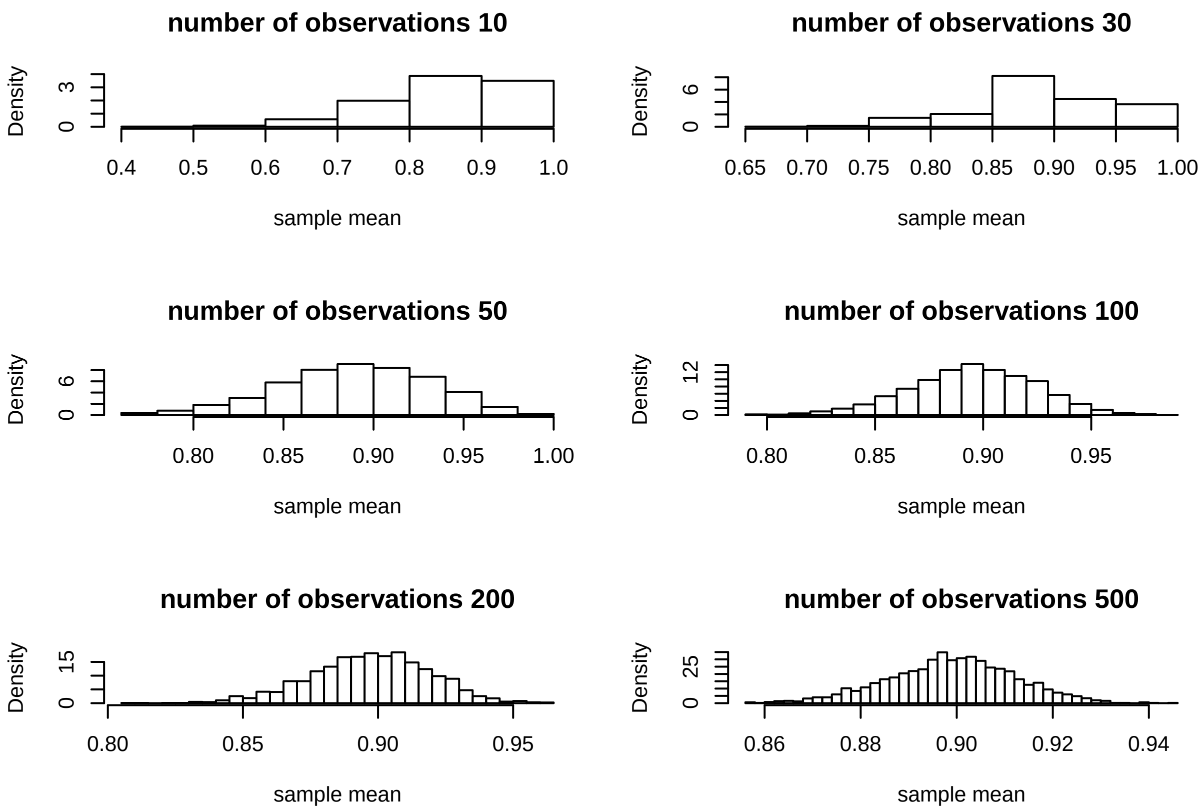
3.d)

```
x <- rnorm(1000)
hist(x, prob=TRUE)
lines(density(x), col="red", lwd=2)
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
lines(xfit, yfit, col="blue", lwd=2)
```

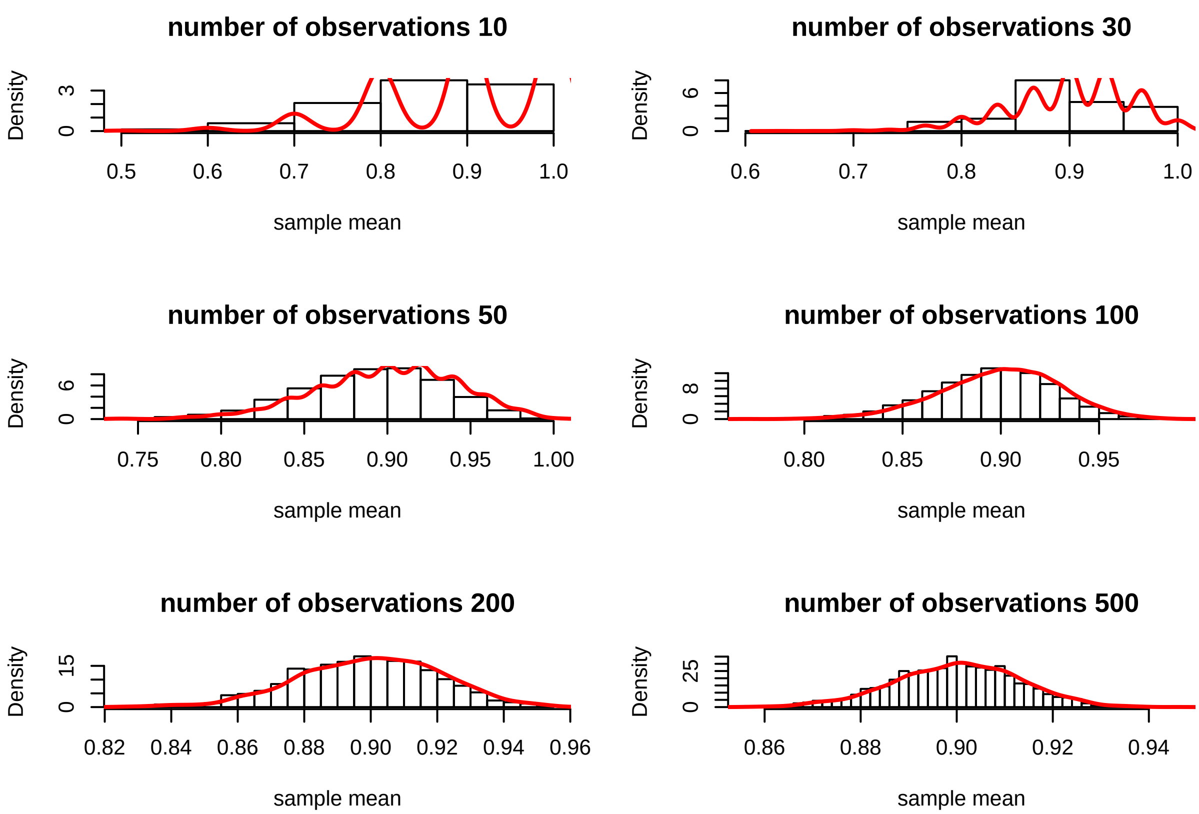


3.e)

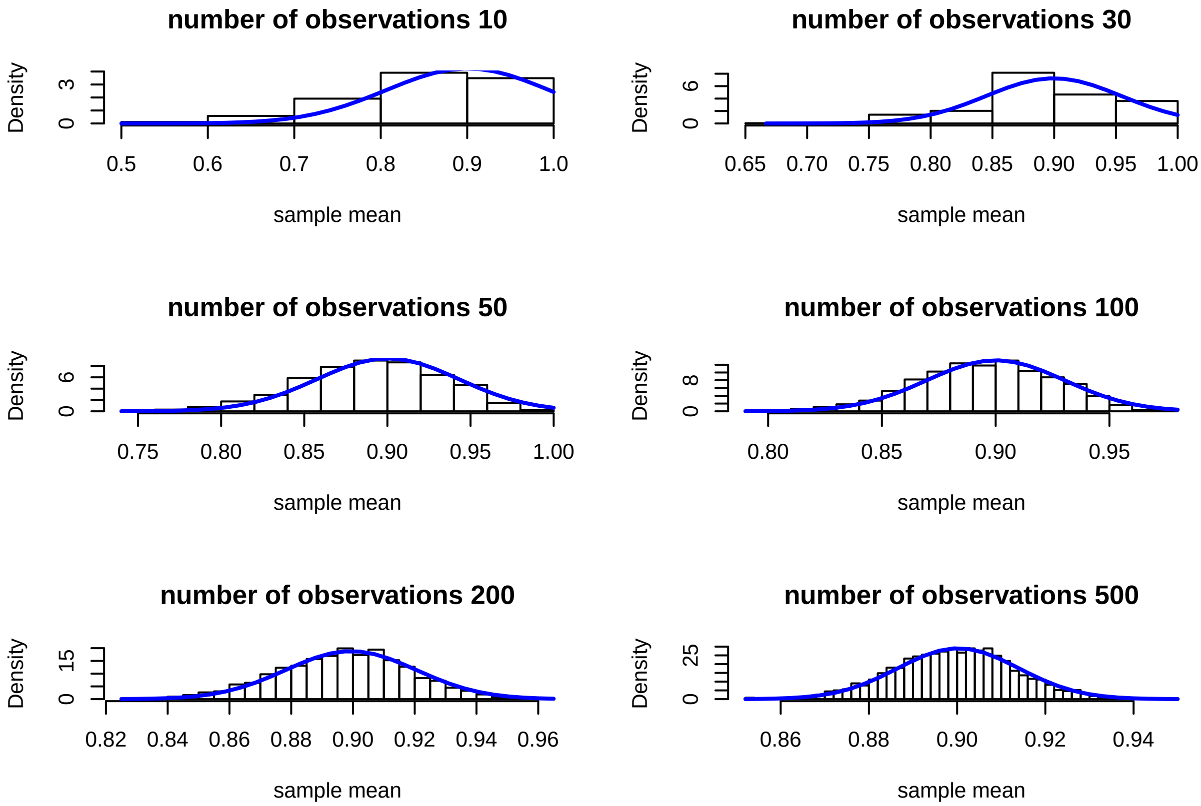
```
centralLimit <- function(nreps = 2500, nvec= c(10,30,50,100,200,500), bvec= c(5,10,15,25,30,40), ps = 0
par(mfrow = c(3,2))
for(i in 1:length(nvec)){
  mySamples <- lapply(rep(nvec[i],nreps), rbinom, size=1, ps)
  myMeans <- sapply(mySamples, mean)
   hist(myMeans, breaks = bvec[i] , probability = T, main = paste("number of observations",nvec[i]),xlab

if(density) {
  lines(density(myMeans), col = "red", lwd=2) }

if(normal){
  xfit<-seq(min(myMeans),max(myMeans),length=30)
  yfit<-dnorm(xfit,mean=mean(myMeans),sd=sd(myMeans))
  lines(xfit, yfit, col= "blue", lwd=2) }

}

}
centralLimit()
```



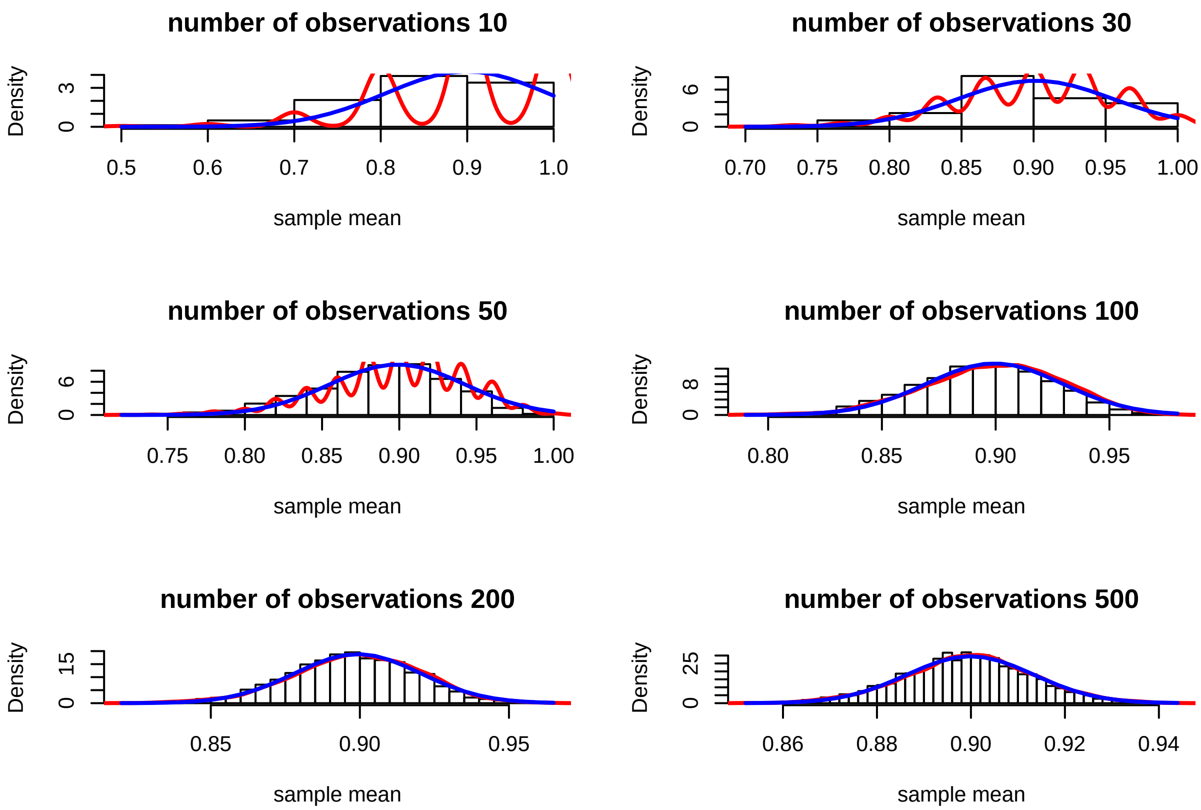

```
centralLimit(density = T)
```



```r
centralLimit(normal = T)
```




```r
centralLimit(density = T,normal = T)
```


number of observations 10
Density
0
3
0.5
0.6
0.7
0.8
0.9
1.0
sample mean
number of observations 30
Density
0
6
0.70
0.75
0.80
0.85
0.90
0.95
1.00
sample mean
number of observations 50
Density
0
6
0.75
0.80
0.85
0.90
0.95
1.00
sample mean
number of observations 100
Density
0
8
0.80
0.85
0.90
0.95
sample mean
number of observations 200
Density
0
15
0.85
0.90
0.95
sample mean
number of observations 500
Density
0
25
0.86
0.88
0.90
0.92
0.94
sample mean