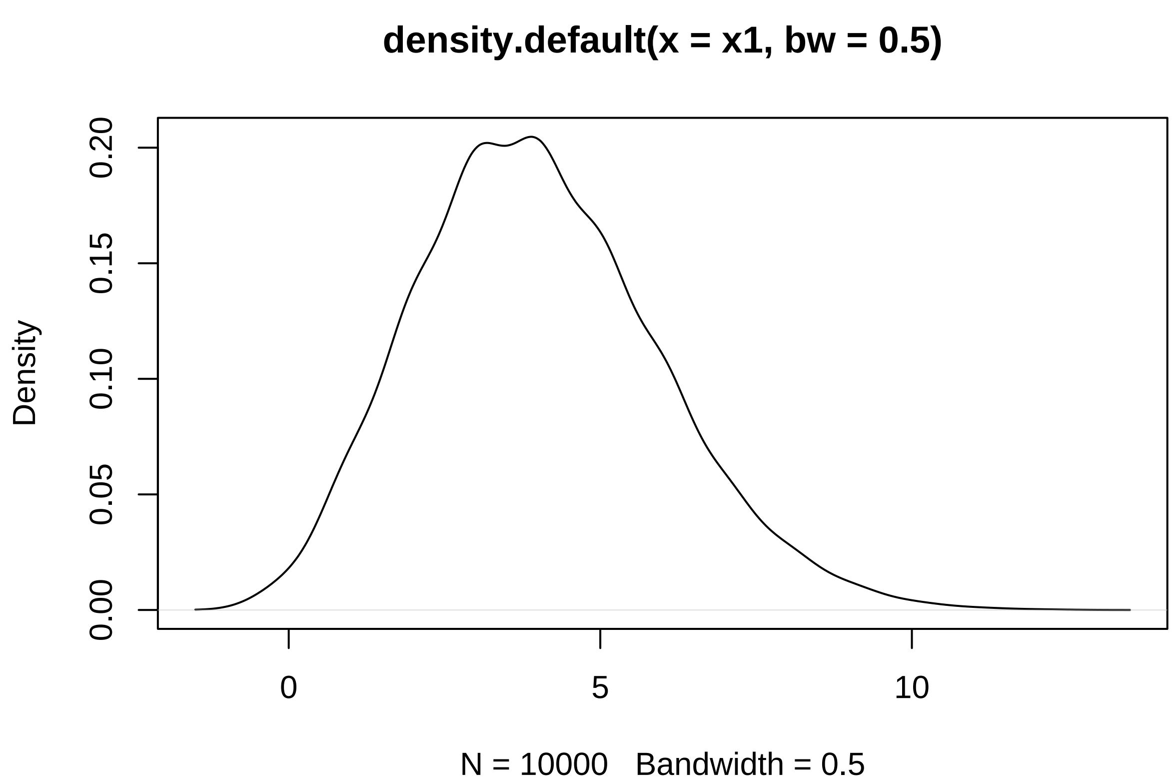
WLLN and CLT

Arturo Valdivia

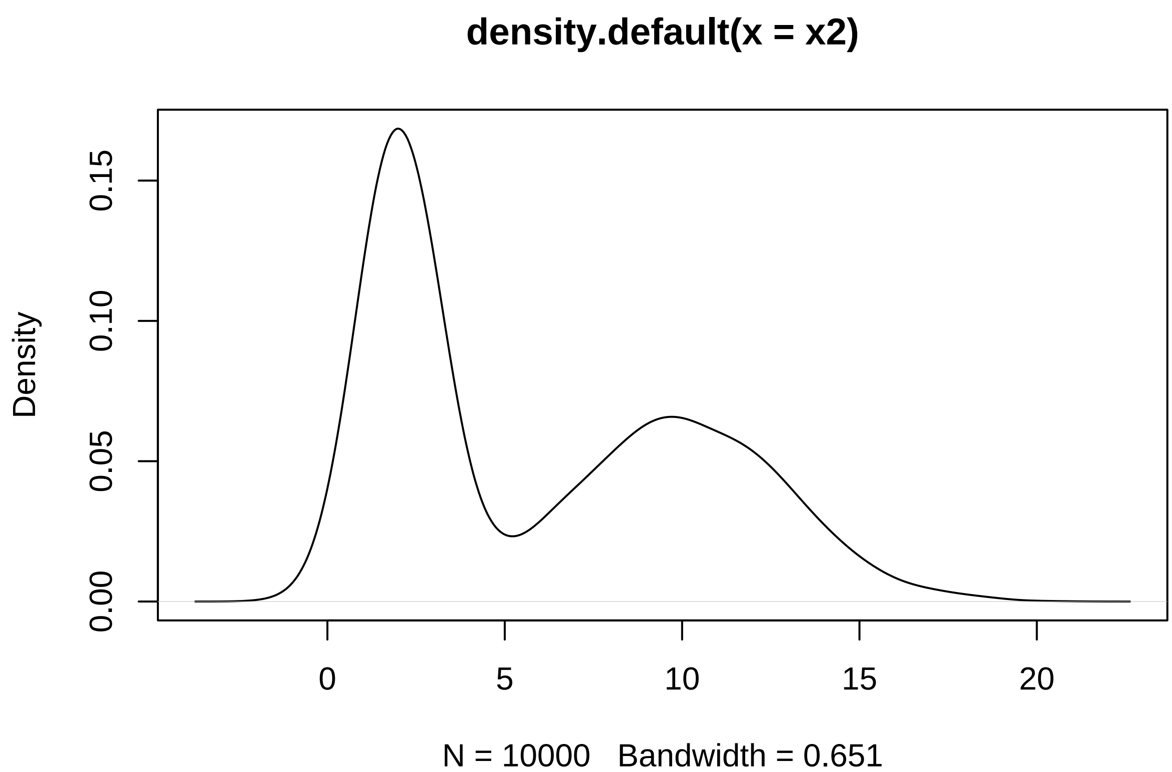
2/21/2023

Let's first create some vectors x1, x2, and x3 Well assume these vectors are the random variables we'll use later on

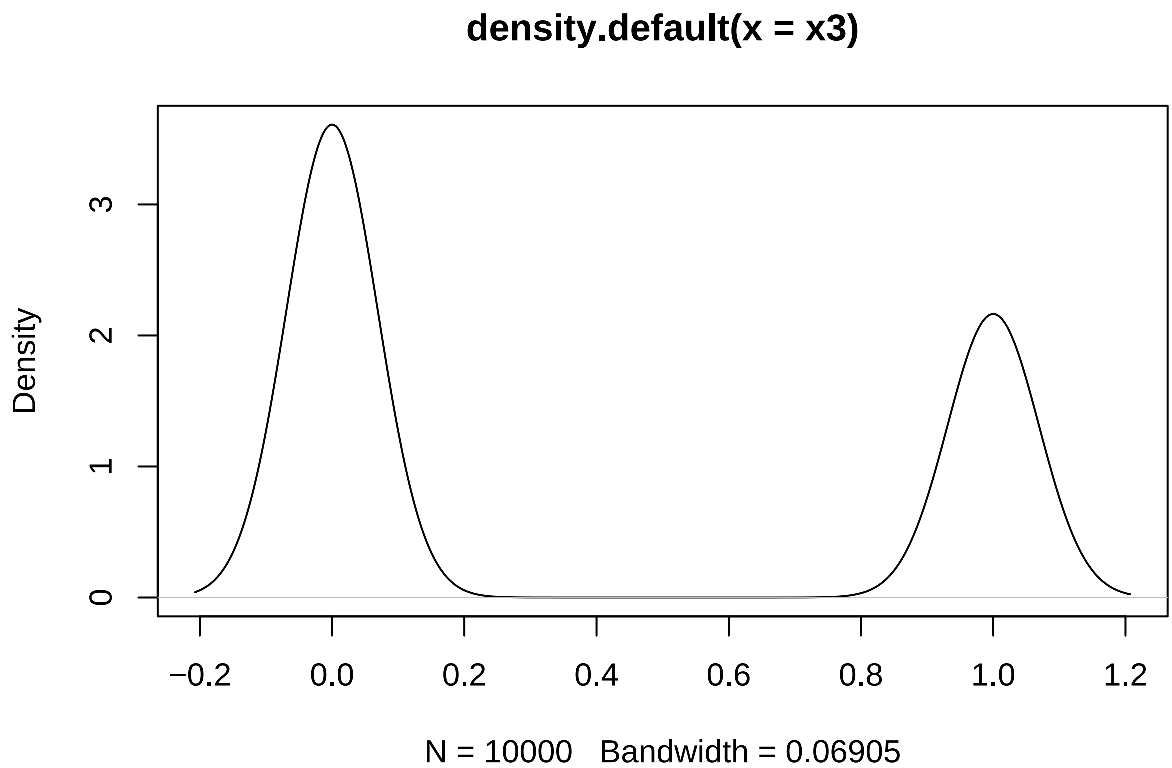
```
N = 10^4
set.seed(101)
x1 = rbinom(N, 40, .1)
x2 = c(rnorm(N/2, 10, 3), rnorm(N/2, 2, 1))
red = as.numeric(moderndive::bowl$color == "red")
x3 = sample(red, N, replace = T)
x4 = c(rep(0,1000),1)

plot(density(x1, bw = .5))
```



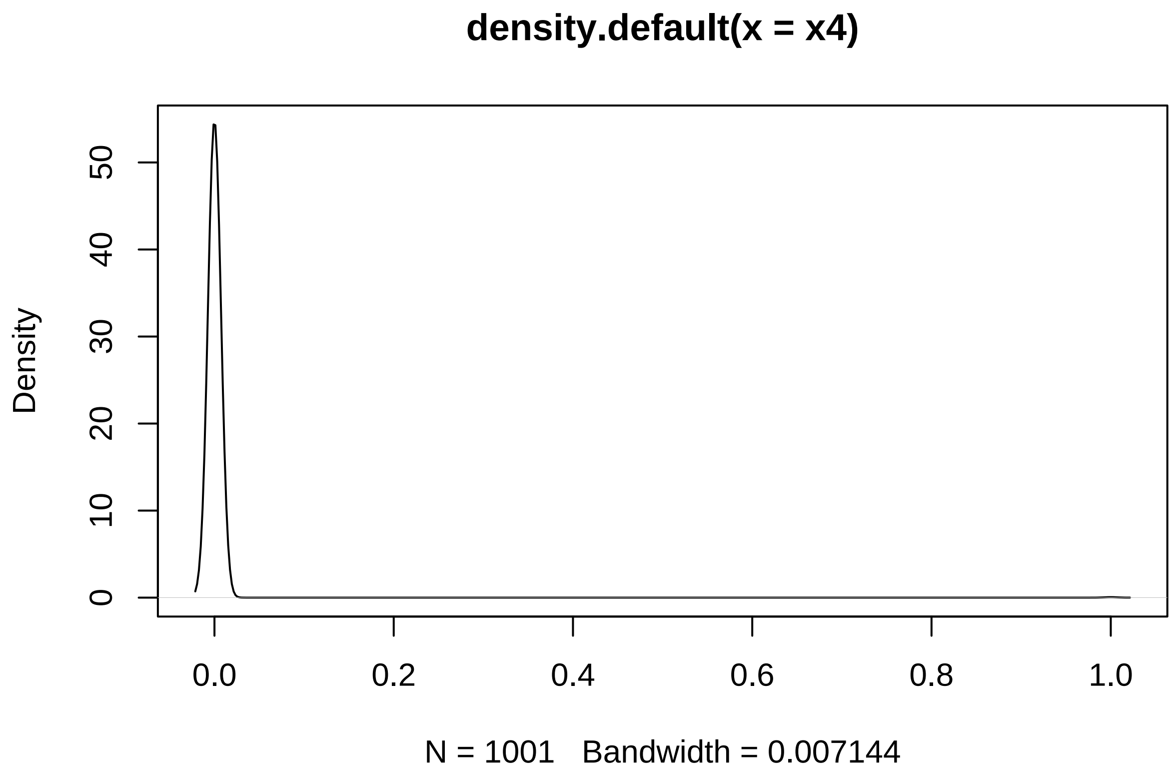
```
plot(density(x2))
```




```
plot(density(x3))
```




```
plot(density(x4))
```



WLLN

Now, let's create a function to illustrate the WLLN (check 2/21/23 slides) The key component is getting many replications of the sample mean we then obtain the probability of the sample mean to be in the interval mu - epsilon, mu + epsilon

```
wlln = function (x, n, N = 10^4, epsilon = 0.1){
  xbar.vec = replicate(N, mean(sample(x, n, replace = T)))
  lb = mean(x) - epsilon # lower bound
  ub = mean(x) + epsilon
  prob = mean(xbar.vec >= lb & xbar.vec <= ub)
  print(data.frame(n = n, probability = round(prob,2)))
}
```

Now we can use our function to check the law of large numbers You can change the 4 arguments as needed; observe that the key argument is 'n' as the large n is the higher the probability that the sample mean is in the small interval

```
wlln(x = x1, n = 10)
```

```
##    n probability
## 1 10        0.13
```

```r
wlln(x = x1, n = 10^2)
```

```
##     n probability
## 1 100         0.4
```

```r
wlln(x = x1, n = 10^3)
```

```
##      n probability
## 1 1000         0.9
```

```r
wlln(x = x1, n = 10^4)
```

```
##       n probability
## 1 10000           1
```

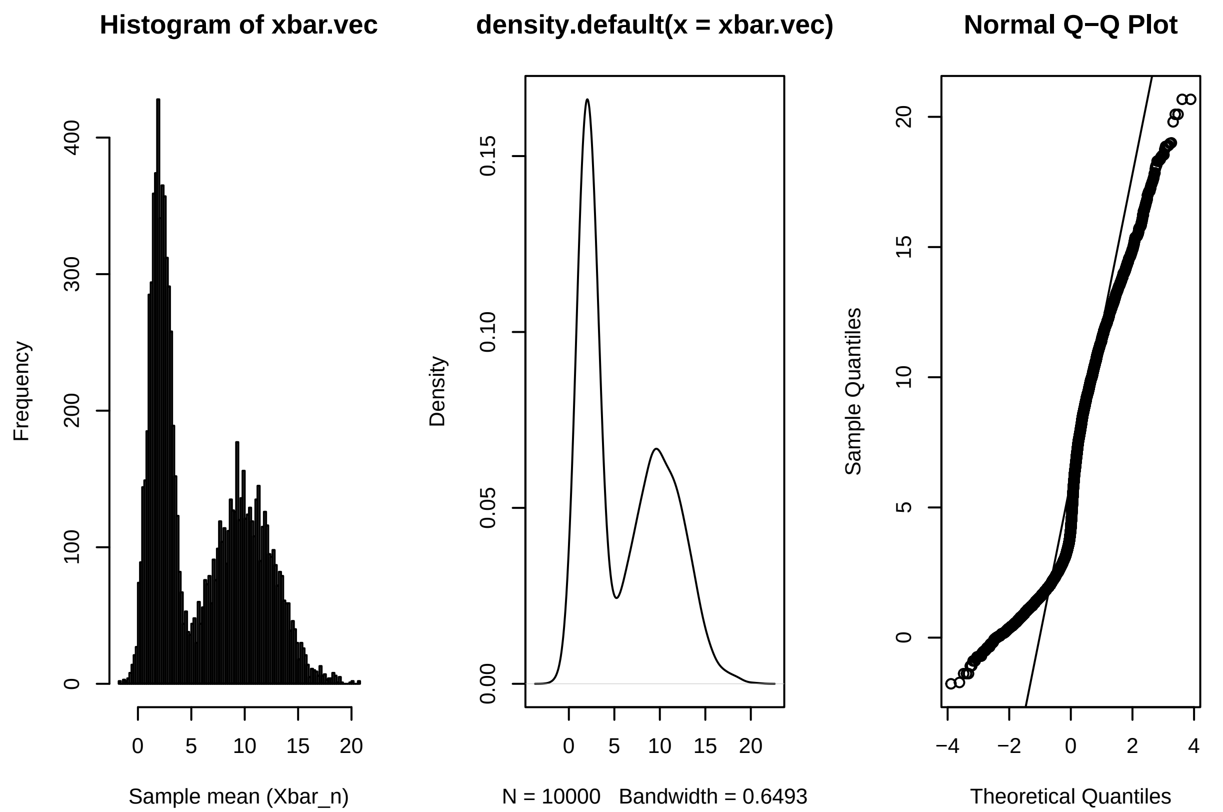
The Central Limit Theorem (CLT)

Similar to our previous function, the key component is the object xbar.vec In addition, the function finds mu, mu__Xbar, sigma, sigma__Xbar, and a histogram

```r
clt = function(x, n, N = 10^4){
  xbar.vec = replicate(N, mean(sample(x, n, replace = T)))
  op = par(mfrow = c(1,3))
  hist(xbar.vec, breaks = 100,
       xlim = c(min(x), max(x)),
       xlab = paste("Sample mean (Xbar_n)"))
  plot(density(xbar.vec))
  qqnorm(xbar.vec);qqline(xbar.vec)
  par(op)
}
```

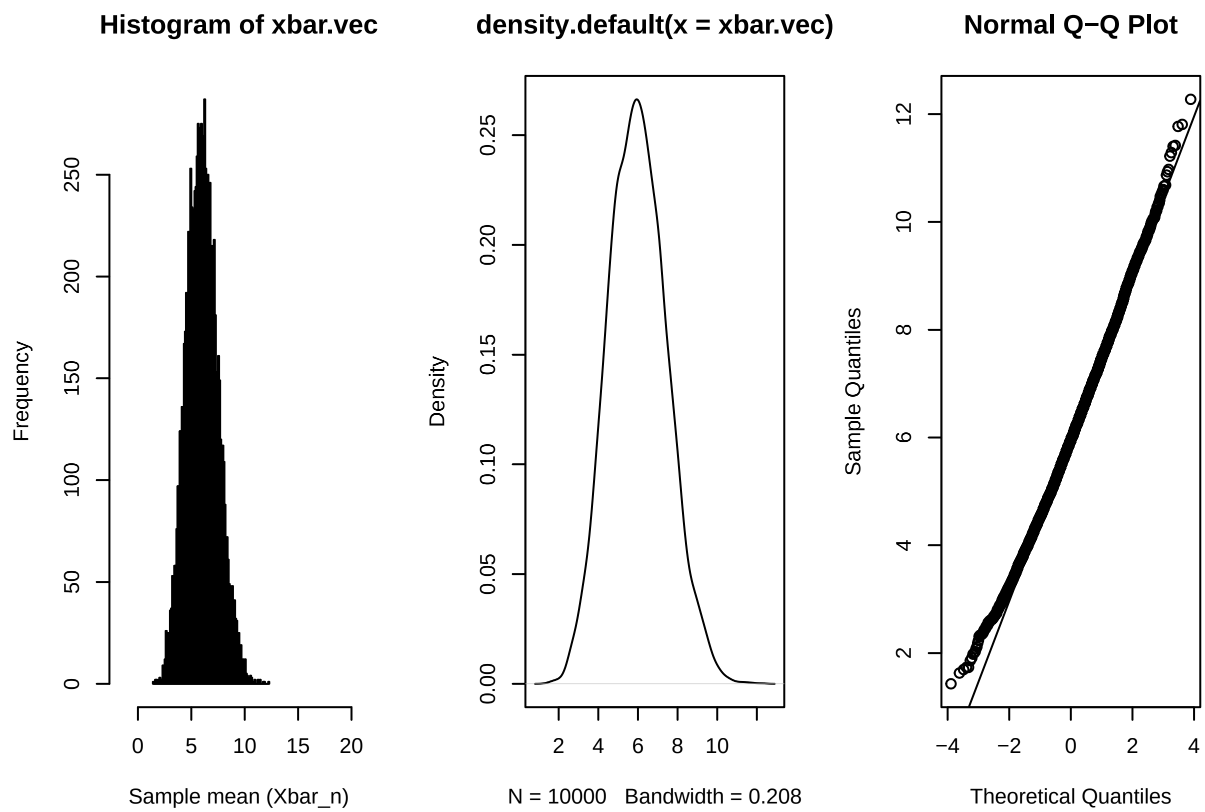
Again, change the argument values, in particular change n to see how the

```r
clt(x = x2, n = 1)
```

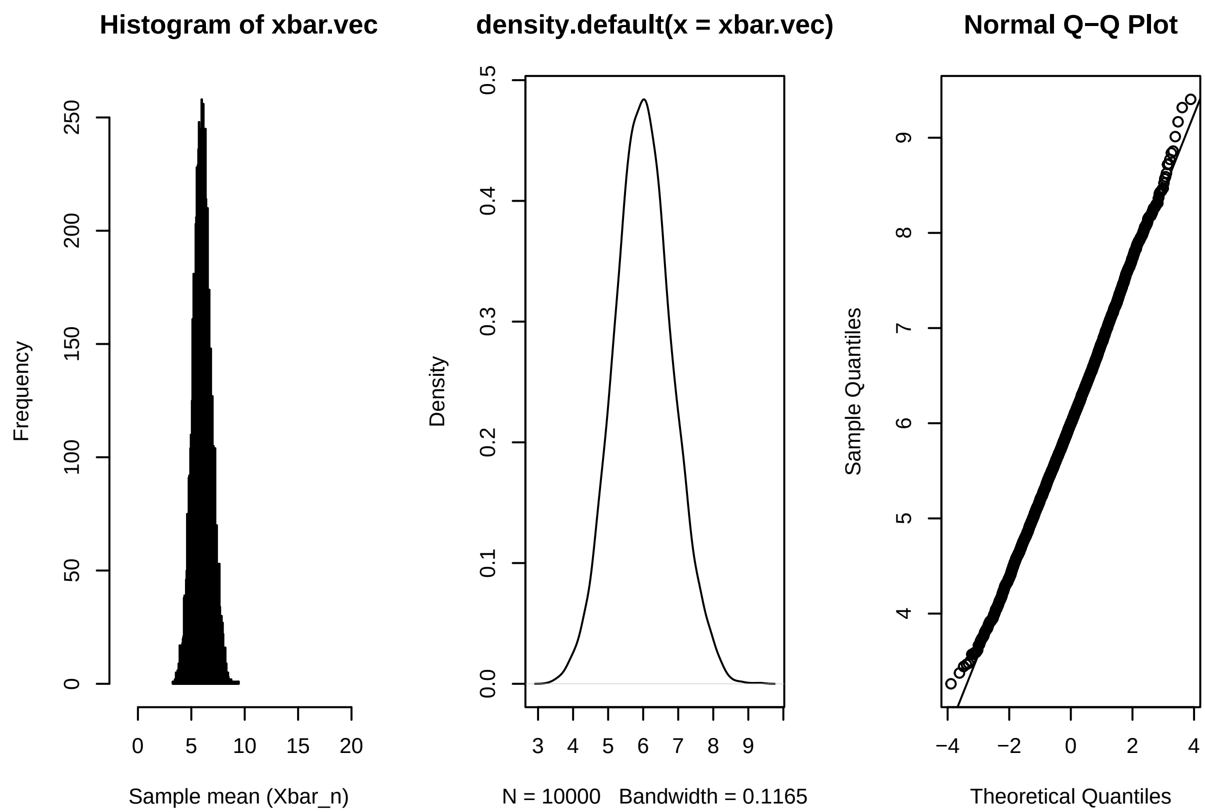


Histogram of xbar.vec
Frequency
0
100
200
300
400
0 5 10 15 20
Sample mean (Xbar_n)
density.default(x = xbar.vec)
Density
0.00
0.05
0.10
0.15
0 5 10 15 20
N = 10000 Bandwidth = 0.6493
Normal Q−Q Plot
Sample Quantiles
0
5
10
15
20
−4 −2 0 2 4
Theoretical Quantiles


```
clt(x = x2, n = 10)
```

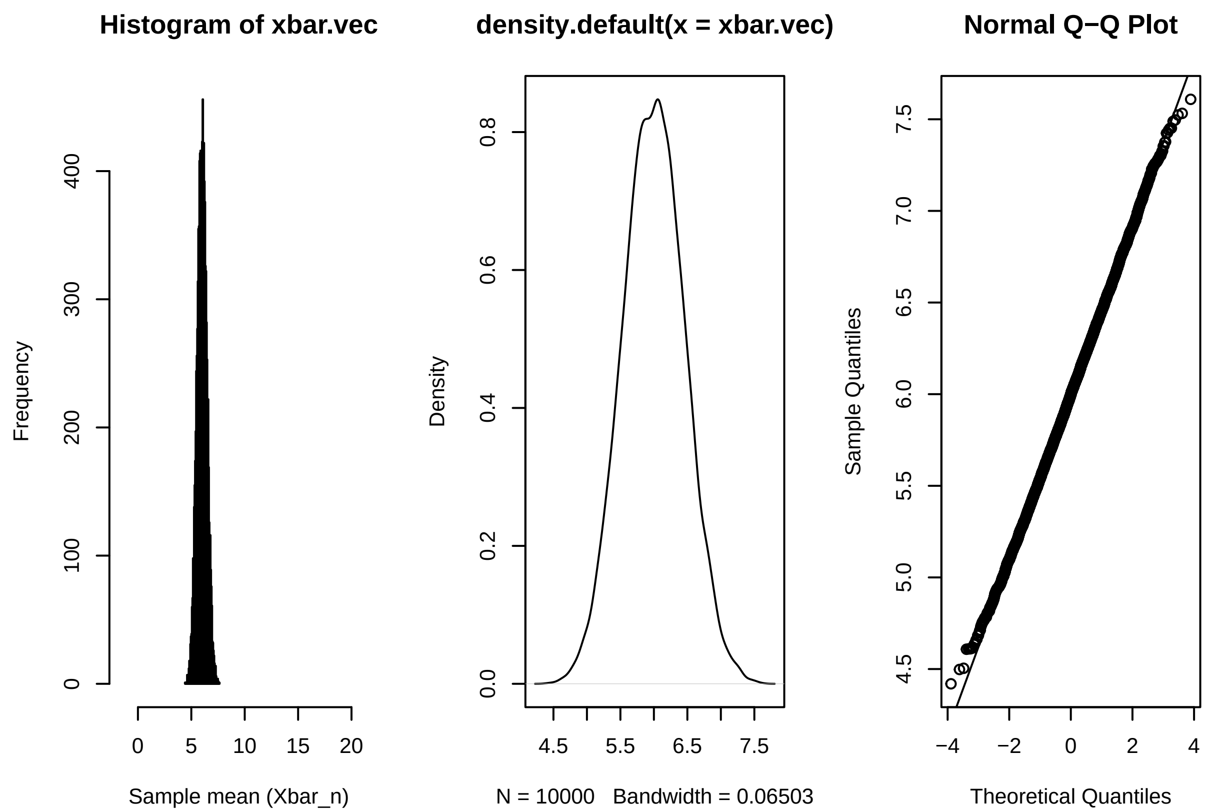

Histogram of xbar.vec
Frequency
0
50
100
150
200
250
0
5
10
15
20
Sample mean (Xbar_n)
density.default(x = xbar.vec)
Density
0.00
0.05
0.10
0.15
0.20
0.25
2
4
6
8
10
N = 10000 Bandwidth = 0.208
Normal Q−Q Plot
Sample Quantiles
2
4
6
8
10
12
−4
−2
0
2
4
Theoretical Quantiles

```
clt(x = x2, n = 30)
```



Histogram of xbar.vec
Frequency
0
50
100
150
200
250
0
5
10
15
20
Sample mean (Xbar_n)
density.default(x = xbar.vec)
Density
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
8
9
N = 10000 Bandwidth = 0.1165
Normal Q−Q Plot
Sample Quantiles
4
5
6
7
8
9
−4
−2
0
2
4
Theoretical Quantiles


```
clt(x = x2, n = 100)
```


Histogram of xbar.vec
Frequency
Sample mean (Xbar_n)
density.default(x = xbar.vec)
Density
N = 10000 Bandwidth = 0.06503
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles