# Assignment 1, EDDA 2017

*Fabio Curi Paix?o and Arash Parnia, group 22*

*11 April 2017*

```
#setwd("C:/Users/Dell/Desktop/Amsterdam/Period 5/Experimental Design and Data Analysis/Assignm
```

```
#load("C:/Users/Dell/Desktop/Amsterdam/Period 5/Experimental Design and Data Analysis/Assignme
load(file="assign1.RData")
```

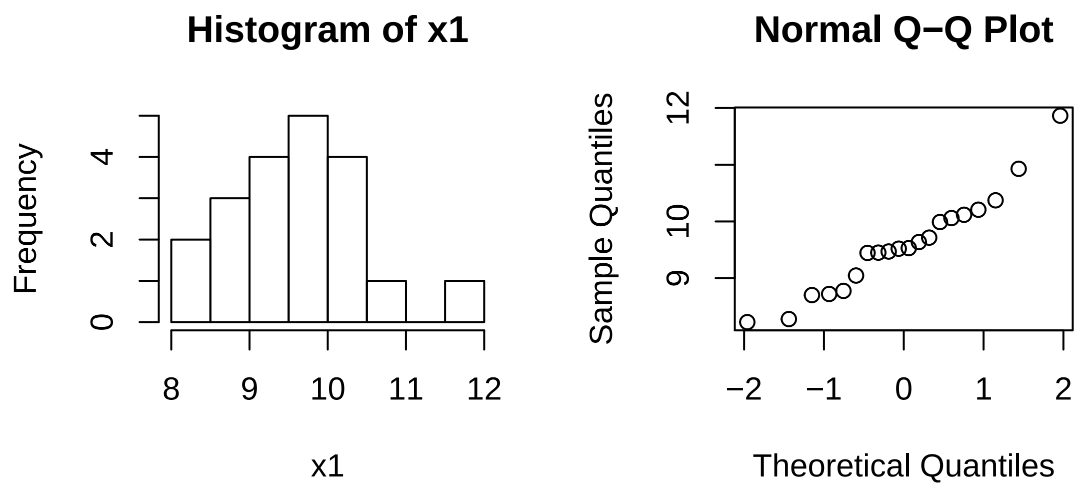
## Introduction

In the present document, the results for the first assignment of the EDDA course are presented.

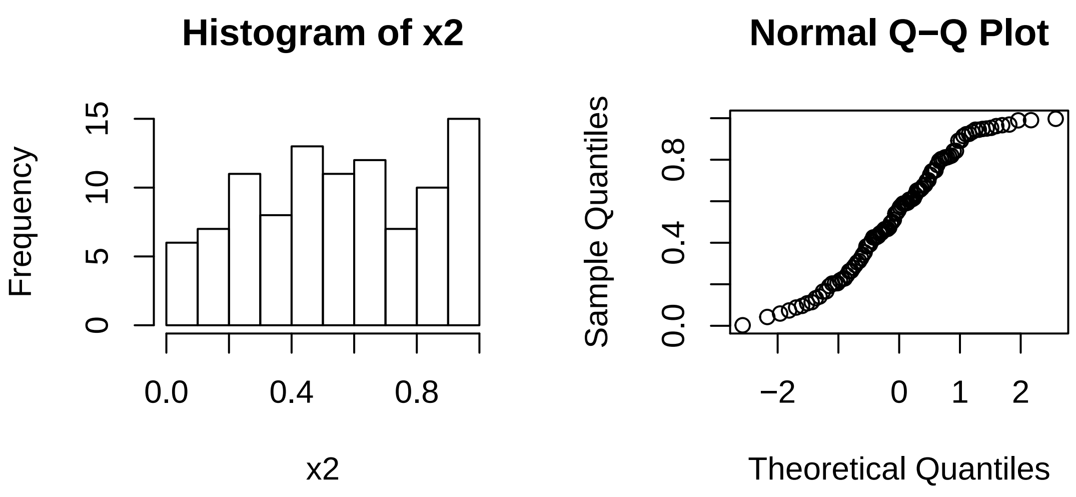
## Exercise 1

The histogram and QQ-plot for x1, x2, x3, x4 and x5 are shown here-after.

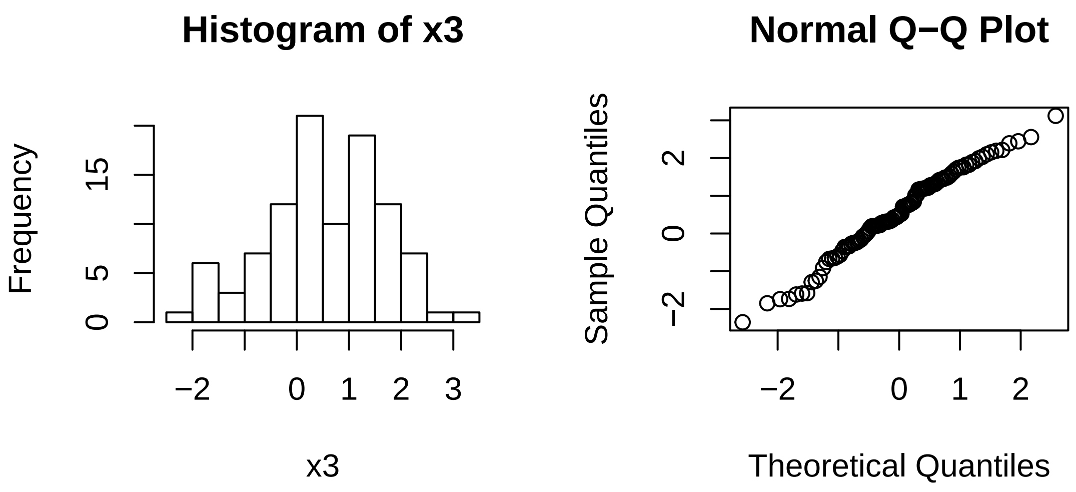
```
par(mfrow=c(1,2));
hist(x1); qqnorm(x1);
```



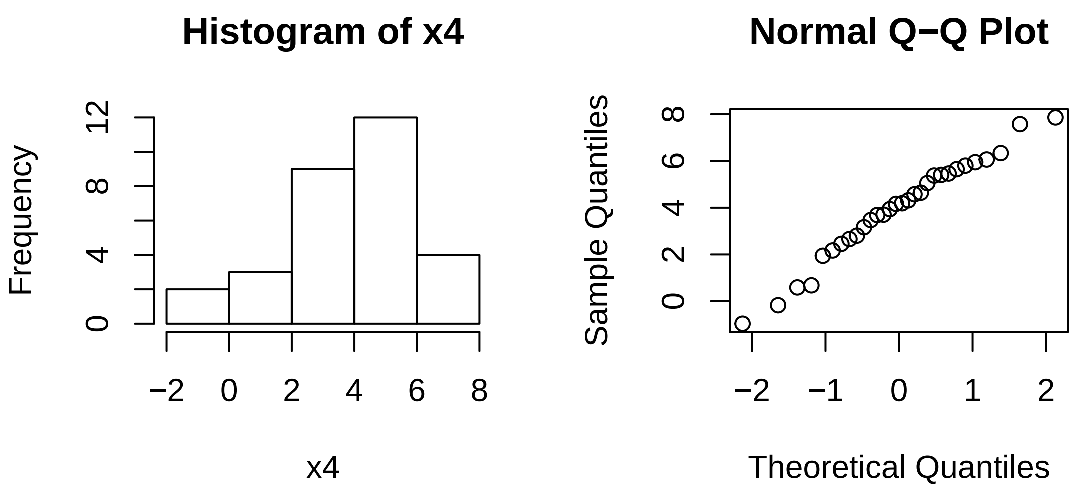
```
par(mfrow=c(1,2));
hist(x2); qqnorm(x2);
```



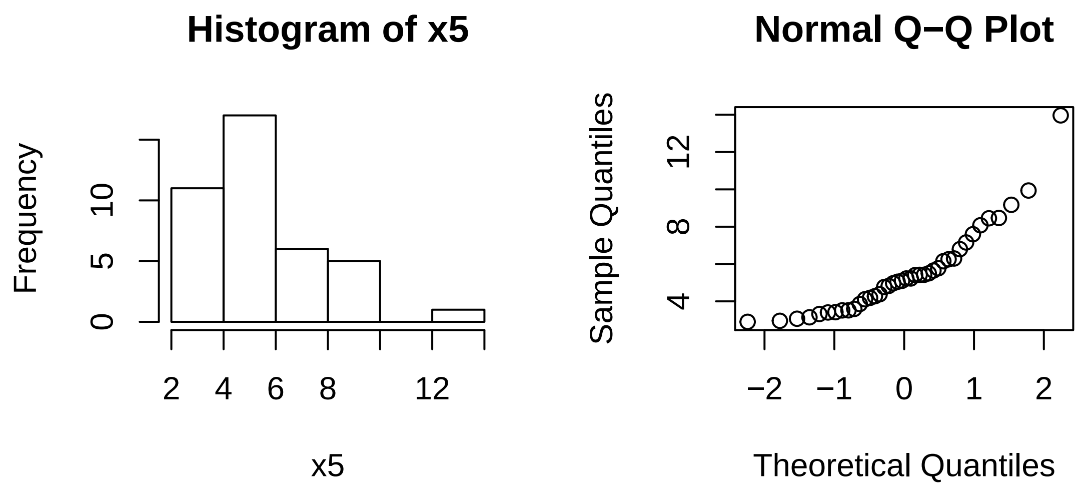
```
par(mfrow=c(1,2));
hist(x3); qqnorm(x3);
```



```
par(mfrow=c(1,2));
hist(x4); qqnorm(x4);
```



```r
par(mfrow=c(1,2));
hist(x5); qqnorm(x5);
```



From the five histogram and QQ-plots here above, we should be able to identify whetherthe data follows a normal distribution. A plot using qqnorm of a sample from a normal distribution will show approximately a straight line, and a deviation from a line indicates that the sample was not taken from a normal population. We have that x1, x2, x3, x4 and x5 have sizes of 20, 100, 100, 30 and 40, respectively.

The elements x2, x3 and x4 look very much like a QQ-plot of a normal distribution. Even though x1 has an uprising behavior, it could still have been sampled from a non-normal distribution as it does not exactly follow a straight line behaviour. Finally, the x5 curve looks more like an exponential one rather than a straight line, thus it does not invite the thought that it has been sampled from a normal distribution.

### Exercise 2

#### Part 1

```r
mu=nu=180
m=n=30
sd=10
B=1000
p=numeric(B)
for (b in 1:B) {x=rnorm(m,mu,sd)
y=rnorm(n,nu,sd)
p[b]=t.test(x,y,var.equal=TRUE)[[3]]}
power=mean(p<0.05)
mean_thres=mean(p[p<0.05])
```

The 1000 p-values are stored within "p". The number of elements smaller than 5% are:

```r
length(p[p<0.05])
```

```
## [1] 48
```

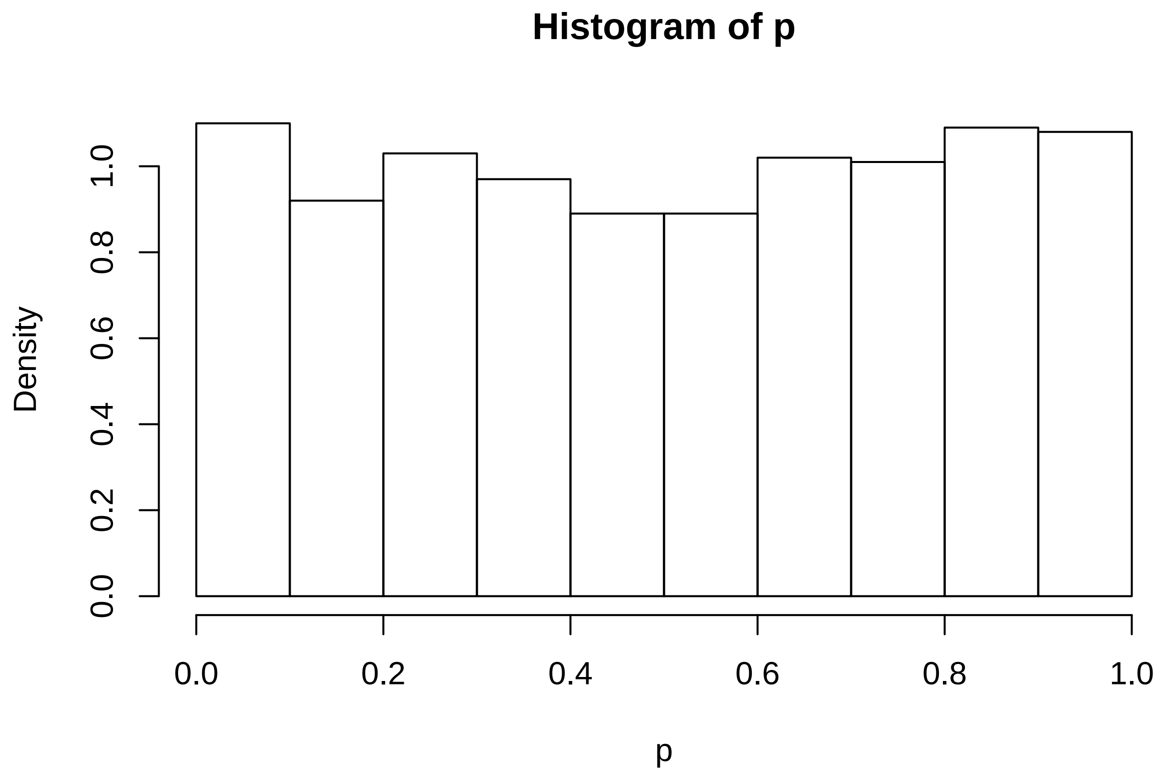
The number of elements smaller than 10% are:

```
length(p[p<0.1])
```

```
## [1] 110
```

The distribution of the p-values is:

```
hist(p,prob=TRUE)
```



**Part 2**

```
mu=nu=180
m=n=30
sd=1
B=1000
p=numeric(B)
for (b in 1:B) {x=rnorm(m,mu,sd)
y=rnorm(n,nu,sd)
p[b]=t.test(x,y,var.equal=TRUE)[[3]]}
power=mean(p<0.05)
mean_thres=mean(p[p<0.05])
```

The 1000 p-values are stored within "p". The number of elements smaller than 5% are:

```
length(p[p<0.05])
```

```
## [1] 43
```

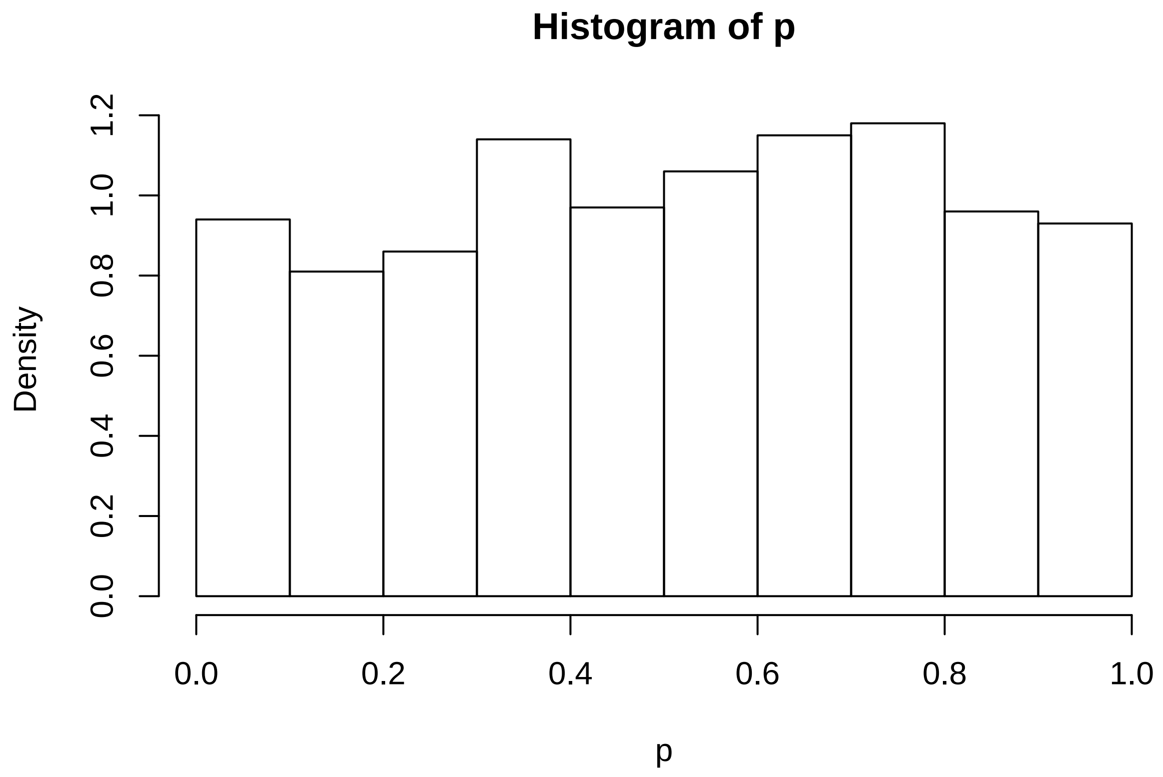
The number of elements smaller than 10% are:

```r
length(p[p<0.1])
```

```
## [1] 94
```

The distribution of the p-values is:

```r
hist(p,prob=TRUE)
```



**Part 3**

```r
mu=180
nu=175
m=n=30
sd=6
B=1000
p=numeric(B)
for (b in 1:B) {x=rnorm(m,mu,sd)
y=rnorm(n,nu,sd)
p[b]=t.test(x,y,var.equal=TRUE)[[3]]}
power=mean(p<0.05)
mean_thres=mean(p[p<0.05])
```

The 1000 p-values are stored within “p”. The number of elements smaller than 5% are:

```r
length(p[p<0.05])
```

```
## [1] 913
```

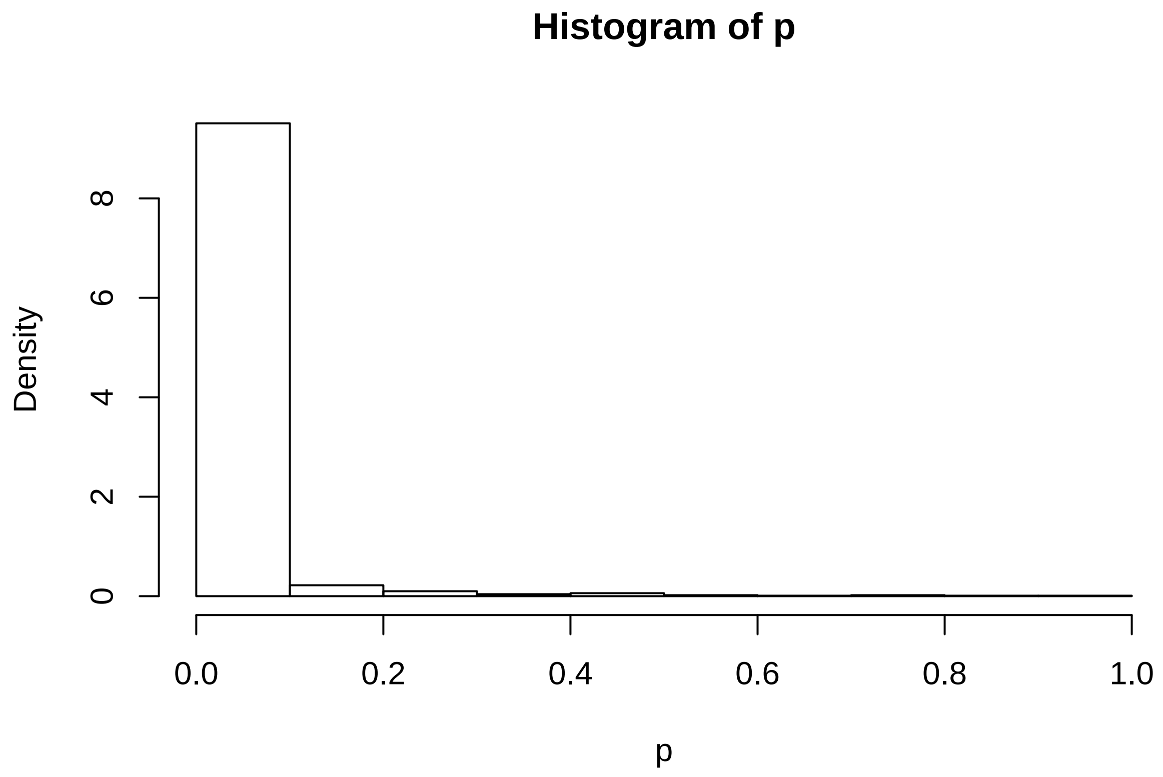
The number of elements smaller than 10% are:

```
length(p[p<0.1])
```

```
## [1] 951
```

The distribution of the p-values is:

```
hist(p,prob=TRUE)
```



### Part 4

The test here is to check whether the null hypothesis that the population means of the two populations are equal. The two first histograms show the frequencies of the p-values for two randomly generated x and y arrays with center values of 180, size 30 and differing only by their standard deviations. For sd=10, the lowest p-values are seen more often than sd=1. Analogically, for sd=1, the highest p-values are seen more often than sd=10. This is explained by the fact that for smaller standard deviation values, the two arrays x and y have their values within a narrower interval, which increases the probability that the H0 hipothesis is true.

## Exercise 3

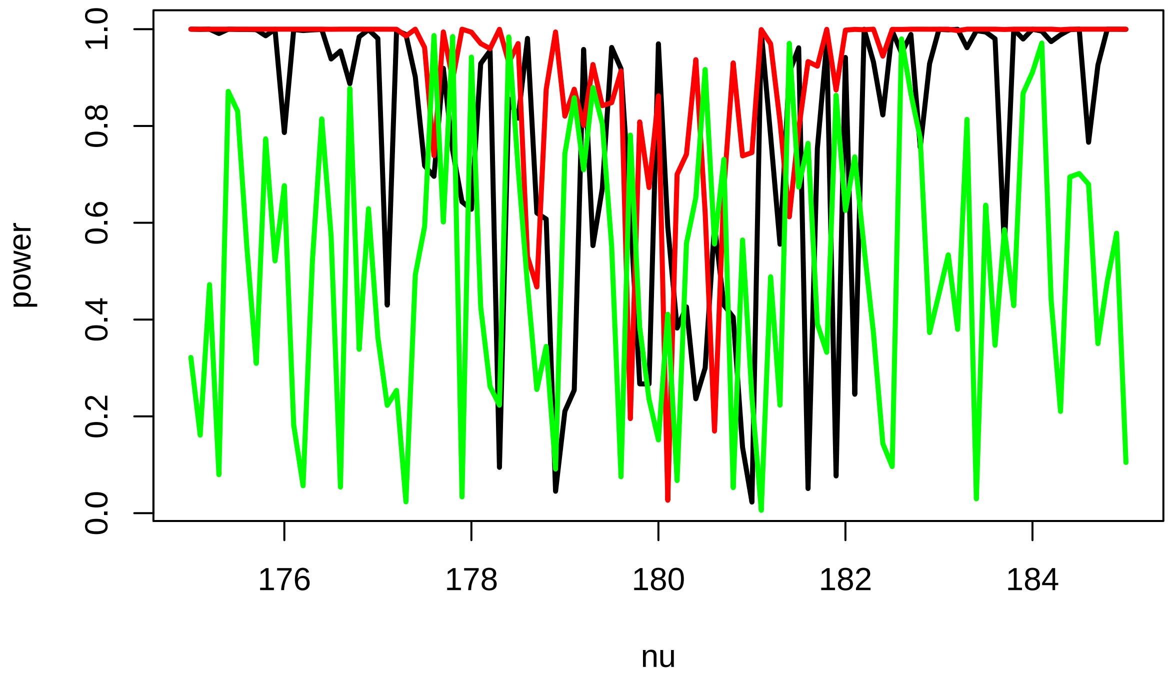
### Parts 1 & 2 & 3

Here after we have the code for computing the power of a test with respect to the different nu values. The power of a test is 1 minus the probability of an error of the second kind, which is the p-value. Thus, the power of a test is here defined as 1 - p-value.

```r
mu=180
m=n=30
sd=5
nu = seq(175,185,by=0.1)
size=length(nu)
p=numeric(size)
for (b in 1:size) {x=rnorm(m,mu,sd)
y=rnorm(n,nu[b],sd)
p[b]=t.test(x,y,var.equal=TRUE)[[3]]}
power=mean(p<0.05)
mean_thres=mean(p[p<0.05])
power=1-p
plot(nu,power,type="l",lwd=2.5)

mu=180
m=n=100
sd=5
nu = seq(175,185,by=0.1)
size=length(nu)
p=numeric(size)
for (b in 1:size) {x=rnorm(m,mu,sd)
y=rnorm(n,nu[b],sd)
p[b]=t.test(x,y,var.equal=TRUE)[[3]]}
power=mean(p<0.05)
mean_thres=mean(p[p<0.05])
power=1-p
lines(nu,power,type="l",col="red",lwd=2.5)

mu=180
m=n=30
sd=100
nu = seq(175,185,by=0.1)
size=length(nu)
p=numeric(size)
for (b in 1:size) {x=rnorm(m,mu,sd)
y=rnorm(n,nu[b],sd)
p[b]=t.test(x,y,var.equal=TRUE)[[3]]}
power=mean(p<0.05)
mean_thres=mean(p[p<0.05])
power=1-p
lines(nu,power,type="l",col="green",lwd=2.5)
```



The plots in the previous Figure represent the following cases:

Black: mu=180; m=n=30; sd=5; Red:mu=180; m=n=100; sd=5; Green: mu=180; m=n=30; sd=100;

Comparing the black and red curves, the principal difference is the length of the sample, which is more than three times bigger in the second case. As the samples are randomly taken from an uniform distribution, the higher the amount of the sample, the better the power of the test in this case, as there will be more numbers randomly generated with central values around 180 and sd of 5. Furthermore, the power of the test is overall lower for the green case, which makes use of a higher standard deviation. Thus, the mean values of x and y are a lot more unlikely to match (which is the hipothesis H0).