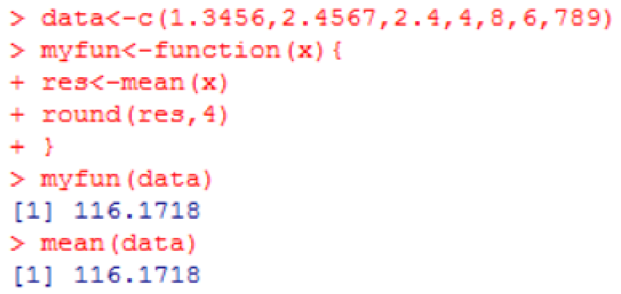
Datamining TP 3

YAO ZELIANG & ZHAO HE

Exercice 1

Create a function that will calculate the mean of a vector with an accuracy of four decimal places



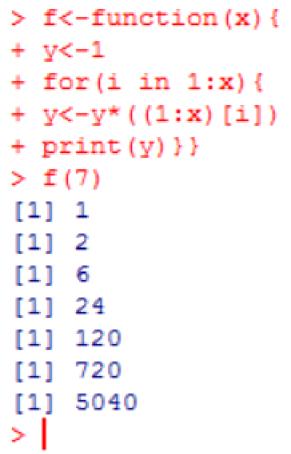
```
> data<-c(1.3456,2.4567,2.4,4,8,6,789)
> myfun<-function(x){
+ res<-mean(x)
+ round(res,4)
+ }
> myfun(data)
[1] 116.1718
> mean(data)
[1] 116.1718
```

Exercice 2

Programme factorial n, $n! = 1\times 2\times ...\times (n-1)$ x n using :

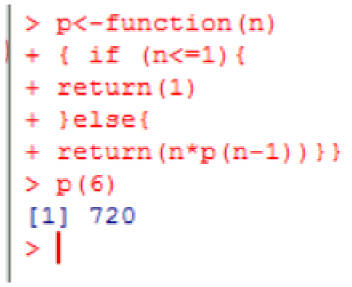
- a loop ***for***
- the ***prod()*** function.

Loop for:



```
> f<-function(x){
+ y<-1
+ for(i in 1:x){
+ y<-y*((1:x)[i])
+ print(y)}}
> f(7)
[1] 1
[1] 2
[1] 6
[1] 24
[1] 120
[1] 720
[1] 5040
> 
```

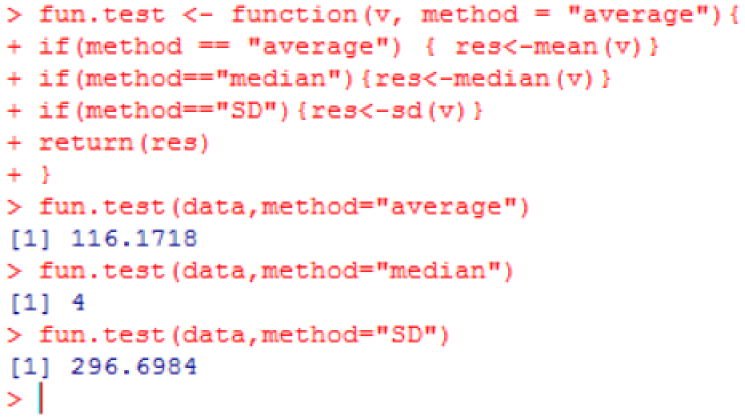
Prod()



```
> p<-function(n)
+ { if (n<=1){
+ return(1)
+ }else{
+ return(n*p(n-1))}}
> p(6)
[1] 720
> 
```

Exercice 3

Create a function that will calculate the mean, median and standard deviation of a vector and will display one of the chosen options "average", "median" or "SD".



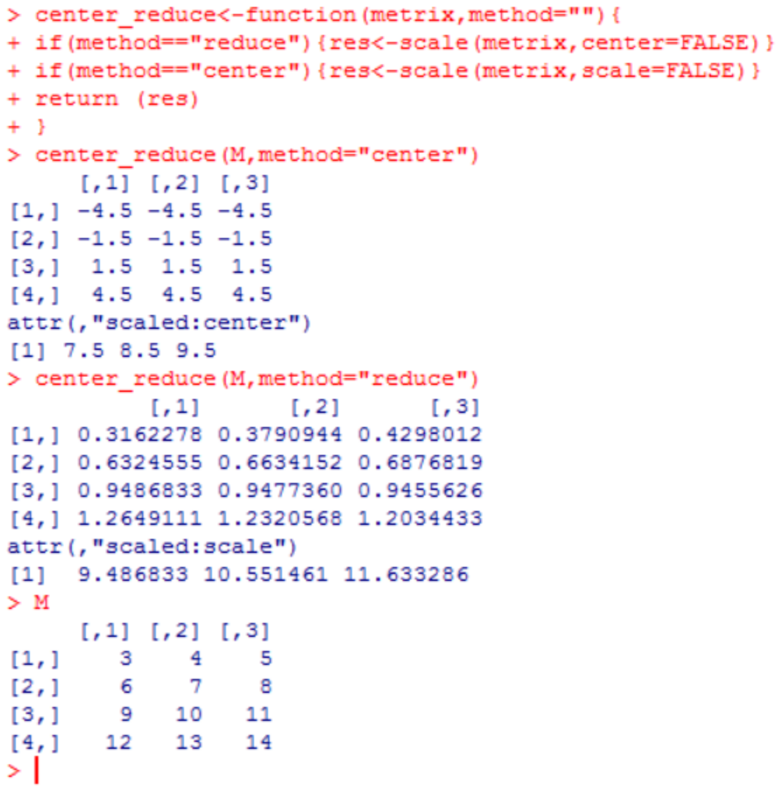
```
> fun.test <- function(v, method = "average"){
+ if(method == "average") { res<-mean(v)}
+ if(method=="median"){res<-median(v)}
+ if(method=="SD"){res<-sd(v)}
+ return(res)
+ }
> fun.test(data,method="average")
[1] 116.1718
> fun.test(data,method="median")
[1] 4
> fun.test(data,method="SD")
[1] 296.6984
> |
```

Exercice 4

Write a function that replaces the negative values of a vector by their absolute values and then displays the modified vector.

Exercice 6

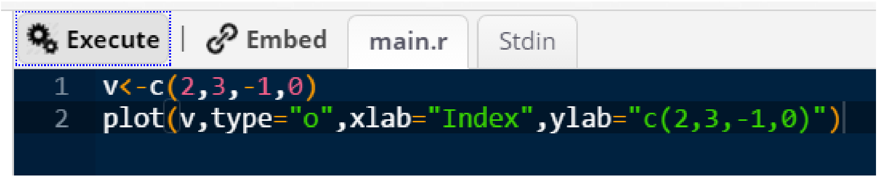
Create a function that will center and reduce (scale) a data matrix.



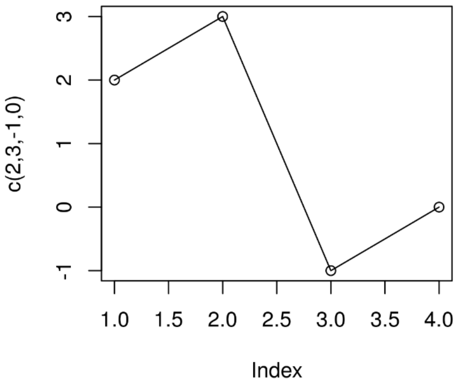
```
> center_reduce<-function(metrix,method=""){
+ if(method=="reduce"){res<-scale(metrix,center=FALSE)}
+ if(method=="center"){res<-scale(metrix,scale=FALSE)}
+ return (res)
+ }
> center_reduce(M,method="center")
     [,1] [,2] [,3]
[1,] -4.5 -4.5 -4.5
[2,] -1.5 -1.5 -1.5
[3,]  1.5  1.5  1.5
[4,]  4.5  4.5  4.5
attr(,"scaled:center")
[1] 7.5 8.5 9.5
> center_reduce(M,method="reduce")
          [,1]      [,2]      [,3]
[1,] 0.3162278 0.3790944 0.4298012
[2,] 0.6324555 0.6634152 0.6876819
[3,] 0.9486833 0.9477360 0.9455626
[4,] 1.2649111 1.2320568 1.2034433
attr(,"scaled:scale")
[1]  9.486833 10.551461 11.633286
> M
     [,1] [,2] [,3]
[1,]    3    4    5
[2,]    6    7    8
[3,]    9   10   11
[4,]   12   13   14
> 
```

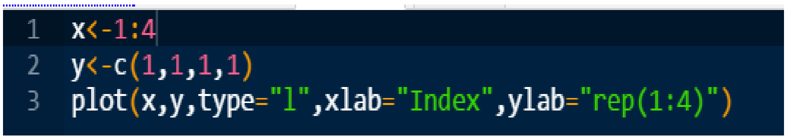
Exercice 7

a) Represent the following graph:

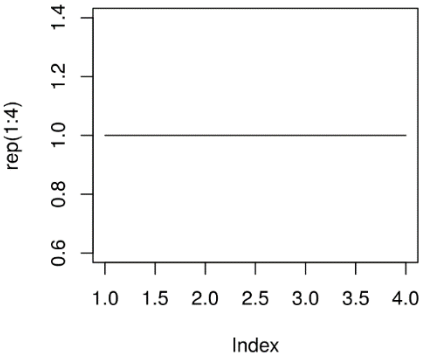


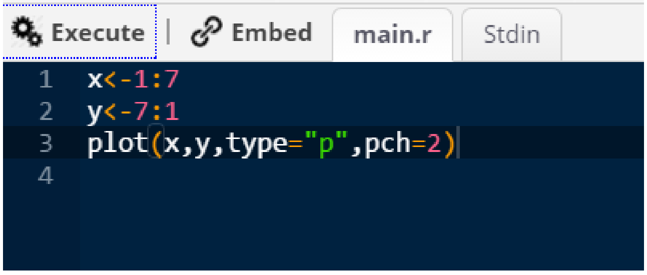
```
v<-c(2,3,-1,0)
plot(v,type="o",xlab="Index",ylab="c(2,3,-1,0)")
```





```r
x<-1:4
y<-c(1,1,1,1)
plot(x,y,type="l",xlab="Index",ylab="rep(1:4)")
```



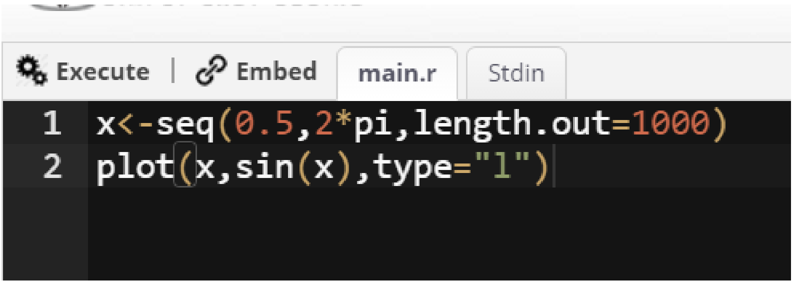


```r
x<-1:7
y<-7:1
plot(x,y,type="p",pch=2)
```

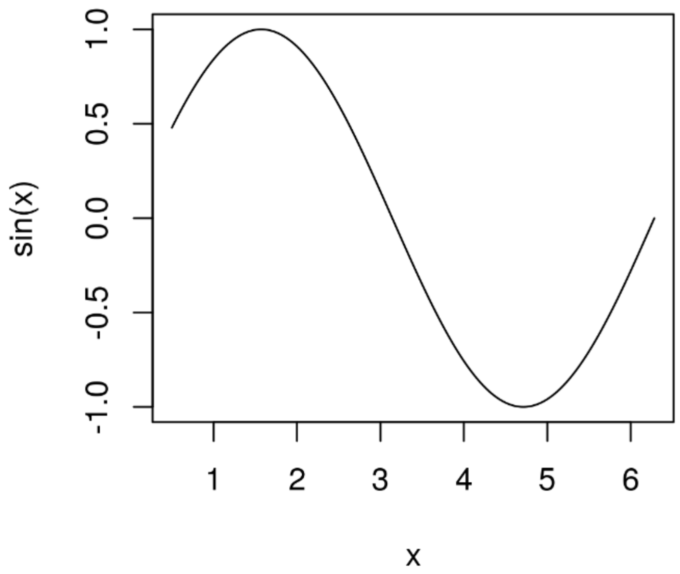
b) Reproduce the graph above but with a graphic at the top right with a width of 1 to 3 with respect to the upper left. You must obtain the following graph:

Exercice 8

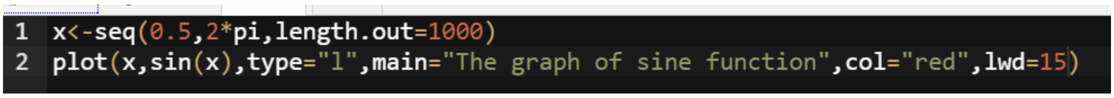
a) Draw the graph of the sine function between 0.5 and 3π (use *pi*).



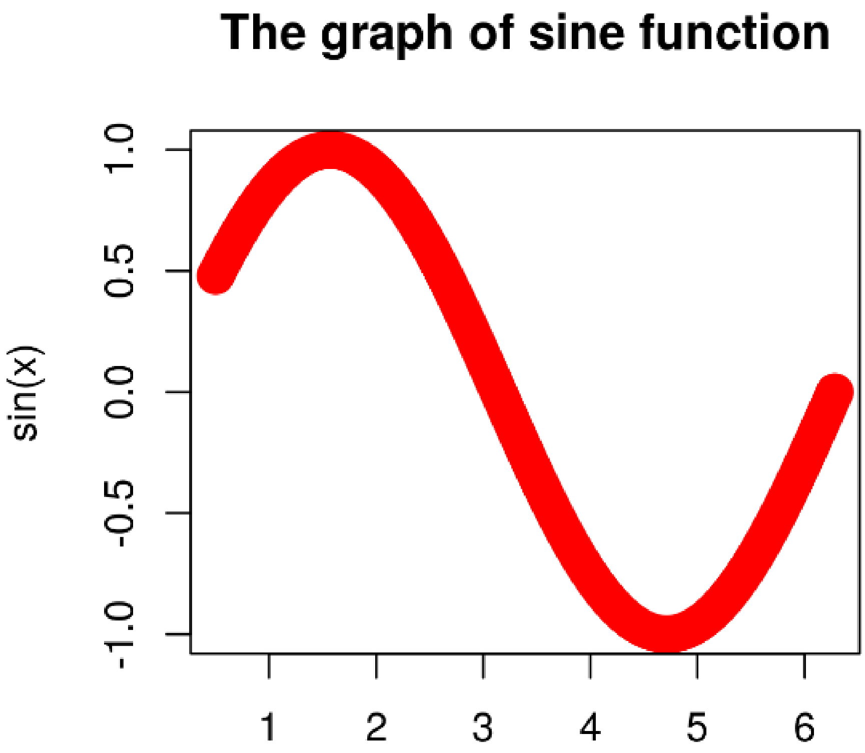
```r
x<-seq(0.5,2*pi,length.out=1000)
plot(x,sin(x),type="l")
```



b) Add the following title: "The graph of the sine function," the color of the curve must be red and the line thickness 15.

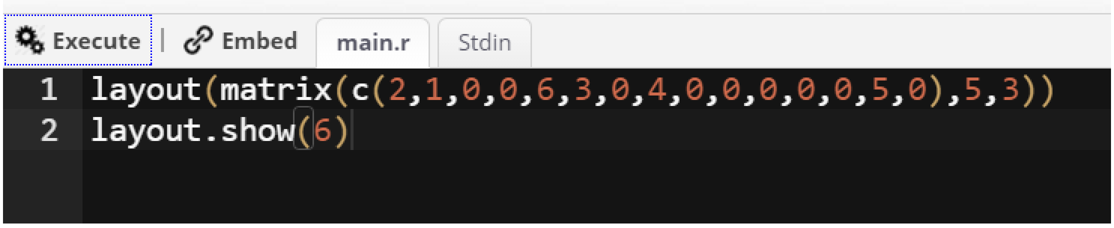


```r
x<-seq(0.5,2*pi,length.out=1000)
plot(x,sin(x),type="l",main="The graph of sine function",col="red",lwd=15)
```

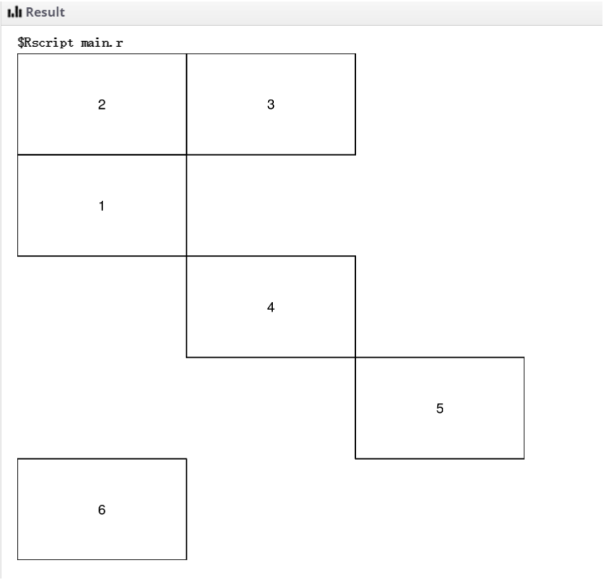


Exercice 9

Write the script that will produce the following breakdown of the graphics window:



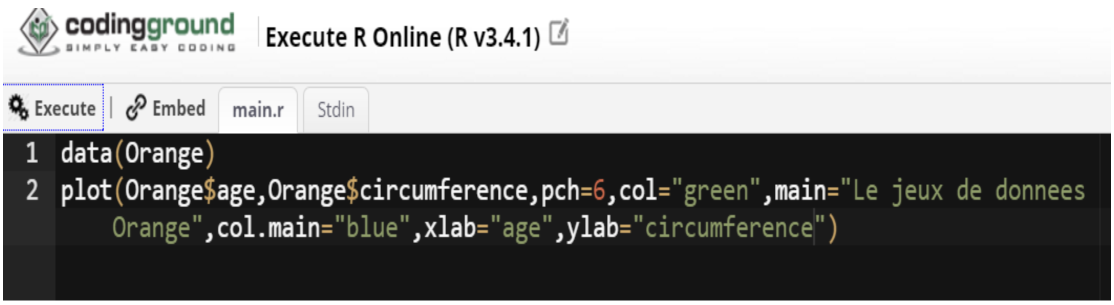
```r
layout(matrix(c(2,1,0,0,6,3,0,4,0,0,0,0,0,5,0),5,3))
layout.show(6)
```



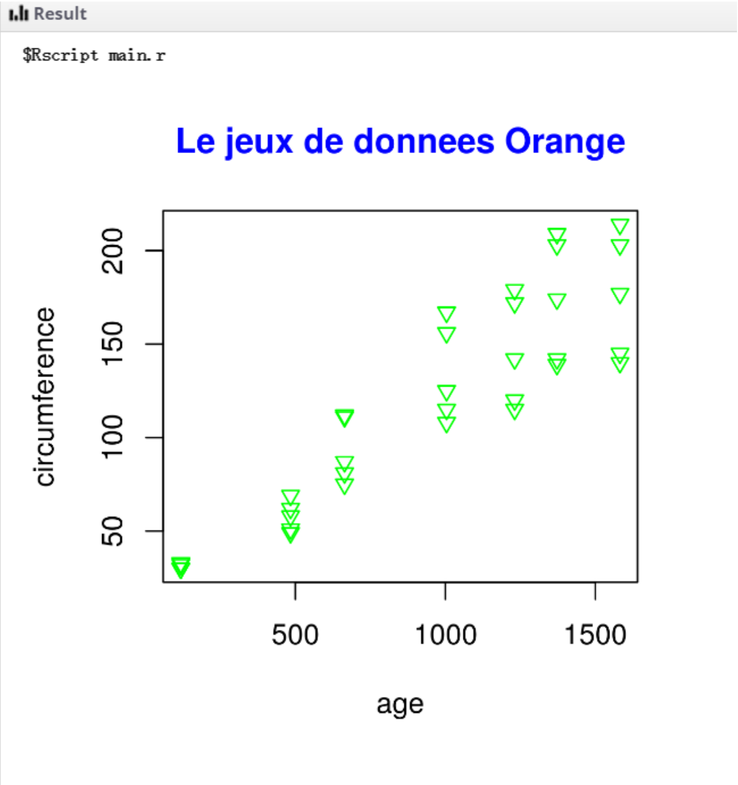
```
$Rscript main.r
```

Exercice 10

Load the Orange dataset. Draw a scatterplot of the variable ***age*** depending on the variable ***circumference***. Change the parameters of the function plot *(pch, col.main, sub, ylab ….)* to obtain the following representation:



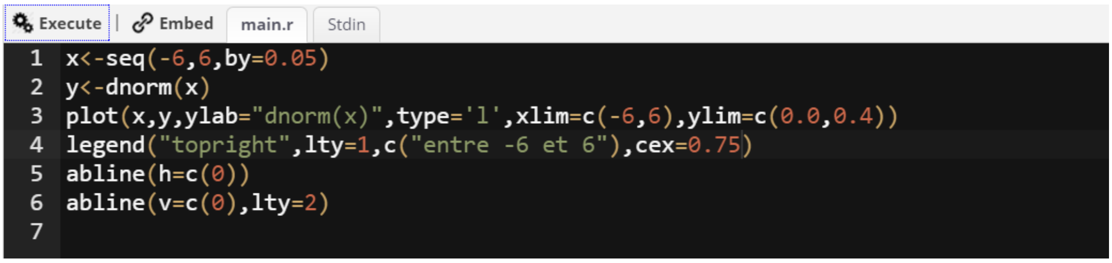
```r
data(Orange)
plot(Orange$age,Orange$circumference,pch=6,col="green",main="Le jeux de donnees Orange",col.main="blue",xlab="age",ylab="circumference")
```



```
$Rscript main.r
```

Exercice 11

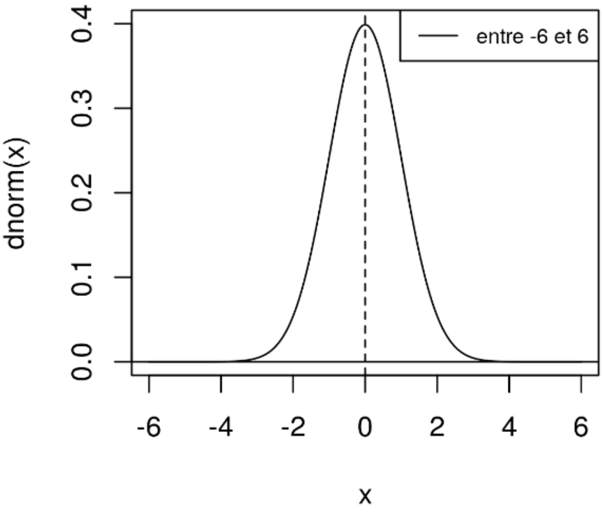
Draw the graph of the normal distribution between -6 and 6 (use dnorm). Add a caption at the top right which will display "Normal distribution between -6 and 6." You should get the following result:



```r
x<-seq(-6,6,by=0.05)
y<-dnorm(x)
plot(x,y,ylab="dnorm(x)",type='l',xlim=c(-6,6),ylim=c(0.0,0.4))
legend("topright",lty=1,c("entre -6 et 6"),cex=0.75)
abline(h=c(0))
abline(v=c(0),lty=2)

```

```
$Rscript main.r
```


entre -6 et 6
dnorm(x)
x