Introduction to R software

Correction of exercises

Céline Keime
keime@igbmc.fr

Exercise 1

```
# Number of flowers from setosa species
> sum(iris$Species=="setosa")
[1] 50

# Sepal length in increasing order
> sort(iris$Sepal.Length)

# Mean sepal length
 > mean(iris$Sepal.Length)
[1] 5.843333

# Number of flowers with a sepal length higher than 5 cm
> sum(iris$Sepal.Length > 5 )
[1] 118

# Number of setosa flowers with a sepal length larger than 5 cm
> sum( (iris$Sepal.Length> 5) & (iris$Species=="setosa" ) )
[1] 22
```

Exercise 2

```
> sl = iris$Sepal.Length

# Class of sl object
> class(sl)
[1] "numeric”

# Minimal, maximal and mean sepal length
> min(sl)
[1] 4.3
> max(sl)
[1] 7.9

# Length of the 10 largest sepals
> sort(sl, decreasing=TRUE)[1:10]
 [1] 7.9 7.7 7.7 7.7 7.7 7.6 7.4 7.3 7.2 7.2
```

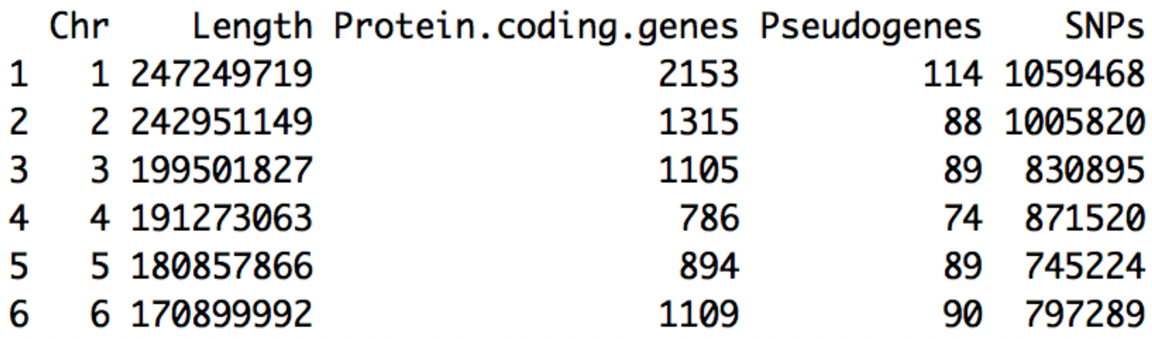
Exercise 3

```
# Class of the Species object from the iris dataset
> class(iris$Species)
[1] "factor"

# Levels of this factor
> levels(iris$Species)
[1] "setosa"     "versicolor" "virginica"

# Number of flowers for each species
> table(iris$Species)
    setosa versicolor  virginica
        50         50         50

# Species of the flower with the smallest sepal length
> iris$Species[iris$Sepal.Length == min(iris$Sepal.Length)]
[1] setosa
```

Exercise 4

# Importation of the humanGenomeSummary.txt file into R
huge=read.table("humanGenomeSummary.txt", header=TRUE ,sep="\t")

# Visualisation of the created object
head(huge)



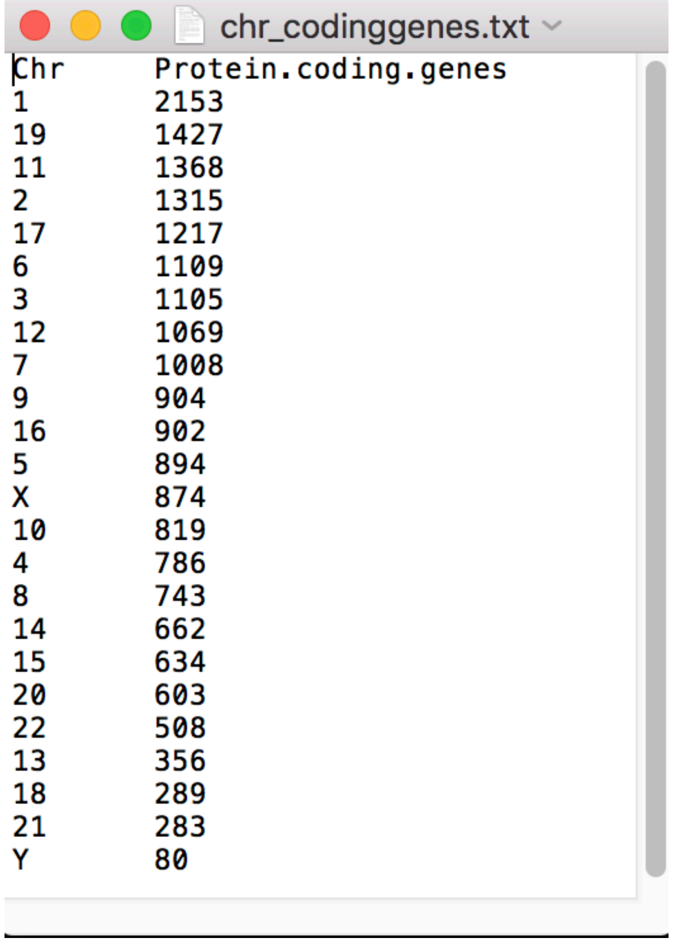
```
  Chr    Length Protein.coding.genes Pseudogenes    SNPs
1   1 247249719                 2153         114 1059468
2   2 242951149                 1315          88 1005820
3   3 199501827                 1105          89  830895
4   4 191273063                  786          74  871520
5   5 180857866                  894          89  745224
6   6 170899992                 1109          90  797289
```

# Total number of protein coding genes
sum(huge$Protein.coding.genes)
[1] 21108

# Chromosomes with more than 1,000 annotated protein coding genes
huge$Chr[huge$Protein.coding.genes>1000]
[1] 1 2 3 6 7 11 12 17 19

Exercise 4

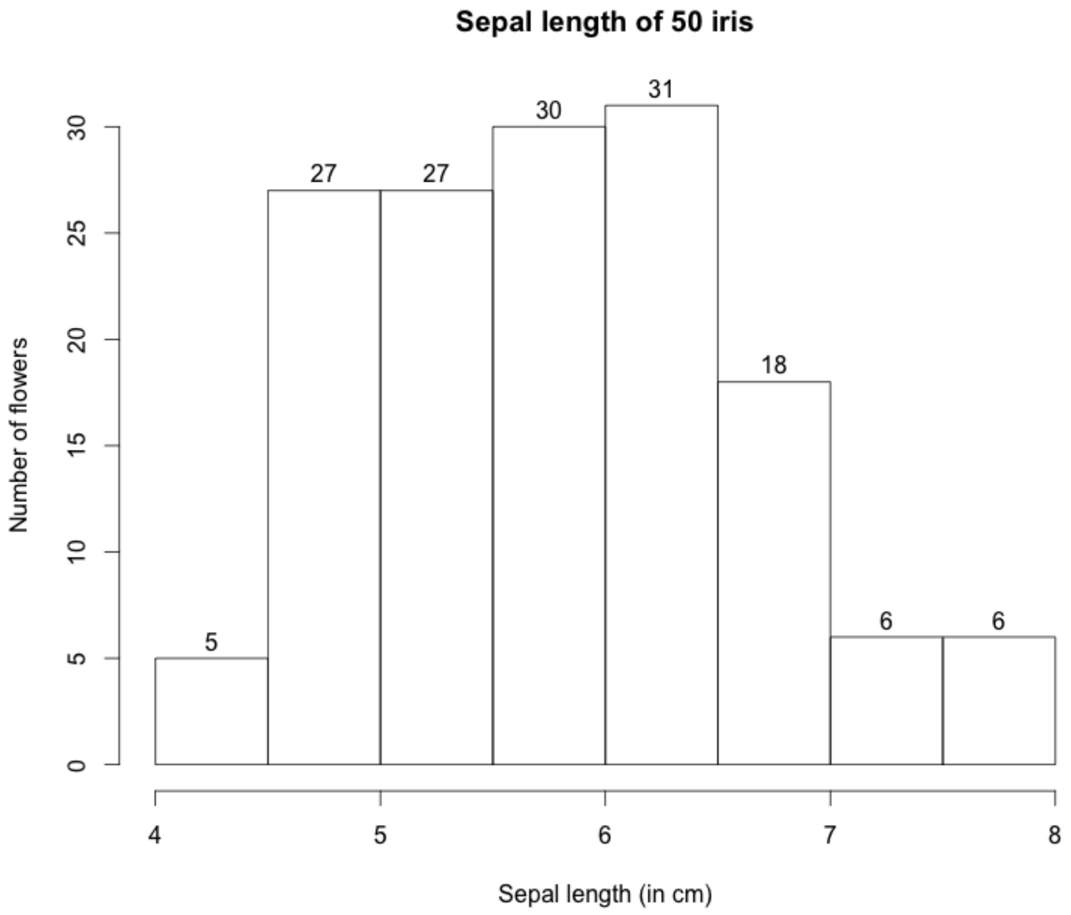
```
write.table( huge[order(huge$Protein.coding.genes, decreasing=TRUE), c(1,3)],
    file="chr_codinggenes.txt", sep="\t", quote=FALSE, row.names=FALSE)
```



```
Chr	Protein.coding.genes
1	2153
19	1427
11	1368
2	1315
17	1217
6	1109
3	1105
12	1069
7	1008
9	904
16	902
5	894
X	874
10	819
4	786
8	743
14	662
15	634
20	603
22	508
13	356
18	289
21	283
Y	80
```

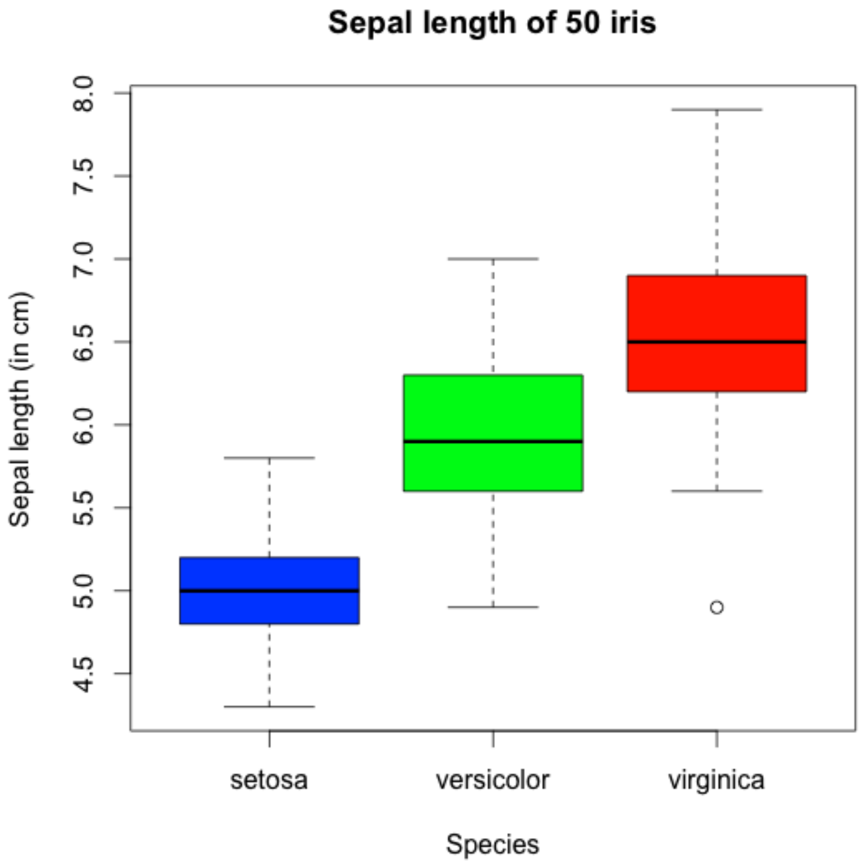
Exercise 5 : histogram

```
hist(iris$Sepal.Length, labels=TRUE, xlab="Sepal length (in cm)",
ylab="Number of flowers", main="Sepal length of 50 iris")
```



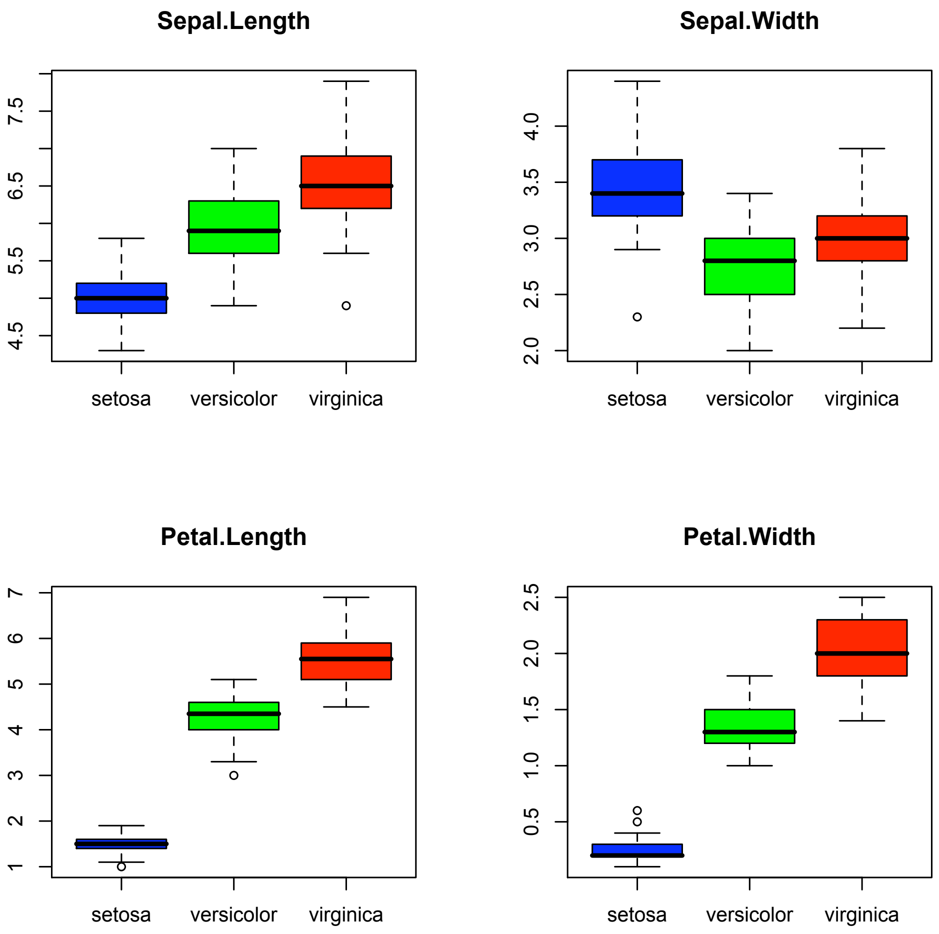
Exercise 5 : boxplot

```
boxplot(iris$Sepal.Length ~ iris$Species, col=c("blue","green","red"),
main= "Sepal length of 50 iris",
xlab= "Species", ylab= "Sepal length (in cm)")
```



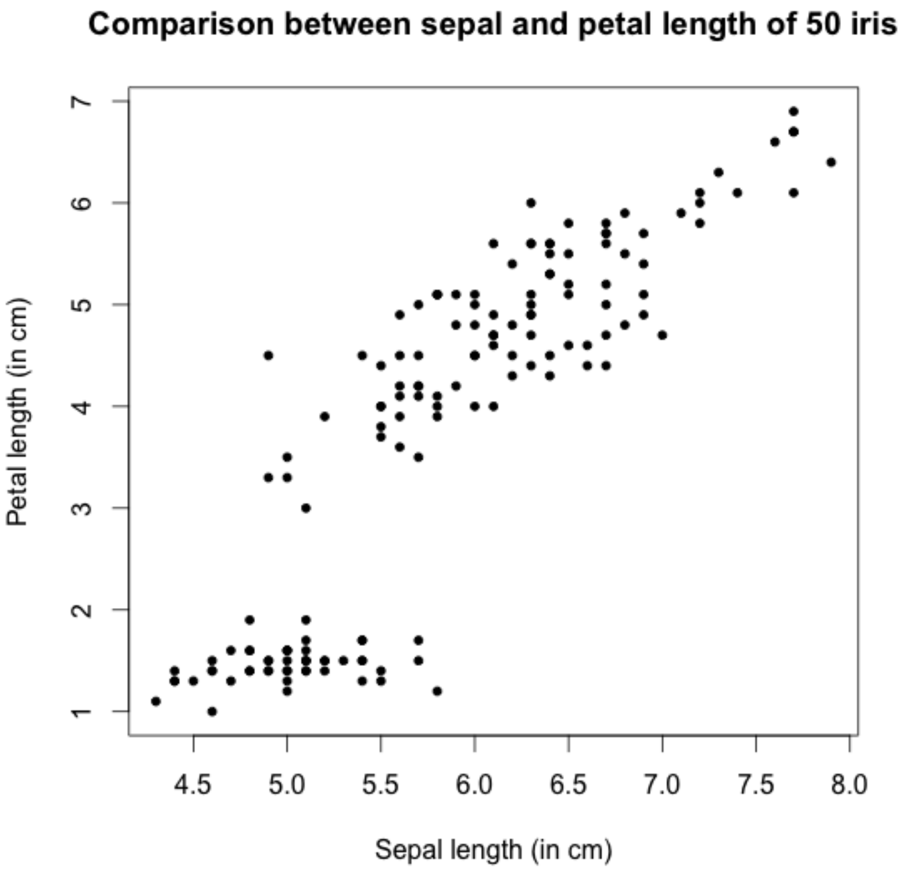
Exercise 5 : boxplots and graphics windows partition

```
par(mfrow=c(2,2))
for (i in 1:4){ # for each quantitative variable
     boxplot( iris[,i]~iris$Species, main=colnames(iris)[i], col=c("blue","green","red"))
}
```



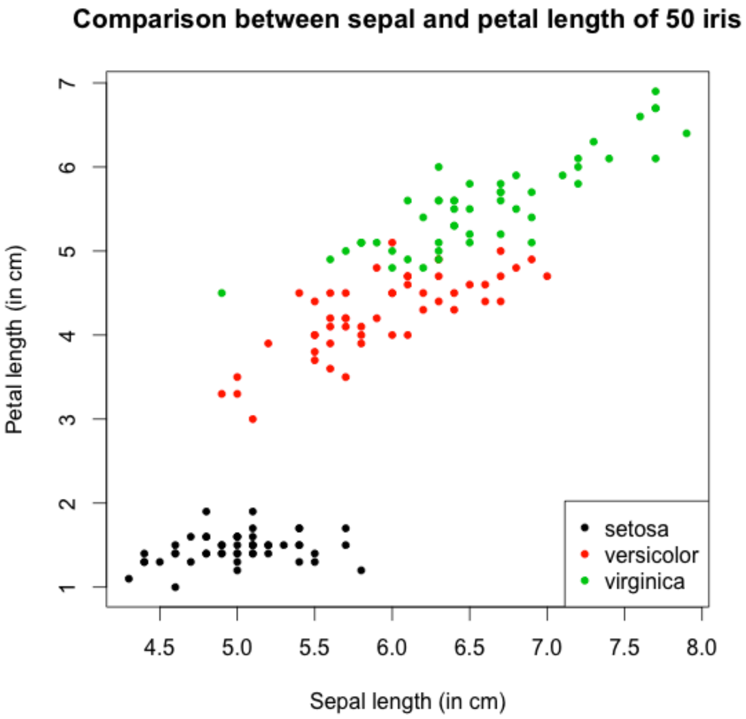
Exercise 5 : scatterplot

```
plot(iris$Sepal.Length, iris$Petal.Length,
xlab="Sepal length (in cm)", ylab=”Petal length (in cm)",
main="Comparison between sepal and petal length of 50 iris", pch=20)
```



Exercise 5 : scatterplot with different colours

```
# graph
plot(iris$Sepal.Length, iris$Petal.Length,
xlab="Sepal length (in cm)", ylab="Petal length (in cm)",
main="Comparison between sepal and petal length of 50 iris", pch=20,
col=iris$Species)

# legend
legend("bottomright", legend=levels(iris$Species), col=1:3, pch=20)
```

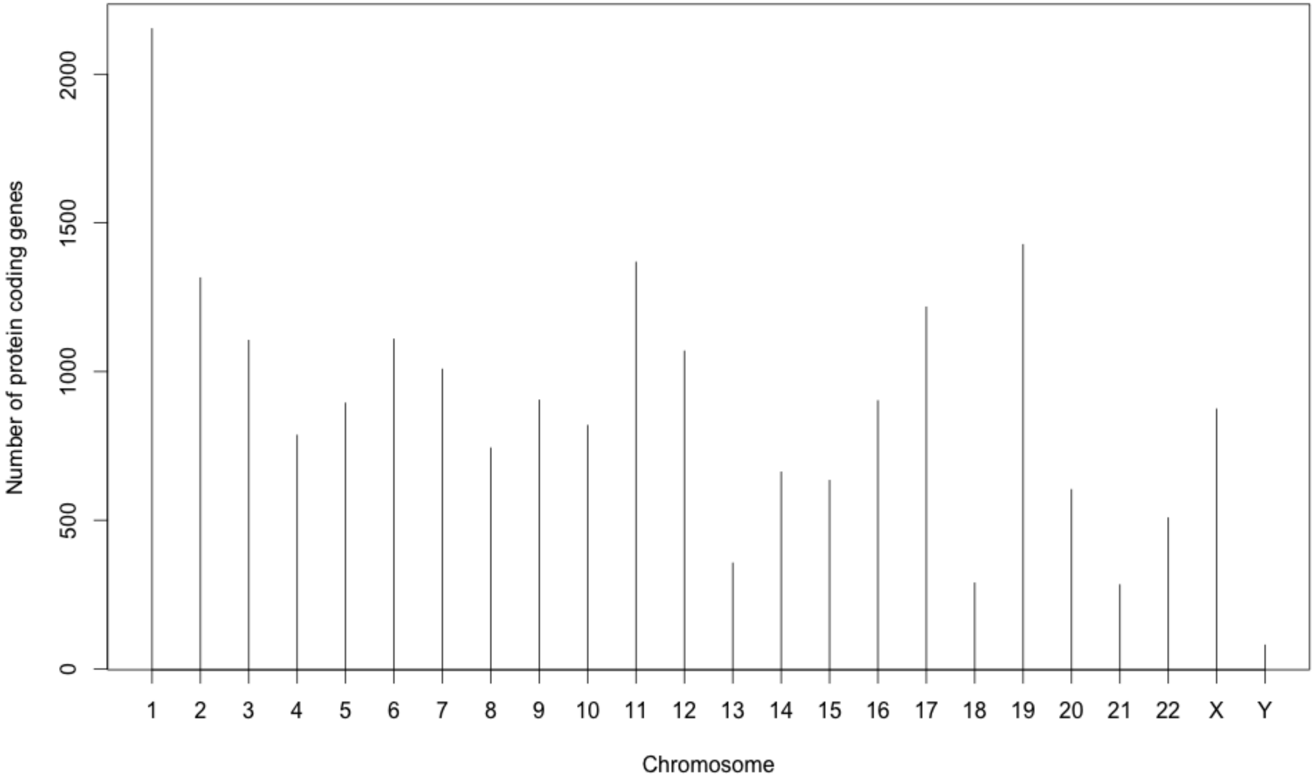


Exercise 6

```
# Mean number of protein coding genes per chromosome
mean(huge$Protein.coding.genes )
[1] 879.5

# Maximum number of protein coding genes per chromosome
max(huge$Protein.coding.genes)
[1] 2153

# The above information can also be obtained using the summary function
summary(huge$Protein.coding.genes)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   80.0   626.2   884.0   879.5  1106.0  2153.0

# Chromosome with the highest number of coding genes
huge[huge$Protein.coding.genes ==max(huge$Protein.coding.genes ),1]
[1] 1

# Chromosome with the smallest number of coding genes
huge[huge$Protein.coding.genes ==min(huge$Protein.coding.genes ),1]
[1] Y
```

Exercise 6

```
# Bar-chart
plot(huge$Protein.coding.genes, xlab="Chromosome",
ylab="Number of protein coding genes", type="h", xaxt="n")
axis(1, at=1:length(huge$Chr), labels=huge$Chr)
```

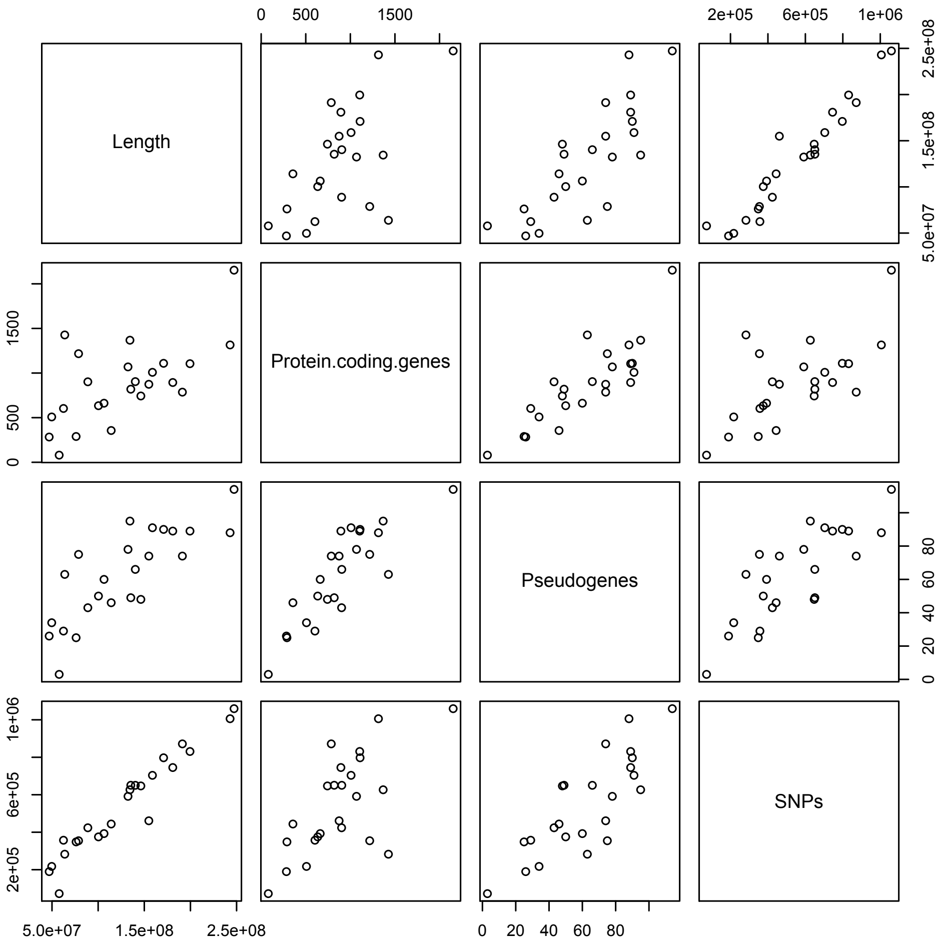


Exercise 7

```
# Pearson correlation coefficient
# between chromosome size and the number of protein coding genes
cor(huge$Length, huge$Protein.coding.genes)
[1] 0.6305566

# Pearson correlation coefficients between all pairs of quantitative variables
cor(huge[,2:5])
                        Length Protein.coding.genes Pseudogenes      SNPs
Length               1.0000000            0.6305566   0.8070697 0.9631934
Protein.coding.genes 0.6305566            1.0000000   0.8527515 0.6731252
Pseudogenes          0.8070697            0.8527515   1.0000000 0.8158501
SNPs                 0.9631934            0.6731252   0.8158501 1.0000000
```

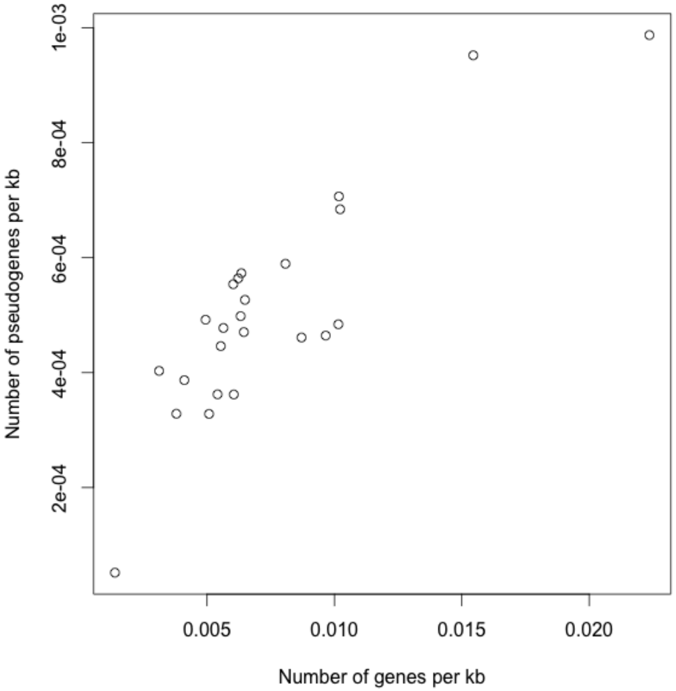
Exercise 7

```
# Graphical representation
pairs(huge[,2:5])
```



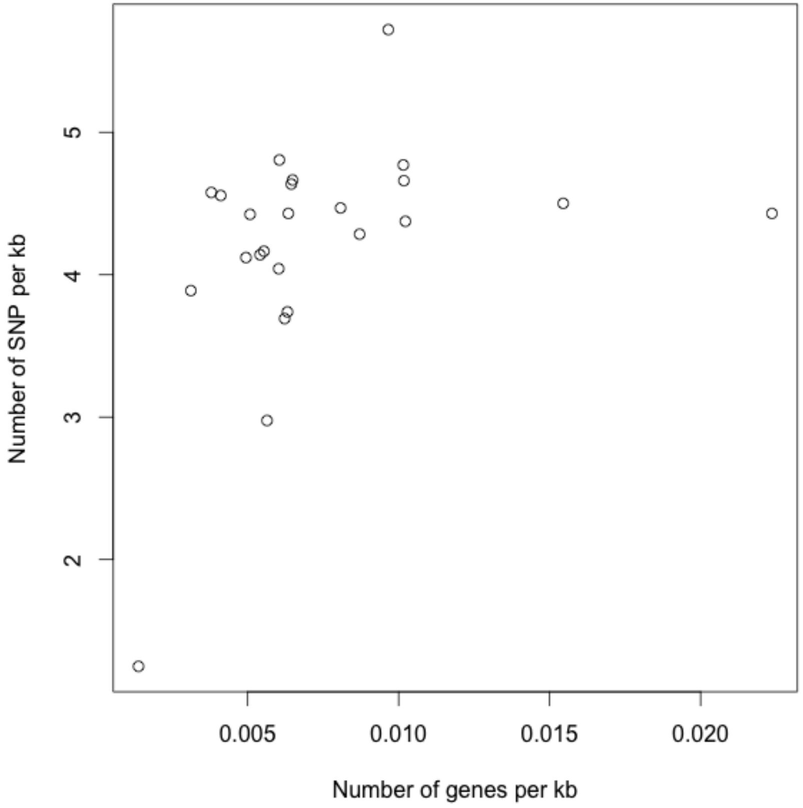
Exercise 7

```
# Calculation of the number of genes, pseudogenes and SNP per kb for each chromosome
hugeNorm = huge[,3:5]/huge[,2]*1000

# Correlation between genes and pseudogenes numbers per kb
cor(hugeNorm$Protein.coding.genes, hugeNorm$Pseudogenes)
[1] 0.87249
plot(hugeNorm$Protein.coding.genes, hugeNorm$Pseudogenes,
xlab="Number of genes per kb",ylab="Number of pseudogenes per kb")
```



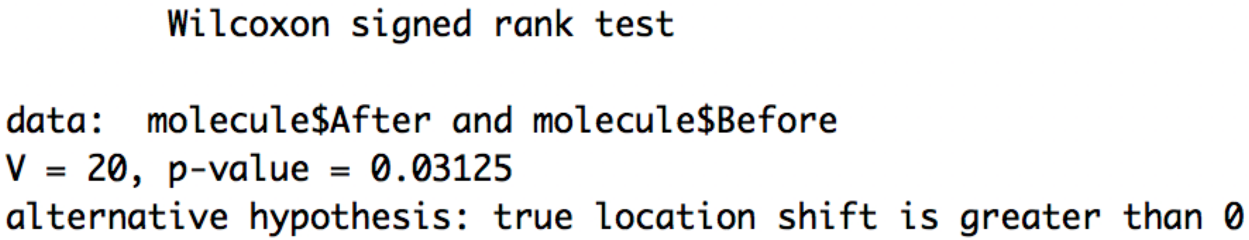
Exercise 7

```
# Correlation between SNP and genes numbers
cor(hugeNorm$Protein.coding.genes, hugeNorm$SNPs)
[1] 0.3849074
plot(hugeNorm$Protein.coding.genes, hugeNorm$SNPs,
xlab= "Number of genes per kb",ylab= "Number of SNP per kb")
```



Exercise 8

```
molecule = read.table("molecule.txt", header=TRUE, sep="\t")

# Wilcoxon test
# H0 : the blood level of this molecule does not increase significantly after the treatment
# H1 : the blood level of this molecule increases significantly after the treatment
# Let α=0.05

wilcox.test(molecule$After, molecule$Before, alternative="greater", paired=TRUE)
```



```
        Wilcoxon signed rank test

data:  molecule$After and molecule$Before
V = 20, p-value = 0.03125
alternative hypothesis: true location shift is greater than 0
```

```
# Conclusion :
We reject H0, the blood level of this molecule significantly increases after the treatment
```