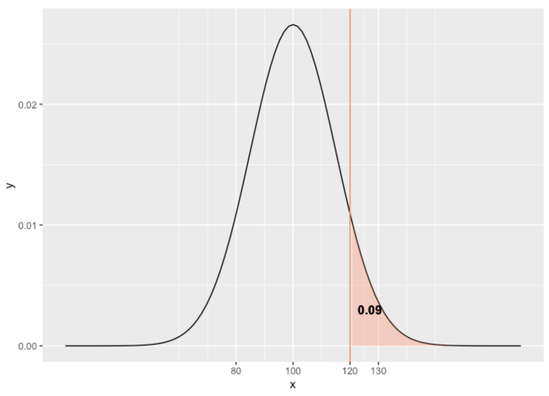
Fiche MAD 1

TP1 - Multivariate Normal and R

Knowing that IQ is a normal measure of mean 100 and standard deviation 15, what is the probability of having an IQ more than 120? less than 100?

pnorm(120, mean = 100, sd = 15, lower.tail = F, log.p = FALSE)
1 - pnorm(4/3)



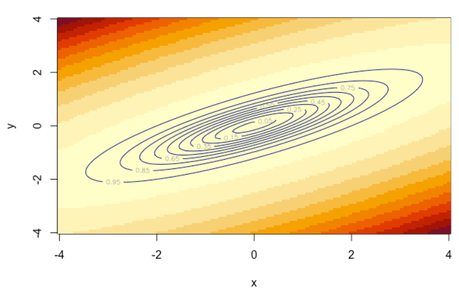
Show that the maximum likelihood estimator of the variance is biased and propose an unbiased estimator.

$$\begin{aligned} \mathrm{E}[\ ^2] &= E[\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2] \\ &= \mathrm{E}[\ \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \bar{X}^2] \\ &= \sigma^2 + \mu^2 - \frac{\sigma^2}{n} - \mu^2 \end{aligned}$$

Exceptionally large or small -> supp or less than parameters min or max try to find those parameters and then :
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width))+ geom$_p$*oint*(*colour* = *as.numeric*(*irisDOLLARSpecies*), *size* = *flower.outliers* ∗ 2 + 1) By doing that u will highlight the points that more or less bigger than usual values.

Generate 1000 observations of a two-dimensional normal distribution N(μ,$\sum$) with
$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} and \Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 0.75 \end{pmatrix}$.
After, draw the ellipses of equiprobability of the multiples of 5 percent. (Care abt * into percent*percent)

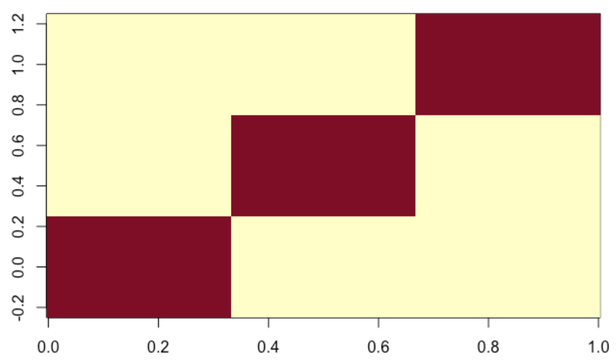
```
sigma<-matrix(c(2,1,1,0.75),2,2)
mu <- c(0,0)

cholesky-sigma =chol(sigma)
t(chol(sigma)) * chol(sigma)

Y<- matrix(rnorm(2000),1000,2) * chol(sigma) + mu
plot(Y,xlab="x",ylab="y",pch='.')

Q<-qchisq(p=seq(0.05,0.95,by=0.1),df=2)
x<-seq(-4,4,length=100)
y<-seq(-4,4,length=100)

sigmainv<-solve(sigma)

a<-sigmainv[1,1]
b<-sigmainv[2,2]
c<-sigmainv[1,2]
z<-outer(x,y,function(x,y) (a*x**2+b*y**2+2*c*x*y))
Fonction is t(y) * y
image(x,y,z)
contour(x,y,z,col="blue4",levels=Q,
labels=seq(from=0.05,to=0.95,by=0.1),add=T)
```



TP2 - Clustering

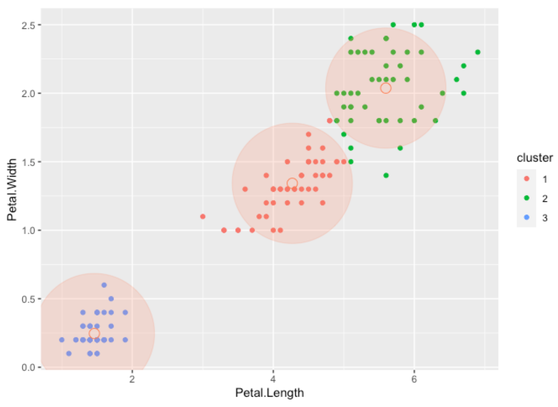
Clustering is statistique method in Data Analysis who organise values in group by their "degré de similitude". His objective is to identify and visualize sets of similar items in terms of define criterias.
Consider the Iris data set. Write a R code wich produces the partition matrix. Compute the gravity centers of the quantitative variables in the three classes using a matrix formula. With matrix multiplication P*P instead or percent*percent.

```
data(iris)
X<-iris[,1:4]
library(nnet)
C<-class.ind(irisDOLLARSpecies) Matrice partition
t((t(X) P*P C))/diag(t(C) P*P C)
image(C)
```



In this graphic we can see that there is 3 groups actually, this we lead us to our result. We need to understand why we do have 3 groups, there is actually 3 types of flowers so it s normal that there is 3 groups. We will use the kmeans method to try to plot our data :

```
kmeans.res <- iris P>P select(c(-Species,-Sepal.Length,-Sepal.Width)) P>P kmeans(3,nstart = 10)
cluster<-as.factor(kmeans.resDOLLARcluster)
centers <- as.data.frame(kmeans.resDOLLARcenters)
```



So we do get again 3 groups divided by commune values.

K-Means : With points and one int k, the algorithm aim to divide points in k group name clusters homogeneous and compact. We choose randomly k individuals in our values, and then we compare each points distance to all k chosen values. And then when all individuals have been looked at, we choose a new "gravity center" and we continue to look into our individuals whenever our center point doesn t move anymore.

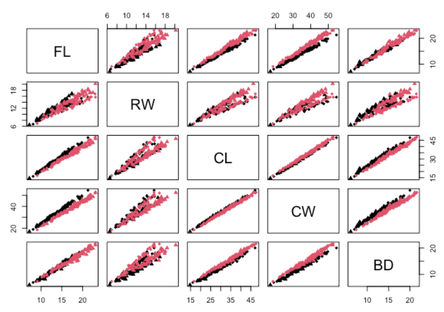
Strengths : Small calcul time / Easy Weakness : Need to know K / Two iterations might differ

Number of partition of n objects verifies :

$$\begin{aligned} \mathrm{B}_{n+1} &= \sum_{k=0}^{n} C_k^n B_k \,, \\ \mathrm{B}_0 &= 1 \end{aligned}$$

Clustering of crabs by sex :

```
pairs(crabs[,4:8], col=as.numeric(crabs$sp),pch=as.numeric(crabs$sex)+16)
```

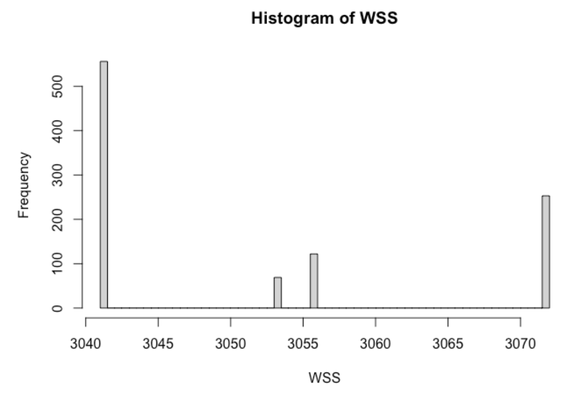


We now need to apply K-Means algorithm to this data with K = 4.

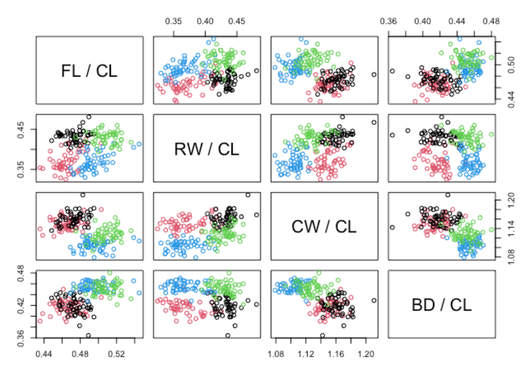
```
Kmeans.res<-crabs[,4:8] P>P kmeans(4,nstart = 1)
Kmeans.res
TrueClasses<-matrix(1:4,2,2)
colnames(TrueClasses)<-levels(crabsDOLLARsex)
rownames(TrueClasses)<-levels(crabsDOLLARsp)
TrueClasses=diag(TrueClasses[crabsDOLLARsex, crabsDOLLARsp])
table(Kmeans.resDOLLARcluster,TrueClasses)
```

This will allow us to get back a matrix KxK with some values. The within sum of squares :

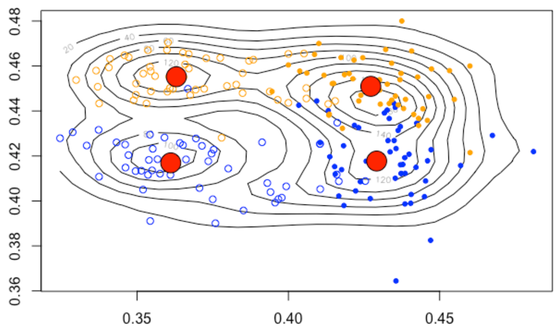
```
WSS = rep(0,1000)
for(i in 1:1000)
WSS[i] = (crabs[,4:8] P>P kmeans(4,nstart = 1))DOLLARtot.withinss
plot(WSS)
```



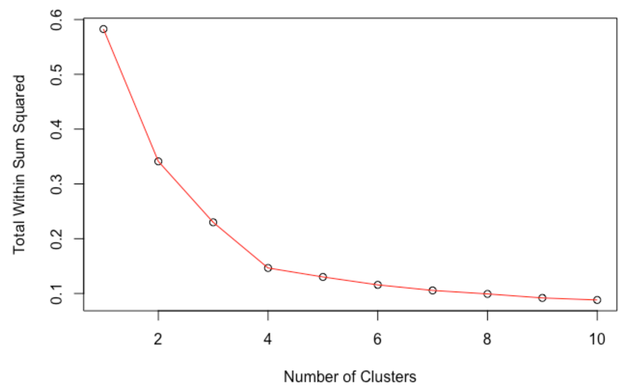
After that we can find our most correlated variable by doing a corrplot and look at the correlation matrix, when we got this variable we divide all other columns by this one and then we exclude this one. By doing this we should face a new dataset and get new values and new plot where we can continue to find new groups again.



We see here our 4 groups with one different color and for this plot one variable is missing. Our goal is now to represent those 4 groups into a 2 dimension plan :



They are still four groups, four points of interest, and they are still separate with sex criteria and with color criteria so 2 groups into 2 groups.
We are now looking into the relation between the number of groups in *K-Means* and the within sum of squares :



So the within sum of square is slowly becoming null when the number of clusters is evolving.