TP 2 : Lois continues et analyse en composantes principales

1 Lois usuelles continues

1.1 Densités

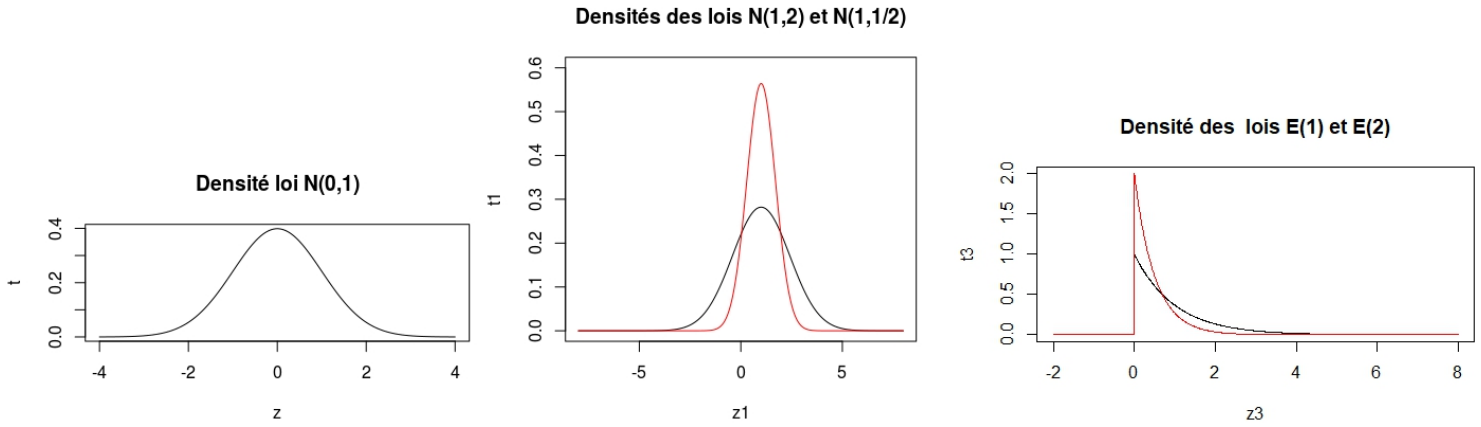
Tracer la densité de la loi $\mathcal{N}(0,1)$. Pour cela, on peut utiliser les commandes suivantes :

```
x <- seq(from=-4, to=4, by=0.01)
y <- dnorm(x, mean=0, sd=1)
plot(x, y, type='l')
```

Tracer de même la densité des lois suivantes :

(a) les lois normales $\mathcal{N}(1,2)$ (de moyenne 1 et variance 2) et $\mathcal{N}(1,0.5)$ (sur un même graphique) ;
(b) la loi exponentielle $\mathcal{E}(1)$ (quelle est sa dentité ?) ;
(c) la loi de Cauchy de paramètres 0 et 1 (quelle est sa dentité ?).

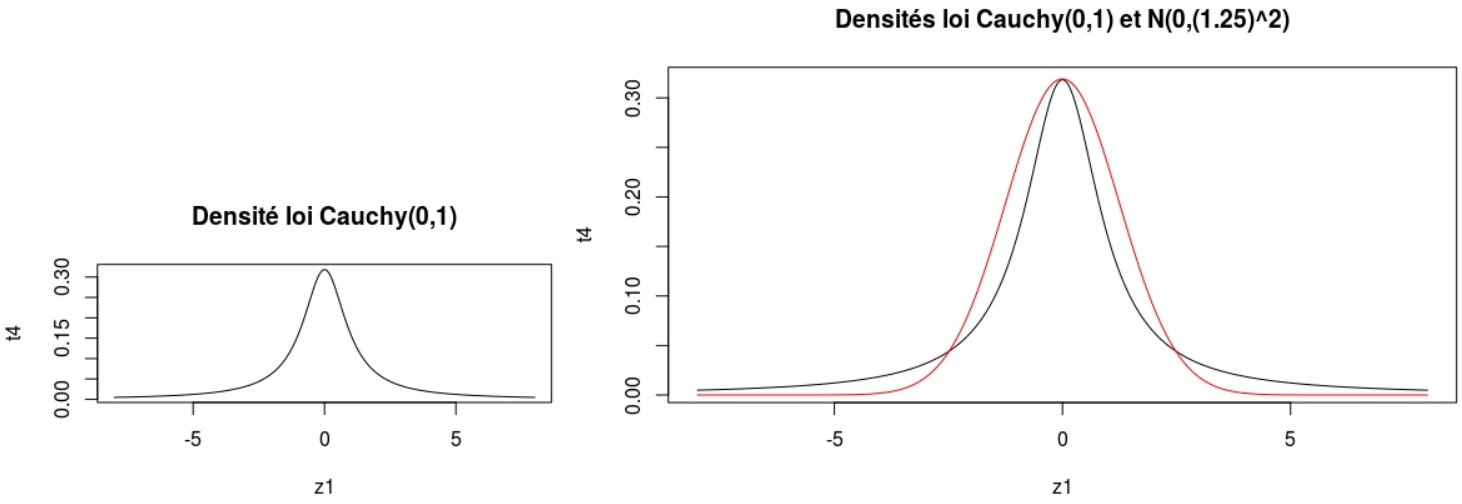
```
z <- seq(from=-4, to=4, by=0.01)

t <- dnorm(z, mean=0, sd=1)

plot(z, t, type='l',main="Densité loi N(0,1)")

z1 <- seq(from=-8, to=8, by=0.01)

t1 <- dnorm(z1, mean=1, sd=sqrt(2))

plot(z1, t1, type='l',main="Densités des lois N(1,2) et N(1,1/2)",ylim=c(0,0.6))

lines(z1, t2, type='l',col="red")
```



Dans le dernier graphique on a aussi tracé une loi normale centrée dont la densité en 0 a la même valeur que celle de la loi de Cauchy.

```
z3 <- seq(from=-2, to=8, by=0.01)

t3 <- dexp(z3, rate=1)
```

```
t3b <- dexp(z3, rate=2)

plot(z3, t3, type='l',main="Densités des lois E(1) et E(2)",ylim=c(0,2))

lines(z3, t3b, type='l',col="red")


t4 <- dcauchy(z1, location=0, scale= 1)

plot(z1, t4, type='l',main="Densité de la loi Cauchy(0,1)")

dnorm(0,0,(1:200)/100)/dcauchy(0)
#On trouve écart type 1.25 pour la valeur la plus proche de 1

t4 <- dcauchy(z1, location=0, scale= 1)

t5<- dnorm(z1, 0, 125/100)

plot(z1, t4, type='l',main="Densités loi Cauchy(0,1) et N(0,(1.25)^2)")

lines(z1, t5, type='l',col="red")
```

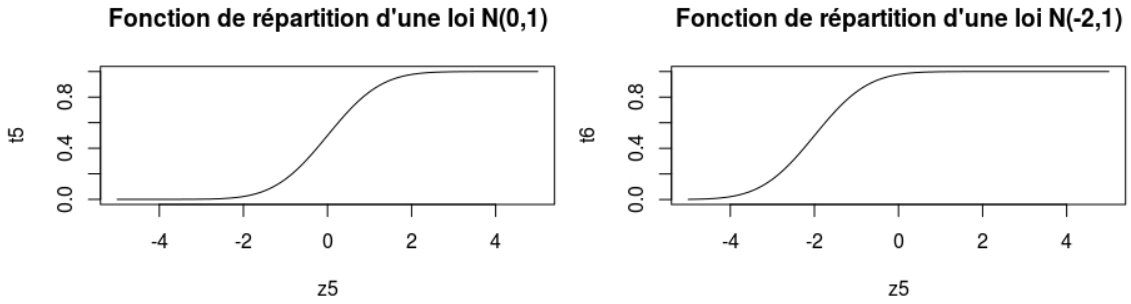


1.2 Fonctions de répartition

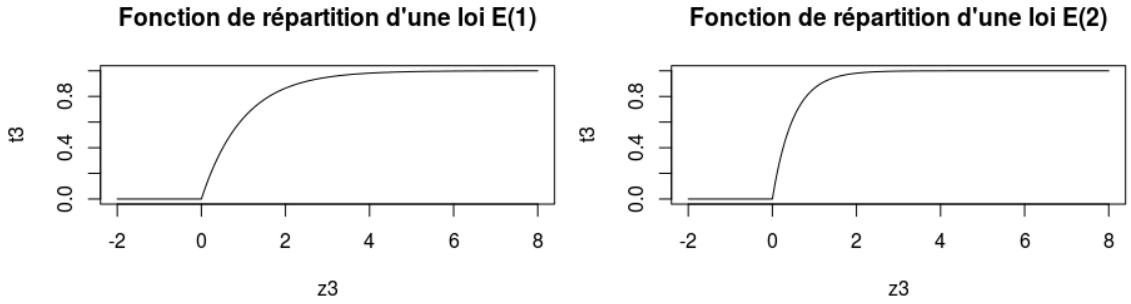
Tracer la fonction de répartition des lois $\mathcal{N}(0,1)$, $\mathcal{N}(-2,1)$ et $\mathcal{E}(2)$.

```
z5 <- seq(from=-5, to=5, by=0.01)

t5 <- pnorm(z5, mean=0, sd=1)

plot(z5, t5, type='l',main="Fonction de répartition d'une loi N(0,1)")
```

```
t6 <- pnorm(z5, mean=-2, sd=1)

plot(z5, t6, type='l',main="Fonction de répartition d'une loi N(-2,1)")
```




```
z3 <- seq(from=-2, to=8, by=0.01)

t3 <- pexp(z3, rate=2)

plot(z3, t3, type='l',main="Fonction de répartition d'une loi E(2)")
```



Pour X de loi $\mathcal{N}(0,1)$, calculer $P(X \leq 2)$, $P(-1 \leq X \leq 0.5)$ et trouver u tel que $P(X \leq u) = 0.9$.

```
pnorm(2, mean=0, sd=1)
[1] 0.9772499

 pnorm(1/2, mean=0, sd=1)-pnorm(-1, mean=0, sd=1)
[1] 0.5328072

 qnorm(0.9, mean=0, sd=1)
[1] 1.281552
```

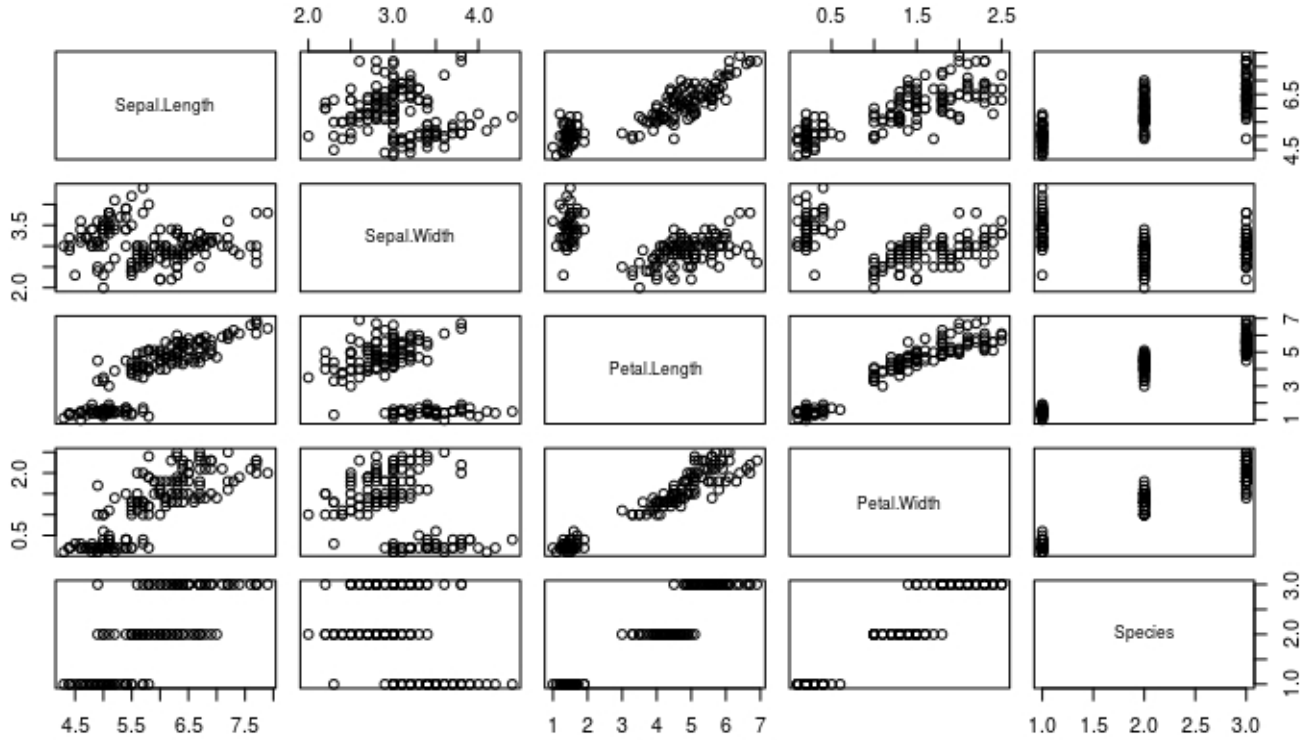
2 Analyse en composantes principales

On utilise le jeu de données `iris` inclus dans R. Cet échantillon décrit, pour 150 fleurs, la longueur et la largeur du sépale et du pétale, ainsi que l'espèce parmi trois possibles (lesquelles ?). Un des buts de l'étude est de séparer les trois espèces d'iris.

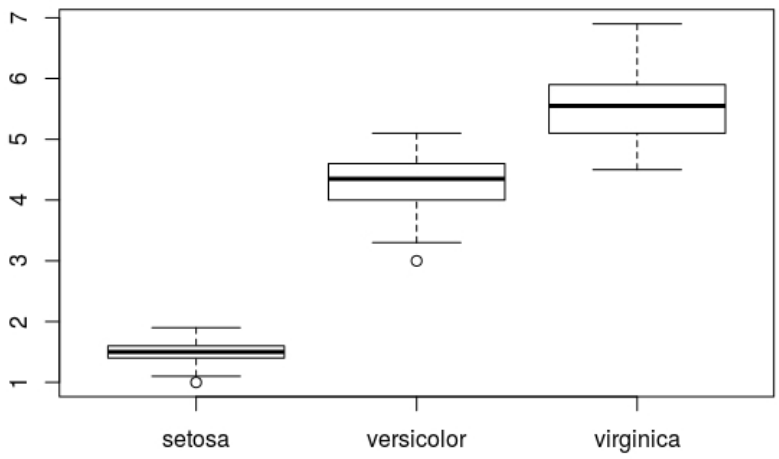
1. Visualisation préliminaire : **View**(iris) et **pairs**(iris). Calculer le résumé numérique : **summary**(iris).

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

Voici les nuages de point de chaque paire de variables, l'une en fonction de l'autre :



Tracer les *boxplots* par espèce avec : **boxplot**(iris$Petal.Length~iris$Species).



2. Évaluer les corrélations entre les quatre caractéristiques numériques (avec la fonction **cor**).

```
attach(iris)

DF<-data.frame(Sepal.Length,Sepal.Width,Petal.Length,Petal.Width)

cor(DF)
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

3. Évaluer l'échantillon des variables centrées réduites associées aux quatre différentes caractéristiques quantitatives (on pourra convertir en matrice A le **data.frame** et utiliser la commande suivante : **apply**(A,MARGIN=2,FUN=**sd**)).
   Méthode 1 :

```
DF2<-matrix(unlist(DF),ncol=4,nrow=150)
DFSD<-apply(DF2,MARGIN=2,FUN=sd);DFSD
[1] 0.8280661 0.4358663 1.7652982 0.7622377

DFM<-apply(DF2,MARGIN=2,FUN=mean);DFM
[1] 5.843333 3.057333 3.758000 1.199333

DFR<-t((t(DF2)-DFM)/DFSD);DFR

      [,1]       [,2]       [,3]        [,4]
[1,] -0.89767388  1.01560199 -1.33575163 -1.3110521482
[2,] -1.13920048 -0.13153881 -1.33575163 -1.3110521482
```

   La variable *DFR* donne la matrice des variables centrées réduites.
   Méthode 2 : DFR2<-**scale**(DF2);.
4. Évaluer la covariance de ces variables centrées réduites (avec la fonction **cov**). Comparer au résultat de la commande **cor**(iris [,1:4]) . Interpréter.
   On peut vérifier que la covariance est bien la corrélation d'origine.

```
cov(DFR)

      [,1]       [,2]       [,3]       [,4]
[1,]  1.0000000 -0.1175698  0.8717538  0.8179411
[2,] -0.1175698  1.0000000 -0.4284401 -0.3661259
[3,]  0.8717538 -0.4284401  1.0000000  0.9628654
[4,]  0.8179411 -0.3661259  0.9628654  1.0000000
```

5. Effectuer l'analyse en composantes principales. On rappelle qu'il faut :
   (a) trouver les valeurs propres de la matrices des corrélations **C**=**cov**(DFR) des variables. En déduire combien de composantes sont nécessaires pour capturer l'essentiel des données ;

```
EigenDF<-eigen(cor(DF2))
EigenDF$values
[1] 2.91849782 0.91403047 0.14675688 0.02071484

cumsum(EigenDF$values)/sum(EigenDF$values)
[1] 0.7296245 0.9581321 0.9948213 1.0000000
```

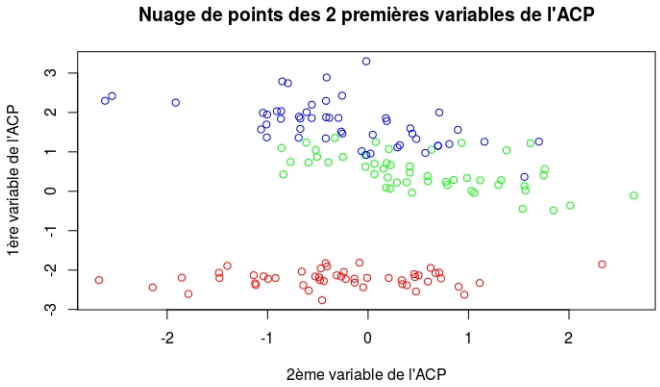
   En faisant les sommes cumulées des valeurs propres divisées par la somme totale, on voit que la première variable capture déjà 73% de la variance des données, soit juste en dessous du seuil "usuel" de 75%. Les 2 premières variables capturent 95%>75%. Il suffit donc de 2 variables.

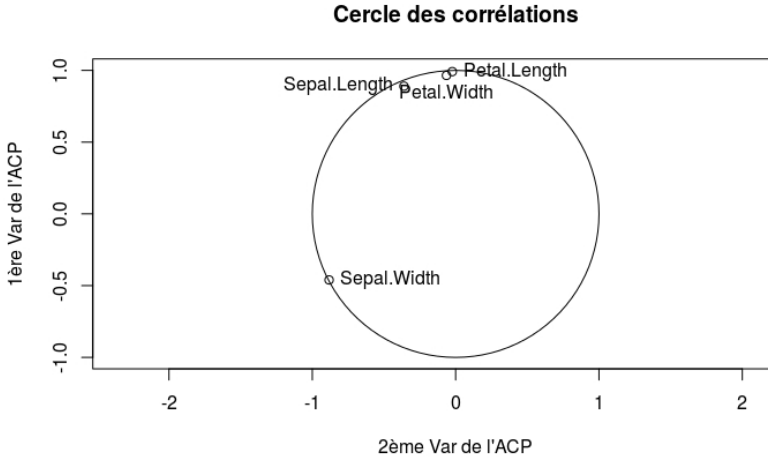
(b) trouver la matrice de passage M de la base initiale vers la base de diagonalisation de **C** (c'est la matrice dont les colonnes sont les vecteurs propres de **C**) ;

```
M<-EigenDF$vectors;M
     [,1]       [,2]       [,3]       [,4]
[1,]  0.5210659 -0.37741762  0.7195664  0.2612863
[2,] -0.2693474 -0.92329566 -0.2443818 -0.1235096
[3,]  0.5804131 -0.02449161 -0.1421264 -0.8014492
[4,]  0.5648565 -0.06694199 -0.6342727  0.5235971
```

(c) obtenir les données[1] dans la nouvelle base des composantes principales et tracer le nouveau nuage, si possible en trois couleurs (une par espèce ; option **col**='green' pour tracer en vert) ;

```
NewDF<-DFR%*% M;NewDF

     [,1]        [,2]       [,3]        [,4]
[1,] -2.25714118 -0.478423832  0.127279624  0.024087508
[2,] -2.07401302  0.671882687  0.233825517  0.102662845
...

plot(NewDF[,1]~NewDF[,2],ylab="1ère variable de l'ACP",
xlab="2ème variable de l'ACP",
main="Nuage de points des 2 premières variables de l'ACP",
col=rep(c('red','green','blue'),each=50))
```





(d) ceci fait, représenter par un nuage de points les deux composantes principales de l'ACP puis le cercle des corrélations.Quelles sont les variables de départ bien représentées par ces deux composantes ?

La longueur et la largeur des pétales sont représentées par la première variable et aussi la longueur des sépales (à peine moins bien). La largeur des sépales est représentée par un mélange de la deuxième et de la première variable (avec une corrélation négative à la première variable). Tous les points sont très proche du cercle, dont toutes les variables sont très bien représentées par l'ensemble des 2 variables principales.

```
install.packages("plotrix");library("plotrix")
MixCor<-cor(DFR,NewDF)
MixCor

     [,1]       [,2]       [,3]       [,4]
[1,]  0.8901688 -0.36082989  0.27565767  0.03760602
```

1. Rappel : agir sur les colonnes d'une matrice, c'est multiplier *à droite* par une matrice : par M ou par M^{-1} ?

```
[2,] -0.4601427 -0.88271627 -0.09361987 -0.01777631
[3,]  0.9915552 -0.02341519 -0.05444699 -0.11534978
[4,]  0.9649790 -0.06399985 -0.24298265  0.07535950
```

```
plot(MixCor[,1]~MixCor[,2],ylim=c(-1,1),xlim=c(-1,1),
ylab="1ère Var de l'ACP",xlab="2ème Var de l'ACP",
asp = 1,main="Cercle des corrélations")

text(x=MixCor[1:4,2],y=MixCor[1:4,1],colnames(DF[1:4]),pos=c(2,4,4,1));
draw.circle(0,0,1)
```

6. Calculer de deux façons la variance empirique des deux composantes principales. **cov**(NewDF[,1:2]) donne cette matrice, et c'est la même réponse que la matrice diagonale avec sur la diagonale les valeurs propres de la matrice de covariance

```
diag(eigen(cov(NewDF[,1:2]))$values)
```

```
     [,1]      [,2]
[1,] 2.918498 0.0000000
[2,] 0.000000 0.9140305
```