Devoir_2

EL_Hadrami

10/11/2020

```
library(tidyverse)
```

```
## -- Attaching packages -------------------------------------------------------------------------------

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts -------------------------------------------------------------------------------- ti
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Exercice.13

```
# Creation d'un data-frame "acteur"
Mort.à <- c(93,53,72,68,68,53)
Années.de.carrière <- c(66,25,48,37,31,32)
Nombre.de.films <- c(211,58,98,140,74,81)
Prénom <- c("Michel","André","Jean","Louis","Lino","Jacques")
Nom <- c("Galabru","Raimbourg","Gabin","De Funès","Ventura","Villeret")
Date.du.décès <- c("04-01-2016","23-09-1970","15-10-1976","27-01-1983","22-10-1987","28-01-2005")
data.acteur <- data.frame(Mort.à,Années.de.carrière,Nombre.de.films,Prénom,Nom,Date.du.décès)
 #utilisation d'un dplyer pour renommer la premiere variable
data.acteur.r <- rename(data.acteur,"Age.du.décès"=Mort.à)
# extraction de la colonne Prénom
prenom.extract <- data.acteur$Prénom
data.acteur.arrange <- arrange(data.acteur.r,Age.du.décès)
```

Exercice.14

Question 1.

```
w <-read.delim(file="/home/ndoye/M1_SSD/Analyse_données/TP_M1MIASHS_AD/TP_2/data/fromages1-TP-M1.txt")
```

Question 2.

```
w <- rename(w,"mean.score"=Y,"c.a.a"=X1,"c.h.s"=X2,"c.a.l"=X3)
w$X1
```

```
## NULL
```

Question 3.:Les caracteristiques de w:

```
print(w)
```

```
##    mean.score c.a.a  c.h.s c.a.l
## 1        12.3 4.543  3.135  0.86
```

```
## 2         20.9 5.159  5.043  1.53
## 3         39.0 5.366  5.438  1.57
## 4         47.9 5.759  7.496  1.81
## 5          5.6 4.663  3.807  0.99
## 6         25.9 5.697  7.601  1.09
## 7         37.3 5.892  8.726  1.29
## 8         21.9 6.078  7.966  1.78
## 9         18.1 4.898  3.850  1.29
## 10        21.0 5.242  4.174  1.58
## 11        34.9 5.740  6.142  1.68
## 12        57.2 6.446  7.908  1.90
## 13         0.7 4.477  2.996  1.06
## 14        25.9 5.236  4.942  1.30
## 15        54.9 6.151  6.752  1.52
## 16        40.9 6.365  9.588  1.74
## 17        15.9 4.787  3.912  1.16
## 18         6.4 5.412  4.700  1.49
## 19        18.0 5.247  6.174  1.63
## 20        38.9 5.438  9.064  1.99
## 21        14.0 4.564  4.949  1.15
## 22        15.2 5.298  5.220  1.33
## 23        32.0 5.455  9.242  1.44
## 24        56.7 5.855 10.199  2.01
## 25        16.8 5.366  3.664  1.31
## 26        11.6 6.043  3.219  1.46
## 27        26.5 6.458  6.962  1.72
## 28         0.7 5.328  3.912  1.25
## 29        13.4 5.802  6.685  1.08
## 30         5.5 6.176  4.787  1.25
```

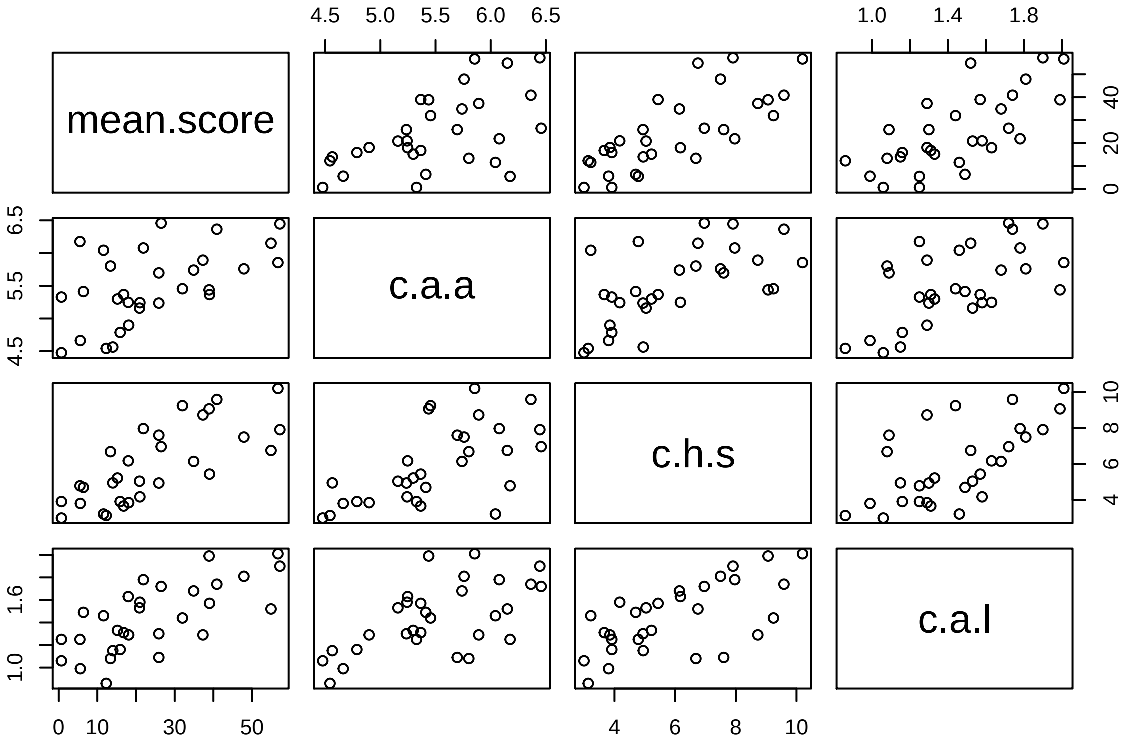
Question 4.:Les parametres statistiques des variables

```r
summary(w)
```

```
##    mean.score         c.a.a          c.h.s            c.a.l
##  Min.   : 0.70   Min.   :4.477   Min.   : 2.996   Min.   :0.860
##  1st Qu.:13.55   1st Qu.:5.237   1st Qu.: 3.978   1st Qu.:1.250
##  Median :20.95   Median :5.425   Median : 5.329   Median :1.450
##  Mean   :24.53   Mean   :5.498   Mean   : 5.942   Mean   :1.442
##  3rd Qu.:36.70   3rd Qu.:5.883   3rd Qu.: 7.575   3rd Qu.:1.667
##  Max.   :57.20   Max.   :6.458   Max.   :10.199   Max.   :2.010
```

Question 5.:

```r
pairs(w)
```



la commande pairs permet de tracer une nuage de point pour chaque variable afin de voir les differentes correlations qui peuvent exister

Question 6.:Construction d'une nouvelle data frame

```
nv.c.a.a <- c(w$c.a.a[w$c.a.a > 5.1],rep(NA,6))
nv.c.a.l <- c(w$c.a.l[w$c.a.l < 1.77],rep(NA,5))
ww <- data.frame(w$mean.score,nv.c.a.a,w$c.h.s,nv.c.a.l)
ww <- rename(ww,"mean.score"=w.mean.score,"c.a.a"=nv.c.a.a,"c.h.s"=w.c.h.s,"c.a.l"=nv.c.a.l)
```

Question 7.: Les caracteristiques de ww

```
print(ww)
```

```
##    mean.score c.a.a  c.h.s c.a.l
## 1        12.3 5.159  3.135  0.86
## 2        20.9 5.366  5.043  1.53
## 3        39.0 5.759  5.438  1.57
## 4        47.9 5.697  7.496  0.99
## 5         5.6 5.892  3.807  1.09
## 6        25.9 6.078  7.601  1.29
## 7        37.3 5.242  8.726  1.29
## 8        21.9 5.740  7.966  1.58
## 9        18.1 6.446  3.850  1.68
## 10       21.0 5.236  4.174  1.06
## 11       34.9 6.151  6.142  1.30
## 12       57.2 6.365  7.908  1.52
## 13        0.7 5.412  2.996  1.74
## 14       25.9 5.247  4.942  1.16
## 15       54.9 5.438  6.752  1.49
## 16       40.9 5.298  9.588  1.63
## 17       15.9 5.455  3.912  1.15
```

```
## 18        6.4 5.855  4.700  1.33
## 19       18.0 5.366  6.174  1.44
## 20       38.9 6.043  9.064  1.31
## 21       14.0 6.458  4.949  1.46
## 22       15.2 5.328  5.220  1.72
## 23       32.0 5.802  9.242  1.25
## 24       56.7 6.176 10.199  1.08
## 25       16.8    NA  3.664  1.25
## 26       11.6    NA  3.219    NA
## 27       26.5    NA  6.962    NA
## 28        0.7    NA  3.912    NA
## 29       13.4    NA  6.685    NA
## 30        5.5    NA  4.787    NA
```

Question 8.: Les parametres statistiques de la variable ww

```
summary(ww)
```

```
##    mean.score        c.a.a          c.h.s            c.a.l
##  Min.   : 0.70   Min.   :5.159   Min.   : 2.996   Min.   :0.860
##  1st Qu.:13.55   1st Qu.:5.356   1st Qu.: 3.978   1st Qu.:1.160
##  Median :20.95   Median :5.718   Median : 5.329   Median :1.310
##  Mean   :24.53   Mean   :5.709   Mean   : 5.942   Mean   :1.351
##  3rd Qu.:36.70   3rd Qu.:6.052   3rd Qu.: 7.575   3rd Qu.:1.530
##  Max.   :57.20   Max.   :6.458   Max.   :10.199   Max.   :1.740
##                  NA's   :6                        NA's   :5
```

<u>Exercice 15.</u>

1.

```
df.airquality <- airquality
```

2.Affichage des noms des variables

```
names(df.airquality)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

3.nombre de ligne et de colonne

```
nrow(df.airquality)
```

```
## [1] 153
```

```
ncol(df.airquality)
```

```
## [1] 6
```

4.Les parametres statistiques

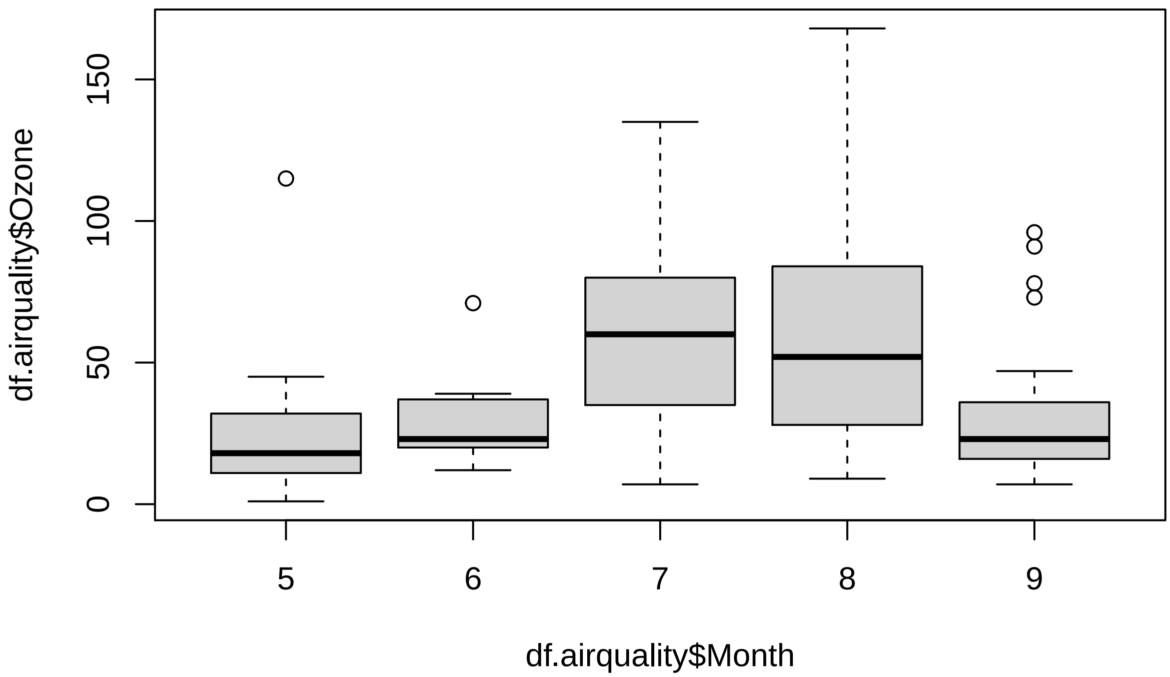
```
summary(df.airquality)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
```

```
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

5.representation de la boite a moustache

```
boxplot(df.airquality$Ozone~df.airquality$Month)
```



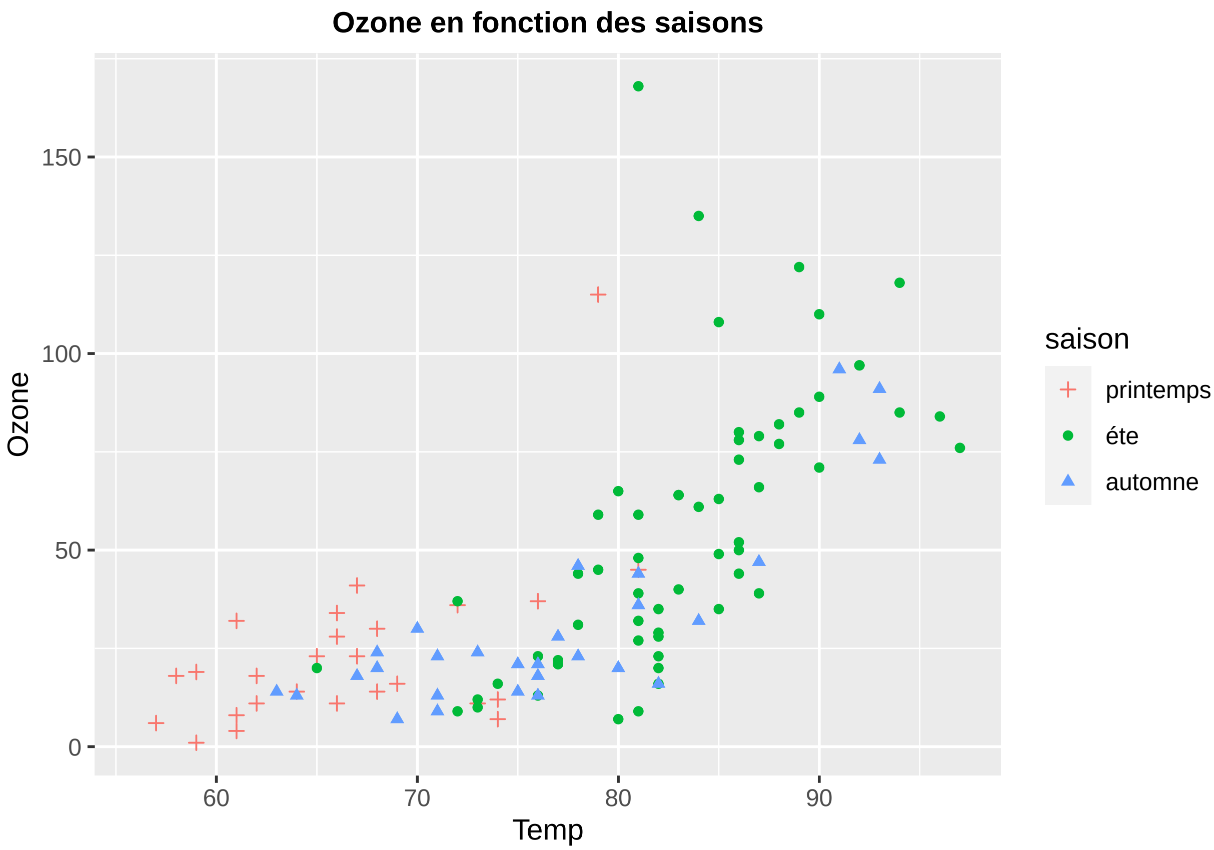
6.Creation d'une variable qualitative "saison"

```
saison <-factor(df.airquality$Month,levels=c(5:9))
levels(saison)[levels(saison)==5] <- "printemps"
levels(saison)[levels(saison)==6] <- "éte"
levels(saison)[levels(saison)==7] <- "éte"
levels(saison)[levels(saison)==8] <- "éte"
levels(saison)[levels(saison)==9] <- "automne"
df.airquality$season = saison
```

7.

```
g <- ggplot(data = df.airquality) +
  geom_point(mapping = aes(x =Temp, y = Ozone, shape = saison,color=saison)) + scale_shape_manual(values
  ggtitle("Ozone en fonction des saisons")
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```



<u>Exercice 16</u>

1.Simulation de 100 valeurs suivant une loi normale

```
n <- 100
e <- rnorm(n,0,25)
```

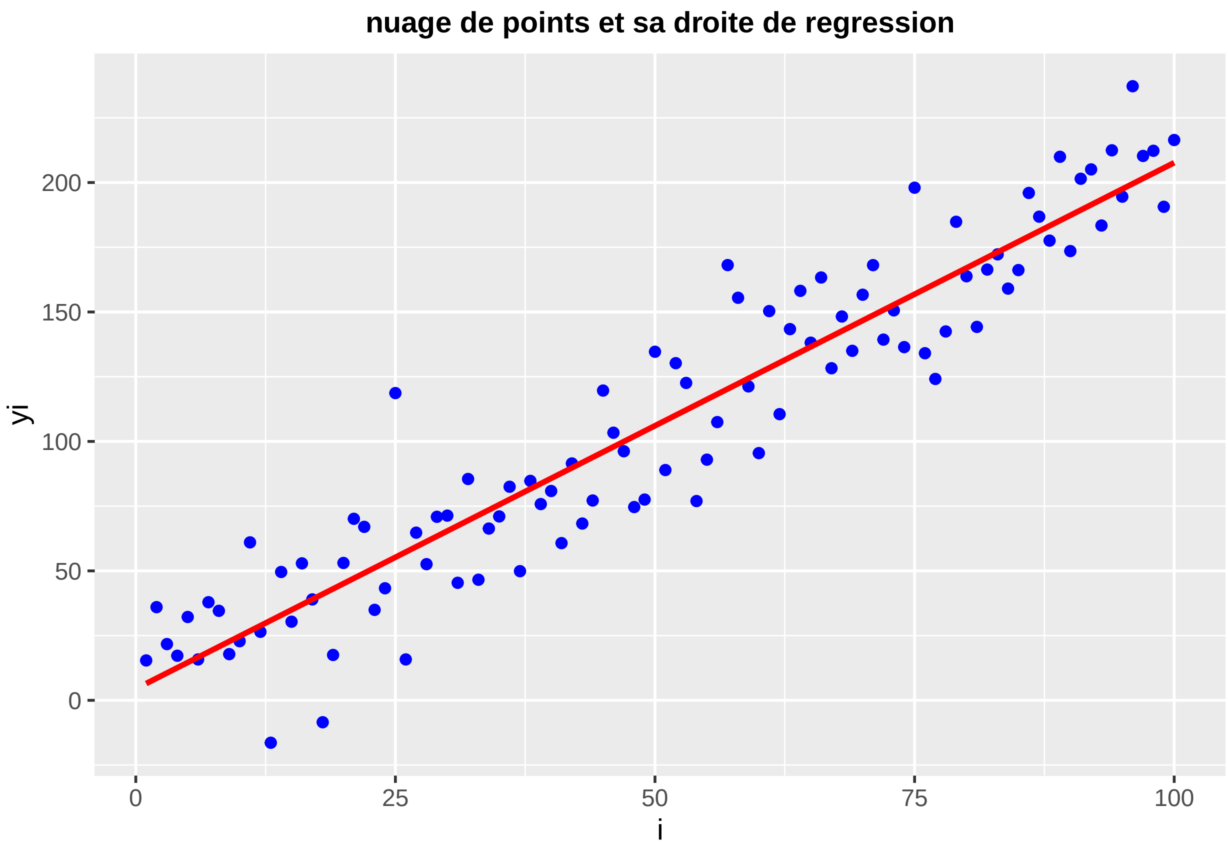
2. Pour tout $i \in 1, \dots, 100$,on pose $y_i = 1.7 + 2.1i + e_i$

```
i <- c(1:100)
yi <- 1.7 + 2.1 * i + e[i]
```

2.a et 2.b: representation d'un nuage de point et une droite (i, y_i)

```
data.f <- data.frame(i,yi)
g <- ggplot(data = data.f) +
geom_point(mapping = aes(x = i, y = yi),colour="blue") +
geom_smooth(mapping = aes(x = i, y = yi),se = FALSE,colour="red",fill="red",method = lm) +
ggtitle("nuage de points et sa droite de regression")
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



<u>Exercice 17</u>

```
# Creation d'une matrice
ligne1 <- c(68,119,26,7)
ligne2 <- c(15,54,14,10)
ligne3 <- c(5,29,14,16)
ligne4 <- c(20,84,17,94)
mat <- matrix(c(ligne1,ligne2,ligne3,ligne4),nrow = 4,ncol=4,byrow = T,dimnames = list(c("marron","nois
```

2. Calculer la matrice des fréquences (arrondit au 100ème près)

```
# conversion de la matrice "mat" en table
tab <- as.table(mat)
```

matrice de frequences

```
freq <- round(prop.table(tab)*100,2)
```

3. les lois marginales (nommer c pour le vecteur colonne et r pour le vecteur ligne)

```
l <- round(prop.table(tab,margin = 1),2)
c <- round(prop.table(tab,margin = 2),2) # pour le vecteur colonne
```

4.Profils lignes

```
L <- round(sweep(mat,1,rowSums(mat),'/'),2) * 100
```

5.Profils colonnes

```
C <- round(sweep(mat,2,colSums(mat),'/'),2) * 100
```

6. La distance de chi-deux entre les profils lignes

```r
d.chi <- 0
distancechideux <- function(L){
  for(i in 1:nrow(L)){

    d.chi <- sum(d.chi + (mat[i,] - L[i,]) ^ 2 / L[i,])
  }
  return (d.chi)
}
```

Exercice 18

```r
tableau <- matrix(c(290,410,110,190), ncol=2, byrow=TRUE)
colnames(tableau) <- c("Bleu","Brun")
rownames(tableau) <- c("Celib","Marie")
tableau <- as.table(tableau)
```

```r
print(tableau)
```

```
##       Bleu Brun
## Celib  290  410
## Marie  110  190
```

```r
barplot(tableau)
```