Projet STA203 - Apprentissage statistique

Anthony Kalaydjian - Mathieu Occhipinti

2023-04-29

```
rm(list=ls())
setwd(getwd())
library(ggplot2)
set.seed(150)
```

Introduction

Partie I

Analyse descriptive

On commence par importer les données et regarder de manière générale de quoi est composé notre jeu de donnée.

```
setwd(getwd())
data <- read.csv("Music_2023.txt",sep=";",header=TRUE)

dim(data)
```

```
## [1] 4278  192
```

```
n <- nrow(data)
p <- ncol(data)
```

Les dimmensions du dataset importé sont correctes. Il y a bien 192 variables pour 4278 vecteurs de données.

```
summary(data)
```

```
##      PAR_TC           PAR_SC          PAR_SC_V         PAR_ASE1
##     PAR_ASE2          PAR_ASE3          PAR_ASE4          PAR_ASE5
##     PAR_ASE6          PAR_ASE7          PAR_ASE8          PAR_ASE9
##    PAR_ASE10         PAR_ASE11         PAR_ASE12         PAR_ASE13
##    PAR_ASE14         PAR_ASE15         PAR_ASE16         PAR_ASE17
##    PAR_ASE18         PAR_ASE19         PAR_ASE20         PAR_ASE21
##    PAR_ASE22         PAR_ASE23         PAR_ASE24         PAR_ASE25
##    PAR_ASE26         PAR_ASE27         PAR_ASE28         PAR_ASE29
##    PAR_ASE30         PAR_ASE31         PAR_ASE32         PAR_ASE33
```

```
##    PAR_ASE34         PAR_ASE_M         PAR_ASEV1           PAR_ASEV2
##    PAR_ASEV3           PAR_ASEV4           PAR_ASEV5
##    PAR_ASEV6           PAR_ASEV7           PAR_ASEV8
##    PAR_ASEV9           PAR_ASEV10          PAR_ASEV11
##    PAR_ASEV12          PAR_ASEV13          PAR_ASEV14
##    PAR_ASEV15          PAR_ASEV16          PAR_ASEV17
##    PAR_ASEV18          PAR_ASEV19          PAR_ASEV20
##    PAR_ASEV21          PAR_ASEV22          PAR_ASEV23
##    PAR_ASEV24          PAR_ASEV25          PAR_ASEV26
##    PAR_ASEV27          PAR_ASEV28          PAR_ASEV29
##    PAR_ASEV30          PAR_ASEV31          PAR_ASEV32
##    PAR_ASEV33          PAR_ASEV34          PAR_ASE_MV          PAR_ASC
##    PAR_ASC_V           PAR_ASS           PAR_ASS_V           PAR_SFM1
##     PAR_SFM2            PAR_SFM3            PAR_SFM4          PAR_SFM5
##     PAR_SFM6           PAR_SFM7           PAR_SFM8           PAR_SFM9
##    PAR_SFM10          PAR_SFM11          PAR_SFM12          PAR_SFM13
##    PAR_SFM14          PAR_SFM15          PAR_SFM16          PAR_SFM17
##    PAR_SFM18         PAR_SFM19          PAR_SFM20         PAR_SFM21
##    PAR_SFM22         PAR_SFM23          PAR_SFM24         PAR_SFM_M
##    PAR_SFMV1           PAR_SFMV2           PAR_SFMV3
##    PAR_SFMV4           PAR_SFMV5           PAR_SFMV6
##    PAR_SFMV7           PAR_SFMV8           PAR_SFMV9
##    PAR_SFMV10          PAR_SFMV11          PAR_SFMV12
##    PAR_SFMV13          PAR_SFMV14          PAR_SFMV15
##    PAR_SFMV16          PAR_SFMV17          PAR_SFMV18
##    PAR_SFMV19          PAR_SFMV20          PAR_SFMV21
##    PAR_SFMV22          PAR_SFMV23          PAR_SFMV24
##    PAR_SFM_MV          PAR_MFCC1          PAR_MFCC2          PAR_MFCC3
##    PAR_MFCC4          PAR_MFCC5           PAR_MFCC6          PAR_MFCC7
##    PAR_MFCC8          PAR_MFCC9          PAR_MFCC10         PAR_MFCC11
##    PAR_MFCC12         PAR_MFCC13          PAR_MFCC14          PAR_MFCC15
##    PAR_MFCC16          PAR_MFCC17         PAR_MFCC18         PAR_MFCC19
##    PAR_MFCC20         PAR_MFCCV1        PAR_MFCCV2         PAR_MFCCV3
##    PAR_MFCCV4         PAR_MFCCV5           PAR_MFCCV6         PAR_MFCCV7
##    PAR_MFCCV8         PAR_MFCCV9        PAR_MFCCV10        PAR_MFCCV11
##   PAR_MFCCV12        PAR_MFCCV13         PAR_MFCCV14         PAR_MFCCV15
##   PAR_MFCCV16         PAR_MFCCV17        PAR_MFCCV18        PAR_MFCCV19
##   PAR_MFCCV20       PAR_THR_1RMS_TOT  PAR_THR_2RMS_TOT    PAR_THR_3RMS_TOT
##  PAR_THR_1RMS_10FR_MEAN PAR_THR_1RMS_10FR_VAR PAR_THR_2RMS_10FR_MEAN
##  PAR_THR_2RMS_10FR_VAR PAR_THR_3RMS_10FR_MEAN PAR_THR_3RMS_10FR_VAR
##  PAR_PEAK_RMS_TOT PAR_PEAK_RMS10FR_MEAN PAR_PEAK_RMS10FR_VAR    PAR_ZCD
##   PAR_1RMS_TCD       PAR_2RMS_TCD        PAR_3RMS_TCD      PAR_ZCD_10FR_MEAN
##  PAR_ZCD_10FR_VAR    PAR_1RMS_TCD_10FR_MEAN PAR_1RMS_TCD_10FR_VAR
##  PAR_2RMS_TCD_10FR_MEAN PAR_2RMS_TCD_10FR_VAR PAR_3RMS_TCD_10FR_MEAN
##  PAR_3RMS_TCD_10FR_VAR    GENRE
##  [ getOption("max.print") est atteint -- 6 lignes omises ]
```

```
## A FAIRE : Analyse uni-bi variée

##Question : Comment choisir les variables qu'on observe ?
```

```
# Proportion des genres musicaux
freq<-count(data,'GENRE')
```

```
freq
```

```
##   "GENRE"    n
## 1   GENRE 4278
```

```
prop_classical<-freq[1,2]/n
prop_jazz<-freq[2,2]/n

prop_classical
```

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

```
prop_jazz
```

```
## [1] NA
```

```
#ggplot(data, aes(x=reorder(GENRE, GENRE, function(x)-(length(x)/n)))) +
#geom_bar(fill='red') +  labs(x='Genre')


summary(data[,0:20])
```

```
##      PAR_TC           PAR_SC          PAR_SC_V          PAR_ASE1
##  Min.   :0.8377   Min.   :  34.1   Min.   :    605   Min.   :-0.2225
##  1st Qu.:2.3254   1st Qu.: 450.4   1st Qu.:  22593   1st Qu.:-0.1721
##  Median :2.4996   Median : 589.8   Median :  46062   Median :-0.1571
##  Mean   :2.4817   Mean   : 632.6   Mean   : 105223   Mean   :-0.1557
##  3rd Qu.:2.6437   3rd Qu.: 766.9   3rd Qu.:  99092   3rd Qu.:-0.1403
##  Max.   :4.4046   Max.   :3044.4   Max.   :5003700   Max.   :-0.0533
##     PAR_ASE2          PAR_ASE3          PAR_ASE4          PAR_ASE5
##  Min.   :-0.22612   Min.   :-0.22728   Min.   :-0.22387   Min.   :-0.22444
##  1st Qu.:-0.17257   1st Qu.:-0.16659   1st Qu.:-0.15751   1st Qu.:-0.15010
##  Median :-0.15761   Median :-0.14988   Median :-0.13994   Median :-0.13323
##  Mean   :-0.15529   Mean   :-0.14902   Mean   :-0.14072   Mean   :-0.13520
##  3rd Qu.:-0.13884   3rd Qu.:-0.13165   3rd Qu.:-0.12325   3rd Qu.:-0.11791
##  Max.   :-0.05673   Max.   :-0.05856   Max.   :-0.06055   Max.   :-0.06524
##     PAR_ASE6          PAR_ASE7          PAR_ASE8          PAR_ASE9
##  Min.   :-0.22526   Min.   :-0.21973   Min.   :-0.22003   Min.   :-0.20936
##  1st Qu.:-0.14389   1st Qu.:-0.14104   1st Qu.:-0.13740   1st Qu.:-0.13584
##  Median :-0.12965   Median :-0.12812   Median :-0.12569   Median :-0.12458
##  Mean   :-0.13162   Mean   :-0.13030   Mean   :-0.12736   Mean   :-0.12557
##  3rd Qu.:-0.11625   3rd Qu.:-0.11649   3rd Qu.:-0.11485   3rd Qu.:-0.11373
##  Max.   :-0.06715   Max.   :-0.07032   Max.   :-0.06982   Max.   :-0.06812
##    PAR_ASE10          PAR_ASE11          PAR_ASE12          PAR_ASE13
##  Min.   :-0.20371   Min.   :-0.19532   Min.   :-0.18544   Min.   :-0.19821
##  1st Qu.:-0.13388   1st Qu.:-0.13159   1st Qu.:-0.13081   1st Qu.:-0.13237
##  Median :-0.12289   Median :-0.12133   Median :-0.12116   Median :-0.12172
##  Mean   :-0.12287   Mean   :-0.12135   Mean   :-0.12079   Mean   :-0.12226
##  3rd Qu.:-0.11129   3rd Qu.:-0.11013   3rd Qu.:-0.11047   3rd Qu.:-0.11110
##  Max.   :-0.06098   Max.   :-0.05382   Max.   :-0.05741   Max.   :-0.06622
##    PAR_ASE14          PAR_ASE15          PAR_ASE16          PAR_ASE17
```

```
##  Min.   :-0.17919   Min.   :-0.18552   Min.   :-0.18993   Min.   :-0.20037
##  1st Qu.:-0.13284   1st Qu.:-0.13726   1st Qu.:-0.14191   1st Qu.:-0.14447
##  Median :-0.12124   Median :-0.12668   Median :-0.13256   Median :-0.13446
##  Mean   :-0.12198   Mean   :-0.12728   Mean   :-0.13274   Mean   :-0.13535
##  3rd Qu.:-0.11027   3rd Qu.:-0.11721   3rd Qu.:-0.12248   3rd Qu.:-0.12540
##  Max.   :-0.06047   Max.   :-0.06275   Max.   :-0.07463   Max.   :-0.06379
```

```
summary(data$PAR_SC_V)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     605   22593   46062  105223   99092 5003700
```

```
summary(data$PAR_SC)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    34.1   450.4   589.8   632.6   766.9  3044.4
```

```
summary(data$PAR_ASC_V)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005926 0.162052 0.290915 0.425129 0.528360 3.506300
```

On remarque que les variables PAR_SC_V et PAR_SC ont des ordres de grandeurs bien supérieurs à celui des autres variables. Il est donc judicieux d'appliquer une transformation log afin d'obtenir des ordres de grandeurs similaires entre toutes nos variables explicatives.

Par ailleurs, il est indiqué dans la description du jeu de donnée, que les variables 148 à 167 sont les mêmes que celles de 128 à 147. Ainsi, on peut s'en séparer sans risquer de perdre des informations sur notre jeu de donnée.

```
rm(list=ls())
data <- read.csv("Music_2023.txt",sep=";",header=TRUE)
data.old <- data
n.old <- nrow(data)
p.old <- ncol(data)

data <- data.old[,-c(128:147)]
data$PAR_SC <- log(data$PAR_SC)
data$PAR_SC_V <- log(data$PAR_SC_V)

n <- nrow(data)
p <- ncol(data)
```

Variables très corrélées

```
corr <- cor(x=data[,-p])

#selection des indices de la matrice de correlation > 0.99
high.corr.index.new <- which(corr > 0.99, arr.ind = TRUE) %>% unname
```

```
#selection des indices appartenant a la matrice triangulaire inferieure stricte,
#pour retirer les doublons, ainsi que les elements diagonaux.
lower.tri <- lower.tri(corr, diag=FALSE)
high.corr.index.new <- high.corr.index.new[which(lower.tri[high.corr.index.new]==TRUE),]
high.corr.index.new
```

```
##      [,1] [,2]
## [1,]   37   36
## [2,]   72   71
## [3,]  164  160
```

```
correlated.variables <- matrix(c(names(data)[high.corr.index.new[,1]],
                         names(data)[high.corr.index.new[,2]]), nrow=nrow(high.corr.index.new))
correlated.variables
```

```
##      [,1]                [,2]
## [1,] "PAR_ASE34"         "PAR_ASE33"
## [2,] "PAR_ASEV34"        "PAR_ASEV33"
## [3,] "PAR_ZCD_10FR_MEAN" "PAR_ZCD"
```

On remarque que les deux premiers couples de variables très corrélées sont en fait les deux dernières mesures associées respectivement aux variables PAR_ASE et PAR_ASEV.

Le dernier couple de corrélation très élevée montre que la variable *PAR_ZCD* est très corrélée avec *PAR_ZCD_10FR_MEAN* qui semble être une moyenne de *PAR_ZCD*.

On veillera à bien retirer à chaque fois l'une des deux variables très corrélées, en effet les garder augmenterait la dimmension et la complexité du modèle, sans pour autant apporter de l'information utile.

On retirera par exemple les variables *PAR_ASE34*, *PAR_ASEV34* et *PAR_ZCD_10FR_MEAN*

```
data <- data[,-high.corr.index.new[,1]]
n <- nrow(data)
p <- ncol(data)

dim(data)
```

```
## [1] 4278  169
```

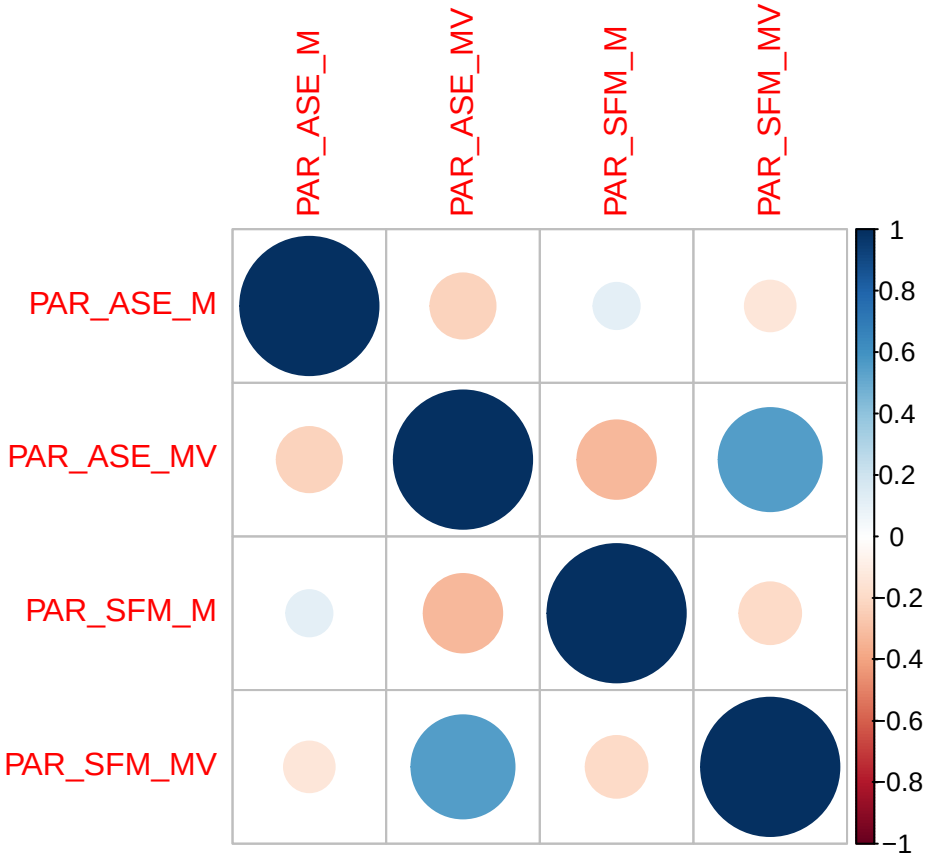
Cas des variables *PAR_ASE_M*, *PAR_ASE_MV*, *PAR_SFM_M* et *PAR_SFM_MV*

A FAIRE

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
var.data <- data[ , names(data) %in% c("PAR_ASE_M", "PAR_ASE_MV", "PAR_SFM_M", "PAR_SFM_MV")]
var.corr <- cor(var.data)
corrplot(var.corr)
```



Echantillon d'apprentissage

```
set.seed(103)
train=sample(c(TRUE,FALSE),n,rep=TRUE,prob=c(2/3,1/3))
```