Projet STA203

Ray Loic

25/04/2020

Dans ce projet nous allons étudier un jeu de données musical. Ce jeu de données contient 191 variable quantitatives et 1 variables qualitative qui représente le genre de musique. Nous allons implémenter 3 méthodes permettant de prédire la variable de genre à partir des autres variables. Commençons donc par importer le jeu de données.

```
data=read.table("Music.txt", header = TRUE, sep=";")
#head(data)
#summary(data) #trop long
#str(data)
ncol(data) #nombre de variable
```

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

```
nrow(data) #nombre d'observation
```

```
## [1] 6447
```

Partie 1

Question 1

Il y a trop de variables pour pouvoir extraire des informations en regardant les données bruts. Mais on peut toutefois faire des analyses univariée et bivariée sur quelques variables.

Regardons tout d'abord le nombre d'individu de chaque classe.

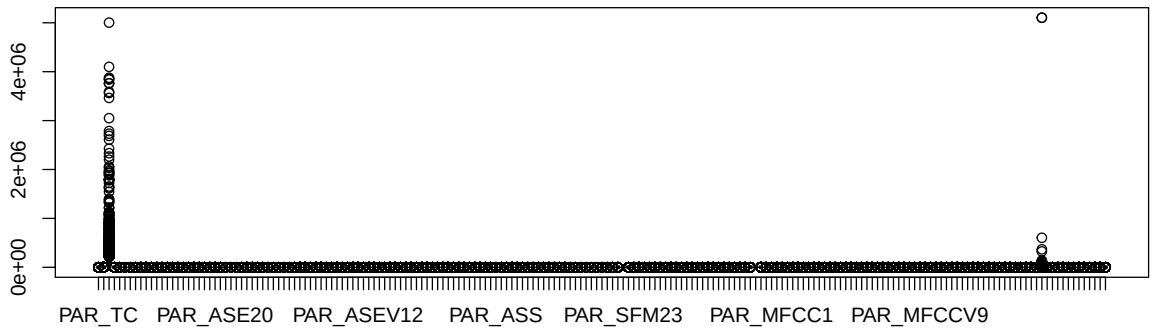
```
summary(data[192]) #nombre de d'individu de chaque groupes
```

```
##        GENRE
##  Classical:3444
##  Jazz     :3003
```

On peut déjà remarquer que le jeu de données est relativement équilibrés, ce qui nous permet de faire une étude qui ne soit pas trop biaisée.

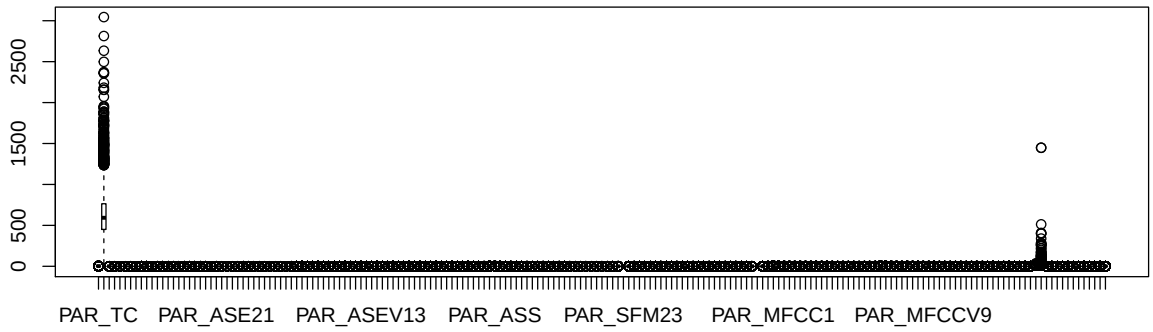
Faisons une analyse univariée, sans considérer la variable qualitative 192.

```r
boxplot(data[-192])
```



Cela ne ce voit pas bien sur le graphique mais les variables 3 **PAR_SC_V** , et 179 **PAR_PEAK_RMS10FR_VAR** prennent des valeurs bien plus élevées que les autres variables. On peut notamment remarquer que ces deux variables ont des variances élevées (du fait de la répartions des points au dessus de leur boites) Affichons un nouveau boxplot sans prendre en compte ces variables.

```r
boxplot(data[-c(3,179,192)])
```



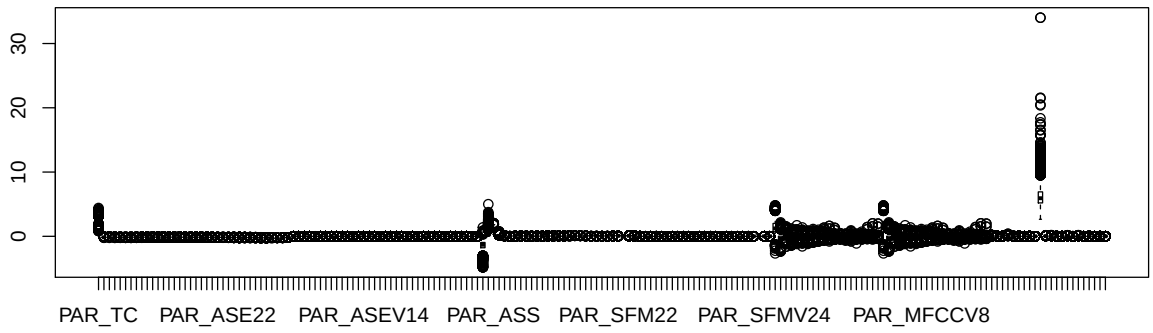

```r
#boxplot((data[c(-3,-179,-192)])[2:10])
```

Nous voyons ici que les variables et 2 **PAR_SC** et 178 **PAR_PEAK_RMS10FR_MEAN** prennent elles aussi des valeurs bien supérieures aux autres variables, avec là encore une grande variance entre les valeurs. Par ailleurs cela n'est pas étonnant car ces variables et les variables précédentes sont reliées. En effet les variables que nous avons ici représente des moyennes, et les variables précédetent représentaient la variance associée.

Faisons un dernier boxplot sans ces variables.

ON POURRAIT FAIRE UN TRUC MIEUX EN ECRIVANT UNE FONCTION QUI RENVOIE LE NOMS DE TOUTES LES VARIAVLES QUI PRENNENT DES VALEURS SUPERIEURES A UNE CONSTANT GENRE 1

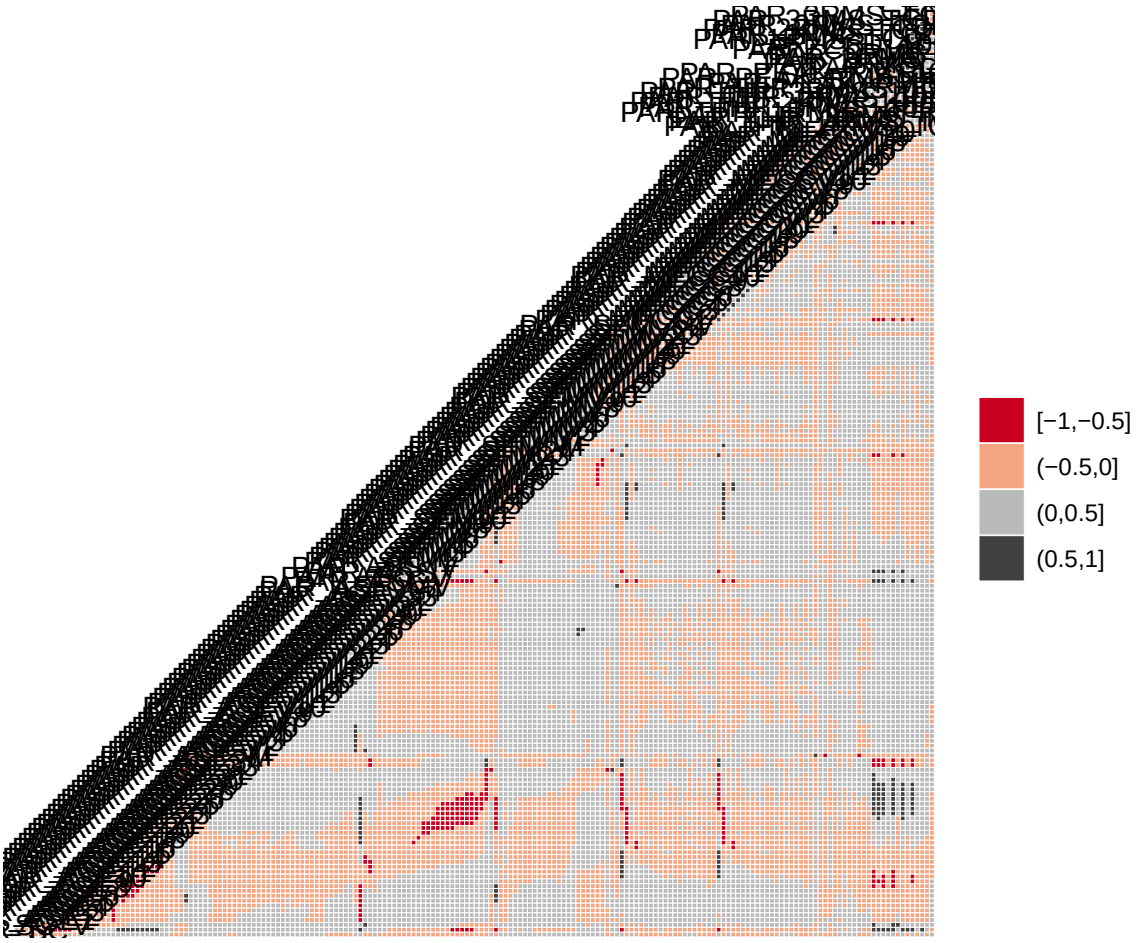
```
boxplot(data[-c(2,3,178,179,192)])
```




```
#boxplot((data[-c(2,3,178,179,192)])[175:180])
```

Interessons nous maintenant aux corrélations entre les variable, en faisant une étude bivariée du jeu de données. Pour cela calculons et affichons la matrice de corrélations.

```
matrice_data=data.matrix(data)
correlation_data=cor(matrice_data)
ggcorr(matrice_data,nbreaks = 4, palette = "RdGy")
```




```
#corrplot(correlation_data, tl.pos='n')
```

Comme cela était attendu, le graphique est quasiment illisible. Mais on parvient tout de même à diserner des zones de forte covariance. Implementons une fonction qui affiche les variables dont la covariance est comprise entre 2 bornes, afin de retirer des informations plus pertinentes de la matrice de corrélation.

```
print_corr_borne= function(mat_cor,seuil_min,seuil_max){
  l=nrow(mat_cor)
  found=FALSE
  for(i in 1:l){
    for(j in 1:i){
      if(mat_cor[i,j]>seuil_min &&mat_cor[i,j]<seuil_max){
        #Affiche le nom des variables correspondantes
        found=TRUE
        print(names(data)[c(i,j)])
      }
    }
  }
  if(!found){
    print("Il n'y a aucune covariance n'est comprise entre ces bornes")
  }
}
```

Nous pouvons alors afficher les variables trés corrélées, dont la covariance se trouve dans $]0.99; 1[$

```
print_corr_borne(correlation_data,0.99,1)
```

```
## [1] "PAR_ASE34" "PAR_ASE33"
## [1] "PAR_ASEV34" "PAR_ASEV33"
## [1] "PAR_MFCCV1" "PAR_MFCC1"
## [1] "PAR_MFCCV5" "PAR_MFCC5"
## [1] "PAR_MFCCV12" "PAR_MFCC12"
## [1] "PAR_MFCCV13" "PAR_MFCC13"
## [1] "PAR_MFCCV14" "PAR_MFCC14"
## [1] "PAR_MFCCV19" "PAR_MFCC19"
## [1] "PAR_ZCD_10FR_MEAN" "PAR_ZCD"
```

Ainsi que les variables trés anti-corrélées, dont la covariance se trouve dans $]-1; 0.99[$

```
print_corr_borne(correlation_data,-1,-0.99)
```

```
## [1] "Il n'y a aucune covariance n'est comprise entre ces bornes"
```

On remarque donc que les variables très corrélées sont de type *MFCCV* et *MFCC* ainsi que des variables *ASE*.

Considérons les variables 128 à 147 et 148 à 167. En regardant le jeu de données et son déscriptif, il semblerait que ces deux groupes de variables soient égaux. Pour le confirmer on écrit un script qui renvoie le nombre de différence entre ces 2 groupes.

```
#Egalite 128:147 et 148:167
dif=0
for(i in 128:147){
  dif=sum(data[i]!=data[i+20])
}
dif
```

```
## [1] 0
```

Comme indiqué dans le descriptif du dataset, les colonnes 128 à 147 et 148 à 167 ont les mêmes valeurs. On ne considèrera donc pas les colonnes 148 à 167 dans la suite.

Les données **PAR_ASE_M**, **PAR_ASE_MV**, **PAR_SFM_M** et **PAR_SFM_MV** représentent les moyennes des variables 4 à 37, 39 à 72, 78 à 101, et 103 à 126. Pour réduire le nombre de variable il peut être préférable dans un premier temps de pas considéré les colonnes 4 à 37, 39 à 72, 78 à 101, et 103 à 126 comme les variables **PAR_ASE_M**, **PAR_ASE_MV**, **PAR_SFM_M** et **PAR_SFM_MV** en sont des agréats.

On réalise les opérations de nettoyage précédement expliquées et on note X le nouveau data frame de données que nous allons utiliser dans la suite. Et *Genre* le data-frame d'une colonne contenant la variable qualitative Genre.

```
#Colonnes que nous n'utiliserons pas dans la suite
del=c(148:167,
      4:37,
      39:72,
      78:101,
      103:126)
#
X=data[,-c(del,192)]
#log des variables PAR_SC_V et PAR_ASC_V
X["PAR_SC_V"]=log(data["PAR_SC_V"])
X["PAR_ASC_V"]=log(data["PAR_ASC_V"])
```

```
#
GENRE=data[,192]
```

Nous cherchons à déterminer un modèle logistique permettant de d'estimer les valeurs de la variable *GENRE*. Cette variable prend deux valeurs *Classical* et *Jazz*.

MODELE BERNOULLI... A ECRIRE

Question 2

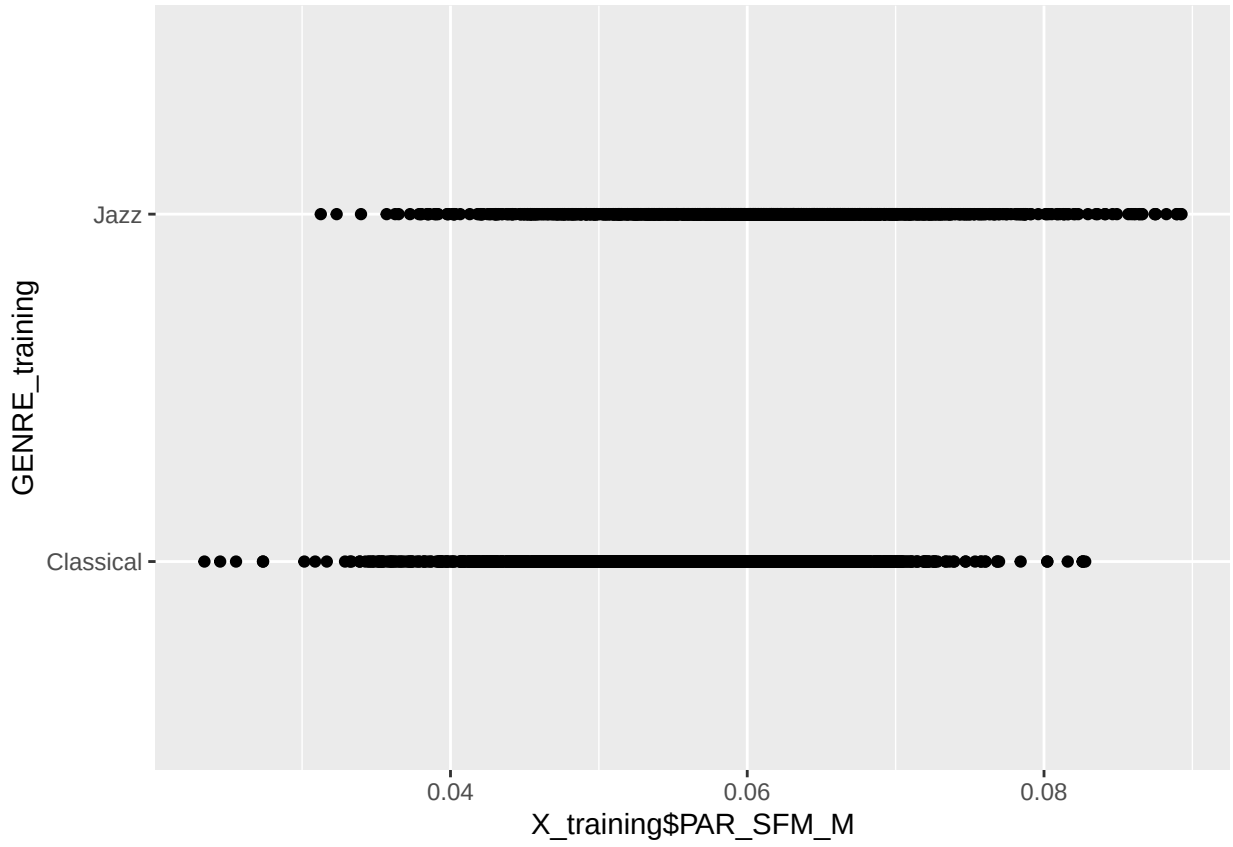
```
set.seed(103)
n=nrow(data)
train=sample(c(TRUE,FALSE),n,rep=TRUE,prob=c(2/3,1/3))
X_training=X[train,]
X_test=X[!train,]

GENRE_training=GENRE[train]
GENRE_test=GENRE[!train]
```

////// APRES CA C'EST A REVOIR

c("PAR_TC","PAR_SC", "PAR_SC_V", "PAR_ASE_M", "PAR_ASE_MV", "PAR_SFM_M", "PAR_SFM_MV")

Traçons la réponse *GENRE* en fonction des différentes variables.

```
ggplot()+
  geom_point(aes(x=X_training$PAR_SFM_M ,y=GENRE_training))
```



. . . . .

Question 3

Mod0

ModT

Mod1

Mod2

ModAIC

Partie 2

Question 1: K-NN

Question 2: Implémentation K-NN

```
library(class)
```