STA211 : pré-traitement, étude de cas

V. Audigier, N. Niang

17 février, 2025

1 Introduction

L'objet de ce document est de montrer comment mettre en oeuvre les méthodes présentées dans la section pré-traitement. Nous appliquerons ces méthodes sur les données *German Credit* à l'aide du logiciel R.

R est un des logiciels libres les plus utilisés par les statisticiens et data-miners en raison de la richesse de ses fonctionnalités. Cette richesse est notamment due à ses librairies (ou packages) proposées par la communauté des utilisateurs. Le logiciel dans sa version de base est un peu austère, il est conseillé de l'utiliser via une interface graphique telle que Rstudio. En particulier, R n'est pas seulement un logiciel, c'est aussi un langage de programmation. L'utilisation d'une interface comme RStudio (https://www.rstudio.com/products/rstudio/download/#download) facilitera grandement l'écriture de lignes de codes. De nombreuses documentations sur le logiciel sont disponibles sur internet. Parmi elles, l'auditeur pourra consulter :

- https://cran.r-project.org/doc/contrib/Goulet_introduction_programmation_R.pdf (https://cran.r-project.org/doc/contrib/Goulet_introduction_programmation_R.pdf) : ce libre constitue une bonne introduction au logiciel
- http://duclert.org/ (http://duclert.org/) : site référençant les fonctions de base très souvent utilisées en R

Les livres suivants pourront également être lus :

- *Statistique avec R*, 3eme édition Cornillon, P.A., Guyader A., Husson F., Jégou N., Josse J., Kloareg M., Matzner-Løber E., Rouvière L. (2012) Presses Universitaires de Rennes : ce livre fournit à la fois les

bases du logiciel et permet de mettre en oeuvre plusieurs dizaines de méthodes statistiques fréquemment utilisées
- *An Introduction to Statistical Learning with Applications in R*, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013), Springer : par rapport au précédent, ce livre permettra de mettre en oeuvre des méthodes plus avancées et plus en phase avec le cours de STA211

Après avoir importé les données à l'aide du logiciel, nous donnerons un aperçu général du jeu données puis développerons la partie exploratoire. A la suite de cette exploration, nous envisagerons différentes façons de prétraiter les données.

NB : Afin de vous approprier le code fourni, **il ne faut pas hésiter à modifier les arguments des fonctions**. Ceci vous permettra d'apprécier le rôle de ces arguments sur les sorties. Vous pouvez également aller consulter les aides des fonctions à la fin desquelles vous trouverez notamment des exemples.

2 Importation des données

```
don <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/g
erman.data",
                  sep = " ",
                  stringsAsFactors = TRUE)
```

La commande *str* permet d'avoir un rapide aperçu des données importées

```
str(don)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ V1 : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...
##  $ V2 : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ V3 : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 ...
##  $ V4 : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...
##  $ V5 : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ V6 : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...
##  $ V7 : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...
##  $ V8 : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ V9 : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...
##  $ V10: Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ V11: int  4 2 3 4 4 4 4 2 4 2 ...
##  $ V12: Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...
##  $ V13: int  67 22 49 45 53 35 53 35 61 28 ...
##  $ V14: Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ V15: Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...
##  $ V16: int  2 1 1 1 2 1 1 1 1 2 ...
##  $ V17: Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3 2 3 4 2 4 ...
##  $ V18: int  1 1 2 2 2 2 1 1 1 1 ...
##  $ V19: Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...
##  $ V20: Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...
##  $ V21: int  1 2 1 1 2 1 1 1 1 2 ...
```

On constate que les noms des variables, comme ceux de leurs modalités (pour les variables qualitatives) ne sont pas du tout explicites. Ceci rend l'interprétation très difficile. Pour y remédier, on renomme les différentes variables et les modalités selon le descriptif des données (disponible ici (https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/german.doc)).

```r
# Modification des noms des variables
colnames(don)<-c(
  "Status",
  "Duration",
  "History",
  "Purpose",
  "Credit.Amount",
  "Savings account/bonds",
  "Length.of.current.employment",
  "Instalment.per.cent",
  "Sex.Marital.Status",
  "Guarantors",
  "Duration.in.Current.address",
  "Property",
  "Age.years",
  "Other.installment.plans",
  "Housing",
  "No.of.Credits.at.this.Bank",
  "Job",
  "No.of.dependents"
  ,"Telephone",
 "Foreign.Worker",
 "Creditability")


# Modification des noms des modalités des variables qualitatives
levels(don$Status) <- c("lt.0", "0.to.200", "gt.200", "none")

levels(don$History) <-
  c("noCredit.allPaid",
    "thisBank.AllPaid",
    "paidDuly",
    "delay",
    "critical")

levels(don$Purpose) <-
  c(
    "NewCar",
    "UsedCar",
    "Other",
    "Furniture.Equipment",
    "Radio.Television",
    "DomesticAppliance",
    "Repairs",
    "Education",
    "Retraining",
    "Business"
  )

levels(don$`Savings account/bonds`) <-
  c("lt.100", "100.to.500", "500.to.1000", "gt.1000", "Unknown")

levels(don$Length.of.current.employment) <-
  c("lt.1", "1.to.4", "4.to.7", "gt.7", "Unemployed")
```

```
levels(don$Sex.Marital.Status) <-
  c(
    "Male.Divorced.Seperated",
    "Female.NotSingle",
    "Male.Single",
    "Male.Married.Widowed"
  )

levels(don$Guarantors) <- c("None", "CoApplicant", "Guarantor")

levels(don$Property) <-
  c("RealEstate", "Insurance", "CarOther", "Unknown")

levels(don$Other.installment.plans) <- c("Bank", "Stores", "None")

levels(don$Housing) <- c("Rent", "Own", "ForFree")

levels(don$Job) <-
  c(
    "UnemployedUnskilled",
    "UnskilledResident",
    "SkilledEmployee",
    "Management.SelfEmp.HighlyQualified"
  )

levels(don$Foreign.Worker) <- c("yes", "no")

levels(don$Telephone) <- c("none", "yes")
```

On modifie également le type de certaines variables. En effet, R interprête les types de variables pour proposer des analyses statistiques adaptées. Par exemple, si une variable est de type *factor*, un appel à la fonction *lm* (permettant de construire un modèle de régression) où cette variable serait une variable explicative amènera à considérer un coefficient de régression pour chacune de ses modalités, là où un seul coefficient serait considéré si la variable était de type *numeric*.

```
#Codage des variables quantitatives en type "numeric" (plutot que "integer")

##pour la variable Duration
don$Duration <- as.numeric(don$Duration)
class(don$Duration)

## pour les variables Credits.Amount et Age.years
don$Credit.Amount <- as.numeric(don$Credit.Amount)
don$Age.years <- as.numeric(don$Age.years)

#Codage de la variable réponse en type "factor"
don$Creditability <- as.factor(don[, "Creditability"])
levels(don$Creditability) <- c("good", "bad")
```

3 Aperçu général

```
#nombre d'individus et variables
dim(don)
```

```
## [1] 1000   21
```

```
#nature des variables
table(sapply(don, class))
```

```
##
##  factor integer numeric
##      14       4       3
```

```
#variables qualitatives
var.factor <- which(sapply(don, class)=="factor")
names(var.factor)
```

```
##  [1] "Status"                       "History"
##  [3] "Purpose"                      "Savings account/bonds"
##  [5] "Length.of.current.employment" "Sex.Marital.Status"
##  [7] "Guarantors"                   "Property"
##  [9] "Other.installment.plans"      "Housing"
## [11] "Job"                          "Telephone"
## [13] "Foreign.Worker"               "Creditability"
```

```
#variables quantitatives
var.numeric <- which(sapply(don, class)=="numeric"|sapply(don, class)=="integer")
names(var.numeric)
```

```
## [1] "Duration"                    "Credit.Amount"
## [3] "Instalment.per.cent"         "Duration.in.Current.address"
## [5] "Age.years"                   "No.of.Credits.at.this.Bank"
## [7] "No.of.dependents"
```

```
#nombre de données manquantes
sum(is.na(don))
```

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

4 Exploration

4.1 Analyse univariée

4.1.1 Variables quantitatives

4.1.1.1 Indicateurs statistiques

```
# installation du package stargazer
install.packages("stargazer")

# chargement de la librarie
library(stargazer)

# affichage de quelques indicateurs statistiques pour variables quantitatives
stargazer(don,
          summary.stat = c("n", "min", "p25", "median", "mean", "p75", "max", "sd"),
          type = "text")
```

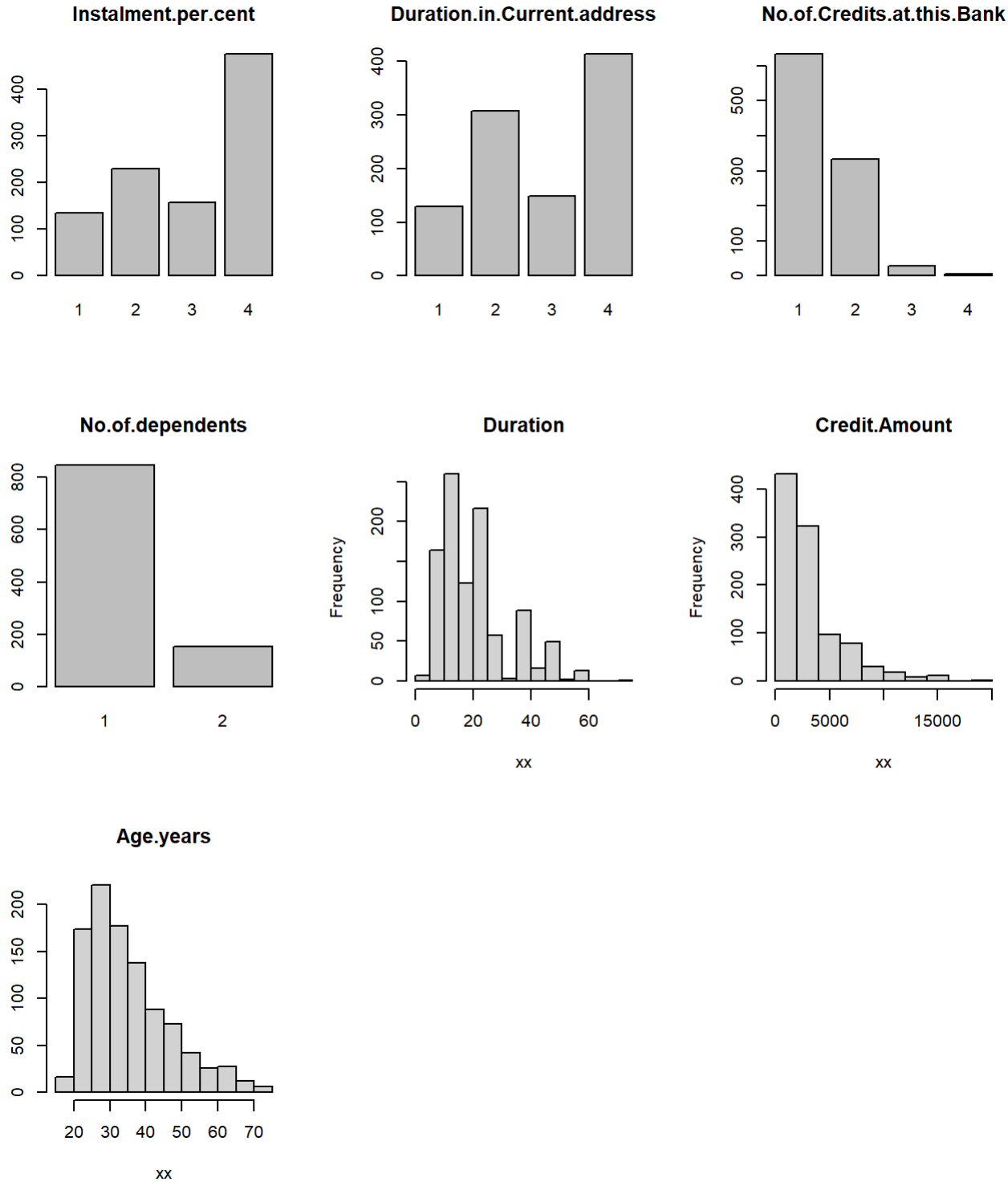
Statistic	N	Min	Pctl(25)	Median	Mean	Pctl(75)	Max	St. Dev.
Duration	1,000	4	12	18	20.903	24	72	12.059
Credit.Amount	1,000	250	1,365.5	2,319.5	3,271.258	3,972.2	18,424	2,822.737
Instalment.per.cent	1,000	1	2	3	2.973	4	4	1.119
Duration.in.Current.address	1,000	1	2	3	2.845	4	4	1.104
Age.years	1,000	19	27	33	35.546	42	75	11.375
No.of.Credits.at.this.Bank	1,000	1	1	1	1.407	2	4	0.578
No.of.dependents	1,000	1	1	1	1.155	1	2	0.362

On peut remarquer qu'aucune des variables n'est constante. De telles variables n'auraient en effet aucun intérêt pour l'analyse.

4.1.1.2 Représentations graphiques

On commence par visualiser les distributions marginales des variables via des diagrammes en barres et des histogrammes. Les variables *Duration*, *Credit.Amount* et *Age.years* ayant un grand nombre de valeurs distinctes, on les représentera via des histogrammes.

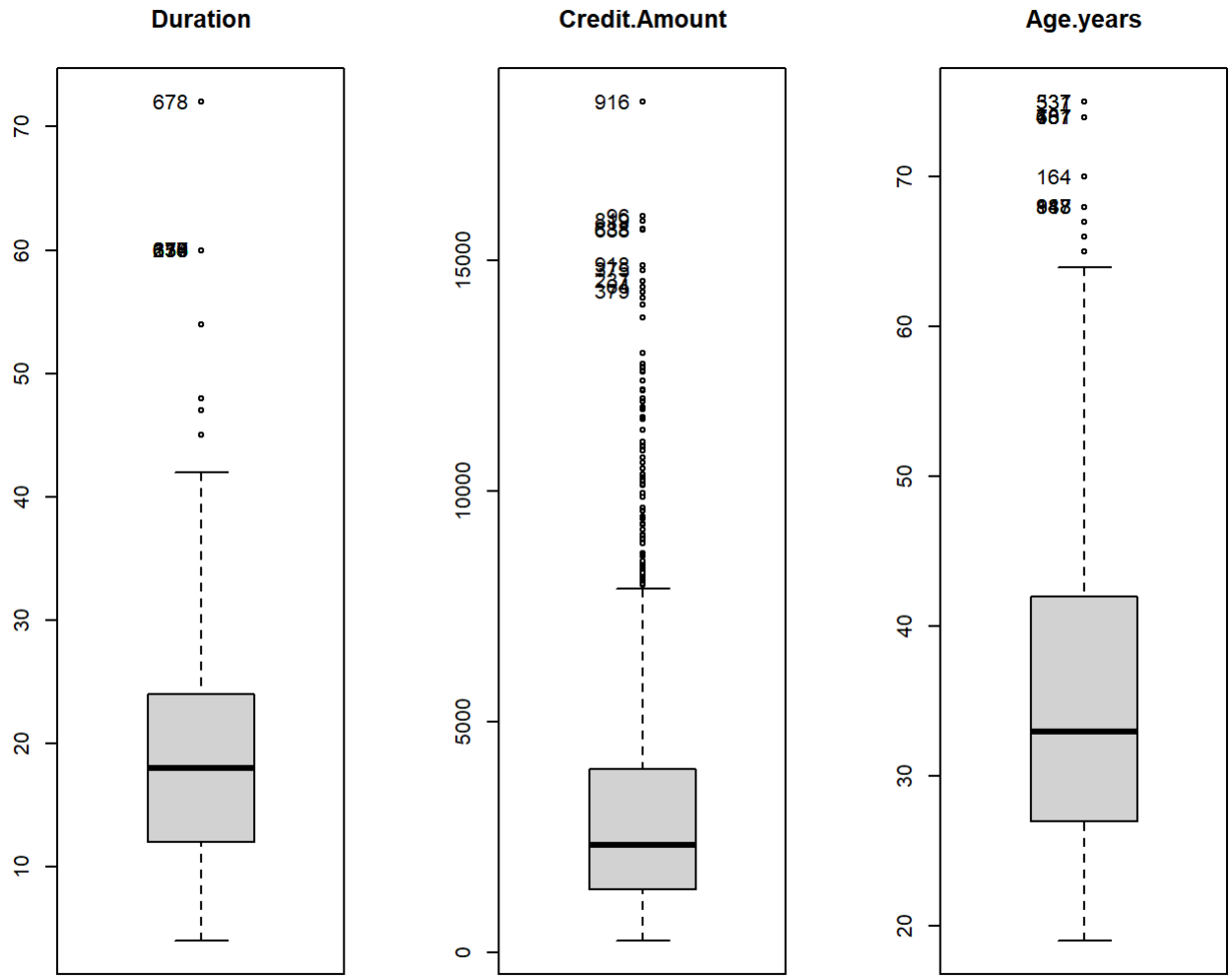
```
#diagrammes en barres
varbarplot <- c("Instalment.per.cent", "Duration.in.Current.address", "No.of.Credits.at.this.
Bank", "No.of.dependents")

mapply(don[,varbarplot],
       FUN = function(xx,name){barplot(table(xx), main = name)},
       name = varbarplot)

#histogrammes
varhist <- c("Duration", "Credit.Amount", "Age.years")
mapply(don[,varhist],
       FUN = function(xx,name){hist(xx, main = name)},
       name = varhist)
```



On identifie clairement une queue à droite sur les variables *Duration*, *Credit Amount* et *Age*. Une transformation logarithmique pourrait permettre de rendre ces distributions symétriques.

On repère ensuite les potentielles valeurs aberrantes par les boîtes à moustaches.

```
library(car)
mapply(don[,varhist],
       FUN = function(xx,name){Boxplot(xx, main = name, id.n = 2, ylab="")},
       name = varhist)
```

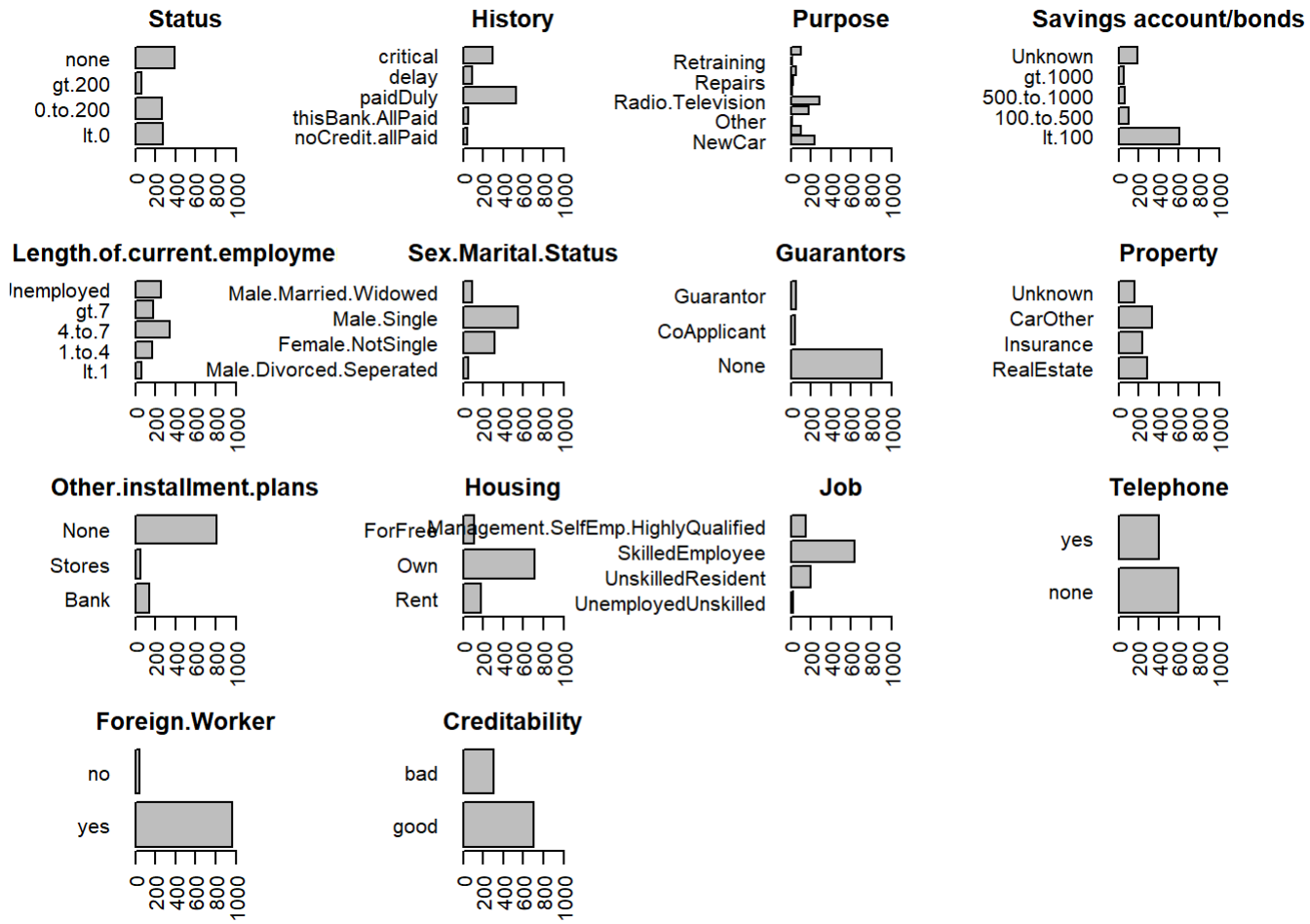


Bien que différentes valeurs semblent relativement élevées, il est difficile de prendre dès à présent la décision de gérer ces valeurs (par exemple en les considérant comme manquantes et en les imputant). Ce choix pourra être fait a posteriori si nécessaire.

4.1.2 Variables qualitatives

On repère des modalités rares via les diagrammes en barres (on pourrait de façon équivalente visualiser cette information sous forme de tableaux)

```
mapply(don[,var.factor],
       FUN = function(xx,name){barplot(table(xx),
                                       main = name,
                                       horiz = TRUE,
                                       las = 2,
                                       xlim = c(0,1000))},
       name = names(var.factor))
```



En fonction de la distribution de la variable réponse au sein de chaque modalité définie par ces variables, il pourra pertinent ou non d'effectuer certains regroupements. On aura besoin pour cela d'effectuer une analyse bivariée.

4.2 Analyse bivariée

Etant dans un cas de classification supervisée, nous distinguons l'analyse bivariée des couples mettant en jeu la variable réponse, de celle des couples ne portant que sur des variables explicatives.

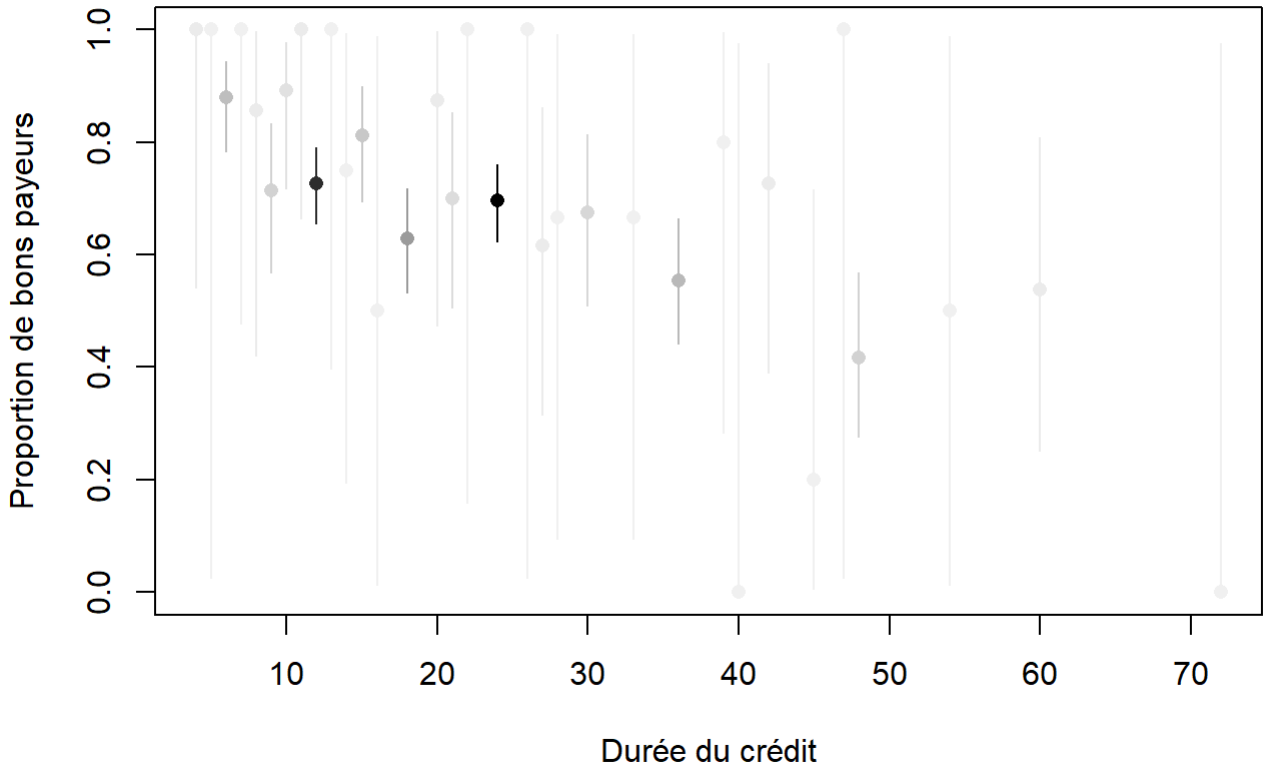
4.2.1 Lien entre variables explicatives et variable réponse

4.2.1.1 Linéarité

Afin d'identifier la nature du lien entre les variables quantitatives et la réponse, on représente la proportion de bons payeurs en fonction des variables quantitatives. Ce type d'analyse dans le cas d'une variable explicative continue nécessitera d'effectuer une discrétisation.

On commence par identifier le lien entre la variable *Creditability* et la variable *Duration*.

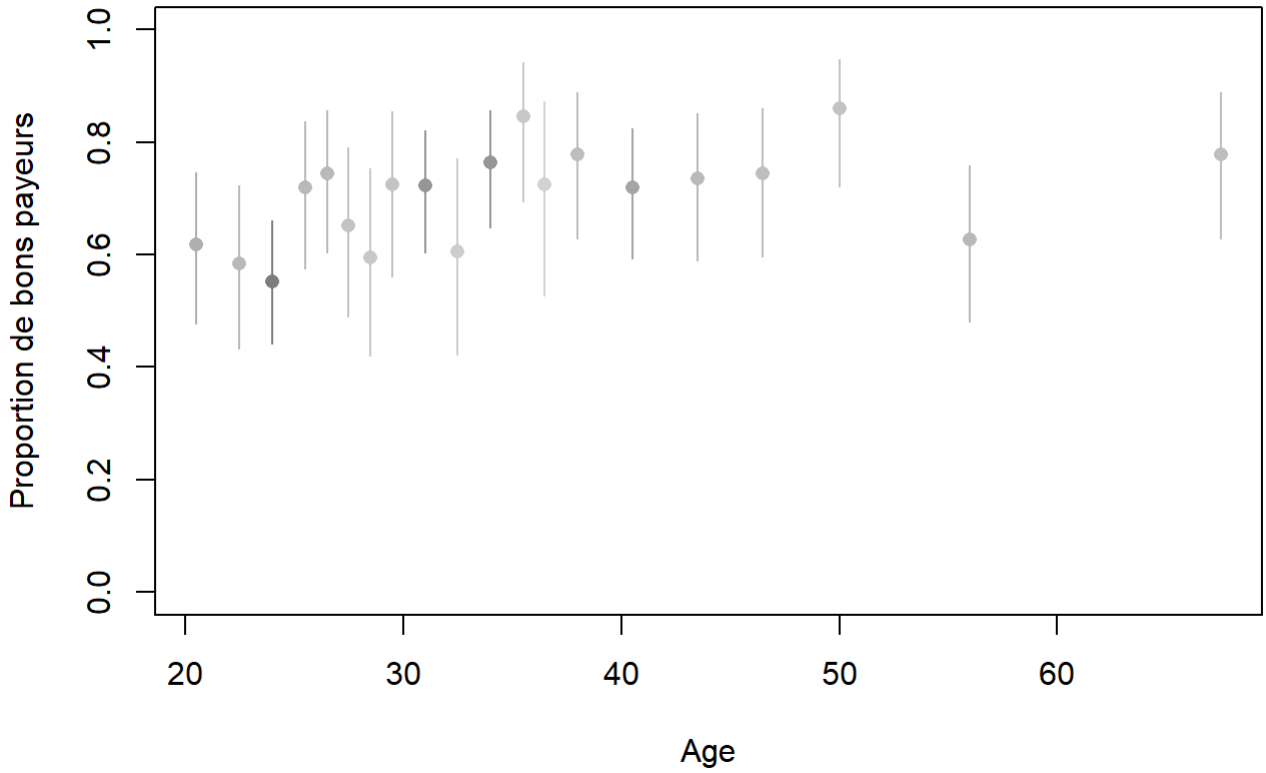
```r
# calcul de la proportion de bons payeurs pour chaque valeur de la durée de credit
cont.table <- table(don$Duration,don$Creditability)
prof.lignes <- prop.table(cont.table, 1)

#calcul des bornes de l'intervalle de confiance associée
res.binom.test <- mapply(cont.table[, 1],
                         FUN = binom.test,
                         n = rowSums(cont.table),
                         SIMPLIFY = FALSE)
ci<-sapply(res.binom.test, "[[", "conf.int")

# représentation des proportions en fonction de la durée de crédit
# (coloration en fonction du nombre d'observations pour la durée considérée)
abscisses <- as.numeric(rownames(prof.lignes))
col <- gray.colors(184,.95,0)[rowSums(cont.table)]

# affichage des proportions
plot(abscisses,
     prof.lignes[,1],
     pch = 16,
     col = col,
     xlab = "Durée du crédit",
     ylab = "Proportion de bons payeurs")

# affichage des intervalles de confiance
for(ii in 1:length(abscisses)){
  segments(x0 = abscisses[ii],
           y0 = ci[1,ii],
           x1 = abscisses[ii],
           y1 = ci[2,ii],
           col = col[ii])
}
```



Sur ce graphique, un point est d'autant plus noir que le nombre d'individus utilisés pour déterminer la proportion correspondante est grande. Les points les plus noirs sont donc ceux avec les intervalles de confiance les plus courts. On constate que la proportion de bons payeurs décroît de façon relativement linéaire avec la durée du Crédit.

On effectue la même opération pour la variable *Age* en effectuant au préalable une discrétisation en 20 classes.

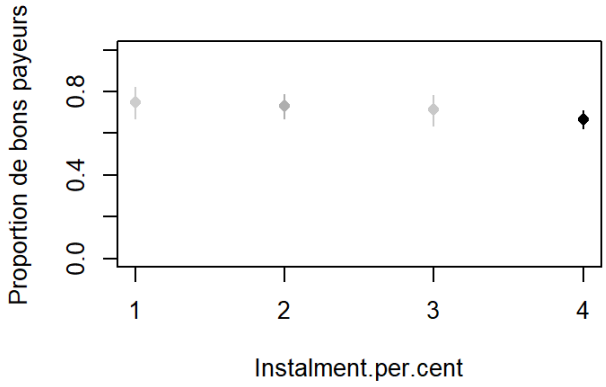
```
ordrequantiles <- seq(0, 1, 1/20)
Age.years.new <- cut(don$Age.years,
                     breaks = quantile(don$Age.years, probs = ordrequantiles))


cont.table <- table(Age.years.new,don$Creditability)
prof.lignes <- prop.table(cont.table, 1)
res.binom.test <- mapply(cont.table[,1],
                         FUN = binom.test,
                         n = rowSums(cont.table),
                         SIMPLIFY = FALSE)
ci <- sapply(res.binom.test, "[[", "conf.int")
abscisses <- hist(don$Age.years, quantile(don$Age.years,probs =ordrequantiles), plot = FALSE)
$mids
col <- gray.colors(112,.95,0)[rowSums(cont.table)]


plot(abscisses,
     prof.lignes[,1],
     pch = 16,
     col = col,
     xlab = "Age",
     ylab = "Proportion de bons payeurs",
     ylim = c(0, 1))

for(ii in 1:length(abscisses)){
  segments(x0 = abscisses[ii],
           y0 = ci[1,ii],
           x1 = abscisses[ii],
           y1 = ci[2,ii],
           col = col[ii])
}
```

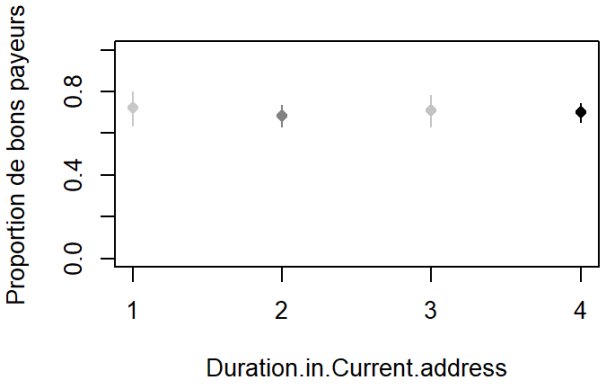


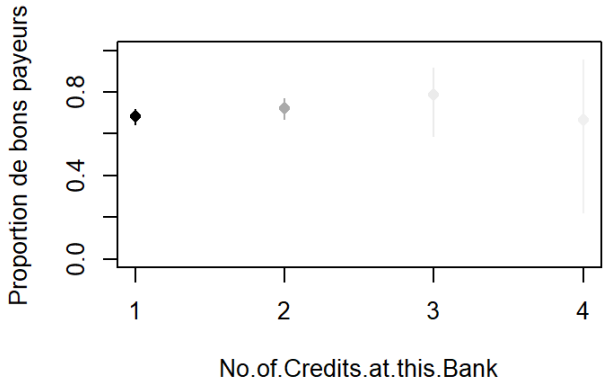
On voit que la liaison est plutôt non-monotone. On pourra découper la variable Age en 3 classes pour gérer cette non-linéarité (résultat en figure 5.1).

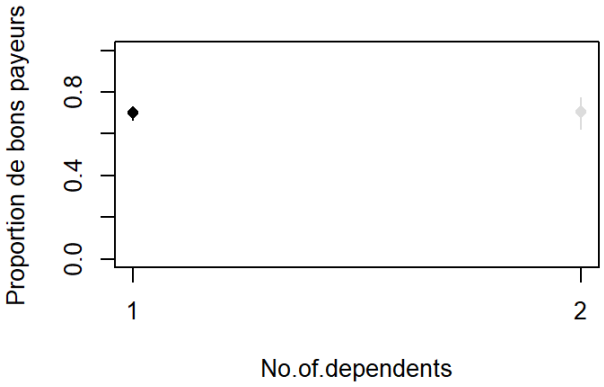
On visualise enfin le lien avec les autres variables quantitatives (discrètes avec peu de modalités)

```
par(mfrow=c(2,2),mar=c(4,5, 3, 2) + 0.1)

for(ii in c("Instalment.per.cent", "Duration.in.Current.address", "No.of.Credits.at.this.Bank", "No.of.dependents")){
  xx <- don[,ii]
  cont.table <- table(cbind.data.frame(xx,don$Creditability))
  prof.lignes <- prop.table(cont.table,1)
  res.binom.test <- mapply(cont.table[,1],
                           FUN=binom.test,
                           n=rowSums(cont.table),
                           SIMPLIFY = FALSE)
  ci <- sapply(res.binom.test,"[[","conf.int")

  #graphique pour la variable courante
  abscisses <- as.numeric(rownames(prof.lignes))
  col <- gray.colors(max(rowSums(cont.table)),.95,0)[rowSums(cont.table)]
  plot(x = abscisses,
       y=prof.lignes[,1],
       pch=16,
       col=col,
       ylab="Proportion de bons payeurs",
       xlab=ii,
       xaxt="n",
       ylim=c(0,1))
  axis(side=1,at = abscisses,labels = abscisses,xlab=varbarplot[ii])
  for(ii in 1:length(abscisses)){
    segments(x0=abscisses[ii],
             y0=ci[1,ii],
             x1=abscisses[ii],
             y1=ci[2,ii],
             col=col[ii])
  }
}
```





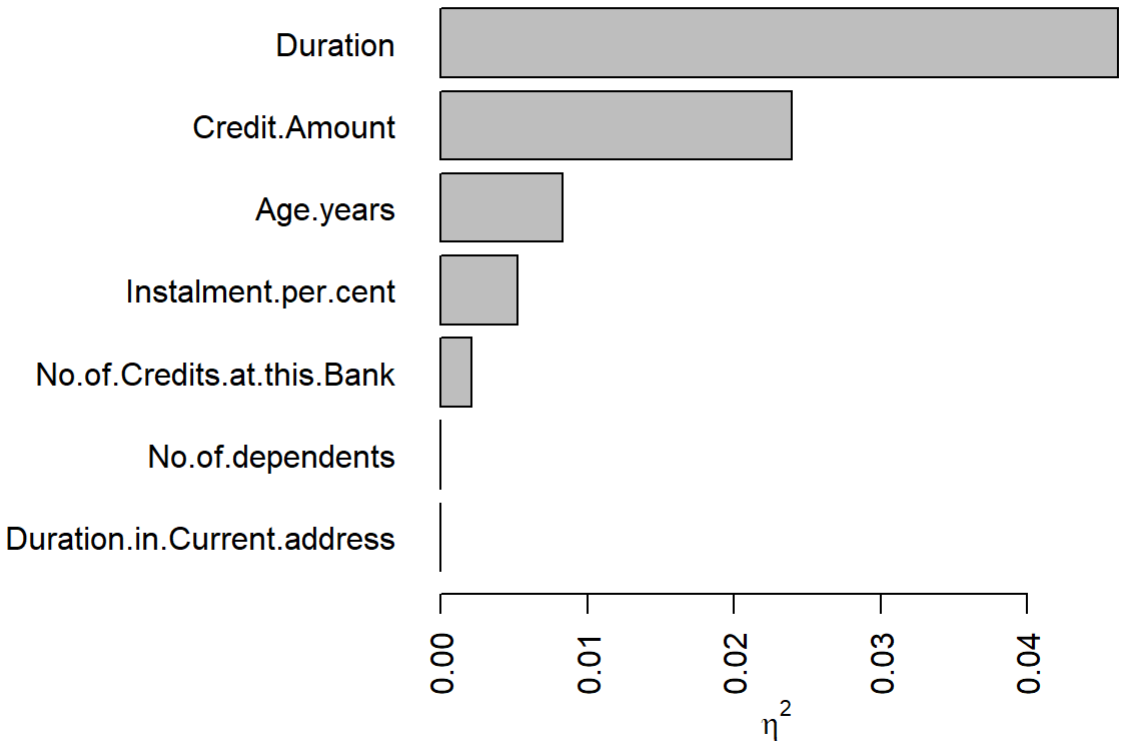




4.2.1.2 Identification des variables les plus discriminantes

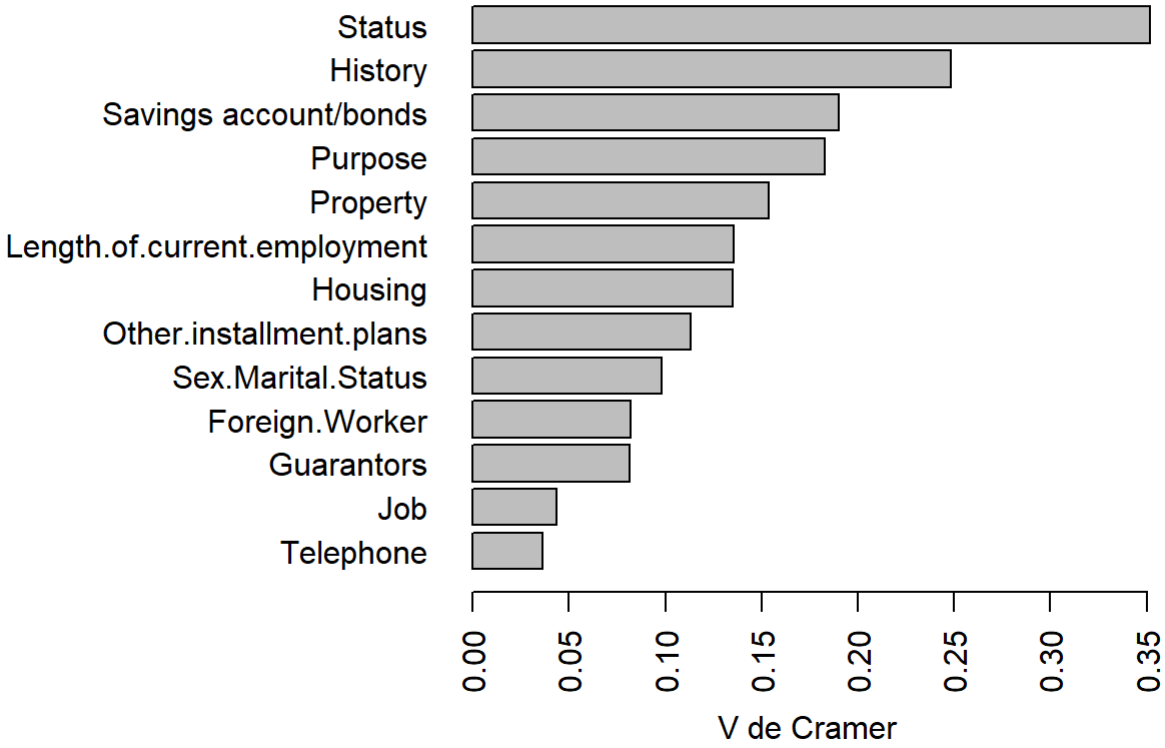
On regarde ensuite quelles sont les variables les plus liées au statut bon/mauvais payeur.

```
#Chargement du package BioStatR
install.packages("BioStatR")
library(BioStatR)

#calcul des rapport de corrélation
res.eta2 <- sapply(don[,var.numeric], eta2 ,y = don$Creditability)

#tri par valeurs décroissantes
res.eta2 <- sort(res.eta2)

#représentation
par(mar = c(5, 15, 4, 2) + 0.1, mfrow = c(1,1))#pour gérer les marges du graphique
barplot(res.eta2, horiz = TRUE, las = 2, xlab = expression(eta^2))
```



Parmi les variables quantitatives, les liaisons les plus fortes sont observées pour les variables *Duration* et *Credit Amount*. Au contraire, les variables *No.of.dependents* et *Duration in Current address* n'apparaissent pas comme discriminantes.

```
#Creation d'une matrice contenant les variables qualitatives et quantitatives discrètes (sans la variable Creditability)
don.cramer <- don[,c(var.factor)]
don.cramer <- don.cramer[,-which(colnames(don.cramer)=="Creditability")]

#calcul du V de cramer entre Creditability et les autres variables non continues de don.cramer
library(DescTools)
res.cramer <- sapply(don.cramer,
                     FUN = function(xx,yy){CramerV(table(xx,yy))},
                     yy = don$Creditability)

#tri par valeurs décroissantes
res.cramer <- sort(res.cramer)

#représentation
par(mar=c(5, 15, 4, 2) + 0.1)
barplot(res.cramer, horiz = TRUE, las = 2, xlab = "V de Cramer")
```



Parmi les variables qualitatives, les variables les plus liées sont *Status* et *History*, tandis que les variables *Job* et *Telephone* ne semblent pas discriminantes.

Ces analyses pourront être utiles en vue d'une réduction du nombre de colonnes, les variables les moins discriminantes seront a priori amenées à être écartées en priorité. Attention toutefois car on a évalué ici un lien direct entre les variables explicatives et la réponse, il est possible que la liaison soit plus complexe, mettant en jeu des liaisons de type interaction par exemple. Pour les détecter, on pourrait utiliser une régression logistique ou des arbres binaires (cf Tufféry (2007)).

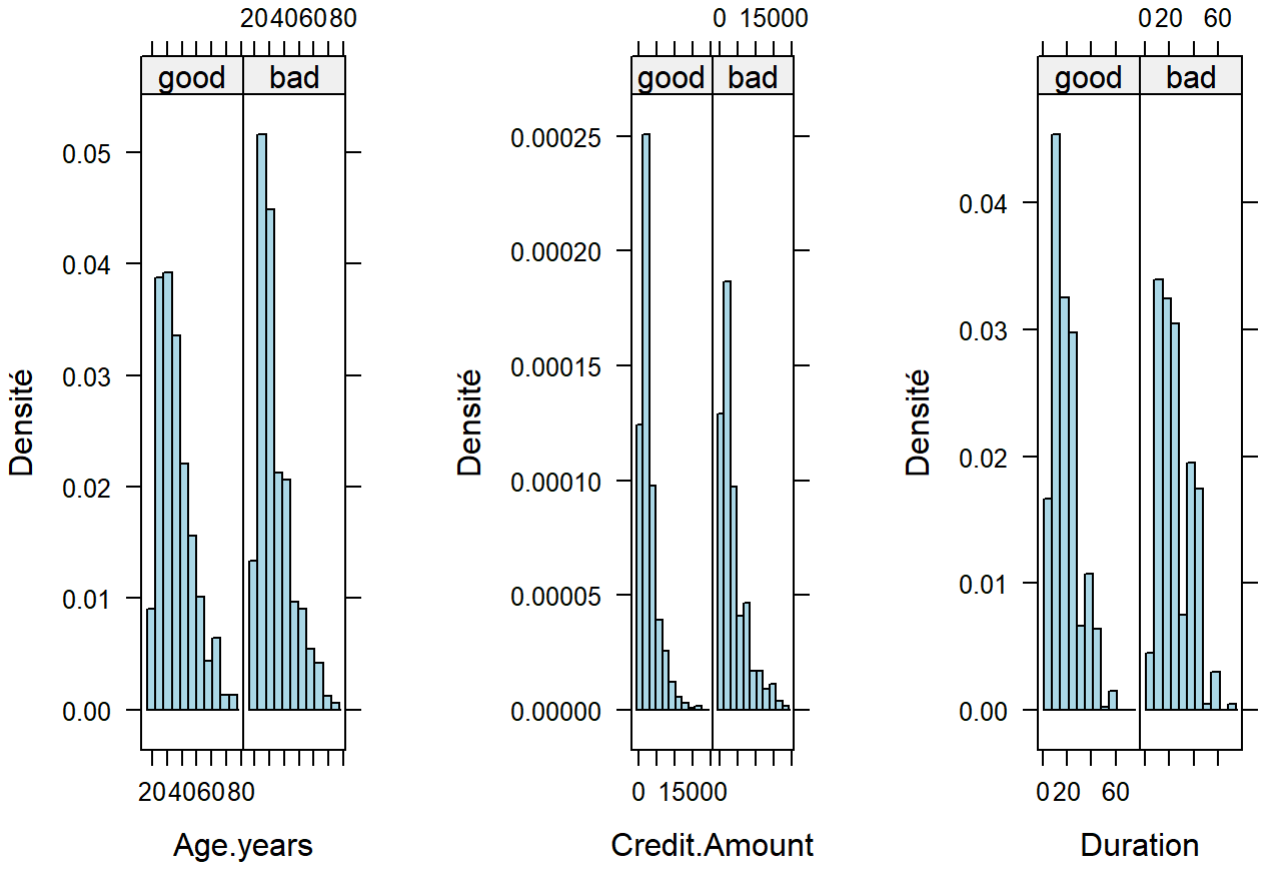
On pourra tester le caractère significatif des liaisons entre la variable réponse qualitative en utilisant la fonction *catdes* du package *FactoMineR* permettant de décrire une partition (ici selon les modalités *good* et *bad* de la variable réponse) à partir des variables quantitatives, et des modalités des variables qualitatives (voir Lebart, Morineau, and Piron (2006) pour la méthode et le site du package (http://factominer.free.fr/factomethods/categories-description.html) pour la lecture des sorties de la fonction). Notons néanmoins que ceci n'est pertinent que si le nombre d'observations est modéré, sans quoi toutes les variables risqueraient d'être considérées comme statistiquement reliées à la variable réponse. Par ailleurs, pour éviter les hypothèses paramétriques, il pourra être préférable d'utiliser des intervalles de confiance bootstrap (Lejeune (2010)).

```
# installation et chargement du package FactoMineR
install.packages("FactoMineR")
library(FactoMineR)
catdes(don, num.var = ncol(don))
```

4.2.1.3 Allure des distributions conditionnelles

La distribution des variables continues conditionnellement à la variable réponse est parfois déterminante pour l'utilisation de certains modèles, notamment l'analyse linéaire discriminante. On analyse donc la nature de ces distributions.

```
# Chargement de la librarie lattice permettant de faire des graphiques relativement avancés
# et de la librairie gridExra, utilisée ici pour positionner les graphiques les uns à côté de
s autres

library(lattice);library(gridExtra)

# distribution conditionnelle de Age.Years
plot1 <- lattice::histogram(~Age.years|Creditability,
                            data = don,
                            type = "density",
                            col = "lightblue",
                            ylab = "Densité")
# distribution conditionnelle de Credit.Amount
plot2 <- lattice::histogram(~Credit.Amount|Creditability,
                            data = don,
                            type = "density",
                            col = "lightblue",
                            ylab ="Densité")
# distribution conditionnelle de Duration
plot3 <- lattice::histogram(~Duration|Creditability,
                            data = don,
                            type = "density",
                            col = "lightblue",
                            ylab = "Densité")

# affichage
grid.arrange(plot1, plot2, plot3, nrow =1, ncol = 3)
```



Clairement, les distributions conditionnelles des variables *Age*, *Credit Amont* et *Duration* ne sont pas normales. Peut-être sera-t-il nécessaire de transformer ces variables par la suite si cette normalité était requise par les méthodes employées. Notons qu'il est aussi possible de comparer certains indicateurs statistiques entre les deux groupes

```
by(don[,-ncol(don)],
   INDICES = don$Creditability,
   FUN = stargazer,
   summary.stat = c("n", "min", "p25", "median", "mean", "p75", "max", "sd"),
   type = "text")
```

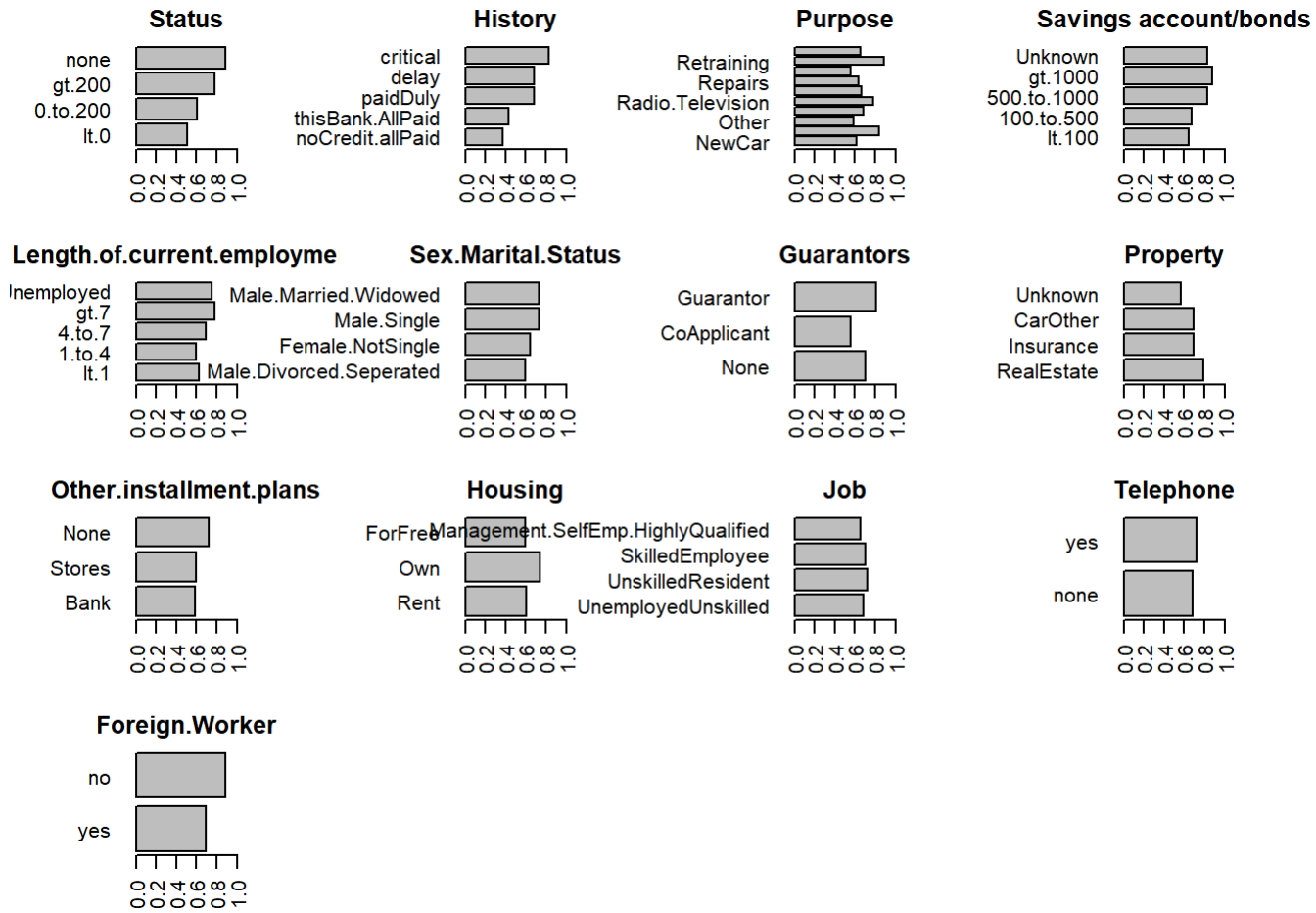
Statistic	N	Min	Pctl(25)	Median	Mean	Pctl(75)	Max	St. Dev.
Duration	700	4	12	18	19.207	24	60	11.080
Credit.Amount	700	250	1,375.5	2,244	2,985.457	3,634.8	15,857	2,401.472
Instalment.per.cent	700	1	2	3	2.920	4	4	1.128
Duration.in.Current.address	700	1	2	3	2.843	4	4	1.108
Age.years	700	19	27	34	36.224	42.2	75	11.381
No.of.Credits.at.this.Bank	700	1	1	1	1.424	2	4	0.585
No.of.dependents	700	1	1	1	1.156	1	2	0.363

Statistic	N	Min	Pctl(25)	Median	Mean	Pctl(75)	Max	St. Dev.
Duration	300	6	12	24	24.860	36	72	13.283
Credit.Amount	300	433	1,352.5	2,574.5	3,938.127	5,141.5	18,424	3,535.819
Instalment.per.cent	300	1	2	4	3.097	4	4	1.088
Duration.in.Current.address	300	1	2	3	2.850	4	4	1.095
Age.years	300	19	25	31	33.963	40	74	11.222
No.of.Credits.at.this.Bank	300	1	1	1	1.367	2	4	0.560
No.of.dependents	300	1	1	1	1.153	1	2	0.361

Cela sera particulièrement utile en présence d'un grand nombre de variables quantitatives. Par exemple, la comparaison des moyennes et écart-types de la variable *Credit.Amount* dans les deux groupes met en évidence une asymétrie à droite dans chacun d'entre eux, ce qui est incompatible avec une hypothèse de normalité.

Pour les variables qualitatives, il sera intéressant d'identifier la distribution conditionnelle de la variable réponse en fonction des modalités des variables. Ceci permettra d'effectuer des regroupements de modalités préservant au mieux les liaisons entre ces variables et la variable réponse. On choisit donc de représenter la proportion de bons payeurs en fonction des modalités des différentes variables explicatives qualitatives.

```
var.expl.quali <- names(var.factor[-length(var.factor)])
mapply(don[,var.expl.quali], FUN = function(xx,name){
  tmp <- table(xx,don$Creditability)
  tmp <- tmp/rowSums(tmp)
  barplot(tmp[,"good"], main = name, horiz = TRUE, las = 2, xlim = c(0,1))
}, name = var.expl.quali)
```



Par exemple, on voit que la proportion de bons payeurs est sensiblement la même que le client prenne la modalité *thisBank.AllPaid* ou *noCredit.allPaid* de la variable *History*. Ces deux modalités étant rares (cf Section 4.1.2), on pourra les fusionner si cela est nécessaire pour les méthodes d'analyse employées.

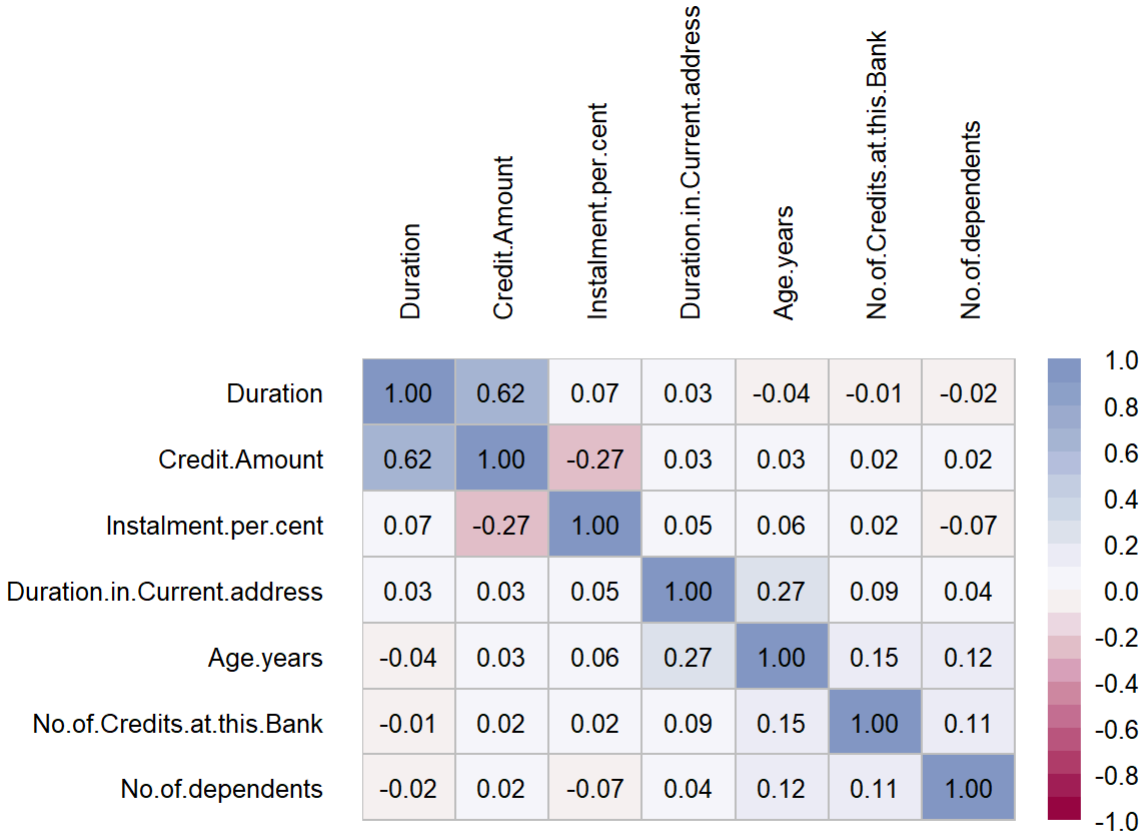
4.2.2 Lien entre variables explicatives

Des liaisons trop fortes entre variables explicatives peuvent conduire à de grande instabilité dans les modèles. L'analyse des liaisons entre variables explicatives permettra de détecter les couples de variables les plus liées.

4.2.2.1 Variables quantitatives

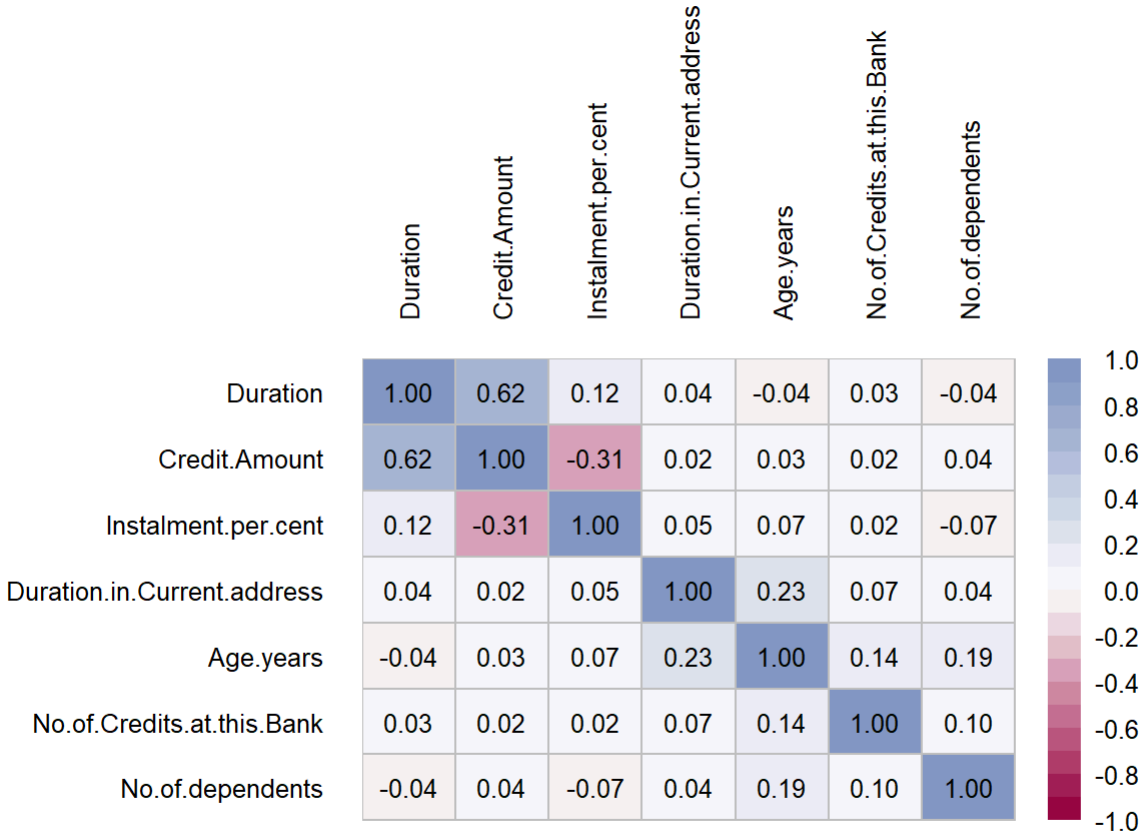
On détermine les valeurs des coefficients de corrélation linéaire et de Spearman entre les variables quantitatives.

```
matcor <- cor(don[, var.numeric])
PlotCorr(matcor, mar = c(3, 10, 10, 8))
text(x = rep(1:ncol(matcor), ncol(matcor)),
     y = rep(1:ncol(matcor), each = ncol(matcor)),
     label = sprintf("%0.2f", matcor[,ncol(matcor):1]),
     cex = 0.8,
     xpd = TRUE)
```



vince/2025-02-17

```
matcor <- cor(don[,var.numeric],method = "spearman")
PlotCorr(matcor, mar = c(3, 10, 10, 8))
text(x = rep(1:ncol(matcor), ncol(matcor)),
     y = rep(1:ncol(matcor), each = ncol(matcor)),
     label = sprintf("%0.2f", matcor[, ncol(matcor):1]),
     cex = 0.8,
     xpd = TRUE)
```



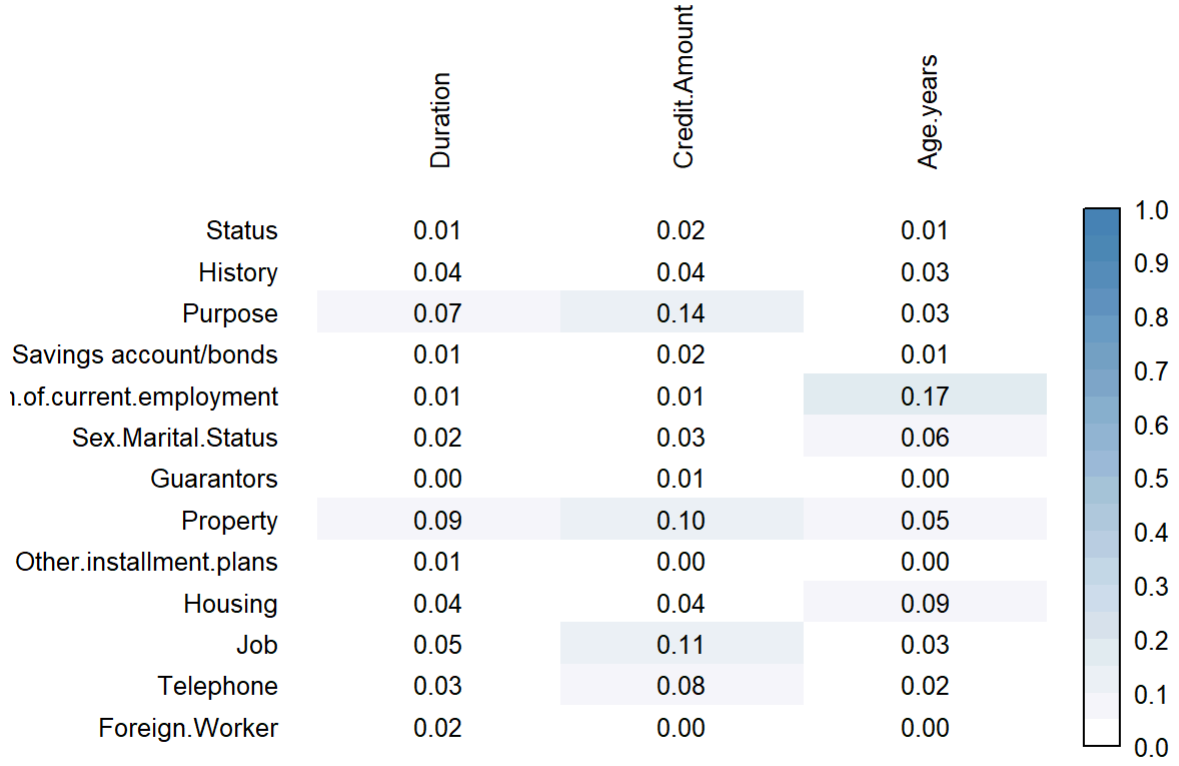
vince/2025-02-17

Les coefficients de corrélation et de Spearman sont plutôt proches pour les différents couples de variables. Dans le cas contraire, on aurait essayé de comprendre cette différence en analysant la forme des nuages de points pour les couples. Notons que l'on aurait pu également comparer le coefficient de corrélation au carré et le η^2. On constate que les liaisons ne sont pas très fortes, on ne s'attend donc pas à des problèmes de colinéarité entre ces variables.

4.2.2.2 Variables quantitatives et qualitatives

De même, on calcule le η^2 entre les variables quantitatives et les variables qualitatives.

```r
# creation d'une matrice vide avec en ligne les variables quantitatives et en colonne les var
iables qualitatives
mateta2 <- matrix(NA,13,3)
rownames(mateta2) <- c("Status", "History", "Purpose", "Savings account/bonds", "Length.of.cu
rrent.employment",
"Sex.Marital.Status", "Guarantors", "Property", "Other.installment.plans",
"Housing", "Job", "Telephone", "Foreign.Worker")
colnames(mateta2) <- c("Duration", "Credit.Amount", "Age.years")

# calcul des différents eta carré
for(ii in seq(nrow(mateta2))){
  for(jj in seq(ncol(mateta2))){
    mateta2[ii,jj] <- eta2(don[,colnames(mateta2)[jj]],
                           don[,rownames(mateta2)[ii]])
  }
}

# affichage
PlotCorr(mateta2,
         border = NA,
         cols = colorRampPalette(c("white", "steelblue"), space = "rgb")(20),
         breaks=seq(0, 1, length=21),
         args.colorlegend = list(labels=sprintf("%.1f", seq(0, 1, length = 11)), frame = TRU
E))
text(x = rep(1:ncol(mateta2), each=nrow(mateta2)),
     y = rep(nrow(mateta2):1, ncol(mateta2)),
     label = sprintf("%0.2f", mateta2[,1:ncol(mateta2)]),
     cex = 0.8,
     xpd = TRUE)
```



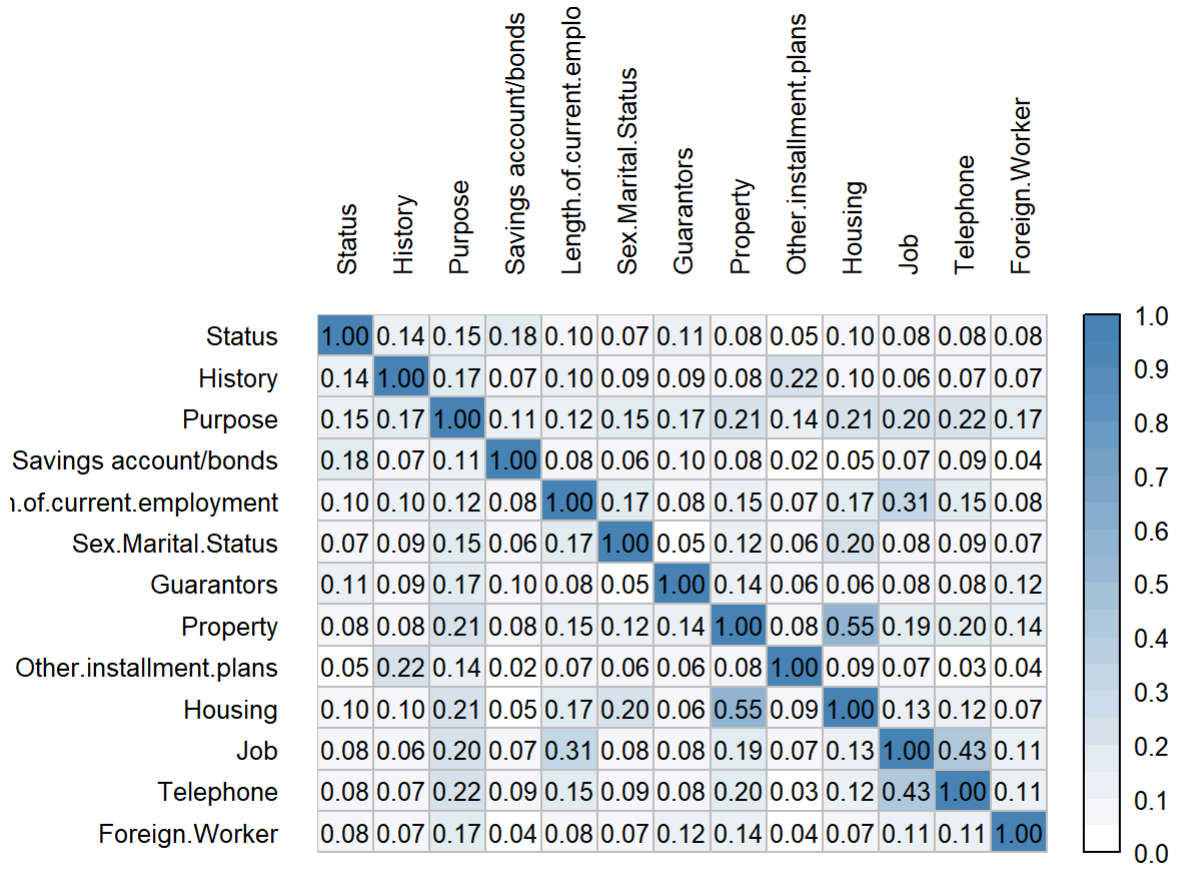
vince/2025-02-17

Les liaisons entre variables explicatives quantitatives et qualitatives semblent plutôt ténues.

4.2.2.3 Variables qualitatives

De la même façon, on détermine les V de Cramer entre les variables explicatives qualitatives.

```
matcram <- PairApply(don[,var.expl.quali], CramerV, symmetric = TRUE)
PlotCorr(matcram,
         cols = colorRampPalette(c("white", "steelblue"), space = "rgb")(20),
         breaks = seq(0, 1, length=21),
         args.colorlegend = list(labels=sprintf("%.1f", seq(0, 1, length = 11)), frame=TRUE))
text(x = rep(1:ncol(matcram), ncol(matcram)),
     y = rep(1:ncol(matcram),each=ncol(matcram)),
     label = sprintf("%0.2f", matcram[,ncol(matcram):1]),
     cex = 0.8,
     xpd = TRUE)
```



vince/2025-02-17

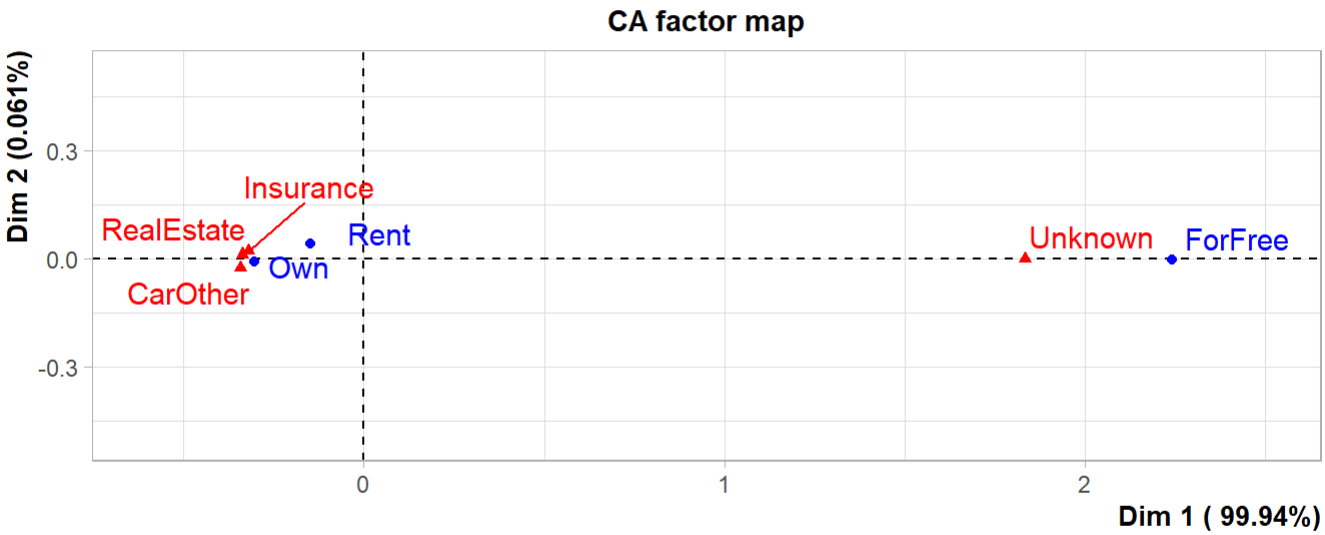
La liaison la plus forte porte sur le couple de variable *Housing* et *Property*. Pour mieux comprendre cette association, on peut commencer par regarder la table de contingence

```
cont.table <- table(don[,c("Housing","Property")])
cont.table
```

```
##          Property
## Housing   RealEstate Insurance CarOther Unknown
##   Rent            55        46       60      18
##   Own            226       184      271      32
##   ForFree          1         2        1     104
```

Clairement, il existe une association très forte, entre les modalités *ForFree* de la variable *Housing* et *Unknown* de la variable *Property*. Pour aller plus loin, on peut effectuer une analyse factorielle des correspondances (AFC) entre les deux variables.

```
res.CA <- CA(cont.table)
```



Il apparaît clairement que la diversité des profils-lignes et colonnes se résume à la première dimension, opposant à droite les deux modalités précédemment citées et à gauche les autres. Il pourra être judicieux de remplacer ces deux variables par une composante principale unique issue d'une AFC. Notons qu'il sera nécessaire que cette composante soit de longueur égale au nombre d'individus. Pour cette raison, on effectuera l'AFC du tableau disjonctif complet (ou de façon équivalente, une ACM sur les deux variables).

4.3 Analyse multivariée

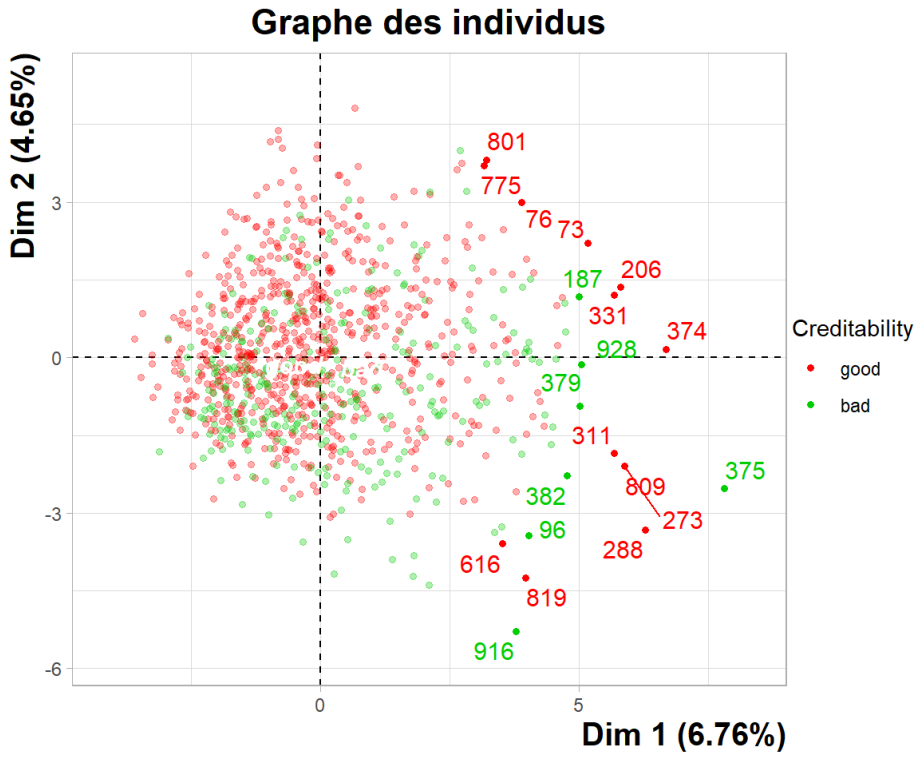
L'analyse multivariée va permettre notamment de résumer les liaisons entre les variables explicatives et d'identifier des groupes d'individus aux profils similaires. Les variables étant de natures différentes, on effectue d'abord une AFDM que l'on complètera par une CAH effectuée sur les premières composantes.

NB : l'AFDM est une méthode sensible aux modalités rares. Il conviendrait de commencer par gérer ces modalités avant d'effectuer l'analyse. Nous y reviendrons dans la section 5.2.2.

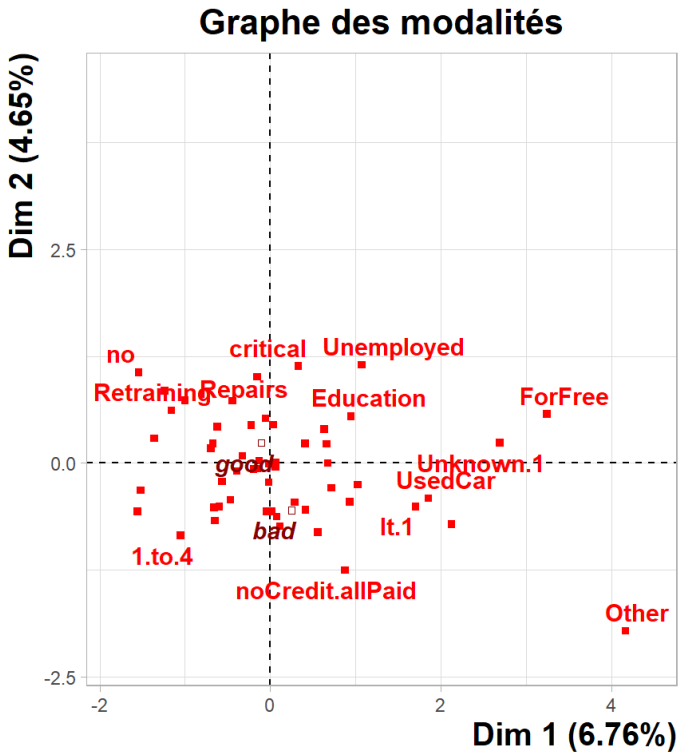
4.3.1 Analyse factorielle

On effectue l'AFDM en représentant ici le graphe des individus, celui des modalités, et celui des variables.

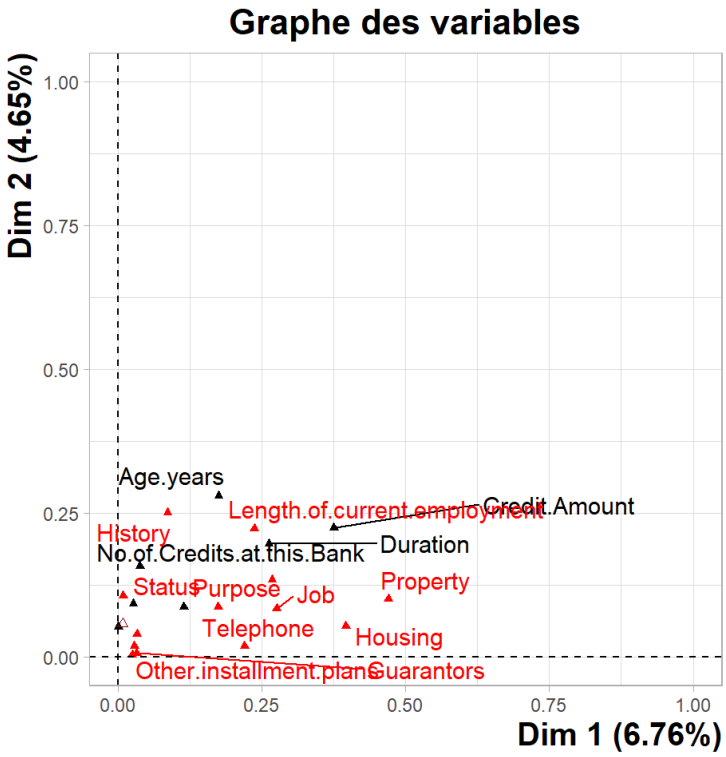
```
#On réalise l'AFDM en mettant la variable Creditability en variable illustrative.

res.famd <- FAMD(don, graph = FALSE, sup.var = ncol(don), ncp = Inf)

#On affiche les graphiques relatifs au premier plan

#individus (aux plus fortes contributions)
plot.FAMD(res.famd,
          choix = "ind",
          habillage = as.numeric(ncol(don)),
          invisible = "quali",
          select = "contrib 20",
          title = "Graphe des individus",
          cex.lab = 1.5,
          cex.main = 1.5,
          cex.axis = 1.5)
```




```
#modalités (aux plus fortes contributions)
plot.FAMD(
  res.famd,
  choix = "ind",
  invisible = "ind",
  cex.lab = 1.5,
  cex.main = 1.5,
  cex.axis = 1.5,
  col.lab = TRUE,
  title = 'Graphe des modalités'
)
```




```
#variables
plot.FAMD(
  res.famd,
  choix = "var",
  title = "Graphe des variables",
  cex.lab = 1.5,
  cex.main = 1.5,
  cex.axis = 1.5
)
```



Le graphe des individus est plutôt homogène, nous ne remarquons pas de structure particulière dans le nuage correspondant. Le graphe des modalités suggère que la variable Creditability semble peu liée à la première principale dimension de variabilité du jeu de données, mais plus à la deuxième (bien que ce lien soit

ténu). Le graphe des variables met en évidence qu'il existe des groupes de variables liées entre elles. Ceci n'est pas surprenant car il existe une structure particulière sur les variables, certaines étant relatives au type de crédit demandé, tandis que d'autres portent sur le profil financier et personnel du demandeur. L'AFDM met en avant que cette structure dans la composition des variables se traduit au niveau des données. Il pourra être pertinent d'en tenir compte en utilisant des méthodes multi-tableaux par exemple (méthodes abordées ultérieurement dans le cours).

Notons que l'on pourra utiliser le package *FactoInvestigate* pour obtenir une description automatique des différents plans.

```
install.packages("FactoInvestigate")
library(FactoInvestigate)
Investigate(res.famd, display.HCPC = FALSE)
```

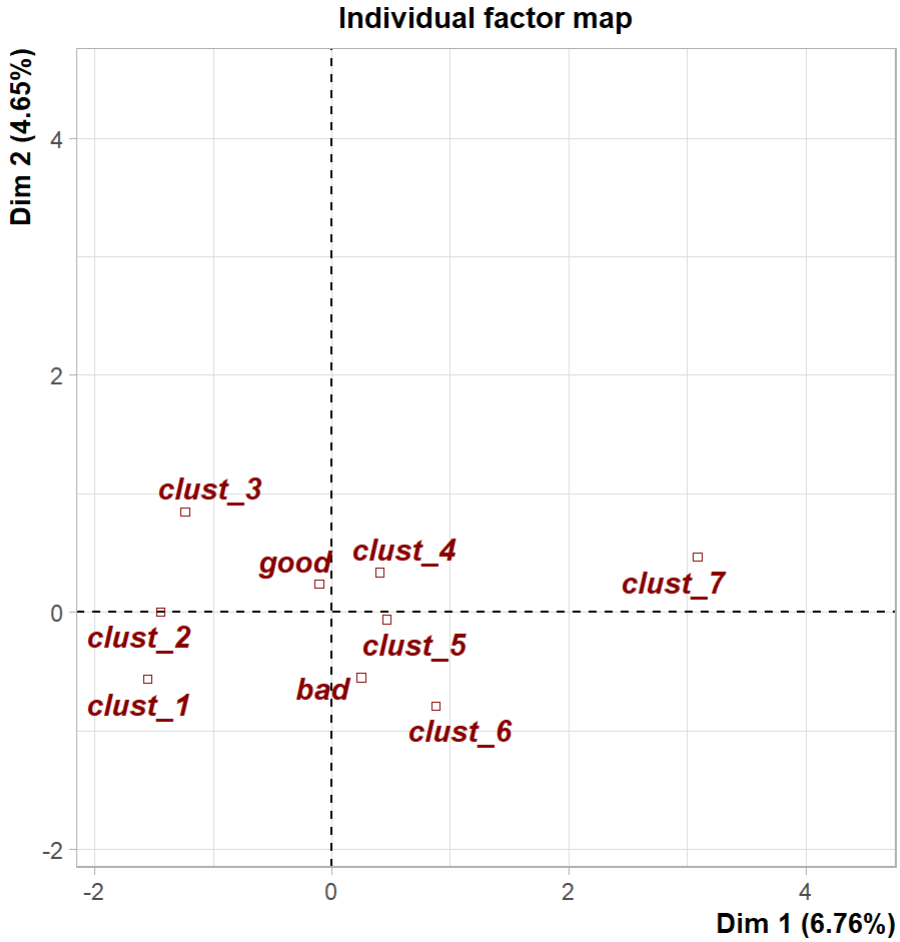
Nous n'insistons pas davantage sur l'interprétation qui n'est pas l'objet de ce document et qui par ailleurs a déjà été abordée en pré-requis.

4.3.2 Classification

On complète cette analyse par une CAH sur les composantes de l'AFDM. Pour cela, on retient les premières composantes telles que l'inertie cumulée atteigne 80% (i.e. les 31 premières).

```
# Choix du nombre de composantes
ncp <- which(res.famd$eig[,3]>80)[1]

# On effectue l'AFDM en conservant les 31 premières dimensions
res.famd <- FAMD(don, graph = FALSE, sup.var =  ncol(don), ncp = ncp)

# On effectue la CAH
res.cah <- HCPC(res.famd,
                nb.clust = -1,
                graph=FALSE,
                description = TRUE)

# On analyse le diagramme des gains d'inertie ce qui nous amène à retenir 7 classes (ignorer
le coloriage qui correspond à un choix automatique peu pertinent ici). On effectue ce choix e
t on consolide la partition obtenue.

plot.HCPC(res.cah, choice = "bar")
```


```
res.cah <- HCPC(res.famd,
                nb.clust = 7,
                graph = FALSE,
                description = FALSE,
                consol = TRUE)

# On représente les classes sur le graphe de l'AFDM

res.famd.clust <- FAMD(res.cah$data.clust,
                   graph = FALSE,
                   sup.var = c(ncol(don),#variable cible
                               ncol(res.cah$data.clust) #variable partition
                               ))

plot.FAMD(res.famd.clust,
         choix = "ind",
         invisible = c("ind","quali"))
```



On peut voir que la structure en classes est bien résumée par la première dimension, faisant apparaître un continuum entre les classes.

On pourra générer un premier descriptif des classes à l'aide des variables via la fonction *Investigate* et affiner celui-ci à l'aide de la fonction *catdes*.

```
Investigate(res.FAMD, nclust = 7, ncp = 31)
catdes(res.cah$data.clust,
       num.var = ncol(res.cah$data.clust))
```

5 Pré-traitement

5.1 Transformations

5.1.1 Variables quantitatives

Une des transformations les plus couramment appliquée aux variables quantitatives est la discrétisation. Celle-ci permettra notamment d'harmoniser la nature des variables. Par ailleurs, il arrive fréquemment que les modèles statistiques reposent sur une hypothèse de normalité des variables. Quand cette hypothèse n'est pas vérifiée, on peut souhaiter effectuer une transformation des variables pour s'y ramener. De même, quand la relation entre une variable explicative continue et une variables réponse n'est pas linéaire, on pourra préférer découper cette variable explicative en classes, en particulier quand le lien n'est pas monotone (voir Section 4.2.1.1).

5.1.1.1 Découpage en classes

Comme pour la détermination des classes d'un histogramme, le découpage en classes d'une variable quantitative peut être effectué de différentes façons. En particulier, on peut faire un découpage ``métier'', en choisissant des classes classiques pour le critère mesuré, ou par des approches plus statistiques, notamment selon les quantiles. Dans le cas d'un problème de classification supervisée, il sera préférable d'utiliser l'algorithme MDLPC de Fayyad et Irani (Fayyad and Irani (1993)), disponible dans le package *discretization*.

```
install.packages("discretization")
library(discretization)
res.mdlp <- mdlp(don[,c("Duration", "Credit.Amount", "Age.years","Creditability")])
str(res.mdlp$Disc.data)
```

```
## 'data.frame':    1000 obs. of  4 variables:
##  $ Duration     : int  1 2 1 2 2 2 2 2 1 2 ...
##  $ Credit.Amount: int  1 2 1 2 2 2 1 2 1 2 ...
##  $ Age.years    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Creditability: Factor w/ 2 levels "good","bad": 1 2 1 1 2 1 1 1 1 2 ...
```

Les variables *Duration* et *Credit.Amount* ont été discrétisées en deux classes, tandis qu'une seule classe a été considérée pour la variable *Age.years*, ce qui n'est d'aucun intérêt pour l'analyse. Nous reviendrons sur le découpage de cette variable dans la Section 5.1.1.2.

On peut vérifier que la discrétisation préserve la liaison avec la variable réponse à l'aide d'un test du chi-deux.

```
sapply(c("Duration", "Credit.Amount"),
       FUN=function(var,res.mdlp){
         chisq.test(res.mdlp$Disc.data[,var],res.mdlp$Disc.data$Creditability)$p.value
       },
       res.mdlp = res.mdlp)
```

```
##      Duration Credit.Amount
##  2.923046e-08  3.171735e-07
```

Les p-valeurs obtenurs traduisent une liaison claire des variables *Duration* et *Credit.Amount* avec la variable réponse. On inclut ces deux nouvelles variables dans le jeu de données

```
don$Duration.cat <- res.mdlp$Disc.data$Duration
don$Credit.Amount.cat <- res.mdlp$Disc.data$Credit.Amount
```

5.1.1.2 Linéarité

Comme évoqué en Section 4.2.1.1, la liaison entre la variable *Age.years* et *Creditability* n'est pas linéaire. On effectue un découpage de la variable *Age.years* en 3 classes pour gérer cette non-linéarité.

```
don$Age.years.cat <- cut(don$Age.years,breaks = c(0,24,35,100))
```

et on vérifie graphiquement que la linéarité a été améliorée

```
cont.table <- table(don$Age.years.cat,don$Creditability)
prof.lignes <- prop.table(cont.table,1)
res.binom.test <- mapply(cont.table[,1],
                         FUN=binom.test,
                         n=rowSums(cont.table),
                         SIMPLIFY = FALSE)
ci <- sapply(res.binom.test,"[[","conf.int")
abscisses <- c((19+24)/2,(24+35)/2,(35+75)/2)

plot(abscisses,
     prof.lignes[,1],
     pch = 16,
     col = 1,
     xlab = "Age",
     ylab = "Proportion de bons payeurs",
     ylim = c(0,1))

for(ii in 1:length(abscisses)){
  segments(x0 = abscisses[ii],
           y0 = ci[1,ii],
           x1 = abscisses[ii],
           y1 = ci[2,ii],
           col = 1)
}
```

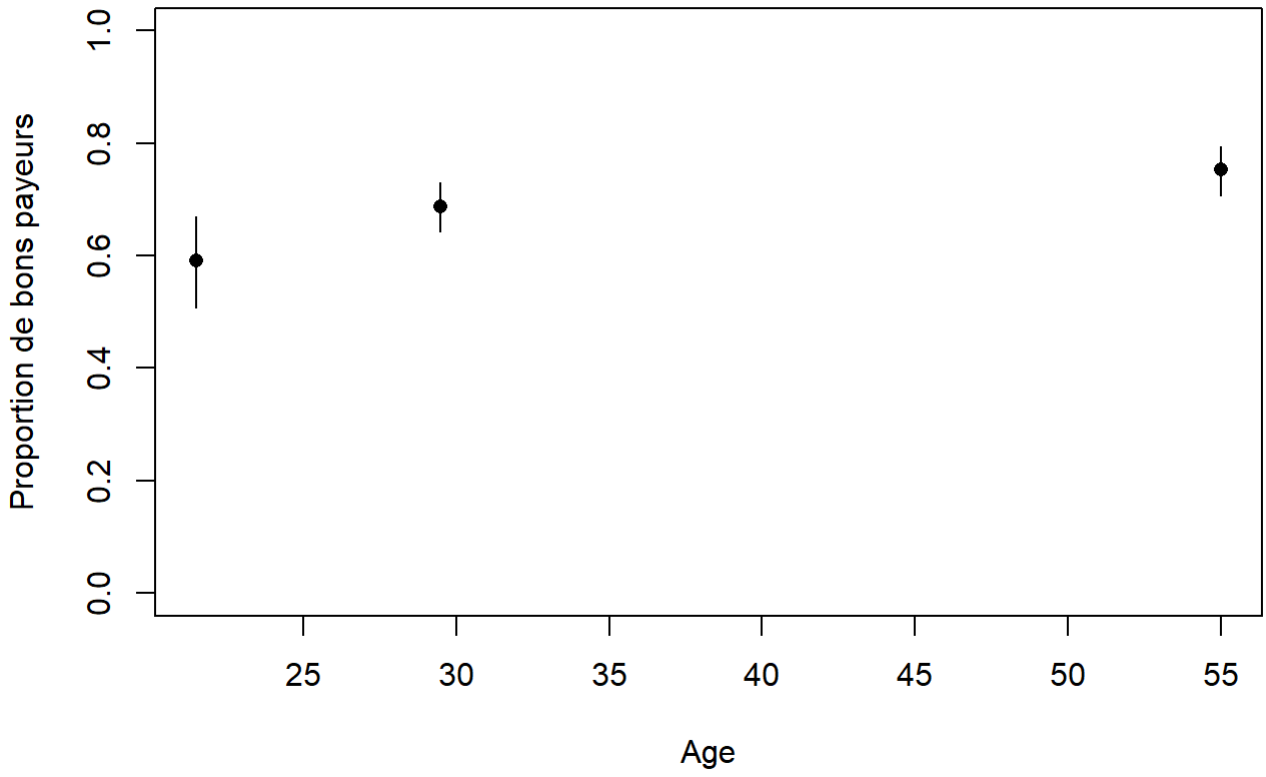


Figure 5.1: Proportion de bons payeurs en fonction de l'âge après discrétisation en 3 classes.

Par ailleurs, on peut vérifier que cette liaison est bien statistiquement significative

```
chisq.test(don$Age.years.cat, don$Creditability)$p.value
```

```
## [1] 0.0008298968
```

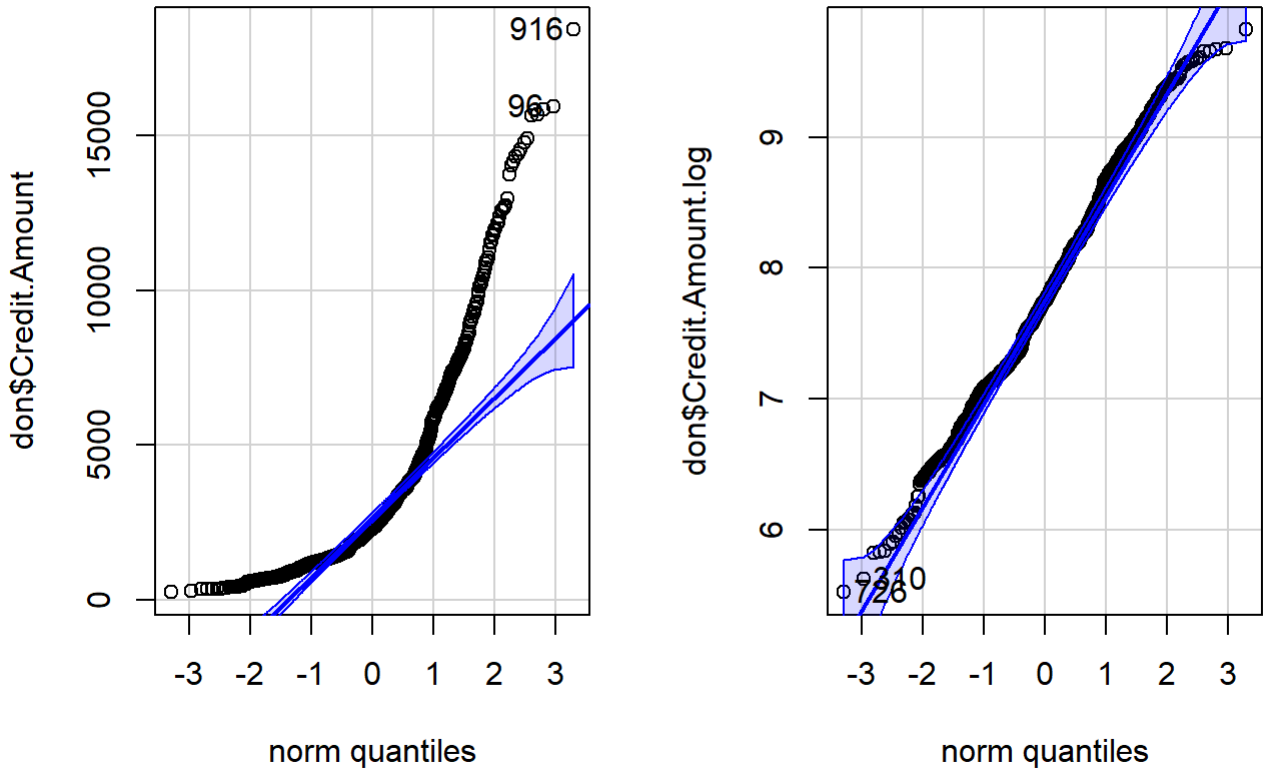
5.1.1.3 Normalité

Nous avons vu en Section 4.1.1.2 que les variables *Credit.Amount*, *Duration* et *Age.years* n'étaient pas normales. On peut se ramener à la normalité en effectuant une transformation logarithmique pour la variable *Credit.Amount*.

```
don$Credit.Amount.log <- log(don$Credit.Amount)
```

On compare les droites de Henry avant et après transformation pour vérifier l'amélioration

```
par(mfrow=c(1,2))
qqPlot(don$Credit.Amount)
qqPlot(don$Credit.Amount.log)
```



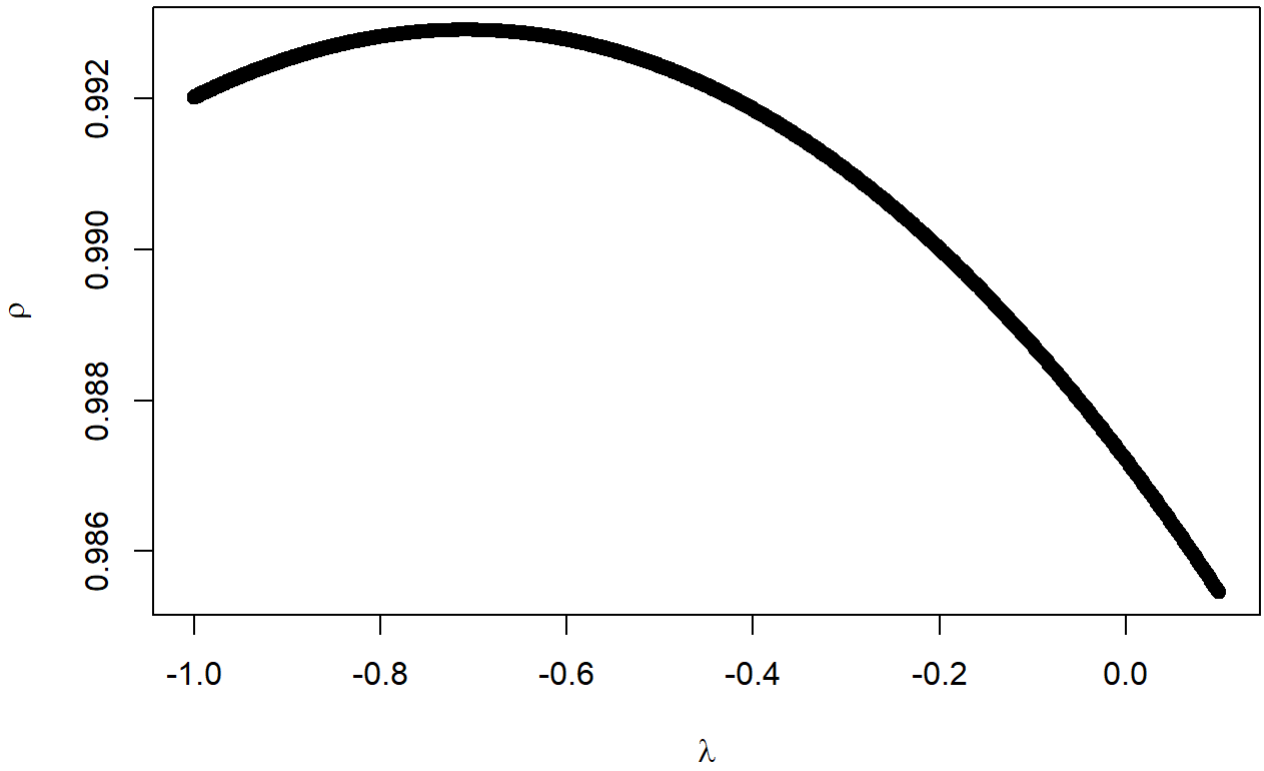
Le résultat est satisfaisant. On peut aussi vérifier que le coefficient d'asymétrie est plus proche de 0.

```
#coefficient d'asymetrie
coefasym <- function(x){
  m <- mean(x)
  mu2 <- mean( (x-m)^2 )
  mu3 <- mean( (x-m)^3 )
  sigma <- sqrt(mu2)
  gamma1 <- mu3/sigma^3
  return(gamma1)
}

coefasym(don$Credit.Amount)
coefasym(don$Credit.Amount.log)
```

```
##     Credit.Amount Credit.Amount.log
##         1.9467020         0.1290919
```

Il paraît difficile de trouver une transformation du même type qui soit satisfaisante pour la variable *Age*, on utilise la méthode de Box-Cox.

```
# on crée une fonction qui prend en entrée une grille pour le paramètre lambda,
# ainsi que la variable à transformer et qui, pour chaque valeur de lambda,
# renvoie le coefficient de correlation entre les quantiles

myBoxCox <- function(lambda.grid,var){
  res.cor <- rep(NA,length(lambda.grid))
  comp <- 0
  probs <- seq(1/length(var),(length(var)-1)/length(var),1/length(var))
  quantilenormale <- qnorm(probs)
  for (lambda in lambda.grid){
    #on incremente un compteur
    comp <- comp+1
    #on effectue la transformation pour le lambda courant
    var.boxcox <- BoxCox(var,lambda = lambda)
    #on calcule le coefficient de corrélation entre les quantiles
    # de la variable transformée et ceux d'une loi normale
    res.cor[comp] <- cor(quantile(var.boxcox, probs = probs),
                         quantilenormale)
  }
  return(res.cor)
}


# on définit une grille et on calcule la correlation entre les quantiles
# en fonction du parametre de la grille
lambda.grid <- seq(-1,0.1,1/1000)
res.cor <- myBoxCox(lambda.grid = lambda.grid, var = don$Age.years)
# on affiche l'évolution du coefficient de corrélation en fonction de lambda
plot(lambda.grid,res.cor,
     xlab = expression(lambda),
     ylab = expression(rho))
```




```
#on identifie la valeur de lambda qui maximise la correlation
lambda.grid[which.max(res.cor)]
```

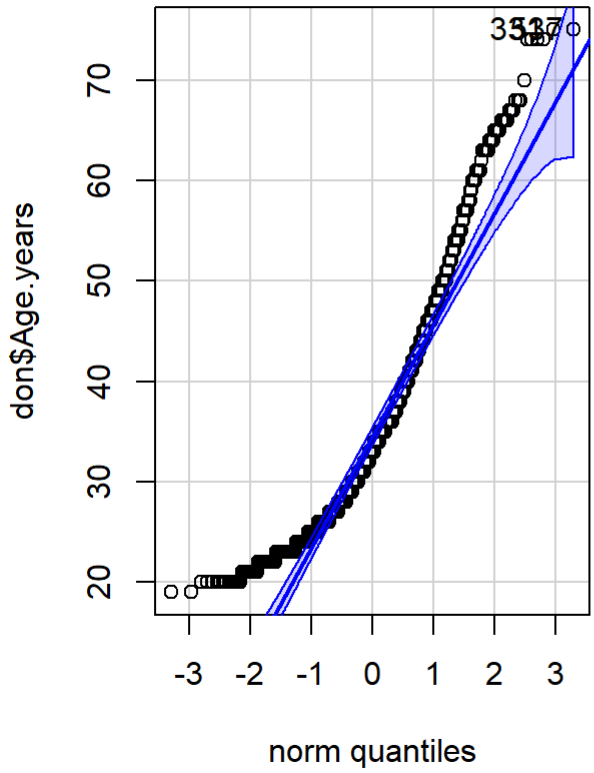
```
## [1] -0.709
```

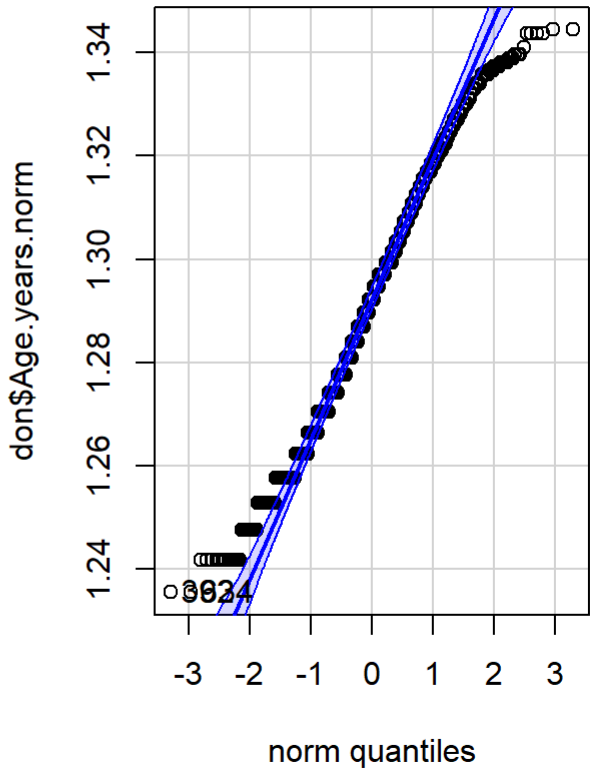
Le maximum est atteint en -0.709, on retient donc la transformation correspondante.

```
#on effectue donc la transformation pour lambda = -0.709
don$Age.years.norm <- BoxCox(don$Age.years, lambda = -0.709)
```

On vérifie ensuite la qualité de la transformation en comparant les droites de Henry et les coefficients d'asymétrie.

```
qqPlot(don$Age.years)
qqPlot(don$Age.years.norm)

#on calcule le coefficient d'asymétrie
coefasym(don$Age.years)
coefasym(don$Age.years.norm)
```

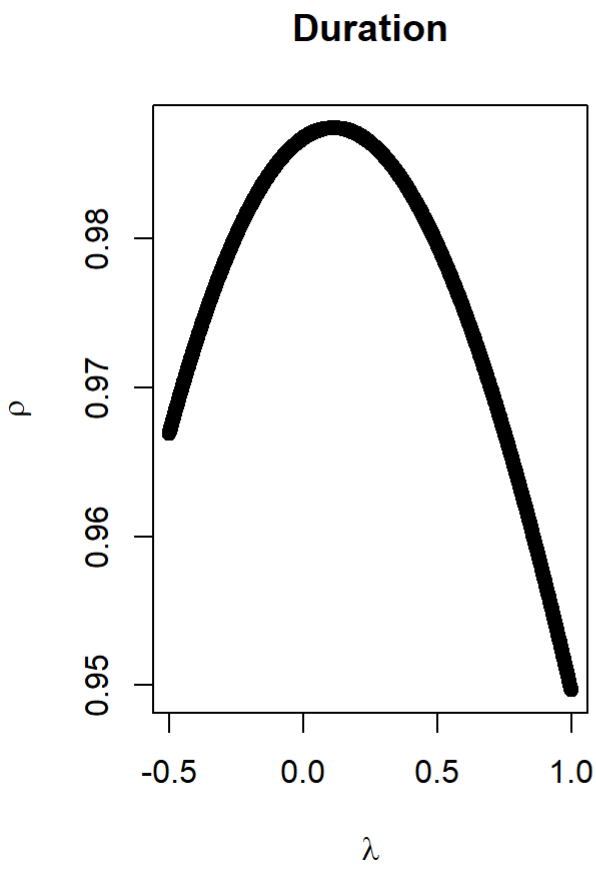


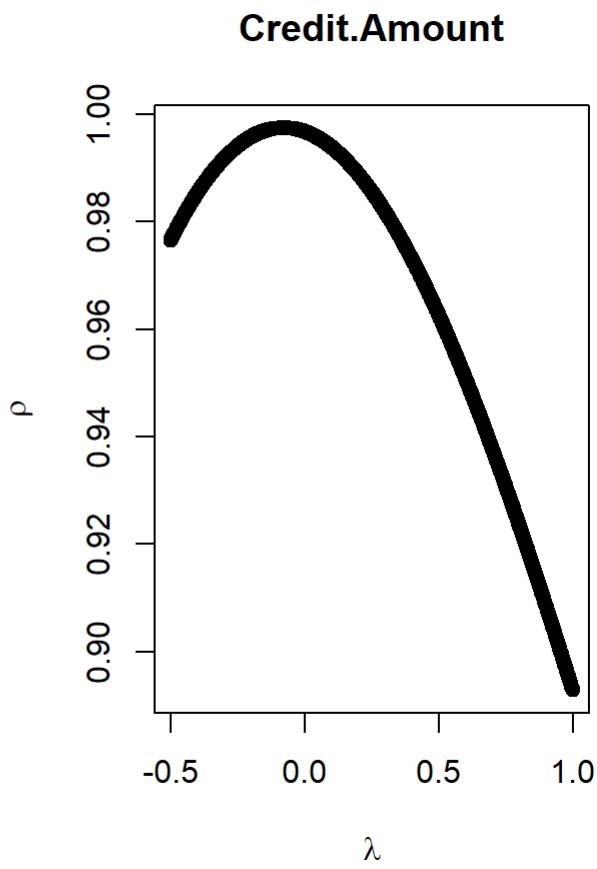



```
##      Age.years Age.years.norm
##     1.01920752     0.02737609
```

On peut appliquer la même stratégie à la variable *Duration*, voire à la variable *Credit.Amount*.

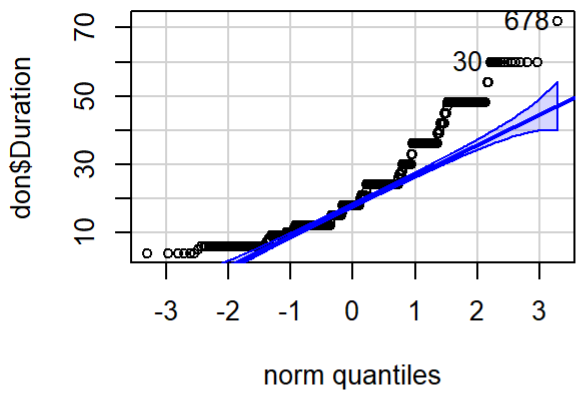
```
par(mfrow=c(1,2))

# Duration
lambda.grid <- seq(-.5,1,1/1000)
res.cor <- myBoxCox(lambda.grid = lambda.grid,var = don$Duration)
plot(lambda.grid, res.cor, xlab = expression(lambda), ylab = expression(rho), main = "Duratio
n")
don$Duration.norm <- BoxCox(don$Duration, lambda = lambda.grid[which.max(res.cor)])

#Credit.Amount
lambda.grid <- seq(-.5,1,1/1000)
res.cor <- myBoxCox(lambda.grid = lambda.grid, var = don$Credit.Amount)
plot(lambda.grid, res.cor, xlab=expression(lambda), ylab = expression(rho), main = "Credit.Am
ount")
```

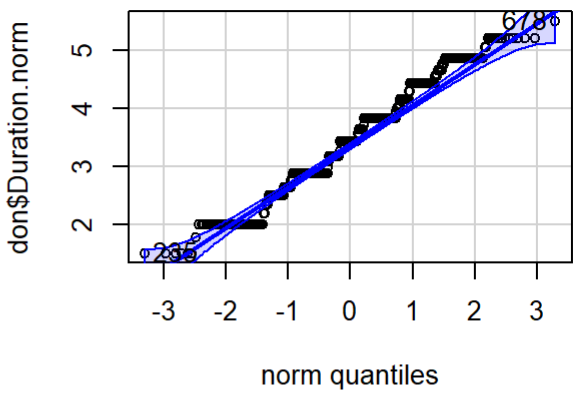


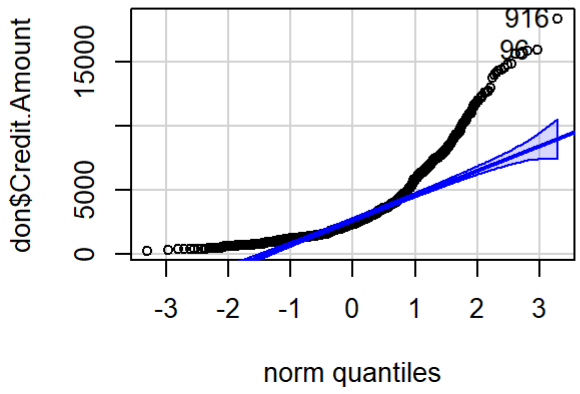


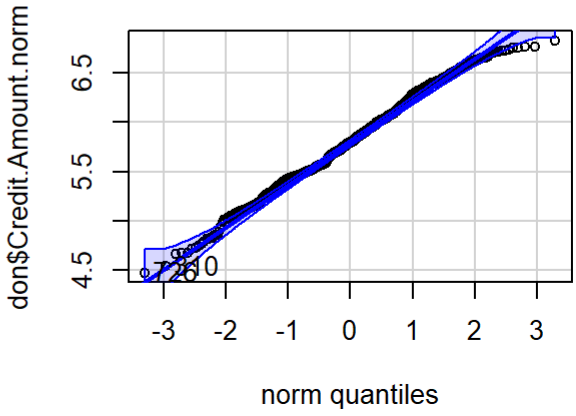

```
don$Credit.Amount.norm <- BoxCox(don$Credit.Amount, lambda = lambda.grid[which.max(res.cor)])
```

```
par(mfrow=c(2,2), mar = c(4,5, 3, 2) + 0.1)
qqPlot(don$Duration)
qqPlot(don$Duration.norm)
qqPlot(don$Credit.Amount)
qqPlot(don$Credit.Amount.norm)
```










```
sapply(don[,c("Credit.Amount","Credit.Amount.log","Credit.Amount.norm")],coefasym)
```

```
##      Credit.Amount  Credit.Amount.log Credit.Amount.norm
##         1.94670202         0.12909188        -0.02401317
```

Bien que la transformation logarithmique de la variable Credit.Amount soit satisfaisante, nous retiendrons plutôt celle de Box-Cox pour laquelle le coefficient d'asymétrie est plus proche de 0.

Remarquons qu'il existe d'autres façons de choisir le paramètre λ. En particulier, la fonction *BoxCoxLambda* du package *DescTools* propose une estimation par maximum de vraisemblance.

5.1.2 Variables qualitatives

Différentes méthodes sont souvent mises en défaut en présence de modalités rares comme les modèles de régression logistique, l'AFDM ou l'ACM qui est fortement influencée par ce type de données. Une stratégie de pré-traitement consiste alors à fusionner les modalités de faible effectif. L'étude exploratoire univariée effectuée en Section 4.1.2 a permis d'identifier des modalités rares sur les variables *Status*, *History*, *Purpose*, *Savings account/bonds*,*Length.of.current.employment*,*Sex.Marital.Status*, *Other.installment.plans*,*Job*, tandis que l'étude bivariée effectuée en Section 4.2.1.3 a permis d'identifier les modalités pour lesquelles la distribution de la variable réponse est similaire. A partir de là, on peut décider de fusionner les modalités suivantes :

- *History* : noCredit.allPaid et thisBank.AllPaid
- *Purpose* : Other, Education, Retraining
- *Purpose* : DomesticAppliance, Repairs
- *Savings account/bonds* : gt.1000 et 500.to.1000
- *Sex.Marital.Status* : Male.Divorced.Seperated et Male.Single
- *Guarantor* : CoApplicant et None

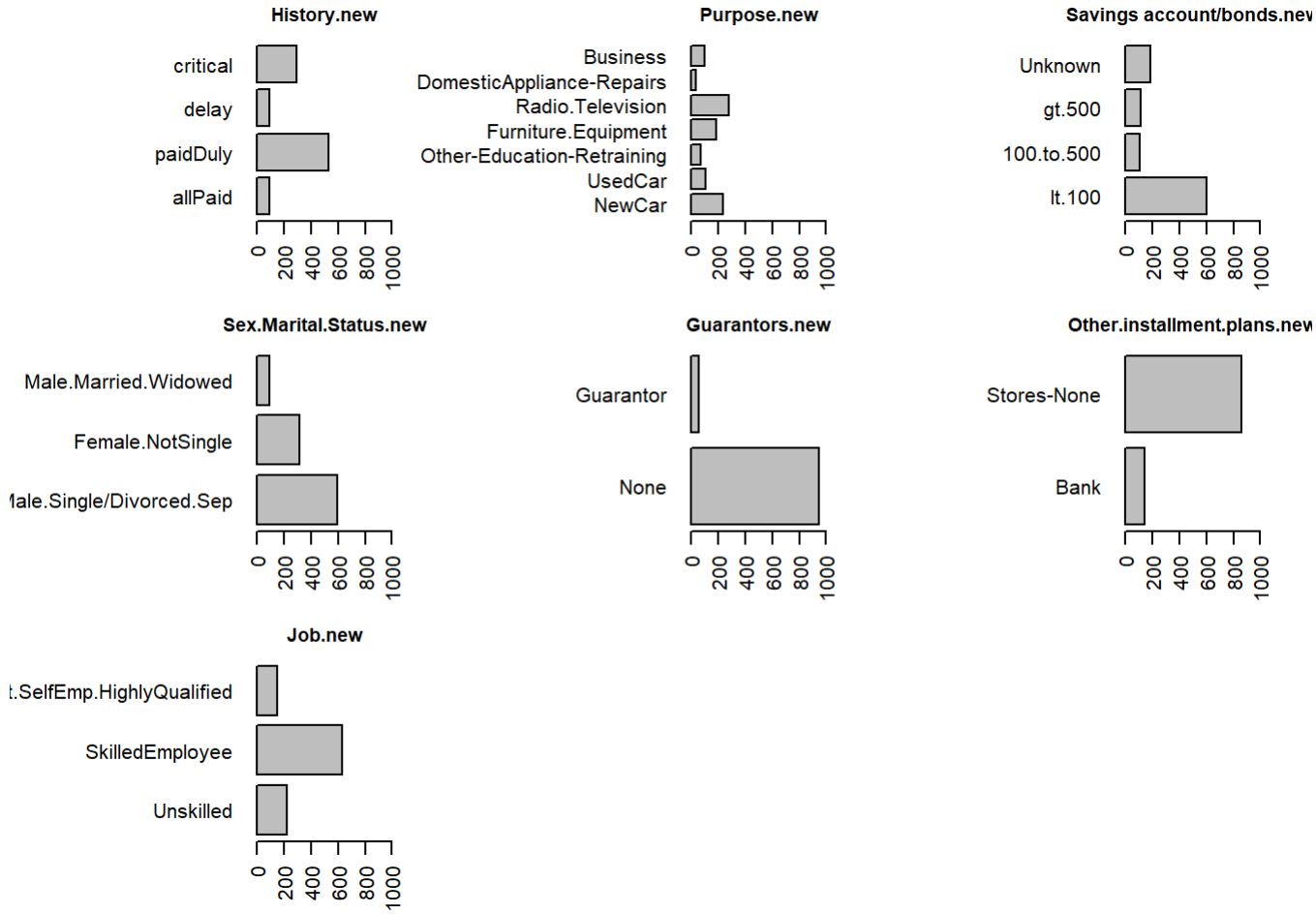
- *Other.installment.plans* : Stores et None
- *Job* : UnemployedUnskilled et UnskilledResident

On décide également d'enlever la variable *Foreign.Worker* dans la mesure où elle ne comporte que deux modalités, dont une rare et qu'elle ne semble pas très discriminante (cf Section 4.2.1.2).

```
# creation de nouvelles variables
don$History.new <- don$History
don$Purpose.new <- don$Purpose
don$`Savings account/bonds.new` <- don$`Savings account/bonds`
don$Sex.Marital.Status.new <- don$Sex.Marital.Status
don$Guarantors.new <- don$Guarantors
don$Other.installment.plans.new <- don$Other.installment.plans
don$Job.new <- don$Job

# fusion des modalités
levels(don$History.new) <- c("allPaid", "allPaid", "paidDuly", "delay",
                             "critical")
levels(don$Purpose.new) <- c("NewCar", "UsedCar", "Other-Education-Retraining", "Furniture.Eq
uipment", "Radio.Television",
"DomesticAppliance-Repairs", "DomesticAppliance-Repairs", "Other-Education-Retraining", "Othe
r-Education-Retraining", "Business")
levels(don$`Savings account/bonds.new`) <- c("lt.100",
                                             "100.to.500", "gt.500", "gt.500", "Unknown")
levels(don$Sex.Marital.Status.new) <- c("Male.Single/Divorced.Sep", "Female.NotSingle", "Mal
e.Single/Divorced.Sep", "Male.Married.Widowed")
levels(don$Guarantors.new) <- c("None", "None", "Guarantor")
levels(don$Other.installment.plans.new) <- c("Bank","Stores-None", "Stores-None")
levels(don$Job.new) <- c("Unskilled", "Unskilled", "SkilledEmployee",
                         "Management.SelfEmp.HighlyQualified")

# suppression de la variables Foreign. Worker
don$Foreign.Worker <- NULL
```

On voit que les modalités rares ont été correctement pré-traitées

```
par(mfrow=c(3,3), mar = c(3, 10, 2, 2) + 0.1)
mapply(don[,colnames(don)%in%paste0(names(var.factor), ".new")],
       FUN = function(xx,name){barplot(table(xx), main = name, horiz = TRUE, las = 2, xlim =
c(0,1000), cex.main = .9)},
       name = colnames(don)[colnames(don)%in%paste0(names(var.factor), ".new")])
```



5.2 Réduction des données

5.2.1 En lignes

Ayant déjà abordé la classification en Section 4.3.2, nous montrons ici comment réduire le nombre de lignes en effectuant un échantillonnage (simple, systématique ou stratifié).

```
# échantillonnage simple
set.seed(0)
ech.simple <- sample(seq(nrow(don)),size=500)

# échantillonnage systématique
set.seed(0)
ech.syst <- seq(1,nrow(don),2)

# échantillonnage stratifié sur la réponse (proportionnel)
set.seed(0)

## on identifie les bons payeurs (strate1) et les mauvais (strate0)

strate1 <- which(don$Creditability=="good")
strate0 <- which(don$Creditability!="good")

## dans chaque strate, on tire la moitiée des individus au hasard

ech.strat.1 <- strate1[sample(seq(length(strate1)),
                              size = ceiling(length(strate1)/2))]
ech.strat.0 <- strate0[sample(seq(length(strate0)),
                              size = ceiling(length(strate0)/2))]
## les données issues de chaque strate sont alors agrégées
ech.strat <- c(ech.strat.1,ech.strat.0)
```

Il est toujours utile de vérifier ensuite que l'échantillon est bien représentatif du jeu de données initial. Pour cela, on peut utiliser la fonction *catdes*. Afin de tenir compte des problèmes de multiplicité des tests, on applique une correction de Bonferroni (cf Wikistat (2019)).

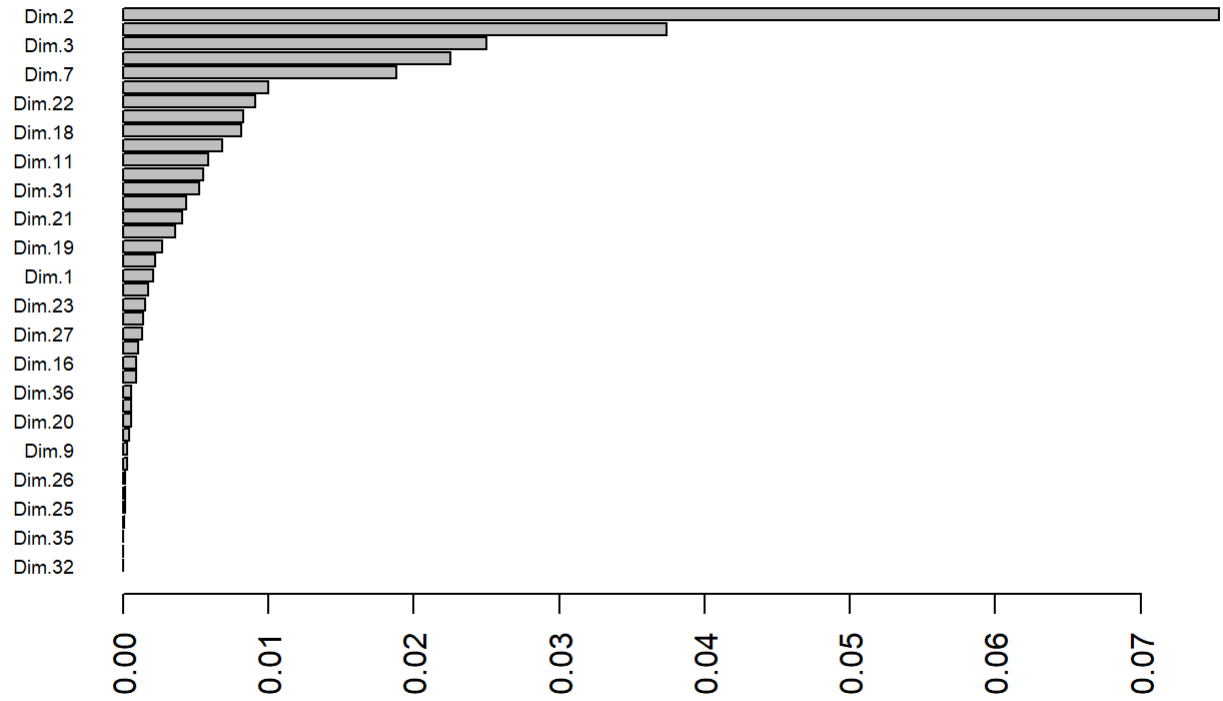
```
var.simple <- factor(seq(nrow(don))%in%ech.simple)
catdes(cbind.data.frame(don,var.simple),ncol(don)+1,proba = 0.05/ncol(don))

var.syst <- factor(seq(nrow(don))%in%ech.syst)
catdes(cbind.data.frame(don,var.syst),ncol(don)+1,proba = 0.05/ncol(don))

var.strat <- factor(seq(nrow(don))%in%ech.strat)
catdes(cbind.data.frame(don,var.strat),ncol(don)+1,proba = 0.05/ncol(don))
```

Pour chaque méthode d'échantillonnage, aucune des variables ne permet de discriminer les individus séléctionnés de ceux présents dans le jeu de données initial. On peut donc considérer que chacun des échantillons est représentatif de celui-ci.

5.2.2 En colonnes

Etant donné que nous sommes dans une problématique supervisée, la réduction du nombre de colonnes doit être effectuée en tenant compte de la variable réponse. Au vu des méthodes abordées jusqu'ici dans le cours, nous nous limiterons à proposer une réduction du nombre de colonnes en utilisant les méthodes factorielles, en l'occurrence l'AFDM, bien que des approches de type PLS seraient préférables. On recherche donc à identifier les composantes les plus liées à la variable réponse. Pour cela, on les hiérarchise selon le rapport de corrélation. Notons que nous avons commencé à prétraiter les données, il convient de refaire l'AFDM à partir des nouvelles données, notamment pour gérer le problème de sensibilité aux modalités rares.

```
don.famd <- don[,c("Status",
"Length.of.current.employment", "Instalment.per.cent",
"Duration.in.Current.address",
"Property", "Housing",
"No.of.Credits.at.this.Bank", "No.of.dependents", "Telephone",
"Duration.cat", "Credit.Amount.cat", "Age.years.cat",
"History.new", "Purpose.new", "Savings account/bonds.new", "Sex.Marital.Status.new",
"Guarantors.new", "Other.installment.plans.new", "Job.new","Creditability")]


res.famd <- FAMD(don.famd,
             graph = FALSE,
             sup.var = ncol(don.famd),
             ncp = Inf)

ordre <- order(res.famd$quali.sup$eta2)
barplot(res.famd$quali.sup$eta2[ordre],
        names.arg = colnames(res.famd$quali.sup$eta2)[ordre],
        las = 2,
        horiz = TRUE,
        cex.names = .6)
```



En fonction du nombre de colonnes désiré, on retiendra un certain nombre de composantes parmi les plus liées. En plus de limiter le nombre de colonnes, cette opération rendra les données quantitatives et décorrélées.

6 Conclusion

La fouille des données est un processus itératif. A ce stade, nous ne savons pas encore précisément quelles seront les méthodes supervisées que nous appliquerons sur les données. Or, ceci est un point important pour définir un prétraitement optimal. Par conséquent, il sera probablement nécessaire de revenir sur cette étape par la suite.

Le logiciel R nous a permis de mettre en oeuvre les différentes méthodes de pré-traitement. Nous avons notamment pu proposer des graphiques avancés mettant en évidence l'information souhaitée. Les sorties graphiques sont de qualité, mais on peut aller encore plus loin en utilisant notamment le package *ggplot2*. Nous avons aussi pu générer un grand nombre de sorties à partir d'un nombre réduit de lignes de code, notamment en utilisant les fonctions avancées (*mapply*, *tapply*, *sapply*, etc), mais un utilisateur plus novice pourra toujours générer les sorties une à une en appelant à chaque fois les fonctions dont il a besoin.

Références

Fayyad, Usama M., and Keki B. Irani. 1993. "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning." In *13th International Joint Conference on Artificial Intelligence*, 1022–29.

Lebart, L., A. Morineau, and M. Piron. 2006. *Statistique Exploratoire Multidimensionnelle: Visualisations Et Inférences En Fouille de Données*. Sciences Sup. Mathématiques. Dunod. https://horizon.documentation.ird.fr/exl-doc/pleins_textes/divers11-10/010007837.pdf (https://horizon.documentation.ird.fr/exl-doc/pleins_textes/divers11-10/010007837.pdf).

Lejeune, Michel. 2010. *Statistique : La Théorie Et Ses Applications*. Statistique Et Probabilités Appliquées. Springer.

Tufféry, S. 2007. *Data Mining Et Statistique décisionnelle: L'intelligence Des Données*. Editions Technip.

Wikistat. 2019. "A Propos de La Méthode Bonferroni — Wikistat." https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-modmixt8-bonfer.pdf (https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-modmixt8-bonfer.pdf).