STA211 – Entreposage et fouille de données
Rapport du challenge

Daniel PONT
pont.daniel@gmail.com
Tél. : 06 81 71 64 31
5 juillet 2019

Table des matières

1. Introduction

Ce rapport résume l'étude effectuée dans le cadre du challenge de l'UE STA211 du CNAM au 2ème semestre 2018/2019. Le challenge porte sur des données de recensement extraites des enquêtes sur la population américaine de 1994 et de 1995, menées par le US Census Bureau. Les données contiennent 40 variables démographiques ou liées à l'emploi et portent sur un ensemble d'environ 300000 individus. La variable à prédire est la variable binaire *outcome* dont les modalités sont -50000 et +50000, indiquant que le citoyen gagne plus ou moins de 50K USD par an. Les données sont fournies sous la forme d'un échantillon d'apprentissage et d'un échantillon test. L'échantillon d'apprentissage représente 2/3 des données, tandis que l'échantillon test en représente 1/3. L'échantillonnage a été stratifié selon la variable outcome. Le présent rapport répond aux deux objectifs du challenge :

1. Présenter les analyses exploratoires et pré traitements des données d'apprentissage (correction de valeurs manquantes, sélection des variables explicatives…).

2. Prédire la variable réponse à l'aide des méthodes étudiées en cours et comparer les performances de ces méthodes.

3. Importation des données

```
load("data_train.rda")
load("data_test.rda")
str(data_train)
```

```
## 'data.frame': 199526 obs. of 41 variables:
## $ age : int 37 4 71 8 12 37 14 28 44 23 ...
## $ class of worker : Factor w/ 9 levels " Federal government",..: 1 4 4 4 4 5 4 5 2 5 ...
## $ detailed industry recode : int 47 0 0 0 0 33 0 33 43 5 ...
## $ detailed occupation recode : int 26 0 0 0 0 19 0 29 12 36 ...
## $ education : Factor w/ 17 levels " 10th grade",..: 17 11 17 11 11 13 11 17 15 13 ...
## $ wage per hour : int 0 0 0 0 0 500 0 1405 0 0 ...
## $ enroll in edu inst last wk : Factor w/ 3 levels " College or university",..: 3 3 3 3 3 3 3 3 3 3 ...
## $ marital stat : Factor w/ 7 levels " Divorced"," Married-A F spouse present",..: 3 5 7 5 5 3 5 5 5 ...
## $ major industry code : Factor w/ 24 levels " Agriculture",..: 19 15 15 15 15 20 15 20 6 11...
## $ major occupation code : Factor w/ 15 levels " Adm support including clerical",..: 1 7 7 7 ...
## $ race : Factor w/ 5 levels " Amer Indian Aleut or Eskimo",..: 5 3 5 5 4 5 5 3 5 5 ...
## $ hispanic origin : Factor w/ 10 levels " All other"," Central or South American",..: 6 1 1 1 2 1...
## $ sex : Factor w/ 2 levels " Female"," Male": 1 1 1 1 1 1 2 1 1 2 ...
## $ member of a labor union : Factor w/ 3 levels " No"," Not in universe",..: 2 2 2 2 2 1 2 1 2 2 ...
## $ reason for unemployment : Factor w/ 6 levels " Job leaver",..: 4 4 4 4 4 4 4 4 4 4 ...
## $ full or part time employment stat : Factor w/ 8 levels " Children or Armed Forces",..: 2 1 1 6 1 ...
## $ capital gains : int 0 0 0 0 0 0 0 0 3325 0 ...
## $ capital losses : int 0 0 0 0 0 0 0 0 0 0 ...
## $ from stocks : int 0 0 0 0 0 200 0 0 0 0 ...
## $ tax filer stat : Factor w/ 6 levels " Head of household",..: 3 5 6 5 5 3 5 5 6 3 ...
## $ region of previous residence : Factor w/ 6 levels " Abroad"," Midwest",..: 4 4 4 4 4 4 4 4 4 6 ...
## $ state of previous residence : Factor w/ 50 levels " Abroad"," Alabama",..: 36 36 36 36 36 6 ...
## $ detailed household and family stat : Factor w/ 38 levels " Child <18 ever marr not in subfamily",..:
## $ detailed household summary in household : Factor w/ 8 levels " Child 18 or older",..: 8 7...
## $ migration code-change in msa : Factor w/ 9 levels " Abroad to MSA",..: NA NA NA NA NA NA 5 NA 7 ...
## $ migration code-change in reg : Factor w/ 8 levels " Abroad"," Different county same state",..: NA ...
## $ migration code-move within reg : Factor w/ 9 levels " Abroad"," Different county same state",..
## $ live in this house 1 year ago : Factor w/ 3 levels " No"," Not in universe under 1 year old",..
## $ migration prev res in sunbelt : Factor w/ 3 levels " No"," Not in universe",..: NA NA NA ...
## $ num persons worked for employer : int 6 0 0 0 0 1 0 0 6 4 ...
## $ family members under 18 : Factor w/ 5 levels " Both parents present",..: 5 2 5 1 1 5 3 ...
```

```
## $ country of birth father : Factor w/ 42 levels " Cambodia"," Canada",..: 40 40 40 40 8 40 16 40..
## $ country of birth mother : Factor w/ 42 levels " Cambodia"," Canada",..: 26 40 40 40 8 40 ...
## $ country of birth self : Factor w/ 42 levels " Cambodia"," Canada",..: 40 40 40 40 8 40 ...
## $ citizenship : Factor w/ 5 levels " Foreign born- Not a citizen of U S ",..: 5 5 5 5 1 5 1...
## $ business or self employed : int 0 0 0 0 0 2 0 0 0 0 ...
## $ fill inc questionnaire for veteran's admi: Factor w/ 3 levels " No"," Not in universe",..: 2 2 2
## $ veterans benefits : int 2 0 2 0 0 2 0 2 2 2 ...
## $ weeks worked in year : int 52 0 0 0 0 52 0 0 52 52 ...
## $ year : int 95 95 95 95 95 95 95 94 95 94 ...
## $ outcome : Factor w/ 2 levels "-50000","+50000": 1 1 1 1 1 1 1 1 1 1 ...
```

Les noms des variables sont explicites mais comportent des espaces ce qui peut poser un problème dans les traitements ultérieurs. On les corrige de façon à obtenir une syntaxe valide en R :

```
colnames(data_train) <- make.names(names(data_train))
```

On transforme ensuite les variables qualitatives (*detailed.industry.recode*, *detailed.occupation.recode*, etc.) de type “integer” en “factor”. Les différentes modalités sont listées dans ce document :

https://www2.census.gov/programs-surveys/cps/techdocs/cpsmar95.pdf

```
not.in.univ_yes_no <- c("Not in universe","Yes","No")

industry_levels <- c("Not in universe", "Agriculture Service", "Other Agriculture",  "Mining","
Construction","Lumber and wood products, except furniture",…)

occupation_levels <- c("Not in universe",  "Public Administration", "Other Executive,
 Administrators, and Managers", "Management Related Occupations", "Engineers",…)

data_train$detailed.industry.recode. <- as.factor(data_train$detailed.industry.recode.)
levels(data_train$detailed.industry.recode.) <- industry_levels

data_train$detailed.occupation.recode <- as.factor(data_train$detailed.occupation.recode)
levels(data_train$detailed.occupation.recode) <- occupation_levels

data_train$business.or.self.employed <- as.factor(data_train$business.or.self.employed)
levels(data_train$ business.or.self.employed) <- not.in.univ_yes_no

data_train$veterans.benefits <-  as.factor(data_train$veterans.benefits)
levels(data_train$veterans.benefits) <- not.in.univ_yes_no

data_train$year <- as.factor(data_train$year)
```

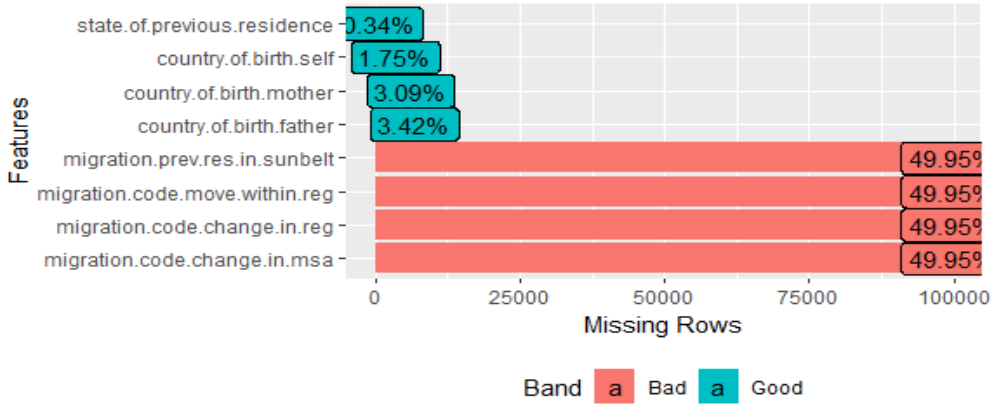
On code enfin les variables quantitatives en type “numeric” (plutôt que “integer”)

```
library(dplyr)
int_cols <- which(sapply(data_train, class)=="integer")
for(i in int_cols){
    data_train[,i] <- as.numeric(data_train[,i])
}
```

4. Aperçu général

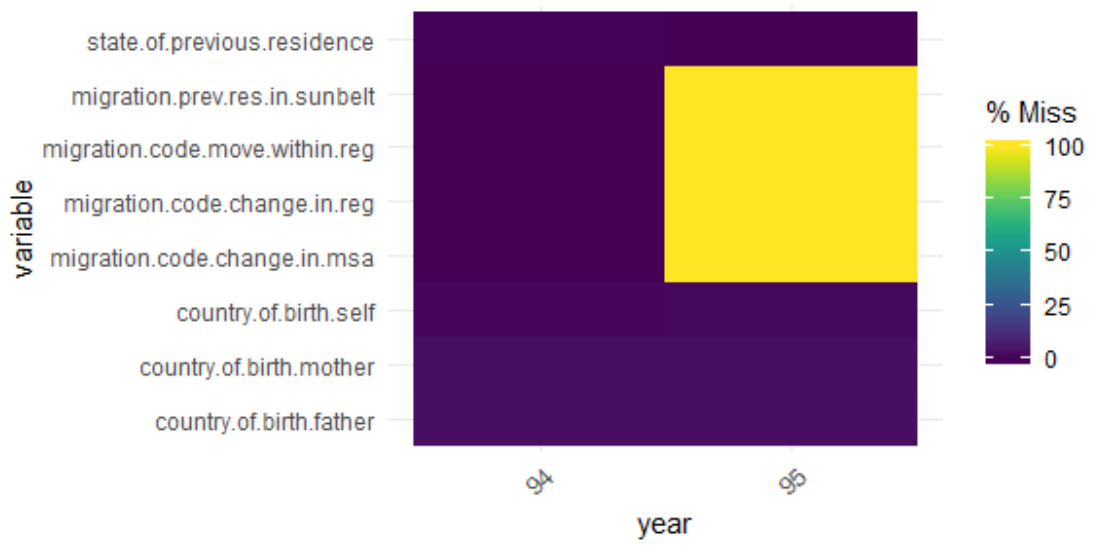
```
#nombre d'individus et variables
dim(data_train)

## [1] 199526     41

#nature des variables
table(sapply(data_train,class))

##
##  factor numeric
##      34       7

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

##  [1] "class.of.worker"
##  [2] "detailed.industry.recode."
##  [3] "detailed.occupation.recode"
##  [4] "education"
##  [5] "enroll.in.edu.inst.last.wk"
##  [6] "marital.stat"
##  [7] "major.industry.code"
##  [8] "major.occupation.code"
##  [9] "race"
## [10] "hispanic.origin"
## [11] "sex"
## [12] "member.of.a.labor.union"
## [13] "reason.for.unemployment"
## [14] "full.or.part.time.employment.stat"
## [15] "tax.filer.stat"
## [16] "region.of.previous.residence"
## [17] "state.of.previous.residence"
## [18] "detailed.household.and.family.stat"
## [19] "detailed.household.summary.in.household"
## [20] "migration.code.change.in.msa"
## [21] "migration.code.change.in.reg"
## [22] "migration.code.move.within.reg"
## [23] "live.in.this.house.1.year.ago"
## [24] "migration.prev.res.in.sunbelt"
## [25] "family.members.under.18"
## [26] "country.of.birth.father"
## [27] "country.of.birth.mother"
## [28] "country.of.birth.self"
## [29] "citizenship"
## [30] "business.or.self.employed"
## [31] "fill.inc.questionnaire.for.veteran.s.admi"
## [32] "veterans.benefits"
## [33] "year"
## [34] "outcome"

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

## [1] "age"                            "wage.per.hour"
## [3] "capital.gains"                  "capital.losses"
```

```
## [5] "from.stocks"                       "num.persons.worked.for.employer"
## [7] "weeks.worked.in.year"

#données manquantes
library(DataExplorer)
cols_with_missing_data <- colnames(data_train)[ apply(data_train, 2, anyNA) ]
data_train  %>% select(cols_with_missing_data) %>% plot_missing()
```



8 des 41 variables présentent des données manquantes et on distingue clairement deux groupes parmi ces 8 variables. Un pour lequel le taux de données manquantes est faible (inférieur à 3.5%) : état de résidence précédent, pays de naissance de la personne recensée, de sa mère et de son père. Un autre groupe de variables, relatif à des données de migration présente un taux de valeurs manquantes voisin de 50%. Ceci s'explique par le fait que ces données sont disponibles pour l'année 1994 mais pas pour l'année 1995 :

```
library(naniar)
data_missing_by_year <- data_train  %>% select(cbind(cols_with_missing_data),"year")
gg_miss_fct(x = data_missing_by_year,fct=year)
```



4. Exploration

4.1 Analyse univariée

4.1.1 Variables quantitatives

4.1.1.1 Indicateurs statistiques

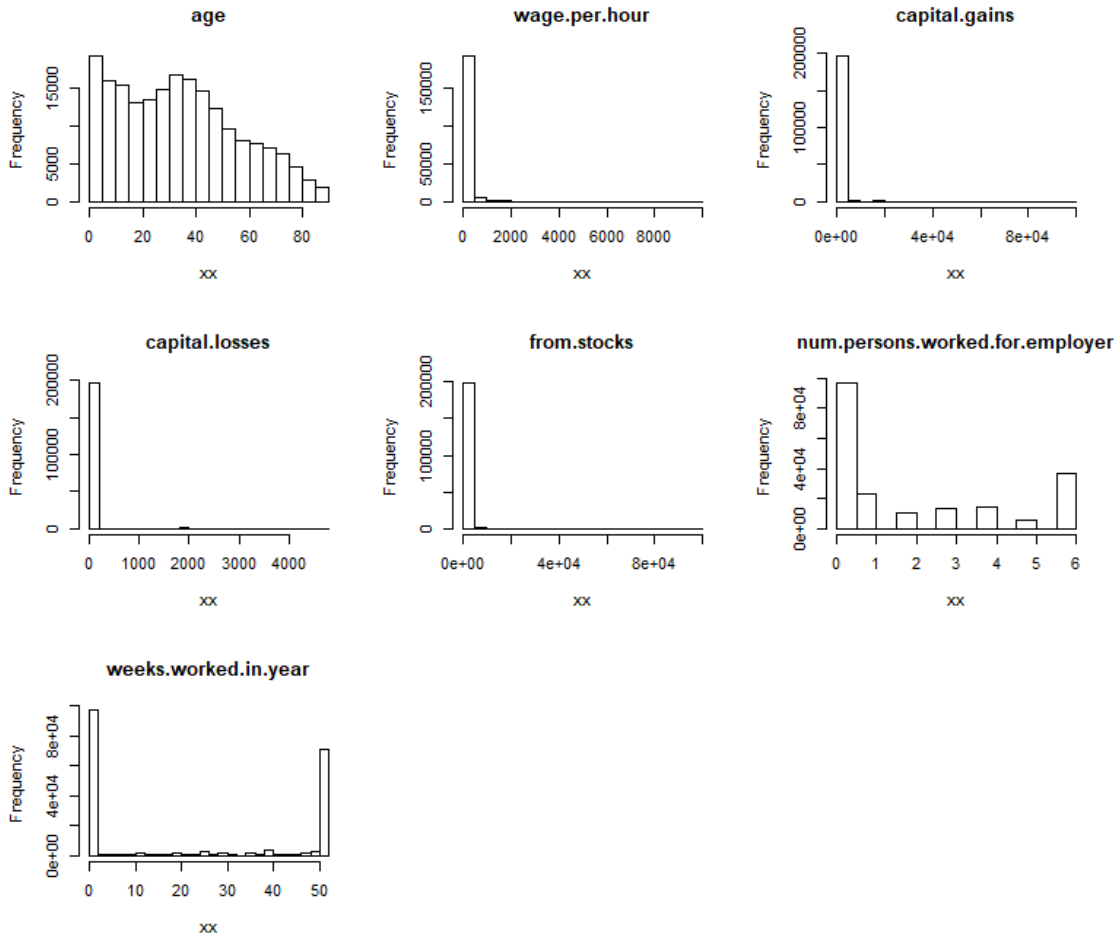
```
# chargement de la bibliothèque
library(stargazer)

# affichage de quelques indicateurs statistiques pour variables quantitatives
stargazer(data_train,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.
## -----------------------------------------------------------------------------------------------
## age                              199,526  0    15      33   34.502     50      90    22.351
## wage.per.hour                    199,526  0    0       0    54.948     0     9,999   274.167
## capital.gains                    199,526  0    0       0    430.726    0     99,999 4,666.440
## capital.losses                   199,526  0    0       0    36.389     0     4,608   268.271
## from.stocks                      199,526  0    0       0    198.124    0     99,999 1,965.004
## num.persons.worked.for.employer 199,526  0    0       1     1.951     4       6      2.364
## weeks.worked.in.year             199,526  0    0       8    23.106     52      52    24.399
## -----------------------------------------------------------------------------------------------
-
```

Aucune des variables quantitatives n'est constante mais on note que plusieurs variables (*capital.gains*, *capital.losse*s, *from.stocks* et *wage.per.hour*) ont une valeur médiane égale à leur valeur minimale (zéro) ce qui indique une forte proportion de valeurs nulles.

4.1.1.2 Représentations graphiques

On commence par visualiser les distributions marginales des variables. Le nombre de valeurs distinctes par variable étant élevé, on utilise des histogrammes plutôt que des diagrammes en barres.

```
par(mfrow=c(3,3))
varhist<-c( "age", "wage.per.hour","capital.gains", "capital.losses","from.stocks",
            "num.persons.worked.for.employer","weeks.worked.in.year")
mapply(data_train[,varhist],
       FUN=function(xx,name){hist(xx,main=name)},
              name=varhist)
```



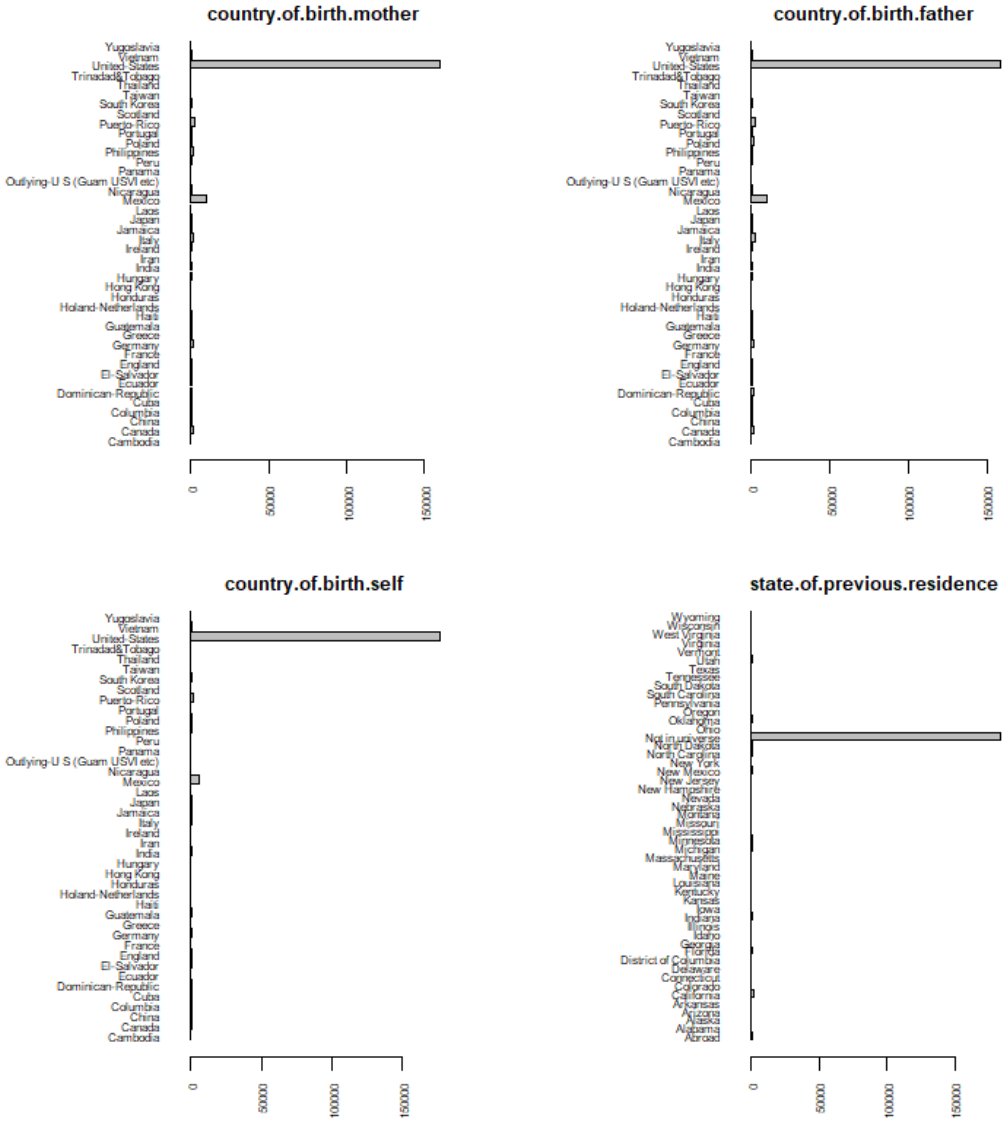
On note les points suivants ;

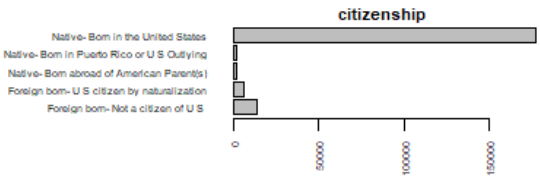
- La variable décrivant l'âge de la population ne présente pas difficulté d'interprétation ou de traitement. Au besoin on peut transformer cette variable quantitative en variable qualitative en créant des tranches d'âges
- La distribution de la variable *weeks.worked.in.year* présente clairement deux catégories majoritaires et une catégorie ultra minoritaire : les personnes ne travaillant pas, les personnes travaillant à temps plein et celles ayant travaillé à temps partiel ou de façon intermittente
- Les variables *capital.gains*, *capital.losses*, *from.stocks* et *wage.per.hour* ont une écrasante majorité (plus de 90% pour *wage.per.hour*, plus de 95% pour les 3 autres) de valeurs nulles. Ceci peut s'expliquer par le fait que ces variables ne concernent pas l'ensemble de la population (par exemple les enfants ne touchent pas de salaires et n'ont pas d'investissements en bourse, les valeurs associées à ces variables sont donc nulles). On peut donc considérer que chacune de ces variables porte 2 informations distinctes : une qualitative (la variable est pertinente pour l'individu si la valeur associée est strictement positive), une qualitative (la valeur elle-même)

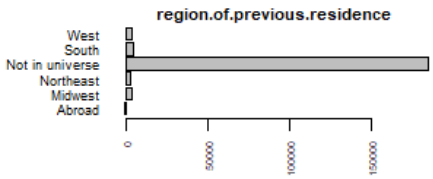
4.1.2 Variables qualitatives

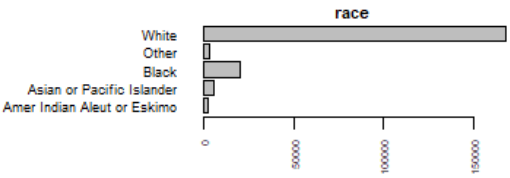
On repère les modalités rares et sur-représentées via des diagrammes en barres (seul le code permettant d'afficher les deux premiers graphiques est inclus ci-dessous, les autres graphiques peuvent être générés par des instructions similaires ne présentant pas d'intérêt particulier)

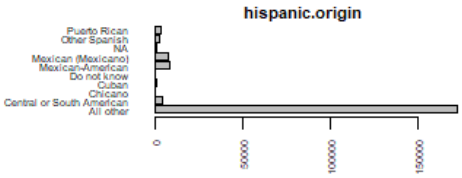
```
par(mar=c(3, 10, 1, 1))
par(mfrow=c(1,2))
par(cex.main=0.8)
barplot(table(data_train$country.of.birth.mother),horiz=TRUE,las=2,
        cex.names=0.5,cex.axis = 0.5,main="country.of.birth.mother")
barplot(table(data_train$country.of.birth.father),horiz=TRUE,las=2,
        cex.names=0.5,cex.axis = 0.5,main="country.of.birth.father")
```

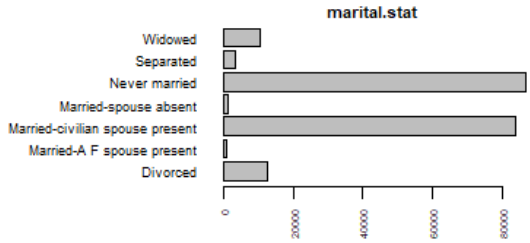


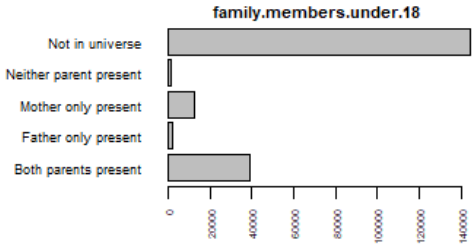

citizenship
Native- Born in the United States
Native- Born in Puerto Rico or U S Outlying
Native- Born abroad of American Parent(s)
Foreign born- U S citizen by naturalization
Foreign born- Not a citizen of U S
0
50000
100000
150000

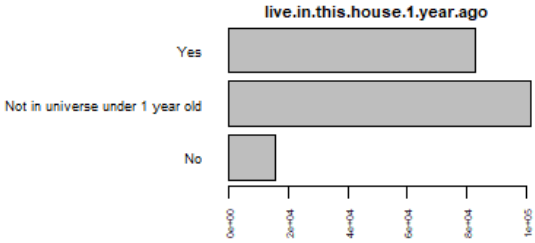

region.of.previous.residence
West
South
Not in universe
Northeast
Midwest
Abroad
0
50000
100000
150000

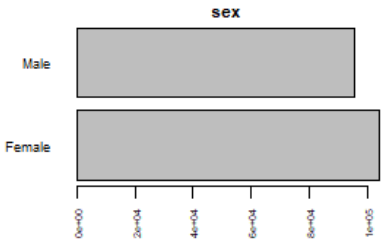

race
White
Other
Black
Asian or Pacific Islander
Amer Indian Aleut or Eskimo
0
50000
100000
150000

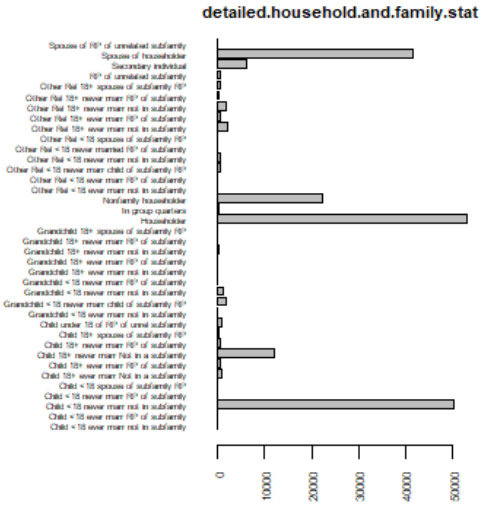

hispanic.origin
Puerto Rican
Other Spanish
NA
Mexican (Mexicano)
Mexican-American
Do not know
Cuban
Chicano
Central or South American
All other
0
50000
100000
150000

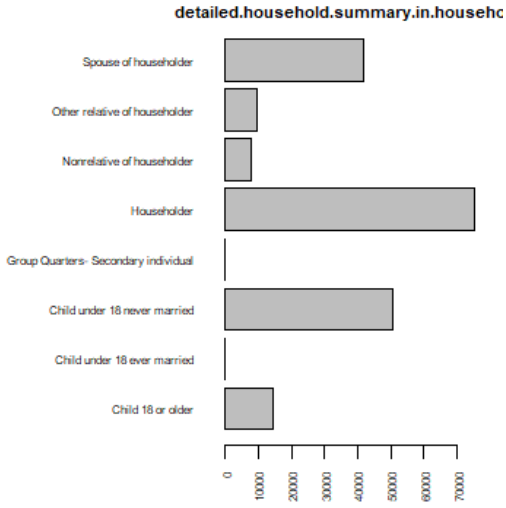

marital.stat
Widowed
Separated
Never married
Married-spouse absent
Married-civilian spouse present
Married-A F spouse present
Divorced
0
20000
40000
60000
80000

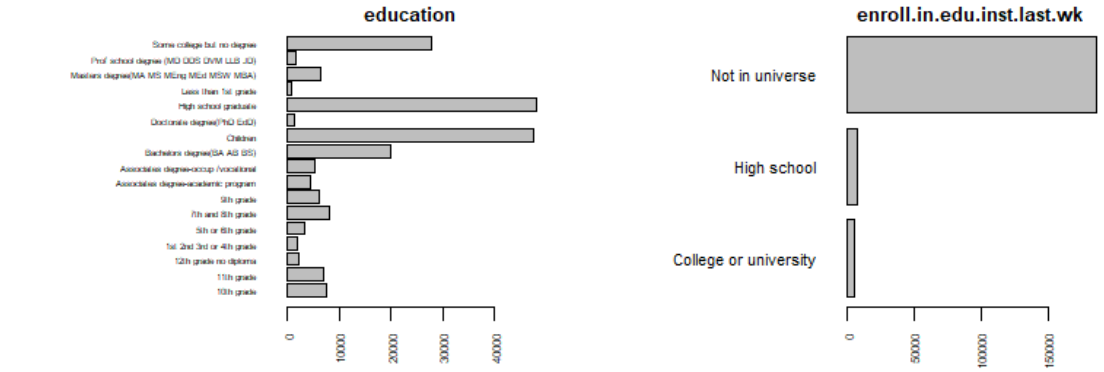

family.members.under.18
Not in universe
Neither parent present
Mother only present
Father only present
Both parents present
0
20000
40000
60000
80000
100000
120000
140000

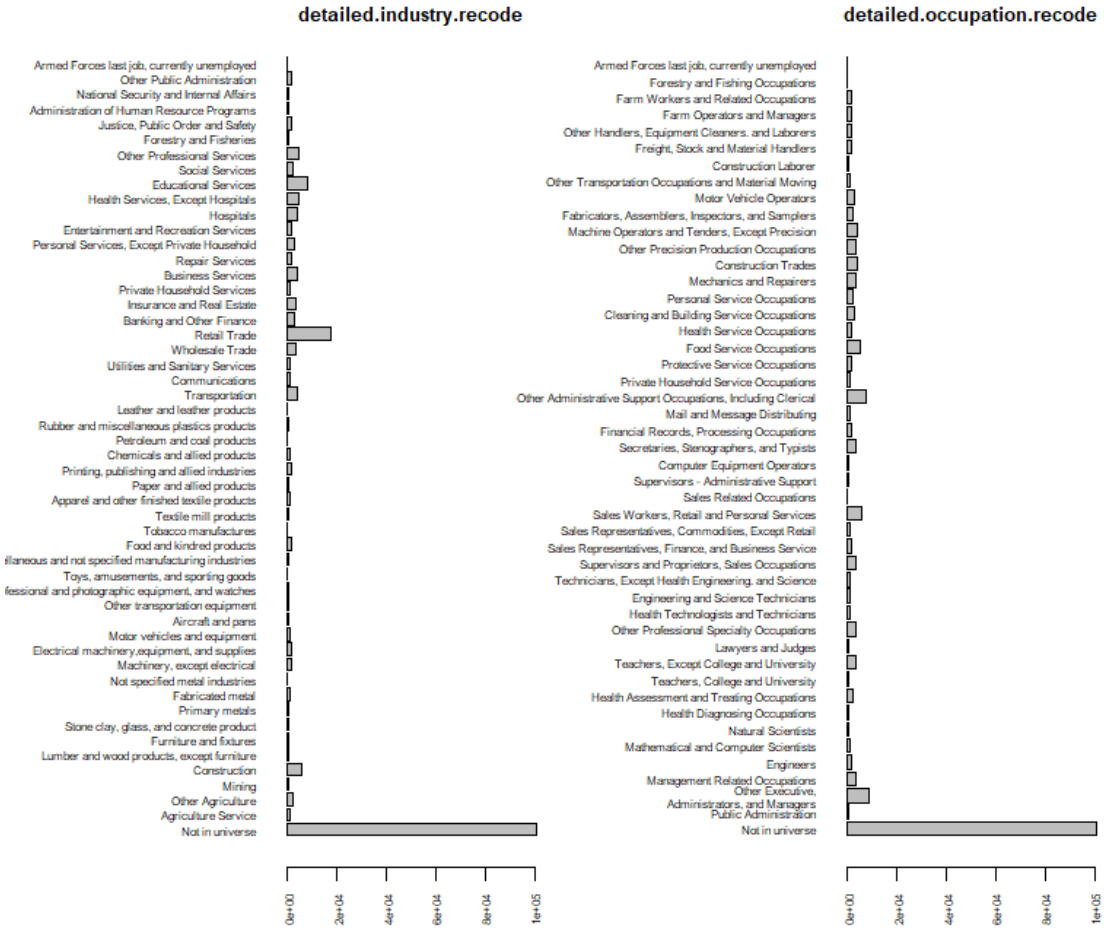

live.in.this.house.1.year.ago
Yes
Not in universe under 1 year old
No
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05

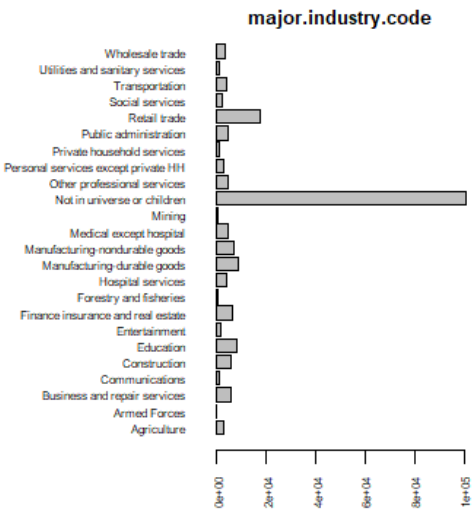

sex
Male
Female
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05

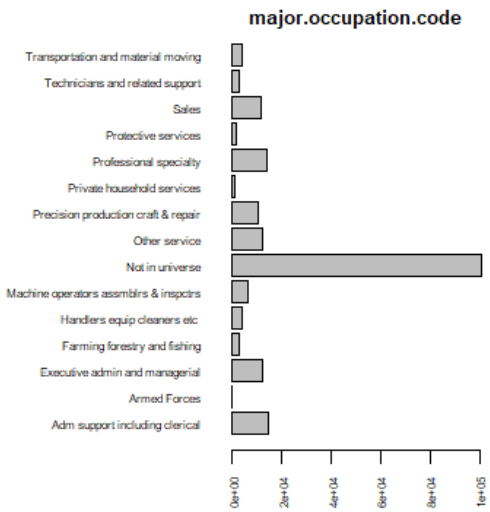

detailed.household.and.family.stat
Spouse of RP* of unrelated subfamily
Spouse of householder
Secondary individual
RP* of unrelated subfamily
Other Rel 18+ spouse of subfamily RP*
Other Rel 18+ never marr RP* of subfamily
Other Rel 18+ never marr not in subfamily
Other Rel 18+ ever marr RP* of subfamily
Other Rel 18+ ever marr not in subfamily
Other Rel < 18 spouse of subfamily RP*
Other Rel < 18 never married RP* of subfamily
Other Rel < 18 never marr not in subfamily
Other Rel < 18 never marr child of subfamily RP*
Other Rel < 18 ever marr RP* of subfamily
Other Rel < 18 ever marr not in subfamily
Nonfamily householder
In group quarters
Householder
Grandchild 18+ spouse of subfamily RP*
Grandchild 18+ never marr RP* of subfamily
Grandchild 18+ never marr not in subfamily
Grandchild 18+ ever marr RP* of subfamily
Grandchild 18+ ever marr not in subfamily
Grandchild < 18 never marr RP* of subfamily
Grandchild < 18 never marr not in subfamily
Grandchild < 18 never marr child of subfamily RP*
Grandchild < 18 ever marr not in subfamily
Child under 18 of RP* of unrel subfamily
Child 18+ spouse of subfamily RP*
Child 18+ never marr RP* of subfamily
Child 18+ never marr Not in a subfamily
Child 18+ ever marr RP* of subfamily
Child 18+ ever marr Not in a subfamily
Child < 18 spouse of subfamily RP*
Child < 18 never marr RP* of subfamily
Child < 18 never marr not in subfamily
Child < 18 ever marr RP* of subfamily
Child < 18 ever marr not in subfamily
0
10000
20000
30000
40000
50000


detailed.household.summary.in.household
Spouse of householder
Other relative of householder
Nonrelative of householder
Householder
Group Quarters- Secondary individual
Child under 18 never married
Child under 18 ever married
Child 18 or older
0
10000
20000
30000
40000
50000
60000
70000


education
Some college but no degree
Prof school degree (MD DDS DVM LLB JD)
Masters degree(MA MS MEng MEd MSW MBA)
Less than 1st grade
High school graduate
Doctorate degree(PhD EdD)
Children
Bachelors degree(BA AB BS)
Associates degree-occup /vocational
Associates degree-academic program
9th grade
7th and 8th grade
5th or 6th grade
1st 2nd 3rd or 4th grade
12th grade no diploma
11th grade
10th grade
0
10000
20000
30000
40000
enroll.in.edu.inst.last.wk
Not in universe
High school
College or university
0
50000
100000
150000


detailed.industry.recode
Armed Forces last job, currently unemployed
Other Public Administration
National Security and Internal Affairs
Administration of Human Resource Programs
Justice, Public Order and Safety
Forestry and Fisheries
Other Professional Services
Social Services
Educational Services
Health Services, Except Hospitals
Hospitals
Entertainment and Recreation Services
Personal Services, Except Private Household
Repair Services
Business Services
Private Household Services
Insurance and Real Estate
Banking and Other Finance
Retail Trade
Wholesale Trade
Utilities and Sanitary Services
Communications
Transportation
Leather and leather products
Rubber and miscellaneous plastics products
Petroleum and coal products
Chemicals and allied products
Printing, publishing and allied industries
Paper and allied products
Apparel and other finished textile products
Textile mill products
Tobacco manufactures
Food and kindred products
ffaneous and not specified manufacturing industries
Toys, amusements, and sporting goods
fessional and photographic equipment, and watches
Other transportation equipment
Aircraft and pars
Motor vehicles and equipment
Electrical machinery,equipment, and supplies
Machinery, except electrical
Not specified metal industries
Fabricated metal
Primary metals
Stone clay, glass, and concrete product
Furniture and fixtures
Lumber and wood products, except furniture
Construction
Mining
Other Agriculture
Agriculture Service
Not in universe
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
detailed.occupation.recode
Armed Forces last job, currently unemployed
Forestry and Fishing Occupations
Farm Workers and Related Occupations
Farm Operators and Managers
Other Handlers, Equipment Cleaners. and Laborers
Freight, Stock and Material Handlers
Construction Laborer
Other Transportation Occupations and Material Moving
Motor Vehicle Operators
Fabricators, Assemblers, Inspectors, and Samplers
Machine Operators and Tenders, Except Precision
Other Precision Production Occupations
Construction Trades
Mechanics and Repairers
Personal Service Occupations
Cleaning and Building Service Occupations
Health Service Occupations
Food Service Occupations
Protective Service Occupations
Private Household Service Occupations
Other Administrative Support Occupations, Including Clerical
Mail and Message Distributing
Financial Records, Processing Occupations
Secretaries, Stenographers, and Typists
Computer Equipment Operators
Supervisors - Administrative Support
Sales Related Occupations
Sales Workers, Retail and Personal Services
Sales Representatives, Commodities, Except Retail
Sales Representatives, Finance, and Business Service
Supervisors and Proprietors, Sales Occupations
Technicians, Except Health Engineering. and Science
Engineering and Science Technicians
Health Technologists and Technicians
Other Professional Specialty Occupations
Lawyers and Judges
Teachers, Except College and University
Teachers, College and University
Health Assessment and Treating Occupations
Health Diagnosing Occupations
Natural Scientists
Mathematical and Computer Scientists
Engineers
Management Related Occupations
Other Executive,
Administrators, and Managers
Public Administration
Not in universe
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05

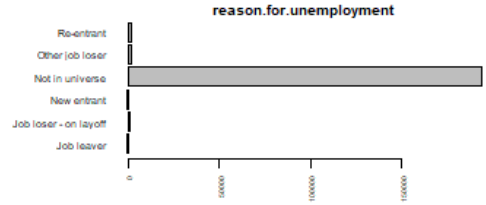

major.industry.code
Wholesale trade
Utilities and sanitary services
Transportation
Social services
Retail trade
Public administration
Private household services
Personal services except private HH
Other professional services
Not in universe or children
Mining
Medical except hospital
Manufacturing-nondurable goods
Manufacturing-durable goods
Hospital services
Forestry and fisheries
Finance insurance and real estate
Entertainment
Education
Construction
Communications
Business and repair services
Armed Forces
Agriculture
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05

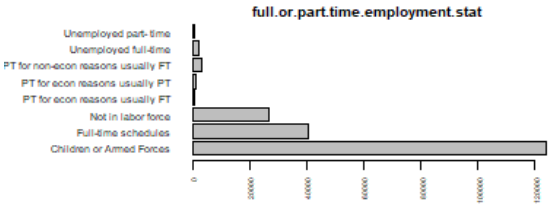

major.occupation.code
Transportation and material moving
Technicians and related support
Sales
Protective services
Professional specialty
Private household services
Precision production craft & repair
Other service
Not in universe
Machine operators assmblrs & inspctrs
Handlers equip cleaners etc
Farming forestry and fishing
Executive admin and managerial
Armed Forces
Adm support including clerical
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05

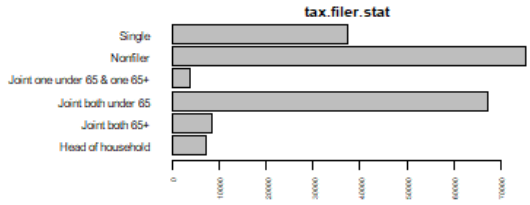

class.of.worker
Without pay
State government
Self-employed-not incorporated
Self-employed-incorporated
Private
Not in universe
Never worked
Local government
Federal government
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05

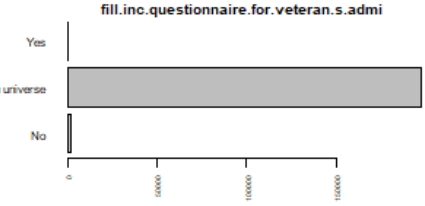

member.of.a.labor.union
Yes
Not in universe
No
0
50000
100000
150000

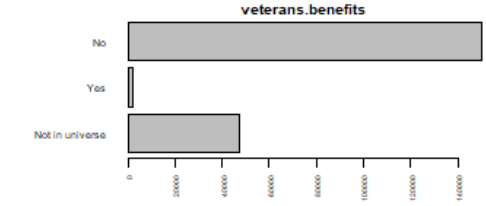

business.or.self.employed
No
Yes
Not in universe
0
50000
100000
150000

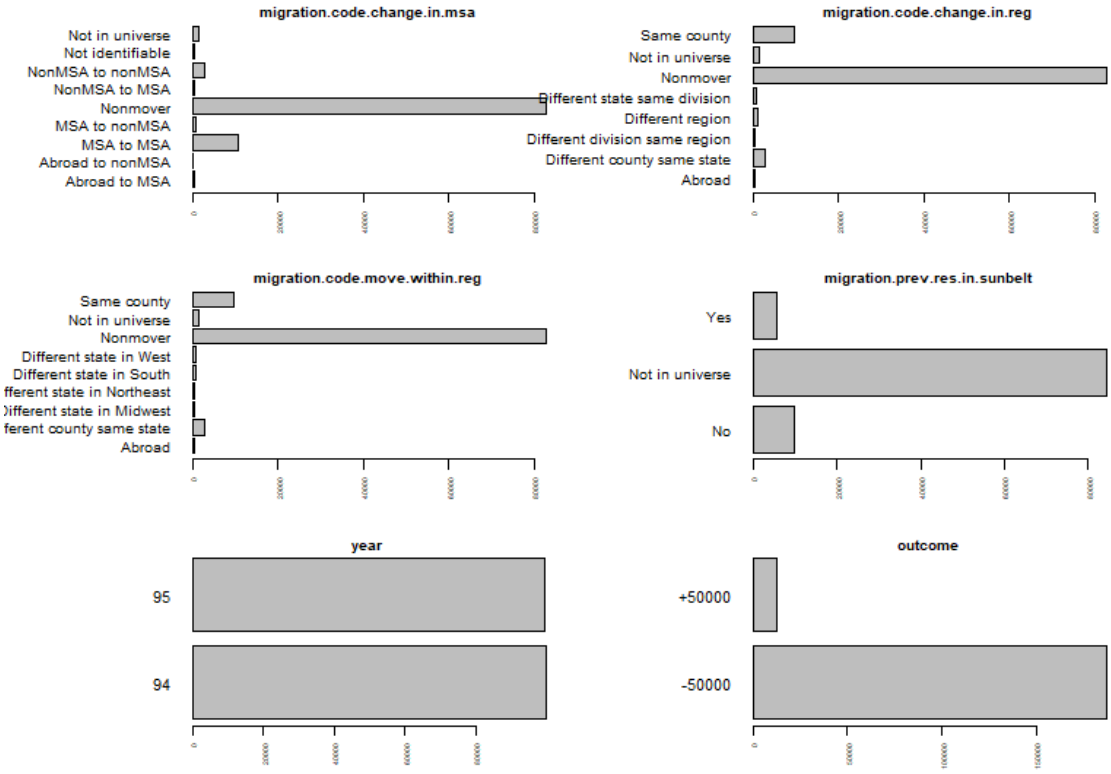

reason.for.unemployment
Re-entrant
Other job loser
Not in universe
New entrant
Job loser - on layoff
Job leaver
0
50000
100000
150000


full.or.part.time.employment.stat
Unemployed part-time
Unemployed full-time
PT for non-econ reasons usually FT
PT for econ reasons usually PT
PT for econ reasons usually FT
Not in labor force
Full-time schedules
Children or Armed Forces
0
20000
40000
60000
80000
100000
120000


tax.filer.stat
Single
Nonfiler
Joint one under 65 & one 65+
Joint both under 65
Joint both 65+
Head of household
0
10000
20000
30000
40000
50000
60000
70000


fill.inc.questionnaire.for.veteran.s.admi
Yes
Not in universe
No
0
50000
100000
150000


veterans.benefits
No
Yes
Not in universe
0
20000
40000
60000
80000
100000
120000
140000



Sur la trentaine de variables qualitatives, plus de la moitié présente une modalité qui concentre la grande majorité des observations (ex: modalité *not in universe* de la variable *state.of.previous.residence*. On remarque également la présence de variables qualitatives possédant plusieurs dizaines de modalités (ex : *country.of.birth.father*, *country.of.birth.mother*, *country.of.birth.self* , detailed.industry.recode, *detailed.occupation.recodes*) avec une très forte proportion de modalités rares.

4.2 Analyse bivariée

Cette étude étant effectuée dans un contexte de classification supervisée, nous effectuerons dans un premier temps une analyse bivariée entre les variables explicatives et la variable réponse. Dans un deuxième temps nous effectuerons une analyse bivariée des couples de variables explicatives.

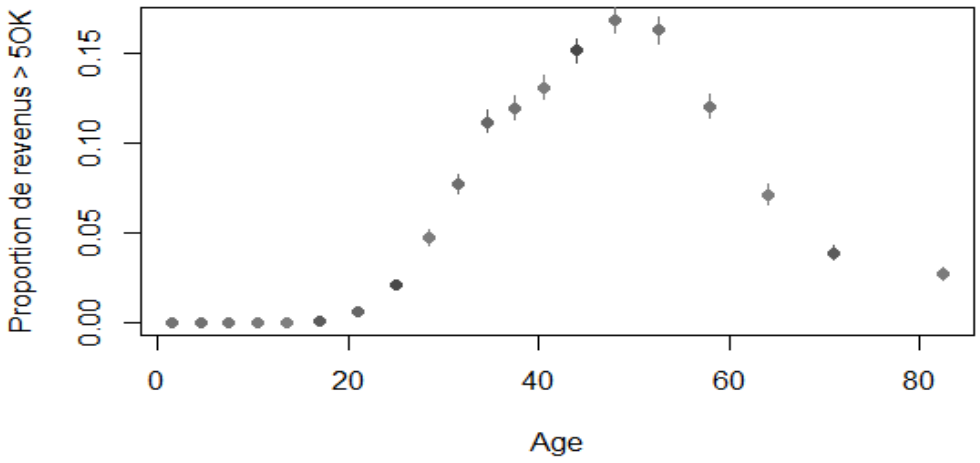
4.2.1 Lien entre variables explicatives et variable réponse

4.2.1.1 Linéarité

Commençons par la variable *age* (en effectuant au préalable une discrétisation en 20 classes) :

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

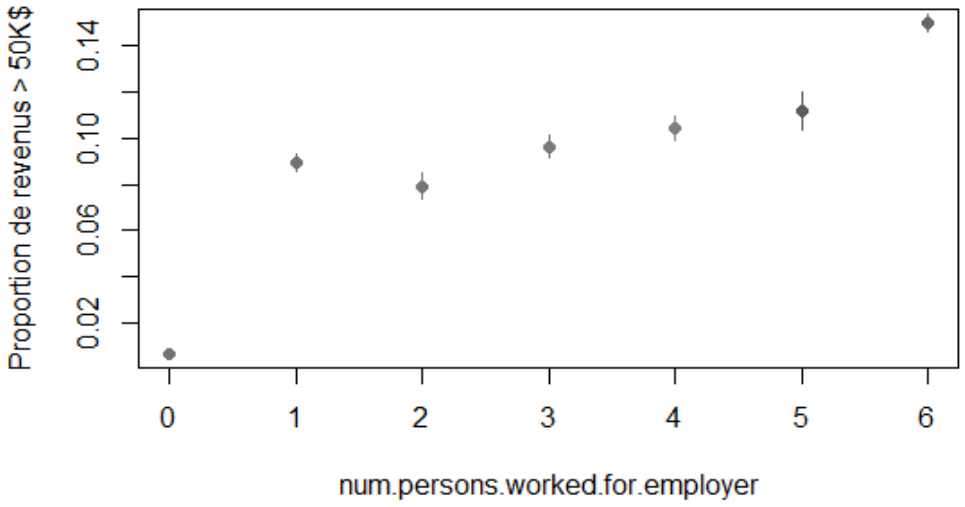
```
cont.table<-table(age.new,data_train$outcome)
prof.lignes<-prop.table(cont.table,1)

res.binom.test<-mapply(cont.table[,2],
                       FUN=binom.test,
                       n=rowSums(cont.table),
                       SIMPLIFY = FALSE)
ci<-sapply(res.binom.test,"[[","conf.int")
abscisses <- hist(data_train$age,quantile(data_train$age,probs =ordrequantiles),plot = FALSE)$m
ids
col<-gray.colors(12000,.95,0)[rowSums(cont.table)]

plot(abscisses,
     prof.lignes[,2],
     pch=16,
     col=col,
     xlab="Age",
     ylab="Proportion de revenus > 50K")

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. Le lien entre âge et niveau de revenu est clairement non linéaire et non monotone. Les revenus correspondent à une évolution professionnelle type - de 0 à 18 ans : phase d'éducation (aucun revenu) de 18 à 35 ans : début de carrière, augmentation rapide des revenus, 35 à 50 ans : l'augmentation tend à ralentir et atteint un maximum vers 50 ans puis diminution des revenus sur la fin de la carrière (50 à 65 ans) avec un plateau à la retraite (plus de 65 ans).

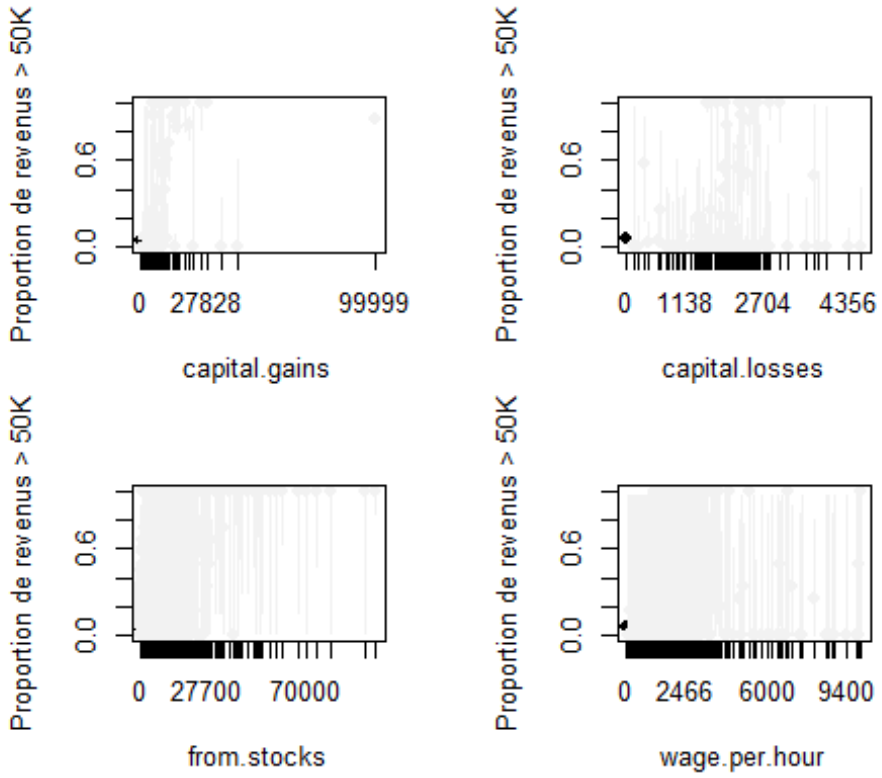
Etudions ensuite la variable *num.persons.worked.for.employer* :

```
# calcul de la proportion de revenus > 50K$ pour chaque valeur de
# "num.persons.worked.for.employer"
cont.table<-table(data_train$num.persons.worked.for.employer,data_train$outcome)
prof.lignes<-prop.table(cont.table,1)

#calcul des bornes de l'intervalle de confiance associée
res.binom.test<-mapply(cont.table[,2], FUN=binom.test, n=rowSums(cont.table),
                       SIMPLIFY = FALSE)
ci<-sapply(res.binom.test,"[[","conf.int")

# représentation des proportions en fonction de "num.persons.worked.for.employer"
# (coloration en fonction du nombre d'observations pour la variable)
abscisses<-as.numeric(rownames(prof.lignes))

# affichage des proportions
plot(abscisses,
     prof.lignes[,2],
     pch=16,
     col=col,
     xlab="num.persons.worked.for.employer",
     ylab="Proportion de revenus > 50K$")

# 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])
}
```



Il existe un lien plutôt monotone entre le niveau de revenu et la variable *num.persons.worked.for.employer* (taille de l'entreprise) . Quand la valeur de cette dernière augmente (autrement dit plus l'entreprise est grande) plus les salaires sont élevés.

```r
par(mfrow=c(2,2),mar=c(4,5, 3, 2) + 0.1)
varbarplot <- c("capital.gains", "capital.losses", "from.stocks", "wage.per.hour")

for(ii in  varbarplot){
  xx<-data_train[,ii]
  cont.table<-table(cbind.data.frame(xx,data_train$outcome))
  prof.lignes<-prop.table(cont.table,1)
  res.binom.test<-mapply(cont.table[,2],
                         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[,2],
       pch=16,
       col=col,
       ylab="Proportion de revenus > 50K",
       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])
  }
}
```

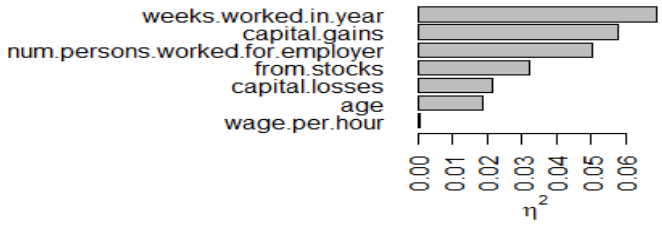


Le lien entre la variable *outcome* et les variables *capital.gains*, *capital.losses* , *from.stocks* *wage.per.hour* n'est ni linéaire, ni monotone mais la proportion de hauts revenus semble plus importante quand ces variables ont des valeurs différentes de zéro.

4.2.1.2 Identification des variables les plus discriminantes

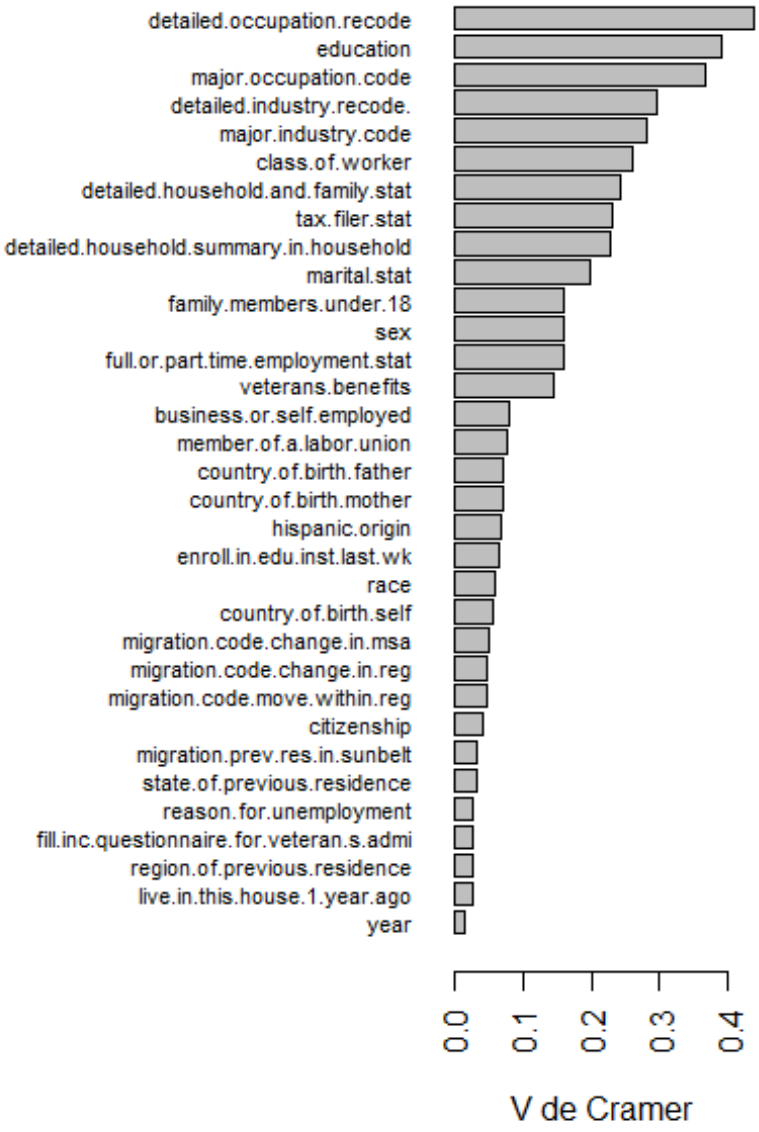
On regarde ensuite quelles sont les variables les plus liées au niveau de revenu.

```
#Chargement du package BioStatR
library(BioStatR)

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

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

#représentation
par(mar=c(5, 15, 4, 2) + 0.1)#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 *weeks.worked.in.year*, *capital.gains* et *num.persons.worked.for.employer*. De manière assez surprenante la variable *wage.per.hour* ne semble pas discriminante pour déterminer le niveau de revenu.

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

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

#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",cex.names=0.7)
```



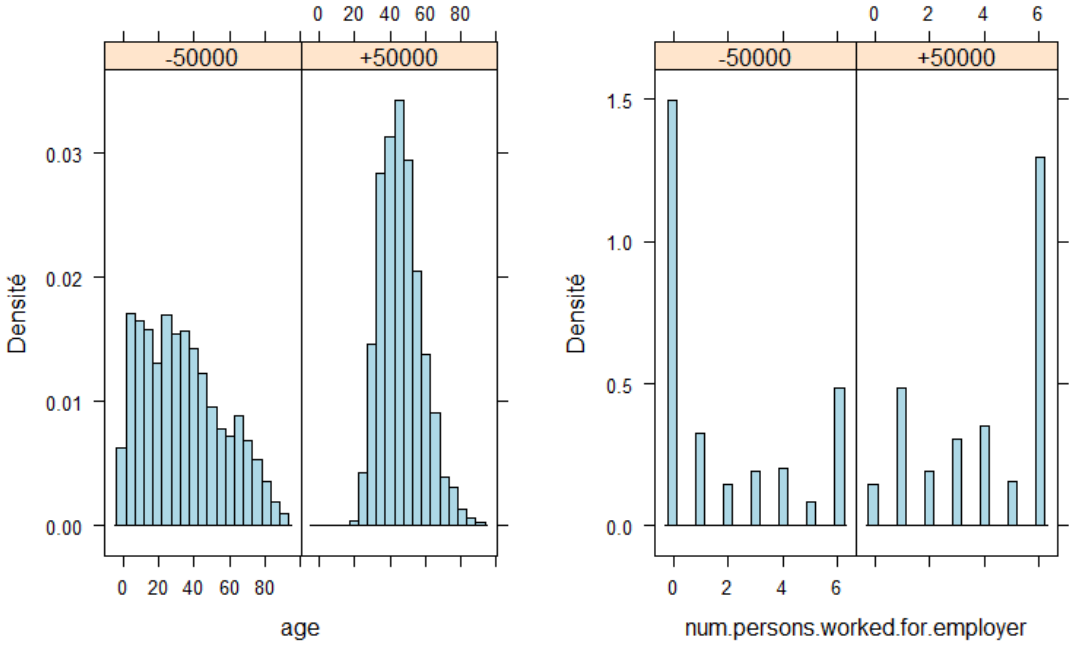
Parmi les variables qualitatives, les variables les plus liées sont *detailed.occupation.recode* et *education*, tandis que les variables *live.in.this.house.1.year.ago* et *year* ne semblent pas discriminantes.

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 écarter en priorité. Cette conclusion doit cependant être nuancée 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.

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.

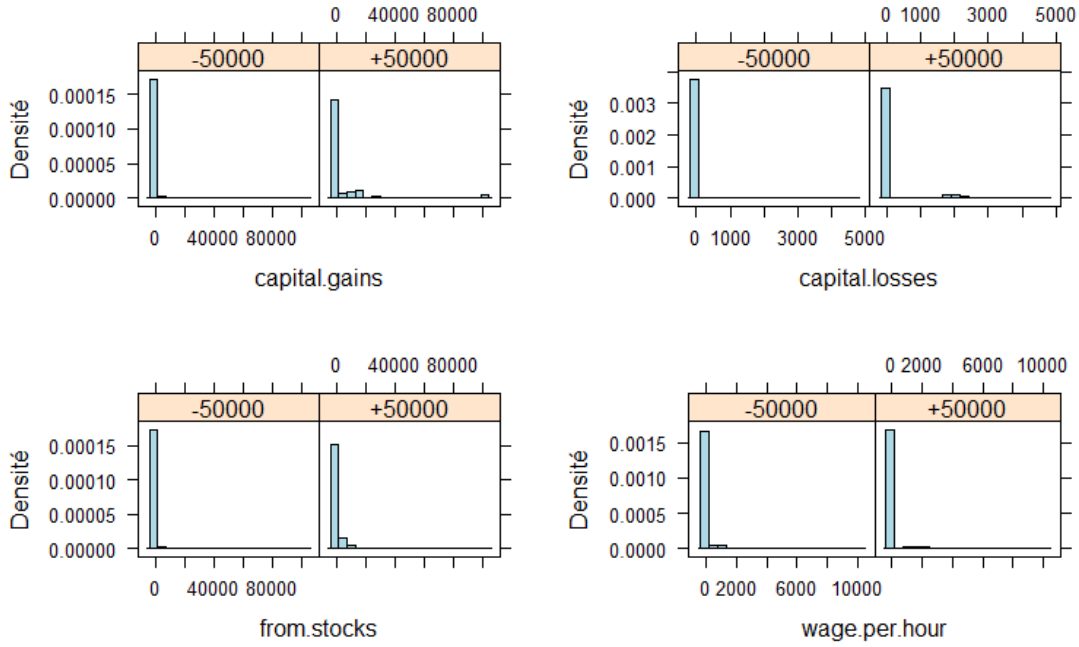
```
# Chargement de la librarie lattice permettant de faire des graphiques relativement avancés
# et de la librairie gridExra, utilisée ici pour positionner les graphiques les uns à côté des
autres

library(lattice);library(gridExtra)

# distribution conditionnelle de "age"
plot1<-lattice::histogram(~age|outcome,data=data_train,type="density",col="lightblue",
                          ylab="Densité")
# distribution conditionnelle de "num.persons.worked.for.employer"
plot2<-lattice::histogram(~num.persons.worked.for.employer|outcome,data=data_train,
                          type="density",col="lightblue",ylab="Densité")

# affichage
grid.arrange(plot1,plot2,nrow=1,ncol=2)
```



Les distributions conditionnelles des variables *age* et *num.persons.worked.for.employer* ne sont pas normales ce qui peut nécessiter une transformation en fonction de la méthode de classification que nous emploierons.

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

library(lattice);library(gridExtra)

# distribution conditionnelle de "capital.gains"
plot1<-lattice::histogram(~capital.gains|outcome,data=data_train,type="density",col="lightblue"
```

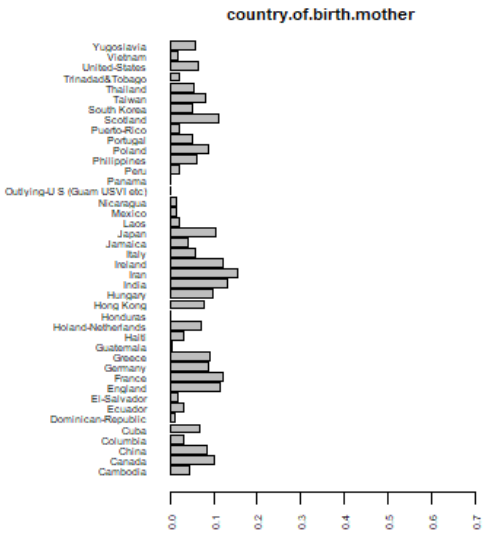
```
,
                          ylab="Densité")
# distribution conditionnelle de "capital.losses"
plot2<-lattice::histogram(~capital.losses|outcome,data=data_train,
                          type="density",col="lightblue",ylab="Densité")
# distribution conditionnelle de "from.stocks"
plot3<-lattice::histogram(~from.stocks|outcome,data=data_train,
                          type="density",col="lightblue",ylab="Densité")
# distribution conditionnelle de "wage.per.hour"
plot4<-lattice::histogram(~wage.per.hour|outcome,data=data_train,
                          type="density",col="lightblue",ylab="Densité")

# affichage
grid.arrange(plot1,plot2,plot3,plot4, nrow=2,ncol=2)
```

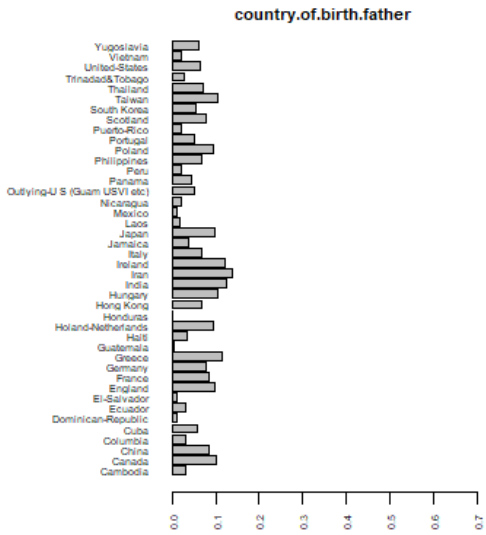


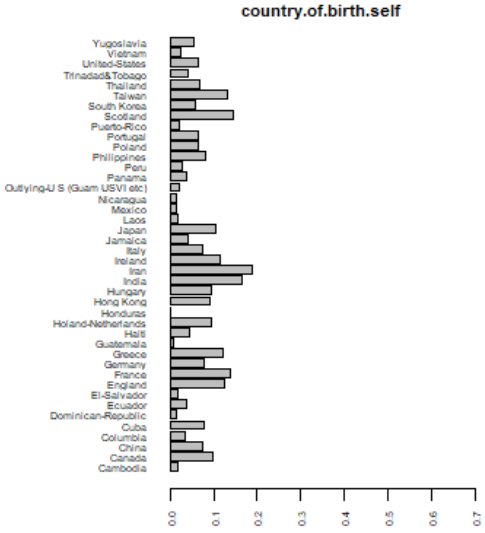
Les distributions conditionnelles des variables *capital.gains*, *capital.losses*, *from.stocks* et *wage.per.hour* ne sont clairement pas normales et sont marquées par un pic en zéro.

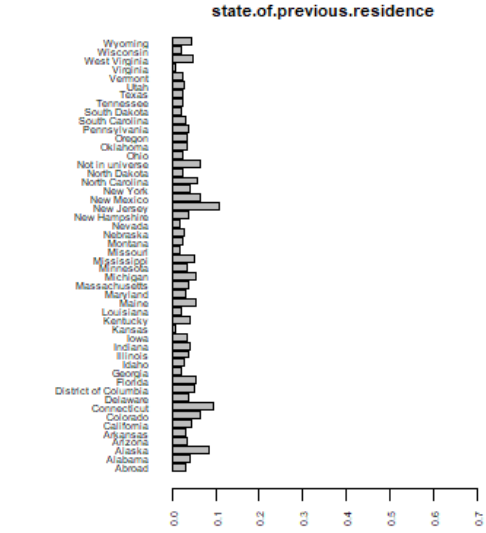
Pour les variables qualitatives, il sera intéressant d'identifier la distribution conditionnelle de la variable réponse en fonction des modalités des variables. Ceci peut permettre d'effectuer des regroupements de modalités préservant au mieux les liaisons entre ces variables et la variable réponse. On choisit donc de représenter la proportion de revenu élevé (>50K$) en fonction des modalités des différentes variables explicatives qualitatives.

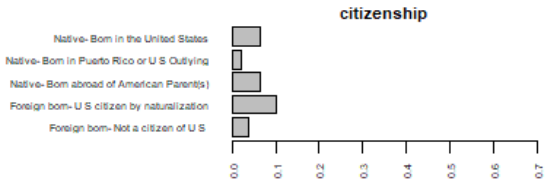
```
var.expl.quali<-names(var.factor[-length(var.factor)])
mapply(data_train[,var.expl.quali],FUN=function(xx,name){
  tmp<-table(xx,data_train$outcome)
  tmp<-tmp/rowSums(tmp)
  barplot(tmp[,"+50000"],main=name,horiz = TRUE,las=2,xlim=c(0,0.7))
},name=var.expl.quali)
```

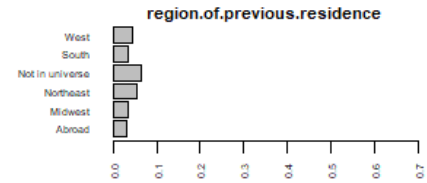

country.of.birth.mother
Yugoslavia
Vietnam
United-States
Trinadad&Tobago
Thailand
Taiwan
South Korea
Scotland
Puerto-Rico
Portugal
Poland
Philippines
Peru
Panama
Outlying-U S (Guam USVI etc)
Nicaragua
Mexico
Laos
Japan
Jamaica
Italy
Ireland
Iran
India
Hungary
Hong Kong
Honduras
Holand-Netherlands
Haiti
Guatemala
Greece
Germany
France
England
El-Salvador
Ecuador
Dominican-Republic
Cuba
Columbia
China
Canada
Cambodia
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

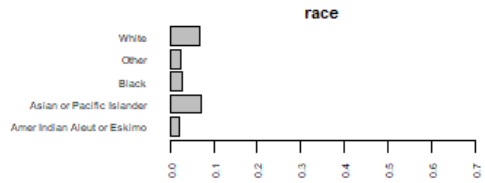

country.of.birth.father
Yugoslavia
Vietnam
United-States
Trinadad&Tobago
Thailand
Taiwan
South Korea
Scotland
Puerto-Rico
Portugal
Poland
Philippines
Peru
Panama
Outlying-U S (Guam USVI etc)
Nicaragua
Mexico
Laos
Japan
Jamaica
Italy
Ireland
Iran
India
Hungary
Hong Kong
Honduras
Holand-Netherlands
Haiti
Guatemala
Greece
Germany
France
England
El-Salvador
Ecuador
Dominican-Republic
Cuba
Columbia
China
Canada
Cambodia
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

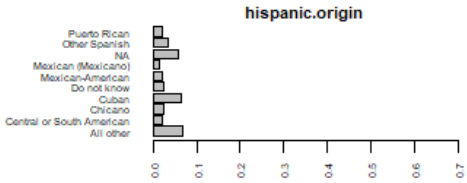

country.of.birth.self
Yugoslavia
Vietnam
United-States
Trinadad&Tobago
Thailand
Taiwan
South Korea
Scotland
Puerto-Rico
Portugal
Poland
Philippines
Peru
Panama
Outlying-U S (Guam USVI etc)
Nicaragua
Mexico
Laos
Japan
Jamaica
Italy
Ireland
Iran
India
Hungary
Hong Kong
Honduras
Holand-Netherlands
Haiti
Guatemala
Greece
Germany
France
England
El-Salvador
Ecuador
Dominican-Republic
Cuba
Columbia
China
Canada
Cambodia
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

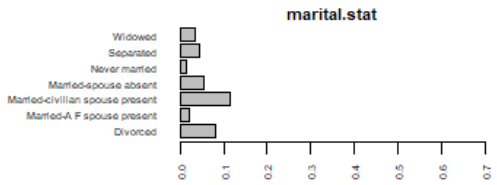

state.of.previous.residence
Wyoming
Wisconsin
West Virginia
Virginia
Vermont
Utah
Texas
Tennessee
South Dakota
South Carolina
Pennsylvania
Oregon
Oklahoma
Ohio
Not in universe
North Dakota
North Carolina
New York
New Mexico
New Jersey
New Hampshire
Nevada
Nebraska
Montana
Missouri
Mississippi
Minnesota
Michigan
Massachusetts
Maryland
Maine
Louisiana
Kentucky
Kansas
Iowa
Indiana
Illinois
Idaho
Georgia
Florida
District of Columbia
Delaware
Connecticut
Colorado
California
Arkansas
Arizona
Alaska
Alabama
Abroad
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

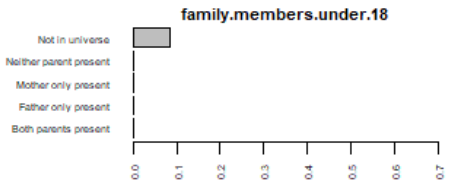

citizenship
Native- Born in the United States
Native- Born in Puerto Rico or U S Outlying
Native- Born abroad of American Parent(s)
Foreign born- U S citizen by naturalization
Foreign born- Not a citizen of U S
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

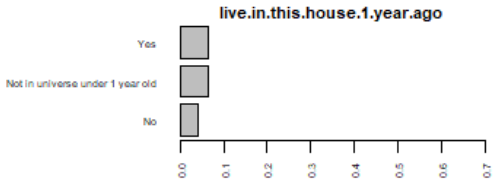

region.of.previous.residence
West
South
Not in universe
Northeast
Midwest
Abroad
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

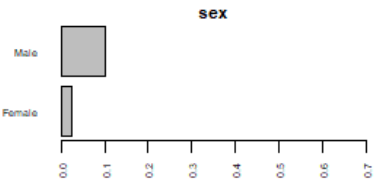

race
White
Other
Black
Asian or Pacific Islander
Amer Indian Aleut or Eskimo
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

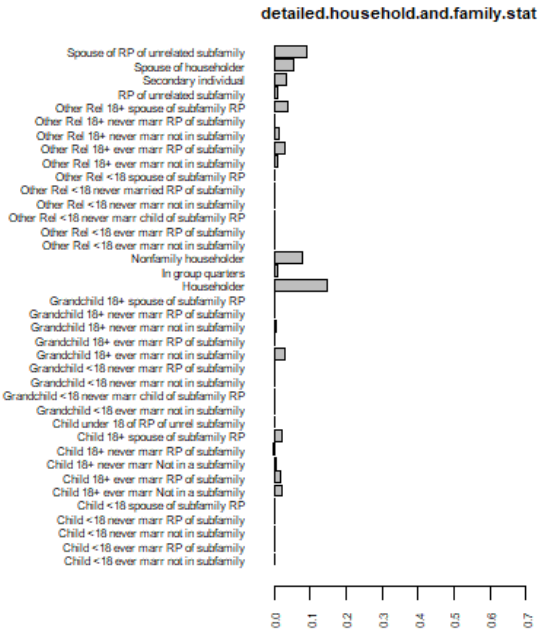

hispanic.origin
Puerto Rican
Other Spanish
NA
Mexican (Mexicano)
Mexican-American
Do not know
Cuban
Chicano
Central or South American
All other
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


marital.stat
Widowed
Separated
Never married
Married-spouse absent
Married-civilian spouse present
Married-A F spouse present
Divorced
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

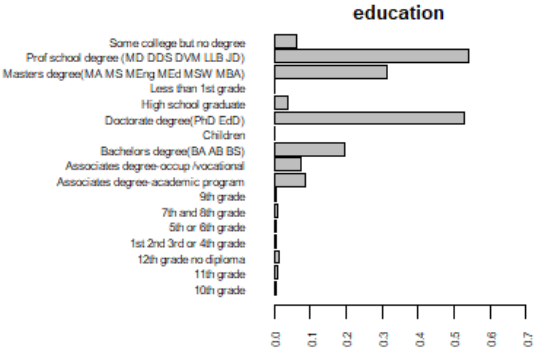

family.members.under.18
Not in universe
Neither parent present
Mother only present
Father only present
Both parents present
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

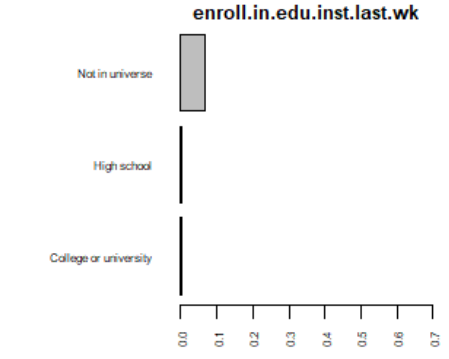

live.in.this.house.1.year.ago
Yes
Not in universe under 1 year old
No
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

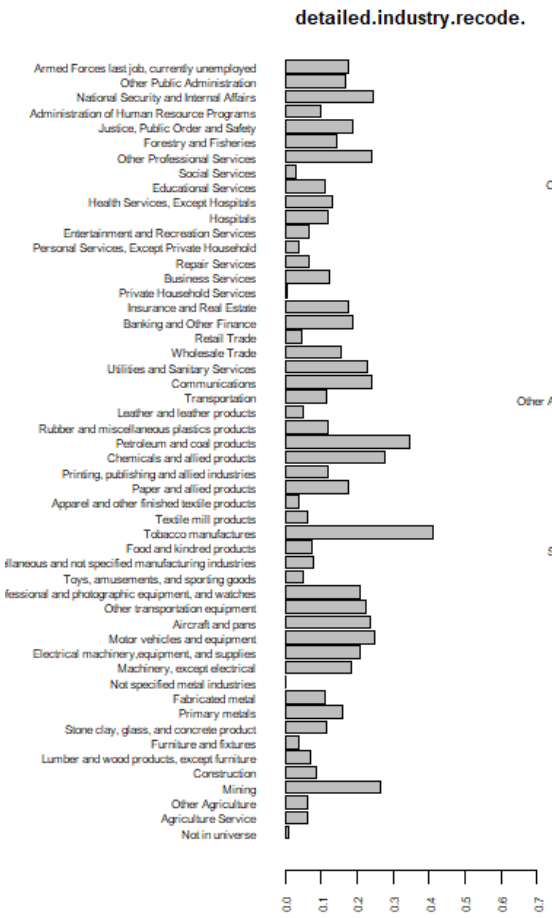

sex
Male
Female
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

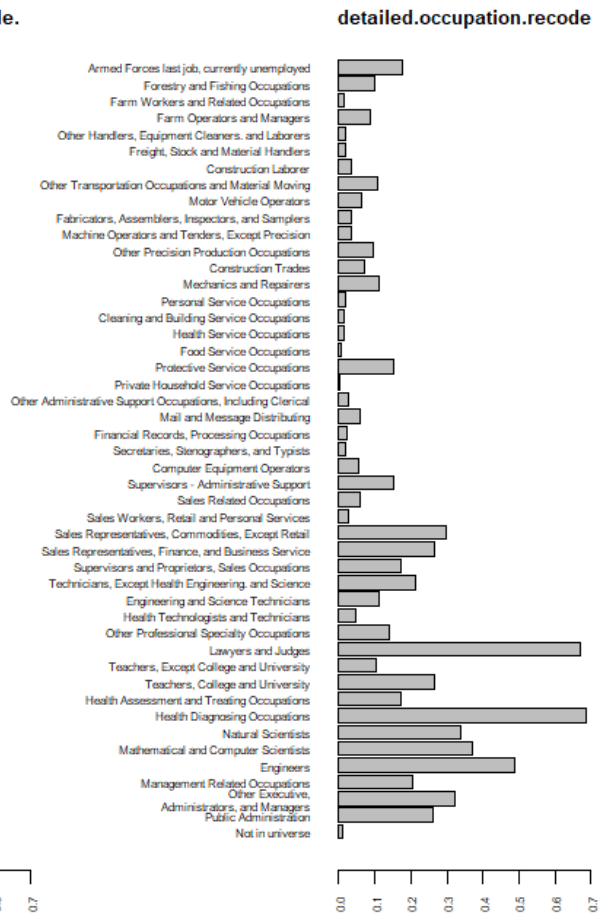

detailed.household.and.family.stat
Spouse of RP of unrelated subfamily
Spouse of householder
Secondary individual
RP of unrelated subfamily
Other Rel 18+ spouse of subfamily RP
Other Rel 18+ never marr RP of subfamily
Other Rel 18+ never marr not in subfamily
Other Rel 18+ ever marr RP of subfamily
Other Rel 18+ ever marr not in subfamily
Other Rel <18 spouse of subfamily RP
Other Rel <18 never married RP of subfamily
Other Rel <18 never marr not in subfamily
Other Rel <18 never marr child of subfamily RP
Other Rel <18 ever marr RP of subfamily
Other Rel <18 ever marr not in subfamily
Nonfamily householder
In group quarters
Householder
Grandchild 18+ spouse of subfamily RP
Grandchild 18+ never marr RP of subfamily
Grandchild 18+ never marr not in subfamily
Grandchild 18+ ever marr RP of subfamily
Grandchild 18+ ever marr not in subfamily
Grandchild <18 never marr RP of subfamily
Grandchild <18 never marr not in subfamily
Grandchild <18 never marr child of subfamily RP
Grandchild <18 ever marr not in subfamily
Child under 18 of RP of unrel subfamily
Child 18+ spouse of subfamily RP
Child 18+ never marr RP of subfamily
Child 18+ never marr Not in a subfamily
Child 18+ ever marr RP of subfamily
Child 18+ ever marr Not in a subfamily
Child <18 spouse of subfamily RP
Child <18 never marr RP of subfamily
Child <18 never marr not in subfamily
Child <18 ever marr RP of subfamily
Child <18 ever marr not in subfamily
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

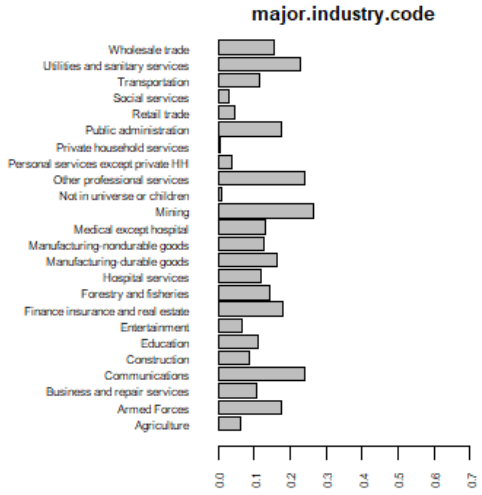

detailed.household.summary.in.househc
Spouse of householder
Other relative of householder
Nonrelative of householder
Householder
Group Quarters- Secondary individual
Child under 18 never married
Child under 18 ever married
Child 18 or older
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

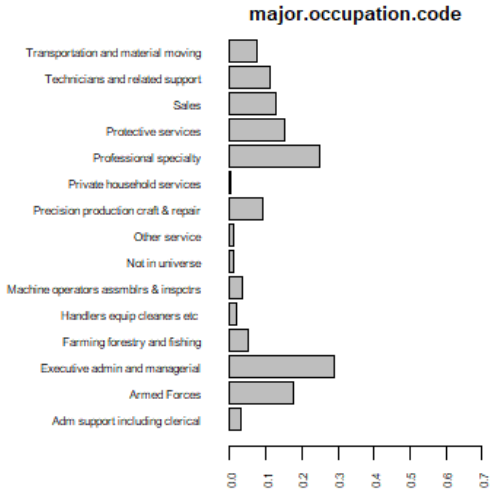

education
Some college but no degree
Prof school degree (MD DDS DVM LLB JD)
Masters degree(MA MS MEng MEd MSW MBA)
Less than 1st grade
High school graduate
Doctorate degree(PhD EdD)
Children
Bachelors degree(BA AB BS)
Associates degree-occup /vocational
Associates degree-academic program
9th grade
7th and 8th grade
5th or 6th grade
1st 2nd 3rd or 4th grade
12th grade no diploma
11th grade
10th grade
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

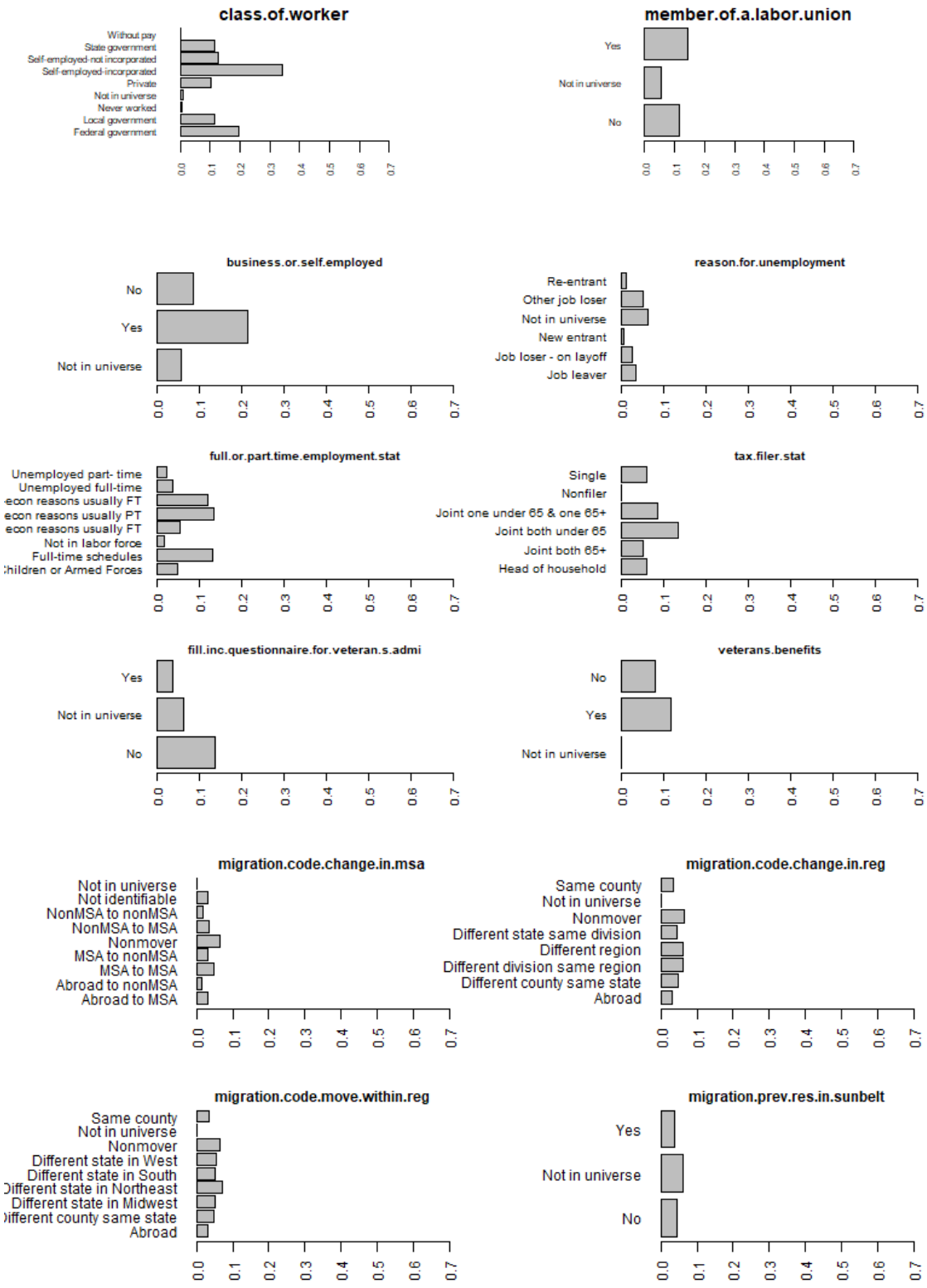

enroll.in.edu.inst.last.wk
Not in universe
High school
College or university
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


detailed.industry.recode.
Armed Forces last job, currently unemployed
Other Public Administration
National Security and Internal Affairs
Administration of Human Resource Programs
Justice, Public Order and Safety
Forestry and Fisheries
Other Professional Services
Social Services
Educational Services
Health Services, Except Hospitals
Hospitals
Entertainment and Recreation Services
Personal Services, Except Private Household
Repair Services
Business Services
Private Household Services
Insurance and Real Estate
Banking and Other Finance
Retail Trade
Wholesale Trade
Utilities and Sanitary Services
Communications
Transportation
Leather and leather products
Rubber and miscellaneous plastics products
Petroleum and coal products
Chemicals and allied products
Printing, publishing and allied industries
Paper and allied products
Apparel and other finished textile products
Textile mill products
Tobacco manufactures
Food and kindred products
ilaneous and not specified manufacturing industries
Toys, amusements, and sporting goods
fessional and photographic equipment, and watches
Other transportation equipment
Aircraft and parts
Motor vehicles and equipment
Electrical machinery,equipment, and supplies
Machinery, except electrical
Not specified metal industries
Fabricated metal
Primary metals
Stone clay, glass, and concrete product
Furniture and fixtures
Lumber and wood products, except furniture
Construction
Mining
Other Agriculture
Agriculture Service
Not in universe
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


detailed.occupation.recode
Armed Forces last job, currently unemployed
Forestry and Fishing Occupations
Farm Workers and Related Occupations
Farm Operators and Managers
Other Handlers, Equipment Cleaners. and Laborers
Freight, Stock and Material Handlers
Construction Laborer
Other Transportation Occupations and Material Moving
Motor Vehicle Operators
Fabricators, Assemblers, Inspectors, and Samplers
Machine Operators and Tenders, Except Precision
Other Precision Production Occupations
Construction Trades
Mechanics and Repairers
Personal Service Occupations
Cleaning and Building Service Occupations
Health Service Occupations
Food Service Occupations
Protective Service Occupations
Private Household Service Occupations
Other Administrative Support Occupations, Including Clerical
Mail and Message Distributing
Financial Records, Processing Occupations
Secretaries, Stenographers, and Typists
Computer Equipment Operators
Supervisors - Administrative Support
Sales Related Occupations
Sales Workers, Retail and Personal Services
Sales Representatives, Commodities, Except Retail
Sales Representatives, Finance, and Business Service
Supervisors and Proprietors, Sales Occupations
Technicians, Except Health Engineering. and Science
Engineering and Science Technicians
Health Technologists and Technicians
Other Professional Specialty Occupations
Lawyers and Judges
Teachers, Except College and University
Teachers, College and University
Health Assessment and Treating Occupations
Health Diagnosing Occupations
Natural Scientists
Mathematical and Computer Scientists
Engineers
Management Related Occupations
Other Executive,
Administrators, and Managers
Public Administration
Not in universe
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


major.industry.code
Wholesale trade
Utilities and sanitary services
Transportation
Social services
Retail trade
Public administration
Private household services
Personal services except private HH
Other professional services
Not in universe or children
Mining
Medical except hospital
Manufacturing-nondurable goods
Manufacturing-durable goods
Hospital services
Forestry and fisheries
Finance insurance and real estate
Entertainment
Education
Construction
Communications
Business and repair services
Armed Forces
Agriculture
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


major.occupation.code
Transportation and material moving
Technicians and related support
Sales
Protective services
Professional specialty
Private household services
Precision production craft & repair
Other service
Not in universe
Machine operators assmblrs & inspctrs
Handlers equip cleaners etc
Farming forestry and fishing
Executive admin and managerial
Armed Forces
Adm support including clerical
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7



Au vu des graphiques ci-dessus il est envisageable de fusionner certaines modalités, par exemple les modalités *Other Rel <18…* de la variable *detailed.household.and.family.stat*, qui sont rares (cf. Section 4.1.2) et pour lesquels le niveau de revenu est faible.

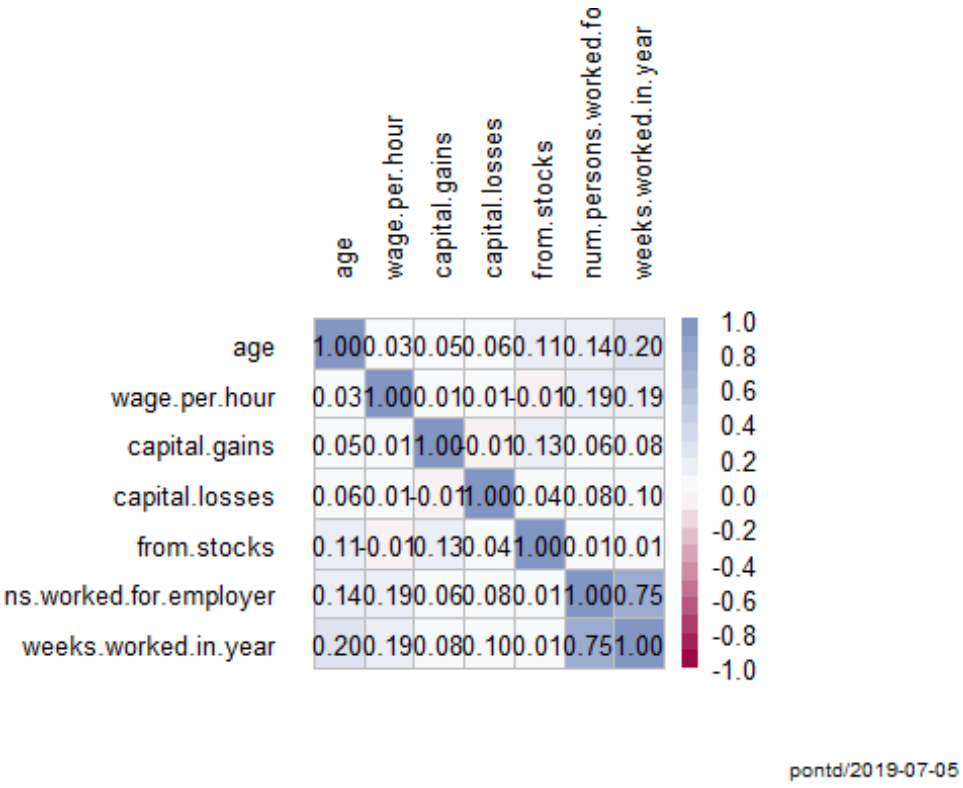
4.2.2 Lien entre variables explicatives

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

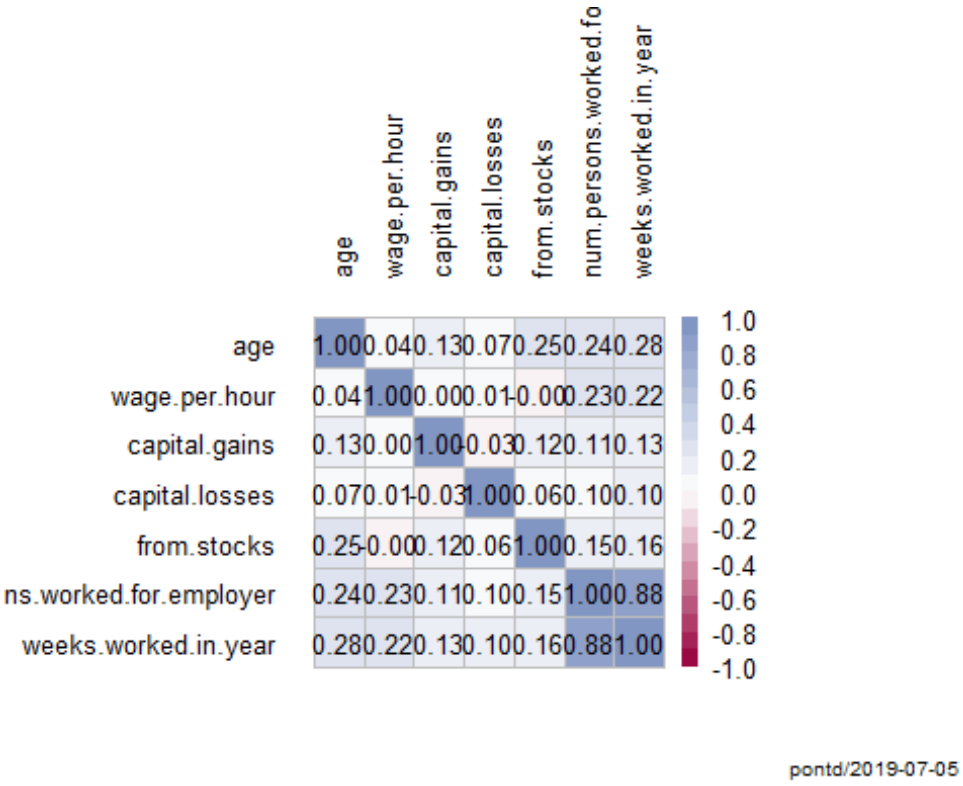
On détermine les valeurs des coefficients de corrélation linéaire et de Spearman entre les variables quantitatives.

```
matcor<-cor(data_train[,var.numeric])
PlotCorr(matcor)
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)
```



pontd/2019-07-05

```
matcor<-cor(data_train[,var.numeric],method = "spearman")
PlotCorr(matcor)
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)
```



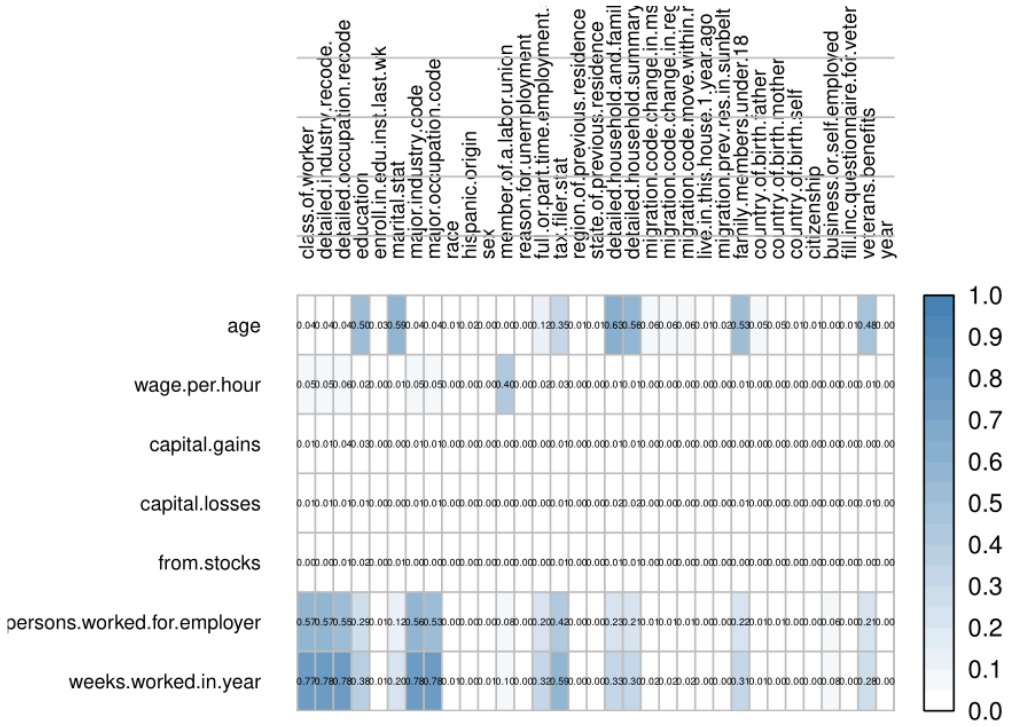
On constate que les coefficients de corrélation et de Spearman sont plutôt proches pour les différents couples de variables. Par ailleurs à l'exception du couple *weeks.worked.in.year*, *num.persons.worked.for.employer,* les liaisons ne sont pas très fortes, on ne s'attend donc pas à des problèmes de colinéarité entre ces variables.

4.2.2.2 Variables quantitatives et qualitatives

On calcule le η^2 entre les variables quantitatives et les variables qualitatives

```
# creation d'une matrice vide avec en ligne les variables quantitatives et
# en colonne les variables qualitatives
mateta2<-matrix(NA,33,7)
rownames(mateta2)<- names(var.factor)[-34]
colnames(mateta2)<- names(var.numeric)

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

# affichage
PlotCorr(mateta2,
         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),
         cex.axis =0.7 )
text(x=rep(1:nrow(mateta2),ncol(mateta2)),
     y=rep(1:ncol(mateta2),each=nrow(mateta2)),
```

```
    label=sprintf("%0.2f", mateta2[,ncol(mateta2):1]),
    cex=0.3,
    xpd=TRUE
    )
```



On note les corrélations suivantes :

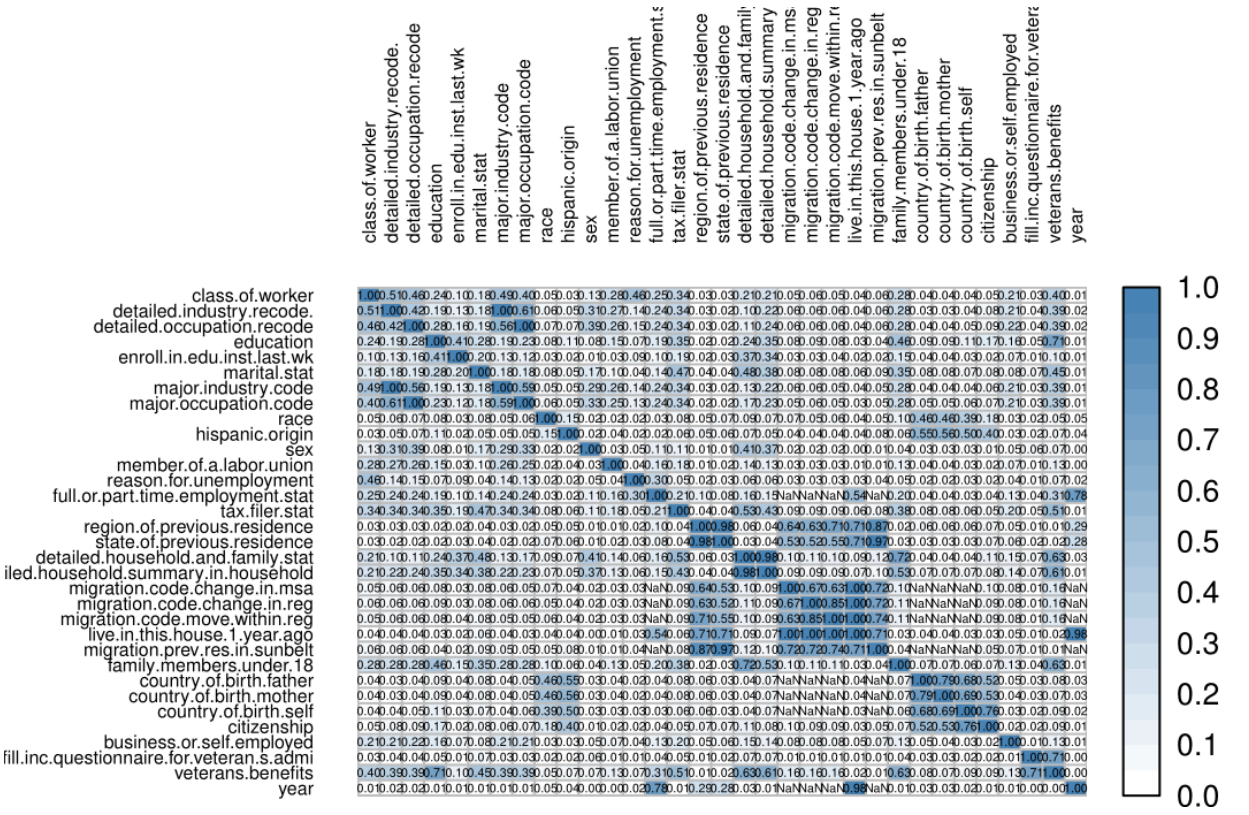
- De façon assez naturelle la variable *age* est liée aux variables *education*, *marital.stat*, *detailed.household.and.family.stat*, *detailed .household.summary.in.household*, *family.members.under.18*, *veterans.benefits*
- La variable *wage.per.hour* est liée à la variable *member.of.a.labor.union* (aux USA le fait d'appartenir à un syndicat offre des garanties quant au salaire)
- Les variables *weeks.worked.in.year* et *num.persons.worked.for.employer* sont liées aux variables *class.of.worker*, *detailed.industry.recode*, *detailed.occupation.recode*, *major.industry.code*, *major.occupation.code* et *tax.filer.sta*". Cela peut s'expliquer par le fait que le travail à temps partiel ou intérimaire concerne plus certains secteurs d'activité et certains postes que d'autres.

4.2.2.3 Variables qualitatives

On détermine les V de Cramer entre les variables explicatives qualitatives.

```
matcram<-PairApply(data_train[,var.factor[-34]], 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),
         cex.axis =0.5)
```

```
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=0.3, xpd=TRUE)
```



On remarque la présence de blocs de variables fortement corrélées entre elles, et donc l'interprétation est plutôt évidente :

- *migration.code.change.in.msa*,*migration.code.change.in.reg*, *migration.code.move.within.reg*, *live.in.this.house.1.year.ago*, *migration.prev.res.in.sunbelt*
- *region.of.previous.residence*, *sate.of.previous.residence* et variables *migration.code….*
- *country.of.birth.father*, *country.of.birth.mother*, *country.of.birth.self*
- *citizenship* et variables *country.of.birth….*
- *race*, *hispanic.origin* et variables *country.of.birth….*
- *detailed.industry.recode*, *detailed.occupation.recode*, *major.industry.code*, *major.occupation.code*, *class.of.worker*
- *detailed.household.and.family.stat* et *detailed.household.summary.in.household*
- "fill.inc.questionnaire.for.veteran.s.*admin*", "veterans.benefits"

Au-delà de ces liens simples, on note aussi les corrélations suivantes:

- *veterans.benefits* avec *family.members.under.18*, *detailed.household.and.family.stat* et *detailed.household.summary.in.household*. Ces corrélations s'expliquent par le fait que les bénéficiaires d'une allocation pour vétérans ont un certain âge, âge qui est lui-même lié à certaines situations familiales, par exemple le fait d'être marié, d'avoir des enfants de plus de 18 ans, etc. Une autre variable fortement associée à *veterans.benefits* et la

variable *education*. Pour mieux comprendre la nature de ce lien on peut construire la table de contingence :

```
table(data_train[,c("education","veterans.benefits")])
```

```
##                                          veterans.benefits
## education                                 Not in universe   Yes    No
##    10th grade                                           1    89  7429
##    11th grade                                           0    39  6906
##    12th grade no diploma                                0    18  2182
##    1st 2nd 3rd or 4th grade                             0    40  1782
##    5th or 6th grade                                     0    58  3306
##    7th and 8th grade                                    0   175  7911
##    9th grade                                            0    61  6162
##    Associates degree-academic program                   0    71  4250
##    Associates degree-occup /vocational                  0    72  5228
##    Bachelors degree(BA AB BS)                           0   204 19786
##    Children                                         47423     0    15
##    Doctorate degree(PhD EdD)                            0    18  1254
##    High school graduate                                 0   619 47601
##    Less than 1st grade                                  0     7   824
##    Masters degree(MA MS MEng MEd MSW MBA)               0    92  6428
##    Prof school degree (MD DDS DVM LLB JD)               0    19  1741
##    Some college but no degree                           0   414 27301
```

Dans ce cas on voit immédiatement la corrélation entre la modalité *children* de la variable *education* et la modalité *Not in universe* de *veterans.benefits*

Pour finir on note une corrélation importante entre la variable *year* et les variables *live.in.this.house.1.year.ago* et *full.or.part.time.employment.stat*. Il est difficile d'expliquer quels facteurs ont pu changer entre 1994 et 1995 pour expliquer un tel lien. Il peut s'agir d'une différence de méthodologie dans le recensement.

4.3 Analyse multivariée

Cette analyse a pour but de mettre en évidence les liaisons entre variables explicatives ainsi que les groupes d'individus au profil similaire. Nous effectuerons une ACM complétée par une CAH sur les composantes principales.

4.3.1 Analyse factorielle

On commence par discrétiser les variables quantitatives. On opte pour un découpage quantiles en 4 classes pour la variable *age*. Pour les variables *capital.gains*, *capital.losses*, *from.stocks* et *wage.per.hour* qui comportent plus de 90% de valeurs égales à zéro on réalise un découpage binaire (0, > 0). Pour la variable *weeks.worked.in.year*, on crée 4 catégories : 0, [1,25],[26,51] et 52. Enfin on laisse la variable *num.persons.worked.for.employer* (qui ne comporte que 7 valeurs) inchangée.

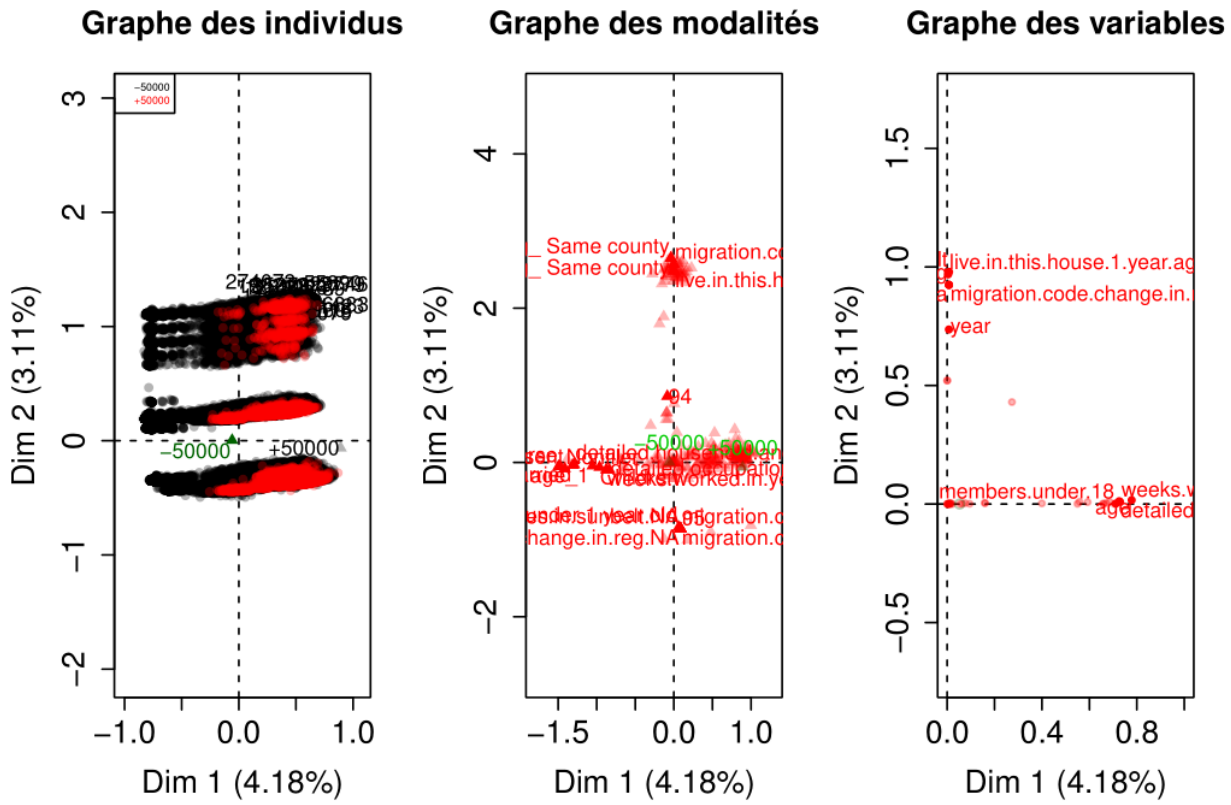
```
data_train_cat <- data_train

# Traitement de "age"
breaks<-c(-Inf,quantile(data_train_cat[["age"]],
                        na.rm=T)[-1])
data_train_cat[["age"]]<-cut(data_train_cat[["age"]],
```

```
                      breaks=breaks,labels=F);
data_train_cat[["age"]]<-as.factor(data_train_cat[["age"]])

# Traitement de "capital.gains","capital.losses","from.stocks","wage.per.hour"
vars.0.spike <- c("capital.gains","capital.losses","from.stocks","wage.per.hour")
for(v in vars.0.spike)
{
    data_train_cat[data_train_cat[[v]]>0,v] <- 1
    data_train_cat[[v]] <- as.factor(data_train_cat[[v]])
}

# Traitement de "weeks.worked.in.year"
data_train_cat[ data_train_cat[["weeks.worked.in.year"]]>0 &
                data_train_cat[["weeks.worked.in.year"]]<26
               ,"weeks.worked.in.year"] <- 1
data_train_cat[ data_train_cat[["weeks.worked.in.year"]]>25 &
                data_train_cat[["weeks.worked.in.year"]]<52
               ,"weeks.worked.in.year"] <- 2
data_train_cat[ data_train_cat[["weeks.worked.in.year"]]==52 ,"weeks.worked.in.year"] <- 3
data_train_cat[["weeks.worked.in.year"]] <- as.factor(data_train_cat[["weeks.worked.in.year"]])

# Traitement de "num.persons.worked.for.employer"
data_train_cat$num.persons.worked.for.employer <- as.factor(data_train_cat$num.persons.worked.f
or.employer)
```

On effectue l'ACM en représentant les graphe des individus, des modalités, et des variables

```
#On réalise l'ACM en mettant la variable Outcome en variable illustrative.
#On gère les modalités rares (fréquence relative <5%) en effectuant de la ventilation
library(FactoMineR)


res.mca<-MCA(data_train_cat,graph=FALSE,quali.sup=ncol(data_train_cat),level.ventil = 0.05)

#On affiche les graphiques relatifs au premier plan
par(mfrow=c(1,3))
#individus (aux plus fortes contributions)
plot.MCA(res.mca, choix="ind",
         habillage = as.numeric(ncol(data_train_cat)),
         invisible="var",
         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.MCA(res.mca, choix="ind",
         label = "var",
         invisible="ind",
         cex.lab=1.5,
         cex.main=1.5,
         cex.axis=1.5,
         col.lab = TRUE,
         title="Graphe des modalités",
         selectMod = "contrib 20")
# on ajoute les modalités de la variable illustrative
text(res.mca$quali.sup$coord[,1:2],labels = rownames(res.mca$quali.sup$coord),pos = 3,col=3)

#variables (aux plus fortes coordonnées)
plot(res.mca, choix="var",
```

```
    select="coord 10",
    title="Graphe des variables",
    cex.lab=1.5,
    cex.main=1.5,
    cex.axis=1.5)
```



Le graphe des individus et celui des modalités montrent clairement une structure. Les individus ayant des revenus supérieurs à 50 K$ annuel sont regroupés sur la partie droite du graphe, pour laquelle Dim1 > 0. Dim2 en revanche ne semble pas très discriminante pour séparer les revenus élevés des autres revenus (on note simplement un regroupement en clusters lié aux variables *migrations…* : le cluster du bas correspond aux individus pour lesquels ces variables ont des données manquantes - 50% de la population - celui du milieu correspond à la modalité *Non mover* et le cluster du haut aux autres modalités).

Pour avoir plus d'informations sur Dim1 on peut utiliser la bibliothèque FactoInvestigate ou simplement les fonctions summary() et dimdesc()

```
summary(res.mca)
dimdesc(res.mca)
```

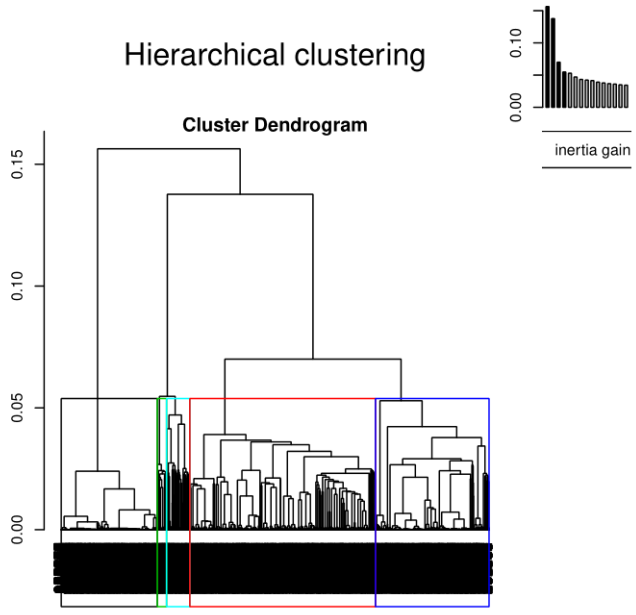
Les résultats de ces fonctions (non reproduits dans leur intégralité ici car particulièrement volumineux) montrent que les variables qui contribuent le plus à Dim1 sont *age*, *weeks.worked.in.year*, *family.members.under.18*, *veterans.benefits*, *num.persons.worked.for.employer*, *tax.filer.stat*, *detailed.household.summary.in.household class.of.worker*, *education*. Quant aux modalités associées aux salaires élevées on retrouve les adultes (*age* supérieur au premier quartile) les personnes travaillant (*weeks.worked.in.year*

>0), et celles pour lesquelles les variables *capital.gains*, *capital.losses* et *from.stocks* sont différentes de 0.

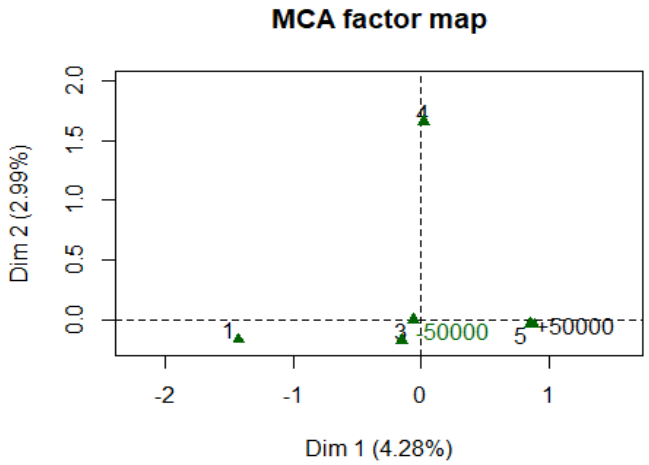
4.3.2 Classification

On complète cette analyse par une CAH sur les composantes de l'ACM. Pour cela, on retient les premières composantes telles que l'inertie cumulée atteigne 80%

```
set.seed(0)

# le traitement  ne pouvant être réalisé sur l'ensemble des données
#  on réalise un échantillonage sratifié de 10% de la population
strate1<- which(data_train_cat$outcome == "+50000")
strate0<- which(data_train_cat$outcome == "-50000")
ech.strat.1<-strate1[sample(seq(length(strate1)),
                            size=ceiling(length(strate1)/10))]
ech.strat.0<-strate0[sample(seq(length(strate0)),
                            size=ceiling(length(strate0)/10))]
## les données issues de chaque strate sont alors agrégées
ech.strat<-c(ech.strat.1,ech.strat.0)
data_train_cat_ech <- data_train_cat[ech.strat,]

# Choix du nombre de composantes ncp telles que l'inertie cumulée atteigne 80%
ncp<-which(res.mca$eig[,3]>80)[1]

# On effectue l'ACM en conservant les ncp premières dimensions
res.mca<-MCA(data_train_cat_ech,graph=FALSE,quali.sup=ncol(data_train_cat_ech),
             level.ventil = 0.05,ncp=ncp)

# On effectue la CAH (avec 5 classes d'après le diagramme des gains d'inertie)
res.cah<-HCPC(res.mca,nb.clust=5,graph=FALSE, description = FALSE)

# On affiche le dendogramme
plot(res.cah, choice="tree")
```



```
# On représente les classes sur le graphe de l'ACM
set.seed(0)
res.mca.clust<-MCA(res.cah$data.clust,
                   graph=FALSE,
                   quali.sup=c(ncol(data_train_cat_ech),ncol(res.cah$data.clust)),
                   level.ventil = 0.05)
plot.MCA(res.mca.clust,
         habillage = ncol(res.cah$data.clust),
         choix="ind",invisible=c("ind","var"))
```



Ce graphe confirme que la 1ère dimension résume bien la structure en classes.

5 Pré-traitement

5.1 Données manquantes

Pour les variables *country…* et *state.of.previous.residence* qui ont des taux de données manquantes faibles (inférieurs à 3.5%) on remplace les valeurs manquantes par une modalité spécifique.

Quant aux variables *migrations…* qui comportent 50% de données manquantes et pour lesquelles la corrélation avec la variable *outcome* est faible, on les supprime.

```
library(forcats)

vars_country_state <- c("country.of.birth.father", "country.of.birth.mother",
                        "country.of.birth.self", "state.of.previous.residence")

#remplace les valeurs manquantes par la modalité "(Missing)"
for(v in vars_country_state )
{
    data_train[,v]<-fct_explicit_na(data_train[,v])
    data_test[,v]<-fct_explicit_na(data_test[,v])
}
```

```
drop.cols <- colnames(data_train)[ apply(data_train, 2, anyNA) ]
training <-  data_train %>% select(-one_of(drop.cols))
```

5.2 Variables quantitatives

5.2.1 Linéarité

Dans le paragraphe 4.2.1.1, nous avons constaté que les liens entre les variables outcome et les variables qualitatives ne sont pas linéaires. On peut bien sûr essayer d'améliorer la linéarité avec des transformations adaptées (notamment pour la variable *num.persons.worked.for.employer*). Pour les autres variables la linéarisation est plus délicate à mettre en œuvre. Ce point a joué un rôle important dans le choix des méthodes d'apprentissage (cf. chapitre 6).

5.2.2 Normalité

Dans le paragraphe 4.2.1.3, nous avons constaté que les distributions des variables continues conditionnellement à la variable réponse ne sont pas normales. Dans le cas des variables *capital.gains*, *capital.losses*, *from.stocks* et *wage.per.hour* en particulier il ne s'agit pas d'une simple asymétrie : on observe un véritable pic en zéro. Dans ce cas une simple transformation logarithmique (ou même de Box-Cox) ne permet pas forcément de se ramener de manière satisfaisante à la normalité et il peut s'avérer plus pertinent de traiter séparément les valeurs strictement positives de la valeur zéro… Quitte à transformer la variable quantitative originale en variable binaire.

5.3 Variables qualitatives

Sur les 40 variables explicatives, 33 sont qualitatives. L'analyse exploratoire a mis en évidence une hétérogénéité importante parmi ces variables tant en termes de cardinalité que de distribution des modalités :

- Certaines variables n'ont que 2 ou 3 modalités (ex: *sex*, *member.of.a.labor.union*), d'autres une cinquantaine (*state.of.previous.residence*, *detailed.occupation.recode*,…)
- Plus du tiers des variables ont une modalité largement majoritaire (fréquence d'apparition supérieure à 85%)
- Plus des 2/3 des variables présentent des modalités rares (fréquence d'apparition inférieure à 5%)

Ces hétérogénéités (notamment la présence de modalités rares) posent des problèmes et peuvent mettre en défaut aussi bien d'éventuelles transformations en amont de la classification (ex: MCA) que l'algorithme d'apprentissage supervisé lui-même (ex: régression logistique). Il est donc intéressant de voir quelles corrections on peut apporter.

Certaines variables nominales comportant des modalités rares peuvent être ignorées dans un premier temps car peu discriminantes. C'est le cas des variables *country….* D'une part la corrélation entre ces variables et le niveau de revenu est limitée (cf. analyse bivariée). D'autre part sur les 42 modalités que comportent ces variables 41 sont rares dont 40 avec une fréquence d'apparition inférieure à 1%. Prises individuellement ces modalités ne peuvent pas peser de manière significative dans le modèle prédictif. Les fusionner n'aurait pas beaucoup de sens car cela effacerait toute différence de niveaux de revenus d'un pays à l'autre. Le même raisonnement d'applique à la variable *state.of.previous.residence*.

Dans d'autre cas on peut envisager des regroupements de modalités. Mieux : ces regroupements sont déjà inclus dans le jeu de données original. Si on s'intéresse par exemple aux informations relatives au foyer des personnes recensées, on trouve deux variables dont l'une (*detailed.household.and.family.stat*) n'est que la version plus détaillée de l'autre (*detailed.household.summary.in.household*). Utilisée dans une optique de classification en niveaux de revenus, la version détaillée offre plus de bruit que d'informations pertinentes par rapport à la version synthétique (cf. analyse bivariée) on peut donc l'ignorer.

Cependant l'intérêt de ce type de regroupement n'est pas toujours évident. Considérons par exemple *detailed.occupation.recode*. Cette variable offre des informations sur la même caractéristique (le type de poste professionnel occupé par la personne recensée) que *major.occupation.code* avec une granularité plus fine (47 modalités contre 15). Sachant que sur les 47 modalités de *detailed.occupation.recode.* il y a une forte proportion de modalités rares il est tentant de ne prendre en compte dans le modèle que *major.occupation.code* et d'écarter *detailed.occupation.recode.* Le problème c'est que la différence de granularité des modalités n'est pas directement corrélée au niveau de revenus. Ainsi parmi les individus dont *major.occupation.code* est *Professional specialty* on trouve plusieurs *detailed.occupation.code* avec des niveaux de revenus très différents :

- *Health Diagnosing Occupations* (taux de revenus >50K$: presque 70%)
- *Other Professional Specialty Occupations* (taux de revenus >50K$: inférieur à 30%)

Dans ce cas il est difficile de juger a priori si le fait de ne garder que la variable synthétique (*major.occupation.code*) a un impact positif sur le modèle prédictif (diminution du bruit, meilleur généralisation) ou si le fait d'éliminer les détails fournis par *detailed.occupation.recode.* conduit à une perte d'informations significatives.

5.2 Réduction des données

5.2.1 En lignes

Durant la phase d'apprentissage de certains modèles, notamment pour le réglage des hyperparamètres, nous travaillerons sur des échantillons stratifiés respectant la proportion bas revenu / haut revenu. En effet, vu le nombre de données (près de 200000 individus), évaluer les performances relatives de chaque jeu de paramètres sur l'ensemble de la population prendrait trop de temps. Un exemple de code implémentant cet échantillonnage a été utilisé dans le paragraphe 4.3.2 Classification.

5.2.2 En colonnes

L'exploration des données nous a amené à envisager d'exclure certaines colonnes correspondant à des variables peu discriminantes (ex: *country…*) et/ou comportant un taux de données manquantes élevées (*migration…*).

Un autre facteur à considérer est que certains algorithmes de classification (ex: SVM) ne gèrent pas les variables explicatives nominales mais uniquement les variables numériques. Une solution classique pour palier à cette limitation est d'utiliser un encodage "one-hot" : chaque variable nominative est transformée en autant de variables binaires qu'elle comporte de modalités. Or notre jeu de données comporte des variables avec plusieurs dizaines de modalités. Ainsi à partir des 40 variables explicatives originales, l'encodage one-hot produit 470 variables ce qui augmente de façon importante le coût de traitement.

En écartant les variables *country…*,*migration…*, *state.of.previous.residence* et *detailed.household.and.family.stat* l'encodage ne produit plus que 236 variables soit un gain appréciable.

On peut si nécessaire, au risque de perdre des informations pertinentes, procéder à un filtrage plus agressif en excluant *detailed.industry.recode.* et *detailed.occupation.recode* qui :

- Sont très corrélées aux variables *major.industry.code* et *major.occupation.code*
- Comportent près de 100 modalités (dont de nombreuses modalités rares).

Enfin, une solution classique pour réduire le nombre de colonnes est de procéder à une analyse factorielle afin de ne retenir que les composants principaux résultant. Malheureusement, les déséquilibres importants dans la distribution de nos données (dues aux variables nominales comportant à la fois des modalités rares non fusionnables et ultra-majoritaires) biaisent fortement ce type de réduction. Nous avons donc évité cette option.

6 Apprentissage supervisé

Les méthodes supervisées étudiées dans le cadre de STA211 couvrent un vaste panel de techniques. Il aurait été impossible de les comparer toutes exhaustivement dans le cadre du challenge. Néanmoins ces méthodes peuvent être regroupées en quelques grandes familles d'algorithmes, les algorithmes d'une même famille ayant des propriétés et des performances comparables pour un problème donné.

Nous avons donc choisi de nous concentrer sur trois familles différentes et retenu une implémentation pour chacune d'elle. Le processus de sélection de ces familles s'est d'abord effectué par élimination selon un critère simple - l'adéquation entre :

- Les propriétés des familles algorithmiques vues en cours
- Les spécificités du jeu de données mis en évidence durant le pré-traitement

Nous avons ainsi rapidement écarté les réseaux de neurones. Ils peuvent s'avérer extrêmement performants mais sont intéressants surtout :

- Dans des domaines impliquant des signaux bas niveaux (reconnaissance d'image, reconnaissance vocale, …)
- Avec des configurations matérielles (GPU, clusters…) autorisant la mise en œuvre de réseaux profonds.

De plus, ils présentent des difficultés importantes. Leur paramétrage (ex. : choix du nombre de couches, du nombre de neurones par couche, etc..) est délicat, leur coût en ressources matérielles est important et leur durée d'apprentissage potentiellement élevée. C'est pourquoi, étant dans un contexte de classification binaire sur des données de relativement haut niveau , et avec des ressources matérielles limitées (PC portable standard) nous avons choisis d'éliminer cette option.

Nous avons également choisi d'écarter les méthodes paramétriques (régression logistique, analyse linéaire discriminante…). En effet l'exploration des données n'a pas permis de vérifier les hypothèses de linéarité et de normalité du lien entre les variables explicatives et la variable réponse, souvent importantes pour ces méthodes. Nous aurions pu, en transformant les variables originales, nous rapprocher de ces hypothèses, mais nous avons simplement préféré privilégier les méthodes plus indépendantes de la forme du lien entre variables d'entrée et variable de sortie.

Ainsi nous avons finalement retenu :

- Une méthode simple et efficace dans une grande variété de situations : **SVM**
- Une méthode de bagging : **forêt aléatoire**
- Une méthode de boosting : **xgboost**

Nous avons employé SVM afin d'obtenir un score de référence, quant aux deux dernières méthodes, de type ensembliste, nous les avons choisies car elles :

- Sont bien adaptées à des problèmes de classification où les variables explicatives sont de nature mixte (qualitative et continues)
- Ne requièrent pas d'hypothèse particulière sur la distribution des données
- Ne sont pas perturbées par la présence de variables fortement corrélées (exemple : *major.occupation.code*, *detailed.occupation.recode*)
- Sont simples à mettre en œuvre
- Ne nécessitent pas un pré-traitement sophistiqué ou des réglages poussés pour obtenir de très bonnes performances
- Apparaissent fréquemment dans les solutions remportant les challenges de type "kaggle"

6.1 SVM

Notre première tentative basée sur SVM, a consisté à exécuter naïvement la méthode "svmLinear" du package R "caret", avec les paramètres par défaut et une validation croisée k-fold (en choisissant k=10) sur l'ensemble des données d'apprentissage. Le programme a procédé à l'ajustement du modèle toute une nuit et au bout de 11 heures l'apprentissage n'était pas terminé. Nous avons alors décidé :

- D'utiliser un échantillonnage stratifié plutôt que l'ensemble du jeu de données pour l'apprentissage.
- D'éliminer la surcouche logicielle fournie par caret et d'utiliser une implémentation directe de SVM. En R il existe plusieurs solutions, les plus connues étant "e1071" et "kernlab". Cette dernière plus récente, offre un choix de noyaux plus large et a la réputation d'être la plus rapide (cf. https://www.thekerneltrip.com/statistics/kernlab-vs-e1071/). C'est donc elle que nous avons retenue.

Nous avons ensuite procédé au réglage de l'hyperparamètre principal (le coût C) avec:

- Une validation croisée k-fold (k=3) portant sur un échantillon d'apprentissage représentant 40% des données étiquetées (ce taux d'échantillonnage assez faible -en général on utilise plutôt 60% des données étiquetées - a été contraint par la nécessité de réaliser l'apprentissage en moins de 8h : traitement démarré le soir et contrôlé le matin suivant)
- Une grille de 8 valeurs pour C
- Le noyau par défaut recommandé par les auteurs de kernlab : RBF

```
library(kernlab)

# échantillon stratifié de 40% des données étiquetées
# utilis pour l'apprentissage
training_sample
#le reste des données étiquetées est utilisé pour le test
test_sample

# création de la grille d'hyperparamètres
costs <- c(0.1, 0.25, 0.5 ,1,2,4,8,16)

bestCost <- 0
bestError <-1

# parcours de la grille
for( cost in costs)
{
    # ajustement du modèle avec validation 5 plis
    # (mécanisme intégré à la bibliothèque kernlab)
    model <- ksvm(outcome ~ ., training_sample,
                  kernel="rbfdot",
                  scaled=TRUE,C=cost,cross=5)



    # calcul des prédictions
    pred <- predict(model,test_sample)
```

```
    test_sample$prediction <- pred

    error <- nrow(test_sample[test_sample$outcome!=test_sample$prediction,]) /
                nrow(test_sample)

    if(error < bestError){
        bestError <-error
        bestCost <- cost
        print(paste("!!!! bestError",bestError," bestCost",bestCost,sep=" "))
        write.csv(pred, "predictions.csv",row.names = FALSE)
    }
}
```

Au bout de quelques heures nous avons obtenu :

- Le modèle présentant la valeur de C optimale parmi celles proposées (4)
- **Une première estimation du taux de mauvais classement : 0.0489** (mesurée sur les 60% de données initiales constituant l'échantillon de validation)

Remarque : nous aurions pu poursuivre l'étude des modèles SVM en gardant la valeur optimale de l'hyperparamètre C et en créant une nouvelle partition apprentissage-test avec un ratio plus important de données d'apprentissage. Cependant nous avons préféré privilégier les forêts aléatoires, celles-ci s'étant rapidement avérées plus performantes.

6.2 Bagging : Forêt aléatoire

Les forêts aléatoires sont une méthode ensembliste basée sur les arbres de décision. Elles incluent donc le processus de sélection des variables, peuvent être appliquées directement sur un jeu de données où les liaisons entre variables sont fortes et ne nécessitent pas de pré-traitement importants. Par rapport aux arbres de décision simples, les forêts aléatoires possèdent un avantage important : une amélioration accrue de la stabilité mais aussi de la qualité des résultats. Le prix à payer pour ces avantages - une plus grande complexité rendant l'interprétation du lien entre les entrées et la sortie difficile (contrairement aux arbres de décision) - n'est pas un problème dans une optique de performance prédictive.

Il existe différente solutions en R : nous avons choisi la bibliothèque "ranger" pour sa rapidité d'apprentissage https://arxiv.org/pdf/1508.04409.pdf

D'autre part, nous avons utilisé une méthodologie analogue à celle employée pour le modèle SVM :

1. Sélection d'un échantillon d'apprentissage représentant 40% des données (ratio relativement faible afin de limiter le temps de traitement),
2. Parcours d'une grille afin de déterminer la valeur optimale du principal hyperparamètre (nombre d'arbres : *mtry*),
3. Choix du meilleur modèle avec un processus de validation croisée k-fold (k=5) intégrée à la bibliothèque,
4. Estimation de l'erreur de généralisation sur les 60% de données non employées pour l'apprentissage.

```
library(ranger)

# définition de la grille de tuning : on recherche la valeur optimale du paramètre
# "mtry" (nombre de variables tirées aléatoirement à chaque noeud de l'arbre)
tuneGrid <- expand.grid(mtry=c(10,15,20,25,30,35,40,45),splitrule="gini",min.node.size=1)

# l'estimation de l'erreur pour chaque valeur de la grille est réalisée
# avec le mécanisme de validation croisée  intégrée à la bibliothèque ranger
trControl <- trainControl(method="cv", number=5,search="grid", verboseIter = TRUE)

rangerTuning <- train(outcome~., data=training_sample,
                      method="ranger",
                      metric="Accuracy",
                      tuneGrid=tuneGrid,
                      trControl=trControl)


# on extrait le paramètre mtry optimal et on l'utilise pour construiree modèle final
# sur l'ensemble des données d'apprentissage
tuneGrid$mtry <- rangerTuning$bestTune$mtry
rangerFinal <- train(outcome~., data=training,
                     method="ranger", metric="Accuracy",
                     tuneGrid=tuneGrid,
                     trControl=trControl)

predictions<-predict.train(object=rangerFinal,testing)

write.csv(predictions, "predictions.csv",row.names = FALSE)
```

L'erreur estimée est de 0.0434. Les résultats sont nettement meilleurs que ceux obtenus avec SVM, pour une durée d'entraînement du même ordre de grandeur.

6.3 Boosting : Xgboost

Xgboost (https://arxiv.org/pdf/1603.02754.pdf) a été créé dans le cadre d'un projet de recherche par Tianqi Chen, puis a rejoint les bibliothèques open-source développées par la DMLC (Distributed Machine Learning Community). Il s'agit d'une solution qui est devenue incontournable dans les compétitions de type « Kaggle » tant ses performances sont élevées.

Xgboost fait partie des méta-algorithmes de boosting, plus précisément de "gradient boosting". L'idée centrale du "gradient boosting" est née de l'observation de Leo Breiman selon laquelle le boosting peut être interprété comme un algorithme d'optimisation sur une fonction de coût appropriée. Dans cas, il est donc possible d'employer la méthode de descente du gradient pour effectuer l'optimisation. Xgboost implémente ce principe et a été développé avec pour objectifs spécifiques de :

- Réduire drastiquement le temps d'apprentissage des modèles
- Exploiter au maximum chaque bit de mémoire disponible.

Concrètement, dans le cadre du challenge et avec un PC portable standard (Core i5 7ème génération, 16Go de RAM, 256Go SSD, pas de carte graphique dédiée, Windows 10 home) :

- La création d'un modèle basé sur 40% des données d'apprentissage et une validation k-fold (k=5) prend de **50 minutes à 1h** pour kernlab (SVM) comme pour ranger (forêts aléatoires).
- Xgboost quant à lui, réalise la phase d'apprentissage sur l'ensemble des données avec une validation croisée k-fold (k=10) en seulement **10 minutes**… avec une précision nettement supérieure comme nous allons le voir.

Le programme xgboost que nous avons développé est similaire à ceux vus précédemment pour les autres méthodes. La principale différence est le nombre d'hyperparamètres pris en compte dans la grille lors de la phase de tuning. Xgboost permet de spécifier une dizaine d'hyperparamètres. Nous nous sommes concentrés sur trois principaux, ayant observé peu d'améliorations en faisant varier les autres. Ces trois paramètres sont :

- *Colsample_bytree* : la proportion des variables explicatives à prendre ne compte
- *Max_depth* : la profondeur maximale de chaque arbre
- *Min_child_weight* : si la partition de l'arborescence aboutit à un nœud feuille dont la somme des poids est inférieure à min_child_weight, le processus stoppe le partitionnement

Une autre particularité de xgboost est que pour une configuration d'hyperparamètre donnée, P, l'apprentissage s'effectue en deux phases :

- Une première phase, avec validation croisée, permet de déterminer en fonction de P, le nombre d'itérations optimale et l'erreur de généralisation associée
- Une deuxième phase est alors requise pour construire le modèle final à partir de P et du nombre d'itérations optimale.

Le code (simplifié) est le suivant :

```
library(xgboost)

dtrain <- xgb.DMatrix(data = sparse.model.matrix(outcome~.-1, data = training),
                      label=as.numeric(training$outcome)-1)
testing$outcome=0
dtest <- sparse.model.matrix(outcome~.-1, data = testing)

# définition de la grille d'hyperparamètres
colsamples_bytree <- seq(from=0.35,to=1,by=0.025)
max_depths <- 6:7
min_child_weights <- 1:2

# initialisation de l'objets contenant les  valeurs d'hyperparamètres optimales
best_params <- c()
best_params$colsample_bytree <- 1
best_params$max_depth <- 1
best_params$min_child_weight <- 1
best_params$error <- 1
```

```
for(colsample_bytree in colsamples_bytree)
{
    for (max_depth in max_depths){
        for (min_child_weight in min_child_weights){

            # définition d'un jeu de paramètres à partir des valeurs courantes de la grille
            param <- list(objective   = "binary:logistic",
                          eval_metric = "error",
                          colsample_bytree = colsample_bytree,
                          max_depth   = max_depth,
                          eta         = 0.1,
                          min_child_weight = min_child_weight)

            # ETAPE 1
            # recherche du nombre d'itérations optimal pour le jeu de paramètres "param"
            xgbcv <- xgb.cv(params = param
                            ,data = dtrain
                            ,nrounds = 800
                            ,nfold = 10
                            ,showsd = T
                            ,stratified = T
                            ,print_every_n  = 10
                            ,early_stopping_rounds = 50
                            ,maximize = F
            )

            # ETAPE 2
            # apprentissage du modèle final avec "param"
            # et le nombre d'itération déterminée à l'étape 1
            xgbmodel <- xgboost(params  = param,
                                data    = dtrain,
                                nthread = 2,
                                nrounds = xgbcv$best_iteration,
                                print_every_n = 100,
                                verbose = 1)

            #erreur de test estimée lors de l'étape 1
            error <- xgbcv$evaluation_log$test_error_mean[xgbcv$best_iteration]


            if(error < best_params$error){
                best_params$max_depth <- max_depth
                best_params$colsample_bytree <- colsample_bytree
                best_params$min_child_weight <- min_child_weight
                best_params$best_iteration <- xgbcv$best_iteration
                best_params$error <- error

                # ETAPE 3: CALCUL DES PREDICTIONS
                predictions <- predict(xgbmodel, dtest)
                predictions[predictions<0.5]=-1
                predictions[predictions>=0.5]=1
                predictions[predictions==-1]="-50000"
                predictions[predictions==1]="+50000"
                write.csv(predictions,"predictions.csv" ,row.names = FALSE)
            }
            i <- i+1
        }
    }
}
```

Nous avons obtenu , avec la meilleure configuration d'hyperparamètres , un taux de mauvais classement de **0.0409** (résultat fourni par le mécanise de validation croisée intégré à xgboost) qui s'est traduit au niveau de l'échantillon test de RChallenge par un score de *0.0395*.

A la suite de ce résultat, nous avons observé que certaines valeurs prédites par notre modèle étaient très instables (elles variaient beaucoup en fonction des hyperparamètres). Nous avons alors utilisé une grille de tuning centrée sur les meilleurs hyperparamètres pour produire N prédictions issus de N modèles distincts et procédé à un vote par majorité afin de créer un modèle "moyenné". Ceci nous a permis d'obtenir un score sur l'échantillon de test RChallenge de *0.0394*.

Finalement nous avons inclus les données des test complétées par les valeurs prédites dans le jeu de données d'apprentissage. Ceci nous permis d'obtenir un score de *0.0393* sur l'échantillon de test RChallenge.

7 Conclusion

Commençons par rappeler les scores (taux de mauvais classements), tels que nous les avons évalués (par validations croisées) sur les trois méthodes supervisées étudiées :

SVM (kernlab avec noyau RBF)	Forêt aléatoire (ranger)	Xgboost
0.0489	0.0434	0.0409

Clairement xgboost se détache des deux autres algorithmes, particulièrement de SVM. Il convient cependant de nuancer ces chiffres. Si les trois méthodes ont été présentées séquentiellement dans ce rapport, en pratique nous les avons rapidement évaluées en parallèle à l'issue de la phase de pré-traitement. Constatant que xgboost était la plus prometteuse, nous lui avons consacrées plus du temps et d'effort que les deux autres méthodes, notamment en termes de réglages, ce qui biaise les résultats.

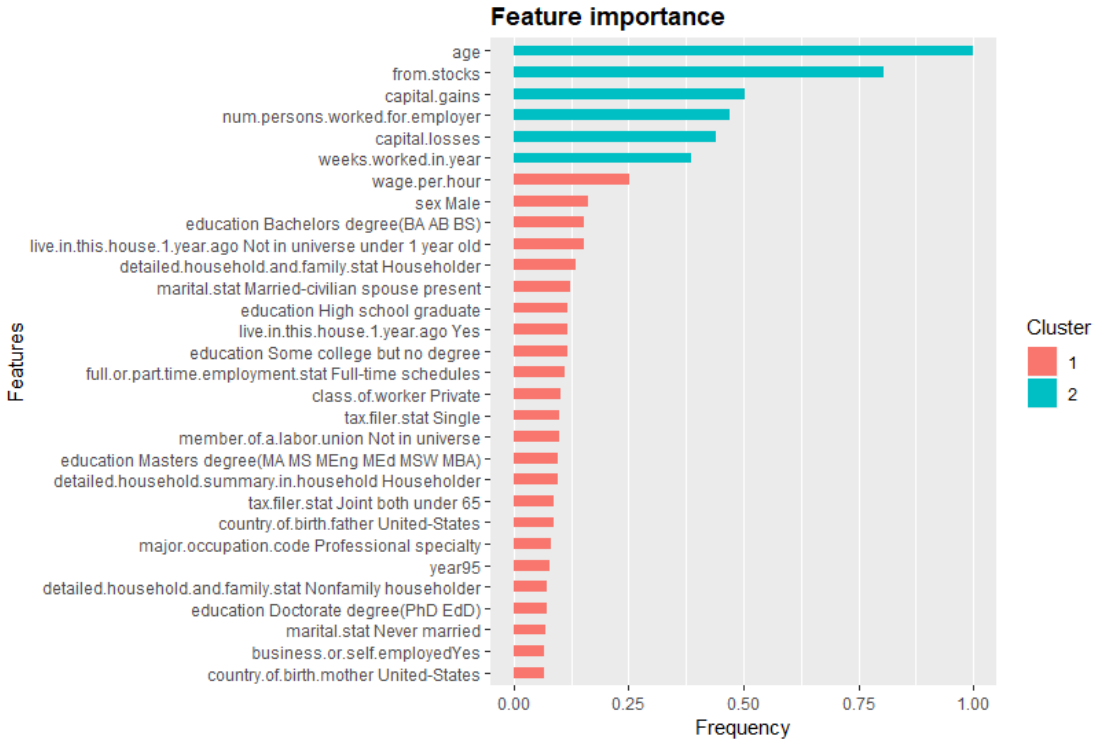
Ceci-dit, même en améliorant le pré-traitement et le réglage des hyperparamètres pour les solutions basées sur SVM et les forêts aléatoires, xgboost arriverait certainement en tête, pour deux raisons :

- **L'efficacité** - Xgboost nous a permis de prendre en compte l'ensemble des données, sans réduction ni en colonnes (excepté les colonnes *migration…* - à cause de leur taux de donnes manquantes de 50%) ni en lignes. Le passage à l'échelle a en revanche constitué un problème à la fois pour SVM et pour les forêts aléatoires. Avec ces deux méthodes nous avons dû réduire à la fois le nombre des individus mais aussi celui des caractéristiques en éliminant en plus des variables *migration…*, les variables *country…*, *state.of.previous.residence* et *detailed…* comme indiqué dans le chapitre sur le pré-traitement.
- **La vitesse de traitement** - Xgboost s'est révélé nettement plus rapide (facteur supérieur à 10), avec notre jeu de données, que les deux algorithmes avec lesquels

nous l'avons mis en concurrence : pour un temps donné, il nous a donc été possible de tester beaucoup plus de configurations différentes (en termes de pré-traitements, de sélection d'hyperparamètres)

Un dernier élément vient confirmer que, dans le cadre notre étude, la supériorité de xgboost par rapport aux forêts aléatoires vient de sa capacité à travailler sur l'ensemble des données sans réduction de dimensions.

Parmi les outils fournis avec xgboost, on trouve des méthodes qui permettent de quantifier l'importance relative des différentes variables explicatives pour un modèle prédictif donné, comme représenté dans le graphique ci-dessous :



En utilisant ces méthodes, nous avons identifié, parmi les caractéristiques générées à partir de l'encodage « one hot » des variables explicatives, les 160 plus « importantes ».

Puis nous avons réexécuté le programme xgboost du paragraphe 6.3. Nous avons alors obtenu un taux d'erreur de 0.0436 soit quasiment le même score que celui obtenu avec les forêts aléatoires. Au vu de ce résultats, on peut également imaginer qu'une réduction des données en colonnes avec une méthode d'analyse factorielle n'aurait sans doute pas été pertinente (tout au moins pour xgboost) .

Pour finir, un mot sur les idées permettant de réduire encore le taux d'erreur. A l'heure actuelle, quasiment toutes les solutions qui remportent des compétitions de type « kaggle » utilisent des techniques dites de stacking / blending (cf. https://mlwave.com/kaggle-ensembling-guide/). Ce type de techniques permettrait certainement d'améliorer les scores que nous avons obtenus.