Projet Actuariat

Solène Corre, Florentin Dehooghe,François Delhaye

04 avril 2020

Table des matières

1 Présentation du projet

2 Exploration des jeux de données freMPL1 et freMPL2

Un peu à la manière du machine learning, les données contenues dans freMPL2 serviront de données d'entraînement de notre modèle et les données de freMPL1 serviront pour tester notre modèle final.

2.1 freMPL2

2.1.1 Première exploration du jeu de données d'entraînement : freMPL2

Dans un premier temps, regardons les premières lignes du jeu de données

	1	2	3
Exposure	0.583	0.416	0.583
LicAge	579	361	366
RecordBeg	2004-06-01	2004-01-01	2004-06-01
RecordEnd	NA	2004-06-01	NA
VehAge	10+	1	2
Gender	Male	Female	Female
MariStat	Other	Other	Other
SocioCateg	CSP60	CSP1	CSP1
VehUsage	Private	Professional	Professional
DrivAge	83	55	55
HasKmLimit	0	0	0
BonusMalus	50	58	72
VehBody	sedan	sedan	sedan
VehPrice	N	D	D
VehEngine	injection	injection	injection
VehEnergy	regular	regular	regular
VehMaxSpeed	190-200 km/h	160-170 km/h	160-170 km/h
VehClass	H	B	B
RiskVar	14	15	15
ClaimAmount	0	0	0
Garage	None	None	None
ClaimInd	0	0	0

Les dimmensions de notre jeu de données sont (48295, 22). Ainsi, notre jeu contient 48295 données différentes, toutes définies par 22 caractéristiques différentes.

Les noms des différentes caractéristiques sont :

- **Exposure** :
- **LicAge** :
- **RecordBeg** :
- **RecordEnd** :
- **VehAge** :
- **Gender** :
- **MariStat** :
- **SocioCateg** :
- **VehUsage** :
- **DrivAge** :
- **HasKmLimit** :
- **BonusMalus** :
- **VehBody** :
- **VehPrice** :
- **VehEngine** :
- **VehEnergy** :
- **VehMaxSpeed** :
- **VehClass** :
- **RiskVar** :
- **ClaimAmount** :
- **Garage** :
- **ClaimInd** :

2.1.2 Nettoyage de données

Remarquons qu'il serait intéressant de faire un peu de nettoyage de données avant d'effectuer quelconques travaux sur celles-ci. Pour cela, nous allons créer une fonction qui servira à nettoyer les 2 dataframes.

Cette fonction (appelée nettoyage_dataframe) prend l'un des deux dataframes en paramètres et effectue les opérations suivantes :

- Suppression des données des individus assurés moins d'un jour (Exposure)
- Modification des données des individus ayant un ClaimAmount négatif
- Suppression de la colonne associée au sexe de la personne
- Réduction du nombre de catégories socioprofessionnels
- Traduction des données (VehBody, MariStat, VehUsage, VehEngine, VehEnergy, Garage)

2.1.3 Statistiques descriptives

Regardons maintenant les différents types d'objets figurant dans les colonnes :

Nous avons donc des objets de type numeric, de type factor, de type int et même de type date.

Regardons maintenant plus précisement les valeurs particulières de ces colonnes (valeurs minimum et maximum, moyenne, médiane,quantiles, ...)

On remarquera qu'il existe des données manquantes dans la colonne RecEnd, ce qui signifie que les individus concernés sont toujours assurés.

On peut aussi utiliser la méthode describe du package Hmisc pour avoir un aperçu de la dispersion des données.

Mais cela ne vaut pas une représentation graphique.

2.1.4 Représentations graphiques des données

```
# On met les colonnes dans le même ordre

freMPL1 <- freMPL1[,c(1:17,19,18,20:21)]
```

```
## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2
```

```
## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2
```

```
## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2

## pdf
##   2
```

2.1.5 ACP

L'ACP permet d'analyser et de visualiser un jeu de données contenant des individus décrits par plusieurs variables quantitatives. C'est une méthode statistique qui permet d'explorer des données dites multivariées (données avec plusieurs variables). Chaque variable pourrait être considérée comme une dimension différente. L'analyse en composantes principales est utilisée pour extraire et de visualiser les informations importantes contenues dans une table de données multivariées. L'ACP synthétise cette information en seulement quelques nouvelles variables appelées composantes principales. Ces nouvelles variables correspondent à une combinaison linéaire des variables originels. Le nombre de composantes principales est inférieur ou égal au nombre de variables d'origine.

2.1.5.1 Exécution sur nos données freMPL2

Attention, les valeurs doivent être numériques.

On va donc convertir nos valeurs en numérique :

```
freMPL2$LicAge <- as.numeric(freMPL2$LicAge)
freMPL2$DrivAge <- as.numeric(freMPL2$DrivAge)
freMPL2$BonusMalus <- as.numeric(freMPL2$BonusMalus)
freMPL2$RiskVar <- as.numeric(freMPL2$RiskVar)
freMPL2$ClaimAmount <- as.numeric(freMPL2$ClaimAmount)
```

Certaines colonnes sont catégorisés et pourraient nous être utiles pour éxécuter notre ACP. Il n'est cependant pas judicieux d'appliquer une conversion numérique à ces colonnes puisqu'on leur attribue une valeur arbitraire nous faisant penser à une classification des différents facteurs possibles. Pour éviter cela, on va donc utiliser la méthode model.matrix() qui crée une matrice binaire spécifiant à quel facteur correspond une ligne du dataframe.

```
library("FactoMineR")
library("factoextra") # Pour la visualisation
```

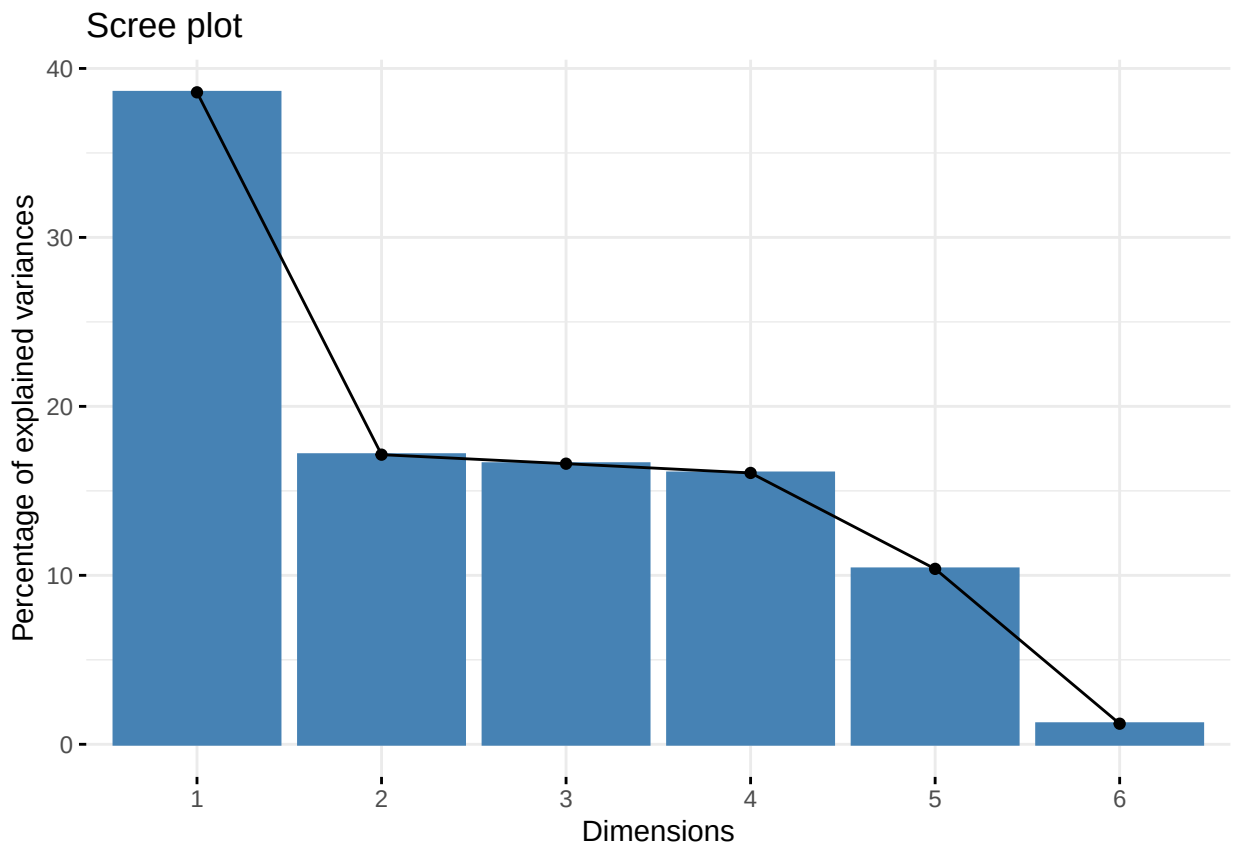
```
freMPL2.active <- freMPL2[,c(1:2,9,11, 18:19)]
freMPL2.pca <- PCA(freMPL2.active, graph = FALSE)
```

Affichage du résultat :

```
get_eigenvalue(freMPL2.pca)
```

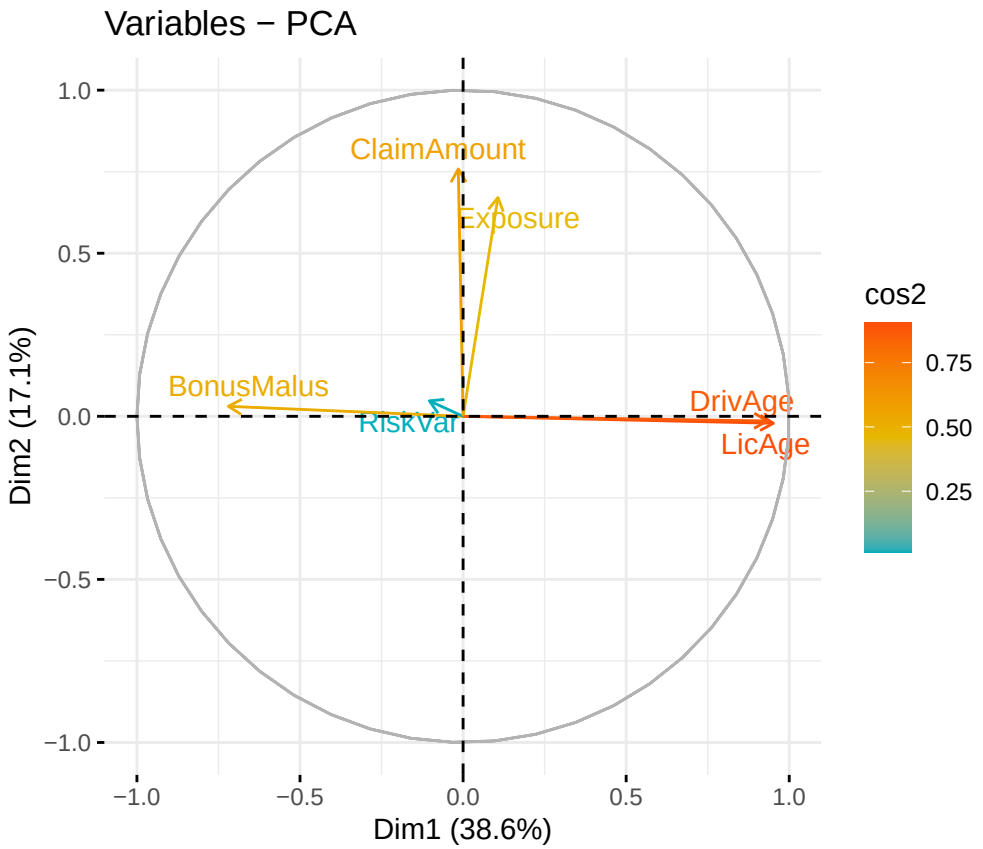
```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 2.31520784        38.586797                    38.58680
## Dim.2 1.02853013        17.142169                    55.72897
## Dim.3 0.99649119        16.608187                    72.33715
## Dim.4 0.96366164        16.061027                    88.39818
## Dim.5 0.62299351        10.383225                    98.78141
## Dim.6 0.07311569         1.218595                   100.00000
```

```
fviz_eig(freMPL2.pca)
```



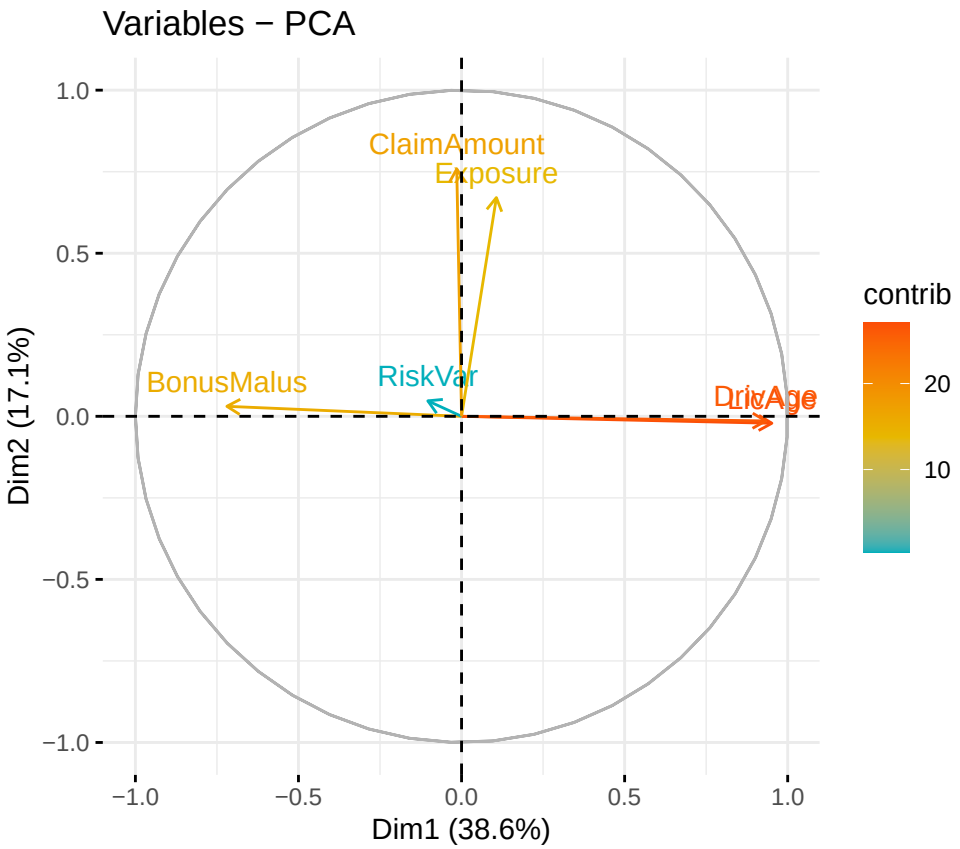
utilisation de cos2 pour juger de la qualité de la représentation :

```
fviz_pca_var(freMPL2.pca, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE # Évite le chevauchement de texte
             )
```



contribution des colonnes aux dimensions :

```
fviz_pca_var(freMPL2.pca, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")
             )
```



Description des dimensions

Dans les sections précédentes, nous avons décrit comment mettre en évidence les variables en fonction de leurs contributions aux composantes principales.

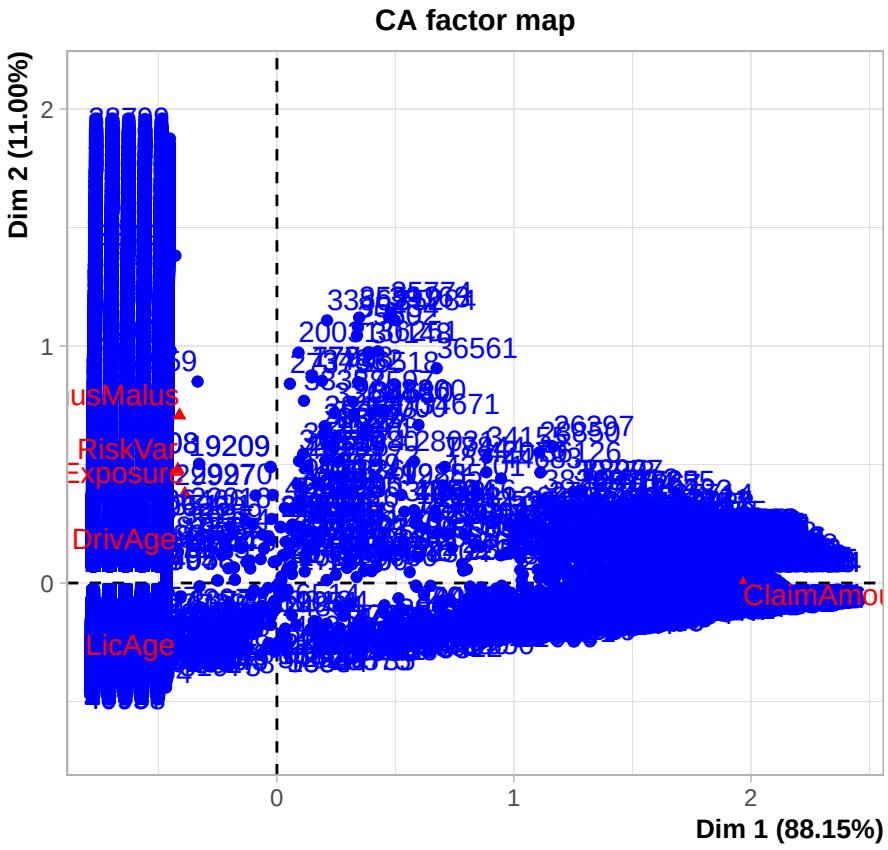
Notez également que la fonction dimdesc() [dans FactoMineR], pour dimension description (en anglais), peut être utilisée pour identifier les variables les plus significativement associées avec une composante principale donnée . Elle peut être utilisée comme suit:

```
freMPL2.desc <- dimdesc(freMPL2.pca, axes = c(1,2), proba = 0.05)
head(freMPL2.desc)
```

AFC

L' analyse factorielle des correspondances est une extension de l'analyse en composantes principales pour analyser l'association entre deux variables qualitatives (ou catégorielles). L'AFC permet de résumer et de visualiser l'information contenue dans le tableau de contingence formé par les deux variables catégorielles. Le tableau de contingence contient les fréquences formées par les deux variables.

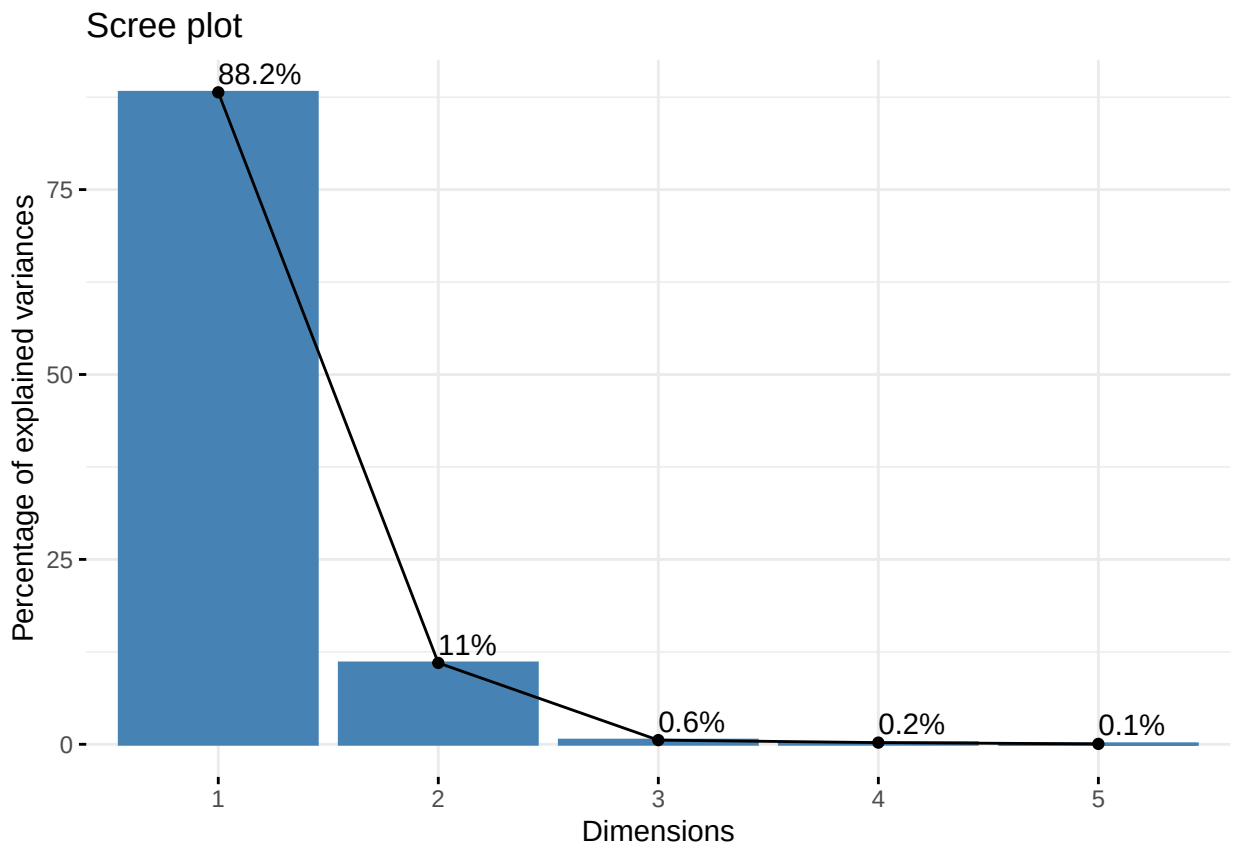
```
library ("FactoMineR")
freMPL2.ca <- CA (freMPL2.active)
```



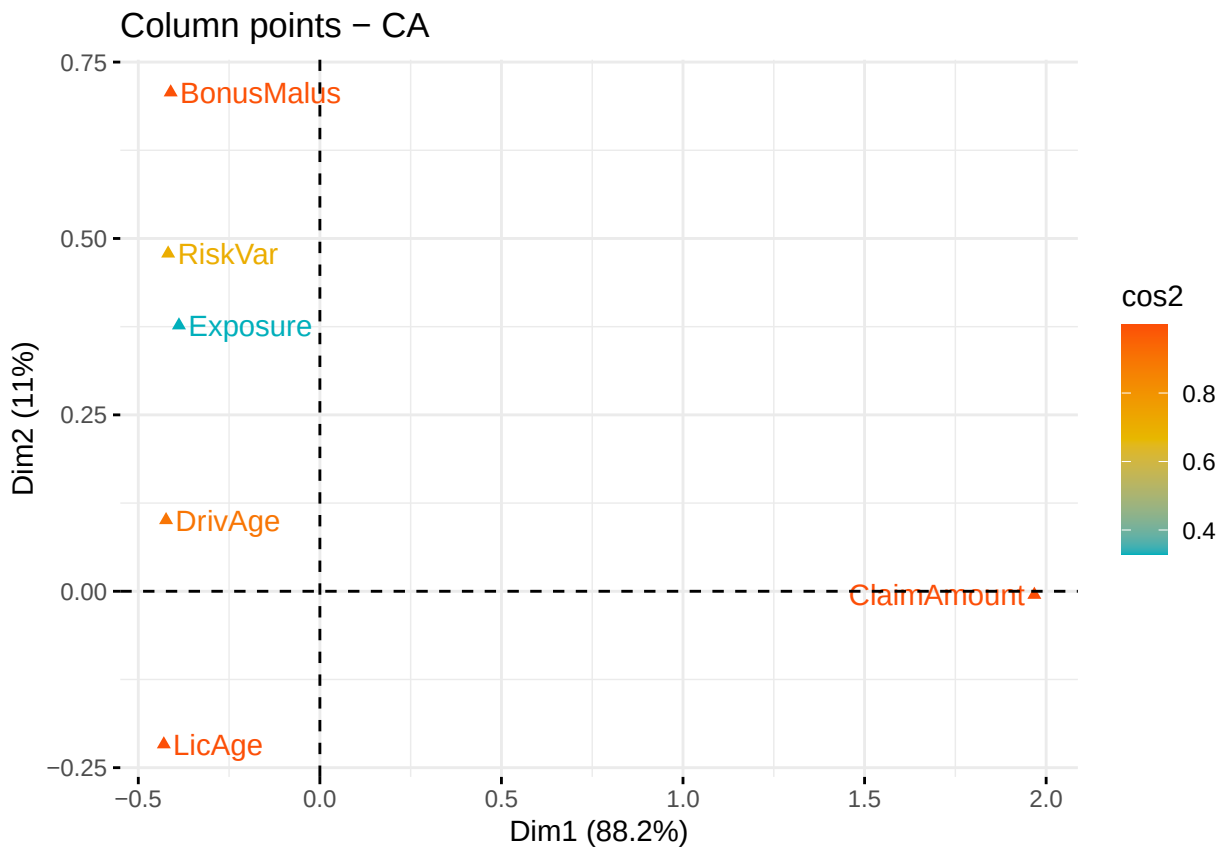

```
get_eigenvalue(freMPL2.ca)
```

```
##         eigenvalue variance.percent cumulative.variance.percent
## Dim.1 0.8368987402      88.15185447                     88.15185
## Dim.2 0.1044266947      10.99942723                     99.15128
## Dim.3 0.0053632523       0.56491976                     99.71620
## Dim.4 0.0021692873       0.22849443                     99.94470
## Dim.5 0.0005250479       0.05530412                    100.00000
```

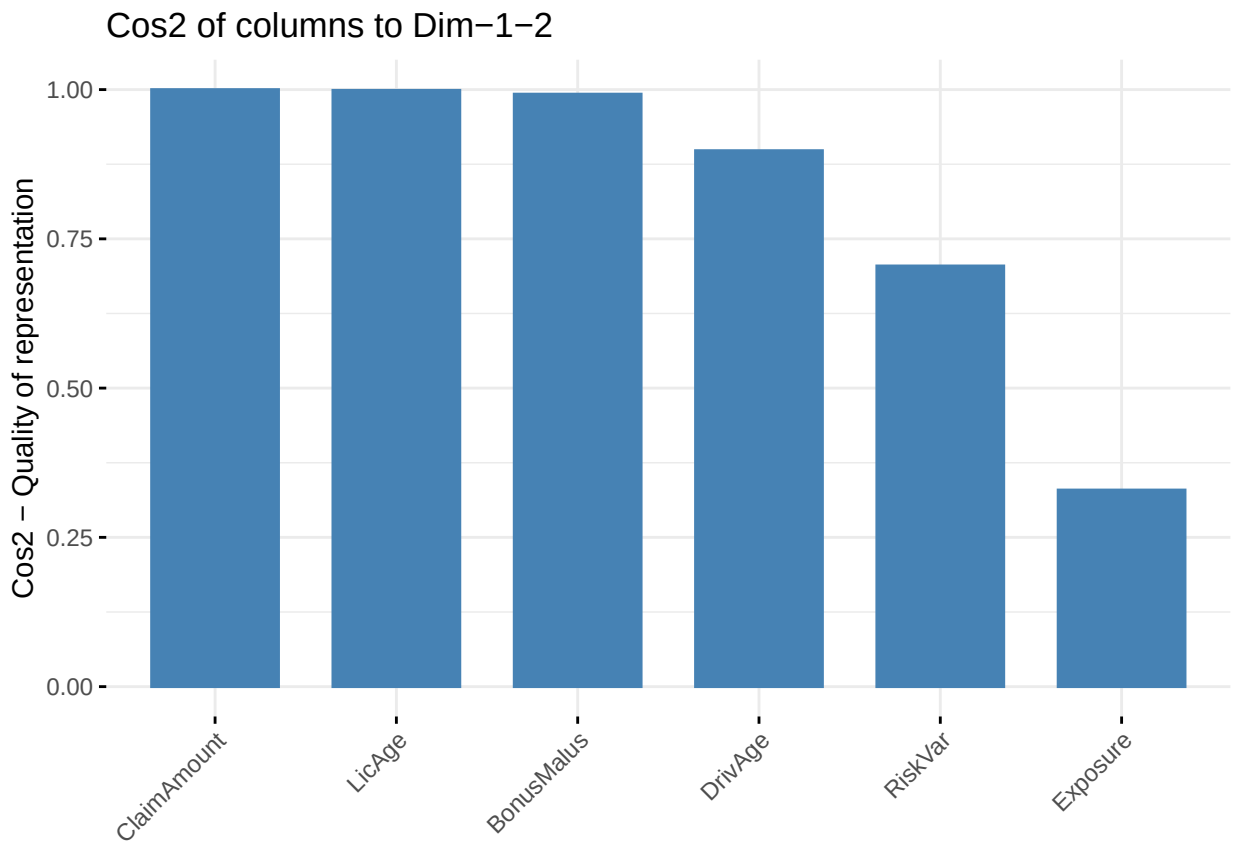
```
fviz_screeplot (freMPL2.ca, addlabels = TRUE)
```




```
fviz_ca_col (freMPL2.ca, col.col = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE)
```




```
fviz_cos2 (freMPL2.ca, choice = "col", axes = 1:2)
```




```
res.desc <- dimdesc(freMPL2.ca, axes = c(1, 2))
res.desc[[1]]$col
```

```
##                   coord
## LicAge      -0.4297012
## DrivAge     -0.4235388
## RiskVar     -0.4177907
## BonusMalus  -0.4107983
## Exposure    -0.3880433
## ClaimAmount  1.9675625
```

3 GLM

```
#calibration d'une loi de Poisson
fpois <- glm(RiskVar~DrivAge+VehAge+VehClass+VehBody+VehEnergy, offset=log(Exposure), family=poisson("l
summary(fpois)
```

```
##
## Call:
## glm(formula = RiskVar ~ DrivAge + VehAge + VehClass + VehBody +
##     VehEnergy, family = poisson("log"), data = freMPL2, offset = log(Exposure))
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.9745  -1.7698   0.2224   2.7176  13.2888
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            3.515e+00  1.210e-02 290.407  < 2e-16 ***
## DrivAge               -2.799e-03  8.888e-05 -31.488  < 2e-16 ***
## VehAge1               -5.541e-02  6.103e-03  -9.079  < 2e-16 ***
## VehAge10+              9.702e-02  4.991e-03  19.437  < 2e-16 ***
## VehAge2               -5.895e-02  6.039e-03  -9.762  < 2e-16 ***
## VehAge3                1.043e-02  6.193e-03   1.685 0.092013 .
## VehAge4                6.832e-03  6.226e-03   1.097 0.272487
## VehAge5                4.730e-02  6.338e-03   7.463 8.44e-14 ***
## VehAge6-7              3.468e-02  5.847e-03   5.932 2.99e-09 ***
## VehAge8-9              6.245e-02  5.815e-03  10.739  < 2e-16 ***
## VehClassA              1.008e-02  8.825e-03   1.142 0.253440
## VehClassB              2.338e-02  7.879e-03   2.968 0.003000 **
## VehClassH             -5.229e-02  8.376e-03  -6.243 4.30e-10 ***
## VehClassM1             3.697e-02  7.786e-03   4.748 2.06e-06 ***
## VehClassM2             1.873e-02  8.196e-03   2.285 0.022285 *
## VehBodymicrovan        9.029e-02  1.042e-02   8.666  < 2e-16 ***
## VehBodyautobus         8.086e-03  2.049e-02   0.395 0.693148
## VehBodycoupé           3.463e-02  9.953e-03   3.480 0.000502 ***
## VehBodyautre microvan  6.972e-03  9.894e-03   0.705 0.481050
## VehBodyberline         4.838e-02  7.617e-03   6.353 2.12e-10 ***
## VehBodySUV             2.242e-02  9.951e-03   2.253 0.024253 *
## VehBodybreak           1.919e-02  9.526e-03   2.015 0.043946 *
## VehBodycamionnette    -3.745e-02  1.021e-02  -3.667 0.000245 ***
## VehEnergyGPL           7.207e-01  1.669e-01   4.319 1.57e-05 ***
## VehEnergyélectrique   -4.001e-01  1.314e-01  -3.045 0.002324 **
## VehEnergyessence      -6.305e-02  3.118e-03 -20.222  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 521321  on 47496  degrees of freedom
## Residual deviance: 516924  on 47471  degrees of freedom
## AIC: 723959
##
## Number of Fisher Scoring iterations: 5
```

```
#calibration d'une loi Poisson sur-dispersée
fpois2 <- glm(RiskVar~DrivAge+VehAge+VehClass+VehBody+VehEnergy, offset=log(Exposure), family=quasipois
summary(fpois2)
```

```
##
## Call:
## glm(formula = RiskVar ~ DrivAge + VehAge + VehClass + VehBody +
##     VehEnergy, family = quasipoisson("log"), data = freMPL2,
##     offset = log(Exposure))
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -7.9745  -1.7698   0.2224   2.7176  13.2888
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.5150786  0.0698155  50.348  < 2e-16 ***
## DrivAge                -0.0027987  0.0005127  -5.459 4.81e-08 ***
## VehAge1                -0.0554086  0.0352004  -1.574 0.115474
## VehAge10+               0.0970155  0.0287893   3.370 0.000753 ***
## VehAge2                -0.0589466  0.0348309  -1.692 0.090583 .
## VehAge3                 0.0104348  0.0357222   0.292 0.770205
## VehAge4                 0.0068320  0.0359108   0.190 0.849115
## VehAge5                 0.0472995  0.0365552   1.294 0.195699
## VehAge6-7               0.0346833  0.0337227   1.028 0.303727
## VehAge8-9               0.0624515  0.0335435   1.862 0.062636 .
## VehClassA               0.0100788  0.0509043   0.198 0.843050
## VehClassB               0.0233825  0.0454447   0.515 0.606886
## VehClassH              -0.0522911  0.0483134  -1.082 0.279111
## VehClassM1              0.0369664  0.0449084   0.823 0.410427
## VehClassM2              0.0187327  0.0472769   0.396 0.691934
## VehBodymicrovan         0.0902890  0.0600976   1.502 0.133007
## VehBodyautobus          0.0080860  0.1181992   0.068 0.945460
## VehBodycoupé            0.0346315  0.0574060   0.603 0.546331
## VehBodyautre microvan   0.0069717  0.0570701   0.122 0.902773
## VehBodyberline          0.0483842  0.0439318   1.101 0.270751
## VehBodySUV              0.0224200  0.0573957   0.391 0.696079
## VehBodybreak            0.0191920  0.0549484   0.349 0.726886
## VehBodycamionnette     -0.0374459  0.0588941  -0.636 0.524899
## VehEnergyGPL            0.7206873  0.9625394   0.749 0.454020
## VehEnergyélectrique    -0.4000731  0.7577335  -0.528 0.597511
## VehEnergyessence       -0.0630509  0.0179840  -3.506 0.000455 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 33.2696)
##
##     Null deviance: 521321  on 47496  degrees of freedom
## Residual deviance: 516924  on 47471  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

4 Bibliographie

5 Annexes

5.1 Affichage de l'implementation de la fonction nettoyage_dataframe :

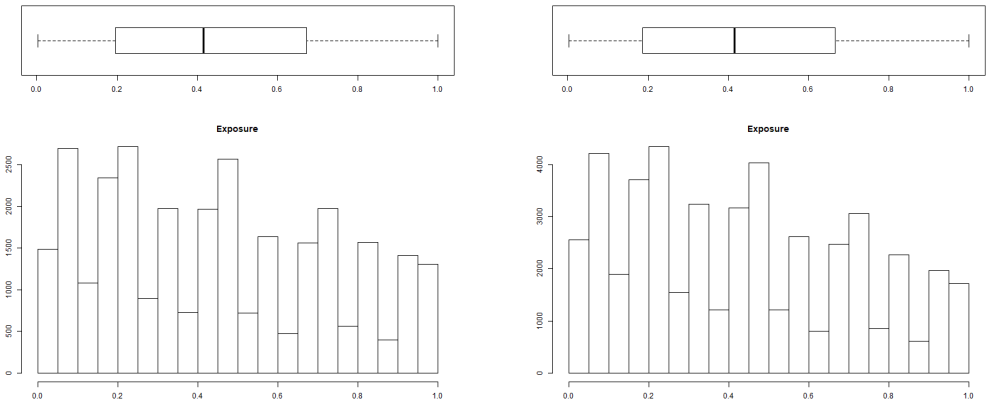
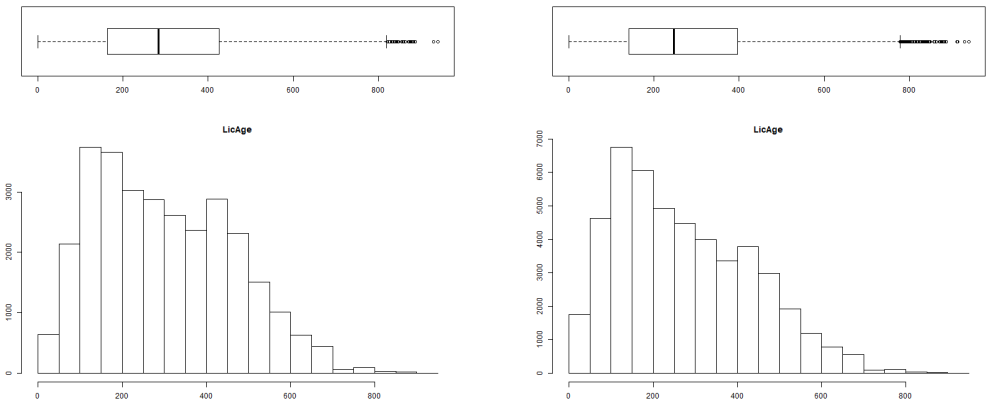
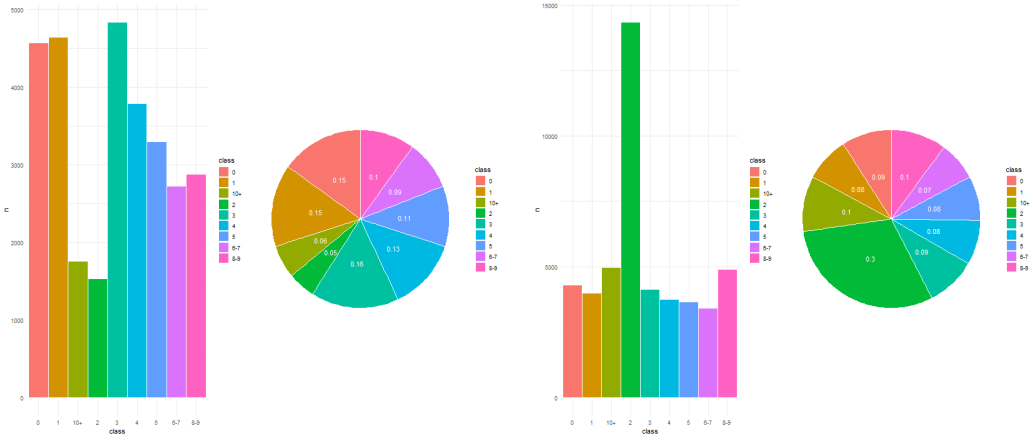
```
nettoyage_dataframe <- function(dt){

  # Suppression des données des individus assurés moins d'un jour (Exposure)
  dt <- subset(dt,dt$Exposure>1/365.25)
```

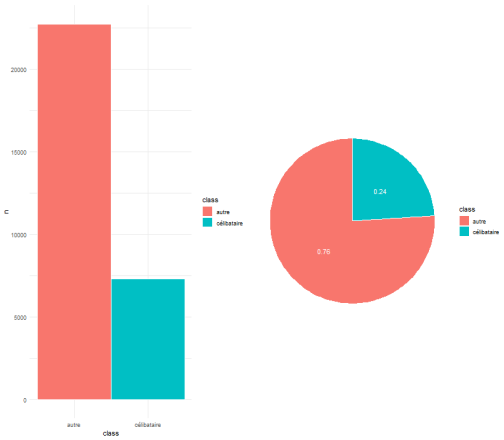
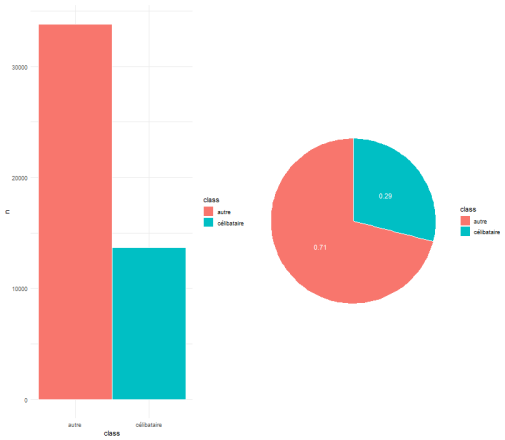
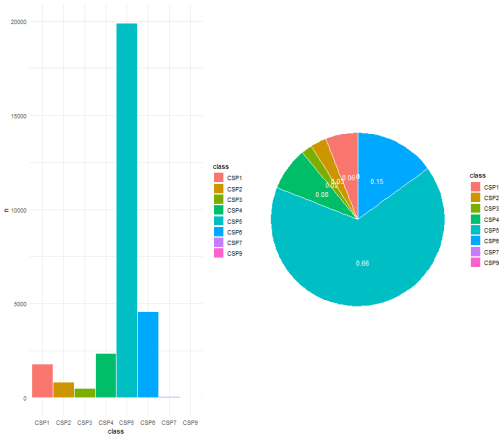
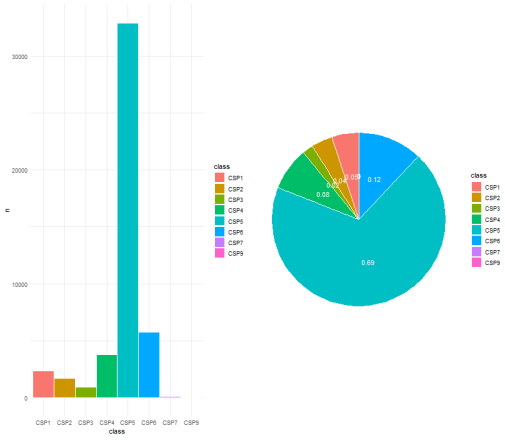
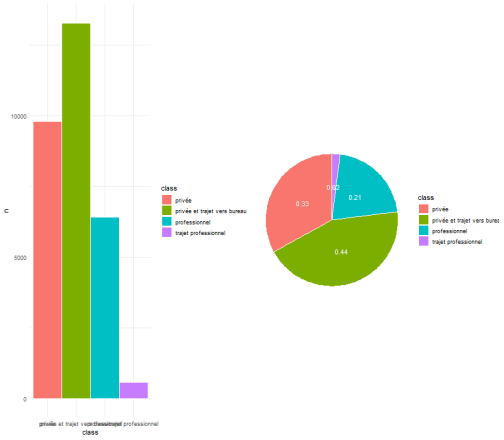
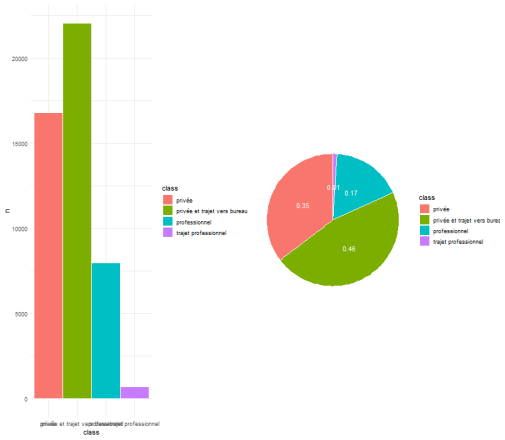
```
# Modification des données des individus ayant un ClaimAmount négatif
dt <- subset(dt,dt$ClaimAmount>=0)

# Suppression de la colonne associée au sexe de la personne
dt <- dt[,-6]

# Réduction du nombre de catégories socioprofessionnels
levels(dt$SocioCateg) <- c(levels(dt$SocioCateg), "CSP4", "CSP6",
                                  "CSP9")
for (i in 1:dim(dt)[1]){
  if (dt$SocioCateg[i]%in%c("CSP1","CSP16","CSP18","CSP19")){
    dt$SocioCateg[i]<-"CSP1"
    }
  if (dt$SocioCateg[i]%in%c("CSP2", "CSP20", "CSP21", "CSP22", "CSP23",
                                  "CSP25", "CSP26","CSP27", "CSP28")){
    dt$SocioCateg[i]<-"CSP2"
    }
  if (dt$SocioCateg[i]%in%c("CSP3", "CSP30", "CSP31", "CSP32", "CSP33",
                                  "CSP35", "CSP36","CSP37", "CSP38", "CSP39")){
    dt$SocioCateg[i]<-"CSP3"
    }
  if (dt$SocioCateg[i]%in%c("CSP40", "CSP41", "CSP42", "CSP43", "CSP46",
                                  "CSP47", "CSP48","CSP49")){
    dt$SocioCateg[i]<-"CSP4"
    }
  if (dt$SocioCateg[i]%in%c("CSP5", "CSP50", "CSP51", "CSP55", "CSP56",
                                  "CSP57", "CSP59")){
    dt$SocioCateg[i]<-"CSP5"
    }
  if (dt$SocioCateg[i]%in%c("CSP6", "CSP60", "CSP61", "CSP62", "CSP63",
                                  "CSP65", "CSP66")){
    dt$SocioCateg[i]<-"CSP6"
    }
  if (dt$SocioCateg[i]%in%c("CSP7", "CSP70", "CSP73", "CSP74", "CSP77")){
    dt$SocioCateg[i]<-"CSP7"
    }
  if (dt$SocioCateg[i]%in%c("CSP9", "CSP91")){
    dt$SocioCateg[i]<-"CSP9"
    }
}
dt$SocioCateg <- droplevels(dt$SocioCateg)

# Traduction des données (VehBody, MariStat, VehUsage, VehEngine, VehEnergy, Garage)
for (i in 1:dim(dt)[2]){
  # Type de véhicules
  if (colnames(dt)[i]=="VehBody"){
    levels(dt$VehBody) <- c(levels(dt$VehBody), "autobus", "coupé",
                                  "autre microvan", "berline","SUV", "break",
                                  "camionnette")
    dt$VehBody[dt$VehBody == "bus"]<-"autobus"
    dt$VehBody[dt$VehBody == "coupe"]<-"coupé"
    dt$VehBody[dt$VehBody == "other microvan"]<-"autre microvan"
    dt$VehBody[dt$VehBody == "sedan"]<-"berline"
```

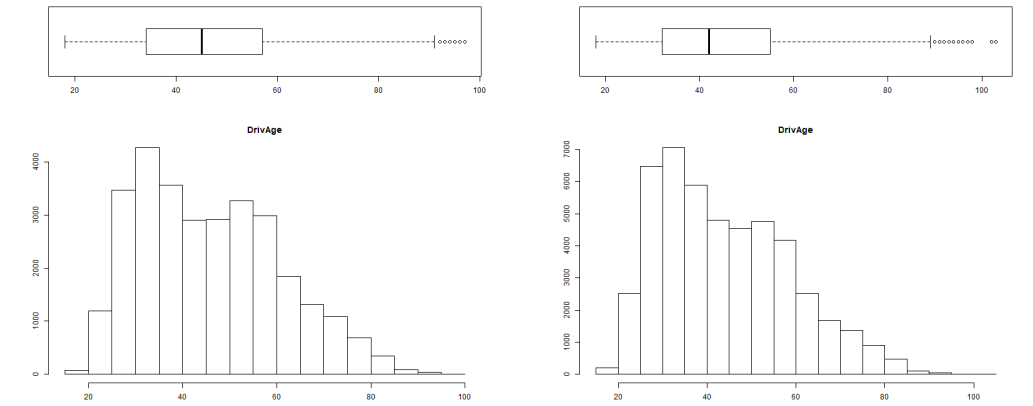
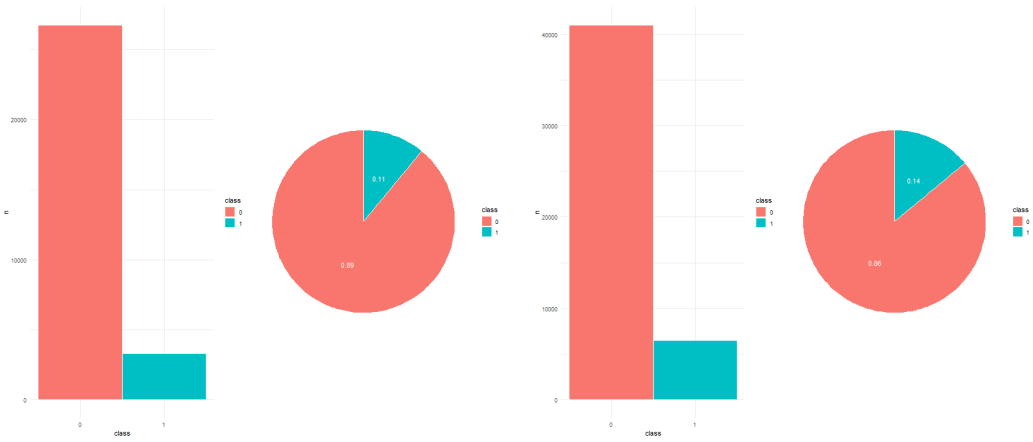
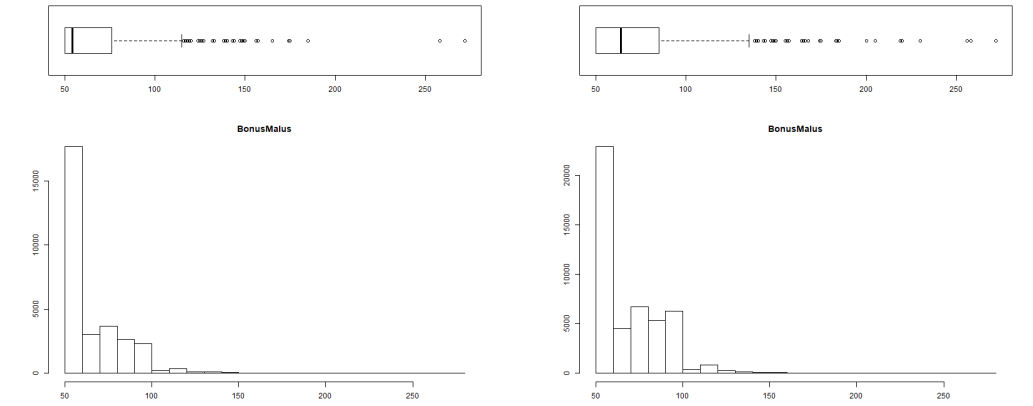
```
    dt$VehBody[dt$VehBody == "sport utility vehicle"]<-"SUV"
    dt$VehBody[dt$VehBody == "station wagon"]<-"break"
    dt$VehBody[dt$VehBody == "van"]<-"camionnette"
    dt$VehBody <- droplevels(dt$VehBody)
    }
  # Statut marital
  if (colnames(dt)[i]=="MariStat"){
    levels(dt$MariStat) <- c(levels(dt$MariStat), "célibataire", "autre")
    dt$MariStat[dt$MariStat == "Alone"]<-"célibataire"
    dt$MariStat[dt$MariStat == "Other"]<-"autre"
    dt$MariStat <- droplevels(dt$MariStat)
    }
  # Utilisation du véhicule
  if (colnames(dt)[i]=="VehUsage"){
    levels(dt$VehUsage) <- c(levels(dt$VehUsage), "privée",
                                  "privée et trajet vers bureau", "professionnel",
                                  "trajet professionnel" )
    dt$VehUsage[dt$VehUsage == "Private"]<-"privée"
    dt$VehUsage[dt$VehUsage == "Private+trip to office"]<-
    "privée et trajet vers bureau"
    dt$VehUsage[dt$VehUsage == "Professional"]<-"professionnel"
    dt$VehUsage[dt$VehUsage == "Professional run"]<-
    "trajet professionnel"
    dt$VehUsage <- droplevels(dt$VehUsage)
    }
  # Moteur du véhicule
  if (colnames(dt)[i]=="VehEngine"){
    levels(dt$VehEngine) <- c(levels(dt$VehEngine),
                                   "injection directe surpuissante",
                                   "électrique", "injection surpuissante")
    dt$VehEngine[dt$VehEngine == "direct injection overpowered"]<-
    "injection directe surpuissante"
    dt$VehEngine[dt$VehEngine == "electric"]<-"électrique"
    dt$VehEngine[dt$VehEngine == "injection overpowered"]<-
    "injection surpuissante"
    dt$VehEngine <- droplevels(dt$VehEngine)
    }
  # Energie utilisée par le véhicule
  if (colnames(dt)[i]=="VehEnergy"){
    levels(dt$VehEnergy) <- c(levels(dt$VehEnergy), "électrique", "essence")
    dt$VehEnergy[dt$VehEnergy == "regular"]<-"essence"
    dt$VehEnergy[dt$VehEnergy == "eletric"]<-"électrique"
    dt$VehEnergy <- droplevels(dt$VehEnergy)
    }
  # Garage
  if (colnames(dt)[i]=="Garage"){
    levels(dt$Garage) <- c(levels(dt$Garage), "aucun", "garage indépendant",
                                 "concessionnaire")
    dt$Garage[dt$Garage == "None"]<-"aucun"
    dt$Garage[dt$Garage == "Private garage"]<-"garage indépendant"
    dt$Garage[dt$Garage == "Collective garage"]<-"concessionnaire"
    dt$Garage <- droplevels(dt$Garage)
  }
}
```

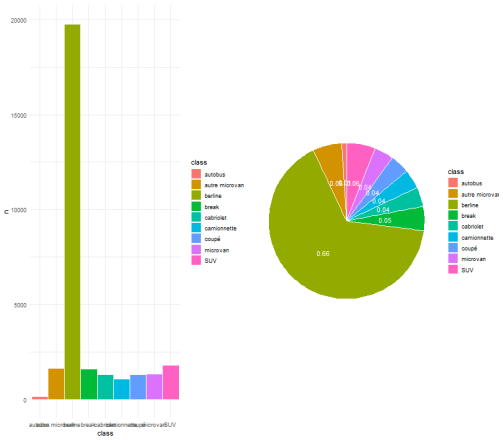
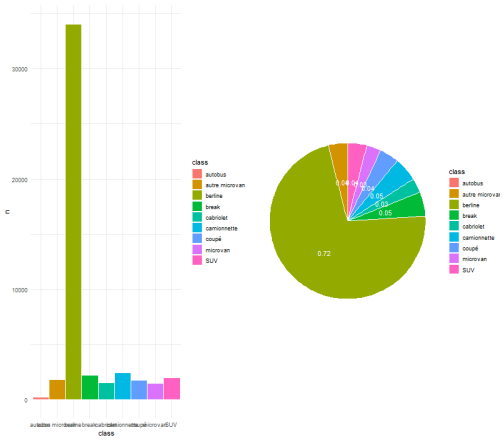
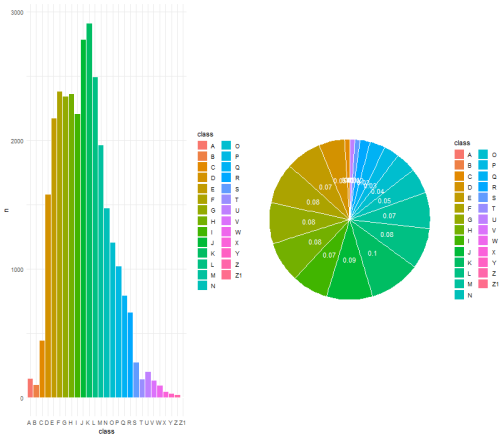
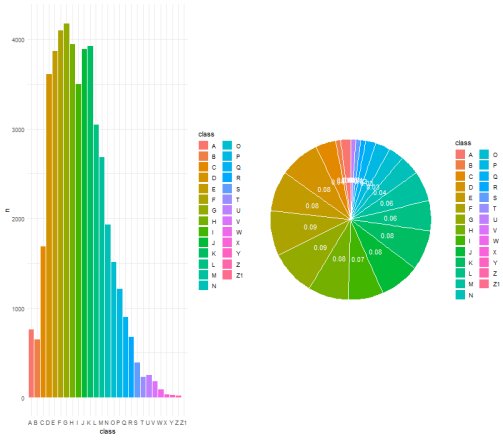
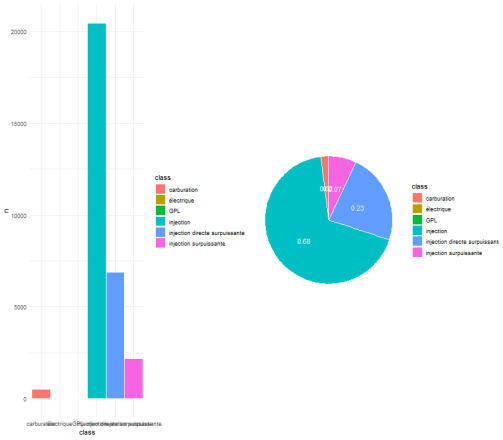
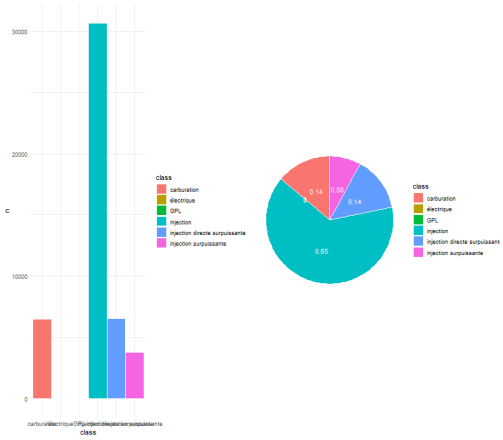
```
  return (dt)
}
```

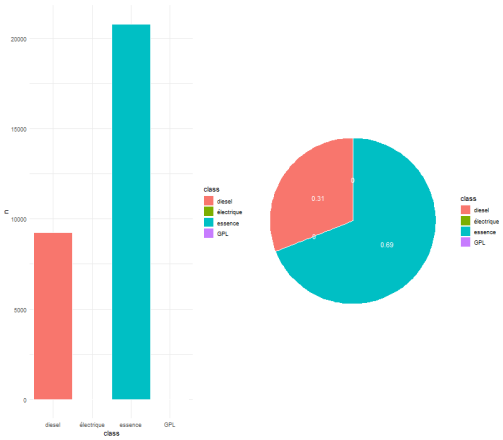
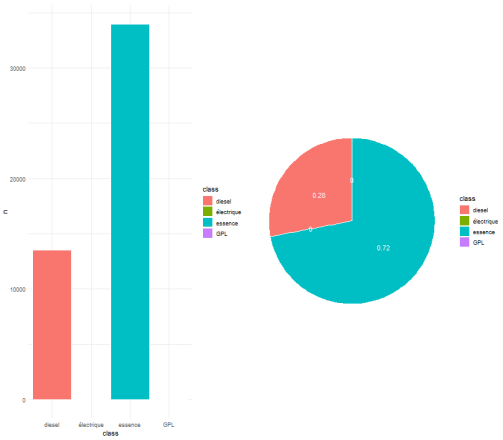
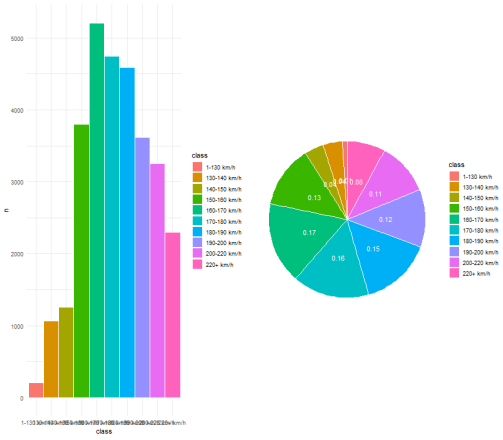
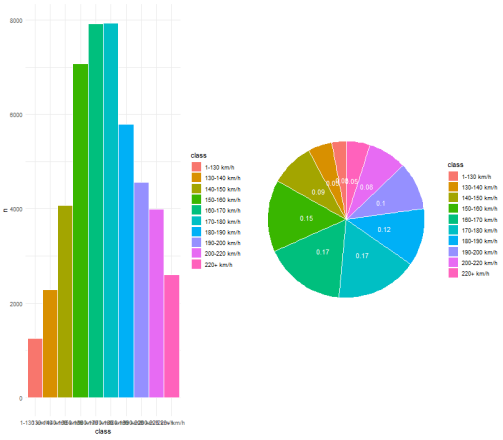
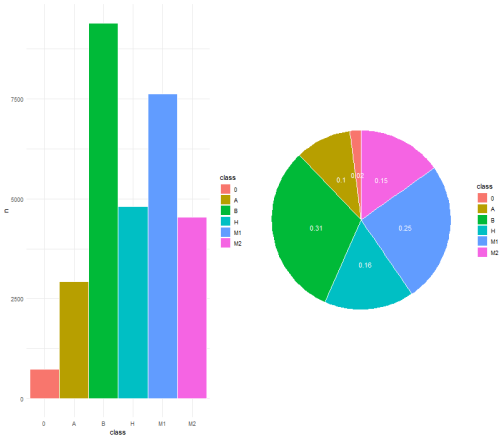
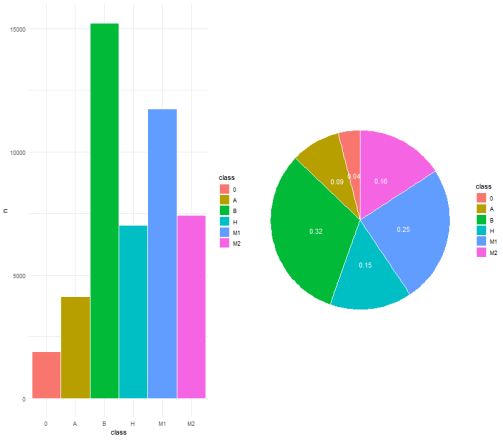
5.2 Affichage de l'ensemble des représentations graphiques

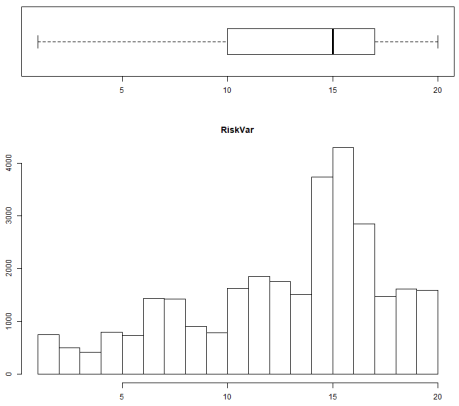
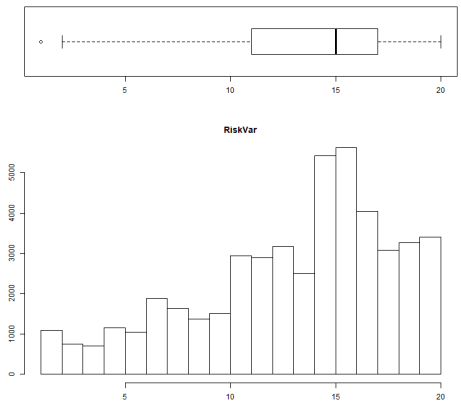
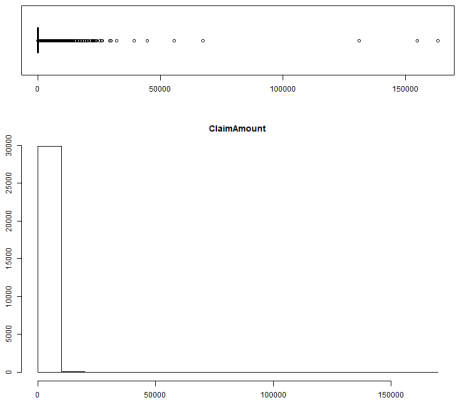
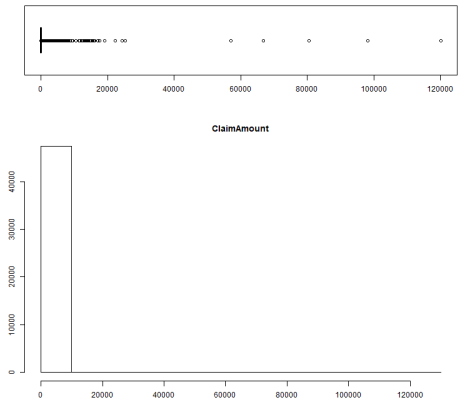
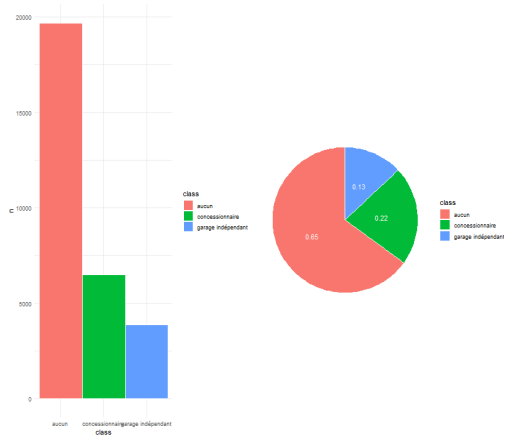
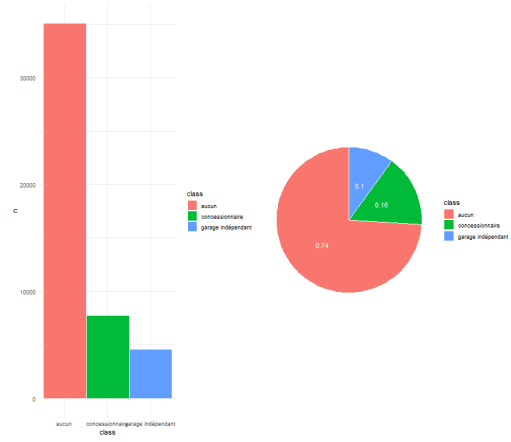
	freMPL1	freMPL2
Exposure		
LicAge		
VehAge		

	freMPL1	freMPL2
Gender		
MariStat		
SocioCateg		

	freMPL1	freMPL2
VehUsage		
DrivAge		
HasKmLimit		

	freMPL1	freMPL2
BonusMalus		
VehBody		
VehPrice		

	freMPL1	freMPL2
VehEngine		
VehEnergy		
VehMaxSpeed		

	freMPL1	freMPL2
VehClass		
RiskVar		
ClaimAmount		

	freMPL1	freMPL2
Garage