

Entreprise Telco :

Introduction

Le but de ce projet est de déterminer nos potentiels clients Churner. En effet, Telclo est une entreprise de télécommunication proposant un service téléphonique mais aussi d'autres services tels qu'un accès à internet, un service de sauvegarde à distance, une protection des appareils, un support, et autres services... Le but est de cibler nos clients futurs churners afin de potentiellement entamer des démarches afin de les reconquérir (offre promotionnelle,...) ou bien encore de déterminer les raisons de leurs départs afin d'établir peut-être une offre plus adaptée à certains clients. Il n'y a dans nos données pas de dimension temporelle. Nous avons pour ainsi dire pas de dates, on ne travaillera donc pas avec une dimension temporelle mais on aura à disposition l'ancienneté du contrat ("tenure") qui nous sera particulièrement utile dans l'exercice. Par commodité et pour s'adapter aux contraintes imposées, nous augmenterons nos NA afin de pouvoir les remplacer, de plus nous avons transformé certaines variables catégorielles binaires en variables numériques.

Voici la liste des variables mises à disposition:

Customer ID : identifiant du client.

Gender : Sexe du client (M/F).

SeniorCitizen: Client est-il senior (1) ou non (0) ?

Partner : Le client a-t'il un partenaire (Yes) ou non (No)?

Dependents: Le client a-t'il des personnes à charge ou non ? (Yes, No)

Tenure : Nombre de mois que le client a passé dans la compagnie

PhoneService: Le client a-t'il souscrit à un service téléphonique ? (Yes, No)

MultipleLines: Le client a-t'il plusieurs lignes ? (Yes, No, No phone service)

InternetService: A-t'il souscrit à un abonnement internet ? (DSL, Fiber optic, No)

OnlineSecurity: Le client a-t'il souscrit au service de sécurité en ligne (Yes, No, No internet service)

OnlineBackup: Le client a-t'il souscrit au service de sauvegarde en ligne (Yes, No, No internet service)

DeviceProtection: Le client a-t'il souscrit au service de protection de l'appareil(Yes, No, No internet service)

TechSupport : Le client a-t'il souscrit au service d'assistance ?(Yes, No, No internet service)

StreamingTV: Service de TV? (Yes, No, No internet service)

StreamingMovies: Service de film en streaming ?(Yes, No, No internet service)

Contract: Durée de vie du contrat (Month-to-month, One year, Two year)

PaperlessBilling: Facturation en papier (Yes, No)

PaymentMethod: Le type de paiement choisi (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))

MonthlyCharges: Le montant facturé mensuellement au client

TotalCharges: Le montant total facturé au client

Churn: Le client a-t'il quitté l'entreprise ? (Yes or No)

Paramétrage des données

Extraction de la table

```
base=read.csv("Telco_projet.csv",header=TRUE, sep=",",na.strings= "")  
head(base)
```

```

##  customerID gender SeniorCitizen Partner Dependents tenure PhoneService
## 1 7590-VHVEG Female          0 Yes     No      1       No
## 2 5575-GNVDE Male           0 No      No     34      Yes
## 3 3668-QPYBK Male           0 No      No      2      Yes
## 4 7795-CFOCW Male           0 No      No     45      No
## 5 9237-HQITU Female         0 No      No      2      Yes
## 6 9305-CDSKC Female         0 No      No      8      Yes
##   MultipleLines InternetService OnlineSecurity OnlineBackup DeviceProtection
## 1 No phone service          DSL      No     Yes      No
## 2 No                         DSL      Yes    No      Yes
## 3 No                         DSL      Yes    Yes      No
## 4 No phone service          DSL      Yes    No      Yes
## 5 No                         Fiber optic No    No      No
## 6 Yes                        Fiber optic No    No      Yes
##   TechSupport StreamingTV StreamingMovies Contract PaperlessBilling
## 1 No            No           No Month-to-month Yes
## 2 No            No           No One year        No
## 3 No            No           No Month-to-month Yes
## 4 Yes           No           No One year        No
## 5 No            No           No Month-to-month Yes
## 6 No            Yes          Yes Month-to-month Yes
##   PaymentMethod MonthlyCharges TotalCharges Churn
## 1 Electronic check        29.85     29.85   No
## 2 Mailed check           56.95    1889.50  No
## 3 Mailed check           53.85     108.15  Yes
## 4 Bank transfer (automatic) 42.30    1840.75  No
## 5 Electronic check        70.70     151.65 Yes
## 6 Electronic check        99.65    820.50  Yes

```

```
str(base)
```

```

## 'data.frame': 7043 obs. of 21 variables:
## $ customerID : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",...: 5376 3963 2565 5536 6512 6552 1003 477
## 1 5605 4535 ...
## $ gender      : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
## $ SeniorCitizen : int  0 0 0 0 0 0 0 0 0 ...
## $ Partner     : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2 1 ...
## $ Dependents  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 2 ...
## $ tenure      : int  1 34 2 45 2 8 22 10 28 62 ...
## $ PhoneService : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
## $ MultipleLines : Factor w/ 3 levels "No","No phone service",...: 2 1 1 2 1 3 3 2 3 1 ...
## $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",...: 1 1 1 1 2 2 2 1 2 1 ...
## $ OnlineSecurity : Factor w/ 3 levels "No","No internet service",...: 1 3 3 3 1 1 1 3 1 3 ...
## $ OnlineBackup   : Factor w/ 3 levels "No","No internet service",...: 3 1 3 1 1 1 3 1 1 3 ...
## $ DeviceProtection: Factor w/ 3 levels "No","No internet service",...: 1 3 1 3 1 3 1 1 3 1 ...
## $ TechSupport    : Factor w/ 3 levels "No","No internet service",...: 1 1 1 3 1 1 1 1 3 1 ...
## $ StreamingTV    : Factor w/ 3 levels "No","No internet service",...: 1 1 1 1 1 3 3 1 3 1 ...
## $ StreamingMovies : Factor w/ 3 levels "No","No internet service",...: 1 1 1 1 1 3 1 1 3 1 ...
## $ Contract       : Factor w/ 3 levels "Month-to-month",...: 1 2 1 2 1 1 1 1 2 ...
## $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
## $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",...: 3 4 4 1 3 3 2 4 3 1 ...
## $ MonthlyCharges : num  29.9 57 53.9 42.3 70.7 ...
## $ TotalCharges   : num  29.9 1889.5 108.2 1840.8 151.7 ...
## $ Churn          : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...

```

```
summary(base)
```

```

##      customerID      gender   SeniorCitizen   Partner   Dependents
## 0002-ORFBO: 1 Female:3488 Min. :0.0000 No :3641 No :4933
## 0003-MKNFE: 1 Male :3555 1st Qu.:0.0000 Yes:3402 Yes:2110
## 0004-TLHLJ: 1                   Median :0.0000
## 0011-IGKFF: 1                   Mean  :0.1621
## 0013-EXCHZ: 1                   3rd Qu.:0.0000
## 0013-MHZWF: 1                   Max.  :1.0000
## (Other) :7037
##      tenure   PhoneService   MultipleLines   InternetService
## Min.   : 0.00  No :682       No    :3390  DSL  :2421
## 1st Qu.: 9.00 Yes:6361     No phone service: 682 Fiber optic:3096
## Median :29.00                    Yes           :2971  No   :1526
## Mean   :32.37
## 3rd Qu.:55.00
## Max.  :72.00
##
##      OnlineSecurity   OnlineBackup
## No   :3498  No   :3088
## No internet service:1526  No internet service:1526
## Yes  :2019  Yes  :2429
##
##      DeviceProtection   TechSupport
## No   :3095  No   :3473
## No internet service:1526  No internet service:1526
## Yes  :2422  Yes  :2044
##
##      StreamingTV   StreamingMovies   Contract
## No   :2810  No   :2785 Month-to-month:3875
## No internet service:1526  No internet service:1526 One year   :1473
## Yes  :2707  Yes  :2732 Two year   :1695
##
##      PaperlessBilling   PaymentMethod   MonthlyCharges
## No :2872 Bank transfer (automatic):1544 Min.   : 18.25
## Yes:4171 Credit card (automatic) :1522 1st Qu.: 35.50
##                   Electronic check   :2365 Median  : 70.35
##                   Mailed check     :1612 Mean   : 64.76
##                   3rd Qu.: 89.85
##                   Max.   :118.75
##
##      TotalCharges   Churn
## Min.   : 18.8  No :5174
## 1st Qu.: 401.4 Yes:1869
## Median :1397.5
## Mean   :2283.3
## 3rd Qu.:3794.7
## Max.  :8684.8
## NA's   :11

```

Transformation variable categorielle en numerique

```
## [1] "Female" "Male"
```

```
## [1] "No"  "Yes"
```

```
## [1] "No"  "Yes"
```

```
## [1] "No"  "Yes"
```

Insertion des NA pour variables numériques

```

insert_nas <- function(x) {
  len <- length(x)
  n<- sample(1:floor(0.1*len),1)
  i <- sample(1:len,n)
  x[i] <- NA
  x
}

base[,c('TotalCharges','PaperlessBilling')] <- data.frame(sapply(base[,c('TotalCharges','PaperlessBilling')], insert_nas))

```

Insertion des NA pour variables qualitatives

```
base[,c('StreamingMovies','Contract')] <- data.frame(sapply(base[,c('StreamingMovies','Contract')], insert_nas))
```

Churner , Churner ! Qui-est-tu ?

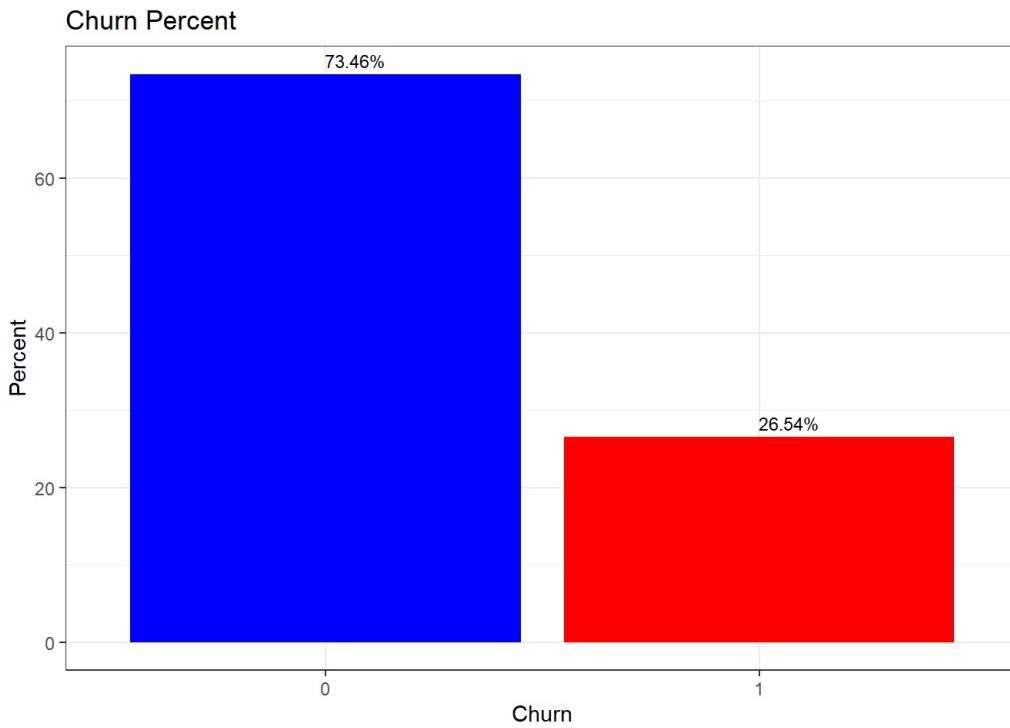
Analyse graphique univariée

Proportion de la variable cible

```

table <- base
options(repr.plot.width = 6, repr.plot.height = 4)
table %>%
  group_by(Churn) %>%
  summarise(Count = n())%>%
  mutate(percent = prop.table(Count)*100)%>%
  ggplot(aes(reorder(Churn, -percent), percent), fill = Churn)+
  geom_col(fill = c("blue", "red"))+
  geom_text(aes(label = sprintf("%.2f%%", percent)), hjust = 0.01,vjust = -0.5, size =3)+
  theme_bw()+
  xlab("Churn") +
  ylab("Percent")+
  ggtitle("Churn Percent")

```



On peut observer ici la répartition de Churner dans notre population. On voit ainsi qu'on possède plus d'un quart de churning contre trois quart de non churning. On peut donc ici appliquer nos algorithmes assez serainement car il n'y a pas d'oversampling (= trop petite représentation de la variable ciblée).

Représentation graphique des variables avec NA

```

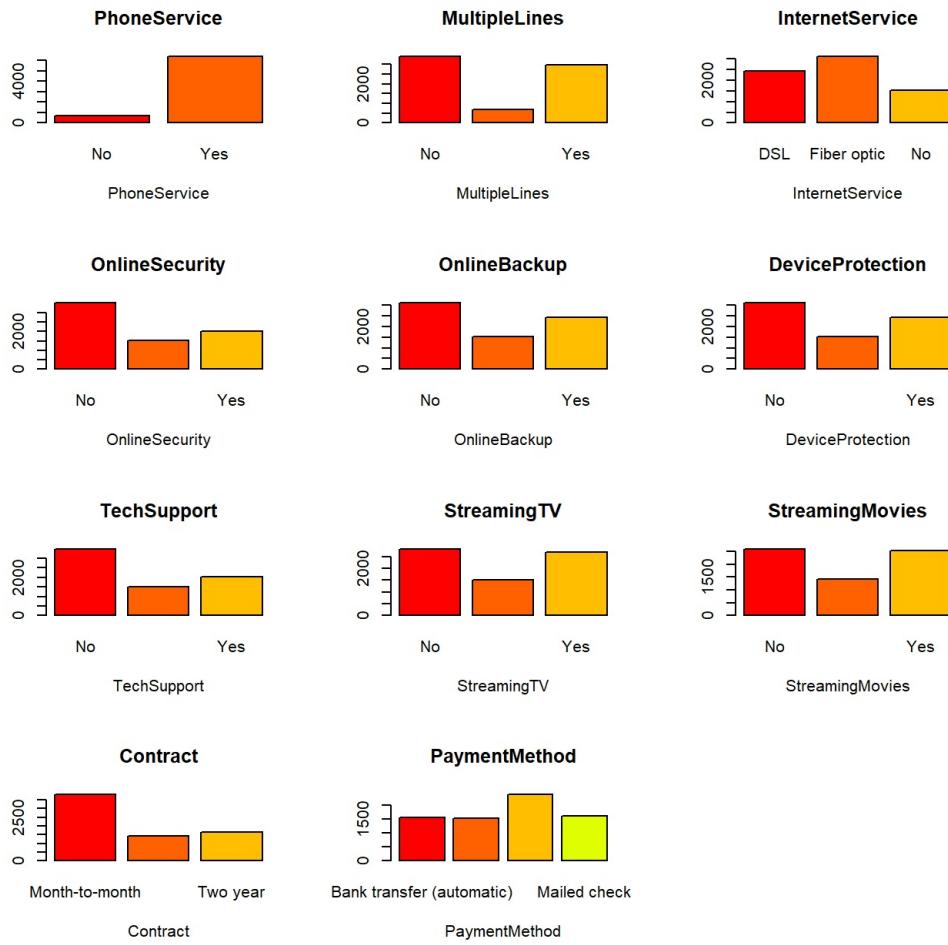
numerik <- Filter(is.numeric,table)
karacter <- Filter(is.factor,table)
karacter$'customerID' =NULL
#describe(karacter)

```

```

#Graph variable catégorielle
par(mfrow=c(3,3))
for(x in seq(1,length(karacter)))plot(karacter[,x], xlab=names(karacter[x]),
                                         col=rainbow(16),
                                         main=names(karacter[x]), horiz=F)

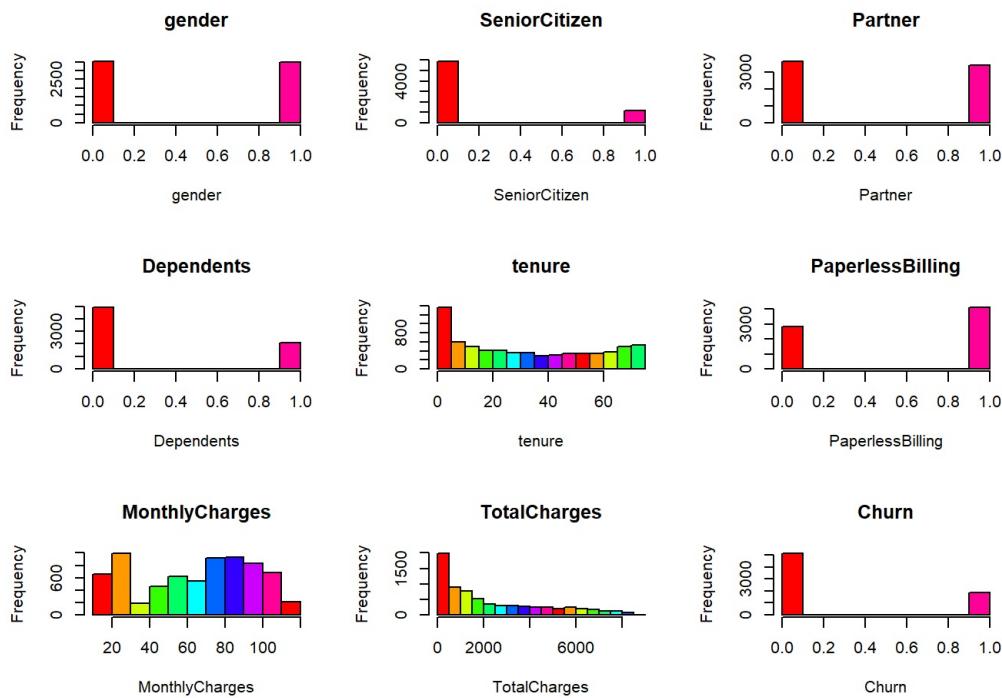
```



```

#Graph variable numérique
par(mfrow=c(3,3))#Pour mettre les 6 histos... sur le même graph
for(x in seq(1, length(numerik)))hist(numerik[,x], xlab=names(numerik[x]),
                                         col=rainbow(10), main=names(numerik[x]))

```



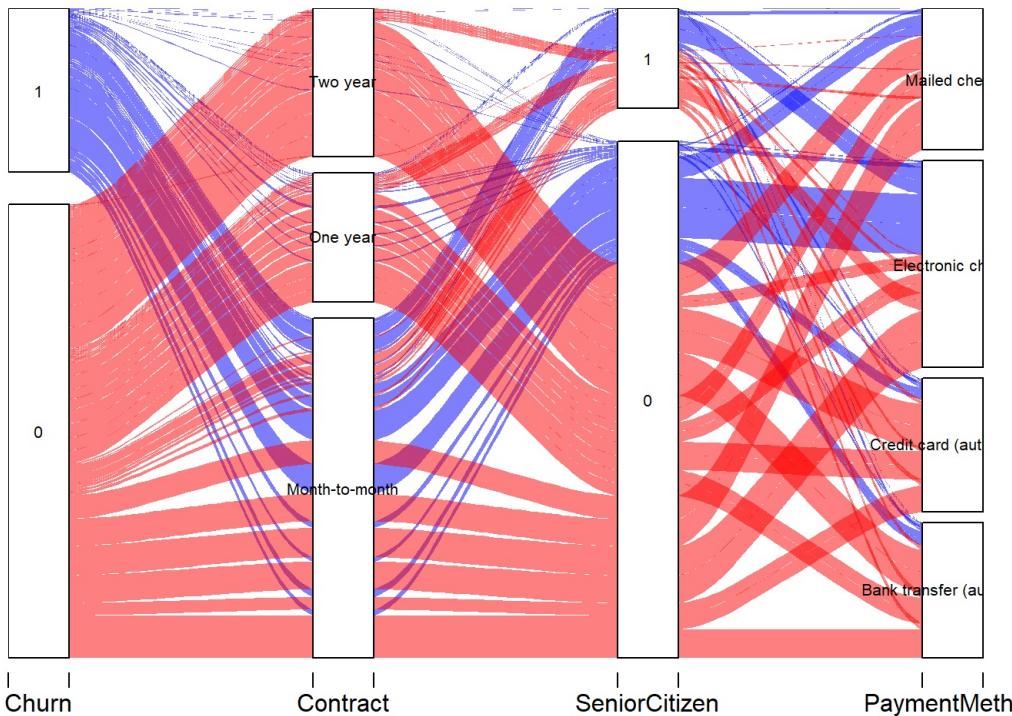
Analyse graphique multivariée

Diagramme alluvial

```
par(mfrow=c(1,1))
table2 <- table %>%
  group_by(Churn, Contract, SeniorCitizen, PaymentMethod, gender ) %>%
  summarise(N = n()) %>%
  ungroup %>%
  na.omit

## Warning: Factor `Contract` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

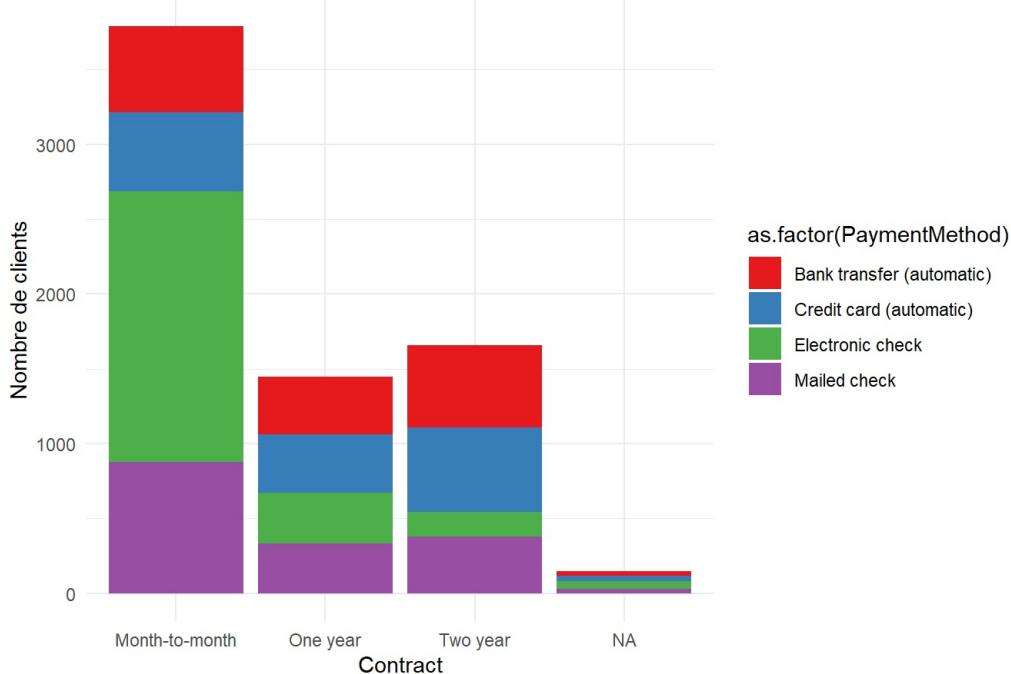
```
alluvial(table2[, c(1:4)],
  freq=table2$N, border=NA,
  col=ifelse(table2$Churn == "1", "blue", "red"),
  cex=0.65,
  ordering = list(
    order(table2$Churn, table2$SeniorCitizen=="Yes"),
    order(table2$gender, table2$SeniorCitizen=="Yes"),
    NULL,
    NULL))
```



Le graphique alluvial permet de voir certaines interactions ou profil type du caractère ciblé. On voit ici un chemin qui se dessine pour un Churner. Ici, nous pouvons voir qu'un Churner aura plus de chances d'avoir un contrat month-to-month, qu'il ne sera pas senior et qu'il paiera plutôt par 'electronic check'. Cependant, avec ce genre de graph, il faut s'assurer de l'indépendance de chacune des variables utilisées. En effet, on pourrait se demander si un Churn, qui est visiblement majoritairement en contrat mois/mois n'a pas pour seul choix le paiement par 'electronic check'. Il faudra donc vérifier cela afin de pouvoir conjecturer sur une potentielle causalité.

```
ggplot(table) +
  geom_bar(aes(x=Contract, fill=as.factor(PaymentMethod))) +
  ylab("Nombre de clients") +
  ggtitle("Repartition du mode de paiement en fonction du type de contrat") +
  scale_fill_brewer(palette="Set1") +
  theme_minimal()
```

Repartition du mode de paiement en fonction du type de contrat



Ce graph valide la tendance des churners vu dans le graph alluvial, en effet , on voit que le contrat month/month a toutes les possibilités de paiement offertes. Il n'y a donc pas de dépendance entre ces variables. Nous pouvons donc valider le profil du Churner vu au dessus.

Représentation de la part de Churn en fonction de la durée du contrat

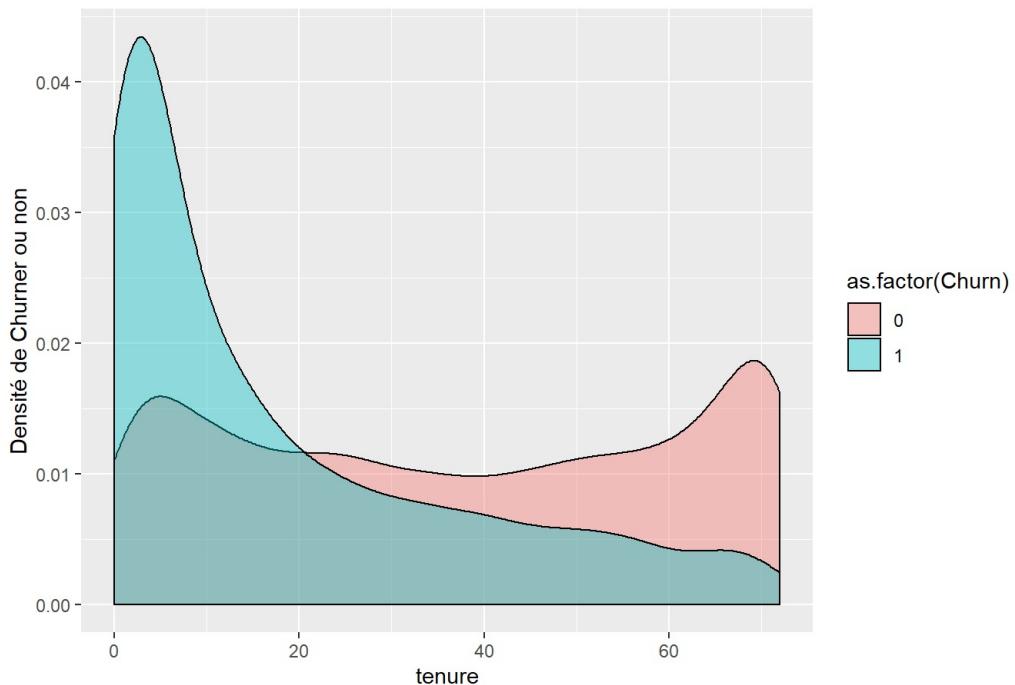
```

ggplot(table) +
  geom_density(aes(x=tenure, fill=as.factor(Churn)),alpha=0.4)+

  ggtitle("Densité de Churn en fonction de la durée du contrat ")+
  ylab("Densité de Churn ou non")

```

Densité de Churn en fonction de la durée du contrat



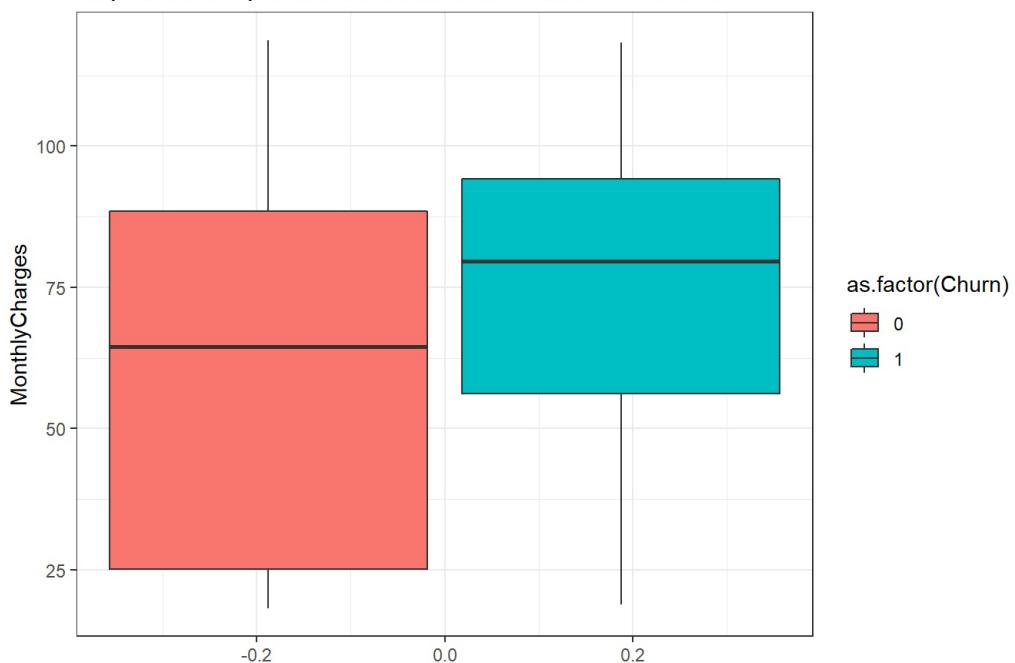
Représentation des sommes payées en fonction du Churn

```

ggplot(table, aes(y= MonthlyCharges,  fill = as.factor(Churn))) +
  geom_boxplot()+
  theme_bw()+
  ggtitle("Répartition des paiements au mois en fonction du Churn")+
  xlab(" ")

```

Répartition des paiements au mois en fonction du Churn



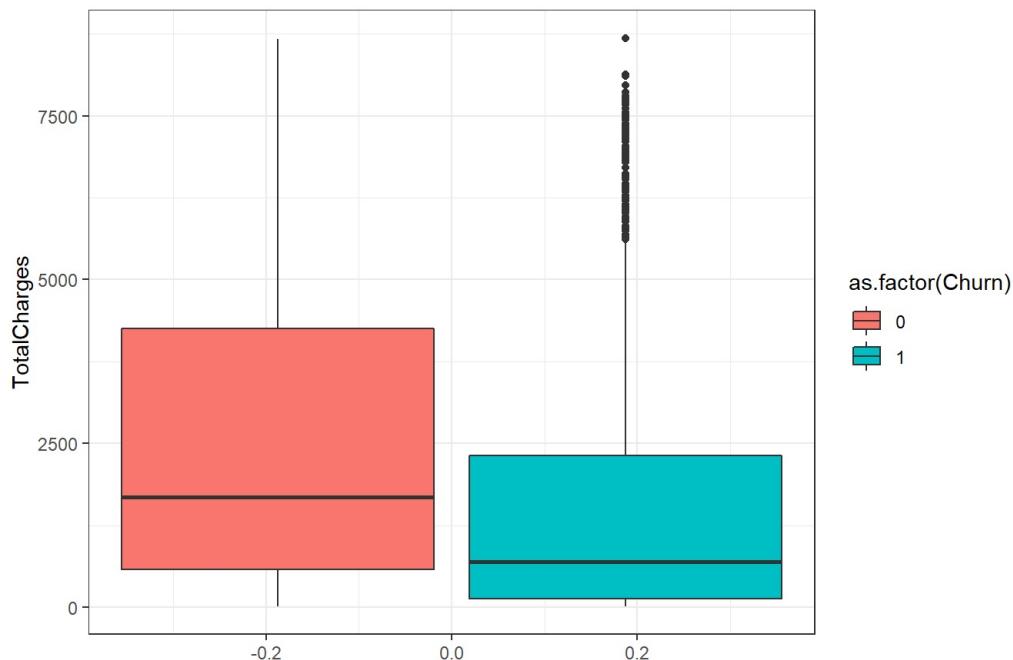
```

ggplot(table, aes(y= TotalCharges,  fill = as.factor(Churn))) +
  geom_boxplot()+
  theme_bw()+
  ggtitle("Répartition des paiements cumulés en fonction du Churn")+
  xlab(" ")

```

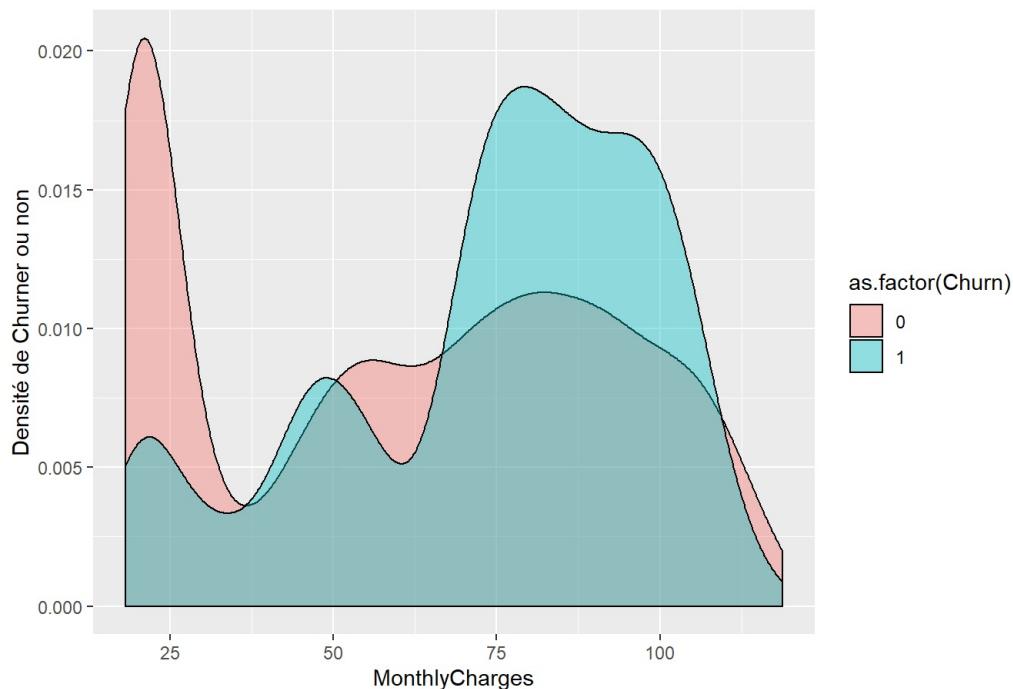
```
## Warning: Removed 53 rows containing non-finite values (stat_boxplot).
```

Répartition des paiements cumulés en fonction du Churn

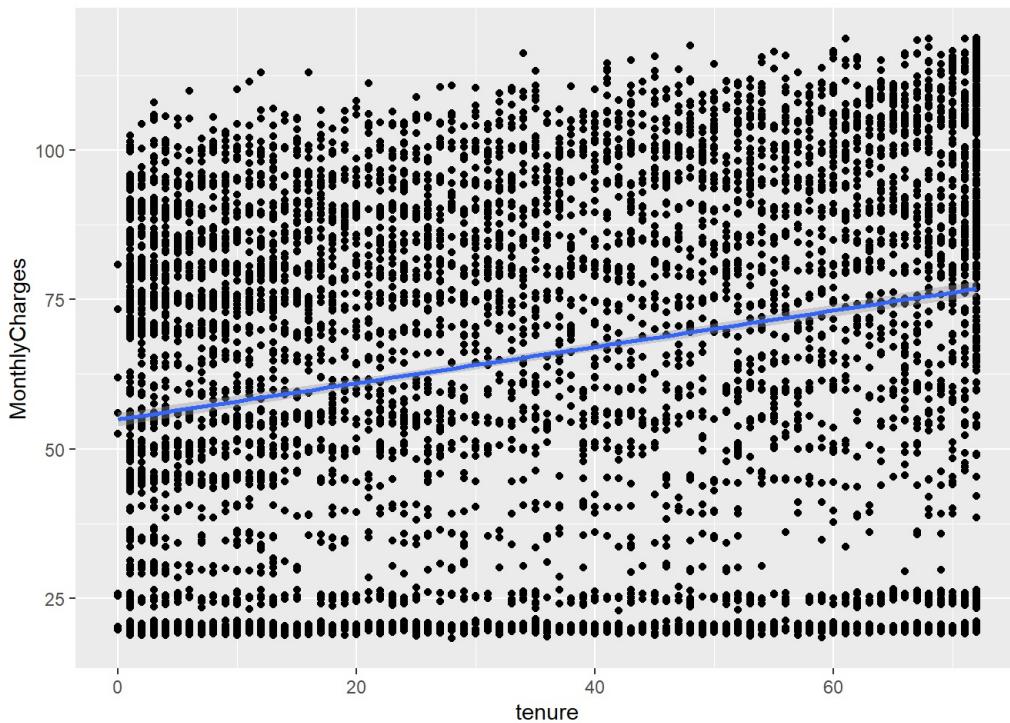


```
ggplot(table) +  
  geom_density(aes(x=MonthlyCharges, fill=as.factor(Churn)),alpha=0.4)+  
  
  ggtitle("Densité de Churn en fonction du montant payé au mois ")+  
  ylab("Densité de Churner ou non")
```

Densité de Churn en fonction du montant payé au mois



```
ggplot(data = table, aes(x = tenure, y =MonthlyCharges )) + geom_point() + geom_smooth(method = "lm")
```



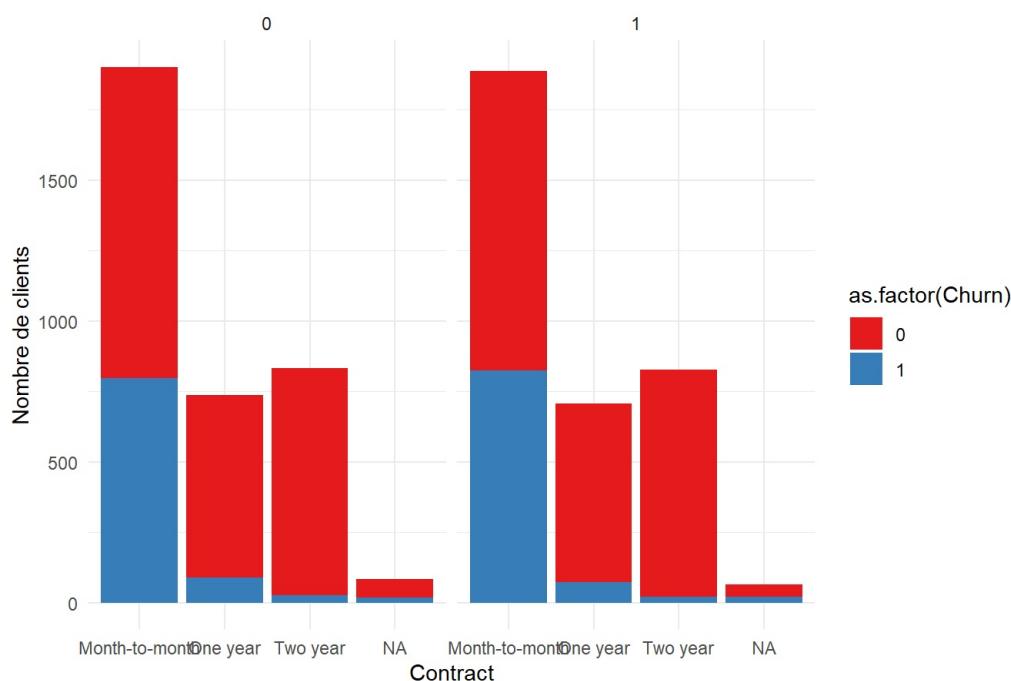
Nous avons ici toute une série de graphique mettant en relation le Churn avec les charges régulières. D'une part, les boxplots nous montrent l'absence d'ouliers. Même si certaines valeurs dans 'TotalCharges' sont importantes, celles-ci sont expliquées par une forte ancienneté conjugué à un abonnement mensuel élevé. Ces valeurs ne sont donc pas aberrantes et ne nécessitent donc pas de retraitement. Sur le graph de densité, nous pouvons voir que les churners payent un abonnement mensuel plus important que les non-churners ce qui pourrait être une raison de Churn. Enfin sur le 4ème graph, on peut voir un fait plutôt étonnant, le prix payé mensuellement ne diminue pas avec l'ancienneté du contrat. Cela pourrait s'expliquer par le fait que les clients possédant un abonnement à option (qui payent donc plus cher), ont plus de chances d'avoir une longue ancienneté.

Représentation du Churn en fonction du type de contrat par sexe

```
ggplot(table) +
  geom_bar(aes(x=Contract, fill=as.factor(Churn)))+
  facet_wrap(~gender, ncol = 2, nrow = 1)+

  ylab("Nombre de clients") +
  ggtitle("Part de Churn H/F par type de contrat") +
  scale_fill_brewer(palette="Set1") +
  theme_minimal()
```

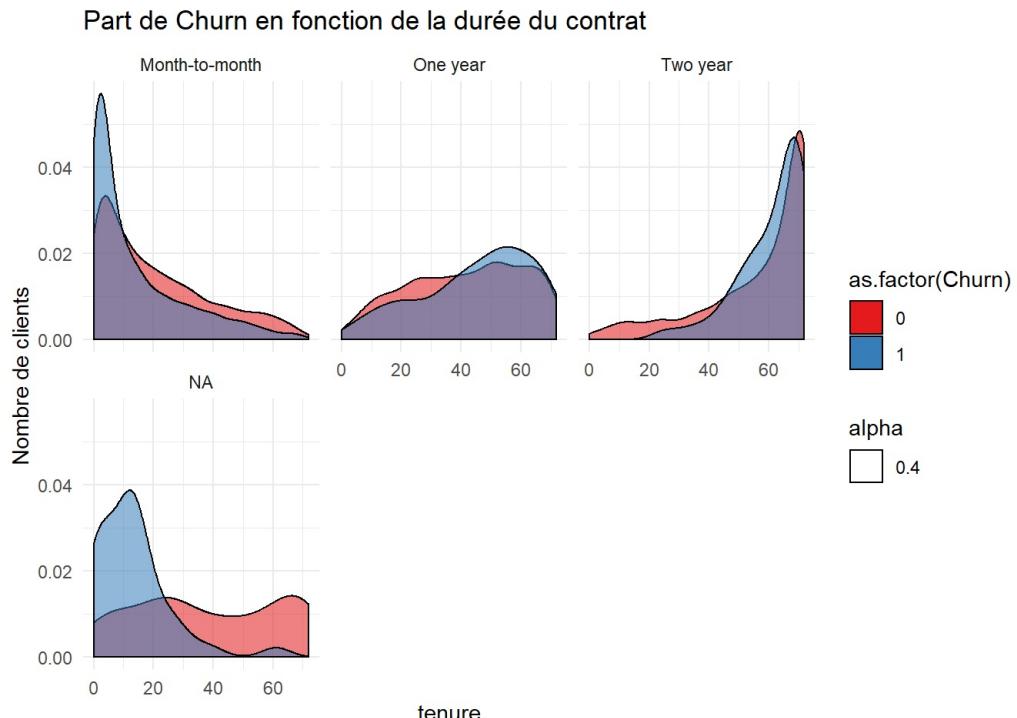
Part de Churn H/F par type de contrat



Nous pouvons voir ici que le type de contrat a une importance déterminante dans le Churn. En effet, l'offre de contrat mois/mois entraîne un fort Churn ce qui tend à rejoindre l'analyse du premier graph alluvial. Cependant cette analyse est à modérée, car cette offre est peut-être destinée à des clients considérés comme fragile qui ne seraient peut être pas client chez Telco avec d'autres offres de contrat long terme.

Représentation du Churn en densité en fonction de la durée du contrat

```
ggplot(table) +
  geom_density(aes(x=tenure, fill=as.factor(Churn), alpha=0.4)) +
  facet_wrap(~Contract, ncol = 3) +
  ylab("Nombre de clients") +
  ggtitle("Part de Churn en fonction de la durée du contrat") +
  scale_fill_brewer(palette="Set1") +
  theme_minimal()
```

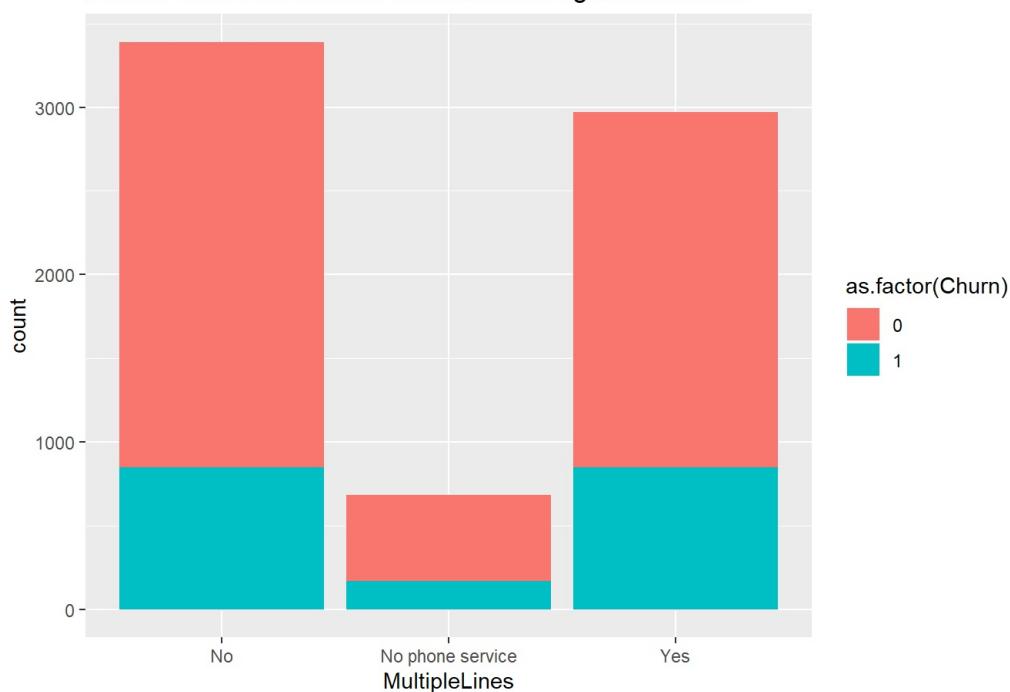


Dans la logique des choses, nous pouvons voir que la population churn en fonction de l'ancienneté du contrat pour les contrat long terme (1 an & 2 ans) cependant dans le cas du contrat "month to month", nous voyons un pic de Churn dès les premiers jours du début du contrat. C'est une situation qui se remarque souvent dans les études de Churn, en effet, le client peut parfois changer d'avis dans les premiers jours de la signature du contrat et se rétracter (avec remboursement ou non). Cependant, cela peut-être aussi due à une mauvaise segmentation de notre offre et de client ayant besoin d'un abonnement téléphonique inférieur à un mois. Peut-être qu'avec une offre de contrat d'une semaine ou journalière, on pourrait capter plus de client qui ne veulent pas aujourd'hui payer un mois inutilement.

Quelle est l'importance de la détention de plusieurs lignes

```
ggplot(table) +
  geom_bar(aes(x=MultipleLines, fill=as.factor(Churn)))+
  ggtitle("Part de Churn en fonction du nombre de lignes détenues")
```

Part de Churn en fonction du nombre de lignes détenues



Sur ce graphique, nous constatons qu'il n'y a pas de relation déterminante entre le fait de posséder un abonnement téléphonique (simple ou multilignes) dans le churn

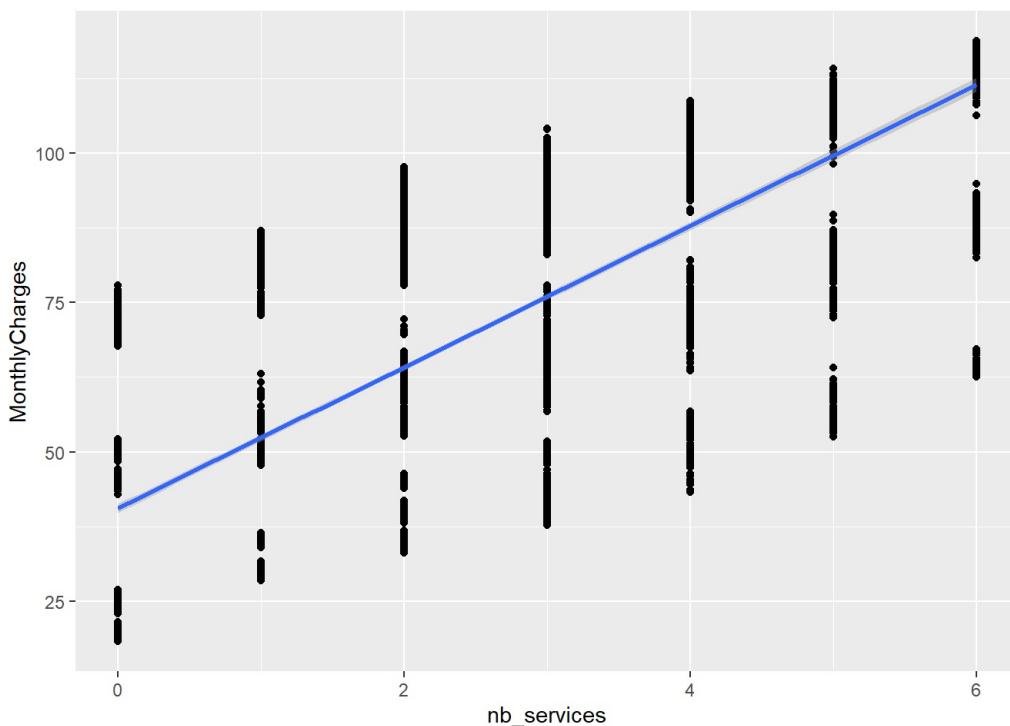
Quelle est l'importance du Churn en fonction du nombre de service souscrit ?

```
services <- table
services$OnlineSecurity <- ifelse(services$OnlineSecurity == 'Yes', 1, 0)
services$OnlineBackup <- ifelse(services$OnlineBackup == 'No' | services$OnlineBackup == 'No internet service', 0, 1)
services$DeviceProtection <- ifelse(services$DeviceProtection == 'No' | services$DeviceProtection == 'No internet service', 0, 1)
services$TechSupport <- ifelse(services$TechSupport == 'No' | services$TechSupport == 'No internet service', 0, 1)
services$StreamingTV <- ifelse(services$StreamingTV == 'No' | services$StreamingTV == 'No internet service', 0, 1)
services$StreamingMovies <- ifelse(services$StreamingMovies == 'No' | services$StreamingMovies == 'No internet service', 0, 1)
#services$InternetService <- ifelse(services$InternetService == 'No', 0, 1)
services$nb_services = services$OnlineSecurity + services$OnlineBackup + services$DeviceProtection + services$TechSupport + services$StreamingTV + services$StreamingMovies
table$nb_services <- services$nb_services

ggplot(data = table, aes(x = nb_services, y = MonthlyCharges)) + geom_point() + geom_smooth(method = "lm")

## Warning: Removed 535 rows containing non-finite values (stat_smooth).

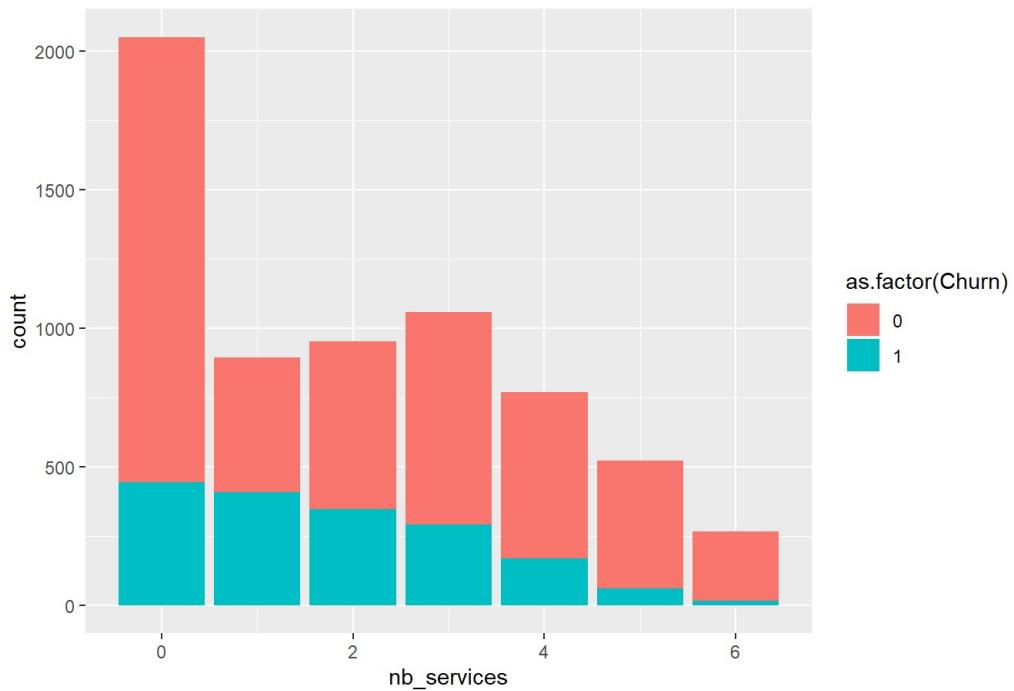
## Warning: Removed 535 rows containing missing values (geom_point).
```



```
ggplot(services) +
  geom_bar(aes(x=nb_services, fill=as.factor(Churn)))+
  ggtitle("Part de Churn en fonction du nombre de services souscrits")
```

```
## Warning: Removed 535 rows containing non-finite values (stat_count).
```

Part de Churn en fonction du nombre de services souscrits

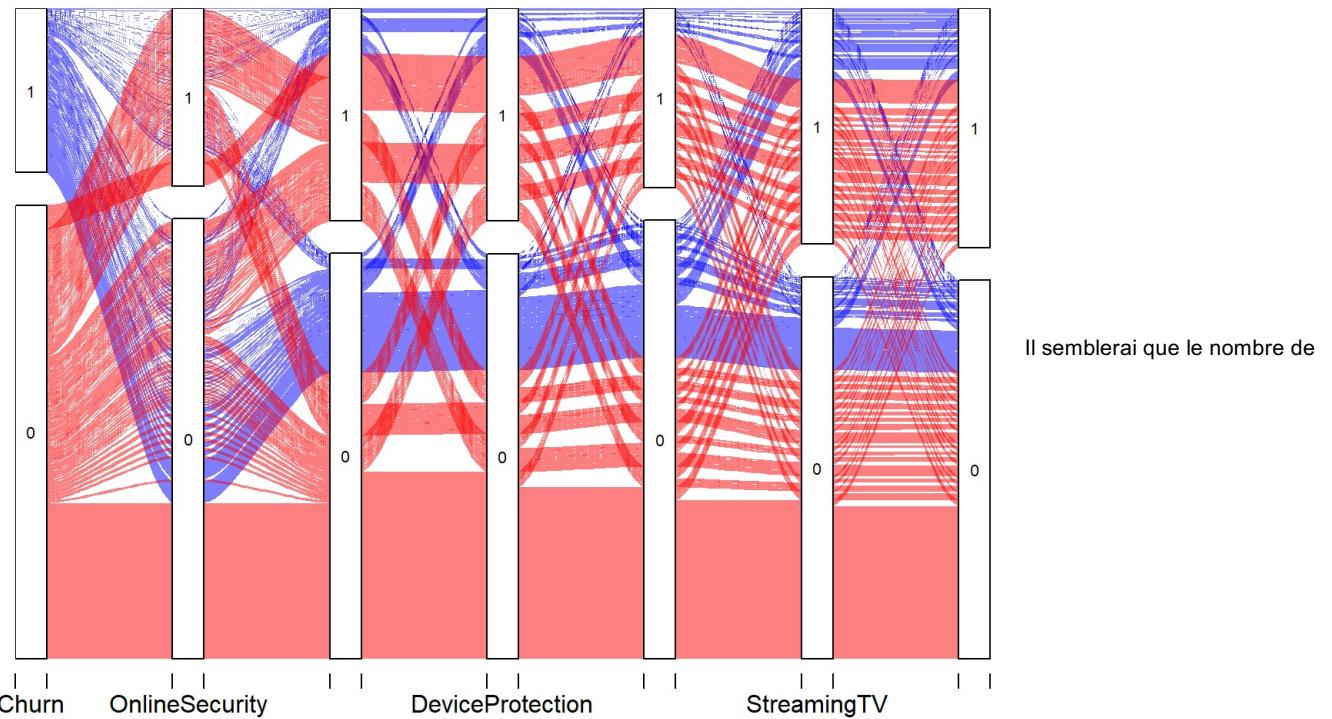


```

table3 <- services %>%
  group_by(Churn, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, gender)
) %>%
  summarise(N = n()) %>%
  ungroup %>%
  na.omit

alluvial(table3[, c(1:7)],
  freq=table3$N, border=NA,
  col=ifelse(table3$Churn == "1", "blue", "red"),
  cex=0.65,
  ordering = list(
    order(table3$Churn, table3$OnlineSecurity==1),
    order(table3$gender, table3$OnlineSecurity=="Yes"),
    NULL,
    NULL,
    NULL,
    NULL,
    NULL))

```



Remplacement des valeurs manquantes

la typologie de valeurs manquantes dans le cas de notre dataset est MCAR, missing completely at random. Nous avons introduit 10% de NA sur 4 variables qualitatives et quatre variables quantitatives de manière indépendante et aléatoire, chaque Na présent dans une variable n'a pas d'impact sur les autres Na présents dans chaque autre variable

```
sum(is.na(table))
```

```
## [1] 1331
```

Nous avons ici 1289 valeurs manquantes à traiter.

On décide de remplacer les valeurs manquantes par la méthode KNN qui nous permet ici la plus adéquat à la vue du peu d'observation dont nous disposons. Le remplacement par 0 aurait été inadéquat car on aurait eu des sommes payées égales à 0. De même, le remplacement par la moyenne ou la médiane aurait pu être aussi judicieux mais moins intéressant que les kNN qui suit

```
set.seed(496)
#NOMBRE DE NA
navarspl=colSums(is.na(table))/nrow(table) #valeur manquante
#navarspl
tab.kNN=kNN(table[,-21], k=10, imp_var=F)#peut etre long k=5 est le nombre de voisin qu il va regarder pour calculer la dist
#nume.kNN
summary(tab.kNN)
```



```
##      customerID      gender   SeniorCitizen      Partner
## 0002-ORFBO: 1  Min.   :0.0000  Min.   :0.0000  Min.   :0.000
## 0003-MKNFE: 1  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.000
## 0004-TLHLJ: 1  Median :0.0000  Median :0.0000  Median :0.000
## 0011-IGKFF: 1  Mean    :0.4952  Mean    :0.1621  Mean    :0.483
## 0013-EXCHZ: 1  3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:1.000
## 0013-MHZWF: 1  Max.   :1.0000  Max.   :1.0000  Max.   :1.000
## (Other)   :7037
##      Dependents      tenure PhoneService      MultipleLines
##  Min.   :0.0000  Min.   : 0.00  No   : 682  No       :3390
##  1st Qu.:0.0000  1st Qu.: 9.00  Yes  :6361  No phone service: 682
##  Median :0.0000  Median :29.00        Yes   :2971
##  Mean   :0.2996  Mean   :32.37
##  3rd Qu.:1.0000  3rd Qu.:55.00
##  Max.   :1.0000  Max.   :72.00
##
##      InternetService      OnlineSecurity      OnlineBackup
##  DSL      :2421  No           :3498  No       :3088
##  Fiber optic:3096  No internet service:1526  No internet service:1526
##  No       :1526  Yes          :2019  Yes       :2429
##
##      DeviceProtection      TechSupport
##  No       :3095  No           :3473
##  No internet service:1526  No internet service:1526
##  Yes      :2422  Yes          :2044
##
##      StreamingTV      StreamingMovies      Contract
##  No       :2810  No           :2680  Month-to-month:3893
##  No internet service:1526  No internet service:1512  One year     :1459
##  Yes      :2707  Yes          :2851  Two year     :1691
##
##      PaperlessBilling      PaymentMethod  MonthlyCharges
##  Min.   :0.0000  Bank transfer (automatic):1544  Min.   : 18.25
##  1st Qu.:0.0000  Credit card (automatic)  :1522  1st Qu.: 35.50
##  Median :1.0000  Electronic check       :2365  Median : 70.35
##  Mean   :0.5934  Mailed check        :1612  Mean   : 64.76
##  3rd Qu.:1.0000                    :          3rd Qu.: 89.85
##  Max.   :1.0000                    :          Max.   :118.75
##
##      TotalCharges      nb_services
##  Min.   : 18.8  Min.   :0.000
##  1st Qu.: 401.4 1st Qu.:0.000
##  Median :1396.9  Median :2.000
##  Mean   :2281.3  Mean   :2.036
##  3rd Qu.:3783.2  3rd Qu.:3.000
##  Max.   :8684.8  Max.   :6.000
##
```

```
tab.kNN$Churn <- table$Churn
```

N.B : Etant donné l'aspect prédictif de l'algorithme kNN nous avons enlever la variables Churn pour éviter de biaiser notre modèle.

```

nume.kNN <- Filter(is.numeric,tab.kNN)
kar.kNN <- Filter(is.factor,tab.kNN)
kar.kNN$'customerID' =NULL
#describe(karacter)

```

Etude des interactions de variables

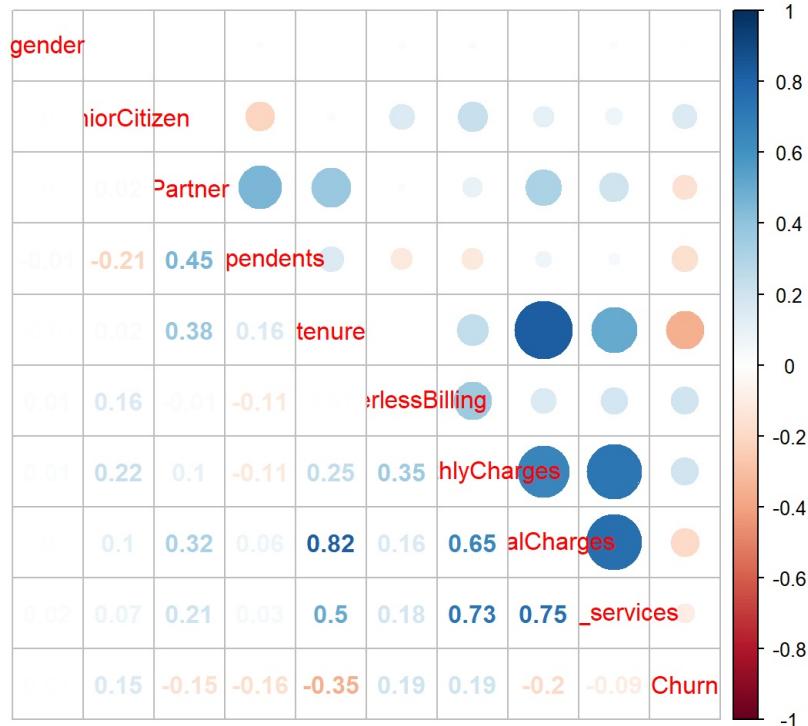
Variables numériques

Etude de corrélation

```

#Analyse corrélation variable numériques
par(mfrow=c(1,1))
M <- cor(nume.kNN)
corrplot.mixed(M)

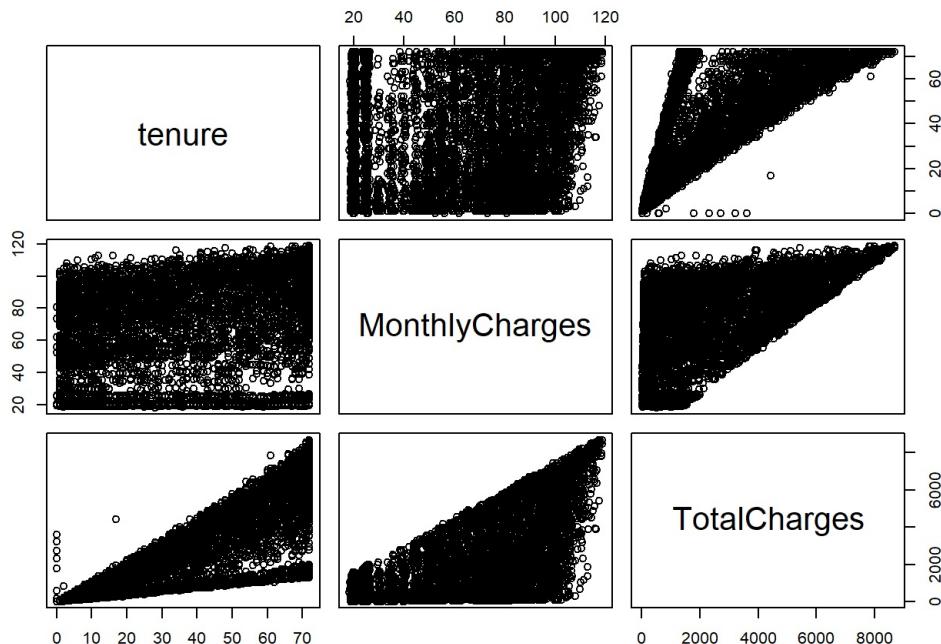
```



```

pairs(nume.kNN[, c("tenure", "MonthlyCharges", "TotalCharges")])

```



Nous étudions ici les interactions entre les variables numériques. Nous pouvons voir sur notre matrice de corrélation, de forts lien entre certaines variables. Nous pouvons citer notamment les corrélations entre TotalCharges et tenure , TotalCharges et MonthlyCharges. Nous pouvons dès lors soupçonner la présence de colinéarité entre ces variables afin de connaître la force du lien, pour savoir si leur lien peut-être négatif dans notre modèle.

Sur le graphique "pairs", nous avons mis en relation nos variables corrélées afin de connaître leurs répartitions. Nous pouvons y voir l'effet croissant de l'ancienneté sur le montant total de charge ce qui est logique mais nous pouvons voir aussi l'effet croissant des charges mensuelles sur le montant total de charges, ce qui n'est pas intéressant il s'agit d'une même information.

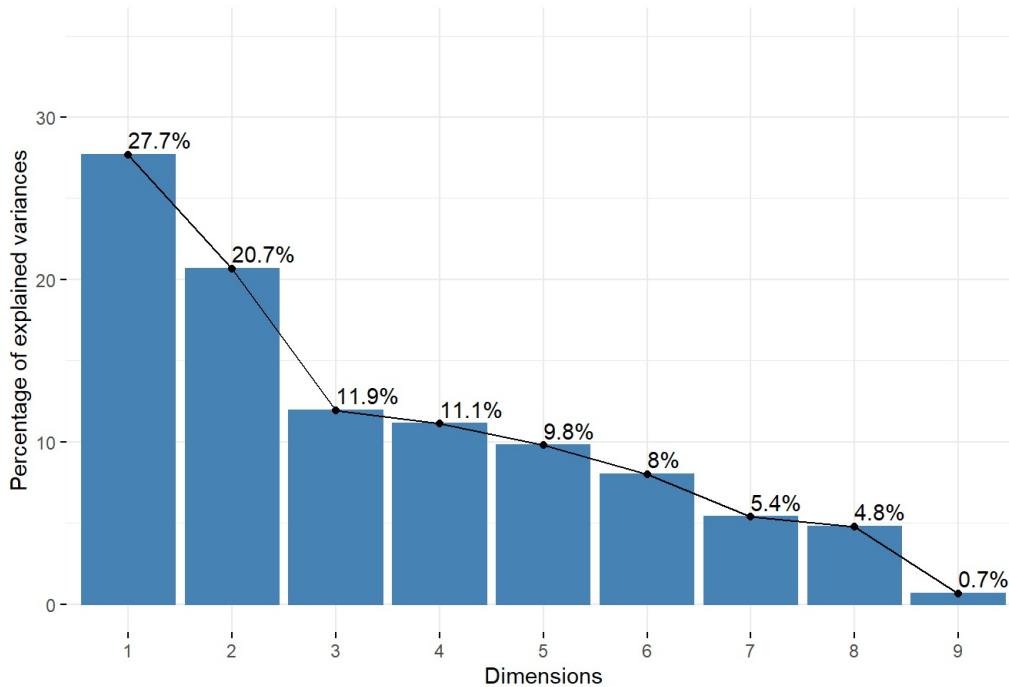
Analyse Composantes principales (ACP)

```
res.pca <- PCA(nume.kNN, quanti.sup=c(9), graph = FALSE)
#Valeurs propres et axes
eig.val <- get_eigenvalue(res.pca)
eig.val
```

	eigenvalue	variance.percent	cumulative.variance.percent
## Dim.1	2.49191602	27.6879558	27.68796
## Dim.2	1.86072148	20.6746831	48.36264
## Dim.3	1.07457121	11.9396802	60.30232
## Dim.4	0.99952243	11.1058048	71.40812
## Dim.5	0.87959974	9.7733304	81.18145
## Dim.6	0.72142326	8.0158140	89.19727
## Dim.7	0.48337084	5.3707871	94.56806
## Dim.8	0.42934105	4.7704561	99.33851
## Dim.9	0.05953396	0.6614885	100.00000

```
#Graphique
fviz_eig(res.pca, addlabels = TRUE, ylim = c(0, 35))
```

Scree plot

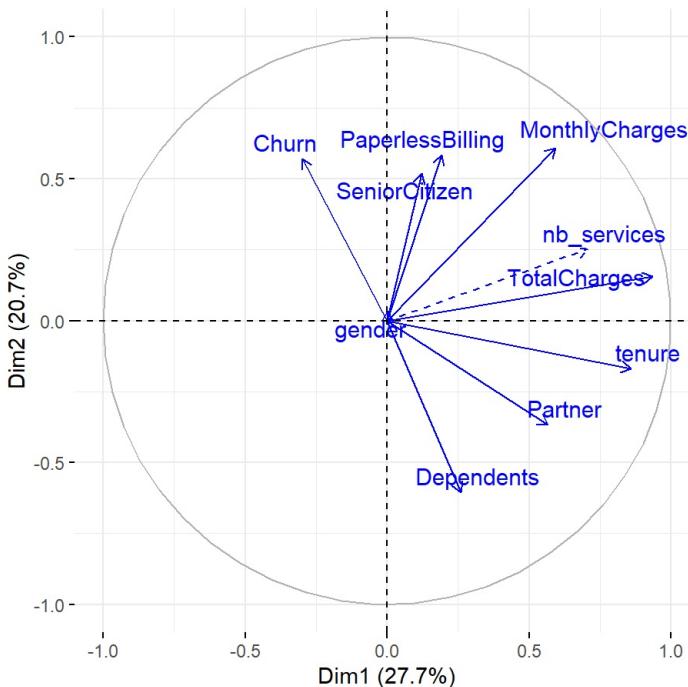


```
#Extraction des résultats (variables)
var <- get_pca_var(res.pca)
var
```

```
## Principal Component Analysis Results for variables
## =====
##   Name      Description
## 1 "$coord"  "Coordinates for the variables"
## 2 "$cor"    "Correlations between variables and dimensions"
## 3 "$cos2"   "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

```
#Cercle de correlation
fviz_pca_var(res.pca, col.var ="blue", repel = TRUE)
```

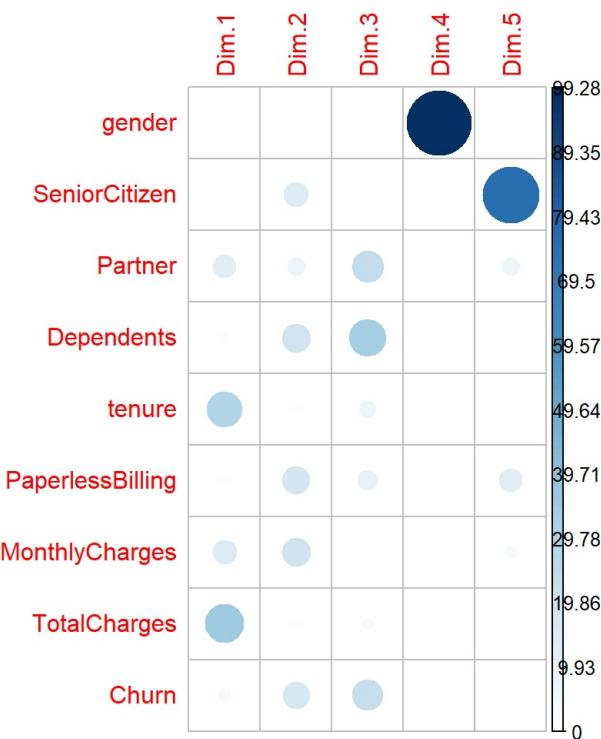
Variables - PCA



```
#Contribution des variables
var$contrib
```

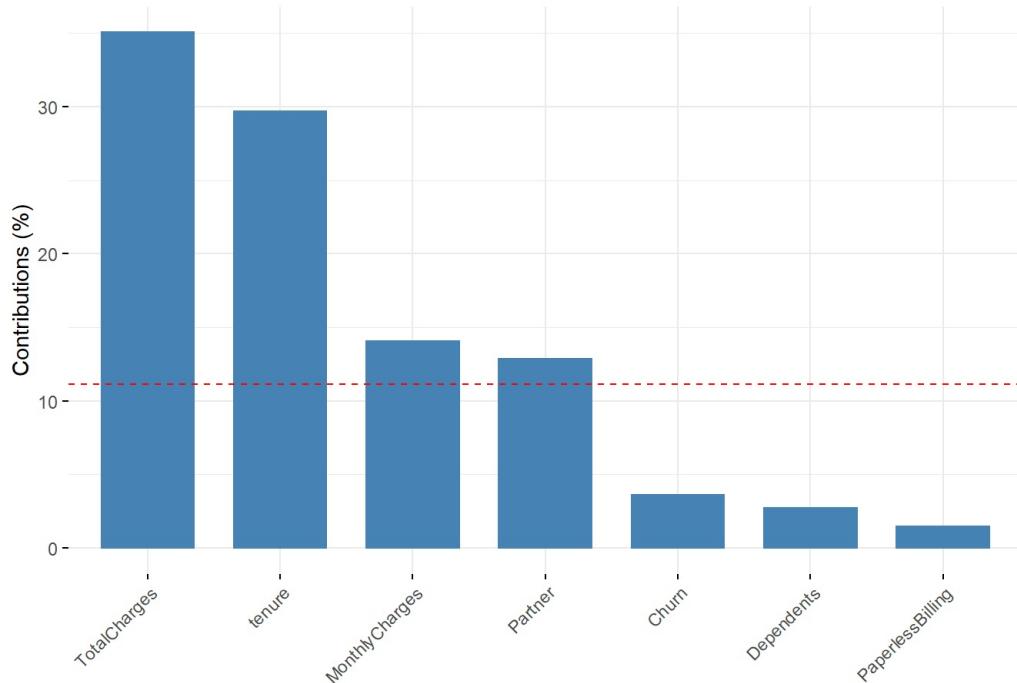
```
##           Dim.1    Dim.2    Dim.3    Dim.4    Dim.5
## gender      0.000151801  0.05838906  0.4510149 99.282957457  0.18115848
## SeniorCitizen 0.581863077 14.45302915  0.0283593  0.257454079 74.94371092
## Partner     12.849870196  7.18400877 23.8629911  0.078724067 7.26987470
## Dependents   2.718679787 19.65576420 32.9444771  0.125792994  0.08439997
## tenure       29.653262352  1.53502152  6.5981806  0.021817618  0.01015740
## PaperlessBilling 1.480808911 18.32375300  9.3082933  0.027145859 12.69744876
## MonthlyCharges 14.067634129 19.93063320  0.6076679  0.001372235 3.56192319
## TotalCharges  35.047790459  1.30322755  3.2488964  0.008622150  0.99175051
## Churn        3.599939288 17.55617356 22.9501194  0.196113541  0.25957607
```

```
corrplot(var$contrib, is.corr=FALSE)
```



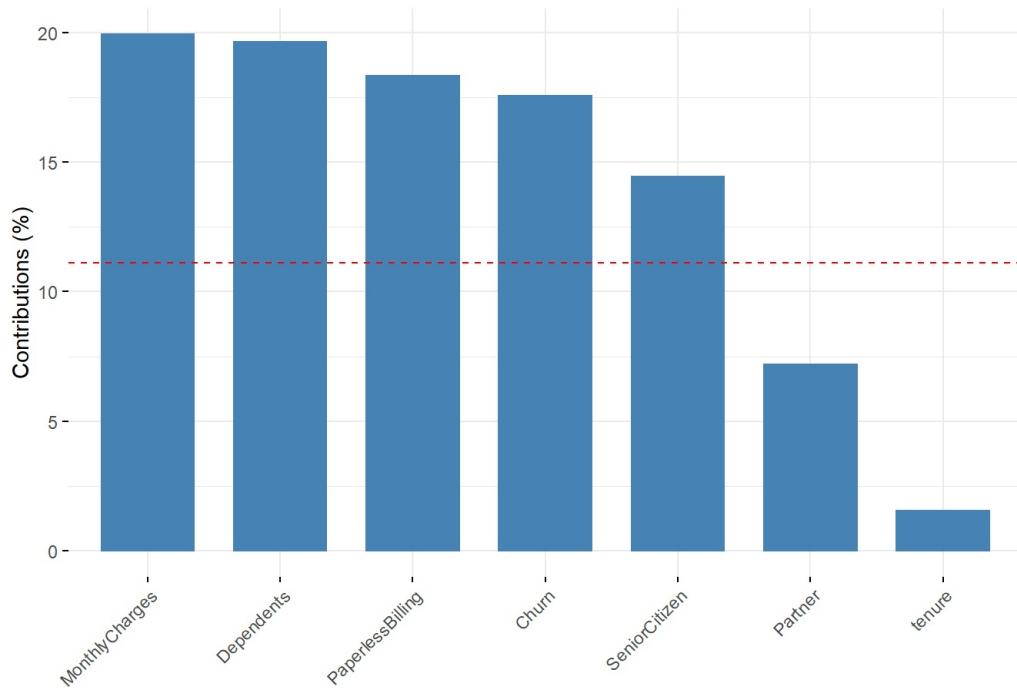
```
fviz_contrib(res.pca, choice = "var", axes = 1, top = 7) #histo par axe
```

Contribution of variables to Dim-1



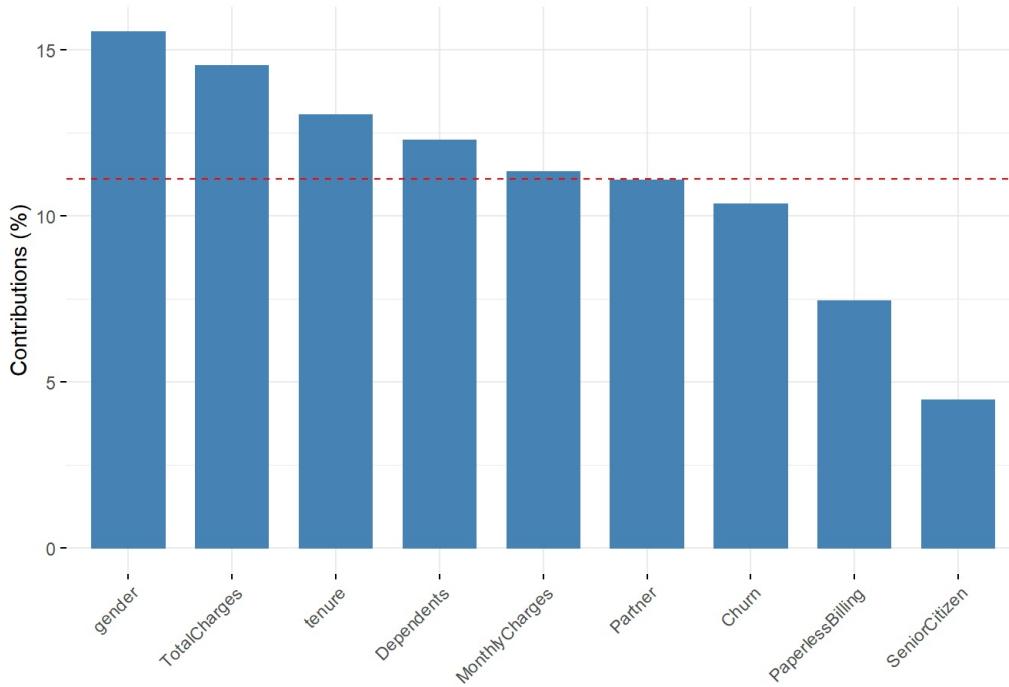
```
fviz_contrib(res.pca, choice = "var", axes = 2, top = 7)
```

Contribution of variables to Dim-2



```
fviz_contrib(res.pca, choice = "var", axes = 1:4, top = 10) #les 3 axes cumul?s
```

Contribution of variables to Dim-1-2-3-4



Scree plot: Nous pouvons observer sur ce graphiques le pourcentage de variance expliquée à l'aide des valeurs propres, pour chaque dimensions de notre ACP. On peut apercevoir que les deux premières dimensions ont une bonne variance cumulée puisqu'elle représentent plus de 51% de variance expliquée à elles seules. La coudée se situe au niveau de la troisième dimension avec un total de variance cumulée à 64.03%, les 5 premières dimensions se situent au dessus de 10%, pour un total de 86.7%.

Cercle de corrélation : Nous avons ici la projection des variables quantitatives de notre dataset sur le cercle de corrélations de notre ACP, avec les deux premières variables ayant une variance expliquée des plus élevées (30.6% pour la 1e et 20.9% pour le 2nde). Plus une variable est proche de l'axe d'une dimension, et proche du cercle de corrélation , plus celle-ci explique l'axe concerné. En l'occurrence, la variables 'TotalCharges' est la plus contributrice de la 1e dimension positivement , car proche du cercle de corrélation et de l'axe de la 1e dimension. De même la variable 'tenure' est la seconde variable la plus contributrice de cette dimension. A l'inverse la variable 'genre' étant très proche de 0, ne contribue ni à la première ni à la seconde dimension. La variable 'MonthlyCharges' contribue presque à 75% à la 1e dimension et 50% à la seconde. La variables 'Dependents' est celle qui contribue le plus négativement à la seconde dimension (à 75%). La variable 'Partner' contribue positivement à 50% à la Dim1 et -50% à la dim2. Les variables SeniorCitizen et 'PaperLessBilling' semble évoluer de la même manière car elles sont proches l'une de l'autre, contribuant toutes deux environ positivement à 50% à la 2nde dimension. La variable cible 'Churn' est plutôt proche de l'origine il contribue très peu négativement à la première dimension et à environ 25% à la 2nde dimension.

Graph de Corrélation des variables quantitatives aux dimension de l'ACP: La matrice de corrélation confirme ce que nous avons constaté à savoir que les variables contribuant le plus à la 1e dimension sont TotalCharges et Tenure. De même pour la seconde dimension avec la variable Dependents. On constate que le genre est la variable qui contribue presque à 100% à la 3e dimension. Les variables Paperlessbilling, Dependents, Partner, Seniorcitizen et tenure contribue à la 4e dimension. Enfin SeniorCitizen contribue à le plus à la 5e dimension.

Graphiques de contribution: Les deux graphiques ci-dessus confirme les observations effectuées plus haut, sur le cercle de corrélation et la matrice de corrélation des variables aux dimensions.

Graph contribution dim 1 à 4: Sur la contribution des variables aux dimension 1 à 4 ont s'aperçoit que le gender devance les autres dimensions, notamment grâce à sa contribution totale à la 3e dimension.

Etude de la colinéarité

Au sens strict, on parle de multicolinéarité parfaite lorsqu'une des variables explicatives d'un modèle est une combinaison linéaire d'une ou plusieurs autres variables explicatives introduites dans le même modèle. L'absence de multicolinéarité parfaite est une des conditions requises pour pouvoir estimer un modèle linéaire et, par extension, un modèle linéaire généralisé (dont les modèles de régression logistique). La multicolinéarité n'a aucune incidence sur l'adéquation de l'ajustement, ni sur la qualité de la prévision. Cependant, les coefficients individuels associés à chaque variable explicative ne peuvent pas être interprétés de façon fiable. c'est pour cela qu'il nous est nécessaire de la traiter.

```
reg <- glm(Churn ~ ., data = nume.kNN, family= binomial())
print(summary(reg))
```

```

## 
## Call:
## glm(formula = Churn ~ ., family = binomial(), data = nume.kNN)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.8697 -0.6927 -0.3731  0.7150  3.0907
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.869e+00  1.260e-01 -14.838 < 2e-16 ***
## gender                  1.473e-02  6.322e-02   0.233  0.815824
## SeniorCitizen          4.268e-01  8.290e-02   5.148 2.63e-07 ***
## Partner                 2.716e-02  7.584e-02   0.358  0.720279
## Dependents              -2.954e-01  8.681e-02  -3.403 0.000667 ***
## tenure                 -6.230e-02  5.347e-03 -11.651 < 2e-16 ***
## PaperlessBilling        5.358e-01  7.182e-02   7.460 8.64e-14 ***
## MonthlyCharges          3.338e-02  1.980e-03  16.862 < 2e-16 ***
## TotalCharges            2.108e-04  6.131e-05   3.439 0.000584 ***
## nb_services             -2.969e-01  3.162e-02  -9.388 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 8150.1 on 7042 degrees of freedom
## Residual deviance: 6146.6 on 7033 degrees of freedom
## AIC: 6166.6
##
## Number of Fisher Scoring iterations: 6

```

```
print(vif(reg))
```

	gender	SeniorCitizen	Partner	Dependents
##	1.001268	1.115230	1.382332	1.290524
##	tenure	PaperlessBilling	MonthlyCharges	TotalCharges
##	12.725115	1.108372	2.976800	16.998788
##	nb_services			
##	2.667144			

```

# Make predictions
predictions <- reg %>% predict(nume.kNN)
# Model performance
data.frame(
  RMSE = RMSE(predictions, nume.kNN$Churn),
  R2 = R2(predictions, nume.kNN$Churn)
)

```

```
##      RMSE      R2
## 1 2.315238 0.2321196
```

Le but ici est de déterminé la valeur de la corrélation à travers la colinéarité de nos variables. Une erreur fréquente est de confondre multicolinéarité et corrélation. Si des variables colinéaires sont de facto fortement corrélées entre elles, deux variables corrélées ne sont pas forcément colinéaires. En termes non statistiques, il y a colinéarité lorsque deux ou plusieurs variables mesurent la "même chose". Lors d'une colinéarité parfaite, les variables auront une corrélation égale à 1 mais à contrario, des variables corrélées à 1 ne sont pas forcément colinéaires. Pour estimer cette colinéarité, nous allons utiliser la méthode FIV ("facteurs d'inflation de la variance")

Nous pouvons voir sur l'output la présence de 2 variables qu'on peut fortement soupçonner de colinéarité de par leur VIF élevé. Il s'agit de tenure (13,37) et TotalCharges (17,2).

```

reg2 <- glm(Churn ~. -tenure , data = nume.kNN, family= binomial())
vif(reg2)

```

	gender	SeniorCitizen	Partner	Dependents
##	1.000949	1.111754	1.347553	1.293070
##	PaperlessBilling	MonthlyCharges	TotalCharges	nb_services
##	1.095372	2.211043	2.042526	2.399990

```
summary(reg2)
```

```

## 
## Call:
## glm(formula = Churn ~ . - tenure, family = binomial(), data = nume.kNN)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max 
## -2.0344 -0.6428 -0.4247  0.7025  2.6416 
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -2.941e+00  1.035e-01 -28.429 < 2e-16 ***
## gender       8.576e-03  6.285e-02   0.136  0.891476    
## SeniorCitizen 4.153e-01  8.269e-02   5.023  5.08e-07 ***
## Partner      -7.423e-02  7.424e-02  -1.000  0.317372    
## Dependents   -3.132e-01  8.580e-02  -3.650  0.000262 *** 
## PaperlessBilling 5.231e-01  7.068e-02   7.401  1.36e-13 *** 
## MonthlyCharges 4.714e-02  1.727e-03  27.295 < 2e-16 *** 
## TotalCharges  -4.886e-04  2.302e-05 -21.221 < 2e-16 *** 
## nb_services   -2.905e-01  3.058e-02  -9.497 < 2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 8150.1 on 7042 degrees of freedom
## Residual deviance: 6316.9 on 7034 degrees of freedom
## AIC: 6334.9
##
## Number of Fisher Scoring iterations: 5

```

```

# Make predictions
predictions2 <- reg2 %>% predict(nume.kNN)
# Model performance
data.frame(
  RMSE = RMSE(predictions2, nume.kNN$Churn),
  R2 = R2(predictions2, nume.kNN$Churn)
)

```

```

##      RMSE      R2
## 1 1.970191 0.2484877

```

Les FIV estimentent de combien la variance d'un coefficient est "augmentée" en raison d'une relation linéaire avec d'autres prédicteurs. Ainsi, un FIV de 1,8 nous dit que la variance de ce coefficient particulier est supérieure de 80 % à la variance que l'on aurait dû observer si ce facteur n'est absolument pas corrélé aux autres prédicteurs.

En supprimant tenure de la régression nous observons que la régression est de meilleurs qualité de part des R² et AIC croissants et un RMSE qui diminue.

N.B: si la 2 variables sont parfaitement colinéaires alors le coefficient de la variable sera doublé.

Variables qualitatives

Analyse composantes multiples (ACM)

```

res.mca <- MCA(kar.kNN, graph=FALSE, level.ventil = 0.05) #le level.ventil permet de ventiler les modalités s'ils représentent moins de 5% 

# garde les eigen values à chaque dimension pour servir en coordonnées pour faire une classification dessus
# comme k means
# le v kramer si proche de 1 indique si liaison mais pas l'intensité ni le sens ni le lien de causalité

# Obtention des valeurs propres

eig <- get_eigenvalue(res.mca)
eig

```

```

##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  6.697304e-01    3.348652e+01            33.48652
## Dim.2  2.376381e-01    1.188190e+01            45.36842
## Dim.3  2.161029e-01    1.080515e+01            56.17357
## Dim.4  1.267837e-01    6.339184e+00            62.51275
## Dim.5  9.324541e-02    4.662271e+00            67.17502
## Dim.6  9.087517e-02    4.543758e+00            71.71878
## Dim.7  8.583820e-02    4.291910e+00            76.01069
## Dim.8  7.676096e-02    3.838048e+00            79.84874
## Dim.9  6.890935e-02    3.445467e+00            83.29421
## Dim.10 6.601809e-02    3.300904e+00            86.59511
## Dim.11 6.445709e-02    3.222854e+00            89.81797
## Dim.12 6.070318e-02    3.035159e+00            92.85312
## Dim.13 5.228885e-02    2.614442e+00            95.46757
## Dim.14 4.731909e-02    2.365954e+00            97.83352
## Dim.15 4.215907e-02    2.107954e+00            99.94148
## Dim.16 1.170496e-03    5.852481e-02           100.00000
## Dim.17 2.814253e-28    1.407126e-26           100.00000
## Dim.18 2.507251e-28    1.253625e-26           100.00000
## Dim.19 9.913811e-29    4.956905e-27           100.00000
## Dim.20 8.164451e-29    4.082226e-27           100.00000
## Dim.21 7.440770e-29    3.720385e-27           100.00000
## Dim.22 2.869696e-29    1.434848e-27           100.00000

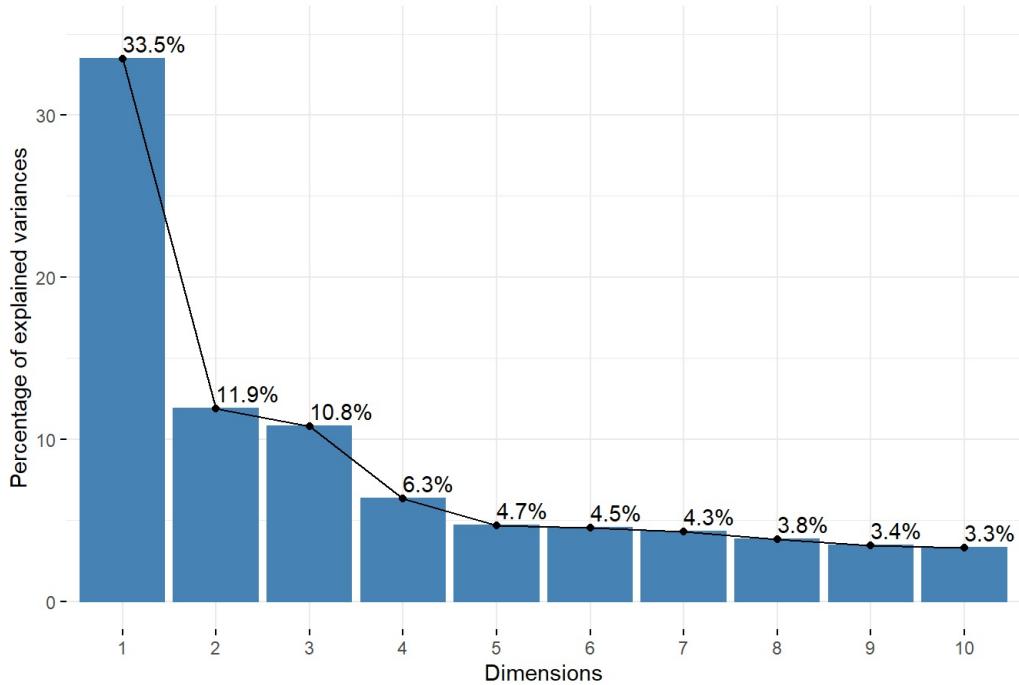
```

```

#Visualisation des axes
fviz_screplot (res.mca, addlabels = TRUE, ylim = c (0, 35))

```

Scree plot



```

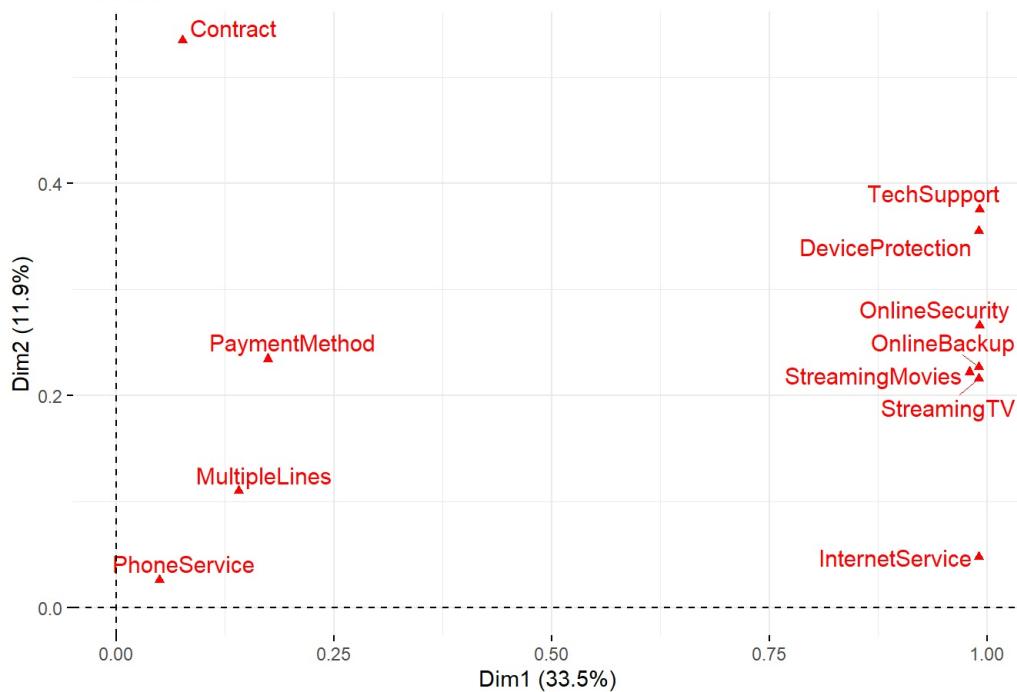
# Travaillons sur les variables
var <- get_mca_var(res.mca)
var$contrib

```

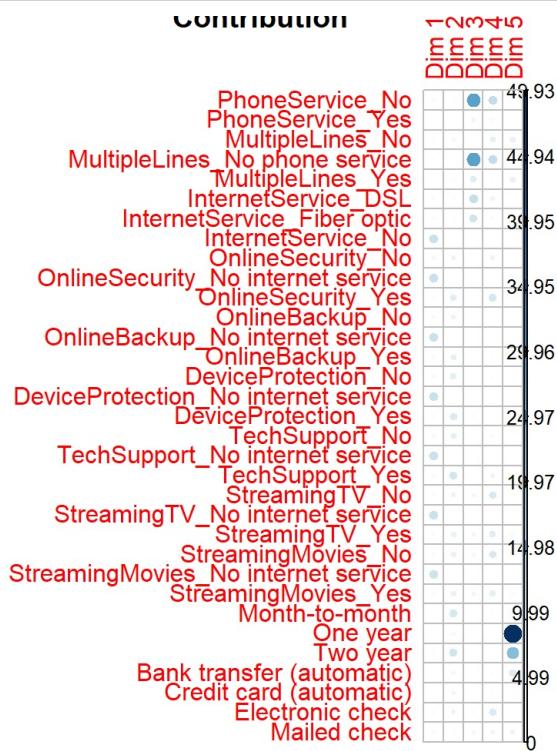
	Dim 1	Dim 2	Dim 3
##			
## PhoneService_No	6.074249e-01	0.898790065	2.676434e+01
## PhoneService_Yes	6.512558e-02	0.096364538	2.869561e+00
## MultipleLines_No	8.981135e-01	2.101226936	1.209359e-03
## MultipleLines_No phone service	6.074249e-01	0.898790065	2.676434e+01
## MultipleLines_Yes	4.081937e-01	1.197235671	6.329348e+00
## InternetService_DSL	1.252689e+00	1.079680248	1.163808e+01
## InternetService_Fiber optic	1.661419e+00	0.727436006	9.693667e+00
## InternetService_No	1.053461e+01	0.008824521	1.898241e-02
## OnlineSecurity_No	1.976157e+00	3.570229998	3.274433e-01
## OnlineSecurity_No internet service	1.053461e+01	0.008824521	1.898241e-02
## OnlineSecurity_Yes	9.436282e-01	6.598471471	4.012195e-01
## OnlineBackup_No	1.691281e+00	3.671951949	1.489936e-01
## OnlineBackup_No internet service	1.053461e+01	0.008824521	1.898241e-02
## OnlineBackup_Yes	1.223833e+00	4.995460532	2.963975e-01
## DeviceProtection_No	1.700733e+00	5.776667206	3.407493e-01
## DeviceProtection_No internet service	1.053461e+01	0.008824521	1.898241e-02
## DeviceProtection_Yes	1.214628e+00	7.792566686	5.917231e-01
## TechSupport_No	1.975467e+00	5.135486244	1.294875e-01
## TechSupport_No internet service	1.053461e+01	0.008824521	1.898241e-02
## TechSupport_Yes	9.454641e-01	9.211921618	1.225083e-01
## StreamingTV_No	1.482520e+00	3.908233126	2.326347e+00
## StreamingTV_No internet service	1.053461e+01	0.008824521	1.898241e-02
## StreamingTV_Yes	1.431351e+00	4.346038076	2.747068e+00
## StreamingMovies_No	1.379938e+00	4.222239846	2.412709e+00
## StreamingMovies_No internet service	1.044656e+01	0.009870526	1.940769e-02
## StreamingMovies_Yes	1.475826e+00	4.262511151	2.583863e+00
## Month-to-month	3.792254e-01	8.592186270	2.901425e-04
## One year	2.088102e-02	1.938567699	1.319902e-03
## Two year	6.402337e-01	9.949438957	3.551124e-03
## Bank transfer (automatic)	1.273355e-04	2.067653055	5.138802e-02
## Credit card (automatic)	6.066681e-04	2.805092672	2.214319e-04
## Electronic check	9.916881e-01	3.504195327	1.082046e+00
## Mailed check	1.371772e+00	0.588746937	2.238833e+00
##		Dim 4	Dim 5
## PhoneService_No	12.30478643	0.13087975	
## PhoneService_Yes	1.31926809	0.01403238	
## MultipleLines_No	5.11604716	4.61811155	
## MultipleLines_No phone service	12.30478643	0.13087975	
## MultipleLines_Yes	0.54089043	4.50367691	
## InternetService_DSL	3.32293241	0.22118733	
## InternetService_Fiber optic	1.58285423	0.10390369	
## InternetService_No	0.25404039	0.01775466	
## OnlineSecurity_No	3.74411438	0.03010726	
## OnlineSecurity_No internet service	0.25404039	0.01775466	
## OnlineSecurity_Yes	8.91090464	0.01266713	
## OnlineBackup_No	0.07999885	0.09074272	
## OnlineBackup_No internet service	0.25404039	0.01775466	
## OnlineBackup_Yes	0.51610900	0.05477275	
## DeviceProtection_No	0.91671075	0.53355617	
## DeviceProtection_No internet service	0.25404039	0.01775466	
## DeviceProtection_Yes	0.46547072	0.86766817	
## TechSupport_No	1.29665545	0.01040432	
## TechSupport_No internet service	0.25404039	0.01775466	
## TechSupport_Yes	3.68566574	0.06154891	
## StreamingTV_No	8.33925819	0.91371662	
## StreamingTV_No internet service	0.25404039	0.01775466	
## StreamingTV_Yes	6.57294035	1.15335693	
## StreamingMovies_No	8.26089875	0.99946146	
## StreamingMovies_No internet service	0.25627961	0.02097437	
## StreamingMovies_Yes	5.84664197	1.15509609	
## Month-to-month	0.65071752	1.63456032	
## One year	0.32340012	49.93265996	
## Two year	0.48403311	21.37980221	
## Bank transfer (automatic)	0.60315731	7.47791841	
## Credit card (automatic)	1.01072009	0.07601031	
## Electronic check	7.51384560	0.18076010	
## Mailed check	2.50667030	3.58501647	

#Correlation entre les variables et les axes... En rouge les variables actives et en bleu les illustratives
 fviz_mca_var (res.mca, choice = "mca.cor", repel = TRUE)

Variables - MCA

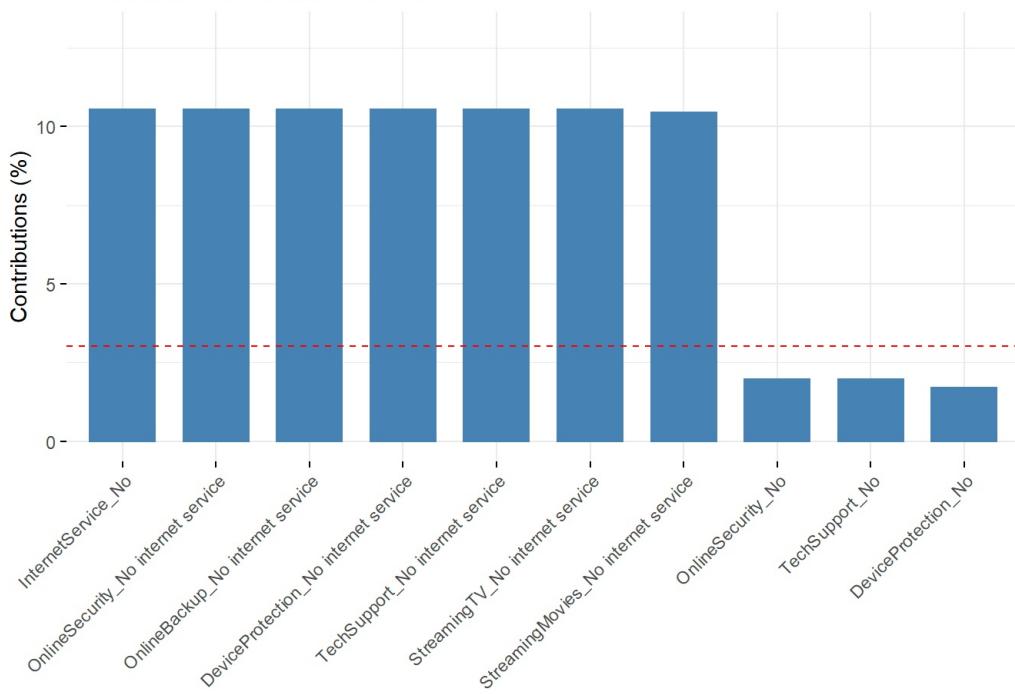


```
#contrib graph (des modalités)
corrplot(var$contrib, is.corr=FALSE, main="Contribution")
```



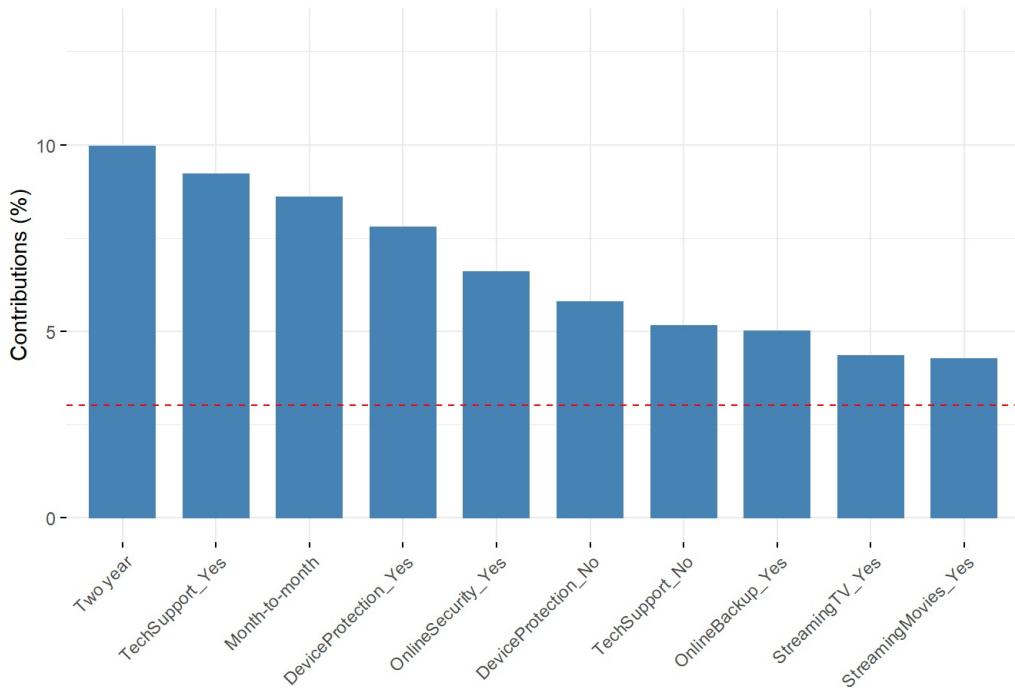
```
fviz_contrib(res.mca, choice = "var", axes = 1, top = 10, ylim = c (0, 13)) #histo par axe
```

Contribution of variables to Dim-1



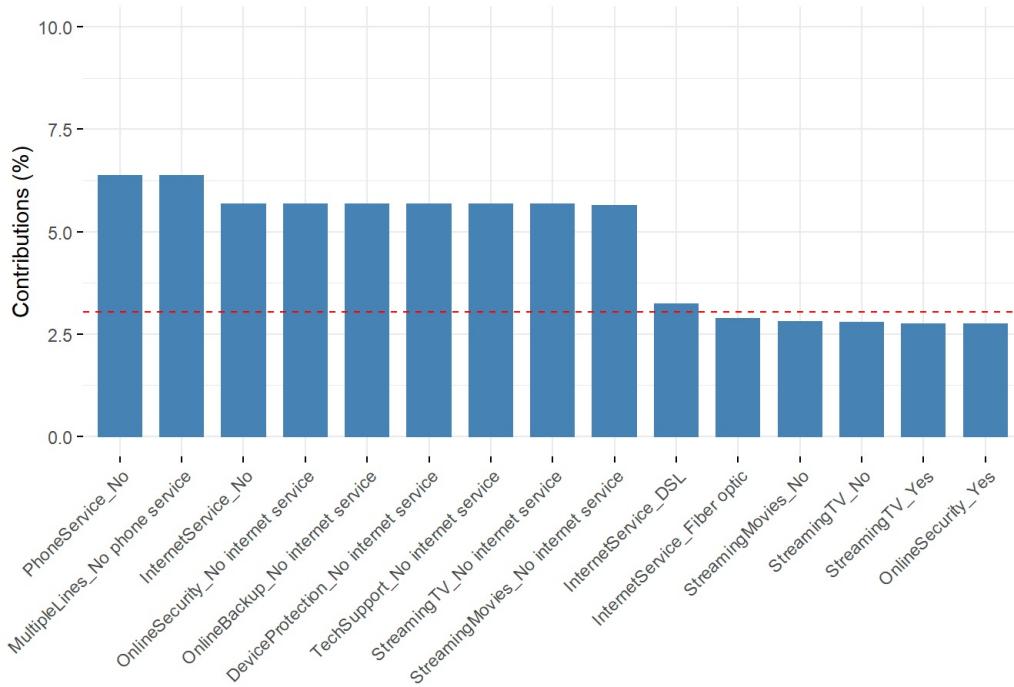
```
fviz_contrib(res.mca, choice = "var", axes = 2, top = 10, ylim = c (0, 13))
```

Contribution of variables to Dim-2



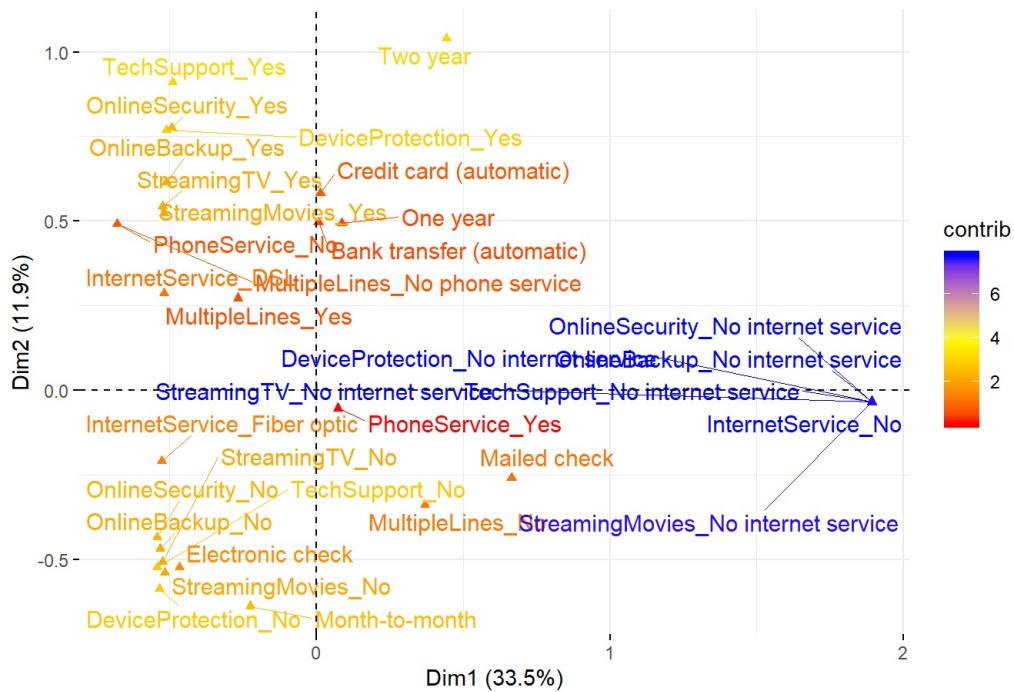
```
fviz_contrib(res.mca, choice = "var", axes = 1:4, top = 15, ylim = c (0, 10))
```

Contribution of variables to Dim-1-2-3-4



```
#contribution a la variable
#graph couleur par contrib
fviz_mca_var(res.mca, col.var = "contrib", gradient.cols = c("red", "yellow", "blue"), repel = TRUE)
```

Variable categories - MCA



Graph Scree Plot: On observe ici le pourcentage de variance expliquée par chaque dimension de notre ACM, la coudée se situant entre 2 ou 3 dimension. un pourcentage important est expliqué par la première dimension (33.5%), les dimension deux 3 expliquant plus de 10% à 12 et 10.8 pour-cent.

Graph variables MCA: La représentation ci-dessus nous permet d'observer des groupes de variables qualitatives. Sur la droite on observe un premier groupe de variables proches, qui sont 'TechSupport', 'DeviceProtection', 'OnlineBackup', 'StreamingTV', 'StreamingMovies', 'Online Backup', et un peu plus bas 'InternetService'. On peut constater que ce groupe s'apparente à un premier groupe de variables assimilé à une offre de services qui compose le contrat de chaque individus. Un second groupe de variables sur la gauche 'PaymentMethod' 'MultipleLines' 'PhoneService' compose un deuxième groupe qui s'apparente à un profil de client possédant ou non un service téléphonique, plusieurs ligne et payant de différentes manières. La variable 'contract' semble isolé de tout groupe.

Graph Corrélation aux dimensions : La matrice de contribution de l'ACM représente la contribution de chaque modalité des variables qualitatives à chaque dimmensions, A l'aide des graphiques ci-dessous on observe un peu mieux les modalités les plus contributrices des deux premières dimension et des 4 premières dimensions cumulés. Concernant la Dim 3 les modalités PhoneService_No et MultipleLines_NoPhoneService contribue le plus à la Dim4, les type de contrat One year et two year contribue le plus à la Dim5.

Graph Contributions 1&2: Les 7 premières modalités ci-dessus contribuent à plus de 10% à la Dim 1. InternetService_No, OnlineSecurity_No internet Service, OnlineBackup_No internet Service... jusqu'à Streamingmovies_No Internet service. On s'aperçoit que l'ensemble des modalités ci-dessus correspondent aux modalités de certaines variables du premier groupe que nous avions identifiés sur l'ACM. Les

contributions les plus élevées des modalités des variables à la seconde dimension sont le type de contrat_2 year, Tech_Support_Yes, contract_month-to-month, suivent ensuite les modalités de Deviceprotection, Online_security_yes, OnlibeBackup_yes et stremingmovies.

Graph contribution 1&2&3&4:On constate que la contribution des modalités des variables aux 4 dimension correspondent essentiellement à des réponses négatives.

Graph modalités MCA:Le graphique ci dessus permet d'observer plusieurs groupes de modalités vis à vis des 2 premières dimensions de l'ACM, colorés par rapport à leur contributions. on observe un premier groupe le plus contributeur en bleu sur la droite, avec notamment les modalités 'Online Backup_No internet Service', 'StreeamingMovies_No internet Service ', 'streamingTV_No internet Service' 'TechSupport_no internet Service' Un grand groupe sur la gauche, avec trois 'sous-groupes', moins contributeur en rouge au milieu et orange/jaune sur les parties hautes et basses du groupe. Les modalités Contract_two Year et Techsupport_yes Deviceprotection_yes notamment en haut à gauche contribue moyennement aux dimensions de même à l'opposé de l'axe on observe notamment les modalités Contract_month-to-month, DeviceProtection_No, StreamingMovies_No, Streamingtv_No. Au milieu des axes de ce grand groupes se situent les modalités les moins contributrices aux dimensions notamment 'phoneService_Yes', Contract_OneYear, BankTransfer(automatic), CreditCard(automatic).

Suspicion colinéarité entre MultipleLines et InternetServices

```
test <- table(kar.kNN$MultipleLines,kar.kNN$InternetService)
test
```

```
##                                     DSL Fiber optic   No
##  No           1048        1158 1184
##  No phone service 682          0    0
##  Yes          691        1938 342
```

```
chisq.test(test)
```

```
## 
##  Pearson's Chi-squared test
##
## data: test
## X-squared = 2217, df = 4, p-value < 2.2e-16
```

```
cramer.v(test)
```

```
## [1] 0.3967257
```

```
#sqrt(2217/(2*(1048 + 1158 + 1184 + 682 + 691 + 1938 +342)))
```

En général on accepte l'hypothèse d'indépendance lorsque p-value est supérieure à 5 % (0,05).

En l'occurrence p-value est inférieure à 1 pour mille (0,001) on rejette donc largement l'hypothèse d'indépendance. On peut donc affirmer avec moins d'une chance sur mille de se tromper qu'il existe un lien statistique entre les lignes et les colonnes de notre tableau. Et par conséquent entre nos variables MultipleLines et InternetService. Il nous reste maintenant qu'à quantifier cette relation afin de déterminer si celle-ci peut-être préjudiciable ou non. Pour la quantifier nous utilisons la méthode du V de Cramer

Int V.Cramer : [0;0.1[Relation nulle ou très faible [0.1;0.2[Relation faible]0.2;0.3[relation moyenne [0.3;inf[relation forte` La relation ici est donc forte mais étant donné que nous n'utiliserons pas de GLM dans la partie supervisée, nous nous en soucierons peu.

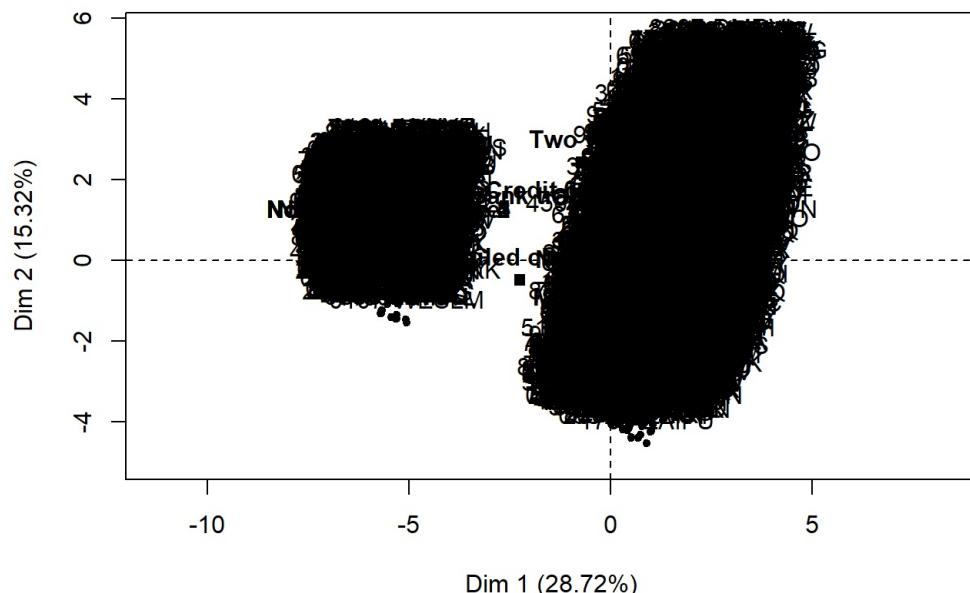
Analyse non supervisée

Tandem Analysis

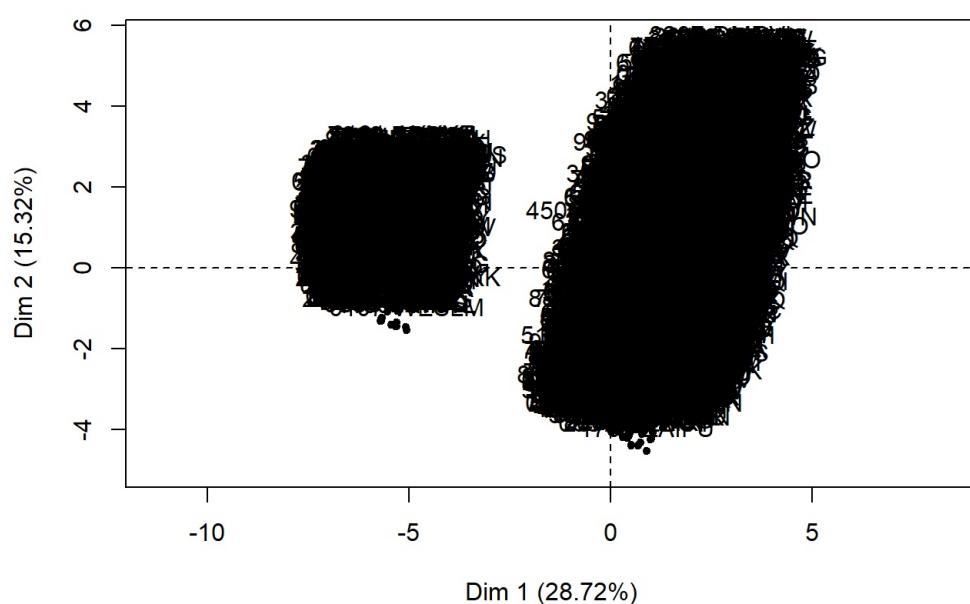
```
set.seed(496)
rownames(tab.kNN) <- tab.kNN$customerID
tab.kNN$'customerID' =NULL
data <- tab.kNN[1:3000,]

##Réalisation de l'Analyse factorielle multi dimensionnelle
res.famda <- FAMD(data,sup.var=data$Churn , graph = T)
```

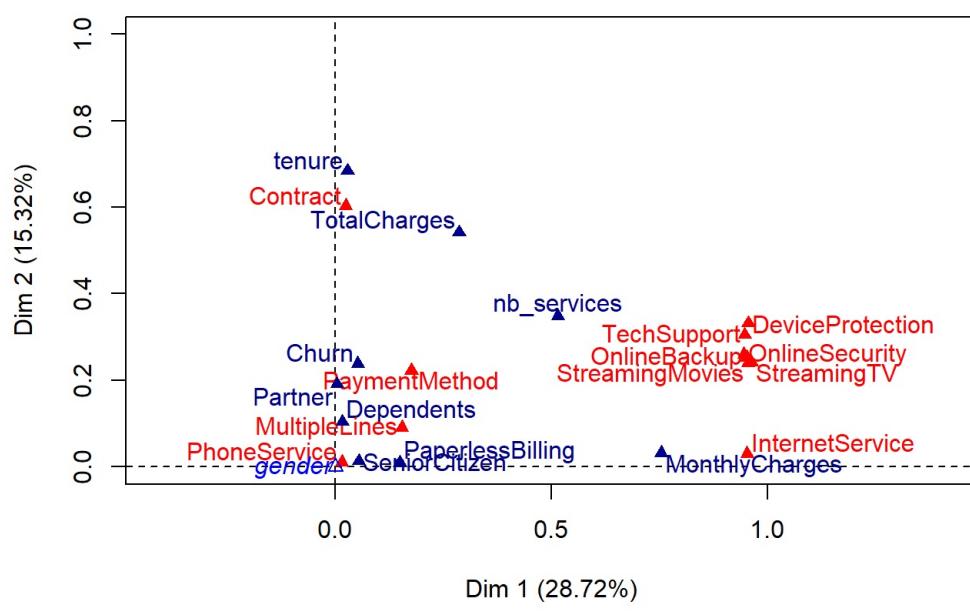
Individual factor map



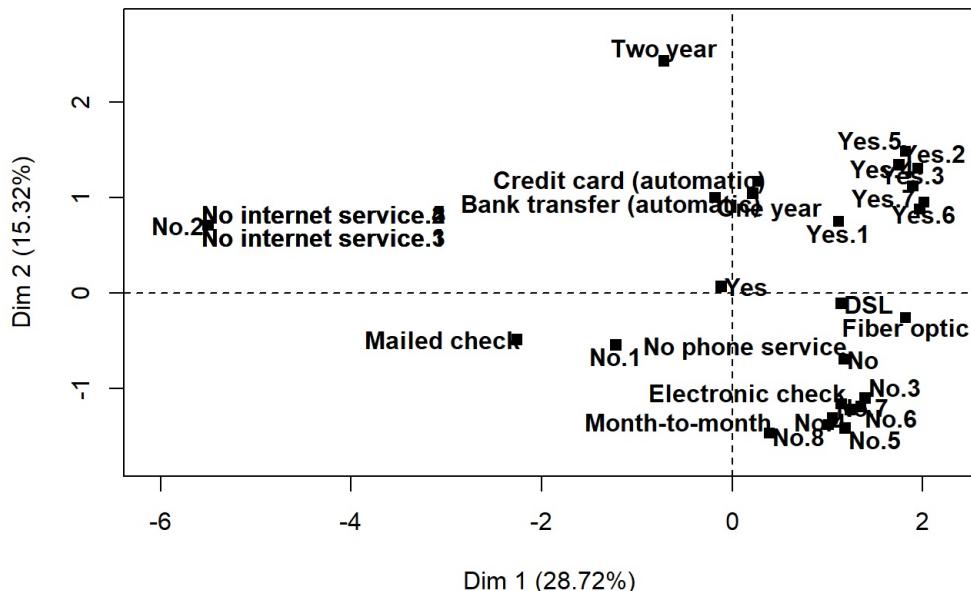
Individual factor map



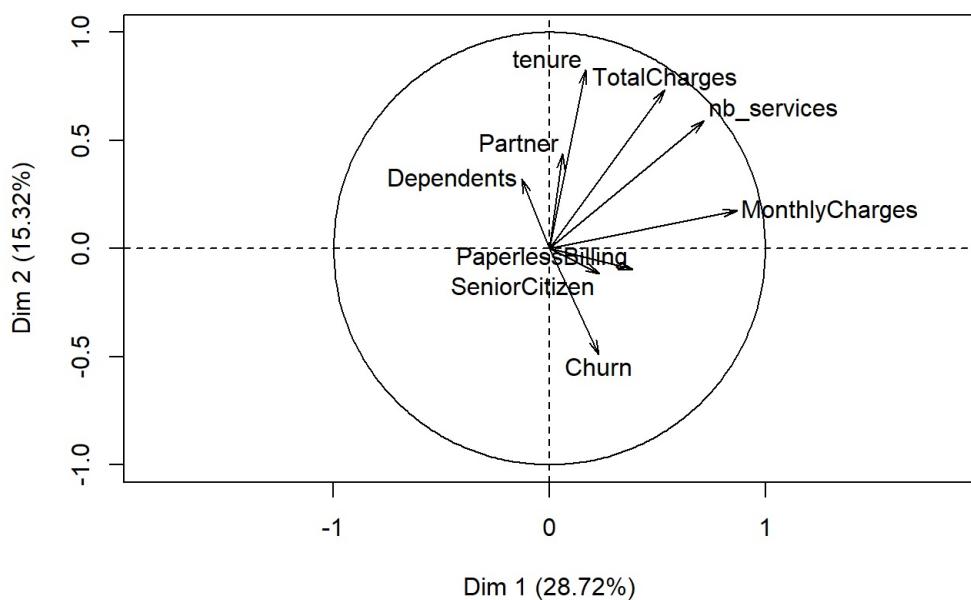
Graph of the variables



Graph of the categories



Graph of the quantitative variables

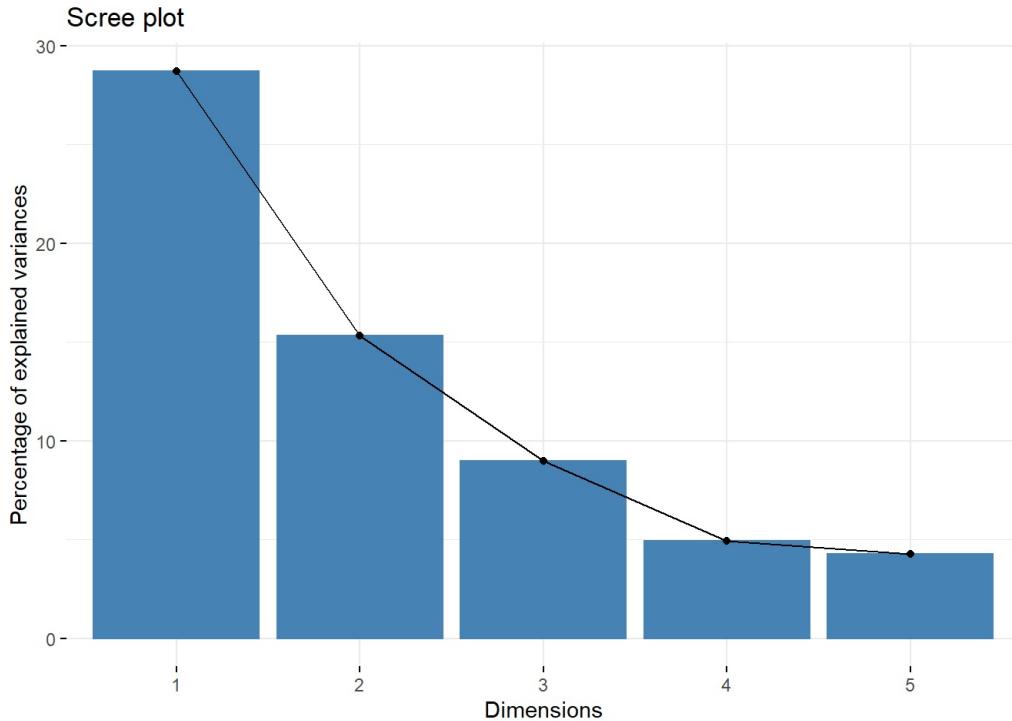


Interprétation graph variables (quanti/quali), modalités et cercle de corrélation:

Le diagnostic de ce graphique révèle que les variables de types « services » que les clients avaient souscrits semblent être les variables plus influentes au niveau de la première composante. En effet, la famille de variables « services » est en grande partie (y compris online security, streaming, device protection...) située à l'extrême droite de l'axe 1. Par ailleurs les variables de type « démographique » à savoir partner, dependents, senior, gender ne semblent pas être déterminantes au niveau de la première composante. Autrement dit, le niveau de dispersion des individus que capte cette première dimension est en grande partie du aux variables « services ». S'agissant de la seconde composante, on peut noter que les individus se distinguent le plus par leur ancienneté (c'est-à-dire vis-à-vis de la variable tenure), leur type de contrat et également par le total des charges.

Le graphique suivant vient affiner le graphique ci-dessus en y figurant non pas les variables mais plutôt les modalités que prennent ces variables. Comme nous l'avons constaté, les variables « services » sont relativement plus déterminantes au niveau de la première composante. Ici, dans ce graphique on peut corroborer ce constat. Par exemple, la modalité « no internet service » qui est une modalité commune à plusieurs variables (comme onlinesecurity, onlinebackup...) se trouve bien à l'extrême gauche de l'axe 1 pour la plupart des variables en question, alors que naturellement les réponses de types « yes » ou « no » sont du côté droit. Ceci traduit le fait que les différences entre les individus sont en rapport avec leurs choix de divers services que la compagnie leur offre en options. S'agissant de la deuxième composante, on remarque qu'à l'extrême positive (coordonées positives) de cet axe 2, on trouve la modalité « two year » qui indique que le contrat est relativement sur une longue durée, alors que l'autre extrémité négative, on y voit des modalités month-to-month, c'est-à-dire plutôt des clients du court terme qui évitent de s'engager sur plus long terme. La visualisation des variables quantitatives dans le même plan factoriel montre que la variables « monthlycharges » est la plus corrélée avec la première composante première. Dans le même temps, les variables tenure et Churn semblent être le plus corrélées à la deuxième composante principale.

```
eig.val <- get_eigenvalue(res.famd)
fviz_screepplot(res.famd)
```



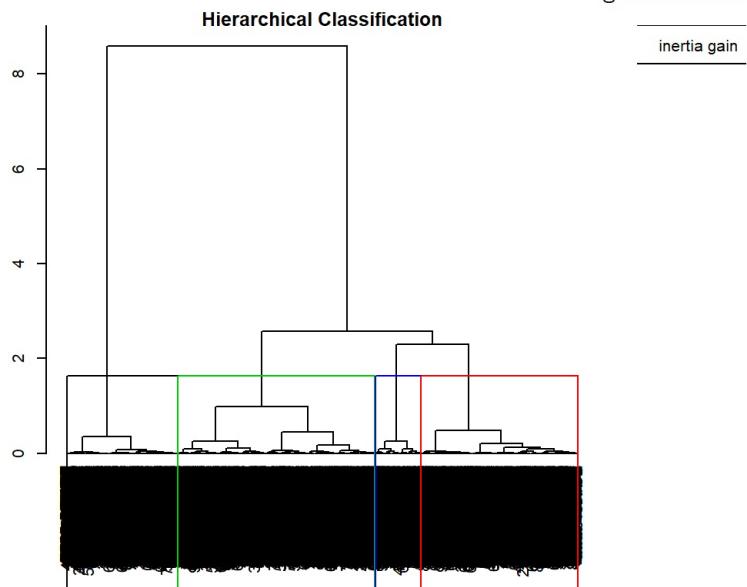
```
eig.val
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1      8.904598           28.724511                  28.72451
## Dim.2      4.748143           15.316590                  44.04110
## Dim.3      2.777825            8.960727                  53.00183
## Dim.4      1.532737            4.944312                  57.94614
## Dim.5      1.322238            4.265284                  62.21142
```

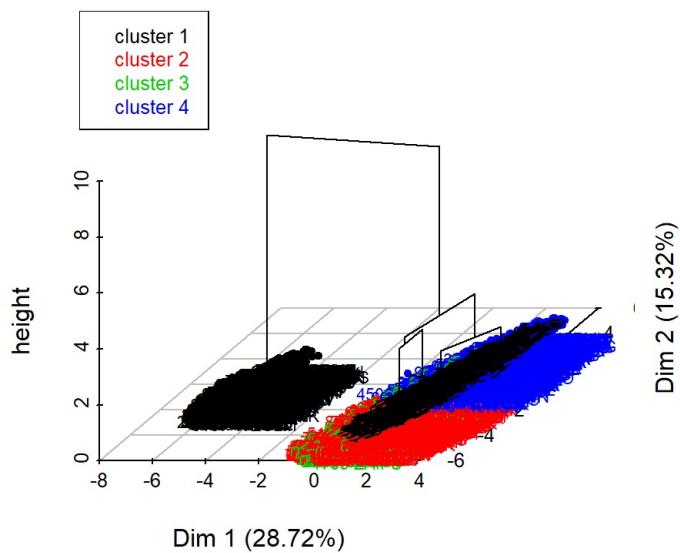
Interprétation Scree plot: La représentation de nuage des points-variables dans le plan factoriel permet d'avoir une idée plus claire de contributions de chaque variable dans la constitution de deux composantes principales (axe 1 et axe 2). Le premier axe capte à lui seul 28 % de l'inertie totale alors que le second retient uniquement environ 16 %. Ce premier plan factoriel retient environ 44 %.

```
## Réalisation de la CAH sur les données traitées par la FAMD
res.hcpc <- HCPC(res.famd,
                   nb.clust = -1, graph = TRUE)
```

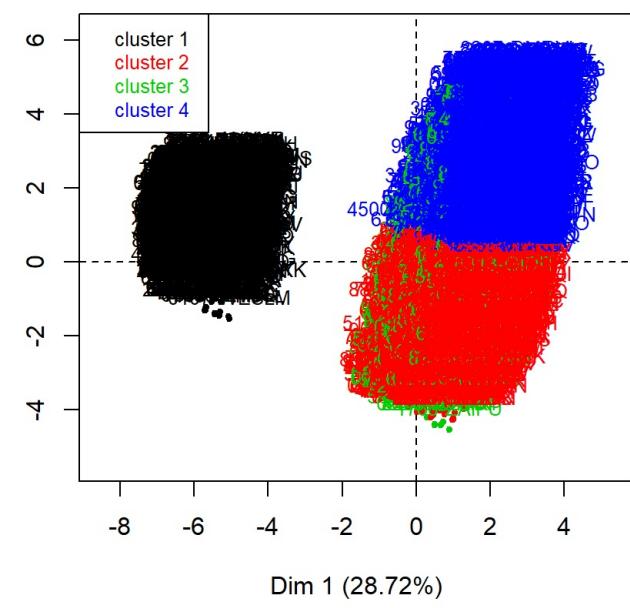
Hierarchical Clustering



Hierarchical clustering on the factor map



Factor map

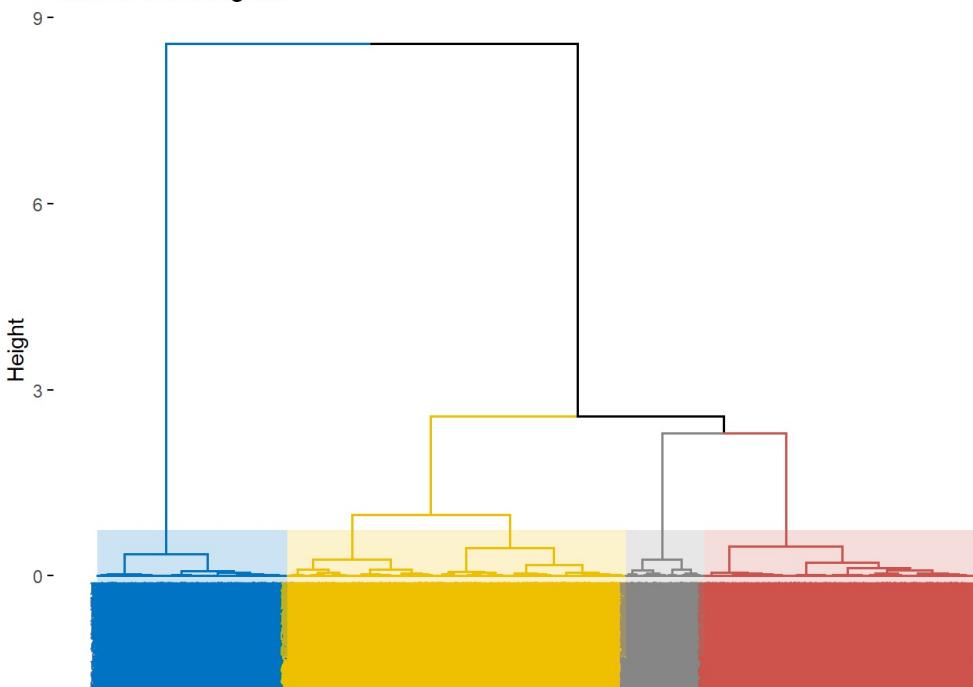


```

fviz_dend(res.hcpc,
          cex = 0.7,                      # Taille du texte
          palette = "jco",                 # Palette de couleur ?ggpubr::ggpar
          rect = TRUE, rect_fill = TRUE,   # Rectangle autour des groupes
          rect_border = "jco",             # Couleur du rectangle
          labels_track_height = 0.8       # Augment l'espace pour le texte
)

```

Cluster Dendrogram

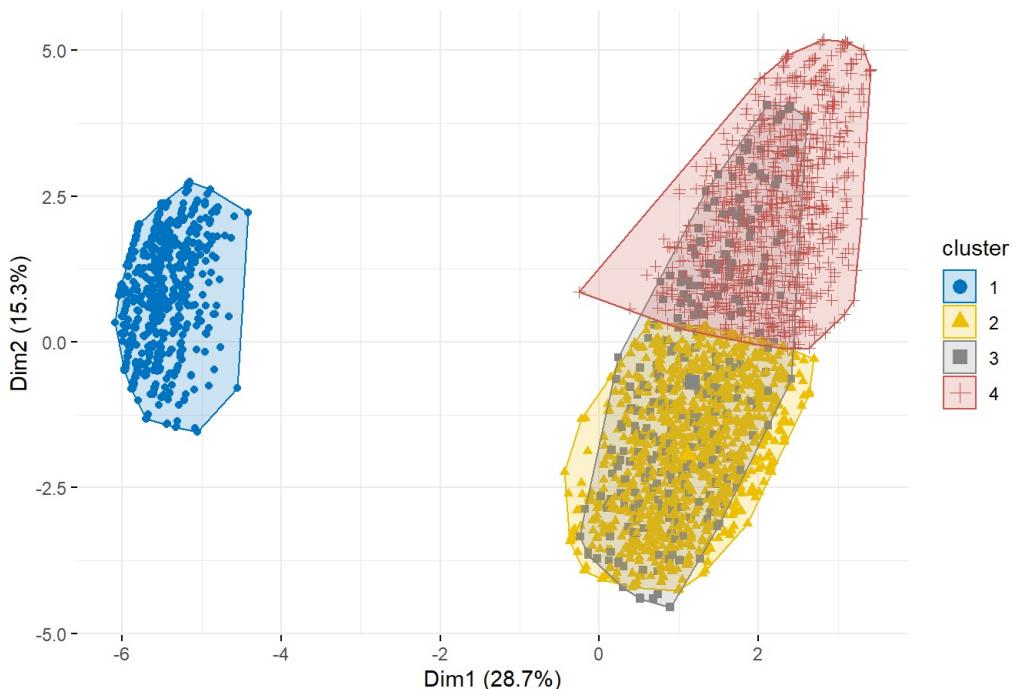


```

fviz_cluster(res.hcpc,
             geom = "point",
             repel = TRUE,                  # Evite le chevauchement des textes
             show.clust.cent = TRUE,        # Montre le centre des clusters
             palette = "jco",               # Palette de couleurs, voir ?ggpubr::ggpar
             ggtheme = theme_minimal(),
             main = "Factor map"
)

```

Factor map



```
head(res.hcpc$data.clust, 30)
```

```
## gender SeniorCitizen Partner Dependents tenure PhoneService
```

	MultipleLines	InternetService	OnlineSecurity			
## 7590-VHVEG	1	0	1	0	1	No
## 5575-GNVDE	0	0	0	0	34	Yes
## 3668-QPYBK	0	0	0	0	2	Yes
## 7795-CFOCW	0	0	0	0	45	No
## 9237-HQITU	1	0	0	0	2	Yes
## 9305-CDSKC	1	0	0	0	8	Yes
## 1452-KIOVK	0	0	0	1	22	Yes
## 6713-OKOMC	1	0	0	0	10	No
## 7892-POOKP	1	0	1	0	28	Yes
## 6388-TABGU	0	0	0	1	62	Yes
## 9763-GRSKD	0	0	1	1	13	Yes
## 7469-LKBCI	0	0	0	0	16	Yes
## 8091-TTVAX	0	0	1	0	58	Yes
## 0280-XJGEX	0	0	0	0	49	Yes
## 5129-JLPIS	0	0	0	0	25	Yes
## 3655-SNQYZ	1	0	1	1	69	Yes
## 8191-XWSZG	1	0	0	0	52	Yes
## 9959-WOFKT	0	0	0	1	71	Yes
## 4190-MFLUW	1	0	1	1	10	Yes
## 4183-MYFRB	1	0	0	0	21	Yes
## 8779-QRDMV	0	1	0	0	1	No
## 1680-VDCWW	0	0	1	0	12	Yes
## 1066-JKSGK	0	0	0	0	1	Yes
## 3638-WEABW	1	0	1	0	58	Yes
## 6322-HRPFA	0	0	1	1	49	Yes
## 6865-JZNKO	1	0	0	0	30	Yes
## 6467-CHFZW	0	0	1	1	47	Yes
## 8665-UTDHZ	0	0	1	1	1	No
## 5248-YGIJN	0	0	1	0	72	Yes
## 8773-HHUOZ	1	0	0	1	17	Yes
##	MultipleLines	InternetService	OnlineSecurity			
## 7590-VHVEG	No	phone service	DSL			No
## 5575-GNVDE		No	DSL			Yes
## 3668-QPYBK		No	DSL			Yes
## 7795-CFOCW	No	phone service	DSL			Yes
## 9237-HQITU		No	Fiber optic			No
## 9305-CDSKC	Yes		Fiber optic			No
## 1452-KIOVK	Yes		Fiber optic			No
## 6713-OKOMC	No	phone service	DSL			Yes
## 7892-POOKP	Yes		Fiber optic			No
## 6388-TABGU	No		DSL			Yes
## 9763-GRSKD	No		DSL			Yes
## 7469-LKBCI	No		No No internet service			
## 8091-TTVAX	Yes		Fiber optic			No
## 0280-XJGEX	Yes		Fiber optic			No
## 5129-JLPIS	No		Fiber optic			Yes
## 3655-SNQYZ	Yes		Fiber optic			Yes
## 8191-XWSZG	No		No No internet service			
## 9959-WOFKT	Yes		Fiber optic			Yes
## 4190-MFLUW	No		DSL			No
## 4183-MYFRB	No		Fiber optic			No
## 8779-QRDMV	No	phone service	DSL			No
## 1680-VDCWW	No		No No internet service			
## 1066-JKSGK	No		No No internet service			
## 3638-WEABW	Yes		DSL			No
## 6322-HRPFA	No		DSL			Yes
## 6865-JZNKO	No		DSL			Yes
## 6467-CHFZW	Yes		Fiber optic			No
## 8665-UTDHZ	No	phone service	DSL			No
## 5248-YGIJN	Yes		DSL			Yes
## 8773-HHUOZ	No		DSL			No
##	OnlineBackup	DeviceProtection	TechSupport			
## 7590-VHVEG	Yes		No			No
## 5575-GNVDE	No		Yes			No
## 3668-QPYBK	Yes		No			No
## 7795-CFOCW	No		Yes			Yes
## 9237-HQITU	No		No			No
## 9305-CDSKC	No		Yes			No
## 1452-KIOVK	Yes		No			No
## 6713-OKOMC	No		No			No
## 7892-POOKP	No		Yes			Yes
## 6388-TABGU	Yes		No			No
## 9763-GRSKD	No		No			No
## 7469-LKBCI	No internet service	No internet service	No internet service			
## 8091-TTVAX	No		Yes			No
## 0280-XJGEX	Yes		Yes			No
## 5129-JLPIS	No		Yes			Yes
## 3655-SNQYZ	Yes		Yes			Yes
## 8191-XWSZG	No internet service	No internet service	No internet service			

	No	Yes	No	
## 9959-WOKFT	No	Yes	No	
## 4190-MFLUW	No	Yes	Yes	
## 4183-MYFRB	Yes	Yes	No	
## 8779-QRDMV	No	Yes	No	
## 1680-VDCWW	No internet service	No internet service	No internet service	
## 1066-JKSGK	No internet service	No internet service	No internet service	
## 3638-WEABW	Yes	No	Yes	
## 6322-HRPFA	Yes	No	Yes	
## 6865-JZNKO	Yes	No	No	
## 6467-CHFZW	Yes	No	No	
## 8665-UTDHZ	Yes	No	No	
## 5248-YGIJN	Yes	Yes	Yes	
## 8773-HHUOZ	No	No	No	
##	StreamingTV	StreamingMovies	Contract	
## 7590-VHVEG	No	No Month-to-month		
## 5575-GNVDE	No	No One year		
## 3668-QPYBK	No	No Month-to-month		
## 7795-CFOCW	No	No One year		
## 9237-HQITU	No	No Month-to-month		
## 9305-CDSKC	Yes	Yes Month-to-month		
## 1452-KIOVK	Yes	No Month-to-month		
## 6713-OKOMC	No	No Month-to-month		
## 7892-POOKP	Yes	Yes Month-to-month		
## 6388-TABGU	No	No Month-to-month		
## 9763-GRSKD	No	No Month-to-month		
## 7469-LKBCI	No internet service	No internet service	Month-to-month	
## 8091-TTVAX	Yes	Yes One year		
## 0280-XJGEX	Yes	Yes Month-to-month		
## 5129-JLPIS	Yes	Yes Month-to-month		
## 3655-SNQYZ	Yes	Yes Two year		
## 8191-XWSZG	No internet service	No internet service	One year	
## 9959-WOKFT	Yes	Yes Two year		
## 4190-MFLUW	No	No Month-to-month		
## 4183-MYFRB	No	Yes Month-to-month		
## 8779-QRDMV	No	Yes Month-to-month		
## 1680-VDCWW	No internet service	No internet service	One year	
## 1066-JKSGK	No internet service	No internet service	Month-to-month	
## 3638-WEABW	No	No Two year		
## 6322-HRPFA	No	No Month-to-month		
## 6865-JZNKO	No	No Month-to-month		
## 6467-CHFZW	Yes	Yes Month-to-month		
## 8665-UTDHZ	No	No Month-to-month		
## 5248-YGIJN	Yes	Yes Two year		
## 8773-HHUOZ	Yes	Yes Month-to-month		
##	PaperlessBilling	PaymentMethod	MonthlyCharges	
## 7590-VHVEG	1	Electronic check	29.85	
## 5575-GNVDE	0	Mailed check	56.95	
## 3668-QPYBK	1	Mailed check	53.85	
## 7795-CFOCW	0	Bank transfer (automatic)	42.30	
## 9237-HQITU	1	Electronic check	70.70	
## 9305-CDSKC	1	Electronic check	99.65	
## 1452-KIOVK	1	Credit card (automatic)	89.10	
## 6713-OKOMC	0	Mailed check	29.75	
## 7892-POOKP	1	Electronic check	104.80	
## 6388-TABGU	0	Bank transfer (automatic)	56.15	
## 9763-GRSKD	1	Mailed check	49.95	
## 7469-LKBCI	0	Credit card (automatic)	18.95	
## 8091-TTVAX	0	Credit card (automatic)	100.35	
## 0280-XJGEX	1	Bank transfer (automatic)	103.70	
## 5129-JLPIS	1	Electronic check	105.50	
## 3655-SNQYZ	0	Credit card (automatic)	113.25	
## 8191-XWSZG	0	Mailed check	20.65	
## 9959-WOKFT	0	Bank transfer (automatic)	106.70	
## 4190-MFLUW	0	Credit card (automatic)	55.20	
## 4183-MYFRB	1	Electronic check	90.05	
## 8779-QRDMV	1	Electronic check	39.65	
## 1680-VDCWW	0	Bank transfer (automatic)	19.80	
## 1066-JKSGK	0	Mailed check	20.15	
## 3638-WEABW	1	Credit card (automatic)	59.90	
## 6322-HRPFA	0	Credit card (automatic)	59.60	
## 6865-JZNKO	1	Bank transfer (automatic)	55.30	
## 6467-CHFZW	1	Electronic check	99.35	
## 8665-UTDHZ	1	Electronic check	30.20	
## 5248-YGIJN	1	Credit card (automatic)	90.25	
## 8773-HHUOZ	1	Mailed check	64.70	
##	TotalCharges	nb_services	Churn	clust
## 7590-VHVEG	29.85	1	0	3
## 5575-GNVDE	1889.50	2	0	2
## 3668-QPYBK	108.15	1	1	2

```

## 7795-CFOCW 1840.75    3   0   3
## 9237-HQITU 151.65     1   1   2
## 9305-CDSKC 820.50     3   1   2
## 1452-KIOVK 1949.40    2   0   2
## 6713-OKOMC 301.90     1   0   3
## 7892-POOKP 3046.05    4   1   4
## 6388-TABGU 3487.95    2   0   2
## 9763-GRSKD 587.45     1   0   2
## 7469-LKBCI 326.80     0   0   1
## 8091-TTVAX 5681.10    3   0   4
## 0280-XJGEX 5036.30    4   1   4
## 5129-JLPIS 2686.05    5   0   4
## 3655-SNQYZ 7895.15    6   0   4
## 8191-XWSZG 1022.95    0   0   1
## 9959-WOKFT 7382.25    4   0   4
## 4190-MFLUW 528.35     2   1   2
## 4183-MYFRB 1862.90    3   0   2
## 8779-QRDMV 39.65      2   1   3
## 1680-VDCWW 202.25     0   0   1
## 1066-JKSGK 20.15      0   1   1
## 3638-WEABW 3505.10    2   0   4
## 6322-HRPFA 2970.30    3   0   4
## 6865-JZNKO 1530.60    2   0   2
## 6467-CHFZW 4749.15    3   1   4
## 8665-UTDHZ 30.20      1   1   3
## 5248-YGIJN 6369.45    6   0   4
## 8773-HHUOZ 1093.10    2   1   2

```

```

a <- res.hcpc$data.clust
a$Churn <- data$Churn
table(a$Churn, a$clust)

```

```

##
##      1   2   3   4
## 0 612 612 204 792
## 1  39 555  65 121

```

La classification permet de mieux visualiser les différences entre les différents profils de clients en les classant dans un nombre restreint de classes dont l'inertie intra-classe est la plus minimale possible et l'inertie interclasse la plus forte possible. L'idée est d'agrégier les classes tout en minimisant la perte d'inertie inter. Le premier graphique montre la classification hiérarchique ascendante ainsi que la perte d'inertie inter consécutives à chaque agrégation des classes. Ce calcul est à la base de la méthode de Ward qui détermine le nombre de classes optimales minimisant la perte d'inertie inter. Ainsi, ce graphique montre, par exemple, que le fait de passer de deux classes à une seule fait perdre en inertie inter environ 8 (à comparer par rapport à l'inertie totale qui n'est d'autre le nombre de variables retenues dans le modèle). On peut noter qu'en passant de quatre classes à trois, que la perte d'inertie commence à être plus conséquente. C'est ce qui explique notre choix de retenir quatre classes. La visualisation dans le plan factoriel de différents individus selon leur appartenance aux quatre classes ainsi identifiées nous permet de bien voir ce qui distingue ces classes. La classe 1 se distingue par rapport aux trois autres classes. La classe 1 se distingue notamment selon la première composante. Le recouplement avec ce que nous avons développé plus haut implique que comme tous les clients de classe 1 sont du côté gauche de l'axe 1 et que les modalités qui se trouvent dans cette région sont surtout des gens n'ayant pas service internet et qui sont également des « no partner » (qui n'ont pas de partenaires). Les classes 2 à 4 se distinguent très peu selon la première composante tant que tous les individus sont du côté positif de cet axe. La deuxième composante a plus de pouvoir de différenciation notamment entre la classe 2 et la classe 4. Si l'on transpose aussi les modalités dans ce même plan factoriel où sont situées les différentes classes, on peut dire que les clients appartenant à la classe 4 semblent être, par exemple, plus enclins à avoir des contrats sur une longue durée que la classe 2 englobe des clients plutôt de courte durée et qui sont probablement de potentiels « chameurs ». Le croisement de quatre classes avec selon si le client est « chameur » ou pas montre qu'effectivement une grande partie des clients contenus dans la classe sont des chameurs, et dans une moindre proportion la classe 4 a relativement une part de clients chameurs également.

K-Means

Le recours à la méthode K-means qui est un algorithme d'agrégation autour des centres mobiles. Cet algorithme débute avec un choix arbitraire des centres de classe et ensuite affecter chaque individu selon sa proximité au centre le plus proche.

```

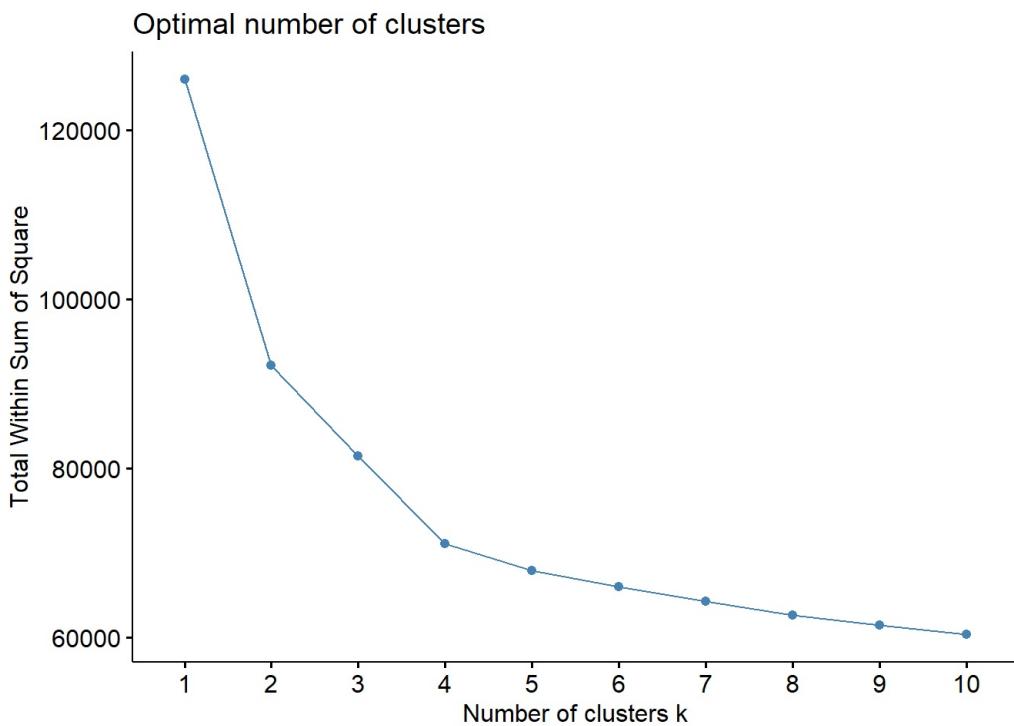
set.seed(496)
dmy2 <- dummyVars(~ ., data = tab.kNN)
dmy2 <- data.frame(predict(dmy2, newdata = tab.kNN))
dmy2$"Churn"=NULL
dmy2 <- dmy2[1:3000,]

dta.cr <- scale(dmy2)

d.dta <- dist(dta.cr)

## Detection du nombre optimal de classes
## Avec factoextra
fviz_nbclust(dta.cr, hcut, method = "wss") #1er output

```



1er output: Le graphique qui présente le nombre optimal des classes indique la variabilité intra ou within qui décroît avec le nombre de classes. On voit que la courbe est relativement coudée au niveau de nombre de classes 4, ce qui indique que le nombre optimal est quatre classes. Ainsi, la CAH ainsi que la méthode K-means s'accordent sur le nombre de classes.

```
set.seed(496)
## CAH
cah.ward <- hclust(d.dta,method="ward.D2")
cah.ward
```

```
##
## Call:
## hclust(d = d.dta, method = "ward.D2")
##
## Cluster method : ward.D2
## Distance       : euclidean
## Number of objects: 3000
```

```
set.seed(496)
groupes.cah <- cutree(cah.ward,k=4) ## Couper le dendrogramme en 4 groupes comme vu sur le graphique précédent

#K-Means
groupes.kmeans <- kmeans(d.dta,centers=4,nstart=5)
head(groupes.kmeans$cluster,20)
```

```
## 7590-VHVEG 5575-GNVDE 3668-QPYBK 7795-CFOCW 9237-HQITU 9305-CDSKC 1452-KIOVK
##      1      2      2      1      2      2      2
## 6713-OKOMC 7892-P0OKP 6388-TABGU 9763-GRSKD 7469-LKBCI 8091-TTVAX 0280-XJGEX
##      1      3      2      2      4      3      3
## 5129-JLPIS 3655-SNQYZ 8191-XWSZG 9959-WOFKT 4190-MFLUW 4183-MYFRB
##      3      3      4      3      2      2
```

```
## Correspondance avec les groupes de la CAH

print(table(groupes.cah,groupes.kmeans$cluster)) #2ème output
```

```
##
## groupes.cah   1   2   3   4
##      1  269   0   0   0
##      2   0 1146 190   0
##      3   0    39 705   0
##      4   0    0   0 651
```

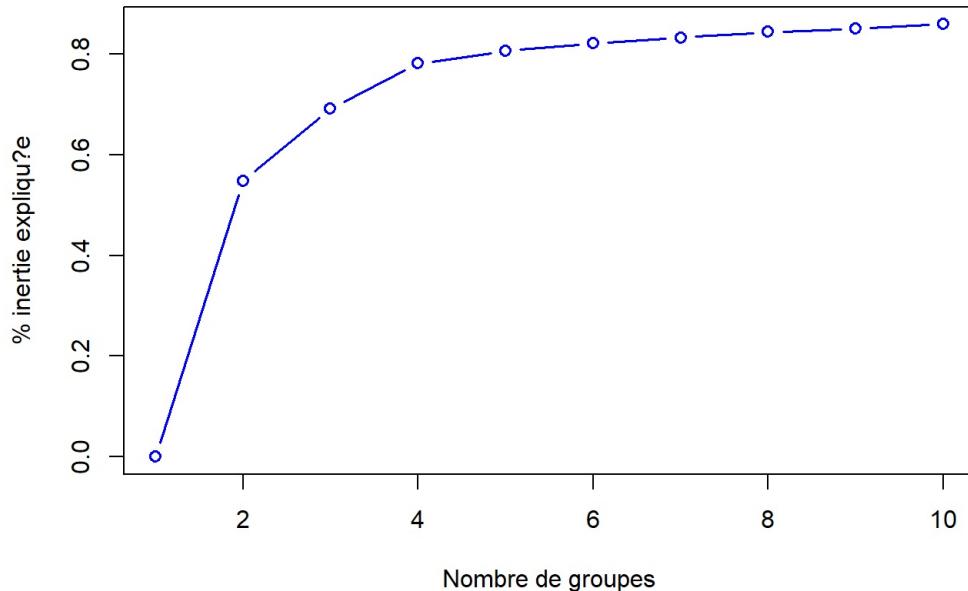
2eme output: La classification selon cette méthode montre une part de l'inertie between dans l'inertie totale qui représente environ 77%. D'ailleurs, le graphique où on a mis en abscisses le nombre de classes et en ordonnées, la part de l'inertie expliquée montre que l'augmentation de cette part d'inertie expliquée s'accroît fortement quand passe d'une classe à deux, mais s'affaiblit quand le nombre de classes augmente. On peut remarquer qu'à partir de la classe 4, la hausse de l'inertie expliquée commence à s'accroître encore d'une façon très modeste. Ceci corrobore le choix de quatre classes.

```

## Evaluer la proportion d'inertie expliquée
inertie.expl <- rep(0,times=10)
for (k in 2:10){
  clus <- kmeans(d.dta,centers=k,nstart=5)
  inertie.expl[k] <- clus$betweenss/clus$totss
}

#graphique de l'inertie expliquée (3ème output)
plot(1:10,inertie.expl,type="b",xlab="Nombre de groupes",
      ylab="% inertie expliquée", col="blue", lwd=1.5)

```



3ème output: Pour comparer les similarités dans les compositions de différentes classes, nous avons fait un recouplement de deux méthodes. Nous voyons beaucoup de divergence entre les classes affectés par ces méthodes différentes. On juge donc que cette méthode n'est pas fiable pour notre problème, ou tout du moins, moins fiable que la première.

Apprentissage supervisé

Préparation des données

Nous séparons ici notre base de données en divers jeu. Il nous faut ici au minimum une base de test et une base d'entraînement. Une base d'entraînement sur laquel le modèle va s'entraîner et une base de test pour voir la qualité de notre modèle.

```

set.seed(496)
dmy <- tab.kNN
rownames(dmy) <- dmy$customerID
dmy$"customerID" <- NULL
dmy <- dummyVars(" ~ .", data = dmy)
dmy <- data.frame(predict(dmy, newdata = tab.kNN))

ind=createDataPartition(dmy$Churn,times=1,p=0.8,list=FALSE)

train_val=dmy[ind,]
test_val=dmy[-ind,]
X_train=train_val[,-43]
y_train=as.factor(train_val[,43])
X_test=test_val[,-43]
y_test=as.factor(test_val[,43])

train_val2=train_val[,c("tenure","TotalCharges","MonthlyCharges","Contract.Month.to.month","Contract.Two.year","TechSupport.No","InternetService.DSL","InternetService.Fiber.optic","gender","Contract.Two.year","Partner","OnlineBackup.No","OnlineSecurity.No","PaperlessBilling","PaymentMethod.Electronic.check","nb_services","Contract.One.year","Churn")]
test_val2=test_val[,c("tenure","TotalCharges","MonthlyCharges", "Contract.Month.to.month","Contract.Two.year","TechSupport.No","InternetService.DSL","InternetService.Fiber.optic","gender","Contract.Two.year","Partner","OnlineBackup.No","OnlineSecurity.No","PaperlessBilling","PaymentMethod.Electronic.check","nb_services","Contract.One.year","Churn")]
X_train2=train_val2[,-18]
y_train2=as.factor(train_val2[,18])
X_test2=test_val2[,-18]
y_test2=as.factor(test_val2[,18])

```

Abre de décision

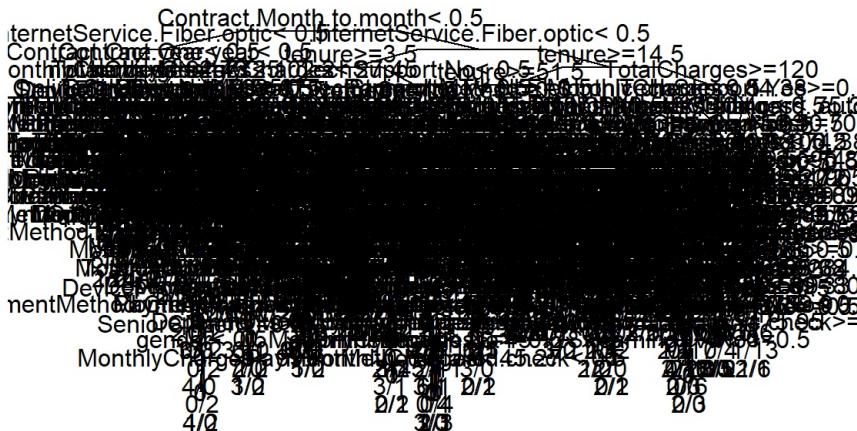
L'objectif de notre arbre de décision est donc de classer les clients en 2 classes en fonction de leur churn ou non. A chaque étape, l'algorithme va chercher le critère permettant de séparer au mieux ces deux populations. Il existe plusieurs critères de séparation, dans notre exemple nous allons utiliser l'indice de Gini qui est utilisé pour les arbres CART

Pour construire notre arbre de décision, nous allons utiliser la commande `rpart`. Nous avons vu dans le cours, certains des paramètres adaptés à notre situation : nous avons ici la méthode qui est de type "class" car nous sommes dans une situation où la target est binaire; nous avons le paramètre `minsplit` qui est fixé à 5, c'est-à-dire que l'arbre continuera à séparer les données tant que l'arbre contiendra au moins 5 données. Enfin le paramètre `CP` est indicateur d'amélioration du modèle, si celui-ci est fixé à 0 alors le découpage à quand même lieu même si celui-ci n'améliore pas le modèle.

```

set.seed(496)
#Construction de l'arbre
class.churn <- rpart(Churn~, data=train_val, method="class",control= rpart.control(minsplit=5,cp=0))
#affichage du résultat
plot(class.churn , uniform=TRUE, branch =0.5, margin =0.1)
text(class.churn, all=FALSE, use.n=TRUE)

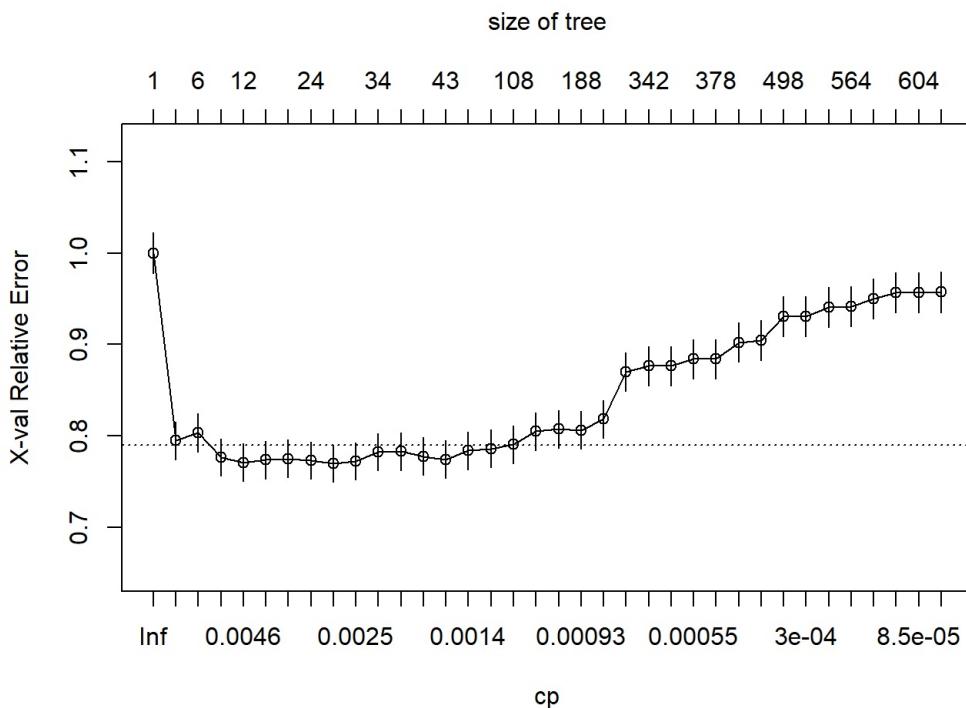
```



Résultat: Nous avons ici un arbre illisible dont il nous est impossible de tirer la moindre conclusion. Il est beaucoup trop développé et nécessite un élagage afin de le simplifier et pour éviter surtout le surapprentissage.

L'élagage fonctionne par niveau, nous devons tout d'abord le fixer. Pour cela, nous créons un graph qui nous permettra de visualiser le taux de mauvais classement en fonction de la complexité paramétrée (CP). On pourra ainsi minimiser l'erreur pour un complexité donnée.

```
set.seed(496)
#On cherche à minimiser l'erreur pour définir le niveau d'élagage
plotcp(class.churn)
```



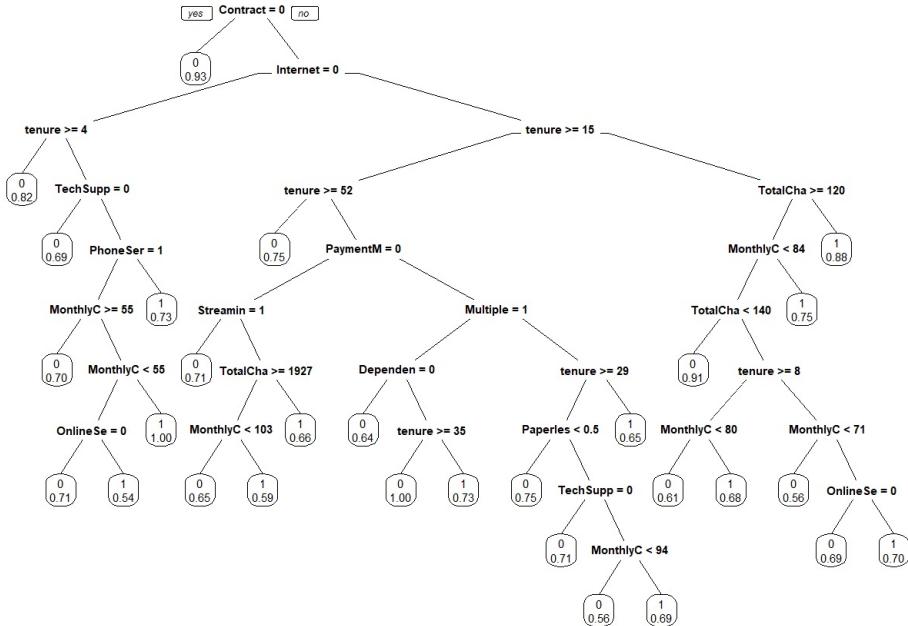
```
#Affichage du cp optimal
print(class.churn$cptable[which.min(class.churn$cptable[,4]),1])
```

```
## [1] 0.002710027
```

```
#Elagage de l'arbre avec le CP optimal
set.seed(496)
class.churn_Opt <- prune(class.churn, cp=class.churn$cptable[which.min(class.churn$cptable[,4]),1])
```

Il ne nous reste maintenant plus qu'à représenter ce nouvel arbre avec la complexité optimale. L'arbre reste encore assez compliqué à lire sur la console R mais les règles de construction de l'arbre sont, elles, beaucoup plus simples, chaque ligne y représente une feuille. On peut voir qu'à la première étape, l'algorithme prend tous les individus et recherche la variables qui sépare le mieux les individus en 2 populations. La variable est sélectionnée selon le critère de l'indice de Gini. Plus celui-ci sera élevé, mieux le découpage sera. Chacun de ces découpages conduit à diviser l'échantillon en deux populations de part leurs facteurs discriminants les plus marquants.

```
set.seed(496)
#Représentation graphique de l'arbre optimal
prp(class.churn_Opt, extra=8)
```



```
#affichage des regles de construction de l'arbre
print(class.churn_Opt)
```

```

## n= 5635
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 5635 1476 0 (0.73806566 0.26193434)
##    2) Contract.Month.to.month< 0.5 2530 167 0 (0.93399209 0.06600791) *
##    3) Contract.Month.to.month>=0.5 3105 1309 0 (0.57842190 0.42157810)
##      6) InternetService.Fiber.optic< 0.5 1403 378 0 (0.73057733 0.26942267)
##        12) tenure>=3.5 936 172 0 (0.81623932 0.18376068) *
##        13) tenure< 3.5 467 206 0 (0.55888651 0.44111349)
##          26) TechSupport.No< 0.5 224 70 0 (0.68750000 0.31250000) *
##          27) TechSupport.No>=0.5 243 107 1 (0.44032922 0.55967078)
##            54) PhoneService.Yes>=0.5 170 83 0 (0.51176471 0.48823529)
##              108) MonthlyCharges>=55.225 27 8 0 (0.70370370 0.29629630) *
##              109) MonthlyCharges< 55.225 143 68 1 (0.47552448 0.52447552)
##                218) MonthlyCharges< 54.675 137 68 1 (0.49635036 0.50364964)
##                  436) OnlineSecurity.No< 0.5 21 6 0 (0.71428571 0.28571429) *
##                  437) OnlineSecurity.No>=0.5 116 53 1 (0.45689655 0.54310345) *
##                    219) MonthlyCharges>=54.675 6 0 1 (0.00000000 1.00000000) *
##                    55) PhoneService.Yes< 0.5 73 20 1 (0.27397260 0.72602740) *
##      7) InternetService.Fiber.optic>=0.5 1702 771 1 (0.45299647 0.54700353)
##        14) tenure>=14.5 908 376 0 (0.58590308 0.41409692)
##          28) tenure>=51.5 166 41 0 (0.75301205 0.24698795) *
##          29) tenure< 51.5 742 335 0 (0.54851752 0.45148248)
##            58) PaymentMethod.Electronic.check< 0.5 332 124 0 (0.62650602 0.37349398)
##              116) StreamingMovies.No>=0.5 170 49 0 (0.71176471 0.28823529) *
##              117) StreamingMovies.No< 0.5 162 75 0 (0.53703704 0.46296296)
##                234) TotalCharges>=1926.9 133 56 0 (0.57894737 0.42105263)
##                  468) MonthlyCharges< 102.725 96 34 0 (0.64583333 0.35416667) *
##                  469) MonthlyCharges>=102.725 37 15 1 (0.40540541 0.59459459) *
##                    235) TotalCharges< 1926.9 29 10 1 (0.34482759 0.65517241) *
##                    59) PaymentMethod.Electronic.check>=0.5 410 199 1 (0.48536585 0.51463415)
##                      118) MultipleLines.No>=0.5 121 50 0 (0.58677686 0.41322314)
##                        236) Dependents< 0.5 94 34 0 (0.63829787 0.36170213) *
##                        237) Dependents>=0.5 27 11 1 (0.40740741 0.59259259)
##                          474) tenure>=35 5 0 0 (1.00000000 0.00000000) *
##                          475) tenure< 35 22 6 1 (0.27272727 0.72727273) *
##                        119) MultipleLines.No< 0.5 289 128 1 (0.44290657 0.55709343)
##                          238) tenure>=28.5 165 80 0 (0.51515152 0.48484848)
##                            476) PaperlessBilling< 0.5 24 6 0 (0.75000000 0.25000000) *
##                            477) PaperlessBilling>=0.5 141 67 1 (0.47517730 0.52482270)
##                              954) TechSupport.No< 0.5 31 9 0 (0.70967742 0.29032258) *
##                                955) TechSupport.No>=0.5 110 45 1 (0.40909091 0.59090909)
##                                  1910) MonthlyCharges< 94 45 20 0 (0.55555556 0.44444444) *
##                                  1911) MonthlyCharges>=94 65 20 1 (0.30769231 0.69230769) *
##                                    239) tenure< 28.5 124 43 1 (0.34677419 0.65322581) *
##      15) tenure< 14.5 794 239 1 (0.30100756 0.69899244)
##        30) TotalCharges>=120 592 214 1 (0.36148649 0.63851351)
##          60) MonthlyCharges< 84.375 326 148 1 (0.45398773 0.54601227)
##          120) TotalCharges< 140.4 11 1 0 (0.90909091 0.09090909) *
##            121) TotalCharges>=140.4 315 138 1 (0.43809524 0.56190476)
##              242) tenure>=7.5 113 52 0 (0.53982301 0.46017699)
##                484) MonthlyCharges< 80.075 85 33 0 (0.61176471 0.38823529) *
##                485) MonthlyCharges>=80.075 28 9 1 (0.32142857 0.67857143) *
##                  243) tenure< 7.5 202 77 1 (0.38118812 0.61881188)
##                    486) MonthlyCharges< 70.675 43 19 0 (0.55813953 0.44186047) *
##                    487) MonthlyCharges>=70.675 159 53 1 (0.33333333 0.66666667)
##                      974) OnlineSecurity.No< 0.5 13 4 0 (0.69230769 0.30769231) *
##                      975) OnlineSecurity.No>=0.5 146 44 1 (0.30136986 0.69863014) *
##                        61) MonthlyCharges>=84.375 266 66 1 (0.24812030 0.75187970) *
##                          31) TotalCharges< 120 202 25 1 (0.12376238 0.87623762) *

```

Nous pouvons voir sur cet arbre le cheminement des individus et la manière dont l'algorithme prédit. En effet, à chaque feuille de cet arbre, on peut voir 2 chiffres. 0 ou 1 pour dire de quel type sera l'individus qui se retrouvera à ce niveau de l'arbre, et la proportion associée de cette classe. Par exemple, si on prend les deux feuilles à n'importe quelle extrémité de l'arbre, si l'individu respecte la condition de la variable il est marqué comme non churner et se retrouve classé dans la feuille de gauche avec le 0 indiquant son état. A contrario s'il ne respecte pas la condition il est marqué comme churner et se retrouve donc sur la feuille de droite avec un "1" symbolisant son churn. En dessous de cette modalité binaire, on retrouve aussi la proportion de personne prédite justement.

Nous allons maintenant observer la qualité de prédiction de ce modèle:

```

#Prediction du modèle sur les données test
class.churn.test_Predict <- predict(class.churn_Opt, newdata=test_val, type = "class")
confusionMatrix(class.churn.test_Predict,y_test)

```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction  0   1
##           0 920 220
##           1  95 173
##
##                   Accuracy : 0.7763
##                   95% CI : (0.7536, 0.7978)
## No Information Rate : 0.7209
## P-Value [Acc > NIR] : 1.275e-06
##
##                   Kappa : 0.384
##
## McNemar's Test P-Value : 2.816e-12
##
##                   Sensitivity : 0.9064
##                   Specificity : 0.4402
##                   Pos Pred Value : 0.8070
##                   Neg Pred Value : 0.6455
##                   Prevalence : 0.7209
##                   Detection Rate : 0.6534
## Detection Prevalence : 0.8097
## Balanced Accuracy : 0.6733
##
## 'Positive' Class : 0
##

```

```

#erruer de classement
mc <- table(test_val$Churn, class.churn.test_Predict)
erreur.classement <- 1.0-(mc[1,1]+mc[2,2])/ sum(mc)
print(erreur.classement)

```

```
## [1] 0.2237216
```

```
#ou encore
1-(945+176)/(945+182+105+176)
```

```
## [1] 0.2038352
```

```

#taux de prediction (VN/ VN+FP) (spécificité = capacité du modèle à détecter les négatifs = ceux qui vont survivre )
prediction=mc[2,2]/sum(mc[2,])
print(prediction)

```

```
## [1] 0.4402036
```

```
(176)/(176+182)
```

```
## [1] 0.4916201
```

L'erreur de classement nous donne le pourcentage d'erreur du modèle. Nous avons vu dans le cours cette formule mais elle peut être réécrite ainsi: $1 - (\text{VP} + \text{VN})/\text{Total}$

Nous obtenons comme résultat: une Accuracy très bonne qui représente le pourcentage de client bien classés. Attention, ce chiffre est à prendre avec des pincettes car, étant donné le déséquilibrage des données que nous avons (80% non churners vs 20% churners) il se peut que le modèle classe tous les individus en non churners et qu'il se trompe ainsi sur tous les churners en ayant une bonne AUC. Ce modèle serait dès lors mauvais et inutile. Il nous faudra alors regarder d'autres indicateurs comme la matrice de confusion ou on peut constater que le modèle a capté autour de 50% de churners. Ce qui reste un bon score. Cet indicateur correspond au taux de prediction et est calculé sur : le nombre de chunner prédit / le nombre de chunner observé

Nous allons maintenant essayer d'améliorer notre résultat avec des modèles plus élaborés

Random Forest de Breiman

Il s'agit d'un algorithme qui effectue un apprentissage en parallèle sur de multiples arbres de décision construits aléatoirement et entraînés sur des sous-ensembles de données différents. Le nombre idéal d'arbres, qui peut aller jusqu'à plusieurs centaines voire plus, est un paramètre important : il est très variable et dépend du problème. Concrètement, chaque arbre de la forêt aléatoire est entraîné sur un sous ensemble aléatoire de données selon le principe du bagging, avec un sous ensemble aléatoire de features (caractéristiques variables des données) selon le principe des « projections aléatoires ». Les prédictions sont ensuite moyennées lorsque les données sont quantitatives ou utilisées pour un vote pour des données qualitatives, dans le cas des arbres de classification. L'algorithme des forêts aléatoires est connu pour être un des

classificateurs les plus efficaces « out-of-the-box » (c'est-à-dire nécessitant peu de prétraitement des données). Il a été utilisé dans de nombreuses applications, y compris grand public, comme pour la classification d'images de la caméra de console de jeu Kinect* dans le but d'identifier des positions du corps.

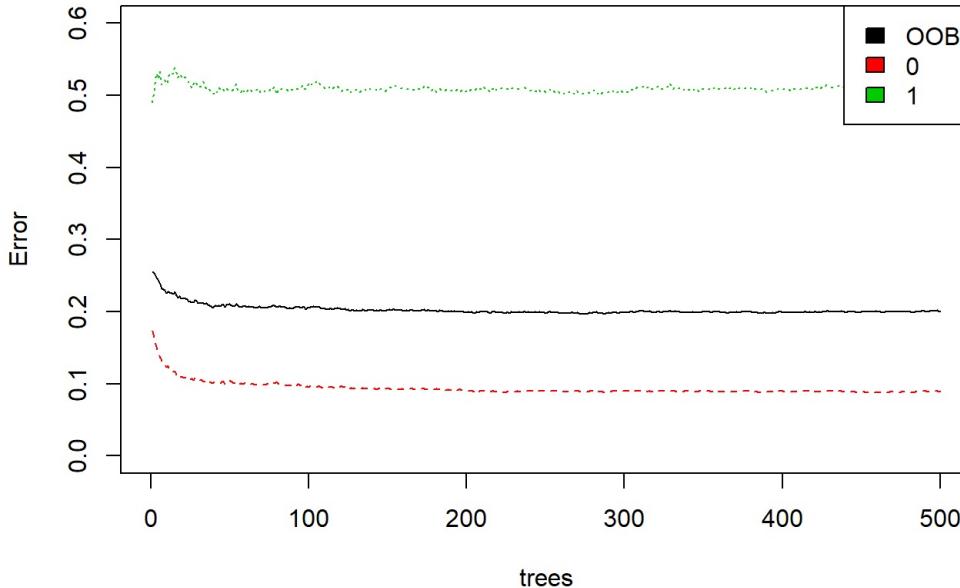
Nous testons tous d'abord un randomForest avec des paramètres basiques prédéfinis dans la fonction.

```
set.seed(496)
#Entraînement du modèle
rf.1 <- randomForest(x = X_train,y=y_train, importance = FALSE, ntree = 500)
rf.1
```

```
##
## Call:
##   randomForest(x = X_train, y = y_train, ntree = 500, importance = FALSE)
##   Type of random forest: classification
##   Number of trees: 500
##   No. of variables tried at each split: 6
##
##       OOB estimate of error rate: 20%
## Confusion matrix:
##      0 1 class.error
## 0 3791 368 0.08848281
## 1 759 717 0.51422764
```

```
#graph permettant de visualiser le taux de mauvais classement en fonction du nombre d'arbre
plot(rf.1, ylim=c(0,0.6))
legend('topright', colnames(rf.1$err.rate), col=1:3, fill=1:3)
```

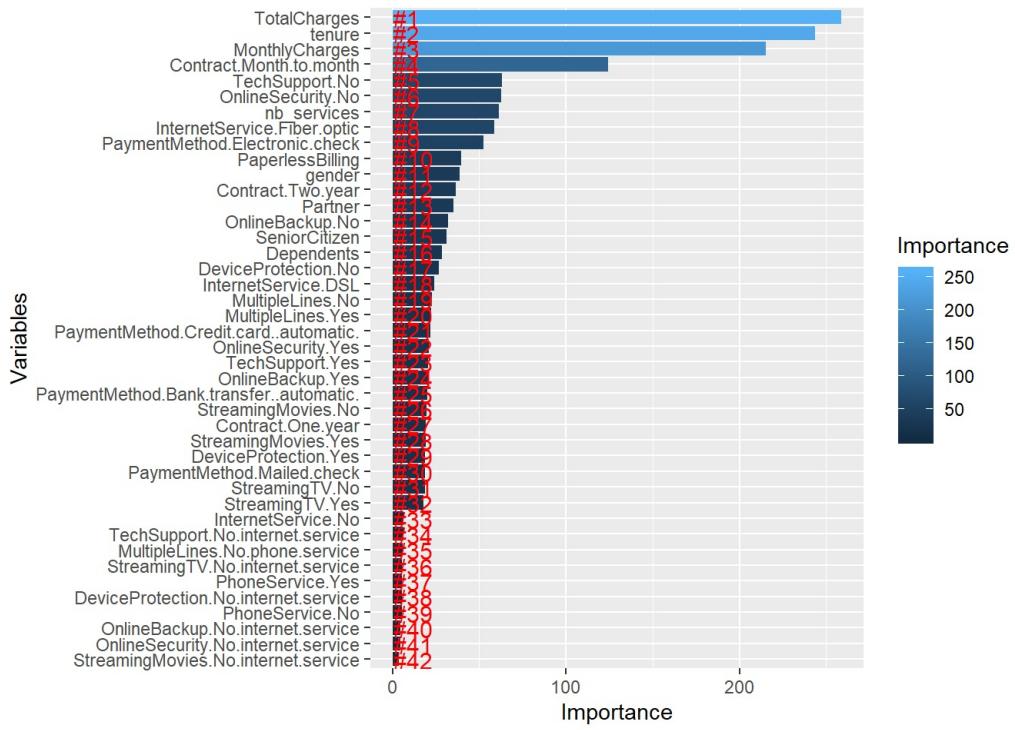
rf.1



```
importance<- importance(rf.1)
#Attribution de l'importance des variables par Moyenne decroissante du coefficient de Gini
varImportance <- data.frame(Variables = row.names(importance),Importance = round(importance[, 'MeanDecreaseGini'],2))

# Classement des variables par niveau d'importance
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))

# Graph permettant d'évisualiser l'importance des variables
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance, fill = Importance)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = Variables, y = 0.5, label = Rank),
            hjust=0, vjust=0.55, size = 4, colour = 'red') +
  labs(x = 'Variables') +
  coord_flip()
```



```
#Résultats
pred=predict(rf.1,newdata = X_test)
confusionMatrix(pred,y_test)
```

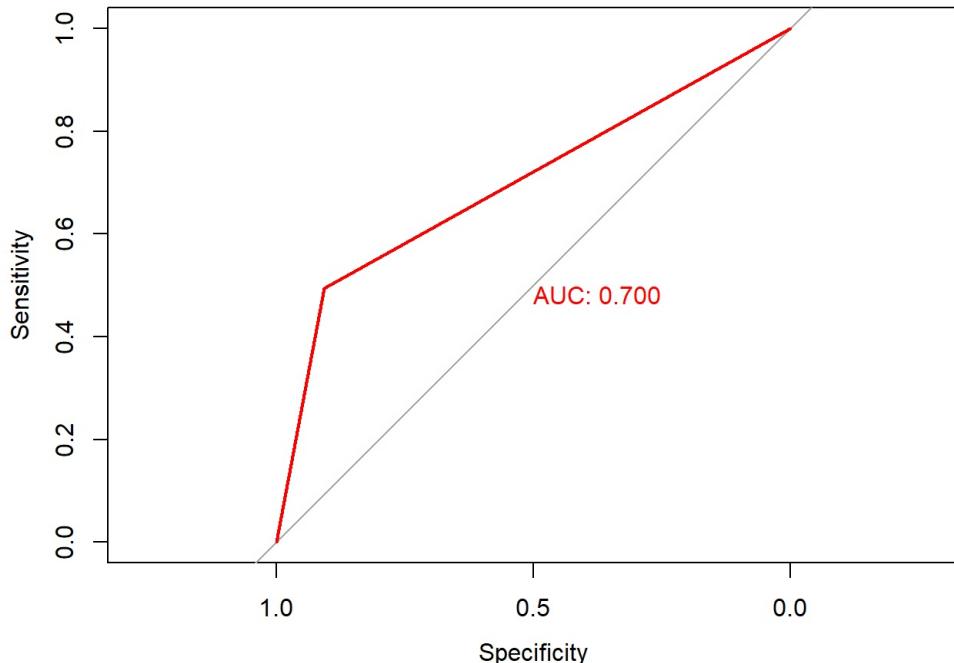
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0    1
##           0 920 199
##           1  95 194
##
##                 Accuracy : 0.7912
##                 95% CI : (0.769, 0.8122)
##     No Information Rate : 0.7209
##     P-Value [Acc > NIR] : 8.631e-10
##
##                 Kappa : 0.4353
##
##     Mcnemar's Test P-Value : 1.889e-09
##
##                 Sensitivity : 0.9064
##                 Specificity : 0.4936
##     Pos Pred Value : 0.8222
##     Neg Pred Value : 0.6713
##     Prevalence : 0.7209
##     Detection Rate : 0.6534
## Detection Prevalence : 0.7947
##     Balanced Accuracy : 0.7000
##
##     'Positive' Class : 0
##
```

```
rf.roc <- roc(response = y_test, predictor = as.numeric(pred))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(rf.roc, col = "red", print.auc = TRUE)
```



```
187/(171+187)
```

```
## [1] 0.5223464
```

Observations: Dans ce premier randomFOrst réalisé, nous pouvons voir quelques éléments importants: Le premier graph nous montre le taux d'erreur du modèle entraîné en fonction du nombre d'arbre sélectionné. Nous pouvons voir que dans notre cas précis, augmenter de beaucoup les arbres ne serait pas très efficace car cela ne nous ferait pas gagner plus de précision (diminution du taux d'erreur). Nous pouvons voir qu'à partir de 150 arbres le modèle ne s'améliore plus. Il ne sert donc à rien d'augmenter notre nombre d'arbre car cela serait gaspiller du temps inutilement et contribuer à du sur-apprentissage. Nous avons aussi à disposition un classement de variables par importance soit par décroissance de l'accuracy contribuée moyenne soit part par gini décroissant. Attardons nous sur le critère Gini, on peut y voir que les différentes variables n'ont pas la même importance dans le modèle. Certaines mêmes sont inutiles. Nous allons donc reproduire le même RandomFOrst en gardant nos variables majeures. Cela aura pour avantage de garder une bonne prédiction tout en limitant la durée du calcul.

Résultat: A la vue de la matrice de confusion , le modèle prédit bien les churners même si cela reste en dessous de nos espérances,nous prédisons tout de même mieux que l'arbre de décision et la détection de Churner est autour de 50%. Nous gagnons 3 points de pourcentage par rapport à l'arbre. L'AUC quant à elle est aussi bonne à 0.8. Cela nous permet donc d'avoir une courbe ROC de bonne qualité et un modèle fonctionnel.

```
#Nouveau randomForest avec les 17 variables les plus importantes précédentes
set.seed(496)
rf.2 <- randomForest(x = X_train2,y=y_train2, importance = TRUE, ntrees = 150)
rf.2
```

```
##
## Call:
##   randomForest(x = X_train2, y = y_train2, ntrees = 150, importance = TRUE)
##   Type of random forest: classification
##   Number of trees: 150
##   No. of variables tried at each split: 4
##
##       OOB estimate of  error rate: 19.7%
## Confusion matrix:
##   0  1 class.error
## 0 3751 408  0.0981005
## 1  702 774  0.4756098
```

```
pred2=predict(rf.2,newdata = X_test2)
confusionMatrix(pred2,y_test2)
```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction  0   1
##           0 912 197
##           1 103 196
##
##                   Accuracy : 0.7869
##                   95% CI : (0.7646, 0.8081)
##       No Information Rate : 0.7209
##   P-Value [Acc > NIR] : 8.351e-09
##
##                   Kappa : 0.4287
##
## Mcnemar's Test P-Value : 7.902e-08
##
##           Sensitivity : 0.8985
##           Specificity : 0.4987
##      Pos Pred Value : 0.8224
##      Neg Pred Value : 0.6555
##          Prevalence : 0.7209
##      Detection Rate : 0.6477
## Detection Prevalence : 0.7876
##   Balanced Accuracy : 0.6986
##
## 'Positive' Class : 0
##

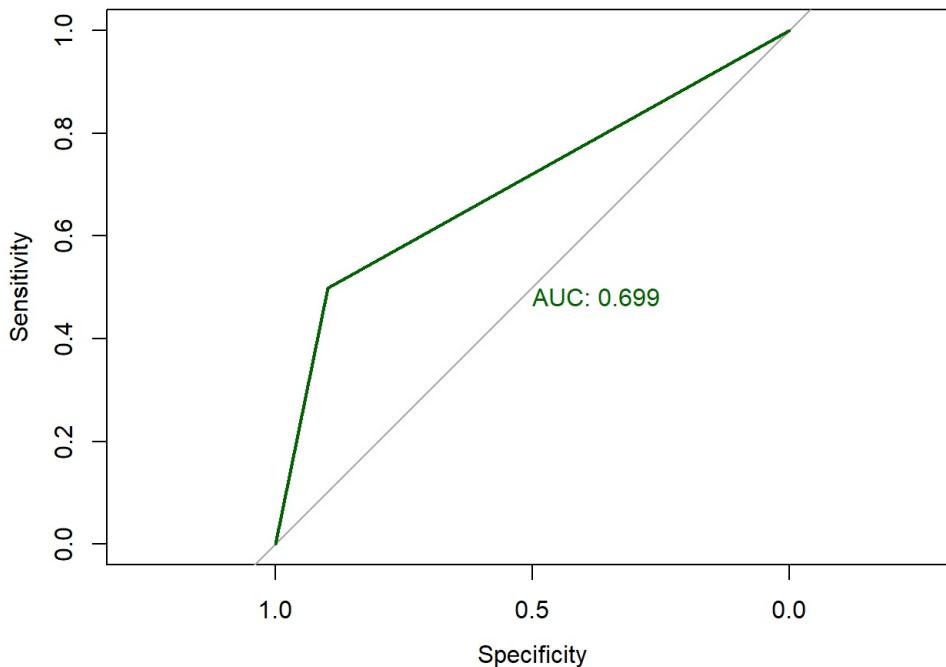
```

```
rf.roc2 <- roc(response = y_test2, predictor = as.numeric(pred2))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(rf.roc2, col = "darkgreen", print.auc = TRUE)
```



L'AUC est moins bonne car on a supprimé de l'information mais moins de temps de calcul. Nous décidons donc de continuer avec l'ensemble des variables étant donné la taille raisonnable de nos observations.

Pour tenter d'améliorer notre modèle, nous décidons d'utiliser de nouveaux paramètres ainsi que la technique de la K-Fold cross-validation qu'on repètera 4 fois afin d'améliorer nos résultats tout en limitant le surapprentissage.

```
library("mlbench")
```

```
## Warning: package 'mlbench' was built under R version 3.6.2
```

```

library("caret")
library("randomForest")
set.seed(496)

#Paramètres Validation croisée
rf_Control <- trainControl(method = "cv", number = 5)

#Paramètre du RandomForest
tunegrid <- expand.grid(.mtry=c(25:42), .ntree=c(50,100,200))

#Customisation de notre RandomForest afin de l'adapter à notre situation avec les paramètres adéquats
customRF <- list(type = "Classification", library = "randomForest", loop = NULL)
customRF$parameters <- data.frame(parameter = c("mtry", "ntree"), class = rep("numeric", 2), label = c("mtry", "ntree"))
customRF$grid <- function(x, y, len = NULL, search = "grid") {}
customRF$fit <- function(x, y, wts, param, lev, last, weights, classProbs, ...) {
  randomForest(x, y, mtry = param$mtry, ntree=param$ntree, ...)
}

customRF$predict <- function(modelFit, newdata, preProc = NULL, submodels = NULL)
  predict(modelFit, newdata)
customRF$prob <- function(modelFit, newdata, preProc = NULL, submodels = NULL)
  predict(modelFit, newdata, type = "prob")
customRF$sort <- function(x) x[order(x[,1]),]
customRF$levels <- function(x) x$classes

#Entraînement du modèle
set.seed(496)
rf.cv<- train(x = X_train, y = y_train, method=customRF, tuneGrid=tunegrid, trControl=rf_Control)
rf.cv

```

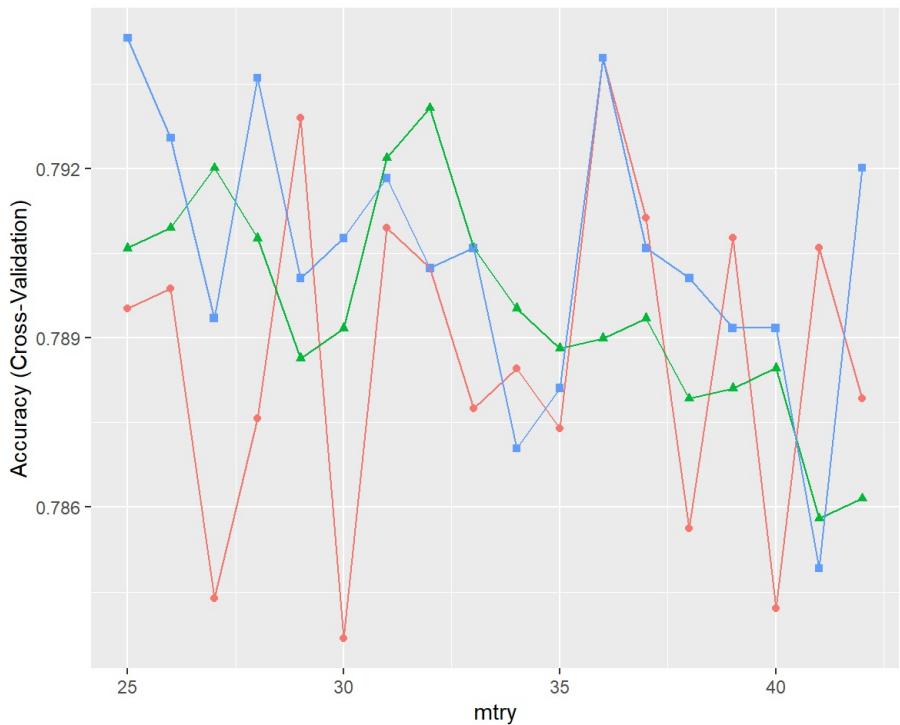
```

## 5635 samples
## 42 predictor
## 2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4508, 4508, 4508, 4507, 4509
## Resampling results across tuning parameters:

##          mtry  ntree Accuracy   Kappa
## 25         50    0.7895267 0.4164230
## 25        100    0.7905932 0.4203842
## 25        200    0.7943175 0.4308142
## 26         50    0.7898827 0.4211204
## 26        100    0.7909498 0.4210666
## 26        200    0.7925467 0.4254105
## 27         50    0.7843820 0.4063012
## 27        100    0.7920133 0.4245237
## 27        200    0.7893519 0.4159135
## 28         50    0.7875776 0.4138774
## 28        100    0.7907698 0.4187330
## 28        200    0.7936097 0.4267533
## 29         50    0.7929000 0.4248369
## 29        100    0.7886428 0.4146484
## 29        200    0.7900600 0.4168069
## 30         50    0.7836712 0.3984382
## 30        100    0.7891750 0.4163800
## 30        200    0.7907708 0.4214734
## 31         50    0.7909484 0.4210790
## 31        100    0.7921922 0.4246377
## 31        200    0.7918346 0.4227586
## 32         50    0.7902395 0.4193451
## 32        100    0.7930772 0.4256015
## 32        200    0.7902405 0.4209317
## 33         50    0.7877528 0.4122680
## 33        100    0.7905932 0.4200189
## 33        200    0.7905951 0.4192948
## 34         50    0.7884630 0.4171686
## 34        100    0.7895290 0.4187987
## 34        200    0.7870444 0.4090977
## 35         50    0.7874004 0.4084109
## 35        100    0.7888204 0.4167704
## 35        200    0.7881096 0.4143337
## 36         50    0.7939653 0.4291036
## 36        100    0.7889979 0.4159550
## 36        200    0.7939651 0.4314040
## 37         50    0.7911246 0.4243099
## 37        100    0.7893517 0.4181157
## 37        200    0.7905936 0.4198211
## 38         50    0.7856277 0.4068885
## 38        100    0.7879319 0.4150467
## 38        200    0.7900616 0.4186277
## 39         50    0.7907738 0.4207567
## 39        100    0.7881109 0.4152793
## 39        200    0.7891754 0.4156612
## 40         50    0.7842082 0.4013411
## 40        100    0.7884668 0.4165739
## 40        200    0.7891757 0.4170021
## 41         50    0.7905935 0.4224019
## 41        100    0.7858034 0.4084589
## 41        200    0.7849158 0.4083128
## 42         50    0.7879317 0.4153872
## 42        100    0.7861587 0.4090695
## 42        200    0.7920137 0.4253619
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were mtry = 25 and ntree = 200.

```

#Evolution de l'accuracy en fonction des différents paramètres sélectionnés
`ggplot(rf.cv)`



```
#Résultats
pred3 <- predict(rf.cv,test_val)
confusionMatrix(pred3,y_test)
```

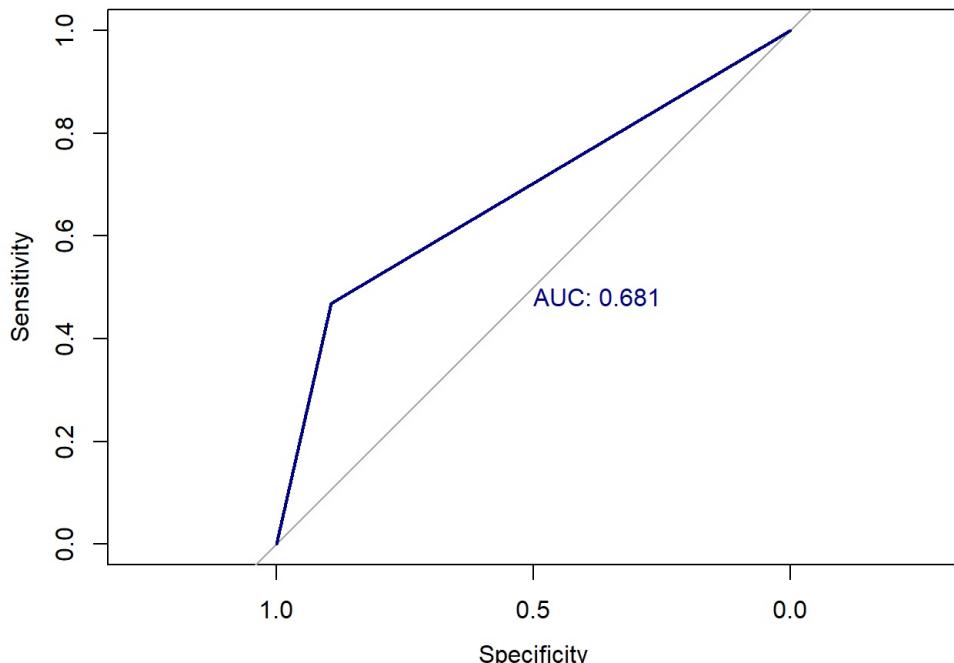
```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction 0 1
##          0 907 209
##          1 108 184
##
##          Accuracy : 0.7749
##          95% CI : (0.7521, 0.7964)
##          No Information Rate : 0.7209
##          P-Value [Acc > NIR] : 2.330e-06
##
##          Kappa : 0.3927
##
##          Mcnemar's Test P-Value : 1.948e-08
##
##          Sensitivity : 0.8936
##          Specificity : 0.4682
##          Pos Pred Value : 0.8127
##          Neg Pred Value : 0.6301
##          Prevalence : 0.7209
##          Detection Rate : 0.6442
##          Detection Prevalence : 0.7926
##          Balanced Accuracy : 0.6809
##
##          'Positive' Class : 0
##
```

```
rf.roc3 <- roc(response = y_test, predictor = as.numeric(pred3))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(rf.roc3, col = "darkblue",print.auc = TRUE)
```



Sur ce dernier modèle de RandomForest, nous avons poussé celui dans l'espoir de le rendre plus robuste et plus efficient. Nous avons d'une part utiliser la méthode de la Cross-Validation, la validation croisée est un moyen de prédire l'efficacité d'un modèle sur un ensemble de validation hypothétique . Il existe 2 méthodes de validation croisée : la K-Fold ainsi que la LOOCV. Nous avons choisi d'utiliser la K-FOLD ici. Nous avons donc découpé notre jeu d'entraînement en 5 échantillons. Quatre sont utilisés pour entraîner le modèle et le dernier est utilisé pour la validation. Cette façon de faire rend notre prédiction sur le test plus robuste et potentiellement avec de meilleurs résultats. Dans le pire des cas on ne peut avoir une AUC qui chute totalement avec cette technique. Nous avons choisi aussi de "tuner" notre modèle avec quelques paramètres tels que le nombre de variables utilisées au hasard à chaque fractionnement(mtry)pouvant aller de 25 à 42 variables. Aussi nous faisons varier le nombre d'arbre à créer par l'algorithme. En gardant toujours à l'esprit qu'un trop grand nombre d'arbre ne contribue pas forcément à augmenter la qualité de la prediction et peu même nous faire tendre vers du sur-apprentissage, tout en consommant beaucoup de temps de calcul. Faire varier le nombre d'arbre à utiliser nous permettra de déterminer environ le nombre d'arbre optimal. On se doute d'avance que ce chiffre sera probablement compris entre 50 et 100 suite à notre premier randomFOrest, où nous avons pu observer que l'erreur ne diminuait plus après la création de 50 arbres, mais dans la logique de notre projet de recherche et dans un soucis de rigoureuxité nous préférons tester notre paramètre mtry avec chacune des itérations du nombre d'arbre. En effet, le modèle est les variables utilisées pouvant différer, le nombre d'arbre nécessaire peut être différent.

Résultat: Sans être exceptionnels, les résultats sont plutôt bon. Cependant on pense qu'on aurait encore eu de meilleurs résultats en faisant varier plus amplement le nombre de variables utilisées dans l'arbre (mtry). De même, les résultats auraient encore été plus robuste et meilleurs, si nous avions effectué une validation croisée répétée n fois (par 5 par exemple)

Gradient Boosting

Les algorithmes de Boosting jouent un rôle crucial dans la conciliation biais variance. Contrairement aux algorithmes de bagging, qui ne contrôlent que les grosses variances dans le modèle, le boosting gère les 2 aspects, et est considéré comme plus efficace. En une explication succincte, le boosting peut-être vu comme une aggrégation d'une succession d'arbres à faible pouvoir prédictif (weak learner) par des poids en fonction de si la prédiction est juste ou non. Ainsi, le boosting est une technique séquentielle qui marche sur le principes des ensembles. Il combine un ensemble de weak learners et offre une précision de prédiction améliorée. A tout instant t, les résultats du modèle sont pondérés en fonction des résultats de l'instant précédent t-1. Les résultats justes ont un poids attribué plus faible que les résultats non justes.

Le descente de gradient est la technique utilisée dans tous les algorithmes de boosting. Le but est d'utiliser l'algorithme afin de déterminer le minimum d'une fonction, la fonction de coût. Nombre de fonctions de coût sont prédéfinies dans les packages, cependant nous avons aussi la possibilité d'en créer une propre à notre besoin. Il s'agit d'une technique robuste qui a largement fait ses preuves. On trouve d'ailleurs beaucoup de variantes permettant encore d'améliorer les résultats. On peut citer les plus connus comme l'Adaboost, le XGboost, le LightBoost, le Catboost.

Pour débuter, étant donné la puissance limitée de nos appareils, nous décidons de lancer une première fois notre algorithme, avec une série de paramètres simples, sur un nombre d'arbre assez large (2000) pour déterminer la fourchette sur laquelle se situerait notre nombre d'arbre optimal en recherchant nos paramètres optimaux. En effet, si par chance nous nous situons autour de 500 arbres pour avoir nos meilleurs résultats prédictifs sur le jeu de test alors il y a peu de chance que notre second modèle varie énormément en fonction des paramètres optimaux. Donc ,nous préférons segmenter notre travail de la sorte.

```
set.seed(496)
#Réalisons tout d'abord un GBM classique afin de voir comment celui-ci évolue.
grad.boost=gbm(Churn ~ . ,data = train_val,distribution = "gaussian",n.trees = 2000,
               shrinkage = 0.01, interaction.depth = 4,cv.folds = 5)
grad.boost
```

```

## gbm(formula = Churn ~ ., distribution = "gaussian", data = train_val,
##       n.trees = 2000, interaction.depth = 4, shrinkage = 0.01,
##       cv.folds = 5)
## A gradient boosted model with gaussian loss function.
## 2000 iterations were performed.
## The best cross-validation iteration was 547.
## There were 42 predictors of which 34 had non-zero influence.

```

```

#Nous créons un vecteur qui fera varier le nombre d'arbre utilisé pour la prédiction sur le test
n.trees = seq(from=100 ,to=2000, by=100)

#Generating a Prediction matrix for each Tree
predmatrix<-predict(grad.boost,test_val,n.trees = n.trees)

#ON calculera ici l'erreur moyenne obtenu par chaque salve d'arbre (ex: erreur moyenne du modèle à 100 arbres vs erreur moyenne du modèle à 8300 arbres)
test.error<-with(test_val,apply( (predmatrix-Churn)^2,2,mean))
head(test.error)

```

```

##      100      200      300      400      500      600
## 0.1570054 0.1471632 0.1441265 0.1428453 0.1424847 0.1423342

```

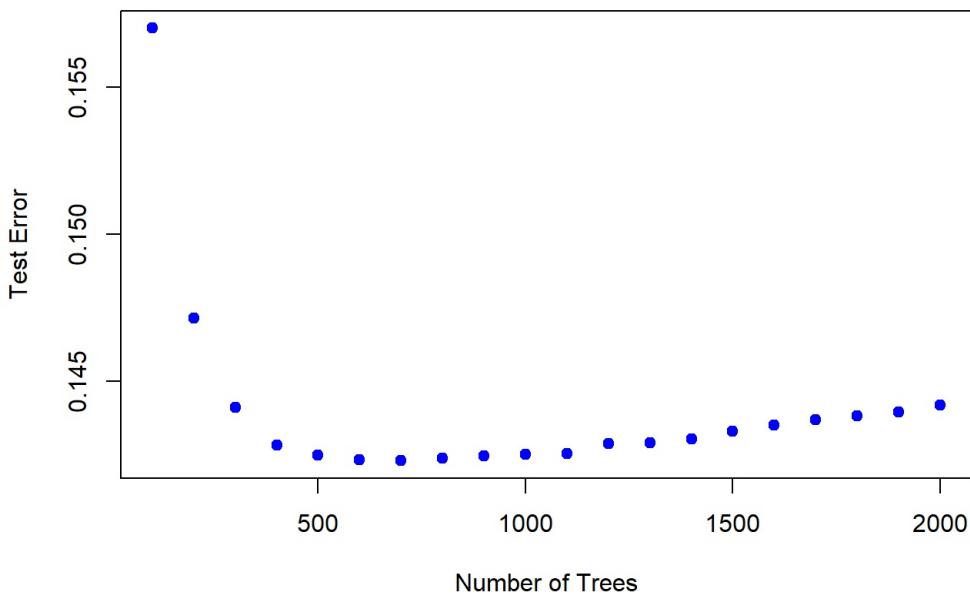
```

a=data.frame(test.error)
#Représentation de l'évolution de ces erreurs moyennes par centaines d'arbres

plot(n.trees , test.error , pch=19,col="blue",xlab="Number of Trees",ylab="Test Error", main = "Performance of Boosting on Test Set")

```

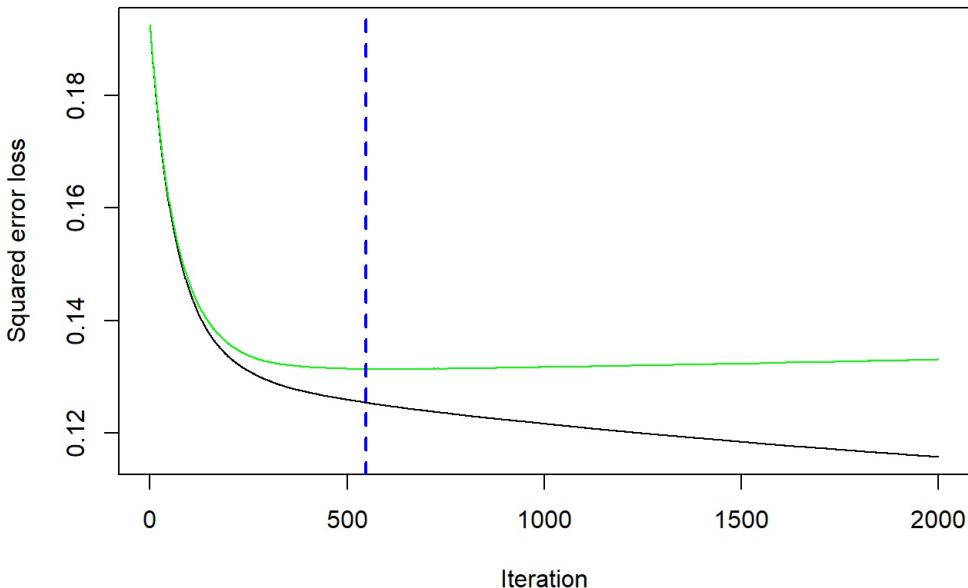
Performance of Boosting on Test Set



```

# Représentation graphique de la loss function loss function sur la table de test (vert) et d'entraînement (noir)
gbm.perf(grad.boost, method = "cv")

```



```
## [1] 547
```

Ici l'importance des variables n'est pas très importante car à chaque arbre est affecté un poids en fonction de si celui-ci prédit bien ou non la target. Nous allons observer plus précisément ici l'évolution de l'erreur sur le jeu de test en fonction du nombre d'arbre utilisé. Nous constatons que le graphique est concave et possède donc un minimum. Ce minimum nous ne pouvons pas le déterminer ici, du fait de l'incrémentation par centaine choisie, mais nous pouvons cependant émettre la conjecture suivante : nous pensons que le nombre d'arbre optimal se situera dans un intervalle $[0;1000]$ d'arbres créés.

Sur le graph du nombre d'itération optimal d'arbre, nous pouvons voir 2 courbes, l'une verte montre la variation de l'erreur du test tandis que l'autre, noire, représente la variation de l'erreur sur les données d'entraînement renforcées par validation croisée. Nous remarquons quelque chose d'intéressant, à partir d'un certain niveau, la courbe verte reste plus ou moins constante tandis que la courbe noir tend inexorablement vers 0. L'interprétation se veut alors des plus simples, plus il y a d'arbre plus l'erreur du jeu d'entraînement est faible car celui tend vers le sur-apprentissage, et n'améliore pas les résultats du jeu de test, et peu même les rendre moins bons voir nuls (on ne le voit pas sur notre graph mais si nous avions mis 10000 itérations la courbe verte deviendrait croissante lorsque l'itération tendrait vers l'infini). Ces deux courbes, l'algorithme nous trace une droite verticale qui nous fournit le nombre optimal d'arbre. C'est le nombre qui minimise l'erreur tout en évitant au mieux le sur-apprentissage. Cette observation concorde donc avec notre premier graph et nous amènera à tester nos paramètres optimaux sur un intervalle de $[0;1000]$ arbres.

```

set.seed(496)
#Déterminons les paramètres optimaux de notre gradient boosting machine

hyper_grid <- expand.grid(
  shrinkage = c(.01, .1, .3),
  interaction.depth = c(1, 3, 5),
  n.minobsinnnode = c(5, 10, 15),
  bag.fraction = c(.65, .8, 1),
  optimal_trees = 0 ,           # paramètre qui sera rempli par la suite une fois le nombre optimal d'arbre déterminé
  min_RMSE = 0                 # de la même façon on cherchera le RMSE le plus petit pour sélectionner nos paramètres
)

# Préparation de nos données d'entraînement conformément à l'algorithme.
train_val_index <- sample(1:nrow(train_val), nrow(train_val))
train_gbm<- train_val[train_val_index, ]

# grid search
for(i in 1:nrow(hyper_grid)) {

  set.seed(496)

  # Entrainement
  gbm.tune <- gbm(
    formula = Churn ~.,
    distribution = "bernoulli",
    data = train_gbm,
    n.trees = 1000,
    interaction.depth = hyper_grid$interaction.depth[i],
    shrinkage = hyper_grid$shrinkage[i],
    n.minobsinnnode = hyper_grid$n.minobsinnnode[i],
    bag.fraction = hyper_grid$bag.fraction[i],
    train.fraction = .75,
    verbose = FALSE
  )

  # Ajout des paramètres signalant l'optimalité
  hyper_grid$optimal_trees[i] <- which.min(gbm.tune$valid.error)
  hyper_grid$min_RMSE[i] <- sqrt(min(gbm.tune$valid.error))
}

# Paramètres optimaux
hyper_grid %>%
  dplyr::arrange(min_RMSE) %>%
  head(5)

```

	shrinkage	interaction.depth	n.minobsinnnode	bag.fraction	optimal_trees
## 1	0.3	3	5	0.80	34
## 2	0.3	1	10	0.65	180
## 3	0.3	1	15	0.65	174
## 4	0.3	3	15	0.80	23
## 5	0.3	3	10	0.80	22
## min_RMSE					
## 1	0.8903593				
## 2	0.8907314				
## 3	0.8908950				
## 4	0.8910015				
## 5	0.8910879				

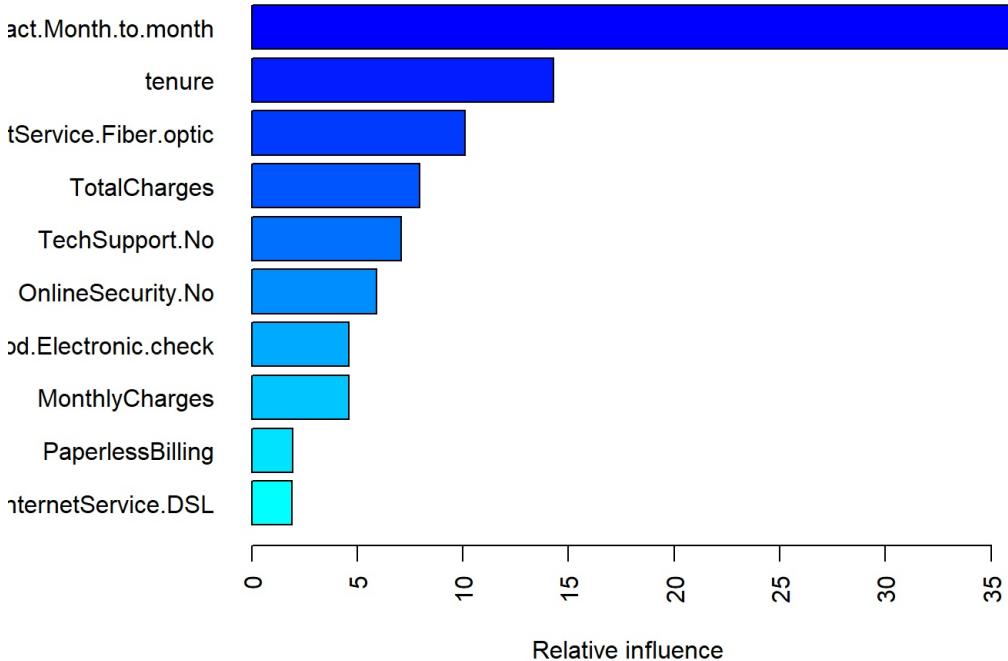
```

hyper_grid=hyper_grid[order(hyper_grid$min_RMSE),]

# Entraînement du GBM avec paramètres optimaux réinsérer directement dans l'algorithme d'entraînement.
gbm.fit.final <- gbm(
  formula = Churn ~ .,
  distribution = "bernoulli",
  data = train_gbm,
  n.trees = hyper_grid[1,5],
  interaction.depth = hyper_grid[1,2],
  shrinkage = hyper_grid[1,1],
  n.minobsinnode = hyper_grid[1,3],
  bag.fraction = hyper_grid[1,4],
  train.fraction = 1,
  verbose = FALSE
)

# Importance des variables
par(mar = c(5, 8, 1, 1))
summary(
  gbm.fit.final,
  cBars = 10,
  method = relative.influence, # also can use permutation.test.gbm
  las = 2
)

```



```

##                                     var
## Contract.Month.to.month          Contract.Month.to.month
## tenure                           tenure
## InternetService.Fiber.optic     InternetService.Fiber.optic
## TotalCharges                     TotalCharges
## TechSupport.No                   TechSupport.No
## OnlineSecurity.No                OnlineSecurity.No
## PaymentMethod.Electronic.check   PaymentMethod.Electronic.check
## MonthlyCharges                   MonthlyCharges
## PaperlessBilling                 PaperlessBilling
## InternetService.DSL              InternetService.DSL
## Contract.Two.year                Contract.Two.year
## nb_services                      nb_services
## MultipleLines.No                 MultipleLines.No
## OnlineBackup.No                  OnlineBackup.No
## SeniorCitizen                    SeniorCitizen
## Dependents                       Dependents
## StreamingTV.Yes                  StreamingTV.Yes
## Contract.One.year                Contract.One.year
## StreamingMovies.Yes               StreamingMovies.Yes
## PaymentMethod.Credit.card..automatic. PaymentMethod.Credit.card..automatic.
## PhoneService.No                  PhoneService.No
## gender                           gender
## Partner                          Partner

```

```

## PhoneService.Yes          PhoneService.Yes
## MultipleLines.No.phone.service  MultipleLines.No.phone.service
## MultipleLines.Yes          MultipleLines.Yes
## InternetService.No        InternetService.No
## OnlineSecurity.No.internet.service  OnlineSecurity.No.internet.service
## OnlineSecurity.Yes         OnlineSecurity.Yes
## OnlineBackup.No.internet.service  OnlineBackup.No.internet.service
## OnlineBackup.Yes           OnlineBackup.Yes
## DeviceProtection.No        DeviceProtection.No
## DeviceProtection.No.internet.service  DeviceProtection.No.internet.service
## DeviceProtection.Yes        DeviceProtection.Yes
## TechSupport.No.internet.service  TechSupport.No.internet.service
## TechSupport.Yes            TechSupport.Yes
## StreamingTV.No             StreamingTV.No
## StreamingTV.No.internet.service  StreamingTV.No.internet.service
## StreamingMovies.No          StreamingMovies.No
## StreamingMovies.No.internet.service  StreamingMovies.No.internet.service
## PaymentMethod.Bank.transfer..automatic. PaymentMethod.Bank.transfer..automatic.
## PaymentMethod.Mailed.check   PaymentMethod.Mailed.check
##                                     rel.inf
## Contract.Month.to.month    36.00337979
## tenure                      14.30913977
## InternetService.Fiber.optic 10.10246681
## TotalCharges                7.95265055
## TechSupport.No              7.06562271
## OnlineSecurity.No           5.90198407
## PaymentMethod.Electronic.check 4.62119786
## MonthlyCharges             4.61565720
## PaperlessBilling            1.95790106
## InternetService.DSL        1.92581462
## Contract.Two.year          1.36933333
## nb_services                 0.95944016
## MultipleLines.No            0.85784997
## OnlineBackup.No             0.58840885
## SeniorCitizen               0.57646933
## Dependents                  0.26957916
## StreamingTV.Yes             0.25680295
## Contract.One.year           0.21968235
## StreamingMovies.Yes          0.15611843
## PaymentMethod.Credit.card..automatic. 0.11339156
## PhoneService.No             0.10474861
## gender                       0.07236084
## Partner                      0.00000000
## PhoneService.Yes             0.00000000
## MultipleLines.No.phone.service 0.00000000
## MultipleLines.Yes            0.00000000
## InternetService.No          0.00000000
## OnlineSecurity.No.internet.service 0.00000000
## OnlineSecurity.Yes           0.00000000
## OnlineBackup.No.internet.service 0.00000000
## OnlineBackup.Yes             0.00000000
## DeviceProtection.No          0.00000000
## DeviceProtection.No.internet.service 0.00000000
## DeviceProtection.Yes         0.00000000
## TechSupport.No.internet.service 0.00000000
## TechSupport.Yes              0.00000000
## StreamingTV.No              0.00000000
## StreamingTV.No.internet.service 0.00000000
## StreamingMovies.No           0.00000000
## StreamingMovies.No.internet.service 0.00000000
## PaymentMethod.Bank.transfer..automatic. 0.00000000
## PaymentMethod.Mailed.check   0.00000000

```

```

#Prediction
predgbm <- predict(gbm.fit.final, n.trees = gbm.fit.final$n.trees, X_test,type="response")
head(predgbm,5)

```

```

## [1] 0.48401835 0.07740953 0.75276774 0.22294812 0.46851831

```

```

predictiongbm=as.factor(ifelse(predgbm>0.51,1,0))

#Results
confusionMatrix(predictiongbm,y_test)

```

```

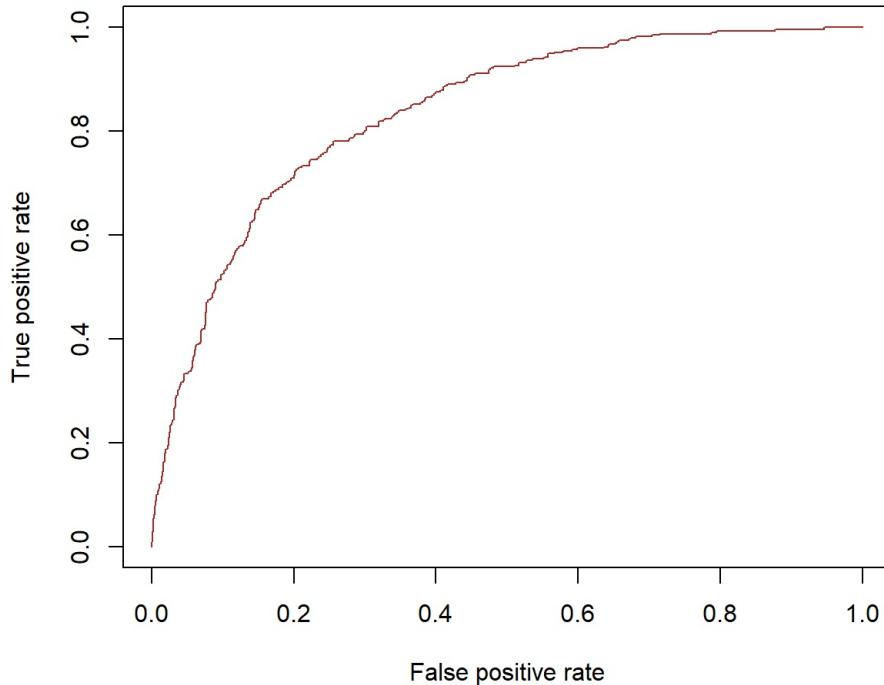
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  0   1
##           0 926 200
##           1  89 193
##
##                   Accuracy : 0.7947
##                   95% CI : (0.7727, 0.8156)
##       No Information Rate : 0.7209
##   P-Value [Acc > NIR] : 1.160e-10
##
##                   Kappa : 0.4416
##
## Mcnemar's Test P-Value : 9.762e-11
##
##             Sensitivity : 0.9123
##             Specificity : 0.4911
##      Pos Pred Value : 0.8224
##      Neg Pred Value : 0.6844
##          Prevalence : 0.7209
##      Detection Rate : 0.6577
## Detection Prevalence : 0.7997
##     Balanced Accuracy : 0.7017
##
## 'Positive' Class : 0
##

```

```

Errors=prediction(predgbm,y_test)
ROC <- performance(Errors,"tpr","fpr")
plot(ROC,col="brown")

```



```

AUC <- performance(Errors,"auc")
AUC<-as.numeric(AUC@y.values)
AUC

```

```

## [1] 0.8347372

```

Après le premier algorithme qui nous a donné une idée assez large du nombre d'arbre où se situerait notre meilleur prédition, nous décidons d'améliorer notre algorithme en lui proposant une liste de différents paramètres qu'il pourra modifier pour trouver la meilleure prédition. Ces paramètres seront sélectionnés sur le critères de la combinaison qui donnera le plus faible RMSE associé.

Voici un petit glossaire des paramètres chalengés:

le Shinkage est l'équivalent du learning rate qui contrôle la descente de radient. Cette valeur est comprise entre 0 et 1 et est en général optimale vers 0,1.

interaction.depth :Le nombre de feuille maximal accepté. Etant donné la théorie du boosting qui veut agréger des prédicteurs faibles, tous les arbres créés ne doivent pas être trop profond. N.B : un profondeur égale à 1 equivaut

en principe à plus ou moins un GLM.

optimal_trees : le nombre d'arbre permettant d'obtenir le meilleur résultat sur ce jeu de données. Il s'agit donc du nombre d'arbre optimal.

RMSE: erreur quadratique moyenne que nous souhaitons minimiser ici. Il s'agit de la racine carrée du carré moyen des erreurs.

n.minobsinnode : Le nombre minimal d'observation dans le dernier noeud de l'arbre.

bag.fraction: part des observations du jeu d'entraînement sélectionné aléatoirement. En temps normal, ce paramètre est fixé à 0.5.

Une fois ces paramètres obtenus nous les reincorporons dans notre algorithme d'entraînement et appliquons notre prédiction sur le jeu de test. Nous obtenons alors des probabilités prédictives de Churn. Nous utilisons alors la fonction ifelse afin de fixer le seuil de probabilité auquel l'individus Churn ou non. Il s'agit de l'algorithme qui nous donne les meilleurs résultats avec une AUC généralement supérieure à 0.8 est un nombre de churning particulièrement bien prédict.

Conclusion

L'apprentissage supervisé semble plus efficace que le non supervisé. De même nous préfèrerons utiliser un gradient boosting pour déterminer nos clients churning. Nous avons pu voir que pour la représentation, il est plus parlant d'utiliser une classification, cependant la régression (comme dans le gradient boosting) pourrait nous permettre de mieux cibler nos churning. En effet, un service commercial pourrait par exemple nous demander de sortir un nombre précis de churning par mois, tout en limitant le nombre de faux pour ne pas passer d'appel inutile par exemple, dans ce cas là on fera varier notre probabilité seuil pour obtenir un nombre de churning avec très peu d'erreurs. Dans ce cas là, nous cherchons utilisons la DataScience comme vecteur de productivité

```
"finish"
```

```
## [1] "finish"
```