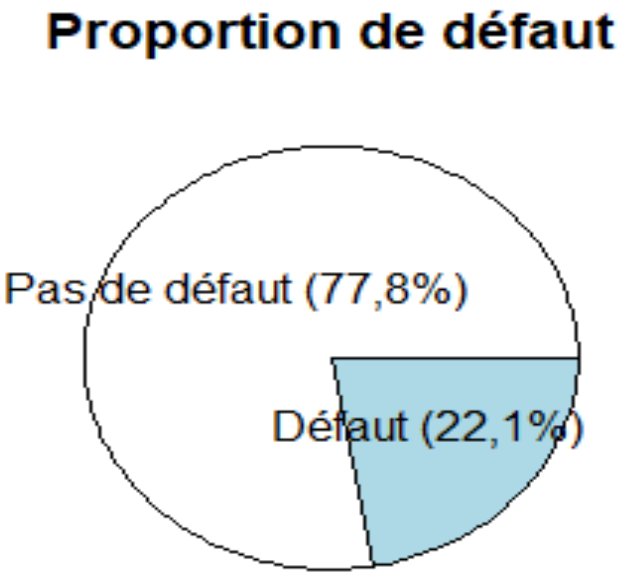
Dumontel Pierre **Magistère 2**
Rouet William **Avril 2021**

Default of credit card clients Data Set

L'objectif de ce travail a été d'utiliser un jeu de données provenant de http://uci.edu/ml/datasets. Dans un premier de temps de le prendre en main et de présenter son contenu. Puis dans un second temps, d'appliquer les méthodes de prédiction vu en cours d'Introduction au Machine Learning (24h). Parmi les méthodes supervisées vu en cours, nous avons appliqué C.A.R.T, K.N.N, Bagging et Random Forest.

I/ Visualisation des données

La variable à prédire



Default payment (Yes = 1, No = 0).

Les variables explicatives

2 principaux types de variables :

1/ Les caractéristiques des personnes

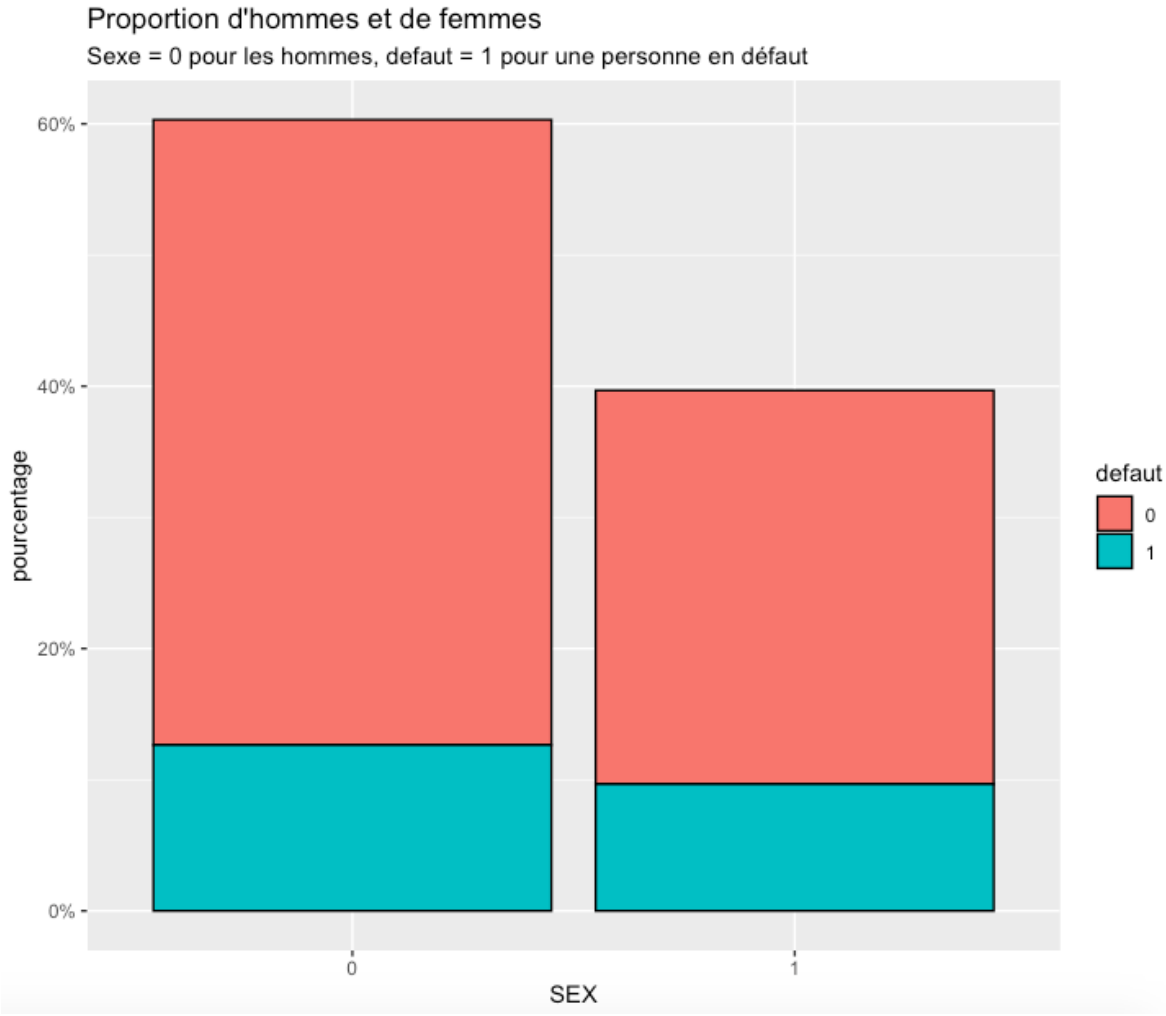
2/ Les historiques bancaires

1/ Les caractéristiques personnelles

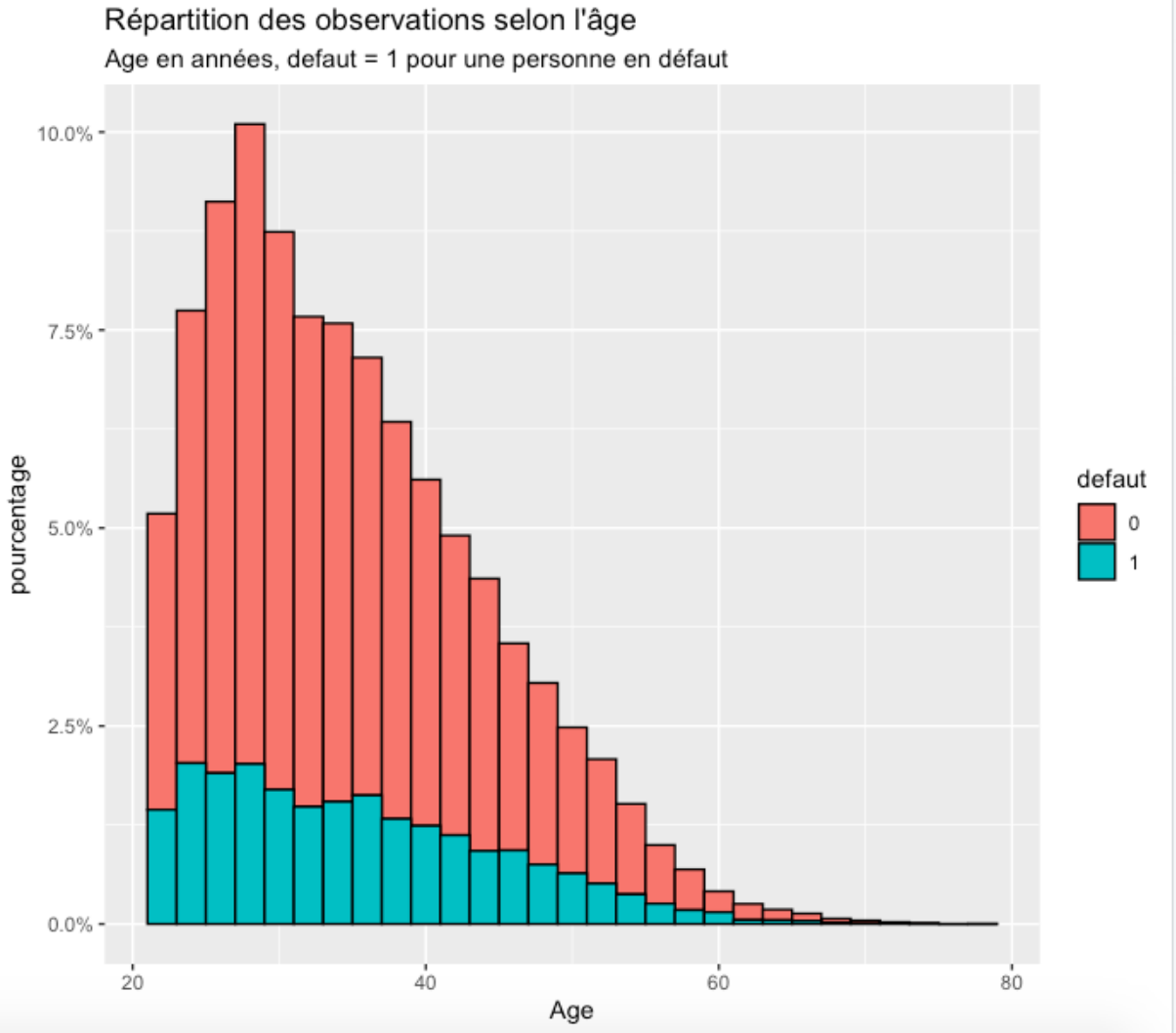
=> Pas de profil type pour une personne en défaut

•Informations standards sur les clients : Sex (1 = homme, 0 = femme), Education (1 = graduate school; 2 = university, 3 = high school, 4 = autres), Statut marital (1= married, 2 = single, 3 = others) et enfin l'âge en année.

•Sexe :



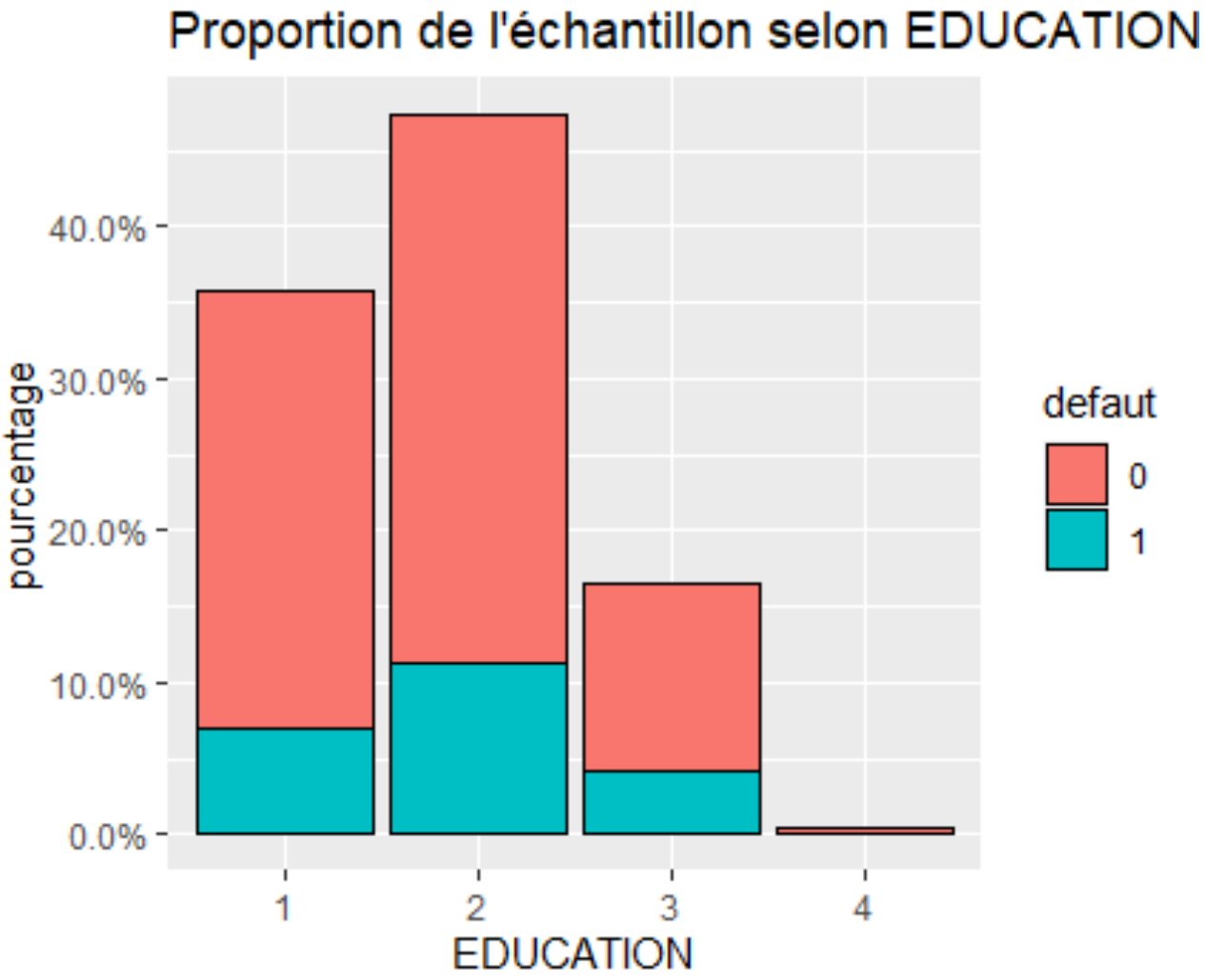
•Âge :



•Education :

Niveau d'éducation	Graduate School (1)	University (2)	High School (3)	Others (4)
Nombre d'obs	10 581 obs	14 024	4 873	123
% en défaut	19%	23%	25%	5%

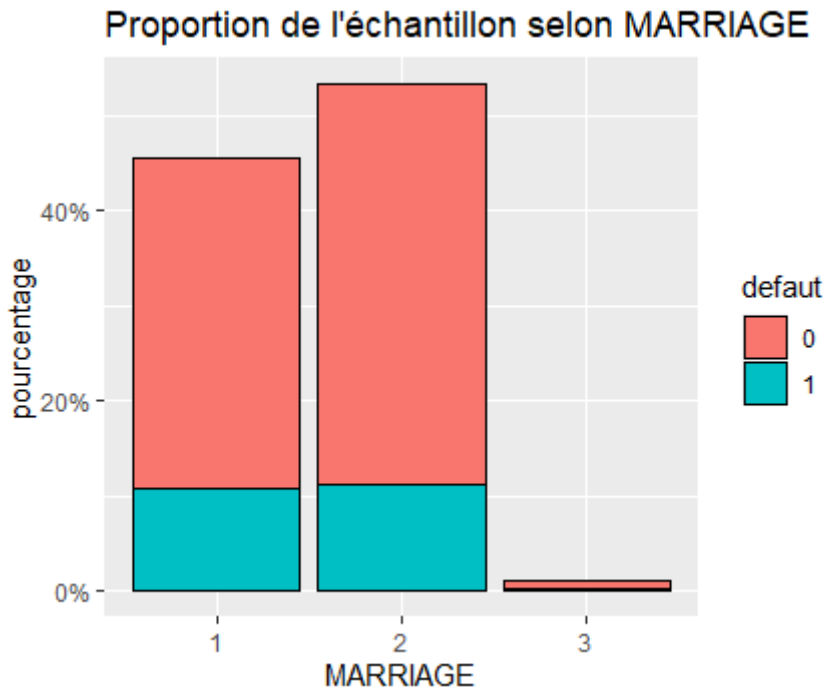
Le groupe des personnes qui n'ont pas fait d'études supérieures est le groupe avec le taux de défaut le plus élevé


Proportion de l'échantillon selon EDUCATION
pourcentage
40.0%
30.0%
20.0%
10.0%
0.0%
1
2
3
4
EDUCATION
defaut
0
1

•Marriage :

MARRIAGE	Married (1)	Single (2)	Others (3)
Population	13 477	15 806	318
% de défaut	24%	21%	26%

Les personnes célibataires ont le taux de défaut le plus faible.

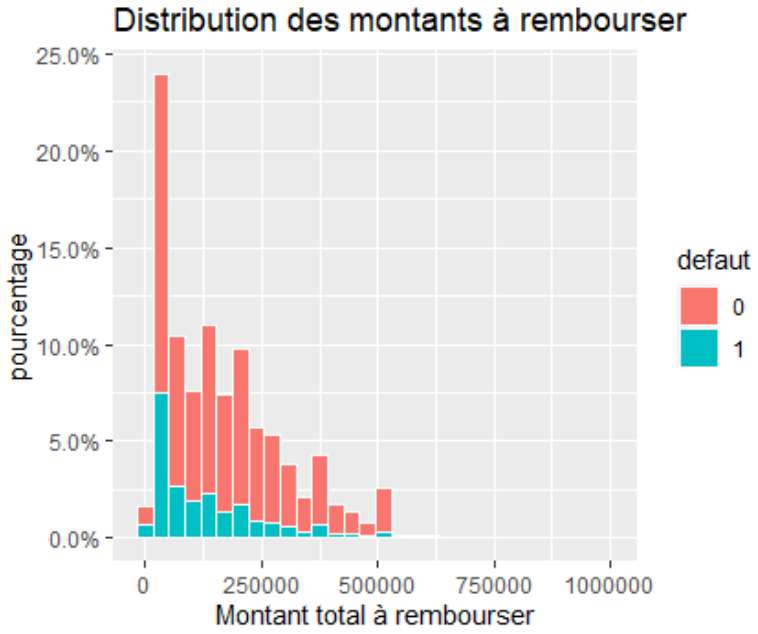


2/ Historiques bancaires

•Montant du crédit initial en NT dollar (LIMIT_BAL)

Min	1st Qu.	Median	Mean	3rd Qu	Max
10 000	50 000	140 000	167 550	240 000	1 000 000

Les montants à rembourser sont très regroupés entre 10 000 et 240 000 Nouveaux dollars de Taïwan (75% de notre échantillon). Pas de proportion de défaut anormale selon la distribution de notre échantillon -> probabilité de défaut et le montant des emprunts initiaux de semblent pas corrélés.



•Historique des remboursements

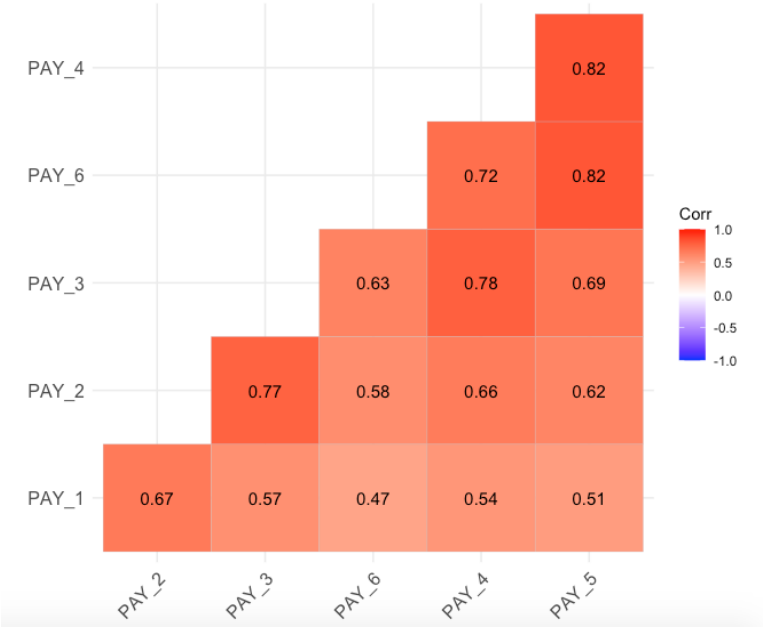
Pay_i pour i entre 0 et 6 : historique des défauts de paiements entre Avril et Septembre 2005. Pay_0 : le statut de remboursement en Septembre 2005 … Pay_6 : le statut de remboursement en Avril 2005.

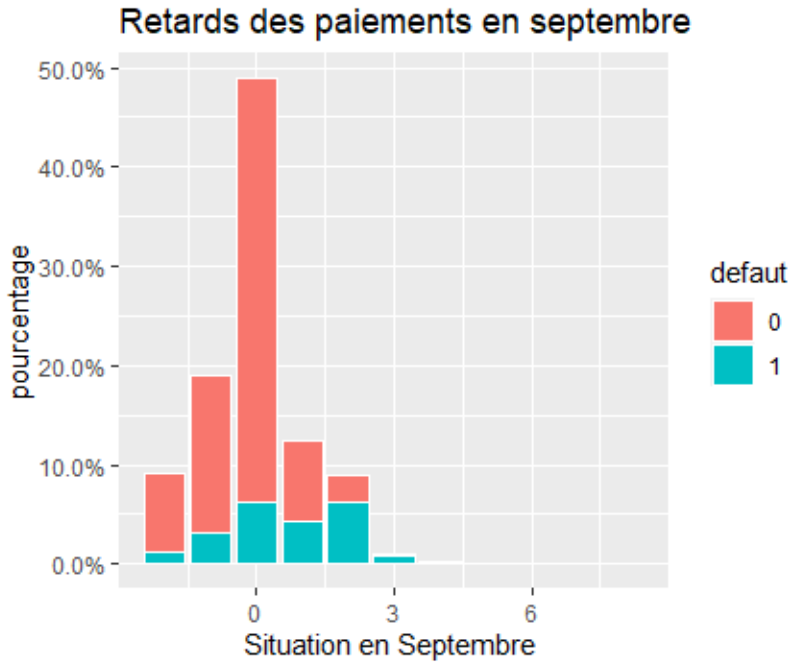
L'échelle de mesure du statut de remboursement est telle que :

-1 -> à jour puis variable prend la valeur du nombre de mois en retard

-1	1	2	3	4	5	6	7	8	9
A jour	retard de 1 mois	retard de 2 mois	retard de 3 mois	retard de 4 mois	retard de 5 mois	retard de 6 mois	retard de 7 mois	retard de 8 mois	retard de 9 mois

Les situations des paiements sont très corrélées d'un mois à l'autre.

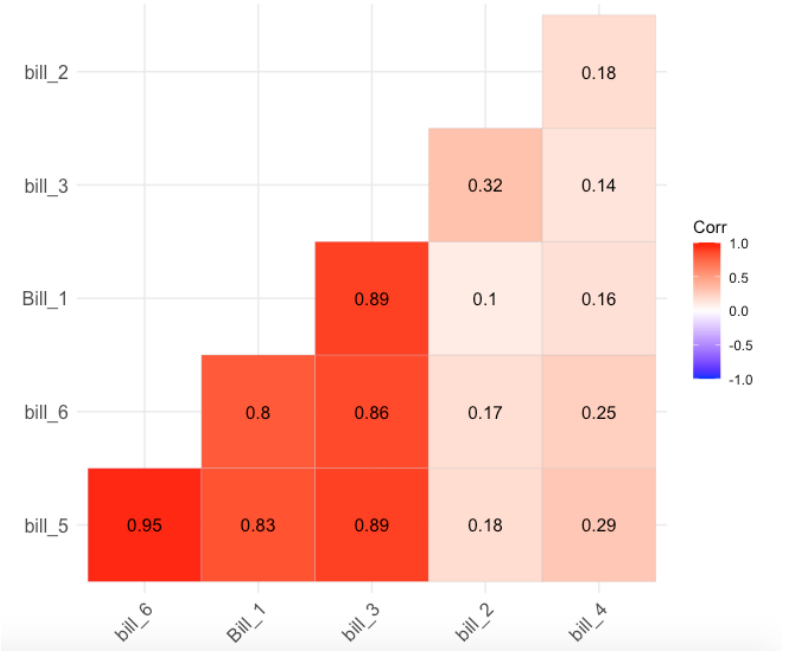




La situation initiale (septembre 2005) est assez stable avec la moitié de notre échantillon à jour dans ses paiements. La proportion d'individus en défaut augmente avec le nombre de mois de retards sur les paiements.

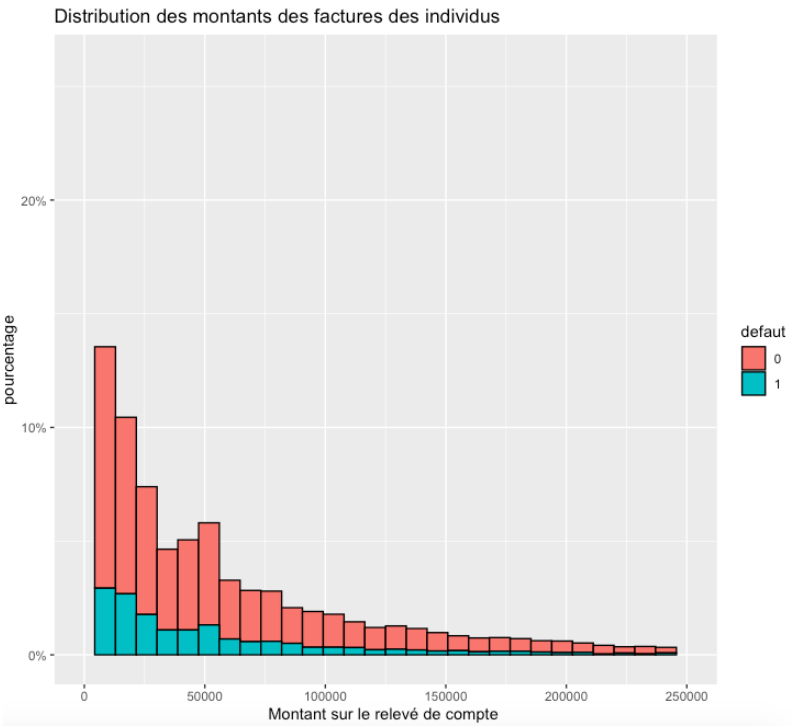
•Montant en dollar NT des dernières factures :

BILL_AMT_i pour i entre {1:6}
Situation entre avril et septembre 2005.



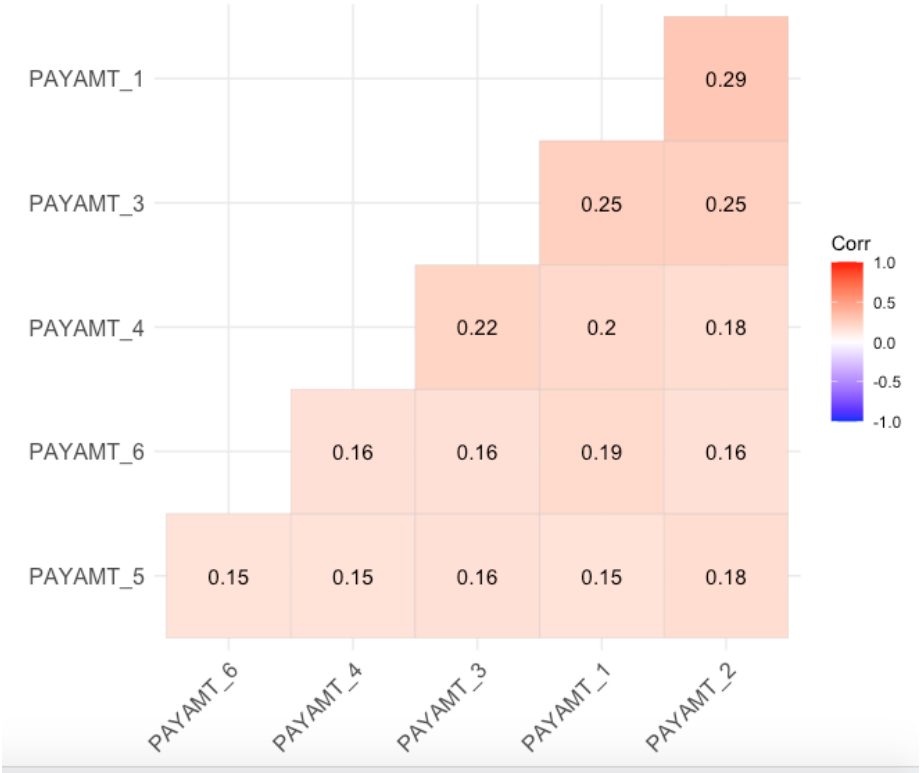
Très corrélés : bill_1; bill_3; bill_6; bill_5.

Peu corrélés : bill_2; bill_4



La majorité de notre échantillon a des montants de factures inférieurs à 100 000 NT dollars. Le montant des dernières factures d'un individu ne semble pas corrélé avec sa probabilité de faire défaut.

•Montant du paiement précédent en NT dollar



PAY_AMT_i pour i entre {1:6}.

Les montants des derniers paiements en date des individus ne semblent pas corrélés entre eux.

II/ Prédiction des personnes qui sont en défaut?

On veut prédire la variable “Défaut de paiement” en fonction de 23 variables explicatives.

On commence par créer un train-test sur le dataset.

```
Ind.test = sample(n,n/3)
Learn = data[-Ind.test,]
Test = data[Ind.test,]
```

```
> dim(Learn)
[1] 19734    24
> dim(Test)
[1] 9867   24
```

Arbres de décisions :

Premier arbre avec les paramètres par défaut de rpart :

1) root 19734 4359 0 (0.79 0.22)
  2) PAY_0={-2,-1,0,1} 17682 2939 (0.83 0.17) *
  3) PAY_0={2,3,4,5,6,7,8} 2052 632 (0.31 0.69) *

Arbre à deux branches divisé selon les valeurs de PAY_0 (situation des paiements en cours en septembre 2005). PAY_i très corrélés influencent beaucoup la probabilité de faire défaut.

```
> prev_arbre <- predict(defaut_Tree,newdata=Test,type="class")
> err_arbre <- sum(prev_arbre!=Test$defaut)/nrow(Test)
> err_arbre
[1] 0.1817168
```

Matrice de Confusion :

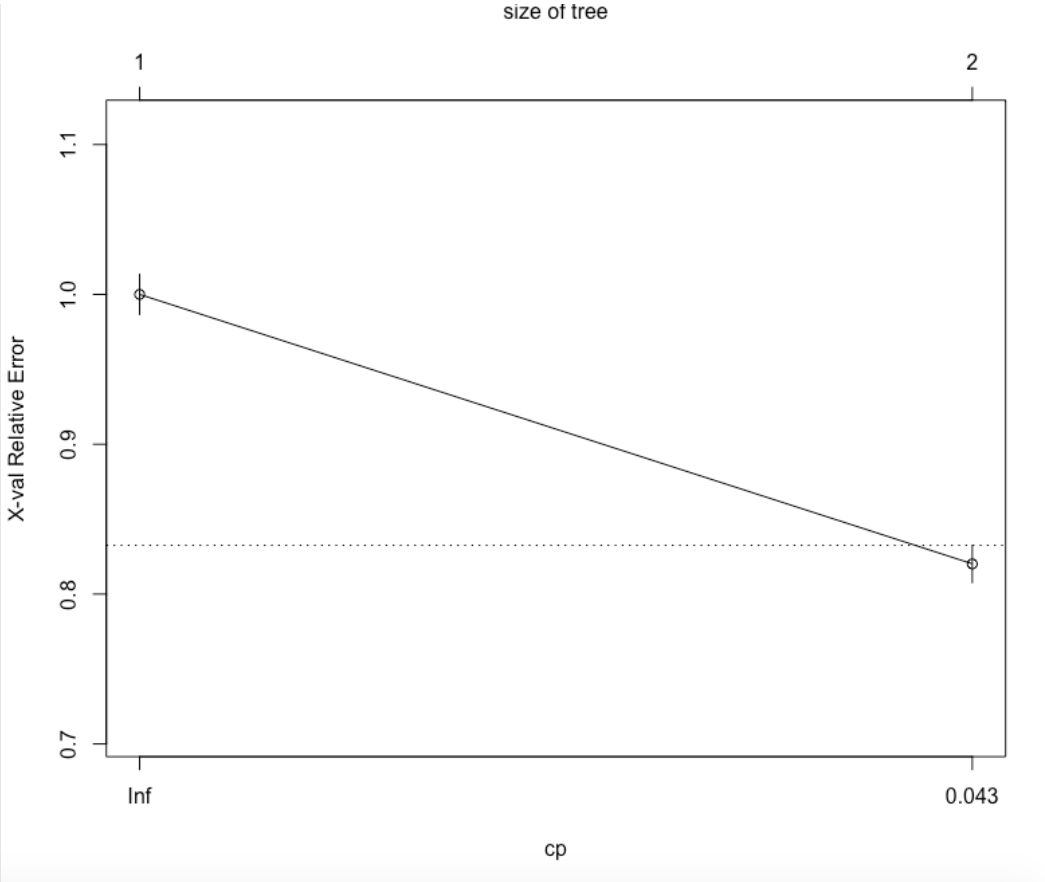
	Non défaut prévu(0)	Défaut prévu (1)
Non défaut (0)	7 324	297
Défaut (1)	1 496	750

Score : 91 %

Simplification de l’arbre :

On procède par validation croisée pour déterminer la qualité de découpe optimale (cp) de l’arbre qui va permettre de minimiser les erreurs de prédiction de notre modèle.

La fonction rpart réalise par défaut une estimation des performances de l'arbre par validation croisée à 10 blocs pour chaque niveau de simplification pertinent.
Performances par validation croisée :

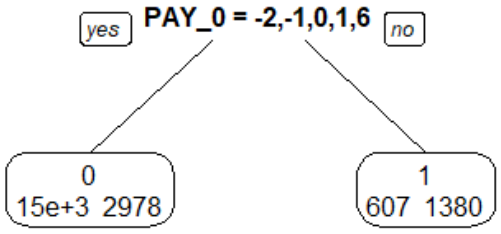


La courbe indique le taux de mauvaises classifications relativement au score d'origine, estimé par la validation croisée. On s’en sert pour calculer un arbre optimal.

Arbre optimal :

cp=defaut_Tree$cptable[which.min(defaut_Tree$cptable[,4]),1] = 0.043
(qualité de découpe optimale de l'arbre)

1) root 19734 4358 0 (0.78 0.22)
  2) PAY_0=-2,-1,0,1,6 17747 2978 (0.83 0.17)
  3) PAY_0=2,3,4,5,7,8 1987 607 (0.31 0.69)



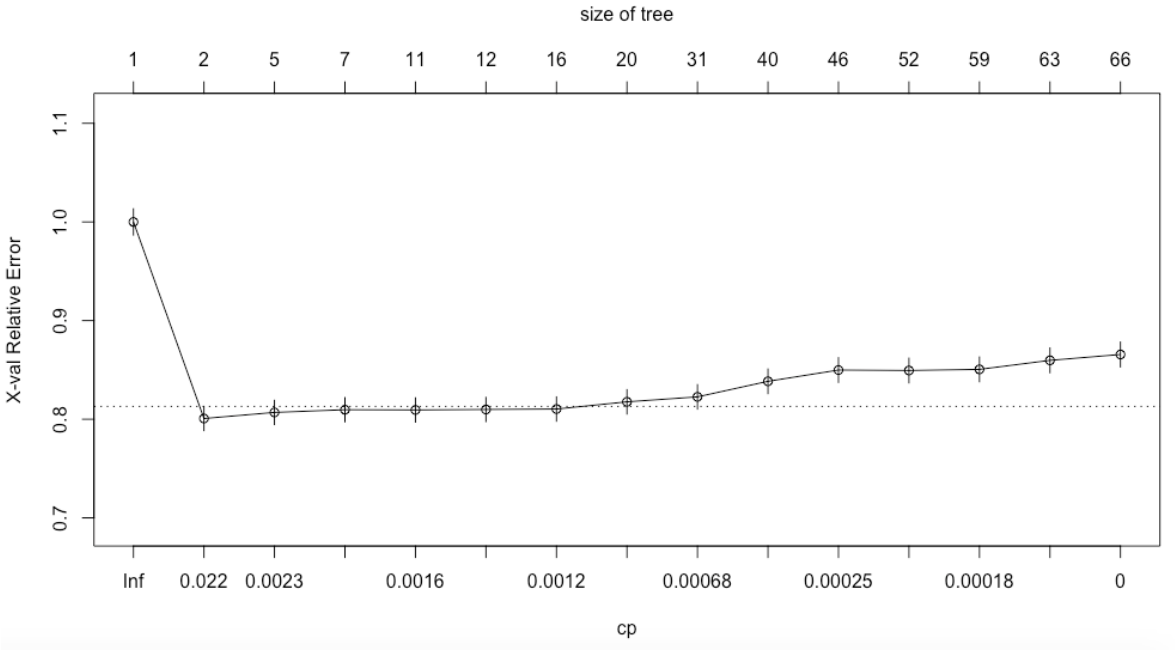
Matrice de confusion :

	Non défaut prévu(0)	Défaut prévu (1)
Non défaut (0)	7 303	317
Défaut (1)	1 463	784

Score = 91%

Augmenter le nombre de branches dans l'arbre :

Utilisation de les hyper paramètres minsplit = 80, pour avoir au moins 80 observations par feuilles et cp = 0 pour ne pas avoir de contraintes d' élagage.

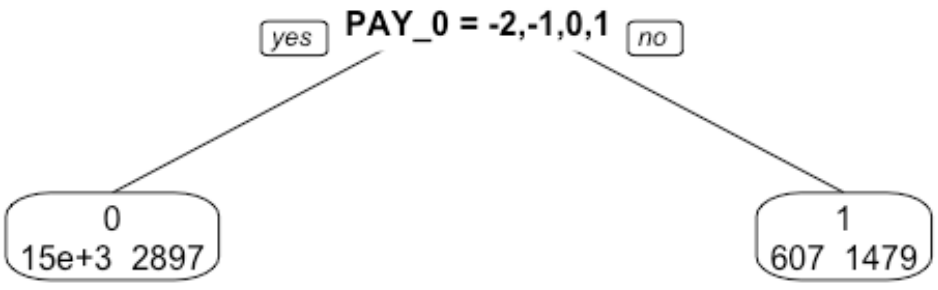


	Non défaut prévu(0)	Défaut prévu (1)
Non défaut (0)	7 131	507
Défaut (1)	1 428	801

Score = 89,9%

L'arbre contient trop de branches pour que rendre le « plot » visible.

On cherche à nouveau la valeur de découpage optimal



	Non défaut prévu(0)	Défaut prévu (1)
Non défaut (0)	7 316	322
Défaut (1)	1 538	691

Score = 91,3%

K plus proches voisins

Etape 1 : Matrice de confusion avec k =1
Erreur = 0.307

	Pas de défaut	Défaut
Pas de défaut	6 157	1 464
Défaut	1 562	684

Score = 90%

Etape 2 : on fait une boucle pour k entre 1 et 10 pour chercher la valeur minimum d'erreur sur l'échantillon d'apprentissage :
Atteint pour k = 10, l'erreur est de 0.237.
Matrice de confusion pour k = 10

	Pas de défaut	Défaut
Pas de défaut	7 208	413
Défaut	1 940	306

Score = 96%

Random Forest

```
Call:
 randomForest(formula = Learn$defaut ~ ., data = Learn)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 18.44%
Confusion matrix:
      0    1 class.error
0 14496  879  0.05717073
1  2759 1600  0.63294334
```

Score = 90%

On utilise la fonction importance( ) de R pour classer les variables selon leur importance dans la classification :

```
[1]  6 12  5 13  1 18 14 15 16 17 19 20 23 21 22  7  8  9 11 10  3  4  2
```

-La variable 6 est la situation de remboursement du dernier mois observé (Pay_0)
-La variable 12 est le montant des factures pour le dernier mois observé (BILL_AMT1)
-Les variables 2,4,3 sont les caractéristiques individuelles.

Et enfin l'erreur sur l'échantillon test : 0.1822236
=> L'estimation OOB est légèrement pessimiste.

Bagging :

Bagging → 1) **bootstrap** (échantillonnage avec remise) => Autant de modèles que d'échantillons sont entrainés

2) **aggregating** (dans le cas de la classification la prédiction du modèle est obtenue par vote de majorité)

Relation décroissante, convexe entre le nombre de réplications et le pourcentage d'erreurs de classification.

Bagging classification trees with **10 bootstrap replications**

Call: bagging.data.frame(formula = defaut ~ ., data = Learn, nbagg = 10,
coob = TRUE, control = rpart.control(minsplit = 2, cp = 0))

Out-of-bag estimate of misclassification error: **0.2361**

****

Bagging classification trees with **50 bootstrap replications**

Call: bagging.data.frame(formula = defaut ~ ., data = Learn, nbagg = 50,
coob = TRUE, control = rpart.control(minsplit = 2, cp = 0))

Out-of-bag estimate of misclassification error: **0.1942**

****

Bagging classification trees with **100 bootstrap replications**

Call: bagging.data.frame(formula = defaut ~ ., data = Learn, nbagg = 100,
coob = TRUE, control = rpart.control(minsplit = 2, cp = 0))

Out-of-bag estimate of misclassification error: **0.1892**

****

Bagging classification trees with **200 bootstrap replications**

Call: bagging.data.frame(formula = defaut ~ ., data = Learn, nbagg = 200,
coob = TRUE, control = rpart.control(minsplit = 2, cp = 0))

Out-of-bag estimate of misclassification error: **0.1866**

Erreur sur l'échantillon test : **0.1817168** → estimation pessimiste

Conclusions sur la partie prédiction :

•On a appliqué plusieurs méthodes de classification (Arbre de décisions, KNN, Bagging et Random Forest).

-Avec toutes les méthodes qu'on a appliqué on a eu une erreur entre 0,18 et 0,25.

-Les matrices de confusion vont dans ce sens, avec un nombre relativement important de personnes faux positif et vrai négatif.

-La méthode qui a le score le plus élevé est le KNN avec un k =10.

•Cela rejoint bien la visualisation des données :

-Pas de profil type sur les caractéristiques personnelles.

-La variable qui reste l'indicateur le plus important c'est les historiques de paiements récents (dernier mois observé) : c'est la variable de segmentation dans l'arbre de décision et la variable classée première dans l'importance de Random Forest.

Et on a vu dans la visualisation des données que c'est celle qui représente le mieux le fait que les personnes en retard de paiements sont celles sujettes à être en défaut.