

# BDA-LD

Benjamin Bernaud

2023-11-27

Notes : On part d'une architecture similaire au dossier vagrant. Il faut définir le chemin du dossier userscripts  
Le dossier BDA-LD doit contenir un dossier data qui contient les fichiers CSV

```
setwd("C:/Users/Berna/Documents/BDA-LD/userscripts")
```

## BDA-LD - Partie Analyse de données

### Chargement des librairies

```
install.packages("fpc", repos = "http://cran.us.r-project.org")
```

```
## Installation du package dans 'C:/Users/Berna/AppData/Local/R/win-library/4.3'  
## (car 'lib' n'est pas spécifié)  
  
## le package 'fpc' a été décompressé et les sommes MD5 ont été vérifiées avec succès  
##  
## Les packages binaires téléchargés sont dans  
## C:\Users\Berna\AppData\Local\Temp\RtmpusIG3Z\downloaded_packages
```

```
install.packages("regclass", repos = "http://cran.us.r-project.org")
```

```
## Installation du package dans 'C:/Users/Berna/AppData/Local/R/win-library/4.3'  
## (car 'lib' n'est pas spécifié)  
  
## le package 'regclass' a été décompressé et les sommes MD5 ont été vérifiées avec succès  
##  
## Les packages binaires téléchargés sont dans  
## C:\Users\Berna\AppData\Local\Temp\RtmpusIG3Z\downloaded_packages
```

```
install.packages("rpart.plot", repos = "http://cran.us.r-project.org")
```

```
## Installation du package dans 'C:/Users/Berna/AppData/Local/R/win-library/4.3'  
## (car 'lib' n'est pas spécifié)  
  
## le package 'rpart.plot' a été décompressé et les sommes MD5 ont été vérifiées avec succès  
##  
## Les packages binaires téléchargés sont dans  
## C:\Users\Berna\AppData\Local\Temp\RtmpusIG3Z\downloaded_packages
```

```
install.packages("C50", repos = "http://cran.us.r-project.org")
```

```
## Installation du package dans 'C:/Users/Berna/AppData/Local/R/win-library/4.3'  
## (car 'lib' n'est pas spécifié)  
  
## le package 'C50' a été décompressé et les sommes MD5 ont été vérifiées avec succès  
## Warning: impossible de supprimer l'installation précédente du package 'C50'
```

```

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problème lors de la
## copie de
## C:\Users\Berna\AppData\Local\R\win-library\4.3\00LOCK\C50\libs\x64\C50.dll vers
## C:\Users\Berna\AppData\Local\R\win-library\4.3\C50\libs\x64\C50.dll: Permission
## denied

## Warning: 'C50' restauré

##
## Les packages binaires téléchargés sont dans
## C:\Users\Berna\AppData\Local\Temp\RtmpusIG3Z\downloaded_packages
install.packages("tree", repos = "http://cran.us.r-project.org")

## Installation du package dans 'C:/Users/Berna/AppData/Local/R/win-library/4.3'
## (car 'lib' n'est pas spécifié)

## le package 'tree' a été décompressé et les sommes MD5 ont été vérifiées avec succès
## Warning: impossible de supprimer l'installation précédente du package 'tree'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problème lors de la
## copie de
## C:\Users\Berna\AppData\Local\R\win-library\4.3\00LOCK\tree\libs\x64\tree.dll
## vers C:\Users\Berna\AppData\Local\R\win-library\4.3\tree\libs\x64\tree.dll:
## Permission denied

## Warning: 'tree' restauré

##
## Les packages binaires téléchargés sont dans
## C:\Users\Berna\AppData\Local\Temp\RtmpusIG3Z\downloaded_packages
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.3      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.4      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.0
## v purrr      1.0.2

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
library(regclass)

## Warning: le package 'regclass' a été compilé avec la version R 4.3.2

## Le chargement a nécessité le package : bestglm

## Warning: le package 'bestglm' a été compilé avec la version R 4.3.2

## Le chargement a nécessité le package : leaps

## Warning: le package 'leaps' a été compilé avec la version R 4.3.2

## Le chargement a nécessité le package : VGAM

## Warning: le package 'VGAM' a été compilé avec la version R 4.3.2

```

```
## Le chargement a nécessité le package : stats4
## Le chargement a nécessité le package : splines
## Le chargement a nécessité le package : rpart
## Le chargement a nécessité le package : randomForest

## Warning: le package 'randomForest' a été compilé avec la version R 4.3.2

## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attachement du package : 'randomForest'
##
## L'objet suivant est masqué depuis 'package:dplyr':
##
##      combine
##
## L'objet suivant est masqué depuis 'package:ggplot2':
##
##      margin
##
## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```
library(cluster)
```

```
## Warning: le package 'cluster' a été compilé avec la version R 4.3.2
```

```
library(ggplot2)
```

```
library(fpc)
```

```
## Warning: le package 'fpc' a été compilé avec la version R 4.3.2
```

```
library(dplyr)
```

```
library(rpart.plot)
```

```
## Warning: le package 'rpart.plot' a été compilé avec la version R 4.3.2
```

```
library(C50)
```

```
## Warning: le package 'C50' a été compilé avec la version R 4.3.2
```

```
library(tree)
```

```
## Warning: le package 'tree' a été compilé avec la version R 4.3.2
```

## 1) Analyse exploratoire des données :

L'analyse exploratoire des données vous permettra d'identifier d'éventuels problèmes dans les données (valeurs incohérentes, codage des valeurs manquantes, etc.) et découvrir d'éventuelles propriétés de l'espace des données (valeurs doublons, variables liées, variables d'importance particulière ou bien inutiles, etc.). Appliquez pour cela les différentes méthodes d'analyse exploratoire des données vues en cours (statistiques descriptives, histogrammes, nuages de points, boîtes à moustaches, etc.).

### Chargement des données

```
catalogue <- read.csv("../files/Catalogue.csv", header = TRUE, sep = ",", dec = ".", stringsAsFactors =
head(catalogue)
```

```
##   marque   nom puissance      longueur nbPlaces nbPortes couleur occasion
## 1  Volvo S80 T6      272 tr\xe8s longue      5      5   blanc   false
## 2  Volvo S80 T6      272 tr\xe8s longue      5      5    noir   false
## 3  Volvo S80 T6      272 tr\xe8s longue      5      5   rouge   false
## 4  Volvo S80 T6      272 tr\xe8s longue      5      5    gris    true
## 5  Volvo S80 T6      272 tr\xe8s longue      5      5   bleu    true
## 6  Volvo S80 T6      272 tr\xe8s longue      5      5    gris   false
##   prix
## 1 50500
## 2 50500
## 3 50500
## 4 35350
## 5 35350
## 6 50500
```

```
str(catalogue)
```

### Catalogue

```
## 'data.frame':   270 obs. of  9 variables:
## $ marque   : Factor w/ 21 levels "Hyunda\xef","Audi",...: 21 21 21 21 21 21 21 21 21 21 ...
## $ nom      : Factor w/ 32 levels "1007 1.4","120i",...: 26 26 26 26 26 26 26 26 26 26 ...
## $ puissance: int   272 272 272 272 272 272 272 272 272 ...
## $ longueur : Factor w/ 4 levels "courte","longue",...: 4 4 4 4 4 4 4 4 4 ...
## $ nbPlaces : int    5 5 5 5 5 5 5 5 5 ...
## $ nbPortes : int    5 5 5 5 5 5 5 5 5 ...
## $ couleur  : Factor w/ 5 levels "blanc","bleu",...: 1 4 5 3 2 3 2 5 1 4 ...
## $ occasion : Factor w/ 2 levels "false","true": 1 1 1 2 2 1 1 2 2 ...
## $ prix     : int  50500 50500 50500 35350 35350 50500 50500 35350 35350 ...
```

```
summary(catalogue)
```

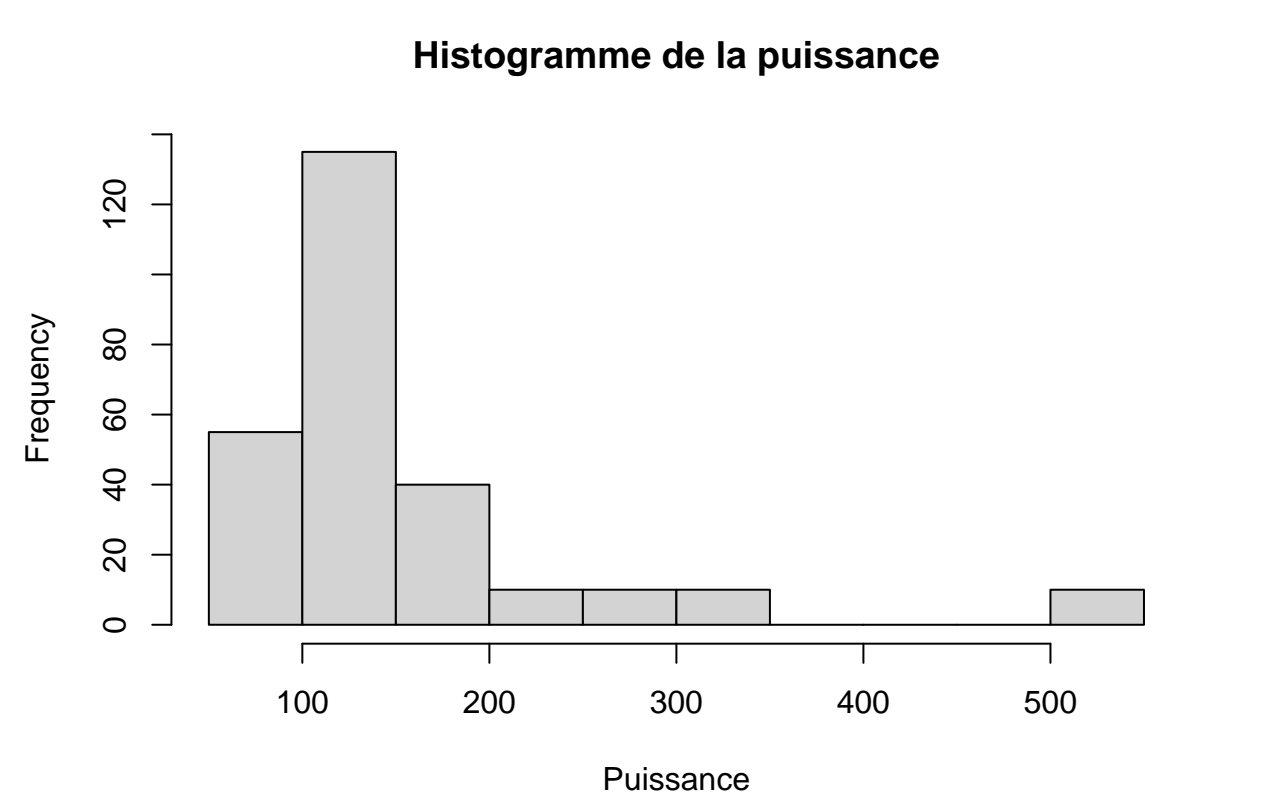
```
##      marque      nom      puissance      longueur
## Renault   : 40   1007 1.4   : 10   Min.    : 55.0   courte      :60
## Volkswagen: 40   120i    : 10   1st Qu.:109.0   longue       :90
## Audi       : 20   9.3 1.8T : 10   Median :147.0   moyenne      :70
## BMW        : 20   A2 1.4   : 10   Mean    :157.6   tr\xe8s longue:50
## Mercedes  : 20   A200     : 10   3rd Qu.:170.0
## Nissan     : 15   A3 2.0 FSI: 10   Max.    :507.0
## (Other)    :115   (Other)  :210
##      nbPlaces      nbPortes      couleur      occasion      prix
## Min.    :5.000   Min.    :3.000   blanc:54   false:160   Min.    : 7500
## 1st Qu.:5.000   1st Qu.:5.000   bleu :54   true :110   1st Qu.: 16029
## Median :5.000   Median :5.000   gris :54                   Median : 20598
## Mean    :5.222   Mean    :4.815   noir :54                   Mean    : 26668
## 3rd Qu.:5.000   3rd Qu.:5.000   rouge:54                   3rd Qu.: 30000
## Max.    :7.000   Max.    :5.000                   Max.    :101300
##
```

Statistiques descriptives pour une colonne spécifique, par exemple, 'puissance'

```
summary(catalogue$puissance)
```

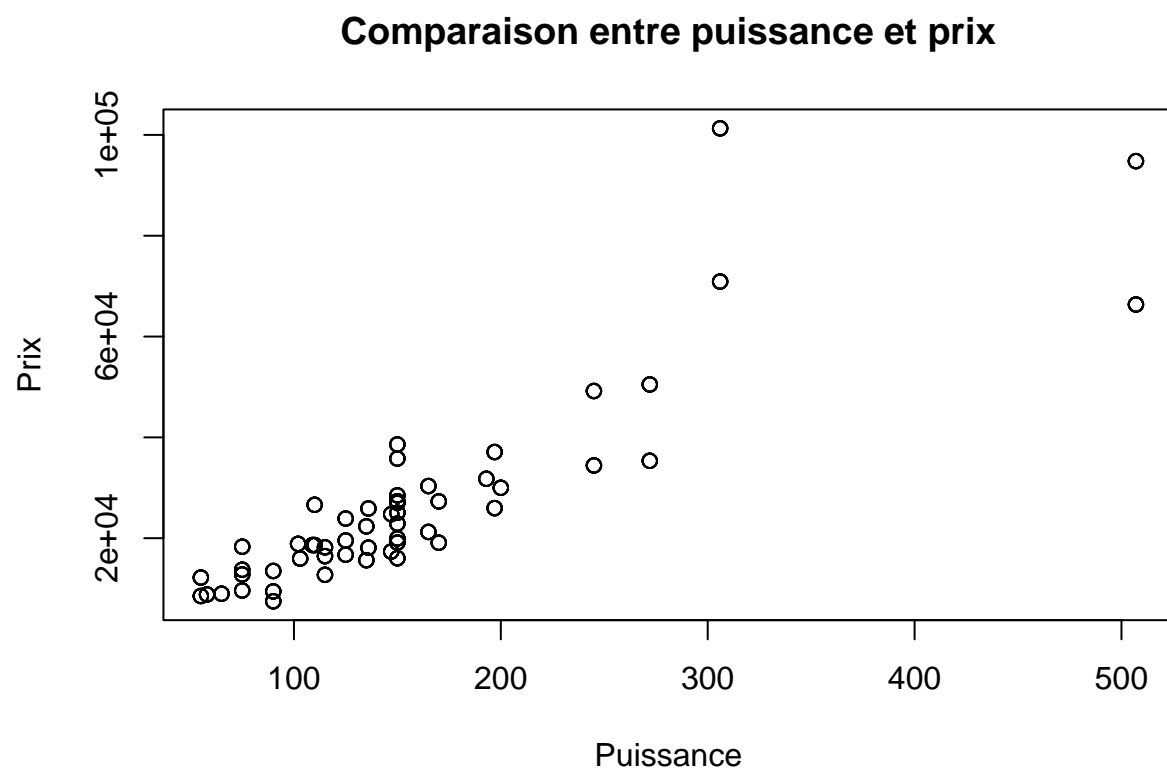
```
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      55.0  109.0   147.0   157.6   170.0   507.0
```

```
hist(catalogue$puissance, main="Histogramme de la puissance", xlab="Puissance")
```



Comparaison entre puissance et prix

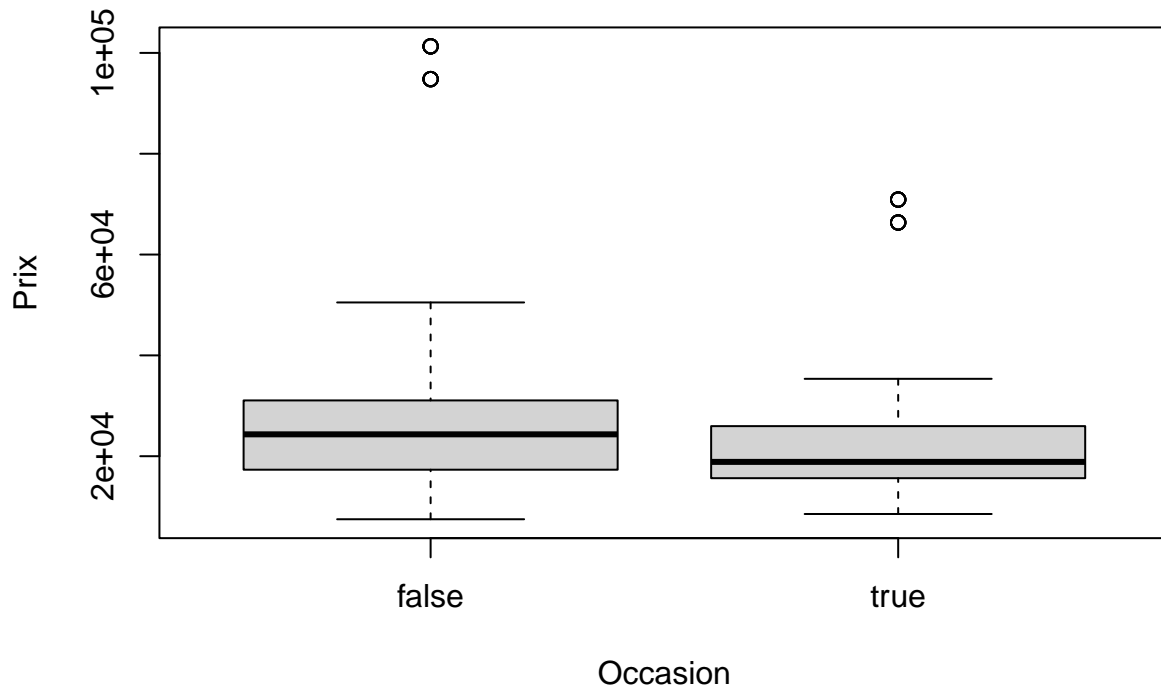
```
plot(catalogue$puissance, catalogue$prix, main="Comparaison entre puissance et prix", xlab="Puissance",
```



Prix en fonction du statut occasion

```
boxplot(catalogue$prix ~ catalogue$occasion, main="Boîte à moustaches pour le prix en fonction de l'occasion")
```

## Boîte à moustaches pour le prix en fonction de l'occasion



On vérifie si il y a des données manquantes

```
sum(is.na(catalogue))
```

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

On vérifie si il y a des lignes en double

```
doublons <- duplicated(catalogue)
```

```
catalogue[doublons, ]
```

```
## [1] marque    nom      puissance longueur nbPlaces nbPortes couleur
```

```
## [8] occasion  prix
```

```
## <0 lignes> (ou 'row.names' de longueur nulle)
```

Recherche de données liées :

```
matrice_cor <- cor(catalogue[, sapply(catalogue, is.numeric)])
```

```
print(matrice_cor)
```

```
##          puissance  nbPlaces  nbPortes      prix
## puissance  1.0000000 -0.05708192 0.3109884  0.87545111
## nbPlaces  -0.05708192  1.00000000 0.1129385 -0.08189026
## nbPortes   0.31098839  0.11293849 1.0000000  0.27147998
## prix       0.87545111 -0.08189026 0.2714800  1.00000000
```

La puissance et le prix ont l'air assez liés !

## 2) Identification des catégories de véhicules :

Vous devez à partir des informations du Catalogue identifier des catégories de véhicules (citadine, routière, sportive, etc.) en fonction de leur taille, puissance, prix, etc. Ces catégories doivent correspondre à divers besoins de la part des clients (une grande voiture pour les familles nombreuses, une petite voiture pour circuler en ville, etc.). Ces catégories de véhicules constitueront les classes à prédire durant les étapes suivantes du processus.

Premièrement on définit des critères d'identification : - Puissance : Utiliser la puissance du véhicule pour distinguer entre des catégories telles que citadine, routière, sportive, etc. - Taille : Utiliser la longueur du véhicule pour également contribuer à la classification. Les citadines ont tendance à être plus courtes que les routières par exemple. - Prix : Utiliser la fourchette de prix pour distinguer entre des catégories de véhicules plus abordables ou haut de gamme.

Pour définir les seuils de classification, on part du principe que l'ensemble proposé est homogène et représentatif. Ainsi on peut se baser sur ces données pour définir différentes classes.

Différentes classes de puissances

```
summary(catalogue$puissance)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      55.0   109.0   147.0   157.6   170.0   507.0
```

Différentes classes de tailles

```
summary(catalogue$longueur)
```

```
##      courte      longue  moyenne tr\xe8s longue
##          60          90          70          50
```

Différentes classes de prix

```
summary(catalogue$prix)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      7500  16029  20598  26668  30000  101300
```

```
# On définit les seuils en fonction des stats affichées juste au dessus
```

```
seuil_puissance <- c(0, 100, 170, Inf)
```

```
seuil_prix <- c(0, 16000, 30000, Inf)
```

```
# Assigner les catégories en fonction des seuils
```

```
catalogue$puissance_class <- cut(catalogue$puissance, breaks = seuil_puissance, labels = c("Puissance faible", "Puissance moyenne", "Puissance élevée"))
```

```
catalogue$prix_class <- cut(catalogue$prix, breaks = seuil_prix, labels = c("Prix faible", "Prix moyen", "Prix élevé"))
```

```
head(catalogue, 20)
```

```
##      marque      nom puissance      longueur nbPlaces nbPortes couleur
## 1      Volvo      S80 T6      272 tr\xe8s longue      5      5      blanc
## 2      Volvo      S80 T6      272 tr\xe8s longue      5      5      noir
## 3      Volvo      S80 T6      272 tr\xe8s longue      5      5      rouge
## 4      Volvo      S80 T6      272 tr\xe8s longue      5      5      gris
## 5      Volvo      S80 T6      272 tr\xe8s longue      5      5      bleu
## 6      Volvo      S80 T6      272 tr\xe8s longue      5      5      gris
## 7      Volvo      S80 T6      272 tr\xe8s longue      5      5      bleu
## 8      Volvo      S80 T6      272 tr\xe8s longue      5      5      rouge
## 9      Volvo      S80 T6      272 tr\xe8s longue      5      5      blanc
## 10     Volvo      S80 T6      272 tr\xe8s longue      5      5      noir
## 11 Volkswagen Touran 2.0 FSI      150      longue      7      5      rouge
```

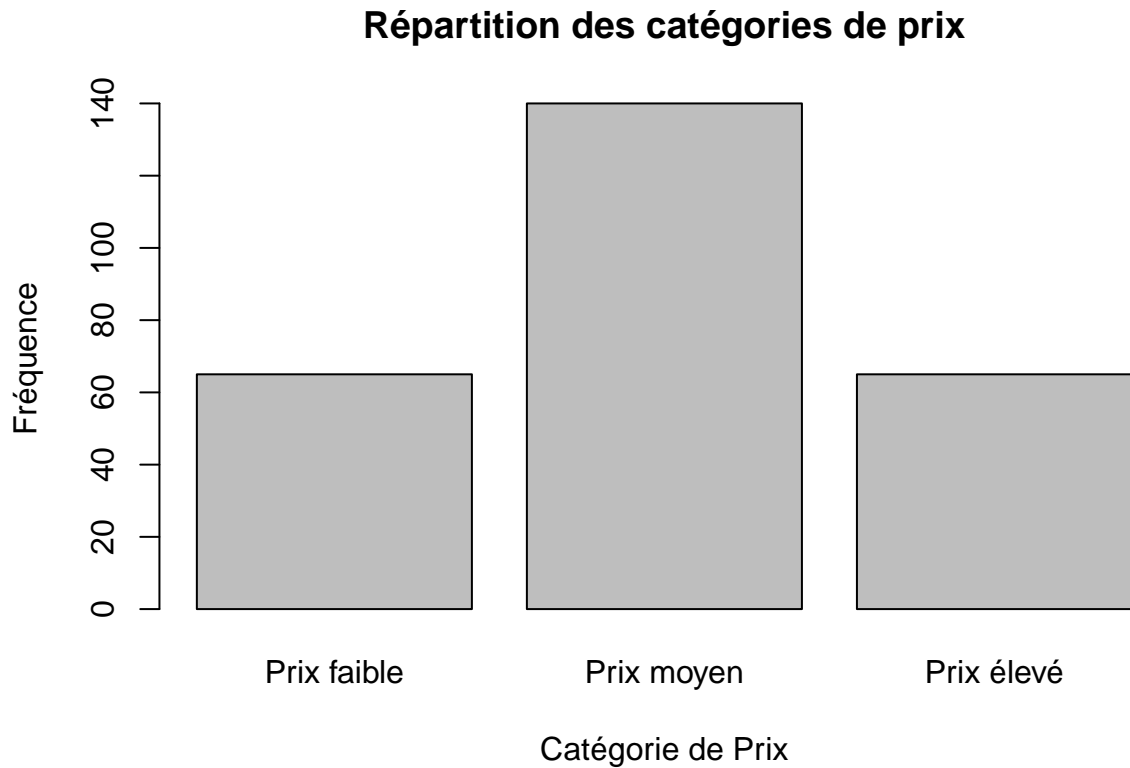


```
## 12 Volkswagen Touran 2.0 FSI      150      longue      7      5      gris
## 13 Volkswagen Touran 2.0 FSI      150      longue      7      5      bleu
## 14 Volkswagen Touran 2.0 FSI      150      longue      7      5      gris
## 15 Volkswagen Touran 2.0 FSI      150      longue      7      5      bleu
## 16 Volkswagen Touran 2.0 FSI      150      longue      7      5      blanc
## 17 Volkswagen Touran 2.0 FSI      150      longue      7      5      noir
## 18 Volkswagen Touran 2.0 FSI      150      longue      7      5      rouge
## 19 Volkswagen Touran 2.0 FSI      150      longue      7      5      blanc
## 20 Volkswagen Touran 2.0 FSI      150      longue      7      5      noir
##      occasion  prix  puissance_class  prix_class
## 1      false 50500  Puissance élevée  Prix élevé
## 2      false 50500  Puissance élevée  Prix élevé
## 3      false 50500  Puissance élevée  Prix élevé
## 4       true 35350  Puissance élevée  Prix élevé
## 5       true 35350  Puissance élevée  Prix élevé
## 6      false 50500  Puissance élevée  Prix élevé
## 7      false 50500  Puissance élevée  Prix élevé
## 8       true 35350  Puissance élevée  Prix élevé
## 9       true 35350  Puissance élevée  Prix élevé
## 10      true 35350  Puissance élevée  Prix élevé
## 11      false 27340  Puissance moyenne  Prix moyen
## 12      true 19138  Puissance moyenne  Prix moyen
## 13      true 19138  Puissance moyenne  Prix moyen
## 14      false 27340  Puissance moyenne  Prix moyen
## 15      false 27340  Puissance moyenne  Prix moyen
## 16      true 19138  Puissance moyenne  Prix moyen
## 17      true 19138  Puissance moyenne  Prix moyen
## 18      true 19138  Puissance moyenne  Prix moyen
## 19      false 27340  Puissance moyenne  Prix moyen
## 20      false 27340  Puissance moyenne  Prix moyen
```

```
# Tableau croisé entre la catégorie de puissance et la catégorie de taille
table_croisee <- table(catalogue$puissance_class, catalogue$longueur)
print(table_croisee)
```

```
##
##      courte longue moyenne tr\xe8s longue
##  Puissance faible      50      0      5      0
##  Puissance moyenne     10     80     65      0
##  Puissance élevée      0     10      0     50
```

```
# Diagramme en barres pour visualiser la répartition des catégories de prix
barplot(table(catalogue$prix_class), main="Répartition des catégories de prix", xlab="Catégorie de Prix")
```



Finalement, grâce a ces classifications, on pourrait affecter une classe à chacun des véhicules :

```
catalogue$classe <- NA
```

```
catalogue <- catalogue %>%
```

```
  mutate(classe = case_when(
```

```
    puissance_class == "Puissance élevée" & longueur != "Courte" ~ "Routière",
```

```
    puissance_class == "Puissance élevée" & longueur == "Courte" & prix_class == "Prix élevé" ~ "Sportive",
```

```
    puissance_class == "Puissance moyenne" & longueur == "Courte" & prix_class == "Prix élevé" ~ "Sportive",
```

```
    puissance_class == "Puissance moyenne" & longueur == "Courte" & prix_class != "Prix élevé" ~ "Citadine",
```

```
    puissance_class == "Puissance moyenne" & longueur != "Courte" ~ "Routière",
```

```
    puissance_class == "Puissance faible" ~ "Citadine",
```

```
    TRUE ~ "?"
```

```
  ))
```

```
head(catalogue, 25)
```

##	marque	nom	puissance	longueur	nbPlaces	nbPortes	couleur
## 1	Volvo	S80	T6	272 tr\	5	5	blanc
## 2	Volvo	S80	T6	272 tr\	5	5	noir
## 3	Volvo	S80	T6	272 tr\	5	5	rouge
## 4	Volvo	S80	T6	272 tr\	5	5	gris
## 5	Volvo	S80	T6	272 tr\	5	5	bleu
## 6	Volvo	S80	T6	272 tr\	5	5	gris
## 7	Volvo	S80	T6	272 tr\	5	5	bleu
## 8	Volvo	S80	T6	272 tr\	5	5	rouge
## 9	Volvo	S80	T6	272 tr\	5	5	blanc

```
## 10      Volvo      S80 T6      272 tr\xe8s longue      5      5      noir
## 11 Volkswagen Touran 2.0 FSI      150      longue      7      5      rouge
## 12 Volkswagen Touran 2.0 FSI      150      longue      7      5      gris
## 13 Volkswagen Touran 2.0 FSI      150      longue      7      5      bleu
## 14 Volkswagen Touran 2.0 FSI      150      longue      7      5      gris
## 15 Volkswagen Touran 2.0 FSI      150      longue      7      5      bleu
## 16 Volkswagen Touran 2.0 FSI      150      longue      7      5      blanc
## 17 Volkswagen Touran 2.0 FSI      150      longue      7      5      noir
## 18 Volkswagen Touran 2.0 FSI      150      longue      7      5      rouge
## 19 Volkswagen Touran 2.0 FSI      150      longue      7      5      blanc
## 20 Volkswagen Touran 2.0 FSI      150      longue      7      5      noir
## 21 Volkswagen      Polo 1.2 6V      55      courte      5      3      blanc
## 22 Volkswagen      Polo 1.2 6V      55      courte      5      3      blanc
## 23 Volkswagen      Polo 1.2 6V      55      courte      5      3      noir
## 24 Volkswagen      Polo 1.2 6V      55      courte      5      3      noir
## 25 Volkswagen      Polo 1.2 6V      55      courte      5      3      bleu
##      occasion prix      puissance_class      prix_class      classe
## 1      false 50500      Puissance élevée      Prix élevé      Routière
## 2      false 50500      Puissance élevée      Prix élevé      Routière
## 3      false 50500      Puissance élevée      Prix élevé      Routière
## 4      true 35350      Puissance élevée      Prix élevé      Routière
## 5      true 35350      Puissance élevée      Prix élevé      Routière
## 6      false 50500      Puissance élevée      Prix élevé      Routière
## 7      false 50500      Puissance élevée      Prix élevé      Routière
## 8      true 35350      Puissance élevée      Prix élevé      Routière
## 9      true 35350      Puissance élevée      Prix élevé      Routière
## 10     true 35350      Puissance élevée      Prix élevé      Routière
## 11     false 27340      Puissance moyenne      Prix moyen      Routière
## 12     true 19138      Puissance moyenne      Prix moyen      Routière
## 13     true 19138      Puissance moyenne      Prix moyen      Routière
## 14     false 27340      Puissance moyenne      Prix moyen      Routière
## 15     false 27340      Puissance moyenne      Prix moyen      Routière
## 16     true 19138      Puissance moyenne      Prix moyen      Routière
## 17     true 19138      Puissance moyenne      Prix moyen      Routière
## 18     true 19138      Puissance moyenne      Prix moyen      Routière
## 19     false 27340      Puissance moyenne      Prix moyen      Routière
## 20     false 27340      Puissance moyenne      Prix moyen      Routière
## 21     true 8540      Puissance faible      Prix faible      Citadine
## 22     false 12200      Puissance faible      Prix faible      Citadine
## 23     false 12200      Puissance faible      Prix faible      Citadine
## 24     true 8540      Puissance faible      Prix faible      Citadine
## 25     true 8540      Puissance faible      Prix faible      Citadine
```

### 3) Application des catégories de véhicules définies aux données des Immatriculations :

Les données d'Immatriculations contiennent les informations sur les véhicules vendus cette année. L'objectif est d'attribuer à chacun de ces véhicules la catégorie qui lui correspond en utilisant le modèle définissant les catégories de véhicules généré précédemment.

```
immatriculations <- read.csv("../files/Immatriculations.csv", header = TRUE, sep = ",", dec = ".", stringsAsFactors = FALSE)
head(immatriculations)
```

```
##      immatriculation      marque      nom puissance      longueur nbPlaces
## 1      3176 TS 67      Renault      Laguna 2.0T      170      longue      5
```

```
## 2      3721 QS 49      Volvo      S80 T6      272 tr\xe8s longue      5
## 3      9099 UV 26 Volkswagen Golf 2.0 FSI      150      moyenne      5
## 4      3563 LA 55      Peugeot      1007 1.4      75      courte      5
## 5      6963 AX 34      Audi      A2 1.4      75      courte      5
## 6      5592 HQ 89      Skoda Superb 2.8 V6      193 tr\xe8s longue      5
##      nbPortes couleur occasion prix
## 1          5      blanc      false 27300
## 2          5      noir      false 50500
## 3          5      gris      true 16029
## 4          5      blanc      true 9625
## 5          5      gris      false 18310
## 6          5      bleu      false 31790
```

```
immatriculations$puissance_class <- cut(immatriculations$puissance, breaks = seuil_puissance, labels = c("Puissance faible", "Puissance moyenne", "Puissance \xe9lev\xe9e"))
immatriculations$prix_class <- cut(immatriculations$prix, breaks = seuil_prix, labels = c("Prix faible", "Prix \xe9lev\xe9", "Prix tr\xe8s \xe9lev\xe9"))
```

```
immatriculations$classe <- NA
```

```
immatriculations <- immatriculations %>%
  mutate(classe = case_when(
    puissance_class == "Puissance \xe9lev\xe9e" & longueur != "Courte" ~ "Routi\xe8re",
    puissance_class == "Puissance \xe9lev\xe9e" & longueur == "Courte" & prix_class == "Prix \xe9lev\xe9" ~ "Sportive",
    puissance_class == "Puissance moyenne" & longueur == "Courte" & prix_class == "Prix \xe9lev\xe9" ~ "Sportive",
    puissance_class == "Puissance moyenne" & longueur == "Courte" & prix_class != "Prix \xe9lev\xe9" ~ "Citadine",
    puissance_class == "Puissance moyenne" & longueur != "Courte" ~ "Routi\xe8re",
    puissance_class == "Puissance faible" ~ "Citadine",
    TRUE ~ "?"
  ))
```

```
head(immatriculations, 25)
```

```
##      immatriculation      marque      nom puissance      longueur nbPlaces
## 1      3176 TS 67      Renault      Laguna 2.0T      170      longue      5
## 2      3721 QS 49      Volvo      S80 T6      272 tr\xe8s longue      5
## 3      9099 UV 26 Volkswagen Golf 2.0 FSI      150      moyenne      5
## 4      3563 LA 55      Peugeot      1007 1.4      75      courte      5
## 5      6963 AX 34      Audi      A2 1.4      75      courte      5
## 6      5592 HQ 89      Skoda Superb 2.8 V6      193 tr\xe8s longue      5
## 7      674 CE 26      Renault Megane 2.0 16V      135      moyenne      5
## 8      1756 PR 31      Mercedes      A200      136      moyenne      5
## 9      6705 GX 50      BMW      120i      150      moyenne      5
## 10     4487 DR 75      Saab      9.3 1.8T      150      longue      5
## 11     7080 NW 34      Jaguar X-Type 2.5 V6      197      longue      5
## 12     9626 HF 36      Audi      A2 1.4      75      courte      5
## 13     2401 PA 98      Volvo      S80 T6      272 tr\xe8s longue      5
## 14     826 YF 89      Renault      Laguna 2.0T      170      longue      5
## 15     8216 GR 23      Skoda Superb 2.8 V6      193 tr\xe8s longue      5
## 16     8076 YM 23      Jaguar X-Type 2.5 V6      197      longue      5
## 17     9277 JN 49      BMW      M5      507 tr\xe8s longue      5
## 18     4231 HC 31      Audi      A2 1.4      75      courte      5
## 19     2319 IQ 28      Ford      Mondeo 1.8      125      longue      5
## 20     148 RS 75      BMW      M5      507 tr\xe8s longue      5
## 21     6786 JV 36      Skoda Superb 2.8 V6      193 tr\xe8s longue      5
## 22     8049 KN 17      Renault Megane 2.0 16V      135      moyenne      5
## 23     9610 BR 52 Volkswagen New Beetle 1.8      110      moyenne      5
```

```
## 24      8745 KJ 12 Volkswagen      Polo 1.2 6V      55      courte      5
## 25      5805 YN 37      BMW      M5      507 tr\xe8s longue      5
##      nbPortes couleur occasion prix puissance_class prix_class classe
## 1      5      blanc      false 27300 Puissance moyenne Prix moyen Routi re
## 2      5      noir      false 50500 Puissance  lev e Prix  lev  Routi re
## 3      5      gris      true 16029 Puissance moyenne Prix moyen Routi re
## 4      5      blanc      true 9625 Puissance faible Prix faible Citadine
## 5      5      gris      false 18310 Puissance faible Prix moyen Citadine
## 6      5      bleu      false 31790 Puissance  lev e Prix  lev  Routi re
## 7      5      gris      false 22350 Puissance moyenne Prix moyen Routi re
## 8      5      noir      true 18130 Puissance moyenne Prix moyen Routi re
## 9      5      noir      true 25060 Puissance moyenne Prix moyen Routi re
## 10     5      gris      true 27020 Puissance moyenne Prix moyen Routi re
## 11     5      blanc      true 25970 Puissance  lev e Prix moyen Routi re
## 12     5      rouge      false 18310 Puissance faible Prix moyen Citadine
## 13     5      bleu      true 35350 Puissance  lev e Prix  lev  Routi re
## 14     5      rouge      false 27300 Puissance moyenne Prix moyen Routi re
## 15     5      bleu      false 31790 Puissance  lev e Prix  lev  Routi re
## 16     5      noir      false 37100 Puissance  lev e Prix  lev  Routi re
## 17     5      rouge      true 66360 Puissance  lev e Prix  lev  Routi re
## 18     5      rouge      false 18310 Puissance faible Prix moyen Citadine
## 19     5      gris      false 23900 Puissance moyenne Prix moyen Routi re
## 20     5      blanc      true 66360 Puissance  lev e Prix  lev  Routi re
## 21     5      gris      false 31790 Puissance  lev e Prix  lev  Routi re
## 22     5      blanc      false 22350 Puissance moyenne Prix moyen Routi re
## 23     5      blanc      true 18641 Puissance moyenne Prix moyen Routi re
## 24     3      gris      false 12200 Puissance faible Prix faible Citadine
## 25     5      gris      true 66360 Puissance  lev e Prix  lev  Routi re
```

#### 4) Fusion des donn es Clients et Immatriculations :

Les donn es Clients contiennent les informations sur les clients ayant les v hicules vendus cette ann e. L'objectif est de faire la fusion entre les donn es des Clients et des Immatriculations afin d'obtenir sur une m me ligne l'ensemble des informations sur le client ( ge, sexe, etc.) et sur le v hicule qu'il a achet  (avec sa cat gorie). Cet ensemble de donn es servira lors des  tapes suivantes pour l'apprentissage de la cat gorie de v hicules (variable cible) la plus adapt e   un client selon ses caract ristiques (variables pr dictives).

```
# On r cupp re les 2 fichiers clients
clients_14 <- read.csv("../files/Clients_14.csv", header = TRUE, sep = ",", dec = ".", stringsAsFactors
clients_19 <- read.csv("../files/Clients_19.csv", header = TRUE, sep = ",", dec = ".", stringsAsFactors

# On combine les 2 pour avoir une variable qui contient l'enti ret  des clients
clients <- bind_rows(clients_14, clients_19)

head(clients)
```

```
##      age      sexe taux situationFamiliale nbEnfantsAcharge X2eme.voiture
## 1  24 Masculin 432      C\ libataire      0      false
## 2  61      M 179      C\ libataire      0      false
## 3  45      M 157      En Couple      4      true
## 4  25      F 1017      En Couple      1      false
## 5  40      M 1149      C\ libataire      0      false
## 6  26      M 161      En Couple      2      false
##      immatriculation
## 1      1482 YR 80
```

```
## 2      1313 NY 98
## 3      6881 00 47
## 4      2600 NK 11
## 5      2834 ZK 77
## 6      5131 NB 60
```

On Vérifie les valeurs dupliquées dans la colonne d'immatriculation

```
clients[duplicated(clients$immatriculation), "immatriculation"]
```

```
## [1] 8382 KY 76 842 XZ 38 2218 EA 59 8678 IV 87 9309 UX 45 9934 HV 57
## [7] 7656 IM 66 756 YT 43 5690 AU 79 8090 ZX 48 3138 NT 61 8088 ZM 38
## [13] 6951 TN 13 875 FJ 81 1347 WR 64 6790 LZ 78 3549 KQ 78 7093 KY 39
## [19] 6460 CS 92 1046 KA 19 1158 UX 15 1430 OU 30 4007 HK 54 420 KK 32
## [25] 832 FX 26 4786 ZB 68 3727 WV 25
## 199973 Levels: 0 BJ 79 0 GV 37 0 IP 35 0 LW 29 0 MY 95 0 OQ 78 0 RO 80 ... 9999 OK 47
```

On voit que le fichier clients contient des erreurs. On a plusieurs enregistrements pour la même immatriculation ! On va donc les supprimer pour n'en garder qu'un seul.

```
clients_unique <- distinct(clients, immatriculation, .keep_all = TRUE)
head(clients_unique, 25)
```

```
##      age      sexe taux situationFamiliale nbEnfantsAcharge X2eme.voiture
## 1     24 Masculin 432      C\xe9libataire              0         false
## 2     61         M 179      C\xe9libataire              0         false
## 3     45         M 157      En Couple                  4          true
## 4     25         F 1017     En Couple                  1         false
## 5     40         M 1149     C\xe9libataire              0         false
## 6     26         M 161      En Couple                  2         false
## 7     49         F 818      Seule                     3         false
## 8     23         M 1382     C\xe9libataire              0         false
## 9     57         M 202      C\xe9libataire              0         false
## 10    58         M 486      C\xe9libataire              0          ?
## 11    83         M 800      C\xe9libataire              0         false
## 12    28         M 504      En Couple                  2          true
## 13    41         M 928      Seule                     2         false
## 14    28         M 547      En Couple                  1         false
## 15    22         F 446      C\xe9libataire              0         false
## 16    51         M 202      C\xe9libataire              0         false
## 17    44         F 438      En Couple                  1         false
## 18    27         M 546      En Couple                  1         false
## 19    21         M 970      C\xe9libataire              0         false
## 20    20         M 491      En Couple                  4         false
## 21    30         M 1229     En Couple                  1         false
## 22    81         F 1025     C\xe9libataire              0         false
## 23    52      Homme 457      En Couple                  1         false
## 24    56         F 494      En Couple                  4         false
## 25    59         M 492      C\xe9libataire              0         false
##      immatriculation
## 1      1482 YR 80
## 2      1313 NY 98
## 3      6881 00 47
## 4      2600 NK 11
## 5      2834 ZK 77
## 6      5131 NB 60
```

```
## 7      5471 BI 54
## 8      913 UF 28
## 9      708 VO 14
## 10     4457 UN 75
## 11     950 LD 79
## 12     8655 FP 18
## 13     4980 QH 29
## 14     1079 PT 86
## 15     3176 HT 73
## 16     2783 UN 68
## 17      437 XU 92
## 18     671 DF 37
## 19     2615 BK 75
## 20     2489 SP 23
## 21     5032 GZ 89
## 22     7066 NS 68
## 23     8535 YJ 56
## 24     8105 HP 71
## 25     2896 EP 88
```

On fait une jointure entre les immatriculations et les clients, en se basant sur le champ immatriculation.

```
donnees_clients <- left_join(immatriculations, clients_unique, by = "immatriculation")

tail(donnees_clients, 25)
```

##	immatriculation	marque	nom	puissance	longueur		
## 1999976	4030 YB 47	Volvo	S80 T6	272	tr\xe8s longue		
## 1999977	8227 CV 11	Peugeot	1007 1.4	75	courte		
## 1999978	2879 MN 71	BMW	M5	507	tr\xe8s longue		
## 1999979	7654 MW 51	Peugeot	1007 1.4	75	courte		
## 1999980	1090 VC 58	Mercedes	A200	136	moyenne		
## 1999981	3172 BT 98	Mini	Copper 1.6 16V	115	courte		
## 1999982	6076 TR 42	Jaguar	X-Type 2.5 V6	197	longue		
## 1999983	2100 CA 13	Mercedes	S500	306	tr\xe8s longue		
## 1999984	8598 KV 69	Audi	A2 1.4	75	courte		
## 1999985	5294 AW 72	Jaguar	X-Type 2.5 V6	197	longue		
## 1999986	8635 BV 94	Audi	A2 1.4	75	courte		
## 1999987	8206 AO 91	Mercedes	S500	306	tr\xe8s longue		
## 1999988	8117 MO 41	Jaguar	X-Type 2.5 V6	197	longue		
## 1999989	3282 YF 65	Volkswagen	Polo 1.2 6V	55	courte		
## 1999990	3859 VN 85	BMW	M5	507	tr\xe8s longue		
## 1999991	4327 NK 30	Jaguar	X-Type 2.5 V6	197	longue		
## 1999992	8790 NR 54	Volkswagen	Polo 1.2 6V	55	courte		
## 1999993	6623 GH 58	Ford	Mondeo 1.8	125	longue		
## 1999994	1679 JI 70	Volkswagen	Polo 1.2 6V	55	courte		
## 1999995	8221 KM 45	Audi	A2 1.4	75	courte		
## 1999996	771 CQ 78	Mercedes	S500	306	tr\xe8s longue		
## 1999997	8182 PL 97	Lancia	Ypsilon 1.4 16V	90	courte		
## 1999998	8550 AP 53	Ford	Mondeo 1.8	125	longue		
## 1999999	737 MK 20	Audi	A2 1.4	75	courte		
## 2000000	403 PT 42	Volkswagen	Golf 2.0 FSI	150	moyenne		
##	nbPlaces	nbPortes	couleur	occasion	prix	puissance_class	prix_class
## 1999976	5	5	bleu	false	50500	Puissance élevée	Prix élevé
## 1999977	5	5	noir	false	13750	Puissance faible	Prix faible

##	1999978	5	5	blanc	false	94800	Puissance élevée	Prix élevé
##	1999979	5	5	rouge	false	13750	Puissance faible	Prix faible
##	1999980	5	5	rouge	false	25900	Puissance moyenne	Prix moyen
##	1999981	5	5	noir	false	18200	Puissance moyenne	Prix moyen
##	1999982	5	5	noir	false	37100	Puissance élevée	Prix élevé
##	1999983	5	5	gris	false	101300	Puissance élevée	Prix élevé
##	1999984	5	5	gris	false	18310	Puissance faible	Prix moyen
##	1999985	5	5	gris	false	37100	Puissance élevée	Prix élevé
##	1999986	5	5	gris	false	18310	Puissance faible	Prix moyen
##	1999987	5	5	bleu	false	101300	Puissance élevée	Prix élevé
##	1999988	5	5	gris	false	37100	Puissance élevée	Prix élevé
##	1999989	5	3	gris	false	12200	Puissance faible	Prix faible
##	1999990	5	5	gris	true	66360	Puissance élevée	Prix élevé
##	1999991	5	5	bleu	true	25970	Puissance élevée	Prix moyen
##	1999992	5	3	blanc	false	12200	Puissance faible	Prix faible
##	1999993	5	5	noir	false	23900	Puissance moyenne	Prix moyen
##	1999994	5	3	noir	true	8540	Puissance faible	Prix faible
##	1999995	5	5	blanc	true	12817	Puissance faible	Prix faible
##	1999996	5	5	gris	true	70910	Puissance élevée	Prix élevé
##	1999997	5	3	blanc	true	9450	Puissance faible	Prix faible
##	1999998	5	5	rouge	false	23900	Puissance moyenne	Prix moyen
##	1999999	5	5	bleu	true	12817	Puissance faible	Prix faible
##	2000000	5	5	blanc	false	22900	Puissance moyenne	Prix moyen
##		classe	age	sexe	taux	situationFamiliale	nbEnfantsAcharge	
##	1999976	Routière	24	F	497	En Couple	3	
##	1999977	Citadine	77	M	520	C\xe9libataire	0	
##	1999978	Routière	59	M	1114	En Couple	3	
##	1999979	Citadine	58	F	547	C\xe9libataire	0	
##	1999980	Routière	71	F	1320	C\xe9libataire	0	
##	1999981	Routière	24	F	1381	C\xe9libataire	0	
##	1999982	Routière	37	M	1252	En Couple	2	
##	1999983	Routière	62	F	788	Seule	3	
##	1999984	Citadine	34	M	1083	C\xe9libataire	0	
##	1999985	Routière	29	M	992	En Couple	2	
##	1999986	Citadine	32	M	704	C\xe9libataire	0	
##	1999987	Routière	68	M	785	En Couple	1	
##	1999988	Routière	48	M	1181	En Couple	0	
##	1999989	Citadine	53	M	578	C\xe9libataire	0	
##	1999990	Routière	29	M	520	En Couple	4	
##	1999991	Routière	19	F	519	En Couple	2	
##	1999992	Citadine	46	M	463	En Couple	0	
##	1999993	Routière	26	M	458	En Couple	1	
##	1999994	Citadine	69	F	246	C\xe9libataire	0	
##	1999995	Citadine	22	F	555	C\xe9libataire	0	
##	1999996	Routière	62	F	433	En Couple	2	
##	1999997	Citadine	28	F	165	C\xe9libataire	0	
##	1999998	Routière	31	M	497	En Couple	1	
##	1999999	Citadine	48	M	479	En Couple	1	
##	2000000	Routière	25	M	527	C\xe9libataire	0	
##		X2eme.voiture						
##	1999976	false						
##	1999977	false						
##	1999978	false						
##	1999979	false						



```
## 1999980      false
## 1999981      false
## 1999982      false
## 1999983      false
## 1999984      false
## 1999985      false
## 1999986      false
## 1999987      false
## 1999988      false
## 1999989      false
## 1999990      false
## 1999991      false
## 1999992       true
## 1999993      false
## 1999994      false
## 1999995      false
## 1999996      false
## 1999997      false
## 1999998      false
## 1999999       true
## 2000000      false
```

## 5) Création d'un modèle de classification supervisée pour la prédiction de la catégorie de véhicules :

L'objectif de cette étape est de créer à partir du résultat de la fusion précédente un classifieur (modèle de classification supervisée) permettant d'associer aux caractéristiques des clients (âge, sexe, etc.) une catégorie de véhicules. Testez les différentes approches et algorithmes (arbres de décision, random forests, support vector machines, réseaux de neurones, deep learning, etc.), avec pour chaque algorithme plusieurs paramétrages testés, afin d'obtenir un classifieur aussi performant que possible. L'évaluation et la comparaison des performances de chaque configuration algorithmique (un algorithme et un paramétrage spécifiques) testée sera réalisée grâce aux matrices de confusion et mesures d'évaluation calculées à partir des résultats des tests des classifieurs.

```
data <- na.omit(donnees_clients)
sum(is.na(data))
```

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

```
data$situationFamiliale <- iconv(data$situationFamiliale, to = "UTF-8", sub = "byte")
head(data)
```

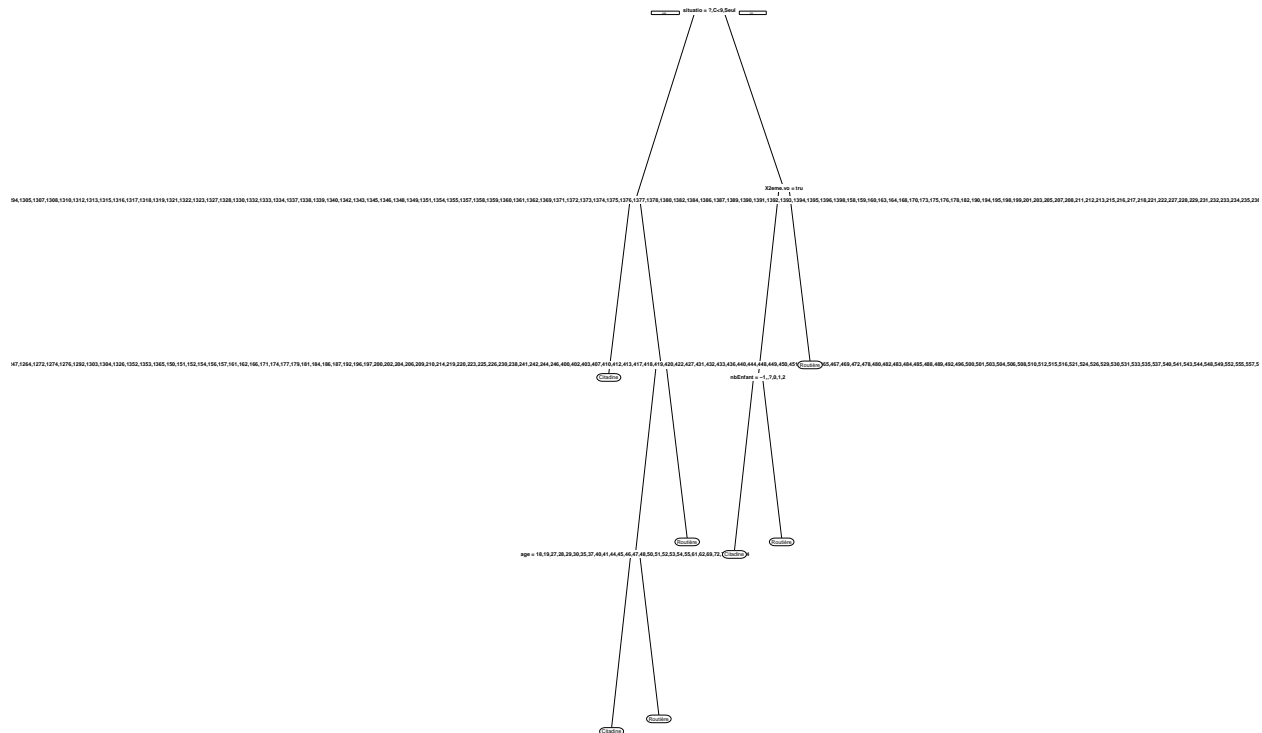
```
##      immatriculation  marque      nom puissance      longueur nbPlaces
## 503      3541 BP 36 Volkswagen Polo 1.2 6V      55      courte      5
## 5501     2210 HK 34      Audi      A2 1.4      75      courte      5
## 11115     4313 XE 20      Audi      A2 1.4      75      courte      5
## 14897     7483 VF 89    Peugeot    1007 1.4      75      courte      5
## 18177     2088 GE 10 Volkswagen Golf 2.0 FSI    150     moyenne      5
## 19629     2184 SJ 32      Volvo      S80 T6     272 tr\xe8s longue      5
##      nbPortes couleur occasion  prix  puissance_class  prix_class  classe
## 503      3      noir      true  8540  Puissance faible  Prix faible  Citadine
## 5501      5      blanc     false 18310  Puissance faible  Prix moyen  Citadine
## 11115      5      blanc     false 18310  Puissance faible  Prix moyen  Citadine
## 14897      5      blanc     false 13750  Puissance faible  Prix faible  Citadine
## 18177      5      bleu     false 22900  Puissance moyenne  Prix moyen  Routi\u00e8re
## 19629      5      blanc     true 35350  Puissance \u00e9lev\u00e9e  Prix \u00e9lev\u00e9  Routi\u00e8re
```

```
##      age sexe taux situationFamiliale nbEnfantsAcharge X2eme.voiture
## 503    18  F 1220      C<e9>libataire              0      false
## 5501   57  F 1284      C<e9>libataire              0      false
## 11115  31  M  209      En Couple                  3      false
## 14897  58  M  415      En Couple                  1       true
## 18177  60  M  544      En Couple                  0      false
## 19629  27  M 1249      En Couple                  3       true
```

```
data_train <- data[1:10000,]
data_test  <- data[10001:200000,]
```

```
tree1 <- rpart(classe ~ age + sexe + taux + situationFamiliale + nbEnfantsAcharge + X2eme.voiture, data,
prp(tree1))
```

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



```
test_tree1 <- predict(tree1, data_test, type="class")
table(test_tree1)
```

```
## test_tree1
## Citadine Routière
##      48525      141475
```

```
data_test$Tree1 <- test_tree1
```

```
taux_succes1 <- nrow(data_test[data_test$classe==data_test$Tree1,]) / nrow(data_test)
taux_succes1
```

```
## [1] 0.8658421
```

```
head(data_test, 30)
```

##	immatriculation	marque	nom	puissance	longueur		
## 1409541	3148 GC 29	Audi	A2 1.4	75	courte		
## 1409542	7479 CT 58	Renault	Megane 2.0 16V	135	moyenne		
## 1409543	9769 UH 98	Jaguar	X-Type 2.5 V6	197	longue		
## 1409544	3837 AG 79	Audi	A2 1.4	75	courte		
## 1409545	2239 JA 73	BMW	M5	507	tr\xe8s longue		
## 1409546	7240 FR 96	Audi	A2 1.4	75	courte		
## 1409547	756 DM 43	Skoda	Superb 2.8 V6	193	tr\xe8s longue		
## 1409548	5730 NJ 38	Volkswagen	Polo 1.2 6V	55	courte		
## 1409549	4407 CH 38	Jaguar	X-Type 2.5 V6	197	longue		
## 1409550	8207 BR 34	Mercedes	S500	306	tr\xe8s longue		
## 1409551	98 BB 10	Mercedes	A200	136	moyenne		
## 1409552	719 LI 72	Mercedes	A200	136	moyenne		
## 1409553	3848 UW 90	Peugeot	1007 1.4	75	courte		
## 1409554	1833 CF 18	Audi	A2 1.4	75	courte		
## 1409555	4902 LQ 45	BMW	M5	507	tr\xe8s longue		
## 1409556	624 MG 96	Audi	A2 1.4	75	courte		
## 1409557	9843 CV 41	Skoda	Superb 2.8 V6	193	tr\xe8s longue		
## 1409558	5167 CX 73	Nissan	Maxima 3.0 V6	200	tr\xe8s longue		
## 1409559	2929 LM 63	Jaguar	X-Type 2.5 V6	197	longue		
## 1409560	2046 MO 68	Nissan	Primera 1.6	109	longue		
## 1409561	9531 NE 10	Mercedes	S500	306	tr\xe8s longue		
## 1409562	4540 LC 67	BMW	M5	507	tr\xe8s longue		
## 1409563	1959 JX 78	Audi	A2 1.4	75	courte		
## 1409564	6404 GB 87	BMW	M5	507	tr\xe8s longue		
## 1409565	836 MQ 10	Peugeot	1007 1.4	75	courte		
## 1409566	5706 ER 69	Audi	A2 1.4	75	courte		
## 1409567	5034 GA 51	Volkswagen	Polo 1.2 6V	55	courte		
## 1409568	7549 EF 94	Volkswagen	Polo 1.2 6V	55	courte		
## 1409569	5022 SG 72	BMW	M5	507	tr\xe8s longue		
## 1409570	9636 QX 13	Nissan	Maxima 3.0 V6	200	tr\xe8s longue		
##	nbPlaces	nbPortes	couleur	occasion	prix	puissance_class	prix_class
## 1409541	5	5	blanc	false	18310	Puissance faible	Prix moyen
## 1409542	5	5	bleu	false	22350	Puissance moyenne	Prix moyen
## 1409543	5	5	rouge	true	25970	Puissance \xe9lev\xe9e	Prix moyen
## 1409544	5	5	noir	false	18310	Puissance faible	Prix moyen
## 1409545	5	5	rouge	false	94800	Puissance \xe9lev\xe9e	Prix \xe9lev\xe9
## 1409546	5	5	bleu	false	18310	Puissance faible	Prix moyen
## 1409547	5	5	noir	false	31790	Puissance \xe9lev\xe9e	Prix \xe9lev\xe9
## 1409548	5	3	bleu	false	12200	Puissance faible	Prix faible
## 1409549	5	5	bleu	false	37100	Puissance \xe9lev\xe9e	Prix \xe9lev\xe9
## 1409550	5	5	noir	false	101300	Puissance \xe9lev\xe9e	Prix \xe9lev\xe9
## 1409551	5	5	bleu	false	25900	Puissance moyenne	Prix moyen
## 1409552	5	5	blanc	true	18130	Puissance moyenne	Prix moyen
## 1409553	5	5	noir	false	13750	Puissance faible	Prix faible
## 1409554	5	5	noir	false	18310	Puissance faible	Prix moyen
## 1409555	5	5	blanc	false	94800	Puissance \xe9lev\xe9e	Prix \xe9lev\xe9
## 1409556	5	5	blanc	true	12817	Puissance faible	Prix faible
## 1409557	5	5	blanc	false	31790	Puissance \xe9lev\xe9e	Prix \xe9lev\xe9
## 1409558	5	5	bleu	false	30000	Puissance \xe9lev\xe9e	Prix moyen
## 1409559	5	5	gris	false	37100	Puissance \xe9lev\xe9e	Prix \xe9lev\xe9
## 1409560	5	5	rouge	false	18650	Puissance moyenne	Prix moyen

##	1409561	5	5	noir	true	70910	Puissance élevée	Prix élevé
##	1409562	5	5	blanc	true	66360	Puissance élevée	Prix élevé
##	1409563	5	5	noir	true	12817	Puissance faible	Prix faible
##	1409564	5	5	blanc	false	94800	Puissance élevée	Prix élevé
##	1409565	5	5	noir	false	13750	Puissance faible	Prix faible
##	1409566	5	5	noir	false	18310	Puissance faible	Prix moyen
##	1409567	5	3	noir	true	8540	Puissance faible	Prix faible
##	1409568	5	3	rouge	false	12200	Puissance faible	Prix faible
##	1409569	5	5	gris	true	66360	Puissance élevée	Prix élevé
##	1409570	5	5	gris	false	30000	Puissance élevée	Prix moyen
##		classe	age	sexe	taux	situation	Familiale	nbEnfantsAcharge
##	1409541	Citadine	81	M	1153		En Couple	0
##	1409542	Routière	47	F	432	C<e9>	libataire	0
##	1409543	Routière	31	F	471		En Couple	1
##	1409544	Citadine	48	F	701	C<e9>	libataire	0
##	1409545	Routière	19	F	949		En Couple	3
##	1409546	Citadine	67	M	1129		En Couple	2
##	1409547	Routière	32	F	151		En Couple	3
##	1409548	Citadine	36	M	592	C<e9>	libataire	0
##	1409549	Routière	52	F	755		En Couple	2
##	1409550	Routière	78	F	950		En Couple	0
##	1409551	Routière	74	M	814	C<e9>	libataire	0
##	1409552	Routière	83	M	535	C<e9>	libataire	
##	1409553	Citadine	19	F	594	C<e9>	libataire	0
##	1409554	Citadine	45	Homme	1003			0
##	1409555	Routière	19	M	1116		En Couple	2
##	1409556	Citadine	57	F	505		En Couple	0
##	1409557	Routière	29	M	249		En Couple	
##	1409558	Routière	38	M	248		En Couple	0
##	1409559	Routière	28	M	1369		En Couple	1
##	1409560	Routière	59	M	196		En Couple	1
##	1409561	Routière	63	M	468		En Couple	4
##	1409562	Routière	55	M	546		En Couple	3
##	1409563	Citadine	40	M	581	C<e9>	libataire	0
##	1409564	Routière	31	F	1023		En Couple	4
##	1409565	Citadine	51	M	444	C<e9>	libataire	0
##	1409566	Citadine	41	M	1204	C<e9>	libataire	0
##	1409567	Citadine	69	M	166	C<e9>	libataire	0
##	1409568	Citadine	18	M	463		En Couple	2
##	1409569	Routière	50	M	543		En Couple	4
##	1409570	Routière	72	F	165		En Couple	4
##		X2eme.voiture		Tree1				
##	1409541		true	Citadine				
##	1409542		false	Citadine				
##	1409543		false	Routière				
##	1409544		false	Citadine				
##	1409545		true	Routière				
##	1409546		true	Citadine				
##	1409547		false	Routière				
##	1409548		false	Routière				
##	1409549		false	Routière				
##	1409550		false	Routière				
##	1409551		false	Citadine				
##	1409552		false	Routière				

```
## 1409553      false Citadine
## 1409554      false Routière
## 1409555      false Routière
## 1409556       true Citadine
## 1409557      false Routière
## 1409558      false Routière
## 1409559      false Routière
## 1409560      false Routière
## 1409561      false Routière
## 1409562      false Routière
## 1409563      false Citadine
## 1409564      false Routière
## 1409565      false Citadine
## 1409566      false Routière
## 1409567      false Citadine
## 1409568       true Citadine
## 1409569      false Routière
## 1409570      false Routière
```

```
#tree2 <- C5.0(classe ~ ., data, control = C5.0Control(minCases = 5, winnow=F, CF=F))
#plot(tree2, type="simple")
```

```
#tree3 <- tree(classe ~ ., data)
```

```
#plot(tree3)
#text(tree3, pretty=0)
```

## 6) Application du modèle de prédiction aux données Marketing :

Les données Marketing contiennent les informations sur les clients pour lesquels on souhaite prédire une catégorie de véhicules. L'objectif est de prédire pour chacun de ces clients la catégorie de véhicules qui lui correspond le mieux en utilisant le classifieur généré durant l'étape précédente.

```
marketing <- read.csv("../files/Marketing.csv", header = TRUE, sep = ",", dec = ".", stringsAsFactors =
marketing$situationFamiliale <- iconv(marketing$situationFamiliale, to = "UTF-8", sub = "byte")
```

```
head(marketing)
```

```
##   age sexe taux situationFamiliale nbEnfantsAcharge X2eme.voiture
## 1  21   F 1396      C<e9>libataire             0         false
## 2  35   M  223      C<e9>libataire             0         false
## 3  48   M  401      C<e9>libataire             0         false
## 4  26   F  420      En Couple                 3          true
## 5  80   M  530      En Couple                 3         false
## 6  27   F  153      En Couple                 2         false
```

```
marketing$age <- as.factor(marketing$age)
marketing$taux <- as.factor(marketing$taux)
marketing$nbEnfantsAcharge <- as.factor(marketing$nbEnfantsAcharge)
```

```
class_treeN <- predict(tree1, marketing, type="class")
table(class_treeN)
```

```
## class_treeN
## Citadine Routière
##           8          12
```

```
marketing$Prediction <- class_treeN
marketing
```

##	age	sexe	taux	situationFamiliale	nbEnfantsAcharge	X2eme.voiture	Prediction
## 1	21	F	1396	C<e9>libataire	0	false	Citadine
## 2	35	M	223	C<e9>libataire	0	false	Citadine
## 3	48	M	401	C<e9>libataire	0	false	Routière
## 4	26	F	420	En Couple	3	true	Routière
## 5	80	M	530	En Couple	3	false	Routière
## 6	27	F	153	En Couple	2	false	Routière
## 7	59	F	572	En Couple	2	false	Routière
## 8	43	F	431	C<e9>libataire	0	false	Routière
## 9	64	M	559	C<e9>libataire	0	false	Citadine
## 10	22	M	154	En Couple	1	false	Routière
## 11	79	F	981	En Couple	2	false	Routière
## 12	55	M	588	C<e9>libataire	0	false	Citadine
## 13	19	F	212	C<e9>libataire	0	false	Citadine
## 14	34	F	1112	En Couple	0	false	Routière
## 15	60	M	524	En Couple	0	true	Citadine
## 16	22	M	411	En Couple	3	true	Routière
## 17	58	M	1192	En Couple	0	false	Routière
## 18	54	F	452	En Couple	3	true	Routière
## 19	35	M	589	C<e9>libataire	0	false	Citadine
## 20	59	M	748	En Couple	0	true	Citadine