Projet Analyse de données

Auto-mpg

Idir SADAOUI, Hamady CISSÉ

Contents

1 Introduction

Ce projet est effectué dans le cadre d'améliorer notre compréhension et notre manipulation des outils de la librarie **tidyverse** étudiés en cours. Nous avons choisi d'étudier un dataset sur des véhicules, en voici une brève présentation :

Nous disposons d'un document comportant les données de 398 véhicules.
Ce document provient de la bibliothèque StatLib qui est affilié à l'Université Carnegie Mellon située à Pittsburgh en Pennsylvanie ; il a déjà été utilisé entre autre dans l'exposition de l'American Statistical Assosiation de 1983.
On sait de plus que les données du document ont été légèrement modifié pour corriger l'absence de données à certains endroits.

Ce sujet nous a plu tout simplement parce que l'on est fan de voiture, ainsi les variables de ce jeu de données sont plus facilement compréhensibles pour nous.

Par ailleurs, voici une brève explication de chaque variable :

- *Mpg* est l'abréviation de "Miles Per Gallon", c'est la distance parcourue pour un gallon d'essence et elle est exprimée en Miles. (variable continue)
- Les *Cylindres* sont des éléments du moteur où se déplace le piston. (variable discrète)
- La *Cylindrée* est le volume balayé par le piston, elle est exprimée ici en Cubic Inch (Cu In). (variable continue)
- La *Puissance* du véhicule qui est exprimée en Cheveaux (Cv). (variable continue)
- Le *Poids* qui est exprimé en Livre (Lbs). (variable continue)
- L' *Accélération* qui est le rapport entre une variation de vitesse et l'unité de temps et qui est exprimée en Yard par seconde carré. (variable continue)
- L' *Année* du véhicule (entre 1970 et 1982).
- L' *Origine* du véhicule avec 1 pour l'Amérique, 2 pour l'Europe et 3 pour l'Asie.

Quelques questions auxquelles nous tenterons de répondre dans notre analyse :

- Comment évolue la consommation en fonction des variables ?
- L'origine du véhicule est elle importante pour expliquer la consommation ?
- Et enfin, Quelles sont les marques de véhicules qui consomment le plus ?

Voici le lien hypertexte de notre jeu de données : lien

1.1 Importation des Packages

Voici les packages dont on aura besoin tout au long de notre analyse :

```
library(tidyverse)
library(readxl)
library(dplyr)
library(corrplot)
library(cowplot)
library(ggridges)
library(ggforce)
library(treemap)
library(wordcloud)
library(packcircles)
```

1.2 Importation des données

Nous allons importer nos données grâce aux fonctions d'importation de tidyverse qui transforme directement en tibble, le séparateur est initialisé automatiquement et reconnait les formats data, double et chaine de caractères.

```
setwd("~/Desktop")
auto <- read_excel('auto-mpg.xlsx',col_names = F,na = '?',skip = 1)
dim(auto)
```

```
## [1] 398   9
```

On est en présence d'un jeu de données de 9 variables et 398 observations.
On pense que toutes les variables sont interéssantes donc on décide de les garder pour notre analyse.

1.3 Corrections préliminaires des données

1.3.1 Optimisation des données

On va maintenant optimiser notre jeu de données pour une analyse plus agréable.
Tout d'abord, on va nommer les 9 colonnes qui correspondent à nos variables avec la fonction `rename` :

```
auto %>% rename(Mpg = 1, Cylindres = 2, Cylindree = 3, Puissance = 4, Poids = 5,
                Acceleration = 6, Annee = 7, Origine = 8, Nom = 9) -> auto
```

Ensuite grâce à la commande `str` on va inspecter rapidement le jeu de données :

```
auto %>% str
```

```
## tibble [398 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Mpg         : chr [1:398] "18.0" "15.0" "18.0" "16.0" ...
##  $ Cylindres   : num [1:398] 8 8 8 8 8 8 8 8 8 8 ...
##  $ Cylindree   : chr [1:398] "307.0" "350.0" "318.0" "304.0" ...
##  $ Puissance   : chr [1:398] "130.0" "165.0" "150.0" "150.0" ...
##  $ Poids       : chr [1:398] "3504." "3693." "3436." "3433." ...
##  $ Acceleration: chr [1:398] "12.0" "11.5" "11.0" "12.0" ...
##  $ Annee       : num [1:398] 70 70 70 70 70 70 70 70 70 70 ...
##  $ Origine     : num [1:398] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Nom         : chr [1:398] "Chevrolet-chevelle malibu" "Buick-skylark 320" "Plymouth-satellite" "Al
```

On remarque que les variables *Mpg*, *Cylindree*, *Puissance*, *Poids* et *Acceleration* sont comptées comme des chaînes de caractères alors qu'elles sont censées être numérique.
On va donc les modifier grâce aux fonctions `mutate` et `parse_number` qui vont permettre de transformer les données de ces variables en données numériques.

```
auto %>% mutate(Mpg = parse_number(Mpg), Cylindree = parse_number(Cylindree),
                Puissance = parse_number(Puissance),Poids = parse_number(Poids),
                Acceleration = parse_number(Acceleration)) -> auto
```

On revérifie rapidement si le changement à bien été effectué :

```
auto %>% str
```

```
## tibble [398 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Mpg         : num [1:398] 18 15 18 16 17 15 14 14 14 15 ...
##  $ Cylindres   : num [1:398] 8 8 8 8 8 8 8 8 8 8 ...
##  $ Cylindree   : num [1:398] 307 350 318 304 302 429 454 440 455 390 ...
##  $ Puissance   : num [1:398] 130 165 150 150 140 198 220 215 225 190 ...
##  $ Poids       : num [1:398] 3504 3693 3436 3433 3449 ...
##  $ Acceleration: num [1:398] 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ Annee       : num [1:398] 70 70 70 70 70 70 70 70 70 70 ...
##  $ Origine     : num [1:398] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Nom         : chr [1:398] "Chevrolet-chevelle malibu" "Buick-skylark 320" "Plymouth-satellite" "Al
```

C'est bien le cas.

Ensuite on décide d'ajouter 1900 à chaque valeurs de la colonne *Annee* pour avoir les années au bon format :

```
auto %>% mutate(Annee = Annee+1900) -> auto
auto %>% select(Annee) %>% head()
```

```
## # A tibble: 6 x 1
##   Annee
##   <dbl>
## 1  1970
## 2  1970
## 3  1970
## 4  1970
## 5  1970
## 6  1970
```

La variable *Origine* est définie de tel sorte qu'elle soit égale à 1 pour les véhicules américains, 2 pour les européens et 3 pour les asiatiques, on décide de modifier les valeurs de la variable en remplaçant les chiffres par leur continent (en chaîne de caractère) grâce aux fonctions `mutate` et `case_when` :

```
mutate(auto,Origine = case_when(Origine == 1  ~ "Amérique",
                                Origine == 2 ~ "Europe",
                                Origine == 3 ~ "Asie")) -> auto
auto %>% select(Origine) %>% head()
```

```
## # A tibble: 6 x 1
##   Origine
##   <chr>
## 1 Amérique
## 2 Amérique
## 3 Amérique
## 4 Amérique
## 5 Amérique
## 6 Amérique
```

On va ensuite séparer la variable *Nom* en deux pour isoler la marque du véhicule ainsi que son modèle, ceci grâce à la fonction `separate` :

```
auto %>% separate(Nom, into = c("Marque", "Modele"), sep = "-" ) -> auto
auto %>% select(Marque,Modele) %>% head()
```

```
## # A tibble: 6 x 2
##   Marque    Modele
##   <chr>     <chr>
## 1 Chevrolet chevelle malibu
## 2 Buick     skylark 320
## 3 Plymouth  satellite
## 4 AMC       rebel sst
## 5 Ford      torino
## 6 Ford      galaxie 500
```

Les deux nouvelles colonnes s'appellent *Marque* et *Modele*.

1.3.2 Gestion des valeurs manquantes

On va vérifier si le jeu de données contient des valeurs manquantes notamment avec la fonction `is.na`:

```
auto %>% is.na() %>% any()
```

```
## [1] TRUE
```

```
auto %>% is.na() %>% colSums()
```

```
##          Mpg    Cylindres    Cylindree    Puissance        Poids Acceleration
##            0            0            0            6            0            0
##       Annee      Origine       Marque       Modele
##            0            0            0            0
```

On voit qu'il y a 6 valeurs manquantes au niveau de la variable *Puissance* et étant donnée ce faible nombre, on décide de supprimer les observations contenant une valeur manquante à *Puissance*.

```
auto %>% na.omit() -> auto
```

Enfin, pour finir ces corrections, on décide de créer une variable que l'on va nommer *Numero* qui sera un numéro unique pour référencer toutes les observations :

```
auto %>% mutate(Numero = 1:nrow(auto)) %>% relocate(Numero, .before = Mpg) -> auto
auto %>% head()
```

```
## # A tibble: 6 x 11
##   Numero   Mpg Cylindres Cylindree Puissance Poids Acceleration Annee Origine
##    <int> <dbl>     <dbl>     <dbl>     <dbl> <dbl>        <dbl> <dbl> <chr>
## 1      1    18         8       307       130  3504         12    1970 Amérique
## 2      2    15         8       350       165  3693         11.5  1970 Amérique
## 3      3    18         8       318       150  3436         11    1970 Amérique
## 4      4    16         8       304       150  3433         12    1970 Amérique
## 5      5    17         8       302       140  3449         10.5  1970 Amérique
## 6      6    15         8       429       198  4341         10    1970 Amérique
## # ... with 2 more variables: Marque <chr>, Modele <chr>
```

Puis on va scinder le jeu de données en deux tables, une contenant les variables numériques ainsi que l'origine et l'autre contenant la marque, le modèle et de nouveau l'origine, cette dernière sera nommée *marque* :

```
marque <- matrix(rep(NA,nrow(auto)*4), ncol=4, nrow=nrow(auto))
marque <- as_tibble(marque)
marque %>% rename(Numero = V1, Marque = V2, Modele = V3, Origine = V4) -> marque
marque %>% mutate(Numero = auto$Numero,
                  Marque = auto$Marque,
                  Modele = auto$Modele,
                  Origine = auto$Origine) -> marque
auto %>% select(-c(Marque, Modele)) -> auto
head(auto)
```

```
## # A tibble: 6 x 9
##   Numero   Mpg Cylindres Cylindree Puissance Poids Acceleration Annee Origine
##    <int> <dbl>     <dbl>     <dbl>     <dbl> <dbl>        <dbl> <dbl> <chr>
## 1      1    18         8       307       130  3504         12    1970 Amérique
## 2      2    15         8       350       165  3693         11.5  1970 Amérique
## 3      3    18         8       318       150  3436         11    1970 Amérique
## 4      4    16         8       304       150  3433         12    1970 Amérique
## 5      5    17         8       302       140  3449         10.5  1970 Amérique
## 6      6    15         8       429       198  4341         10    1970 Amérique
```

```
head(marque)
```

```
## # A tibble: 6 x 4
##   Numero Marque    Modele          Origine
##    <int> <chr>     <chr>           <chr>
## 1      1 Chevrolet chevelle malibu Amérique
## 2      2 Buick     skylark 320     Amérique
## 3      3 Plymouth  satellite       Amérique
## 4      4 AMC       rebel sst       Amérique
## 5      5 Ford      torino          Amérique
## 6      6 Ford      galaxie 500     Amérique
```

On peut enfin se lancer dans notre analyse de données.

2 Analyse descriptive du jeu de données

2.1 Description des données

```
attach(auto)
attach(marque)
summary(auto)
```

```
##      Numero            Mpg          Cylindres       Cylindree
##  Min.   :  1.00   Min.   : 9.00   Min.   :3.000   Min.   : 68.0
##  1st Qu.: 98.75   1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0
##  Median :196.50   Median :22.75   Median :4.000   Median :151.0
##  Mean   :196.50   Mean   :23.45   Mean   :5.472   Mean   :194.4
##  3rd Qu.:294.25   3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8
##  Max.   :392.00   Max.   :46.60   Max.   :8.000   Max.   :455.0
##    Puissance         Poids       Acceleration       Annee
##  Min.   : 46.0   Min.   :1613   Min.   : 8.00   Min.   :1970
##  1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78   1st Qu.:1973
##  Median : 93.5   Median :2804   Median :15.50   Median :1976
##  Mean   :104.5   Mean   :2978   Mean   :15.54   Mean   :1976
##  3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:1979
##  Max.   :230.0   Max.   :5140   Max.   :24.80   Max.   :1982
##    Origine
##  Length:392
##  Class :character
##  Mode  :character
##
##
##
```

Une vue rapide de ces données nous montre :

- Une grande différence entre la consommation minimum qui est de 46.60 Mpg et la consommation maximum qui est de 9 Mpg.
- Une autre grande différence entre le poids minimum qui est de 1613 Lbs et le poids maximum qui est de 5140 Lbs.

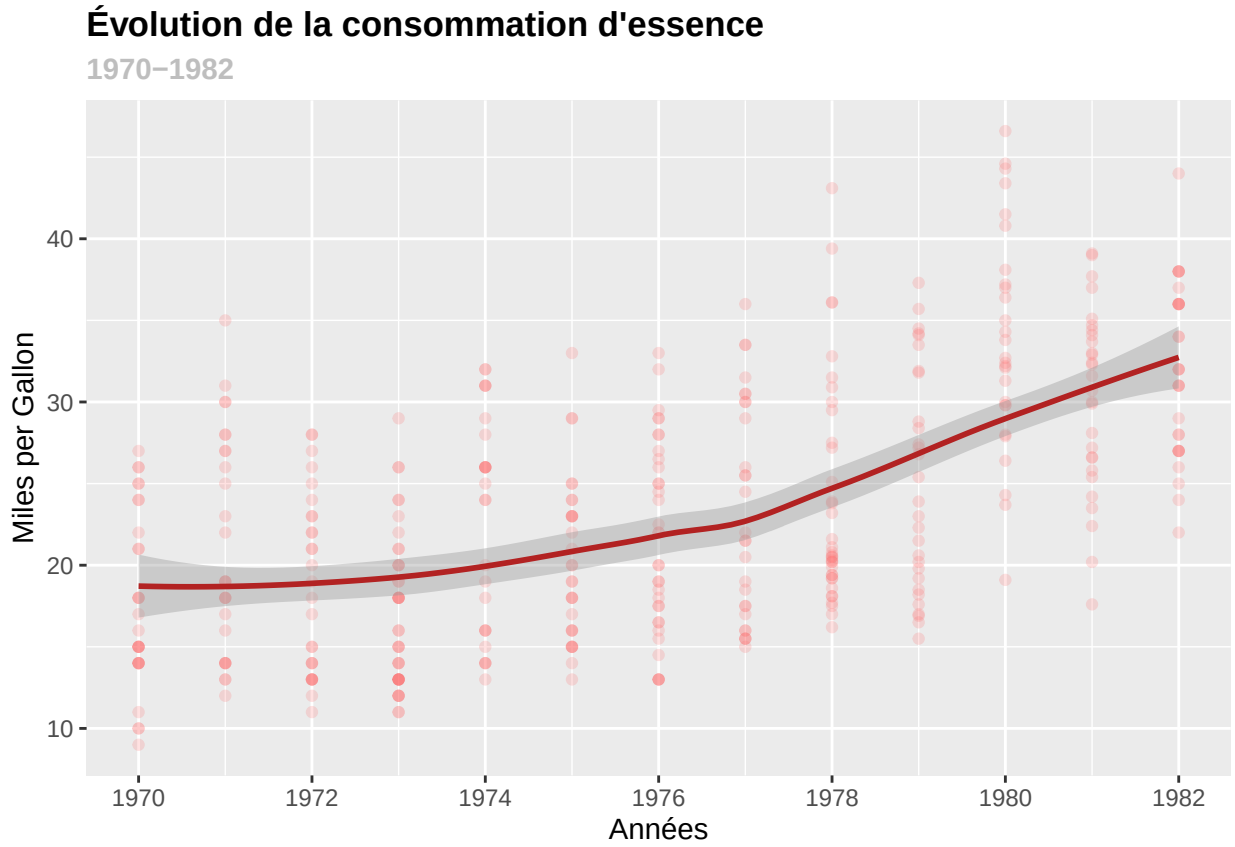
Cela montre à première vue que le jeu de données traite tout type de véhicules.

2.2 La variable *Mpg*

Dans cette analyse, nous allons nous concentrer sur la variable *Mpg*, on va commencer par avoir un premier point de vue concernant cette variable.
On va regarder le graphique de l'évolution des Miles per Gallon entre 1970 et 1982 en utilisant des fonctions issues de `ggplot` :

```
ggplot(auto,aes(Annee, Mpg))  +
  geom_point( color = "indianred1", alpha=0.2) +
  geom_smooth(col= "firebrick") +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  ggtitle("Évolution de la consommation d'essence",
          subtitle = "1970-1982") +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey")) +
  labs(x = "Années", y = "Miles per Gallon")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



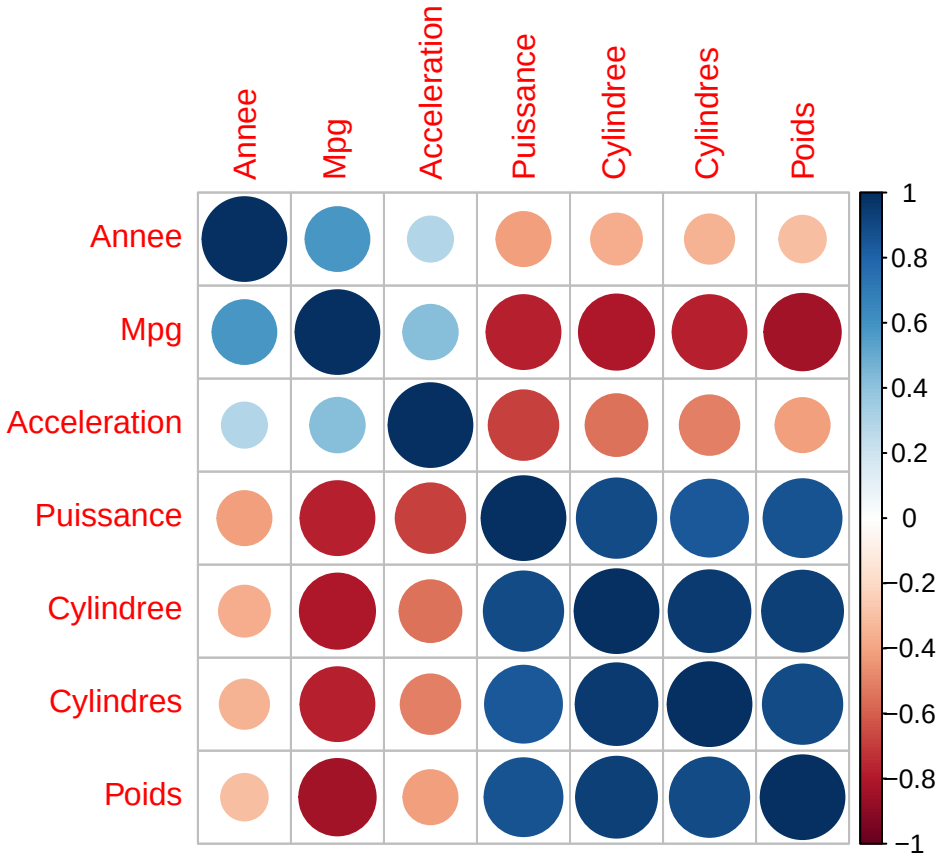
On voit que les véhicules passent d'en moyenne 18 Mpg en 1970 à en moyenne 32 Mpg en 1982.
Au fil du temps, les véhicules consomment moins d'essence. On va analyser cela de plus près.

2.3 Description des variables

2.3.1 Matrice de corrélation des variables

On va construire la matrice de corrélation pour savoir quelles variables sont liées et à quel point, pour cela on va utiliser la fonction `corrplot` :

```
auto %>% select(-c(Numero,Origine)) %>% cor() %>% corrplot(order = 'AOE')
```



On voit ici d'une part les fortes corrélations négatives entre *Mpg* et *Puissance*, *Mpg* et *Cylindree*, *Mpg* et *Cylindres* et enfin *Mpg* et *Poids* et d'autre part les fortes corrélations positives des variables *Puissance*, *Cylindree*, *Cylindres* et *Poids*.

2.3.2 Évolution des variables

On va maintenant faire la moyenne de chaque variable pendant chaque année, pour cela on va d'abord utiliser la fonction `group_by` pour grouper en année croissante puis la fonction `summarise` pour éffectuer nos calculs :

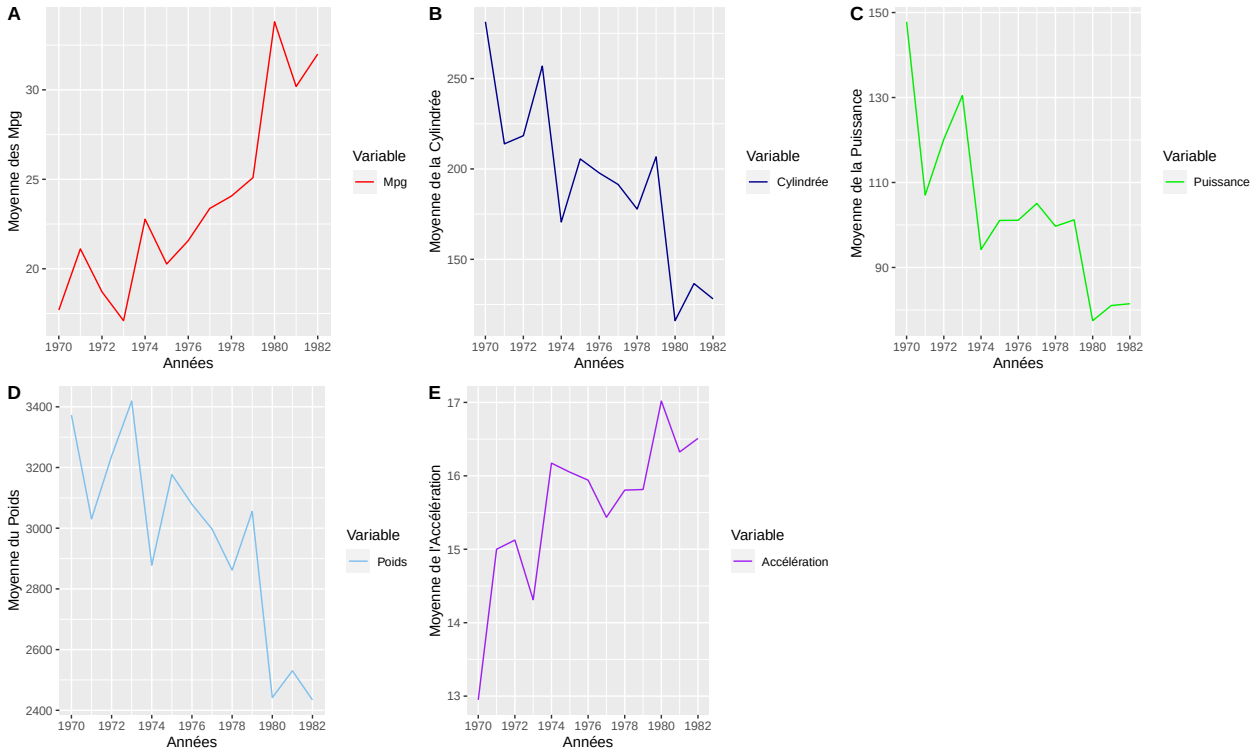
```
auto %>% group_by(Annee) %>%
  summarise(moy_mpg = mean(Mpg), moy_cylindree =  mean(Cylindree),
            moy_puissance = mean(Puissance), moy_poids = mean(Poids),
            moy_acceleration = mean(Acceleration)) -> moyenne_par_an
moyenne_par_an
```

```
## # A tibble: 13 x 6
##    Annee moy_mpg moy_cylindree moy_puissance moy_poids moy_acceleration
##    <dbl>   <dbl>         <dbl>         <dbl>     <dbl>            <dbl>
##  1  1970    17.7          281.         148.      3373.             12.9
##  2  1971    21.1          214.         107.      3031.             15
##  3  1972    18.7          218.         120.      3238.             15.1
##  4  1973    17.1          257.         130.      3419.             14.3
##  5  1974    22.8          171.          94.2     2878.             16.2
##  6  1975    20.3          206.         101.      3177.             16.0
##  7  1976    21.6          198.         101.      3079.             15.9
##  8  1977    23.4          191.         105.      2997.             15.4
##  9  1978    24.1          178.          99.7     2862.             15.8
## 10  1979    25.1          207.         101.      3055.             15.8
## 11  1980    33.8          116.          77.5     2442.             17.0
## 12  1981    30.2          137.          81.0     2530.             16.3
## 13  1982    32            128.          81.5     2434.             16.5
```

On peut représenter graphiquement ces résultats pour une meilleure compréhension avec la fonction `geom_line`:

```
ggplot(moyenne_par_an) +
  geom_line(aes(x = Annee, y = moy_mpg,color='Mpg')) +
  scale_color_manual(values = c('Mpg' = 'red')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne des Mpg") -> A
ggplot(moyenne_par_an) +
  geom_line(aes(x = Annee, y = moy_cylindree,color='Cylindrée')) +
  scale_color_manual(values = c('Cylindrée' = 'darkblue')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable')  +
  labs(x = "Années", y = "Moyenne de la Cylindrée") -> B
ggplot(moyenne_par_an) +
  geom_line(aes(x = Annee, y = moy_puissance,color='Puissance')) +
  scale_color_manual(values = c('Puissance' = 'green2')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne de la Puissance ") -> C
ggplot(moyenne_par_an) +
  geom_line(aes(x = Annee, y = moy_poids,color='Poids')) +
  scale_color_manual(values = c('Poids' = 'skyblue2')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne du Poids") -> D
ggplot(moyenne_par_an) +
  geom_line(aes(x = Annee, y = moy_acceleration,color='Accélération')) +
```

```
  scale_color_manual(values = c('Accélération' = 'purple')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne de l'Accélération") -> E

plot_grid(A, B, C, D, E, labels=c("A", "B", "C", "D", "E"), ncol = 3, nrow = 2)
```



On voit que l'évolution de chaque variable quantitative respecte bien la matrice de corrélation étant donnée que l'on à des fortes baisses de *Puissance*, *Poids* et *Cylindree* au fil des années (elles sont deux à deux corrélées positivement) et une forte diminution de la consommation.

3 Analyse des variables

3.1 La variable *Origine*

On regarde maintenant le nombre de véhicules total pour chaque *Origine* car l'origine est l'autre variable importante dans ce jeu de données:

```
auto %>% group_by(Origine) %>%
  count(Origine) %>% arrange(n)
```

```
## # A tibble: 3 x 2
## # Groups:   Origine [3]
##   Origine      n
##   <chr>    <int>
## 1 Europe      68
## 2 Asie        79
## 3 Amérique   245
```

La grande majorité du jeu de données est composé de véhicules américains.
Ensuite on va regarder le nombre moyen de *Mpg* en fonction de l' *Origine*:

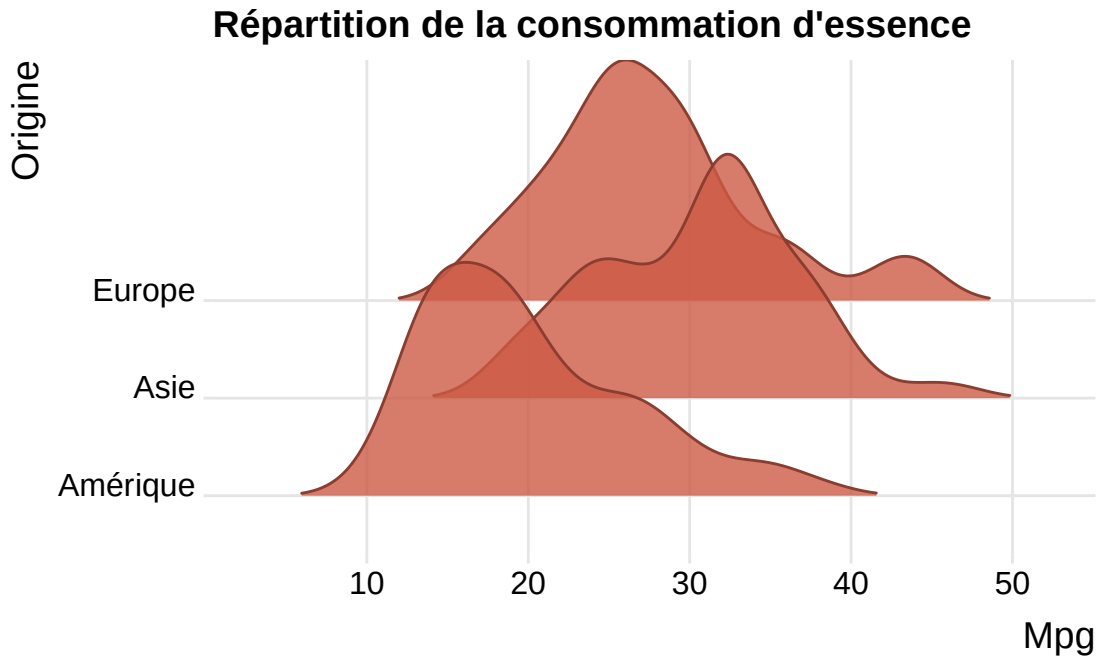
```
auto %>% group_by(Origine) %>%
  summarise(moy_par_origine = mean(Mpg)) %>% arrange(moy_par_origine)
```

```
## # A tibble: 3 x 2
##   Origine  moy_par_origine
##   <chr>              <dbl>
## 1 Amérique            20.0
## 2 Europe              27.6
## 3 Asie                30.5
```

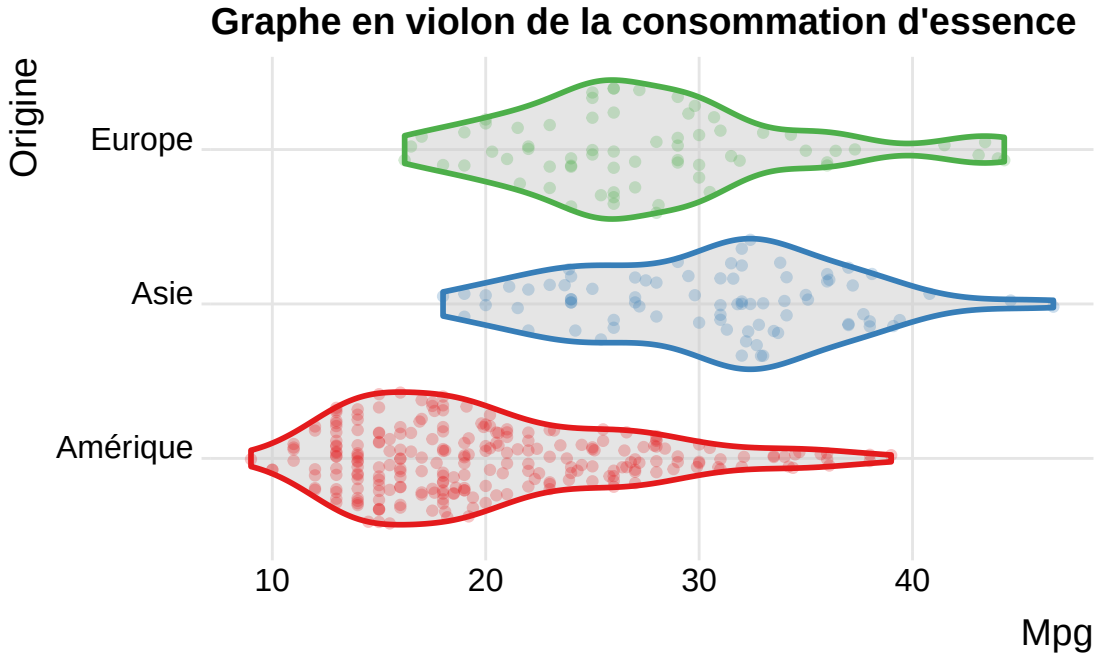
On peut représenter cela de deux façons différentes notamment grâce au fonction `geom_density_ridges` et `geom_violin` :

```
ggplot(auto, aes(x = Mpg, y = factor(Origine))) +
  geom_density_ridges(alpha = .8, color = "coral4",
                      scale = 2.5, rel_min_height = .01, fill = "coral3") +
  labs(x = "Mpg", y = "Origine") +
  ggtitle("Répartition de la consommation d'essence") +
  theme(plot.title = element_text(face = "bold")) +
  theme_ridges()
```

```
## Picking joint bandwidth of 2.02
```




```
ggplot(auto, aes(x = Origine, y = Mpg, color = Origine)) +
  labs(x = "Origine", y = "Mpg") +
  scale_color_brewer(palette = "Set1", guide = "none") +
  geom_violin(fill = "gray80", size = 1, alpha = .5) +
  geom_sina(alpha = .25) +
  ggtitle("Graphe en violon de la consommation d'essence") +
  theme(plot.title = element_text(face = "bold")) +
  theme_ridges() + coord_flip()
```



C'est les véhicules américains qui consomment le plus, avec une moyenne de 20.03347 Mpg.
Mais on peut se demander si cette moyenne ne serait pas biaisée au vue de la forte proportion de véhicules américains dans le jeu de données.

3.2 La variable *Cylindre*

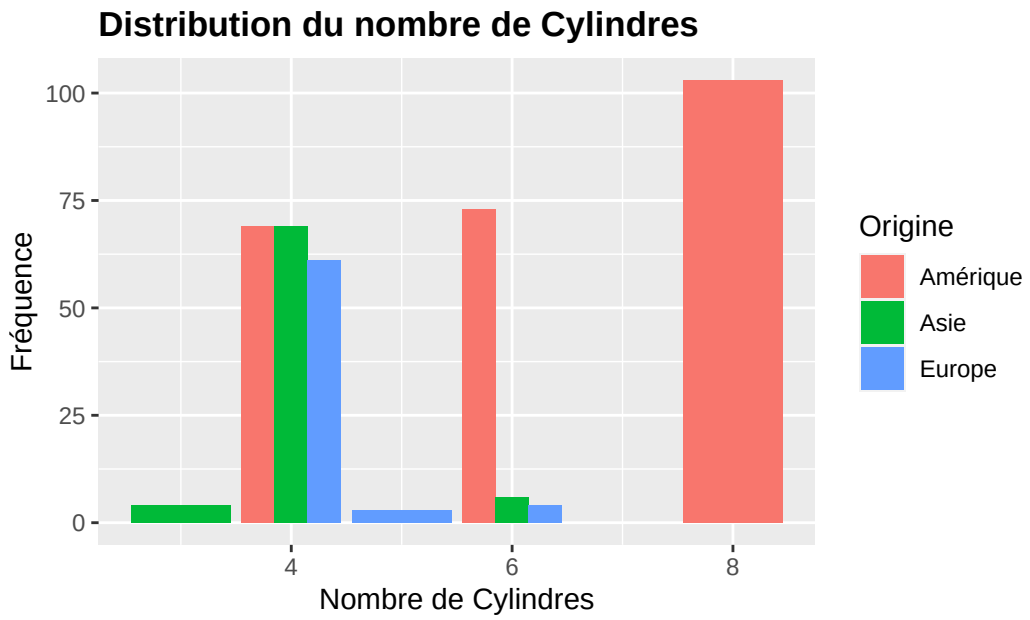
C'est pour cela que l'on va commencer par compter le nombre de véhicules selon leur *Origine* et leur nombre de *Cylindres*.
La variable *Cylindres* nous permet d'avoir un début d'interprétation car elle est fortement corrélée avec les autres variables importantes.

```
auto %>% arrange(Cylindres) %>% group_by(Cylindres) %>% count(Origine) %>%
  rename(nbr_voiture = n)
```

```
## # A tibble: 9 x 3
## # Groups:   Cylindres [5]
##   Cylindres Origine  nbr_voiture
##       <dbl> <chr>          <int>
## 1         3 Asie               4
## 2         4 Amérique          69
## 3         4 Asie              69
## 4         4 Europe            61
## 5         5 Europe             3
## 6         6 Amérique          73
## 7         6 Asie               6
## 8         6 Europe             4
## 9         8 Amérique         103
```

Les véhicules américains sont présent un peu partout mais surtout à 8 cylindres où il y en a 103 ; tandis que la majorité des véhicules asiatiques et européens possèdent 4 cylindres.
On peut représenter graphiquement ce résultat notamment avec la fonction `geom_bar` :

```
ggplot(auto) + geom_bar(aes(x = Cylindres, fill = Origine), position = "dodge") +
  theme(plot.title = element_text(face = "bold")) +
  ggtitle("Distribution du nombre de Cylindres") +
  theme(plot.title = element_text(face = "bold")) +
  labs(x = "Nombre de Cylindres",y = "Fréquence")
```



On peut aussi calculer la moyenne des *Mpg* par *Origine* en fonction des *Cylindres* on a :

```
auto %>%
  group_by(Cylindres, Origine) %>%
  summarise(moy_Mpg = mean(Mpg))
```

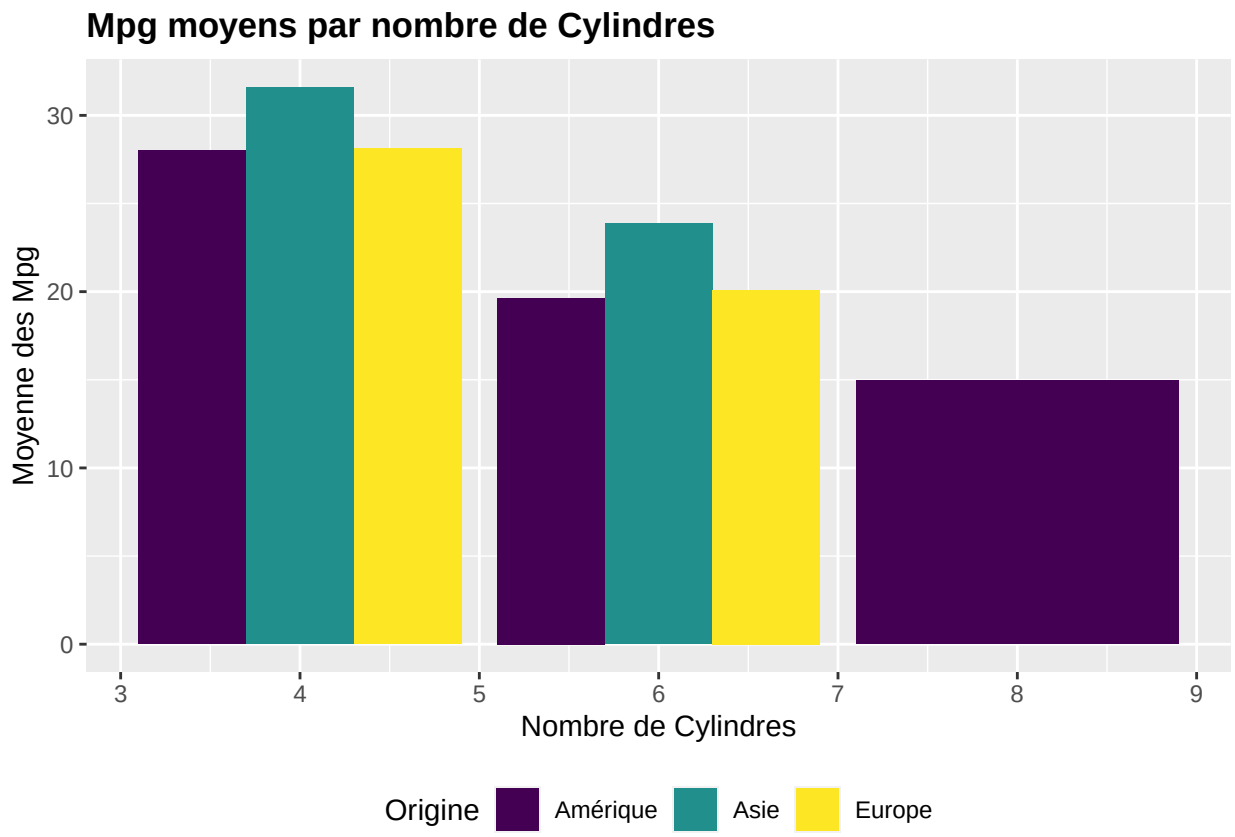
```
## `summarise()` has grouped output by 'Cylindres'. You can override using the `.groups` argument.
```

```
## # A tibble: 9 x 3
## # Groups:   Cylindres [5]
##   Cylindres Origine  moy_Mpg
##       <dbl> <chr>      <dbl>
## 1         3 Asie        20.6
## 2         4 Amérique    28.0
## 3         4 Asie        31.6
## 4         4 Europe      28.1
## 5         5 Europe      27.4
## 6         6 Amérique    19.6
## 7         6 Asie        23.9
## 8         6 Europe      20.1
## 9         8 Amérique    15.0
```

On peut représenter ces résultats graphiquement avec encore une fois la fonction `geom_bar` et on ne prend pas en compte les véhicules possèdant 3 et 5 cylindres car peu représentatifs :

```
auto %>% filter(Cylindres != 3 & Cylindres != 5) %>%
  group_by(Cylindres, Origine) %>%
  summarise(moy_Mpg = mean(Mpg))%>%
  ggplot()+
  geom_bar(mapping = aes(x = Cylindres, y = moy_Mpg, fill = Origine),
           stat = "identity",position = 'dodge')+
  scale_fill_viridis_d(option  = "viridis")+
  theme(legend.position = 'bottom') +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey")) +
  ggtitle("Mpg moyens par nombre de Cylindres") +
  labs(x = "Nombre de Cylindres", y = "Moyenne des Mpg")
```

```
## `summarise()` has grouped output by 'Cylindres'. You can override using the `.groups` argument.
```



On remarque que plus le véhicule possèdent de cylindres plus sa consomamtion est grande, ce qui est bien en accord avec la matrice de corrélation.

3.3 La variable *Poids*

On va maintenant s'intéresser à la variable *Poids*, tout d'abord on à décidé de créer des catégories de poids à partir des quantiles de la variable pour une meilleure analyse, ceci grâce à la fonction `case_when` :

```
summary(Poids)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1613    2225    2804    2978    3615    5140
```

```
categorie_poids <- case_when(auto$Poids < 2226 ~ "Faible",
                             auto$Poids >= 2226 & auto$Poids < 2805 ~ "Moyen",
                             auto$Poids >= 2805 & auto$Poids < 3616  ~ "Lourd",
                             auto$Poids >= 3616 ~ "Très lourd") %>%
  fct_relevel(c("Faible", "Moyen","Lourd","Très lourd"))
```

Ensuite on va dénombrer les véhicules en fonction de leur catégorie de poids :

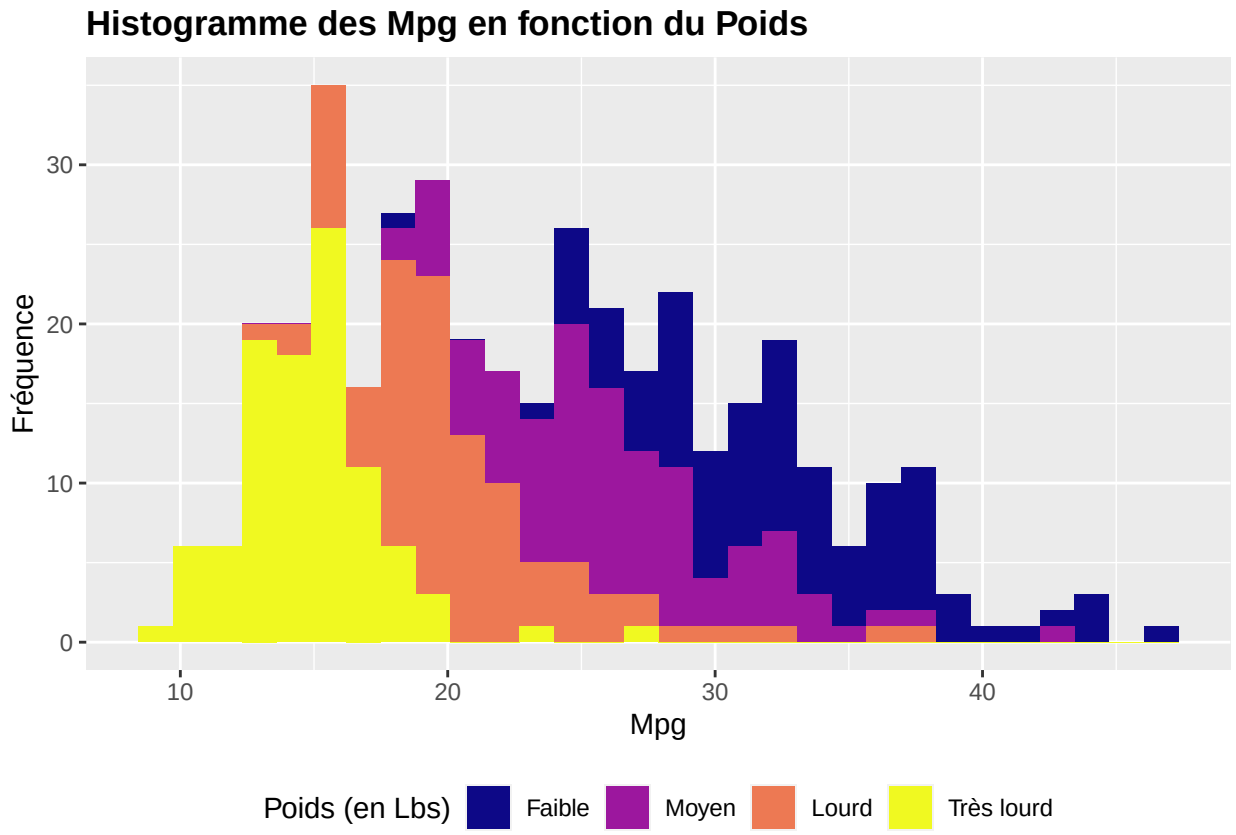
```
bind_cols(Numero = Numero, categorie_poids = categorie_poids, Origine = Origine) %>%
  group_by(categorie_poids) %>% count(Origine) %>% rename(nbr_voiture = n)
```

```
## # A tibble: 11 x 3
## # Groups:   categorie_poids [4]
##    categorie_poids Origine  nbr_voiture
##    <fct>           <chr>          <int>
##  1 Faible          Amérique          21
##  2 Faible          Asie              44
##  3 Faible          Europe            33
##  4 Moyen           Amérique          51
##  5 Moyen           Asie              29
##  6 Moyen           Europe            18
##  7 Lourd           Amérique          76
##  8 Lourd           Asie               6
##  9 Lourd           Europe            16
## 10 Très lourd      Amérique          97
## 11 Très lourd      Europe             1
```

et enfin on va repésenter cela graphiquement en construisant l'histogramme des Mpg avec les différentes catégories de poids grâce à la fonction `geom_histogram` :

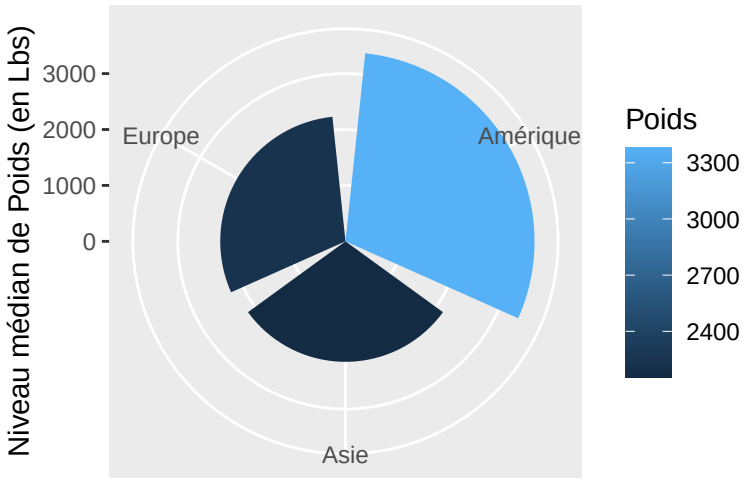
```
bind_cols(Numero = Numero, Mpg = Mpg, categorie_poids = categorie_poids) -> Mpg_cat_Poi
ggplot(Mpg_cat_Poi, mapping = aes( x= Mpg, fill = categorie_poids))+
  geom_histogram()+
  scale_fill_viridis_d(option = "plasma")+
  ggtitle("Histogramme des Mpg en fonction du Poids")+
  theme(legend.position = 'bottom') +
  theme(plot.title = element_text(face = "bold")) +
  labs(x = "Mpg", y = "Fréquence", fill = 'Poids (en Lbs)') +
  theme(legend.position = 'bottom')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Sans surprise, les véhicules considérés comme “très lourd” sont ceux qui consomment le plus, suivis des “véhicules lourd”, des véhicules “moyennement lourd” et enfin des véhicules “léger”.
La quasi-totalité des véhicules “très lourd” sont américains encore une fois.
On le voit très bien dans le graphique suivant :

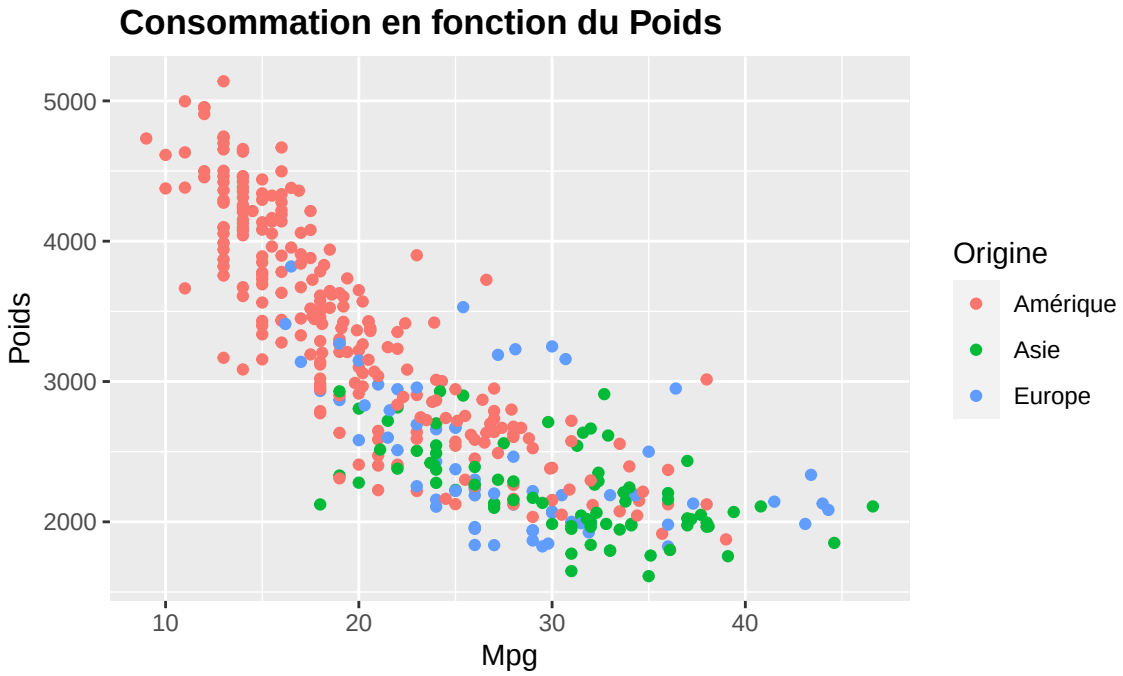
```
auto %>%
  group_by(Origine) %>%
  summarise(Poids = median(Poids)) %>%
  ggplot(aes(x = Origine, y = Poids)) +
  geom_col(aes(fill = Poids), color = NA) +
  labs(x = "", y = "Niveau médian de Poids (en Lbs)") +
  coord_polar()
```



3.4 Les autres variables

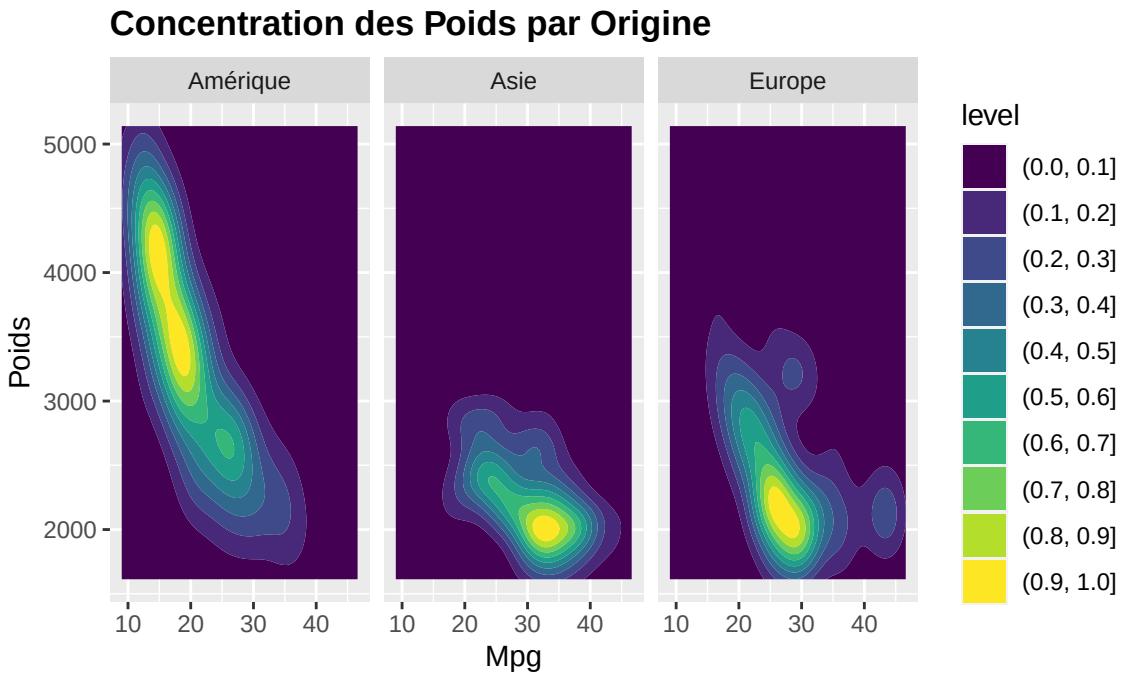
On peut regarder le nuage de points des *Mpg* en fonction du *Poids* avec la fonction `geom_point` :

```
ggplot(auto) +
  geom_point(mapping = aes(x = Mpg, y = Poids, color= Origine)) +
  labs(x = 'Mpg', y = 'Poids') +
  ggtitle(' Consommation en fonction du Poids ') +
  theme(plot.title = element_text(face = "bold"))
```

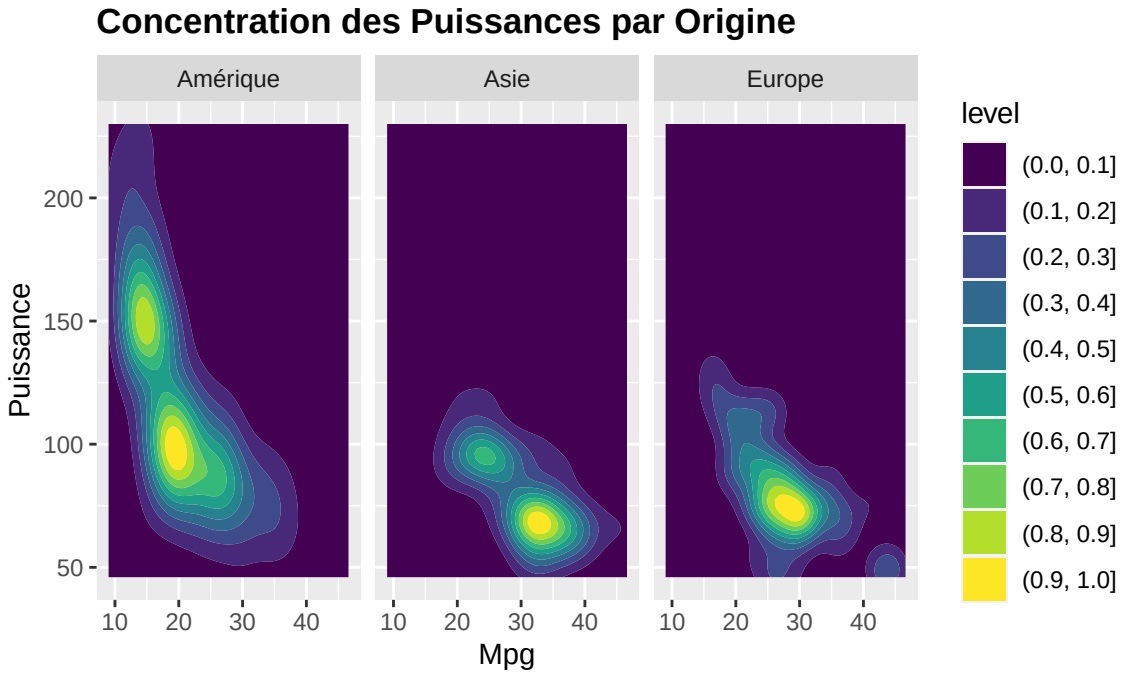


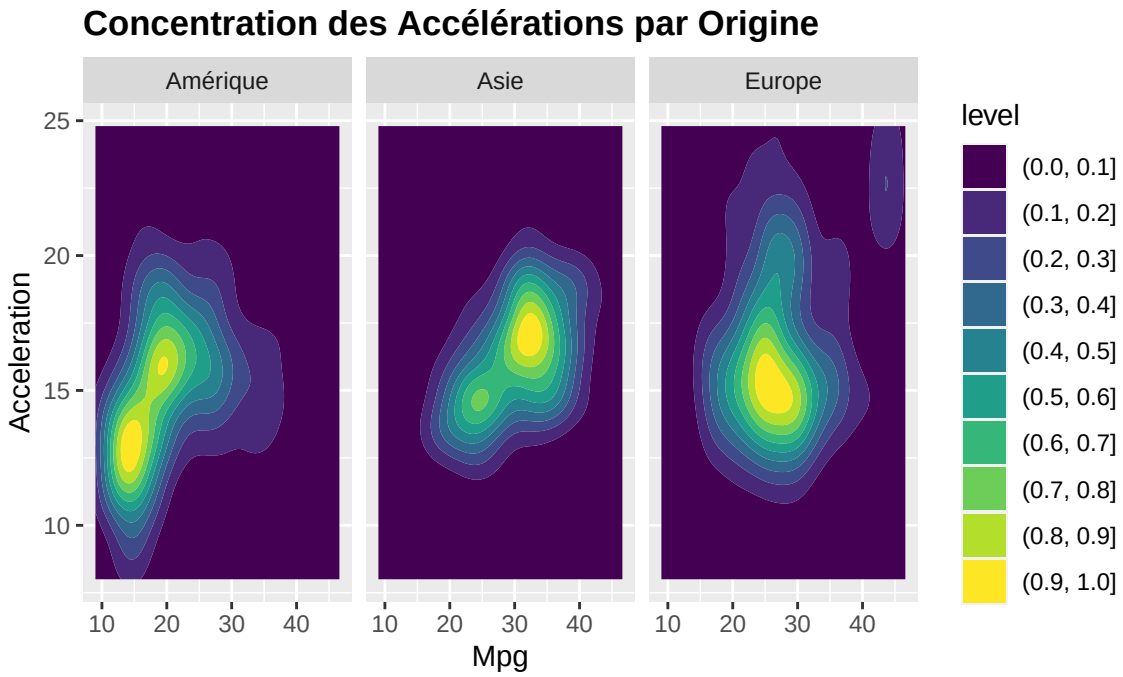
Une autre façon plus agréable de visualiser cela est la visualisation 2d grâce à la fonction `geom_density_2d` :

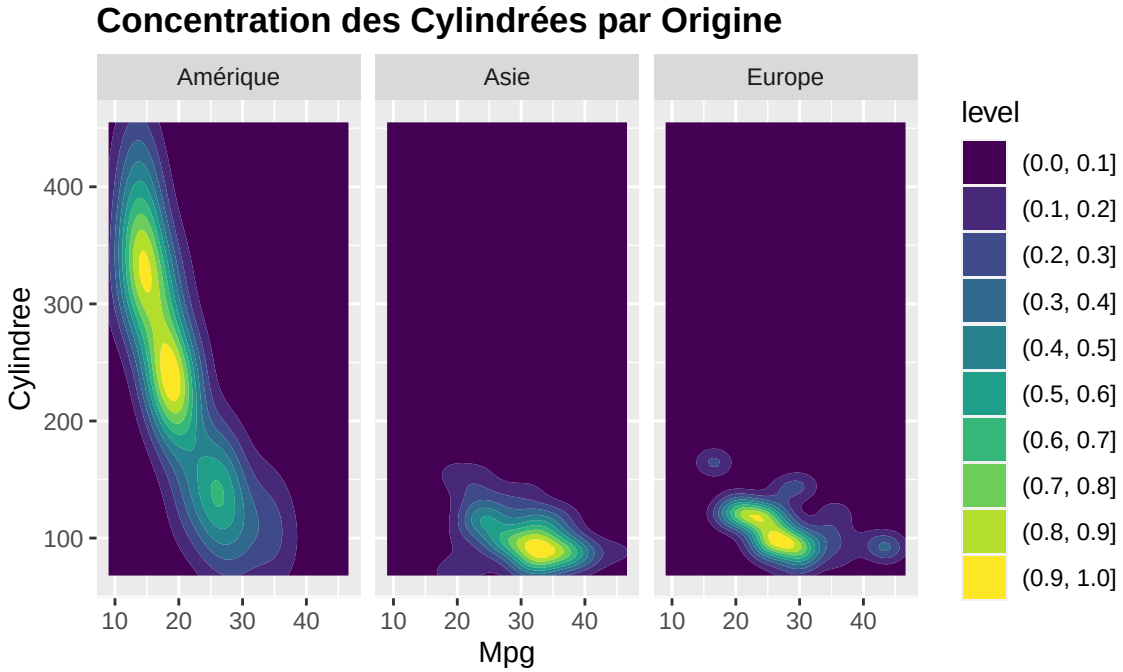
```
auto %>%
  ggplot(aes(x = Mpg, y = Poids)) +
  geom_density_2d_filled(contour_var = "ndensity") +
  facet_wrap(vars(Origine)) +
  ggtitle('Concentration des Poids par Origine') +
  theme(plot.title = element_text(face = "bold"))
```



On a séparé les graphes en fonction de l'origine et avec cette méthode on voit où les véhicules de chaque origine sont le plus concentrés ; on voit ici que les véhicules américains sont très concentrés entre 3000 et 4500 Lbs alors que les véhicules asiatiques et européens sont plus ou moins concentrés au même endroit à environ 2000 Lbs.
Dans la suite on va faire une analyse similaire des variables *Puissance*, *Acceleration* et *Cylindree* en les représentant par le graphique en 2d que l'on à considéré plus lisible que le nuage de point.
On a décidé de ne pas afficher les lignes de commande dans le pdf/html car les codes sont semblables à celui fait précédemment sur le *Poids* (on a juste à changé la valeur de y par *Puissance*, *Acceleration* et *Cylindree*)







On remarque que les variables *Puissance* et de *Cylindree* suivent le même schéma c'est à dire que les véhicules américains ont toujours plus de puissance et de cylindrée que les véhicules asiatiques et européens.
Seule la variable *Acceleration* semble être plus ou moins répartie de la même façon quelque soit l'origine.
On peut émettre l'hypothèse que l'accélération d'un véhicule n'influe pas directement sur la consommation.

On va calculer la moyenne de l'accélération par origine :

```
auto %>%
  group_by(Origine) %>%
  summarise(moy_acceleration = mean(Acceleration),
            nbr_voiture = n())
```

```
## # A tibble: 3 x 3
##   Origine  moy_acceleration nbr_voiture
##   <chr>               <dbl>       <int>
## 1 Amérique             15.0         245
## 2 Asie                 16.2          79
## 3 Europe               16.8          68
```

On voit bien que les véhicules américains, asiatiques et européens ont quasiment la même accélération moyenne.

3.5 À propos des marques des véhicules

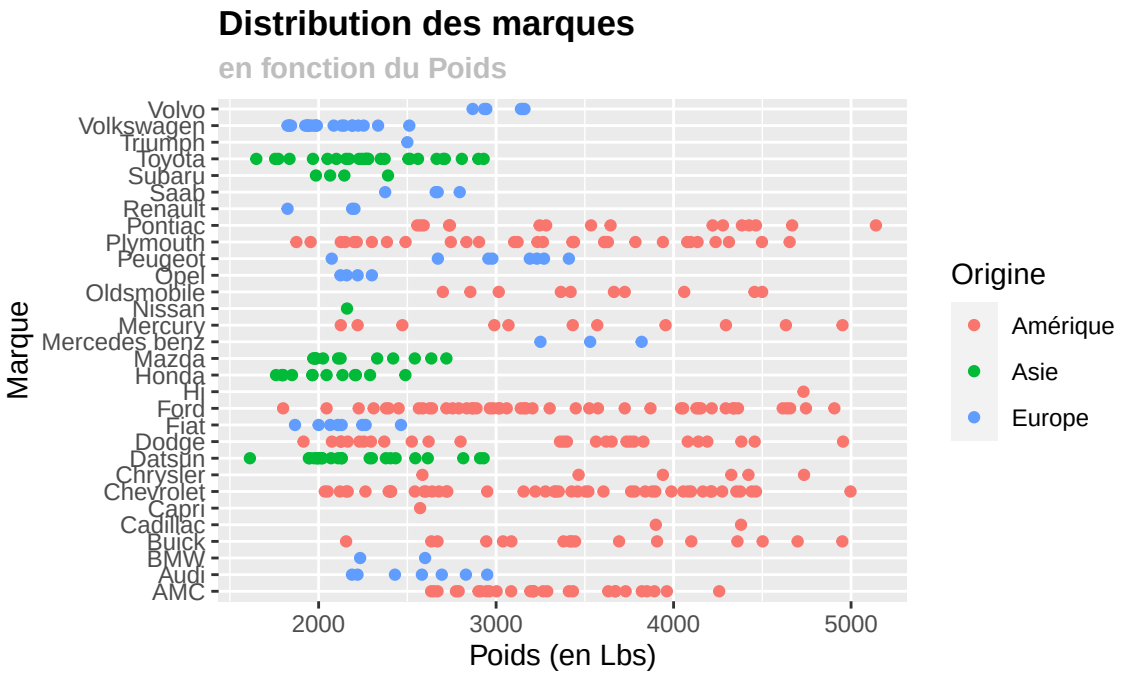
On va maintenant faire l'analyse des caractéristiques mais cette fois ci avec les marques des véhicules.
Tout d'abord on va créer une nouvelle table avec l'ensemble des données de la table *auto* et la variable *Marque* de la table *marque* :

```
bind_cols(auto,Marque = Marque) -> auto_Marque
auto_Marque
```

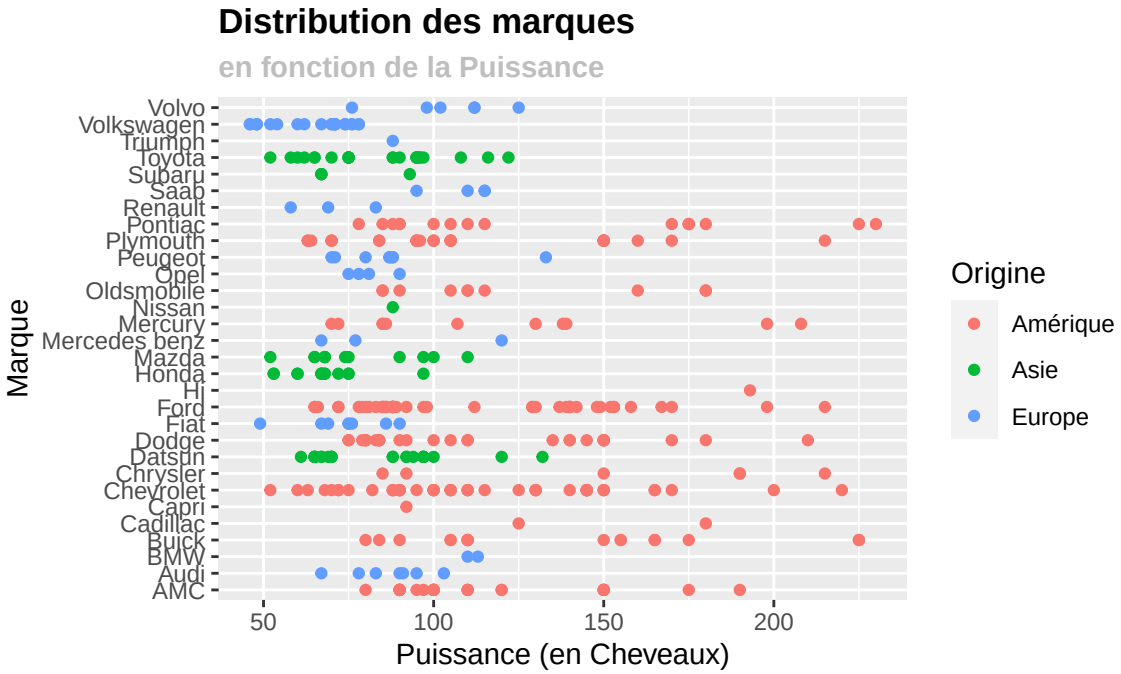
```
## # A tibble: 392 x 10
##    Numero   Mpg Cylindres Cylindree Puissance Poids Acceleration Annee Origine
##     <int> <dbl>     <dbl>     <dbl>     <dbl> <dbl>        <dbl> <dbl> <chr>
##  1      1    18         8       307       130  3504         12    1970 Amérique
##  2      2    15         8       350       165  3693         11.5  1970 Amérique
##  3      3    18         8       318       150  3436         11    1970 Amérique
##  4      4    16         8       304       150  3433         12    1970 Amérique
##  5      5    17         8       302       140  3449         10.5  1970 Amérique
##  6      6    15         8       429       198  4341         10    1970 Amérique
##  7      7    14         8       454       220  4354          9    1970 Amérique
##  8      8    14         8       440       215  4312          8.5  1970 Amérique
##  9      9    14         8       455       225  4425         10    1970 Amérique
## 10     10    15         8       390       190  3850          8.5  1970 Amérique
## # ... with 382 more rows, and 1 more variable: Marque <chr>
```

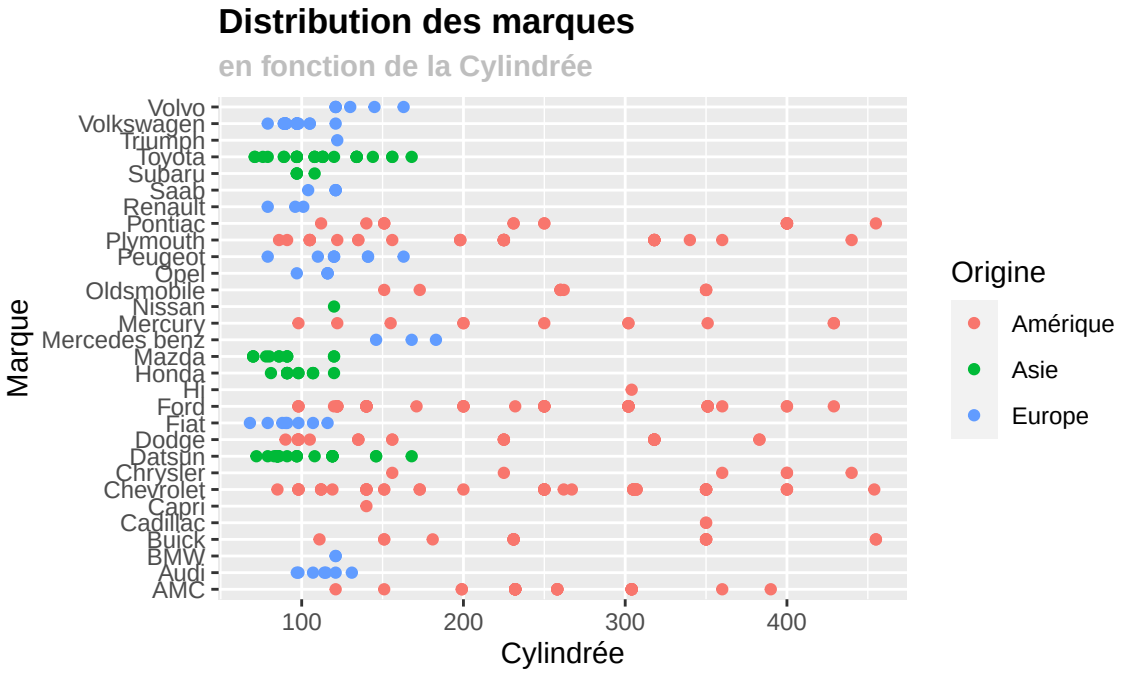
Cette nouvelle table s'appelle *auto_Marque*.
Puis à l'aide d'un `geom_point` on va construire un graphique qui va nous montrer la répartition de tous les véhicules de chaque marque en fonction de leurs caractéristiques.
On commence par exemple par le *Poids* :

```
ggplot(auto_Marque) +
  geom_point(mapping = aes(x = Poids, y = Marque, color = Origine)) +
  labs( x = 'Poids (en Lbs)', y = 'Marque') +
  ggtitle('Distribution des marques', subtitle = "en fonction du Poids") +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey"))
```



Puis comme précédemment, on va construire le même graphique pour la *Puissance* et la *Cylindree* sans afficher les commandes (car similaire à celle ci dessus), on a :





Grâce à ces graphiques on a une vision plus détaillée des résultats précédents car on regarde vraiment les observations et leurs répartions et non plus les moyennes.
L'interprétation de ces graphiques est la même que pour les analyses précédentes.

Par ailleurs on remarque que certaines marques sont présentent plusieurs fois dans le jeu de données comme par exemple *Chevrolet* ou *Ford*.

On va dénombrer le nombre de véhicule en fonction de leur marque :

```
marque %>% count(Marque) %>% rename(nbr_voiture = n) -> Marque_count
Marque_count %>% arrange(desc(nbr_voiture))
```

```
## # A tibble: 30 x 2
##    Marque     nbr_voiture
##    <chr>            <int>
##  1 Ford                48
##  2 Chevrolet           47
##  3 Plymouth            31
##  4 Dodge               28
##  5 AMC                 27
##  6 Toyota              26
##  7 Datsun              23
##  8 Volkswagen          22
##  9 Buick               17
## 10 Pontiac             16
## # ... with 20 more rows
```

Les véhicules de chez *Ford* sont ceux qui reviennent le plus dans le jeu de données, on pense qu'une petite étude sur les observations de la marque *Ford* serait intéréssante pour savoir comment la marque à fait évoluer ses véhicules en 13 ans.

3.6 *Ford*, un exemple représentatif

On va d'abord voir si les véhicules *Ford* sont présent chaque année :

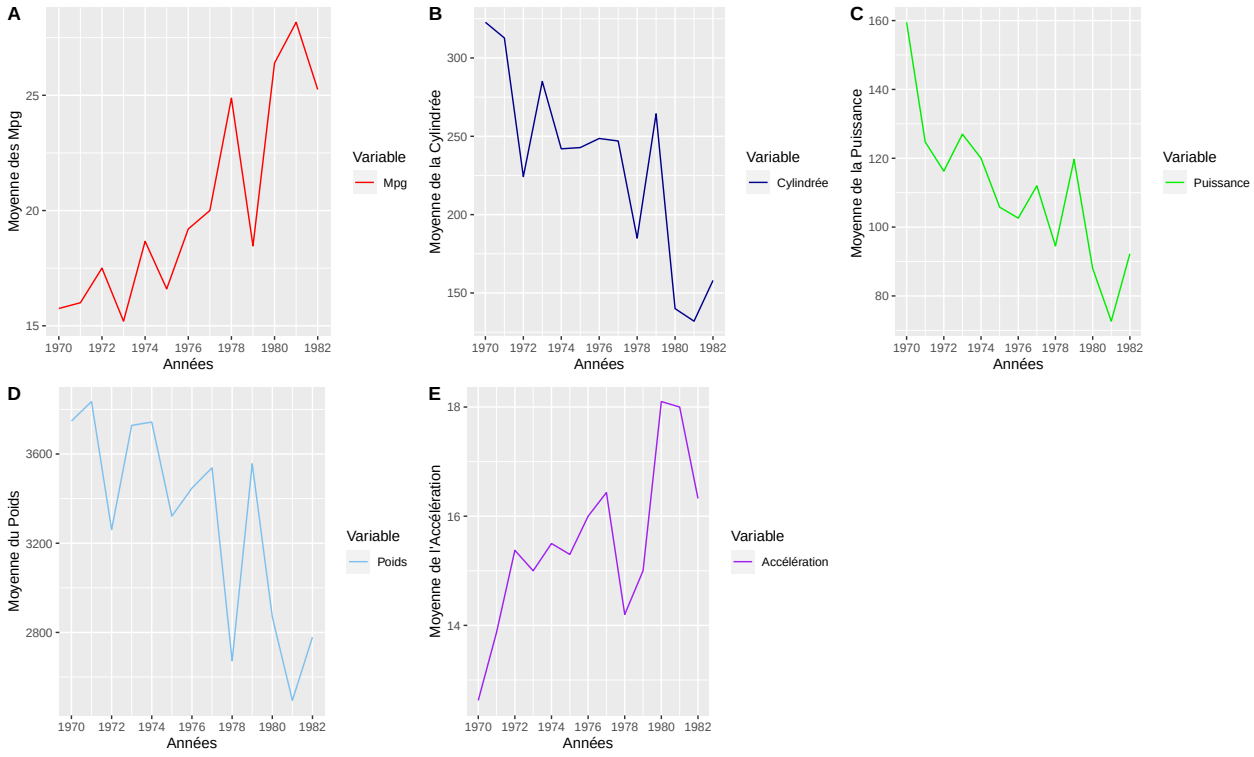
```
auto_Marque %>% filter(Marque == 'Ford') %>% group_by(Annee) %>% count(Marque) %>%
  rename(nbr_voiture = n)
```

```
## # A tibble: 13 x 3
## # Groups:   Annee [13]
##    Annee Marque nbr_voiture
##    <dbl> <chr>        <int>
##  1  1970 Ford             4
##  2  1971 Ford             4
##  3  1972 Ford             4
##  4  1973 Ford             5
##  5  1974 Ford             3
##  6  1975 Ford             5
##  7  1976 Ford             5
##  8  1977 Ford             3
##  9  1978 Ford             4
## 10  1979 Ford             3
## 11  1980 Ford             1
## 12  1981 Ford             3
## 13  1982 Ford             4
```

C'est bien le cas.
On va ensuite faire la même étude qu'en début de projet c'est à dire une suite de graphiques avec chaque caractéristique des véhicules *Ford* au fil du temps :

```
auto_Marque %>% filter(Marque == 'Ford') %>% group_by(Annee) %>%
  summarise(moy_mpg = mean(Mpg), moy_cylindree =  mean(Cylindree),
            moy_puissance = mean(Puissance), moy_poids = mean(Poids),
            moy_acceleration = mean(Acceleration)) -> moyenne_par_an_ford

ggplot(moyenne_par_an_ford) +
  geom_line(aes(x = Annee, y = moy_mpg,color='Mpg')) +
  scale_color_manual(values = c('Mpg' = 'red')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne des Mpg") -> A
ggplot(moyenne_par_an_ford) +
  geom_line(aes(x = Annee, y = moy_cylindree,color='Cylindrée')) +
  scale_color_manual(values = c('Cylindrée' = 'darkblue')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable')  +
  labs(x = "Années", y = "Moyenne de la Cylindrée") -> B
ggplot(moyenne_par_an_ford) +
  geom_line(aes(x = Annee, y = moy_puissance,color='Puissance')) +
  scale_color_manual(values = c('Puissance' = 'green2')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne de la Puissance ") -> C
ggplot(moyenne_par_an_ford) +
  geom_line(aes(x = Annee, y = moy_poids,color='Poids')) +
  scale_color_manual(values = c('Poids' = 'skyblue2')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
```

```
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne du Poids") -> D
ggplot(moyenne_par_an_ford) +
  geom_line(aes(x = Annee, y = moy_acceleration,color='Accélération')) +
  scale_color_manual(values = c('Accélération' = 'purple')) +
  scale_x_continuous(breaks=seq(1970,1982,2)) +
  labs(color = 'Variable') +
  labs(x = "Années", y = "Moyenne de l'Accélération") -> E

plot_grid(A, B, C, D, E, labels=c("A", "B", "C", "D", "E"), ncol = 3, nrow = 2)
```



On voit bien que les caractéristiques des véhicules *Ford* suivent la même tendance que la moyenne des véhicules du jeu de données sur la durée (1970-1982).
On pense qu'au fil du temps, les constructeurs ont choisi d'améliorer leurs véhicules de tel sorte que ces derniers consomment moins de carburant.
Parmi les nombreuses hypothèses possibles pour expliquer le choix des constructeurs, on pourrait citer les deux chocs pétroliers qui interviennent au début et à la fin des années 70.

4 Exploration des données textuelles

4.1 Fréquence des marques et wordcloud

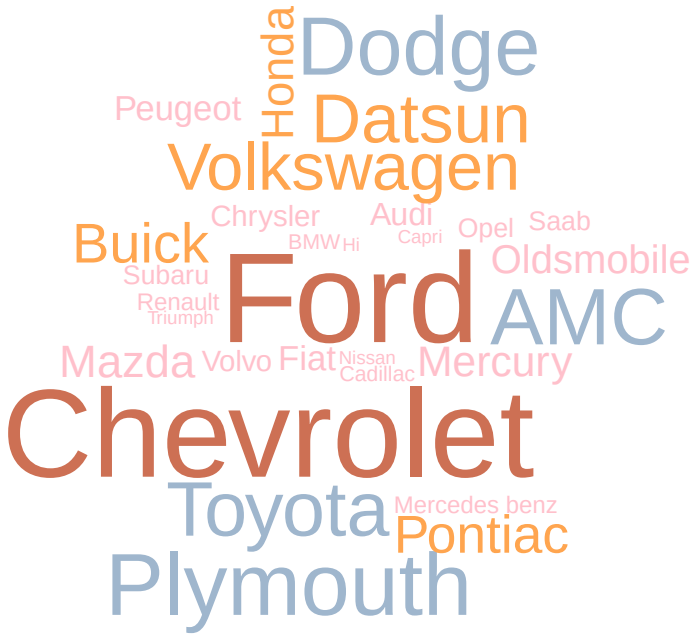
Pour finir cette étude, on va analyser les marques et les modèles des véhicules.
On reprend la table *Marque_Count* faite précédemment:

```
marque %>% count(Marque) -> Marque_count
Marque_count
```

```
## # A tibble: 30 x 2
##    Marque        n
##    <chr>     <int>
##  1 AMC          27
##  2 Audi          7
##  3 BMW           2
##  4 Buick        17
##  5 Cadillac      2
##  6 Capri         1
##  7 Chevrolet    47
##  8 Chrysler      6
##  9 Datsun       23
## 10 Dodge        28
## # ... with 20 more rows
```

On peut faire une représentation graphique de ce calcul, notamment grâce aux fonctions du package `wordcloud` qui vont nous afficher les marques les plus présentent de ce jeu de données :

```
wordcloud(words = Marque_count$Marque,
          freq = Marque_count$n,
          min.freq = 1,
          max.words=100,
          colors = c("pink","tan1", "slategray3", "salmon3"))
```

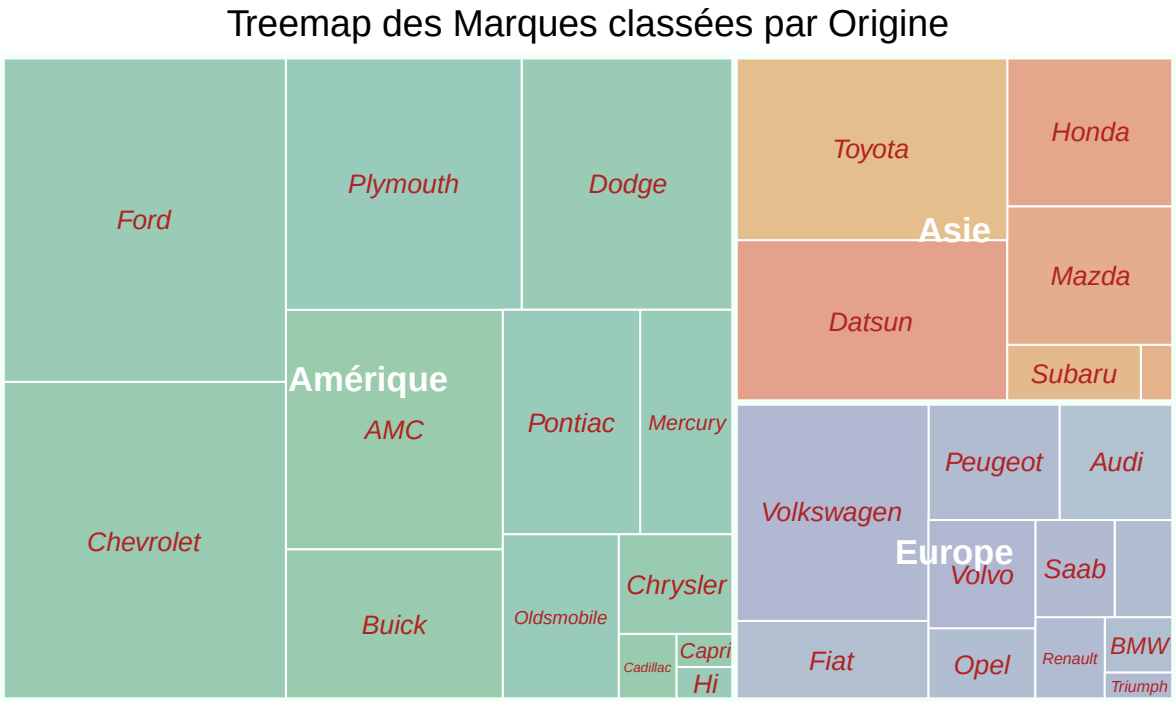


On voit que les marques qui reviennent le plus souvent sont *Chevrolet*, *Ford* ou encore *Plymouth*.

4.2 Treemap et Circle Plot en fonction de la consommation

Une autre façon plus précise de représenter cela est d'utiliser les fonctions du package `treemap` où on peut afficher les fréquences de chaque marque en les classant par origine :

```
auto_Marque %>% group_by(Origine) %>% count(Marque) %>%
  rename(nbr_voiture = n) -> auto_Marque_Or
treemap(auto_Marque_Or, index=c('Origine','Marque'),vSize='nbr_voiture',
        palette = 'Pastel2', type="index",
        border.col = c("mintcream","white"),
        fontsize.labels=c(13,10),
        fontcolor.labels=c("white","firebrick"),
        fontface.labels=c(2,3),
        bg.labels=c("transparent"),
        title ="Treemap des Marques classées par Origine")
```



On voit très bien les véhicules qui reviennent le plus dans le jeu de données selon leur origine, on a déjà vu que *Ford* était la marque américaine la plus redondante, et nous avons *Toyota* pour l'Asie et *Volkswagen* pour l'Europe.

On s'interésse maintenant aux modèles des voitures, on va chercher à afficher les modèles de voitures qui consomment le moins d'essence.
Pour cela on a commencer par créer une table contenant les *Mpg* et les variables de *marque* :

```
bind_cols(marque, Mpg = Mpg) -> auto_Marque_Modele
```

la table s'appelle *auto_Marque_Modele*.

On va grouper la table par *Marque* et *Modele* et arranger par ordre décroissant en fonction des *Mpg*.

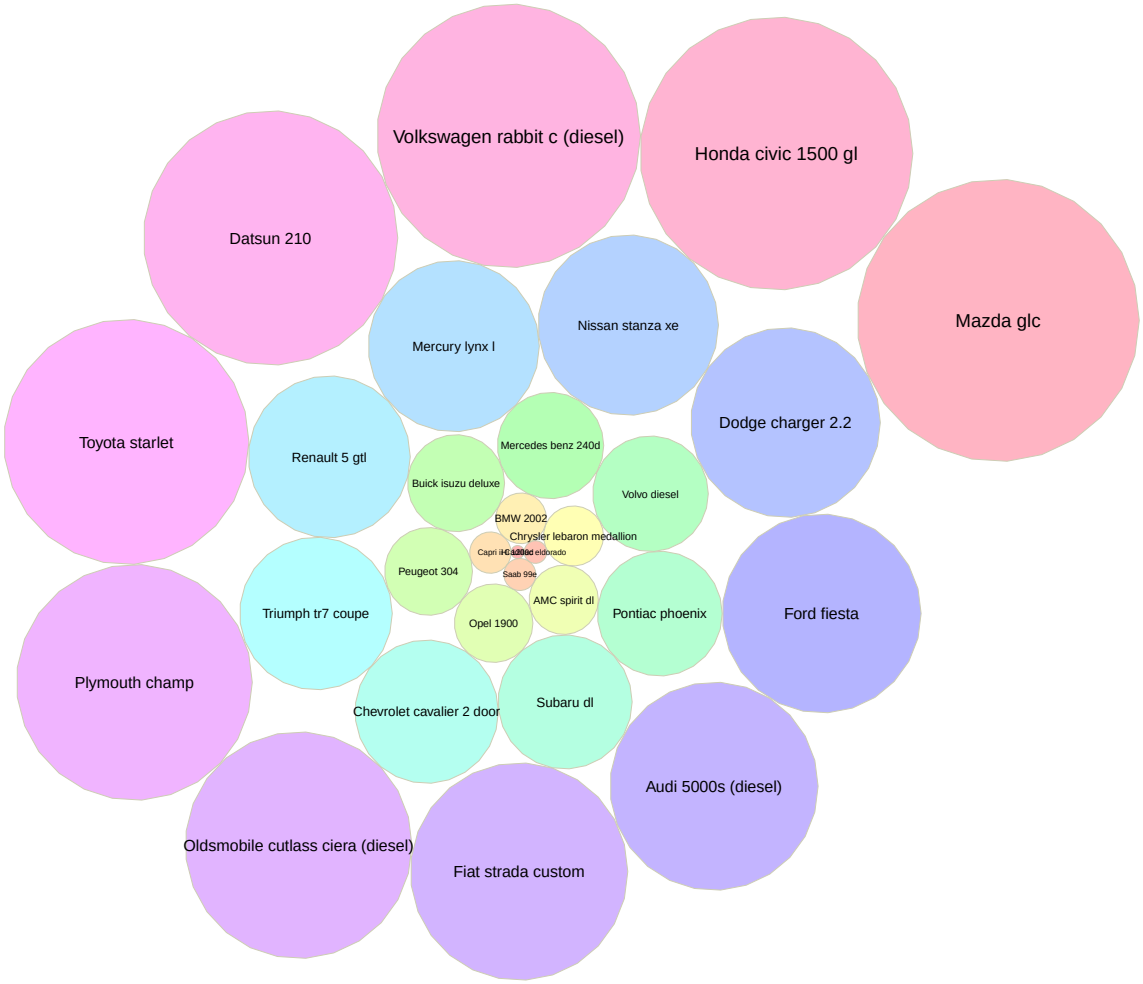
Ensuite on va garder qu'un seul véhicule par marque (celui qui consomme le moins) pour respecter l'équité car toutes les marques n'ont pas la même fréquence d'apparition.
Puis on va unir les colonnes *Marque* et *Modele* pour avoir la marque et le modèle du véhicule selectionné et ceci grâce à la fonction `unite`.

Par ailleurs on a décidé de construire un vecteur arbitraire pour régir la taille des cercles du circle plot car si on prend les *Mpg*, on ne voit pas beaucoup de différence.

Pour cela on a contruit un vecteur de même longueur que *aut_Mar_Mod_min* qui commence par 1 et de terme suivant égal à la somme de l'indice de son emplacement dans le vecteur et du terme précédent (premier terme 1, deuxieme terme 2+1 = 3, troisième terme 3+3 = 6 ,...).
C'est un peu artisanal mais ça nous permet d'avoir un meilleur rendu.

Et enfin on affiche le `Circle plot` correspondant :

```
auto_Marque_Modele %>% arrange(Mpg) %>%
  group_by(Marque,Modele) %>%
  arrange(desc(Mpg)) -> aut_Mar_Mod_min

#suppression des marques redontantes (c'est pour ça que le 'arrange(desc(Mpg))'
#est important dans le code précédent car les véhicules sont classés
#par ordre croissant de consommation).
aut_Mar_Mod_min[!duplicated(aut_Mar_Mod_min$Marque), ] -> aut_Mar_Mod_min
aut_Mar_Mod_min %>% arrange(Mpg) -> aut_Mar_Mod_min

#création du vecteur arbitraire qui va régire la taille des carrés
#le plus grand terme de ce vecteur sera attribué au véhicule qui consomme le moins.
taille <- c(1)
for(i in 1:(nrow(aut_Mar_Mod_min)-1)){
  taille[i+1] = taille[i]+i+1
}

#Unification des colonnes Marque et Modele.
aut_Mar_Mod_min %>%
  unite(Nom,Marque,Modele,sep = " ",remove = T) -> aut_Mar_Mod_min

#Rassemblement des colonnes de aut_Mar_Mod_min et de la colonnes taille.
bind_cols(aut_Mar_Mod_min,taille=taille) -> aut_Mar_Mod_min

#Circle plot correspondant
dat <- circleProgressiveLayout(aut_Mar_Mod_min$taille, sizetype='area')
aut_Mar_Mod_min <- cbind(aut_Mar_Mod_min,dat)
dat_1 <- circleLayoutVertices(dat)
ggplot()  +
  geom_polygon(data = dat_1, aes(x, y, group = id, fill=as.factor(id)),
               color = 'lightyellow3',lwd=0.02, alpha = 0.3) +
  geom_text(data = aut_Mar_Mod_min, aes(x, y, label = Nom,size = taille)) +
  scale_size_continuous(range = c(1,2.4)) +
  scale_fill_manual(values = rainbow(nrow(aut_Mar_Mod_min))) +
  theme_void() +
  theme(legend.position="none") +
  coord_equal()
```



On voit que c'est la *Mazda glc* (voiture asiatique) qui consomme le moins suivi de la *Honda civic 1500 gl* (voiture asiatique), on remarque que dans les dix premiers véhicules qui consomment le moins, seuls trois véhicules sont américains alors qu'ils sont en grande majorité dans le jeu de données.
Les véhicules asiatiques dominent clairement en matière de basse consommation.

5 Conclusion

Dans cet étude, on remarque qu'au fil du temps, les véhicules du jeu de données consomment de moins en moins d'essence, nous avons fait une analyse des données pour tenter d'expliquer cette baisse.
Nous en avons déduit plusieurs choses :

La consommation d'essence est fortement corrélée à certaines variables : la *Puissance*, la *Cylindrée*, le nombre de *Cylindres* et le *Poids*.
Toutes ces variables influent sur la consommation et on remarque qu'en fonction de l' *Origine*, il y avait de fortes fluctuations.

La variable *Origine* est primordiale pour comparer les véhicules des différents continents et nous en avons déduit que les véhicules américains ont les meilleures caractéristiques, mais malheuresement ils ont aussi la plus grande consommation d'essence en moyenne.

Les marques qui consomment le plus sont américaines tandis que celles qui dominent le marché de la basse consommation sont asiatiques.

Nous aurions aimer avoir plus de variables explicatives telles que la boite de vitesse, un indicateur de l'usure des pneus ou encore l'utilisation du chauffage ou de la climatisation afin d'affiner nos résultats.

Par ailleurs, une regression linéaire aurait été intéressante pour savoir si certaines variables pourraient être négligées.
Une classification ascendante hierarchique nous aurait permis de rassembler les véhicules en plusieurs groupes.