Tp n:2

2024-09-17

A. Importation et présentation des données

auto=read.csv2("/home/danielao/Desktop/L3iA/LanguageR/TP2/auto.csv", header=T, stringsAsFactors = T, na.strings = c("NA","na","")) #Importation de le fichier auto.csv  
  
str(auto)# Affichage

## 'data.frame': 175 obs. of 19 variables:  
## $ POSSESSION : Factor w/ 1 level "Oui": 1 1 1 1 1 1 1 1 1 1 ...  
## $ APPARTENANCE: Factor w/ 3 levels "famille","societe",..: 1 1 1 3 3 3 1 3 3 3 ...  
## $ MARQUE : Factor w/ 12 levels "Autre","BMW",..: 3 3 1 9 8 9 9 8 5 6 ...  
## $ TYPE : Factor w/ 2 levels "Neuve","Occasion": 2 2 1 1 2 1 1 2 1 2 ...  
## $ ANCIENNETE : int 5 NA 2 4 8 2 2 3 3 4 ...  
## $ PUISSANCE : Factor w/ 5 levels "10CV\_et\_plus",..: 2 5 NA 3 2 2 2 2 2 4 ...  
## $ CARBURANT : Factor w/ 4 levels "Essence","Gasoil",..: 4 4 2 1 1 2 2 4 1 1 ...  
## $ OPTIONS : Factor w/ 6 levels "Airbags","Autres",..: 6 2 1 3 2 1 1 1 3 3 ...  
## $ NOTE : Factor w/ 13 levels "10.00","11.00",..: 2 3 7 6 1 8 5 2 3 8 ...  
## $ ENTRETIEN : int 0 0 0 1 1 0 0 1 0 5 ...  
## $ KILOMETRAGE : int 40 1300 400 1500 500 500 100 500 1000 3000 ...  
## $ DEPENSE : int 0 500 100 2000 400 600 0 1000 1200 2000 ...  
## $ CRITERE1 : Factor w/ 8 levels "confort","consommation",..: 2 3 3 1 5 1 5 1 1 5 ...  
## $ CRITERE2 : Factor w/ 8 levels "confort","consommation",..: 7 2 5 5 8 7 1 7 5 8 ...  
## $ CRITERE3 : Factor w/ 9 levels "confort","consommation",..: 3 7 2 8 2 2 9 5 4 1 ...  
## $ CRITERE4 : Factor w/ 9 levels "confort","consommation",..: 9 9 1 4 7 5 3 4 7 8 ...  
## $ CRITERE5 : Factor w/ 9 levels "confort","consommation",..: 1 5 9 9 4 3 2 2 8 6 ...  
## $ SEXE : Factor w/ 2 levels "Femme","Homme": 1 2 2 2 1 1 2 2 1 2 ...  
## $ CSP : Factor w/ 9 levels "Agriculteur",..: 7 7 7 3 7 6 7 8 6 3 ...

#Transformer la variable « puissance » en variable qualitative ordinale en donnant le bon ordre aux modalités.  
unique(auto$PUISSANCE)

## [1] 4-5\_CV moins\_de\_4CV <NA> 6-7\_CV 8-9\_CV   
## [6] 10CV\_et\_plus  
## Levels: 10CV\_et\_plus 4-5\_CV 6-7\_CV 8-9\_CV moins\_de\_4CV

auto$PUISSANCE = ordered(auto$PUISSANCE,   
 levels = c("moins\_de\_4CV", "4-5\_CV", "6-7\_CV", "8-9\_CV", "10CV\_et\_pl7us"))  
  
#Verification  
summary(auto$PUISSANCE)

## moins\_de\_4CV 4-5\_CV 6-7\_CV 8-9\_CV 10CV\_et\_pl7us   
## 12 70 47 24 0   
## NA's   
## 22

# a) Nombre d'individus (lignes)  
nombre\_individus = nrow(auto)  
nombre\_individus

## [1] 175

# b) Nombre de variables (colonnes)  
nombre\_variables = ncol(auto)  
nombre\_variables

## [1] 19

# c) Résumé statistique de toutes les variables  
summary(auto)

## POSSESSION APPARTENANCE MARQUE TYPE ANCIENNETE   
## Oui:175 famille: 30 Renault :36 Neuve :94 Min. : 0.000   
## societe: 7 Citroen :21 Occasion:81 1st Qu.: 2.000   
## vous :138 Peugeot :21 Median : 4.000   
## Autre :20 Mean : 4.159   
## Volkswagen:18 3rd Qu.: 5.000   
## Ford :14 Max. :15.000   
## (Other) :45 NA's :37   
## PUISSANCE CARBURANT OPTIONS NOTE   
## moins\_de\_4CV :12 Essence :59 Airbags :42 13.00 :33   
## 4-5\_CV :70 Gasoil :72 Autres :34 12.00 :29   
## 6-7\_CV :47 GPL :23 Climatisation :54 15.00 :23   
## 8-9\_CV :24 Sans plomb:21 Jantes alu :15 11.00 :21   
## 10CV\_et\_pl7us: 0 Lecteur CD :15 14.00 :20   
## NA's :22 Ordinateur de bord: 2 17.00 :15   
## NA's :13 (Other):34   
## ENTRETIEN KILOMETRAGE DEPENSE CRITERE1   
## Min. : 0.000 Min. : 40 Min. : 0 esthetique :62   
## 1st Qu.: 0.000 1st Qu.:1000 1st Qu.: 500 confort :48   
## Median : 1.000 Median :1500 Median :1000 cout\_entretien:47   
## Mean : 2.103 Mean :1671 Mean :1132 pollution : 6   
## 3rd Qu.: 3.000 3rd Qu.:2000 3rd Qu.:1500 vitesse : 5   
## Max. :15.000 Max. :6000 Max. :5000 consommation : 3   
## (Other) : 4   
## CRITERE2 CRITERE3 CRITERE4   
## consommation :32 esthetique :32 consommation :38   
## securite :32 consommation :30 vitesse :24   
## cout\_entretien:28 cout\_entretien:30 confort :23   
## esthetique :26 confort :24 cout\_entretien:21   
## confort :25 vitesse :23 pollution :21   
## vitesse :22 pollution :14 esthetique :20   
## (Other) :10 (Other) :22 (Other) :28   
## CRITERE5 SEXE CSP   
## pollution :31 Femme:84 Cadre :32   
## confort :30 Homme:91 Employe :26   
## consommation :29 Etudiant:23   
## esthetique :21 Inactif :17   
## vitesse :19 Ouvrier :15   
## cout\_entretien:13 (Other) :39   
## (Other) :32 NA's :23

Partie 2 B. Pour la variable APPARTENANCE, écrire le code R pour obtenir :

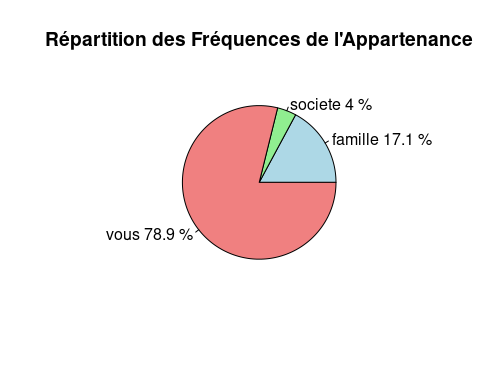
# 1. Le tableau des effectifs pour la variable APPARTENANCE  
tableau\_effectifs\_appartenance = table(auto$APPARTENANCE)  
tableau\_effectifs\_appartenance

##   
## famille societe vous   
## 30 7 138

# 2. Le tableau des fréquences pour la variable APPARTENANCE  
tableau\_frequences\_appartenance = prop.table(table(auto$APPARTENANCE))  
tableau\_frequences\_appartenance

##   
## famille societe vous   
## 0.1714286 0.0400000 0.7885714

# 3. Graphique en secteurs circulaires pour la variable APPARTENANCE  
  
# Créer le graphique en secteurs circulaires  
pie(tableau\_frequences\_appartenance,   
 col = c("lightblue", "lightgreen", "lightcoral"),   
 main = "Répartition des Fréquences de l'Appartenance",   
 labels = paste(names(tableau\_frequences\_appartenance), round(tableau\_frequences\_appartenance\*100, 1), "%"))



C. Pour la variable MARQUE, écrire le code R pour obtenir :

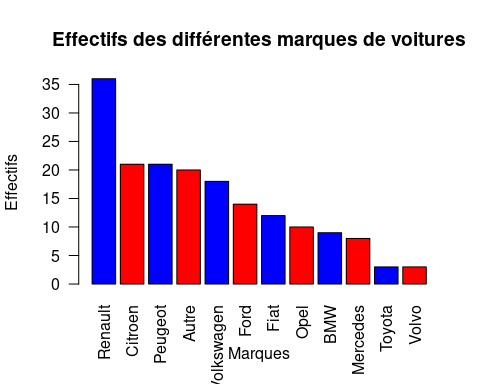
# 4. Le tableau des effectifs pour la variable MARQUE  
tableau\_effectifs\_MARQUE = sort(table(auto$MARQUE), decreasing = TRUE)  
tableau\_effectifs\_MARQUE

##   
## Renault Citroen Peugeot Autre Volkswagen Ford Fiat   
## 36 21 21 20 18 14 12   
## Opel BMW Mercedes Toyota Volvo   
## 10 9 8 3 3

# 5. Le tableau des fréquences pour la variable MARQUE  
tableau\_frequences\_MARQUE = prop.table(table(auto$MARQUE))  
tableau\_frequences\_MARQUE

##   
## Autre BMW Citroen Fiat Ford Mercedes Opel   
## 0.11428571 0.05142857 0.12000000 0.06857143 0.08000000 0.04571429 0.05714286   
## Peugeot Renault Toyota Volkswagen Volvo   
## 0.12000000 0.20571429 0.01714286 0.10285714 0.01714286

# 6. Graphique en tuyaux d'orgue pour représenter les effectifs de la variable MARQUE  
  
# Créer le graphique en tuyaux d'orgue (barplot)  
barplot(tableau\_effectifs\_MARQUE,   
 col = c("blue", "red"),  
 main = "Effectifs des différentes marques de voitures",   
 xlab = "Marques",   
 ylab = "Effectifs",   
 las = 2)

 D. Pour la variable ANCIENNETE, écrire le code R pour obtenir :

# 7. Moyenne, écart type, médiane, et écart interquartile pour la variable ANCIENNETE  
  
# Vérifier si ANCIENNETE est bien numérique  
summary(auto$ANCIENNETE)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.000 2.000 4.000 4.159 5.000 15.000 37

# Calcul de la moyenne  
moyenne\_anciennete = mean(auto$ANCIENNETE, na.rm = TRUE)  
moyenne\_anciennete

## [1] 4.15942

# Calcul de l'écart type  
ecart\_type\_anciennete = sd(auto$ANCIENNETE, na.rm = TRUE)  
ecart\_type\_anciennete

## [1] 2.885267

# Calcul de la médiane  
mediane\_anciennete = median(auto$ANCIENNETE, na.rm = TRUE)  
mediane\_anciennete

## [1] 4

# Calcul de l'écart interquartile (IQR)  
ecart\_interquartile\_anciennete = IQR(auto$ANCIENNETE, na.rm = TRUE)  
ecart\_interquartile\_anciennete

## [1] 3

# 8. La liste des différentes valeurs pour la variable ANCIENNETE  
  
valeurs\_anciennete = unique(auto$ANCIENNETE)  
valeurs\_anciennete

## [1] 5 NA 2 4 8 3 1 6 0 7 10 9 15 12 11

#Type de la variable ANCIENNETE  
type\_anciennete = class(auto$ANCIENNETE)  
type\_anciennete

## [1] "integer"

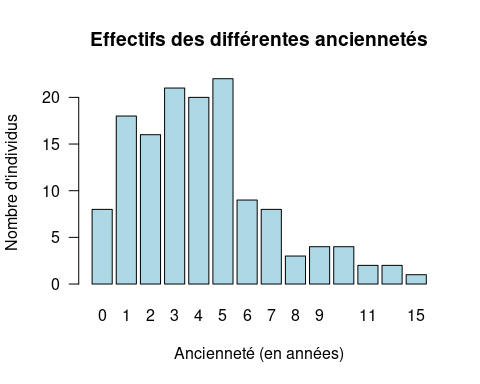
# 9. Le tableau des effectifs pour la variable ANCIENNETE  
tableau\_effectifs\_anciennete = table(auto$ANCIENNETE)  
tableau\_effectifs\_anciennete

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 15   
## 8 18 16 21 20 22 9 8 3 4 4 2 2 1

# 10. Le tableau des fréquences pour la variable ANCIENNETE  
  
tableau\_frequences\_anciennete = prop.table(table(auto$ANCIENNETE))  
tableau\_frequences\_anciennete

##   
## 0 1 2 3 4 5   
## 0.057971014 0.130434783 0.115942029 0.152173913 0.144927536 0.159420290   
## 6 7 8 9 10 11   
## 0.065217391 0.057971014 0.021739130 0.028985507 0.028985507 0.014492754   
## 12 15   
## 0.014492754 0.007246377

# 11. Diagramme en bâtons pour représenter les effectifs de la variable ANCIENNETE  
  
# Créer le diagramme en bâtons pour les effectifs de ANCIENNETE  
barplot(tableau\_effectifs\_anciennete,   
 col = "lightblue",   
 main = "Effectifs des différentes anciennetés", # Titre du graphique  
 xlab = "Ancienneté (en années)", # Titre de l'axe des x  
 ylab = "Nombre d'individus", # Titre de l'axe des y  
 las = 1) # Orientation des étiquettes sur l'axe des x



E. Pour la variable KILOMETRAGE, écrire le code R pour obtenir :

# 12. Moyenne, écart type, médiane, et écart interquartile pour la variable KILOMETRAGE  
  
# Calcul de la moyenne  
moyenne\_kilometrage = mean(auto$KILOMETRAGE, na.rm = TRUE)  
moyenne\_kilometrage

## [1] 1670.514

# Calcul de l'écart type  
ecart\_type\_kilometrage = sd(auto$KILOMETRAGE, na.rm = TRUE)  
ecart\_type\_kilometrage

## [1] 1008.266

# Calcul de la médiane  
mediane\_kilometrage = median(auto$KILOMETRAGE, na.rm = TRUE)  
mediane\_kilometrage

## [1] 1500

# Calcul de l'écart interquartile (IQR)  
ecart\_interquartile\_kilometrage = IQR(auto$KILOMETRAGE, na.rm = TRUE)  
ecart\_interquartile\_kilometrage

## [1] 1000

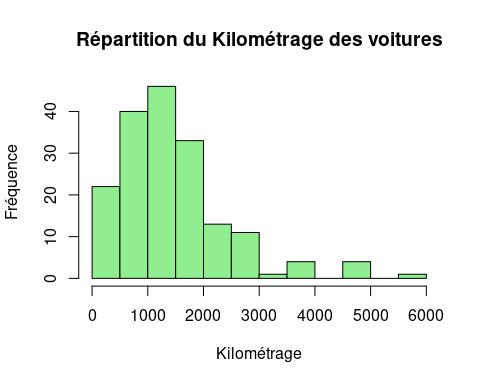
# 13. La liste des différentes valeurs pour la variable KILOMETRAGE  
valeurs\_kilometrage = unique(auto$KILOMETRAGE)  
valeurs\_kilometrage

## [1] 40 1300 400 1500 500 100 1000 3000 800 2000 5000 700 1200 4000 300  
## [16] 2500 6000 1750 2200 3500 750 200

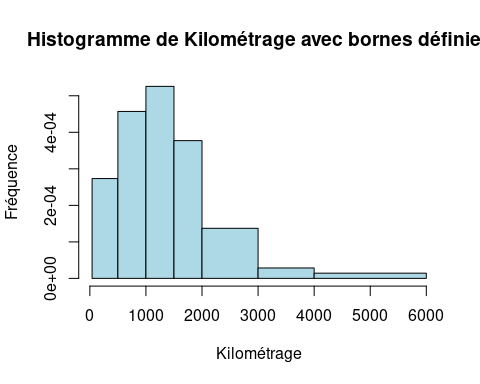
# Type de la variable KILOMETRAGE  
type\_kilometrage = class(auto$KILOMETRAGE)  
type\_kilometrage

## [1] "integer"

# 14. Histogramme pour la variable KILOMETRAGE avec titres et couleur personnalisés  
hist(auto$KILOMETRAGE,   
 col = "lightgreen",   
 main = "Répartition du Kilométrage des voitures",   
 xlab = "Kilométrage",   
 ylab = "Fréquence",   
 breaks = 20) # Ajuster le nombre de barres selon les besoins



# 15. Histogramme avec des bornes spécifiques pour KILOMETRAGE  
hist(auto$KILOMETRAGE,   
 breaks = c(40, 500, 1000, 1500, 2000, 3000, 4000, 6000),   
 col = "lightblue",   
 main = "Histogramme de Kilométrage avec bornes définies",   
 xlab = "Kilométrage",   
 ylab = "Fréquence")



# 16. Discrétisation de la variable KILOMETRAGE selon des bornes définies  
bornes\_kilometrage = c(0, 500, 1000, 1500, 2000, 3000, 4000, 6000)  
  
# Utiliser cut() pour discrétiser la variable  
auto$KILOMETRAGE\_DISCRET = cut(auto$KILOMETRAGE,   
 breaks = bornes\_kilometrage,   
 include.lowest = TRUE)  
head(auto$KILOMETRAGE\_DISCRET)

## [1] [0,500] (1e+03,1.5e+03] [0,500] (1e+03,1.5e+03]  
## [5] [0,500] [0,500]   
## 7 Levels: [0,500] (500,1e+03] (1e+03,1.5e+03] ... (4e+03,6e+03]

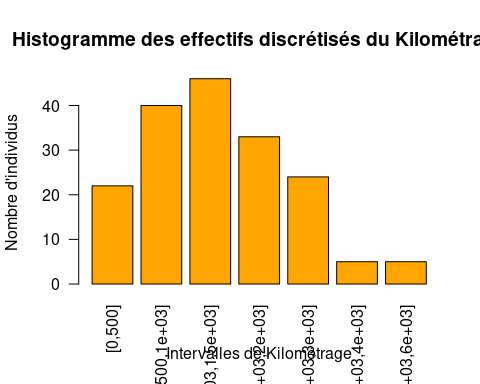
# 17. Tableau des effectifs pour la discrétisation de KILOMETRAGE  
tableau\_effectifs\_kilometrage\_discret = table(auto$KILOMETRAGE\_DISCRET)  
tableau\_effectifs\_kilometrage\_discret

##   
## [0,500] (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03] (2e+03,3e+03]   
## 22 40 46 33 24   
## (3e+03,4e+03] (4e+03,6e+03]   
## 5 5

# 18. Tableau des fréquences pour la discrétisation de KILOMETRAGE  
tableau\_frequences\_kilometrage\_discret = prop.table(table(auto$KILOMETRAGE\_DISCRET))  
tableau\_frequences\_kilometrage\_discret

##   
## [0,500] (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03] (2e+03,3e+03]   
## 0.12571429 0.22857143 0.26285714 0.18857143 0.13714286   
## (3e+03,4e+03] (4e+03,6e+03]   
## 0.02857143 0.02857143

# 19. Histogramme pour la discrétisation de KILOMETRAGE  
barplot(tableau\_effectifs\_kilometrage\_discret,   
 col = "orange",   
 main = "Histogramme des effectifs discrétisés du Kilométrage",   
 xlab = "Intervalles de Kilométrage",   
 ylab = "Nombre d'individus",  
 las=2)



#———————————————————————————————————#

#—————————————-TP3————————————————————–#

#———————————————————————————————————#

A. Pour le croisement MARQUE – CARBURANT, écrire le code R pour :

# 1. Tableau des effectifs pour le croisement MARQUE – CARBURANT  
tableau\_effectifs\_croisement = table(auto$MARQUE, auto$CARBURANT)  
tableau\_effectifs\_croisement

##   
## Essence Gasoil GPL Sans plomb  
## Autre 12 6 2 0  
## BMW 6 2 1 0  
## Citroen 6 11 0 4  
## Fiat 2 7 1 2  
## Ford 3 8 0 3  
## Mercedes 4 4 0 0  
## Opel 4 6 0 0  
## Peugeot 4 8 3 6  
## Renault 11 13 9 3  
## Toyota 2 0 0 1  
## Volkswagen 5 4 7 2  
## Volvo 0 3 0 0

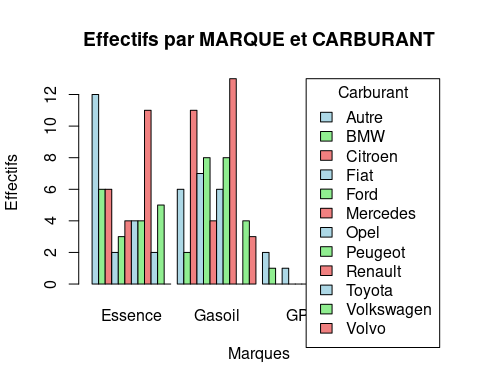
# 2. Tableau des fréquences conditionnelles en ligne (MARQUE)  
frequences\_ligne = prop.table(tableau\_effectifs\_croisement, margin = 1)  
frequences\_ligne

##   
## Essence Gasoil GPL Sans plomb  
## Autre 0.60000000 0.30000000 0.10000000 0.00000000  
## BMW 0.66666667 0.22222222 0.11111111 0.00000000  
## Citroen 0.28571429 0.52380952 0.00000000 0.19047619  
## Fiat 0.16666667 0.58333333 0.08333333 0.16666667  
## Ford 0.21428571 0.57142857 0.00000000 0.21428571  
## Mercedes 0.50000000 0.50000000 0.00000000 0.00000000  
## Opel 0.40000000 0.60000000 0.00000000 0.00000000  
## Peugeot 0.19047619 0.38095238 0.14285714 0.28571429  
## Renault 0.30555556 0.36111111 0.25000000 0.08333333  
## Toyota 0.66666667 0.00000000 0.00000000 0.33333333  
## Volkswagen 0.27777778 0.22222222 0.38888889 0.11111111  
## Volvo 0.00000000 1.00000000 0.00000000 0.00000000

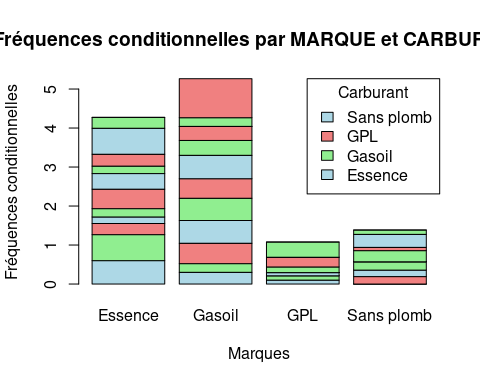
# 3. Tableau des fréquences conditionnelles en colonne (CARBURANT)  
frequences\_colonne = prop.table(tableau\_effectifs\_croisement, margin = 2)  
frequences\_colonne

##   
## Essence Gasoil GPL Sans plomb  
## Autre 0.20338983 0.08333333 0.08695652 0.00000000  
## BMW 0.10169492 0.02777778 0.04347826 0.00000000  
## Citroen 0.10169492 0.15277778 0.00000000 0.19047619  
## Fiat 0.03389831 0.09722222 0.04347826 0.09523810  
## Ford 0.05084746 0.11111111 0.00000000 0.14285714  
## Mercedes 0.06779661 0.05555556 0.00000000 0.00000000  
## Opel 0.06779661 0.08333333 0.00000000 0.00000000  
## Peugeot 0.06779661 0.11111111 0.13043478 0.28571429  
## Renault 0.18644068 0.18055556 0.39130435 0.14285714  
## Toyota 0.03389831 0.00000000 0.00000000 0.04761905  
## Volkswagen 0.08474576 0.05555556 0.30434783 0.09523810  
## Volvo 0.00000000 0.04166667 0.00000000 0.00000000

# Le tableau en ligne est plus utile si vous souhaitez observer comment chaque marque se répartit selon le carburant.  
  
  
# 4. Histogramme non empilé des effectifs pour MARQUE – CARBURANT  
barplot(tableau\_effectifs\_croisement,   
 beside = TRUE, # Non empilé  
 col = c("lightblue", "lightgreen", "lightcoral"),   
 main = "Effectifs par MARQUE et CARBURANT",   
 xlab = "Marques",   
 ylab = "Effectifs",   
 legend = rownames(tableau\_effectifs\_croisement),   
 args.legend = c(title = "Carburant", x = "topright"))



# 5. Histogramme empilé des fréquences conditionnelles jugées les plus parlantes (en ligne, par exemple)  
barplot(frequences\_ligne,   
 col = c("lightblue", "lightgreen", "lightcoral"),   
 main = "Fréquences conditionnelles par MARQUE et CARBURANT",   
 xlab = "Marques",   
 ylab = "Fréquences conditionnelles",   
 legend = colnames(frequences\_ligne),   
 args.legend = c(title = "Carburant", x = "topright"))



B. Pour le croisement MARQUE – SEXE, écrire le code R pour :

# 1. Tableau des effectifs pour le croisement MARQUE – SEXE  
tableau\_effectifs\_MARQUE\_SEXE = table(auto$MARQUE, auto$SEXE)  
tableau\_effectifs\_MARQUE\_SEXE

##   
## Femme Homme  
## Autre 8 12  
## BMW 2 7  
## Citroen 10 11  
## Fiat 7 5  
## Ford 6 8  
## Mercedes 2 6  
## Opel 4 6  
## Peugeot 13 8  
## Renault 20 16  
## Toyota 1 2  
## Volkswagen 10 8  
## Volvo 1 2

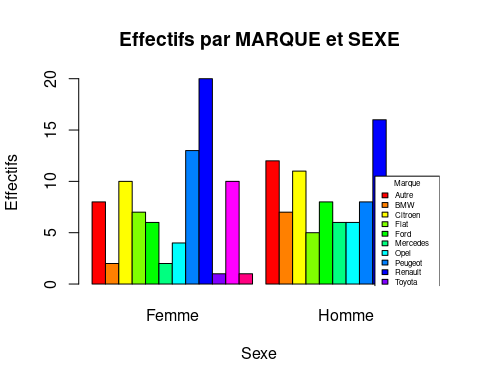
# 2. Fréquences conditionnelles en ligne (par MARQUE)  
frequences\_ligne\_MARQUE\_SEXE = prop.table(tableau\_effectifs\_MARQUE\_SEXE, margin = 1)  
frequences\_ligne\_MARQUE\_SEXE

##   
## Femme Homme  
## Autre 0.4000000 0.6000000  
## BMW 0.2222222 0.7777778  
## Citroen 0.4761905 0.5238095  
## Fiat 0.5833333 0.4166667  
## Ford 0.4285714 0.5714286  
## Mercedes 0.2500000 0.7500000  
## Opel 0.4000000 0.6000000  
## Peugeot 0.6190476 0.3809524  
## Renault 0.5555556 0.4444444  
## Toyota 0.3333333 0.6666667  
## Volkswagen 0.5555556 0.4444444  
## Volvo 0.3333333 0.6666667

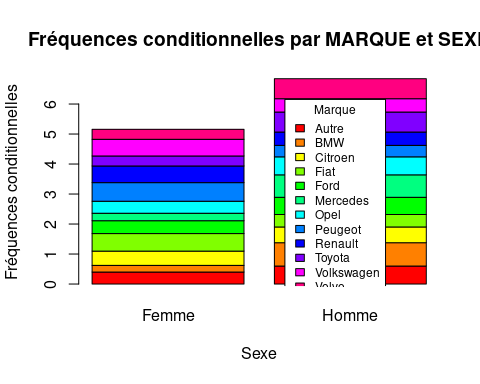
# 3. Fréquences conditionnelles en colonne (par SEXE)  
frequences\_colonne\_MARQUE\_SEXE = prop.table(tableau\_effectifs\_MARQUE\_SEXE, margin = 2)  
frequences\_colonne\_MARQUE\_SEXE

##   
## Femme Homme  
## Autre 0.09523810 0.13186813  
## BMW 0.02380952 0.07692308  
## Citroen 0.11904762 0.12087912  
## Fiat 0.08333333 0.05494505  
## Ford 0.07142857 0.08791209  
## Mercedes 0.02380952 0.06593407  
## Opel 0.04761905 0.06593407  
## Peugeot 0.15476190 0.08791209  
## Renault 0.23809524 0.17582418  
## Toyota 0.01190476 0.02197802  
## Volkswagen 0.11904762 0.08791209  
## Volvo 0.01190476 0.02197802

# Define a color vector for each MARQUE  
couleurs\_marques = rainbow(length(rownames(tableau\_effectifs\_MARQUE\_SEXE)))  
  
# 4. Histogramme non empilé avec la légende  
bp = barplot(tableau\_effectifs\_MARQUE\_SEXE,   
 beside = TRUE,   
 col = couleurs\_marques, # Assign unique colors to each MARQUE  
 main = "Effectifs par MARQUE et SEXE",   
 xlab = "Sexe",   
 ylab = "Effectifs")  
  
# Ajouter la légende manuellement avec la fonction legend()  
legend("topright",   
 legend = rownames(tableau\_effectifs\_MARQUE\_SEXE),   
 fill = couleurs\_marques,   
 title = "Marque",   
 inset = c(0,0.47),   
 cex = 0.5)



# 5. Histogramme empilé des fréquences conditionnelles pour MARQUE – SEXE  
bp <- barplot(frequences\_ligne\_MARQUE\_SEXE,   
 col = couleurs\_marques, # Assign unique colors to each MARQUE  
 main = "Fréquences conditionnelles par MARQUE et SEXE",   
 xlab = "Sexe",   
 ylab = "Fréquences conditionnelles")  
  
# Ajouter la légende manuellement avec une taille réduite  
legend("topright",   
 legend = rownames(frequences\_ligne\_MARQUE\_SEXE),   
 fill = couleurs\_marques,   
 title = "Marque",   
 inset = c(0.15, 0.10),   
 cex = 0.75) # Adjust the text size and overall legend block size



C. Pour le croisement APPARTENANCE – KILOMETRAGE

# 7. Tableau des moyennes, écart types et quartiles conditionnels selon APPARTENANCE  
  
# Moyenne pour chaque groupe d'APPARTENANCE  
moyennes\_appartenance = tapply(auto$KILOMETRAGE, auto$APPARTENANCE, mean, na.rm = TRUE)  
moyennes\_appartenance

## famille societe vous   
## 1028.000 4000.000 1692.029

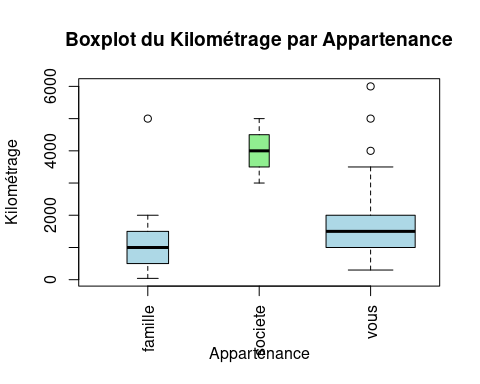
# Ecart type pour chaque groupe d'APPARTENANCE  
ecart\_types\_appartenance = tapply(auto$KILOMETRAGE, auto$APPARTENANCE, sd, na.rm = TRUE)  
ecart\_types\_appartenance

## famille societe vous   
## 913.4640 816.4966 846.8759

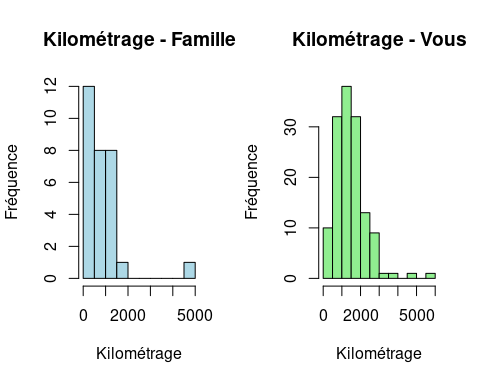
# Quartiles pour chaque groupe d'APPARTENANCE  
quartiles\_appartenance = tapply(auto$KILOMETRAGE, auto$APPARTENANCE, quantile, na.rm = TRUE)  
quartiles\_appartenance

## $famille  
## 0% 25% 50% 75% 100%   
## 40 500 1000 1500 5000   
##   
## $societe  
## 0% 25% 50% 75% 100%   
## 3000 3500 4000 4500 5000   
##   
## $vous  
## 0% 25% 50% 75% 100%   
## 300 1000 1500 2000 6000

# 8. Boxplot pour KILOMETRAGE par APPARTENANCE avec personnalisation  
  
boxplot(auto$KILOMETRAGE ~ auto$APPARTENANCE,   
 col = c("lightblue", "lightgreen"), # Couleurs personnalisées  
 main = "Boxplot du Kilométrage par Appartenance",   
 xlab = "Appartenance",   
 ylab = "Kilométrage",   
 varwidth = TRUE, # Largeur des boxplots proportionnelle à la taille des groupes  
 las = 3) # Orientation des étiquettes de l'axe des x



# 9. Histogrammes pour chaque type d'appartenance, côte à côte  
  
# Ajuster la disposition des fenêtres graphiques pour afficher 2 graphiques côte à côte  
par(mfrow = c(1, 2)) # 1 ligne, 2 colonnes  
  
# Histogramme pour la première catégorie d'APPARTENANCE  
hist(auto$KILOMETRAGE[auto$APPARTENANCE == "famille"],   
 col = "lightblue",   
 main = "Kilométrage - Famille",   
 xlab = "Kilométrage",   
 ylab = "Fréquence",   
 breaks = 10)  
  
# Histogramme pour la deuxième catégorie d'APPARTENANCE  
hist(auto$KILOMETRAGE[auto$APPARTENANCE == "vous"],   
 col = "lightgreen",   
 main = "Kilométrage - Vous",   
 xlab = "Kilométrage",   
 ylab = "Fréquence",   
 breaks = 10)



# Reset graphic parameters  
par(mfrow = c(1, 1))

D. Pour le croisement MARQUE – DEPENSE

# 10. Moyenne, écart type et quartiles conditionnels selon MARQUE  
  
# Moyenne pour chaque groupe de MARQUE  
moyennes\_depense = tapply(auto$DEPENSE, auto$MARQUE, mean, na.rm = TRUE)  
moyennes\_depense

## Autre BMW Citroen Fiat Ford Mercedes Opel   
## 1300.0000 2138.8889 723.8095 1108.3333 857.1429 1906.2500 965.0000   
## Peugeot Renault Toyota Volkswagen Volvo   
## 1035.7143 1026.3889 1350.0000 1250.0000 733.3333

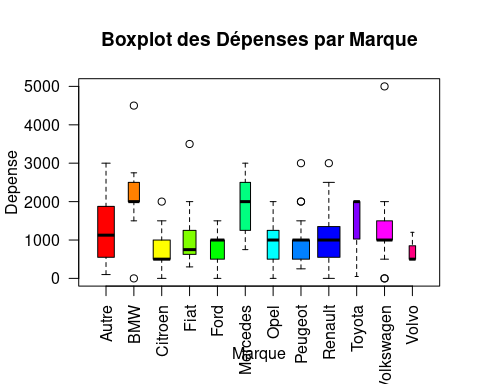
# Ecart type pour chaque groupe de MARQUE  
ecart\_types\_depense = tapply(auto$DEPENSE, auto$MARQUE, sd, na.rm = TRUE)  
ecart\_types\_depense

## Autre BMW Citroen Fiat Ford Mercedes Opel   
## 797.5291 1179.9835 510.7884 884.1620 420.0994 778.4772 558.7933   
## Peugeot Renault Toyota Volkswagen Volvo   
## 646.1700 684.8864 1125.8330 1088.0365 404.1452

# Quartiles pour chaque groupe de MARQUE  
quartiles\_depense = tapply(auto$DEPENSE, auto$MARQUE, quantile, na.rm = TRUE)  
quartiles\_depense

## $Autre  
## 0% 25% 50% 75% 100%   
## 100.0 575.0 1125.0 1812.5 3000.0   
##   
## $BMW  
## 0% 25% 50% 75% 100%   
## 0 2000 2000 2500 4500   
##   
## $Citroen  
## 0% 25% 50% 75% 100%   
## 0 500 500 1000 2000   
##   
## $Fiat  
## 0% 25% 50% 75% 100%   
## 300.0 687.5 750.0 1125.0 3500.0   
##   
## $Ford  
## 0% 25% 50% 75% 100%   
## 0 500 1000 1000 1500   
##   
## $Mercedes  
## 0% 25% 50% 75% 100%   
## 750 1375 2000 2500 3000   
##   
## $Opel  
## 0% 25% 50% 75% 100%   
## 0.0 600.0 1000.0 1187.5 2000.0   
##   
## $Peugeot  
## 0% 25% 50% 75% 100%   
## 250 500 1000 1000 3000   
##   
## $Renault  
## 0% 25% 50% 75% 100%   
## 0 575 1000 1275 3000   
##   
## $Toyota  
## 0% 25% 50% 75% 100%   
## 50 1025 2000 2000 2000   
##   
## $Volkswagen  
## 0% 25% 50% 75% 100%   
## 0 1000 1000 1500 5000   
##   
## $Volvo  
## 0% 25% 50% 75% 100%   
## 500 500 500 850 1200

# 11. Boxplot pour DEPENSE par MARQUE avec personnalisation  
  
boxplot(auto$DEPENSE ~ auto$MARQUE,   
 col = rainbow(length(unique(auto$MARQUE))), # Couleurs pour chaque MARQUE  
 main = "Boxplot des Dépenses par Marque",   
 xlab = "Marque",   
 ylab = "Depense",   
 varwidth = TRUE, # Largeur des boxplots proportionnelle à la taille des groupes  
 las = 2) # Orientation verticale des étiquettes de l'axe des x



E. Pour le croisement ANCIENNETE – DEPENSE

# 12. Moyenne, écart type et quartiles conditionnels selon ANCIENNETE  
  
# Moyenne pour chaque groupe d'ANCIENNETE  
moyennes\_depense\_anciennete = tapply(auto$DEPENSE, auto$ANCIENNETE, mean, na.rm = TRUE)  
moyennes\_depense\_anciennete

## 0 1 2 3 4 5 6 7   
## 1500.0000 1722.2222 996.8750 1335.7143 1150.0000 870.4545 888.8889 768.7500   
## 8 9 10 11 12 15   
## 633.3333 512.5000 762.5000 500.0000 1100.0000 500.0000

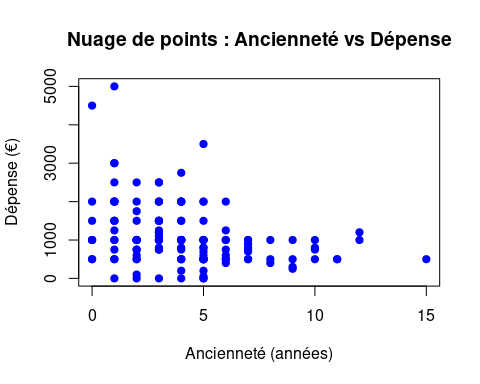
# Ecart type pour chaque groupe d'ANCIENNETE  
ecart\_types\_depense\_anciennete = tapply(auto$DEPENSE, auto$ANCIENNETE, sd, na.rm = TRUE)  
ecart\_types\_depense\_anciennete

## 0 1 2 3 4 5 6 7   
## 1309.3073 1169.1151 666.2004 610.3570 711.0037 801.8816 504.2183 198.0936   
## 8 9 10 11 12 15   
## 321.4550 342.4787 205.6494 0.0000 141.4214 NA

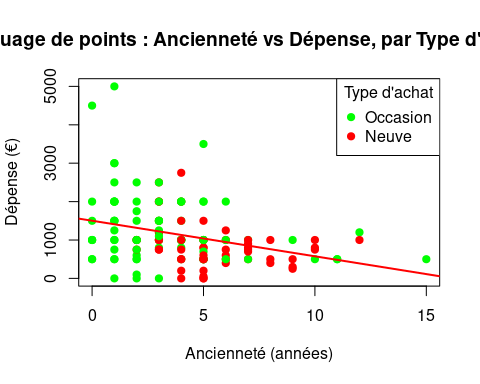
# Quartiles pour chaque groupe d'ANCIENNETE  
quartiles\_depense\_anciennete = tapply(auto$DEPENSE, auto$ANCIENNETE, quantile, na.rm = TRUE)  
quartiles\_depense\_anciennete

## $`0`  
## 0% 25% 50% 75% 100%   
## 500 875 1000 1625 4500   
##   
## $`1`  
## 0% 25% 50% 75% 100%   
## 0 1000 1500 2000 5000   
##   
## $`2`  
## 0% 25% 50% 75% 100%   
## 0 575 1000 1125 2500   
##   
## $`3`  
## 0% 25% 50% 75% 100%   
## 0 1000 1200 1500 2500   
##   
## $`4`  
## 0% 25% 50% 75% 100%   
## 0.0 687.5 1000.0 1625.0 2750.0   
##   
## $`5`  
## 0% 25% 50% 75% 100%   
## 0 500 650 1000 3500   
##   
## $`6`  
## 0% 25% 50% 75% 100%   
## 400 500 750 1000 2000   
##   
## $`7`  
## 0% 25% 50% 75% 100%   
## 500 650 775 925 1000   
##   
## $`8`  
## 0% 25% 50% 75% 100%   
## 400 450 500 750 1000   
##   
## $`9`  
## 0% 25% 50% 75% 100%   
## 250.0 287.5 400.0 625.0 1000.0   
##   
## $`10`  
## 0% 25% 50% 75% 100%   
## 500.0 687.5 775.0 850.0 1000.0   
##   
## $`11`  
## 0% 25% 50% 75% 100%   
## 500 500 500 500 500   
##   
## $`12`  
## 0% 25% 50% 75% 100%   
## 1000 1050 1100 1150 1200   
##   
## $`15`  
## 0% 25% 50% 75% 100%   
## 500 500 500 500 500

# 13. Nuage de points pour ANCIENNETE et DEPENSE  
plot(auto$ANCIENNETE, auto$DEPENSE,   
 main = "Nuage de points : Ancienneté vs Dépense",   
 xlab = "Ancienneté (années)",   
 ylab = "Dépense (€)",   
 pch = 19, col = "blue")

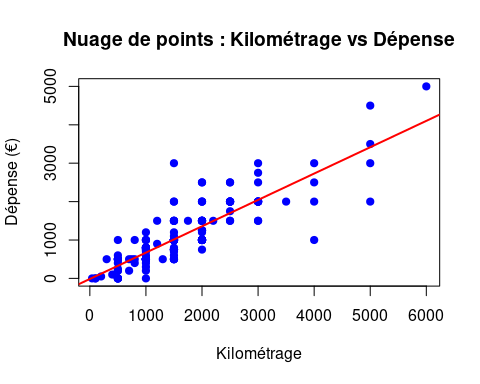


# 14. Nuage de points coloré selon le type d'achat du véhicule  
  
# Définir les couleurs pour chaque type d'achat  
couleurs\_type = c("Occasion" = "green", "Neuve" = "red")  
  
# Créer le nuage de points coloré  
plot(auto$ANCIENNETE, auto$DEPENSE,   
 col = couleurs\_type[auto$TYPE], # Coloration selon le type d'achat  
 main = "Nuage de points : Ancienneté vs Dépense, par Type d'achat",   
 xlab = "Ancienneté (années)",   
 ylab = "Dépense (€)",   
 pch = 19)  
  
# Ajouter la légende  
legend("topright", legend = names(couleurs\_type), col = couleurs\_type, pch = 19, title = "Type d'achat")  
  
  
# 15. Ajouter la courbe de régression au nuage de points  
  
# Créer un modèle linéaire pour la régression  
modele\_regression = lm(auto$DEPENSE ~ auto$ANCIENNETE)  
  
# Tracer la courbe de régression  
abline(modele\_regression, col = "red", lwd = 2) # Ligne rouge plus épaisse pour la régression



F. Pour le croisement KILOMETRAGE – DEPENSE

# 16. Nuage de points pour KILOMETRAGE et DEPENSE  
plot(auto$KILOMETRAGE, auto$DEPENSE,   
 main = "Nuage de points : Kilométrage vs Dépense",   
 xlab = "Kilométrage",   
 ylab = "Dépense (€)",   
 pch = 19, col = "blue")  
  
  
# 17. Tracer la droite des moindres carrés et donner son équation  
  
# Créer un modèle linéaire pour la régression  
modele\_kilometrage\_depense = lm(auto$DEPENSE ~ auto$KILOMETRAGE)  
  
# Tracer la droite des moindres carrés  
abline(modele\_kilometrage\_depense, col = "red", lwd = 2)



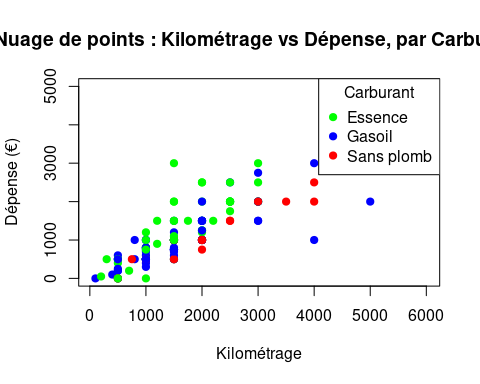
# Afficher l'équation de la droite  
coefficients(modele\_kilometrage\_depense)

## (Intercept) auto$KILOMETRAGE   
## -14.3566066 0.6862298

# 18. Coefficient de corrélation entre KILOMETRAGE et DEPENSE  
correlation = cor(auto$KILOMETRAGE, auto$DEPENSE, use = "complete.obs")  
correlation

## [1] 0.8513176

# 19. Nuage de points coloré selon le type de carburant  
  
# Définir les couleurs pour chaque type de carburant  
couleurs\_carburant = c("Essence" = "green", "Gasoil" = "blue", "Sans plomb" = "red")  
  
# Créer le nuage de points coloré  
plot(auto$KILOMETRAGE, auto$DEPENSE,   
 col = couleurs\_carburant[auto$CARBURANT], # Coloration selon le type de carburant  
 main = "Nuage de points : Kilométrage vs Dépense, par Carburant",   
 xlab = "Kilométrage",   
 ylab = "Dépense (€)",   
 pch = 19)  
  
# Ajouter la légende  
legend("topright", legend = names(couleurs\_carburant), col = couleurs\_carburant, pch = 19, title = "Carburant")



# 20. Recodage de KILOMETRAGE en classes d'intervalles  
bornes\_kilometrage = c(0, 500, 1000, 1500, 2000, 3000, 4000, 6000)  
  
# Discrétisation de KILOMETRAGE  
auto$KILOMETRAGE\_CLASSE = cut(auto$KILOMETRAGE, breaks = bornes\_kilometrage, include.lowest = TRUE)  
head(auto$KILOMETRAGE\_CLASSE)

## [1] [0,500] (1e+03,1.5e+03] [0,500] (1e+03,1.5e+03]  
## [5] [0,500] [0,500]   
## 7 Levels: [0,500] (500,1e+03] (1e+03,1.5e+03] ... (4e+03,6e+03]

# 21. Moyenne, écart type et quartiles conditionnels selon les classes de KILOMETRAGE  
  
# Moyenne pour chaque classe de KILOMETRAGE  
moyennes\_kilometrage\_classe = tapply(auto$DEPENSE, auto$KILOMETRAGE\_CLASSE, mean, na.rm = TRUE)  
moyennes\_kilometrage\_classe

## [0,500] (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03] (2e+03,3e+03]   
## 263.6364 595.0000 1033.6957 1331.8182 2020.8333   
## (3e+03,4e+03] (4e+03,6e+03]   
## 2100.0000 3600.0000

# Ecart type pour chaque classe de KILOMETRAGE  
ecart\_types\_kilometrage\_classe = tapply(auto$DEPENSE, auto$KILOMETRAGE\_CLASSE, sd, na.rm = TRUE)  
ecart\_types\_kilometrage\_classe

## [0,500] (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03] (2e+03,3e+03]   
## 279.9583 244.6347 453.0213 440.8476 396.4288   
## (3e+03,4e+03] (4e+03,6e+03]   
## 741.6198 1193.7336

# Quartiles pour chaque classe de KILOMETRAGE  
quartiles\_kilometrage\_classe = tapply(auto$DEPENSE, auto$KILOMETRAGE\_CLASSE, quantile, na.rm = TRUE)  
quartiles\_kilometrage\_classe

## $`[0,500]`  
## 0% 25% 50% 75% 100%   
## 0 0 200 500 1000   
##   
## $`(500,1e+03]`  
## 0% 25% 50% 75% 100%   
## 0 500 500 800 1200   
##   
## $`(1e+03,1.5e+03]`  
## 0% 25% 50% 75% 100%   
## 500 750 1000 1000 3000   
##   
## $`(1.5e+03,2e+03]`  
## 0% 25% 50% 75% 100%   
## 750 1000 1250 1500 2500   
##   
## $`(2e+03,3e+03]`  
## 0% 25% 50% 75% 100%   
## 1500.0 1937.5 2000.0 2000.0 3000.0   
##   
## $`(3e+03,4e+03]`  
## 0% 25% 50% 75% 100%   
## 1000 2000 2000 2500 3000   
##   
## $`(4e+03,6e+03]`  
## 0% 25% 50% 75% 100%   
## 2000 3000 3500 4500 5000

# 22. Boxplot pour DEPENSE par classes de KILOMETRAGE  
boxplot(auto$DEPENSE ~ auto$KILOMETRAGE\_CLASSE,   
 col = "lightblue",   
 main = "Boxplot des Dépenses par Classes de Kilométrage",   
 xlab = "Classes de Kilométrage",   
 ylab = "Dépense (€)",   
 varwidth = TRUE)

