# Projet analyses statistiques du data set Ames Housing avec R

Khedri maha

2BDAD2

Table des matières

[Contexte du dataset Ames Housing : 1](#_Toc197591196)

[Préparation du Jeu de Données : Nettoyage et Gestion des Valeurs Manquantes 3](#_Toc197591197)

[1- Supprimer l’espace dans le nom du colonne: 3](#_Toc197591198)

[2- Supprimer la colonne PID(ID): 3](#_Toc197591199)

[3- Gérer les valeurs manquantes : 3](#_Toc197591200)

[4- Gérer les doublons: 5](#_Toc197591201)

[Paramètres statistiques usuels 5](#_Toc197591202)

[1- Table de statistiques descriptives: 5](#_Toc197591203)

[Analyses univariées : 6](#_Toc197591204)

[1- Catégorielles : 6](#_Toc197591205)

[2- Numériques : 7](#_Toc197591206)

[a- Variable continue: 7](#_Toc197591207)

[b- Variable discrete: 9](#_Toc197591208)

[Analyses bivariées 11](#_Toc197591209)

[1- Deux variables quantitatives: 11](#_Toc197591210)

[2- Deux variables qualitatives: 16](#_Toc197591211)

# 

# Contexte du dataset Ames Housing :

Le dataset "Ames Housing" a été créé par Dean De Cock dans le but de proposer une alternative plus complexe et réaliste au célèbre dataset "Boston Housing" utilisé en apprentissage automatique. Il contient des informations détaillées sur les ventes de maisons à Ames, Iowa (États-Unis), sur une période couvrant plusieurs années.

Ce dataset est idéal pour explorer :

- la préparation des données (nettoyage, gestion des valeurs manquantes, encodage),

- l’analyse exploratoire (EDA).

- la modélisation prédictive (régression linéaire, arbres de décision, etc.).

- la sélection de variables importantes qui influencent le prix d'une maison.

df <- read.csv("./AmesHousing.csv")

head(df)

|  | **Order**  <int> | **PID**  <int> | **MS.SubClass**  <int> | **MS.Zoning**  <chr> | **Lot.Frontage**  <int> | **Lot.Area**  <int> |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 526301100 | 20 | RL | 141 | 31770 |  |
| 2 | 2 | 526350040 | 20 | RH | 80 | 11622 |  |
| 3 | 3 | 526351010 | 20 | RL | 81 | 14267 |  |
| 4 | 4 | 526353030 | 20 | RL | 93 | 11160 |  |
| 5 | 5 | 527105010 | 60 | RL | 74 | 13830 |  |
| 6 | 6 | 527105030 | 60 | RL | 78 | 9978 |  |
|  |  |  |  |  |  |  |  |

6 rows | 1-7 of 82 columns

* La commande head() permet d’afficher un aperçu du dataframe

dim(df)

[1] 2930 82

#### Le dataframe Ames Housing comprend 2930 observations et 82 variables décrivant les caractéristiques des maisons à Ames, Iowa , avec comme objectif principal la prédiction du prix de vente (SalePrice).

str(df)

'data.frame': 2930 obs. of 82 variables:

$ Order : int 1 2 3 4 5 6 7 8 9 10 ...

$ PID : int 526301100 526350040 526351010 526353030 527105010 527105030 527127150 527145080 527146030 527162130 ...

$ MS.SubClass : int 20 20 20 20 60 60 120 120 120 60 ...

$ MS.Zoning : chr "RL" "RH" "RL" "RL" ...

$ Lot.Frontage : int 141 80 81 93 74 78 41 43 39 60 ...

$ Lot.Area : int 31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...

$ Street : chr "Pave" "Pave" "Pave" "Pave" ...

$ Alley : chr NA NA NA NA ...

$ Lot.Shape : chr "IR1" "Reg" "IR1" "Reg" ...

$ Land.Contour : chr "Lvl" "Lvl" "Lvl" "Lvl" ...

#### Types de données :

* 28 colonnes numériques : Ces colonnes contiennent des valeurs continues ou discrètes (ex. : surface du terrain, surface habitable, nombre de chambres).
* 43 colonnes catégorielles :Ces colonnes représentent des caractéristiques qualitatives, telles que le type de rue, le type de toiture, ou encore le quartier.Certaines colonnes comportent des valeurs manquantes, telles que Alley, Mas Vnr Type, et Pool QC, ce qui peut nécessiter un traitement spécifique .

# Préparation du Jeu de Données : Nettoyage et Gestion des Valeurs Manquantes

## Supprimer l’espace dans le nom du colonne:

names(df) <- gsub(" ", "", names(df))

## Supprimer la colonne PID(ID):

df$PID <- NULL

Verifier que la colonne a ete supprime:

dim(df)

[1] 2930 81

## Gérer les valeurs manquantes :

# Calculer le pourcentage de valeurs manquantes

pourcentage <- colSums(is.na(df)) / nrow(df) \* 100

# Afficher les colonnes avec des valeurs manquantes et leur pourcentage

pourcentage[pourcentage > 0][order(-pourcentage[pourcentage > 0])]

Pool.QC Misc.Feature Alley Fence Fireplace.Qu Lot.Frontage Garage.Yr.Blt Garage.Qual Garage.Cond

99.55631399 96.38225256 93.24232082 80.47781570 48.53242321 16.72354949 5.42662116 5.39249147 5.39249147

Garage.Type Garage.Finish Bsmt.Qual Bsmt.Cond Bsmt.Exposure BsmtFin.Type.1 BsmtFin.Type.2 Mas.Vnr.Area Bsmt.Full.Bath

5.35836177 5.35836177 2.69624573 2.69624573 2.69624573 2.69624573 2.69624573 0.78498294 0.06825939

Bsmt.Half.Bath BsmtFin.SF.1 BsmtFin.SF.2 Bsmt.Unf.SF Total.Bsmt.SF Garage.Cars Garage.Area

0.06825939 0.03412969 0.03412969 0.03412969 0.03412969 0.03412969 0.03412969

#### Si une colonne a plus de 40% de valeurs manquantes, on la supprime:

# Identifier les colonnes avec plus de 40% de valeurs manquantes

colonnes\_supprimer <- names(pourcentage[pourcentage > 40])

# Supprimer les colonnes avec plus de 40% de valeurs manquantes

df <- df[, !(names(df) %in% colonnes\_supprimer)]

#### Pour les colonnes ayant moins de 40% de valeurs manquantes, on remplace les valeurs manquantes par la médiane (pour les variables numériques) ou le mode (pour les variables catégorielles) :

for (col in names(df)) {

if (any(is.na(df[[col]]))) {

if (is.numeric(df[[col]])) {

# Remplacer les NA par la médiane pour les colonnes numériques

df[[col]][is.na(df[[col]])] <- median(df[[col]], na.rm = TRUE)

} else {

# Remplacer les NA par la valeur la plus fréquente (mode) pour les colonnes catégorielles

mode\_val <- names(sort(table(df[[col]]), decreasing = TRUE))[1]

df[[col]][is.na(df[[col]])] <- mode\_val } }}

#### Verification :

# Calculer le pourcentage de valeurs manquantes

pourcentage <- colSums(is.na(df)) / nrow(df) \* 100

# Afficher les colonnes avec des valeurs manquantes et leur pourcentage

pourcentage[pourcentage > 0][order(-pourcentage[pourcentage > 0])]

named numeric(0)

#### Aucune valeur manquante.

## Gérer les doublons:

doublons <- sum(duplicated(df))

print(doublons)

[1] 0

#### Il n’y a aucune ligne dupliquée dans le Dataset

# 

# Paramètres statistiques usuels

## Table de statistiques descriptives:

summary(df)

Order MS.SubClass MS.Zoning

Min. : 1.0 Min. : 20.00 Length:2930

1st Qu.: 733.2 1st Qu.: 20.00 Class :character

Median :1465.5 Median : 50.00 Mode :character

Mean :1465.5 Mean : 57.39

3rd Qu.:2197.8 3rd Qu.: 70.00

Max. :2930.0 Max. :190.00

Lot.Frontage Lot.Area Street

Min. : 21.00 Min. : 1300 Length:2930

1st Qu.: 60.00 1st Qu.: 7440 Class :character

Median : 68.00 Median : 9436 Mode :character

Mean : 69.02 Mean : 10148

3rd Qu.: 78.00 3rd Qu.: 11555

Max. :313.00 Max. :215245

Lot.Shape Land.Contour Utilities

Length:2930 Length:2930 Length:2930

Class :character Class :character Class :character

Mode :character Mode :character Mode :character

Lot.Config Land.Slope Neighborhood

Length:2930 Length:2930 Length:2930

Class :character Class :character Class :character

Mode :character Mode :character Mode :character

Condition.1 Condition.2 Bldg.Type

Length:2930 Length:2930 Length:2930

Class :character Class :character Class :character

Mode :character Mode :character Mode :character

# Analyses univariées :

## Catégorielles :

La variable **Neighborhood** représente le quartier où se situe chaque maison dans la ville d’Ames. C’est une variable catégorielle avec plusieurs modalités, chacune correspondant à un quartier différent.

table(df$Neighborhood)

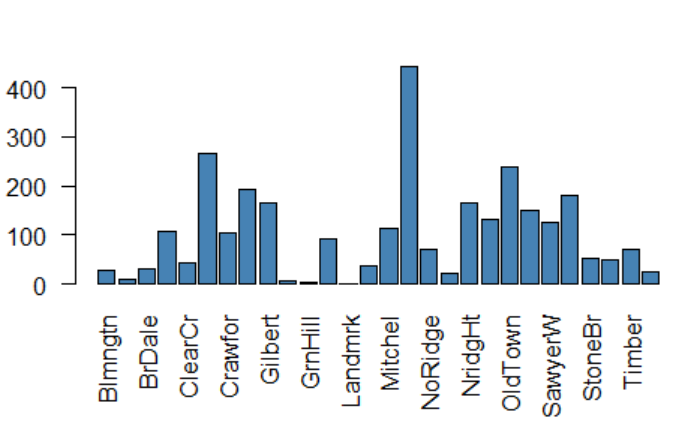
Blmngtn Blueste BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert Greens GrnHill IDOTRR Landmrk MeadowV Mitchel NAmes NoRidge

28 10 30 108 44 267 103 194 165 8 2 93 1 37 114 443 71

NPkVill NridgHt NWAmes OldTown Sawyer SawyerW Somerst StoneBr SWISU Timber Veenker

23 166 131 239 151 125 182 51 48 72 24

barplot(table(df$Neighborhood), las=2, col="steelblue")



#### Le résultat montre combien de maisons sont présentes dans chaque quartier du dataset. Par exemple :

* Le quartier NAmes contient 443 maisons, c’est le plus représenté.
* CollgCr en a 267, OldTown en a 239, etc.
* Certains quartiers comme Landmrk ou GrnHill ont très peu d’observations (1 ou 2 maisons).

## Numériques :

### Variable continue:

On va analyser la variable continue SalePrice — Prix de vente de la maison

hist\_obj <- hist(df$SalePrice,

col = "blue",

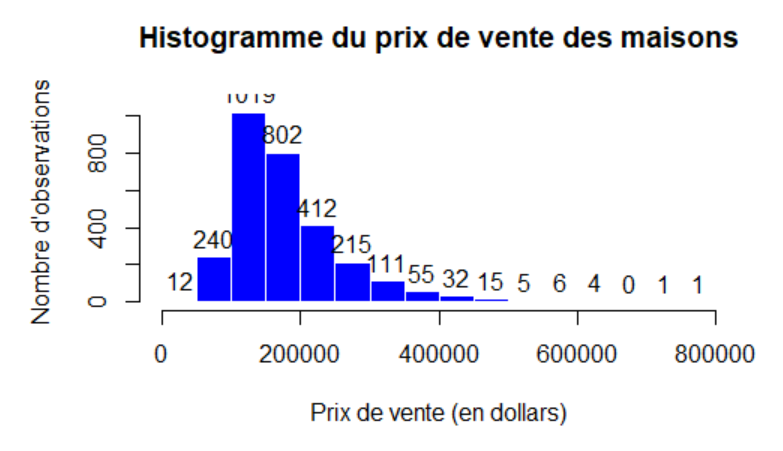
border = "white",

labels = TRUE,

xlab = "Prix de vente (en dollars)",

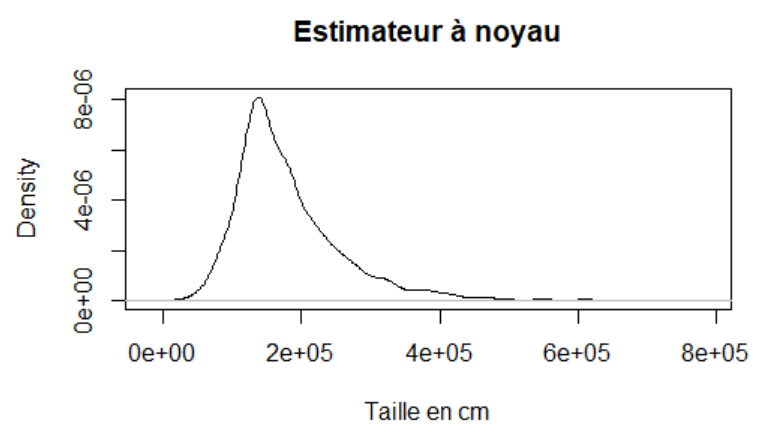
ylab = "Nombre d'observations",

main = "Histogramme du prix de vente des maisons")



plot (density(df$SalePrice), main="Estimateur à noyau",

xlab="Taille en cm")



#### L’histogramme de ‘SalePrice’ montre que les prix de vente sont majoritairement concentrés sur la gauche de la distribution, ce qui signifie qu’une grande partie des maisons ont des prix relativement bas. Il y a probablement quelques maisons très chères qui créent une queue à droite (valeurs extrêmes), mais la majorité des prix se situe dans une plage plus modeste. Cela suggère une distribution asymétrique à droite.

* Forme : La distribution est asymétrique à droite, avec une majorité de maisons ayant des prix de vente plus bas. Une queue à droite est présente, ce qui signifie que quelques maisons ont des prix beaucoup plus élevés que la majorité.
* Centre : Le centre de la distribution se situe autour de 180,000 USD. Cela suggère que la médiane et la moyenne sont probablement proches, bien que la queue à droite puisse influencer légèrement la moyenne vers le haut.
* Dispersion : La dispersion des prix de vente est assez large, avec des prix allant de relativement bas à très élevés. Les 1er et 3e quartiles (120,000 USD et 250,000 USD) capturent la majorité des données.
* Outliers (valeurs aberrantes) : Les outliers sont présents dans la queue à droite de la distribution, où l’on observe des maisons dont les prix sont beaucoup plus élevés que la majorité. Ces valeurs peuvent être identifiées comme des outliers en utilisant l’IQR.

### Variable discrete:

On va analyser la variable discrète MS Zoning — Zonage résidentiel

q=(levels(factor(df$`MS.Zoning`)))

[1] "A (agr)" "C (all)" "FV" "I (all)" "RH" "RL" "RM"

freq\_table <- table(df$`MS.Zoning`)

prc <- (freq\_table / nrow(df)) \* 100

prc <- round(prc, 3) # 3 chiffres apres virgule

prc

A (agr) C (all) FV I (all) RH RL RM

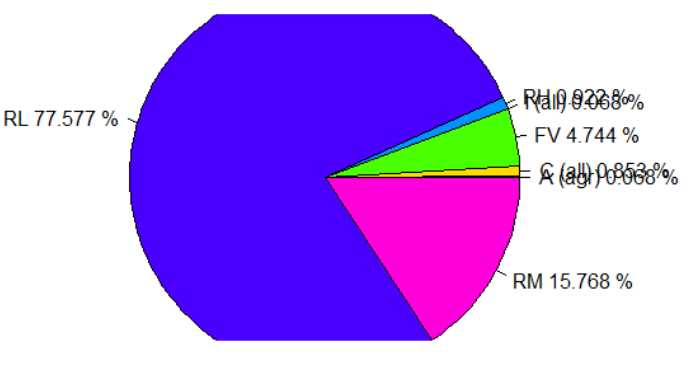
0.068 0.853 4.744 0.068 0.922 77.577 15.768

#### La fonction table() montre la distribution de fréquence de la variable MS Zoning

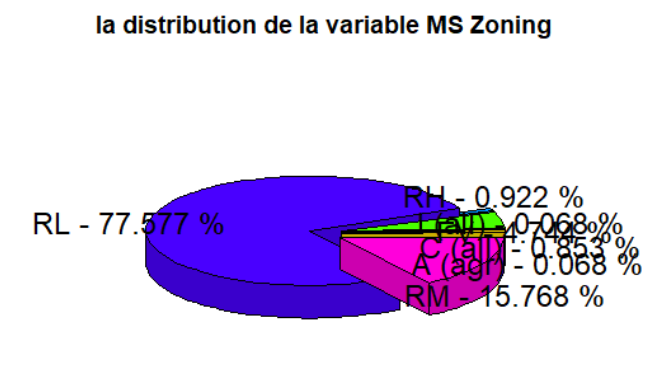
#### La variable ‘MS Zoning’ indique la classification des zones dans lesquelles se trouvent les propriétés. Elle contient différentes catégories :

* RL (Residential Low Density) : Zones résidentielles à faible densité (majoritaire avec 2273 enregistrements).
* RM (Residential Medium Density) : Zones résidentielles à densité moyenne (462 enregistrements).
* FV (Floating Village Residential) : Zones résidentielles flottantes (139 enregistrements).
* RH (Residential High Density) : Zones résidentielles à haute densité (27 enregistrements).
* C (all) : Zones commerciales (25 enregistrements).
* I (all) : Zones industrielles (2 enregistrements).
* A (agr) : Zones agricoles (2 enregistrements).

pie(freq\_table,col = rainbow(length(q)), labels = paste(q, prc, "%"),radius = 1.3 )



pie3D(freq\_table,explode = 0.1, main = "la distribution de la variable MS Zoning", labels = paste(q,"-",prc,"%"))



# Analyses bivariées

## Deux variables quantitatives:

On choisit ici :

* **SalePrice** (prix de vente de la maison)
* **TotRms.AbvGrd** (nombre de pièces au-dessus du sol)

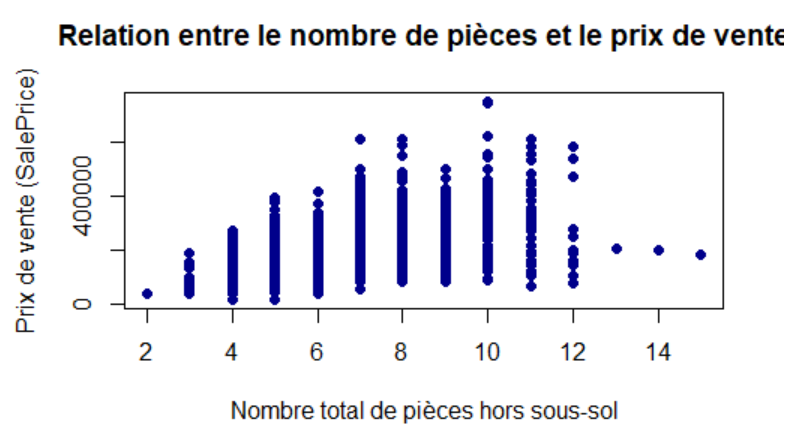
plot(x = df$TotRms.AbvGrd, y = df$SalePrice,

xlab = "Nombre total de pièces hors sous-sol",

ylab = "Prix de vente (SalePrice)",

main = "Relation entre le nombre de pièces et le prix de vente",

pch = 16, col = "darkblue")



tmp <- df[, c("TotRms.AbvGrd", "SalePrice")]

tmp <- tmp[complete.cases(tmp), ]

dens <- kde2d(tmp$TotRms.AbvGrd, tmp$SalePrice)

filled.contour(dens,

color = terrain.colors,

xlab = "Nombre total de pièces",

ylab = "Prix de vente",

main = "Carte de densité : TotRms.AbvGrd vs SalePrice")

tmp <- df[, c("TotRms.AbvGrd", "SalePrice")]

tmp <- tmp[complete.cases(tmp), ]

# Densité 2D

dens <- kde2d(tmp$TotRms.AbvGrd, tmp$SalePrice)

# Représentation

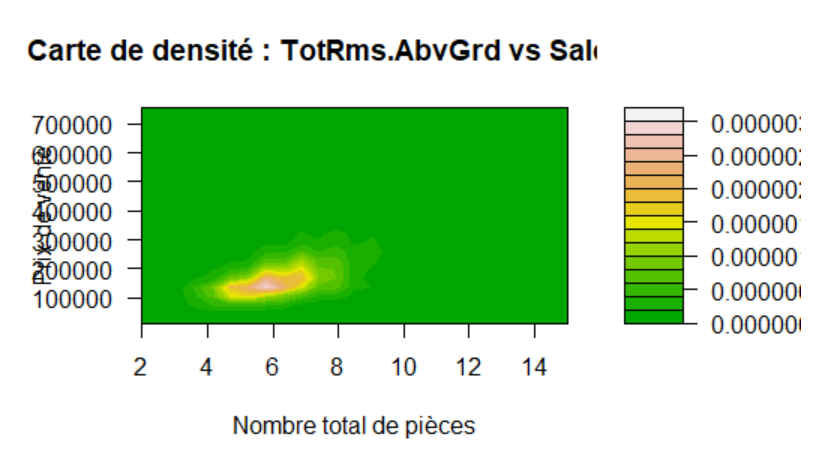
filled.contour(dens,

color = terrain.colors,

xlab = "Nombre total de pièces",

ylab = "Prix de vente",

main = "Carte de densité : TotRms.AbvGrd vs SalePrice")



Ici, on explore la relation entre le prix de vente(SalePrice) et nombre de pièces au-dessus du sol (TotRms.AbvGrd), en ajoutant une variable qualitative (Neighborhood) pour colorer les points selon deux quartiers choisis.

Choisir deux quartiers, par exemple "NAmes" et "OldTown"

plot(df$TotRms.AbvGrd[df$Neighborhood == "NAmes"],

df$SalePrice[df$Neighborhood == "NAmes"],

pch = 16, col = "blue",

main = "Prix de vente selon le nombre de pièces\n(NAmes vs OldTown)",

xlab = "Nombre total de pièces hors sous-sol",

ylab = "Prix de vente", las = 1)

# Deuxième quartier : OldTown (rouge)

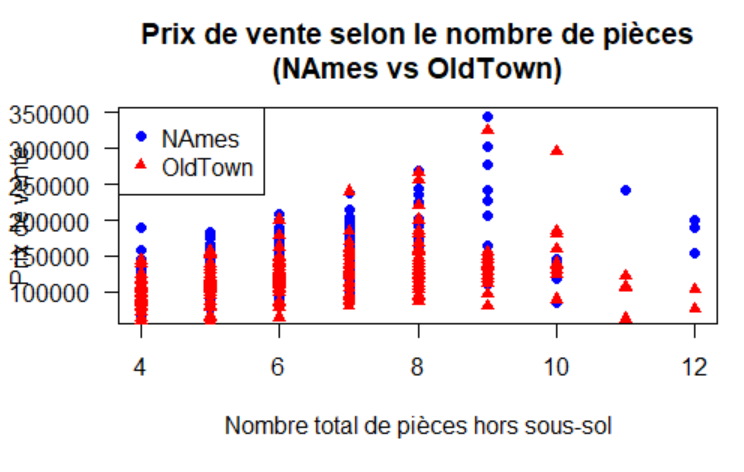
points(df$TotRms.AbvGrd[df$Neighborhood == "OldTown"],

df$SalePrice[df$Neighborhood == "OldTown"],

pch = 17, col = "red")

legend("topleft", legend = c("NAmes", "OldTown"),

col = c("blue", "red"), pch = c(16, 17))



#### Le graphique comparant les quartiers NAmes et OldTown montre que, de manière générale, le prix de vente des maisons augmente avec le nombre total de pièces hors sous-sol. Toutefois, à nombre de pièces équivalent, les maisons situées dans le quartier NAmes tendent à être vendues à un prix plus élevé que celles d’OldTown, ce qui suggère une influence notable du quartier sur la valeur des biens immobiliers. Malgré cette tendance, une certaine dispersion des prix est observée pour un même nombre de pièces, indiquant que d'autres variables entrent également en jeu dans la détermination du prix de vente..

correlation <- cor(df$SalePrice, df$TotRms.AbvGrd, use = "complete.obs")

print(paste("Coefficient de corrélation: ", correlation))

[1] "Coefficient de corrélation: 0.495474416857035"

* Le **coefficient de corrélation de 0.495** entre le nombre total de pièces hors sous-sol (TotRms.AbvGrd) et le prix de vente (SalePrice) indique une **corrélation modérément positive**.

Cela signifie que, **en général**, à mesure que le nombre de pièces augmente, le prix de vente a tendance à augmenter également.

regression\_model <- lm(SalePrice ~ TotRms.AbvGrd, data = df)

summary(regression\_model)

Call:

lm(formula = SalePrice ~ TotRms.AbvGrd, data = df)

Residuals:

Min 1Q Median 3Q Max

-244131 -39148 -10680 30188 484696

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 18665.4 5407.0 3.452 0.000564 \*\*\*

TotRms.AbvGrd 25163.8 815.3 30.866 < 2e-16 \*\*\*

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 69400 on 2928 degrees of freedom

Multiple R-squared: 0.2455, Adjusted R-squared: 0.2452

F-statistic: 952.7 on 1 and 2928 DF, p-value: < 2.2e-16

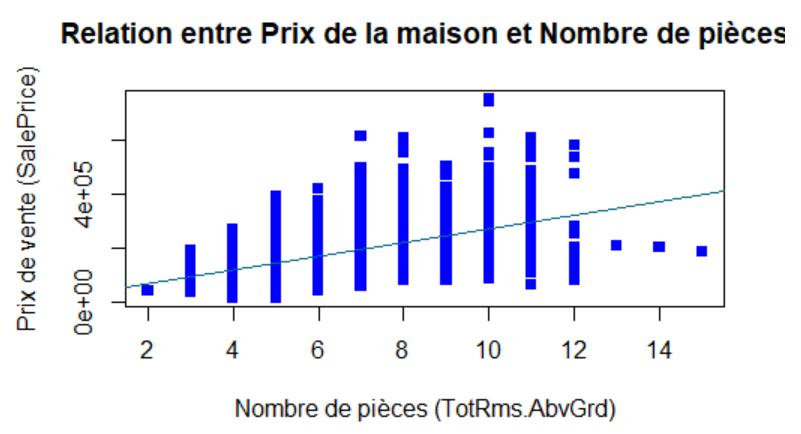
* **Modèle linéaire obtenu :** SalePrice = 18 665 + 25 164 \* TotRms.AbvGrd
* **Pente (25 164) :** Chaque pièce supplémentaire hors sous-sol est associée à une augmentation moyenne de 25 164 $ du prix de vente.
* **Ordonnée à l’origine (18 665) :** Prix estimé d’un bien avec 0 pièce (valeur théorique sans signification pratique directe).
* **Coefficient de détermination (R² = 0.245) :** Le modèle explique environ 24,5 % de la variabilité du prix de vente.
* **Conclusion :** Le nombre de pièces a un effet modéré sur le prix de vente, mais d’autres facteurs importants influencent également ce dernier.

# Visualisation

plot(df$TotRms.AbvGrd, df$SalePrice, main = "Relation entre Prix de la maison et Nombre de pièces",

xlab = "Nombre de pièces (TotRms.AbvGrd)", ylab = "Prix de vente (SalePrice)", col = "blue", pch = 15)

abline(regression\_model, col = "deepskyblue4")



# Intervalle de confiance pour SalePrice

result\_saleprice <- t.test(df$SalePrice)

cat("Intervalle de confiance pour SalePrice:\n")

print(result\_saleprice$conf.int)

# Intervalle de confiance pour TotRms.AbvGrd

result\_totrms <- t.test(df$TotRms.AbvGrd)

cat("\nIntervalle de confiance pour TotRms.AbvGrd:\n")

print(result\_totrms$conf.int)

Intervalle de confiance pour SalePrice:

[1] 177902.3 183689.9

attr(,"conf.level")

[1] 0.95

Intervalle de confiance pour TotRms.AbvGrd:

[1] 6.386025 6.499982

attr(,"conf.level")

[1] 0.95

* L'intervalle de confiance à 95 % pour le **prix de vente** (SalePrice) se situe entre **177,902.3 et 183,689.9**. Cela signifie qu'avec une confiance de 95 %, la valeur moyenne réelle du prix de vente dans la population se trouve dans cet intervalle.
* L'intervalle de confiance à 95 % pour le **nombre total de chambres au-dessus du sol** (TotRms.AbvGrd) se situe entre **6.39 et 6.50**. Cela indique que la moyenne réelle du nombre de chambres au-dessus du sol dans la population est estimée être comprise entre ces deux valeurs

## Deux variables qualitatives:

On choisit ici :

* **CentralAir**: This variable indicates whether the house has central air conditioning ("Y" for yes, "N" for no).
* **Fireplaces**: This variable indicates the number of fireplaces in the house, with values such as "0", "1", "2", etc.

t = table(df$Central.Air, df$Fireplaces)

t

0 1 2 3 4

N 163 31 2 0 0

Y 1259 1243 219 12 1

prop.table(t)

0 1 2 3

N 0.0556313993 0.0105802048 0.0006825939 0.0000000000

Y 0.4296928328 0.4242320819 0.0747440273 0.0040955631

4

N 0.0000000000

Y 0.0003412969

chisq.test(df$Central.Air, df$Fireplaces)

Pearson's Chi-squared test

data: df$Central.Air and df$Fireplaces

X-squared = 101.66, df = 4, p-value <

0.00000000000000022

mosaicplot(Central.Air ~ Fireplaces, data = df, shade = TRUE, main = "Mosaic Plot")

