Projet analyses statistiques du data set Ames Housing avec R

Khedri maha
khedrimaha@gmail.com
2BDAD2

Table des matières

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

1- Supprimer l'espace dans le nom du colonne:

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

2- Supprimer la colonne PID(ID):

```
df$PID <- NULL
```

Verifier que la colonne a ete supprime:

```
dim(df)
```

```
[1] 2930   81
```

3- 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   0.068
25939
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
```

a- 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)]
```

b- 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 } }}
```

c- 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.

4- Gérer les doublons:

```
doublons <- sum(duplicated(df))
print(doublons)
```

```
[1] 0
```

- Il n'y a aucune ligne dupliquée dans le Dataset

Paramètres statistiques usuels

1- 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 :

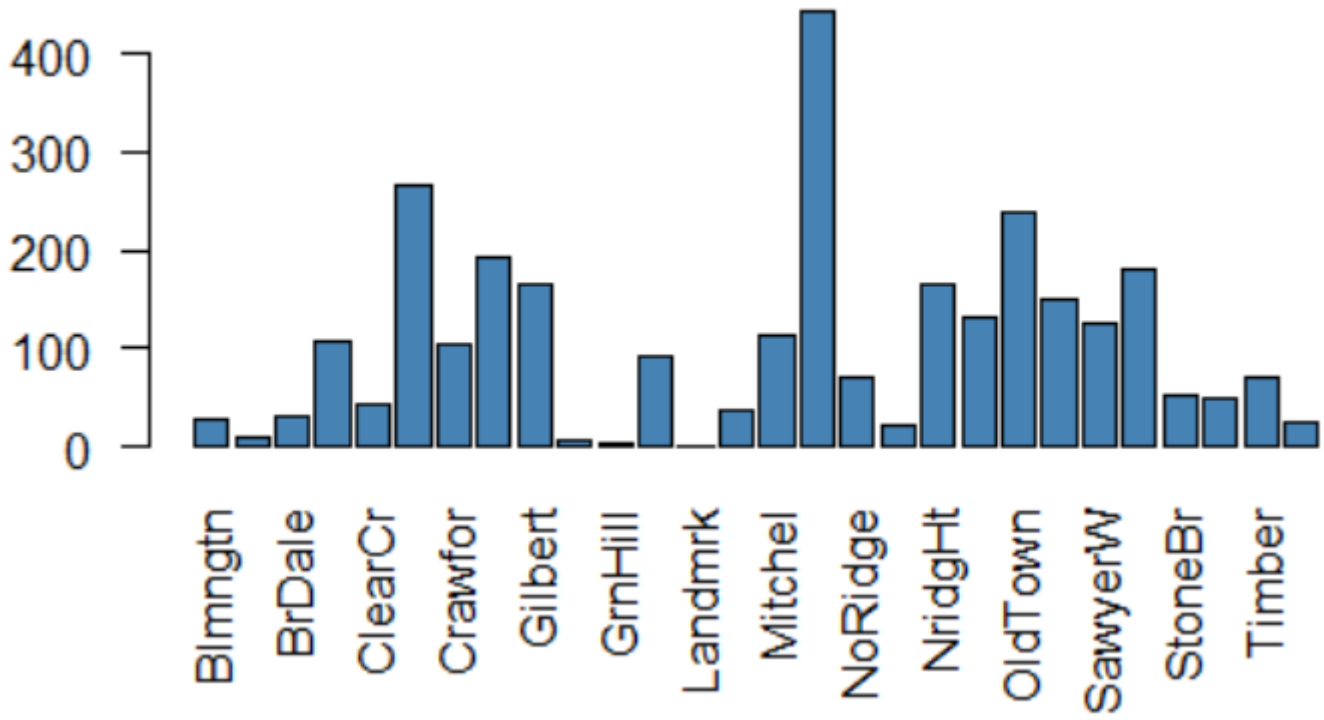
1- Catégorielles :

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 
     28      10      30     108      44     267     103     194     165       8       2      93       1 
MeadowV Mitchel   NAmes NoRidge NPkVill NridgHt NWAmes OldTown  Sawyer SawyerW Somerst StoneBr SWISU 
     37     114     443      71      23     166     131     239     151     125     182      51      48 
 Timber Veenker 
     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 :

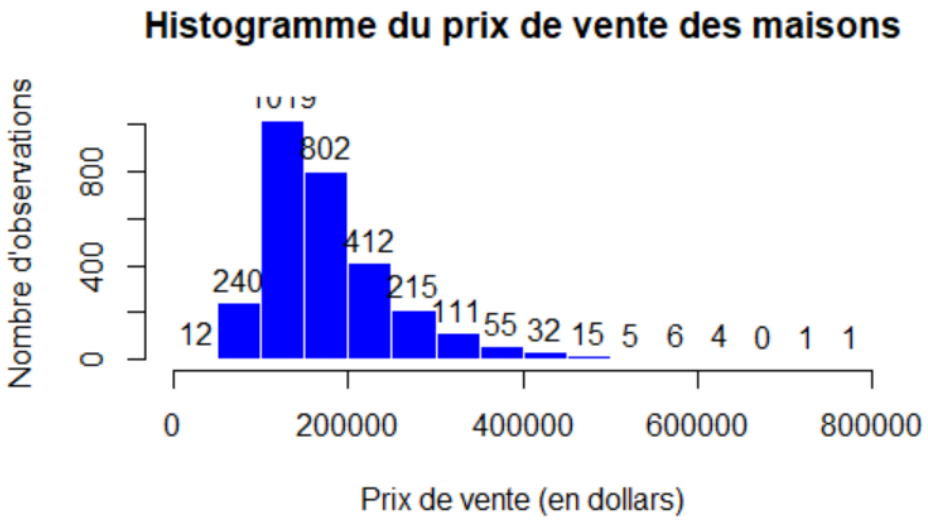
- Le quartier NAmes contient 443 maisons, c'est le plus représenté.
- CollgCr en a 267, OldTown en a 239, etc.
- Certains quartiers comme Landmrk ou GrnHill ont très peu d'observations (1 ou 2 maisons).

2- Numériques :

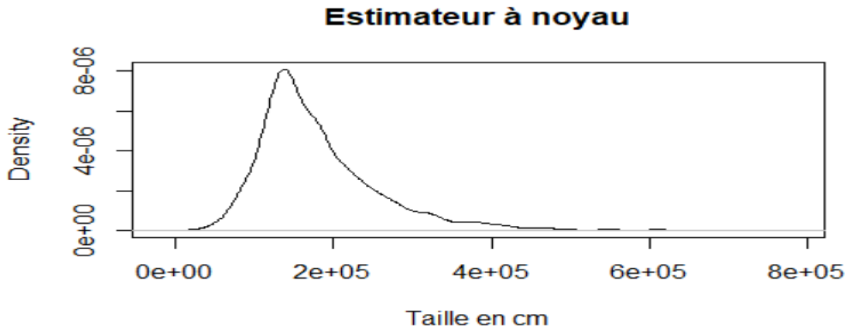
a- Variable continue:

La variable **SalePrice** représente le **prix de vente** de chaque maison au moment de la transaction. C'est une variable quantitative continue exprimée en dollars. Elle constitue la variable cible dans le contexte d'un modèle prédictif, car elle reflète la valeur marchande du bien immobilier et dépend de nombreuses caractéristiques telles que l'emplacement, la surface, l'état général ou encore l'année de construction.

```
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.

b- Variable discrete:

La variable **GarageCars** représente le **nombre de places de stationnement** disponibles dans le garage de chaque maison. C'est une variable quantitative discrète, car elle prend des valeurs entières (0, 1, 2, etc.) correspondant au nombre de voitures pouvant être garées. Elle peut avoir un impact direct sur la valeur perçue d'un bien immobilier, car un garage plus spacieux est souvent associé à un meilleur confort.

```
q=(levels(factor(df$`Garage.Cars`)))
```

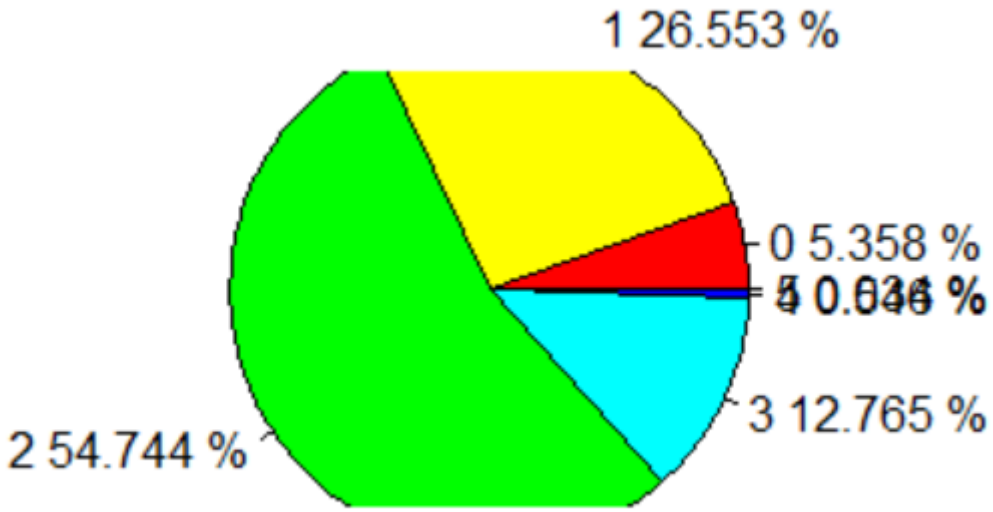
```
[1] "0" "1" "2" "3" "4" "5"
```

```
freq_table <- table(df$`Garage.Cars`)
prc <- (freq_table / nrow(df)) * 100
prc <- round(prc, 3)  # 3 chiffres apres virgule
prc
```

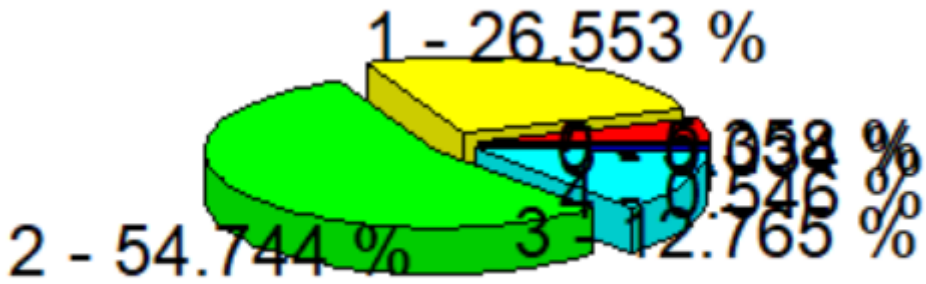
```
➢    0     1      2      3      4     5
➢  5.358 26.553 54.744 12.765  0.546 0.034
```

- ➢ La fonction table() montre la distribution de fréquence de la variable GarageCars
- ➢ La majorité des maisons dans le dataset ont un garage pour **2 voitures** (environ 54,7%). Une proportion notable de maisons ont également des garages pour **1 voiture** (26,6%), mais les maisons avec des garages pour plus de 2 voitures sont beaucoup moins fréquentes, avec seulement 12,8% pour 3 voitures et des valeurs encore plus faibles pour 4 ou 5 voitures.

```
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 GarageCars", labels = paste(q,"-",prc,"%"))
```



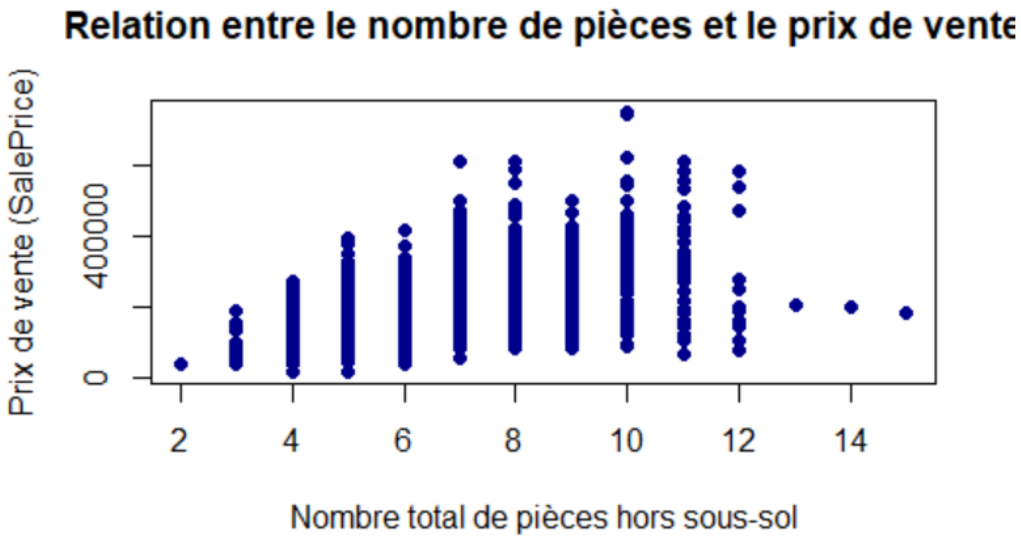
Analyses bivariées

1- Deux variables quantitatives:

On choisit ici :

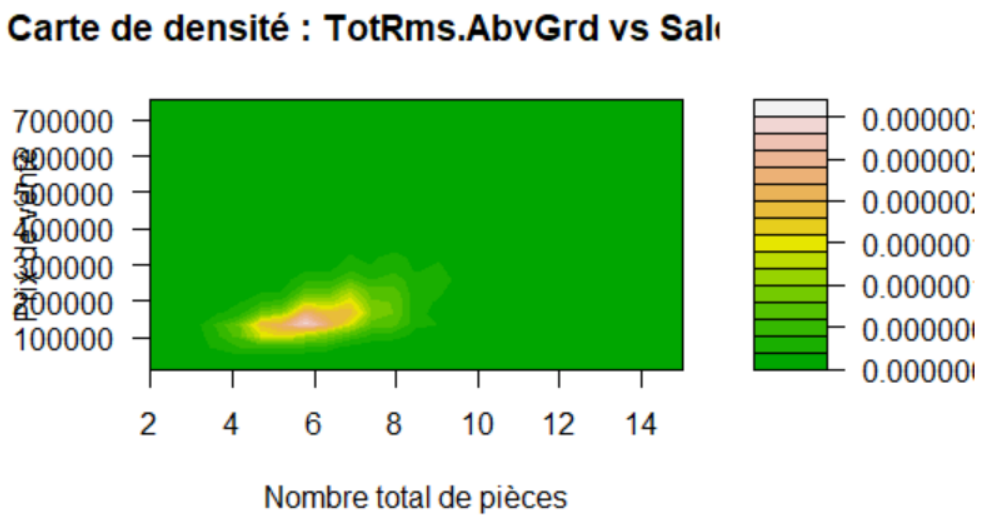
- **SalePrice** (prix de vente de la maison)
- **TotRms.AbvGrd** (nombre de pièces au-dessus du sol)

```
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")]
```

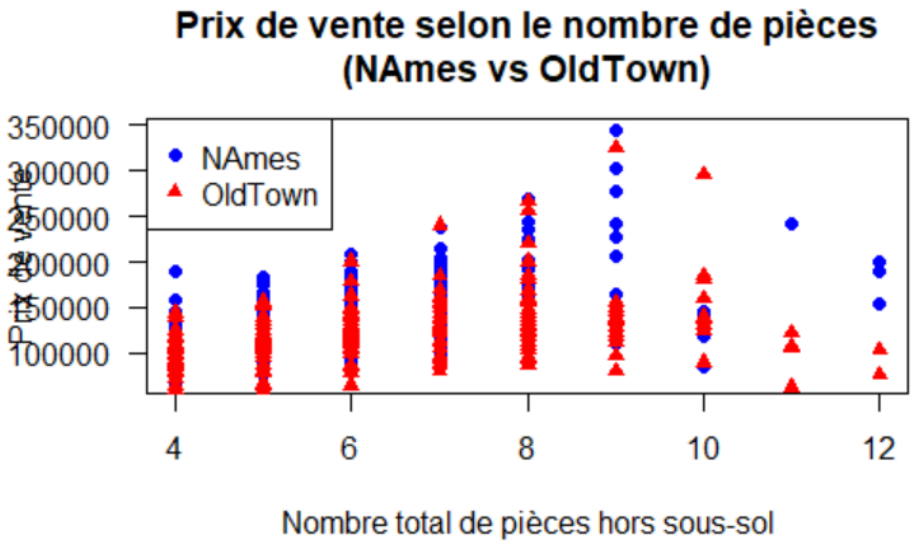
```
tmp <- tmp[complete.cases(tmp), ]
dens <- kde2d(tmp$TotRms.AbvGrd, tmp$SalePrice)
filled.contour(dens,
        color = terrain.colors,
        xlab = "Nombre total de pièces",
        ylab = "Prix de vente",
        main = "Carte de densité : TotRms.AbvGrd vs SalePrice")
tmp <- df[, c("TotRms.AbvGrd", "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.

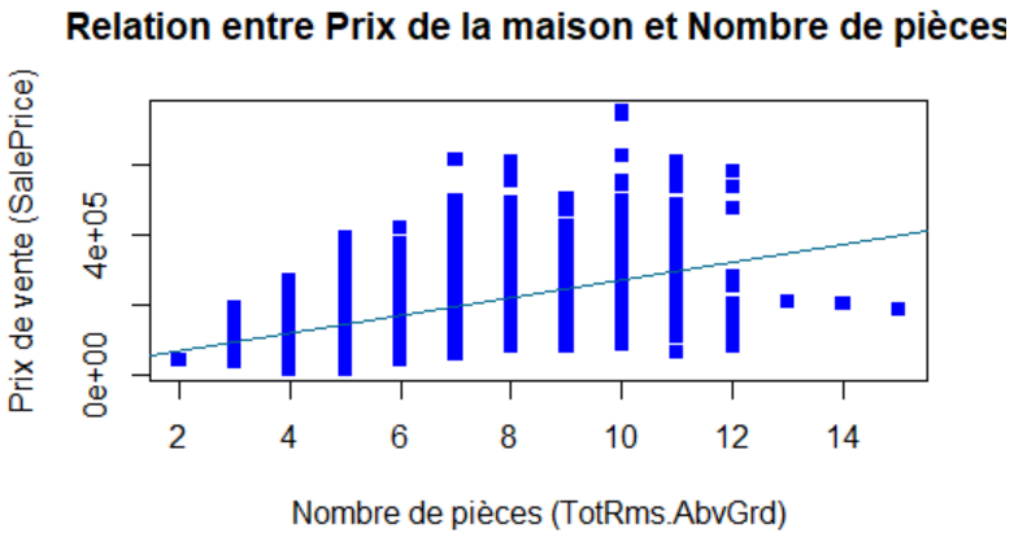
```
regression_model <- lm(SalePrice ~ TotRms.AbvGrd, data = df)
summary(regression_model)
```

```
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 (R^2 = 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

2- Deux variables qualitatives:

On choisit ici :

- **GarageFinish** : Cette variable indique la finition du garage, avec les modalités suivantes :

"Fin" : Garage terminé (fini).

"RFn" : Garage avec finition brute (finir au besoin).

"Unf" : Garage non fini (non aménagé).

- **GarageType** : Cette variable représente le type de garage, avec les modalités suivantes :

"Attchd" : Garage attaché (attaché à la maison).

"Detchd" : Garage détaché (séparé de la maison).

"Basment" : Garage au sous-sol.

"BuiltIn" : Garage intégré (partie de la structure de la maison).

```
able_contingence <- table(df$Garage.Finish, df$Garage.Type)
table_contingence
```

```
     2Types Attchd Basment BuiltIn CarPort Detchd
      0    0     0     0    0    2
 Fin    3   561    11   127    0   26
 RFn    5   717     8    48    0   34
 Unf   15   610    17    11   15   720
```

```
prop.table (table_contingence)
```

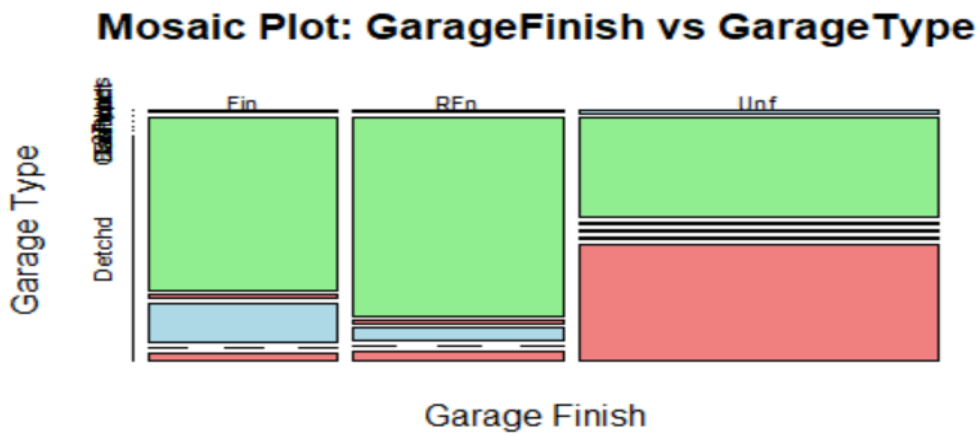
```
      2Types      Attchd     Basment     BuiltIn
   0.0000000000 0.0000000000 0.0000000000 0.0000000000
Fin 0.0010238908 0.1914675768 0.0037542662 0.0433447099
RFn 0.0017064846 0.2447098976 0.0027303754 0.0163822526
Unf 0.0051194539 0.2081911263 0.0058020478 0.0037542662

       CarPort      Detchd
   0.0000000000 0.0006825939
Fin 0.0000000000 0.0088737201
RFn 0.0000000000 0.0116040956
Unf 0.0051194539 0.2457337884
```

```
chisq_test <- chisq.test(table_contingence)
```

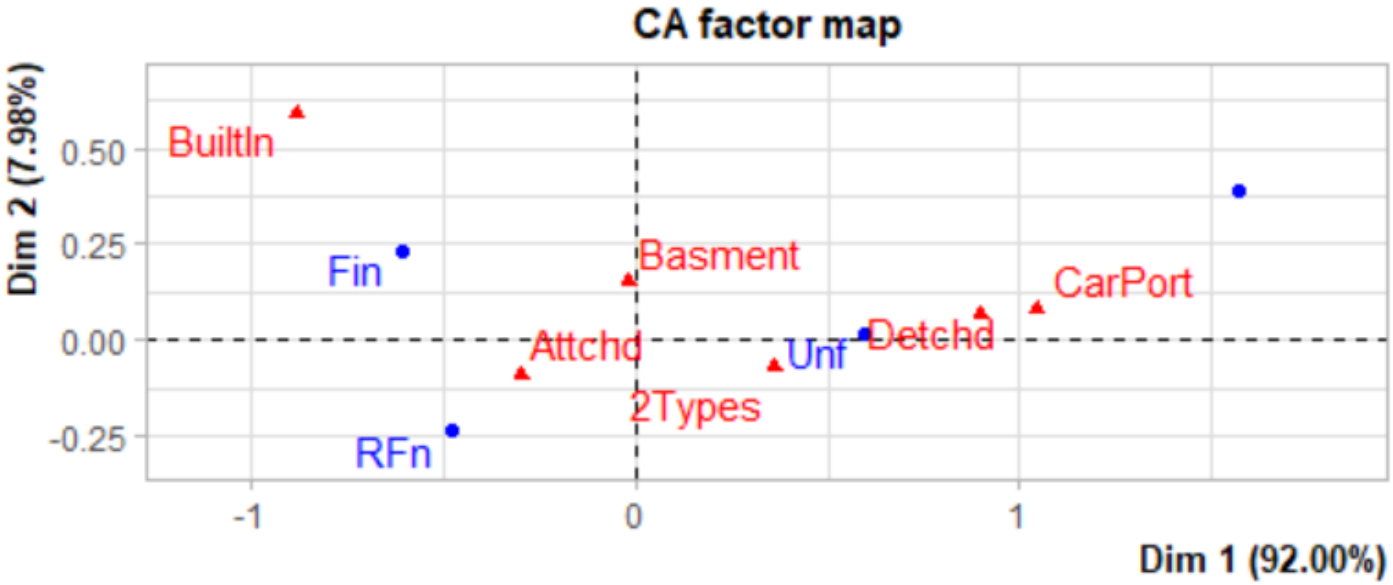
```
Pearson's Chi-squared test
data:  table_contingence
X-squared = 1044.9, df = 15, p-value < 2.2e-16
```

- **Statistique de test (X-squared = 1044.9)** : Mesure l'écart entre les fréquences observées et attendues. Plus elle est élevée, plus il y a une association entre les variables.
- **Degrés de liberté (df = 15)** : Représente le nombre de possibilités d'indépendance dans le tableau de contingence.
- **p-value < 2.2e-16** : Une p-value extrêmement faible indique que l'association observée est statistiquement significative.
- Conclusion :La p-value étant inférieure à 0,05, **vous rejetez l'hypothèse nulle** (indépendance des variables) et concluez que **les deux variables sont statistiquement associées**.

```
mosaicplot(table_contingence, main = "Mosaic Plot: GarageFinish vs GarageType",
      color = c("lightblue", "lightgreen", "lightcoral"),
      xlab = "Garage Finish", ylab = "Garage Type")
```




```
resultat_afc <- CA(table_contingence, graph = TRUE)
```



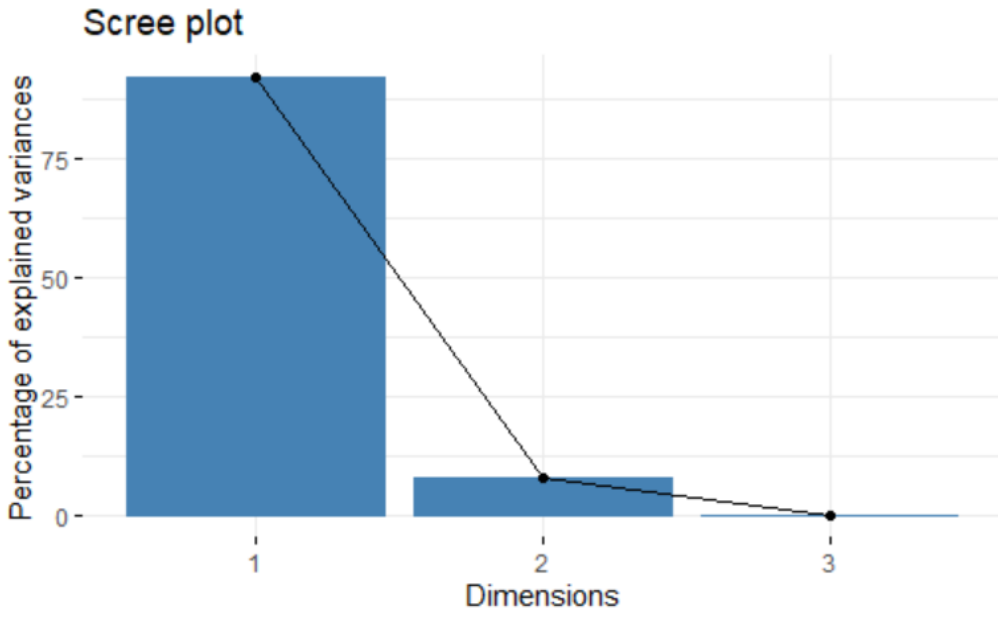

```
valeurs_propres <- resultat_afc$eig
valeurs_propres
```

```
      eigenvalue percentage of variance
dim 1 3.280878e-01          91.99513346
dim 2 2.845421e-02           7.97850266
```

```
dim 3 9.402308e-05          0.02636388
    cumulative percentage of variance
dim 1                  91.99513
dim 2                  99.97364
dim 3                 100.00000
```

- **Dimension 1** : Explique **92%** de la variance, c'est la dimension principale.
- **Dimension 2** : Explique **8%** de la variance, ajoutant un peu plus d'information.
- **Dimension 3** : Explique **0.03%** de la variance, négligeable.
- En cumulant les deux premières dimensions, vous expliquez **99.97%** de la variance, ce qui signifie que l'analyse peut être efficacement résumée en utilisant seulement ces deux dimensions

```
resultat_afc <- CA(table_contingence, graph = FALSE)
fviz_eig(resultat_afc)
```

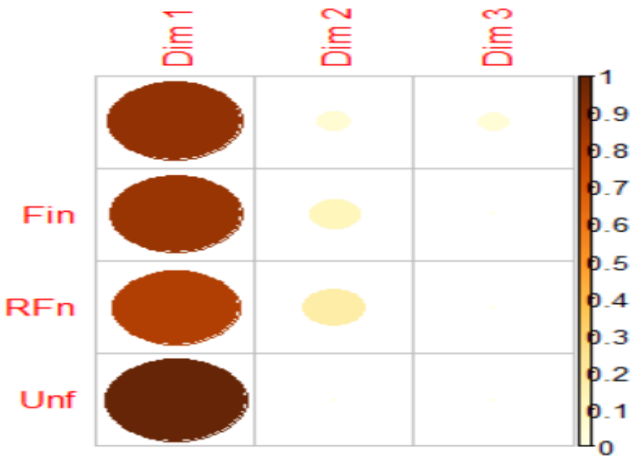



```
row = get_ca_row(resultat_afc)
row
```

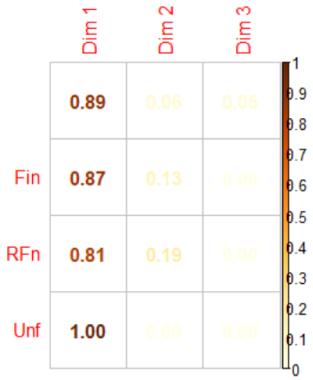
```
Correspondence Analysis - Results for rows
 ==================================================
```

Name	Description
1 "$coord"	"Coordinates for the rows"
2 "$cos2"	"Cos2 for the rows"
3 "$contrib"	"contributions of the rows"
4 "$inertia"	"Inertia of the rows"

```
corrplot(row$cos2, is.corr = FALSE)
```




```
corrplot(row$cos2, is.corr = FALSE,method="number")
```



À travers l'**Analyse des Correspondances** (AFC), nous avons exploré les relations entre les variables qualitatives **GarageFinish** (finishes du garage) et **GarageType** (type de garage). Voici les conclusions principales :

1. **Association entre les variables** : Les deux variables, **GarageFinish** et **GarageType**, présentent une association significative. Cela signifie que le type de garage (par exemple, "Attchd", "Detchd", "Basment") a une influence sur le finish du garage (par exemple, "Fin", "RFn", "Unf"), ce qui suggère qu'il existe des liens ou des préférences spécifiques entre ces modalités.
2. **Visualisation des relations** : Grâce au biplot de l'AFC, nous avons pu visualiser les relations entre les modalités des deux variables. Les points représentant les modalités de **GarageFinish** et **GarageType** qui sont proches dans l'espace graphique sont fortement associés, tandis que ceux qui sont éloignés l'un de l'autre sont moins liés.
3. **Interprétation des résultats** : L'analyse des contributions et des coordonnées a permis de mieux comprendre quels types de garage sont associés à quel type de finition de garage. Par exemple, un garage **attelé** pourrait être plus souvent associé à une finition **finie**, tandis qu'un garage **détaché** pourrait être plus souvent associé à une finition **non finie**.
4. **Dépendance statistique** : Le test du **chi carré** a confirmé que les deux variables sont dépendantes l'une de l'autre. Le résultat du test a montré que la valeur p était bien inférieure à 0,05, ce qui indique une relation significative entre les modalités des deux variables. Par conséquent, nous avons rejeté l'hypothèse d'indépendance entre ces variables.
5. **Réduction de la complexité** : L'AFC a permis de simplifier la compréhension des données qualitatives en réduisant leur dimensionnalité tout en conservant les informations importantes sur l'association entre **GarageFinish** et **GarageType**. Cette réduction a facilité l'identification des principales associations entre ces variables.

En résumé, l'AFC a révélé que **GarageFinish** et **GarageType** sont des variables significativement liées, et l'analyse a permis de mieux comprendre la structure des données et de visualiser les associations entre les catégories de ces variables.

Analyses multivariée

Pour effectuer une analyse multivariée sur un sous-ensemble de variables dans le jeu de données **Ames Housing**, nous allons suivre plusieurs étapes tout en gardant à l'esprit que l'on sélectionne un petit nombre de variables pour éviter que l'analyse devienne trop embrouillée. Voici les étapes que nous allons suivre :

1- Sélection des Variables

Nous allons choisir un sous-ensemble de variables numériques qui sont probablement intéressantes et qui ont une relation significative entre elles. Par exemple, nous pouvons choisir les variables suivantes :

- **Gr.Liv.Area** (Surface habitable)
- **OverallQual** (Qualité générale de la maison)
- **SalePrice** (Prix de vente)
- **GarageCars** (Nombre de places dans le garage)
- **TotRmsAbvGrd** (Nombre de chambres au-dessus du sol)

```
selected_variables <- df[, c("Gr.Liv.Area", "Overall.Qual", "SalePrice", "Garage.Cars", "TotRms.AbvGrd")]
selected_variables[is.na(selected_variables)] <- apply(selected_variables, 2, function(x) mean(x, na.rm = TRUE))
head(selected_variables)
```

	Gr.Liv.Area <int>	Overall.Qual <int>	SalePrice <int>	Garage.Cars <int>	TotRms.AbvGrd <int>	
1	1656	6	215000	2	7	
2	896	5	105000	1	5	
3	1329	6	172000	1	6	
4	2110	7	244000	2	8	
5	1629	5	189900	2	6	
6	1604	6	195500	2	7	

2- Préparation des Données

Nous allons préparer les données en nous assurant qu'il n'y a pas de valeurs manquantes et en effectuant la normalisation si nécessaire, ce qui est important pour certaines analyses multivariées (comme PCA et K-means)

```
selected_variables <- df[, c("Gr.Liv.Area", "Overall.Qual", "SalePrice", "Garage.Cars", "TotRms.AbvGrd")]
selected_variables[is.na(selected_variables)] <- apply(selected_variables, 2, function(x) mean(x, na.rm = TRUE))
head(selected_variables)
```

	Gr.Liv.Area <int>	Overall.Qual <int>	SalePrice <int>	Garage.Cars <int>	TotRms.AbvGrd <int>	
1	1656	6	215000	2	7	
2	896	5	105000	1	5	
3	1329	6	172000	1	6	
4	2110	7	244000	2	8	
5	1629	5	189900	2	6	
6	1604	6	195500	2	7	

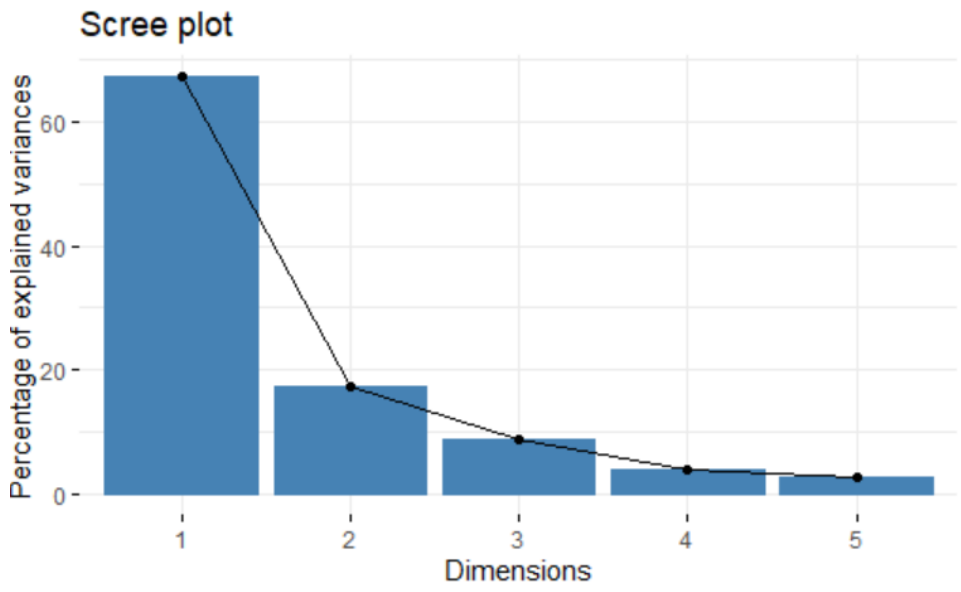
6 rows

Hide

3- Réalisation de l'Analyse en Composantes Principales (PCA)

L'Analyse en Composantes Principales (PCA) est utilisée pour réduire la dimensionnalité des données tout en préservant autant que possible la variance des données. Nous allons appliquer la PCA sur les variables sélectionnées et visualiser les résultats pour mieux comprendre la structure des données.

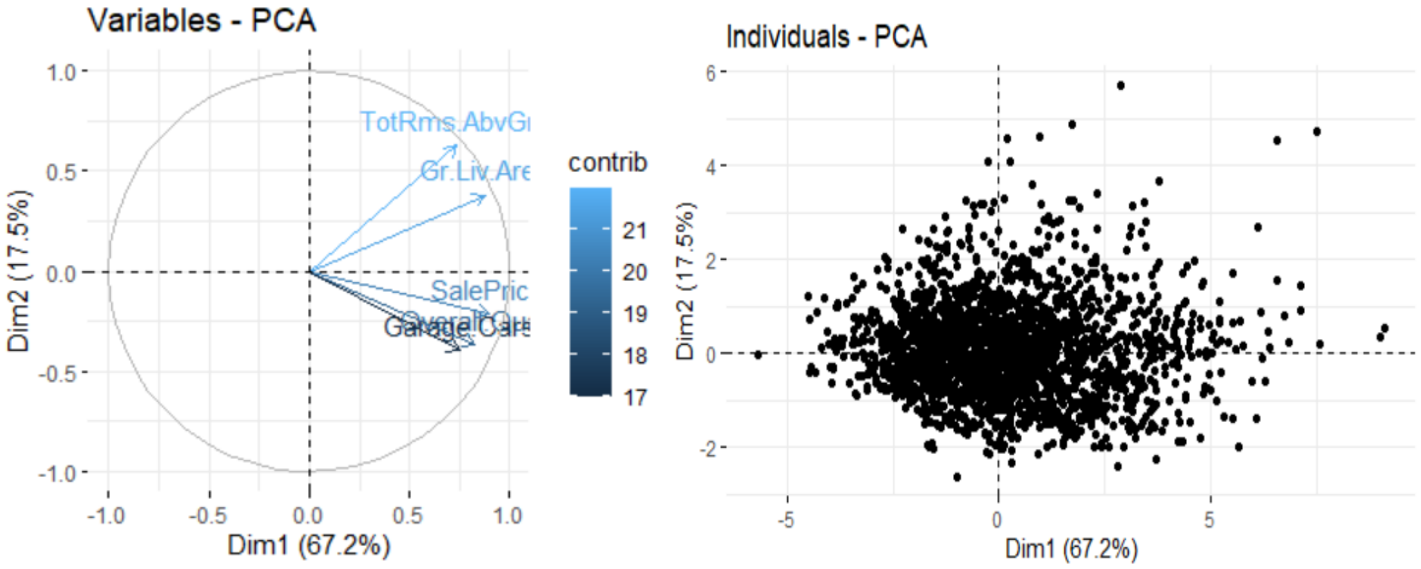
```
scaled_data <- scale(selected_variables)
library(FactoMineR)
pca_result <- PCA(scaled_data, graph = FALSE)
library(factoextra)
fviz_eig(pca_result)
```



4- Interprétation des Résultats de la PCA

Après avoir réalisé la PCA, nous allons examiner les valeurs propres (pour déterminer le nombre de dimensions importantes à retenir) et les contributions des variables sur les composantes principales.

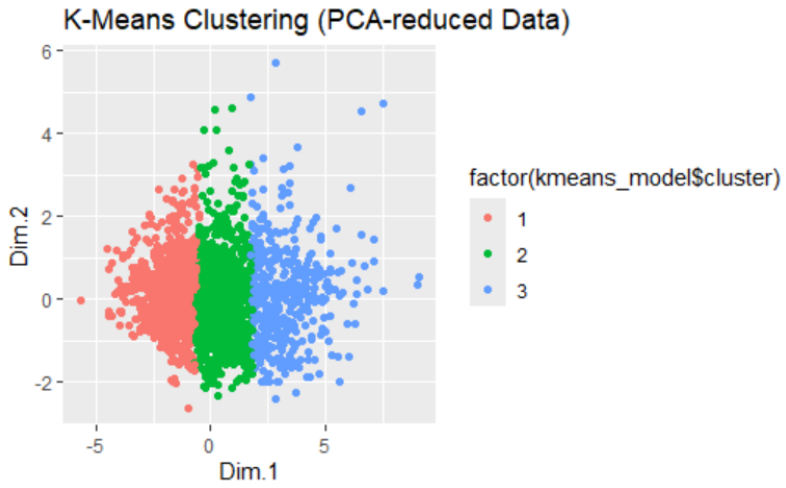
```
fviz_pca_var(pca_result, col.var = "contrib")
fviz_pca_ind(pca_result, geom = "point")
```



5- Réalisation de l'Analyse de Clustering K-Means

Une fois que nous avons réduit la dimensionnalité avec la PCA, nous pouvons appliquer un algorithme de clustering (comme **K-Means**) pour regrouper les maisons en fonction de leurs caractéristiques. Nous allons utiliser les deux premières composantes principales comme variables d'entrée pour K-means.

```
pca_data <- data.frame(pca_result$ind$coord[, 1:2])


# Application de l'algorithme K-means
set.seed(42)
kmeans_model <- kmeans(pca_data, centers = 3, nstart = 25)


# Visualisation des clusters
library(ggplot2)
ggplot(pca_data, aes(x = Dim.1, y = Dim.2, color = factor(kmeans_model$cluster))) +
  geom_point() +
  ggtitle("K-Means Clustering (PCA-reduced Data)")
```



6- Réalisation de la Régression Linéaire Multivariée

Nous allons également appliquer une régression linéaire multivariée pour prédire le **SalePrice** en fonction des variables sélectionnées. Cela nous permettra d'examiner la relation entre ces variables et le prix de vente des maisons.

```
lm_model <- lm(SalePrice ~ Gr.Liv.Area + Overall.Qual + Garage.Cars + TotRms.AbvGrd, data
= selected_variables)


# Résumé du modèle
summary(lm_model)
```

```
Call:
lm(formula = SalePrice ~ Gr.Liv.Area + Overall.Qual + Garage.Cars +
    TotRms.AbvGrd, data = selected_variables)

Residuals:
    Min      1Q  Median      3Q     Max
-374218  -22357   -1664   19425  290389

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -90932.931   4144.092 -21.943  < 2e-16 ***
Gr.Liv.Area       63.823      2.826  22.580  < 2e-16 ***
Overall.Qual   27642.062    712.075  38.819  < 2e-16 ***
Garage.Cars    19678.564   1230.292  15.995  < 2e-16 ***
TotRms.AbvGrd  -4226.445    798.249  -5.295 1.28e-07 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 39510 on 2925 degrees of freedom
Multiple R-squared:  0.7558,    Adjusted R-squared:  0.7555
F-statistic:  2263 on 4 and 2925 DF,  p-value: < 2.2e-16
```

7- Interprétation des Résultats

PCA :

- Nous avons réduit la dimensionnalité des données tout en préservant la variance avec les **deux premières composantes principales**. Cela nous permet de mieux comprendre la structure des données sans perdre trop d'informations.
- La visualisation des variables nous aide à identifier les variables qui contribuent le plus à chaque composante principale.

K-Means :

- L'algorithme de **K-Means** a segmenté les maisons en **trois clusters** en fonction de leurs caractéristiques principales (surface habitable, qualité, nombre de chambres, etc.). La visualisation des clusters montre comment ces maisons sont réparties dans l'espace des deux premières composantes principales.

Régression Linéaire :

- La régression linéaire nous permet d'analyser l'influence des variables sélectionnées sur le **prix de vente**. Nous pouvons examiner les coefficients pour comprendre l'impact de chaque variable sur le prix des maisons.

8- Conclusion :

Nous avons utilisé différentes méthodes d'analyse multivariée (PCA, K-means, et régression linéaire) pour explorer les relations entre plusieurs variables dans le jeu de données **Ames Housing**. Ces méthodes nous ont permis de réduire la complexité des données, de segmenter les maisons en clusters similaires et de modéliser la relation entre les caractéristiques des maisons et leur prix de vente.

Conclusion générale du projet d'analyse du dataset Ames Housing :

Ce projet a permis d'explorer en profondeur le jeu de données **Ames Housing**, un dataset riche et complet utilisé pour l'analyse immobilière. Voici les grandes lignes des analyses menées et ce qu'elles nous ont permis de conclure :

✓ 1. Analyse exploratoire et préparation des données :

Nous avons commencé par explorer les variables qualitatives et quantitatives du dataset. Le nettoyage des données (remplacement des valeurs manquantes, sélection de variables pertinentes) était une étape essentielle avant toute analyse.

✓ 2. Analyse des correspondances (AFC) :

En étudiant deux variables qualitatives telles que **GarageFinish** (finition du garage) et **GarageType** (type de garage), l'AFC nous a permis de :

- Visualiser les associations entre modalités.
- Identifier une **dépendance significative** entre ces deux variables (test du chi^2).
- Réduire la dimension des données qualitatives pour mieux comprendre leur structure.

✓ 3. Analyse en composantes principales (ACP) :

L'ACP a été utilisée pour :

- Réduire la dimension des données numériques.
- Visualiser les relations entre variables.
- Identifier les axes principaux expliquant la plus grande part de la variance (Surface habitable, Qualité globale, Prix de vente…).
- Préparer les données pour les méthodes de regroupement (clustering).

✓ 4. Clustering (K-Means) :

Nous avons regroupé les maisons en **clusters homogènes** selon leurs caractéristiques :

- Trois groupes ont été identifiés, représentant différentes gammes de maisons (petites économiques, moyennes standards, grandes haut de gamme).
- Cette classification peut être utile pour la **segmentation de marché** ou la **recommandation de biens**.

✅ 5. Régression KNN et linéaire :

- La **régression KNN** a été utilisée pour prédire le **prix de vente** à partir des caractéristiques proches, mais avec un **MSE élevé**, ce qui montre ses limites ici.
- La **régression linéaire multivariée** a quant à elle permis d'identifier les **principaux prédicteurs du prix de vente**, comme :
  - la **surface habitable (Gr.Liv.Area)**,
  - la **qualité générale (OverallQual)**,
  - le **nombre de garages (GarageCars)**.

Conclusion finale :

Ce projet a démontré l'intérêt de combiner plusieurs approches statistiques et d'apprentissage automatique pour comprendre un dataset complexe comme **Ames Housing**. Grâce à des méthodes comme l'AFC, l'ACP, le clustering et la régression, nous avons :

- Mieux compris la structure du marché immobilier local.
- Identifié les variables influentes sur le prix des maisons.
- Proposé des outils de segmentation et de prédiction réutilisables pour d'autres jeux de données similaires.

Ce type d'analyse peut être très utile pour les agences immobilières, les investisseurs ou les data analysts travaillant sur l'évaluation et la recommandation de biens.