

La **régression logistique** : peut également être appliquée pour prédire l'attrition des employés dans une entreprise. L'objectif est de déterminer si un employé va quitter l'entreprise ou non, en fonction de diverses caractéristiques telles que la satisfaction au travail, le salaire, et le nombre d'années passées dans l'entreprise.

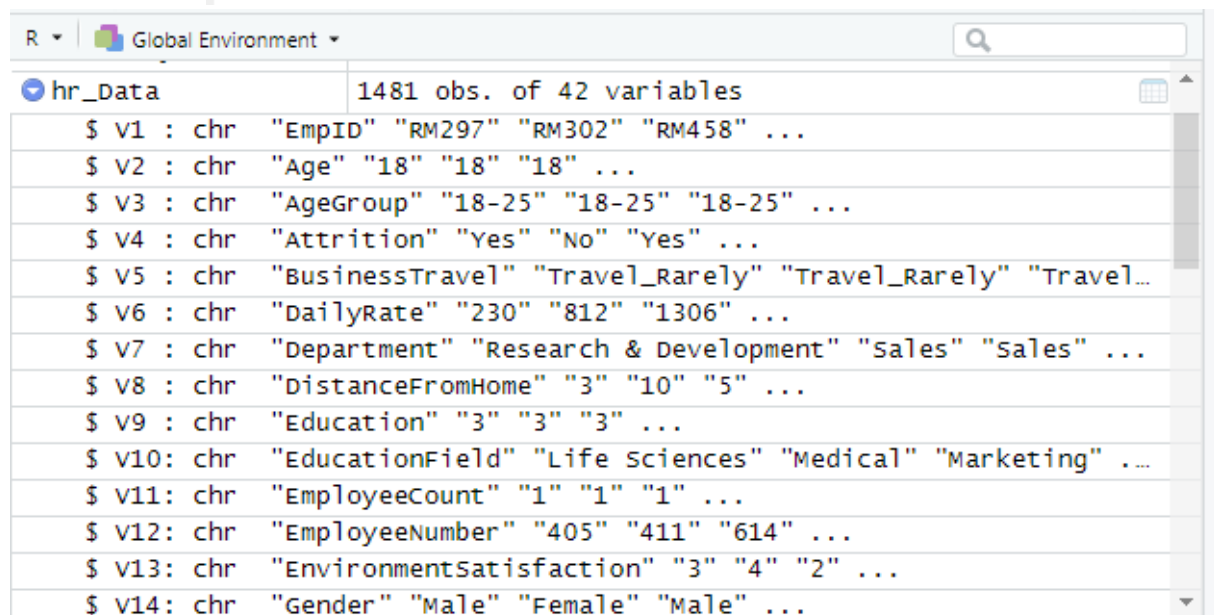
Objectif du Projet :

Construire un modèle de régression logistique pour prédire la probabilité qu'un employé quitte l'entreprise (attrition) en utilisant des données RH.

Étapes du Projet :

- 1- **Chargement des données** : Chargez les données dans R.
 - Importation des bibliothèques nécessaires.
 - Chargement du dataset.

```
# Chargement des packages :  
library(tidyverse)  
library(caret)  
library(dplyr)  
  
#Étape 1: charger les données  
hr_Data<- HR_Analytics
```



R	Global Environment	1481 obs. of 42 variables
hr_Data		
\$ v1 :	chr	"EmpID" "RM297" "RM302" "RM458" ...
\$ v2 :	chr	"Age" "18" "18" "18" ...
\$ v3 :	chr	"AgeGroup" "18-25" "18-25" "18-25" ...
\$ v4 :	chr	"Attrition" "Yes" "No" "Yes" ...
\$ v5 :	chr	"BusinessTravel" "Travel_Rarely" "Travel_Rarely" "Travel..."
\$ v6 :	chr	"DailyRate" "230" "812" "1306" ...
\$ v7 :	chr	"Department" "Research & Development" "Sales" "Sales" ...
\$ v8 :	chr	"DistanceFromHome" "3" "10" "5" ...
\$ v9 :	chr	"Education" "3" "3" "3" ...
\$ v10 :	chr	"EducationField" "Life Sciences" "Medical" "Marketing" ...
\$ v11 :	chr	"EmployeeCount" "1" "1" "1" ...
\$ v12 :	chr	"EmployeeNumber" "405" "411" "614" ...
\$ v13 :	chr	"EnvironmentSatisfaction" "3" "4" "2" ...
\$ v14 :	chr	"Gender" "Male" "Female" "Male" ...

R	Global Environment	
\$ V15:	chr	"HourlyRate" "54" "69" "69" ...
\$ V16:	chr	"JobInvolvement" "3" "2" "3" ...
\$ V17:	chr	"JobLevel" "1" "1" "1" ...
\$ V18:	chr	"JobRole" "Laboratory Technician" "Sales Representative"...
\$ V19:	chr	"Jobsatisfaction" "3" "3" "2" ...
\$ V20:	chr	"Maritalstatus" "single" "single" "single" ...
\$ V21:	chr	"MonthlyIncome" "1420" "1200" "1878" ...
\$ V22:	chr	"Salaryslab" "Upto 5k" "Upto 5k" "Upto 5k" ...
\$ V23:	chr	"MonthlyRate" "25233" "9724" "8059" ...
\$ V24:	chr	"NumCompaniesworked" "1" "1" "1" ...
\$ V25:	chr	"over18" "Y" "Y" "Y" ...
\$ V26:	chr	"OverTime" "No" "No" "Yes" ...
\$ V27:	chr	"Percentsalaryhike" "13" "12" "14" ...
\$ V28:	chr	"PerformanceRating" "3" "3" "3" ...
\$ V29:	chr	"Relationshipsatisfaction" "3" "1" "4" ...
\$ V30:	chr	"StandardHours" "80" "80" "80" ...
\$ V31:	chr	"StockoptionLevel" "0" "0" "0" ...
\$ V32:	chr	"TotalworkingYears" "0" "0" "0" ...
\$ V33:	chr	"TrainingTimesLastYear" "2" "2" "3" ...
\$ V34:	chr	"workLifeBalance" "3" "3" "3" ...
\$ V35:	chr	"YearsAtCompany" "0" "0" "0" ...
\$ V36:	chr	"YearsInCurrentRole" "0" "0" "0" ...
\$ V37:	chr	"YearsSinceLastPromotion" "0" "0" "0" ...
\$ V38:	chr	"YearswithCurrManager" "0" "0" "0" ...

- 2- Préparation des Données : L'étape de préparation des données est cruciale dans tout projet d'analyse de données ou de modélisation.

#Étape 2: Préparation des données

```
hr_Data <- hr_Data %>%  
  mutate(Attrition = factor(V4, levels = c('No', 'Yes')),  
         Age = factor(V2),  
         Department = factor(V7),  
         Education = factor(V9),  
         EducationField = factor(V10),  
         Gender = factor(V14),  
         Maritalstatus = factor(V20),  
         PerformanceRating = factor(V28),  
         OverTime = factor(V26),  
         Salaryslab = factor(V22),  
         Percentsalaryhike = factor(V27),  
         MonthlyRate = factor(V23),  
         JobRole = factor(V18),  
         Jobsatisfaction = factor(V19),  
         Environmentsatisfaction = factor(V13),  
         Relationshipsatisfaction = factor(V29),  
         workLifeBalance = factor(V34),  
         DistanceFromHome = factor(V8),  
         BusinessTravel = factor(V5)  
  ) %>%  
  drop_na()
```

- Ici, nous utilisons l'opérateur `%>%` du package « dplyr » pour appliquer une série de transformations à cet ensemble de données.
- « `mutate()` » est une fonction de « dplyr » qui permet de créer ou de transformer des colonnes dans un data frame.
- **Transformation en facteurs** : Les variables sont transformées en facteurs. Les facteurs sont utilisés en R pour les variables catégorielles, ce qui est nécessaire pour les modèles de régression logistique.
- « `drop_na()` » est une fonction de « dplyr » qui permet de supprimer les lignes contenant des valeurs manquantes (NA) dans le data frame. Ceci est important car les modèles de machine learning ne peuvent pas gérer les valeurs manquantes et cela pourrait entraîner des erreurs ou des biais dans le modèle.

Importance de cette étape :

- **Préparation pour la modélisation** : Transformer les variables en facteurs est crucial pour que les algorithmes de modélisation puissent les interpréter correctement.
- **Nettoyage des données** : La suppression des valeurs manquantes garantit que les données sont propres et prêtes pour l'analyse, évitant ainsi les erreurs pendant la modélisation.
- **Standardisation** : Assurer que toutes les variables sont dans un format compatible pour l'analyse et la modélisation permet d'obtenir des résultats fiables et comparables.

3- Exploration des Données :

Utiliser des résumés statistiques et des visualisations aide à mieux comprendre les données et à prendre des décisions plus éclairées pour les étapes suivantes du projet.

```
# Étape 2 : Exploration des Données
summary(hr_Data)
```

- La fonction « `summary()` » en R fournit un résumé statistique de chaque colonne de l'ensemble de données.
- Pour les variables catégorielles (facteurs) : La fonction « `summary()` » en R fournit un résumé sur la distribution des niveaux, c'est-à-dire le nombre de fois que chaque niveau apparaît.

```

V41              V42              Attrition      Age
Length:1480      Length:1480      No :1242    34      : 78
Class :character  Class :character  Yes: 238    35      : 78
Mode :character   Mode :character   31      : 70
                  36      : 70
                  29      : 69
                  32      : 61
                  (Other):1054

      Department      Education      EducationField      Gender
Department      : 0  1      :172  EducationField : 0  Female:591
Human Resources  : 63 2      :283  Human Resources : 27 Gender: 0
Research & Development:967 3      :578  Life Sciences   :607 Male :889
Sales            :450 4      :399  Marketing        :161
                  5      : 48  Medical          :470
                  Education: 0  other                : 83
                  Technical Degree:132

      MaritalStatus      PerformanceRating      OverTime      SalarySlab
Divorced      :328  3      :1253  No      :1062  10k-15k      :150
MaritalStatus: 0  4      : 227  OverTime: 0  15k+        :133
Married       :679  PerformanceRating: 0  Yes      : 418  5k-10k       :444
Single        :473                                     salaryslab: 0
                                                         upto 5k      :753

PercentSalaryHike  MonthlyRate      JobRole
11      :211      4223      : 3  Sales Executive      :329
13      :210      9150      : 3  Research Scientist    :293
14      :203      10228     : 2  Laboratory Technician :261
12      :199      10494     : 2  Manufacturing Director :147
15      :102      11162     : 2  Healthcare Representative:132
18      : 90      11591     : 2  Manager               :102
(Other):465      (Other):1466  (Other)              :216

```

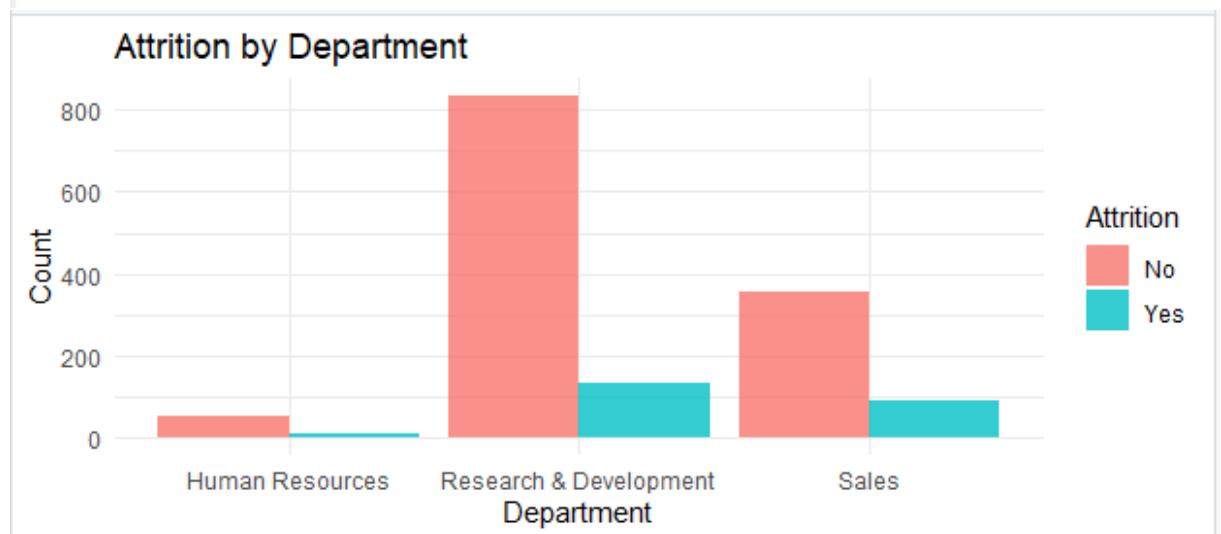
- **Attrition** : La variable Attrition montre qu'il y a 1242 employés sans attrition (No) et 238 avec attrition (Yes).
- **Age** : La distribution des âges montre combien d'employés sont dans chaque catégorie d'âge.
- **Department** : Montre la distribution des employés dans les départements Sales, R&D, et HR.
- **Education** : Montre la distribution des niveaux d'éducation.
- **Gender** : Montre la distribution des genres (Male et Female).
- **MaritalStatus** : Montre la distribution des statuts maritaux.
- **PerformanceRating** : Montre la distribution des évaluations de performance.
- **OverTime** : Montre la distribution des employés travaillant des heures supplémentaires ou non.
- **SalarySlab** : Montre la distribution des tranches de salaire.
- **PercentSalaryHike** : Montre la distribution des pourcentages d'augmentation de salaire.
- **MonthlyRate** : Montre la distribution des taux mensuels.
- **JobRole** : Montre la distribution des rôles de l'emploi.

4- Visualisation des Données :

Ces visualisations permettent de détecter des tendances et des schémas dans les données d'attrition, en fournissant des insights précieux pour comprendre les facteurs qui peuvent influencer les employés à quitter l'entreprise.

Visualisation de l'attrition par Département :

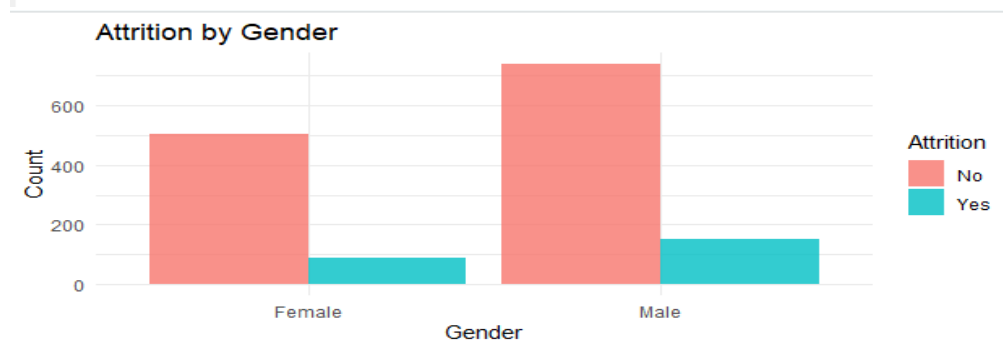
```
# visualisation de l'attrition par département
ggplot(data = hr_Data, aes(x = Department, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "Attrition by Department", x = "Department", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise (attrition) ou restent, répartis par département.
- Il permet d'identifier si certains départements ont des taux d'attrition plus élevés que d'autres.

Visualisation de l'attrition par Genre :

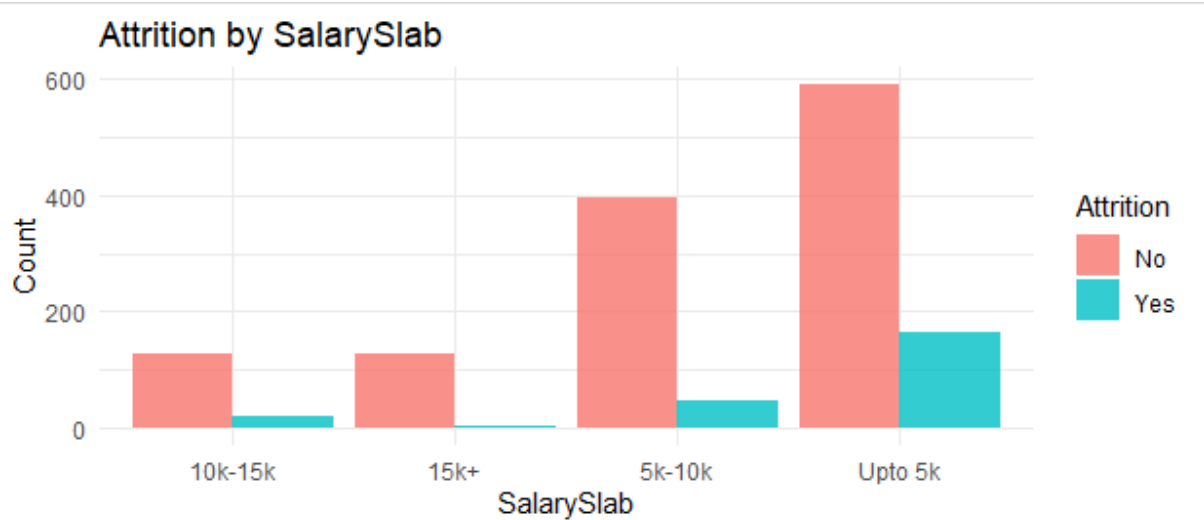
```
# visualisation de l'attrition par Gender
ggplot(data = hr_Data, aes(x = Gender, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "Attrition by Gender", x = "Gender", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par genre.
- Il permet d'explorer les différences potentielles dans les taux d'attrition entre les hommes et les femmes.

Visualisation de l'attrition par Tranche de Salaire :

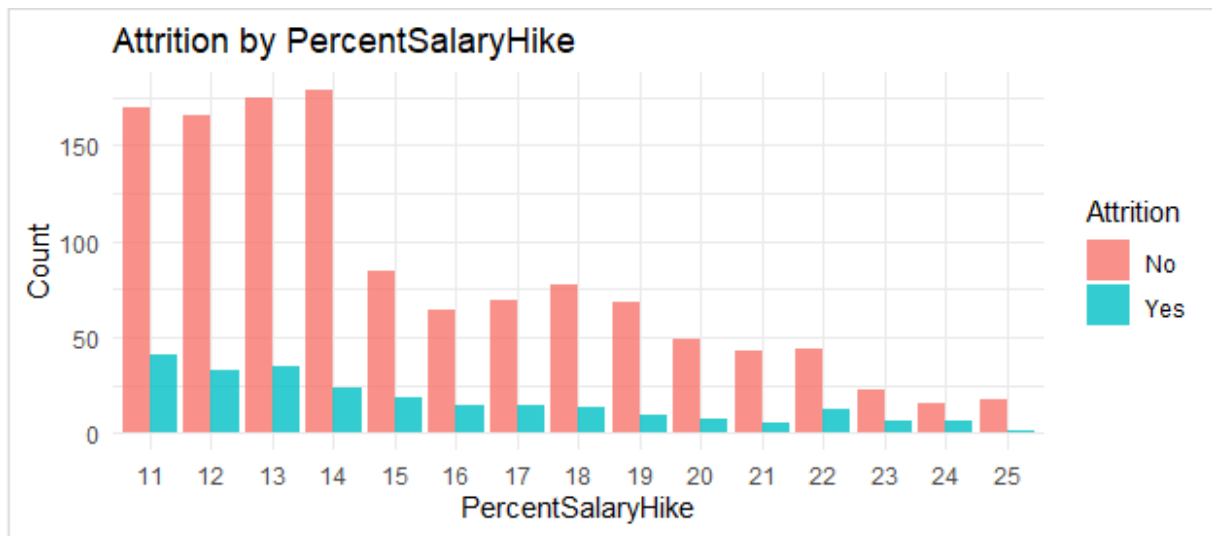
```
# Visualisation de l'attrition par SalarySlab
ggplot(data = hr_Data, aes(x = SalarySlab, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "Attrition by SalarySlab", x = "SalarySlab", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par tranche de salaire.
- Il permet de voir si les employés dans certaines tranches de salaire ont des taux d'attrition plus élevés.

Visualisation de l'attrition par Pourcentage d'Augmentation de Salaire :

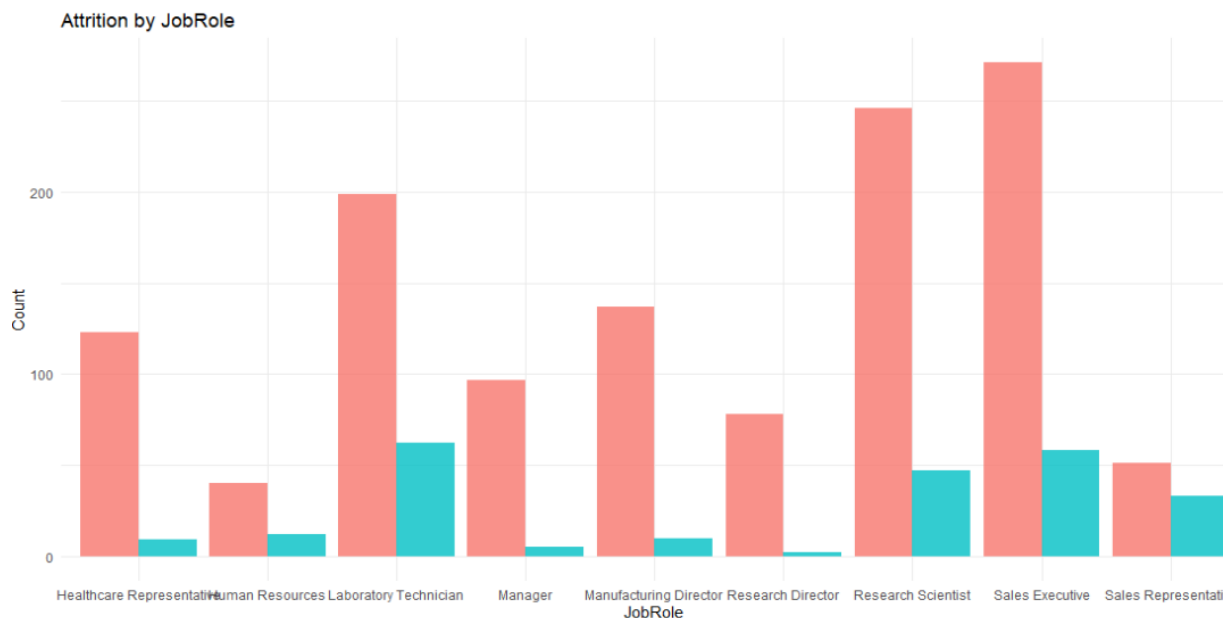
```
# Visualisation de l'attrition par PercentsalaryHike
ggplot(data = hr_Data, aes(x = PercentsalaryHike, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "Attrition by PercentsalaryHike", x = "PercentsalaryHike",
        y = "Count") + theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par pourcentage d'augmentation de salaire.
- Il permet d'explorer si les augmentations de salaire influencent l'attrition.

Visualisation de l'attrition par Rôle de l'Emploi :

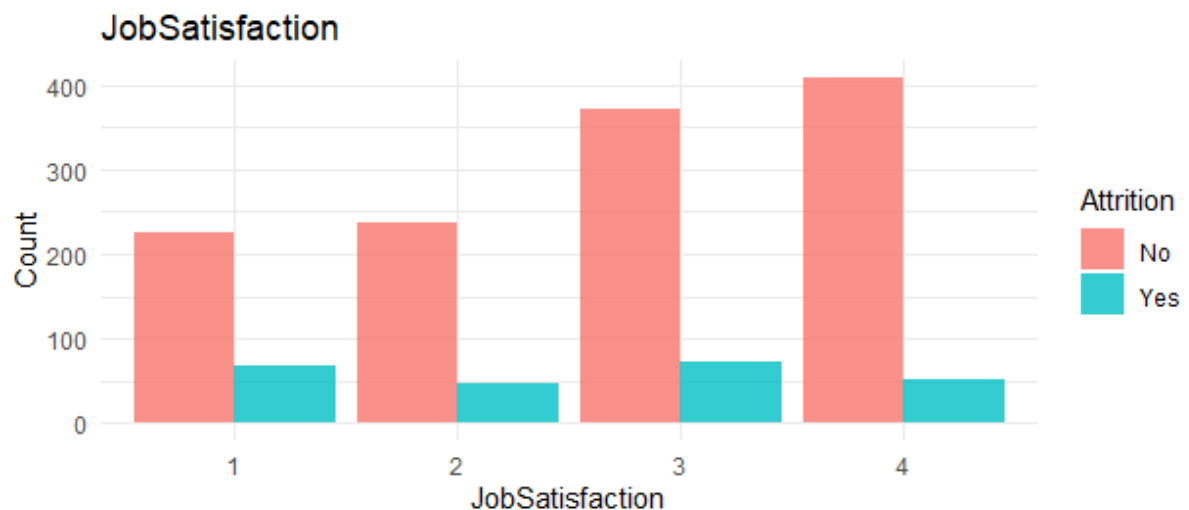
```
# Visualisation de l'attrition par JobRole
ggplot(data = hr_Data, aes(x = JobRole, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "Attrition by JobRole", x = "JobRole", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par rôle de l'emploi.
- Il aide à identifier si certains rôles de l'emploi ont des taux d'attrition plus élevés.

Visualisation de l'attrition par Satisfaction au Travail :

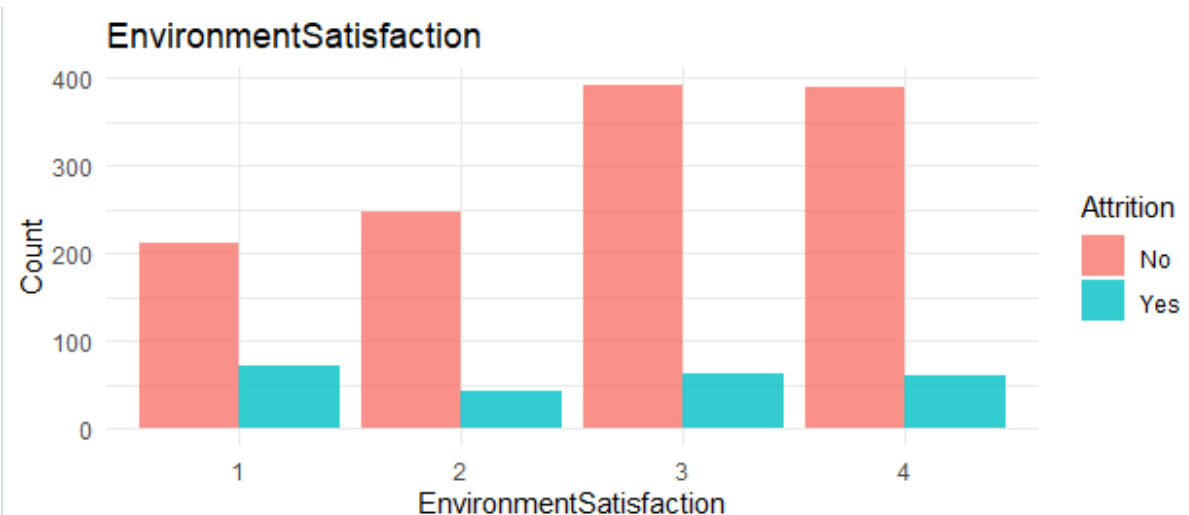
```
# visualisation de l'attrition par JobSatisfaction
ggplot(data = hr_Data, aes(x = JobSatisfaction, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "JobSatisfaction", x = "JobSatisfaction", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par niveau de satisfaction au travail.
- Il permet de voir si la satisfaction au travail influence l'attrition.

Visualisation de l'attrition par Satisfaction de l'Environnement :

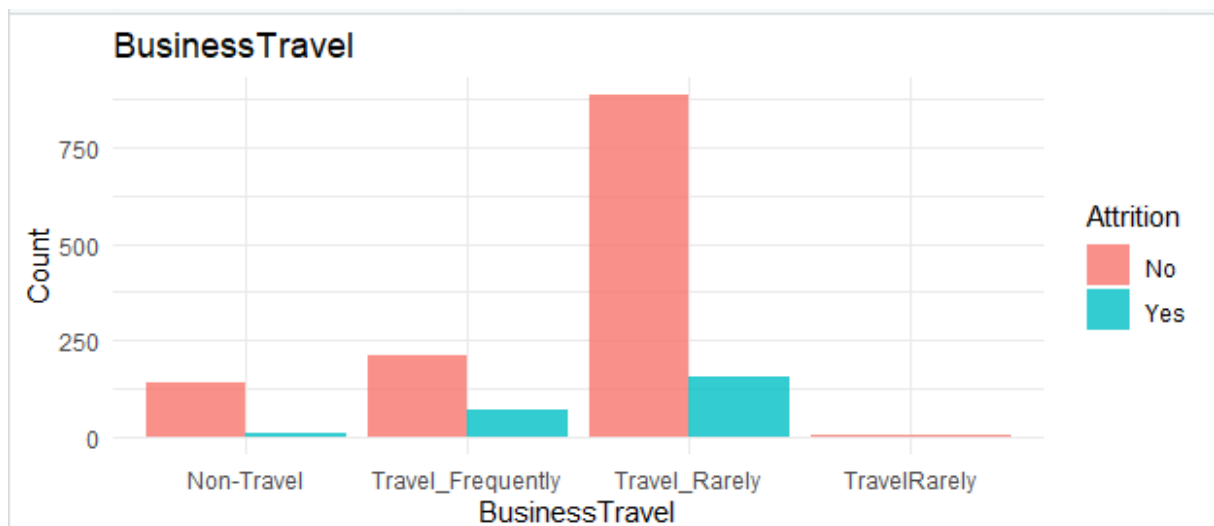
```
# visualisation de l'attrition par EnvironmentSatisfaction
ggplot(data = hr_Data, aes(x = EnvironmentSatisfaction, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "EnvironmentSatisfaction", x = "EnvironmentSatisfaction",
        y = "Count") + theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par niveau de satisfaction de l'environnement.
- Il aide à comprendre si la satisfaction de l'environnement de travail joue un rôle dans l'attrition.

Visualisation de l'attrition par Fréquence de Voyage d'Affaires :

```
# visualisation de l'attrition par BusinessTravel
ggplot(data = hr_Data, aes(x = BusinessTravel, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "BusinessTravel", x = "BusinessTravel", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par fréquence de voyage d'affaires.
- Il permet de voir si les voyages fréquents influencent l'attrition.

Visualisation de l'attrition par Équilibre Vie-Travail :

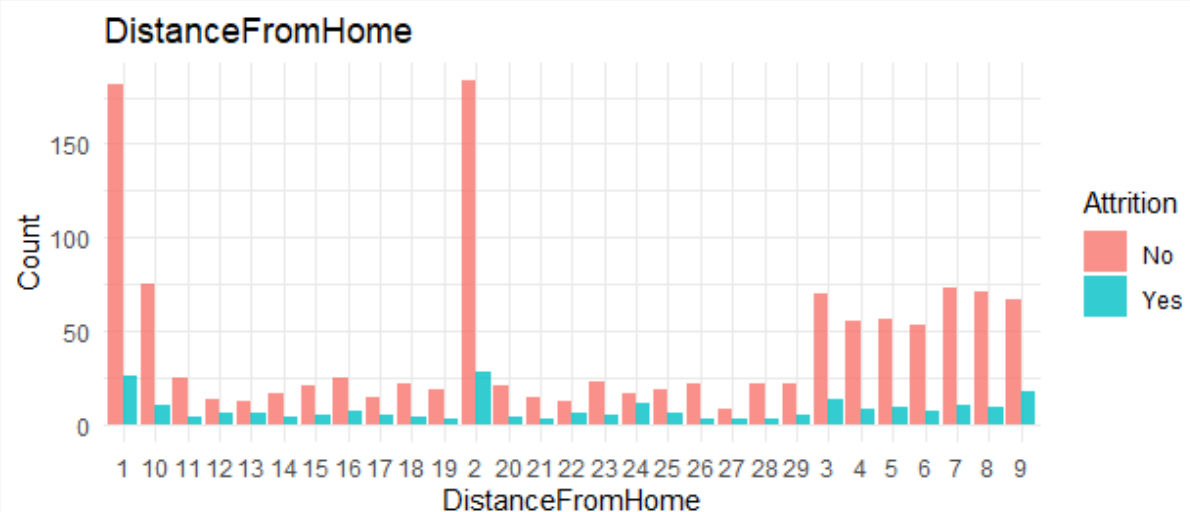
```
# visualisation de l'attrition par workLifeBalance
ggplot(data = hr_Data, aes(x = workLifeBalance, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "workLifeBalance", x = "workLifeBalance", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par niveau d'équilibre vie-travail.
- Il aide à identifier si un bon équilibre vie-travail réduit l'attrition.

Visualisation de l'attrition par Distance du Domicile :

```
# visualisation de l'attrition par DistanceFromHome
ggplot(data = hr_Data, aes(x = DistanceFromHome, fill = Attrition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(title = "DistanceFromHome", x = "DistanceFromHome", y = "Count") +
  theme_minimal()
```



- Ce graphique à barres montre le nombre d'employés qui quittent l'entreprise ou restent, répartis par distance entre le domicile et le lieu de travail.
- Il permet d'explorer si la distance du domicile influence les taux d'attrition.

5- Séparation des Données :

Diviser les données en ensembles d'entraînement et de test est crucial pour l'évaluation de la performance des modèles de machine learning.

```
# Etape 5 : Diviser les données en ensembles d'entraînement et de test
set.seed(123)
train_index <- createDataPartition(hr_Data$Attrition, p = 0.7, list = FALSE)
train_data <- hr_Data[train_index, ]
test_data <- hr_Data[-train_index, ]
```

- En fixant la graine aléatoire avec « `set.seed(123)` », vous vous assurez que chaque fois que vous exécutez ce code, les données seront divisées de la même manière.
- « `createDataPartition` » est une fonction du package `caret` qui crée une partition des données tout en maintenant la proportion des classes dans la variable cible `Attrition`
- `p = 0.7` : La proportion des données qui iront dans l'ensemble d'entraînement (70%).
- Utiliser l'indice généré pour extraire les lignes correspondantes de `hr_Data` et créer l'ensemble d'entraînement.
- L'ensemble d'entraînement est utilisé pour ajuster (entraîner) le modèle.
- Utiliser les indices restants pour créer l'ensemble de test.
- L'ensemble de test est utilisé pour évaluer les performances du modèle après l'entraînement, en s'assurant que le modèle généralise bien aux nouvelles données.

6- Créer le modèle de régression logistique :

Cette étape consiste à créer un modèle de régression logistique pour prédire la probabilité de l'attrition des employés en fonction de plusieurs variables explicatives (prédicteurs).

```
# Etape 6 : Créer le modèle de régression logistique
model <- glm(Attrition ~ Department + Education + EducationField +
             Gender + MaritalStatus + PerformanceRating + OverTime +
             SalarySlab + PercentsalaryHike + JobSatisfaction +
             JobRole + Environmentsatisfaction + Relationshipsatisfaction +
             workLifeBalance + DistanceFromHome + BusinessTravel,
             data = train_data, family = binomial)
```

- `glm (Generalized Linear Model)` : Une fonction en R pour ajuster des modèles linéaires généralisés, dont la régression logistique.
- **Formule** : `Attrition ~ ...` : Spécifie que nous souhaitons modéliser `Attrition` (variable cible) en fonction des autres variables (`Department`, `Education`, `JobSatisfaction`, etc.).

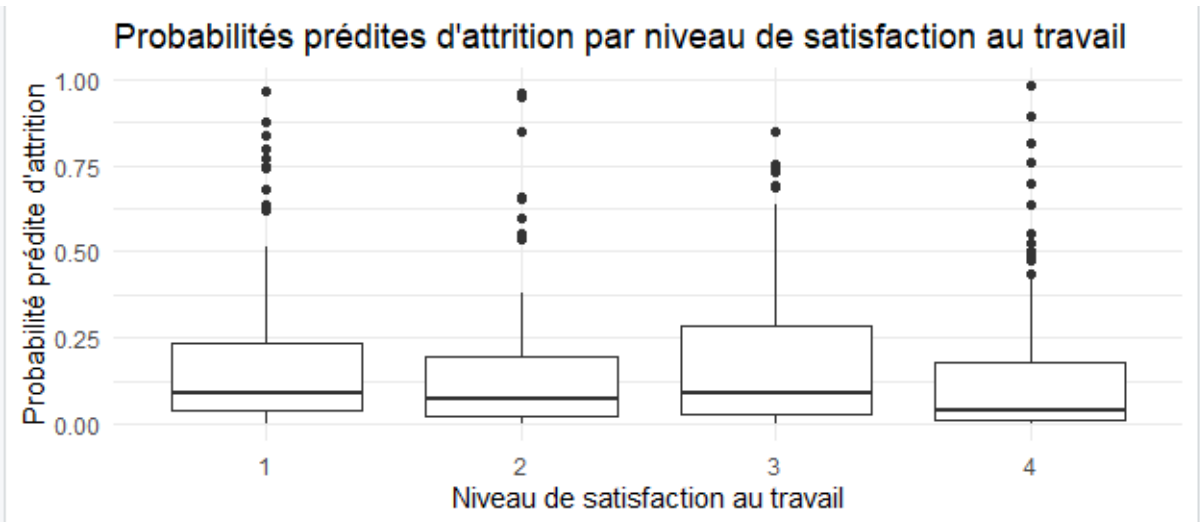
```
# Résumé du modèle
summary(model)
```

- Certaines variables, comme `MaritalStatusSingle`, `OverTimeYes`, `JobSatisfaction4`, et `EnvironmentSatisfaction4`, sont significatives et ont une forte influence sur l'attrition.

Tracer une visualisation simple des probabilités prédites d'attrition en fonction de la satisfaction au travail :

```
# Prédire les probabilités d'attrition pour les données de test
test_data$predicted_prob <- predict(model, newdata = test_data,
                                   type = "response")

# Tracer les probabilités prédites d'attrition par niveau de satisfaction au tra
library(ggplot2)
ggplot(test_data, aes(x = factor(JobSatisfaction), y = predicted_prob)) +
  geom_boxplot() +
  labs(title = "Probabilités prédites d'attrition par niveau de satisfaction au tra",
       x = "Niveau de satisfaction au travail",
       y = "Probabilité prédite d'attrition") +
  theme_minimal()
```



Ce graphique à boîte (boxplot) des probabilités prédites d'attrition pour chaque niveau de satisfaction au travail, offrant une visualisation claire des tendances et des distributions.

7- Validation et Évaluation du Modèle :

```
# Étape 7 : validation et évaluation du modèle

# Prédiction
# Prédiction sur les données de test
predicted <- predict(model, newdata = test_data, type = "response")
predicted
# Créer predicted_class
predicted_class <- ifelse(predicted > 0.5, "Yes", "No")
predicted_class
# Comparer avec les valeurs réelles
actual <- as.character(test_data$Attrition)
actual

# Créer une confusion matrix
confusion_matrix <- confusionMatrix(data = factor(predicted_class,
                                                  levels = c("Yes", "No")),
                                   confusion_matrix)
```

La matrice de confusion et les statistiques associées donnent des informations cruciales sur la performance du modèle de régression logistique.

```
> confusion_matrix
Confusion Matrix and Statistics

              Reference
Prediction Yes  No
Yes         23  22
No          48 350

              Accuracy : 0.842
              95% CI : (0.8046, 0.8747)
No Information Rate : 0.8397
P-Value [Acc > NIR] : 0.480023

              Kappa : 0.3109

McNemar's Test P-Value : 0.002807

              Sensitivity : 0.32394
              Specificity : 0.94086
              Pos Pred Value : 0.51111
              Neg Pred Value : 0.87940
              Prevalence : 0.16027
              Detection Rate : 0.05192
              Detection Prevalence : 0.10158
              Balanced Accuracy : 0.63240

              'Positive' Class : Yes
```

- Vrai Positifs (TP): 23 (le modèle a correctement prédit "Yes" pour 23 cas)
- Faux Positifs (FP): 22 (le modèle a prédit "Yes" à tort pour 22 cas)
- Faux Négatifs (FN): 48 (le modèle a prédit "No" à tort pour 48 cas)
- Vrai Négatifs (TN): 350 (le modèle a correctement prédit "No" pour 350 cas)
- **Accuracy (Exactitude): 0.842**
 - La proportion de prédictions correctes ($(TP + TN) / (TP + FP + FN + TN)$).
 - Dans ce cas, le modèle a une exactitude de 84.2%.
- **P-Value [Acc > NIR]: 0.480023**
 - La p-value pour le test que l'exactitude du modèle est supérieure au taux d'information. Une p-value élevée (>0.05) indique que le modèle n'est pas significativement meilleur que le choix aléatoire basé sur la classe majoritaire.
- **Sensitivity (Sensibilité): 0.32394**
 - La proportion de vrais positifs correctement identifiés ($TP / (TP + FN)$). Ici, 32.39% des "Yes" sont correctement prédits.
- **Specificity (Spécificité): 0.94086**
 - La proportion de vrais négatifs correctement identifiés ($TN / (TN + FP)$). Ici, 94.09% des "No" sont correctement prédits.

- **Pos Pred Value (Valeur Prédictive Positive): 0.51111**
 - La proportion des prédictions "Yes" qui sont correctes ($TP / (TP + FP)$).
- **Neg Pred Value (Valeur Prédictive Négative): 0.87940**
 - La proportion des prédictions "No" qui sont correctes ($TN / (TN + FN)$).

8- Tracer la courbe ROC :

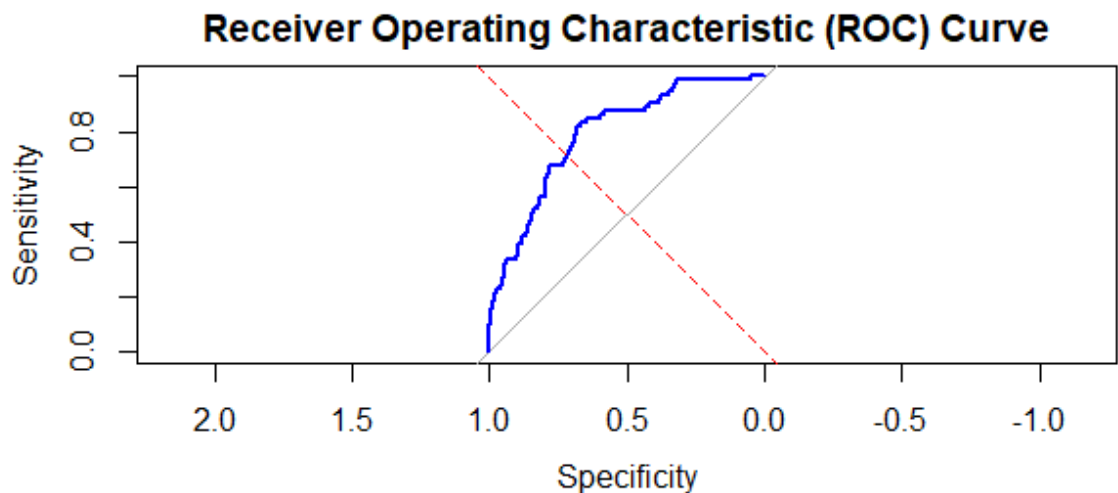
Il n'est pas possible de tracer directement le modèle de régression logistique comme on le ferait pour une simple régression linéaire. Cependant, nous pouvons tracer la courbe ROC (Receiver Operating Characteristic) pour visualiser la performance du modèle. La courbe ROC montre la sensibilité (taux de vrais positifs) par rapport à la spécificité (taux de faux positifs) pour différents seuils de probabilité.

```
# Tracer la courbe ROC
library(pROC)

# Prédiction sur l'ensemble de test
predicted <- predict(model, newdata = test_data, type = "response")

# Tracer la courbe ROC
roc_curve <- roc(test_data$Attrition, predicted)
plot(roc_curve, main = "Receiver Operating Characteristic (ROC) Curve",
     col = "blue")

# Ajouter la ligne de référence (chance)
abline(a=0, b=1, col="red", lty=2)
```



- ⇒ Bien que le modèle ait une bonne spécificité, sa sensibilité est relativement faible, ce qui pourrait signifier qu'il manque de nombreux cas d'attrition.

9- Amélioration du Modèle :

```
# Étape 9 : Amélioration du Modèle
# Sélection des variables significatives
model2 <- step(model)

summary(model2)

# Réévaluation du modèle ajusté
predictions2 <- predict(model2, testData, type = 'response')
predicted_classes2 <- ifelse(predictions2 > 0.5, 'Yes', 'No')
confusionMatrix(factor(predicted_classes2), testData$Attrition)
```

Dans cette étape, vous utilisez la fonction « step » pour effectuer une sélection automatique des variables significatives dans votre modèle de régression logistique. Cette fonctionnalité vous permet de comparer différents modèles en ajoutant ou en retirant des variables de manière itérative, en utilisant des critères comme l'AIC pour choisir le meilleur modèle.

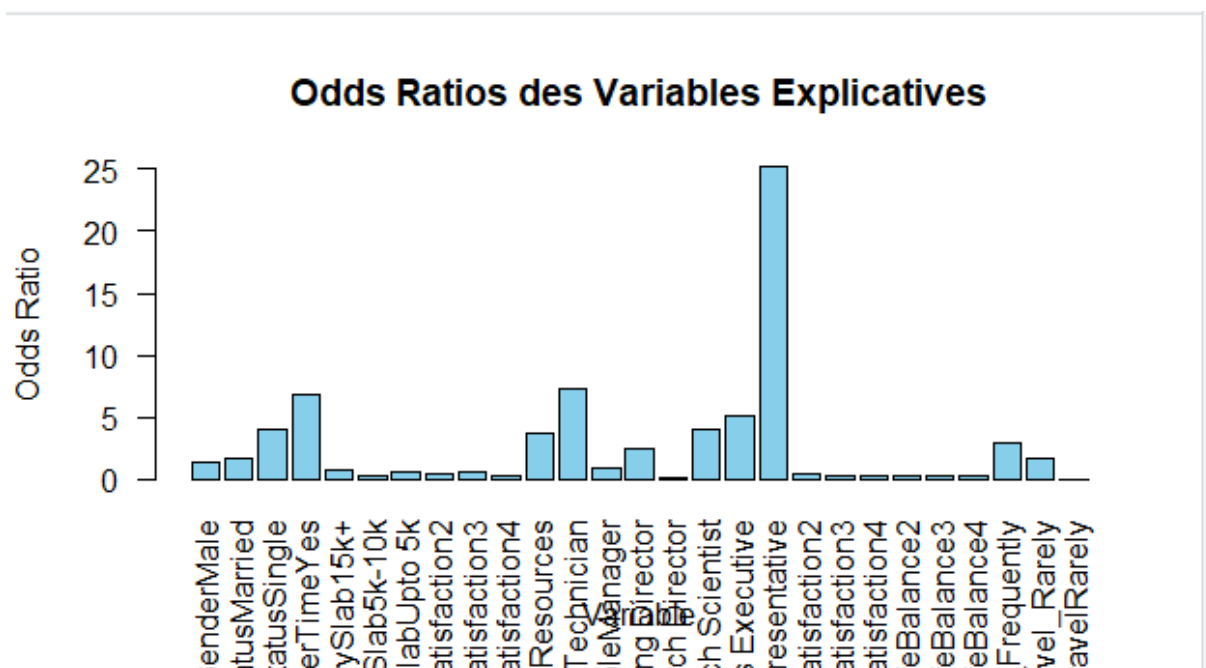
```
# Interprétation des Résultats
exp(coef(model2)) # Odds Ratios
```

- **Intercept:** L'intercept est la valeur de l'odds ratio lorsque toutes les autres variables sont maintenues constantes. Ici, l'odds ratio est proche de zéro, ce qui signifie que le groupe de référence a une probabilité très faible d'attrition.
- **GenderMale:** Les employés masculins ont environ 1,43 fois plus de chances d'attrition que les employés féminins, toutes choses étant égales par ailleurs.
- **MaritalStatusMarried:** Les employés mariés ont environ 1,77 fois plus de chances d'attrition que les employés célibataires.
- **MaritalStatusSingle:** Les employés célibataires ont environ 4 fois plus de chances d'attrition que les employés célibataires.
- **OverTimeYes:** Les employés faisant des heures supplémentaires ont environ 6,85 fois plus de chances d'attrition que ceux qui n'en font pas.
- **SalarySlab15k+:** Les employés gagnant plus de 15 000 unités monétaires ont environ 0,79 fois moins de chances d'attrition que ceux gagnant moins de 5 000 unités monétaires.

⇒ Ces rapports de cotes vous aident à comprendre l'impact de chaque variable sur la probabilité d'attrition des employés.

10- Tracé des Rapports de Cotes des Variables Explicatives :

```
# Étape 10 : Tracer le graphique des rapports de cotes  
barplot(odds_ratios, las=2, col="skyblue", main="Odds Ratios des variables  
Explicatives", ylab="Odds Ratio", xlab="Variable")
```



Ce graphique des rapports de cotes (odds ratios) des variables explicatives est important car il permet de visualiser l'impact de chaque variable sur la probabilité d'attrition des employés. Les odds ratios indiquent comment un changement d'une unité dans la variable affecte les chances de l'attrition. Une valeur supérieure à 1 indique une augmentation des chances, tandis qu'une valeur inférieure à 1 indique une diminution des chances. Cela aide à identifier les variables les plus influentes dans le modèle et à comprendre leur effet sur la variable cible.

