

employee.csv

2024-04-19

R Markdown

Nos conectamos al dataset:

Columns:

Education: The educational qualifications of employees, including degree, institution, and field of study.

Joining Year: The year each employee joined the company, indicating their length of service.

City: The location or city where each employee is based or works.

Payment Tier: Categorization of employees into different salary tiers.

Age: The age of each employee, providing demographic insights.

Gender: Gender identity of employees, promoting diversity analysis.

Ever Benched: Indicates if an employee has ever been temporarily without assigned work.

Experience in Current Domain: The number of years of experience employees have in their current field.

Leave or Not: a target column

```
url <- "https://raw.githubusercontent.com/chelocoder/ProyectoR/main/Employee.csv"
datos <- read.csv(url)
head(datos)
```

```
##   Education JoiningYear      City PaymentTier Age Gender EverBenched
## 1 Bachelors      2017 Bangalore         3  34   Male           No
## 2 Bachelors      2013      Pune         1  28  Female           No
## 3 Bachelors      2014 New Delhi         3  38  Female           No
## 4  Masters      2016 Bangalore         3  27   Male           No
## 5  Masters      2017      Pune         3  24   Male           Yes
## 6 Bachelors      2016 Bangalore         3  22   Male           No
##   ExperienceInCurrentDomain LeaveOrNot
## 1                        0           0
## 2                        3           1
## 3                        2           0
## 4                        5           1
## 5                        2           1
## 6                        0           0
```

1. Estadísticos Descriptivos

Obtenemos los estadísticos descriptivos de todas las variables del dataset

```
# Obtener estadísticos descriptivos
summary_stats <- summary(datos)
```

```
# Mostrar estadísticos descriptivos
print(summary_stats)
```

```
##      Education      JoiningYear      City      PaymentTier
## Length:4653      Min.    :2012      Length:4653      Min.    :1.000
## Class :character  1st Qu.:2013      Class :character  1st Qu.:3.000
## Mode  :character  Median :2015      Mode  :character  Median :3.000
##                                     Mean  :2015      Mean  :2.698
##                                     3rd Qu.:2017      3rd Qu.:3.000
##                                     Max.   :2018      Max.   :3.000
##      Age      Gender      EverBenched
## Min.    :22.00      Length:4653      Length:4653
## 1st Qu.:26.00      Class :character      Class :character
## Median :28.00      Mode  :character      Mode  :character
## Mean    :29.39
## 3rd Qu.:32.00
## Max.    :41.00
## ExperienceInCurrentDomain      LeaveOrNot
## Min.    :0.000      Min.    :0.0000
## 1st Qu.:2.000      1st Qu.:0.0000
## Median :3.000      Median :0.0000
## Mean    :2.906      Mean    :0.3439
## 3rd Qu.:4.000      3rd Qu.:1.0000
## Max.    :7.000      Max.    :1.0000
```

1.1. Tendecia Central

1.1.1 Variables Numéricas

Obtenemos los Estadísticos descriptivos de solamente las variables numéricas

```
##      JoiningYear      PaymentTier      Age      ExperienceInCurrentDomain
## Min.    :2012      Min.    :1.000      Min.    :22.00      Min.    :0.000
## 1st Qu.:2013      1st Qu.:3.000      1st Qu.:26.00      1st Qu.:2.000
## Median :2015      Median :3.000      Median :28.00      Median :3.000
## Mean    :2015      Mean    :2.698      Mean    :29.39      Mean    :2.906
## 3rd Qu.:2017      3rd Qu.:3.000      3rd Qu.:32.00      3rd Qu.:4.000
## Max.    :2018      Max.    :3.000      Max.    :41.00      Max.    :7.000
##      LeaveOrNot
## Min.    :0.0000
## 1st Qu.:0.0000
## Median :0.0000
## Mean    :0.3439
## 3rd Qu.:1.0000
## Max.    :1.0000
```

Obtenemos los estadísticos descriptivos de cada variable:

a.Variable JoiningYear

a.1.Valor Mínimo:

```
# Obtener el valor mínimo
min <- min(datos$JoiningYear)

# Imprimir el valor mínimo de la columna
print(min)
```

```
## [1] 2012
```

a.2.Primer Cuartil:

```
# Obtener el primer cuartil
q1 <- quantile(datos$JoiningYear, 0.25)

# Imprimir el primer cuartil
print(q1)
```

```
## 25%
## 2013
```

a.3. Mediana:

```
# Calcular la mediana de la variable
mediana_joining_year <- median(datos$JoiningYear)

# Imprimir la mediana de la variable
print(mediana_joining_year)
```

```
## [1] 2015
```

a.4.Tercer Cuartil:

```
# Obtener el tercer cuartil
q3 <- quantile(datos$JoiningYear, 0.75)

# Imprimir el tercer cuartil
print(q3)
```

```
## 75%
## 2017
```

a.5.Valor Máximo:

```
# Calcular el valor máximo de la variable
max <- max(datos$JoiningYear)

# Imprimir el valor máximo de la variable
print(max)
```

```
## [1] 2018
```

a.6.Recorrido (Rango):

```
# Calcular el rango de la variable
rango <- max(datos$JoiningYear) - min(datos$JoiningYear)

# Imprimir el rango de la variable "JoiningYear"
print(rango)
```

```
## [1] 6
```

a.7.Modas:

```
# Calcular la moda de la variable
moda <- as.numeric(names(sort(-table(datos$JoiningYear))[1]))

# Imprimir la moda de la variable
print(moda)
```

```
## [1] 2017
```

a.8.Frecuencia:

```
# Calcular la frecuencia de la variable
frecuencia <- table(datos$JoiningYear)

# Imprimir la frecuencia de la variable
print(frecuencia)
```

```
##
## 2012 2013 2014 2015 2016 2017 2018
## 504 669 699 781 525 1108 367
```

```
#Imprimir la frecuencia máxima
frecuencia_maxima <- max(frecuencia)
print(frecuencia_maxima)
```

```
## [1] 1108
```

b.Variable PaymentTier

b.1.Valor Mínimo:

```
# Obtener el valor mínimo
min <- min(datos$PaymentTier)

# Imprimir el valor mínimo de la columna
print(min)
```

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

b.2.Primer Cuartil:

```
# Obtener el primer cuartil
q1 <- quantile(datos$PaymentTier, 0.25)

# Imprimir el primer cuartil
print(q1)
```

```
## 25%
## 3
```

b.3. Mediana:

```
# Calcular la mediana de la variable
mediana <- median(datos$PaymentTier)

# Imprimir la mediana de la variable
print(mediana)
```

```
## [1] 3
```

b.4.Tercer Cuartil:

```
# Obtener el tercer cuartil
q3 <- quantile(datos$PaymentTier, 0.75)

# Imprimir el tercer cuartil
print(q3)
```

```
## 75%
## 3
```

b.5.Valor Máximo:

```
# Calcular el valor máximo de la variable
max <- max(datos$PaymentTier)

# Imprimir el valor máximo de la variable
print(max)
```

```
## [1] 3
```

b.6.Recorrido (Rango):

```
# Calcular el rango de la variable
rango <- max(datos$PaymentTier) - min(datos$PaymentTier)

# Imprimir el rango de la variable
print(rango)
```

```
## [1] 2
```

b.7.Modas:

```
# Calcular la moda de la variable
moda <- as.numeric(names(sort(-table(datos$PaymentTier))[1]))

# Imprimir la moda de la variable
print(moda)
```

```
## [1] 3
```

b.8.Frecuencia:

```
# Calcular la frecuencia de la variable
frecuencia <- table(datos$PaymentTier)

# Imprimir la frecuencia de la variable
print(frecuencia)
```

```
##
##      1      2      3
## 243  918 3492
```

```
#Imprimir la frecuencia máxima
frecuencia_maxima <- max(frecuencia)
print(frecuencia_maxima)
```

```
## [1] 3492
```

c.Variable Age

c.1.Valor Mínimo:

```
# Obtener el valor mínimo
min <- min(datos$Age      )

# Imprimir el valor mínimo de la columna
print(min)
```

```
## [1] 22
```

c.2.Primer Cuartil:

```
# Obtener el primer cuartil
q1 <- quantile(datos$Age      , 0.25)

# Imprimir el primer cuartil
print(q1)
```

```
## 25%
## 26
```

c.3. Mediana:

```
# Calcular la mediana de la variable  
mediana <- median(datos$Age)
```

```
# Imprimir la mediana de la variable  
print(mediana)
```

```
## [1] 28
```

c.4.Tercer Cuartil:

```
# Obtener el tercer cuartil  
q3 <- quantile(datos$PaymentTier, 0.75)
```

```
# Imprimir el tercer cuartil  
print(q3)
```

```
## 75%  
## 3
```

c.5.Valor Máximo:

```
# Calcular el valor máximo de la variable  
max <- max(datos$Age )
```

```
# Imprimir el valor máximo de la variable  
print(max)
```

```
## [1] 41
```

c.6.Recorrido (Rango):

```
# Calcular el rango de la variable  
rango <- max(datos$Age) - min(datos$Age)
```

```
# Imprimir el rango de la variable  
print(rango)
```

```
## [1] 19
```

c.7.Modas:

```
# Calcular la moda de la variable  
moda <- as.numeric(names(sort(-table(datos$Age))[1]))
```

```
# Imprimir la moda de la variable  
print(moda)
```

```
## [1] 26
```

c.8.Frecuencia:

```
# Calcular la frecuencia de la variable
frecuencia <- table(datos$Age)
```

```
# Imprimir la frecuencia de la variable
print(frecuencia)
```

```
##
##  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
##  49  48 385 418 645 625 630 230 220 125 132 124 136 123 139 141 136 131 134  82
```

```
#Imprimir la frecuencia máxima
frecuencia_maxima <- max(frecuencia)
print(frecuencia_maxima)
```

```
## [1] 645
```

d.Variable ExperienceInCurrentDomain

d.1.Valor Mínimo:

```
# Obtener el valor mínimo
min <- min(datos$ExperienceInCurrentDomain)

# Imprimir el valor mínimo de la columna
print(min)
```

```
## [1] 0
```

d.2.Primer Cuartil:

```
# Obtener el primer cuartil
q1 <- quantile(datos$ExperienceInCurrentDomain, 0.25)

# Imprimir el primer cuartil
print(q1)
```

```
## 25%
##    2
```

d.3. Mediana:

```
# Calcular la mediana de la variable
mediana <- median(datos$ExperienceInCurrentDomain)

# Imprimir la mediana de la variable
print(mediana)
```

```
## [1] 3
```

d.4.Tercer Cuartil:


```
# Obtener el tercer cuartil
q3 <- quantile(datos$ExperienceInCurrentDomain, 0.75)

# Imprimir el tercer cuartil
print(q3)
```

```
## 75%
## 4
```

d.5.Valor Máximo:

```
# Calcular el valor máximo de la variable
max <- max(datos$ExperienceInCurrentDomain)

# Imprimir el valor máximo de la variable
print(max)
```

```
## [1] 7
```

d.6.Recorrido (Rango):

```
# Calcular el rango de la variable
rango <- max(datos$ExperienceInCurrentDomain) - min(datos$ExperienceInCurrentDomain)

# Imprimir el rango de la variable
print(rango)
```

```
## [1] 7
```

d.7.Modas:

```
# Calcular la moda de la variable
moda <- as.numeric(names(sort(-table(datos$ExperienceInCurrentDomain))[1]))

# Imprimir la moda de la variable
print(modas)
```

```
## [1] 2
```

d.8.Frecuencia:

```
# Calcular la frecuencia de la variable
frecuencia <- table(datos$ExperienceInCurrentDomain)

# Imprimir la frecuencia de la variable
print(frecuencia)
```

```
##
## 0 1 2 3 4 5 6 7
## 355 558 1087 786 931 919 8 9
```

```
#Imprimir la frecuencia máxima
frecuencia_maxima <- max(frecuencia)
print(frecuencia_maxima)
```

```
## [1] 1087
```

e.Variable LeaveOrNot

e.1.Valor Mínimo:

```
# Obtener el valor mínimo
min <- min(datos$LeaveOrNot)

# Imprimir el valor mínimo de la columna
print(min)
```

```
## [1] 0
```

e.2.Primer Cuartil:

```
# Obtener el primer cuartil
q1 <- quantile(datos$LeaveOrNot, 0.25)

# Imprimir el primer cuartil
print(q1)
```

```
## 25%
## 0
```

e.3. Mediana:

```
# Calcular la mediana de la variable
mediana <- median(datos$LeaveOrNot)

# Imprimir la mediana de la variable
print(mediana)
```

```
## [1] 0
```

e.4.Tercer Cuartil:

```
# Obtener el tercer cuartil
q3 <- quantile(datos$LeaveOrNot, 0.75)

# Imprimir el tercer cuartil
print(q3)
```

```
## 75%
## 1
```

e.5.Valor Máximo:

```
# Calcular el valor máximo de la variable
max <- max(datos$LeaveOrNot)
```

```
# Imprimir el valor máximo de la variable
print(max)
```

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

e.6.Recorrido (Rango):

```
# Calcular el rango de la variable
rango <- max(datos$LeaveOrNot) - min(datos$LeaveOrNot)
```

```
# Imprimir el rango de la variable
print(rango)
```

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

e.7.Modas:

```
# Calcular la moda de la variable
moda <- as.numeric(names(sort(-table(datos$LeaveOrNot))[1]))
```

```
# Imprimir la moda de la variable
print(modas)
```

```
## [1] 0
```

e.8.Frecuencia:

```
# Calcular la frecuencia de la variable
frecuencia <- table(datos$LeaveOrNot)
```

```
# Imprimir la frecuencia de la variable
print(frecuencia)
```

```
##
##      0      1
## 3053 1600
```

```
#Imprimir la frecuencia máxima
frecuencia_maxima <- max(frecuencia)
print(frecuencia_maxima)
```

```
## [1] 3053
```

1.1.2 Variables Categóricas

Obtenemos los estadísticos descriptivos de las variables categóricas

```
# Seleccionar solo las columnas categóricas
categorical_data <- datos %>%
  select_if(is.character)

# Obtener estadísticos descriptivos para las variables categóricas
summary_stats_categorical <- summary(categorical_data)

# Imprimir los estadísticos descriptivos
print(summary_stats_categorical)
```

```
##      Education      City      Gender      EverBenched
## Length:4653      Length:4653      Length:4653      Length:4653
## Class :character  Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character  Mode  :character
```

Obtenemos los estadísticos descriptivos de cada variable:

a.Variable Education

a.1.Frecuencia:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$Education)

# Imprimir los valores y sus frecuencias
print(frecuencia)
```

```
##
## Bachelors  Masters      PHD
##      3601      873      179
```

a.2.Moda:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$Education)

# Encontrar el valor que más se repite
valor_mas_repetido <- names(frecuencia)[which.max(frecuencia)]

# Imprimir el valor que más se repite
print(valor_mas_repetido)
```

```
## [1] "Bachelors"
```

b.Variable City

b.1.Frecuencia:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$City)

# Imprimir los valores y sus frecuencias
print(frecuencia)
```

```
##
## Bangalore New Delhi      Pune
##      2228      1157      1268
```

b.2.Moda:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$City)

# Encontrar el valor que más se repite
valor_mas_repetido <- names(frecuencia)[which.max(frecuencia)]

# Imprimir el valor que más se repite
print(valor_mas_repetido)
```

```
## [1] "Bangalore"
```

c.Variable Gender

c.1.Frecuencia:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$Gender)

# Imprimir los valores y sus frecuencias
print(frecuencia)
```

```
##
## Female      Male
##      1875      2778
```

c.2.Moda:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$Gender)

# Encontrar el valor que más se repite
valor_mas_repetido <- names(frecuencia)[which.max(frecuencia)]

# Imprimir el valor que más se repite
print(valor_mas_repetido)
```

```
## [1] "Male"
```

d.Variable EverBenched

d.1.Frecuencia:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$EverBenched)

# Imprimir los valores y sus frecuencias
print(frecuencia)
```

```
##
##   No   Yes
## 4175  478
```

d.2.Modas:

```
# Calcular la frecuencia de cada valor en la variable categórica
frecuencia <- table(datos$EverBenched)

# Encontrar el valor que más se repite
valor_mas_repetido <- names(frecuencia)[which.max(frecuencia)]

# Imprimir el valor que más se repite
print(valor_mas_repetido)
```

```
## [1] "No"
```

1.2. Dispersión

1.1.1 Variables Numéricas

```
# Seleccionar solo las columnas numéricas
numeric_data <- datos %>%
  select_if(is.numeric)

# Calcular el rango
rango <- apply(numeric_data, 2, function(x) max(x) - min(x))

# Calcular la varianza
varianza <- apply(numeric_data, 2, var)

# Calcular la desviación estándar
desviacion_estandar <- apply(numeric_data, 2, sd)

# Crear un dataframe con los resultados
estadisticos_dispersion <- data.frame(
  Variable = names(numeric_data),
  Rango = rango,
  Varianza = varianza,
  Desviacion_Estandar = desviacion_estandar
)

# Imprimir los estadísticos de dispersión
print(estadisticos_dispersion)
```

```
##
##           Variable Rango  Varianza
## JoiningYear      6  3.4721732
## PaymentTier      2  0.3152098
## Age             19 23.2911158
## ExperienceInCurrentDomain ExperienceInCurrentDomain 7  2.4281129
## LeaveOrNot       1  0.2256701
##           Desviacion_Estandar
```

```
## JoiningYear          1.8633768
## PaymentTier          0.5614355
## Age                  4.8260870
## ExperienceInCurrentDomain 1.5582403
## LeaveOrNot           0.4750475
```

2. Análisis Exploratorio de Datos (EDA)

```
# Conteo de datos faltantes en cada columna
missing_data_counts <- sapply(datos, function(x) sum(is.na(x)))

# Mostrando el conteo de datos faltantes
missing_data_counts
```

```
##           Education          JoiningYear          City
##           0              0              0
##           PaymentTier          Age          Gender
##           0              0              0
##           EverBenched ExperienceInCurrentDomain          LeaveOrNot
##           0              0              0
```

```
summary(datos)
```

```
##      Education          JoiningYear          City          PaymentTier
## Length:4653      Min.   :2012      Length:4653      Min.   :1.000
## Class :character  1st Qu.:2013      Class :character  1st Qu.:3.000
## Mode  :character  Median :2015      Mode  :character  Median :3.000
##                               Mean  :2015      Mean  :2.698
##                               3rd Qu.:2017      3rd Qu.:3.000
##                               Max.   :2018      Max.   :3.000
##      Age          Gender          EverBenched
## Min.   :22.00      Length:4653      Length:4653
## 1st Qu.:26.00      Class :character  Class :character
## Median :28.00      Mode  :character  Mode  :character
## Mean   :29.39
## 3rd Qu.:32.00
## Max.   :41.00
## ExperienceInCurrentDomain  LeaveOrNot
## Min.   :0.000      Min.   :0.0000
## 1st Qu.:2.000      1st Qu.:0.0000
## Median :3.000      Median :0.0000
## Mean   :2.906      Mean   :0.3439
## 3rd Qu.:4.000      3rd Qu.:1.0000
## Max.   :7.000      Max.   :1.0000
```

```
# Identificando variables numéricas
numeric_vars <- sapply(datos, function(x) is.numeric(x) || is.integer(x))
numeric_variables <- names(datos)[numeric_vars]

# Identificando variables categóricas
categorical_vars <- sapply(datos, function(x) is.factor(x) || is.character(x) || is.logical(x))
```

```
categorical_variables <- names(datos)[categorical_vars]

# Mostrando las variables numéricas y categóricas
list(numeric = numeric_variables, categorical = categorical_variables)
```

```
## $numeric
## [1] "JoiningYear"          "PaymentTier"
## [3] "Age"                  "ExperienceInCurrentDomain"
## [5] "LeaveOrNot"
##
## $categorical
## [1] "Education"  "City"      "Gender"    "EverBenched"
```

```
# Cargar la librería ggplot2
library(ggplot2)

# Crear un data frame con los conteos de cada año
joining_year_data <- as.data.frame(table(datos$JoiningYear))

# Crear el gráfico de barras usando ggplot2
ggplot(joining_year_data, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  ggtitle("Gráfico de Barras de JoiningYear") +
  xlab("Año de Ingreso") +
  ylab("Frecuencia")
```

```
# Cargar la librería ggplot2
library(ggplot2)

# Crear un data frame con los conteos de cada categoría de PaymentTier
payment_tier_data <- as.data.frame(table(datos$PaymentTier))

# Crear el gráfico de barras usando ggplot2
ggplot(payment_tier_data, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "coral") +
  ggtitle("Gráfico de Barras de PaymentTier") +
  xlab("Tier de Pago") +
  ylab("Frecuencia")
```

```
# Cargar la librería ggplot2
library(ggplot2)

# Crear un histograma usando ggplot2
ggplot(datos, aes(x = Age)) +
  geom_histogram(binwidth = 1, fill="blue", color="black") +
  ggtitle("Histograma de Age") +
  xlab("Edad") +
  ylab("Frecuencia")
```

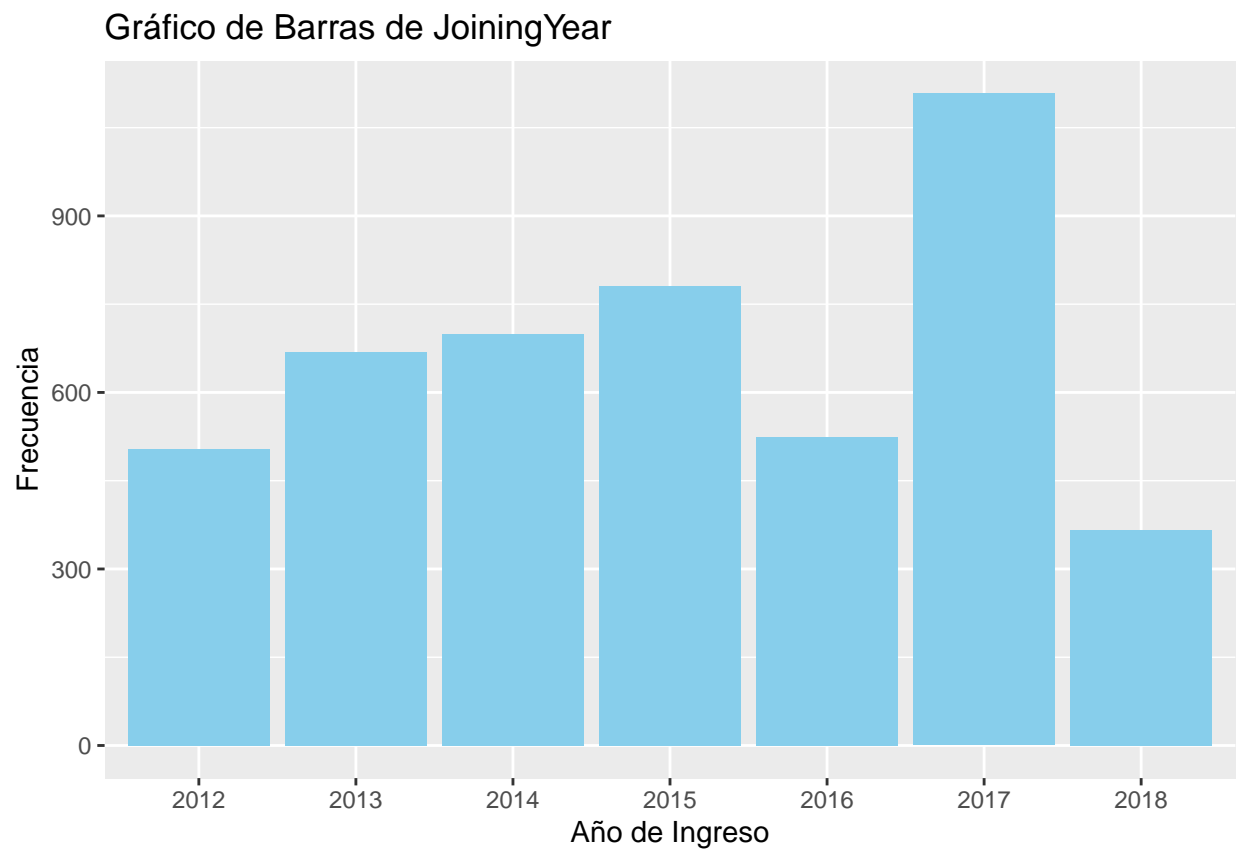



Figure 1: Gráfico de Barras de JoiningYear con ggplot2

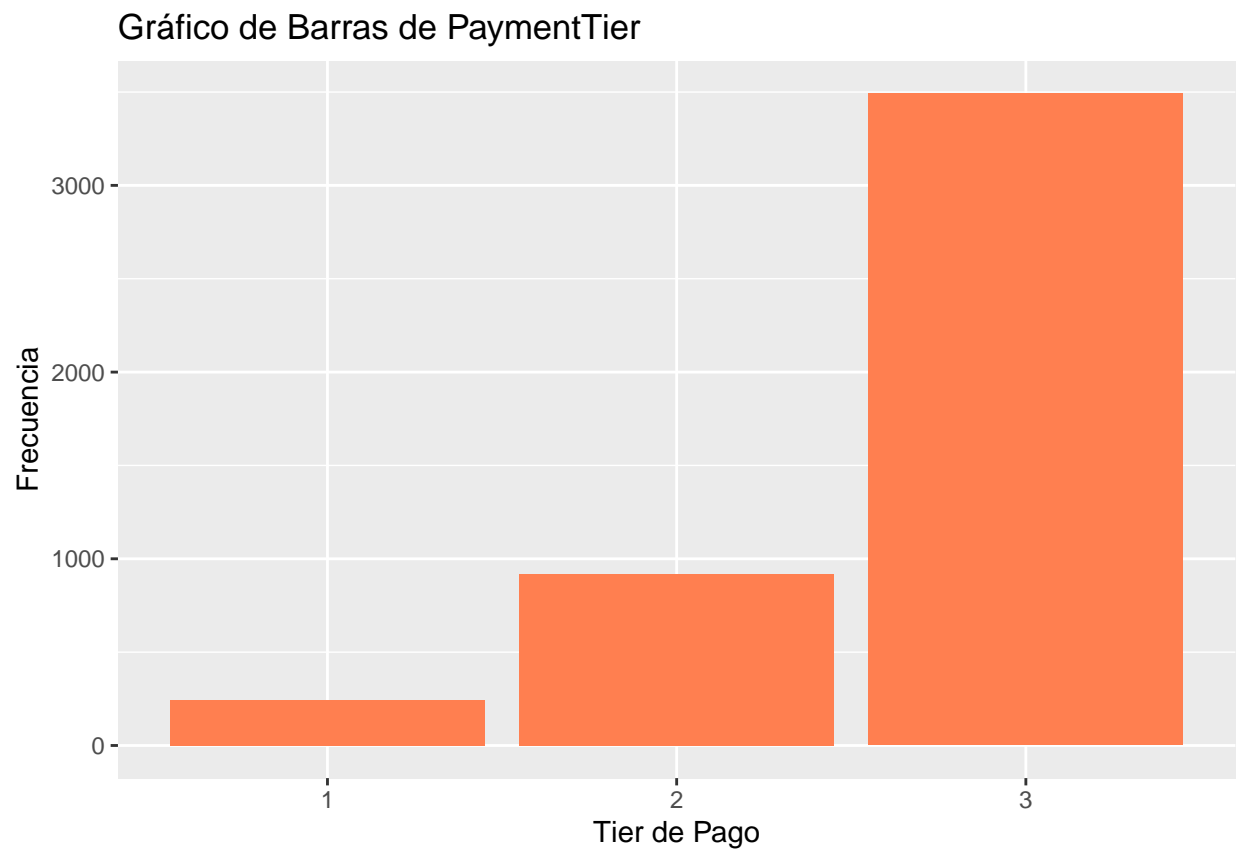


Figure 2: Gráfico de Barras de PaymentTier con ggplot2

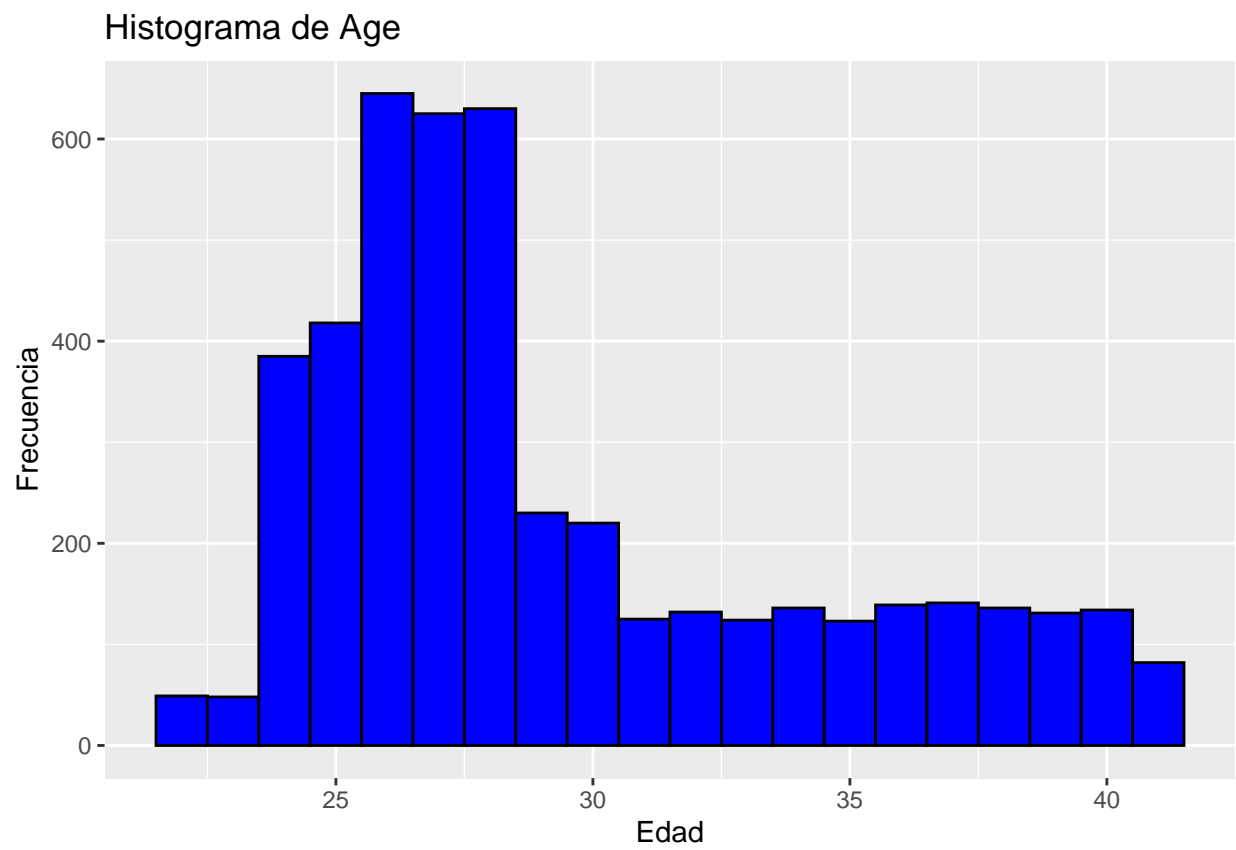


Figure 3: Histograma de Age con ggplot2

```
# Cargar la librería ggplot2
library(ggplot2)

# Creando un box plot usando ggplot2
ggplot(datos, aes(y = Age)) +
  geom_boxplot(fill = "lightgreen") +
  ggtitle("Box Plot de Age") +
  ylab("Edad")
```

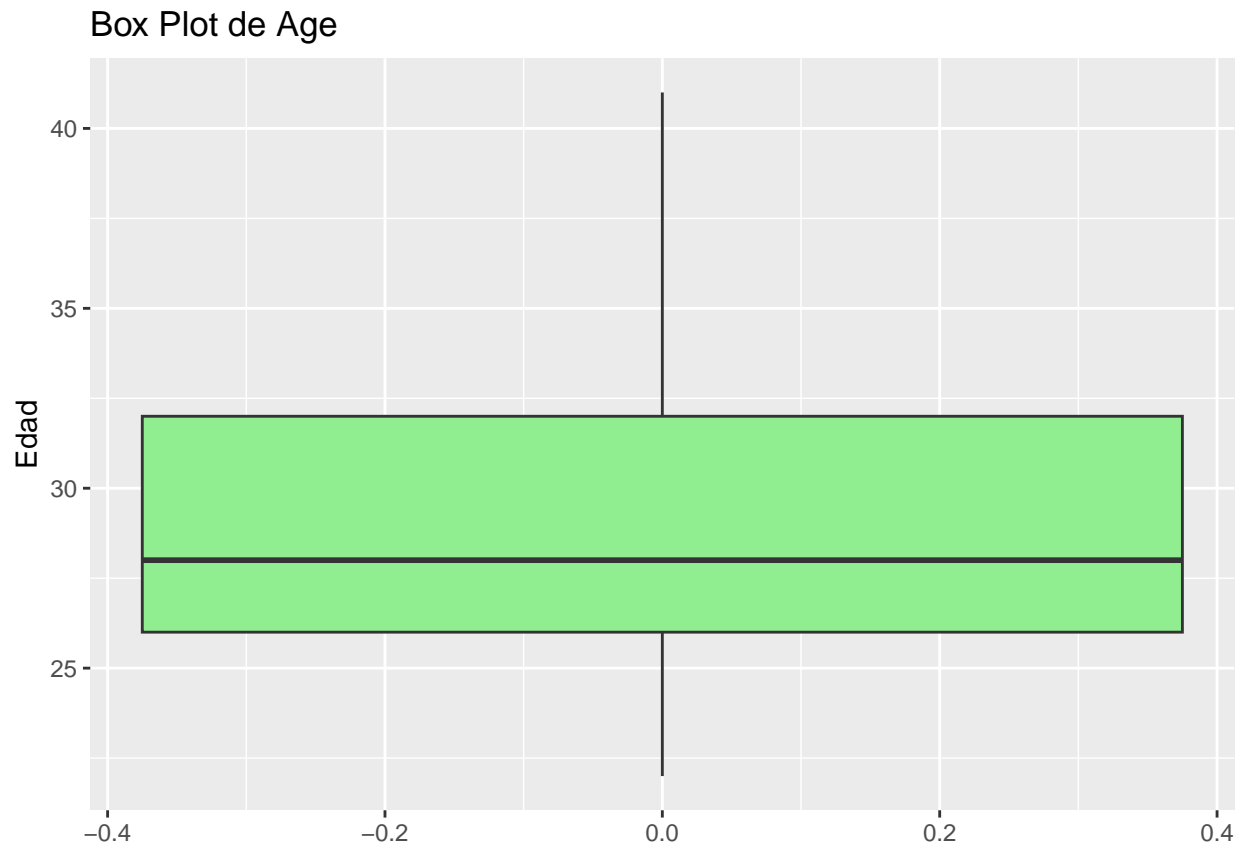


Figure 4: Box Plot de Age con ggplot2

```
# Cargar la librería ggplot2
library(ggplot2)

# Crear un data frame con los conteos de cada valor de ExperienceInCurrentDomain
experience_data <- as.data.frame(table(datos$ExperienceInCurrentDomain))

# Crear el gráfico de barras usando ggplot2
ggplot(experience_data, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  ggtitle("Gráfico de Barras de ExperienceInCurrentDomain") +
  xlab("Años de Experiencia en el Dominio Actual") +
  ylab("Frecuencia")
```

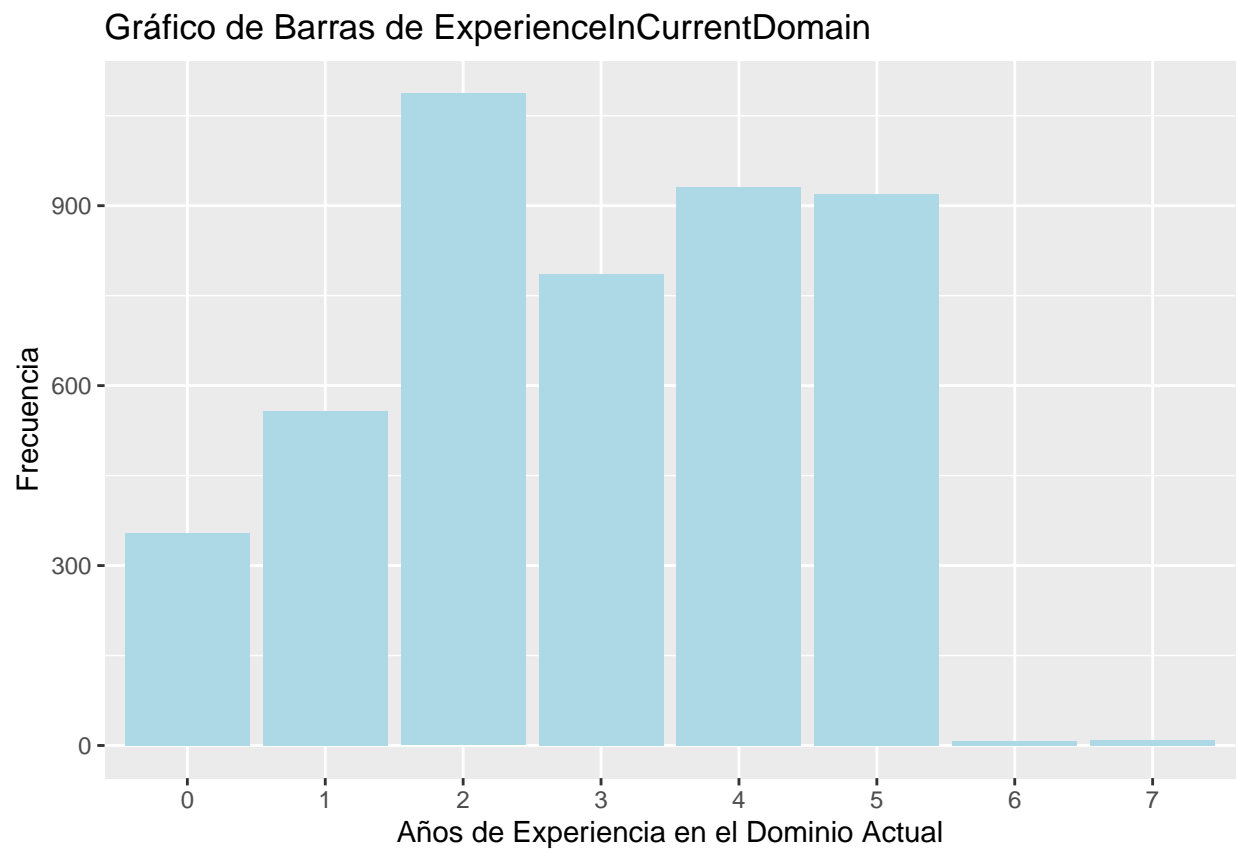


Figure 5: Gráfico de Barras de ExperienceInCurrentDomain con ggplot2

```
# Cargar la librería ggplot2
library(ggplot2)

# Crear un data frame con los conteos de cada categoría de LeaveOrNot
leave_data <- as.data.frame(table(datos$LeaveOrNot))

# Crear el gráfico de barras usando ggplot2
ggplot(leave_data, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "salmon") +
  ggtitle("Gráfico de Barras de LeaveOrNot") +
  xlab("Decisión") +
  ylab("Frecuencia")
```

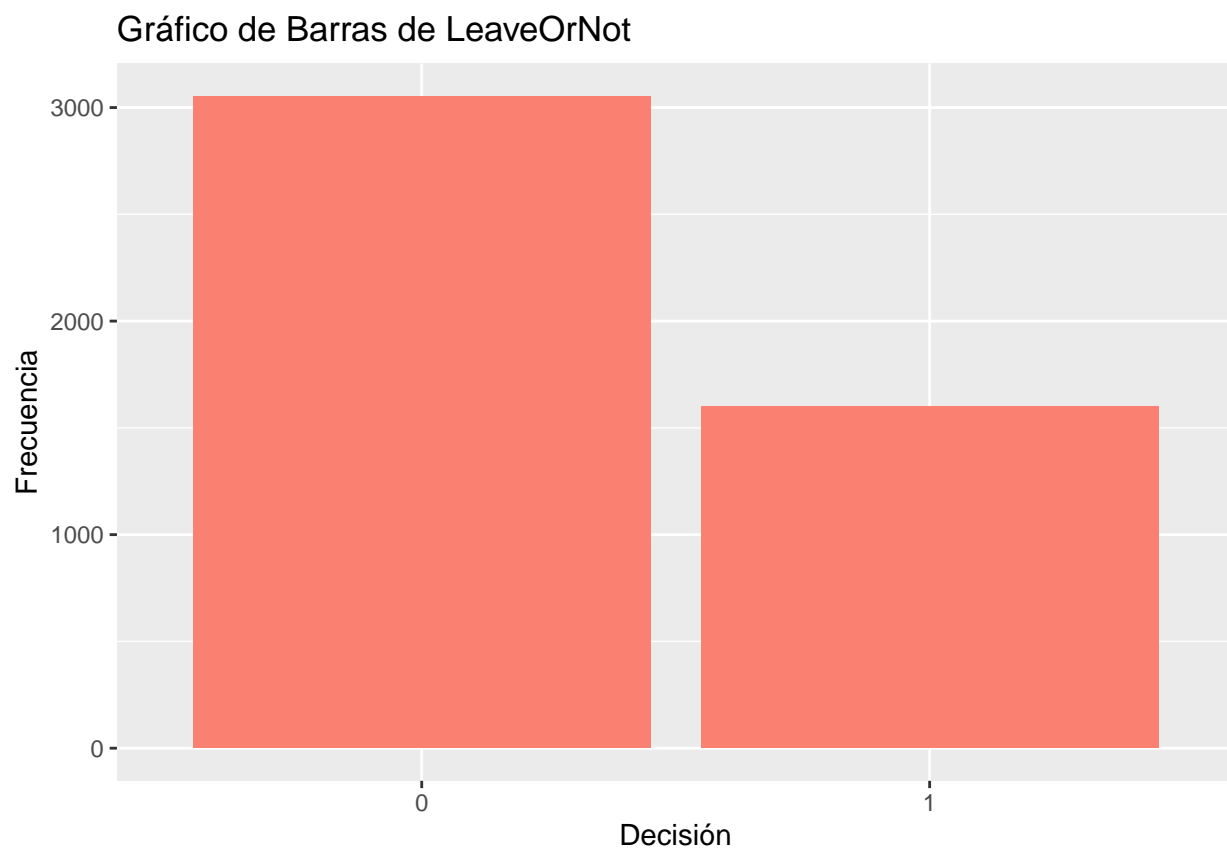


Figure 6: Gráfico de Barras de LeaveOrNot con ggplot2

```
# Identifying character variables
char_vars <- sapply(datos, class) == "character"

# Extracting unique values for each character variable
unique_values <- lapply(datos[, char_vars, drop = FALSE], unique)

# Displaying the unique values
unique_values

## $Education
```

```
## [1] "Bachelors" "Masters"    "PHD"
##
## $City
## [1] "Bangalore" "Pune"      "New Delhi"
##
## $Gender
## [1] "Male"    "Female"
##
## $EverBenched
## [1] "No"    "Yes"
```

```
# Cargar la librería ggplot2
```

```
library(ggplot2)
```

```
# Crear un data frame con los conteos de cada categoría de Education
education_data <- as.data.frame(table(datos$Education))
```

```
# Crear el gráfico de barras usando ggplot2
```

```
ggplot(education_data, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "turquoise") +
  ggtitle("Gráfico de Barras de Education") +
  xlab("Nivel de Educación") +
  ylab("Frecuencia")
```

Gráfico de Barras de Education

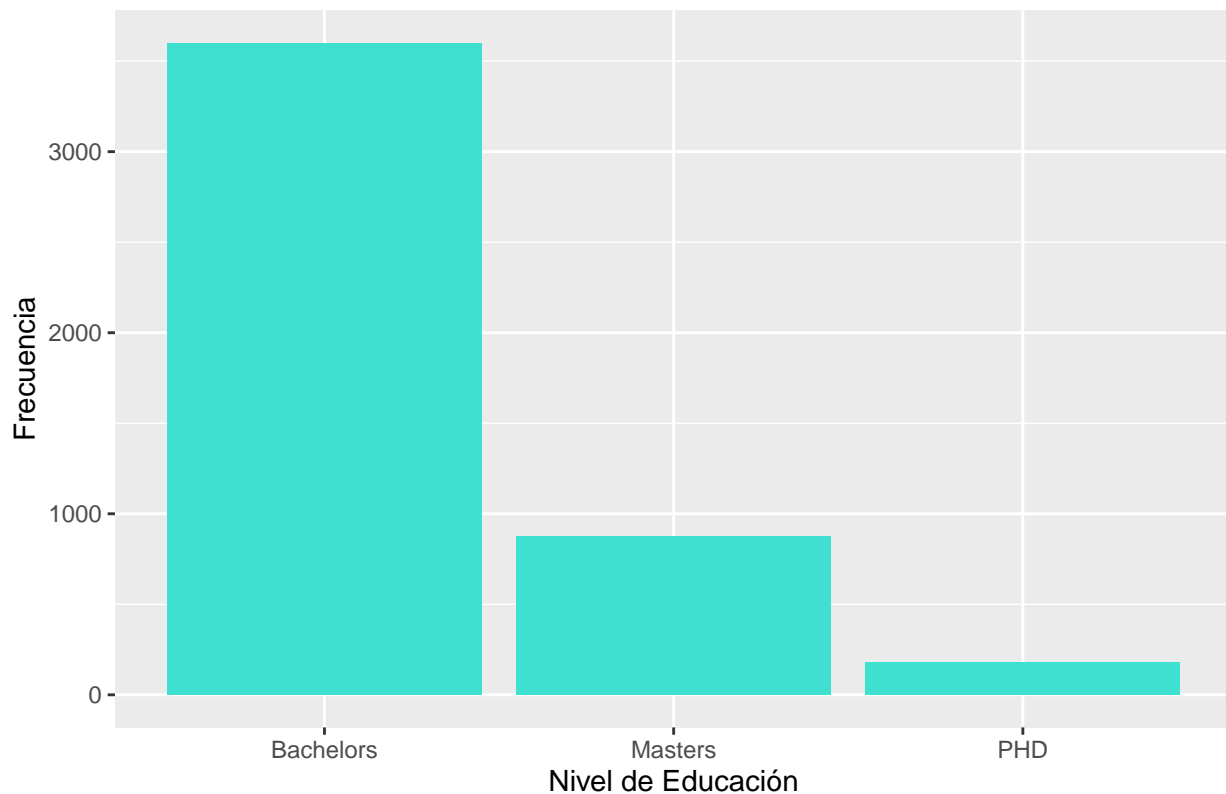


Figure 7: Gráfico de Barras de Education con ggplot2

```

# Cargar la librería ggplot2
library(ggplot2)

# Crear un data frame con los conteos de cada ciudad
city_data <- as.data.frame(table(datos$City))

# Crear el gráfico de barras usando ggplot2
ggplot(city_data, aes(x = reorder(Var1, -Freq), y = Freq)) +
  geom_bar(stat = "identity", fill = "cadetblue") +
  ggtitle("Gráfico de Barras de City") +
  xlab("Ciudad") +
  ylab("Frecuencia") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotating x-axis labels for better visibility

```

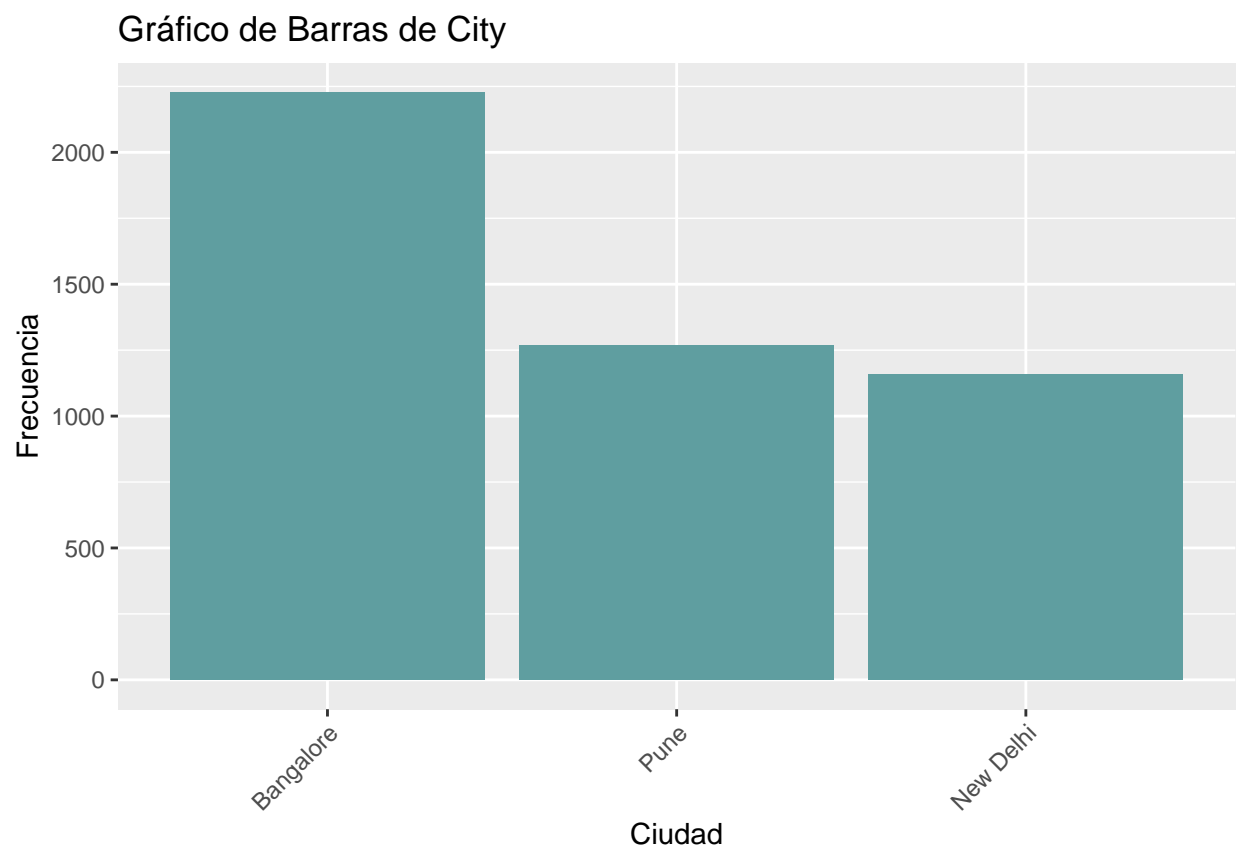


Figure 8: Gráfico de Barras de City con ggplot2

```

# Cargar la librería ggplot2
library(ggplot2)

# Crear un data frame con los conteos de cada género
gender_data <- as.data.frame(table(datos$Gender))

# Crear el gráfico de barras usando ggplot2
ggplot(gender_data, aes(x = Var1, y = Freq)) +

```



```
geom_bar(stat = "identity", fill = "orchid") +
ggtitle("Gráfico de Barras de Gender") +
xlab("Género") +
ylab("Frecuencia")
```

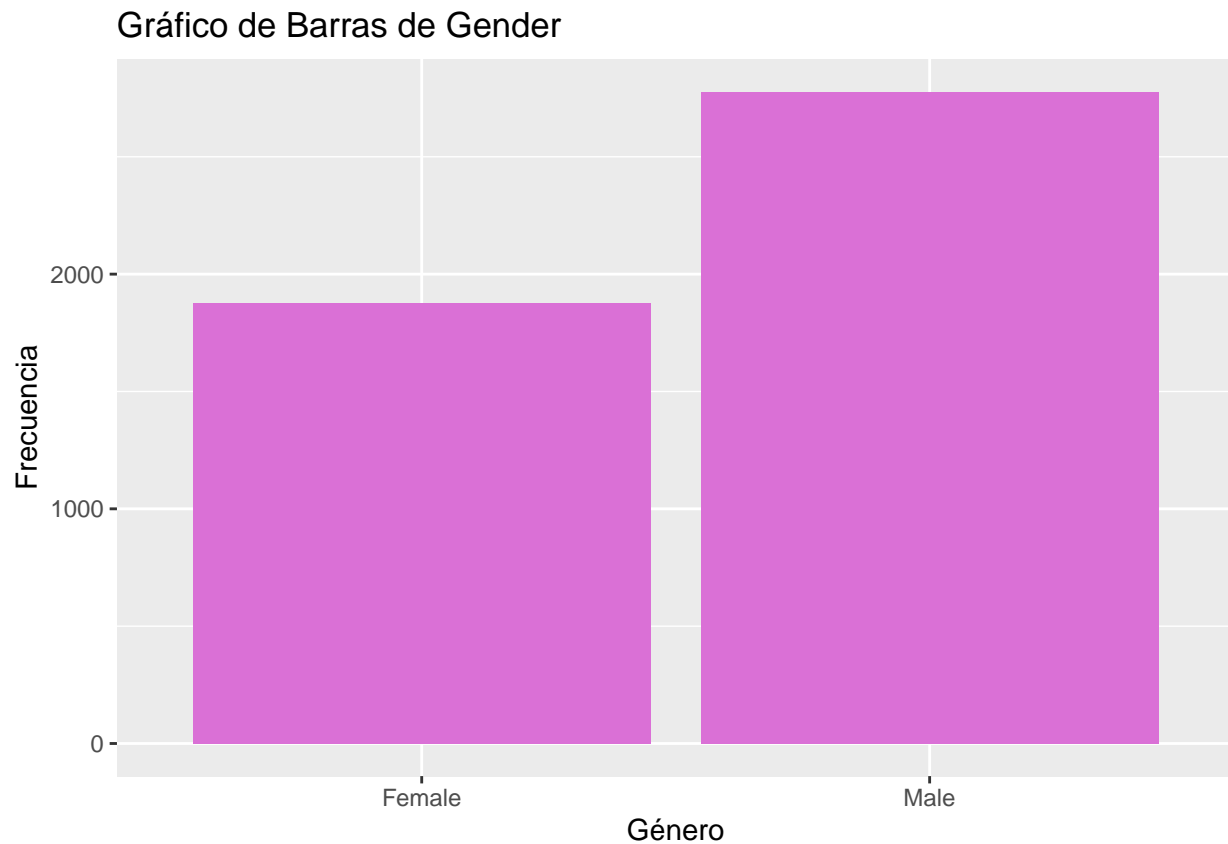


Figure 9: Gráfico de Barras de Gender con ggplot2

```
# Cargar la librería ggplot2
library(ggplot2)

# Crear un data frame con los conteos de cada categoría de EverBenched
everbenched_data <- as.data.frame(table(datos$EverBenched))

# Crear el gráfico de barras usando ggplot2
ggplot(everbenched_data, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "lightcoral") +
  ggtitle("Gráfico de Barras de EverBenched") +
  xlab("Ha Sido Bencheado") +
  ylab("Frecuencia")

# Converting 'LeaveOrNot' to a factor if it's not already
datos$LeaveOrNot <- as.factor(datos$LeaveOrNot)
# Optionally, convert other categorical variables
```

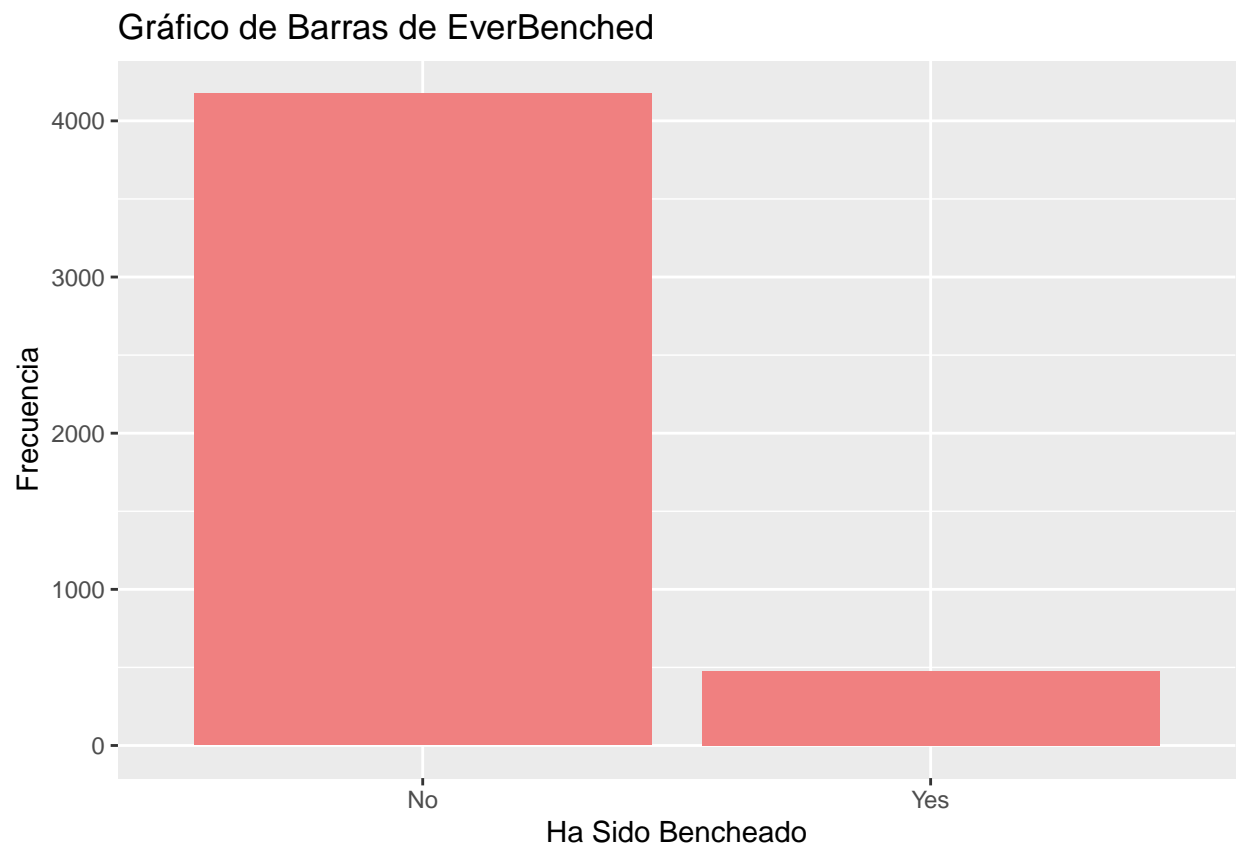


Figure 10: Gráfico de Barras de EverBenched con ggplot2

```

datos$PaymentTier <- as.factor(datos$PaymentTier)

# Now, subset the dataset with the required variables
selected_variables <- datos[, c("Age", "JoiningYear", "ExperienceInCurrentDomain", "PaymentTier", "LeaveOrNot")]

# Creating the pairplot using GGally
library(GGally)
ggpairs(selected_variables, aes(color = LeaveOrNot),
        lower = list(continuous = wrap("points", alpha = 0.5, size = 1)),
        diag = list(continuous = wrap("barDiag")),
        upper = list(continuous = wrap("cor", size = 3)),
        title = "Pairplot with LeaveOrNot as Hue")

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

# Subsetting the dataset to include only numerical variables
numerical_data <- datos[sapply(datos, is.numeric)]

# Calculating the correlation matrix
cor_matrix <- cor(numerical_data, use="pairwise.complete.obs") # Handling missing values by considering

# Display the correlation matrix
cor_matrix

##              JoiningYear      Age ExperienceInCurrentDomain
## JoiningYear      1.00000000  0.01316529      -0.03652462
## Age              0.01316529  1.00000000      -0.13464285
## ExperienceInCurrentDomain -0.03652462 -0.13464285      1.00000000

# Visualizing the correlation matrix
corrplot(cor_matrix, method="circle", type="upper", order="hclust",
        tl.col="black", tl.srt=45) # Rotate labels for better readability

```

3. Estadísticos Inferenciales

```

#install.packages("ggplot2")
#install.packages("conflicted")

```

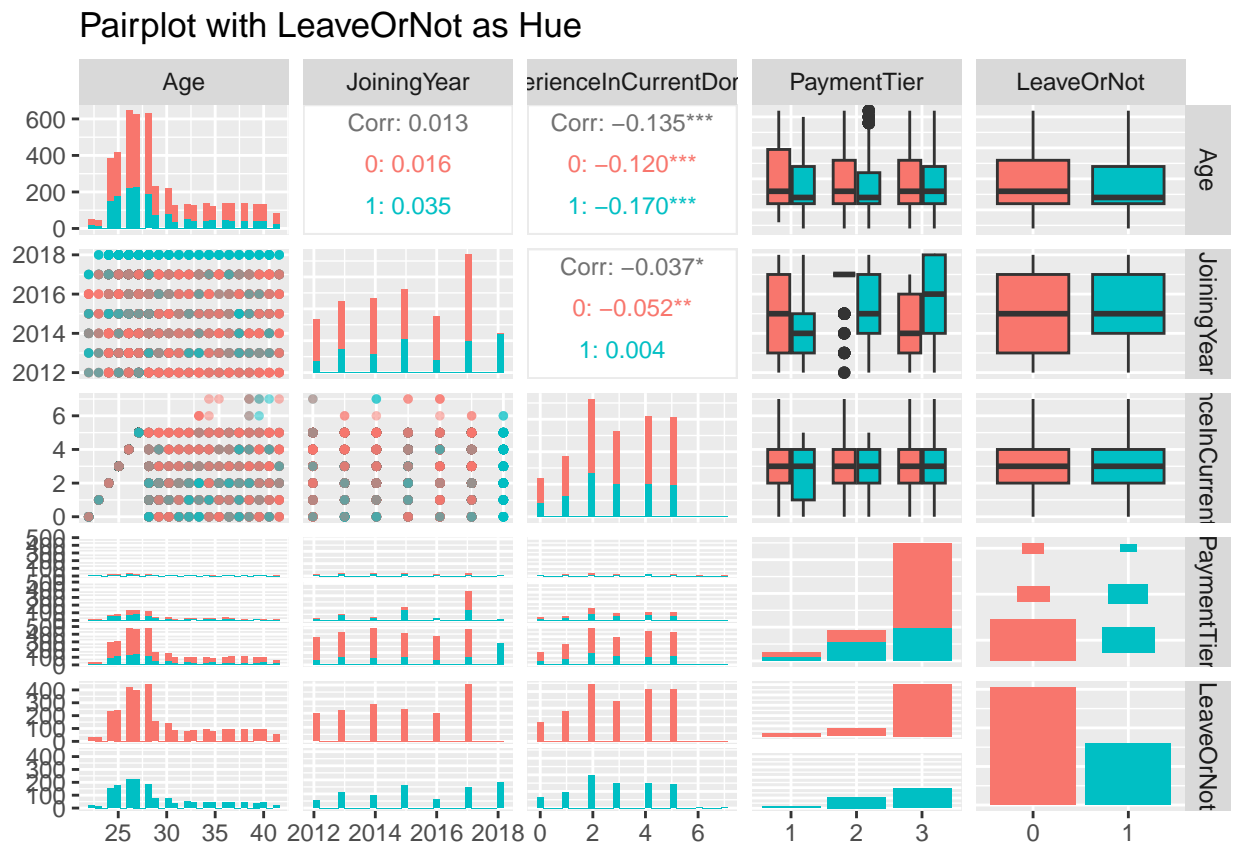


Figure 11: Pairplot with LeaveOrNot as Hue

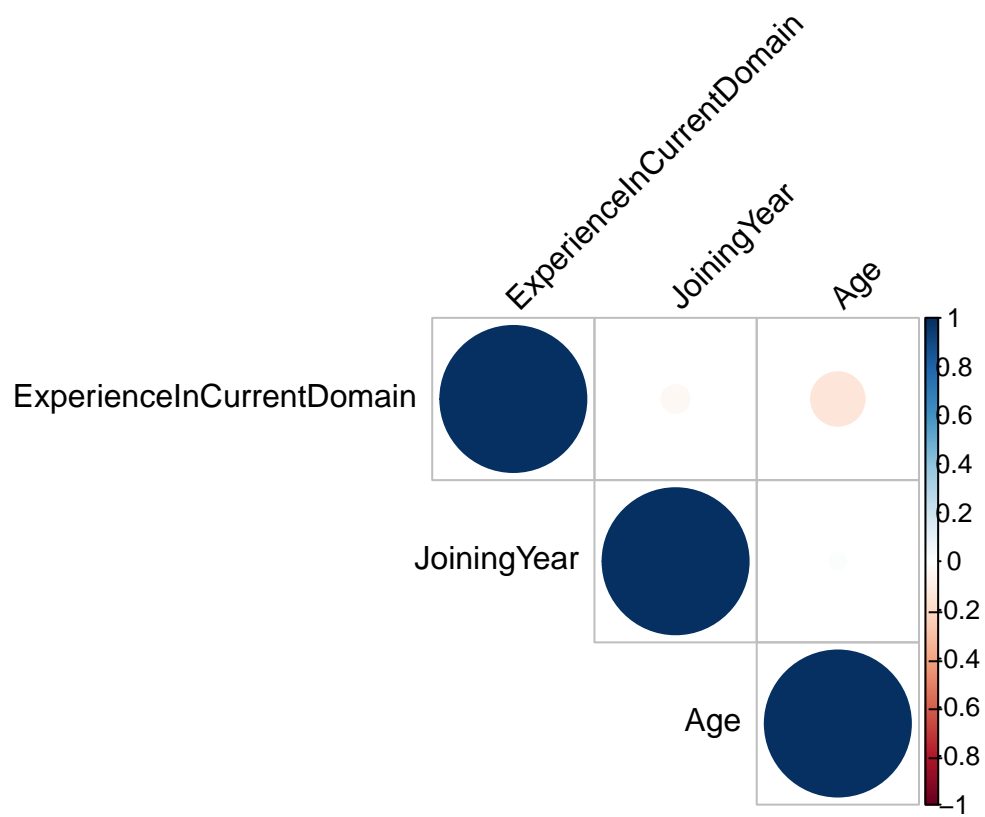


Figure 12: Correlation Matrix Visualization

```
# Cargamos las librerías necesarias
```

```
library(conflicted)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v forcats 1.0.0 v stringr 1.5.1
```

```
## v lubridate 1.9.3 v tibble 3.2.1
```

```
## v purrr 1.0.2 v tidyr 1.3.1
```

```
## v readr 2.1.5
```

```
library(ggplot2)
```

```
library(reshape2)
```

```
# CARGAMOS LOS DATOS
```

```
df <- datos
```

CALCULAMOS LA MATRIZ DE CORRELACION

```
cor_matrix <- cor(df[, sapply(df, is.numeric)])
```

```
cor_matrix
```

```
##               JoiningYear      Age ExperienceInCurrentDomain
## JoiningYear      1.00000000  0.01316529             -0.03652462
## Age              0.01316529  1.00000000             -0.13464285
## ExperienceInCurrentDomain -0.03652462 -0.13464285             1.00000000
```

VISUALIZAMOS LA MATRIZ DE CORRELACION

```
hot_matrix <- melt(cor_matrix)
```

```
ggplot(hot_matrix, aes(Var1, Var2, fill=value)) +
```

```
  geom_tile(color="white") +
```

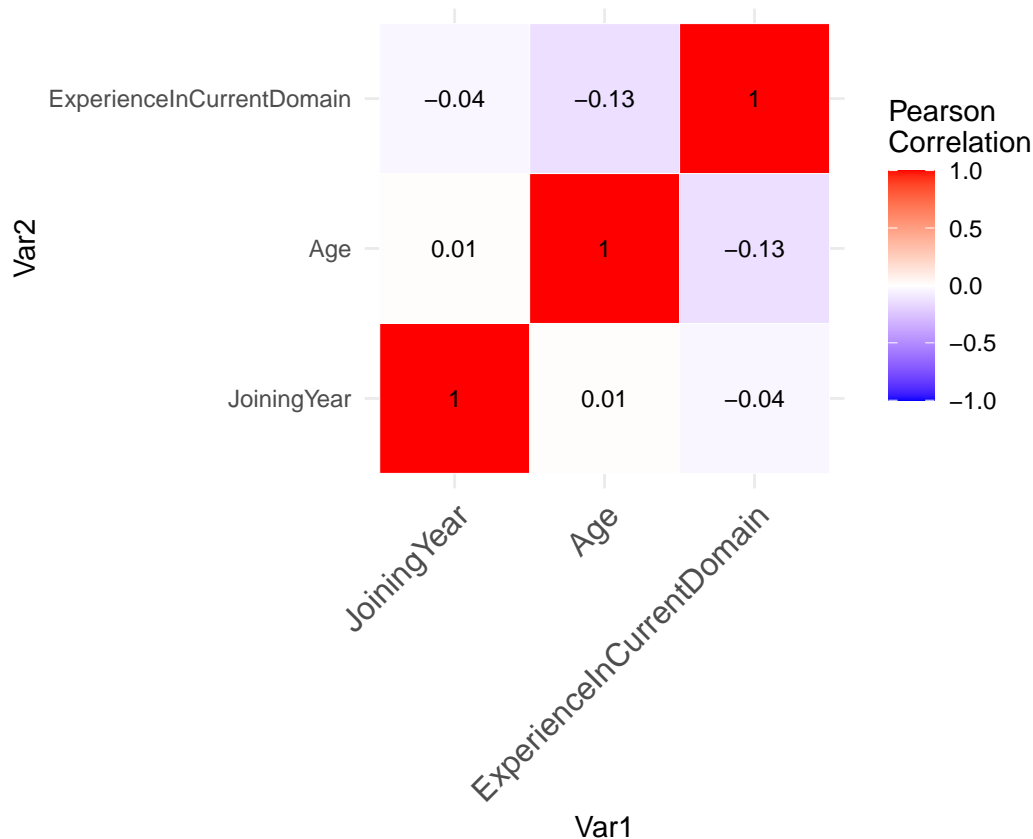
```
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",  
                        midpoint=0, limit=c(-1,1), space="Lab",  
                        name="Pearson\nCorrelation") +
```

```
  geom_text(aes(Var1, Var2, label = round(value, 2)), color = "black", size = 3) +
```

```
  theme_minimal() +
```

```
  theme(axis.text.x = element_text(angle = 45, vjust = 1,  
                                     size = 12, hjust = 1)) +
```

```
  coord_fixed()
```

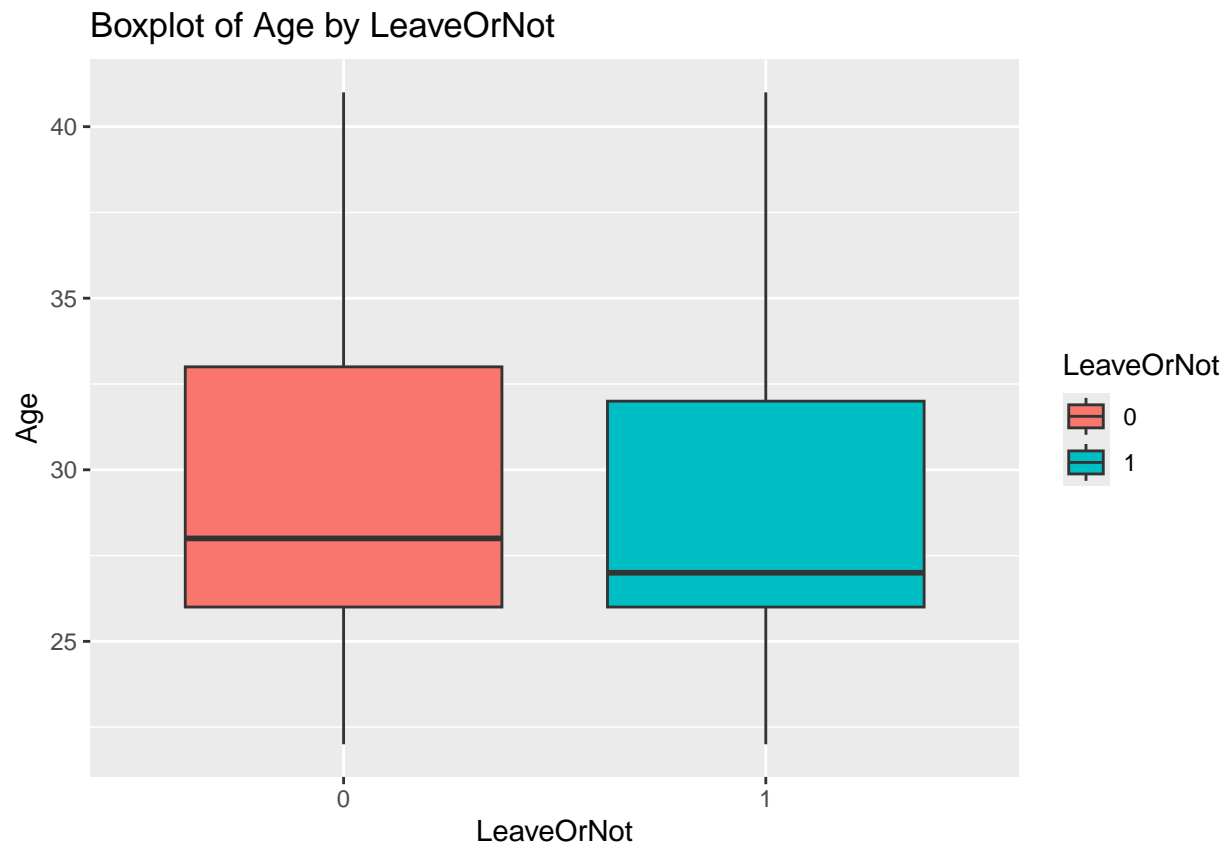


La matriz de correlación muestra las siguientes relaciones entre las variables numéricas:

- La correlación entre las variables no muestra relaciones fuertemente lineales. La mayoría de los coeficientes son cercanos a cero, indicando poca o ninguna correlación lineal.
- La variable LeaveOrNot tiene una correlación positiva de 0.18 con JoiningYear (año de ingreso), lo que podría indicar que mientras más tiempo lleva un empleado en la empresa, es más probable que decida no dejarla.
- La variable LeaveOrNot tiene una relación negativa con PaymentTier (nivel de salario) de -0.20, lo que podría indicar que a menor salario un empleado en la empresa, es más probable que decida dejar la empresa.
- Las otras correlaciones son relativamente bajas, sugiriendo que estas variables no están fuertemente influenciadas entre sí en términos lineales.

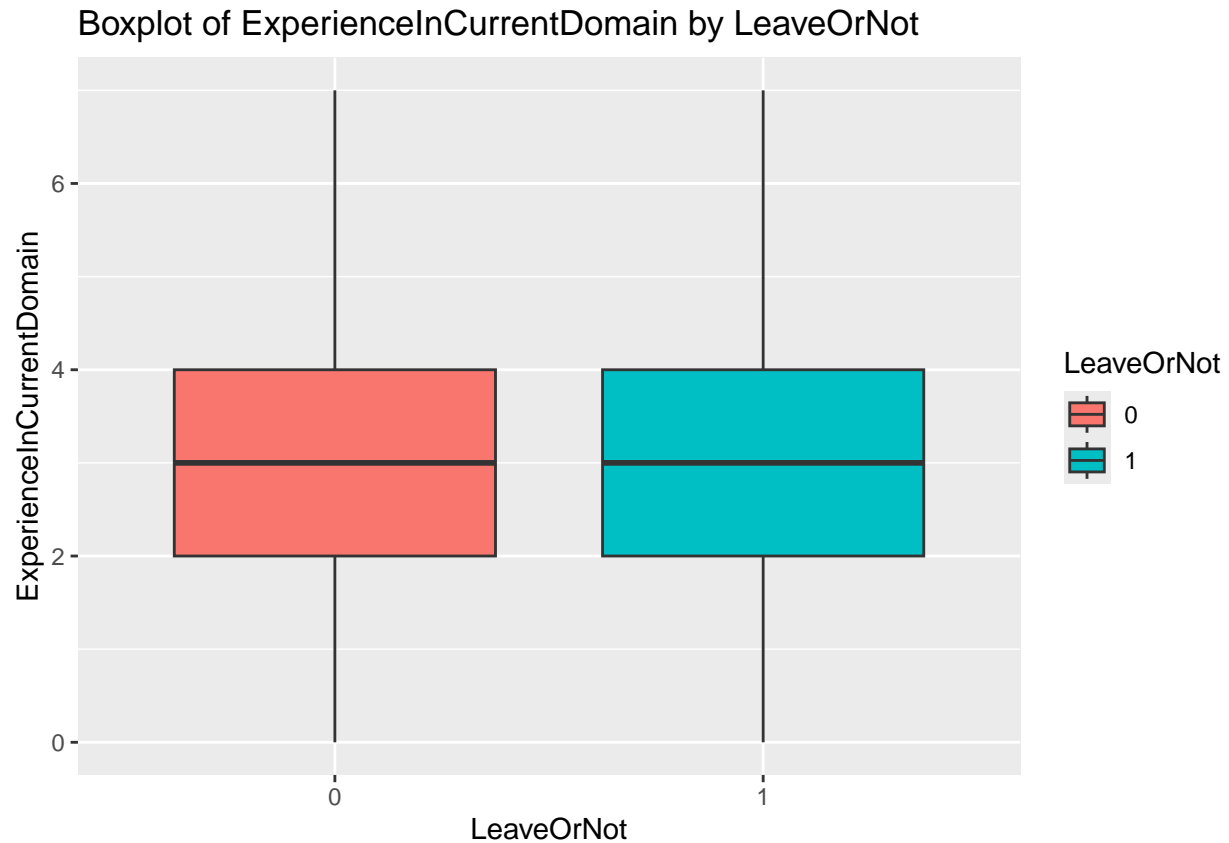
REALIZAMOS UN BOX PLOT PARA VER LAS RELACIONES ENTRE LA VARIABLE EDAD Y LA VARIABLE TARGET

```
ggplot(df, aes(x=LeaveOrNot, y=Age, fill=LeaveOrNot)) +
  geom_boxplot() +
  labs(title = "Boxplot of Age by LeaveOrNot", x = "LeaveOrNot", y = "Age")
```



REALIZAMOS UN BOX PLOT PARA VER LAS RELACIONES ENTRE LA VARIABLE ExperienceInCurrentDomain Y LA VARIABLE TARGET

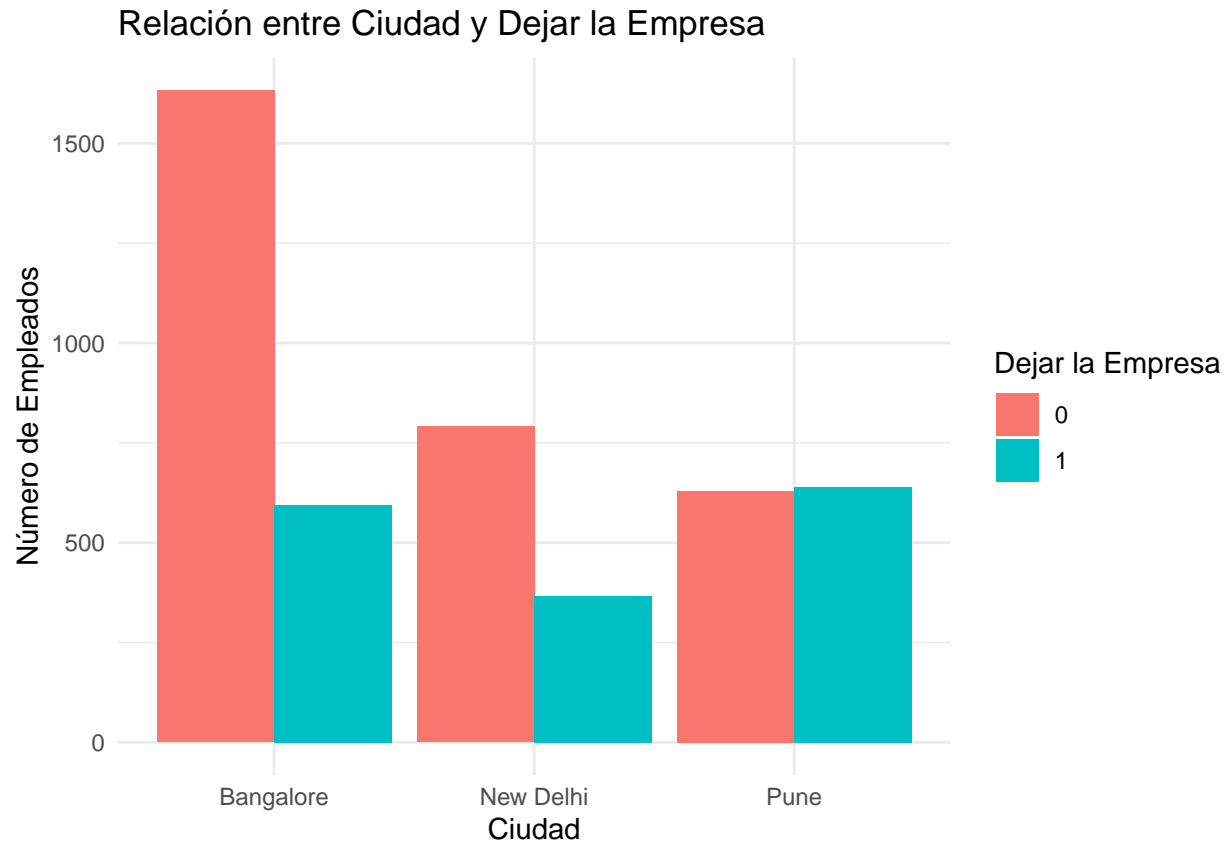
```
ggplot(df, aes(x=LeaveOrNot, y=ExperienceInCurrentDomain, fill=LeaveOrNot)) +  
  geom_boxplot() +  
  labs(title = "Boxplot of ExperienceInCurrentDomain by LeaveOrNot", x = "LeaveOrNot", y = "ExperienceInCurrentDomain")
```

- Edad vs. Dejar la Empresa: No parece haber una diferencia significativa en la edad de los empleados que deciden dejar la empresa y los que se quedan.
- Experiencia en el Dominio Actual vs. Dejar la Empresa: Similarmente, la experiencia en el dominio actual no muestra una diferencia clara en términos de afectar la decisión de dejar o no la empresa.

GRAFICO DE RELACION ENTRE VARIABLES “City” VS “LeaveOrNot”

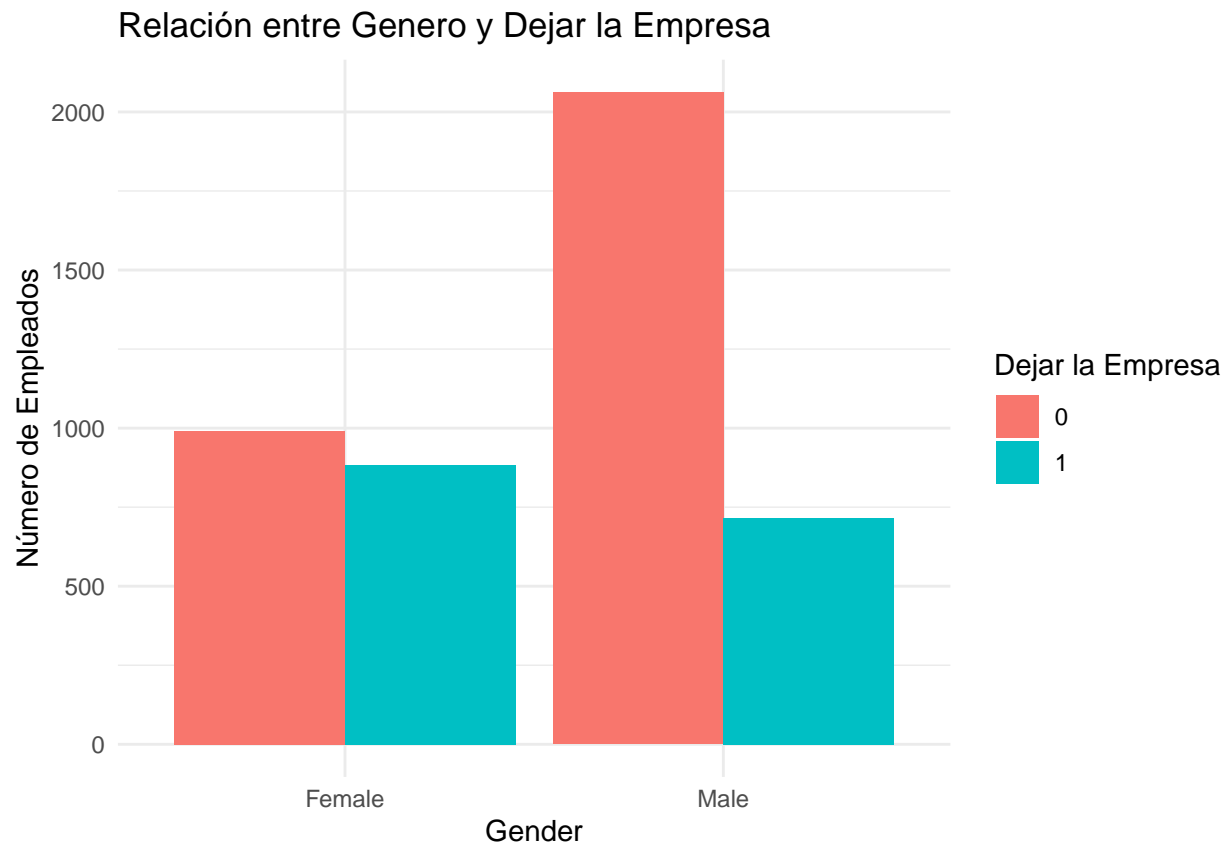
```
ggplot(data = df, aes(x = City, fill = factor(LeaveOrNot))) + # Define las variables para el eje X y e
  geom_bar(position = "dodge", stat = "count") + # Crea barras separadas para cada grupo en LeaveOrNot
  labs(title = "Relación entre Ciudad y Dejar la Empresa",
        x = "Ciudad",
        y = "Número de Empleados",
        fill = "Dejar la Empresa") + # Personaliza las etiquetas y título
  theme_minimal()
```



- Del grafico, se puede observar que la mayoría de los empleados que deciden dejar la empresa son de la ciudad de Bangalore y New Delhi.
- Por otro lado, la ciudad de Pune tiene una cantidad similar de empleados que deciden dejar la empresa y los que se quedan.

GRAFICO DE RELACION ENTRE VARIABLES “Gender” VS “LeaveOrNot”

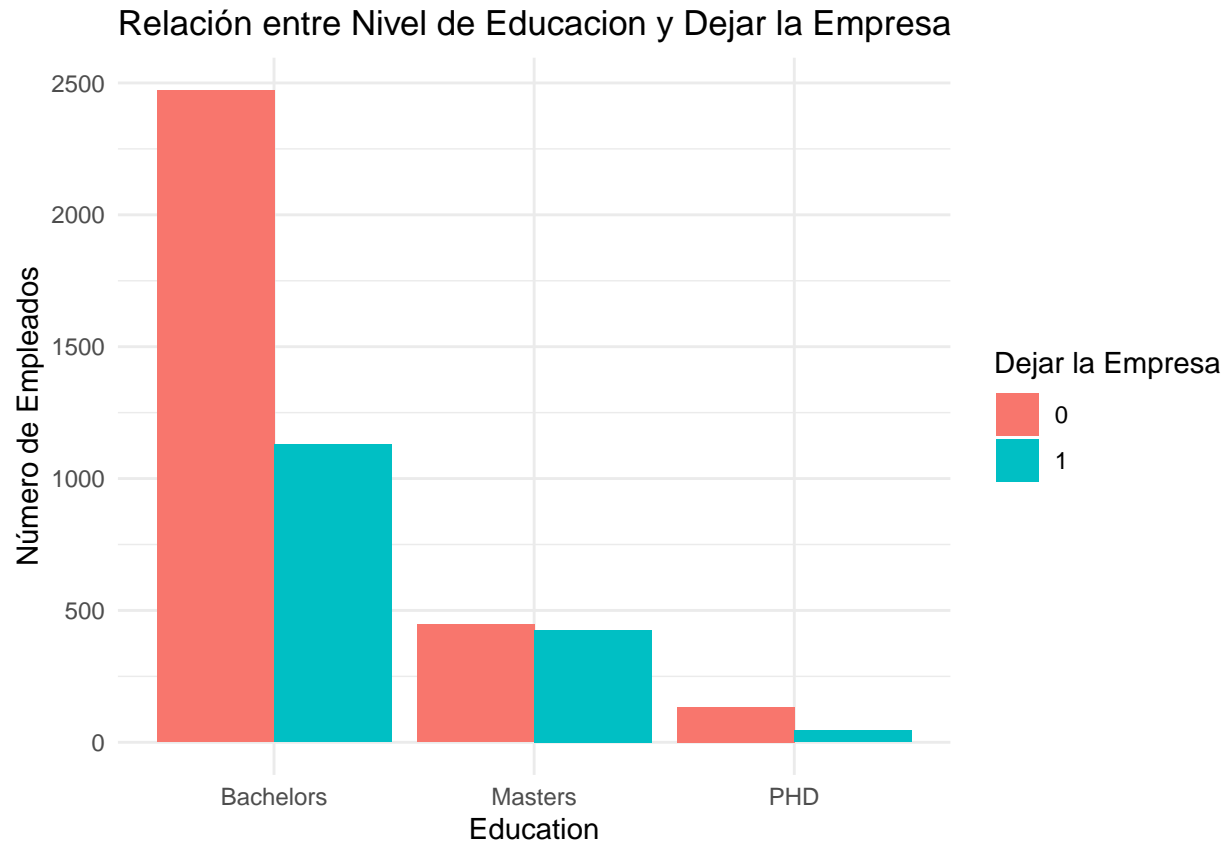
```
ggplot(data = df, aes(x = Gender, fill = factor(LeaveOrNot))) + # Define las variables para el eje X y
  geom_bar(position = "dodge", stat = "count") + # Crea barras separadas para cada grupo en LeaveOrNot
  labs(title = "Relación entre Genero y Dejar la Empresa",
        x = "Gender",
        y = "Número de Empleados",
        fill = "Dejar la Empresa") + # Personaliza las etiquetas y título
  theme_minimal()
```



- Del gráfico, se puede observar que la mayoría de los empleados que deciden dejar la empresa son de género masculino.
- Por otro lado, la cantidad de empleados que deciden quedarse en la empresa es similar para ambos géneros.

GRAFICO DE RELACION ENTRE VARIABLES “Education” VS “LeaveOrNot”

```
ggplot(data = df, aes(x = Education, fill = factor(LeaveOrNot))) + # Define las variables para el eje .
  geom_bar(position = "dodge", stat = "count") + # Crea barras separadas para cada grupo en LeaveOrNot
  labs(title = "Relación entre Nivel de Educacion y Dejar la Empresa",
        x = "Education",
        y = "Número de Empleados",
        fill = "Dejar la Empresa") + # Personaliza las etiquetas y título
  theme_minimal()
```



- Del grafico, se puede observar que la mayoría de los empleados que deciden dejar la empresa tienen un nivel de educación de “Bachelors”.
- Por otro lado, la cantidad de empleados que deciden quedarse en la empresa es similar para todos los niveles de educación Master y PHD.

PRUEBA DE HIPOTESIS

-Para determinar si estas diferencias son estadísticamente significativas, procederemos a realizar pruebas de Chi-cuadrado. Estas pruebas nos permitirán evaluar si las diferencias observadas en cada grupo son suficientes para sugerir una dependencia entre las variables categóricas y la decisión de dejar la empresa.

```
# PRUEBA DE CHI-CUADRADO PARA CIUDAD
city_table <- table(df$City, df$LeaveOrNot)
city_table
```

```
##
##           0    1
## Bangalore 1633 595
## New Delhi  791 366
## Pune       629 639
```

```
CH2_CITY <- chisq.test(city_table)
CH2_CITY
```

```
##
```

```
## Pearson's Chi-squared test
##
## data:  city_table
## X-squared = 206.16, df = 2, p-value < 2.2e-16
```

```
# PRUEBA DE CHI-CUADRADO PARA GENERO
gender_table <- table(df$Gender, df$LeaveOrNot)
gender_table
```

```
##
##           0      1
## Female  991  884
## Male   2062  716
```

```
CH2_GENDER <- chisq.test(gender_table)
CH2_GENDER
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  gender_table
## X-squared = 225.7, df = 1, p-value < 2.2e-16
```

```
# PRUEBA DE CHI-CUADRADO PARA EDUCACION
education_table <- table(df$Education, df$LeaveOrNot)
education_table
```

```
##
##           0      1
## Bachelors 2472 1129
## Masters   447  426
## PHD        134   45
```

```
CH2_EDUCATION <- chisq.test(education_table)
CH2_EDUCATION
```

```
##
## Pearson's Chi-squared test
##
## data:  education_table
## X-squared = 101.83, df = 2, p-value < 2.2e-16
```

- La prueba de Chi-cuadrado para la variable “City” muestra un valor de p-value < 2.2e-16, lo que indica que hay una relación significativa entre la ciudad de los empleados y su decisión de dejar la empresa.
- La prueba de Chi-cuadrado para la variable “Gender” muestra un valor de p-value < 2.2e-16, lo que indica que hay una relación significativa entre el género de los empleados y su decisión de dejar la empresa.
- La prueba de Chi-cuadrado para la variable “Education” muestra un valor de p-value < 2.2e-16, lo que indica que hay una relación significativa entre el nivel de educación de los empleados y su decisión de dejar la empresa.

CONCLUSIONES

- Los empleados de la ciudad de Bangalore y New Delhi tienen una mayor propensión a dejar la empresa en comparación con los de Pune.
- Los empleados de género masculino tienen una mayor propensión a dejar la empresa en comparación con los de género femenino.
- Los empleados con un nivel de educación de “Bachelors” tienen una mayor propensión a dejar la empresa en comparación con los de “Masters” y “PHD”.