PRA2 - Tipología y ciclo de vida de los datos: Limpieza y análisis de datos

Sergio Funes, David Ortiz

Junio de 2021

Contents

1. Descripción del dataset

Dataset recruitment data en Kaggle.

614 filas, 11 atributos originales:

- **Número de serie**. Número del registro.
- **Género**. Si el sujeto es hombre o mujer.
- **Experiencia con Python**. Si el sujeto tiene experiencia de programación en Python.
- **Años de experiencia**. Número de años de experiencia del sujeto.
- **Educación**. Si el sujeto es graduado universitario o no.
- **Prácticas**. Si el sujeto ha hecho unas prácticas o no.
- **Puntuación**. Puntuación del sujeto.
- **Salario** (*10e4). Salario del sujeto, en decenas de miles de rupias.
- **Histórico de ofertas**. Si el sujeto ha tenido ofertas laborales.

- **Localización**. Tipo de localización del sujeto en relación con la ciudad: urbana, semiurbana o rural.
- **Estado de contratación**. Si el sujeto fue contratado o no.

¿Importancia del dataset?

2. Integración y selección de los datos de interés

```
data <- read.csv("recruitment_decision_tree.csv", header = TRUE, stringsAsFactors = TRUE)
head(data)
```

```
##   Serial_no Gender Python_exp Experience_Years    Education Internship Score
## 1         1   Male        Yes                0     Graduate         No  5139
## 2         2   Male         No                1     Graduate         No  4583
## 3         3   Male         No                0     Graduate        Yes  3000
## 4         4   Male         No                0 Not Graduate         No  2583
## 5         5   Male        Yes                0     Graduate         No  6000
## 6         6   Male         No                2     Graduate        Yes  5417
##   Salary...10E4 Offer_History Location Recruitment_Status
## 1             0             1    Urban                  Y
## 2           128             1    Rural                  N
## 3            66             1    Urban                  Y
## 4           120             1    Urban                  Y
## 5           141             1    Urban                  Y
## 6           267             1    Urban                  Y
```

```
sapply(data, function(x) class(x))
```

```
##          Serial_no             Gender         Python_exp   Experience_Years
##          "integer"           "factor"           "factor"          "integer"
##          Education         Internship              Score      Salary...10E4
##           "factor"           "factor"          "integer"          "integer"
##      Offer_History           Location Recruitment_Status
##          "integer"           "factor"           "factor"
```

```
data <- data[, -(1)]
```

3. Limpieza de los datos

3.1. Ceros y elementos vacíos

```
sapply(data, function(x) sum(is.na(x)))
```

```
##             Gender         Python_exp   Experience_Years          Education
##                  0                  0                 15                  0
##         Internship              Score      Salary...10E4      Offer_History
##                  0                  0                 21                 50
##           Location Recruitment_Status
##                  0                  0
```

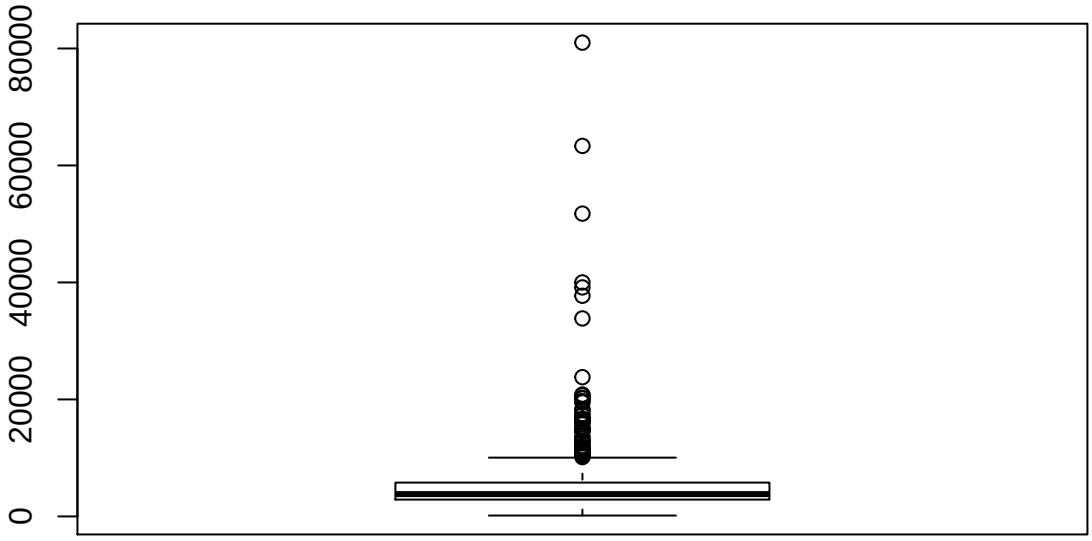
```
suppressWarnings(suppressMessages(library(VIM)))

data$Experience_Years <- kNN(data)$Experience_Years
data$Salary...10E4 <- kNN(data)$Salary...10E4
data$Offer_History <- kNN(data)$Offer_History
```

```
sapply(data, function(x) sum(is.na(x)))
```

```
##             Gender         Python_exp   Experience_Years          Education
##                  0                  0                  0                  0
##         Internship              Score      Salary...10E4      Offer_History
##                  0                  0                  0                  0
##           Location Recruitment_Status
##                  0                  0
```

3.2. Valores extremos

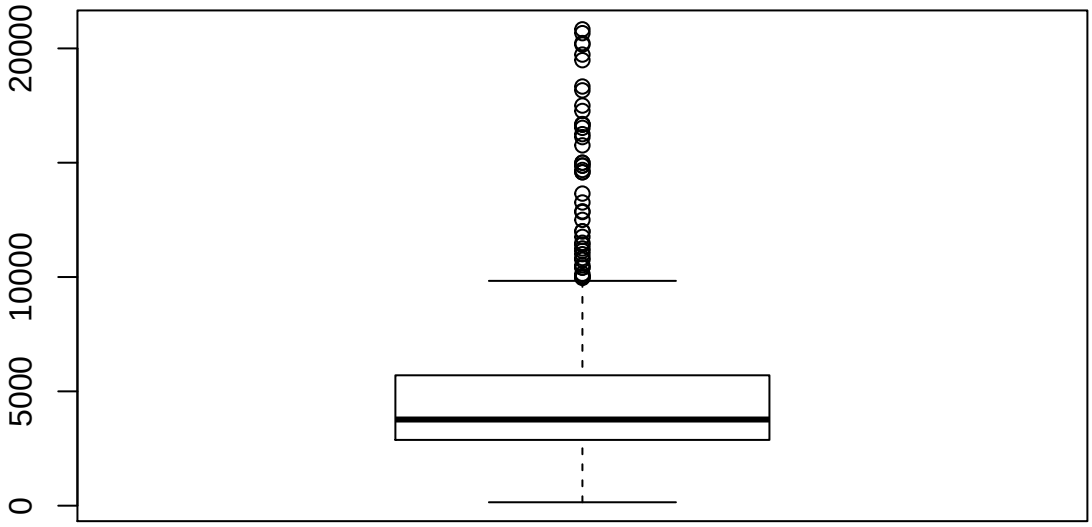
```
boxplot(data$Score)
```



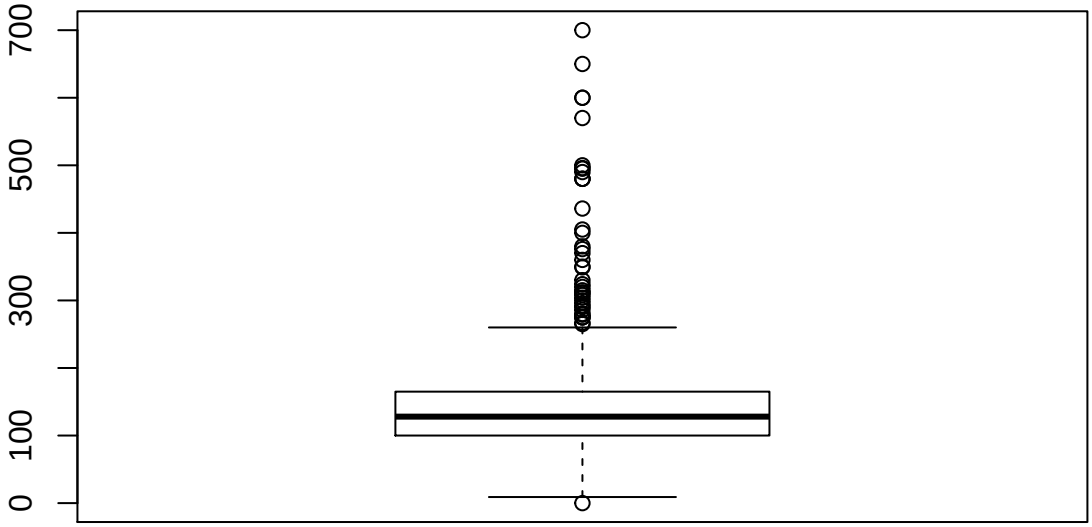

```
boxplot.stats(data$Score)$out
```

```
##  [1] 12841 12500 11500 10750 13650 11417 14583 10408 23803 10513 20166 14999
## [13] 11757 14866 39999 51763 33846 39147 12000 11000 16250 14683 11146 14583
## [25] 20667 20233 15000 63337 19730 15759 81000 14880 12876 10416 37719 16692
## [37] 16525 16667 10833 18333 17263 20833 13262 17500 11250 18165 10139 19484
## [49] 16666 16120 12000
```

```
data$Score[data$Score > 23000] <- NA

data$Score <- kNN(data)$Score

boxplot(data$Score)
```



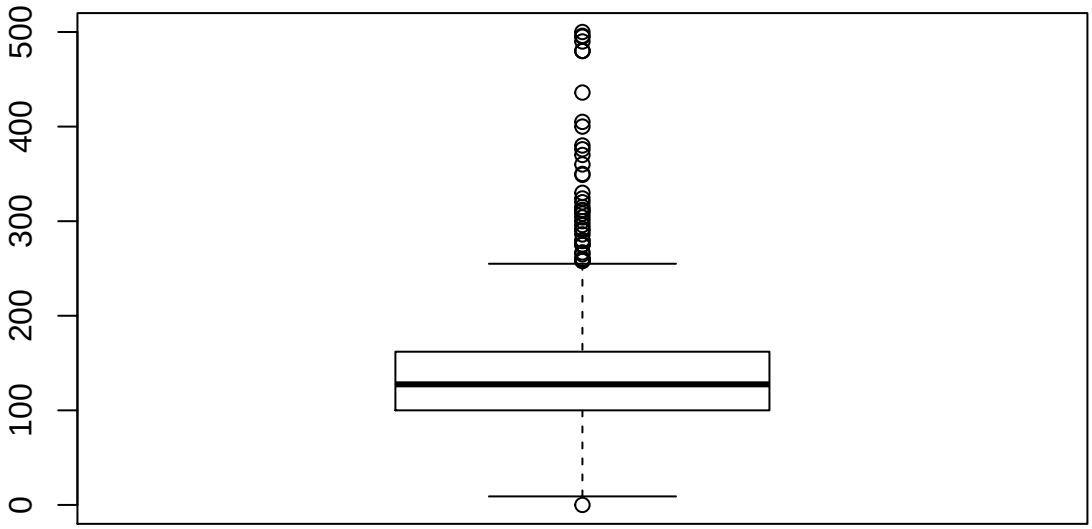

```
boxplot(data$Salary...10E4)
```




```
boxplot.stats(data$Salary...10E4)$out
```

```
##  [1]   0 267 349 315 320 286 312 265 370 650 290 600 275 700 495 280 279 304 330
## [20] 436 480 300 376 490 308 570 380 296 275 360 405 500 480 311 480 400 324 600
## [39] 275 292 350 496
```

```
data$Salary...10E4[data$Salary...10E4 > 500] <- NA

data$Salary...10E4 <- kNN(data)$Salary...10E4

boxplot(data$Salary...10E4)
```



3.3. Exportación de los datos preprocesados

```
write.csv(data, "recruitment_decision_tree_clean.csv")
```

4. Análisis de los datos

4.1. Selección de datos

Pasamos a la selección de las variables de interés para su análisis estadístico posterior. Encontramos aquí las variables categóricas más relevantes.

```
# Per gender
data.women <- data[data$Gender == "Female", ]
data.men <- data[data$Gender == "Male", ]

# Per location type
data.urban <- data[data$Location == "Urban", ]
data.semiurban <- data[data$Location == "Semiurban", ]
data.rural <- data[data$Location == "Rural", ]

# Per education level
data.graduated <- data[data$Education == "Graduate", ]
data.not_graduated <- data[data$Education == "Not Graduate", ]

# Target var: recruitment status
data.recruited <- data[data$Recruitment_Status == "Y", ]
data.not_recruited <- data[data$Recruitment_Status == "N", ]
```

4.2. Comprobación de normalidad y homogeneidad de la varianza

Se comprueba ahora la normalidad de las variables cuantitativas, mediante el test de normalidad de Anderson-Darling. Si el p-valor de es superior al nivel de significancia de $\alpha = 0,05$, entonces podemos concluir que la variable en cuestión es normal.

```
library(nortest)

num_vars = c("Experience_Years", "Score", "Salary...10E4")
```

```r
alpha = 0.05

for (col in num_vars){
  p_value = ad.test(data[, col])$p.value
  if (p_value > alpha){
    print(paste0(col, ": Normal (p = ", p_value, ")"))
  } else {
    print(paste0(col, ": Not normal (p = ", p_value, ")"))
  }
}
```

```
## [1] "Experience_Years: Not normal (p = 3.7e-24)"
## [1] "Score: Not normal (p = 3.7e-24)"
## [1] "Salary...10E4: Not normal (p = 3.7e-24)"
```

Por tanto vemos que las tres variables numéricas siguen una distribución no normal.

Para estudiar la homogeneidad de las varianzas aplicamos el test de Fligner-Killeen. Se pretende comprobar esta en la relación del género de los candidatos con el salario asociado a estos. Para este test, la hipótesis nula, H_0, sostiene la igualdad de varianzas para ambos grupos. Por su parte, H_1 supondría varianzas diferentes.

Si el p-valor es superior a un nivel de significancia del 0,05, se aceptará dicha hipótesis nula y por tanto la igualdad de varianzas.

```r
fligner.test(Salary...10E4 ~ Gender, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Salary...10E4 by Gender
## Fligner-Killeen:med chi-squared = 3.5148, df = 2, p-value = 0.1725
```

Puesto que el p-valor es mayor a 0,05, se acepta la hipótesis nula y por tanto se asume la igualdad de varianzas para ambas muestras.

4.3. Pruebas estadísticas

5. Representación de resultados

6. Resolución y conclusiones

7. Código