PRACTICA 2: LIMPIEZA Y VALIDACIÓN DE LOS DATOS

Sabela de La Torre y Gervasio Cuenca

31 de mayo 2019

Contents

---

1 Descripción del dataset

¿Por qué es importante y qué pregunta / problema pretende responder?

El dataset escogido describe la probabilidad de ser aceptado en la universidad, en función de una serie de parámetros basados en la actividad escolar de los candidatos.

El dataset se ha extraído de kaggle, se puede acceder a él desde el siguiente link: https://www.kaggle.com/mohansacharya/graduate-admissions/downloads/graduate-admissions.zip/2

Con este dataset pretendemos averiguar la probabilidad que tiene un alumno de ser aceptado en la universidad basándonos en sus cualificaciones académicas. Consideramos que puede ser un estudio interesante, ya que puede servir como herramienta para los orientadores escolares para guiar a los alumnos en sus elecciones de estudios superiores.

También comprobaremos si hay diferencia en la probabilidad de ser admitidos entre los alumnos que optan a universidades con calificación alta y aquellos que optan a universidades con calificación baja.

El dataset consta de 500 registros, correspondientes a 500 alumnos y 9 atributos, a continuación describimos cada uno de estos atributos:

- Serial Nº: Identificador de alumno.
- GRE score: Nota obtenida en el Grade Record Examinations, sería el equivalente a la selectividad española.
- TOEFL Score: Test de inglés como lengua extranjera.
- University rating: Clasificación de la Universidad. (1-5)
- SOP: Declaración de propósito, dónde el candidato explica por qué es un buen candidato para ser admitido en la universidad. (1-5)
- LOR: Carta de recomendación. (1-5)
- CGPA: Cumulative Grade Point Average (1-9)
- Research: Experiencia en investigación (0,1)
- Chance.of.Admit: Confianza del encuestado en ser aceptado. (0-1)

---

2 Integración y selección de los datos de interés a analizar

Los datos se encuentran en formato csv, realizaremos la carga de todos los registros para su posterior tratamiento.

```r
## Realizamos la carga de los datos
alumnos <- read.csv("Admission_Predict_Ver1.1.csv", header=TRUE)

## Comprobamos los datos cargados y los tipos de variables asignados
str(alumnos)
```

```
## 'data.frame':    500 obs. of  9 variables:
##  $ Serial.No.       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ GRE.Score        : int  337 324 316 322 314 330 321 308 302 323 ...
##  $ TOEFL.Score      : int  118 107 104 110 103 115 109 101 102 108 ...
##  $ University.Rating: int  4 4 3 3 2 5 3 2 1 3 ...
##  $ SOP              : num  4.5 4 3 3.5 2 4.5 3 3 2 3.5 ...
##  $ LOR              : num  4.5 4.5 3.5 2.5 3 3 4 4 1.5 3 ...
##  $ CGPA             : num  9.65 8.87 8 8.67 8.21 9.34 8.2 7.9 8 8.6 ...
##  $ Research         : int  1 1 1 1 0 1 1 0 0 0 ...
##  $ Chance.of.Admit  : num  0.92 0.76 0.72 0.8 0.65 0.9 0.75 0.68 0.5 0.45 ...
```

Se puede comprobar que los datos asignados a las variables del nuestro set de datos son los correctos. Por otro lado, revisando los datos, vemos que no necesitaremos la columna de `Serial.no`, ya que no es necesario para nuestro estudio.

```r
##Eliminamos la primera columna
alumnos_estudio <- alumnos[,-1]


##Comprobamos las variables que nos han quedado en el set de de datos y sus tipos
sapply(alumnos_estudio, function(x) class(x))
```

```
##         GRE.Score       TOEFL.Score University.Rating               SOP
##         "integer"         "integer"         "integer"         "numeric"
##               LOR              CGPA          Research   Chance.of.Admit
##         "numeric"         "numeric"         "integer"         "numeric"
```

3 Limpieza de los datos

3.1 Ceros y elementos vacíos

Primero buscamos si hay valors vacíos:

```r
colSums(is.na(alumnos_estudio))
```

```
##         GRE.Score       TOEFL.Score University.Rating               SOP
##                 0                 0                 0                 0
##               LOR              CGPA          Research   Chance.of.Admit
##                 0                 0                 0                 0
```

y vemos que no tenemos ninguno.

Ahora analizaremos los datos que tenemos en cada una de las variables (rango, media, mediana, mínimo, máximo y cuartiles) mediante la función `summary`:

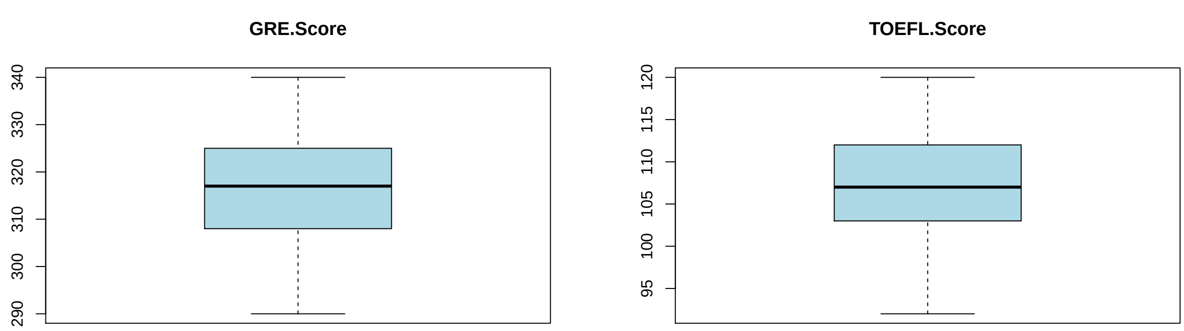
```r
summary(alumnos_estudio)
```

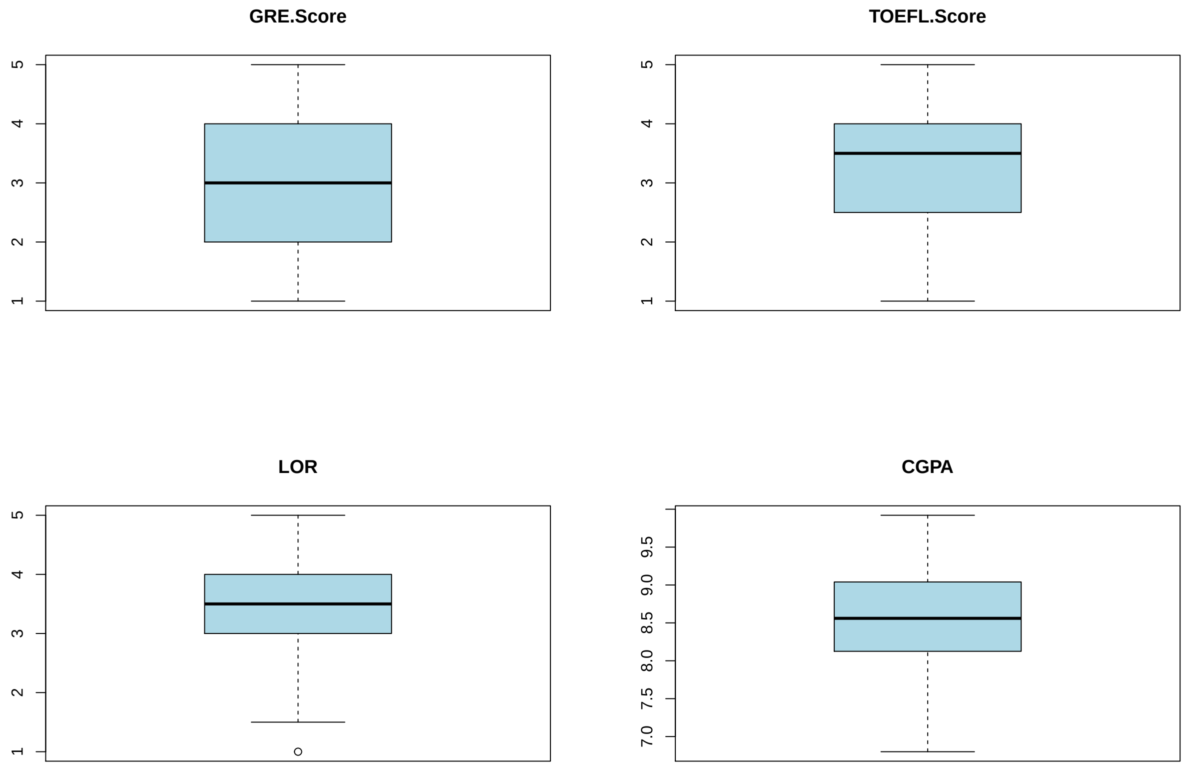
```
##    GRE.Score      TOEFL.Score    University.Rating      SOP
##  Min.   :290.0   Min.   : 92.0   Min.   :1.000     Min.   :1.000
##  1st Qu.:308.0   1st Qu.:103.0   1st Qu.:2.000     1st Qu.:2.500
##  Median :317.0   Median :107.0   Median :3.000     Median :3.500
##  Mean   :316.5   Mean   :107.2   Mean   :3.114     Mean   :3.374
##  3rd Qu.:325.0   3rd Qu.:112.0   3rd Qu.:4.000     3rd Qu.:4.000
##  Max.   :340.0   Max.   :120.0   Max.   :5.000     Max.   :5.000
##       LOR             CGPA          Research    Chance.of.Admit
##  Min.   :1.000   Min.   :6.800   Min.   :0.00   Min.   :0.3400
##  1st Qu.:3.000   1st Qu.:8.127   1st Qu.:0.00   1st Qu.:0.6300
##  Median :3.500   Median :8.560   Median :1.00   Median :0.7200
##  Mean   :3.484   Mean   :8.576   Mean   :0.56   Mean   :0.7217
##  3rd Qu.:4.000   3rd Qu.:9.040   3rd Qu.:1.00   3rd Qu.:0.8200
##  Max.   :5.000   Max.   :9.920   Max.   :1.00   Max.   :0.9700
```

Vemos que el valor 0 solamente lo encontramos en la variable `Research`, cosa que ya sabíamos porque se trata de una variable binaria.

3.2 Identificación y tratamiento de valores extremos

Una herramienta gràfica muy útil para la detección de valores extremos es el diagrama de caja. Este se basa en los valores de los cuartiles. Usaremos la función `boxplot` para dibujar los diagramas para cada una de las variables:

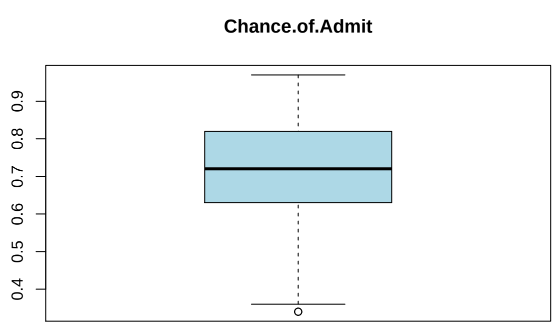




Vemos que en la variable `LOR` (carta de recomendación) tenemos un único *outlier* correspondiente al valor 1:

```
boxplot.stats(alumnos_estudio$LOR)$out
```

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



En la variable `Change.of.Admit` encontramos dos *outliers* con valor 0.34:

```
boxplot.stats(alumnos_estudio$Chance.of.Admit)$out
```

```
## [1] 0.34 0.34
```

Observando el conjunto de datos, vemos que estos valores son completamente aceptables y, por tanto, no son *outliers* reales. Por último, dado que la variable `Research` es una variable binaria y puede considerarse categórica, no tiene sentido representarla mediante un boxplot y por ese motivo la hemos otimido.

4 Análisis de los datos

Selección de los grupos de datos que se quieren analizar/comparar

Porcedemos a realizar algunas agrupaciones de datos que nos resultan interesantes, aunque no todos serán utilitzados para hacer cálculos estadísticos.

- Agrupación por alumnos que optan a universidades con calificación baja vs calificación alta
- Agrupación por alumnos con experiencia en investigación (en apartados posteriores veremos que la variable `Research` no es relevante a la hora de determinar la proabilidad de admisión)
- Agrupación por alumnos por tipo de universidad

```
# Agrupación por alumnos que optan a universidades de calificación baja vs calificación alta
alumnos.universidades.calif.alta <- alumnos_estudio[alumnos_estudio$University.Rating >= 3,]
alumnos.universidades.calif.baja <- alumnos_estudio[alumnos_estudio$University.Rating < 3,]

# Agrupación por alumnos con experiencia en investigación
alumnos.investigadores <- alumnos_estudio[alumnos_estudio$Research == 1,]
alumnos.no.investigadores <- alumnos_estudio[alumnos_estudio$Research == 0,]

# Agrupación por tipo de universidad
alumnos.universidades.top <- alumnos_estudio[alumnos_estudio$University.Rating == 5,]
alumnos.universidades.Buenas <- alumnos_estudio[alumnos_estudio$University.Rating == 4,]
alumnos.universidades.Medias <- alumnos_estudio[alumnos_estudio$University.Rating == 3,]
alumnos.universidades.Acepables <- alumnos_estudio[alumnos_estudio$University.Rating == 2,]
alumnos.universidades.Flojas <- alumnos_estudio[alumnos_estudio$University.Rating == 1,]
```

4.1 Comprobación de la normalidad y homogeneidad de la varianza

4.1.1 Comprobación de la normalidad

Comprobamos si los datos siguen una distribución normal mediante la función `shapiro.test`: si *p-value* ≤ 0.05 se rechaza la hipótesis nula y se concluye que los datos **no** siguen una distribución normal.

```
alpha <- 0.05
col.names = colnames(alumnos_estudio)
var.no.normales <- c()
for (i in 1:ncol(alumnos_estudio)) {
  # Aplicamos el test Shapiro-Wilk
  p_val = shapiro.test(alumnos_estudio[,i])$p.value
  if (p_val <= alpha) {
    var.no.normales <- c(var.no.normales, col.names[i])
  }
}
cat("Variables que no siguen una distribución normal: ")
```

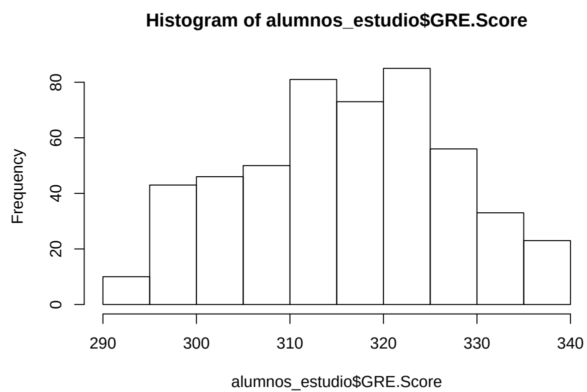
```
## Variables que no siguen una distribución normal:
```

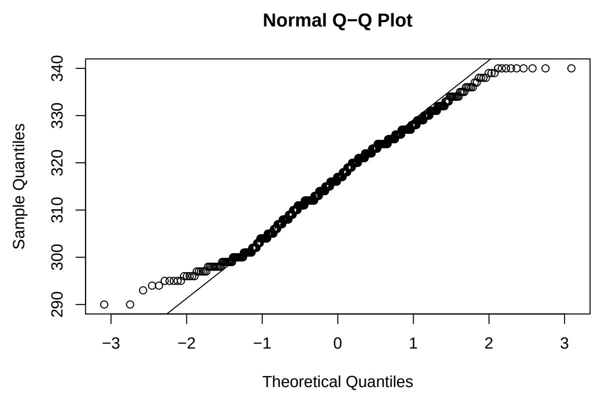
```
cat(var.no.normales, sep=", ")
```

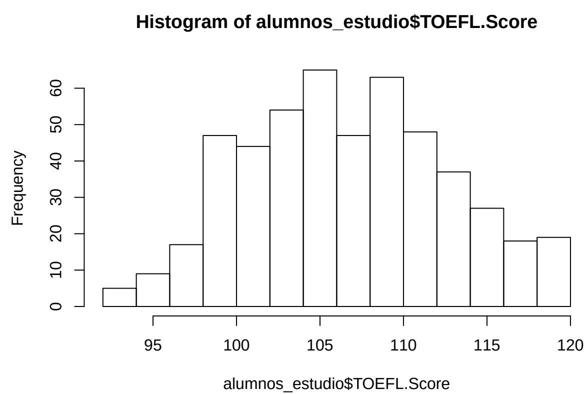
```
## GRE.Score, TOEFL.Score, University.Rating, SOP, LOR, CGPA, Research, Chance.of.Admit
```

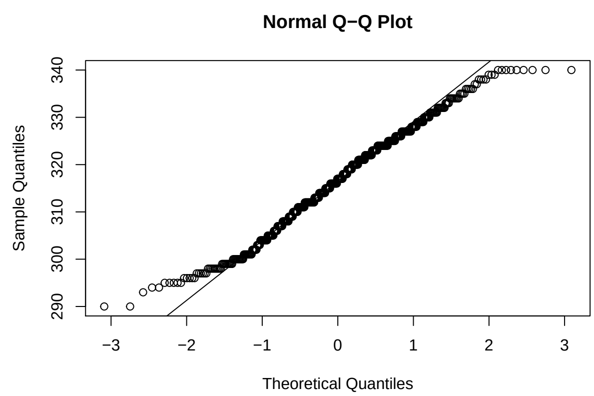
Por lo tanto, **ninguna** de las varaibles de nuestro conjunto de datos sigue una distribución normal. Ahora bien, por el **teorema del límite central** sabemos que si la muestra es suficientemente grande (n>30), la distribución de la media de cualquier conjunto de datos se parece a una normal. Así pues, podremos aplicar tests paramétricos pese a que nuestros datos no siguen una distribución normal.

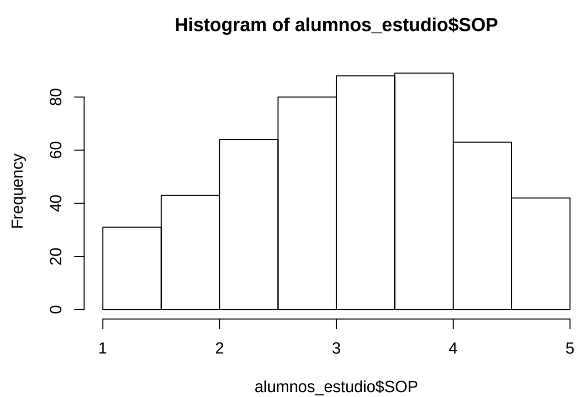
Complementaremos los resultados obtenidos con `shapiro.test` con histogramas (`hist`) y gráficos Q-Q (`qqnorm` y `qqline`) para representar de manera visual si aquellas variables que pueden tomar valores continuos (por tanto, todas menos `Research` y `University.Rating`) siguen una distribución normal:

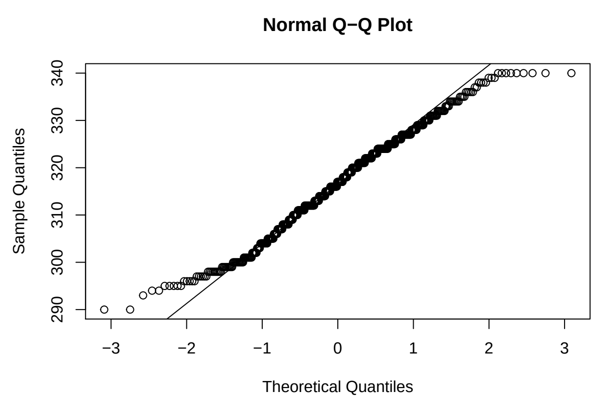


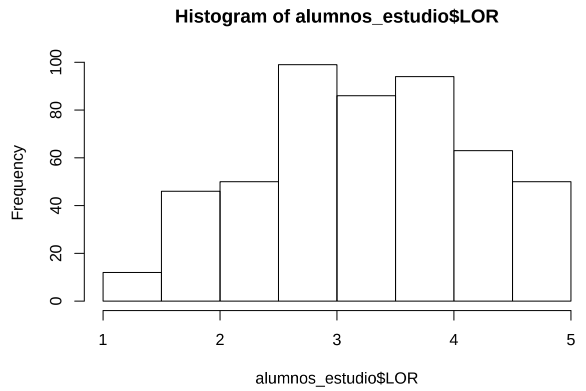


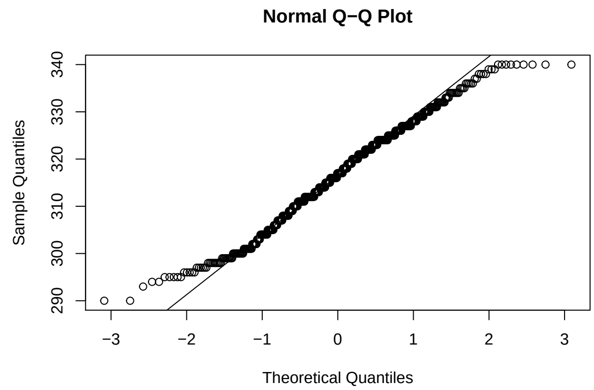


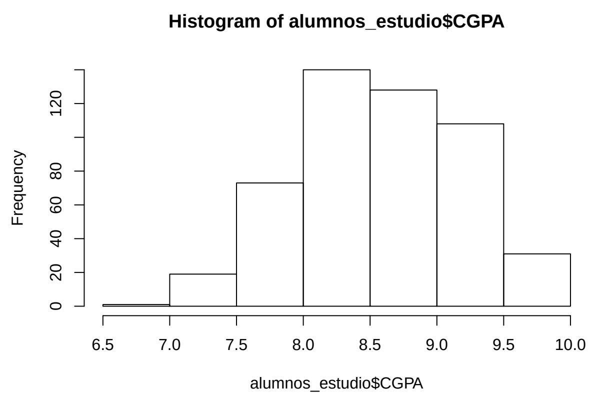


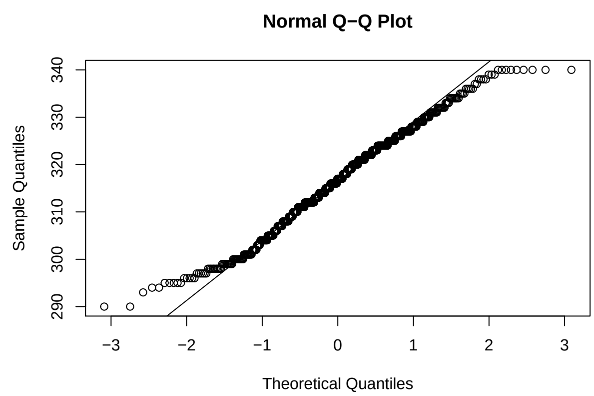


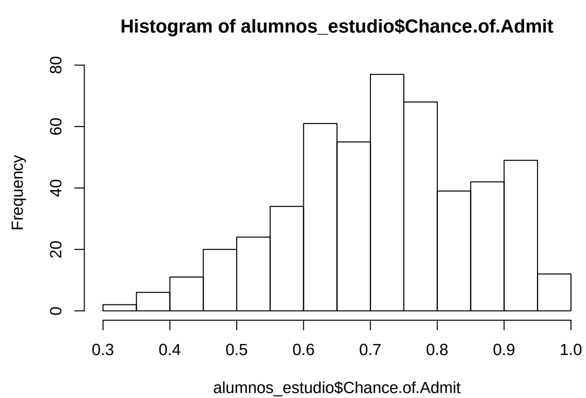


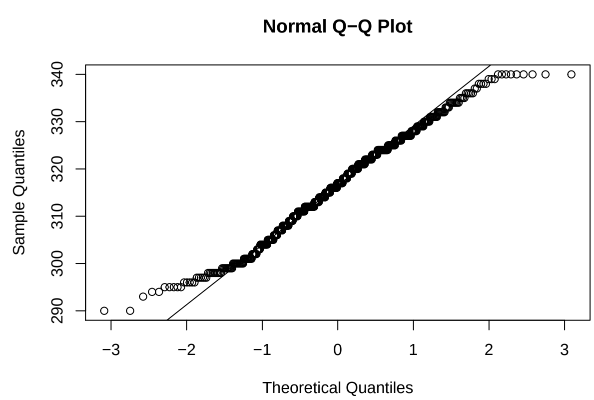

Histogram of alumnos_estudio$LOR
Frequency
alumnos_estudio$LOR

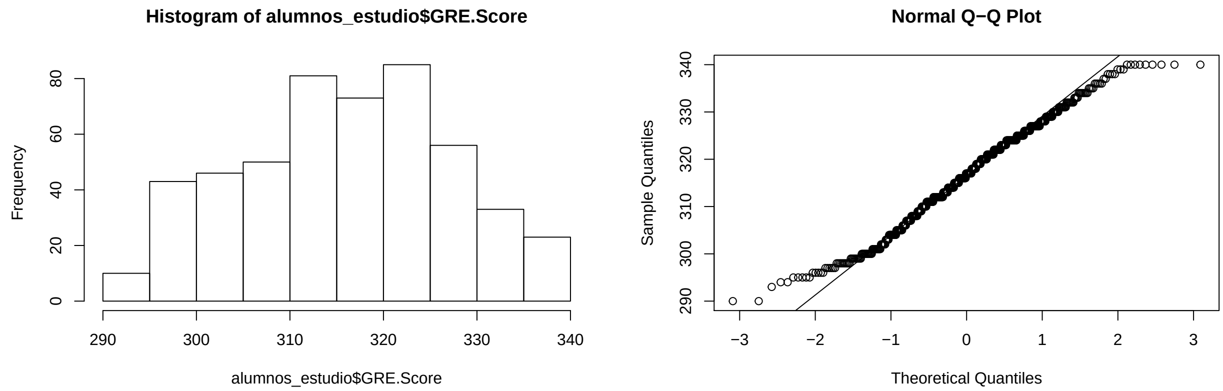

Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles

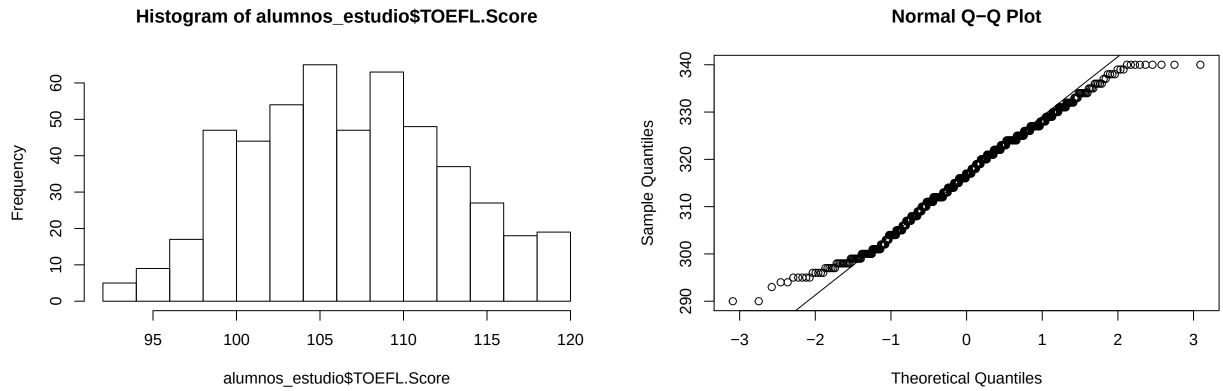

Histogram of alumnos_estudio$CGPA
Frequency
alumnos_estudio$CGPA


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles


Histogram of alumnos_estudio$Chance.of.Admit
Frequency
alumnos_estudio$Chance.of.Admit


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles





Como podemos observar, se aproximan bastante a la normal pero no del todo, por lo tanto, se confirma el resultado obtenido mediante el test de Shapiro-Wilk: las variables no siguen una distribución normal.

4.1.2 Homogeneidad de la varianza

Estudiaremos la homocedasticidad, o igualdad de varianzas, entre los grupos formados por alumnos con experiencia en investigación frente a los que no:

```
fligner.test(Chance.of.Admit ~ Research, data = alumnos_estudio)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Chance.of.Admit by Research
## Fligner-Killeen:med chi-squared = 2.0601, df = 1, p-value = 0.1512
```

En este test, la hipótesis nula es que las varianzas de los dos grupos son iguales, por tant, dado que *p-value* > 0.05, no podemos rechazar la hipótesis nula y **no** podemos afirmar que las varianzas sean significativamente diferentes.

Repetimos el estudio para la clasificación de las universidades:

```
fligner.test(Chance.of.Admit ~ University.Rating, data = alumnos_estudio)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Chance.of.Admit by University.Rating
```

```
## Fligner-Killeen:med chi-squared = 15.887, df = 4, p-value =
## 0.003175
```

Partimos de la misma hipótesis que el caso anterior, en este caso *p-value* < 0.05, por lo tento podemos rechazar la hipótesis nula y **podemos afirmar** que las varianzas sean significativamente diferentes.

Dado que en el siguiente apartado realizaremos un contraste de hipótesis entre los grupos que alumnos que optan a universidades calificadas como altas y los alumnos que optan a aquellas calificadas como bajas, procederemos a comparar las varianzas entre estos dos grupos:

```
fligner.test(Chance.of.Admit ~ University.Rating >= 3, data = alumnos_estudio)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Chance.of.Admit by University.Rating >= 3
## Fligner-Killeen:med chi-squared = 2.4304, df = 1, p-value = 0.119
```

Vemos que obtenemos un *p-value* $= 0.119$, por tanto, al mayor que $\alpha = 0.05$, no podemos afirmar que las varianzas sean significativamente diferentes.

4.2 Aplicación de pruebas estadísticas

Dado que nuestro conjunto de datos contiene más de 30 muestras, ya hemos visto que por el **teorema del límite central** podemos aplicar tests paramétricos aunque nuestros datos no sigan una distribución normal pero deberemos comprobar siempre la igualdad de varianzas. De no cumplirse, tendremos que aplicar un test no paramétrico.

4.2.1 ¿Es menor la confianza en ser admitido según la universidad a la que optan los alumnos?

En esta prueba buscaremos si la confianza en ser admitido, `Chance.of.Admit`, es menor entre los alumnos que optan a universidades calificadas como bajas, es decir, `University.Rating<3`, y la confianza entre los que optan a aquellas calificadas como altas, `University.Rating>=3`.

En este caso, la hipótsis nula, H_0, es que la confiança media de ambas poblaciones, μ_1 y μ_2, es igual y la hipótesis alternativa, H_1, que $\mu_1 < \mu_2$ (bilateral), donde μ_1 es la confiança media de los alumnos que optan por una universidad calificada como baja y μ_2 el otro grupo.

$$\begin{cases} H_0: & \mu_1 = \mu_2 \\ H_1: & \mu_1 < \mu_2 \end{cases}$$

Dado que en el apartado anterior hemos visto que la clasificación por universidades cumple la igualdad de varianzas (o, más bien, no podemos afirmar que sean significativamente diferentes) y que la variable no sigue una distribución normal, seremos conservadores y aplicaremos un test no paramétrico como la prueba de Mann-Whitney para datos independientes (podríamos haber optado por aplicar un test parámetrico y ser menos conservadores ya que el **teorema del límite central** así nos lo permite).

Por lo tanto usaremos la función `wilcox.test` para realizar el contraste de hipótesis usando un valor $\alpha = 0.05$:

```
wilcox.test(
  alumnos.universidades.calif.baja$Chance.of.Admit,
  alumnos.universidades.calif.alta$Chance.of.Admit,
  alternative = "less",
  conf.level = 0.05
)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
```

```
## data:  alumnos.universidades.calif.baja$Chance.of.Admit and alumnos.universidades.calif.alta$Chance.o
## W = 8869, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
```

Podemos ver que $p\text{-}value = 2.2e^{-16} < 0.05$, por tanto, podemos **rechazar** la hipótesis nula y afirmar que la confianza en ser admitidos de los alumnos que optan por universidades de calificación baja és **significativamente inferior** a la confianza del otro grupo de alumnos. Este resultado es en parte sorprendente en parte esperado ya que la lógica hace pensar que las universidades con calificación baja tienen criterios de acceso menos restrictivos (lo cual haría pensar que los alumnos tendrían mayor confianza en ser admitidos) pero a la vez si un alumno opta por una universidad de más bajo nivel es posible que sea porque no es un buen estudiante y, por consecuencia, confíe menos en ser admitido.

4.2.2 ¿Qué variables afectan más a la posibilidad de ser admitido en una universidad?

Para intentar contestar a esta pregunta, estudiaremos la correlación entre las diferentes variables de nuetros modelo con la probabilidad de ser admitido. Para ello calcularemos el coeficiente de correlación que mide la asociación entre dos variables. Los posibles valores que puede tomar el coeficiente de correlación varia entre -1 y 1, donde el valor de los extremos indican una correlación perfecta u el 0 indica la ausencia de correlación. EL signo es positivo cuando ambas variables se incrementan o disminuyen simultaneamente, el signo es negativo cuando los valores elevados de una variable se asocian a valores pequeños de otra.

En este caso untilizaremos la correlación de Spearman como test no parámetrico ya que las variables no siguen una distribución normal, aunque sería válido usar la correlación de Pearson, por el **teorema del límite central** citado con anterioridad.

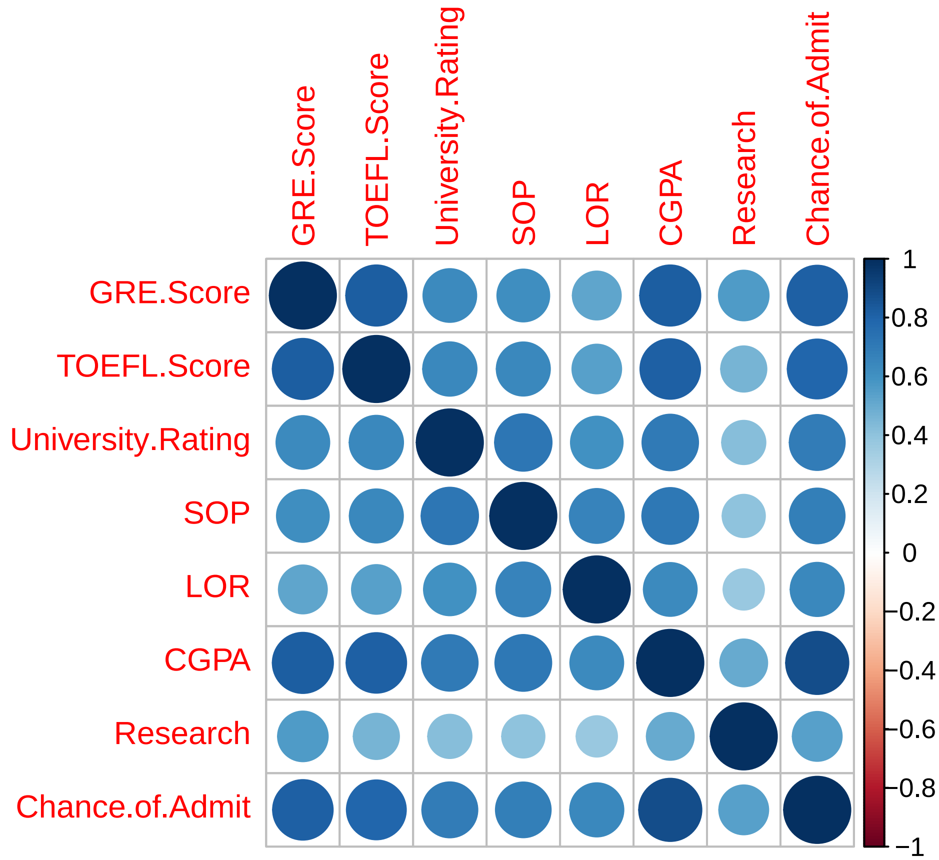
```r
alumnos.correlacion <- matrix(nc = 2, nr = 0)
colnames(alumnos.correlacion) <- c("estimate", "p-value")

## Realizamos el cálculo de la correlación
for (i in 1:(ncol(alumnos_estudio) - 1)) {
  test = cor.test(alumnos_estudio[, i],
                  alumnos_estudio[, length(alumnos_estudio)],
                  method = "spearman",
                  exact = FALSE)
  estimado = test$estimate
  p_valor = test$p.value

  ##Añadimos el valor a la matriz
  valores = matrix(ncol = 2, nrow = 1)
  valores[1][1] = estimado
  valores[2][1] = p_valor
  alumnos.correlacion <- rbind(alumnos.correlacion, valores)
  rownames(alumnos.correlacion)[nrow(alumnos.correlacion)] <-
    colnames(alumnos_estudio)[i]
}

print(alumnos.correlacion)
```

```
##                    estimate       p-value
## GRE.Score         0.8222012 5.734552e-124
## TOEFL.Score       0.7936342 1.504956e-109
## University.Rating 0.7037425  5.889501e-76
## SOP               0.7027994  1.133632e-75
## LOR               0.6436271  7.989633e-60
## CGPA              0.8887857 7.372294e-171
## Research          0.5657155  1.224593e-43
```

Analizando los resultados vemos que las dos variables que tienen una mayor correlación con la posibilidad de ser admitido son `CGPA` y `GRE.Score`. Hemos añadido el *p-value* porque nos puede dar el peso estadístico de la correlación obtenida.

También podemos representar gráficamente esta correlación entre las diferentes variables mediante `corrplot`:

```
corr.res <- cor(alumnos_estudio)
corrplot(corr.res, method="circle")
```



Como ya hemos visto en el cálculo con `corr.test`, las variables que más influyen en `Chance.of.Admit` son `CGPA`, `GRE.Score` y `TOEFL.Score`.

4.2.3 Modelo de regresión lineal, para predecir la posibilidad de ser admitido en una Universidad.

La regresion lineal es un modelo matemático que tiene como objetivo aproximar la relación de dependencia lineal entre una variable dependiente y una o una serie de variables independientes.

La regresión lineal puede ser simple o múltiple en función de las variables independientes que se incluyan en la fórmula que se introduce como argumento.

Para intentar predecir la poibidad de ser adminitido en la universidad, utilizaremos las variables con correlación superior a 0.7, en este caso todas menos `LOR` y `Research.`

Para ello, prepararemos dos set de datos, uno con el 85% de los datos, que usaremos para entrenar los modelos y escoger el que mejor resultado de, y el segundo con el 15% restante como test de pruebas, para predecir el campo que buscamos y compararlo con el valor real.

```
## Creamos los sets de datos.
h <- holdout(alumnos_estudio$University.Rating, ratio = 0.85, mode="statified")
```

```r
alumnos_train <- alumnos_estudio[h$tr,]
alumnos_test <- alumnos_estudio[h$ts,]

##Generamos los diferentes modelos.

alumnos_m1 <- lm(Chance.of.Admit ~ CGPA + GRE.Score + TOEFL.Score , data = alumnos_train)
alumnos_m2 <- lm(Chance.of.Admit ~ CGPA + GRE.Score + University.Rating , data = alumnos_train)
alumnos_m3 <- lm(Chance.of.Admit ~ CGPA + GRE.Score + SOP , data = alumnos_train)
alumnos_m4 <- lm(Chance.of.Admit ~ CGPA + TOEFL.Score + University.Rating , data = alumnos_train)
alumnos_m5 <- lm(Chance.of.Admit ~ CGPA + TOEFL.Score + SOP , data = alumnos_train)
alumnos_m6 <- lm(Chance.of.Admit ~ CGPA + University.Rating + SOP , data = alumnos_train)
alumnos_m7 <- lm(Chance.of.Admit ~ GRE.Score + TOEFL.Score + University.Rating , data = alumnos_train)
alumnos_m8 <- lm(Chance.of.Admit ~ GRE.Score + TOEFL.Score + SOP , data = alumnos_train)
alumnos_m9 <- lm(Chance.of.Admit ~ TOEFL.Score + University.Rating + SOP , data = alumnos_train)

regresion <- matrix(c(1 , summary(alumnos_m1)$r.squared,
                      2 , summary(alumnos_m2)$r.squared,
                      3 , summary(alumnos_m3)$r.squared,
                      4 , summary(alumnos_m4)$r.squared,
                      5 , summary(alumnos_m5)$r.squared,
                      6 , summary(alumnos_m6)$r.squared,
                      7 , summary(alumnos_m7)$r.squared,
                      8 , summary(alumnos_m8)$r.squared,
                      9 , summary(alumnos_m9)$r.squared),ncol = 2, byrow = TRUE)
colnames(regresion) <- c("Modelo", "Bondad")

knitr::kable(regresion)  %>%
kable_styling("striped", full_width = F)
```

Modelo	Bondad
1	0.7965212
2	0.7992814
3	0.7988146
4	0.7942960
5	0.7919859
6	0.7829520
7	0.7368794
8	0.7404319
9	0.7069258

```r
## Usamos el modelo que mejor resultado ha dado.
Modelo = regresion[which.max(regresion[,2]),1]
Modelo
```

```
## Modelo
##      2
```

Vemos que la mayor bondad la encontramos en el modelo indicado aunque diferentes ejecuciones producen diferentes resultados.

A continuación, predecimos los valores usando el conjunto de datos de test y los comparamos con los reales:

```r
if (Modelo == 1) Prediccion<-predict(alumnos_m1, alumnos_test, type="response")
if (Modelo == 2) Prediccion<-predict(alumnos_m2, alumnos_test, type="response")
```

```r
if (Modelo == 3) Prediccion<-predict(alumnos_m3, alumnos_test, type="response")
if (Modelo == 4) Prediccion<-predict(alumnos_m4, alumnos_test, type="response")
if (Modelo == 5) Prediccion<-predict(alumnos_m5, alumnos_test, type="response")
if (Modelo == 6) Prediccion<-predict(alumnos_m6, alumnos_test, type="response")
if (Modelo == 7) Prediccion<-predict(alumnos_m7, alumnos_test, type="response")
if (Modelo == 8) Prediccion<-predict(alumnos_m8, alumnos_test, type="response")
if (Modelo == 9) Prediccion<-predict(alumnos_m9, alumnos_test, type="response")

## Comparamos los resultados predecidos con los reales, añadimos la columna de diferencia entre ambos v
Resultados<-data.frame(
            real=alumnos_test$Chance.of.Admit,
            predicted= Prediccion,
            dif=alumnos_test$Chance.of.Admit- Prediccion  )
colnames(Resultados)<-c("Real","Predecido", "Diferencia")

summary(Resultados)
```

```
##       Real          Predecido        Diferencia
##  Min.   :0.3600   Min.   :0.3988   Min.   :-0.2034700
##  1st Qu.:0.6400   1st Qu.:0.6369   1st Qu.:-0.0215911
##  Median :0.7200   Median :0.7202   Median : 0.0075381
##  Mean   :0.7109   Mean   :0.7110   Mean   :-0.0000392
##  3rd Qu.:0.8000   3rd Qu.:0.7815   3rd Qu.: 0.0372797
##  Max.   :0.9700   Max.   :0.9799   Max.   : 0.1166458
```

Como hemos visto que según cómo se partan los grupos el mejor modelo obtenido varía notablemente, realizaremos el mismo cálculo pero partiendo múltiples veces los datos en diferentes grupos de prueba y test para evitar sobreestimar/infraestimar los resultados por culpa de esta única partición.

Para ello, utilizaremos la función `trainControl` y el mètodo *leave-one-out* que consiste en realizar la validación cruzada de tipo *k-fold* donde k se ajusta al número de muestras del conjunto de datos:

```r
# Leave-one-out
train_control <- trainControl(method="LOOCV")
modelo1 <- train(Chance.of.Admit ~ CGPA + GRE.Score + TOEFL.Score,
               data=alumnos_estudio, trControl=train_control, method="lm")
modelo2 <- train(Chance.of.Admit ~ CGPA + GRE.Score + University.Rating,
                 data=alumnos_estudio, trControl=train_control, method="lm")
modelo3 <- train(Chance.of.Admit ~ CGPA + GRE.Score + SOP,
                 data=alumnos_estudio, trControl=train_control, method="lm")
modelo4 <- train(Chance.of.Admit ~ CGPA + TOEFL.Score + University.Rating,
                 data=alumnos_estudio, trControl=train_control, method="lm")
modelo5 <- train(Chance.of.Admit ~ CGPA + TOEFL.Score + SOP,
                 data=alumnos_estudio, trControl=train_control, method="lm")
modelo6 <- train(Chance.of.Admit ~ CGPA + University.Rating + SOP,
                 data=alumnos_estudio, trControl=train_control, method="lm")
modelo7 <- train(Chance.of.Admit ~ GRE.Score + TOEFL.Score + University.Rating,
                 data=alumnos_estudio, trControl=train_control, method="lm")
modelo8 <- train(Chance.of.Admit ~ GRE.Score + TOEFL.Score + SOP,
                 data=alumnos_estudio, trControl=train_control, method="lm")
modelo9 <- train(Chance.of.Admit ~ TOEFL.Score + University.Rating + SOP,
                 data=alumnos_estudio, trControl=train_control, method="lm")
#print(modelo1)
#summary(modelo1)
```

```r
regresion <- matrix(c(1 , summary(modelo1)$r.squared,
                      2 , summary(modelo2)$r.squared,
                      3 , summary(modelo3)$r.squared,
                      4 , summary(modelo4)$r.squared,
                      5 , summary(modelo5)$r.squared,
                      6 , summary(modelo6)$r.squared,
                      7 , summary(modelo7)$r.squared,
                      8 , summary(modelo8)$r.squared,
                      9 , summary(modelo9)$r.squared),
                    ncol = 2, byrow = TRUE)
colnames(regresion) <- c("Modelo", "Bondad")

knitr::kable(regresion)  %>%
kable_styling("striped", full_width = F)
```

Modelo	Bondad
1	0.8046167
2	0.8055430
3	0.8045239
4	0.8011373
5	0.7993696
6	0.7894303
7	0.7333357
8	0.7356165
9	0.6960131

```r
regresion[which.max(regresion[,2]),1]
```

```
## Modelo
##      2
```

Vemos que, utilizando múltiples agrupaciones, el mejor modelo es el número 2.

5 Representación de los resultados

En la propia resolución de los ejericios hemos incluido las tablas con los resultados y los gráficos que hemos considerado significativos.

6 Resolución del problema

A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Mediante los tests realizados, hemos podido comprobar que las variables `CGPA`, `GRE.Score` y `TOEFL.Score` están altamente relacionadas con la probabilidad de que el alumno sea admitido en la universidad deseada. Además, como la correlación es positiva, sabemos que cuanto más alto sea el valor de estas variables, más alta será también la probabilidad de admisión. Así pues, la media de notas acumulada (`CGPA`), la nota obtenida en la selectividad (`GRE.Score`) y la nota en el test de inglés (`TOEFL.Score`) son críticas a la hora de aumentar las posiblidades de ser admitido.

También hemos podido determinar que hay una diferencia significativa en la confianza en ser admitidos entre aquellos alumnos que optan por universidades de calificación alta y aquellos que optan por las de calificación baja, siendo más alta la de los primeros.

7 Código

El código necesario para resolver la práctica se ha incluido en este mismo documento.

8 Contribuciones

Contribuciones	Firma
Búsqueda previa	Gervasio Cuenca, Sabela de la Torre
Redacción de las respuestas	Gervasio Cuenca, Sabela de la Torre
Desarrollo código	Gervasio Cuenca, Sabela de la Torre