Tipología y ciclo de vida de los datos

Práctica 2

Universitat Oberta de Catalunya

Javier Martínez Arellano

Enero
2019

Universitat Oberta de Catalunya

Javier Martínez Arellano

Índice

PRÁCTICA 2

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset.

¿Por qué es importante y qué pregunta/problema pretende responder?
El conjunto de datos que se usará este caso práctico, obtenido del siguiente enlace *https://www.kaggle.com/spscientist/students-performance-in-exams*, recoge las notas de distintas pruebas (matemáticas, lectura y escritura) de un total de 1000 personas, así como información complementaria de cada individuo como género, raza/etnia, etc.

El dataset recoge datos de individuos que han realizado varios exámenes (matemáticas, lectura y escritura), así como las notas obtenidas. Se compone de las siguientes características:

Gender: Género (male/female)
race/ethnicity: raza o etnia (grupo a, grupo b, …)
parental level of education: Nivel de estudios de los padres (high school, master's degree,…)
lunch: si tienen costeada la comida (free/reduced, standard)
test preparation course: Si han realizado un test de preparación (completed/none)
math score: nota en la prueba de matemáticas.
reading score: nota en la prueba de lectura.
writing score: nota en la prueba escrita

Mediante el tratamiento de estos datos se pretende responder a si las mujeres obtienen mejores resultados en la prueba de matemáticas así como averiguar si la nota de esta prueba guarda alguna relación con la notas de alguna de las otras dos, intentando

encontrar un modelo para predecir la nota de matemáticas mediante las otras dos variables cuantitativas seleccionadas y las variables cualitativas.
Este tipo de información podría ser requerida por el centro que realiza las pruebas a modo de control de los estudiantes, de las materias examinadas y control de becas, por ejemplo.

2. Integración y selección de los datos de interés a analizar.

Usaremos este dataset como única fuente de datos. Comenzaremos con la lectura de los datos, pre-visualizando algunos valores de cada variable (columnas), comprobando el tipo de cada variable y pasaremos los enteros a numéricos.

```
➢ datos <- read.csv("Students.csv")
➢ summary(datos)
```

```
   gender     race.ethnicity      parental.level.of.education          lunch
female:518   group A: 89    associate's degree:222          free/reduced:355
male  :482   group B:190    bachelor's degree :118          standard    :645
             group C:319    high school       :196
             group D:262    master's degree   : 59
             group E:140    some college      :226
                            some high school  :179
test.preparation.course   math.score      reading.score    writing.score
completed:358           Min.   :  0.00   Min.   : 17.00   Min.   : 10.00
none     :642           1st Qu.: 57.00   1st Qu.: 59.00   1st Qu.: 57.75
                        Median : 66.00   Median : 70.00   Median : 69.00
                        Mean   : 66.09   Mean   : 69.17   Mean   : 68.05
                        3rd Qu.: 77.00   3rd Qu.: 79.00   3rd Qu.: 79.00
                        Max.   :100.00   Max.   :100.00   Max.   :100.00
```

Con este resumen de los datos, vemos los distintos valores de las variables cualitativas y los valores mínimo y máximo de las cuantitativas. Así, sin profundizar, ya vemos un nota = 0 en la prueba de matemáticas que nos indica que tendremos que revisar este valor para ver qué significado tiene, y una posible actuación, si se realiza alguna, ya que no debería obtenerse esta nota si un individuo se presenta a la prueba.

```
➢ library(knitr)
➢ tipos <- sapply(datos, class)
➢ kable(data.frame(variables=names(tipos),clase=as.vector(tipos)))
```

```
|variables                    |clase   |
|:----------------------------|:-------|
|gender                       |factor  |
|race.ethnicity               |factor  |
|parental.level.of.education  |factor  |
|lunch                        |factor  |
|test.preparation.course      |factor  |
|math.score                   |integer |
|reading.score                |integer |
|writing.score                |integer |
```

```
➢ datos[6:8] <- lapply(datos[6:8], as.numeric)
➢ kable(data.frame(variables=names(tipos),clase=as.vector(tipos)))
```

```
|variables                    |clase   |
|:----------------------------|:-------|
|gender                       |factor  |
|race.ethnicity               |factor  |
|parental.level.of.education  |factor  |
|lunch                        |factor  |
|test.preparation.course      |factor  |
|math.score                   |numeric |
|reading.score                |numeric |
|writing.score                |numeric |
```

A continuación, se eliminarán las variables que no se usarán en el análisis de los datos (*):

Educación de los padres, comida costeada y realización del test de preparación

Objetivo Inicial en la eliminación de datos:

#Eliminación de datos irrelevantes

```
➢ datos <- datos[,-(3:5)]
➢ summary(datos)
```

```
   gender    race.ethnicity   math.score     reading.score    writing.score
female:518   group A: 89    Min.   :  0.00   Min.   : 17.00   Min.   : 10.00
male  :482   group B:190    1st Qu.: 57.00   1st Qu.: 59.00   1st Qu.: 57.75
             group C:319    Median : 66.00   Median : 70.00   Median : 69.00
             group D:262    Mean   : 66.09   Mean   : 69.17   Mean   : 68.05
             group E:140    3rd Qu.: 77.00   3rd Qu.: 79.00   3rd Qu.: 79.00
                            Max.   :100.00   Max.   :100.00   Max.   :100.00
```

(*)Correción del Obtetivo Inicial:

A priori, se iban a descartar varias variables cualitativas como comida costeada, educación de los padres y realización del test de preparación pero manteniendo dicha información se han mejorado los resultados de los modelos predictivos. De esta forma, mantenemos todas las variables del dataset.

3. Limpieza de los datos.

3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

```
#CEROS Y ELEMENTOS VACIOS
> sapply(datos, function(x) sum(is.na(x)))
```

```
                gender           race.ethnicity  parental.level.of.education                        lunch
                     0                        0                            0                            0
 test.preparation.course               math.score               reading.score               writing.score
                     0                        0                            0                            0
```

Con este comando y como se ha visto previamente con la ayuda del comando Summary(datos), vemos que la muestra no contiene elementos vacíos, en cambio, sí contiene valores 0 en las pruebas que no se han realizado o no se ha informado la nota.

Como se veía en la vista previa de los datos, existe algún registro en la prueba de matemáticas con valor 0.

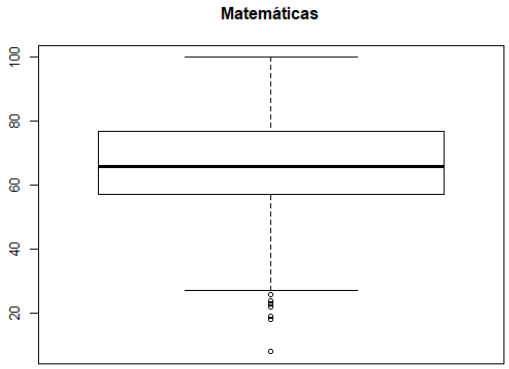
En este caso, se procederá a eliminar estos registros ya que una nota no informada no es útil para este análisis.

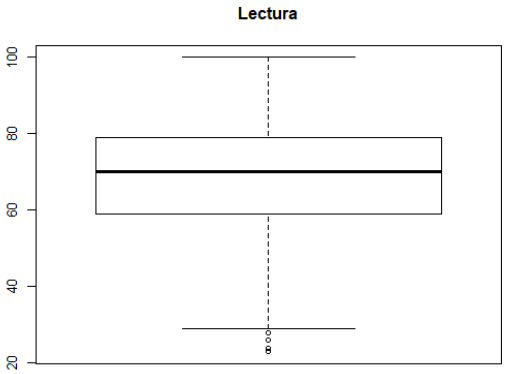
```
   #Borrar registros con resultado = 0 en alguna prueba
➢  nrow(datos)
[1] 1000
➢  datos <- datos[!datos$math.score==0,]
➢  nrow (datos)
[1] 999
```

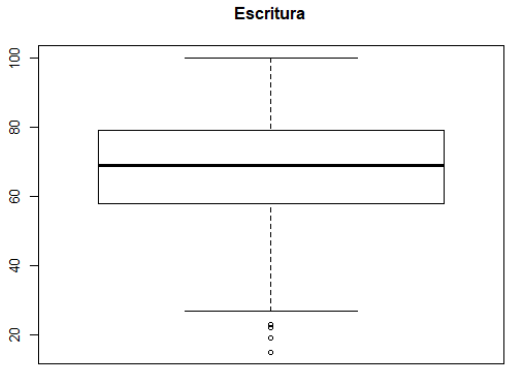
Se ha eliminado 1 registro.

3.2. Identificación y tratamiento de valores extremos.

Los valores extremos son aquellos que se distancian mucho del resto dando a pensar que no se han obtenido de la misma manera que el resto o que pueden ser incorrectos. Vamos a analizar estos valores en las variables cuantitativas.








```
#OUTLIERS
➢ boxplot.stats(datos$math.score)$out
[1] 18 22 24 26 19 23  8

➢ boxplot.stats(datos$reading.score)$out
[1] 17 26 28 23 24 24

➢ boxplot.stats(datos$writing.score)$out
[1] 10 22 19 15 23
```

Una vez revisado los valores 0 en los resultados de las notas, el resto de valores son perfectamente válidos aunque haya valores más distanciados de los más comunes, por lo que no se procederá a realizar ninguna acción sobre los mismos.

Después del tratamiento de los datos, antes del análisis, podemos guardar el conjunto resultante en el mismo formato que el dataset original.

```
write.csv(datos, file = "ResultadoNotas.csv" )
```

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Se procede a seleccionar los datos que pueden ser útiles en el análisis..

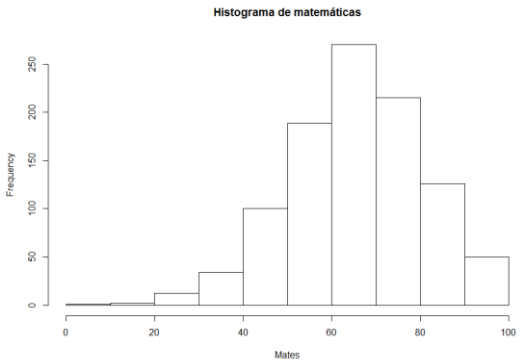
```
#selección de datos para el análisis
Mates    <- datos$math.score
Lectura  <- datos$reading.score
Escritura<- datos$writing.score
Genero   <- datos$gender
Raza     <- datos$race.ethnicity
Padres   <- datos$parental.level.of.education
Comida   <- datos$lunch
TestPrep <- datos$test.preparation.course
```

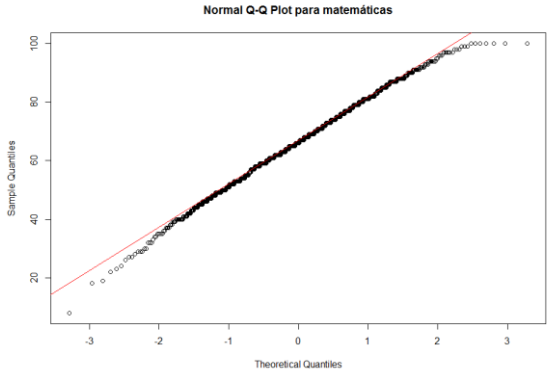
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

A continuación se presentan los gráficos de histograma y quantile-quantile Plot de cada una de las pruebas.

Matemáticas:

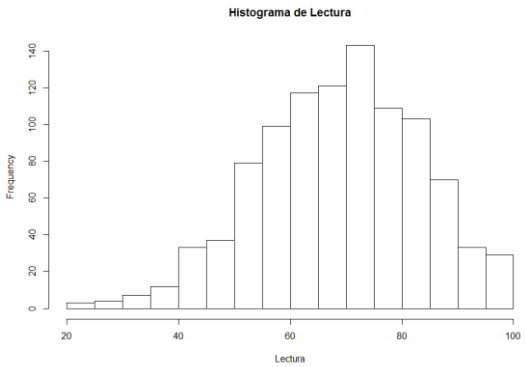
```
hist(Mates,main = paste("Histograma de matemáticas"))
qqnorm(Mates,main = paste("Normal Q-Q Plot para matemáticas"))
qqline(Mates,col="red")
```

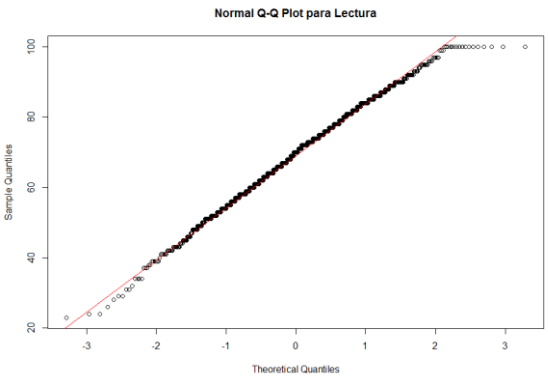




Lectura:

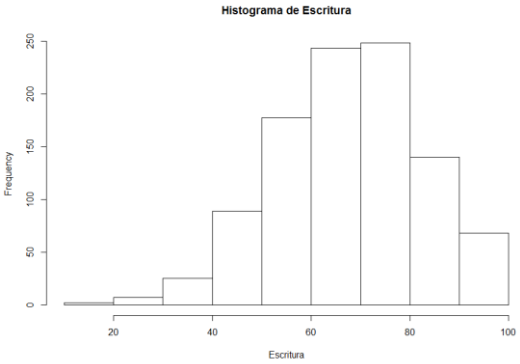
```
hist(Lectura, main = paste("Histograma de Lectura"))
qqnorm(Lectura,main = paste("Normal Q-Q Plot para Lectura"))
qqline(Lectura,col="red")
```

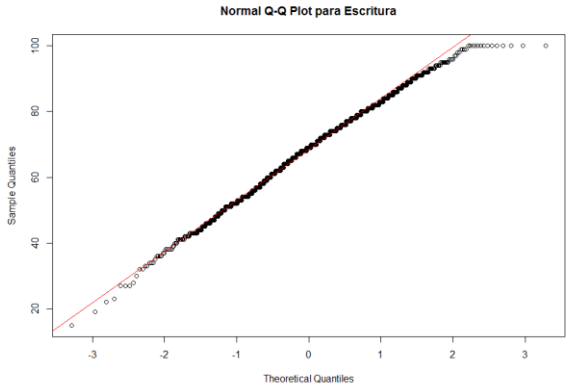




Escritura:

```
hist(Escritura, main = paste("Histograma de Escritura"))
qqnorm(Escritura,main = paste("Normal Q-Q Plot para Escritura"))
qqline(Escritura,col="red")
```





Para comprobar la normalidad de las variables cuantitativas, se usará el test de Shapiro Wilk con un alpha de 0.05, donde la hipótesis nula es que los valores siguen una distribución normal.

```
> NombreColumnas <- c("Matemáticas","Lectura","Escritura")
> NormalidadSp <-matrix(c(shapiro.test(Mates)$p.value,
> +                       shapiro.test(Lectura)$p.value,
> +                       shapiro.test(Escritura)$p.value), ncol = 3,
  byrow = T)
> colnames(NormalidadSp) <- NombreColumnas
> NormalidadSp
```

```
     Matemáticas          Lectura         Escritura
[1,] 0.002366662 0.0001895538       8.816204e-05
```

Según estos resultados, las 3 devuelve un p-value inferior a alpha con lo que no siguen una distribución normal.

Comparando las gráficas y los resultados del test de Shapiro wilk nos lleva a hacer alguna prueba más para confirmar los resultados. Estos resultados pueden deberse a que hay muchos valores que se repiten.

Vamos a analizar los resultados, separándolos por género:

```
> tapply(Mates,Genero,shapiro.test)
```

```
$female
        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.99458, p-value = 0.06441
```

```
$male
        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.99356, p-value = 0.03802
```

Vemos que para las mujeres, sí sigue una distribución normal.

```
➢ tapply(Lectura,Genero,shapiro.test)

$female
        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.98797, p-value = 0.000292


$male
        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.99462, p-value = 0.08965
```

En este caso, es el grupo de los hombres el que sigue una distribución normal

```
➢ tapply(Escritura,Genero,shapiro.test)
$female
        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.9829, p-value = 9.171e-06


$male
        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.99481, p-value = 0.104
```

Y es esta prueba, el grupo de los hombres el que sigue una distribución normal.

Es conocido que al haber muchos valores repetidos, en el test de Shapiro Wilk puede haber algunas imprecisiones, por ello, podemos comprobar estos datos con el test de Lillie:

```
#Test de Lillie
 ➢  NormalidadLi <-matrix(c(lillie.test(Mates)$p.value,
+                           lillie.test(Lectura)$p.value,
+                           lillie.test(Escritura)$p.value), ncol = 3,
byrow = T)
 ➢  colnames(NormalidadLi) <- NombreColumnas
 ➢  NormalidadLi
     Matemáticas      Lectura    Escritura
[1,]  0.05583007 0.0001390798 0.0005397114
```

En este caso nos indica que las notas de la prueba de matemáticas sí siguen una distribución normal.

A continuación se comprobará la homogeneidad de la varianza de las notas de la prueba de matemáticas por género.

```
 ➢  fligner.test(Mates ~ Genero, data = datos)

Fligner-Killeen test of homogeneity of variances

data:  Mates by Genero
Fligner-Killeen:med chi-squared = 0.18773, df = 1, p-value = 0.6648
```

Con este resultado con un p-value mayor a 0.05, podemos aceptar la homogeneidad de ambas varianzas

Como extensión a esto, podemos realizar la misma comprobación para las otras dos pruebas.

```
#Lectura
 ➢  fligner.test(Lectura ~ Genero, data = datos)

        Fligner-Killeen test of homogeneity of variances

data:  Lectura by Genero
Fligner-Killeen:med chi-squared = 0.0001607, df = 1, p-value = 0.9899
```

Con un valor muy elevado de P-value, aceptaríamos la homogeneidad de varianzas

```
#Escritura
➢  fligner.test(Escritura ~ Genero, data = datos)

        Fligner-Killeen test of homogeneity of variances

data:  Escritura by Genero
Fligner-Killeen:med chi-squared = 0.028291, df = 1, p-value = 0.8664
```

Del mismo modo, aceptaríamos a la homogeneidad de varianzas.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

Correlación de variables

Vamos a analizar qué variable influye más en la nota de la prueba de matemáticas, para ellos calculamos la correlación con el resto de las pruebas:

```
➢  corr_matrix <- matrix(nc = 2, nr = 0)
➢  colnames(corr_matrix) <- c("Estimación", "p-value")
➢  for (i in 4:5) {
+     spearman_test = cor.test(datos[,i],
+                              datos[,"math.score"],
+                              method = "spearman")
+     corr_coef = spearman_test$estimate
+     p_val = spearman_test$p.value
+     # incluiomos el valor en la matriz de correlaciones
+     pair = matrix(ncol = 2, nrow = 1)
+     pair[1][1] = corr_coef
+     pair[2][1] = p_val
+     corr_matrix <- rbind(corr_matrix, pair)
+     rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(datos)[i]
+ }
```

```
➢  print(corr_matrix)
              Estimación       p-value
reading.score  0.8034746 8.656027e-227
writing.score  0.7776720 3.459858e-203
```

Ambas variables tienen una correlación significativa con la prueba de matemáticas, siendo la prueba de lectura la que tiene valores más próximos a -1 o 1

¿Quién saca mejor nota media en la prueba de matemáticas: hombres o mujeres?

Para ello realizaremos un contraste de hipótesis, tomando un alpha de 0,05, usando la media de las dos poblaciones: hombres y mujeres.

$$H0 : \mu1 - \mu2 <= 0$$
$$H1 : \mu1 - \mu2 > 0$$

Donde μ1 es la media de los hombres y μ2, la media de las mujeres. Como hipótesis nula se establece que las mujeres sacan mejores notas.

Como se cumplen las condiciones de normalidad y homogeneidad de la varianza para los test de matemáticas, procederemos a usar el t-test.

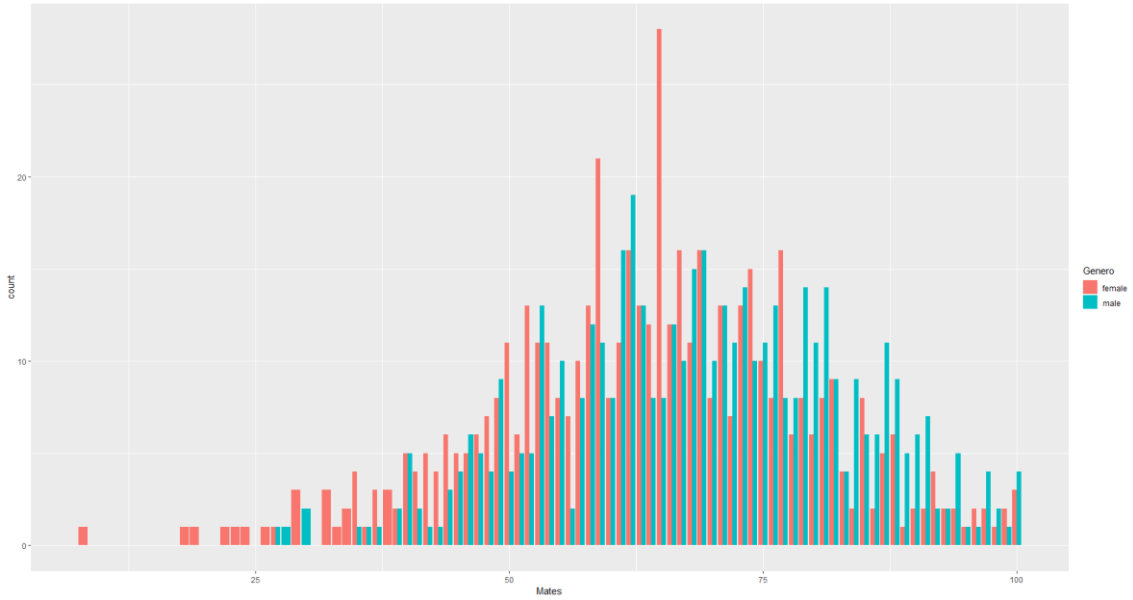
```
> MatesHombre <- datos[datos$gender=="male",]$math.score
> MatesMujer  <- datos[datos$gender=="female",]$math.score
> t.test(MatesHombre, MatesMujer, alternative = "greater")

        Welch Two Sample t-test

data:  MatesHombre and MatesMujer
t = 5.3077, df = 996.91, p-value = 6.842e-08
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 3.429699      Inf
sample estimates:
mean of x mean of y
 68.72822  63.75629
```

Con un p-value inferior al nivel de significación fijado, rechazamos la hipótesis nula. Podemos concluir, por tanto, que las notas de los hombres son mejores.

```
> ggplot() + aes(x=Mates, fill=Genero) + geom_bar(position ='dodge' )
```



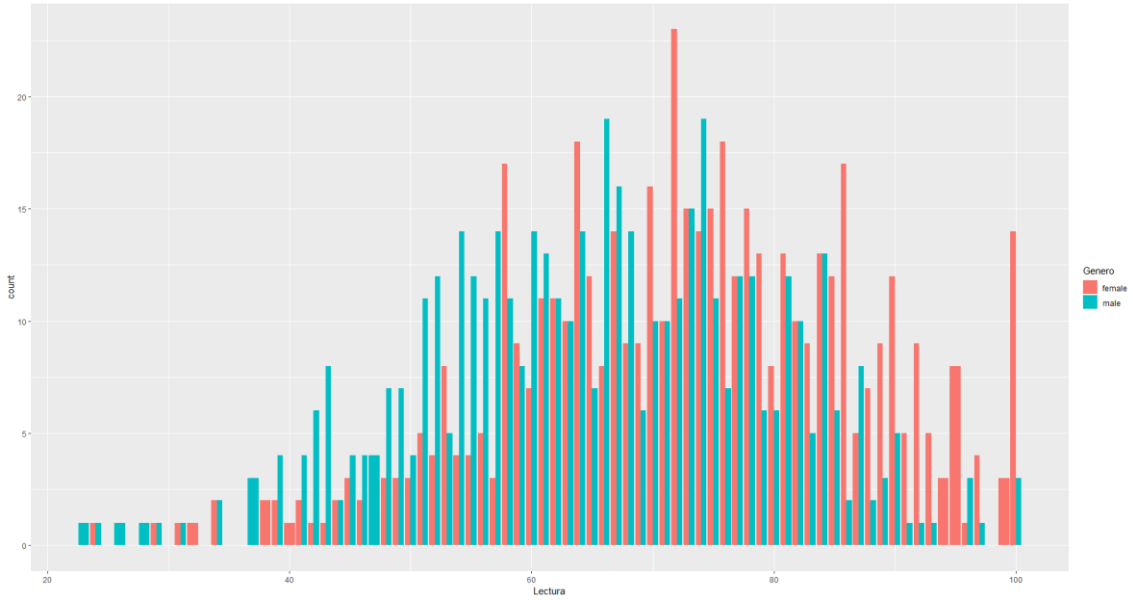
Como extensión a las preguntas planteadas, comprobamos si pasaría lo mismo con las otras dos pruebas.

```
##Para el caso de la prueba de lectura
➢ LectuHombre <- datos[datos$gender=="male",]$reading.score
➢ LectuMujer  <- datos[datos$gender=="female",]$reading.score
➢ t.test(LectuHombre, LectuMujer, alternative = "greater")

        Welch Two Sample t-test

data:  LectuHombre and LectuMujer
t = -8.1397, df = 994.27, p-value = 1
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -8.707574       Inf
sample estimates:
mean of x mean of y
 65.47303  72.71567

➢ ggplot() + aes(x=Lectura, fill=Genero) + geom_bar(position='dodge')
```

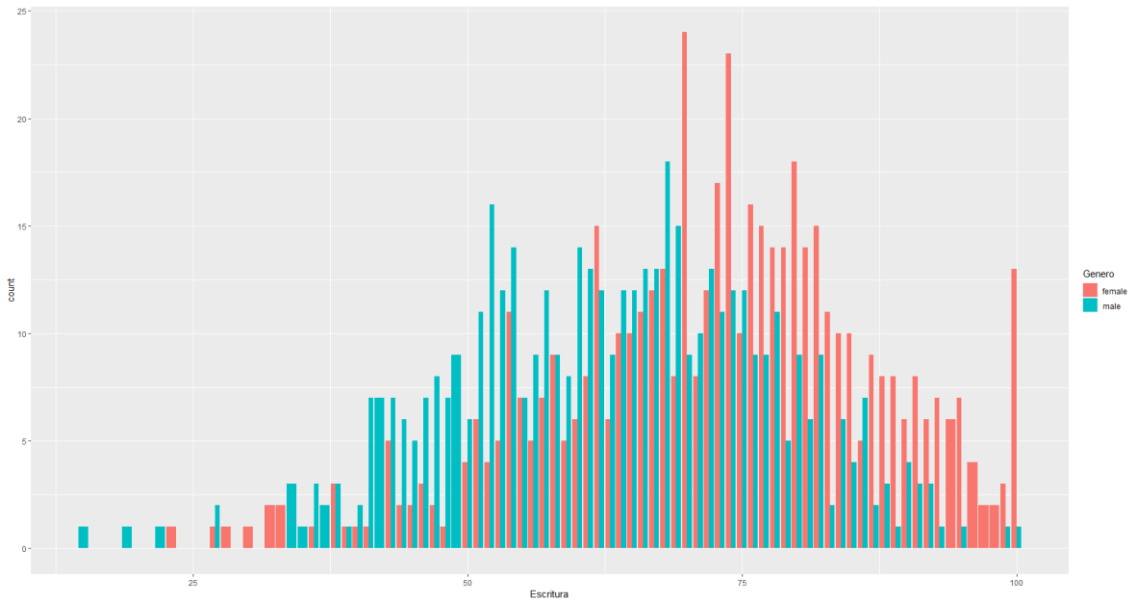



```
##Para el caso de la prueba de escritura
 ➢  EscriHombre <- datos[datos$gender=="male",]$writing.score
 ➢  EscriMujer  <- datos[datos$gender=="female",]$writing.score
 ➢  t.test(EscriHombre, EscriMujer, alternative = "greater")

        Welch Two Sample t-test

data:  EscriHombre and EscriMujer
t = -10.209, df = 995.7, p-value = 1
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -10.77284       Inf
sample estimates:
mean of x mean of y
 63.31120  72.58801

 ➢  ggplot() + aes(x=Escritura,fill=Genero) + geom_bar(position='dodge')
```



Contrariamente a lo que pasa con la prueba de matemáticas y de forma muy significativa, en las pruebas de Lectura y Escritura, con un P-value = 1, no podríamos rechazar la hipótesis nula y aceptaríamos que las mujeres sacan, en ambas, mejores notas que los hombres.

Para cada caso se muestra una gráfica donde, visualmente, poder ver las diferencias entre las notas de hombres y mujeres mostrando en el eje X, las distintas notas y en el eje Y el número de individuos que obtuvieron esa nota.

Estimación de notas de matemáticas por medio de una regresión lineal

1- Elección de modelo basado en la bondad del ajuste

Regresores cuantitativos:

- Lectura
- Escritura

Regresores cualitativos:

- Género (G)
- Raza (R)
- Educación de los padres (P)

- Comida costeada (C)
- Realización del test de preparación (T)

Variable a predecir:

- Mates

```
> modeloLectura   <- lm(Mates ~ Lectura, data = datos)
> modeloEscritura <- lm(Mates ~ Escritura, data = datos)
> modeloAmbas     <- lm(Mates ~ Lectura + Escritura, data = datos)
> modeloLER     <- lm(Mates ~ Lectura + Escritura + Raza, data = dato
  s)
> modeloLEG     <- lm(Mates ~ Lectura + Escritura + Genero, data =dat
  os)
> modeloLEGR     <- lm(Mates ~ Lectura + Escritura + Genero + Raza, d
  ata = datos)
> modeloLEGRP     <- lm(Mates ~ Lectura + Escritura + Genero + Raza +
  Padres, data = datos)
> modeloLEGRC     <- lm(Mates ~ Lectura + Escritura + Genero + Raza +
  Comida, data = datos)
> modeloLEGRT     <- lm(Mates ~ Lectura + Escritura + Genero + Raza +
  TestPrep, data = datos)
> modeloLEGRPCT   <- lm(Mates ~ Lectura + Escritura + Genero + Raza +
  Padres + Comida + TestPrepestep, data = datos)
```

Adjunto captura con los comandos más estructurados (tabulados)

```
modeloLectura   <- lm(Mates ~ Lectura, data = datos)
modeloEscritura <- lm(Mates ~ Escritura, data = datos)
modeloAmbas     <- lm(Mates ~ Lectura + Escritura, data = datos)
modeloLER       <- lm(Mates ~ Lectura + Escritura + Raza, data = datos)
modeloLEG       <- lm(Mates ~ Lectura + Escritura + Genero, data = datos)
modeloLEGR      <- lm(Mates ~ Lectura + Escritura + Genero + Raza, data = datos)
modeloLEGRP     <- lm(Mates ~ Lectura + Escritura + Genero + Raza + Padres, data = datos)
modeloLEGRC     <- lm(Mates ~ Lectura + Escritura + Genero + Raza + Comida, data = datos)
modeloLEGRT     <- lm(Mates ~ Lectura + Escritura + Genero + Raza + TestPrep, data = datos)
modeloLEGRPCT   <- lm(Mates ~ Lectura + Escritura + Genero + Raza + Padres + Comida + TestPrep, data = datos)
```

```
> tablaCoef <- matrix(c(1 , summary(modeloLectura)$r.squared,
+                       2 , summary(modeloEscritura)$r.squared,
+                       3 , summary(modeloAmbas)$r.squared,
+                       4 , summary(modeloLER)$r.squared,
+                       5 , summary(modeloLEG)$r.squared,
+                       6 , summary(modeloLEGR)$r.squared,
+                       7 , summary(modeloLEGRP)$r.squared,
+                       8 , summary(modeloLEGRC)$r.squared,
+                       9 , summary(modeloLEGRT)$r.squared,
+                       10, summary(modeloLEGRPCT)$r.squared
+ ), ncol = 2, byrow = TRUE)
colnames(tablaCoef) <- c("Modelo", "R^2")
> tablaCoef
```

```
  Modelo       R^2
 [1,]      1 0.6641399
 [2,]      2 0.6390615
 [3,]      3 0.6695021
 [4,]      4 0.6867489
 [5,]      5 0.8377030
 [6,]      6 0.8496834
 [7,]      7 0.8517593
 [8,]      8 0.8632171
 [9,]      9 0.8619210
[10,]     10 0.8743918
```

El modelo 10, modeloLEGRPCT, que usa las notas de las otras dos pruebas, el género, la raza/etnia, la educación de los padres, la comida costeada y el test de preparación, es el que tiene mayor coeficiente de determinación y con él intentaremos predecir el valor de la nota de matemáticas del registro que se eliminó al principio del tratamiento de datos porque no tenía informada la nota.

Si recuperamos los valores de este registro, antes de ser eliminado, sus valores son:

```
➢ datos2 <- read.csv("Students.csv")
➢ datos2[datos2$math.score==0,]
   gender race.ethnicity parental.level.of.education        lunch test
.preparation.course
60 female        group C            some high school free/reduced
none
   math.score reading.score writing.score
60          0            17            10
```

Vemos que tiene una nota en Lectura = 17 y en Escritura = 10, es mujer y del grupo c, con q ue lo que podemos intentar predecir el valor de la prueba de Matemáticas:

```
> registro <- data.frame(
+   Lectura   = 17,
+   Escritura = 10,
+   Genero    = "female",
+   Raza      = "group C",
+   Padres    = "some high school",
+   Comida    = "free/reduced",
+   TestPrep  = "none"
+ )

➢ predict(modeloLEGR, registro)
       1
4.241291
```

Al ser notas enteras, redondeamos obteniendo un valor de 4

2- Evaluación del modelo basado en predicciones.

Con el caso anterior, es difícil sacar conclusiones, de modo que vamos a dividir los datos en dos partes para entrenar un modelo con una de las partes y comprobar las estimaciones con la otra partición y poder contrastar las estimaciones y las notas obtenidas. La división la haremos con una probabilidad de reparto de 75%/25%

```
> particion <- sample(2,nrow(datos),replace=TRUE, prob=c(0.75,0.25))
> TrainData <- datos[particion==1,]
> TestData  <- datos[particion==2,]
```

El reparto obtenido es: para la partición de entrenamiento (training) 740 registros y para el de validación(Test) 259. Recordamos que el total de la muestra era 1000 menos uno que hemos eliminado, 999.

```
> nrow(TrainData)
[1] 740
```

```
> modeloTrain <- lm(formula = math.score ~ reading.score + writing.score + gender + race.ethnicity + parental.level.of.education + lunch + test.preparation.course, data=TrainData)
```

Revisemos el modelo:

```
> summary(modeloTrain)
```

```
Call:
lm(formula = math.score ~ reading.score + writing.score + gender +
    race.ethnicity + parental.level.of.education + lunch + test.preparation.course,
    data = TrainData)

Residuals:
     Min       1Q   Median       3Q      Max
-17.5850  -3.5231   0.1234   3.5549  14.0417

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                             -12.22831    1.45873  -8.383 2.68e-16 ***
reading.score                             0.26693    0.04913   5.434 7.55e-08 ***
writing.score                             0.70470    0.05065  13.914  < 2e-16 ***
gendermale                               13.27998    0.43115  30.802  < 2e-16 ***
race.ethnicitygroup B                     0.73143    0.78294   0.934   0.3505
race.ethnicitygroup C                     0.43401    0.73466   0.591   0.5549
race.ethnicitygroup D                     0.23645    0.76765   0.308   0.7582
race.ethnicitygroup E                     4.68540    0.83594   5.605 2.96e-08 ***
parental.level.of.educationbachelor's degree -1.28571    0.71411  -1.800   0.0722 .
parental.level.of.educationhigh school    0.37081    0.62464   0.594   0.5529
parental.level.of.educationmaster's degree -1.39080    0.93272  -1.491   0.1364
parental.level.of.educationsome college   0.09586    0.58777   0.163   0.8705
parental.level.of.educationsome high school  0.76343    0.64230   1.189   0.2350
lunchstandard                             3.76111    0.42980   8.751  < 2e-16 ***
test.preparation.coursenone               3.34464    0.45558   7.342 5.68e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.316 on 725 degrees of freedom
Multiple R-squared:  0.8787,    Adjusted R-squared:  0.8764
F-statistic: 375.2 on 14 and 725 DF,  p-value: < 2.2e-16
```

El coeficiente R^2 es muy similar al obtenido en el modelo anterior.
A continuación vamos a comprobar la validez del modelo creado comparando los valores estimados con los reales en la partición de datos de Test.

```
➢ predicTestData <- predict(modeloTrain, TestData, type = "response")

➢ TestResult<-data.frame(
+   real      = TMates,
+   predicted = predicTestData,
+   porc      = predicTestData*100/TMates,
+   dif       = 100 - predicTestData*100/TMates,
+   difPos    = abs(100 - predicTestData*100/TMates)
+ )
➢ colnames(TestResult)<-c("Real","Predecido","Porcentaje","Dif%","Dif
  %Pos")
```

A continuación guardamos los valores que acabamos de predecir porque se adjuntarán aquí sólo unos pocos, de ejemplo:

```
#Guardar datos estimados
➢ write.csv(TestResult, file = "NotasEstimadas.csv" )
```

Mostramos parte de estos valores. Se pueden consultar todos en el fichero que acabamos de crear, incluido en la carpeta data del repositorio.

```
kable(TestResult)
```

	Real	Predecido	Porcentaje	Dif%	Dif%Pos	DifPos
6	71	72.73107	102.43812	-2.4381202	2.4381202	1.7310653
12	40	52.57657	131.44142	-31.4414189	31.4414189	12.5765676
15	50	48.50683	97.01365	2.9863487	2.9863487	1.4931743
26	73	73.49711	100.68097	-0.6809670	0.6809670	0.4971059
35	97	93.94742	96.85301	3.1469879	3.1469879	3.0525783
38	50	50.77733	101.55467	-1.5546697	1.5546697	0.7773349
42	58	62.71730	108.13327	-8.1332698	8.1332698	4.7172965
43	53	56.89663	107.35214	-7.3521397	7.3521397	3.8966340
46	65	63.47126	97.64809	2.3519061	2.3519061	1.5287390
49	57	61.31854	107.57639	-7.5763911	7.5763911	4.3185429
56	33	33.16760	100.50788	-0.5078760	0.5078760	0.1675991
70	39	52.56317	134.77737	-34.7773672	34.7773672	13.5631732
72	63	64.22266	101.94072	-1.9407246	1.9407246	1.2226565
74	61	60.27225	98.80696	1.1930383	1.1930383	0.7277534
76	44	42.85067	97.38789	2.6121134	2.6121134	1.1493299
79	61	63.02427	103.31847	-3.3184685	3.3184685	2.0242658
90	73	76.61912	104.95770	-4.9577020	4.9577020	3.6191225
91	65	65.39288	100.60443	-0.6044252	0.6044252	0.3928764
93	71	80.08379	112.79407	-12.7940684	12.7940684	9.0837886
95	79	83.49357	105.68807	-5.6880694	5.6880694	4.4935748
100	65	55.40421	85.23725	14.7627465	14.7627465	9.5957852
104	60	50.78108	84.63513	15.3648708	15.3648708	9.2189225
107	87	90.88660	104.46736	-4.4673596	4.4673596	3.8866029
112	62	58.17395	93.82896	6.1710413	6.1710413	3.8260456
122	91	94.13385	103.44379	-3.4437860	3.4437860	3.1338452
126	87	81.94268	94.18699	5.8130097	5.8130097	5.0573184
130	51	42.61126	83.55149	16.4485052	16.4485052	8.3887376
133	87	74.91511	86.10933	13.8906736	13.8906736	12.0848860
146	22	25.31175	115.05341	-15.0534112	15.0534112	3.3117505
149	68	67.58439	99.38881	0.6111856	0.6111856	0.4156062
151	62	72.08517	116.26640	-16.2664036	16.2664036	10.0851703
153	59	57.45987	97.38961	2.6103902	2.6103902	1.5401302
156	70	77.83349	111.19070	-11.1907028	11.1907028	7.8334919
157	66	67.54771	102.34502	-2.3450151	2.3450151	1.5477100
161	82	74.75182	91.16076	8.8392431	8.8392431	7.2481793
170	67	57.65067	86.04578	13.9542186	13.9542186	9.3493265
181	62	71.09899	114.67580	-14.6757962	14.6757962	9.0989936
184	65	62.86142	96.70988	3.2901241	3.2901241	2.1385807
188	62	71.34998	115.08062	-15.0806176	15.0806176	9.3499829
190	77	78.82493	102.37003	-2.3700340	2.3700340	1.8249262
191	66	65.53794	99.29991	0.7000920	0.7000920	0.4620607
196	61	59.71970	97.90114	2.0988597	2.0988597	1.2803044
199	45	44.84969	99.66598	0.3340222	0.3340222	0.1503100
202	65	67.33227	103.58811	-3.5881100	3.5881100	2.3322715
203	69	77.76971	112.70972	-12.7097215	12.7097215	8.7697078
205	59	49.22904	83.43905	16.5609460	16.5609460	9.7709582
210	58	54.73688	94.37393	5.6260656	5.6260656	3.2631180
218	34	30.61572	90.04622	9.9537789	9.9537789	3.3842848
225	52	61.32507	117.93283	-17.9328293	17.9328293	9.3250713

232	46	49.66704	107.97183	-7.9718303	7.9718303	3.6670419
239	54	58.80504	108.89822	-8.8982189	8.8982189	4.8050382
241	73	66.32071	90.85029	9.1497118	9.1497118	6.6792896
242	80	78.92284	98.65355	1.3464452	1.3464452	1.0771562
243	56	48.12384	85.93543	14.0645692	14.0645692	7.8761587
245	75	77.53478	103.37971	-3.3797070	3.3797070	2.5347803
249	65	56.75492	87.31526	12.6847371	12.6847371	8.2450791
251	47	53.18633	113.16241	-13.1624071	13.1624071	6.1863313
253	60	61.04211	101.73685	-1.7368514	1.7368514	1.0421108
256	62	68.47396	110.44186	-10.4418642	10.4418642	6.4739558
267	63	71.12660	112.89936	-12.8993600	12.8993600	8.1265968
269	88	83.27379	94.62931	5.3706929	5.3706929	4.7262098
271	69	67.10933	97.25990	2.7401043	2.7401043	1.8906720
273	47	46.37296	98.66587	1.3341252	1.3341252	0.6270388
276	83	79.56974	95.86716	4.1328401	4.1328401	3.4302573
277	85	82.82537	97.44161	2.5583904	2.5583904	2.1746319
281	53	52.24267	98.57108	1.4289163	1.4289163	0.7573256
283	73	70.34976	96.36954	3.6304637	3.6304637	2.6502385
296	67	63.95971	95.46225	4.5377456	4.5377456	3.0402895
297	46	46.82266	101.78839	-1.7883877	1.7883877	0.8226583
302	56	56.45508	100.81264	-0.8126393	0.8126393	0.4550780
304	80	74.27668	92.84584	7.1541554	7.1541554	5.7233243
319	63	75.17983	119.33306	-19.3330583	19.3330583	12.1798267
320	56	53.09966	94.82081	5.1791867	5.1791867	2.9003445
323	71	73.40505	103.38740	-3.3874010	3.3874010	2.4050547
328	28	24.02097	85.78919	14.2108142	14.2108142	3.9790280
330	41	46.99341	114.61808	-14.6180803	14.6180803	5.9934129
335	83	85.79721	103.37014	-3.3701361	3.3701361	2.7972129
336	61	56.60540	92.79574	7.2042608	7.2042608	4.3945991
339	24	21.78158	90.75657	9.2434254	9.2434254	2.2184221
346	72	69.88955	97.06882	2.9311770	2.9311770	2.1104474
352	66	64.79837	98.17935	1.8206523	1.8206523	1.2016305
353	63	69.25962	109.93590	-9.9359008	9.9359008	6.2596175
357	63	67.42717	107.02725	-7.0272535	7.0272535	4.4271697
370	73	78.19937	107.12243	-7.1224264	7.1224264	5.1993713
373	74	79.64889	107.63364	-7.6336374	7.6336374	5.6488917
377	80	77.68685	97.10856	2.8914405	2.8914405	2.3131524
378	85	82.44619	96.99552	3.0044813	3.0044813	2.5538091
380	66	67.64696	102.49539	-2.4953875	2.4953875	1.6469557
388	57	59.58614	104.53708	-4.5370822	4.5370822	2.5861369
391	73	65.96242	90.35947	9.6405251	9.6405251	7.0375833
393	76	78.03824	102.68189	-2.6818937	2.6818937	2.0382392
394	57	59.12450	103.72719	-3.7271855	3.7271855	2.1244957
399	74	66.63707	90.05009	9.9499076	9.9499076	7.3629316
405	54	55.12285	102.07935	-2.0793452	2.0793452	1.1228464
409	52	43.05745	82.80278	17.1972195	17.1972195	8.9425541
410	87	87.37128	100.42676	-0.4267638	0.4267638	0.3712845
414	63	71.40719	113.34474	-13.3447430	13.3447430	8.4071881
417	71	71.63386	100.89275	-0.8927539	0.8927539	0.6338552
418	74	75.29258	101.74672	-1.7467232	1.7467232	1.2925751
419	68	67.93032	99.89753	0.1024689	0.1024689	0.0696788
424	59	71.36169	120.95202	-20.9520212	20.9520212	12.3616925
439	70	59.85029	85.50042	14.4995810	14.4995810	10.1497067
443	59	59.99031	101.67849	-1.6784858	1.6784858	0.9903066
449	47	51.13606	108.80014	-8.8001371	8.8001371	4.1360645
…	...	…	…	…	…	…
999	68	66.94795	98.45286	1.5471375	1.5471375	1.0520535

```
> mean(TestResult$`Dif%Pos`)
[1] 7.072005

> mean(TestResult$DifPos)
[1] 4.418292
```

Si sacamos la media de las diferencias porcentuales positivas, obtenemos 7 puntos porcentuales y en la diferencia positiva entre notas, obtenemos un 4. Cabría valorar si la predicción es bastante precisa teniendo en cuenta que las notas reales están entre 1 y 100.

Si intentamos predecir la nota de matemáticas del registro borrado, como en el caso anterior, obtenemos lo siguiente:

```
> registroBorrado <- data.frame(
+   reading.score   = 17,
+   writing.score   = 10,
+   gender          = "female",
+   race.ethnicity  = "group C",
+   parental.level.of.education = "some high school",
+   lunch                       = "free/reduced",
+   test.preparation.course     = "none"
+ )

> predict(modeloTrain, registroBorrado)
       1
3.898638
```

Redondenado, 4. Como en la predicción previa.

5. Representación de los resultados a partir de tablas y gráficas.

Durante el tratamiento y el análisis de los datos se ha incluido apoyo gráfico y tablas que muestran los resultados obtenidos de los distintos comandos que se han ido ejecutando.

6. Resolución del problema.

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

En este caso práctico se ha realizado un tratamiento de los datos, pre-procesamiento, en el que se han comprobado los valores nulos, ceros y extremos. Una vez preparados los datos, se han exportado a un nuevo fichero csv, dando paso al análisis. En el cuál se ha orientado a responder a las preguntar fijadas al comienzo. Para ello se han realizado varias pruebas estadísticas.

Se ha mostrado la relación de las distintas variables cuantitativas con la nota de la prueba de matemáticas y, mediante contrastes de hipótesis, hemos podido comprobar si las mujeres obtenían mejores notas en esta prueba. Además, se añade esta prueba con las notas de lectura y escritura.

Al final, por medio de regresión lineal, hemos conseguido un modelo mediante el cual estimar el valor de la nota de la prueba de matemáticas usando, como ejemplo, un registro que fue descartado para el análisis por no disponer de esta información. Además se ha evaluado el modelo por medio de predicciones y se han comprobado las estimaciones con un subconjunto, Test, de valores reales.

7. Código

Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

El código usado está en R y se ajunta el repositorio de Github, en la siguiente dirección:
https://github.com/jmartinezare/TyCVD_Practica2/tree/master/code

Repositorio del caso práctico:
https://github.com/jmartinezare/TyCVD_Practica2/

6. Referencias

Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.

Squire, Megan (2015). Clean Data. Packt Publishing Ltd.

Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.