Introducción al Aprendizaje Estadístico - Practica 1

Benjamin Tourn

8/4/2021

1. Análisis Exploratorio de Datos

Ejercicio 3

Carga de datos 'Hitters.csv' y llamado a las librerias correspondientes

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.0.5
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

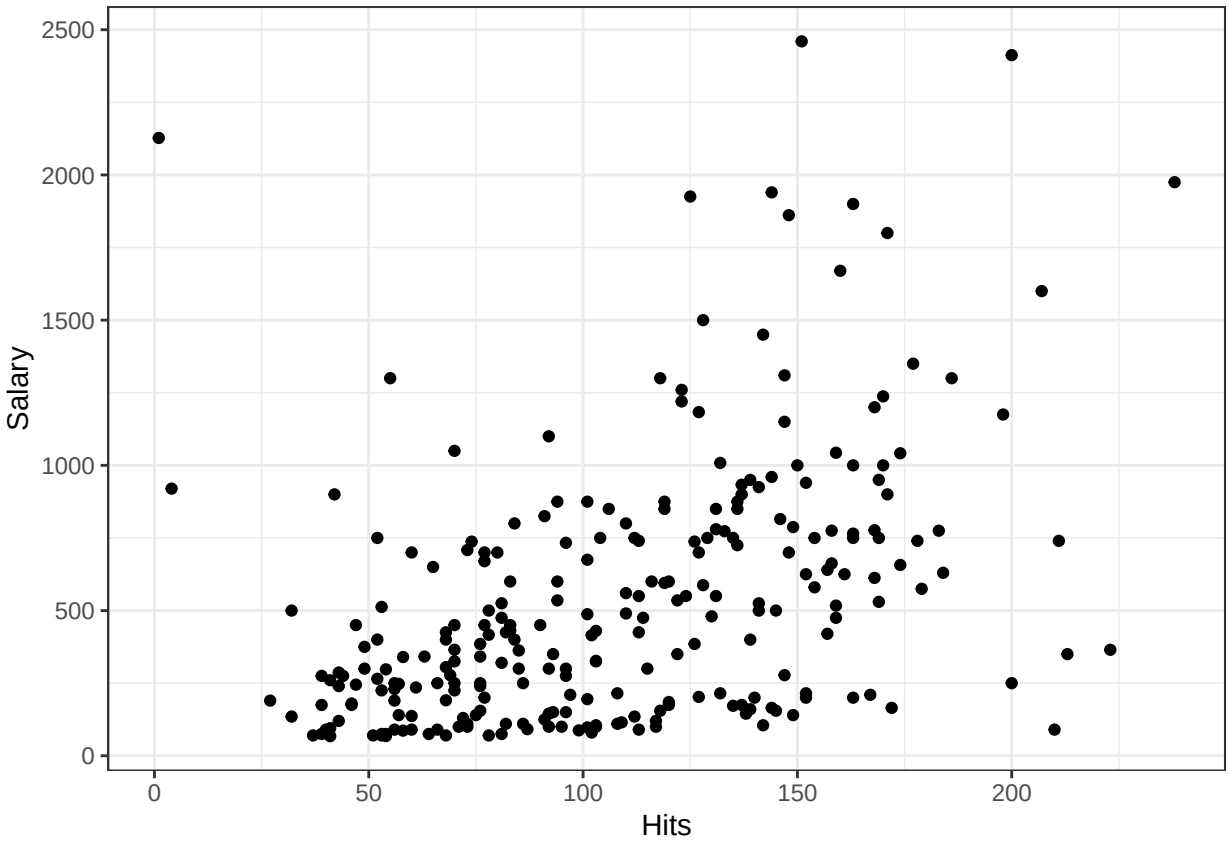
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
datos <- ISLR::Hitters
```

a) A continuación se grafica la variable respuesta *Salary* en función del predictor *Hits*:

```
ggplot(data = datos) +
    geom_point(mapping = aes(x = Hits, y = Salary)) + theme_bw()
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```



La distribución de puntos sugiere una tendencia aproximadamene lineal ascendente, aunque se evidencia una gran dispersión de la nube de puntos. Siendo que la tendencia mencionada tiene pendiente positiva, esto sugiere la existencia de una relación entre el predictor *Hits* y la respuesta *Salary* donde a medida que aumenta el número de hits del jugador aumenta su salario, lo cual es razonable.

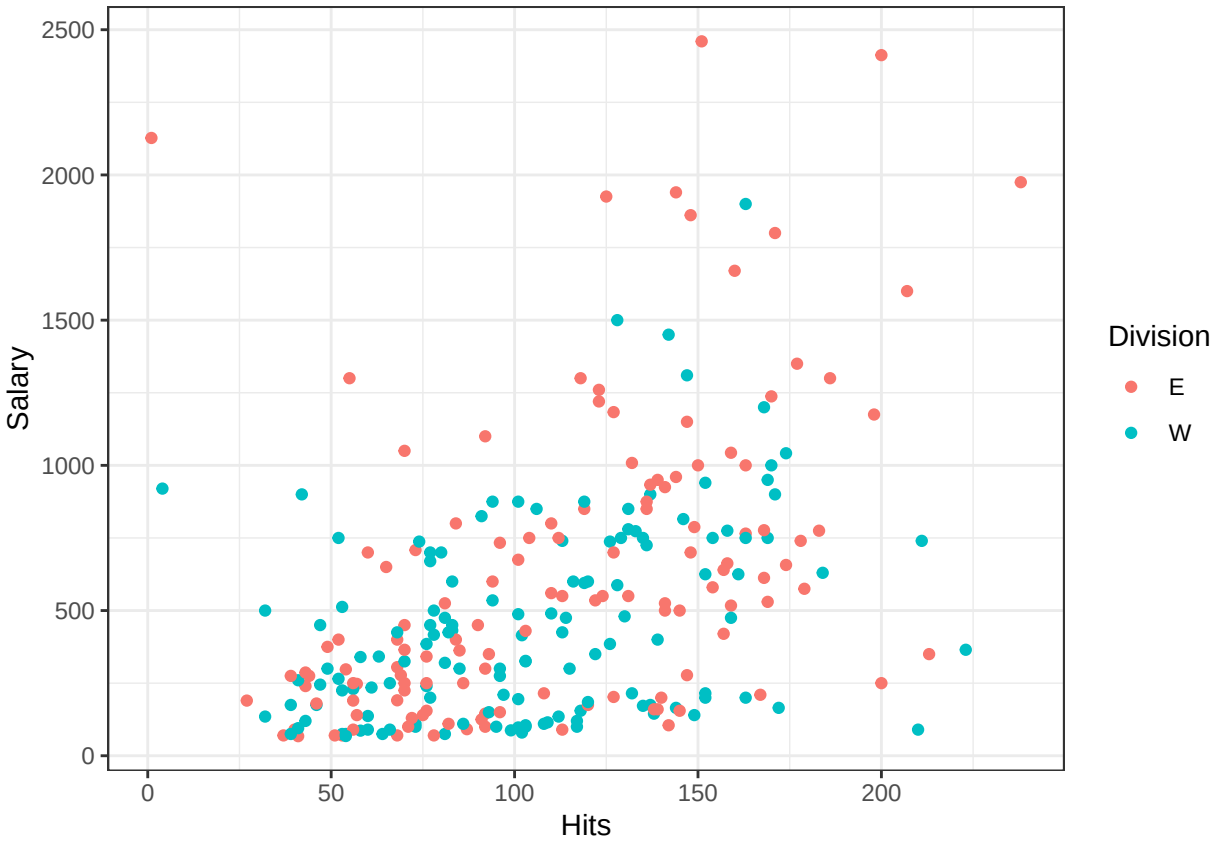
A su vez, existen algunos *outliers*, por ejemplo en la zona próxima al eje de ordenadas de la gráfica, donde para valores muy bajos de *Hits* existen valores elevados de *Salary*, lo cual es contrario a la hipótesis original. Estos datos pueden ser errores, o también algún caso particular donde un jugador percibe un salario muy alto a pesar de no tener hits en su haber (por ejemplo, algún jugador que es considerado una leyenda para un equipo, pero que debido a su edad no es productivo en el número de hits, sin embargo genera un impacto motivacional positivo en sus compañeros).

Por último, existe en la zona inferior del gráfico una concentración de observaciones que se manifiestan también en contra de la hipótesis original de tendencia lineal ascendente, aunque no es lo suficientemente densa como para invalidarla.

b) La siguiente es la gráfica de la variable respuesta *Salary* en función del predictor *Hits*, utilizando la variable *Division* para discriminar los datos:

```
ggplot(data = datos) +
    geom_point(mapping = aes(x = Hits, y = Salary, color = Division)) + theme_bw()
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```



Al discriminar los datos utilizando la variable *Division* se puede afirmar que cada distribución tiene una tendencia aproximadamente lineal ascendente, aunque con pendiente diferente, siendo la pendiente para el caso de los datos correspondientes a la división "E" (puntos rojos) mayor a la pendiente de los datos de la división "W" (puntos azules). Esta observación radica en el hecho que los puntos azules se encuentran mayormente concentrados en la mitad inferior del diagrama, mientras que existen puntos rojos en la zona superior derecha del diagrama.

La tendencia para cada subconjunto de datos es más evidente que en el ítem a), dado que para cada uno de los subconjuntos la dispersión de los datos es menor que para el conjunto total.

Estas tendencias sugieren que los jugadores pertenecientes a la división "E" perciben salarios mas altos que los jugadores de la división "W" para un mismo valor de hits.

c) En base a los gráficos anteriores, se percibe que el gráfico del ítem b) es más descriptivo que el del ítem a), basado en el hecho que el segundo permite visualizar tendencias por separado que son más evidentes (es decir, menos dispersas) que en el primer caso.

d) En este ítem se crea la variable *Hits2* que agrupa la variable *Hits* en cuatro categorías con aproximadamente la misma cantidad de observaciones, mediante el uso de la función `mutate()`:

```
Hits <- datos$Hits
datos <- datos %>% mutate(datos, Hits2 = cut(Hits, fivenum(Hits), include.lowest = TRUE))
```

Mediante la función `head()` visualizamos un vista parcial del DataFrame obtenido:

```
head(datos)
```

```
##                  AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun
## -Andy Allanson     293   66     1   30  29    14     1    293    66      1
## -Alan Ashby        315   81     7   24  38    39    14   3449   835     69
```

```
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     63
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575    225
## -Andres Galarraga   321   87    10   39  42    30     2    396   101     12
## -Alfredo Griffin    594  169     4   74  51    35    11   4408  1133     19
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Andy Allanson       30   29     14      A        E     446      33     20
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
## -Andre Dawson       828  838    354      N        E     200      11      3
## -Andres Galarraga    48   46     33      N        E     805      40      4
## -Alfredo Griffin    501  336    194      A        W     282     421     25
##                   Salary NewLeague     Hits2
## -Andy Allanson        NA         A   (64,96]
## -Alan Ashby        475.0         N   (64,96]
## -Alvin Davis       480.0         A  (96,137]
## -Andre Dawson      500.0         N (137,238]
## -Andres Galarraga   91.5         N   (64,96]
## -Alfredo Griffin   750.0         A (137,238]
```

Podemos apreciar que se incorporó una nueva columna `Hits2`, la cual establece a cuál de los 4 niveles de la variable original *Hits* pertenece cada observación.

e) Para realizar un análisis exploratorio se deben llevar a cabo dos estudios: uno numérico y uno gráfico. Los mismos serán denominados como "Resumen numérico" y "Resumen gráfico", respectivamente.

Resumen numérico

Dado que la variable respuesta es cuantitativa, se pueden estudiar medidas de centro y de variabilidad, tales como la media, mediana, cuartiles, mínimo, máximo, desvío estándar.

Para obtener la media de los salarios a partir de los datos que incluyen la variable *Hit2*, y discriminando los datos mediante la variable *Division*, se obtiene:

```
medias <- group_by(datos, Hits2, Division) %>% summarise(Media = mean(Salary, na.rm=T))
```

```
## `summarise()` has grouped output by 'Hits2'. You can override using the `.groups` argument.
```

```
print(medias, n=Inf)
```

```
## # A tibble: 8 x 3
## # Groups:   Hits2 [4]
##   Hits2     Division Media
##   <fct>     <fct>    <dbl>
## 1 [1,64]    E         361.
## 2 [1,64]    W         281.
## 3 (64,96]   E         353.
## 4 (64,96]   W         392.
## 5 (96,137]  E         758.
## 6 (96,137]  W         451.
## 7 (137,238] E         890.
## 8 (137,238] W         680.
```

De las misma manera se puede proceder par obtener los valores de mínimo, primer cuartil, mediana, tercer cuartil, y máximo, dados por la función `fivenum()`:

```
cincoNum <- group_by(datos, Hits2, Division) %>% summarise(cincoNumeros = fivenum(Salary, na.rm=T))
```

```
## `summarise()` has grouped output by 'Hits2', 'Division'. You can override using the `.groups` argumen
```

```
print(cincoNum, n=Inf)
```

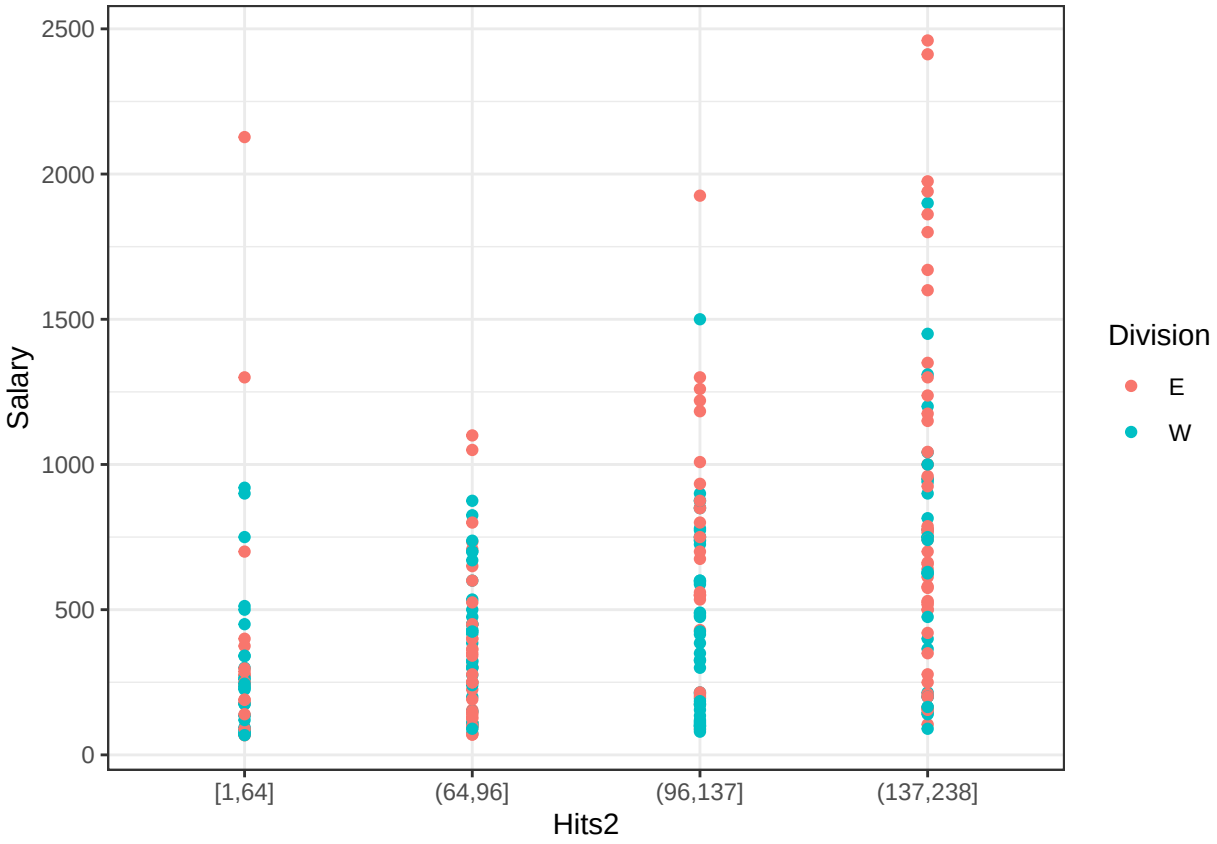
```
## # A tibble: 40 x 3
## # Groups:   Hits2, Division [8]
##    Hits2     Division cincoNumeros
##    <fct>     <fct>           <dbl>
##  1 [1,64]    E                67.5
##  2 [1,64]    E                90
##  3 [1,64]    E               244.
##  4 [1,64]    E               298.
##  5 [1,64]    E              2127.
##  6 [1,64]    W                68
##  7 [1,64]    W                92.5
##  8 [1,64]    W               228.
##  9 [1,64]    W               341.
## 10 [1,64]    W               920
## 11 (64,96]   E                70
## 12 (64,96]   E               142.
## 13 (64,96]   E               289.
## 14 (64,96]   E               450
## 15 (64,96]   E              1100
## 16 (64,96]   W                75
## 17 (64,96]   W               220
## 18 (64,96]   W               401.
## 19 (64,96]   W               518.
## 20 (64,96]   W               875
## 21 (96,137]  E                90
## 22 (96,137]  E               550
## 23 (96,137]  E               750
## 24 (96,137]  E               933.
## 25 (96,137]  E              1926.
## 26 (96,137]  W                80
## 27 (96,137]  W               172
## 28 (96,137]  W               415
## 29 (96,137]  W               738.
## 30 (96,137]  W              1500
## 31 (137,238] E               105
## 32 (137,238] E               500
## 33 (137,238] E               752.
## 34 (137,238] E              1175
## 35 (137,238] E              2460
## 36 (137,238] W                90
## 37 (137,238] W               215
## 38 (137,238] W               740
## 39 (137,238] W               940
## 40 (137,238] W              1900
```

Resumen gráfico

A continuación se muestra la gráfica de la respuesta *Salary* en función de la nueva variable *Hits2*, también discriminada con la variable *Division*:

```
ggplot(data = datos) +
    geom_point(mapping = aes(x = Hits2, y = Salary, color = Division)) + theme_bw()
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```

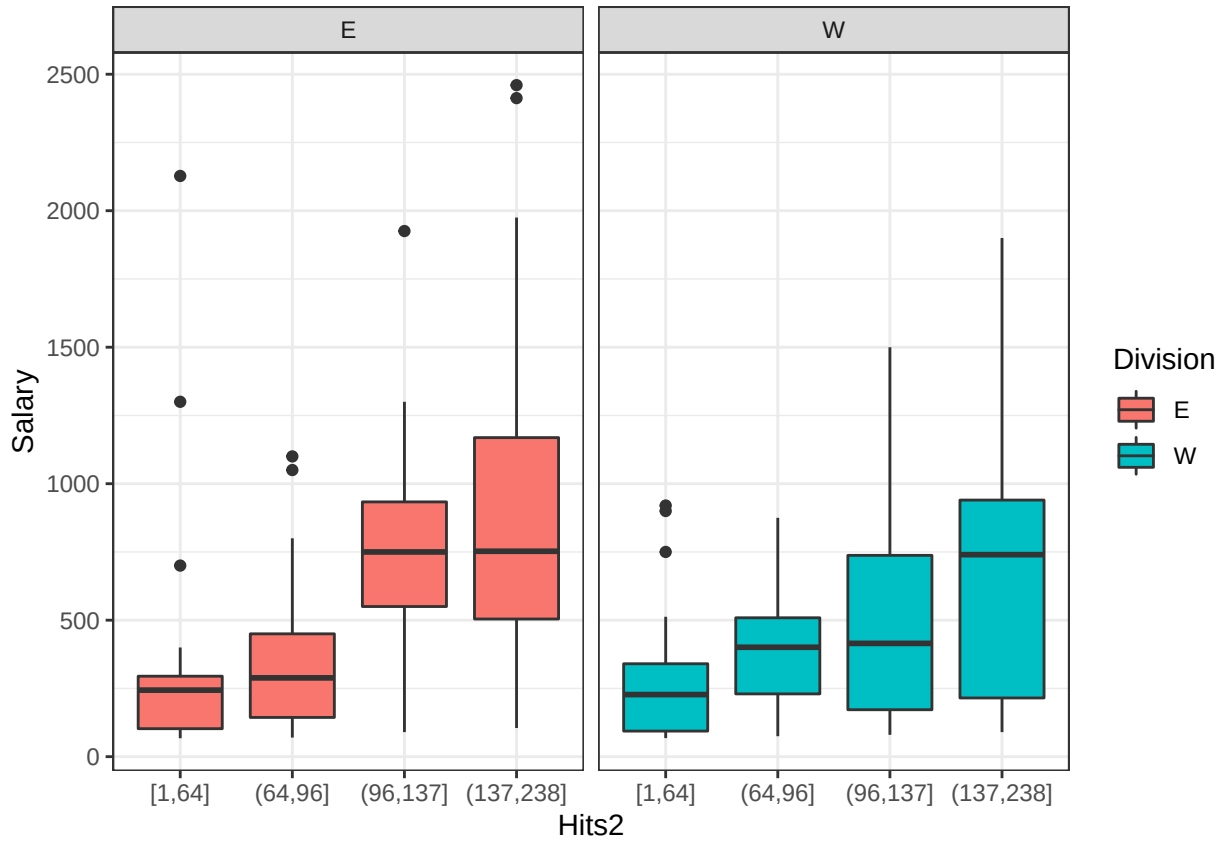


La gráfica resulta aún más descriptiva que en los casos anteriores ya que se puede apreciar claramente cómo la distribución de los valores de *Salary* va ampliando su rango en la medida que la variable categórica *Hits2* incrementa. Para los dos primeros niveles de *Hits2*, i.e. (1,64] y (64,96], prácticamente no se distinguen diferencias entre los dos subconjuntos de datos, excluyendo del análisis los *outliers*. La diferencia entre ellos es notoria en el tercer y cuarto nivel, donde se puede apreciar que los salarios correspondientes a jugadores de la división "E" son más altos que los de la división "W" para los rangos mostrados de hits.

Otra gráfica ilustrativa es la gráfica de tipo `boxplot` que se muestra a continuación:

```
ggplot(datos, aes(x=Hits2, y=Salary, fill = Division)) +
    geom_boxplot() +
    facet_wrap(~Division) + theme_bw()
```

```
## Warning: Removed 59 rows containing non-finite values (stat_boxplot).
```



En ella se aprecia claramente la tendencia ascendente de la variable *Salary* para cada una de los dos valores de *Division*, y para los cuatro niveles de la nueva variable *Hits2*. A su vez, quedan resumidas los valores dados en el Resumen Numérico referidos a los cinco números dados por la función `fivenum()`.

Las dos gráficas mostradas en este Resumen, sumadas a los valores numéricos mostrados en el resumen correspondiente, confirman las observaciones realizadas en los ítems previos.

2. Vecinos más cercanos, errores en entrenamiento y en prueba

Ejercicio 3

En primer lugar, se debe crear el DataFrame correspondiente a la tabla dada en el enunciado, de la siguiente manera:

```
X1 <- c(0, 0, 2, -1, 0, 0)
X2 <- c(1, 1, 0, 0, 0, 3)
X3 <- c(3, 2, 0, 1, 0, 0)
Y  <- c('Poco', 'Mucho', 'Poco', 'Mucho', 'Poco', 'Poco')
df <- data.frame(X1, X2, X3, Y)
print(df)
```

```
##   X1 X2 X3     Y
## 1  0  1  3  Poco
## 2  0  1  2 Mucho
## 3  2  0  0  Poco
## 4 -1  0  1 Mucho
## 5  0  0  0  Poco
## 6  0  3  0  Poco
```

Se desea predecir el valor de `Y` para el caso donde `X1=X2=X3=1`.

En este ejercicio se utilizará el paquete **kknn** que contiene herramientas para realizar análisis de vecinos más cercanos

```
library(kknn)
```

```
## Warning: package 'kknn' was built under R version 4.0.5
```

a) La distancia euclídea entre cada uno de los puntos dados en la tabla y el punto `X1=X2=X3=1` viene dada por la siguiente expresión:

$$d_i = \sqrt{(X_{i1} - X_{p1})^2 + (X_{i2} - X_{p2})^2 + (X_{i3} - X_{p3})^2}$$

donde d_i es la distancia entre el i-ésimo punto definido por las coordenadas X_{i1}, X_{i2}, X_{i3} dadas en la i-ésima fila de la tabla, y el punto considerado dado por las coordenadas $X_{p1} = 1, X_{p2} = 1, X_{p3} = 1$.

En R se pueden obtener las distancias correspondientes mediante el siguiente código, el cual lleva a cabo el cálculo de forma vectorizada (no necesitando el uso de bucle `for`):

```
coordPunto1 <- c(1, 1, 1)
distancias <- sqrt(rowSums((df[,1:3] - coordPunto1)^2))
print(distancias)
```

```
## [1] 2.236068 1.414214 1.732051 2.236068 1.732051 2.449490
```

Se concluye que la observación 6 se encuentra más lejos del punto dado, mientras que la observación 2 es la más cercana a dicho punto.

b) Con $K = 1$ se considera sólamente un solo vecino más cercano, el cual es el punto 2 de coordenadas (0, 1, 2). En este caso, la predicción para el punto considerado sería `Y=Mucho` dado que se cuenta con un solo punto para realizar la estimación cuyo valor es `Y=Mucho`.

c) Con $K = 3$ se consideran tres vecinos más cercanos, a saber: los puntos 2, 3, y 5 de la tabla (en este caso es sencillo determinar los puntos más cercanos a simple vista, sino sería necesario usar una función de R), cuyas respuestas son `Mucho`, `Mucho`, y `Poco`, respectivamente. Dado que de las tres respuestas dos adoptan el valor `Mucho`, la proporción es 2/3 - 1/3, con lo cual, $P(y = 1|X = (1, 1, 1)) = 0.\hat{6}$. Entonces, la predicción de la respuesta para el punto considerado es también `Y=Mucho`.

d) Para este caso, se debe realizar un bucle for (....)

```
filasMenos1 <- nrow(df) - 1
prediccion <- data.frame(matrix(ncol = 2, nrow = filasMenos1))
x <- c("Predicción para K=1", "Predicción para K=3")
colnames(prediccion) <- x

for (i in 1:filasMenos1) {
    coords_i <- df[1,1:3] %>% as.numeric()
    sub_df <- df[-i,]
    distancias_i <- sqrt(rowSums((sub_df[,1:3] - coords_i)^2))
    indiceK1_i <- which.min(distancias_i)[1]
    prediccion[i,1] <- sub_df[indiceK1_i,4]
    indiceK3_i <- order(distancias_i)[1:3]
    subset <- sub_df[indiceK3_i,4]
    prom <- sum(subset == "Mucho")/3
    pred_K3 <- if (prom >= 0.5){
        'Mucho'
    } else {
        'Poco'
```

```
    }
    prediccion[i,2] <- pred_K3
}
print(prediccion)
```

```
##   Predicción para K=1 Predicción para K=3
## 1               Mucho                Poco
## 2                Poco                Poco
## 3               Mucho                Poco
## 4                Poco                Poco
## 5                Poco                Poco
```