Práctica 2 - Limpieza y Análisis de datos

Tipología y ciclo de vida de los datos | 2022/23 - 2

Jose Luis Santos Durango | María Isabel González Sánchez

15 de junio, 2023

Contents

1. Descripción del dataset. Integración y selección

En la primera práctica, nuestro *Web-Scraping* se centró en la web **Expatistan**, una página que nos ofrece información sobre el coste de vida en distintos países del mundo y distintas ciudades, así como la opción de realizar comparativas. Este tema surgió desde el interés común en los integrantes del grupo por los viajes y, por ende, la necesidad de prepararse para tales aventuras. Gracias a su información detallada en temas de comida, alojamiento y transporte entre otros y el uso de *crowdsourcing* como método de obtención de datos, decidimos decantarlos por focalizar todos nuestros esfuerzos en ella.

Una vez puestos en marcha, pudimos apreciar su potencial estadístico. Tanto para los apartados de países, como para los apartados de ciudades, se nos ofrecían un conjunto de variables útiles e interesantes, que nos generaron varias preguntas detalladas en la primera práctica. Sin embargo, para esta práctica nos centraremos solo en uno de los conjuntos de datos extraído, dada la similitud entre ambos: **"cost_of_living_countries.csv"**. Este presenta las siguientes variables:

1. **Ranking.position**: posición numérica del ranking de países de mayor a menor coste de vida.
2. **Country**: nombre del país al que pertenecen el coste de vida detallado en los *Items*.
3. **Category**: clasificación cualitativa que engloba un conjunto de *Items*.
4. **Items**: productos o servicios ofrecidos en el país de los cuales queremos saber su coste.
5. **Original.Currency**: abreviatura en mayúsculas de la moneda local utilizada por cada país.
6. **Original.Currency.Value**: valor monetario en la divisa local de un *Item* concreto.
7. **Exchanged.Currency**: divisa común para uso estadístico en el caso de hacer comparativas entre países que, por defecto, es el Euro.
8. **Exchanged.Currency.value**: valor monetario en Euros de un *Item* concreto.

Además, hemos considerado apropiado ampliarlo con otros datos relevantes: el **PIB anual** y el **Salario Mínimo Interprofesional**. Los dos provienen de la web Datosmacro Expansion, donde la información viene catalogada por país, así como la **Cotización de Divisas frente al Euro** que usaremos para ciertas conversiones. Dado que este nuevo dataset requiere de uniones y demás acciones no relevantes a la práctica (sin tener en cuenta la limpieza que será en el siguiente apartado), se ha adjuntado un notebook de Python con su creación.

En cuanto a las preguntas, recuperaremos y reformularemos algunas de las planteadas anteriormente en la práctica 1:

1. Un estudiante quiere decidir si ir de Erasmus por Europa o América y necesita estudiar el alquiler en los países que los conforman. Por ende, ¿un mes de alquiler en una zona normal es significativamente mayor en Europa que en América?
2. ¿Hay alguna relación entre las categorías de gastos básicos y el salario mínimo de un país? ¿Y entre el PIB anual?
3. Una vez acabada la carrera y el Máster, queremos recorrer mundo y trabajar en un país con un SMI más alto que España. ¿Cuáles serían los países candidatos?

2. Limpieza de los datos

2.1 Carga de los ficheros de datos

```
# Cargamos el fichero de datos
df_countries <- read.csv("../datasets/cost_of_living_countries_updated.csv", sep=",")

# Veamos la dimensión del fichero
dim(df_countries)
```

```
## [1] 3774   11
```

```
# veamos como identifica R cada variable del dataset
str(df_countries)
```

```
## 'data.frame':    3774 obs. of  11 variables:
##  $ Ranking.position        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Country                 : chr  "BERMUDA" "BERMUDA" "BERMUDA" "BERMUDA" ...
##  $ Category                : chr  "Food" "Food" "Food" "Food" ...
##  $ Items                   : chr  "Basic lunchtime menu (including a drink) in the business district
##  $ Original.Currency       : chr  "BMD" "BMD" "BMD" "BMD" ...
##  $ Original.Currency.Value : chr  "BD$36" "BD$21" "BD$24" "BD$4.98" ...
##  $ Exchanged.Currency      : chr  "EUR" "EUR" "EUR" "EUR" ...
##  $ Exchanged.Currency.Value: chr  "(€32)" "(€19)" "(€22)" "(€4.49)" ...
##  $ PIB.anual               : chr  "" "" "" "" ...
##  $ SMI..dolares.           : chr  "" "" "" "" ...
##  $ SMI..euros.             : chr  "" "" "" "" ...
```

2.2 Conversión columnas de valores a tipo cuantitativas

Como hemos podido observar, las columnas que guardan la información del coste de las categorías/productos, contienen la información entre paréntesis y con el símbolo de la moneda, lo cual hace que el campo sea considerado como un string en lugar de un número. Eliminaremos estos caracteres y convertiremos las columnas a valores numéricos.

```
# Convertir las columnas a valores numéricos y eliminar símbolos de moneda y paréntesis
df_countries$Original.Currency.Value <- as.numeric(gsub("[^0-9.]",
                                                        "",
                                                        df_countries$Original.Currency.Value))
df_countries$Exchanged.Currency.Value <- as.numeric(gsub("[^0-9.]",
                                                         "",
                                                         df_countries$Exchanged.Currency.Value))
```

Además, si nos centramos en el SMI y el PIB, presentan caracteres no propios y necesitan una conversión y limpieza. Primero, el PIB anual necesitamos eliminar el caracter de millón y la divisa, para luego multiplicar el número resultante por 10^6. Luego a los SMI en euros y dólares se les debe quitar también la divisa y convertirlos a tipo numérico:

```
# Eliminaremos los caracteres " M€" y transformaremos los dígitos resultantes a un número
df_countries$PIB.anual <- as.numeric(gsub("[^0-9]", "", df_countries$PIB.anual))

# Dado que el PIB anual es en millones de euros, calcularemos el valor real
df_countries$PIB.anual <- df_countries$PIB.anual * 1000000

# Eliminaremos las divisas, transformaremos la coma decimal y pasamos a un número
df_countries$SMI..dolares. <- gsub("\\.|\\$", "", df_countries$SMI..dolares.)
df_countries$SMI..dolares. <- gsub(",", ".", df_countries$SMI..dolares.)
df_countries$SMI..dolares. <- as.numeric(gsub("[^0-9.]", "", df_countries$SMI..dolares.))

df_countries$SMI..euros. <- gsub("\\.|\\€", "", df_countries$SMI..euros.)
df_countries$SMI..euros. <- gsub(",", ".", df_countries$SMI..euros.)
df_countries$SMI..euros. <- as.numeric(gsub("[^0-9.]", "", df_countries$SMI..euros.))
```

2.3 Estudio de las variables categóricas y reordenación de columnas

En este apartado vamos a ver los valores distintos de cada una de las columnas categóricas, con el fin de estudiar si pudiera haber algún error tipográfico que genere una categoría/subcategoría/moneda innecesaria.

```r
# Estudio de los distintos valores de categoría/subcategoría
categories <- sort(unique(df_countries$Category))
items <- sort(unique(df_countries$Items))
eur <- sort(unique(df_countries$Exchanged.Currency))
```

Al ordenar de forma alfabética hemos podido ver que no hay ninguna categoría errónea que sea derivada de alguna correcta, por lo que no haremos ningún cambio. Además dado que la moneda de cambio es siempre el Euro, eliminaremos esta columna para ahorrar tiempo de procesamiento y añadiremos esa información al nombre del campo directamente.

```r
# Eliminamos la columna Exchanged.Currency
df_countries <- subset(df_countries, select = -Exchanged.Currency)
colnames(df_countries)[7] <- "EUR.Value"
```

2.4 Estudio de valores nulos

```r
# veamos los valores NA
na_counts <- colSums(is.na(df_countries))
print(na_counts)
```

```
##        Ranking.position                 Country                Category
##                       0                       0                       0
##                   Items       Original.Currency Original.Currency.Value
##                       0                       0                      48
##               EUR.Value               PIB.anual           SMI..dolares.
##                      22                     612                     918
##             SMI..euros.
##                    1020
```

Podemos ver que hay ciertas categorías en las que el valor en EUR existe, pero sin embargo el valor en la moneda original no está registrado. Esto puede deberse a un error en el scraping de la página web ya que el valor original es en la moneda original, y para convertirlo a EUR se aplica un cambio de divisa registrado, tal como se explicó en la PRA1, por lo que esos registros los vamos a eliminar para evitar tener datos contaminados.

```r
# eliminar registros con NA en las columnas: Original.Currency.Value, EUR.Value
df_countries <- df_countries[!is.na(df_countries$Original.Currency.Value) |
                             is.na(df_countries$EUR.Value), ]
```

Sin embargo, para los registros donde ambos valores aparecen como nulos vamos a realizar una imputación por sus vecinos más cercanos, a partir de la variable numérica que corresponde al ranking, y a la misma subcategoría, es decir consideraremos el valor para ese *Item* del país más cercano en el ranking que tenga un valor no nulo. Para ello seguiremos los siguientes pasos:

1. Factorizamos la variable *Items* para poder usar el método kNN (no funciona con variables categóricas).
2. Aplicamos kNN para calcular cuáles serán los registros que se consideran como vecinos más cercanos.
3. Sustituimos los valores nulos de la columna *EUR.Value* por los valores de los registros que nos da el algoritmo kNN.

```r
# Factorizamos la variable "Items"
df_countries$Items_factor <- as.numeric(as.factor(df_countries$Items))

# Definimos una función para aplicar el algoritmo kNN
imputar_valores_na <- function(variable, df_countries, cuantitative_variables, vecinos) {
  # Aplicamos el algoritmo kNN
  new_df_countries <- kNN(df_countries[, cuantitative_variables], variable = variable, k = vecinos)
```

```
  # Veamos qué filas presentan valores nulos en la columna a tratar
  rows_na <- which(is.na(df_countries[[variable]]))

  # Imputamos los valores NA usando el vecino más cercano por categoría y ranking
  df_countries[rows_na, ][[variable]] <- new_df_countries[new_df_countries[[paste0(variable, "_imp")]] =

  # Devolvemos el dataset actualizado
  return(df_countries)
}

# Definimos que variables cuantitativas usaremos
cuantitative_variables <- which( colnames(df_countries) %in% c("Ranking.position",
                                                               "Items_factor",
                                                               "EUR.Value"))
# Actualizamos el dataset
df_countries <- imputar_valores_na("EUR.Value",
                                   df_countries,
                                   cuantitative_variables,
                                   1)
```

Aunque los valores originales estén ya limpios, entramos en un nuevo problema de valores nulos: el PIB y el SMI. No todos los países del dataset original están contemplados en la web de Datosmacro. En este caso, optaremos por eliminar los registros donde no tengamos el SMI en dólares evitando trabajar con países en los que no tenemos el salario mínimo, y una vez eliminados, imputaremos el PIB en caso de no existir en el país, por el método kNN descrito anteriormente.

```
# Eliminamos registros de países sin SMI en dólares
df_countries <- df_countries[!is.na(df_countries$SMI..dolares.), ]

# Definimos que variables cuantitativas usaremos para imputar el PIB
cuantitative_variables <- which(colnames(df_countries) %in% c("Ranking.position",
                                                              "PIB.anual",
                                                              "SMI..dolares."))
# Rellenamos los valores del PIB
df_countries <- imputar_valores_na("PIB.anual",
                                   df_countries,
                                   cuantitative_variables,
                                   11)
```

Por último, nos queda la transformación a euros y la haremos usando el último dataset extraído de Datosmacros: la Cotización de divisas.

```
# Cargamos el dataset de divisas
df_divisas <- read_excel("../datasets/PIB_SMI_divisas.xlsx", sheet="Divisas")

# Obtenemos la tasa de conversión de euros a dolares
tasa_conversion <- df_divisas$Cambio[df_divisas$Países == "Dólares USA [+]"]

# Rellenamos los nulos
df_countries$SMI..euros. <- ifelse(is.na(df_countries$SMI..euros.) | df_countries$SMI..euros. == 0,
                                   df_countries$SMI..dolares. / tasa_conversion,
                                   df_countries$SMI..euros.
                                   )

# Dado que nuestros estudios serán en Euros para partir de un marco común, eliminaremos todas las colum
```

```r
cols_eliminar <- c("Original.Currency",
                   "Original.Currency.Value",
                   "SMI..dolares.")

df_countries <- df_countries %>% select(-all_of(cols_eliminar))

# Renombraremos SMI..euros. como SMI
df_countries <- df_countries %>% rename(SMI = `SMI..euros.`)

write.csv(df_countries, "../datasets/cost_of_living_countries_clean.csv", row.names = FALSE)

na_counts <- colSums(is.na(df_countries))
print(na_counts)
```

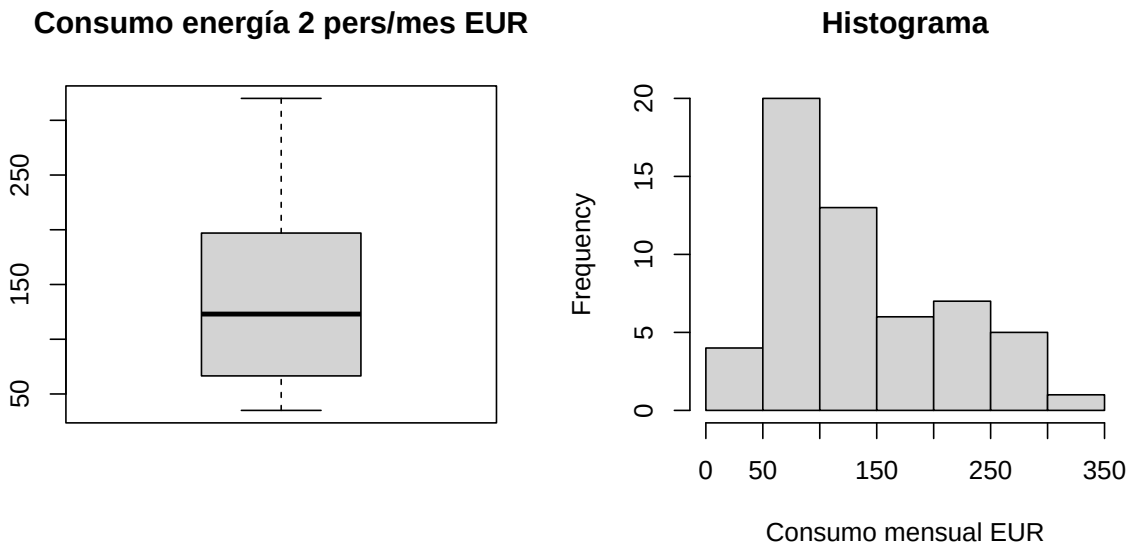
```
## Ranking.position          Country         Category            Items
##                0                0                0                0
##        EUR.Value       PIB.anual              SMI     Items_factor
##                0                0                0                0
```

2.5 Estudio de outliers

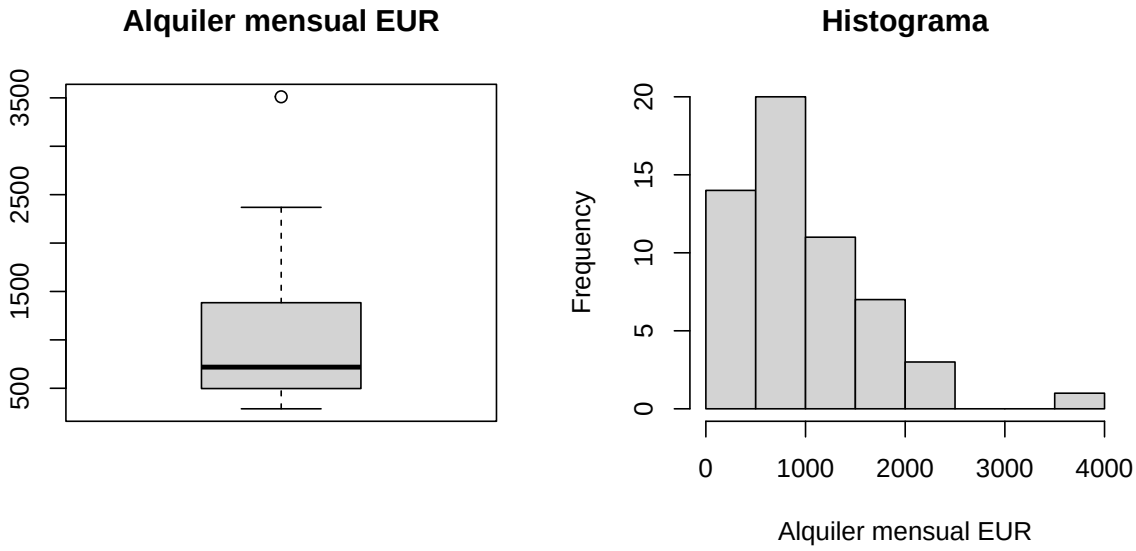
Dado que tenemos un dataset donde el análisis puede ser bastante extenso debido a la gran cantidad de categorías y países que tenemos, hay que prestar especial atención en qué queremos comparar y qué datos vamos a considerar. Por ejemplo, no tiene mucho sentido estudiar datos del precio de un vehículo y compararlos con el precio del abono mensual (ambos dentro de la categoría *Transportation*). Por este motivo, seleccionaremos 5 campos en los que vamos a basar nuestro análisis para estudiar la existencia de posibles outliers:

- Utilities 1 month (heating, electricity, gas ...) for 2 people in $85m^2$ flat
- Monthly rent for $85m^2$ (900 sqft) furnished accommodation in normal area
- Categoría Food
- SMI
- PIB.anual

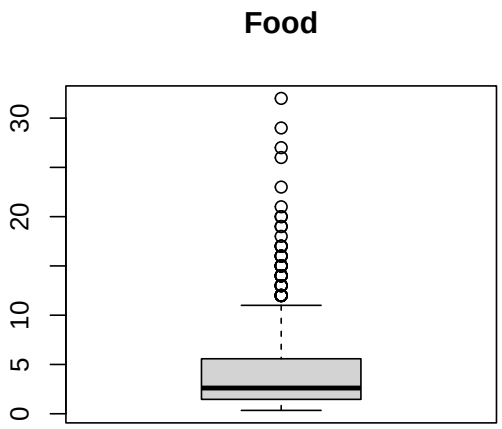
```r
# Outliers item: Utilities 1 month (heating, electricity, gas ...) for 2 people in 85m2 flat
par(mfrow=c(1,2))
df_a<-subset(df_countries,
             Items == "Utilities 1 month (heating, electricity, gas ...) for 2 people in 85m2 flat")
boxplot(df_a$EUR.Value, main="Consumo energía 2 pers/mes EUR")
hist(df_a$EUR.Value, main="Histograma",xlab="Consumo mensual EUR")
```

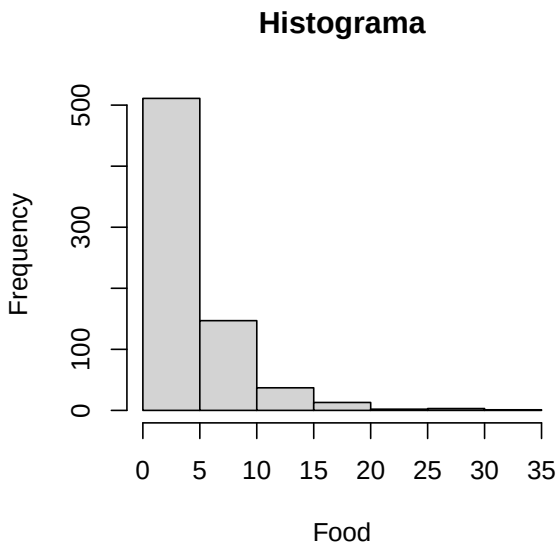



```
# Outliers item: Monthly rent for 85 m2 (900 sqft) furnished accommodation in normal area
par(mfrow=c(1,2))
df_b<-subset(df_countries,
             Items == "Monthly rent for 85 m2 (900 sqft) furnished accommodation in normal area")
boxplot(df_b$EUR.Value, main="Alquiler mensual EUR")
hist(df_b$EUR.Value, main="Histograma",xlab="Alquiler mensual EUR")
```

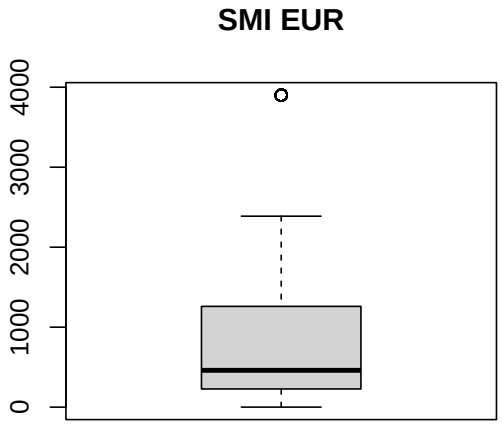


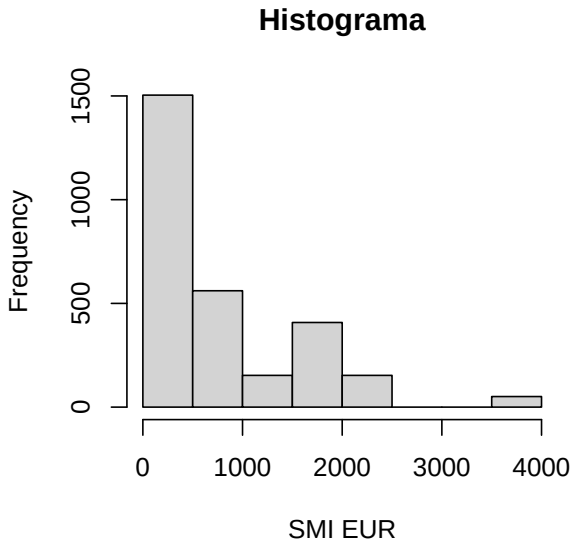

```
# Outliers category: Food
par(mfrow=c(1,2))
df_c<-subset(df_countries,
             Category == "Food")
boxplot(df_c$EUR.Value, main="Food")
hist(df_c$EUR.Value, main="Histograma",xlab="Food")
```



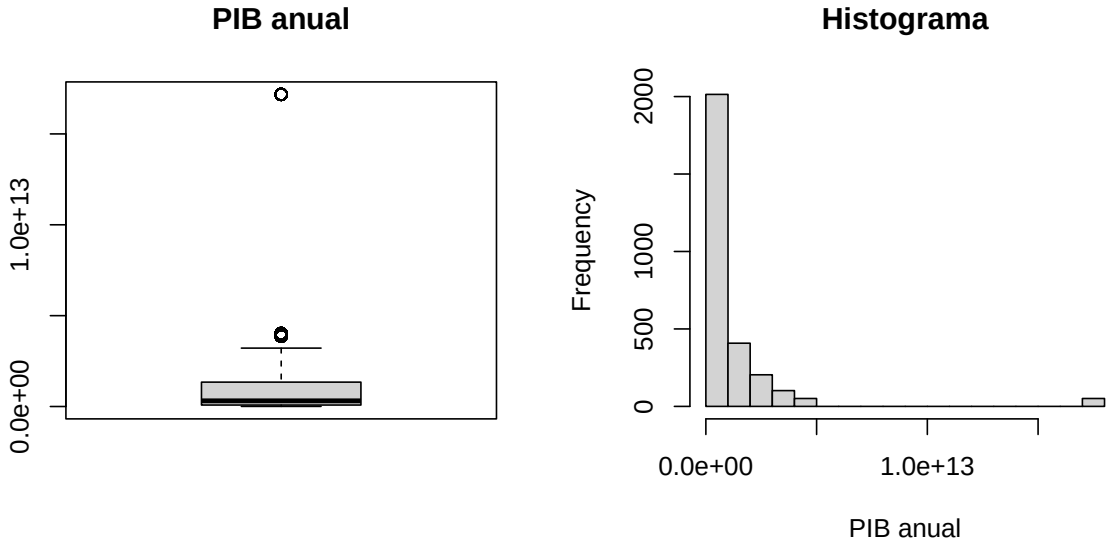



```
# Outliers: SMI
par(mfrow=c(1,2))
boxplot(df_countries$SMI, main="SMI EUR")
hist(df_countries$SMI, main="Histograma",xlab="SMI EUR")
```






```
# Outliers: PIB
par(mfrow=c(1,2))
boxplot(df_countries$PIB.anual, main="PIB anual")
hist(df_countries$PIB.anual, main="Histograma",xlab="PIB anual")
```



En las categorías anteriores podemos observar algunos *outliers* a considerar que pueden influir en el análisis.

```
# Filtramos el dataset aplicando unas condiciones para cada Item
df_countries <- filter(df_countries, Items_factor == 42 & EUR.Value <= 2600 | Items_factor != 42) # Uti
df_countries <- filter(df_countries, Category == "Food" & EUR.Value <= 10 | Category != "Food") #Food
df_countries <- filter(df_countries, SMI <= 2500) # SMI
df_countries <- filter(df_countries, PIB.anual <= 1e13) # PIB.anual
```

3. Guardamos los datos pre-procesados

```
# Guardamos los datos preprocesados en un fichero
write.csv(df_countries, "../datasets/cost_of_living_countries_clean.csv", row.names = FALSE)
```

4. Análisis de los datos

4.1 Datos a comparar. Valores mínimos de gasto medio por categoría.

Con el objetivo de realizar un análisis generalizado del coste de vida por países, vamos a generar un dataset con los promedios de gasto para cada una de las 6 categorías que existen, independientemente del *Item*, para así poder realizar inferencia estadística (test de hipótesis, estimación de parámetros, intervalos de confianza) entre las distintas categorías o países.

```
# Creamos el dataset con las columnas que usaremos
df_countries_2 <- df_countries[,c(1,2,3,5,6,7)]

# Agrupamos los datos y calculamos el valor medio de la columna EUR.Value
df_countries_avg <- aggregate(df_countries_2$EUR.Value, by = list(df_countries_2$Ranking.position,
                                                                  df_countries_2$Country,
                                                                  df_countries_2$Category,
                                                                  df_countries_2$PIB.anual,
                                                                  df_countries_2$SMI),
                              FUN = mean)

colnames(df_countries_avg) <- c("Ranking.position", "Country", "Category", "PIB.anual", "SMI", "€.Avg.E
```

```
df_countries_summary <- spread(df_countries_avg,
                               Category,
                               `€.Avg.Expense`)
str(df_countries_summary)
```

```
## 'data.frame':    54 obs. of  10 variables:
##  $ Ranking.position: int  5 6 7 11 12 13 14 15 16 17 ...
##  $ Country         : chr  "BAHAMAS" "HONG KONG" "IRELAND" "NETHERLANDS" ...
##  $ PIB.anual       : num  9.47e+09 3.12e+11 5.03e+11 9.41e+11 7.81e+10 ...
##  $ SMI             : num  804 734 1910 1934 2387 ...
##  $ Clothes         : num  97.5 52.2 75.8 80.5 84 ...
##  $ Entertainment   : num  139.5 64.1 85.4 98.8 42 ...
##  $ Food            : num  4.66 3.8 3.77 3.77 3.87 ...
##  $ Housing         : num  729 1059 816 721 801 ...
##  $ Personal Care   : num  20.95 8.23 14.43 12.21 10.83 ...
##  $ Transportation  : num  2911 13052 10437 9571 9847 ...
```

Como ya hemos comentado, el objetivo de este dataset es poder realizar una inferencia estadística sobre los distintos países o sobre la media de gastos básicos en una categoría concreta. Debemos aclarar aquí, que los valores almacenados para cada categoría *no* indican el valor medio de gasto en el país en cuestión para esa categoría, si no un cálculo aproximado del valor medio que se podría gastar en cada categoría si se realizara/adquiriese cada uno de los *Items* detallados en el conjunto de datos original. Es decir, que el gasto medio de transporte en Bermuda sea 11714.35€, no quiere decir que se gaste eso de media en transporte, pero sí consideramos el promedio de comprar un coche, comprar un litro de gasolina y adquirir un abono transporte. Debemos considerar estos valores como un indicador de la categoría, y no como un gasto medio de la misma.

4.2 Normalidad y Homogeneidad de la varianza

Evaluar la normalidad de los datos y la homogeneidad de las varianzas es importante para asegurarnos de que los análisis estadísticos que realizamos son válidos y que podemos interpretar los resultados de manera adecuada. Estas evaluaciones nos permiten confirmar si los supuestos necesarios para aplicar ciertas técnicas estadísticas se cumplen. Si los datos no siguen una distribución normal o si las varianzas no son similares entre grupos, es posible que los resultados de nuestros análisis no sean confiables. Por lo tanto, al verificar la normalidad de los datos y la homogeneidad de las varianzas, estamos asegurándonos de que nuestras conclusiones se basen en fundamentos sólidos y que nuestras inferencias y decisiones sean acertadas.

4.2.1 Normalidad

4.2.1.1 Normalidad de las variables medias

Estudiemos la normalidad de las variables resultantes en el dataset con los valores medios.

```
# Normalidad: PIB.anual, Food, Clothes, Entertainment, Housing, SMI
shapiro.test(df_countries_summary$PIB.anual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_countries_summary$PIB.anual
## W = 0.73382, p-value = 1.476e-08
```

```
shapiro.test(df_countries_summary$Food)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  df_countries_summary$Food
## W = 0.85818, p-value = 1.361e-05
```

```
shapiro.test(df_countries_summary$Clothes)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_countries_summary$Clothes
## W = 0.9854, p-value = 0.7502
```

```
shapiro.test(df_countries_summary$Entertainment)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_countries_summary$Entertainment
## W = 0.96029, p-value = 0.071
```

```
shapiro.test(df_countries_summary$Housing)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_countries_summary$Housing
## W = 0.89048, p-value = 0.0001353
```

```
shapiro.test(df_countries_summary$SMI)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_countries_summary$SMI
## W = 0.82555, p-value = 1.75e-06
```

Podemos comprobar que en los test anteriores, las variables: *PIB.anual*, *Food*, *Housing* y *SMI* siguen distribuciones normales mediante el test de **Shapiro-Wilk** con un p-valor menor que un nivel de significancia $\alpha = 0.05$. En las variables *Entertainment* y *Clothes* no podemos decir lo mismo, a pesar de que en el caso de nuestra población, al ser una muestra grande con n>30 podríamos asumir la hipótesis de normalidad de las medias por el teorema del límite central.

4.2.1.2 Normalidad de la variable EUR.Value para el Item: Monthly rent for $85m^2$ (900 sqft) furnished accommodation in normal area

Ahora comprobaremos la normalidad para la variable *EUR.value* para el *Item* que querremos estudiar en los próximos análisis:

```
# Recogemos las filas que traten sombre este Item
df_countries_test <- filter(df_countries, Items == "Monthly rent for 85 m2 (900 sqft) furnished accommo

# Normalidad: EUR.Value
shapiro.test(df_countries_test$EUR.Value)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  df_countries_test$EUR.Value
## W = 0.88564, p-value = 0.0001084
```

Como se puede observar, el p-value resultante es menor al valor de significancia $\alpha = 0.05$, indicando normalidad en la variable.

4.2.2 Homocedasticidad

Comprobemos si la variable *EUR.Value* cumple la igualdad de varianzas entre dos grupos a comparar para poder aplicar contraste de hipótesis a los datos. Para ello, escogeremos dos Items similares que queremos estudiar y aplicaremos el Test de Levene:

- "Monthly rent for 85 m2 (900 sqft) furnished accommodation in normal area"
- "Monthly rent for 85 m2 (900 sqft) furnished accommodation in expensive area"

```
# Conseguimos los dos grupos de los cuales queremos estudiar su homogeneidad de varianzas
df_countries_homocedasticidad <- filter(df_countries, Items  %in% c("Monthly rent for 85 m2 (900 sqft) i

# Homocedasticidad: EUR.Value en función de Items
leveneTest(y = df_countries_homocedasticidad$EUR.Value, group = df_countries_homocedasticidad$Items, cen
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value Pr(>F)
## group   1  2.4585 0.1199
##       105
```

Para estos resultados del **Test de Levene**, podemos observar que su $p-value = 0.1199$ resultante es mayor al nivel de significancia $\alpha = 0.05$. Por ende, no tenemos pruebas suficientes como para rechazar la hipótesis nula, es decir, las varianzas no difieren significativamente entre ellas y podemos asumir homogeneidad de varianza entre los 2 grupos de alquiler.

4.3 Pruebas estadísticas

Dentro de todas las pruebas estadísticas que hay, nos hemos decantado por aquellas que nos ayudarán a resolver las preguntas planteadas en el apartado uno: el contraste de hipótesis, la regresión lineal y la regresión logística.

4.3.2 Contraste de hipótesis

Queremos estudiar si un alquiler en una zona normal, es decir, el *Item* "Monthly rent for 85 m2 (900 sqft) furnished accommodation in normal area" es significativamente mayor en Europa que en América. Para ello, lo mejor es hacer un contraste de hipótesis al 95% de confianza. Por ende, podemos establecer las siguientes hipótesis nula H_0 y alternativa H_1:

$H_0 : \theta_0 = \theta_1$
$H_1 : \theta_0 > \theta_1$

En este caso tendremos que:

1. H_0 establece que "la media del alquiler mensual en un zona normal en Europa es **igual** que la misma en América".
2. H_1, "la media del alquiler mensual en un zona normal en Europa es **mayor** que en América".
3. θ_0 es el valor monetario medio en euros del alquiler en Europa.

4. θ_1 es el valor monetario medio en euros del alquiler en América.

Para este estudio, realizaremos un **contraste de dos muestras independientes sobre la media con varianzas conocidas**, siendo esta muestras independientes θ_0 y θ_1. La idea es refutar la hipótesis nula basándonos en evidencia positiva y superior al umbral de aceptación establecido: a base de test con un nivel de confianza del 95%.

Usaremos este tipo de contraste ya que la pregunta nos plantea dos muestras que definiremos uniformemente, es decir, escogeremos 5 países de cada continente para conformar sus muestras. De estas mismas, podemos calcular su media sabiendo el número de países involucrados y, por ende, su varianza será conocida. El procedimiento a seguir será aplicar el **Teorema del Límite Central** para intentar rechazar la hipótesis nula. De esta forma, comenzaremos por un test unilateral por la derecha, donde la zona de aceptación de la hipótesis nula estará comprendida entre (-∞, $z_{1-\infty}$]:

```
# Definimos la función para calcular el valor observado, el valor crítico y el valor p
def_valores_alquiler <- function(theta_0, theta_1, alpha) {
    # Primero conseguimos la media y la varianza de cada muestra
    alquiler_mean_EUR <- mean(theta_0)
    alquiler_mean_AME <- mean(theta_1)
    alquiler_var_EUR <- var(theta_0, na.rm = TRUE)
    alquiler_var_AME <- var(theta_1, na.rm = TRUE)

    # Luego calculamos el valor observado
    n_EUR <- length(theta_0)
    n_AME <- length(theta_1)
    valor_obs <- (alquiler_mean_EUR - alquiler_mean_AME) / sqrt(alquiler_var_EUR / n_EUR + alquiler_var

    # Calculamos el valor crítico
    valor_crit <- qnorm(alpha, lower.tail = FALSE)

    # Calcular el valor p
    valor_p <- pnorm(valor_obs, lower.tail = FALSE)

    # Devolvemos los valores obtenidos
    return(c(valor_obs, valor_crit, valor_p))
}
# Primero definiremos las muestras
df_Europa <- subset(df_countries,
                    Items == "Monthly rent for 85 m2 (900 sqft) furnished accommodation in normal area"
                    & Country %in% c("FRANCE", "ITALY", "FINLAND", "SWITZERLAND", "GERMANY"))
df_America <- subset(df_countries,
                    Items == "Monthly rent for 85 m2 (900 sqft) furnished accommodation in normal area"
                    & Country %in% c("UNITED STATES", "MEXICO", "CANADA", "URUGUAY", "BOLIVIA"))

# Calculamos los valores al 95% de confianza y lo mostramos por pantalla
valores_alquiler <- def_valores_alquiler(df_Europa$EUR.Value, df_America$EUR.Value, 0.05)
cat("El valor observado: ", valores_alquiler[1],
    "\nEl valor crítico: ", valores_alquiler[2],
    "\nEl valor p al 95%: ", valores_alquiler[3], "")
```

```
## El valor observado:  2.964443
## El valor crítico:  1.644854
## El valor p al 95%:  0.001516159
```

4.3.1 Regresión lineal

En este apartado vamos a realizar una regresión lineal para estudiar el *SMI* de un país en función de los valores medios de las categorías que siguen una distribución normal: *Food*, *Housing*, *PIB.anual*. De esta forma, a partir de los valores medios de gastos mínimos (recordamos que el dataset no guarda los valores medios de gasto, si no una muestra de gasto mínimo) en un país determinado, podremos dar respuesta a la pregunta 2.

```
# Generamos el modelo lineal
model_SMI <- lm(SMI ~ Food + Housing + PIB.anual, data = df_countries_summary)
summary(model_SMI)
```

```
##
## Call:
## lm(formula = SMI ~ Food + Housing + PIB.anual, data = df_countries_summary)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1228.17  -242.05    39.31   213.85  1061.40
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.572e+02  2.798e+02  -1.634 0.108526
## Food         7.340e+01  9.415e+01   0.780 0.439308
## Housing      1.954e+00  3.733e-01   5.235 3.29e-06 ***
## PIB.anual    2.265e-10  6.164e-11   3.675 0.000581 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 456.2 on 50 degrees of freedom
## Multiple R-squared:  0.5852, Adjusted R-squared:  0.5603
## F-statistic: 23.51 on 3 and 50 DF,  p-value: 1.239e-09
```

Si queremos interpretar la calidad del ajuste de este modelo, debemos estudiar el **Coeficiente de Determinación** R^2, que consiste en la proporción de variabilidad explicada por el modelo con respecto a la variabilidad total. Lo que buscaremos es que este coeficiente se aproxime lo máximo a 1, ya que indicará que el modelo hace un buen ajuste. Dentro del *summary* del modelo, lo podremos encontrar bajo el nombre "Multiple R-squared" y es 0.5852. Esto nos muestra que el 58.52% de la variable dependiente **SMI** puede ser explicada por las variables independientes **Food**, **Housing** y **PIB.anual**.

Por otro lado, observamos que solo las categorías *Housing* y *PIB.anual* tienen una influencia significante en la definición del salario mínimo de un país ya que su *p-value* es menor al nivel de significancia $\alpha = 0.05$. sin embargo, no podemos decir los mismo de la categoría *Food*, al menos con los datos recogidos en nuestro dataset. Esto nos explica por que el R^2 llega solo a casi el 60%: faltan más variables para completar el concepto del *SMI*, aunque es un ajuste bastante decente y que puede dar resultados.

4.3.2 Regresión logística

Para poder empezar el último análisis estadístico, crearemos una nueva variable dicotómica para ser nuestra variable dependiente: SMI_bin. Esta nueva variable está relacionada con los valores de la variable **SMI** y se codificará de la siguiente forma: "1", para valores de **SMI** superiores al SMI de España, y "0" con valores de **SMI** inferiores al de España. Vamos a crearla:

```
#Extraemos el SMI de España
SMI_españa <- df_countries_summary$SMI[df_countries_summary$Country == "SPAIN"]

# Crearemos la nueva variable dicotómica SMI_bin
df_countries_summary$SMI_bin <- ifelse(df_countries_summary$SMI > SMI_españa, 1, 0)
```

Dado que queremos definir un modelo de regresión logística, el siguiente paso es separar los datos en dos conjuntos aleatorios: el conjunto de entrenamiento (training, 80% de los datos) y el conjunto de prueba (testing, 20% de los datos). Esto nos ayudará a estimar de forma más objetiva la precisión del modelo. Por ende, ajustaremos el modelo con el conjunto de entrenamiento, y evaluaremos la precisión con el conjunto de prueba.

```
# Fijaremos una semilla aleatoria para reproducibilidad
set.seed(27)

# Dividiremos el conjunto de datos original en conjuntos de entrenamiento y prueba
train_test <- createDataPartition(df_countries_summary$SMI_bin, p = 0.80 , list = FALSE)

# Para ello nos quedaremos como Train las filas del conjunto de datos que pertenezcan a la partición
train_dataset <- df_countries_summary[train_test, ]

# Y dejaremos el resto como Test
test_dataset <- df_countries_summary[-train_test, ]
```

Dado que la categoría *Food* no ayudaba al modelo lineal, para el modelo logístico la hemos intercambiado por *Clothes* para estudiar si ayuda al ajuste de la nueva variable *SMI_re* o no:

```
# Definiremos el modelo logístico con SMI_bin como variable dependiente y las categorías y el PIB, como
modelo_SMI_bin <- glm(SMI_bin ~ PIB.anual + Clothes + Housing, train_dataset, family = binomial)

# Mostraremos los valores resultantes
summary(modelo_SMI_bin)
```

```
##
## Call:
## glm(formula = SMI_bin ~ PIB.anual + Clothes + Housing, family = binomial,
##     data = train_dataset)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.13042  -0.24267  -0.10314  -0.01784   1.65898
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.617e+01  5.560e+00  -2.908  0.00363 **
## PIB.anual    1.610e-12  6.506e-13   2.474  0.01335 *
## Clothes      1.351e-01  5.783e-02   2.336  0.01950 *
## Housing      7.062e-03  3.217e-03   2.195  0.02815 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 47.164  on 43  degrees of freedom
## Residual deviance: 18.567  on 40  degrees of freedom
## AIC: 26.567
##
## Number of Fisher Scoring iterations: 7
```

A diferencia de *Food*, el p-value de *Clothes* es menor al valor de significancia $\alpha = 0.05$. Por ende, se debería

tener en cuenta junto con *Housing* y *PIB.anual* que también cumplen la condición. Por otro lado, al ser un modelo de regresión de logística, el valor que debemos estudiar es el AIC: cuanto más cerca del 0, mejor se ajusta el modelo a los datos aportados. Como vemos el $AIC = 26.567$ es bastante cercano a cero, indicando un buen ajuste.

Ahora analizaremos dicho ajuste con el conjunto de validación. Para ello, predeciremos las etiquetas de *SMI_bin* para los valores restantes y estudiaremos tanto su matriz de confusión como su curva ROC:

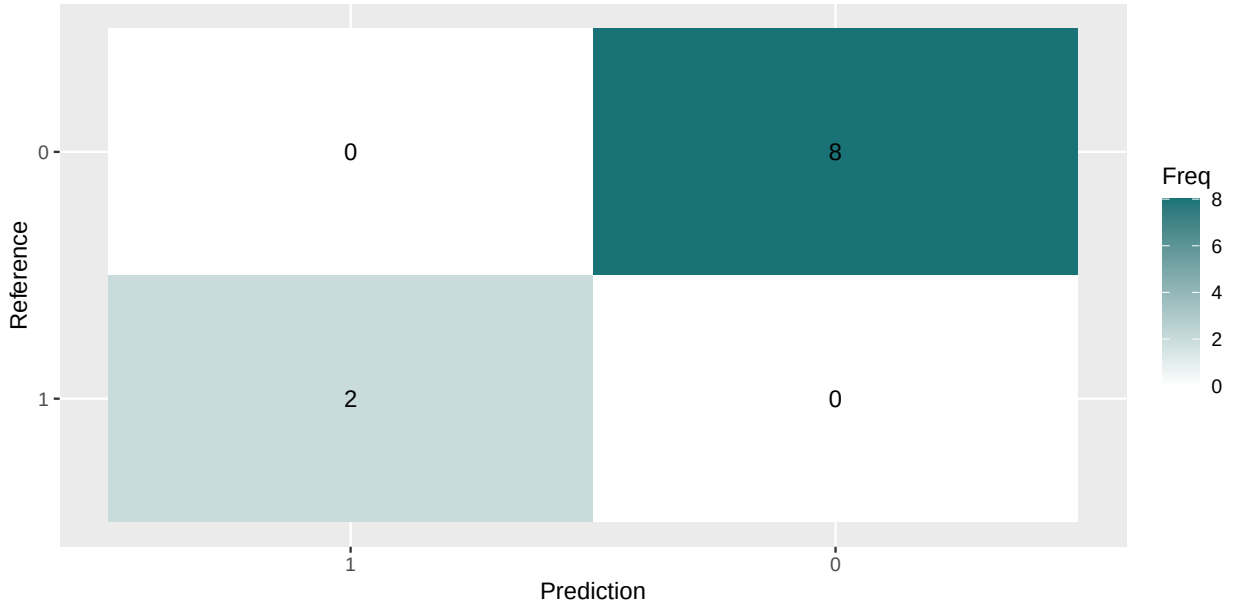
```
# Extraemos las predicciones del modelo para el conjunto de datos de testing
predicciones <- predict(modelo_SMI_bin, newdata = test_dataset, type = "response")

# Dado que queremos estar seguros de nuestra predicción,
# discretizamos dichas predicciones en función de su valor:
#   - 1 si el valor es mayor o igual a 0.80
#   - 0 si el valor es menor a 0.80
y_pred <- ifelse(predicciones >= 0.8, 1, 0)

# Calculamos la matriz de confusión
matriz_confusion <- confusionMatrix(factor(y_pred), factor(test_dataset$SMI_bin), dnn = c("Prediction",

# Preparamos el plot para enseñar la matriz de confusión
plt <- as.data.frame(matriz_confusion$table)
plt$Prediction <- factor(plt$Prediction, levels = rev(levels(plt$Prediction)))
plt$Reference <- factor(plt$Reference, levels = rev(levels(plt$Reference)))

ggplot(plt, aes(Prediction, Reference, fill = Freq)) +
       geom_tile() +
       geom_text(aes(label = Freq)) +
       scale_fill_gradient(low = "white", high = "#197376") +
       labs(x = "Prediction", y = "Reference") +
       scale_x_discrete(labels = levels(plt$Prediction)) +
       scale_y_discrete(labels = levels(plt$Reference))
```




```
# Mostramos los resultados
cat("Sensibilidad: ", round(matriz_confusion$byClass["Sensitivity"], 4))
```

```
## Sensibilidad:  1
```

```
cat("Especificidad: ", round(matriz_confusion$byClass["Specificity"], 4))
```

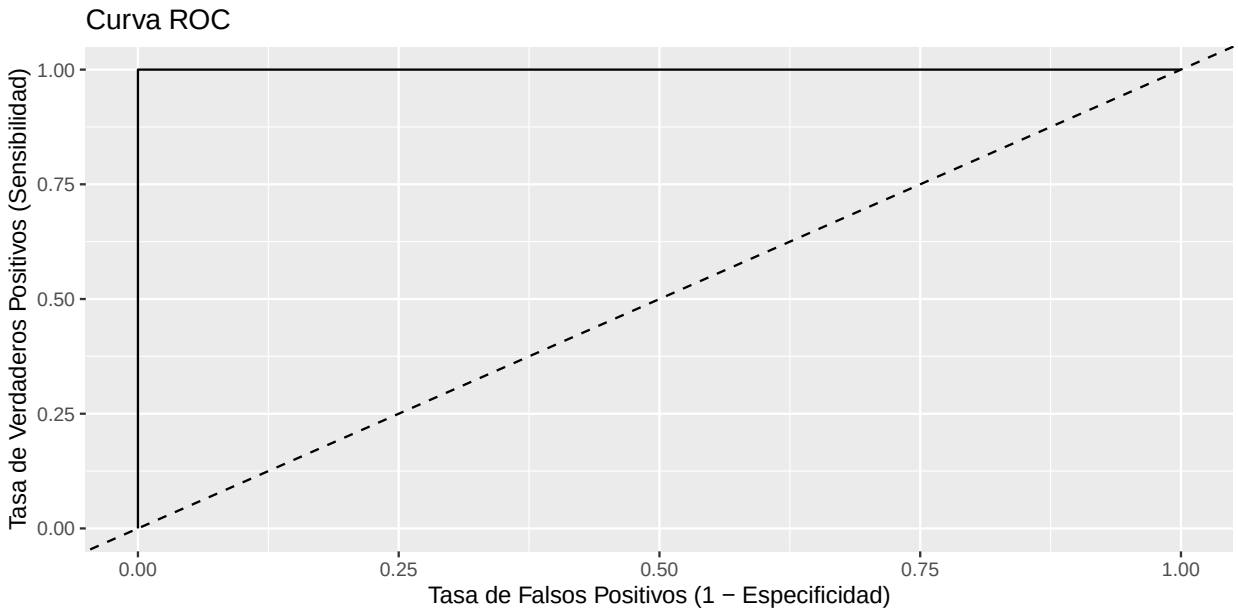
```
## Especificidad:  1
```

Observando estos resultados, el modelo funciona excelentemente bien para el desbalanceo que hay entre las etiquetas "superior al SMI de España" (predicción ≥ 0.80) y las etiquetas de "inferior al SMI de España" (predicción < 0.80). Este desajuste en la cantidad de datos por clase siempre puede causar problemas en el entrenamiento de modelos y en su posterior evaluación. Sin embargo, su especificidad y su sensibilidad son redondas, convirtiéndose en un excelente modelo de regresión logística.

Por último, enseñaremos la gráfica de la curva ROC y junto con el área debajo de la curva (AUC):

```
# Crearemos la instancia del objeto de la curva ROC y extraeremos sus valores
roc <- roc(test_dataset$SMI_bin, predicciones)
roc_valores <- coords(roc)

# Mostraremos la gráfica de la curva ROC
roc_plot <- ggplot(roc_valores, aes(x = 1 - specificity, y = sensitivity)) +
            geom_path() +
            geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
            labs(title = "Curva ROC",
                 x = "Tasa de Falsos Positivos (1 - Especificidad)",
                 y = "Tasa de Verdaderos Positivos (Sensibilidad)")
roc_plot
```




```
# Calcularemos el área debajo de la curva (AUC)
auc <- auc(roc)

# Mostraremos el valor resultante en porcentaje de área abarcada
cat("Área bajo la curva (AUC): ", round(auc*100, 2), "%")
```

```
## Área bajo la curva (AUC):  100 %
```

La curva es perfecta, con un área del 100%, corroborando lo expuesto anteriormente: es un gran modelo.

5. Resolución del problema

5.1 Primera pregunta

Para la primera pregunta nos habíamos planteado si un mes de alquiler en una zona normal es significativamente mayor en Europa que en América. Para poder responderla, estudiaremos los resultados del análisis por contraste de hipótesis. Dado que el **valor observado = 2.964443** es positivo, todo nos afirma que este valor no está en la zona de aceptación de la hipótesis nula: $(\text{-}\infty, z_{1-\infty}]$. Por lo tanto, podemos rechazar la hipótesis nula planteada de igualdad de medias de alquiler entre Europa y América

Ahora, sabiendo el **valor *p*** podemos tomar una decisión sobre las hipótesis. Observando que $P(z_{obs} \geq 2.964443) = 0.001516159$ y nuestro valor de significancia es $\alpha = 0.05$, podemos concluir que existe evidencia para rechazar la hipótesis nula ya que el valor p es inferior a 0.05. Por ende, queda demostrado que el alquiler de un mes en una zona normal en Europa es significativamente mayor que en América al 95% de confianza. Cabe comentar que los países de América incluidos en la muestra son en su mayoría países de América del Sur, lo cual indica que los resultados son coherentes.

5.2 Segunda pregunta

Como hemos comentado podemos ver que el *SMI* de un país podría determinarse a partir de los datos de gasto mínimo medio en *Housing* y del *PIB.anual* de un país. Para poder aplicar este modelo, vamos a considerar los datos de un país que hemos eliminado del análisis por falta del *SMI* y por un alto *PIB anual*: **Singapur**.

```
# Datos extraídos de los ficheros de pre-procesamiento
Housing <- 91.50
PIB.anual <- 4.43e12
Food <- 15

# creamos el sub-dataframe necesario para la predicción
df <- data.frame(Housing = Housing, PIB.anual = PIB.anual, Food = Food)
SMI_Singapur <- predict(model_SMI, newdata = df)

# Mostramos la predicción
SMI_Singapur
```

```
##        1
## 1826.059
```

Por lo tanto, podríamos considerar que el *SMI* de Singapur es cercano a los 1.826€, lo cuál coincide con los datos recogidos por la web de **Business Times**

5.3 Tercera pregunta

Finalmente, solo queda responder la última pregunta. Nos habíamos cuestionado a que países podríamos ir a trabajar que presenten un SMI mayor al de España. Como se ha querido asegurar fuertemente que las predicciones eran correctas, la discretización de las mismas en el análisis del modelo de regresión logística se llevó a cabo con un nivel del 80% de confianza mínimo. Luego, lo único que falta es extraer los países con predicciones "superiores al SMI de España":

```
# Creamos un dataframe con las predicciones y los países cuya predicción sea 1
predicciones_df <- subset(data.frame(Country = test_dataset$Country, SMI = test_dataset$SMI, y_pred = y

# Los países candidatos son:
predicciones_df
```

```
##        Country    SMI y_pred
## 7  NEW ZEALAND 2156.3      1
```

```
## 10      ISRAEL 1528.8      1
```

6. Repositorio Github y vídeo

El *link* al repositorio de Github con toda la solución es el siguiente:
https://github.com/Tipologia-y-Ciclo-de-Vida-de-los-Datos/Practica-2--Limpieza-y-Analisis-de-datos

El vídeo de la práctica se colgará en el apartado de *Video Práctica* del foro de la asignatura.

Tabla de contribuciones

CONTRIBUCIONES	FIRMA
Investigación previa	JLSD, MIGS
Redacción de las respuestas	JLSD, MIGS
Desarrollo del código	JLSD, MIGS
Participación en el vídeo	JLSD, MIGS

Bibliografía

- **Recurso web: Datosmacro**
- **Esteban Vega Lozano. Pre procesamiento de los datos**
- **Josep Gibergans Baguena. Regresión lineal simple**
- **Josep Gibergans Baguena. Regresión lineal múltiple**
- **Montserrat Guillén Estany, María Teresa Alonso Alonso. Modelos de regresión logística**
- **Laia Subirats Maté, Diego Oswaldo Pérez Trenard, Mireia Calvo González. Introducción a la limpieza y análisis de los datos**