Práctica2

Sergio García Fernández

3 de junio de 2018

El objetivo de esta actividad será el tratamiento de un dataset. 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?
- 2. Integración y selección de los datos de interés a analizar
- 3. Limpieza de los datos
  - 3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?
  - 3.2. Identificación y tratamiento de valores extremos.
- 4. Análisis de los datos
  - 4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)
  - 4.2. Comprobación de la normalidad y homogeneidad de la varianza
  - 4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos
- 5. Representación de los resultados a partir de tablas y gráficas
- 6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?
- 7. Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

Los datos a tratar corresponden a la informació del World Happiness Report del año 2016 que muestra una serie de variables asociadas a la felicidad en distintos países del mundo. Las variables del fichero son: Country, Region, Happiness.Rank, Happiness.Score, Lower.Confidence.Interval, Upper.Confidence.Interval, GDP.per.Capita, Family, Life.Expectancy, Freedom, Government.Corruption, Generosity, Dystopia.Residual. Intenta responder a la felicidad de cada país, teniendo en cuenta una serie de variables como la esperanza de vida, la renta per cápita, la corrupción … Es importante para determinar en que países existe un nivel mayor de felicidad.

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

```
datos <- read.csv(file="C:\\Users\\Sergio\\Documents\\2016_raw.csv",head=TRUE,
sep=",")
names (datos) = c("Country", "Region", "HR", "HS", "LCI", "UCI", "GPC", "Famil
y", "LE", "Freedom", "GC", "Generosity", "DR")
names(datos)
```

```
##  [1] "Country"    "Region"     "HR"         "HS"         "LCI"
##  [6] "UCI"        "GPC"        "Family"     "LE"         "Freedom"
## [11] "GC"         "Generosity" "DR"
```

Cambio el nombre de las variables para facilitar trabajar con ellas. Creo un dataframe con las variables interesantes para el estudio.

```
df <- data.frame(datos$Country,datos$Region,datos$HR,datos$HS,datos$LCI,datos
$UCI,datos$GPC,datos$LE,datos$GC)
colnames(df)<-c("Country","Region","HR","HS","LCI","UCI","GPC","LE","GC")
```

3. Limpieza de los datos.

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

```
sapply(df,class)
```

```
##   Country    Region        HR        HS       LCI       UCI       GPC
##  "factor"  "factor" "integer"  "factor" "numeric" "numeric"  "factor"
##        LE        GC
## "numeric" "numeric"
```

```
df[,4] <-  as.numeric( sub(",","\\.",df[,4]))
df[,7] <-  as.numeric( sub(",","\\.",df[,7]))
df$HS <- as.numeric(as.character(df$HS))
df$GPC <- as.numeric(as.character(df$GPC))
sapply(df,class)
```

```
##   Country    Region        HR        HS       LCI       UCI       GPC
##  "factor"  "factor" "integer" "numeric" "numeric" "numeric" "numeric"
##        LE        GC
## "numeric" "numeric"
```

```
kk <- trimws(df$Country)
kk <- toupper(kk)
table(kk)
```

```
## kk
##             AFGHANISTAN                 ALBANIA                 ALGERIA
##                       1                       1                       1
##                  ANGOLA               ARGENTINA                 ARMENIA
##                       1                       1                       1
##               AUSTRALIA                 AUSTRIA              AZERBAIJAN
##                       1                       1                       1
##                 BAHRAIN              BANGLADESH                 BELARUS
##                       1                       1                       1
##                 BELGIUM                  BELIZE                   BENIN
##                       1                       1                       1
##                  BHUTAN                 BOLIVIA  BOSNIA AND HERZEGOVINA
##                       1                       1                       1
##                BOTSWANA                  BRAZIL                BULGARIA
##                       1                       1                       1
##            BURKINA FASO                 BURUNDI                CAMBODIA
##                       1                       1                       1
##                CAMEROON                  CANADA                    CHAD
##                       1                       1                       1
##                   CHILE                   CHINA                COLOMBIA
##                       1                       1                       1
##                 COMOROS     CONGO (BRAZZAVILLE)        CONGO (KINSHASA)
##                       1                       1                       1
##              COSTA RICA                 CROATIA                  CYPRUS
##                       1                       1                       1
##          CZECH REPUBLIC                 DENMARK      DOMINICAN REPUBLIC
##                       1                       1                       1
##                 ECUADOR                   EGYPT             EL SALVADOR
##                       1                       1                       1
##                 ESTONIA                ETHIOPIA                 FINLAND
##                       1                       1                       1
##                  FRANCE                   GABON                 GEORGIA
##                       1                       1                       1
##                 GERMANY                   GHANA                  GREECE
##                       1                       1                       1
##                GUATEMALA                  GUINEA                   HAITI
##                       1                       1                       1
##                HONDURAS               HONG KONG                 HUNGARY
##                       1                       1                       1
##                 ICELAND                   INDIA               INDONESIA
##                       1                       1                       1
##                    IRAN                    IRAQ                 IRELAND
##                       1                       1                       1
##                  ISRAEL                   ITALY              IVORY COAST
##                       1                       1                       1
##                 JAMAICA                   JAPAN                  JORDAN
##                       1                       1                       1
##              KAZAKHSTAN                   KENYA                  KOSOVO
##                       1                       1                       1
##                  KUWAIT              KYRGYZSTAN                    LAOS
##                       1                       1                       1
##                  LATVIA                 LEBANON                 LIBERIA
```

```
##                     1                     1                     1
##                 LIBYA             LITHUANIA            LUXEMBOURG
##                     1                     1                     1
##             MACEDONIA            MADAGASCAR               MALAWI
##                     1                     1                     1
##              MALAYSIA                  MALI                 MALTA
##                     1                     1                     1
##            MAURITANIA            MAURITIUS                MEXICO
##                     1                     1                     1
##               MOLDOVA              MONGOLIA            MONTENEGRO
##                     1                     1                     1
##               MOROCCO               MYANMAR               NAMIBIA
##                     1                     1                     1
##                 NEPAL           NETHERLANDS           NEW ZEALAND
##                     1                     1                     1
##             NICARAGUA                 NIGER               NIGERIA
##                     1                     1                     1
##          NORTH CYPRUS                NORWAY              PAKISTAN
##                     1                     1                     1
## PALESTINIAN TERRITORIES                PANAMA              PARAGUAY
##                     1                     1                     1
##                  PERU           PHILIPPINES                POLAND
##                     1                     1                     1
##              PORTUGAL           PUERTO RICO                 QATAR
##                     1                     1                     1
##               ROMANIA                RUSSIA               RWANDA
##                     1                     1                     1
##          SAUDI ARABIA               SENEGAL                SERBIA
##                     1                     1                     1
##          SIERRA LEONE             SINGAPORE              SLOVAKIA
##                     1                     1                     1
##              SLOVENIA               SOMALIA      SOMALILAND REGION
##                     1                     1                     1
##          SOUTH AFRICA           SOUTH KOREA           SOUTH SUDAN
##                     1                     1                     1
##                 SPAIN             SRI LANKA                 SUDAN
##                     1                     1                     1
##              SURINAME                SWEDEN           SWITZERLAND
##                     1                     1                     1
##                 SYRIA                TAIWAN            TAJIKISTAN
##                     1                     1                     1
##              TANZANIA              THAILAND                  TOGO
##                     1                     1                     1
##    TRINIDAD AND TOBAGO               TUNISIA                TURKEY
##                     1                     1                     1
##          TURKMENISTAN                UGANDA               UKRAINE
##                     1                     1                     1
##   UNITED ARAB EMIRATES        UNITED KINGDOM         UNITED STATES
##                     1                     1                     1
##               URUGUAY            UZBEKISTAN             VENEZUELA
##                     1                     1                     1
##               VIETNAM                 YEMEN                ZAMBIA
```

```
##                       1                       1                       1
##                ZIMBABWE
##                       1
```

```
df$Country <-as.factor(kk)
kk <- trimws( df$Region )
kk <- gsub("AFRCA", "AFRICA",kk)
df$Region <-as.factor(kk)
```

Se cambia el nombre de los países a mayúsculas para trabajar mejor con ellos.

```
kk <- which(df$UCI- df$LCI < 0)
kk1 <- df$UCI[kk]
df$UCI[kk] <- df$LCI[kk]
df$LCI[kk] <- kk1
df$HR<-rank(-df$HS)
```

Solucionamos las inconsistencias en las variables. Y comprobamos que el ranking esté correctamente ordenado. Si existiera valores en el ranking vacios, se complentarian con el valor correcto.

```
df_aux<-data.frame(aggregate(df$HS ~ df$Region, df, FUN = function(x) mean(as.
numeric(as.character(x)))))
df$HS[1]=NA
df$HS[8]=NA
df$HS[24]=NA
vector<-which(is.na(df$HS))
for(i in vector){df$HS[i]<-df_aux$df.HS[which(df_aux$df.Region==df$Region
[i])] }
```

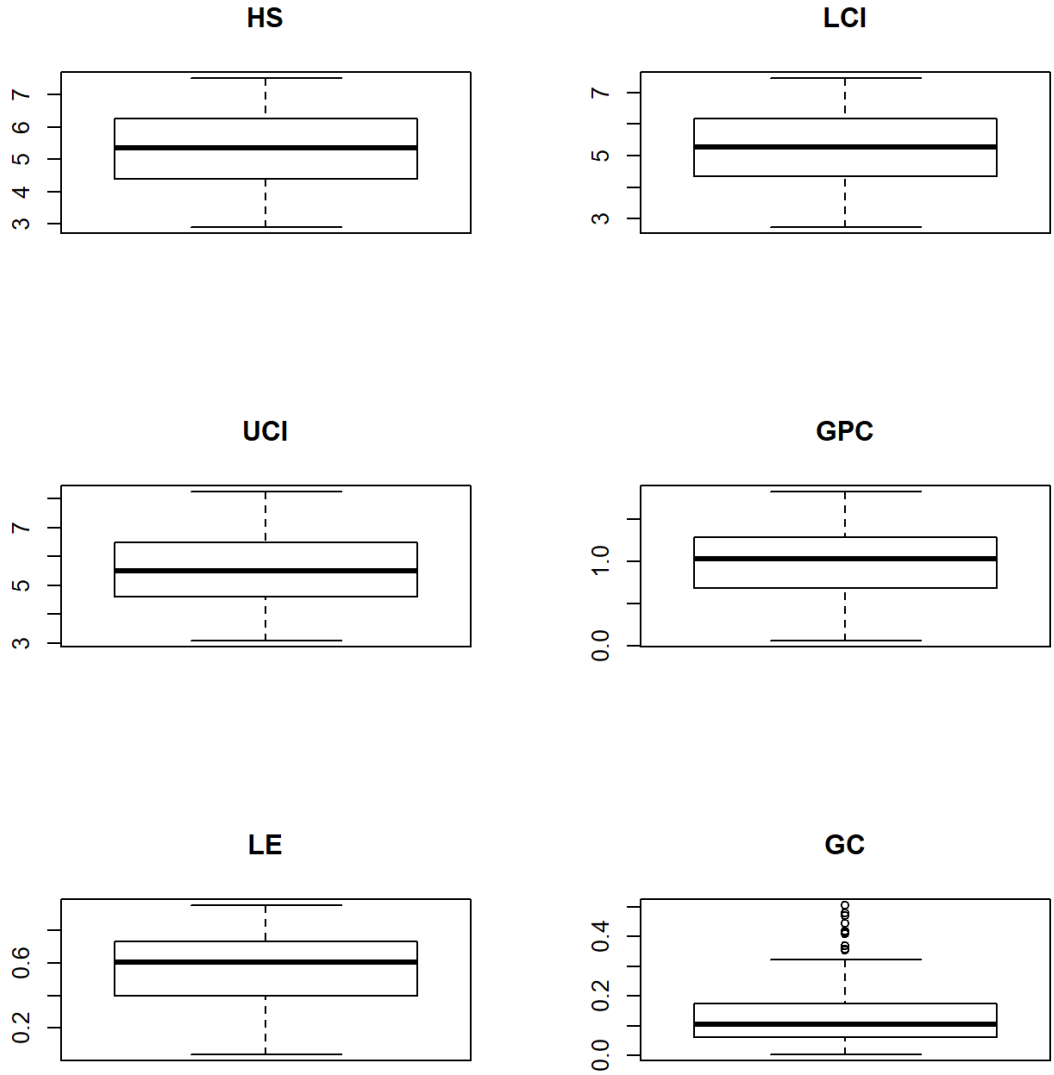
Fuerzo a que haya varios vacios en la columna de HS, y los rellenamos con la media de los valores de su región, obviando los valores vacios. En caso de que se diera en otra de las columnas numéricas, lo haría de la misma manera.

3.2. Identificación y tratamiento de valores extremos.

Eliminamos los registros que tienen algún cero en las diferentes columnas

```
df<-df[which(df$HS!="0"),]
df<-df[which(df$LCI!="0"),]
df<-df[which(df$UCI!="0"),]
df<-df[which(df$GPC!="0"),]
df<-df[which(df$LE!="0"),]
df<-df[which(df$GC!="0"),]
```

Se eliminan dos registros



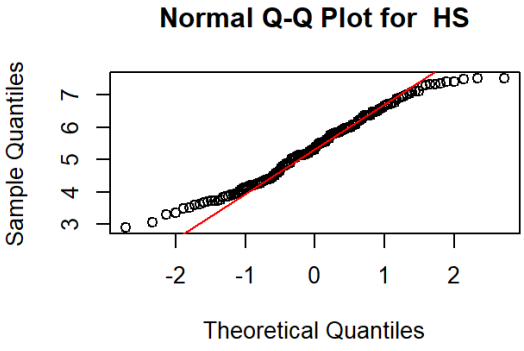
Comprobamos en los gráficos que no existen outliers.

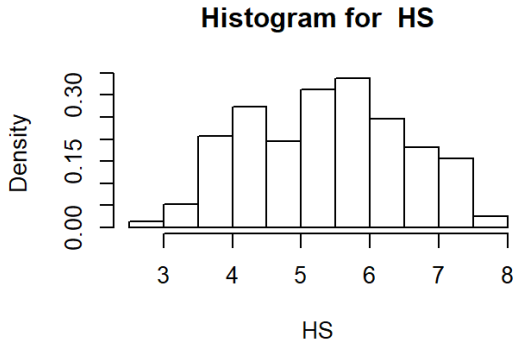
4. Análisis de los datos.

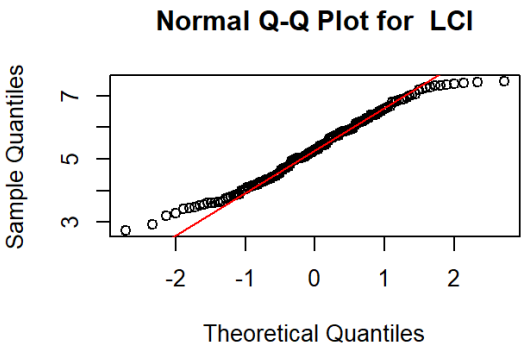
4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

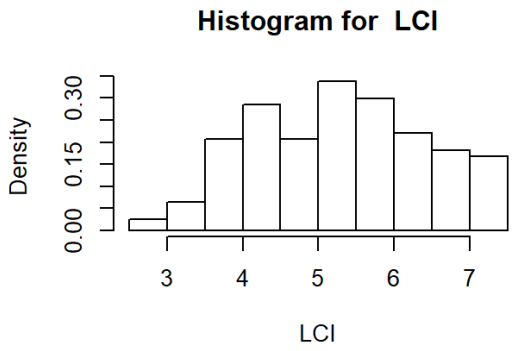
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

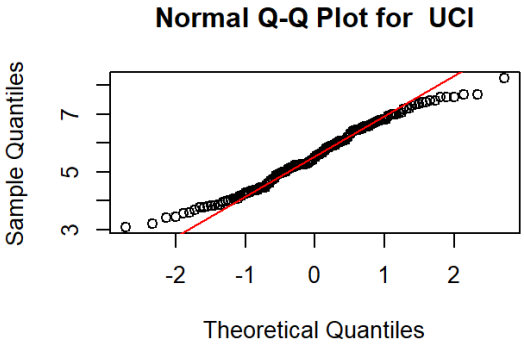
```
par(mfrow=c(2,2))
for(i in 4:ncol(df)) {
  if (is.numeric(df[,i])){
    qqnorm(df[,i],main = paste("Normal Q-Q Plot for ",colnames(df)[i]))
    qqline(df[,i],col="red")
    hist(df[,i],
      main=paste("Histogram for ", colnames(df)[i]),
      xlab=colnames(df)[i], freq = FALSE)
  }
}
```

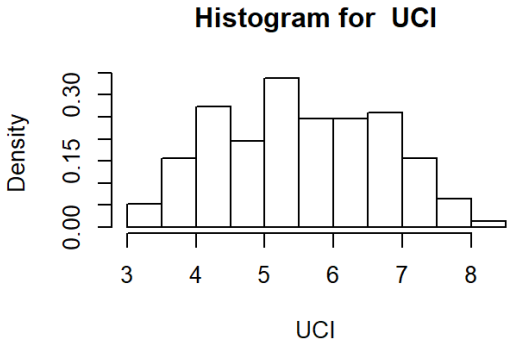


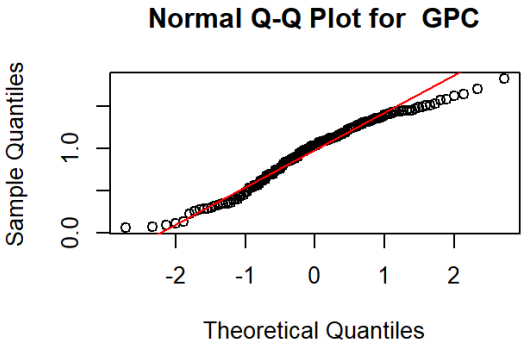


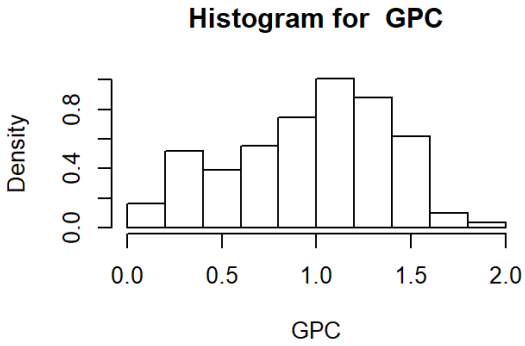


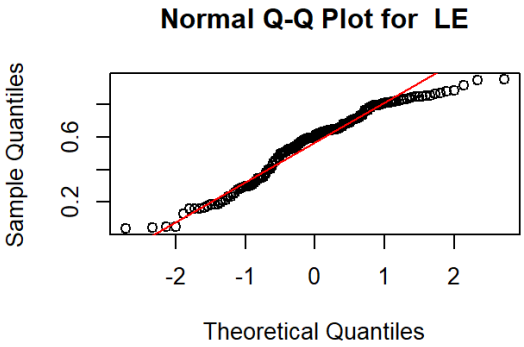


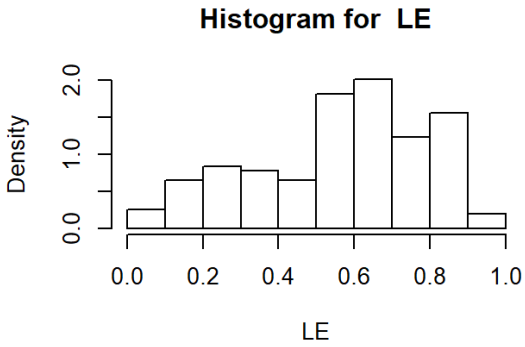


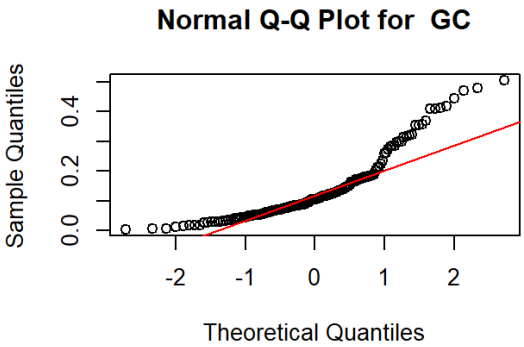


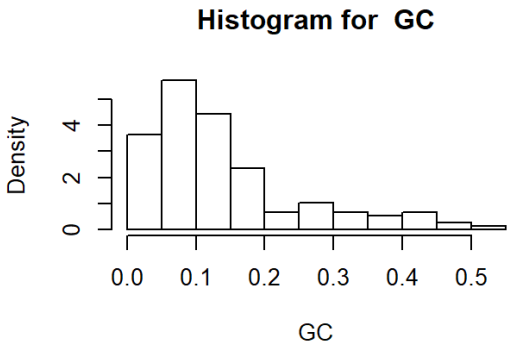













```
shapiro.test(df$HS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$HS
## W = 0.97725, p-value = 0.01187
```

```
shapiro.test(df$LCI)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$LCI
## W = 0.97812, p-value = 0.01492
```

```
shapiro.test(df$UCI)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$UCI
## W = 0.98052, p-value = 0.02811
```

```
shapiro.test(df$GPC)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$GPC
## W = 0.96798, p-value = 0.001194
```

```
shapiro.test(df$LE)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$LE
## W = 0.95382, p-value = 5.477e-05
```

```
shapiro.test(df$GC)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$GC
## W = 0.85399, p-value = 4.536e-11
```

El test nos indica que ninguna variable esta normalizada, ya que el p-valor es inferior al coeficiente 0.05, por lo que se puede rechazar la hipotesis nula y entender que no es normal.

Que no sea normal no quiere decir que no pueda ser normalizable, ya que segun el teorema del limite central al tener mas de 30 elementos en las observaciones podemos aproximarla como una distribución normal de media 0 y desviación estandard 1.

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.

Vamos a realizar la comparación de la felicidad en Europa, con las del resto del mundo.

```
eu<-c("CENTRAL AND EASTERN EUROPE","WESTERN EUROPE")
df_europe <- df[which(df$Region %in% eu),]
df_rest <- df[which(!df$Region %in% eu),]
Model<- lm(HS~GPC+LE+GC, data=df_europe)
summary(Model)
```

```
##
## Call:
## lm(formula = HS ~ GPC + LE + GC, data = df_europe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48513 -0.45034  0.07974  0.31711  1.35343
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9686     0.4961   5.984 3.30e-07 ***
## GPC           1.6642     0.5371   3.098  0.00335 **
## LE            0.7048     1.1287   0.624  0.53546
## GC            2.8945     0.6708   4.315 8.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5533 on 45 degrees of freedom
## Multiple R-squared:  0.6617, Adjusted R-squared:  0.6391
## F-statistic: 29.33 on 3 and 45 DF,  p-value: 1.151e-10
```

El coeficiente de la bondad de ajuste es 0.6261 y el coeficiente ajustado es: 0.6012

```
Modelrest<- lm(HS~GPC+LE+GC, data=df_rest)
summary(Modelrest)
```

```
##
## Call:
## lm(formula = HS ~ GPC + LE + GC, data = df_rest)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.24148 -0.44043  0.00418  0.42766  1.26549
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9641     0.1564  18.957  < 2e-16 ***
## GPC           1.2272     0.2490   4.928 3.26e-06 ***
## LE            2.2087     0.4505   4.902 3.62e-06 ***
## GC            0.2209     0.6569   0.336    0.737
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6288 on 101 degrees of freedom
## Multiple R-squared:  0.7078, Adjusted R-squared:  0.6991
## F-statistic: 81.55 on 3 and 101 DF,  p-value: < 2.2e-16
```

El coeficiente de la bondad de ajuste es 0.67 y el coeficiente ajustado es: 0.6603

Al comparar el ajuste en ambos modelos, comprobamos que el ajuste es mejor para el resto de la muestra, que no se encuentran en Europa, que las que se encuentra en Europa. Realizamos una predicción con unos datos en ambos modelos para compararlos

```
newdata=data.frame(GPC=1.5, LE=0.69, GC=0.35)
predict(Model, newdata)
```

```
##       1
## 6.96426
```

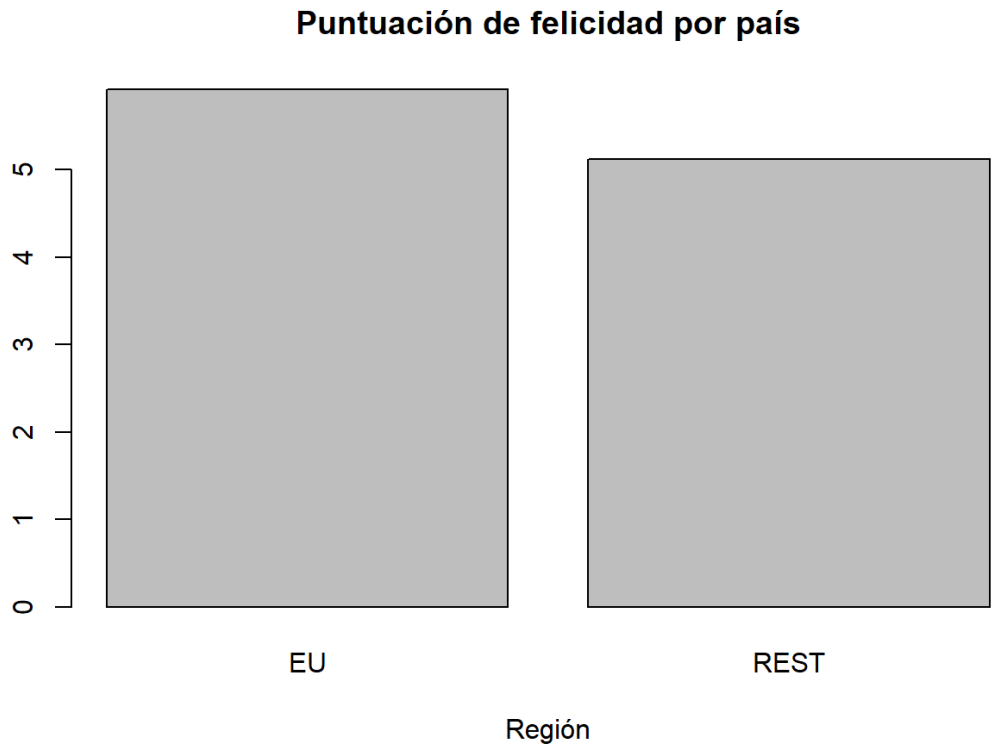
```
predict(Modelrest,newdata)
```

```
##        1
## 6.406256
```

El valor de la felicidad para los datos es de 40.21791 para el primer modelo y 74.6098 para el segundo.

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

```
eurHS=sum(as.numeric(df_europe$HS))/length(df_europe$HS)
restHS=sum(as.numeric(df_rest$HS))/length(df_rest$HS)
counts <- c(eurHS, restHS)
barplot(counts, names=c("EU", "REST"), main="Puntuación de felicidad por paí
s",
    xlab="Región")
```



Como se puede comprobar, la felicidad en Europa es superior a la del resto de los países en la muestra.

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

Nos queda que:

-La felicidad media para los países europeos estudiados es de r toString(round(eurHS, digits=0)). -La felicidad media para el resto de los países, que no se encuentran en Europa, es de r toString(round(restHS, digits=0)) dolares.
Por lo que se puede concluir por los datos estudiados, que se vive con mayor felicidad en un país europeo, que en el resto del mundo.

7. Código

Procedemos a exportar los datos sobre los que se ha trabajado

```
write.csv(df, file = "C:/Users/Sergio/Desktop/Master/Tipologia y ciclo de vida de los datos/Practica2/HS2016.csv")
```