PRA 2

Douglas Hernandez | Ernesto Miguez

2023-01-12

Carga de datos y paquetes

```
# Configurando la sesión de R

options(scipen=999)

# Carga de paquetes a utilizar
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1
──

## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.6     ✓ dplyr   1.0.8
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

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

```
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(arules)
```

```
## Loading required package: Matrix

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

```
##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following objects are masked from 'package:base':
##
##     abbreviate, write

# Estableciendo el directorio de trabajo de los datos

setwd("/Users/dahv10/Desktop/PRA 2/data/")

# Cargando los datos para la practica

# Datos Forbes

forbes <- read.csv("Forbes/forbes_2000.csv")

# Datos GDP

gdp <- read.csv("gdp/gdp_2.csv")

# Eliminando las comas del campo GDP para que sea reconocido como un valor numér
ico

gdp$GDP <- str_replace_all(gdp$GDP,",","")

# Dividiendo entre un billón para facilitar la lectura de las cifras de GDP

gdp <- gdp %>%
  mutate(GDP2 = as.numeric(gdp$GDP) / 1000000000)
```

Unión de tablas de datos

```
# Se unieron las tablas de forbes con gdp para complementar el análisis

data1 <- left_join(forbes, gdp, by = c("Country" = "Country.Name"))

# contabilizando los na luego de la unión de las tablas

colSums(is.na(data1))
```

```
##           X2022.Ranking       Organization.Name                Industry
##                       0                       0                       0
##                 Country           Year.Founded                     CEO
##                       0                       0                       0
##      Revenue..Billions.      Profits..Billions.      Assets..Billions.
##                       0                       0                       0
## Market.Value..Billions.         Total.Employees                    Code
##                       0                       0                     191
##                     GDP             GDP.Growth                    year
##                     191                     191                     191
##                    GDP2
##                     191
```

Tratamiento de NAs

Se aprecian un total de 191 NAs debido a que el campo clave que es el nombre de los países no son iguales en ambas tablas de datos.

```
# Modificando los nombres de países en la data de gdp para que coincidan con la
información de la data de
# Forbes

nuevos_nombres_gdp <- c("Korea, Rep." = "South Korea",
                        "Hong Kong SAR, China" =  "Hong Kong",
                        "Russian Federation" = "Russia",
                        "Egypt, Arab Rep." = "Egypt")

gdp$Country.Name <- str_replace_all(gdp$Country.Name, nuevos_nombres_gdp)

# Realizamos de nuevo el join para corregir el cruce de datos hecho anteriorment
e

data1 <- left_join(forbes, gdp, by = c("Country" = "Country.Name"))

# contabilizando los na luego de la unión de las tablas

colSums(is.na(data1))
```

```
##           X2022.Ranking       Organization.Name                Industry
##                       0                       0                       0
##                 Country           Year.Founded                     CEO
##                       0                       0                       0
```

```
##      Revenue..Billions.      Profits..Billions.       Assets..Billions. 
##                       0                       0                       0 
## Market.Value..Billions.         Total.Employees                    Code 
##                       0                       0                      48 
##                     GDP            GDP.Growth                    year 
##                      48                      48                      48 
##                    GDP2 
##                      48
```

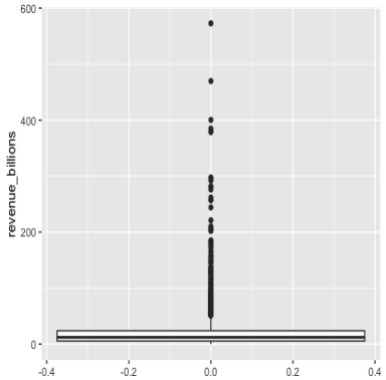
Luego de corregir los nombres de los países, aun quedan un total de 48 registros los cuales corresponden a Taiwan. En la data de gdp extraída de Kaggle, se aprecia que no existen registros para-Taiwan.

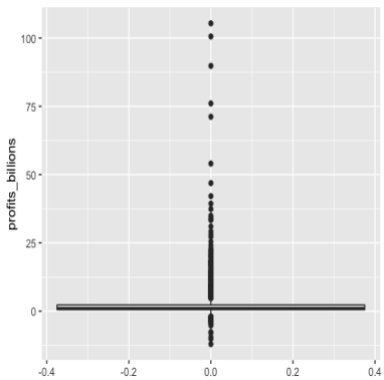
Excluiremos estos registros con el fin de tener un dataset completo

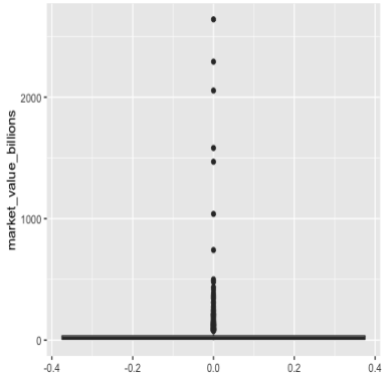
```
# Exclusión de registros con al menos un campo con na

data1 <- na.omit(data1)

# Limpiando los nombres de las variables para facilitar el análisis

data1 <- data1 %>% janitor::clean_names()
```

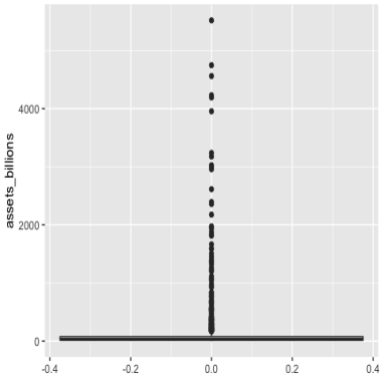
Análisis Outliers

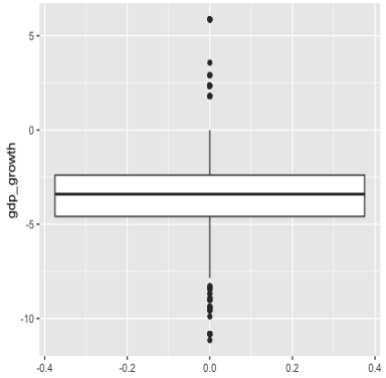
```
# Seleccionando las variables cuantitativas para construir un boxplot

data_cuanti <- data1 %>%
  select(7:10, 14, 16)

# Seleccionando las variables cuantitativas para construir un boxplot

# Graficando boxplots para cada una de las variables cuantitativas

for (column in colnames(data_cuanti)){

  p <- ggplot(data_cuanti, aes_string(x=column)) +
    geom_boxplot() +
    coord_flip()

  print(p)
}
```

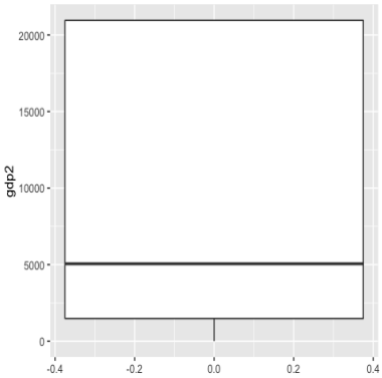










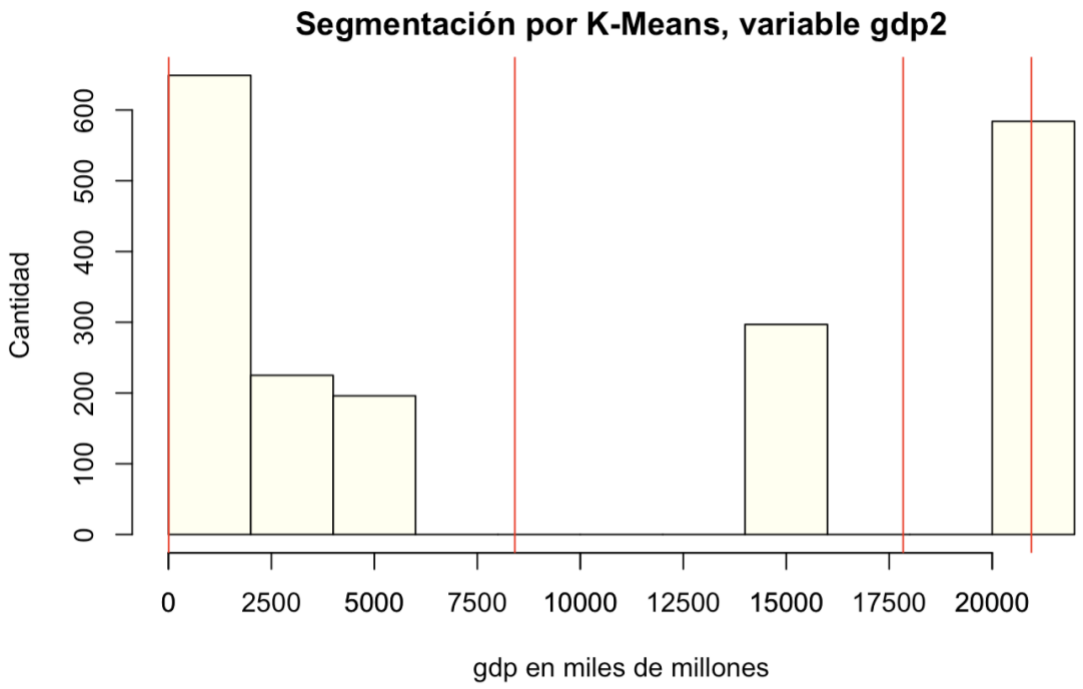


Con los boxplots podemos observar la existencia de valores extremos en la mayoría de las variables, esto se debe principalmente a que las empresas listadas en el dataset de Forbes provienen de economías de distintos tamaños.

Clasificación: K Means

Realizaremos una segmentación mediante el algoritmo de K-Means tomando como variable para segmentar el gdp.

```
set.seed(123)

clasificacion <- table(discretize(data1$gdp2, "cluster"))



# Graficando un histograma con los limites definidos por kmeans

hist(data1$gdp2, main = "Segmentación por K-Means, variable gdp2",
     xlab = "gdp en miles de millones",
     ylab = "Cantidad",
     col="ivory")

axis(side=1, at=seq(0,20000, 2500), labels=seq(0,20000, 2500))

abline(v=discretize(data1$gdp2, method = 'cluster', onlycuts = TRUE), col='red')
```



Al hacer uso del algoritmo de K-Means, se observa la existencia de 3 grupos de acuerdo con el GDP de los países analizados.

Filtrado de datos

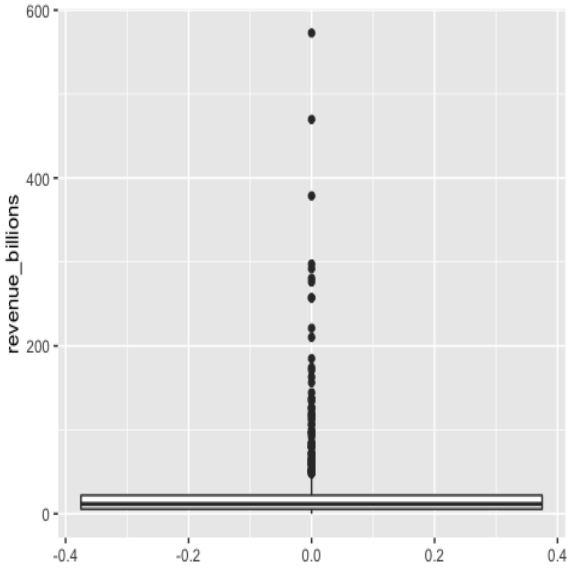
Para continuar con el desarrollo de esta practica, nos enfocaremos en las empresas para las cuales la economía del país de procedencia de estas sea superior a 17500 de acuerdo a lo observado en el gráfico.

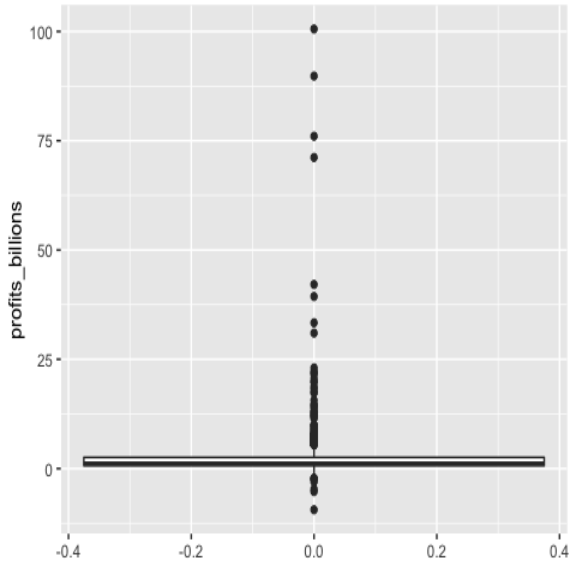
```
# Filtrando los datos: Solo nos quedamos con las compañías en las cuales su país de procedencia tenga
# un gdp mayor a 17500

forbes_a <- data1 %>%
  filter(gdp2 >= 17500)

# Países presentes en los datos filtrados
unique(forbes_a$country)

## [1] "United States"

# Cantidad de registros en la data filtrada
nrow(forbes_a)

## [1] 584
```

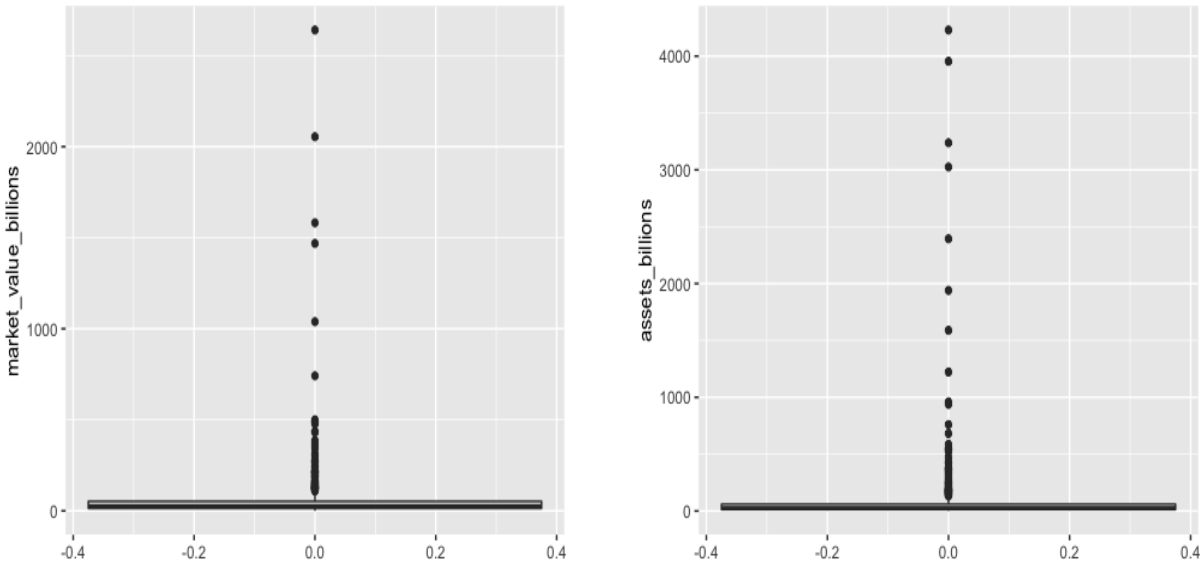
Al realizar el filtro descrito anteriormente, se comprobó que en el dataset de forbes_a solamente quedan empresas provenientes de Estados Unidos con un total de 584 compañías.

Realizamos de nuevo los boxplots para validar la dispersión en este nuevo dataset.

```
for (column in colnames(forbes_a[,c(7:10)])){

  p <- ggplot(forbes_a[,c(7:10)], aes_string(x=column)) +
    geom_boxplot() +
    coord_flip()

  print(p)
}
```







Al analizar solo las compañías provenientes de Estados Unidos, seguimos observando valores outliers. Realizaremos la inputación de la mediana con el objetivo de tratar estos valores.

```
# Análisis descriptivo para determinar las medidas de tendencia central y dispersión de los datos
```

```
summary(forbes_a)
```

```
##  x2022_ranking    organization_name    industry           country
##  Min.   :   1.0   Length:584         Length:584         Length:584
##  1st Qu.: 438.8   Class :character   Class :character   Class :character
##  Median : 926.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 940.2
##  3rd Qu.:1450.2
##  Max.   :1992.0
##   year_founded      ceo            revenue_billions profits_billions
##  Min.   :   0   Length:584         Min.   :  0.03   Min.   : -9.4200
##  1st Qu.:1913   Class :character   1st Qu.:  5.60   1st Qu.:  0.6675
##  Median :1969   Mode  :character   Median : 11.62   Median :  1.1600
##  Mean   :1907                      Mean   : 26.05   Mean   :  3.1785
##  3rd Qu.:1994                      3rd Qu.: 22.01   3rd Qu.:  2.5625
##  Max.   :2022                      Max.   :572.75   Max.   :100.5600
##  assets_billions   market_value_billions total_employees        code
##  Min.   :   1.98   Min.   :   0.41       Length:584         Length:584
##  1st Qu.:  12.92   1st Qu.:  13.82       Class :character   Class :character
##  Median :  24.70   Median :  26.48       Mode  :character   Mode  :character
##  Mean   :  94.73   Mean   :  65.40
##  3rd Qu.:  59.19   3rd Qu.:  52.01
##  Max.   :4229.90   Max.   :2640.32
```

```
##      gdp              gdp_growth          year           gdp2
##  Length:584         Min.   :-3.405   Min.   :2020   Min.   :20953
##  Class :character   1st Qu.:-3.405   1st Qu.:2020   1st Qu.:20953
##  Mode  :character   Median :-3.405   Median :2020   Median :20953
##                     Mean   :-3.405   Mean   :2020   Mean   :20953
##                     3rd Qu.:-3.405   3rd Qu.:2020   3rd Qu.:20953
##                     Max.   :-3.405   Max.   :2020   Max.   :20953
```

Tratamiento Outliers.

Vamos a crear nuevas variables que permitan categorizar los registros como “Outliers” y “No Outliers”

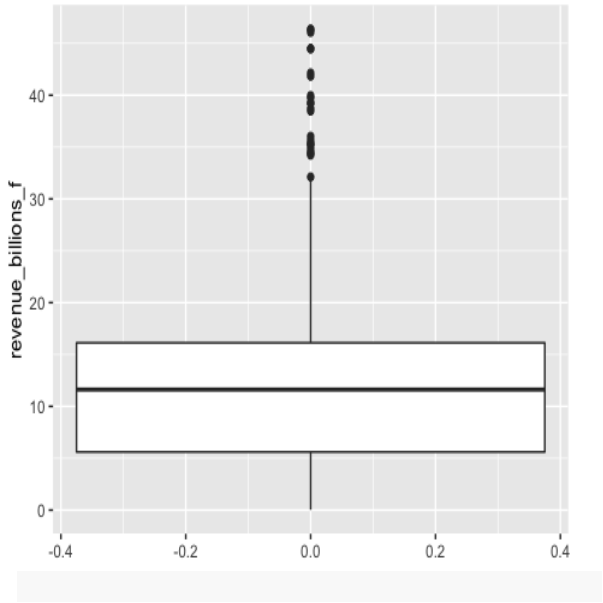
```
forbes_a <- forbes_a %>%
  mutate(revenue_billions_2 = case_when(revenue_billions < 5.60 - (1.5*(22.01-5.
60)) | revenue_billions > 22.01 + (1.5*(22.01-5.60)) ~ "Outlier",
                                        TRUE ~ "No outlier"),
         profits_billions_2 = case_when(profits_billions < 0.6675 - (1.5*(2.5625
-0.6675)) | profits_billions > 2.5625 + (1.5*(2.5625-0.6675)) ~ "Outlier",
                                       TRUE ~ "No outlier"),
         assets_billions_2 = case_when(assets_billions < 12.92 - (1.5*(59.19-12.
92)) | assets_billions > 59.19 + (1.5*(59.19-12.92)) ~ "Outlier",
                                       TRUE ~ "No outlier"),
         market_value_billions_2 = case_when(market_value_billions < 13.82 - (1.
5*(52.01-13.82)) | market_value_billions > 52.01 + (1.5*(52.01-13.82)) ~ "Outlie
r",
                                     TRUE ~ "No outlier"))
```

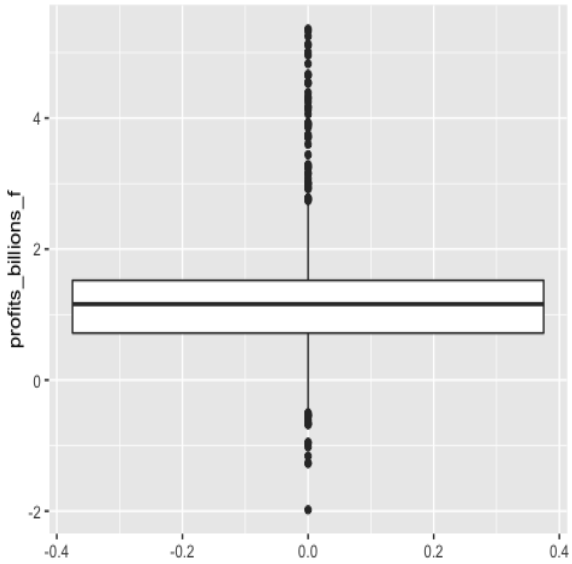
Haciendo uso de la variable creada en el paso anterior, inputaremos la mediana para los outliers.

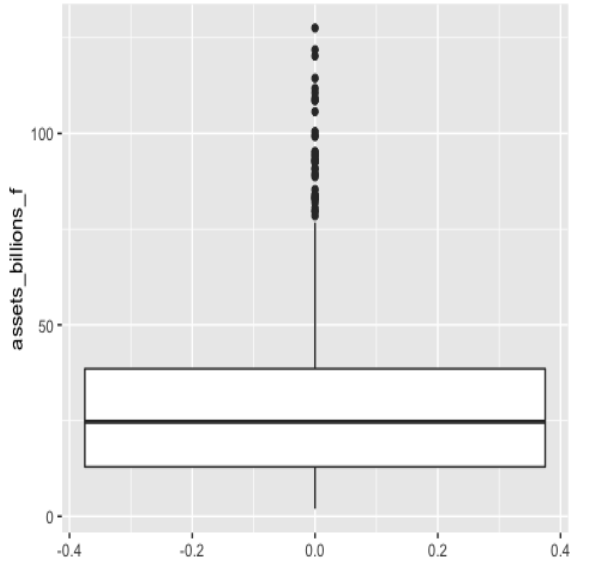
```
forbes_a <- forbes_a %>%
  mutate(revenue_billions_f = ifelse(revenue_billions_2 == "No outlier", revenue
_billions, median(revenue_billions)),
         profits_billions_f = ifelse(profits_billions_2 == "No outlier", profits
_billions, median(profits_billions)),
         assets_billions_f = ifelse(assets_billions_2 == "No outlier", assets_bi
llions, median(assets_billions)),
         market_value_billions_f = ifelse(market_value_billions_2 == "No outlier
", market_value_billions, median(market_value_billions)))
```

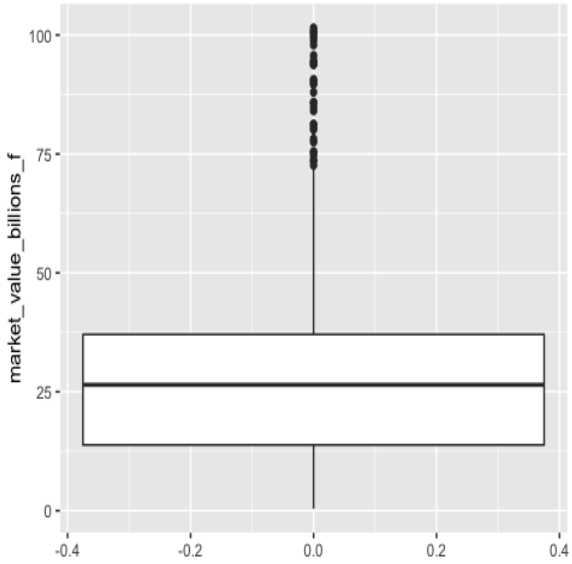
Al graficar de nuevo los boxplots luego de inputar la mediana a los outliers, se aprecia como mejora la dispersión de las variables.

```
for (column in colnames(forbes_a[,c(21:24)])){

  p <- ggplot(forbes_a[,c(21:24)], aes_string(x=column)) +
    geom_boxplot() +
    coord_flip()

  print(p)
}
```









Selección de datos.

Luego de realizar estos ajustes, seleccionaremos un subconjunto de variables para facilitar el análisis y realizamos de nuevo un análisis descriptivo para entender mejor las variables escogidas.

```
forbes_a_def <- forbes_a %>%
  select(1:6, 21:24)
```

Análisis Descriptivo

```
skimr::skim(forbes_a_def)
```

Data summary

Name	forbes_a_def
Number of rows	584
Number of columns	10
_______________________	
Column type frequency:	
character	4
numeric	6
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
organization_name	0	1	2	54	0	584	0
industry	0	1	5	32	0	28	0
country	0	1	13	13	0	1	0
ceo	0	1	7	46	0	582	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
x2022_ranking	0	1	940.21	580.44	1.00	438.75	926.00	1450.25	1992.00	▇▇▇▇▇
year_founded	0	1	1907.00	303.25	0.00	1912.75	1969.00	1994.00	2022.00	▁▁▁▁▇
revenue_billions_f	0	1	12.22	8.88	0.03	5.60	11.62	16.13	46.38	▇▃▁▁
profits_billions_f	0	1	1.26	1.08	-1.98	0.72	1.16	1.52	5.36	▁▇▁▁
assets_billions_f	0	1	29.76	24.49	1.98	12.92	24.70	38.55	127.53	▇▂▁▁▁
market_value_billions_f	0	1	28.95	21.17	0.41	13.82	26.46	37.07	101.48	▇▃▁▁

Se puede apreciar que hay un total de cuatro variables cualitativas y 6 variables cuantitativas. Para las variables de interes se observa que la mediana de revenue_billions_f es de 11.6 billones de dólares, la mediana de profits_billions_f es de 1.16 billones de dólares, la mediana de assets_billions_f es de 24.7 billones de dolares y la mediana de market_value_billions_f es de 26.5 billones de dólares.

Test Normalidad

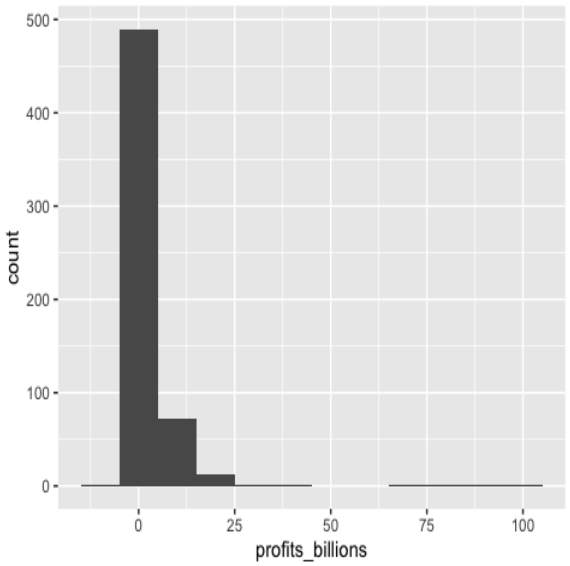
Continuaremos analizando la normalidad de las variables cuantitativas seleccionadas.

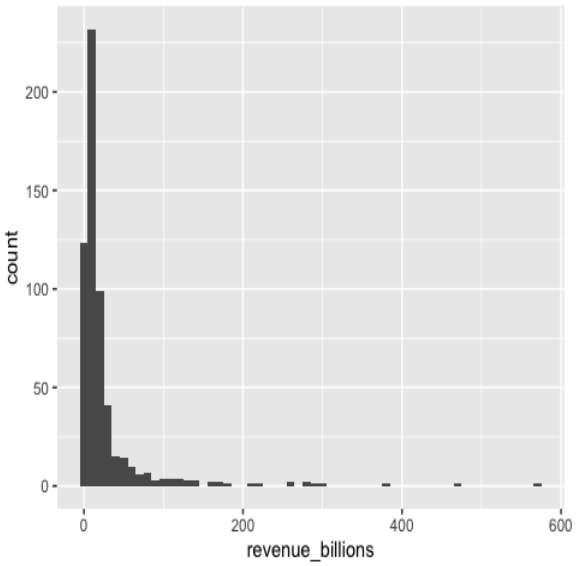
```
shapiro.test(forbes_a_def$revenue_billions_f)

##
##  Shapiro-Wilk normality test
##
## data:  forbes_a_def$revenue_billions_f
## W = 0.89448, p-value < 0.00000000000000022

shapiro.test(forbes_a_def$profits_billions_f)

##
##  Shapiro-Wilk normality test
##
## data:  forbes_a_def$profits_billions_f
## W = 0.88857, p-value < 0.00000000000000022

shapiro.test(forbes_a_def$assets_billions_f)

##
##  Shapiro-Wilk normality test
##
## data:  forbes_a_def$assets_billions_f
## W = 0.835, p-value < 0.00000000000000022

shapiro.test(forbes_a_def$market_value_billions_f)

##
##  Shapiro-Wilk normality test
##
## data:  forbes_a_def$market_value_billions_f
## W = 0.88288, p-value < 0.00000000000000022
```

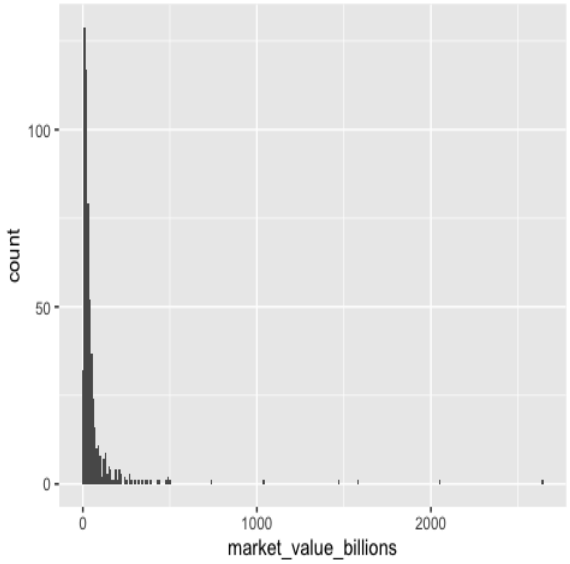
En los resultados del test de Shapiro Wilk se observan que los P-Valores de las cuatro pruebas son menores a 0.05 por lo cual podemos concluir que las distribuciones de estas variables difieren significativamente de una distribución normal.

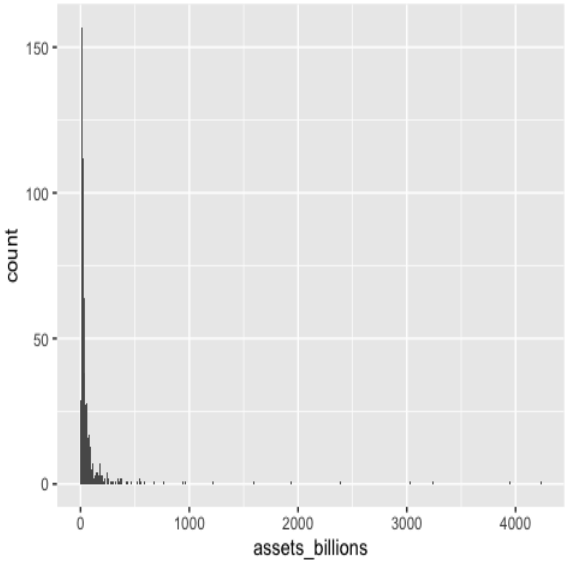
Analizamos los histogramas de cada variable para ver el sego de las mismas, en donde se observa que están sesgadas de forma positiva.

```r
for (column in colnames(forbes_a[,c(7:10)])){

  p <- ggplot(forbes_a[,c(7:10)], aes_string(x=column)) +
    geom_histogram(binwidth = 10, bins = 30)

  print(p)
}
```









Test de Homocedasticidad.

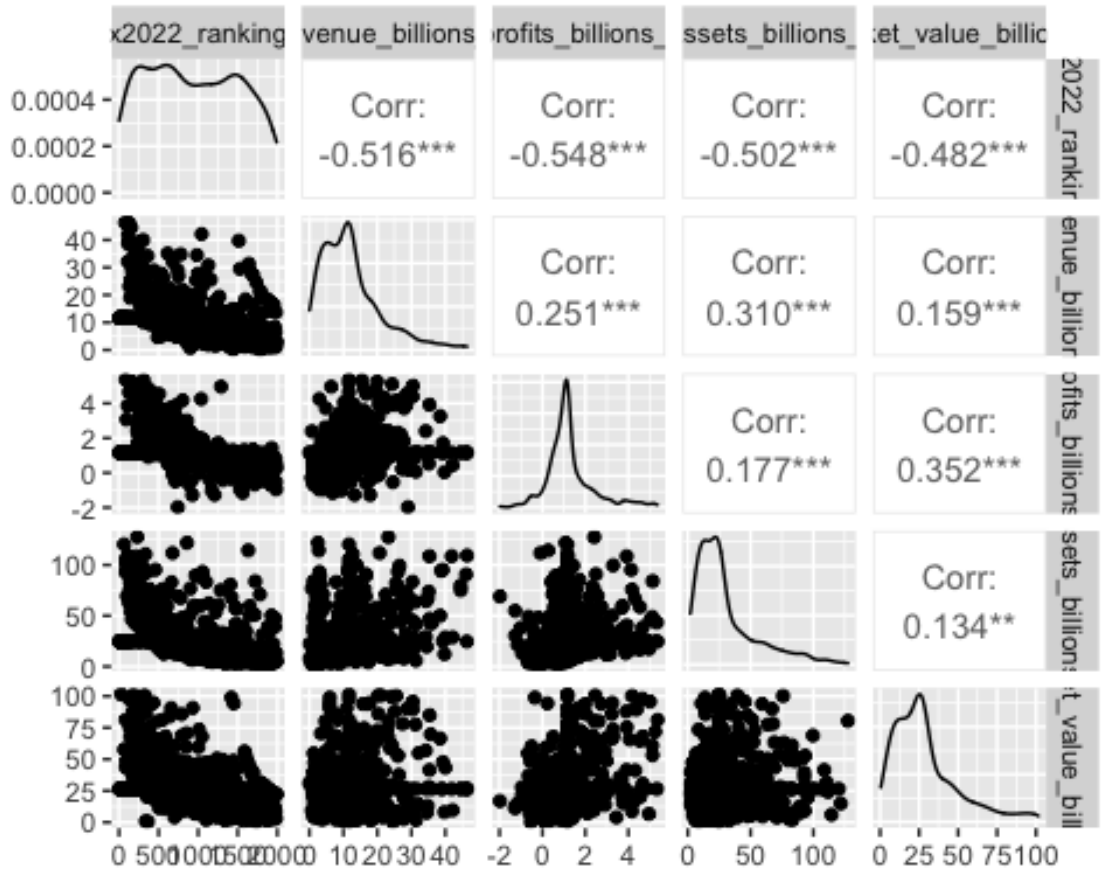
Realizamos el análisis de la homocedasticidad de la varianza para las variables cuantitativas de nuestro dataset, en donde concluimos que estas son homocedásticas.

```
car::leveneTest(forbes_a_def$revenue_billions_f ~ forbes_a_def$industry)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group  27  1.1266 0.3019
##       556

car::leveneTest(forbes_a_def$profits_billions_f ~ forbes_a_def$industry)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group  27  1.3786 0.09828 .
##       556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

car::leveneTest(forbes_a_def$assets_billions_f ~ forbes_a_def$industry)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group  27  1.4925 0.05382 .
##       556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

car::leveneTest(forbes_a_def$market_value_billions_f ~ forbes_a_def$industry)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group  27  0.7704 0.7921
##       556
```

Análisis de Correlación.

Definiremos como variable objetivo a revenue_billions_f para estudiar la correlación de esta con el resto de variables cuantitativas

```
GGally::ggpairs(forbes_a_def[, c(1, 7:10)])

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```



En la visualización de la matriz de correlación se observa que existe una correlación inversa media entre revenue_billions_f y el ranking y una correlación directa baja entre revenue_billions_f y el resto de variables.

Análisis de Regresión.

Crearemos un modelo de regresión lineal que permita predecir la variable revenue_billions_f haciendo uso de las variables x2022_ranking, assets_billions_f y profits_billions_f.

```
modelo_1 <- lm(revenue_billions_f ~ x2022_ranking + assets_billions_f + profits_
billions_f, data=forbes_a_def)
```

```
summary(modelo_1)
```

```
##
## Call:
## lm(formula = revenue_billions_f ~ x2022_ranking + assets_billions_f +
##     profits_billions_f, data = forbes_a_def)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.557  -5.222  -2.117   3.075  31.970
##
## Coefficients:
##                      Estimate Std. Error t value            Pr(>|t|)
## (Intercept)        19.1419772  1.2954722  14.776 <0.0000000000000002 ***
## x2022_ranking      -0.0077014  0.0007444 -10.346 <0.0000000000000002 ***
## assets_billions_f   0.0231049  0.0150009   1.540               0.124
## profits_billions_f -0.2918228  0.3506839  -0.832               0.406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.602 on 580 degrees of freedom
## Multiple R-squared:  0.2706, Adjusted R-squared:  0.2668
## F-statistic: 71.72 on 3 and 580 DF,  p-value: < 0.00000000000000022
```

En el summary del modelo se observa que el ajuste de R cuadrado ajustado no es bueno ya que este solo explica un 26.68% de la variabilidad de revenue_billions_f y que solamente el ranking es significativa para el modelo.

Construiremos un modelo nuevo solamente con la variable ranking para validar los resultados.

```
modelo_2 <- lm(revenue_billions_f ~ x2022_ranking,
               data=forbes_a_def)

summary(modelo_2)
```

```
##
## Call:
## lm(formula = revenue_billions_f ~ x2022_ranking, data = forbes_a_def)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.808  -5.276  -2.129   2.899  32.030
##
## Coefficients:
##                 Estimate Std. Error t value            Pr(>|t|)
## (Intercept)   19.6404329  0.5999791   32.73 <0.0000000000000002 ***
## x2022_ranking -0.0078921  0.0005431  -14.53 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.612 on 582 degrees of freedom
## Multiple R-squared:  0.2662, Adjusted R-squared:  0.265
## F-statistic: 211.1 on 1 and 582 DF,  p-value: < 0.00000000000000022
```

Se aprecia en el nuevo modelo que la variable ranking sigue siendo significativa pero aun el R2 ajustado sigue siendo muy bajo para considerar este modelo como predictivo.

Análisis de la Varianza: ANOVA

Como parte final de la practica realizaremos un análisis de la varianza para ver si el revenue_billions es distinto por industria.

```
aov_test <- aov(revenue_billions_f ~ industry, data = forbes_a_def)

summary(aov_test)
```

```
##              Df Sum Sq Mean Sq F value         Pr(>F)
## industry     27   7397  273.97   3.951 0.000000000285 ***
## Residuals   556  38558   69.35
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En vista de que el P-Valor es menor a 0.05, podemos concluir que hay diferencias significativas de revenue_billions_f entre las industrias presentes en los datos.

Contribuciones	Firma
Investigación previa	DH , EM
Redacción de las respuestas	DH , EM
Desarrollo del código	DH , EM
Participación en el vídeo	DH , EM