Netflix ETL

ETL (Extraer, Transformar, Cargar)

El siguiente documento tiene como objetivo explicar el proceso de ETL realizado en base a mis conocimientos actuales, el resultado obtenido puede no ser muy bueno en su totalidad, en caso de encontrar fallas en el analisis la documentacion queda adjuntada con el codigo realizado en RStudio para su correccion de parte de cualquier persona que posea los conocimientos adecuados.

Primero cargamos las bibliotecas y el set de datos.

```
library(tidyquant)
library(readr)
library(plotly)
library(webshot)
library(orca)
library(corrplot)
library(dplyr)

datanet <- read_csv("netflix1.csv")
```

El set de datos tiene origen del servicio de Streaming "Netflix" donde se almacenan de forma historica las series y peliculas que se fueron añadiendo al catalogo, los datos con los que contamos son: Id del programa, tipo de programa (Serie de TV o Pelicula), director, pais de origen, fecha de adhesion a la plataforma, fecha de filmacion original del programa, clasificacion, duracion del programa y genero.

```
head(datanet)
```

```
## # A tibble: 6 x 10
##   show_id type    title director country date_added release_year rating duration
##   <chr>   <chr>   <chr> <chr>    <chr>   <chr>             <dbl> <chr>  <chr>
## 1 s1      Movie   Dick~ Kirsten~ United~ 9/25/2021          2020 PG-13  90 min
## 2 s3      TV Show Gang~ Julien ~ France  9/24/2021          2021 TV-MA  1 Season
## 3 s6      TV Show Midn~ Mike Fl~ United~ 9/24/2021          2021 TV-MA  1 Season
## 4 s14     Movie   Conf~ Bruno G~ Brazil  9/22/2021          2021 TV-PG  91 min
## 5 s8      Movie   Sank~ Haile G~ United~ 9/24/2021          1993 TV-MA  125 min
## 6 s9      TV Show The ~ Andy De~ United~ 9/24/2021          2021 TV-14  9 Seaso~
## # i 1 more variable: listed_in <chr>
```

Una vez aclarado el contexto del entorno de trabajo continuemos con el analisis.

Normalizamos las informacion estandarizando el nombre de las variables de ingles a español y convirtiendo el tipo de dato.

```r
datanet_names<-c("Id_Programa","Tipo","Titulo","Director","Pais"
              ,"Fecha_Adhesion","Año_de_lanzamiento","Clasificacion"
              ,"Duracion","Genero")

names(datanet) <- datanet_names

datanet$Id_Programa <- as.character(datanet$Id_Programa)
datanet$Tipo <- as.character(datanet$Tipo)
datanet$Titulo <- as.character(datanet$Titulo)
datanet$Director <- as.character(datanet$Director)
datanet$Pais <- as.character(datanet$Pais)
datanet$Fecha_Adhesion <- as.Date(datanet$Fecha_Adhesion, format ="%m/%d/%Y")
datanet$Año_de_lanzamiento <- as.numeric(datanet$Año_de_lanzamiento)
datanet$Clasificacion <- as.character(datanet$Clasificacion)
datanet$Duracion <- as.character(datanet$Duracion)
datanet$Genero <- as.character(datanet$Genero)
```

Aqui hay algo interesante, con el metodo sapply verificamos si existen valores nulos en nuestro set de datos y en primera instancia parece que no hay evidencia de ello.

```r
datanetNA <- datanet

sapply(datanetNA, function(x) sum(is.na(x)))
```

```
##        Id_Programa               Tipo             Titulo           Director
##                  0                  0                  0                  0
##               Pais     Fecha_Adhesion Año_de_lanzamiento      Clasificacion
##                  0                  0                  0                  0
##           Duracion             Genero
##                  0                  0
```

Pero si indagamos un poco mas podemos observar que en las columnas Director y Pais existen observaciones con el valor "Not Given" o no dado esto se puede considerar como un valor NULO.

```r
print(datanetNA[c(34,53),c(4,5)])
```

```
## # A tibble: 2 x 2
##   Director   Pais
##   <chr>      <chr>
## 1 Not Given  Pakistan
## 2 Adam Salky Not Given
```

```r
print(any(datanet$Director == 'Not Given' | datanet$Pais == "Not Given")) #TRUE
```

```
## [1] TRUE
```

Identificados dichos valores los transformarmos a tipo 'NA' para luego ser eliminados.

```
datanetNA$Director[datanetNA$Director == 'Not Given'] <- NA
datanetNA$Pais[datanetNA$Pais == 'Not Given'] <- NA

print(datanetNA[c(34,53),c(4,5)])
```

```
## # A tibble: 2 x 2
##   Director   Pais
##   <chr>      <chr>
## 1 <NA>       Pakistan
## 2 Adam Salky <NA>
```

```
sapply(datanetNA, function(x) sum(is.na(x)))
```

```
##        Id_Programa               Tipo             Titulo           Director
##                  0                  0                  0               2588
##               Pais     Fecha_Adhesion Año_de_lanzamiento      Clasificacion
##                287                  0                  0                  0
##           Duracion             Genero
##                  0                  0
```

Importante: si sacamos el total de NA's de cada variable dividido el numero de observaciones podemos ver el porcentaje total que representa esos NA's por variable siendo un 29% para Director y un 3% para Pais.

```
sum(is.na(datanetNA$Director))/nrow(datanetNA) #0.29
```

```
## [1] 0.2944255
```

```
sum(is.na(datanetNA$Pais))/nrow(datanetNA) #0.03
```

```
## [1] 0.03265074
```

Si hacemos cuentas vemos que se pierde un 32.70% de la informacion total esto se obtiene haciendo la diferencia entre el total de los datos originales y los datos eliminados dividido el total de observaciones multiplicado por 100.

```
datanet_clean1 <- na.omit(datanetNA) # 32.70%
```

Otra alternativa si se quiere resguardar la informacion podria ser remover las variables mencionadas.

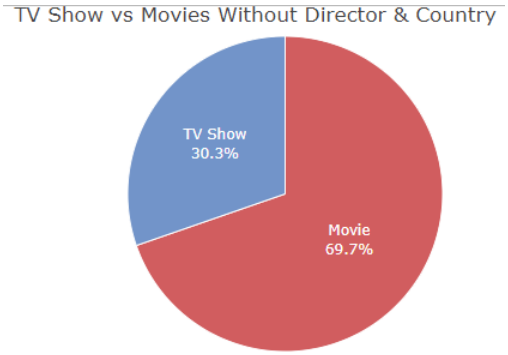
```r
datanet_clean2 <- datanetNA[,-c(4,5)] # 0.0%
```

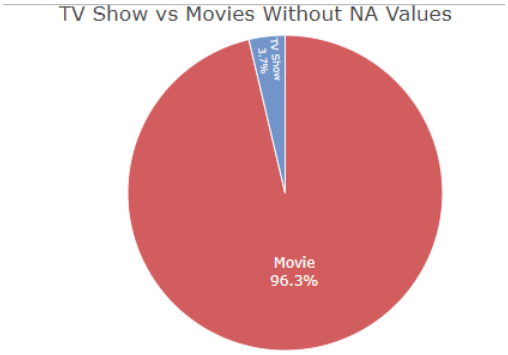
Para concluir en proceso de Extraccion, Transformacion y Cargado es una tarea compleja para garantizar la calidad de los datos a la hora de realizar un analisis y toma de decisiones, en el camino podremos encontrarnos con diversas tareas como la estandarizacion de formatos, la eliminacion de valores nulos, deteccion de errores, etc. El exito de este proceso dependera mucho de las herramientas y las tecnicas que se vayan a utilizar para convertir los datos crudos en informacion valiosa para la toma decisiones de la empresa.

Como un Extra adjunto un grafico de tarta realizado con plotly junto a su codigo comentado:

```r
#fig1 <- plot_ly(data = datanet_clean1, labels = ~type, values = ~length
#(datanet_clean1$type), type = 'pie',
#               textposition = 'inside',
#               textinfo = 'label+percent',
#               insidetextfont = list(color = '#FFFFFF'),
#               hoverinfo = 'text',
#               marker = list(colors = colors,
#                             line = list(color = '#FFFFFF', width = 1)),
              #The 'pull' attribute can also be used to create space between the sectors
#               showlegend = FALSE)

#fig1 <- fig1 %>% layout(title = 'TV Show vs Movies Without NA Values',
#                       xaxis = list(showgrid = FALSE, zeroline = FALSE,

#showticklabels = FALSE),

#                       yaxis = list(showgrid = FALSE, zeroline = FALSE,
# showticklabels = FALSE))

#fig1



# fig2 <- plot_ly(data = datanet_clean2, labels = ~type, values = ~length
# (datanet_clean2$type), type = 'pie',
#               textposition = 'inside',
#               textinfo = 'label+percent',
#               insidetextfont = list(color = '#FFFFFF'),
#               hoverinfo = 'text',
#               marker = list(colors = colors,
#                             line = list(color = '#FFFFFF', width = 1)),
              #The 'pull' attribute can also be used to create space between the sectors
#               showlegend = FALSE)

# fig2 <- fig2 %>% layout(title = 'TV Show vs Movies Without Director & Country',
#                       xaxis = list(showgrid = FALSE, zeroline = FALSE,

# showticklabels = FALSE),
#                       yaxis = list(showgrid = FALSE, zeroline = FALSE,

# showticklabels = FALSE))
```

```
# fig2
```


TV Show vs Movies Without Director & Country
TV Show
30.3%
Movie
69.7%


TV Show vs Movies Without NA Values
TV Show
3.7%
Movie
96.3%