

Practica 2: Limpieza y análisis de datos

Daniel Bagan y Rafael García

Contents

1 Descripción del dataset	1
2 Integración y selección de los datos de interés a analizar	2
3 Limpieza de los datos.	3
4 Análisis de los datos.	11
4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)	11
4.2 Comprobación de la normalidad y homogeneidad de la varianza.	11
4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos	19
5 Representación de los resultados a partir de tablas y gráficas.	38
6 Resolución del problema	41
Generación de CSV resultante	43
Firma	43

```
library(knitr)
library(stringr)
library(dplyr)
library(ggplot2)
library(ggpubr)
library(corrplot)
require(gridExtra)
library(caret)
library(psych)
library(nortest)
```

1 Descripción del dataset

Para el desarrollo de esta práctica, se ha seleccionado el dataset **Listado de películas** extraído en la primera práctica de la asignatura, desde la página web Rotten Tomatoes (<https://www.rottentomatoes.com>). Este dataset ha sido publicado en Zenodo con el DOI: 10.5281/zenodo.4265051 y puede ser visualizado en el siguiente link: <https://zenodo.org/record/4265051#.X6mORGgReUk>.

Este juego de datos se ha elegido por varios motivos, además de nuestro interés por el cine. En primer lugar, porque ha sido creado por nosotros mismos como resultado de la primera práctica de la asignatura.

En segundo lugar, porque este juego de datos es especialmente apropiado para llevar a cabo el proceso de limpieza; por una parte, un porcentaje elevado de las variables debe ser limpiado para facilitar su tratamiento posterior, y por otra parte, hay una buena cantidad de variables derivadas que se pueden crear a partir de las ya existentes. Por último, porque es adecuado para plantear diferentes tipos de problemas, como regresión (predecir la puntuación de una película), clasificación (clasificar las películas en buenas y malas), o establecer relaciones entre las distintas variables o contrastes de hipótesis, para contestar dudas como: ¿las películas de Netflix son significativamente mejores que las de HBO?, ¿las películas de drama son suelen ser más largas que las de comedia? o ¿en invierno se estrenan más películas que en verano?

El juego de datos contiene información sobre todas las películas disponibles en la página web de críticas Rotten Tomatoes. Las variables que contiene son las siguientes:

- **X:** número de película.
- **Title:** título de la película.
- **Tomatometer:** puntuación (sobre 100%) que otorga la propia página a la película, basada en la opinión de cientos de críticos. Concretamente, se trata del porcentaje de críticos que han puntuado positivamente la película.
- **Audience score:** porcentaje de usuarios de la web que han valorado la película positivamente.
- **Rating:** clasificación por edades de la película y motivo de la clasificación. Por ejemplo, $R(\text{SexualContent}|\text{SomeDrugMaterial})$ indicaría una clasificación de “Restringido” (los menores de 17 años acompañados de un adulto) por contenido sexual y drogas.
- **Genre:** género o géneros de la película.
- **Director:** director de la película.
- **Producer:** productor de la película.
- **Writer:** escritor del guión de la película.
- **Release Date (Theaters):** fecha de lanzamiento en cines.
- **Release Date (Streaming):** fecha de lanzamiento en streaming.
- **Runtime:** duración de la película.
- **Production Co:** compañía de producción.

2 Integración y selección de los datos de interés a analizar

En cuanto a las selección de datos de interés a analizar, cabe mencionar que, en el momento de recogida de los datos, ya se realizó una selección de las variables que podrían ser útiles, descartando aquellas sin valor aparente, como **Aspect Ratio**, o con una cantidad de datos vacíos demasiado elevada, como **Box Office**.

Los campos de interés a analizar en los que no se realizará ninguna modificación, salvo una limpieza simple, son: **Title**, **Tomatometer**, **Audience score**, **Rating**, **Director** y **Runtime**.

Los siguientes campos se crearán a partir de otros o se dividirán en varios:

- **Fresh.** La página web Rotten Tomatoes otorga la clasificación *Fresh* a las películas con más del 60% de crítica positiva, mientras que las demás son clasificadas como *rotten* (podridas).
- **Parental Control:** se trata de un atributo categórico dicotómico, creado a partir de la información del campo Rating. Indica si una película tiene alguna restricción de edad o no.
- **Genre:** se dividirá la variable en 3 diferentes: **género primario, secundario y terciario**.
- **Production:** contiene la compañía de producción de la película, o el nombre del productor en caso de que esta no figure en los datos.

- `Release.isWide`: atributo categórico dicotómico. Indica si el estreno de una película ha sido *wide* (estrenada en la mayoría de cines simultáneamente) o *limited* (sólo en unos pocos cines). Se obtiene a partir de `Release.Date.Theaters`.
- `Release.Date`: fecha de lanzamiento de la película, ya sea en cines o streaming.
- `Release.Season`: estación de lanzamiento de la película.
- `DirectorIsWriter`: atributo categórico dicotómico. Indica si el director es también el escritor de la película.

3 Limpieza de los datos.

A continuación se procederá a realizar la limpieza de cada una de las variables del dataset y en cada una se irá comentando como se gestionaron valores extremos, NA, ceros y elementos vacíos.

En este apartado también se crearán los distintos campos comentados anteriormente, derivados de la información contenida en el dataset original.

```
df <- read.csv(file="https://raw.githubusercontent.com/dbagan13/WebScraping/main/csv/recogiendo_tomates.csv", header=TRUE, sep=", ")
# Eliminando columna cargada como X
df <- subset(df, select = -c(X))
# Tratamiento inicial de nulos y cadenas vacías
df$Director[df$Director==""] <- NA
df$Producer[df$Producer==""] <- NA
df$Writer[df$Writer==""] <- NA
df$Production.Co[df$Production.Co==""] <- NA
```

Se eliminó la columna X ya que realmente no aporta al análisis. Se le asignó el valor NA a todos los registros que contienen cadenas vacías o nulos.

```
# Eliminando simbolo %
df.Tomatometer <- as.integer(str_remove_all(df$Tomatometer, "%"))
# Agregando la variable Tomatometer sin simbolo al dataset original
df$Tomatometer <- df.Tomatometer
# Conteo de NA
summary(df$Tomatometer)
```

Limpieza variable Tomatometer.

```
##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.   NA's
##      0.00  40.00  67.00  61.46  86.00 100.00      47
# Eliminando NA de Tomatometer
df <- df[complete.cases(df$Tomatometer), ]
```

Para el caso de la variable `Tomatometer`, se decidió remover el simbolo de %, para poder trabajar la variable como números enteros. También, por ser una variable de interés para futuros análisis y que sólo posee 47 NA, se decidió eliminar estos registros.

Creación de la variable Fresh. Como se comentó inicialmente, esta variable se calculará siguiendo el criterio establecido en la propia pág de Rotten Tomatoes; es decir:

- Fresh (Yes) si `Tomatometer > 60`
- Rotten si (No) `Tomatometer <= 60`

```

# La variable Fresh será igual a Yes cuando el Tomatometer sea >= 60 y No en caso
# contrario
df.fresh <- df$Tomatometer
df.fresh[df.fresh >= 60] <- "Yes"
df.fresh[df.fresh != "Yes"] <- "No"
df$Fresh <- df.fresh

```

Limpieza variable Audience.score. Al igual que se hizo con la variable Tomatometer, a esta variable le quitaremos el símbolo de % y eliminaremos los NA (331 registros), ya que esta es también una variable de interés para futuros análisis.

```

# Eliminando simbolo %
df.Audience.score <- as.integer(str_remove_all(df$Audience.score, "%"))
# Agregando la variable Audience Score sin símbolo al dataset original
df$Audience.score <- df.Audience.score
# conteo de NA
summary(df$Audience.score)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##      0.00    43.00    60.00    58.35    75.00   100.00      331

# Eliminando NA's de variable Audience Score.
df <- df[complete.cases(df$Audience.score), ]

```

Limpieza de variable Rating. En este caso, limpiaremos la variable tal que solo quede el tipo de clasificación que pertenezca cada película: G, NC-17, PG, PG-13, R, TV14, TVG, TVMA ó TVPG.

Para los casos en que la película no tuviera explícita una categoría, se decidió asignarles la G.

```

df.Rating <- str_remove_all(df$Rating, "\\([a-zA-Z \\.,|/-]*\\)")
# Agregando la variable Rating al dataframe original
df$Rating <- df.Rating
# Imputando el valor G en los valores vacíos de la variable Rating
df$Rating <- na_if(df$Rating, "")
df$Rating[is.na(df$Rating)] <- "G"
levels(factor(df$Rating))

## [1] "G"      "NC-17"   "PG"     "PG-13"   "R"      "TV14"   "TVG"    "TVMA"   "TVPG"

```

Creación de variable Parental.Control Esta variable se creó a partir de la variable Rating. Es una variable dicotómica donde el valor será 1 cuando la película pertenezca a la categoría G (público General) y será 0 cuando la película pertenezca a cualquier otra categoría que requiera de algún tipo de control parental.

```

p.control <- str_replace_all(df$Rating, "G", "Yes")
p.control <- str_replace_all(p.control, "NC-17|PG-13|PG|R|TV14|TVG|TVMA|TVPG",
                             "No")
p.control <- p.control
df["Parental.Control"] <- p.control

```

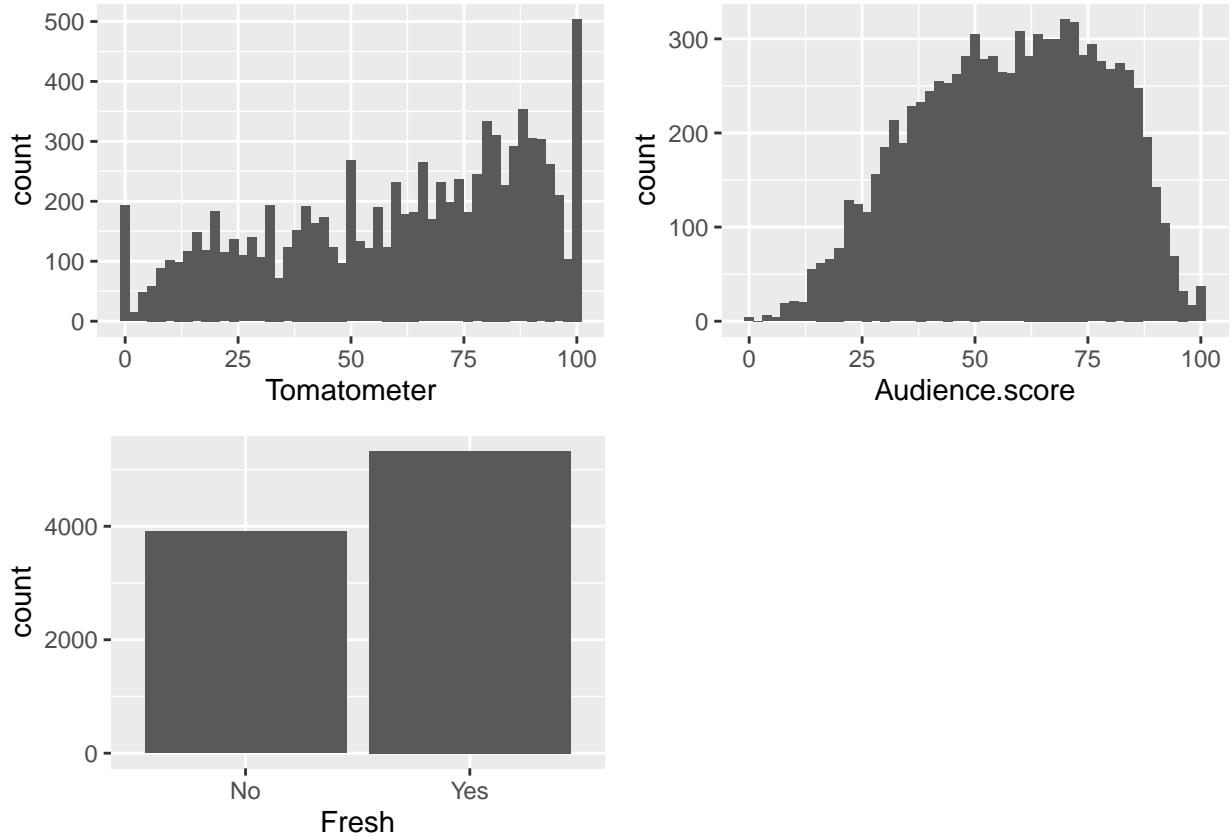
A continuación, se presenta un gráfico representativo para cada una de estas primeras 5 variables:

```

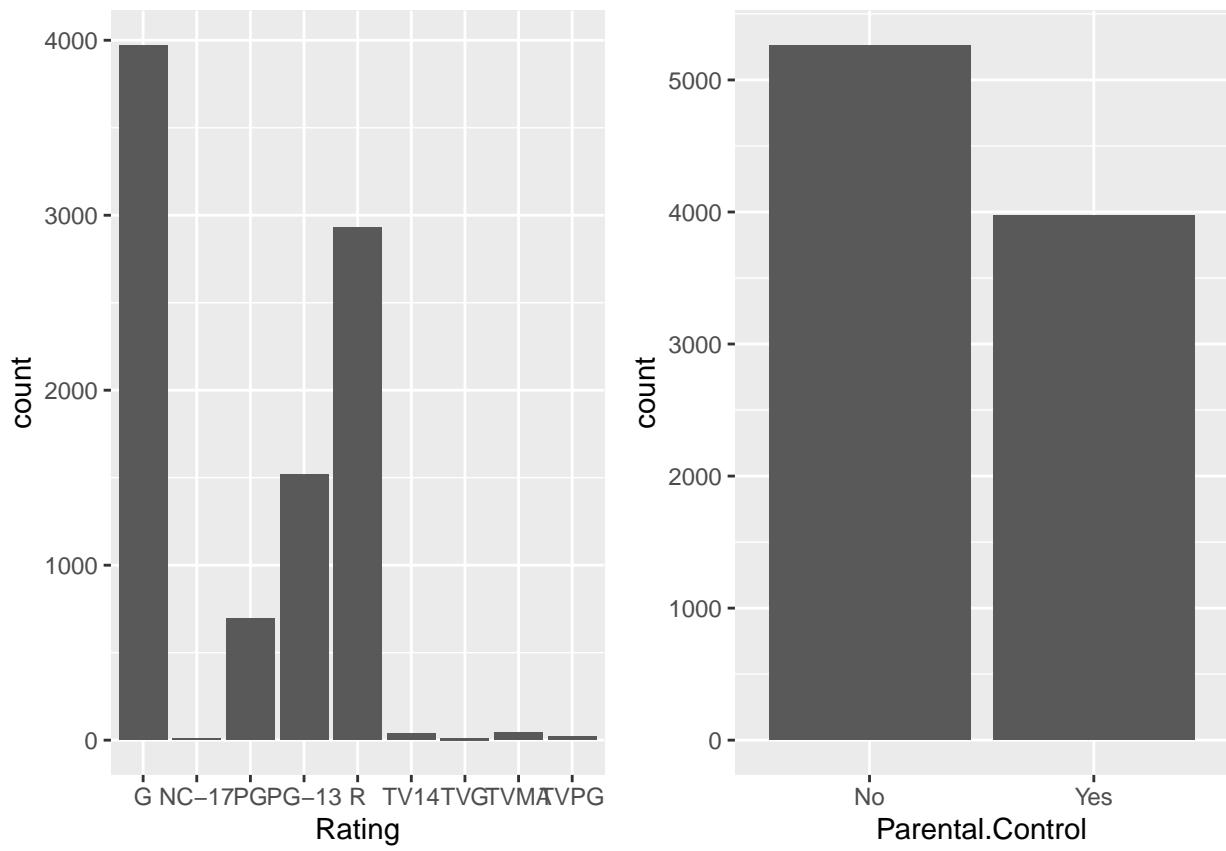
plot1 <- ggplot(df, aes(Tomatometer)) + geom_histogram(binwidth=2)
plot2 <- ggplot(df, aes(Audience.score)) + geom_histogram(binwidth = 2)
plot3 <- ggplot(df, aes(Fresh)) + geom_bar()
plot4 <- ggplot(df, aes(Rating)) + geom_bar()

```

```
plot5 <- ggplot(df, aes(Parental.Control)) + geom_bar()  
ggarrange(plot1, plot2, plot3)
```



```
ggarrange(plot4, plot5)
```



Tratamiento de variable Genre. Debido a que una misma película puede tener asignado más de un género, se dividirá esta variable en tres: Genre1, Genre2 y Genre3 y se imputarán en estas las tres principales géneros (si los tuviere) a los que pertenezca la película en cuestión, respectivamente

```
df.Genre <- df$Genre
# Generación variable Genre1
df.Genre1 <- str_extract(df.Genre, "[a-zA-Z]*")
#df.Genre1
# Generación variable Genre2
df.Genre2 <- str_extract(df.Genre, "[a-zA-Z]*")
df.Genre2 <- str_remove_all(df.Genre2, ", ")
#df.Genre2
# Generación variable Gen3
df.Genre3 <- str_extract(df.Genre, "([a-zA-Z]*,){2}[a-zA-Z]*")
df.Genre3 <- str_extract(df.Genre3, "[a-zA-Z]*$")
df.Genre3<- str_remove_all(df.Genre3, ", ")
#df.Genre3
# Eliminando variable Genre y agregando variables nuevas Gen1, Gen2 y Gen3
# al dataset original
df <- subset(df, select = -c(Genre))
df[["Genre1"]] <- df.Genre1
df[["Genre2"]] <- df.Genre2
df[["Genre3"]] <- df.Genre3
```

Tratamiento variable Runtime. El formato original de esta variable era una **string** que representaba la hora y los minutos tal que, por ejemplo: 1h30m. Este formato se cambió y se colocó la duración de la

película en minutos.

En aquellos casos donde existía un valor NA se imputó el valor de la media de la variable.

```
horas <- as.integer(substr(df$Runtime, 1, 1))
minutos <- as.integer(substr(df$Runtime, 3, 4))
minutos[is.na(minutos)] <- as.integer(str_sub(df$Runtime[is.na(minutos)],
                                                -2, -2))
minutos[is.na(minutos)] <- 0
runtime <- 60*horas+minutos
# Para aquellas películas que no tienen Runtime, se les ha colocado la media
runtime[is.na(runtime)] <- mean(runtime, na.rm = TRUE)
df$Runtime <- as.integer(runtime)
```

```
# Corrigiendo nombre de variables.
colnames(df)[8] <- "Release.Date.Theaters"
colnames(df)[9] <- "Release.Date.Streamings"
```

Tratamiento de variables Release.Date.Streaming y Release.Date. Theater. Creación de variable Realese.IsWide Estas dos variables se usaron en general para la creación de las variables derivadas Realese.IsWide y Seasons.

En el caso de Realise.IsWide, las palabras `wide` y `limited` se encontraban dentro de la string correspondiente a la variable `Realese.Date.Theaters`. Así que, en este caso, sólo hubo que filtrar la cadena de texto.

```
#Wide o limited
theaters <- df$Release.Date.Theaters
release_type <- df$Release.Date.Theaters
release_type[str_sub(release_type, -4, -1)=="wide"] <- "Yes"
release_type[str_sub(release_type, -7, -1)=="limited"] <- "No"
df$Release.isWide <- release_type
```

Para la creación de la variable Seasons realizó una limpieza de las variables `Release.Date.Streaming` y `Release.Date.Theaters` de tal forma que:

- En caso de que un película tuviera ambos campos, se determinó cual de las dos fechas fue primero.
- A través de una transformación de las cadenas de texto, se determinó en que Estación del año fue estrenada la película.

```
#Separacion fecha
months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
           "Oct", "Nov", "Dec")
streamings.year <- as.integer(str_sub(df$Release.Date.Streamings, -4, -1))
streamings.month <- substr(df$Release.Date.Streamings, 1, 3)
streamings.day <- as.integer(gsub(",","",substr(df$Release.Date.Streamings,
                                                 4, 5)))
streamings.month <- match(streamings.month,months)
streamings.date <- paste(str_sub(streamings.year, -2, -1), streamings.month,
                         streamings.day, sep="/")
streamings.date <- as.Date(streamings.date,format="%D")
```

```
#Limpieza
theaters <- str_remove_all(theaters,"limited")
theaters <- str_remove_all(theaters,"wide")
theaters <- str_sub(theaters, 1, -3)
#Separacion fecha
```

```

theaters.year <- as.integer(str_sub(theaters, -4, -1))
theaters.month <- substr(theaters, 1, 3)
theaters.day <- as.integer(gsub(",","",substr(theaters, 4, 5)))
theaters.month <- match(theaters.month,months)
theaters.date <- paste(str_sub(theaters.year, -2, -1),theaters.month,
                      theaters.day,sep="/")
theaters.date <- as.Date(theaters.date,format="%D")

na_date <- as.Date("68/12/31",format="%D")
theaters.date[is.na(theaters.date)] <- na_date
streamings.date[is.na(streamings.date)] <- na_date
df$Release.Date <- pmin(streamings.date, theaters.date)
df$Release.Date[df$Release.Date==na_date] <- NA
df$Release.YearDay <- as.integer(strftime(df$Release.Date, format = "%j"))
df$Release.Year <- strftime(df$Release.Date, format = "%Y")

#Estación
spring <- as.integer(strftime("2020-03-21", format = "%j"))
summer <- as.integer(strftime("2020-06-22", format = "%j"))
autumn <- as.integer(strftime("2020-09-23", format = "%j"))
winter <- as.integer(strftime("2020-12-22", format = "%j"))
df$Release.Season[!is.na(df$Release.Year)] <- "winter"
df$Release.Season[df$Release.YearDay>spring] <- "spring"
df$Release.Season[df$Release.YearDay>summer] <- "summer"
df$Release.Season[df$Release.YearDay>autumn] <- "autumn"
df$Release.Season[df$Release.YearDay>winter] <- "winter"

```

Finalmente, se eliminaron las columnas YearDay, Release.Date.Streamings y Release.Date.Theaters.

```

df$Release.YearDay <- NULL
df$Release.Date.Streamings <- NULL
df$Release.Date.Theaters <- NULL

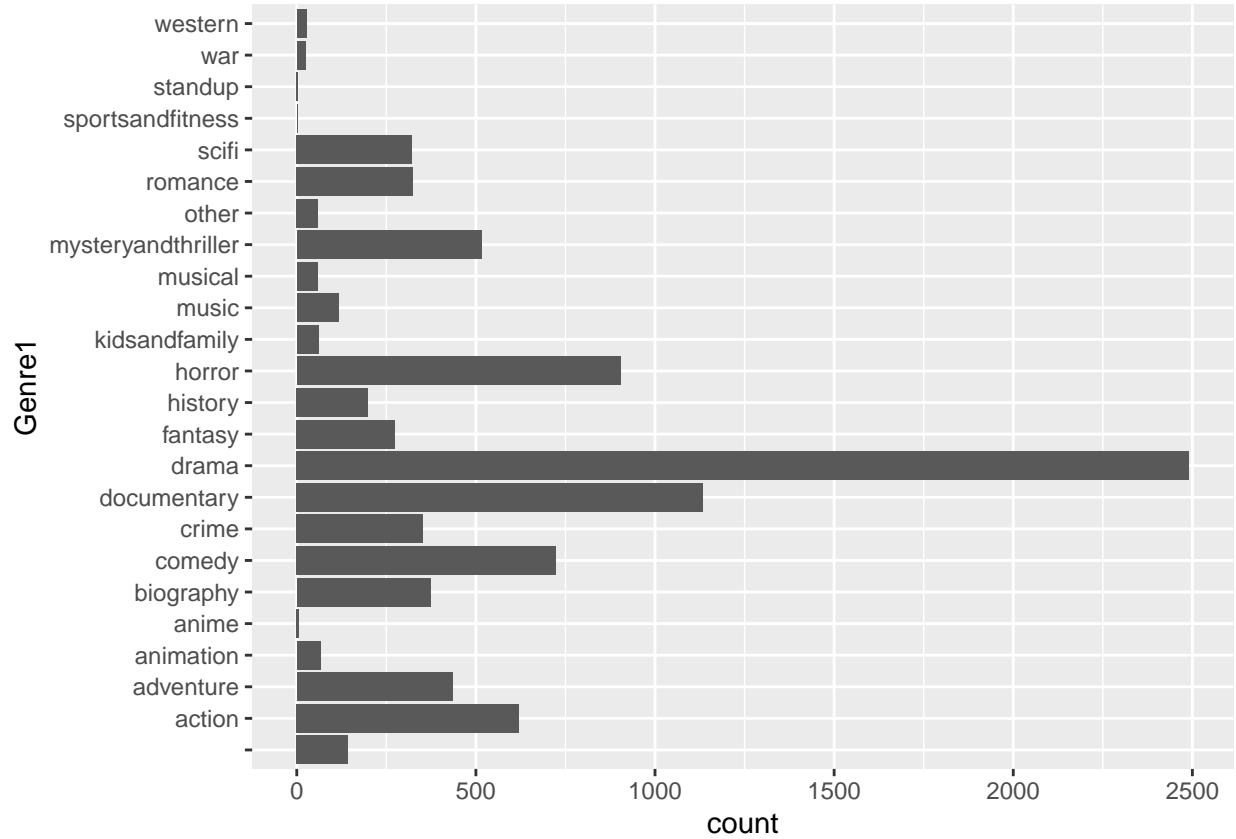
```

A continuación, se presentará una gráfica representativa de las variables: `Genre1`, `Runtime`, `Release.IsWide` y `Release.Season`:

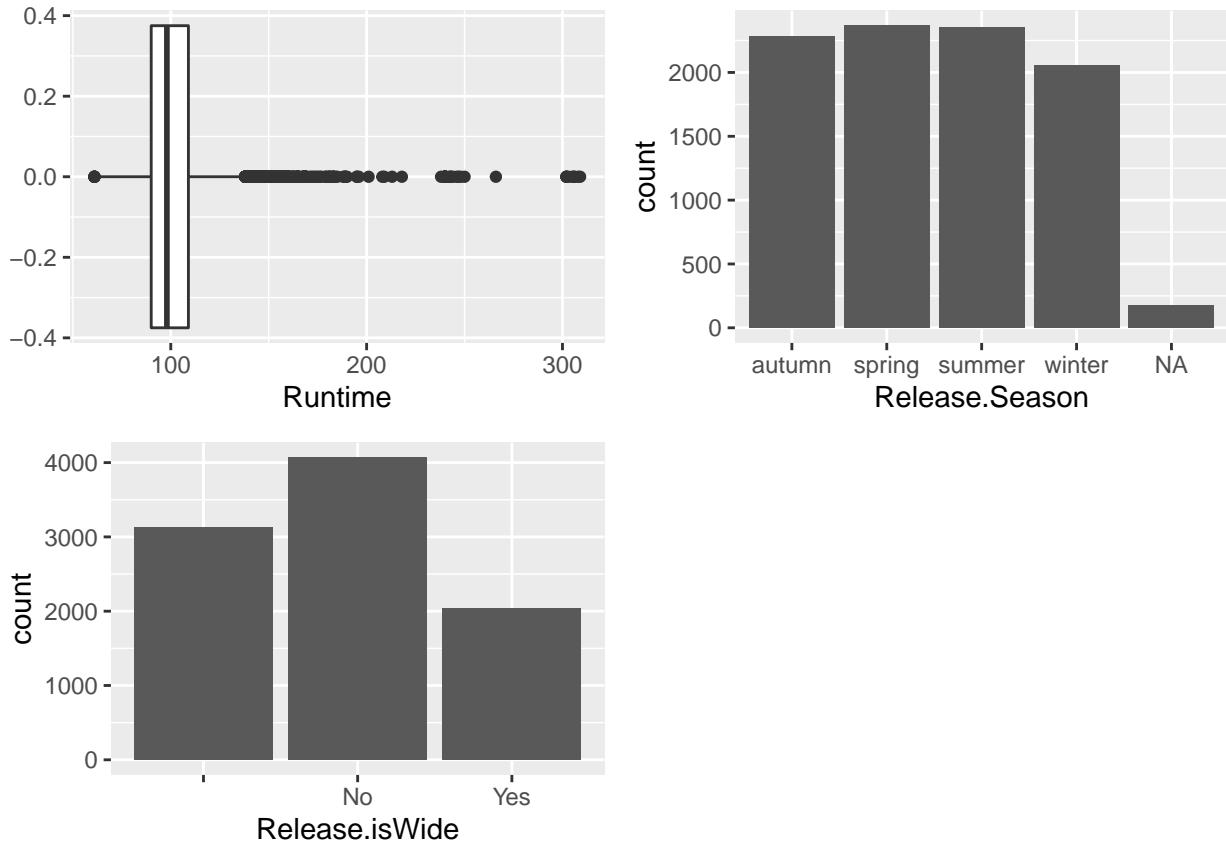
```

plot6 <- ggplot(df, aes(Genre1)) + geom_bar() + coord_flip()
plot7 <- ggplot(df, aes(Runtime)) + geom_boxplot()
plot8 <- ggplot(df, aes(Release.Season)) + geom_bar()
plot9 <- ggplot(df, aes(Release.isWide)) + geom_bar()
plot6

```



```
ggarrange(plot7, plot8, plot9)
```



De lo observado en los gráficos se puede detallar que:

- El género que más películas tiene es la categoría **drama**
- En cuanto a la duración de las películas, se observa que la media esta al rededor de 100min. Se puede ver que existen una serie de películas cuya duración es superior a 150min. Aunque estas aparezcan en el gráfico de Boxplot como valores extremos, no se eliminaron ya que son casos posibles.
- En cuanto a la variable **Release.Season**, se observa que existe una distribución uniforme a lo largo de todas las estaciones del años.
- En cuanto a la variable **Release.IsWide**, se observa que existe casi el doble de películas que sólo tiene un estreno local. Sólo al rededor de 2000 películas tuvieron un estreno mundial.

Tratamiento de variables Director, Production.Co, Producer y Writer. Para finalizar la limpieza de los datos, se creará la variable **DirectorIsWriter**, que servirá para determinar si el director es la misma persona que el escritor. Esto, para posteriores análisis, será interesante.

```

df$Director[df$Director=="" | df$Director=="UnknownDirector"] <- NA
# Creación de variable DirectorIsWriter
df$DirectorIsWriter<- (df$Director == df$Writer)
df$DirectorIsWriter[df$DirectorIsWriter] <- "Yes"
df$DirectorIsWriter[df$DirectorIsWriter =="FALSE"] <- "No"
df$Production.Co[is.na(df$Production.Co)] <- df$Producer[is.na(df$Production.Co)]
colnames(df)[9] <- "Production"
df$Writer <- NULL
df$Producer <- NULL

# Reordenando dataframe
df <- df[c("Title","Tomatometer","Fresh","Audience.score","Rating",

```

```
"Parental.Control", "Genre1", "Genre2", "Genre3", "Director", "Production",
"DirectorIsWriter", "Runtime", "Release.Date", "Release.Year",
"Release.Season", "Release.isWide")]
```

4 Análisis de los datos.

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

Correlación

En primer lugar, se realizará un análisis de la correlación entre las variables del juego de datos. Se utilizará una matriz de correlación para estimar las relaciones entre las variables numéricas del juego de datos (`Audience.score`, `Tomometer` y `Runtime`).

Además se presentarán gráficos con el fin de averiguar las relaciones de las variables con las puntuaciones de las películas y estimar su influencia.

Clasificación con regresión logística

Se creará un modelo de regresión logística para clasificar las películas según si son *fresh* o *rotten*, es decir, si tienen una crítica mayormente positiva o no (variable `Fresh`). Al igual que en el modelo de regresión, se seleccionarán las variables de entrada entre `Parental.Control`, `Genre1`, `Director`, `Runtime`, `Production`, `Release.isWide`, `Release.Year`, `Release.Season` y `DirectorIsWriter`.

Regresión lineal

Se realizará un análisis de regresión lineal múltiple para predecir la `Audience.score` de una película. Para ello se escogerán, a partir del análisis de correlación de las variables, cuáles son las variables de entrada que se utilizarán entre `Runtime`, `Parental.Control`, `Release.Year`, `Genre1`, `Release.isWide` y `DirectorIsWriter`.

Contraste de hipótesis

Se realizarán dos contrastes de hipótesis para tratar de responder dos preguntas:

- ¿Las películas dirigidas por el propio escritor son mejores (tienen mayor `Tomometer`) que las que tienen un director distinto? Se utilizarán las variables `Tomometer` y `DirectorIsWriter`.
- ¿Las películas del género drama tienen una mejor puntuación por parte la audiencia que las películas de terror? Se utilizarán las variables `Audience.score` y `Genre1`.

4.2 Comprobación de la normalidad y homogeneidad de la varianza.

En este apartado se analizará la normalidad y la homogenidad de la varianza de las variables aleatorias del dataset:

- `Tomometer`
- `Audience.Score`
- `Runtime`

El resto de variables del dataset son de carácter categórico o dicotómico y por ende no cabe en ellas tales análisis.

Para el estudio de normalidad, procederemos a aplicar tres métodos:

- **Método gráfico:** se graficará cada variable junto con una curva de distribución normal con la misma media y desviación estandar que muestra cada una. A partir de esto, se podrá observar si estas variables

se asemejan una distribución normal. Además, se graficarán los cuantiles de estas variables junto con los cuantiles teóricos de una distribución normal con misma media y desviación estandar y se evaluará si dichos cuantiles se encuentran más o menos alineados entorno a esta.

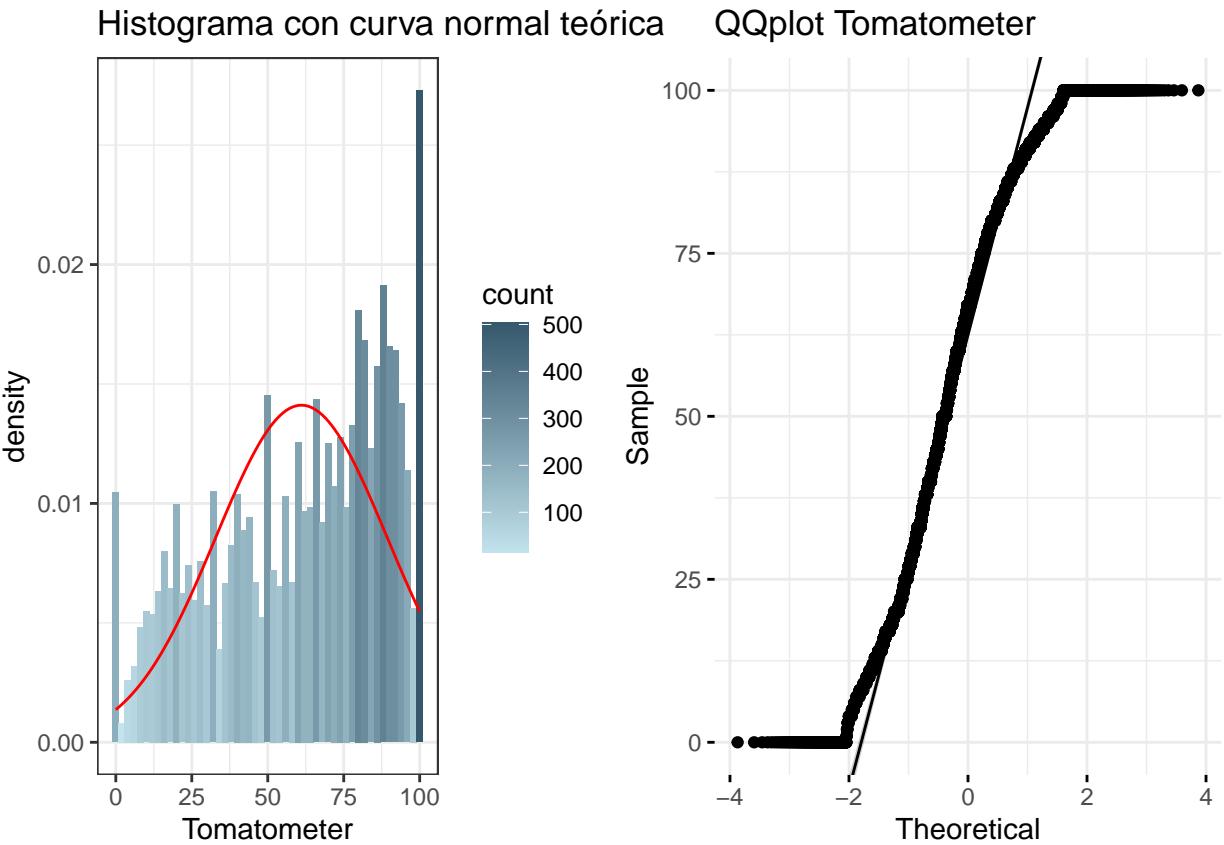
- **Método analítico:** Se calculará asimetría y curtosis de cada variable y se comprobará si presentan valores similares a los presentes en una distribución normal
- **Método de contraste:** Se realizará un test de Anderson-Darling.

Luego, una vez finalizada la aplicación de estos métodos, nos apoyaremos en los resultados obtenidos para llegar a una conclusión sobre la normalidad de estas variables.

4.2.1 Aplicación del Método Gráfico:

Variable Tomatometer :

```
# Gráfica de histograma de Tomatometer junto con una curva
# de distribución normal
plot1 <- ggplot(df, aes(Tatomometer)) +
  geom_histogram(aes(y = ..density.., fill = ..count..), binwidth = 2) +
  scale_fill_gradient(low = "#C3E4ED", high = "#35586C") +
  stat_function(fun = dnorm, colour = "red",
                args = list(mean = mean(df$Tatometer),
                            sd = sd(df$Tatometer))) +
  ggtitle("Histograma con curva normal teórica") +
  theme_bw()
plot2 <- ggqqplot(df, x="Tatometer", add = "qqline", ggtheme = theme_minimal(),
                  title = "QQplot Tomatometer", ylim = c(0,100) )
ggarrange(plot1, plot2)
```



De los gráficos anteriores de la variable **Tomatometer**, se puede señalar lo siguiente:

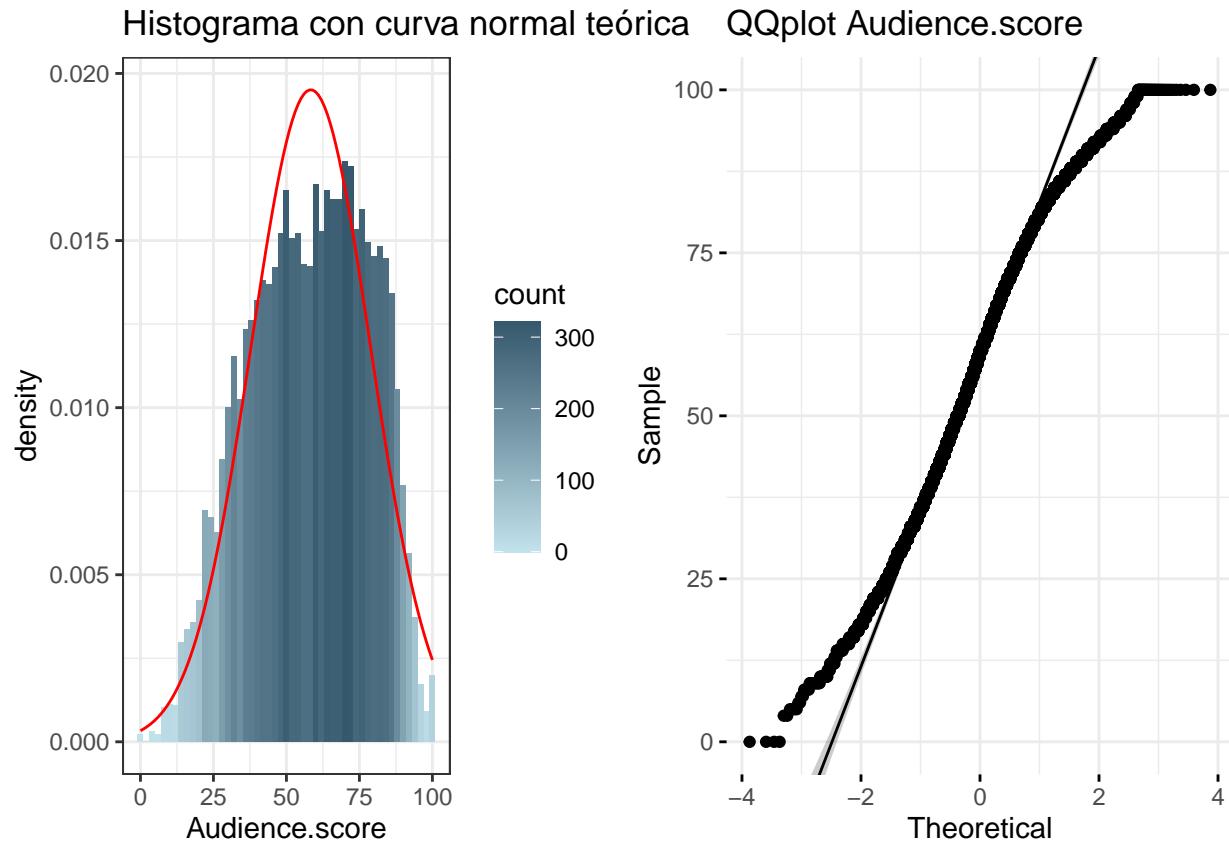
- En el histograma, visualmente, no se detalla que la variable siga una distribución normal. Existe un comportamiento marcado de gran cantidad de valores entre el 75% y 100%.
- En el gráfico QQ se observa que las películas que poseen un **Tomatometer** entre 12,5% y 87,5% se ajustan bastante bien a la línea recta teórica, pero luego encontramos en ambos extremos que los puntos se separan significativamente de la recta, obteniendo un resultado similar al de una gráfica QQ de una distribución uniforme; esto es: en el extremo derecho los puntos se encuentran por debajo de la recta, indicando que los valores en los datos no son tan extremos como lo serían los esperados por una distribución normal. De forma similar en el lado izquierdo, es decir: no son tan extremos como lo serían los esperados por una distribución normal. Esto se debe a que la variable **Tomatometer** posee como característica el hecho de que se encuentra acotada entre 0 y 100.

Variable Audience.score :

```
plot1 <- ggplot(df, aes(Audience.score)) +
  geom_histogram(aes(y = ..density.., fill = ..count..), binwidth = 2) +
  scale_fill_gradient(low = "#C3E4ED", high = "#35586C") +
  stat_function(fun = dnorm, colour = "red",
                args = list(mean = mean(df$Audience.score),
                            sd = sd(df$Audience.score))) +
  ggtitle("Histograma con curva normal teórica") +
  theme_bw()

plot2 <- ggqqplot(df, x="Audience.score", add = "qqline",
                  ggtheme = theme_minimal(),
                  title = "QQplot Audience.score" , ylim = c(0,100))
```

```
ggarrange(plot1,plot2)
```

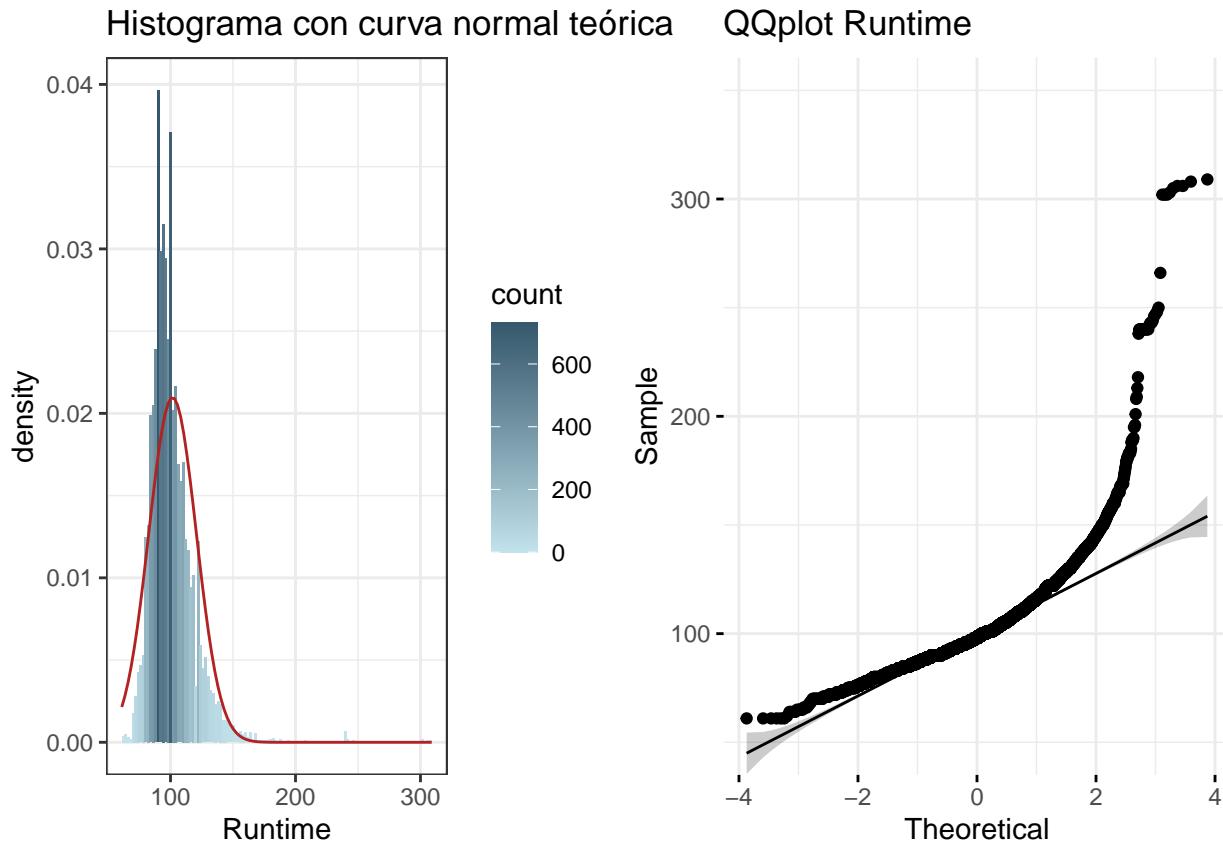


De los gráficos de la variable `Audience.score` se observa:

- En el histograma se detalla un comportamiento similar a una curva de distribución normal.
- En el gráfico QQ se vuelve a observar, al igual que con la variable `Tomatometer`, que en los extremos los datos se separan de la recta. Esto, de nuevo, es debido a la característica propia de la variable de ser acotada entre 0 y 100.

Variable `Runtime` :

```
plot1 <- ggplot(df, aes(Runtime)) +
  geom_histogram(aes(y = ..density.., fill = ..count..), binwidth = 2) +
  scale_fill_gradient(low = "#C3E4ED", high = "#35586C") +
  stat_function(fun = dnorm, colour = "firebrick",
                args = list(mean = mean(df$Runtime),
                            sd = sd(df$Runtime))) +
  ggtitle("Histograma con curva normal teórica") +
  theme_bw()
plot2 <- gqqplot(df, x="Runtime", add = "qqline",
                 ggtheme = theme_minimal(),
                 title = "QQplot Runtime" , ylim = c(50,350))
ggarrange(plot1,plot2)
```



De los gráficos de la variable `Runtime` se detalla:

- El histograma presenta una fuerte concentración de datos aproximadamente entorno a los valores 90 - 100. También se observa una distribución con asimetría hacia la derecha (positiva), presentando una larga cola derecha.
- En el gráfico QQ se comprueba el comportamiento señalado en el histograma: los datos a la derecha de la curva se encuentran significativamente por encima de la recta, lo cual indica que estos poseen valores mucho mayores a los esperados en una distribución normal con media y desviación estandar similar al conjunto de datos.

4.2.2 Aplicación del método Analítico:

En este apartado se aplicará un estudio de Curtosis y asimetría a cada una de las variables.

```
# Cálculo de la curtosis
curtosis <- kurtosi(df$Tomatometer)
# Cálculo de la asimetría
asimetria <- skew(df$Tomatometer)
sprintf("Curtosis: %.3f, Asimetría: %.3f", curtosis, asimetria)
```

Variable Tomatometer:

```
## [1] "Curtosis: -0.945, Asimetría: -0.452"
```

Los valores obtenidos, tanto en la curtosis como en la asimetría, están en el rango de -1 a 1, lo cual señala una leve desviación de la normalidad.

```

# Cálculo de la curtosis
curtosis <- kurtosi(df$Audience.score)
# Cálculo de la asimetría
asimetria <- skew(df$Audience.score)
sprintf("Curtosis: %.3f, Asimetría: %.3f", curtosis, asimetria)

```

Variable Audience Score:

```
## [1] "Curtosis: -0.803, Asimetría: -0.206"
```

Los valores de curtosis y asimetría de la variable `Audience.score` señalan también una leve desviación de la normalidad. Pese a este resultado, todavía se podría considerar como cierto el supuesto de normalidad de esta variable.

```

# Cálculo de la curtosis
curtosis <- kurtosi(df$Runtime)
# Cálculo de la asimetría
asimetria <- skew(df$Runtime)
sprintf("Curtosis: %.3f, Asimetría: %.3f", curtosis, asimetria)

```

Variable Runtime:

```
## [1] "Curtosis: 20.635, Asimetría: 2.986"
```

En este caso, tanto la curtosis como la asimetría señalan que no es una curva normal. El valor elevado de la curtosis se debe a la forma “puntiaguda” de la curva producto de la concentración de datos en torno a los valores de 90 y 100. Por su parte, el valor de la asimetría se debe al sesgo derecho que presenta la curva.

De acuerdo a estos resultados, no se puede aceptar el supuesto de normalidad en esta variable.

4.2.3 Aplicación del método de contraste.

Debido a que la muestra es mucho mayor a 5000 registros, el test de Shapiro-Wilk no puede ser aplicado en R. Por su parte, el test de Kolmogorov-Smirnov, en la documentación correspondiente a su implementación en R se señala que este test puede presentar errores si en la data tenemos valores repetidos (que es nuestro caso), es por esto que se usará el test de Anderson-Darling para comprobar la normalidad en estas variables.

Este test tiene como hipótesis nula: ***Los datos tienen una distribución normal.*** Así, si el p-valor < nivel de significación (0.05), se rechazará la hipótesis nula.

```

# variable Tomatometer.
ad.test(df$Tomatometer)

##
## Anderson-Darling normality test
##
## data: df$Tomatometer
## A = 164.74, p-value < 2.2e-16

# variable Audience.score
ad.test(df$Audience.score)

##
## Anderson-Darling normality test
##
## data: df$Audience.score
## A = 46.697, p-value < 2.2e-16

```

```

# variable Runtime
ad.test(df$Runtime)

##
## Anderson-Darling normality test
##
## data: df$Runtime
## A = 248.48, p-value < 2.2e-16

```

Para las tres variables, el p-valor es menor a 0.05, por lo que, de acuerdo a este método de contraste, se debe rechazar el supuesto de normalidad.

Finalmente, a partir de los métodos utilizados para comprobar la normalidad en las variables, se puede concluir:

- Las variables `Tomatometer`, `Audience.score` y `Runtime` no poseen una distribución normal. Pese a esto, de acuerdo al teorema del límite central, se puede afirmar que al ser una muestra lo suficientemente grande, la media de la muestra se acerca a la media poblacional.

4.2.4 Comprobación de homogeneidad en la varianza.

A continuación, se estudiará la homogeneidad de varianzas utilizando el test de **Fligner-Killeen**. Este se trata de un método no paramétrico. Es también una alternativa cuando no se cumple la condición de normalidad entre las muestras.

Se aplicará el test de homogeneidad de varianza para conocer si existe variación de esta en:

- Las variables `Tomatometer`, `Audience.score` y `Runtime`, dependiendo si en cada caso la película tiene alguna restricción de control parental o no.
- Las variables `Tomatometer`, `Audience.score` y `Runtime`, dependiendo al `Genre1` que pertenezca la película.
- Las variables `Tomatometer`, `Audience.score` y `Runtime`, dependiendo de la época del año en que fueron estrenadas (variable `Release.Season`)

```

# Para variable Tomatometer
fligner.test(df$Tomatometer ~ df$Parental.Control)

```

Homogeneidad en la varianza dependiendo de Parental.control

```

##
## Fligner-Killeen test of homogeneity of variances
##
## data: df$Tomatometer by df$Parental.Control
## Fligner-Killeen:med chi-squared = 132.22, df = 1, p-value < 2.2e-16
# Para variable Audience.Score
fligner.test(df$Audience.score ~ df$Parental.Control)

```

```

##
## Fligner-Killeen test of homogeneity of variances
##
## data: df$Audience.score by df$Parental.Control
## Fligner-Killeen:med chi-squared = 0.13414, df = 1, p-value = 0.7142
# Para variable Runtime
fligner.test(df$Runtime ~ df$Parental.Control)

```

```

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: df$Runtime by df$Parental.Control
## Fligner-Killeen:med chi-squared = 0.19308, df = 1, p-value = 0.6604

```

En el caso de muestras divididas a partir de la variable `Parental.Control`, se observa que:

- con respecto a la variable `Tomatometer`, el p-value < 0.05, por lo que se rechaza la hipótesis nula de que exista homogeneidad en la varianza
- con respecto a las variables `Audience.scroe` y `Runtime`, el p-value en ambos casos es mayor a 0.05, por lo que se acepta la hipótesis nula de que la varianza es homogena.

```

# Para variable Tomatometer
fligner.test(df$Tomatometer ~ df$Genre1)

```

Homogeneidad en la varianza dependiendo de `Genre1`

```

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: df$Tomatometer by df$Genre1
## Fligner-Killeen:med chi-squared = 513.78, df = 23, p-value < 2.2e-16
# Para variable Audience.Score
fligner.test(df$Audience.score ~ df$Genre1)

```

```

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: df$Audience.score by df$Genre1
## Fligner-Killeen:med chi-squared = 193.83, df = 23, p-value < 2.2e-16
# Para variable Runtime
fligner.test(df$Runtime ~ df$Genre1)

```

```

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: df$Runtime by df$Genre1
## Fligner-Killeen:med chi-squared = 662.05, df = 23, p-value < 2.2e-16

```

Se observa en este caso que para las 24 muestras separadas por el genero al que pertenece cada película, en ninguno de los casos evaluados el p-valor fue superior a 0.05, por lo que se rechaza la posibilidad de homogeneidad en la varianza.

```

# Para variable Tomatometer
fligner.test(df$Tomatometer ~ df$Release.Season)

```

Homogeneidad en la varianza dependiendo de `Release.Seasons`

```

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: df$Tomatometer by df$Release.Season
## Fligner-Killeen:med chi-squared = 7.5746, df = 3, p-value = 0.05567

```

```

# Para variable Audience.Score
fligner.test(df$Audience.score ~ df$Release.Season)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: df$Audience.score by df$Release.Season
## Fligner-Killeen:med chi-squared = 4.8365, df = 3, p-value = 0.1842
# Para variable Runtime
fligner.test(df$Runtime ~ df$Release.Season)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: df$Runtime by df$Release.Season
## Fligner-Killeen:med chi-squared = 15.212, df = 3, p-value = 0.001644

```

Por último, se puede observar que en los 4 grupos formados por la variable `Release.Season`, para los casos estudiados observamos que :

- el p-valor en la variable `Tomatometeres` ligeramente superio a 0.05, pero lo suficiente como para aceptar que existe homogeneidad en la varianza.
- el p-valor en la variable `Audience.score` es > 0.05 por lo que nuevamente se acepta la hipótesis de homogeneidad.
- el p-valor de la variable `Runtimess` inferior a 0.05, por lo que se rechaza la hipótesis de homogeneidad en la varianza.

4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos

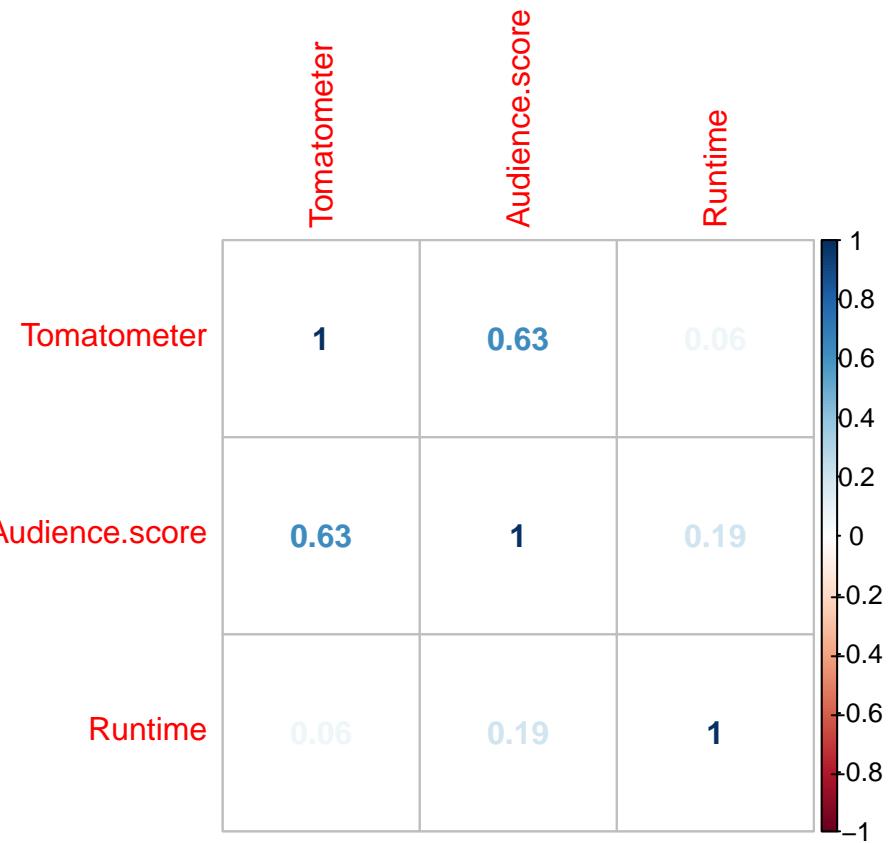
Correlación

Como se ha comentado en el apartado 4.1, en primer lugar se crea la matriz de correlación, utilizando las variables numéricas y las dicotómicas, y se comentan las correlaciones más apreciables.

```

cor.df <- cor(df[c(2,4,13)], use="complete.obs")
corrplot(cor.df, method = 'number')

```

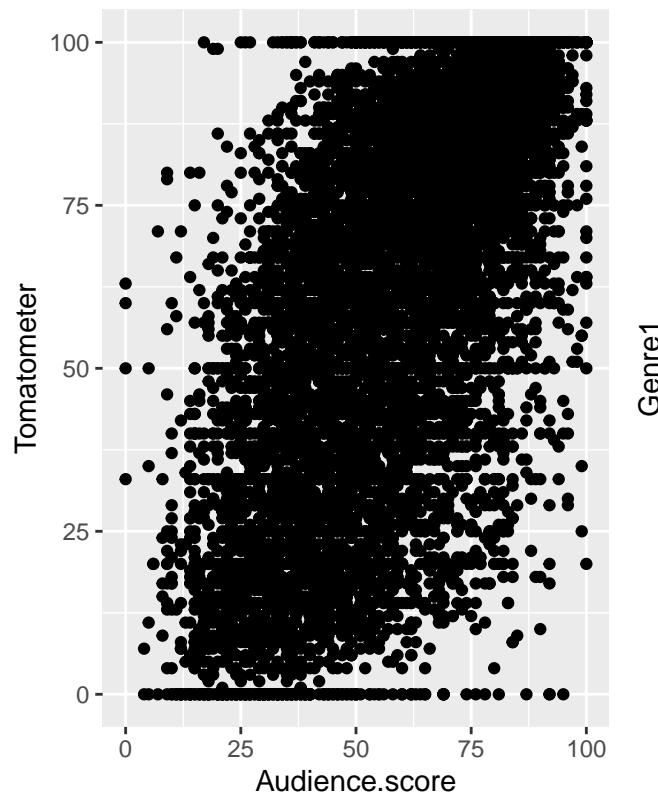


- Existe una correlación evidente entre **Tomatometer** y **Audience.score**, ya que ambas variables puntuán la valoración de las películas. Sin embargo, esta correlación es de 0,62, y aunque la correlación es clara, no es tan fuerte como se podría esperar.
- Las películas tienen una ligera tendencia a estar mejor puntuadas cuando su duración es mayor, ya que la correlación de **Audience.Score** y **Runtime** es positiva, 0.19.

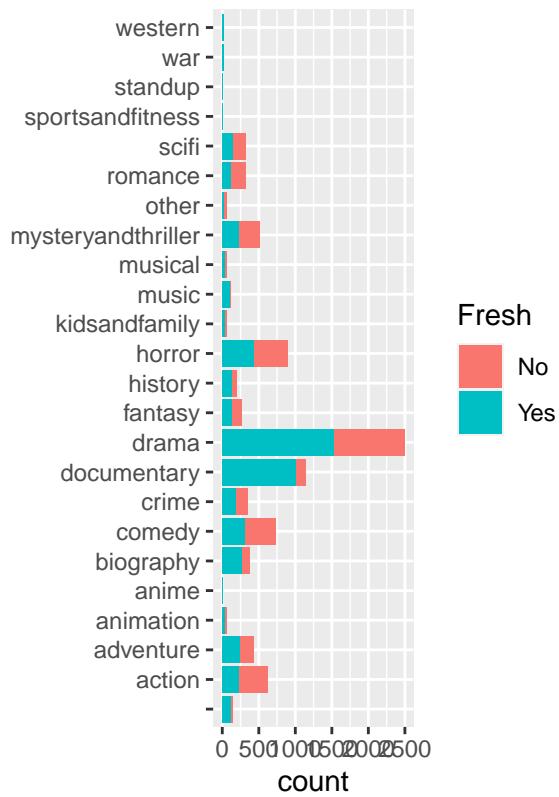
A continuación se representa la relación entre las “etiquetas” Audience.Score, Tomatometer y Fresh con las demás variables.

```
p41 <- ggplot(aes(y = Tomatometer, x = Audience.score), data = df) +
  geom_point() + ggtitle("Audience Score and Tomatometer")
p42 <- ggplot(data = df, aes(x=Genre1, fill=Fresh)) + geom_bar() +
  coord_flip() + ggtitle("Fresh by genre")
grid.arrange(p41, p42, ncol=2, nrow=1)
```

Audience Score and Tomatometer



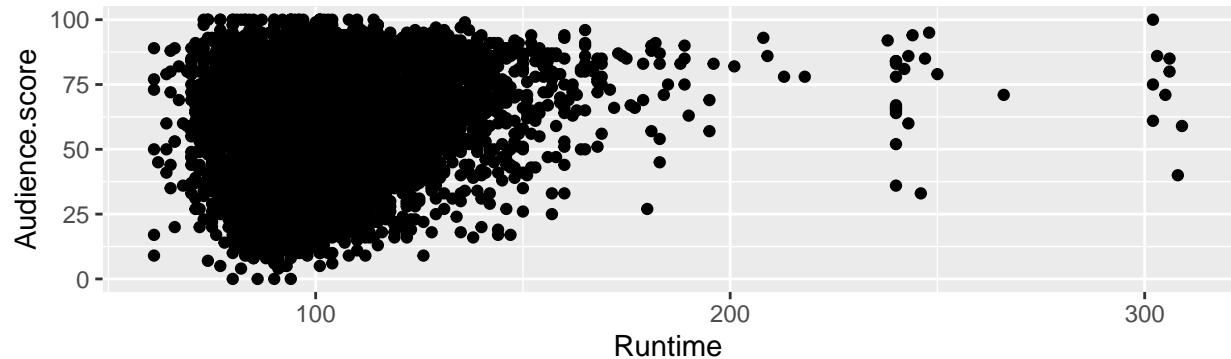
Fresh by genre



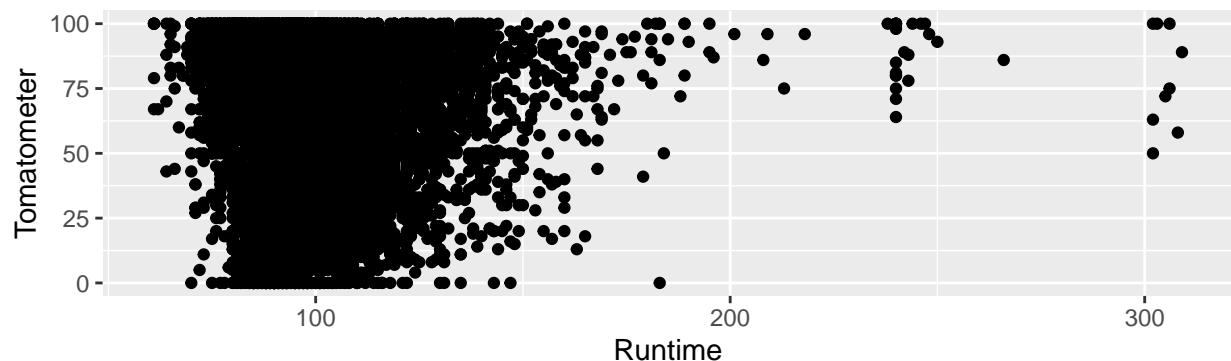
- De igual manera que en la matriz de relaciones, Tomatometer y Audience.score presentan una relación directa, generalmente cuando crece Audience.score lo hace también Tomatometer.
- En cuanto a los géneros, destaca los documentales y las biografías, que tienen un porcentaje de *fresh* alto, mientras que comedia o acción tienden a tener críticas negativas. El género con mayor número de películas es drama.

```
p43 <- ggplot(df, aes(x=Runtime, y=Audience.score)) +
  geom_point() + ggtitle("Audience score by runtime")
p44 <- ggplot(df, aes(x=Runtime, y=Tomatometer)) +
  geom_point() + ggtitle("Tomatometer score by runtime")
grid.arrange(p43, p44, ncol=1, nrow=2)
```

Audience score by runtime

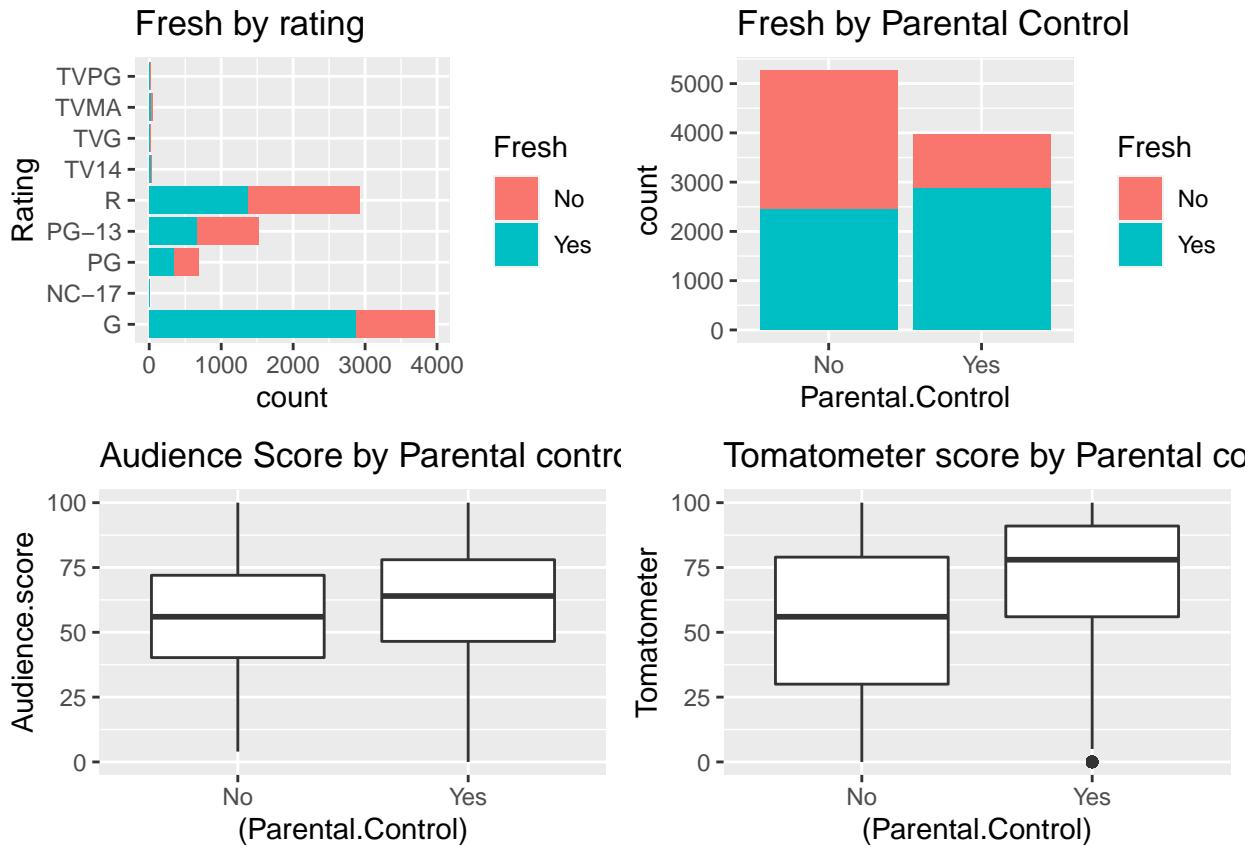


Tomatometer score by runtime



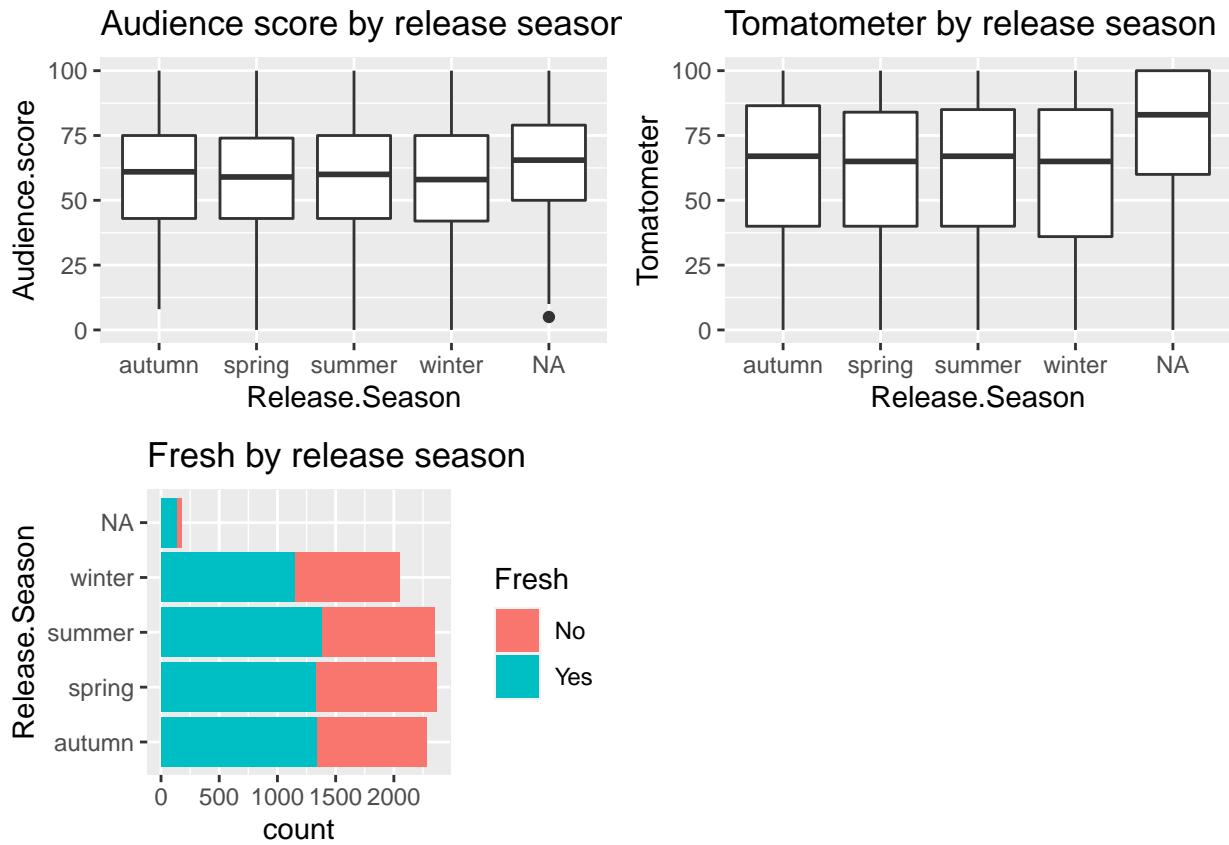
- En cuanto al Runtime, se aprecia una ligera tendencia a que las películas con mayor duración tengan una mayor puntuación, tal y como se indicaba en la matriz de correlaciones.

```
p45 <- ggplot(data = df, aes(x=Rating, fill=Fresh))+geom_bar() + coord_flip() +
  ggtitle("Fresh by rating")
p46 <- ggplot(data = df, aes(x=Parental.Control, fill=Fresh))+geom_bar() +
  ggtitle("Fresh by Parental Control")
p47 <- ggplot(aes(y = Audience.score, x = (Parental.Control)), data = df) +
  geom_boxplot() + ggtitle("Audience Score by Parental control")
p48 <- ggplot(aes(y = Tomatometer, x = (Parental.Control)), data = df) +
  geom_boxplot() + ggtitle("Tomatometer score by Parental control")
grid.arrange(p45, p46, p47, p48, ncol=2, nrow=2)
```



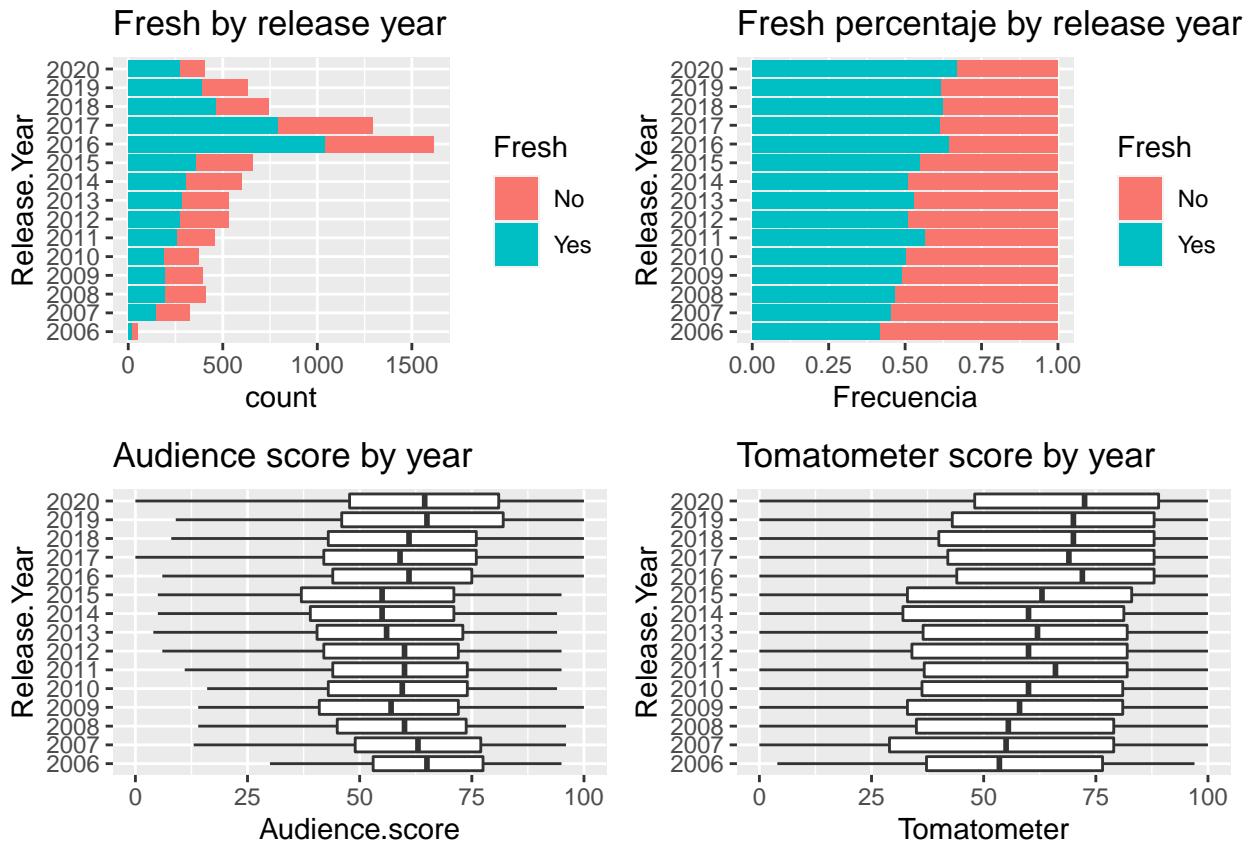
- En cuanto a las restricciones de edad de la películas, el gráfico de arriba a la derecha muestra claramente que el porcentaje de películas con crítica positiva es mucho mayor en las películas con restricciones que en las que son para todos los públicos.
- Llama la atención que esta diferencia es mucho más alta en el caso de las puntuaciones de críticos (Tomatometer) que en el público general.

```
p49 <- ggplot(data = df, aes(x=Release.Season, fill=Fresh)) + geom_bar() + coord_flip() +
  ggtitle("Fresh by release season")
p410 <- ggplot(aes(y = Audience.score, x = Release.Season), data = df) +
  geom_boxplot() + ggtitle("Audience score by release season")
p411 <- ggplot(aes(y = Tomatometer, x = Release.Season), data = df) + geom_boxplot() +
  ggtitle("Tomatometer by release season")
grid.arrange(p410, p411, p49, ncol=2, nrow=2)
```



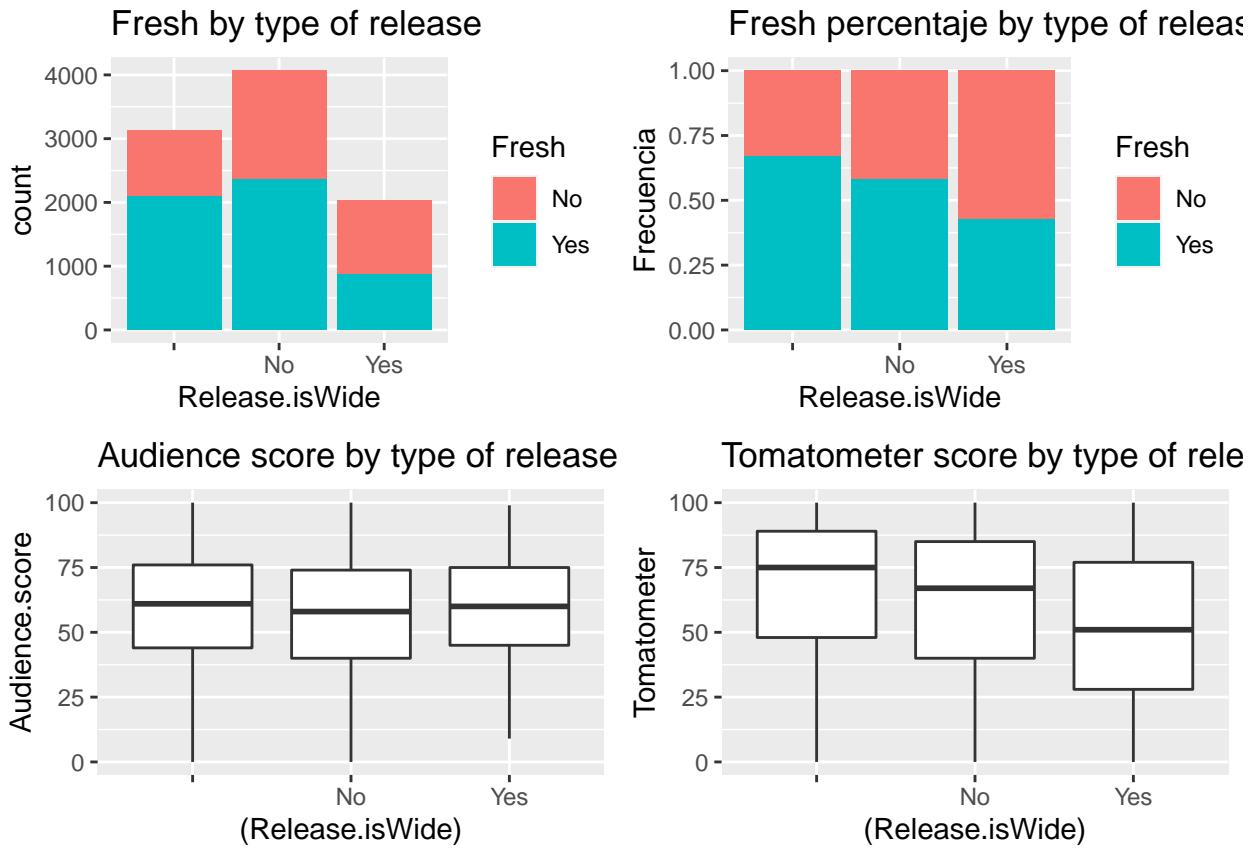
- En cuanto a la estación en que se estrenan las películas, no se aprecia ninguna tendencia significativa, más allá de que en invierno se estrenan menos películas que el resto del año.

```
p412 <- ggplot(data = subset(df, Release.Year>2005),
                 aes(x=Release.Year, fill=Fresh))+geom_bar() +
                 ggttitle("Fresh by release year") + coord_flip()
p411a <- ggplot(data = subset(df, Release.Year>2005),
                  aes(x=Release.Year,fill=Fresh))+geom_bar(position="fill") +
                  ggttitle("Fresh porcentaje by release year") +
                  coord_flip() +ylab("Frecuencia")
p413 <- ggplot(aes(y = Audience.score, x = Release.Year),
                 data = subset(df, Release.Year>2005)) + geom_boxplot() +
                 ggttitle("Audience score by year") + coord_flip()
p414 <- ggplot(aes(y = Tomatometer, x = Release.Year),
                 data = subset(df, Release.Year>2005)) + geom_boxplot() +
                 ggttitle("Tomatometer score by year") + coord_flip()
grid.arrange(p412, p411a, p413, p414, ncol=2, nrow=2)
```



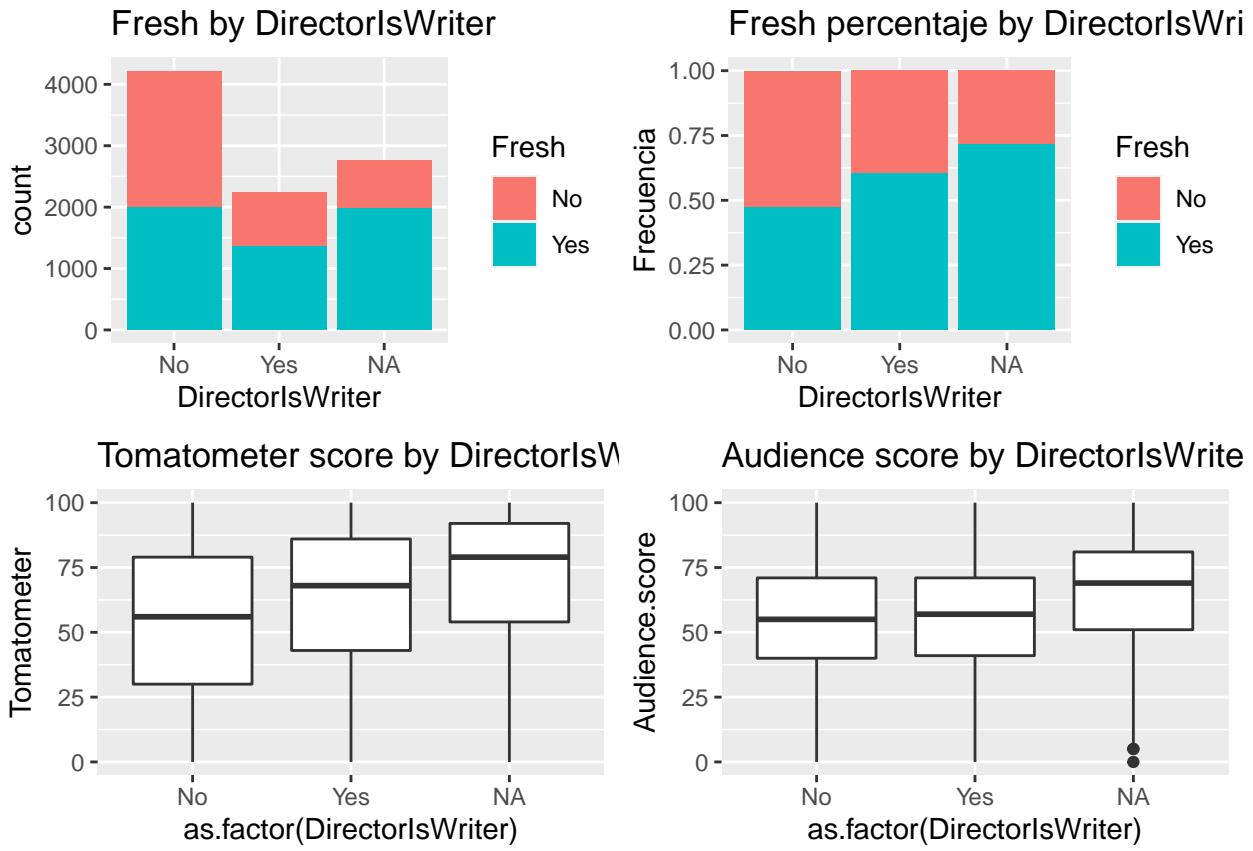
- En cuanto al año de estreno de las películas, se puede apreciar que la opinión de la crítica ha ido mejorando paulatinamente con el paso de los años, y cada año el porcentaje de películas con críticas positivas parece mejorar ligeramente.
- Por otra parte, en 2006 se alcanzó un pico de estrenos, y desde ese año cada año se estrenan menos películas.

```
p415 <- ggplot(data = df,aes(x=Release.isWide,fill=Fresh))+geom_bar() +
  ggtitle("Fresh by type of release")
p415a <- ggplot(data = df,aes(x=Release.isWide,fill=Fresh)) +
  geom_bar(position="fill") +
  ggtitle("Fresh porcentaje by type of release") +
  ylab("Frecuencia")
p416 <- ggplot(aes(y = Audience.score, x = (Release.isWide)), data = df) +
  geom_boxplot() + ggtitle("Audience score by type of release")
p417 <- ggplot(aes(y = Tomatometer, x = (Release.isWide)), data = df) +
  geom_boxplot() + ggtitle("Tomatometer score by type of release")
grid.arrange(p415, p415a, p416, p417, ncol=2, nrow=2)
```



- Curiosamente, las películas con un estreno “limited” tienen un mayor porcentaje de críticas positivas por parte de los críticos que las “wide”, mientras que en el caso de la audiencia no es así.

```
p418 <- ggplot(data = df,aes(x=DirectorIsWriter,fill=Fresh)) +
  geom_bar() + ggttitle("Fresh by DirectorIsWriter")
p418a <- ggplot(data = df,aes(x=DirectorIsWriter,fill=Fresh)) +
  geom_bar(position="fill") +
  ggttitle("Fresh porcentaje by DirectorIsWriter") +
  ylab("Frecuencia")
p419 <- ggplot(aes(y = Audience.score, x = as.factor(DirectorIsWriter)),
  data = df) + geom_boxplot() +
  ggttitle("Audience score by DirectorIsWriter")
p420 <- ggplot(aes(y = Tomatometer, x = as.factor(DirectorIsWriter)),
  data = df) + geom_boxplot() +
  ggttitle("Tomatometer score by DirectorIsWriter")
grid.arrange(p418, p418a, p420, p419, ncol=2, nrow=2)
```



- En cuanto las películas que han sido dirigidas por el propio escritor, al igual que sucede con el tipo de estreno, se aprecia una diferencia clara en el caso de las puntuaciones de los críticos (Tomatometer y Fresh) en favor de las películas que han sido dirigidas por su escritor, mientras que en el caso del público general no se aprecia esta tendencia.

Clasificación con regresión logística

A continuación, se crea un modelo de regresión logística, con regresores tanto cuantitativos como cualitativos, para tratar de predecir el valor de la etiqueta dicotómica Fresh de las distintas películas, es decir, clasificarlas en “fresh” o “rotten”.

A partir del análisis de correlaciones, se usarán las variables que tienen una correlación apreciable con la etiqueta Fresh: Parental.Control, DirectorIsWriter, Runtime, Genre1, Release.Year y Release.isWide.

En primer lugar, se preparan los datos para la creación del modelo, seleccionando las variables especificadas y eliminando los elementos nulos. Se eliminan los años anteriores al 2006, ya que tienen una cantidad de registros muy escasa. Además, se genera una **distribución de Bernoulli con la etiqueta Fresh**.

```
# Preparación de datos
df2 <- df[c("Audience.score", "Fresh", "Parental.Control", "DirectorIsWriter",
           "Runtime", "Release.Year", "Release.isWide", "Genre1")]
df2 <- df2[complete.cases(df2), ]
df2 <- df2[df2$Release.Year>2005,]
Fresh.log <- df2$Fresh
Fresh.log[df2$Fresh=="Yes"] <- 1
Fresh.log[df2$Fresh=="No"] <- 0
Fresh.log <- as.integer(Fresh.log)
```

Se crea el modelo de regresión logística, donde la variable dependiente es Fresh y las independientes Parental.Control, DirectorIsWriter, Runtime, Gendre1, Release.Year y Release.isWide.

```

attach(df2)
#Creación del modelo
model.log=glm(formula=Fresh.log~Runtime+factor(Parental.Control) +
               factor(Release.Year) + factor(DirectorIsWriter) +
               factor(Release.isWide) +
               factor(Genre1),family=binomial(link=logit))
summary(model.log)

##
## Call:
## glm(formula = Fresh.log ~ Runtime + factor(Parental.Control) +
##       factor(Release.Year) + factor(DirectorIsWriter) + factor(Release.isWide) +
##       factor(Genre1), family = binomial(link = logit))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max 
## -3.0748 -1.0747  0.5373  1.0723  1.8545 
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                 -14.699708 324.743766 -0.045 0.963896  
## Runtime                      0.015390  0.001847  8.331 < 2e-16 ***
## factor(Parental.Control)Yes  0.561641  0.064349  8.728 < 2e-16 ***
## factor(Release.Year)2007     11.906877 324.743886  0.037 0.970752  
## factor(Release.Year)2008     12.127277 324.743741  0.037 0.970211  
## factor(Release.Year)2009     12.145367 324.743726  0.037 0.970166  
## factor(Release.Year)2010     12.284733 324.743723  0.038 0.969824  
## factor(Release.Year)2011     12.484351 324.743719  0.038 0.969334  
## factor(Release.Year)2012     12.240183 324.743717  0.038 0.969933  
## factor(Release.Year)2013     12.295110 324.743717  0.038 0.969799  
## factor(Release.Year)2014     12.221528 324.743715  0.038 0.969979  
## factor(Release.Year)2015     12.311962 324.743714  0.038 0.969757  
## factor(Release.Year)2016     12.500724 324.743711  0.038 0.969294  
## factor(Release.Year)2017     12.448998 324.743712  0.038 0.969421  
## factor(Release.Year)2018     12.688182 324.743714  0.039 0.968833  
## factor(Release.Year)2019     12.702811 324.743716  0.039 0.968798  
## factor(Release.Year)2020     12.748173 324.743729  0.039 0.968686  
## factor(DirectorIsWriter)Yes  0.388977  0.057736  6.737 1.61e-11 ***
## factor(Release.isWide)No     -0.190777  0.074918 -2.546 0.010881 *  
## factor(Release.isWide)Yes    -0.320916  0.092327 -3.476 0.000509 ***  
## factor(Genre1)adventure     0.905448  0.144434  6.269 3.64e-10 ***  
## factor(Genre1)animation     1.300402  0.293826  4.426 9.61e-06 ***  
## factor(Genre1)anime         13.099646 150.338572  0.087 0.930565  
## factor(Genre1)biography     1.227745  0.165898  7.401 1.36e-13 ***  
## factor(Genre1)comedy        0.291360  0.131789  2.211 0.027050 *  
## factor(Genre1)crime         0.468941  0.159639  2.938 0.003309 **  
## factor(Genre1)documentary   2.087394  0.202435 10.311 < 2e-16 ***  
## factor(Genre1)drama         0.893137  0.109212  8.178 2.89e-16 ***  
## factor(Genre1)fantasy       0.515180  0.170170  3.027 0.002466 **  
## factor(Genre1)history       0.762314  0.201209  3.789 0.000151 ***  
## factor(Genre1)horror         0.411219  0.124107  3.313 0.000922 ***  
## factor(Genre1)kidsandfamily 0.862609  0.316992  2.721 0.006504 **
```

```

## factor(Genre1)music          1.669686  0.398386  4.191 2.78e-05 ***
## factor(Genre1)musical        0.388028  0.422976  0.917 0.358946
## factor(Genre1)mysteryandthriller 0.100133  0.141268  0.709 0.478438
## factor(Genre1)other          -0.027272  0.320738 -0.085 0.932238
## factor(Genre1)romance         0.001411  0.166973  0.008 0.993260
## factor(Genre1)scifi           0.378333  0.156819  2.413 0.015841 *
## factor(Genre1)sportsandfitness 12.327678 324.743726  0.038 0.969719
## factor(Genre1)war              0.273576  0.574731  0.476 0.634070
## factor(Genre1)western          0.614705  0.484949  1.268 0.204952
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 8932.1  on 6448  degrees of freedom
## Residual deviance: 8188.1  on 6408  degrees of freedom
## AIC: 8270.1
##
## Number of Fisher Scoring iterations: 11

```

Finalmente, se predicen las etiquetas del conjunto de datos y se obtiene la matriz de confusión y la precisión del modelo.

```

#Predicción de valores
model.pred <- data.frame(values=predict(model.log,as.factor(Fresh),
                                         type = "response"))
model.pred$predicted <- factor(ifelse(model.pred$values>0.5, "Yes", "No"))
conf_matrix <- confusionMatrix(model.pred$predicted,(as.factor(Fresh)))

```

Regresión lineal múltiple

A continuación, se crea un modelo de regresión lineal múltiple, con regresores tanto cuantitativos como cualitativos, para tratar de predecir el valor de la etiqueta Audience.score de las distintas películas.

```

model.lin=lm(Audience.score~Runtime + factor(Parental.Control) +
             factor(Release.Year) + factor(Genre1) +
             factor(DirectorIsWriter) + factor(Release.isWide),data=df2)
(model.lin)

##
## Call:
## lm(formula = Audience.score ~ Runtime + factor(Parental.Control) +
##     factor(Release.Year) + factor(Genre1) + factor(DirectorIsWriter) +
##     factor(Release.isWide), data = df2)
##
## Coefficients:
##                               (Intercept)                    Runtime
##                               24.5006                     0.2584
## factor(Parental.Control)Yes   factor(Release.Year)2007    0.5168
##                               2.7378                     -4.7400
## factor(Release.Year)2008   factor(Release.Year)2009   -6.8774
##                               -4.7800                    factor(Release.Year)2011
##                               -4.7800                     -5.0284
## factor(Release.Year)2012   factor(Release.Year)2013   -5.2857
##                               -5.1284

```

```

##           factor(Release.Year)2014          factor(Release.Year)2015
##                               -6.5116                  -7.4913
##           factor(Release.Year)2016          factor(Release.Year)2017
##                               -4.4588                  -4.4785
##           factor(Release.Year)2018          factor(Release.Year)2019
##                               -1.7084                  2.9276
##           factor(Release.Year)2020         factor(Genre1)adventure
##                               3.7024                  8.5274
##           factor(Genre1)animation        factor(Genre1)anime
##                               17.0126                 28.9468
##           factor(Genre1)biography       factor(Genre1)comedy
##                               12.8847                  2.9121
##           factor(Genre1)crime          factor(Genre1)documentary
##                               3.8367                  25.3813
##           factor(Genre1)drama          factor(Genre1)fantasy
##                               10.4174                  6.9297
##           factor(Genre1)history        factor(Genre1)horror
##                               11.1597                  -5.8987
##           factor(Genre1)kidsandfamily   factor(Genre1)music
##                               13.7398                  18.9352
##           factor(Genre1)musical        factor(Genre1)mysteryandthriller
##                               9.7596                  -2.0931
##           factor(Genre1)other          factor(Genre1)romance
##                               1.2515                  3.2009
##           factor(Genre1)scifi          factor(Genre1)sportsandfitness
##                               1.9660                  9.6219
##           factor(Genre1)war            factor(Genre1)western
##                               11.0134                  7.9654
##           factor(DirectorIsWriter)Yes factor(Release.isWide)No
##                               1.3826                  -0.9466
##           factor(Release.isWide)Yes
##                               5.8787

```

A continuación se obtienen los residuos del modelo, que serán utilizados para representar la distribución de errores y su similitud con una distribución normal.

```

residuos <- rstandard(model.lin)
valores_ajustados <- fitted(model.lin)

```

Contraste de Hipótesis.

Para finalizar el estudio sobre este dataset, se realizarán dos contraste de hipótesis:

Contraste 1: Las películas cuyo director es también el escritor son mejores (tienen mayor Tomatometer) que las que no cumplen esta condición Lo primero será separar las dos muestras, por un lado las películas cuyo director es también el escritor y las que no y las graficamos para observar su distribución:

```

# Separamos las muestras y nos quedamos solo con la variable Tomatometer
director.and.writer <- df[df$DirectorIsWriter == "Yes",]
just.director <- df[df$DirectorIsWriter == "No",]
# Nos quedamos solo con las variables DirectorIsWriter y Tomatometer
director.and.writer <- subset(director.and.writer, select = c(DirectorIsWriter,
                                                               Tomatometer))
just.director <- subset(just.director, select = c(DirectorIsWriter,

```

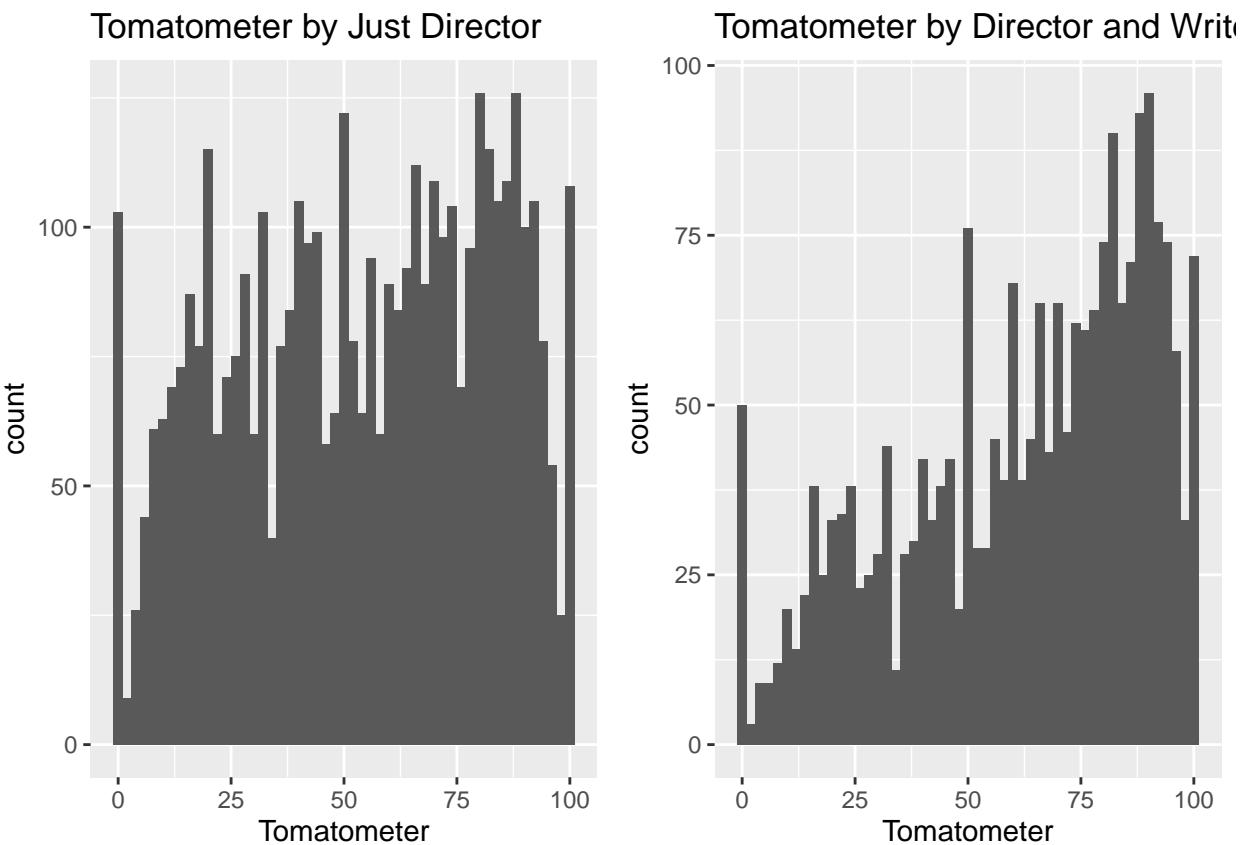
```

    Tomatometer))

# Quitamos los valores NA
director.and.writer <- director.and.writer[!is.na(director.and.writer),]
just.director <- just.director[!is.na(just.director),]

# Graficamos ambas variables para ver su distribución
plot1 <- ggplot(just.director, aes(Tomatometer)) + geom_histogram(binwidth=2) +
  ggtitle("Tomatometer by Just Director")
plot2 <- ggplot(director.and.writer, aes(Tomatometer)) +
  geom_histogram(binwidth=2) +
  ggtitle("Tomatometer by Director and Writer")
ggarrange(plot1,plot2)

```



A simple vista, en el gráfico de Tomatometer by Director and Writer, parece existir una tendencia a mejorar la puntuación cuando el director es también el escritor.

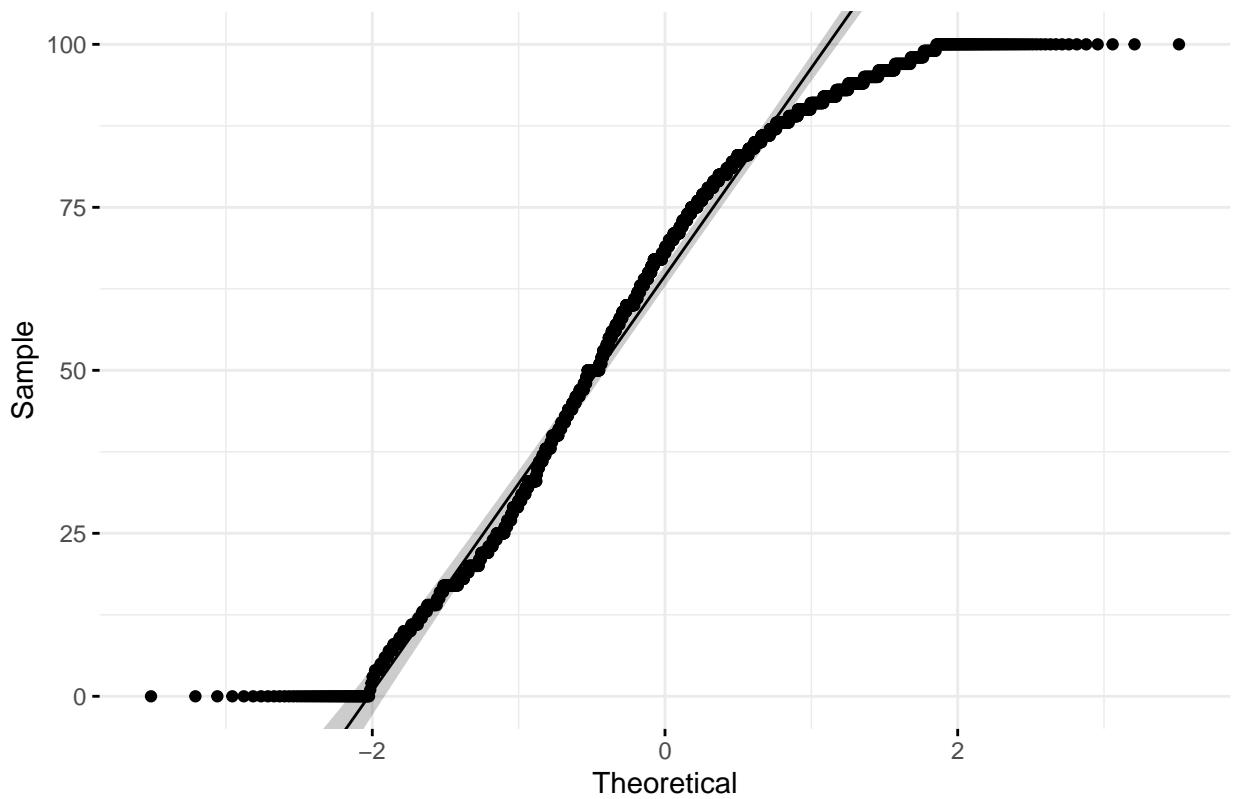
Se comprobará el supuesto de normalidad de ambas variables, para ello se realizará un test de Shapiro-Wilck y un grafico QQ:

```

ggqqplot(director.and.writer, x="Tomatometer", add = "qqline",
          ggtheme = theme_minimal(),
          title = "QQplot Tomatometer by Director and Writer", ylim = c(0,100) )

```

QQplot Tomatometer by Director and Writer

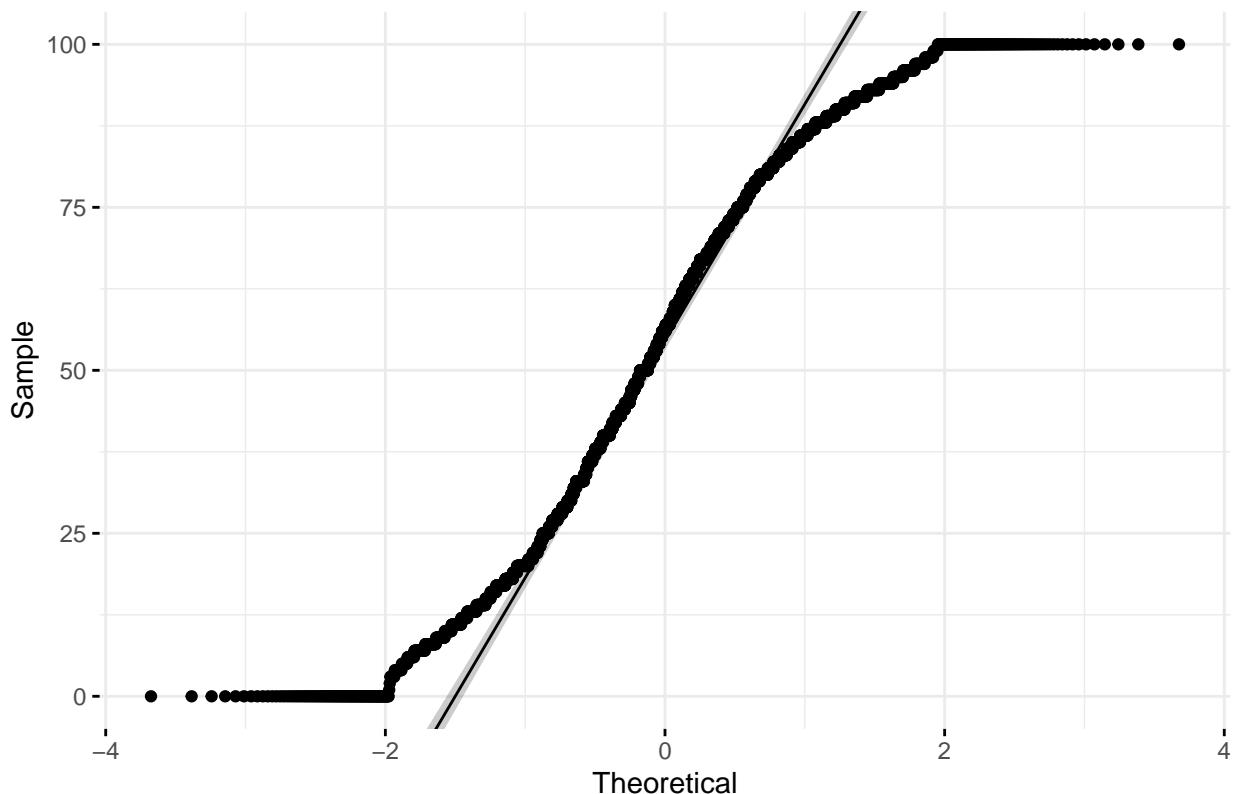


```
shapiro.test(director.and.writer$Tomatometer)

##
## Shapiro-Wilk normality test
##
## data: director.and.writer$Tomatometer
## W = 0.93497, p-value < 2.2e-16

ggqqplot(just.director, x="Tomatometer", add = "qqline",
          ggtheme = theme_minimal(),
          title = "QQplot Tomatometer by Just Director", ylim = c(0,100) )
```

QQplot Tomatometer by Just Director



```
shapiro.test(just.director$Tomatometer)
```

```
##
## Shapiro-Wilk normality test
##
## data: just.director$Tomatometer
## W = 0.95663, p-value < 2.2e-16
```

En ambos casos el p-valor del test de Shapiro es menor a 0.05, por lo que no se puede suponer normalidad.

Consideraciones:

Se realizarán las siguientes consideraciones:

- Ambas muestras no distribuyen normalmente, pero al ser lo suficientemente grandes, por el teorema del límite central, se puede afirmar que la media de cada muestra se acerca a la media poblacional.

- Cada muestra, pertenece a una población distinta e independiente.

- Se desconoce la varianza de las poblaciones, pero se suponen iguales.

Así, se plantearan las siguientes Hipótesis Nula: **La media de la puntuación Tomatometer de las películas cuyo director es también el escritor, es IGUAL al de las películas donde esto no sucede**

Esta hipótesis se traduce en:

H₀: Xm_director.and.writer - Xm_just.director = 0 H₁: Xm_director.and.writer - Xm_just.director > 0

```
t.test(director.and.writer$Tomatometer, just.director$Tomatometer,
       alternative = "greater")
```

```

## Welch Two Sample t-test
##
## data: director.and.writer$Tomatometer and just.director$Tomatometer
## t = 11.683, df = 4723.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
## 7.204935      Inf
## sample estimates:
## mean of x mean of y
## 62.69022 54.30436

```

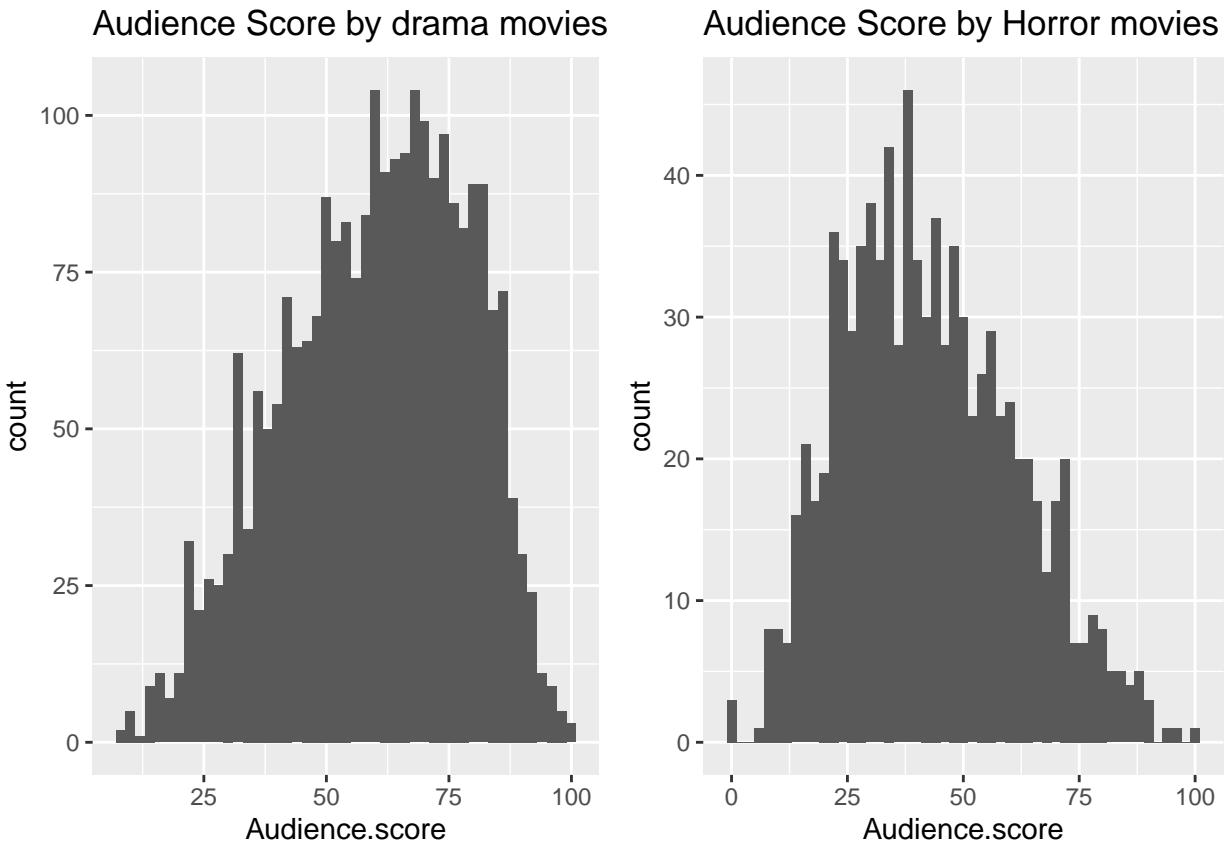
Como se observa, el p-valor obtenido es menor a 0.05, por lo que se rechaza la hipótesis nula de que las medias de la puntuación Tomatometer son iguales, por lo que se acepta la hipótesis alternativa de que **La media de las puntuaciones Tomatometerde las películas cuyo director es también el escritor es mayor que la media de las películas que no cumplen esta condición**

Contraste 2: Las películas de categoría Drama tienen mejor Audience.score que las películas de categoría Horror. De nuevo, lo primero que haremos será obtener del dataset las dos muestras que deseamos y graficarlas para ver sus distribuciones:

```

# Obtenemos las películas pertenecientes a cada categoría
drama.movies <- df[df$Genre1=="drama",]
horror.movies <- df[df$Genre1 == "horror",]
## Nos quedamos solo con las variables DirectorIsWriter y Tomatometer
drama.movies <- subset(drama.movies, select = c(Genre1, Audience.score))
horror.movies <- subset(horror.movies, select = c(Genre1, Audience.score))
# Graficamos ambas variables para ver su distribución
plot1 <- ggplot(drama.movies, aes(Audience.score)) + geom_histogram(binwidth=2) +
  ggtitle("Audience Score by drama movies")
plot2 <- ggplot(horror.movies, aes(Audience.score)) +
  geom_histogram(binwidth=2) +
  ggtitle("Audience Score by Horror movies")
ggarrange(plot1,plot2)

```



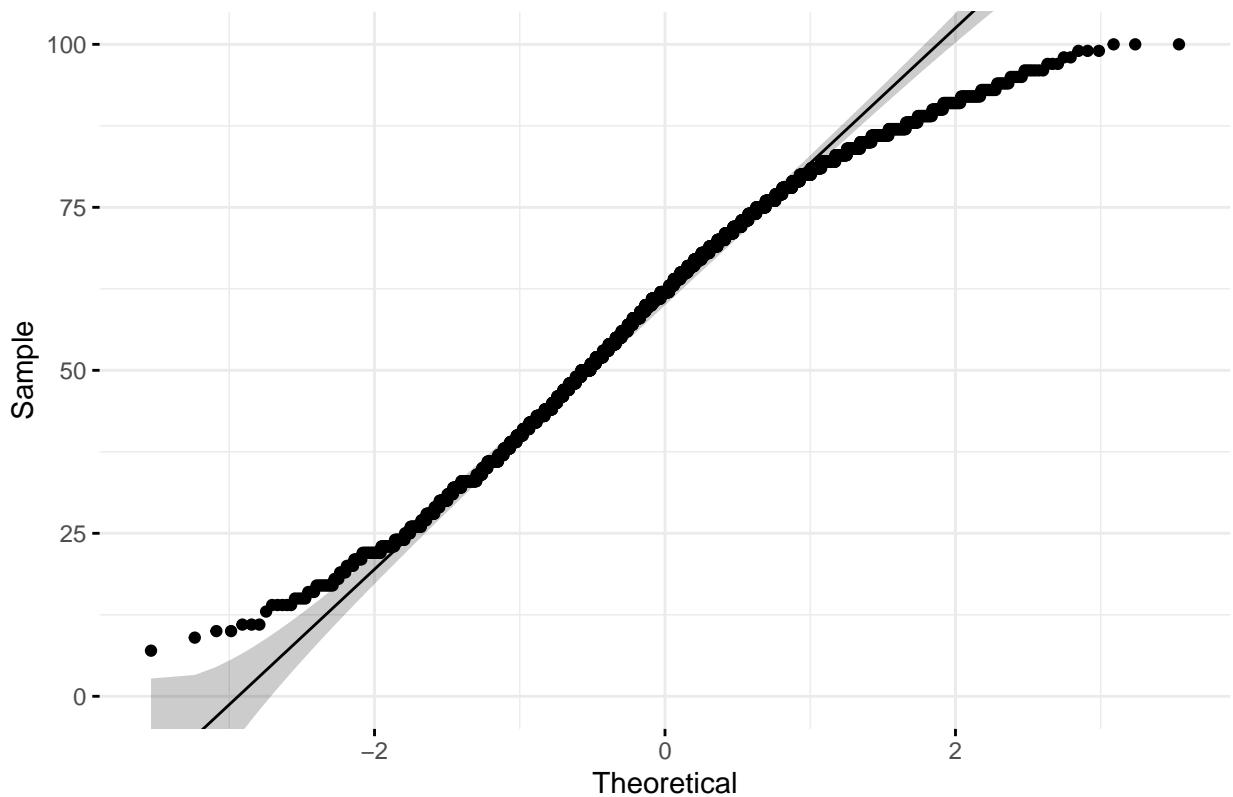
Por observación de las gráficas, se observa que:

- La gráfica de Audience Score by drama movies, presenta una cierta asimetría hacia la izquierda, aunque su distribución se parece bastante a la distribución normal.
- La gráfica de Audience Score by Horror movies, presenta a su vez un asimetría hacia la derecha; pero de igual forma parece tener una distribución que tiende a ser normal.

Pasaremos ahora a comprobar el supuesto de normalidad. Para esto, realizaremos nuevamente un gráfico QQ y un test de Shapiro-Wilk para cada muestra

```
ggqqplot(drama.movies, x="Audience.score", add = "qqline",
          ggtheme = theme_minimal(),
          title = "QQplot Audience Score by drama movie", ylim = c(0,100) )
```

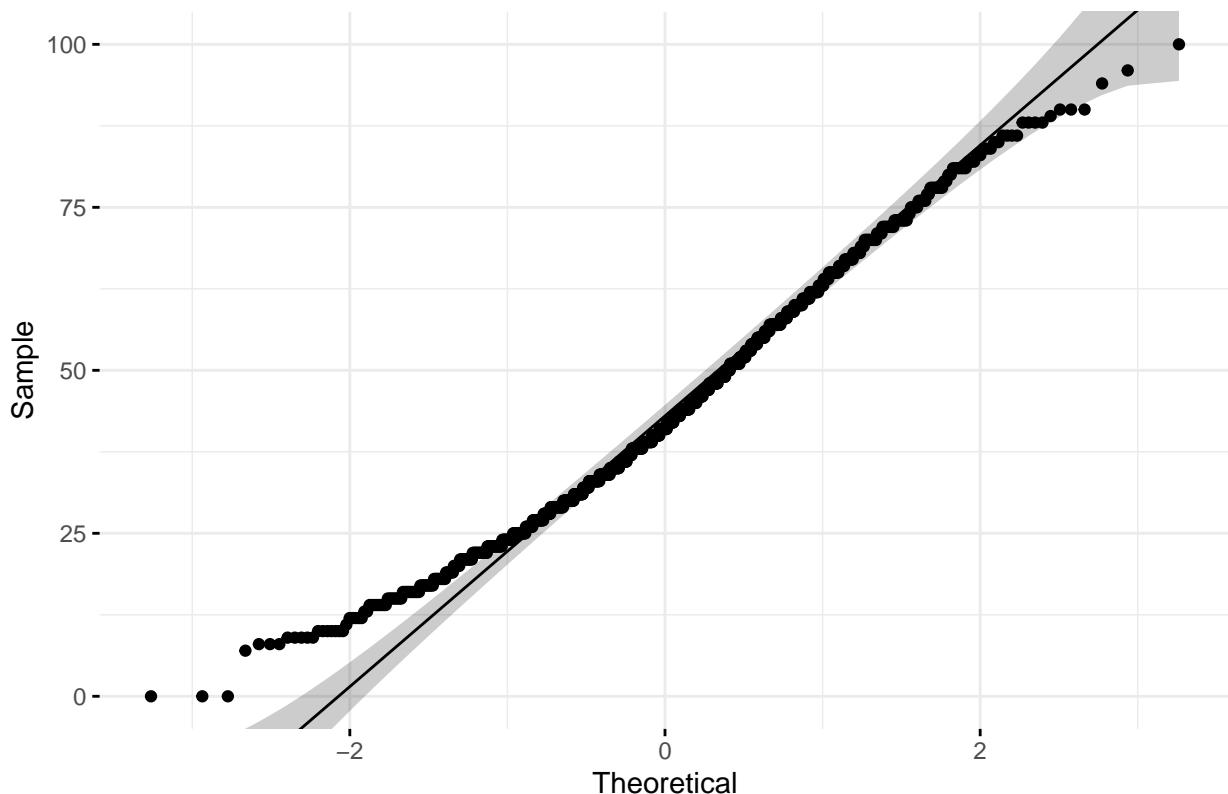
QQplot Audience Score by drama movie



```
shapiro.test(drama.movies$Audience.score)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: drama.movies$Audience.score  
## W = 0.98083, p-value < 2.2e-16  
ggqqplot(horror.movies, x="Audience.score", add = "qqline",  
         ggtheme = theme_minimal(),  
         title = "QQplot Audience Score by drama movie", ylim = c(0,100) )
```

QQplot Audience Score by drama movie



```
shapiro.test(horror.movies$Audience.score)
```

```
##
## Shapiro-Wilk normality test
##
## data: horror.movies$Audience.score
## W = 0.98464, p-value = 3.92e-08
```

Se observa que en ambas no es posible aceptar el supuesto de normalidad. Pero nuevamente, a partir del teorema del límite central y debido a que ambas muestras son mayores a 30, se puede afirmar que la media de cada muestra se acerca a la media poblacional.

Así, se planteará la siguiente Hipótesis Nula: La media de la puntuación Audience.score de las películas cuya categoría es drama es IGUAL al de las películas cuya categoría es terror.

Lo anterior se traduce en:

$H_0: \bar{X}_{\text{drama.movies}} - \bar{X}_{\text{horror.movies}} = 0$ $H_1: \bar{X}_{\text{drama.movies}} - \bar{X}_{\text{horror.movies}} > 0$

```
t.test(drama.movies$Audience.score, horror.movies$Audience.score,
        alternative = "greater")
```

```
##
## Welch Two Sample t-test
##
## data: drama.movies$Audience.score and horror.movies$Audience.score
## t = 23.811, df = 1597.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
```

```

## 95 percent confidence interval:
## 16.05114      Inf
## sample estimates:
## mean of x mean of y
## 60.49438 43.25138

```

Del contraste de hipótesis se observa que el p-valor es menor a 0.05, por lo que se rechazaría la hipótesis nula de que ambas medias son iguales.

Por tal motivo, se acepta la hipótesis alternativa de que **La media de la puntuación Audience.score de las películas de drama es mayor que la media de dicha puntuación en las películas de horror**

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

Correlación

En el análisis de correlaciones, todos los gráficos y tablas se han presentado en el apartado 4.3, ya que su aplicación consiste básicamente en mostrar y interpretar los gráficos, y estas interpretaciones eran necesarias para elegir las variables en las siguientes pruebas.

Clasificación con regresión logística

La siguiente tabla muestra la matriz de confusión del modelo de regresión logística, que representa los falsos positivos, los falsos negativos, los verdaderos negativos y los verdaderos positivos al aplicar el modelo sobre el total del conjunto.

```

#Matriz de confusión
conf_matrix$table

```

```

##             Reference
## Prediction   No  Yes
##           No 1905 1154
##           Yes 1205 2185

```

Finalmente se muestra la precisión del modelo, obtenida a partir de la matriz de confusión anterior, mediante la fórmula $(tp+tn)/(tp+tn+fp+fn)$, o lo que es lo mismo, número de aciertos sobre el total de datos.

```

#Precisión del modelo
conf_matrix$overall

```

```

##          Accuracy       Kappa AccuracyLower AccuracyUpper AccuracyNull
## 6.342069e-01 2.670779e-01 6.223154e-01 6.459768e-01 5.177547e-01
## AccuracyPValue McnemarPValue
## 1.702939e-79 3.032674e-01

```

Regresión lineal múltiple

El siguiente fragmento muestra un resumen del modelo

```
summary(model.lin)
```

```

##
## Call:
## lm(formula = Audience.score ~ Runtime + factor(Parental.Control) +
##     factor(Release.Year) + factor(Genre1) + factor(DirectorIsWriter) +
##     factor(Release.isWide), data = df2)
##
## Residuals:

```

```

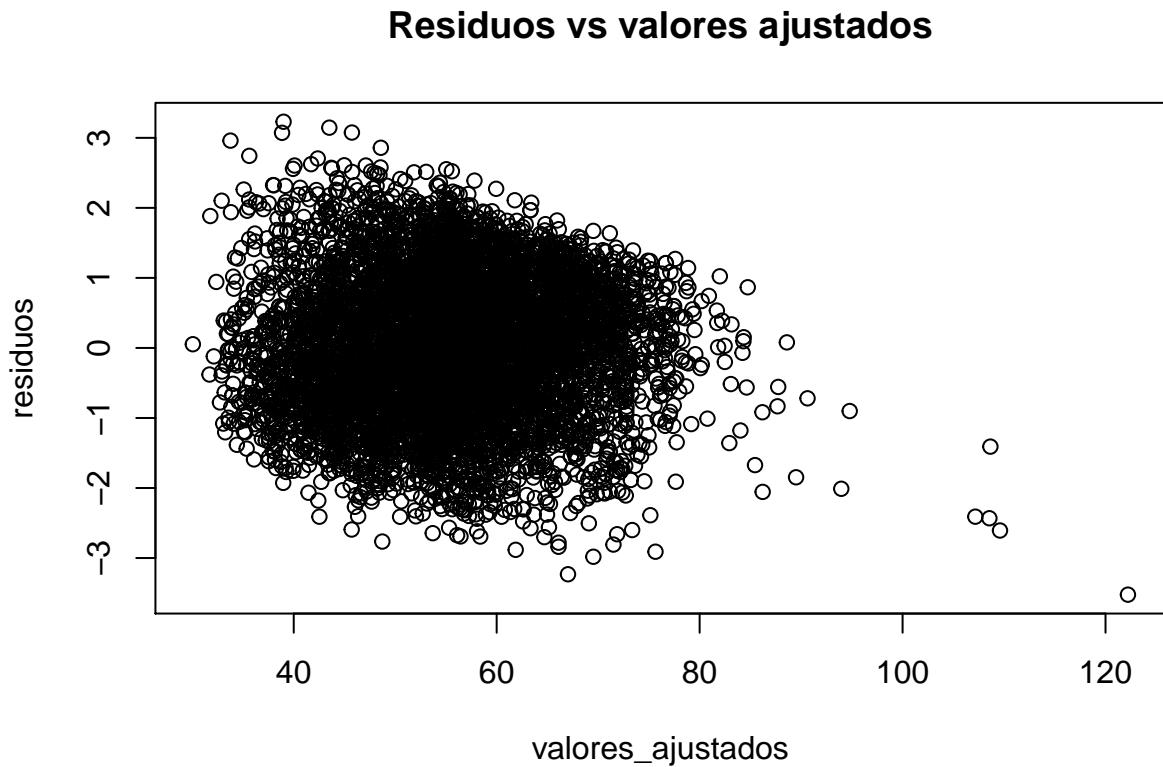
##      Min     1Q   Median     3Q    Max
## -61.210 -12.796    0.006  12.871  56.999
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                24.50057  17.75396  1.380  0.16763
## Runtime                     0.25837   0.01406 18.371 < 2e-16 ***
## factor(Parental.Control)Yes 2.73777   0.54035  5.067 4.16e-07 ***
## factor(Release.Year)2007    0.51680  17.90733  0.029  0.97698
## factor(Release.Year)2008   -4.74002  17.73208 -0.267  0.78924
## factor(Release.Year)2009   -6.87744  17.71282 -0.388  0.69783
## factor(Release.Year)2010   -4.78004  17.70945 -0.270  0.78723
## factor(Release.Year)2011   -5.02842  17.70460 -0.284  0.77641
## factor(Release.Year)2012   -5.12840  17.70280 -0.290  0.77206
## factor(Release.Year)2013   -5.28568  17.70183 -0.299  0.76526
## factor(Release.Year)2014   -6.51163  17.69940 -0.368  0.71296
## factor(Release.Year)2015   -7.49126  17.69895 -0.423  0.67212
## factor(Release.Year)2016   -4.45882  17.69395 -0.252  0.80105
## factor(Release.Year)2017   -4.47845  17.69607 -0.253  0.80022
## factor(Release.Year)2018   -1.70843  17.69829 -0.097  0.92310
## factor(Release.Year)2019    2.92759  17.70144  0.165  0.86864
## factor(Release.Year)2020    3.70239  17.71664  0.209  0.83447
## factor(Genre1)adventure    8.52737  1.21191  7.036 2.18e-12 ***
## factor(Genre1)animation   17.01260  2.48194  6.855 7.82e-12 ***
## factor(Genre1)anime       28.94678  8.88085  3.259  0.00112 **
## factor(Genre1)biography   12.88466  1.34854  9.555 < 2e-16 ***
## factor(Genre1)comedy      2.91207  1.09343  2.663  0.00776 **
## factor(Genre1)crime       3.83675  1.34759  2.847  0.00443 **
## factor(Genre1)documentary 25.38126  1.40087 18.118 < 2e-16 ***
## factor(Genre1)drama       10.41736  0.90226 11.546 < 2e-16 ***
## factor(Genre1)fantasy    6.92975  1.43742  4.821 1.46e-06 ***
## factor(Genre1)history     11.15970  1.69192  6.596 4.57e-11 ***
## factor(Genre1)horror      -5.89868  1.03043 -5.724 1.08e-08 ***
## factor(Genre1)kidsandfamily 13.73982  2.73352  5.026 5.13e-07 ***
## factor(Genre1)music        18.93515  2.94403  6.432 1.35e-10 ***
## factor(Genre1)musical     9.75959  3.62728  2.691  0.00715 **
## factor(Genre1)mysteryandthriller -2.09314  1.17450 -1.782  0.07477 .
## factor(Genre1)other       1.25152  2.60006  0.481  0.63029
## factor(Genre1)romance     3.20088  1.36795  2.340  0.01932 *
## factor(Genre1)scifi       1.96595  1.31728  1.492  0.13563
## factor(Genre1)sportsandfitness 9.62187  17.71264  0.543  0.58700
## factor(Genre1)war         11.01344  4.97469  2.214  0.02687 *
## factor(Genre1)western     7.96537  4.13900  1.924  0.05434 .
## factor(DirectorIsWriter)Yes 1.38258  0.48159  2.871  0.00411 **
## factor(Release.isWide)No   -0.94658  0.62456 -1.516  0.12967
## factor(Release.isWide)Yes   5.87874  0.77348  7.600 3.38e-14 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.68 on 6408 degrees of freedom
## Multiple R-squared:  0.2309, Adjusted R-squared:  0.2261
## F-statistic:  48.1 on 40 and 6408 DF,  p-value: < 2.2e-16

```

Los siguientes gráficos muestran; en primer lugar, la distribución de errores y, en segundo lugar, su similitud

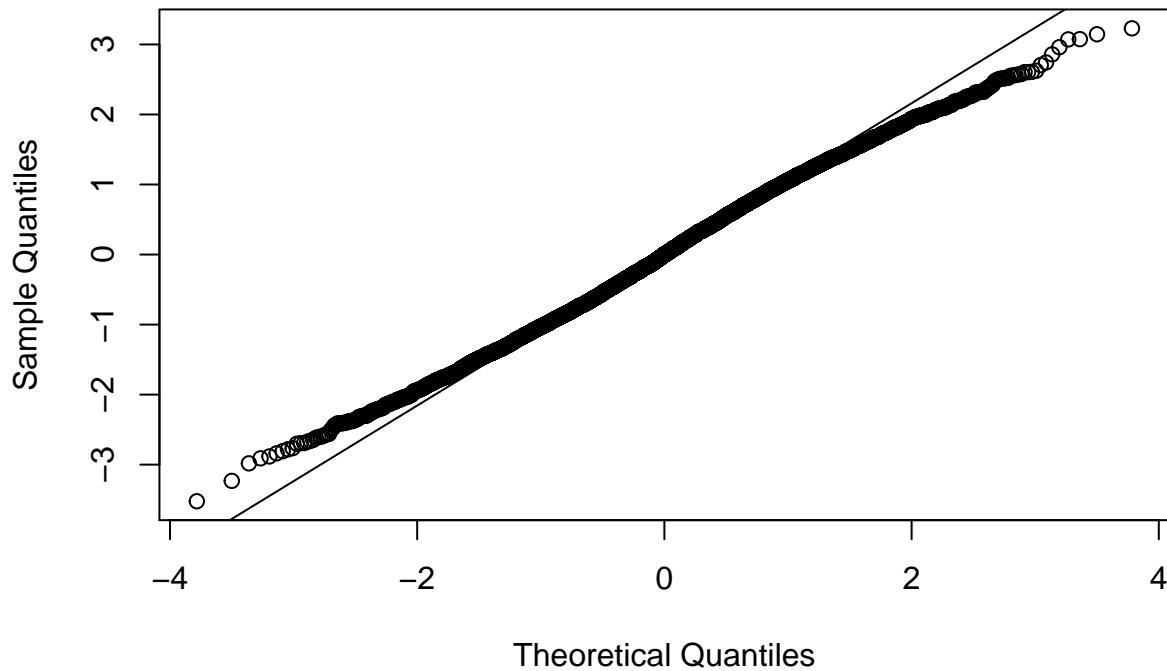
con una distribución normal.

```
plot(valores_ajustados, residuos, main="Residuos vs valores ajustados")
```



```
qqnorm(residuos, main="QQplot normalidad residuos")
qqline(residuos)
```

QQplot normalidad residuos



Contraste de Hipótesis

En el apartado 4.3, en la sección correspondiente al desarrollo de los dos contrastes de hipótesis, se encuentran todos los gráficos y tablas correspondientes a los mismos, ya que dicho estudio se apoya necesariamente en estos.

6 Resolución del problema

Correlación

Las conclusiones a cada gráfico se han comentado en el apartado 4.3. En este apartado, se muestra un resumen de esas conclusiones.

- A mayor tiempo de duración, las películas tienden a tener una valoración ligeramente mejor por parte de la audiencia.
- Los documentales y biografías suelen ser recibidas positivamente por la crítica, mientras que en los géneros de comedia y acción sucede lo contrario.
- Las películas con restricciones de edad tienden a tener una mejor valoración que las que son para todos los públicos.
- La estación en que se ha lanzado una película no influye en su puntuación.
- La opinión de la crítica mejora paulatinamente con el paso de los años.
- Las películas que se estrenan en todos los cines suelen tener una peor crítica que las limitadas.

- Las películas dirigidas por el propio escritor tienden a tener una mejor crítica que las que tienen directores y escritores diferentes.
- Las variables Genre2 y Genre3 se han descartado por tener demasiados valores perdidos. Director y Production no se utilizaron por tener una cantidad de categorías demasiado grande. Rating y Release.Date no se usaron porque en su lugar se usaron Parental.Control y Release.Year. Release.Season no se utilizó por no tener ningún efecto apreciable en las puntuaciones de las películas.
- Además de las etiquetas Tomatometer, Audience.score y Fresh, las variables que se usaron en los modelos de clasificación o regresión son: **Parental.Control**, **DirectorIsWriter**, **Runtime**, **Genre1**, **Release.Year** y **Release.isWide**.

Clasificación con regresión logística

- Se obtiene un AIC de 8250,1.
- El modelo clasifica correctamente 4066 de los 6431 registros del conjunto de datos, obteniendo 1157 falsos negativos y 1208 falsos positivos.
- La precisión del modelo es del 63,23%, lo cual se podría catalogar como aceptable.
- El porcentaje es claramente superior al 50%, pero no lo suficiente como para que sea posible clasificar con garantías si una película tendrá una crítica favorable o no a partir de su duración, género, año y tipo de estreno, presencia o de restricciones de edad y coincidencia del director con el escritor. Por tanto, el modelo no permite responder el problema con grandes resultados, pero sí clasificar correctamente el 63,2% de películas.

Regresión lineal múltiple

- El gráfico de residuos frente valores ajustados, muestra un patrón aleatorio de los residuos, mientras que el QQ Plot indica que la mayoría de datos siguen la distribución normal, aunque en los extremos se desvían.
- El coeficiente de determinación ajustado obtenido con el modelo es de 0.227, lo cual se podría calificar como un ajuste bastante pobre.
- El modelo permite obtener una ligera aproximación de la puntuación de una película, pero no es suficiente para concluir en que haya bondad en el ajuste, por lo que los resultados no permiten resolver la cuestión de cuál será la puntuación de una película a partir de su duración, género, año y tipo de estreno, presencia o de restricciones de edad y coincidencia del director con el escritor.

Contraste de Hipótesis.

- **Contraste 1:**

- Se comprobó que las dos muestras obtenidas no presentaban una distribución normal. Aún así se pudo realizar el contraste de hipótesis debido al tamaño de las muestras y al hecho de que comparamos sus medias.
- Se rechazó la hipótesis nula de que la media muestral de la puntuación **Tomatometer** de las películas cuyo director es también el escritor era igual a la media muestral de las películas que no cumplían con ese requisito.
- Se comprobó, a través de este contraste, que si existe una tendencia positiva a estar mejor valorada una película cuando su director es también el escritor.

- **Contraste 2:**

- De nuevo se comprobó que las dos muestras obtenidas no presentaban una distribución normal. Aún así se pudo realizar el contraste de hipótesis debido al tamaño de las muestras y al hecho de que comparamos sus medias.

- Se rechazó la hipótesis nula de que la media muestral de la puntuación Audience.score de las películas cuya categoría es drama era igual a la media muestral de las películas de categoría horror.
- Se aceptó la hipótesis nula de estas medias muestrales no eran iguales y que de hecho, las películas de drama estan mejor puntuadas por el público que las películas de horror.

Generación de CSV resultante

```
write.csv(df, '.\\recogiendo_tomates_final.csv', row.names=FALSE)
```

Firma

Contribuciones	Firma
Investigación previa	D.B.M, R.G.H
Redacción de respuestas	D.B.M, R.G.H
Desarrollo de código	D.B.M, R.G.H