Trabajo Práctico N° 3

Barraza, Veronica y Maldonado, Kevin

Julio 2022

Índice

Planteo del problema

En este trabajo práctico se utilizará un dataset que contiene 4000 títulos de una plataforma de streaming. El archivo credits_train contiene los actores y directores para estas peículas y series. El objetivo de trabajo es **predecir la calificación de IMDB para las películas** a partir de otras covariables para cada título. Consideraremos la pérdida cuadrática como forma de evaluar modelos.

Análisis exploratorio de datos (EDA)

En esta sección vamos a realizar una exploración del dataset. (a) ¿Hay algún género que parezca estar más asociado con el puntaje del título? (b) ¿Cómo fue evolucionando este puntaje a lo largo del tiempo? (c) ¿Hay algún actor o director asociado con mayores o menores puntajes? (d) ¿Las películas más populares son las mejor puntuadas?

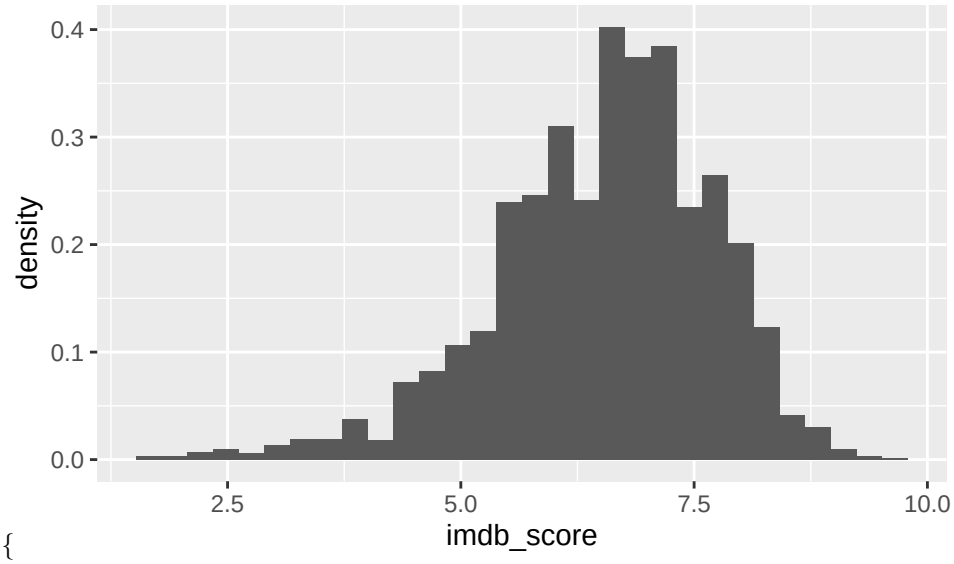
Antes de comenzar el análisis de EDA, tenemos que realizar una limpieza del dataset. Para esto eliminamos valores nulos, duplicados y dos columnas que presentaban un porcentaje muy alto de valores nulos.

Veamos cómo se distribuyen las películas y las series en el dataset.

Ahora podemos visualizar cómo es la distribución de la variable de interés imdb_score.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

\begin{figure}



{

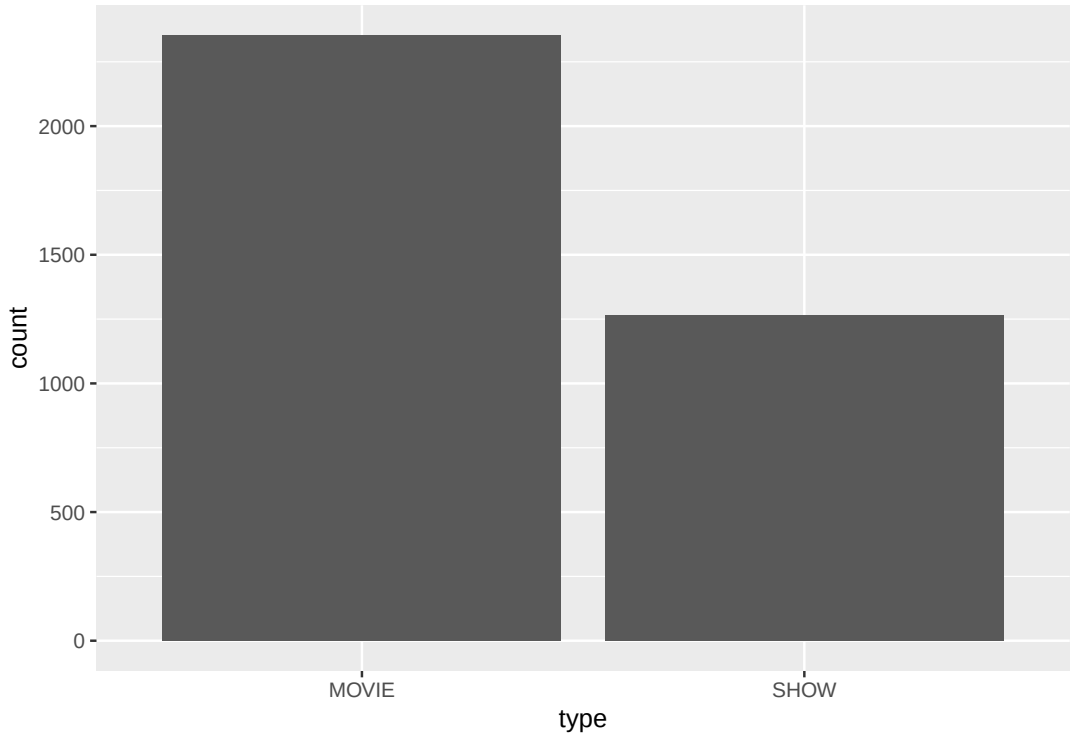


Figura 1: N° de observaciones con información de peliculas y series

}

\caption{ Distribución de IMDB_SCORE} \end{figure}

A continuación se muestra la distribución del score en función de los distintos géneros. A nivel general se observa que la mediana se encuentra cercana a 7, con rangos dinámicos que varian entre 3 a 8. Es de notar cierta diversidad en los scores, con las películas de horror teniendo el puntaje más bajo, y las películas de guerra e historia el más alto.

Finalmente, si vemos cómo evoluciona el score a lo largo de los años, se observa una tendencia a la baja en ambos, con las series apareciendo más tardiamente y con un score consistentemente mayor. La tendencia a la baja podría ser un sesgo del dataset: quizá en IMDB se ingresan todas las películas/series nuevas, pero solo las películas/series viejas de mejor calidad. La diferencia entre los tipos podría manifestar una diferente población de votantes: quizá tienen criterios distintos.

```
## `geom_smooth()` using formula 'y ~ x'
```

\begin{figure}

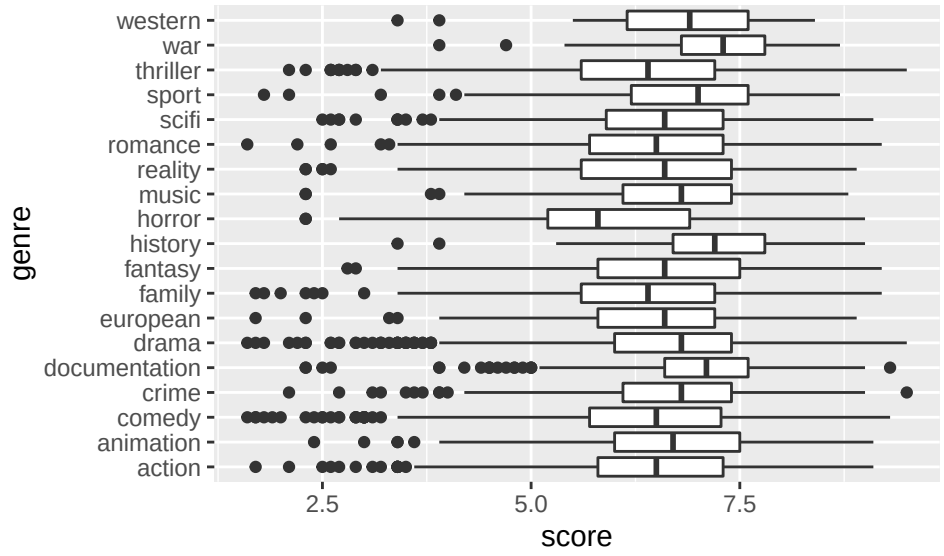
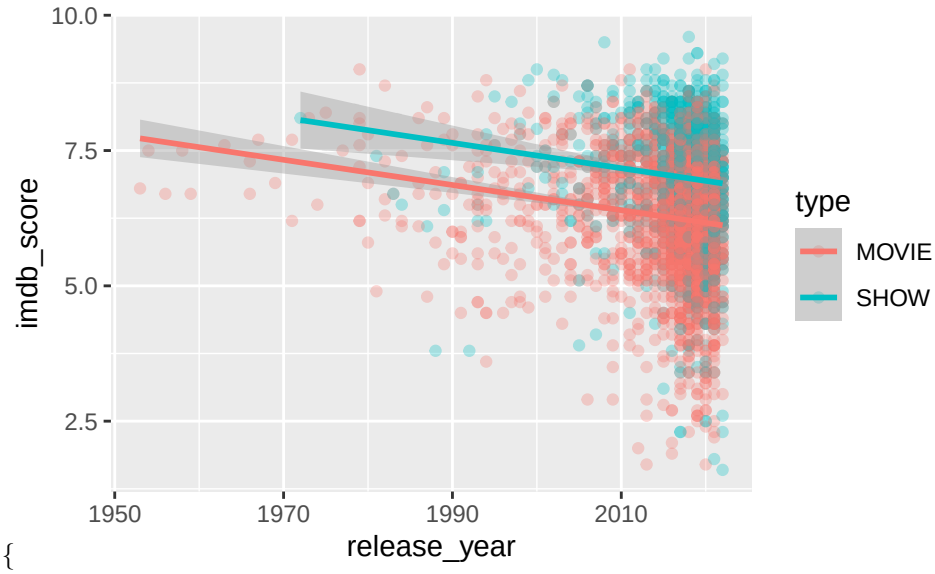


Figura 2: Boxplot: distribución del score en función de los distintos géneros



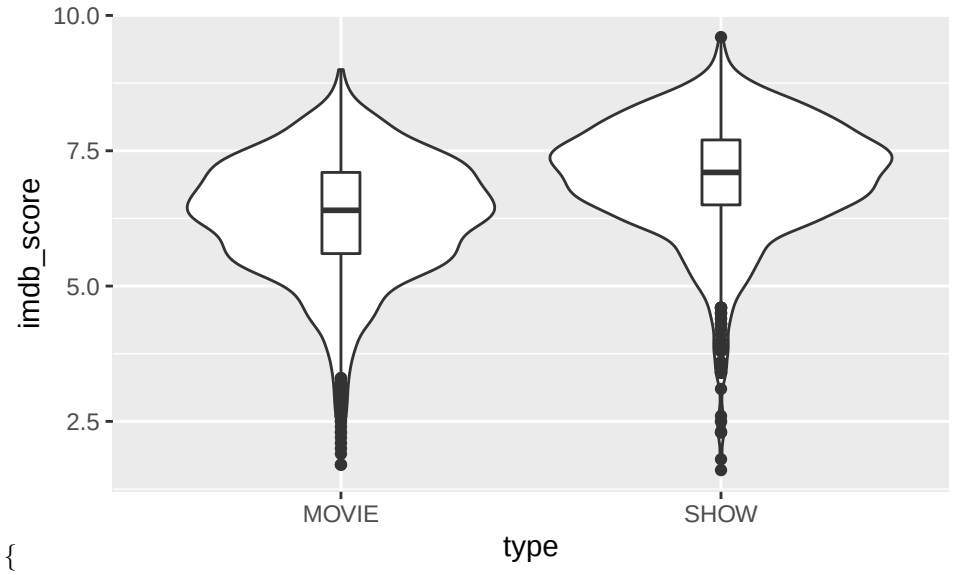
{

}

\caption{ IMDB_score versus año de estreno, en función del tipo (películas o serie)} \end{figure}

Observemos esta distribución entre tipos de manera global.

\begin{figure}



{

}

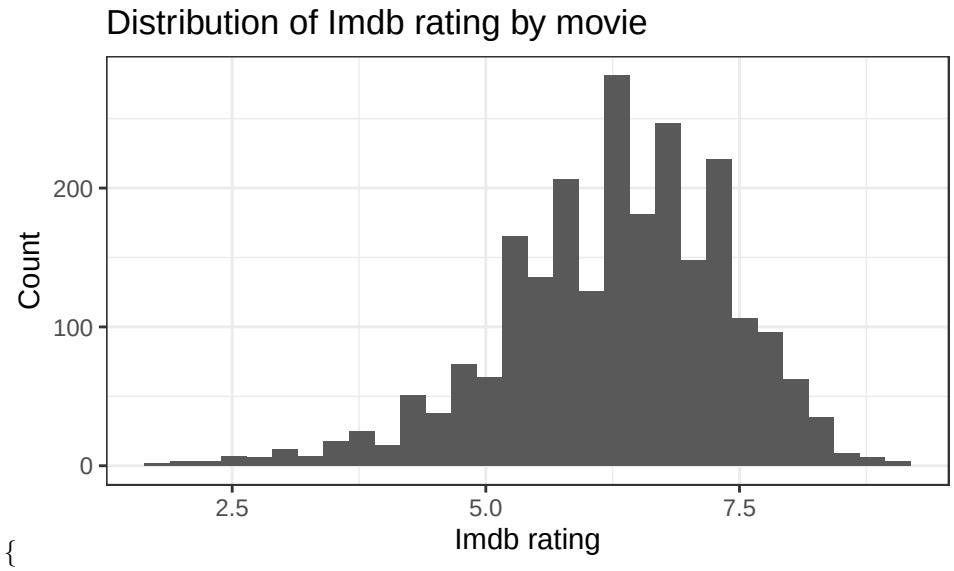
\caption{ Distribución de IMDB_SCORE para cada tipo} \end{figure}

Se puede más explicitamente la diferencia entre las medianas de los puntajes de ambos tipos.

Ahora podemos visualizar como es la distribución de la variable de interes imdb_score solamente para las observaciones relacionadas a las peliculas.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

\begin{figure}



{

}

\caption{ Distribución de IMDB_SCORE para las películas} \end{figure}

En la siguiente figura se observa la relación entre el score y el número de votantes. Es notable la relación entre ambos: pareciera ser que las películas de mayor calidad, con un score más alto, tienen muchos votos. Hay mucho más ruido para las películas con poca cantidad de votos.

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

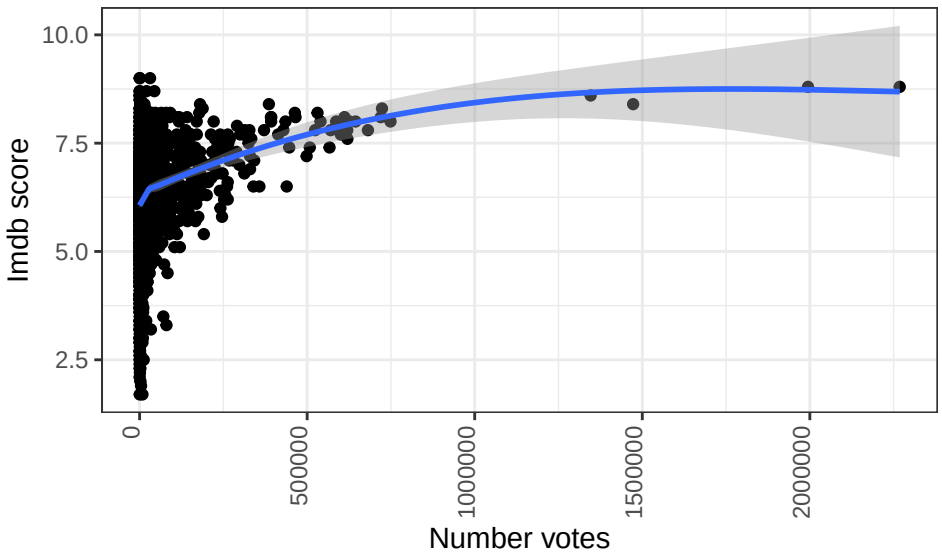
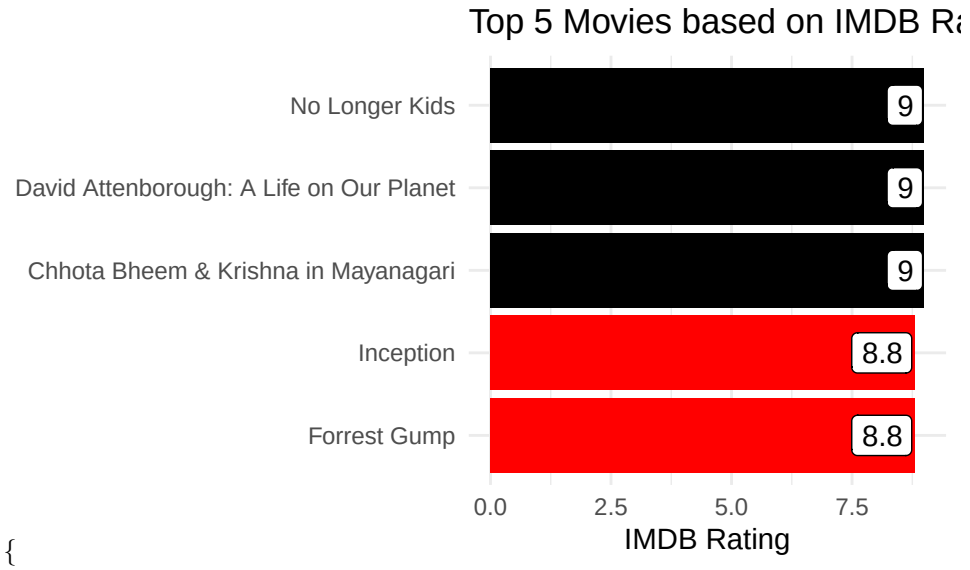


Figura 3: Relación entre IMDB score verus n°votantes

A continuación podemos ver las 5 películas que poseen el score más alto.

```
## # A tibble: 5 x 12
## # Groups:   imdb_id [5]
##      X1 id     title     type  description        release_year runtime genres
##   <dbl> <chr>  <chr>     <chr> <chr>                     <dbl>   <dbl> <chr>
## 1    18 tm765~ No Longe~ MOVIE "By coincidence, ~         1979     235 ['comedy~
## 2   600 tm166~ Chhota B~ MOVIE "Bheem and his Fr~         2011      66 ['animat~
## 3  2303 tm853~ David At~ MOVIE "The story of lif~         2020      83 ['docume~
## 4    68 tm122~ Forrest ~ MOVIE "A man with a low~         1994     142 ['drama'~
## 5   181 tm926~ Inception MOVIE "Cobb, a skilled ~         2010     148 ['scifi'~
## # ... with 4 more variables: production_countries <chr>, imdb_id <chr>,
## #   imdb_score <dbl>, imdb_votes <dbl>
```

\begin{figure}



{

}

\caption{ Top 5 peliculas basadas según el IMDB_SCORE} \end{figure}

Antes de analizar la relación entre el género de las películas y el score, vamos a visualizar la frecuencia de los géneros para este dataset. Los géneros más frecuentes son : drama, comedia, acción, romance y thriller. En el próximo gráfico, podemos ver los box-plot de los scores para los géneros más frecuentes. En general no pareciera haber una gran diferencia de score entre estos géneros más votados. Los géneros de documentales y crimen presentan una mediana mayor al resto y con un menor rango dinámico.

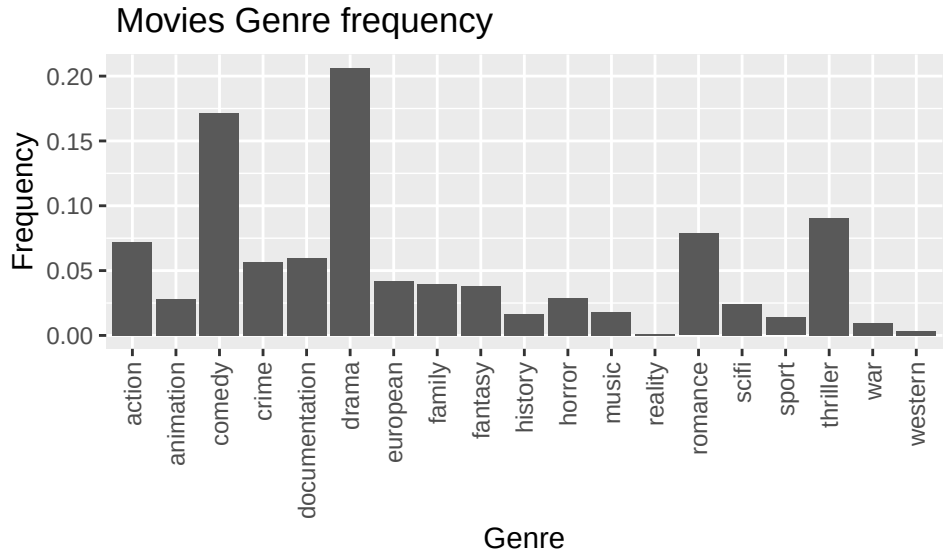
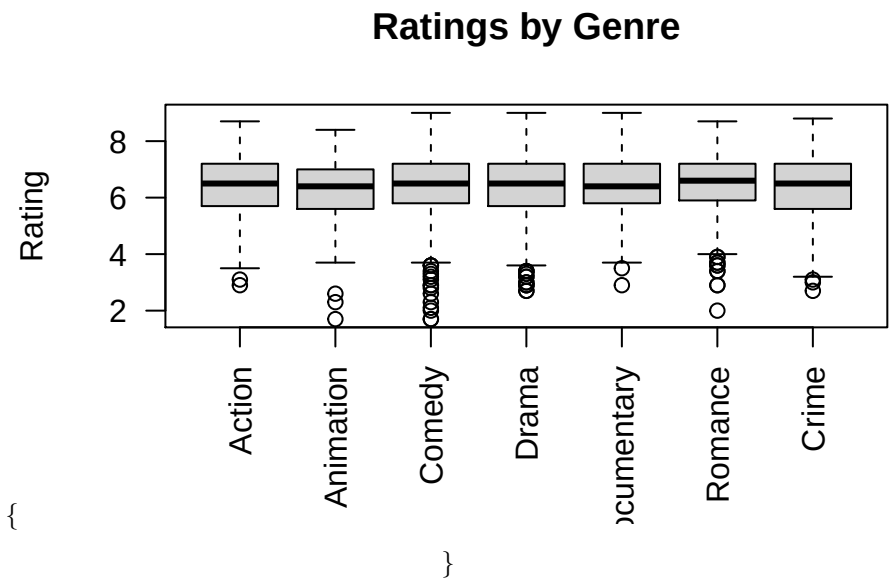


Figura 4: Frecuencia de peliculas con los diez generos más relevantes

‘boxplot(Action*rating*, *Animation*rating, Comedy*rating*, *Drama*rating, Documentary*rating*, *Romance*rating, Short$rating, names = c(“Action”, “Animation”, “Comedy”, “Drama”, “Documentary”, “Romance”, “Short”), main = “Ratings by Genre”, ylab = “Rating”)

\begin{figure}



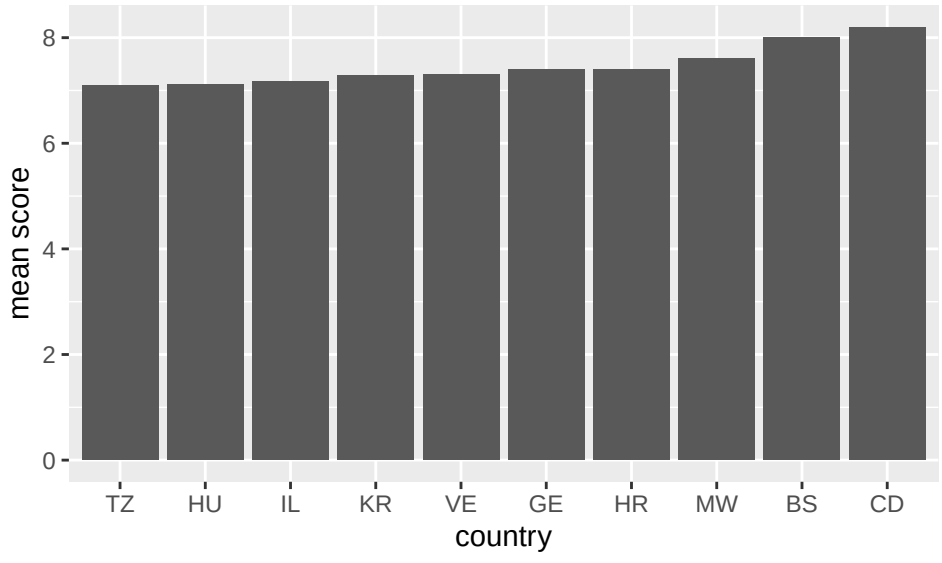
{

}

\caption{ Boxplot de IMDB_SCORE en función de los generos más relevantes} \end{figure}

En la siguiente gráfica podemos observar el score promedio para los países que produjeron las películas con mayor score.

\begin{figure}



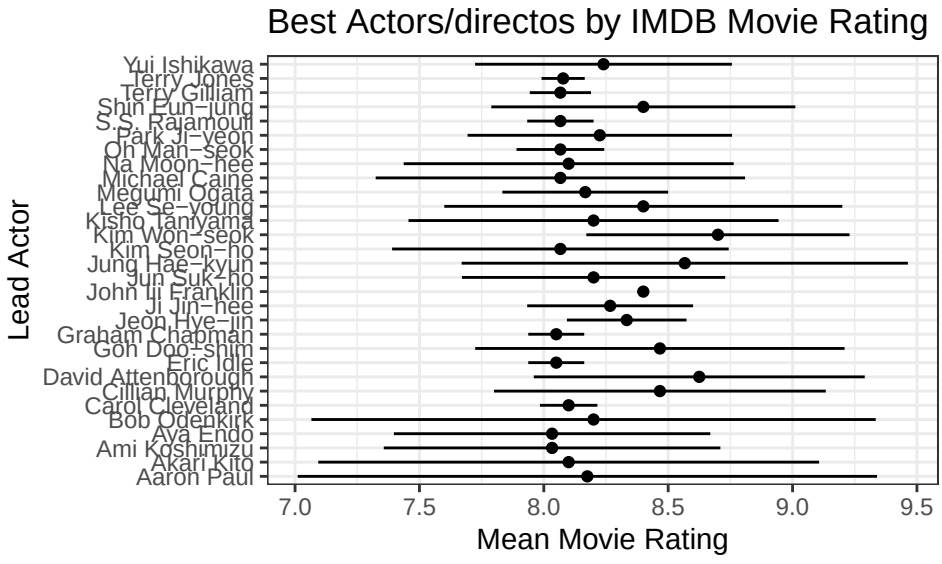
{

}

\caption{ Promedio de IMDB_SCORE en función de los países} \end{figure}

Finalmente, en esta última figura podemos ver la distribución de los score en función de los actores/directores de las misma. De la misma es posible observar que los actores/directores varian en función del rango dinámico, pero en general la mediana se encuentra cercana a 8.

\begin{figure}



{

}

\caption{ Boxplot de IMDB_SCORE de actores/directores} \end{figure} ### Modelos lineales mixtos

Los modelos lineales mixtos fueron propuestos por (Laird and Ware 1982) y en ellos se asume que existe una relación entre el vector de observaciones Y_i del sujeto o grupo i y las covariables.

La forma general de un modelo lineal mixto es:

$$Y = Xb + Zu + e$$

donde: Y es el vector de respuesta (datos), X y Z son matrices de diseño conocidas, b es un vector de parámetros fijos, u (efectos aleatorios) y e (error) son vectores aleatorios no observables, con esperanza nula.

2) A continucación vamos a implementar estos modelos en el marco del objetivo de este trabajo práctico: estimar el IMDB score de las peliculas.

a) Plantear un modelo de efectos fijos para predecir el puntaje de IMDB únicamente en función del país de origen.

- Modelo lineal sin intercept con un efecto fijo por país

b) Plantear un modelo de efectos aleatorios para predecir el puntaje de IMDB únicamente en función del país de origen

- Modelo mixto con intercept fijo y un efecto aleatorio por país

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

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

##
## Attaching package: 'lme4'

## The following object is masked from 'package:nlme':
##
##     lmList

## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ (1 | country)
##    Data: countriesScoresDf
##
## REML criterion at convergence: 12473.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8992 -0.6027  0.0730  0.6914  2.6847
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  country  (Intercept) 0.09351  0.3058
##  Residual             1.21819  1.1037
## Number of obs: 4085, groups:  country, 91
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  6.46713    0.05012     129
```

c) Mostrar las estimaciones de los efectos de ambos modelos en un mismo gráfico e interpretar cómo se diferencian.

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggpubr':
##
##     get_legend
```

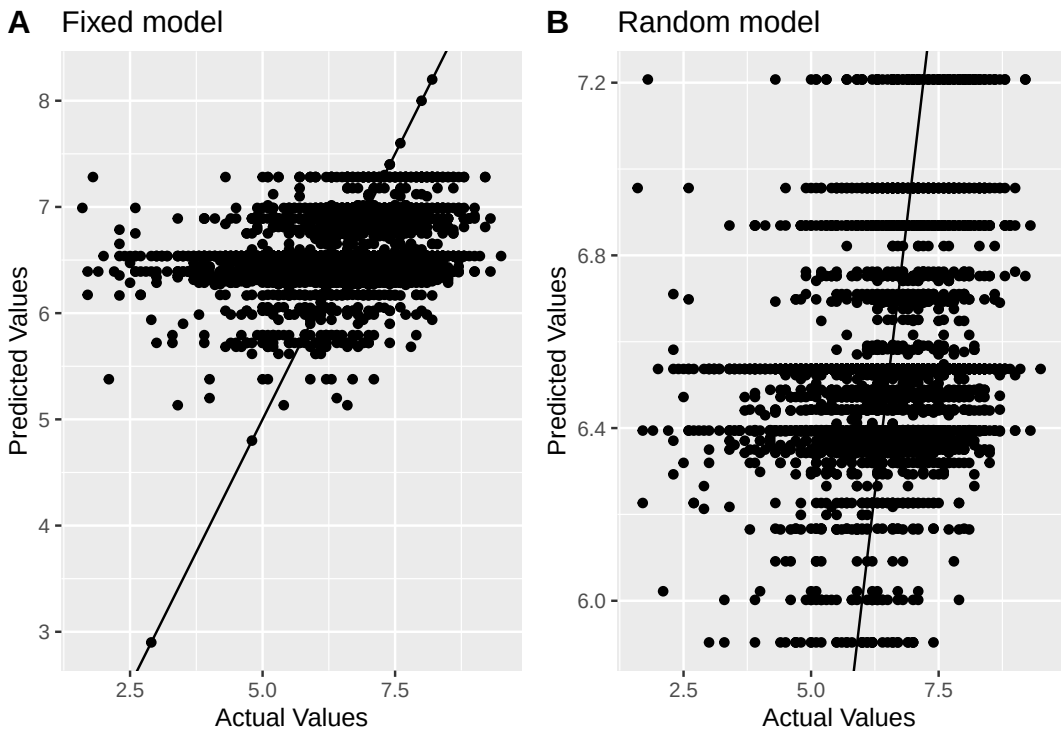


Figura 5: Comparación entre los valores observados y estimados de cada modelo

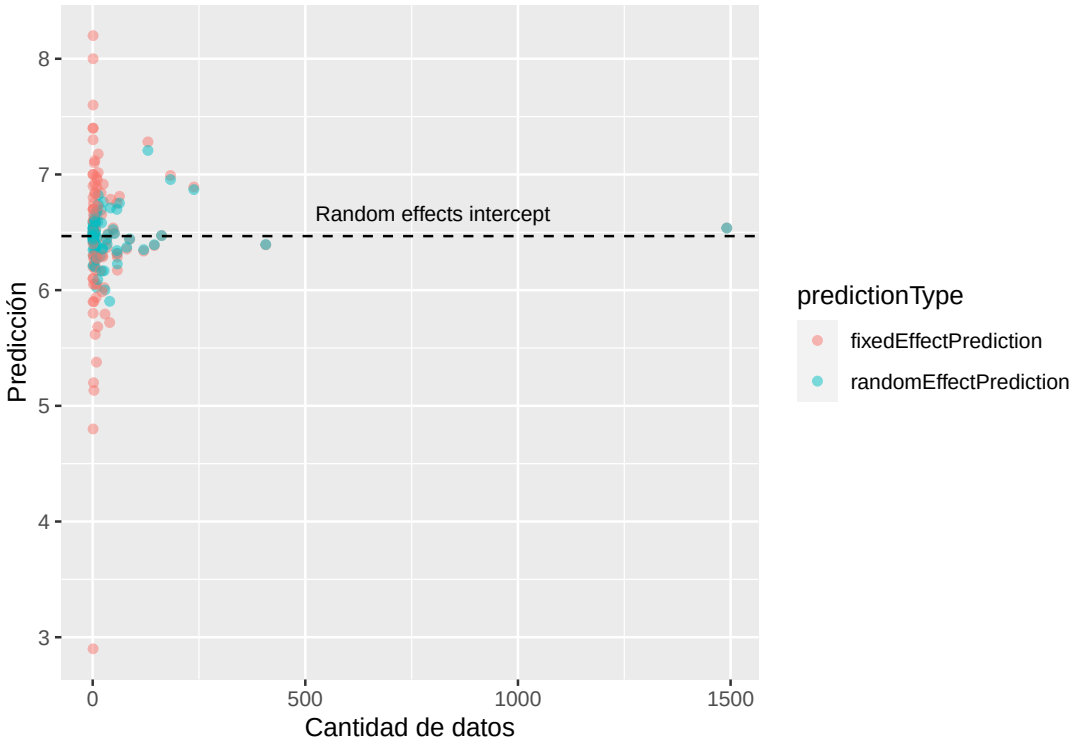


Figura 6: Comparación entre los valores observados y estimados de cada modelo, en función de la cantidad de datos

Veamos ahora cómo se comportan los distintos modelos en función de la cantidad de datos que tenemos para cada país. En este gráfico tenemos un punto para cada país, ubicado de acuerdo a la cantidad de datos y la predicción de score de cada modelo. Marcamos con una línea punteada la tendencia central (intercept) que nos da el modelo de efectos aleatorios.

Es de observar que el modelo de efectos aleatorios tiende a evaluar a los países más cerca de este punto medio dado por el intercept, y la diferencia es más notable para los países con menor cantidad de datos.

3)

a) Usando el modelo de efectos aleatorios del item anterior, decidir, usando la función anova, si agregar´ıa la variable release year.

```
## boundary (singular) fit: see ?isSingular
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ releaseYear + (releaseYear | country)
##    Data: countriesScoresDf
##
## REML criterion at convergence: 12433.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8978 -0.5996  0.0719  0.7161  2.7420
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.  Corr
##  country  (Intercept) 1.201e+00 1.0959608
##           releaseYear 4.951e-07 0.0007036 -1.00
##  Residual             1.202e+00 1.0965426
## Number of obs: 4085, groups:  country, 91
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 41.352264   4.860433   8.508
## releaseYear -0.017304   0.002412  -7.175
##
## Correlation of Fixed Effects:
##             (Intr)
## releaseYear -1.000
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: countriesScoresDf
## Models:
## fit_2: score ~ (1 | country)
## fit_3: score ~ releaseYear + (releaseYear | country)
##       npar   AIC   BIC  logLik deviance  Chisq Df Pr(>Chisq)
## fit_2    3 12475 12494 -6234.5    12469
## fit_3    6 12431 12469 -6209.5    12419 49.971  3  8.103e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

De los resultados observamos que al incorporar en el modelo la variable release year el BIC disminuyó y además resultó estadisticamente significativo, lo que implica que al incorporar dicha variable se redujo la varianza del modelo.

(b) Usando el modelo de efectos aleatorios del item anterior, decidir si agregaría la variable release_year separando la data en dos: entrenamiento y testeo (estimar los coeficientes usando la data de entrenamiento y evualuarlo usando la de testeo).

A partir de ahora vamos a separar nuestro dataset en dos: uno de training o entrenamiento para ajustar el modelo y otro de testeo, para evaluar la performance del mismo.

En la siguiente tabla se muestra el RMSE de los dos modelos con y sin release_year. Del mismo se observa que se obtuvo un menor RMSE al incorporar el año, y cuando gráficamos la relación 1:1 entre las observaciones y predicciones también podemos ver que el modelo que no incluye el año presenta predicciones de valores constantes para varios valores del score observado.

```
## boundary (singular) fit: see ?isSingular
```

```
##             modelo     RMSE
## 1 Sin release_year 1.124635
## 2 Con release_year 1.116546
```

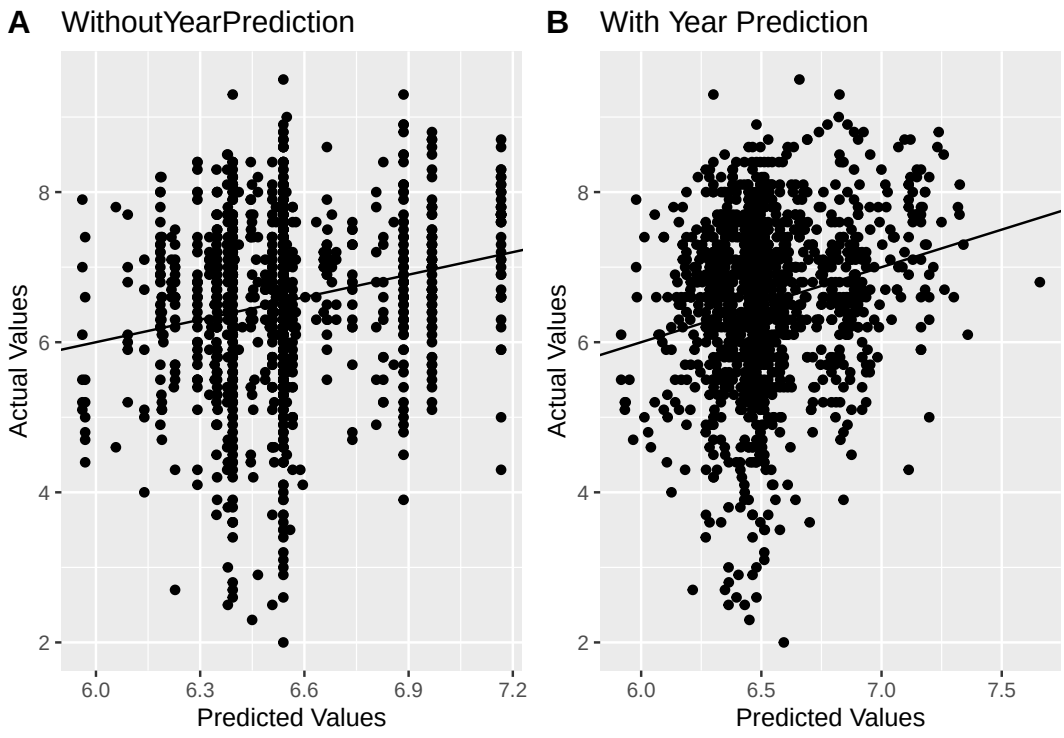


Figura 7: Comparación entre los valores observados y estimados de cada modelo

(c) Comparar ambos items anteriores.

Si bien en este caso se llega a la misma conclusión, siempre es conveniente evaluar los modelos utilizando un dataset independiente. De esta forma podremos analizar si el modelo puede generalizar correctamente y estimar la variable de interés usando muestras que no las ha analizado para estimar los parámetros de los modelos.

Modelos aditivos generalizados (GAM) Los GAMs (del inglés generalized additive models) son una generalización de los GLMs para incorporar formas no lineales de los predictores (splines, Polinomios, o funciones Step, etc. . .). El proceso de suavización en GAMs se lleva a cabo a través de los suavizadores (smoothers), entre los que destacan, entre otros, los Splines penalizados P-Splines.

Al igual que en la sección anterior, vamos a implementar estos modelos para estimar la variable IMDB_score.

a) Usando únicamente la variable release_year, predecir la popularidad de cada título (usando un tipo de modelo que crea adecuado) con una curva de splines penalizados. Usar k = 1, 2, 3, 5, 10, 20, 50 nodos y comparar todas las curvas estimadas en un mismo gráfico.

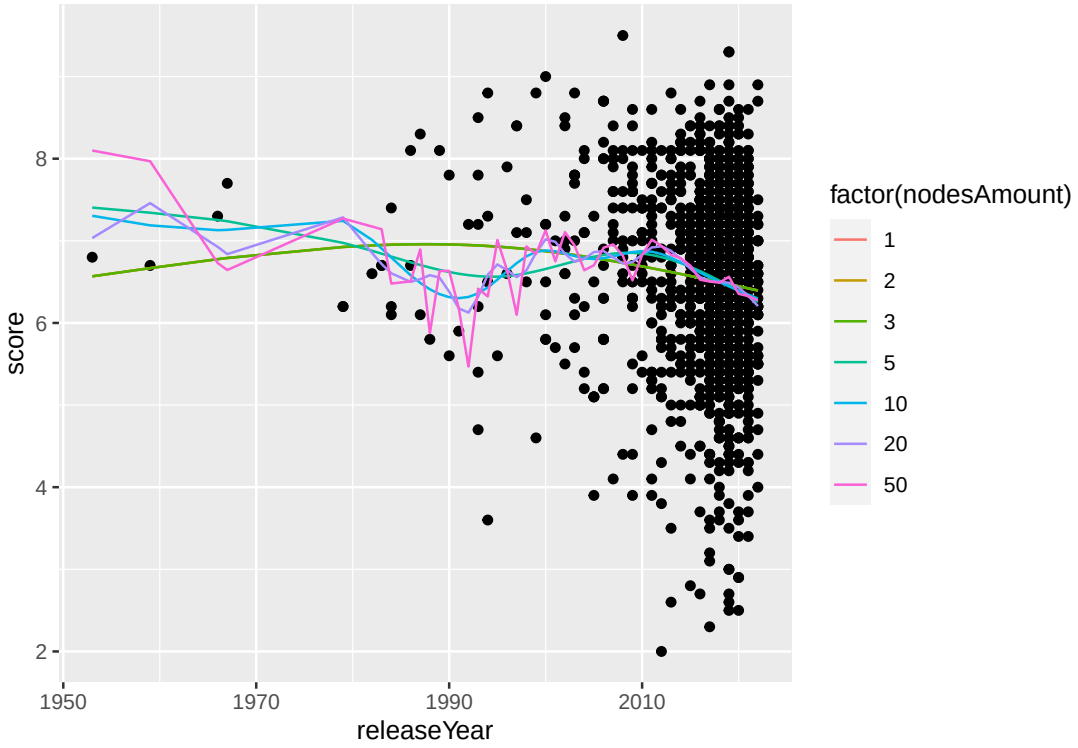


Figura 8: Comparación entre los valores observados y estimados de cada modelo

Se puede ver cómo las curvas con más nodos son más ruidosas en comparación a la de menor k.

Comparemos las predicciones con los distintos modelos:

- Cuando comparamos en la siguiente figura como varía la métrica de RMSE versus k, vemos que el valor más bajo del mismo se obtuvo para los k=1,2 y 5.

Ahora podemos visualirar las estimaciones en un scatter para ver cuanto se alejan de la relación 1:1 entre los valores estimados y predichos. De esta gráfico se observa que ningún modelo parece estimar correctamente el score de las películas, dado que vemos que para distintos valores del score observado se obtiene una predicción que se encuentra entre 6 y 7 del score.

Comparación de modelos En esta sección vamos a implementar una diversidad de modelos y comparar su performance.

a) Dividir al conjunto de datos en entrenamiento y testeo (tambien puede usar otra técnica, como validación cruzada). Con todas las variables que tiene disponibles, probar al menos 10 modelos diferentes y elegir el que minimice el error cuadrático médio de predicción para el rating de IMDB.

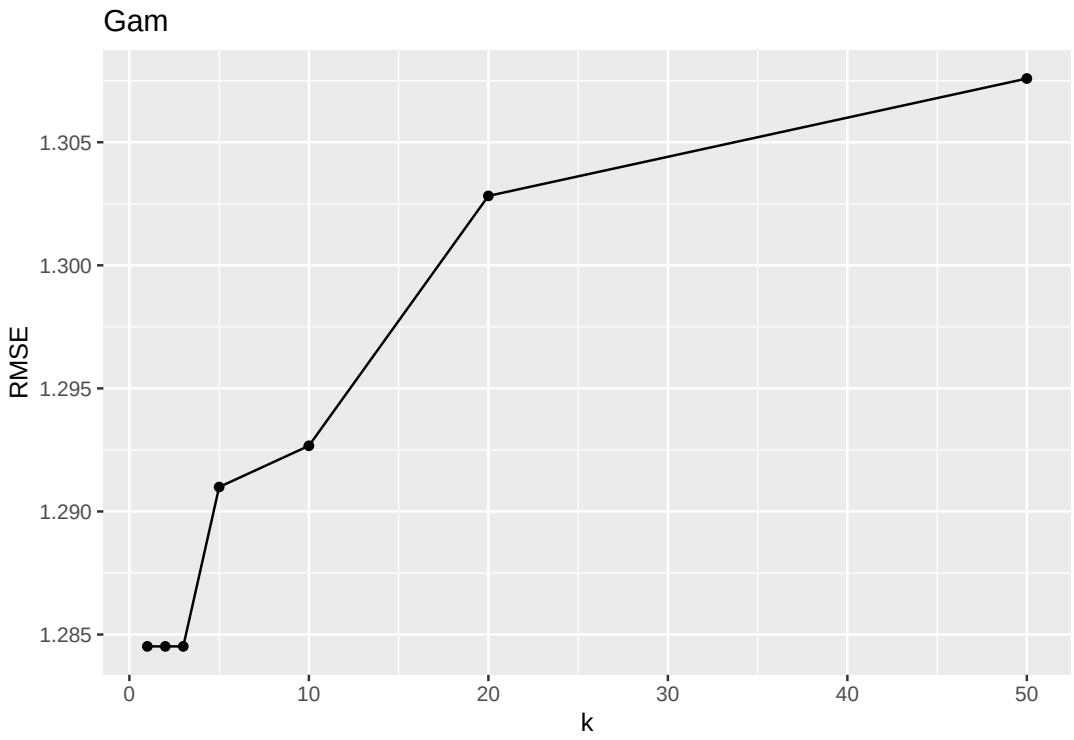


Figura 9: Comparación de RMSE para los modelos en función del valor de k

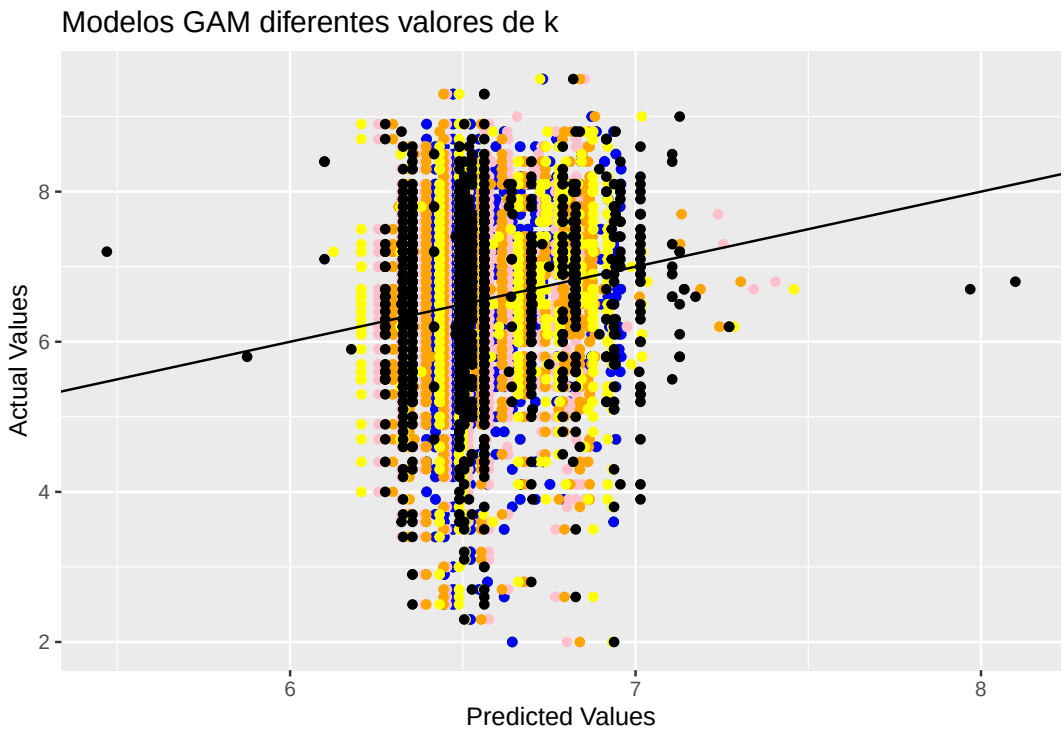


Figura 10: Comparación entre los valores observados y estimados de cada modelo

Primero vamos a generar un nuevo dataframe con todos los features que podemos utilizar.

Ahora vamos a seleccionar distintos modelos y compararlos para quedarnos con el que tenga el menor RMSE. Para este punto también vamos a utilizar la librería caret que nos va a permitir configurar los modelos, hacer selección de variables de una forma más simple. Los modelos que vamos a evalurar serán distintas variaciones de los modelos introducidos en las secciones anteriores.

a) Modelos lineales mixtos

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: imdb_score ~ 1 + crime + documentation + drama + european + family +
##     history + horror + music + romance + scifi + sport + thriller +
##     war + western + (1 | release_year) + (1 | runtime) + (1 |      name)
##    Data: Features_train
##
## REML criterion at convergence: 73018.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.3820 -0.4645  0.0448  0.5991  3.4661
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  name         (Intercept) 0.09707  0.3116
##  runtime      (Intercept) 0.39957  0.6321
##  release_year (Intercept) 0.55128  0.7425
##  Residual                 0.59566  0.7718
## Number of obs: 29229, groups:  name, 24270; runtime, 169; release_year, 57
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    6.58656    0.11274  58.423
## crime          0.02003    0.01511   1.325
## documentation  1.03989    0.02157  48.218
## drama          0.43084    0.01171  36.801
## european       0.11072    0.01630   6.792
## family         0.06414    0.01837   3.491
## history        0.04102    0.02412   1.701
## horror        -0.43608    0.01957 -22.286
## music          0.13473    0.02594   5.193
## romance       -0.20508    0.01341 -15.298
## scifi          0.06616    0.02001   3.306
## sport          0.11718    0.02844   4.120
## thriller      -0.13498    0.01427  -9.462
## war            0.37249    0.03097  12.028
## western        0.61513    0.04425  13.903

##
## Correlation matrix not shown by default, as p = 15 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

b) Modelos lineales mixtos

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: imdb_score ~ 1 + crime + documentation + drama + romance + scifi +
```

```
##     thriller + (runtime | production_countries)
##    Data: Features_train
##
## REML criterion at convergence: 74689.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.9917 -0.4672  0.0314  0.5859  6.3089
##
## Random effects:
##  Groups               Name        Variance  Std.Dev. Corr
##  production_countries (Intercept) 8.7802672 2.96315
##                       runtime     0.0008086 0.02844  -0.98
##  Residual                         0.7274915 0.85293
## Number of obs: 29229, groups:  production_countries, 290
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    6.01328    0.04628 129.941
## crime          0.12927    0.01567   8.251
## documentation  0.99603    0.02077  47.959
## drama          0.45694    0.01222  37.401
## romance       -0.18910    0.01412 -13.390
## scifi         -0.07543    0.02145  -3.517
## thriller      -0.28345    0.01428 -19.853
##
## Correlation of Fixed Effects:
##             (Intr) crime  dcmntt drama  romanc scifi
## crime       -0.025
## documentatn -0.095 -0.005
## drama       -0.164 -0.054  0.187
## romance     -0.060  0.109  0.102 -0.129
## scifi       -0.048  0.129  0.071  0.085  0.016
## thriller    -0.064 -0.395  0.121 -0.027  0.157 -0.241
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 2.19473 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
```

c) Modelos lineales mixtos

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: imdb_score ~ 1 + crime + documentation + drama + european + family +
##     history + horror + music + romance + scifi + sport + thriller +
##     war + western + (1 | production_countries) + (1 | name)
##    Data: Features_train
##
## REML criterion at convergence: 76864.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.2809 -0.4846  0.0384  0.5377  3.0634
##
## Random effects:
##  Groups               Name        Variance Std.Dev.
##  name                 (Intercept) 0.08032  0.2834
```

```
##  production_countries (Intercept) 0.51107  0.7149
##  Residual                         0.71149  0.8435
## Number of obs: 29229, groups:  name, 24270; production_countries, 290
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    5.92458    0.04709 125.801
## crime          0.10409    0.01649   6.311
## documentation  0.78340    0.02242  34.939
## drama          0.51529    0.01251  41.184
## european       0.27054    0.02555  10.588
## family        -0.21532    0.01981 -10.868
## history        0.24271    0.02620   9.265
## horror        -0.53629    0.02105 -25.476
## music          0.29533    0.02637  11.201
## romance       -0.14502    0.01439 -10.079
## scifi          0.01378    0.02198   0.627
## sport          0.30762    0.02988  10.295
## thriller      -0.13164    0.01550  -8.495
## war            0.51216    0.03514  14.576
## western        0.71695    0.04298  16.680

##
## Correlation matrix not shown by default, as p = 15 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

d) Modelos aditivos utilizando la libreria caret

Para el siguiente modelo, se eliminaron del dataset algunas variables dado por el tiempo que tardaba el modelo en realizar el ajuste de los parámetros (estas variables fueron: nombre, año y país).

```
##
## Family: Gamma
## Link function: log
##
## Formula:
## .outcome ~ action + animation + comedy + crime + documentation +
##     drama + european + family + fantasy + history + horror +
##     romance + scifi + thriller + s(runtime)
##
## Parametric coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.8328202  0.0025096 730.327  < 2e-16 ***
## action        -0.0243868  0.0023734 -10.275  < 2e-16 ***
## animation      0.1208302  0.0042918  28.154  < 2e-16 ***
## comedy        -0.0285925  0.0021121 -13.537  < 2e-16 ***
## crime          0.0108669  0.0025277   4.299 1.72e-05 ***
## documentation  0.1399218  0.0036079  38.782  < 2e-16 ***
## drama          0.0541566  0.0020468  26.459  < 2e-16 ***
## european       0.0213452  0.0026194   8.149 3.82e-16 ***
## family        -0.0405138  0.0033026 -12.267  < 2e-16 ***
## fantasy        0.0135183  0.0030514   4.430 9.45e-06 ***
## history        0.0192692  0.0036168   5.328 1.00e-07 ***
## horror        -0.0777712  0.0032779 -23.726  < 2e-16 ***
## romance       -0.0347522  0.0022180 -15.669  < 2e-16 ***
```

```
## scifi         -0.0001309  0.0033626  -0.039    0.969
## thriller      -0.0334918  0.0024915 -13.442  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df     F p-value
## s(runtime) 8.245  8.814 488.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.283   Deviance explained = 25.2%
## GCV = 0.02451  Scale est. = 0.021369  n = 29229
```

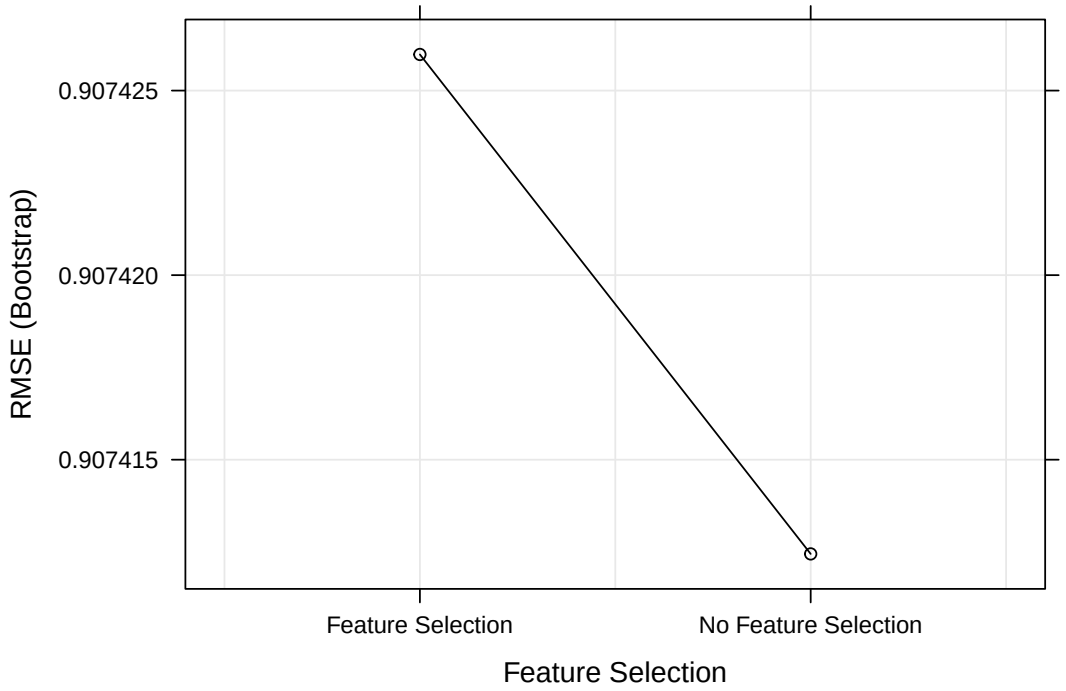


Figura 11: RMSE selección de variables

e) Modelos aditivos utilizando la libreria caret

```
##
## Family: Gamma
## Link function: log
##
## Formula:
## .outcome ~ action + animation + comedy + crime + documentation +
##     drama + european + family + fantasy + history + horror +
##     romance + scifi + thriller + s(runtime)
##
## Parametric coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.8328202  0.0025096 730.327  < 2e-16 ***
## action        -0.0243868  0.0023734 -10.275  < 2e-16 ***
## animation      0.1208302  0.0042918  28.154  < 2e-16 ***
```

```
## comedy        -0.0285925  0.0021121 -13.537  < 2e-16 ***
## crime          0.0108669  0.0025277   4.299 1.72e-05 ***
## documentation  0.1399218  0.0036079  38.782  < 2e-16 ***
## drama          0.0541566  0.0020468  26.459  < 2e-16 ***
## european       0.0213452  0.0026194   8.149 3.82e-16 ***
## family        -0.0405138  0.0033026 -12.267  < 2e-16 ***
## fantasy        0.0135183  0.0030514   4.430 9.45e-06 ***
## history        0.0192692  0.0036168   5.328 1.00e-07 ***
## horror        -0.0777712  0.0032779 -23.726  < 2e-16 ***
## romance       -0.0347522  0.0022180 -15.669  < 2e-16 ***
## scifi         -0.0001309  0.0033626  -0.039    0.969
## thriller      -0.0334918  0.0024915 -13.442  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F p-value
## s(runtime) 8.245  8.814 488.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.283   Deviance explained = 25.2%
## GCV = 0.02451  Scale est. = 0.021369  n = 29229
```

f) Elastic-Net Regularized Generalized Linear Models with caret

```
##             Length Class      Mode
## a0            74   -none-     numeric
## beta        1406   dgCMatrix  S4
## df            74   -none-     numeric
## dim            2   -none-     numeric
## lambda        74   -none-     numeric
## dev.ratio     74   -none-     numeric
## nulldev        1   -none-     numeric
## npasses        1   -none-     numeric
## jerr           1   -none-     numeric
## offset         1   -none-     logical
## call           5   -none-     call
## nobs           1   -none-     numeric
## lambdaOpt      1   -none-     numeric
## xNames        19   -none-     character
## problemType    1   -none-     character
## tuneValue      2   data.frame list
## obsLevels      1   -none-     logical
## param          1   -none-     list
```

g) Elastic-Net Regularized Generalized Linear Models with caret

```
##             Length Class      Mode
## a0            67   -none-     numeric
## beta        1340   dgCMatrix  S4
## df            67   -none-     numeric
## dim            2   -none-     numeric
## lambda        67   -none-     numeric
## dev.ratio     67   -none-     numeric
## nulldev        1   -none-     numeric
```

```
## npasses        1   -none-     numeric
## jerr           1   -none-     numeric
## offset         1   -none-     logical
## call           5   -none-     call
## nobs           1   -none-     numeric
## lambdaOpt      1   -none-     numeric
## xNames        20   -none-     character
## problemType    1   -none-     character
## tuneValue      2   data.frame list
## obsLevels      1   -none-     logical
## param          1   -none-     list
```

h) Modelo GLM bayesiano utilizando la libreria caret

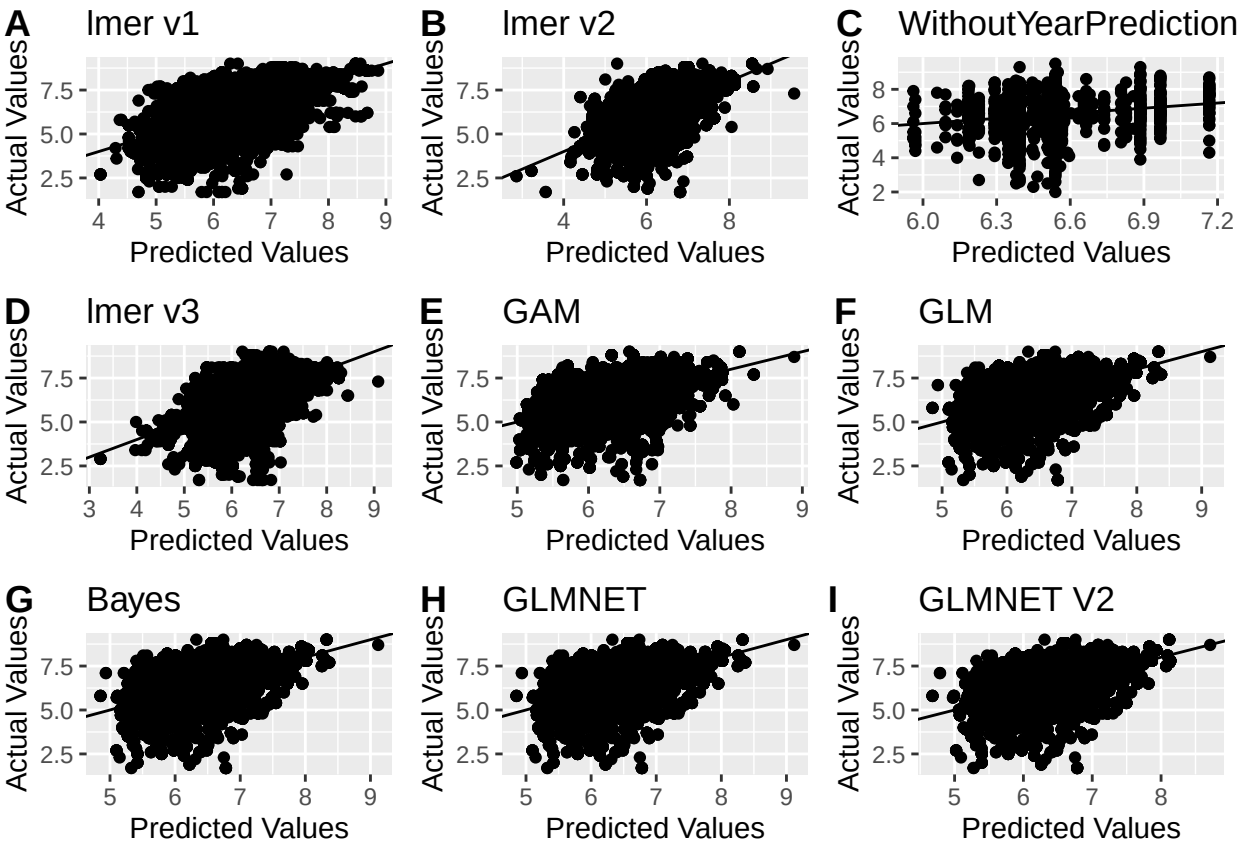
```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.12614  -0.07846   0.01396   0.08764   0.42324
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.593e+00  4.853e-03 328.353  < 2e-16 ***
## action        -2.444e-02  2.394e-03 -10.206  < 2e-16 ***
## animation      1.215e-01  4.292e-03  28.312  < 2e-16 ***
## comedy        -2.943e-02  2.153e-03 -13.672  < 2e-16 ***
## crime          1.241e-02  2.560e-03   4.849 1.25e-06 ***
## documentation  1.379e-01  3.733e-03  36.939  < 2e-16 ***
## drama          5.511e-02  2.065e-03  26.690  < 2e-16 ***
## european       1.856e-02  2.644e-03   7.017 2.31e-12 ***
## family        -4.594e-02  3.351e-03 -13.709  < 2e-16 ***
## fantasy        1.381e-02  3.076e-03   4.489 7.20e-06 ***
## history        3.629e-03  3.826e-03   0.948    0.343
## horror        -8.369e-02  3.307e-03 -25.308  < 2e-16 ***
## music          2.478e-02  4.210e-03   5.886 4.01e-09 ***
## reality        1.365e-01  3.234e-02   4.222 2.43e-05 ***
## romance       -3.413e-02  2.244e-03 -15.210  < 2e-16 ***
## scifi          3.588e-04  3.403e-03   0.105    0.916
## sport          2.017e-02  4.561e-03   4.423 9.78e-06 ***
## thriller      -2.869e-02  2.521e-03 -11.380  < 2e-16 ***
## war            6.372e-02  4.752e-03  13.408  < 2e-16 ***
## western        8.590e-02  6.812e-03  12.609  < 2e-16 ***
## runtime        2.129e-03  3.973e-05  53.576  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.02186632)
##
##     Null deviance: 956.58  on 29228  degrees of freedom
## Residual deviance: 731.09  on 29208  degrees of freedom
## AIC: 82401
##
## Number of Fisher Scoring iterations: 6
```

g) Modelos lineales mixtos en carte utilizando selección de variables. En este apartado, vamos a comparar distintos modelos basados en la selección de variables y quedarnos con la que tiene mejor performace.

Resultados generales

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.12645  -0.07848   0.01422   0.08727   0.42341
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.593e+00  4.822e-03 330.386  < 2e-16 ***
## action        -2.442e-02  2.313e-03 -10.561  < 2e-16 ***
## animation      1.217e-01  4.273e-03  28.478  < 2e-16 ***
## comedy        -2.960e-02  2.144e-03 -13.807  < 2e-16 ***
## crime          1.234e-02  2.535e-03   4.867 1.14e-06 ***
## documentation  1.382e-01  3.719e-03  37.163  < 2e-16 ***
## drama          5.519e-02  2.062e-03  26.764  < 2e-16 ***
## european       1.859e-02  2.640e-03   7.042 1.93e-12 ***
## family        -4.594e-02  3.350e-03 -13.716  < 2e-16 ***
## fantasy        1.371e-02  3.072e-03   4.462 8.14e-06 ***
## horror        -8.379e-02  3.302e-03 -25.379  < 2e-16 ***
## music          2.472e-02  4.208e-03   5.874 4.30e-09 ***
## reality        1.367e-01  3.234e-02   4.226 2.38e-05 ***
## romance       -3.424e-02  2.240e-03 -15.284  < 2e-16 ***
## sport          1.996e-02  4.552e-03   4.384 1.17e-05 ***
## thriller      -2.873e-02  2.498e-03 -11.504  < 2e-16 ***
## war            6.514e-02  4.497e-03  14.487  < 2e-16 ***
## western        8.576e-02  6.803e-03  12.606  < 2e-16 ***
## runtime        2.135e-03  3.909e-05  54.619  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.02186374)
##
##     Null deviance: 956.58  on 29228  degrees of freedom
## Residual deviance: 731.11  on 29210  degrees of freedom
## AIC: 82397
##
## Number of Fisher Scoring iterations: 4
```

```
##            lmer.v1   lmer v2   lmer v3    glmnet   glmnet2       gam     bayes
## RMSE     0.8296435 0.8490823 0.8843064 0.9179656 0.9147441 0.9056191 0.9158944
## Rsquared 0.4019698 0.3738801 0.3209527 0.2679540 0.2730772 0.2874719 0.2712616
## MAE      0.6176438 0.6083375 0.6433031 0.6960171 0.6926851 0.6844227 0.6951478
##                gml
## RMSE     0.9158422
## Rsquared 0.2713429
## MAE      0.6948584
```



Predicciones utilizando el mejor modelo obtenido anteriormente

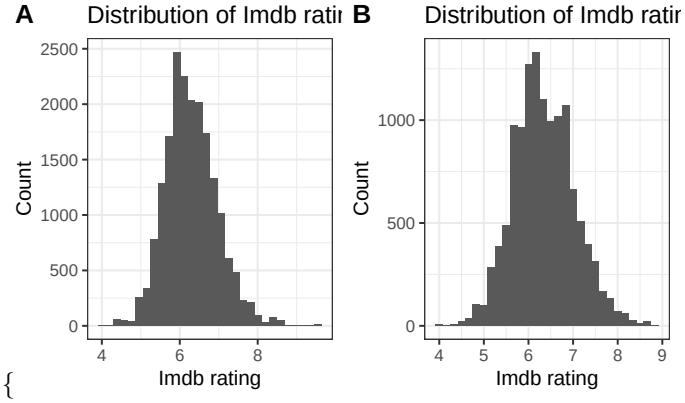
Entre estos, el mejor modelo parece ser lmer.v1, que es un modelo lineal mixto, y es con el que realizaremos las predicciones del punto 6.

Primero vamos a realizar los mismos pasos de ingenieria de features que el dataset de training, y luego realizar las predicciones.

Si bien no contamos con los valores reales de los score, vamos a realizar un breve análisis visual relacionando las predicciones con los features. De los gráficos podemos ver que el rango dinámico de las estimaciones del test es similar a las predicciones del dataset de entrenamiento. Además, se observa que presentan una patrón similar de asociación con las variables runtime y release year.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

\begin{figure}

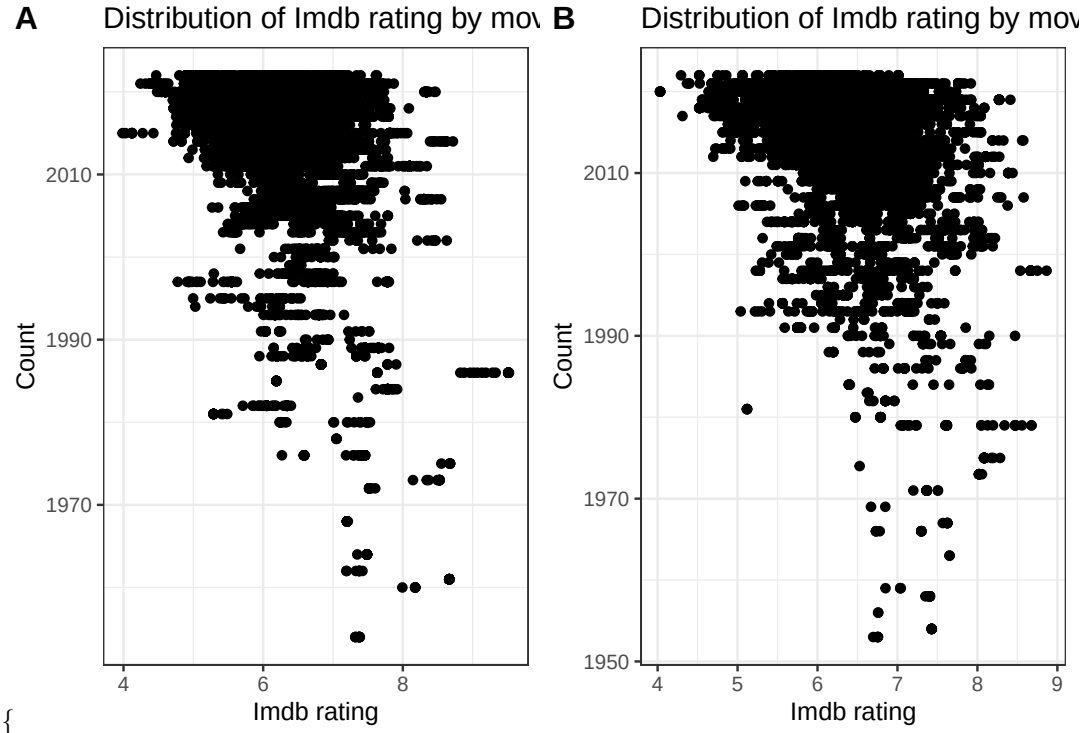


{

}

\caption{ Distribución de las predicciones de IMDB_SCORE para las películas} \end{figure}

\begin{figure}

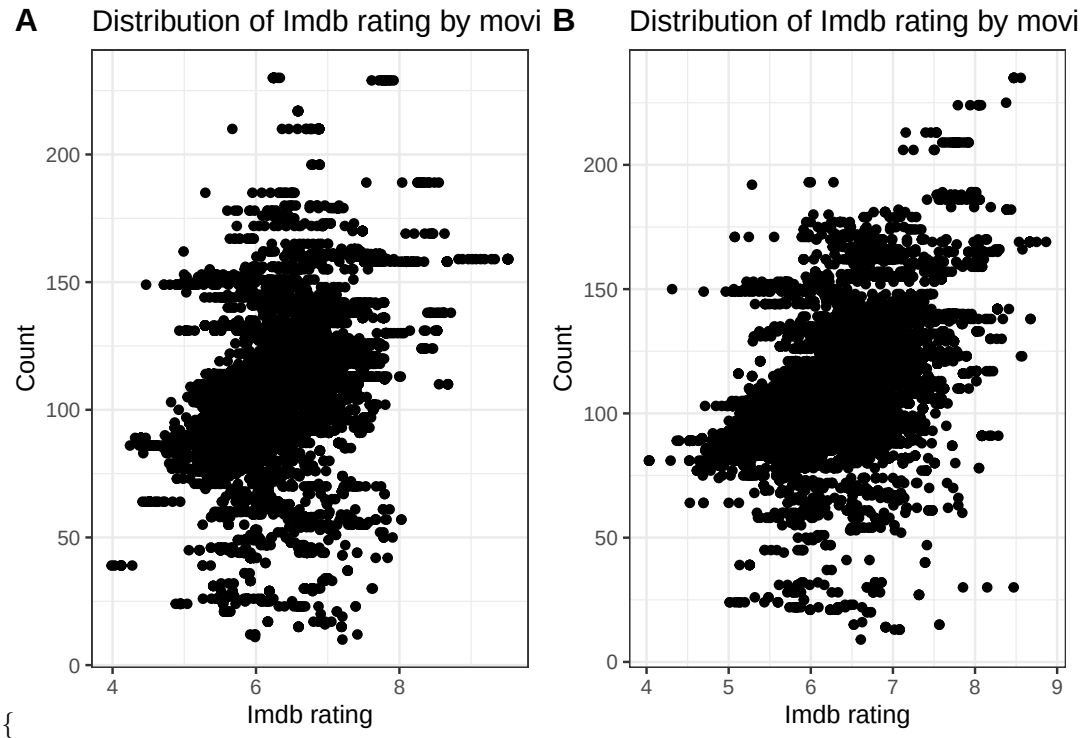


{

}

\caption{ Relación entre las predicciones de IMDB_SCORE y el año de estreno para las películas} \end{figure}

\begin{figure}



{

}

\caption{ Relación entre las predicciones de IMDB_SCORE y el runtime para las películas} \end{figure}