Indicium - Desafio Cientista de Dados - Renato Susin

Renato Susin

2025-09-07

Instalar e carregar bibliotecas necessárias

```
#install.packages <- c("randomForest", "caret", "quanteda", "topicmodels", "ExpDes.pt", "rvest", "xml2"
library(randomForest)
library(caret)
library(ExpDes.pt)
library(tidyr)
library(dplyr)
library(quantmod)
library(rvest)
library(xml2)
library(tibble)
library(stringr)
library(ggplot2)
library(mvtnorm)
library(GGally)
library(mvShapiroTest)
library(psych)
library(stringr)
library(quanteda)
library(topicmodels)
library(tm)
library(tidytext)
library(reshape2)
```

Carregando e manipulando as as bases de dados que usaremos no modelo

```
# Carregar o data frame
Filmes_1 <- read.csv("~/Downloads/desafio_indicium_imdb.csv", header=TRUE)

# Ajustar linha 966 (ela tem um erro no ano de lançamento do filme)
Filmes_1 [966, "Released_Year"] <- 1995

# Vamos aumentar nossa base de dados, também vamos incluir o orçamento na análise

#(do banco de dados : https://www.kaggle.com/datasets/ramakrushnamohapatra/movies?resource=download)
Filmes_2 <- read.csv("~/Downloads/Movies.csv", header=TRUE)


Filmes_1$Gross <- gsub("\\$|,","",Filmes_1$Gross)
# Organizar as duas bases de dados para juntar
```

```r
Filmes1 <- matrix(nrow = 1)
Filmes1$Titulo <- as.character(Filmes_1$Series_Title)
Filmes1$Ano <- as.integer(Filmes_1$Released_Year)
Filmes1$NotaIMDB <- as.numeric(Filmes_1$IMDB_Rating)
Filmes1$Genero <- as.character(Filmes_1$Genre)
Filmes1$Votos <- as.integer(Filmes_1$No_of_Votes)
Filmes1$Ator1 <- as.character(Filmes_1$Star1)
Filmes1$Ator2 <- as.character(Filmes_1$Star2)
Filmes1$Ator3 <- as.character(Filmes_1$Star3)
Filmes1$Diretor <- as.character(Filmes_1$Director)
Filmes1$Faturamento <- as.numeric(Filmes_1$Gross)
Filmes1$Classificacao <- as.factor(Filmes_1$Certificate)
Filmes1 <- as.data.frame(Filmes1)


Filmes2 <- matrix(nrow = 1)
Filmes2$Titulo <- as.character(Filmes_2$movie_title)
Filmes2$Ano <- as.integer(Filmes_2$title_year)
Filmes2$NotaIMDB <- as.numeric(Filmes_2$imdb_score)
Filmes2$Genero <- as.character(Filmes_2$genres)
Filmes2$Votos <- as.integer(Filmes_2$num_voted_users)
Filmes2$Ator1 <- as.character(Filmes_2$actor_1_name)
Filmes2$Ator2 <- as.character(Filmes_2$actor_2_name)
Filmes2$Ator3 <- as.character(Filmes_2$actor_3_name)
Filmes2$Diretor <- as.character(Filmes_2$actor_3_name)
Filmes2$Faturamento <- as.numeric(Filmes_2$gross)
Filmes2$Classificacao <- as.factor(Filmes_2$content_rating)
Filmes2 <- as.data.frame(Filmes2)


Filmes <- rbind(Filmes1, Filmes2)

#Eliminar os filmes repetidos nos dois dataframes
Filmes <- Filmes %>%
  mutate(
    Titulo_padr = tolower(str_replace_all(Titulo, "[[:punct:]]|\\s+", "")) # Tiro toda pontuação e espa
  ) %>%
  group_by(Titulo_padr, Ano) %>%
  slice_head(n = 1) %>%
  ungroup()


# Incluir o orçamento na base de dados
Filmes_orcamento <- Filmes_2[,c(12,23)]


Filmes_orcamento <- Filmes_orcamento %>%
  mutate(
    Titulo_padr = tolower(str_replace_all(movie_title, "[[:punct:]]|\\s+", ""))
  )

Filmes <- merge(Filmes,Filmes_orcamento, by = "Titulo_padr")
```

```r
Filmes$Orcamento <- as.numeric(Filmes$budget)

#tirar as colunas desnecessárias
Filmes <- Filmes [,c(-1,-2,-14,-15)]


Filmes <- na.omit(Filmes)# Excluir linhas com dados faltantes
```

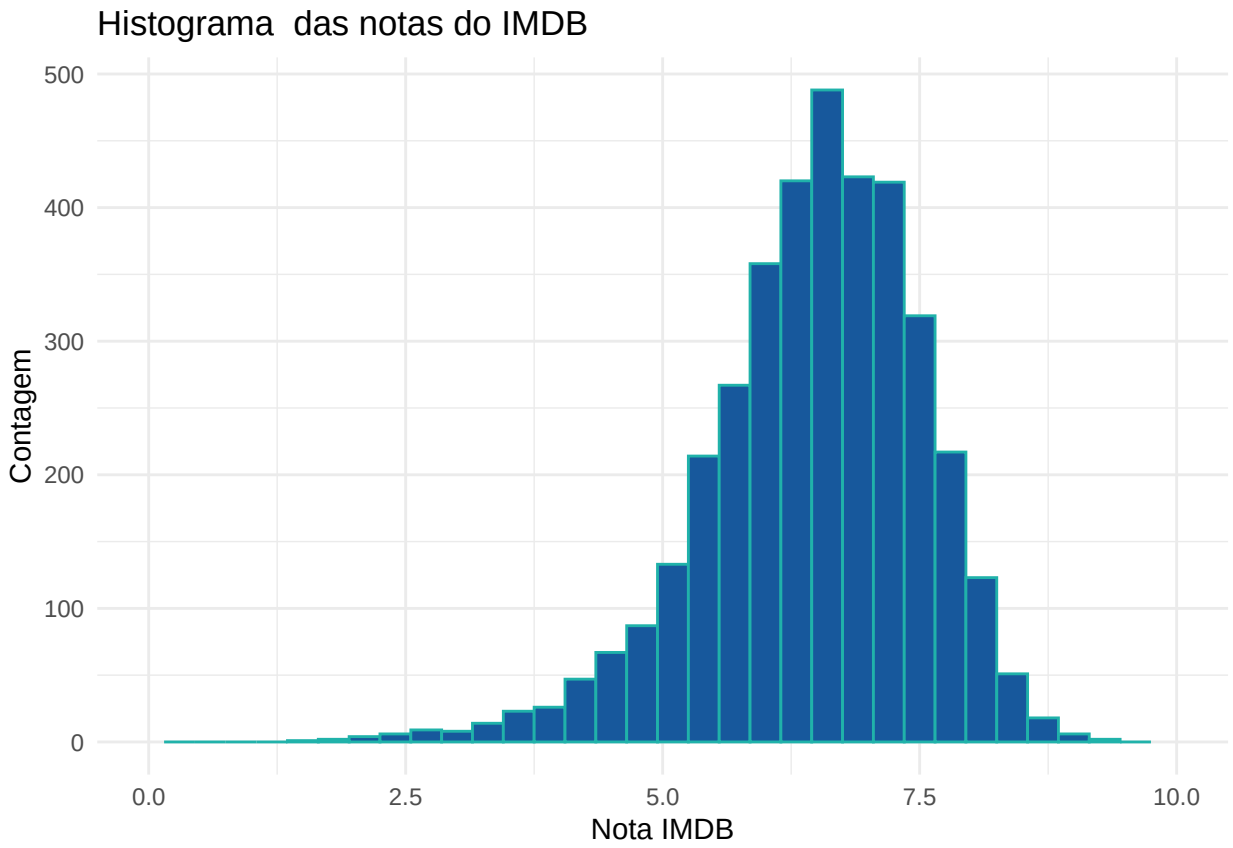
Vamos carregar os dados de inflação para corrigir os dados monetários nominais para reais

```r
# Precisamos deflcionar os dados de faturamento dos filmes para ter o faturamento real
#link para a obtenção dos dados:
#https://inflationdata.com/Inflation/Consumer_Price_Index/HistoricalCPI.aspx?reloaded=true#Table

IndicePrecosUSA <- read.csv("~/Downloads/Índice de Preços USA - Página1.csv")

# Baixar CPI - Índice de preços dos USA

# Obter os números índice para a correção monetária

IndicePrecosUSA$Num_ind <- IndicePrecosUSA$Indice/315.605 #Transformar tudo em valores de 2024

IndicePrecosUSA <- as.data.frame(IndicePrecosUSA)

#Inserir no dataframe
Filmes <-merge(Filmes, IndicePrecosUSA, by = "Ano")


Filmes$FaturamentoReal <- Filmes$Faturamento/Filmes$Num_ind
Filmes$OrcamentoReal <- Filmes$Orcamento/Filmes$Num_ind

Filmes$LucroReal <- Filmes$FaturamentoReal - Filmes$OrcamentoReal


Filmes <- Filmes %>%
  filter(LucroReal >= -100000000)  # Excluir quem teve prejuizo maior que 100M
#Podem enviesar o modelo ou ter dados equivocados
```

Análise Exploratória dos Dados

Histograma de frequência das notas IMDb observadas

```r
ggplot(data = Filmes, aes(x = NotaIMDB))+
  geom_histogram(binwidth = 0.3, fill = "#17589c", color = "lightseagreen")+ #Tentativa de imitar as co
  xlim(0,10)+
  labs(
    title = "Histograma  das notas do IMDB",
    x = "Nota IMDB",
    y = "Contagem")+
  theme_minimal()
```




```
print(mean(Filmes$NotaIMDB)) #6.65
```

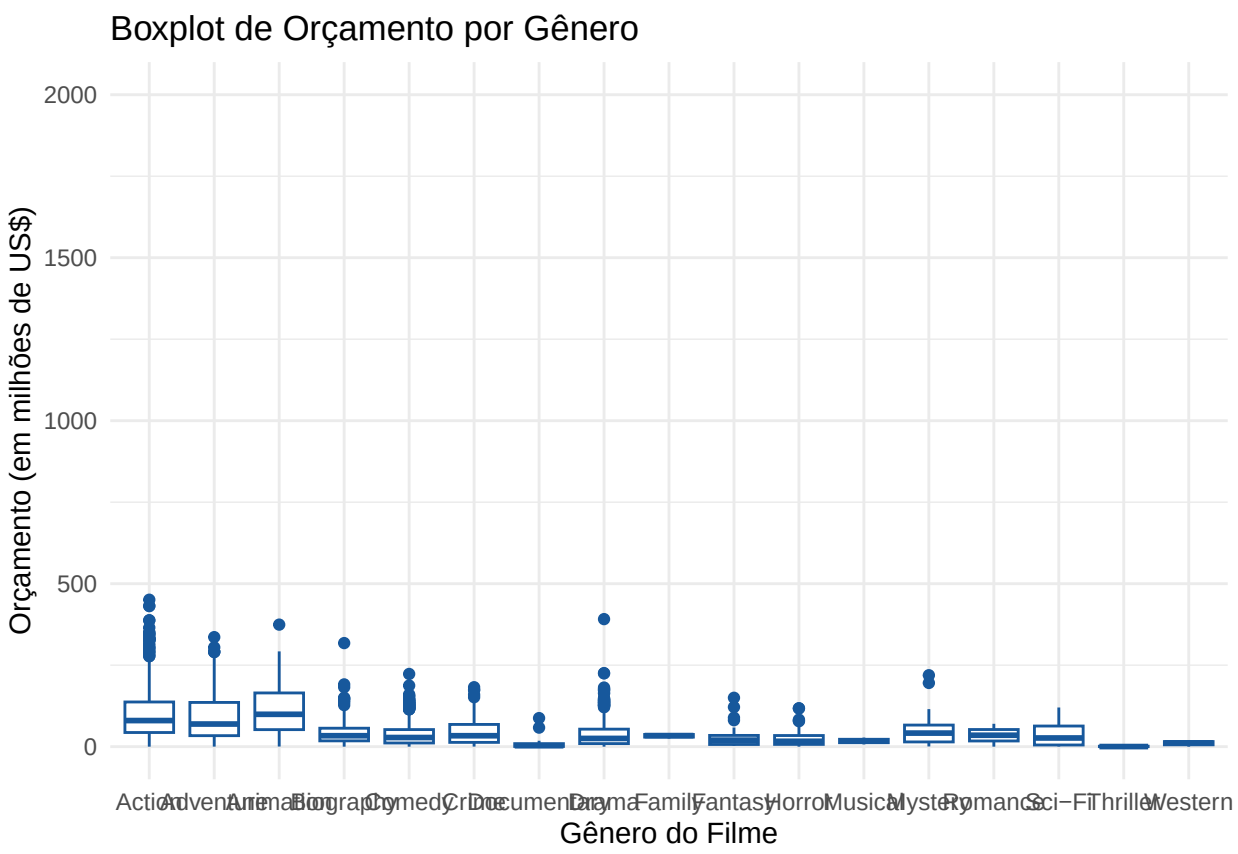
```
## [1] 6.468417
```

Análise de orçamento e lucro por gênero

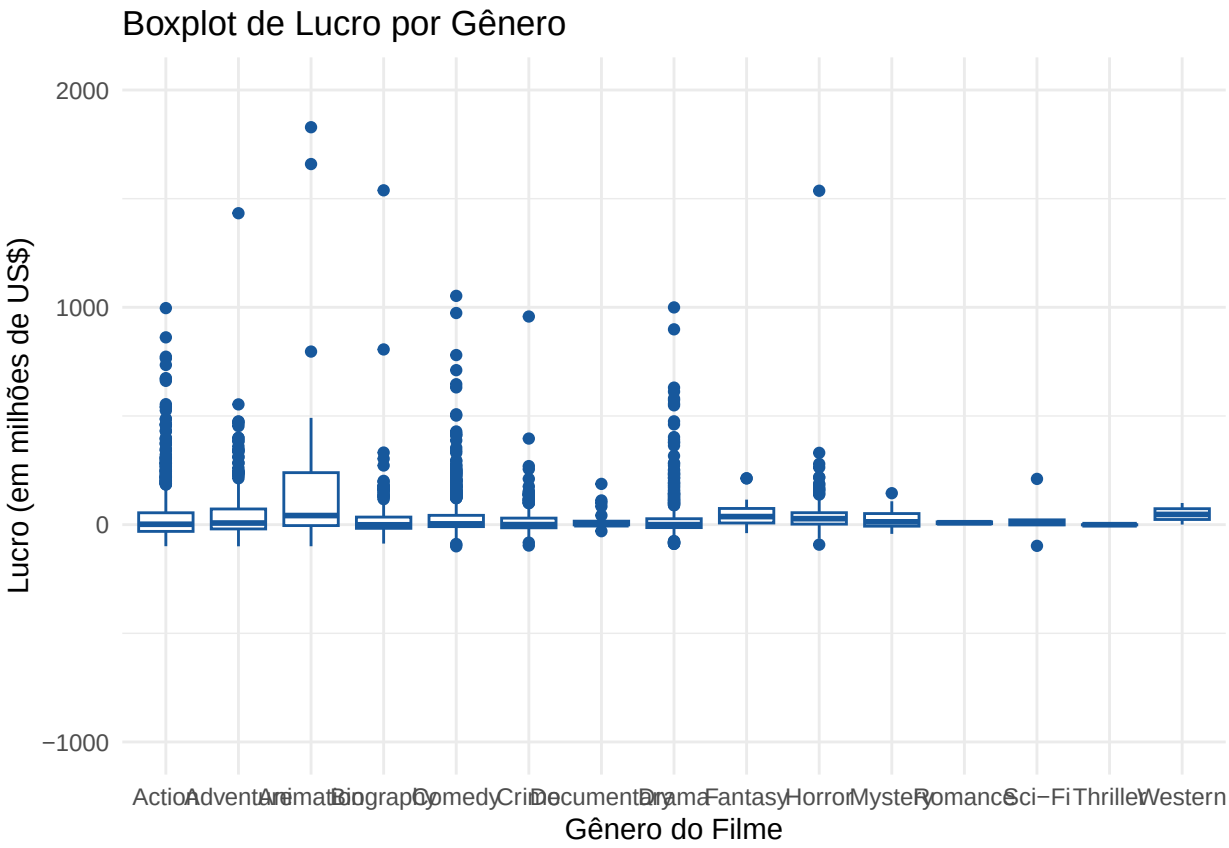
Vamos analisar o orçamento e lucro de cada gênero de filme com o intuito de verificar qual a categoria que os estúdios devem investir com o intuito de maximizar a renda

```
Filmes <- Filmes %>%
  separate(Genero, into = c("Genero1", "Genero2", "Genero3"), sep = ",", fill = "right", extra = "drop")

Filmes <- Filmes %>%
  separate(
    col = Genero1,
    into = c("Genero1", "Genero2", "Genero3", "Genero4", "Genero5"),
    sep = "\\|",
    fill = "right",
    extra = "drop"
  )

#Boxplot de orçamento por gênero
ggplot(data = Filmes, aes(x = Genero1, y = OrcamentoReal/1000000))+
  geom_boxplot(fill = "white", color = "#17589c")+
  ylim(0,2000)+
  labs(
    title = "Boxplot de Orçamento por Gênero",
    x = "Gênero do Filme",
```

```
    y = "Orçamento (em milhões de US$)")+
  theme_minimal()
```




```
#Boxplot de lucro por gênero


# Por ter poucos filmes dos gêneros Família e Musical, estão enviesando o boxplot
# Vamos tirá-los

Filmes_filtrados <- Filmes %>%
  filter(!Genero1 %in% c("Family","Musical"))


ggplot(Filmes_filtrados, aes(x = Genero1, y = LucroReal / 1000000)) +
  geom_boxplot(fill = "white", color = "#17589c") +
  coord_cartesian(ylim = c(-1000, 2000)) +
  labs(
    title = "Boxplot de Lucro por Gênero",
    x = "Gênero do Filme",
    y = "Lucro (em milhões de US$)"
  ) +
  theme_minimal()
```




```
# Agora vamos ver a probabilidade de lucro por cada gênero

#Vamos primeiro criar um dataframe com filmes lucrativos
Filmes_lucrativos <- Filmes %>%
  filter(LucroReal >= 0 )


Genero_filmes_lucrativos <- as.data.frame(table(Filmes_lucrativos$Genero1))

Genero_filmes_lucrativos <- Genero_filmes_lucrativos %>%
  filter(Freq >= 10 ) #Tirar generos  com poucos filmes para não enviesar

Filmes_lucrativos <- Filmes %>%
  filter(LucroReal >= 0 )

#Agora a mesma coisa com todos, para calcular a probabilidade de um filme dar lucro
Genero_filmes <- as.data.frame(table(Filmes$Genero1))


Tabela_Generos <- merge(Genero_filmes_lucrativos, Genero_filmes, by = "Var1")
Tabela_Generos$ProbLucro <- Tabela_Generos$Freq.x/Tabela_Generos$Freq.y
colnames(Tabela_Generos) <- c("Gênero", "Filmes lucrativos", "Filmes totais",
                              "Probabilidade de Lucro")
```

Com base nisso, verificamos que o gênero Animação é o mais lucrativo, mas não deve ser o preterido pelos estúdios menores em decorrência da alta necessidade de capital.

A média de faturamento por gênero confirma o que foi analisado pelos gráficos

```r
media_orcamento_genero <- Filmes %>%
  group_by(Genero1) %>%
  summarise(
    MediaOrcamento = mean(OrcamentoReal),

  )

media_lucro_genero <- Filmes %>%
  group_by(Genero1) %>%
  summarise(
    MediaLucro = mean(LucroReal),

  )
```

Agora vamos transformar o número de filmes que cada diretor e ator participou como proxy para a experiência e qualidade do artista, que será utilizada em modelos que necessitam de variáveis numéricas, apenas:

```r
# Diretor
Diretor <- table(Filmes$Diretor)

Diretor <- as.data.frame(Diretor)

colnames(Diretor) <- c("Diretor", "NumeroFilmesDir")

Filmes <- merge(Filmes, Diretor, by = "Diretor")


# Fazer o mesmo para atores

#Como tem-se tres colunas para atores, vamos unir tudo em apenas um data.frame
Ator1 <- table(Filmes$Ator1)
Ator2 <- table(Filmes$Ator2)
Ator3 <- table(Filmes$Ator3)

# O número que aparecerá no dataframe principal é a soma dos três principais atores no elenco
Ator1 <- as.data.frame(Ator1)
Ator2 <- as.data.frame(Ator2)
Ator3 <- as.data.frame(Ator3)

colnames(Ator1) <- c("Ator", "NumeroFilmesAct")
colnames(Ator2) <- c("Ator", "NumeroFilmesAct")
colnames(Ator3) <- c("Ator", "NumeroFilmesAct")


Ator_filmes <- bind_rows(
  Ator1 %>% mutate(Fonte = "Ator1"),
  Ator2 %>% mutate(Fonte = "Ator2"),
  Ator3 %>% mutate(Fonte = "Ator3"),
) %>%
  group_by(Ator) %>%
  summarise(NumeroFilmesAct = sum(NumeroFilmesAct, na.rm = TRUE)) %>%
  as.data.frame()
```

```r
Filmes$NumeroFilmesAct <-
  Ator_filmes$NumeroFilmesAct[match(Filmes$Ator1,Ator_filmes$Ator)]+
  Ator_filmes$NumeroFilmesAct[match(Filmes$Ator2,Ator_filmes$Ator)]+
  Ator_filmes$NumeroFilmesAct[match(Filmes$Ator3,Ator_filmes$Ator)]

Filmes$NumeroFilmesAct <- Filmes$NumeroFilmesAct/3  # Calcular a média para não enviesar o modelo
```

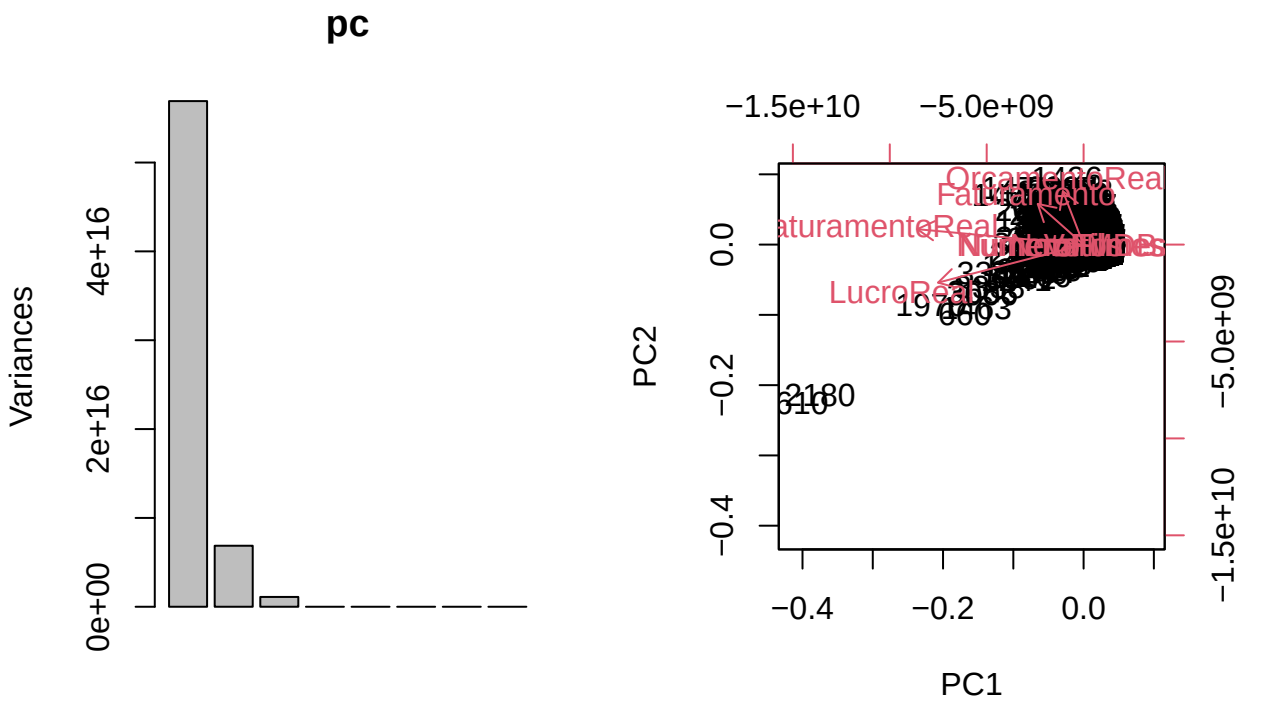
PCA

Para analisar a variabilidade, faremos uma análise dos componentes principais, também tentando reduzir a dimensionalidade do modelo

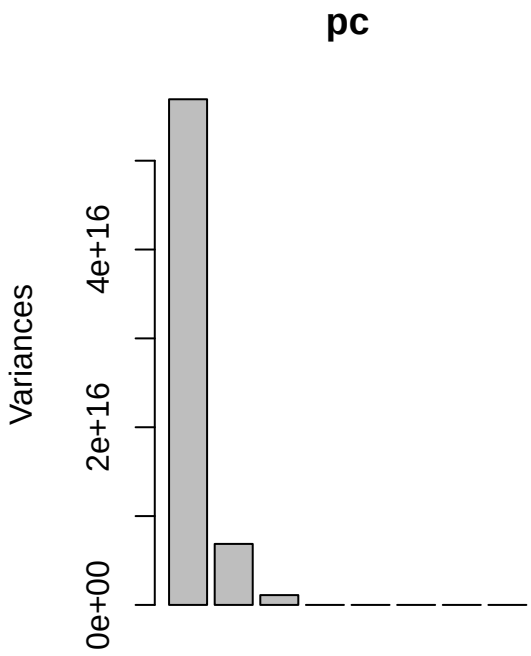
```r
# Deixando apenas as variáveis numéricas para fazer o PCA

Filmes_num <- Filmes [,c(4,14,10,19,20,21,22,23)]
pc<-prcomp(Filmes_num) ; pc
```

```
## Standard deviations (1, .., p=8):
## [1] 2.385672e+08 8.286681e+07 3.322650e+07 1.589873e+05 5.190726e+00
## [6] 1.308634e+00 9.091391e-01 1.173611e-07
##
## Rotation (n x k) = (8 x 8):
##                             PC1           PC2           PC3           PC4
## NotaIMDB        -1.020080e-09  2.356381e-10  3.586597e-09  2.790914e-06
## Faturamento     -2.022792e-01  5.166110e-01  8.319815e-01 -1.711633e-03
## Votos           -3.434514e-04  7.763844e-04  1.491703e-03  9.999985e-01
## FaturamentoReal -7.336754e-01  1.958975e-01 -3.000187e-01  4.346524e-05
## OrcamentoReal   -9.146196e-02  6.791323e-01 -4.439377e-01  1.035427e-04
## LucroReal       -6.422134e-01 -4.832348e-01  1.439189e-01 -6.007750e-05
## NumeroFilmesDir -9.494813e-10  2.769622e-09  2.269160e-09  2.095675e-06
## NumeroFilmesAct -2.049058e-09  2.021111e-08  3.480580e-10  2.698953e-06
##                             PC5           PC6           PC7           PC8
## NotaIMDB        -7.953021e-03  3.741658e-02  9.992681e-01 -5.452457e-09
## Faturamento      6.488992e-09 -5.171789e-10  1.537047e-09 -1.880871e-15
## Votos            2.873216e-06  1.778571e-06 -2.836689e-06  4.680789e-14
## FaturamentoReal  5.353108e-09  1.762779e-09 -3.013397e-09 -5.773503e-01
## OrcamentoReal    1.423308e-08 -1.380430e-09  4.365939e-09  5.773503e-01
## LucroReal       -8.416212e-09 -1.434315e-09  2.246606e-09  5.773503e-01
## NumeroFilmesDir -7.635712e-02 -9.964048e-01  3.670165e-02 -2.816859e-09
## NumeroFilmesAct -9.970488e-01  7.600935e-02 -1.078145e-02  5.277594e-10
```

```r
#Pllot de gráficos
par(mfrow=c(1,2))
screeplot(pc)
biplot(pc)
```




```
plot(pc)
```



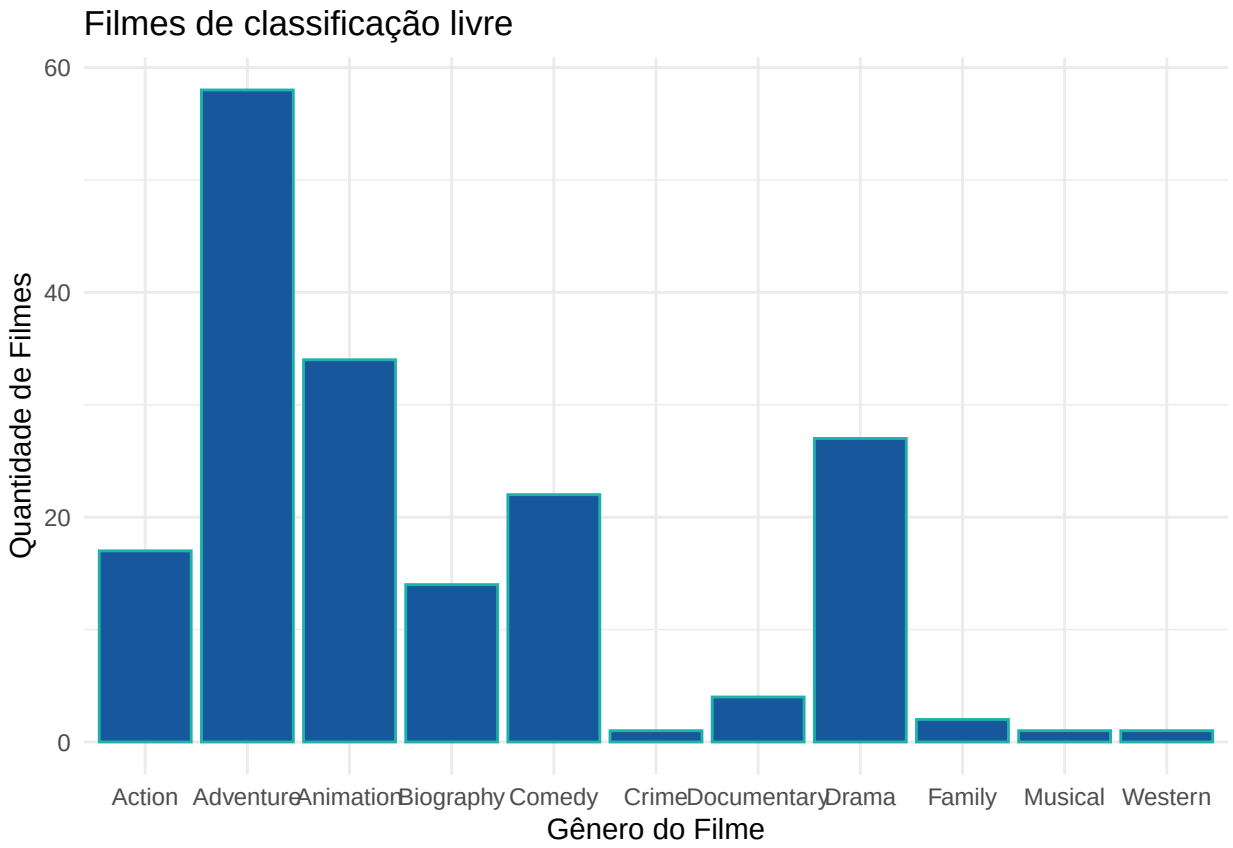
2.A Recomendação de filmes à pessoa desconhecida

Vamos analisar por gênero, priorizando os filmes sem restrições em decorrência de não saber a idade das pessoas que assistirão à obra.

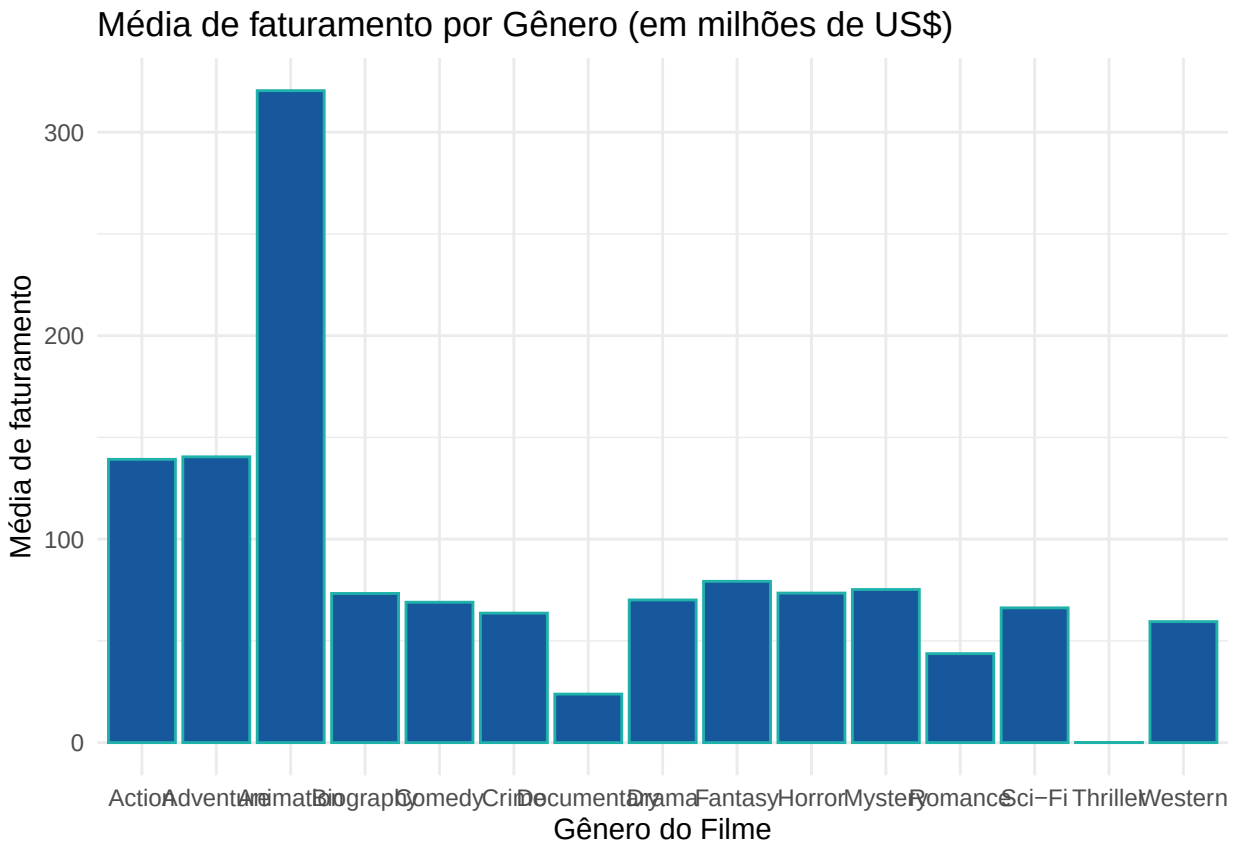
Outro fator importante é o faturamento de cada gênero, daremos preferência a gêneros de melhor bilheteria, por haver melhor adesão do público

```
# Por não conhecer a pessoa, vamos priorizar um filme sem restrição de idade
# Também um filme com bom faturamento
```

```r
#Antes disso, vamos ter que padronizar os dados dentro da coluna classificacao

Filmes$Classificacao_Padronizada <- NA

# Loop pra padronizar
for (i in 1:nrow(Filmes)) {
  if (Filmes$Classificacao[i] %in% c("Approved", "Passed", "U", "G")) {
    Filmes$Classificacao_Padronizada[i] <- "Livre"
  } else if (Filmes$Classificacao[i] %in% c("PG", "GP", "M", "TV-PG", "U/A")) {
    Filmes$Classificacao_Padronizada[i] <- "10 anos"
  } else if (Filmes$Classificacao[i] %in% c("PG-13", "UA", "16")) {
    Filmes$Classificacao_Padronizada[i] <- "12 anos"
  } else if (Filmes$Classificacao[i] %in% c("R", "TV-14", "TV-MA")) {
    Filmes$Classificacao_Padronizada[i] <- "16 anos"
  } else if (Filmes$Classificacao[i] %in% c("NC-17", "X")) {
    Filmes$Classificacao_Padronizada[i] <- "18 anos"
  } else {
    Filmes$Classificacao_Padronizada[i] <- "Sem classificação"
  }
}

# Veremos a quantidade de filmes livres
Filmes_livre <- Filmes %>%
  filter(Classificacao_Padronizada =="Livre")


ggplot(Filmes_livre, aes(x = Genero1)) +
  geom_bar(stat = "count", fill = "#17589c", color = "lightseagreen") +
  labs(
    title = "Filmes de classificação livre",
    x = "Gênero do Filme",
    y = "Quantidade de Filmes"
  ) +
  theme_minimal()
```




```
# Agora o faturamento de cada gênero, para avaliar a bilheteria

ggplot(Filmes_filtrados, aes(x = Genero1, y = FaturamentoReal/1000000)) +
  stat_summary(fun = mean, geom = "bar", fill = "#17589c", color = "lightseagreen") +
  labs(
    title = "Média de faturamento por Gênero (em milhões de US$)",
    x = "Gênero do Filme",
    y = "Média de faturamento"
  ) +
  theme_minimal()
```



Nota-se uma primazia dos filmes de animação e aventura, um filme que atende aos requisitos analisados, e ainda bem avaliado dentro do IMDb (8,9) é a obra japonesa "A Viagem de Chihiro"(2001) sendo uma boa opção para alguém que não conhecemos

Fatores que mais influenciam a renda de um filme

Para isso, calcularemos a matriz de correlações das variáveis numéricas, e verificaremos que as que mais se correlacionam com o faturamento são o número de votos e o orçamento

```
# 2.B Fatores que se relacionam com a expectativa de faturamento do filme

# para isso, vamos usar novamente o vetor Filmes_num

#Algumas variáveis de escalas diferentes estarão em escala logaritmica

Filmes_num$logfat <- log(Filmes$FaturamentoReal)
Filmes_num$logvot <- log(Filmes$Votos)
Filmes_num$logorc <- log(Filmes$OrcamentoReal)

#tirar as variáveis de maior escala que nao foram passadas para log

Filmes_num_filt <- Filmes_num[,c(1,7,8,9,10,11)]

#Matriz de correlação
Mat_cor <- cor(Filmes_num_filt)
```

Estimando o gênero do filme com base na sinopse

Faremos um modelo de LCA para verificar quais palavras mais se repetem nas sinopses de cada gênero de filme:

```
Filmes_sinopse <- Filmes_1[,c(2,6,8)]
colnames(Filmes_sinopse)
```

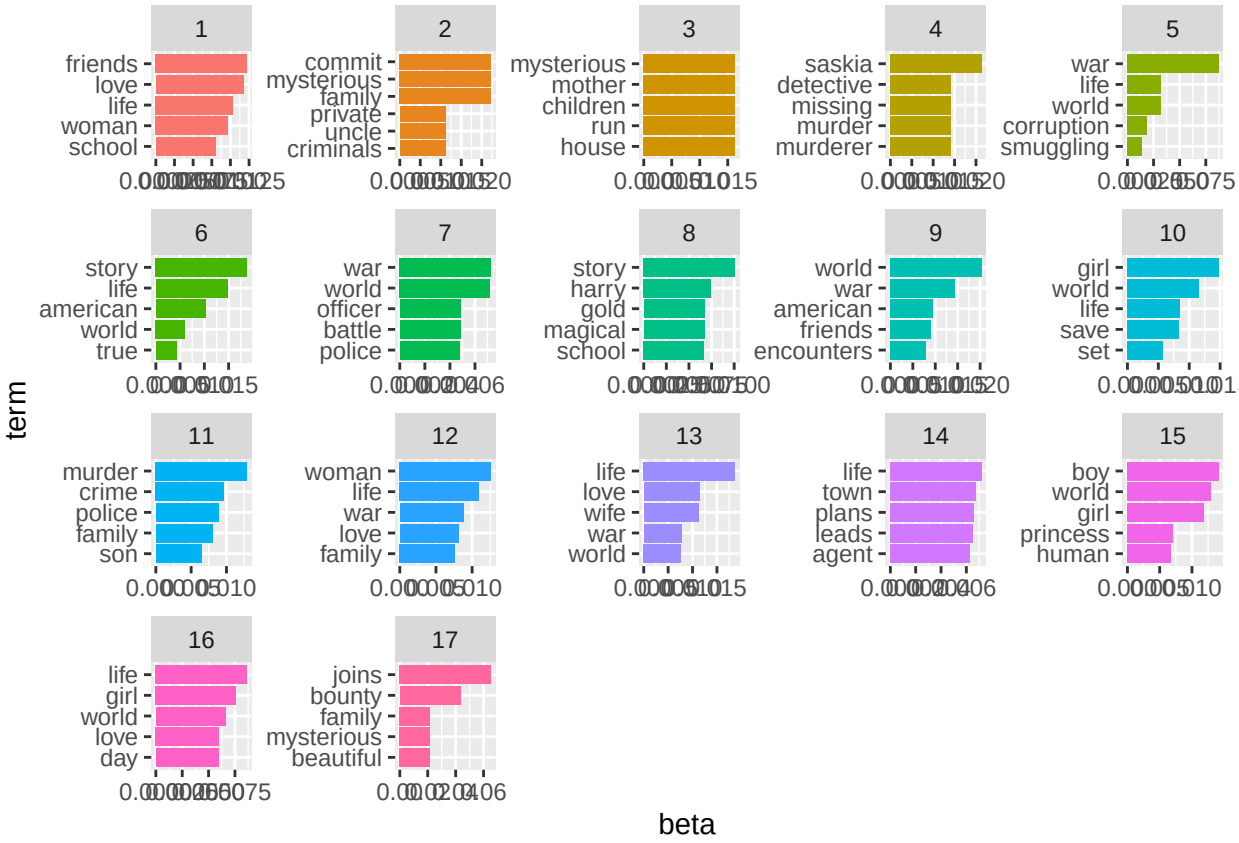
```
## [1] "Series_Title" "Genre"        "Overview"
```

```
#Separar o Gênero do subgênero
Filmes_sinopse <- Filmes_sinopse %>%
  separate(Genre, into = c("Genero1", "Genero2", "Genero3"), sep = ",", fill = "right", extra = "drop")

# Apenas Genero 1 e sinopse
colnames(Filmes_sinopse)
```

```
## [1] "Series_Title" "Genero1"      "Genero2"      "Genero3"      "Overview"
```

```
Filmes_sinopse <-Filmes_sinopse[,c(1,2,5)]

#Dividir em documentos, cada um representando um Gênero

genero_palavras <- Filmes_sinopse %>%
  group_by(Series_Title) %>%
  mutate(Overview_count = cumsum(!is.na(Overview))) %>%
  ungroup() %>%
  filter(Overview_count > 0) %>%
  unite(Document, Genero1, Overview_count, sep = "_")

# Separar por palavra
sinopse_palavras <- genero_palavras %>%
  unnest_tokens(word, Overview)

# Contagem de palavras
cont_palavras <- sinopse_palavras %>%
  anti_join(stop_words) %>%
  count(Document, word, sort = TRUE)

# Deixar em DTM
palavras_dtm <- cont_palavras %>%
  cast_dtm(Document, word, n)

# Definir o valor de K
k <- nrow(Genero_filmes)

#Modelo LDA
palavras_lda <- LDA(palavras_dtm, k = k, control = list(seed = 1234))

# Separar por tópicos
topicos_palavras <- tidy(palavras_lda, matrix = "beta")

# Separar os de maior ocorrência
top_palavras <- topicos_palavras %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
```

```
  ungroup() %>%
  arrange(topic, -beta)

#plotar gráfico

top_palavras %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
```

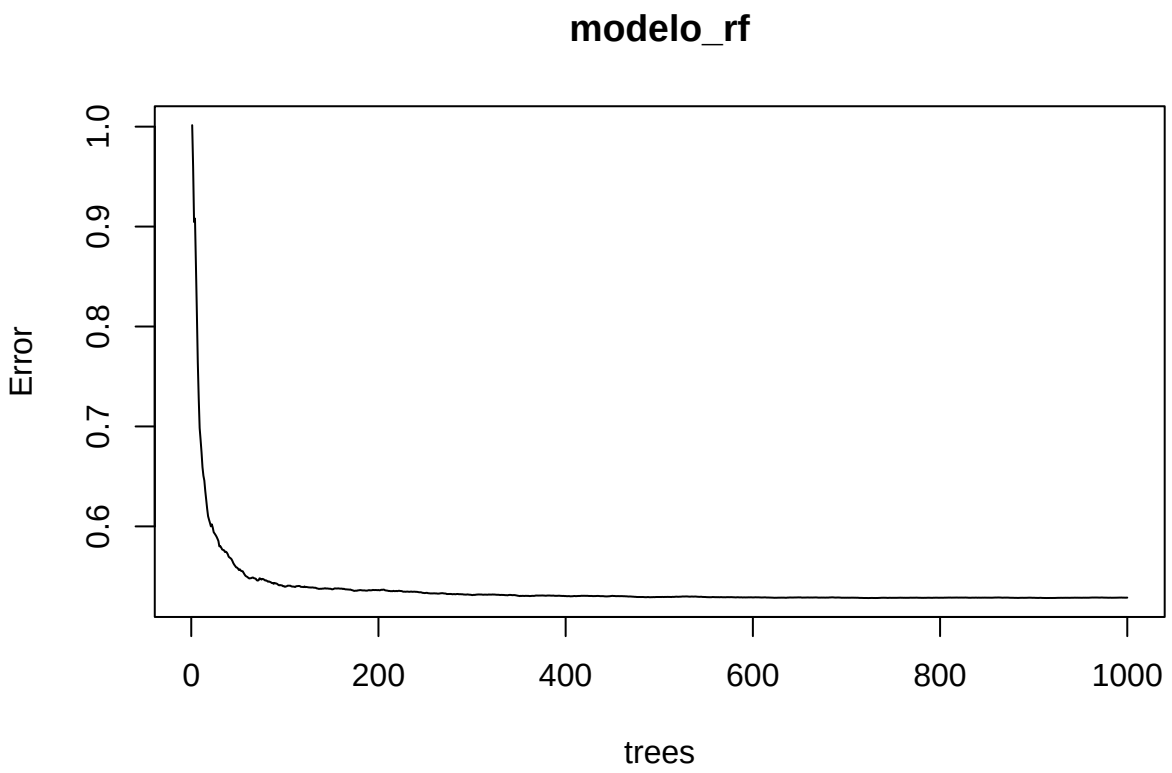


Verificamos um certo agrupamento de algumas palavras, e podemos fazer algumas inferências, como por exemplo, o tópico 11 está relacionados a filmes sobre crimes, então, sim, pode - se inferir o gênero com base na sinopse

#Previsão de nota do IMDb com base em Random Forest

```
# Usaremos um novo data frame para não "contaminar"o antigo
Filmes_rf <- Filmes[,c(2,4,5,10,19,20,21,22,23,24)]

# Vamos transformar todas as variáveis categóricas em fatores
Filmes_rf$Ano <- as.integer(Filmes_rf$Ano)
Filmes_rf$Genero1 <- as.factor(Filmes_rf$Genero1)
Filmes_rf$Classificacao_Padronizada <-as.factor(Filmes$Classificacao_Padronizada)
Filmes_rf <- na.omit(Filmes_rf)
# Fixar semente
```

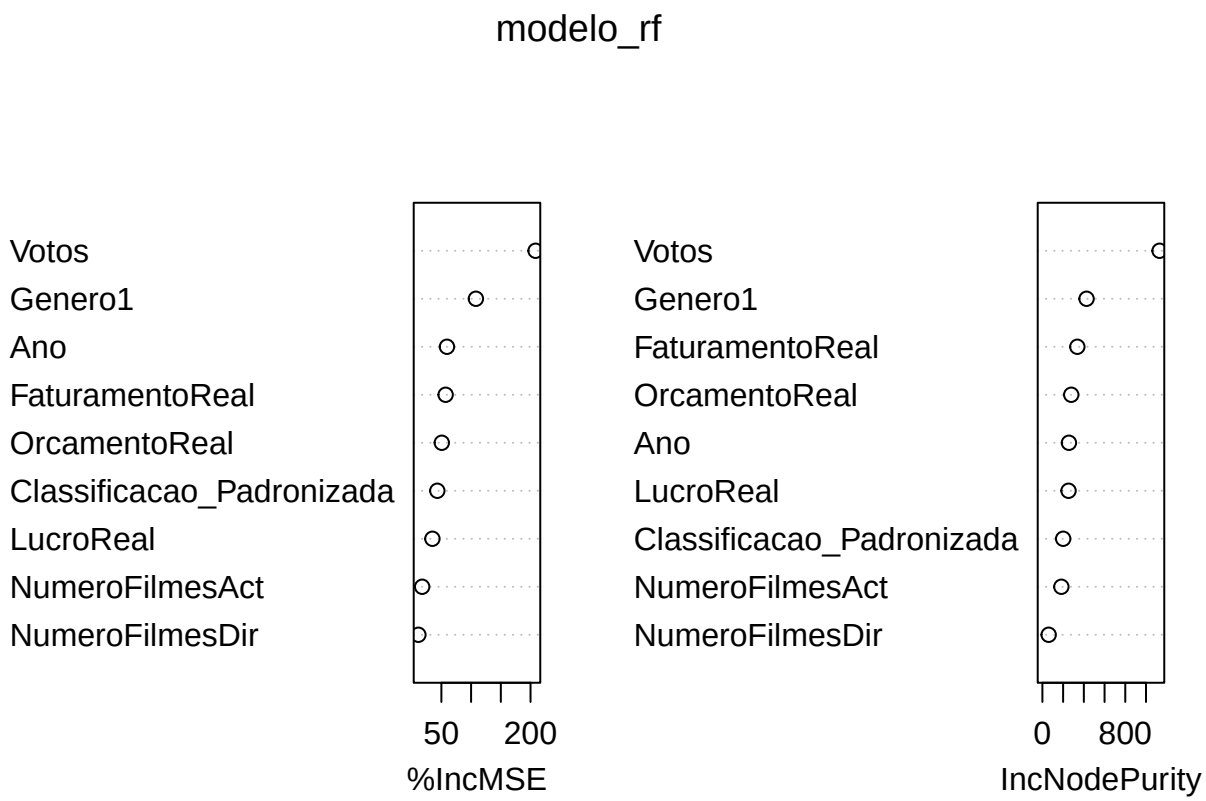
```r
set.seed(123)

# Definindo treino e controle
samp <- sample(nrow(Filmes_rf), 0.8 * nrow(Filmes_rf))
train <- Filmes_rf[samp,]
test <- Filmes_rf [-samp,]

modelo_rf <- randomForest(NotaIMDB ~., data = train,
                          ntree = 1000, mtry = 5, importance = TRUE)


modelo_rf2 <- predict(modelo_rf,test)

plot(modelo_rf)
```




```r
importance(modelo_rf)
```

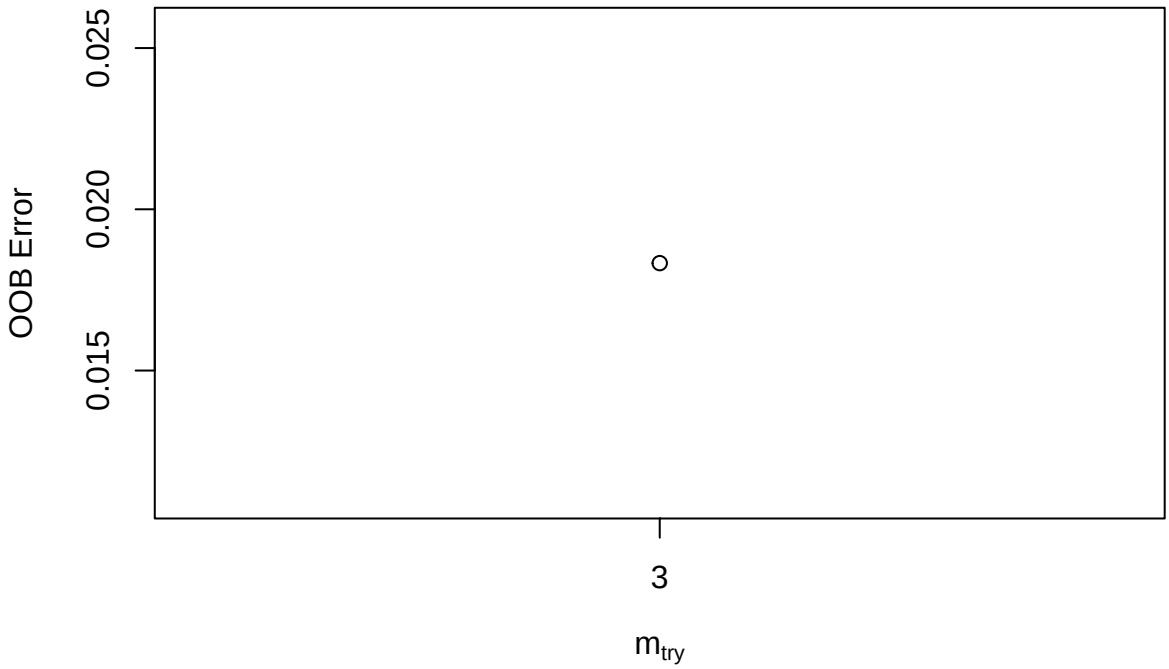
```
##                             %IncMSE IncNodePurity
## Ano                        59.38795     256.28588
## Genero1                   108.08608     426.30207
## Votos                     208.49123    1131.20194
## FaturamentoReal            57.15771     337.19856
## OrcamentoReal              50.68501     278.52054
## LucroReal                  34.90310     252.34905
## NumeroFilmesDir            11.32895      60.55439
## NumeroFilmesAct            17.67864     183.16140
## Classificacao_Padronizada  43.16011     200.62555
```

```r
varImpPlot(modelo_rf)
```

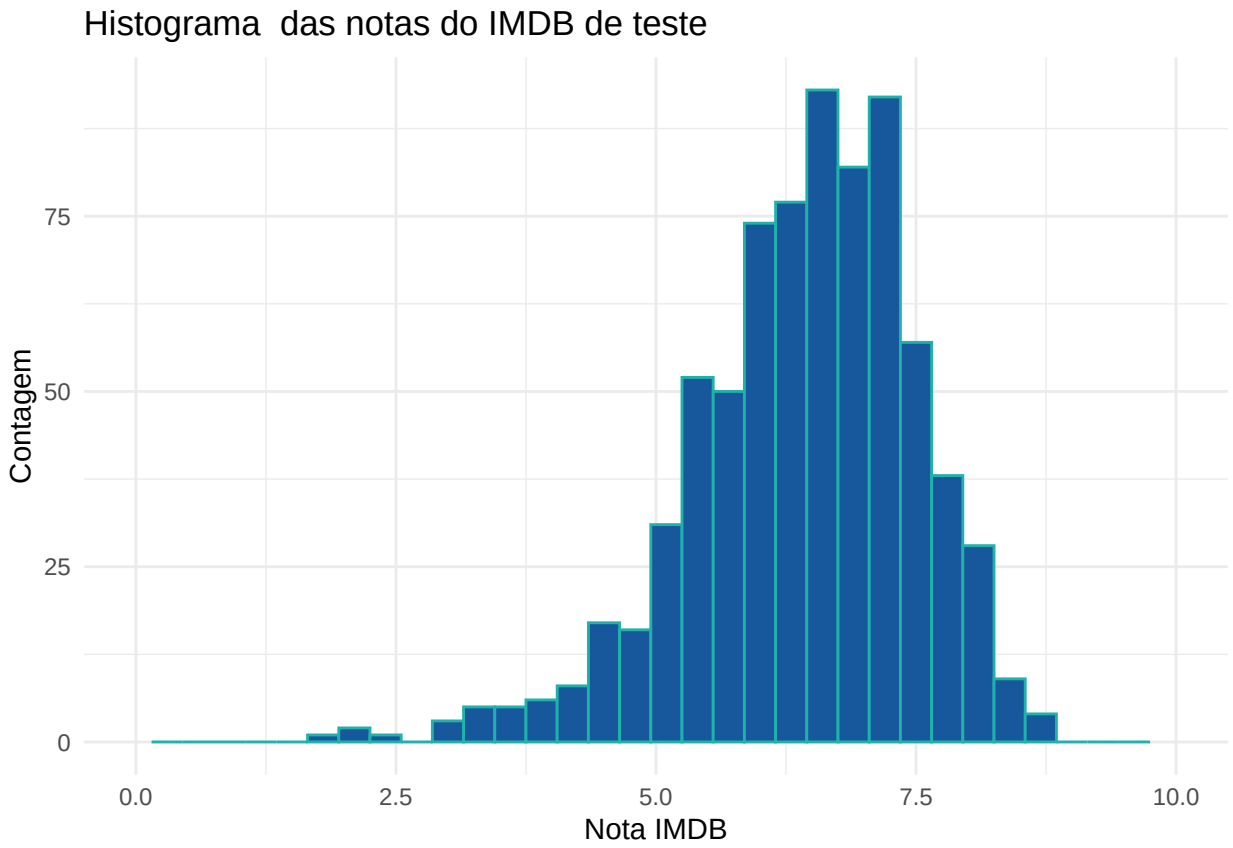



```r
mtry <- tuneRF(train, train$NotaIMDB, ntreeTry = 400,
               stepFactor = 1, improve = 0.05, trace = TRUE, plot = TRUE)
```

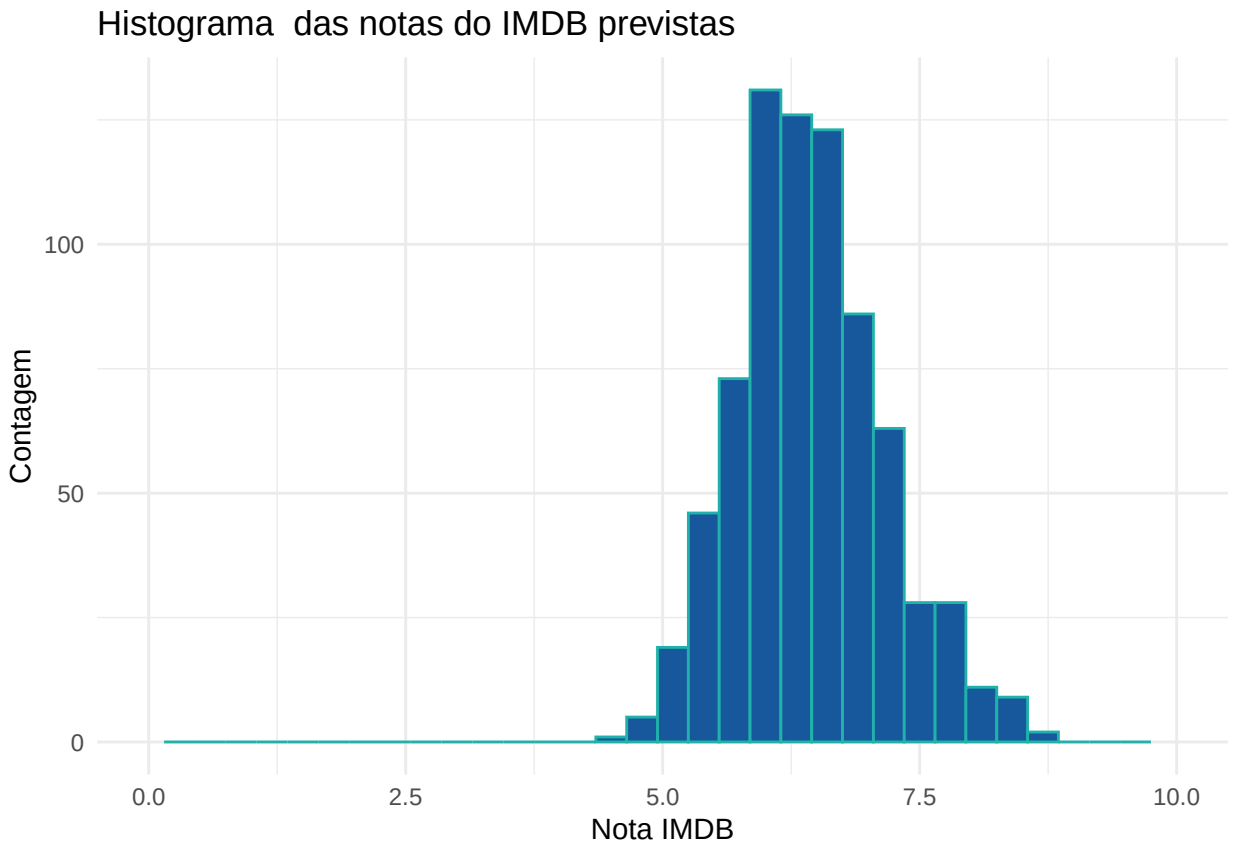
```
## mtry = 3  OOB error = 0.0183306
## Searching left ...
## Searching right ...
```




```r
previsoes_rf <- data.frame(NotaIMDB_Prevista = modelo_rf2)

#Gerar histogramas para comparar os valores de teste com o de rf2



ggplot(data = test, aes(x = NotaIMDB))+
  geom_histogram(binwidth = 0.3, fill = "#17589c", color = "lightseagreen")+ #Tentativa de imitar as co
  xlim(0,10)+
  labs(
    title = "Histograma  das notas do IMDB de teste",
    x = "Nota IMDB",
    y = "Contagem")+
  theme_minimal()
```




```
ggplot(data = previsoes_rf, aes(x = NotaIMDB_Prevista))+
  geom_histogram(binwidth = 0.3, fill = "#17589c", color = "lightseagreen")+ #Tentativa de imitar as co
  xlim(0,10)+
  labs(
    title = "Histograma  das notas do IMDB previstas",
    x = "Nota IMDB",
    y = "Contagem")+
  theme_minimal()
```



Com base nos histogramas analisados, verificamos uma boa aderência do modelo sobre os dados observados

Prever a nota do IMDb de um filme com Random Forest

```
#########################
# 4. Prever a nota do IMDB de um filme
# # {'Series_Title': 'The Shawshank Redemption',
# 'Released_Year': '1994',
# 'Certificate': 'A',
# 'Runtime': '142 min',
# 'Genre': 'Drama',
# 'Overview': 'Two imprisoned men bond over a number of years, finding solace and eventual redemption t
# 'Meta_score': 80.0,
# 'Director': 'Frank Darabont',
# 'Star1': 'Tim Robbins',
# 'Star2': 'Morgan Freeman',
# 'Star3': 'Bob Gunton',
# 'Star4': 'William Sadler',
# 'No_of_Votes': 2343110,
# 'Gross': '28,341,469'}


# Vamos criar um novo dataframe para esse filme novo

# Para facilitar, copiaremos o formato dos dados da coluna de treino
Filmes_rf_2 <- train[1, ]
Filmes_rf_2[1, ] <- NA
```

```r
#Inserir os dados do filme Um sonho de liberdade
Filmes_rf_2$Ano <- as.integer(1994)
Filmes_rf_2$Genero1 <- factor("Drama", levels = levels(train$Genero1))
Filmes_rf_2$Votos <- 2343110
Filmes_rf_2$FaturamentoReal <- 28341469/0.47432709
Filmes_rf_2$OrcamentoReal <- 25000000/0.47432709
Filmes_rf_2$LucroReal <- Filmes_rf_2$FaturamentoReal - Filmes_rf_2$OrcamentoReal
Filmes_rf_2$NumeroFilmesDir <- 2
Filmes_rf_2$NumeroFilmesAct <- 13.7
Filmes_rf_2$Classificacao_Padronizada <- factor("16 anos", levels = levels(train$Classificacao_Padroniza


#Previsão
previsao_filme <- predict(modelo_rf, newdata = Filmes_rf_2)
```

A nota de 8,8 se aproxima dos 9,3 no IMDb, o que confirma a eficácia dessa técnica para esse tipo de análise