**Análise - Inteligência Computacional - COC 361**

**Lucas Rolim e Anderson Barbosa**

30 de outubro de 2016

## Descrição do problema

Problema: É possível prever a nota IMDB de um filme antes mesmo de ele ser lançado?

Essa será a principal pergunta que irá guiar o desenvolvimento de uma análise de mais de 5000 filmes presentes no site do [IMDB](file:///C:\Users\Lucas\Desktop\inteligencia_computacional\Análise%20da%20nota%20IMDB%20de%20Filmes\IMDB-Movies-Analysis\www.imdb.com). Utilizando técnicas de aprendizado de máquina e estatística nós iremos não só tentar criar um **modelo de aproximação de funções** que seja capaz de responder à pergunta anterior de maneira satistatória, como também iremos criar visualizações que nos permitam entender as relações entre essas variáveis.

Esse problema é importante não só para ajudar espectadores em geral a reduzir sua taxa de decepção ao ir ao cinema, mas também se caracteriza como de suma importância para a indústria cinematrográfica em si. Filmes, antes de tudo, são investimentos. Conseguir prever se um filme será um fiasco ou um sucesso de bilheteria pode poupar milhões as produtoras de filmes e investidores.

*Para os que desconhecem, o IMDB.com é a principal referência mundial na avaliação da qualidade de filmes. O site conta com dezenas de milhares de filmes e também com avaliações e críticas produzidas por centenas de milhares de usuários e críticos profissionais*

## Pesquisa Bibliográfica

Os dados utilizados nessa análise foram minerados por Chuan Sun no site do IMDB e depois disponibilizados na plataforma Kaggle. Tanto o tópico desse conjunto de dados quanto uma explicação em detalhes sobre o processo de mineração podem ser encontrados no seguinte link:

* [IMDB 5000 Movie Dataset](https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset)

## Descrição dos dados

O conjunto de dados possui informação de **5043 filmes**, lançados ao longo dos últimos 100 anos em 66 países. Ainda, conta com 2399 nomes de diretores e centenas de nomes de atores atrizes.

**As 28 variáveis disponíveis são:**

| **Variável** | **Significado** |
| --- | --- |
| color | Filme colorido ou não |
| num\_voted\_users | Número de usuários que deram uma classificação ao filme no site do IMDB |
| cast\_total\_facebook\_likes | Número total de curtidas de toda página do elenco no Facebook |
| facenumber\_in\_poster | Número de faces humanas que aparecem no poster do filme |
| plot\_keywords | Palavras-chave da trama |
| movie\_imdb\_link | Endereço da página do filme no IMDB |
| num\_user\_for\_reviews | Número de críticas de usuários do IMDB |
| language | Idioma do filme |
| country | País em que o filme foi produzido |
| content\_rating | Avaliação do conteúdo |
| budget | Orçamento do filme |
| title\_year | Ano de lançamento do filme |
| imdb\_score | Nota do filme no IMDB |
| aspect\_ratio | Não conseguimos identificar \* |
| movie\_facebook\_likes | Número de curtidas da página do filme no Facebook |
| director\_name | Nome do diretor do filme |
| num\_critic\_for\_reviews | Número de críticas externas ao IMDB |
| actor\_(1,2,3)\_name | Nomes dos três principais atores do filme |
| movie\_title | Título do filme |
| actor\_(1,2,3)\_facebook\_likes | Números de curtidas das páginas dos três principais atores do filme no Facebook |
| director\_facebook\_likes | Número de curtidas da página do diretor no Facebook |
| genres | Gênero do filme |
| gross | lucro bruto de bilheteria obtido pelo filme |
| duration | Duração do filme em minutos |

## Ferramentas utilizadas

A principal ferramenta utilizada nessa análise foi a linguagem R e o ambiente RStudio. Ademais, as seguintes bibliotecas foram utilizadas.

* rMarkdown
* ggplot2
* caTools
* randomForest
* dplyr
* rpart
* rpart.plot
* corrplot

library(dplyr)  
library(corrplot)  
library(ggplot2)  
library(caTools)  
library(rpart)  
library(rpart.plot)  
library(randomForest)

O formato do conjunto de cados que necessitamos ler é *.csv*, o qual já possui suporte nativo do próprio R. O formato desse relatório, por sua vez, é *.rmd*, um formato que possibilita uma explicação mais bem elaborada do processo executado.

Por fim, a linguagem R foi escolhida por ser a que possui bibliotecas no estado da arte no que tange a tratamentos estatísticos e criação de modelos de regressão, classificação e agrupamentos. Essa linguagem é uma "sucessora" da linguagem S e conta com a ampla adoção acadêmica e diversas bibliotecas otimizadas.

## Pré-processamento

O pré-processamento de dados é um passo importante no processo de mineração de dados. Os métodos de coleta de dados, em geral, não são controlados rigorosamente, o que resulta em um conjunto de dados com valores fora do intervalo, combinações impossíveis de dados, valores ausentes etc. Desta forma, o pré-processamento é necessário para evitar que a análise dos dados produza resultados enganosos. Ela envolve limpeza de dados, normalização, transformação de variáveis, entre outros. O resultado final do pré-processamento de dados é o conjunto de treinamento final, que será utilizado para o ajuste do modelo.

### Leitura dos Dados

Em primeiro lugar, vamos dar uma primeira olhada no formato do nosos conjunto de dados. Vamos verificar quais são as variáveis e seus tipos.

imdb\_data = read.csv("movie\_metadata.csv",stringsAsFactors = FALSE)  
str(imdb\_data)

## 'data.frame': 5043 obs. of 28 variables:  
## $ color : chr "Color" "Color" "Color" "Color" ...  
## $ director\_name : chr "James Cameron" "Gore Verbinski" "Sam Mendes" "Christopher Nolan" ...  
## $ num\_critic\_for\_reviews : int 723 302 602 813 NA 462 392 324 635 375 ...  
## $ duration : int 178 169 148 164 NA 132 156 100 141 153 ...  
## $ director\_facebook\_likes : int 0 563 0 22000 131 475 0 15 0 282 ...  
## $ actor\_3\_facebook\_likes : int 855 1000 161 23000 NA 530 4000 284 19000 10000 ...  
## $ actor\_2\_name : chr "Joel David Moore" "Orlando Bloom" "Rory Kinnear" "Christian Bale" ...  
## $ actor\_1\_facebook\_likes : int 1000 40000 11000 27000 131 640 24000 799 26000 25000 ...  
## $ gross : int 760505847 309404152 200074175 448130642 NA 73058679 336530303 200807262 458991599 301956980 ...  
## $ genres : chr "Action|Adventure|Fantasy|Sci-Fi" "Action|Adventure|Fantasy" "Action|Adventure|Thriller" "Action|Thriller" ...  
## $ actor\_1\_name : chr "CCH Pounder" "Johnny Depp" "Christoph Waltz" "Tom Hardy" ...  
## $ movie\_title : chr "AvatarÂ " "Pirates of the Caribbean: At World's EndÂ " "SpectreÂ " "The Dark Knight RisesÂ " ...  
## $ num\_voted\_users : int 886204 471220 275868 1144337 8 212204 383056 294810 462669 321795 ...  
## $ cast\_total\_facebook\_likes: int 4834 48350 11700 106759 143 1873 46055 2036 92000 58753 ...  
## $ actor\_3\_name : chr "Wes Studi" "Jack Davenport" "Stephanie Sigman" "Joseph Gordon-Levitt" ...  
## $ facenumber\_in\_poster : int 0 0 1 0 0 1 0 1 4 3 ...  
## $ plot\_keywords : chr "avatar|future|marine|native|paraplegic" "goddess|marriage ceremony|marriage proposal|pirate|singapore" "bomb|espionage|sequel|spy|terrorist" "deception|imprisonment|lawlessness|police officer|terrorist plot" ...  
## $ movie\_imdb\_link : chr "http://www.imdb.com/title/tt0499549/?ref\_=fn\_tt\_tt\_1" "http://www.imdb.com/title/tt0449088/?ref\_=fn\_tt\_tt\_1" "http://www.imdb.com/title/tt2379713/?ref\_=fn\_tt\_tt\_1" "http://www.imdb.com/title/tt1345836/?ref\_=fn\_tt\_tt\_1" ...  
## $ num\_user\_for\_reviews : int 3054 1238 994 2701 NA 738 1902 387 1117 973 ...  
## $ language : chr "English" "English" "English" "English" ...  
## $ country : chr "USA" "USA" "UK" "USA" ...  
## $ content\_rating : chr "PG-13" "PG-13" "PG-13" "PG-13" ...  
## $ budget : num 2.37e+08 3.00e+08 2.45e+08 2.50e+08 NA ...  
## $ title\_year : int 2009 2007 2015 2012 NA 2012 2007 2010 2015 2009 ...  
## $ actor\_2\_facebook\_likes : int 936 5000 393 23000 12 632 11000 553 21000 11000 ...  
## $ imdb\_score : num 7.9 7.1 6.8 8.5 7.1 6.6 6.2 7.8 7.5 7.5 ...  
## $ aspect\_ratio : num 1.78 2.35 2.35 2.35 NA 2.35 2.35 1.85 2.35 2.35 ...  
## $ movie\_facebook\_likes : int 33000 0 85000 164000 0 24000 0 29000 118000 10000 ...

### Normalização dos Dados e escolha de variáveis

Algumas das variáveis de texto do conjunto de dados serão convertidas em variáveis numéricas através de um critério de normalização definido pelas seguintes regras:

* Se o filme foi produzido no EUA a variável **country** será 1, caso contrário será 0;
* Se o filme foi produzido em inglês a variável **language** será 1, caso contrário será 0.
* Criação da variável **directorIMDB**, que terá o número de filmes já feito pelo diretor.
* Criação da variável **directorMovies**, que terá o número de filmes já feito pelo diretor do filme.

Essas regras de normalização foram definidas pensando no fato da indústria cinematográfica americana ser a mais bem desenvolvida e com maiores orçamentos do mundo, além de produzir os maiores sucessos das últimas décadas. Logo, espera-se que filmes americanos tenham uma probabilidade maior de pontuar melhor na escala IMDB de maneira geral.

Por fim, passaremos a considerar apenas os valores numéricos do modelo e também desconsideraremos as entradas com valores ausentes.

*A criação das variáveis directorIMDB e directorMovies será feita após o particionamento do conjunto de dados. Esse cuidado é necessário para que não sejam levados em conta no cálculo da média do IMDB do diretor os filmes que estão no conjunto de testes*

# Essa versão do directorMovies será usada na Análise Exploratória  
directorMovies <- summarise(group\_by(imdb\_data,director\_name),mean=mean(imdb\_score),number = n())  
directorMovies <- directorMovies[2:nrow(directorMovies),]  
  
imdb\_data$language[imdb\_data$language !="English"] = 0  
imdb\_data$language[imdb\_data$language =="English"] = 1  
imdb\_data$language <- as.numeric(imdb\_data$language)  
imdb\_data$country[imdb\_data$country != "USA"] = 0  
imdb\_data$country[imdb\_data$country == "USA"] = 1  
imdb\_data$country <- as.numeric(imdb\_data$country)  
  
temp <- imdb\_data$director\_name  
  
imdb\_data <- imdb\_data[sapply(imdb\_data,is.numeric)]  
NoTNAIndex = complete.cases(imdb\_data)  
imdb\_data <- imdb\_data[which(NoTNAIndex),]  
temp= temp[which((NoTNAIndex))]

Nesse ponto é importante notar que, com a **exclusão de filmes que possuiam variáveis ausentes**, nosso conjunto de dados foi reduzido para um universo de 3801 amostras. Ou seja, **2059 registros possuiam algum valor ausente**, um total de 40% dos dados. Essa redução foi considerável, mas se faz necessária e o volume de dados restante aparenta ser grande o suficiente para não comprometer a análise.

Por fim, após a limpezadas variáveis para apenas variáveis numéricas, restaram 18 variáveis. Lembrando que no momento de aplicação dos modelos serão adiconadas mais duas, *DirectorMovie e DirecotrIMDB*, somando um total de 20 variáveis a serem trabalhadas.

## Análise Exporatória

## Estatísticas básicas:

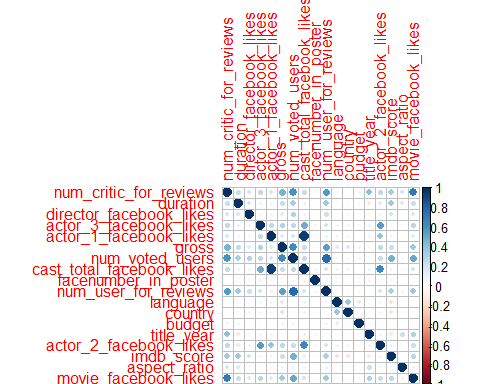
## num\_critic\_for\_reviews duration director\_facebook\_likes  
## Min. : 1.0 Min. : 37.0 Min. : 0.0   
## 1st Qu.: 75.0 1st Qu.: 96.0 1st Qu.: 11.0   
## Median :137.0 Median :106.0 Median : 62.0   
## Mean :165.8 Mean :110.2 Mean : 798.6   
## 3rd Qu.:223.0 3rd Qu.:120.0 3rd Qu.: 234.0   
## Max. :813.0 Max. :330.0 Max. :23000.0   
## actor\_3\_facebook\_likes actor\_1\_facebook\_likes gross   
## Min. : 0.0 Min. : 0 Min. : 162   
## 1st Qu.: 187.0 1st Qu.: 736 1st Qu.: 7689458   
## Median : 433.0 Median : 1000 Median : 29200000   
## Mean : 763.8 Mean : 7673 Mean : 52005759   
## 3rd Qu.: 690.0 3rd Qu.: 13000 3rd Qu.: 66466372   
## Max. :23000.0 Max. :640000 Max. :760505847   
## num\_voted\_users cast\_total\_facebook\_likes facenumber\_in\_poster  
## Min. : 5 Min. : 0 Min. : 0.000   
## 1st Qu.: 18915 1st Qu.: 1865 1st Qu.: 0.000   
## Median : 53028 Median : 3969 Median : 1.000   
## Mean : 104677 Mean : 11415 Mean : 1.379   
## 3rd Qu.: 126916 3rd Qu.: 16143 3rd Qu.: 2.000   
## Max. :1689764 Max. :656730 Max. :43.000   
## num\_user\_for\_reviews language country budget   
## Min. : 1.0 Min. :0.000 Min. :0.0000 Min. :2.180e+02   
## 1st Qu.: 107.0 1st Qu.:1.000 1st Qu.:1.0000 1st Qu.:1.000e+07   
## Median : 208.0 Median :1.000 Median :1.0000 Median :2.500e+07   
## Mean : 333.6 Mean :0.954 Mean :0.7906 Mean :4.585e+07   
## 3rd Qu.: 397.0 3rd Qu.:1.000 3rd Qu.:1.0000 3rd Qu.:5.000e+07   
## Max. :5060.0 Max. :1.000 Max. :1.0000 Max. :1.222e+10   
## title\_year actor\_2\_facebook\_likes imdb\_score aspect\_ratio   
## Min. :1920 Min. : 0 Min. :1.600 Min. : 1.18   
## 1st Qu.:1999 1st Qu.: 374 1st Qu.:5.900 1st Qu.: 1.85   
## Median :2005 Median : 680 Median :6.600 Median : 2.35   
## Mean :2003 Mean : 2003 Mean :6.466 Mean : 2.11   
## 3rd Qu.:2010 3rd Qu.: 975 3rd Qu.:7.200 3rd Qu.: 2.35   
## Max. :2016 Max. :137000 Max. :9.300 Max. :16.00   
## movie\_facebook\_likes  
## Min. : 0   
## 1st Qu.: 0   
## Median : 224   
## Mean : 9262   
## 3rd Qu.: 11000   
## Max. :349000

**Legenda:**

|  |  |
| --- | --- |
| Métrica | Significado |
| Min.: | Valor mínimo registrado da variável |
| 1st Qu. | Percentil 25, valor abaixo do qual se encontram 25% dos registros |
| Median: | Percentil 50, ou mediana |
| 3rd Qu.: | Percentil 75 |
| Max.: | Valor máximo registrado da variável |

### Matriz de Correlação

corrplot(cor(imdb\_data))

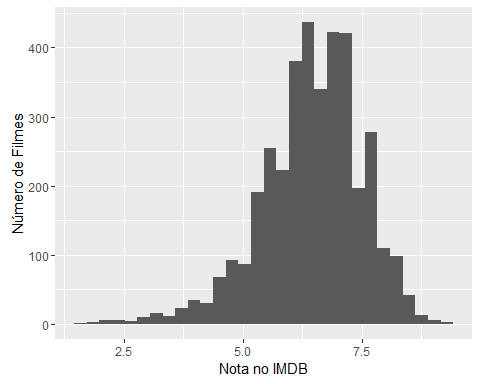


Analisando esta matriz de correlação, é possível ver que as seguintes variáveis apresentam forte correlação positiva: **cast\_total\_facebook\_likes** e **actor\_1\_facebook\_likes**, o que já era esperado, pois o conjunto de curtidas de um ator é um subconjunto do conjunto de curtidas totais do elenco do filme. **movie\_facebook\_likes** e **num\_critic\_for\_reviews**, indicando que o número de críticas externas que o filme recebe contribui para o seu número de curtidas no facebook **num\_voted\_users e num\_user\_for\_reviews**, indicando que quanto mais usuários dão uma nota ao filme, mais usuários escrevem um crítica para ele; **Cast\_total\_facebook\_likes** e **actor\_2\_facebook\_likes**, que, assim como a primeira correlação, já era esperada; e **gross** e **num\_voted\_users**, indicando que filmes que geram uma receita maior tendem a receber mais votos de usuários do IMDB. Outra coisa interessante a ser observada, é que o orçamento (**budget**) do filme não apresenta correlação com nenhuma variável, indicando que a quantidade de dinheiro investida no filme não influencia em como o filme será recebido pelo público.

### Histogramas

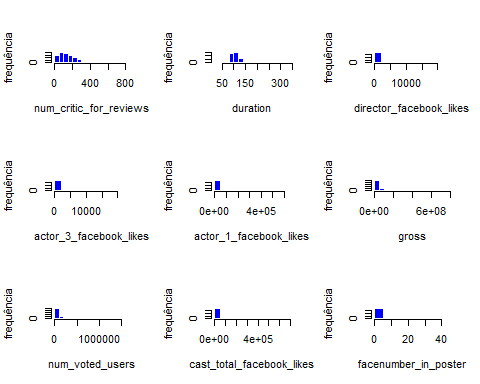
**Notas IMDB dos filmes**

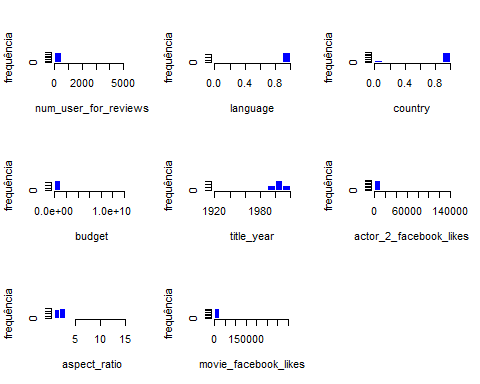
ggplot(imdb\_data,aes(x=imdb\_score)) + geom\_histogram() + xlab("Nota no IMDB") + ylab("Número de Filmes")



A partir do histograma acima, é possível observar que a variável imdb\_score segue uma distribuição normal com média aproximadamente igual 6,5. O valor que se esperava para essa média seria algo em torno de 5,0, uma vez que as notas variam de 0 a 10. Uma possível explicação para a média ter sido diferente da esperada é que o número de registros não é grande o suficiente ou que muito poucos filmes tendem a ser classificados com a nota menor que 2.

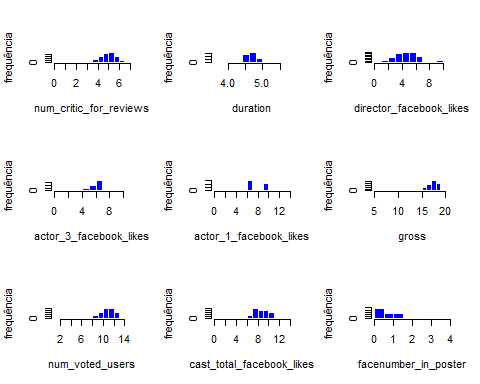
**Outros Histogramas**

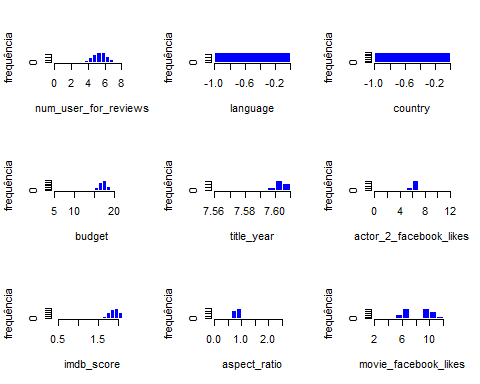




**Transformação de Variáveis**

Sabemos que nosos modelo de Regressão funciona de maneira mais precisa quando temos variáveis que seguem uma Distriguição Normal. Dessa forma, utilizaremos os histograma para analisar algumas variáveis que podem ser transformadas de modo a aproximar melhor esse tipo de distribuição:





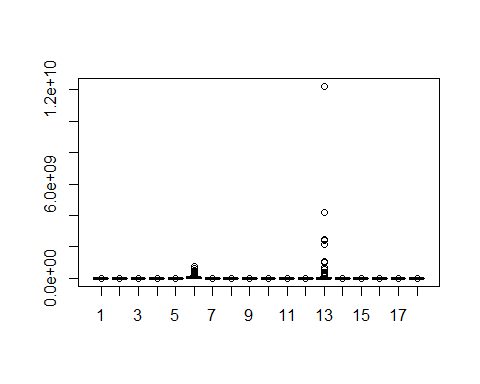
Ao analisar essa variáveis percebemos que de fato a distribuição de muitas delas podem ser aproximadas por uma normal se aplicarmos o logaritmo. O problema que encontramos nesse caso é o fato de muitos valores *"infinito"* serem gerados. Logo, no momento, decidimos usar a análise com log apenas para os outliers, e não para os modelos.

### Outliers

Outliers, ou valores aberrantes, são registros que apresentam uma distância muito grande dos demais. A importância da detecção e remoção destes reside no fato de que a presença deles pode levar a resultados enganosos na análise de dados.

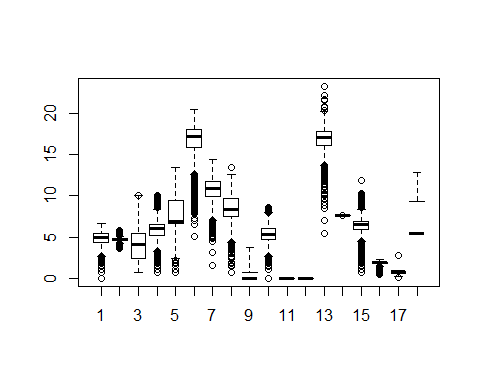
**Boxplots das variáveis**

boxplot(imdb\_data, names=c('1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18'))



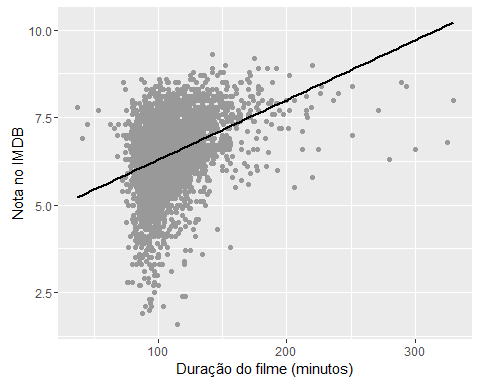
Para observar melhor os outliers das variáveis, plotamos os boxplots dos logaritmos das variáveis.

boxplot(log(imdb\_data), names=c('1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18'))



### Relação entre a nota IMDB e a duração do filme

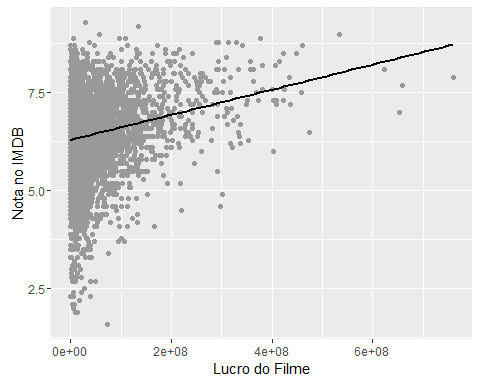
ggplot(imdb\_data,aes(x=duration, y=imdb\_score)) + geom\_point(colour="grey60") +  
 stat\_smooth(method=lm, se=FALSE, colour="black") + ylab("Nota no IMDB") + xlab("Duração do filme (minutos)")



Este scatterplot mostra que a duração do filme e sua nota no IMDB apresentam uma fraca correlação positiva.

### Relação entre a nota IMDB e o lucro

ggplot(imdb\_data,aes(x=gross, y=imdb\_score)) + geom\_point(colour="grey60") +  
 stat\_smooth(method=lm, se=FALSE, colour="black") + ylab("Nota no IMDB") + xlab("Lucro do Filme")

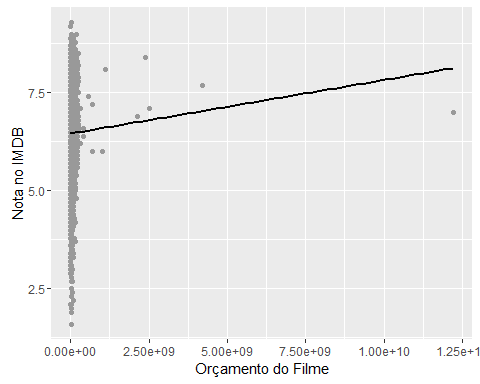


Este scatterplot, assim como o anterior, mostra que o lucro obtido pelo filme e sua nota no IMDB apresentam uma fraca correlação positiva.

### Relação entre a nota IMDB e orçamento

**Com outliers**

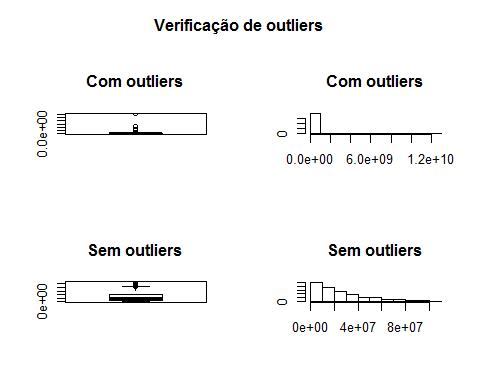
ggplot(imdb\_data,aes(x=budget, y=imdb\_score)) + geom\_point(colour="grey60") +  
 stat\_smooth(method=lm, se=FALSE, colour="black") + ylab("Nota no IMDB") + xlab("Orçamento do Filme")



No scatterplot acima, observamos que os registros não estão distribuidos de maneira comportada, o que se deve à presença de outliers. Por isso, removeremos os outliers relacionados a variável budget (orçamento) e plotar novamente o gráfico.

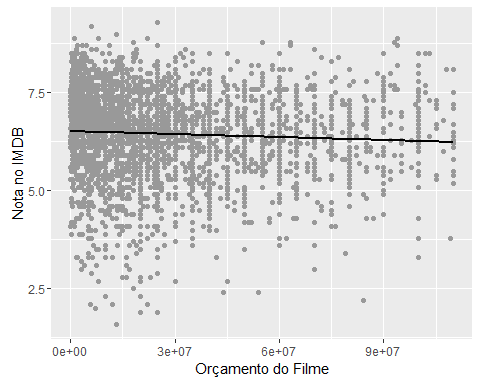
**Sem outliers**

outlierKD <- function(dt, var) {  
 var\_name <- eval(substitute(var),eval(dt))  
 na1 <- sum(is.na(var\_name))  
 m1 <- mean(var\_name, na.rm = T)  
 par(mfrow=c(2, 2), oma=c(0,0,3,0))  
 boxplot(var\_name, main="Com outliers")  
 hist(var\_name, main="Com outliers", xlab=NA, ylab=NA)  
 outlier <- boxplot.stats(var\_name)$out  
 mo <- mean(outlier)  
 var\_name <- ifelse(var\_name %in% outlier, NA, var\_name)  
 boxplot(var\_name, main="Sem outliers")  
 hist(var\_name, main="Sem outliers", xlab=NA, ylab=NA)  
 title("Verificação de outliers", outer=TRUE)  
 na2 <- sum(is.na(var\_name))  
 cat("Outliers identificados:", na2 - na1, "\n")  
 cat("Proporção (%) de outliers:", round((na2 - na1) / sum(!is.na(var\_name))\*100, 1), "\n")  
 cat("Média dos outliers:", round(mo, 2), "\n")  
 m2 <- mean(var\_name, na.rm = T)  
 cat("Média sem remover outliers:", round(m1, 2), "\n")  
 cat("Média se removermos outliers:", round(m2, 2), "\n")  
 dt[as.character(substitute(var))] <- invisible(var\_name)  
 assign(as.character(as.list(match.call())$dt), dt, envir = .GlobalEnv)  
 return(invisible(dt))  
}  
outlierKD(imdb\_data, budget)



## Outliers identificados: 278   
## Proporção (%) de outliers: 7.9   
## Média dos outliers: 256441913   
## Média sem remover outliers: 45854762   
## Média se removermos outliers: 29237326

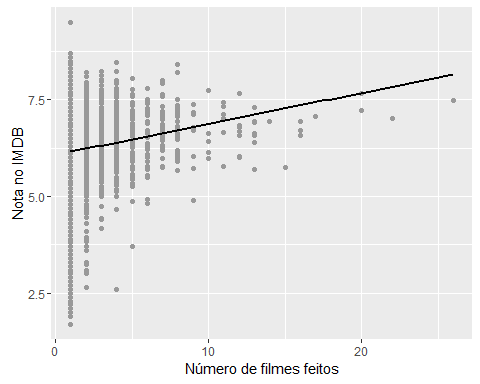
ggplot(imdb\_data,aes(x=budget, y=imdb\_score)) + geom\_point(colour="grey60") +  
 stat\_smooth(method=lm, se=FALSE, colour="black") + ylab("Nota no IMDB") + xlab("Orçamento do Filme")



Este novo gráfico confirma o que foi observado na matriz de correlação acima, que a nota no IMDB de um filme e seu orçamento estão descorrelacionados.

### Relação entre a nota IMDB e o Diretor

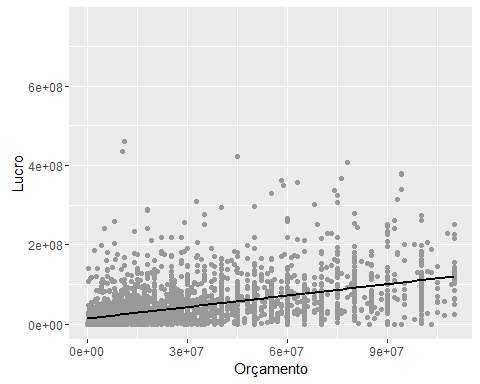
ggplot(directorMovies,aes(x=number, y=mean)) + geom\_point(colour="grey60") +  
 stat\_smooth(method=lm, se=FALSE, colour="black") + ylab("Nota no IMDB") + xlab("Número de filmes feitos")



Este scatterplot mostra que o diretor do filme e sua nota no IMDB apresentam forte correlação positiva.

### Relação entre o lucro e o orçamento

ggplot(imdb\_data,aes(x=budget, y=gross)) + geom\_point(colour="grey60") +  
 stat\_smooth(method=lm, se=FALSE, colour="black") + ylab("Lucro") + xlab("Orçamento")



Desse gráfico não podemos inferir nenhuma conclusão clara sobre a relação entre lucro e orçamento de um filme. Os pontos fora da reta são muitos para serem considerados outliers.

# Conclusões gerais da Análise Exploratória

Ao analisar nosso conjunto de dados um primeiro aspecto que podemos notar é a grande assimetria das variáveis. A maior parte delas possui uma distribuição assimétrica e só pode de alguma forma ser comparada com uma distribuição normal quando aplicamos a função logaritmo. Essa é uma característica bem interessante e que pode ser um indicador que um processo de *feature scaling* se faz necessário para a aplicação dos modelosde regressão.

Outra característica que deve ser levada em conta é a presença de um grande número de outliers em algumas variáveis, como o orçamento. Além de deixar os modelos de certa forma mais sensível, pode se fazer necessário retirar esses outliers para tornar futuras conclusões válidas.

Por fim, um outro aspecto interessante que merece ser comentado é o fato de não existir uma correlação forte entre quase nenhuma das variáveis. A falta de correlação entre orçamento e lucro, como tivemos a oportunidade de verificar tanto na matriz de correlação quanto em um scatterplot, foi a mais intrigante.

### Particionamento dos conjunto de dados

Utilizaremos dois conjuntos de dados ao longo das análises, um para treino e um para testes. Essa estratégia visa evitar *(ou ao menos identificar)* o problema de overfitting do modelo no conjunto de treino.

A proporção da separação foi de 70% dos dados para treino e 30% para testes. Essas proporção foi retirada da literatura.

imdb\_data$director\_name <- temp  
  
set.seed(161095)  
split = sample.split(imdb\_data$imdb\_score,SplitRatio = 0.7)  
trainData = subset(imdb\_data,split == TRUE)  
testData = subset(imdb\_data,split == FALSE)  
  
# Trecho de criação de novas variáveis para o diretor  
  
directorMean <-function(data){  
 data = directorHistory [which(directorHistory ==data),2]  
 data  
}  
  
directorMNumber <-function(data){  
 data = directorHistory [which(directorHistory ==data),3]  
 data  
}  
  
directorHistory = summarise(group\_by(trainData,director\_name),mean=mean(imdb\_score),number = n())  
directorIMDB = sapply(trainData$director\_name,directorMean)  
directorMovies = sapply(trainData$director\_name,directorMNumber)  
trainData$directorIMDB <- as.numeric(directorIMDB)  
trainData$directorMovies <- as.numeric(directorMovies)  
  
  
directorHistory = summarise(group\_by(testData,director\_name),mean=mean(imdb\_score),number = n())  
directorIMDB\_t = sapply(testData$director\_name,directorMean)  
directorMovies\_t = sapply(testData$director\_name,directorMNumber)  
testData$directorIMDB <- as.numeric(directorIMDB\_t)  
testData$directorMovies <- as.numeric(directorMovies\_t)  
  
trainData = subset(trainData, select=-c(director\_name))  
testData = subset(testData, select=-c(director\_name))  
  
  
  
# fim da normalização do director\_name  
  
trainData<- trainData[sapply(trainData,is.numeric)]  
testData <- testData[sapply(testData ,is.numeric)]  
trainData <- na.exclude(trainData)  
testData <- na.exclude(testData)

### Definição das Variáveis

Nesse trecho iremos definir de maneira genérica as variáveis que serão utilizadas daqui em diante. Esse processo poupa o trabalho de reescrever código depois e também facilita a manutenção do mesmo.

goal\_variable = "imdb\_score"  
dependent\_variable = paste(goal\_variable," ~ ")  
independent\_variables = paste(colnames(trainData)[colnames(trainData) != goal\_variable],collapse = "+")  
model\_variables = paste(dependent\_variable,independent\_variables)

## Metodologia

A estratégia adotada será partir do modelo mais simples e rápido para o modelo mais complexo e robusto. Visto isso, em nossa análise testaremos os quatro seguintes modelos:

* Modelo "Burro", ou previsão pela média;
* Modelo de Regressão Linear;
* Modelo da Árvore de Regressão;
* Modelo de Florestas Aleatórias.

Além disso, para facilitar a avaliação dos modelos, adotaremos um indicador numérico unitário de erro. O indicador escolhido será o **RMSE**. O RMSE *(root-mean-square error)* representa o desvio padrão das diferenças entre os valores estimados e os valores esperados e é calculado como a raiz quadrada do SSE *(sum of squared errors of prediction* dividido pelo número de registros. O interessante desse indicador é que ele dá uma estimativa de erro na mesma escala da variável.

### Modelo "Burro"

O modelo "Burro" consiste em estimar os valores de uma variável através da média dos valores observados dessa mesma variável.

DumbPredict = mean(trainData$imdb\_score)  
SSE = sum((testData$imdb\_score - DumbPredict)^2)  
RMSE = sqrt(SSE / nrow(testData))  
RMSE

## [1] 1.056718

Este é um resultado RMSE muito ruim, o que já era esperado, pois, como o nome indica, este modelo não é adequado para fazer estimativas na maior parte dos casos em que não se tem um volume extremamente grande de dados.

## Modelo de Regressão Linear

O modelo de Regressão Linear consiste em estimar uma variável *(chamada de variável dependente)* a partir de outra variável *(chamada de variável independente)* estimando os parâmetros da relação linear entre elas.

Uma relação linear é do tipo: y(t) = ax(t) + b, onde y(t) é a variável dependente e x(t) é a variável independente.

##   
## Call:  
## lm(formula = model\_variables, data = trainData)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.5180 -0.2133 0.0227 0.2265 1.7292   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.258e+01 2.435e+00 5.166 2.59e-07 \*\*\*  
## num\_critic\_for\_reviews 2.191e-04 1.515e-04 1.447 0.14808   
## duration 3.250e-03 5.266e-04 6.171 7.92e-10 \*\*\*  
## director\_facebook\_likes -1.003e-05 3.840e-06 -2.611 0.00908 \*\*   
## actor\_3\_facebook\_likes -2.596e-05 1.837e-05 -1.413 0.15784   
## actor\_1\_facebook\_likes -3.196e-06 1.183e-05 -0.270 0.78705   
## gross 1.132e-09 2.598e-10 4.356 1.38e-05 \*\*\*  
## num\_voted\_users 1.686e-06 1.394e-07 12.089 < 2e-16 \*\*\*  
## cast\_total\_facebook\_likes 3.802e-06 1.182e-05 0.322 0.74774   
## facenumber\_in\_poster -6.399e-03 4.946e-03 -1.294 0.19583   
## num\_user\_for\_reviews -1.108e-04 5.007e-05 -2.213 0.02701 \*   
## language -1.649e-01 5.372e-02 -3.070 0.00216 \*\*   
## country -6.343e-02 2.677e-02 -2.370 0.01788 \*   
## budget -5.337e-09 4.699e-10 -11.359 < 2e-16 \*\*\*  
## title\_year -5.949e-03 1.211e-03 -4.913 9.55e-07 \*\*\*  
## actor\_2\_facebook\_likes -6.811e-07 1.230e-05 -0.055 0.95583   
## aspect\_ratio -1.171e-02 2.597e-02 -0.451 0.65209   
## movie\_facebook\_likes -1.291e-06 8.029e-07 -1.608 0.10801   
## directorIMDB 8.753e-01 1.313e-02 66.661 < 2e-16 \*\*\*  
## directorMovies 1.088e-03 3.828e-03 0.284 0.77628   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4855 on 2453 degrees of freedom  
## Multiple R-squared: 0.7957, Adjusted R-squared: 0.7941   
## F-statistic: 502.9 on 19 and 2453 DF, p-value: < 2.2e-16

## [1] 0.3809591

Como esperado, o modelo de Regressão Linear obteve um resultado melhor que o modelo de previsão pela média, porém ainda não apresenta um bom resultado. Muito disso porque, como vimos em diversas plotagens anteriormente, a maior parte das variáveis não segue uma relação linear.

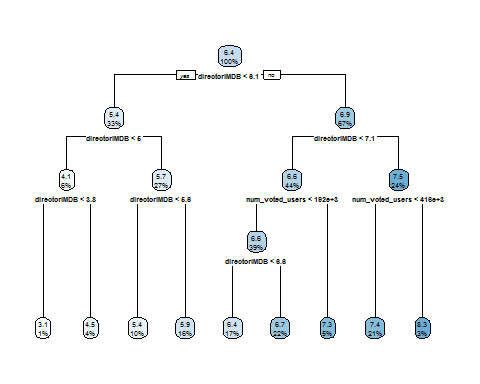
## Modelo da Árvore de Regressão

O modelo da Árvore de Regressão consiste em criar uma árvore de decisão cujo resultado é um número que representa o valor estimado da variável. Os nós de decisão da árvore de regressão são formados por outras variáveis do conjunto de dados que são selecionadas pelo algoritmo.

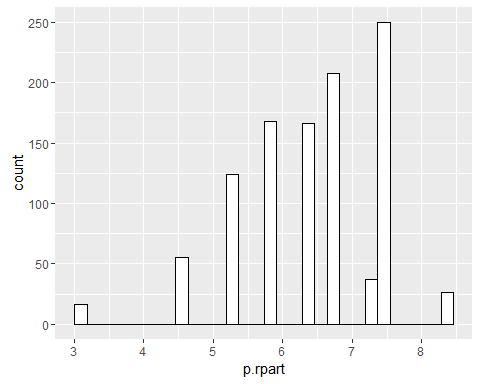
set.seed(3)  
m.rpart <- rpart(imdb\_score~.,data=trainData)  
m.rpart

## n= 2473   
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 2473 2830.305000 6.442539   
## 2) directorIMDB< 6.1225 807 701.198500 5.412763   
## 4) directorIMDB< 4.975 147 122.120000 4.142857   
## 8) directorIMDB< 3.75 36 8.965556 3.061111 \*  
## 9) directorIMDB>=3.75 111 57.365590 4.493694 \*  
## 5) directorIMDB>=4.975 660 289.217300 5.695606   
## 10) directorIMDB< 5.629167 252 72.355670 5.357540 \*  
## 11) directorIMDB>=5.629167 408 170.272100 5.904412 \*  
## 3) directorIMDB>=6.1225 1666 858.800500 6.941357   
## 6) directorIMDB< 7.069048 1082 406.184400 6.642514   
## 12) num\_voted\_users< 192187 960 314.766700 6.563437   
## 24) directorIMDB< 6.591667 420 127.677400 6.355476 \*  
## 25) directorIMDB>=6.591667 540 154.797500 6.725185 \*  
## 13) num\_voted\_users>=192187 122 38.178440 7.264754 \*  
## 7) directorIMDB>=7.069048 584 176.955600 7.495034   
## 14) num\_voted\_users< 416153 518 114.193400 7.387838 \*  
## 15) num\_voted\_users>=416153 66 10.092730 8.336364 \*

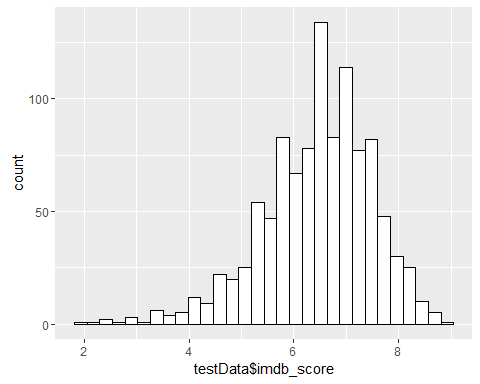
rpart.plot(m.rpart)



p.rpart <- predict(m.rpart,testData)  
tree\_dataframe <- data.frame(p.rpart,testData$imdb\_score)  
ggplot(tree\_dataframe, aes(x=p.rpart)) + geom\_histogram(fill="white", colour="black")



ggplot(tree\_dataframe, aes(x=testData$imdb\_score)) + geom\_histogram(fill="white", colour="black")



cor(p.rpart,testData$imdb\_score)

## [1] 0.9061751

RMSE=sqrt(mean((p.rpart-testData$imdb\_score)^2))  
RMSE

## [1] 0.4479516

Diferentemente do que era esperado, o modelo de árvore de regressão obteve um resultado pior do que o modelo de regressão linear, que é mais simples. Uma possível causa deste resultado é a sensibilidade desse modelo a outliers.

## Modelo de Florestas Aleatórias

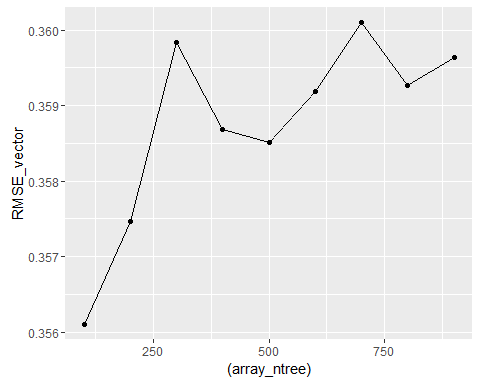
O modelo de florestas aleatórias cria um determinado número de árvores nas quais ele distribui os parâmetros aleatóriamente. Em geral, de todas essas árvores geradas aleatóriamente, aquela que obtem o melhor resultado é melhor do que a árvore criada pelo algoritmo no modelo de árvore de regressão simples. Nesta análise, foram criadas florestas aleatórias com diferentes números de árvores para obter aquele que gera o melhor resultado.

array\_ntree<- c(100,200,300,400,500,600,700,800,900)  
RMSE\_vector <- c()  
for(i in array\_ntree){  
 set.seed(1995)  
 stevenForest = randomForest(imdb\_score ~.,data=trainData,ntree = i)  
 predictForest = predict(stevenForest,newdata= testData)  
 SSE = sum((testData$imdb\_score - predictForest)^2)  
 RMSE = sqrt(SSE / nrow(testData))  
 RMSE\_vector <- c(RMSE\_vector,RMSE)  
}  
min(RMSE\_vector)

## [1] 0.3561032

**Desempenho do modelo de acordo com o número de árvores utilizadas**

ggplot(data.frame(array\_ntree,RMSE\_vector), aes(x=(array\_ntree), y=RMSE\_vector)) + geom\_line() + geom\_point()



Como esperado, o modelo de florestas aleatórias, por ser o modelo mais complexo e robusto, obteve um excelente resultado, sendo melhor do que todos os testados anteriormente.