Imdb_Movie_Analysis

David Kressley

12/1/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: ’lubridate’

## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
imdb <- read_csv("imdb.csv",
                 col_types = cols(Date = col_date(format = "%Y"),
                                  Rate = col_number(),
                                  Votes = col_number(),
                                  Duration = col_number()))
```

```
## New names:
## * ‘‘ -> ...1
```

```
library(DataExplorer)

imdb <- imdb[-c(1,2,6,31)]

imdb$Date <- year(imdb$Date)

imdb_cleaned <- imdb %>%
  filter(Date < '2022')
```

```
imdb_cleaned$Certificate <- as.factor(imdb_cleaned$Certificate)

imdb_cleaned[imdb_cleaned == 'No Rate'] <- NA
imdb_cleaned <- imdb_cleaned %>%
                  drop_na() %>%
                  unique()

imdb_factors <- list('Alcohol', 'Frightening', 'Nudity', 'Profanity', 'Violence')

for(feat in imdb_factors){
  imdb_cleaned[[feat]] <- ordered(imdb_cleaned[[feat]], levels = c("None", "Mild", "Moderate", 'Severe'
}
```

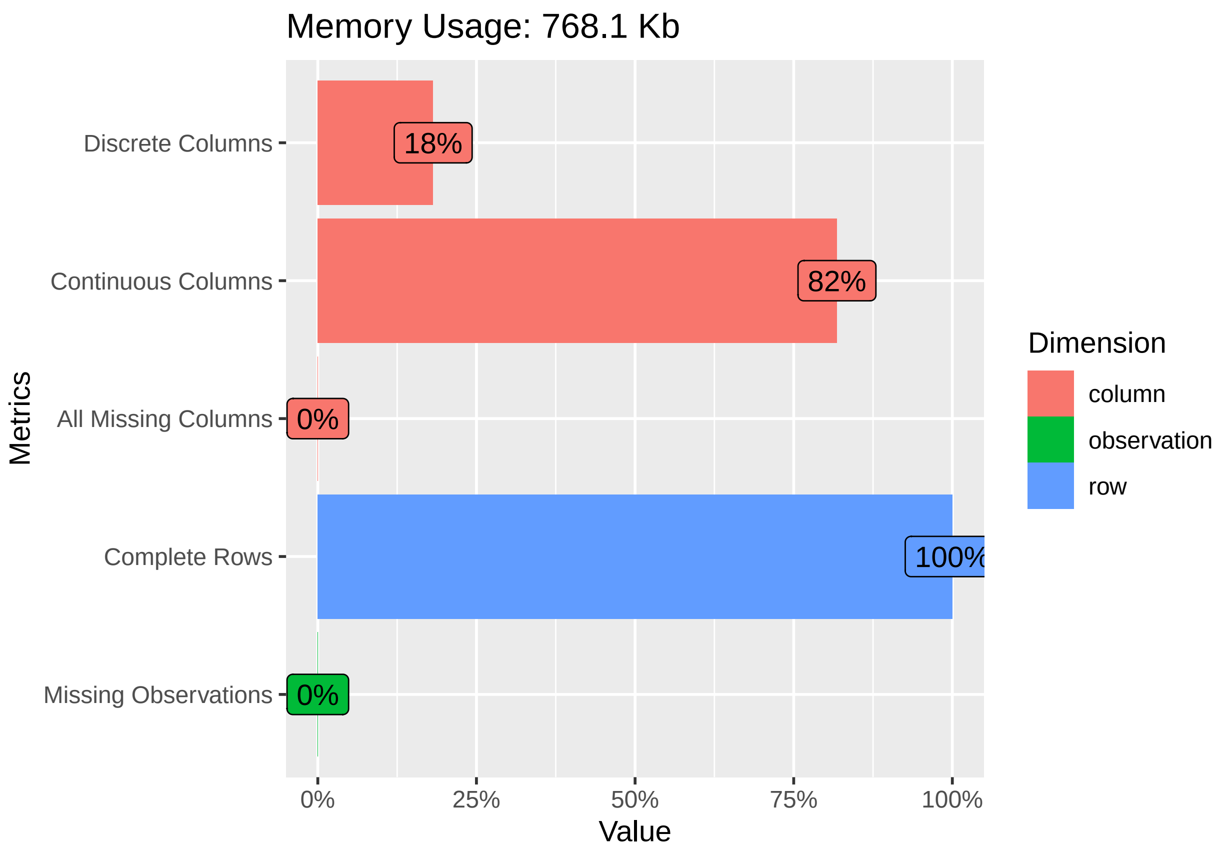
```
# Shape of data
dim(imdb_cleaned)
```

```
## [1] 3225   33
```

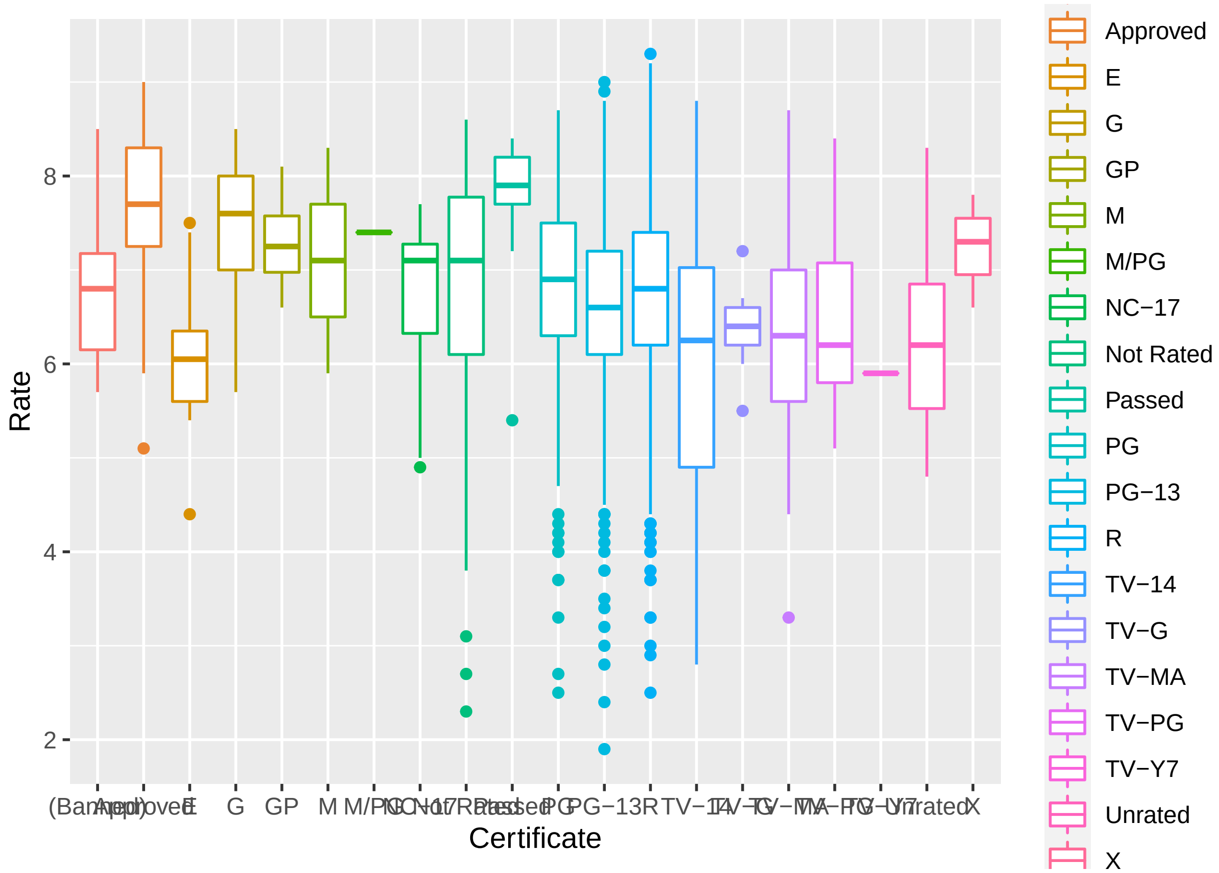
```
# Feature types
print(sapply(imdb_cleaned, class))
```

```
## $Date
## [1] "numeric"
##
## $Rate
## [1] "numeric"
##
## $Votes
## [1] "numeric"
##
## $Action
## [1] "numeric"
##
## $Adventure
## [1] "numeric"
##
## $Animation
## [1] "numeric"
##
## $Biography
## [1] "numeric"
##
## $Comedy
## [1] "numeric"
##
## $Crime
## [1] "numeric"
##
## $Documentary
## [1] "numeric"
##
## $Drama
## [1] "numeric"
##
## $Family
## [1] "numeric"
##
## $Fantasy
## [1] "numeric"
##
## $`Film-Noir`
## [1] "numeric"
##
## $History
## [1] "numeric"
##
## $Horror
## [1] "numeric"
##
## $Music
## [1] "numeric"
##
## $Musical
## [1] "numeric"
##
## $Mystery
## [1] "numeric"
##
## $Romance
## [1] "numeric"
##
## $`Sci-Fi`
## [1] "numeric"
##
## $Short
## [1] "numeric"
##
## $Sport
## [1] "numeric"
##
## $Thriller
## [1] "numeric"
##
## $War
## [1] "numeric"
##
## $Western
## [1] "numeric"
##
## $Duration
## [1] "numeric"
##
## $Certificate
## [1] "factor"
##
## $Nudity
## [1] "ordered" "factor"
##
## $Violence
## [1] "ordered" "factor"
##
## $Profanity
## [1] "ordered" "factor"
##
## $Alcohol
## [1] "ordered" "factor"
##
## $Frightening
## [1] "ordered" "factor"
```

```
plot_intro(imdb_cleaned)
```



```
imdb_cleaned %>%
  ggplot(aes(x = Certificate, y = Rate, col = Certificate)) + geom_boxplot()
```



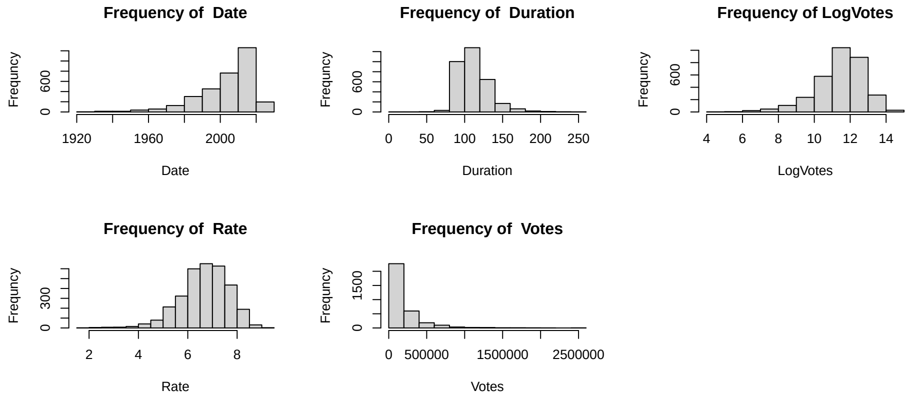
```
summary(imdb_cleaned)
```

```
##       Date           Rate           Votes            Action
##  Min.   :1922   Min.   :1.900   Min.   :    128   Min.   :0.0000
##  1st Qu.:1997   1st Qu.:6.200   1st Qu.:  47733   1st Qu.:0.0000
##  Median :2009   Median :6.800   Median : 110741   Median :0.0000
##  Mean   :2004   Mean   :6.725   Mean   : 184181   Mean   :0.2998
##  3rd Qu.:2017   3rd Qu.:7.400   3rd Qu.: 230200   3rd Qu.:1.0000
##  Max.   :2021   Max.   :9.300   Max.   :2474122   Max.   :1.0000
##
##    Adventure        Animation        Biography           Comedy
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000
##  Mean   :0.2416   Mean   :0.0493   Mean   :0.06109   Mean   :0.3042
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
##
##      Crime         Documentary            Drama            Family
##  Min.   :0.0000   Min.   :0.0000000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.0000000   Median :0.0000   Median :0.00000
##  Mean   :0.1721   Mean   :0.0003101   Mean   :0.4766   Mean   :0.05116
##  3rd Qu.:0.0000   3rd Qu.:0.0000000   3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.0000000   Max.   :1.0000   Max.   :1.00000
```

```
##
##     Fantasy         Film-Noir          History            Horror
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.1029   Mean   :0.00124   Mean   :0.03039   Mean   :0.1829
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##
##      Music            Musical            Mystery          Romance
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :0.000000   Median :0.0000   Median :0.0000
##  Mean   :0.01922   Mean   :0.009612   Mean   :0.1184   Mean   :0.1299
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.0000   Max.   :1.0000
##
##      Sci-Fi           Short               Sport             Thriller
##  Min.   :0.0000   Min.   :0.0000000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000000   Median :0.00000   Median :0.0000
##  Mean   :0.1147   Mean   :0.0006202   Mean   :0.01519   Mean   :0.2019
##  3rd Qu.:0.0000   3rd Qu.:0.0000000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000000   Max.   :1.00000   Max.   :1.0000
##
##       War             Western            Duration        Certificate
##  Min.   :0.00000   Min.   :0.00000   Min.   : 11.0   R        :1485
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.: 97.0   PG-13    : 918
##  Median :0.00000   Median :0.00000   Median :109.0   PG       : 409
##  Mean   :0.01395   Mean   :0.01178   Mean   :112.1   Not Rated: 130
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:123.0   TV-MA    :  76
##  Max.   :1.00000   Max.   :1.00000   Max.   :242.0   G        :  57
##                                                      (Other)  : 150
##       Nudity         Violence       Profanity        Alcohol
##  None    : 947   None    : 318   None    : 383   None    : 491
##  Mild    :1328   Mild    : 985   Mild    :1164   Mild    :2052
##  Moderate: 683   Moderate:1122   Moderate:1045   Moderate: 549
##  Severe  : 267   Severe  : 800   Severe  : 633   Severe  : 133
##
##
##
##    Frightening
##  None    : 430
##  Mild    : 925
##  Moderate:1252
##  Severe  : 618
##
##
##
```

```r
par(mfrow=c(2,2))

imdb_continous <- list('Date', 'Duration', 'Rate', 'Votes')
```

```r
for(feat in imdb_continous){
  hist(imdb_cleaned[[feat]], main = paste('Frequency of ',feat), xlab = feat, ylab = 'Frequncy')
}

 hist(log(imdb_cleaned$Votes), main = 'Frequency of LogVotes', xlab = 'LogVotes', ylab = 'Frequncy')
```



```r
imdb_scatter_feat <- colnames(imdb_cleaned[-c(2,3)])

for(i in imdb_scatter_feat){
  imdb_cleaned %>% ggplot(aes(x = i, y = Rate)) + geom_point()
  print(plot)
}
```

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fc4d2485168>
## <environment: namespace:base>
```

```
library(ggridges)
library(gridExtra)
```

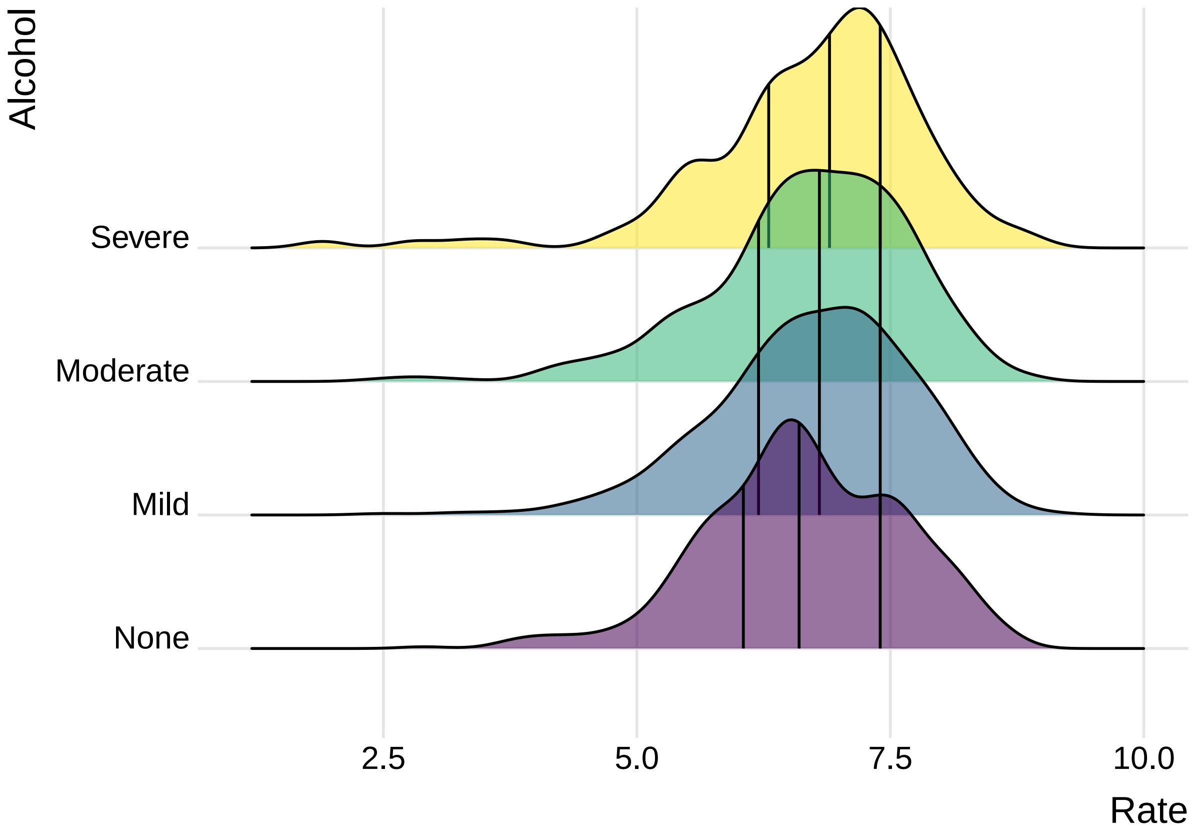
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
##     combine
```

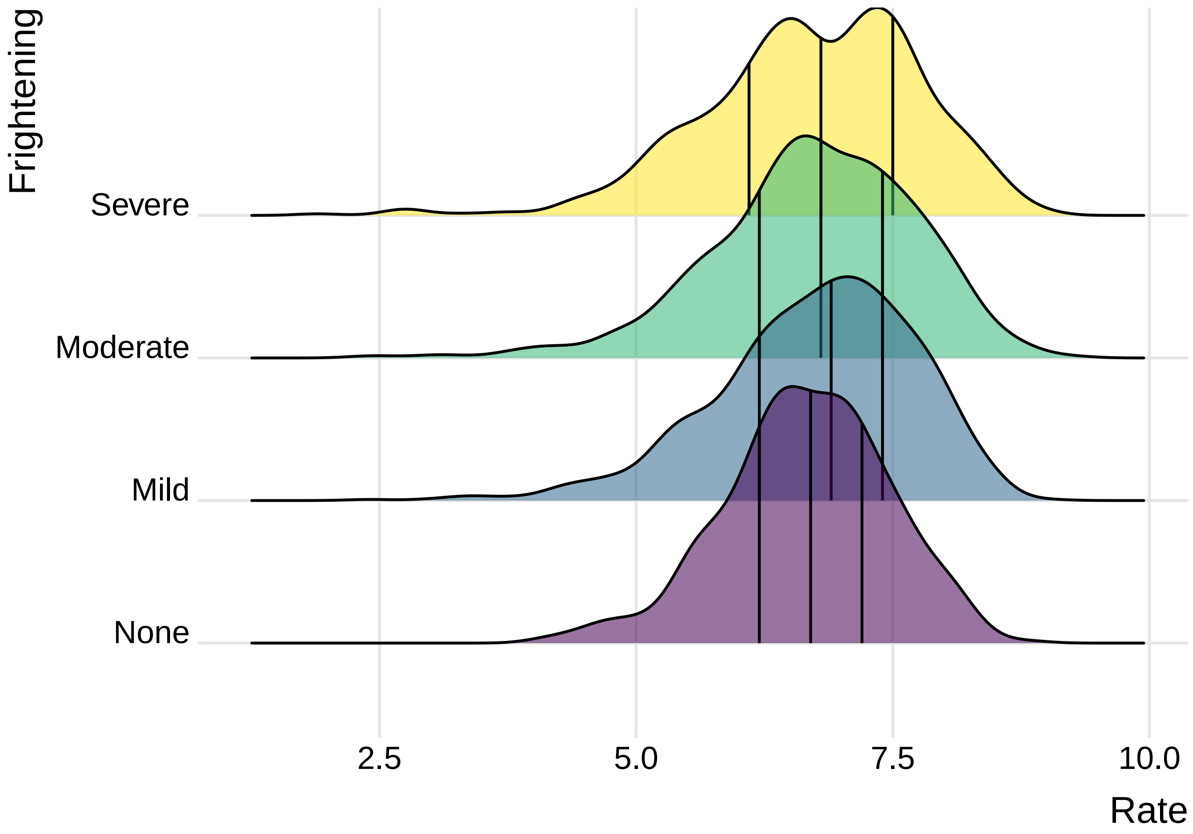
```
p1 <- ggplot(imdb_cleaned, aes(x = Rate, y = Alcohol, fill = Alcohol, alpha = .7)) +
  geom_density_ridges(quantile_lines = TRUE) +
  theme_ridges() +
  theme(legend.position = 'none')

p2 <- ggplot(imdb_cleaned, aes(x = Rate, y = Frightening, fill = Frightening, alpha = .7)) +
  geom_density_ridges(quantile_lines = TRUE) +
  theme_ridges() +
  theme(legend.position = 'none')

p3 <- ggplot(imdb_cleaned, aes(x = Rate, y = Nudity, fill = Nudity, alpha = .7)) +
  geom_density_ridges(quantile_lines = TRUE) +
  theme_ridges() +
  theme(legend.position = 'none')

p4 <- ggplot(imdb_cleaned, aes(x = Rate, y = Profanity, fill = Profanity, alpha = .7)) +
  geom_density_ridges(quantile_lines = TRUE) +
  theme_ridges() +
  theme(legend.position = 'none')

p5 <- ggplot(imdb_cleaned, aes(x = Rate, y = Violence, fill = Violence, alpha = .7)) +
  geom_density_ridges(quantile_lines = TRUE) +
  theme_ridges() +
  theme(legend.position = 'none')

p1
```

```
## Picking joint bandwidth of 0.233
```



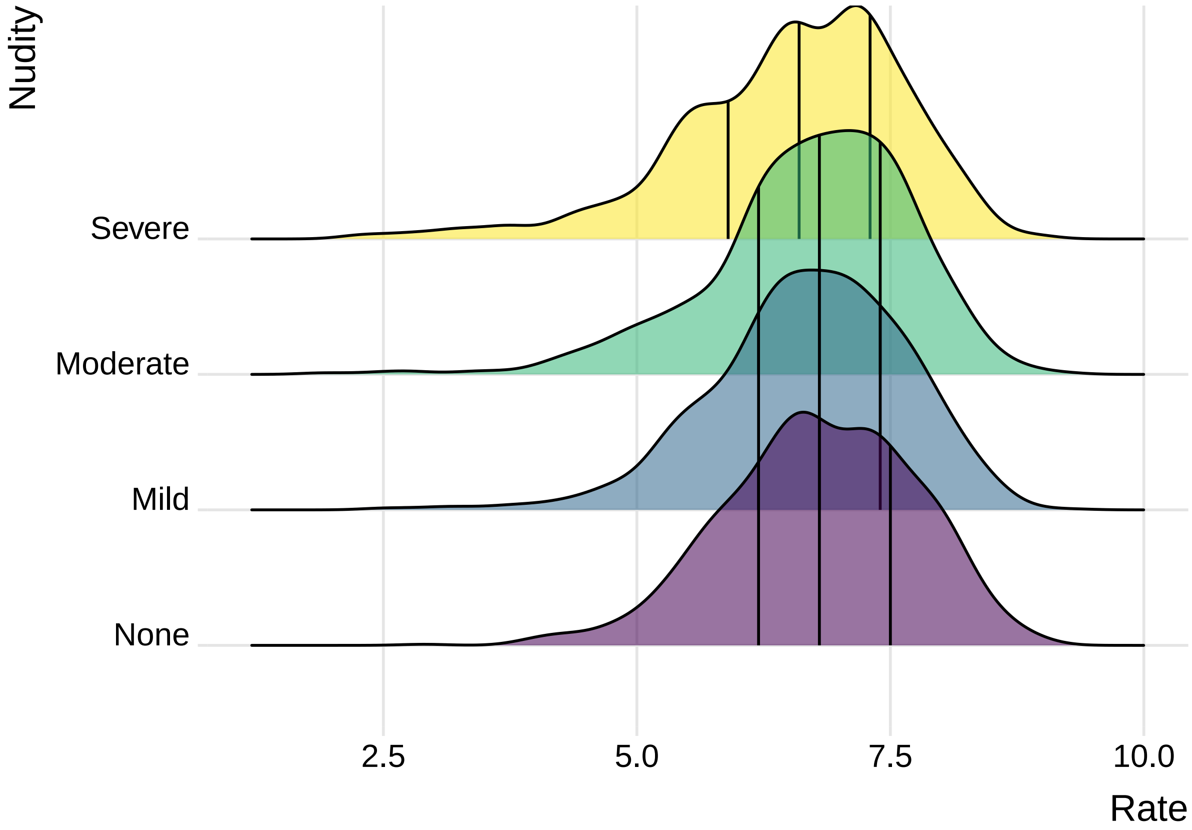
```
p2
```

```
## Picking joint bandwidth of 0.215
```



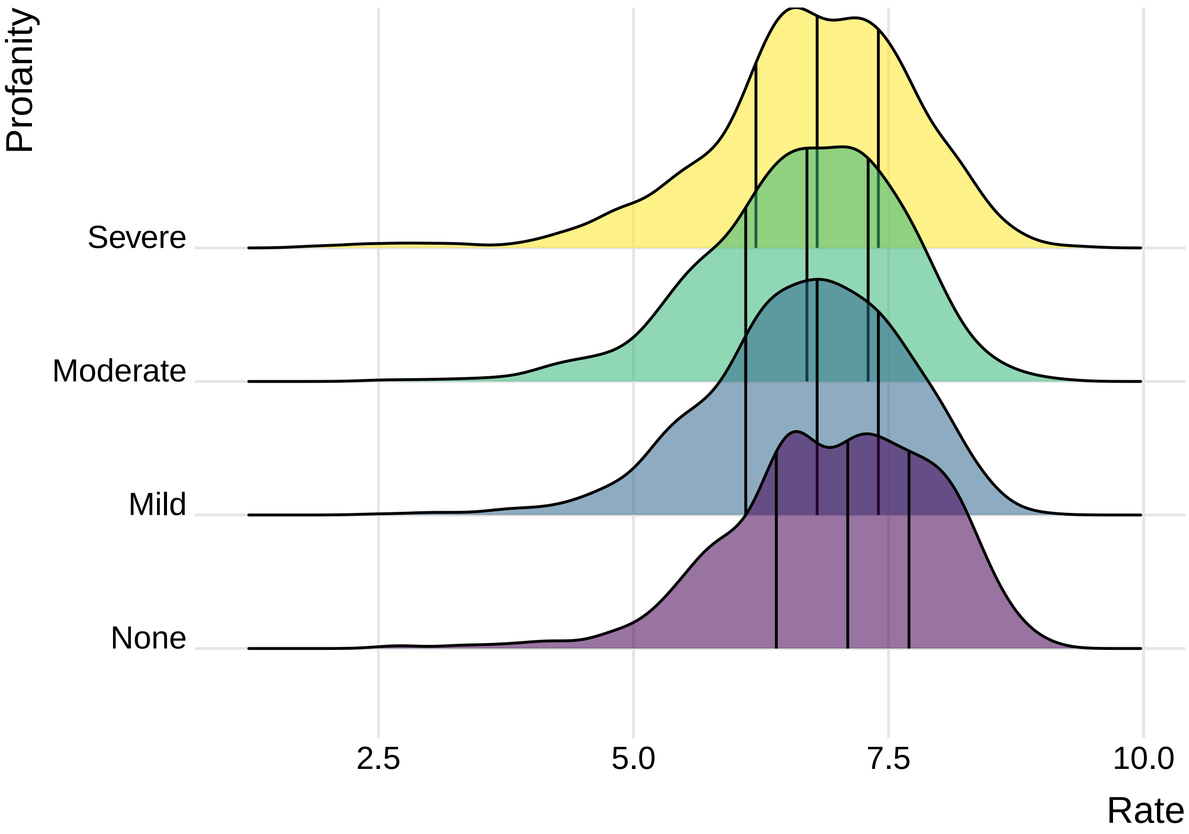
```
p3
```

```
## Picking joint bandwidth of 0.233
```



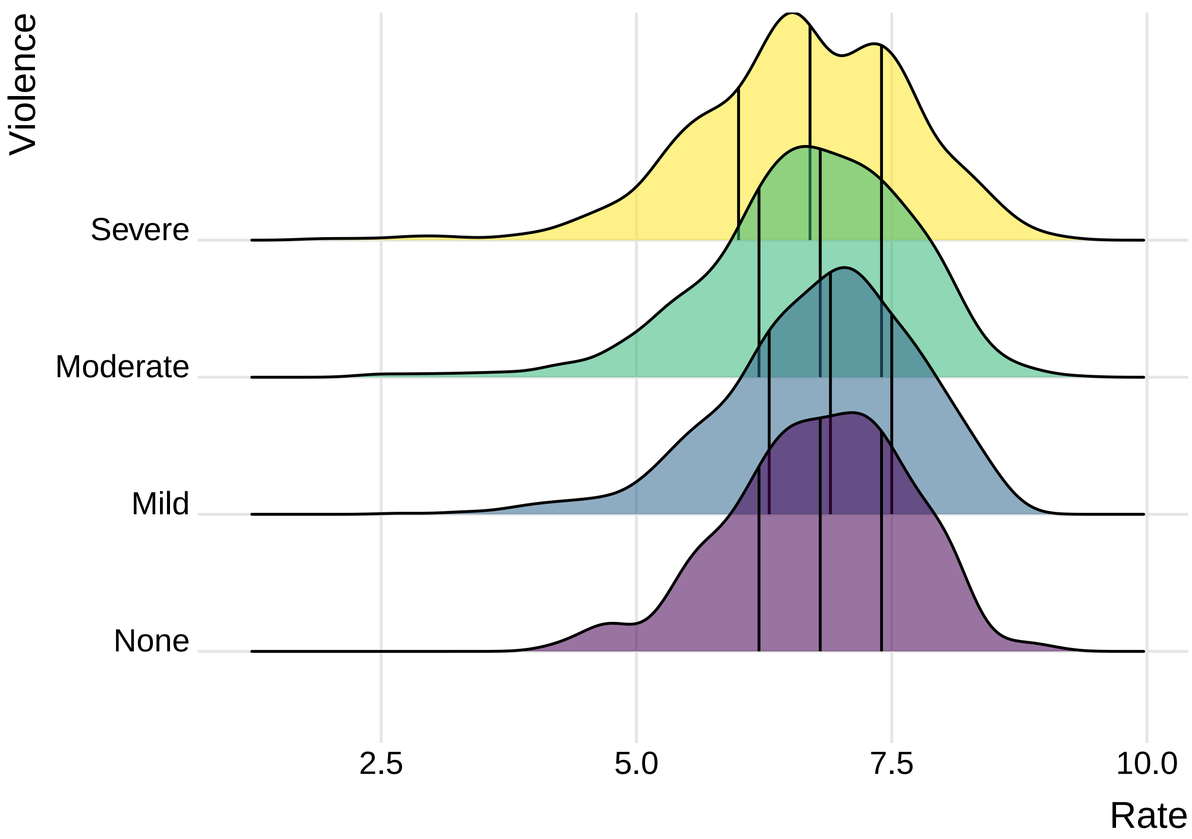
```
p4
```

```
## Picking joint bandwidth of 0.224
```



```
p5
```

```
## Picking joint bandwidth of 0.223
```

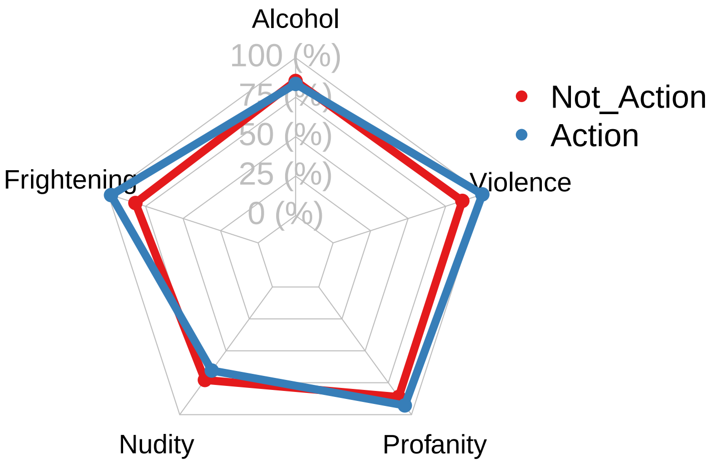


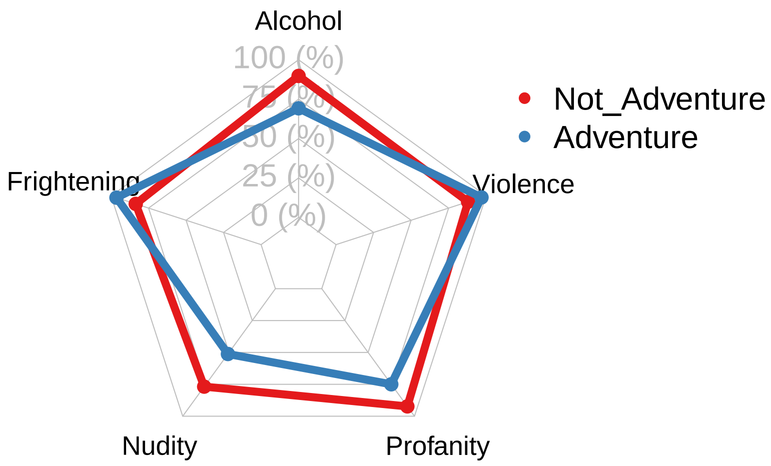
```
library(fmsb)
library(RColorBrewer)
library(scales)
```

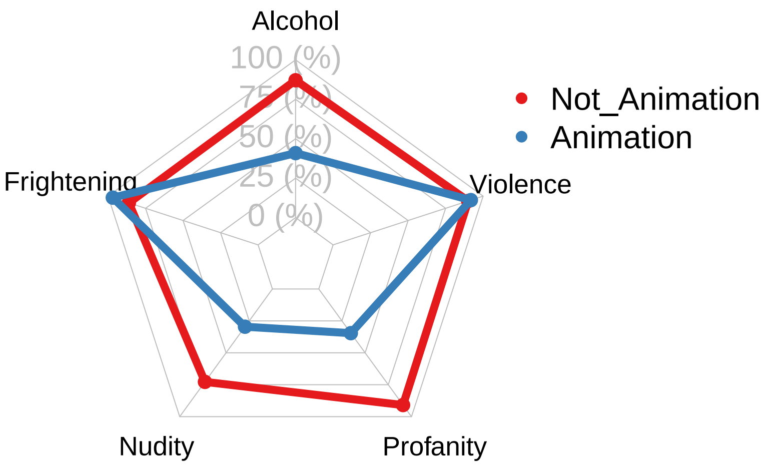
```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

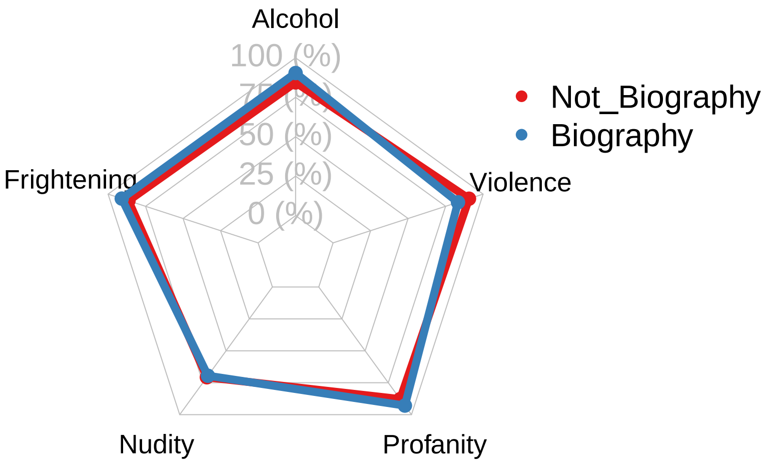
```
# Color vectors
coul <- brewer.pal(3,'Set1')
colors_border <- coul

for(i in 4:26){

  current_feature <- colnames(imdb_cleaned[i])

  imdb_select <- imdb_cleaned %>% select(current_feature, 'Alcohol', 'Frightening', 'Nudity', 'Profanit
  imdb_cleaned

  feature_count <- imdb_select %>% group_by(imdb_select[1]) %>% count(imdb_select[1])

  movie_without <- imdb_select %>% filter(imdb_select[1] == 0) %>%
    summarize(Alcohol = sum(Alcohol != 'None'),
              Frightening = sum(Frightening != 'None'),
              Nudity = sum(Nudity != 'None'),
              Profanity = sum(Profanity != 'None'),
              Violence = sum(Violence != 'None'))

  movie_with <- imdb_select %>% filter(imdb_select[1] == 1) %>%
    summarize(Alcohol = sum(Alcohol != 'None'),
              Frightening = sum(Frightening != 'None'),
              Nudity = sum(Nudity != 'None'),
              Profanity = sum(Profanity != 'None'),
              Violence = sum(Violence != 'None'))

  feature_count <- rbind(movie_without, movie_with) %>% cbind(feature_count)

  for(i in imdb_factors){
    feature_count[i] = feature_count[i] / feature_count['n']
  }

  feature_count <- feature_count %>%
    select(unlist(imdb_factors))

  feature_count <- rbind(rep(1,5) , rep(0,5) , feature_count)
  row.names(feature_count) <- c('Max','Min',paste0('Not_', colnames(imdb_select[1])),paste0(colnames(im

  # plot with default options:
  radarchart(feature_count  , axistype=1 ,
    #custom polygon
    pcol=colors_border, plwd=4 , plty=1,
    #custom the grid
    cglcol="grey", cglty=1, axislabcol="grey", cglwd=0.5,
    #custom labels
    vlcex=0.8
    )
  legend(x=1, y=1, legend = rownames(feature_count[-c(1,2),]), bty = "n", pch=20 , col=colors_border ,
}
```

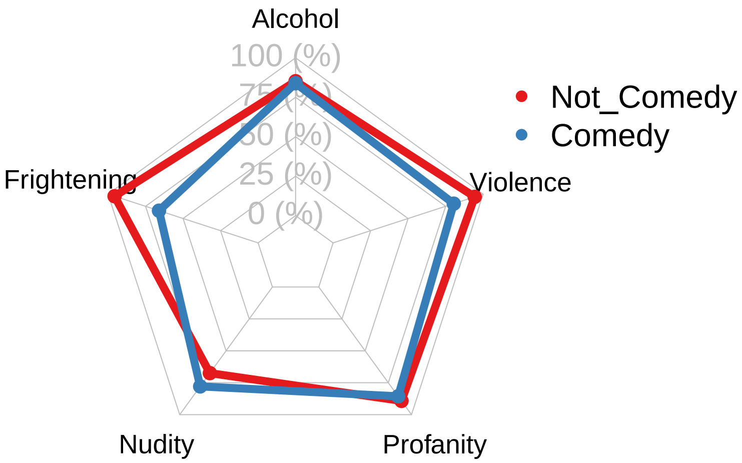
```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(current_feature)` instead of `current_feature` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

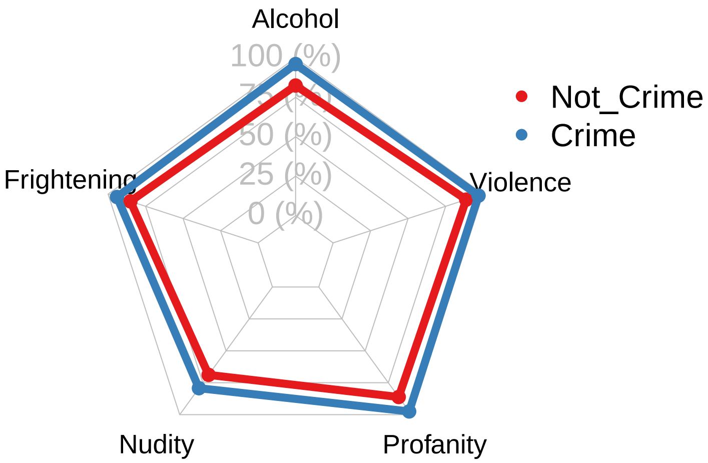

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Action
● Action

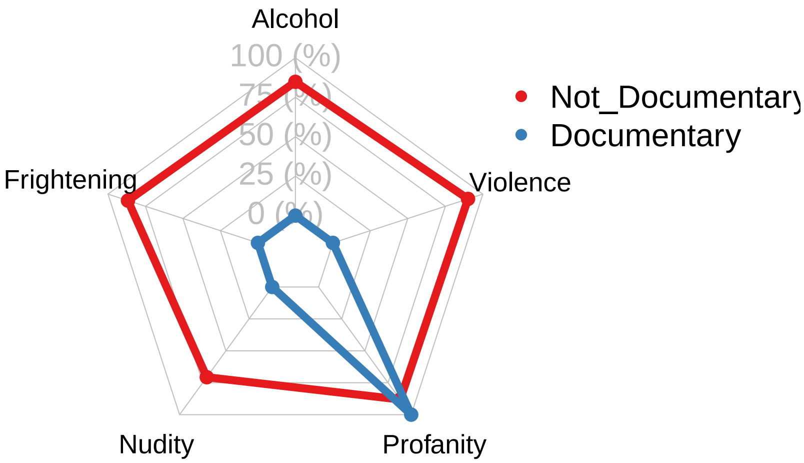

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Adventure
● Adventure

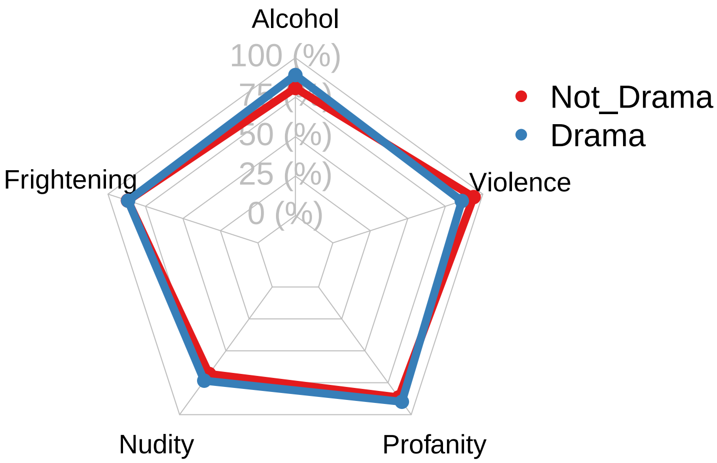

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Animation
● Animation

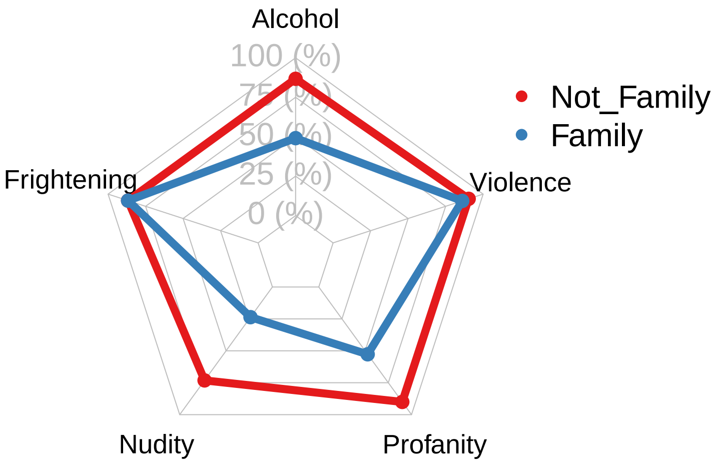

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Biography
● Biography

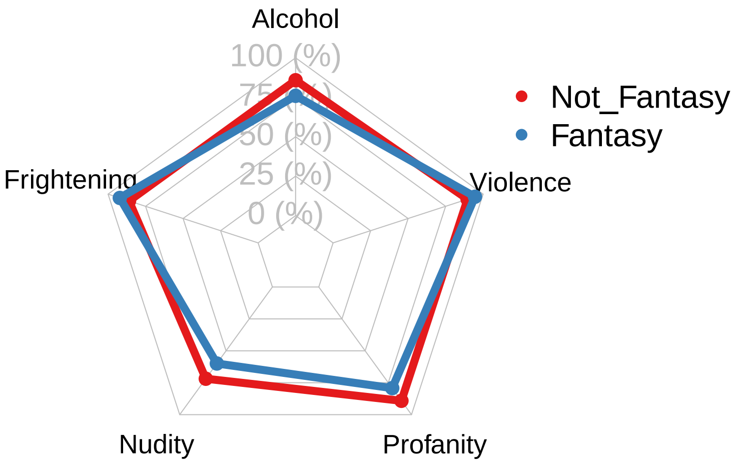

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Comedy
● Comedy

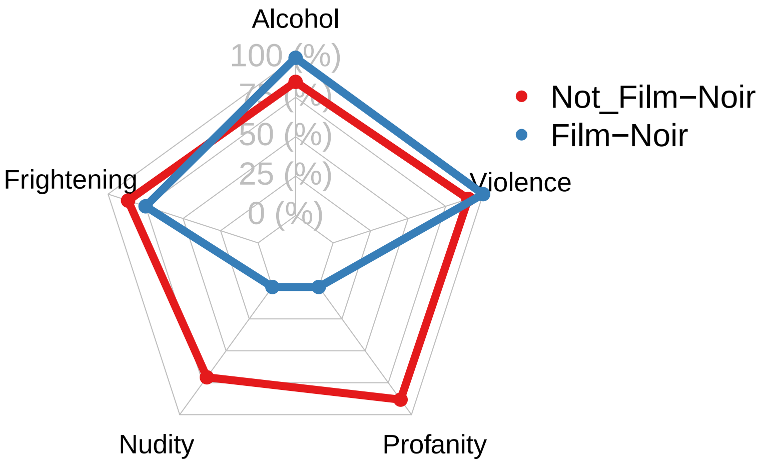

Alcohol
Violence
Profanity
Nudity
Frightening
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
● Not_Crime
● Crime

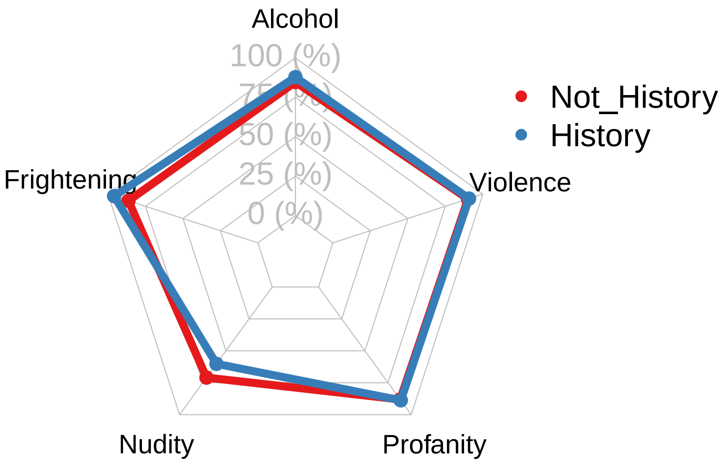

Alcohol
Violence
Profanity
Nudity
Frightening
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
● Not_Documentary
● Documentary

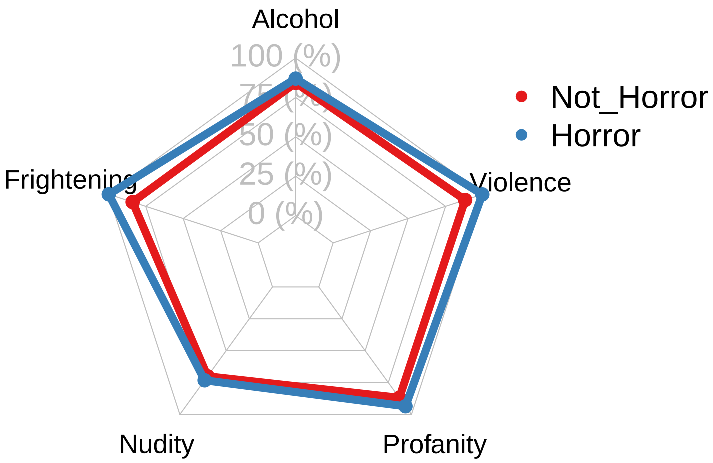

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Drama
● Drama

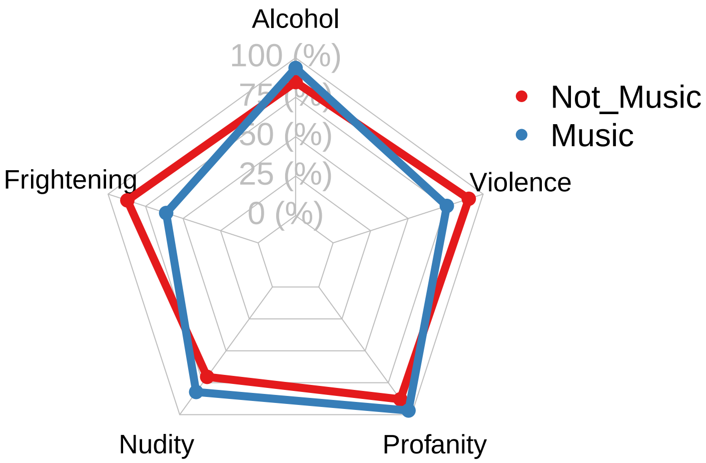

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Family
● Family

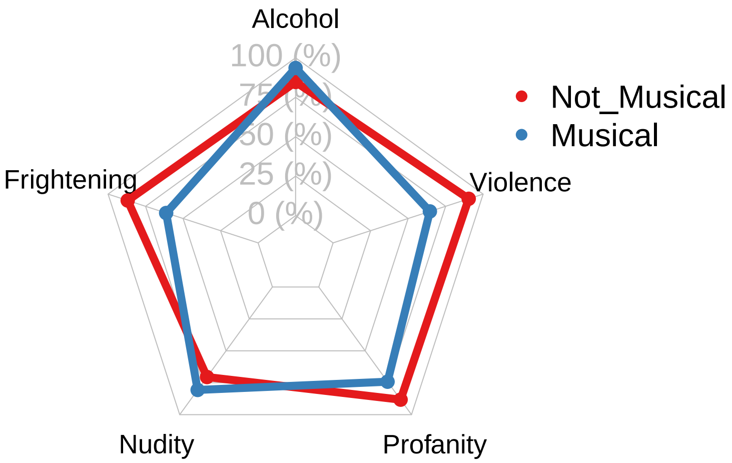

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Fantasy
● Fantasy

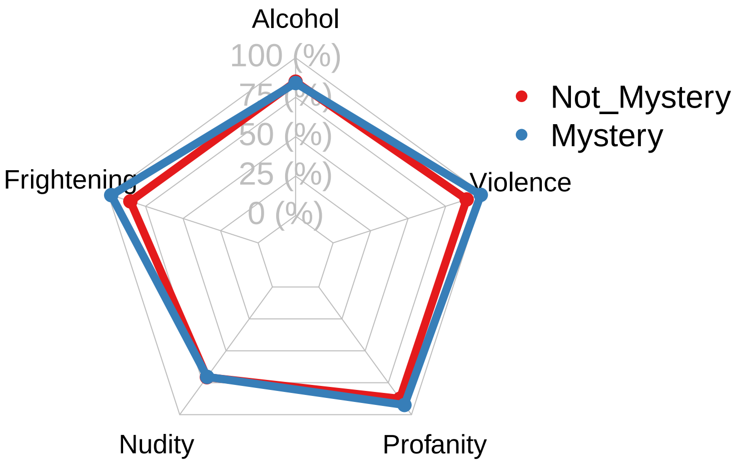

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Film−Noir
● Film−Noir

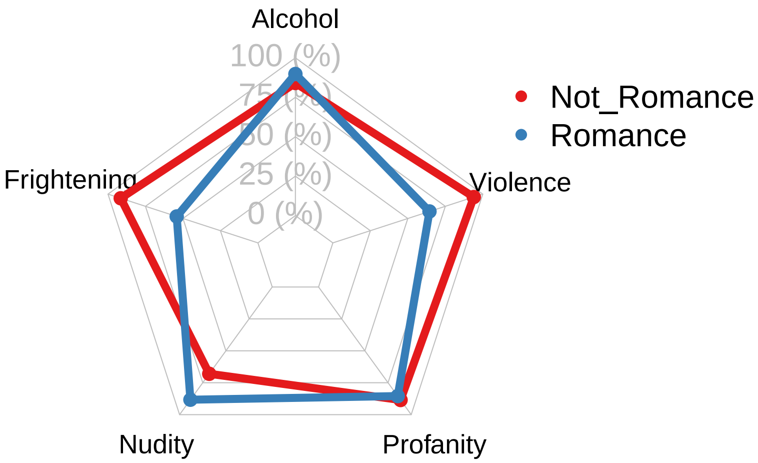

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_History
● History

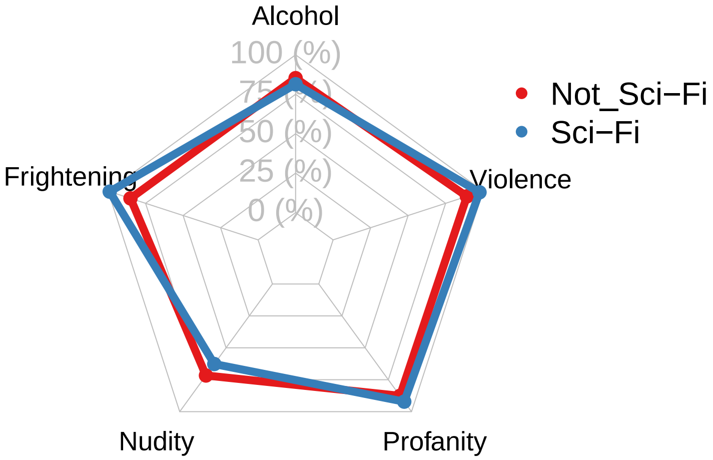

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Horror
● Horror

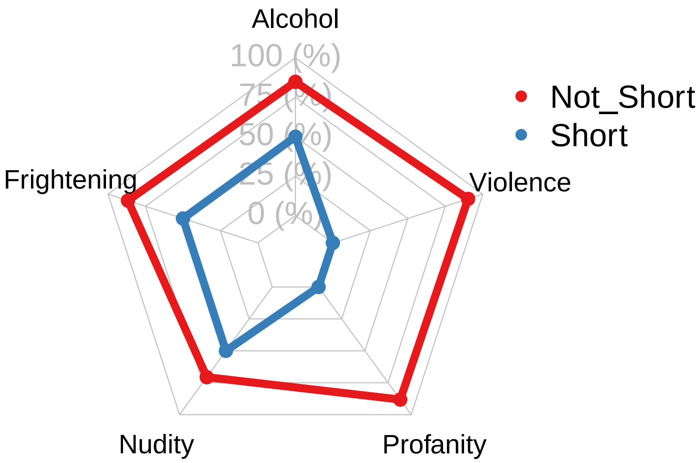

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Music
● Music

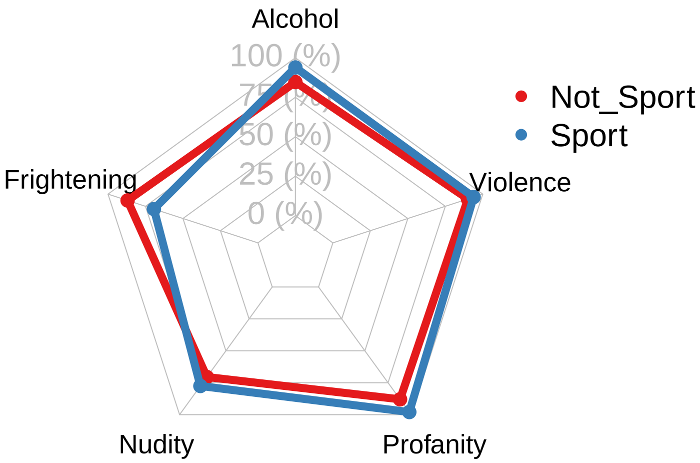

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Musical
● Musical

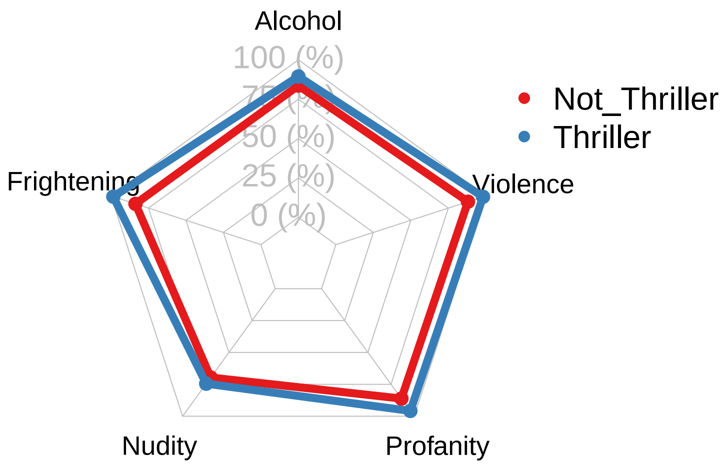

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Mystery
● Mystery

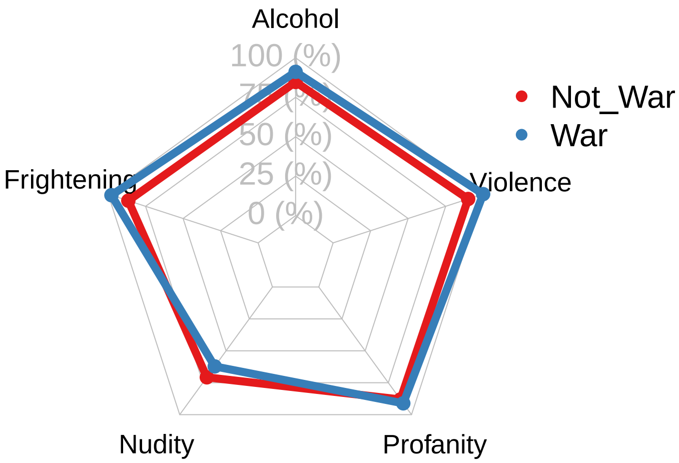

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Romance
● Romance

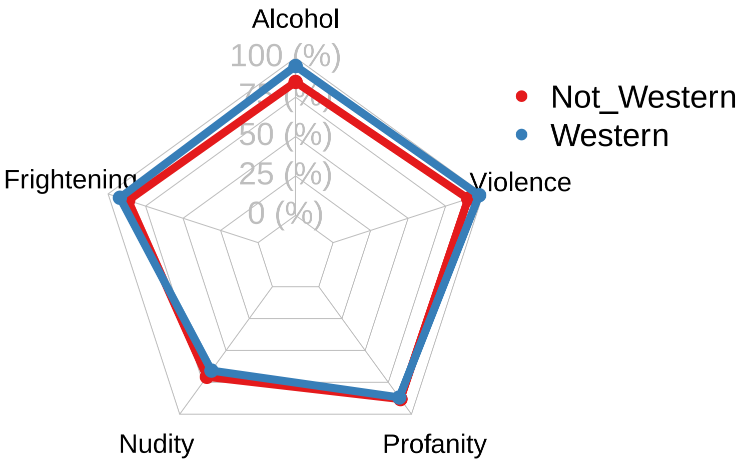

Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Sci−Fi
● Sci−Fi


Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Short
● Short


Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Sport
● Sport


Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_Thriller
● Thriller


Alcohol
100 (%)
75 (%)
50 (%)
25 (%)
0 (%)
Violence
Profanity
Nudity
Frightening
● Not_War
● War



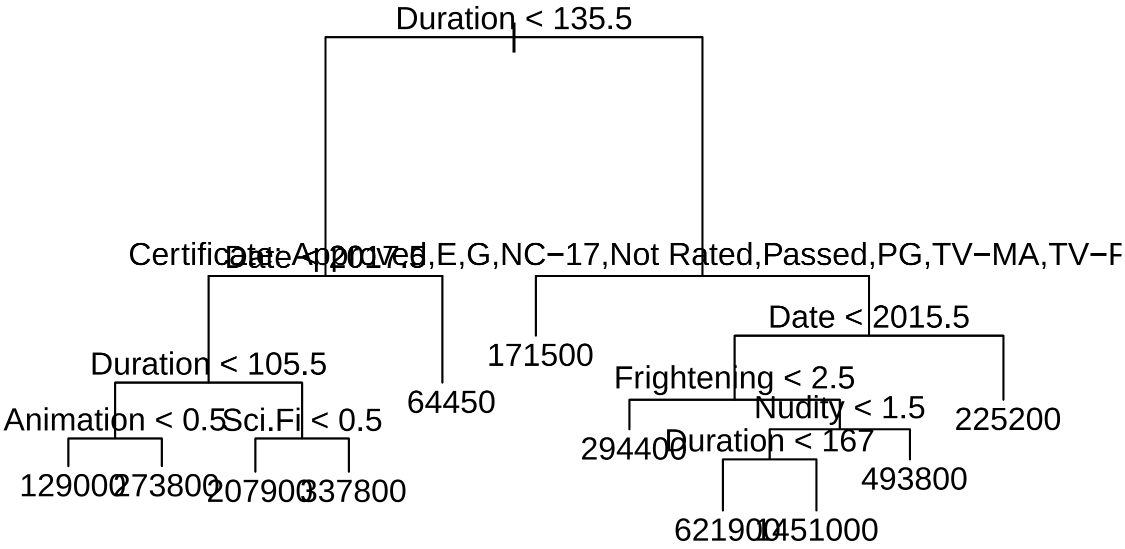
```
# Load library to begin tree modeling
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
# Creating a copy df to begin modeling
imdb_copy <- data.frame(imdb_cleaned)

# Code to save -unused
for(i in imdb_factors){
 imdb_copy[[i]] <- as.numeric(imdb_copy[[i]] )
}

## Regression Tree

# Creating saturated model to test Votes vs all features - Rate
model_tree_v <- tree(Votes~ . -Rate, control=tree.control(nobs = 3225, mindev = 0.01), data = imdb_copy
summary(model_tree_v)
```

```
##
## Regression tree:
## tree(formula = Votes ~ . - Rate, data = imdb_copy, control = tree.control(nobs = 3225,
##     mindev = 0.01))
## Variables actually used in tree construction:
## [1] "Duration"    "Date"         "Animation"   "Sci.Fi"       "Certificate"
## [6] "Frightening" "Nudity"
## Number of terminal nodes:  11
## Residual mean deviance:  3.909e+10 = 1.256e+14 / 3214
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -668500  -99360  -42300       0   43430 1980000
```

```r
plot(model_tree_v)
text(model_tree_v,pretty=0)
```

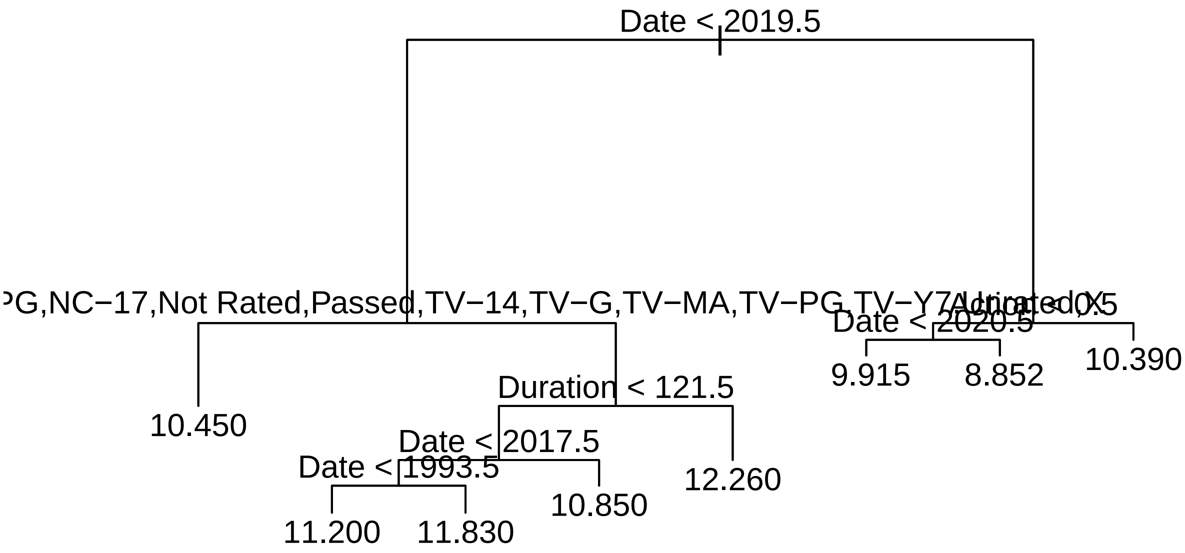


```r
# Creating saturated model to test log(Votes) vs all features - Rate
model_tree_lv <- tree(log(Votes)~ . -Rate, control=tree.control(nobs = 3225, mindev = 0.01), data = imdb
summary(model_tree_lv)
```

```
##
## Regression tree:
## tree(formula = log(Votes) ~ . - Rate, data = imdb_copy, control = tree.control(nobs = 3225,
##     mindev = 0.01))
## Variables actually used in tree construction:
## [1] "Date"        "Certificate" "Duration"    "Action"
## Number of terminal nodes:  8
## Residual mean deviance:  1.101 = 3542 / 3217
## Distribution of residuals:
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.64300 -0.61790  0.05076  0.00000  0.67000  3.21500
```

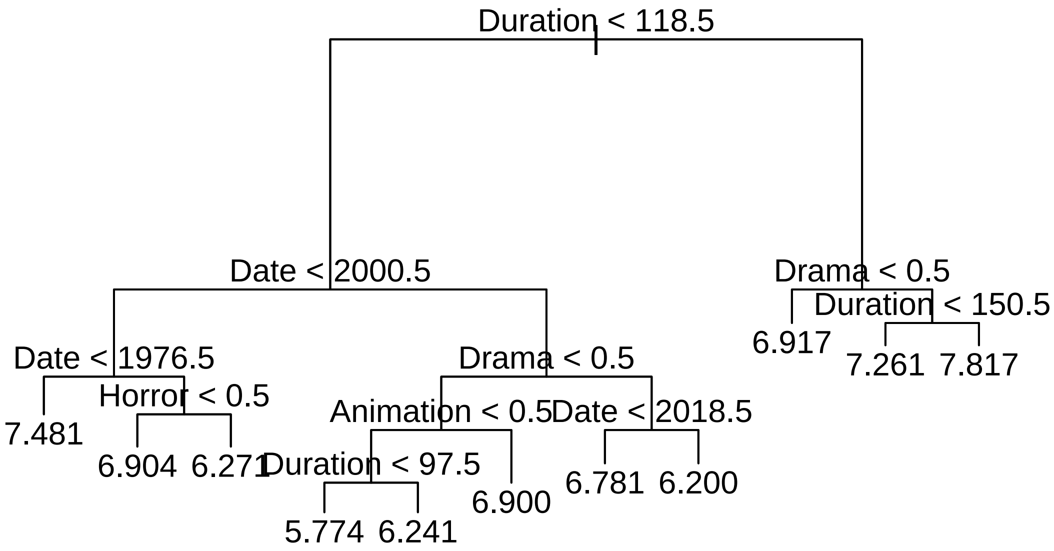
```
plot(model_tree_lv)
text(model_tree_lv,pretty=0)
```



```
# Creating saturated model to test Rate vs all features - Votes
model_tree_r <- tree(Rate~ . -Votes, control=tree.control(nobs = 3225, mindev = 0.01), data = imdb_copy
summary(model_tree_r)
```

```
##
## Regression tree:
## tree(formula = Rate ~ . - Votes, data = imdb_copy, control = tree.control(nobs = 3225,
##     mindev = 0.01))
## Variables actually used in tree construction:
## [1] "Duration"  "Date"      "Horror"    "Drama"     "Animation"
## Number of terminal nodes:  11
## Residual mean deviance:  0.6747 = 2169 / 3214
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.61700 -0.47370  0.05882  0.00000  0.52630  2.60000
```

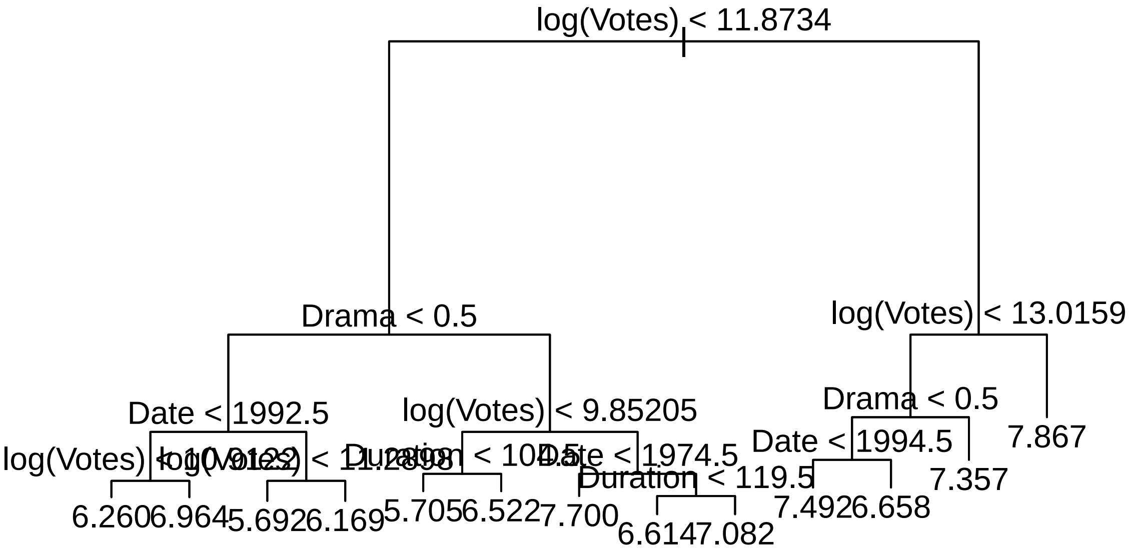
```
plot(model_tree_r)
text(model_tree_r,pretty=0)
```



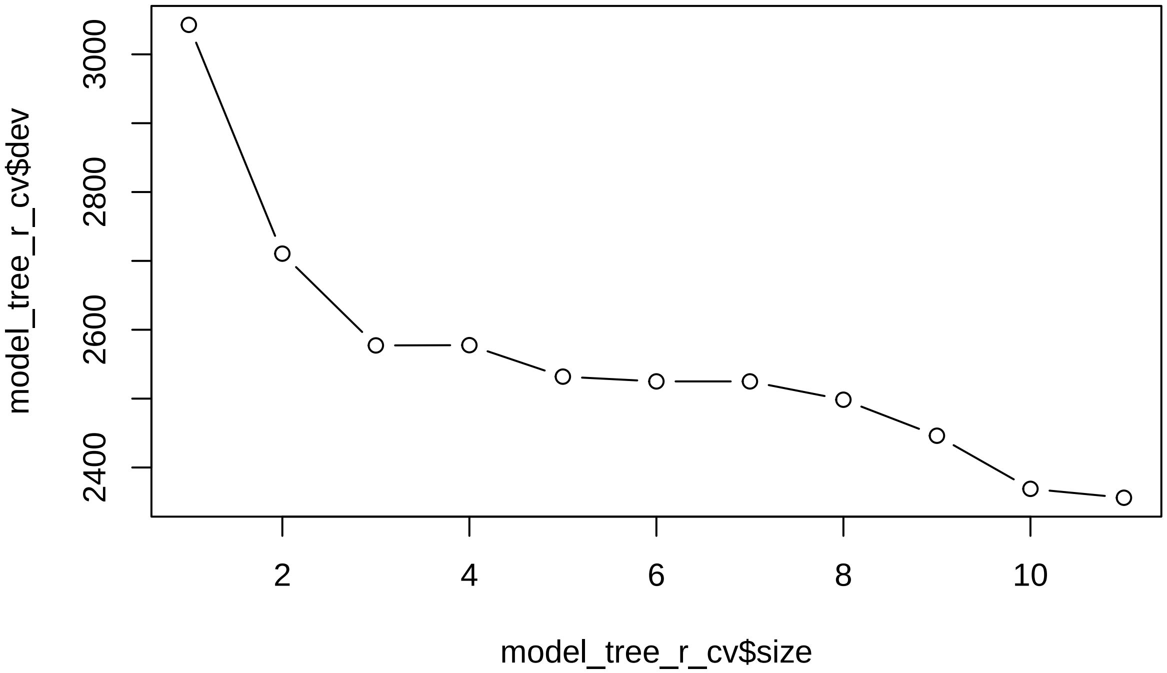
```
# Creating saturated model to test Rate vs all features, log(Votes)
model_tree_lr <- tree(Rate~. + log(Votes) -Votes, control=tree.control(nobs = 3225, mindev = 0.01), data
summary(model_tree_lr)
```

```
##
## Regression tree:
## tree(formula = Rate ~ . + log(Votes) - Votes, data = imdb_copy,
##     control = tree.control(nobs = 3225, mindev = 0.01))
## Variables actually used in tree construction:
## [1] "log(Votes)" "Drama"      "Date"       "Duration"
## Number of terminal nodes:  13
## Residual mean deviance:  0.5153 = 1655 / 3212
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.2690 -0.3825  0.0417  0.0000  0.4429  2.4080
```

```
plot(model_tree_lr)
text(model_tree_lr,pretty=0)
```



```
# Cross Validating a tree
model_tree_r_cv <- cv.tree(model_tree_r)

plot(model_tree_r_cv$size, model_tree_r_cv$dev, type = 'b')
```



```
cbind('Size' = model_tree_r_cv$size, 'Deviance' = model_tree_r_cv$dev)
```

```
##        Size Deviance
##  [1,]    11 2356.047
##  [2,]    10 2368.967
##  [3,]     9 2446.120
##  [4,]     8 2498.409
##  [5,]     7 2525.007
##  [6,]     6 2525.007
##  [7,]     5 2531.915
##  [8,]     4 2577.641
##  [9,]     3 2577.256
## [10,]     2 2710.542
## [11,]     1 3042.800
```

```
# # Training Regression Tree Model
# set.seed(123)
#
# train <- sample(1:nrow(imdb_cleaned), 0.7 * nrow(imdb_cleaned))
#
# model_tree_train <- tree(Rate ~., imdb_cleaned, subset = train)
#
# yhat <- predict(model_tree_train, newdata = imdb_cleaned[-train,])
#
# model_tree_test <- imdb_cleaned[-train, 'Rate']
```

```
#
# plot(yhat, model_tree_test)
# abline(0,1)
```