ASM Homework 1

Linear Model for IDMB data

Maria Gkotsopoulou & Ricard Monge Calvo & Amalia Vradi

13/10/2019

```r
############################################################
###############       DATA LOAD            ###############
############################################################

imdb <- read.csv("IMDB.csv", stringsAsFactors = F, sep=";")


data.frame(variable = names(imdb),
           class = sapply(imdb, class),
           first_values = sapply(imdb, function(x) paste0(head(x),collapse = ", ")),
           row.names = NULL)
```

```
##                 variable     class
## 1             movietitle character
## 2                  gross   integer
## 3                 budget   integer
## 4               duration   integer
## 5              titleyear   integer
## 6             directorfl   integer
## 7               actor1fl   integer
## 8               actor2fl   integer
## 9               actor3fl   integer
## 10                castfl   integer
## 11 facenumber_in_poster   integer
## 12                 genre character
##                                                                                first_values
## 1  10 Days in a Madhouse, 12 Years a Slave, 13 Going on 30, 21 & Over, 21 Grams, 25th Hour
## 2                                 14616, 56667870, 56044241, 25675765, 16248701, 13060843
## 3                              12000000, 20000000, 37000000, 13000000, 20000000, 15000000
## 4                                                              111, 134, 98, 93, 124, 108
## 5                                                      2015, 2013, 2004, 2013, 2003, 2002
## 6                                                                      0, 0, 56, 24, 0, 0
## 7                                                      1000, 2000, 3000, 552, 6000, 22000
## 8                                                          445, 660, 2000, 528, 979, 3000
## 9                                                            247, 500, 533, 499, 430, 346
## 10                                                     2059, 4251, 6742, 2730, 7567, 26050
## 11                                                                       1, 0, 1, 0, 0, 0
## 12                                         Drama, Drama, Comedy, Comedy, Drama, Drama
```

We first check for any missing values and see that there are no NAs.

```r
summary(imdb)
```

```
##   movietitle             gross                budget
##  Length:940         Min.   :     3330   Min.   :   400000
##  Class :character   1st Qu.: 11816543   1st Qu.: 10000000
```

```
##  Mode  :character   Median : 33428175   Median : 24000000
##                     Mean   : 57813237   Mean   : 40484550
##                     3rd Qu.: 70756664   3rd Qu.: 48000000
##                     Max.   :760505847   Max.   :300000000
##     duration       titleyear      directorfl         actor1fl
##  Min.   : 74.0   Min.   :2000   Min.   :    0.0   Min.   :     0.0
##  1st Qu.: 95.0   1st Qu.:2004   1st Qu.:   11.0   1st Qu.:   831.5
##  Median :104.0   Median :2008   Median :   56.0   Median :  2000.0
##  Mean   :108.9   Mean   :2008   Mean   :  757.2   Mean   :  9006.8
##  3rd Qu.:119.0   3rd Qu.:2012   3rd Qu.:  189.8   3rd Qu.: 13000.0
##  Max.   :280.0   Max.   :2016   Max.   :22000.0   Max.   :640000.0
##     actor2fl           actor3fl           castfl
##  Min.   :     0.0   Min.   :    0.0   Min.   :     0
##  1st Qu.:   462.5   1st Qu.:  255.0   1st Qu.:  2422
##  Median :   756.0   Median :  501.0   Median :  4868
##  Mean   :  2391.7   Mean   :  891.1   Mean   : 13466
##  3rd Qu.:  1000.0   3rd Qu.:  748.2   3rd Qu.: 17659
##  Max.   :137000.0   Max.   :19000.0   Max.   :656730
##  facenumber_in_poster    genre
##  Min.   : 0.000       Length:940
##  1st Qu.: 0.000       Class :character
##  Median : 1.000       Mode  :character
##  Mean   : 1.624
##  3rd Qu.: 2.000
##  Max.   :31.000
```

Given the range of gross and budget we can switch to working in unit numbers by dividing by a million.

```
imdb<- imdb%>%
        mutate(gross = gross/1000000,
               budget = budget/1000000)
```

Exploratory Data Analysis

We are interested in predicting the gross of a movie basic on its characteristics. First let's analyze the target variable.

```
basicStats(imdb%>%dplyr::select(gross))
```

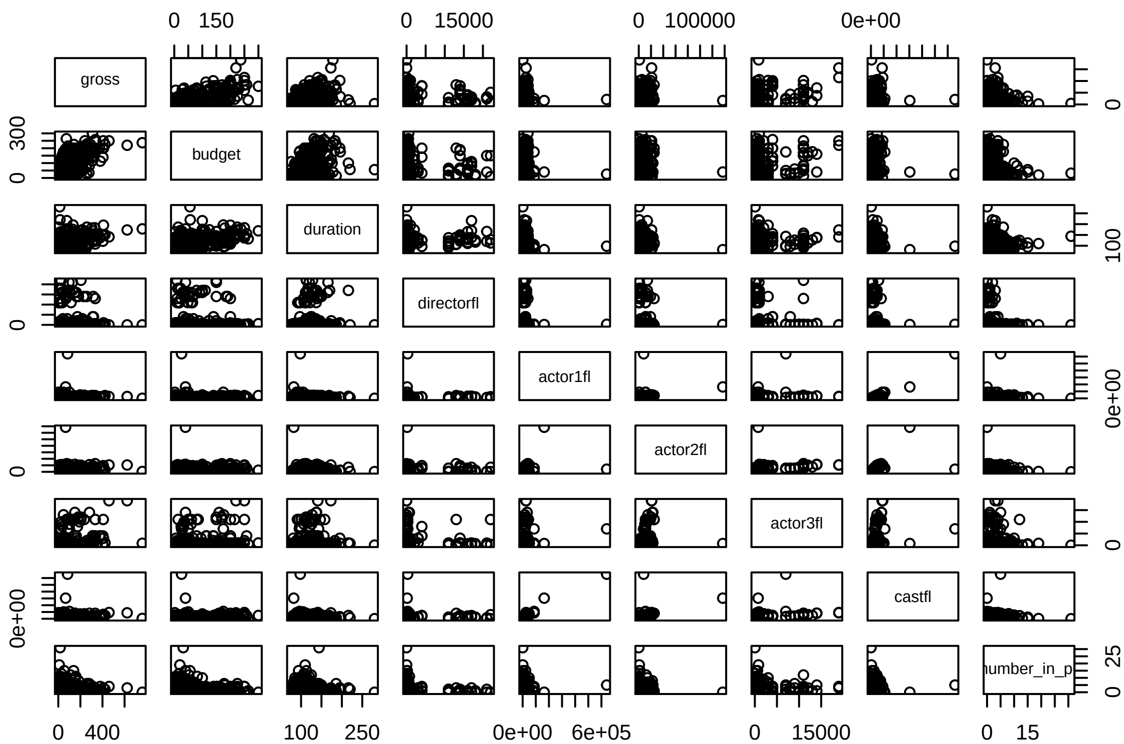
```
##                     gross
## nobs           940.000000
## NAs              0.000000
## Minimum          0.003330
## Maximum        760.505847
## 1. Quartile     11.816543
## 3. Quartile     70.756664
## Mean            57.813237
## Median          33.428175
## Sum          54344.442575
## SE Mean          2.515068
## LCL Mean        52.877432
## UCL Mean        62.749041
## Variance      5946.031921
## Stdev           77.110518
```

```
## Skewness        3.099129
## Kurtosis       14.530608
```

Using the basicStats we obtain the excess kurtosis, $K(X) - 3$ and we see that we have a considerable positive one and that it has a right skewed distribution. So, it is not normal. We should consider that the skewness and kurtosis could be due to outliers.

We look at an overview of the relationship between all variables in our dataset:

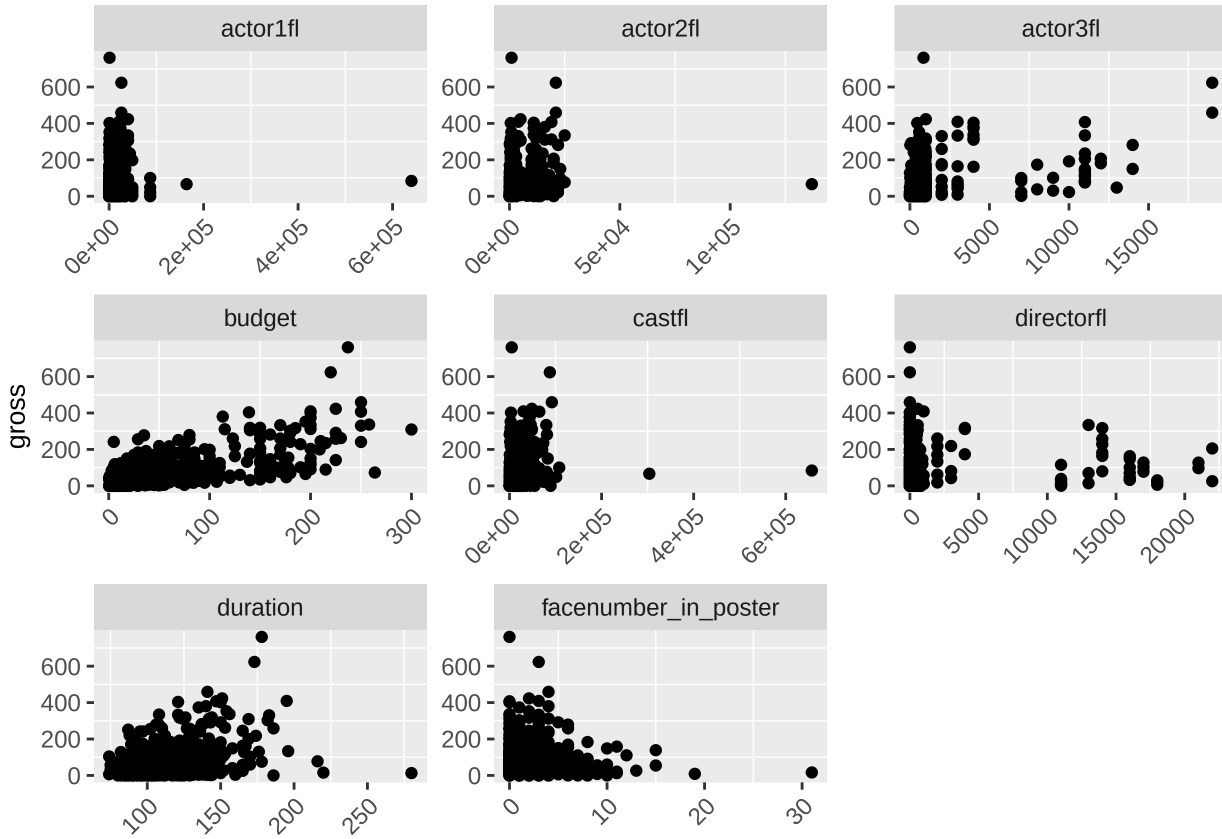
```r
pairs(~.,imdb %>% select(-c(movietitle,genre, titleyear)))
```



In this plot we observe that some variables seem to be correlated, such as *actor*1*fl* with *castfl*, as well as, *budget* with *duration*. However, this correlation would present a problem, in the form of multicolinearity, in the case that both variables were to be included in the final model.

We now look closer into the relation between *gross* and all the numerical variables.

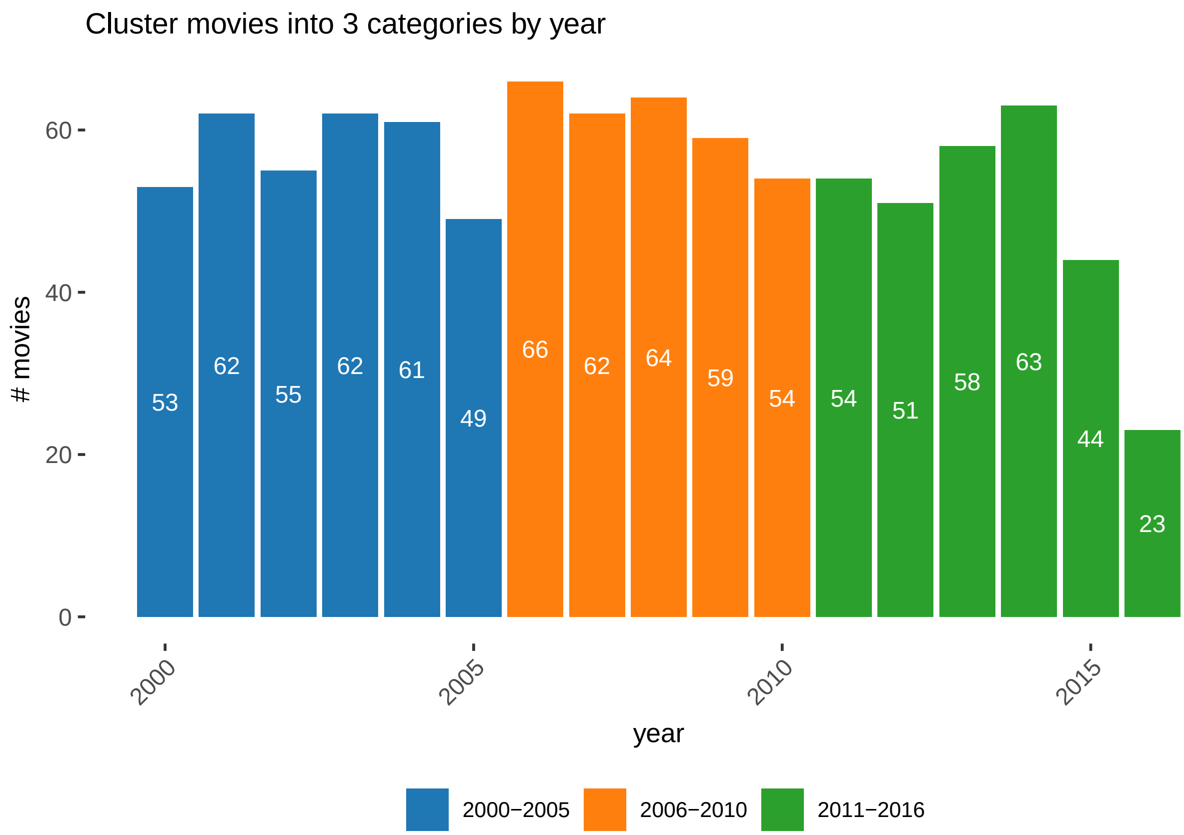
```r
imdb %>%
  select(-c(movietitle,genre, titleyear)) %>%
  gather(-gross, key = "some_var_name", value = "some_value_name") %>%
  ggplot(aes(x = some_value_name, y = gross)) +
  geom_point() +
  facet_wrap(~ some_var_name, scales = "free")+
  ggstyleFonts +
  theme(panel.grid.major = element_blank(),
        axis.title.x = element_blank())
```



We observe more of a linear relation between the pairs of gross and budget, as well as with, duration. We can't discern any pattern between the pairs of gross and the Facebook variables: $directorfl, actor1fl, actor2fl, actor3fl, castfl$. The $actor3fl, directorfl$ could be separated in 2 clusters at the cutoff point of 5000 likes and for the latter at the cutoff point of 10000 likes.

We create a categorial variable ($yearcat$) with 3 levels: 2000-2005, 2006-2010 and 2011-2016 based on the $titleyear$ of the movie.

```
imdb <- imdb%>%
        mutate(yearcat = ifelse(titleyear< 2006, "2000-2005",
                          ifelse(titleyear< 2011, "2006-2010","2011-2016" )))
ggplot(imdb%>%
         group_by(titleyear,yearcat)%>%
         summarise(movies = n()) ,
       aes(x=titleyear, y=movies, fill= yearcat))+
  geom_bar(stat="identity") +
  geom_text(aes(label= movies),
            position=position_stack(vjust=0.5), colour="white" ,size=3) +
  scale_fill_d3(name="") +
  labs(y = "# movies", x= "year" , title = "Cluster movies into 3 categories by year") +
  ggstyle+
  theme(legend.position="bottom")
```



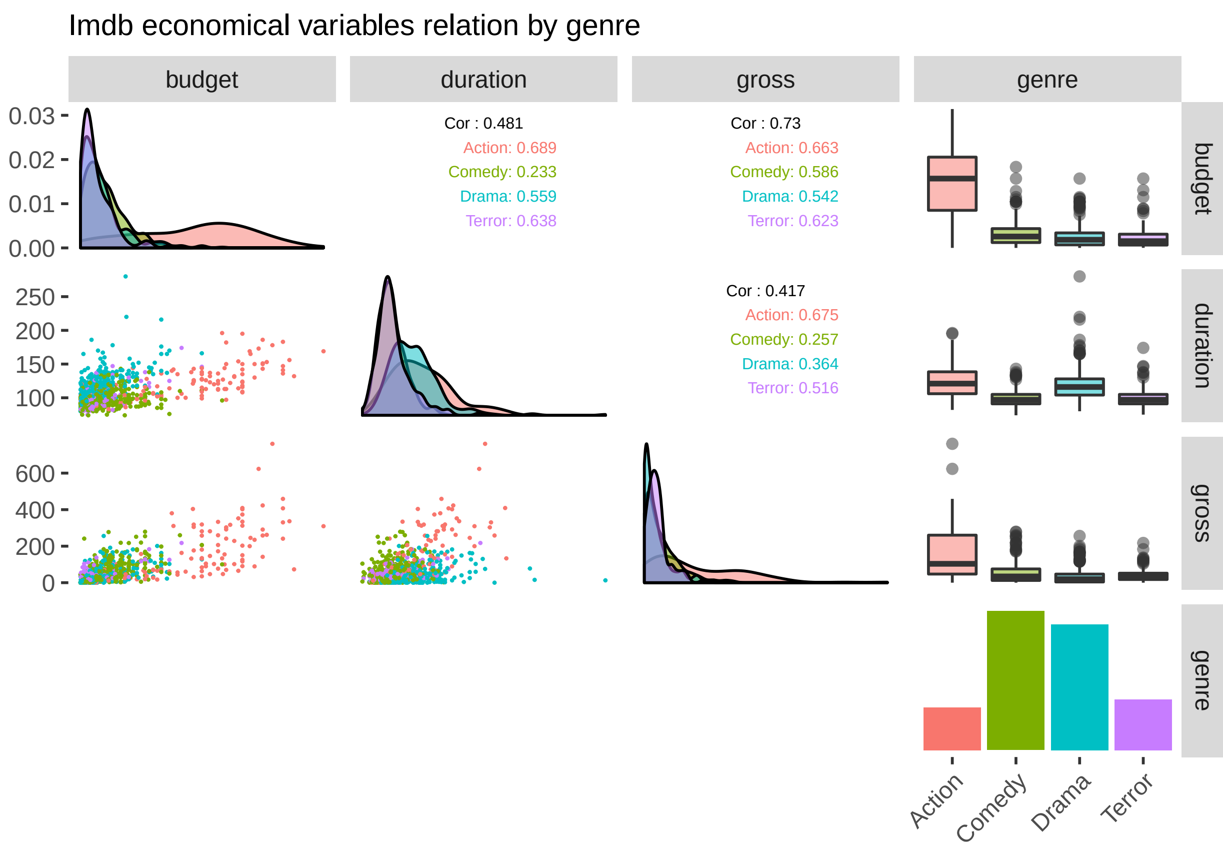

```r
imdb%>%
  group_by(titleyear,yearcat)%>%
  summarise(movies = n()) %>%
  group_by(yearcat) %>%
  mutate(avgMovies = mean(movies)) %>%
  summarise(movies = sum(movies),
            avgMovies = max(avgMovies)) %>%
  mutate(pcn = movies/sum(movies))
```

```
## # A tibble: 3 x 4
##   yearcat   movies avgMovies   pcn
##   <chr>      <int>     <dbl> <dbl>
## 1 2000-2005    342      57   0.364
## 2 2006-2010    305      61   0.324
## 3 2011-2016    293      48.8 0.312
```

The movies are roughly uniformly distributed between the three categories. However, on average more movies were released between the years 2006 and 2010. In addition, based on the significant difference between 2016 and all the previous years it is highly probable that we don't have data for the whole year. So, we have two categorical variables: the year category and the genre. Let's see how the economical variables relates to *genre*.

```r
imdb %>%
  select(c(budget,duration,gross,genre)) %>%
  ggpairs(.,
          title = "Imdb economical variables relation by genre",
          mapping = ggplot2::aes(colour=genre),
          lower = list(#continuous = wrap("smooth", alpha = 0.3, size=0.1),
                       continuous = wrap("points", size=0.1),
```

```r
                     discrete = "blank", combo="blank"),
        diag = list(discrete="barDiag",
                    continuous = wrap("densityDiag", alpha=0.5 )),
        upper = list(combo = wrap("box_no_facet", alpha=0.5),
                     continuous = wrap("cor", size=2, alignPercent=0.8))) +
  ggstyle+
  theme(panel.grid.major = element_blank())
```



We observe two outliers in the Action genre based on their *gross* value, which turn out to be blockbusters.

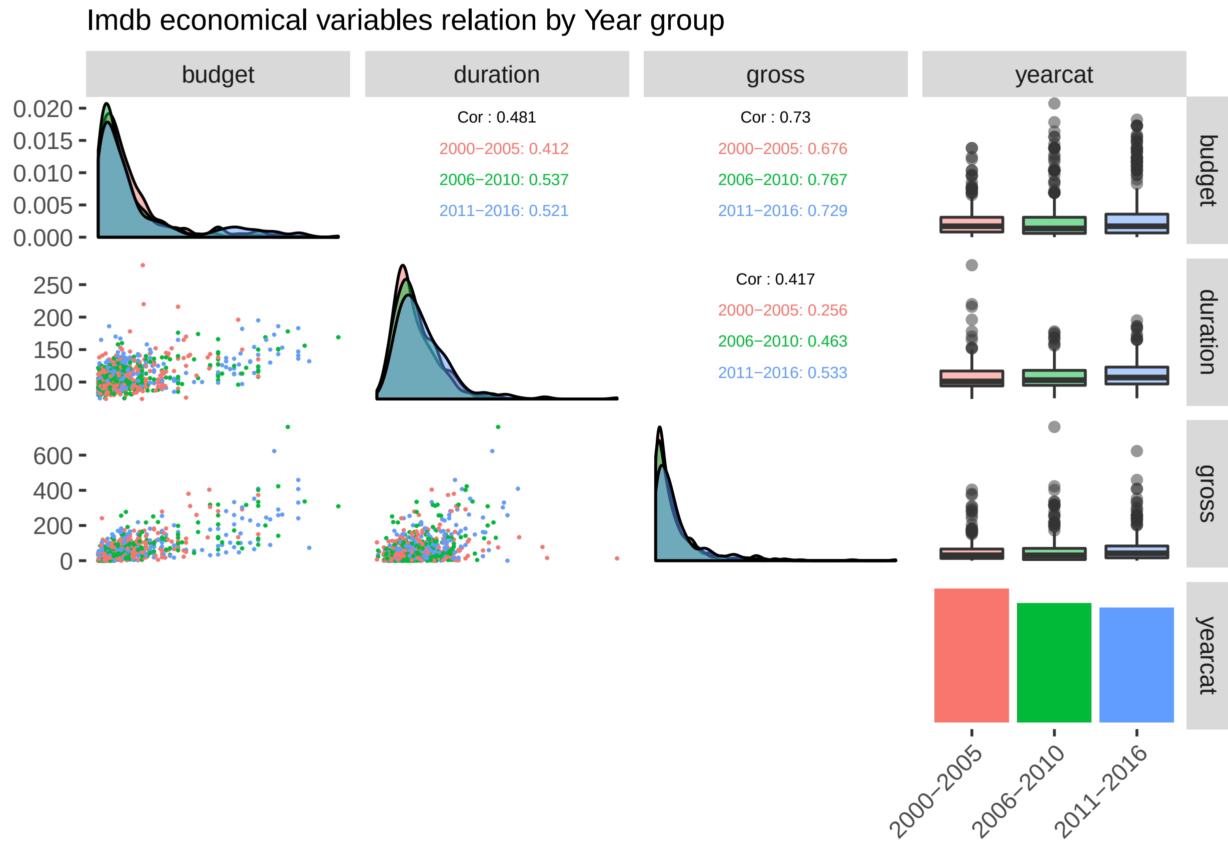
```r
imdb %>% dplyr::filter(gross> 600) %>% pull(movietitle)
```

```
## [1] "Avatar"        "The Avengers"
```

The distribution of *gross* for the Action genre is skewed to the right and has a higher IQR than the rest of the genres. However, it is also the genre with the smallest number of movies. Similarly, *budget* has excess kurtosis with more heavier tails than *gross* especially for the action movies. In the linear relation that we observed before between *gross* and *budget* we add now the genre which confirms this relation, particularly more for the Action movies.

```r
imdb %>%
  select(c(budget,duration,gross,yearcat)) %>%
  ggpairs(.,
        title = "Imdb economical variables relation by Year group",
        mapping = ggplot2::aes(colour=yearcat),
        lower = list(#continuous = wrap("smooth", alpha = 0.3, size=0.1),
          continuous = wrap("points", size=0.1),
          discrete = "blank", combo="blank"),
        diag = list(discrete="barDiag",
```

```r
                        continuous = wrap("densityDiag", alpha=0.5 )),
           upper = list(combo = wrap("box_no_facet", alpha=0.5),
                         continuous = wrap("cor", size=2, alignPercent=0.8))) +
  ggstyle+
  theme(panel.grid.major = element_blank())
```



On the other hand, we don't observe any differences between the different years.

Fit complete model

We first fit the complete model including as predictors, all the numerical variables, the two categorical variables, the categorical-categorical interactions and the interaction between numerical-categorical.

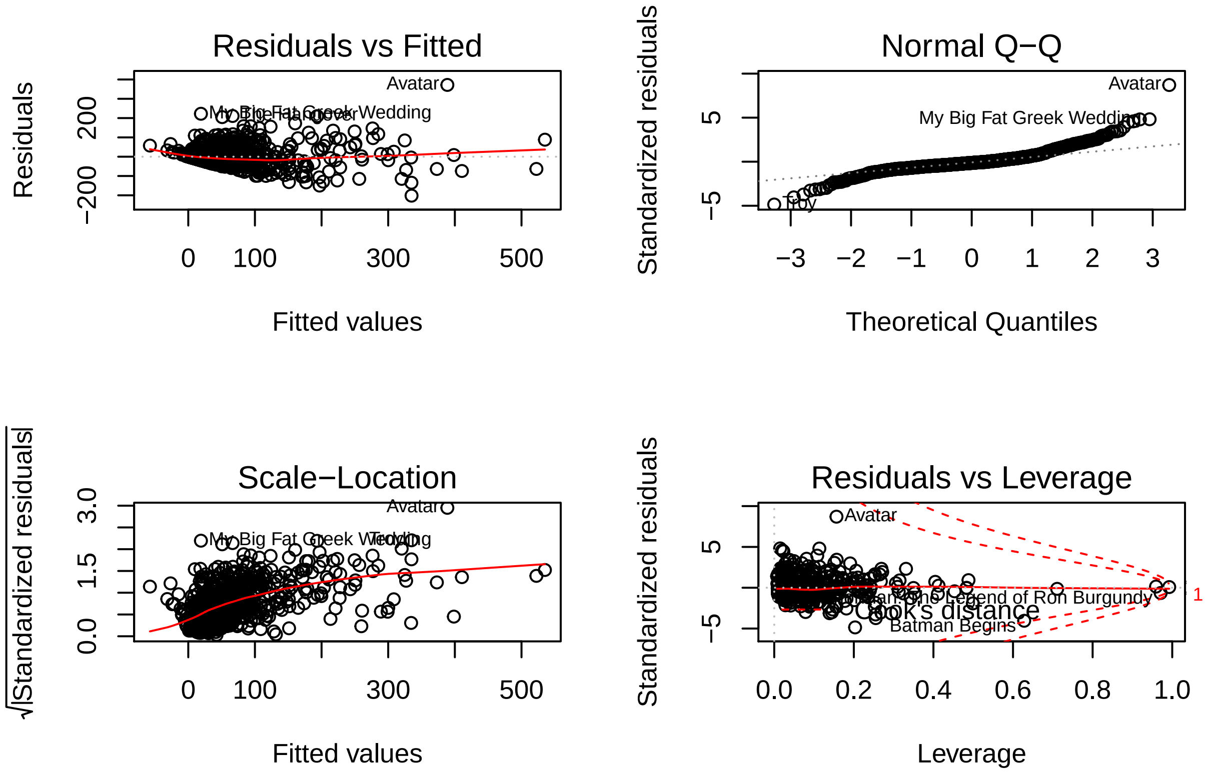
```r
rownames(imdb) <- imdb$movietitle
imdb <- imdb %>% dplyr::select(-c(movietitle,titleyear))
mc<-lm(gross~(.-genre-yearcat)*(genre+yearcat)+genre:yearcat, imdb)

glance(mc)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <int>  <dbl> <dbl>
## 1     0.659         0.636  46.5      28.8 5.50e-166    60 -4912. 9946.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

Roughly 64% of the variance found in the response variable (*gross*) can be explained by the predictor variables. The obtained p-value (*omnibustest*) indicates that the overall model is significant.

```r
op<-par(mfrow=c(2,2))
plot(mc)
```




```r
par(op)
```

From the *Normal Q-Q* plot we see that there is assymetry in the distribution and we can conclude that normality of the residuals is not met. From the *Residuals vs Fitted$ plot*, we seek to validate the assumption of homoskedasticity, which does not seem to hold in our case. What's more, we observe, a non random distribution of the points along the $y - axis$. All in all, we can't validate this model. We look into this with more detail with the final model.

Select significant variables

We use the stepwise procedure, by using the *BIC* criterion, to select the significant variables. Since our objective is the prediction of *gross* revenue per movie, we choose as starting point the complete model, in contrast to starting form the null which would result in a simpler model, albeit loss in predictability.

```r
summary(m1<-step(mc,direction="both",k=log(nrow(imdb)), trace = 0))
```

```
##
## Call:
## lm(formula = gross ~ budget + duration + actor1fl + actor2fl +
##     castfl + genre + yearcat + budget:yearcat + duration:genre +
##     actor1fl:genre + castfl:genre, data = imdb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -220.59  -22.63   -7.31   14.33  364.11
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -2.257e+02  2.367e+01  -9.538  < 2e-16 ***
```

```
## budget                    9.984e-01  9.146e-02  10.916  < 2e-16 ***
## duration                  2.031e+00  2.122e-01   9.568  < 2e-16 ***
## actor1fl                 -7.284e-03  8.989e-04  -8.103 1.70e-15 ***
## actor2fl                 -3.135e-03  1.037e-03  -3.022  0.00258 **
## castfl                    5.718e-03  6.329e-04   9.036  < 2e-16 ***
## genreComedy               1.745e+02  3.146e+01   5.546 3.82e-08 ***
## genreDrama                2.218e+02  2.712e+01   8.178 9.58e-16 ***
## genreTerror               2.087e+02  3.618e+01   5.768 1.10e-08 ***
## yearcat2006-2010         -2.321e+00  5.005e+00  -0.464  0.64289
## yearcat2011-2016          1.460e+01  5.144e+00   2.838  0.00464 **
## budget:yearcat2006-2010   3.773e-02  9.337e-02   0.404  0.68624
## budget:yearcat2011-2016  -4.085e-01  9.043e-02  -4.518 7.07e-06 ***
## duration:genreComedy     -1.336e+00  2.942e-01  -4.541 6.33e-06 ***
## duration:genreDrama      -1.970e+00  2.330e-01  -8.452  < 2e-16 ***
## duration:genreTerror     -1.717e+00  3.369e-01  -5.099 4.16e-07 ***
## actor1fl:genreComedy      4.942e-03  1.074e-03   4.602 4.77e-06 ***
## actor1fl:genreDrama       3.686e-03  1.133e-03   3.254  0.00118 **
## actor1fl:genreTerror      3.622e-03  1.384e-03   2.616  0.00903 **
## castfl:genreComedy       -3.340e-03  7.326e-04  -4.559 5.84e-06 ***
## castfl:genreDrama        -2.189e-03  7.194e-04  -3.043  0.00241 **
## castfl:genreTerror       -2.311e-03  8.920e-04  -2.590  0.00974 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.94 on 918 degrees of freedom
## Multiple R-squared:  0.6378, Adjusted R-squared:  0.6295
## F-statistic: 76.96 on 21 and 918 DF,  p-value: < 2.2e-16
```

Similarly to the complete model, we see that roughly 63% of the variance can be explained and the obtained p-value indicates that the overall model is significant.

In addition, we see that neither $actor3fl$ nor $directorfl$ are included in the model and hence, we could consider categorizing these variables as it was mentioned in the exploratory analysis and see whether we obtain a better model.

When dealing with categorical variables we should use the *Anova* method. The $p-value$ obtained will allow us to say if the interaction variables are significant.

```r
car::Anova(m1)
```

```
## Anova Table (Type II tests)
##
## Response: gross
##                 Sum Sq  Df  F value    Pr(>F)
## budget          451563   1 204.9587 < 2.2e-16 ***
## duration         57495   1  26.0962 3.951e-07 ***
## actor1fl        105109   1  47.7078 9.236e-12 ***
## actor2fl         20121   1   9.1325 0.0025808 **
## castfl          157997   1  71.7127 < 2.2e-16 ***
## genre            57406   3   8.6853 1.099e-05 ***
## yearcat           1002   2   0.2275 0.7965685
## budget:yearcat  100097   2  22.7163 2.349e-10 ***
## duration:genre  161349   3  24.4114 3.339e-15 ***
## actor1fl:genre   46983   3   7.1083 0.0001007 ***
```

```
## castfl:genre     47668   3   7.2119 8.709e-05 ***
## Residuals      2022528 918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the interaction variables *budget:yearcat*, *duration:genre*, *actor1fl:genre* and *castfl:genre* are significant, so we keep them in our model. Nevertheless, the variable *yearcat* seems to not be significant, so we could remove it from our model. **PREGUNTAR SI TIENE SENTIDO QUEDARSE CON YEARCAT Y/O SUS INTERACCIONES –> IF THE INTERACTION OF THE VARIABLE WITH OTHERS IS SIGNIFICANT BUT THE VARIABLE ITSELF IS NOT, WE KEEP BOTH VARIABLE AND INTERACTION.**

Check for multicollinearity

Strong associations between predictors will increase standard errors, and therefore increase the probability of a type-II error. The diagnostic that we will use is the variance-inflation factor.

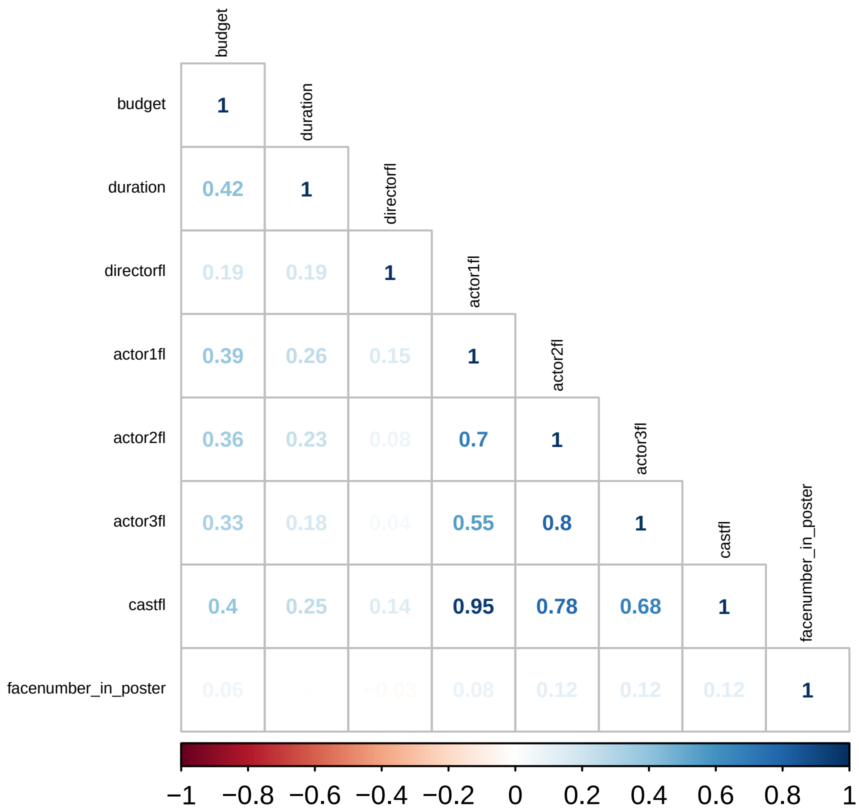
```
car::vif(m1)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## budget             8.800183  1        2.966510
## duration           8.298531  1        2.880717
## actor1fl         198.636593  1       14.093849
## actor2fl          17.143583  1        4.140481
## castfl           132.845959  1       11.525882
## genre          72188.876411  3        6.452753
## yearcat            3.227704  2        1.340366
## budget:yearcat    13.103015  2        1.902579
## duration:genre 81351.402009  3        6.582550
## actor1fl:genre 94415.699560  3        6.747984
## castfl:genre   67158.366057  3        6.375536
```

COMO SE LEE ESTE RESULTADO???? –> LEER COLUMNA "NORMALIZADA" CUANDO TENEMOS CATEGORICAL VARIABLES. SI SALE > 5 ES POSIBLE QUE TENGA COLLINIARITY. TRY WITHOUT ONE THE VARIABLES WITH HIGH VIF AND SEE IF THE VIF IMPROVES. IF WE WANT TO BETTER INTERPRET THE RELATIONS WITH THE RESPONSE VARIABLES, WE WANT TO REMOVE THE COLLINIARITY TO BETTER INTERPRET THE COEFFICIENTS, EVEN WITH WORSE R^2.

$castfl$ and $actor1fl, actor2fl$ which are included in the model are correlated, as well as $actor1fl$ and $actor2fl$. Consequently, the coefficients can't be directly interpreted. The following correlations confirm this idea:

```
corr_mat=cor(imdb %>%
              select_if(is.numeric) %>%
               select(-c(gross)),method="s")
corrplot::corrplot(corr_mat,type = "lower", method = "number",
                   tl.col = "black", tl.cex = 0.5,number.cex = 0.7)
```



To proceed we would need to select which of this three correlated variables would result to a better model.

```r
lm.actor1 <- update(m1,.~.-(castfl+actor2fl))
lm.actor2 <- update(m1,.~.-(castfl+actor1fl))
lm.cast <- update(m1,.~.-(actor1fl+actor2fl))

glance(lm.actor1)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <int>  <dbl> <dbl>
## 1     0.634         0.626  47.1      79.6 1.01e-184    21 -4945. 9934.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

```r
glance(lm.actor2)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <int>  <dbl> <dbl>
## 1     0.638         0.629  46.9      77.0 1.03e-185    22 -4941. 9927.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

```r
glance(lm.cast)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <int>  <dbl> <dbl>
## 1     0.634         0.626  47.1      79.6 1.01e-184    21 -4945. 9934.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

Since we do not obtain a siginificantly better model with any of the variables, we conclude we need other information, such as domain expert knowledge, to decide which predictor to keep. In our case, we decide to keep *castfl* as its and added variable of the likes of the whole movie cast (it includes the other measures).

```
m1 <- update(m1,.~.-actor1fl-actor2fl)
```

We check the *Anova* of the new model to see how it has changed.

```
car::Anova(m1)
```

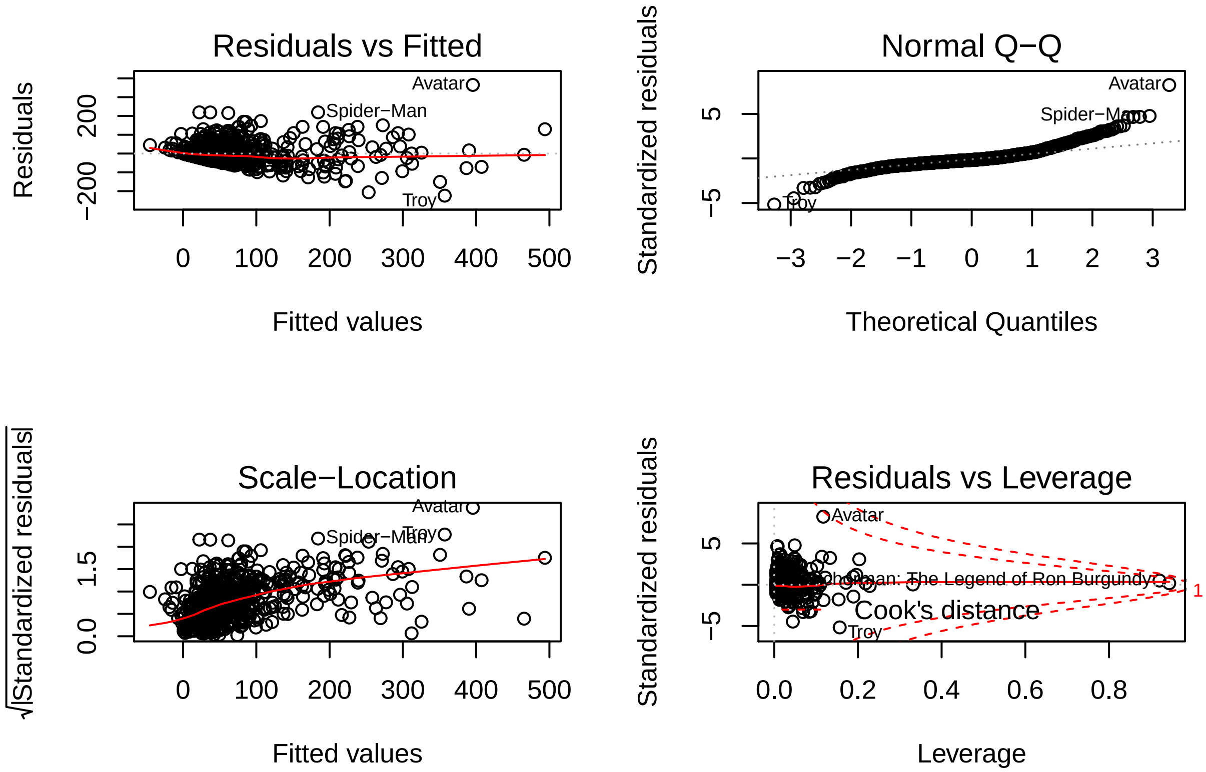
```
## Anova Table (Type II tests)
##
## Response: gross
##                  Sum Sq  Df F value    Pr(>F)
## budget           458768   1 206.403 < 2.2e-16 ***
## duration          60467   1  27.204 2.262e-07 ***
## castfl           127264   1  57.257 9.280e-14 ***
## genre             74532   3  11.177 3.295e-07 ***
## yearcat             827   2   0.186    0.8303
## budget:yearcat    93148   2  20.954 1.262e-09 ***
## duration:genre   161420   3  24.208 4.407e-15 ***
## genre:actor1fl   139159   4  15.652 2.135e-12 ***
## castfl:genre     100795   3  15.116 1.310e-09 ***
## Residuals       2042648 919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
car::vif(m1)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## budget         8.799477e+00  1        2.966391
## duration       8.293560e+00  1        2.879854
## castfl         6.684936e+01  1        8.176146
## genre          7.207881e+04  3        6.451112
## yearcat        3.223282e+00  2        1.339907
## budget:yearcat 1.299691e+01  2        1.898716
## duration:genre 8.127410e+04  3        6.581507
## genre:actor1fl 1.385808e+05  4        4.392511
## castfl:genre   4.405568e+04  3        5.942926
```

Validate model's assumptions

```
op<-par(mfrow=c(2,2))
plot(m1)
```




```r
par(op)
```

Model interpretation

```r
# plot(allEffects(m1))
```

TO INTERPRET VARIABLES WITH INTERACTIONS, WE PLOT THE EFFECT OF THE RESPONSE TO THE VARIABLE. WHEN INTERACTION IS WITH A CATEGORICAL VARIABLE THE EFFECTS ARE PLOTS STRATIFIED BY CATEGORICAL VARIABLE LEVEL.