ASM Homework 1

Linear Model for IDMB data

Maria Gkotsopoulou & Ricard Monge Calvo & Amalia Vradi

13/10/2019

```
############################################################
###############       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.

```
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))
```

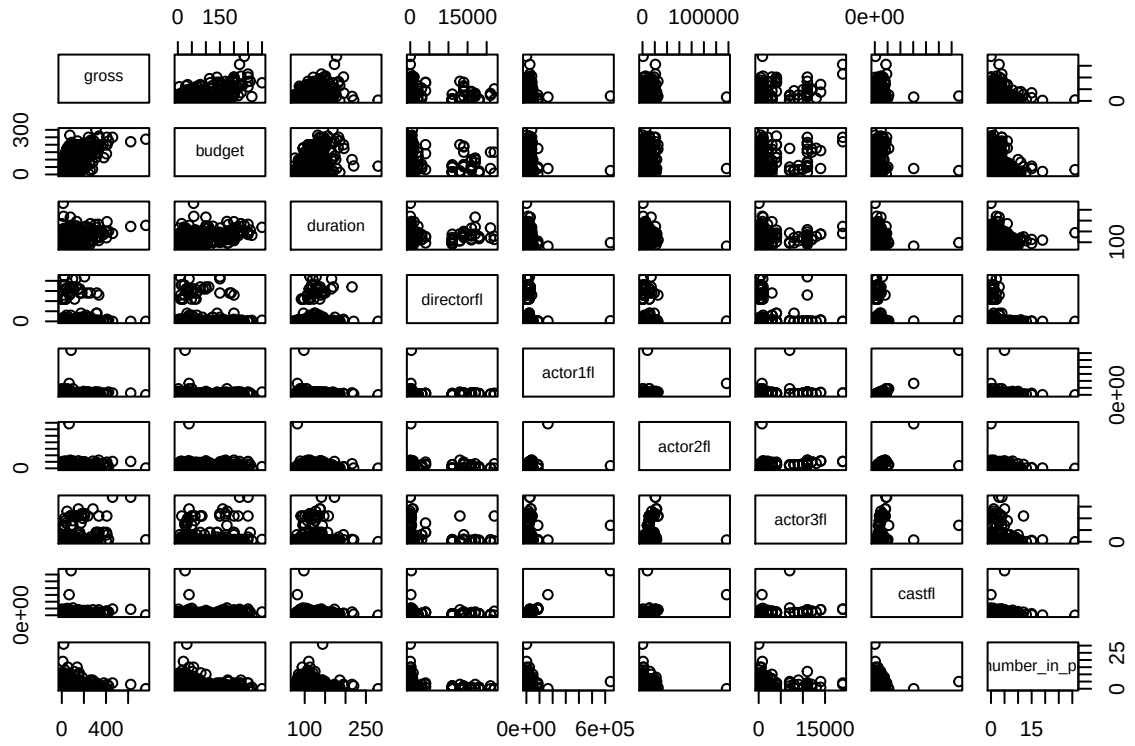
```
##                     gross
## nobs           940.000000
## NAs              0.000000
## Minimum          0.003330
## Maximum        760.505847
## 1. Quartile     11.816543
## 3. Quartile     70.756664
## Mean            57.813237
## Median          33.428175
## Sum          54344.442575
## SE Mean          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:

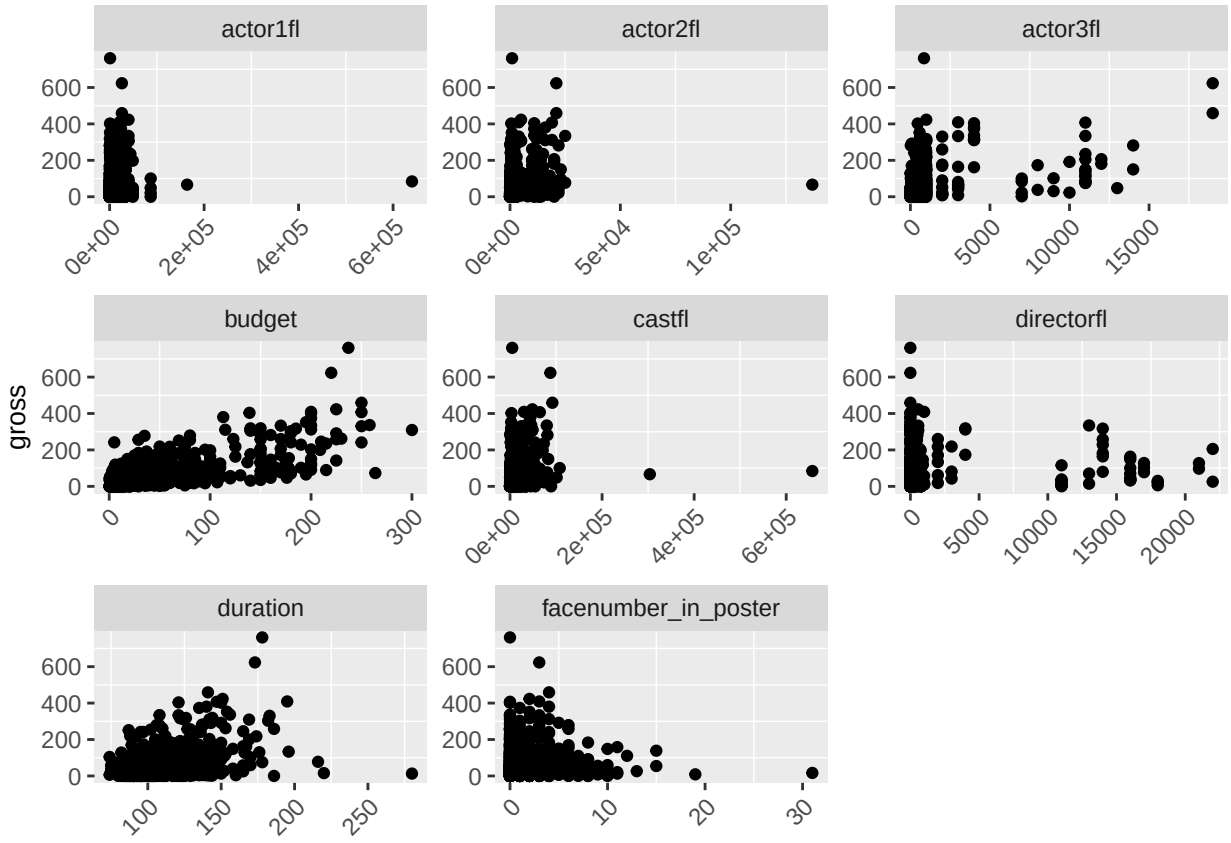
```
pairs(~.,imdb %>% select(-c(movietitle,genre, titleyear)))
```



In this plot we observe that some variables seem to be correlated, such as $actor1fl$ 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.

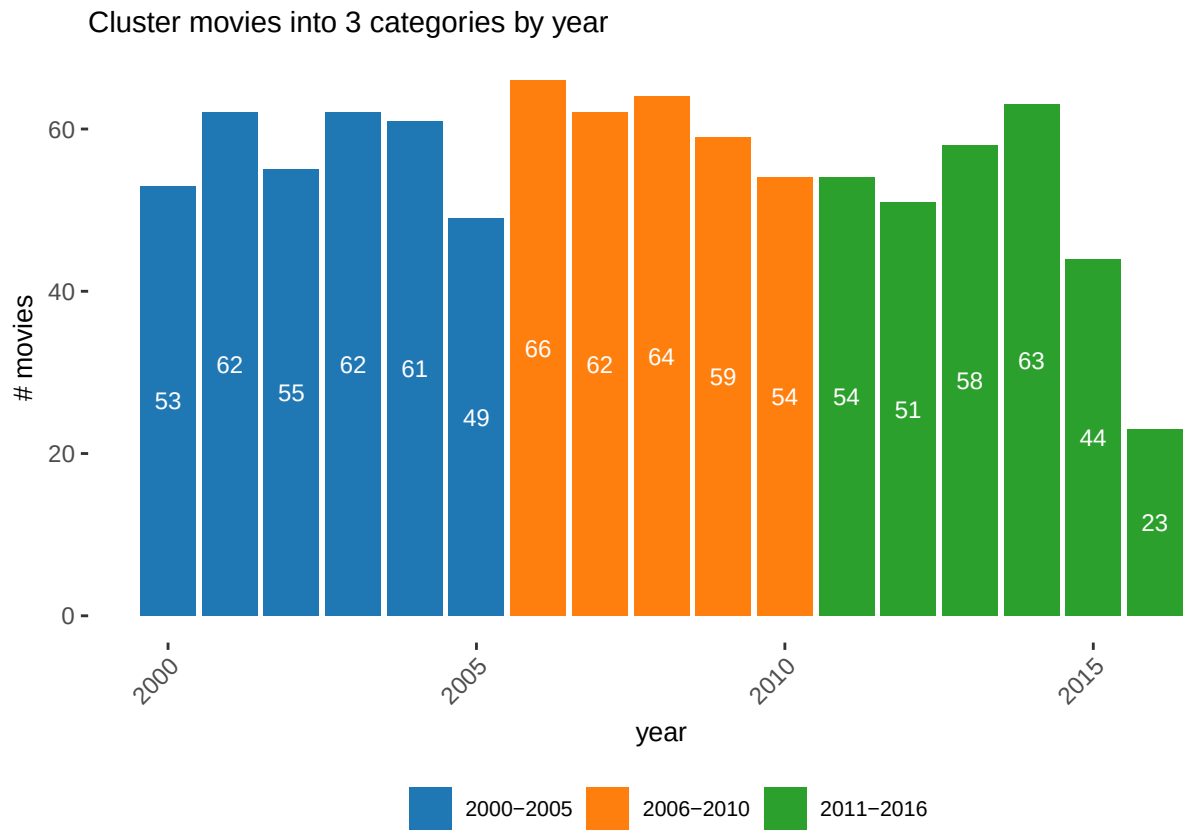
```
imdb %>%
  select(-c(movietitle,genre, titleyear)) %>%
  gather(-gross, key = "some_var_name", value = "some_value_name") %>%
  ggplot(aes(x = some_value_name, y = gross)) +
  geom_point() +
  facet_wrap(~ some_var_name, scales = "free")+
  ggstyleFonts +
  theme(panel.grid.major = element_blank(),
        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: $director fl, actor1 fl, actor2 fl, actor3 fl, cast fl$. The $actor3 fl, director fl$ 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 = as.factor(
                          ifelse(titleyear< 2006, "2000-2005",
                          ifelse(titleyear< 2011, "2006-2010","2011-2016" ))
                          ),
                genre = as.factor(genre))
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")
```



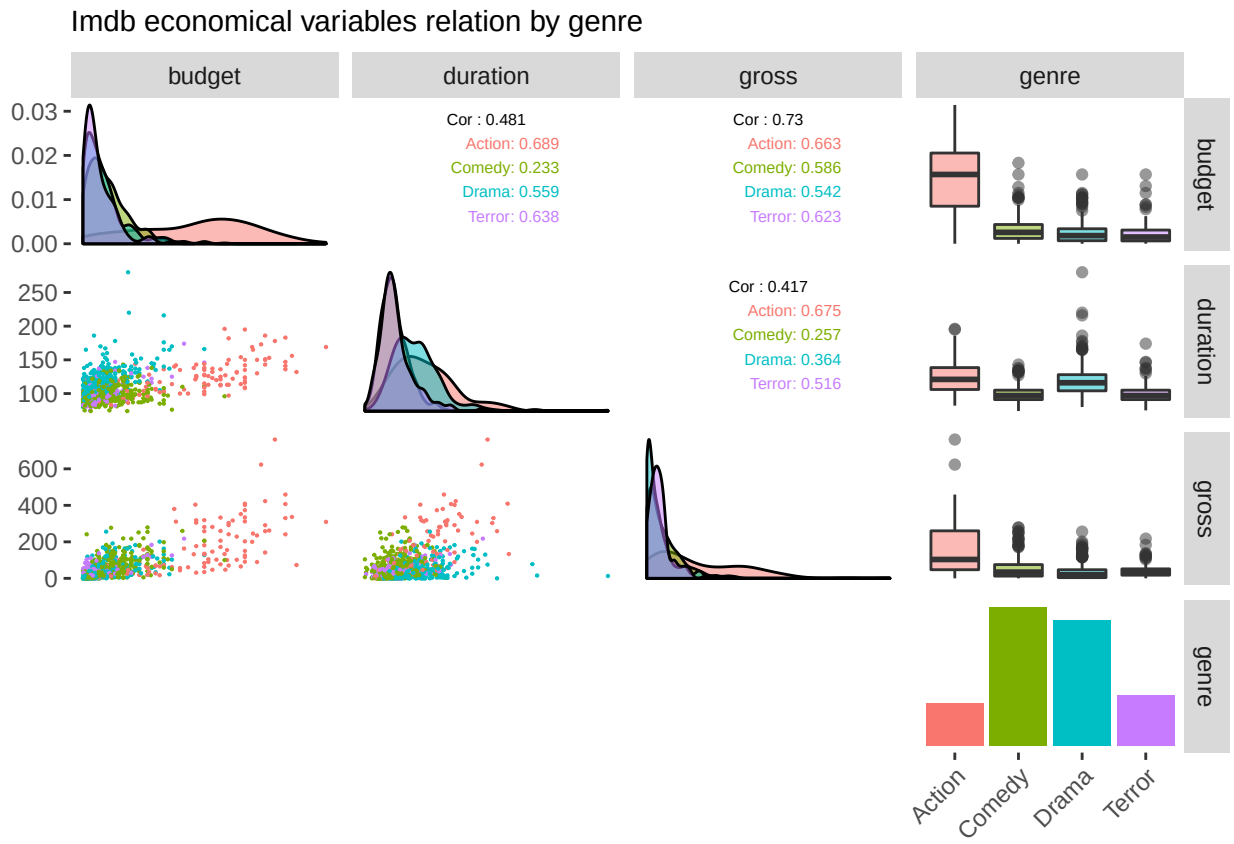

```
imdb%>%
  group_by(titleyear,yearcat)%>%
  summarise(movies = n()) %>%
  group_by(yearcat) %>%
  mutate(avgMovies = mean(movies)) %>%
  summarise(movies = sum(movies),
            avgMovies = max(avgMovies)) %>%
  mutate(pcn = movies/sum(movies))
```

```
## # A tibble: 3 x 4
##   yearcat   movies avgMovies   pcn
##   <fct>      <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*.

```
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),
```

```
                    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.

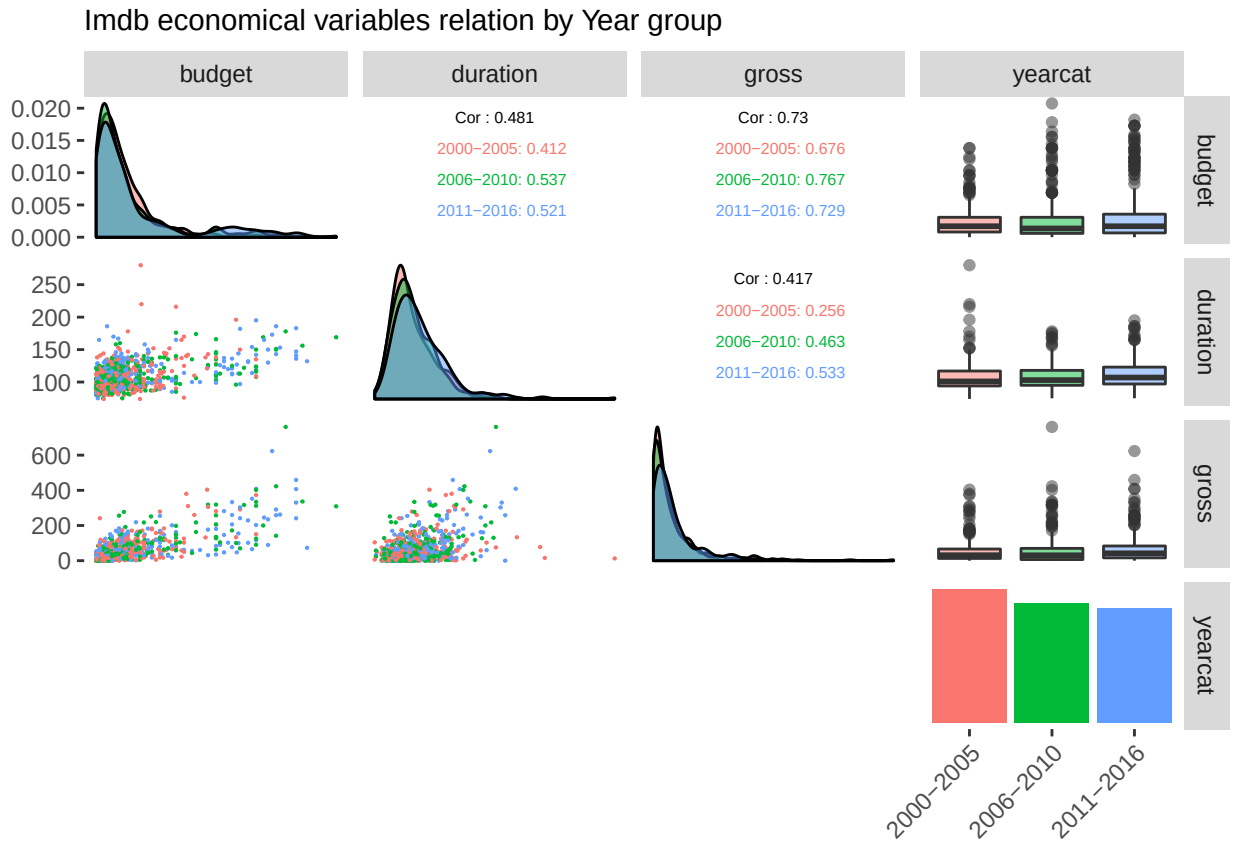
```
imdb %>% dplyr::filter(gross> 600) %>% pull(movietitle)
```

```
## [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.

```
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",
```

```
                    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.

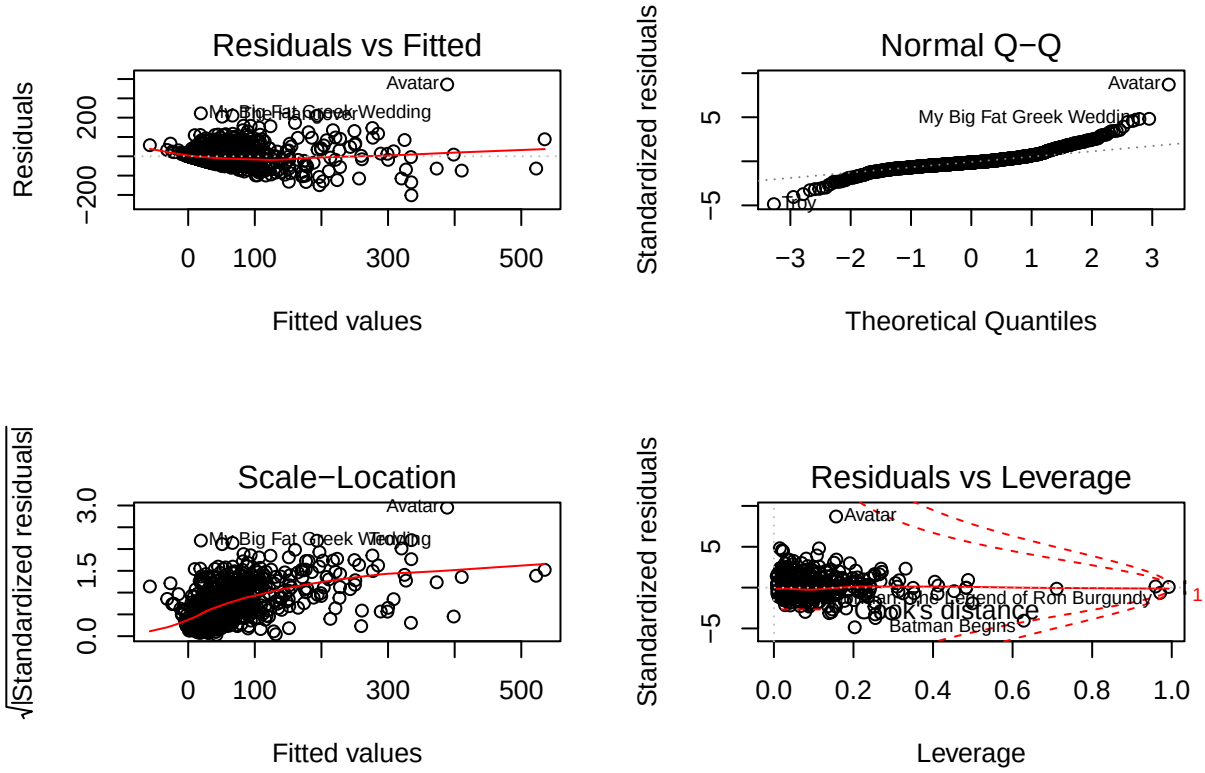
```
rownames(imdb) <- imdb$movietitle
imdb <- imdb %>% dplyr::select(-c(movietitle,titleyear))
mc<-lm(gross~.*(genre+yearcat), imdb)

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 (*Omnibus test*) indicates that the overall model is significant.

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




```
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 *Scale vs Location* plot, we seek to validate the assumption of homoskedasticity, which does not seem to hold in our case. What's more, from the *Residual vs Fitted* plot 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 interpretability of the model we choose as starting point the null model, in contrast to starting form the complete.

```
m0 <- lm(gross~1, imdb)
summary(m1<-step(m0,scope=list(upper=mc), direction="both",
                 k=log(nrow(imdb)), trace = 0))
```

```
##
## Call:
## lm(formula = gross ~ budget + actor3fl + duration + genre + duration:genre,
##     data = imdb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -216.59  -23.62   -8.88   16.16  414.66
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          -2.269e+02  2.408e+01  -9.424  < 2e-16 ***
## budget                8.362e-01  5.992e-02  13.955  < 2e-16 ***
## actor3fl              6.171e-03  8.624e-04   7.156 1.68e-12 ***
## duration              2.075e+00  2.163e-01   9.595  < 2e-16 ***
## genreComedy           1.808e+02  3.229e+01   5.599 2.85e-08 ***
## genreDrama            2.205e+02  2.781e+01   7.927 6.41e-15 ***
## genreTerror           2.116e+02  3.716e+01   5.694 1.67e-08 ***
## duration:genreComedy -1.390e+00  2.980e-01  -4.666 3.53e-06 ***
## duration:genreDrama  -1.936e+00  2.348e-01  -8.246 5.55e-16 ***
## duration:genreTerror -1.719e+00  3.425e-01  -5.017 6.28e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.77 on 930 degrees of freedom
## Multiple R-squared:  0.6039, Adjusted R-squared:    0.6
## F-statistic: 157.5 on 9 and 930 DF,  p-value: < 2.2e-16
```

Contrary to the complete model, we see that just 60% of the variance can be explained, although the obtained p-value indicates that the overall model is significant. However, we see that neither *actor1fl*, *actor2fl*, *castfl*, *directorfl*, *facenumber_in_poster* nor *yearcat* are included. For this reason, we consider exploring the stepwise procedure starting from the complete model.

```
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 this case, we see that *actor3fl* is not included but *actor1fl*, *actor2fl*, *castfl* and *yearcat* are . Likewise, *directorfl* and *facenumber_in_poster* are not included in the model. We decide to continue our analysis with this last 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.

```
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. Although the variable *yearcat* seems to not be significant, we decide to keep it in our model due to its interaction being significant.

Check for multicollinearity

Strong associations between predictors will increase standard errors, and therefore increase the probability of a type-II error, as well as affect the value of the coefficients. In order to detect it in our model, 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
```

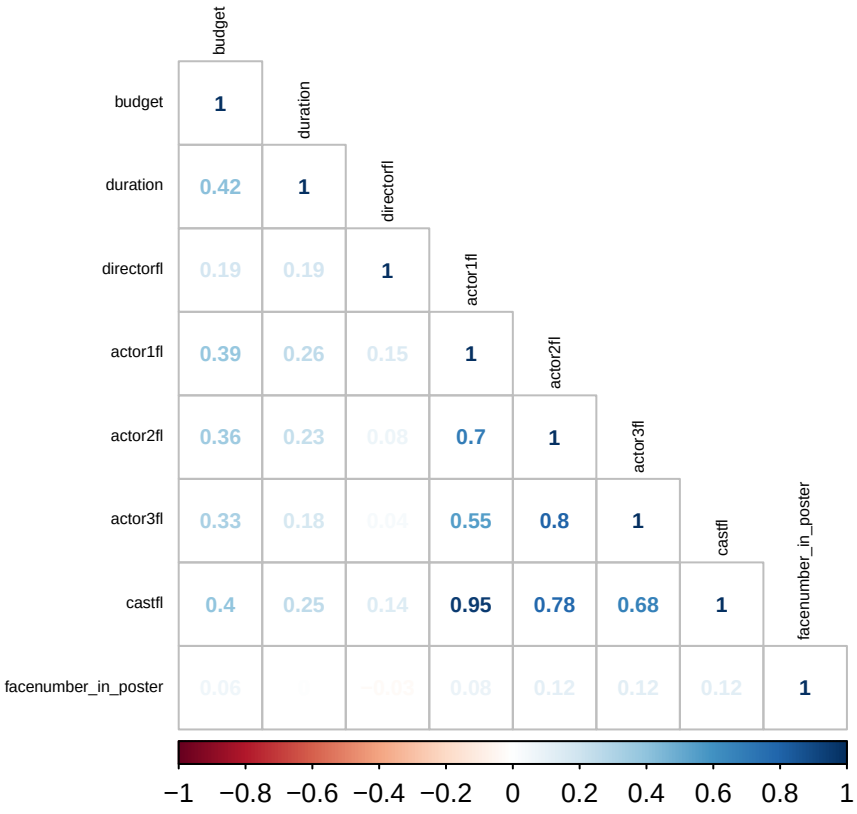
```
## actor1fl          198.636593  1       14.093849
## actor2fl           17.143583  1        4.140481
## castfl            132.845959  1       11.525882
## genre           72188.876411  3        6.452753
## yearcat             3.227704  2        1.340366
## budget:yearcat     13.103015  2        1.902579
## duration:genre  81351.402009  3        6.582550
## actor1fl:genre  94415.699560  3        6.747984
## castfl:genre    67158.366057  3        6.375536
```

From the normalized inflation factor (second column in the previous output) we conclude that *actor1fl* and *castfl* may be causing multicollinearity in the model. This could be perhaps due to a correlation between them as we saw in the exploratory analysis. Consequently, the coefficients can't be directly interpreted.

By definition a categorical variable that is included in an interaction term (as well as the interaction terms themselves) will have a high VIF factor and hence there is no reason to further investigate.

The following figure confirms our suspicions about the correlated variables:

```
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 (*actor1fl*, *actor2fl* or *castfl*) would result in a better model and whether the VIF factor will be corrected.

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

```
lm.cast <- update(m1,.~.-(actor1fl+actor2fl+actor1fl:genre))

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     0.593         0.586  49.6      84.2 1.17e-167    17 -4995. 10026.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

```
car::vif(lm.actor1)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## budget             8.708185  1        2.950963
## duration           8.216017  1        2.866360
## actor1fl          40.551250  1        6.367986
## genre          70086.199392  3        6.421041
## yearcat            3.174304  2        1.334788
## budget:yearcat    12.552239  2        1.882263
## duration:genre 76965.709029  3        6.522031
## actor1fl:genre    66.617846  3        2.013408
```

```
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     0.597         0.591  49.3      105. 2.85e-172    14 -4991. 10012.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

```
car::vif(lm.actor2)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## budget             8.649875  1        2.941067
## duration           8.096795  1        2.845487
## actor2fl           1.093484  1        1.045698
## genre          70443.208040  3        6.426480
## yearcat            3.161674  2        1.333458
## budget:yearcat    12.451950  2        1.878492
## duration:genre 75540.191566  3        6.501741
```

```
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     0.609         0.602  48.6      89.9 1.39e-175    17 -4976. 9988.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

```
car::vif(lm.cast)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## budget             8.715421  1        2.952189
## duration           8.249398  1        2.872177
## castfl            15.691127  1        3.961203
## genre          70346.302232  3        6.425006
```

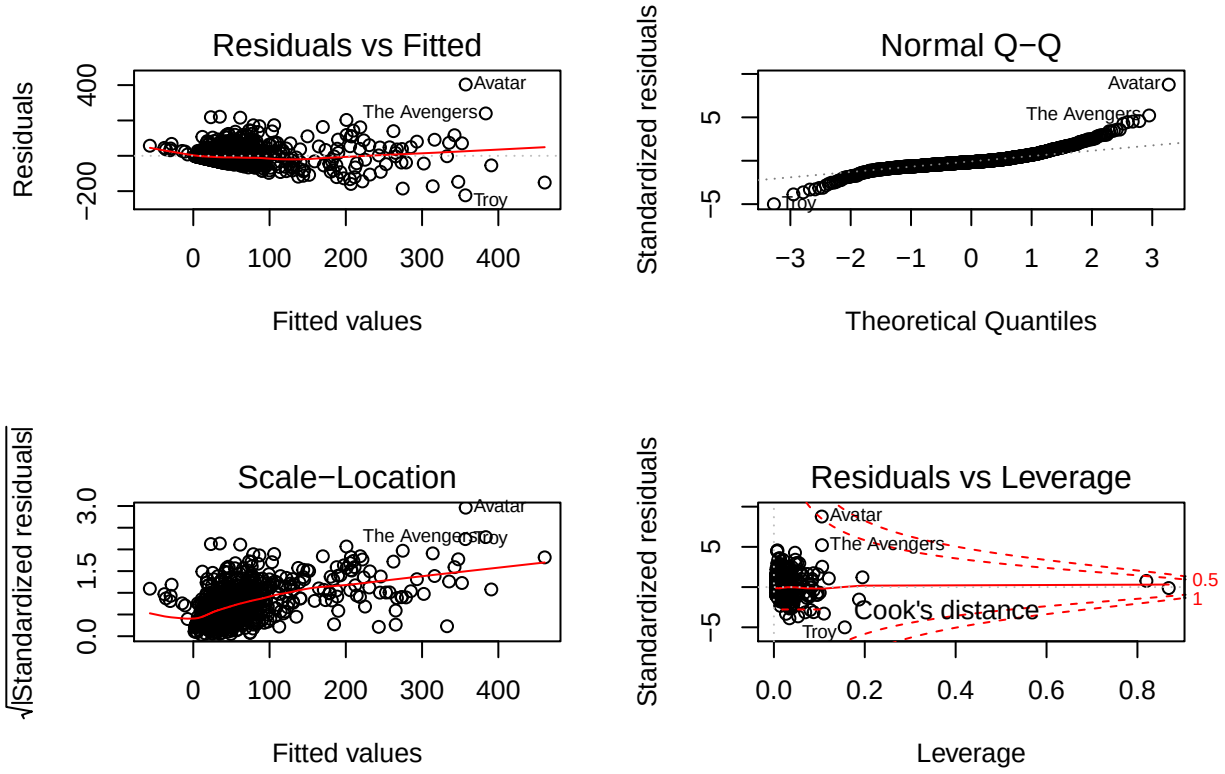
```
## yearcat             3.177529  2        1.335126
## budget:yearcat     12.728134  2        1.888823
## duration:genre  77858.748432  3        6.534583
## castfl:genre       26.744898  3        1.729313
```

Comparing the R^2 between the 3 models we see that we do not obtain a siginificantly better model with any of the variables. On the other hand, looking at the change in the VIF we can conclude that best correction is obtained by the *castfl* model. Given that we lack expert domain knowledge to guide us we decide to keep *castfl* as it is an added variable of the likes of the whole movie cast (it includes the other measures).

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

Validate model's assumptions

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

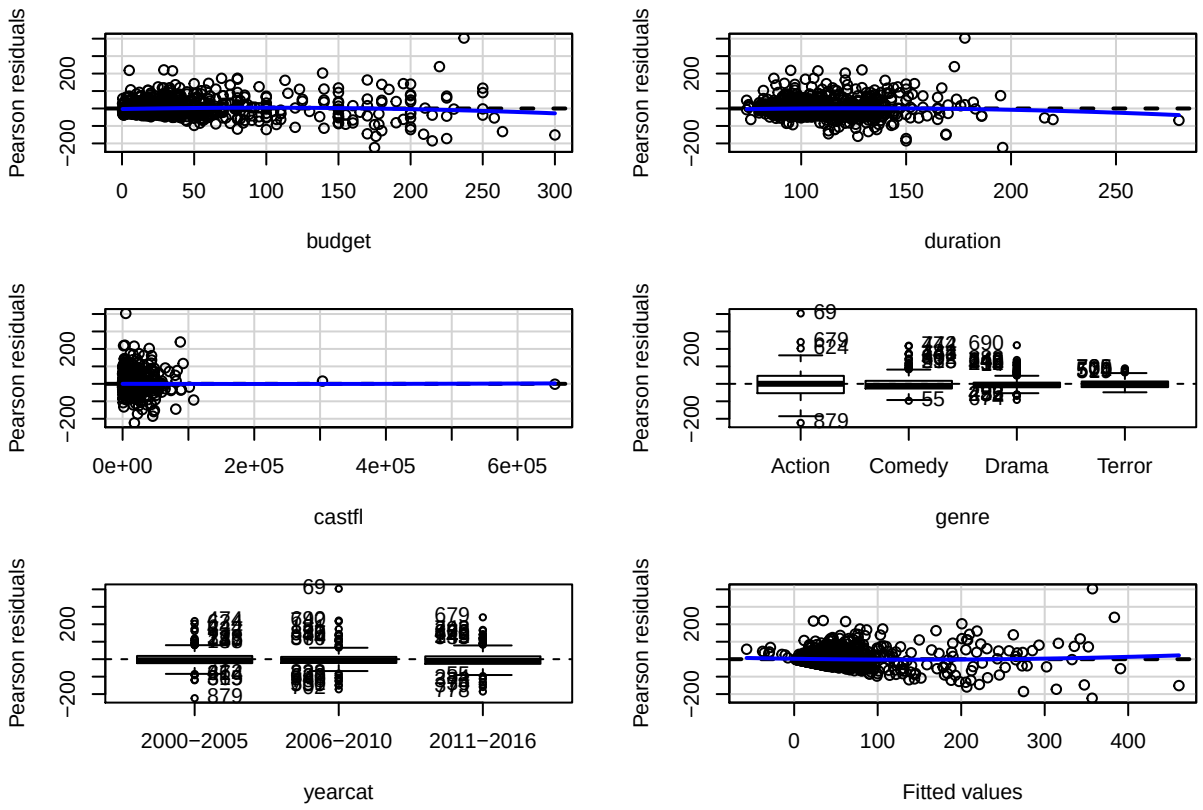



```
par(op)
```

From the *Normal Q-Q* plot we see that there is still assymetry in the distribution and we can conclude that normality of the residuals is not met.

To look at the *Residual vs Fitted* plot in more detail, we plot it for each predictor.

```
car::residualPlots(m1)
```




```
##            Test stat Pr(>|Test stat|)
## budget       -2.3411          0.01944 *
## duration     -1.3569          0.17513
## castfl        0.1328          0.89440
## genre
## yearcat
## Tukey test    1.5148          0.12983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

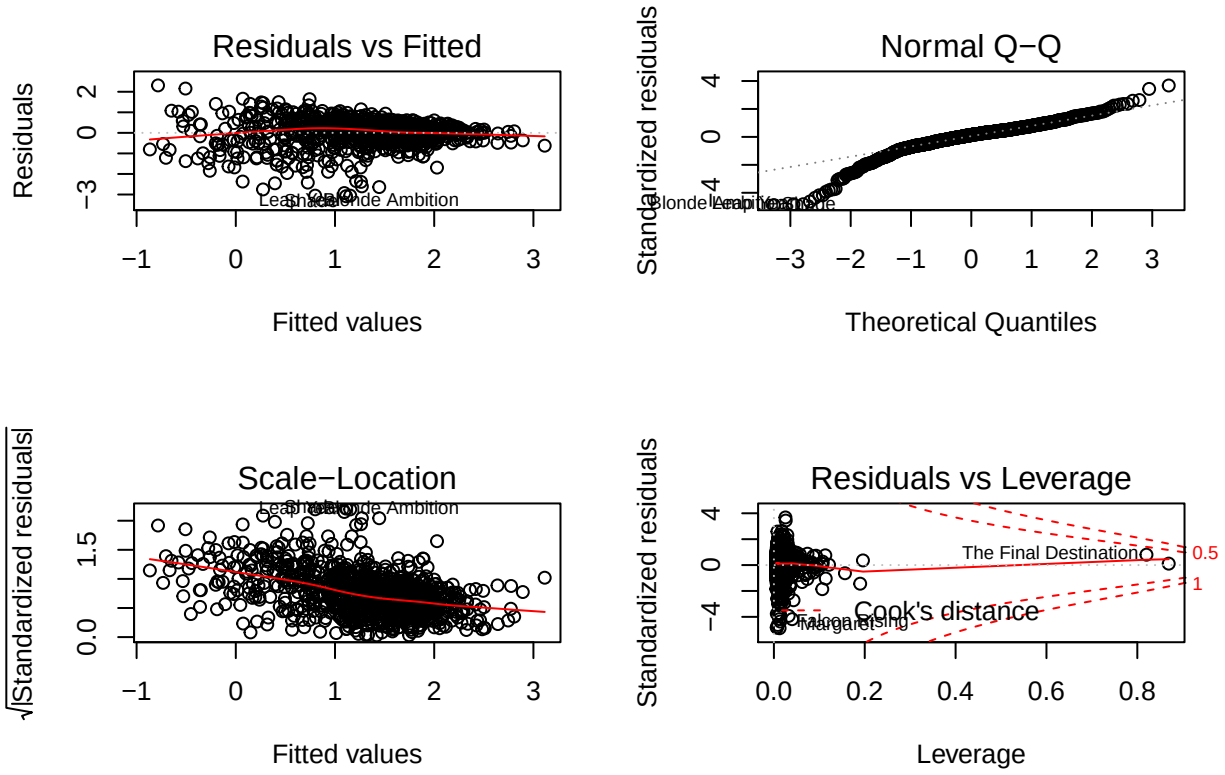
Looking at both the figures, as well as, the curvature test, we conclude that a quadratic term is not needed for any of the variables.

From the *Scale-Location* plot, we seek to validate the assumption of homoskedasticity, which does not seem to hold in our case. Consequently, we consider a log transformation to both *gross* and *budget* measures and see whether the obtained model better meets the assumptions.

```
mlog <- lm(log10(gross) ~ log10(budget) + duration + castfl + genre + yearcat +
             budget:yearcat + duration:genre + castfl:genre,
           data = imdb)
glance(mlog)
```

```
## # 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     0.497         0.487 0.636      53.5 1.46e-124    18  -899. 1836.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

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




```
par(op)
```

We now have better agreement with the model's assumptions at the cost of a decrease in the model's R^2.

Regarding the *Residuals vs Leverage* plot, we see that with our regular final model, without the logs, there are some points that are influential and could be considered to be outliers. However, in the transformed model they do not appear to be so.

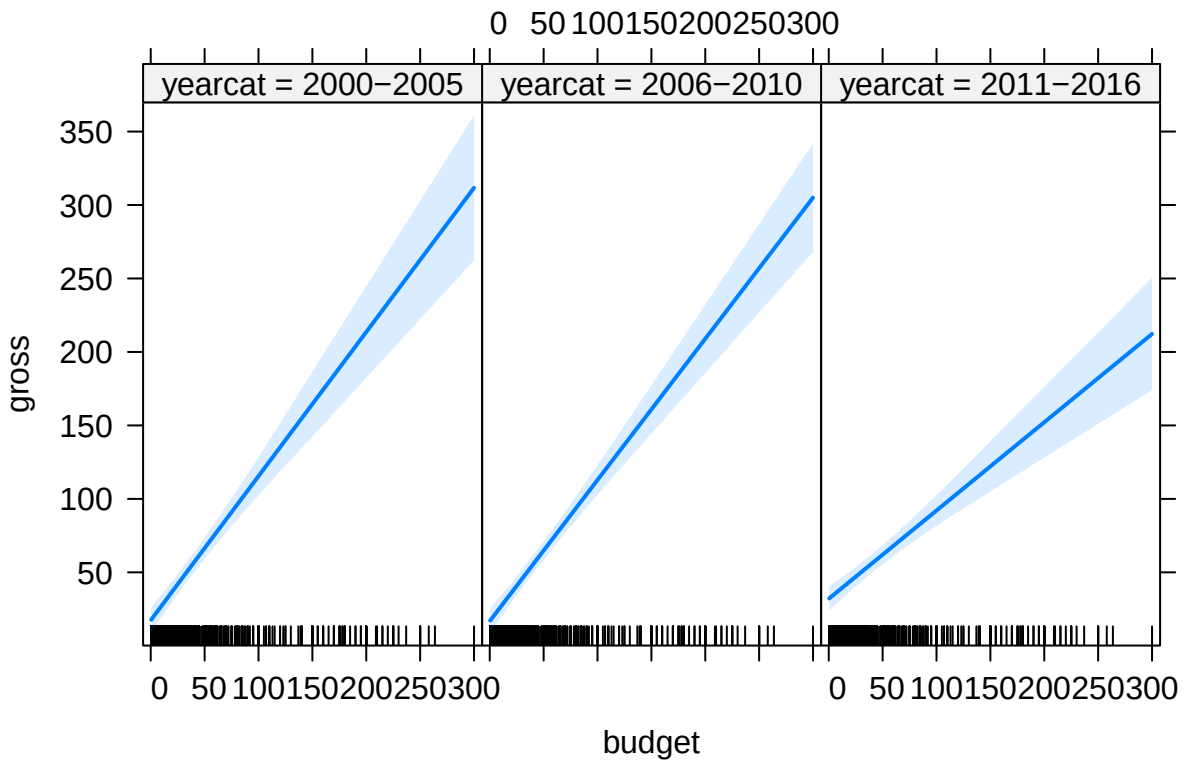
Having mentioned that, we decide to keep the previous model as it has a better fit with our data and a more intuitive interpretation.

Model interpretation

For the model interpretation, we will use the *effect* plots to interpret the coefficients of our final model, taking into account the interaction terms.

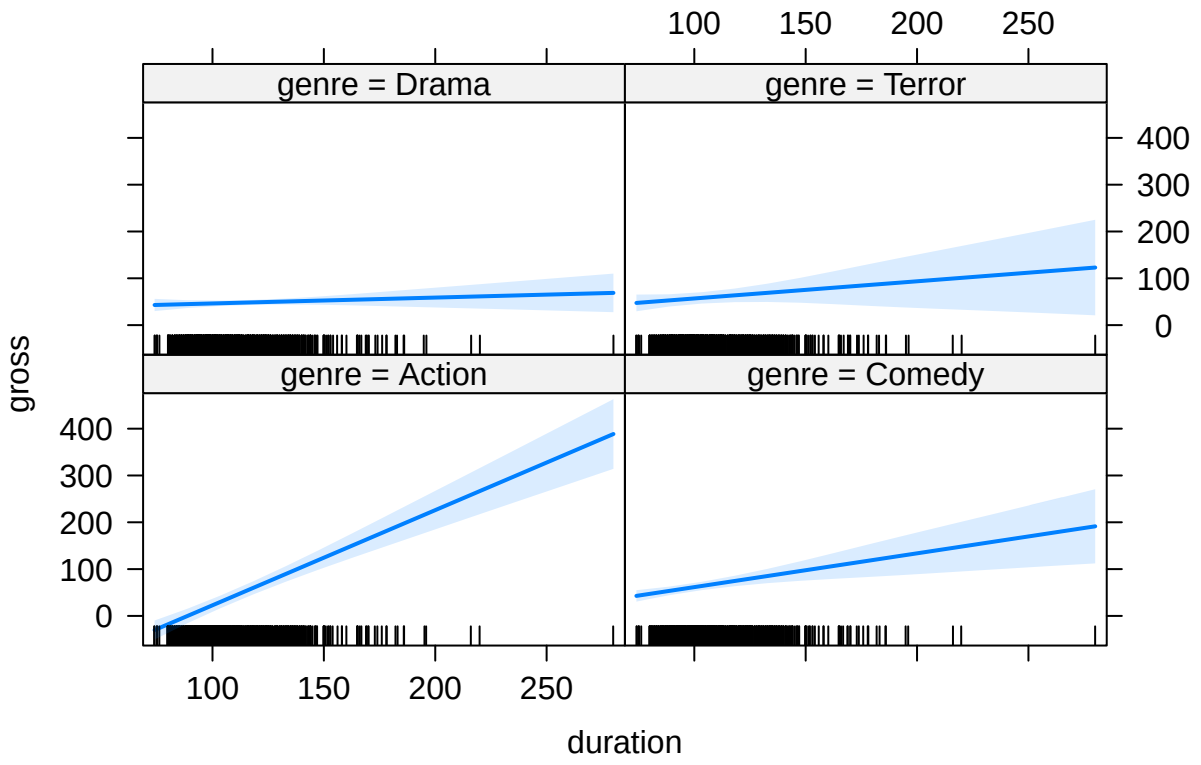
In first place, we look at the effect of the budget variable.

```
m1.effects <- effects::allEffects(m1)
plot(m1.effects, "budget:yearcat")
```



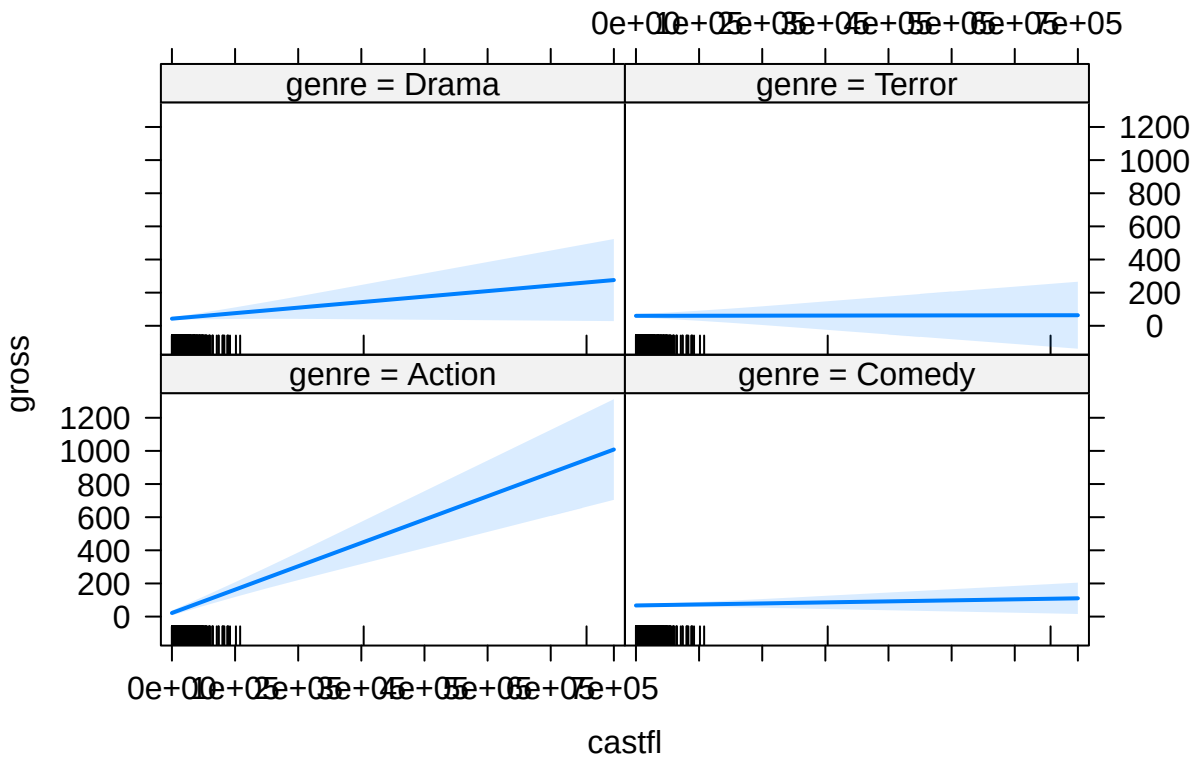
We clearly see a similar linear increase across all categories of the *yearcat* variable, being almost identical for *2000-2005* and *2006-2010* and slightly flatter for *2011-2016*. This means that the more budget a movie has, the more probable it is it has a bigger revenue. Specifically, it means that for every additional million dollars of budget we have an increase of 0.9809901, 0.9608053 and 0.6008523 million dollars in gross revenue, respectively for each category.

```
plot(m1.effects, "duration:genre")
```



Overall, the duration of Drama and Terror movies cause a lesser increase to the gross revenue than Action and Comedy. Specifically, for every increase in 1 minute the gross revenue increases 0.1255314 and 0.3669757 million dollars for Drama and Terror respectively. On the other hand, for every increase in 1 minute in Action and Comedy movies gross revenue increases 2.0325769 and 0.7215926 million dollars respectively.

```
plot(m1.effects, "castfl:genre")
```



Contrary to the effect of *duration*, we now sauses a higher increase to the gross revenue compared to Terror and Comedy. Specifically, for every thousand cast likes the gross revenue increases 1.4093738 and 0.3329645 million dollars for Action and Drama respectively. On the other hand, for every thousand cast likes in Comedy and Terror movies gross revenue increases 0.0616537 and 0.0052251 million dollars respectively.