STA302 Final Project

Preparation

Import Data

```
install.packages("MASS")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("carData")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
data <- read.csv ("movie_metadata.csv")
nrow(data)
```

```
## [1] 5043
```

split data in half

```
set.seed(42)
s <- sample(1:nrow(data), 2521 , replace=F)
train_0 <- data[s ,]
test <- data[-s,]

summary(train_0)
```

```
##     color           director_name      num_critic_for_reviews    duration
##  Length:2521        Length:2521        Min.   :  1.0          Min.   :  7.0
##  Class :character   Class :character   1st Qu.: 52.0          1st Qu.: 93.0
##  Mode  :character   Mode  :character   Median :114.0          Median :103.0
##                                        Mean   :146.8          Mean   :107.5
##                                        3rd Qu.:202.0          3rd Qu.:118.0
##                                        Max.   :813.0          Max.   :511.0
##                                        NA's   :26             NA's   :9
##  director_facebook_likes actor_3_facebook_likes actor_2_name
##  Min.   :    0.0         Min.   :    0.0        Length:2521
```

```
##  1st Qu.:    7.0         1st Qu.:  142.0        Class :character
##  Median :   50.0         Median :  373.5        Mode  :character
##  Mean   :  756.4         Mean   :  679.2
##  3rd Qu.:  189.0         3rd Qu.:  642.0
##  Max.   :23000.0         Max.   :23000.0
##  NA's   :60              NA's   :13
##  actor_1_facebook_likes     gross              genres
##  Min.   :     0.0       Min.   :      162   Length:2521
##  1st Qu.:   635.8       1st Qu.:  5677242   Class :character
##  Median :  1000.0       Median : 26288320   Mode  :character
##  Mean   :  6776.9       Mean   : 50131759
##  3rd Qu.: 11000.0       3rd Qu.: 62598270
##  Max.   :640000.0       Max.   :760505847
##  NA's   :5              NA's   :434
##  actor_1_name       movie_title        num_voted_users
##  Length:2521        Length:2521        Min.   :      6
##  Class :character   Class :character   1st Qu.:   9785
##  Mode  :character   Mode  :character   Median :  36587
##                                        Mean   :  89875
##                                        3rd Qu.: 105478
##                                        Max.   :1689764
##
##  cast_total_facebook_likes actor_3_name       facenumber_in_poster
##  Min.   :     0            Length:2521        Min.   : 0.000
##  1st Qu.:  1484            Class :character   1st Qu.: 0.000
##  Median :  3197            Mode  :character   Median : 1.000
##  Mean   : 10011                               Mean   : 1.387
##  3rd Qu.: 14007                               3rd Qu.: 2.000
##  Max.   :656730                               Max.   :31.000
##                                               NA's   :6
##  plot_keywords      movie_imdb_link    num_user_for_reviews   language
##  Length:2521        Length:2521        Min.   :   1.00      Length:2521
##  Class :character   Class :character   1st Qu.:  69.75      Class :character
##  Mode  :character   Mode  :character   Median : 159.00      Mode  :character
##                                        Mean   : 282.78
##                                        3rd Qu.: 335.00
##                                        Max.   :4667.00
##                                        NA's   :9
##    country          content_rating         budget            title_year
##  Length:2521        Length:2521        Min.   :2.180e+02   Min.   :1927
##  Class :character   Class :character   1st Qu.:6.000e+06   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :2.000e+07   Median :2006
##                                        Mean   :4.130e+07   Mean   :2003
##                                        3rd Qu.:4.600e+07   3rd Qu.:2011
##                                        Max.   :1.222e+10   Max.   :2016
```

```
##                                       NA's   :258         NA's   :62
##  actor_2_facebook_likes   imdb_score    aspect_ratio    movie_facebook_likes
##  Min.   :    0.0        Min.   :1.90   Min.   : 1.200   Min.   :     0
##  1st Qu.:  309.0        1st Qu.:5.90   1st Qu.: 1.850   1st Qu.:     0
##  Median :  619.0        Median :6.60   Median : 2.350   Median :   183
##  Mean   : 1675.1        Mean   :6.47   Mean   : 2.262   Mean   :  8501
##  3rd Qu.:  926.5        3rd Qu.:7.20   3rd Qu.: 2.350   3rd Qu.:  8000
##  Max.   :27000.0        Max.   :9.50   Max.   :16.000   Max.   :349000
##  NA's   :10                            NA's   :146
```

```
summary(test)
```

```
##     color           director_name      num_critic_for_reviews    duration
##  Length:2522        Length:2522        Min.   :  1.0          Min.   :  7.0
##  Class :character   Class :character   1st Qu.: 48.0          1st Qu.: 93.0
##  Mode  :character   Mode  :character   Median :104.0          Median :103.0
##                                        Mean   :133.6          Mean   :106.9
##                                        3rd Qu.:186.0          3rd Qu.:117.0
##                                        Max.   :739.0          Max.   :325.0
##                                        NA's   :24             NA's   :6
##  director_facebook_likes actor_3_facebook_likes actor_2_name
##  Min.   :    0.0         Min.   :    0.0        Length:2522
##  1st Qu.:    6.0         1st Qu.:  124.0        Class :character
##  Median :   47.0         Median :  369.5        Mode  :character
##  Mean   :  617.1         Mean   :  610.8
##  3rd Qu.:  208.0         3rd Qu.:  625.0
##  Max.   :22000.0         Max.   :20000.0
##  NA's   :44              NA's   :10
##  actor_1_facebook_likes     gross              genres
##  Min.   :     0         Min.   :      703   Length:2522
##  1st Qu.:   592         1st Qu.:  4928640   Class :character
##  Median :   969         Median : 24770850   Mode  :character
##  Mean   :  6344         Mean   : 46793015
##  3rd Qu.: 11000         3rd Qu.: 61173742
##  Max.   :260000         Max.   :474544677
##  NA's   :2              NA's   :450
##  actor_1_name       movie_title        num_voted_users
##  Length:2522        Length:2522        Min.   :      5
##  Class :character   Class :character   1st Qu.:   7804
##  Mode  :character   Mode  :character   Median :  31790
##                                        Mean   :  77464
##                                        3rd Qu.:  89438
##                                        Max.   :1238746
##
##  cast_total_facebook_likes actor_3_name       facenumber_in_poster
```

```
##  Min.   :     0           Length:2522        Min.   : 0.000
##  1st Qu.:  1341           Class :character   1st Qu.: 0.000
##  Median :  2958           Mode  :character   Median : 1.000
##  Mean   :  9388                              Mean   : 1.355
##  3rd Qu.: 13472                              3rd Qu.: 2.000
##  Max.   :303717                              Max.   :43.000
##                                              NA's   :7
##  plot_keywords      movie_imdb_link    num_user_for_reviews   language
##  Length:2522        Length:2522        Min.   :   1.0       Length:2522
##  Class :character   Class :character   1st Qu.:  60.0       Class :character
##  Mode  :character   Mode  :character   Median : 153.0       Mode  :character
##                                        Mean   : 262.8
##                                        3rd Qu.: 321.0
##                                        Max.   :5060.0
##                                        NA's   :12
##    country          content_rating        budget            title_year
##  Length:2522        Length:2522        Min.   :1.400e+03   Min.   :1916
##  Class :character   Class :character   1st Qu.:6.000e+06   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :1.850e+07   Median :2005
##                                        Mean   :3.822e+07   Mean   :2002
##                                        3rd Qu.:4.000e+07   3rd Qu.:2010
##                                        Max.   :4.200e+09   Max.   :2016
##                                        NA's   :234         NA's   :46
##  actor_2_facebook_likes   imdb_score     aspect_ratio    movie_facebook_likes
##  Min.   :     0         Min.   :1.600   Min.   : 1.180   Min.   :     0.0
##  1st Qu.:   258         1st Qu.:5.800   1st Qu.: 1.850   1st Qu.:     0.0
##  Median :   579         Median :6.500   Median : 2.350   Median :   147.5
##  Mean   :  1628         Mean   :6.415   Mean   : 2.178   Mean   :  6551.2
##  3rd Qu.:   905         3rd Qu.:7.200   3rd Qu.: 2.350   3rd Qu.:  2000.0
##  Max.   :137000         Max.   :9.200   Max.   :16.000   Max.   :197000.0
##  NA's   :3                              NA's   :183
```

data cleaning

```
table(train_0$content_rating)
```

```
##
##            Approved         G        GP         M     NC-17 Not Rated    Passed
##       148        28        55         2         3         3        58         7
##        PG     PG-13         R     TV-14      TV-G     TV-MA     TV-PG      TV-Y
##       341       732      1056        17         8        11         6         1
##   Unrated         X
##        38         7
```

```r
summary(train_0$content_rating)
```

```
##    Length     Class      Mode
##      2521 character character
```

```r
dim(train_0)
```

```
## [1] 2521   28
```

```r
train_1 <- subset(as.data.frame(train_0),
              select = c(num_critic_for_reviews,imdb_score, actor_1_facebook_likes, cast


train_1$content_rating <- ifelse(train_1$content_rating %in% c("R", "PG", "PG-13"),
                                 train_1$content_rating,
                                 "other")

table(train_1$content_rating)
```

```
##
## other    PG PG-13     R
##   392   341   732  1056
```

```r
dim(train_1)
```

```
## [1] 2521   11
```

```r
train_1 <- train_1[complete.cases(train_1),]
summary(train_1)
```

```
##  num_critic_for_reviews   imdb_score    actor_1_facebook_likes
##  Min.   :  1.0          Min.   :1.900   Min.   :     0
##  1st Qu.: 77.0          1st Qu.:5.900   1st Qu.:   756
##  Median :139.0          Median :6.600   Median :  1000
##  Mean   :170.6          Mean   :6.493   Mean   :  7918
##  3rd Qu.:226.0          3rd Qu.:7.200   3rd Qu.: 13000
##  Max.   :813.0          Max.   :9.300   Max.   :640000
##  cast_total_facebook_likes facenumber_in_poster    duration
##  Min.   :     0            Min.   : 0.000       Min.   : 34.0
##  1st Qu.:  1951            1st Qu.: 0.000       1st Qu.: 95.0
##  Median :  4279            Median : 1.000       Median :106.0
##  Mean   : 11734            Mean   : 1.378       Mean   :110.4
##  3rd Qu.: 16435            3rd Qu.: 2.000       3rd Qu.:120.0
##  Max.   :656730            Max.   :31.000       Max.   :330.0
##  director_facebook_likes movie_facebook_likes     gross
##  Min.   :    0.0         Min.   :     0       Min.   :      162
##  1st Qu.:   10.0         1st Qu.:     0       1st Qu.:  7097467
```

```
##  Median :   57.0         Median :   279       Median : 28711194
##  Mean   :  871.0         Mean   : 10226       Mean   : 52774965
##  3rd Qu.:  216.8         3rd Qu.: 12000       3rd Qu.: 65488078
##  Max.   :23000.0         Max.   :349000       Max.   :760505847
##      budget          content_rating
##  Min.   :2.180e+02   Length:1942
##  1st Qu.:1.000e+07   Class :character
##  Median :2.500e+07   Mode  :character
##  Mean   :4.669e+07
##  3rd Qu.:5.018e+07
##  Max.   :1.222e+10
```

```
dim(train_0)
```

```
## [1] 2521   28
```

```
train_1 <- subset(as.data.frame(train_0),
              select = c(num_critic_for_reviews,imdb_score, actor_1_facebook_likes, cast


train_1$content_rating <- ifelse(train_1$content_rating %in% c("R", "PG", "PG-13"),
                                 train_1$content_rating,
                                 "other")

table(train_1$content_rating)
```

```
##
## other    PG PG-13     R
##   392   341   732  1056
```

```
dim(train_1)
```

```
## [1] 2521   11
```

```
train_1 <- train_1[complete.cases(train_1),]
summary(train_1)
```

```
##  num_critic_for_reviews   imdb_score    actor_1_facebook_likes
##  Min.   :  1.0          Min.   :1.900   Min.   :     0
##  1st Qu.: 77.0          1st Qu.:5.900   1st Qu.:   756
##  Median :139.0          Median :6.600   Median :  1000
##  Mean   :170.6          Mean   :6.493   Mean   :  7918
##  3rd Qu.:226.0          3rd Qu.:7.200   3rd Qu.: 13000
##  Max.   :813.0          Max.   :9.300   Max.   :640000
##  cast_total_facebook_likes facenumber_in_poster    duration
##  Min.   :     0            Min.   : 0.000       Min.   : 34.0
##  1st Qu.:  1951            1st Qu.: 0.000       1st Qu.: 95.0
##  Median :  4279            Median : 1.000       Median :106.0
```

```
##  Mean   : 11734            Mean   : 1.378       Mean   :110.4
##  3rd Qu.: 16435            3rd Qu.: 2.000       3rd Qu.:120.0
##  Max.   :656730            Max.   :31.000       Max.   :330.0
##  director_facebook_likes movie_facebook_likes     gross
##  Min.   :    0.0         Min.   :     0       Min.   :      162
##  1st Qu.:   10.0         1st Qu.:     0       1st Qu.:  7097467
##  Median :   57.0         Median :   279       Median : 28711194
##  Mean   :  871.0         Mean   : 10226       Mean   : 52774965
##  3rd Qu.:  216.8         3rd Qu.: 12000       3rd Qu.: 65488078
##  Max.   :23000.0         Max.   :349000       Max.   :760505847
##      budget          content_rating
##  Min.   :2.180e+02   Length:1942
##  1st Qu.:1.000e+07   Class :character
##  Median :2.500e+07   Mode  :character
##  Mean   :4.669e+07
##  3rd Qu.:5.018e+07
##  Max.   :1.222e+10
```

conducting regression

```
model_1 <- lm(gross ~ num_critic_for_reviews+imdb_score+ actor_1_facebook_likes+ cast_to

summary(model_1)
```

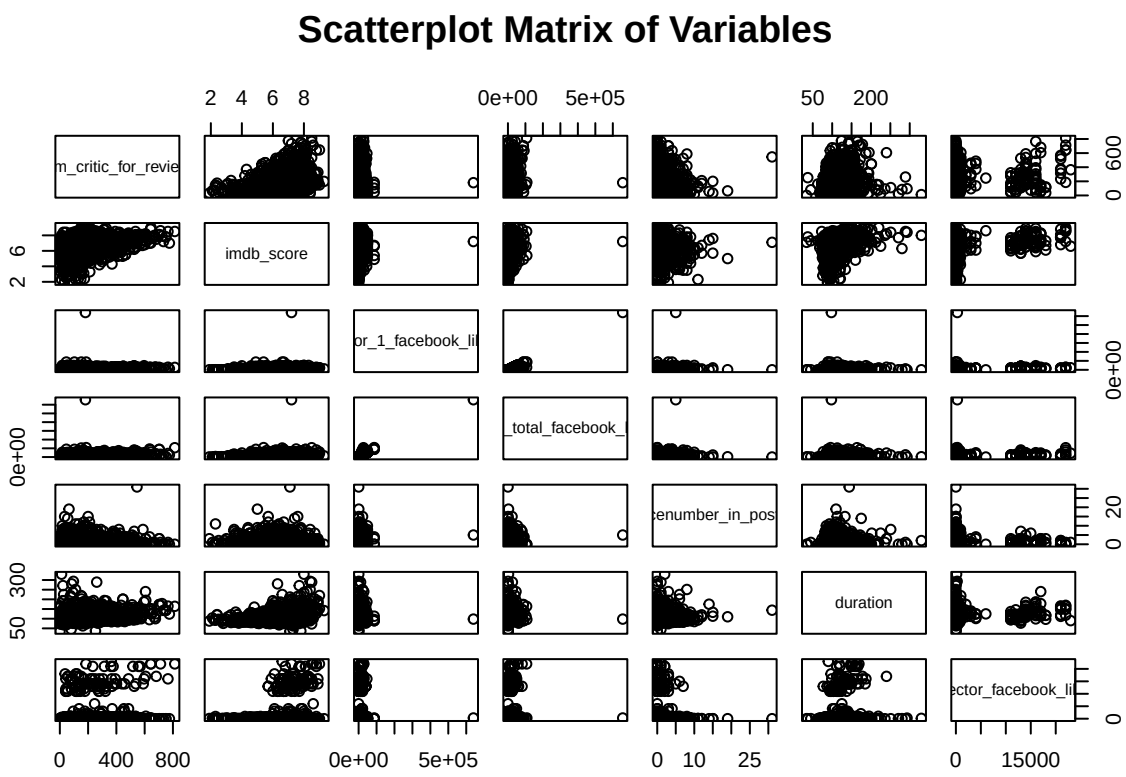
```
##
## Call:
## lm(formula = gross ~ num_critic_for_reviews + imdb_score + actor_1_facebook_likes +
##     cast_total_facebook_likes + duration + director_facebook_likes +
##     movie_facebook_likes + budget + content_rating, data = train_1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -171494438  -32146963   -9210627   19256993  544908568
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -6.443e+07  1.130e+07  -5.701 1.37e-08 ***
## num_critic_for_reviews     2.120e+05  1.549e+04  13.691  < 2e-16 ***
## imdb_score                 5.413e+06  1.510e+06   3.585 0.000346 ***
## actor_1_facebook_likes    -2.353e+03  2.599e+02  -9.055  < 2e-16 ***
## cast_total_facebook_likes  2.294e+03  2.267e+02  10.119  < 2e-16 ***
## duration                   3.875e+05  6.359e+04   6.094 1.32e-09 ***
## director_facebook_likes    4.606e+02  4.318e+02   1.067 0.286305
## movie_facebook_likes      -1.027e+02  8.316e+01  -1.235 0.216967
```

```
## budget                       6.495e-03  4.807e-03   1.351 0.176828
## content_ratingPG             2.848e+07  6.434e+06   4.425 1.02e-05 ***
## content_ratingPG-13          8.111e+06  5.998e+06   1.352 0.176387
## content_ratingR             -2.594e+07  5.734e+06  -4.523 6.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59150000 on 1930 degrees of freedom
## Multiple R-squared:  0.3653, Adjusted R-squared:  0.3617
## F-statistic:   101 on 11 and 1930 DF,  p-value: < 2.2e-16
```

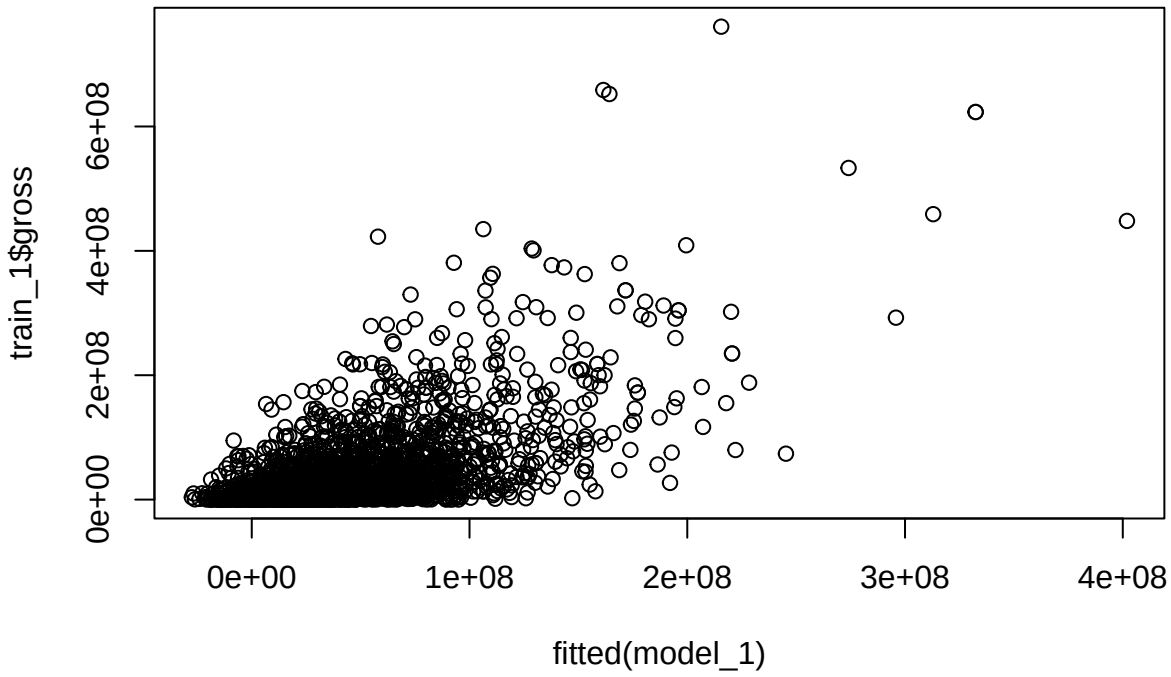
Model checking

Check conditions

```
# conditional mean predictors
pairs(train_1[, 1:7],
      main="Scatterplot Matrix of Variables")
```




```
# conditional mean response
plot(train_1$gross ~ fitted(model_1))
```




```
# examine num_critic_for_reviews and imdb_score
 number_imdb <-lm(num_critic_for_reviews ~imdb_score,data=train_1 )
 summary(number_imdb)
```

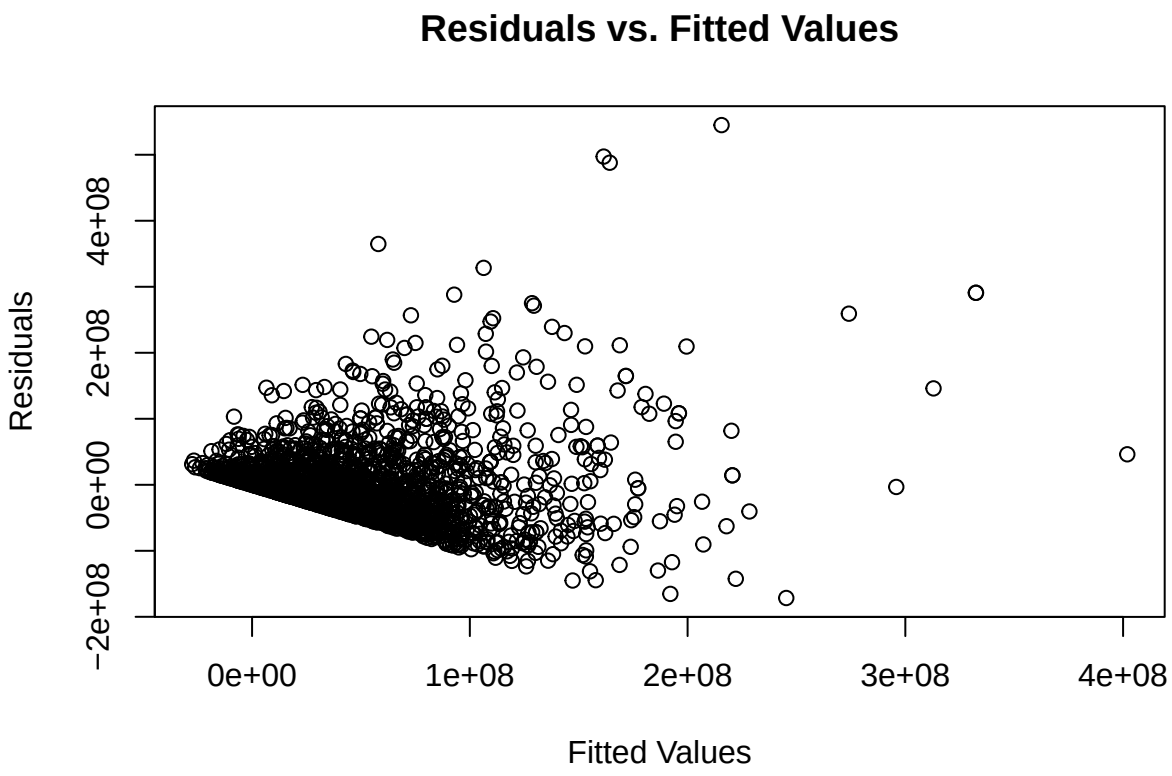
```
##
## Call:
## lm(formula = num_critic_for_reviews ~ imdb_score, data = train_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -227.64  -87.39  -25.70   58.14  582.30
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -111.988     17.491  -6.403 1.91e-10 ***
## imdb_score    43.527      2.659  16.366  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 122.9 on 1940 degrees of freedom
## Multiple R-squared:  0.1213, Adjusted R-squared:  0.1209
## F-statistic: 267.9 on 1 and 1940 DF,  p-value: < 2.2e-16
```

```
 correlation_coefficient <- cor(train_1$num_critic_for_reviews, train_1$imdb_score, meth
 print(correlation_coefficient)
```

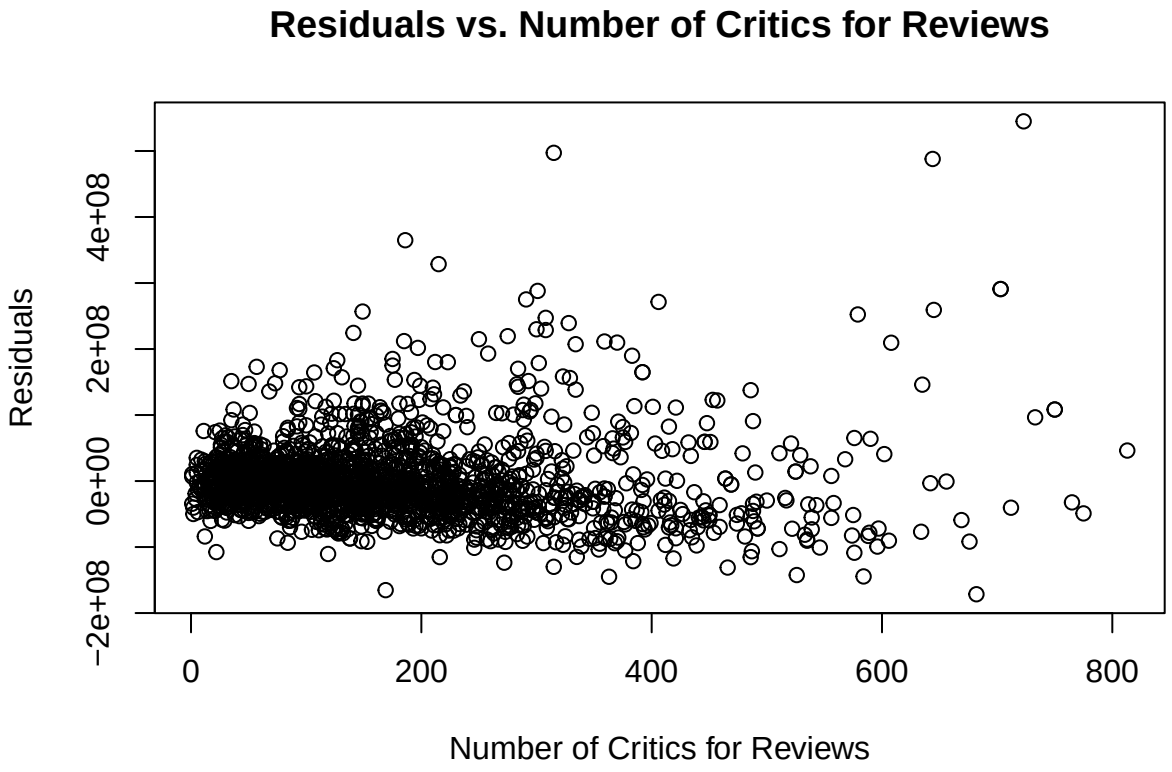
```
## [1] 0.3483127
```

Check assumptions

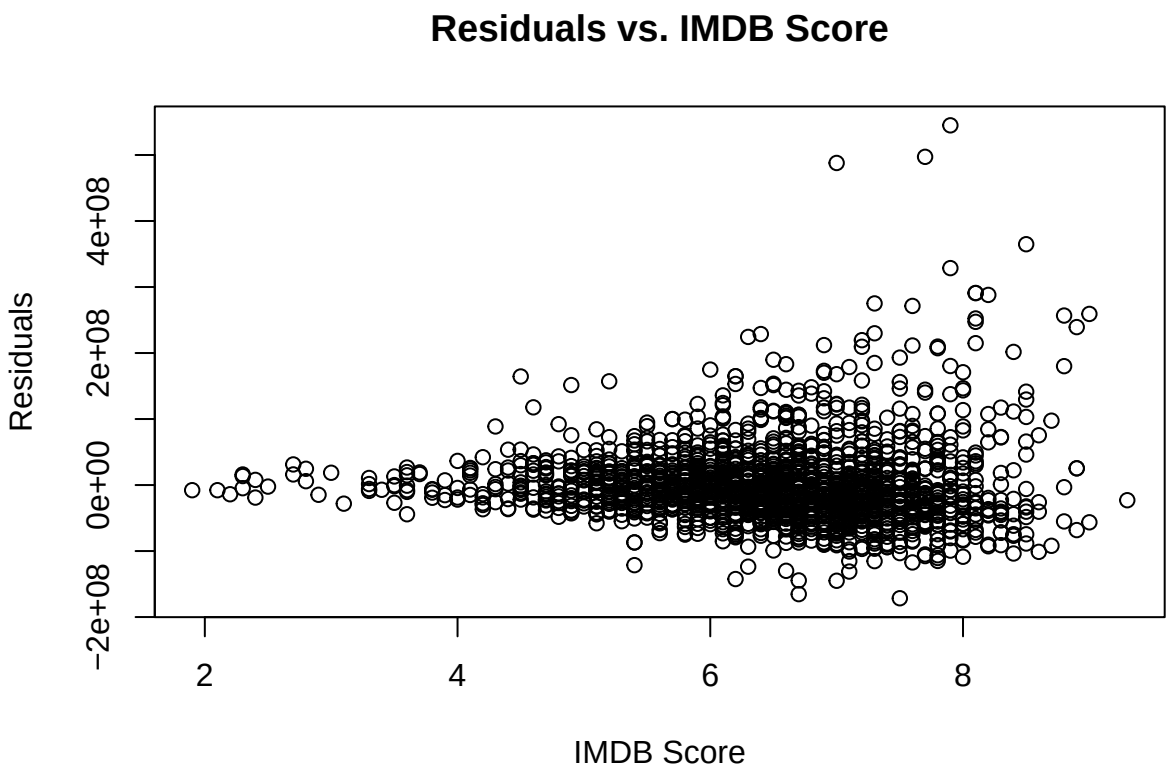
```
# residual vs. fitted
plot(model_1$residuals ~ model_1$fitted.values,
     xlab="Fitted Values",
     ylab="Residuals",
     main="Residuals vs. Fitted Values")
```



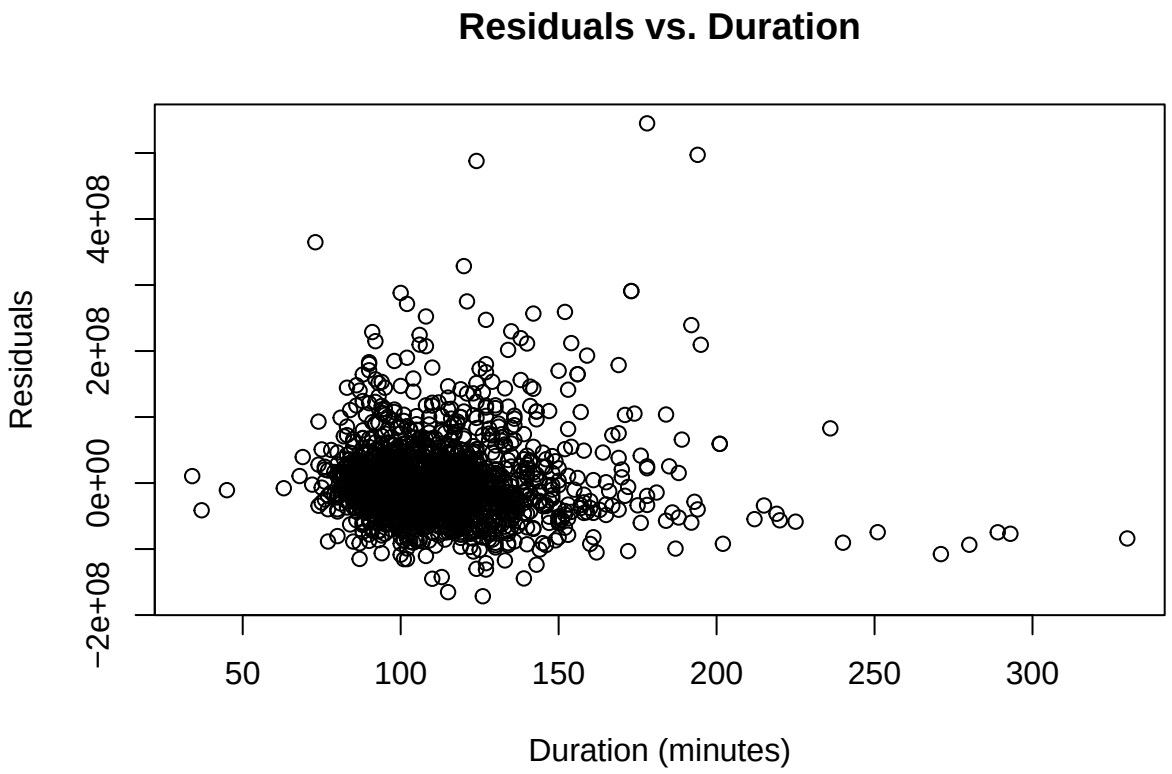

```
# residual vs predictors

# residual vs. num_critic_for_reviews
plot(model_1$residuals ~ num_critic_for_reviews, data=train_1,
     xlab="Number of Critics for Reviews",
     ylab="Residuals",
     main="Residuals vs. Number of Critics for Reviews")
```



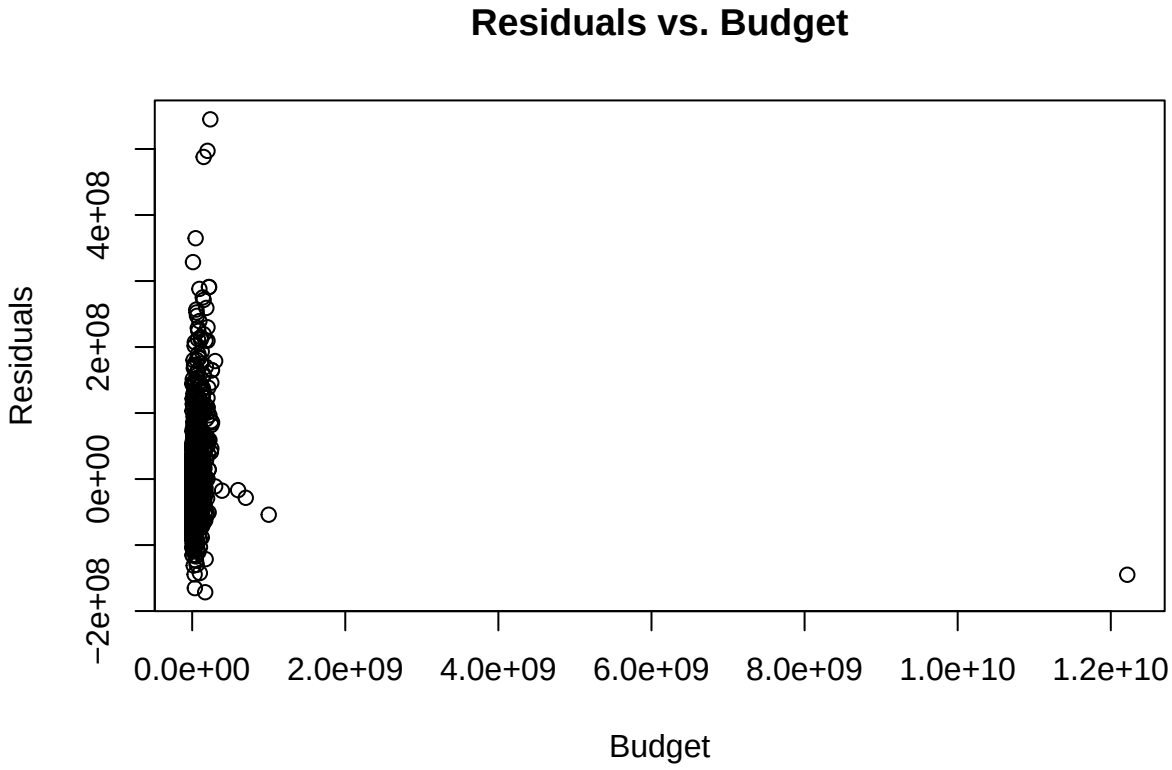

```
# residual vs. imdb_score
plot(model_1$residuals ~ imdb_score, data=train_1,
     xlab="IMDB Score",
     ylab="Residuals",
     main="Residuals vs. IMDB Score")
```



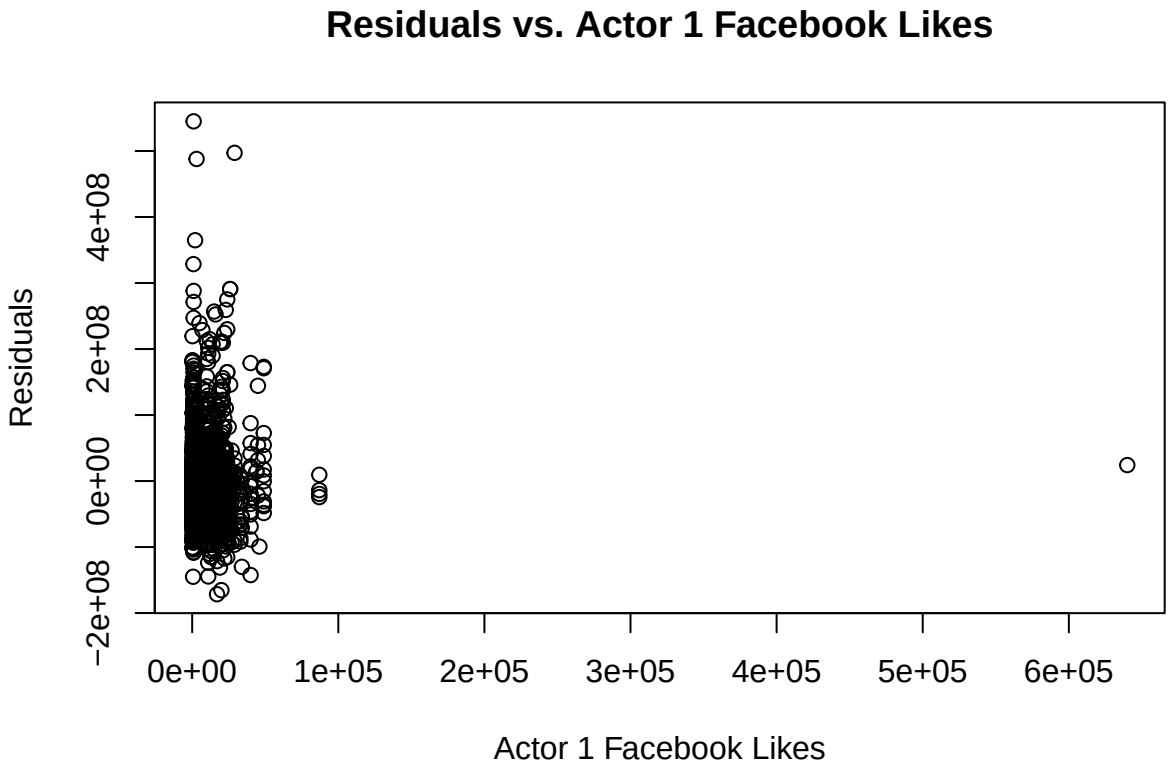

```
# residual vs. duration
plot(model_1$residuals ~ duration, data=train_1,
     xlab="Duration (minutes)",
     ylab="Residuals",
     main="Residuals vs. Duration")
```



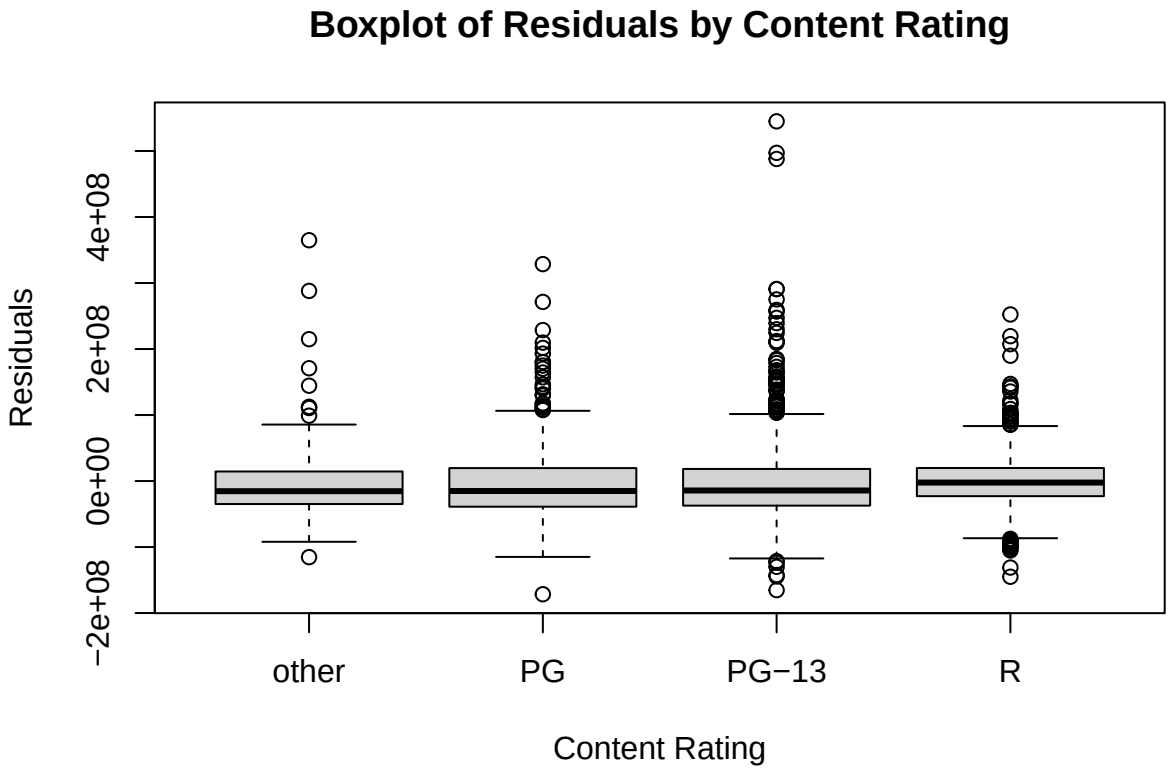

```r
# residual vs. budget
plot(model_1$residuals ~ budget, data=train_1,
     xlab="Budget",
     ylab="Residuals",
     main="Residuals vs. Budget")
```



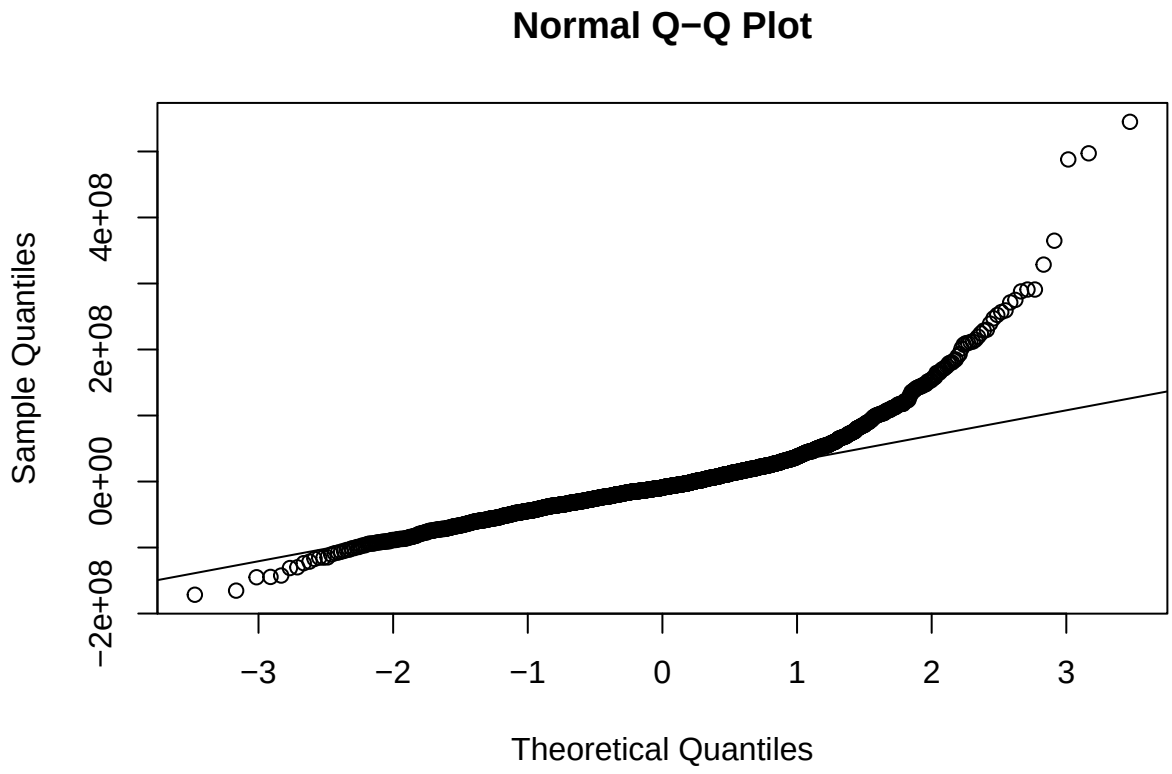

```
# residual vs. actor_1_facebook_likes
plot(model_1$residuals ~ actor_1_facebook_likes, data=train_1,
     xlab="Actor 1 Facebook Likes",
     ylab="Residuals",
     main="Residuals vs. Actor 1 Facebook Likes")
```



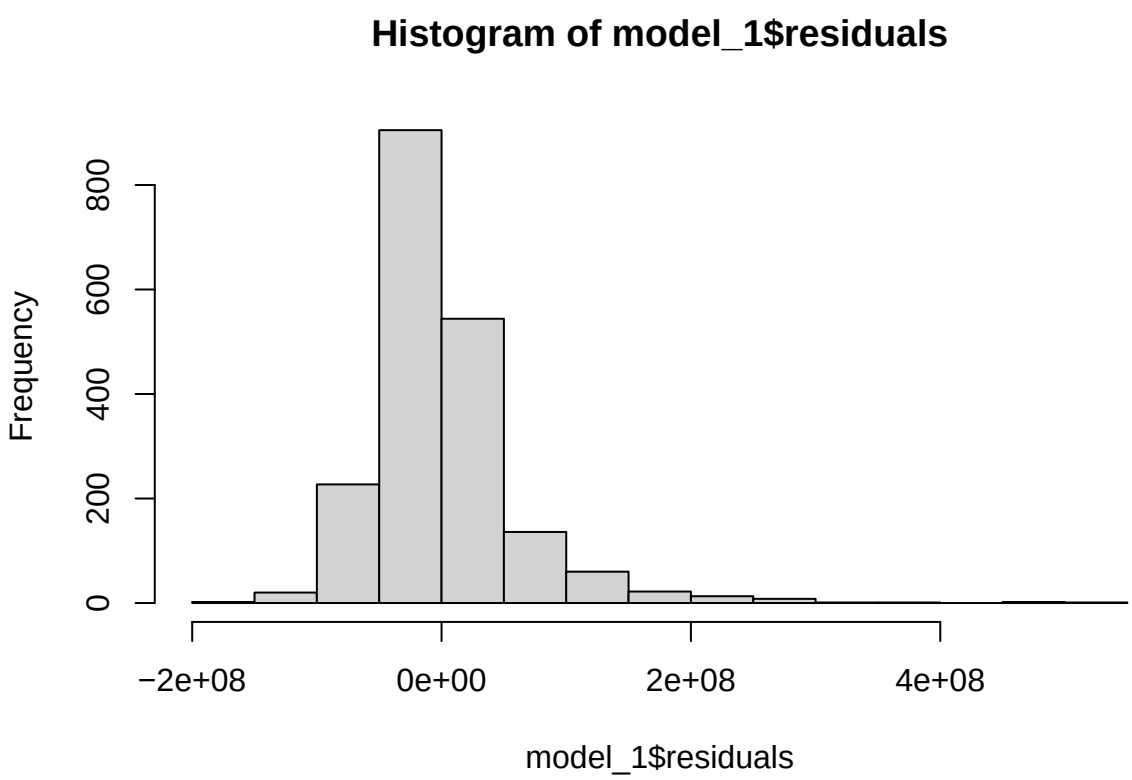

```
# residual vs. content_rating
boxplot(residuals(model_1) ~ train_1$content_rating,
       xlab="Content Rating",
       ylab="Residuals",
       main="Boxplot of Residuals by Content Rating")
```




```
# qq plot
qqnorm(model_1$residuals)
qqline(model_1$residuals)
```



```r
hist(model_1$residuals)
```



due to the failure to satisfy the assumption, we drop the variable facenumber_in_poster.

```r
#new data set
train_2 <- subset(as.data.frame(train_1),
                  select = c(num_critic_for_reviews,imdb_score, actor_1_facebook_likes, cast
summary(train_2)
```

```
##  num_critic_for_reviews   imdb_score    actor_1_facebook_likes
##  Min.   :  1.0          Min.   :1.900   Min.   :     0
##  1st Qu.: 77.0          1st Qu.:5.900   1st Qu.:   756
##  Median :139.0          Median :6.600   Median :  1000
##  Mean   :170.6          Mean   :6.493   Mean   :  7918
##  3rd Qu.:226.0          3rd Qu.:7.200   3rd Qu.: 13000
##  Max.   :813.0          Max.   :9.300   Max.   :640000
##  cast_total_facebook_likes    duration         gross
##  Min.   :     0            Min.   : 34.0   Min.   :      162
##  1st Qu.:  1951            1st Qu.: 95.0   1st Qu.:  7097467
##  Median :  4279            Median :106.0   Median : 28711194
##  Mean   : 11734            Mean   :110.4   Mean   : 52774965
##  3rd Qu.: 16435            3rd Qu.:120.0   3rd Qu.: 65488078
##  Max.   :656730            Max.   :330.0   Max.   :760505847
##      budget          content_rating
##  Min.   :2.180e+02   Length:1942
```

```
##  1st Qu.:1.000e+07   Class :character
##  Median :2.500e+07   Mode  :character
##  Mean   :4.669e+07
##  3rd Qu.:5.018e+07
##  Max.   :1.222e+10
```

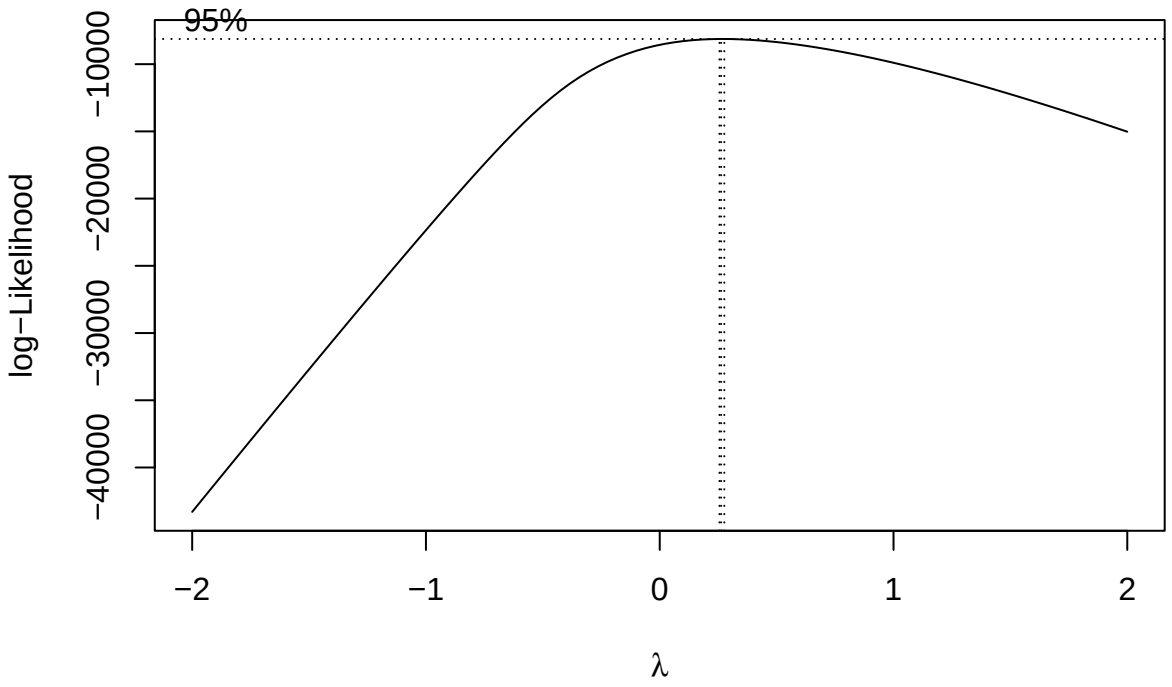
```
model_2 <- lm(gross ~ num_critic_for_reviews+imdb_score+actor_1_facebook_likes+cast_tota
summary(model_2)
```

```
##
## Call:
## lm(formula = gross ~ num_critic_for_reviews + imdb_score + actor_1_facebook_likes +
##     cast_total_facebook_likes + duration + budget + content_rating,
##     data = train_2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -163723097  -32379907   -9048698   19183776  547925869
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -6.438e+07  1.102e+07  -5.842 6.05e-09 ***
## num_critic_for_reviews      2.011e+05  1.188e+04  16.927  < 2e-16 ***
## imdb_score                  5.496e+06  1.499e+06   3.665 0.000254 ***
## actor_1_facebook_likes     -2.330e+03  2.581e+02  -9.030  < 2e-16 ***
## cast_total_facebook_likes   2.276e+03  2.248e+02  10.128  < 2e-16 ***
## duration                    3.894e+05  6.311e+04   6.171 8.24e-10 ***
## budget                      6.722e-03  4.803e-03   1.400 0.161789
## content_ratingPG            2.894e+07  6.426e+06   4.504 7.05e-06 ***
## content_ratingPG-13         8.536e+06  5.992e+06   1.424 0.154477
## content_ratingR            -2.547e+07  5.727e+06  -4.447 9.20e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59160000 on 1932 degrees of freedom
## Multiple R-squared:  0.3645, Adjusted R-squared:  0.3616
## F-statistic: 123.1 on 9 and 1932 DF,  p-value: < 2.2e-16
```

Box-cox transformation

```
train_3 <- subset(as.data.frame(train_2),
                  select = c(num_critic_for_reviews,imdb_score,duration,gross, budget))
```

```
library(MASS)
result<-boxcox(model_2)
```




```
detach(package:MASS, unload=TRUE)

library(car)
```

```
## Loading required package: carData
```

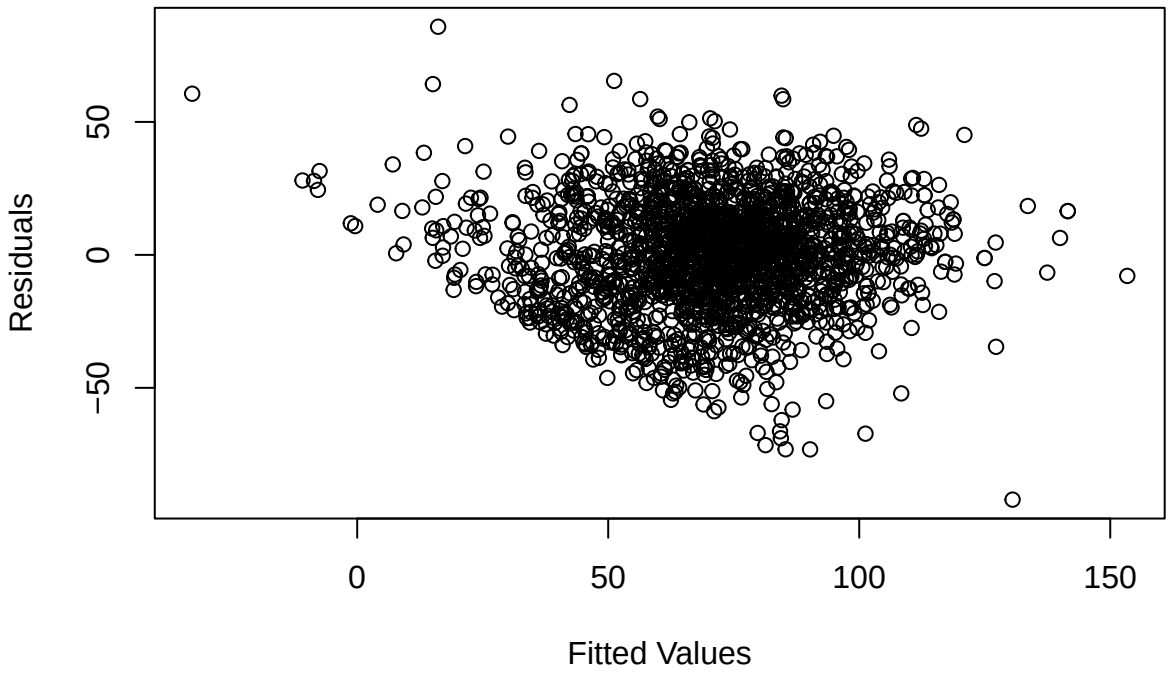
```
boxcoxtable<- powerTransform(cbind(train_3[,1:5]))
summary(boxcoxtable)
```

```
## bcPower Transformations to Multinormality
##                          Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## num_critic_for_reviews    0.3012        0.33       0.2619       0.3405
## imdb_score                2.2991        2.30       2.1034       2.4948
## duration                 -0.4442       -0.50      -0.5527      -0.3357
## gross                     0.2563        0.26       0.2382       0.2744
## budget                    0.1677        0.17       0.1516       0.1838
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                     LRT df       pval
## LR test, lambda = (0 0 0 0 0) 2305.321  5 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                     LRT df       pval
```

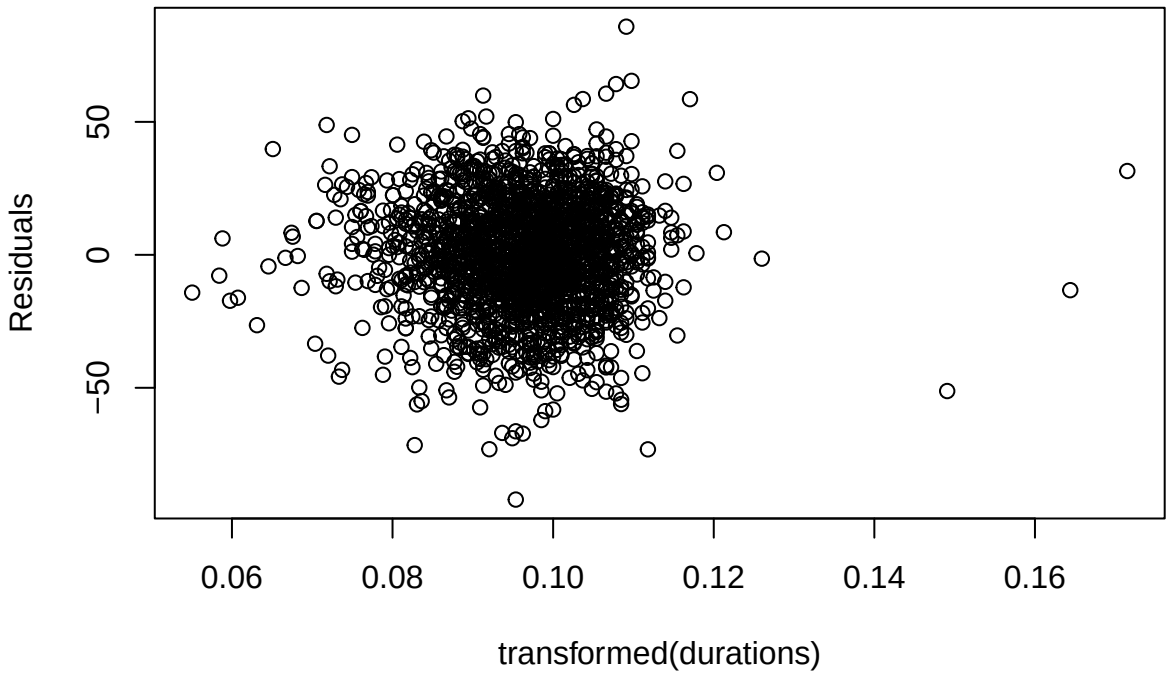
```
## LR test, lambda = (1 1 1 1 1) 15800.27  5 < 2.22e-16
```

Repeat Assumption Checking

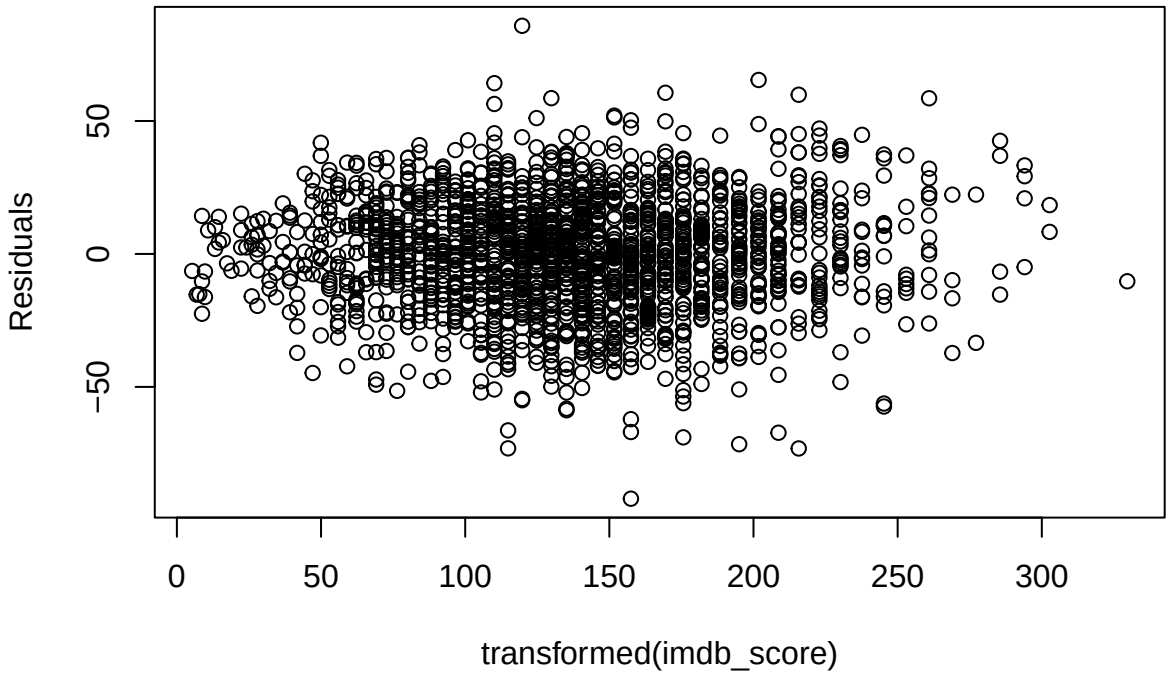
```
t_gross <- train_3$gross ** (1/4)
t_imdb_score <- train_3$imdb_score**(2.6)
t_duration <- train_3$duration**(-1/2)
t_numcritic <- train_3$num_critic_for_reviews**(1/3)
model_3 <- lm(t_gross ~ log(budget) + t_numcritic + t_duration + t_imdb_score +actor_1_f

e_hat3 <- resid(model_3)
y_hat3 <- fitted(model_3)

plot(e_hat3 ~ y_hat3, xlab ="Fitted Values", ylab="Residuals")
```



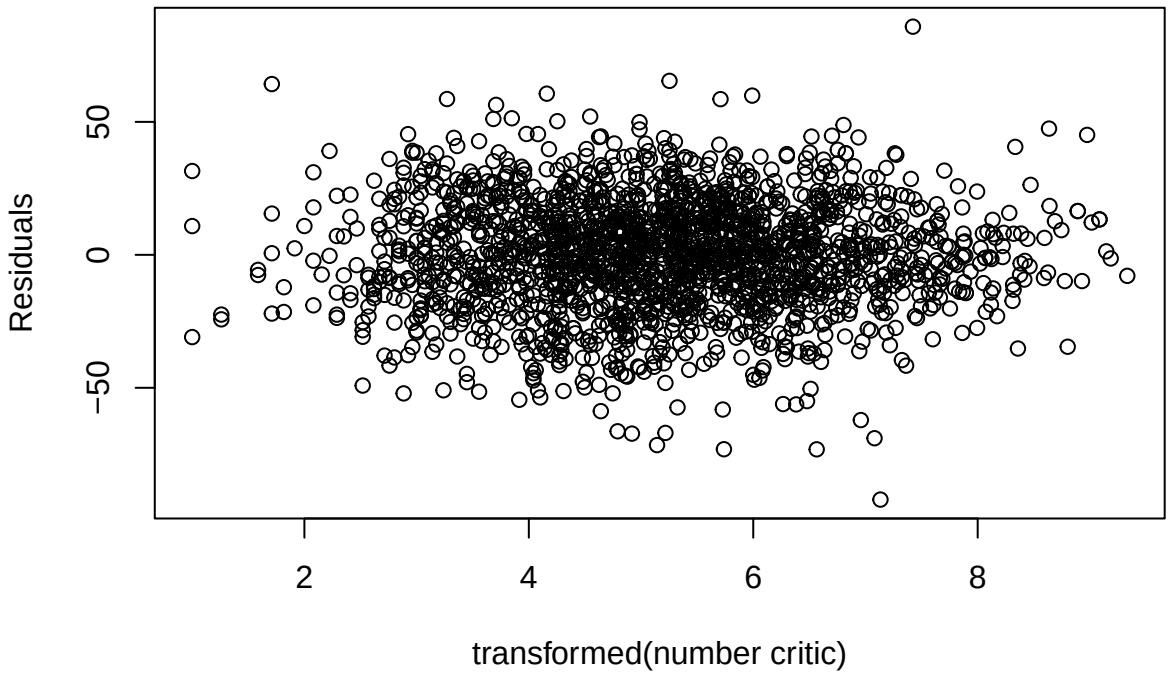

```
plot(e_hat3 ~ t_duration,xlab ="transformed(durations)", ylab = "Residuals")
```



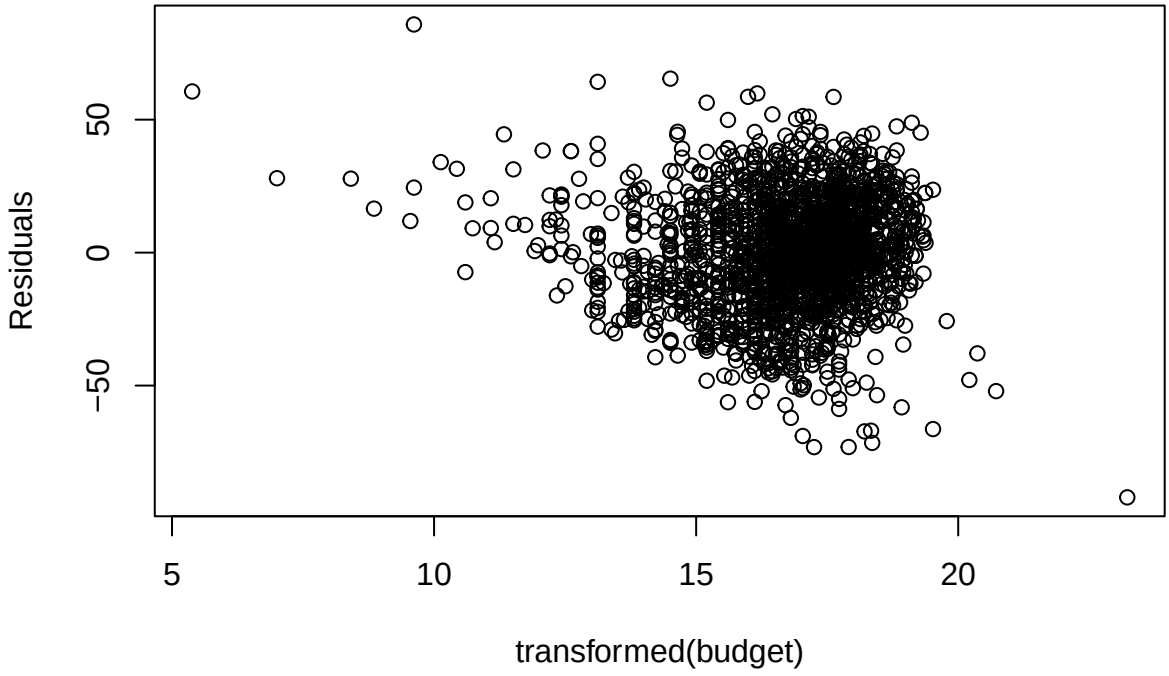

```
plot(e_hat3 ~ t_imdb_score,xlab ="transformed(imdb_score)", ylab = "Residuals")
```



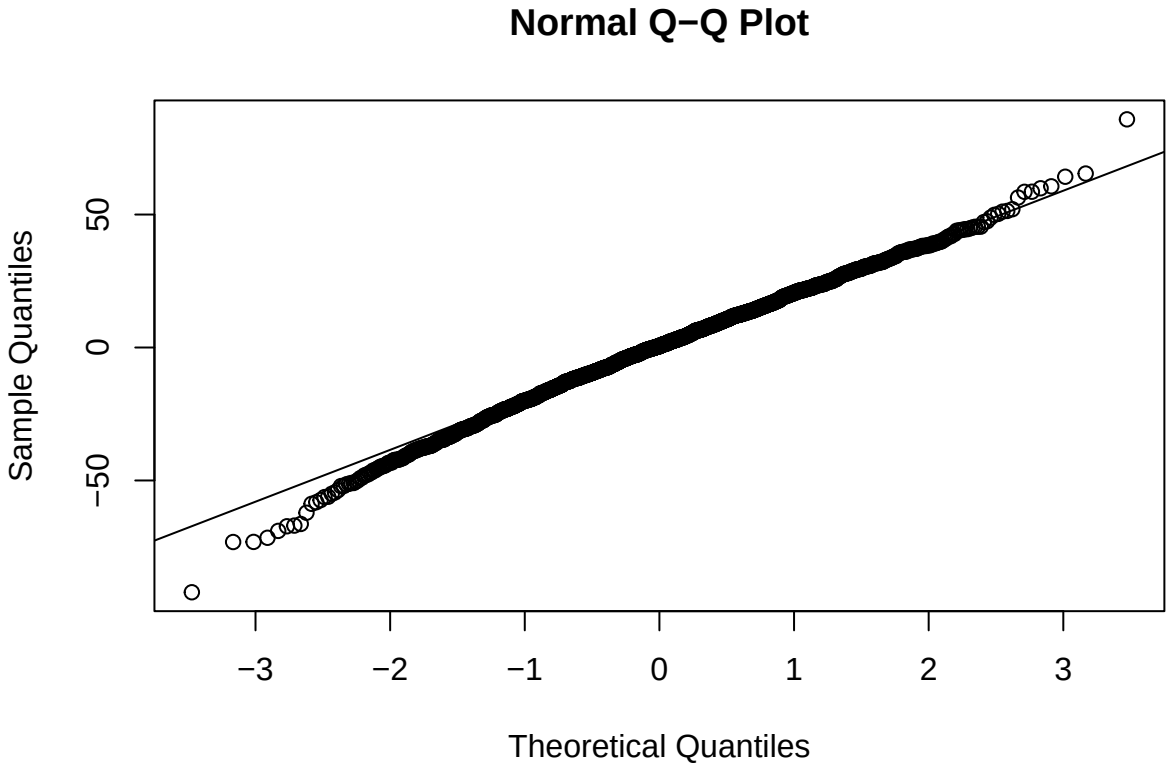

```
plot(e_hat3 ~ t_numcritic,xlab ="transformed(number critic)", ylab = "Residuals")
```




```
plot(e_hat3 ~ log(train_3$budget),xlab ="transformed(budget)", ylab = "Residuals")
```




```
qqnorm(e_hat3);qqline(e_hat3)
```




```
summary(model_3)
```

```
##
## Call:
## lm(formula = t_gross ~ log(budget) + t_numcritic + t_duration +
##     t_imdb_score + actor_1_facebook_likes + cast_total_facebook_likes +
##     content_rating, data = train_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -92.030 -12.647   0.621  13.646  85.802
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -1.110e+02  1.082e+01 -10.262  < 2e-16 ***
## log(budget)                 8.343e+00  3.817e-01  21.855  < 2e-16 ***
## t_numcritic                 5.373e+00  4.166e-01  12.898  < 2e-16 ***
## t_duration                  3.486e+00  6.372e+01   0.055 0.956379
## t_imdb_score                6.204e-02  1.137e-02   5.458 5.43e-08 ***
## actor_1_facebook_likes     -3.641e-04  8.949e-05  -4.068 4.93e-05 ***
## cast_total_facebook_likes   3.603e-04  7.792e-05   4.624 4.02e-06 ***
## content_ratingPG            1.411e+01  2.303e+00   6.126 1.09e-09 ***
## content_ratingPG-13         7.870e+00  2.160e+00   3.643 0.000277 ***
## content_ratingR            -8.051e-01  2.041e+00  -0.394 0.693286
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.71 on 1932 degrees of freedom
## Multiple R-squared:  0.4962, Adjusted R-squared:  0.4939
## F-statistic: 211.4 on 9 and 1932 DF,  p-value: < 2.2e-16
```

```
library(car)
vif(model_3)
```

```
##                               GVIF Df GVIF^(1/(2*Df))
## log(budget)                1.581145  1        1.257436
## t_numcritic                1.535617  1        1.239200
## t_duration                 1.465433  1        1.210551
## t_imdb_score               1.551513  1        1.245597
## actor_1_facebook_likes    11.316400  1        3.363986
## cast_total_facebook_likes 11.785148  1        3.432950
## content_rating             1.310009  3        1.046034
```

Given that the VIF value of actor_1_facebook_likes and cast_total_facebook_likes are 9.702867 and 10.061221, we remove cast_total_facebook_likes from the model.

checking collinearity

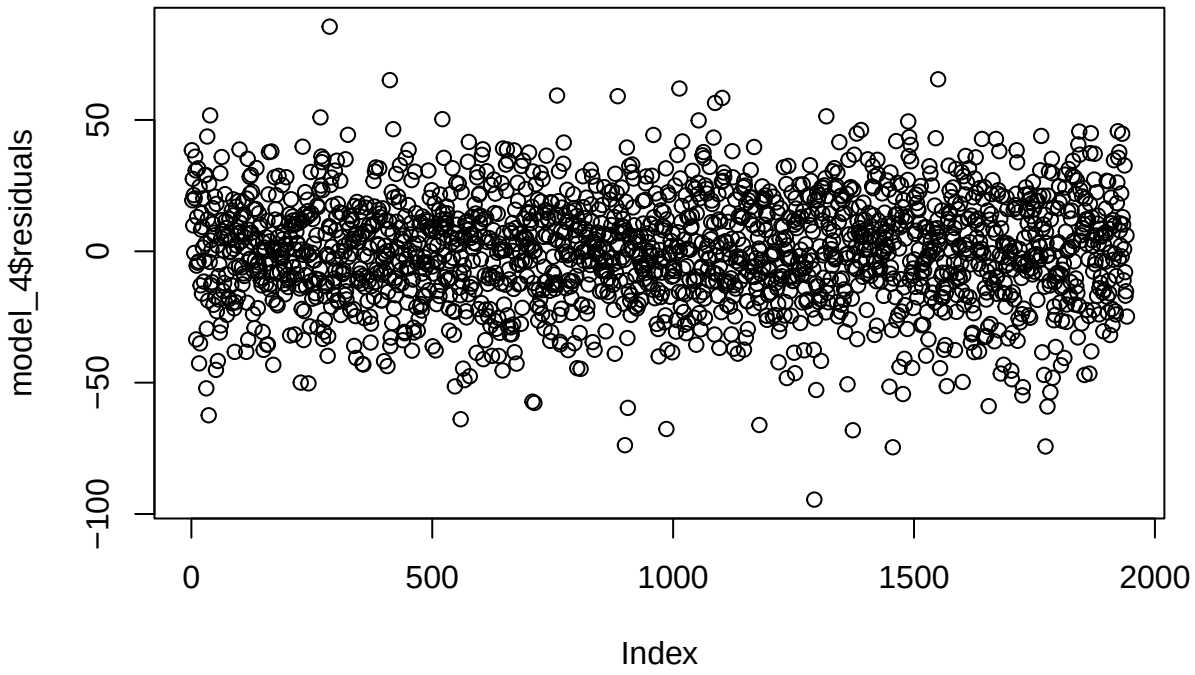
```
model_4 <- lm(t_gross ~ log(budget) + t_numcritic + t_duration + t_imdb_score +actor_1_f
vif(model_4)
```

```
##                              GVIF Df GVIF^(1/(2*Df))
## log(budget)              1.569416  1        1.252763
## t_numcritic              1.491103  1        1.221107
## t_duration               1.456007  1        1.206651
## t_imdb_score             1.551487  1        1.245587
## actor_1_facebook_likes 1.041238  1        1.020411
## content_rating           1.306570  3        1.045576
```

check problematic observations

Outlier

```
plot(model_4$residuals)
```




```
#find outlier
r_i <- rstandard(model_4)
print("outliers (large)")
```

```
## [1] "outliers (large)"
```

```
which(rstandard(model_4) > 4 | rstandard(model_4) < -4)
```

```
## 4794 2989
##  287 1293
```

```
data_outlier <- train_2[-c(2335, 2989, 3860, 2158, 2698, 3304), ]
model_4 <- lm(t_gross ~ log(budget) + t_numcritic + t_duration + t_imdb_score +actor_1_f
```

Leverage

```
leverages <- hatvalues(model_4)
n <- nrow(train_3)
p <- length(coef(model_4))
threshold <- (2 * p) / n
high_leverage_points <- which(leverages > threshold)
print(high_leverage_points)
```

```
## 1903  481  299 4226 1396 4816 4680 4736 4158 5027 5043 2237  715 4749 4813 4933
##   11   20   21   33   34   56   61   95  119  121  124  162  189  200  241  255
## 3409 1671 4794 4201 4199 3279   39  474 4748 2914  757 4450 4334 4900 4517 2333
##  275  285  287  319  320  325  348  352  372  390  411  412  439  456  470  478
## 2519  324 1981 3016 5024  309 1330 1096 4436 3586 4435 4284 4737 4287 5030 5012
```

```
##  485  495  515  535  550  560  579  594  604  605  624  660  664  669  687  689
##  298 4798 5034 4865 2735 1502  239 3609 4859  510 3973 1599 3108 4602 1290 4103
##  694  704  707  711  712  721  737  741  746  759  765  772  776  790  845  854
##   92 4504 3076 4800 1715 2372 2697 2488 4939 3724  884 4999 4312 1285 3281 4963
##  906  934  998 1013 1024 1033 1049 1052 1071 1106 1153 1157 1162 1175 1185 1198
## 4720 4934 2971 2267 4108 3277 4988 2989 1421   75 4956 2783 3567 3319  339 1748
## 1204 1211 1258 1263 1264 1281 1285 1293 1310 1311 1335 1343 1357 1362 1381 1390
## 1958 4724 1572 4790 3220 4156  407 3029 2814 4573 1751 3080 4779 2738 4726 2838
## 1393 1404 1432 1449 1454 1458 1460 1463 1477 1483 1490 1522 1523 1525 1550 1571
## 2255 3511  556 4594  965 4978 4660 4004 1643 4777 4759   59 2357 1528 4499  505
## 1584 1595 1611 1617 1629 1643 1658 1664 1666 1668 1688 1711 1713 1725 1744 1746
## 1471 3274 4019  903 4547 3708 4787 2589 3594 3021  840  385 4796
## 1761 1788 1791 1799 1802 1821 1852 1864 1865 1868 1875 1879 1905
```

```
length(high_leverage_points)
```

```
## [1] 141
```

Influential point

```
# Cook's D (on all fitted values)
di <- cooks.distance(model_4)
cutoff_di <- qf(0.5, 7, 3873)
which(di > cutoff_di)
```

```
## named integer(0)
```

```
# DFFITS (on individual fitted value)
dffits <- dffits(model_4)
cutoff_dffits <- 2*sqrt(p/n)
which(abs(dffits)>cutoff_dffits)
```

```
## 3954 1903 2015 3317 4226  517 2336 4680 1537 5027 5043  715 3820 2151   27 2918
##   10   11   16   31   33   36   55   61   63  121  124  189  227  231  268  270
## 3409 4794 4199 3279 3685 1720 4748 2914 4450   30 4334 4517 3016 2034 5024 4008
##  275  287  320  325  338  357  372  390  412  419  439  470  535  547  550  559
## 3251 1096 3669 4436 3586 4435 4924 3585 3090 2998 5012 3219 2735  239 1296  510
##  564  594  603  604  605  624  647  655  670  671  689  708  712  737  745  759
## 3973 1599 4929 1317  340 2374 3081  328   92  927 2741 3076 4800  632 2488 1868
##  765  772  773  808  829  879  885  900  906  959  986  998 1013 1019 1052 1053
## 2052 3324 4822 4329 4188 1990 3281 1319 4108 1491 2989 2150 4956 3567 3319  383
## 1064 1102 1123 1134 1168 1179 1185 1236 1264 1292 1293 1297 1335 1357 1362 1373
##  339 1748 4724 4790 3220  801 3029 2814 1751  837 2738    1 4726 2356  935 2046
## 1381 1390 1404 1449 1454 1456 1463 1477 1490 1493 1525 1545 1550 1554 1568 1601
## 2426 4004 2493 2120 2357 1528  701 2817 1172 3422 4543 3274 3708 2589 2154 3021
## 1655 1664 1677 1680 1713 1725 1726 1770 1773 1777 1783 1788 1821 1864 1867 1868
```

```
##  840
## 1875
```

```
# DFBETAS (on individual coefficient)
dfbetas <- dfbetas(model_4)
dim(dfbetas)
```

```
## [1] 1942    9
```

```
cutoff_dfbetas <- 2/sqrt(3880)
which(abs(dfbetas[,1]) > cutoff_dfbetas) ##beta0
```

```
##  634  356 4261 3954  860  299 3317  626 4226  517   33 2157 2344  149 4680 2812
##    5    8    9   10   17   21   31   32   33   36   37   39   51   52   61   89
##  162 3578 5027 5043 2767 2085  715  928 3820 1282 4813 1930   27 2918 3866 3409
##  100  114  121  124  161  181  189  197  227  234  241  243  268  270  273  275
## 4794 4523 1185 4199 3279 3685 4656 1720 1634 4748 2914 2588  454 4334 3555  813
##  287  290  309  320  325  338  341  357  362  372  390  407  429  439  445  451
## 2739 2584  324  153 3931 1981 3016 2769 3570 2034 5024  118 3251  696 2051 1399
##  458  462  495  499  501  515  535  542  545  547  550  561  564  574  576  592
## 1096 3669 3586 2975 1747 4435 4681 3523 4924  353 3534 2890 5030 5012  298 3471
##  594  603  605  606  611  624  630  640  647  653  663  675  687  689  694  701
## 4798 5034 3219 2735 1502 3951  239 1296 4859  510 3973 4929  209 3939 4103 4857
##  704  707  708  712  721  733  737  745  746  759  765  773  793  833  854  868
## 2054 3081 1071   92 2263 2177  927 1507  122 2741  112 2736  441 4800  632 1715
##  880  885  904  906  914  944  959  970  975  986 1004 1006 1009 1013 1019 1024
## 2488 2919 4310 1398 3324  845 2621 4822 4230  126  884 1009 4312 4188 3281  144
## 1052 1063 1066 1084 1102 1103 1105 1123 1146 1148 1153 1155 1162 1168 1185 1195
## 1700 1314 4108 3277 4061 4988 4685 1491 2989 2150 1815 4956  523 3567 3319  383
## 1207 1212 1264 1281 1284 1285 1289 1292 1293 1297 1307 1335 1345 1357 1362 1373
##  339 2464 1748 4724 1088 3275  564  525 4790 3220 3029  316 4573 3119  837 3245
## 1381 1382 1390 1404 1415 1418 1424 1434 1449 1454 1463 1481 1483 1487 1493 1515
## 2738 4399 3874 4726 2356 3201 3400  935 3092 3511 3671 2046 1129 3229 3822 4978
## 1525 1527 1530 1550 1554 1555 1565 1568 1587 1595 1599 1601 1604 1618 1626 1643
## 2426 4004 2493 2120 1047 4759 4383   59 4058 2357 1528  701   80 2817 1172 4543
## 1655 1664 1677 1680 1686 1688 1695 1711 1712 1713 1725 1726 1740 1770 1773 1783
## 2161 3274 1723 4547 4729 3363 3708 4818   67 3667 2589 2154 3021  840  345 4304
## 1786 1788 1794 1802 1809 1814 1821 1833 1838 1854 1864 1867 1868 1875 1903 1907
##  698
## 1926
```

```
which(abs(dfbetas[,2]) > cutoff_dfbetas) ##beta1
```

```
## 1098  356 4261 1903 2015  860  299  982 4226  517 2157 4680 4856 2812 4567 4736
##    3    8    9   11   16   17   21   25   33   36   39   61   70   89   92   95
##  162 4892 1611 5027 5043 3698  715  928 3783 4060 3820 4784 2151 1282 4813 1083
##  100  103  117  121  124  166  189  197  205  207  227  229  231  234  241  248
```

```
##  592 2918 3866 4794 4523   95 4367 3279 3773 4656  474 4414 4748 2914 2588 4852
##  269  270  273  287  290  304  310  325  330  341  352  363  372  390  407  430
## 4334 3555  292 4517 2333  324  153 3931 3582 3016 3570 2034 5024 4008 3251 1399
##  439  445  464  470  478  495  499  501  521  535  545  547  550  559  564  592
## 1096 4854 3669 2975 4429 4435 4681 3523 4924  353 5030 5012  298 3471 4798 5034
##  594  597  603  606  621  624  630  640  647  653  687  689  694  701  704  707
## 3219 2735 4596 3951  239 4859  510 3973 4929 4453 4758 3939 3887 4857 3697 2054
##  708  712  724  733  737  746  759  765  773  806  810  833  837  868  873  880
## 1908 3081  719  328 1071   92 2263  927 3347 1212 4105 3104 2741 2421 3076  441
##  883  885  893  900  904  906  914  959  969  971  974  985  986  989  998 1009
## 4800  760  649 2331  127 4310  167 1398  845 2621 4822  529 4230 1009 4999 4312
## 1013 1026 1040 1048 1061 1066 1069 1084 1103 1105 1123 1124 1146 1155 1157 1162
## 2243 4357  144 4963 4524 4898 4108 4988 4685 1491 2989 4394 4533 3703  851 4057
## 1165 1190 1195 1198 1243 1250 1264 1285 1289 1292 1293 1295 1299 1313 1314 1331
## 4956 3567 4539 3319  383 1159  339 2464 1748 4588 4724 1088 3275 3899 4790 4535
## 1335 1357 1360 1362 1373 1376 1381 1382 1390 1402 1404 1415 1418 1425 1449 1451
##  407 3029 2814 4476  316 1751  837 4897 4487 3874  191 4726 3400  935 2222 3671
## 1460 1463 1477 1478 1481 1490 1493 1505 1514 1530 1534 1550 1565 1568 1572 1599
## 4181 4438 3531 3822  711 4978 2426 4660 4004 4537 2493 4383 1721   59 1528  701
## 1607 1608 1625 1626 1641 1643 1655 1658 1664 1675 1677 1695 1703 1711 1725 1726
##  215 1869 2817 1172 3422  854 4543 3274  903 1618 4547 4729 1188 4818 3885 1017
## 1731 1764 1770 1773 1777 1780 1783 1788 1799 1801 1802 1809 1829 1833 1843 1848
## 4948 2589  289  840  385 4796 1403
## 1853 1864 1874 1875 1879 1905 1932
```

```
which(abs(dfbetas[,3]) > cutoff_dfbetas) ##beta2
```

```
## 2609 1098  356 4261 1903  932  860 3944  982 3032 3317  626  517   33 2157 2336
##    1    3    8    9   11   14   17   18   25   28   31   32   36   37   39   55
## 4680 4121 2310 1611 4297 5027 5043 2317 1709 3211 3698 3286 3783 4060 3820 4784
##   61   74   78  117  118  121  124  131  150  157  166  173  205  207  227  229
## 2151 3417 1414 1930 4283 1083 1143 2918 2541 3530 4794 4199 3279 1365 1634 4414
##  231  233  236  243  246  248  266  270  274  277  287  320  325  359  362  363
## 3732 2914 2588 4450 2732   30 4852 3435 4334   71 3555   63  813 2584  292 2333
##  383  390  407  412  413  419  430  433  439  443  445  448  451  462  464  478
## 2138 2059 1981 3016 2769 2034  848 4008 3640 1399 1040 4436 4435 2843 4681 3693
##  505  509  515  535  542  547  555  559  571  592  593  604  624  628  630  633
## 3769 3315 3523 1081 2455  795 3534  312 3090 2998 2890 3471 5034 3219 2735 2943
##  637  638  640  644  657  658  663  668  670  671  675  701  707  708  712  715
## 1502  233 1080 3951  510 3989 3757 3481 3064 3592 3939 3887 3697 1334 2054 1908
##  721  723  729  733  759  763  780  795  811  813  833  837  873  877  880  883
## 3081  328 1071 1744 2263  238 2358  927 3347 1507 1212 4105 3932 2811 3104 2741
##  885  900  904  912  914  918  956  959  969  970  971  974  982  984  985  986
## 3005 2421 3076  441 4800 3101 3084  632 1715 3706 2331 1868 3468 1363 3953 3324
##  988  989  998 1009 1013 1015 1016 1019 1024 1041 1048 1053 1087 1096 1100 1102
```

```
## 2621 4822  529 1345 4230  126 4188 1990 4205 4357 3237 2126 3636 2921 2532 3587
## 1105 1123 1124 1126 1146 1148 1168 1179 1181 1190 1194 1219 1222 1230 1235 1239
## 4524 1099 4206 4108 3548 4061 3462 1491 2989 2150 4533 1815 3703  851 2764 4057
## 1243 1246 1251 1264 1272 1284 1286 1292 1293 1297 1299 1307 1313 1314 1318 1331
## 2848 3567 4539 3319 2608  383 1159 2464 1748 3275  564 3899 4790 4535 3220 3029
## 1350 1357 1360 1362 1367 1373 1376 1382 1390 1418 1424 1425 1449 1451 1454 1463
## 1569 2814 4476 3069 2082 1590  837 2496 2902  565 4109 2164 3245 2738  191   89
## 1467 1477 1478 1480 1488 1489 1493 1494 1496 1499 1508 1511 1515 1525 1534 1538
## 2017    1 4726 3201 2009  935 2222 3137 2457 3092 3785 2046 4181 4438 1546  689
## 1543 1545 1550 1555 1562 1568 1572 1574 1585 1587 1588 1601 1607 1608 1614 1624
## 4978 2852 2837 2426 3093 2493 2120 4383 1721 2357 1528  701 1900 1869 2006  854
## 1643 1646 1651 1655 1670 1677 1680 1695 1703 1713 1725 1726 1747 1764 1766 1780
## 4543 2927 3274 4019 4632 3363  195   18 3885 3261 1017  863 3667 2589 2154 3021
## 1783 1787 1788 1791 1812 1814 1817 1830 1843 1844 1848 1851 1854 1864 1867 1868
##  232    7 2909 2731 1074    9 2616 1403
## 1892 1900 1901 1911 1915 1922 1923 1932
```

```
which(abs(dfbetas[,4]) > cutoff_dfbetas) ##beta3
```

```
##  634  356 4261 3954  299 3317  626 4226  517   33 2157 1877 4680 2812 3578 1611
##    5    8    9   10   21   31   32   33   36   37   39   59   61   89  114  117
## 4297 5027 5043    2 1126 2767 2085  715  928 3037 3783 3820 1930 4283   27 2918
##  118  121  124  126  144  161  181  189  197  204  205  227  243  246  268  270
## 3409 4794 1185 4199 2533 1720 1365 1634 4748 3985 2914 1722 2588 4450  454 4334
##  275  287  309  320  342  357  359  362  372  381  390  400  407  412  429  439
## 3780  813 2739 2584  324  153 1981   65 1400 2769 2034 5024 4008 3251  696 2051
##  442  451  458  462  495  499  515  516  524  542  547  550  559  564  574  576
## 1399 1096 3586 1747 3693  353 1133 2455  795  312 2890  203  298 3219 4566 2735
##  592  594  605  611  633  653  656  657  658  668  675  677  694  708  709  712
## 1502  233 1080  239 1296  510 3989 3973 3416 4929 1724  209  841  340 3356 1334
##  721  723  729  737  745  759  763  765  766  773  782  793  816  829  832  877
## 2054  577  328 1071   92 1744 2263  238 2177  927 1507 3932 2811 2741 3076  112
##  880  897  900  904  906  912  914  918  944  959  970  982  984  986  998 1004
## 2736 4800  632 1715 2488 2919 2399 1398 3324  845 2621 4203  884 1009 4188 1576
## 1006 1013 1019 1024 1052 1063 1076 1084 1102 1103 1105 1149 1153 1155 1168 1171
## 4205 3281 1700 1314 2971 3277 4061 3462 1491 2989 2150 1815 2857 1749 4956  523
## 1181 1185 1207 1212 1258 1281 1284 1286 1292 1293 1297 1307 1308 1316 1335 1345
## 2848 3319  339 1748 4724 1088 3275  564 1572  525 4790 3029  695 2814 4573 3119
## 1350 1362 1381 1390 1404 1415 1418 1424 1432 1434 1449 1463 1470 1477 1483 1487
## 1751 3245 3874  191    1 2356 3201 3400  935 3092 3511 3671 2046 1129 1546 3229
## 1490 1515 1530 1534 1545 1554 1555 1565 1568 1587 1595 1599 1601 1604 1614 1618
## 3822 3406 4004 1643  713 1047 3203 2357 1528 2817 1172 4543 3274 1723 4547 1727
## 1626 1649 1664 1666 1682 1686 1704 1713 1725 1770 1773 1783 1788 1794 1802 1805
## 3363 3708   18 1017 3667 4604 2589 2154 3021  840    7  345 4304 1069
## 1814 1821 1830 1848 1854 1863 1864 1867 1868 1875 1900 1903 1907 1918
```

```
which(abs(dfbetas[,5]) > cutoff_dfbetas) ##beta4
```

```
##  634  356 3954 3944 2454 3032 3317  626  517 2344 1877  427 4680 1537 4121 2310
##    5    8   10   18   22   28   31   32   36   51   59   60   61   63   74   78
## 2831  162 3578 4297 5027 5043 2867  928 3820 1414 1930 4283 3095 3866 2541 3409
##   90  100  114  118  121  124  179  197  227  236  243  246  271  273  274  275
## 3948 4794 3176 2022 4199 2314 3685 2029 1720 2722  900 2914 1722 4450 3435 4334
##  283  287  296  298  320  324  338  349  357  373  379  390  400  412  433  439
## 3780 3555   63  813 2193 2153 2341 3582 4123 3016 5024 3251 3554 1399 1040 3669
##  442  445  448  451  466  492  507  521  534  535  550  564  567  592  593  603
## 2975 4435 4681 3693 3315 3523  353 3585 2455 3090 2998 3471 4798 3219  350  233
##  606  624  630  633  638  640  653  655  657  670  671  701  704  708  719  723
## 3951  239 1296  510 3989 3973 1599 4929 1317  340 3356 3887 2650 4500  852 2374
##  733  737  745  759  763  765  772  773  808  829  832  837  846  860  869  879
## 2054 3081  328 1071 1744 2263 3526 2177 2358  927 1212 4105 3932 3005  112 2736
##  880  885  900  904  912  914  921  944  956  959  971  974  982  988 1004 1006
## 4800  632 1294  649 3706  284 1868  912 2052  933 1398 3324 2621  509 4329 4230
## 1013 1019 1027 1040 1041 1051 1053 1056 1064 1070 1084 1102 1105 1129 1134 1146
##  126  884 1009 2243 4188 4205 3341 2126 2532 1319 3587 1938 4108 3710 4061 3462
## 1148 1153 1155 1165 1168 1181 1201 1219 1235 1236 1239 1249 1264 1269 1284 1286
## 4685 1491 2989 3703  320 4057 1885 1335 4539 3319  383  339 2464 1748 4588 4724
## 1289 1292 1293 1313 1320 1331 1347 1348 1360 1362 1373 1381 1382 1390 1402 1404
## 2840 3275 3899  684 3918 4790 4535 3220  801  407 3029 3069 3119 2082  837 2902
## 1414 1418 1425 1437 1448 1449 1451 1454 1456 1460 1463 1480 1487 1488 1493 1496
## 3245 2738 3874 3217 4183  935 3092 2046 1129 4181 4438  439  965  506 2652 3406
## 1515 1525 1530 1544 1549 1568 1587 1601 1604 1607 1608 1610 1629 1635 1647 1649
## 2426  888 4004 1870 2120  713 1047 4383 3203   59 2357 2166 4115   80  655 1875
## 1655 1659 1664 1674 1680 1682 1686 1695 1704 1711 1713 1714 1728 1740 1742 1754
## 1869 2006 2817 1172 2927 4547  272 2489 3363 3672 1360 4818   67 1979 3885 2589
## 1764 1766 1770 1773 1787 1802 1807 1813 1814 1820 1823 1833 1838 1840 1843 1864
## 3594  598  289    7 4796 4304  838  698 2713
## 1865 1866 1874 1900 1905 1907 1924 1926 1942
```

```
which(abs(dfbetas[,6]) > cutoff_dfbetas) ##beta5
```

```
## 1903 2450 2336 4297    2 2237   27 2533 2029 2540   30 1400 4008 4436 2975 3534
##   11   53   55  118  126  162  268  342  349  389  419  524  559  604  606  663
## 3090 3081  328 4504  927 4105 2741 1868 3710 2989 2150 1748  801    1 1931 2046
##  670  885  900  934  959  974  986 1053 1269 1293 1297 1390 1456 1545 1600 1601
##  711 2426 3527 1047  465 1172 4543 2489 2721  840 3862 4304
## 1641 1655 1667 1686 1694 1773 1783 1813 1834 1875 1902 1907
```

```
which(abs(dfbetas[,7]) > cutoff_dfbetas) ##beta6
```

```
## 2609  634 3954 2015  299 3317 4226 1396  517 2336 4680 1537 4567 2155 5027 5043
##    1    5   10   16   21   31   33   34   36   55   61   63   92   98  121  124
```

```
## 3942 2767 2085  715  928 2666 4813 4933 2918 3409 4794 1185 4201 4199 3279 3685
##  155  161  181  189  197  230  241  255  270  275  287  309  319  320  325  338
## 4656  474 1720 4748 2914 3975  757 4450  530 3435 4334   71 1750 4517 2333 2153
##  341  352  357  372  390  392  411  412  416  433  439  443  457  470  478  492
##  324 3931 5024  848 3180 3251 1330 1096 3669 3586 4429 4435 3585 4284 4287 2998
##  495  501  550  555  558  564  579  594  603  605  621  624  655  660  669  671
##  203 5030 5012  298 4798 3219 2735 2448  239 3609 1296 4859 3821  510 3973 1599
##  677  687  689  694  704  708  712  728  737  741  745  746  747  759  765  772
## 4602 1317 2851 4103 1334 2374 3081   92 4396 1744  238 4504 2315 2811 2741  504
##  790  808  840  854  877  879  885  906  911  912  918  934  966  984  986 1007
## 4800 2372 2697 2488 1367 2052 4310 1184 3724 4329  126 2525 1009 1285 3281 3636
## 1013 1033 1049 1052 1057 1064 1066 1086 1106 1134 1148 1151 1155 1175 1185 1222
## 1319 4108 3277 2989 4394 1421 4956  219 2783 3567 3319  806  339 1748 1958 4724
## 1236 1264 1281 1293 1295 1310 1335 1341 1343 1357 1362 1366 1381 1390 1393 1404
## 2026  564  525 4790 3220 4156  407 3029  557 2814 4573 1751 4369 3080 2738 4726
## 1419 1424 1434 1449 1454 1458 1460 1463 1474 1477 1483 1490 1503 1522 1525 1550
## 3201 3511 2046 3519 3346  965 4978 2313  888 4004 4759   59 4058 1528 3422 4543
## 1555 1595 1601 1602 1603 1629 1643 1645 1659 1664 1688 1711 1712 1725 1777 1783
## 3274  803 4019  903 1618 4547 4729 3708 2394 4406 2589 3594  598 2154 3021  840
## 1788 1790 1791 1799 1801 1802 1809 1821 1845 1860 1864 1865 1866 1867 1868 1875
##  385 1824 4796 1074  838
## 1879 1893 1905 1915 1924
```

```
which(abs(dfbetas[,8]) > cutoff_dfbetas) ##beta7
```

```
##  356 4261 3954 2015 3944  299 3317 4226 1396 4680 2831 4297 5043 3942 2085  715
##    8    9   10   16   18   21   31   33   34   61   90  118  124  155  181  189
## 3783 2666 3417 4813 4283 4933 2541 3409 4794 4201 4199 3279 3685  474 2999 4748
##  205  230  233  241  246  255  274  275  287  319  320  325  338  352  355  372
## 2914 3975  757 4450 4334 4517 2333  324 3582 2034 5024 3180 4008 3251 2051 1330
##  390  392  411  412  439  470  478  495  521  547  550  558  559  564  576  579
## 1399 1040 1096 3586 2975 4435 3693 4924  353 3585 2455 4284 4287 5030 5012  298
##  592  593  594  605  606  624  633  647  653  655  657  660  669  687  689  694
## 4798 5034 2735  239 3609 4859 3821  510 3973 1599 4602 4453 4758 4103 2054 1071
##  704  707  712  737  741  746  747  759  765  772  790  806  810  854  880  904
##   92 4396 4504 3932 4800 2372 3706 2697 2488 1868 4310 3324 3724 4329 4188 1285
##  906  911  934  982 1013 1033 1041 1049 1052 1053 1066 1102 1106 1134 1168 1175
## 3281 4720 2126 2921 2532 4206 4108 3277 4988 2989 2150 1421 2783 3567 3319  339
## 1185 1204 1219 1230 1235 1251 1264 1281 1285 1293 1297 1310 1343 1357 1362 1381
## 1748 1958 4588 4790 3220 4156  407 3029 2814 3069 4573 2082 1751  837 4369 3080
## 1390 1393 1402 1449 1454 1458 1460 1463 1477 1480 1483 1488 1490 1493 1503 1522
## 2738 4726 3511 3519 3531  965 3406 2120 4759  690 1721 3203   59 4058 1528  505
## 1525 1550 1595 1602 1625 1629 1649 1680 1688 1702 1703 1704 1711 1712 1725 1746
## 1172 3422 3274 4019  903 1618 4547 3708 4406 2589 3594 3021  840  385  698
## 1773 1777 1788 1791 1799 1801 1802 1821 1860 1864 1865 1868 1875 1879 1926
```

```
which(abs(dfbetas[,9]) > cutoff_dfbetas) ##beta8
```

```
## 3954 2015  299 4226 1396 2157 4816 4680 3578 3942 2085  715 4749 3820 2666 2151
##   10   16   21   33   34   39   56   61  114  155  181  189  200  227  230  231
## 4813 4933 3409 4201 4199 3279 3685  474 4748 2914 3975 2588  757 4450 4334 4900
##  241  255  275  319  320  325  338  352  372  390  392  407  411  412  439  456
## 4517 2333  324 3016 5024 3180 3251 3554 1330 1096 3586 4435 3315 3523 4924 3585
##  470  478  495  535  550  558  564  567  579  594  605  624  638  640  647  655
## 4284 4287 5030 5012  298 4798 2735  239 3609 4859 3821  510 3989 3973 1599 4602
##  660  669  687  689  694  704  712  737  741  746  747  759  763  765  772  790
## 4103  328   92 4396 4504  927 2741 4800 2372 2697 2488 4310 3468 3324 2621 3724
##  854  900  906  911  934  959  986 1013 1033 1049 1052 1066 1087 1102 1105 1106
## 4329 4188 1285 3281 4720 4108 3277 4988 2989 1421 2764 2783 3567 3319  383  339
## 1134 1168 1175 1185 1204 1264 1281 1285 1293 1310 1318 1343 1357 1362 1373 1381
## 1748 1958 4790 3220 4156  407 3029 2814 4573 1751 4369 3245 3080 2738 4726 3511
## 1390 1393 1449 1454 1458 1460 1463 1477 1483 1490 1503 1515 1522 1525 1550 1595
## 3519 3229  965 2426 4004 3093 4759   59 4058 1528  701  505 1869 1172 3422 3274
## 1602 1618 1629 1655 1664 1670 1688 1711 1712 1725 1726 1746 1764 1773 1777 1788
## 4019  903 1618 4547 3708 3885 4406 2589 3594 3021  840  385 2616
## 1791 1799 1801 1802 1821 1843 1860 1864 1865 1868 1875 1879 1923
```

Model selection

ANOVA and t-test

```
anova(model_4)
```

```
## Analysis of Variance Table
##
## Response: t_gross
##                           Df Sum Sq Mean Sq   F value    Pr(>F)
## log(budget)                1 614365  614365 1417.4354 < 2.2e-16 ***
## t_numcritic                1 125354  125354  289.2121 < 2.2e-16 ***
## t_duration                 1   1165    1165    2.6877    0.1013
## t_imdb_score               1  10385   10385   23.9602 1.065e-06 ***
## actor_1_facebook_likes     1    638     638    1.4731    0.2250
## content_rating             3  55122   18374   42.3913 < 2.2e-16 ***
## Residuals               1933 837828     433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_4)
```

```
##
## Call:
```

```
## lm(formula = t_gross ~ log(budget) + t_numcritic + t_duration +
##     t_imdb_score + actor_1_facebook_likes + content_rating, data = data_outlier)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -94.51 -12.58   0.59  13.57  85.50
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.121e+02  1.087e+01 -10.307  < 2e-16 ***
## log(budget)             8.495e+00  3.823e-01  22.220  < 2e-16 ***
## t_numcritic             5.701e+00  4.126e-01  13.815  < 2e-16 ***
## t_duration             -2.014e+01  6.385e+01  -0.315 0.752420
## t_imdb_score            6.225e-02  1.143e-02   5.449 5.73e-08 ***
## actor_1_facebook_likes  3.024e-05  2.729e-05   1.108 0.267959
## content_ratingPG        1.435e+01  2.315e+00   6.201 6.84e-10 ***
## content_ratingPG-13     8.164e+00  2.171e+00   3.761 0.000174 ***
## content_ratingR        -7.577e-01  2.052e+00  -0.369 0.711973
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.82 on 1933 degrees of freedom
## Multiple R-squared:  0.4906, Adjusted R-squared:  0.4885
## F-statistic: 232.7 on 8 and 1933 DF,  p-value: < 2.2e-16
```

The p-value of t-duration and actor_1_facebook_likes are larger than 0.05. Therefore, we temporarily drop them in the new model.

Partial f-test

```
model_5 <- lm(t_gross ~ log(budget) + t_numcritic + t_imdb_score + content_rating,data =
anova(model_5, model_4)
```

```
## Analysis of Variance Table
##
## Model 1: t_gross ~ log(budget) + t_numcritic + t_imdb_score + content_rating
## Model 2: t_gross ~ log(budget) + t_numcritic + t_duration + t_imdb_score +
##     actor_1_facebook_likes + content_rating
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1935 838404
## 2   1933 837828  2    576.03 0.6645 0.5146
```

The Partial F-test results with p-values of 0.6645 and 0.5146 suggest that the additional variables in model_4 do not significantly improve the model fit compared to model_5.

Adjusted R^2

```
summary(model_4)
```

```
##
## Call:
## lm(formula = t_gross ~ log(budget) + t_numcritic + t_duration +
##     t_imdb_score + actor_1_facebook_likes + content_rating, data = data_outlier)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -94.51 -12.58   0.59  13.57  85.50
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.121e+02  1.087e+01 -10.307  < 2e-16 ***
## log(budget)             8.495e+00  3.823e-01  22.220  < 2e-16 ***
## t_numcritic             5.701e+00  4.126e-01  13.815  < 2e-16 ***
## t_duration             -2.014e+01  6.385e+01  -0.315 0.752420
## t_imdb_score            6.225e-02  1.143e-02   5.449 5.73e-08 ***
## actor_1_facebook_likes  3.024e-05  2.729e-05   1.108 0.267959
## content_ratingPG        1.435e+01  2.315e+00   6.201 6.84e-10 ***
## content_ratingPG-13     8.164e+00  2.171e+00   3.761 0.000174 ***
## content_ratingR        -7.577e-01  2.052e+00  -0.369 0.711973
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.82 on 1933 degrees of freedom
## Multiple R-squared:  0.4906, Adjusted R-squared:  0.4885
## F-statistic: 232.7 on 8 and 1933 DF,  p-value: < 2.2e-16
```

```
summary(model_5)
```

```
##
## Call:
## lm(formula = t_gross ~ log(budget) + t_numcritic + t_imdb_score +
##     content_rating, data = train_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -95.303 -12.652   0.629  13.494  85.517
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -115.67107    5.67774 -20.373  < 2e-16 ***
```

```
## log(budget)            8.56970    0.35514  24.131  < 2e-16 ***
## t_numcritic            5.72736    0.40992  13.972  < 2e-16 ***
## t_imdb_score           0.06460    0.01028   6.285 4.05e-10 ***
## content_ratingPG      14.45511    2.31148   6.254 4.92e-10 ***
## content_ratingPG-13    8.37339    2.14294   3.907 9.65e-05 ***
## content_ratingR       -0.57652    2.02642  -0.285    0.776
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.82 on 1935 degrees of freedom
## Multiple R-squared:  0.4903, Adjusted R-squared:  0.4887
## F-statistic: 310.2 on 6 and 1935 DF,  p-value: < 2.2e-16
```

AIC, BIC

```
AIC(model_4)
```

```
## [1] 17313.46
```

```
BIC(model_4)
```

```
## [1] 17369.17
```

```
AIC(model_5)
```

```
## [1] 17310.79
```

```
BIC(model_5)
```

```
## [1] 17355.36
```

Model validation

Fit the final model in the test data set

```
test <- na.omit(test) # clean the data set
test$content_rating <- ifelse(test$content_rating %in% c("R", "PG", "PG-13"),
                              test$content_rating,
                              "other")

test$t_gross <- test$gross ** (1/4)
test$t_budget <- log(test$budget)
test$t_numcritic <- test$num_critic_for_reviews ** (1/3)
test$t_imdb_score <- test$imdb_score ** (2.6)
```

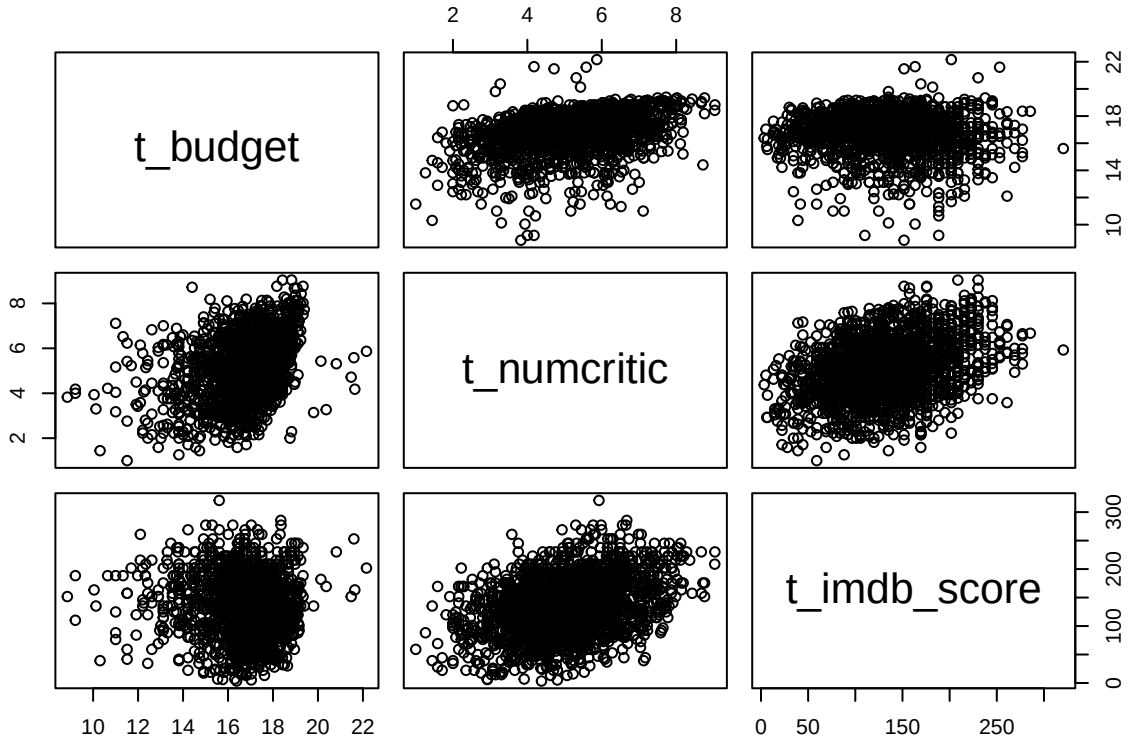
```r
model_test <- lm(t_gross ~ t_budget + t_numcritic + t_imdb_score + content_rating, data=
summary(model_test)
```

```
##
## Call:
## lm(formula = t_gross ~ t_budget + t_numcritic + t_imdb_score +
##     content_rating, data = test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.671 -12.614   1.787  13.171  76.806
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -99.12723    6.12396 -16.187  < 2e-16 ***
## t_budget              7.86335    0.37812  20.796  < 2e-16 ***
## t_numcritic           6.55978    0.42783  15.333  < 2e-16 ***
## t_imdb_score          0.04542    0.01033   4.398 1.15e-05 ***
## content_ratingPG     10.12756    2.49378   4.061 5.08e-05 ***
## content_ratingPG-13   1.97114    2.33465   0.844  0.39861
## content_ratingR      -7.64517    2.23834  -3.416  0.00065 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.77 on 1882 degrees of freedom
## Multiple R-squared:  0.4586, Adjusted R-squared:  0.4569
## F-statistic: 265.7 on 6 and 1882 DF,  p-value: < 2.2e-16
```

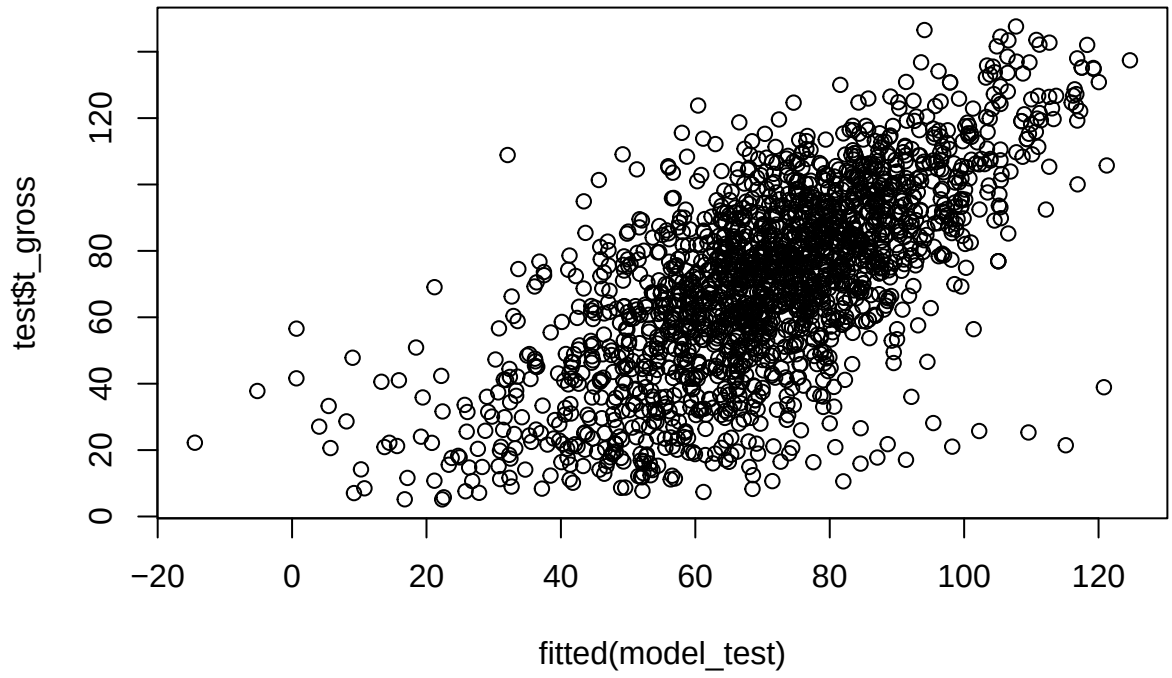
final model checking

Check conditions

```r
# conditional mean predictors
predictor_vars <- test[, c("t_budget", "t_numcritic", "t_imdb_score")] # select only th
pairs(predictor_vars) # use pairs() to create the scatterplot matrix
```




```
# conditional mean response
plot(test$t_gross ~ fitted(model_test))
```

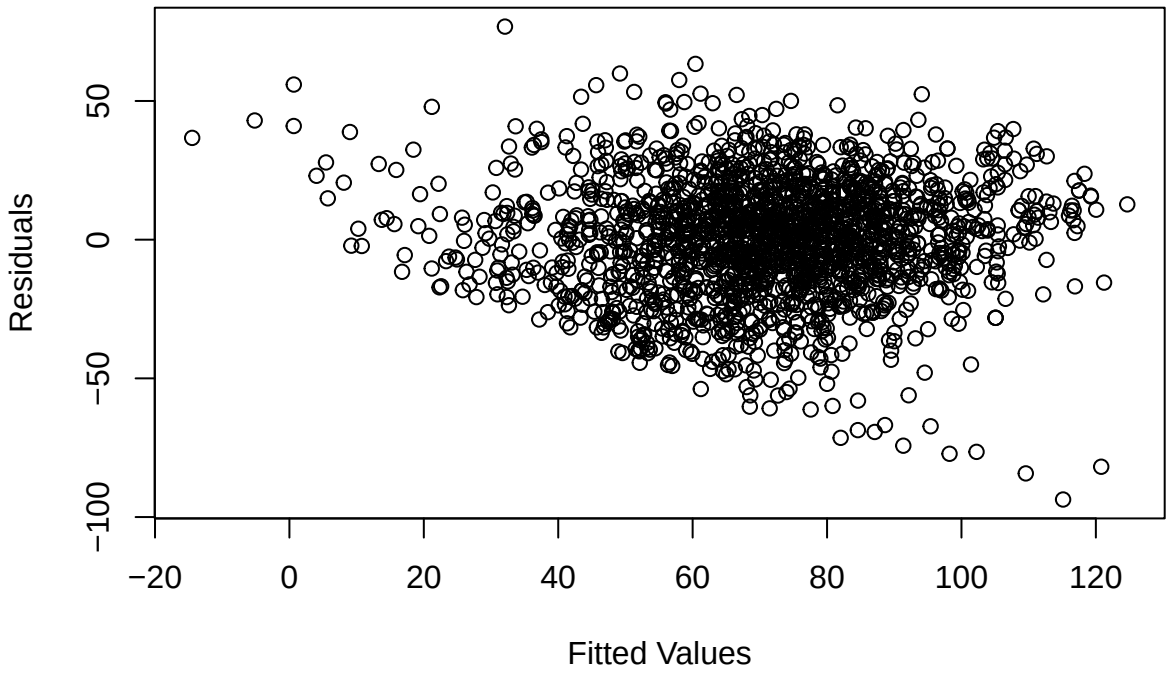


##

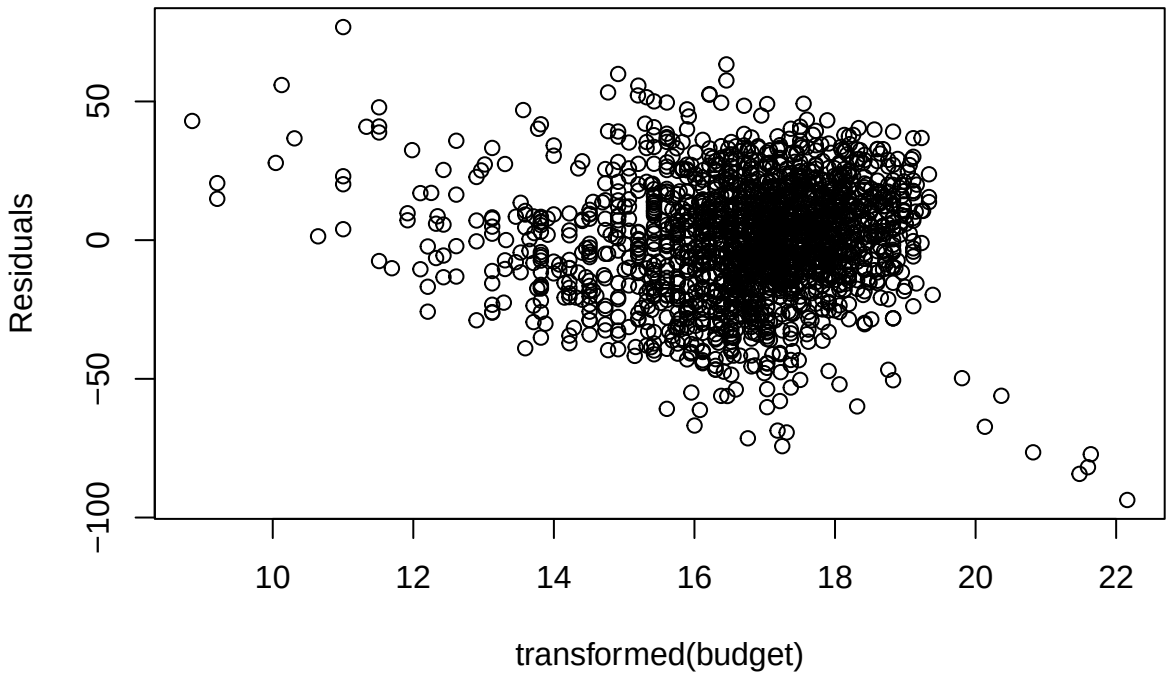
Check assumptions

```
# useful values
e_hat_test <- resid(model_test)
y_hat_test <- fitted(model_test)
```

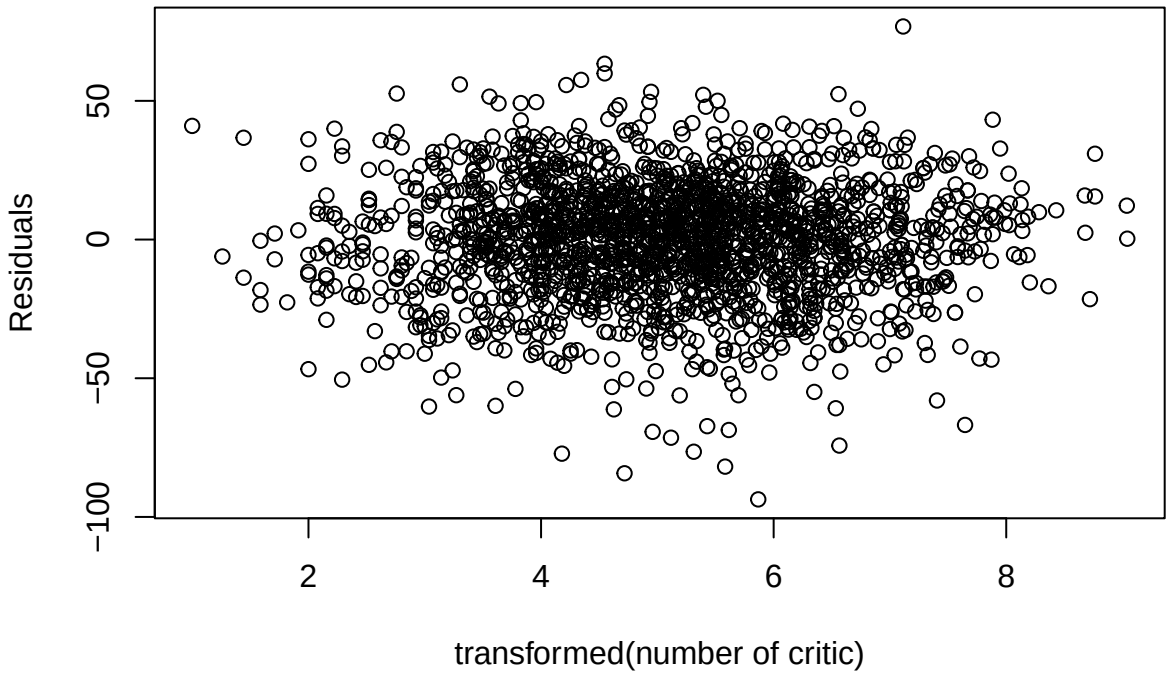
```r
# residual vs fitted
plot(e_hat_test ~ y_hat_test, xlab ="Fitted Values", ylab="Residuals")
```



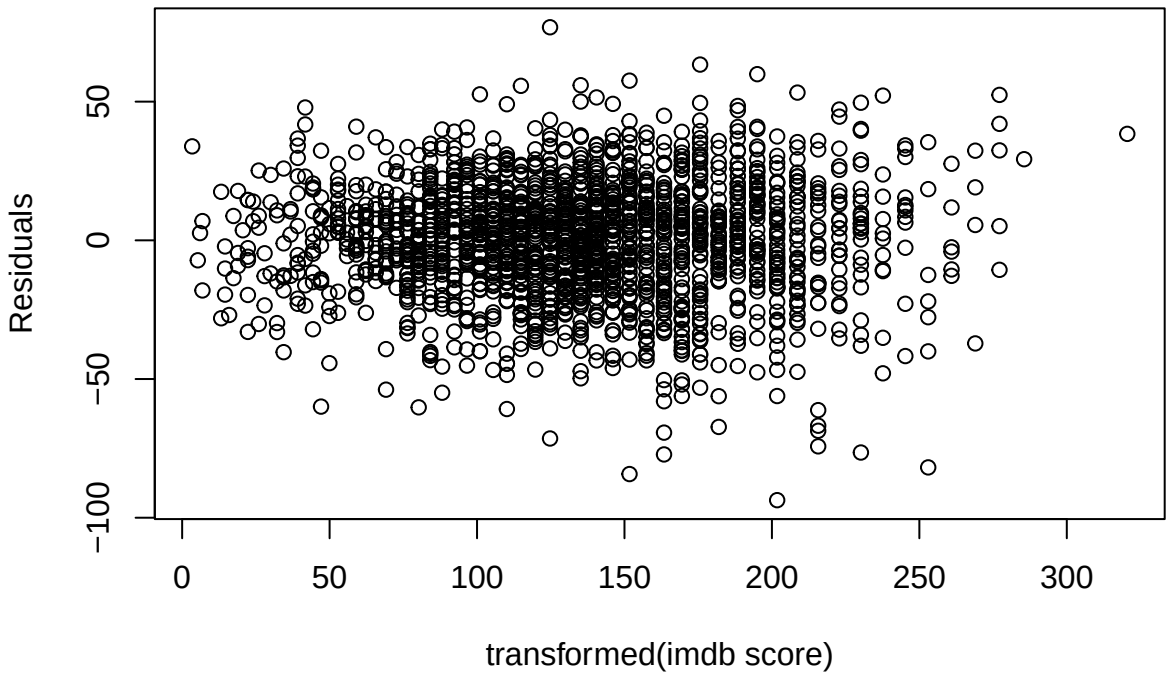

```r
# residual vs predictors
plot(e_hat_test ~ test$t_budget,xlab ="transformed(budget)", ylab = "Residuals")
```



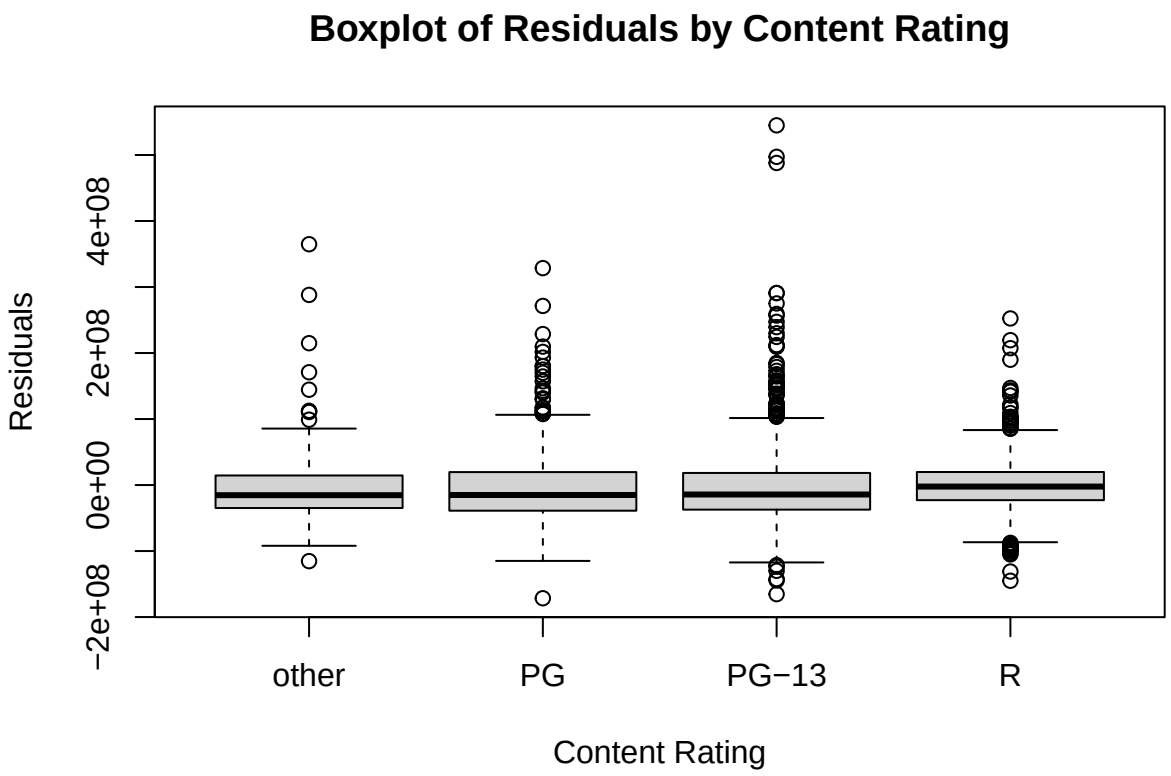
```r
plot(e_hat_test ~ test$t_numcritic,xlab ="transformed(number of critic)", ylab = "Residu
```



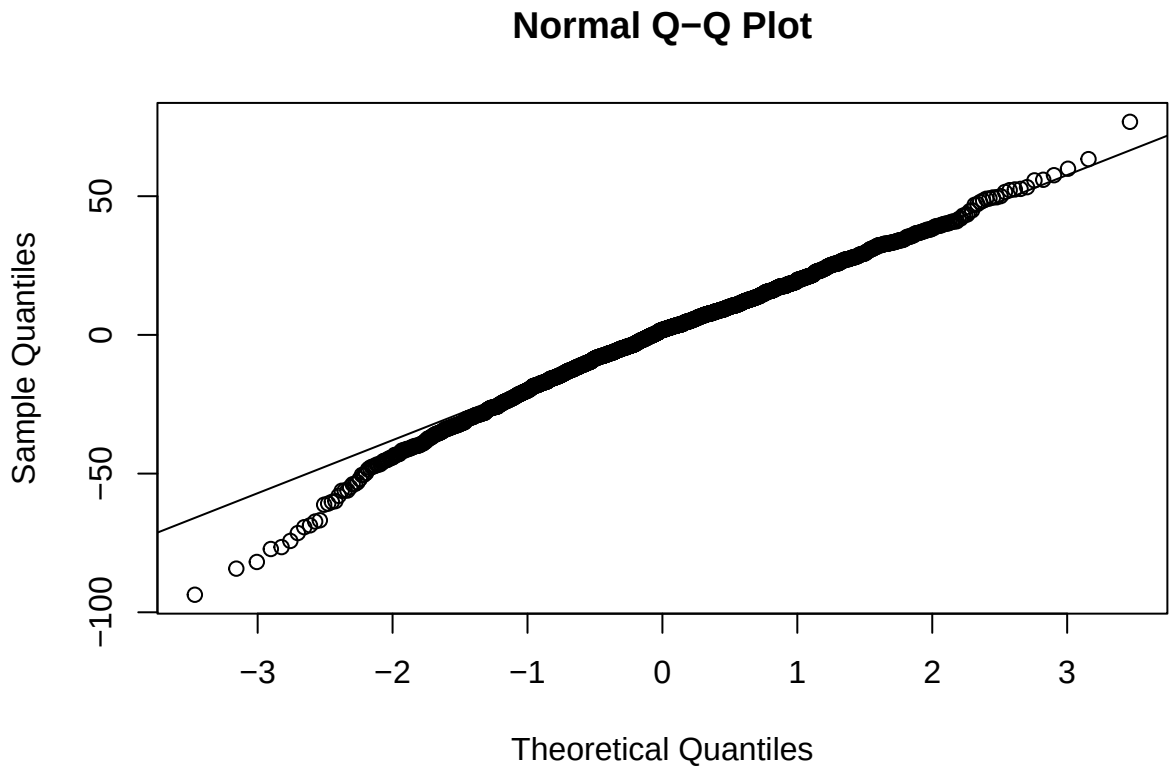

```r
plot(e_hat_test ~ test$t_imdb_score,xlab ="transformed(imdb score)", ylab = "Residuals")
```




```r
boxplot(residuals(model_1) ~ train_1$content_rating,
        xlab="Content Rating",
        ylab="Residuals",
        main="Boxplot of Residuals by Content Rating")
```




```
# qq plot
qqnorm(e_hat_test)
qqline(e_hat_test)
```



##

Check multicollinearity

```r
vif(model_test)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## t_budget       1.338243  1        1.156825
## t_numcritic    1.410735  1        1.187743
## t_imdb_score   1.229828  1        1.108976
## content_rating 1.195484  3        1.030206
```

Problematic observations

```r
# useful values:
n <- nrow(test)
p <- 5 # 5 is used here instead of length(coef(model_test))-1 b/c it counts all the su

# leverage
h_cut <- 2*(p+1)/n
h_ii <- hatvalues(model_test)
print("high leverage")
```

```
## [1] "high leverage"
```

```r
which(h_ii > h_cut)
```

```
##   34   36   42   44   68   79   90  120  129  210  224  271  278  341  354  363
##   11   12   16   18   28   33   37   48   54   95  101  125  129  157  162  165
##  449  501  584  658  964 1079 1144 1157 1161 1191 1339 1501 1518 1568 1589 1609
##  209  229  264  296  430  477  512  519  521  538  602  676  682  708  716  728
## 1610 1636 1703 1882 1934 2008 2038 2045 2115 2184 2186 2189 2259 2284 2302 2324
##  729  738  771  840  867  901  916  920  957  985  987  988 1015 1029 1037 1045
## 2335 2340 2364 2383 2513 2620 2645 2686 2762 2835 2944 2984 2993 3006 3025 3060
## 1053 1055 1061 1066 1127 1176 1186 1211 1240 1270 1320 1330 1334 1341 1349 1364
## 3110 3143 3221 3231 3263 3268 3282 3312 3360 3376 3424 3466 3467 3522 3547 3560
## 1380 1396 1426 1431 1437 1440 1442 1453 1462 1469 1490 1506 1507 1534 1539 1545
## 3572 3589 3597 3683 3704 3771 3830 3832 3840 3852 3860 3878 3890 3891 3971 4025
## 1548 1554 1559 1588 1595 1618 1629 1630 1632 1637 1641 1645 1649 1650 1668 1678
## 4048 4067 4084 4129 4152 4159 4202 4217 4239 4242 4248 4290 4292 4350 4353 4354
## 1684 1690 1695 1705 1709 1711 1723 1729 1732 1734 1735 1744 1745 1751 1752 1753
## 4362 4386 4428 4472 4512 4521 4531 4538 4541 4546 4572 4586 4591 4593 4639 4641
## 1756 1761 1768 1773 1781 1783 1784 1785 1786 1789 1797 1799 1801 1802 1807 1808
## 4672 4675 4689 4692 4707 4708 4711 4727 4751 4752 4756 4785 4789 4792 4801 4803
## 1818 1819 1824 1825 1827 1828 1830 1832 1838 1839 1840 1844 1845 1846 1849 1850
## 4815 4833 4841 4846 4864 4869 4874 4891 4894 4916 4922 4923 4927 4931 4932 4937
## 1852 1855 1856 1858 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872
## 4942 4957 4959 4960 4965 4974 4976 4979 4985 4998 5005 5009 5016 5026 5028 5036
## 1873 1874 1875 1876 1877 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889
```

```r
# outlier
r_i <- rstandard(model_test)
print("outliers (large)")
```

```
## [1] "outliers (large)"
```

```r
which(rstandard(model_test) > 4 | rstandard(model_test) < -4)
```

```
## 2335 3860
## 1053 1641
```

```r
# Cook's Distance
D_cut <- qf(0.5, p+1, n-p-1)
D_i <- cooks.distance(model_test)
print("Cooks")
```

```
## [1] "Cooks"
```

```r
which(cooks.distance(model_test) > D_cut)
```

```
## named integer(0)
```

```r
# DFFITS
fits_cut <- 2*sqrt((p+1)/n)
dffits_i <- dffits(model_test)
print("DFFITS")
```

```
## [1] "DFFITS"
```

```r
which(abs(dffits(model_test)) > fits_cut)
```

```
##   36   44  241  271  341  354  393  449  520  583  584  637  658  815 1079 1144
##   12   18  108  125  157  162  176  209  236  263  264  287  296  370  477  512
## 1152 1182 1191 1323 1326 1339 1497 1518 1525 1556 1589 1607 1609 1610 1703 1867
##  517  533  538  595  598  602  674  682  686  701  716  727  728  729  771  836
## 1874 1882 2003 2008 2045 2048 2062 2149 2324 2335 2340 2364 2379 2494 2590 2601
##  838  840  898  901  920  921  927  969 1045 1053 1055 1061 1065 1117 1160 1164
## 2602 2603 2620 2645 2747 2762 2819 2833 2835 2850 2916 2917 2920 2993 2995 3006
## 1165 1166 1176 1186 1235 1240 1263 1268 1270 1277 1304 1305 1306 1334 1336 1341
## 3025 3125 3206 3221 3231 3236 3242 3265 3312 3344 3360 3368 3424 3467 3547 3558
## 1349 1389 1422 1426 1431 1433 1434 1438 1453 1456 1462 1466 1490 1507 1539 1544
## 3572 3583 3683 3702 3738 3794 3832 3840 3842 3852 3860 3863 3868 3878 3891 3894
## 1548 1551 1588 1594 1608 1623 1630 1632 1633 1637 1641 1642 1643 1645 1650 1651
## 3940 3971 4025 4034 4047 4053 4169 4217 4290 4292 4428 4521 4531 4546 4586 4591
## 1664 1668 1678 1680 1683 1687 1713 1729 1744 1745 1768 1783 1784 1789 1799 1801
## 4641 4672 4675 4708 4727 4756 4785 4792 4803 4815 4937 4942 4957 4959 4960 4965
## 1808 1818 1819 1828 1832 1840 1844 1846 1850 1852 1872 1873 1874 1875 1876 1877
## 4976 4985 5005 5016 5026 5028 5036
```

```
## 1880 1882 1884 1886 1887 1888 1889
```

```
# DFBETAS
beta_cut <- 2/sqrt(n)
dfbetas_i <- dfbetas(model_test)
for(i in 1:(p+1)){
print(paste0("Beta ", i-1))
print(which(abs(dfbetas(model_test)[,i]) > beta_cut)) }
```

```
## [1] "Beta 0"
##   14   28  206  271  341  400  410  583  588  637  931 1144 1152 1193 1324 1339
##    3    9   93  125  157  181  186  263  266  287  417  512  517  540  596  602
## 1525 2324 2335 2620 2762 2835 2880 2993 3006 3100 3231 3265 3312 3360 3424 3660
##  686 1045 1053 1176 1240 1270 1289 1334 1341 1377 1431 1438 1453 1462 1490 1578
## 3683 3702 3738 3832 3840 3852 3860 3863 3878 3886 3891 3894 3940 3971 4012 4053
## 1588 1594 1608 1630 1632 1637 1641 1642 1645 1648 1650 1651 1664 1668 1677 1687
## 4059 4125 4152 4160 4170 4182 4196 4208 4209 4217 4290 4292 4299 4350 4362 4386
## 1688 1704 1709 1712 1714 1716 1719 1726 1727 1729 1744 1745 1746 1751 1756 1761
## 4428 4479 4485 4486 4521 4531 4546 4586 4591 4595 4641 4647 4672 4675 4708 4727
## 1768 1774 1775 1776 1783 1784 1789 1799 1801 1803 1808 1812 1818 1819 1828 1832
## 4751 4752 4756 4781 4785 4792 4793 4803 4815 4833 4853 4869 4894 4916 4923 4932
## 1838 1839 1840 1842 1844 1846 1847 1850 1852 1855 1859 1862 1865 1866 1868 1871
## 4937 4942 4957 4959 4960 4965 4974 4976 4985 5005 5016 5026 5028 5036
## 1872 1873 1874 1875 1876 1877 1879 1880 1882 1884 1886 1887 1888 1889
## [1] "Beta 1"
##   14   44  296  354  393  400  410  449  583  584  588  637  658  666  931 1142
##    3   18  137  162  176  181  186  209  263  264  266  287  296  298  417  511
## 1152 1193 1324 1339 1525 1589 1610 1874 2008 2045 2324 2335 2340 2620 2880 3006
##  517  540  596  602  686  716  729  838  901  920 1045 1053 1055 1176 1289 1341
## 3025 3215 3221 3265 3312 3344 3424 3660 3702 3738 3826 3832 3852 3860 3863 3878
## 1349 1425 1426 1438 1453 1456 1490 1578 1594 1608 1627 1630 1637 1641 1642 1645
## 3894 3940 4012 4034 4047 4053 4059 4125 4160 4169 4170 4176 4196 4208 4209 4217
## 1651 1664 1677 1680 1683 1687 1688 1704 1712 1713 1714 1715 1719 1726 1727 1729
## 4290 4299 4347 4354 4362 4386 4479 4485 4486 4531 4546 4586 4591 4595 4641 4647
## 1744 1746 1750 1753 1756 1761 1774 1775 1776 1784 1789 1799 1801 1803 1808 1812
## 4672 4675 4708 4709 4727 4752 4756 4781 4785 4792 4793 4803 4815 4853 4869 4894
## 1818 1819 1828 1829 1832 1839 1840 1842 1844 1846 1847 1850 1852 1859 1862 1865
## 4916 4932 4937 4942 4957 4959 4960 4965 4974 4976 4985 5005 5016 5026 5028 5036
## 1866 1871 1872 1873 1874 1875 1876 1877 1879 1880 1882 1884 1886 1887 1888 1889
## [1] "Beta 2"
##   37   54   55  175  187  188  241  410  429  440  588  700  815  930  934 1123
##   13   23   24   77   87   88  108  186  197  205  266  315  370  416  418  502
## 1152 1191 1193 1324 1325 1326 1339 1379 1411 1497 1501 1525 1597 1612 1704 1838
##  517  538  540  596  597  598  602  617  636  674  676  686  721  730  772  825
## 1842 1867 1874 1911 2003 2018 2045 2062 2149 2160 2198 2301 2311 2324 2328 2335
```

```
##  829  836  838  853  898  903  920  927  969  972  992 1036 1041 1045 1049 1053
## 2338 2340 2359 2392 2393 2469 2585 2599 2601 2603 2611 2612 2620 2727 2747 2765
## 1054 1055 1058 1073 1074 1107 1159 1162 1164 1166 1171 1172 1176 1230 1235 1242
## 2784 2816 2833 2834 2880 2916 2993 2995 3002 3006 3027 3060 3115 3125 3197 3221
## 1250 1262 1268 1269 1289 1304 1334 1336 1340 1341 1350 1364 1382 1389 1417 1426
## 3236 3242 3265 3285 3312 3334 3344 3368 3407 3421 3424 3436 3464 3517 3522 3524
## 1433 1434 1438 1443 1453 1455 1456 1466 1482 1488 1490 1493 1505 1532 1534 1535
## 3541 3549 3556 3557 3558 3572 3584 3683 3702 3707 3722 3729 3738 3794 3824 3826
## 1537 1540 1542 1543 1544 1548 1552 1588 1594 1597 1603 1605 1608 1623 1626 1627
## 3832 3842 3852 3860 3863 3878 3894 3946 3988 4034 4053 4059 4125 4160 4209 4241
## 1630 1633 1637 1641 1642 1645 1651 1666 1672 1680 1687 1688 1704 1712 1727 1733
## 4242 4319 4347 4362 4486 4546 4558 4586 4595 4672 4708 4727 4752 4756 4792 4793
## 1734 1748 1750 1756 1776 1789 1791 1799 1803 1818 1828 1832 1839 1840 1846 1847
## 4803 4815 4937 4957 4959 4965 5036
## 1850 1852 1872 1874 1875 1877 1889
## [1] "Beta 3"
##   44  206  271  322  341  354  393  397  401  449  628  700  809  931 1139 1182
##   18   93  125  147  157  162  176  178  182  209  283  315  367  417  509  533
## 1191 1193 1323 1339 1379 1394 1401 1404 1411 1525 1556 1589 1601 1607 1609 1612
##  538  540  595  602  617  626  629  631  636  686  701  716  724  727  728  730
## 1703 1994 1999 2003 2005 2018 2045 2048 2159 2160 2163 2324 2334 2335 2364 2494
##  771  893  897  898  900  903  920  921  971  972  974 1045 1052 1053 1061 1117
## 2528 2531 2542 2543 2573 2601 2602 2607 2612 2645 2746 2761 2819 2830 2833 2835
## 1131 1132 1137 1138 1152 1164 1165 1170 1172 1186 1234 1239 1263 1267 1268 1270
## 2850 2880 2891 2915 2917 2920 2984 3006 3010 3025 3027 3060 3078 3197 3231 3235
## 1277 1289 1296 1303 1305 1306 1330 1341 1343 1349 1350 1364 1371 1417 1431 1432
## 3242 3265 3312 3345 3360 3365 3424 3457 3467 3541 3557 3558 3583 3678 3702 3707
## 1434 1438 1453 1457 1462 1465 1490 1501 1507 1537 1543 1544 1551 1584 1594 1597
## 3794 3832 3840 3850 3852 3855 3860 3863 3868 3878 3894 3950 3971 3988 4034 4047
## 1623 1630 1632 1636 1637 1638 1641 1642 1643 1645 1651 1667 1668 1672 1680 1683
## 4053 4059 4071 4106 4125 4159 4160 4169 4241 4290 4319 4428 4531 4546 4672 4708
## 1687 1688 1692 1700 1704 1711 1712 1713 1733 1744 1748 1768 1784 1789 1818 1828
## 4709 4727 4752 4815 4957 4959 5005
## 1829 1832 1839 1852 1874 1875 1884
## [1] "Beta 4"
##   36   42   44  120  210  241  354  363  393  449  520  583  584  658  964 1079
##   12   16   18   48   95  108  162  165  176  209  236  263  264  296  430  477
## 1136 1144 1157 1161 1323 1364 1489 1497 1518 1556 1568 1574 1589 1609 1610 1882
##  508  512  519  521  595  614  670  674  682  701  708  710  716  728  729  840
## 2008 2045 2048 2062 2084 2115 2184 2186 2189 2328 2335 2340 2364 2379 2383 2602
##  901  920  921  927  940  957  985  987  988 1049 1053 1055 1061 1065 1066 1165
## 2603 2604 2605 2645 2762 2819 2835 2916 2920 2993 3006 3025 3110 3195 3206 3221
## 1166 1167 1168 1186 1240 1263 1270 1304 1306 1334 1341 1349 1380 1416 1422 1426
## 3231 3263 3268 3282 3316 3344 3360 3424 3466 3547 3572 3589 3597 3683 3704 3738
## 1431 1437 1440 1442 1454 1456 1462 1490 1506 1539 1548 1554 1559 1588 1595 1608
```

```
## 3794 3840 3860 3878 3891 3940 3971 4025 4048 4152 4159 4202 4217 4239 4290 4292
## 1623 1632 1641 1645 1650 1664 1668 1678 1684 1709 1711 1723 1729 1732 1744 1745
## 4350 4386 4428 4521 4531 4586 4591 4641 4672 4675 4708 4751 4785 4792 4803 4815
## 1751 1761 1768 1783 1784 1799 1801 1808 1818 1819 1828 1838 1844 1846 1850 1852
## 4833 4894 4959 4965 4985 5026 5028 5036
## 1855 1865 1875 1877 1882 1887 1888 1889
## [1] "Beta 5"
##   36   42   44  120  210  354  363  449  584  658  964 1079 1144 1157 1161 1518
##   12   16   18   48   95  162  165  209  264  296  430  477  512  519  521  682
## 1568 1589 1607 1609 1610 1882 2008 2045 2115 2184 2186 2189 2340 2383 2494 2762
##  708  716  727  728  729  840  901  920  957  985  987  988 1055 1066 1117 1240
## 2835 2993 3006 3110 3221 3263 3265 3268 3282 3360 3466 3547 3560 3572 3589 3597
## 1270 1334 1341 1380 1426 1437 1438 1440 1442 1462 1506 1539 1545 1548 1554 1559
## 3683 3704 3840 3860 3878 3891 3971 4025 4048 4152 4202 4217 4239 4292 4350 4428
## 1588 1595 1632 1641 1645 1650 1668 1678 1684 1709 1723 1729 1732 1745 1751 1768
## 4521 4586 4641 4672 4708 4751 4785 4803 4815 4833 4894 4959 4965 4985 5026 5028
## 1783 1799 1808 1818 1828 1838 1844 1850 1852 1855 1865 1875 1877 1882 1887 1888
## 5036
## 1889
```