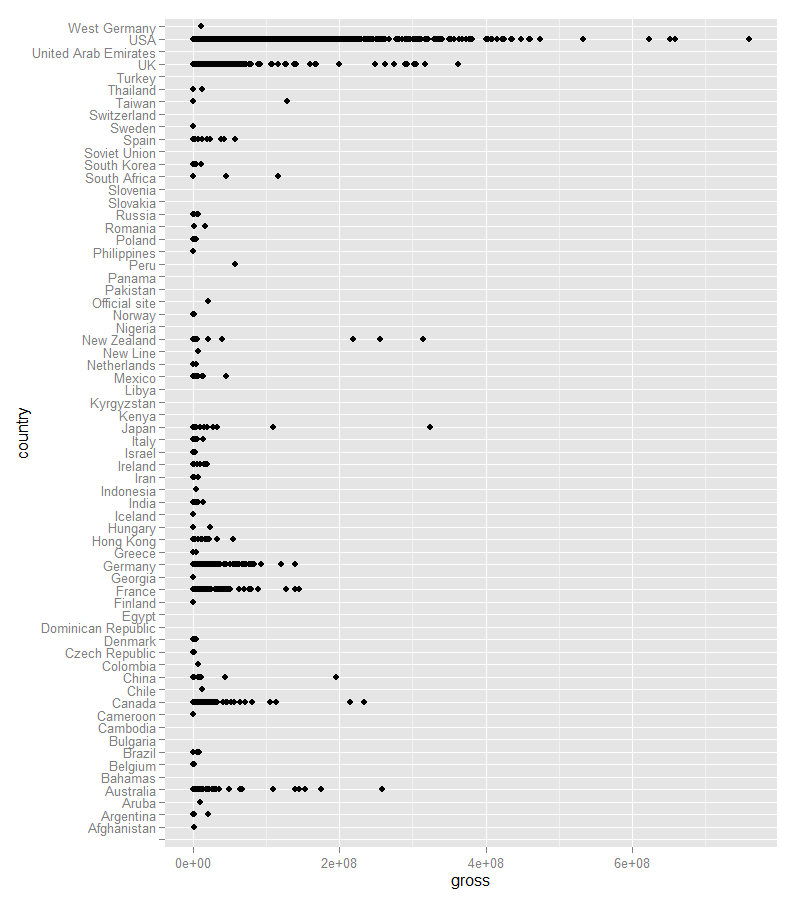
1. Plot a diagram and explain what you can understand about the gross earnings of the movies in this data set from your diagram.

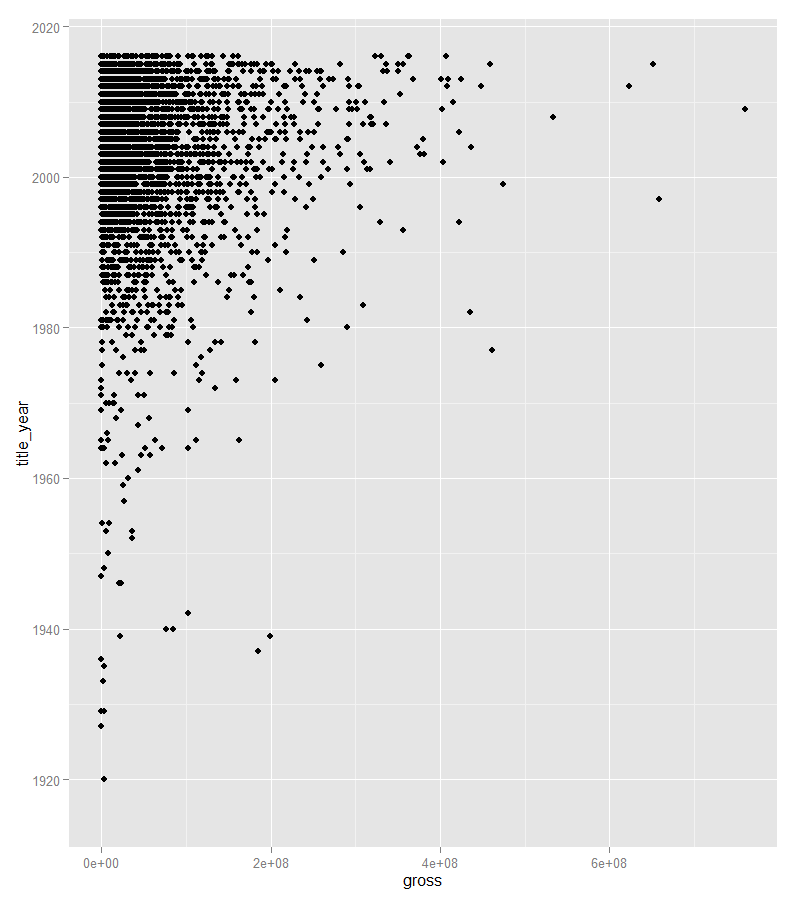
Based on Below Graphs we can understand

1. Movie taken in USA has high gross.
2. Gross is increasing rapidly in recent years.
3. IMbd score 6-9 has high chance more gross income.

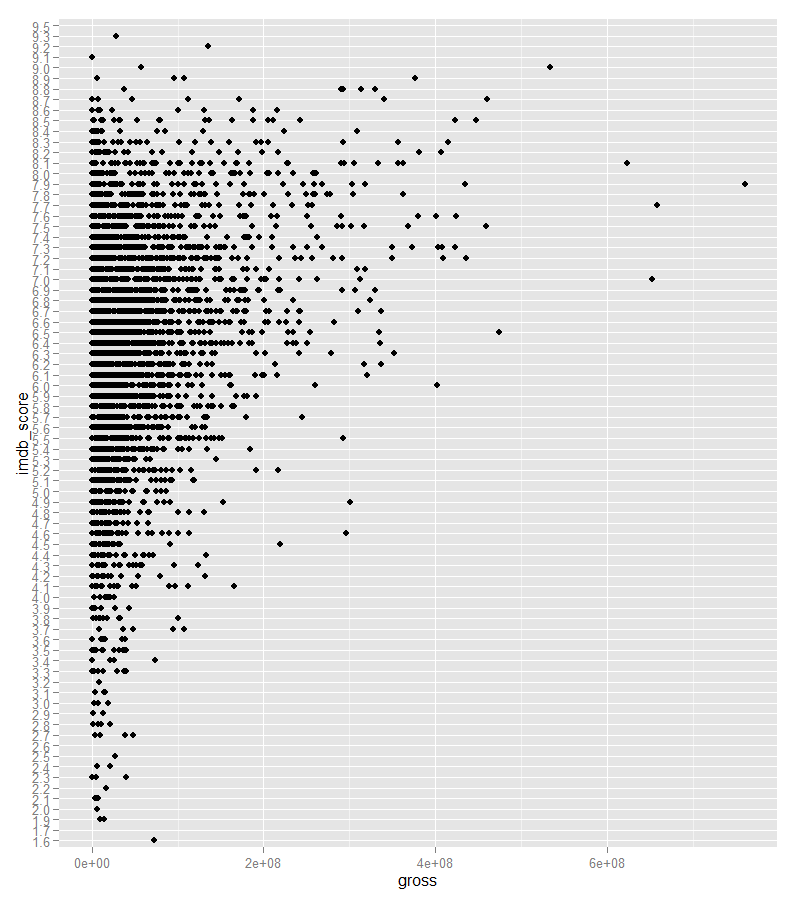
**Country VS Gross**



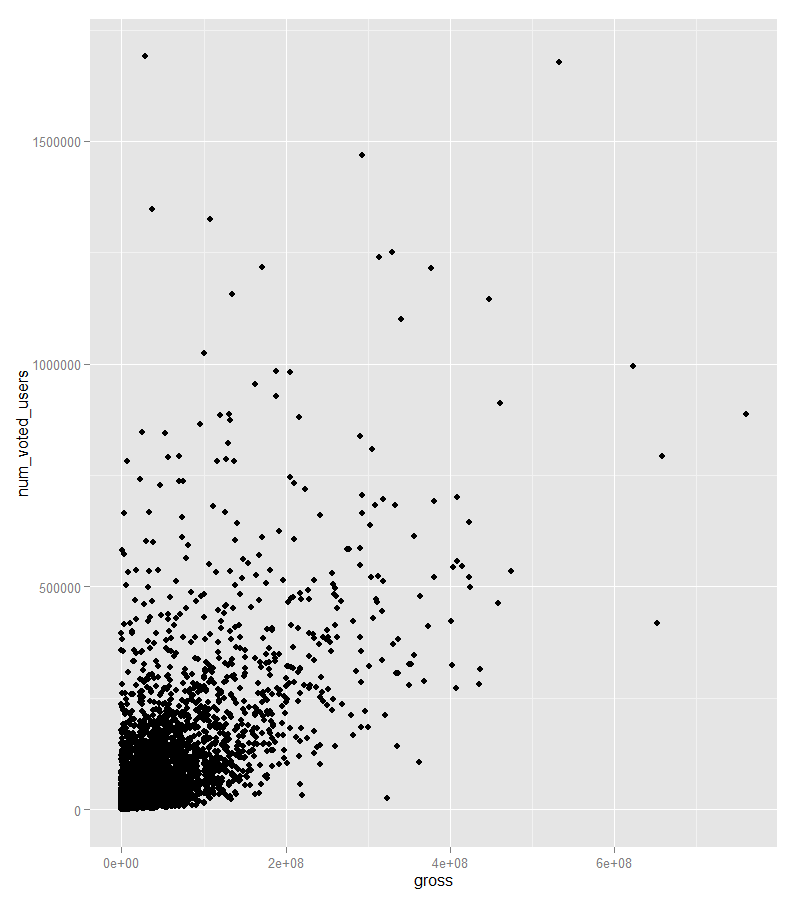
**Year Vs Gross**



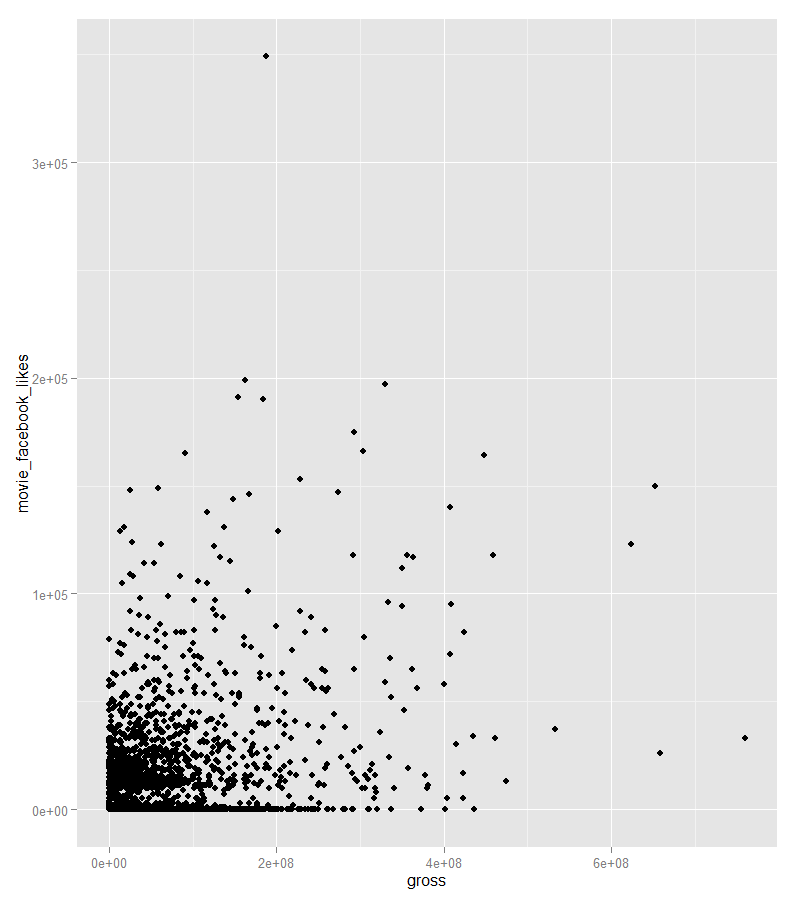
**Imbd\_Score vs Gross**



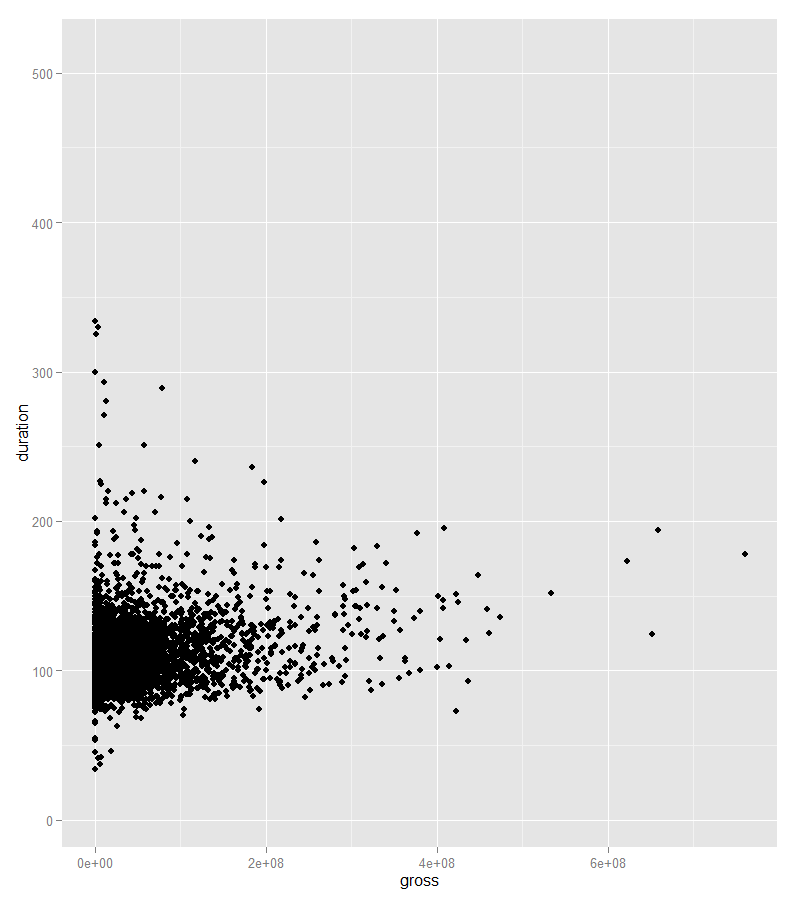
**“No of Voted user” vs Gross**



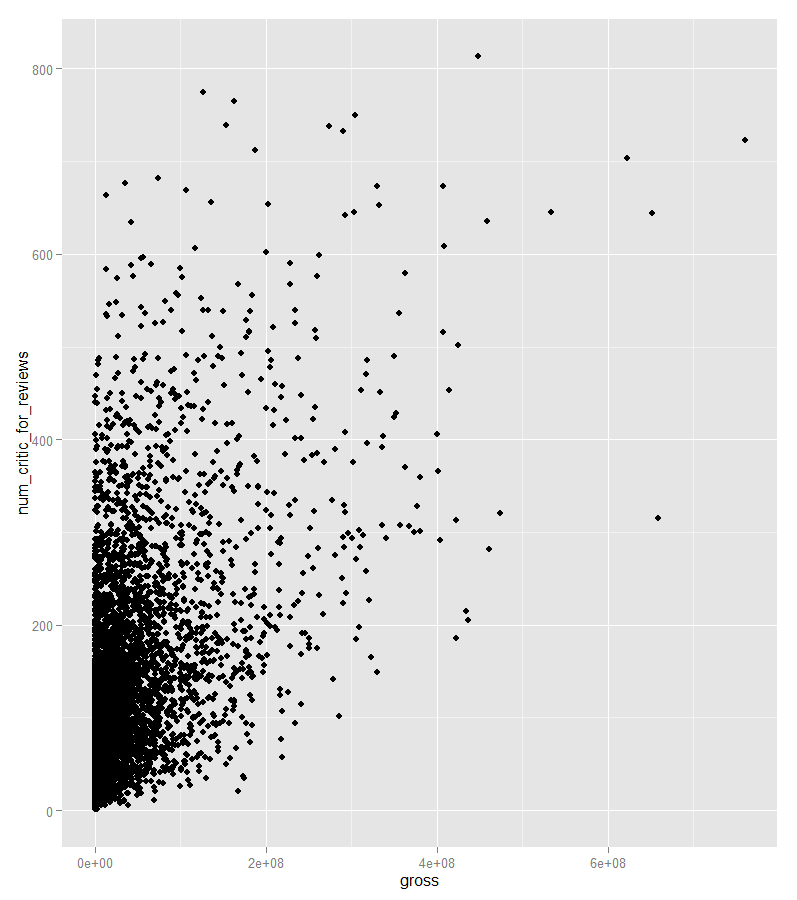
**“movie\_facebook\_likes” vs Gross**



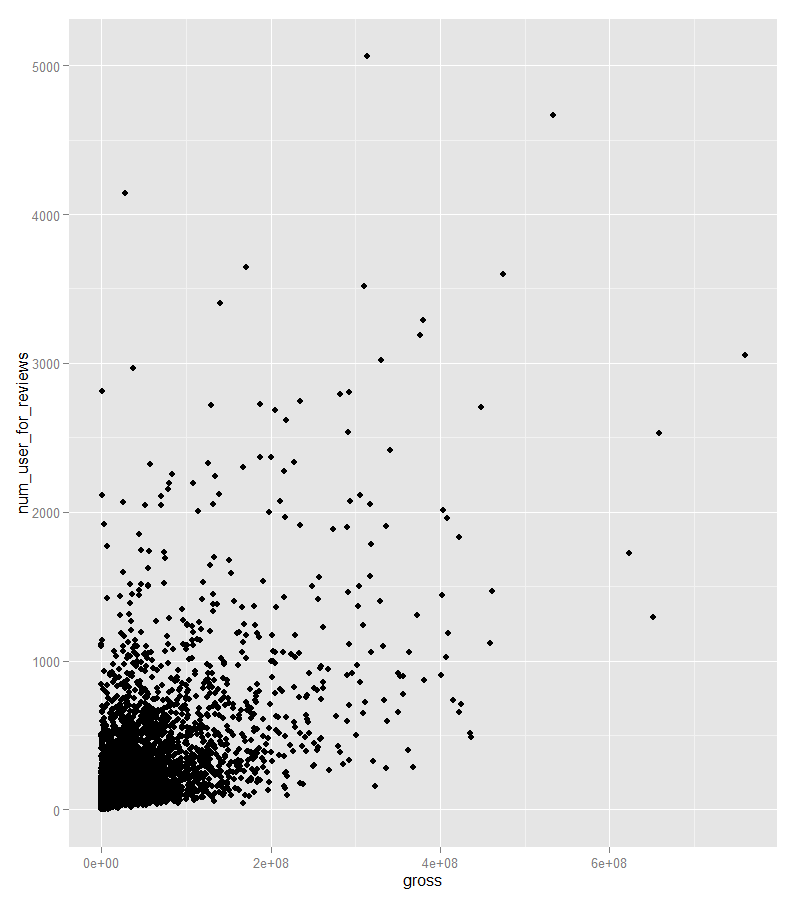
**“Duration” vs Gross**



**“No of Critics” vs Gross**



**“No of User reviews” vs Gross**



1. **Perform data cleaning and transformation wherever required. (Hint: You may need to make use of log transformation.)**

See section in R code bookmarked with question 2:-

In summary:

* Whenever suitable, assign numeric variables with empty values(NA) with 0 (e.g. Facebook like of the actors)
* Whenever suitable, assign non-numeric variables with empty values(NA) with ‘-‘ (e.g. actor name, content rating with NA set to ‘Not Rated’
* After that use ***complete.cases,*** *t*o remove those records with incomplete value (e.g. records with budget or gross)
* apply ***unique*** *to remove those duplicate rows*
* *Create additional measurements such as*
  + - * + *grossInMillions*
        + *grossInMillionsInlog*
        + *profitInMillions*
        + *profitInMillionsInlog*
        + *directorPastAvgGross (director other movie average gross - excluding the current movie gross)*
        + *top3ActorsPastAvgGross (top 3 actors other movie average - excluding the current movie gross)*

The end result produced a clean set called ***imdb\_clean*** for subsequent

sections.

1. **Transform genres, countries and movie languages into dichotomous variables.**

See section in R code bookmarked as question 3-

Break the genres column into multiple columns :-genres\_action, genres\_Adventure, genres\_Animation, genres\_Comedy…

Make sure language, country are a factor column.

It was decided later to break the country column into multiple columns:-fromUSA, fromUK, fromFrance, fromGermany.. The consideration was to make it easier to exclude/include certain country parameter from the regression model during model tuning.

1. **Formulate a suitable regression model for the scenario**

See section in R code bookmarked as question 4-

Two main models were built - one to predict gross and one to predict profit.

The gross model was chosen as it has a better performance in adjusted R square.

Below is the snippet of the code

testSetE <- c(1500:1600, 2500:2600, 3500:3600)

#testSetC

testSet <- testSetE

imdb\_training <- imdb\_budgetinMillions[-testSet,]

imdb\_test <- imdb\_budgetinMillions[testSet,]

PredictGrossModel <- lm((grossInMillionsInlog) ~

log(budgetInMillions) +

log(directorPastAvgGross)+

(top3ActorsPastAvgGross)

(cast\_total\_facebook\_likes) +

(duration) +

fromUSA+

fromUK+fromGermany+fromFrance+

IsEnglish +

+log(num\_critic\_for\_reviews)+log(num\_user\_for\_reviews)+

log(num\_voted\_users)+ genres\_action+genres\_Adventure+genres\_Animation+genres\_Comedy+

genres\_Drama+genres\_Family+genres\_Horror

+genres\_Fantasy+genres\_Crime++genres\_Mystery+

genres\_Romance+genres\_SciFi+genres\_Thriller

, data = imdb\_training)

———

predictionModel <- PredictGrossModel

predictionTesting <- predict(predictionModel, newdata = imdb\_test)

head(exp(predictionTesting))

head(imdb\_test$grossInMillions)

*n<-nrow(imdb\_test)*

*sumSquareError<-sum((imdb\_test$grossInMillions- exp(predictionTesting))^ 2)*

*meanSquareError<-sumSquareError/n*

*rootMeanSquareError<-sqrt(meanSquareError)*

*print(rootMeanSquareError)*

**The test set result of the prediction has rootMeanSquareError= 25.39803**

1. **Compute the coefficient of determination for the regression model. Comment on the adequacy of the model.**

—————

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -8.134e+00 2.470e-01 -32.925 < 2e-16 \*\*\*

log(budgetInMillions) 3.620e-01 2.766e-02 13.088 < 2e-16 \*\*\*

log(directorPastAvgGross) 1.523e-01 2.057e-02 7.401 1.70e-13 \*\*\*

cast\_total\_facebook\_likes -2.732e-06 1.157e-06 -2.360 0.01831 \*

duration 4.881e-04 1.202e-03 0.406 0.68477

fromUSA 9.900e-01 9.682e-02 10.225 < 2e-16 \*\*\*

fromUK 4.941e-01 1.188e-01 4.158 3.30e-05 \*\*\*

fromGermany 1.664e-01 1.678e-01 0.992 0.32145

fromFrance 3.575e-01 1.563e-01 2.287 0.02226 \*

IsEnglish 1.248e+00 1.302e-01 9.583 < 2e-16 \*\*\*

log(num\_critic\_for\_reviews) -3.089e-01 4.409e-02 -7.006 2.96e-12 \*\*\*

log(num\_user\_for\_reviews) 2.505e-01 4.685e-02 5.346 9.58e-08 \*\*\*

log(num\_voted\_users) 7.033e-01 3.657e-02 19.232 < 2e-16 \*\*\*

genres\_action1 4.801e-02 6.261e-02 0.767 0.44323

genres\_Adventure1 -1.190e-02 6.610e-02 -0.180 0.85711

genres\_Animation1 -1.574e-01 1.203e-01 -1.308 0.19081

genres\_Comedy1 1.442e-01 5.859e-02 2.462 0.01388 \*

genres\_Drama1 -2.244e-01 5.613e-02 -3.998 6.53e-05 \*\*\*

genres\_Family1 7.857e-01 8.956e-02 8.773 < 2e-16 \*\*\*

genres\_Horror1 2.450e-01 8.799e-02 2.784 0.00540 \*\*

genres\_Fantasy1 -2.040e-01 7.019e-02 -2.907 0.00368 \*\*

genres\_Crime1 -3.437e-01 6.469e-02 -5.313 1.15e-07 \*\*\*

genres\_Mystery1 -6.673e-03 7.804e-02 -0.086 0.93186

genres\_Romance1 4.951e-02 5.691e-02 0.870 0.38433

genres\_SciFi1 -3.782e-01 7.117e-02 -5.315 1.14e-07 \*\*\*

genres\_Thriller1 4.469e-02 6.126e-02 0.729 0.46577

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.257 on 3299 degrees of freedom

Multiple R-squared: 0.6038, Adjusted R-squared: 0.6008

F-statistic: 201.1 on 25 and 3299 DF, p-value: < 2.2e-16

Predictors budgetInMillions, directorPastAvgGross, fromUSA (Country), from UK (Country), IsEnglish (language), num\_critic\_for\_reviews, num\_user\_for\_reviews, num\_voted\_users and some of the genres as shown in above shows they have high significance in confidence (\*\*\*) as predictors for the regression model.

The p-value of < 2.2e-16 indicates the null hypothesis that this model is invalid cannot be accepted, thus we could use the model to predict. The question is how adequate the model as predictor.

The adjusted R-squared of 0.6008 indicates the regression model has relative high error in its prediction result. That implied the problem nature itself is inherently tough. to predict with great accuracy (at least with the attempted linear regression model).

That conforms to our common sense that the movie gross is inherently hard to predict.

The RMS error of the test result is 25.39803 (gross measured in millions).

That implied the model is more suitable for predicting big budget movie with expectation of big grossing (more than 10 millions let say) as opposed to movie with low budget expecting low grossing.

1. **Construct a 95% confidence interval on movie duration based on the regression model you have built. Comment on this confidence interval.**

confint(predictionModel, 'duration', level=0.95)

2.5 % 97.5 %

duration -0.002011707 0.002707584

> exp(-0.001869194)

[1] 0.9981326

> exp(0.00284546)

[1] 1.00285

For every increase in duration, the gross is estimated to increase between

0.9981326 and 1.00285 (in millions)

1. **Is the regression model that you have built in the previous question significant and why?**

The p-value of < 2.2e-16 indicates the null hypothesis that this model is invalid cannot be accepted, thus the model is significant.

1. **Step through your model and determine if there is any multicollinearity in the predictors and adjust your model accordingly.**

budgetInMillions directorPastAvgGross cast\_total\_facebook\_likes duration num\_critic\_for\_reviews num\_user\_for\_reviews num\_voted\_users

budgetInMillions 1.00 0.19 0.03 0.07 0.11 0.07 0.07

directorPastAvgGross 0.19 1.00 0.12 0.16 0.22 0.26 0.29

cast\_total\_facebook\_likes 0.03 0.12 1.00 0.12 0.24 0.18 0.25

duration 0.07 0.16 0.12 1.00 0.23 0.35 0.34

num\_critic\_for\_reviews 0.11 0.22 0.24 0.23 1.00 **0.57** **0.60**

num\_user\_for\_reviews 0.07 0.26 0.18 0.35 0.57 1.00 0.78

num\_voted\_users 0.07 0.29 0.25 0.34 0.60 **0.78** 1.00

num\_user\_for\_reviews and num\_voted\_users having 0.78 correlation

num\_voted\_users and num\_critic\_for\_reviews having 0.60 correlation

follows by btw num\_user\_for\_reviews and num\_critic\_for\_reviews having 0.57 correlation

One of the predictor (e.g. num\_user\_for\_reviews) can be dropped.

1. **What is the expected gross earnings of a movie which has the following attributes?**

<See section in R code bookmarked as question 9>

534 millions.

The similar model was reused back to predict. As the question didn’t have all the necessary input parameters (*budgetInMillions, cast\_total\_facebook\_likes, directorPastAvgGross*), their values were derived by taking average computation from the training set data, which are resembling the attribute given (see the SQL condition).

avgBudget avgCastTotalFacebookLikes avgdirectorPastAvgGross count(1)

1 127.9929 22114.59 117.0424 183

As the prediction value actually is depends on the estimated average value, another model is built without parameters of *budgetInMillions, cast\_total\_facebook\_likes, directorPastAvgGross.*

*It gave result of 467 millions. Close to earlier prediction*

1. **Suppose a sequel was produced as the movie was well received by many viewers. What is the expected gross earnings of the sequel which has the following attributes?**

<See section in R code bookmarked as question 10>

Expected gross= **1237.036 million**s

which is an increase of 703 millions from 534 millions. A blockbuster!

The prediction was done by adjusting the input parameters to the given input values, for example, the predicted budget (from average estimate) was set to increase by 50%. Input values which didn't match the input parameters were ignored (e.g Actor facebook like) as they weren't significants to this model.

———————try using budget increase alone to predict ————

50% increase in movie production budget as compared to the prequel.

The earlier budget was 128 Millions (assume).

50% increases amounted to 64 Millions.

Gross increases by 1.36 X 64 = 85.76 millions

1.51 X 64 = 96.64 millions

> confint(Qns9predictionModel, 'log(budgetInMillions)', level=0.95)

2.5 % 97.5 %

log(budgetInMillions) 0.3077714 0.416236

> exp(0.3077714)

[1] 1.36039

> exp(0.416236)

[1] 1.516244

——————