Project 1: Explore and Prepare Data

CSE6242 - Data and Visual Analytics - Spring 2017 Due: Sunday, March 5, 2017 at 11:59 PM UTC-12:00 on T-Square Completed by Frank Hahn, gtid: fhahn3

Note: This project involves getting data ready for analysis and doing some preliminary investigations. Project 2 will involve modeling and predictions, and will be released at a later date. Both projects will have equal weightage towards your grade.

Data

In this project, you will explore a dataset that contains information about movies, including ratings, budget, gross revenue and other attributes. It was prepared by Dr. Guy Lebanon, and here is his description of the dataset:

The file movies_merged contains a dataframe with the same name that has 40K rows and 39 columns. Each row represents a movie title and each column represents a descriptor such as Title, Actors, and Budget. I collected the data by querying IMDb's API (see www.omdbapi.com) and joining it with a separate dataset of movie budgets and gross earnings (unknown to you). The join key was the movie title. This data is available for personal use, but IMDb's terms of service do not allow it to be used for commercial purposes or for creating a competing repository.

Objective

Your goal is to investigate the relationship between the movie descriptors and the box office success of movies, as represented by the variable Gross. This task is extremely important as it can help a studio decide which titles to fund for production, how much to bid on produced movies, when to release a title, how much to invest in marketing and PR, etc. This information is most useful before a title is released, but it is still very valuable after the movie is already released to the public (for example it can affect additional marketing spend or how much a studio should negotiate with on-demand streaming companies for "second window" streaming rights).

Instructions

This is an R Markdown Notebook. Open this file in RStudio to get started.

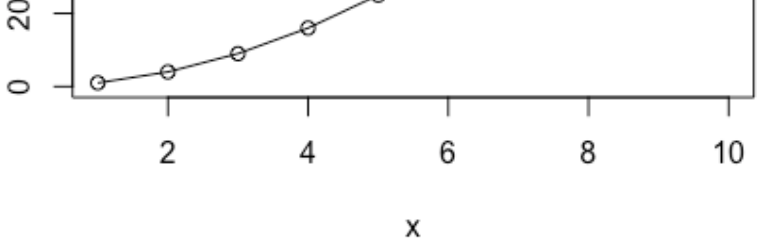
When you execute code within the notebook, the results appear beneath the code. Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
x = 1:10
print(x^2)
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

Plots appear inline too:

```
plot(x, x^2, 'o')
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

Please complete the tasks below and submit this R Markdown file (as **pr1.Rmd**) as well as a PDF export of it (as **pr1.pdf**). Both should contain all the code, output, plots and written responses for each task.

Setup

Load data

Make sure you've downloaded the movies_merged file and it is in the current working directory. Now load it into memory:

```
load('movies_merged')
```

This creates an object of the same name (movies_merged). For convenience, you can copy it to df and start using it:

```
df = movies_merged
cat("Dataset has", dim(df)[1], "rows and", dim(df)[2], "columns", end="\n", file="")
## Dataset has 40789 rows and 39 columns
colnames(df)
##  [1] "Title"            "Year"             "Rated"
##  [4] "Released"         "Runtime"          "Genre"
##  [7] "Director"         "Writer"           "Actors"
## [10] "Plot"             "Language"         "Country"
## [13] "Awards"           "Poster"           "Metascore"
## [16] "imdbRating"       "imdbVotes"        "imdbID"
## [19] "Type"             "tomatoMeter"      "tomatoImage"
## [22] "tomatoRating"     "tomatoReviews"    "tomatoFresh"
## [25] "tomatoRotten"     "tomatoConsensus"  "tomatoUserMeter"
## [28] "tomatoUserRating" "tomatoUserReviews" "tomatoURL"
## [31] "DVD"              "BoxOffice"        "Production"
## [34] "Website"          "Response"         "Budget"
## [37] "Domestic_Gross"   "Gross"            "Date"
```

Load R packages

Load any R packages that you will need to use. You can come back to this chunk, edit it and re-run to load any additional packages later.

```
library(ggplot2)
library(GGally)
library(tm)
## Loading required package: NLP
##
## Attaching package: 'NLP'
## The following object is masked from 'package:ggplot2':
##
##     annotate
library(psych)
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

If you are loading any non-standard packages (ones that have not been discussed in class or explicitly allowed for this project), please mention them below. Include any special instructions if they cannot be installed using the regular install.packages('<pkg name>') command.

command.

Non-standard packages used: None

Tasks

Each task below is worth **10** points, and is meant to be performed sequentially, i.e. do step 2 after you have processed the data as described in step 1. Total points: **100**

Complete each task by implementing code chunks as described by TODO comments, and by responding to questions ("**Q**:") with written answers ("**A**:"). If you are unable to find a meaningful or strong relationship in any of the cases when requested, explain why not by referring to appropriate plots/statistics.

It is OK to handle missing values below by omission, but please omit as little as possible. It is worthwhile to invest in reusable and clear code as you may need to use it or modify it in project 2.

1. Remove non-movie rows

The variable Type captures whether the row is a movie, a TV series, or a game. Remove all rows from df that do not correspond to movies.

```
# TODO: Remove all rows from df that do not correspond to movies
df <-subset(df, df$Type=="movie")
# df[,"Type"]=="movie"
```

Q: How many rows are left after removal? *Enter your response below.*

A: 40,000 rows

2. Process Runtime column

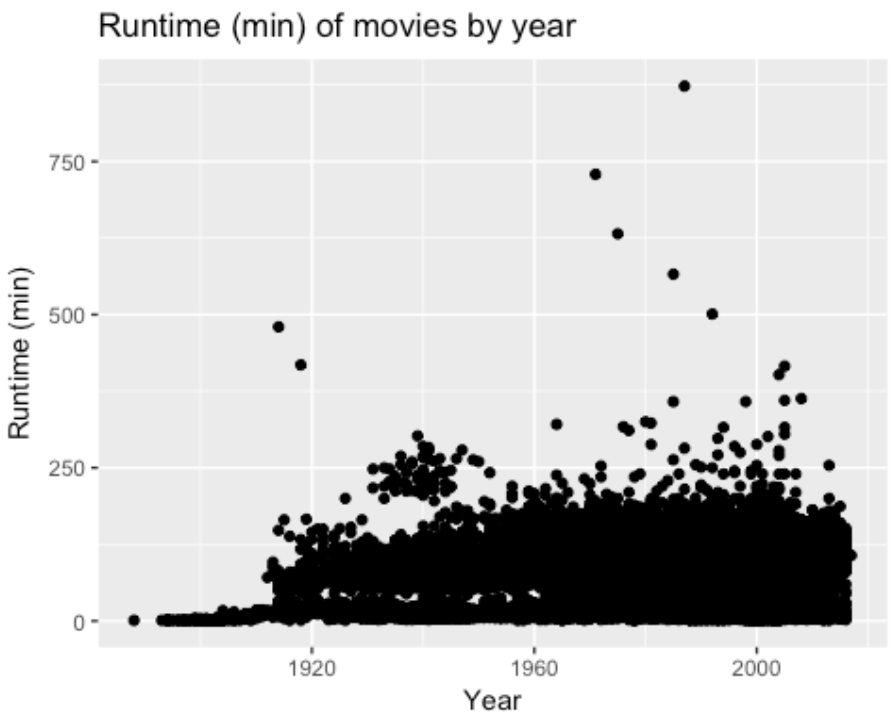
The variable Runtime represents the length of the title as a string. Write R code to convert it to a numeric value (in minutes) and replace df$Runtime with the new numeric column.

```
# TODO: Replace df$Runtime with a numeric column containing the runtime in minutes
# copyRuntime <- df$Runtime
# df$Runtime=0
# typeof(df$Runtime[1])

# library(readr)
# test <- parse_guess(df$Runtime)
converttoMin <- function(movietime){
#if contains h then convert to minutes
if (grepl("h", movietime)) {
# cat("Contains h")
timelist =strsplit(movietime, " ")
hour = strtoi (timelist[[1]][1])
min = strtoi (timelist[[1]][3])
min = hour*60+min
#contains min, strip to the front, capture number
} else if (grepl("min", movietime)) {
# cat("contains just min")
timelist =strsplit(movietime, " ")
min = strtoi (timelist[[1]][1])
#else "N/A"
} else {
# cat("N/A")
min="N/A"
}
return(min)
}
# converttoMin(movietime)

suppressWarnings(df$Runtime <- lapply(df$Runtime, converttoMin ))
suppressWarnings(df$Runtime <- as.numeric(df$Runtime))
```

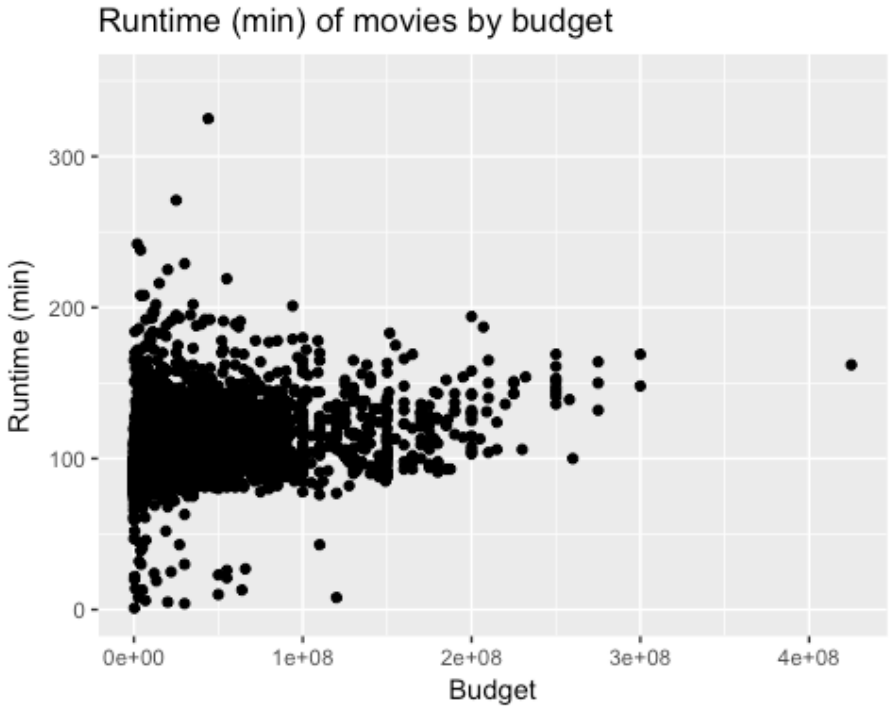
Now investigate the distribution of Runtime values and how it changes over years (variable Year, which you can bucket into decades) and in relation to the budget (variable Budget). Include any plots that illustrate.

```
# TODO: Investigate the distribution of Runtime values and how it varies by Year and Budget

df_noNA <- df[df$Runtime!="N/A",]
```

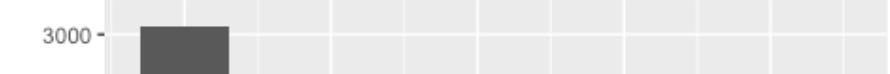
```
ggplot(df_noNA, aes(df_noNA$Year, df_noNA$Runtime))+
 geom_point()+
 ggtitle("Runtime (min) of movies by year")+
 ylab("Runtime (min)")+
 xlab("Year")
```
Warning: Removed 759 rows containing missing values (geom_point).

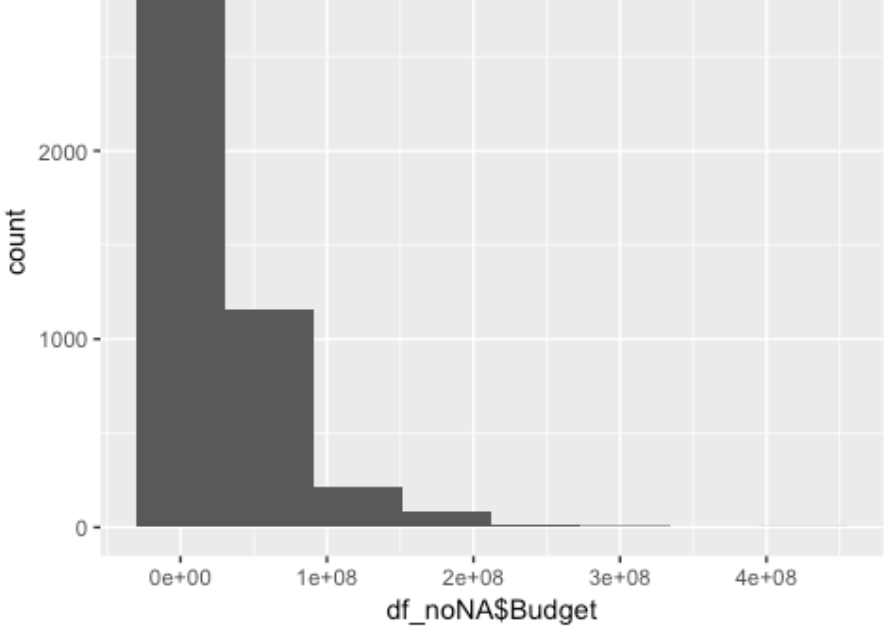



```
ggplot(df_noNA, aes(df_noNA$Budget, df_noNA$Runtime))+
 geom_point() +
 ggtitle("Runtime (min) of movies by budget")+
 ylab("Runtime (min)")+
 xlab("Budget")+
 ylim(0,350)
```
Warning: Removed 35480 rows containing missing values (geom_point).



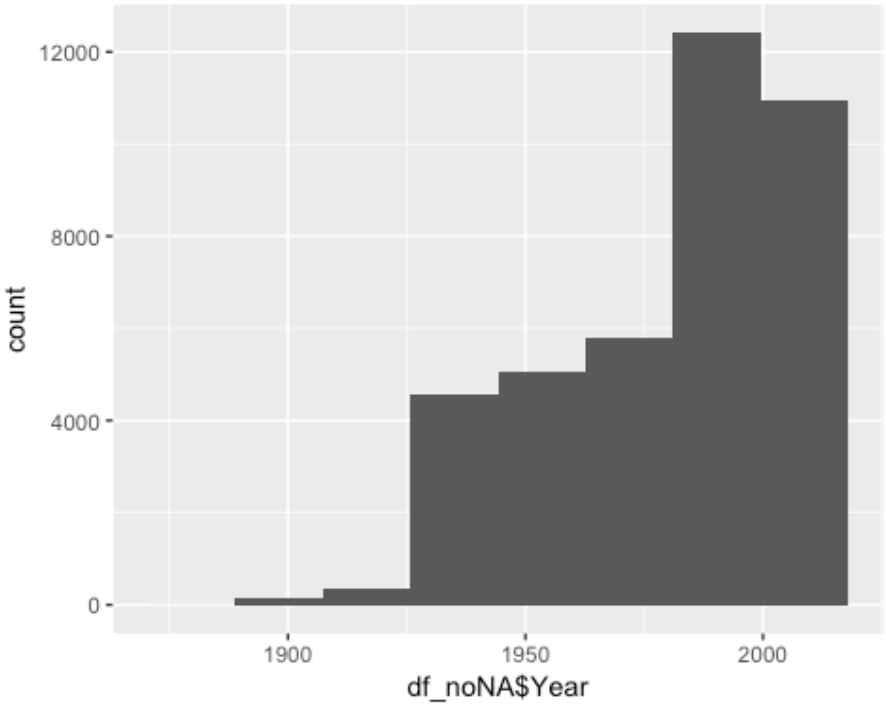

```
# coord_cartesian(ylim = c(0,300))
ggplot(df_noNA, aes(df_noNA$Budget))+
 geom_histogram(bins=8)
```
Warning: Removed 35480 rows containing non-finite values (stat_bin).





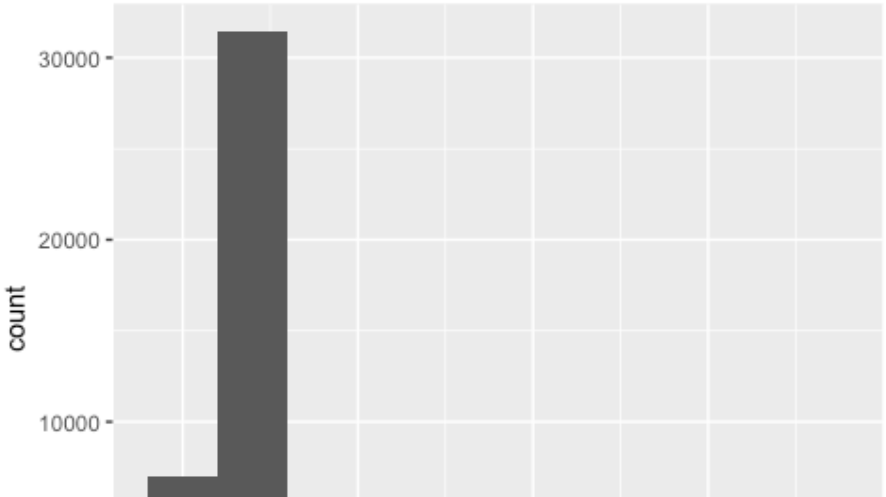

```
ggplot(df_noNA, aes(df_noNA$Year))+
 geom_histogram(bins=8)
```

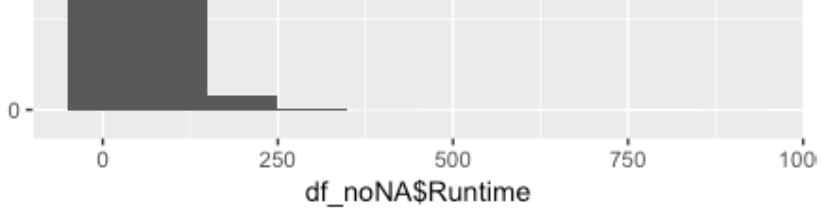
Warning: Removed 759 rows containing non-finite values (stat_bin).




```
ggplot(df_noNA, aes(df_noNA$Runtime))+
 geom_histogram(binwidth = 100)
```

Warning: Removed 759 rows containing non-finite values (stat_bin).





Feel free to insert additional code chunks as necessary.

Q: Comment on the distribution as well as relationships. Are there any patterns or trends that you can observe?

A: Prior to 1920 Movies were very short; however, around that time movies started to become longer to the 60-80 min in length. After 1920, movies progressively became longer by seemed to stabilize at about 200 minutes in length.

Budget does not appear to impact length of movie. The median of length of movie appears to be around 120 minutes regardless of budget of movie.

3. Encode Genre column

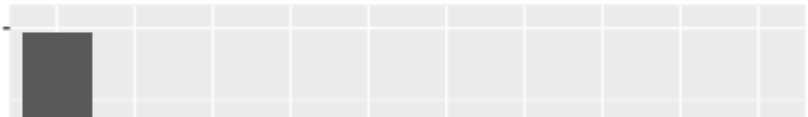
The column Genre represents a list of genres associated with the movie in a string format. Write code to parse each text string into a binary vector with 1s representing the presence of a genre and 0s the absence, and add it to the dataframe as additional columns. Then remove the original Genre column.

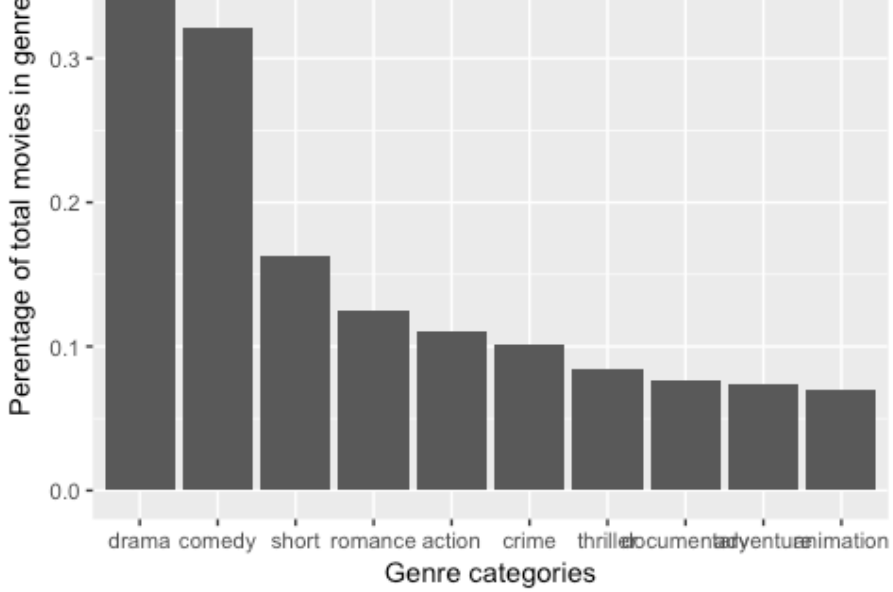
For example, if there are a total of 3 genres: Drama, Comedy, and Action, a movie that is both Action and Comedy should be represented by a binary vector <0, 1, 1>. Note that you need to first compile a dictionary of all possible genres and then figure out which movie has which genres (you can use the R tm package to create the dictionary).

```
# TODO: Replace Genre with a collection of binary columns
df$Genre <- gsub("[][!#$%()*,.:;<=>@^_|~.{}]", "", df$Genre)
df$Genre <- tolower(df$Genre)

Genre_cats <-Corpus(VectorSource(df$Genre))
Genre_dict <- DocumentTermMatrix(Genre_cats)

Genre_categories <- findFreqTerms(Genre_dict, 1)

df[Genre_categories]<-NA

for (genre in Genre_categories){
# cat(genre)
df[[genre]][grepl(genre, df$Genre)]=1
df[[genre]][!grepl(genre, df$Genre)]=0
}

# df$action[grepl("action", df$Genre)]=1
# df$action[!grepl("action", df$Genre)]=0
```

Plot the relative proportions of movies having the top 10 most common genres.

```
# TODO: Select movies from top 10 most common genres and plot their relative proportions
genre_counts <- colSums(df[Genre_categories])
genre_df <- data.frame(Genre_categories, genre_counts)
genre_df$proportion <- genre_counts/40000


top_genre <-genre_df[order(-genre_df$proportion),][1:10,]

# top_genre <-genre_df[genre_df$proportion>.068,]
# top_genre<- transform(top_genre, top_genre=reorder(top_genre$proportion,
top_genre$proportion))

ggplot(top_genre, aes(x= reorder(Genre_categories, -proportion), y= proportion))+
 geom_bar(stat="identity")+
 xlab ("Genre categories")+
 ylab("Perentage of total movies in genre")
```

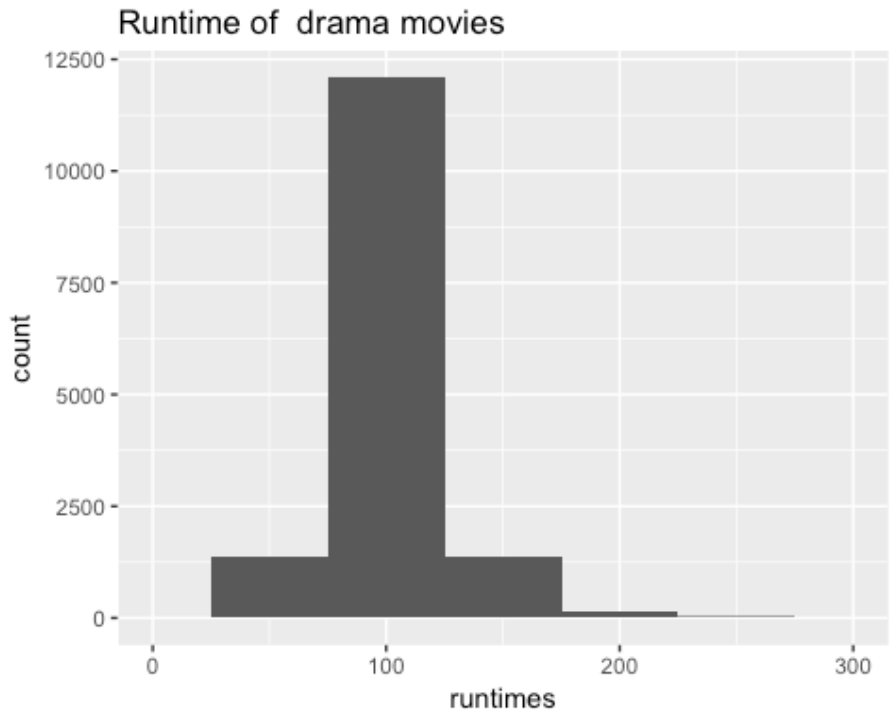




Examine how the distribution of Runtime changes across genres for the top 10 most common genres.

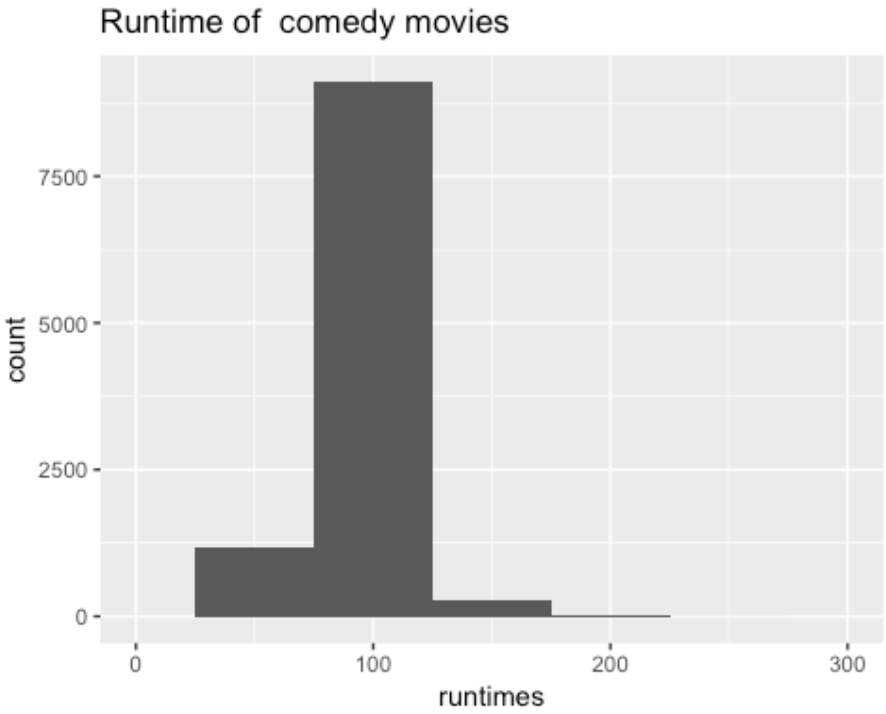
```
# TODO: Plot Runtime distribution for top 10 most common genres
top_genre_vector <- top_genre$Genre_categories
for (x in top_genre_vector){
cat(x, "\n")
typeof(x)
mygenre=x
isgenre <-df[[mygenre]]==1

runtimes <- df[isgenre,]$Runtime
print(describe(runtimes))
print(qplot(runtimes, binwidth=50)+ggtitle(paste("Runtime of ",x, "movies")) + xlim(0,300))

}
```

```
## drama
##    vars     n  mean    sd median trimmed   mad min max range skew kurtosis
## X1    1 15686 96.94 30.06     96   97.44 16.31   1 729   728 1.05    23.47
##      se
## X1 0.24
## Warning: Removed 180 rows containing non-finite values (stat_bin).
```

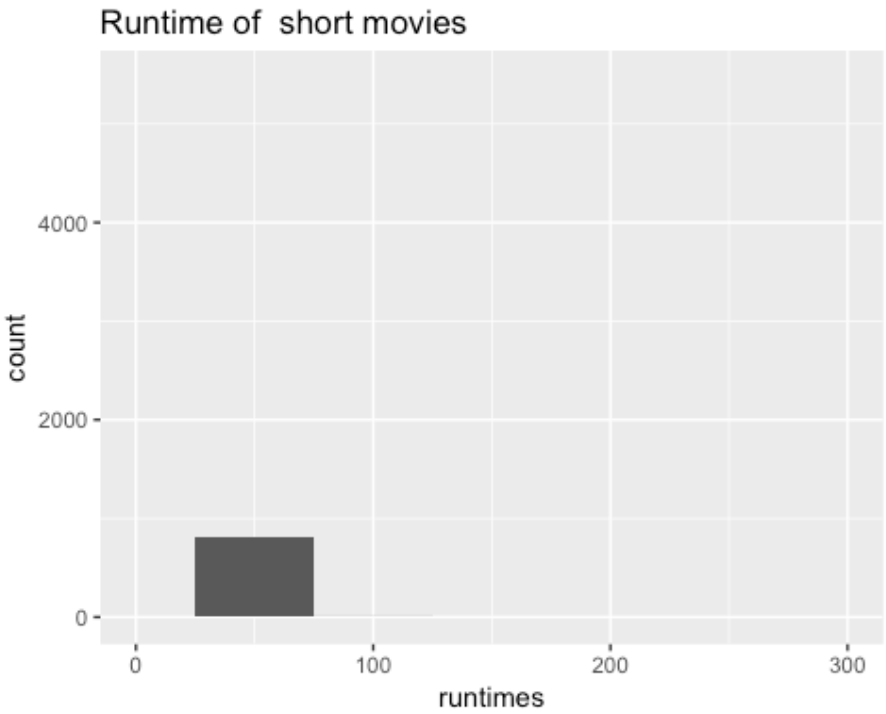



```
## comedy
##    vars     n  mean    sd median trimmed   mad min max range  skew
## X1    1 12733 79.11 35.02     90   82.63 16.31   1 416   415 -0.84
##    kurtosis   se
```

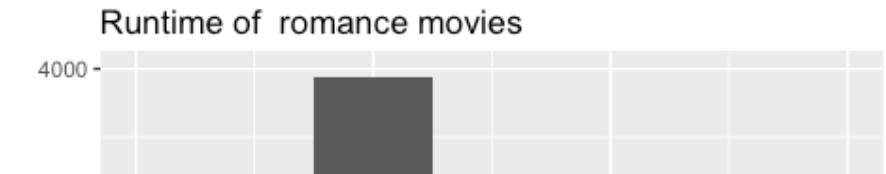
X1 1.08 0.31
Warning: Removed 117 rows containing non-finite values (stat_bin).

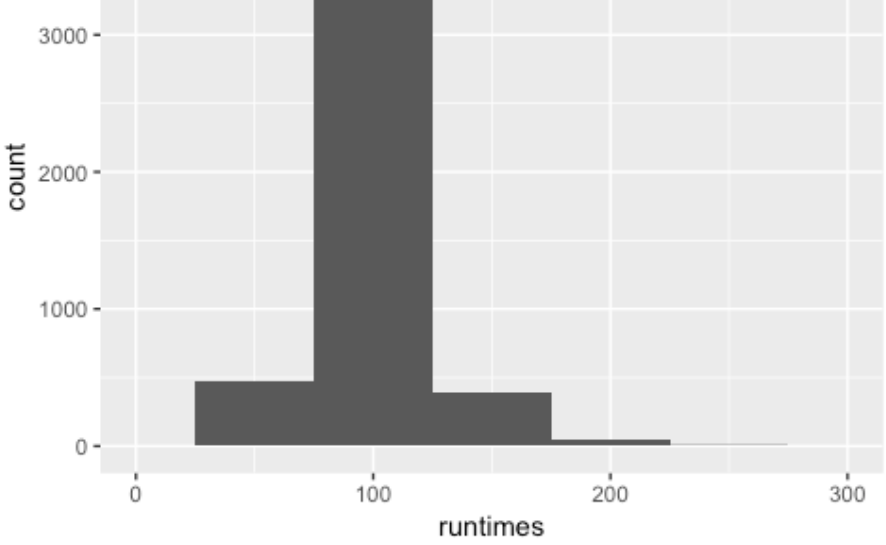


short
vars n mean sd median trimmed mad min max range skew kurtosis
X1 1 6290 13.92 9.47 11 12.7 7.41 1 90 89 1.25 1.89
se
X1 0.12
Warning: Removed 226 rows containing non-finite values (stat_bin).

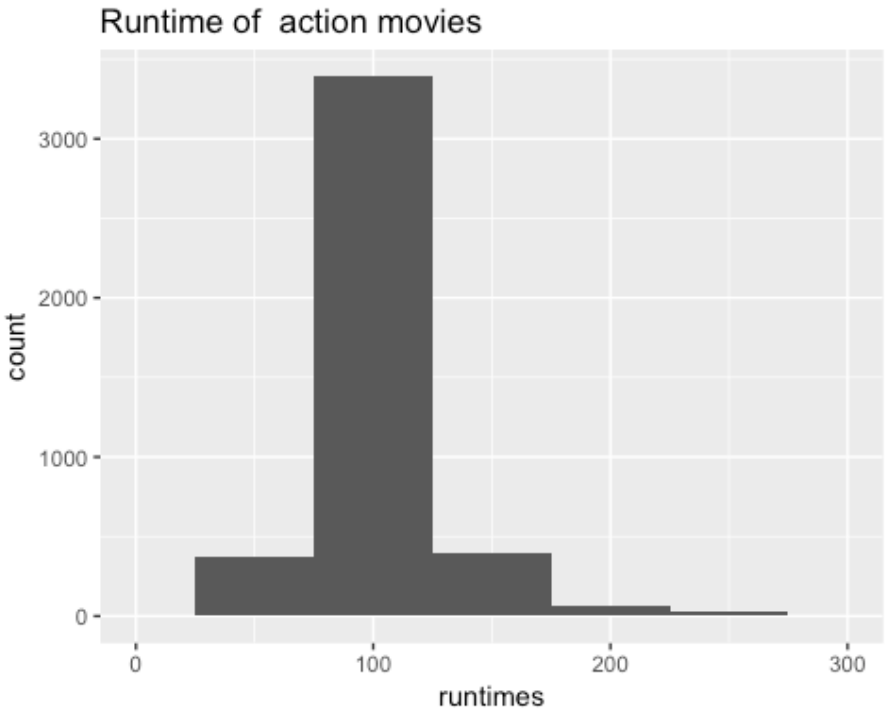


romance
vars n mean sd median trimmed mad min max range skew kurtosis
X1 1 4937 98.07 26.24 96 97.22 14.83 1 632 631 2.1 38.44
se
X1 0.37
Warning: Removed 39 rows containing non-finite values (stat_bin).

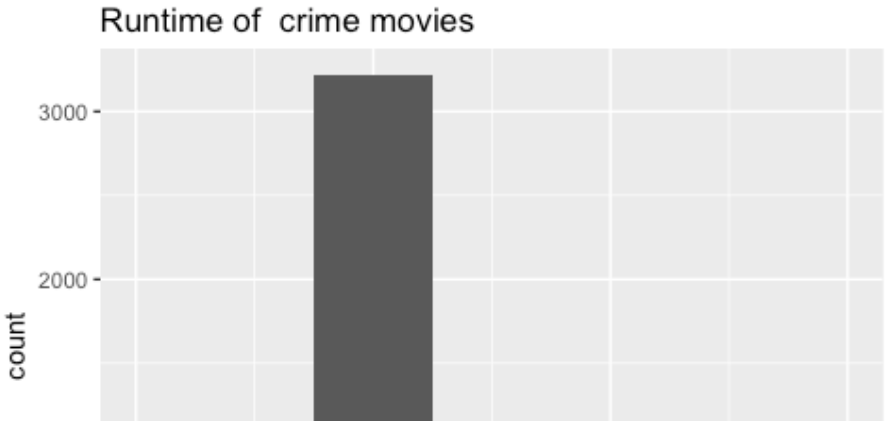


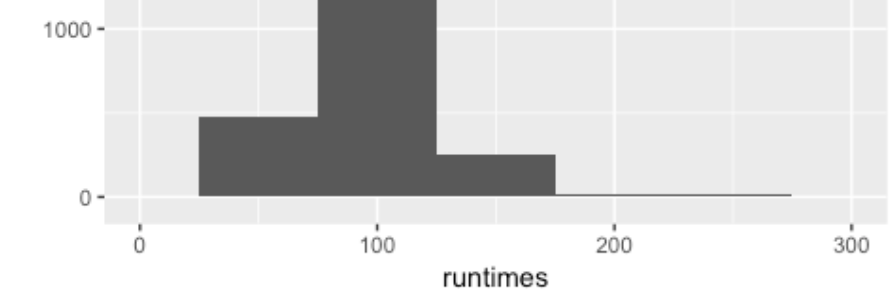


action
vars n mean sd median trimmed mad min max range skew kurtosis
X1 1 4366 98.74 30.13 95 97.04 13.34 1 302 301 1.16 6.88
se
X1 0.46
Warning: Removed 48 rows containing non-finite values (stat_bin).

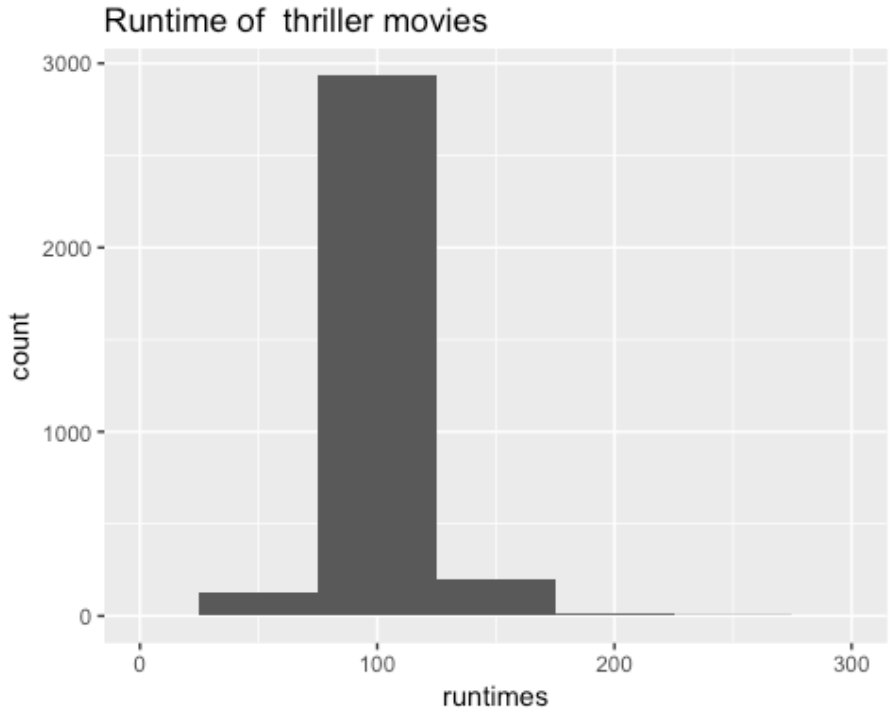


crime
vars n mean sd median trimmed mad min max range skew kurtosis
X1 1 4043 95.85 24.52 95 95.18 14.83 3 302 299 0.9 8.16
se
X1 0.39
Warning: Removed 20 rows containing non-finite values (stat_bin).

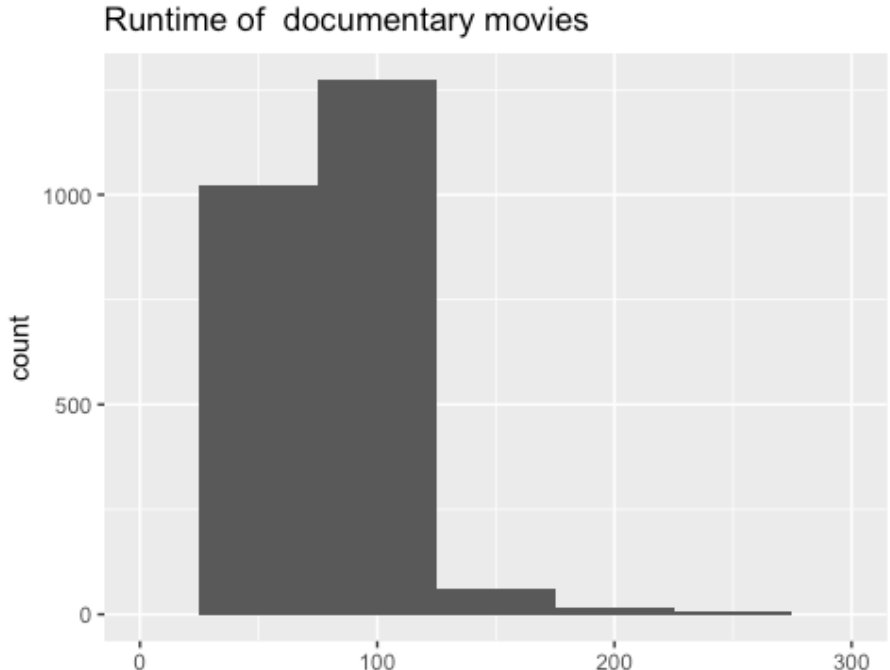




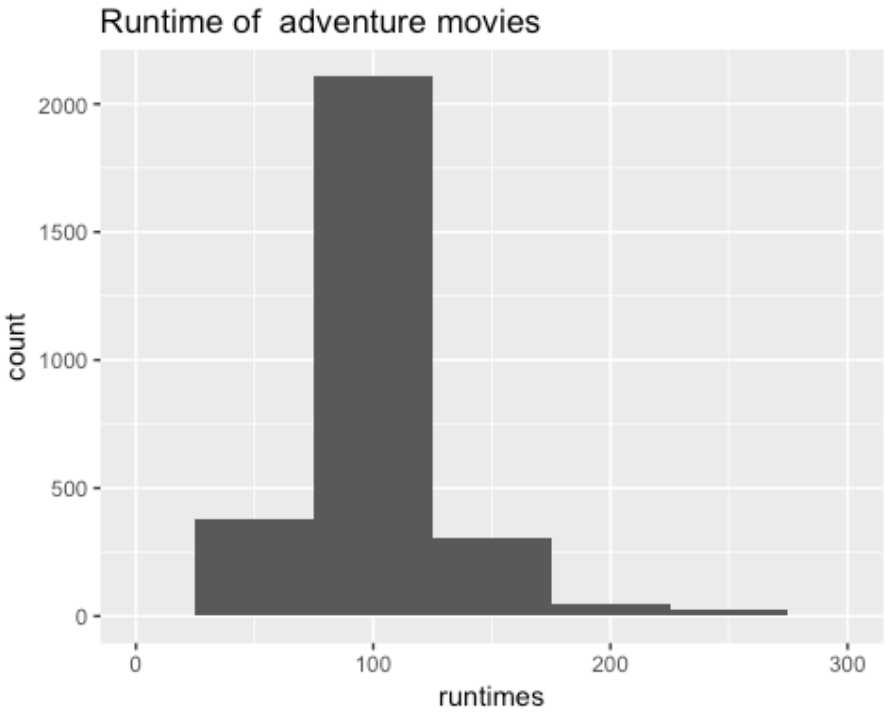
thriller
vars n mean sd median trimmed mad min max range skew kurtosis
X1 1 3346 96.37 21.23 95 96.17 10.38 3 253 250 -0.37 6.98
se
X1 0.37
Warning: Removed 34 rows containing non-finite values (stat_bin).



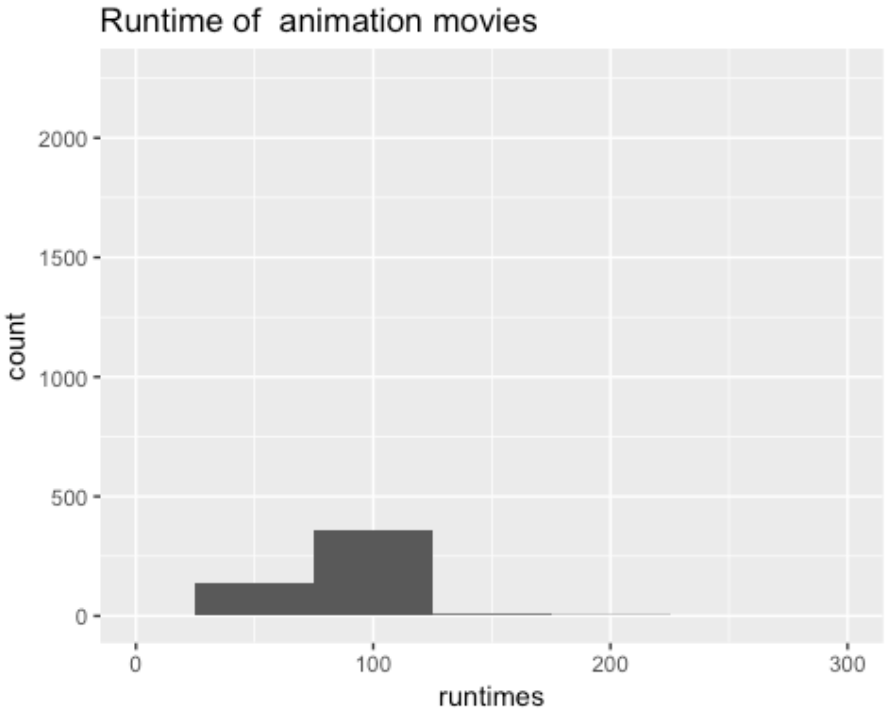
documentary
vars n mean sd median trimmed mad min max range skew kurtosis
X1 1 2927 67.78 43.31 73 66.58 31.13 1 873 872 3.82 53.38
se
X1 0.8
Warning: Removed 132 rows containing non-finite values (stat_bin).




```
## adventure
##    vars    n  mean    sd median trimmed   mad min max range skew kurtosis
## X1    1 2910 99.11 31.77     95   96.56 17.79   3 418   415 1.83    10.23
##      se
## X1 0.59
## Warning: Removed 22 rows containing non-finite values (stat_bin).
```




```
## animation
##    vars    n  mean    sd median trimmed  mad min max range skew kurtosis
## X1    1 2767 20.89 30.24      7   13.82 1.48   1 200   199 1.94     2.54
##      se
## X1 0.57
## Warning: Removed 21 rows containing non-finite values (stat_bin).
```




```
# plot <-df[df$drama==1,]
# ggplot(df, aes(Genre, Runtime))+
#   geom_boxplot()
```

Q: Describe the interesting relationship(s) you observe. Are there any expected or unexpected trends that are evident?

A: First that was clear that Short movies were short with a median of 11 minutes. However, because the range of short movies went into the 90s is suggest that maybe the data is corrupted.

Drama movies, the most common genre with ~40% of movies in the dataset, are most often about 100 minutes in length and has a median of runtime of 96 minutes. This is in line with movies of most of the top categories except short (11 min), animation (7 min), and documentary (73 min). I hypothesize that because these three categories are less mainstream entertainment genres and therefore are different types of movies; whereas, the other categories are more like movie theater type of genres.

4. Eliminate mismatched rows

The dataframe was put together by merging two different sources of data and it is possible that the merging process was inaccurate in some cases (the merge was done based on movie title, but there are cases of different movies with the same title). The first source's release time was represented by the column Year (numeric representation of the year) and the second by the column Released (string representation of release date).

Find and remove all rows where you suspect a merge error occurred based on a mismatch between these two variables. To make sure subsequent analysis and modeling work well, avoid removing more than 10% of the rows that have a Gross value present.

```
# TODO: Remove rows with Released-Year mismatch

# convert
df$Released_year<- as.numeric(format(df$Released,'%Y'))

#need to compare Year and Released_year
sum(is.na(df$Released)&!is.na(df$Gross))
## [1] 45
sum(df$Year==df$Released_year, na.rm=TRUE)
## [1] 29324
# df[is.na(df$Released)&!is.na(df$Gross),]
mismatchexpression <- (abs(df$Year-df$Released_year)<2)
# df$Gross[!is.na(df$Gross)]

df_mismatch <- subset(df, mismatchexpression)

DFwithgross <-sum(!is.na(df$Gross))
DFMMwithgross<- sum(!is.na(df_mismatch$Gross))
deleted <- (DFwithgross-DFMMwithgross)/DFwithgross*100
cat("I deleted ", deleted,"% of the rows with gross")
## I deleted  2.983765 % of the rows with gross
df <- df_mismatch
```

Q: What is your precise removal logic and how many rows did you end up removing?

A: After reviewing some movies e.g., Calling Hedy Lamarr
http://www.imdb.com/title/tt0419624/

although the years maybe not the same they are still correct. It looks like this scenario is that the Year refers to the date it was first shown at a festival and Released is when it was publicly released. Therefore, a strict interpretation of the years having to be equal may be too restrictive.

Therefore, my logic was that if the absolute value of difference between Year and Released was >2 then I assumed it was a bad mismatch and filtered it.

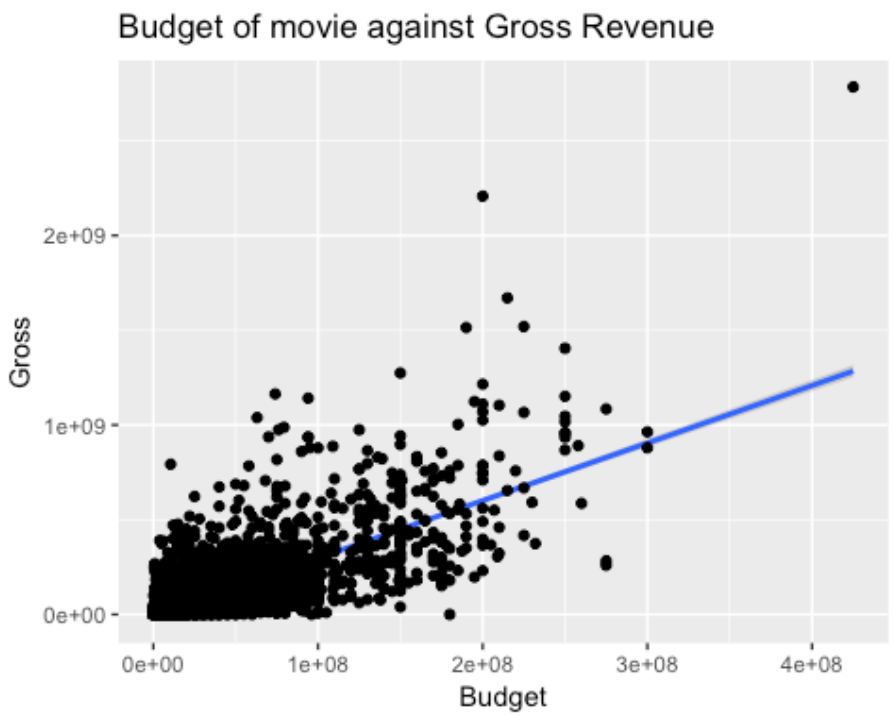
I ended up removing 6768 rows but only <3% of movies with Gross data.

5. Explore Gross revenue

For the commercial success of a movie, production houses want to maximize Gross revenue. Investigate if Gross revenue is related to Budget, Runtime or Genre in any way.

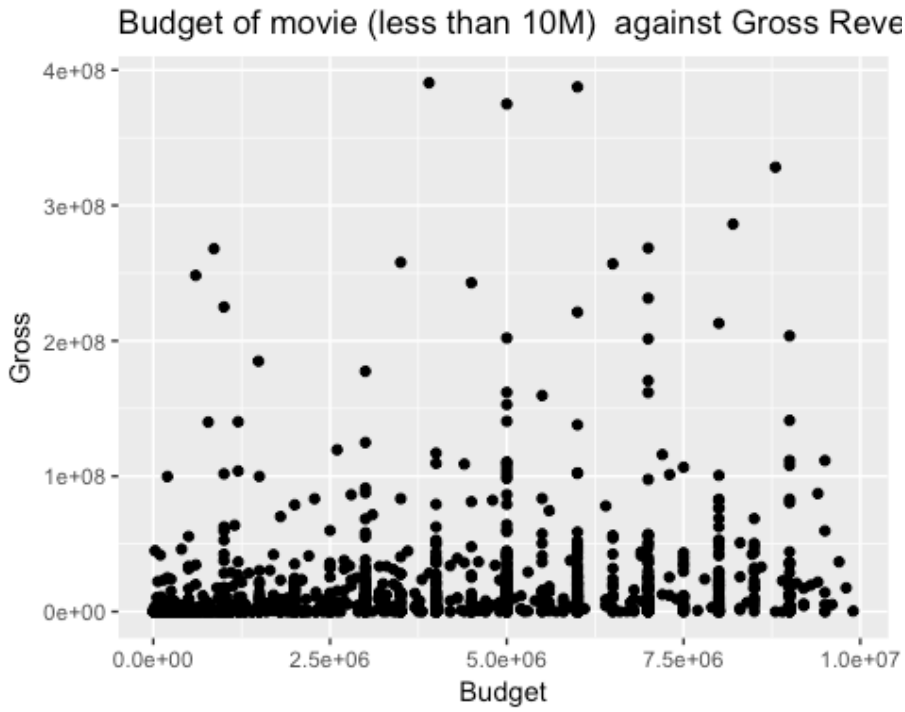
Note: To get a meaningful relationship, you may have to partition the movies into subsets such as short vs. long duration, or by genre, etc.

```
# TODO: Investigate if Gross Revenue is related to Budget, Runtime or Genre

p_wBudget<- ggplot(df, aes(Budget, Gross))
p_wBudget+geom_smooth(method = "gam") + geom_point() +
 ggtitle("Budget of movie against Gross Revenue")
## Warning: Removed 28810 rows containing non-finite values (stat_smooth).
## Warning: Removed 28810 rows containing missing values (geom_point).
```



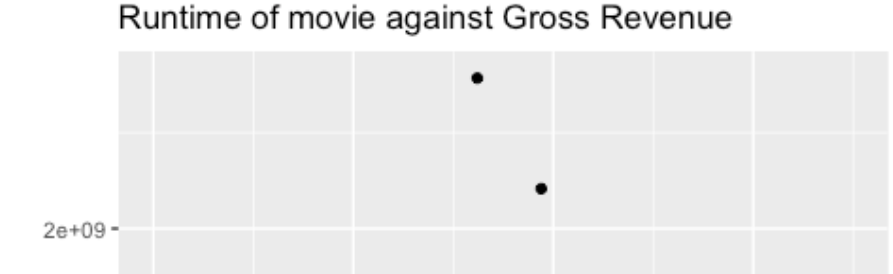

```
df_gross_less10M <- df[df$Budget<10000000,]
p_budget10m <- ggplot(df_gross_less10M, aes(Budget, Gross))
p_budget10m+ geom_point() +
 ggtitle("Budget of movie (less than 10M)  against Gross Revenue")
```

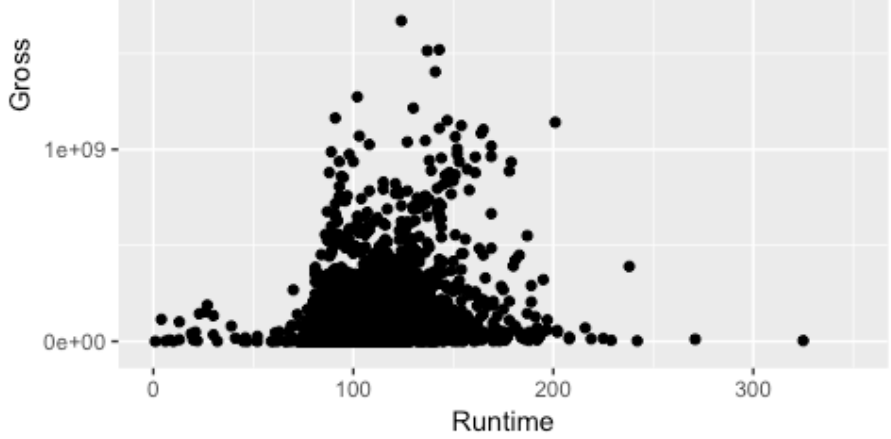
Warning: Removed 28810 rows containing missing values (geom_point).




```
p_wRuntime<- ggplot(df, aes(Runtime, Gross))
p_wRuntime+geom_point() +
 ggtitle("Runtime of movie against Gross Revenue")+
 xlim(0, 350)
```

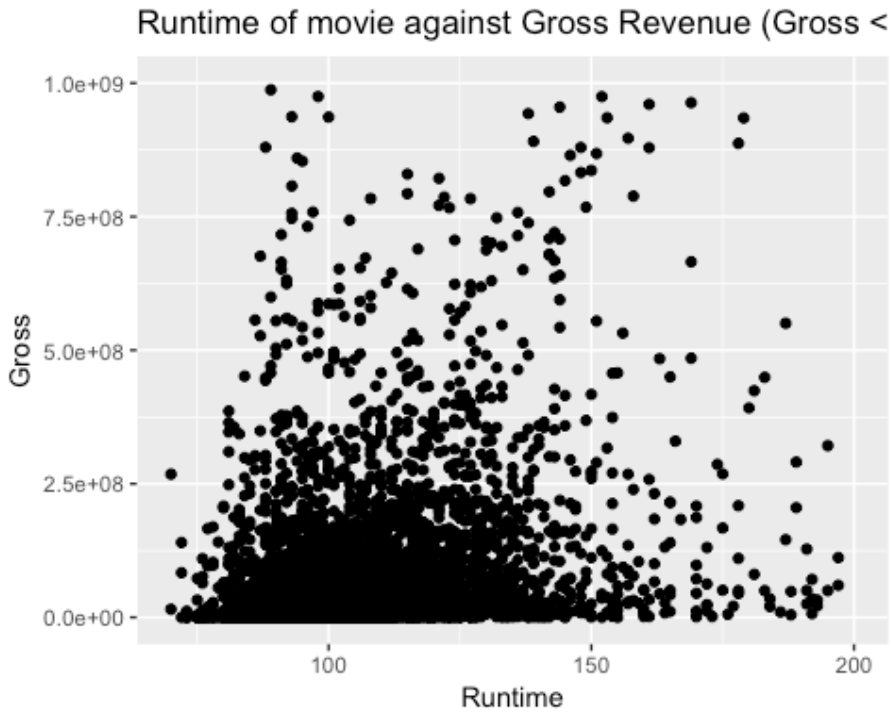
Warning: Removed 28835 rows containing missing values (geom_point).





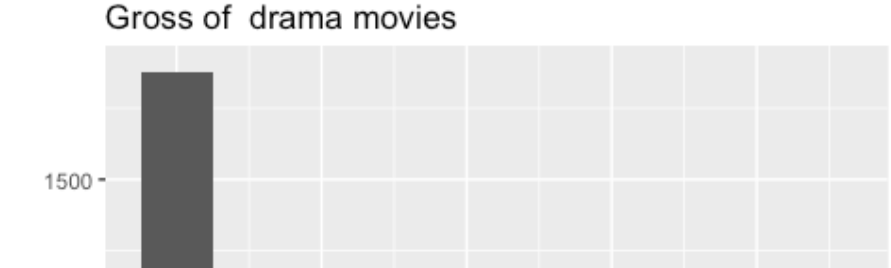

```
p_wRuntime +geom_point()+ xlim(70, 200) + ylim(0, 1000000000) +
 ggtitle("Runtime of movie against Gross Revenue (Gross < 1 Bn and 80-200 min)")
```

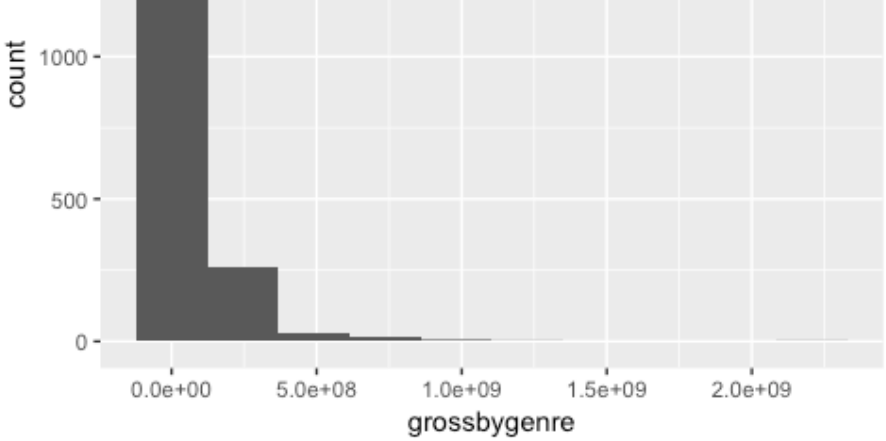
```
## Warning: Removed 28904 rows containing missing values (geom_point).
```




```
# seems to be a correlation with higher grossing movies and 100 minutes but not necessarilly causal
for (x in top_genre_vector){
cat(x, "\n")
mygenre=x
isgenre <-df[[mygenre]]==1

grossbygenre <- df[isgenre,]$Gross
print(describe(grossbygenre))
print(qplot(grossbygenre, bins=10)+ggtitle(paste("Gross of ",x, "movies")) )

}
```

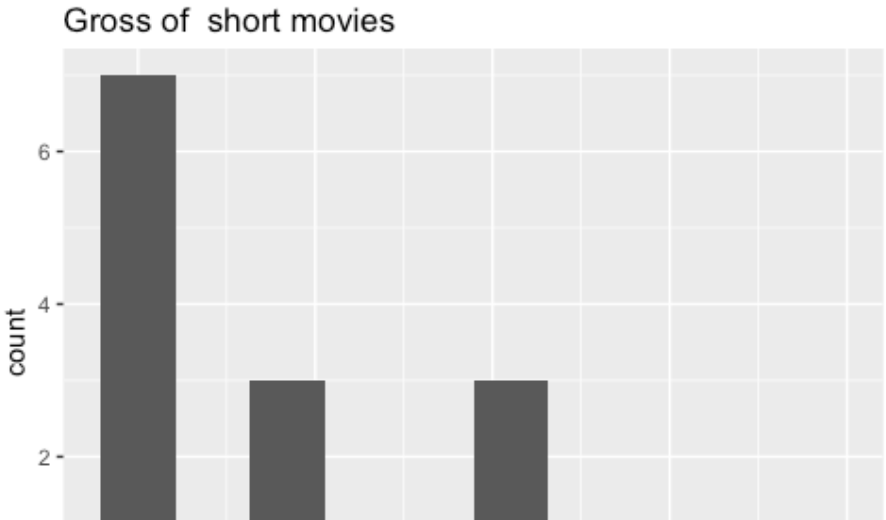
```
## drama
##    vars    n      mean        sd   median  trimmed      mad min        max
## X1    1 2189 61469850 116509794 21758371 36408286 31323768   0 2207615668
##         range skew kurtosis      se
## X1 2207615668 5.93     67.51 2490231
## Warning: Removed 11570 rows containing non-finite values (stat_bin).
```

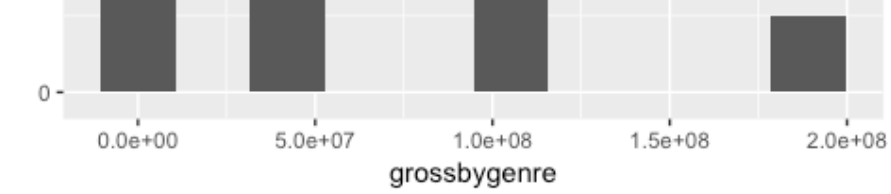




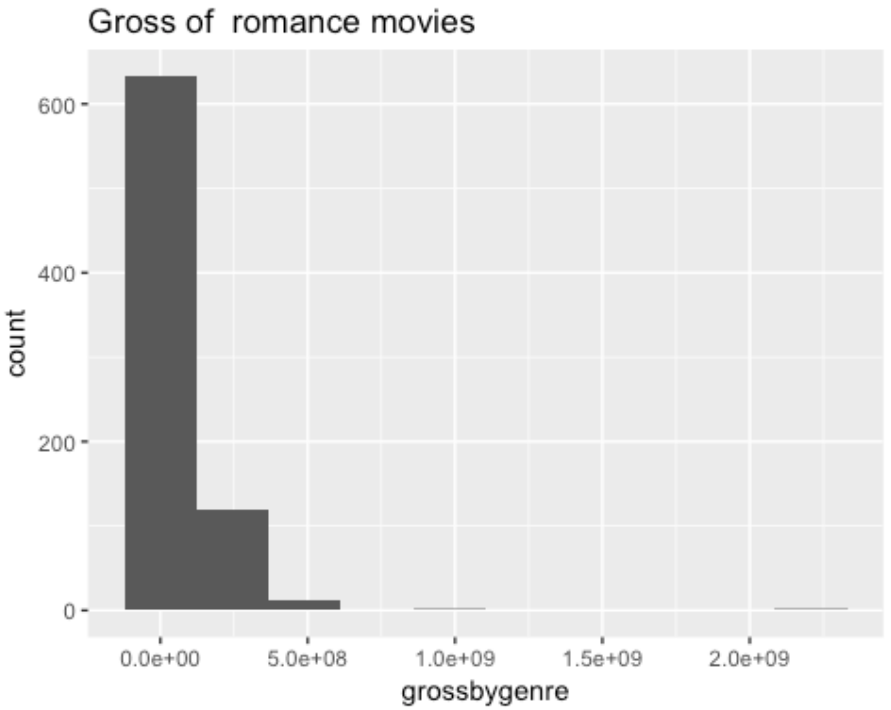
comedy
vars n mean sd median trimmed mad min max
X1 1 1687 82222353 133218073 31063038 52778011 44199627 0 1274234980
range skew kurtosis se
X1 1274234980 3.4 16.25 3243438
Warning: Removed 9639 rows containing non-finite values (stat_bin).

short
vars n mean sd median trimmed mad min max
X1 1 14 46926986 58111302 22344387 38983449 32431000 0 189176423
range skew kurtosis se
X1 189176423 1.07 -0.04 15530899
Warning: Removed 4444 rows containing non-finite values (stat_bin).





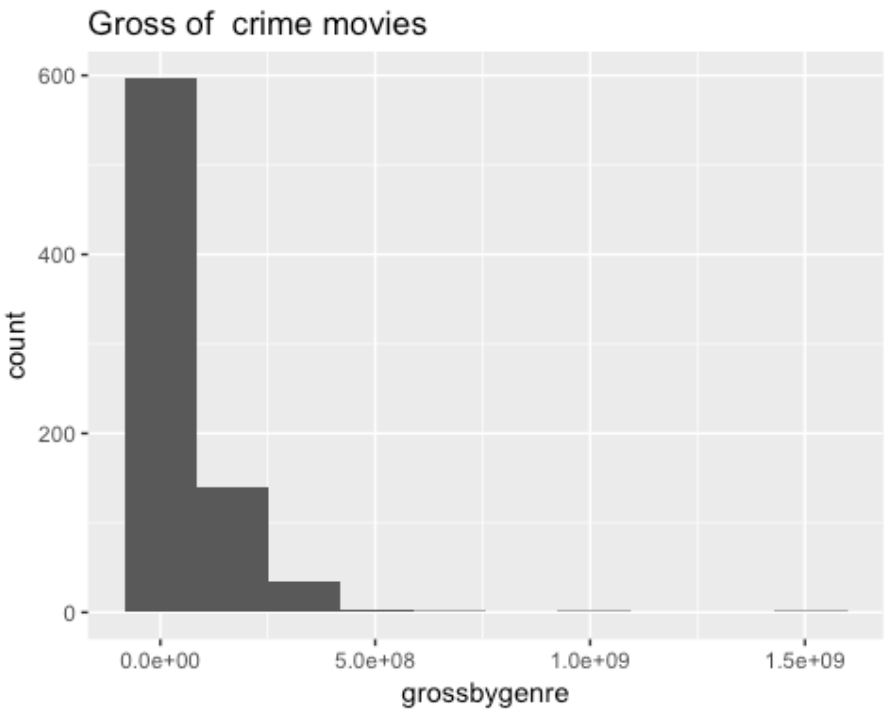
romance
vars n mean sd median trimmed mad min max
X1 1 767 65917082 121387040 24600000 43066760 34465563 0 2207615668
range skew kurtosis se
X1 2207615668 8.41 128.57 4383032
Warning: Removed 3745 rows containing non-finite values (stat_bin).



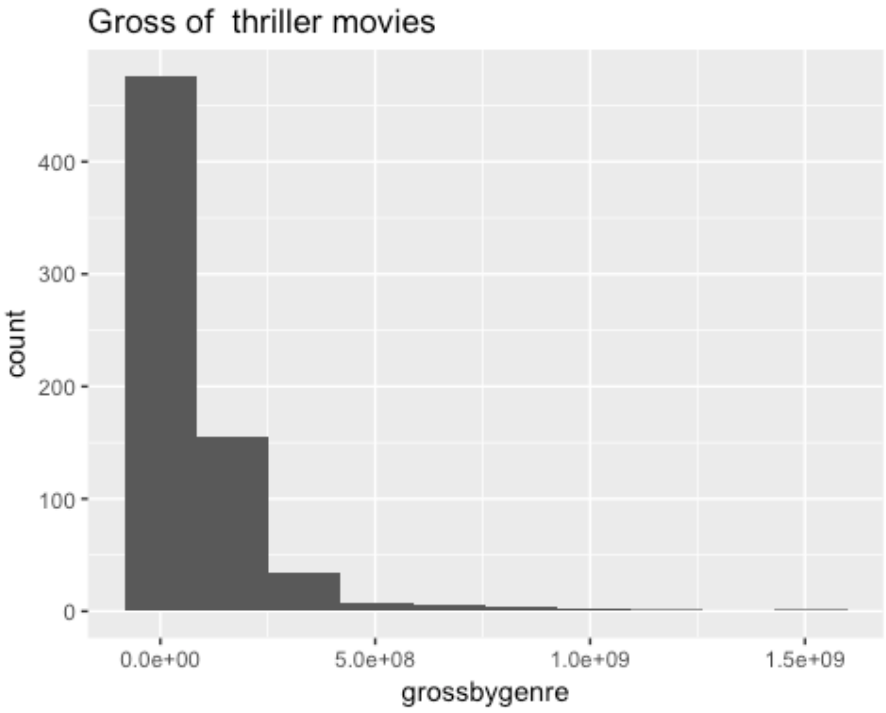
action
vars n mean sd median trimmed mad min
X1 1 1012 160464152 237211238 76820489 110216915 100780390 0
max range skew kurtosis se
X1 2783918982 2783918982 3.51 21.1 7456671
Warning: Removed 2884 rows containing non-finite values (stat_bin).

crime

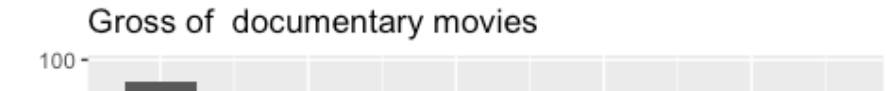
```
##    vars   n      mean        sd   median  trimmed       mad min         max
## X1    1 777 65750678 105883662 29317886 44688250 40439335   0 1514019071
##          range skew kurtosis       se
## X1 1514019071 5.36     53.92 3798555
## Warning: Removed 2901 rows containing non-finite values (stat_bin).
```

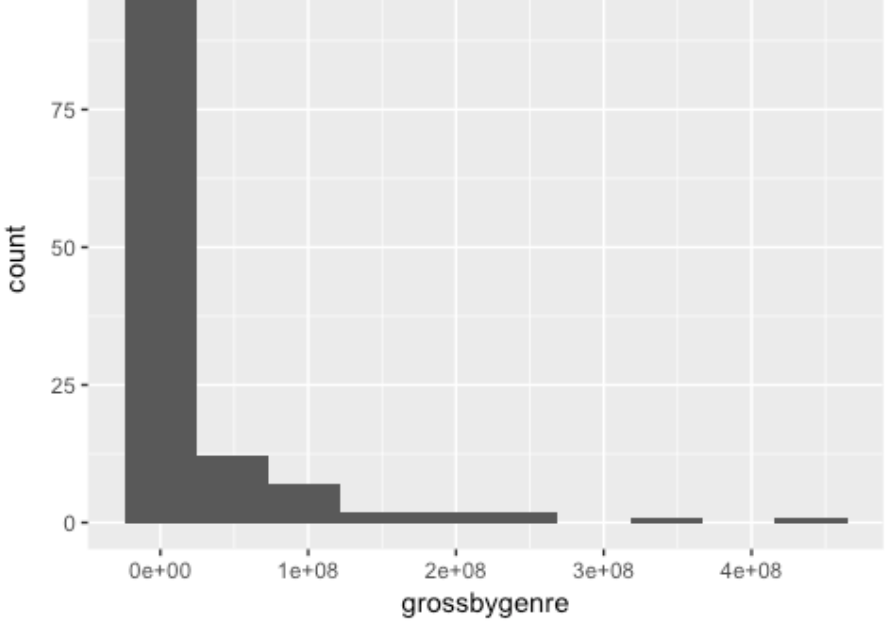



```
## thriller
##    vars   n      mean        sd   median  trimmed       mad min         max
## X1    1 687 87775259 148701491 38959900 56166938 56282430   0 1514019071
##          range skew kurtosis       se
## X1 1514019071 4.13     24.26 5673316
## Warning: Removed 2263 rows containing non-finite values (stat_bin).
```

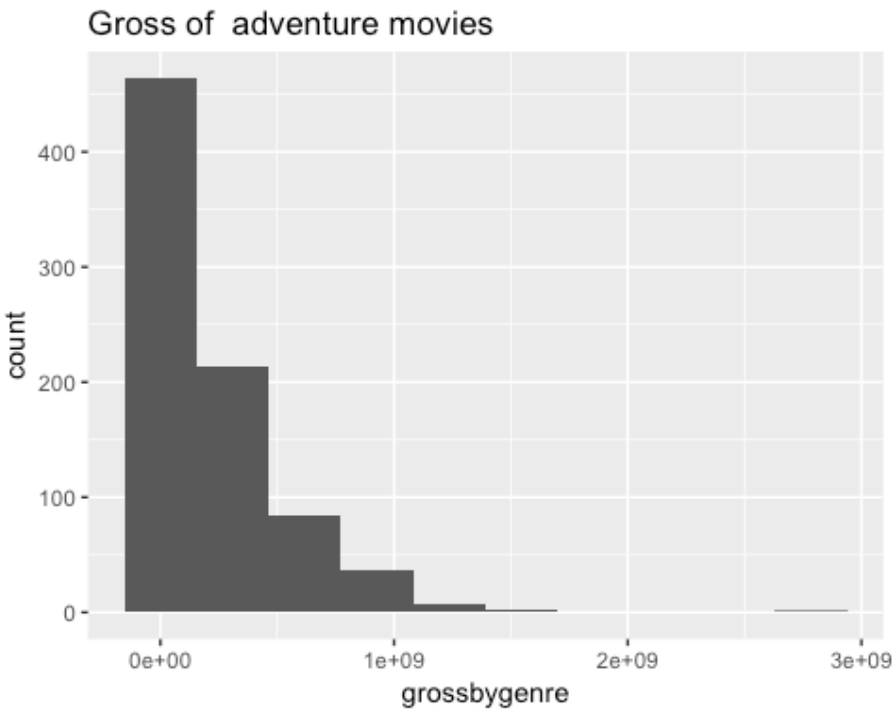



```
## documentary
##    vars   n      mean        sd   median  trimmed      mad min         max
## X1    1 123 27233062 66559881 1162014 10361456 1722802   0 440160956
##          range skew kurtosis       se
## X1 440160956 3.84     16.87 6001502
## Warning: Removed 2045 rows containing non-finite values (stat_bin).
```

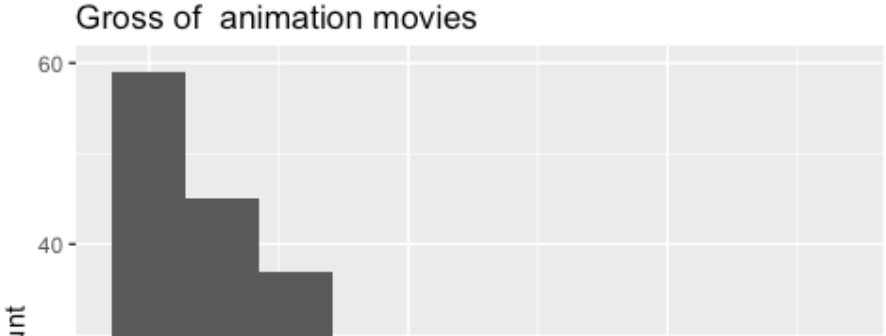




adventure
vars n mean sd median trimmed mad min
X1 1 810 225164210 283376310 109179540 170183187 143112632 0
max range skew kurtosis se
X1 2783918982 2783918982 2.39 10.2 9956829
Warning: Removed 1861 rows containing non-finite values (stat_bin).



animation
vars n mean sd median trimmed mad min
X1 1 202 270803799 265043060 193463686 232505367 234525370 0
max range skew kurtosis se
X1 1274234980 1274234980 1.21 1.16 18648365
Warning: Removed 2140 rows containing non-finite values (stat_bin).



Q: Did you find any observable relationships or combinations of Budget/Runtime/Genre that result in high Gross revenue? If you divided the movies into different subsets, you may get different answers for them - point out interesting ones.

A: Budget vs Gross:
There seems to be a weak correlation between budget and gross revenue. This suggests that in order to make money with movies it is necessary to have the resources and clout for distribution and sales. However, there are movies that despite spending millions are not very profitable.

When segmenting to movies with a budget less than 10M the plots show less of a linear relationship and show that movies of any budget could gross more than $200 M but generally stay under $5M in gross.

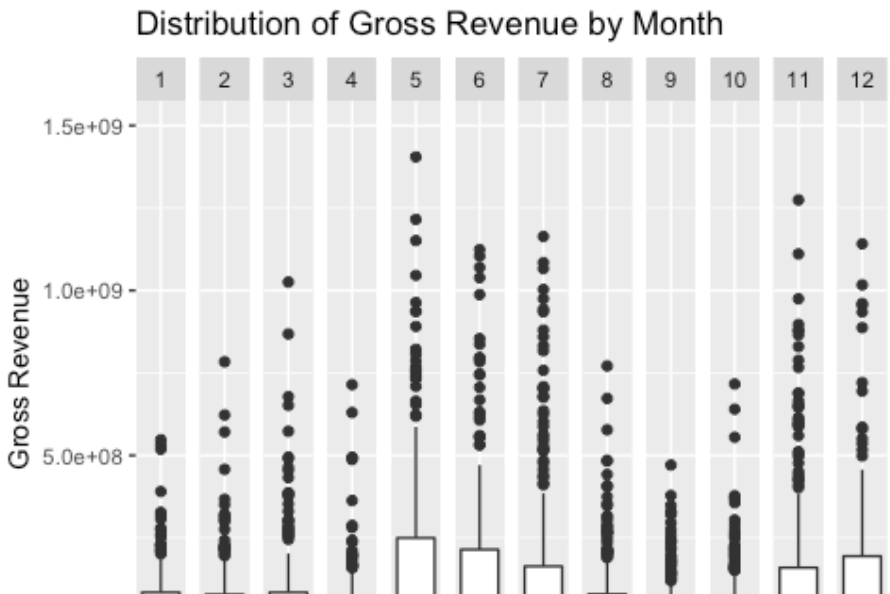
Runtime vs. Gross: The "Runtime of movies against Gross Revenue" plot suggests that in order to Gross in the $1Bn a movie time of greater than 70 minutes correlated. However when further examining the plots @ (Gross < 1 Bn and 80-200 min) it shows in further detail that movies runtime ranges from 80-150 min and these movies concentrate at 0 Gross.

Genre vs Gross: Each other distributions are highly skewed towards gross = 0, meaning regardless of category there are lot of movies that don't make money. Interestingly enough, animations seem to have the least amount of skew and suggests that most animation movies are more profitable. However, there are much less animation movies and have less upside Gross. The top animation made $1,274,234,980 vs top action movie made $2,783,918,982.

Month vs Gross:
When reviewing the boxplot of Gross Revenue by Month the IQR are higher in the months of May, Jun, July, November, December.

```
# TODO: Investigate if Gross Revenue is related to Release Month
df$Released_month<- as.numeric(format(df$Released,'%m'))
```

```
p_GrossVRelMonth <- ggplot(df, aes("", Gross))
p_GrossVRelMonth +geom_boxplot() +facet_grid(.~Released_month)+
 ggtitle("Distribution of Gross Revenue by Month")+
 xlab("Month")+ylab("Gross Revenue") +ylim(0, 1500000000)
## Warning: Removed 28815 rows containing non-finite values (stat_boxplot).
```



6. Process Awards column

The variable Awards describes nominations and awards in text format. Convert it to 2 numeric columns, the first capturing the number of wins, and the second capturing nominations. Replace the Awards column with these new columns, and then study the relationship of Gross revenue with respect to them.

Note that the format of the Awards column is not standard; you may have to use regular expressions to find the relevant values. Try your best to process them, and you may leave the ones that don't have enough information as NAs or set them to 0s.

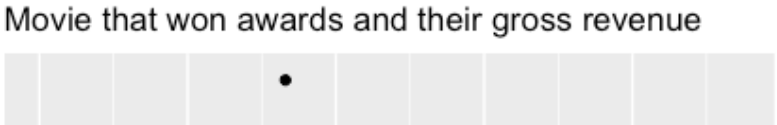
```
# TODO: Convert Awards to 2 numeric columns: wins and nominations
df$Awards<- tolower(df$Awards)

winphrase = "win|won"
nomphrase ="nomination|nominated"

convertawardsWin <- function(awardtext){
win <- 0
awardarray <-strsplit(awardtext, split="[.&]")
for (phrase in awardarray[[1]]){
number <- unique(na.omit(as.numeric(unlist(strsplit(unlist(phrase), "[^0-9]+")))))
if(grepl(winphrase, phrase)){win=win+number
}
}
return(win)
}

convertawardsNoms <- function(awardtext){
nom<-0
awardarray <-strsplit(awardtext, split="[.&]")
for (phrase in awardarray[[1]]){
# cat(phrase, "\n")
number <- unique(na.omit(as.numeric(unlist(strsplit(unlist(phrase), "[^0-9]+")))))
if(grepl(nomphrase, phrase)){nom=nom+number
}
}
return(nom)
}
# convertawardsWin(awardtext)
# convertawardsNoms(awardtext)
df$Wins <- lapply(df$Awards, convertawardsWin )
df$Nominations <- lapply(df$Awards, convertawardsNoms )
df$Wins <- as.numeric(df$Wins)
df$Nominations <- as.numeric(df$Nominations)

sum(df$Wins>0)
## [1] 9636
sum(df$Nominations>0)
## [1] 10461
```

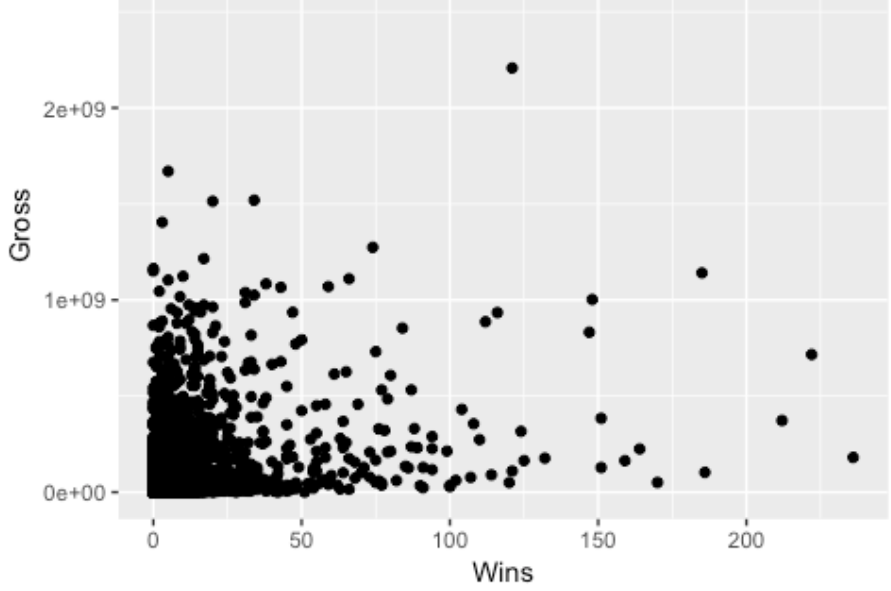
Q: How did you construct your conversion mechanism? How many rows had valid/non-zero wins or nominations?

A: 1) Convert awards to lowercase 2) Identify and define win and nominate phrases 3) create functions to split the string on "." and "&" 4) iterate through string list A) if phrase contains win phrase get number and add to win total B) if phrase contains nom phrase get number and add to nom total 5) Lapply to new Wins and Nominations column

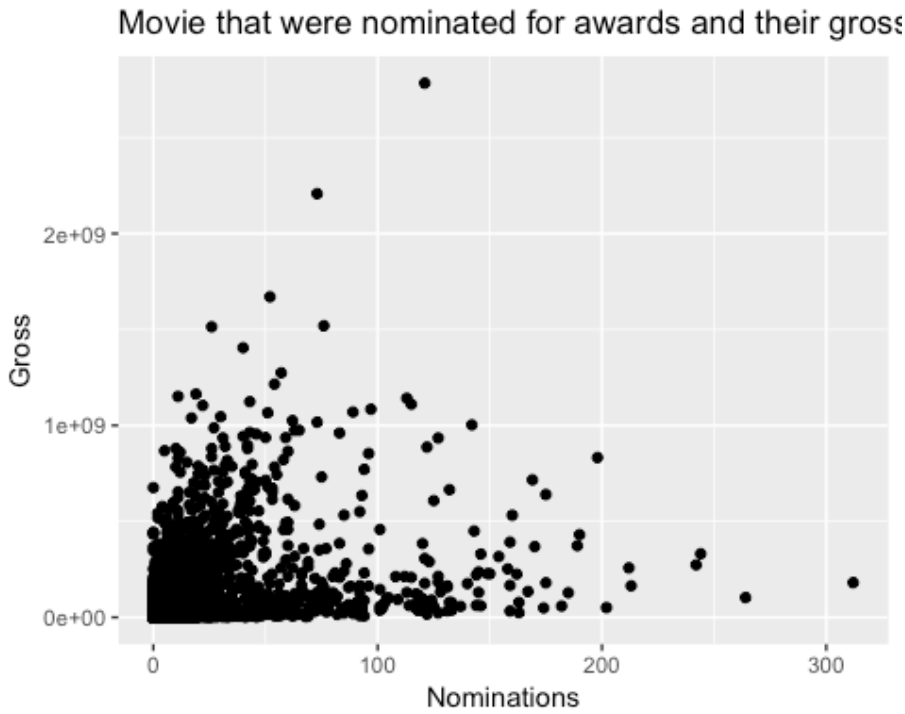
9636 Wins 10461 Nominations

```
# TODO: Plot Gross revenue against wins and nominations
p_wins <- ggplot(df, aes(Wins, Gross))
p_wins + geom_point() + ggtitle("Movie that won awards and their gross revenue")
## Warning: Removed 28810 rows containing missing values (geom_point).
```






```
p_noms <- ggplot(df, aes(Nominations, Gross))
p_noms +geom_point() +ggtitle("Movie that were nominated for awards and their gross revenue")
## Warning: Removed 28810 rows containing missing values (geom_point).
```



Q: How does the gross revenue vary by number of awards won and nominations received?

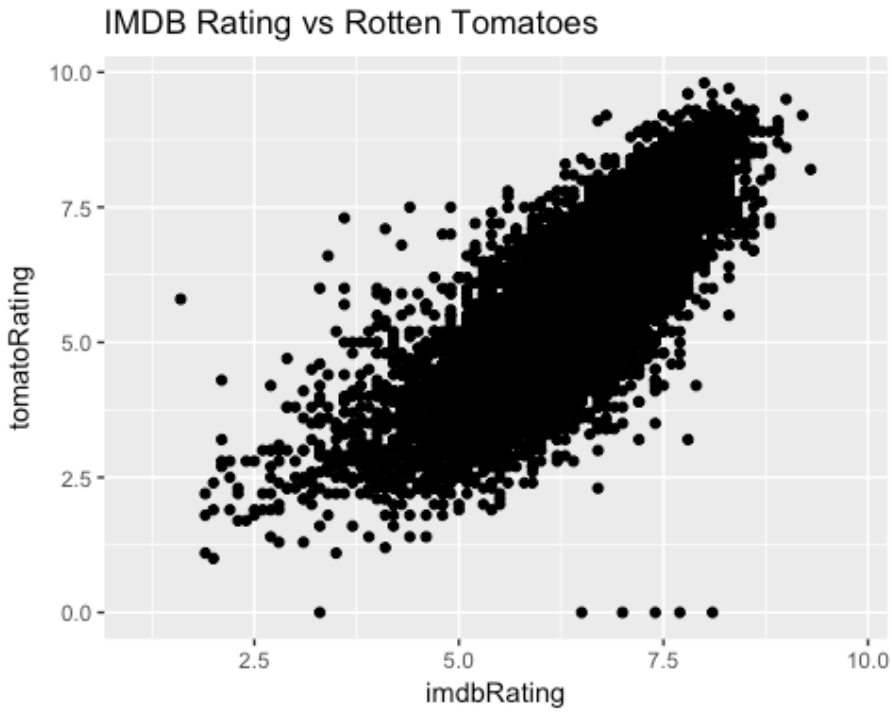
A: Awards won and nominations seems to have no relationship with Gross Revenue. As movies with more wins and nominations increases, the distribution of movies' gross revenue appears to be the same as the overall population of movies. Both wins and nominations have the highest concentration at the origin, indicating that regardless of awards and nominations, it's hard to make money with movies. (and having an award winning movie doesn't mean it will make money)

7. Movie ratings from IMDb and Rotten Tomatoes

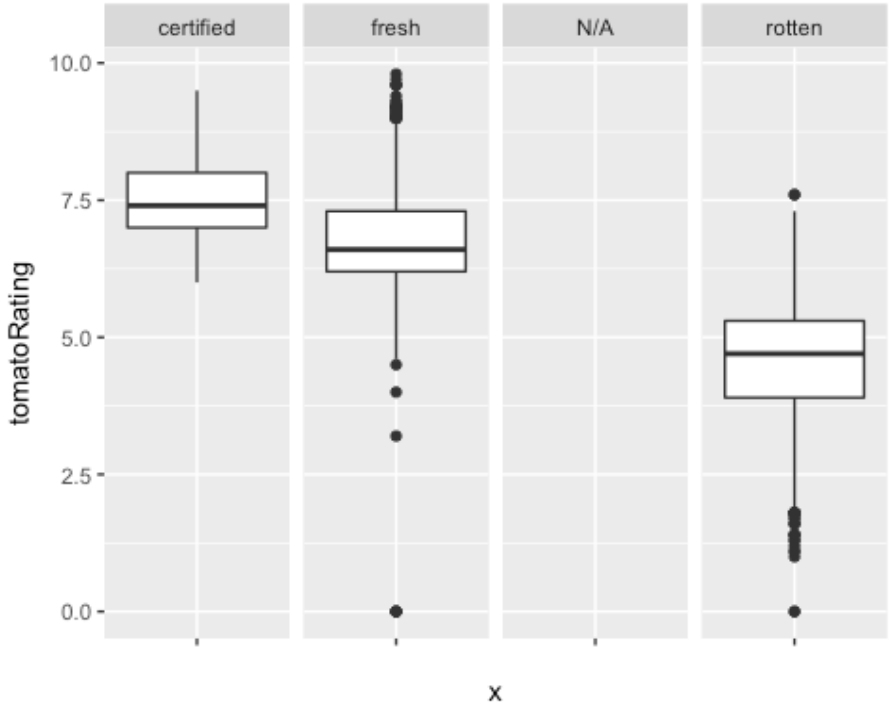
There are several variables that describe ratings, including IMDb ratings (imdbRating represents average user ratings and imdbVotes represents the number of user ratings), and multiple Rotten Tomatoes ratings (represented by several variables pre-fixed by tomato). Read up on such ratings on the web (for example rottentomatoes.com/about and www.imdb.com/help/show_leaf?votestopfaq).

Investigate the pairwise relationships between these different descriptors using graphs.

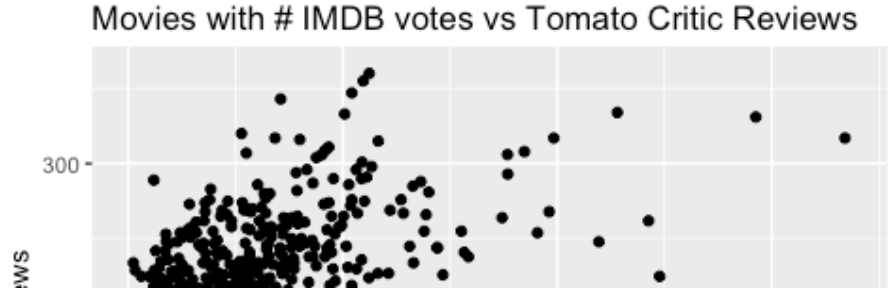
```
# TODO: Illustrate how ratings from IMDb and Rotten Tomatoes are related
p_7 <- ggplot(df, aes(imdbRating, tomatoRating))
p_7 + geom_point() +ggtitle("IMDB Rating vs Rotten Tomatoes")
## Warning: Removed 24202 rows containing missing values (geom_point).
```

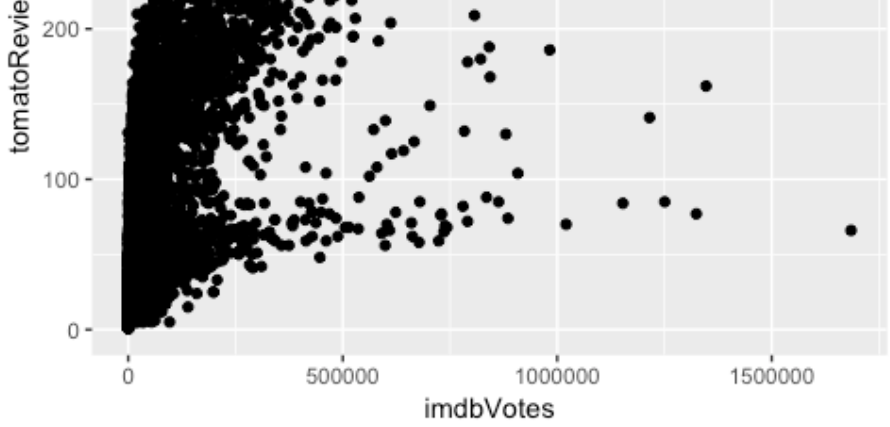



```
p_7_2 <- ggplot(df, aes("", tomatoRating))
p_7_2+geom_boxplot() + facet_grid(.~tomatoImage)
## Warning: Removed 24201 rows containing non-finite values (stat_boxplot).
```

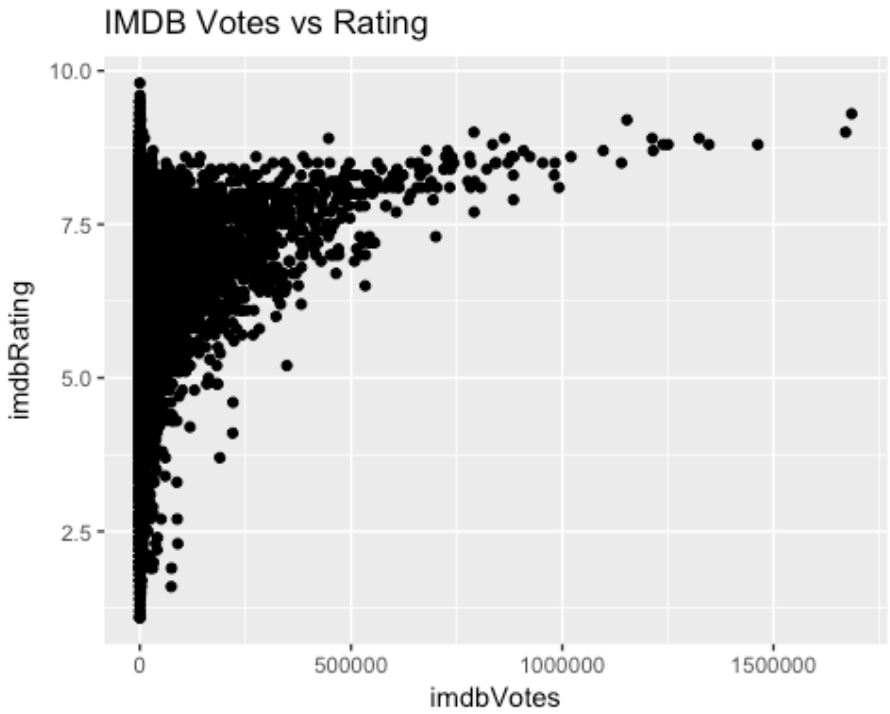



```
# p_7 + geom_point() + facet_grid(.~tomatoImage)+ ggtitle("IMDB Rating vs Rotten Tomatoes")
# df$tomatoImage[df$tomatoImage == "N/A"]=NA
# p_7_1 <- ggplot(df, aes(tomatoRating))
# p_7_1+geom_bar() + facet_grid(.~tomatoImage)
p_7_3 <- ggplot(df, aes(imdbVotes, tomatoReviews))
p_7_3 + geom_point() + ggtitle("Movies with # IMDB votes vs Tomato Critic Reviews")
## Warning: Removed 24170 rows containing missing values (geom_point).
```

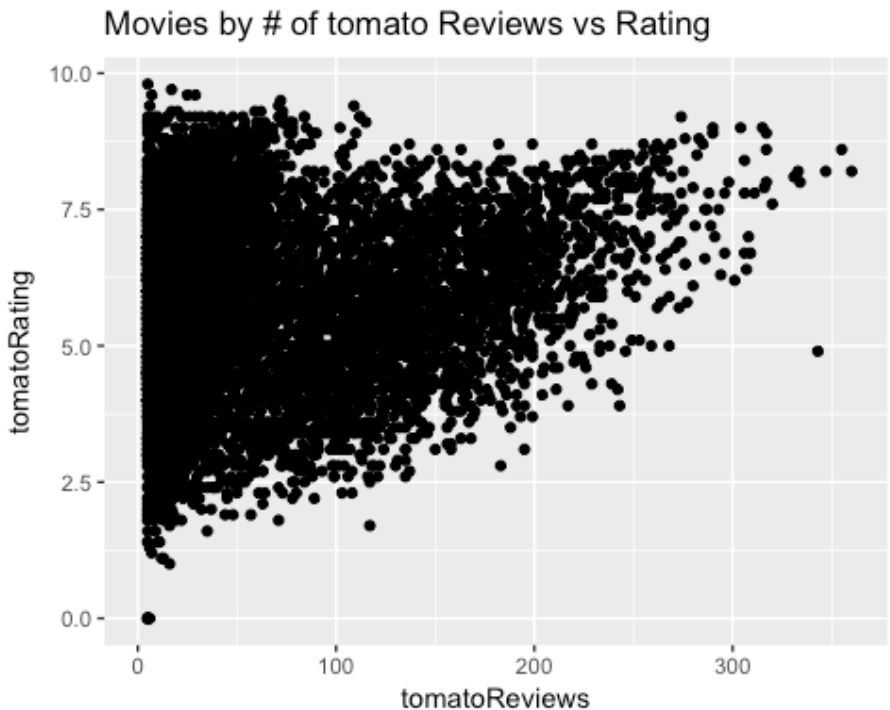





```
p_7_4 <- ggplot(df, aes(imdbVotes, imdbRating))
p_7_4+ geom_point() + ggtitle("IMDB Votes vs Rating")
## Warning: Removed 690 rows containing missing values (geom_point).
```




```
p_7_5 <- ggplot(df, aes(tomatoReviews, tomatoRating))
p_7_5 + geom_point() +  ggtitle("Movies by # of tomato Reviews vs Rating")
## Warning: Removed 24201 rows containing missing values (geom_point).
```



Q: Comment on the similarities and differences between the user ratings of IMDb and the critics ratings of Rotten Tomatoes.

A: At first blush, a comparison of imdbRating and tomatoRating shows strong positive correlation. The strong positive linear relationship is almost textbook. Also, when seems to align when segmenting the data based upon the Rotten Tomato image categories which suggests that the regardless of whether a movie is good or bad(or in Rotten Categories: Certified Fresh, Fresh, and Rotten), the IMDB ratings and Rotten Critic ratings will be consisent.

Differences are apparent in the way the ratings are created. IMDB uses tens of thousands of votes; whereas, Rotten Tomatoes uses < 300 reviews to create their score.

An interesting observation is that as reviews increase the likelihood of moving being highly rated increases. This may be correlated vs causal meaning that this is probably because good movies attract more critical review versus more critical reviews make higher rated movies. (Reference p_7_4 and p_7_5)

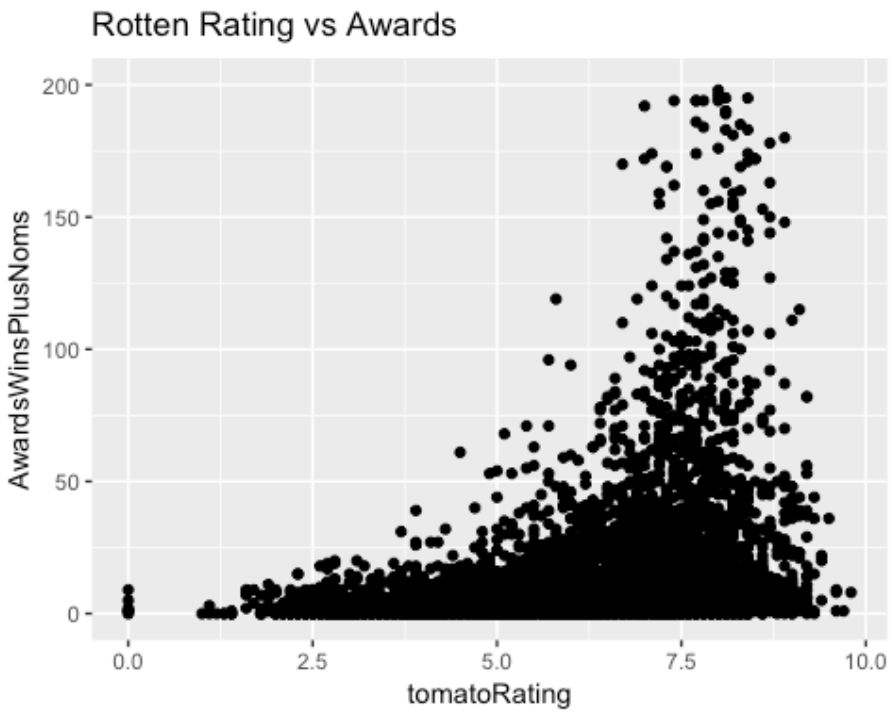
8. Ratings and awards

These ratings typically reflect the general appeal of the movie to the public or gather opinions from a larger body of critics. Whereas awards are given by professional societies that may evaluate a movie on specific attributes, such as artistic performance, screenplay, sound design, etc.

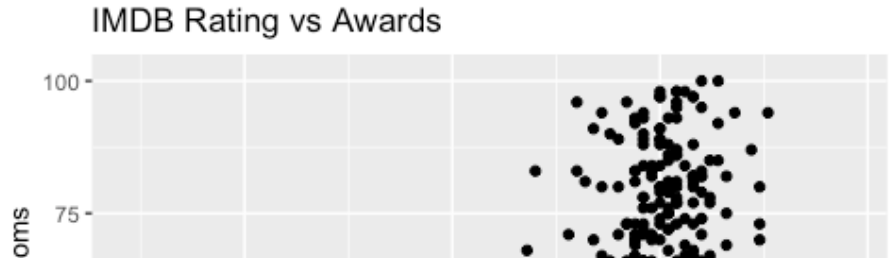
Study the relationship between ratings and awards using graphs (awards here refers to wins and/or nominations).

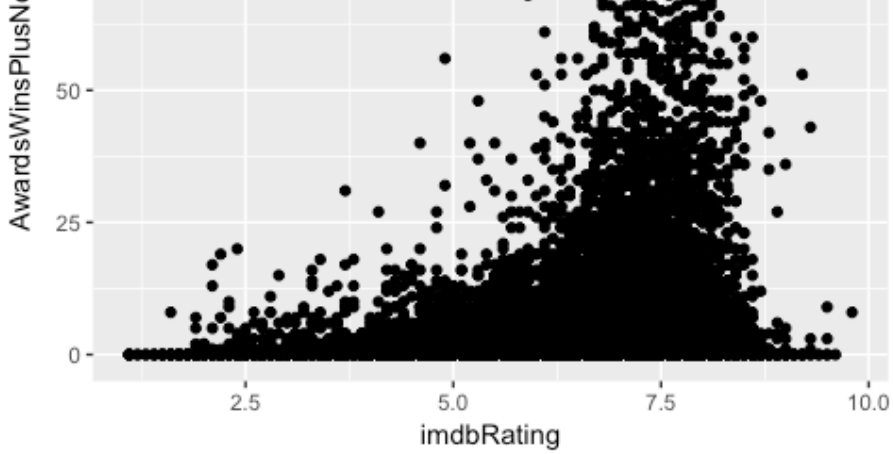
```
# TODO: Show how ratings and awards are related
df$AwardsWinsPlusNoms <- df$Wins + df$Nominations
```

```
p_8 <- ggplot(df, aes(tomatoRating, AwardsWinsPlusNoms ))
p_8 + geom_point() + ggtitle("Rotten Rating vs Awards")+ ylim(0,200)
## Warning: Removed 24251 rows containing missing values (geom_point).
```



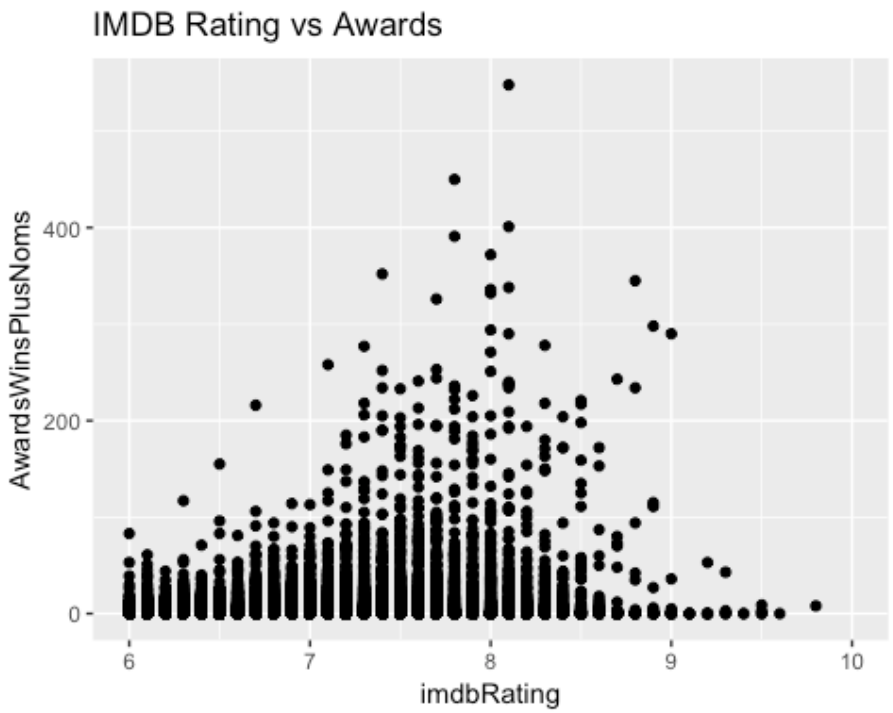

```
p_8_2 <- ggplot(df, aes(imdbRating, AwardsWinsPlusNoms ))
p_8_2 + geom_point()+ ggtitle("IMDB Rating vs Awards") + ylim(0, 100)
## Warning: Removed 829 rows containing missing values (geom_point).
```






```
p_8_2 + geom_point()+ ggtitle("IMDB Rating vs Awards") + xlim(6, 10)
```

```
## Warning: Removed 11740 rows containing missing values (geom_point).
```



Q: How good are these ratings in terms of predicting the success of a movie in winning awards or nominations? Is there a high correlation between two variables?

A: If a movie has a high rating does not ensure that a movie will win awards; however, movies that have more wins and nominations tend to have rating of about 7 or higher for both IMDB and Rotten Ratings.

However, higher ratings correlates with movies with wins and nominations.

9. Expected insights

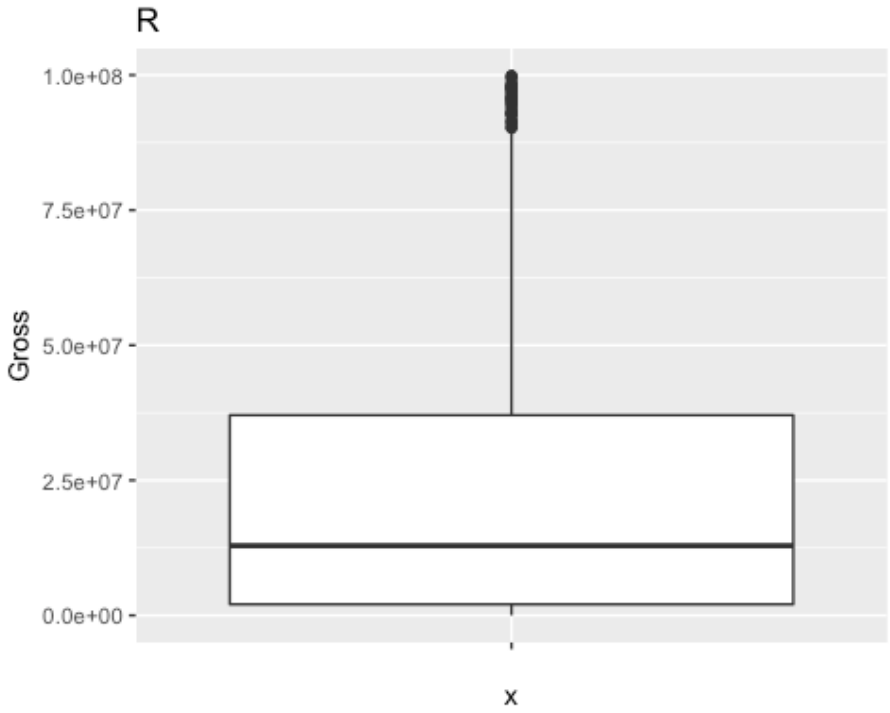
Come up with two new insights (backed up by data and graphs) that is expected. Here "new" means insights that are not an immediate consequence of one of the above tasks. You may use any of the columns already explored above or a different one in the dataset, such as Title, Actors, etc.

```
# TODO: Find and illustrate two expected insights

# Does movie rating (G, PG-13, R) impact profitability

Rated_cats = c("R", "PG-13", "PG", "G")

for (ratings in Rated_cats){
ratingcondition =df$Rated==ratings
sumrated= sum(ratingcondition)
cat(ratings, sumrated,"\n")
df_rated <- df[ratingcondition,]
print(ggplot(df_rated, aes("", Gross))+geom_boxplot() +ggtitle(ratings )+ylim(0,100000000))
print(ggplot(df_rated, aes(Gross)) + geom_histogram(binwidth = 10000000)+ ggtitle(ratings )+
```

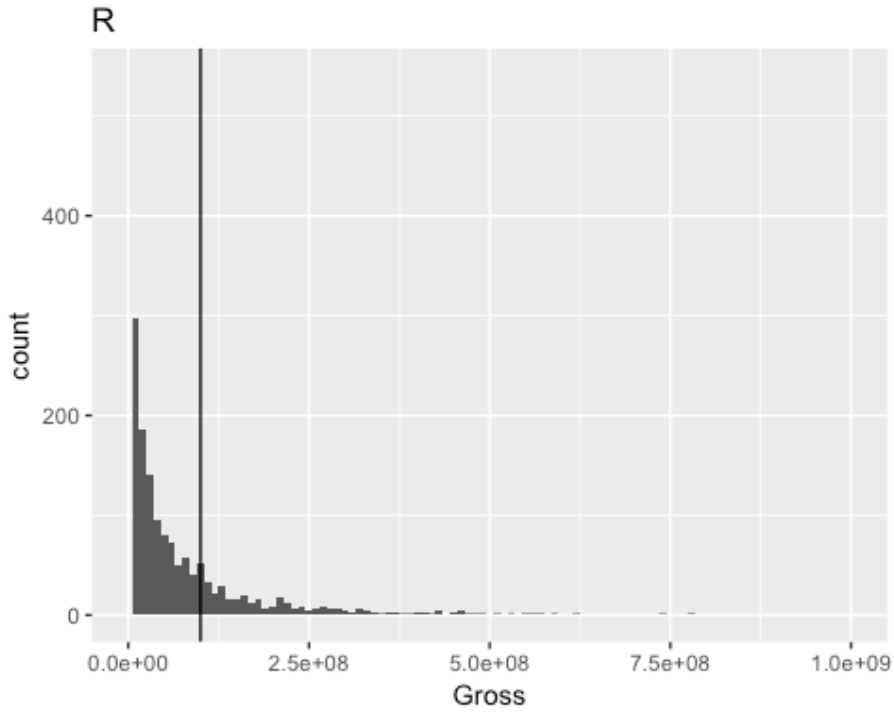
```
      geom_vline(xintercept = 100000000)+ xlim(0, 1000000000))
# print(describe(runtimes))
# print(qplot(runtimes, binwidth=50)+ggtitle(paste("Runtime of ",x, "movies")) + xlim(0,300))
}
```

R 6390

Warning: Removed 4809 rows containing non-finite values (stat_boxplot).

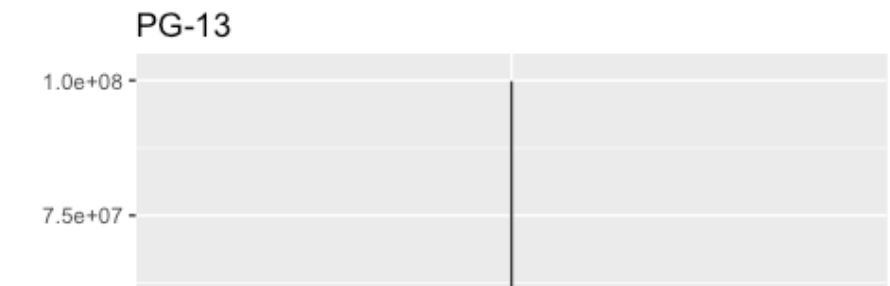


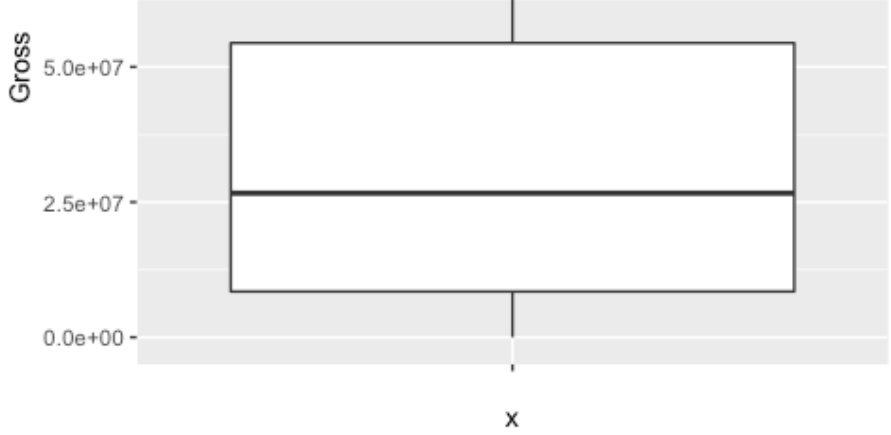
Warning: Removed 4475 rows containing non-finite values (stat_bin).



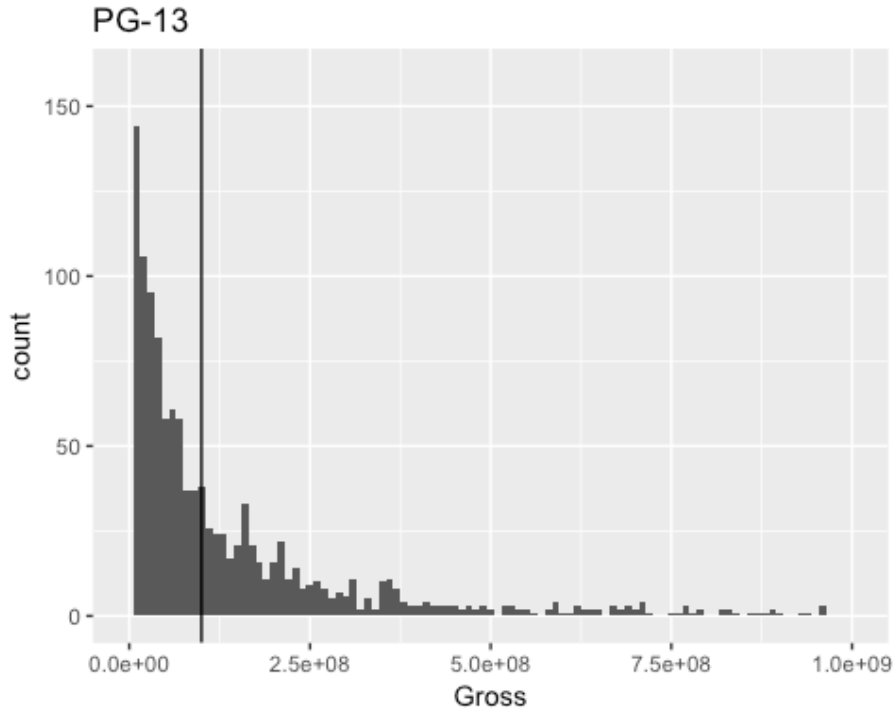
PG-13 2269

Warning: Removed 1413 rows containing non-finite values (stat_boxplot).

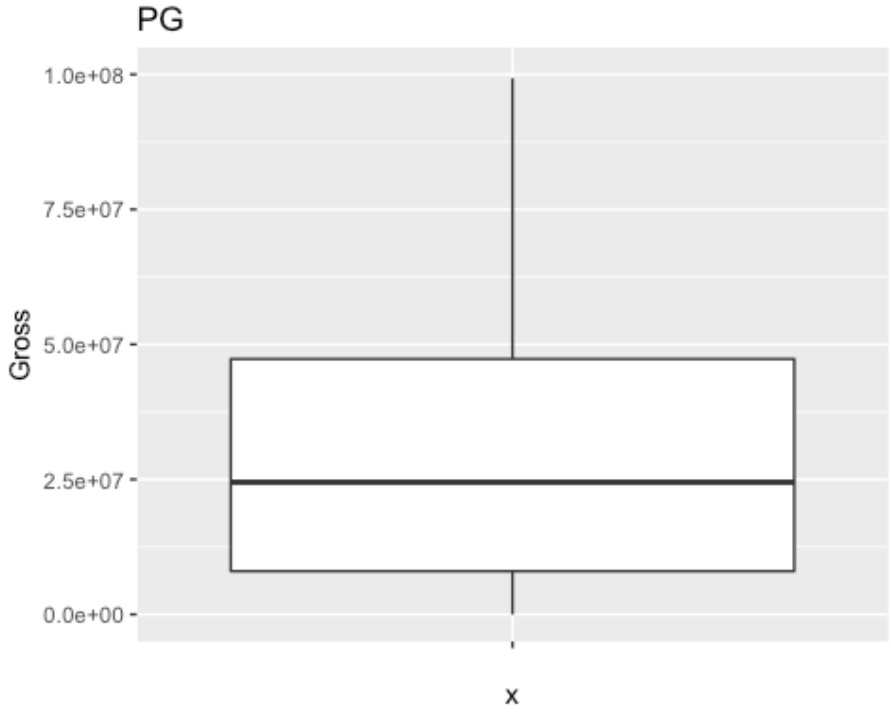




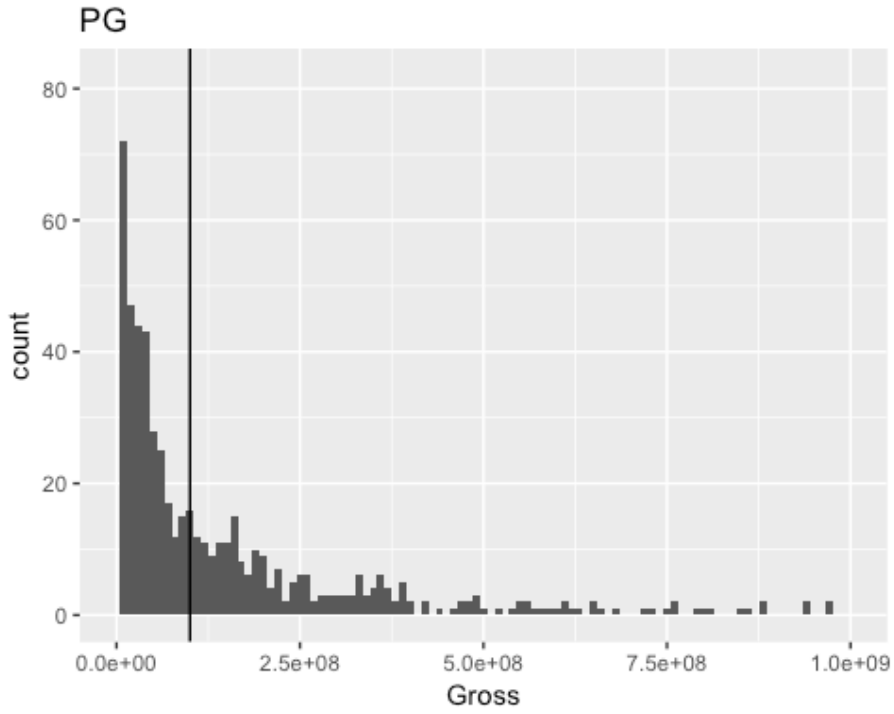
Warning: Removed 931 rows containing non-finite values (stat_bin).



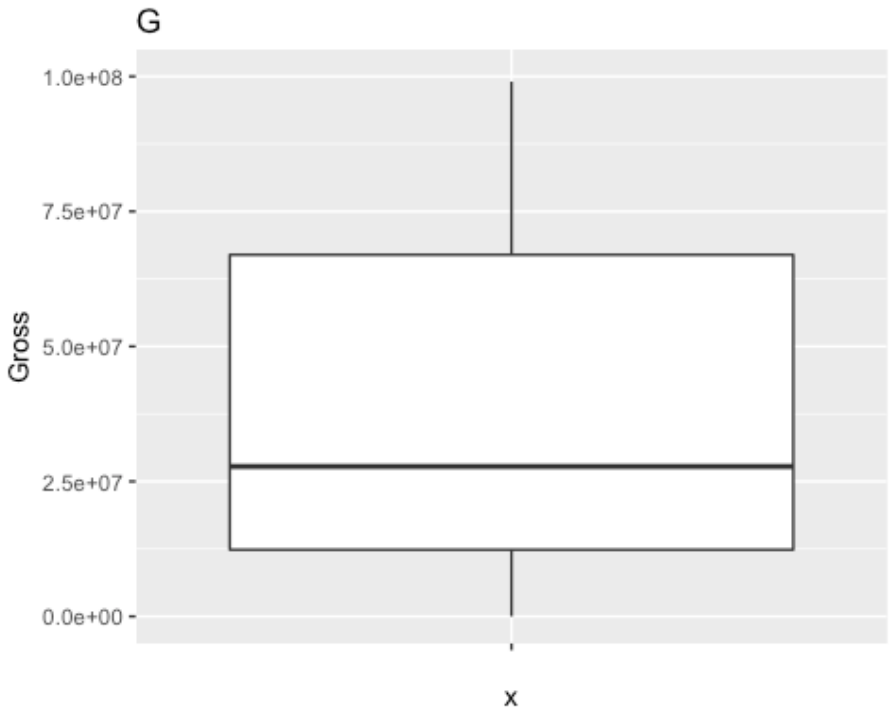
PG 2019
Warning: Removed 1622 rows containing non-finite values (stat_boxplot).



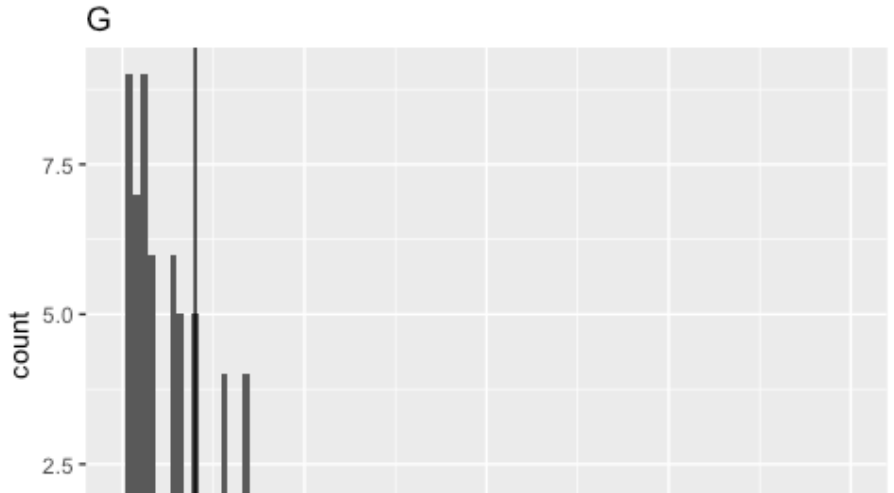
Warning: Removed 1391 rows containing non-finite values (stat_bin).



G 520
Warning: Removed 460 rows containing non-finite values (stat_boxplot).



Warning: Removed 417 rows containing non-finite values (stat_bin).




```
# sum(df$Rated=="R")
# df[df$Rated=="PG",]
# df[df$Rated=="PG-13",]
# df[df$Rated=="G",]
```

```
# p9 <- ggplot(df, aes(Rated))
# p9 + geom_bar()
```

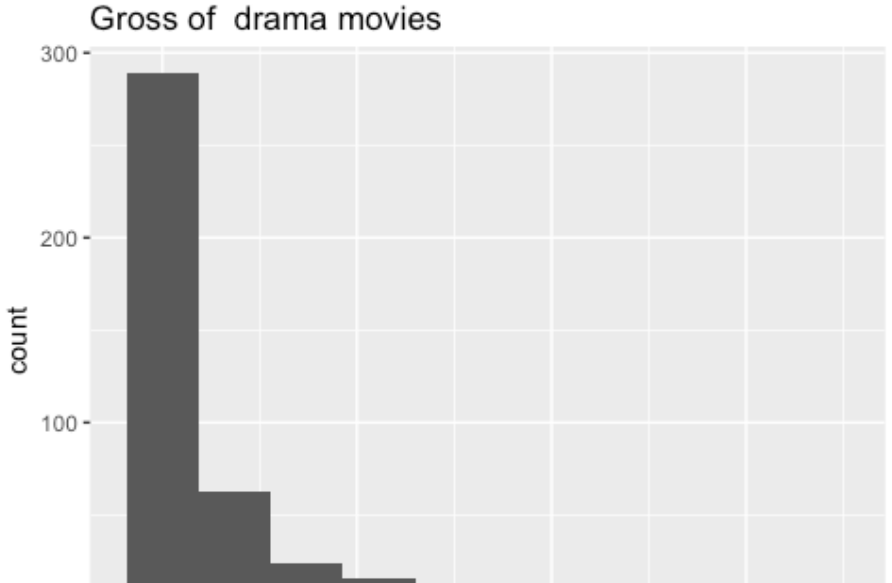
Q: Expected insight #1.

A: I wanted to examine if there's a relationship between movie ratings of R, PG, PG-13, and G as compared to gross revenue. By shear volume rated R movies are the most with 6,390, followed by PG-13(2269), PG(2019), and G (520). What is consistent throughout the categories is that there are lot of movies make less than $100M regardless of Parental Guidance Rating. Despite G having the greatest assumed potential audience, there are less G rated movies in volume.

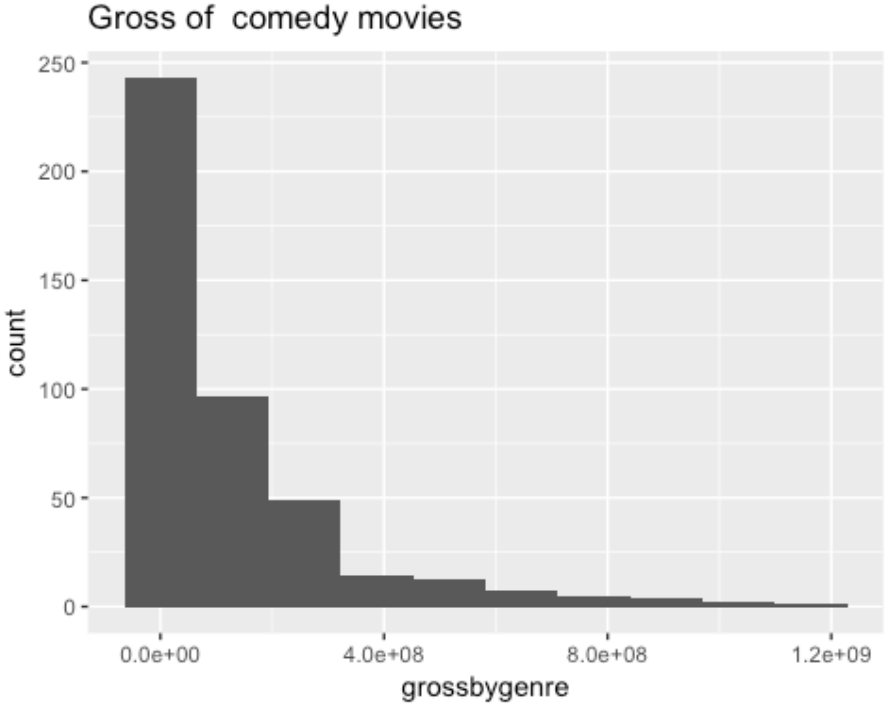
Given the distributions on the histograms look visually similar, I can generally conclude that movie parental guidance rating does not a strong correlation with Gross Revenue. I assumed that G movies because of greater audience would be more profitable but was proven wrong.

Q: Expected insight #2. What genre of movies made the most money in the summer months?

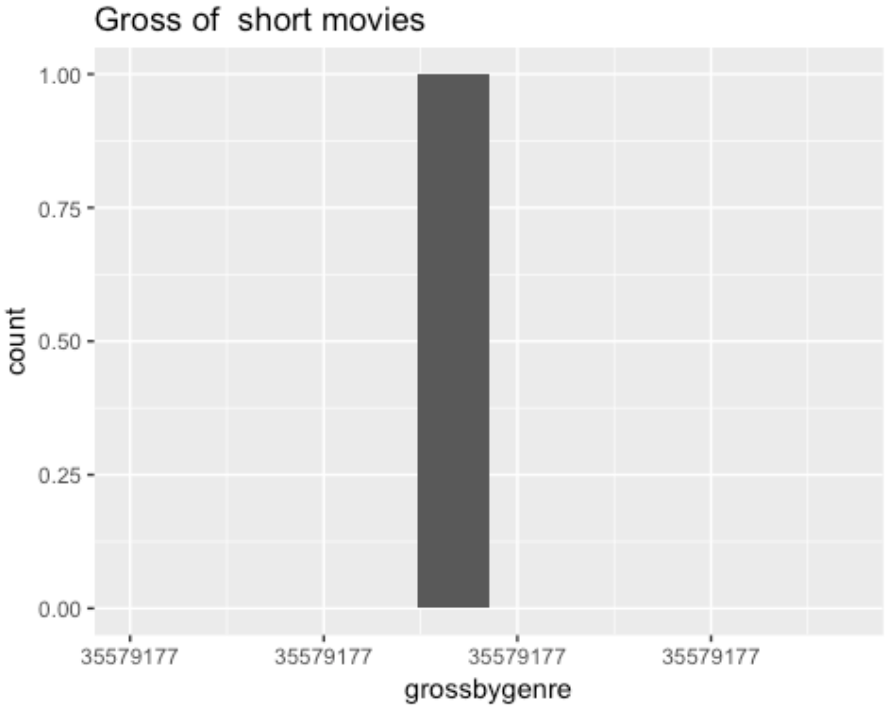
```
df_summer <- df[(df$Released_month==5)|(df$Released_month==6)|(df$Released_month==7),]

for (x in top_genre_vector){
cat(x, "\n")
mygenre=x
isgenre <-df_summer[[mygenre]]==1

grossbygenre <- df_summer[isgenre,]$Gross
print(describe(grossbygenre))
print(qplot(grossbygenre, bins=10)+ggtitle(paste("Gross of ",x, "movies")) )

}
## drama
##    vars   n     mean       sd   median  trimmed      mad min       max
## X1    1 409 72875725 136751547 14314407 39697034 20972229   0 1002891358
##         range skew kurtosis      se
## X1 1002891358 3.24    13.36 6761929
## Warning: Removed 2499 rows containing non-finite values (stat_bin).
```



comedy
vars n mean sd median trimmed mad min max
X1 1 435 126118447 188987879 45680201 83182928 64113688 0 1163624481
range skew kurtosis se
X1 1163624481 2.52 7.15 9061275
Warning: Removed 2223 rows containing non-finite values (stat_bin).

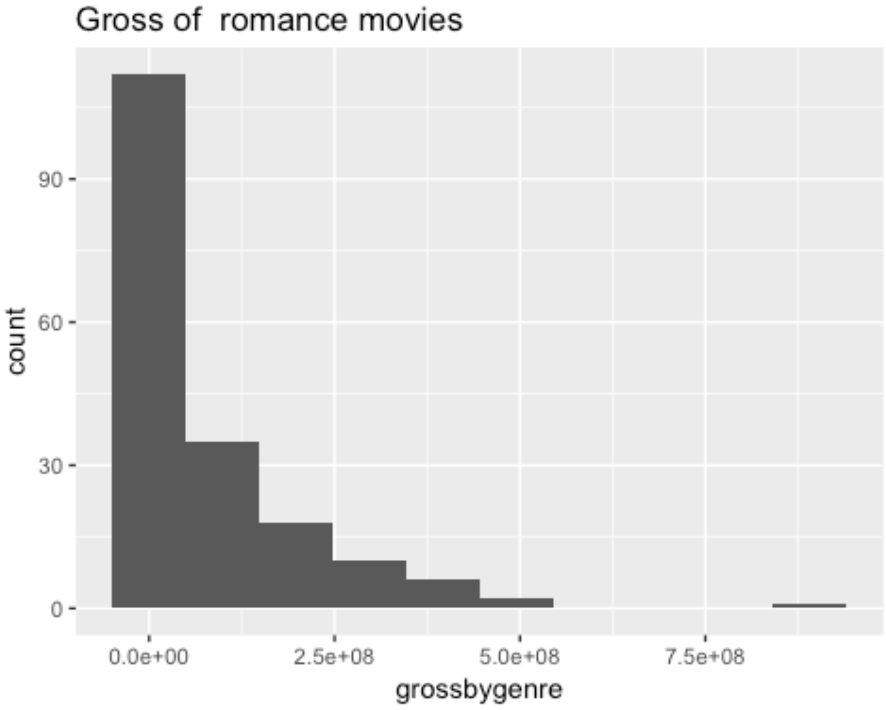


short
vars n mean sd median trimmed mad min max range skew
X1 1 1 35579177 NA 35579177 35579177 0 35579177 35579177 0 NA
kurtosis se
X1 NA NA
Warning: Removed 1119 rows containing non-finite values (stat_bin).

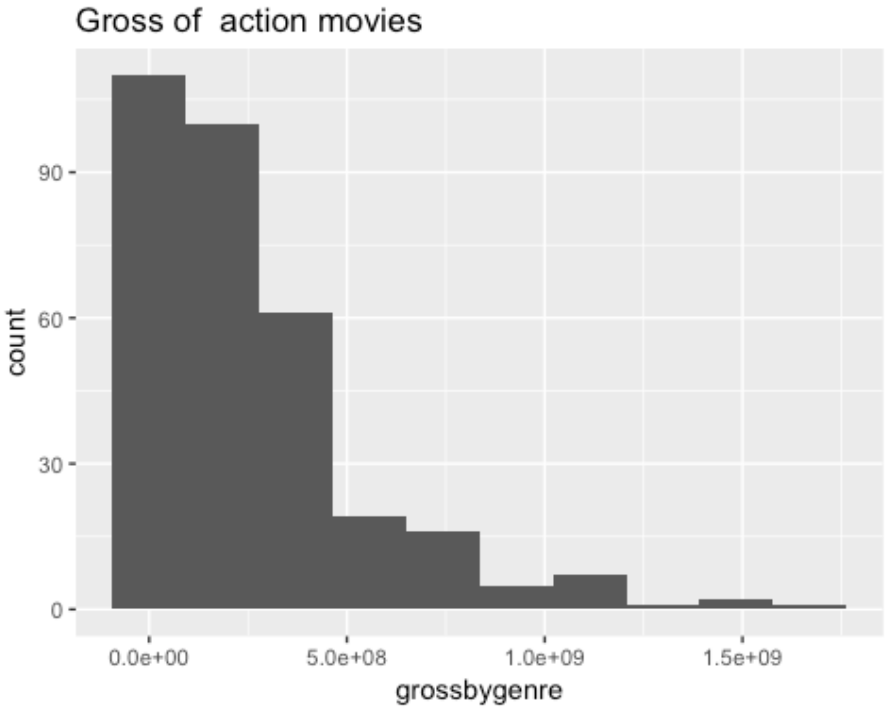


romance
vars n mean sd median trimmed mad min max
X1 1 184 82147473 121739439 24423004 56339963 34884160 0 890875303
range skew kurtosis se

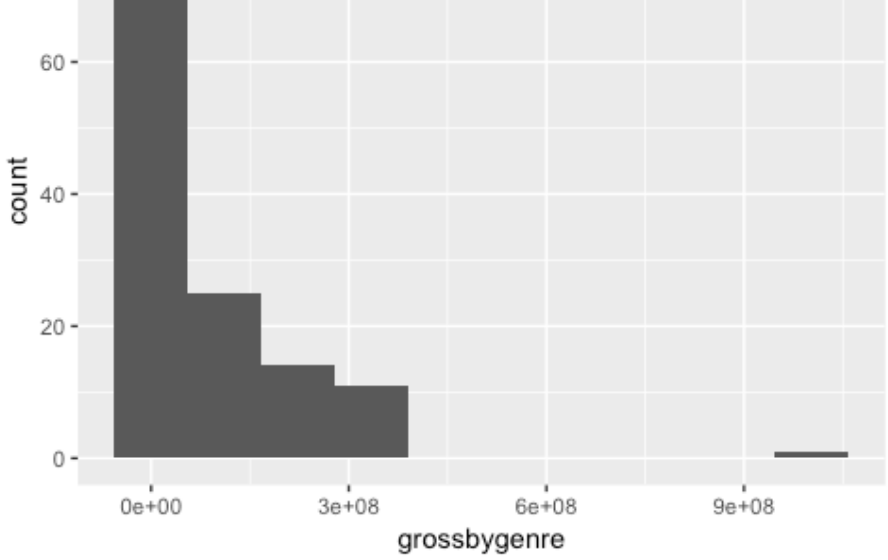
range skew kurtosis se
X1 890875303 2.68 10.71 8974751
Warning: Removed 853 rows containing non-finite values (stat_bin).



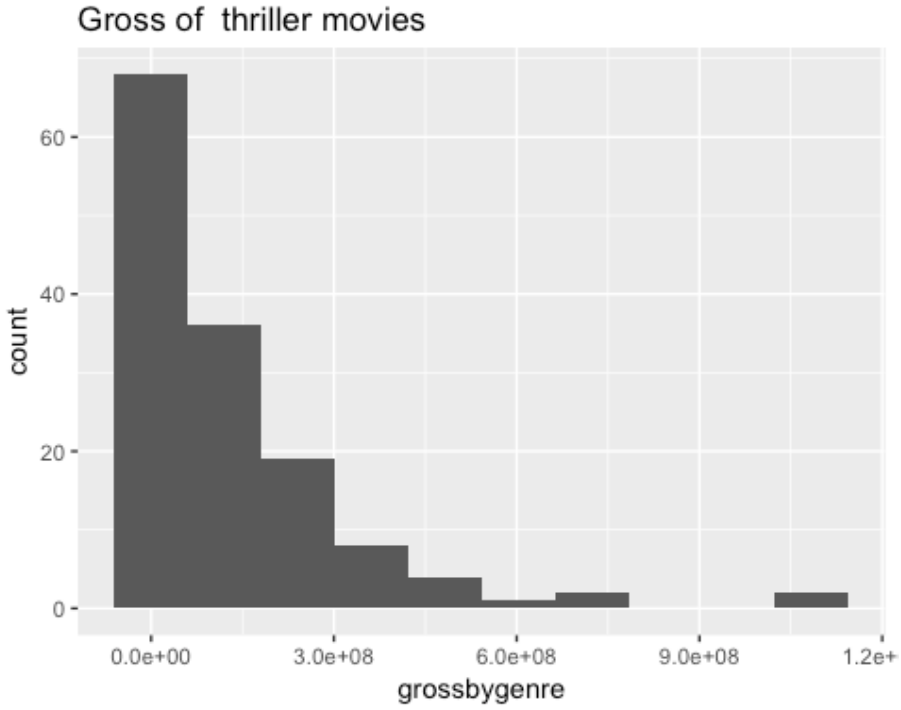
action
vars n mean sd median trimmed mad min
X1 1 322 260145067 285128204 180470010 208317297 226944552 0
max range skew kurtosis se
X1 1670328025 1670328025 1.85 4.08 15889574
Warning: Removed 758 rows containing non-finite values (stat_bin).



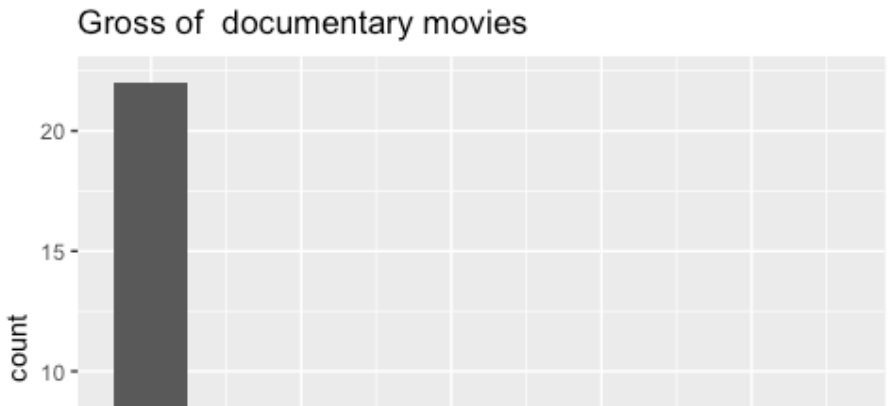
crime
vars n mean sd median trimmed mad min max
X1 1 132 86988614 128877116 30319161 63112069 43808832 0 1002891358
range skew kurtosis se
X1 1002891358 3.24 17.51 11217313
Warning: Removed 688 rows containing non-finite values (stat_bin).

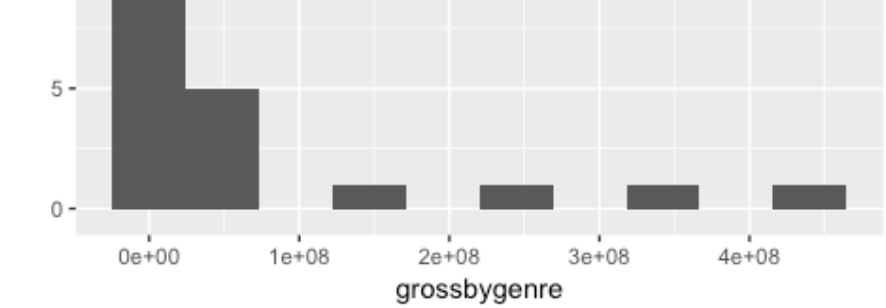


thriller
vars n mean sd median trimmed mad min max
X1 1 140 131790476 184700103 75352146 94125383 109605235 0 1084439099
range skew kurtosis se
X1 1084439099 2.62 8.72 15610008
Warning: Removed 525 rows containing non-finite values (stat_bin).

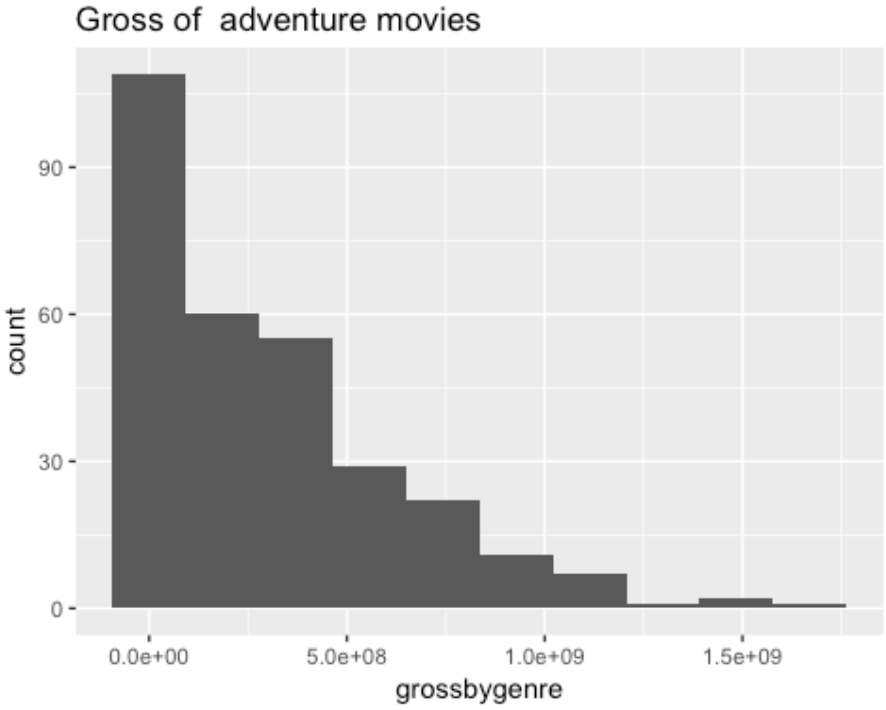


documentary
vars n mean sd median trimmed mad min max
X1 1 31 46654792 106234429 4731944 16755479 7010081 0 440160956
range skew kurtosis se
X1 440160956 2.64 5.99 19080267
Warning: Removed 497 rows containing non-finite values (stat_bin).

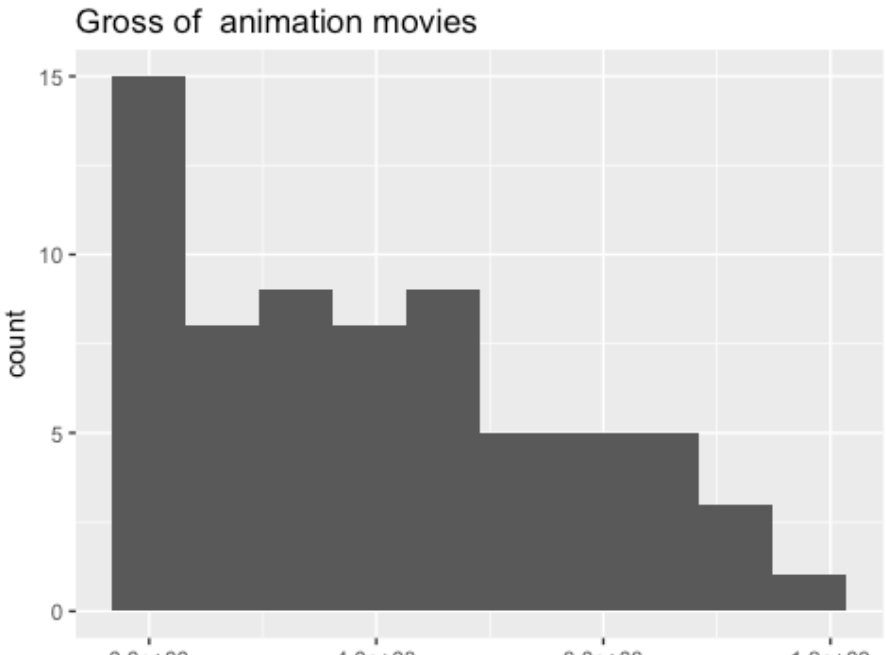


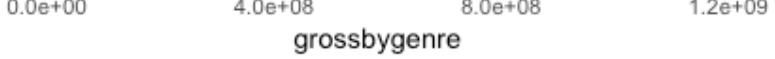


adventure
vars n mean sd median trimmed mad min
X1 1 297 304908661 315130184 218853353 256739260 279114335 0
max range skew kurtosis se
X1 1670328025 1670328025 1.32 1.67 18285708
Warning: Removed 501 rows containing non-finite values (stat_bin).



animation
vars n mean sd median trimmed mad min
X1 1 68 397972795 322387018 345225211 374245683 407027791 0
max range skew kurtosis se
X1 1163624481 1163624481 0.49 -0.9 39095169
Warning: Removed 531 rows containing non-finite values (stat_bin).





A: Given that summer months gross the most money, if I were a Hollywood producer what genre movie should I make?

I hypothesize that adventure and drama categories will have the greatest gross revenue in the summer peak months. Consumers want something exciting to watch in their free time in the summer.

Assumption: I used the same top categories from the initial analysis.

By volume of movies released in summer the order is the following: comedy, drama, action, and adventure.

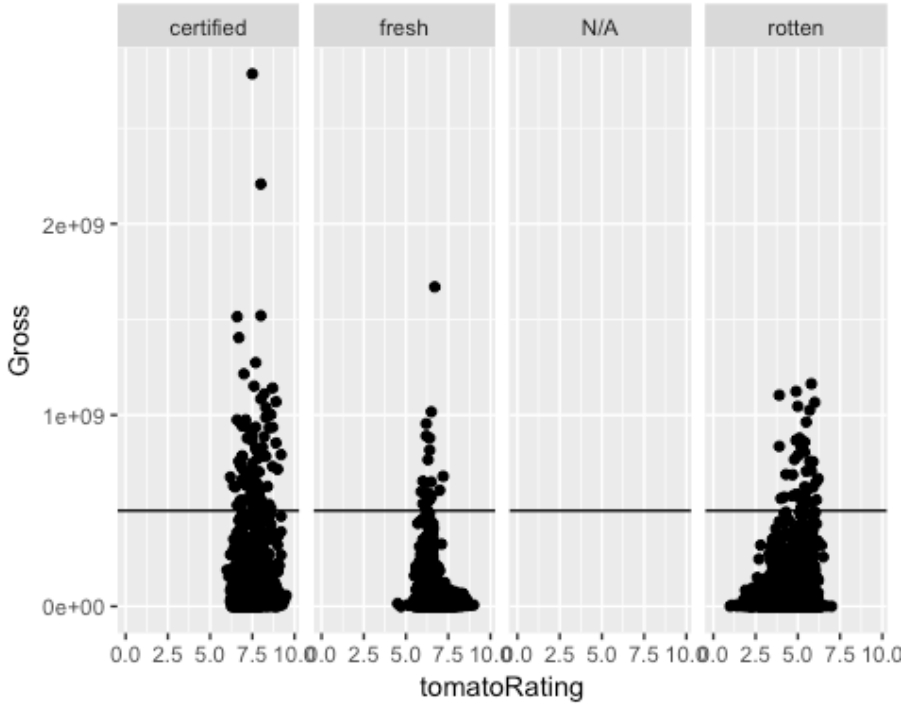
However after reviewing the plots, it suggests that best genres to invest a movie in the summer are comedy, action, and drama. We can conclude this by conducting visual comparison of where the graphs are more concentrated to the right versus closer to zero gross profit.

10. Unexpected insight

Come up with one new insight (backed up by data and graphs) that is unexpected at first glance and do your best to motivate it. Same instructions apply as the previous task.

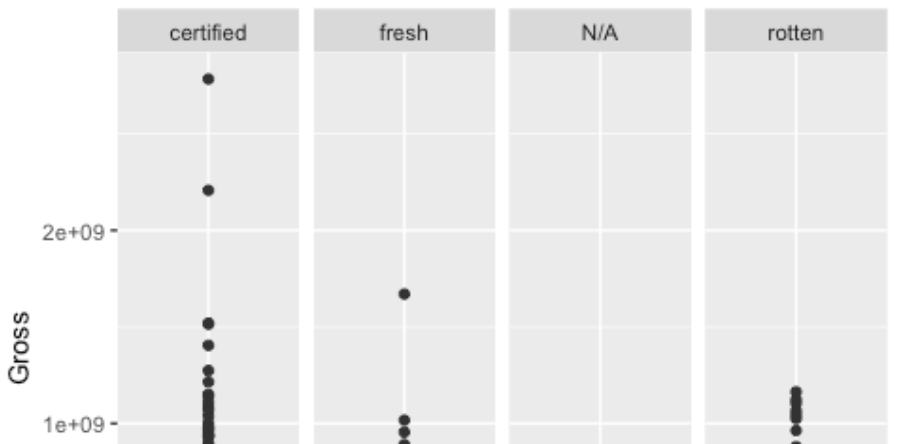
```
# TODO: Find and illustrate one unexpected insight
p_9_2 <- ggplot(df, aes(tomatoRating, Gross))
p_9_2+geom_point() + facet_grid(.~tomatoImage) + geom_hline(yintercept = 500000000)
```

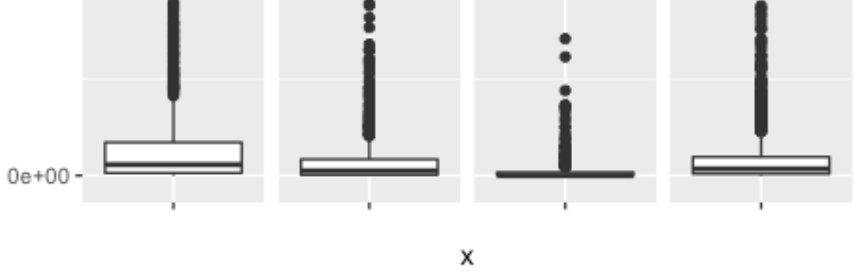
```
## Warning: Removed 29211 rows containing missing values (geom_point).
```




```
p_9_2_2 <- ggplot(df, aes("", Gross))
p_9_2_2+geom_boxplot() + facet_grid(.~tomatoImage)
```

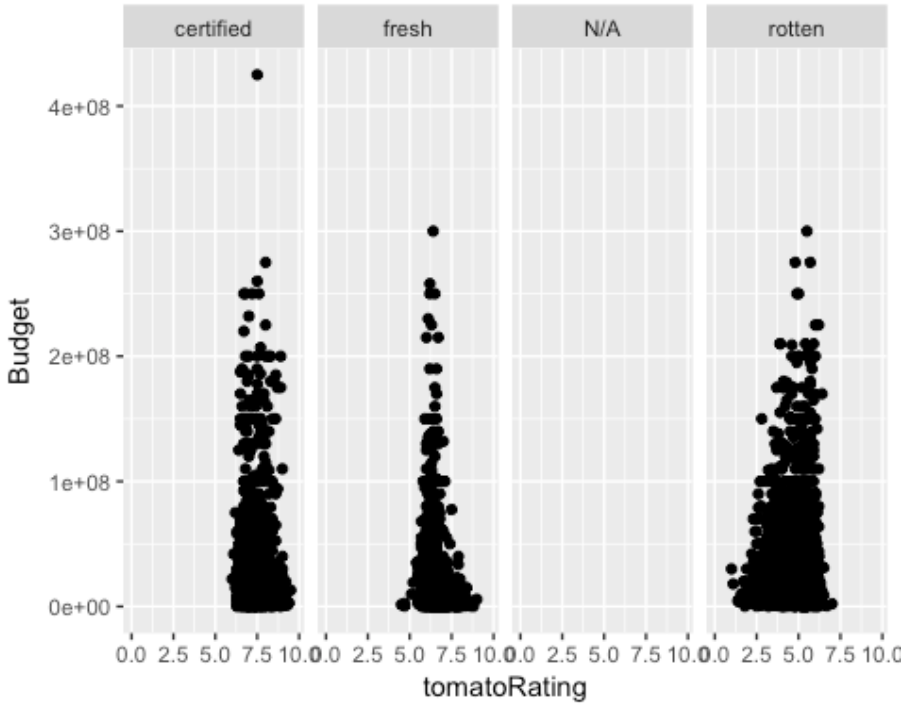
```
## Warning: Removed 28810 rows containing non-finite values (stat_boxplot).
```





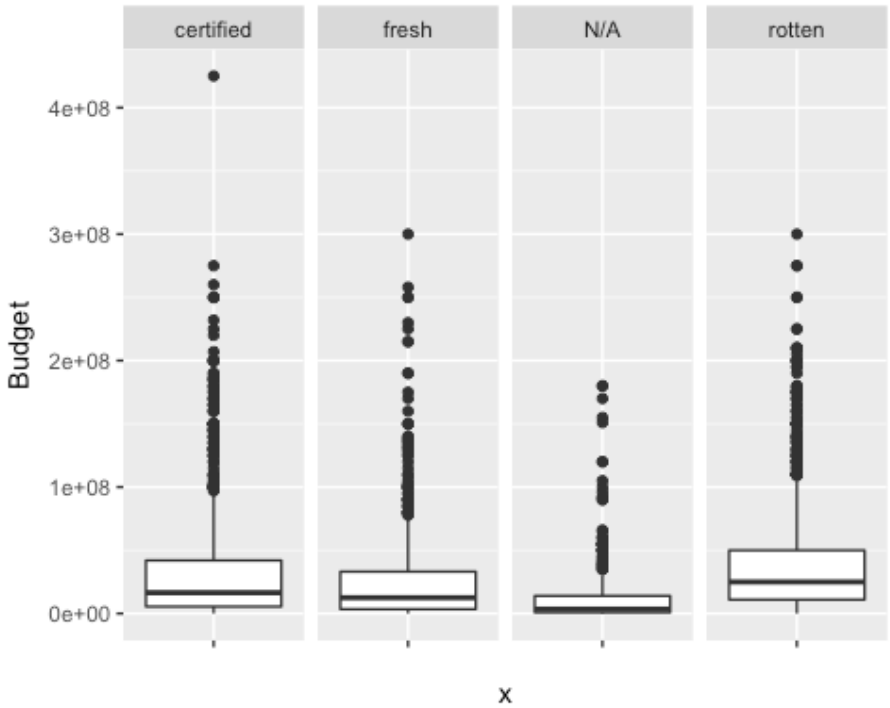

```
p_9_3 <- ggplot(df, aes(tomatoRating, Budget))
p_9_3+geom_point() + facet_grid(.~tomatoImage)
```

Warning: Removed 29211 rows containing missing values (geom_point).




```
p_9_3_2 <- ggplot(df, aes("", Budget))
p_9_3_2+geom_boxplot() + facet_grid(.~tomatoImage)
```

Warning: Removed 28810 rows containing non-finite values (stat_boxplot).



Q: Unexpected insight. How do the movies perform as it pertains to ratings vs gross revenue, budget in the rotten tomatoes categories of Certified, Fresh, and Rotten?

A: The Rotten tomatoes categories are an award given to movies that attain a certain rating. I wanted to examine if movies in these segments have any unique attributes

I wanted to examine if movies in these segments have any unique attributes.

I started my analysis looking at rating vs gross rev. It appears that certified movies are more profitable but have similar shapes however the fresh and rotten distributions seem to widen at the base. This means that potentially more movies in these categories make less money. When we look at the count of movies that made more than $500M (the horizontal line) It looks like certified has the most, followed by rotten, then, fresh. This is counter-intuitive and suggests that not highly rated movies can man make just as much money as highly rated (certified)

The boxplots show the IQR of certified > the other categories suggesting certified movies gross more but each category seems to have just as many outliers. This suggests that despite critic review, a movie can be blockbuster.

In addition, the budget breakout graph shows that rotten, fresh, and certified movies look to be equally expensive, however, there seems to be more lower budget rotten films. The boxplot shows that generally movies of each category are distributed the same.