Movie Reviews

Sandeep V

October,12, 2017

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(rvest)

## Loading required package: xml2

library(tidyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
```

Frame

- 5 graphs for movie rating xlsx
- Correlation between audience and critic ratings evolved throughout year by genre ## Acquire Data Getting data from Movies-ratings.xslx

```
setwd("C:/Users/vasistas/Documents/From_Mydownloads/PGP-BDA/In_class/2nd_Residency/SVAP_Amit/Assignment/Subjective_quiz")
movie_ratings=read.csv("Movie-Ratings.csv",header=TRUE)
Remove unwanted data
```

- Format data types
- Missing data

```r
dim(movie_ratings)
```

```
## [1] 562   6
```

```r
str(movie_ratings)
```

```
## 'data.frame':    562 obs. of  6 variables:
##  $ Film                    : Factor w/ 562 levels "(500) Days of Summer "
,..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Genre                   : Factor w/ 7 levels "Action","Adventure",..:
3 2 1 2 3 1 3 5 3 3 ...
##  $ Rotten.Tomatoes.Ratings..: int  87 9 30 93 55 39 40 50 43 93 ...
##  $ Audience.Ratings..      : int  81 44 52 84 70 63 71 57 48 93 ...
##  $ Budget..million...      : int  8 105 20 18 20 200 30 32 28 8 ...
##  $ Year.of.release         : int  2009 2008 2009 2010 2009 2009 2008 2007
2011 2011 ...
```

```r
column_name <- c('Film','Genre','Rot','Aud','Budget','Year')
colnames(movie_ratings)<-column_name
str(movie_ratings)
```

```
## 'data.frame':    562 obs. of  6 variables:
##  $ Film  : Factor w/ 562 levels "(500) Days of Summer ",..: 1 2 3 4 5 6 7
8 9 10 ...
##  $ Genre : Factor w/ 7 levels "Action","Adventure",..: 3 2 1 2 3 1 3 5 3 3
...
##  $ Rot   : int  87 9 30 93 55 39 40 50 43 93 ...
##  $ Aud   : int  81 44 52 84 70 63 71 57 48 93 ...
##  $ Budget: int  8 105 20 18 20 200 30 32 28 8 ...
##  $ Year  : int  2009 2008 2009 2010 2009 2009 2008 2007 2011 2011 ...
```

Explore

```r
library(ggplot2)
library(RColorBrewer)
library(caTools)
```

```r
summary(movie_ratings)
```

```
##                     Film           Genre          Rot
##  (500) Days of Summer :  1   Action   :154   Min.   : 0.0
##  10,000 B.C.          :  1   Adventure: 29   1st Qu.:25.0
##  12 Rounds            :  1   Comedy   :172   Median :46.0
##  127 Hours            :  1   Drama    :101   Mean   :47.4
##  17 Again             :  1   Horror   : 49   3rd Qu.:70.0
##  2012                 :  1   Romance  : 21   Max.   :97.0
##  (Other)              :556   Thriller : 36
##       Aud             Budget           Year
##  Min.   : 0.00   Min.   :  0.0   Min.   :2007
##  1st Qu.:47.00   1st Qu.: 20.0   1st Qu.:2008
##  Median :58.00   Median : 35.0   Median :2009
##  Mean   :58.83   Mean   : 50.1   Mean   :2009
```

```
##  3rd Qu.:72.00   3rd Qu.: 65.0   3rd Qu.:2010
##  Max.   :96.00   Max.   :300.0   Max.   :2011
##
```

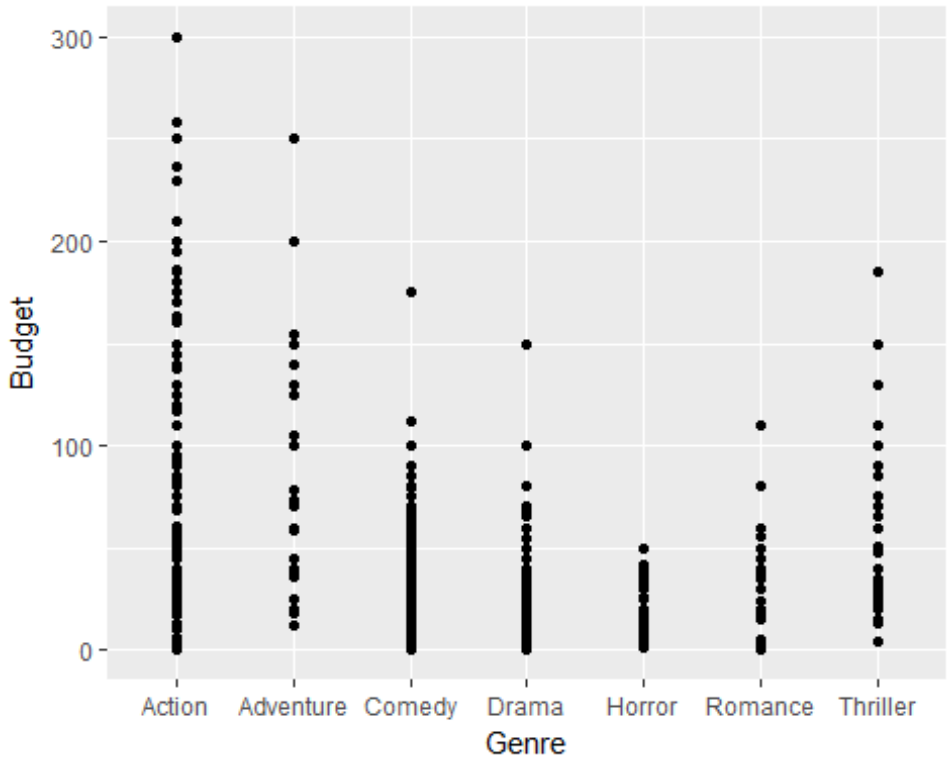
```
#structure and data types will be provided by str functions
str(movie_ratings)
```

```
## 'data.frame':    562 obs. of  6 variables:
##  $ Film  : Factor w/ 562 levels "(500) Days of Summer ",..: 1 2 3 4 5 6 7
8 9 10 ...
##  $ Genre : Factor w/ 7 levels "Action","Adventure",..: 3 2 1 2 3 1 3 5 3 3
...
##  $ Rot   : int  87 9 30 93 55 39 40 50 43 93 ...
##  $ Aud   : int  81 44 52 84 70 63 71 57 48 93 ...
##  $ Budget: int  8 105 20 18 20 200 30 32 28 8 ...
##  $ Year  : int  2009 2008 2009 2010 2009 2009 2008 2007 2011 2011 ...
```

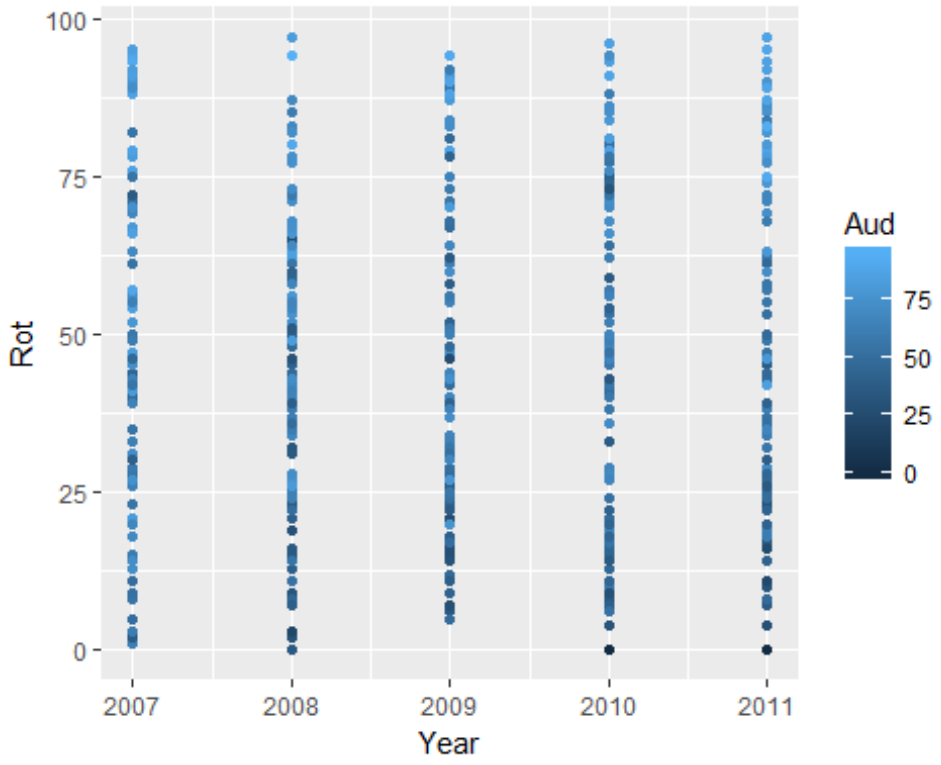
```
ggplot(movie_ratings, aes(x=Genre, y=0)) + geom_jitter() + scale_y_continuous
(limits = c(-2,2))
```


```
# The above plot is known as stripchart which is a univariate plot
# We can see Sction, comedy and Drama are dense while Romance and Adventure h
ave less dense distribution of films
```

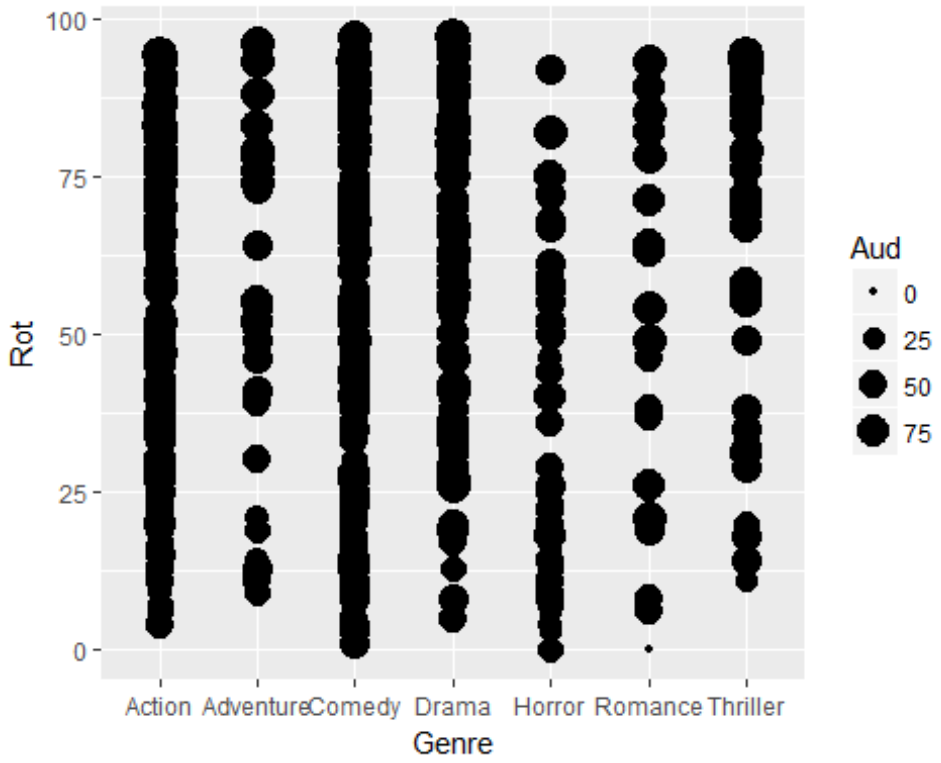
```
ggplot(movie_ratings, aes(x = Genre, y = Budget)) + geom_point()
```



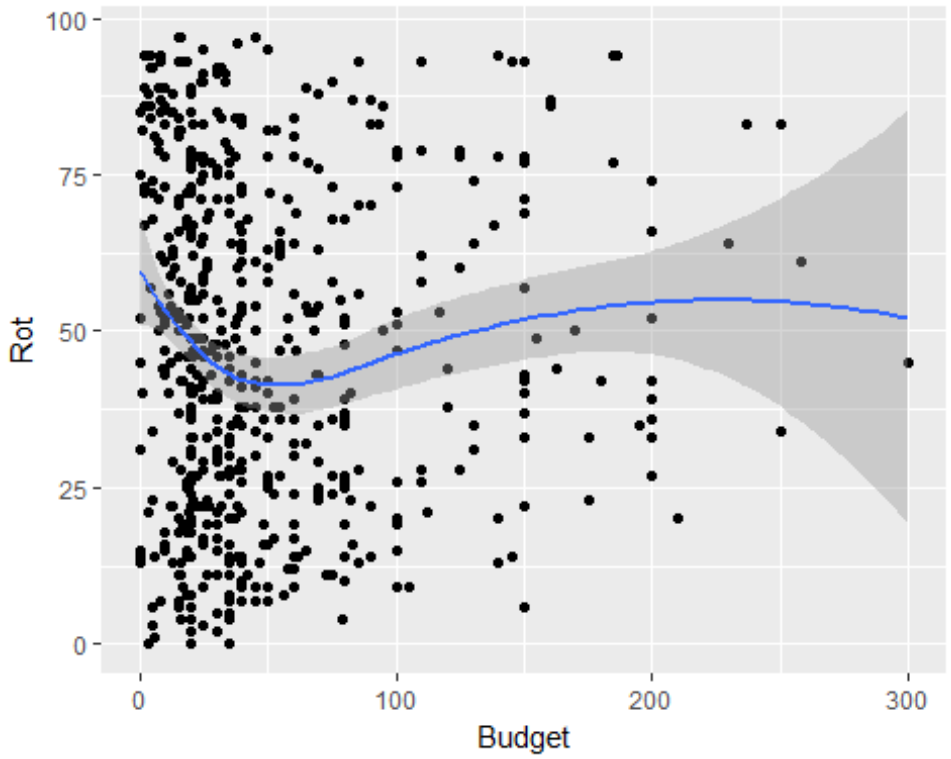

```
# if we observe the dataset movie_ratings we will get to know
# that the variable Genre is categorical in nature
# So we will need to tell ggplot2 that Genre is a categorical variable.
# We can see Highest budget movie is action and Horror combines for least bud
get.

ggplot(movie_ratings, aes(x = Year, y = Rot, color = Aud)) + geom_point()
```



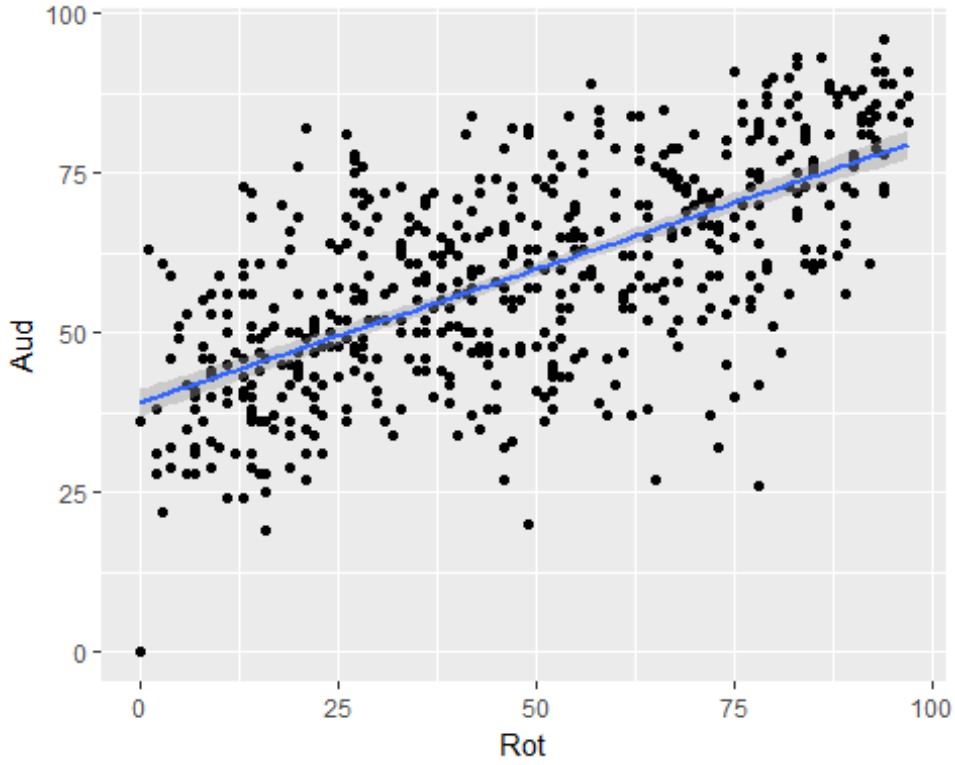

```
# The above plot shows relationship between critic rating and Audience rating
with Year realesed of the movie_ratings
# with varying critic rating of the movie rating Audience rating shown in dif
ferent colors.

ggplot(movie_ratings, aes(x = Genre, y = Rot, size = Aud)) + geom_point()
```



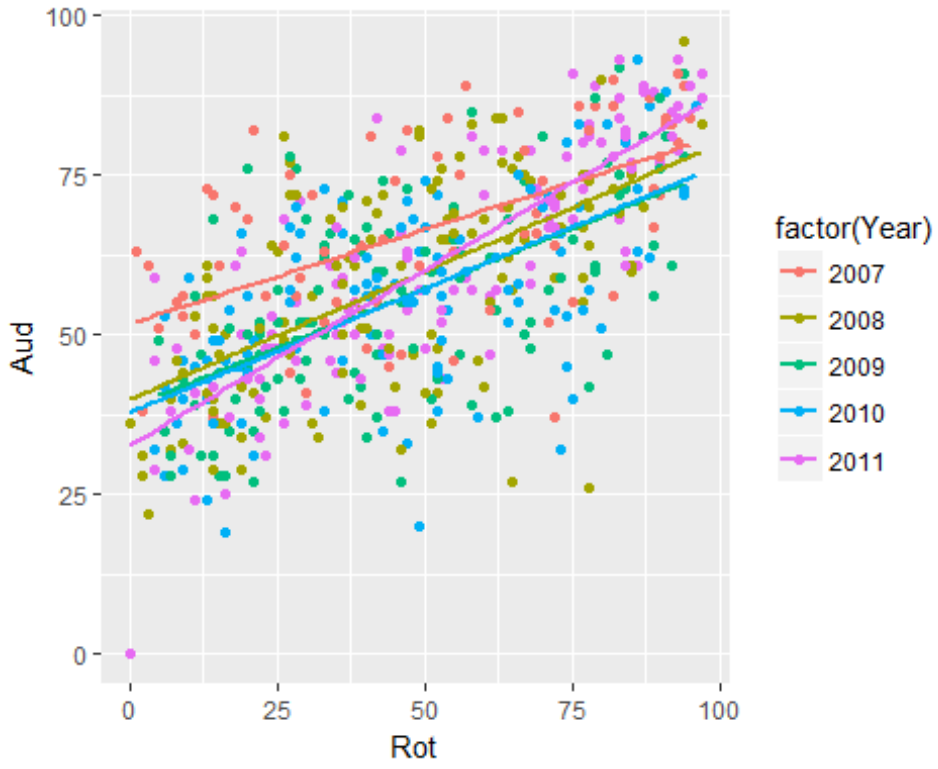

```
#This plot also same as above, but this Audience ratings of
#Movies is shown with varying sizes for Fenre type

ggplot(movie_ratings, aes(x = Budget, y = Rot)) +
  geom_point() + geom_smooth()

## `geom_smooth()` using method = 'loess'
```




```
# Smoothing means to use algorithms to remove noise from a data set,
# allowing some important patterns to stand out.
# To add smoothing lines we would the geom geom_smooth() by default
# it uses LOESS smoothing which stands for Locally Weighted Scatterplot Smoot
hing

ggplot(movie_ratings, aes(x = Rot, y = Aud)) +
  geom_point() + geom_smooth(method = "lm")
```




```
# If we want to change the previous plot to use
# ordinary linear model smoothing we can use the method = "lm" argument.
# The shaded portion in the above plots shows the 95% Confidence Intervals
# which also known as the standard error, we can remove this shaded portion
# using the argument se = FALSE

ggplot(movie_ratings, aes(x = Rot, y = Aud, col = factor(Year))) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE)
```




```
# Sometimes in our data we might like to see patterns in the
# data based on some subgroups or categorical variables which
# can be shown using the aesthetic col
# In the above ggplot command our smooth is calculated for each
# subgroup because there is an invisible aesthetic group which inherits from
col.

ggplot(movie_ratings, aes(x = Rot, y = Aud, col = factor(Year), fill = factor
(Genre))) +
  geom_point(shape = 21, size = 5, alpha = .5)
```


```
# The above plot is used whenever we need to distinguish the
# data points based on four categorical variables - Audience rating, Rotten t
omatoes, Genre and Year
```