Data Visualiztion via ggplot2

Yeon Mi Hwang

10/28/2017

Data Visualiztion via ggplot2

[Introduction]

Have you tried to draw out plots using R base function?
If you have done, how was the result like? Did coding took a while?
Are you satisfied with the aesthetics of the plot?
If you said no to any of these two questions, it is the time to try `ggplot2` to visualize the data! `ggplot2` is a plotting system for R software.
It takes care of cumbersome details which you have to care in R base system.
It is one of the most powerful tools to create graphic models.
I am writing this blog to introduce you to `ggplot2` ,
so you can utilize it later when you have to visualize your data.
This blog is going to recap what we have learned in the class(the very basics of `ggplot2` . types of `ggplot2` graphs) and finally, I will talk about the cool features of `ggplot2` which we did not go over during the class or lab.

[Motivations to use ggplot2]

I brielfy mentioned about the advantage of `ggplot2` in the introduction, and now in this section I am going to persuade you in earnest to use ggplot. First of all, let's first check out the disadvantage of base graphics provided by R.

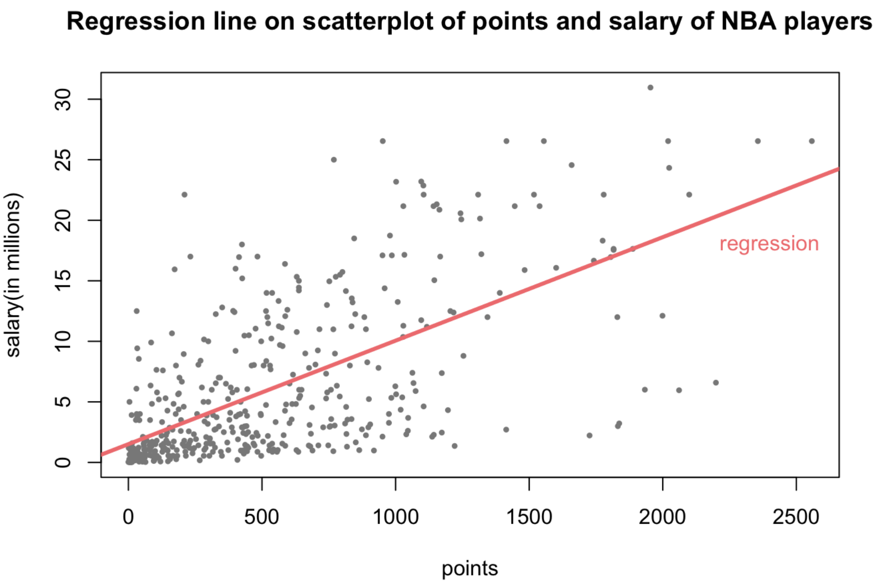
1) Disadvantage of R base graphics(reasons to stop use it!)

- ugly graphics
- laborious procedure
- verbose language and grammar . It is hard to use!

Following is the example of code chunk for R base graphics.

```
dat=read.csv("/Users/hwangyeonmi/nba2017-player-statistics.csv")
```

```
# example graph using R base system
Points=3*dat$Points3+2*dat$Points2+dat$FTM
salaryinmil=dat$Salary/1000000
plot(Points,salaryinmil,
     xlab="points",
     ylab="salary(in millions)",
     col="gray54",bg="gray54",pch=16,cex=0.6,
     main ="Regression line on scatterplot of points and salary of NBA players")
abline(1.509, 0.00855, untf=FALSE, col="lightcoral", lwd=3)
text(2400,18,"regression",col="lightcoral")
```



2) Advantage of ggplot graphics

- it follows a grammar(called grammar of graphics), just like any language
- grammar defines components in a plot
- support a continuum of expertise
- publication quality figure

- effortless!
- very flexible
- many users
- good looking graphics!

reference : [link] (http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html)

[Basics of ggplot2]

1) Terminology

Before we dig in to the world of `ggplot`, let me provide some terminologies that you should be used to when using ggplot2

- `ggplot` : main component or function in which you specify the dataset to be visualized, and also the variables of the plot. (caution! it is not ggplot2)
- `geoms` : specify geometric objects
  for example) `geom_point()`, `geom_bar()`, `geom_line()`
- `aes` : specify aesthetics. you can choose the shape, transparency, color, fill, linetype here!
- `scales` : compenent that define how your data will be plotted.

2) installation of ggplot2

- install `ggplot2` packages.

```
install.packages("ggplot2",dependencies = TRUE)
```

- and don't forget to load it. I loaded `dplyr` also to ease the process of data manipulation.

```
library(dplyr)
```

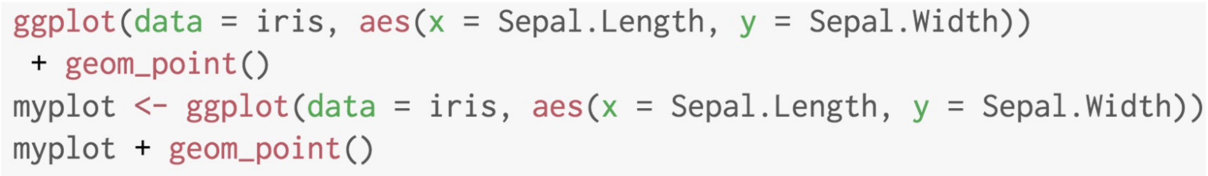
```
##
## Attaching package: 'dplyr'
```

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

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

```
library(ggplot2)
```

3) basic structure



```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))
 + geom_point()
myplot <- ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))
myplot + geom_point()
```

The above image simply displays the basic structure and component of ggplot function.
1) Inside the ggplot2 main function `ggplot`, you have to first specify which data set you are going to plot.

2) In addition to that, you have to specify the aesthetics of the graph. In `aes=` variable, as you can see, you have to specify what is going to be on the x axis and y axis.

3) add some layers of geometric objects! statistical models! and panels!

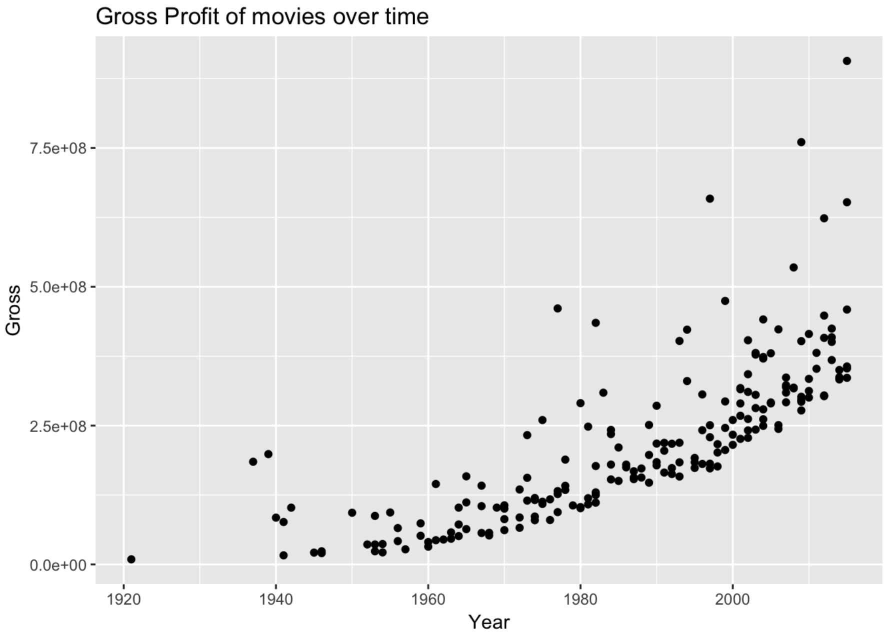
[Types of graph]

1) scatterplot

```
movie1=read.csv("~/stat133/stat133-hws-fall17/post01/data/top_movies_by_title.csv")
movie=select(movie1,Title,Studio,Gross,Year)
```

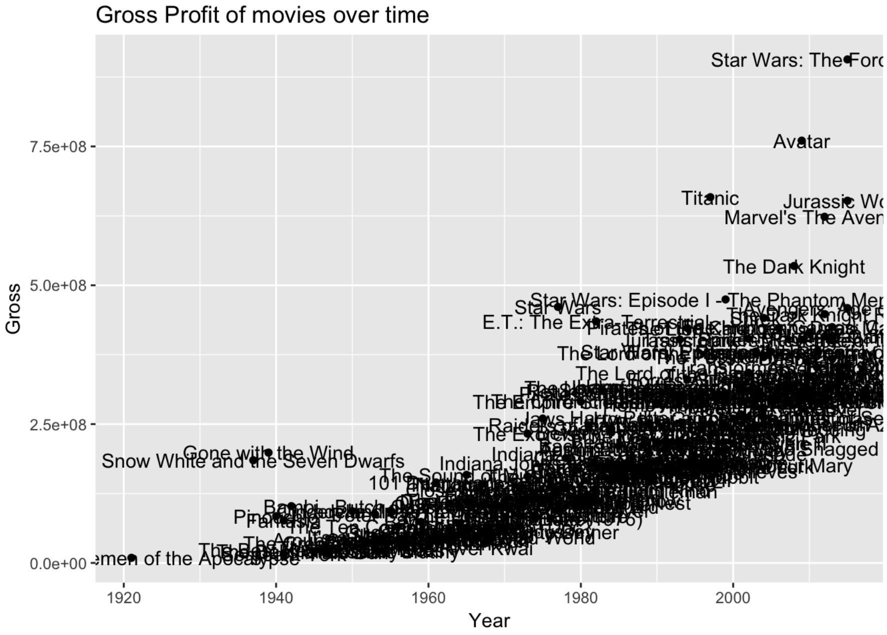
data set reference (https://drive.google.com/a/berkeley.edu/file/d/0B3lTGuQsifiJcUtEQ3R0UUlmRGc/view?usp=sharing)

```
# scatter plot of gross profit of movies over time
movplot=ggplot(movie,aes(Year,Gross))+geom_point()+labs(title="Gross Profit of movies over time")
movplot
```



- `ggplot` can produce a scatterplot by using additional function `geom_point()` as above.
- You can add label to above plot by simply adding another function `geom_text()` My example is not a good example because all the data are pretty packed together, causing overlaps of labels. However, do you get the idea of adding another component `geom_text()` to the previous graph?

```
# scatterplot of gross profit of movies over time with label
movplot+geom_text(aes(label=Title))
```



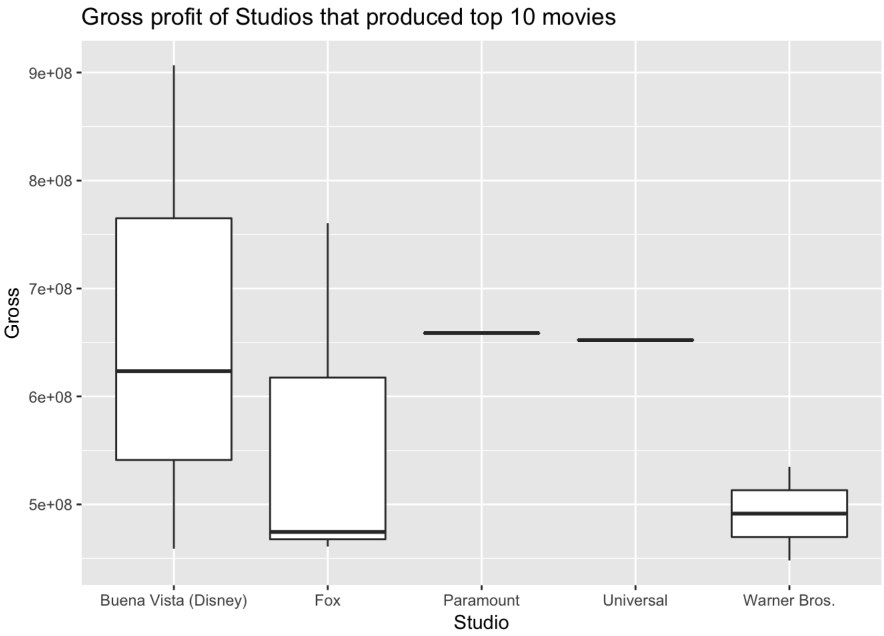
2) boxplot

You can create boxplot by adding `geom_boxplot()` to the main function. Since there are so many catergorical value within categorical variable ‘Studio’, I trimmed the data for simpler outlook of boxplot.

```
# trimmed version of movie data. top 10 movie in terms of gross profit
topmovie=slice(arrange(movie, desc(Gross)),1:10)
topmovie
```

```
## # A tibble: 10 x 4
##                                         Title               Studio
##                                        <fctr>               <fctr>
##  1             Star Wars: The Force Awakens Buena Vista (Disney)
##  2                                   Avatar                  Fox
##  3                                  Titanic            Paramount
##  4                           Jurassic World            Universal
##  5                    Marvel's The Avengers Buena Vista (Disney)
##  6                          The Dark Knight         Warner Bros.
##  7 Star Wars: Episode I - The Phantom Menace                  Fox
##  8                                Star Wars                  Fox
##  9                  Avengers: Age of Ultron Buena Vista (Disney)
## 10                    The Dark Knight Rises         Warner Bros.
## # ... with 2 more variables: Gross <int>, Year <int>
```

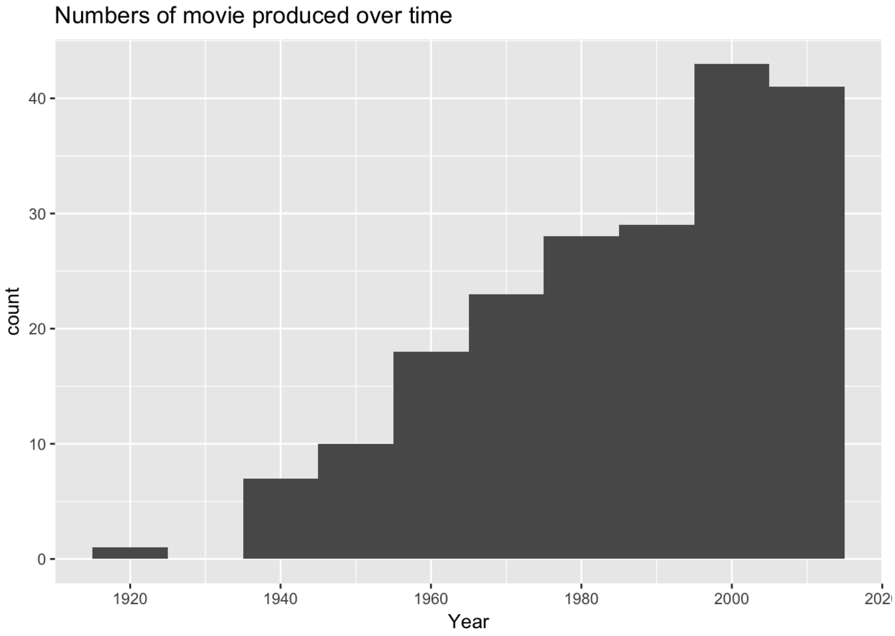
```
#Boxpot of gross profit of studios that produced top 10 movies
ggplot(topmovie,aes(x=Studio,y=Gross))+
  geom_boxplot()+
  labs(title="Gross profit of Studios that produced top 10 movies")
```



3) histogram

- We also can create histogram using ggplot! You just have to add `geom_histogram()` component to the main function `ggplot()`. `binwidth=` variable specifies the number of bin.
  Following graph displays a histogram of number of movies released over time.

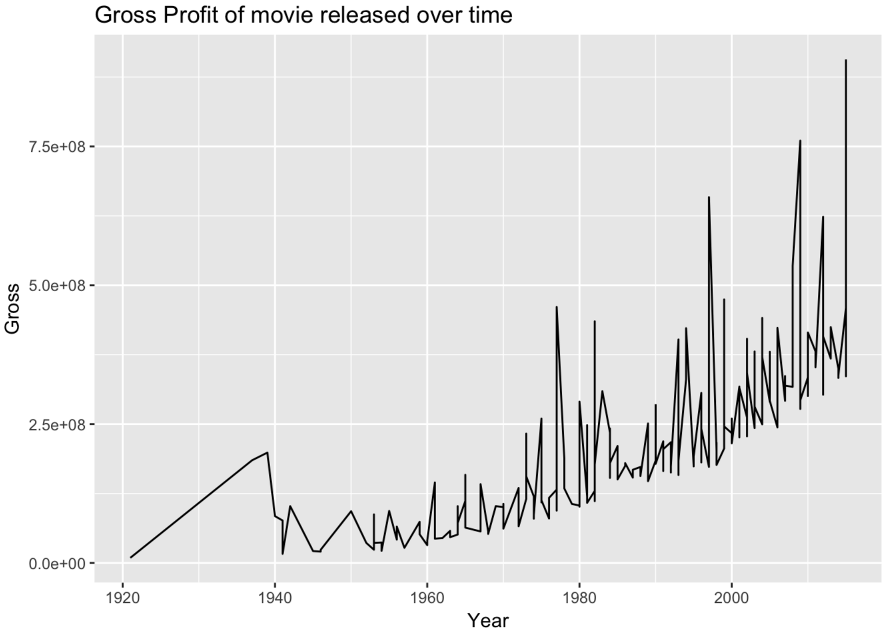
```
# histogram that displays numbers of movie produced over time
ggplot(movie, aes(x=Year))+geom_histogram(binwidth=10)+labs(title="Numbers of movie produced over time")
```



4) Line plot

- Similar to boxplot, you can easily create line plot using `geom_line()`

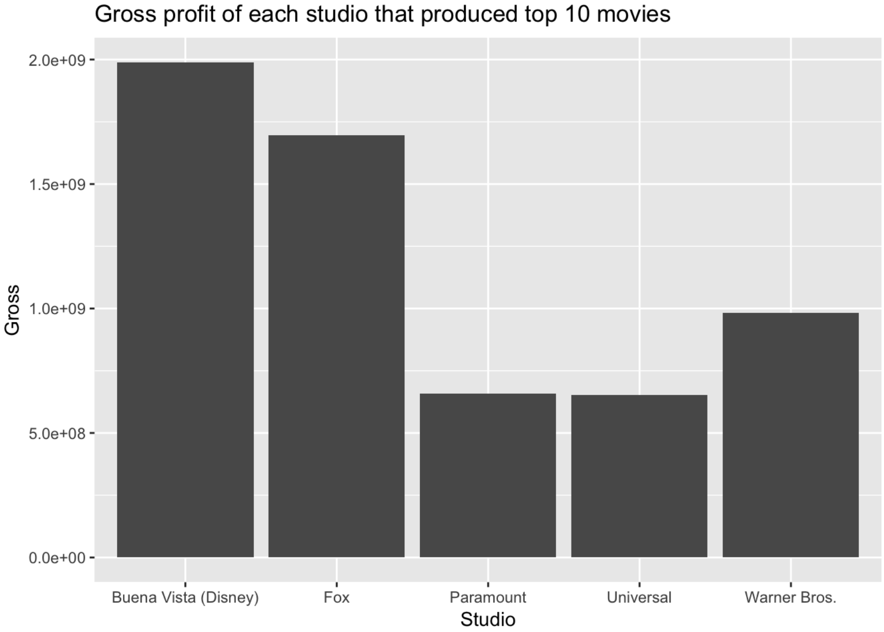
```
# line plot of gross profit of movie released over time
ggplot(movie, aes(x=Year,y=Gross))+geom_line()+labs(title="Gross Profit of movie released over time")
```



5) Bar plot

- bar chart can be produced using `geom_bar()`
- Similar to the boxplot, I am using the trimmed version of dataset in order to make the outlook of the graph simple. The trimmed version of dataset includes only data of top 10 movies(in terms of gross profit)

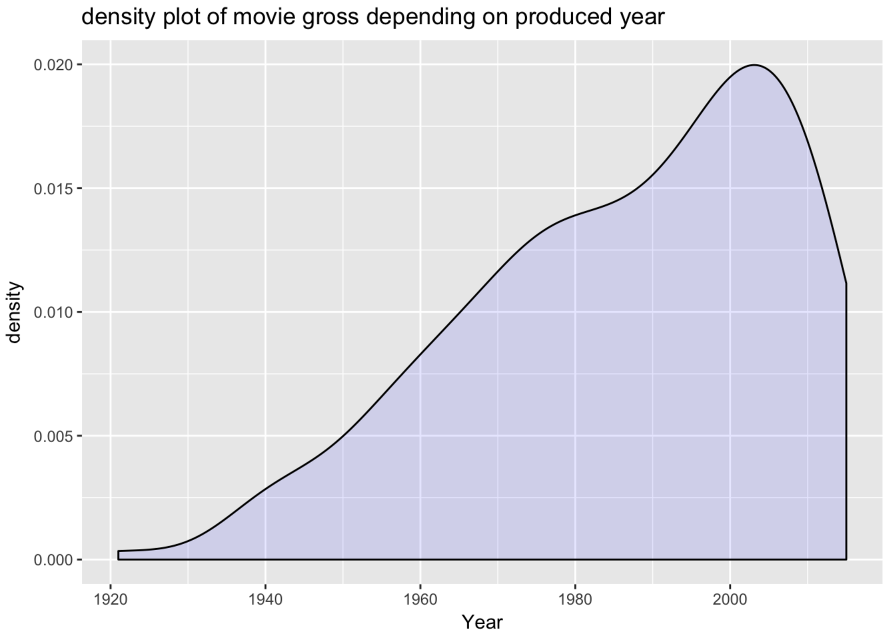
```
ggplot(topmovie,aes(x=Studio,y=Gross))+
  geom_bar(stat="identity")+
  labs(title="Gross profit of each studio that produced top 10 movies ")
```



6) Density plot

density plot can be produced using `geom_density()`

```
# density plot of number of movies released each year.
ggplot(movie, aes(Year))+geom_density(fill="blue",alpha=0.1)+labs(title="density plot of movie gross depending on
produced year")
```

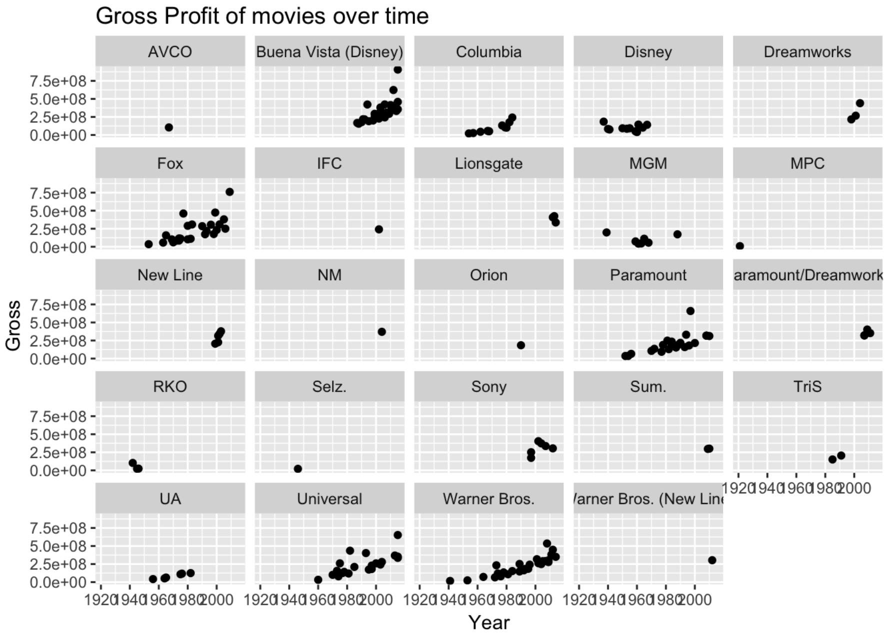


[Cool features of ggplot]

1) faceting

I would like to introduce you to one of the coolest feature of `ggplot`. Ggplot has a capability to display multiple facet.
Facet is basically dividing the major plot into subplot based on the categorical variable. With our data set, the categorical variable that can group individual movies is 'studio'. Let's look at the example.

```
# facet
movplot+facet_wrap(~Studio)
```



Remember, `movplot` was a scatter plot of gross profit of individual movies over time (x axis is Year, y axis is gross) Each Studio produce several movies. In other words, individual movies can be grouped depending on the producing studio. Above plot displays subplot for each studio. Faceting cannot be done if there is no categorical variable.

2) Theme

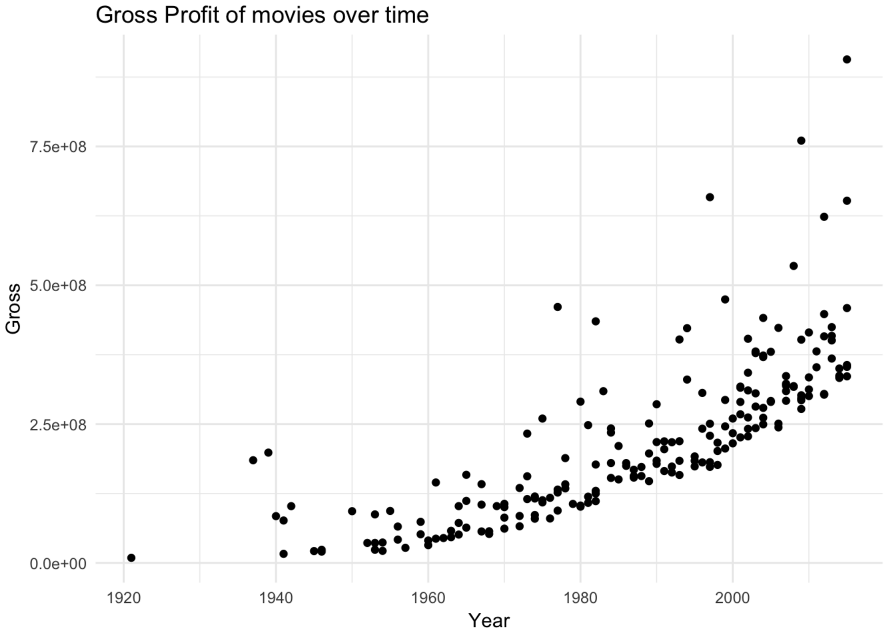
As mentioned in the section of 'advantage of ggplot', `ggplot` produces good looking graphics pretty effortlessly. This can be done thanks to 'theme' of ggplot.
Theme system can handle non-data element.
Non-data element include such things like:

- axis labels
- plot background
- facet label background
- legend appearnce

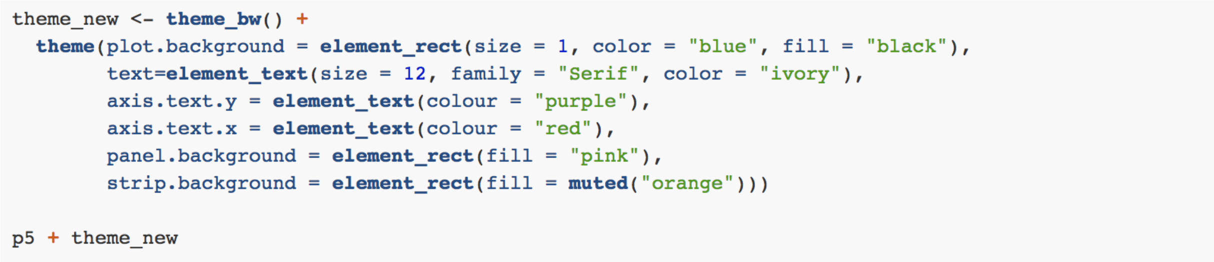
gray theme is the default theme.(the background color of the plot is gray) there are other bulit-in themes including

- `theme_bw()`
- `theme_classc()`
- `theme_minimal()`

```
plottobesaved=movplot+theme_minimal()
plottobesaved
```



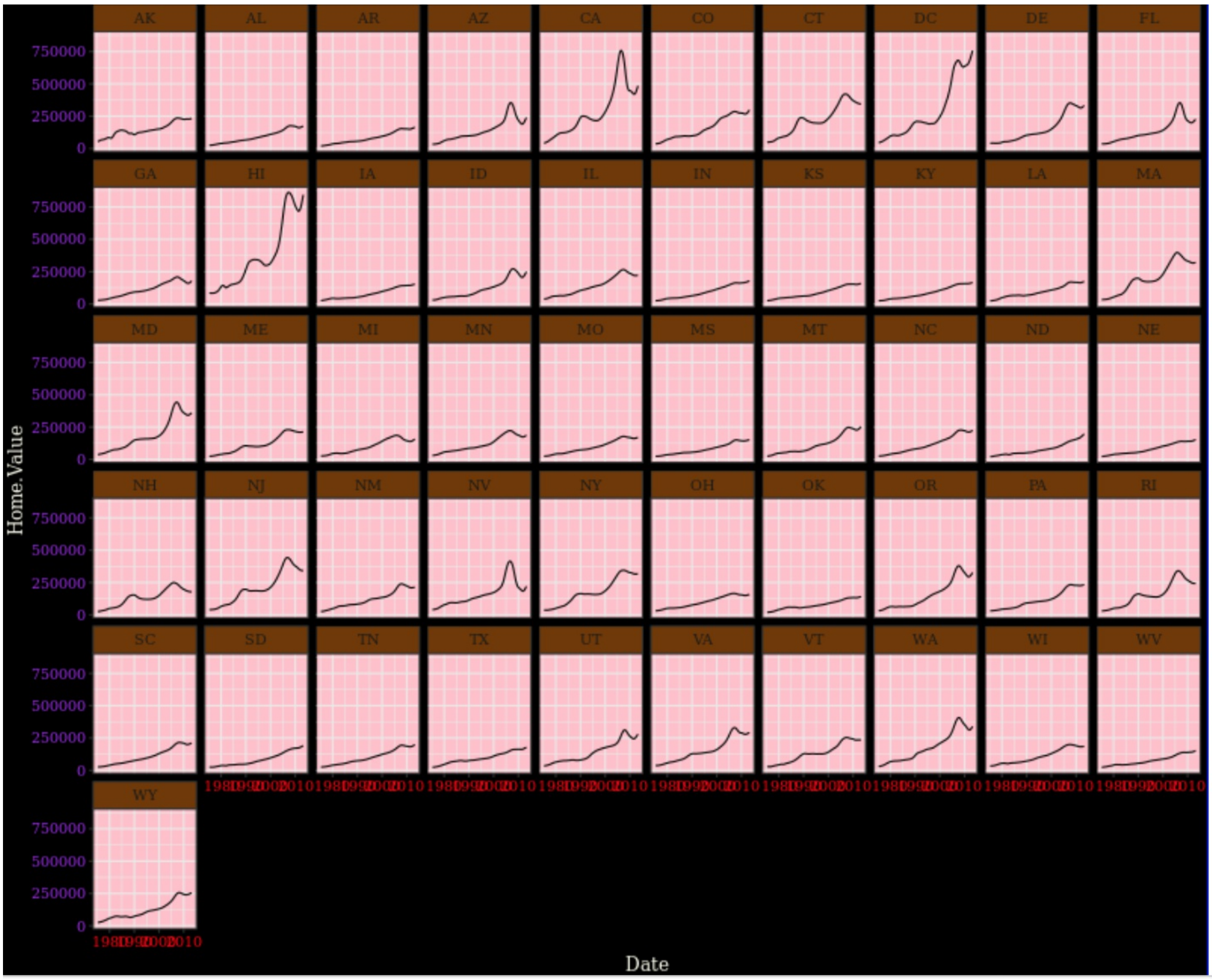
You can even create your own theme!
example code chunk for creating new theme:



```
theme_new <- theme_bw() +
  theme(plot.background = element_rect(size = 1, color = "blue", fill = "black"),
        text=element_text(size = 12, family = "Serif", color = "ivory"),
        axis.text.y = element_text(colour = "purple"),
        axis.text.x = element_text(colour = "red"),
        panel.background = element_rect(fill = "pink"),
        strip.background = element_rect(fill = muted("orange")))

p5 + theme_new
```

output of above code chunk :



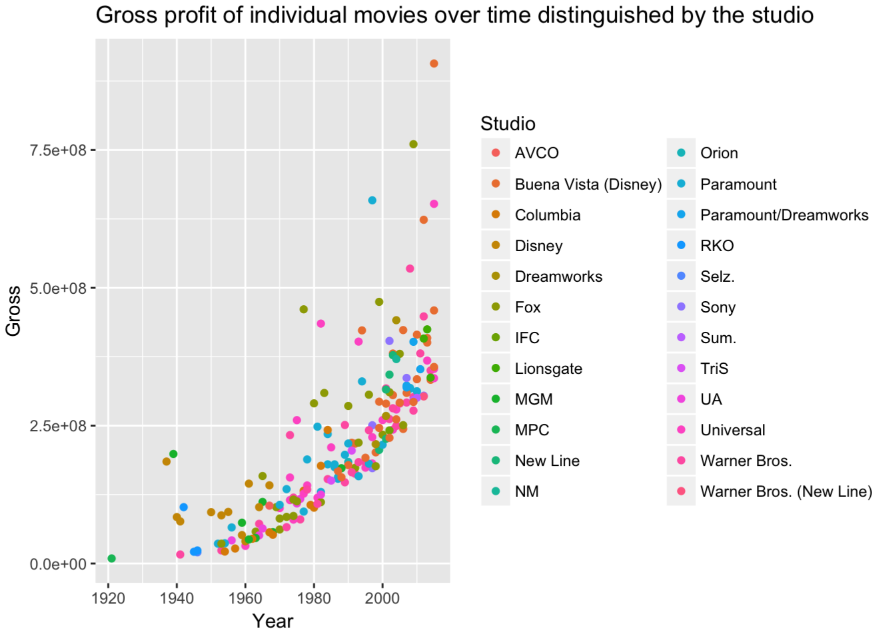
3) Scale

This is a topic not covered during the lecture, but I think this is the coolest part of `ggplot` feature.
There are two types of scale, one is discrete scale and the other is continuous scale.

1.Discrete scale

As the word ‘discrete’ suggests, discrete scale can be applied for categorical value. If you add discrete scale to a normal scatter plot we had, we can see extra information, such as its belonging into a category! Following example gives a nice explanation about what I just said.

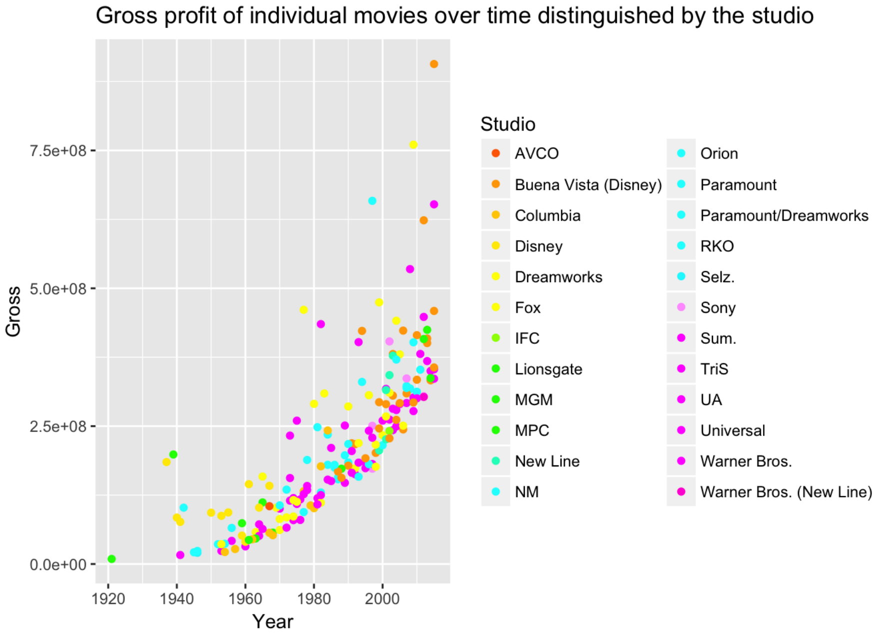
```
# scatter plot with discrete scale
ex=(ggplot(data=movie,aes(x=Year,y=Gross,color=Studio))+
  geom_point()+
  labs(title="Gross profit of individual movies over time distinguished by the studio"))
ex
```



When you look at above graph, you can see that additional information is added.

In addition to a simple scatter plot of gross profit distribution of individual movies, we can see another information of individual movies. We can see which studio produced the movie.
You can even select the parameter of color using `scale_color_hue()`.
`l=` stands for luminance
`c=` stands for chroma(intensity of color)
`h=` stands for hue

```
#selecting parameter of color for discrete scale
ex+scale_color_hue(l=100,c=300)
```

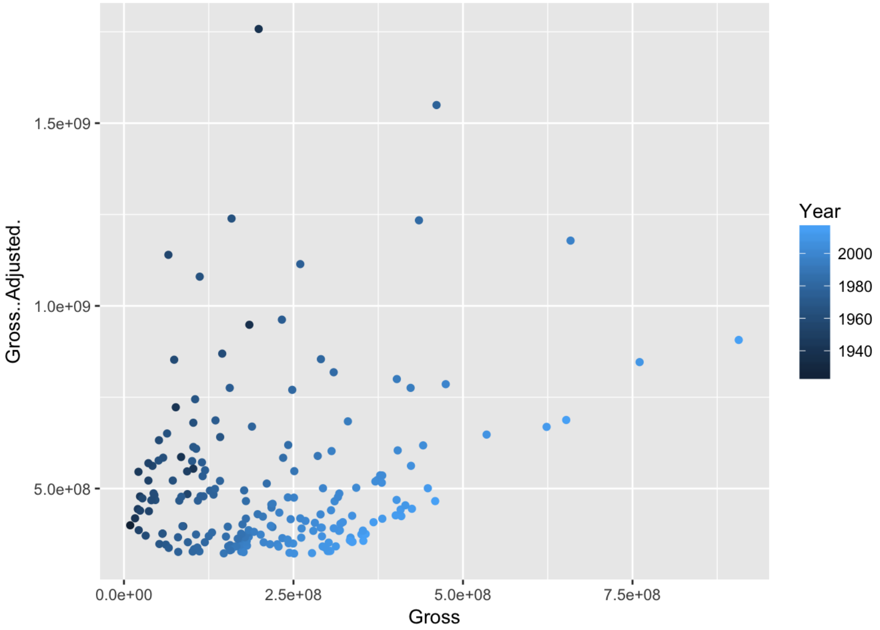


reference : [link] (https://www3.nd.edu/~steve/computing_with_data/12_Scales_themes/scales_themes.html)

2. Continuous Scale

It is also possible to have a scale for continuous variable. From the data set we have, I am going to use year(which is continuous variable) as the scale. The x axis of below graph is gross profit of individual movie, and y axis is the adjusted gross profit of the movie (considering the inflation over time)

```
# scatter plot with continuous scale
plottobesaved=ggplot(data=moviel,aes(x=Gross,y=Gross..Adjusted.,color=Year))+geom_point()
plottobesaved
```



4) Save

The plot you produced can be easily saved by using `ggsave` function!
Unlike other functions I introduced so far, this is not added to the main function `ggplot()`, but rather, it is an independent function.

```
# how to save ggplot
ggsave(plottobesaved,file="minimalmovieplot.jpg",
        path="/Users/hwangyeonmi/stat133/stat133-hws-fall17/post01/pictures")
```

```
## Saving 7 x 5 in image
```

[Take home Message]

Through this blog, I introduced you to the world of `ggplot2` (although you have been exposed to it during the class and lab) !
I talked about the very basics of the ggplot2, such as terminology, basic structure, and advantage of using `ggplot2`.And introduced types of graphs, and cool features of ‘ggplot2’.
In conclusion, `ggplot2` is a very simple plotting system that even produce various and aestheically pleasing graphs.
In addition, it makes the coding process of complex plot very easy(**faceting** and **scale**)

Basically, the take home message of this blog post is to persuade you to use `ggplot2`.
In addition to the aesthetics of `ggplot2`, cool features like faceting, and scale make the data visualization a lot easier and better.
I can’t really find any reason to **not** use `ggplot2`.

[Reference]

tutorial1 : [link] (http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html)
tutorial2 : [link] (https://www3.nd.edu/~steve/computing_with_data/12_Scales_themes/scales_themes.html)
tutorial3: [link] (http://ggplot2.tidyverse.org/reference/theme.html)
tutorial4 : [link] (http://r4stats.com/examples/graphics-ggplot2/)
image 1&2 : [link] (http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html#themes)
video tutorial : [link] (https://www.youtube.com/watch?v=49fADBfcDD4)
dataset : [link] (https://drive.google.com/a/berkeley.edu/file/d/0B3ITGuQsifiJcUtEQ3R0UUlmRGc/view?usp=sharing)