Movie Rating Part2 Rcodes

Peiyun Zhou

I will further demonstrates how to create boxplot, adjust the scale of the plots, and label the plots in this R Markdown document. These data visualization will help us to inspect the relationships between Audiences'/Critics' move ratings and budget, genre, year, ect.

First, load the dataset and rename the columns

```
movies<-read.csv("Movie-Ratings.csv")
colnames(movies)<-c("Film","Genre","CriticRating","AudienceRating","BudgetMillions","Year") # rename th
movies$Year<-as.factor(movies$Year) # Set year as a factor variable
```

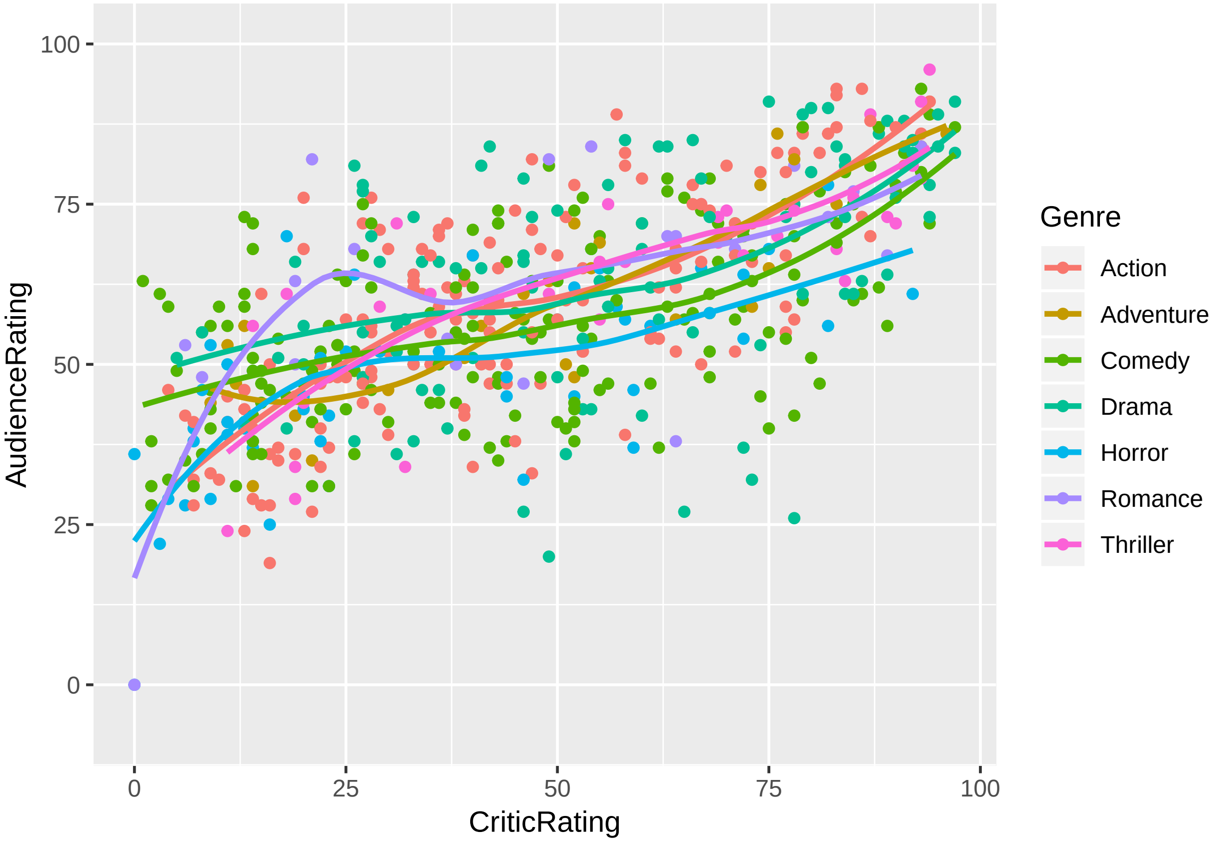
Load the library for visualization

```
library(ggplot2)
```

We can see how the audience rating changes based on the change of the critics rating using the smooth function in the ggplot

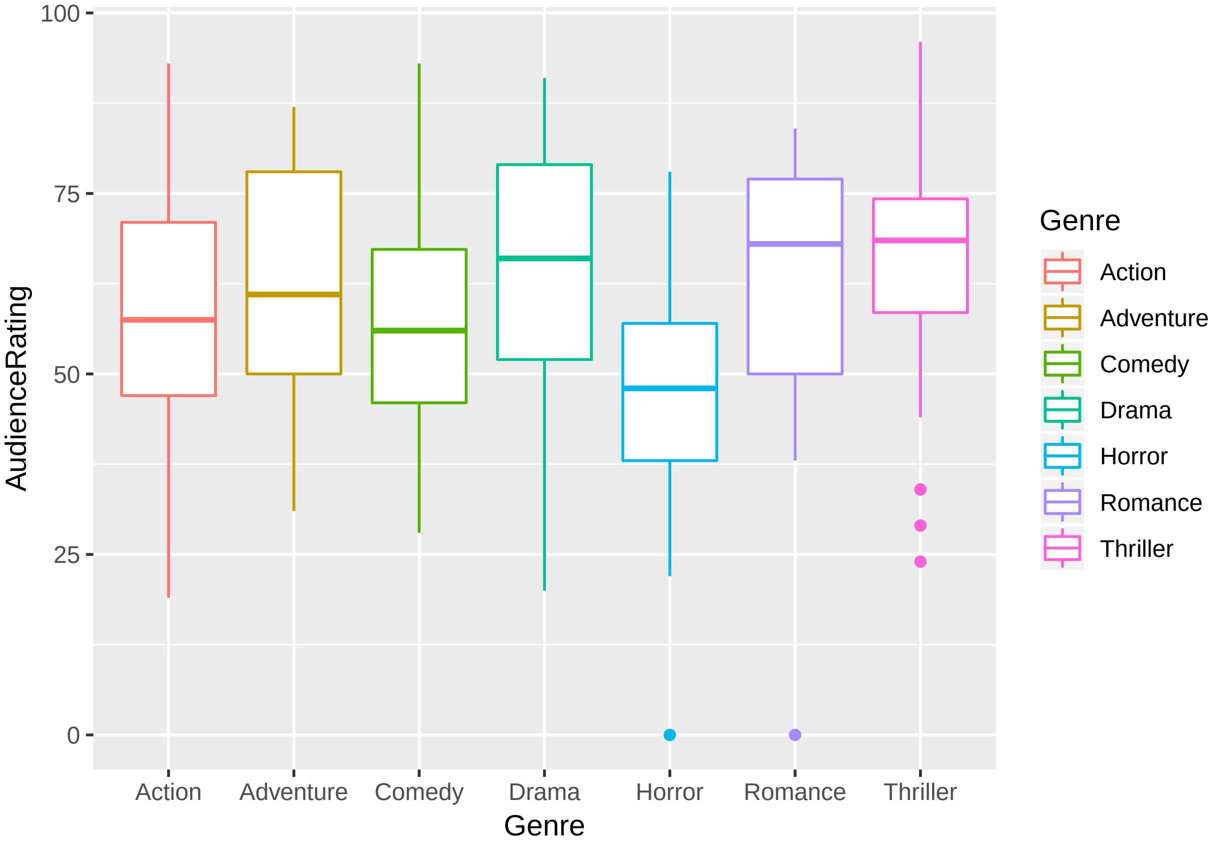
```
u<-ggplot(data=movies,aes(x=CriticRating,y=AudienceRating,color=Genre))
u+geom_point()+geom_smooth(fill=NA)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

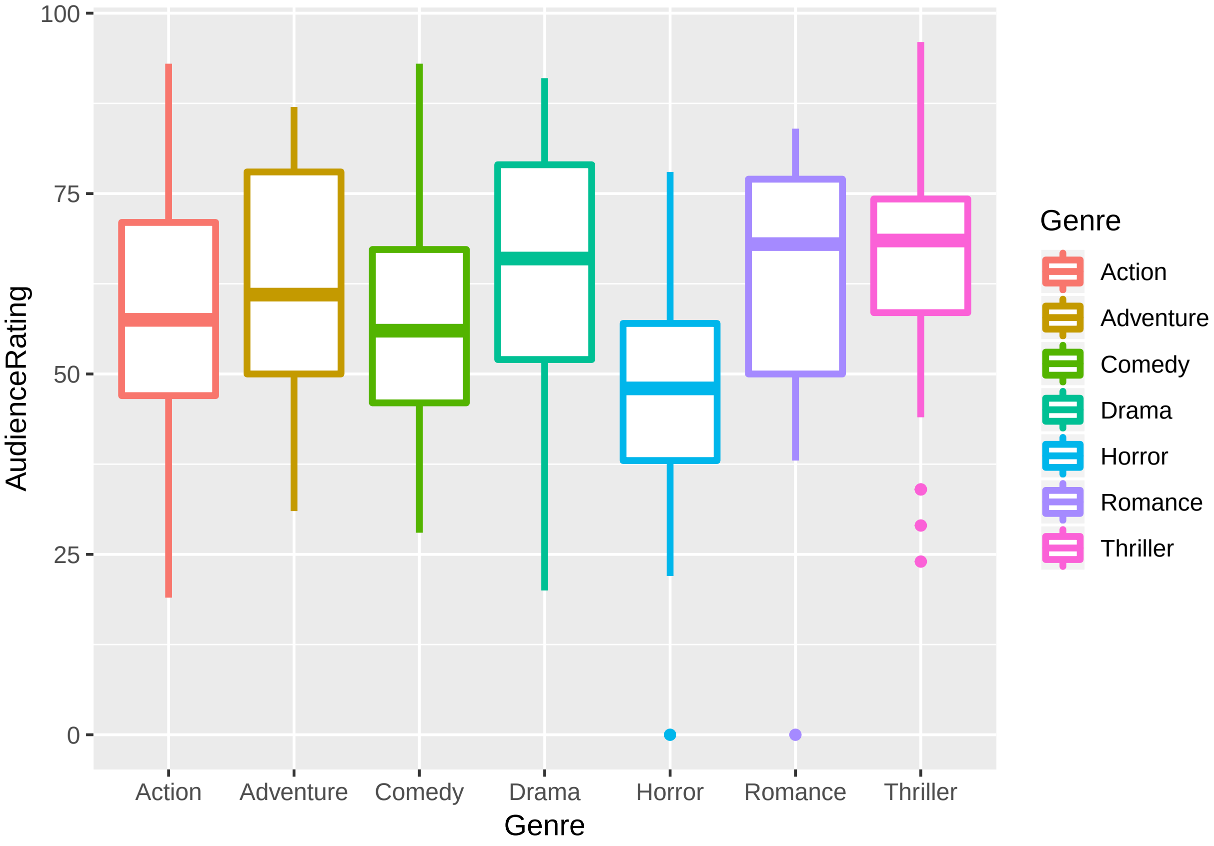


We can further explore how Audiences' ratings differ based on different genre through building boxplots

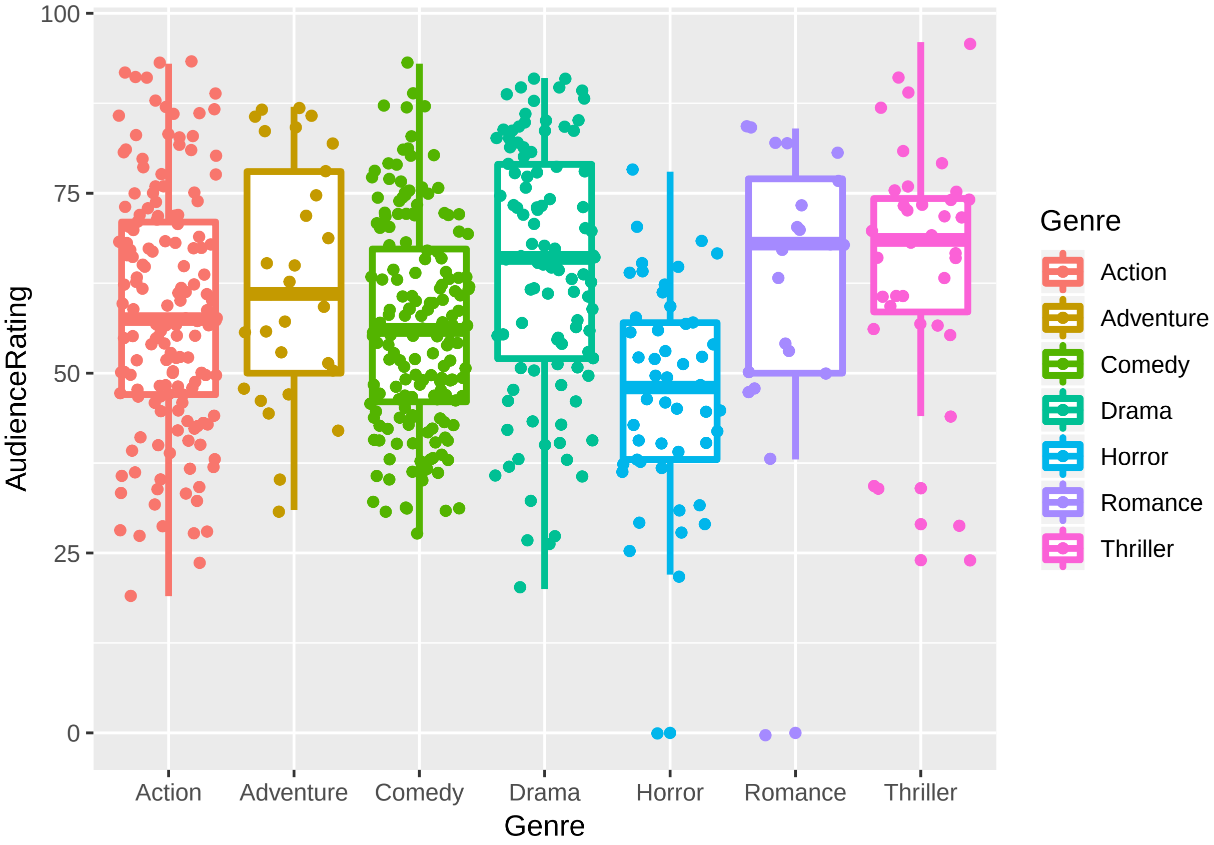
```r
u<-ggplot(data=movies,aes(x=Genre,y=AudienceRating,color=Genre))
u+geom_boxplot()
```



```r
u+geom_boxplot(size=1.2) # Change the size of the boxplot
```

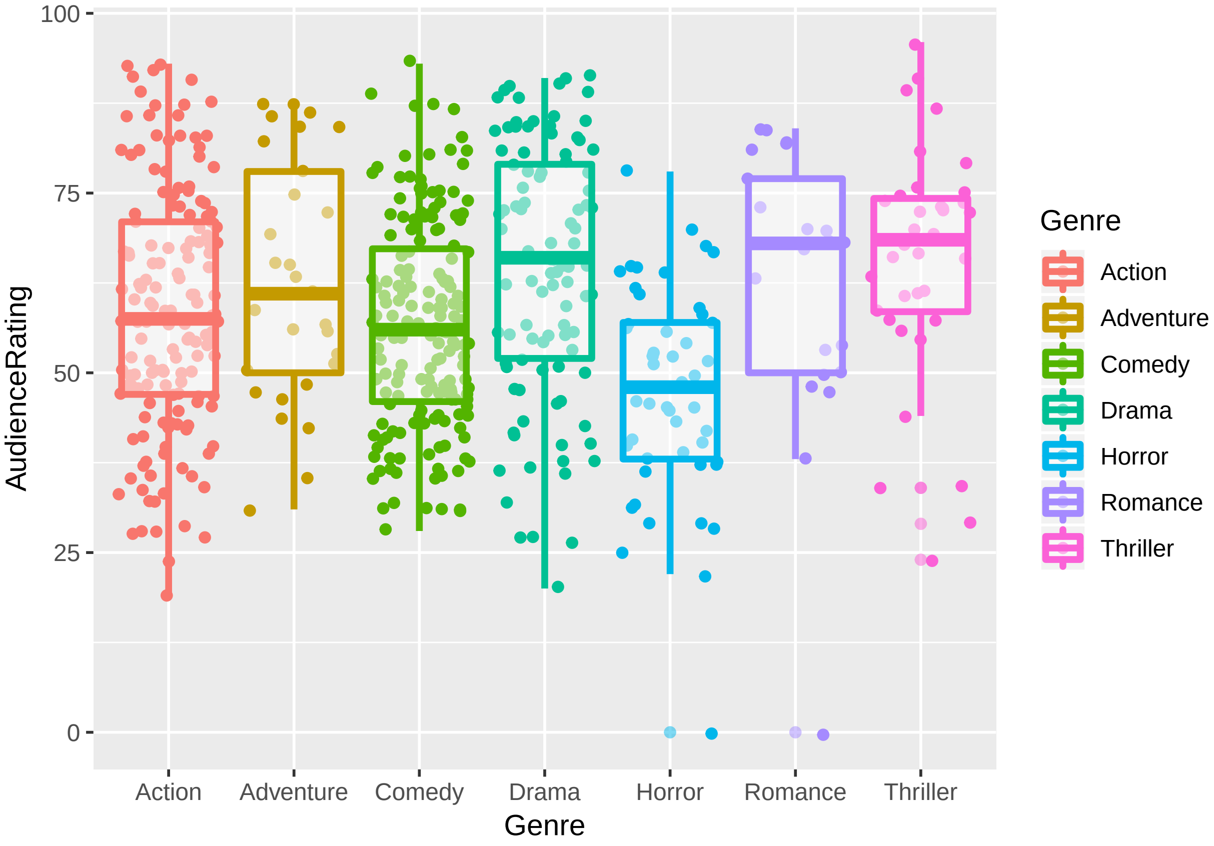


```
u+geom_boxplot(size=1.2)+geom_jitter() # Include the dots
```



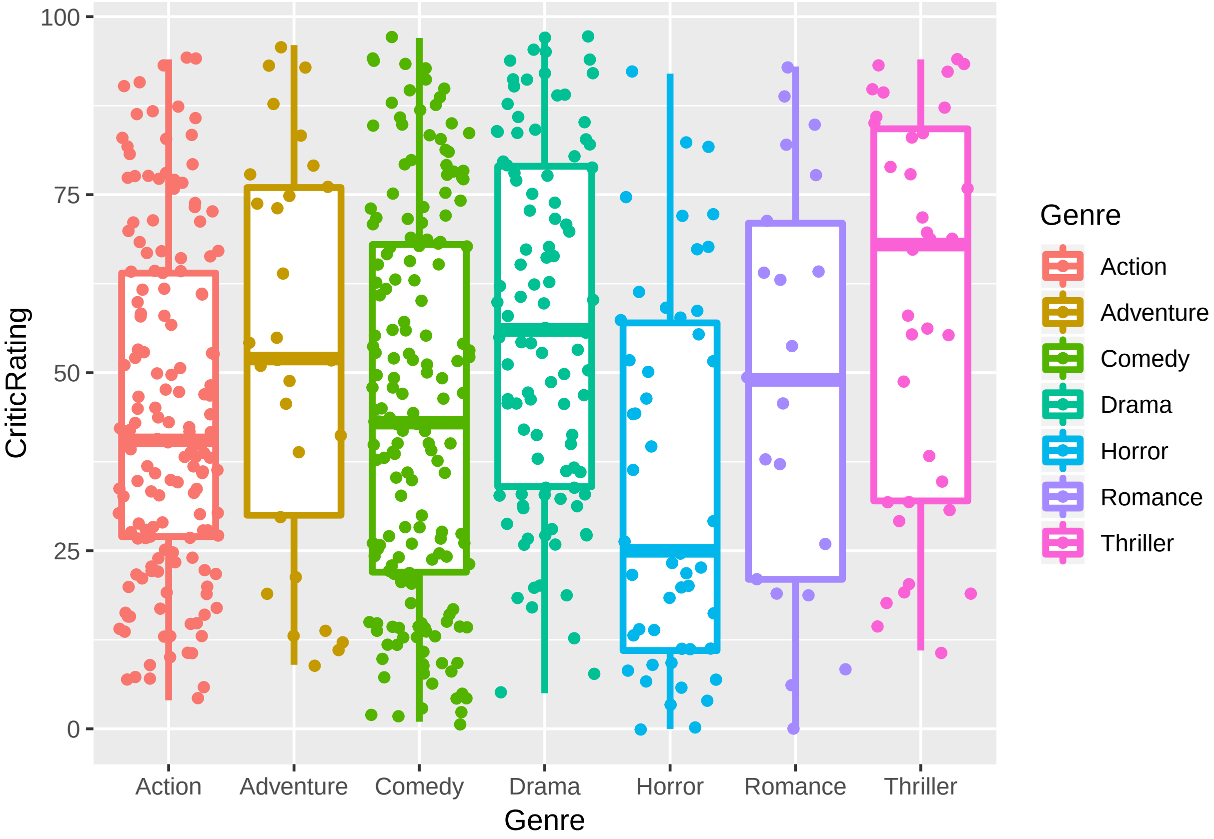
Another way to create boxplot for Audience Rating with manipulation of the transparency of the data points

```
u+geom_jitter()+geom_boxplot(size=1.2,alpha=0.5)
```

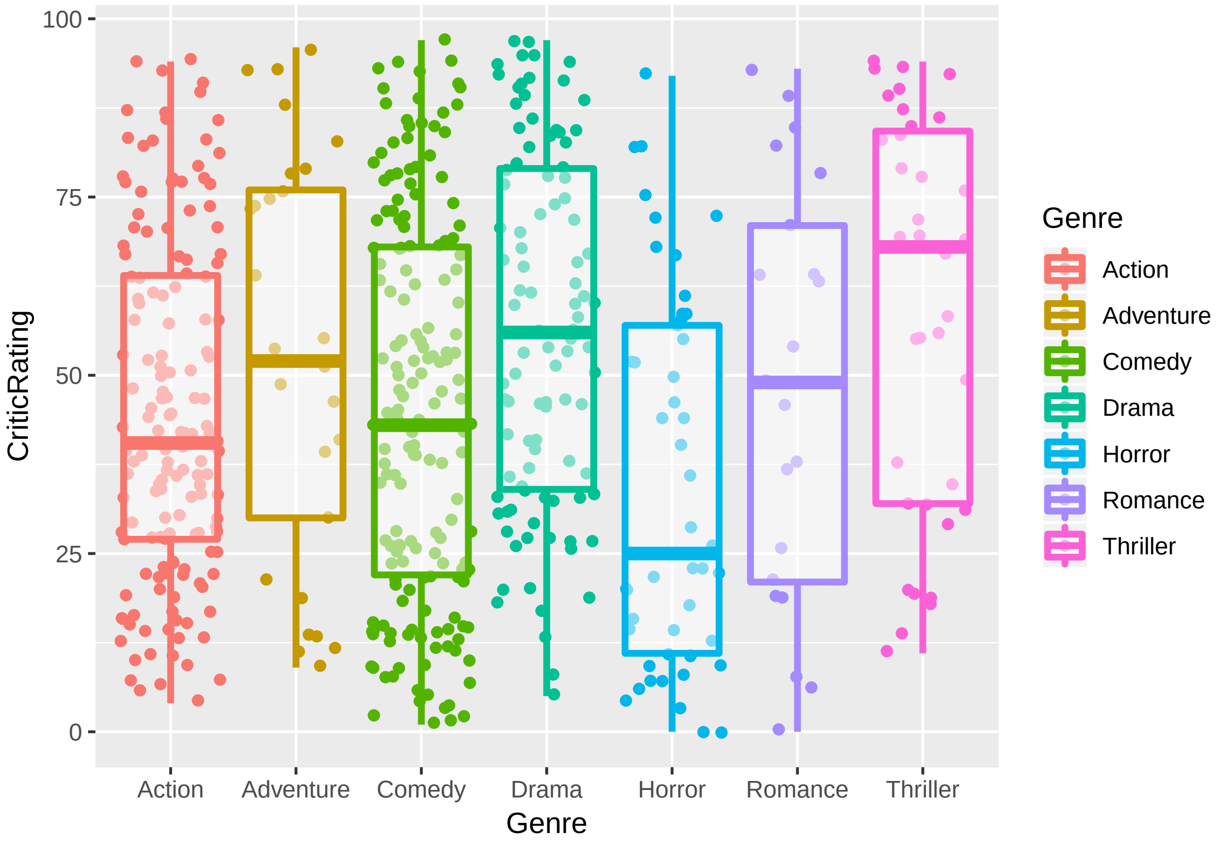


We can also explore how Cristics' ratings change depend on the genre.

```
b<-ggplot(data=movies,aes(x=Genre,y=CriticRating,color=Genre))
b+geom_boxplot(size=1.2)+geom_jitter()
```

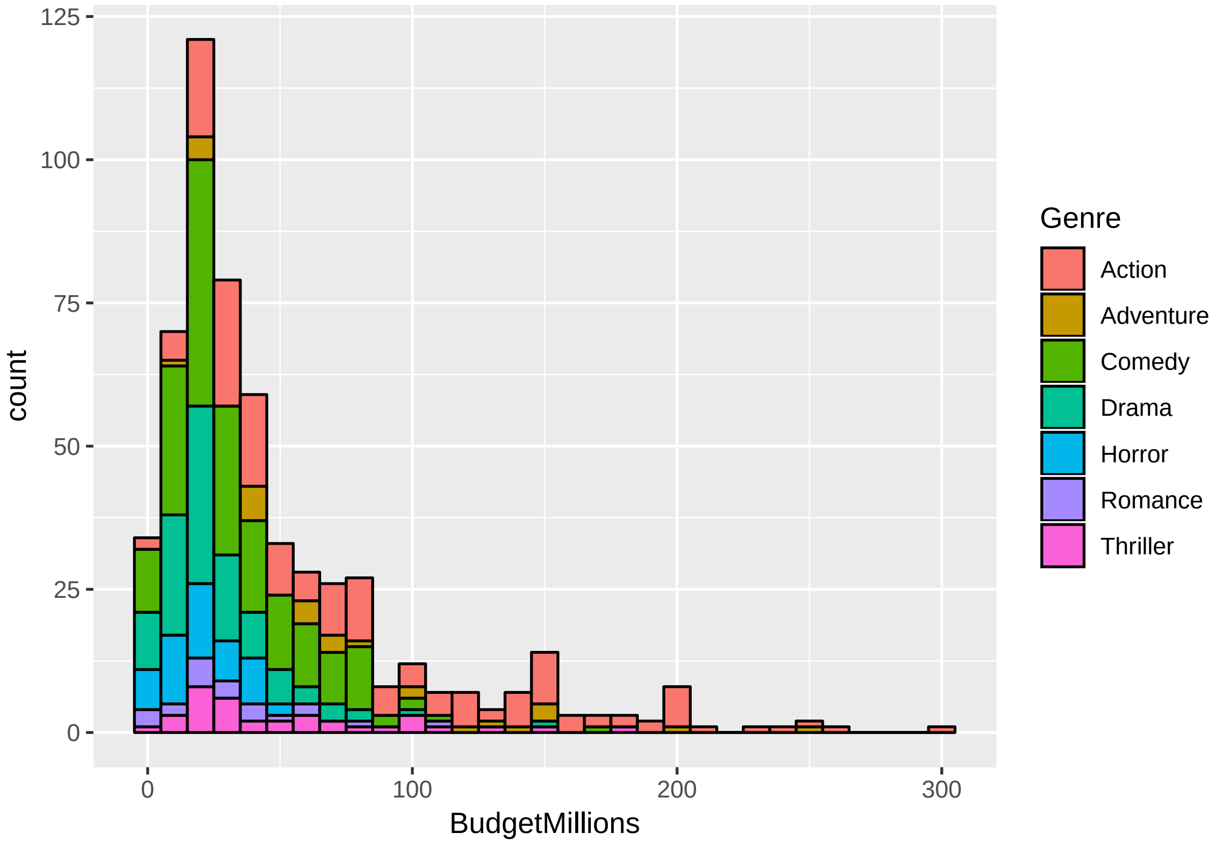


```
# Another way to create boxplot for Critics Rating:
b+geom_jitter()+geom_boxplot(size=1.2,alpha=0.5)
```

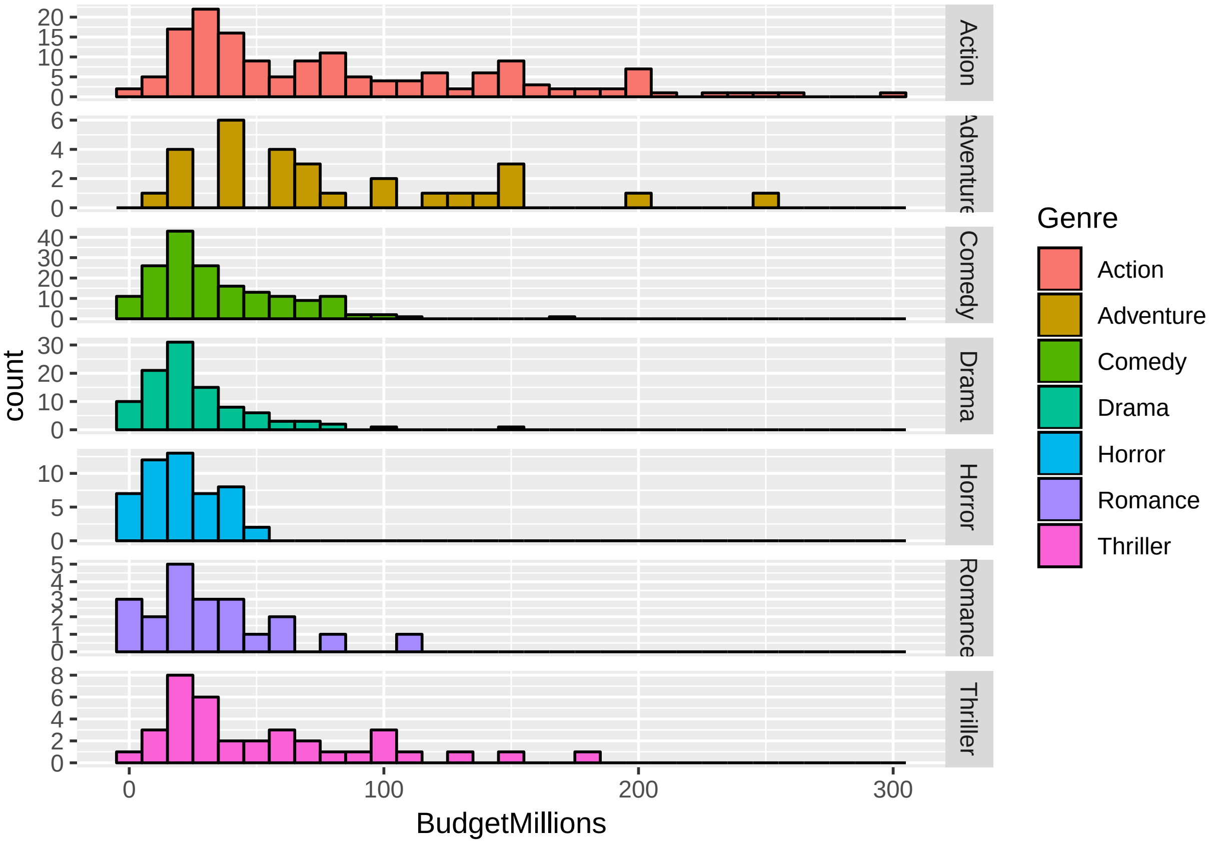


We can use facet in the ggplot to seperate the histographs for movie budget based on the movie genres

```
# Facets
v<-ggplot(data=movies,aes(x=BudgetMillions))
v+geom_histogram(binwidth=10,aes(fill=Genre),color="Black")
```

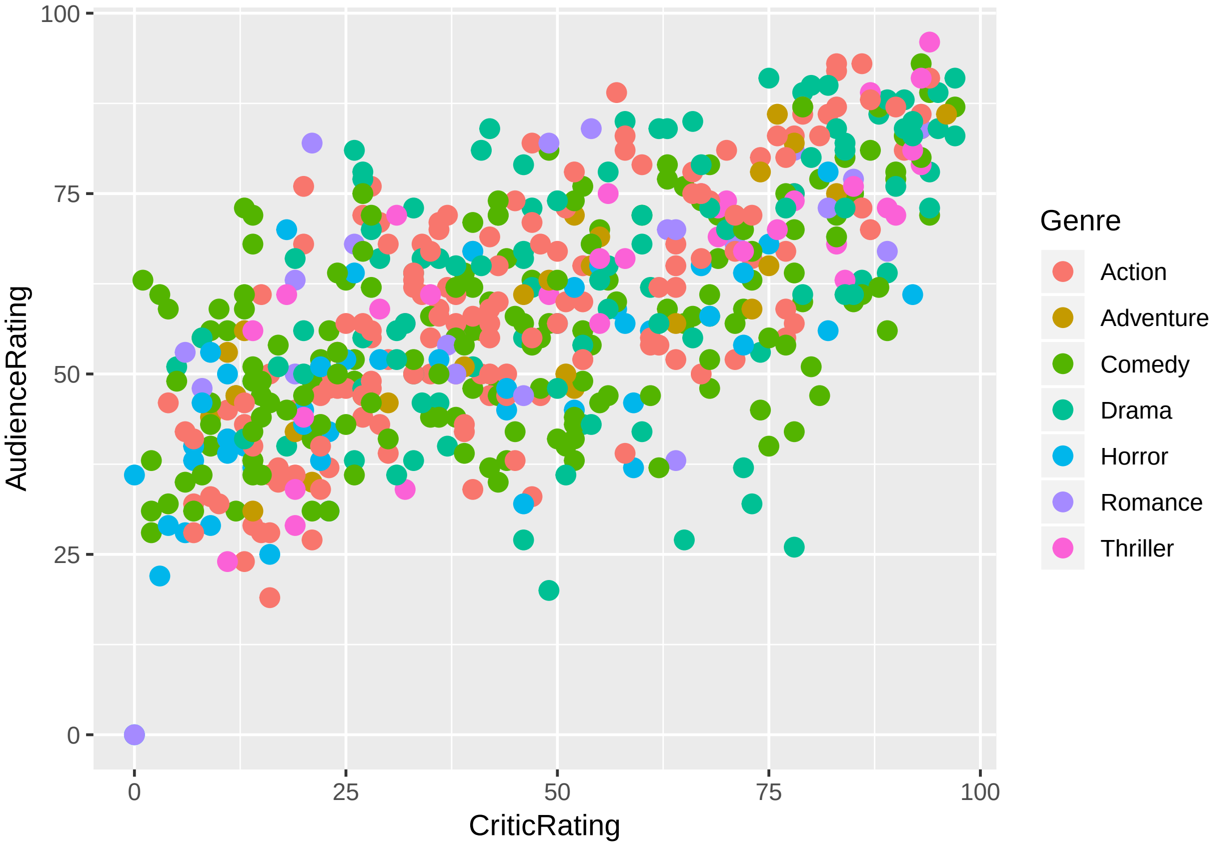


```
v+geom_histogram(binwidth=10,aes(fill=Genre),
                 color="Black")+
  facet_grid(Genre~.,scale="free")
```

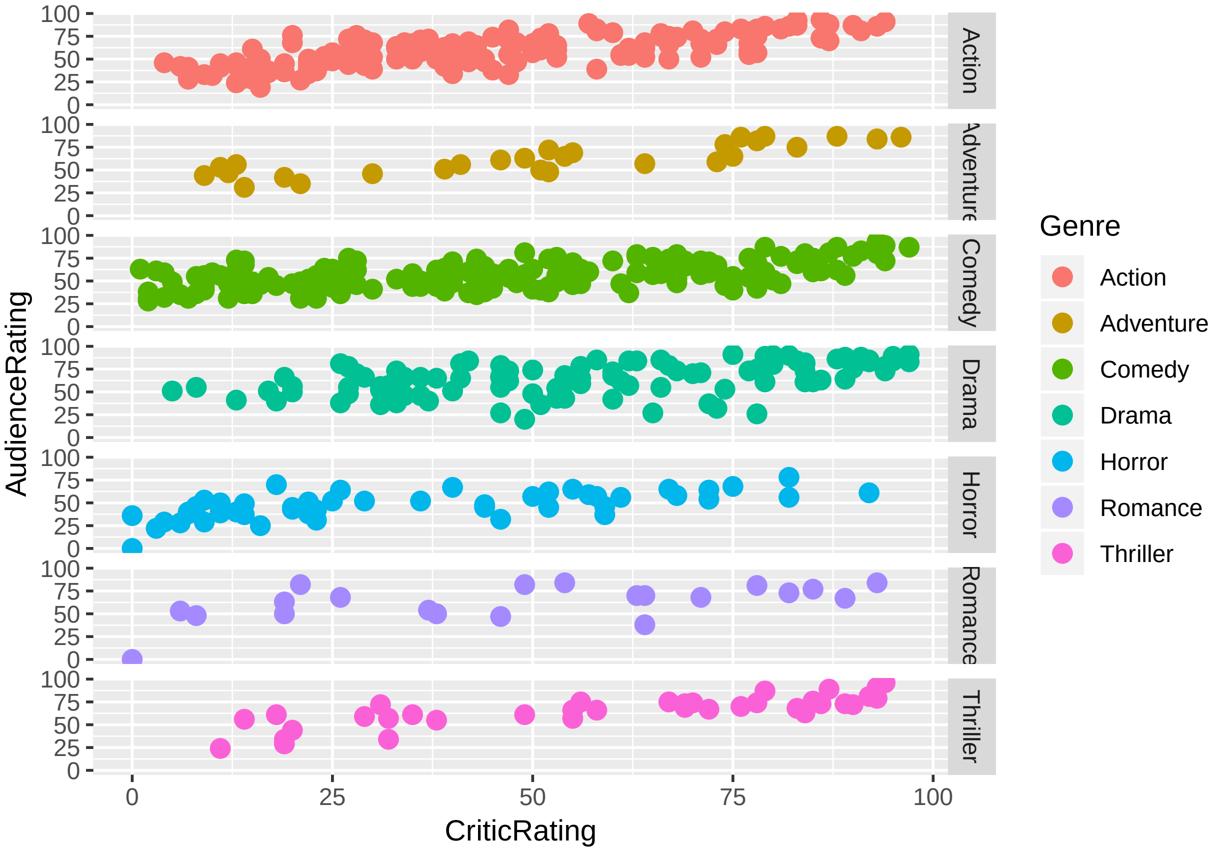


Scatterplot can also be used to visualize the relationships between audiences' ratings and critics ratings based on genre

```
w<-ggplot(data=movies,aes(x=CriticRating,y=AudienceRating,color=Genre))
w+geom_point(size=3)
```

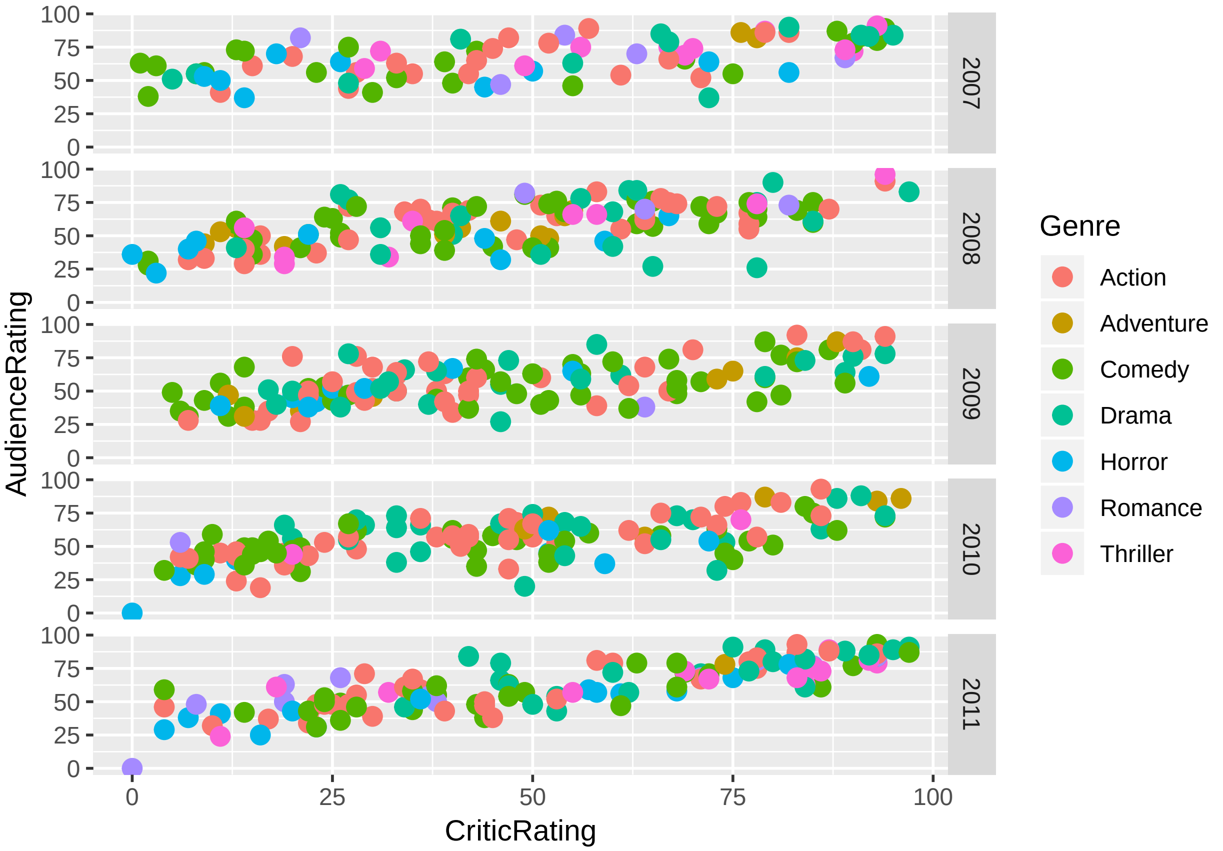


```
w+geom_point(size=3)+
  facet_grid(Genre~.)
```



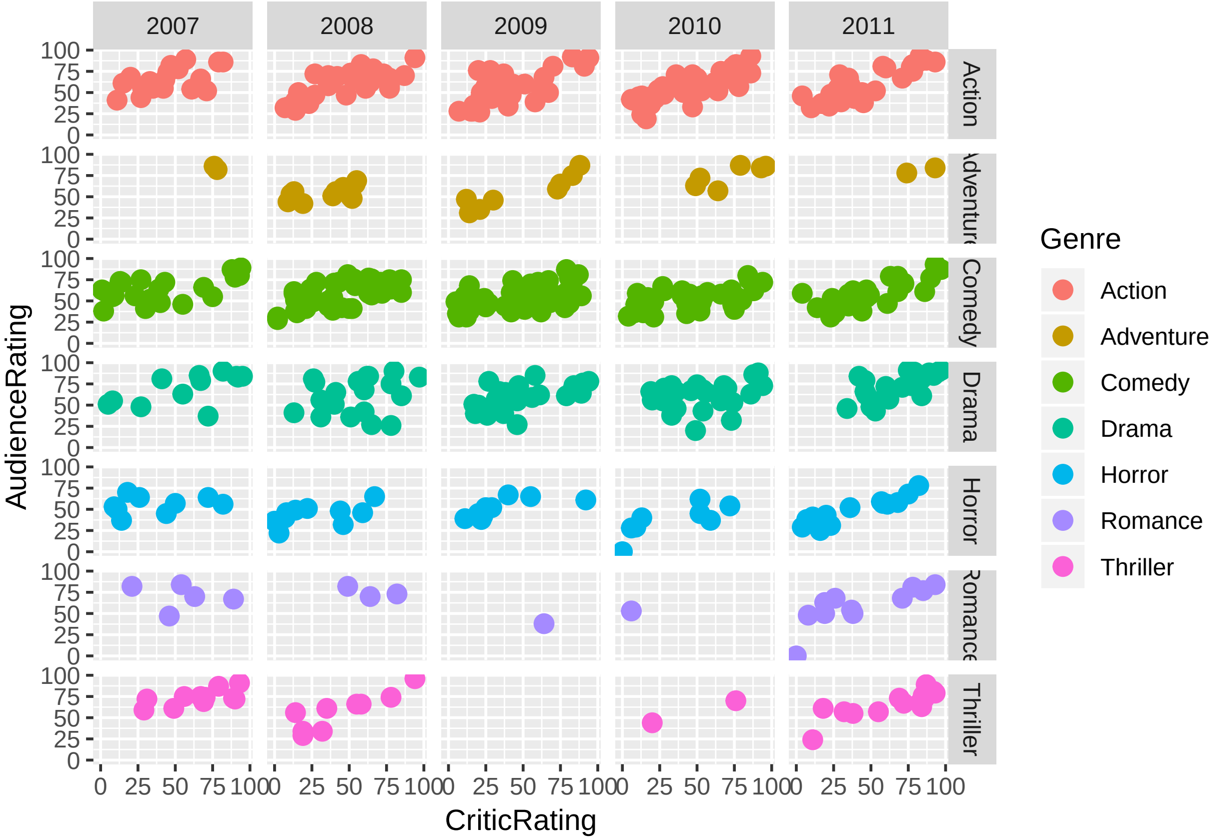
The relationships between audiences' ratings and critics ratings based on Year

```
w+geom_point(size=3)+
  facet_grid(Year~.)
```



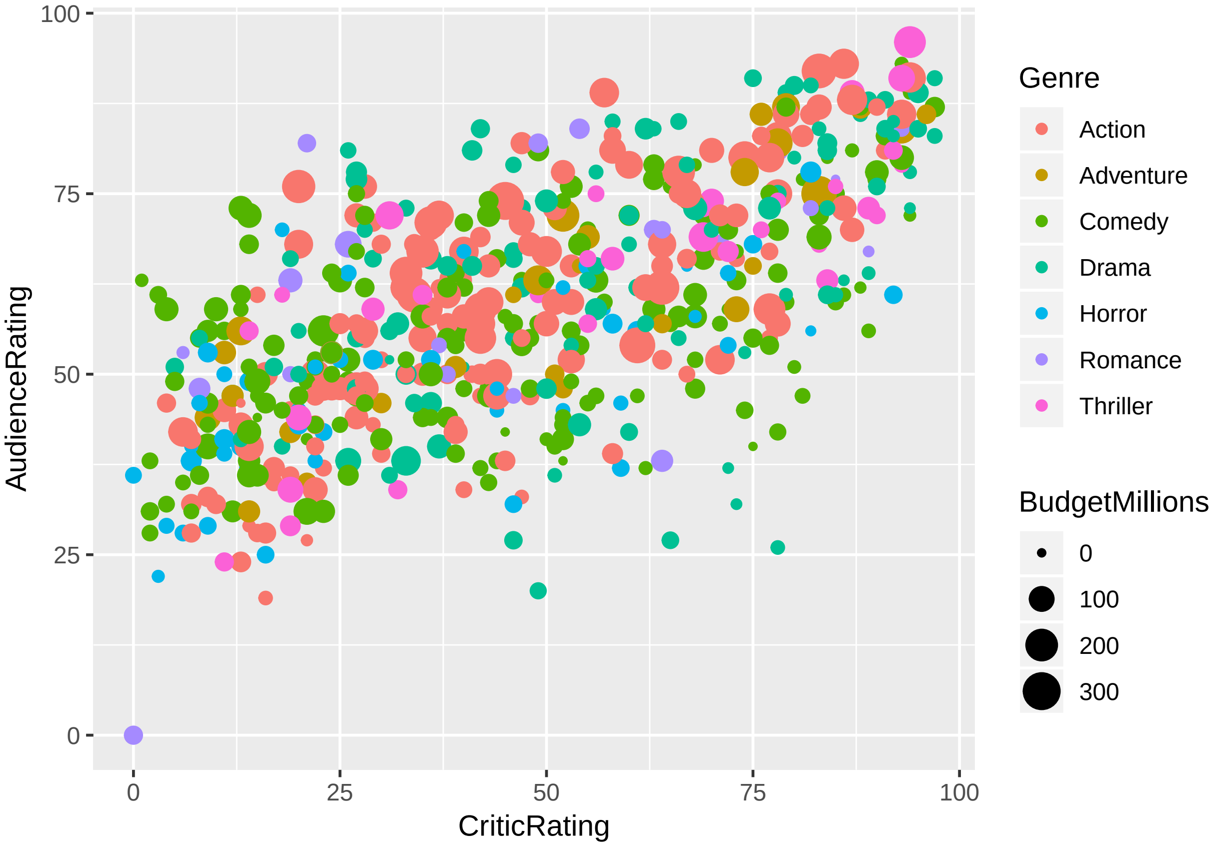
The relationships between audiences' ratings and critics ratings based on Year and Genre

```
w+geom_point(size=3)+
  facet_grid(Genre~Year)
```



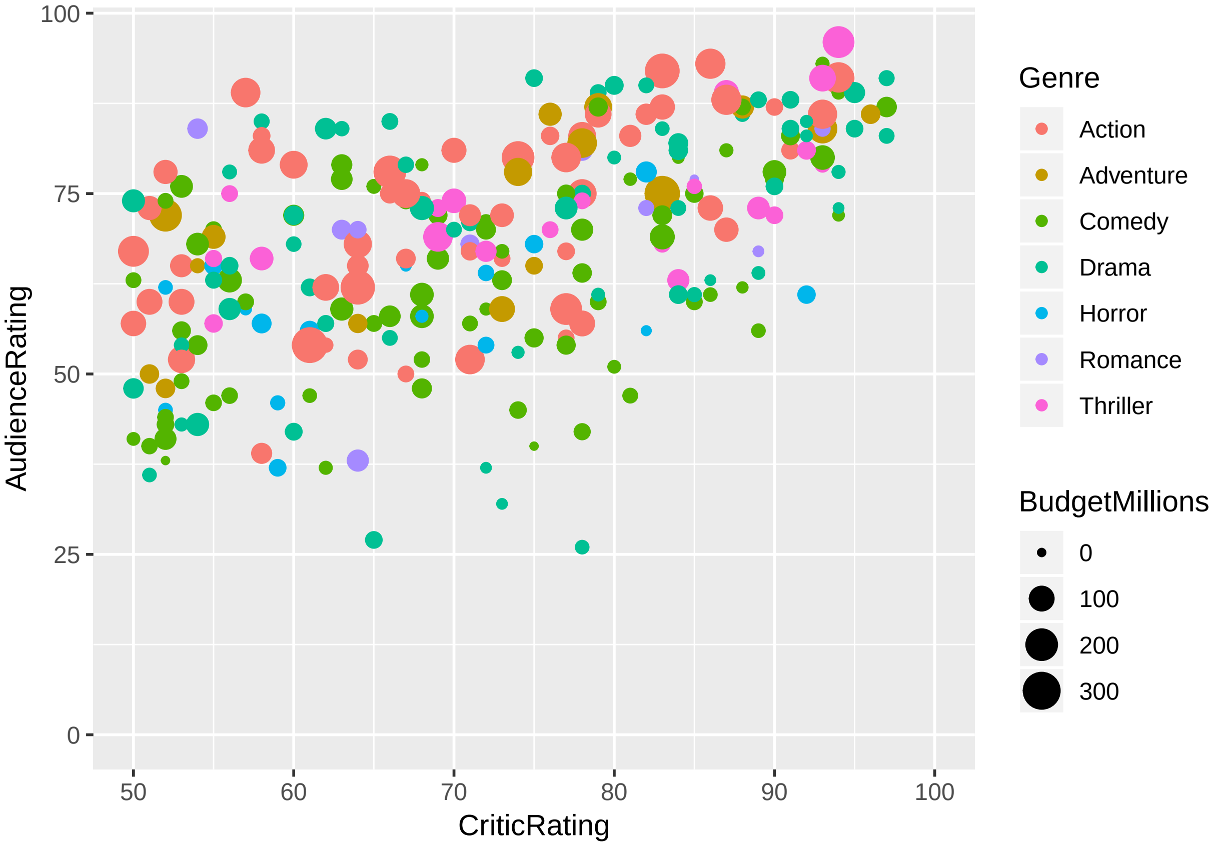
We can further adjust the scale of the plots

```
m<-ggplot(data=movies,aes(x=CriticRating,y=AudienceRating,
                          size=BudgetMillions,
                          color=Genre))
m+geom_point()
```



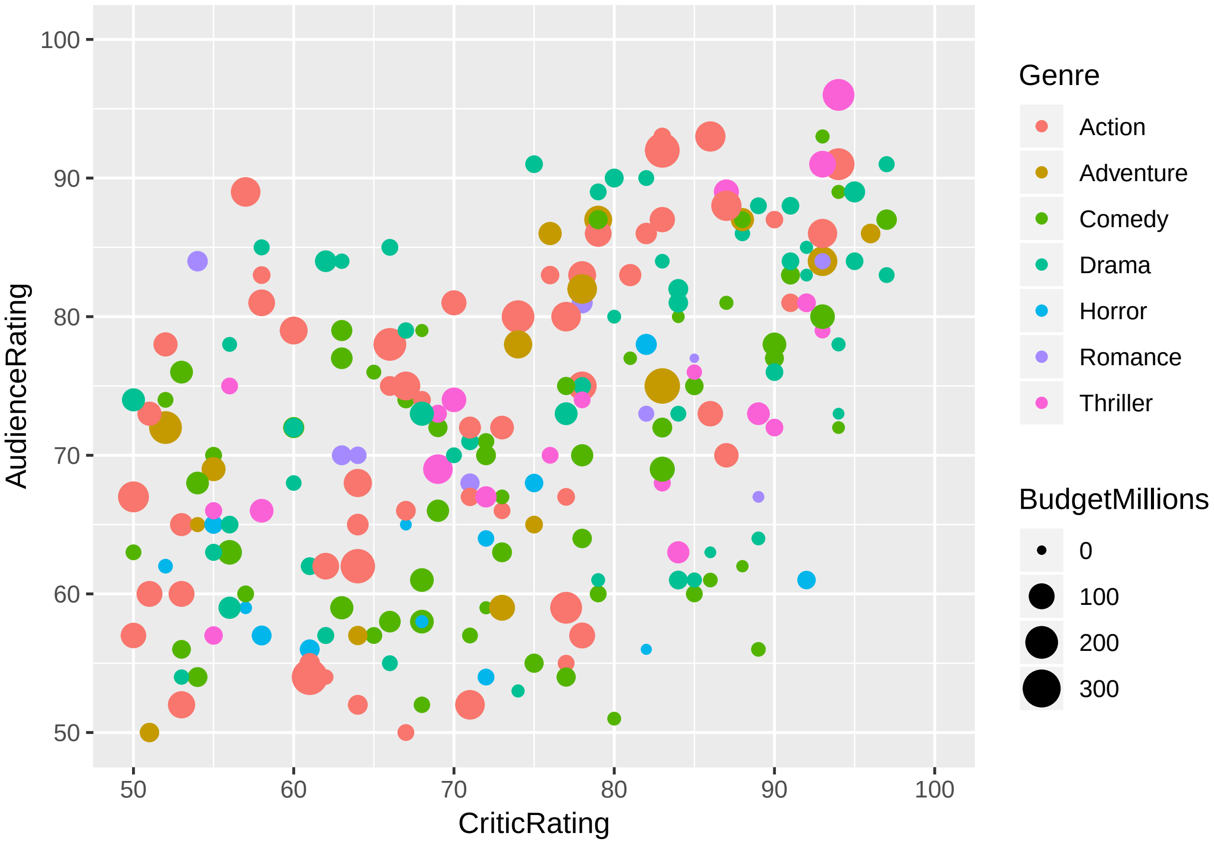
```
m+geom_point()+
  xlim(50,100) # Zoom in and Remove rows that are outside the boundaries
```

```
## Warning: Removed 304 rows containing missing values (geom_point).
```



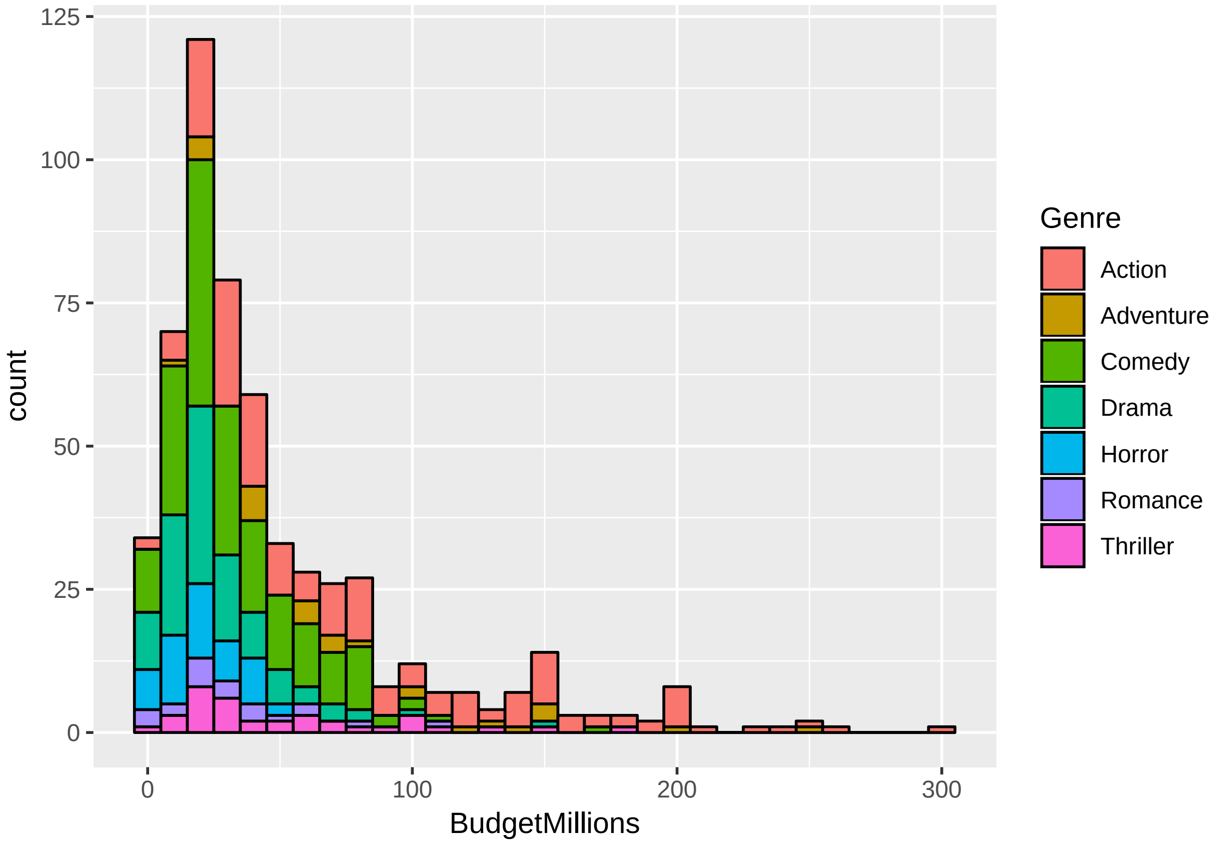
```
m+geom_point()+
  xlim(50,100) +
  ylim(50,100) # Further Zoom in and remove columns that are outside the boundaries
```

```
## Warning: Removed 335 rows containing missing values (geom_point).
```



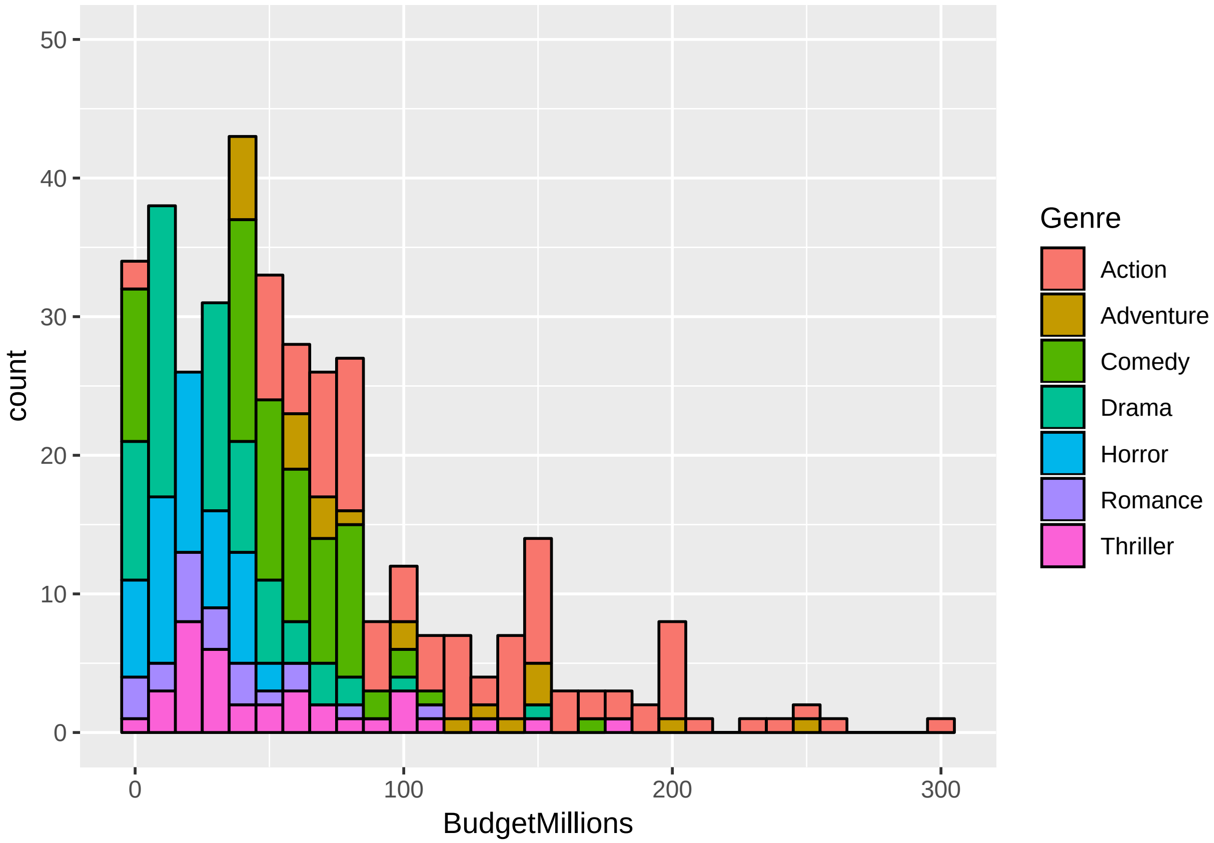
The incorrect way to zoom in data will lead to trimmed data

```
# This does not always work
n<-ggplot(data=movies,aes(x=BudgetMillions))
n+geom_histogram(binwidth = 10,aes(fill=Genre),color="Black")
```



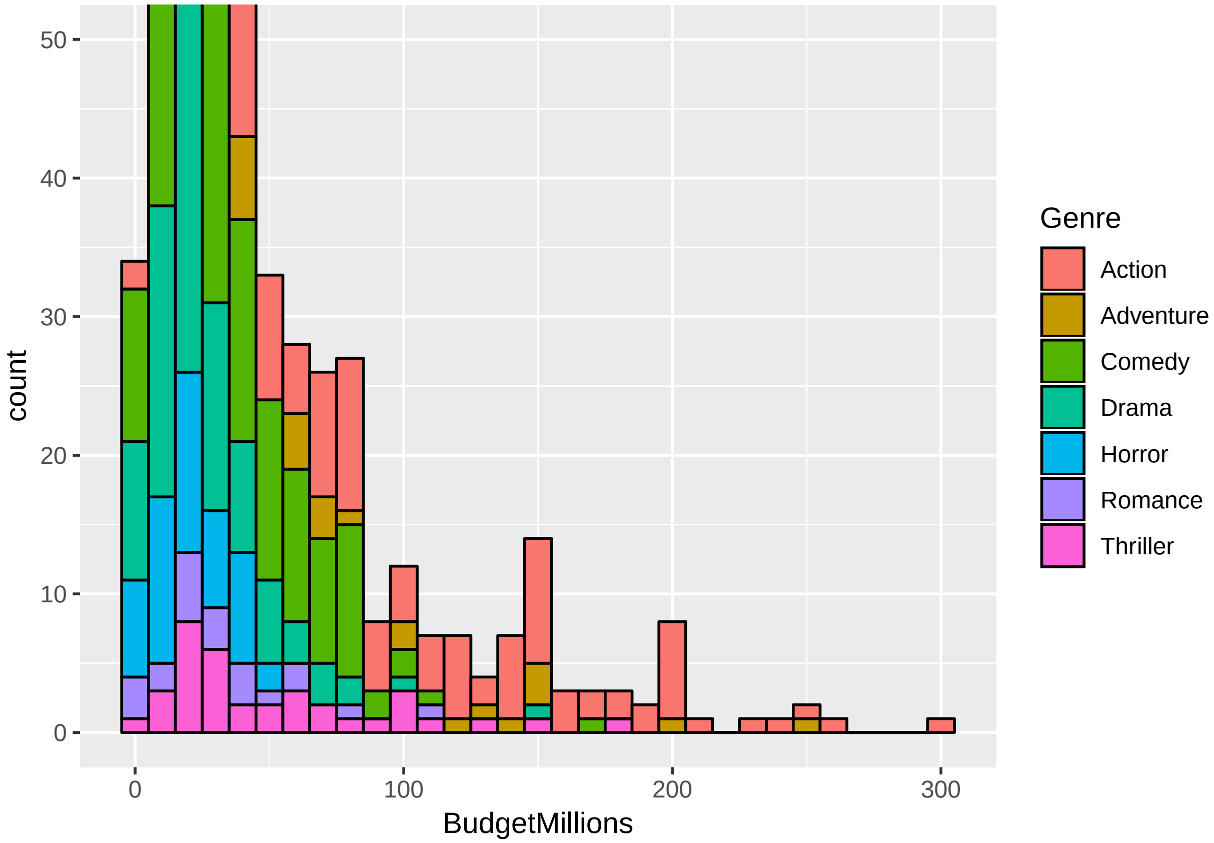
```
n+geom_histogram(binwidth = 10,aes(fill=Genre),color="Black")+
  ylim(0,50) # This would trim the data, rather than zoom in
```

```
## Warning: Removed 11 rows containing missing values (geom_bar).
```



The correct way to create zoom-in plot Use coord_cartesian( )

```
n+geom_histogram(binwidth = 10,aes(fill=Genre),color="Black")+
  coord_cartesian(ylim=c(0,50))  # Zoom in into the chart
```


count
0
10
20
30
40
50
0
100
200
300
BudgetMillions
Genre
Action
Adventure
Comedy
Drama
Horror
Romance
Thriller