Movie Rating

Peiyun Zhou

This R Markdown document demonstrates how to preprocess the movies dataset and explore different questions about relationships between Audience/Critics ratings and Budget, Genre, and Year.

First, we load the dataset "movie_rating.csv", rename the variables, and set Year as a factor variable.

```
movies<-read.csv("Movie-Ratings.csv")
head(movies) # Explore the dataset
```

```
##                    Film     Genre Rotten.Tomatoes.Ratings..
## 1 (500) Days of Summer     Comedy                        87
## 2           10,000 B.C. Adventure                         9
## 3            12 Rounds     Action                        30
## 4             127 Hours Adventure                        93
## 5             17 Again     Comedy                        55
## 6                 2012    Action                        39
##   Audience.Ratings.. Budget..million... Year.of.release
## 1                 81                  8            2009
## 2                 44                105            2008
## 3                 52                 20            2009
## 4                 84                 18            2010
## 5                 70                 20            2009
## 6                 63                200            2009
```

```
colnames(movies)<-c("Film","Genre","CriticRating","AudienceRating","BudgetMillions","Year") # rename th
head(movies) # Recheck the dataset
```

```
##                    Film     Genre CriticRating AudienceRating
## 1 (500) Days of Summer     Comedy           87             81
## 2           10,000 B.C. Adventure            9             44
## 3            12 Rounds     Action           30             52
## 4             127 Hours Adventure           93             84
## 5             17 Again     Comedy           55             70
## 6                 2012    Action           39             63
##   BudgetMillions Year
## 1              8 2009
## 2            105 2008
## 3             20 2009
## 4             18 2010
## 5             20 2009
## 6            200 2009
```

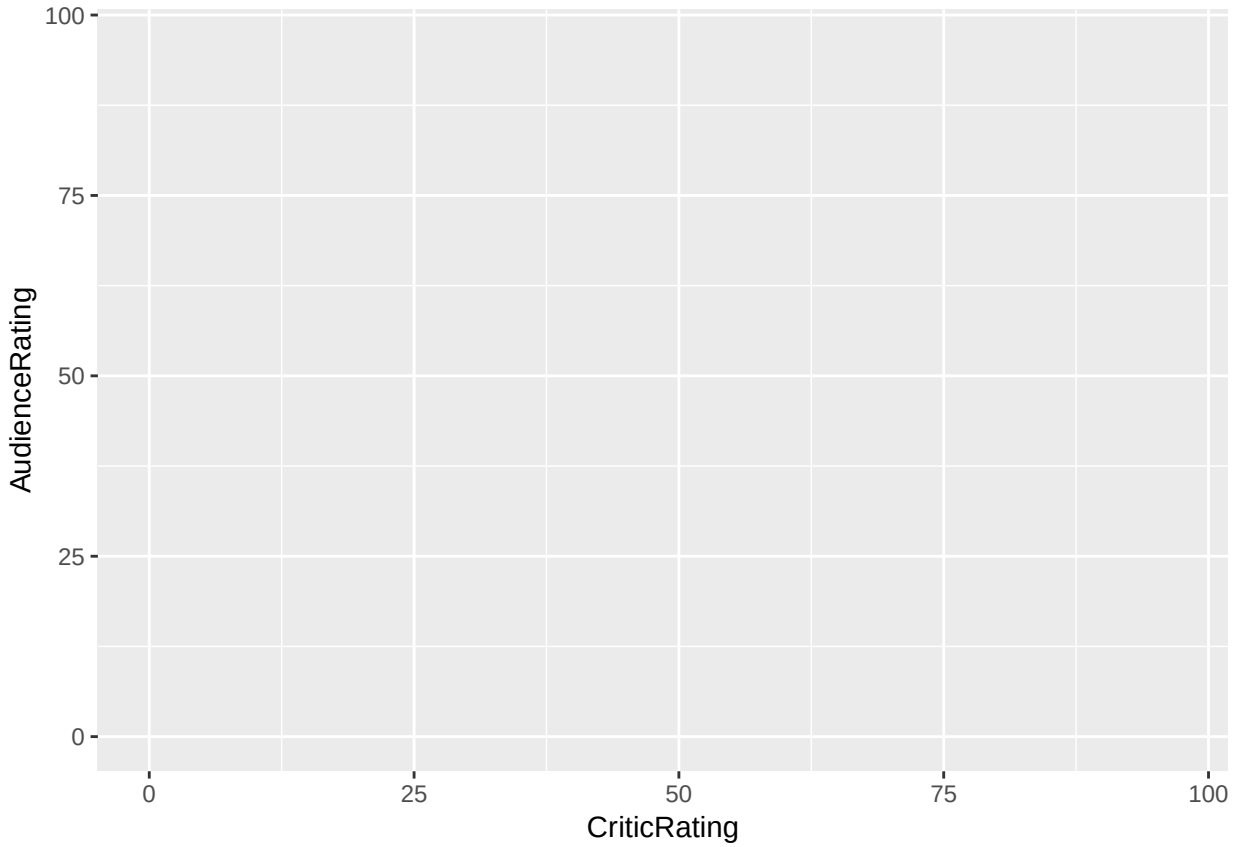
```
# tail(movies)
str(movies) # check the structure of the variables in the dataset
```

```
## '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 ...
##  $ CriticRating  : int  87 9 30 93 55 39 40 50 43 93 ...
##  $ AudienceRating: int  81 44 52 84 70 63 71 57 48 93 ...
##  $ BudgetMillions: int  8 105 20 18 20 200 30 32 28 8 ...
##  $ Year          : int  2009 2008 2009 2010 2009 2009 2008 2007 2011 2011 ...
```

```
movies$Year<-as.factor(movies$Year) # Set year as a factor variable
```

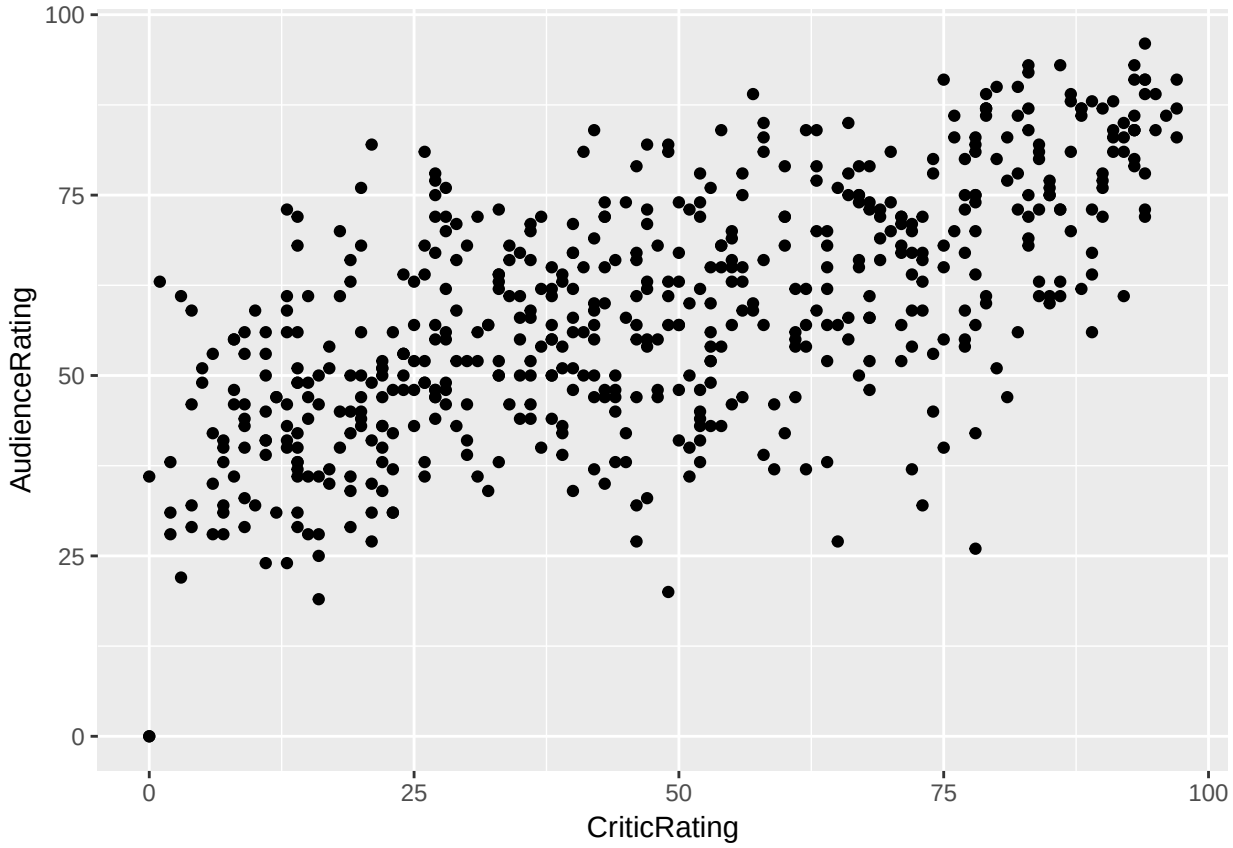
Load the library for visualization Use aec() to map the data to what you want to see Right now we will not see any informaiton about the data

```
library(ggplot2)
ggplot(data=movies,aes(x=CriticRating,y=AudienceRating))
```



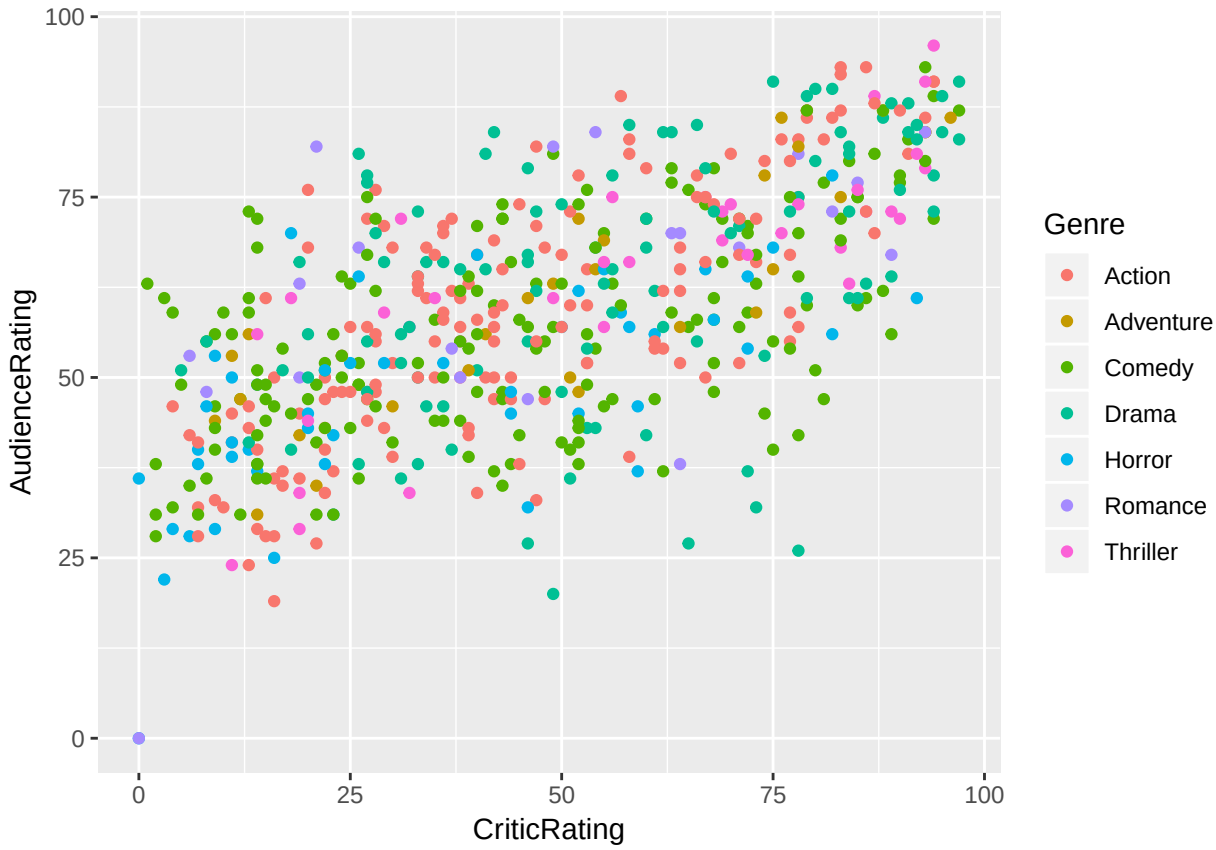
We need to add the geometry layer

```
ggplot(data=movies,aes(x=CriticRating,y=AudienceRating))+
    geom_point()
```



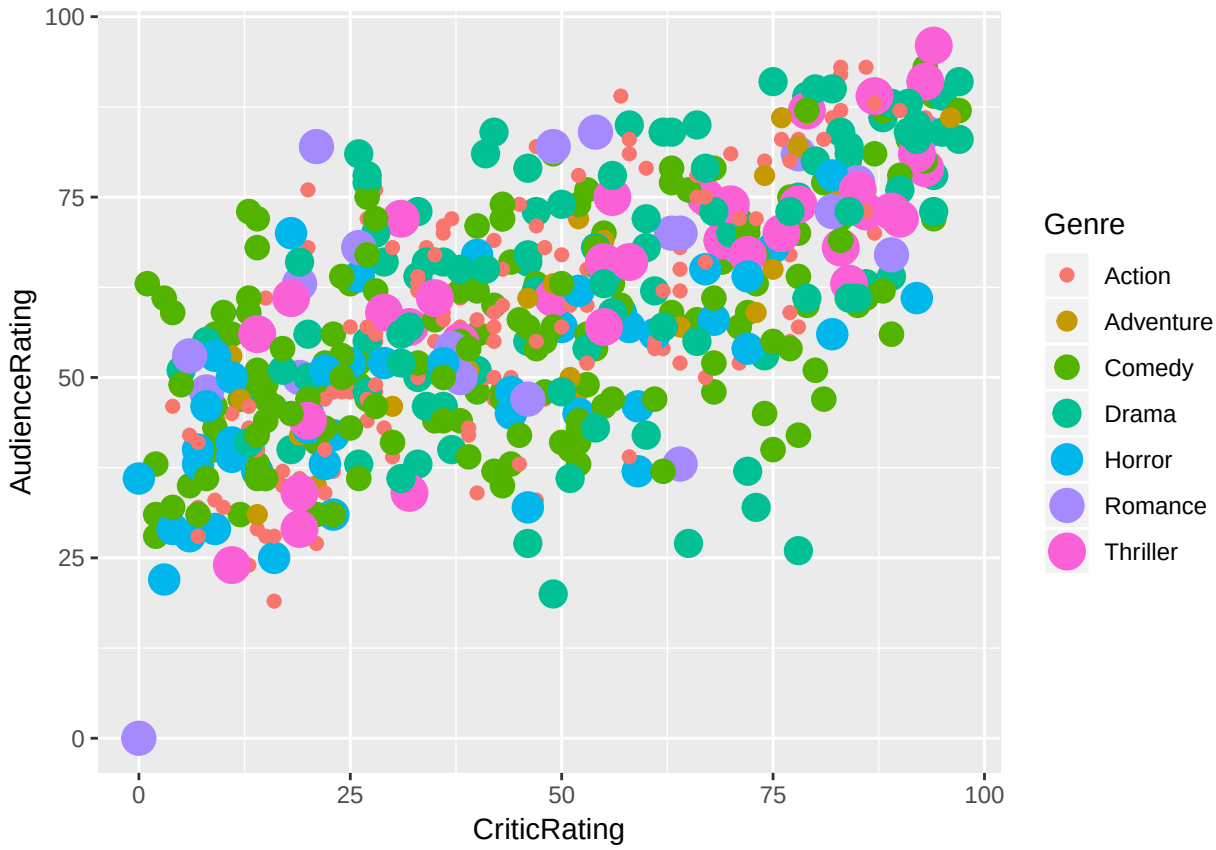
We can assign more parameters to the geometry, such as color, size

```
# Color based on Genre
ggplot(data=movies,aes(x=CriticRating,y=AudienceRating,
                       color=Genre)) +
  geom_point()
```

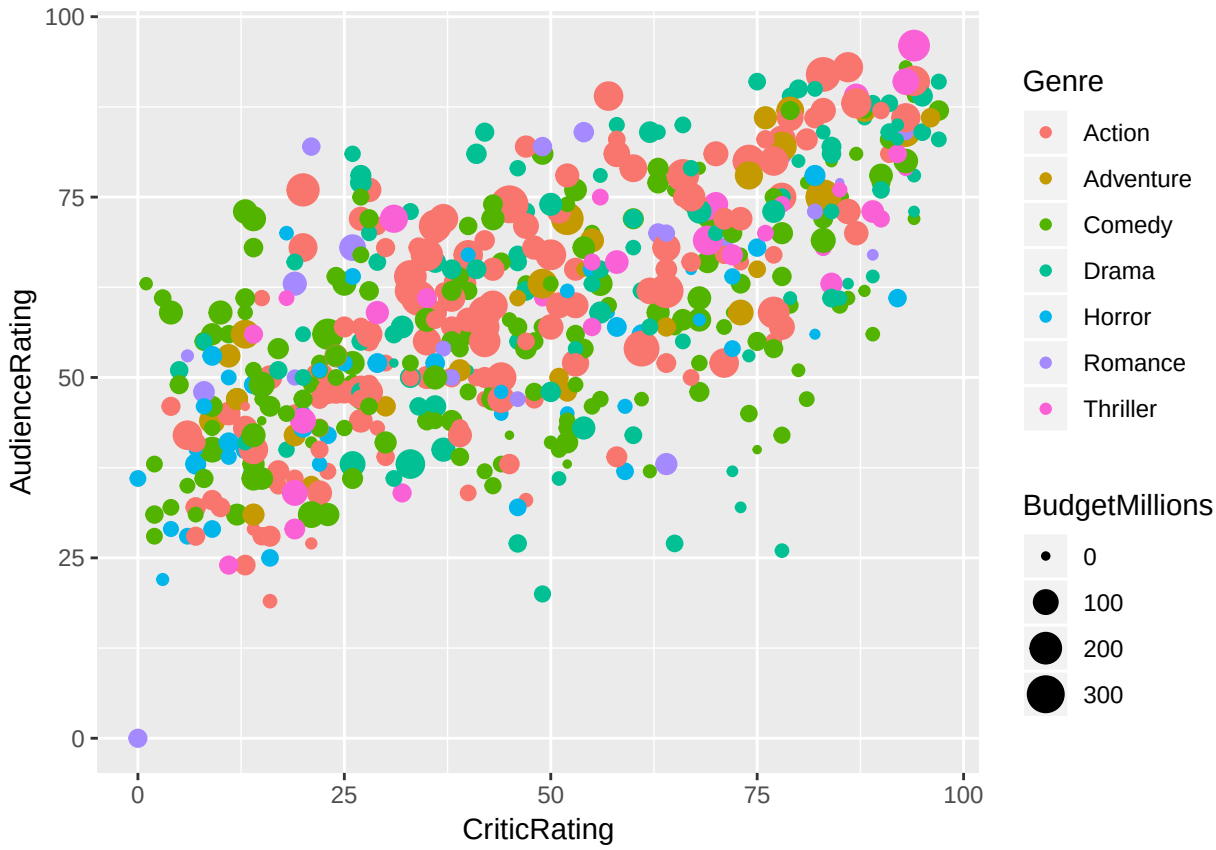



```
# Size and Color based on Genre
ggplot(data=movies,aes(x=CriticRating,y=AudienceRating,
                       color=Genre,size=Genre)) +
  geom_point()
```

```
## Warning: Using size for a discrete variable is not advised.
```

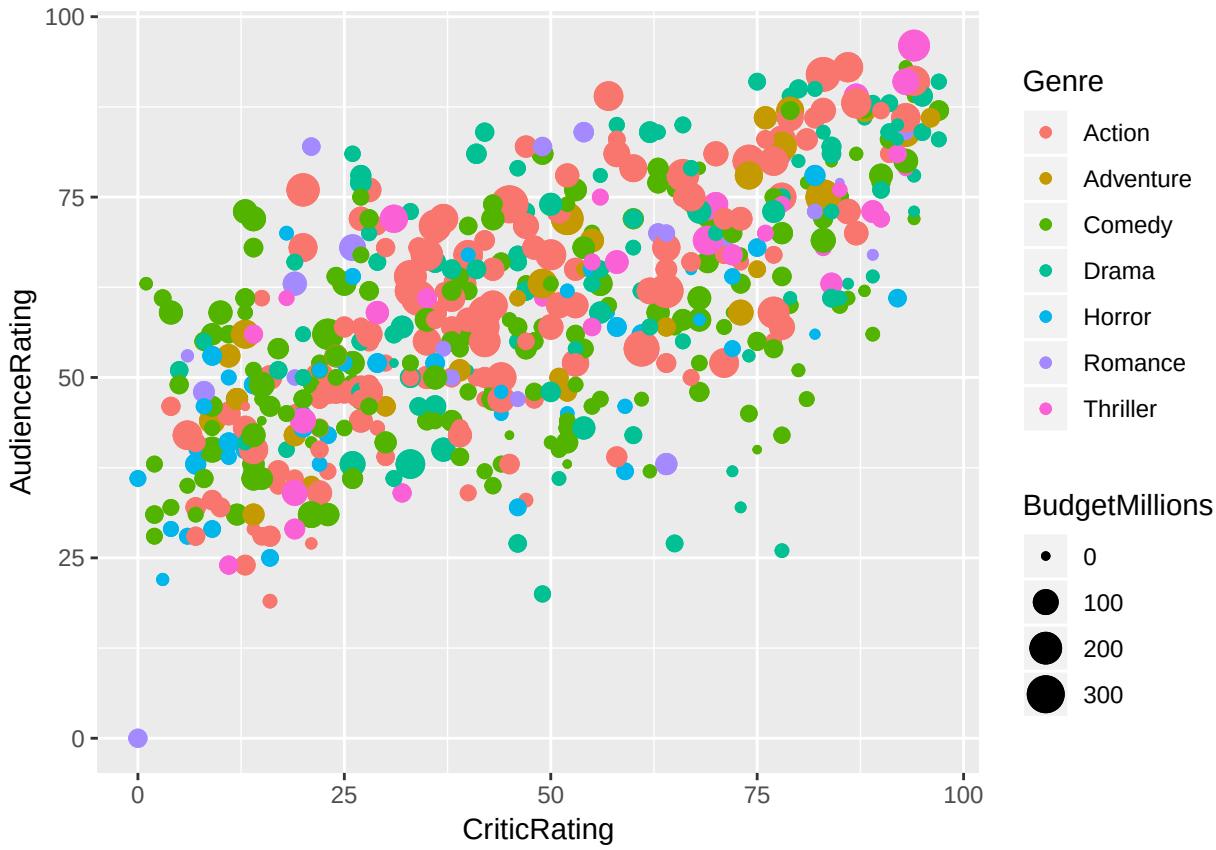



```
# Color based on Genre; Size based on Budget
ggplot(data=movies,aes(x=CriticRating,y=AudienceRating,
                       color=Genre,size=BudgetMillions)) +
  geom_point()
```

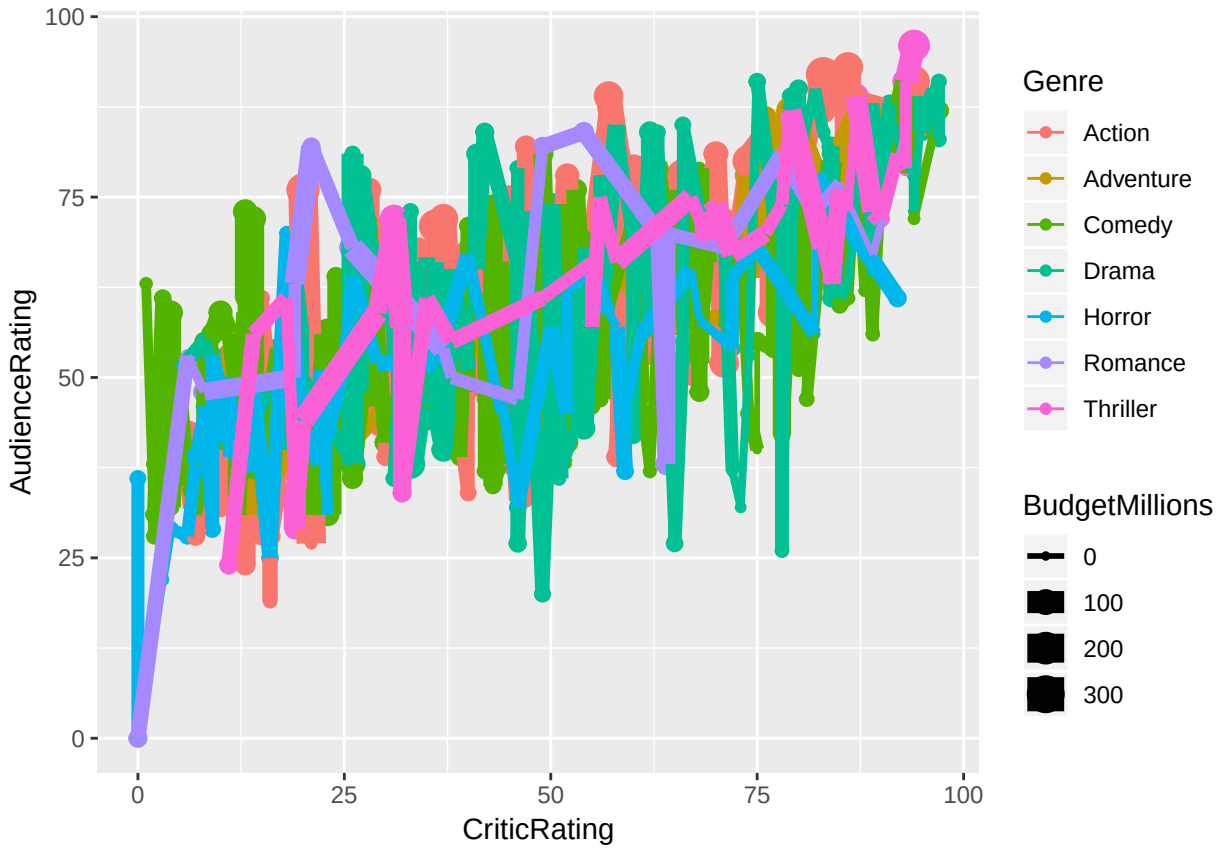


We can add Geom layer, such as points and lines, or multiple layers Yet, when we add points and lines, the plot is not very informative We should override Aesletics to make the plot more informative

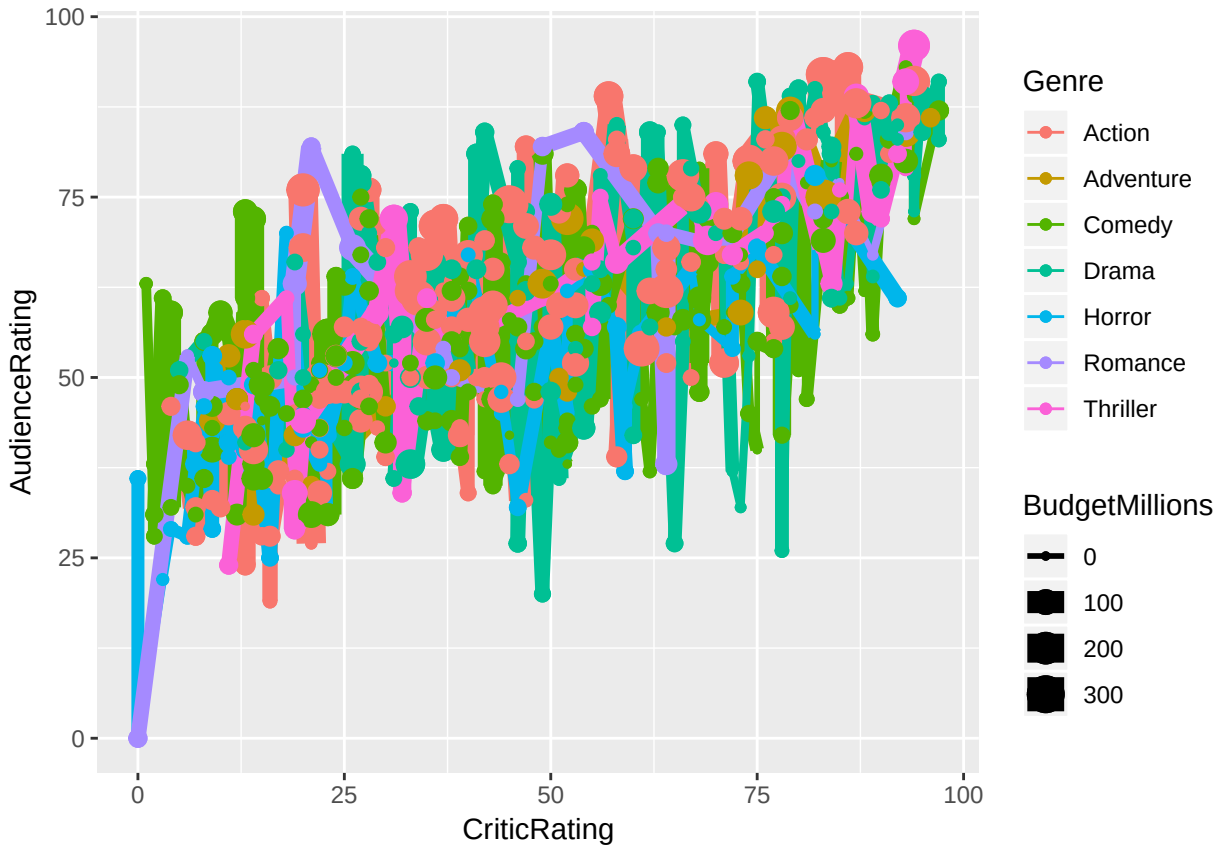
```
p<-ggplot(data=movies,aes(x=CriticRating,y=AudienceRating,
                          color=Genre,size=BudgetMillions)) +
  geom_point()

# Add Geom layer--Points
p+geom_point()
```



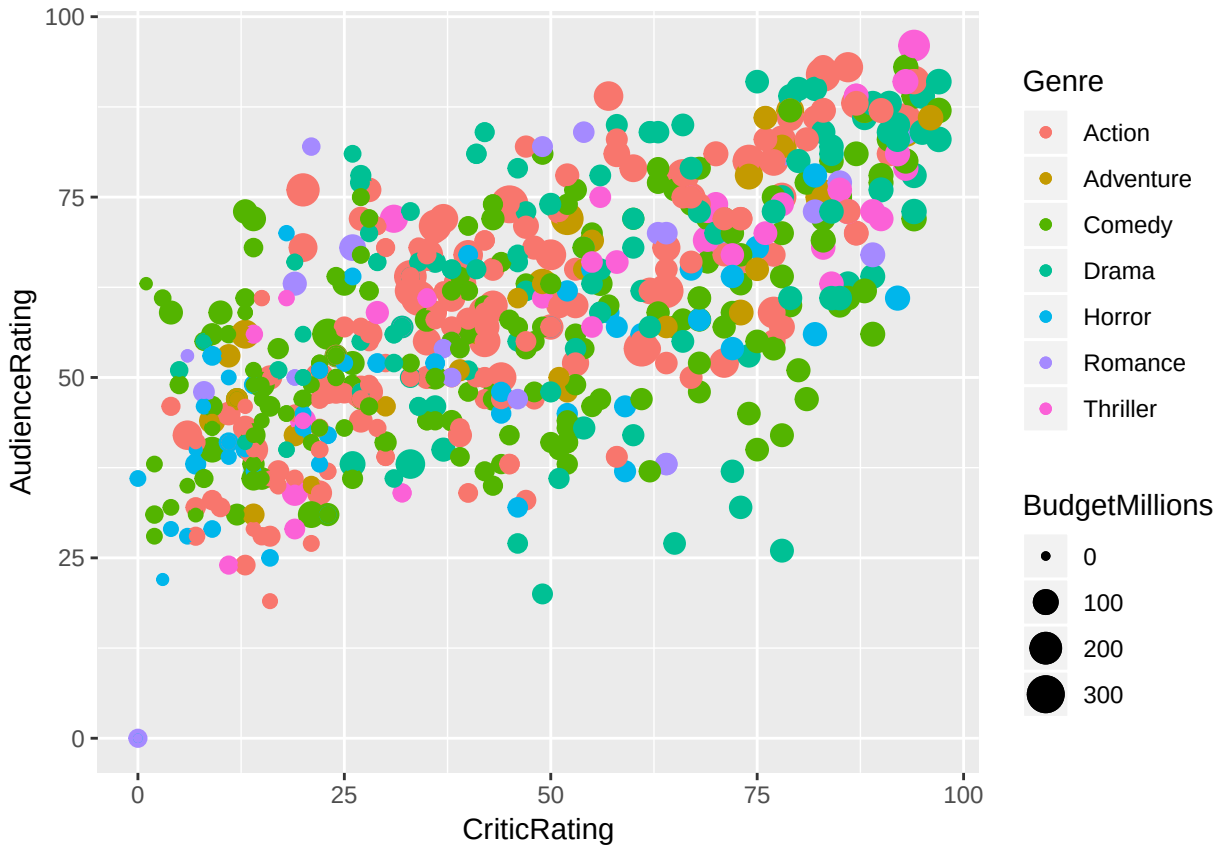

```
# Add Geom layer--Lines
p+geom_line()
```



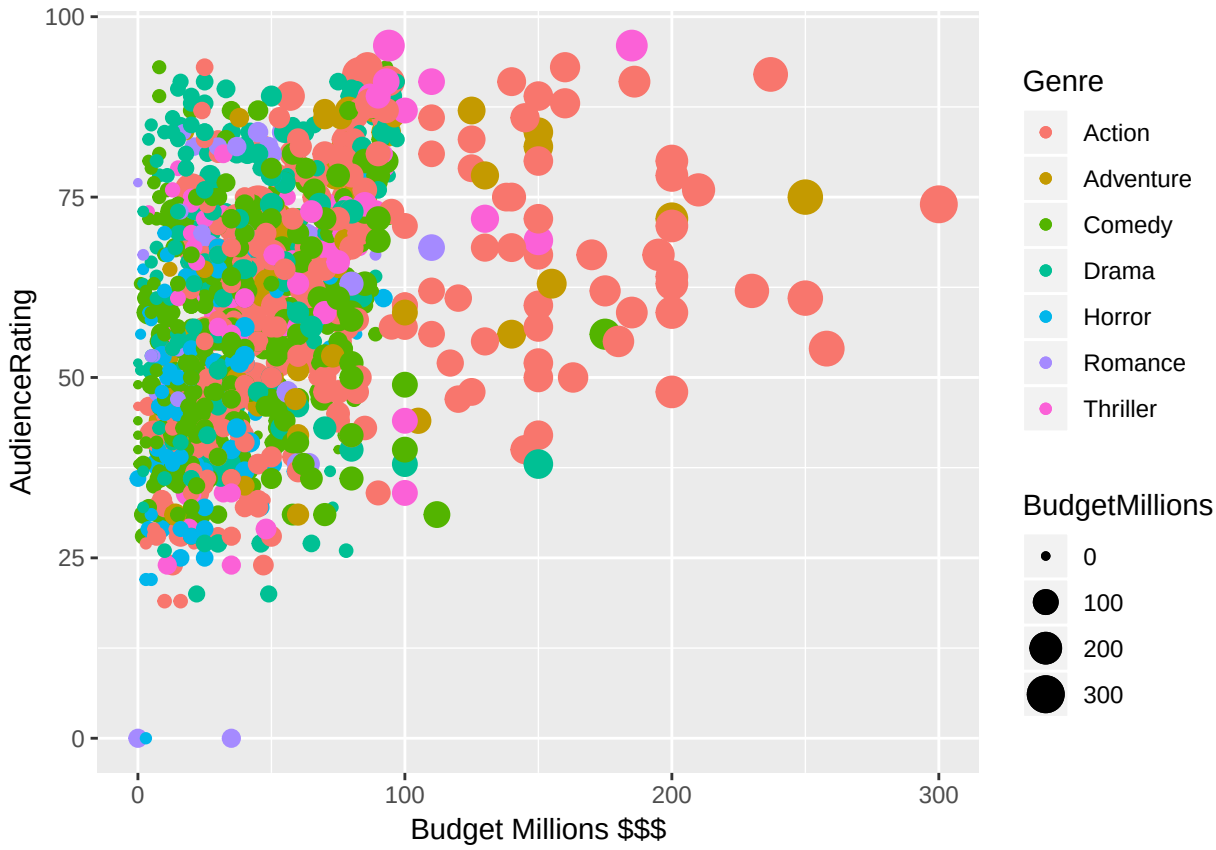

```
#Add multiple Geom layers
p+geom_line()+geom_point()
```



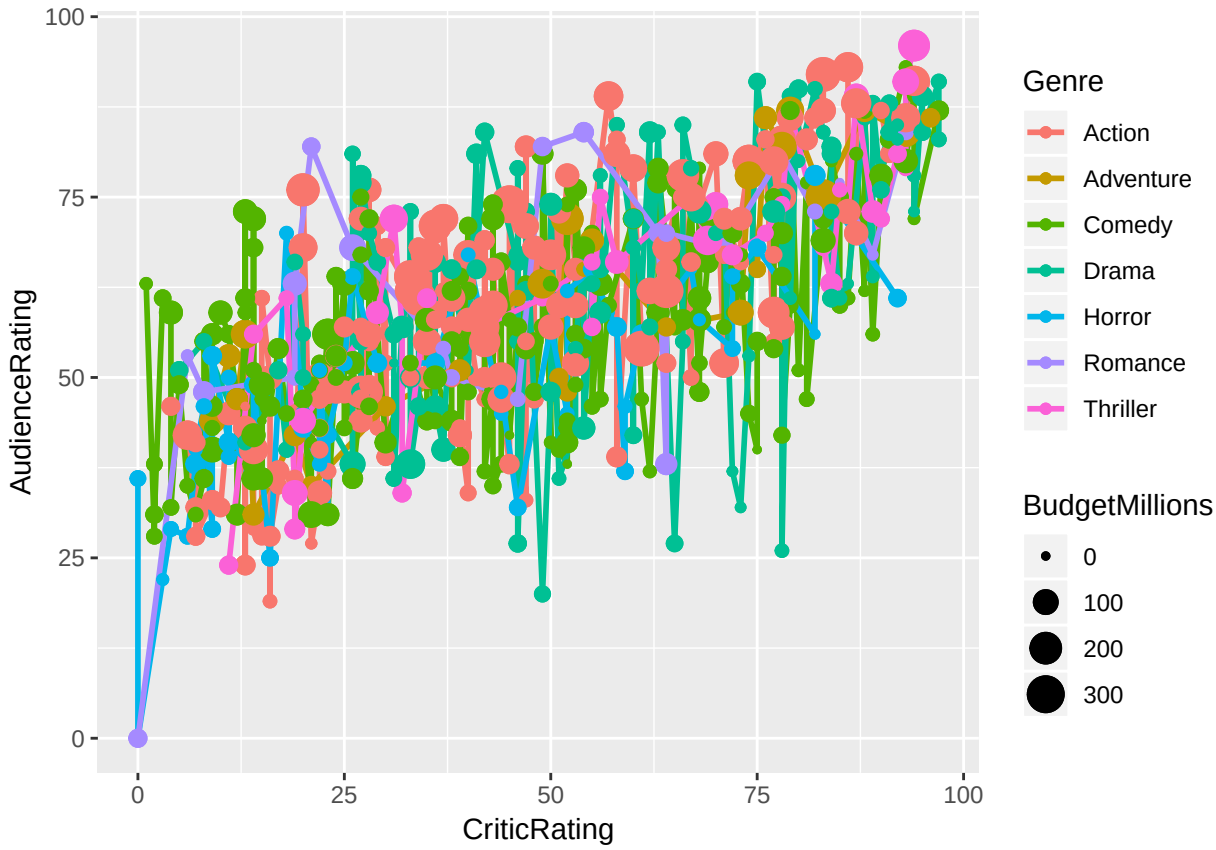

```
# Overriding Aesletics Example 1, override Size
p+geom_point(aes(size=CriticRating))
```




```
# Overriding Aesletics Example 2, override Color
# p+geom_point(aes(color=BudgetMillions))

# Overriding Aesletics Example 3, override X
p+geom_point(aes(x=BudgetMillions))+
  xlab("Budget Millions $$$")
```

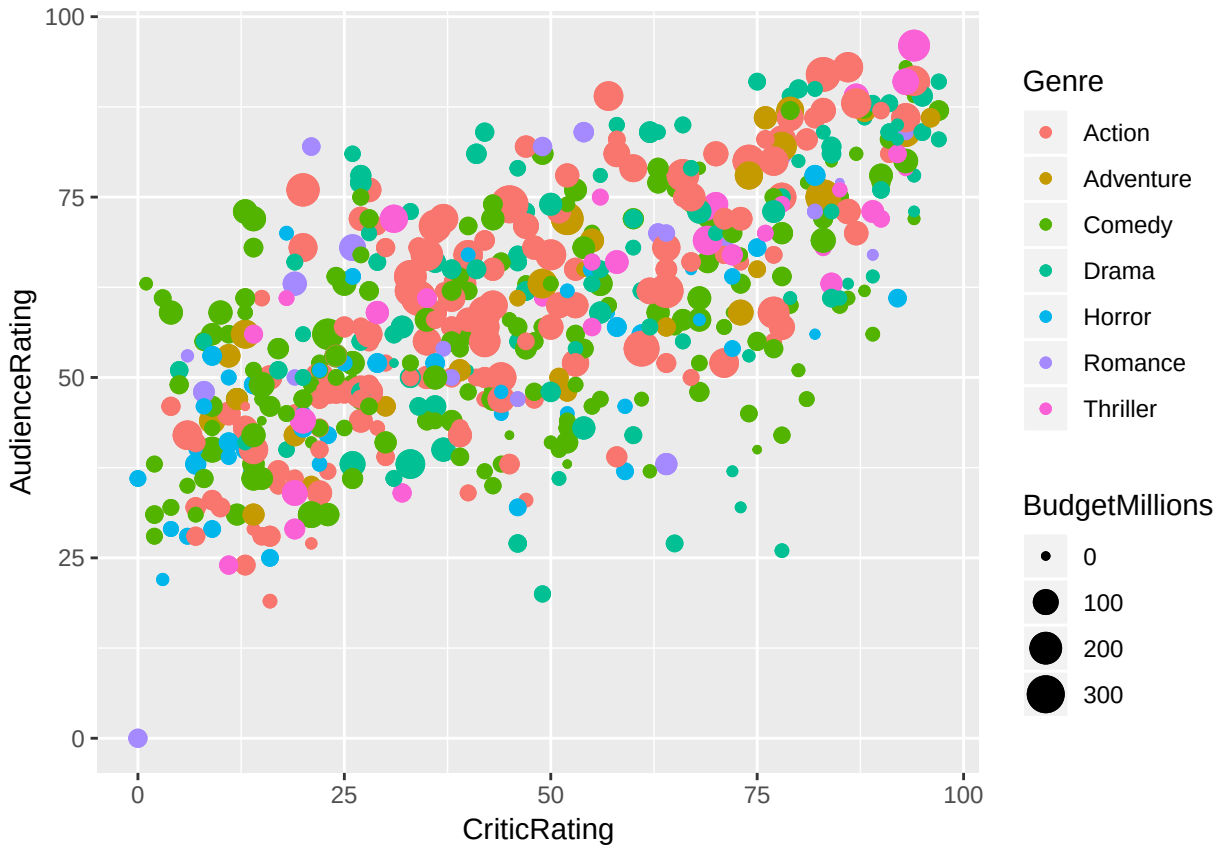



```
# Overriding Aesletics Example 4, reduce line size
p+geom_line(size=1) + geom_point()
```

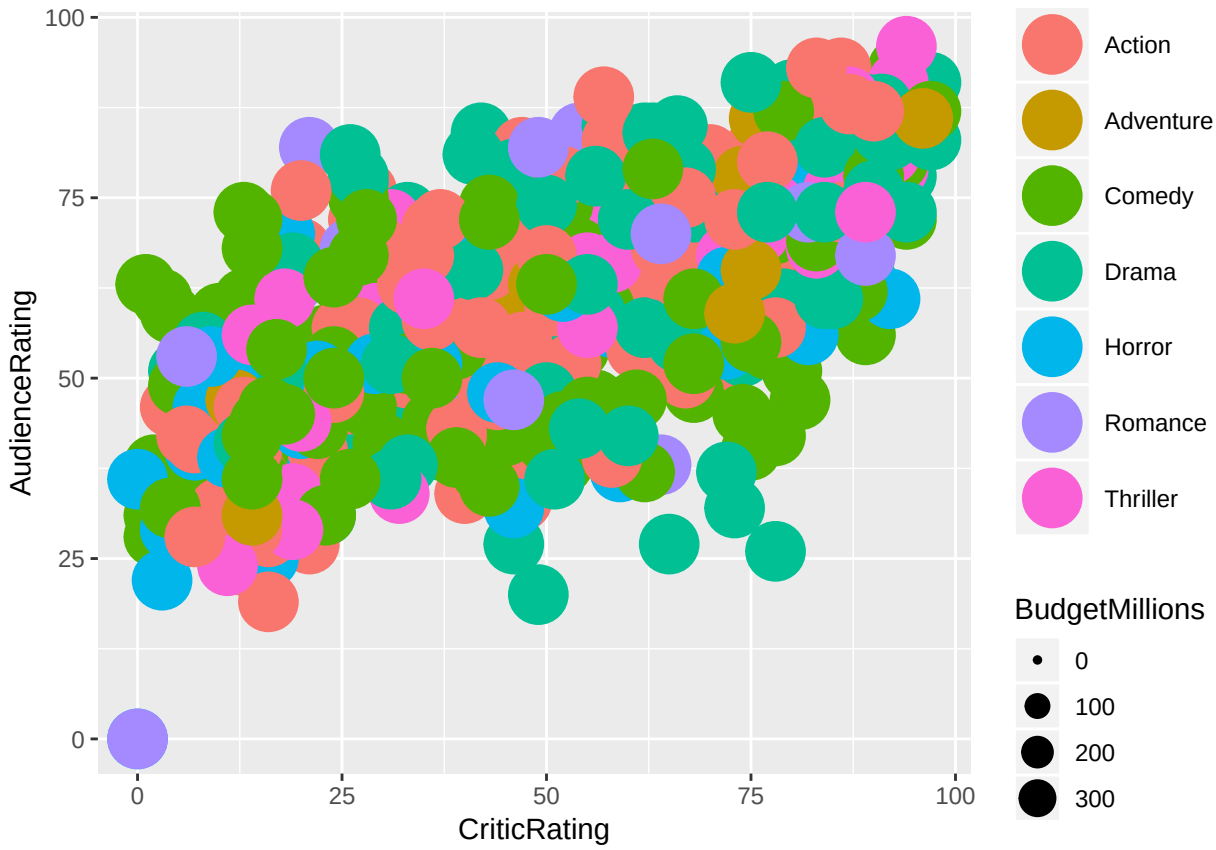


There are differences between mapping and setting When you want to set a color, you do NOT use aes When you want to map a color to a variable, you use aes

```
#1. Mapping
p+geom_point(aes(size=BudgetMillions))
```

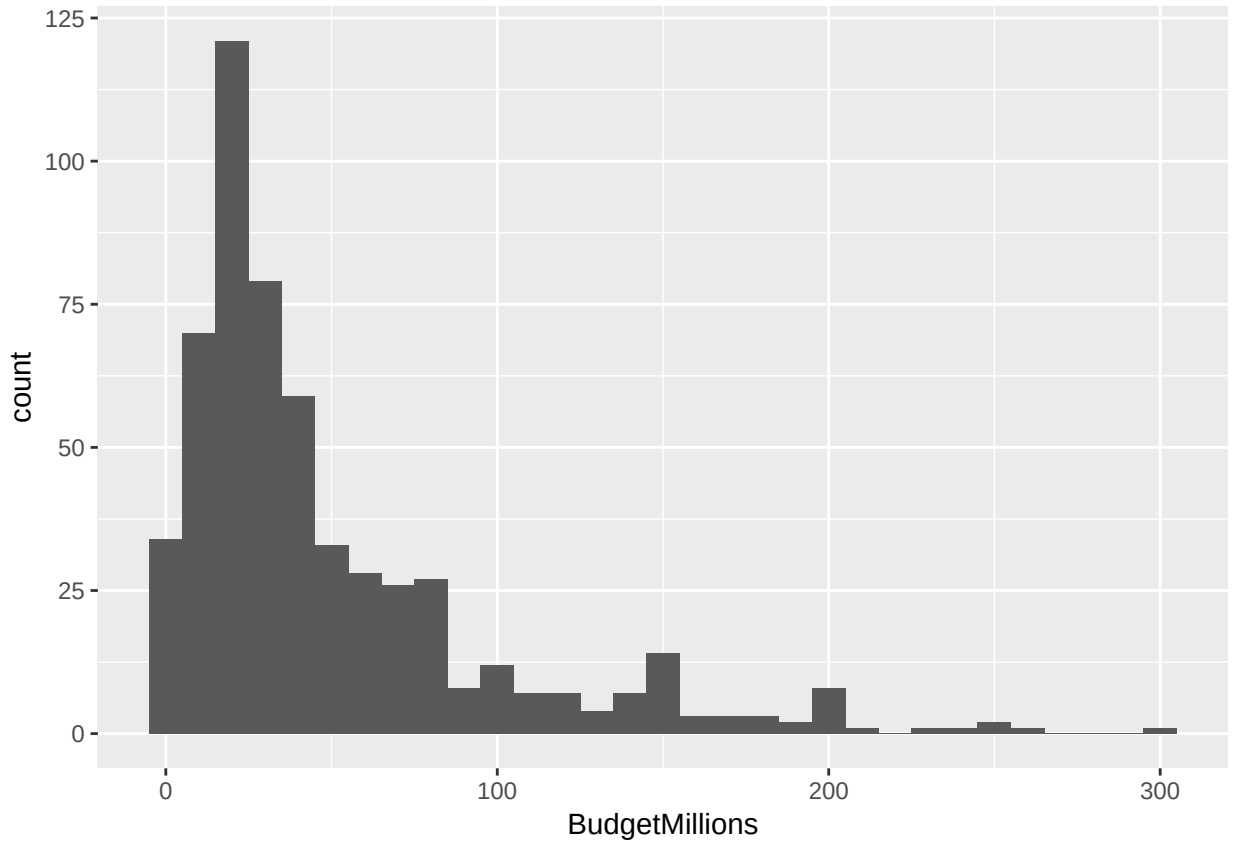



```
#2. Setting
p+geom_point(size=10)
```

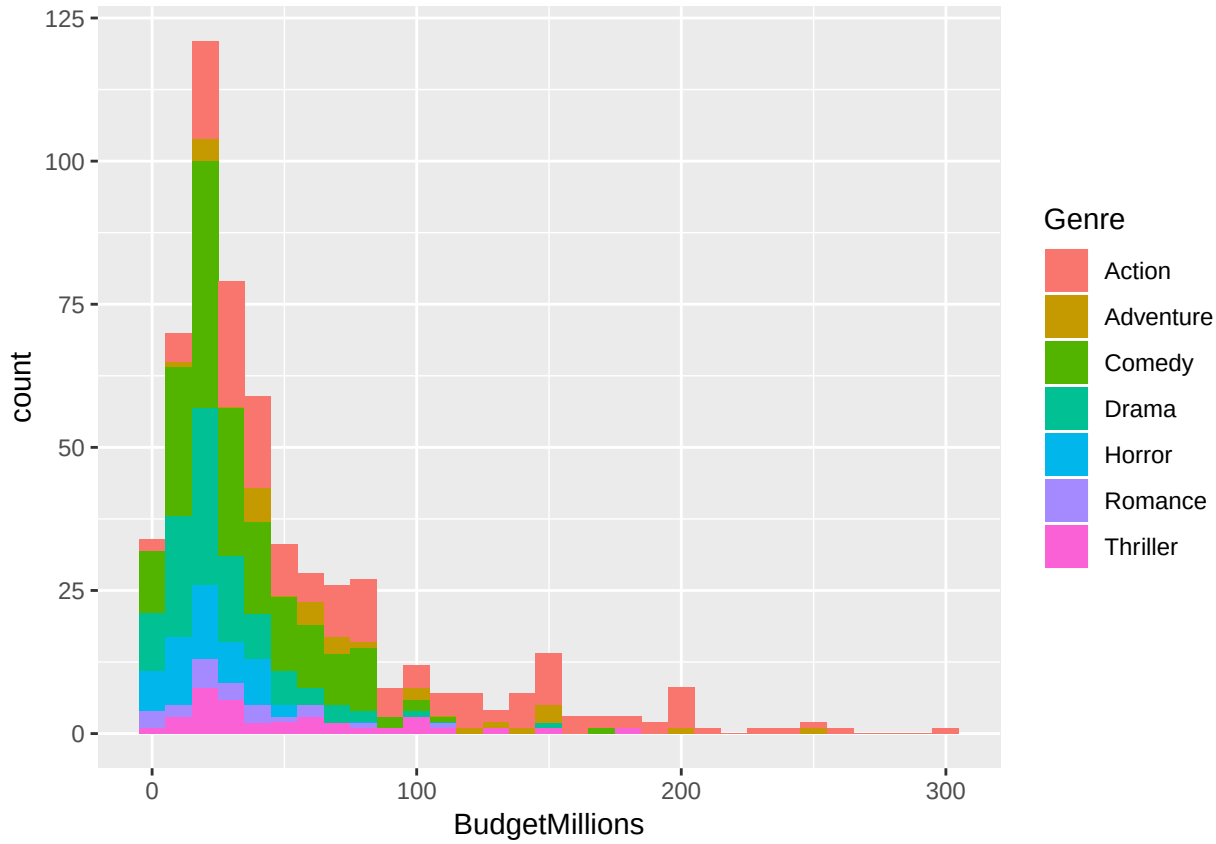


We can also create histographs to visualize the Audience's or Cristic's rating based on different predictors (e.g. Genre)

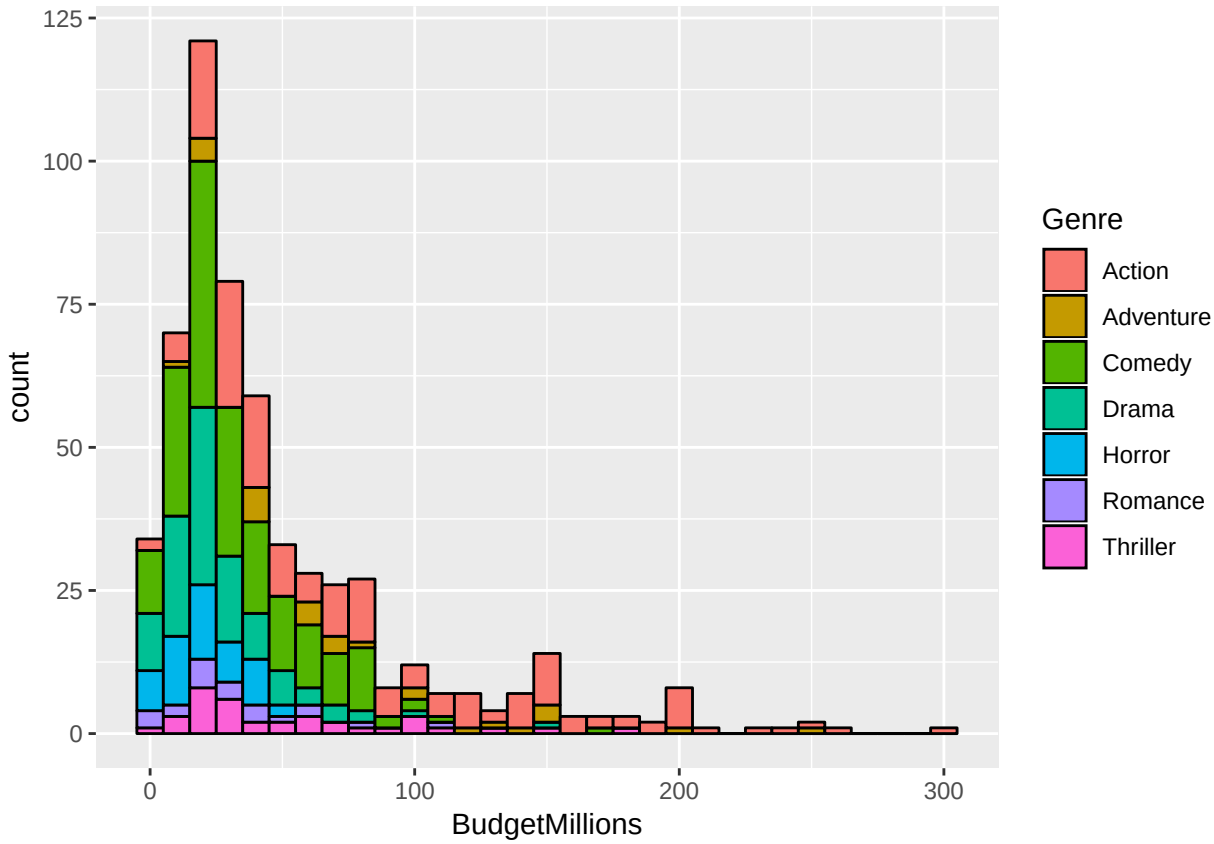
```
s<-ggplot(data=movies,aes(x=BudgetMillions))
s+geom_histogram(binwidth=10)
```




```
# Add color for each genre
s+geom_histogram(binwidth=10,aes(fill=Genre))
```

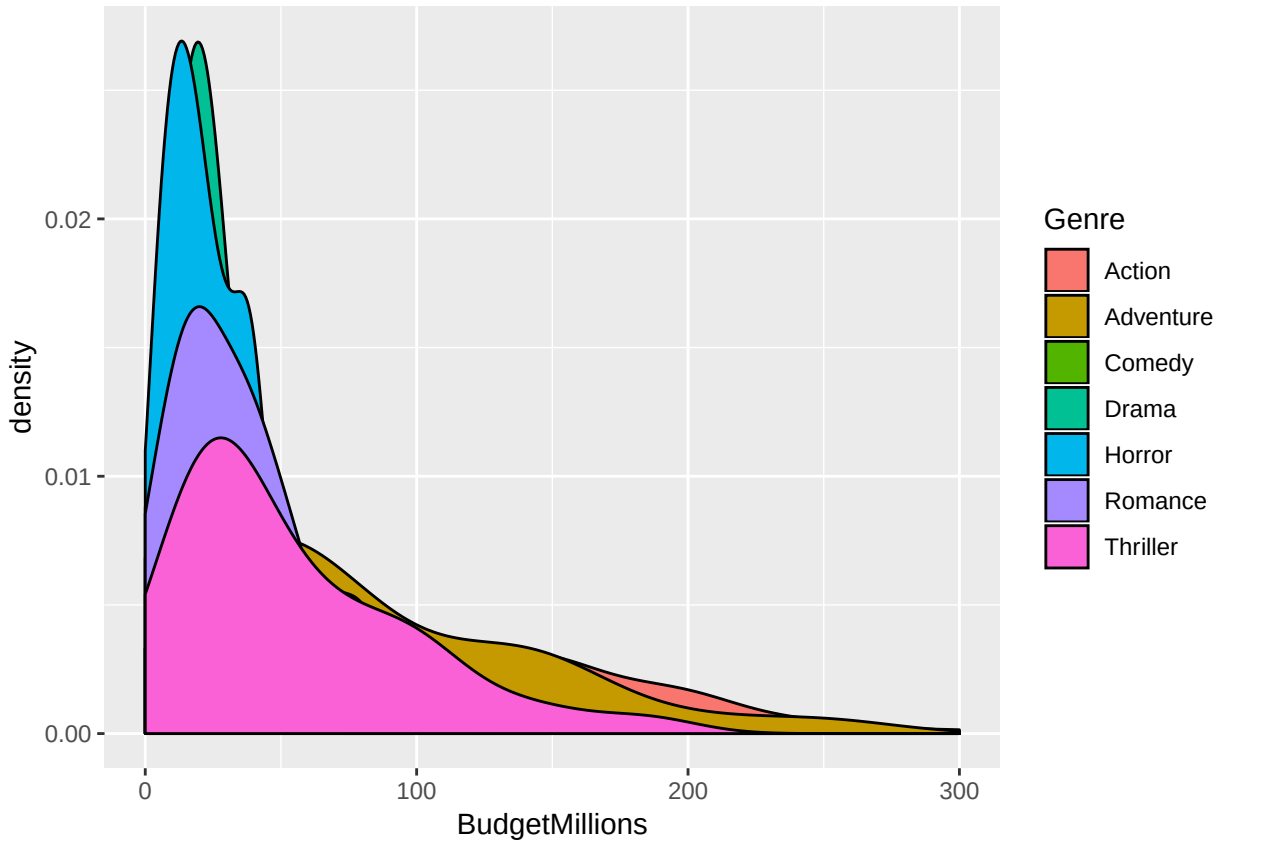



```
# Add a black boarder line
s+geom_histogram(binwidth=10,aes(fill=Genre),color="Black")
```

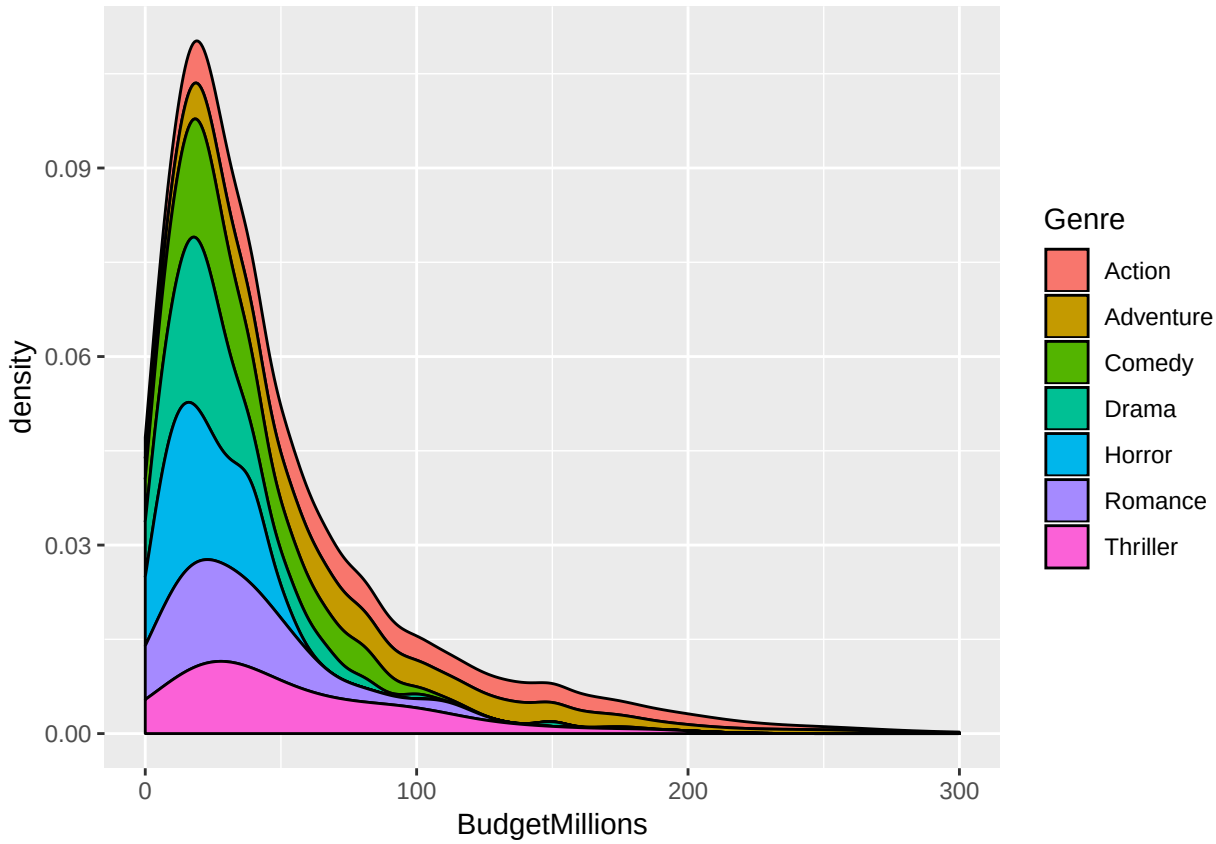


We can also create density charts to explore the data We found that Audiences' ratings showed a normal distribution while the Critics' ratings exhibited an uniform distribution

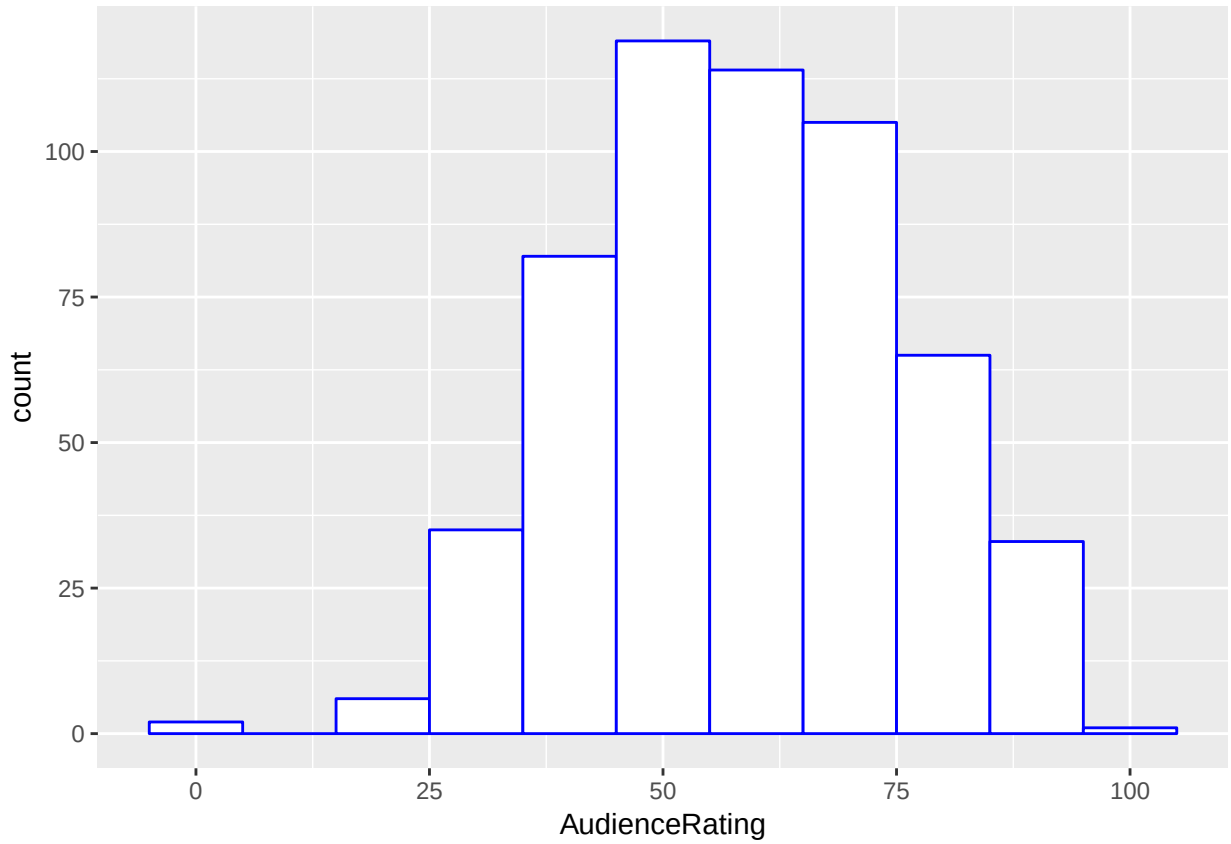
```
s<-ggplot(data=movies,aes(x=BudgetMillions))
s+geom_density(aes(fill=Genre))
```



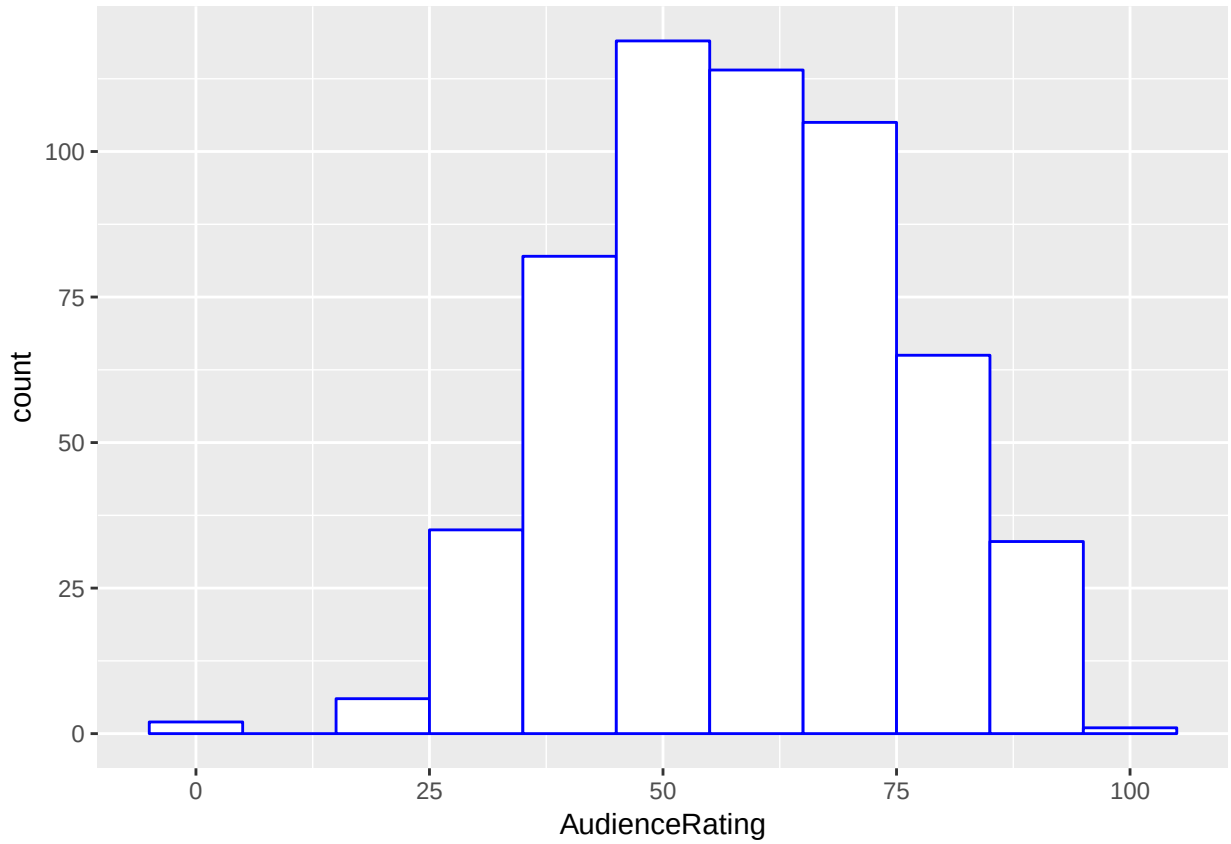

```
s+geom_density(aes(fill=Genre),position="stack")
```



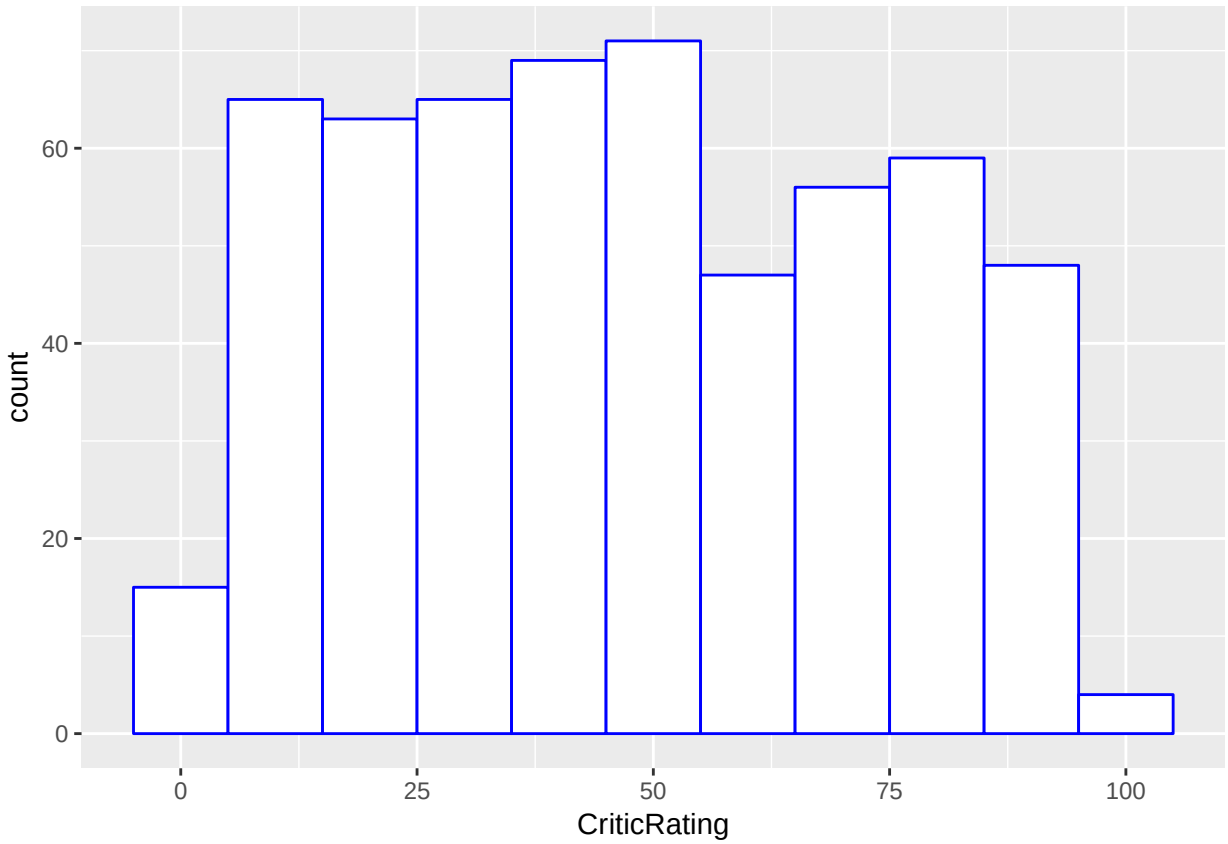

```
# Starting Layers Tips
t<- ggplot(data=movies,aes(x=AudienceRating))
t+geom_histogram(binwidth = 10,fill="White",color="Blue")
```




```
# Another way to make the same plot
t<- ggplot(data=movies)
# Distribution of the Audience Rating; The trend is normal distribution
t+geom_histogram(binwidth = 10,
                 aes(x=AudienceRating),
                     fill="White",color="Blue")
```




```
# Distribution of the Critics Rating; The trend is an uniform distribution
t+geom_histogram(binwidth = 10,
                 aes(x=CriticRating),
                 fill="White",color="Blue")
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.