Contents

[Scatterplot for two numeric variables 2](#_Toc17019515)

[setting the size of bubble by third numeric variable 3](#_Toc17019516)

[Scatterplot with linear regression line 4](#_Toc17019517)

[setting the size and shape of data points by third variable that is of factor type 5](#_Toc17019518)

[Histogram for a continuous variable 7](#_Toc17019519)

[Histogram with shading effect of fill based on frequency count 7](#_Toc17019520)

[Barplots (for categorical data) 8](#_Toc17019521)

[Barplots showing percentage distribution in each category (two categorical variables) 9](#_Toc17019522)

[Stacked barplots ( two categorical variables) 10](#_Toc17019523)

[Bar-plots stacked side by side instead of vertical stacking (two categorical variables) 10](#_Toc17019524)

[Boxplots (one categorical variable on x-axis and one continuous numeric variable on y-axis) 11](#_Toc17019525)

[To fill the boxplots based on variable used on x-axis 13](#_Toc17019526)

[Plotting two variables 14](#_Toc17019527)

[Facet\_grid 15](#_Toc17019528)

[One-dimensional faceting - Two numeric variables and one categorical variable 15](#_Toc17019529)

[2D faceting - Two numeric variables and two categorical variables 16](#_Toc17019530)

[Customizing the plot 17](#_Toc17019531)

[1. To label the country names on the points on the scatterplot, use geom\_text(aes(label=Country)) 17](#_Toc17019532)

[2. To label only selective points on the scatterplot 17](#_Toc17019533)

[3. check\_overlap will automatically make sure that these data points do not overlap 17](#_Toc17019534)

install.packages(‘ggplot2’)

install.packages(‘ggthemes’)

install.packages(‘data.table’)

library(ggplot2)

library(ggthemes)

library(data.table)

# Scatterplot for two numeric variables

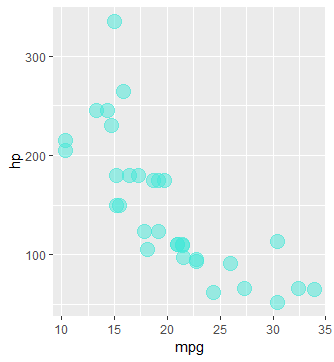
pl <- ggplot(data=mtcars, aes(x=mpg, y=hp)) + geom\_point(color = '#43e8d8', size=5, alpha = 0.5)

print(pl)

# **With alpha=0.5, overlapping points will be darker**

# to get more colors go to [www.color-hex.com](http://www.color-hex.com)

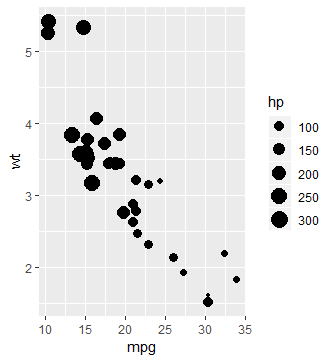
# to set the limits on x-axis and y-axis use coord\_cartesian(xlim = c(lower\_limit, upper\_limit), y=c(lower\_limit, upper\_limit)). So you can say pl + coord\_cartesian(xlim = c(1,4), ylim=c(15,30))



## setting the size of bubble by third numeric variable

pl <- ggplot(mtcars, aes(x=mpg, y=wt)) + geom\_point(aes(size=hp))

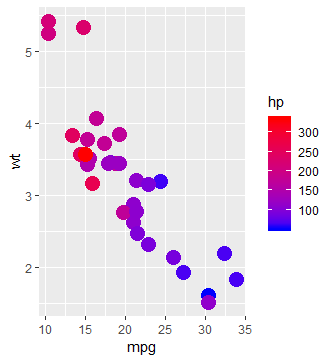
print(pl)



pl <- ggplot(mtcars, aes(x=mpg, y=wt)) + geom\_point(aes(color = hp), size=5)

pl2 <- pl + scale\_color\_gradient(low='blue', high='red')

print(pl2)



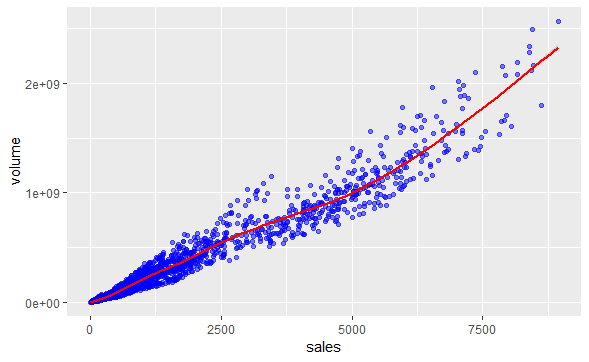
Analysis: Higher hp cars tend to be heavier with less mpg, lower hp cars have lower weight and higher mpg

## Scatterplot with linear regression line

pl <- ggplot(data=txhousing, aes(x=sales, y=volume))

pl2 <- pl + geom\_point(color='blue', alpha=0.5)

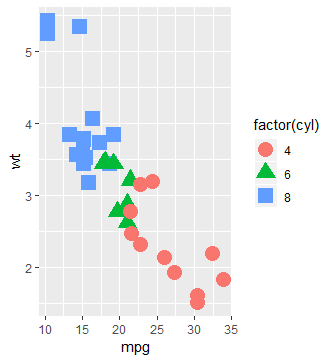
pl3 <- pl2 + geom\_smooth(color='red')

print(pl3)

## setting the size and shape of data points by third variable that is of factor type

pl <- ggplot(mtcars, aes(x=mpg, y=wt)) + geom\_point(aes(shape=factor(cyl), color = factor(cyl)), size=5)

print(pl)



# Histogram for a continuous variable

Install.packages(‘ggplot2movies’)

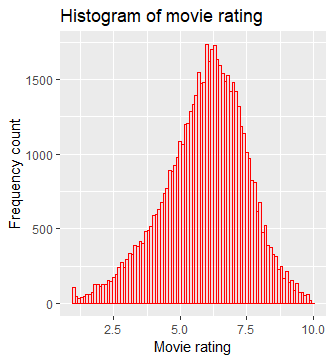
library(ggplot2)

pl <- ggplot(movies, aes(x=rating))

pl2<- pl + geom\_histogram(binwidth = 0.1, color = 'red', fill='pink', alpha = 0.4)

pl3<- pl2 + xlab('Movie rating') + ylab('Frequency count')

print(pl3 + ggtitle('Histogram of movie rating'))



# binwidth for more granular information

# alpha allows to set the transparency of the plotted data

Analysis: showing frequency count of the rating We have around 2000 count of 7.6 rating. Most common rating is around 6. Least common ratings are either zero or 10 which makes sense because we don’t have that many super awful movies or perfect movies.

## Histogram with shading effect of fill based on frequency count

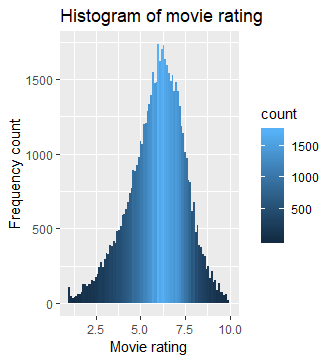
pl <- ggplot(movies, aes(x=rating))

pl2<- pl + geom\_histogram(binwidth = 0.1, **aes(fill=..count..))**

pl3<- pl2 + xlab('Movie rating') + ylab('Frequency count')

print(pl3 + ggtitle('Histogram of movie rating'))

Analysis: As the count gets higher, blue color becomes lighter and as the count gets lower, the blue color become darker



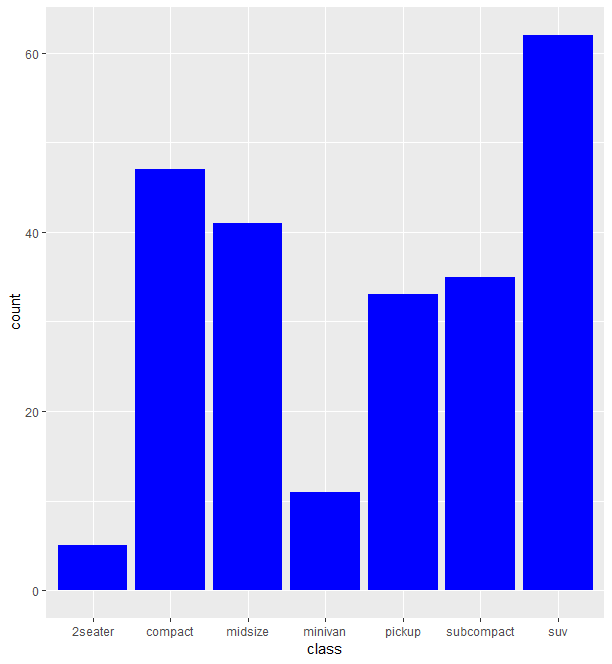
# Barplots (for categorical data)

library(ggplot2)

df<- mpg #mpg is a built-in dataframe in R

pl <- ggplot(data=df, aes(x=class)) + geom\_bar(fill='blue')

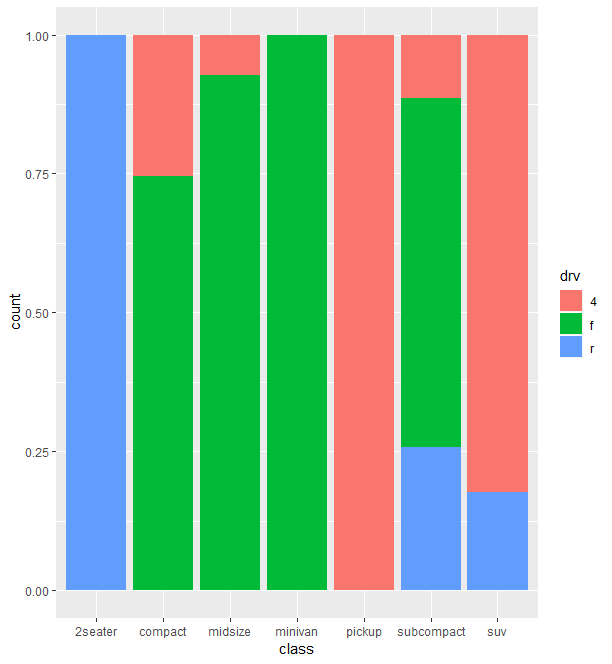
print(pl)



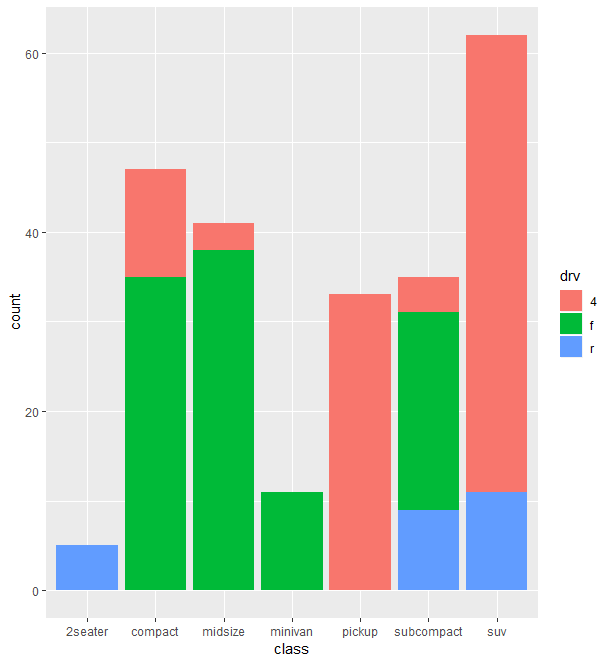
## Barplots showing percentage distribution in each category (two categorical variables)

pl <- ggplot(data=df, aes(x=class)) + geom\_bar(aes(fill=drv), position="fill")

print(pl)

**Analysis: 2 seater is 100% rear-wheel drive, minivan is 100% front-wheel drive, 75% cars in compact category are front-wheel drive.**

## Stacked barplots ( two categorical variables)

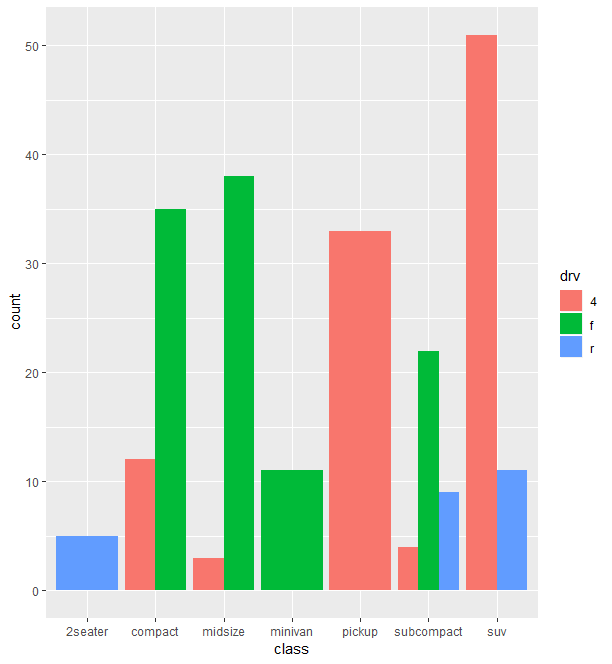


Analysis: 2 seater are all rear-wheel drive vs pickup cars are 4-wheel drive. Minivan is exclusively front-wheel drive.

## Bar-plots stacked side by side instead of vertical stacking (two categorical variables)

pl <- ggplot(data=df, aes(x=class)) + geom\_bar(aes(fill=drv), position="dodge")

print(pl)

Analysis: compact has three times as many front-wheel drive vs 4-wheel drive as denoted by count on y-axis

# Boxplots (one categorical variable on x-axis and one continuous numeric variable on y-axis)

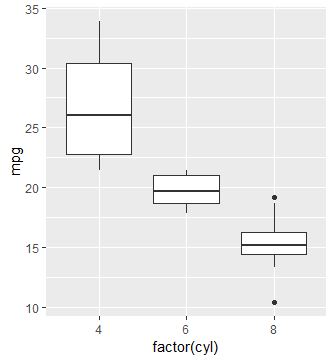
library(ggplot2)

df<-mtcars

**\*\*** variable on x-axis should be of factor type. Variable on y-axis should be of continuous numeric datatype

Analysis: Bottom and top of the box are always the Q1 and Q3, band inseide the box is Q2 or median. The ends represent 1.5 times the inter-quartile range (Q3-Q1) of the lower and upper quartiles.

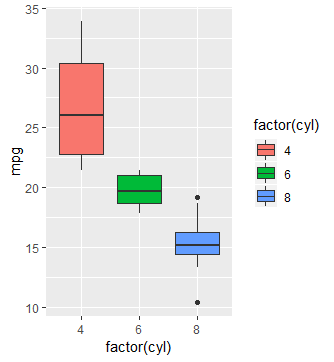
The dots represent outliers.



## To fill the boxplots based on variable used on x-axis

pl <- ggplot(data=df, aes(x=factor(cyl), y=mpg))

print(pl + geom\_boxplot(aes(fill=factor(cyl))))



# Plotting two variables

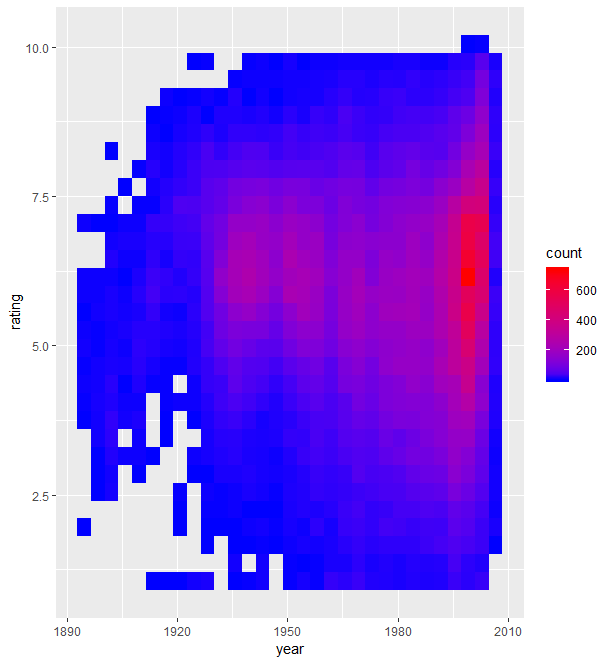
library(ggplot2movies)

df<-movies

pl <- ggplot(data=df, aes(x=year, y=rating))

pl2 <- pl + geom\_bin2d()

print(pl2 + scale\_fill\_gradient(high='red', low='blue'))



Analysis: We have more number of movies between rating 5.0 and 7.5 in the year 2000 (as indicated by the red color rectangles)

# Facet\_grid

## One-dimensional faceting - Two numeric variables and one categorical variable

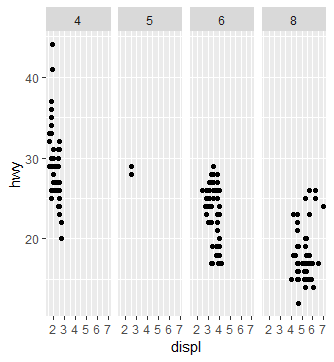
\*\*Int datatype variable on x-axis and y-axis,

df<-mpg

pl <- ggplot(data=df, aes(x=displ, y=hwy))

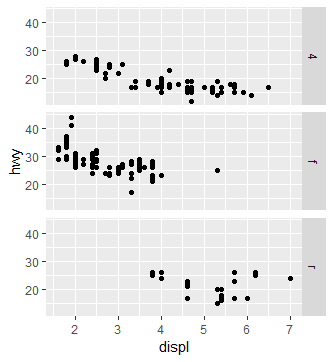
pl2 <- pl + geom\_point()

print(pl2 + facet\_grid(. ~ cyl))



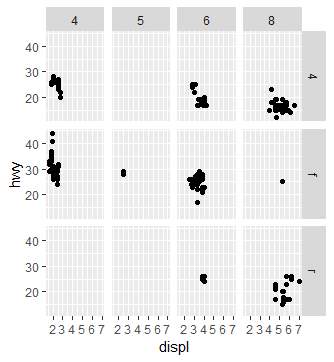
\*\*because cyl is placed after the ~ sign, faceting is done on the x-axis. Ig you want faceting to be done on y-axis then we can use the below command

ggplot(data=df, aes(x=displ, y=hwy)) + geom\_point() + facet\_grid(drv ~ .))



## 2D faceting - Two numeric variables and two categorical variables

print(ggplot(data=df, aes(x=displ, y=hwy)) + geom\_point() + facet\_grid(drv ~ cyl))



# Economist plot

library(dplyr)

library(ggplot2)

library(data.table)

library(ggthemes)

df <- fread('C:\\Users\\aabha.DESKTOP-HG6KK17\\Desktop\\Anita personal\\R-Course-HTML-Notes\\R-Course-HTML-Notes\\R-for-Data-Science-and-Machine-Learning\\Training Exercises\\Capstone and Data Viz Projects\\Data Visualization Project\\Economist\_Assignment\_Data.csv', drop=1)

head(df)

pl <- ggplot(data=df, aes(x=CPI, y=HDI, color=Region)) + geom\_point(size=4, shape=1) + theme\_economist\_white()

pl2 <- pl + geom\_smooth(aes(group=1), method='lm', formula= y ~ log(x), se = F, color='red')

pointsToLabel <- c("Russia", "Venezuela", "Iraq", "Myanmar", "Sudan",

"Afghanistan", "Congo", "Greece", "Argentina", "Brazil",

"India", "Italy", "China", "South Africa", "Spane",

"Botswana", "Cape Verde", "Bhutan", "Rwanda", "France",

"United States", "Germany", "Britain", "Barbados", "Norway", "Japan",

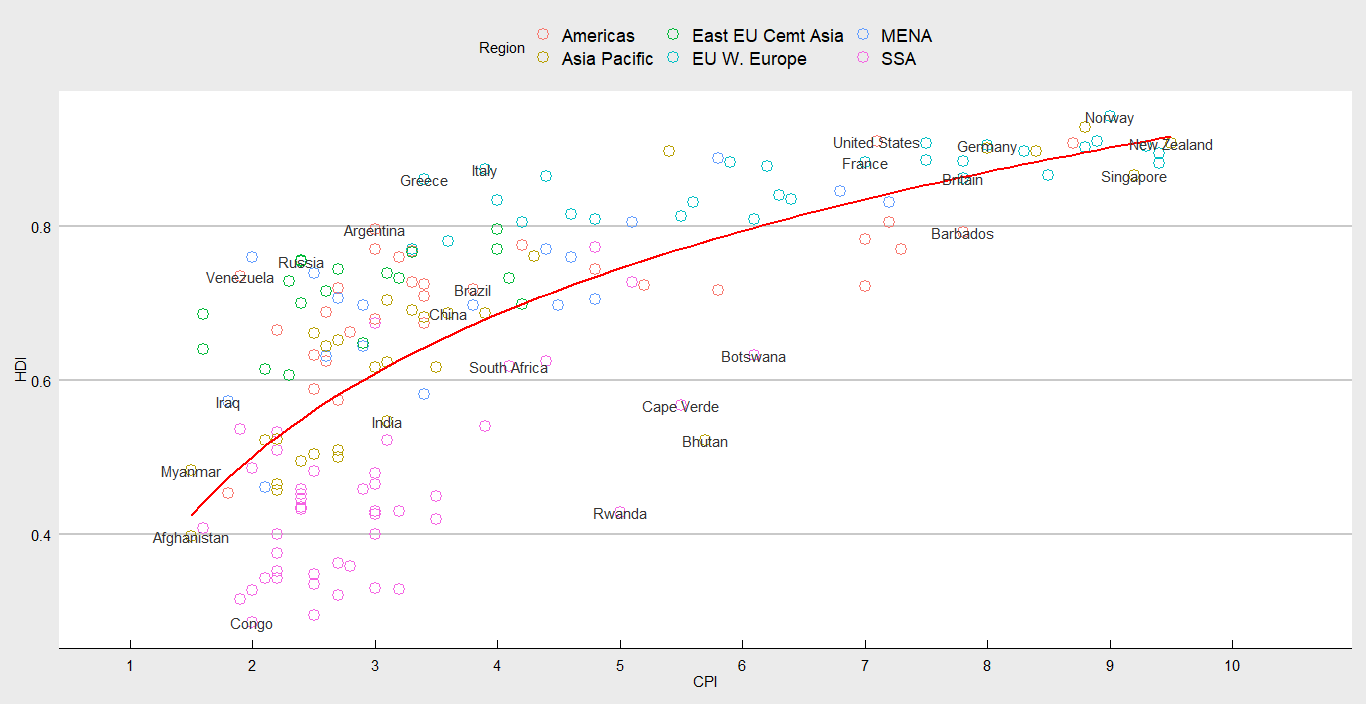
"New Zealand", "Singapore")

pl3 <- pl2 + geom\_text(aes(label = Country), color = "gray20",

data = subset(df, Country %in% pointsToLabel),check\_overlap = TRUE)

pl4 <- pl3 + scale\_x\_continuous(limits=c(.9,10.5), breaks = 1:10)

print(pl4)



**Explanation:**

* shape =1 in geom\_point() gives hollow circles
* geom\_smooth() adds the trend line
* geom\_smooth(aes(group=1) was used to specify that use the 1st variable in the data (which was country) to get one trend line for all the countries together
* method =’lm’ is for linear model
* se =F removes the grey area around the trend line
* formula = y ~ log(x)

# Converting a ggplot2 plot to an interactive visualization with plotly

install.packages(‘plotly’)

library(plotly)

library(ggplot2)

pl <- ggplot(mtcars, aes(x=mpg, y=wt)) + geom\_point()

gpl <- ggplotly(pl) # converting plot created in ggplot to plotly

print(gpl)

Tutorial for making different types of interactive plots can be accessed at <https://plot.ly/ggplot2/>

# Customizing the plot

## To label the country names on the points on the scatterplot, use geom\_text(aes(label=Country))

## To label only selective points on the scatterplot ((see example in economist plot)

geom\_text(aes(label = Country), color = "gray20",

data = subset(df, Country %in% pointsToLabel),check\_overlap = TRUE)

pointsToLabel <- c("Russia", "Venezuela", "Iraq", "Myanmar", "Sudan",

"Afghanistan", "Congo", "Greece", "Argentina", "Brazil",

"India", "Italy", "China", "South Africa", "Spane",

"Botswana", "Cape Verde", "Bhutan", "Rwanda", "France",

"United States", "Germany", "Britain", "Barbados", "Norway", "Japan",

"New Zealand", "Singapore")

So I am going to label the points with the country name, font color will be gray20, data for this labeling is a subset from the df only those countries that are in pointsToLabel

## check\_overlap will automatically make sure that these data points do not overlap(see example in economist plot)

## adding a theme to the plot(see example in economist plot)

you can add different themes to the plots by adding theme parameter like theme\_ fivethirtyeight() or theme\_bw() or theme\_dark() etc.

## add a title to the plot

ggtitle('Histogram of movie rating')

## adding x-axis labels and y-axis labels

## if you want x-axis values to go from 0 to 10 (see example in economist plot)

scale\_x\_continuous(limits=c(.9,10.5), breaks = 1:10)

so on the x-axis, I am going from little less than 1 to little more than 10. And then I am specifying that I want the breakpoints to be at 1,2,3,4,5,6,7,8,9,10