

practical 5

Harshal Maratthe

2023-09-15

```
#install.packages("ggplot2")
#install.packages("plotly")
library("ggplot2")

## Warning: package 'ggplot2' was built under R version 4.2.3

library("plotly")

## Warning: package 'plotly' was built under R version 4.2.3

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

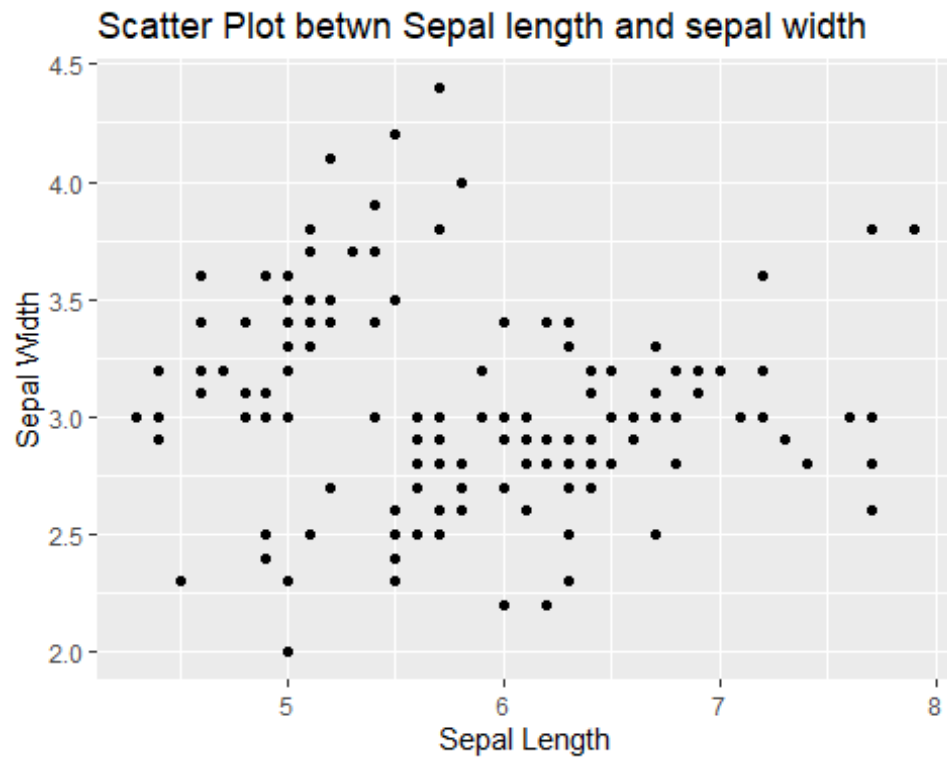
## The following object is masked from 'package:graphics':
##
##     layout

#install.packages("webshot2")
```

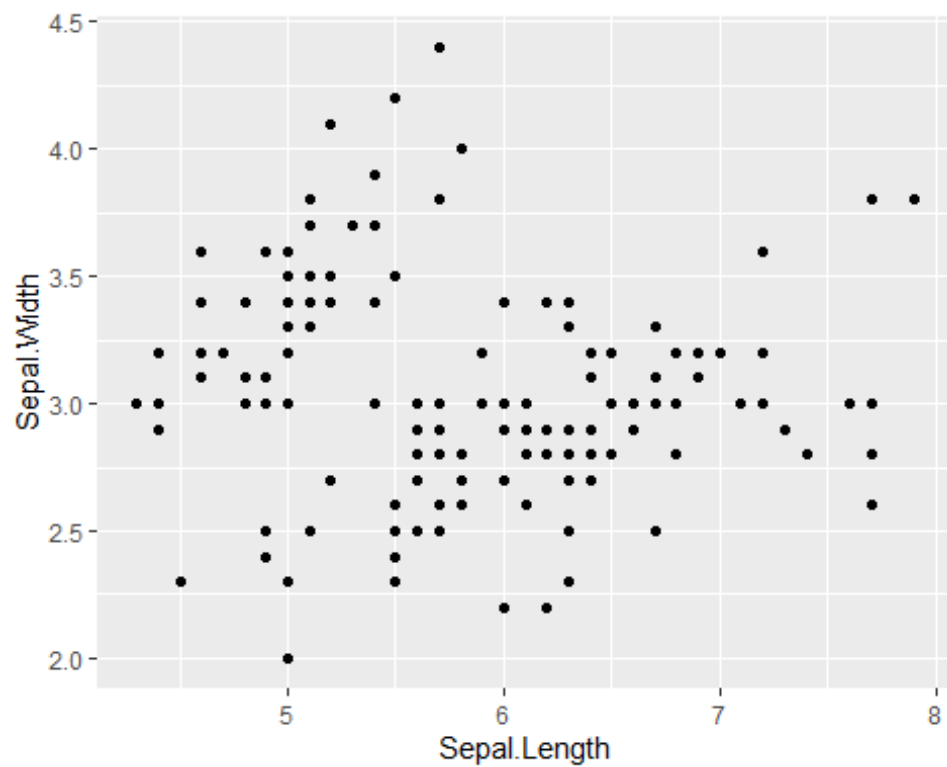
#Q1

```
#View(iris)
#Q1Use the iris dataset to create a scatter plot between Sepal.Length and
Sepal.Width

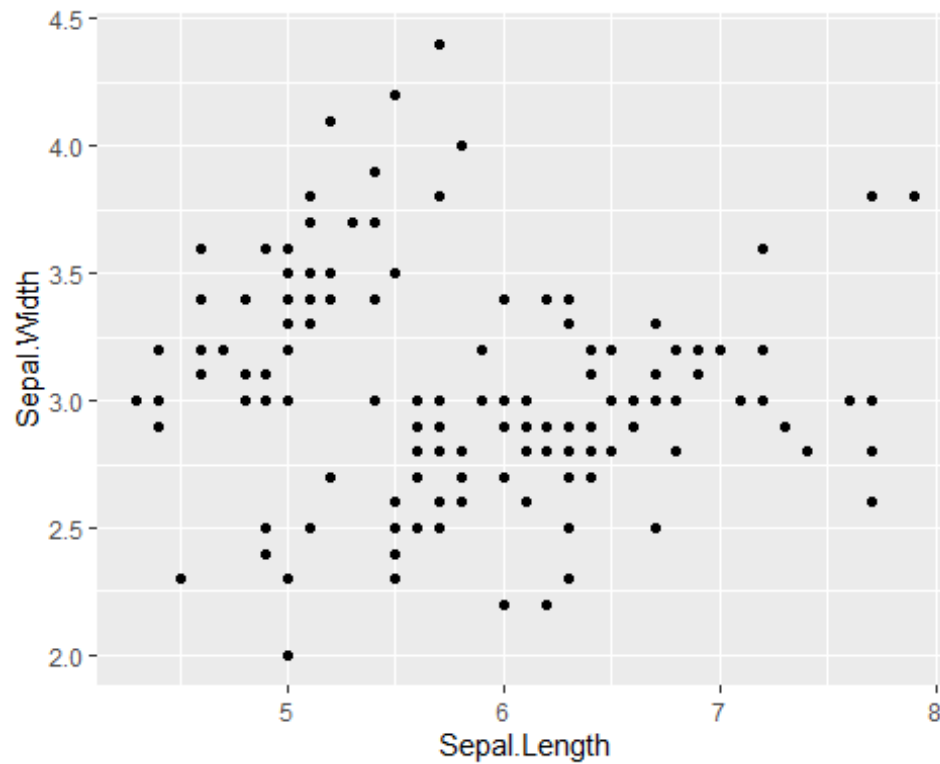
#ggplot()
ggplot(data=iris)+
  geom_point(mapping=aes(x=Sepal.Length,y=Sepal.Width))+
  labs(title = "Scatter Plot betwn Sepal length and sepal width",x="Sepal
Length",y="Sepal Width")
```



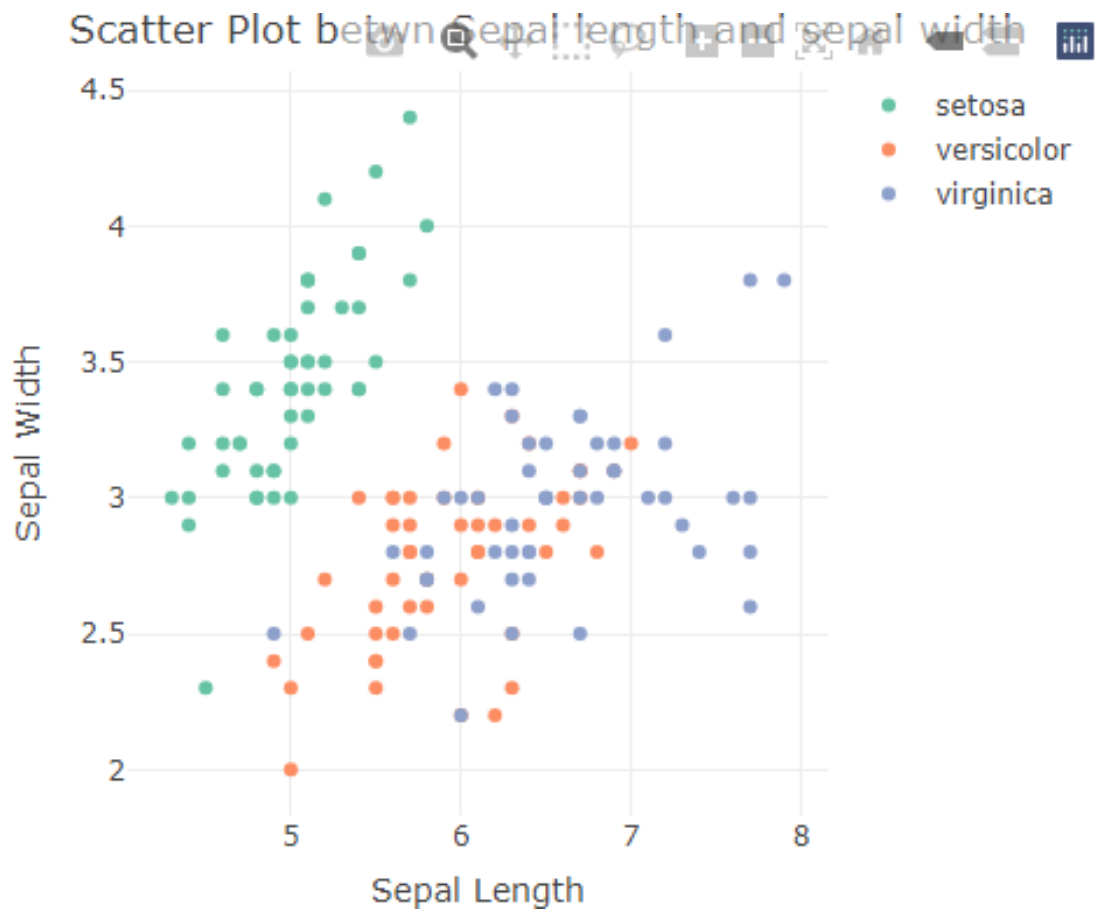
```
#or  
ggplot(data=iris,mapping=aes(x=Sepal.Length,y=Sepal.Width))+  
  geom_point()
```



```
#or
ggplot(data=iris,aes(x=Sepal.Length,y=Sepal.Width))+
  geom_point()
```



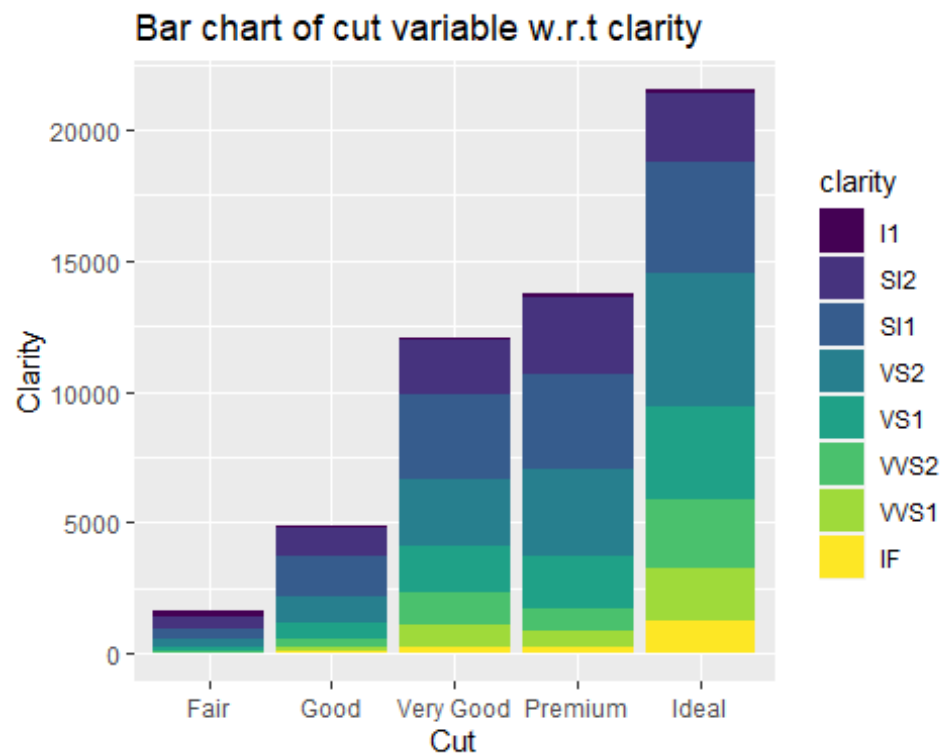
```
#plotly()
plot_ly(data=iris,x=~Sepal.Length,y=~Sepal.Width,type="scatter",mode="markers",
,color = ~Species)%>% layout(title="Scatter Plot betwn Sepal length and
sepal width",xaxis=list(title="Sepal Length"),yaxis=list(title="Sepal
Width"))
```



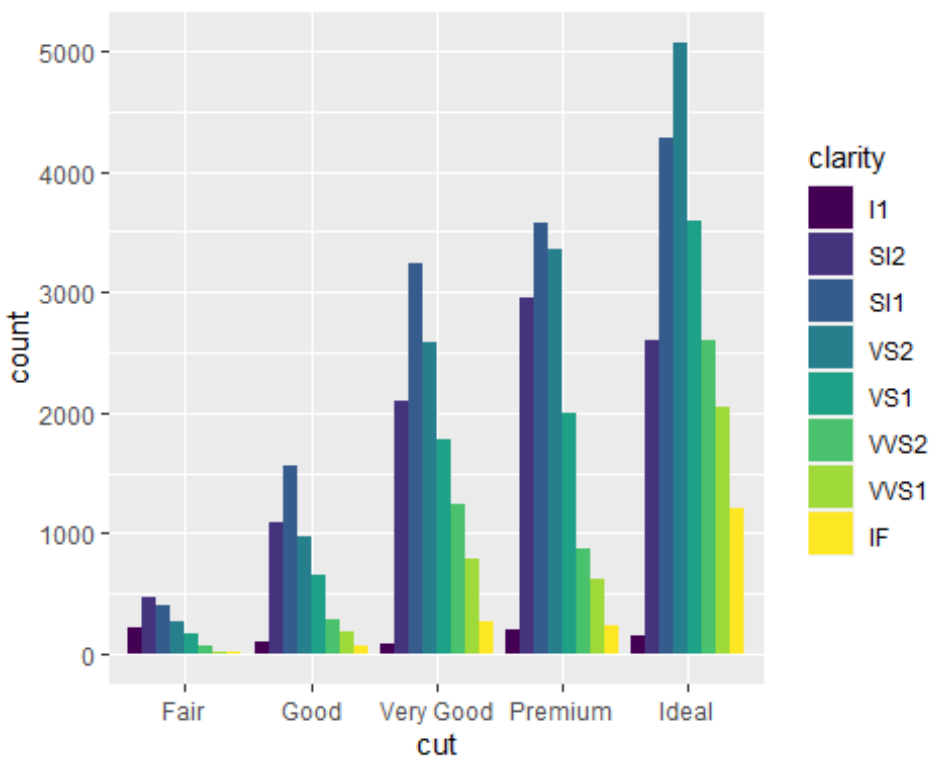
#Q2

#Q2 Create a bar chart using the diamonds dataset, displaying the cut variable with respect to clarity.

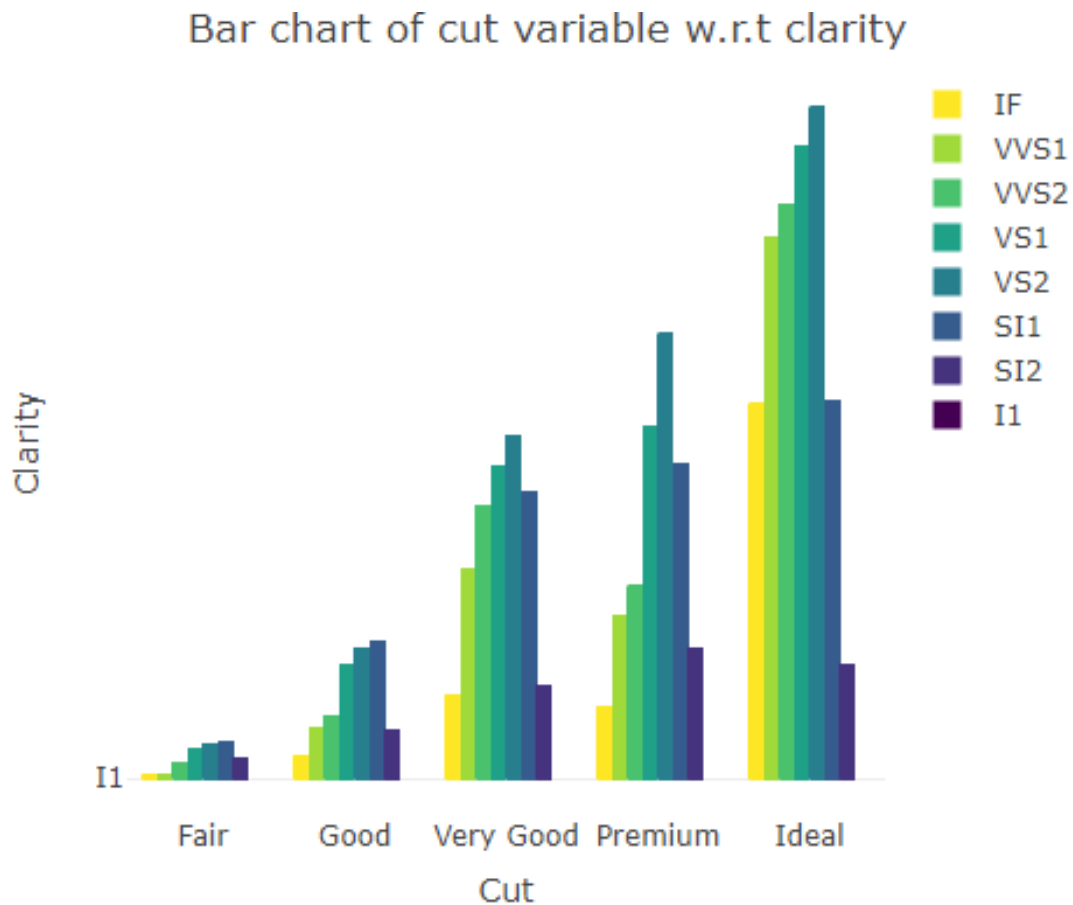
```
#View(diamonds)
ggplot(data=diamonds)+
  geom_bar(mapping=aes(x=cut,fill=clarity))+
  labs(title = "Bar chart of cut variable w.r.t clarity",x="Cut",y="Clarity")
```



```
ggplot(data=diamonds,aes(x=cut,fill=clarity))+  
  geom_bar(position="dodge")
```



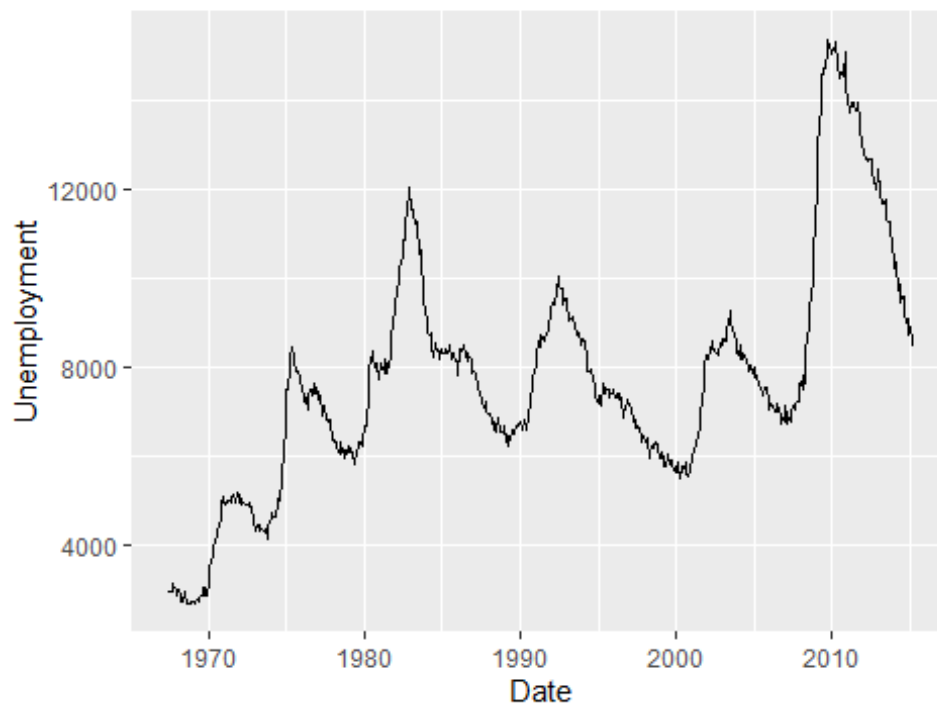
```
plot_ly(data=diamonds,x=~cut,y=~clarity,type="bar",color = ~clarity)%>%
layout(title="Bar chart of cut variable w.r.t
clarity",xaxis=list(title="Cut"),yaxis=list(title="Clarity"))
```



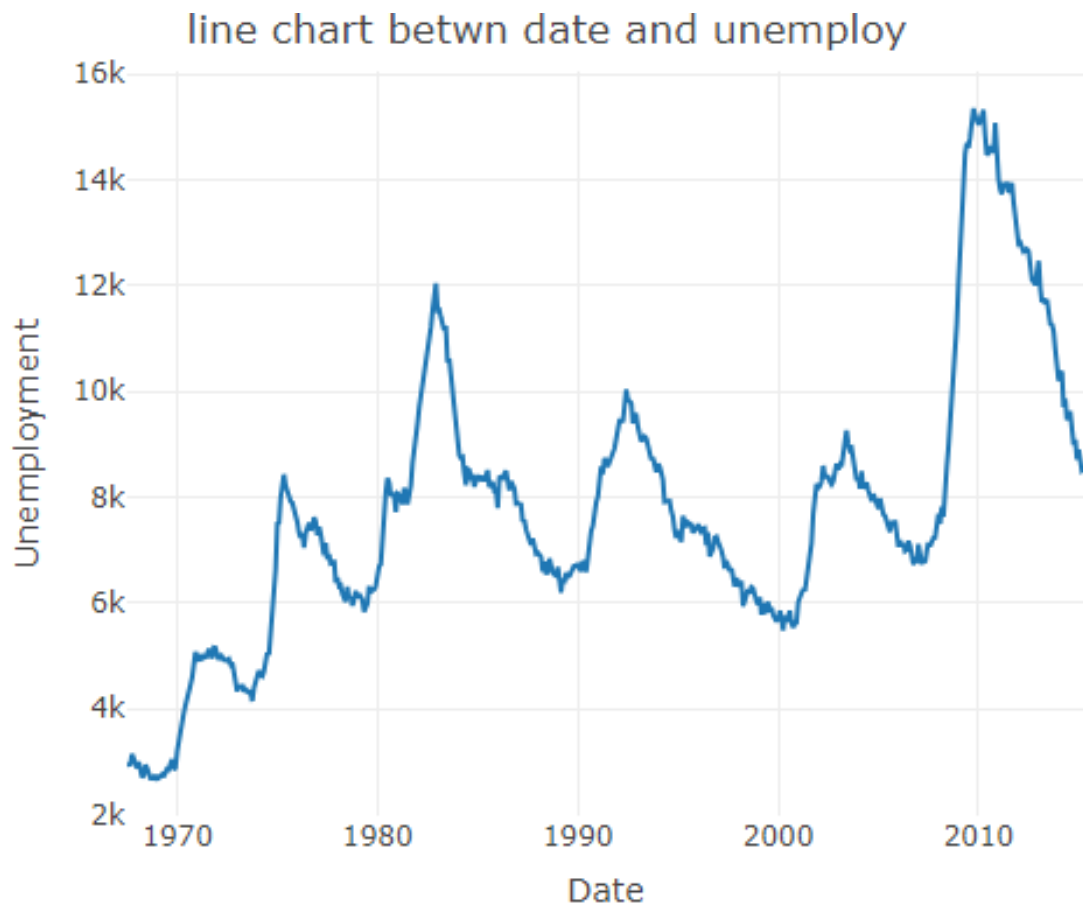
#Q3

```
#Q3Generate a line chart using the economics dataset to visualize the
relationship between the date variable and the unemploy variable
#View(economics)
ggplot(data=economics)+
  geom_line(mapping=aes(x=date,y=unemploy))+
  labs(title = "line chart betwn date and
unemploy",x="Date",y="Unemployment")
```

line chart betwn date and unemploy

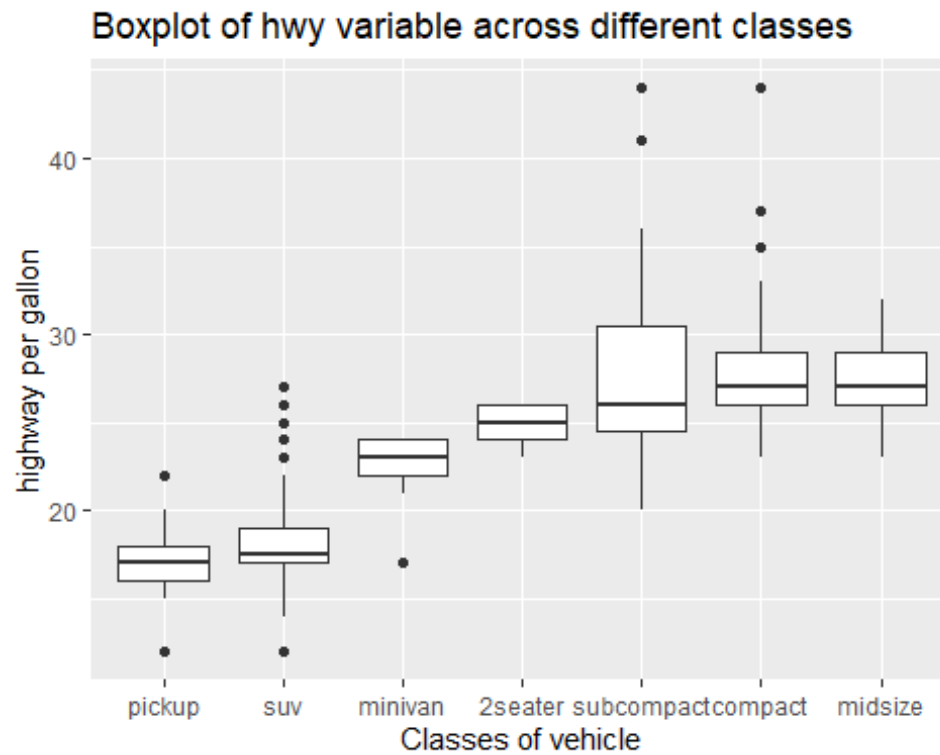


```
plot_ly(data=economics,x=~date,y=~unemploy,type="scatter",mode="line") %>%  
layout(title="line chart betwn date and  
unemploy",xaxis=list(title="Date"),yaxis=list(title="Unemployment"))
```

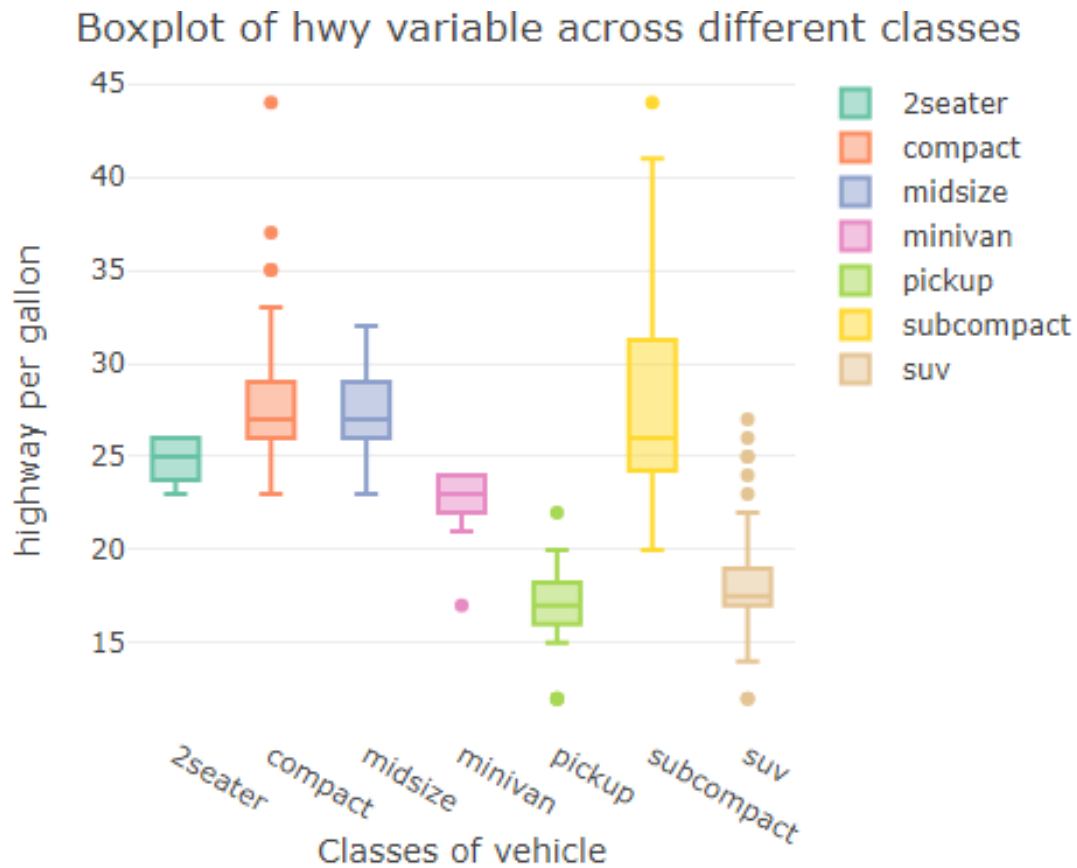


#Q4 ==

```
#Q4Produce a box plot using the mpg dataset to show the distribution of the  
hwy variable across different vehicle classes (class)  
#View(mpg)  
ggplot(data=mpg)+  
  geom_boxplot(mapping = aes(x=reorder(class,hwy,FUN=median),y=hwy))+  
  labs(title = "Boxplot of hwy variable across different classes",x="Classes  
of vehicle",y="highway per gallon")
```

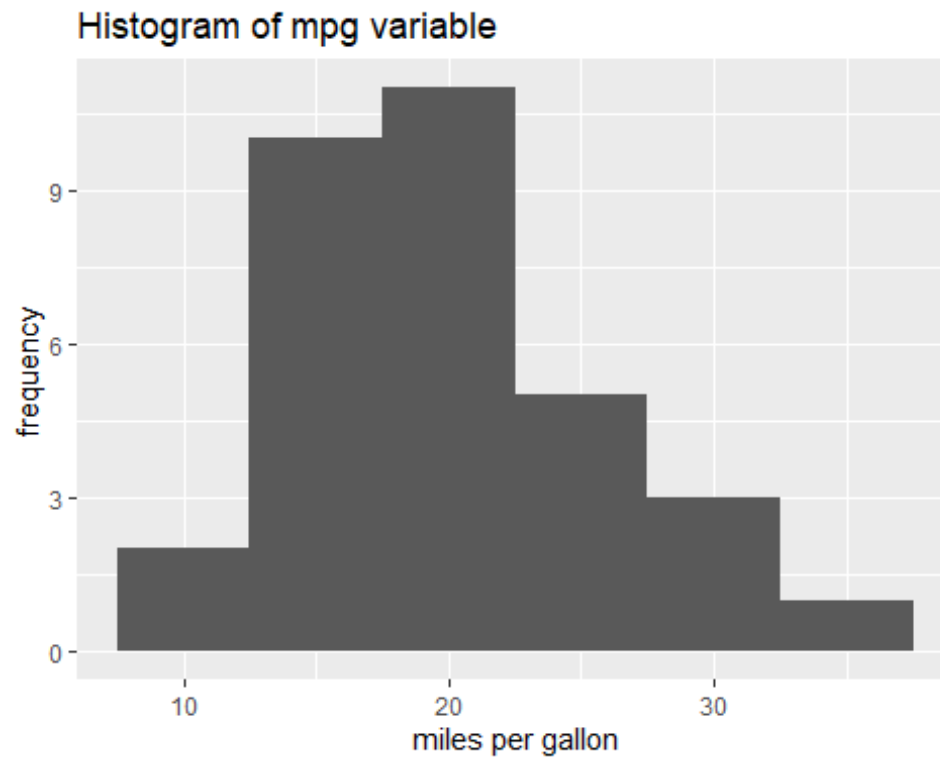



```
plot_ly(data=mpg,x=~class,y=~hwy,type="box",color=~class) %>%
layout(title="Boxplot of hwy variable across different
classes",xaxis=list(title="Classes of vehicle"),yaxis=list(title="highway per
gallon"))
```

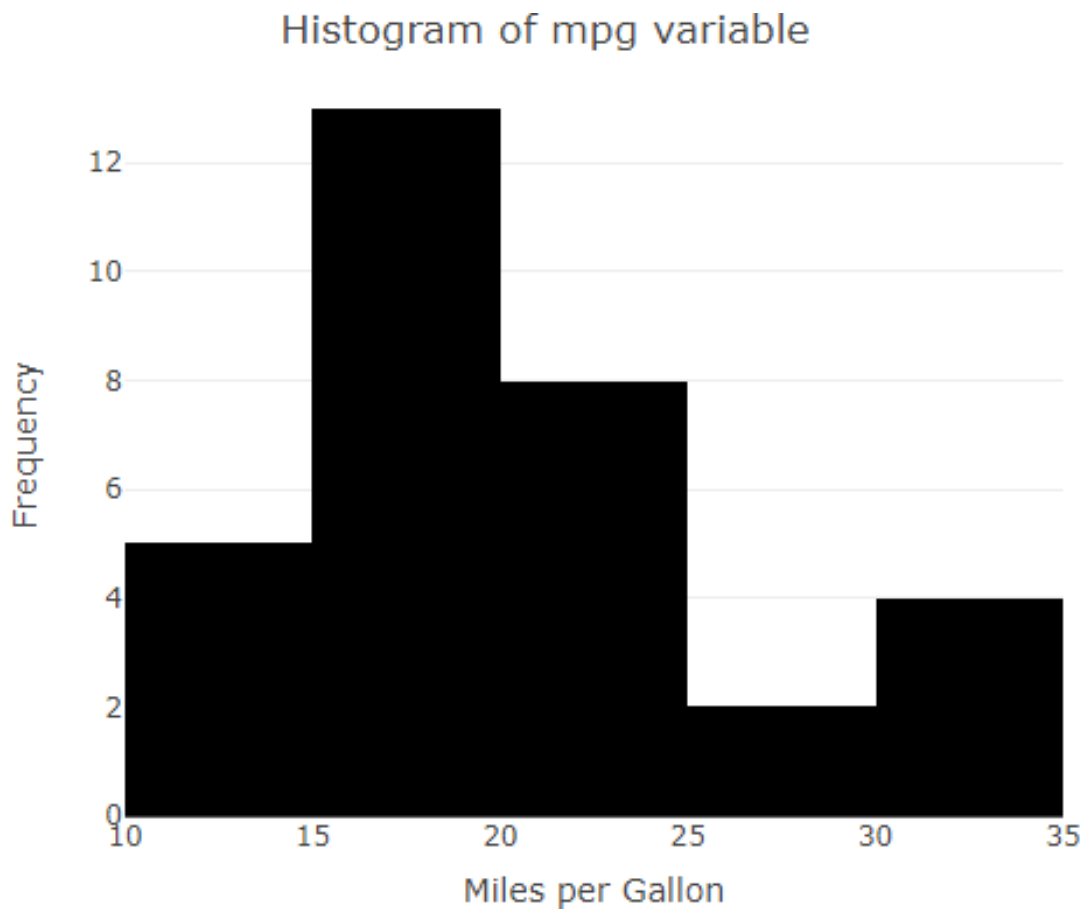


#Q5

```
#Q5 Use the mtcars dataset to create a histogram of mpg values.
#View(mtcars)
ggplot(data=mtcars)+
  geom_histogram(mapping = aes(mpg), binwidth = 5)+
  labs(title = "Histogram of mpg variable", x="miles per
gallon", y="frequency")
```

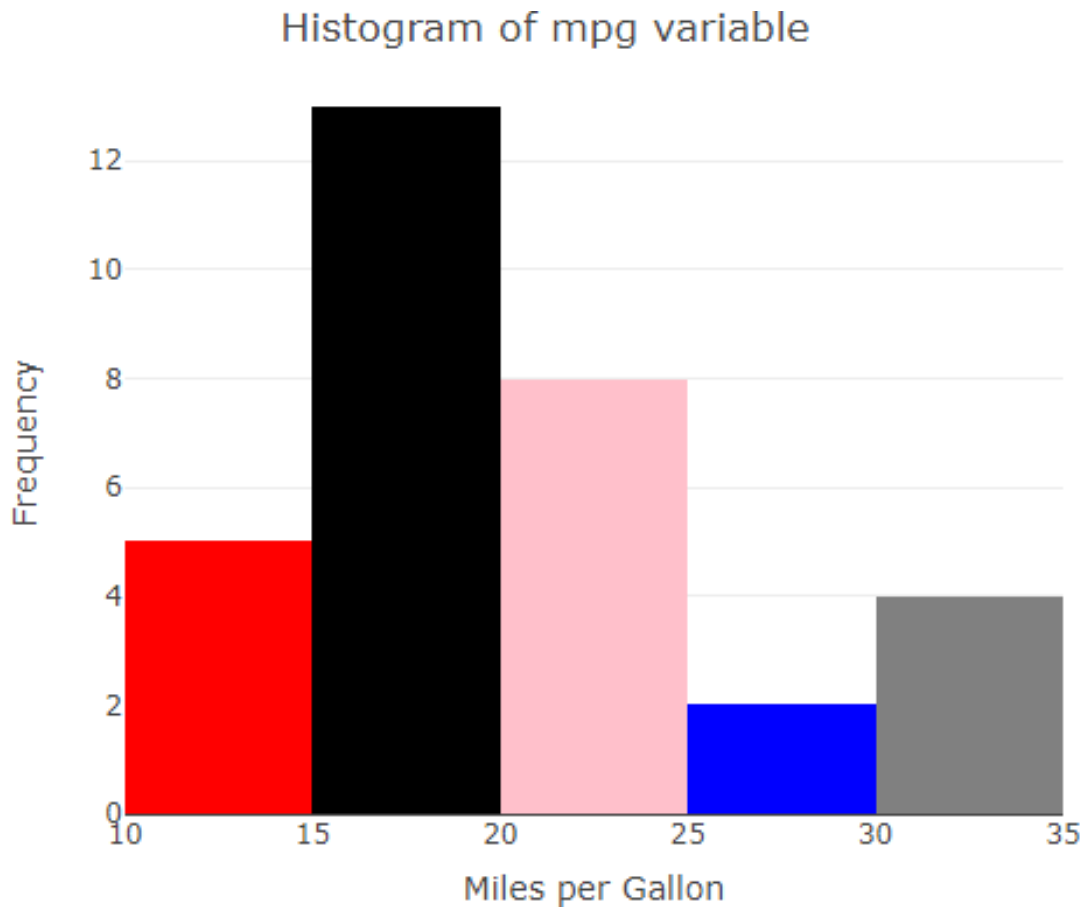


```
plot_ly(data=mtcars,x=~mpg,type = "histogram",color = I("black")) %>%  
layout(title="Histogram of mpg variable",xaxis=list(title="Miles per  
Gallon"),yaxis=list(title="Frequency"))
```



#OR

```
plot_ly(data=mtcars,x=~mpg,type = "histogram",marker
=list(color=c("RED","BLACK","PINK","BLUE","GREY"),size=5))%>%
layout(title="Histogram of mpg variable",xaxis=list(title="Miles per
Gallon"),yaxis=list(title="Frequency"))
```

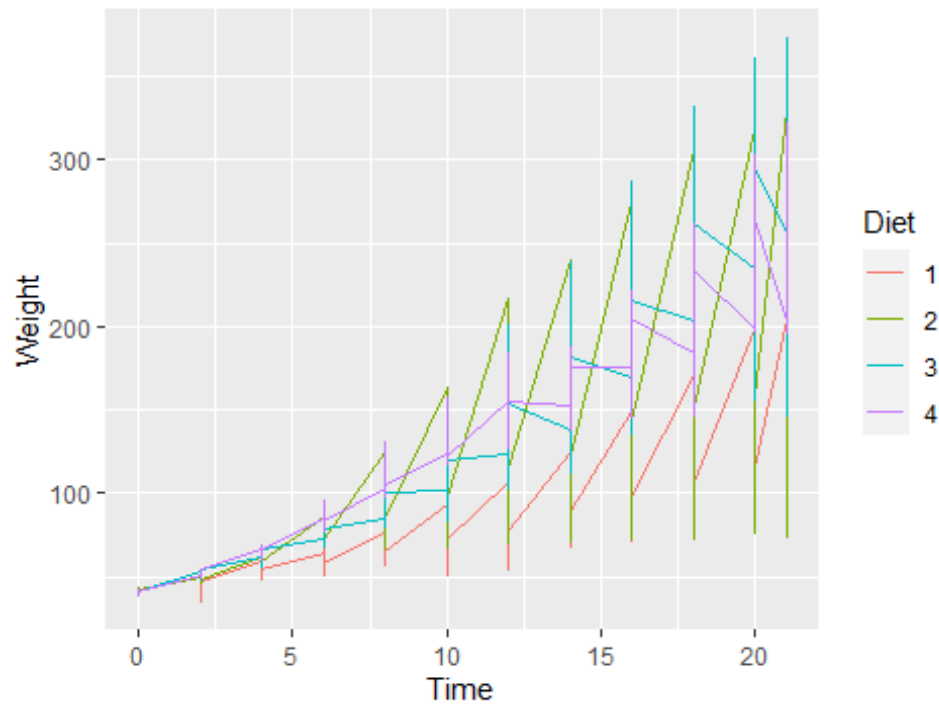


#Q6

#Q6 Using the ChickWeight dataset, draw a line chart to show the weight gain over time for a specific diet.

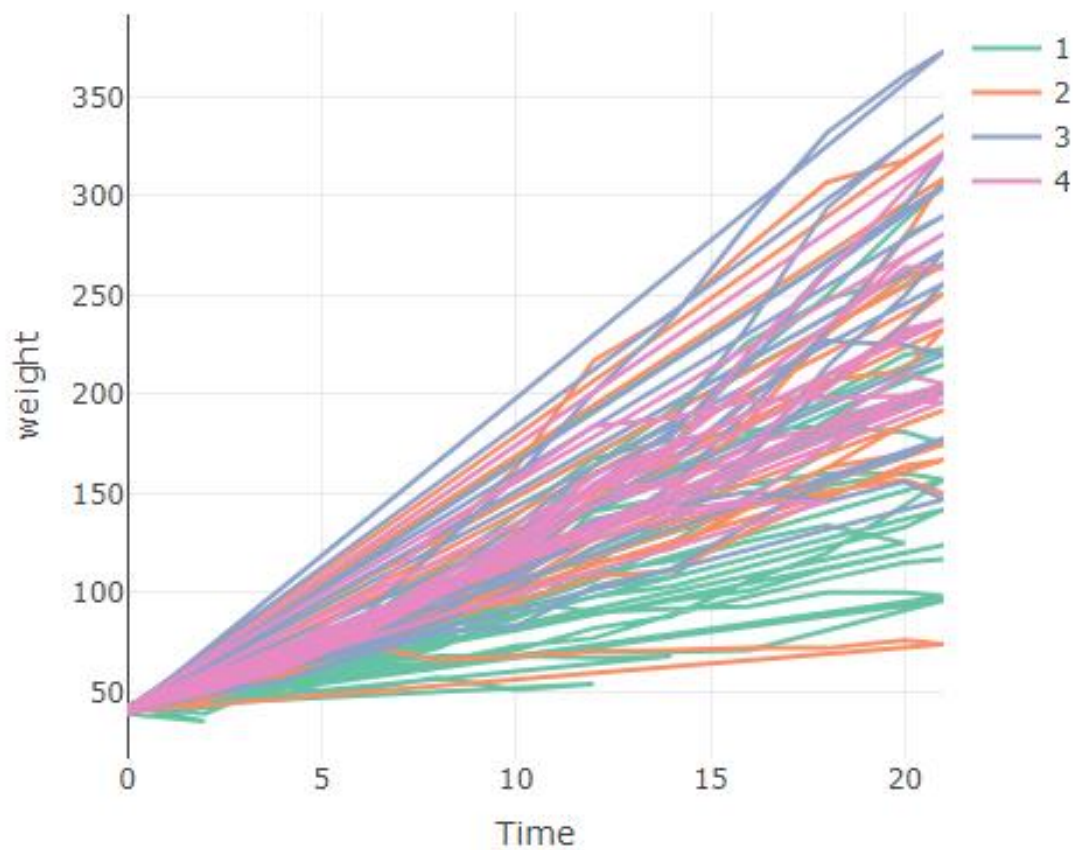
```
#View(ChickWeight)
ggplot(data=ChickWeight)+
  geom_line(mapping=aes(x=Time,y=weight,color=Diet))+
  labs(title = "line chart for weight gain over time for a specific
diet",x="Time",y="Weight")
```

line chart for weight gain over time for a specific diet



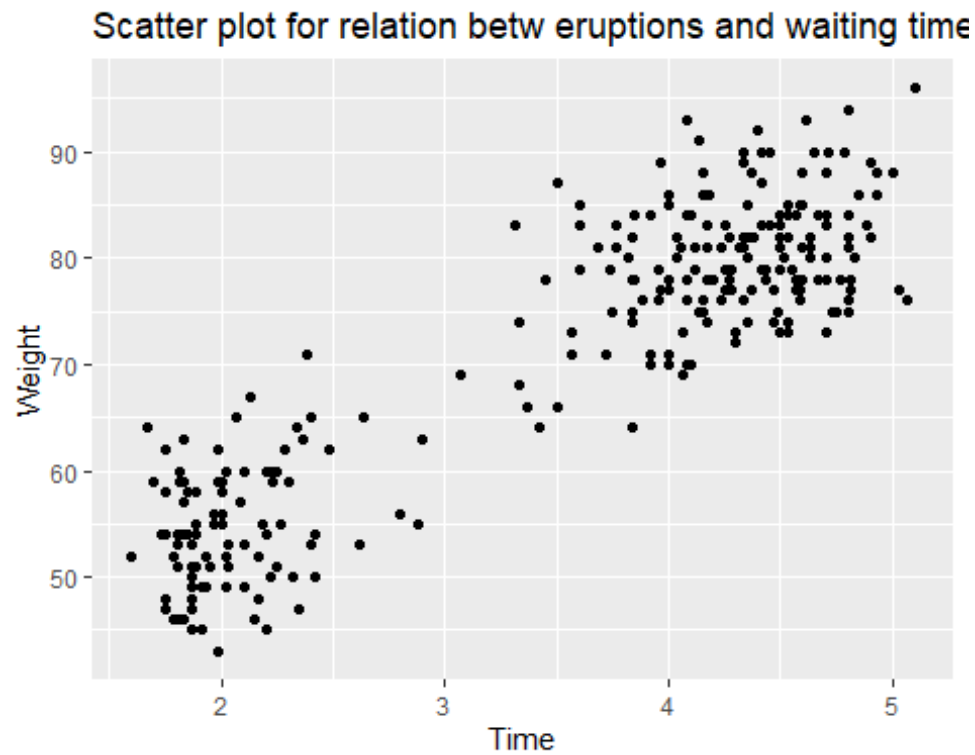
```
plot_ly(data=ChickWeight,x=~Time,y=~weight,type="scatter",mode="lines",color=~Diet) %>% layout(title="line chart for weight gain over time for a specific diet",xaxis=list(title="Time",yaxis=list(title="Weight")))
```

line chart for weight gain over time for a specific diet



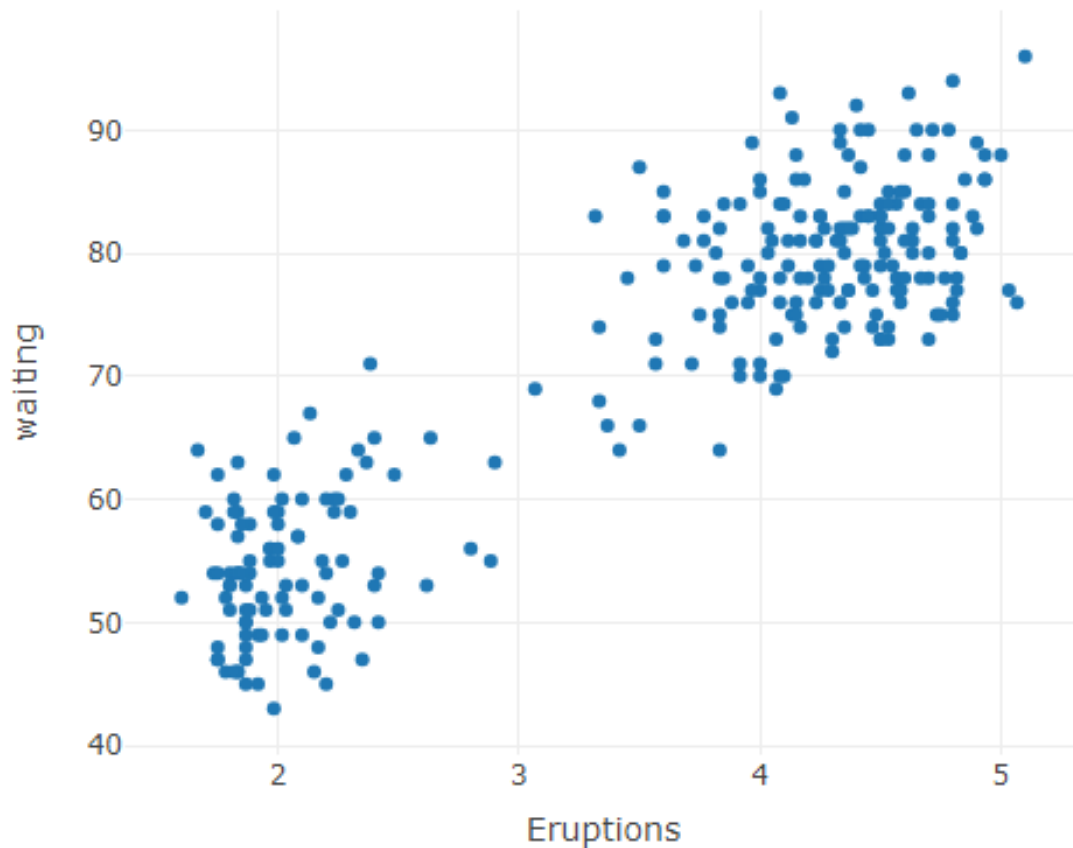
#Q7

```
#Q7 Create a scatter plot with the faithful dataset to visualize the
relationship between eruptions duration and waiting time.
#View(faithful)
ggplot(data=faithful)+
  geom_point(mapping = aes(x=eruptions,y=waiting))+
  labs(title = "Scatter plot for relation betw eruptions and waiting
time",x="Time",y="Weight")
```



```
plot_ly(data=faithful,x=~eruptions,y=~waiting,type="scatter",mode="markers")
%>% layout(title="Scatter plot for relation betw eruptions and waiting
time",xaxis=list(title="Eruptions"),yaxis=list(title="waiting"))
```


Scatter plot for relation betw eruptions and waiting time

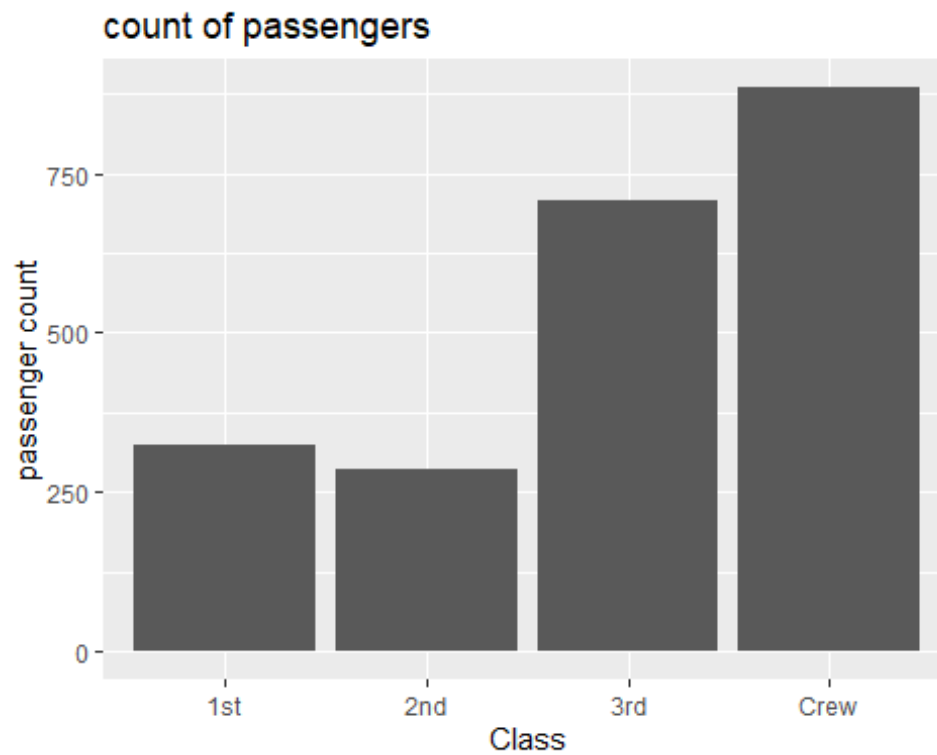


#Q8

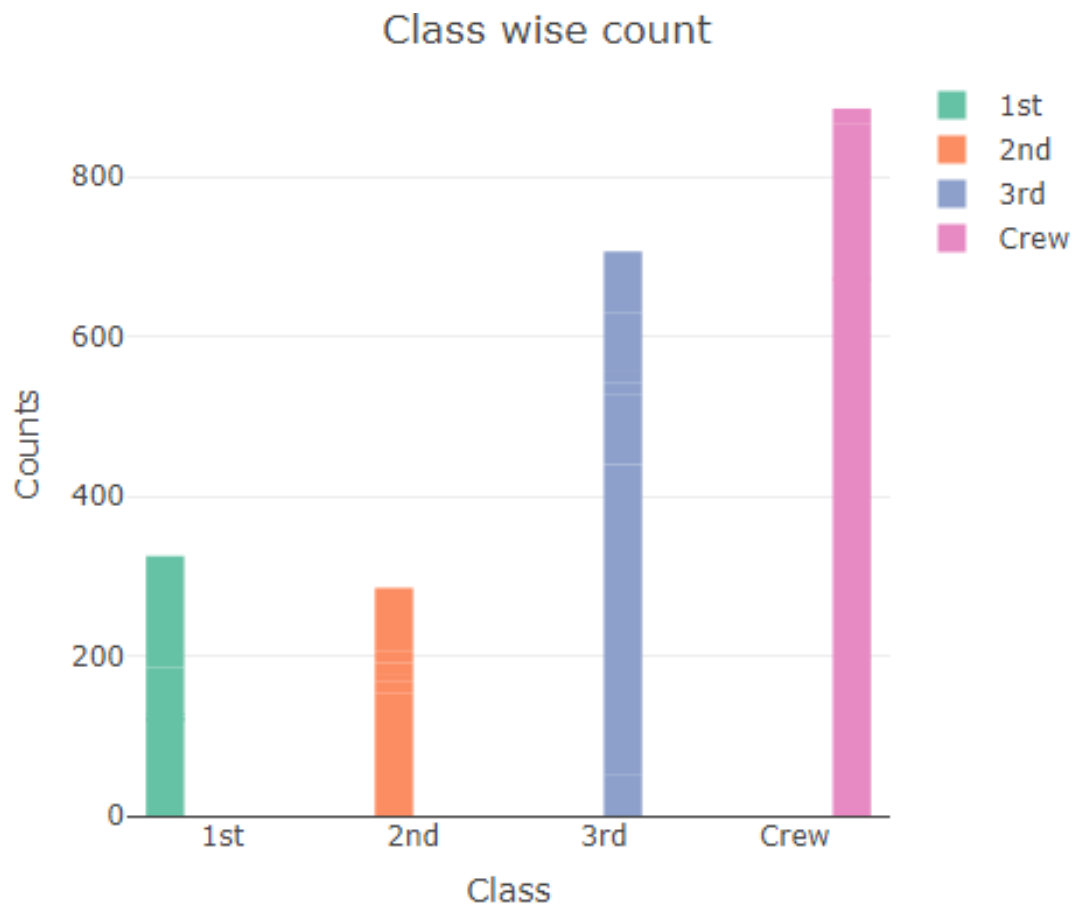
#Q8Generate a bar chart using the Titanic dataset to show the count of passengers by class (Class).

#View(Titanic)

```
ggplot(data=as.data.frame(Titanic))+  
  geom_col(mapping = aes(x=Class,y=Freq))+  
  labs(title = "count of passengers",x="Class",y="passenger count")
```

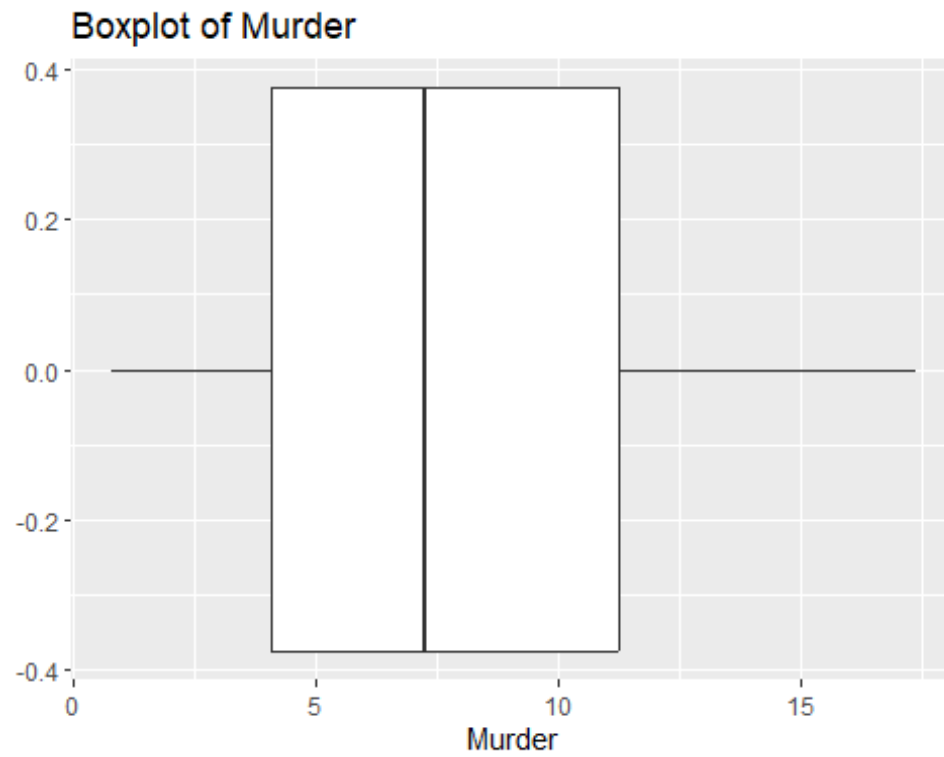


```
plot_ly(data=as.data.frame(Titanic),x=~Class,y=~Freq,type="bar",color=~Class)  
%>% layout(title="Class wise  
count",xaxis=list(title="Class"),yaxis=list(title="Counts"))
```

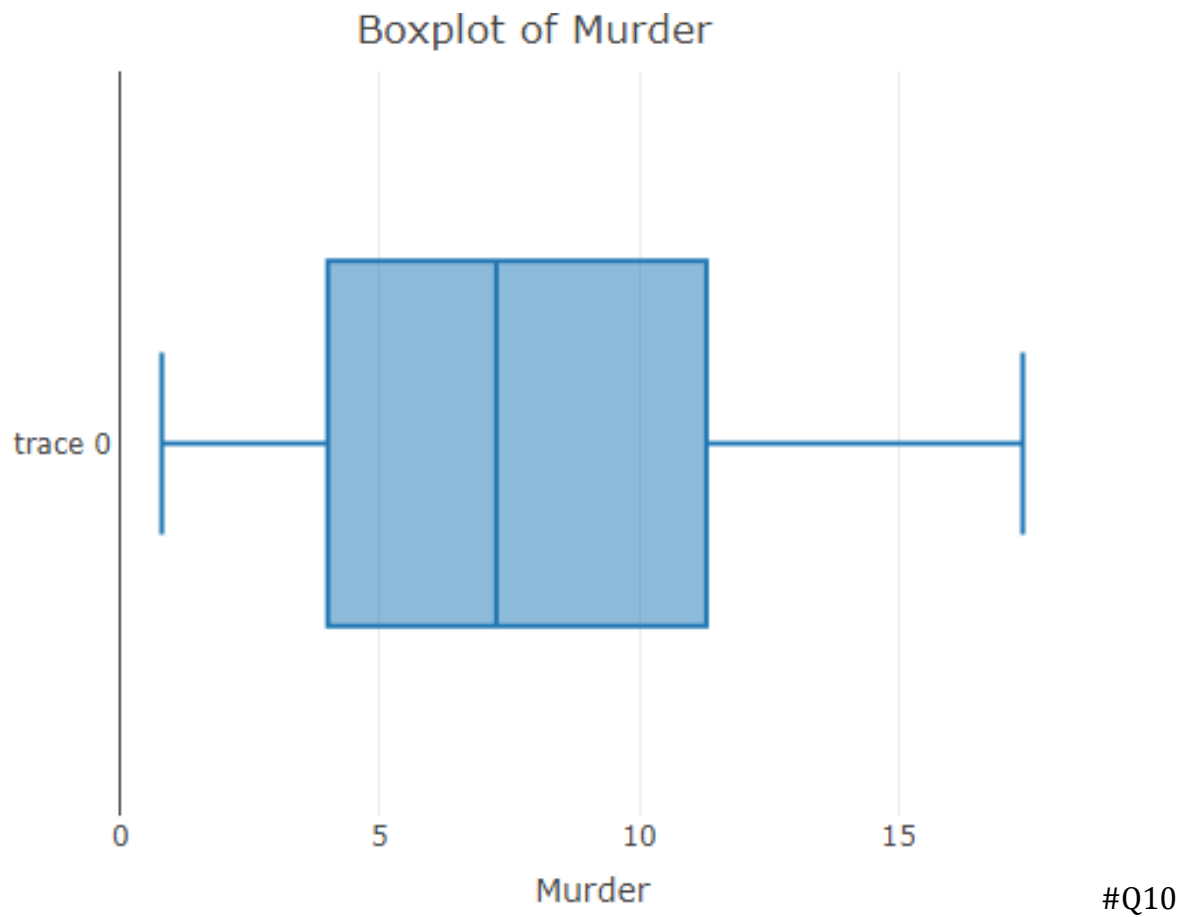


#Q9

```
#Q9 Use the USArrests dataset to create a box plot of the murder rate (Murder) .  
#View(USArrests)  
ggplot(data=USArrests)+  
  geom_boxplot(mapping=aes(x=Murder))+  
  labs(title = "Boxplot of Murder",x="Murder")
```

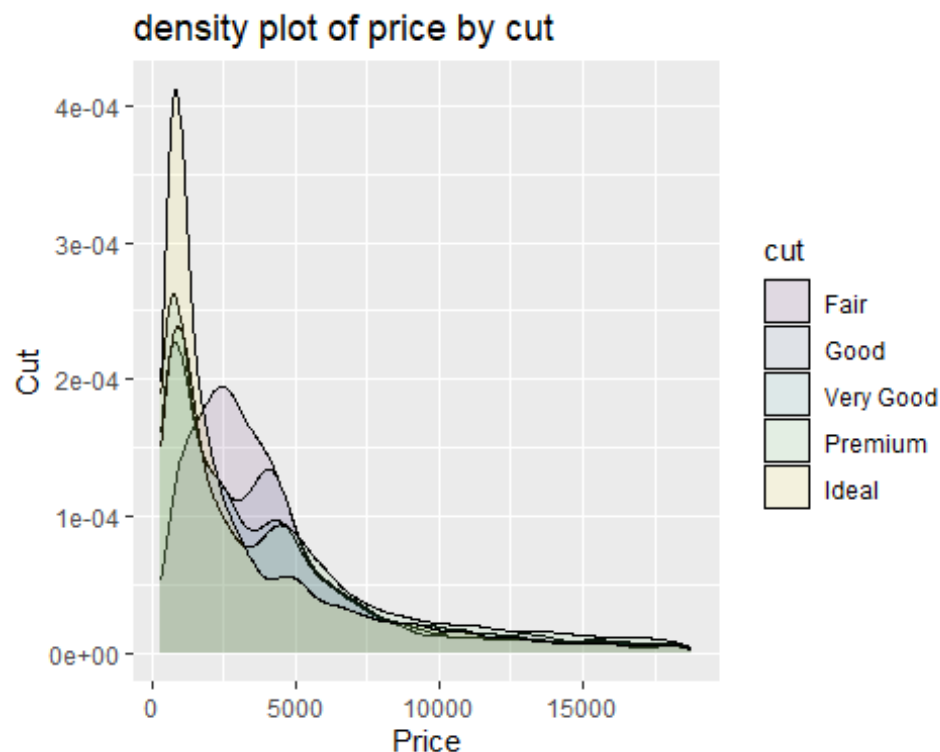


```
plot_ly(data=USArrests,x=~Murder,type="box")%>% layout(title="Boxplot of  
Murder",xaxis=list(title="Murder"))
```



#Q10 Create a density plot using the diamonds dataset to visualize the distribution of diamond prices (price) by diamond cut (cut).

```
#View(diamonds)
fit <- density(diamonds$price)
ggplot(data=diamonds)+
  geom_density(mapping = aes(x=price,fill=cut),alpha=0.1)+
  labs(title="density plot of price by cut",x="Price",y="Cut")
```



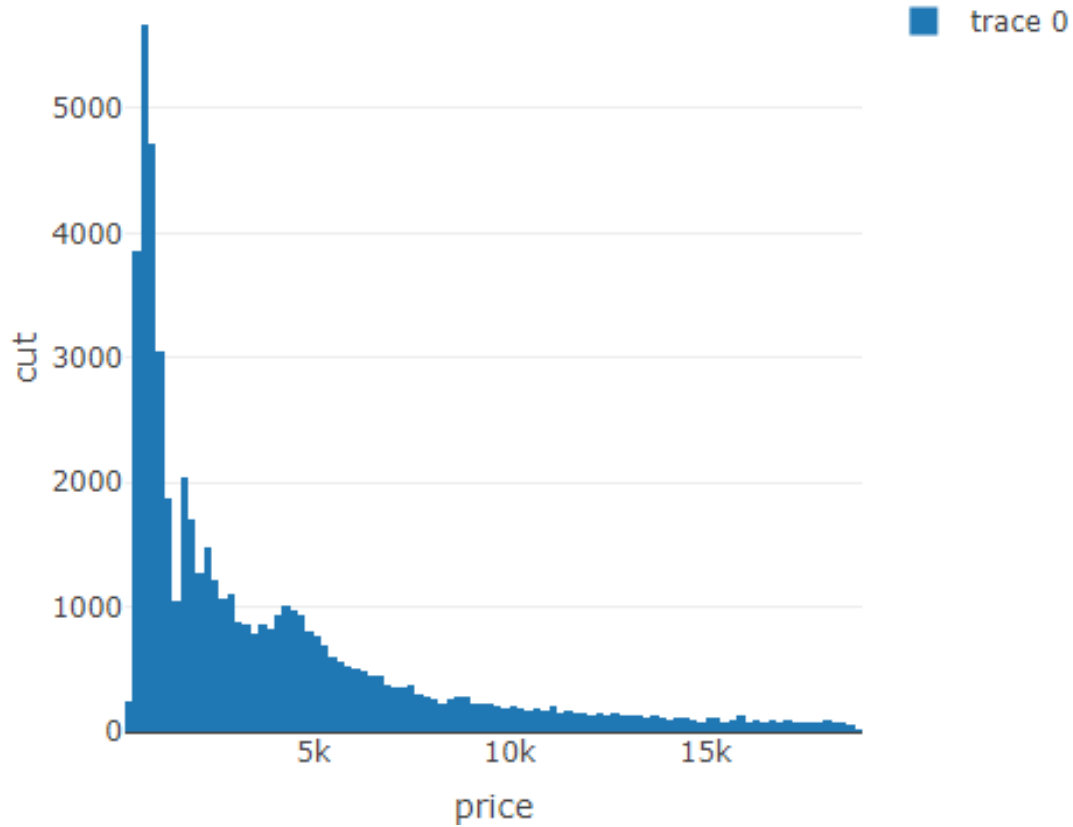
```
plot_ly(data=diamonds,x=~price,y=~cut,type = "histogram") %>%
add_trace(x=fit$price,y=fit$cut,mode="lines",fill="tozeroy")%>% layout(yaxis2
= list(overlying = "y", side = "right"))
```

```
## Warning: Can't display both discrete & non-discrete data on same axis
```

```
## Warning: 'histogram' objects don't have these attributes: 'mode', 'fill'
```

```
## Valid attributes include:
```

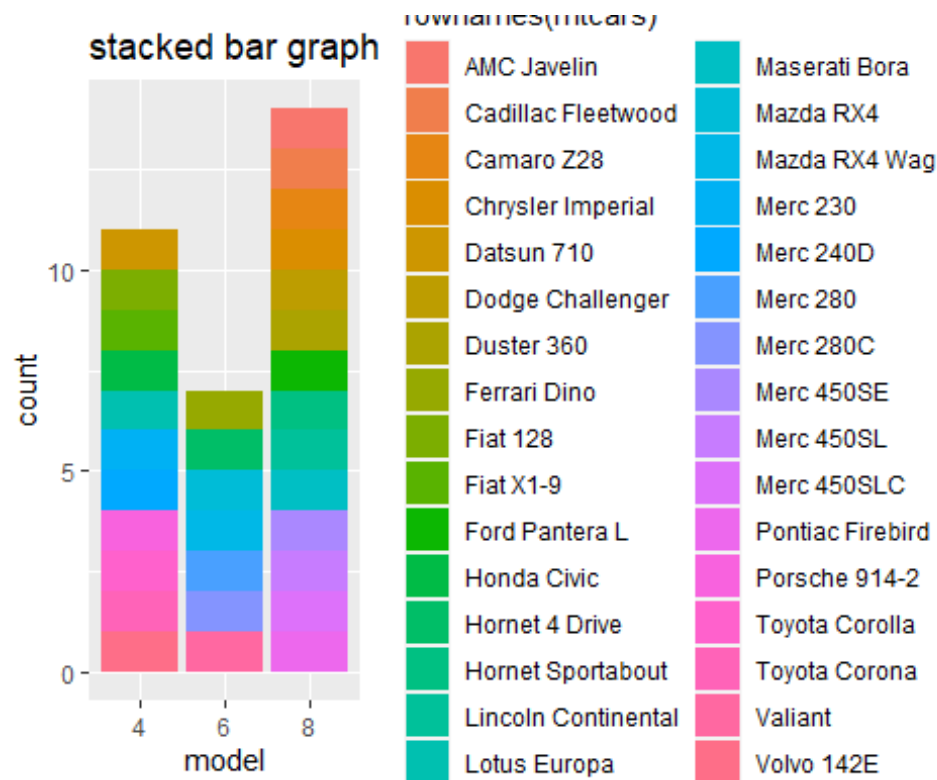
```
## '_deprecated', 'alignmentgroup', 'autobinx', 'autobiny', 'bingroup',
'cliponaxis', 'constrainttext', 'cumulative', 'customdata', 'customdatasrc',
'error_x', 'error_y', 'histfunc', 'histnorm', 'hoverinfo', 'hoverinfosrc',
'hoverlabel', 'hovertemplate', 'hovertemplatesrc', 'hovertext',
'hovertextsrc', 'ids', 'idsrc', 'insidetextanchor', 'insidetextfont',
'legendgroup', 'legendgrouptitle', 'legendrank', 'marker', 'meta', 'metasrc',
'name', 'nbinsx', 'nbinsy', 'offsetgroup', 'opacity', 'orientation',
'outsidetextfont', 'selected', 'selectedpoints', 'showlegend', 'stream',
'text', 'textangle', 'textfont', 'textposition', 'textsrc', 'texttemplate',
'transforms', 'type', 'uid', 'uirevision', 'unselected', 'visible', 'x',
'xaxis', 'xbins', 'xcalendar', 'xhoverformat', 'xsrc', 'y', 'yaxis', 'ybins',
'ycalendar', 'yhoverformat', 'ysrc', 'key', 'set', 'frame', 'transforms',
'_isNestedKey', '_isSimpleKey', '_isGraticule', '_bbox'
```



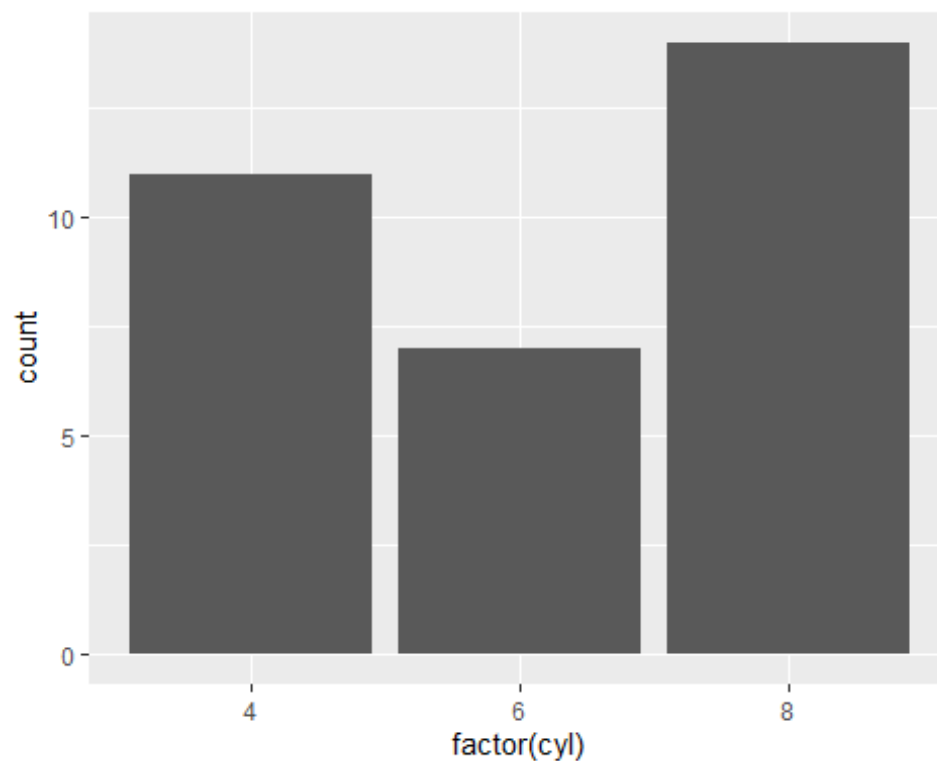
```
#plot_ly(x = x, type = "histogram", name = "Histogram") %>% add_trace(x =
fit$x, y = fit$y, mode = "lines", fill = "tozero", yaxis = "y2", name =
"Density")
```

#11

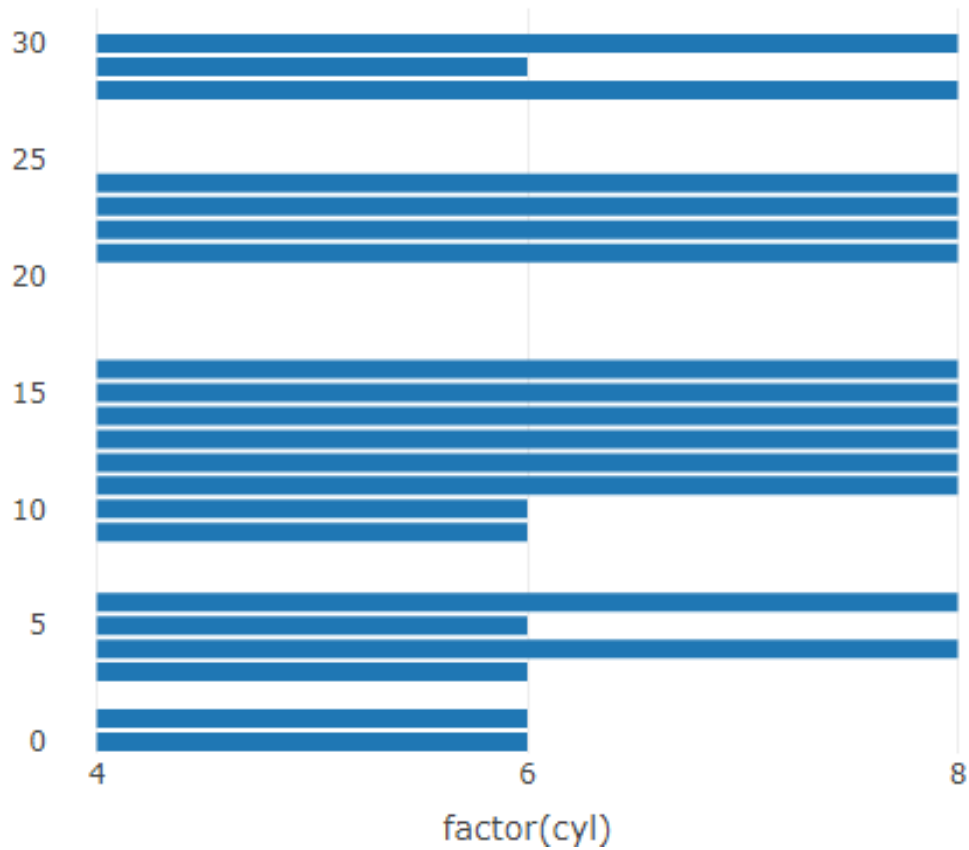
```
View(mtcars)
ggplot(data=mtcars)+
  geom_bar(mapping=aes(x=factor(cyl),fill=rownames(mtcars)))+
  labs(title = "stacked bar graph",x="model")
```



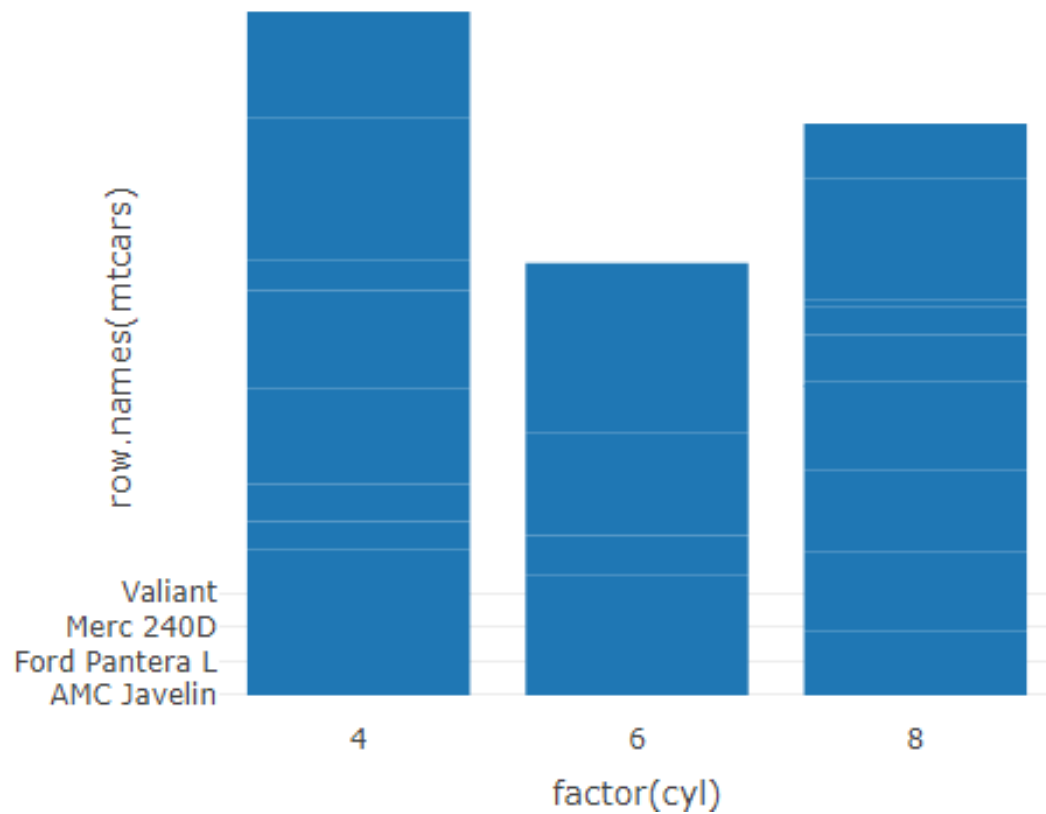
#or
`ggplot(data=mtcars)+
 geom_bar(mapping=aes(x=factor(cyl)))`




```
plot_ly(data=mtcars,x=~factor(cyl),type="bar")
```



```
plot_ly(data=mtcars,x = ~factor(cyl),y=~row.names(mtcars), type = 'bar')
```

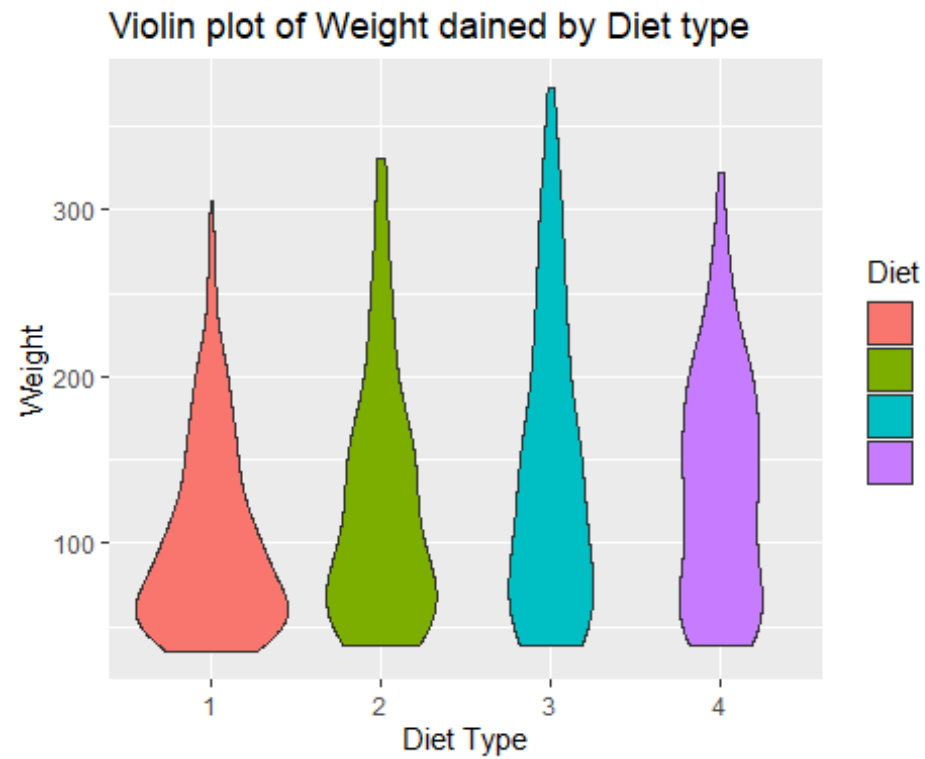


#12

```
View(volcano)
fig <- plot_ly(as.data.frame(volcano), x = ~V1, y = ~V2, z=~V3, colors =
c('#BF382A', '#0C4B8E'))
```

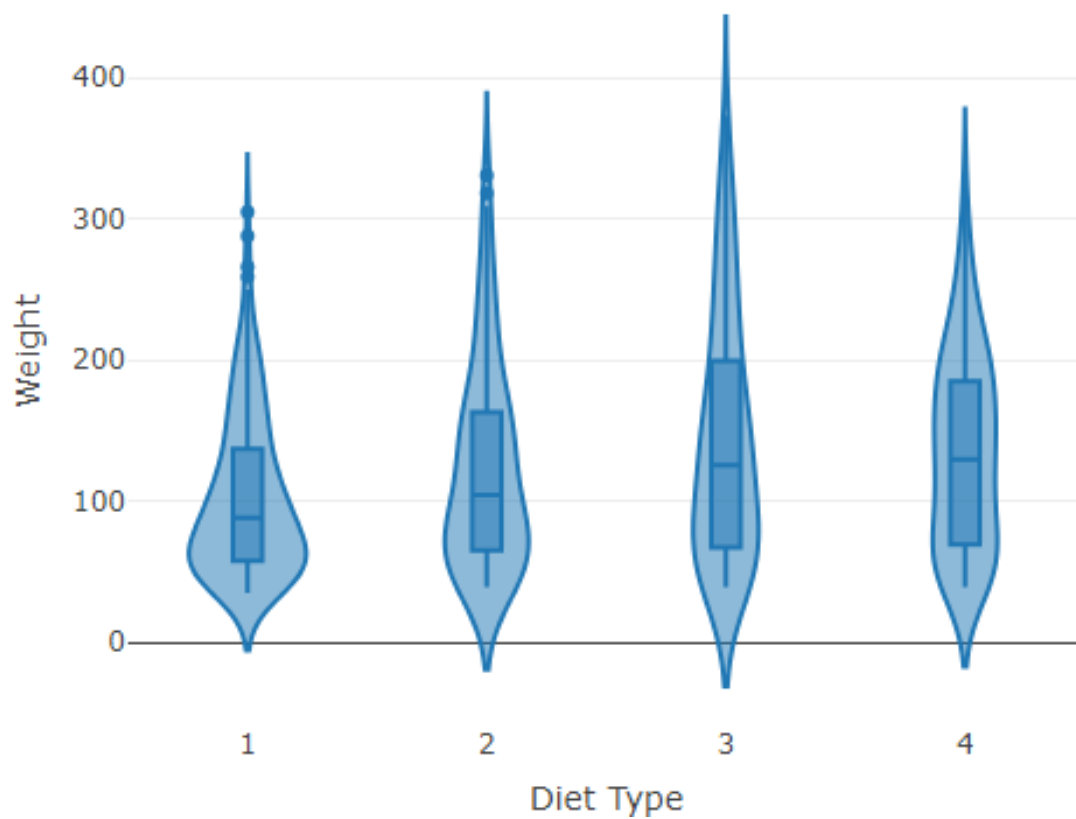
#13

```
ggplot(data=ChickWeight)+
  geom_violin(mapping=aes(x=Diet,y=weight,fill=Diet))+
  labs(title = "Violin plot of Weight gained by Diet type",x="Diet
Type",y="Weight")
```



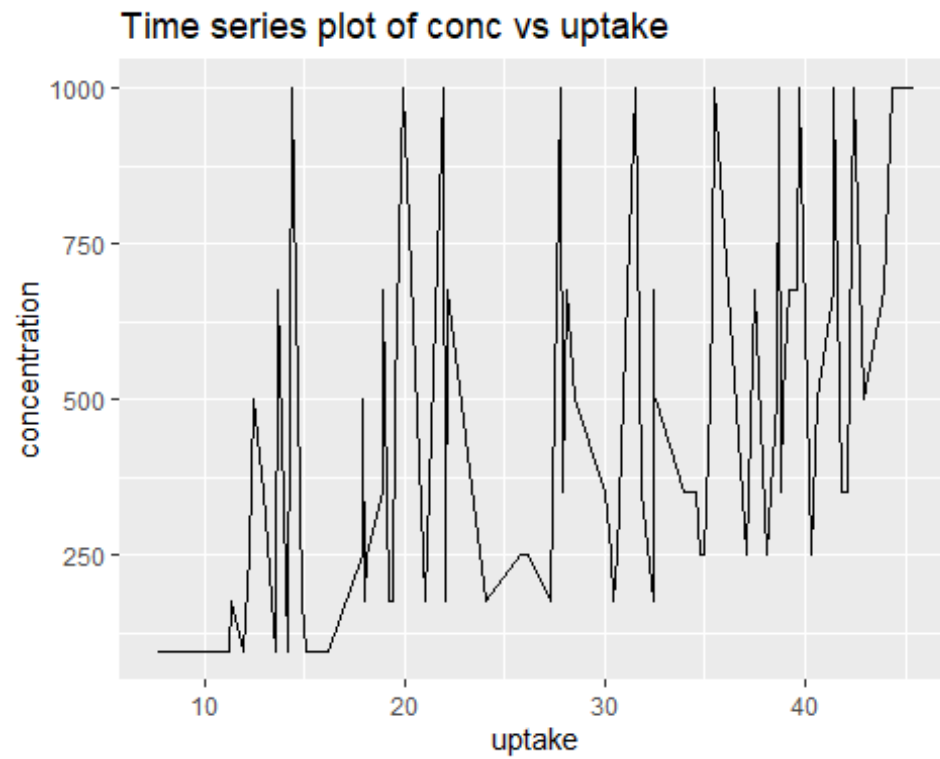
```
plot_ly(data=ChickWeight,x=~Diet,y=~weight,type='violin',box=list(visible=TRUE)) %>%  
  layout(title="Violin plot of Weight dained by Diet",xaxis=list(title="Diet Type"),yaxis=list(title="Weight"))
```

Violin plot of Weight dained by Diet



#Q14

```
View(C02)
ggplot(data=C02)+
  geom_line(mapping=aes(x=uptake,y=conc))+
  labs(title="Time series plot of conc vs
uptake",x="uptake",y="concentration")
```



```
plot_ly(data=C02,x=~uptake,y=~conc,type="scatter",color=~Plant,mode='lines')
```

