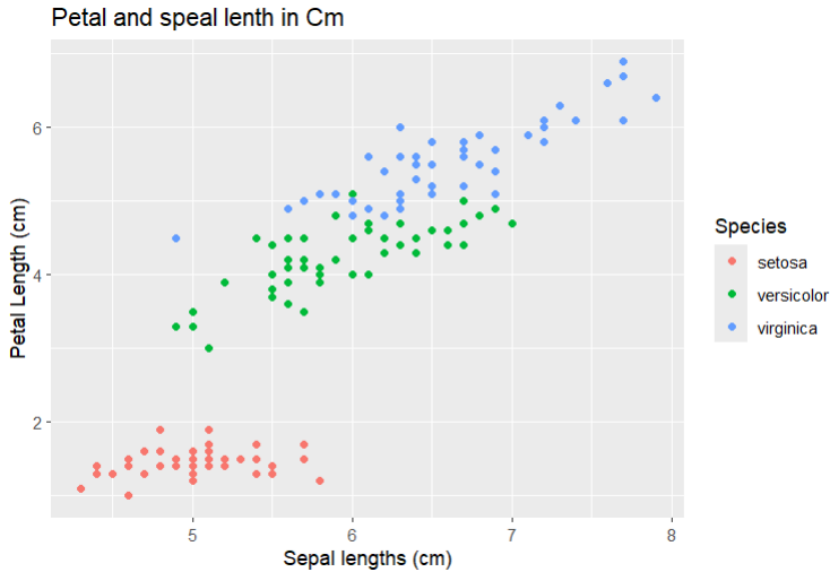
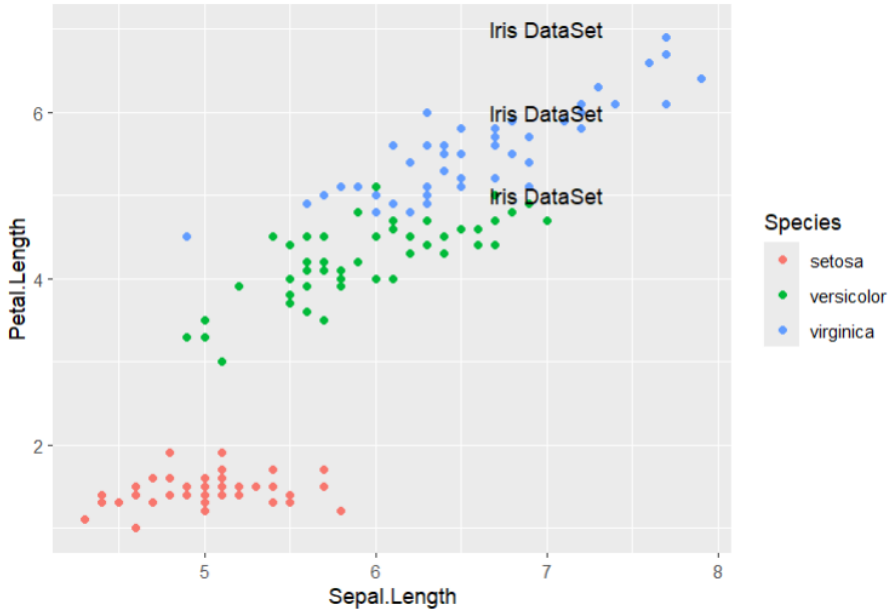
Practical 3: Data Visualization using ggplot2

```
data(iris)
head(iris)
libraray('ggplot2')
# Default Plot in R
IrisPlot <- ggplot(iris,aes(Sepal.Length,Petal.Length,colour=Species))+
geom_point() +
  labs(y="Petal Length (cm)",x="Sepal lengths (cm)")+
  ggtitle("Petal and speal lenth in Cm")
IrisPlot
```

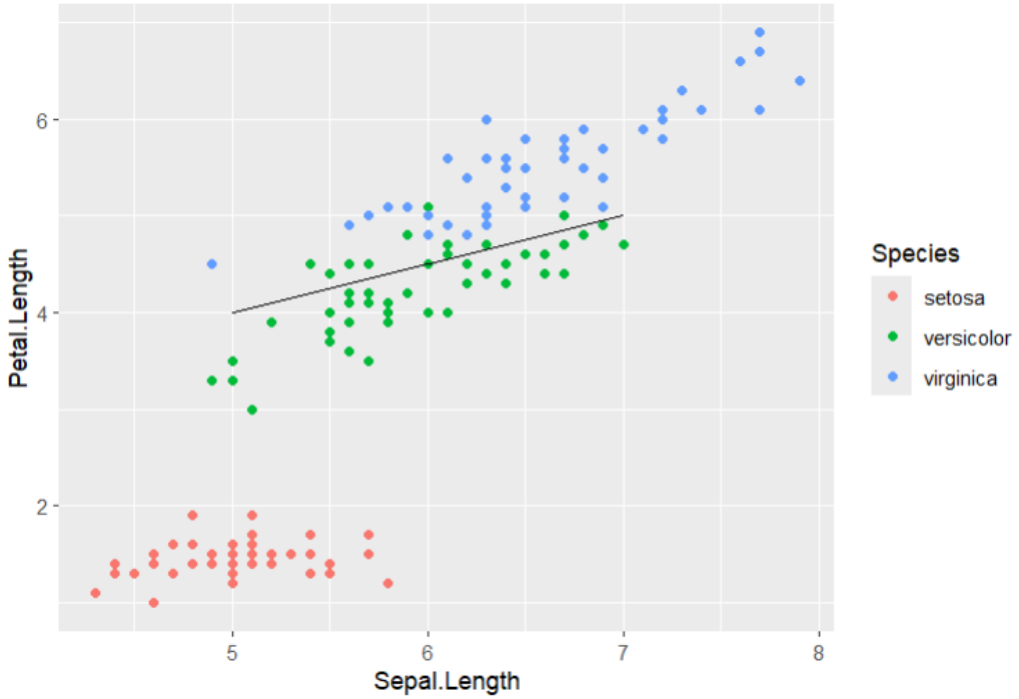



```
p<- ggplot(iris , aes(Sepal.Length,Petal.Length , colour=Species))+
geom_point()
p
# Adding repeat text
p+ annotate("text",x=7,y=5:7,label="Iris DataSet")
```

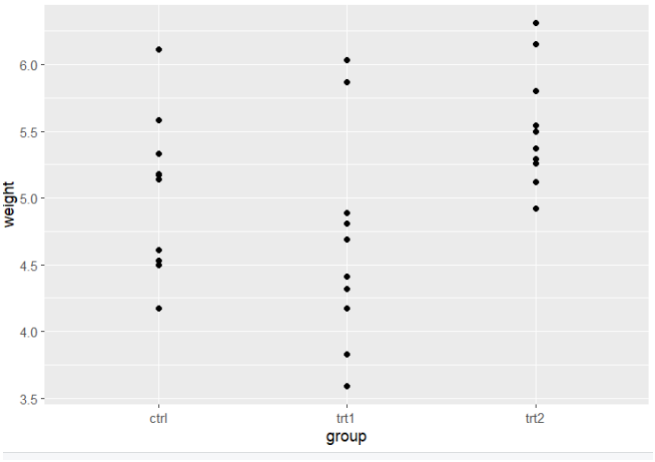


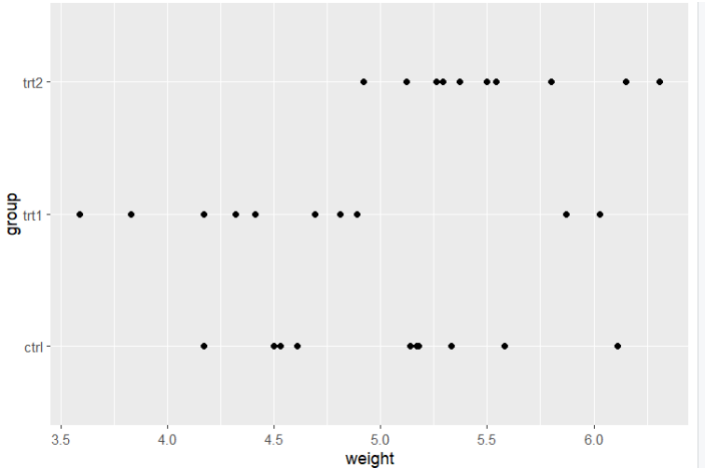
```
# highlighting an AREA
p+ annotate("rect",xmin=5,xmax=7,ymin=4,ymax=6,alpha=.6)
```


```
#Segment
p+ annotate("segment",x=5,xend=7,y=4,yend=5,color="black")
```

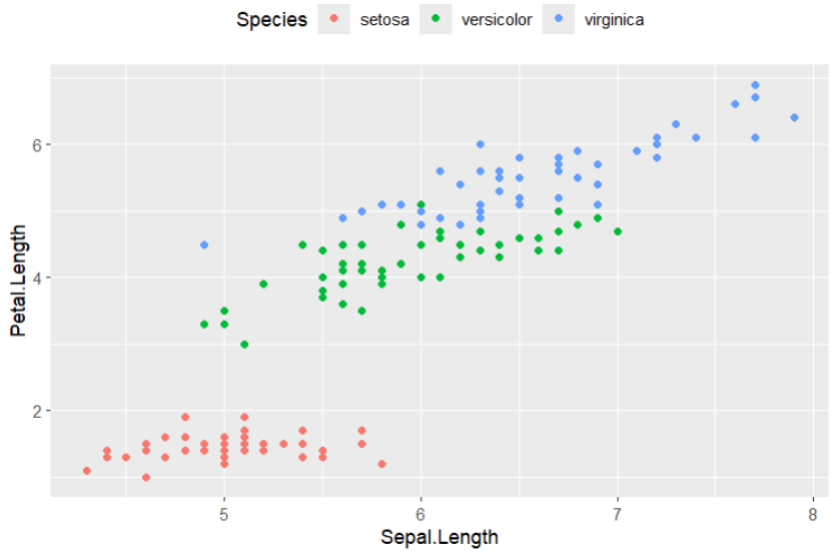



```
# Adding attributes with axes
# Plant Growth Data set
PG <- PlantGrowth
bp<- ggplot(PG,aes(x=group,y=weight)) + geom_point()
bp

bp<- ggplot(PG,aes(x=weight,y=group)) + geom_point()
bp
```

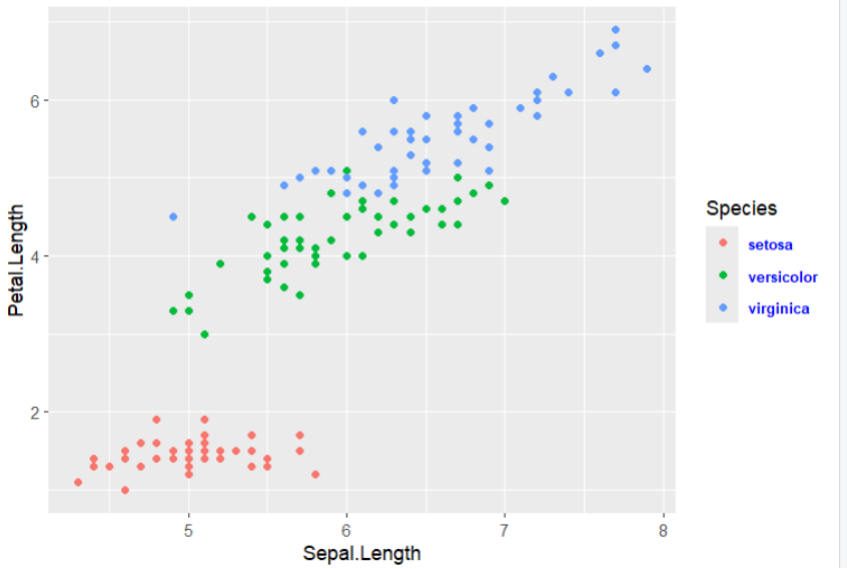




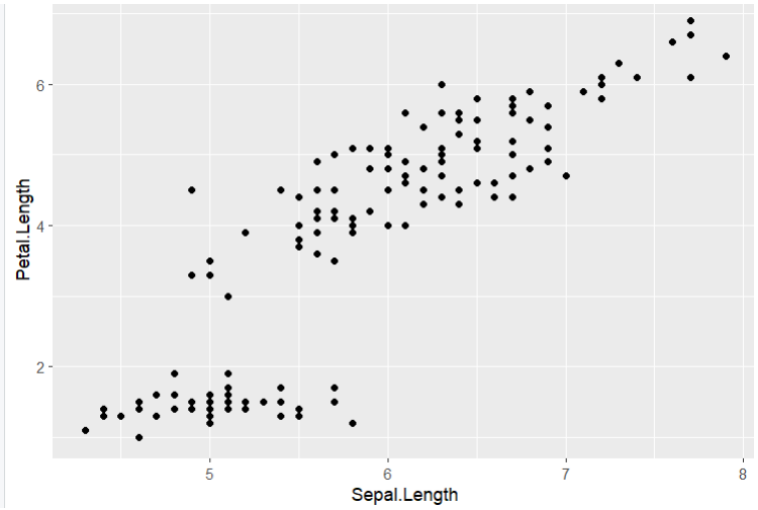

```
#  Working with Legends
# change the legend Position

p+ theme(legend.position = "top")
p+ theme(legend.position = "bottom")
```



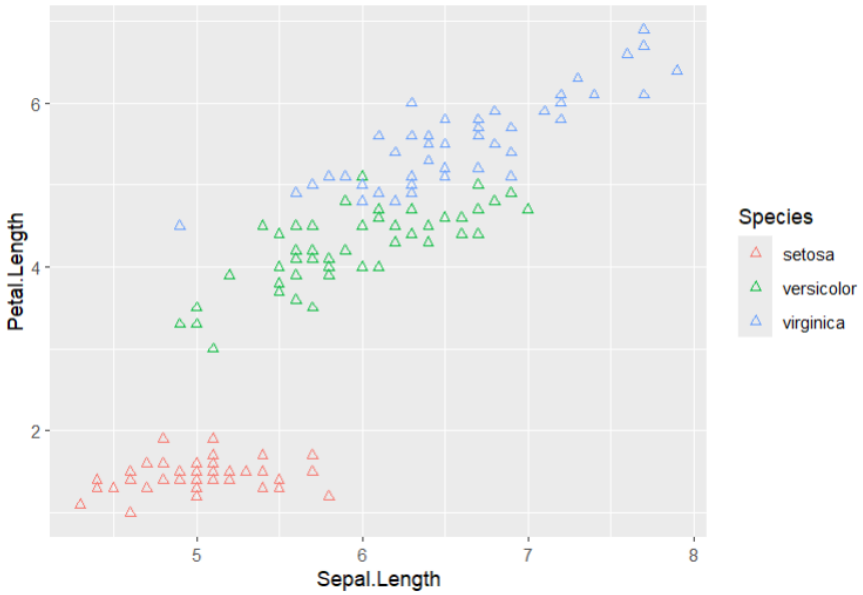

```
# legend Title
p+ theme(legend.title=element_text(colour="red",size=10,face="bold"))
# Legend label
p+ theme(legend.text=element_text(colour="blue",size=8,face="bold"))
```



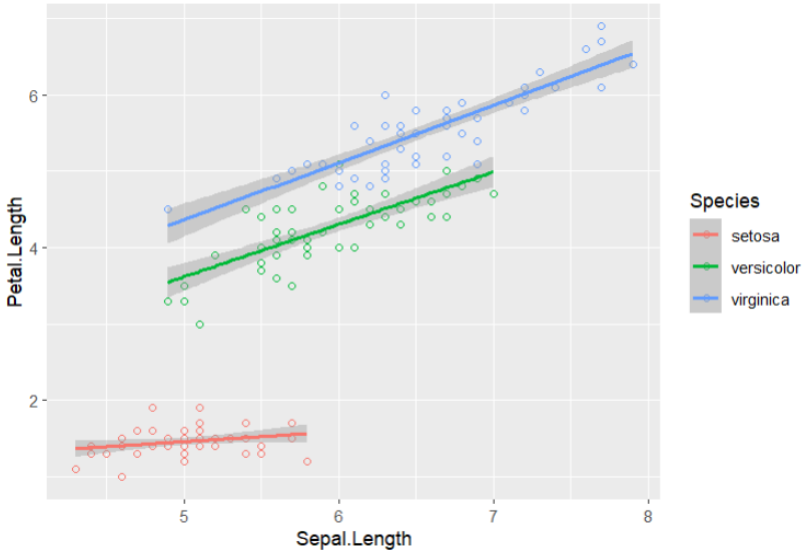
```
# Scatter Plots and Jitter Plots
scatter = ggplot(iris,aes(Sepal.Length,Petal.Length))+ geom_point()
scatter
```



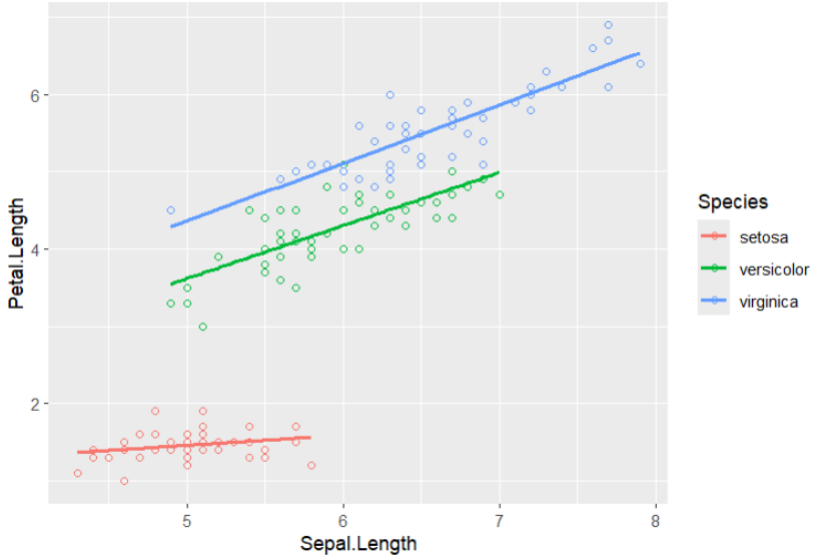

```
# now using color to change the attributes of plot
ggplot(iris,aes(Sepal.Length,Petal.Length,color=Species))+geom_point(shape
=2)
```



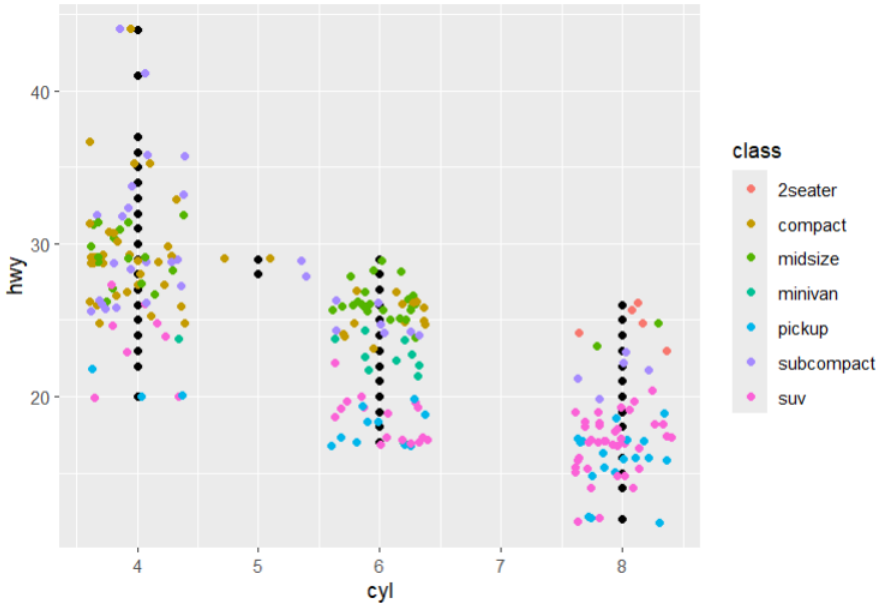

```
# Relationship between the variables
ggplot(iris,aes(Sepal.Length,Petal.Length,colour=Species))+
  geom_point(shape=1)+geom_smooth(method = lm)
```



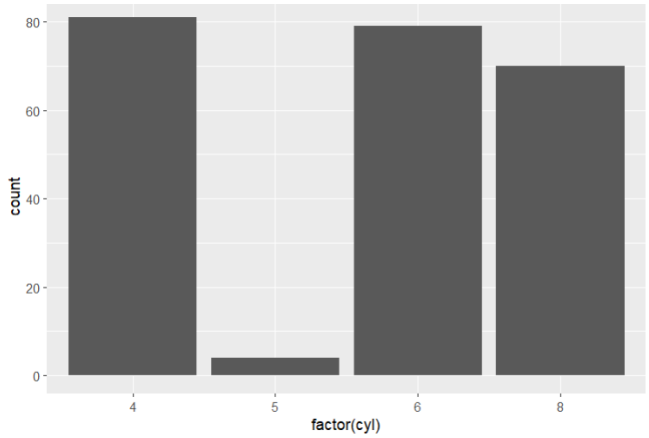
```
#Regression Line With no shaded region
ggplot(iris,aes(Sepal.Length,Petal.Length,colour=Species))+
  geom_point(shape=1)+geom_smooth(method=lm,se=FALSE)
```



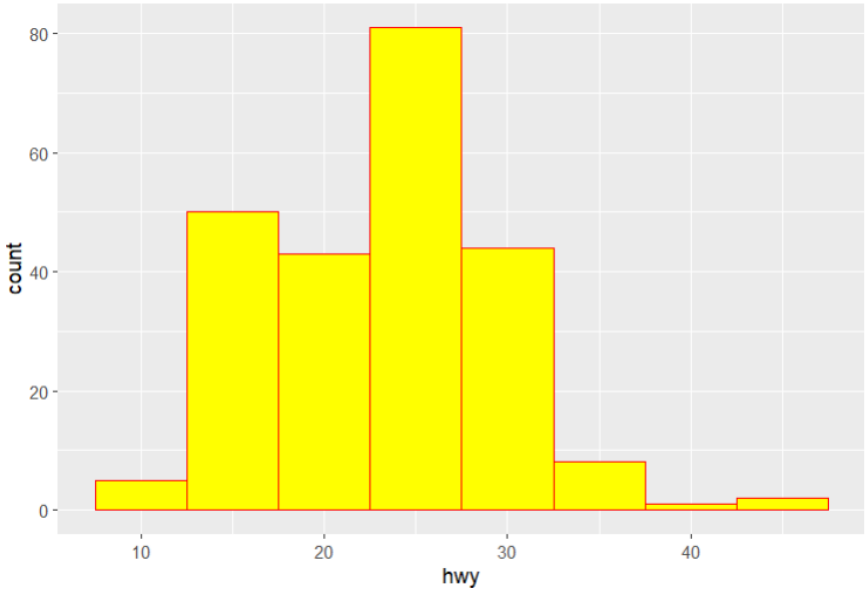

```
# Jitter Plot
ggplot(mpg,aes(cyl,hwy))+ geom_point()+geom_jitter(aes(colour=class))
```



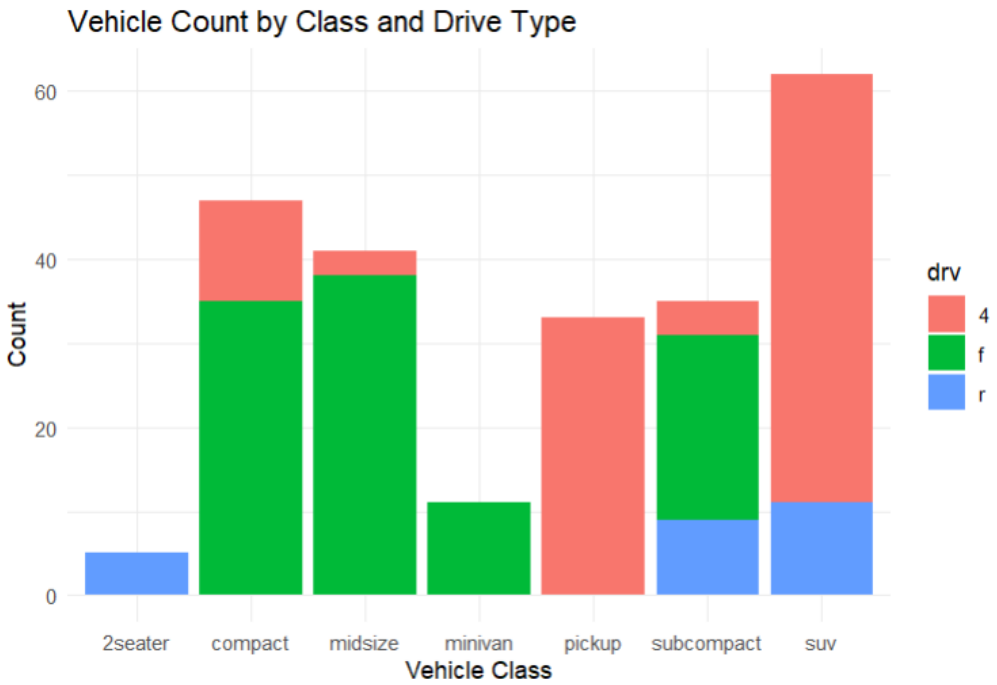

```
# Bar Plots and Histograms
data(mpg)
head(mpg)
hist <- ggplot(mpg,aes(x=factor(cyl)))+ geom_bar(stat="count")
hist
```



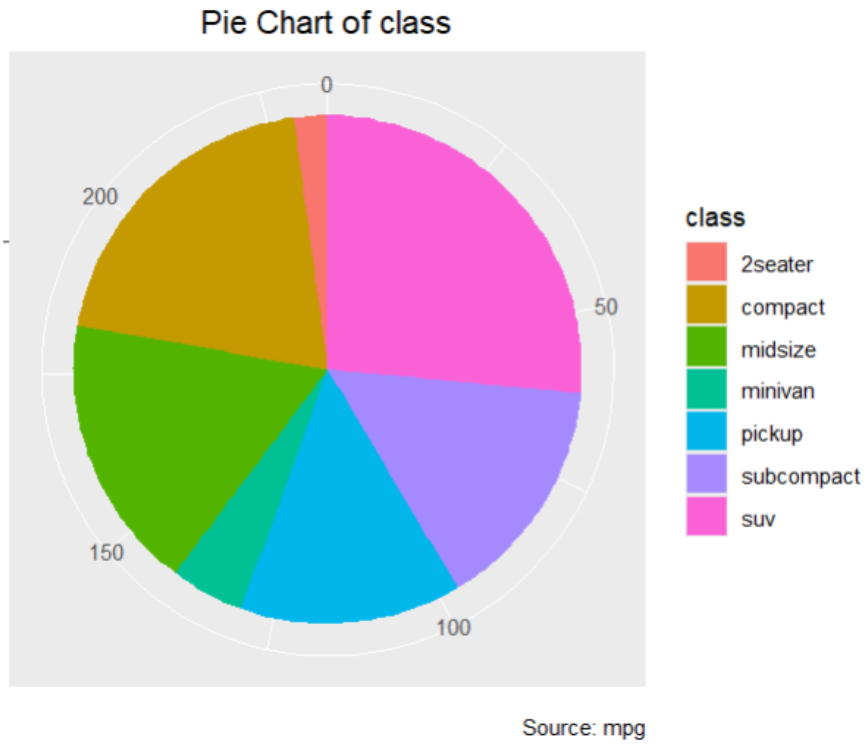
```
# Histogram count plot
ggplot(data=mpg,aes(x=hwy))+
geom_histogram(col="red",fill="yellow",aplha=.5,binwidth = 5)
```



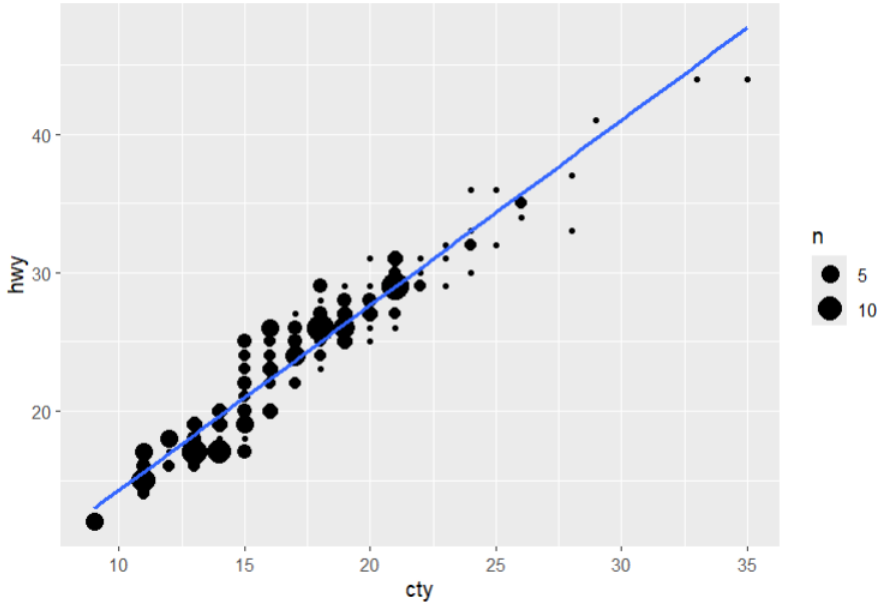

```
# Stacked Bar chart
p <- ggplot(mpg, aes(class, fill = drv)) +
  geom_bar() +
  labs(title = "Vehicle Count by Class and Drive Type",
       x = "Vehicle Class",
       y = "Count") +
  theme_minimal()
p
```



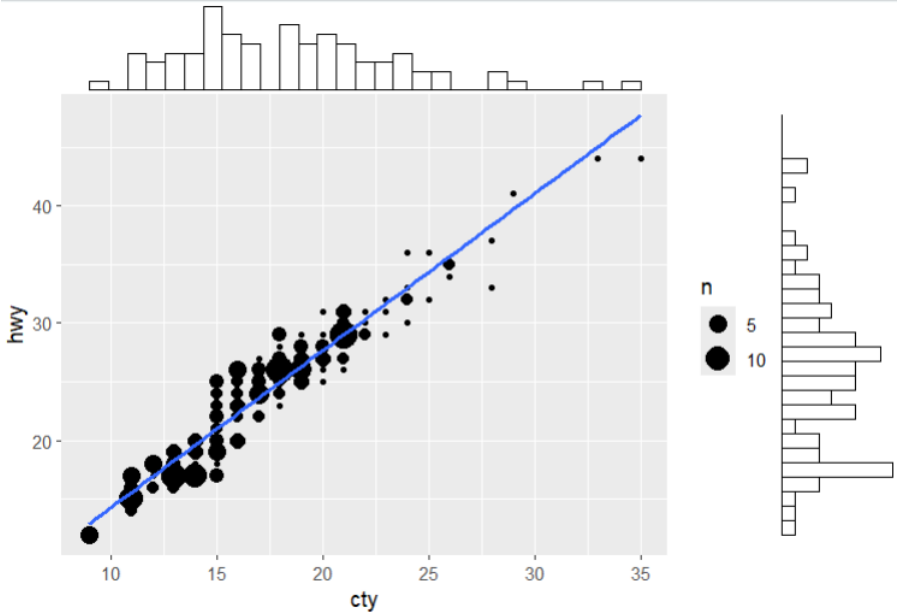
```
# Pie Charts
df<- as.data.frame(table(mpg$class))
colnames(df)<- c("class","freq")
pie <- ggplot(df, aes(x = "", y = freq, fill = factor(class))) +
  geom_bar(width = 1, stat = "identity") +
  theme(axis.line = element_blank(),
        plot.title = element_text(hjust = 0.5)) +
  labs(fill = "class",
       x = NULL,
       y = NULL,
       title = "Pie Chart of class",
       caption = "Source: mpg")
# Creating Co-ordinates
pie+ coord_polar(theta="y",start=0)
```



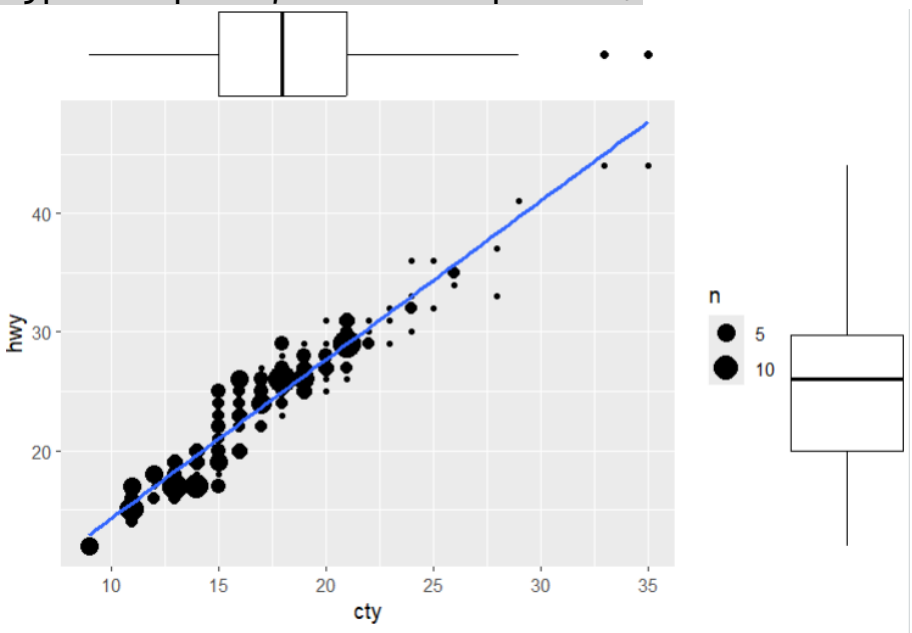

```
# Marginal plot
install.packages("ggExtra")
library("ggExtra")
data(mpg)
head(mpg)

# Plots
g<- ggplot(mpg,aes(cty,hwy)) + geom_count() +
  geom_smooth(method="lm",se=F)
g
```



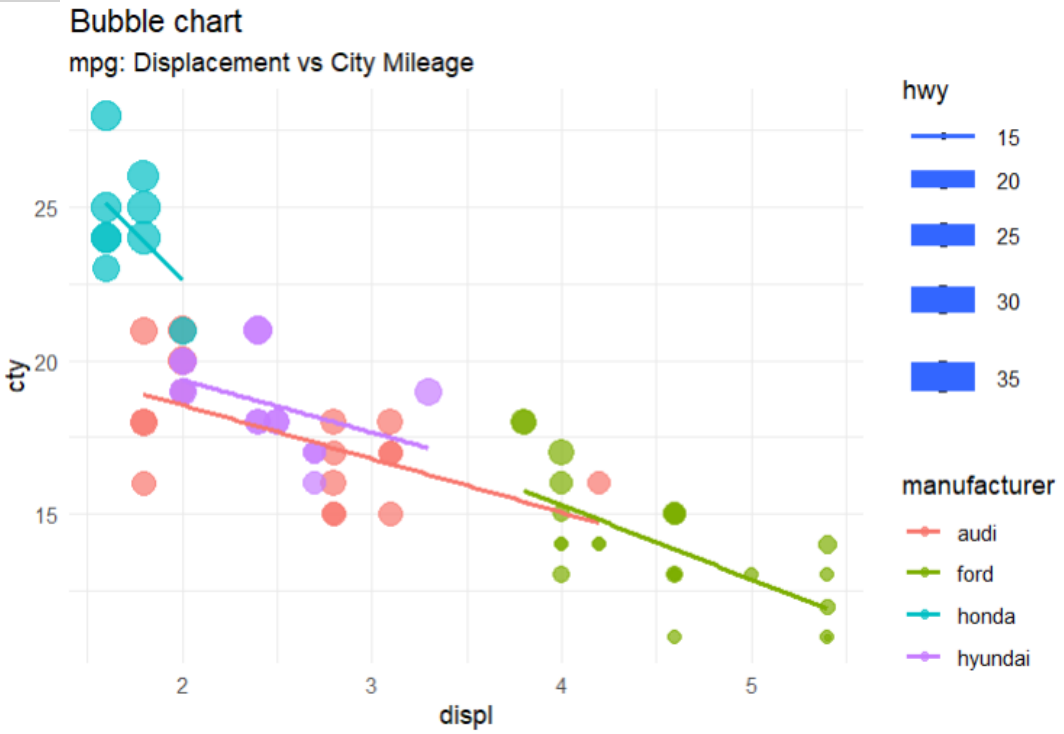

```
# Relation Between Variables
ggMarginal(g,type="histogram",fill="transparent")
```



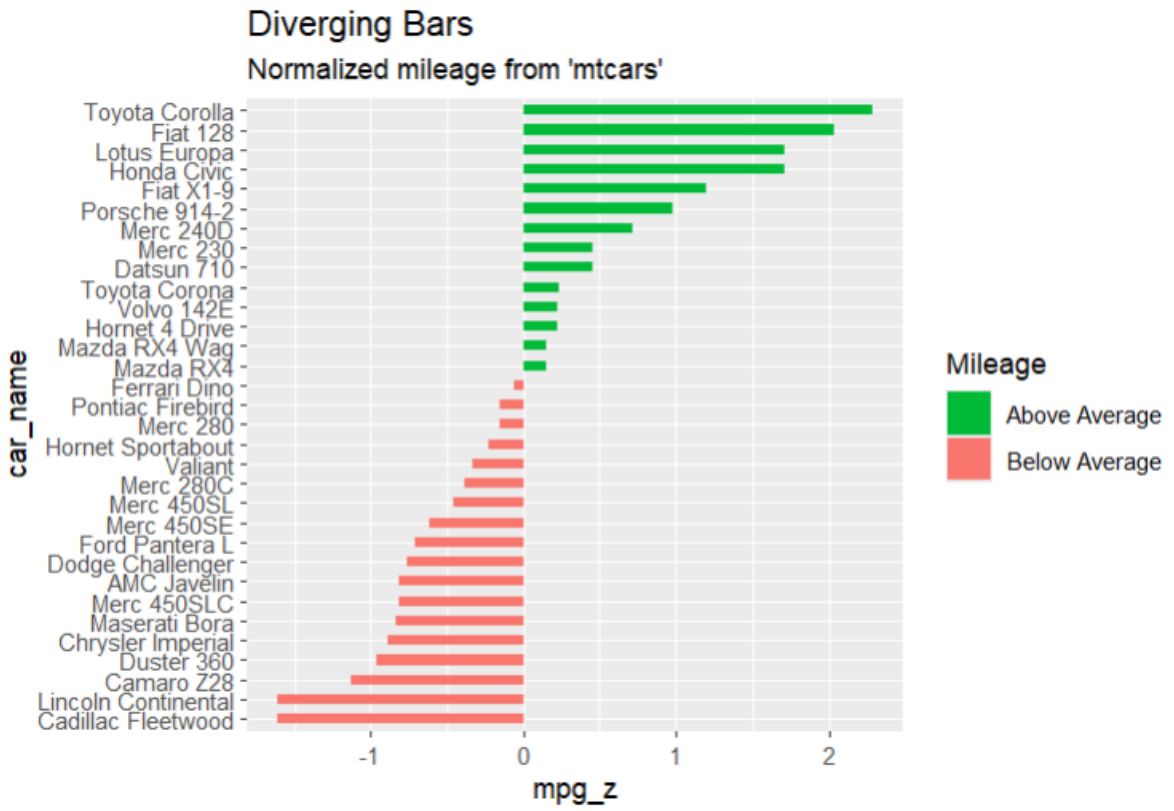

```
ggMarginal(g,type="boxplot",fill="transparent")
```



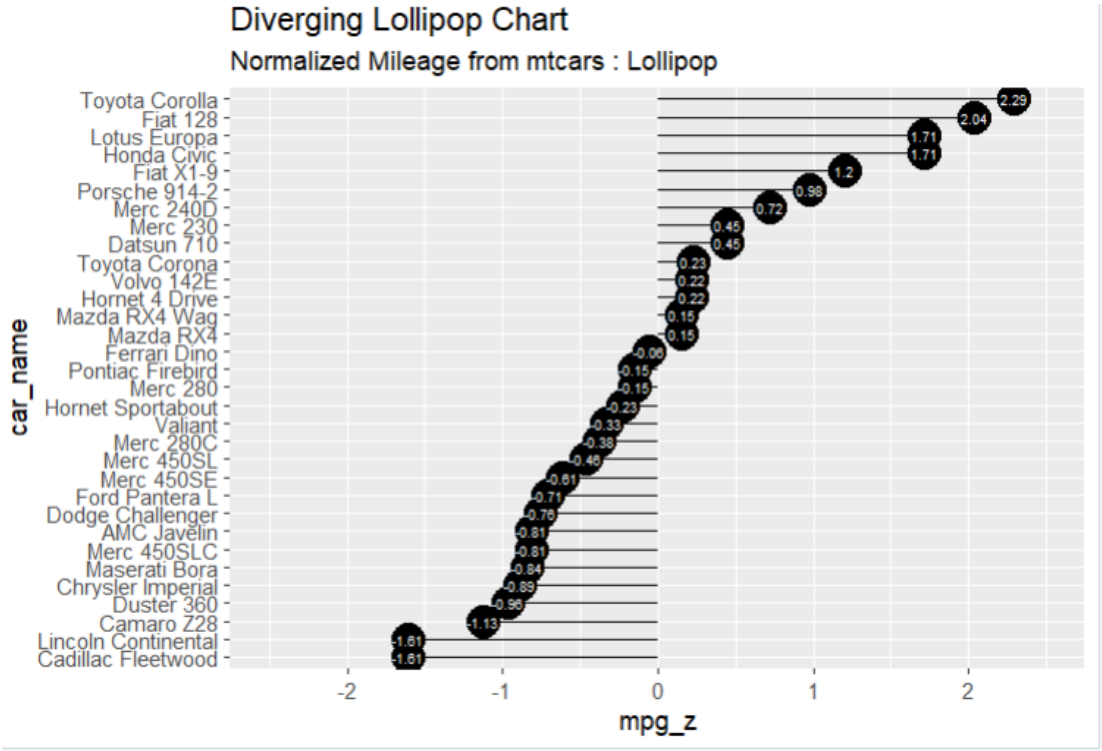
```
# Bubble Plots and Count Charts

filtered_mpg <- subset(mpg, manufacturer %in% c("audi", "ford", "honda",
"hyundai"))

# Create the bubble chart
bubble_chart <- ggplot(filtered_mpg, aes(x = displ, y = cty, size = hwy,
color = manufacturer)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE) + # Add trend lines without
confidence intervals
  labs(title = "Bubble chart",
       subtitle = "mpg: Displacement vs City Mileage",
       x = "displ",
       y = "cty",
       size = "hwy",
       color = "manufacturer") +
  theme_minimal()

bubble_chart
```



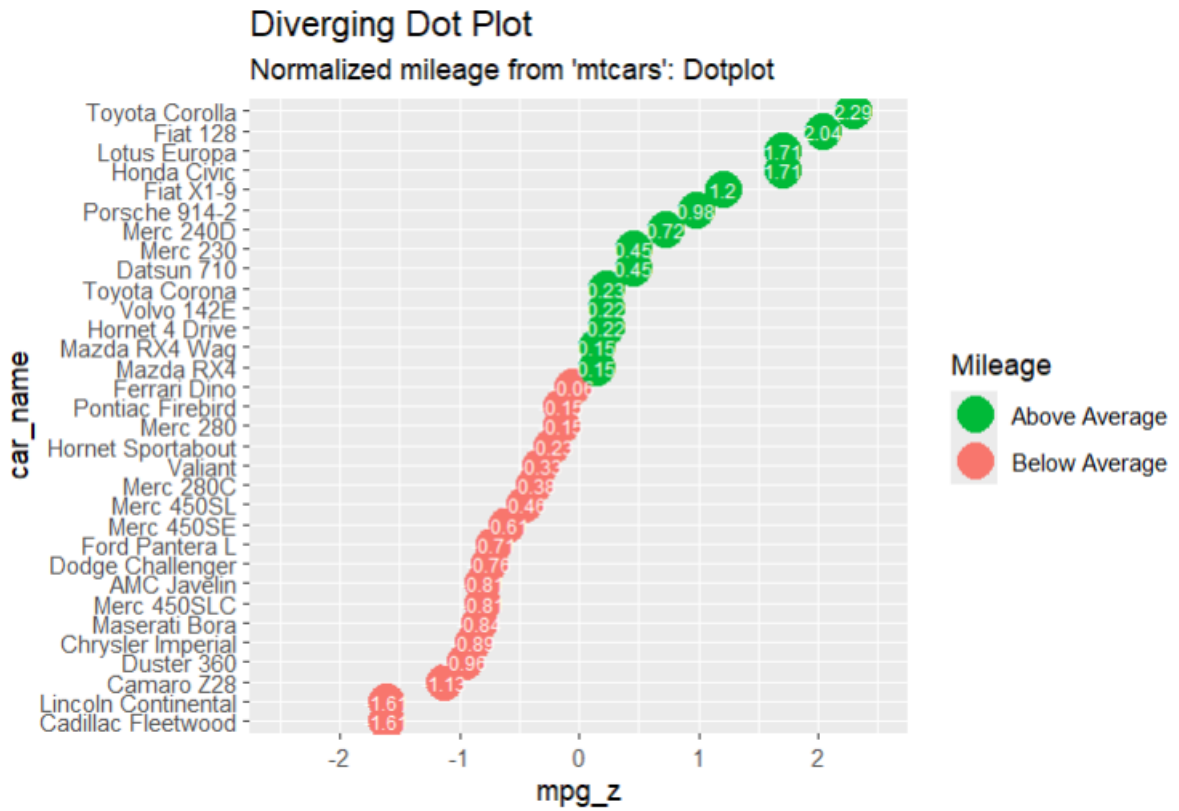
```
# Diverging Charts
# Add car names and calculate z-scores for mpg
mtcars$car_name <- rownames(mtcars)
mtcars$mpg_z <- round((mtcars$mpg - mean(mtcars$mpg)) / sd(mtcars$mpg), 2)

# Define the mpg_type based on z-scores
mtcars$mpg_type <- ifelse(mtcars$mpg_z < 0, "below", "above")

# Sort the data based on mpg_z
mtcars <- mtcars[order(mtcars$mpg_z),]

# Convert car names to a factor with the same order to retain sorted order
in the plot
mtcars$car_name <- factor(mtcars$car_name, levels = mtcars$car_name)

# Plot
ggplot(mtcars, aes(x = car_name, y = mpg_z, label = mpg_z)) +
  geom_bar(stat = 'identity', aes(fill = mpg_type), width = 0.5) +
  scale_fill_manual(name = "Mileage",
                    labels = c("Above Average", "Below Average"),
                    values = c("above" = "#00ba38", "below" = "#f8766d"))
+
  labs(subtitle = "Normalized mileage from 'mtcars'",
       title = "Diverging Bars") +
  coord_flip()
```



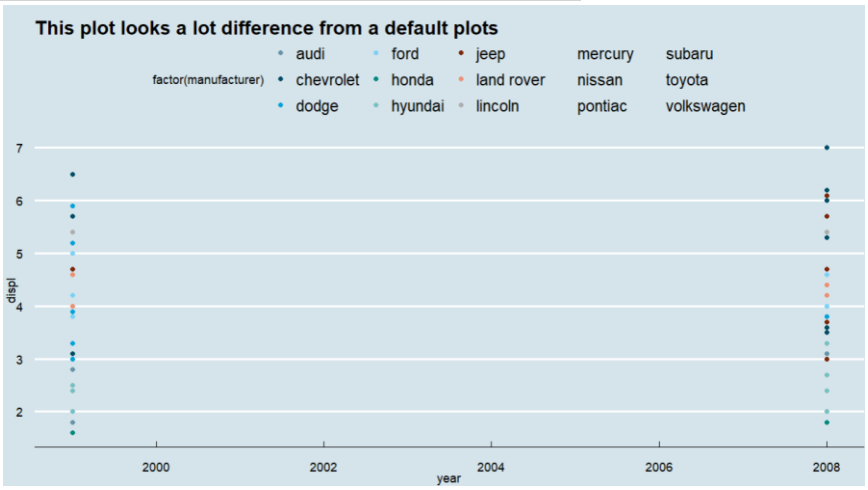
```
# Diverging Lollipop Chart
ggplot(mtcars , aes(x=car_name,y=mpg_z, label=mpg_z))+
  geom_point(stat="identity",fill= "red",size=6) +
  geom_segment(aes(y=0,
                   x= car_name,
                   yend=mpg_z,
                   xend=car_name),
               color="black")+
  geom_text(color="white",size=2)+
  labs(title="Diverging Lollipop Chart",
       subtitle="Normalized Mileage from mtcars : Lollipop" )+
  ylim(-2.5,2.5) +
  coord_flip()
```



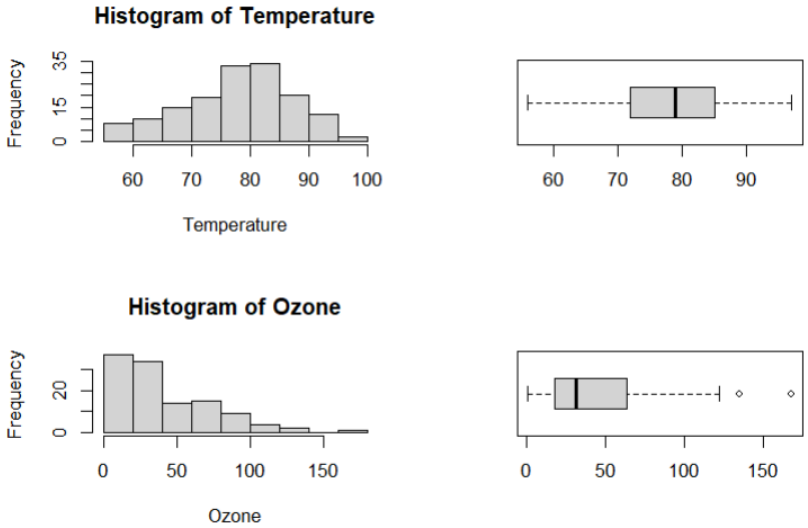

```
# Diverging Dot plot
ggplot(mtcars, aes(x=car_name, y=mpg_z, label=round(mpg_z, 2))) +
  geom_point(stat='identity', aes(col=mpg_type), size=6) +
  scale_color_manual(name="Mileage",
                     labels = c("Above Average", "Below Average"),
                     values = c("above"="#00ba38", "below"="#f8766d")) +
  geom_text(color="white", size=2.5, hjust=0.5, vjust=0.5) +
  labs(title="Diverging Dot Plot",
       subtitle="Normalized mileage from 'mtcars': Dotplot") +
  ylim(-2.5, 2.5) +
  coord_flip()
```



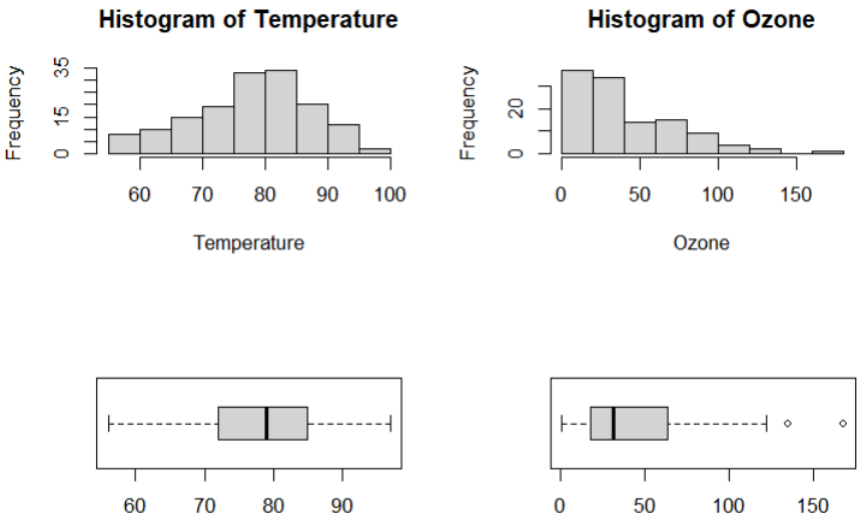

```
# Themes
install.packages("ggthemes")
library(ggthemes)
mpg
ggplot(mpg,aes(year,displ ,color=factor(manufacturer))) +
  geom_point() +
  ggtitle("This plot looks a lot difference from a default plots")+
  theme_economist() + scale_color_economist()
```



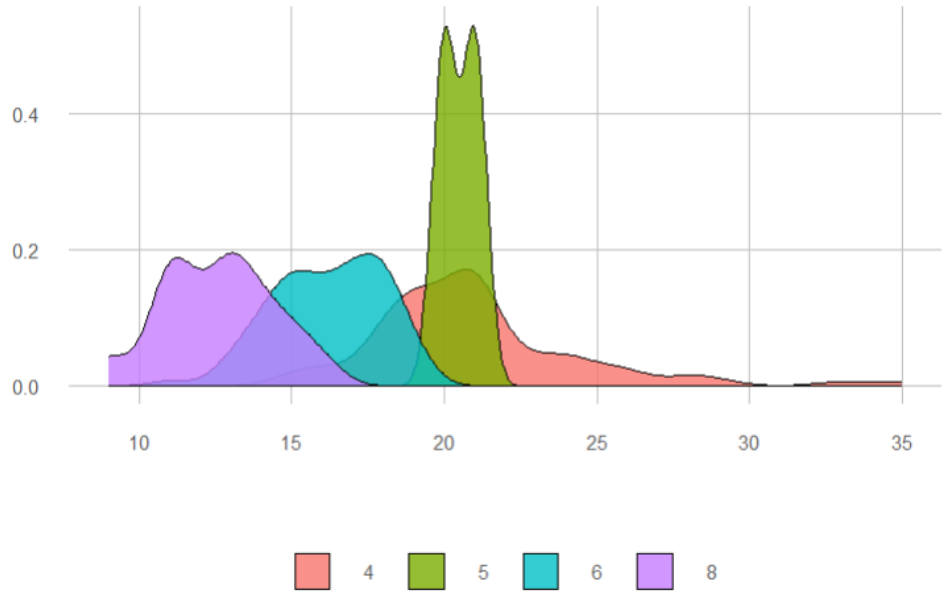
```
# Multi Pnael plots
par(mfrow=c(1,2))
airquality
colnames(airquality)
Temperature <- airquality$Temp
Ozone <- airquality$Ozone
par(mfrow=c(2,2))
hist(Temperature)
boxplot(Temperature,horizontal=TRUE)
hist(Ozone)
boxplot(Ozone,horizontal = TRUE)
```



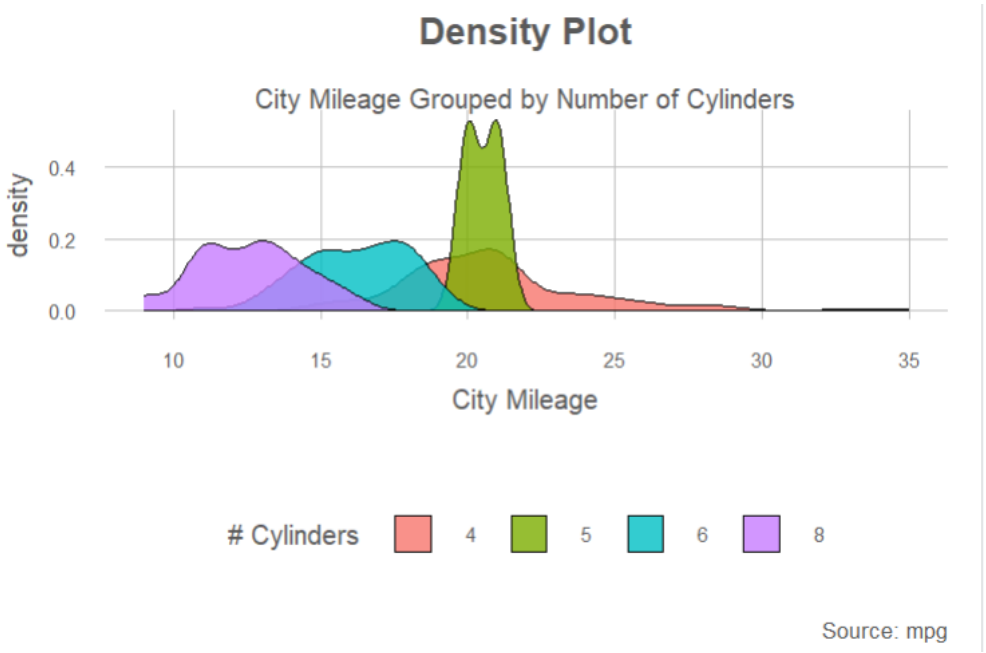

```
# same but now in columns
par(mfcol=c(2,2))
hist(Temperature)
boxplot(Temperature,horizontal=TRUE)
hist(Ozone)
boxplot(Ozone,horizontal = TRUE)
```



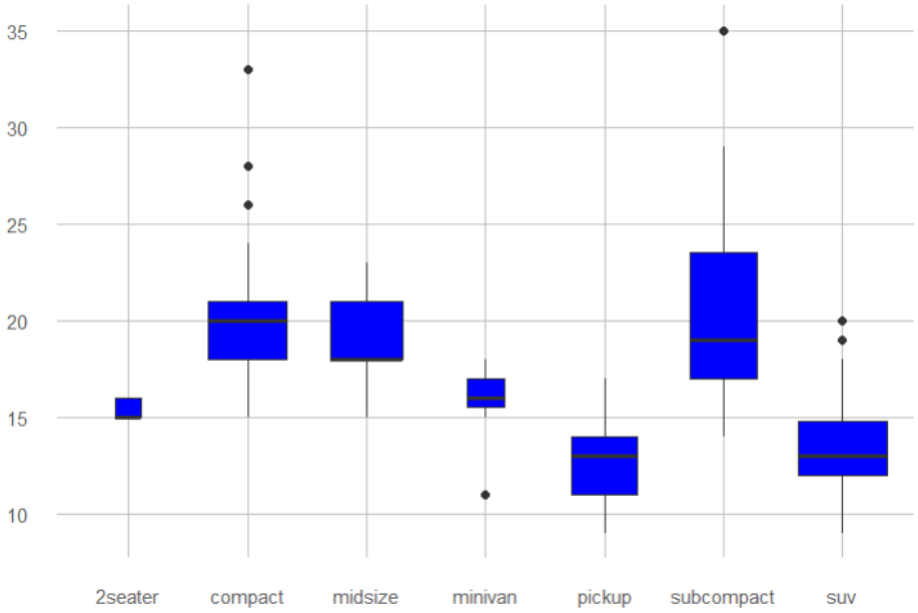
```
# Multiple Plots
head(mpg)
# density plot
p<- ggplot(mpg,aes(cty)) +
  geom_density(aes(fill=factor(cyl)) ,alpha =0.8)
p
```



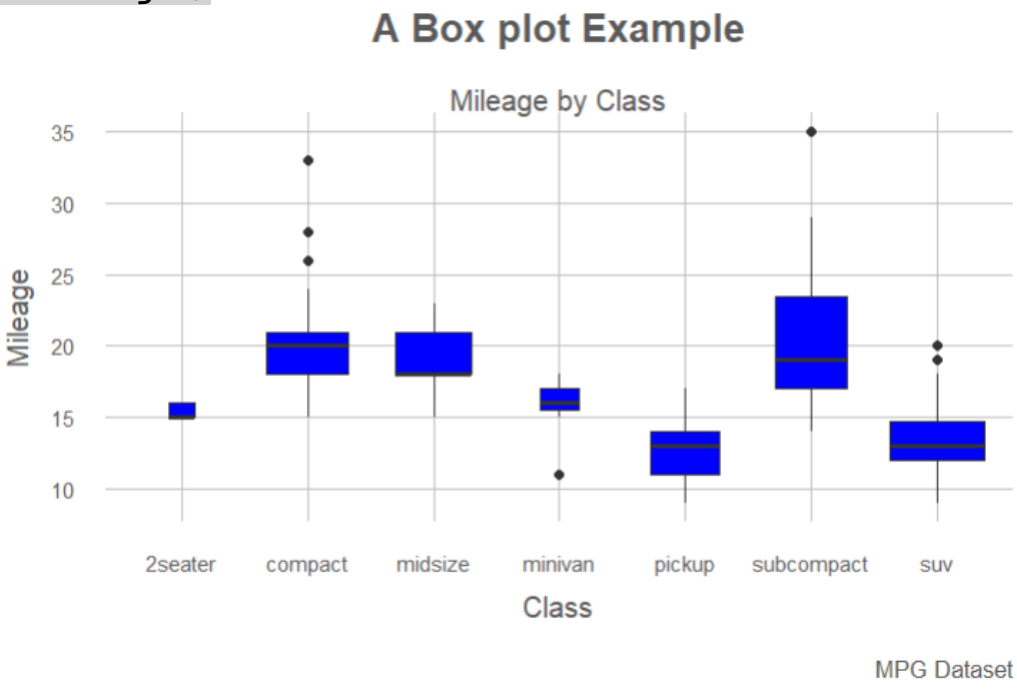

```
p + labs(title="Density Plot",
         subtitle = "City Mileage Grouped by number of cylinders",
         caption = "Source : mpg",
         x="City Mileage",
         fill="# Cylinders")
```



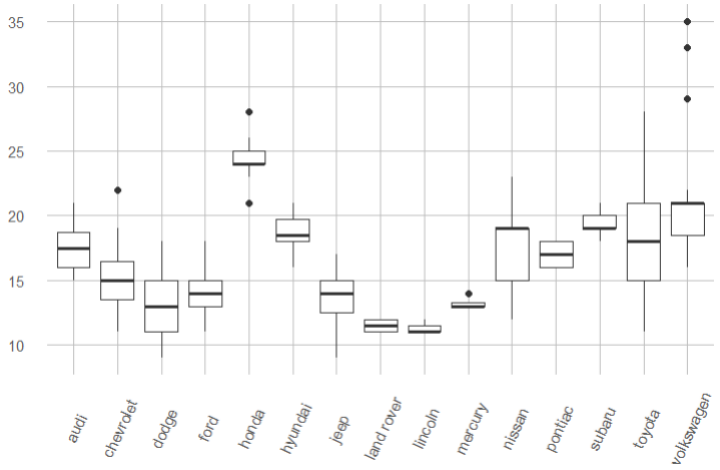
```
# Box Plot
p<- ggplot(mpg,aes(class,cty)) +
  geom_boxplot(varwidth=T , fill ="blue")
```



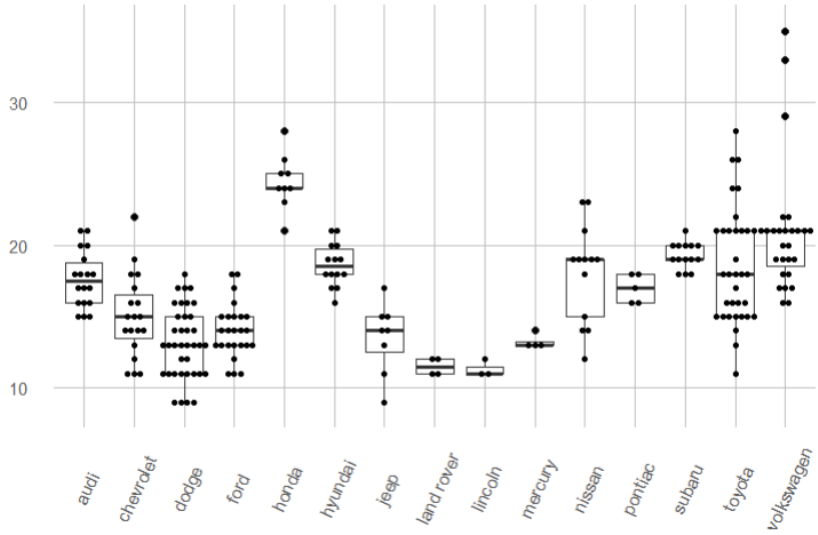

```
p + labs(title="A Box plot Example",
         subtitle="Mileage by Class",
         caption="MPG Dataset",
         x="Class",
         y="Mileage")
```



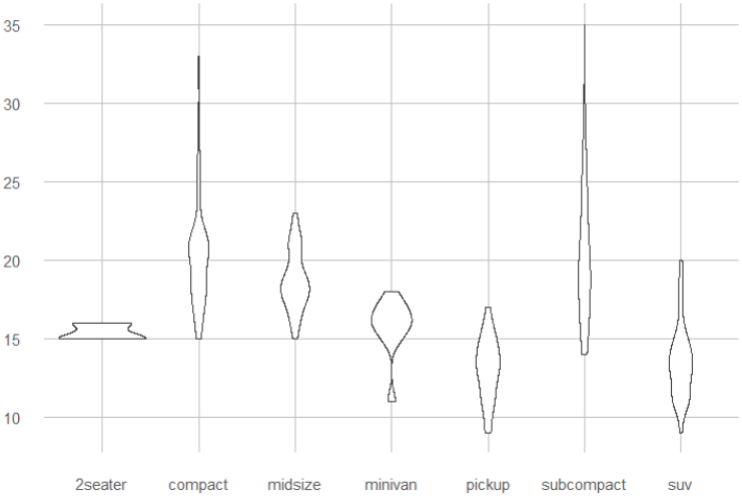
```
# DOT PLOT
p <- ggplot(mpg, aes(manufacturer, cty)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
p
```



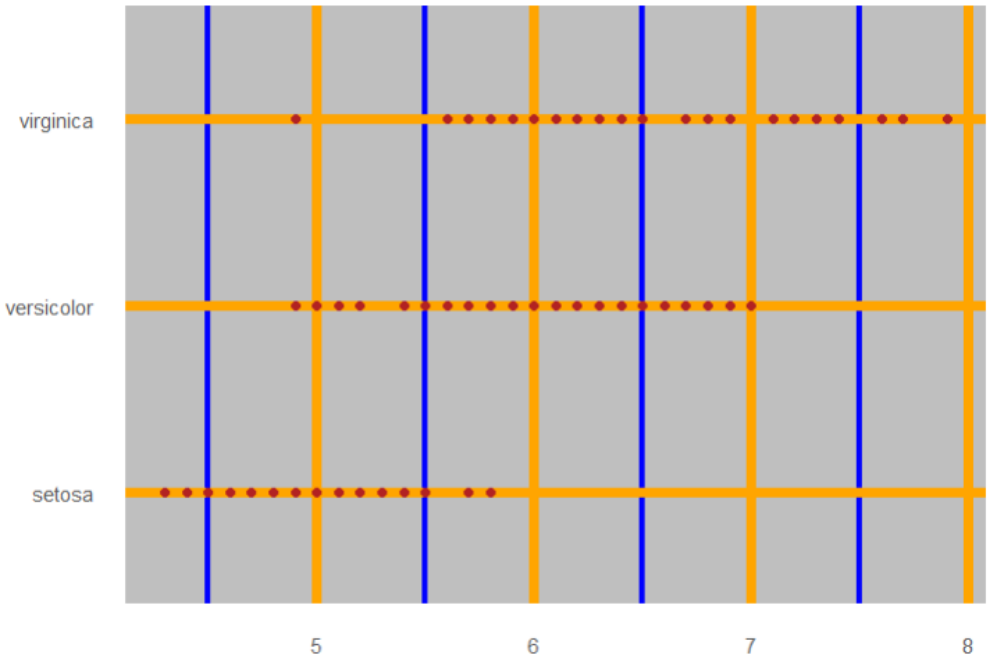

```
p + geom_dotplot(binaxis='y',
                 stackdir='center',
                 dotsize = .5)
```



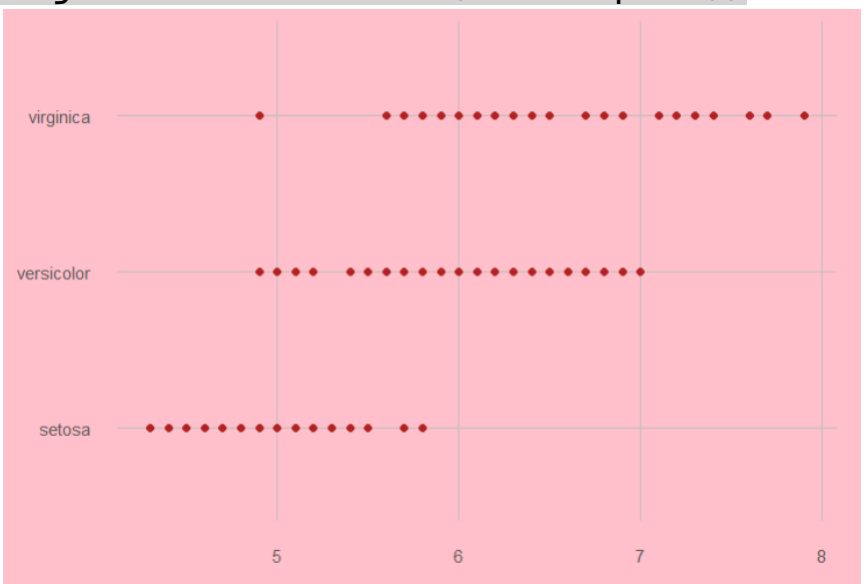

```
# Violin Plot
p <- ggplot(mpg, aes(class, cty))
p + geom_violin()
```



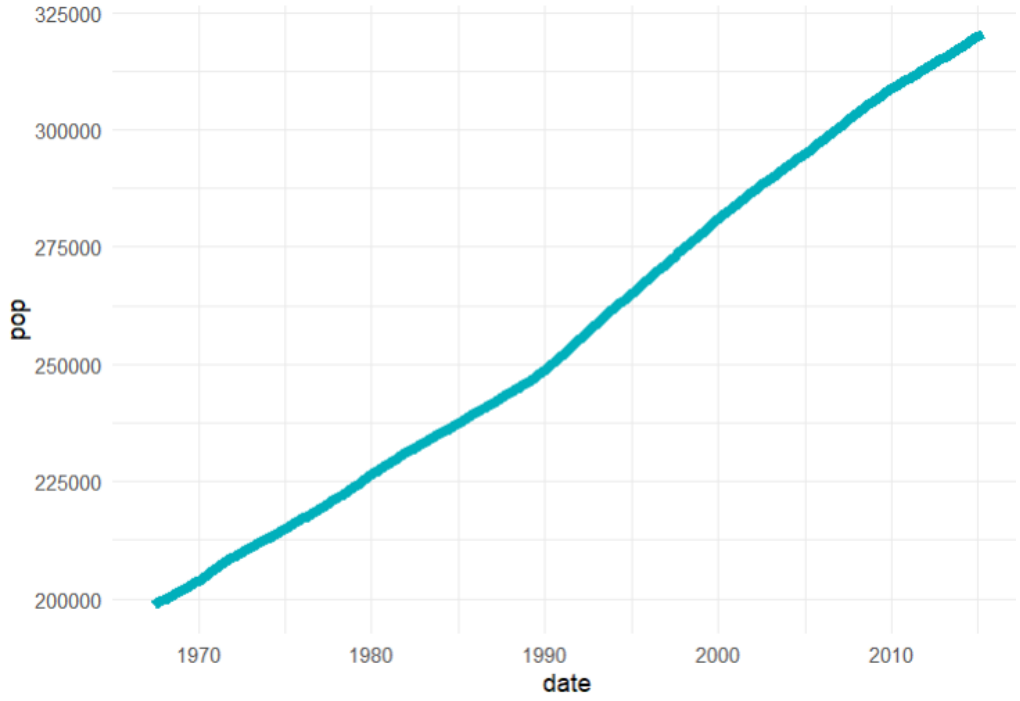
```
# Background Colors
ggplot(iris, aes(Sepal.Length, Species))+geom_point(color="firebrick")+
  theme(panel.background = element_rect(fill = 'grey75'),
        panel.grid.major = element_line(colour = "orange", size=2),
        panel.grid.minor = element_line(colour = "blue"))
```



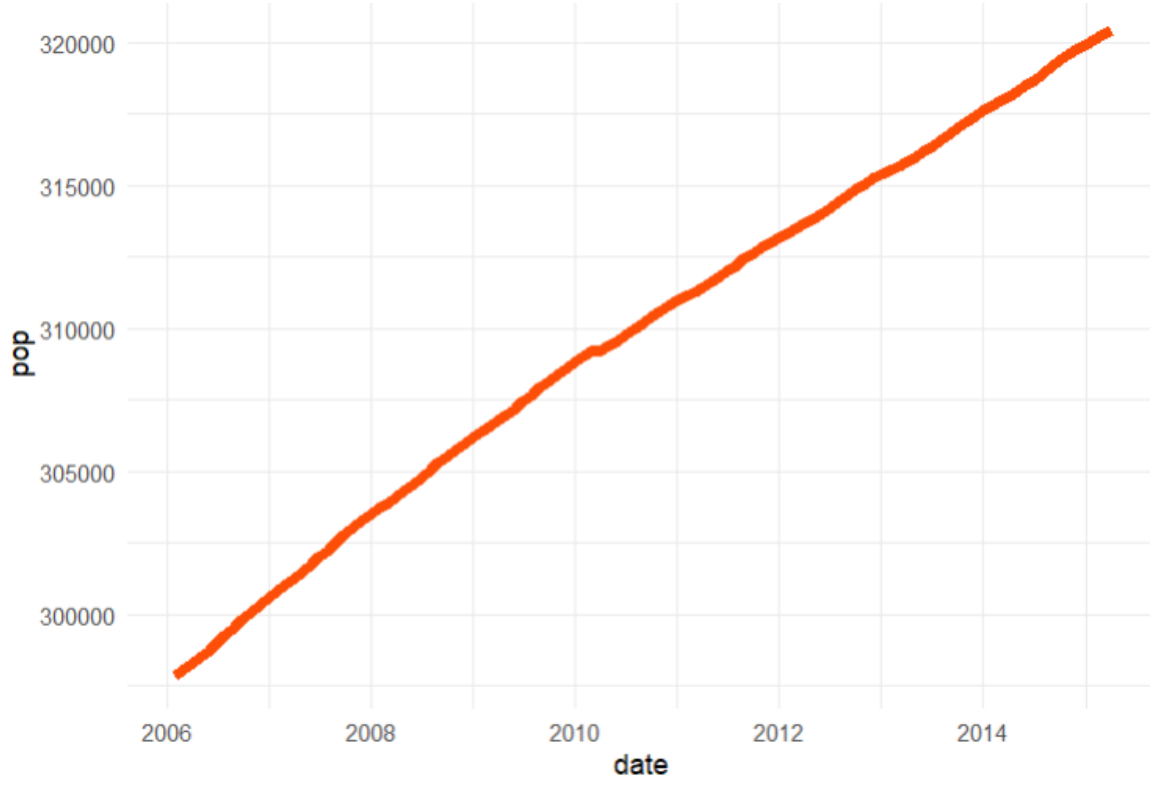

```
ggplot(iris, aes(Sepal.Length, Species))+geom_point(color="firebrick")+
  theme(plot.background = element_rect(fill = 'pink'))
```



```
# Time Series
theme_set(theme_minimal())
head(economics)

# Basic line plot
ggplot(data = economics, aes(x = date, y = pop))+
   geom_line(color = "#00AFBB", size = 2)
```




```
# Plot a subset of the data
ss <- subset(economics, date > as.Date("2006-1-1"))
ggplot(data = ss, aes(x = date, y = pop)) +
  geom_line(color = "#FC4E07", size = 2)
```



```
# Creating Time Series
install.packages("tidyr")
install.packages("dplyr")
library(tidyr)
library(dplyr)
df <- economics %>%
  dplyr::select(date, psavert, uempmed) %>%
  tidyr::gather(key = "variable", value = "value", -date)
head(df, 3)

ggplot(df, aes(x = date, y = value)) +
  geom_line(aes(color = variable), size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(title="Time Series")+
  theme_minimal()
```