Instructor: Nara Yoon, Ph.D. (yoonnx@jmu.edu)

Spring 2024 | Thursdays 4-7pm, Hart 3008

Data Analytics for Managerial Leadership

WK 13: Plotting

PLAN FOR TODAY

Overview of core concepts

- Plot types
- Simple plotting – with **plot()**
- Pretty plotting – with **ggplot()**

READINGS

Baumer, Kaplan, & Horton.

- Data visualization principles (Ch 2.1-2.4)
- Grammar of graphics (Ch 3.1-3.2)

Simple plotting:
plot()

PLOT TYPES

Histogram

Box and whisker plot

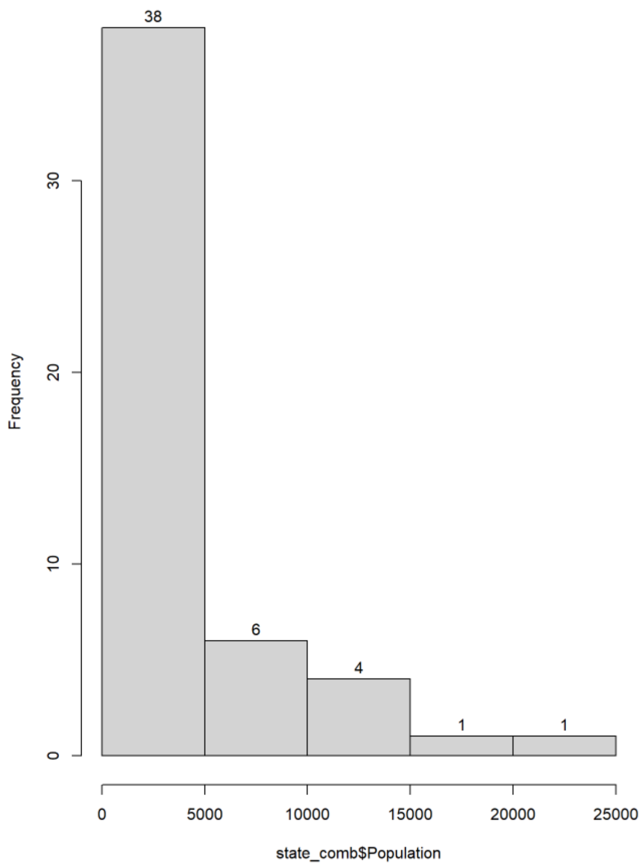
Bar plot

Scatter plot

HISTOGRAMS

Shows **counts** of data

```
hist(state_comb$Population, labels=T)
```

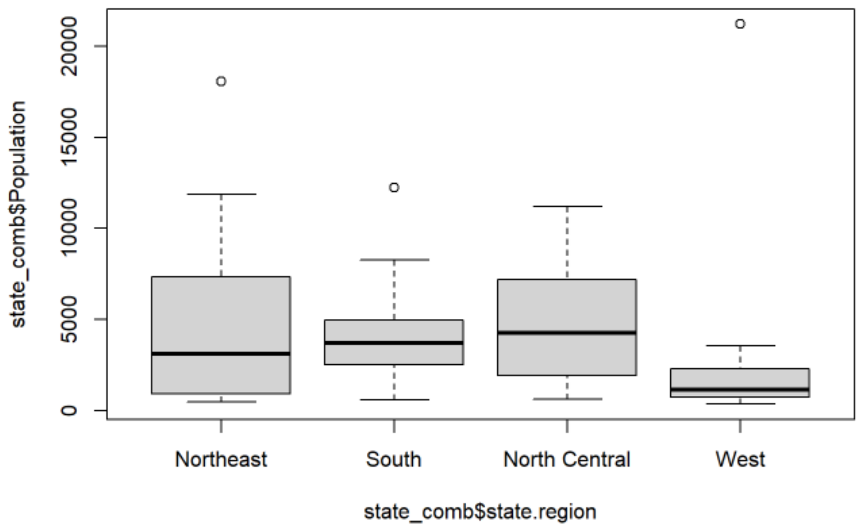


BOX AND WHISKER PLOT

Shows **five number summary**

- Min
- Q1 (first quartile)
- Median
- Q3 (third quartile)
- Max

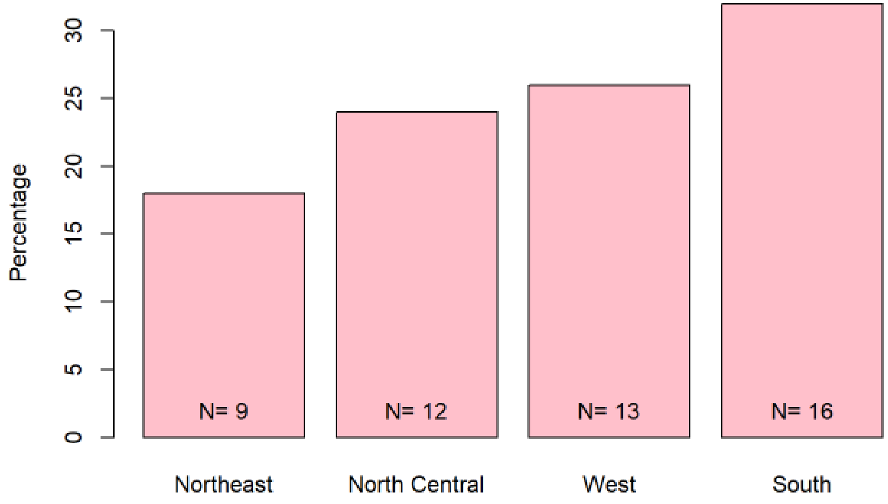
```
boxplot(state_comb$Population~state_comb$state.region)
```



BAR PLOT

Similar to histogram (shows the **counts** of data)

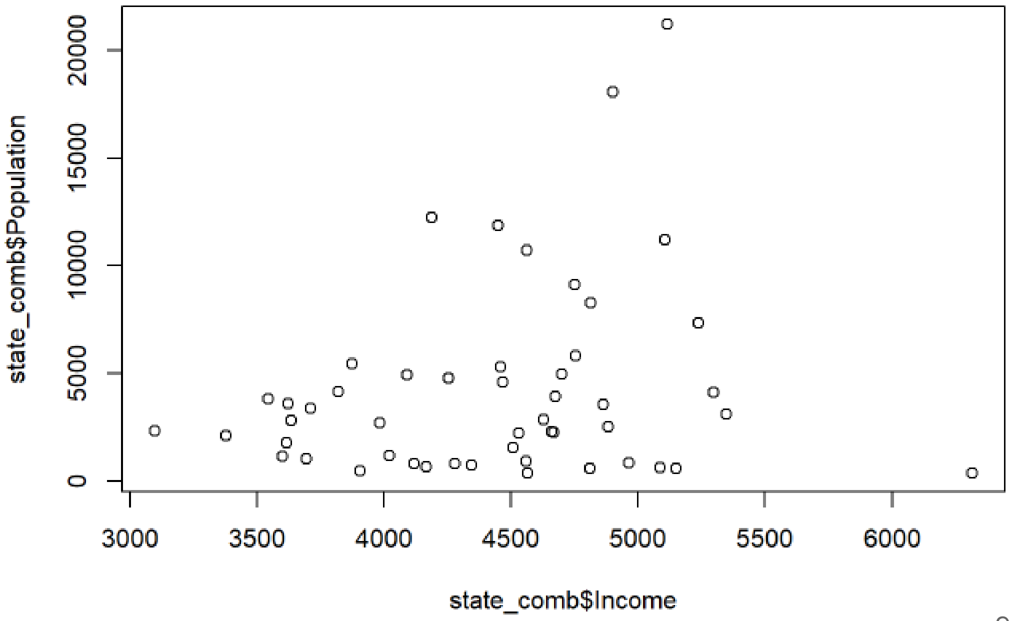
```
count_region <- sort(table(state_comb$state.region))
total_n <- length (state_comb$state.region)
percent <- 100 * (count_region / total_n)
barplot(percent, ylab="Percentage", col="pink")
text(x=seq(0.7, 5, 1.2), 2, paste("N=",count_region))
```



SCATTER PLOT

Shows relationship between “**two numerical**” **variables**

```
plot(state_comb$Income, state_comb$Population, pch=1)
```



More about scatterplot

LOAD DATA (state dataset)

```
state_x77 <- as.data.frame(state.x77)                                    # making as data frame
state_x77 <- state_x77 %>% rownames_to_column(var="state_name")  # assigning the first column as state name
state_abb <- as.data.frame(state.abb)
state_region <- as.data.frame(state.region)
```

```
state_comb <- cbind(state_abb, state_x77, state_region)
```

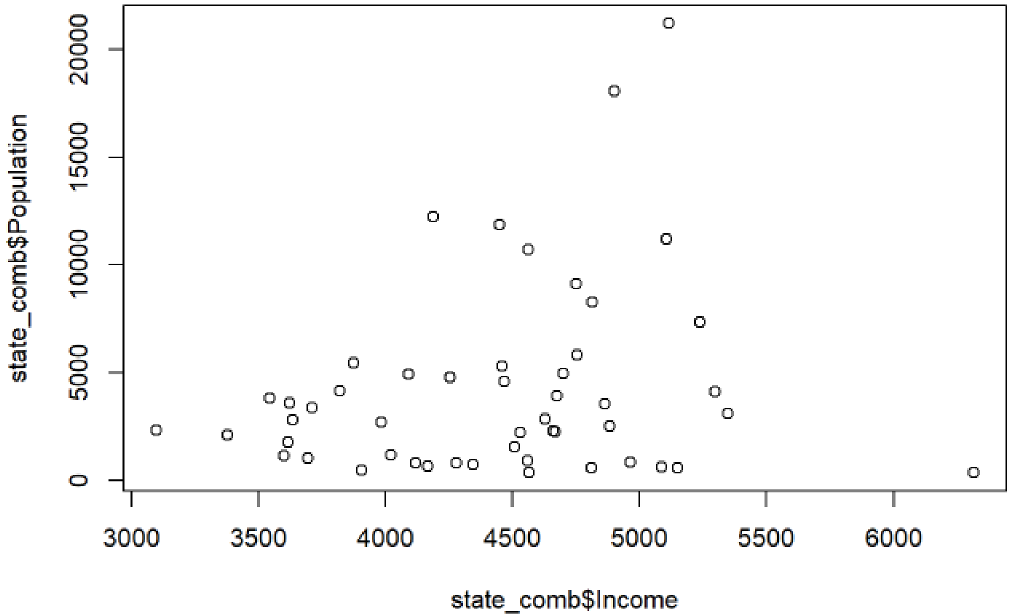
COMBINED DATASET: STRUCTURE

```
str(state_comb)
```

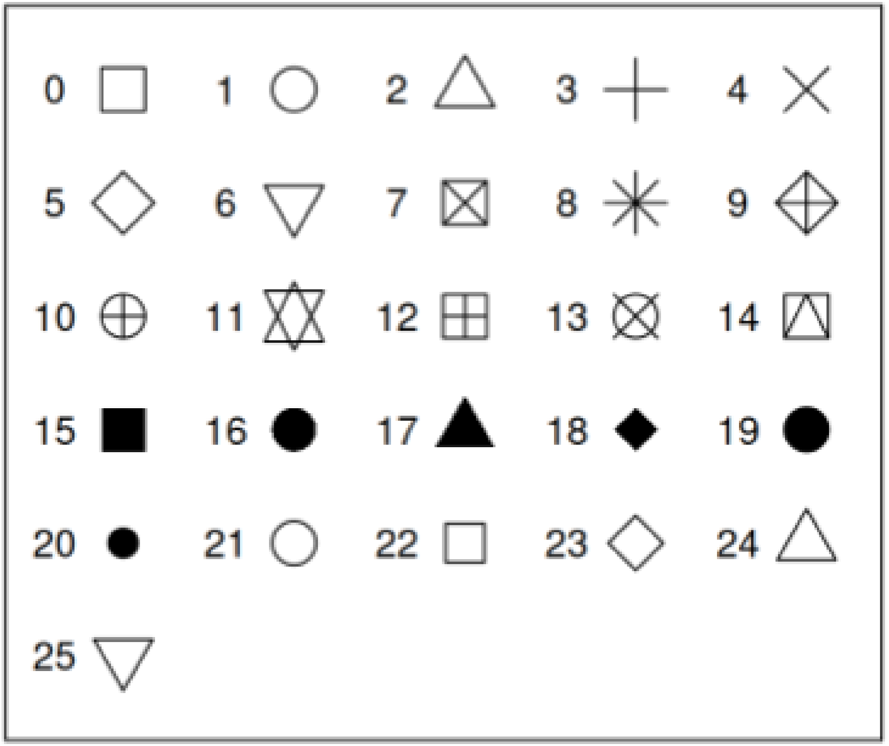
```
## 'data.frame':    50 obs. of  11 variables:
##  $ state.abb   : chr  "AL" "AK" "AZ" "AR" ...
##  $ state_name  : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ Population  : num  3615 365 2212 2110 21198 ...
##  $ Income      : num  3624 6315 4530 3378 5114 ...
##  $ Illiteracy  : num  2.1 1.5 1.8 1.9 1.1 0.7 1.1 0.9 1.3 2 ...
##  $ Life Exp    : num  69 69.3 70.5 70.7 71.7 ...
##  $ Murder      : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ...
##  $ HS Grad     : num  41.3 66.7 58.1 39.9 62.6 63.9 56 54.6 52.6 40.6 ...
##  $ Frost       : num  20 152 15 65 20 166 139 103 11 60 ...
##  $ Area        : num  50708 566432 113417 51945 156361 ...
##  $ state.region: Factor w/ 4 levels "Northeast","South",..: 2 4 4 2 4 4 1 2 2 2 ...
```

START WITH SIMPLE SCATTER PLOT

```r
plot(state_comb$Income, state_comb$Population, pch=1)
```

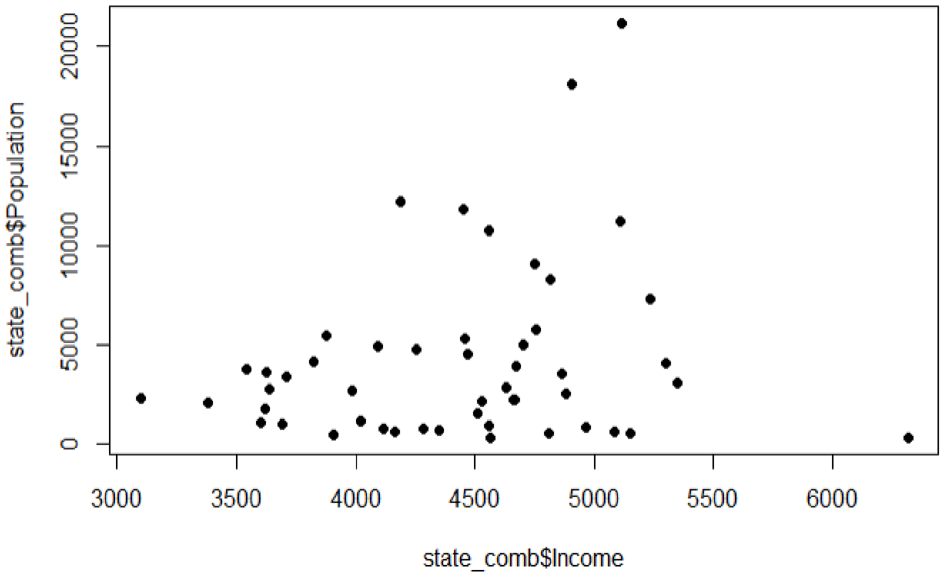


pch symbols



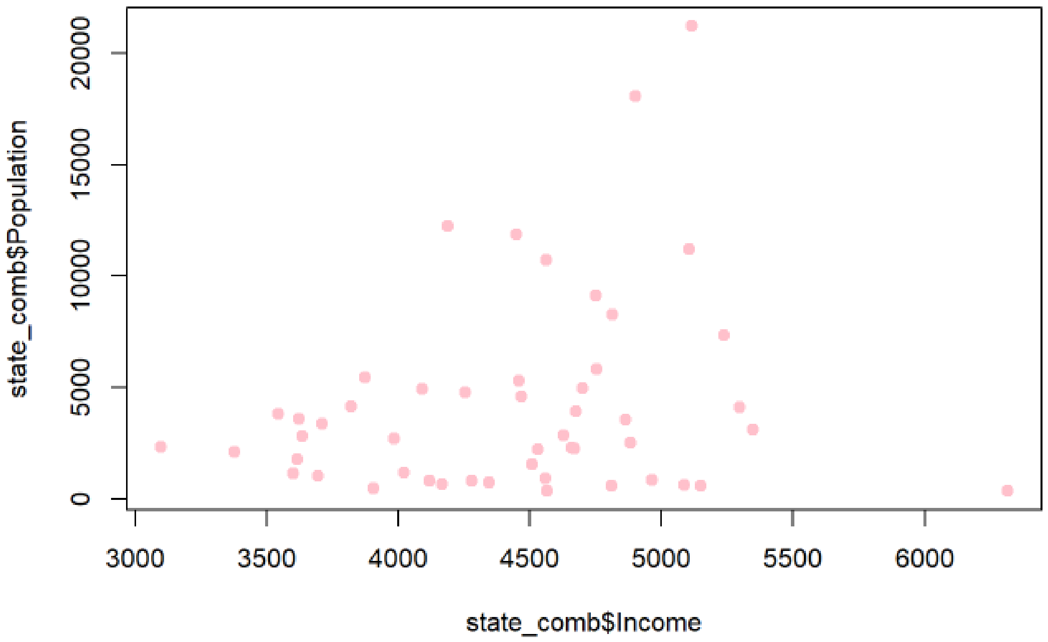
PUTTING DIFFERENT SYMBOL

```
plot(state_comb$Income, state_comb$Population, pch=19)
```



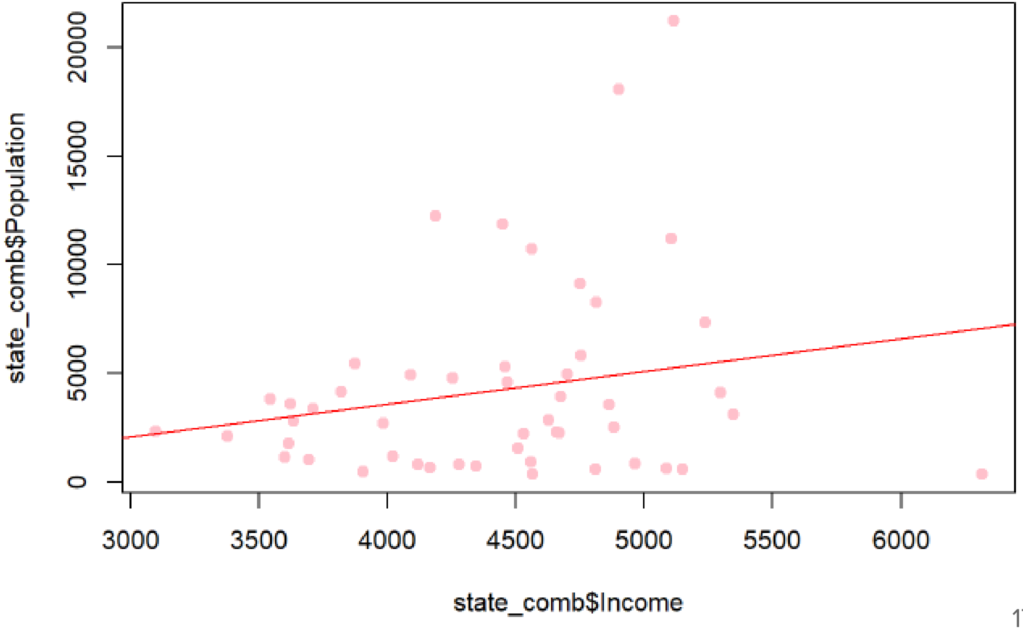
PUTTING COLORS TO SYMBOL

```
plot(state_comb$Income, state_comb$Population, pch=19, col="pink")
```



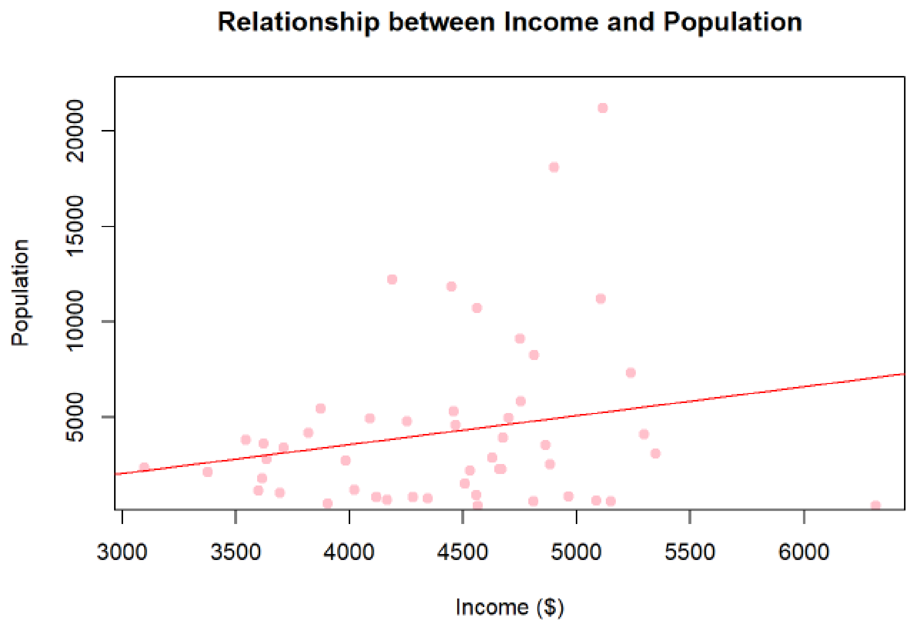
PUTTING LINEAR REGRESSION LINE

```
plot(state_comb$Income, state_comb$Population, pch=19, col="pink")
abline(lm(state_comb$Population~state_comb$Income), col="red")     # regression line (y~x)
```



PUTTING TITLE & LABEL

```
plot(state_comb$Income, state_comb$Population, pch=19, col="pink", ylim=c(1000,22000), ylab="Population", xlab="Income ($)", main ="Relationship between Income and Population" )
abline(lm(state_comb$Population~state_comb$Income),col="red")
```



Exploiting **categorical variable**:
By using different symbol and color!

"state region" is **categorical** (factor) variable with 4 levels →

```
str(state_comb$state.region)
```

```
##  Factor w/ 4 levels "Northeast","South",
```

Which are those 4 level? →

```
> unique(state_comb$state.region)
[1] South         West          Northeast     North Central
```

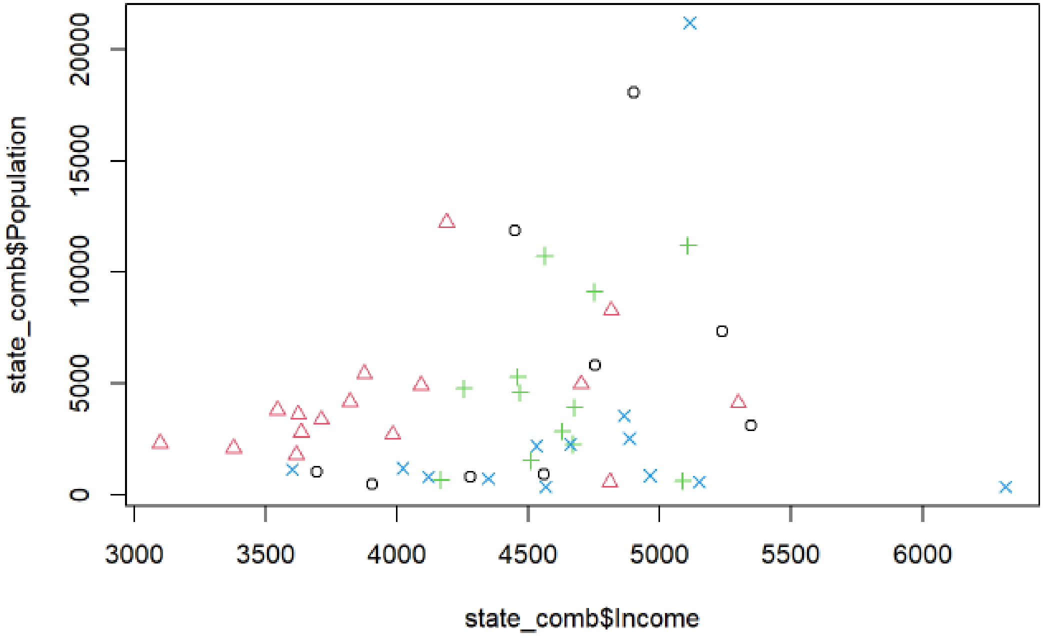
Change variable form as **numerical** →

```
numeric_region <- as.numeric(state_comb$state.region)
numeric_region
```

```
##  [1] 2 4 4 2 4 4 1 2 2 2 4 4 3 3 3 3 2 2 1 2 1 3 3
## [39] 1 2 3 2 2 4 1 2 4 2 3 4
```

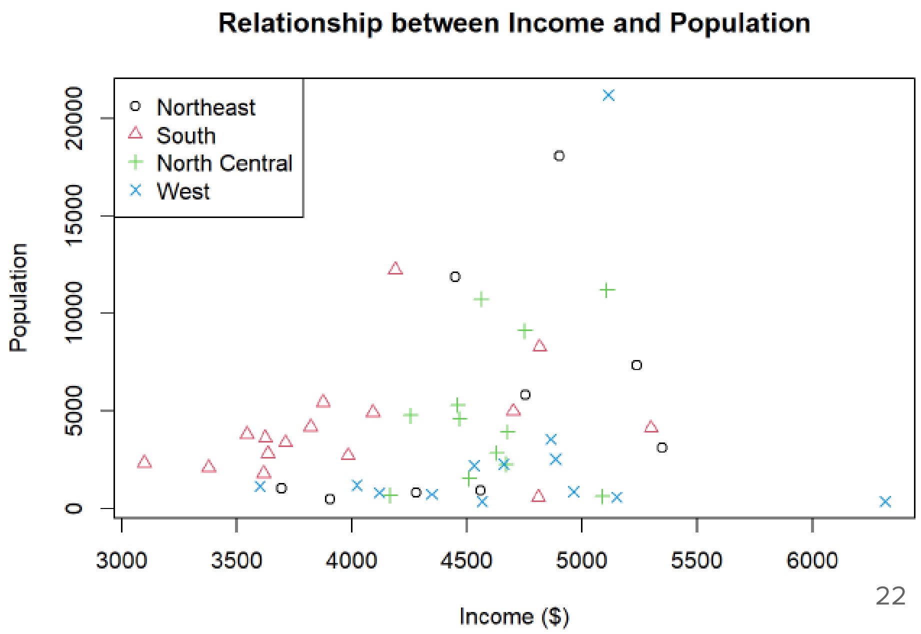
PUTTING DIFFERENT SYMBOL

```
plot(state_comb$Income, state_comb$Population, pch=numeric_region, col=numeric_region)
```

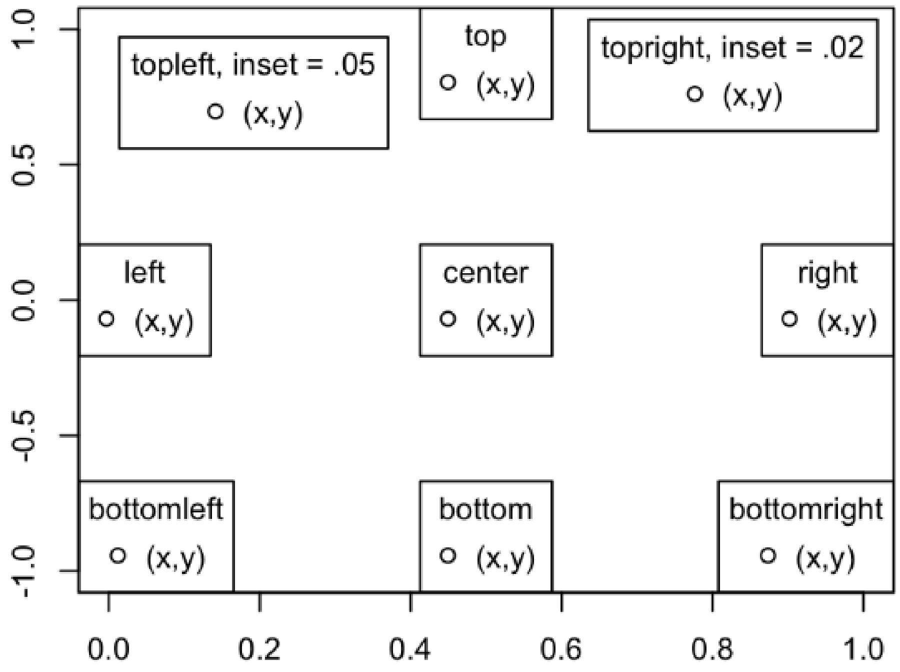


PUTTING TITLE, LABEL, LEGEND

```
plot(state_comb$Income, state_comb$Population,
     pch=numeric_region,
     col=numeric_region,
     xlab="Income ($)",
     ylab="Population",
     main="Relationship between Income and Population")
legend("topleft",
       levels(state_comb$state.region),
       pch=1:4,
       col=1:4)
```



LEGEND POSITION



Exercise scatterplot

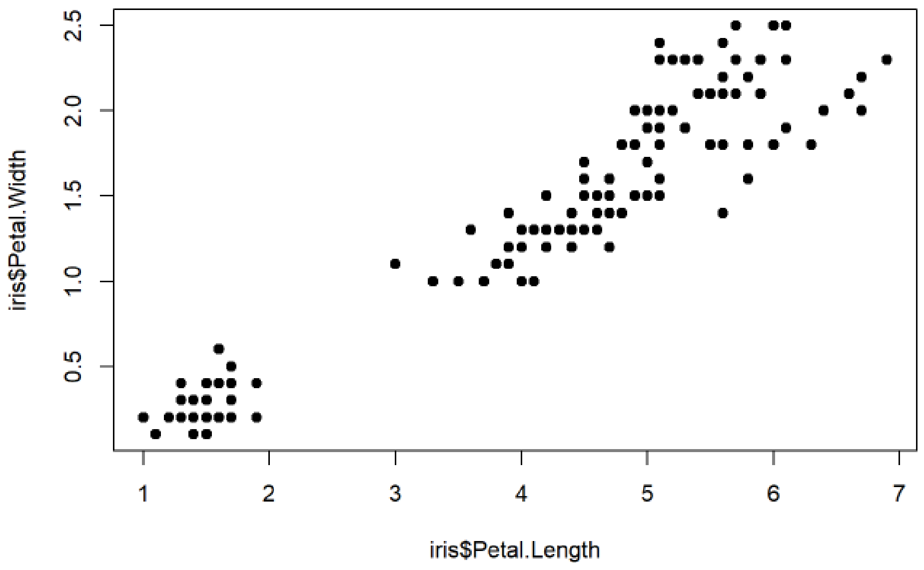
LOAD DATA (iris)

```
iris <- iris
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1
```

SIMPLE PLOTTING:
RELATIONSHIP BTW TWO NUMERIC VARIABLES

```
plot(iris$Petal.Length, iris$Petal.Width, pch=19)
```



Note that "**Species**" variable is **categorical**

```
str(iris$Species)
```

```
##  Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
unique(iris$Species)
```

```
## [1] setosa     versicolor virginica
```

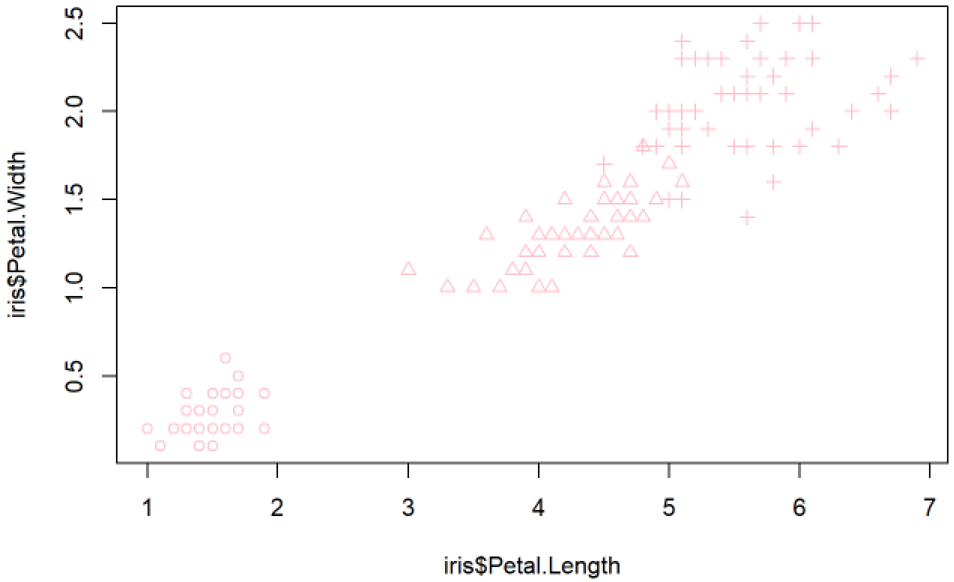
We want to use different symbol (pch) for each species ("Species")

To do so, create a new object in "numerical" variable form

```
numeric_species <- as.numeric(iris$Species)
numeric_species
```

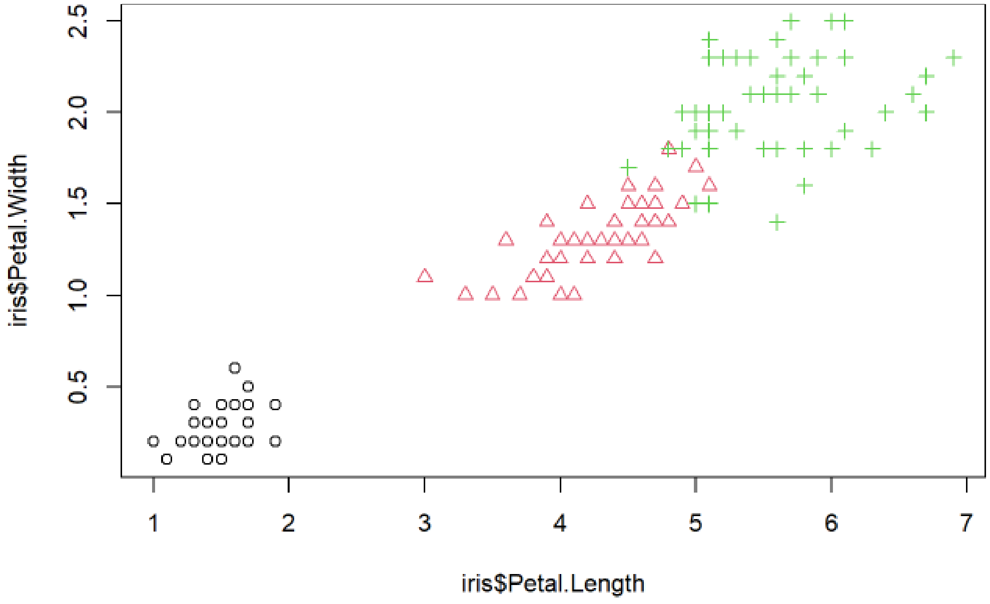
PUTTING DIFFERENT SYMBOL SHAPE

```
plot(iris$Petal.Length, iris$Petal.Width, pch=numeric_species, col="pink")
```



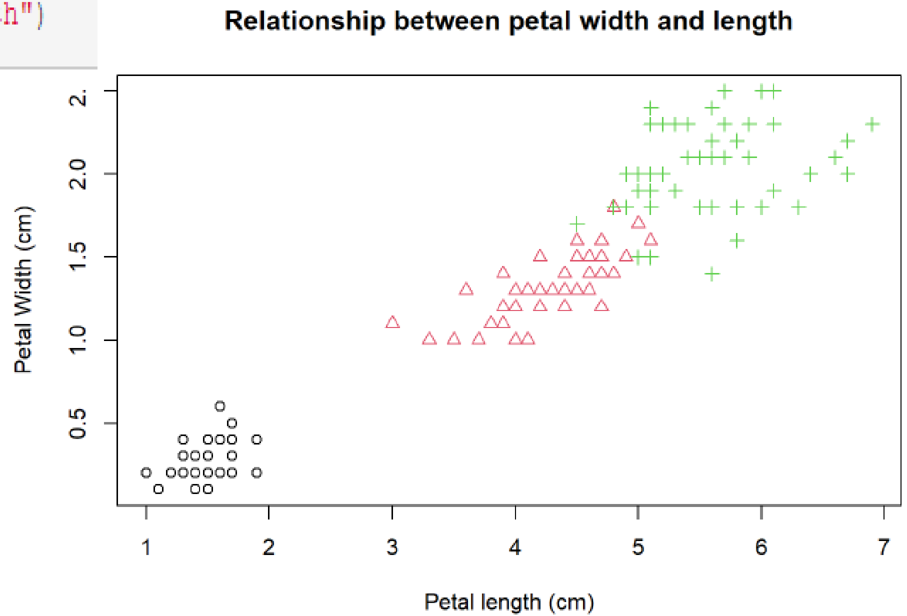
PUTTING DIFFERENT COLORS TO DIFFERENT SYMBOLS

```
plot(iris$Petal.Length, iris$Petal.Width, pch=numeric_species, col=numeric_species)
```



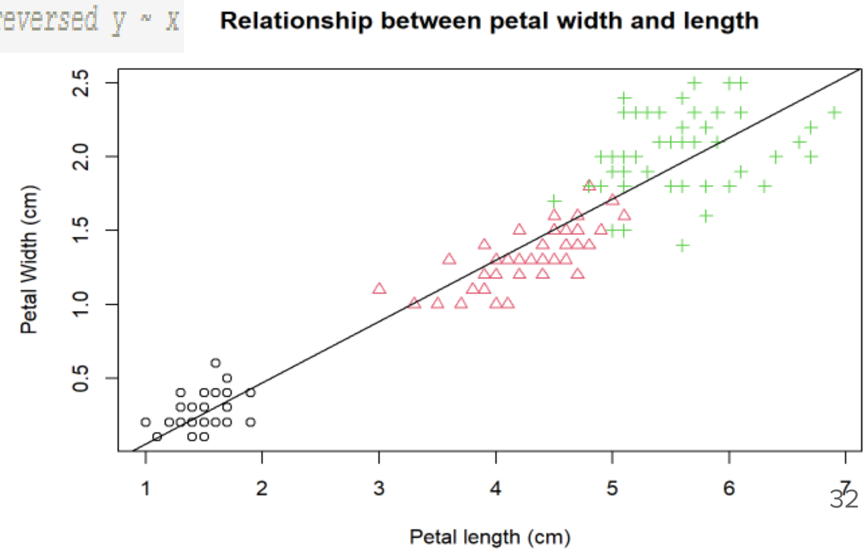
PUTTING TITLE & LABEL

```
plot(iris$Petal.Length, iris$Petal.Width,
     pch=numeric_species,
     col=numeric_species,
     xlab="Petal length (cm)",
     ylab="Petal Width (cm)",
     main="Relationship between petal width and length")
```



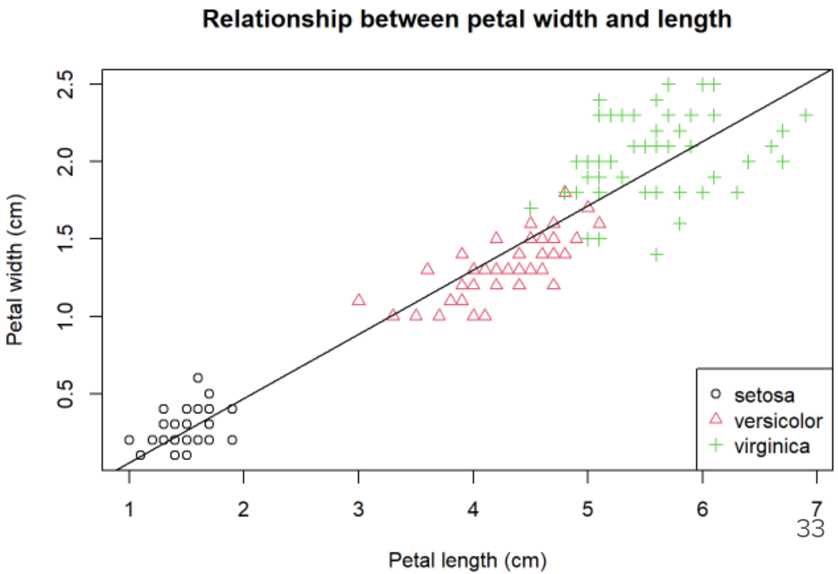
PUTTING LINEAR REGRESSION LINE

```
plot(iris$Petal.Length, iris$Petal.Width,
     pch=numeric_species,
     col=numeric_species,
     xlab="Petal length (cm)",
     ylab="Petal Width (cm)",
     main="Relationship between petal width and length")
abline(lm(iris$Petal.Width~iris$Petal.Length))     # the order is reversed y ~ x
```



PUTTING LEGEND

```
plot(iris$Petal.Length, iris$Petal.Width,
     pch=numeric_species,
     col=numeric_species,
     xlab="Petal length (cm)",
     ylab="Petal width (cm)",
     main="Relationship between petal width and length")
abline(lm(iris$Petal.Width~iris$Petal.Length))
legend("bottomright",
       levels(iris$Species),
       pch=1:3,
       col=1:3)
```



ggplot2: Background

ggplot2

- Geometric shape
- Aesthetic
- Coordinate system
- Scale

INSTALLING & LOADING PACKAGES

```
#install.packages("ggplot2")
#install.packages("tidyverse")
#install.packages("RColorBrewer")

library(ggplot2)
library(tidyverse)
library(RColorBrewer)
```

MAJOR COMPONENTS: EXAMPLES

geom_function

- geom_point (scatterplot)
- geom_bar (bar)
- geom_histogram (histogram)
- geom_boxplot (box and whisker)
- geom_line (line)

aesthetics

- x position
- y position
- color
- fill
- shape

CHECK DATASET DESCRIPTION

Fuel economy data from 1999 to 2008 for 38 popular models of cars

Description

This dataset contains a subset of the fuel economy data that the EPA makes available on https://fueleconomy.gov/. It contains only models which had a new release every year between 1999 and 2008 - this was used as a proxy for the popularity of the car.

LOAD DATA (mpg)

```
mpg.data<-mpg
```

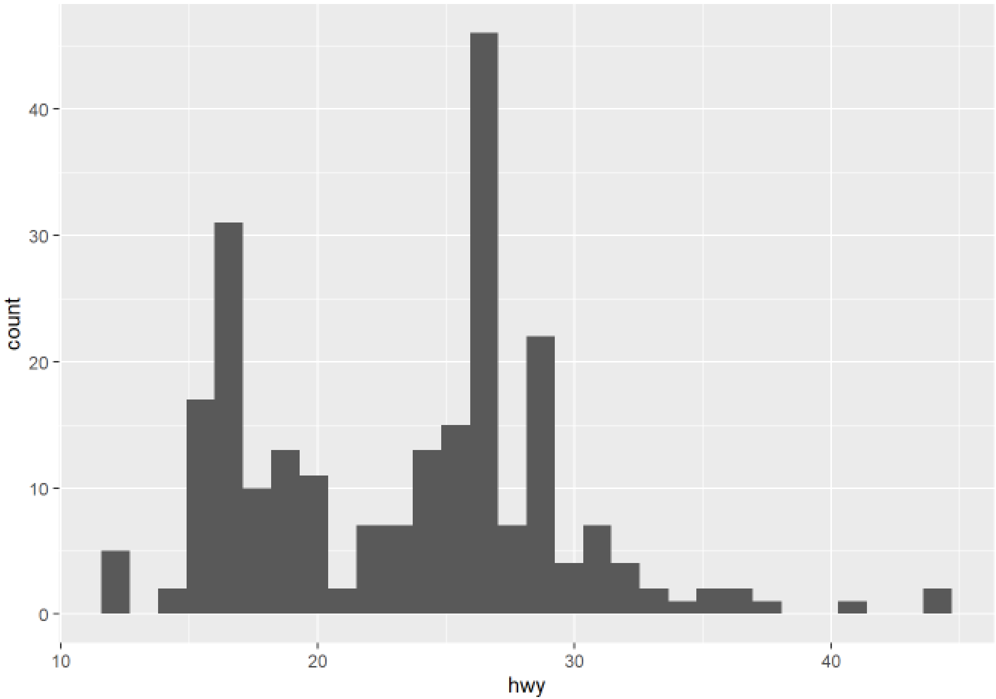
```
str(mpg.data)
```

```
## tibble [234 × 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

mpg dataset: histogram

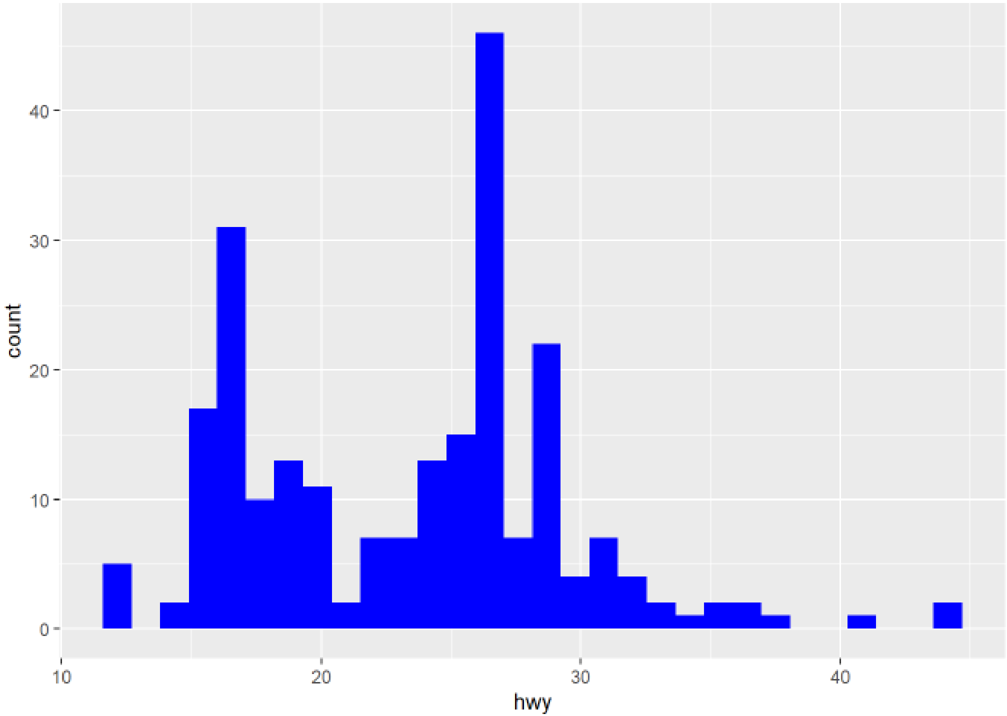
START WITH SIMPLE PLOT

```
ggplot(mpg.data, aes(x=hwy)) + geom_histogram()
```



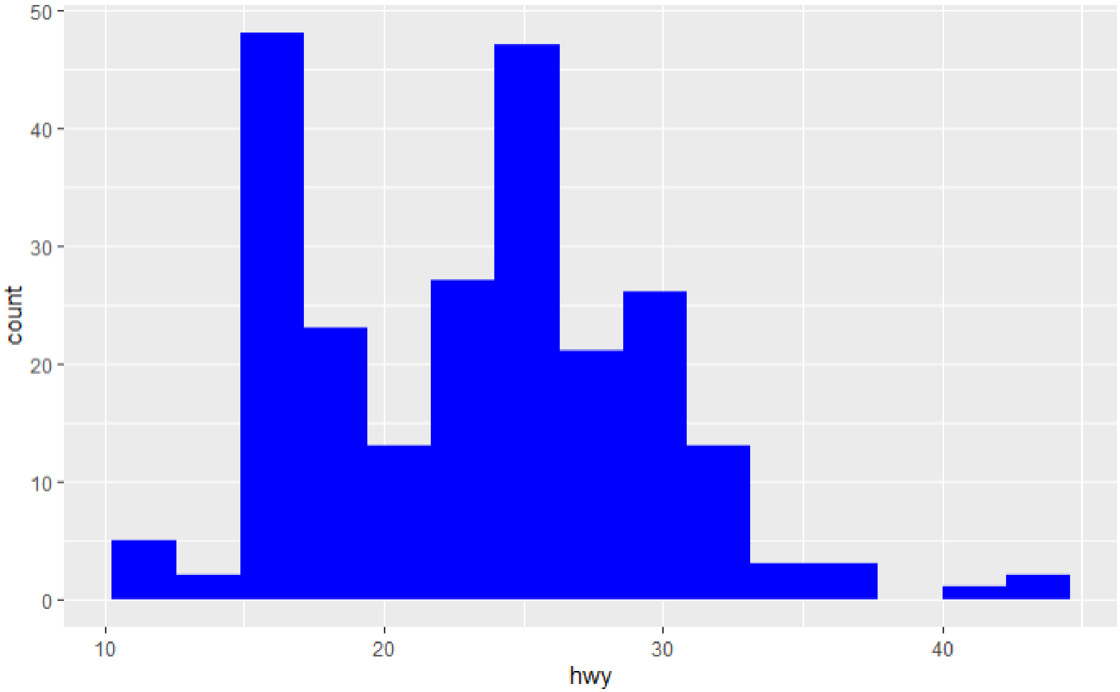
ADD COLOR TO FILL

```
ggplot(mpg.data, aes(x=hwy)) + geom_histogram(fill="blue")
```



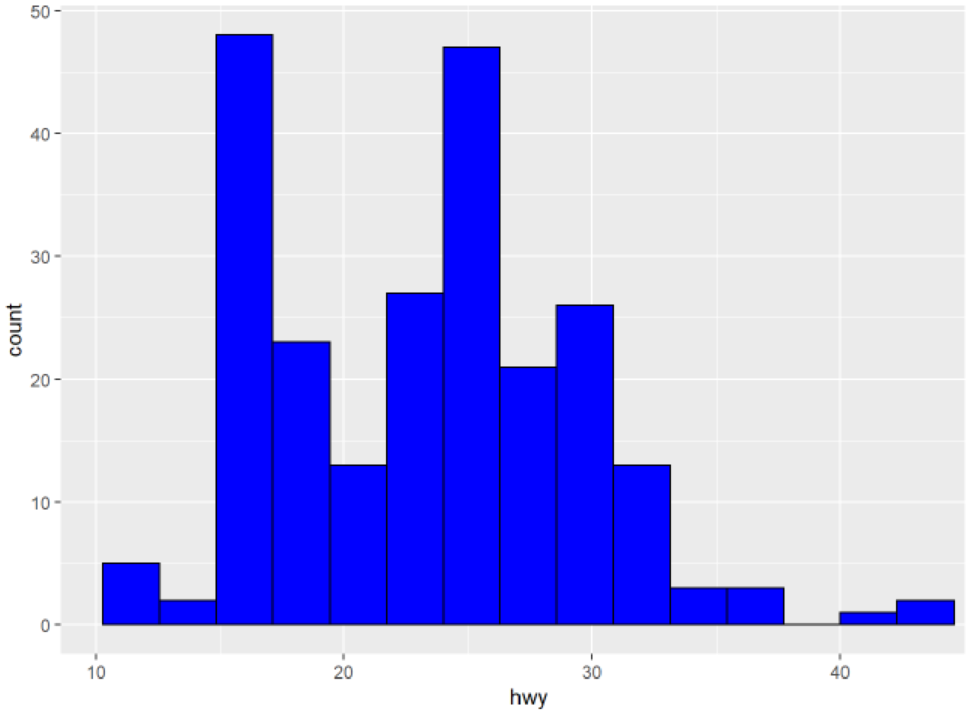
CHANGING BIN SIZE

```
ggplot(mpg.data, aes(x=hwy)) + geom_histogram(fill="blue", bins=15)
```



ADD BIN LAYOUT

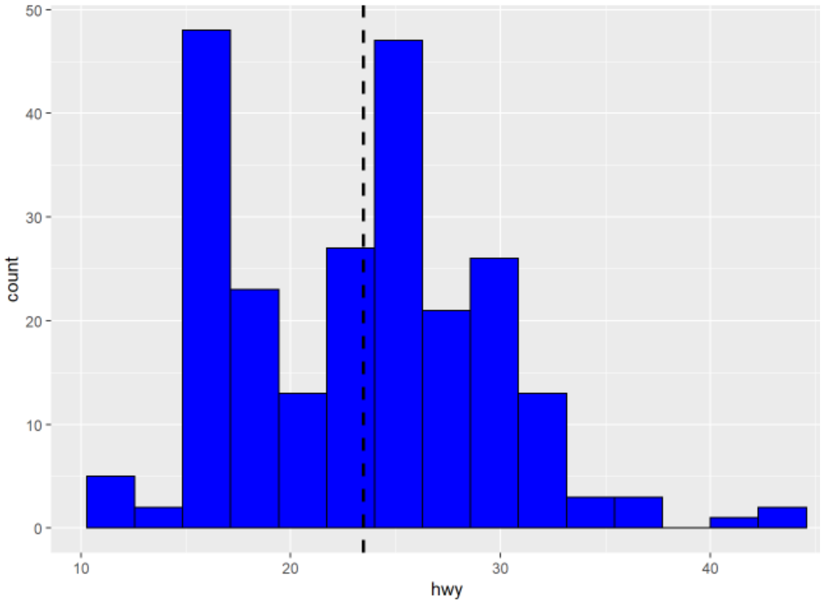
```
ggplot(mpg.data, aes(x=hwy)) + geom_histogram(fill="blue", col="black", bins=15)
```



ADDING MEAN VALUE TO THE PLOT

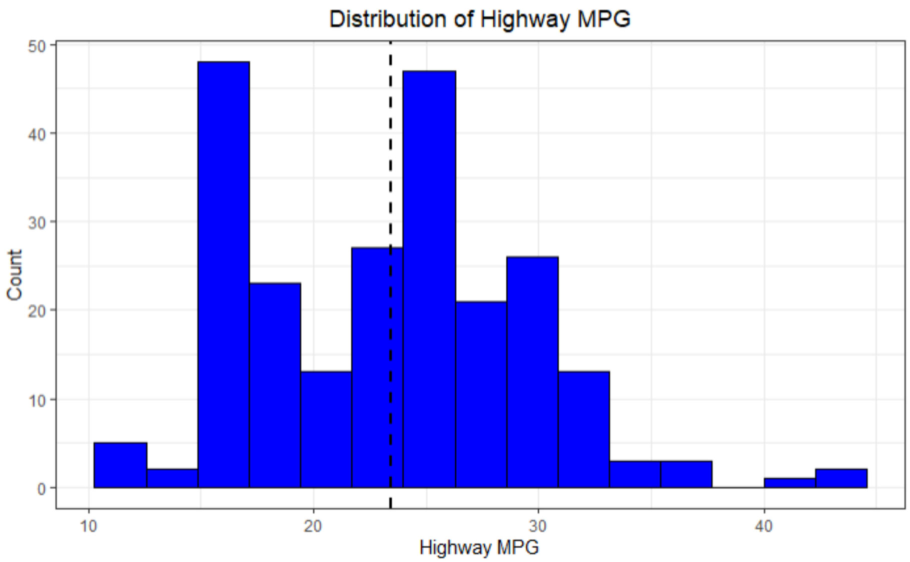
```
Mean.hwy <- mean(mpg.data$hwy, na.rm=TRUE)
```

```
ggplot(mpg.data, aes(x=hwy)) + geom_histogram(fill="blue", col="black", bins=15) +
  geom_vline(aes(xintercept = Mean.hwy), color="black", linetype="dashed", size=1)
```



ADD TITLE & LABELS

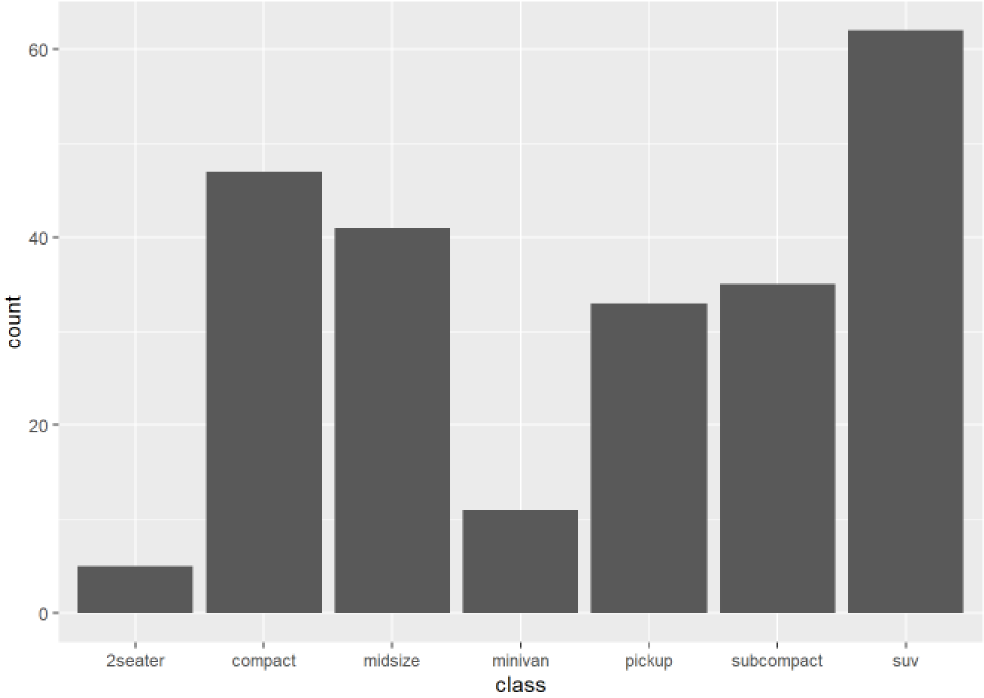
```
# hjust=0.5 indicates at the center; 1=right, 0=left

ggplot(mpg.data, aes(x=hwy)) + geom_histogram(fill="blue", col="black", bins=15) +  geom_vline(aes(xintercept = M
ean.hwy), color="black", linetype="dashed", size=1) + ggtitle("Distribution of Highway MPG") + theme_bw() + theme
(plot.title=element_text(hjust=0.5)) +
  xlab("Highway MPG") + ylab("Count")
```



Bar graph

START WITH SIMPLE GRAPH

```
ggplot(mpg.data, aes(class)) + geom_bar()
```

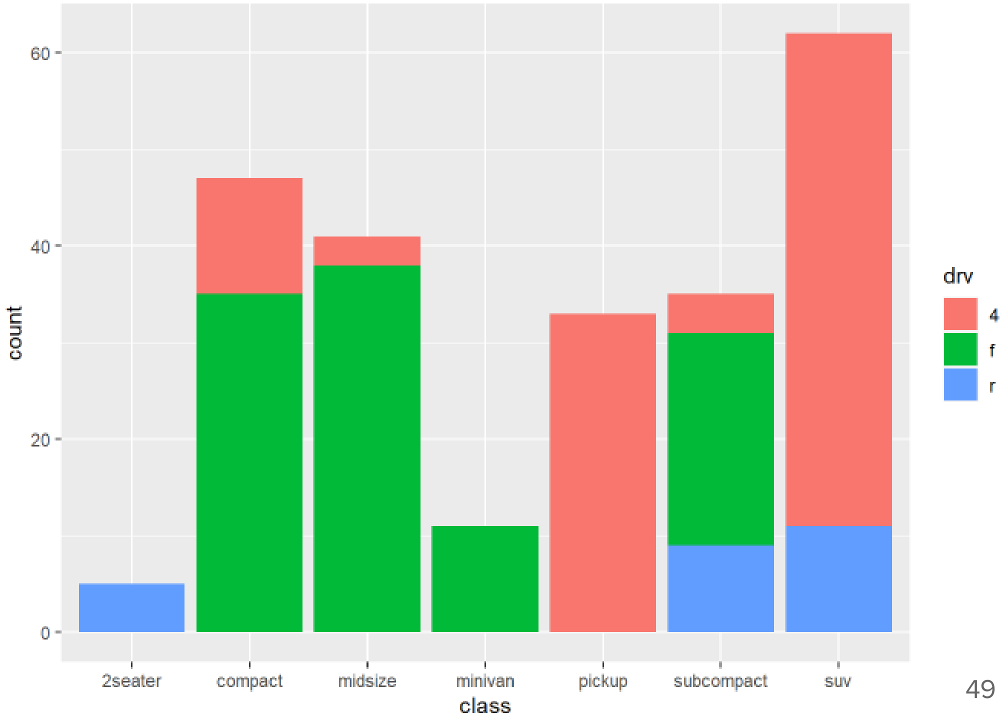


ADD CATEGORICAL VARIABLE

```
class(mpg.data$drv)
```

```
## [1] "character"
```

```
ggplot(mpg.data, aes(class, fill=drv)) + geom_bar()
```

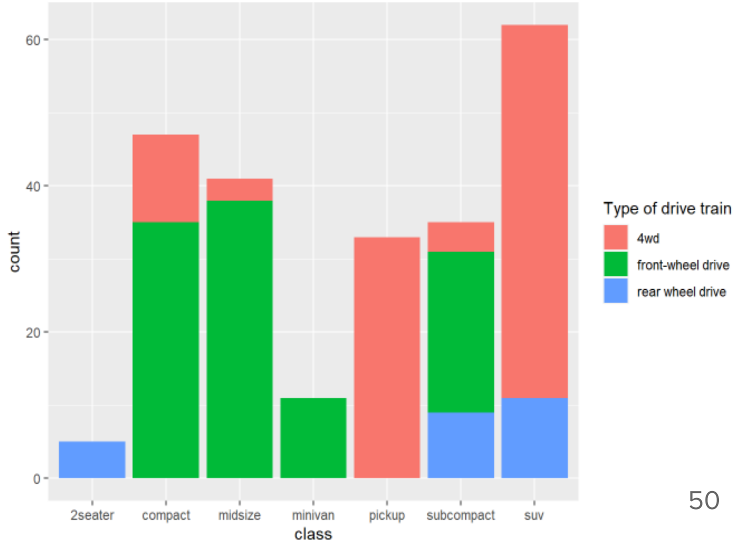


CHANGE LEGEND TO MAKE IT LEGIBLE

```
class(mpg.data$drv)
```

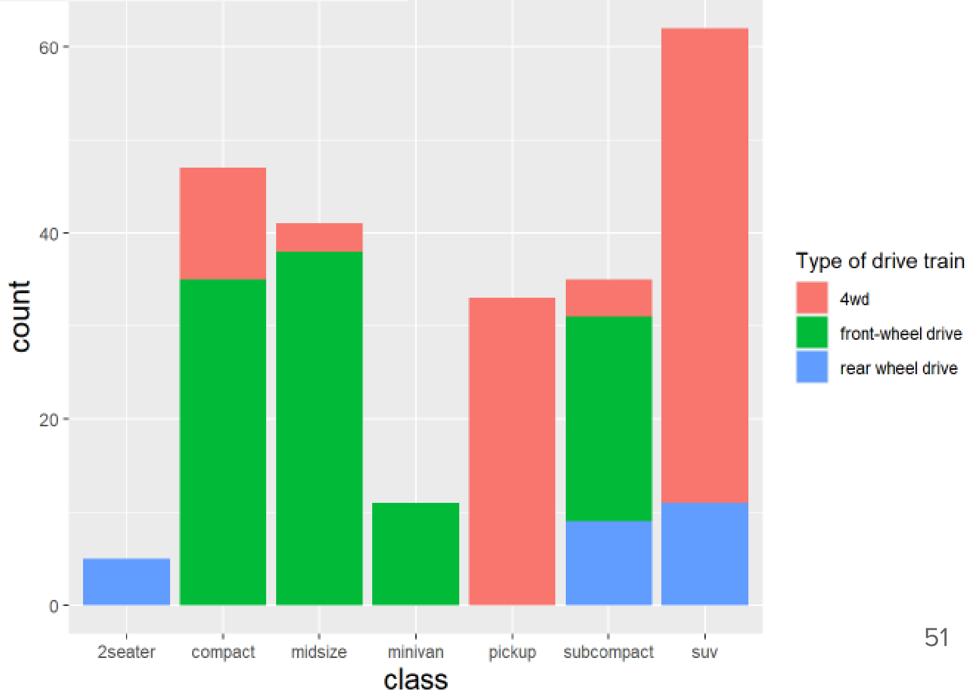
```
## [1] "character"
```

```
ggplot(mpg.data, aes(class, fill=drv)) + geom_bar() +
scale_fill_discrete(labels=c('4wd', 'front-wheel drive', 'rear wheel drive')) +
labs(fill="Type of drive train")
```



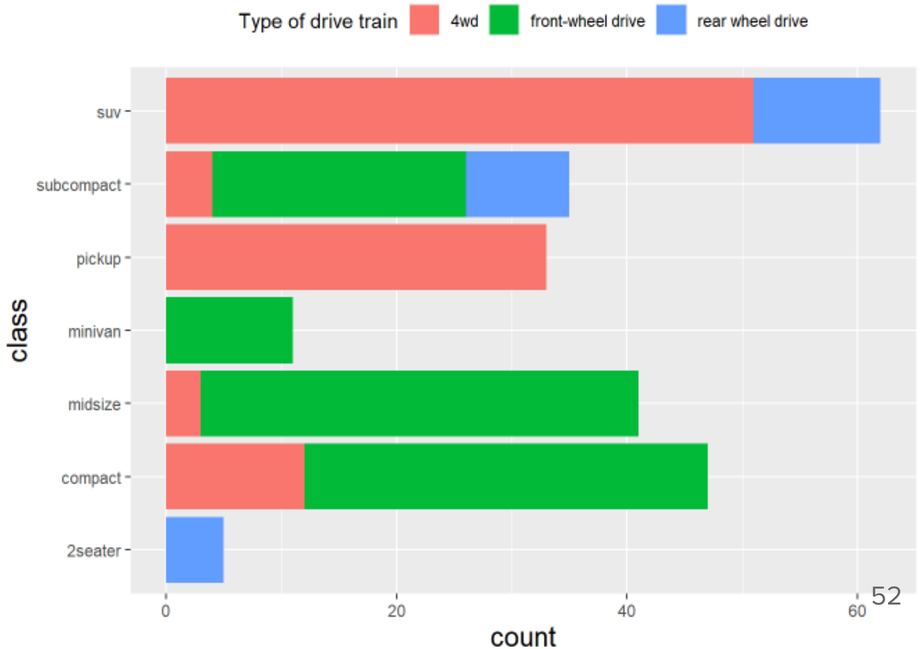
CHANGE AXIS TITLE SIZE

```
ggplot(mpg.data, aes(class, fill=drv)) + geom_bar() +
scale_fill_discrete(labels=c('4wd', 'front-wheel drive', 'rear wheel drive')) +
labs(fill="Type of drive train") +
theme(axis.title.x=element_text(size=15), axis.title.y=element_text(size=15))
```



FLIP THE AXIS

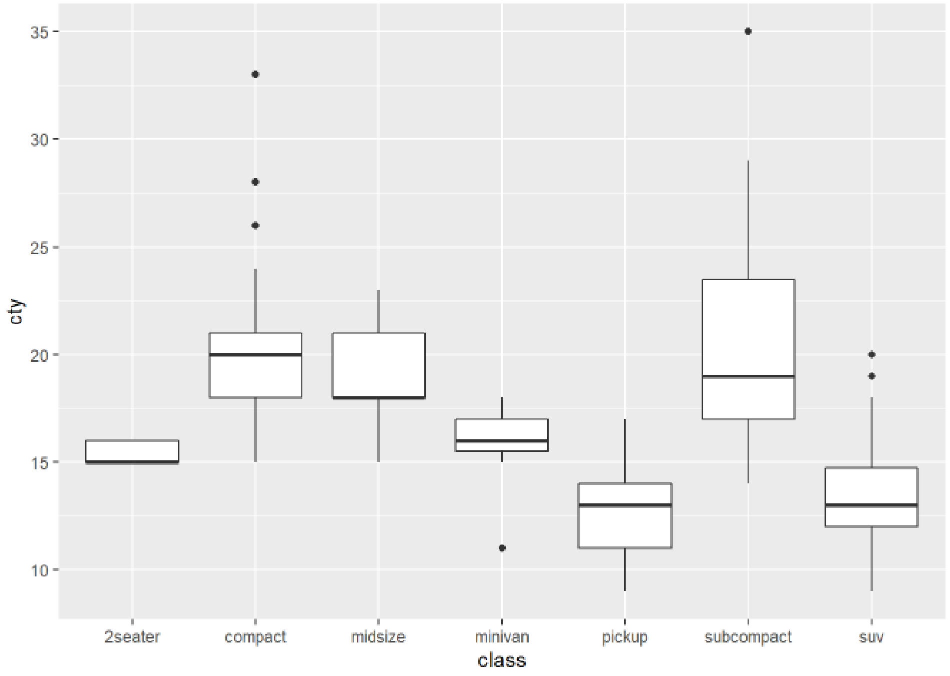
```
ggplot(mpg.data, aes(class, fill = drv)) +
geom_bar(position=position_stack(reverse = TRUE)) +
scale_fill_discrete(labels=c('4wd', 'front-wheel drive', 'rear wheel drive')) +
labs(fill="Type of drive train") +
theme(axis.title.x=element_text(size=15), axis.title.y=element_text(size=15)) +
coord_flip() +
theme(legend.position="top")
```



Box and whisker plots

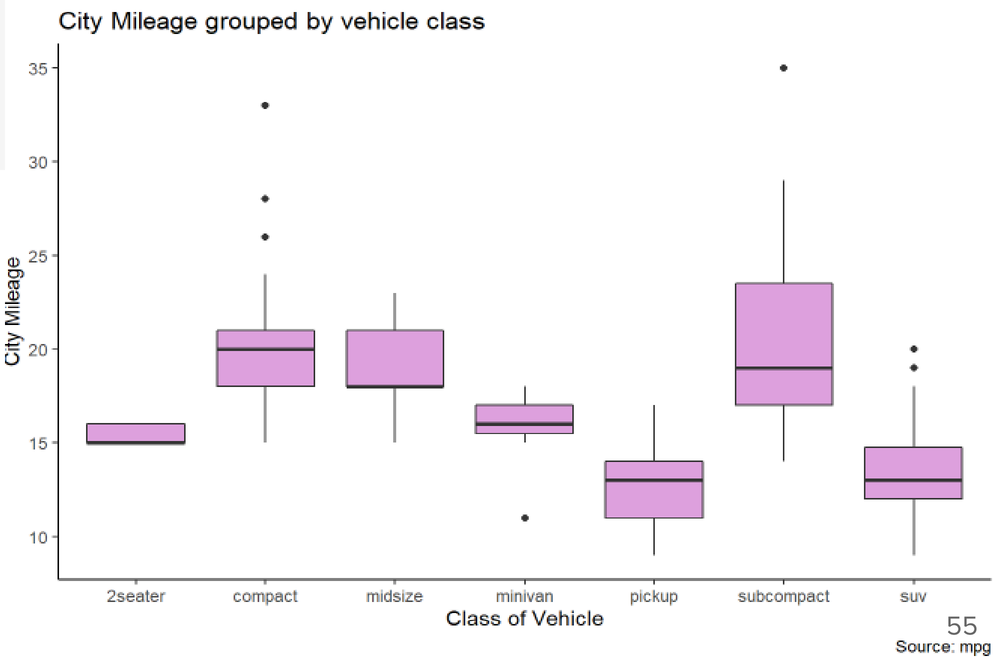
START WITH SIMPLE

```
ggplot(mpg.data, aes(x=class, y=cty)) + geom_boxplot()
```



ADDING MANY OTHER OPTIONS

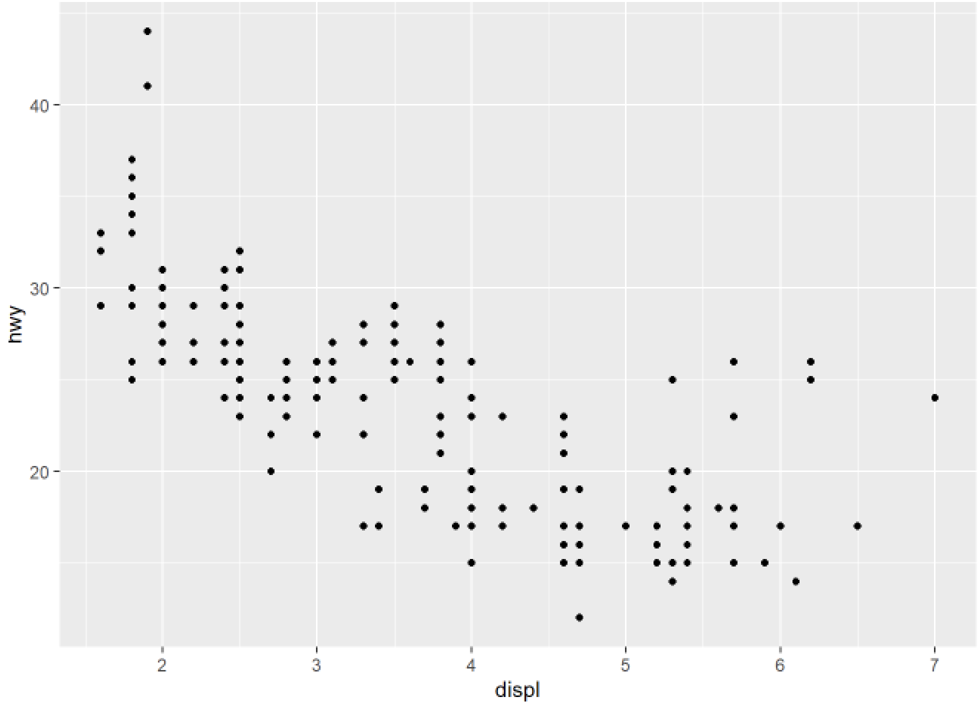
```
ggplot(mpg.data, aes(x=class, y=cty)) +
  geom_boxplot(fill="plum") +
  labs(title="City Mileage grouped by vehicle class",
         caption="Source: mpg",
         x="Class of Vehicle",
         y="City Mileage") +
  theme_classic()
```



Scatterplots

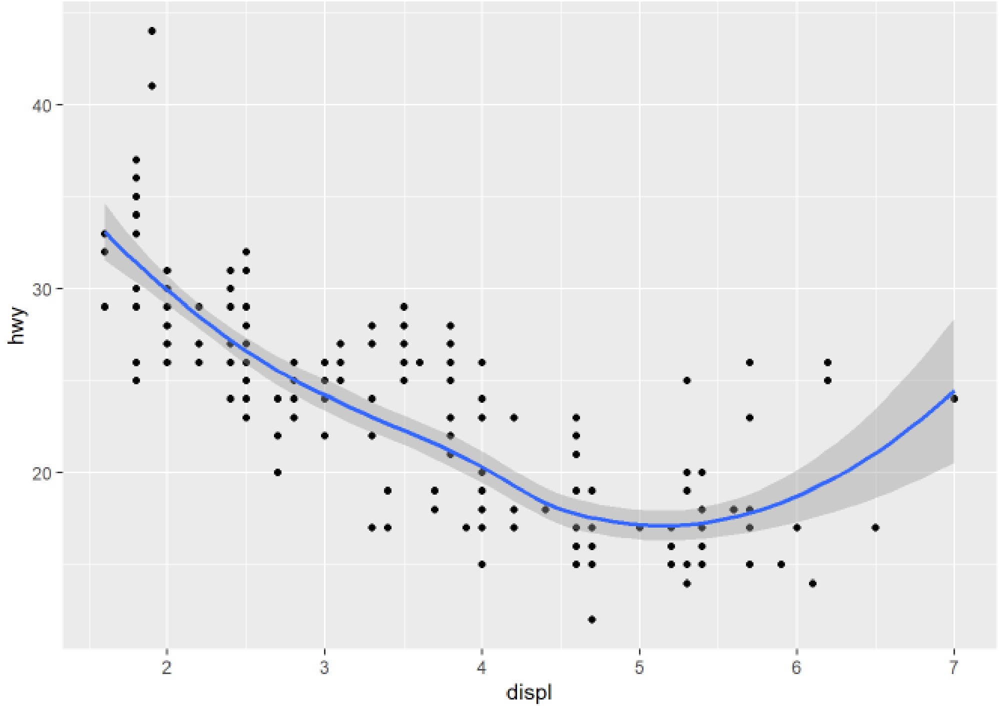
START WITH SIMPLE

```
ggplot(mpg.data, aes(x=displ, y=hwy)) + geom_point()
```

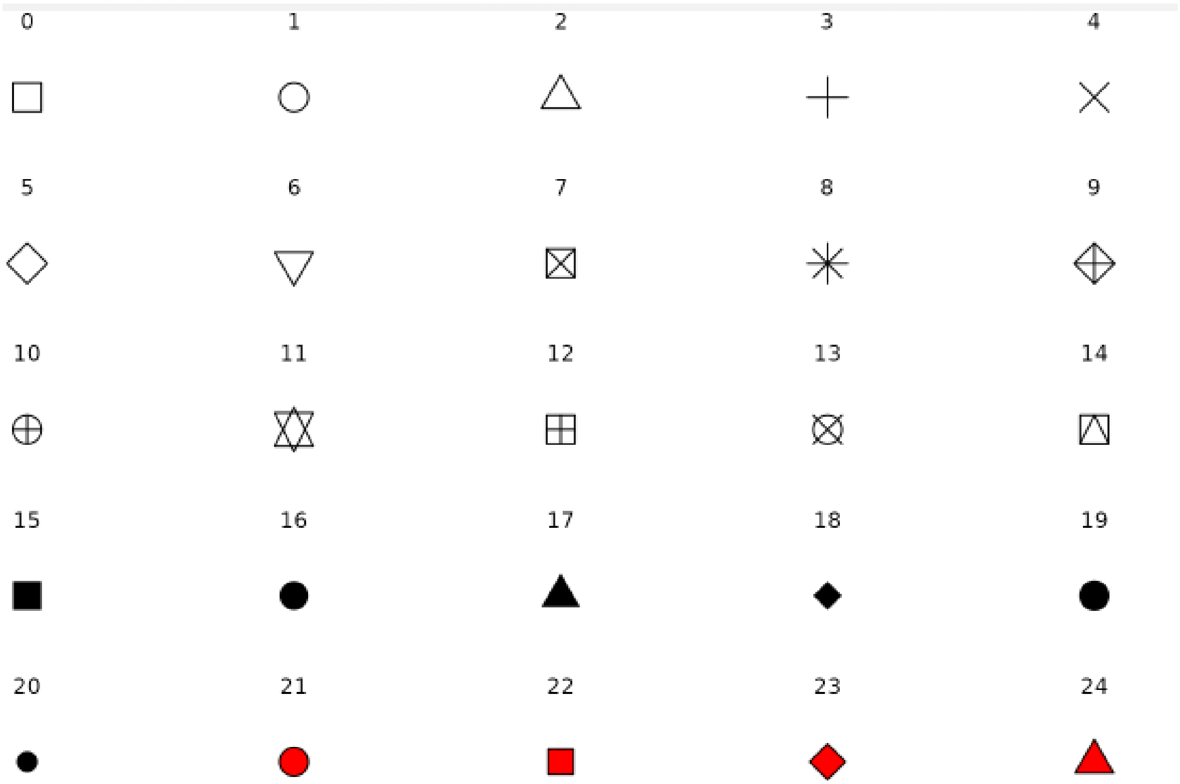


ADDING LINE OF BEST FIT

```
ggplot(mpg.data, aes(x=displ, y=hwy)) + geom_point() + geom_smooth()
```



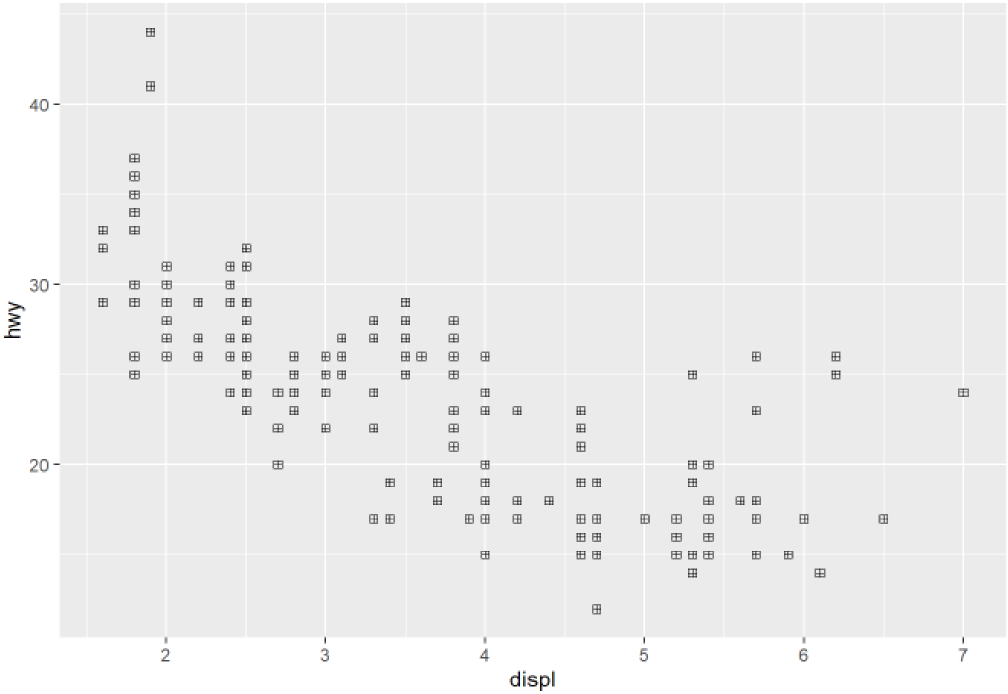
SYMBOL SHAPES



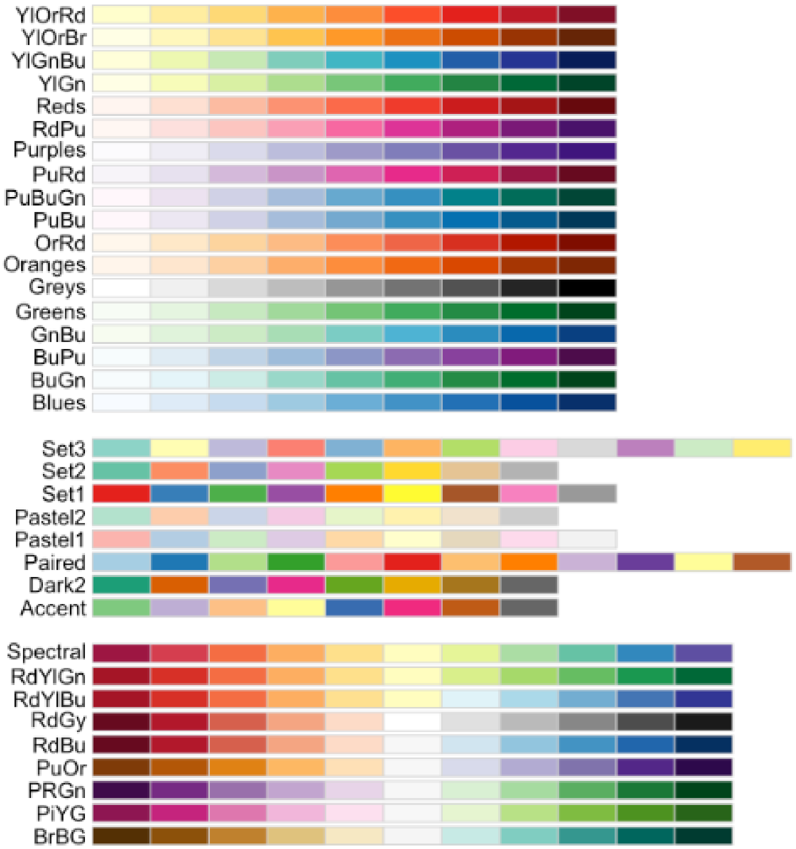
Source: tidyverse ggplot2

CHECK OUT SYMBOL SHAPES

```
ggplot(mpg.data, aes(x=displ, y=hwy)) + geom_point(shape=12)
```



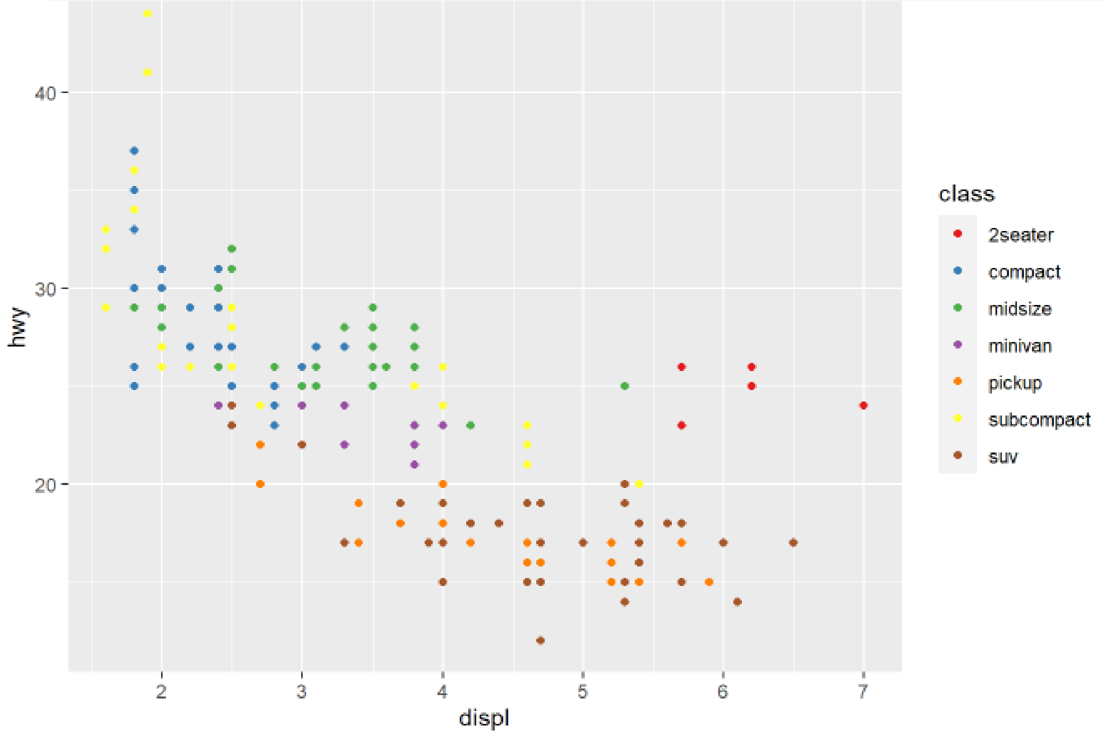
R COLOR BREWER'S PALETTES



[Source: R graph gallery]

CHANGE COLORS

```
ggplot(mpg.data, mapping=aes(x=displ, y=hwy, color=class)) +  geom_point() +
scale_color_brewer(palette="Set1")
```



ADD TITLE, SUBTITLE, & LABELS

```
ggplot(mpg.data, mapping=aes(x=displ, y=hwy, color=class)) +  geom_point() +
scale_color_brewer(palette="Set1") +
labs(x="Engine discplacement", y="Highway miles per gallon",
     title="Relationship between engine displacement and highway miles per gallon",
     subtitle="Data:mpg data",
     color="Type of car")
```