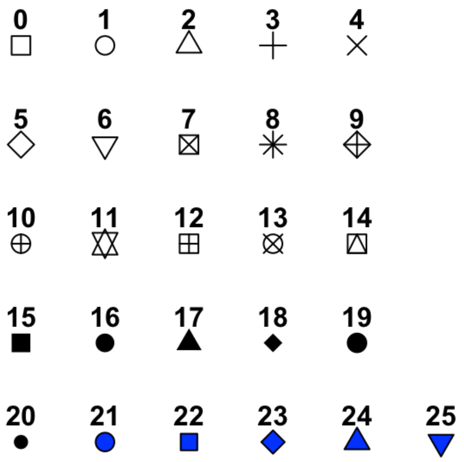
Day 2 : Advance R programming

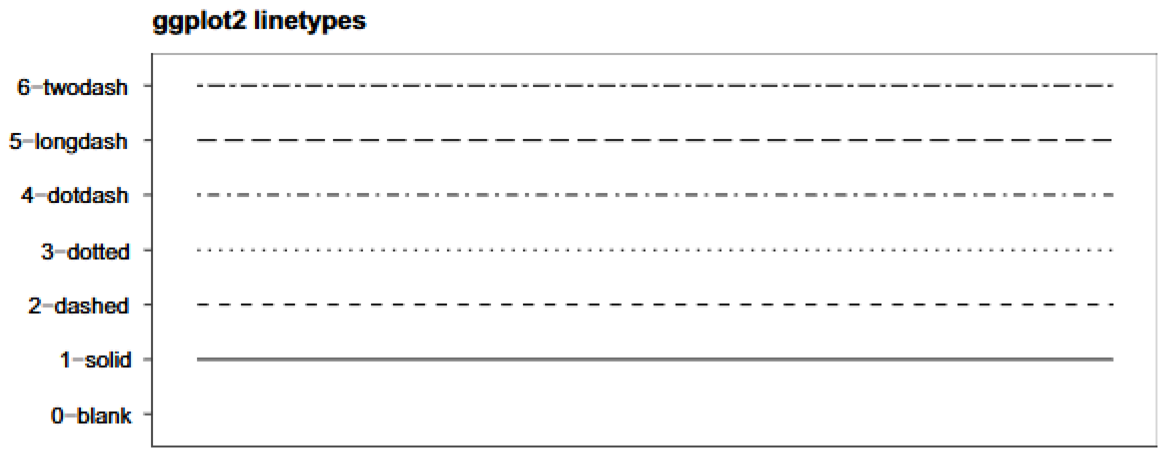
Session 5: Graphing

5.1. ggplot2 basics

shape

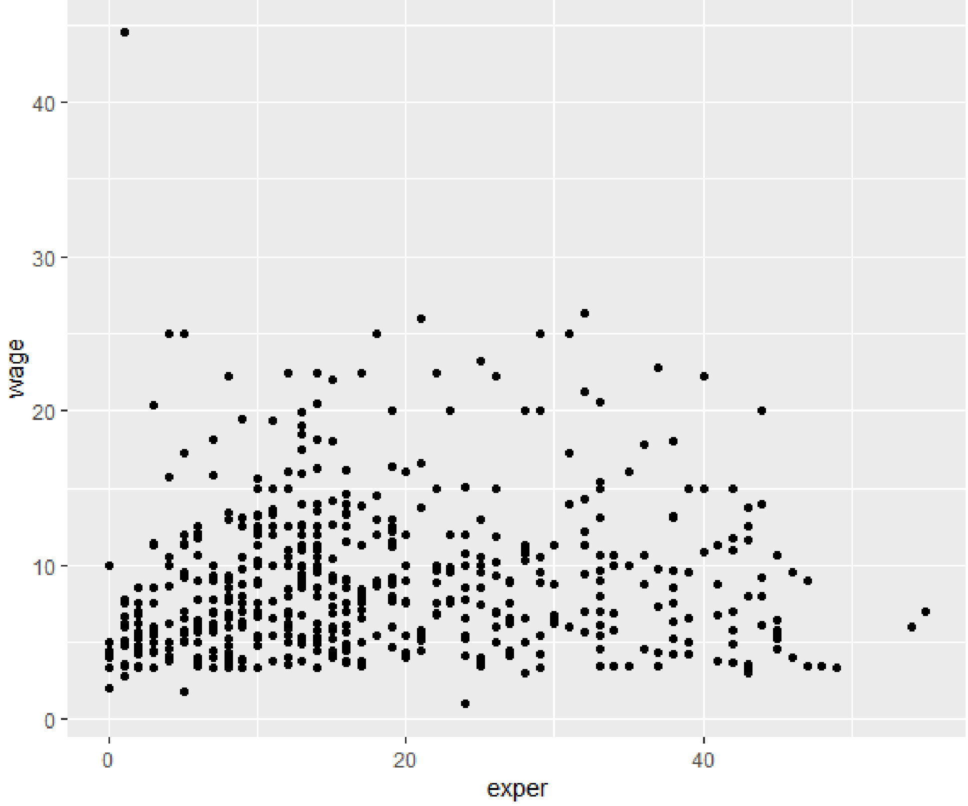


linetype

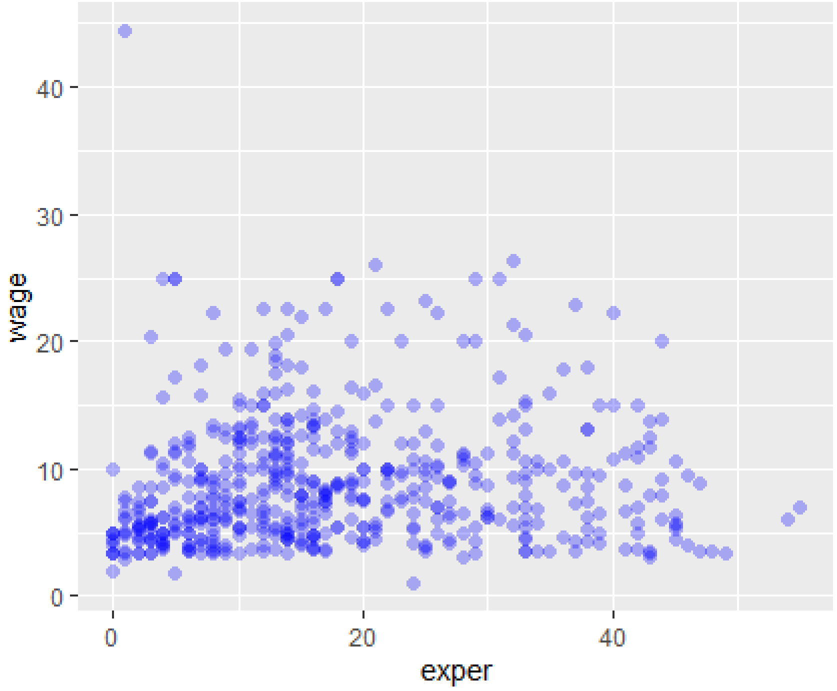


E018-ggplot2_basics.R

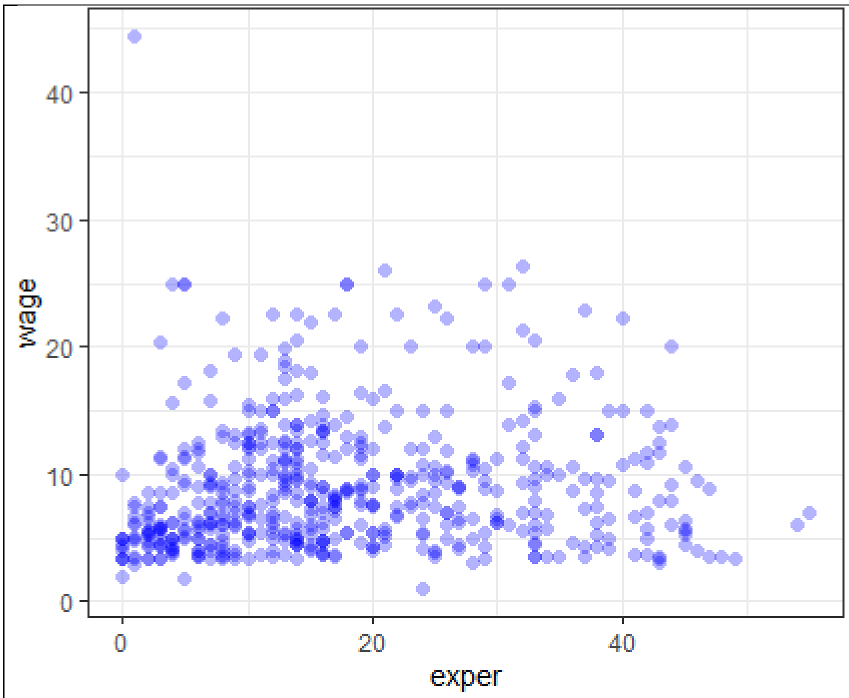
```
#---------------------------------------------------------------------------
# ggplot2 basics
#---------------------------------------------------------------------------
library(ggplot2)
library(mosaicData)

#simple scatter plot
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point()
```



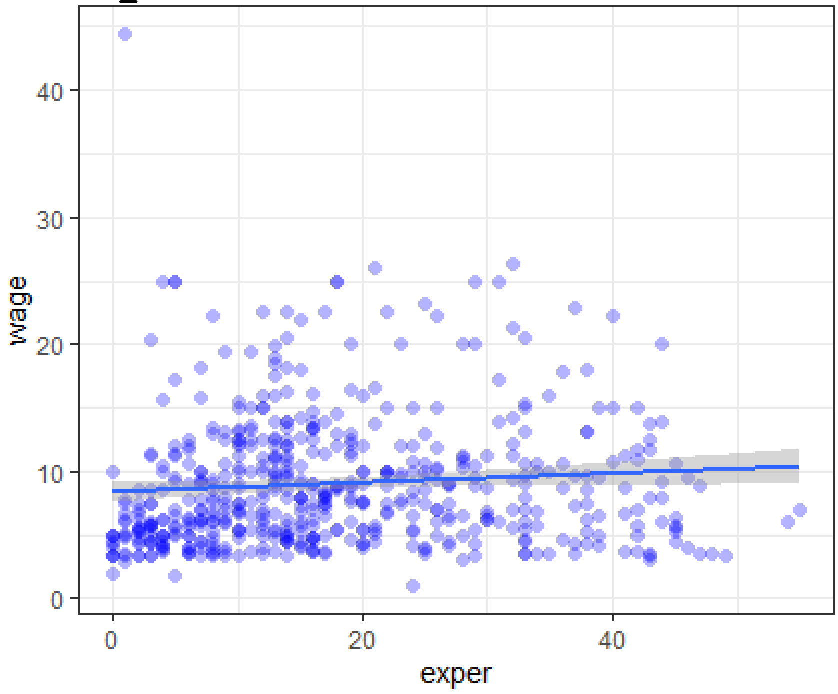

```
#scatter plot with various attributes
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(color = 'blue', shape = 16, alpha = 0.3, size = 2)
```



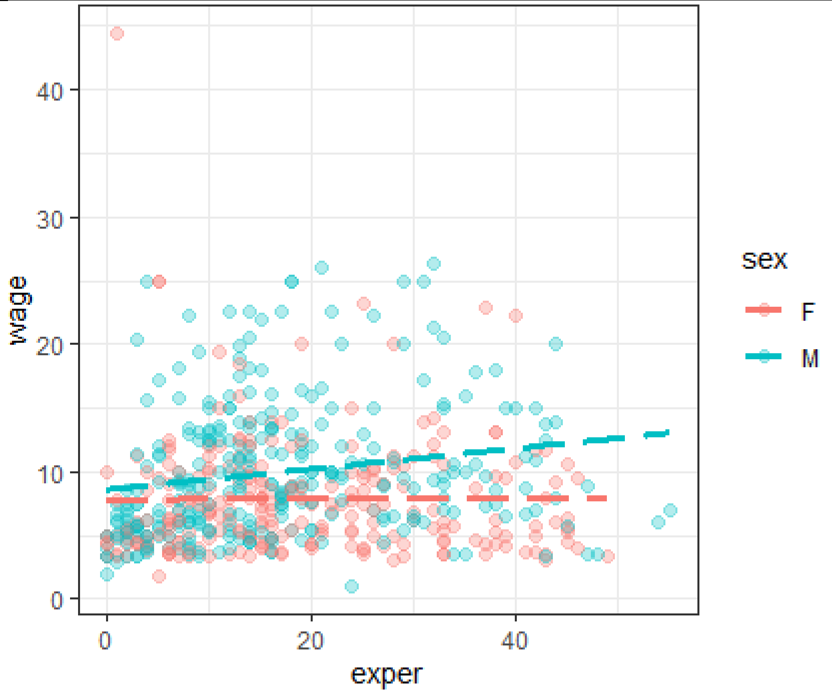

```
#applying ggplot2 themes
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(color = 'blue', shape = 16, alpha = 0.3, size = 2) +
  theme_bw()
```



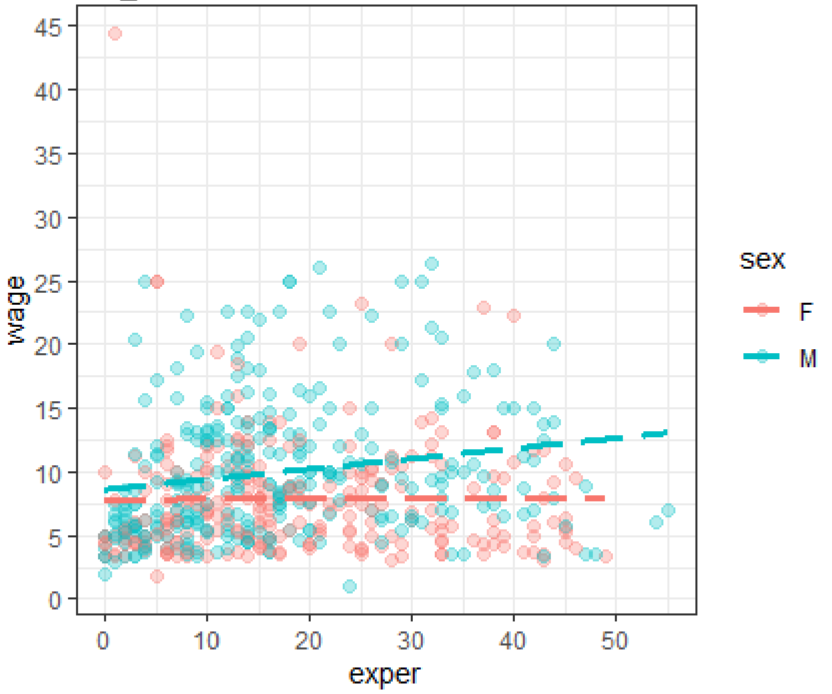

```
#scatter plot and best fit line
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(color = 'blue', shape = 16, alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm') +
  theme_bw()
```



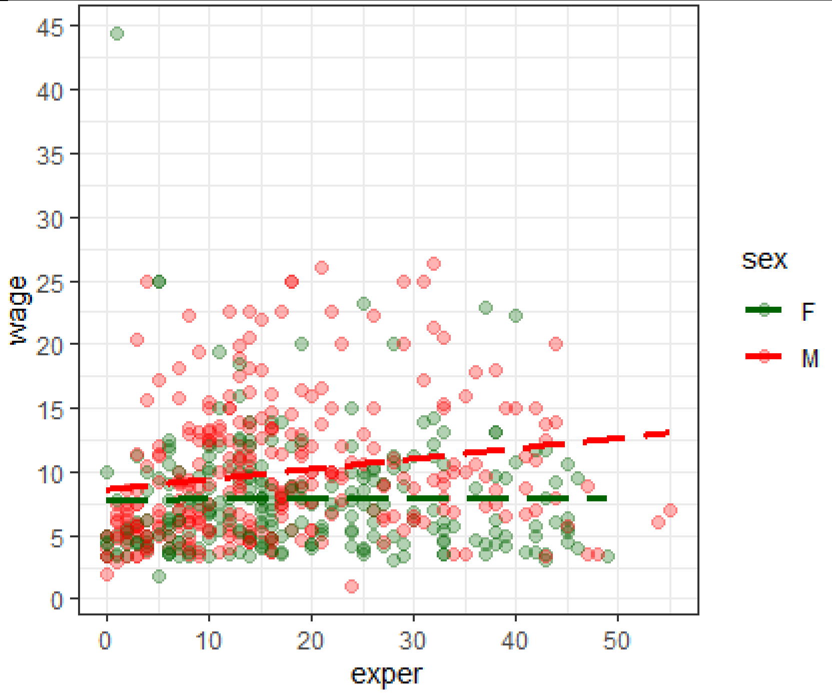

```
# grouping category variable attribute color
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  theme_bw()
```



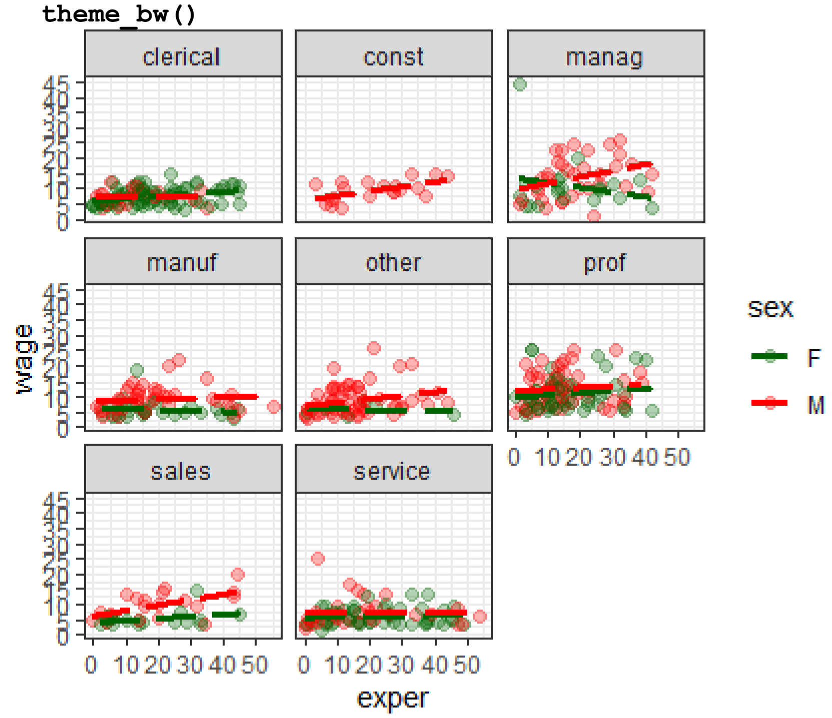

```
# x-axis and y-axis setting scales
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks = seq(0,60,5)) +
  theme_bw()
```



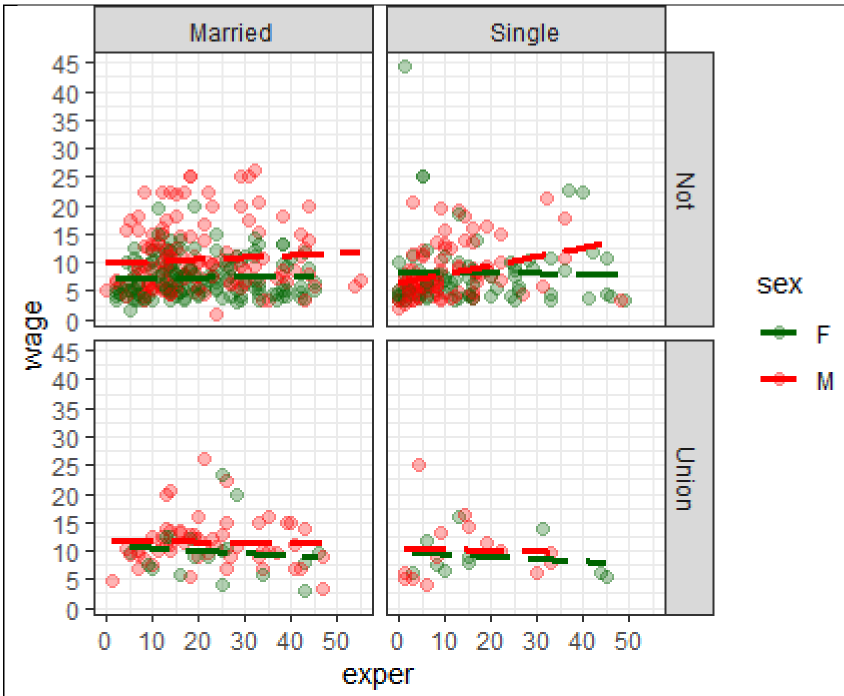

```
# color manual of grouped category variable
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks = seq(0,60,5)) +
  scale_color_manual(values = c('darkgreen','red')) +
  theme_bw()
```



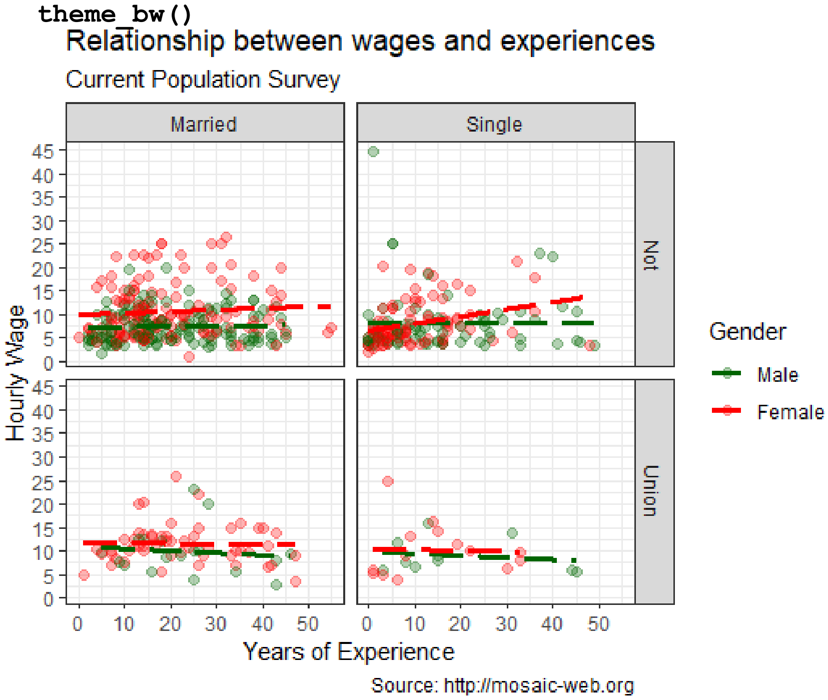

```
# subplots according to a categorical variable (facet_wrap)
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks = seq(0,60,5)) +
  scale_color_manual(values = c('darkgreen','red')) +
  facet_wrap(~sector) +
  theme_bw()
```




```
# subplots according to a categorical variable (facet_grid)
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks = seq(0,60,5)) +
  scale_color_manual(values = c('darkgreen','red')) +
  facet_grid(union ~ married) +
  theme_bw()
```



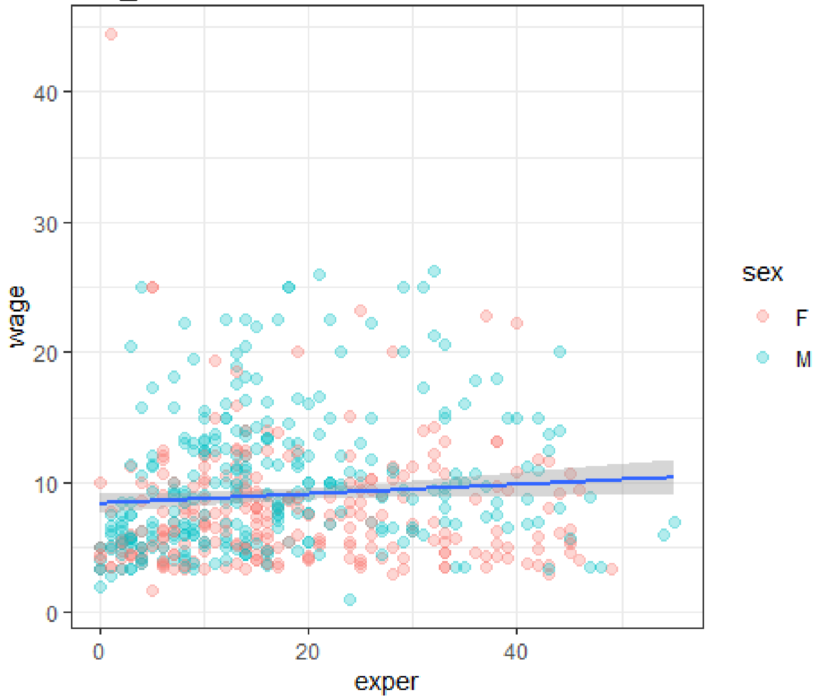

```
# labels
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks = seq(0,60,5)) +
  scale_color_manual(values = c('darkgreen','red'), labels = c('Male',
'Female')) +
  facet_grid(union ~ married) +
  labs(title = 'Relationship between wages and experiences',
       subtitle = 'Current Population Survey',
       caption = "Source: http://mosaic-web.org",
       x = "Years of Experience",
       y = "Hourly Wage",
       color = 'Gender') +
  theme_bw()
```



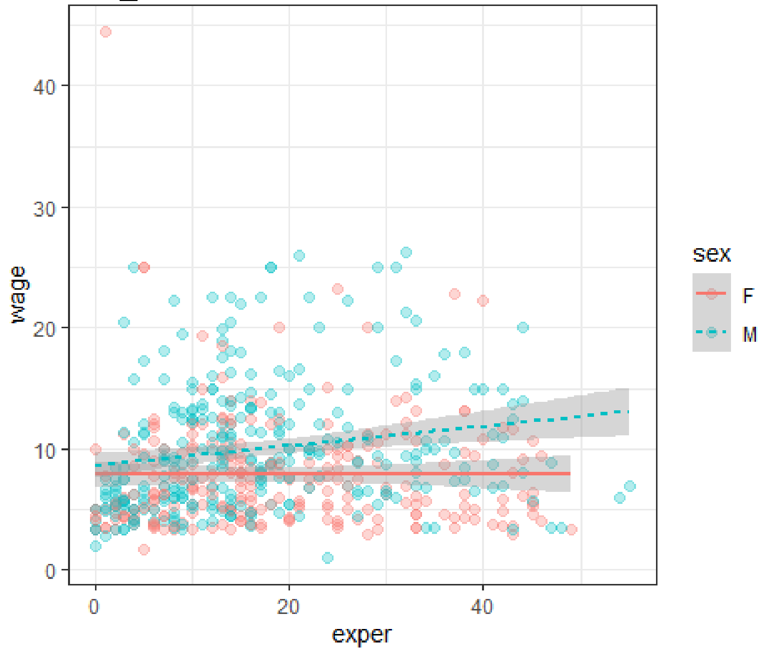
Mapping in individual geom functions, stroing the plot as an R object, and exporting plot

E019-ggplot2_further.R

```
#-----------------------------------------------------------------------------
# ggplot2 further
#-----------------------------------------------------------------------------
library(ggplot2)
library(mosaicData)

# mapping color = sex in geom_point instead in ggplot function
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(mapping = aes(color = sex), alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm') +
  theme_bw()
```




```
#mapping aes in geom_point and geom_smooth
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(mapping = aes(color = sex), alpha = 0.3, size = 2) +
  geom_smooth(mapping = aes(linetype = sex, color = sex), method = 'lm') +
  theme_bw()
```




```
#storing the plot in an object
myplot <- ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(mapping = aes(color = sex), alpha = 0.3, size = 2) +
```

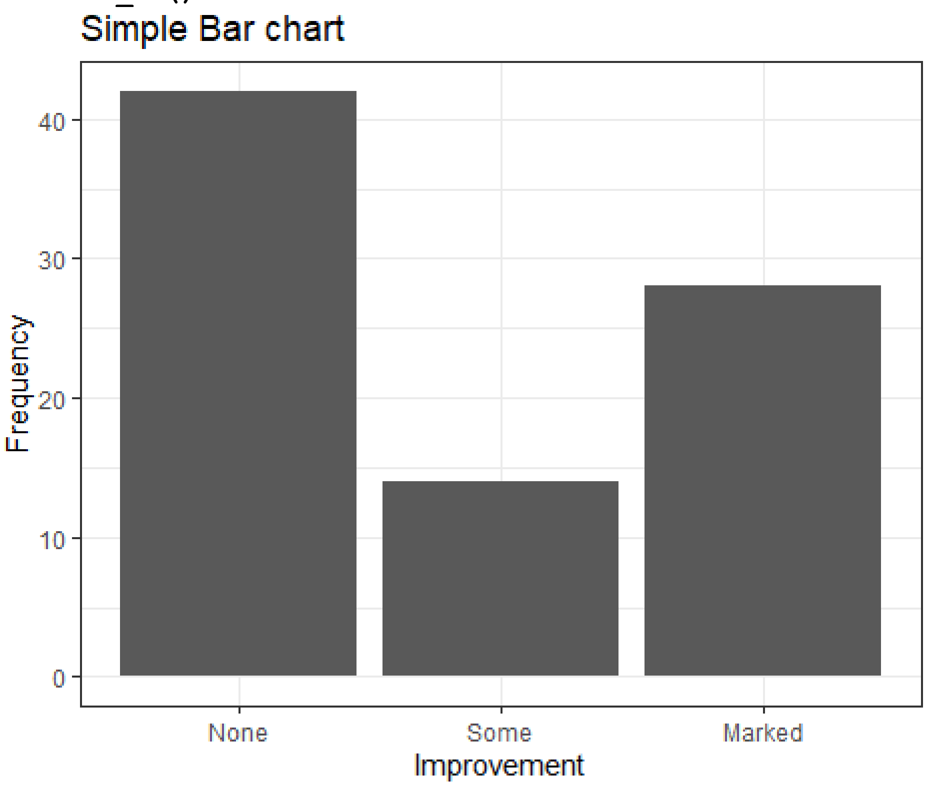
```
  geom_smooth(mapping = aes(linetype = sex, color = sex), method = 'lm') +
  theme_bw()
myplot

#Saving the plot
ggsave(filename = 'myplot.jpg')
ggsave(filename = 'myplot.pdf')
ggsave(filename = 'myplot.png')

ggsave(filename = 'myplot.jpg', plot = myplot, units = 'cm', width = 20,
height = 16)
ggsave(filename = 'myplot.pdf', plot = myplot, units = 'cm', width = 20,
height = 16)
ggsave(filename = 'myplot.png', plot = myplot, units = 'cm', width = 20,
height = 16)
```

5.2. Various types of plots

Bar charts

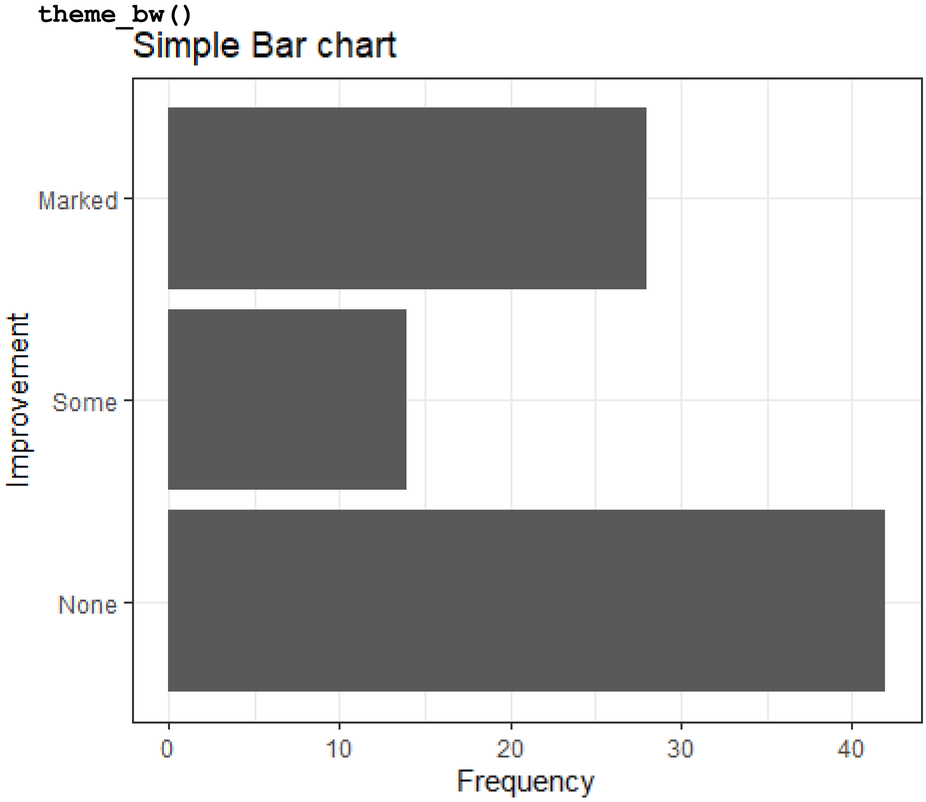
E020-bar_chart.R

```
library(ggplot2)
data(Arthritis, package="vcd")

#simple bar chart
table(Arthritis$Improved)
 None   Some Marked
   42     14     28

ggplot(Arthritis, aes(x=Improved)) + geom_bar() +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Frequency") +
  theme_bw()
```

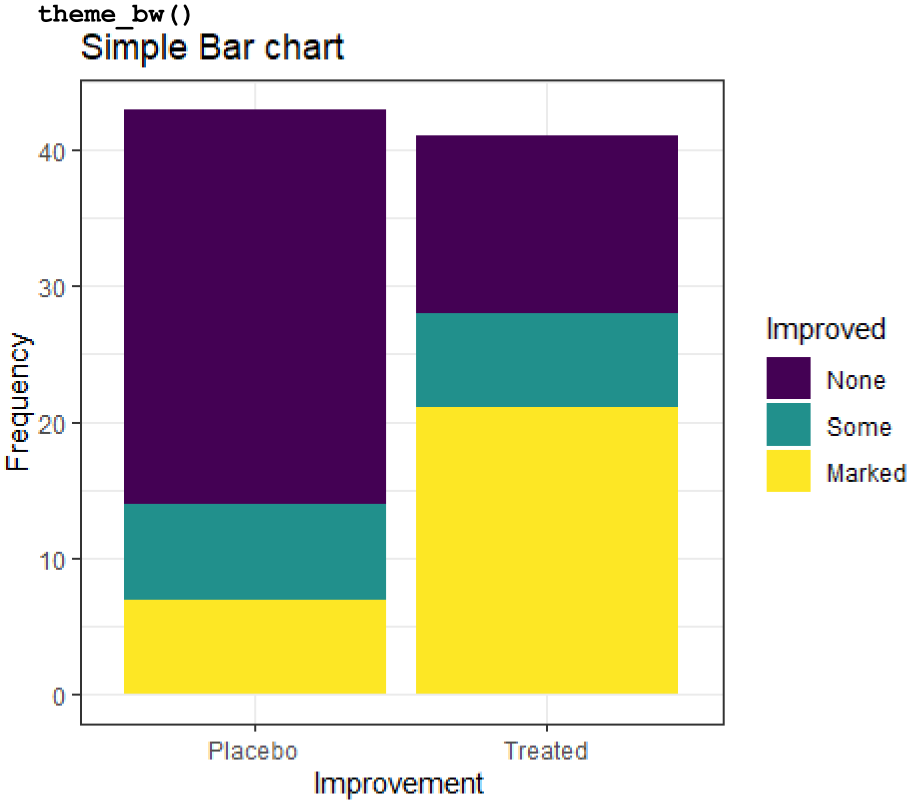



```
#Horizontal bar chart
ggplot(Arthritis, aes(x=Improved)) + geom_bar() +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Frequency") +
```

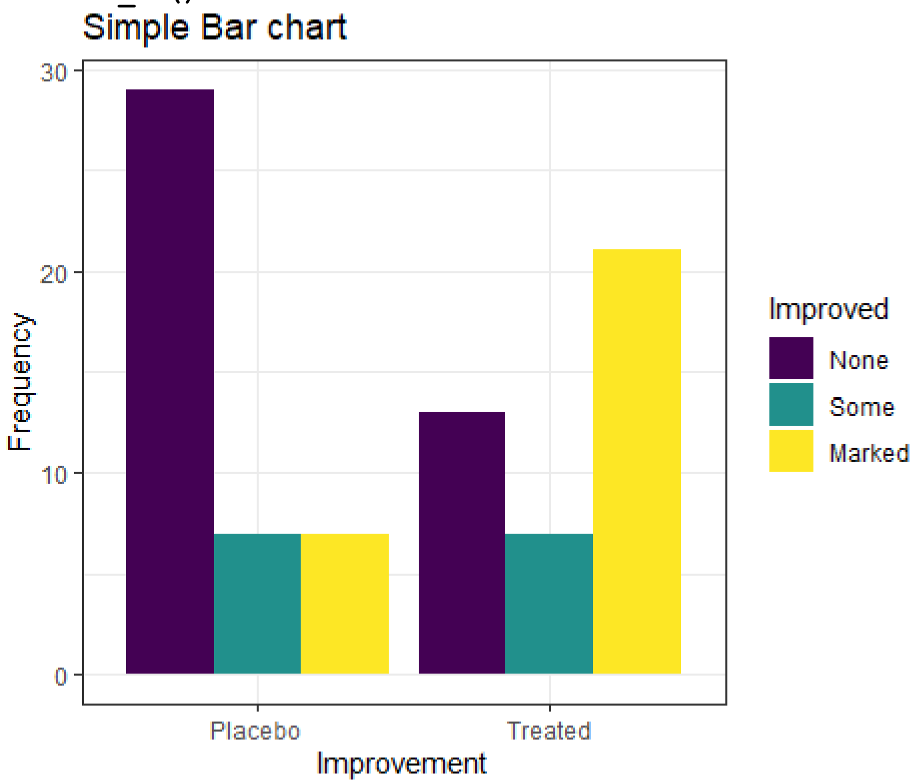
```
  coord_flip()+
  theme_bw()
```



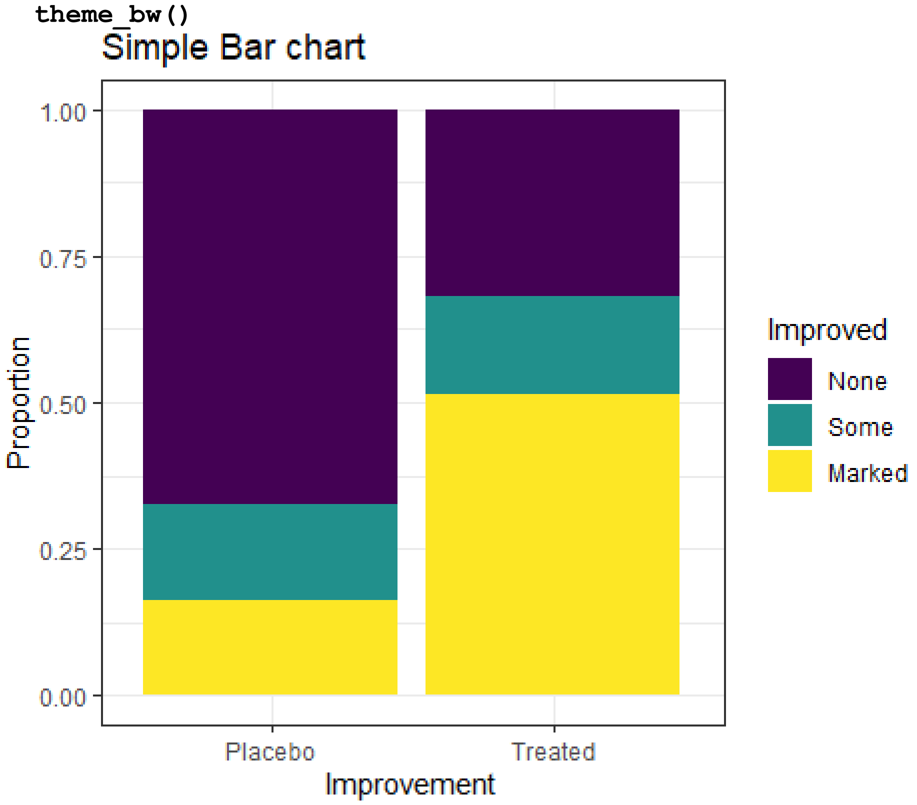

```
#Stacked bar chart
table(Arthritis$Improved, Arthritis$Treatment)
        Placebo Treated
 None        29      13
 Some         7       7
 Marked       7      21

ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +
  geom_bar(position = 'stack') +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Frequency") +
  theme_bw()
```



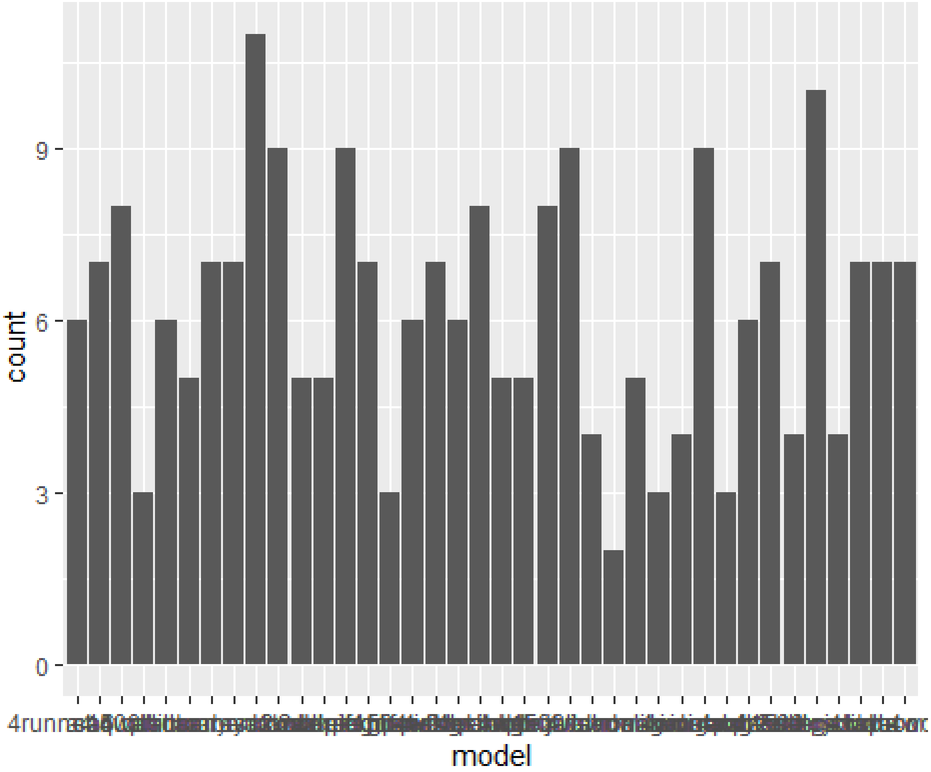
```
#Grouped bar chart
ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +
  geom_bar(position = 'dodge') +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Frequency") +
  theme_bw()
```



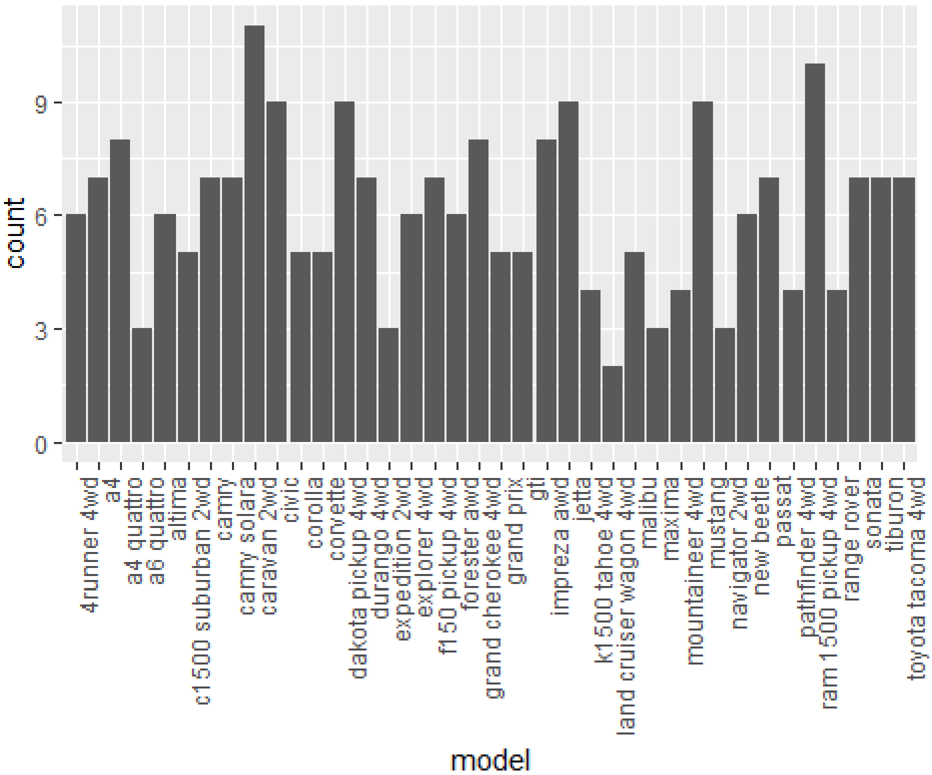

```
#Filled bar chart
ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +
  geom_bar(position = 'fill') +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Proportion") +
  theme_bw()
```



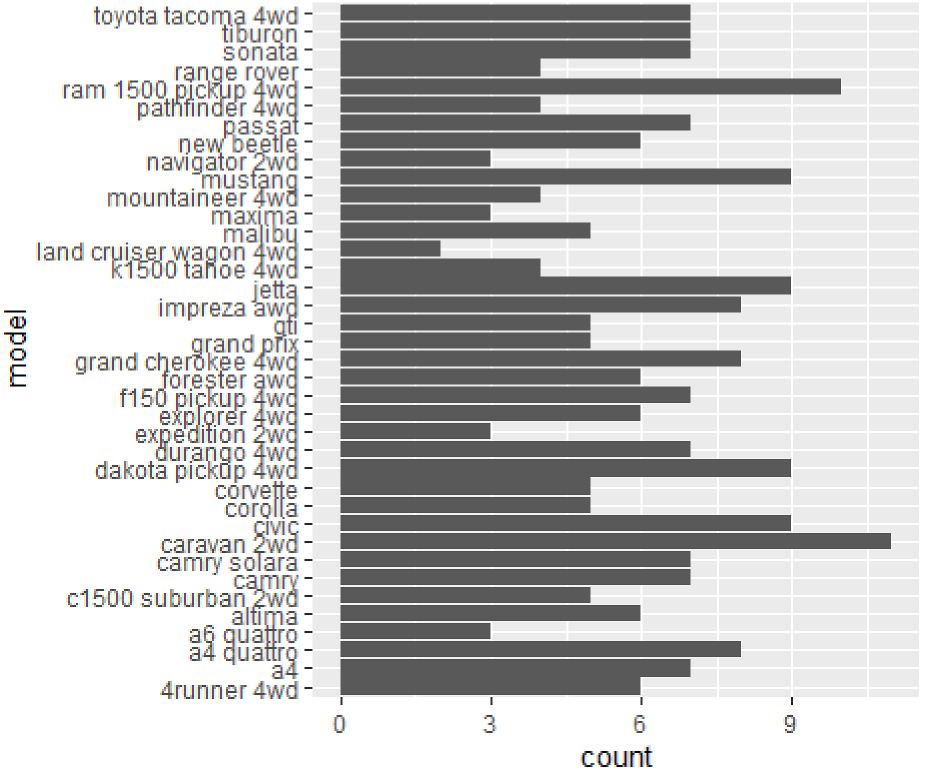
```
#managing congested labels
ggplot(mpg, aes(x=model)) +
  geom_bar() #produce congested labels
```




```
ggplot(mpg, aes(x=model)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



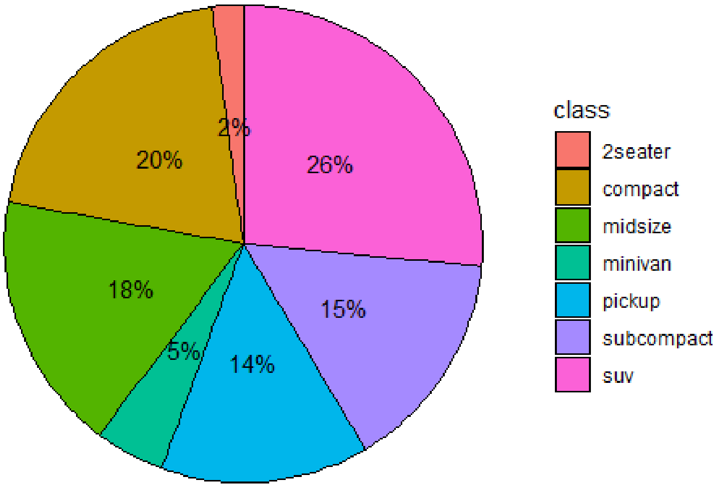

```
#OR
ggplot(mpg, aes(x=model)) +
  geom_bar() +
  coord_flip()
```



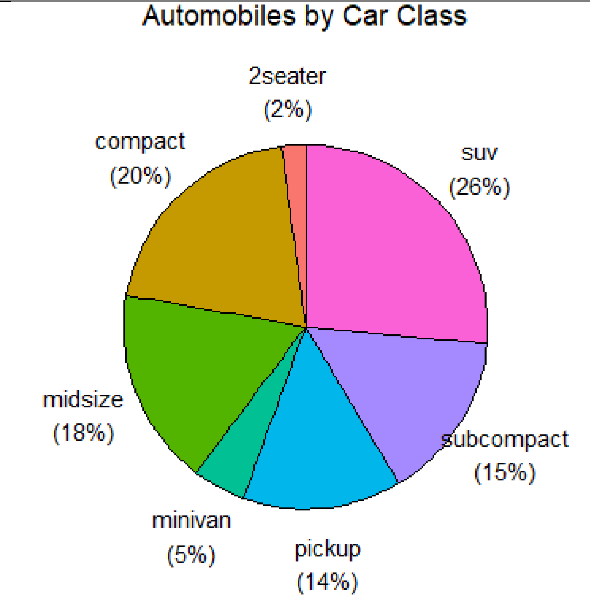
Pie charts

E021-pie_chart.R

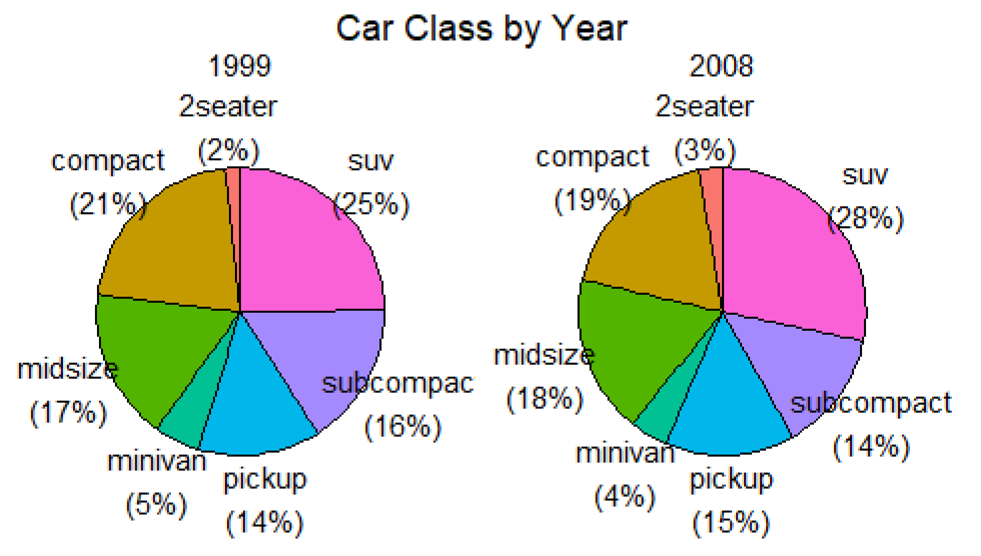
```r
if(!require(remotes)) install.packages("remotes")
remotes::install_github("rkabacoff/ggpie")

library(ggplot2)
library(ggpie)

#simple pie chart
ggpie(mpg, class)
```




```r
# no legend and offset of labels from the pie chart
ggpie(mpg, class, legend=FALSE, offset=1.3,
      title="Automobiles by Car Class")
```



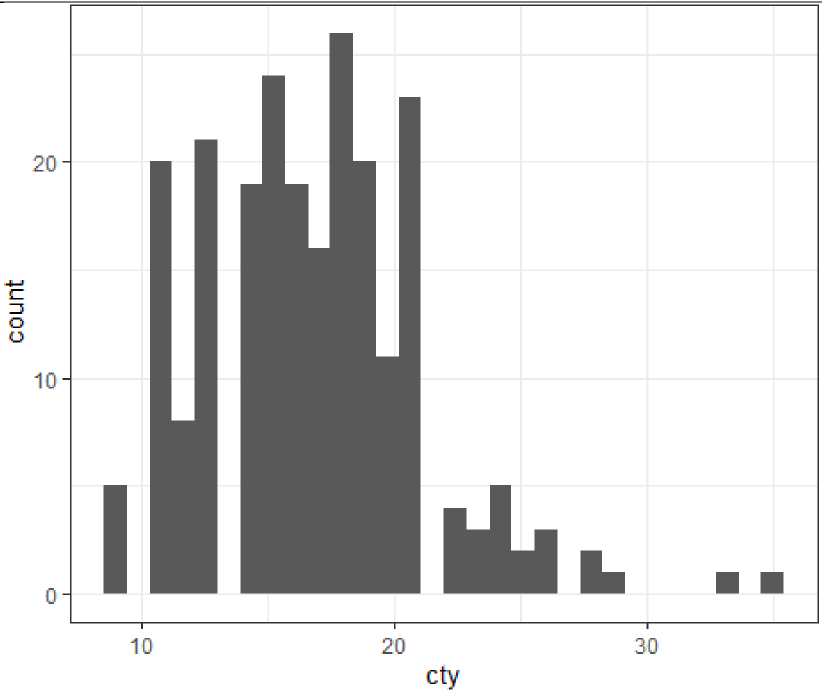

```
# group wise pie charts
ggpie(mpg, class, year,
      legend=FALSE, offset=1.3, title="Car Class by Year")
```



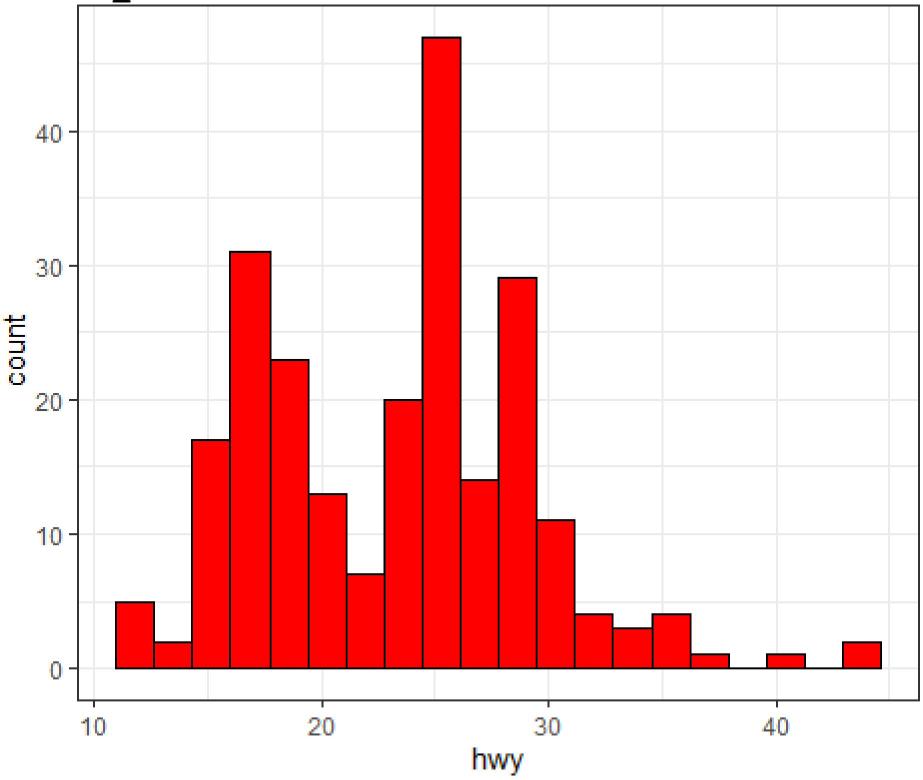
Histograms

E022-histogram.R

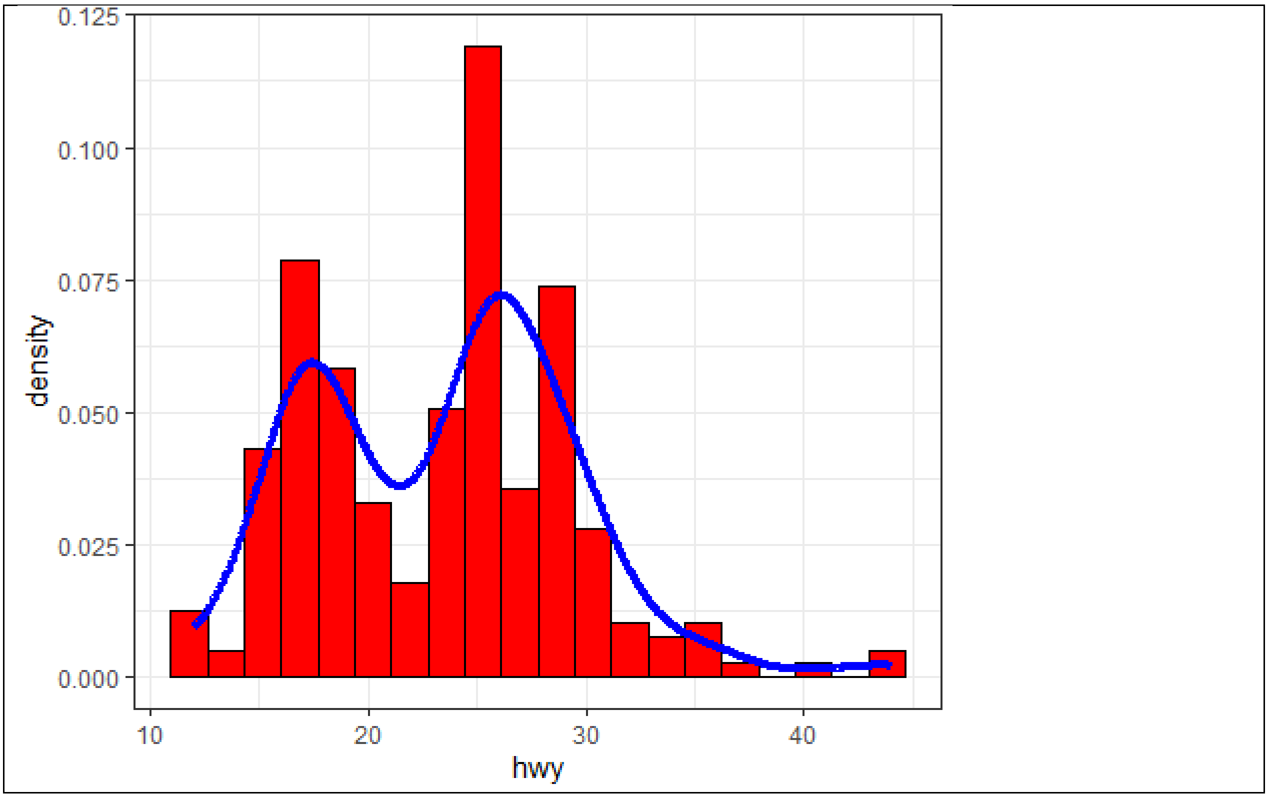
```
library(ggplot2)
library(dplyr)
data(mpg)

#Simple histogram
ggplot(mpg, aes(x=cty)) +
  geom_histogram() +
  theme_bw()
```




```
#Colored histogram with 20 bins
ggplot(mpg, aes(x=hwy)) +
  geom_histogram(bins=20, fill="red", color = 'black') +
  theme_bw()
```

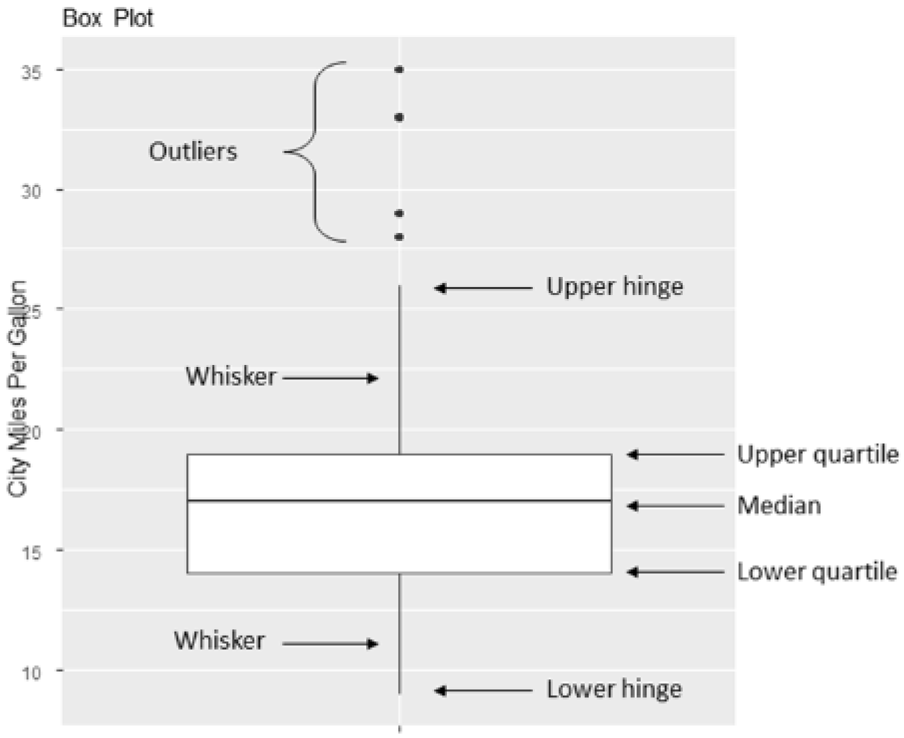



```
#Histogram with density cruve
ggplot(mpg, aes(x=hwy, y = ..density..)) +
  geom_histogram(bins=20, fill="red", color = 'black') +
  geom_density(color = 'blue', size = 1.5) +
  theme_bw()
```



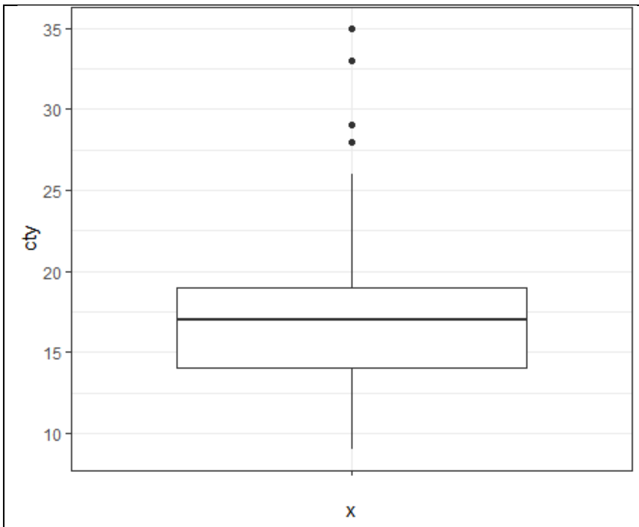
Box plots

A box-and-whiskers plot describes the distribution of a continuous variable by plotting its five-number summary: the minimum, lower quartile (25th percentile), median (50th percentile), upper quartile (75th percentile), and maximum. It can also display observations that may be outliers (values outside the range of Q3 + 1.5 × IQR to Q1 - 1.5 × IQR, where IQR is the interquartile range (Q3 – Q1) defined as the upper quartile minus the lower quartile).

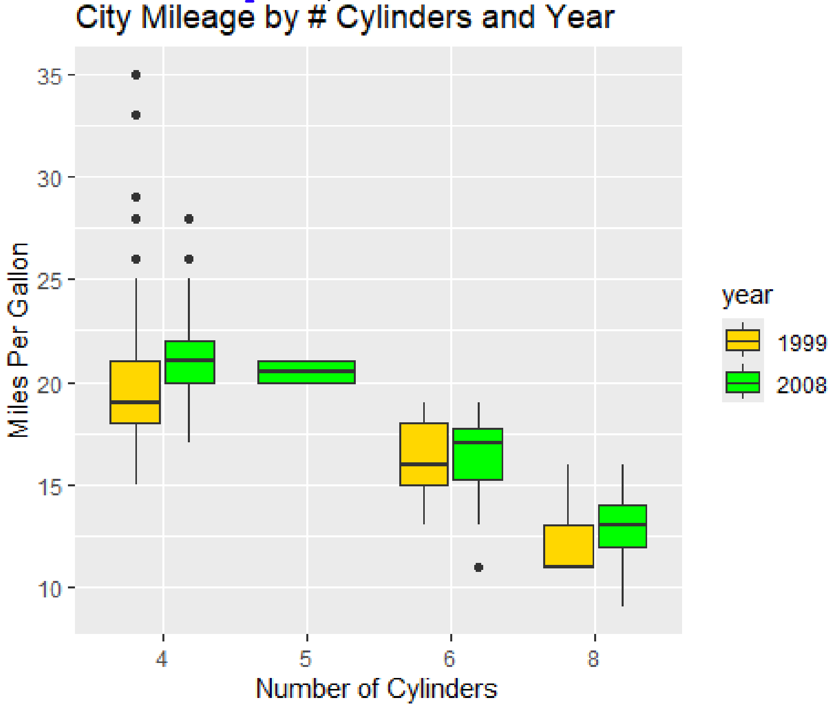


E023-boxplot.R

```
library(ggplot2)

ggplot(mpg, aes(x="", y=cty)) +
  geom_boxplot() +
  theme_bw()
```



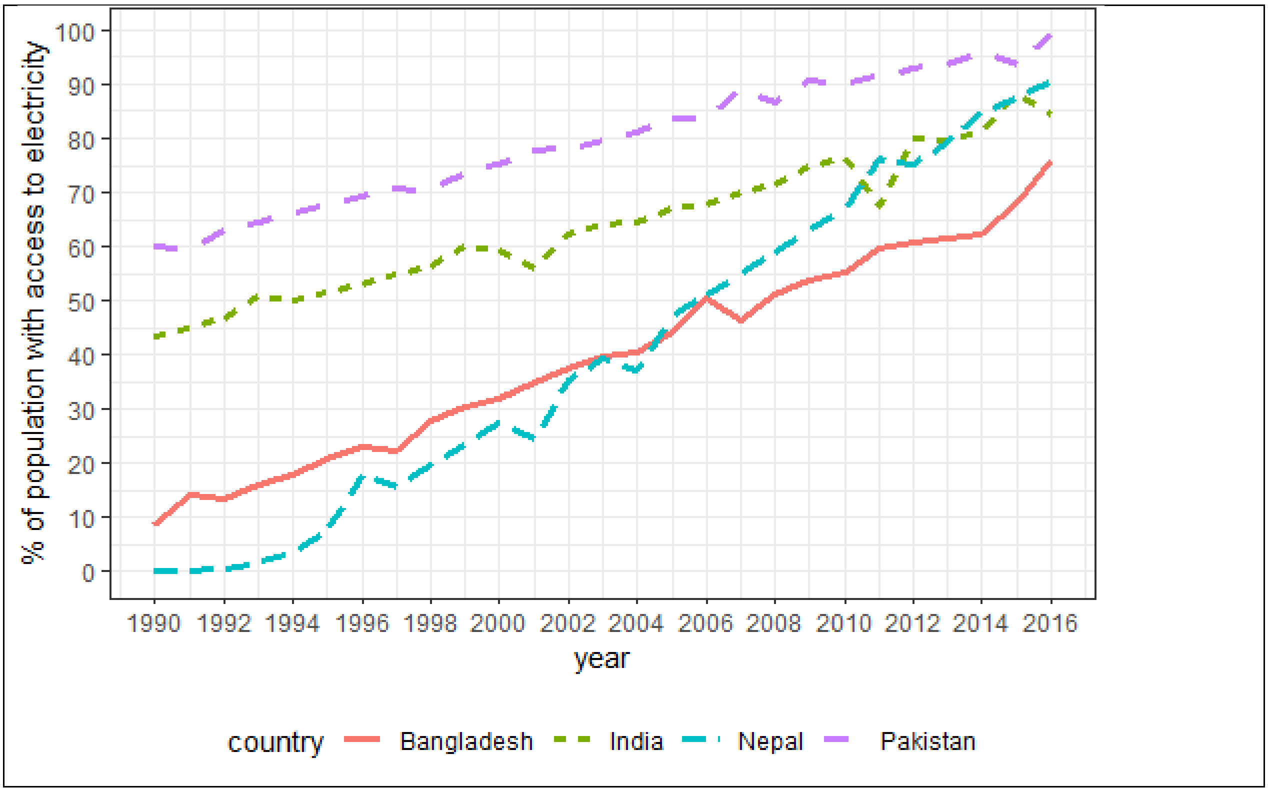

```
ggplot(mpg, aes(x=factor(cyl), y=cty, fill=factor(year))) +
  geom_boxplot() +
  scale_fill_manual(values=c("gold", "green")) +
  labs(x="Number of Cylinders",
       y="Miles Per Gallon",
       title="City Mileage by # Cylinders and Year",
       fill = "year")
```



Line plots

E024-line_plot.R

```
library(ggplot2)
library(dplyr)

wb_energy <- read.csv('data/006-wb_energy.csv')
df <- wb_energy %>% filter(country %in% c('Nepal', 'India', 'Bangladesh',
'Pakistan'))

ggplot(data = df, mapping = aes(x = year, y = ele_total, color = country,
linetype = country )) +
  geom_line(size = 1.3) +
  labs(y = '% of population with access to electricity') +
  theme_bw() +
  theme(legend.position = 'bottom') +
  scale_x_continuous(breaks = seq(1990,2020,2)) +
  scale_y_continuous(breaks = seq(0,100,10))
```



Session 6: R programming

6.1. Conditional execution

E025-conditional_execution.R

```
age <- 61

if (age <= 20) {
  print('Teen')
} else if (age <=60) {
  print('Adult')
} else {
  print('Old')
}
```

6.2. User-written functions

E026-user_written_function.R

```
age_classify <- function(age) {
  if (age <= 20) {
    age_type <- 'Teen'
  } else if (age <=60) {
    age_type <- 'Adult'
  } else {
    age_type <- 'Old'
  }
  return(age_type)
}

age_classify(15)
age_classify(35)
age_classify(85)
```

6.3. Looping

E027-looping.R

```
#-------------------------------------
# For loop
#-------------------------------------
```

```
#finding the sum of squares of 1,2,3,4,5
x <- 0
for (i in c(1,2,3,4,5)) {
  x <- x + i^2
}
print(x)

#finding the sum of 1 to 100
x <- 0
for (i in 1:100) {
  x <- x + i
}
print(x)

#finding the sum of odd numbers from 1 to 100
x <- 0
for (i in 1:100) {
  if (i %% 2 == 1) {
    x <- x + i
  }
}
print(x)


#------------------------------------
# While loop
#------------------------------------

#finding the sum of 1 to 100
x <- 0
i <- 0
while (i <= 100) {
  x <- x + i
  i <- i + 1
}
print(x)

#finding the sum of odd numbers from 1 to 100
x <- 0
i <- 0
while (i <= 100) {
  if (i %% 2 == 1) {
    x <- x + i
  }
  i <- i + 1
}
print(x)
```

Task 5:

Suppose there is no built-in function in R to calculate mean and standard deviation. Write a user defined functions ***func_mean*** and ***func_sd*** to calculate mean and standard deviation of a given vector.

```
vec <- c(3,5,2,3,4,2,5,6,7)
mean(vec) # 4.111111
sd(vec) # 1.763834

func_mean <- function(vv) {
  x <- 0
  count <- 0
  for (i in vv) {
    x <- x + i
    count <- count + 1
  }
  x_bar <- x/count
```

```
  return(x_bar)
}

func_sd <- function(vv) {
  x_bar <- func_mean(vv)
  x <- 0
  count <- 0
  for (i in vv) {
    x <- x + (x_bar - i)^2
    count <- count + 1
  }
  x_sd <- (x/(count-1))^(1/2)
  return(x_sd)
}

func_mean(vec)
func_sd(vec)
```

Session 7: Webscrapping, duplicates, and missing data

7.1. Webscrapping

E028-webscrapping.R

```
#-----------------------------------------------------------------
# importing csv data directly from the web
#-----------------------------------------------------------------
df <- read.csv("http://s.anilz.net/wb_energy")
head(df)

dx <- read.csv("https://data.ny.gov/api/views/d6yy-54nr/rows.csv")
head(dx)

#-----------------------------------------------------------------
# Using rvest package for static website scraping
#-----------------------------------------------------------------
#loading necessary packages
library(rvest) #see https://rvest.tidyverse.org/articles/harvesting-the-
web.html for details
library(dplyr)

#loading webpage content
webpage <- read_html("https://www.sharesansar.com/today-share-price")

#extracting table from the webpage
tables <- html_table(webpage)

#checking the number of tables available in the webpage
length(tables)

df1 <- tables[[1]]
head(df1)

#filtering upper and lower circuit stock
filtered_df1 <- df1 %>% filter(`Diff %` > 9 | `Diff %`< -9) %>%
arrange(`Diff %`)
filtered_df1

#**************************************
#*Obtaining Forex information from NRB
#**************************************
#loading webpage content
webpage <- read_html("https://www.nrb.org.np")

#extracting table from the webpage
tables <- html_table(webpage)

#checking the number of tables available in the webpage
```

```
length(tables)


df1 <- tables[[1]]
df2 <- tables[[2]]

df1
df2

#keeping USD and JPY only
filtered_df1 <- df1 %>% filter(Currency=='USD' | Currency =='JPY')
filtered_df1
```

Task 6 :

Web-scrape the Historical ranking table from https://en.wikipedia.org/wiki/ICC_Men%27s_T20I_Team_Rankings

```
webpage <-
read_html("https://en.wikipedia.org/wiki/ICC_Men%27s_T20I_Team_Rankings")
tables <- html_table(webpage)
length(tables)

df <- tables[[7]]
```

7.2. Finding duplicates

E029-duplicates.R

```
library(dplyr)

# Creating a sample data frame
df <- data.frame(
  ID = c(1, 2, 3, 3, 4, 5, 4, 3),
  Name = c("John", "Jane", "Mark", "Mark", "Luke", "Kate", "Luke", "Mark"),
  Age = c(25, 30, 35, 35, 40, 45, 40, 35)
)

df
  ID Name Age
1  1 John  25
2  2 Jane  30
3  3 Mark  35
4  3 Mark  35
5  4 Luke  40
6  5 Kate  45
7  4 Luke  40
8  3 Mark  35

#showing the duplicated observations
df %>% filter(duplicated(.) == T)
  ID Name Age
1  3 Mark  35
2  4 Luke  40
3  3 Mark  35

#removing the duplicated observations
df %>% filter(!duplicated(.) == T)
  ID Name Age
1  1 John  25
2  2 Jane  30
3  3 Mark  35
4  4 Luke  40
5  5 Kate  45

#counting the duplicate observations
df %>% group_by_all() %>%
  summarise(count = n()) %>%
```

```
  filter(count > 1)
     ID Name    Age count
  <dbl> <chr> <dbl> <int>
1     3 Mark     35     3
2     4 Luke     40     2

#identifying duplicate values based on some variables
library(haven)
df <- read_dta('data/008-nlfs2.dta')

#checking whether there is any duplicate based on selected variables
df %>% count(psu, hhid) %>% filter(n > 1)
    psu  hhid     n
  <dbl> <dbl> <int>
1  1001     1     4
2  1001     2     2
3  1001     7     2
4  1001     9     3
5  1001    10     2
6  1001    12     2
7  1001    13     2
8  1001    16     3
9  1001    17     2
10 1001    18     2
# i 10,937 more rows
# i Use `print(n = ...)` to see more rows

#selecting observations that does not have duplicate based on selected
variables
df %>% filter(duplicated(psu, hhid) == F) %>% count(psu, hhid)
    psu  hhid     n
  <dbl> <dbl> <int>
1  1001     1     1
2  1002     3     1
3  1003    13     1
4  1004    15     1
5  1005    16     1
6  1006    20     1
7  1007    11     1
8  1008    12     1
9  1009    19     1
10 1010     6     1
# i 789 more rows
# i Use `print(n = ...)` to see more rows
```

7.3. Finding missing values

E030-missing_values.R

```
library(haven)
library(dplyr)
df <- read_dta('data/008-nlfs2.dta')
df <- df[c('psu','hhid','q13','q18')]

#Selecting observations with no missing values
df %>% filter(complete.cases(.) == T)

#Selecting observations with missing values
df %>% filter(complete.cases(.) == F)
```

Session 8: Descriptive statistics and hypothesis testing

8.1. Descriptive statistics

E031-descriptive_statistics.R

```
data(mtcars) #from datasets package

vars <- c('mpg','cyl','disp')

#basic descriptive statistics from base package
summary(mtcars[vars])
      mpg             cyl             disp      
 Min.   :10.40   Min.   :4.000   Min.   : 71.1  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8  
 Median :19.20   Median :6.000   Median :196.3  
 Mean   :20.09   Mean   :6.188   Mean   :230.7  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0  
 Max.   :33.90   Max.   :8.000   Max.   :472.0  

#descriptive statistics from other packages
Hmisc::describe(mtcars[vars])
mtcars[vars] 

 3  Variables      32  Observations
--------------------------------------------------------------------------------------------------------------------------
mpg 
       n  missing distinct     Info     Mean  pMedian      Gmd      .05      .10      .25      .50      .75      .90      .95 
      32        0       25    0.999    20.09     19.6    6.796    12.00    14.34    15.43    19.20    22.80    30.09    31.30 

lowest : 10.4 13.3 14.3 14.7 15  , highest: 26   27.3 30.4 32.4 33.9
--------------------------------------------------------------------------------------------------------------------------
cyl 
       n  missing distinct     Info     Mean  pMedian      Gmd 
      32        0        3    0.866    6.188        6    1.948 

Value          4     6     8
Frequency     11     7    14
Proportion 0.344 0.219 0.438

For the frequency table, variable is rounded to the nearest 0
--------------------------------------------------------------------------------------------------------------------------
disp 
       n  missing distinct     Info     Mean  pMedian      Gmd      .05      .10      .25      .50      .75      .90      .95 
      32        0       27    0.999    230.7    223.4    142.5    77.35    80.61   120.83   196.30   326.00   396.00   449.00 

lowest : 71.1 75.7 78.7 79   95.1, highest: 360  400  440  460  472 
--------------------------------------------------------------------------------------------------------------------------

pastecs::stat.desc(mtcars[vars])
                     mpg         cyl         disp
nbr.val       32.0000000  32.0000000 3.200000e+01
nbr.null       0.0000000   0.0000000 0.000000e+00
nbr.na         0.0000000   0.0000000 0.000000e+00
min           10.4000000   4.0000000 7.110000e+01
max           33.9000000   8.0000000 4.720000e+02
range         23.5000000   4.0000000 4.009000e+02
sum          642.9000000 198.0000000 7.383100e+03
median        19.2000000   6.0000000 1.963000e+02
mean          20.0906250   6.1875000 2.307219e+02
SE.mean        1.0654240   0.3157093 2.190947e+01
CI.mean.0.95   2.1729465   0.6438934 4.468466e+01
var           36.3241028   3.1895161 1.536080e+04
std.dev        6.0269481   1.7859216 1.239387e+02
coef.var       0.2999881   0.2886338 5.371779e-01

psych::describe(mtcars[vars])
     vars  n   mean     sd median trimmed    mad  min   max range  skew kurtosis    se
mpg     1 32  20.09   6.03   19.2   19.70   5.41 10.4  33.9  23.5  0.61    -0.37  1.07
cyl     2 32   6.19   1.79    6.0    6.23   2.97  4.0   8.0   4.0 -0.17    -1.76  0.32
disp    3 32 230.72 123.94  196.3  222.52 140.48 71.1 472.0 400.9  0.38    -1.21 21.91

#-----------------------------------
# Descriptive statistics by group
#-----------------------------------
#grouping by one variable
by(mtcars[vars], #dataset
   list(Transmission = mtcars$am), #grouping variable: Transmission (0 = 
automatic, 1 = manual)
   summary) #function
```

```
Transmission: 0
       mpg             cyl             disp
 Min.   :10.40   Min.   :4.000   Min.   :120.1
 1st Qu.:14.95   1st Qu.:6.000   1st Qu.:196.3
 Median :17.30   Median :8.000   Median :275.8
 Mean   :17.15   Mean   :6.947   Mean   :290.4
 3rd Qu.:19.20   3rd Qu.:8.000   3rd Qu.:360.0
 Max.   :24.40   Max.   :8.000   Max.   :472.0
-----------------------------------------------
Transmission: 1
       mpg             cyl             disp
 Min.   :15.00   Min.   :4.000   Min.   : 71.1
 1st Qu.:21.00   1st Qu.:4.000   1st Qu.: 79.0
 Median :22.80   Median :4.000   Median :120.3
 Mean   :24.39   Mean   :5.077   Mean   :143.5
 3rd Qu.:30.40   3rd Qu.:6.000   3rd Qu.:160.0
 Max.   :33.90   Max.   :8.000   Max.   :351.0

#grouping by multiple variables
by(mtcars[vars],
   list(Transmission = mtcars$am, Engine = mtcars$vs), # Transmission (0 = automatic, 1 = manual), Engine (0 = V-shaped, 1 = straight)
   summary)
Transmission: 0
Engine: 0
      mpg           cyl         disp
 Min.   :10.40   Min.   :8   Min.   :275.8
 1st Qu.:14.05   1st Qu.:8   1st Qu.:296.9
 Median :15.20   Median :8   Median :355.0
 Mean   :15.05   Mean   :8   Mean   :357.6
 3rd Qu.:16.62   3rd Qu.:8   3rd Qu.:410.0
 Max.   :19.20   Max.   :8   Max.   :472.0
-----------------------------------------------
Transmission: 1
Engine: 0
      mpg             cyl             disp
 Min.   :15.00   Min.   :4.000   Min.   :120.3
 1st Qu.:16.77   1st Qu.:6.000   1st Qu.:148.8
 Median :20.35   Median :6.000   Median :160.0
 Mean   :19.75   Mean   :6.333   Mean   :206.2
 3rd Qu.:21.00   3rd Qu.:7.500   3rd Qu.:265.8
 Max.   :26.00   Max.   :8.000   Max.   :351.0
-----------------------------------------------
Transmission: 0
Engine: 1
      mpg             cyl             disp
 Min.   :17.80   Min.   :4.000   Min.   :120.1
 1st Qu.:18.65   1st Qu.:4.000   1st Qu.:143.8
 Median :21.40   Median :6.000   Median :167.6
 Mean   :20.74   Mean   :5.143   Mean   :175.1
 3rd Qu.:22.15   3rd Qu.:6.000   3rd Qu.:196.3
 Max.   :24.40   Max.   :6.000   Max.   :258.0
-----------------------------------------------
Transmission: 1
Engine: 1
      mpg           cyl         disp
 Min.   :21.40   Min.   :4   Min.   : 71.1
 1st Qu.:25.05   1st Qu.:4   1st Qu.: 77.2
 Median :30.40   Median :4   Median : 79.0
 Mean   :28.37   Mean   :4   Mean   : 89.8
 3rd Qu.:31.40   3rd Qu.:4   3rd Qu.:101.5
 Max.   :33.90   Max.   :4   Max.   :121.0
```

8.2. Descriptive statistics

E032-frequency_contingency_tables.R

```
Arthritis <- vcd::Arthritis

#simple frequency table
mytable <- table(Arthritis$Improved)
mytable
 None   Some Marked 
    42     14     28 

#proportion table
prop.table(mytable)
     None      Some    Marked 
0.5000000 0.1666667 0.3333333 
```

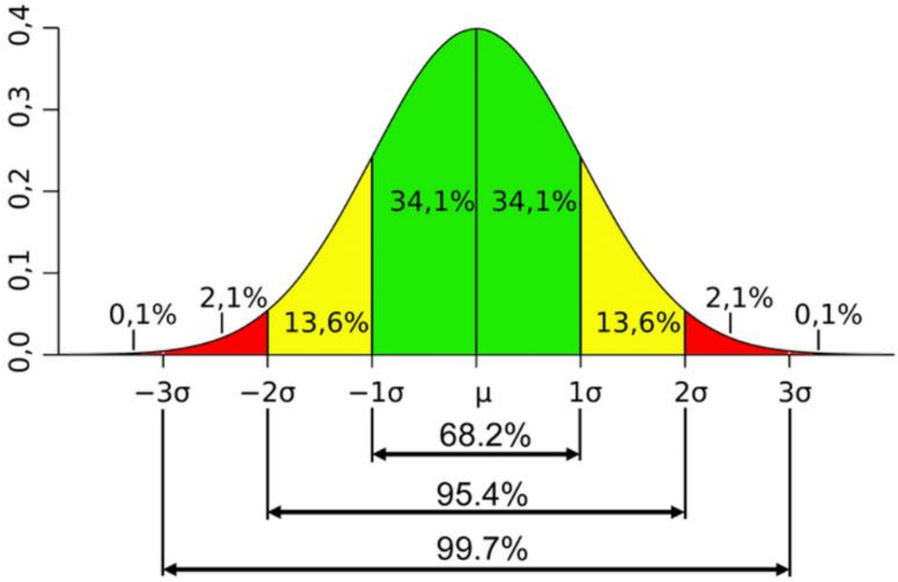
```
prop.table(mytable)*100 #in percentage
    None     Some   Marked 
50.00000 16.66667 33.33333 

#-------------------------------------
# Two-way table
#-------------------------------------
mytable <- xtabs(~ Treatment + Improved, data=Arthritis)
mytable
         Improved
Treatment None Some Marked
  Placebo   29    7      7
  Treated   13    7     21

#calculating sub-total horizontally
margin.table(mytable, 1) # 1 here refers 1st variable i.e. Treatment
Treatment
Placebo Treated 
     43      41 

#proportion table based on horizontal sub-total
prop.table(mytable, 1) * 100
         Improved
Treatment     None     Some   Marked
  Placebo 67.44186 16.27907 16.27907
  Treated 31.70732 17.07317 51.21951

# ********************************************

#calculating sub-total vertically
margin.table(mytable, 2) # 2 here refers 2nd variable i.e. Improved
Improved
  None   Some Marked 
    42     14     28 

#proportion table based on vertical sub-total
prop.table(mytable, 2) * 100
         Improved
Treatment     None     Some   Marked
  Placebo 69.04762 50.00000 25.00000
  Treated 30.95238 50.00000 75.00000

#--------------------------------------------
# Two-way table (add sub-totals and grand totals)
#--------------------------------------------
addmargins(mytable)
         Improved
Treatment None Some Marked Sum
  Placebo   29    7      7  43
  Treated   13    7     21  41
  Sum       42   14     28  84

addmargins(prop.table(mytable)) * 100
         Improved
Treatment       None        Some      Marked         Sum
  Placebo  34.523810   8.333333   8.333333  51.190476
  Treated  15.476190   8.333333  25.000000  48.809524
  Sum      50.000000  16.666667  33.333333 100.000000

#proportion addmargins horizontally
addmargins(prop.table(mytable, 1), 2) * 100
         Improved
Treatment      None      Some    Marked       Sum
  Placebo  67.44186  16.27907  16.27907 100.00000
  Treated  31.70732  17.07317  51.21951 100.00000
```

```
#proportion addmargins vertically
addmargins(prop.table(mytable, 2), 1) * 100
         Improved
Treatment       None      Some    Marked
  Placebo   69.04762  50.00000  25.00000
  Treated   30.95238  50.00000  75.00000
  Sum      100.00000 100.00000 100.00000

#-------------------------------------------------
# Multidimensional table
#-------------------------------------------------
mytable <- xtabs(~ Treatment + Sex + Improved, data=Arthritis)
mytable
, , Improved = None

         Sex
Treatment Female Male
  Placebo     19   10
  Treated      6    7

, , Improved = Some

         Sex
Treatment Female Male
  Placebo      7    0
  Treated      5    2

, , Improved = Marked

         Sex
Treatment Female Male
  Placebo      6    1
  Treated     16    5

#frequency table
ftable(mytable)
                 Improved None Some Marked
Treatment Sex
Placebo   Female            19    7      6
          Male              10    0      1
Treated   Female             6    5     16
          Male               7    2      5

#frequency table defining column variables
ftable(mytable, col.vars = c('Sex','Improved'))
          Sex      Female           Male
          Improved   None Some Marked None Some Marked
Treatment
Placebo                19    7      6   10    0      1
Treated                 6    5     16    7    2      5

#proportion table
prop.table(ftable(mytable, col.vars = c('Sex','Improved'))) * 100
          Sex          Female                          Male
          Improved       None      Some    Marked      None      Some    Marked
Treatment
Placebo             22.619048  8.333333  7.142857 11.904762  0.000000  1.190476
Treated              7.142857  5.952381 19.047619  8.333333  2.380952  5.952381
```

8.3. The concept of normal distribution

a. What is a Normal Distribution?

- **Shape:** The normal distribution looks like a bell-shaped curve.
- **Symmetry:** It is perfectly symmetrical around the center.



b. **Key Characteristics:**

- **Mean (Average):** The center of the curve.
- **Standard Deviation:** Measures the spread of the data.
  - 68.2% of the data falls within 1 standard deviation of the mean.
  - 95.4% falls within 2 standard deviations.
  - 99.7% falls within 3 standard deviations.

c. **Why is it Important?**

- **Natural Occurrences:** Many natural phenomena follow this distribution (e.g., heights, test scores). For example, most students score around the average in a class, fewer scoring very high or very low.
- **Central Limit Theorem:** In large samples, the samples' mean tend to be normally distributed. (Video)
- **Statistical Inferences:** Helps in making predictions and decisions based on data.

8.4. Hypothesis testing

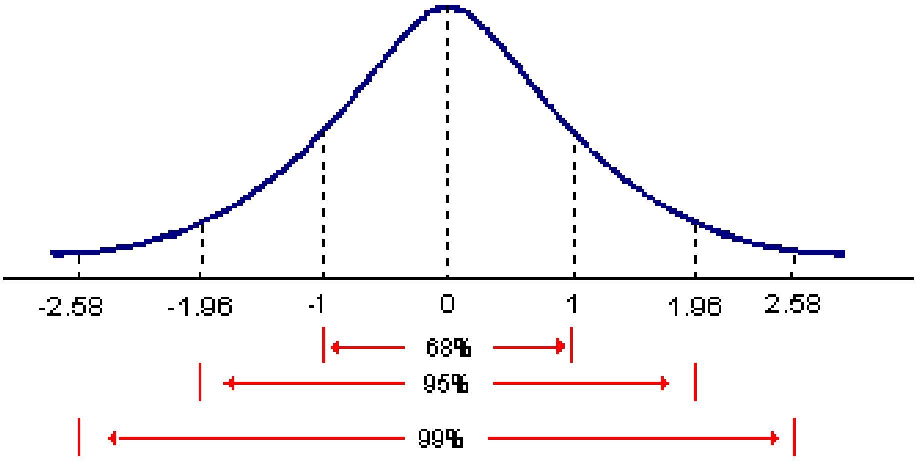
a. What is Hypothesis Testing?

- Hypothesis testing is a method used to decide whether there is enough evidence to support a particular claim about a population based on a sample of data.
- **Null Hypothesis (H_0):** This is the default statement that there is no effect or no difference. It assumes that any observed differences are due to random chance.
  Example: "The average age is equal to 20."
- **Alternative Hypothesis (H_1):** This is what you want to prove, stating there is an effect or a difference.
  Example: "The average age is not equal to 20."

b. Procedure of hypothesis testing

- State the null and alternative hypothesis. (e.g. $H_0: \mu = 0$, $H_1: \mu \neq 0$)
- Collect sample data.
- Calculate sample mean and stadard error ($\frac{s}{\sqrt{n}}$).

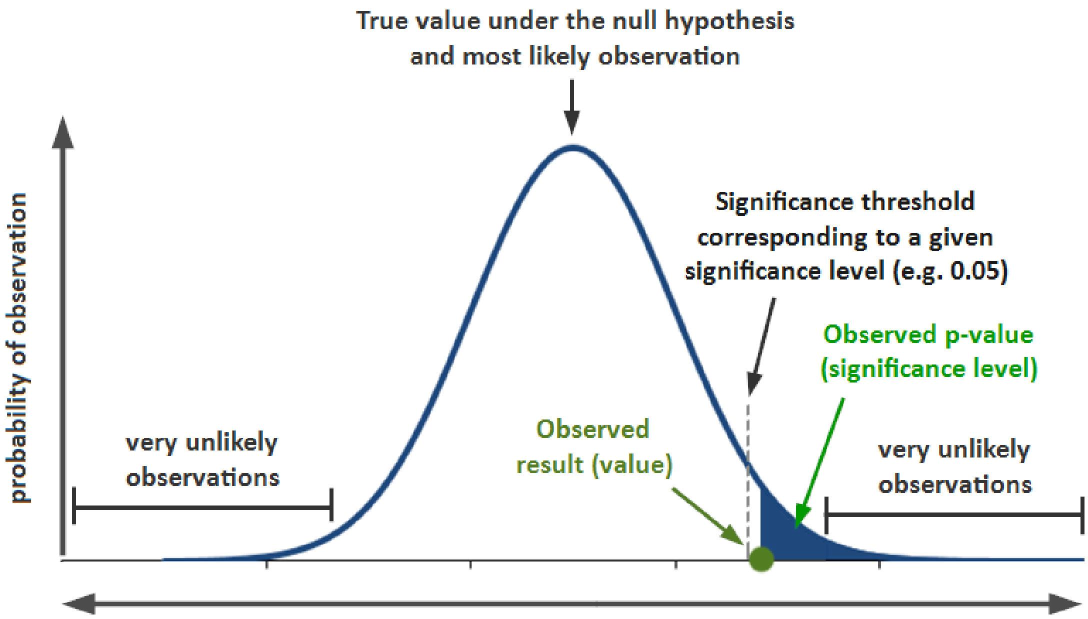
- Calculate t-statistics ($t = \frac{\bar{X}-\mu}{Standar\ Error}$).
- Compare absolute value of t-statistics |t| with critical values for given level of significance (α). [1.65 (10% significance level), 1.96 (5%), 2.58 (1%)]



- Decision: reject null hypothesis if |t| exceeds critical value, otherwise fail to reject null hypothesis.

c. Hypothesis testing with p-value

- p-value : probability (area under normal distribution) beyond |t| i.e., probability of rejecting H_0 when its true.



- **Decision :** reject null hypothesis if p-value is lower than the significance level, otherwise fail to reject null hypothesis.
- Easier to conduct hypothesis testing with p-value. No need to calculate t-statistics and remember different critical values.

E033-hypothesis_testing.R

```
library(dplyr)

# Set seed for reproducibility
set.seed(12345)

# Create a dummy dataset
n <- 1000
group <- rep(0:1, length.out = n)
score <- 50 + group * 10 + rnorm(n, mean = 0, sd = 10)
```

```
# Combine into a data frame
data <- data.frame(group = group, score = score)

# Conducting hypothesis testing (one-sample t-tests)
t.test(data$score, mu = 50)  # H0: pop_mean = 50
        One Sample t-test

data:  data$score
t = 15.613, df = 999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 50
95 percent confidence interval:
 54.77547 56.14850
sample estimates:
mean of x
 55.46198

t.test(data$score, mu = 55)  # H0: pop_mean = 55
        One Sample t-test

data:  data$score
t = 1.3205, df = 999, p-value = 0.187
alternative hypothesis: true mean is not equal to 55
95 percent confidence interval:
 54.77547 56.14850
sample estimates:
mean of x
 55.46198

t.test(data$score, mu = 60)  # H0: pop_mean = 60
        One Sample t-test

data:  data$score
t = -12.972, df = 999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 60
95 percent confidence interval:
 54.77547 56.14850
sample estimates:
mean of x
 55.46198

# ---------------------------------------
# Conducting two-sample t-test
# ---------------------------------------
data0 <- filter(data, group == 0)
data1 <- filter(data, group == 1)
t.test(data0$score, data1$score)
        Welch Two Sample t-test

data:  data0$score and data1$score
t = -15.074, df = 997.82, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -10.763662  -8.284043
sample estimates:
mean of x mean of y
 50.70006  60.22391

#OR

t.test(score ~ group, data = data)  # H0: pop_mean_group1 = pop_mean_group2
```

```
        Welch Two Sample t-test

data:  score by group
t = -15.074, df = 997.82, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
95 percent confidence interval:
 -10.763662  -8.284043
sample estimates:
mean in group 0 mean in group 1 
       50.70006        60.22391 
```

Task 7:

Using NMICS6 data (009-hl.sav), conduct a hypothesis test whether average age between male and female is statistically different.

```
# Load necessary libraries
library(haven)  # For reading SPSS files
library(dplyr)

# Import SPSS file from the URL
df <- read_spss('data/009-hl.sav')

# HL6 -> Age, HL4 -> Sex
# Summarize age for males (HL4 == 1) and females (HL4 == 2)
df_male <- filter(df, HL4 == 1)
df_female <- filter(df, HL4 == 2)
mean(df_male$HL6)
mean(df_female$HL6)
#OR
by(df$HL6, df$HL4, summary) #average age : male = 28.26, female = 28.83

# Conduct hypothesis testing (two-sample t-test)
t.test(HL6 ~ HL4, data = df)
        Welch Two Sample t-test

data:  HL6 by HL4
t = -3.1618, df = 54737, p-value = 0.001569
alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
95 percent confidence interval:
 -0.9125138 -0.2141086
sample estimates:
mean in group 1 mean in group 2 
       28.26344        28.82675 

# Alternatively, use linear regression
regression_result <- lm(HL6 ~ HL4, data = df)
summary(regression_result)
Call:
lm(formula = HL6 ~ HL4, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-28.827 -17.827  -3.827  15.173  69.737 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  27.7001     0.2860  96.846  < 2e-16 ***
HL4           0.5633     0.1775   3.173  0.00151 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.07 on 56593 degrees of freedom
Multiple R-squared:  0.0001779, Adjusted R-squared:  0.0001602 
F-statistic: 10.07 on 1 and 56593 DF,  p-value: 0.001509
```