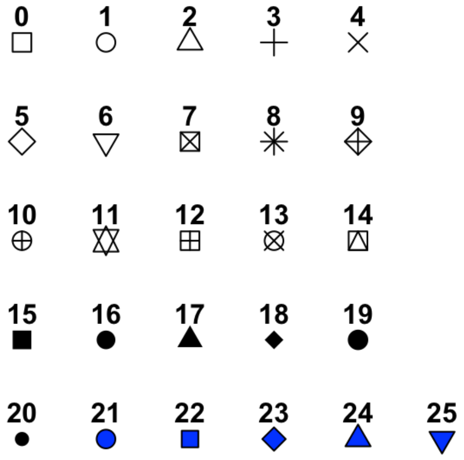
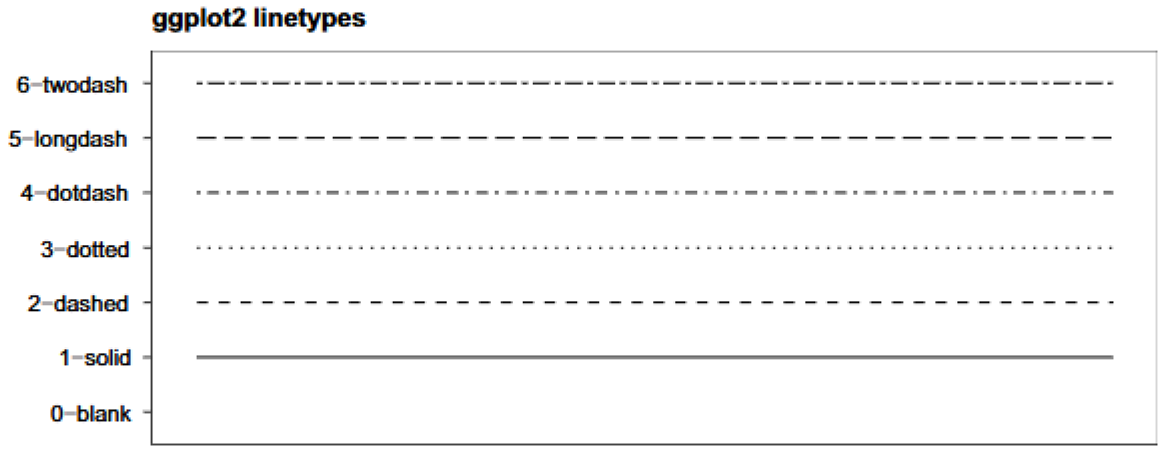
Day 2 : Advance R programming
Session 5: Graphing
5.1. ggplot2 basics

shape

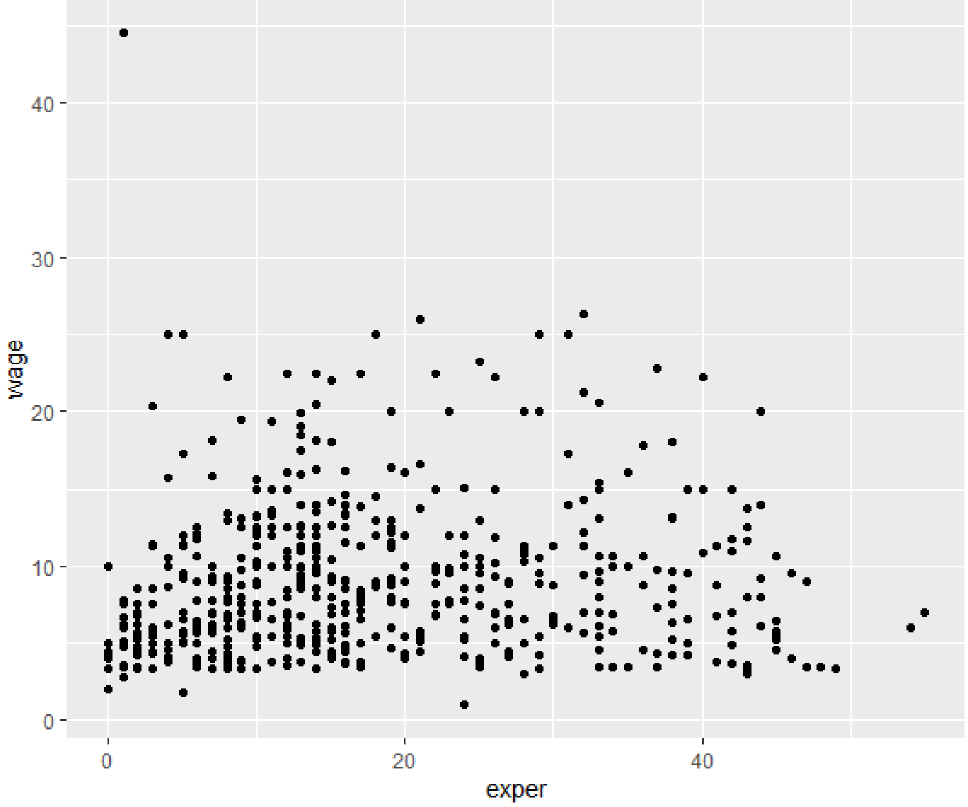


linetype

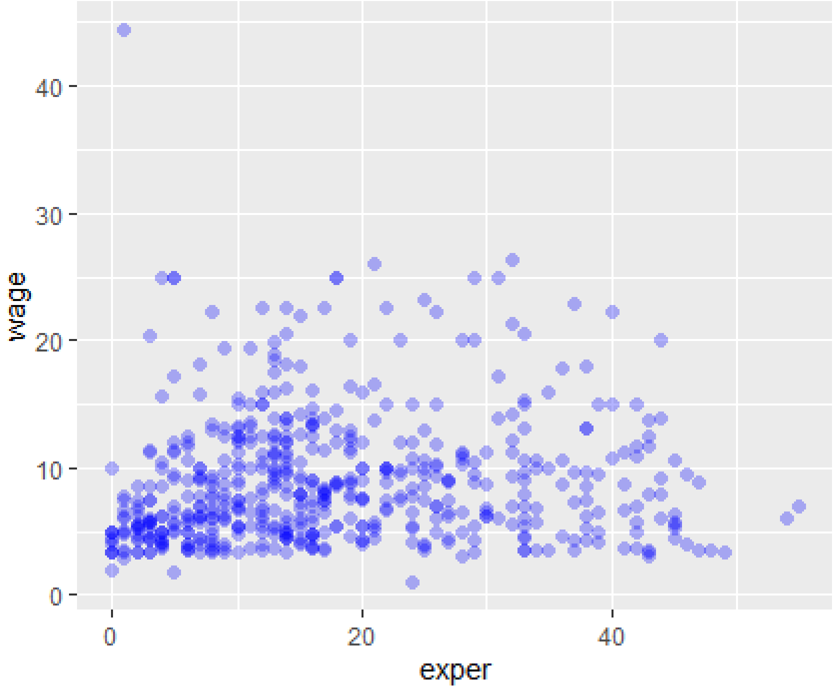


E018-ggplot2_basics.R

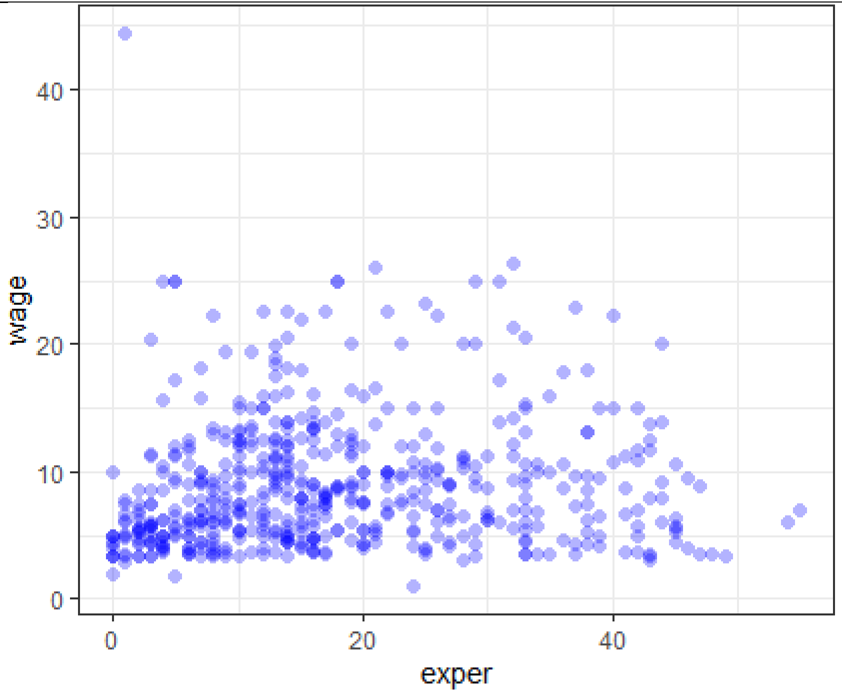
```
#-----------------------------------------------------------------------------
# ggplot2 basics
#-----------------------------------------------------------------------------
library(ggplot2)
library(mosaicData)

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



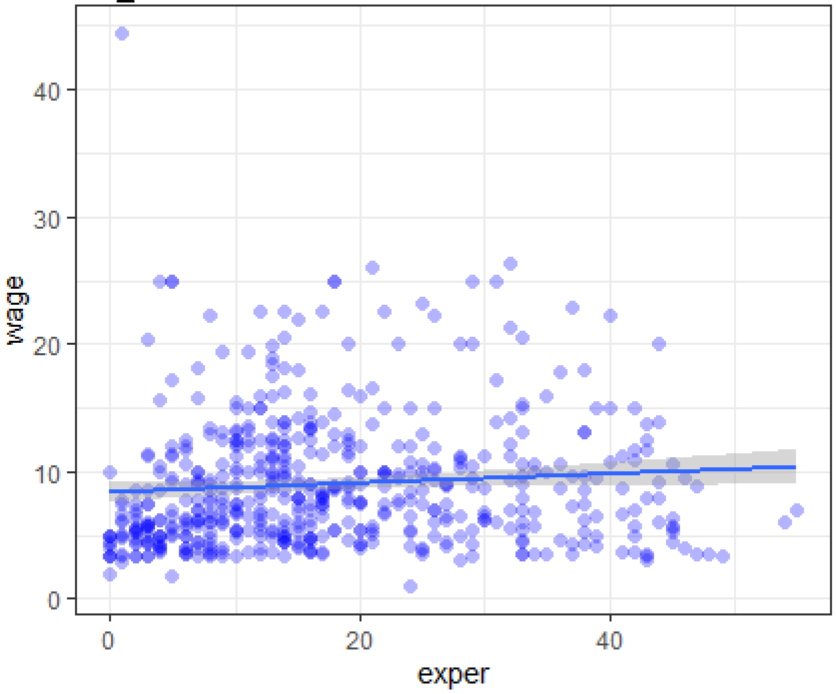
```
#scatter plot with various attributes
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(color = 'blue', shape = 16, alpha = 0.3, size = 2)
```



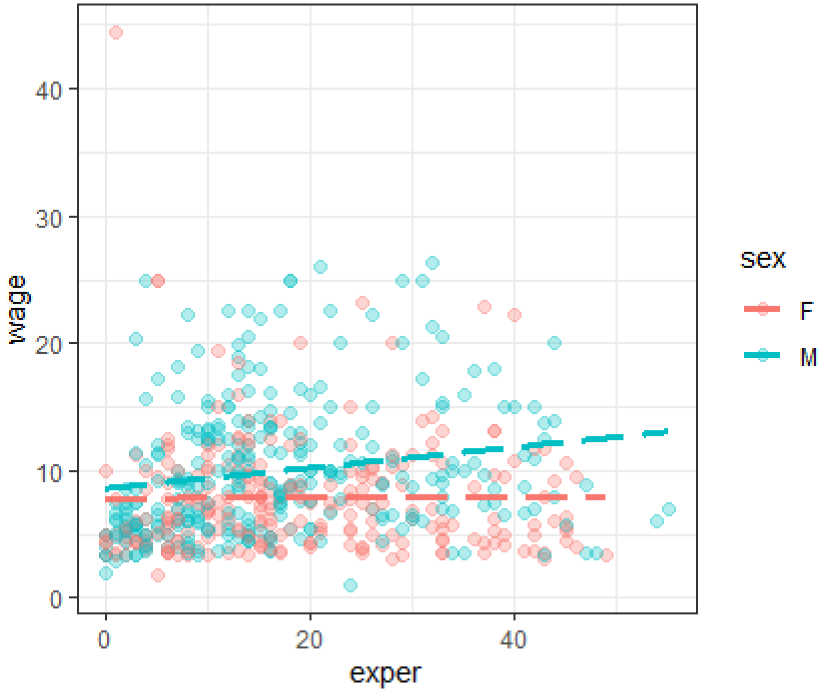
```
#applying ggplot2 themes
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(color = 'blue', shape = 16, alpha = 0.3, size = 2) +
  theme_bw()
```



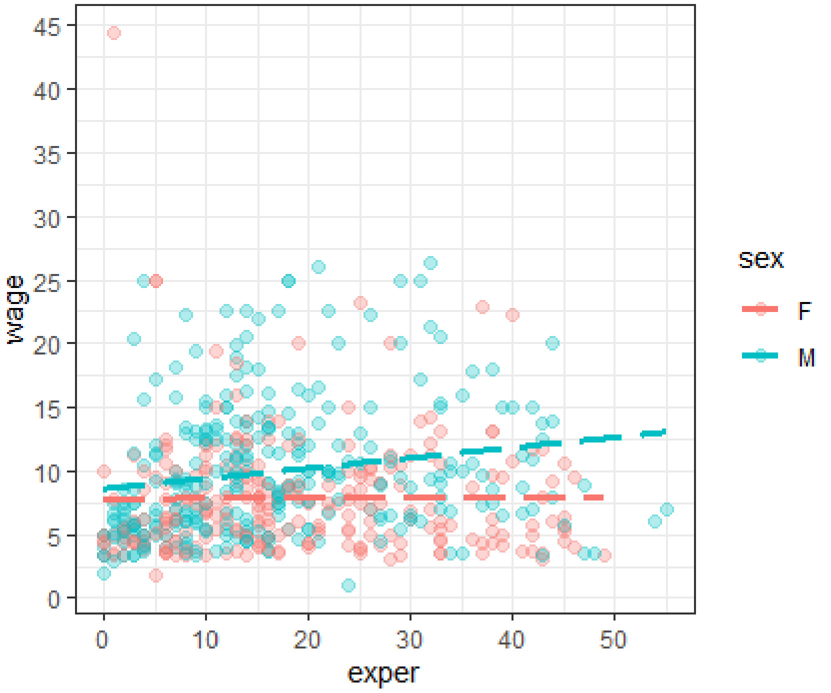
```
#scatter plot and best fit line
ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(color = 'blue', shape = 16, alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm') +
  theme_bw()
```



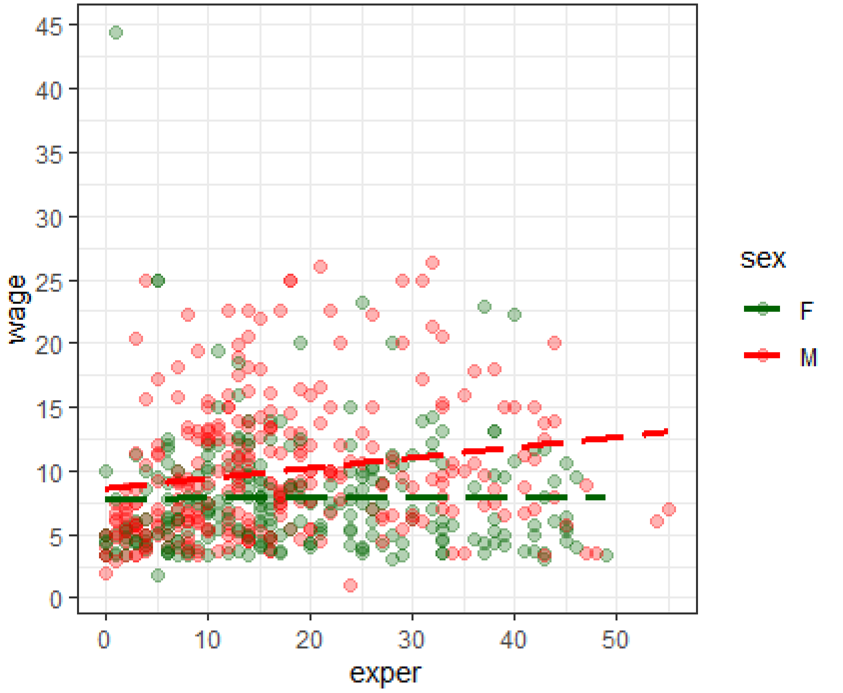
```
# grouping category variable attribute color
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  theme_bw()
```



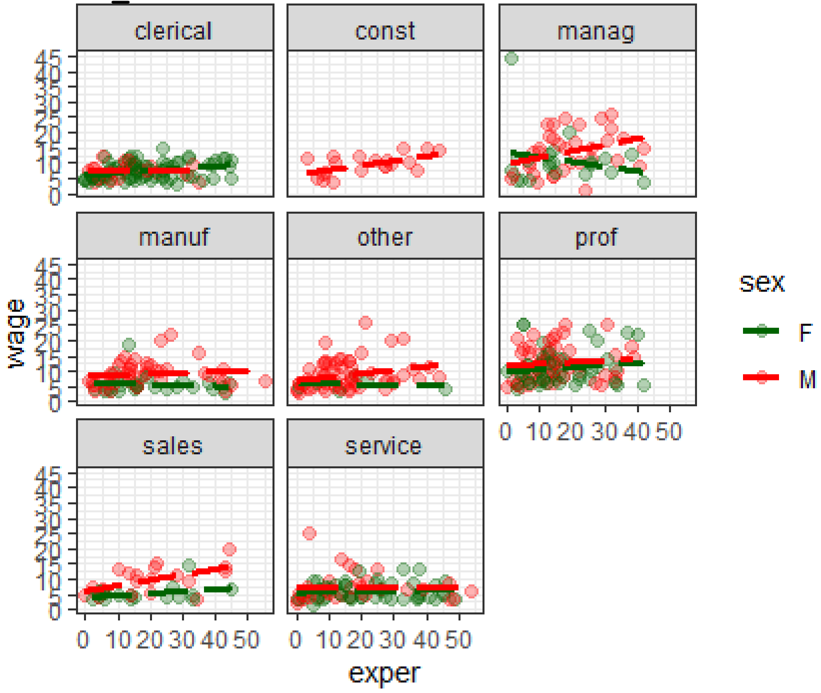
```
# x-axis and y-axis setting scales
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks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



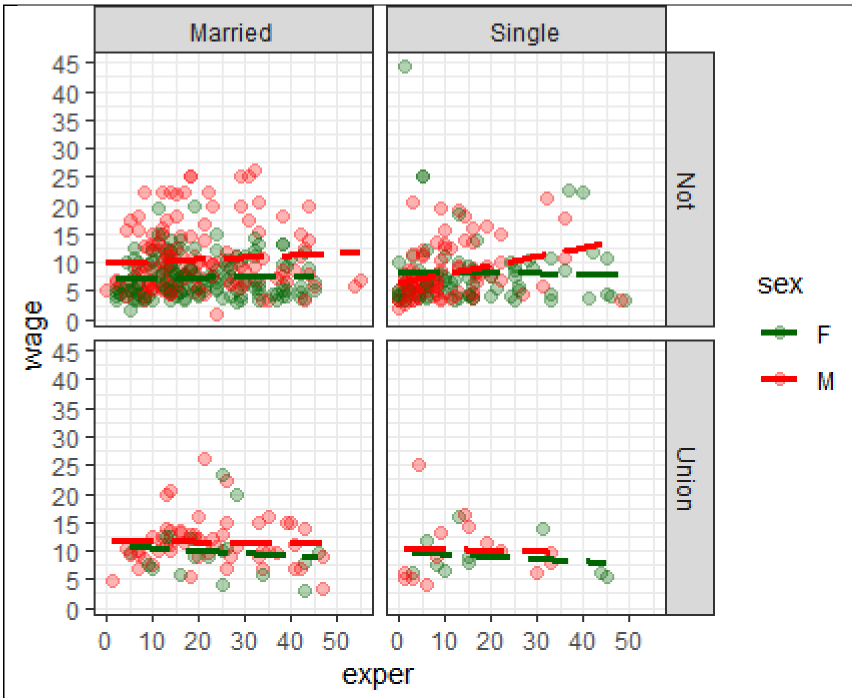
```
# color manual of grouped category variable
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks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



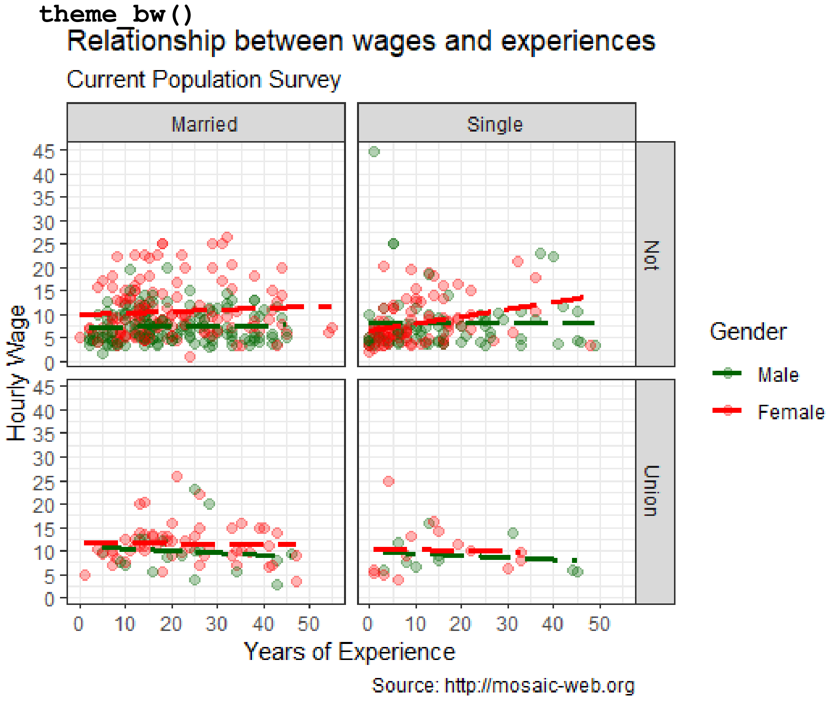
```
# subplots according to a categorical variable (facet_wrap)
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks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



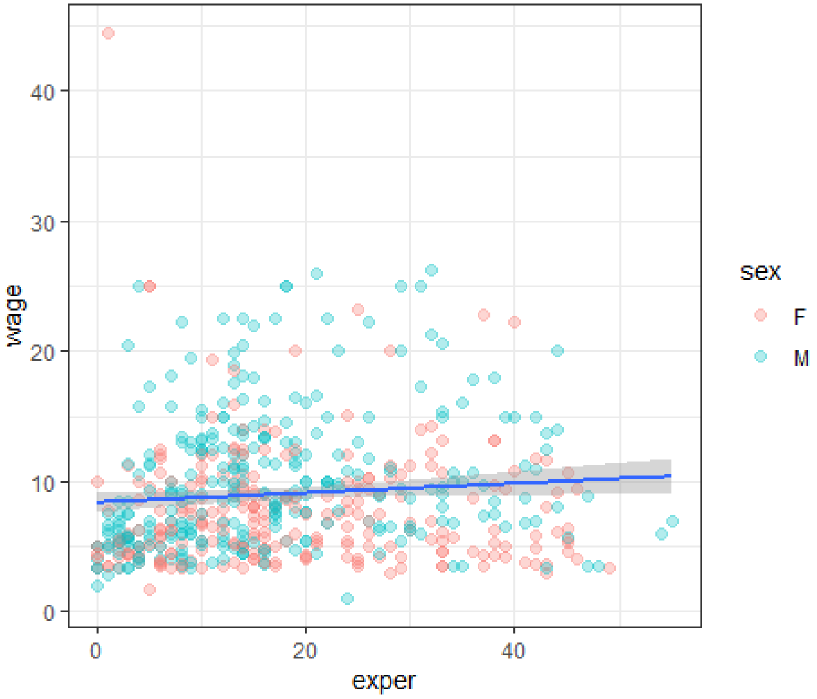
```
# labels
ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +
  geom_point(alpha = 0.3, size = 2) +
  geom_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash')
+
  scale_x_continuous(breaks = seq(0,70,10)) +
  scale_y_continuous(breaks = seq(0,60,5)) +
  scale_color_manual(values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



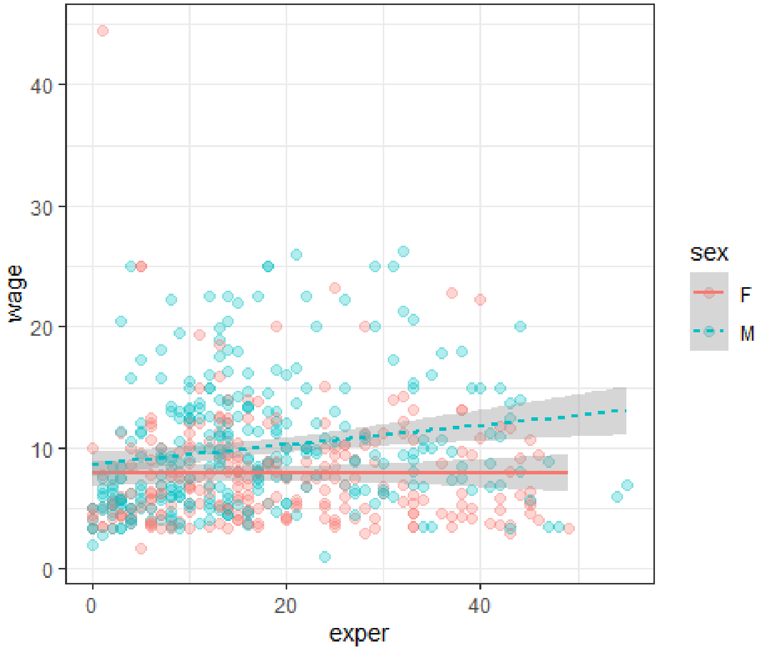
Mapping in individual geom functions, stroing the plot as an R object, and exporting plot

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()
```



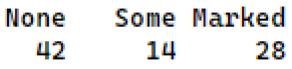
```
#storing the plot in an object
myplot <- ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +
  geom_point(mapping = aes(color = sex), alpha = 0.3, size = 2) +
  geom_smooth(mapping = aes(linetype = sex, color = sex), method = 'lm') +
  theme_bw()
myplot

#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

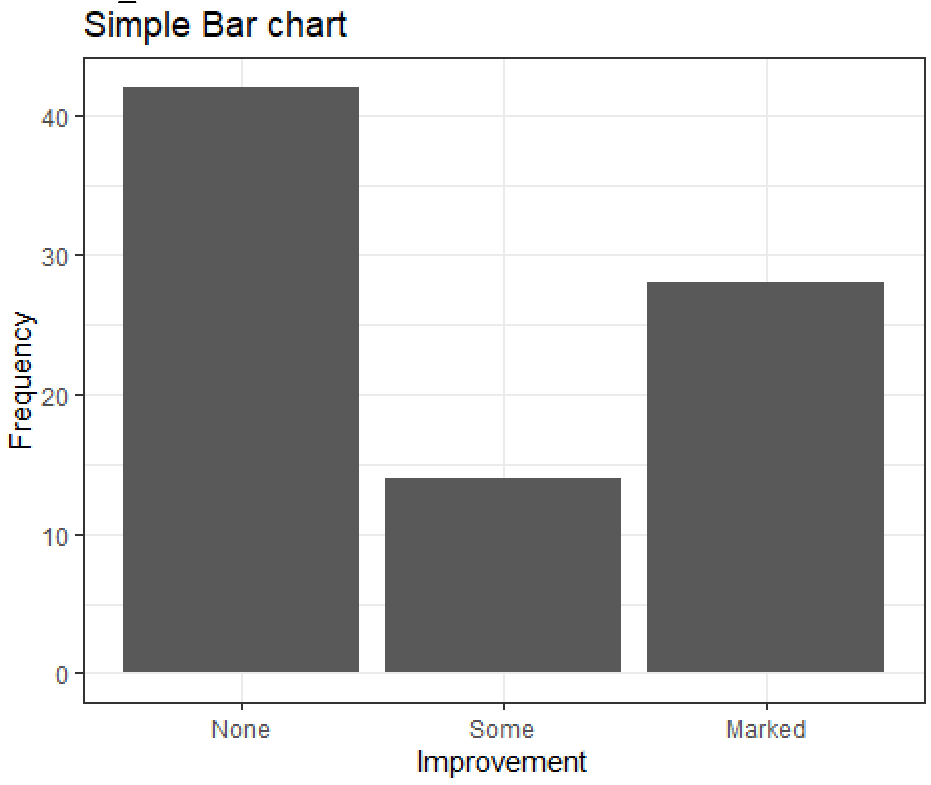
E020-bar_chart.R

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

#simple bar chart
table(Arthritis$Improved)
```

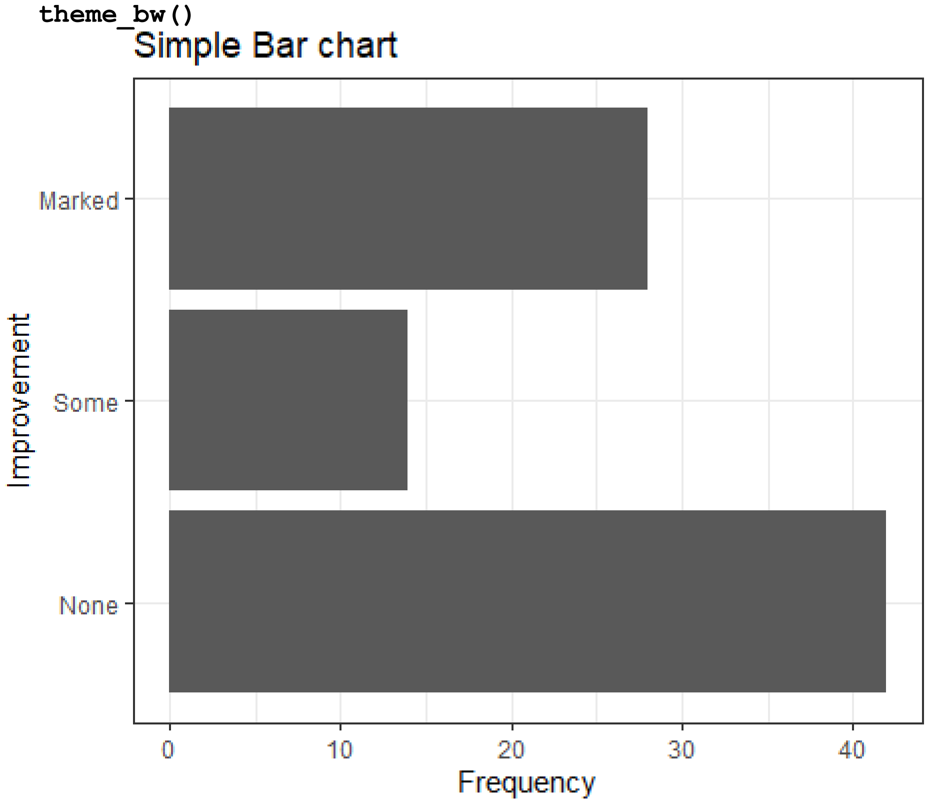


```
None    Some Marked
  42      14     28
```

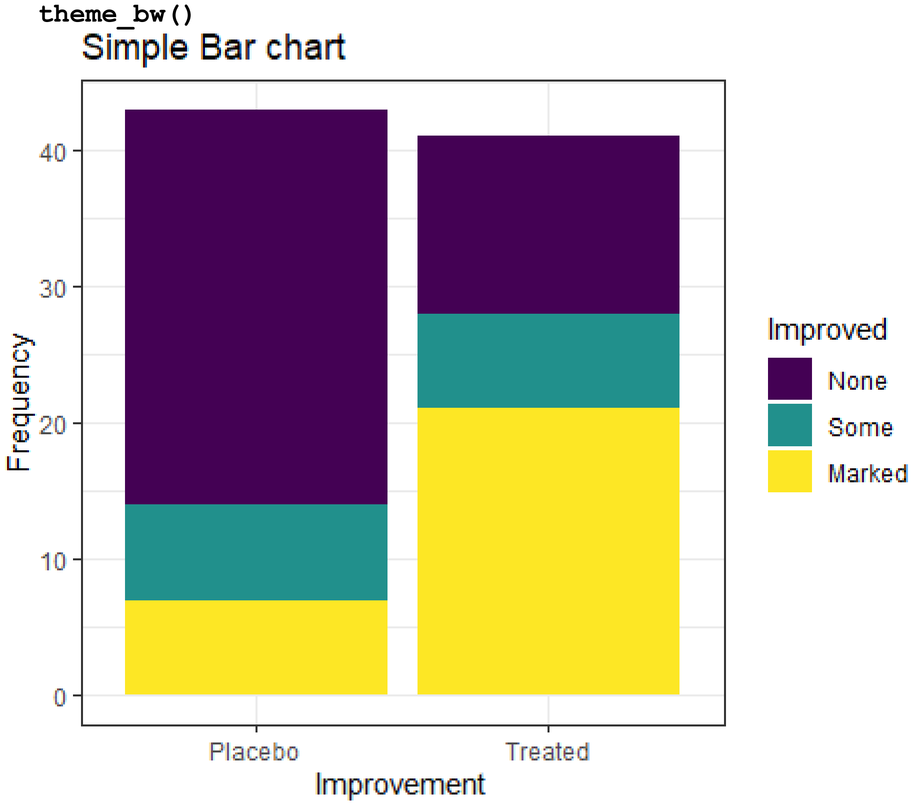
```
ggplot(Arthritis, aes(x=Improved)) + geom_bar() +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Frequency") +
  theme_bw()
```



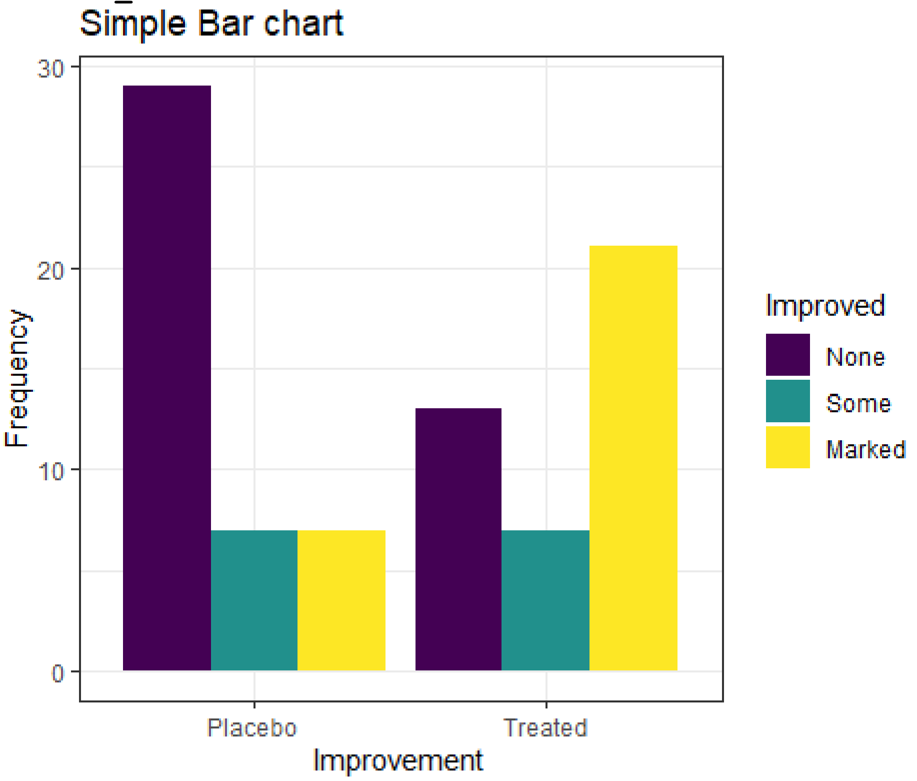
```
#Horizontal bar chart
ggplot(Arthritis, aes(x=Improved)) + geom_bar() +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Frequency") +
  coord_flip()+
  theme_bw()
```



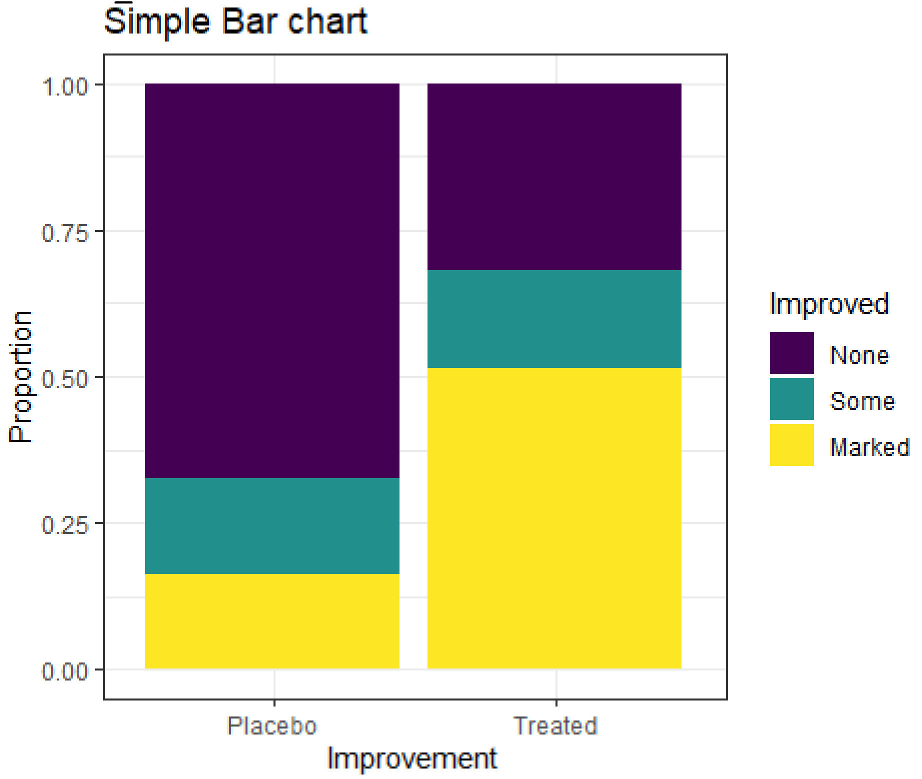
```
#Stacked bar chart
table(Arthritis$Improved, Arthritis$Treatment)
        
         Placebo Treated
  None        29      13
  Some         7       7
  Marked       7      21

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



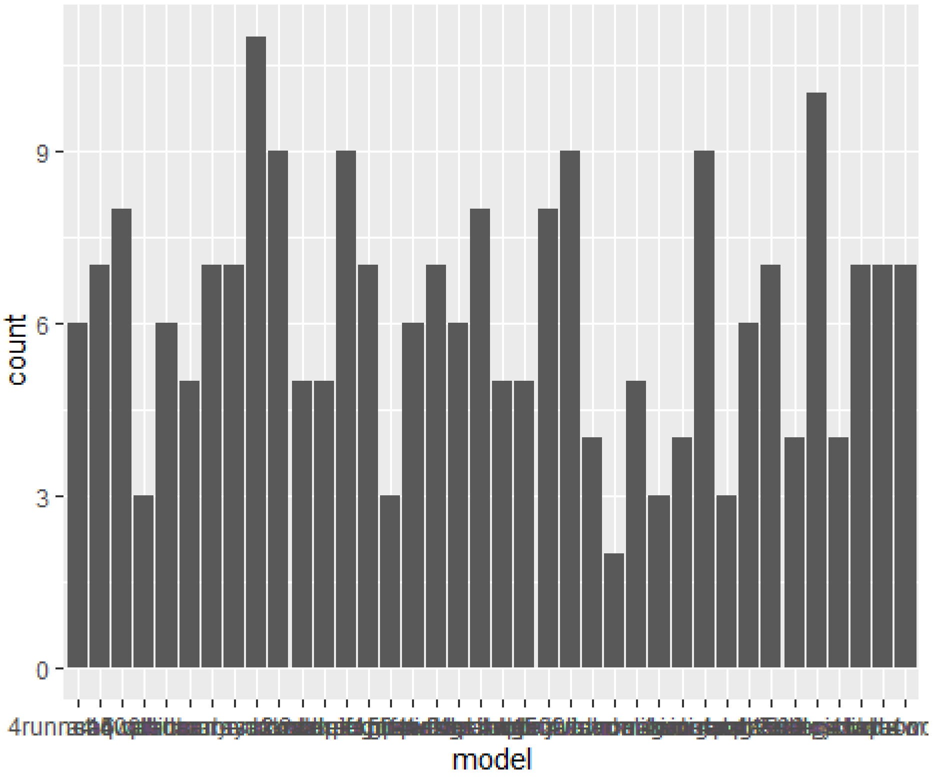
```
#Grouped bar chart
ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +
  geom_bar(position = 'dodge') +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Frequency") +
  theme_bw()
```



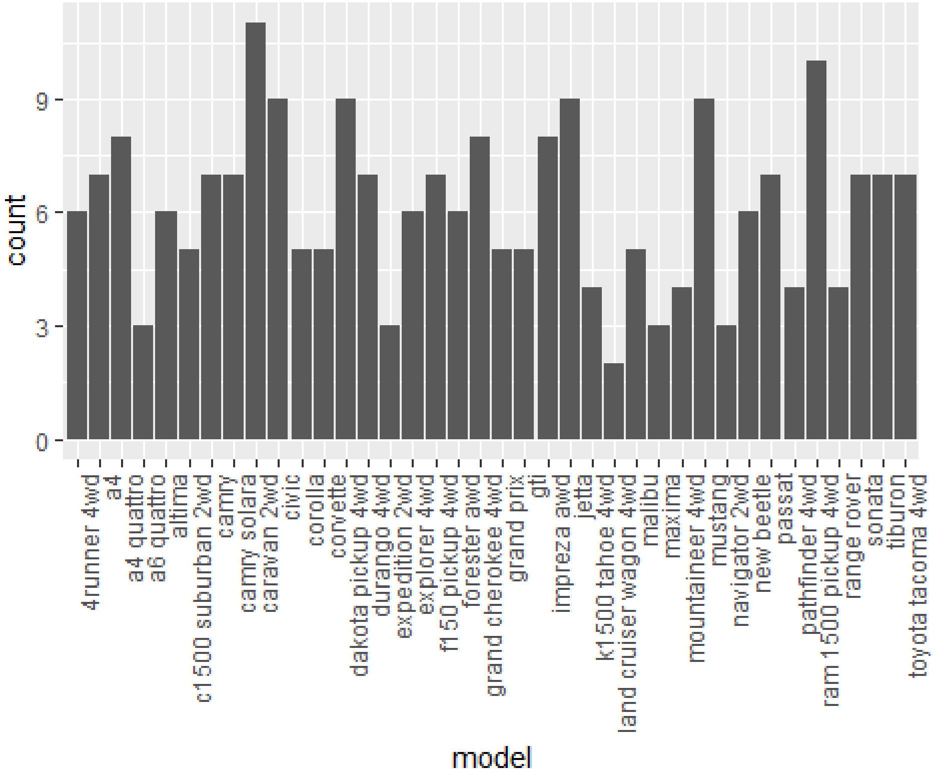
```
#Filled bar chart
ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +
  geom_bar(position = 'fill') +
  labs(title="Simple Bar chart",
       x="Improvement",
       y="Proportion") +
  theme_bw()
```



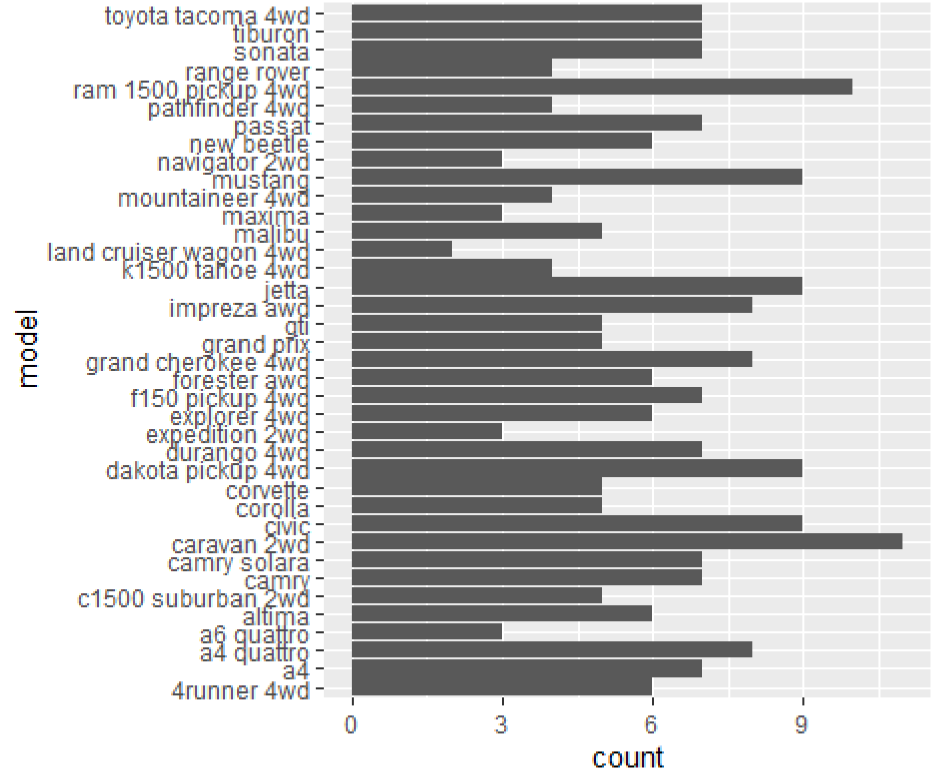
```
#managing congested labels
ggplot(mpg, aes(x=model)) +
  geom_bar() #produce congested labels
```



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



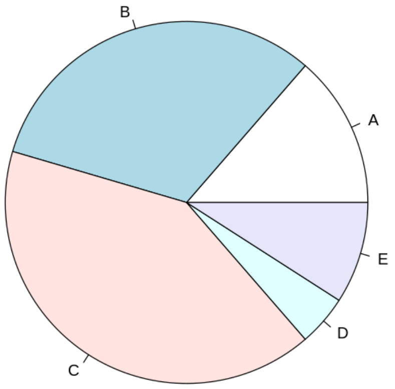
```
#OR
ggplot(mpg, aes(x=model)) +
  geom_bar() +
  coord_flip()
```



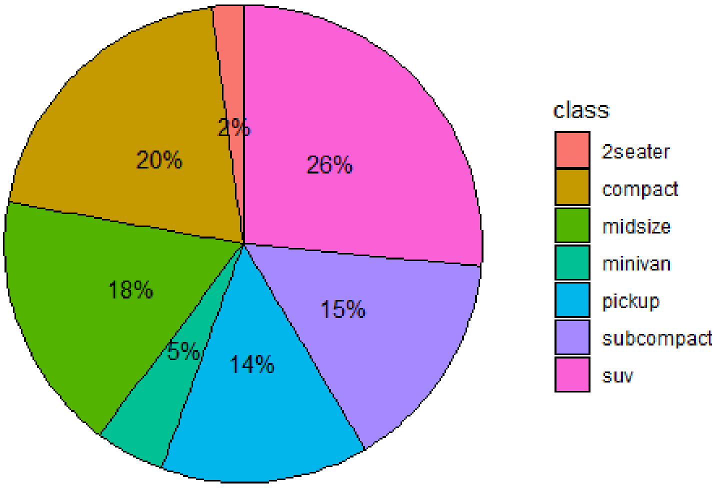
Pie charts

E021-pie_chart.R

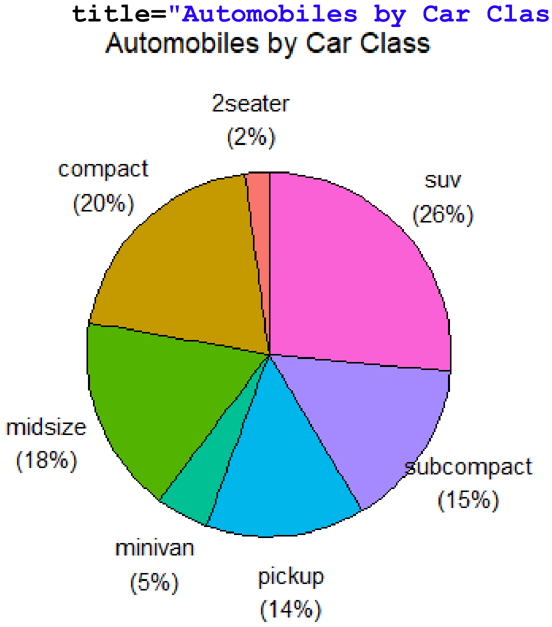
```
#*-------------------------------------------------
#* Pie Chart using inbuilt package graphics
#*-------------------------------------------------
pie(x = c(3,7,9,1,2),
    labels = c("A","B","C","D","E"))
```



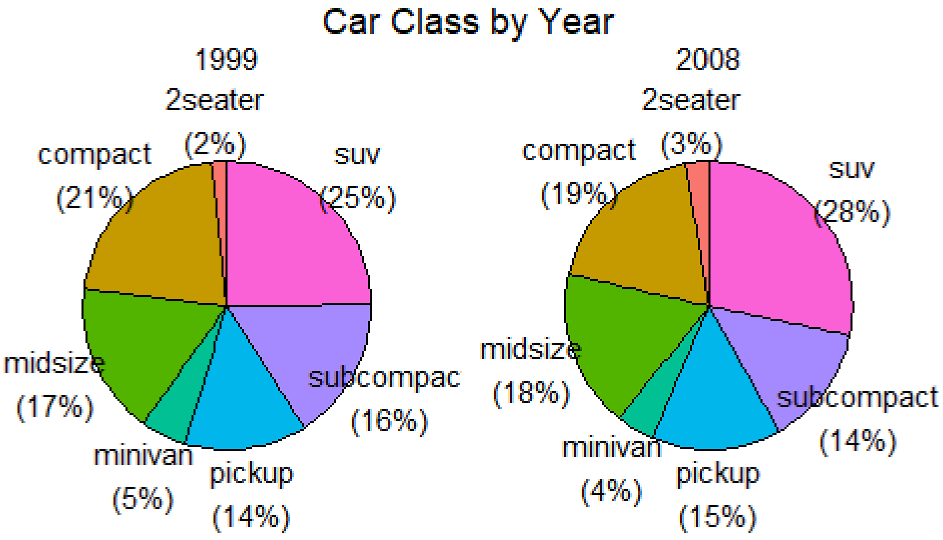
```
#*-------------------------------------------------
#* Pie Chart using package ggpie
#*-------------------------------------------------

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

library(ggplot2)
library(ggpie)

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



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



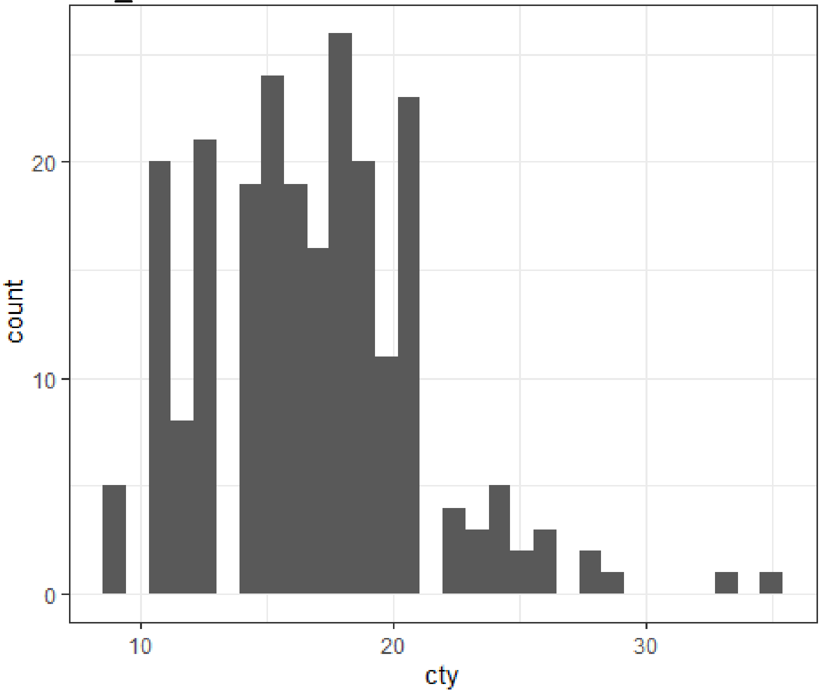
```
# group wise pie charts
ggpie(mpg, class, year,
      legend=FALSE, offset=1.3, title="Car Class by Year")
```



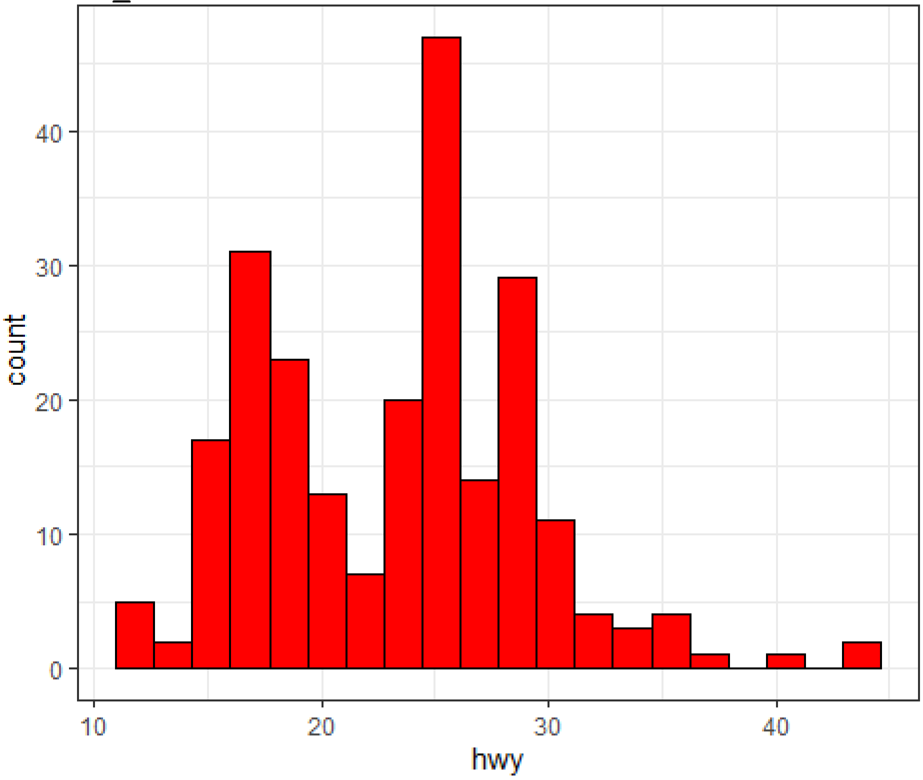
Histograms

E022-histogram.R

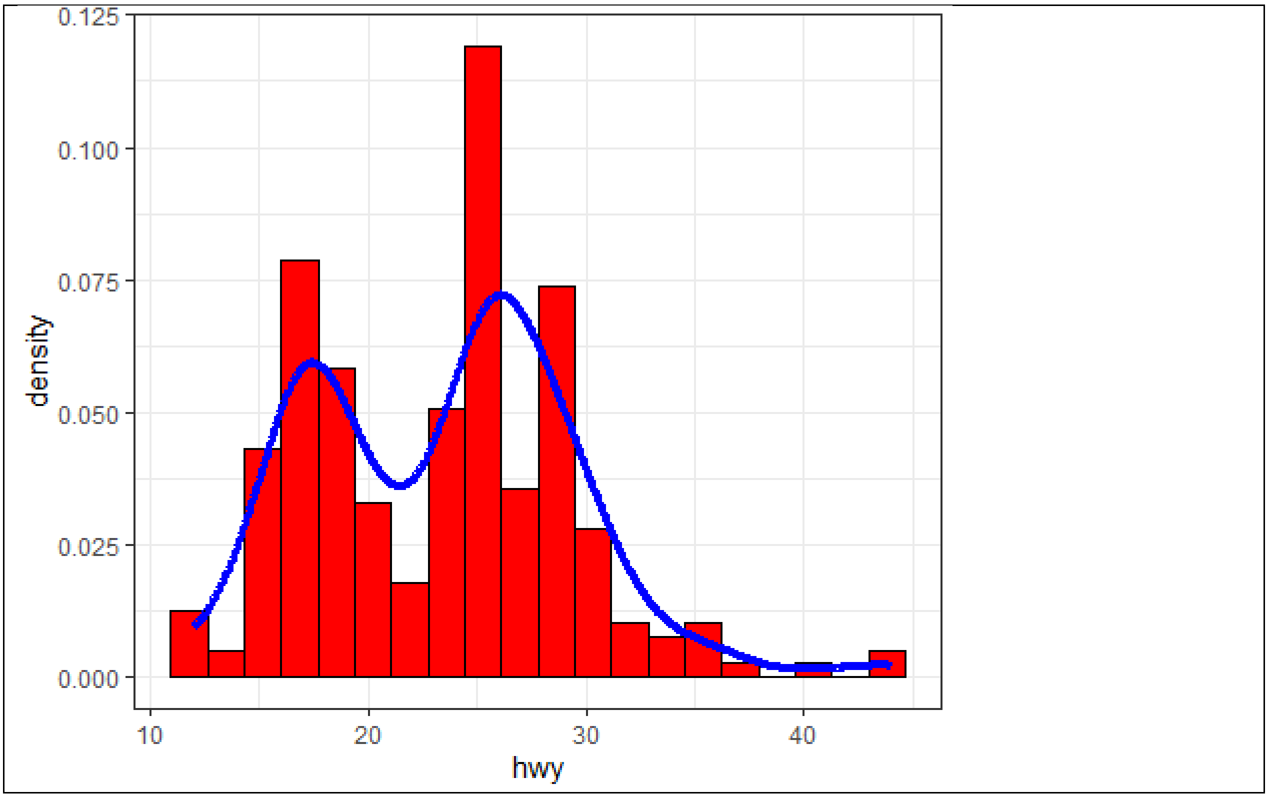
```
library(ggplot2)
library(dplyr)
data(mpg)

#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()
```

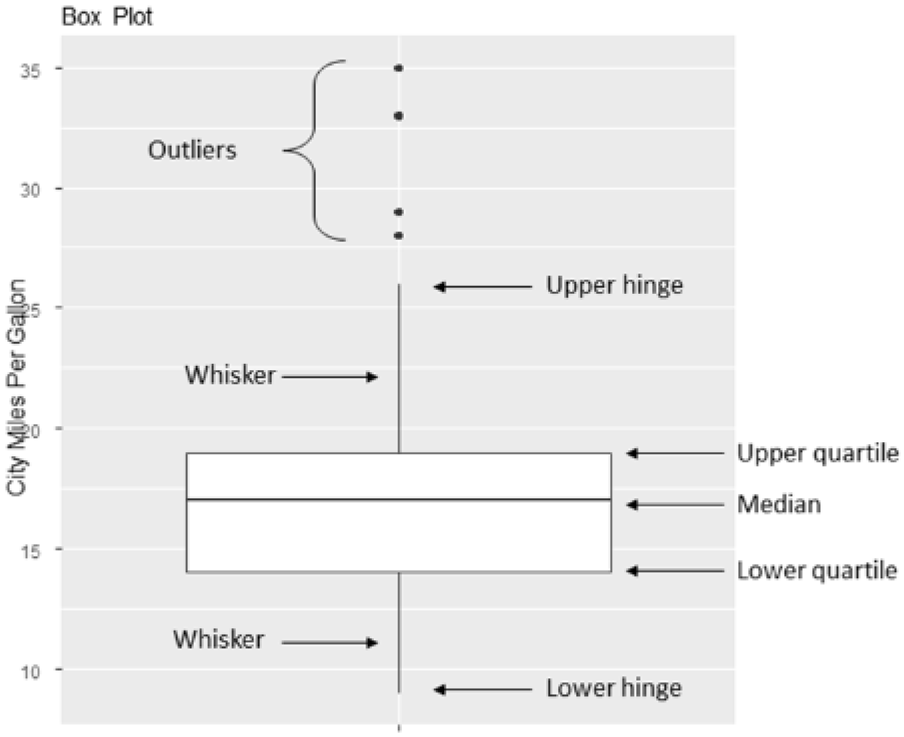


```
#Histogram with density cruve
ggplot(mpg, aes(x=hwy, y = ..density..)) +
  geom_histogram(bins=20, fill="red", color = 'black') +
  geom_density(color = 'blue', size = 1.5) +
  theme_bw()
```



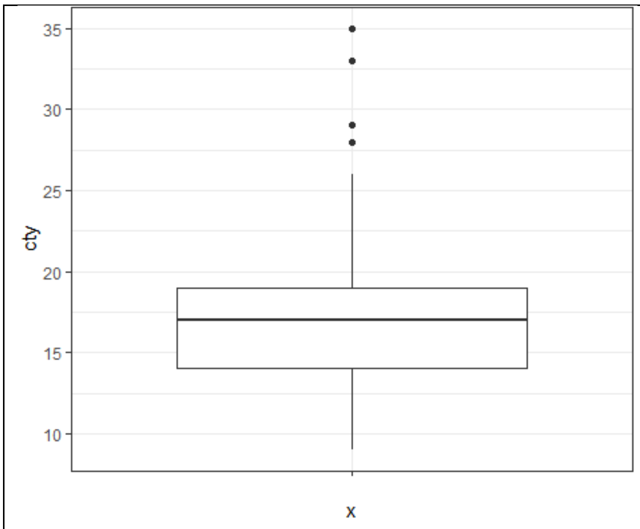
Box plots

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).

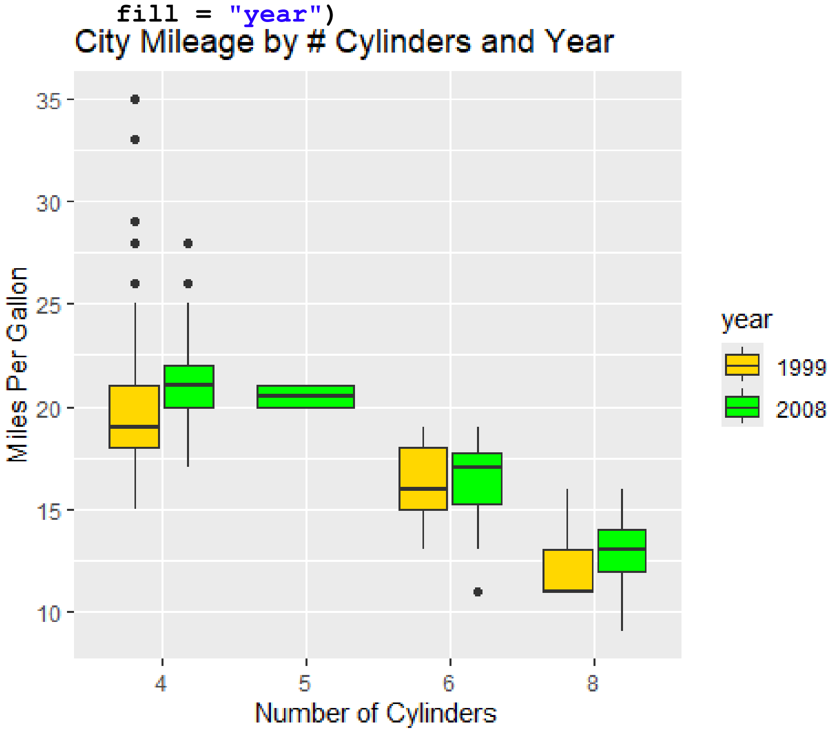


E023-boxplot.R

```
library(ggplot2)

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



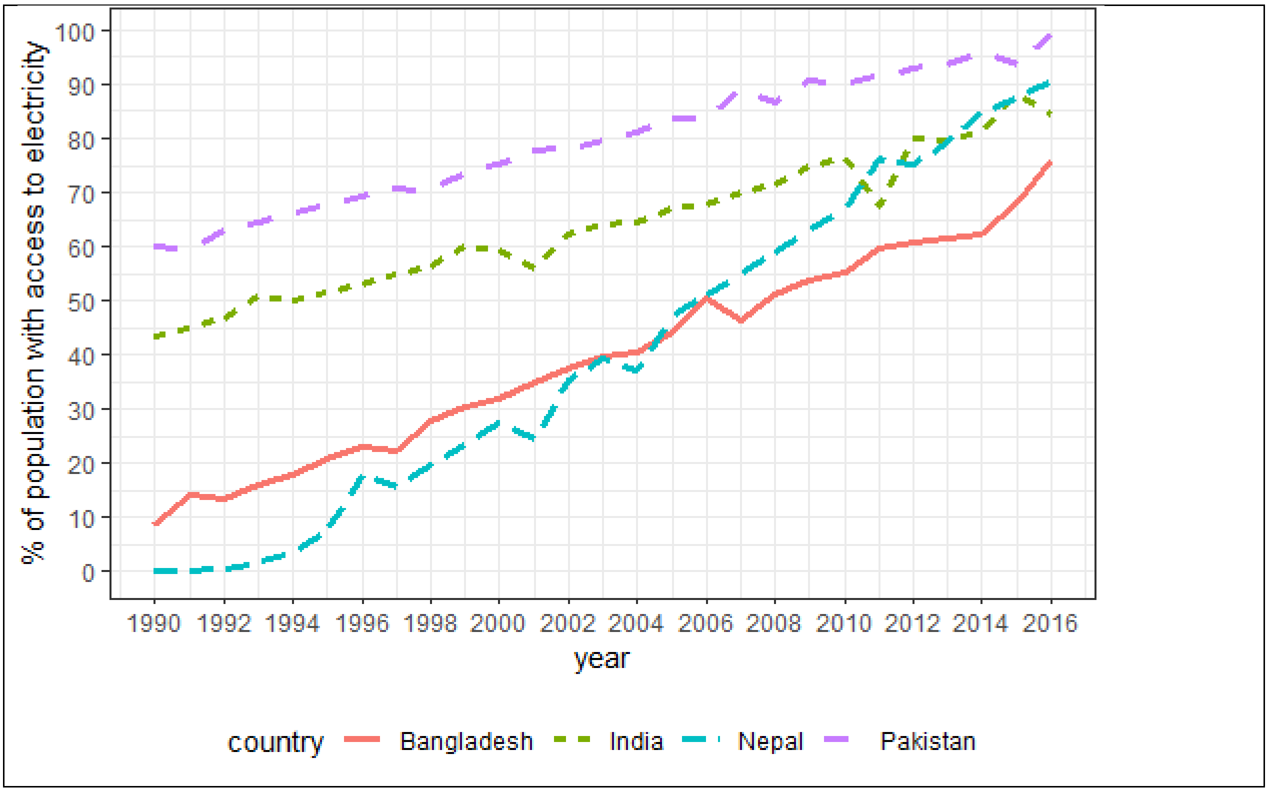
```
ggplot(mpg, aes(x=factor(cyl), y=cty, fill=factor(year))) +
  geom_boxplot() +
  scale_fill_manual(values=c("gold", "green")) +
  labs(x="Number of Cylinders",
       y="Miles Per Gallon",
       title="City Mileage by # Cylinders and Year",
       fill = "year")
```



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.

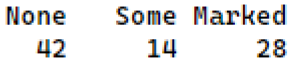
```
vec <- c(3,5,2,3,4,2,5,6,7)
mean(vec) # 4.111111
sd(vec) # 1.763834

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: Cross tabulation

7.1 Frequency tables

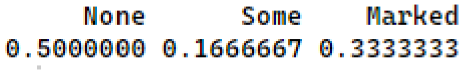
E032-frequency_contingency_tables.R

```
Arthritis <- vcd::Arthritis

#simple frequency table
mytable <- table(Arthritis$Improved)
mytable
```



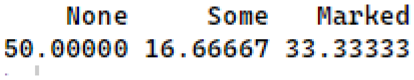
```
 None    Some Marked 
   42      14      28 
```

```
#proportion table
prop.table(mytable)
```



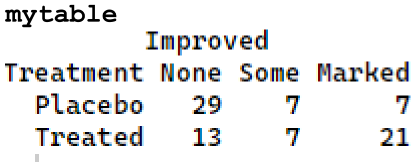
```
     None      Some    Marked 
0.5000000 0.1666667 0.3333333 
```

```
prop.table(mytable)*100 #in percentage
```



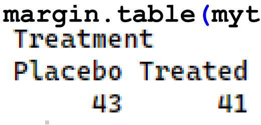
```
    None     Some   Marked 
50.00000 16.66667 33.33333 
```

```
#-------------------------------------
# Two-way table
#-------------------------------------
mytable <- xtabs(~ Treatment + Improved, data=Arthritis)
mytable
```



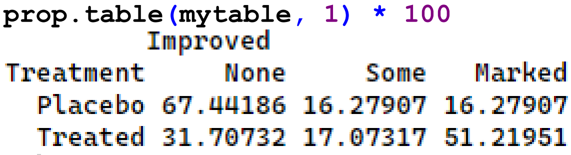
```
         Improved
Treatment None Some Marked
  Placebo   29    7      7
  Treated   13    7     21
```

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



```
Treatment
Placebo Treated 
     43      41 
```

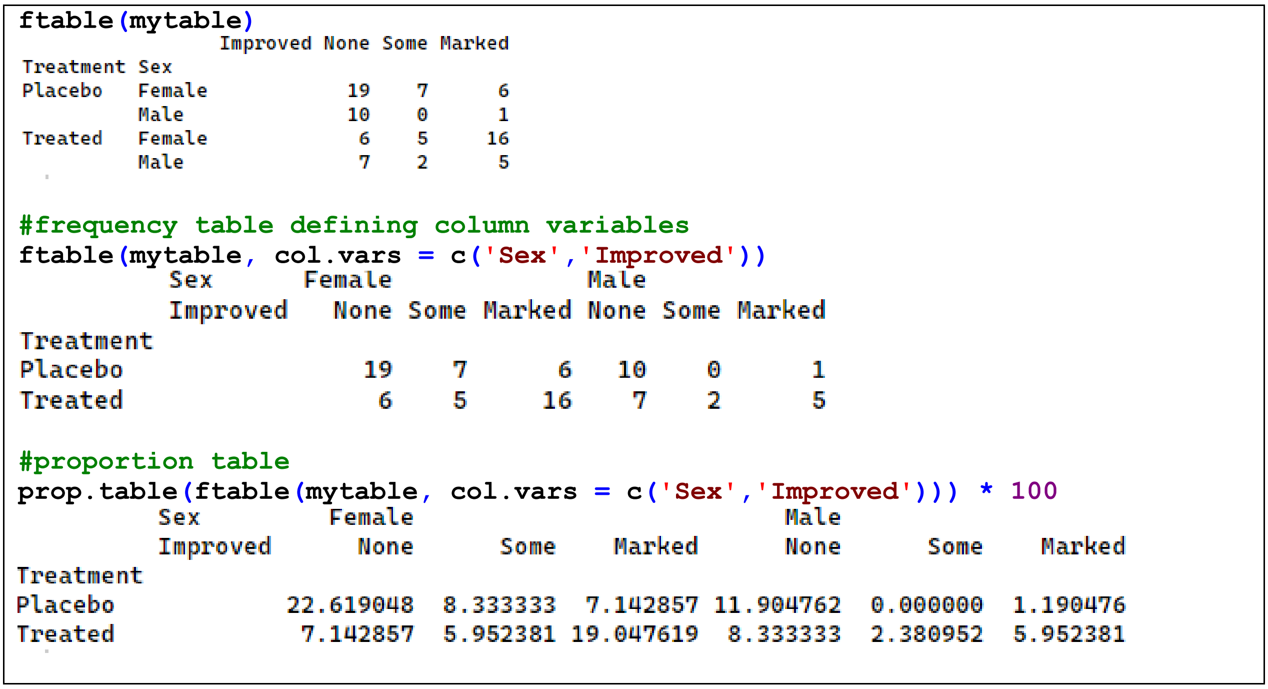
```
#proportion table based on horizontal sub-total
prop.table(mytable, 1) * 100
```



```
         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
```