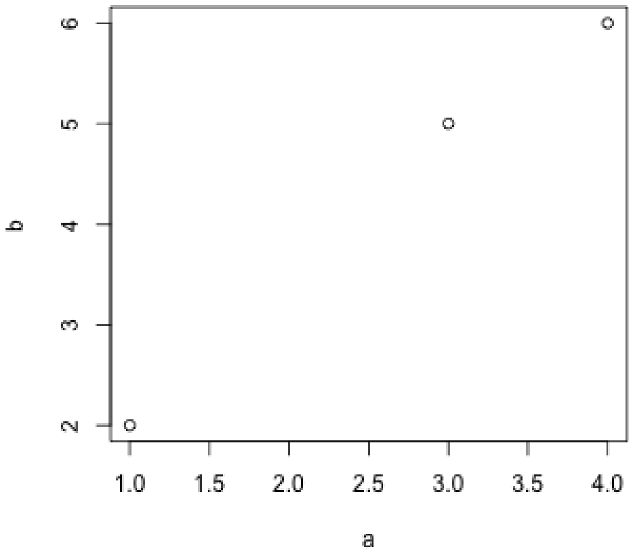
Graph in R

Outline

- Scatter plot
- Histogram
- Bar plot and pie chart
- Boxplot
- qqnorm, qqline, qqplot (distribution comparision)
- ggplot
- Heat Map

1.Plot

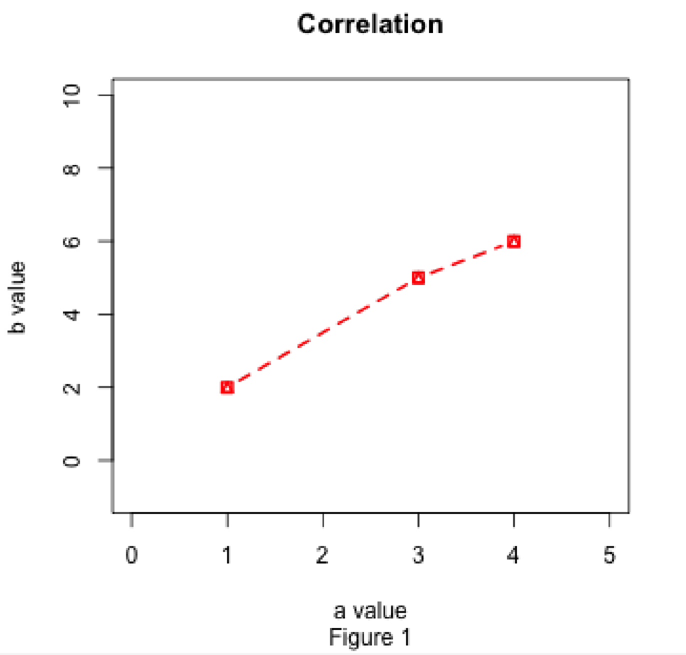
```
a <- c(1, 3, 4)
b <- c(2, 5, 6)
plot(x = a, y = b)
```



Arguments: x and y are coordinates of points in the plot

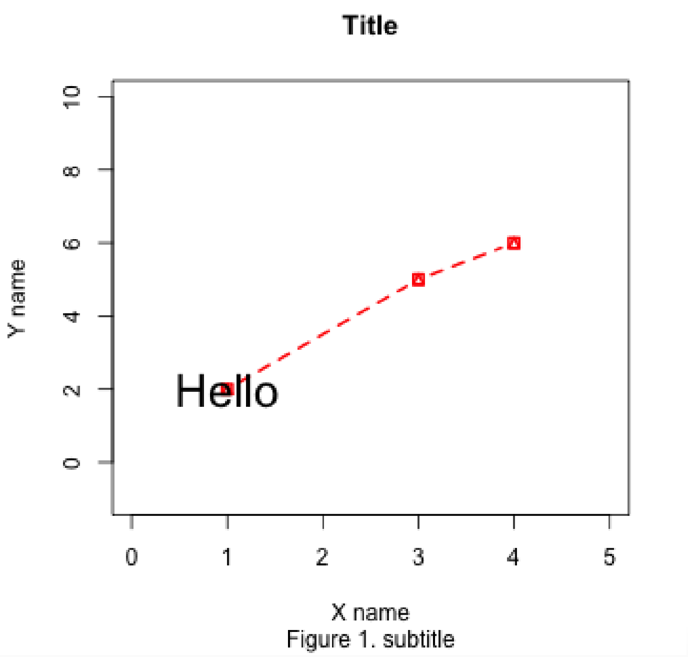
(1) Basic arguments of plot()

```
plot(x = a, y = b,
     type = "b",                              # Type of plot: (points, lines, both, none)
     lty = 2, lwd = 2, col = "red",           # Line: (type, width, color)
     pch = 14,                                # Point: (shape)
     main = "Correlation", sub = "Figure 1",  # Figure: (title; subtitlel)
     xlab = "a value", ylab = "b value",      # x,y axis: (label)
     xlim = c(0,5), ylim = c(-1,10))          # x,y axis: (range)
```



(2) How to add text?

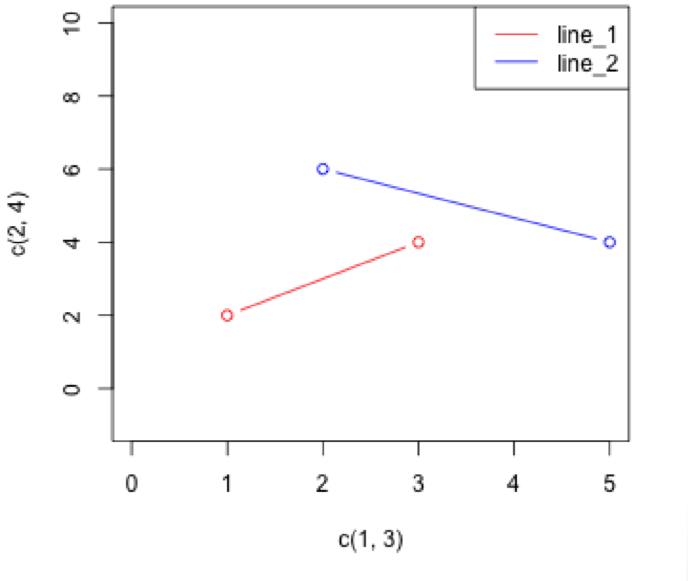
```
plot(x = a, y = b, type = "b", lty = 2, lwd = 2, col = "red", pch = 14,
     main = "Title", sub = "Figure 1. subtitle",
     xlab = "X name", ylab = "Y name", xlim = c(0,5), ylim = c(-1,10))
# add text
# x: coordinates, y: coordinates, labels: text, cex: text size
text(x = a[1], y = b[1], labels = "Hello", cex = 2)
```



(3) How to add new line & legend?

- lines() / legend()

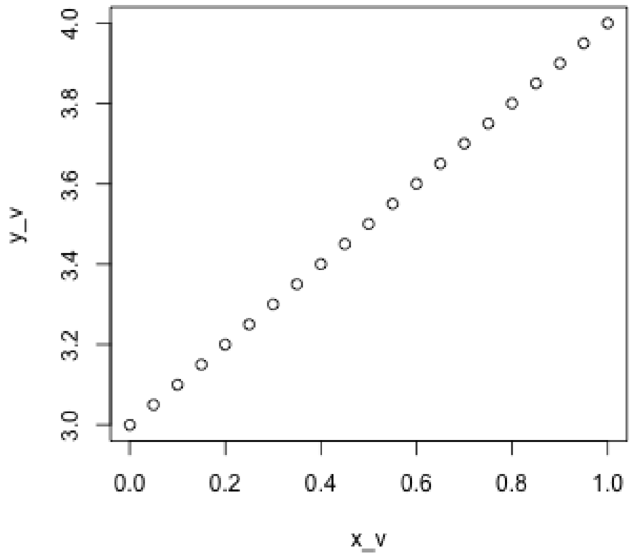
```
plot(x = c(1,3), y = c(2,4), col = "red", type = "b", xlim = c(0,5), ylim = c(-1,10))
lines(x = c(2,5), y = c(6,4), col = "blue", type = "b", xlim = c(0,5), ylim = c(-1,10))

# add legend
legend( x = "topright", col = c("red", "blue"), lty = c(1,1), legend = c("line_1","line_2"))
```



(4) How to set a sequence of values?

- seq()

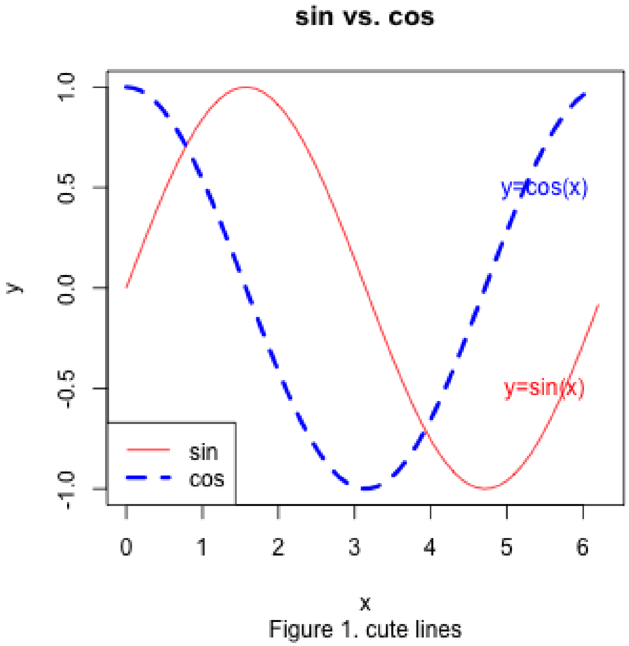
```
x_v <- seq(from = 0, to = 1, by = 0.05) # by:increment of the sequence.
y_v <- x_v + 3
plot(x = x_v, y = y_v )
```



Exercise

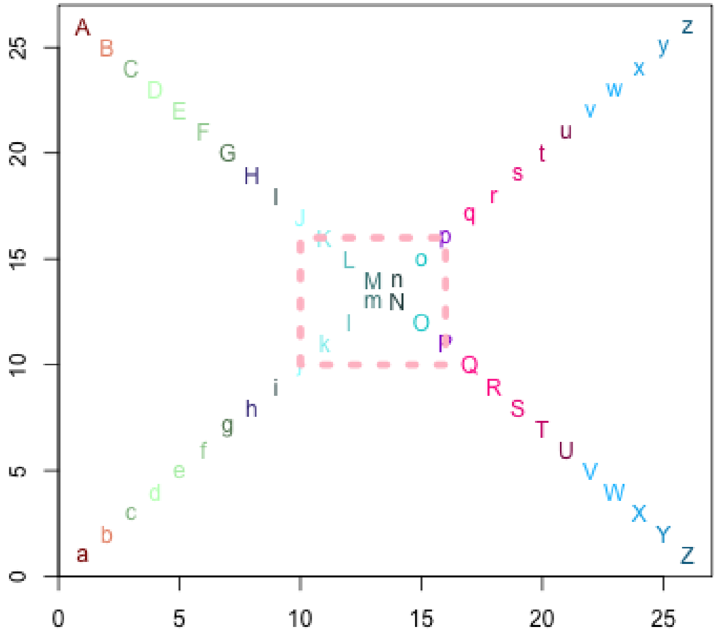
Plot the following two lines:

- y = cos(x): blue, dashed, thick line
- y = sin(x): red, solid, thin line
- Add title, subtitle, legend, functions as below



Exercise

Draw the figure happily

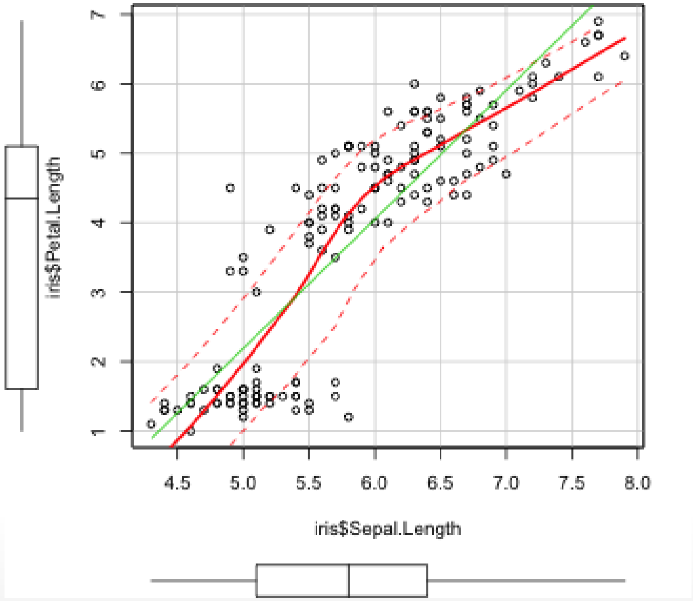


(5) Linear regression

Remove OLS regression line? (reg.line = FALSE)

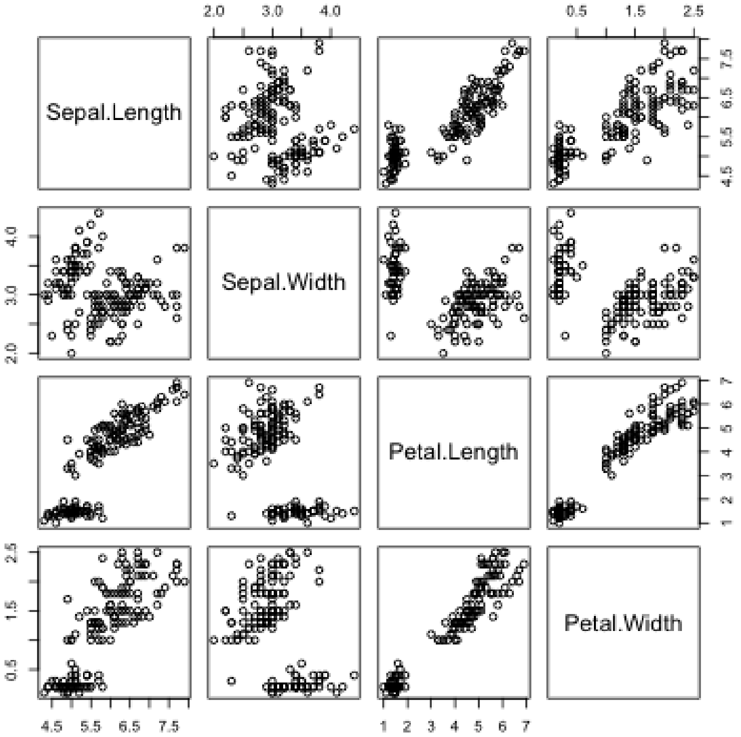
Remove non-par smooth line? (smooth = FALSE)

```
# library(plotly)
# plot_ly(x = iris$Sepal.Length, y = iris$Petal.Length)
library(car)
scatterplot(x = iris$Sepal.Length, y = iris$Petal.Length)
```



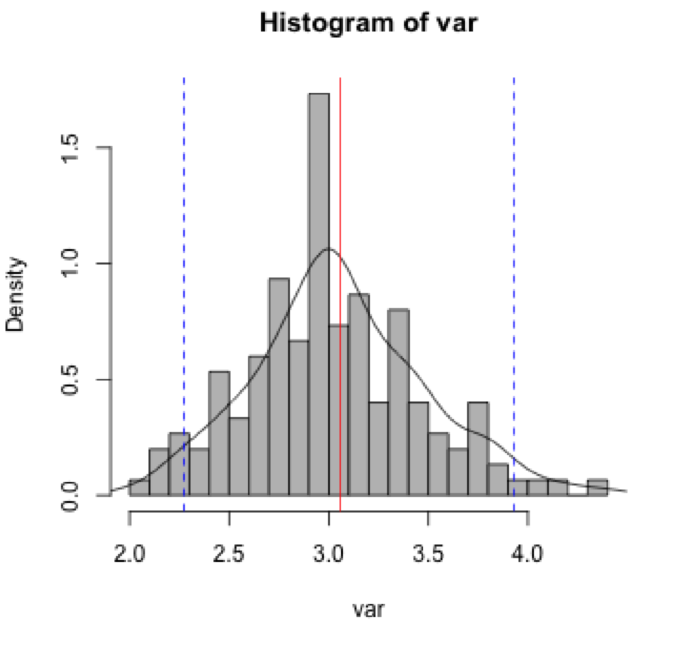
(6) Scatter plot between any two variables - pairs()

```
#pairs(x = iris[,1:4])
pairs(formula = ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = iris)
```



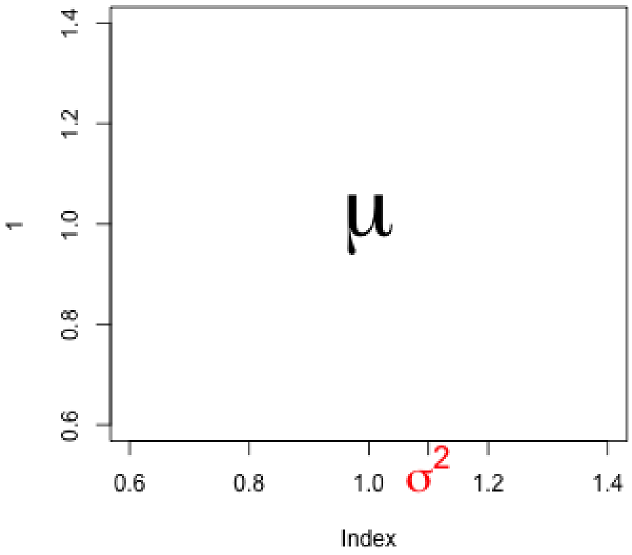
2. Histogram (examine distribution)

```
var <- iris$Sepal.Width
hist(x = var, breaks = 20, col = "gray", freq = F)
lines(density(var))
abline(v = mean(var), col = "red")
abline(v = quantile(var, c(0.025,0.975)), lty = 2, col = "blue")
```



- Add Math formular and axis note

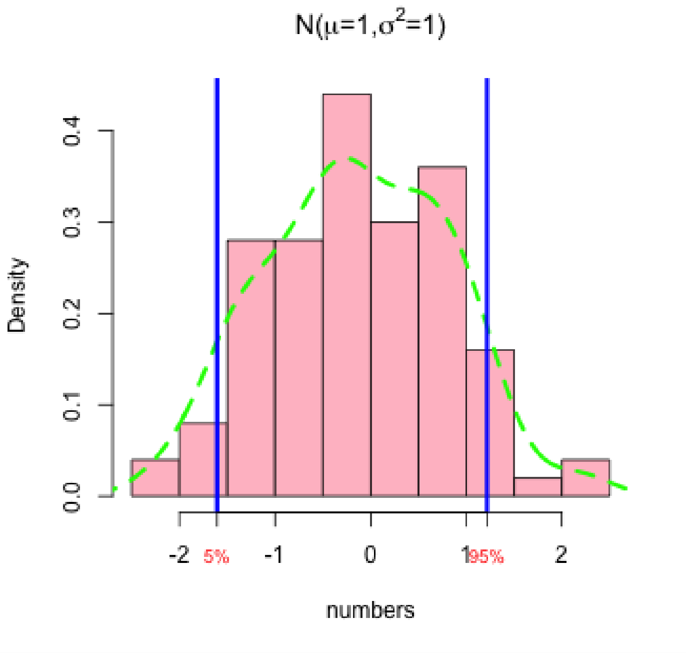
```
plot(1, type = "n")
text(x = 1, y = 1, labels = expression(mu), cex = 4)
axis(side = 1, at = 1.1, labels = expression(sigma^2), cex.axis = 2, col.axis = "red")
```



Exercise

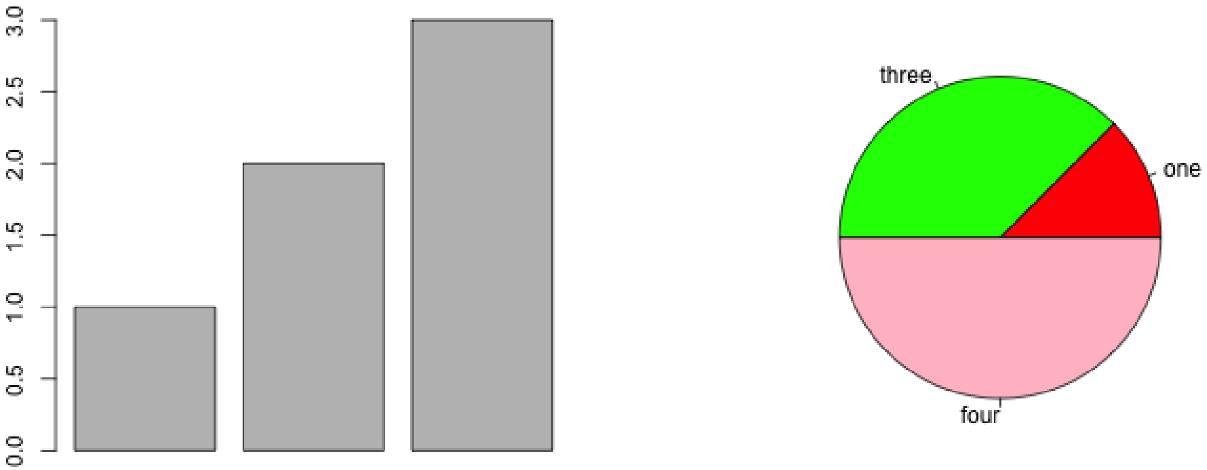
Plot histogram of variable "numbers"; add density line; mark its 5% and 95% quantile as follows.

```
set.seed(10)
numbers <- rnorm(n = 100, mean = 0, sd = 1)
```



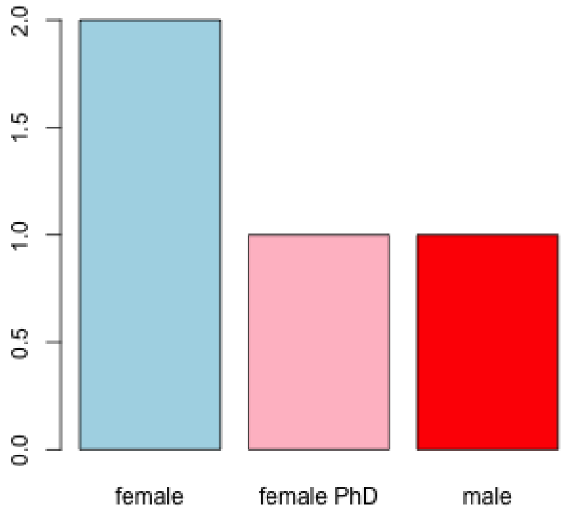
3. Bar plot and pie chart

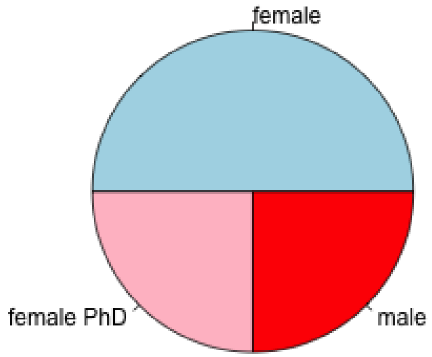
```
par(mfrow = c(1,2)) #layout of multiple figures
barplot(height = c(1,2,3))
pie(x = c(1,3,4),
    label = c("one", "three","four"), col = c("red","green","pink"))#areas of pie slices
```



Exercise

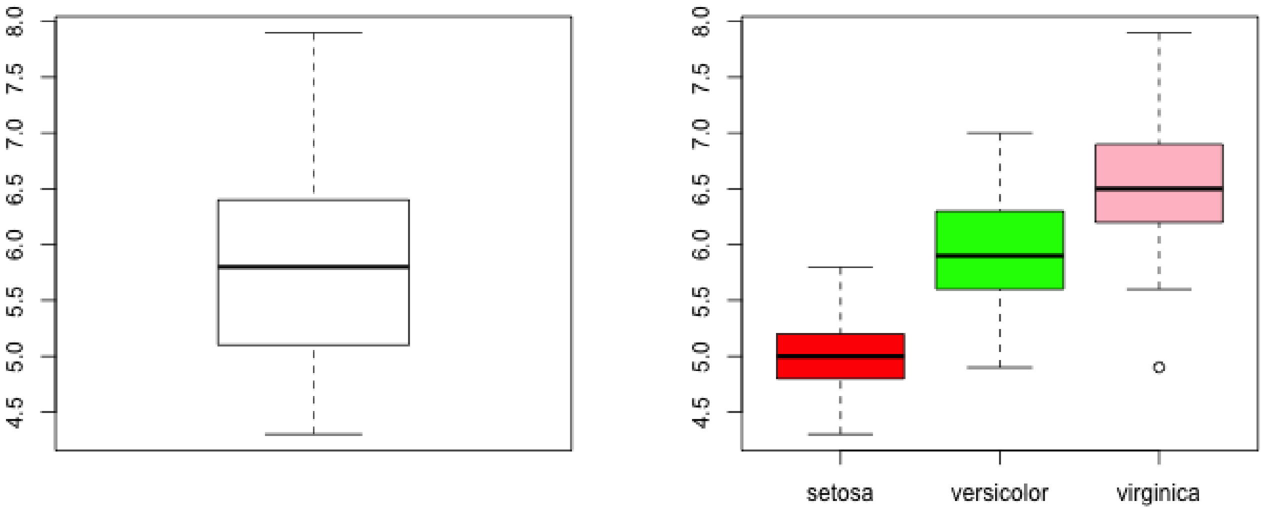
```
gender <- c("male","female","female","female PhD")
tb_g <- table(gender)
```





4. boxplot

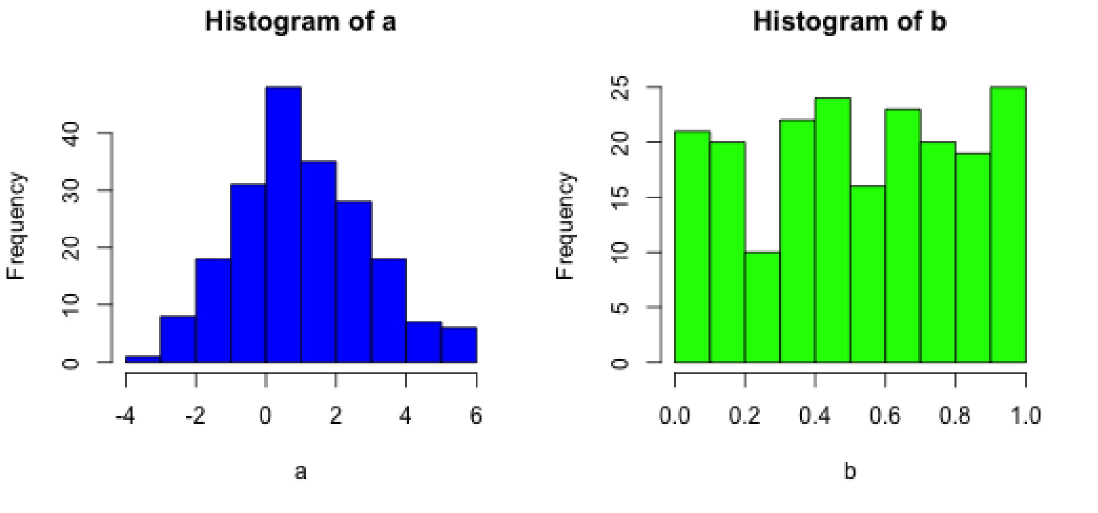
```
par(mfrow = c(1,2))
boxplot(iris$Sepal.Length)
boxplot(Sepal.Length ~ Species , data = iris, col = c("red","green","pink"))
```



5. QQ norm and QQ plot

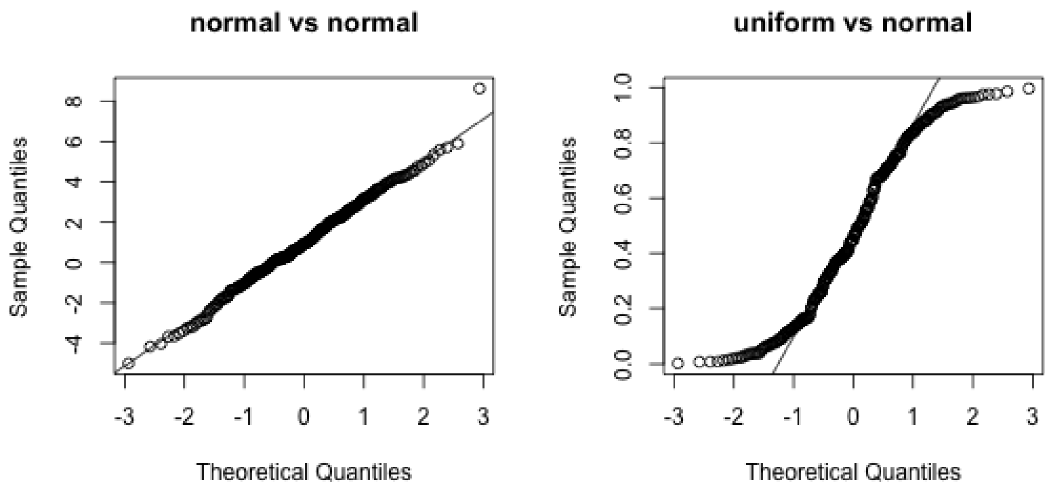
- Generate random numbers

```
par(mfrow = c(1,2))
set.seed(1)
a <- rnorm(n = 200, mean = 1, sd = 2) # normal distribution
b <- runif(n = 200, min = 0, max = 1) # uniform distribution
hist(a, col = "blue")
hist(b, col = "green")
```



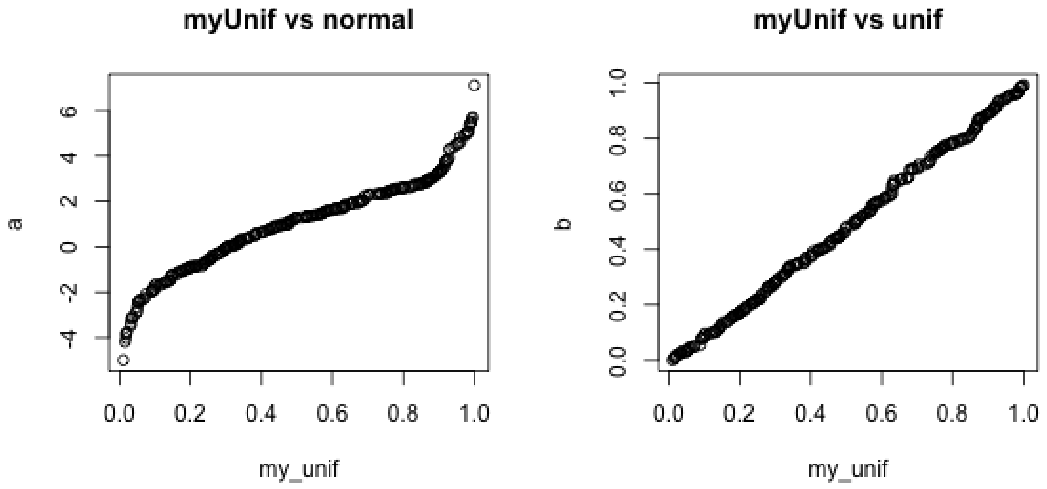
- QQ norm

```
par(mfrow = c(1,2))
a <- rnorm(n = 300, mean = 1, sd = 2)
b <- runif(n = 300, min = 0, max = 1)
qqnorm(a, main = "normal vs normal"); qqline(a)
qqnorm(b, main = "uniform vs normal"); qqline(b)
```



- QQ plot

```r
par(mfrow = c(1,2))
a <- rnorm(n = 300, mean = 1, sd = 2)
b <- runif(n = 300, min = 0, max = 1)

my_unif <- runif(n = 300, min = 0, max = 1)
qqplot(my_unif,a, main = "myUnif vs normal" )
qqplot(my_unif,b, main = "myUnif vs unif" )
```



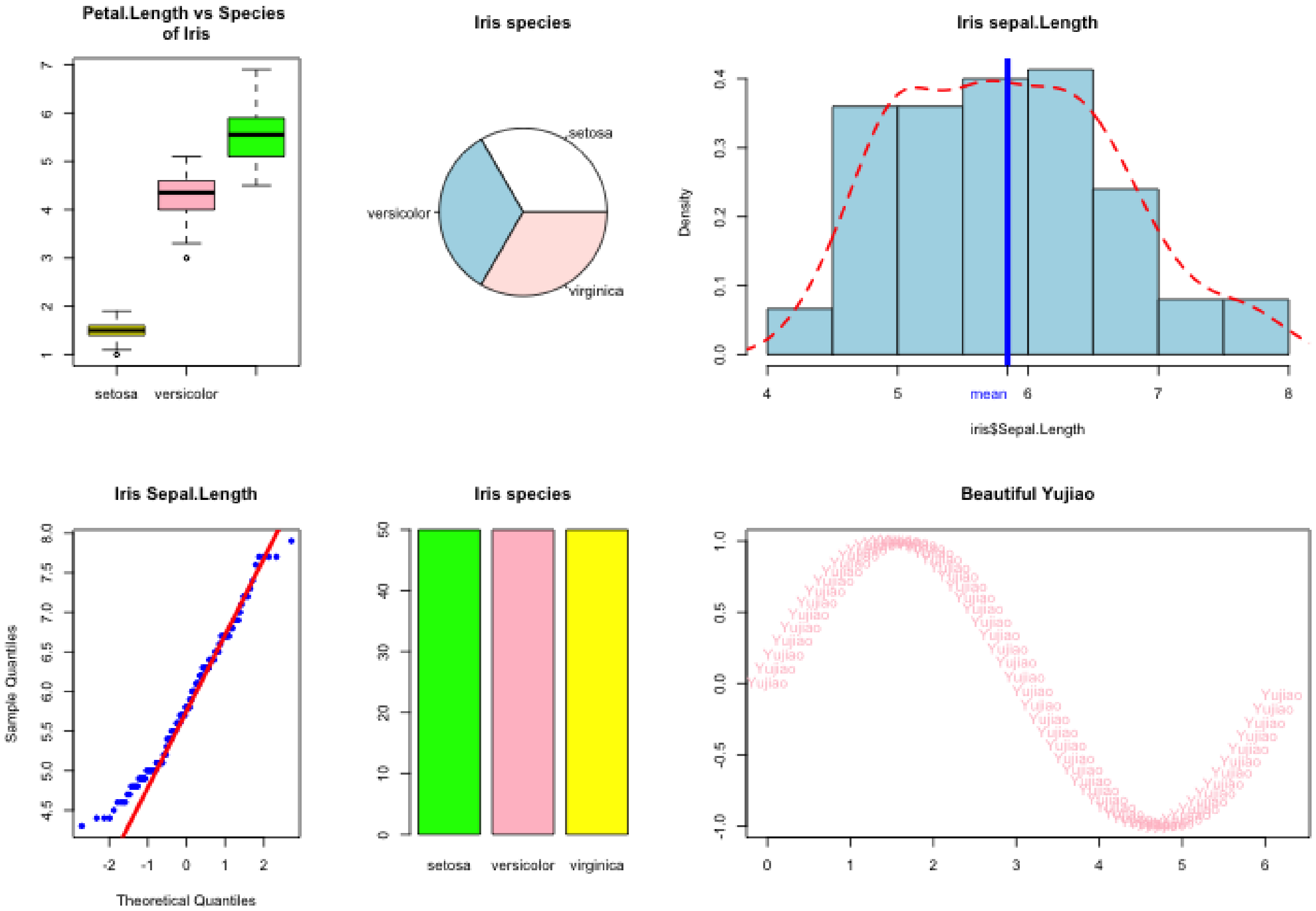
Saving Graphics to Files

After the pdf() command, graphs are redirected to file test.pdf.

```
pdf(file = "test.pdf"); plot(1); dev.off()
#  Works for all common formats similarly: jpeg, png, ps, tiff, ...
jpeg(filename = "test.jpeg"); plot(2); dev.off()
getwd() # check working file path
dir() # show files in current path
setwd() # change working file path
dir.create() # create new folder
file.remove() # remove files
```

Exercise

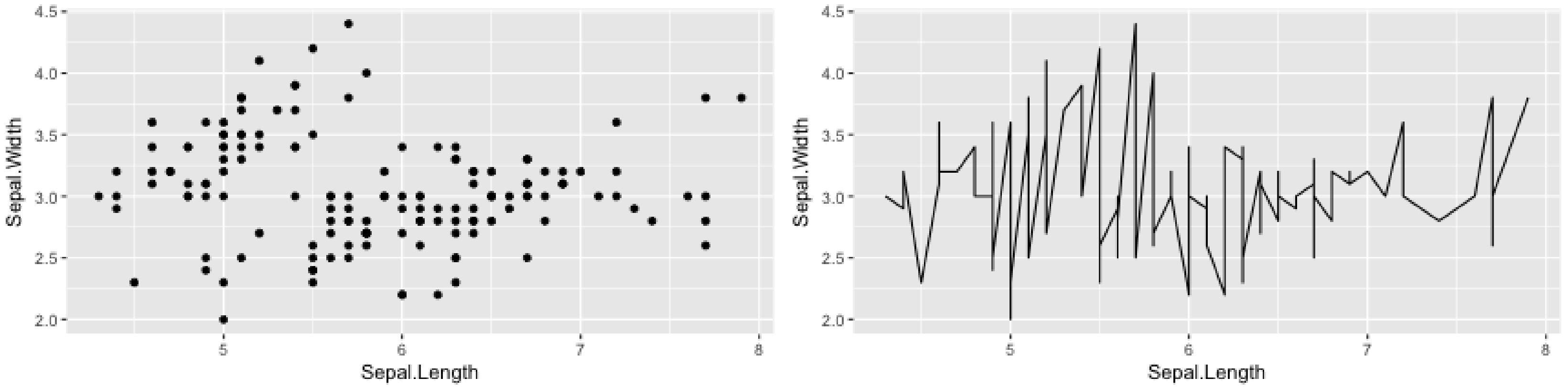
- Plot the figure below and save it as a "myplot.jpeg" file in your desktop



6. ggplot

- Plots can be assembled in pieces

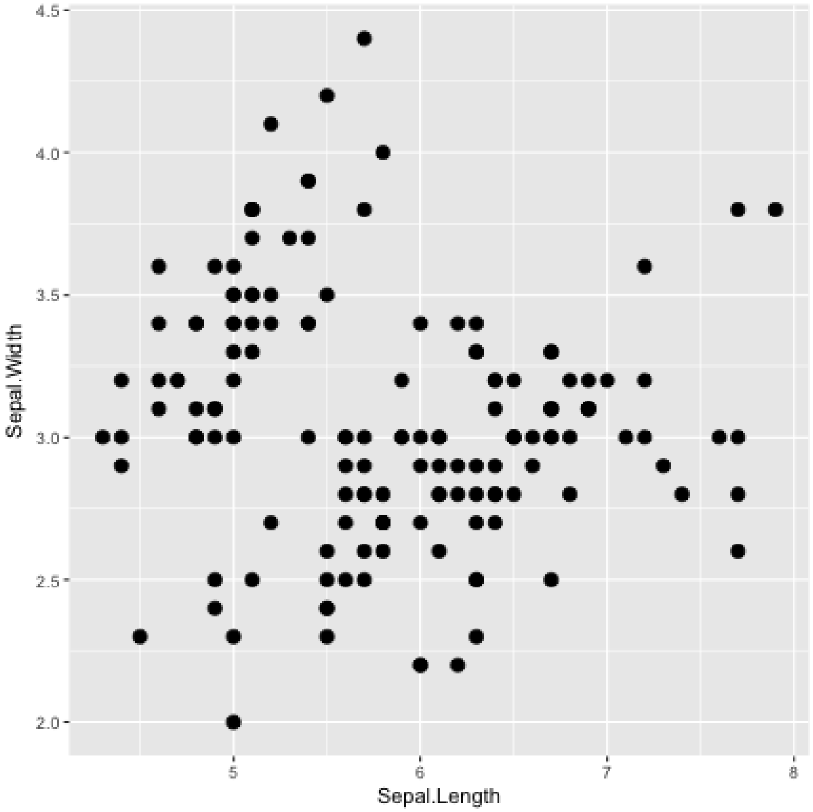
```
library(ggplot2)
baseplot <- ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))
p1 <- baseplot + geom_point()
p2 <- baseplot + geom_line()
#ggsave("p2.jpeg", p2)
library(gridExtra) #layout package for ggplots
p_all <- grid.arrange(p1, p2,   ncol = 2, nrow = 1)
```



Changing aesthetics of a geom

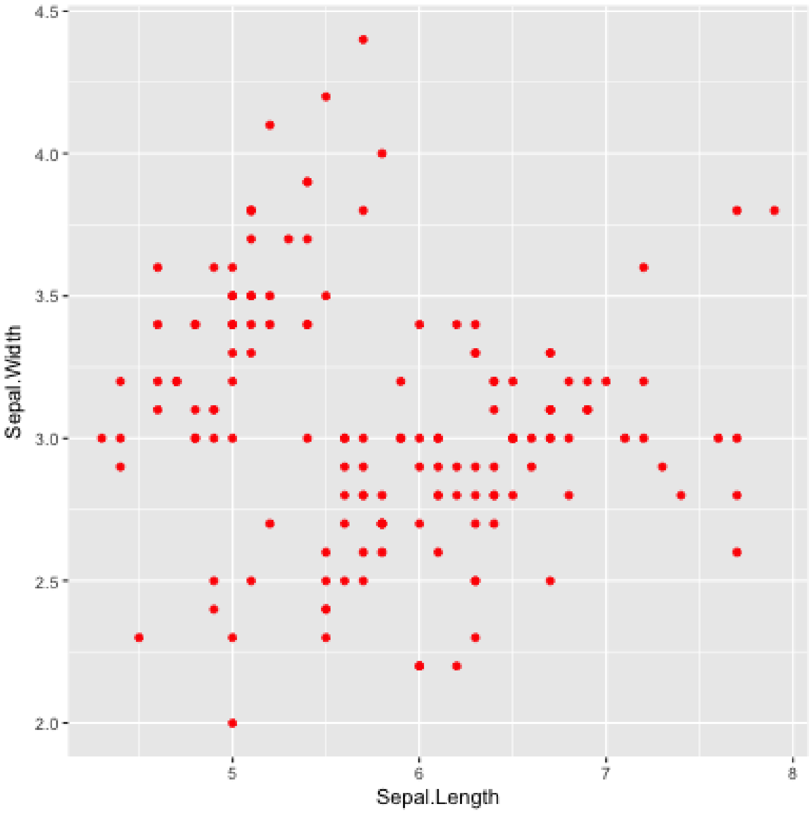
- point size

```
baseplot + geom_point(size = 3)
```



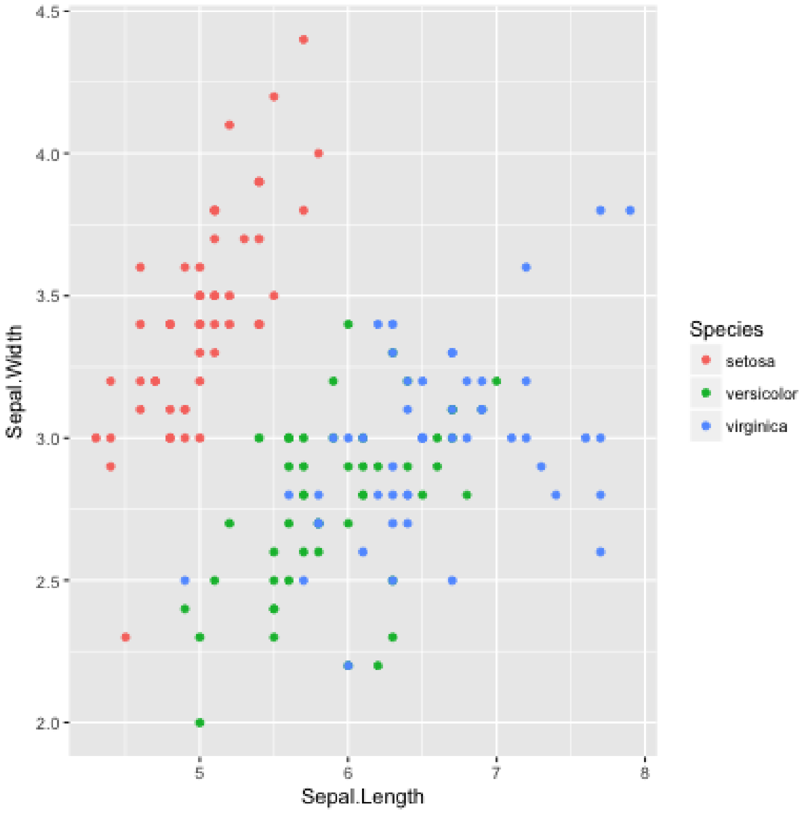
- point color

```
baseplot + geom_point(color = "red")
```



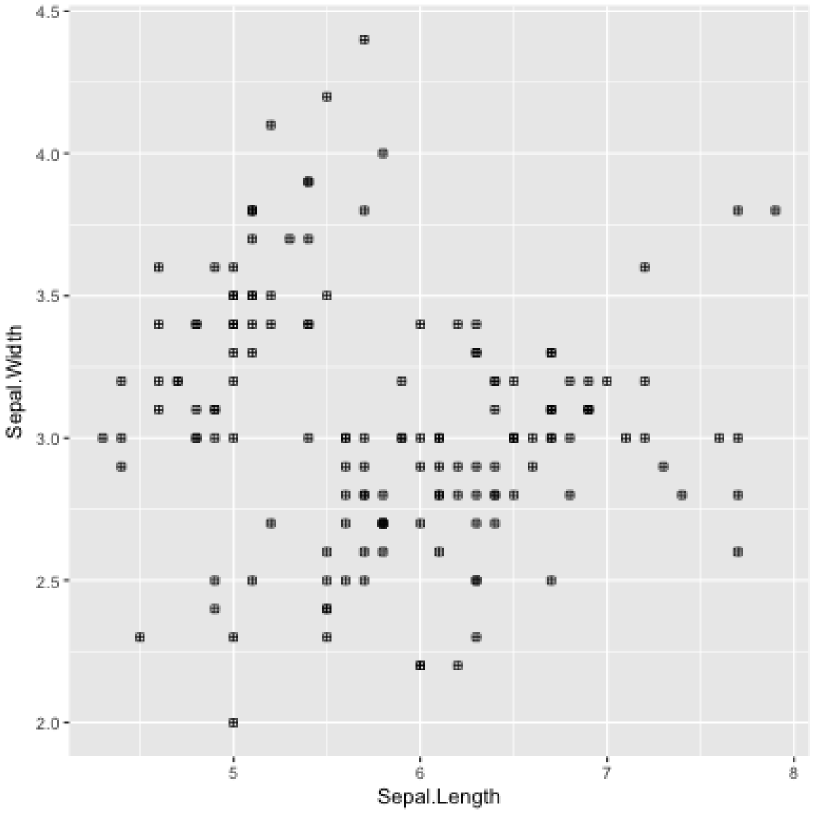
- point color

```
baseplot + geom_point(aes(color = Species))
```



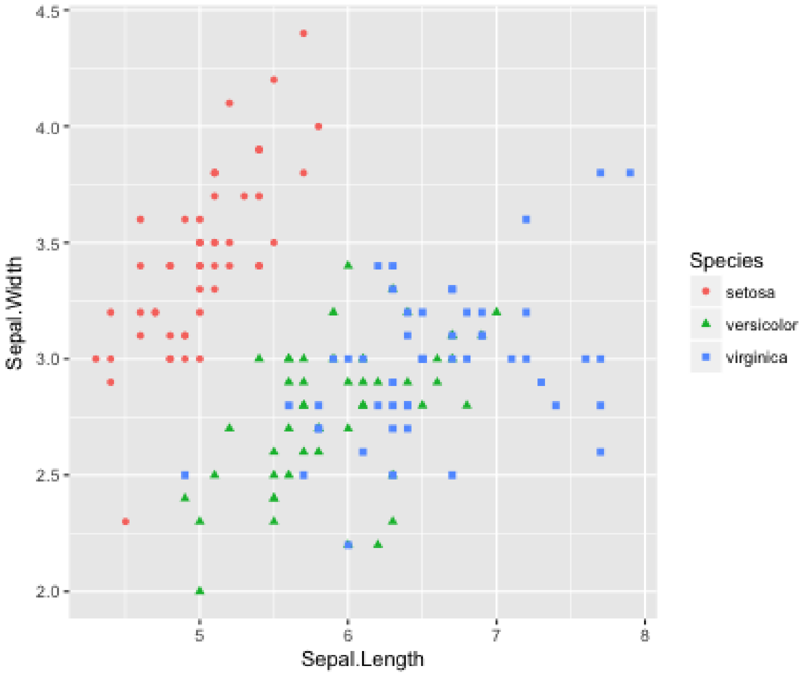
- point shape

```
baseplot + geom_point(shape = 12)
```



- Point shape & color varied with species

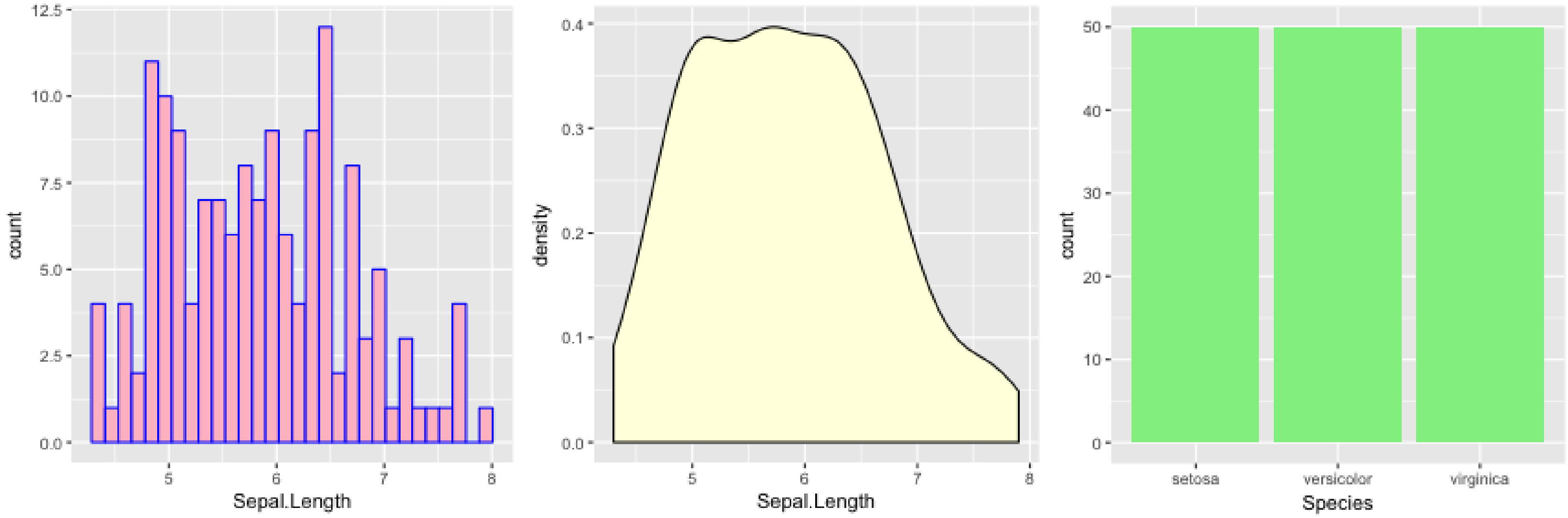
```
baseplot + geom_point(aes(color = Species, shape = Species))
```



Other geom functions:

- histogram: geom_histogram()
- bar plot: geom_bar()

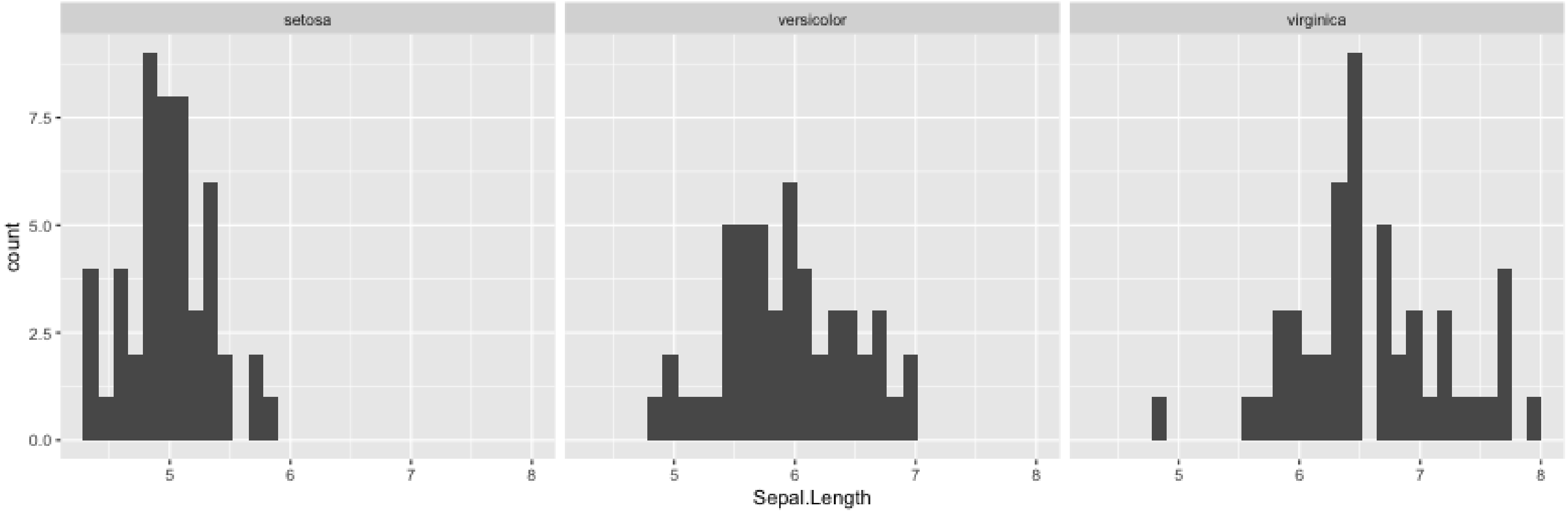
```
P1 <- ggplot(data = iris, aes(Sepal.Length)) + geom_histogram(color = "blue", fill = "pink")
P2 <- ggplot(data = iris, aes(Sepal.Length)) + geom_density(fill = "lightyellow")
P3 <- ggplot(data = iris, aes(Species)) + geom_bar(fill = "lightgreen")
grid.arrange(P1, P2, P3,  ncol = 3, nrow = 1)
```



Facets

Plots can also have facets to make lattice plots

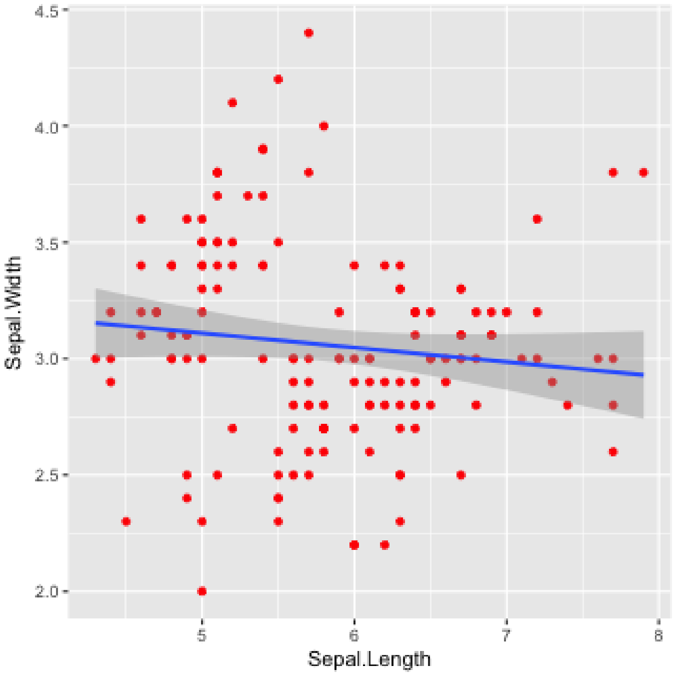
```
ggplot(iris, aes(Sepal.Length)) + geom_histogram() + facet_grid(.~ Species )
```



Adding smoother

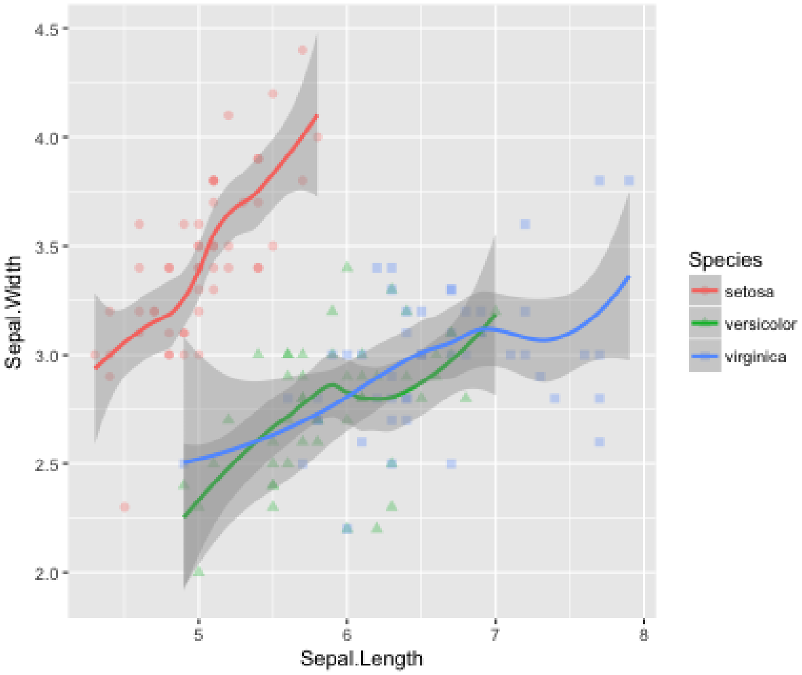
Use stat_smooth() to add a linear fit

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) + geom_point(color = "red") +
            stat_smooth(method = "lm")
```



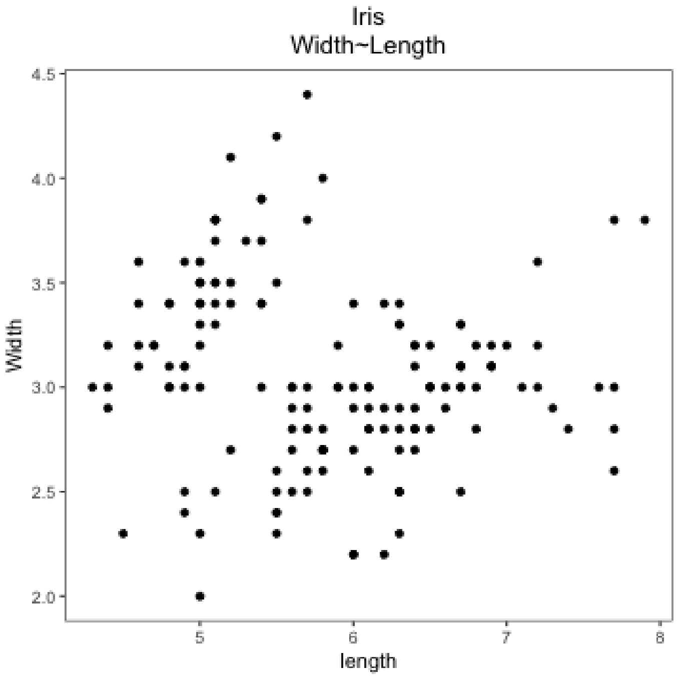
- Color different species

```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = Species))
            geom_point(size = 2, alpha = 0.3) + stat_smooth()
```



Add title and labels

```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) + geom_point() +
            labs(title = "Iris\nWidth~Length", x = "length", y = "Width") +
            theme_bw() +
            theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
            theme(plot.title = element_text(hjust = 0.5))
```

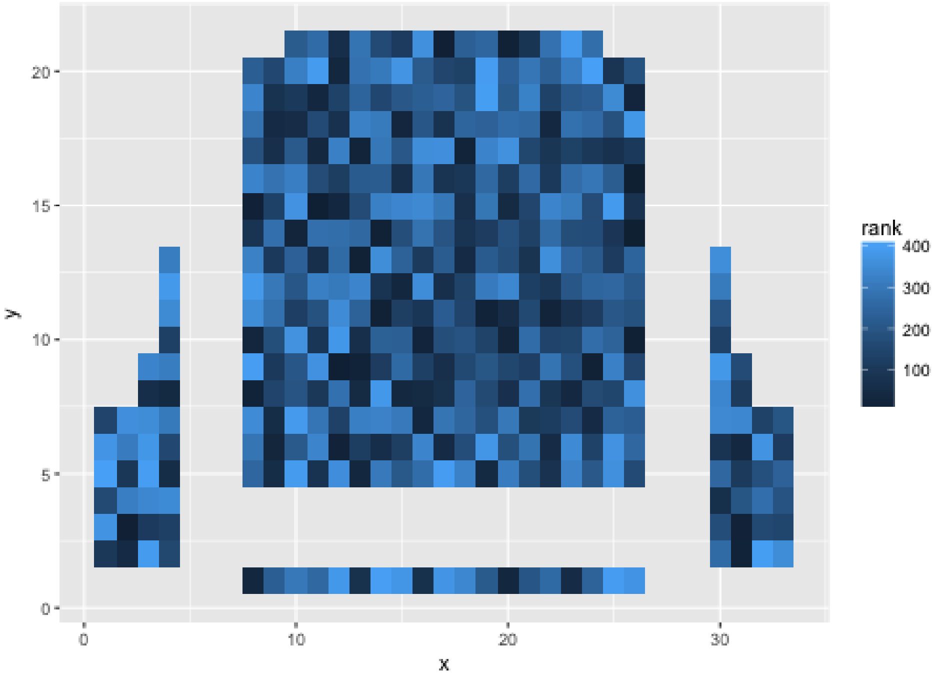


Motion chart

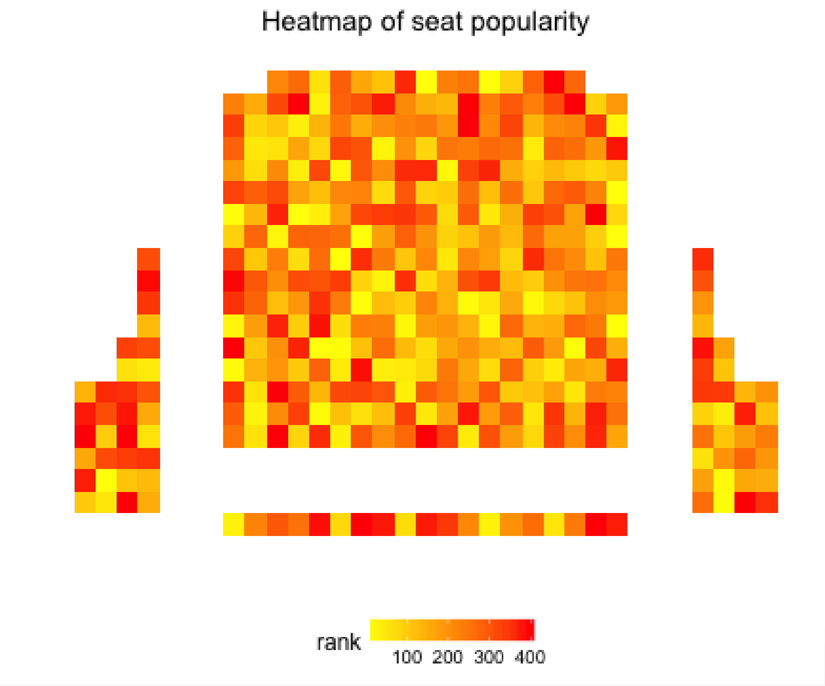
```r
library(googleVis)
M1 <- gvisMotionChart(Fruits, idvar = 'Fruit', timevar = 'Year')
plot(M1)
```

Heatmap

```
dataPath <- "https://raw.githubusercontent.com/liyujiao1026/r_graph/master/seatData.csv"
seatData <- read.csv(dataPath)
ggplot(seatData, aes(y = y, x = x)) +
            geom_tile(aes(fill = rank))
```



```
ggplot(seatData, aes(y=y, x=x)) + geom_tile(aes(fill = rank)) +
            scale_fill_gradient(low = "yellow",high = "red", name = "rank") +
            scale_x_continuous(breaks = c()) +scale_y_continuous(breaks = c()) +
            theme_minimal(base_size = 12) + xlab("") + ylab("") +
            theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
                  legend.position="bottom",legend.key.height = unit(0.4, "cm")) +
            ggtitle("Heatmap of seat popularity") +
            theme(plot.title = element_text(hjust = 0.5))
```


Heatmap of seat popularity
rank