Lecture 6

Exploratory Data Analysis

Assoc. Prof. Dr. Burkay Genç

2024-03-25

Packages used in these slides

```
library(dplyr)
library(tibble)
library(DMwR2)    # Book package, data mining with R, Torgo
library(Hmisc)
library(ggplot2)  # The plotting library
```

Seed used in these slides

```
set.seed(1024)
```

Modelling

From Wikipedia:

> Scientific modelling is a scientific activity, the aim of which is to make a particular part or feature of the world easier to **understand, define, quantify, visualize, or simulate** by referencing it to existing and usually commonly accepted knowledge.
>
> It requires **selecting and identifying relevant aspects of a situation in the real world** and then using different types of models for different aims, such as conceptual models to better *understand*, operational models to *operationalize*, mathematical models to *quantify*, and graphical models to *visualize* the subject.

Modelling

Data Mining Tasks:

- exploratory data analysis
- dependency modeling
- clustering
- anomaly detection
- predictive analytics

Exploratory Data Analysis

Exploratory Data Analysis

Main goal of EDA is to provide useful summaries of the data

- **Textual** summaries
- **Visual** summaries

Answer questions like:

- What is the **most common value** of a variable?
- Do the values of a variable **vary** a lot?
- Are there **strange / unexpected** values in the dataset?

Most “common” value

- numeric
  - mean

  $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

  $\bar{x}$ is an estimate of the population mean μ.
  - median

  $\tilde{x}$ is the item in the middle in a sorted list

Which one is more expensive to compute?

Mean vs Median

Median is less sensitive (more robust) to outlying values

```
x <- rnorm(1000)
mean(x)
```

```
## [1] 0.01229438
```

```
median(x)
```

```
## [1] 0.05130896
```

```
x[1] <- 1000
mean(x)
```

```
## [1] 1.013073
```

```
median(x)
```

```
## [1] 0.05549485
```

Mean vs Median

Be careful with NAs

```
x <- rnorm(100)
x[1] <- NA
mean(x)
```

```
## [1] NA
```

```
median(x)
```

```
## [1] NA
```

```
mean(x, na.rm = T)
```

```
## [1] -0.08823183
```

```
median(x, na.rm = T)
```

```
## [1] -0.1124146
```

Mode

Most “common” value

- nominal
  - mode
    - which value to choose?
    - `c` or `d`?

```
x <- sample(letters[1:5], 999, replace = T,
            prob = c(0.1,0.2,0.3,0.3,0.1))

foo_mode <- function (vec) {
  return (names(which.max(table(vec))))
}

table(x)
```

```
## x
##   a   b   c   d   e
##  90 198 312 312  87
```

```
foo_mode(x)
```

```
## [1] "c"
```

Summarise

```
data(algae, package="DMwR2")
t_alg <- as_tibble(algae)
print(t_alg, width = 70, n = 6)
```

```
## # A tibble: 200 × 18
##   season size  speed   mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <fct>  <fct> <fct>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 winter small medium  8      9.8  60.8  6.24 578   105   170    50
## 2 spring small medium  8.35   8    57.8  1.29 370   429.  559.    1.3
## 3 autumn small medium  8.1   11.4  40.0  5.33 347.  126.  187.   15.6
## 4 spring small medium  8.07   4.8  77.4  2.30  98.2  61.2 139.    1.4
## 5 autumn small medium  8.06   9    55.4 10.4  234.   58.2  97.6  10.5
## 6 winter small high    8.25  13.1  65.8  9.25 430    18.2  56.7  28.4
## # i 194 more rows
## # i 7 more variables: a1 <dbl>, a2 <dbl>, a3 <dbl>, a4 <dbl>,
## #   a5 <dbl>, a6 <dbl>, a7 <dbl>
```

```
t_alg %>% summarise(avgNO3 = mean(NO3, na.rm = T),
                    medA1 = median(a1, na.rm = T))
```

```
## # A tibble: 1 × 2
##   avgNO3 medA1
##    <dbl> <dbl>
## 1   3.28  6.95
```

Summarise

- Use `summarise_all` to obtain summaries for all variables

```
t_alg %>%
  select(mxPH:Cl) %>%
  summarise_all(list(~mean(., na.rm = T),
                     ~median(., na.rm = T)))
```

```
## # A tibble: 1 × 6
##   mxPH_mean mnO2_mean Cl_mean mxPH_median mnO2_median Cl_median
##       <dbl>     <dbl>   <dbl>       <dbl>       <dbl>     <dbl>
## 1      8.01      9.12    43.6        8.06         9.8      32.7
```

Summarise

- Use `group_by` to group data into clusters of similar observations and obtain summaries of clusters.

```
t_alg %>%
  group_by(season, size) %>%
  summarise(nObs = n(), medA7 = median(a7)) %>%
  ungroup() %>% arrange(desc(medA7))
```

```
## `summarise()` has grouped output by 'season'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 12 × 4
##    season size    nObs medA7
##    <fct>  <fct>  <int> <dbl>
##  1 spring large     12  1.95
##  2 summer small     14  1.45
##  3 winter medium    26  1.4
##  4 autumn medium    16  1.05
##  5 spring medium    21  1
##  6 summer medium    21  1
##  7 autumn large     11  0
##  8 autumn small     13  0
##  9 spring small     20  0
## 10 summer large     10  0
## 11 winter large     12  0
## 12 winter small     24  0
```

Variance

- Variance is a measure of the spread of the variable

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

s_x^2 is an estimate of the population variance σ_x^2.

σ_x is the standart deviation of the population.

- A more robust metric is IQR: inter-quartile range
  - IQR is the difference between 1st and 3rd quartiles.
  - IQR covers the most common 50% of the distribution.

Variance

- Spread Metric Functions
  - `sd` : standart deviation
  - `var` : variance
  - `IQR` : Interquartile range
  - `range` : difference between max and min

Variance

- `sd` and `IQR` are similar in scale

```
t_alg %>%
  select(a1:a3) %>%
  summarise_all(list(sd = sd, IQR = IQR))
```

```
## # A tibble: 1 × 6
##   a1_sd a2_sd a3_sd a1_IQR a2_IQR a3_IQR
##   <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1  21.3  11.0  6.95   23.3   11.4   4.93
```

Unknown Variables

- Unknown variables values are a major problem
  - We have to understand how much NAs we have
  - ... and where

```
data(algae, package="DMwR2")
t_alg <- as_tibble(algae)
sum(is.na(t_alg))
```

```
## [1] 33
```

```
nrow(t_alg)
```

```
## [1] 200
```

```
sum(complete.cases(t_alg))
```

```
## [1] 184
```

```
t_alg_comp <- t_alg[complete.cases(t_alg),]
sum(is.na(t_alg_comp))
```

```
## [1] 0
```

Unknown Variables

- Check columns that contain NAs

```
# Number of NAs in each column
na_col <- apply(t_alg, 2, function (col) sum(is.na(col)))
na_col[na_col != 0]
```

```
## mxPH mnO2   Cl  NO3  NH4 oPO4  PO4 Chla
##    1    2   10    2    2    2    2   12
```

```
na_col[na_col == 0]
```

```
## season   size  speed     a1     a2     a3     a4     a5     a6     a7
##      0      0      0      0      0      0      0      0      0      0
```

Outliers

"An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.", Hawkins (1980)

- How to detect? Boxplot rule is one way to do it:

$$IQR = Q_3 - Q_1$$

$$V_l = Q_1 - 1.5 \times IQR$$

$$V_r = Q_3 + 1.5 \times IQR$$

$$V = [V_l, V_r]$$

- ...then, everything outside of V is an outlier

Outliers

```
foo_outliers <- function(x)
{
  Q1 <- quantile(x, 0.25)
  Q3 <- quantile(x, 0.75)
  IQR <- Q3 - Q1
  Vl <- Q1 - 1.5 * IQR
  Vr <- Q3 + 1.5 * IQR
  return (which(x < Vl | x > Vr))
}

x <- rnorm(1000)
outliers <- foo_outliers(x)
x[outliers]
```

```
## [1]  2.713597 -2.793030 -3.336088 -3.552366  2.971351
```

```
x <- x[-outliers]
```

Outliers

- Detection of outliers is a very case sensitive topic
  - Multi-modal distributions
  - Multi-variate outliers
  - Categorical outliers
  - Contextual outliers

```
wage <- c(0, 100, 120, 110, 0, 150, 120, 120, 130, 160, 150, 0, 130)
mean(wage)
```

```
## [1] 99.23077
```

```
foo_outliers(wage)
```

```
## [1]  1  5 12
```

```
wage <- wage[-foo_outliers(wage)]
cat("Average income in this country is", mean(wage))
```

```
## Average income in this country is 129
```

- **WRONG**!!!

Summary

```
data(iris)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Summary

```
describe(iris[,1:2])   # from package Hmisc
```

```
## iris[, 1:2]
##
##  2  Variables      150  Observations
## ---------------------------------------------------------------------------
## Sepal.Length
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      150        0       35    0.998    5.843   0.9462    4.600    4.800
##      .25      .50      .75      .90      .95
##    5.100    5.800    6.400    6.900    7.255
##
## lowest : 4.3 4.4 4.5 4.6 4.7, highest: 7.3 7.4 7.6 7.7 7.9
## ---------------------------------------------------------------------------
## Sepal.Width
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      150        0       23    0.992    3.057   0.4872    2.345    2.500
##      .25      .50      .75      .90      .95
##    2.800    3.000    3.300    3.610    3.800
##
## lowest : 2   2.2 2.3 2.4 2.5, highest: 3.9 4   4.1 4.2 4.4
## ---------------------------------------------------------------------------
```

Summary

If you want to compute summaries **grouped by** a specific variable's values,

```
by(algae[,c("season", "speed", "mxPH")], algae$size, summary)
```

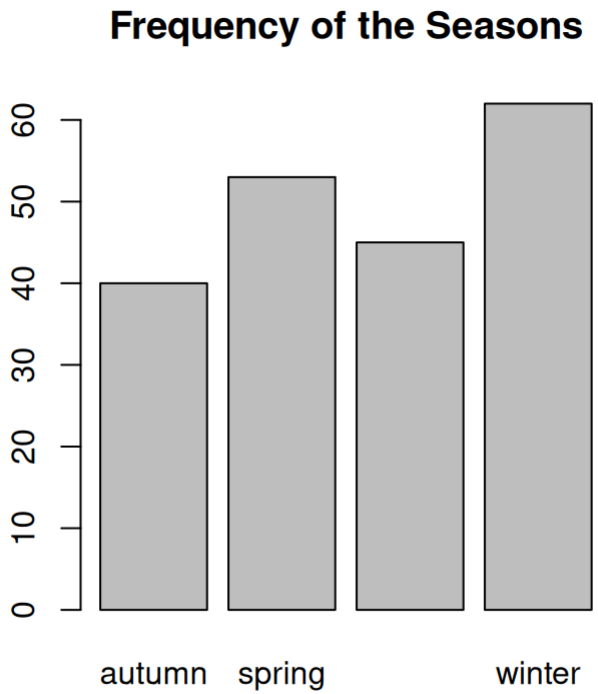
```
## algae$size: large
##     season      speed         mxPH      
##  autumn:11   high  : 7   Min.   :7.300  
##  spring:12   low   :17   1st Qu.:8.200  
##  summer:10   medium:21   Median :8.400  
##  winter:12               Mean   :8.396  
##                          3rd Qu.:8.600  
##                          Max.   :9.500  
## ------------------------------------------------------------ 
## algae$size: medium
##     season      speed         mxPH      
##  autumn:16   high  :34   Min.   :7.300  
##  spring:21   low   :15   1st Qu.:7.800  
##  summer:21   medium:35   Median :8.100  
##  winter:26               Mean   :8.101  
##                          3rd Qu.:8.408  
##                          Max.   :9.700  
## ------------------------------------------------------------ 
## algae$size: small
##     season      speed         mxPH      
##  autumn:13   high  :43   Min.   :5.600  
##  spring:20   low   : 1   1st Qu.:7.410  
##  summer:14   medium:27   Median :7.795  
##  winter:24               Mean   :7.657  
##                          3rd Qu.:8.068  
##                          Max.   :8.700  
##                          NA's   :1
```

Visualization

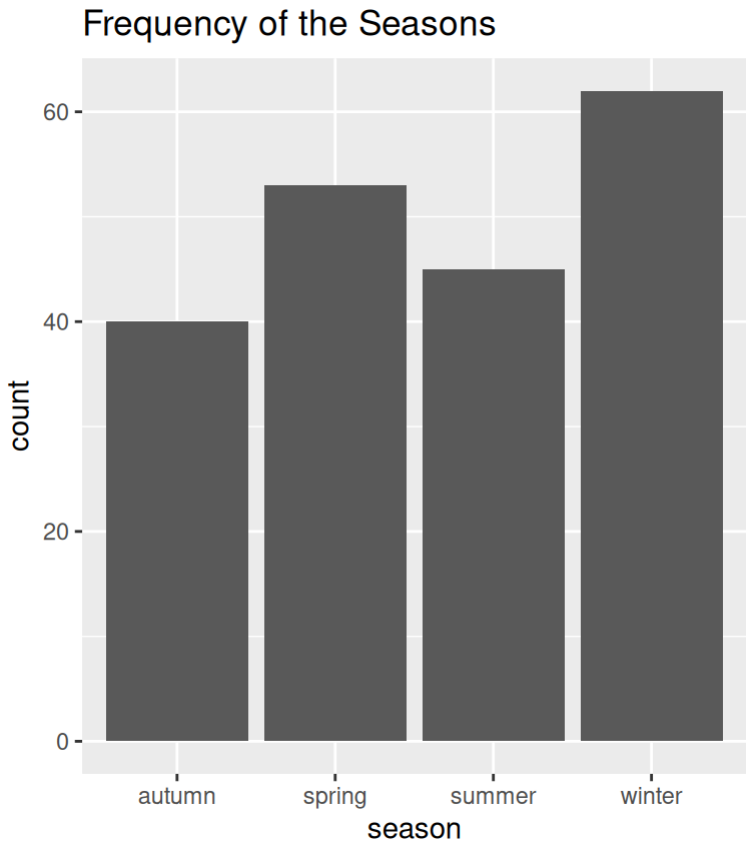
- Visualization is a very important component of understanding your data
- R excels at data visualization
  - standart graphics -> standart plotting functions
  - grid graphics -> ggplot2

Visualization - Categorical

```
data(algae, package="DMwR2")
freqOcc <- table(algae$season)
barplot(freqOcc,
        main = "Frequency of the Seasons")
```

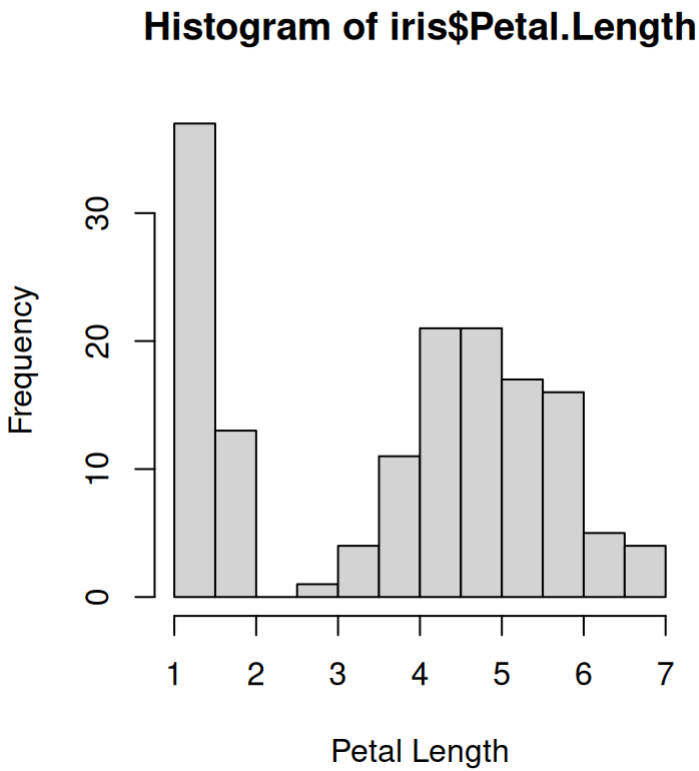



```
ggplot(algae, aes(x = season)) +
  geom_bar() +
  ggtitle("Frequency of the Seasons")
```

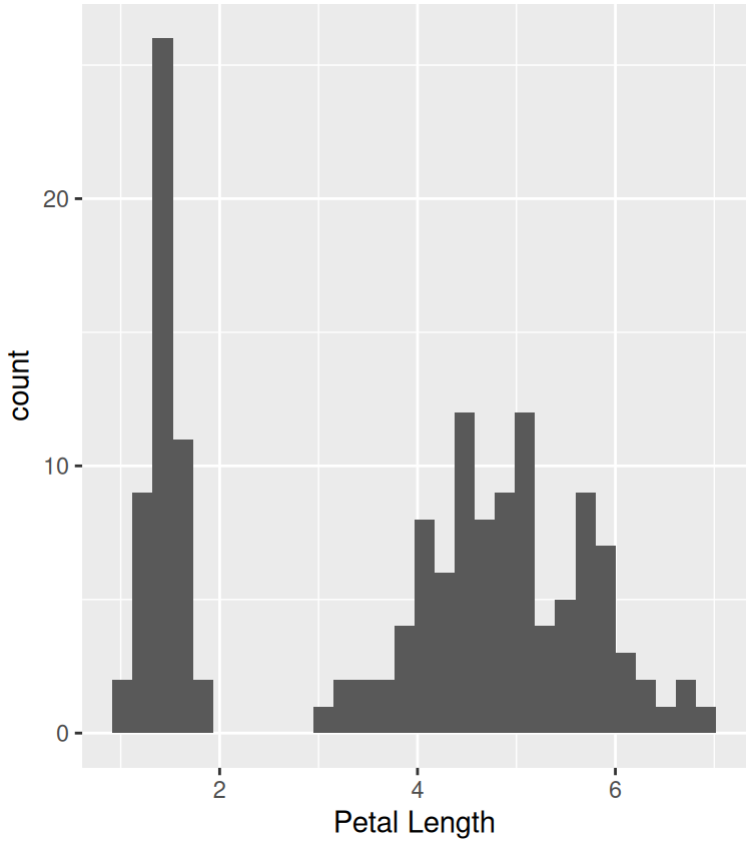


Visualization - Numerical

```
hist(iris$Petal.Length, xlab = "Petal Length")
```

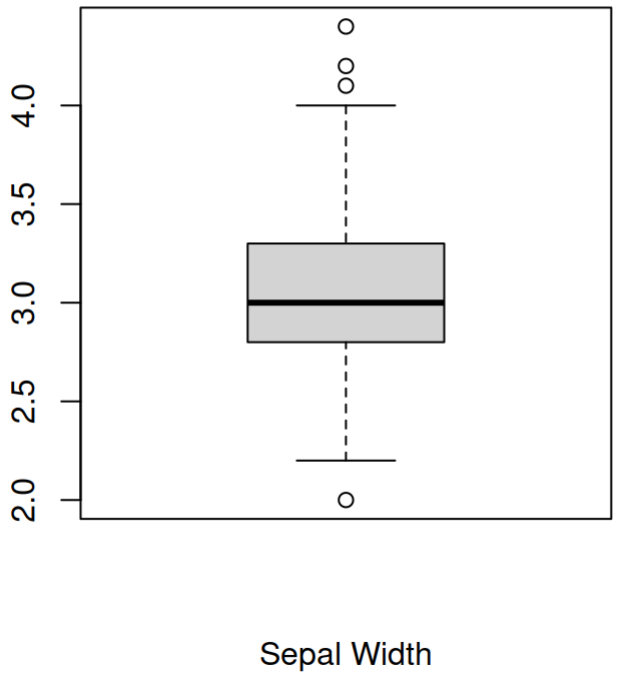



```
ggplot(iris, aes(x = Petal.Length)) +
  geom_histogram() +
  xlab("Petal Length")
```

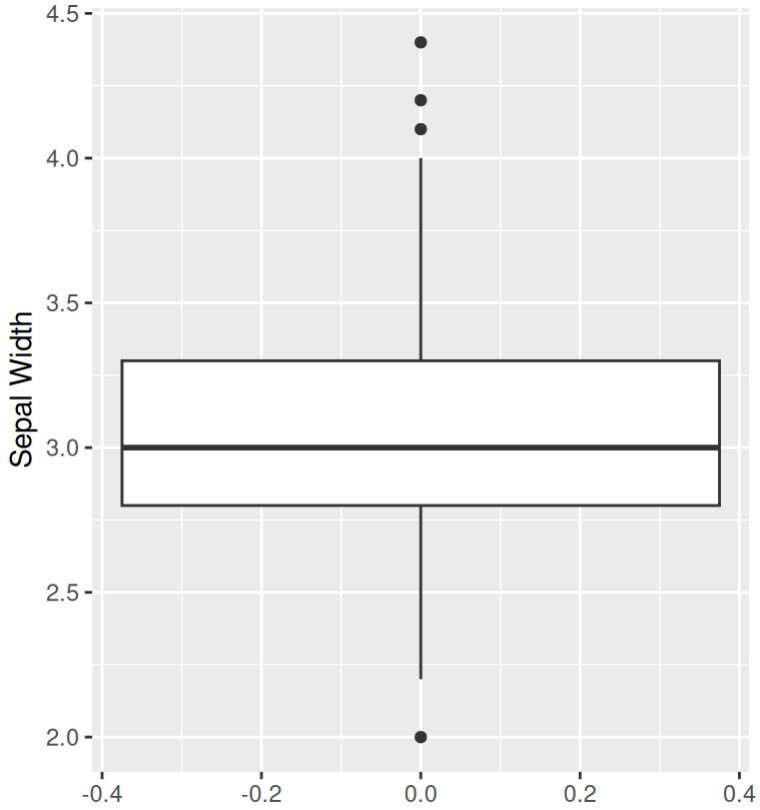


Visualization - Numerical

```
boxplot(iris$Sepal.Width,
        xlab = "Sepal Width")
```

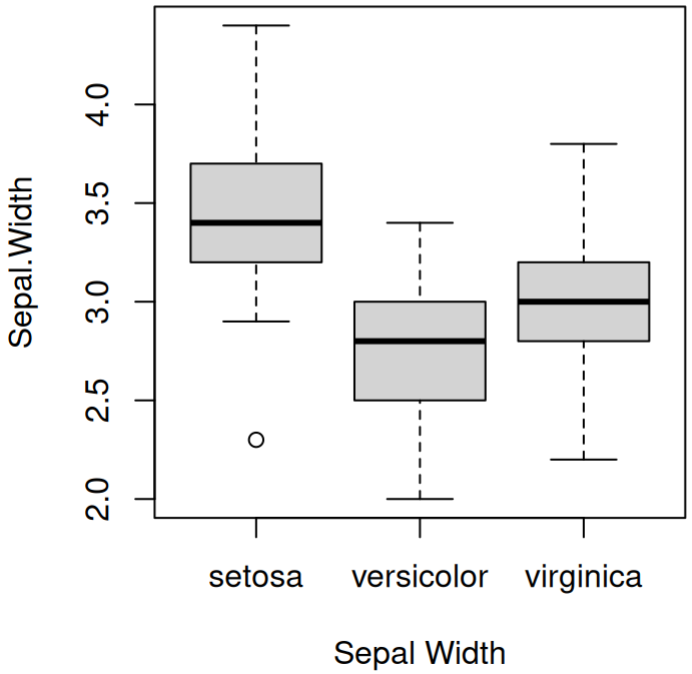



```
ggplot(iris, aes(y = Sepal.Width)) +
  geom_boxplot() +
  xlab("") + ylab("Sepal Width")
```

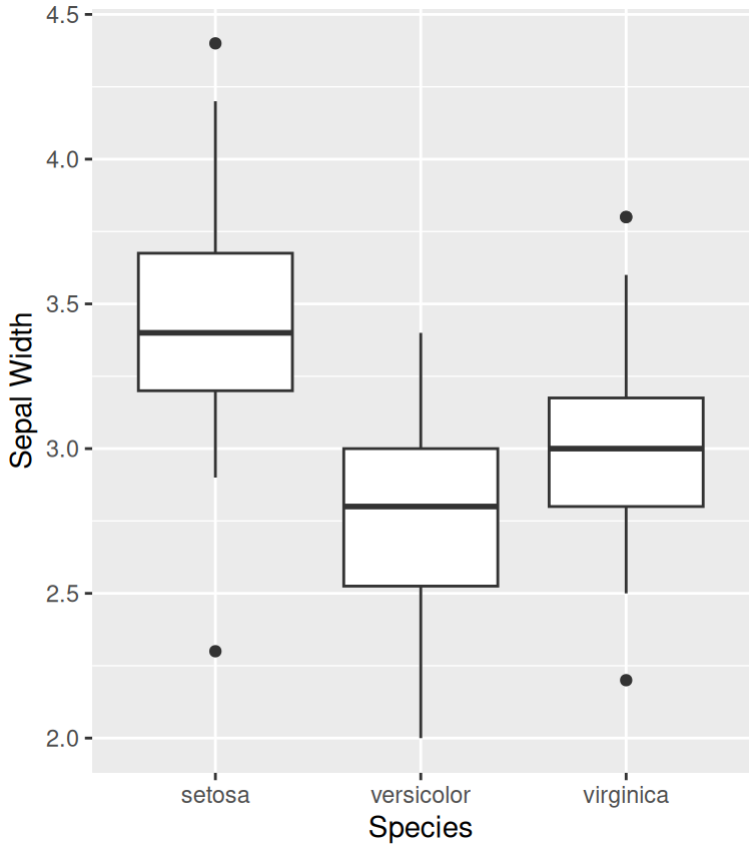


Visualization - Conditioning

```
boxplot(Sepal.Width ~ Species,
        iris,
        xlab = "Sepal Width")
```

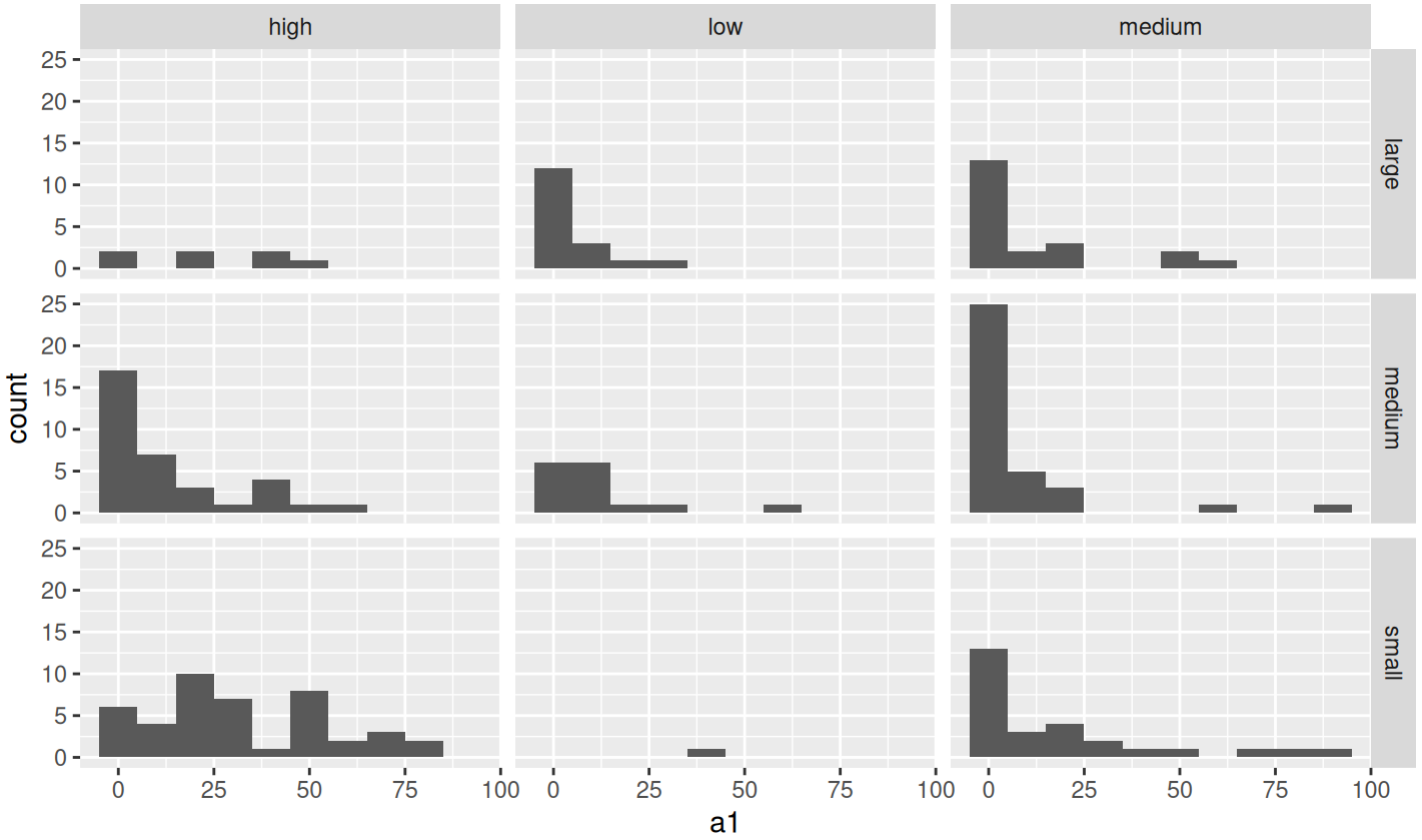



```
ggplot(iris, aes(x = Species,
                 y = Sepal.Width)) +
  geom_boxplot() +
  xlab("Species") +
  ylab("Sepal Width")
```



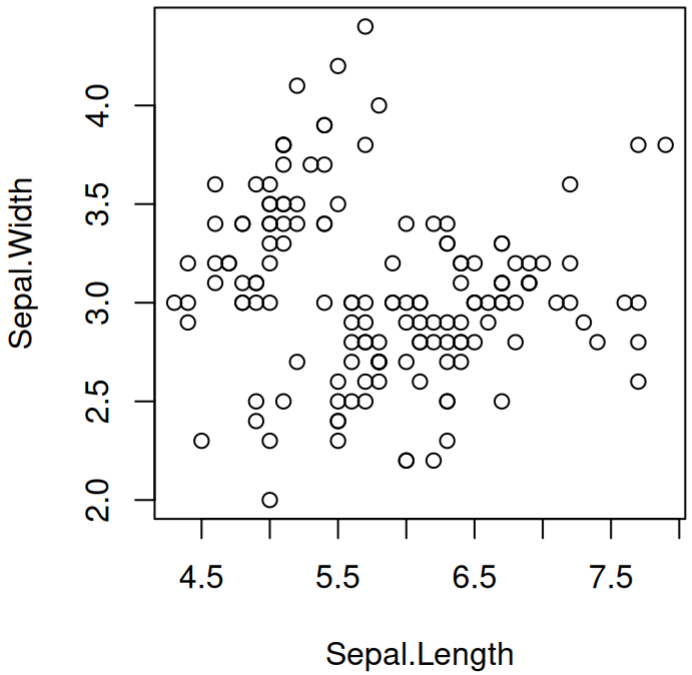
Visualization - Facets

```
ggplot(algae, aes(x = a1)) +
  geom_histogram(binwidth = 10) +
    facet_grid(size ~ speed)
```

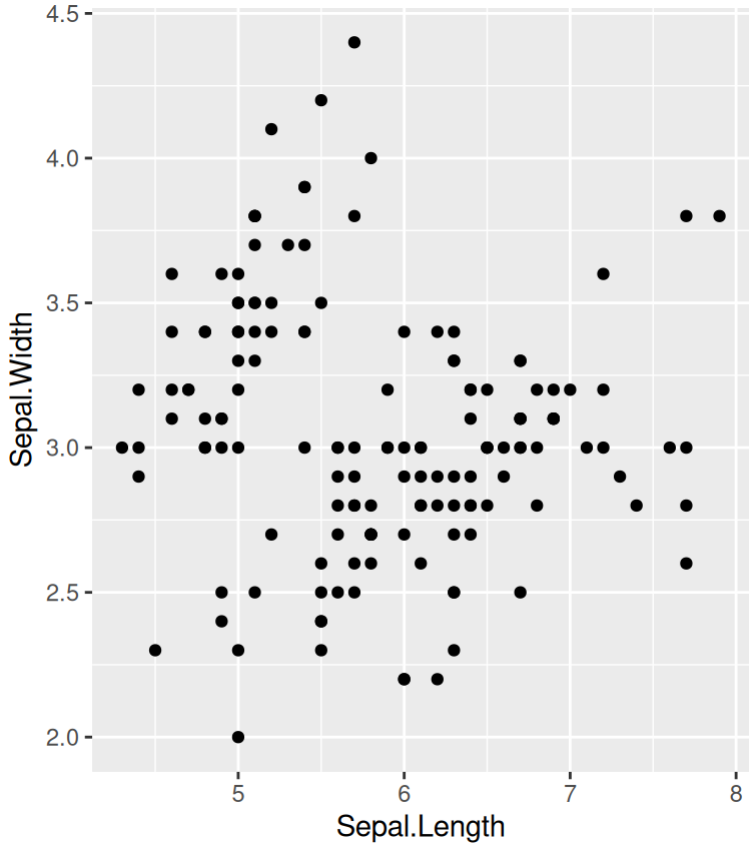


Visualization - Two Variables

```
plot(Sepal.Width ~ Sepal.Length, iris)
```

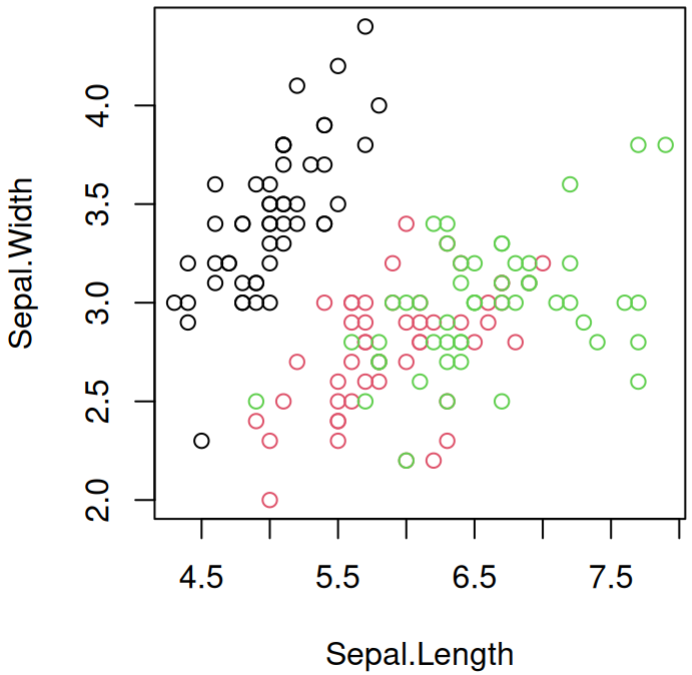



```
ggplot(iris, aes(x = Sepal.Length,
                 y = Sepal.Width)) +
  geom_point()
```

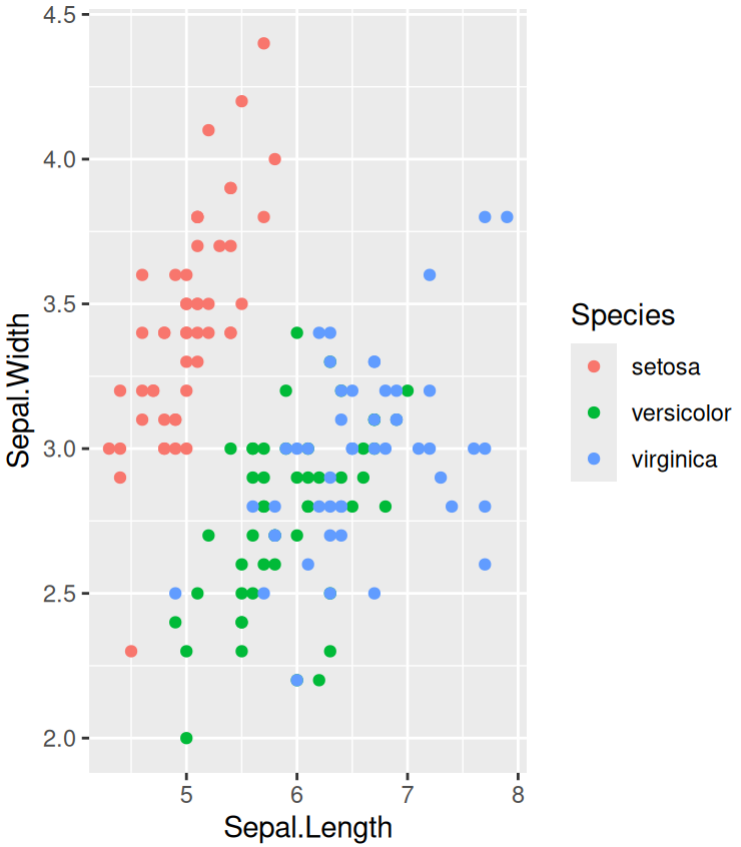


Visualization - Two Variables

```
plot(Sepal.Width ~ Sepal.Length, iris, col =
Species)
```




```
ggplot(iris, aes(x = Sepal.Length,
                 y = Sepal.Width,
                 color = Species)) +
  geom_point()
```



Visualization - Two Variables

```
ggplot(iris, aes(x = Sepal.Length,
                 y = Sepal.Width)) +
  geom_point() +
  facet_wrap( ~ Species)
```