MTH 224O, Spring 2024

Instructor: Bahman Angoshtari

Lecture 21

Section 6.2: histogram, boxplot, scatterplot.

Sections 6.4: point estimation.

0. Overview of the lecture

- We discuss common graphical summaries of data, namely, **Histogram**, **Boxplot**, **Scatterplot**. We also briefly go over R which is the statistical software that we will use during this course.
- One of the major themes in statistics is estimating some numerical characteristic of a population by collecting a sample of observations from it. This is called **point estimation.** We have seen examples of point estimation: sample mean estimates the population mean and sample variance estimates population variance. We start discussing this topic, and continue in the next lecture.

In [1]:
```
# Setting parameters of the Jupyter notebook
# This cell is only usefull if your are using Jupyter
sc = 0.75
options(repr.plot.width=16*sc,
        repr.plot.height=6*sc,
        repr.plot.pointsize = 20, # Text height in pt
        repr.plot.bg        = 'white',
        repr.plot.antialias = 'gray',
        #nice medium-res DPI
        repr.plot.res       = 300,
        #jpeg quality bumped from default
        repr.plot.quality   = 90,
        #vector font family
        repr.plot.family    = 'serif', # Vector font family. 'sans', 'serif'
        "getSymbols.warning4.0"=FALSE)
```

1. Introduction to R

- R is a programming language for statistical computing and graphics. It is widely used among statisticians and contains a large number of ready-to-use functions (called *packages*).
- Start by installing R. Please visit: https://www.r-project.org/.
- This will install the R software. It is also recommended to install an IDE (integrated development environment) for R, such as RStudio. Please visit: https://www.rstudio.com/products/rstudio/download/.
- There are many online resources for learning R. I will briefly explain the codes presented in each class. However, you should thoroughly examine each code on your own, and use the R documentation to learn how they are used.

- Let us open RStudio, and install our first package. As mentioned before, packages include ready-to-use functions. `quantmod` is a package that can be used for obtaining financial data.

- To install the package use Tools -> Install Packages... menu in RStudio, or directly use the following command:

In [2]:
```
install.packages("quantmod")
```

```
Installing package into 'C:/Users/bangoshtari/AppData/Local/R/win-library/4.
2'
(as 'lib' is unspecified)

package 'quantmod' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\bangoshtari\AppData\Local\Temp\Rtmpw7vtrf\downloaded_packag
es
```

- This command copy the package files on our computer (from an online repository). You only need to install a package once.

- However, to be able to use an installed package, we need to load it. This is done by the following command.

In [3]:
```
library('quantmod')
```

```
Loading required package: xts

Loading required package: zoo


Attaching package: 'zoo'


The following objects are masked from 'package:base':

    as.Date, as.Date.numeric


Loading required package: TTR

Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

- As you can see, R automatically load other packages that `quantmod` relies on, namely, packages `xtx` and `zoo` (both are packages that define time-series data types).

Example 1

Recall Example 3 from Lecture 20.

Occupants	1	2	3	4	5
Number of Cars	70	15	10	3	2

Find the sample mean, variance, and quartiles using R.

Solution:

In [4]:
```
data = c(rep(1,70),rep(2,15), rep(3,5), rep(4,3), rep(5,2))
data
```

1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 ·
1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 ·
1 · 1 · 1 · 1 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 3 · 3 · 3 · 3 · 3 · 4 · 4 · 4 · 5 · 5

In [5]:
```
mean(data)
var(data)
sd(data)
```

1.44210526315789

0.802463605823068

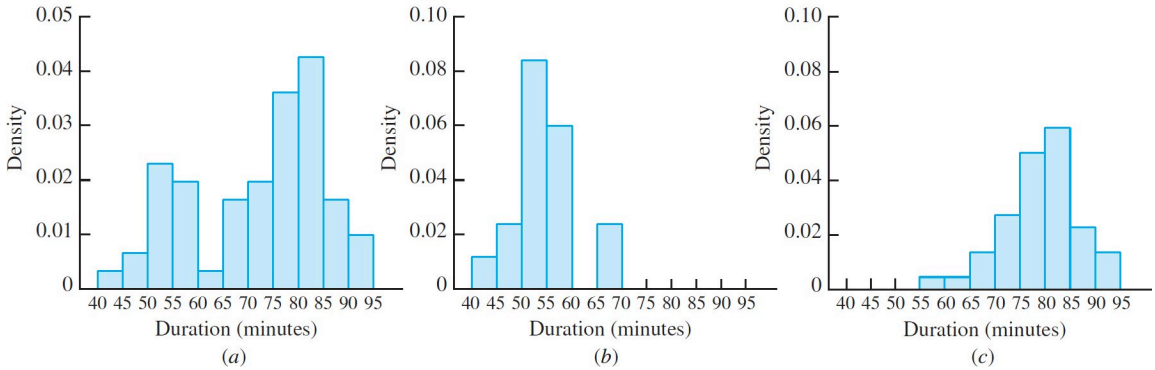
0.895803329879426

```
In [6]: quantile(data, type=2)
```

0%: 1 **25%:** 1 **50%:** 1 **75%:** 2 **100%:** 5

2. Histograms

- A Histogram is a graphical display that gives an idea of the "*shape*" of the sample, indicating regions where sample points are concentrated and regions where they are sparse.
- The bars of the histogram touch each other. A space indicates that there are no observations in that interval.



To draw a histogram:

- Choose boundary points for the class intervals. Usually these intervals are the same width. They are called **bins**.
- Compute the frequencies: this is the number of observations that fall into each bin.
- If the class intervals are the same width, then draw a rectangle for each class, whose height is equal to the frequencies or relative frequencies.
- If the bins are of unequal widths, the heights of the rectangles must be set equal to the densities, where density is the relative frequency divided by the bin width. In other words, the area of the rectangle must be equal to the relative frequency. What is the total area of the rectangles in such a histogram?

Example 2: histogram of Old Faithful data

- A sample of 60 durations of dormant periods of the geyser Old Faithful in Minutes is as follows:

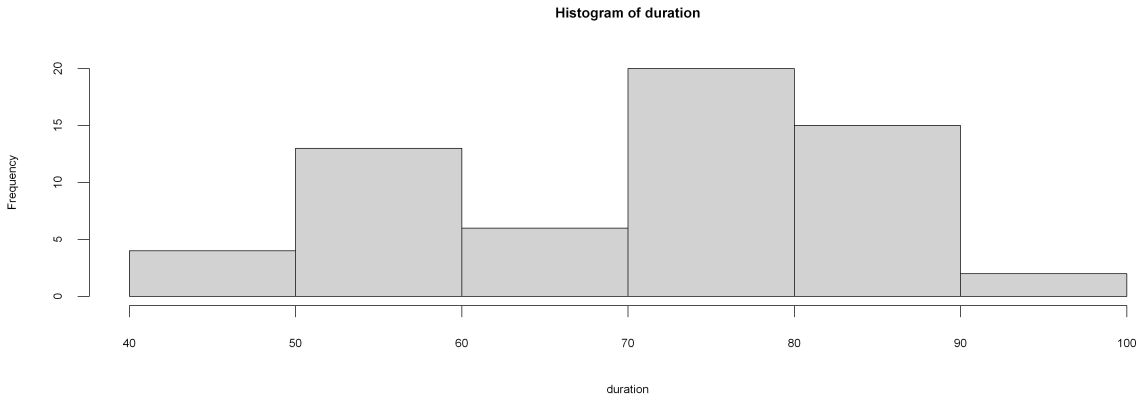
```
42,45,49,50,51,51,51,51,53,53,55,55,56,56,57,58,60,66,67,67,
68,69,70,71,72,73,73,74,75,75,75,75,76,76,76,76,76,79,79,80,
```

```
80,80,80,81,82,82,82,83,83,84,84,84,85,86,86,86,88,90,91,93
```

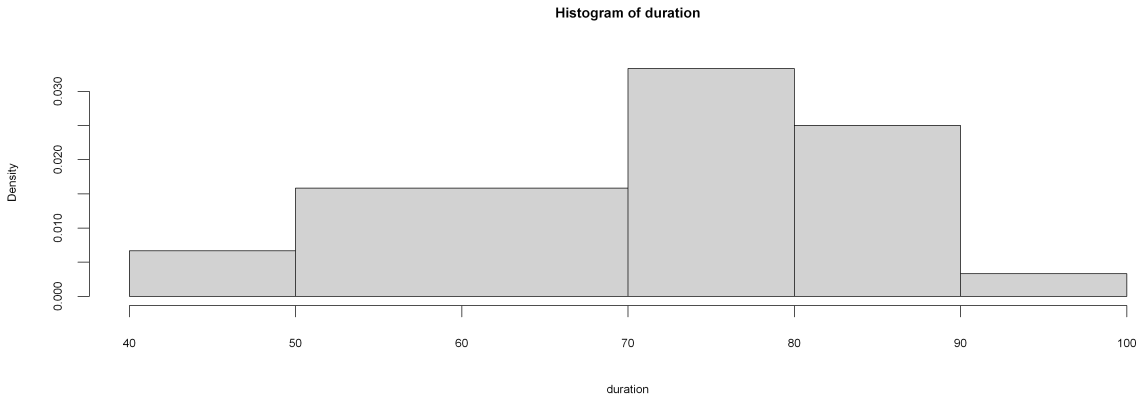
Draw the histogram of this sample using R.

Solution:

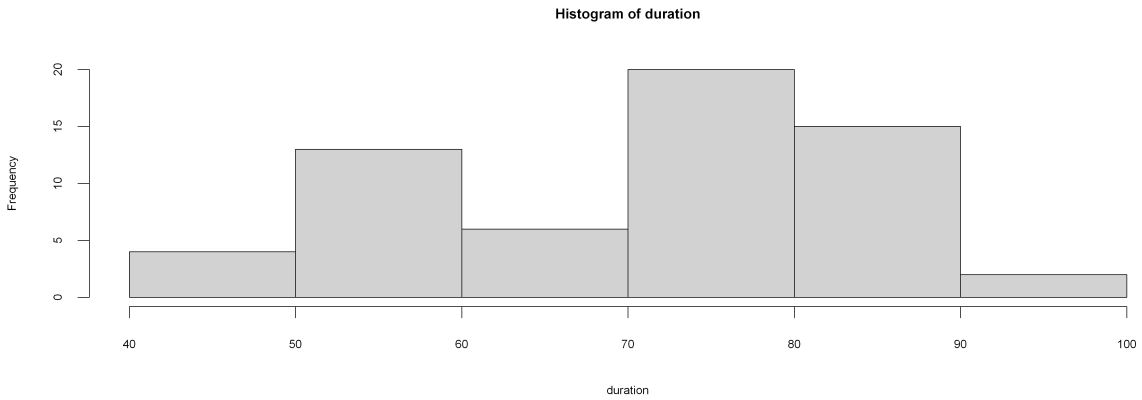
In [7]:
```
duration = c(
  42,45,49,50,51,51,51,51,53,53,55,55,56,56,57,58,60,66,67,67,
  68,69,70,71,72,73,73,74,75,75,75,75,76,76,76,76,76,79,79,80,
  80,80,80,81,82,82,82,83,83,84,84,84,85,86,86,86,88,90,91,93
)
sc = 1.1
options(repr.plot.width=16*sc, repr.plot.height=6*sc)
hist(duration, breaks=6, freq = TRUE)
```



In [8]:
```
duration = c(
  42,45,49,50,51,51,51,51,53,53,55,55,56,56,57,58,60,66,67,67,
  68,69,70,71,72,73,73,74,75,75,75,75,76,76,76,76,76,79,79,80,
  80,80,80,81,82,82,82,83,83,84,84,84,85,86,86,86,88,90,91,93
)
sc = 1.1
options(repr.plot.width=16*sc, repr.plot.height=6*sc)
hist(duration, xlim=c(40,100), breaks = c(40,50,70,80,90, 100), freq=FALSE)
```

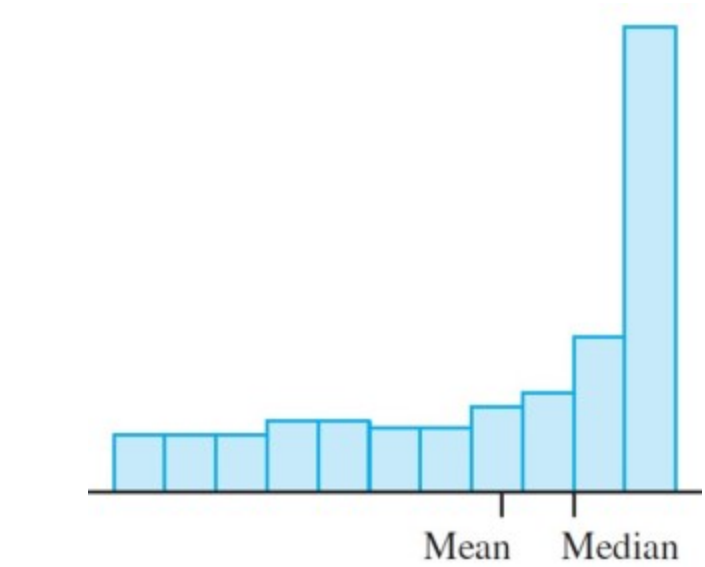


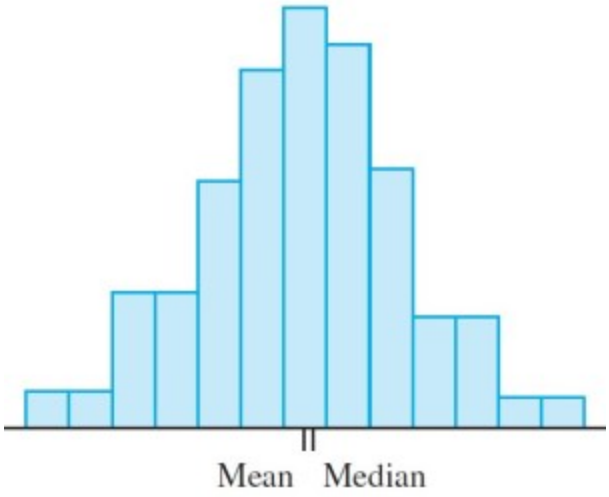
In [9]:
```
hist(duration)
```

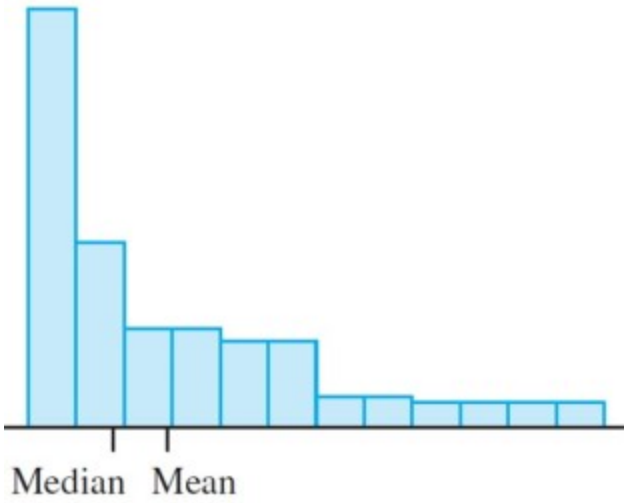


Symmetry and Skewness

- A histogram is perfectly **symmetric** if its right half is a mirror image of its left half.
- Histograms that are not symmetric are referred to as **skewed**.
    - A histogram with a long right-hand tail is said to be **skewed to the right**, or **positively skewed**.
    - A histogram with a long left-hand tail is said to be **skewed to the left**, or **negatively skewed**.





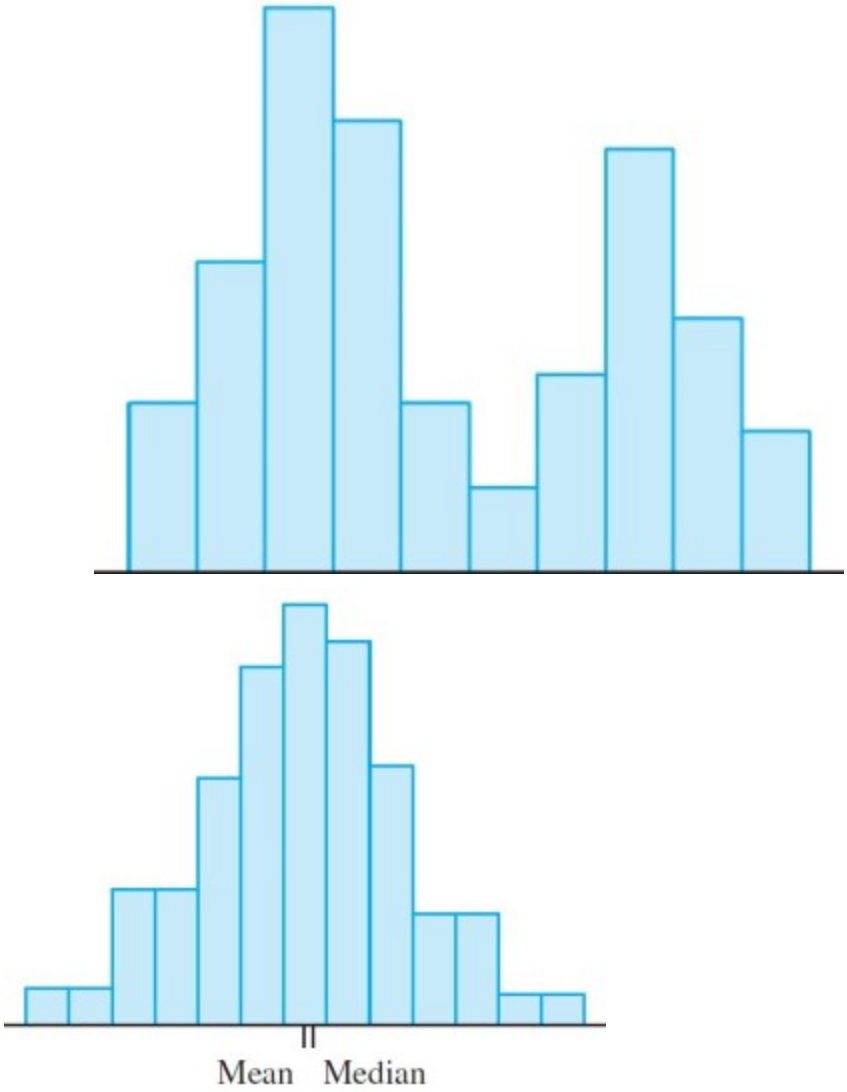


- When a histogram is roughly symmetric, the mean and the median are approximately equal.
- When a histogram is right-skewed, the mean is greater than the median.

- When a histogram is left-skewed, the mean is less than the median.

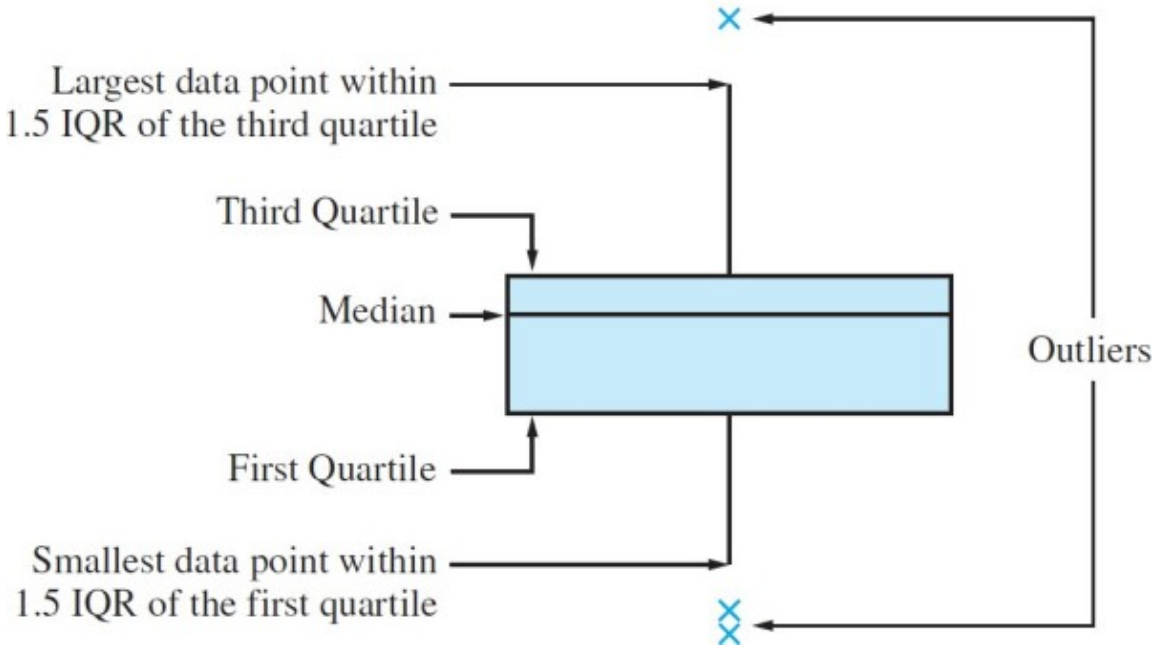
Unimodal and bimodal

- A histogram with only one peak is what we call **unimodal**.
- If a histogram has two peaks then we say that it is **bimodal**. Bimodal histograms often indicate subsamples.
- If there are more than two peaks in a histogram, then it is said to be **multimodal**.



3. Boxplots

- A **boxplot** is a graphic that presents the median, the first and third quartiles, and any outliers present in the sample.
- The **interquartile range (IQR)** is the difference between the third quartile and the first quartile. This is the distance needed to span the middle half of the data.



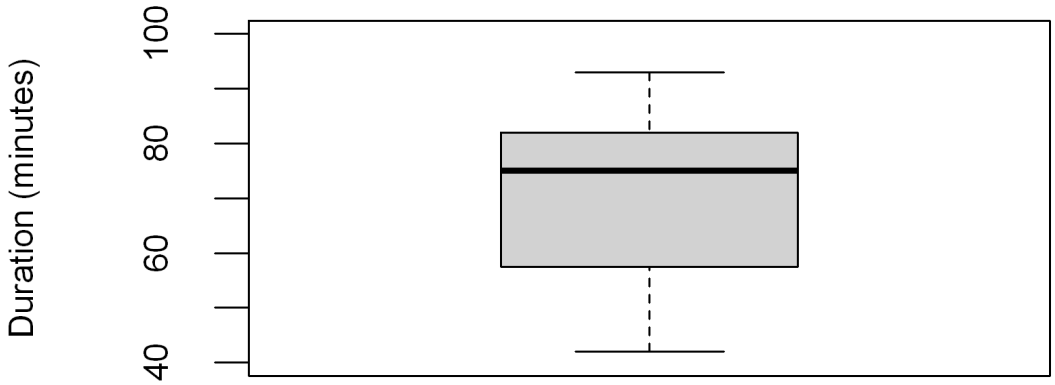
To create a boxplot:

- Compute the median and the first and third quartiles of the sample. Indicate these with horizontal lines. Draw vertical lines to complete the box.
- Find the largest sample value that is not more than 1.5 IQR above the third quartile, and the smallest sample value that is not more than 1.5 IQR below the first quartile. Extend vertical lines (**whiskers**) from the quartile lines to these points.
- Points more than 1.5 IQR above the third quartile, or more than 1.5 IQR below the first quartile, are designated as **outliers**. Plot each outlier individually.

Example 3: boxplot of Old Faithful data

In [10]:
```
duration = c(
  42,45,49,50,51,51,51,51,53,53,55,55,56,56,57,58,60,66,67,67,
  68,69,70,71,72,73,73,74,75,75,75,75,76,76,76,76,76,79,79,80,
  80,80,80,81,82,82,82,83,83,84,84,84,85,86,86,86,88,90,91,93
)
sc = 0.8
options(repr.plot.width=7.5*sc, repr.plot.height=6*sc)

boxplot(duration, ylab = "Duration (minutes)", ylim=c(40,100))
```



- Notice there are no outliers in these data.
- Looking at the four pieces of the boxplot, we can tell that the sample values are comparatively densely packed between the median and the third quartile.
- The distance between the first quartile and the median is greater than the distance between the median and the third quartile. This boxplot suggests that the data are skewed to the left.

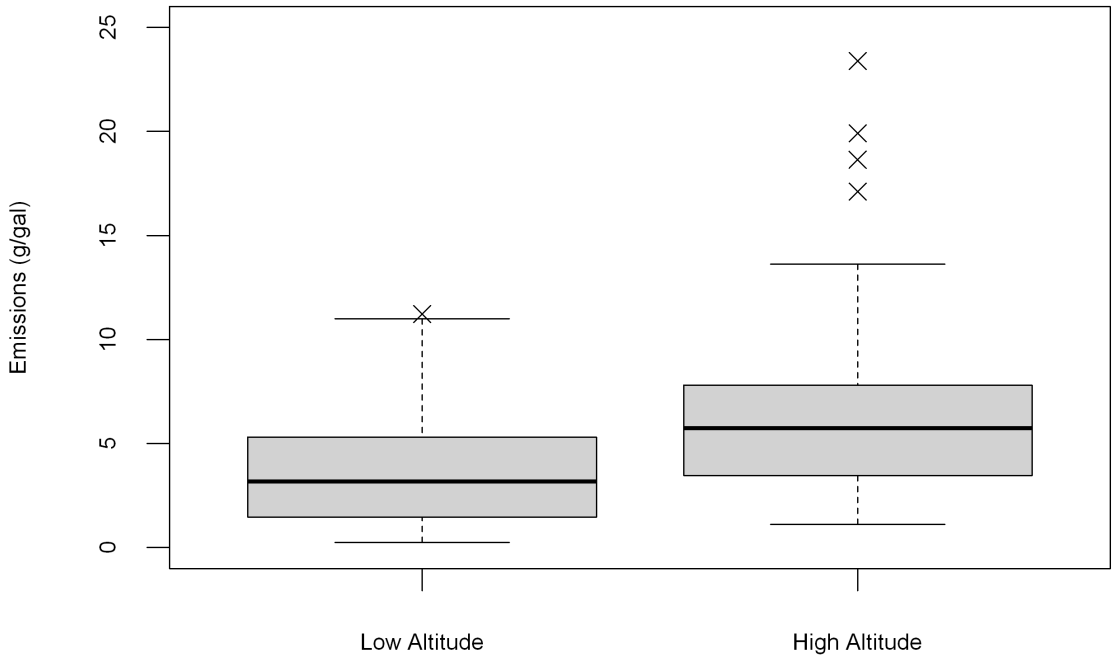
Comparative Boxplots

- Sometimes we want to compare two or more samples.
- We can place the boxplots of the two (or more) samples side-by-side.
- This will allow us to compare how the medians differ between samples, as well as the first and third quartile.
- It also tells us about the difference in spread between the two samples.

Example 4: Comparative boxplots for emissions data

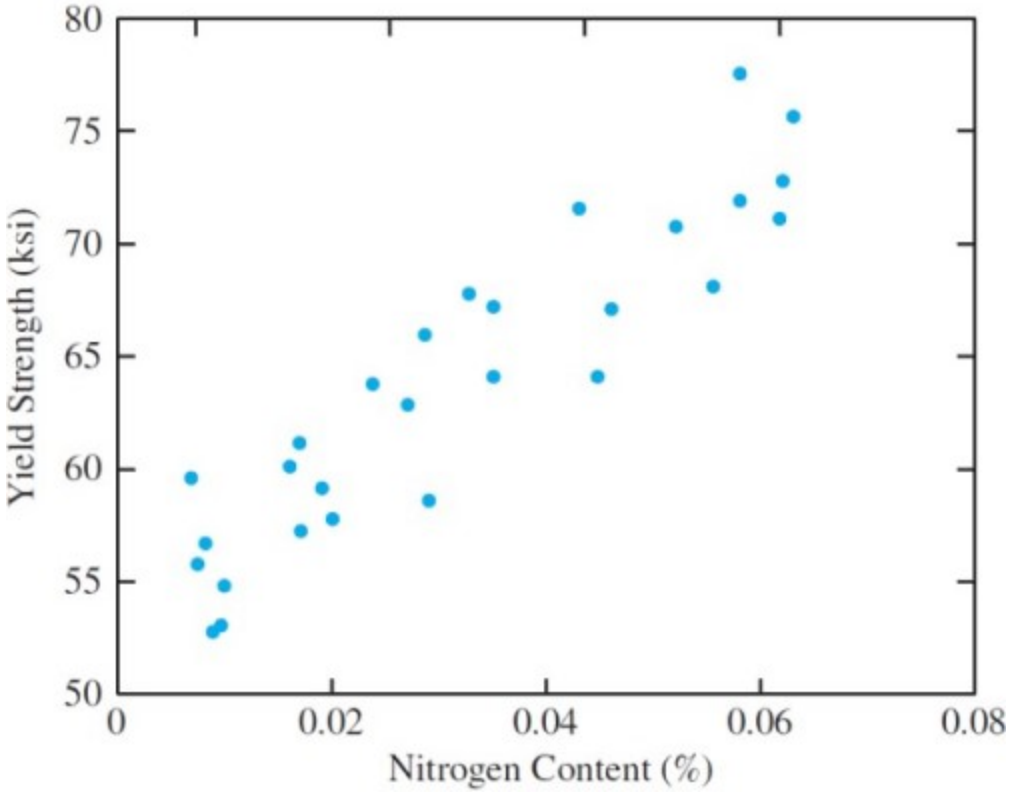
In [11]:
```
low_alt = c(
    1.5, 0.87,1.12,1.25,3.46,1.11,1.12,0.88,1.29,0.94,0.64,1.31,
    2.49,1.48,1.06,1.11,2.15,0.86,1.81,1.47,1.24,1.63,2.14,6.64,
    4.04,2.48,2.98,7.39,2.66,11,4.57,4.38,0.87,1.1,1.11,0.61,
    1.46,0.97,0.9,1.4,1.37,1.81,1.14,1.63,3.67,0.55,2.67,2.63,
    3.03,1.23,1.04,1.63,3.12,2.37,2.12,2.68,1.17,3.34,3.79,1.28,
    2.1,6.55,1.18,3.06,0.48,0.25,0.53,3.36,3.47,2.74,1.88,5.94,
    4.24,3.52,3.59,3.1,3.33,4.58,6.73,7.82,4.59,5.12,5.67,4.07,
    4.01,2.72,3.24,5.79,3.59,3.48,2.96,5.3,3.93,3.52,2.96,3.12,
    1.07,5.3,5.16,7.74,5.41,3.4,4.97,11.23,9.3,6.5,4.62,5.45,4.93,
    6.05,5.82,10.19,3.62,2.67,2.75,8.92,9.93,6.96,5.78,9.14,10.63,
    8.23,6.83,5.6,5.41,6.7,5.93,4.51,9.04,7.71,7.21,4.67,4.49,
    4.63,2.8,2.16,2.97,3.9
)
high_alt = c(
    7.59,6.28,6.07,5.23,5.54,3.46,2.44,3.01,13.63,13.02,23.38,9.24,3.22,
    2.06,4.04,17.11,12.26,19.91,8.5,7.81,7.18,6.95,18.64,7.1,6.04,5.66,
    8.86,4.4,3.57,4.35,3.84,2.37,3.81,5.32,5.84,2.89,4.68,1.85,9.14,8.67,
    9.52,2.68,10.14,9.2,7.31,2.09,6.32,6.53,6.32,2.01,5.91,5.6,5.61,1.5,
    6.46,5.29,5.64,2.07,1.11,3.32,1.83,7.56
)
```

In [12]:
```
sc = 1.2
options(repr.plot.width=7.5*sc, repr.plot.height=6*sc)
boxplot(
    low_alt, high_alt,
    ylab = "Emissions (g/gal)", ylim = c(0,25),
    names = c("Low Altitude", "High Altitude"),
    pch=4
)
```



4. Scatterplot

- A sample that has a pair of values for each observation is called **bivariate**.
- The graphical summary for bivariate data is a **scatterplot**.



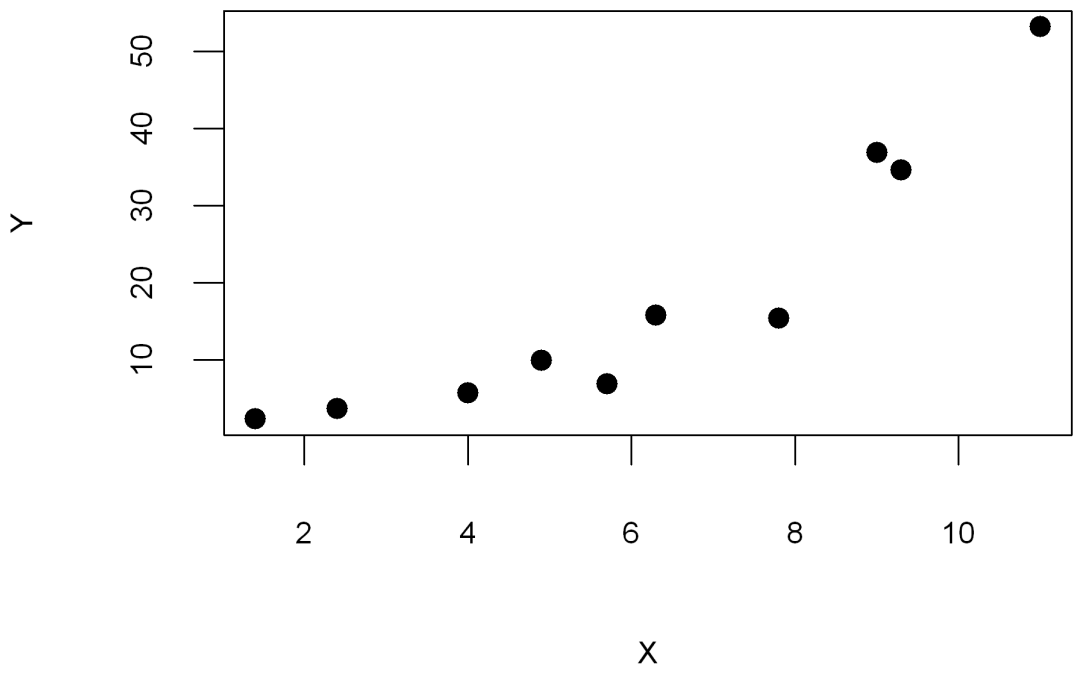
- If the dots in the scatterplot do not appear to have a general trend or pattern, then there is little evidence of a relationship between the two variables.
- If the dots on the scatterplot appear to roughly follow a straight line, then there is evidence of a linear relationship between the two variables.
- When a linear relationship exists, we may be able to use one variable to help predict the values of the other variable.

In [13]:
```
XY = cbind(
    c(1.4,2.4,4,4.9,5.7,6.3,7.8,9,9.3,11),
    c(2.3,3.7,5.7,9.9,6.9,15.8,15.4,36.9,34.6,53.2)
)
XY
```

A matrix: 10 × 2 of type dbl

1.4	2.3
2.4	3.7
4.0	5.7
4.9	9.9
5.7	6.9
6.3	15.8
7.8	15.4
9.0	36.9
9.3	34.6
11.0	53.2

In [14]:
```
sc = 0.9
options(repr.plot.width=7.5*sc, repr.plot.height=6*sc)
plot(XY, xlab = "X", ylab = "Y",pch=16)
```



5. Point estimation

- Assume that $\{X_1, X_2, \ldots, X_N\} = \{X_n\}_{n=1}^N$ is a sample of N observations from a population.
- Assume that any observation X taken from the population is a random variable with a pdf (or pmf) $f(x;\theta)$.
- Here, θ is an unknown "parameter" which we want to estimate.

Examples:

- You toss a coin 100 times. Let X_n be the indicator of getting heads in the n-th flip. Then, $X_n \sim \text{Bern}(p)$ and p is unknown. In the notation above, $\theta = p$.
- The past 300 daily returns of a stock are $R_1, \ldots, R_{300}$. You are certain that the daily returns are normally distributed, but, you don't know the mean and variance, $R_t \sim N(\mu, \sigma^2)$. In the notation above, $\theta = (\mu, \sigma^2)$. So, we can have more than one unknown parameter.
- You believe that your customers arrive according to a Poisson process but you do not know the rate λ. You have the time of arrivals of the past 1000 customers, $T_1, \ldots, T_{1000}$. In this case, we know that $X_n = T_n - T_{n-1}$ is an $\text{Exp}(\lambda)$ random variable.
- So, sometimes we need to do some "pre-processing" to get to our sample. Here, λ is our unknown parameter and the inter-arrival times $T_1, X_2, \ldots, X_{1000}$ is our sample.

Definition

Given a sample $X_1, \ldots, X_N$:

- A **statistic** is a random variable of the form $Y = g(X_1, \ldots, X_N)$.
- A statistic used for estimating an unknown parameter is called a **point estimator.**
- A specific value of a point estimator (by plugging in the observed value of the sample) is called a **point estimate**.
- So, a statistic and a point estimator are both random variables, while a point estimate is a number.

- It is a convention to use $\hat{\theta}$ to represent the point estimator for a parameter θ. Note that $\hat{\theta} = g(X_1, \ldots, X_N)$, and our job is to find the function $g(x_1, \ldots, x_N)$.

- Assume that we want to estimate the parameter λ of an Exp(λ) sample $X_1, \ldots, X_N$.
- We have (at least) two options:
  - Since $\mathbb{E}[X_n] = \frac{1}{\lambda} \approx \overline{X}$, we can set $\hat{\lambda} = \frac{1}{\overline{X}}$.
  - Since Var$(X_n) = \frac{1}{\lambda^2} \approx S_X^2$, we can set $\hat{\lambda} = \frac{1}{S_X}$.

In [15]:
```
X = rexp(20, rate=2)
head(round(X, digits=2), n=20)
c(round(1/mean(X), digits=2), round(1/sd(X), digits=2))
```

0.62 · 0.36 · 0.79 · 0.28 · 0.26 · 0.51 · 0.05 · 0.98 · 0.04 · 0.51 · 0.22 · 0.48 · 1.39 · 0.14 · 0.51 · 0.86 · 0.24 · 0.47 · 0.05 · 0.9

2.07 · 2.8

We have the following two tasks:

- Given a point estimator, how do we determine how good it is?
- What methods can be used to construct good point estimators?