Introduction to Applied Statistics
STAT 5005
Lecture 2: Data Description (Chapter 3)

Jingyu Sun

Fall 2021

Introduction

Data Description

- Statistics is the science of **Learning from Data**, which consists of four steps:

1. Defining the problem
2. Collecting the data
3. **Summarizing the data** (this chapter)
4. Analyzing the data

- In general, we do not observe all units in the population, a sample is selected from the population
- We use the information in the sample to draw conclusions about the population from which the sample was drawn
- The validity of the **inferences** about the population depends on the quality of the sample (cf. sampling designs in previous chapter)
- This requires organizing, summarizing, and describing the data by reducing a large set of measurements to a few summary measures
  - Example: controlling product descriptions "contains at least 50% juice"
- The two major methods for describing a set of measurements are *graphical techniques* and *numerical descriptive techniques*

Statistics and Computing

- Most of the calculations involved in statistical methods must be performed on computers due to the size of modern data sets
- Most of software systems (SAS, Minitab, R, Python, ...) can generate plots, data descriptions, and complex statistical analyses
- Despite initial investment of time and (often) irritation, most people find that they can use any particular system easily
- In this course, we will be using R, but the focus is not on coding or mastering a statistical software, but on interpreting the output from the software
- Be mindful (even skeptical) when reading computer outputs: Did anything go wrong? Was something overlooked?

Data frames

- Data frames are the most common data structure you'll deal with in R.
- A data frame is created with the `data.frame()` function:

```
mydata <- data.frame(col1, col2, col3,...)
```

where `col1` , `col2`, `col3`, and so on are variables

Here is one way to create a data frame:

```
patientID <- c(101, 102, 103, 104)
age <- c(25, 34, 28, 52)
diabetes <- c("Type1", "Type2", "Type1", "Type1")
status <- c("Poor", "Improved", "Excellent", "Poor")
patientdata <- data.frame(patientID, age, diabetes, status)
patientdata
```

```
  patientID age diabetes    status
1       101  25    Type1      Poor
2       102  34    Type2  Improved
3       103  28    Type1 Excellent
4       104  52    Type1      Poor
```

Graphical Methods for Describing Data on a Single Variable

- In many instances, the data can be arranged into categories so that **each measurement is classified into one, and only one, of the categories**
- Example: The `Arthritis` data set is included in the R package `vcd`. It contains 84 observations and 5 variables:
  - `ID`: patient ID
  - `Treatment`: factor indicating treatment (Placebo, Treated)
  - `Sex`: factor indicating sex (Female, Male)
  - `Age`: age of patient
  - `Improved`: ordered factor indicating treatment outcome (None, Some, Marked).

- For accessing the `Arthritis` data set, you first need to install the `vcd` package:

```
install.packages("vcd")
```

You do not need to write the line above once you installed `vcd`

- To display the data set, then type

```
library(vcd) # load the vcd package
View(Arthritis) # display the data
```

- First 6 observations in the data frame:

```
library(vcd)
```

```
## Warning: package 'vcd' was built under R version 4.0.5
```

```
head(Arthritis)
```

```
##   ID Treatment  Sex Age Improved
## 1 57   Treated Male  27     Some
## 2 46   Treated Male  29     None
## 3 77   Treated Male  30     None
## 4 17   Treated Male  32   Marked
## 5 36   Treated Male  46   Marked
## 6 23   Treated Male  58   Marked
```

- Some information on the variables:

```
str(Arthritis)
```

```
## 'data.frame':    84 obs. of  5 variables:
##  $ ID       : int  57 46 77 17 36 23 75 39 33 55 ...
##  $ Treatment: Factor w/ 2 levels "Placebo","Treated": 2 2 2 2 2 2 2 2 2 2 ..
##  $ Sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Age      : int  27 29 30 32 46 58 59 59 63 63 ...
##  $ Improved : Ord.factor w/ 3 levels "None"<"Some"<..: 2 1 1 3 3 3 1 3 1 1 .
```

- We can get the distribution of the `Improved` variable by giving the counts - the total number of patients in each category.

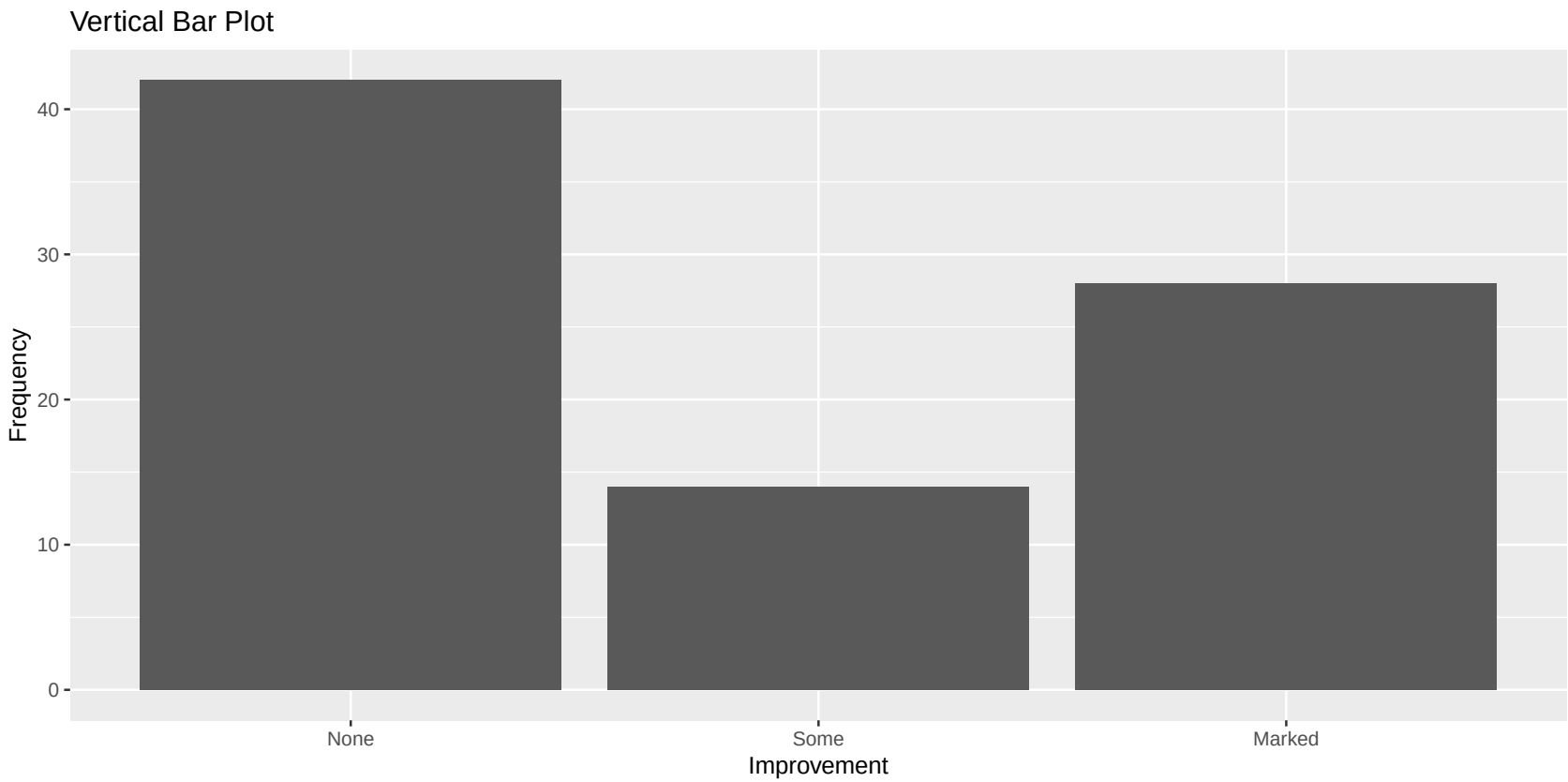
```
table(Arthritis$Improved)
```

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

- A total of 28 patients showed marked improvement, 14 showed some improvement, and 42 showed no improvement.

Bar Chart

- A **bar plot** or **bar chart** is useful to show the distribution of a categorical variable
- Each rectangle represents a category of the variable with a height equal to the frequency (number of observations) in the category
- A bar chart for the `Improved` variable is shown below:



The ggplot2 package

- The R package `ggplot2` is a flexible system that allows users to create new and innovative data visualizations
- To install `ggplot2`, type

```
install.packages("ggplot2")
```

- Then type

```
library(ggplot2)
```

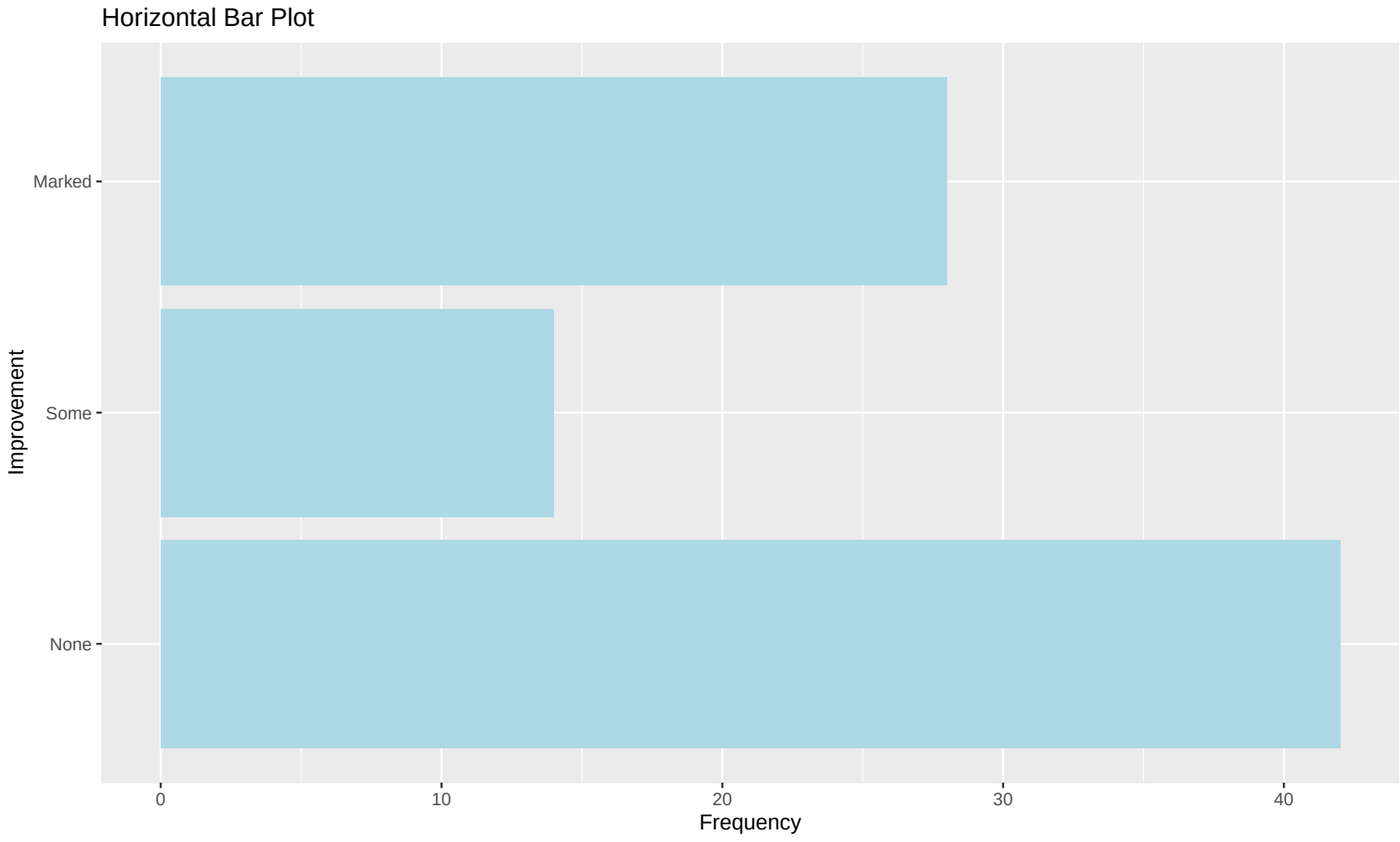
to load the package

R Code for Bar Chart

```
library(ggplot2)
ggplot(Arthritis, aes(x=Improved)) +
  geom_bar() +
  labs(title="Vertical Bar Plot", x="Improvement", y="Frequency")
```

- In ggplot2, plots are built in stages. You can sequence functions (such as geom_bar, labs, ...) for modifying the plot by *adding* them, using the + sign to separate the different function calls

```r
ggplot(Arthritis, aes(x=Improved)) +
  geom_bar(fill="lightblue") +
  labs(title="Horizontal Bar Plot", x="Improvement", y="Frequency") +
  coord_flip()
```



Pie Chart

- Example: **The Economist/YouGov Poll**, August 24-27, 2019

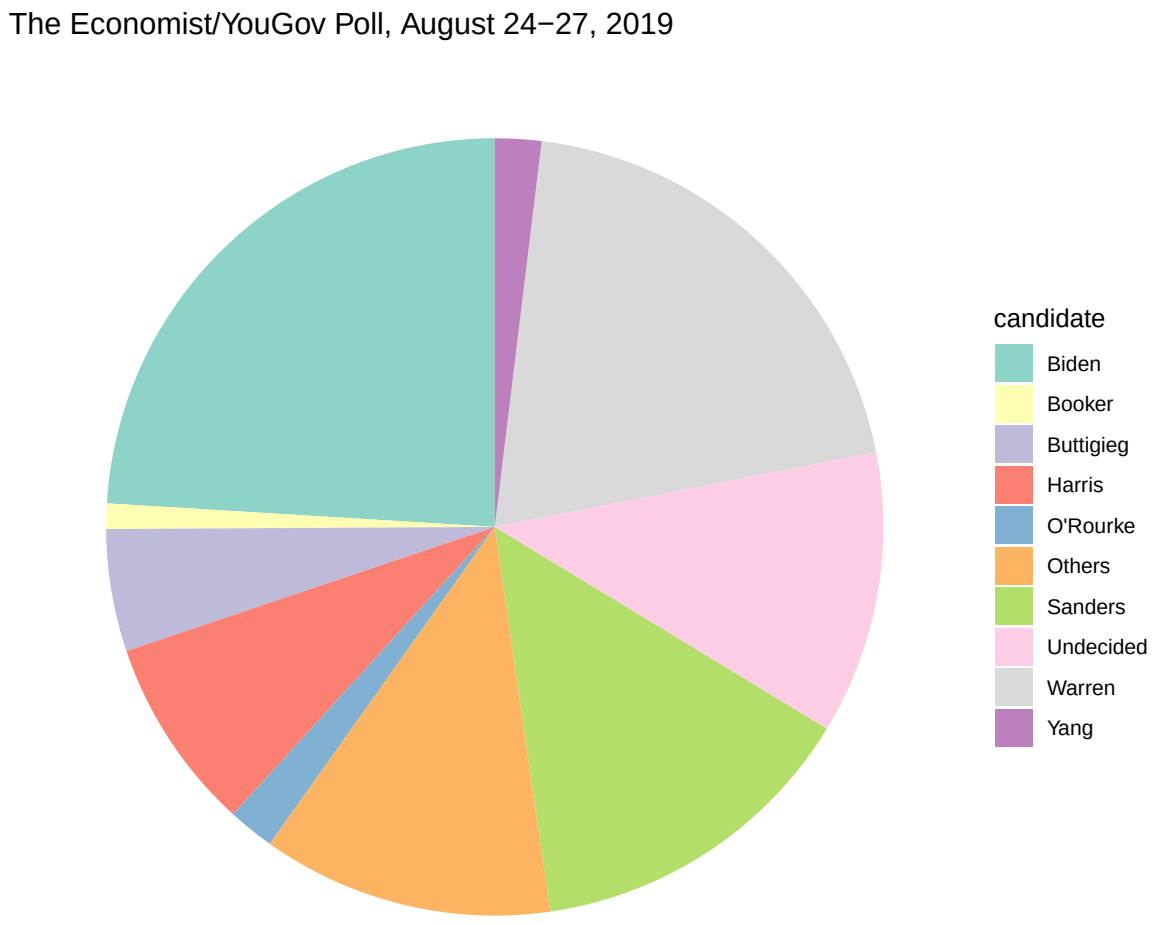
If the Democratic presidential primary or caucus in your state were held today, who would you vote for?

```
##    candidate votes
## 1      Biden   137
## 2     Booker     6
## 3  Buttigieg    29
## 4     Harris    46
## 5   O'Rourke    11
## 6    Sanders    80
## 7     Warren   114
## 8       Yang    11
## 9     Others    69
## 10 Undecided    67
```

- The R code for creating the poll data

```
candidate <- c("Biden","Booker","Buttigieg","Harris","O'Rourke",
               "Sanders","Warren","Yang","Others","Undecided")
votes <- c(137, 6, 29, 46, 11, 80, 114, 11, 69, 67)
poll <- data.frame(candidate, votes)
```

- A simple graphical procedure for this data is the **pie chart**
- Each slice represents a category of the variable with an area proportional to the frequency in the category



R Code for Pie Chart

```
ggplot(poll, aes(x="", y=votes, fill=candidate)) +
  geom_bar(stat="identity", width=1) + # create a basic bar chart
  coord_polar("y", start=0) + # convert to pie
  scale_fill_brewer(palette="Set3") + # add color scale
  labs(title = "The Economist/YouGov Poll, August 24-27, 2019") +
  theme_void()
```

- Results are more easily interpreted by using a pie chart
- Some guidelines:
    1. Choose a small number (five or six) of categories for the variable because too many make the pie chart difficult to interpret
    2. Use percentages to label the slices
    3. Whenever possible, construct the pie chart so that percentages are in either ascending or descending order

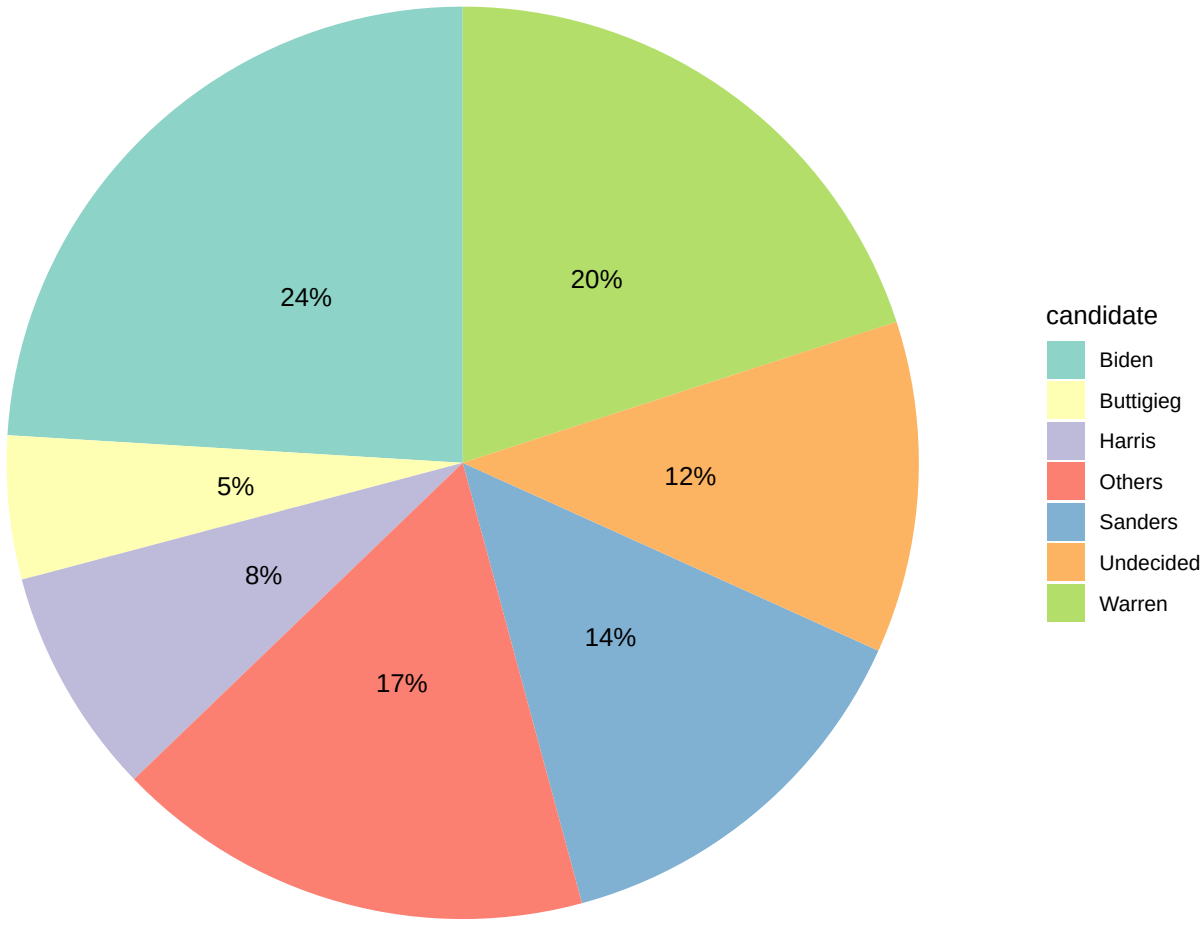
Improved Pie Chart

```r
# Create new data frame by keeping only the candidates with the highest scores
library(RColorBrewer)
candidate <- c("Biden","Buttigieg","Harris", "Sanders","Warren","Others",
               "Undecided")
votes <- c(137, 29, 46, 80, 114, 97, 67)
new.poll <- data.frame(candidate, votes)




ggplot(new.poll, aes(x="", y=votes, fill=candidate)) +
  geom_bar(stat="identity", width=1) + # create a basic bar
  coord_polar("y", start=0) + # convert to pie
  geom_text(aes(label = paste0(round(votes/sum(votes)*100), "%")),
                position = position_stack(vjust = 0.5)) + # label slices with %
  scale_fill_brewer(palette="Set3") + # add color scale
  labs(title = "The Economist/YouGov Poll, August 24-27, 2019") +
  theme_void()
```

The Economist/YouGov Poll, August 24−27, 2019



Histogram

- **Frequency histograms** are graphical techniques applicable only to *quantitative* data
- Example: The data set `singer` that in the `lattice` package contains the heights and voice parts of singers in the New York Choral Society
- To display the data set in R: type

```
library(lattice)
View(singer)
```

- How are the heights of the singers distributed?

Creating a histogram

Let n denote the total number of measurements

1. Group the data into a set of classes, called **class intervals**, typically of equal length

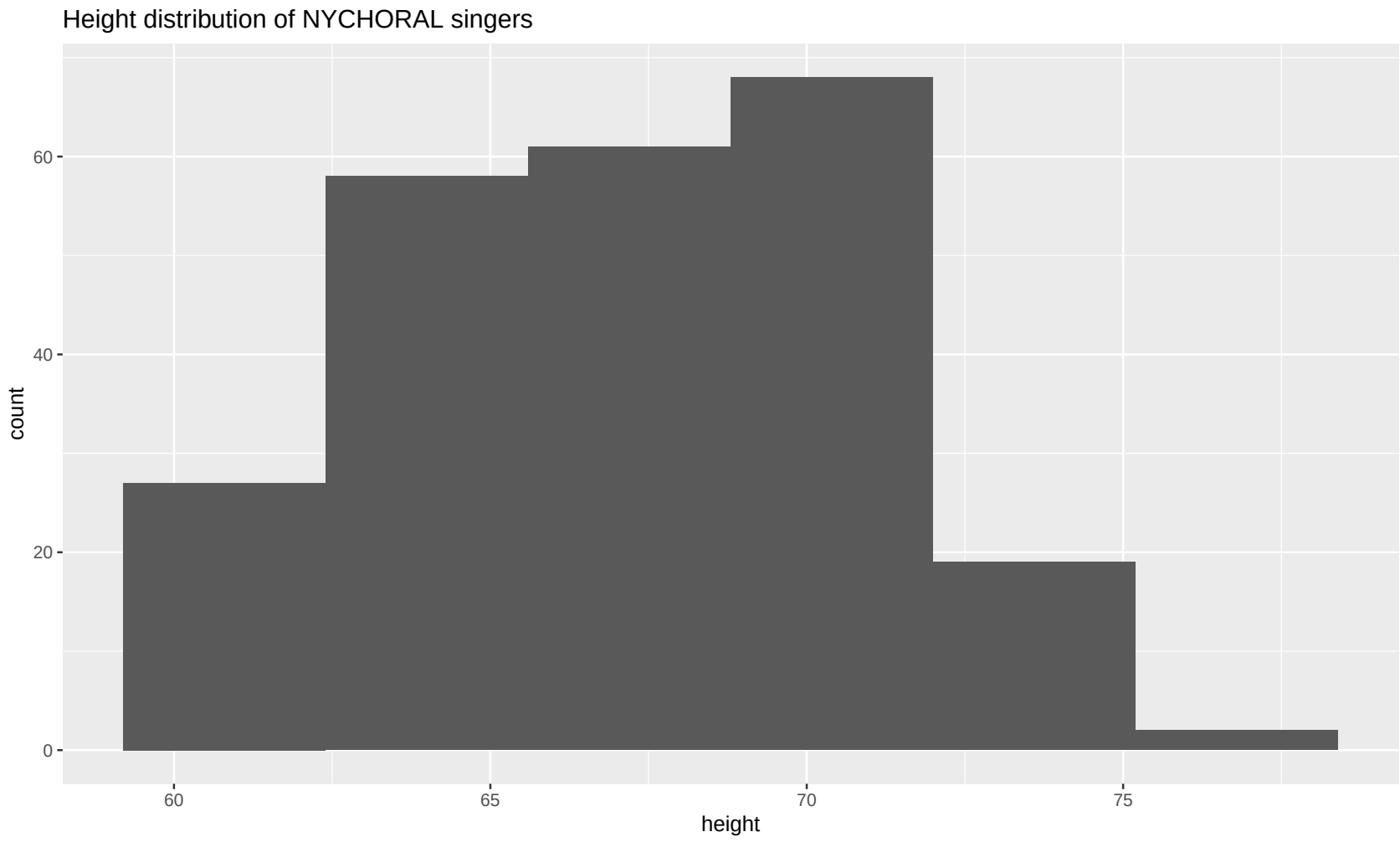
Rely on your common sense to choose an appropriate number of class intervals

2. Determine the **class frequency** f_i, i.e. the number of measurements in each class interval i

3. The **relative frequency** of a class i is defined to be the frequency of the class divided by the total number of measurements, i.e. f_i/n

4. To construct a **frequency histogram** (resp. **relative frequency histogram**), draw a rectangle over each class interval with a height equal to the class frequency (resp. relative frequency)

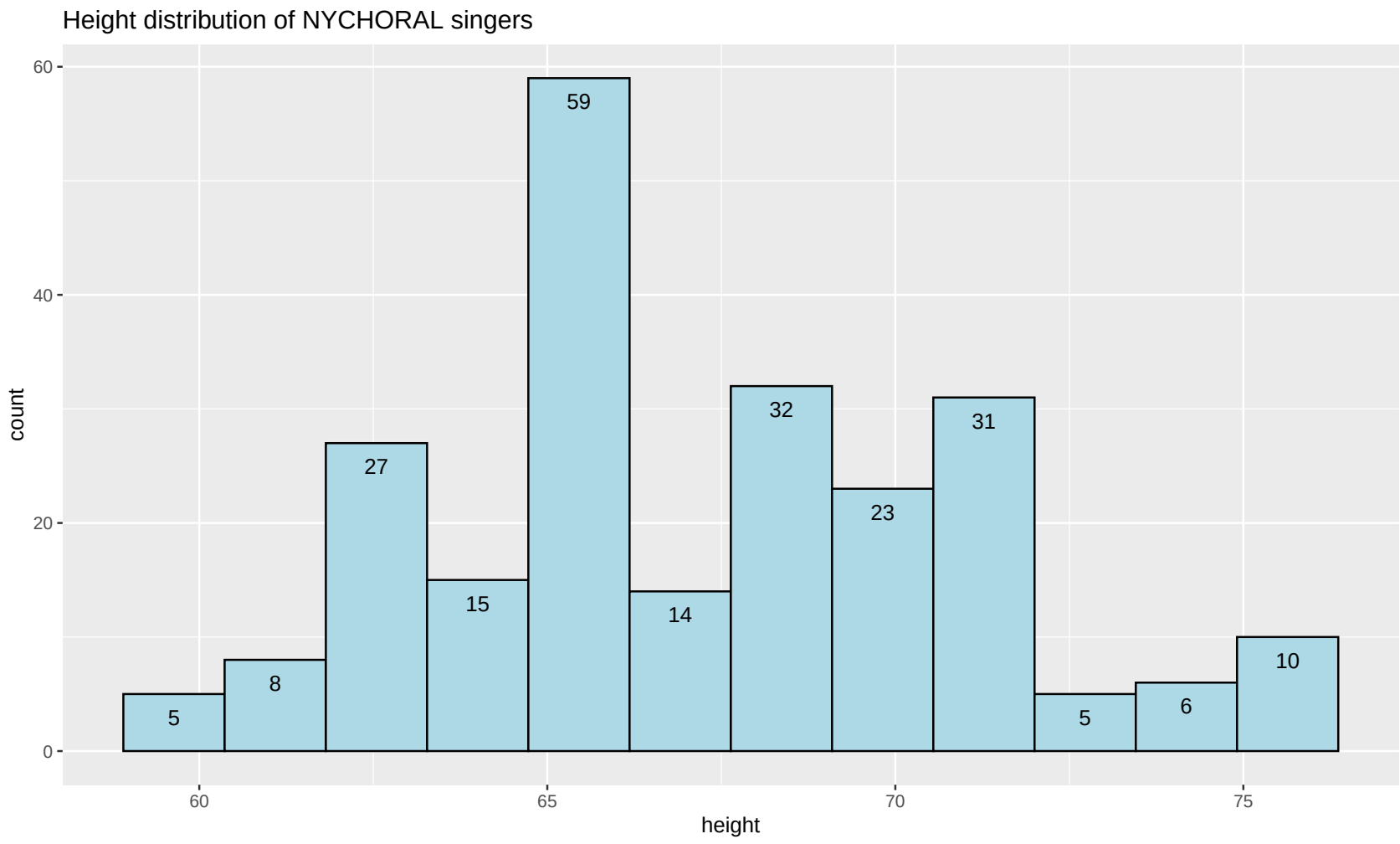
Frequency Histogram



R Code for Frequency Histogram

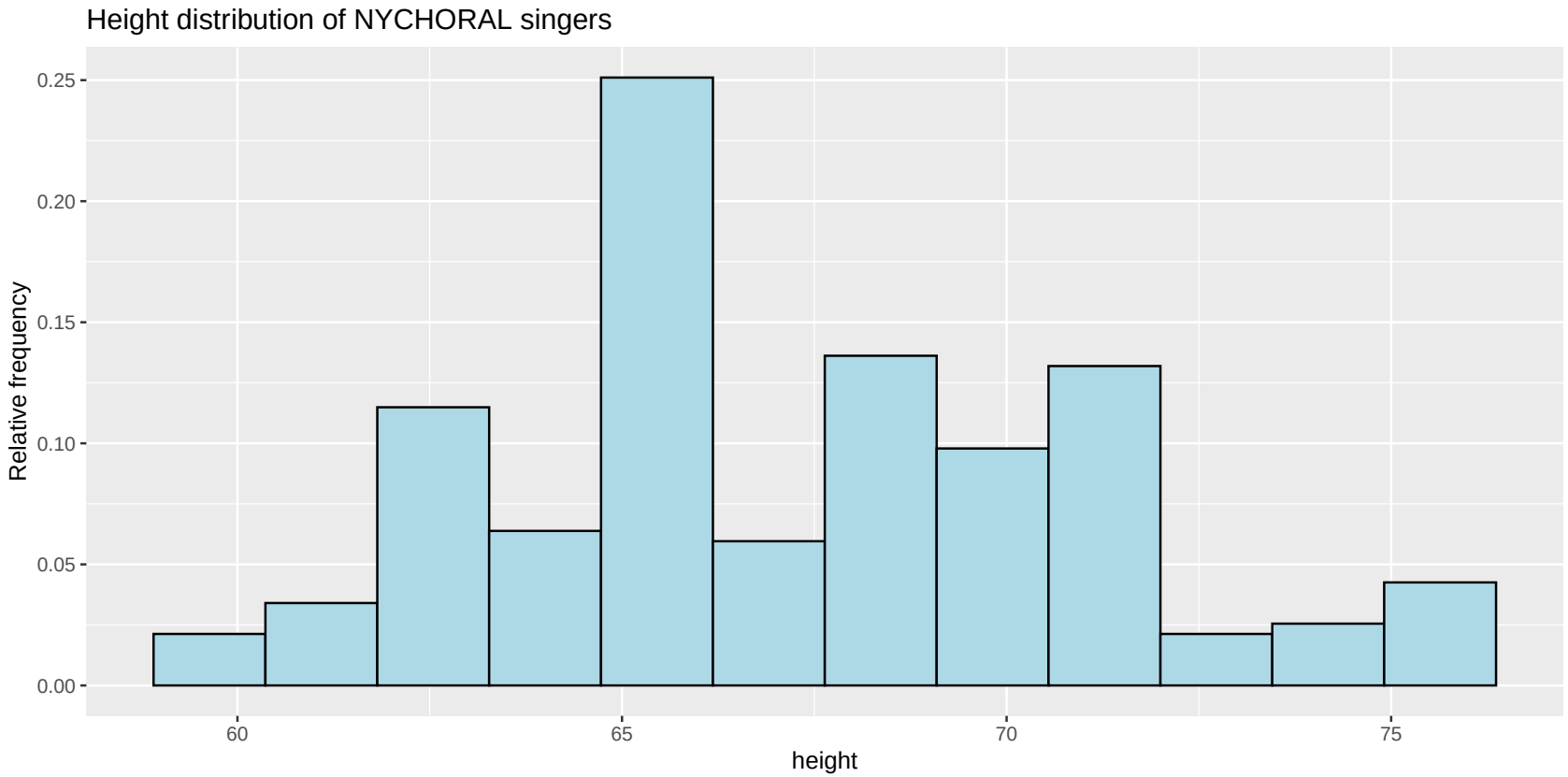
```
ggplot(singer, aes(x=height))+
  geom_histogram(bins = 6) +
  labs(title = "Height distribution of NYCHORAL singers")
```

```
ggplot(singer, aes(x=height))+
  geom_histogram(bins = 12, fill="lightblue", color="black") +
  stat_bin(aes(label=..count..), bins = 12, geom="text", vjust=2) +
  labs(title = "Height distribution of NYCHORAL singers")
```



Relative Frequency Histogram

```
ggplot(singer, aes(x=height))+
  geom_histogram(aes(y=..count../sum(..count..)),
                 bins = 12, fill="lightblue", color="black") +
  labs(title = "Height distribution of NYCHORAL singers",
       y = "Relative frequency")
```



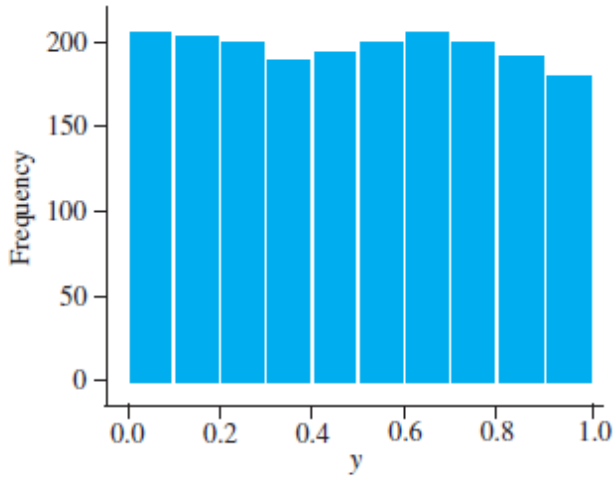
On the Importance of Histograms in Statistics

- The distinction between bar charts and histograms is based on the distinction between *qualitative* and *quantitative* variables
- Histograms play a major role in statistical inference
- If we had an extremely large set of measurements, and if we constructed a histogram using many class intervals, the histogram for the set of measurements would be a smooth curve
- The fraction of the total number of measurements in an interval is equal to the fraction of the total area under the histogram over the interval

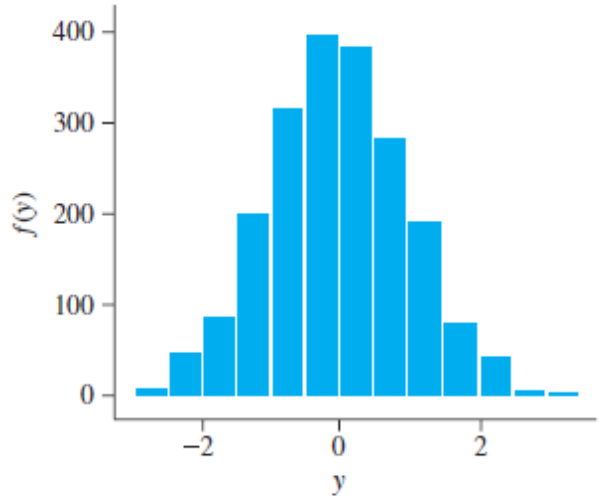
- If a single measurement is selected at random from the set of sample measurements, the chance that the selected measurement lies in a particular interval is equal to the fraction of the total number of sample measurements falling in that interval
- This same fraction is used to estimate the **probability** that a measurement selected from the population lies in the interval of interest

Shape Descriptors for Histograms

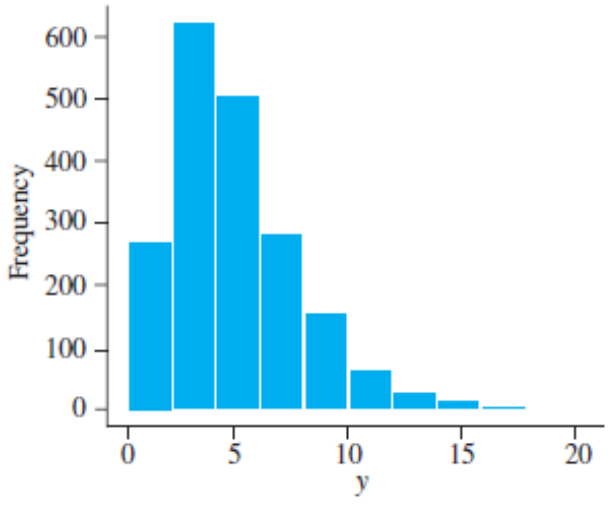
- Because we use proportions rather than frequencies in a relative frequency histogram, we can compare two different samples (or populations) by examining their relative frequency histograms even if the samples (populations) are of different sizes
- When describing relative frequency histograms and comparing the plots from a number of samples, we examine the overall shape in the histogram



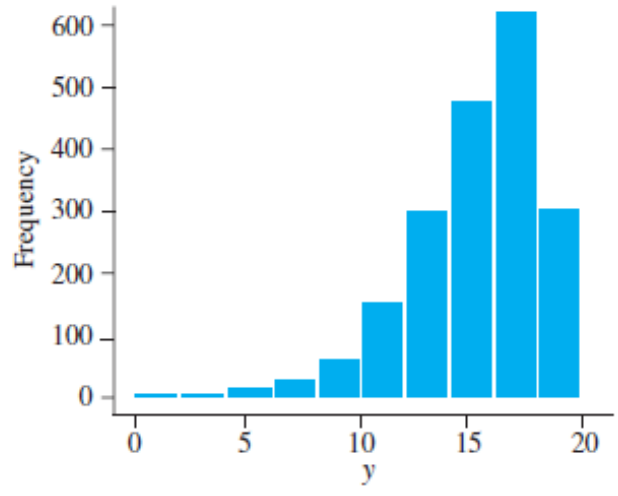
(a) Uniform distribution



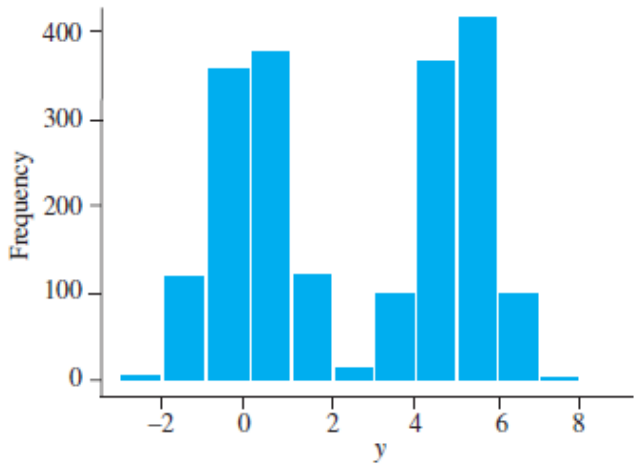
(b) Symmetric, unimodal (normal) distribution



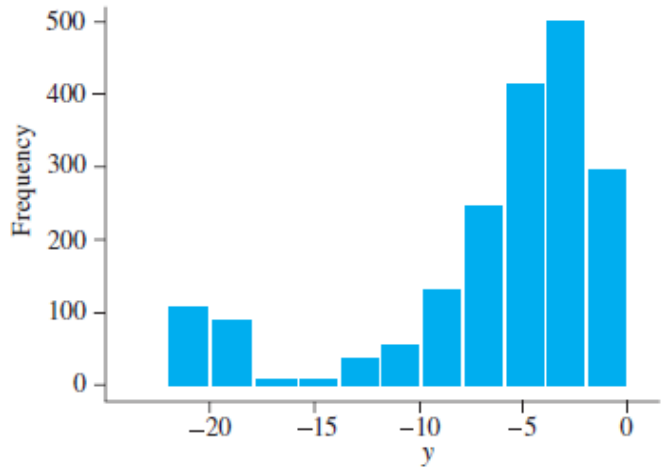
(c) Right-skewed distribution



(d) Left-skewed distribution



(e) Bimodal distribution



(f) Bimodal distribution skewed to left

- A histogram with one major peak is called **unimodal**, see Figures 3.8(b), (c), and (d)
- When the histogram has two major peaks, such as in Figures 3.8(e) and (f), the histogram is **bimodal**. Bimodal histograms are an indication that the sampled data are in fact from two distinct populations
- When every interval has essentially the same number of observations, the histogram is said to be **uniform**; see Figure 3.8(a)

- A histogram is **symmetric** in shape if the right and left sides have essentially the same shape (see Fig. 3.8(a), (b), and (e))
- When the right side (resp. left side) of the histogram, containing the larger half of the observations in the data, extends a greater distance than the left side (resp. right side), the histogram is referred to as **skewed to the right** (resp. skewed to the left); see Fig. 3.8 (c) (resp. 3.8 (d))

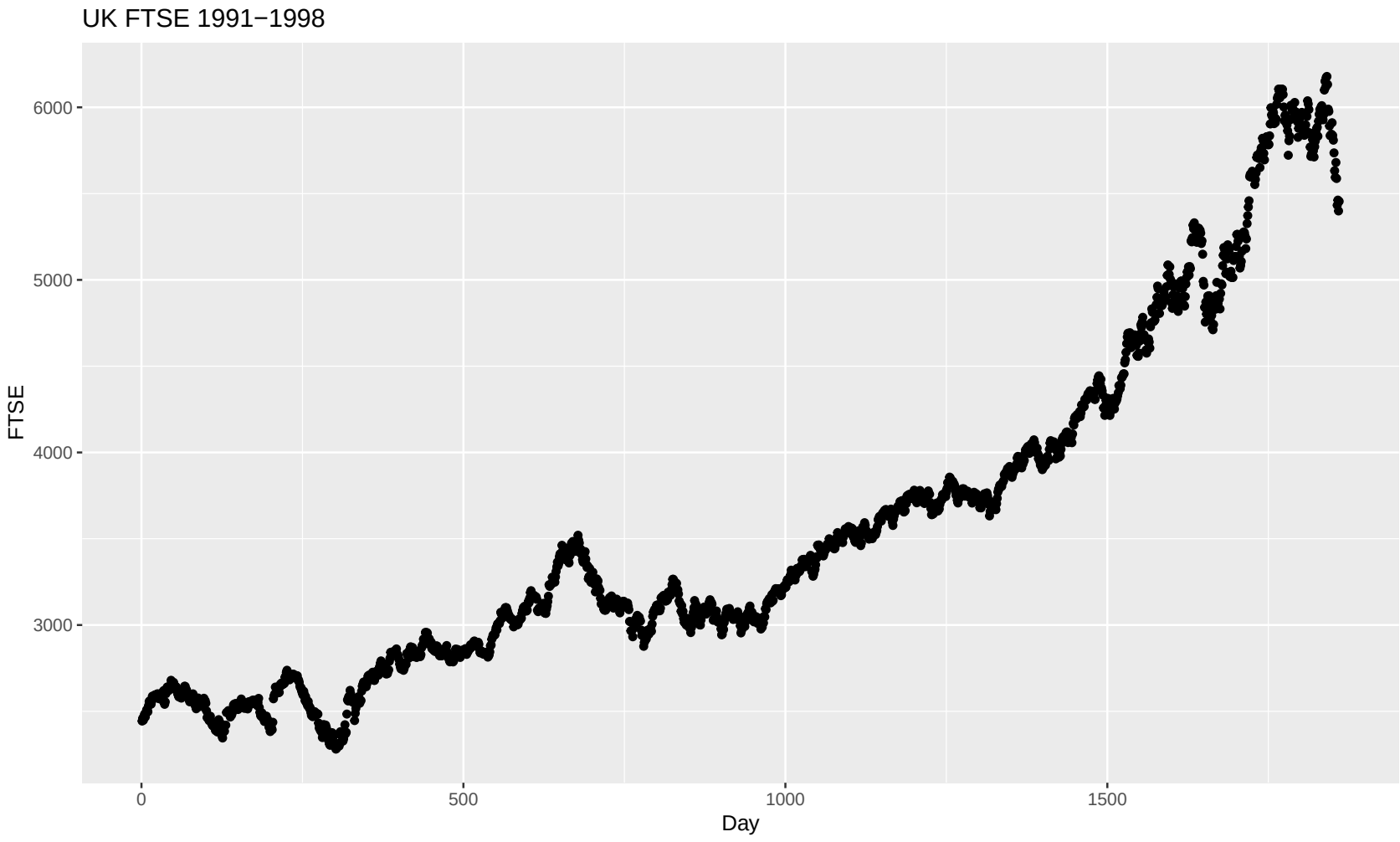
Time Series and Graphics

- A **time series** is a series of points that presents changes in a variable over time
- Usually, time points are labeled chronologically across the horizontal axis, and the numerical values of the variable of interest are labeled along the vertical axis
- These time points are usually equally spaced
- When information about a variable of interest is available in different units of time (e.g. hours, days, months), we must decide which unit or units are most appropriate

Example

- `EuStockMarkets` is a data set in R that contains the daily closing prices of major European stock indices, 1991-1998. The data are sampled in business time, i.e., weekends and holidays are omitted

```r
df <- as.data.frame(EuStockMarkets) # convert time series into data frame
ggplot(df, aes(x=1:nrow(df), y=FTSE)) +
  geom_point() +
  labs(title = "UK FTSE 1991-1998", x="Day")
```



- Time-series plots are useful for examining general trends and seasonal or cyclic patterns
- It also allows us to compare trends over time in a variable for two or more groups

General Guidelines for Successful Graphics

1. Before constructing a graph, set your priorities. What messages should the viewer get?
2. Choose the type of graph (pie chart, bar graph, histogram, and so on)
3. Pay attention to the title. One of the most important aspects of a graph is its title. The title should immediately inform the viewer of the point of the graph and draw the eye toward the most important elements of the graph
4. Fight the urge to use many type sizes, styles, and color changes. The indiscriminate and excessive use of different type sizes, styles, and colors will confuse the viewer. Generally, we recommend using only two typefaces; color changes and italics should be used in only one or two places

5. Convey the tone of your graph by using colors and patterns. Intense, warm colors (yellows, oranges, reds) are more dramatic than the blues and purples and help to stimulate enthusiasm by the viewer. On the other hand, pastels (particularly grays) convey a conservative, businesslike tone. Similarly, simple patterns convey a conservative tone, whereas busier patterns stimulate more excitement
6. Don't underestimate the effectiveness of a simple, straightforward graph

Numerical Descriptors for Data on a Single Variable

Describing Data on a Single Variable: Measures of Central Tendency

- **Numerical descriptive measures** are commonly used to convey a mental image of pictures, objects, and other phenomena
- The two most common numerical descriptive measures are **measures of central tendency** and **measures of variability**
- Measures of central tendency describe the center of the distribution of measurements
- Measures of variability describe how the measurements vary about the center of the distribution

- Fundamental distinction: numerical descriptive measures for a population are called **parameters**, and numerical descriptive measures for a sample are called **statistics**
- The first measure of central tendency we consider is the **mode**
  - The **mode** of a set of measurements is defined to be the measurement that occurs with highest frequency
- Examples: the mode is used as a measure of popularity that reflects central tendency or opinion
- Some distributions have more than one measurement that occurs with the highest frequency

- The **median** of a set of measurements is defined to be the middle value when the measurements are arranged from lowest to highest
- The median value divides the set of measurements into two groups, with an equal number of measurements in each group
- The median for an even number of measurements is the average of the two middle values when the measurements are arranged from lowest to highest
- Example: Scores at a test are 95 86 78 90 62 73 89 92 84 76

Determine the median score.

- The **arithmetic mean**, or **mean**, of a set of measurements is defined to be the sum of the measurements divided by the total number of measurements
- Notations:
  - The population mean is denoted by the Greek letter μ
  - The sample mean is denoted by the symbol $\bar{y}$
- Let $y_1, \ldots, y_n$ be the measurements observed in a sample of size n. Then the sample mean $\bar{y}$ can be calculated as

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} = \frac{y_1 + \ldots + y_n}{n}$$

- The mean is a useful measure of the central value of a set of measurements, but it is subject to distortion due to the presence of one or more extreme values in the set, called **outliers**
- A **trimmed mean** drops the highest and lowest extreme values and averages the rest
- How are the mode, median, mean, and trimmed mean related for a given set of measurements? The answer depends on the *skewness* of the data

R Code for Median, Mean and Trimmed Mean

- Example: Scores at a test are 95 86 78 90 62 73 89 92 84 76

```
score <- c(95,86,78,90,62,73,89,92,84,76)
median(score) # median
```

```
## [1] 85
```

```
mean(score) # mean
```

```
## [1] 82.5
```

```
# Trimmed mean: drop the 2 smallest and 2 largest values
mean(score, trim = 0.2)
```

```
## [1] 83.83333
```

Describing Data on a Single Variable: Measures of Variability

- It is not sufficient to describe a data set using only measures of central tendency, such as the mean or the median
- Example: All the histograms have the same mean but each has a different spread, or variability, about the mean
- The **range** of a set of measurements is defined to be the difference between the largest and the smallest measurements of the set

- The pth percentile of a set of n measurements arranged in order of magnitude is that value that has at most $p\%$ of the measurements below it and at most $(100 - p)\%$ above it
- Specific percentiles of interest are the 25th, 50th, and 75th percentiles, often called the *lower quartile*, the *middle quartile* (median), and the *upper quartile*, respectively

Computation of Percentiles

- Each data value corresponds to a percentile for the percentage of the data values that are less than or equal to it
- Let $y_{(1)}, y_{(2)}, \ldots, y_{(n)}$ denote the ordered observations for a data set, that is

$$y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(n)}$$

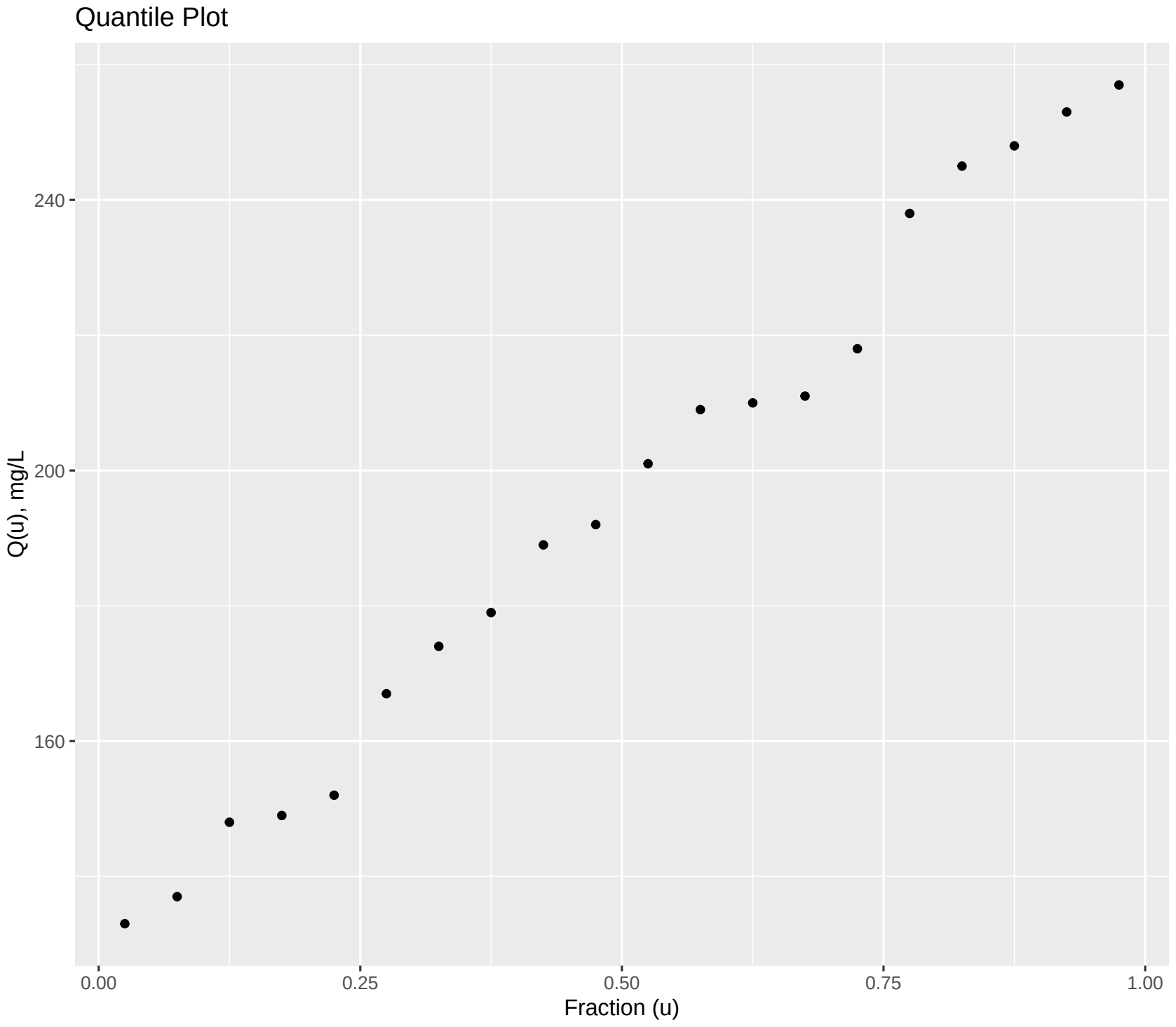
- The ith ordered observation corresponds to the $100(i - 0.5)/n$ percentile

Example

- Serum cholesterol levels for 20 adult patients

Observation (i)	Cholesterol (mg/L)	Percentile
1	133	2.5
2	137	7.5
3	148	12.5
4	149	17.5
5	152	22.5
6	167	27.5
7	174	32.5
8	179	37.5
9	189	42.5
10	192	47.5
11	201	52.5
12	209	57.5
13	210	62.5
14	211	67.5
15	218	72.5
16	238	77.5
17	245	82.5
18	248	87.5
19	253	92.5
20	257	97.5

- **Quantiles** are a generalization of percentiles
- A quantile, denoted $Q(u)$, is a number that divides a sample of n data values into two groups so that the specified fraction u of the data values is less than or equal to the value of the quantile, $Q(u)$
- Equivalence: the 80th percentile is the 0.8 quantile
- Quantile plot: plot $y_{(i)}$ versus $u_i = (i - 0.5)/n$, for $i = 1, \ldots, n$


Quantile Plot
240
200
160
Q(u), mg/L
0.00
0.25
0.50
0.75
1.00
Fraction (u)

R Code for Quantile Plot

```
cholesterol <- c(174,248,137,210,189,253,148,238,152,245,
                 192,209,211,133,257,179,149,167,218,201)
n <- length(cholesterol)
fraction <- (1:n-0.5)/n
df <- data.frame(cholesterol,fraction)


library(ggplot2)
ggplot(df, aes(x=fraction, y=sort(cholesterol))) +
  geom_point() +
  labs(title = "Quantile Plot", x="Fraction (u)", y="Q(u), mg/L")
```

R Code for Quantiles

```
y <- c(318,91,109,64,54,181,126,360,496,424,479,42,246,
       259,25,38,161,334,184,302)
# 34th percentile
quantile(y, 0.34)
```

```
##     34%
## 116.82
```

Interquartile Range

- The **interquartile range** (IQR) of a set of measurements is defined to be the difference between the upper and lower quartiles; that is,

$$\text{IQR} = \text{75th percentile-25th percentile}$$

- The IQR measures the distance needed to cover the middle 50% of the data values
- In most data sets, we would typically need five summary values to provide a minimal description of the data set: smallest value, $y_{(1)}$, lower quartile, $Q(0.25)$, median, upper quartile, $Q(0.75)$, and the largest value, $y_{(n)}$

- For a set of measurements, the **deviation** of a single measurement y from the mean $\bar{y}$ is the value $y - \bar{y}$
- Example: 5 measurements: $y_1 = 68, y_2 = 67, y_3 = 66, y_4 = 63,$ and $y_5 = 61$
- Many different measures of variability can be constructed by using the deviations $y - \bar{y}$
- The sample variance, s^2 of a set of n measurements $y_1, \ldots, y_n$ with mean $\bar{y}$ is

$$s^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}$$

* The sample variance, s^2 is an *unbiased estimator* of the population variance, σ^2. That means if we were to draw a very large number of samples, each of size n, from the population of interest and if we computed s^2 for each sample, the average sample variance would equal the population variance σ^2

- The **standard deviation** of a set of measurements is defined to be the positive square root of the variance
- s denotes the sample standard deviation and σ the corresponding population standard deviation
- In the example above, $\bar{y} = 65, s^2 = 8.5, s = 2.915$

Sample Variance and Sample Standard Deviation in R

```
y <- c(68,67,66,63,61)
var(y) # variance
```

```
## [1] 8.5
```

```
sd(y) # standard deviation
```

```
## [1] 2.915476
```

The Empirical Rule

- A "mound-shaped" (or bell-shaped) histogram is a histogram that has a single peak, symmetrical, and tapers off gradually in the tails
- Give a set of n measurements possessing a mound-shaped histogram, then
  - the interval $\bar{y} \pm s$ contains approximately 68% of the measurements
  - the interval $\bar{y} \pm 2s$ contains approximately 95% of the measurements
  - the interval $\bar{y} \pm 3s$ contains approximately 99.7% of the measurements
- An approximate value for s is found by dividing the range by 4

Example

A recent study shows that the IQ scores recorded from a large sample in the population have a mound-shaped relative frequency histogram with a mean of 100 and a standard deviation of 15.

- What conclusions can we reach about the IQ scores in the population?

- The standard deviation can be deceptive when comparing the amount of variability of different types of populations
- The **coefficient of variation** (CV) is a unit-free number that measures the variability in the values in a population relative to the magnitude of the population mean
- In a process or population with mean μ and standard deviation σ, the coefficient of variation is defined as

$$\text{CV} = \frac{\sigma}{|\mu|}$$

provided $\mu \neq 0$

- For sampled data, we estimate CV with $s/|\bar{y}|$

Example

- Suppose we want to compare two production processes that fill containers with products
- Process A is filling fertilizer bags, which have a nominal weight of 80 pounds. The process produces bags having a mean weight of 80.6 pounds with a standard deviation of 1.2 pounds
- Process B is filling 24-ounce cornflakes boxes, which have a nominal weight of 24 ounces. Process B produces boxes having a mean weight of 24.3 ounces with a standard deviation of 0.4 ounces
- Is process A much more variable than process B?

The Boxplot

- The **boxplot** is a graphical representation of a set of numerical values concerned with the symmetry of the distribution and incorporates numerical measures of central tendency and location to study the variability of the values and the concentration of values in the tails of the distribution

Example

- A criminologist is studying whether there are wide variations in violent crime rates across the United States. Using Department of Justice data from 2000, the crime rates in 90 cities selected from across the United States were obtained
- The median M is 497.5
- Lower quartile $Q_1 = 397$ and upper quartile $Q_3 = 660$

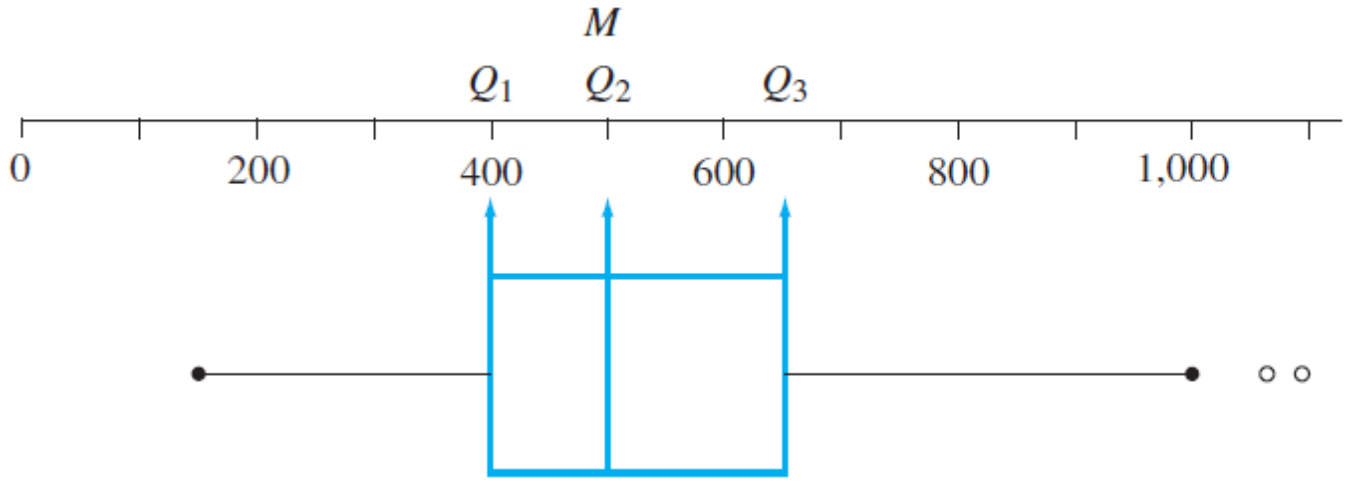
- The lower fence is the value defined as $Q_1 - 1.5(\text{IQR})$
- The upper fence is the value defined as $Q_3 + 1.5(\text{IQR})$
- Any data value below the lower fence or above the upper fence is called an **outlier**

1. Mark off a box from the lower quartile to the upper quartile
2. Draw a solid line across the box to locate the median
3. Two cases:

- If there are no outliers, a straight line is then drawn connecting the box to the largest value; a second line is drawn from the box to the smallest value
- If there are any outliers, a straight line is drawn connecting the box to the smallest number that is not an outlier; a second line is drawn from the box to the largest number that is not an outlier

4. Mark each outlier with a circle

- The center of the distribution of values is indicated by the median line in the boxplot
- A measure of the variability of the scores is given by the interquartile range, the length of the box
- By examining the relative position of the median line, we can gauge the symmetry of the middle 50% of the obervations
- Additional information about skewness is obtained from the lengths of the whiskers; the longer one whisker is relative to the other one, the more skewness there is in the tail with the longer whisker
- A general assessment can be made about the presence of outliers by examining the number of observations classified as outliers

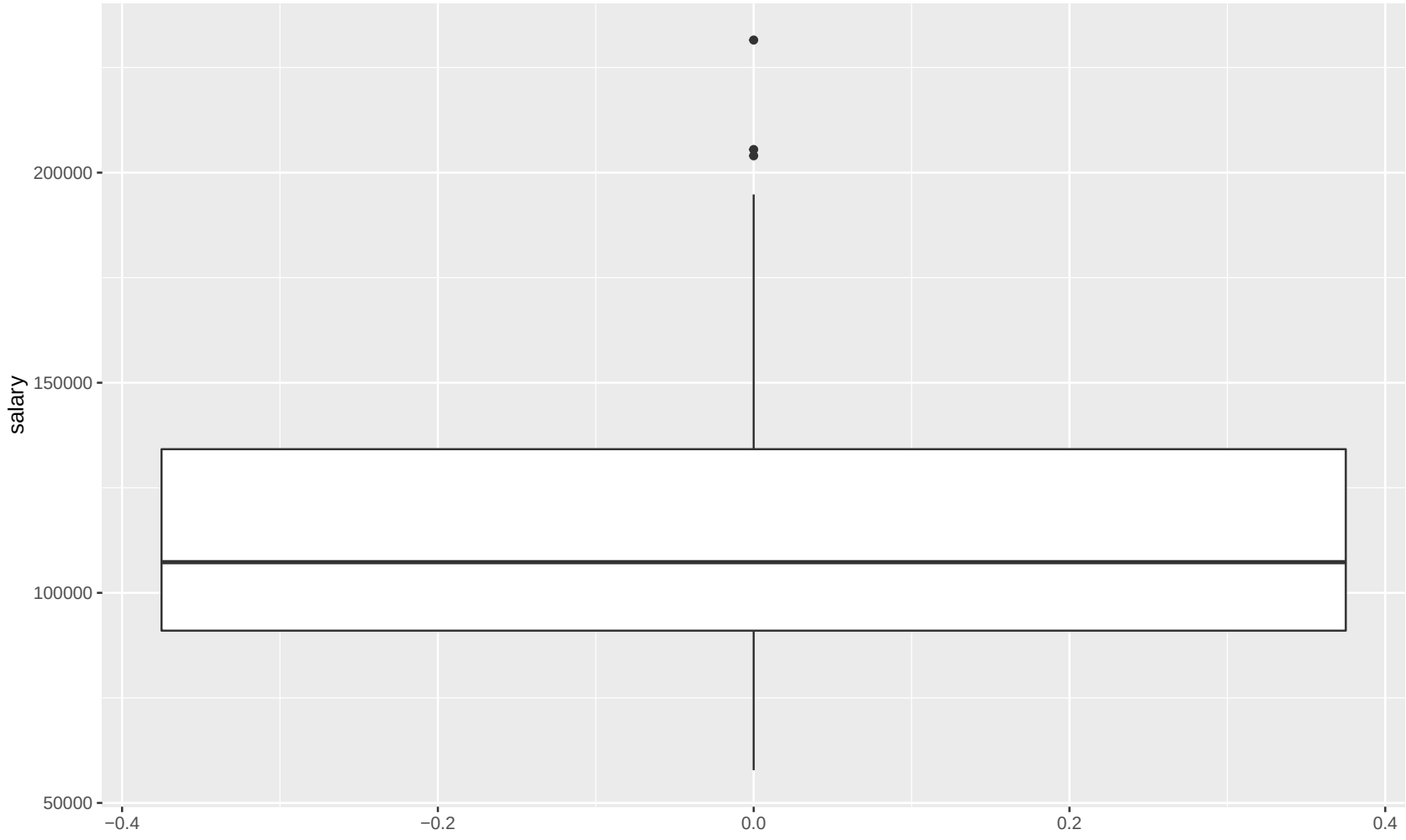


- Boxplot for the US crime rates data set

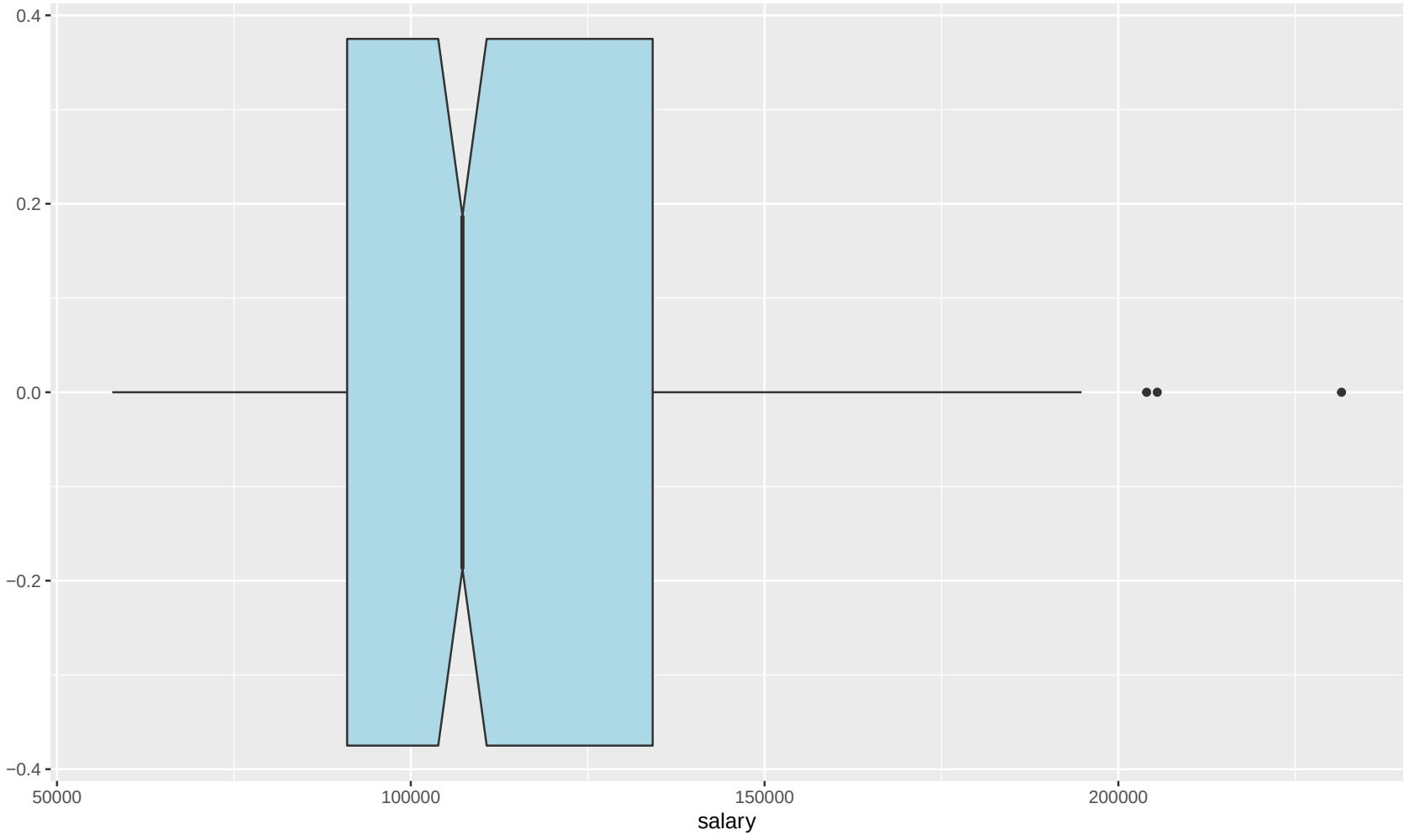
R Code for Boxplot

- The data set `Salaries` in the `carData` contains salary information for university professors and was collected to explore gender discrepancies in income

```
library(carData)
ggplot(Salaries, aes(y=salary)) +
  geom_boxplot()
```

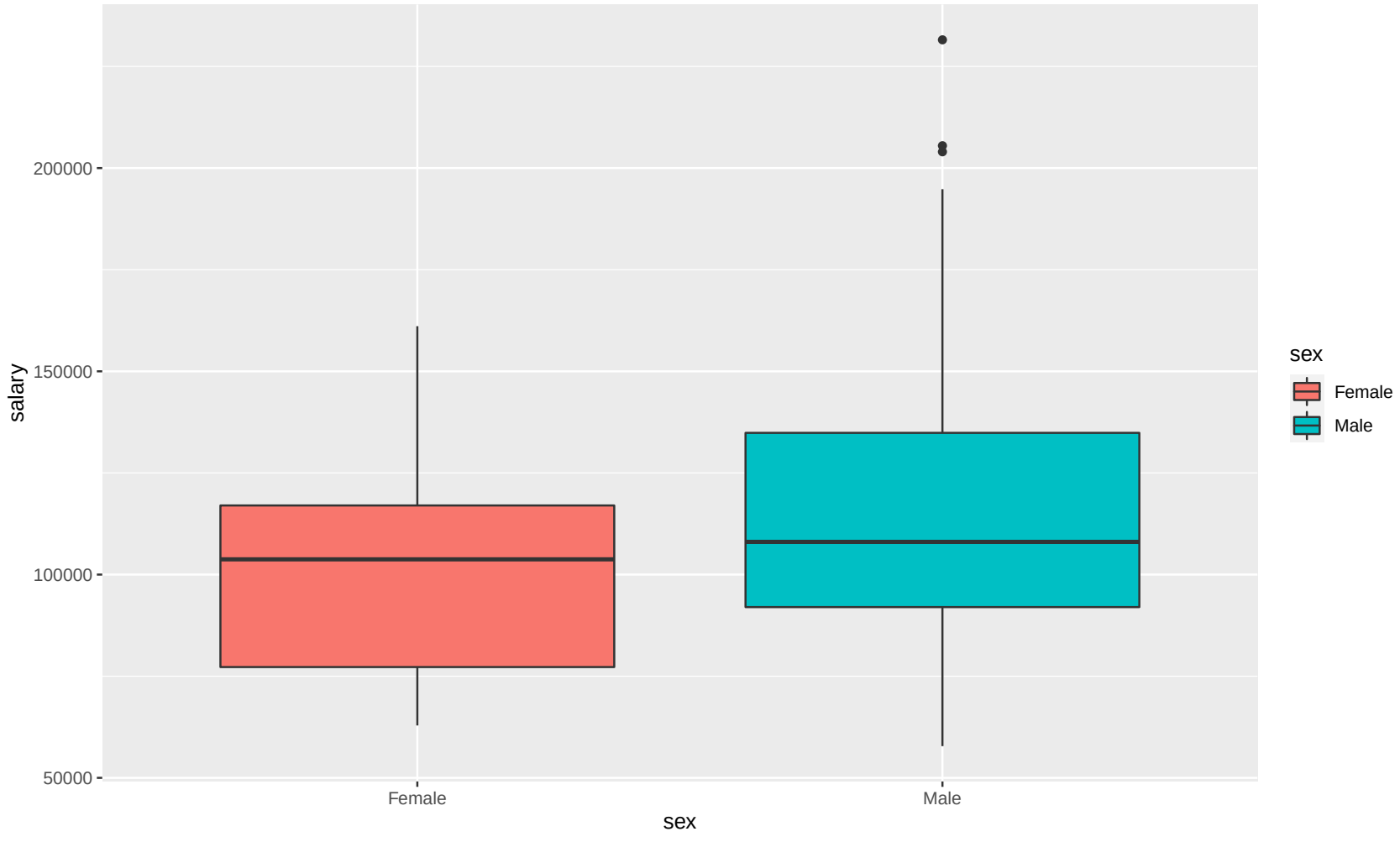


```
ggplot(Salaries, aes(y=salary)) +
  geom_boxplot(fill="lightblue", notch = TRUE) +
  coord_flip()
```



- Boxplots provide a powerful graphical technique for comparing samples from several different treatments or populations, by placing boxplots *side-by-side*

```
ggplot(Salaries, aes(x=sex, y=salary, fill=sex)) +
  geom_boxplot()
```



- What can you infer from that plot?

Summarizing Data from More Than One Variable

Summarizing Data from More Than One Variable

- Frequently, more than one variable is being studied at the same time, and we might be interested in summarizing the data on each variable separately, and also in studying relations among the variables
- Material in this section will provide a brief preview and introduction to contingency tables (Chap. 10), analysis of variance (Chap. 8 and 14 -18), and regression (Chap. 11, 12, and 13)

Summarizing data from two qualitative variables

- In the Arthritis data set, suppose we want to check whether the distribution of improvement levels was different for the drug and the placebo.
- First we have to create a 2-way frequency table, cross-tabulating treatment type and improvement status

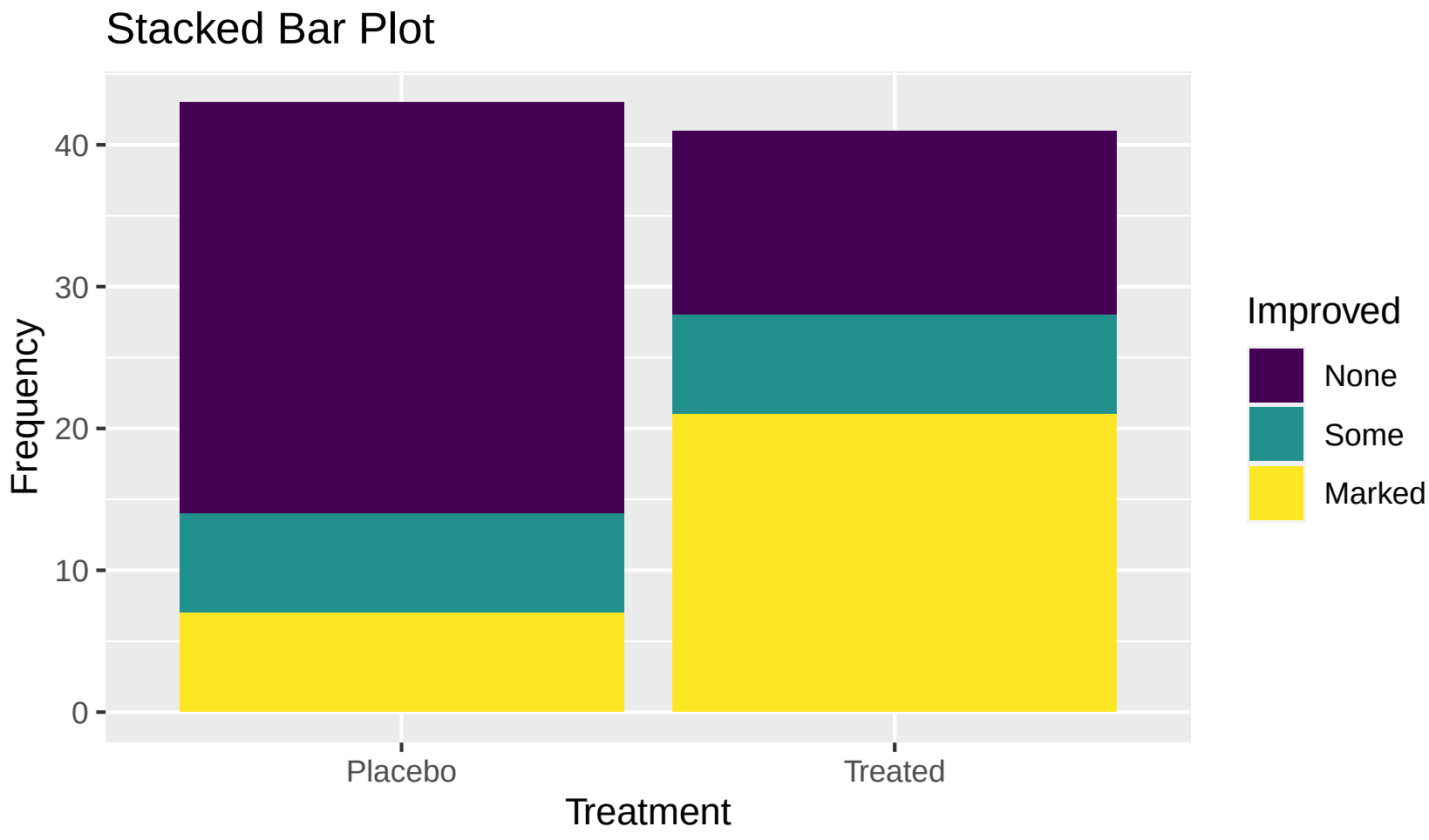
```
table(Arthritis$Improved, Arthritis$Treatment)
```

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

- This table is called a **contingency table**
- The rows of the table identify the categories of one variable, and the columns identify the categories of the other variable. The entries in the table are the number of times each value of one variable occurs with each possible value of the other

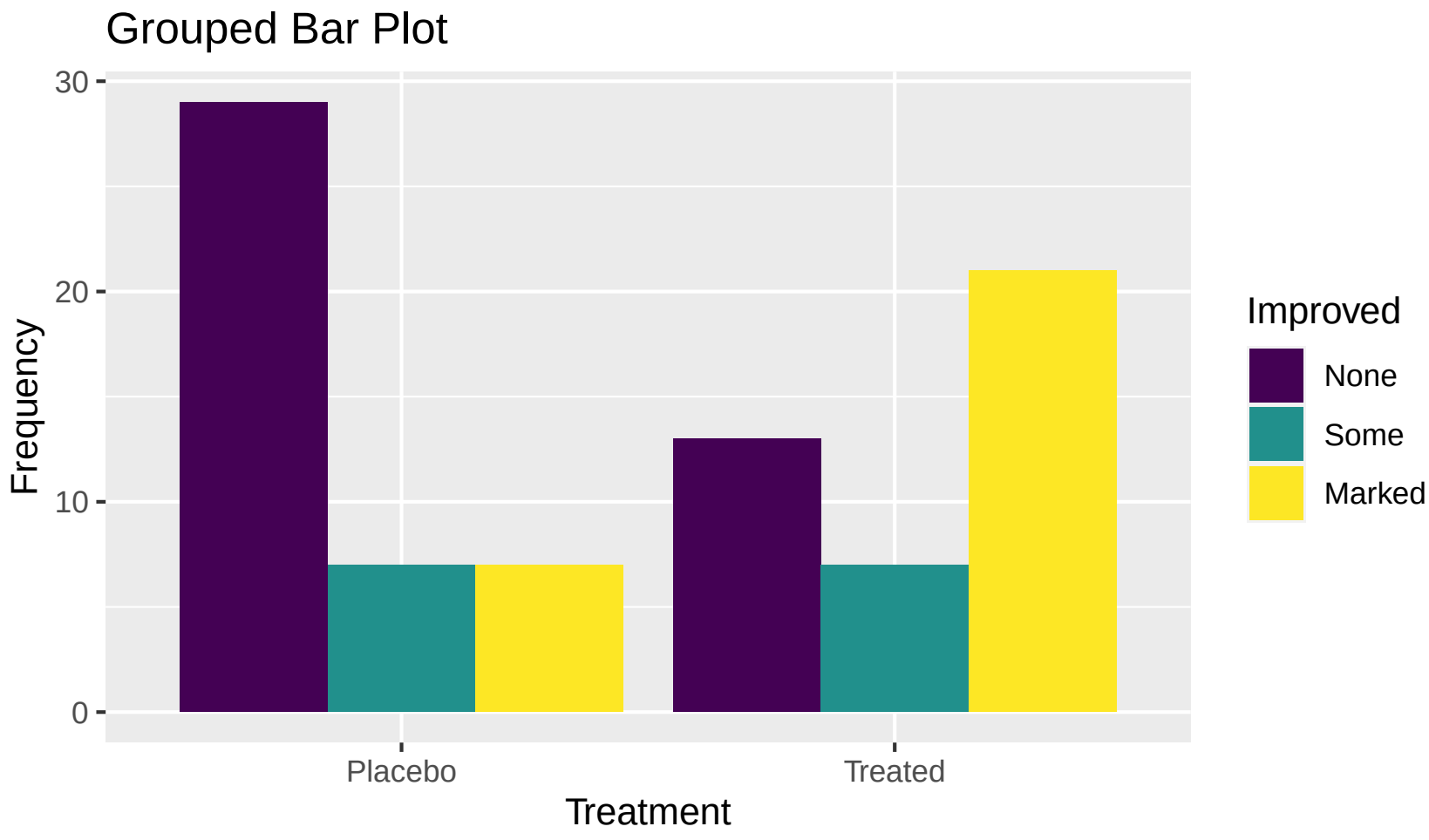
- A **stacked bar graph** provides a convenient method for displaying data from a pair of qualitative variables

```
ggplot(Arthritis, aes(x=Treatment, fill=Improved)) +
  geom_bar() +
  labs(title="Stacked Bar Plot", x="Treatment", y="Frequency")
```



- A **grouped barplot** or **cluster bar graph** also allows one to examine two factors simultaneously

```
ggplot(Arthritis, aes(x=Treatment, fill=Improved)) +
  geom_bar(position = "dodge") +
  labs(title="Grouped Bar Plot", x="Treatment", y="Frequency")
```



- A **scatterplot** is a plot used to give the general shape and direction of the relationship between two *quantitative* variables
- Example: Is there a relationship between hourly wage offered and years of experience?

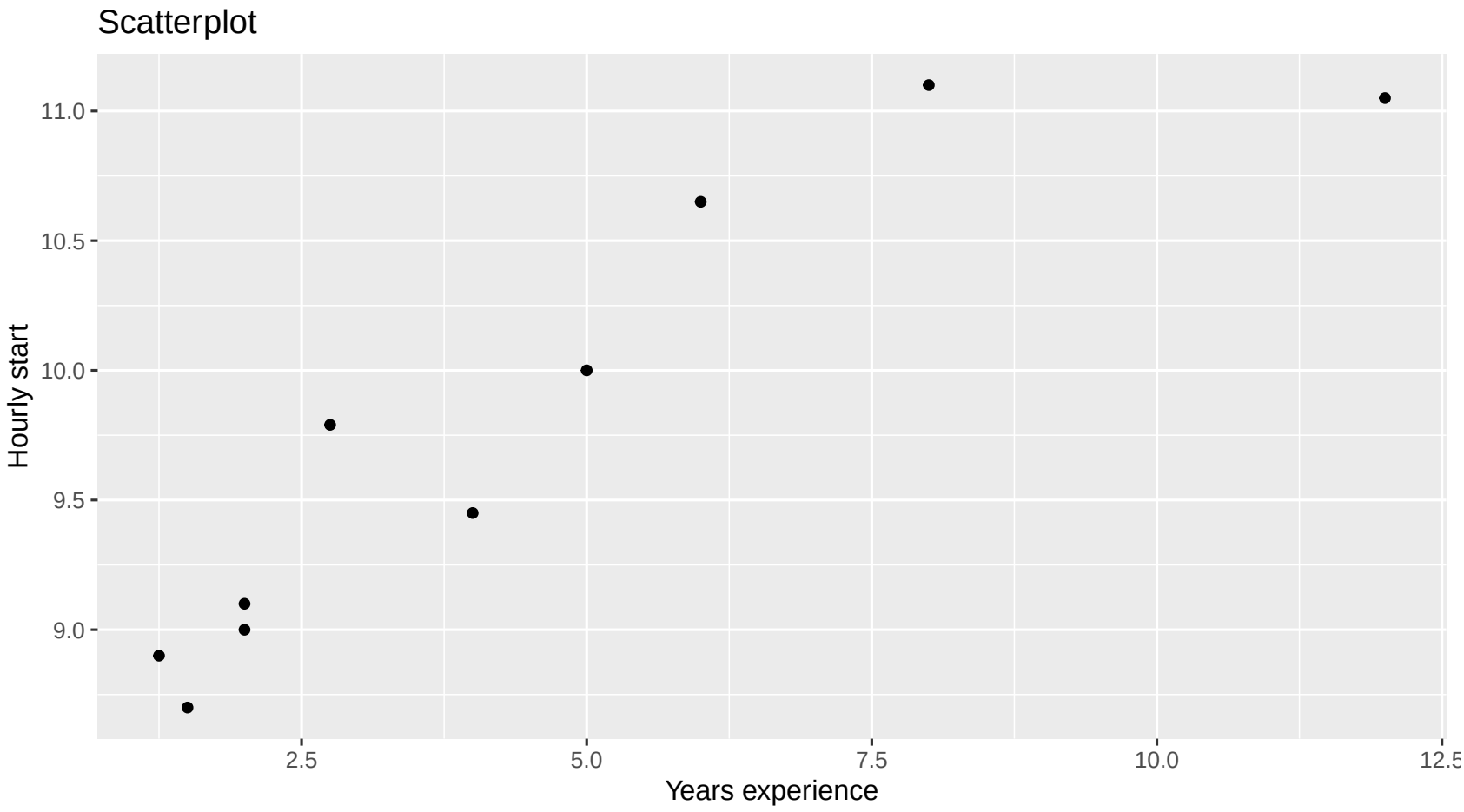
Hourly wage in dollars	Years of experience
8.90	1.25
8.70	1.50
9.10	2.00
9.00	2.00
9.79	2.75
9.45	4.00
10.00	5.00
10.65	6.00
11.10	8.00
11.05	12.00

- Create the wage data frame

```
hrly.start <- c(8.90,8.70,9.10,9.00,9.79,9.45,10.00,10.65,11.10,11.05)
yrs.exp <- c(1.25,1.50,2.00,2.00,2.75,4.00,5.00,6.00,8.00,12.00)
wage <- data.frame(yrs.exp,hrly.start)
```

R Code for Scatterplot

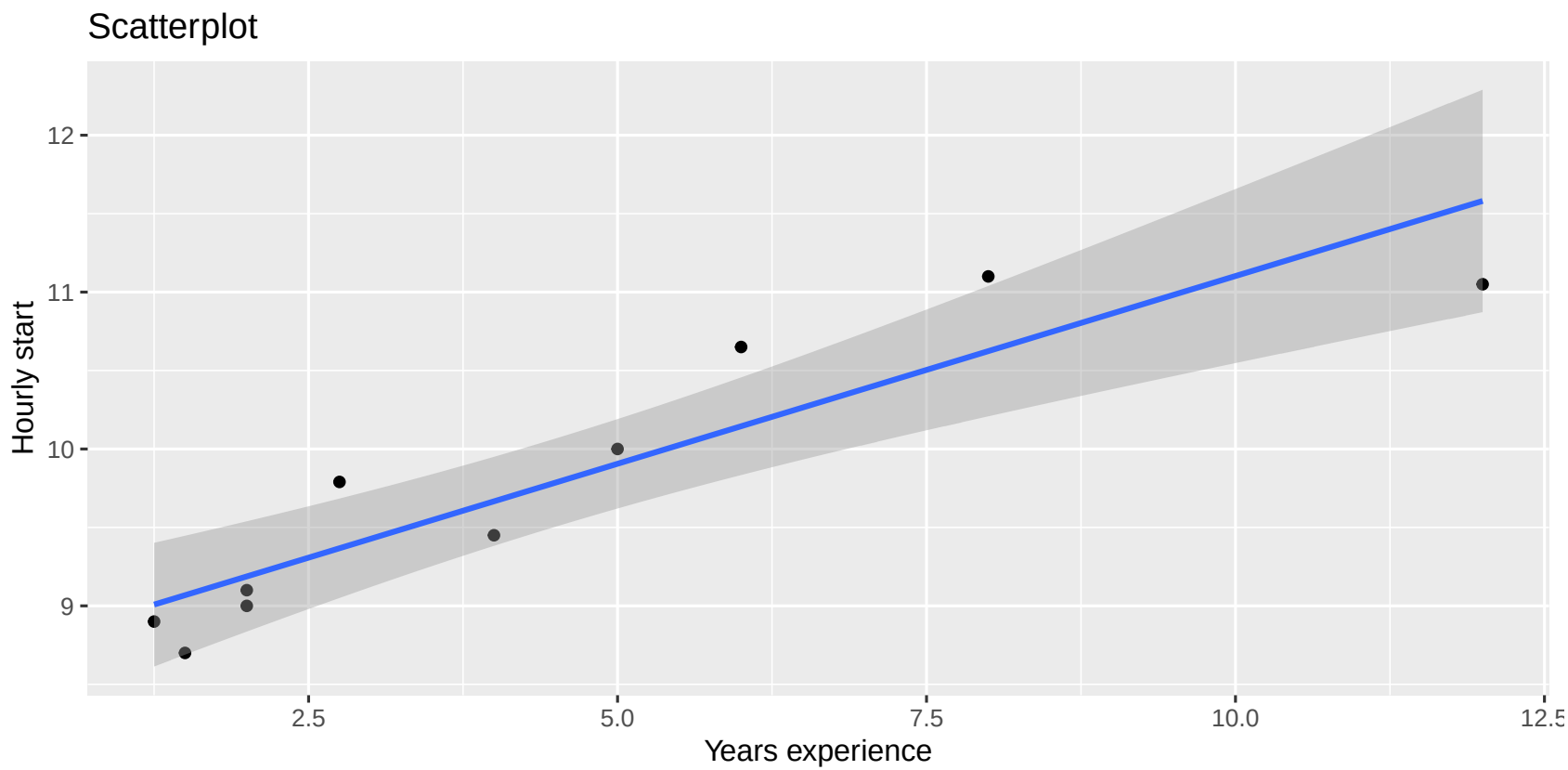
```
ggplot(wage, aes(x=yrs.exp, y=hrly.start)) +
  geom_point() +
  labs(title="Scatterplot", x="Years experience", y="Hourly start")
```



- In many instances the relationship can be summarized by fitting a straight line through the plotted points
- There is a strong relationship if the plotted points are positioned close to the line, and a weak relationship if the points are widely scattered about the line

```
ggplot(wage, aes(x=yrs.exp, y=hrly.start)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title="Scatterplot", x="Years experience", y="Hourly start")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



- It is fairly difficult to determine the strength of relationship between two quantitative variables by visually examining a scatterplot
- The **correlation coefficient** measures the strength of the linear relationship between two quantitative variables. The correlation coefficient is denoted as r.
- Suppose we have data on two variables x and y collected from n individuals or objects with means and standard deviations of the variables given as $\bar{x}$ and s_x for the x-variable and $\bar{y}$ and s_y for the y-variable. The correlation r between x and y is computed as follows

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

- A form of r that is somewhat more direct in its calculation is given by

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Example: the correlation between hourly wage offered and years of experience is

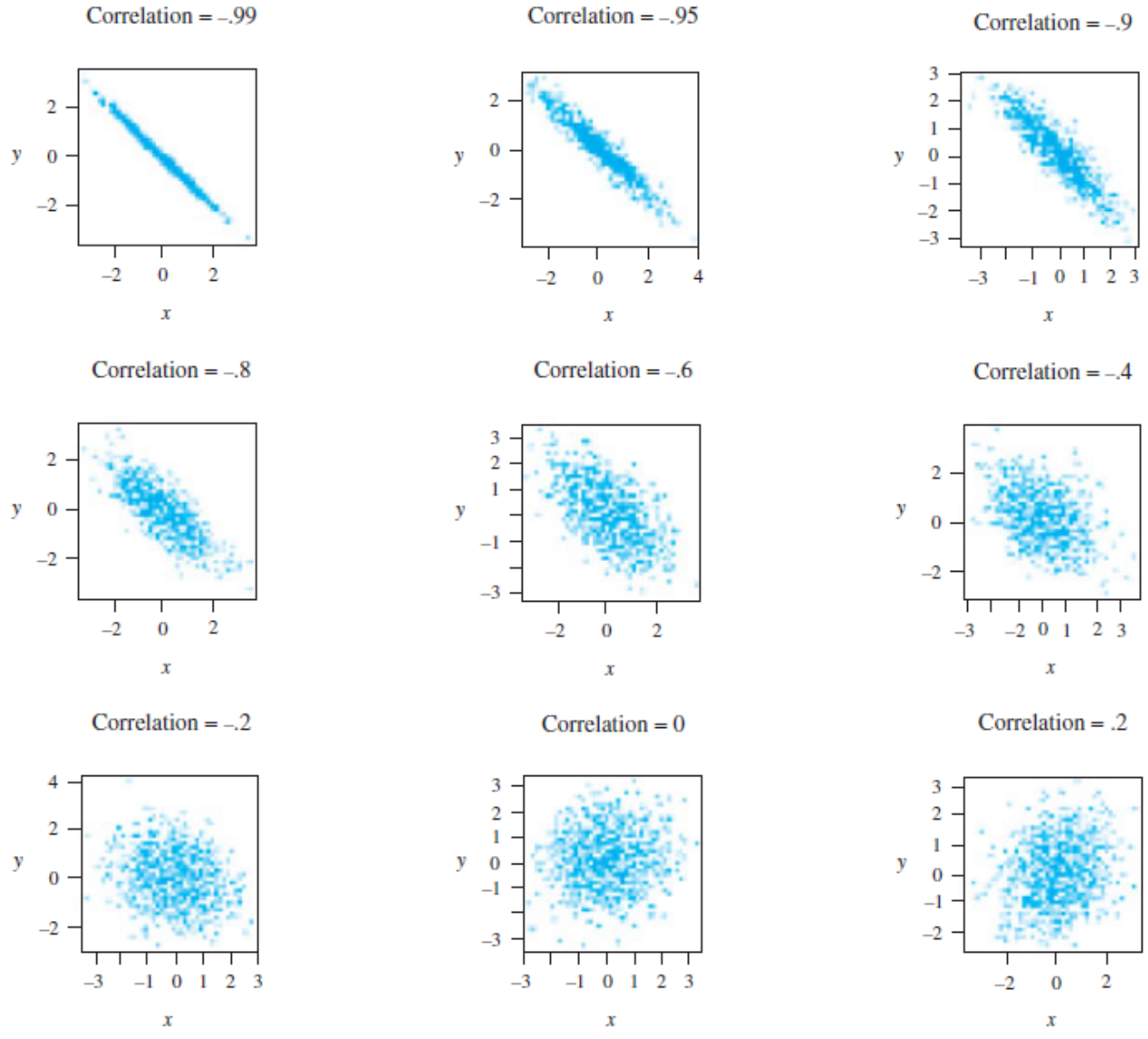
```
cor(wage$yrs.exp,wage$hrly.start)
```

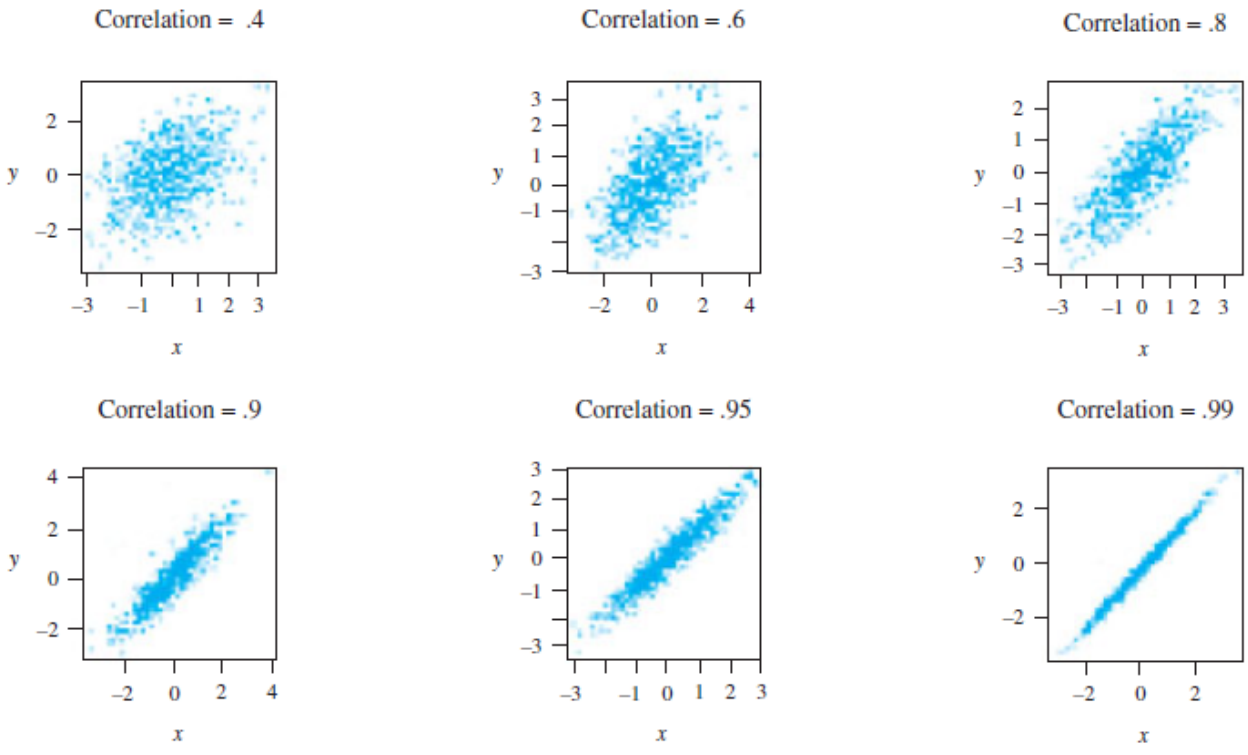
```
## [1] 0.9147556
```

- A correlation **DOES NOT** imply causation!
- Generally, the correlation coefficient, r, is a positive number if y tends to increase as x increases; r is negative if y tends to decrease as x increases; and r is nearly zero if there is either no relation between changes in x and changes in y or there is a nonlinear relation between x and y such that the patterns of increase and decrease in y (as x increases) cancel each other

- Some properties of r include the following:

1. A positive value for r indicates a positive association between the two variables, and a negative value for r indicates a negative association between the two variables
2. The value of r is a number between -1 and $+1$. When the value of r is very close to 1, the points in the scatterplot will lie close to a straight line
3. The value of r does not change if we alter the units of x or y. Correlation is a *unit-free* measure of association
4. Correlation measures the degree of straight line relationship between two variables. The correlation coefficient *does not* describe the closeness of the points (x, y) to a curved relationship


Correlation = –.99
Correlation = –.95
Correlation = –.9
Correlation = –.8
Correlation = –.6
Correlation = –.4
Correlation = –.2
Correlation = 0
Correlation = .2
x
y


Correlation = .4
Correlation = .6
Correlation = .8
Correlation = .9
Correlation = .95
Correlation = .99
y
x

Appendix: ggplot2 functions

Geom functions

- Whereas the `ggplot()` function specifies the data source and variables to be plotted, the *geom functions* specify how these variables are to be visually represented (using points, bars, lines, and shaded regions). Currently, 37 geoms are available. See Table on next slide

Function	Adds	Options
`geom_bar()`	Bar chart	color, fill, alpha
`geom_boxplot()`	Box plot	color, fill, alpha, notch, width
`geom_density()`	Density	plot color, fill, alpha, linetype
`geom_histogram()`	Histogram	color, fill, alpha, linetype, binwidth
`geom_hline()`	Horizontal lines	color, alpha, linetype, size
`geom_jitter()`	Jittered points	color, size, alpha, shape
`geom_line()`	Line graph	colorvalpha, linetype, size
`geom_point()`	Scatterplot	color, alpha, shape, size
`geom_rug()`	Rug plot	color, side
`geom_smooth()`	Fitted line	method, formula, color, fill, linetype, size
`geom_text()`	Text annotations	Many; see the help for this function
`geom_violin()`	Violin plot	color, fill, alpha, linetype
`geom_vline()`	Vertical lines	color, alpha, linetype, size

- Each geom function has a set of options that can be used to modify its representation. Common options are listed in the table on the next slide

Option	Specifies
`color`	Color of points, lines, and borders around filled regions.
`fill`	Color of filled areas such as bars and density regions.
`alpha`	Transparency of colors, ranging from 0 (fully transparent) to 1 (opaque).
`linetype`	Pattern for lines (1 = solid, 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash).
`size`	Point size and line width.
`shape`	Point shapes (same as `pch`, with 0 = open square, 1 = open circle, 2 = open triangle, and so on). See figure 3.4 for examples.
`position`	Position of plotted objects such as bars and points. For bars, `"dodge"` places grouped bar charts side by side, `"stacked"` vertically stacks grouped bar charts, and `"fill"`vertically stacks grouped bar charts and standardizes their heights to be equal. For points, `"jitter"` reduces point overlap.
`binwidth`	Bin width for histograms.
`notch`	Indicates whether box plots should be notched (`TRUE/FALSE`).
`sides`	Placement of rug plots on the graph (`"b"` = bottom, `"l"` = left, `"t"` = top, `"r"` = right, `"bl"` = both bottom and left, and so on).
`width`	Width of box plots.