Chapter 6: Visualizing Data using R and ggplot2 package

Overview: So far, we've discussed different ways of collecting data (methods of sampling) and we've seen how to read an external data set into R and how to access particular values from a data set whether it is a matrix or a data frame. We also learned some basic data wrangling techniques. We will now begin exploring how to visualize the data once it is read into R.

Visualization of data is a very important presentation method. It is a quick way to represent the data and to illustrate what the data is telling you. That being said, caution must be taken when choosing how to display the data visually as not all types of plots are appropriate for all types of data.

Motivating Example: Suppose you have several data sets that you want to visualize. These include the final letter grade distribution for a previous Stat 123 class, the annual lynx trappings in Canada, and the number of gears in a variety of manual and automatic cars. What is the best way to display these data sets?

In this chapter, we will cover several types of plots: bar charts, histograms, scatter plots, and line plots. We will not use R's built-in graphics. Instead, we will use the gglot2 package.

Plotting with ggplot2:

The function that we use to create plots is called *ggplot*(). The general form of the code required to create a graph using the *ggplot*() function is:

Question: What is the aesthetic?

Answer: The aesthetic changes depending on what type of graph you are creating. For a scatterplot, the minimum aesthetic required is the variable you want plotted on the x-axis and the variable you want plotted on the y-axis. You can also include colour in your aesthetic (which will be demonstrated in the examples below).

Question: How do we specify which type of graph we want to create (i.e. what do we type after the + sign?).

Answer: Again, it depends on what type of graph you want:

Bar Graphs: A **Bar graph** works well with categorical variables.

A bar chart can be used:

For example the final letter grade distribution of a former Stat 123 class could be effectively displayed using a bar chart. Suppose you have the following information:

Grades <fctr>	Number <dbl>
A	15
B	18
C	8
D	5
F	2

We sometimes refer this as a frequency table.

Bar Charts using ggplot2:

Start by creating a data frame with at least these two variables. One which contains the names of the categories of your categorical variable and another which contains the number associated with each category.

Next, use the function ggplot() function using a word geom_bar as the type of plot.

We can change the orientation of the bar graph:

We can add title, labels colour, etc. (see R scripts)

Producing bar graphs from different raw data:

Visualizing numerical variables using Histograms

Motivating Example: Let's look the penguin data set in the palmerpeguins package.

A **histogram** is an effective visualization tool when

you have a numerical variable that takes on many different values.

A histogram groups values together and then displays how many individuals in the sample belong to each group of values.

For example, the distribution of the following variables could be displayed using a histogram:

- **weight (mass) of penguins**
- **speed of vehicles crossing an intersection when traffic light turns yellow**
- **SAT scores**

Suppose you were trying to determine the distribution of SAT scores. If you have a sample of 20 people with the following scores:

1002	1250	1165	1204	1530
980	1230	1400	1394	1175
1120	1050	1110	986	1028
1240	1320	1060	1008	1526

How to make a Histogram using ggplot():

To produce a histogram, the word geom_histogram can be used in ggplot().

Let's use the morley data set with the basic command:

```
morley_hist <- ggplot(morley, aes(x = Speed)) + geom_histogram()
```

data from

Notes:

1. Sometimes you may want to indicate where the mean or median is in your data. You can do this by adding:

```
geom_vline(xintercept = , linetype = "dashed", size = 1.0)       for example

ggplot(morley, aes(x = Speed)) + geom_histogram() +
     geom_vline(xintercept = mean(morley$Speed), linetype = "dashed", size = 1)
```

2. You can specify border and fill colours:

```
ggplot(morley, aes(x = Speed)) + geom_histogram(color = "red", fill = "green") +

    geom_vline(xintercept = mean(morley$Speed), linetype = "dashed", size = 1.0)
```

3. You can add titles and labels exactly the same way as the bar graphs.

4. You can adjust interval length of each bar (bin width) by specifying binwidth in aes():

```
    aes(binwidth = ... )
```

   Set the bin width to help tell your story.

5. You can convert frequencies to relative frequencies by using the following in aes():

```
    aes(y = count/sum(count))  or

    aes(y = 100*count/sum(count)) # if you want percentages
```

6. You can specify your x-axis range by adding xlim(,) parameter

```
    + xlim(lower, upper)
```

   By limiting the range, some information is lost.

What can we tell from a Histogram?

Definition: The term distribution is used commonly in statistics. A **distribution** of a variable

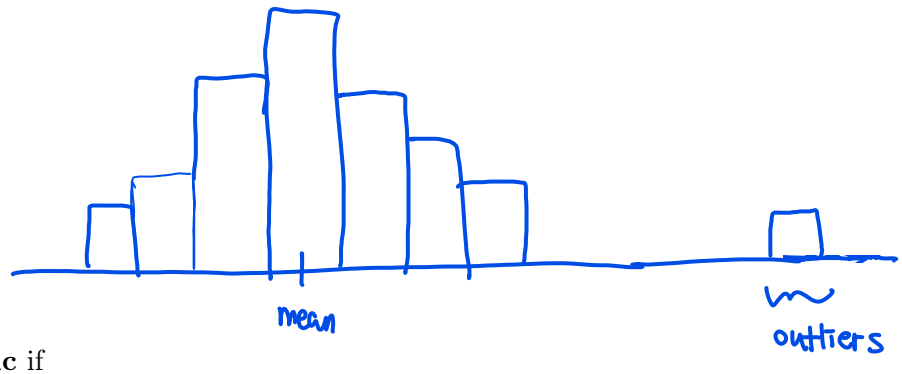
how data are spread over the range of values.

Thus, when we plot data. We are visualizing the distribution of the variable we are plotting.

Definitions: When looking at the distribution of a variable, we look to see if any of the following apply:

- An **outlier** in any graph of data is an observation the falls outside the overall pattern of the graph. This is a value in the data set that lies far away ft the rest of the data.

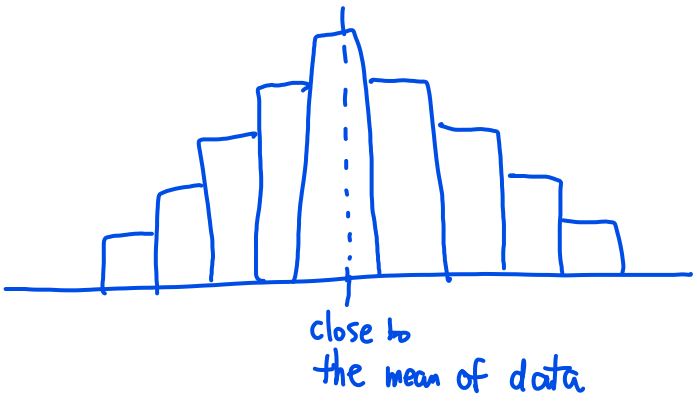
For example:



- A distribution is **symmetric** if

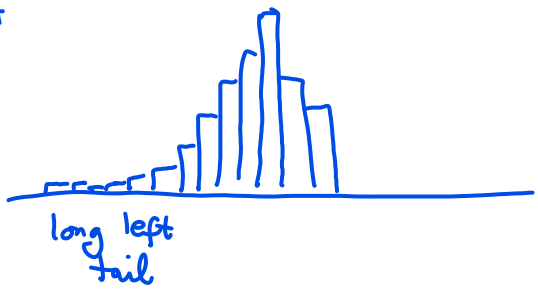
the left and the right side of the histogram are approximately mirror images of each other.

For example:



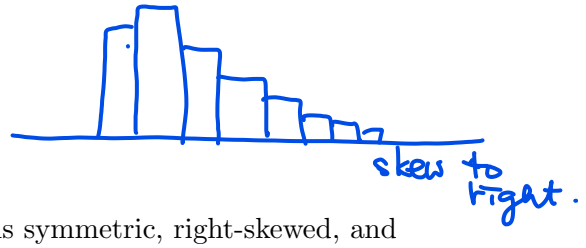
- A distribution is **skewed to the left** if

  left side of histogram extends further out to than the right side.



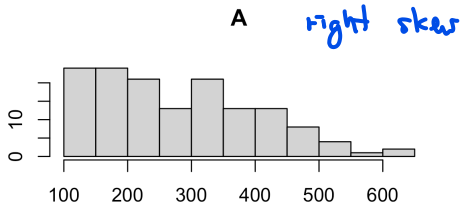
- A distribution is **skewed to the right** if

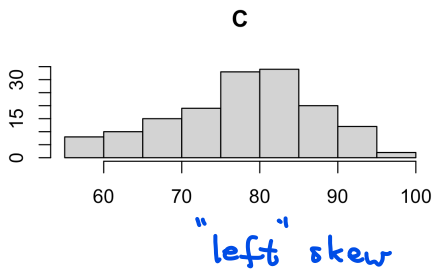
  right side of histogram has long tail.

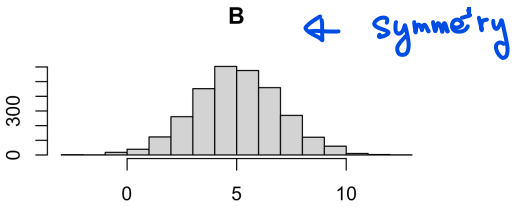


Practice Question:

Consider the following histograms and determine which one is symmetric, right-skewed, and left-skewed.



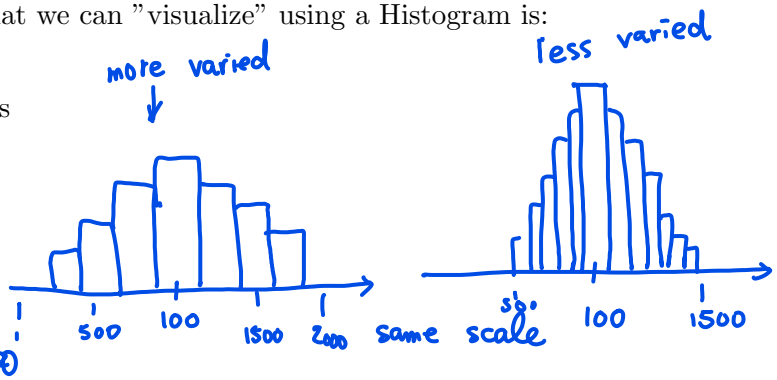




One last characteristic of a distribution that we can "visualize" using a Histogram is:

- The **variability** of a distribution describes

  the spread of the values the variable is on.



Note: We will learn a precise way to compute variability in later.

The next type of plot that we will look at in this chapter is called a **line graph**.

Line Graphs: A **line graph** is often used for numerical variable, collected over time.

When x variable is time, we call this times series.

For example, consider the built in lynx data set. This contains the annual lynx trappings in Canada from 1821 to 1934.

If you use the class() function in R, you will see that this data set is considered to be time series (ts) data.

Line Graphs with ggplot: Time series data is very easy to plot in R. The word is use is geom_line.You will need to specify which variable goes to the x-axis and which goes to the y-axis. For example, the following code creates a line graph for the lynx time series:

```
Year <- 1821:1934
lynx.df <- as.data.frame(cbind(Year, lynx))
head(lynx.df)

ggplot(lynx.df, aes(x = Year, y = lynx)) + geom_line()
```

Question: What can we notice from plotting a line graph?

Answer: We can notice something called **trend** and **seasonal variation**.

Definition:

- A **trend** in a time series is long-term trend (does it go up or down)

- A **seasonal variation** in a time series is a pattern repeat itself over a period of time.