Introduction to *R* — Part 1
Biostatistics (STAT:3510, Bognar)

Accessing *R*

R is available on all UI computers. It is also available as a free download from

`http://www.r-project.org`

R works on Windows, Mac, Linux, Unix, BSD, etc.

On a Mac or Linux, you run *R* by typing `R` (then `Enter`) in a terminal. To open the terminal on a Mac, go to

Applications → Utilities → Terminal

On Windows, open the application called *R* (this opens an application window).

When you open *R*, you will see the command prompt, i.e. `>`. To quit *R*, just type `q()` and hit `Enter`.

Toy Dataset Analysis

Enter Data Into *R*

The toy dataset describes how long (in minutes) it took 7 randomly selected adults to assemble a toy (see below).

To load the data into an object called `toy`, we use vector notation, i.e. `c(my data separated by commas)`. The `c` character stands for *combine*. The assignment operator is a left arrow `<-` (i.e. a less than sign followed by a dash). The full command is

```
toy <- c(5.3, 6.4, 6.7, 6.9, 7.2, 7.2, 7.9)
```

You can see the data inside of `toy` by typing its name.

```
toy

[1] 5.3 6.4 6.7 6.9 7.2 7.2 7.9
```

Sweet tip — you can recall and edit previous commands by using the 'up arrow' on your keyboard

Summary Statistics

Lets have *R* compute some basic summary statistics. We know how to do these things by hand; *R* can do the exact same computations in a flash. The sample mean $\bar{x}$ is found by

```
mean(toy)

[1] 6.8
```

the sample standard deviation s is computed using

```
sd(toy)

[1] 0.8124038
```

and the sample variance s^2 can be found by

```
var(toy)

[1] 0.66
```

We can find the quantities in the 5-number summary as follows.

```
min(toy)

[1] 5.3

max(toy)

[1] 7.9
```

The `quantile` function computes quantiles. For the 25th, 50th, and 75th percentiles, we have

```
quantile(toy, 0.25)

 25%
6.55

quantile(toy, 0.5)

50%
6.9

quantile(toy, 0.75)

75%
7.2
```

The super fast way to get Min, Q_1, Q_2, Q_3, Max, and the sample mean $\bar{x}$ is to use the `summary` function (this function *summarizes* our dataset):

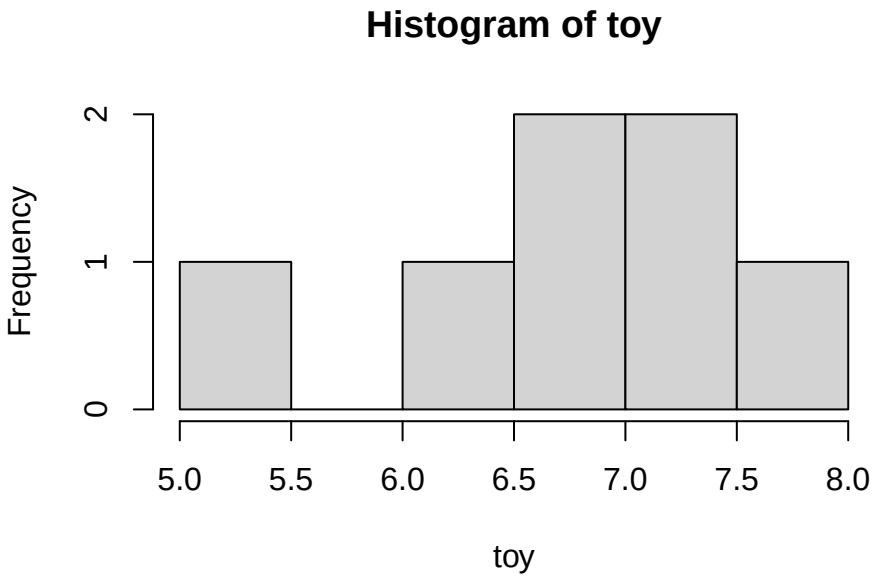
```
summary(toy)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.30    6.55    6.90    6.80    7.20    7.90
```

Statistical Graphics

R is capable of making *publication quality* graphics (much nicer than Excel). For example, to make a histogram of the data, type

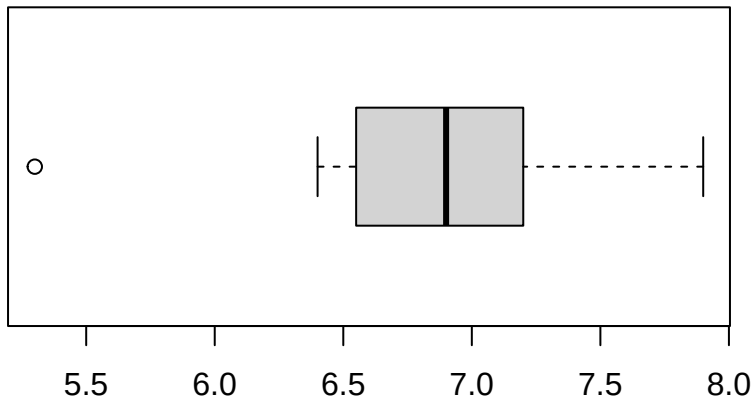
```
hist(toy)
```



Graphics output will appear in a separate window. You should be able to copy and paste the graphics output into Word (or similar).

To make a boxplot of the dataset, type

```
boxplot(toy, horizontal=TRUE)
```



The `horizontal=TRUE` argument makes a horizontal boxplot; the default is a vertical boxplot.

A stem-and-leaf plot can be created using the `stem` command (the stem-and-leaf plot appears in the console window)

```
stem(toy)


  The decimal point is at the |

  5 | 3
  6 | 479
  7 | 229
```

We would like more stems than this — you can increase the number of stems by using the `scale` argument.

```
stem(toy, scale=2)


  The decimal point is at the |

  5 | 3
  5 |
  6 | 4
  6 | 79
  7 | 22
  7 | 9
```

The `scale=2` argument had the effect of splitting each stem into two parts. For example, the 6 stem was split into a low-6 stem $(6.0, \ldots, 6.4)$ and a high-6 stem $(6.5, \ldots, 6.9)$.

You have now used the most prominent software in the Statistical community! *R* is extremely flexible, powerful, and easy to use. It also has publication quality graphics. This intro barely scratched the surface.

To quit *R*, just type `q()` at the command line.