Statistics 411/511
Lab 1

Lab Instructions: The structure of the labs allows you to work along with the TA in lab, or work at your own pace, either in the lab or not. If you want to work along with the TA in lab, please sit near the front. If you prefer to go to lab but work at your own pace, please sit near the back, and wait for the appropriate time to ask any questions.

Lab 1: Introduction to R

Objectives for this Lab

- Introduce RStudio
- Install and load R packages.
- Produce some simple graphs.
- Calculate sample statistics (mean, median, standard deviation, etc.).
- Perform a t-test.

1. Download the file Lab1.r from Canvas. This file contains all the R commands used in this lab. Note the directory to which you download it.

2. Start up RStudio. If you're in Bexell 324, click the window icon at the bottom left of the screen. Type `RStudio` in the "Search programs and files box," then click the RStudio icon. If you need to install the software on your own computer, see the Installing R and RStudio page on the course website.

3. The default RStudio desktop layout shows three panes. Clockwise from left, they are

   - Console
     This is where RStudio communicates with R. You should see some startup information and a "prompt" `>`.
   - History/Environment
     - *History* keeps track of all the R commands issued in the command window.
     - *Environment* lists all the variables in R's workspace.
   - Files/Plots/Packages/Help/Viewer
     - *Files* shows the contents of the current directory.
     - *Plots* will display the plots and graphs as you generate them.
     - *Packages* lists the R packages installed and allows you to install new packages.
     - *Viewer* allows you to view local web content. We'll probably leave this one alone.

4. Open Lab1.r. In RStudio, select File>Open File... Navigate to the directory where you saved the file and click Open. You should now have a source editor pane open at the top left, displaying Lab1.r.

5. Load the Sleuth3 package so you have access to the data sets.

   (a) You'll need to install the package, unless it's already installed on the computer you're using.

      i. From the Packages pane, click the Install button to open the Install Packages dialog box.

ii. Type 'Sleuth3" in the Packages line (case sensitive, "sleuth3" won't work). Click "Install" at the bottom of the dialog box.

(b) Load the Sleuth3 package into the R session's library. You'll have to do this every time you start up RStudio if you want to use the data in the *Sleuth's* case studies and exercises.

```
> library(Sleuth3) # Load the Sleuth3 package.
```

This is the first command in Lab1.r. You can either type the command after the prompt (">") in the Console pane, or place your cursor anywhere in the first line of Lab1.r and click the Run button at the top right of the source editor pane.

All the text in a line after a "#" symbol is called a "comment" and ignored by R.

Note: You'll have to "load" the library again next time you start up RStudio, but once it's "installed" you don't have to do that again. Installing a package downloads it to your computer. Loading it in R tells the current R session about the functions and data it contains.

(c) To get information about the Sleuth3 package, use the `help()` function.

```
> help(package=Sleuth3)
```

We will work with the salary data of case study 1.1.2. This is `case0102` in the Sleuth3 package. In the help window, click on the case0102 link for information and references about this data set.

(d) You can look at the data by typing the name of the data frame:

```
> case0102
```

The Console pane lists the contents of the "data frame," R's term for how it represents a table of data. The data frame contains two columns called "Salary" and "Sex." The 93 rows are numbered sequentially.

6. Draw some histograms of the salary data.

(a) Produce a histogram of all salaries.

```
> hist(case0102$Salary)
```

The command `hist()` draws a histogram of the data given in its "argument." RStudio displays plots in the lower right-hand pane.

The notation `case0102$Salary` refers to the column "Salary" in the data frame "case0102."

(b) It can be tedious to type the data frame name each time you want to refer to a column when issuing a single R command. The `with()` function provides a shortcut:

```
> with(case0102,hist(Salary))
```

You should get the same histogram as before, except the title and horizontal axis label omits the data frame name.

(c) To make separate histograms for males and females as in Display 1.4, we'll need to extract the observations from males, and draw a histogram, then repeat for females. Here, since we need to refer to both Salary and Sex, `with()` really does save some typing.

```
> with(case0102,hist(Salary[Sex=="Male"]))
> with(case0102,hist(Salary[Sex=="Female"]))
```

The syntax `Salary[Sex=="Male"]` extracts the salaries from the rows where Sex is Male. The double "==" is not a typo. The above code produces two graphs. You can use the back arrow in the Plots pane to see the previous graph.

These histograms don't look like the ones in Display 1.4 because the breakpoints are different. You can specify the breakpoints with an optional argument to `hist()`.

```
with(case0102,hist(Salary[Sex=="Female"],
                   breaks=c(3800,4200,4600,5000,5400,5800,6200,6600)))
```

The `c()` command *concatenates* its arguments into a single unit. Setting `breaks=c(3800,...)` tells R the specific breakpoints to use. You can make several other custom adjustments to the histogram. For details, check the help documentation.

```
> help(hist)
```

7. Histograms are useful for seeing the shape of the distribution of data, but they are highly dependent on the breakpoints. *Boxplots* are less subjective. We will use boxplots as an opportunity to introduce the `ggplot2`package, which contains a vast suite of sophisticated graphing tools.

   (a) Install and load the ggplot2 package. Use the same series of steps as in item 5, namely (1) install the package using the "Install" button on the Packages pane, and (2) load the package in the Console pane using the `library()` command.

   (b) Use `qplot()` ("quickplot") to produce side-by-side boxplots of the salary data as in Display 1.12.

   ```
   > qplot(Sex,Salary,data=case0102,geom="boxplot")
   ```

   The first two arguments to `qplot()` are the grouping variable and the quantitative variable, respectively. The next two (`data=`and `geom=`give the data frame and the "geometry" of the plot. Note that the order of the first two arguments matters, but that the second two can come in either order. For example, the following produces the exact same plot

   ```
   > qplot(Sex,Salary,geom="boxplot",data=case0102)
   ```

   whereas the following produces a funny-looking plot and a warning message.

   ```
   > qplot(Salary,Sex,data=case0102,geom="boxplot")
   ```

   (c) You will usually have two or more choices for doing the same thing in R. For example, you can also use `ggplot()` to make a boxplot. The `ggplot()` function has a lot more flexibility than `qplot()`, but it isn't as...quick. Here's a way to make the side-by-side boxplots using `ggplot()`.

   ```
   > ggplot(case0102,aes(x=Sex,y=Salary)) + geom_boxplot()
   ```

   The "gg" in `ggplot()`stands for "grammar of graphics." It's a novel way of making sophisticated, informative graphics. For ST 411/511, `qplot()` will generally be sufficient, but for the moment, check out what this small addition to the command does:

   ```
   > ggplot(case0102,aes(x=Sex,y=Salary)) + geom_boxplot(aes(fill=Sex))
   ```

   We can easily add a title.

   ```
   > ggplot(case0102,aes(x=Sex,y=Salary)) + geom_boxplot(aes(fill=Sex)) +
             ggtitle("Starting Salaries")
   ```

(d) Finally, plain R has a `boxplot()`function that `ggplot()` has largely replaced. Here's how to make side-by-side boxplots with `boxplot()`:

```
> with(case0102,boxplot(Salary~Sex))
```

Remember `with()` from item 6(b)? The variable on the left-hand side of the $\sim$ is read as a dependent ("response") variable, whereas the variable on the right-hand side is an independent variable. In this case, the independent variable is a categorical variable which serves to group the response into separate categories.

8. You may want to include an R graphic in a Word or LaTeX document. To do this, click the Export menu in the Plots pane. You can save as a PDF, copy to the clipboard, or save as an image. The easiest way to insert into Word is to copy to the clipboard, then paste into an open Word document.

9. A stem-and-leaf diagram shows both the distribution of the data and the actual numbers. R's `stem()` function produces these. It won't do back-to-back diagrams as in Display 1.10, so we'll have to do separate commands. Here's the command for the male salaries:

   ```
   > with(case0102,(stem(Salary[Sex=="Male"])))
   ```

   Note that the diagram appears in the Console pane, not the Plots pane.

10. To get summary statistics about a set of data, use `summary()`.

    ```
    > with(case0102,summary(Salary[Sex=="Female"]))
    ```

    Many R commands behave differently depending on what type of argument you give them. `summary()` is one of those. The arguments `Salary[Sex=="Female"]` is *numeric*. On the other hand, `Sex` is a "factor" variable, which is R's name for a categorical variable. You can verify this using `is.factor()`.

    ```
    > with(case0102,is.factor(Sex))
    ```

    If you ask for a summary of a factor variable, you'll get a display of the unique values ("levels") of the factor and the counts in each group.

    ```
    > with(case0102,summary(Sex))
    ```

11. The summary statistics of a numeric variable above includes the minimum, maximum, median, mean, and the 1st and 3rd quartiles, but *not* the sample standard deviation. R will calculate this for you:

    ```
    > with(case0102,(sd(Salary[Sex=="Female"])))
    ```

12. Perform the one-sided two-sample t-test reported in the Summary of Statistical Findings at the bottom of page 4 of the textbook. We will spend more time on the t-test in Chapter 2, but this should not be the first t-test you've ever done.

    ```
    > t.test(Salary~Sex,alternative="less",data=case0102,var.equal=TRUE)
    ```

There's that same "formula" that we used in item 7(d). Setting `alternative="less"` tells R that the alternative hypothesis is that the female salaries are less than the male salaries. The ordering is usually alphabetic, but you can check the output of `summary(Sex)` in 10 to see that "Female" is indeed listed first. Find the p-value and the confidence interval.

13. Close RStudio and quit R.

    (a) Save any script files (extension .R or .r) that you've edited. These files contain the commands needed to reproduce your work. The File menu allows you to save a file.

    (b) Select Quit RStudio... from the File menu to close RStudio. This action automatically quits R as well.