2.2: Why do we use R software?

Introduction

Why do we use R Software? Or put another way: Dr D, Why are you *making* me use **R**?

Truth? You can probably use just about any acceptable statistical application to get the work done and achieve the learning objectives we have for beginning biostatistics. However, we will use the **R statistical language** as our primary statistical software in this course. Part of the justification is that all statistical software applications come with a **learning curve**, so you'd start at zero regardless of which application I used for the course. In selecting software for statistics I have several criteria. The software should be:

- free software
- open source
- widely accessible and compatible with ~~all~~ most personal computers
- well-respected and widely used by professionals
- well-supported for the purposes of data analysis and data processing
- really good for making graphics, from the basics to advanced
- capable to handle diverse kinds of statistical tests
- if not exactly easy, the software should have a reasonable learning curve

R meets all of these criteria. **R history** began back in 1993 and has always been available as free software under the terms of the Free Software Foundation's GNU General Public License in source code form. R compiles and runs on a wide variety of UNIX platforms and similar systems, including GNU/LINUX, FreeBSD, and various Linux distros like the popular Ubuntu®, in addition to their more famous Microsoft Windows® and Apple macOS® distributions. To facilitate access to the software, numerous **mirror sites** are available from sites around the world, with cloud.r-project.org supported by RStudio perhaps the most widely used. From December 2021 to December 2022, more than 6 million downloads of base R were made from the RStudio **CRAN mirror** site (CRAN stands for Comprehensive R Archive Network; a mirror refers to a website or server that holds a copy of files from another website/server to make the files available from more than one place).

Note:

One hundred and four mirror sites as of March 2023, 105 different locations (including R CRAN at r-project.org), from which to download R and related packages. Thus, it's not a simple task to count total downloads of R. RStudio has given access to their `changelog` file, which allow one to track numbers of downloads for any package from their mirror site — https://cloud.r-project.org/. Here's the code and recent counts for downloads of R itself (about 400K over a four week period).

```
install.packages("cranlogs")
library(cranlogs)
# How many downloads of base R first four weeks of Fall semester?
 out <- cran_downloads("R", from = "2023-08-21", to = "2023-09-08")
sum(out$count)
```

R output

```
[1] 398524
```

R is straightforward to use once you learn how to work with the language, but has a steep learning curve; after all, it's a programming language. The GUI **R Commander** helps in this process, and eventually, your use of code will become second nature. After the initial growing pains are behind you, **RStudio** likely will be a better solution over R Commander. However, while we need statistical software to do statistics, students in my BI311 course must keep in mind that learning objectives for most biostatistics course are about the concepts and interpretation of statistics, not just use of the software. In other words, learning how to use R is not the focus of BI311 nor will you likely achieve R programming competency by the end of the semester. I certainly

encourage students to strive for competency and I give frequent bonus opportunities to demonstrate coding skills during the semester.

Thus you might ask if the purpose of the course isn't to learn R, why work with R instead of a more familiar app or software, e.g., **Microsoft Excel**® (hereafter simply referred to as Excel), or **Google Sheets**, or even my favorite open-source office alternative, **LibreOffice Calc**? Or, perhaps even just one of the many online calculators, if the course learning objective is to "just" learn about statistics?

First, I believe that real data derived from real biology or biomedical problems are essential elements to a first course in biostatistics. That's not a particularly unique perspective, although I don't have survey results of other statistics instructors to back up the claim. Real problems involve observations on multiple subjects, many variables — large data sets; this alone precludes use of hand calculations and calculators. As a corollary, we will not spend a great deal of time learning the in's and out's of the algorithms that form particular statistical tests. Now, do understand that there is a tremendous benefit to understanding statistics by working through the equations, by looking at the algorithms, and there's no escaping the need for understanding that probability provides the foundation of **statistics inference** (Chapter 8). Thus, for most of us, the statistical software available to us provides an appropriate framework for applying correct statistical tests to our projects. Therefore, the decision is about which statistical package we should use.

Second, R is perhaps *the* choice in academia for statistical software. A PUBMED search found more than 1500 citations of R. Visit Robert A. Muenchen's web page (The popularity of data analysis software, r4stats.com) to see updated statistics on statistical software use. Those of you continuing on to graduate school or to professional schools will find that many of your statistically literate colleagues use R and not one of the commercial programs. While there are many excellent commercial packages (Table 2.2.1), and in some cases you can make spreadsheet programs do statistics (typically add-ins are required), all statistical software come with steep learning curves. Thus, part of my selling point to you is that learning to use R is at the cutting-edge in your field and, given that all of the software you could use can have have their challenges, it is best to work with something that will be around and is in wide use, without the burden of a financial investment.

Table 2.2.1: Comparison of Commercial Statistical Software Programs

Software	Student license?	Limited or full function version	macOS	Windows 11	Fee*	Academic license type
GraphPad Prism	Subscription, $142 per year	Full	Yes	Yes	$202	annual subscription
JMP	Yes, but with purchase of selected textbook	Limited	Yes	Yes	$100	monthly subscription
Minitab	Subscription, $54.99 per year	Full	Yes	Yes	$1610	annual subscription
IBM SPSS	Rental, $76 per year	Full	Yes	Yes	$260	annual rental
SigmaSTAT	No	NA	No	Yes	$299	perpetual
MySTAT	Yes, free	Limited	No	Yes		NA
SYSTAT	No	NA	No	Yes	$739	perpetual
Stata	Subscription, $94 per year	Full	Yes	Yes	$325	annual subscription

last updated November 2022

see Wikipedia for list of additional software

Third, what about online sites like **plot.ly** where, for free, you can plot and, in some cases, calculate statistics? What about the web application at Brightstat, which claims to provide an SPSS-like experience online (Stricker 2008)? While it is true that there are many wonderful websites that can perform many of the statistical tests we will use this semester, these sites are not suitable for more than occasional use.

How to get started with R

The R statistical language, accompanied by additional packages to extend its capabilities beyond basic math and statistical functions, provided a complete statistical environment. R is best viewed as a programming language for statistics (**data analysis**), and **data processing**. Power users of R learn how to write scripts that do t-tests, ANOVA, regression, etc. The scripts are just lines of code that R understands and it provides the user tremendous control over analysis and inference of data sets. Because of this flexibility and power, however, R can be intimidating at first. So, we'll start slowly with scripts, introducing just what we need to get started and build from there. We'll be addressing R issues in more depth over the next several weeks, but for the first week, our goal(s) should be to make sure each of you knows how to start/exit R, how to create and utilize a **working directory**, and how to use R as a calculator. You obtain your copy of R from the R Project for Statistical Computing, available at https://www.r-project.org. Instructions to install R are provided in Install R. A ten-part tutorial to get started using R is provided in Mike's Workbook for Biostatistics.

Note:

A working directory or working folder is something you create on your computer to contain the files and sub-directories of a project. It sets the default location for files you may need to have R read. For example, all of your work for a course (data files, script files, Markdown files), may be stored in a folder called BI311 on your Desktop. For example, on a macOS, the path to the working folder would be

```
/Users/username/Desktop/BI311
```

Why R Commander?

We utilize an R package that provides a menu-driven context to much of the typical statistics one needs to do biostatistics. The package is called R Commander (**Rcmdr**), which provides a graphic user interface or GUI. Rcmdr therefore significantly eases the learning curve for doing statistics with R. We use a package called R Commander, which provides **drop down menus** for most of the typical kinds of analyses. Rcmdr is in use in many courses across the world (more than 20K downloads in September 2023), and among the other GUI available for R, Rcmdr is among the best supported GUI available for R. R Commander function is extended by plug-ins; as of August 2023, there were 36 plugins that extend Rcmdr's capabilities. Instructions to install R Commander are provided in **Install R Commander**.

Note:

Other options to improve use of R include use of RStudio®, which is an integrated development environment, or IDE. RStudio is really nice to use, and happily, you can run R Commander within RStudio. I am also increasingly using shiny apps within the course to help with concept presentation; in the future, I plan to provide a complete shiny app which would allow BI311 students to work interactively with the statistics presented in this text, something like the radiant-rstats project. However, for use in our course, R Commander provides a familiar look as students develop knowledge in the course: simply point and click to access the statistical functions.

Wait! Why don't we use Microsoft Excel? My instructor in {insert course here} used Excel…

A very reasonable question for you to ask — why don't we use Excel or Google Sheets for statistics? Moreover, it is highly likely that you have gained at least some introduction to descriptive statistics and graphing with spreadsheets in former courses — shouldn't we learn statistics within a framework you are already familiar?

After all, "Can't Microsoft Excel do statistics?" Mostly the answer is, no, not really (Fig. 2.2.1).

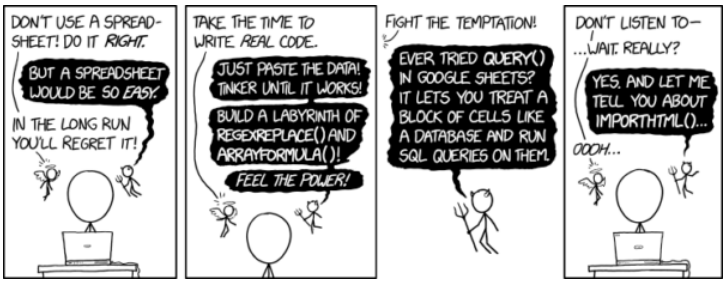

Figure 2.2.1: "Spreadsheets," **xkcd.com** no. 2180

MS Excel, Google Sheets, Apple Numbers, and for that matter, Calc, the spreadsheet application in my favorite office app LibreOffice (LibreOffice is a free, open-source alternative to Microsoft Office), can be used to calculate many descriptive statistics. With some effort, these applications can be extended by use of either Analysis ToolPak or Solver Add-ins to do more complicated statistics like regression and analysis of variance, and curve fitting.

However, use of MS Excel for statistical analysis involves learning a number of commands, syntax, and developing work flows that are neither intuitive nor standard. Some publishers have provided add-ins that are reportedly designed to simplify this process (e.g., MegaStat® by McGraw-Hill or XLStat). None of these options are free and none are in use in any major way by scientists (see The popularity of data analysis software). The free add-ins of Analysis ToolPak and Solver may work for you if you own a Windows PC, but only Solver is included for the Mac versions of Excel. Mac users may download and install StatPlus:MacLE, which is a limited, but free alternative to the Analysis ToolPak add-in; for a complete package a Pro version is available (licenses started at $89, web site: www.analystsoft.com/en/products/statplusmacle/).

An additional caution: you should be aware that there have been reports over the years that algorithms selected by Microsoft for Excel have not always been to industry standards (e.g., McCullogh and Wilson 2005). In short, the fit of Excel and other spreadsheet apps for use in statistics is not a simple one. To do the kinds of statistics we will use routinely in class, Excel would need to be modified with add-ins, and the add-ins would be the result of programming by someone. And you would still need to learn how to write the code.

What about graphics? You may like Microsoft Excel's ability to do graphics. Indeed, Excel, Google Sheets, and LibreOffice Calc can be used to generate many typical kinds of statistical plots. But again, in comparison to R, spreadsheet app graphics are limited and require a deal of effort to generate acceptable plots. I think you'll be surprised at how straight-forward R is. Here's an example, first rendered in Microsoft Excel, then in **base R**. And importantly, the kinds of plots Excel does well at are not necessarily the plots suitable for research publication. For example, Excel allows you to make bar charts easily, but cannot do box plots. Box plots are preferred over bar (column) charts for ratio scale data.

Note:

base R refers to the core R programming language along with many functions and graphics routines. We extend capabilities of base R by adding packages, like R Commander. Definition text

Statistics comparisons between R and MS Excel

About that learning curve. Let's compare R and MS Excel for basic functions common in data analysis. Similar conclusions hold for comparisons to Google Sheets and LibreOffice Calcs. Table 2.2.2 lists the observations we can use to conduct comparisons of the applications.

Table 2.2.2. A simple data set of one variable, A, with 24 observations

varA
12
14
20
25
28

varA
29
32
34
35
39
47
47
50
53
54
71
79
87
89
96
105
122
130
132

One of the first steps in data analysis is to produce what are called descriptive statistics. Common **descriptive statistics** are the **mean** and the **sample standard deviation**. Let's compare Excel and R for retrieving these two statistics.

With Excel, to calculate the arithmetic mean of 24 numbers, enter the values into a single column of 24 rows, then enter " `=average(A2:A25)` ", without the quotes, into a new cell of the spreadsheet. " `A2:A25` " refers to where data would be contained in column `A` rows 2 through 25. Typically the first row in a worksheet would contain the name of the variable, e.g., " `A` ." Depending on the significant figures set, the estimate returned by Excel for the mean of `A` is `59.58333333` .

Similarly, to obtain the standard deviation, type `=stdev(A2:A25)` , into a new cell of the spreadsheet. Again, depending on the significant figures set, Excel returns a value of `37.05215674` for the standard deviation of `A` .

In contrast, to obtain the mean and standard deviation for a variable in an R data set, all you would type at the **R prompt** (`>`), or in the **script window**

Note:

Always run your code as a script. Entering code at the R prompt means you are working at the command-line interface, and you work one line at a time. This is not an efficient way to interact with R. Instead, I recommend you always create and work from a script document. For beginners, that's why I recommend R Commander, which includes a script window. Simply type your code in the script window, highlight the code you wish to run, and run by clicking submit button (or Ctrl+R Win11 or Cmd+Enter macOS). When you are ready to move on from R Commander, RStudio is the IDE of choice.

and then submit, is:

```
A <- c(12, 14, 20, 25, 28, 29, 32, 34, 35, 39, 47, 47, 50, 53, 54, 71, 79, 87, 89, 9
```

where the “c” is a function to **combine** arguments into a vector and saved to the object `A` , followed at the new line by

```
mean(A)
```

Hit enter after entering the command) and R returns

```
[1] 59.58333
```

For the standard deviation, write the R base function `sd()`

```
sd(A)
```

Hit enter after entering the command and R returns

```
[1] 37.05216
```

It’s not much of a difference, but note that to get the mean (arithmetic average) I typed seven characters in R, but 16 characters in Excel; similarly, for the standard deviation I typed in 5 characters in R, but 13 characters in Excel. That’s a savings of 56% and 62%, respectively. Excel tries to help by using AutoComplete to anticipate what you want to enter, but AutoComplete doesn’t always work properly (e.g., see gene name errors generated by use of default Microsoft Excel settings, Ziemann et al 2016).

Note:

I use spreadsheets all of the time for **data entry** and **data management**. Make sure **AutoComplete** and **AutoCorrect** options are turned off and these problems are much less.

In conclusion, R is quicker for descriptive statistics.

Graphics comparison between R and MS Excel

MS Excel is often cited for its graphics capabilities (Camões 2016). We can make the familiar scatter plots, bar charts, and pie charts in Excel. These plots and more are easily obtained in R. I won’t elaborate here about graphics, since we talk at some length about graphics in Chapter 4. But here’s one example in R.

Let’s plot `B` vs `A` . We already provided the data for variable `A` , here’s the data for variable `B` .

```
17, 21, 21, 26, 27, 32, 28, 42, 40, 30, 71, 53, 56, 61, 55, 89, 82, 63, 116, 162, 11(
```

Don’t recall how to assign a set of numbers to an object, B, in R? See above and look again at how we assigned the numbers to object `A` .

To get a simple scatter plot (Fig. 2.2.2), I may write at the R prompt.

```
plot(A,B)
```

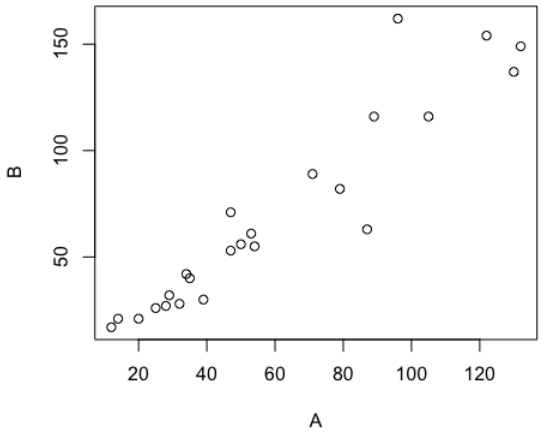

Figure 2.2.2: Basic scatter plot made in R, using `plot(A,B)` .

And here's the comparable default plot (Fig. 2.2.3) from Microsoft Excel, Office 365

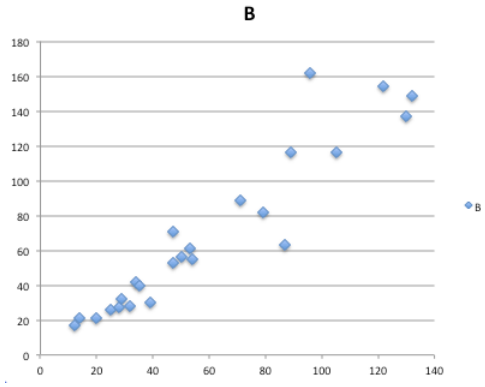

Figure 2.2.3: *Basic scatterplot made in Microsoft Excel.*

Now, both graphs need some work, and to be fair, these are just the defaults. With some effort, you can make an Excel graph look pretty good. But note — the defaults in Excel don't generate axis labels, while R default plot does. Excel adds a useless title and legend; both need to be removed. Excel also adds grid lines where typically one would not include these in a scientific plot.

So, lets count the steps to generate an acceptable scatter plot (Table 2.2.3). I've also added R Commander (`Rcmdr`) steps for comparisons (`Rcmdr` lets you use drop-down menus like Excel or Google Sheets or LibreOffice Calcs).

Table 2.2.3: Steps needed to make a simple scatterplot in R, R Commander, or Microsoft Excel.

Steps	R	Rcmdr	Excel 365
1	write the function	Select Graphs	Highlight columns
2		Select scatterplot	Select from Menu "Insert"
3		Select variables	Select scatterplot
4		Uncheck options	Select type of scatterplot
5			Delete legend
6			Remove grids
7			Insert X-axis label
8			Insert Y-axis label

Conclusion? R is quicker for routine statistical plots like a scatter plot. And I didn't even count the steps needed to change MS Excel's dreadful diamond icon points.

That's one step in R, four steps in `Rcmdr`, but eight steps for Microsoft Excel. LibreOffice Calc is a little better at four steps, but like MS Excel, you'd need to change several components to the graph (Fig. 2.2.4).

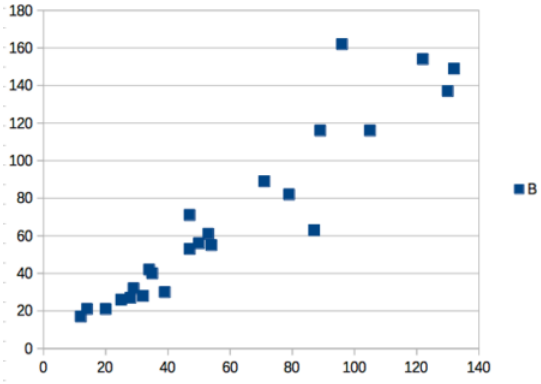

Figure 2.2.4: *Basic scatterplot made in LibreOffice Calc.*

Note:

In R vernacular, these data markers are referred to as **pch**, or **point characters**: `pch = 23` returns a blue diamond character; for a blue square like Figure 2.4, add to the `plot()` command as

```
plot(A,B, pch = 22)
```

So, you're telling me I don't need a spreadsheet application?

No, not at all. We use spreadsheets, and more generally, databases, to store data. Spreadsheets apps are designed to make data entry and data management approachable and efficient. They remain an important tool for researchers (Browman and Woo 2017).

R is not that great of a spreadsheet; packages are available to seamlessly tie your spreadsheet and database data to R via **ODBC**. We will routinely enter and manipulate data in MS Excel, then import the data into R for analysis.

Spreadsheet apps like MS Excel and Google Sheets (see also LibreOffice Calc) are great at being a spreadsheet program; R is great at being a statistical software program. You should take advantage of what the tools do best.

Still not convinced?

R is in use all over the world by students and professionals alike, and if one is going to spend the time to learn how to use a statistics software program, you should learn a standard program, like R.

And it's not just me. Read about R in this 2009 New York Times piece, "Data analysts captivated by R's power." Look who purchased (April 2015) Revolution Analytics, a major player in the development of the R programming language.

Note:

The answer was Microsoft. For several years Microsoft supported R development via Microsoft Machine Learning Server & Microsoft R Open. However, as of July 2023, this service is no longer available. See Microsoft R application network retirement.

Why install R on your computer?

Convenience. Control. Offline.

At the Biology department of Chaminade University, we have installed and maintain R, Rcmdr, and RStudio along with all required packages on our Macbook Pro® Lab computers for your use during class and during optional, proctored biostatistics work sessions. Since 2018, R is increasingly available "in the cloud" (e.g., RStudio Cloud), which would mean you could run R in your browser and avoid installation on your computer. You can run significant analysis with R in the cloud via the free Google Colaboratory and CoCalc platforms that are now available: I encourage you to look into these platforms. Unfortunately, these services are not quite ready for the classroom. For example, RStudio in the Cloud is free to use on a limited basis, but quickly

requires a significant subscription cost with increasing use. Google Colab and GoCalcs require use of Jupyter notebooks, which add yet another layer to the learning curve without focusing on learning statistics. Second, although access to their servers is easy, running simultaneous connections via Chaminade's single public IP address is likely to lead to problems for us. Third, I want you to use R Commander (Rcmdr) to assist in the learning curve — `Rcmdr` cannot be run in the Cloud (i.e., RStudio in the Cloud, Google Colaboratory, or CoCalc).

Therefore, you are encouraged to install R, Rcmdr, and even RStudio onto your own computers, in part because of the convenience, but also because R is not generally available to students on campus, i.e., only the Biology department's computers have the up-to-date R software installed.

To get started, go to your Canvas website and view the file How to install R on your own computer.

An additional benefit to installing a version of R on your computer, you'll understand more about the software if you take the time to install and if need be, troubleshoot your installation of the software. Moreover, there's a considerable amount of help out there for R. For example, a simple Google search (keywords: tutorial "install R"), returns more than 700K hits, and more than 40K January 2023 alone (add "after:2023-01-01" to Google search box). In fact, there's so much out there that you'll want to sample from several sites and select the voice that works best for you.

Questions

1. Conduct the search on Google for tutorials on installing R; find 10 sites and rank them 1 to 10, with 1 being the site you like best and 10 being the one you like least.
    1. For example, I like https://bookdown.org/ndphillips/YaRrr/, which is an online book for working with R and includes detailed instructions for installing R.
2. What are the three reasons I offered to justify use of R over other candidate statistical applications?
3. R may be installed on the public computers available to you in the lab. Check to see if this is true, and if so, what version of R is installed?
4. What does Rcmdr stand for?
5. In your own words, define and contrast GUI applications from IDE applications
6. Try some R work yourself
    1. In R (or Rcmdr), copy and paste the code above for the `A` variable, then create the `B` variable. What happens when you type the variable name by itself at the R prompt?
    2. Make a plot of `A` and `B`, but this time plot `A` against `B`.
        1. What can you conclude about the axis order in the function?
