"there is art to data science"

# NBME useRs

VOL.I. . . No.1 *NOVEMBER 16, 2020* PRICELESS

## Welcome!

Thank you all for your interest in learning more about R and joining the R community at NBME. I'm very excited to report that as of this writing that 46 people have signed up to be a part of the R useRs group[1]. This first newsletter is written assuming that you have little or no experience with R. The vast majority of you (~75%) indicated that you were a novice / new to R, so you are all in good company.

## About R [2]


### Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, . . . ) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

[1] "useRs" is an R-specific stylization of denoting a computer program user. For example, useR! is the international R users conference. Thus the use of 'useRs' is not my cheesy idea, but I'm happy to adopt it.

[2] Taken verbatim from the What is R? page

## The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes:

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

---

That was a lot, but it is negligent to not give this introduction when first learning about R, and I cannot state this information better or more succinctly than the creators of R themselves. Admittedly some the terminology / jargon in this introduction warrants unpacking, and this will come in due time. Now on with the fun!

# Installing R

To use R, one must first install the R software. One can download the appropriate version of R (Windows / Mac / Linux) from the Cromprehensive R Archive Network. For those with previous R experience it is worth noting there were some substantial changes made when R transitioned from the 3.x to 4.x version.

Once you have downloaded R, you will naturally be interested to see what it's like to use the software. You'll be confronted with a screen similar to this (your version of R may vary from mine):

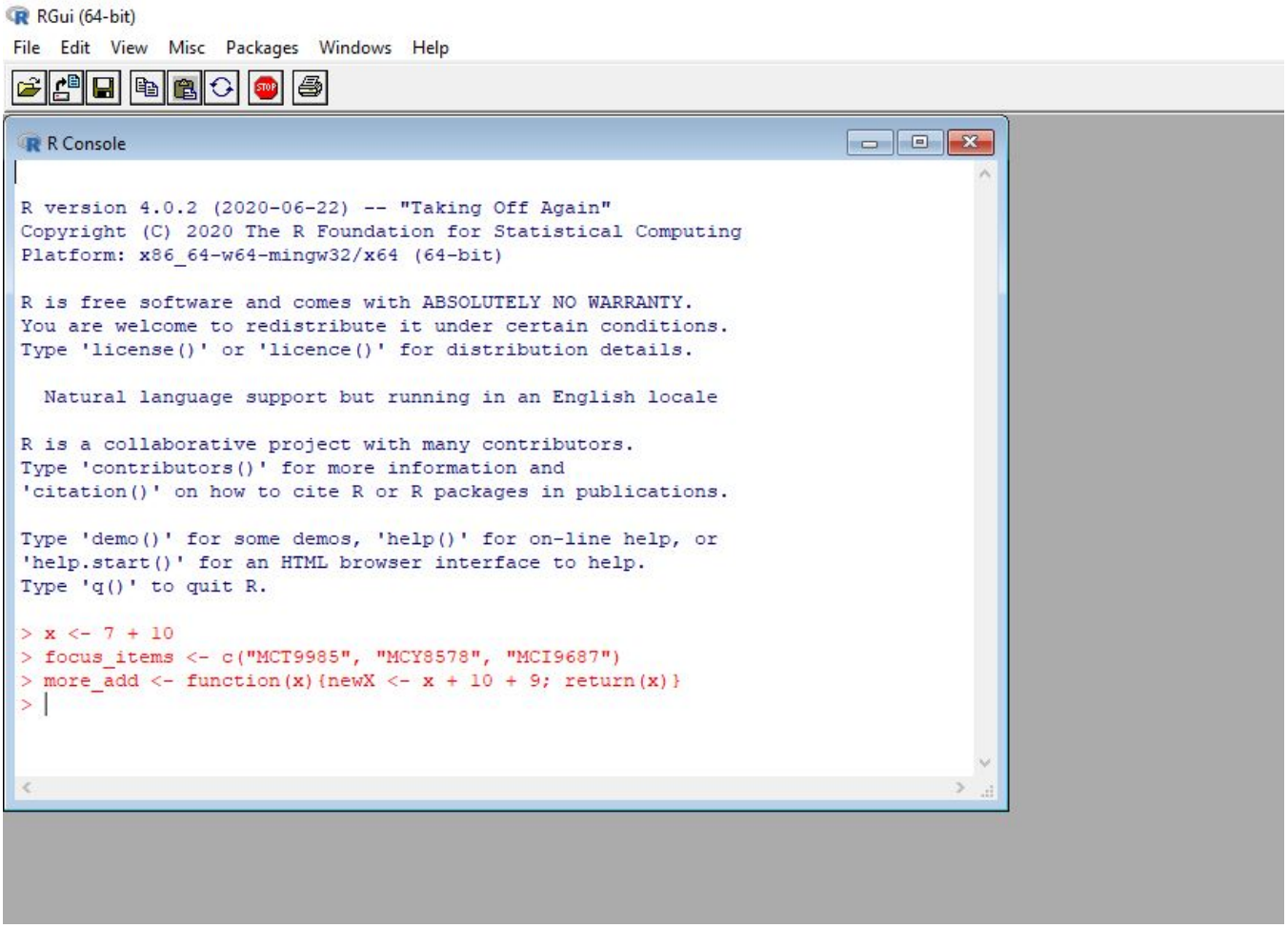


Figure 1: Uninformative R GUI

This is the basic R console, where one can enter syntax and execute a variety of R arguments. However, for those of you with SPSS or SAS experience, this interface may seem difficult. There's not much to it, and once you've typed a command and run it, it's not always clear what happened. For instance, you can see I've entered a variety of commands and there doesn't seem to be any additional information available.

For these reasons (and more), several developers got together years ago and decided to create an interface to make the R workflow experience more user-friendly and efficient. And thus was born RStudio.

# Installing RStudio

I believe that RStudio is one of the main reasons that R has become so popular in the last decade. In addition to creating the user-friendly integrated development environment (IDE), RStudio employs some of the most prolific state-of-the-art developers and R useRs (including my personal hero, Hadley Wickham). The RStudio group continues to push the boundaries of what R can do by developing unique packages that facilitate workflows and far extend R's capabilities. Much of what you will learn and use can probably be traced back to some of these RStudio individuals.

You can navigate to the RStudio installation page by clicking on this sentence. In the vast majority of cases, the Free version of RStudio Desktop will be more than sufficient. RStudio will automatically detect any version of R that is installed, so you can just open RStudio when starting to work.

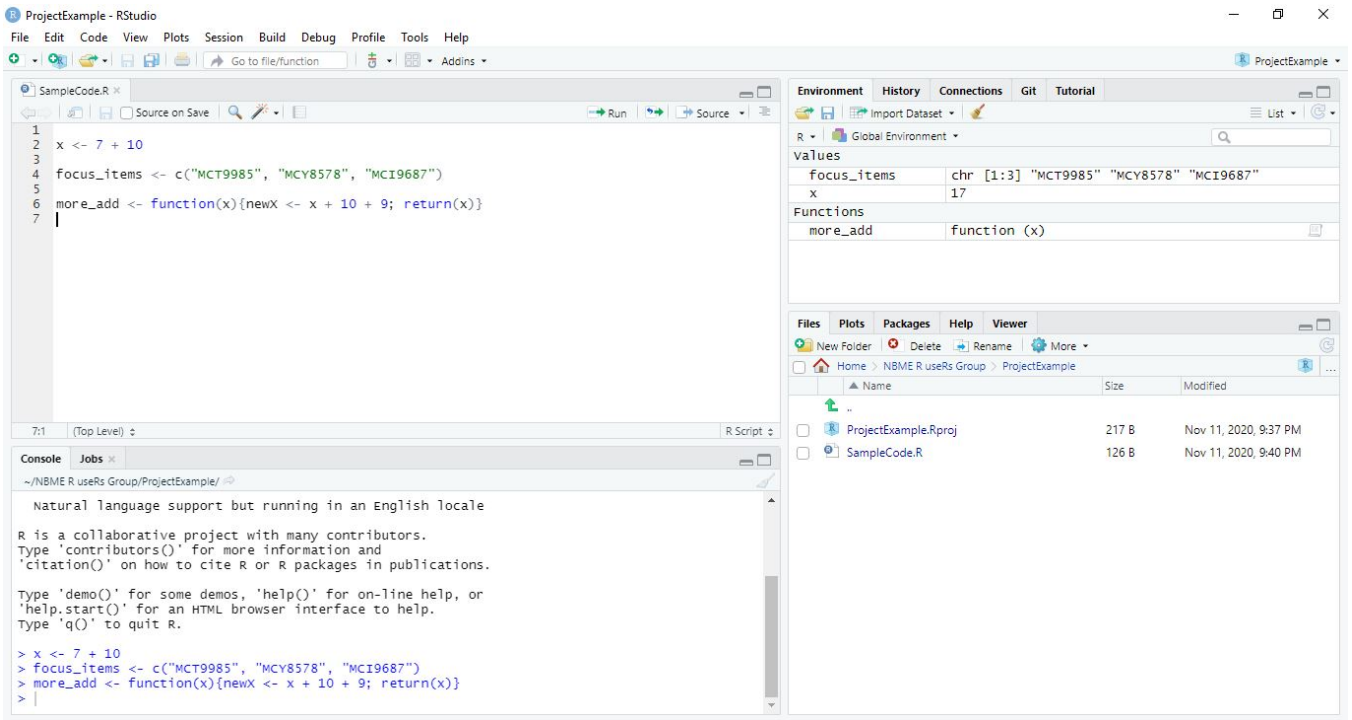


Figure 2: Informative R IDE

This is the basic RStudio interface, which is only a small part of what RStudio can do. I would like to point out the top left corner, which is the syntax window where we have defined the same objects that we saw above in the regular R interface. The lower left corner is the console, which replicates what you saw on the page above. When you run syntax in the top left script window, those commands are mirrored in the console. One can also run commands directly in the console, but this leaves no record of what commands have been run. This can be helpful in some specific instances, but for most purposes you'll want to use the top left window. The top right window displays the local environment and shows you how specific variables have been defined. The bottom right window displays the objects saved in the working directory. Fully going through RStudio warrants several newsletters or a workshop, which we will happily provide in the future.

# Learning Resources

The number of resources available online grows every day. This isn't an exaggeration - there are many blogs, forums, and message boards for R users with active participation. I'll arbitrarily break these resources up into online books, online courses, and websites/blogs of interest. I want to thank **Angelo D'Addario**, **Drew Houriet**, and **Matt Roumaya** for helping me expand these sections.

## Online Books

A cursory search on Amazon shows several books available for learning R / R reference. I've looked through many of these books and think a couple are pretty decent, but there are several ***free*** books online that are just as good.

**Bookdown.org** is one such place to find high-quality free books. These are all books created using the `bookdown` R package and are freely available. My favorite reference / help book here is R for Data Science by Hadley Wickham and Garrett Grolemund. Another good starter is R Programming for Data Science by Roger Peng, who also has a great Coursera course available (see below). I encourage you to look through all of the titles to get a sense of the free resources available to you as you continue to learn.

## Online Courses

The availability and price of the online courses will vary. Some courses are free for a trial period, some are free forever but require payment for an official certificate of completion (after successfully completing required projects, of course), while others require payment with no certificate option. I *think* this may even vary within the different platforms: Datacamp, Coursera, edX, Codeacademy, etc. Below I will mention those courses that people have mentioned to me as useful. **This is by no means an exhaustive list.** It may be a wonderful resource even if it's not mentioned below.

Roger Peng's R Programming Coursera is a good option for learning R online in a class-like environment, and there are multiple Coursera R course offerings. Charlotte Wickham, Hadley's sister, is a professor at Oregon State and also contributes to many of the Codeacademy R courses. The edX R Data Science: The Basics course with accompanying book also comes recommended. Several people recommended the Harvard CS50 Course. The Harvard CS50 course is not R specific, but instead is a general survey of programming / programming theory, which can be valuable knowledge when starting your programming career.

If you venture out into the world of online R courses, please do send me your thoughts (positive or negative) on the course you tried so I can share this information with others.


## Websites / Blogs

r-bloggers is a blog aggregator of content contributed by bloggers who write about R, and you can sign up for a daily newsletter in the upper-right hand corner of the blog. **I highly recommend signing up for that newsletter.** What I like about r-bloggers is that it hosts a wide variety of content for a very large audience. It has everything from beginner tips to extremely esoteric analyses to silly/fun/humorous content. The daily newletter has a table of contents at the top and an excerpt from each entry lower on the page. It takes only a few seconds to check out the table of contents, and only a few minutes to scan the newsletter if you skip irrelevant / uninteresting posts.

Whenever you run into a problem with R where the solution isn't obvious in an R book/course, just search for what you want to know in Google: "how to find leverage statistics"; "Pass a data.frame column name to a function"; or "What does XXX error mean?". There are a number of forums where people have probably posted a similar question to yours and you can get some help. Admittedly it can take a bit of trial and error to figure out what the right search terms might be, but with enough persistence you will most likely find a solution.

# R Packages

As mentioned on the second page, R is the statistical programming powerhouse that it is because of the numerous packages that have been developed that significantly extend the functionality of R. (To be fair, much of the same can be said for Python.) This includes all facets important to data science: infrastructure, data storage, data wrangling, data analysis, graphics, publishing, reporting, custom applications, and much, much, much more.

In order to a package to be an "official" R package, it needs to meet certain requirements set by R, including compatibility capabilities and responding to issues brought about by the R community. However, it is important to note that these standards **do not include ensuring that a package does what it purports to do.** This is especially important when using a package that is innovative or allows for the use of a new statistical model.

R packages are hosted on the CRAN contributed packages page. If you navigate to the page where packages are listed by name, you will see that searching by package names can make it difficult to locate the packages that are most relevant to your task. The TASK VIEWS groups together packages by topic type, although being classified in one of these task views depends on the developer indicating this information when the package is submitted. Thus, the number of packages that truly fall under each topic are more than what are listed on each of the individual pages.

When you install the base R program, it comes with a handful of packages that can be used to accomplish the most basic data science needs. While can do a lot with the base R package, this doesn't mean that the syntax to accomplish your goals is easy or intuitive, nor will the results of your syntax be in a form that is amenable to being saved and properly stored. Fortunately RStudio has developed a suite of packages that significantly improve these basic capabilities. These are packaged together as `tidyverse` and include the `ggplot2`, `dplyr`, `tidyr`, `readr`, `purrr`, `tibble`, `stringr`, and `forcats` packages.

Other packages of note are Rmarkdown and Shiny. Each of these packages warrants a few minutes of your attention and will likely be featured in an upcoming newsletter. Rmarkdown was even used to create this newsletter!

# Featured R packages

Each newsletter will feature *at least* one R package. I hope to use this space to showcase the flexibility and adaptability of R for a wide variety of tasks. This is one place where I hope I can convince you all to make a contribution to the newsletter. Although we are all at the same organization, our uses and needs of R will significantly differ. My contributions will be about statistical analyses, data wrangling, graphics, and custom applications, and not all of this information will be relevant to all users across NBME.

## R package: `tableone`

**tableone** pdf vignette
*A great package to easily calculate descriptive statistics.*
Suggested audience: PADA, CAA, TD; anyone in need of descriptive statistics.

| | Stratified by trt | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | p | test | SMD |
| n | 158 | 154 | | | |
| time (mean (SD)) | 2015.62 (1094.12) | 1996.86 (1155.93) | 0.883 | | 0.017 |
| status (%) | | | 0.894 | | 0.054 |
| 0 | 83 (52.5) | 85 ( 55.2) | | | |
| 1 | 10 ( 6.3) | 9 ( 5.8) | | | |
| 2 | 65 (41.1) | 60 ( 39.0) | | | |
| trt = 2 (%) | 0 ( 0.0) | 154 (100.0) | <0.001 | | NaN |
| age (mean (SD)) | 51.42 (11.01) | 48.58 (9.96) | 0.018 | | 0.270 |
| sex = f (%) | 137 (86.7) | 139 ( 90.3) | 0.421 | | 0.111 |
| ascites = 1 (%) | 14 ( 8.9) | 10 ( 6.5) | 0.567 | | 0.089 |
| hepato = 1 (%) | 73 (46.2) | 87 ( 56.5) | 0.088 | | 0.207 |
| spiders = 1 (%) | 45 (28.5) | 45 ( 29.2) | 0.985 | | 0.016 |
| edema (%) | | | 0.877 | | 0.058 |
| 0 | 132 (83.5) | 131 ( 85.1) | | | |
| 0.5 | 16 (10.1) | 13 ( 8.4) | | | |
| 1 | 10 ( 6.3) | 10 ( 6.5) | | | |
| bili (median [IQR]) | 1.40 [0.80, 3.20] | 1.30 [0.72, 3.60] | 0.842 | nonnorm | 0.171 |
| chol (median [IQR]) | 315.50 [247.75, 417.00] | 303.50 [254.25, 377.00] | 0.544 | nonnorm | 0.038 |
| albumin (mean (SD)) | 3.52 (0.44) | 3.52 (0.40) | 0.874 | | 0.018 |

Figure 3: One possible output from the tableone package.

Most of the time when you are starting an analysis it is important to view the basic descriptive statistics of your sample and key variables. Not only does this help you become more familiar with your sample, but can also forecast potential results or difficulties with analyses (e.g. hints of extreme skew or kurtosis). However, when you need to get the descriptive statistics for many variables, writing the syntax to get this information can be a bit tedious. Or, even when you only have a few variables, it can take some work to get these results into a form that is publication-worthy. The `tableone` package was made to make this process easier. You can explicitly declare what variables you want in the table, you can choose what descriptives are output, and you can even include simple analyses of differences when you're reporting statistics for groups (as has been done above). `tableone` isn't perfect - sometimes the resulting output still needs some grooming prior to publication - but I highly recommend it as part of your normal workflow for any analysis.

## R package: `wesanderson`

**wesanderson** pdf vignette
*Apply color schemes from Wes Anderson movies to your graphs.*
Suggested audience: PADA, CAA, TD; anyone interested in graphing

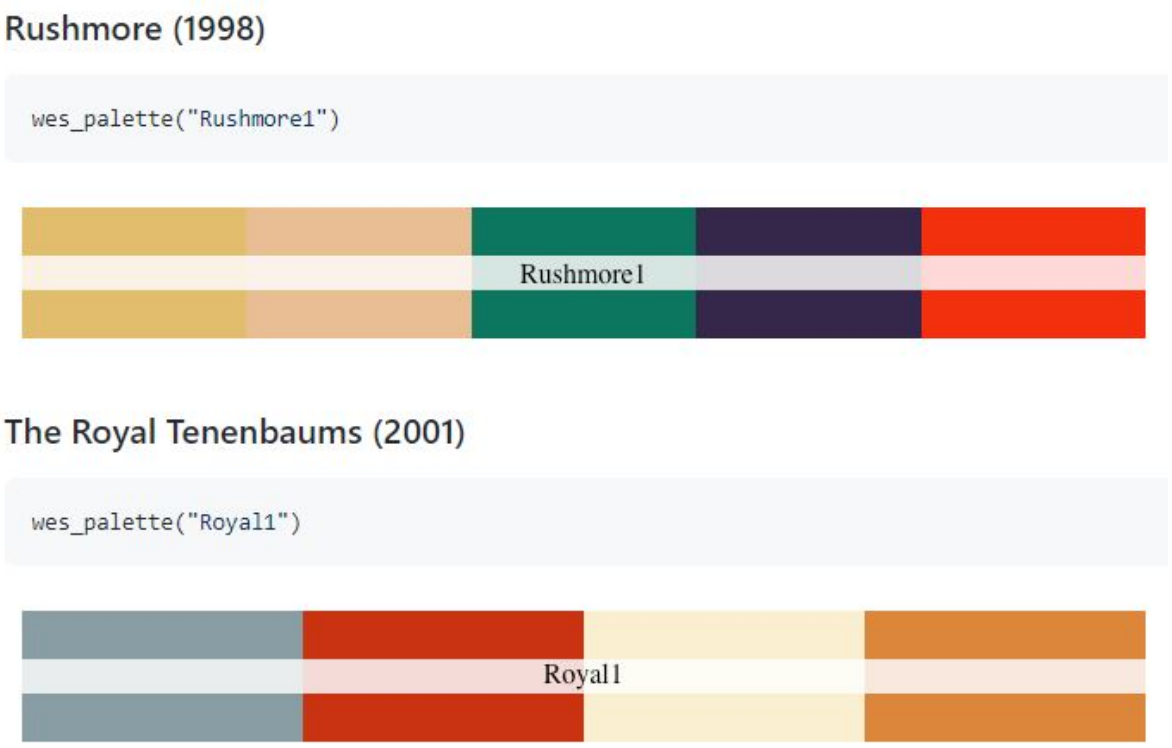


Figure 4: Example palettes from the wesanderson package

I am a huge fan of Wes Anderson movies. I love the universes that he creates for his stories, down to the scenery, costumes, and color palettes. You can bet I was excited to hear that someone had made an R package that allows you to apply these color schemes to your own graphs. Amazing, right?! Well, not really. The first inconvenience is that the palattes can sometimes only be used when you have the right amount of levels for a variable (usually 4 or 5). Even then I didn't think the palettes lent themselves to visualization best practices, and I often abandoned them in favor of better contrasts or gradients. This isn't to say that I don't *ever* use these colors - I've snuck them into a few conference presentations and internal reports - but in general I found I liked the idea of the package better than its actual function.

**Want to contribute an R package review?** There are two ways that you can do so:

- You can click this link to go to a Google form that asks for all of the necessary information.
- You can email me at CRunyon@nbme.org with the necessary information, using the informaiton above as a template. The image is optional - send one if you have it, or I can easily make one.

# *Data Science Oath*[3]

I swear to fulfill, to the best of my ability and judgment, this covenant:

I will respect the hard-won scientific gains of those data scientists in whose steps I walk and gladly share such knowledge as is mine with those who follow.

I will apply, for the benefit of society, all measures which are required, avoiding misrepresentations of data and analysis results.

I will remember that there is art to data science as well as science and that consistency, candor, and compassion should outweigh the algorithm's precision or the interventionist's influence.

I will not be ashamed to say, "I know not," nor will I fail to call in my colleagues when the skills of another are needed for solving a problem.

I will respect the privacy of my data subjects, for their data are not disclosed to me that the world may know, so I will tread with care in matters of privacy and security. If it is given to me to do good with my analyses, all thanks. But it may also be within my power to do harm, and this responsibility must be faced with humbleness and awareness of my own limitations.

I will remember that my data are not just numbers without meaning or context, but represent real people and situations, and that my work may lead to unintended societal consequences, such as inequality, poverty, and disparities due to algorithmic bias. My responsibility must consider potential consequences of my extraction of meaning from data and ensure my analyses help make better decisions.

I will perform personalization where appropriate, but I will always look for a path to fair treatment and nondiscrimination.

I will remember that I remain a member of society, with special obligations to all my fellow human beings, those who need help and those who don't.

If I do not violate this oath, may I enjoy vitality and virtuosity, respected for my contributions and remembered for my leadership thereafter. May I always act to preserve the finest traditions of my calling and may I long experience the joy of helping those who can benefit from my work.

---

## THANK YOU!

Thank you for reading this far! I hope that you have found this newsletter useful. Without your interest and contributions, we couldn't be a group. Future newsletters will not be this long. Or, if they are, it's because there is content will be predictably long and can be consumed only if desired (e.g. multiple R package reviews). Happy coding!

[3]Taken from Data Science for Undergraduates: Opportunities and Options