Introduction to Reproducible Research

March, 2024

Table of Contents

# About this Course

This course will introduce the learner to the basic concepts of reproducibility and replication of scientific research. The premise of the course is that reproducibility failure can be traced to many different root causes and being able to identify these root causes will provide the basis for preventing future failure. The course will provide case studies from three historical data analyses depicting what constitutes a failure of reproducibility. The motivating case study comes from a study describing the development of microarray-based genomic signatures for predicting chemotherapy response. This case study highlights various aspects of data analysis that can lead to reproducibility failure, such as workflow management, data analytic modeling choices, stakeholder engagement, team organization, and communication.

## Learning Objectives

Upon completing this module, a learner will be able to

1. Describe the requirements for reproducible analyses
2. Identify stakeholders affected by reproducibility of an analysis (i.e. collaborators, students, other researchers)
3. Describe the impact of reproducibility on various data analysis stakeholders
4. Characterize the difference between reproducibility and replication
5. Identify organizational or analytic problems that can lead to non-reproducible research

## Available course formats

This course is available in multiple formats which allows you to take it in the way that best suites your needs. You can take it for certificate which can be for free or fee.

* The material for this course can be viewed without login requirement on this [Bookdown website](LINK%20HERE). This format might be most appropriate for you if you rely on screen-reader technology.
* This course can be taken for [free certification through Leanpub](LINK%20HERE).
* This course can be taken on [Coursera for certification here](LINK%20HERE) (but it is not available for free on Coursera).
* Our courses are open source, you can find the [source material for this course on GitHub](https://github.com/fhdsl/Introduction_to_reproducible_research).

# 1 Introduction

This chapter will be about the concept of reproducible research and we will cover some basic concepts and ideas that are related to reproducible research. Before we get to *reproducibility*, we need to cover a little background with respect to how science works (even if you’re not a scientist, this is important). The basic idea is that in science, *replication* is the most important element of verifying and validating findings. So if you claim that X causes Y, or that Vitamin C improves disease, or that something causes a problem, what happens is that other scientists that are independent of you will try to investigate that same question and see if they come up with a similar result. If lots of different people come up with the same result and replicate the original finding, then we tend to think that the original finding was probably true and that this is a real relationship or real finding.

The ultimate standard in strengthening scientific evidence is replication. The goal is to have independent people to do independent things with different data, different methods, and different laboratories and see if you get the same result. There’s a sense that if a relationship in nature is truly there, then it should be robust to having different people discover it in different ways. Replication is particularly important in areas where findings can have big policy impacts or can influence regulatory types of decisions.

## 1.1 What’s Wrong with Replication?

So what’s wrong with replication? There’s really nothing wrong with it. This is what science has been doing for a long time, through hundreds of years. And there’s nothing wrong with it today. But the problem is that it’s becoming more and more challenging to do replication or to replicate other studies. Part of the reason is because studies are getting bigger and bigger.

In order to do big studies you need a lot of money and so, well, there’s a lot of money involved! If you want to do ten versions of the same study, you need ten times as much money and there’s not as much money around as there used to be. Sometimes it’s difficult to replicate a study because if the original study took 20 years to do, it’s difficult to wait around another 20 years for replication. Some studies are just plain unique, such as studying the impact of a massive earthquake in a very specific location and time. If you’re looking at a unique situation in time or a unique population, you can’t readily replicate that situation.

There are a lot of good reasons why you can’t replicate a study. If you can’t replicate a study, is the alternative just to do nothing, just let that study stand by itself? The idea behind a reproducible research is to create a kind of minimum standard or a middle ground where we won’t be replicating a study, but maybe we can do something in between. The basic problem is that you have the gold standard, which is replication, and then you have the worst standard which is doing nothing. What can we do that’s in between the gold standard and doing nothing? That is where reproducibility comes in. That’s how we can kind of bridge the gap between replication and nothing.

In non-research settings, often full replication isn’t even the point. Often the goal is to preserve something to the point where anybody in an organization can repeat what you did (for example, after you leave the organization). In this case, reproducibility is key to maintaining the history of a project and making sure that every step along the way is clear.

## 1.2 Reproducibility to the Rescue

Why do we need this kind of middle ground? We haven’t clearly defined reproducibility yet, but the basic idea is that you need to make the **data** available for the original study and the **computational methods** available so that other people can look at your data and run the kind of analysis that you’ve run, and come to the same findings that you found.

What reproducible research is about is a *validation of the data analysis*. Because you’re not collecting independent data using independent methods, it’s a little bit more difficult to validate the scientific question itself. But if you can take someone’s data and reproduce their findings, then you can, in some sense, validate the data analysis. This involves having the data and the code because more likely than not, the analysis will have been done on the computer using some sort of programming language, like R. So you can take their code and their data and reproduce the findings that they come up with. Then you can at least have confidence that the analysis was done appropriately and that the correct methods were used.

Recently, there’s been a lot of discussion of reproducibility in the media and in the scientific literature. The journal *Science* had a special issue on reproducibility and data replication. Other journals have updated policies on publication to encourage reproducibility. In 2012, a feature on the TV show 60 minutes looked at a major incident at Duke University where many results involving a promising cancer test were found to be not reproducible (more on that later). This led to a number of studies and clinical trials having to be stopped, followed by an investigation which is still ongoing.

## 1.3 Reproducibility Guidelines

Finally, the Institute of Medicine (now the National Academy of Medicine), in response to a lot of events involving reproducibility of scientific studies, issued a report saying that best practices should be done to promote and encourage reproducibility, particularly in what’s called ’omics based research, such as genomics, proteomics, other similar areas involving high-throughput biological measurements. This was a very important report. Of the many recommendations that the IOM made, the key ones were that

* Data and metadata need to be made available;
* Computer code should be fully specified, so that people can examine it to see what was done;
* All the steps of the computational analysis, including any preprocessing of data, should be fully described so that people can study it and reproduce it.

We will expand on these ideas in the chapters that follow.

## 1.4 Summary

* Replication, whereby scientific questions are examined and verified independently by different scientists, is the gold standard for scientific validity.
* Replication can be difficult and often there are no resources to independently replicate a study.
* Reproducibility, whereby data and code are re-analyzed by independent scientists to obtain the same results of the original investigator, is a reasonable minimum standard when replication is not possible.

# 2 From “X” to “Computational X”

What is driving this need for a “reproducibility middle ground” between replication and doing nothing? For starters, there are a lot of new technologies on the scene and in many different fields of study including, biology, chemistry and environmental science. These technologies allow us to collect data at a much higher throughput so we end up with these very complex and very high dimensional data sets. These datasets can be collected almost instantaneously compared to even just ten years ago—the technology has allowed us to create huge data sets at essentially the touch of a button. Furthermore, we the computing power to take existing (already huge) databases and merge them into even bigger and bigger databases. Finally, the massive increase in computing power has allowed us to implement more sophisticated and complex analysis routines.

The analyses themselves, the models that we fit and the algorithms that we run, are much much more complicated than they used to be. Having a basic understanding of these algorithms is difficult, even for a sophisticated person, and it’s almost impossible to describe these algorithms with words alone. Understanding what someone did in a data analysis now requires looking at *code* and scrutinizing the computer programs that people used.

The bottom line with all these different trends is that for every field “X”, there is now “Computational X”. There’s computational biology, computational astronomy—whatever it is you want, there is a computational version of it.

## 2.1 Example: Air Pollution and Health

One example of an area were reproducibility is important comes from the area of air pollution and health. Air pollution and health is a big field and it involves a confluence of features that emphasize the need for reproducibility.

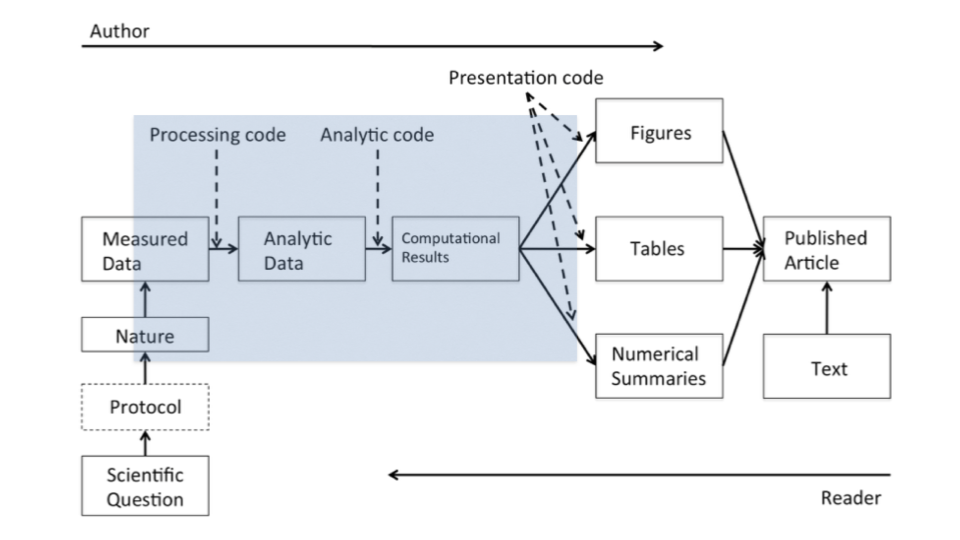
The first feature is that we are estimating very small, but very important, public health effects in the presence of a numerous much stronger signals. You can think about air pollution as something that’s perhaps harmful, but even if it were harmful there are likely many other things that are going to be more harmful that you have to worry about. Pollution is going to be at the very top of the list of things that are going to harm you. In other words, there’s an inherently weak signal there.

Second, the results of a lot of air pollution research inform substantial policy decisions. Many federal air pollution regulations in the United States are based on scientific research in this area and these regulations can affect a lot of stakeholders in government and industry.

Finally, we use a lot of complex statistical methods to do these studies and these statistical methods are subsequently subjected to intense scrutiny. The combination of an inherently weak signal, substantial policy impacts, and complex statistical methods almost require that the research that we do be reproducible.

## 2.2 The Data Science Pipeline

The basic issue is when you read a description of a data analysis, such as in an article or a technical report, for the most part, what you get is the report and nothing else. Of course, everyone knows that behind the scenes there’s *a lot* that went into this report and that’s what I call the **data science pipeline**.



In this pipeline, there are two “actors”: the *author* of the report/article and the *reader*. On the left side, the author is going from left to right along this pipeline. The reader is going from right to left. If you’re the reader you read the article and you want to know more about what happened: Where is the data? What was used here? The basic idea behind reproducibility is to focus on the analytic data and the computational results. With reproducibility the goal is to allow the author of a report and the reader of that report to “meet in the middle”.

## 2.3 Elements of Reproducibility

What do we need for reproducibility? There’s a variety of ways to talk about this, but one basic definition that we’ve come up with is that there are four things that are required to make results reproducible:

1. **Analytic data**. The data that were used for the analysis that was presented should be available for others to access. This is different from the *raw data* because very often in a data analysis the raw data are not all used for the analysis, but rather some subset is used. It may be interesting to see the raw data but impractical to actually have it. Analytic data is key to examining the data analysis.
2. **Analytic code**. The analytic code is the code that was applied to the analytic data to produce the key results. This may be preprocessing code, regression modeling code, or really any other code used to produce the results from the analytic data.
3. **Documentation**. Documentation of that code and the data is very important.
4. **Distribution**. Finally, there needs to be some standard means of distribution, so all this data in the code is easily accessible.

## 2.4 Authors and Readers

It is important to realize that there are multiple players when you talk about reproducibility–there are different types of parties that have different types of interests. There are authors who produce research and they want to make their research reproducible. There are also readers of research and they want to reproduce that work. Everyone needs tools to make their lives easier.

One current challenge is that authors of research have to undergo considerable effort to make their results available to a wide audience. Publishing data and code today is not necessarily a trivial task. Although there are a number of resources available now, that were not available even five years ago, it’s still a bit of a challenge to get things out on the web (or at least distributed widely). Resources like [GitHub](https://github.com) and various data repositories have made a big difference, but there is still a ways to go with respect to building up the public reproducibility infrastructure.

Furthermore, even when data and code are available, readers often have to download the data, download the code, and then they have to piece everything together, usually by hand. It’s not always an easy task to put the data and code together. Also, readers may not have the same computational resources that the original authors did. If the original authors used an enormous computing cluster, for example, to do their analysis, the readers may not have that same enormous computing cluster at their disposal. It may be difficult for readers to reproduce the same results.

Generally the toolbox for doing reproducible research is small, although it’s definitely growing. In practice, authors often just throw things up on the web. There are journals and supplementary materials, but they are famously disorganized. There are only a few central databases that authors can take advantage of to post their data and make it available. So if you’re working in a field that has a central database that everyone uses, that’s great. If you’re not, then you have to assemble your own resources. We will discuss the tools for doing reproducible research in the third course in this series.

## 2.5 Summary

* The process of conducting and disseminating research can be depicted as a “data science pipeline”
* Readers and consumers of data science research are typically not privy to the details of the data science pipeline
* One view of reproducibility is that it gives research consumers partial access to the raw pipeline elements.

# 3 Learning from Failure

Learning about how data analyses succeeds or fails (but more importantly, fails) is extremely challenging without actually going through the process yourself. I don’t think I ever learned about it except through first hand experience, which took place over the course of years. There are a few reasons for this that I have observed over time:

* Success in scientific data analysis is usually concerned with whether the claims made based on the results are *true* or not. If the results feel true, and the analysis appears rigorous, then that’s usually the end of the discussion. Focus is put on the result and what should come next. The underlying idea here is not necessarily misguided: Progress in science depends on independent replication, and any given analysis cannot be assigned too much weight.
* When analyses fail, the results are usually vague and confusing. Furthermore, the public rarely finds out about them because they are not published. This is mostly due to human nature: it’s difficult to motivate oneself to write about an experience that was inconclusive and perhaps incoherent. It can also be embarrassing if honest mistakes were made. Publication of negative studies is a separate matter, because a truly negative study is, in fact, conclusive. But often, we don’t even have that much clarity.
* In the rare cases where we do find out about data analysis failures, the focus is often on who or what is to blame. In cases where criminal activity has taken place, this is an important aspect. However, identifying who or what is to blame usually doesn’t provide us with generalizable knowledge that we can apply to our own data analyses. The underlying assumption of assigning blame is that this failure was a unique situation that could never have happened if the individual to blame had not been involved. Occasionally, there are cases where there is a clear bug in some software that leads to erroneous results. Fixing the bug in the code will “fix” the results, but even in that situation it’s not always clear that the bug is the ultimate cause of failure (although in this case it is the proximate cause).

The case study presented in the next chapter is useful for thinking about what kinds of generalizable knowledge we can obtain from data analysis failures. This case is special because it had serious implications and large parts of it played out in public. While we likely will never know *all* of the details, we know enough to have a meaningful discussion about the lessons learned.

# 4 Case Study: Developing a Genomic Predictor

The case study presented here took place largely at Duke University and has at various times been referred to as the “Duke Saga”. At a high level, it involved a research group that claimed to have developed a genomic signature that predicted whether a person would respond to cancer chemotherapy. When others tried to reproduce the predictive results of the genomic signature, they were unable to do so.

A brief time line is given below:

* In 2006, *Nature Medicine* published a paper by Anil Potti and colleagues titled “Genomic signature to guide the use of chemotherapeutics”. The paper claimed to have developed a classifier based on applying microarray technology to cell lines maintained by the National Cancer Institute (NCI). They claimed the classifier could determine which patients would respond to chemotherapy treatment.
* Keith Baggerly and Kevin Coombes, two biostatisticians at the MD Anderson Cancer Center were inundated by requests from (justifiably) excited colleagues who wanted to use this technology. Baggerly and Coombes attempted to reproduce the results using the published description but were unable to do so. They *were* able to reproduce certain results in the paper after deliberately introducing a series of errors into the data analysis.
* Since this initial incident, a number of other papers from the same lab were scrutinized and numerous errors in analyses were found, many that one might consider basic data handling and wrangling mistakes. In addition, Baggerly and Coombes found circumstantial evidence of deliberate fraud, such as claiming that certain genes were critical to a classifier even though those genes are not included in the microarray claimed to have been used.
* Clinical trials were started at Duke where patients were randomized into different arms of the trial based on the flawed techniques developed by Potti. After numerous scientists wrote a letter to NCI director Harold Varmus, Duke suspended the trials to investigate the situation. An internal Duke panel eventually cleared Potti and colleagues of any wrongdoing and restarted the trials. Time passes and eventually it is discovered by the *The Cancer Letter* that Potti lied on an application for federal funding about once being a Rhodes Scholar. Eventually, the trials were stopped, but only after much public scrutiny and a series of lawsuits.

There is quite a bit more detail to this story, which played out over many years. If you want to hear more about this you can hear about it from [Keith Baggerly himself](https://youtu.be/7gYIs7uYbMo) in this nice lecture.

This saga has been a difficult one to understand from the perspective of drawing generalizable lessons. While it’s fascinating because of the sheer number of problems that occurred, it’s not necessarily clear what intervention could be taken to prevent a similar episode from happening in the future. Problems with reproducibility clearly played a role here given that Baggerly and Coombes were initially unable to reproduce any of the original analyses.

## 4.1 Initial Lessons

While the details of the Duke Saga were at time astonishing, it is difficult to draw any conclusion about what actually went wrong and what approach should be taken to prevent something like this from happening again. Most people on the outside were just speculating about what could have happened and the people who really would know the details weren’t talking very much. Here’s what most people seemed to take away from the publicly available information about the saga:

* **Reproducibility**. There was definitely a reproducible research angle to this saga, in that the analyses that were conducted lacked transparency. There was only sketchy code that was published along with paper and data were not immediately available. However, in a strange sense, much of what came to light did so because the work was ultimately partially reproducible. That is in fact how Baggerly and Coombes discovered all the problems. They were able to reproduce the findings *after* deliberately introducing mistakes in the data. If one went back in time and magically forced everyone in the lab to use R Markdown or Juypter Notebooks, it’s not clear how that would have prevented anything. For starters, everyone within the team had access to the analyses and the data. It’s possible that people outside the team might have discovered problems sooner if the work had been completely reproducible, but Baggerly and Coombes figured things out relatively quickly. Also, that is besides the point: We should not depend on people outside the research team as a primary defense against data analytic failure. It’s not clear that reproducibility is one of the lessons learned from this saga because it’s not clear that it would have made a difference.
* **Expertise**. The basic narrative explaining this saga was that the data analyses were poorly done. Statisticians in particular have focused on the use of proprietary software, non-reproducible workflows (like pointing and clicking in Excel), and incorrect application of otherwise sound statistical methodology (e.g. cross validation). Perhaps if better-trained people had been doing the analyses, none of this would have happened. Perhaps genomic analyses are too complicated for the traditionally trained laboratory scientist. The idea is that this kind of work is “hard to do” and that you need better people (or improve existing people). That is the gist of the summary in [this segment from the television show 60 Minutes](https://youtu.be/66dPIFMJ_-A) on the entire saga.
* **Individual behavior**. Anil Potti, the principal investigator of the study, was eventually fired from Duke over this scandal and most would agree with that decision. If Duke had fired him 10 years ago, then perhaps yes, this research would not have happened at Duke, but it might have happened somewhere else, or it might have happened at Duke but with a different principal investigator. So while Dr. Potti was ultimately responsible for the analyses, his firing does not provide a useful “lesson learned”.

## 4.2 New Information Appears

In January 2015, *The Cancer Letter* [published a blockbuster memo](https://cancerletter.com/articles/20150109_1/) written by Bradford Perez, who in 2008 was a medical student trainee in the Potti lab. He saw what was going on in the lab and recognized its shoddiness. Problems that Baggerly and Coombes had to essentially reverse engineer, Perez saw first hand and immediately recognized them as serious. In fact, in 2008 he wrote a memo to the leadership of his institute describing some of those problems:

“Fifty-nine cell line samples with mRNA expression data…were split in half to designate sensitive and resistant phenotypes. Then in developing the model, *only those samples which fit the model best in cross validation were included*. Over half of the original samples were removed…. This was an incredibly biased approach which does little more than give the appearance of a successful cross validation.” [emphasis added]

He further wrote,

At this point, I believe that the situation is serious enough that all further analysis should be stopped to evaluate what is known about each predictor and it should be reconsidered which are appropriate to continue using and under what circumstances…. I would argue that at this point nothing…should be taken for granted. All claims of predictor validations should be independently and blindly performed.”

The memo was ignored by Institute leadership. Nothing was stopped and nothing was changed at the time. Perez eventually took his name off a series of papers and left the lab.

## 4.3 Lessons Learned

The Perez memo is critical because it fundamentally changes the narrative about what went wrong in this entire saga. Yes, genomic analyses are “hard to do” but clearly there was expertise in the lab to recognize that difficulty and to recognize when statistical methods were being incorrectly applied. The problem was not a lack of training, nor was it simply the result of a few honest data management mistakes here and there. The problem was a breakdown in communication and a total lack of trust between investigators and members of the data analytic team. Perez clearly felt uncomfortable raising these issues in the lab and wrote the memo knowing that he had “much to lose”. He thought the problem in the lab was that statistical methods were being misapplied, but the real problem in the lab was that *he did not feel comfortable discussing it*. A breakdown in the relationship between an analyst and an investigator is a serious *data analytic problem*.

It’s possible to imagine an alternate scenario where a data analyst like Perez sees a problem with the way models are being developed or applied, mentions this to the principal investigator and has a detailed discussion, perhaps seeks outside expertise (e.g. from a statistician), and then modifies the procedure to fix the problem. This is a process that happens pretty much every day in many labs around the world. No data analysis is perfect from start to finish. Changes and course corrections are constantly made along the way. problems that can be traced to data collection can be raised with the principal investigator. When results are given to other investigators, sometimes the results don’t seem right to them and they will seek clarification. Mistakes can be fixed and results can be updated.

When the relationships between an analyst and various members of the investigator team are strong and there is substantial trust between them, honest mistakes are just minor bumps in the road that can be uncovered, discussed, and fixed. When there is a breakdown in those relationships, the exact same mistakes are covered up, denied, and buried. A breakdown in the relationships between analysts and other investigators on the team generally cannot be fixed with a better statistical method, or a reproducible workflow, or open source software. Recognizing that this is the problem is difficult because often there is no easy solution.

## 4.4 Summary

The data analytic lesson learned from the Duke Saga is that data analysts need to be allowed to say “stop”. But also, the ability to do so depends critically on the relationships between the analyst and members of the investigator team. If an analyst feels uncomfortable raising analytic issues with other members, then arguably all analyses done by the team are at risk. No amount of statistical expertise or tooling can fix this fundamental human problem.

# About the Authors

These credits are based on our [course contributors table guidelines](https://www.ottrproject.org/more_features.html#giving-credits-to-contributors).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [Roger Peng](https://rdpeng.org) |
| **Technical** |  |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([ottrpal](https://github.com/jhudsl/ottrpal)) [Candace Savonen](https://www.cansavvy.com/), [John Muschelli](https://johnmuschelli.com/), [Carrie Wright](https://carriewright11.github.io/) |  |
| **Art and Design** |  |
| Illustrator(s) | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| **Funding** |  |
| Funder(s) | NIH Grant R25GM141505 |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.5 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2024-03-25   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.5)   
## bookdown \* 0.24 2024-03-13 [1] Github (rstudio/bookdown@88bc4ea)   
## cachem 1.0.8 2023-05-01 [1] CRAN (R 4.0.2)   
## callr 3.5.0 2020-10-08 [1] RSPM (R 4.0.2)   
## cli 3.6.2 2023-12-11 [1] CRAN (R 4.0.2)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.23 2023-11-01 [1] CRAN (R 4.0.2)   
## fastmap 1.1.1 2023-02-24 [1] CRAN (R 4.0.2)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.4.2 2020-08-27 [1] RSPM (R 4.0.5)   
## htmltools 0.5.7 2023-11-03 [1] CRAN (R 4.0.2)   
## knitr 1.33 2024-03-13 [1] Github (yihui/knitr@a1052d1)   
## magrittr 2.0.3 2022-03-30 [1] CRAN (R 4.0.2)   
## memoise 2.0.1 2021-11-26 [1] CRAN (R 4.0.2)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.4.0 2020-10-07 [1] RSPM (R 4.0.2)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 1.1.3 2024-01-10 [1] CRAN (R 4.0.2)   
## rmarkdown 2.10 2024-03-13 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.4 2023-11-05 [1] CRAN (R 4.0.2)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2024-03-13 [1] Github (R-lib/testthat@e99155a)   
## usethis 1.6.3 2020-09-17 [1] RSPM (R 4.0.2)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2024-03-13 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# References