Course Name

January, 2024

Table of Contents

# About this Course

## 0.1 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](LINK%20HERE).

## Installing package into '/usr/local/lib/R/site-library'  
## (as 'lib' is unspecified)  
## Installing package into '/usr/local/lib/R/site-library'  
## (as 'lib' is unspecified)  
## Installing package into '/usr/local/lib/R/site-library'  
## (as 'lib' is unspecified)  
## Installing package into '/usr/local/lib/R/site-library'  
## (as 'lib' is unspecified)

## Loading required package: MASS

## Loading required package: HistData

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

##   
## Attaching package: 'UsingR'

## The following object is masked from 'package:survival':  
##   
## cancer

## Warning: package 'reshape' was built under R version 4.0.3

# 1 Week 01

## 1.1 Introduction

### 1.1.1 Welcome to Regression Models

I am happy that you’ve chosen to take Regression Models, part of the Johns Hopkins Data Science Specialization on Coursera! This course presents the fundamentals of regression modeling that you will need for the rest of the specialization and ultimately for your work in the field of data science.

We believe that the key word in Data Science is “science”. Our course track is focused on providing you with three things: (1) an introduction to the key ideas behind working with data in a scientific way that will produce new and reproducible insight, (2) an introduction to the tools that will allow you to execute on a data analytic strategy, from raw data in a database to a completed report with interactive graphics, and (3) on giving you plenty of hands on practice so you can learn the techniques for yourself.

Regression Models represents a both fundamental and foundational component of the series, and it presents the single most practical data analysis toolset. Using only a bare minimum of mathematics, we will attempt to provide you with the fundamentals for the application and practice of regression.

We are excited about the opportunity to attempt to scale Data Science education. We intend for the courses to be self-contained, fast-paced, and interactive, and we intend to run them frequently to give people with busy schedules the opportunity to work on material at their own pace.

### 1.1.2 Some Basics

A couple of first week housekeeping items. First, make sure that you’ve had [R Programming](https://www.coursera.org/learn/r-programming) , the [Data Scientist’s Toolbox](https://www.coursera.org/learn/data-scientists-tools), [Reproducible Research](https://www.coursera.org/learn/regression-models/supplement/uCPA0/welcome-to-regression-models) and [Statistical Inference](https://www.coursera.org/learn/statistical-inference) before taking this class. At a minimum you must know: very basic git, basic R and most of the Statistical Inference Coursera class. The small amount of knitr that you need for the project you can pick up quickly.

An important aspect of this class is to peruse the materials in the github repository. All of the most up to date material can be found [here](https://github.com/bcaffo/courses/tree/master/07_RegressionModels). You should clone this repository as your first step in this class and make sure to fetch updates periodically. (Please issue pull requests so that we may improve the materials!) It is one of the most essential components of the Specialization that you start to use Git frequently. We’re practicing what we preach as well by using the tools in the series to create the series, especially git. Note my [GitHub repo](https://github.com/bcaffo/courses) will generally be more up to date than the Data Science Specialization Repo.

The lectures are in the index.Rmd lecture files. In [Developing Data Products](https://www.coursera.org/learn/data-products), we cover how to create these sorts of slides. However, for the time being, you should be able to open them in R Studio and look at their contents. You will see all of the R code to recreate the lectures. Going through the R code is the best way to familiarize yourself with the lecture materials.

#### 1.1.2.1 YouTube

If you’d prefer to watch the videos on YouTube, you can find them [here](https://www.youtube.com/playlist?list=PLpl-gQkQivXjqHAJd2t-J_One_fYE55tC) and [here](https://www.youtube.com/playlist?list=PLpl-gQkQivXhdgUCdaUQcdb31CRe8Mm2y). If you’d like to keep up with the instructors I’m ([**bcaffo?**](#ref-bcaffo)) on twitter, Roger is ([**rdpeng?**](#ref-rdpeng)) and Jeff is ([**jtleek?**](#ref-jtleek)). The Department of Biostat here is ([**jhubiostat?**](#ref-jhubiostat)).

### 1.1.3 Syllabus (xxx)

Course Title: Regression Models

Course Instructor(s):The primary instructor of this class is [Brian Caffo](https://sites.google.com/view/bcaffo/home/). Brian is a professor at Johns Hopkins Biostatistics and co-directs the [SMART working group](https://www.smart-stats.org).

This class is co-taught by Roger Peng and Jeff Leek. In addition, Sean Kross and Nick Carchedi have been helping greatly.

#### 1.1.3.1 Course Description:

Linear models, as their name implies, relates an outcome to a set of predictors of interest using linear assumptions. Regression models, a subset of linear models, are the most important statistical analysis tool in a data scientist’s toolkit. This course covers regression analysis, least squares and inference using regression models. Special cases of the regression model, ANOVA and ANCOVA will be covered as well. Analysis of residuals and variability will be investigated. The course will cover modern thinking on model selection and novel uses of regression models including scatterplot smoothing.

#### 1.1.3.2 Course Content

This class has three main components:

* Least squares and linear regression
* Multivariable regression
* Generalized linear models

The full list of topics are as follows:

* Module 1, least squares and linear regression
  + 01\_01 Introduction
  + 01\_02 Notation
  + 01\_03 Ordinary least squares
  + 01\_04 Regression to the mean
  + 01\_05 Linear regression
  + 01\_06 Residuals
  + 01\_07 Regression inference
* Module 2, Multivariable regression
  + 02\_01 Multivariate regression
  + 02\_02 Multivariate examples
  + 02\_03 Adjustment
  + 02\_04 Residual variation and diagnostics
  + 02\_05 Multiple variables
* Module 3, Generalized linear models
  + 03\_01 GLMs
  + 03\_02 Binary outcomes
  + 03\_03 Count outcomes
  + 03\_04 Olio
* Module 4, Logistic Regression and Poisson Regression
  + 04\_01 Logistic Regression
  + 04\_02Poisson Regression
  + 04\_03 Hodgepodge

#### 1.1.3.3 Book: Regression Models for Data Science in R.

A companion book is available [here](https://leanpub.com/regmods). The book is published via leanpub, and the suggested price is $14.99. You can get it for free or pay what you feel it is worth.

#### 1.1.3.4 Quizzes

There are four weekly quizzes. You must earn a grade of at least 80% to pass a quiz. You may attempt each quiz up to 3 times in 8 hours. The score from your most successful attempt will count toward your final grade.

#### 1.1.3.5 Course Project

The Course Project is an opportunity to demonstrate the skills you have learned during the course. It is graded through peer assessment. You must earn a grade of at least 80% to pass the peer assessment.

#### 1.1.3.6 Grading Policy

You must score at least 80% on all assignments (Quizzes & Project) to pass the course.

Your final grade will be calculated as follows:

Quiz 1 = 15% Quiz 2 = 15% Quiz 3 = 15% Quiz 4 = 15% Course Project = 40%

##### 1.1.3.6.1 swirl Programming Assignment (optional)

In this course, you have the option to use the [swirl](https://swirlstats.com) R package to practice some of the concepts we cover in lectures.

While these lessons will give you valuable practice and you are encouraged to complete as many as possible, please note that they are completely optional and you can get full marks in the class without completing them.

#### 1.1.3.7 Differences of opinion

Keep in mind that currently data analysis is as much art as it is science - so we may have a difference of opinion - and that is ok! Please refrain from angry, sarcastic, or abusive comments on the message boards. Our goal is to create a supportive community that helps the learning of all students, from the most advanced to those who are just seeing this material for the first time.

### 1.1.4 Data Science Specialization Community Site

Since the beginning of the Data Science Specialization, we’ve noticed the unbelievable passion students have about our courses and the generosity they show toward each other on the course forums. A couple students have created quality content around the subjects we discuss, and many of these materials are so good we feel that they should be shared with all of our students.

We’re excited to announce that we’ve created a site using [GitHub Pages](http://datasciencespecialization.github.io/) to serve as a directory for content that the community has created. If you’ve created materials relating to any of the courses in the Data Science Specialization, please send us a pull request so we can add a link to your content on our site. You can find out more about contributing [here.](https://github.com/DataScienceSpecialization/DataScienceSpecialization.github.io#contributing)

We can’t wait to see what you’ve created and where the community can take this site!

### 1.1.5 Where to get more advanced material

If you want more advanced material, I’ve been working on another version of this class. Eventually I hope to have a second Coursera class as well. Currently, you can get the E-Book in progress [here](https://leanpub.com/lm) (it’s variable pricing including free!)

In addition, you can watch the videos as they’re being developed [here.](https://www.youtube.com/playlist?list=PLpl-gQkQivXhdgUCdaUQcdb31CRe8Mm2y)

## 1.2 Introduction to regression and least squares

Regression models are the workhorse of data science. They are the most well described, practical and theoretically understood models in statistics. A data scientist well versed in regression models will be able to solve an incredible array of problems.

Perhaps the key insight for regression models is that they produce highly interpretable model fits. This is unlike machine learning algorithms, which often sacrifice interpretability for improved prediction performance or automation. These are, of course, valuable attributes in their own rights. However, the benefit of simplicity, parsimony and intrepretability offered by regression models (and their close generalizations) should make them a first tool of choice for any practical problem.

### 1.2.1 Introduction to Regression

Hello, I’m Brian Caffo, and I’d like to welcome you to the introduction to regression lecture in the regression Coursera class, part of our data science specialization. Co-taught by my colleagues Jeff Leek and Roger Peng, we all belong to the Department of Biostatistics at the Johns Hopkins Bloomberg School of Public Health.

Regression is a cornerstone for data scientists. Before delving into complex machine learning, linear regression or its generalization, linear models, are often the go-to procedures. The roots of regression trace back to Francis Galton, who coined the term and concept, along with correlation, closely tied to linear regression.

Galton’s prediction of a child’s height from a parent’s height remains historically significant. Jeff Leek [highlights](https://www.nature.com/articles/ejhg20095) its continued relevance in modern genetic analysis, comparing it to Victorian Era measurements. Moving to a more contemporary example, a blog post by Rafael Irazarry on Simply Statistics explores the relationship between Kobe Bryant’s ball-hogging and the Lakers’ performance, utilizing linear regression.

In a modern example, [Simply Statistics](https://simplystatistics.org) blog talks about “[the Lakers wins](https://simplystatistics.org/posts/2013-01-28-data-supports-claim-that-if-kobe-stops-ball-hogging-the-lakers-will-win-more/)” that Data supports claim that if Kobe stops ball hogging the Lakers will win more.The heart of our class is understanding how to formulate and interpret statements like for example in the Simply Statistics blog post “Linear regression suggests an increase of 1% in the percent of shots taken by Kobe results in a drop of 1.16 points.” We’ll delve into good statistical practices, including providing standard errors.

We might want to find a parsimonious and easily described mean relationships between the parent’s and child’s height. So we don’t want anything complicated. We want the simplest possible relationship, and that is what regression is best at. While machine learning and other techniques generate highly elaborate, in many cases, accurate prediction models, they tend to not be parsimonious. They tend not to explain the data, and they tend not to generate new parsimonious knowledge, whereas this is what regression is good at. This is what regression is in fact best at. We can talk about variation that’s unexplained by the regression model. The so called residual variation.

We’re going to connect the results back to the subject of inference. How do we take our data, which is just a sample, it only talks about that data set, and try to figure out what assumptions are needed to extrapolate it to a larger population. This is a deep subject called statistical inference. We have a whole another course of Statistical Inference as part of data science specialization. But we’re going to apply the tools of inference, which we are hoping most of you will have had as a prerequisite. We’re going to apply the tools of inference to this new subject of regression.

Let’s look at Francis Galton’s data, he first used this data in 1885. He’s really an interesting character in history, in general and definitely in the history of statistics. You need to run install.packages("UsingR"). Here UsingR is the package for the book, [Using R for Introductory Statistics](https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf). It is a great book, and they’ve very kindly packaged all these data sets together in a single R package. So you need to use UsingR then the library UsingR to get a lot of the data sets that we are going to talk about. So let’s first look at the marginal distribution of the parents. In other words, distribution of the parents disregarding children. And the marginal distribution of the children, disregarding parents.

install.packages("UsingR")

Parent distribution is all heterosexual couples, correcting for sex by multiplying the female heights by 1.08.

library(UsingR); data(galton); library(reshape); long<-melt(galton);

## Using as id variables

g<- ggplot(long, aes(x=value, fill=variable))   
g<- g+ geom\_histogram(color='black', binwidth=1)  
g<- g+ facet\_grid(.~variable)  
  
g



On the left, we have the children’s heights. The X-axis is in inches, the scale goes from 60 inches to 75. The Y-axis is the count, the number of children that fall in each bin of heights. On the right in the more bluish teal color, we have the parents heights. We’ve broken the association by the children and the parents by not doing a scatter plot, and only looking at the marginal distribution of the children, and the marginal distribution of the parents by themselves. We would like to use these distributions to introduce least squares, and then we’ll build on the bivaried association after that. So consider only the child’s height,forget for the moment about using the parent’s height to predict the child’s heights. We just want to find maybe the best prediction of the child’s heights without any other information. Well, probably the best predictor would be the middle and how could one define the middle?

One definition, let , be the height for child , where in this dataset . So the middle is the value of that minimizes

That’s how we define the middle. It’s also related to physics in this so called physical center of mass of the histogram that we showed on the previously. Imagine of those bars as being physical entities, having weight and you are trying to figure out where you would put your finger to balance it out. That would be the physical center of mass. You might have guessed that the center of the data has to be the mean.

Let’s use our studio’s manipulate function to experiment with trying to find that center of mass.

library(manipulate)   
myHist<-function(mu){  
 mse<-mean((galton$child-height-mu)^2)  
 g<- ggplot(galton, aes(x=childHeight))+geom\_histogram(fill='salmon',color='black',binwidth=1)  
 g<- g+ geom\_vline(xintercept=mu, size=3)  
 g<- g+ ggtitle(paste("mu=",mu,"MSE=",round(mse,2)),sep="")  
 g  
}  
manipulate(myHist(mu),mu=slider(62,74,step=0.5))

fig xxx

Because we’re using manipulate we can move the slider around and monitor the value of and the mean squared error, that is the sum of the squared distances between the observed data points and that particular value of . If you move the slider around, you would notice notice as we get toward the center of the histogram, the mean squared error is going down and if you keep moving the slider way up, it get’s up large again. You can see is the point that balanced out this histogram.

**Notice** For those that are interested, we cover some simple proofs of some of the statements made. If this isn’t your thing, just skip these sections. However, if you’re interested, get a pencil and paper to work along!

The equations above show for any value of , the function is larger than or equal to the specific case when we plug in . Therefore, has to be the unique minimizer of that equation.

At this stage, we haven’t utilized the parent’s heights in our analysis. The initial step in examining this type of data is to construct a scatter plot of child heights against parent heights. Here we employ ggplot, but the plot has several shortcomings.

ggplot(galton, aes(x = parent, y = child)) + geom\_point()



Notably, there’s over-plotting due to numerous parent-child pairs sharing the same x, y values. To address this, we provide an improved plot where the point size reflects the number of parent-child combinations at a specific x, y location. Additionally, color indicates frequency, with lighter colors representing higher frequencies.

y <- galton$child - mean(galton$child)  
x <- galton$parent - mean(galton$parent)  
freqData <- as.data.frame(table(x, y))  
names(freqData) <- c("child", "parent", "freq")  
freqData$child <- as.numeric(as.character(freqData$child))  
freqData$parent <- as.numeric(as.character(freqData$parent))  
myPlot <- function(beta){  
 g <- ggplot(filter(freqData, freq > 0), aes(x = parent, y = child))  
 g <- g + scale\_size(range = c(2, 20), guide = "none" )  
 g <- g + geom\_point(colour="grey50", aes(size = freq+20, show\_guide = FALSE))  
 g <- g + geom\_point(aes(colour=freq, size = freq))  
 g <- g + scale\_colour\_gradient(low = "lightblue", high="white")   
 g <- g + geom\_abline(intercept = 0, slope = beta, size = 3)  
 mse <- mean( (y - beta \* x) ^2 )  
 g <- g + ggtitle(paste("beta = ", beta, "mse = ", round(mse, 3)))  
 g  
}

In order to find the best line, all we have to find is the slope. Well, here’s how we could potentially do that. We would want to find the slope beta that minimizes the sum of the squared distances between the observed data points the and the fitted data points on the line, . We’ll square that distance and add them up and this is directly analogous to finding the least squares mean. This is sort of using the origin as a pivot point and picking the line that minimizes the sum of the squared vertical distances between the points and the line. Notice that there is a point in regression to the origin is useful for explaining things, because we only have one parameter, the slope and we don’t have two parameters, the slope and the intercept. But it’s generally bad practice to force regression lines through the point (0, 0). So, an easy way around this is to subtract the mean from the parent’s heights and the mean from the child’s heights, so that the zero, zero point is right in the middle of the data and that will make this solution a little bit more palatable.

We can find the slope of the line very quickly in R using the lm function. The lm function stands for linear model. We’re going to regress the child’s height on the parent’s height. We’re going to subtract the mean from the child’s height and the mean from the parent’s height, to make sure line is going through the origin. Doing so will give us a line that has slope of 0.646.

lm(I(child - mean(child))~ I(parent - mean(parent)) - 1, data = galton)

##   
## Call:  
## lm(formula = I(child - mean(child)) ~ I(parent - mean(parent)) -   
## 1, data = galton)  
##   
## Coefficients:  
## I(parent - mean(parent))   
## 0.6463

Now what we’re going to do in subsequent sections is to talk about how we get these values? What is the motivation behind it and all the things we can do with this fitted line, we’re going to spend maybe the next several sections talking about this. You have actually learned a lot of material in this very first part, well done!

## 1.3 Linear least squares

Ordinary least squares (OLS) is the workhorse of statistics. It gives a way of taking complicated outcomes and explaining behavior (such as trends) using linearity. The simplest application of OLS is fitting a line through some data. In the next few sections, we cover the basics of linear least squares. We start with defining our notation. These are things you probably already saw in the prerequisite for this course in a Statistical Inference course in Data Science Specialization. However, because they’re so fundamental to regression, we’re going to cover them again, so they’re fresh in our minds. We will try to minimize the amount of mathematics that’s required for this class. Throughout the course we will neither require calculus nor linear algebra. And when it does get a little bit more mathematical, we will let you know when you can skip over those sections.

We might write to describe data points. As an example, consider the data set , where , , and in this case is 3. There’s nothing in particular about the letter . We could have just as easily described to . The last bit of notation that’s important, is we’re typically going to use Greek letters for things we don’t know, such as for a population mean and we’ll use non Greek letters or regular letters to denote things that we can observe. So, is something we can observe. is something we can’t observe and would like to estimate. We can define the empirical mean as

Notice if we subtract the mean from data points, we get data that has mean 0. That is, if we define The mean of the is 0.

* This process is called “centering” the random variables. Recall from the previous section that the mean is the least squares solution for minimizing .

Since we talked about means, let’s talk about variances. The variances is usually denoted by . It’s defined as

This is nothing other than basically the average squared deviation of the observations around the mean. The empirical standard deviation is defined as . Notice that the standard deviation has the same units as the data. It’s nice to work with standard deviations because the variance is expressed in whatever units has squared, whereas the standard deviation is just expressed in the normal units of . Another interesting fact related to standard deviation is scaling, so if we subtract a mean off from every observation, we get a resulting data set that has mean 0. If we divide every observation by the standard deviation, the resulting data set will have standard deviation 1. This is called **scaling** the data. If we take our original data now and subtract off , then take the resulting centered data and scale it by . We get a new data set, let’s call them .

This process of centering and then scaling is called *normalizing* the data. As an example, if something has a value 2 from normalized data, that means that the data point was 2 standard deviations larger than the mean. As its name would suggest, normalization is an attempt to make non-comparable data sets comparable.

The empirical covariance is the most central quantity in regression. Imagine we have two vectors, and , and they’re lined up. So might be the BMI and might be the blood pressure for subject . You could meaningfully do a scatter plot. Then we just define the covariance between X and Y as:

The correlation is defined as:

where and are the estimates of standard deviations for the observations and observations, respectively. In other words, the correlation is simply the covariance then standardized into a unitless quantity. So, the correlation is the covariance of and , which has units, basically units of X times units of Y.

Some facts about correlation: \* \* \* and only when the or observations fall perfectly on a positive or negative sloped line, respectively. \* measures the strength of the linear relationship between the and data, with stronger relationships as heads towards -1 or 1. \* implies no linear relationship.

## 1.4 Regression to the Mean

## 1.5 Practical R Exercises in swirl

## 1.6 Week 1 Quiz

# 2 Statistical linear regression models

# 3 Residuals

# 4 Inference in regression

# 5 For the project

# 6 Practical R Exercises in swirl

# 7 Week 2 Quiz

# 8 Multivariable regression

# 9 Multivariable regression tips and tricks

# 10 Adjustment

# 11 Residuals again

# 12 Model selection

# 13 Practical R Exercises in swirl

# 14 Week 3 Quiz

# 15 (OPTIONAL) Practice exercise in regression modeling

# 16 GLM

# 17 Logistic Regression

# 18 Poisson Regression

# 19 Hodgepodge

# 20 Practical R Exercises in swirl

# 21 Week 4 Quiz

# 22 Course Project

# 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) | [FirstName LastName](link%20to%20personal%20website) |
| Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | Delivered the course in some way - video or audio |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/), [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) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ 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 2024-01-18   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.5)   
## bookdown 0.24 2023-03-28 [1] Github (rstudio/bookdown@88bc4ea)   
## cachem 1.0.7 2023-02-24 [1] CRAN (R 4.0.2)   
## callr 3.5.0 2020-10-08 [1] RSPM (R 4.0.2)   
## cli 3.6.1 2023-03-23 [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 0.20 2023-01-17 [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 0.5.5 2023-03-23 [1] CRAN (R 4.0.2)   
## knitr 1.33 2023-03-28 [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 1.1.0 2023-03-14 [1] CRAN (R 4.0.2)   
## rmarkdown 2.10 2023-03-28 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.3 2022-04-02 [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 2023-03-28 [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 2023-03-28 [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

# 23 References