Computing for Big Data (BST-262)

Christine Choirat

2017-08-18

Contents

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

For now, you have to install the development versions of **bookdown** from Github:

```
devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need to install XeLaTeX.

Chapter 2

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2017) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

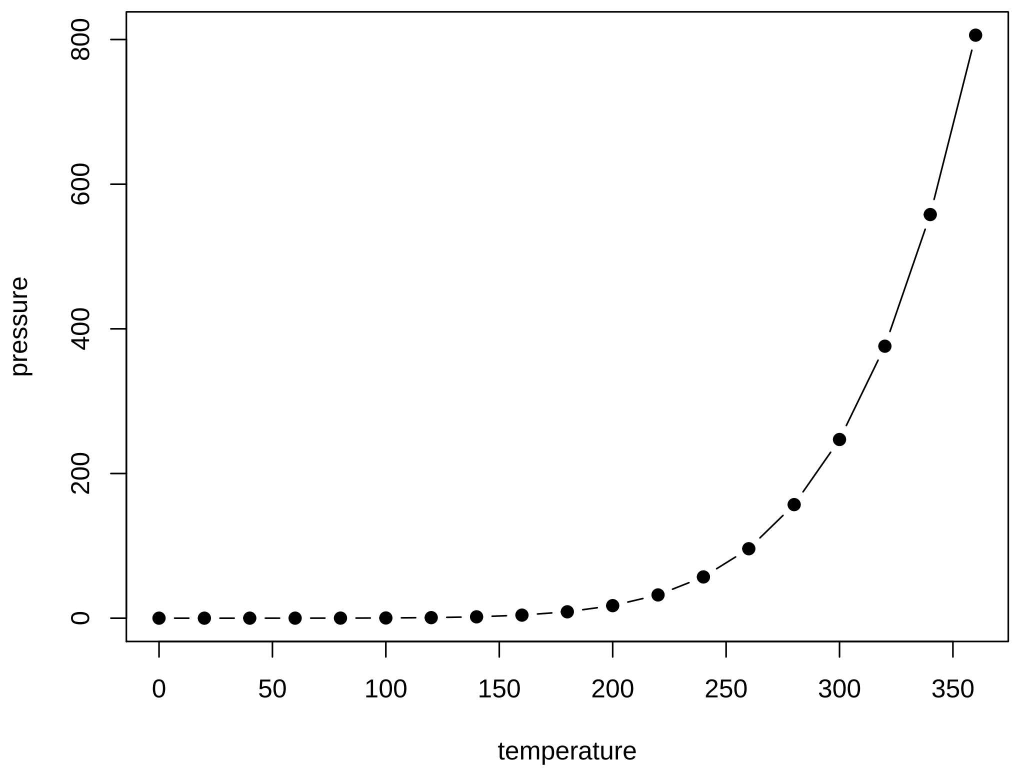

Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Basic tools

3.1 Command line tools

3.2 Git and GitHub

Chapter 4

Packages

4.1 Why?

- Organize your code
- Distribute your code
- Keep versions of your code

4.2 Structure

- Folder hierarchy
  - `NAMESPACE`: package import / export
  - `DESCRIPTION`: metadata
  - `R/`: R code
  - `man/`: object documentation (with short examples)
  - `tests/`
  - `data/`
  - `src/`: compiled code
  - `vignettes/`: manual-like documentation
  - `inst/`: installed files
  - `demo/`: longer examples
  - `exec`, `po`, `tools`

4.3 Building steps

- `R CMD build`
- `R CMD INSTALL`
- `R CMD check`

4.4 `R CMD build`

```
R CMD build --help
```

Build R packages from package sources in the directories specified by ‘pkgdirs’

4.5 R CMD INSTALL

```
R CMD INSTALL --help
```

Install the add-on packages specified by pkgs. The elements of pkgs can be relative or absolute paths to directories with the package sources, or to gzipped package ‘tar’ archives. The library tree to install to can be specified via ‘–library’. By default, packages are installed in the library tree rooted at the first directory in .libPaths() for an R session run in the current environment.

4.6 R CMD check

```
R CMD check --help
```

http://r-pkgs.had.co.nz/check.html

Check R packages from package sources, which can be directories or package ‘tar’ archives with extension ‘.tar.gz’, ‘.tar.bz2’, ‘.tar.xz’ or ‘.tgz’.

A variety of diagnostic checks on directory structure, index and control files are performed. The package is installed into the log directory and production of the package PDF manual is tested. All examples and tests provided by the package are tested to see if they run successfully. By default code in the vignettes is tested, as is re-building the vignette PDFs.

4.7 Building steps with devtools

- `devtools::build`
- `devtools::install`
- `devtools::check`
- and many others: `load_all`, `document`, `test`, `run_examples`, …

4.8 Creating an R package

4.8.1 utils::package.skeleton

```r
package.skeleton() # "in "fresh" session ("anRpackage")
package.skeleton("pkgname") # in "fresh" session

set.seed(02138)
f <- function(x, y) x+y
g <- function(x, y) x-y
d <- data.frame(a = 1, b = 2)
e <- rnorm(1000)
package.skeleton(list = c("f","g","d","e"), name = "pkgname")
```

4.8.2 `devtools::create`

```
devtools::create("path/to/package/pkgname")
```

4.9 Submitting to CRAN

http://r-pkgs.had.co.nz/release.html

4.10 Using GitHub

http://r-pkgs.had.co.nz/git.html

4.11 RStudio and GitHub integration (1 / 7)

4.12 RStudio and GitHub integration (2 / 7)

4.13 RStudio and GitHub integration (3 / 7)

4.14 RStudio and GitHub integration (4 / 7)

4.15 RStudio and GitHub integration (5 / 7)

4.16 Command line

```
git init
git add *
git commit -m "First commit"
git remote add origin git@github.com:harvard-P01/pkgtemplate.git
git push -u origin master
```

4.17 RStudio and GitHub integration (6 / 7)

4.18 RStudio and GitHub integration (7 / 7)

4.19 Installing from GitHub

```
devtools::install_github("harvard-P01/pkgtemplate")
```

```
devtools::install_github("harvard-P01/pkgtemplate",
                         build_vignettes = TRUE)
```

4.20 .gitignore (RStudio default)

```
.Rproj.user
.Rhistory
.RData
```

4.21 .gitignore (GitHub default)

```
# History files
.Rhistory
.Rapp.history

# Example code in package build process
*-Ex.R

# RStudio files
.Rproj.user/

# produced vignettes
vignettes/*.html
vignettes/*.pdf
```

4.22 RStudio projects

- `.Rproj` file extension, in our example `pkgtemplate.Rproj`
- A project has its own:
  - R session
  - .Rprofile (*e.g.*, to customize startup environment)
  - .Rhistory
- Default working directory is project directory
- Keeps track of project-specific recent files

4.23 Project options

```
Version: 1.0

RestoreWorkspace: Default
SaveWorkspace: Default
AlwaysSaveHistory: Default

EnableCodeIndexing: Yes
UseSpacesForTab: Yes
NumSpacesForTab: 2
Encoding: UTF-8

RnwWeave: knitr
```

```
LaTeX: pdfLaTeX

AutoAppendNewline: Yes
StripTrailingWhitespace: Yes

BuildType: Package
PackageUseDevtools: Yes
PackageInstallArgs: --no-multiarch --with-keep.source
```

4.24 Package documentation

- Functions and methods
- Vignettes
  - PDF
  - `knitr` (or `Sweave`)

4.25 Process example

Creating R Packages: A Tutorial (Friedrich Leisch, 2009)

- https://cran.r-project.org/doc/contrib/Leisch-CreatingPackages.pdf

4.26 Adding `linreg.R` in `R/` directory

```
linmodEst <- function(x, y) {
  ## compute QR-decomposition of x
  qx <- qr(x)
  ## compute (x'x)^(-1) x'y
  coef <- solve.qr(qx, y)
  ## degrees of freedom and standard deviation of residuals
  df <- nrow(x) - ncol(x)
  sigma2 <- sum((y - x %*% coef) ^ 2) / df
  ## compute sigma^2 * (x'x)^-1
  vcov <- sigma2 * chol2inv(qx$qr)
  colnames(vcov) <- rownames(vcov) <- colnames(x)
  list(
    coefficients = coef,
    vcov = vcov,
    sigma = sqrt(sigma2),
    df = df
  )
}
```

4.27 Running our function

```
data(cats, package = "MASS")
linmodEst(cbind(1, cats$Bwt), cats$Hwt)
```

```
## $coefficients
## [1] -0.3566624  4.0340627
##
## $vcov
##             [,1]        [,2]
## [1,]  0.4792475 -0.17058197
## [2,] -0.1705820  0.06263081
##
## $sigma
## [1] 1.452373
##
## $df
## [1] 142
```

4.28 And compare with `lm` (1 / 2)

```
lm1 <- lm(Hwt ~ Bwt, data=cats)
lm1
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, data = cats)
##
## Coefficients:
## (Intercept)          Bwt
##     -0.3567       4.0341
```

```
coef(lm1)
```

```
## (Intercept)         Bwt
##  -0.3566624   4.0340627
```

4.29 And compare with `lm` (2 / 2)

```
vcov(lm1)
```

```
##             (Intercept)         Bwt
## (Intercept)   0.4792475 -0.17058197
## Bwt          -0.1705820  0.06263081
```

```
summary(lm1)$sigma
```

```
## [1] 1.452373
```

4.30 Adding ROxygen2 documentation

```
#' Linear regression
#'
```

```
#' Runs an OLS regression not unlike \code{\link{lm}}
#'
#' @param y response vector (1 x n)
#' @param X covariate matrix (p x n) with no intercept
#'
#' @return A list with 4 elements: coefficients, vcov, sigma, df
#'
#' @examples
#' data(mtcars)
#' X <- as.matrix(mtcars[, c("cyl", "disp", "hp")])
#' y <- mtcars[, "mpg"]
#' linreg(y, X)
#'
#' @export
#'
linmodEst <- function(x, y) {
  ## compute QR-decomposition of x
  qx <- qr(x)
  ## compute (x'x)^(-1) x'y
  coef <- solve.qr(qx, y)
  ## degrees of freedom and standard deviation of residuals
  df <- nrow(x) - ncol(x)
  sigma2 <- sum((y - x %*% coef) ^ 2) / df
  ## compute sigma^2 * (x'x)^-1
  vcov <- sigma2 * chol2inv(qx$qr)
  colnames(vcov) <- rownames(vcov) <- colnames(x)
  list(
    coefficients = coef,
    vcov = vcov,
    sigma = sqrt(sigma2),
    df = df
  )
}
```

4.31 Configure Build Tools

4.32 `man/linmodEst.Rd`

```
% Generated by roxygen2 (4.1.1): do not edit by hand
% Please edit documentation in R/linmodEst.R
\name{linmodEst}
\alias{linmodEst}
\title{Linear regression}
\usage{
linmodEst(x, y)
}
\arguments{
\item{y}{response vector (1 x n)}

\item{X}{covariate matrix (p x n) with no intercept}
}
```

```
\value{
A list with 4 elements: coefficients, vcov, sigma, df
}
\description{
Runs an OLS regression not unlike \code{\link{lm}}
}
\examples{
data(mtcars)
X <- as.matrix(mtcars[, c("cyl", "disp", "hp")])
y <- mtcars[, "mpg"]
linmodEst(y, X)
}
```

4.33 Formatted output

4.34 DESCRIPTION

```
Package: pkgtemplate
Type: Package
Title: What the Package Does (Title Case)
Version: 0.1
Date: 2015-10-24
Author: Who wrote it
Maintainer: Who to complain to <yourfault@somewhere.net>
Description: More about what it does (maybe more than one line)
License: What license is it under?
LazyData: TRUE
```

4.35 NAMESPACE

export's automatically generated when parsing ROxygens2 snippets

```
export(linmodEst)
```

4.36 S3 basics

```
hello <- function() {
 s <- "Hello World!"
 class(s) <- "hi"
 return(s)
}

hello()
```

```
## [1] "Hello World!"
## attr(,"class")
## [1] "hi"
```

4.37 S3 basics

```
print.hi <- function(...) {
  print("Surprise!")
}

hello()
```

```
## [1] "Surprise!"
```

4.38 S3 and S4 generics

```
linmod <- function(x, ...)
  UseMethod("linmod")
```

```
linmod.default <- function(x, y, ...) {
  x <- as.matrix(x)
  y <- as.numeric(y)
  est <- linmodEst(x, y)
  est$fitted.values <- as.vector(x %*% est$coefficients)
  est$residuals <- y - est$fitted.values
  est$call <- match.call()
  class(est) <- "linmod"
  return(est)
}
```

4.39 print

```
print.linmod <- function(x, ...) {
  cat("Call:\n")
  print(x$call)
  cat("\nCoefficients:\n")
  print(x$coefficients)
}
```

4.40 print

```
x <- cbind(Const = 1, Bwt = cats$Bwt)
y <- cats$Hw
mod1 <- linmod(x, y)
mod1
```

```
## Call:
## linmod.default(x = x, y = y)
##
## Coefficients:
##      Const        Bwt
## -0.3566624  4.0340627
```

4.41 Other methods

- `summary.linmod`
- `print.summary.linmod`
- `predict.linmod`
- `plot.linmod`
- `coef.linmod`, `vcov.linmod`, …

4.42 Formulas and model frames

```
linmod.formula <- function(formula, data = list(), ...) {
  mf <- model.frame(formula = formula, data = data)
  x <- model.matrix(attr(mf, "terms"), data = mf)
  y <- model.response(mf)
  est <- linmod.default(x, y, ...)
  est$call <- match.call()
  est$formula <- formula
  return(est)
}
```

4.43 Unit tests and `testthat`

http://r-pkgs.had.co.nz/tests.html

In package directory:

```
devtools::use_testthat()
```

pre-populates `test/testthat/`

Test files should start with `test` to be processed.

4.44 `test_coef.R`

```
data(cats, package = "MASS")
l1 <- linmod(Hwt ~ Bwt * Sex, data = cats)
l2 <- lm(Hwt ~ Bwt * Sex, data = cats)

test_that("same estimated coefficients as lm function", {
  expect_equal(l1$coefficients, l2$coefficients)
})
```

```
==> devtools::test()

Loading pkgtemplate
Loading required package: testthat
Testing pkgtemplate
.
Woot!
```

4.45 Vignettes

http://r-pkgs.had.co.nz/vignettes.html

```
devtools::use_vignette("linmod")
```

https://github.com/harvard-P01/pkgtemplate/blob/master/vignettes/linmod.Rmd

Chapter 5

Optimization

In this Chapter, we are going to see how to measure and improve code performance

5.1 Measuring performance

5.1.1 Profiling

5.1.2 Benchmarking

5.2 Improving performance

5.2.1 Introduction to C/C++

5.2.2 Rcpp

Chapter 6

Databases

6.1 Overview

6.2 SQL

6.3 noSQL

6.4 R interfaces

Chapter 7

Big data

We have finished a nice book.

Chapter 8

Visualization

We have finished a nice book.

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2017). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.4.1.