

what should you do next?



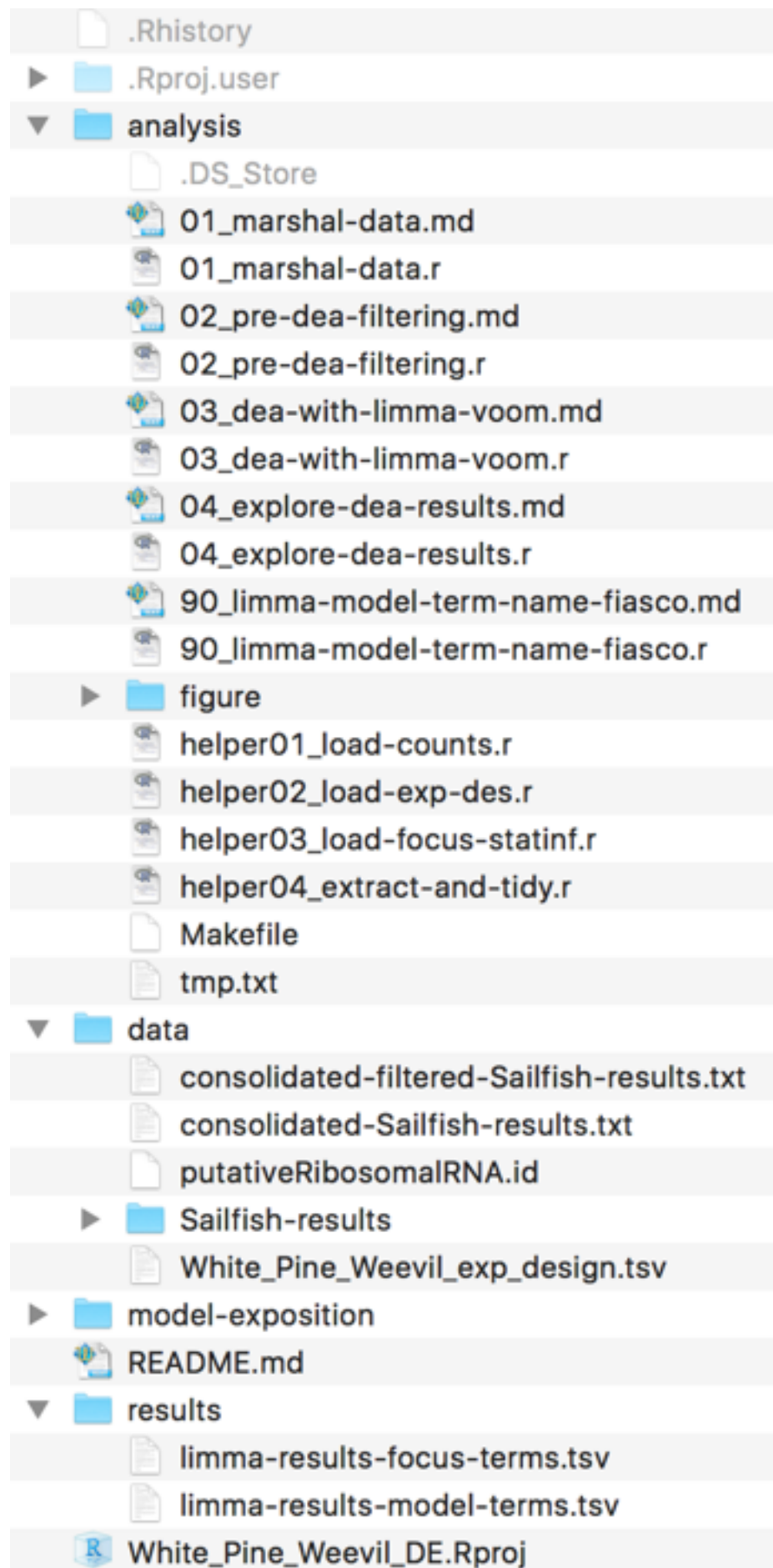
 [@JennyBryan](https://twitter.com/JennyBryan)

 [@jennybc](https://github.com/jennybc)



 [@STAT545](https://twitter.com/STAT545)

 <http://stat545.com>



Data, code, figs, results
living in harmony in one
Big Happy Directory?

NOT SUSTAINABLE

How to deal with paths
and working directory in
real world projects with
subdirectories?



Jenny Bryan @JennyBryan · 5 Dec 2015

.@STAT545 students running each other's #rstats pipelines. I know no better cure for:

```
setwd("C:\\Users\\jenny\\path\\that\\only\\I\\have")
```



Jenny Bryan @JennyBryan · 27 Jan 2015

In @STAT545, students run their peer's data analysis pipeline. That solved the `setwd()` misunderstanding and problem pretty fast. #rstats



~~setwd()~~

How can something that
feels so right be so wrong?

~~setwd()~~

Your collaborators,
future you,
your readers,
your VM,
your Docker container,
your cluster,

.....

will NOT share the same directory structure!



Timothée Poisot @tpoi · Apr 13

If the first line of your #rstats script is "**setwd(...**" I will come into your lab and SET YOUR COMPUTER ON FIRE.



13



33



Make your path handling portable!

Build paths relative to “project root”.

<http://poisotlab.io/2016/04/14/project-organization/>

rmarkdown / RStudio presents a special challenge

during interactive execution, working directory can be anything you want

“project root” is possibly the best choice and the path of least resistance

however, at render time, working directory = directory where the .R or .Rmd file lives

rprojroot

<https://cran.r-project.org/package=rprojroot>

ezknitr

<https://cran.r-project.org/package=ezknitr>

rprojroot

Robust, reliable and flexible paths to files below a project root. The 'root' of a project is defined as a directory that matches a certain criterion, e.g., it contains a certain regular file.

ezknitr

An extension of 'knitr' that adds flexibility in several ways. One common source of frustration with 'knitr' is that it assumes the directory where the source file lives should be the working directory, which is often not true. 'ezknitr' addresses this problem by giving you complete control over where all the inputs and outputs are, and adds several other convenient features to make rendering markdown/HTML documents easier.

`source(my_personal_functions.R)`

Do you do this at the
top of every script?

`read.csv(really_important_data.csv)`

Do you do this at the
top of every script?

Yes, you admit it?

It's time to make an R package!

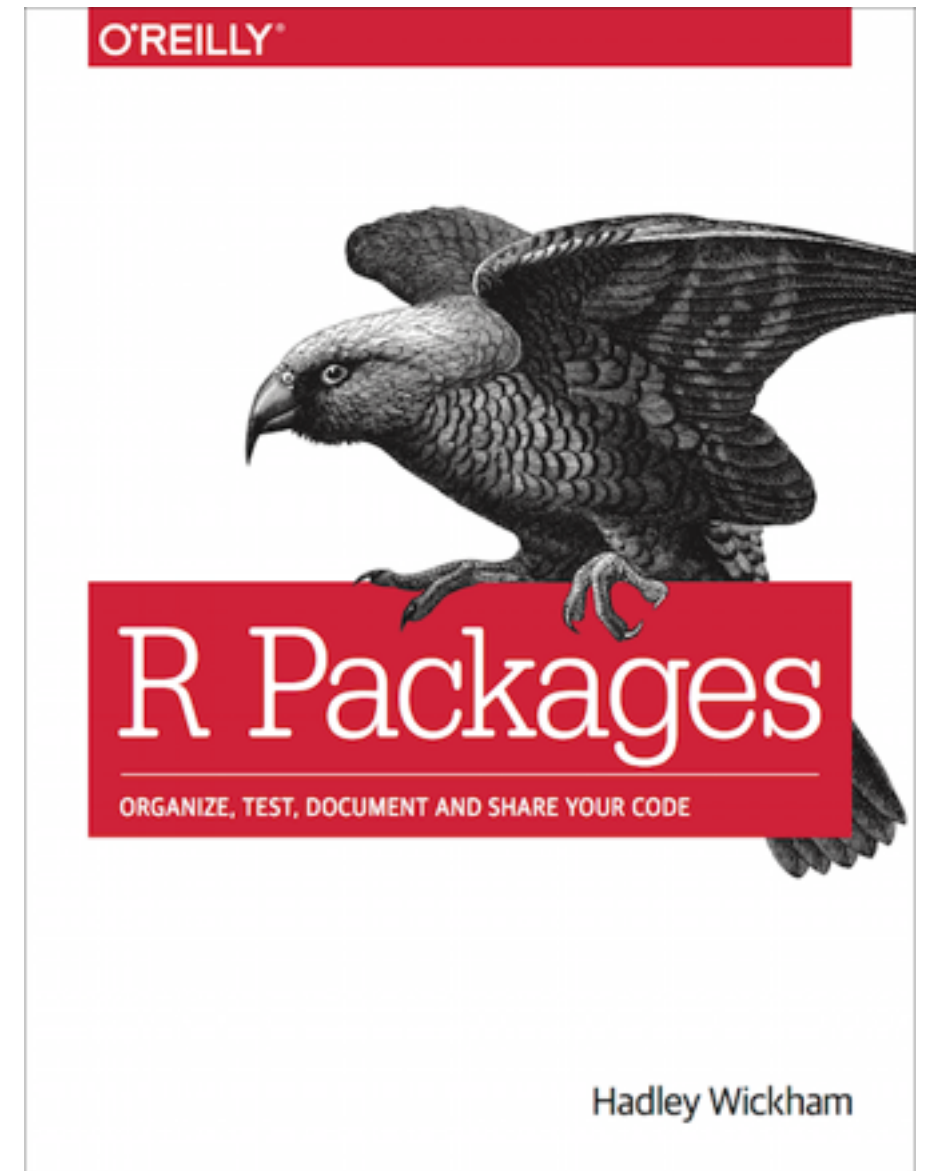
package = fundamental unit of R-ness

can bundle functions or data or both

<http://r-pkgs.had.co.nz>

Not So Standard Deviations

A statistics (etc.) blog by
Hilary Parker



<https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/>

http://stat545.com/packages00_index.html

STAT
545

Home

FAQ

Syllabus

Topics

People

Write your own R package

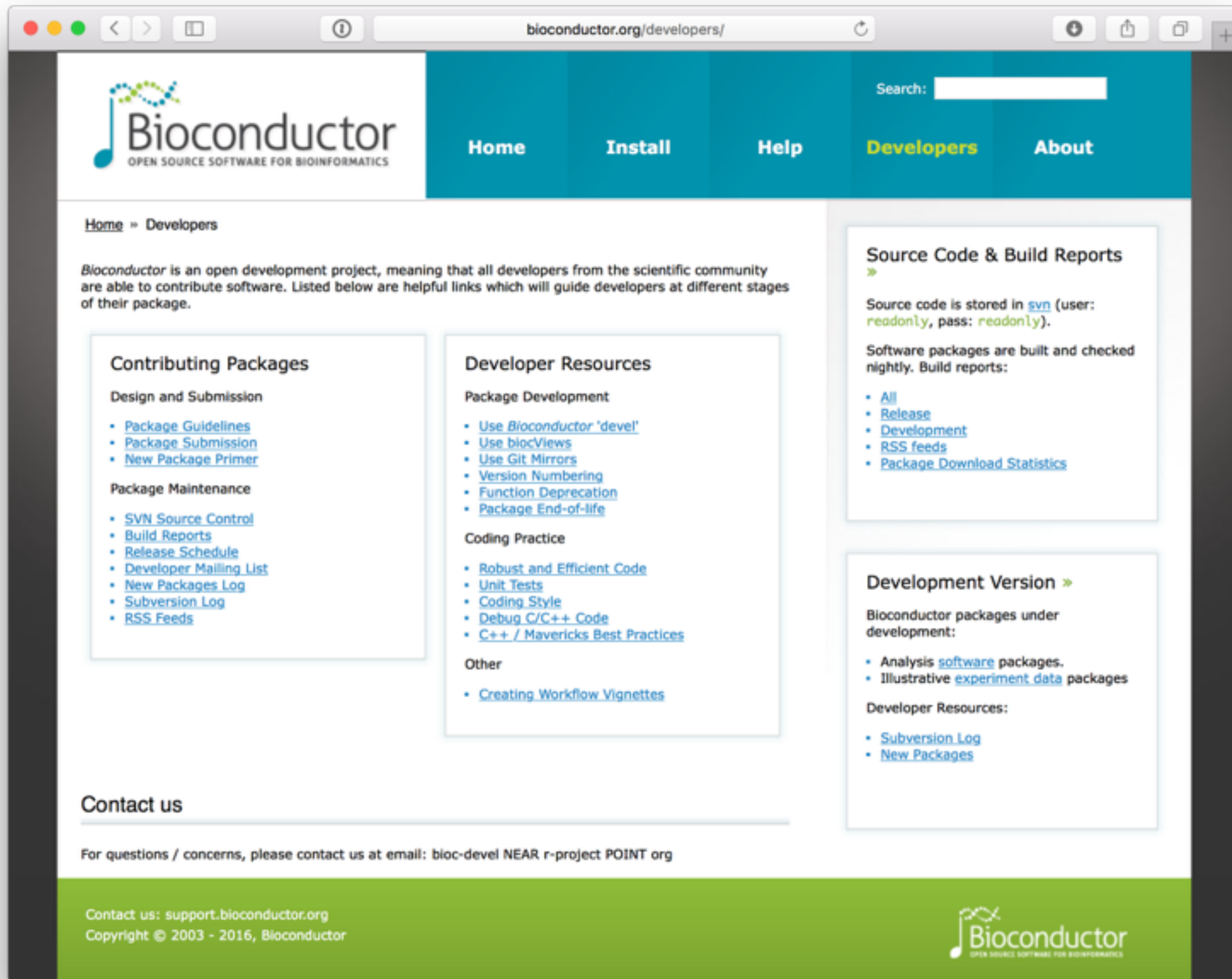
- [Setting the stage](#)
- [Prepare your system for package development](#)
- [Hands-on activity](#)
- [Resources](#)

<https://speakerdeck.com/jennybc/ubc-stat545-2015-writing-your-first-r-package>

devtools

<https://cran.r-project.org/package=devtools>





The image shows a browser window displaying the Bioconductor developers page. The browser's address bar shows the URL `bioconductor.org/developers/`. The page features a teal header with the Bioconductor logo and navigation links for Home, Install, Help, Developers, and About. A search bar is located in the top right corner. The main content area is divided into several sections: a breadcrumb trail (Home » Developers), an introductory paragraph about the open development project, and three primary columns of links. The left column, 'Contributing Packages', includes sub-sections for 'Design and Submission' and 'Package Maintenance'. The middle column, 'Developer Resources', includes 'Package Development' and 'Coding Practice'. The right column contains 'Source Code & Build Reports' and 'Development Version'. A 'Contact us' section is at the bottom left, and a footer with contact information and copyright is at the bottom.

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help **Developers** About

Home » Developers

Bioconductor is an open development project, meaning that all developers from the scientific community are able to contribute software. Listed below are helpful links which will guide developers at different stages of their package.

Contributing Packages

Design and Submission

- [Package Guidelines](#)
- [Package Submission](#)
- [New Package Primer](#)

Package Maintenance

- [SVN Source Control](#)
- [Build Reports](#)
- [Release Schedule](#)
- [Developer Mailing List](#)
- [New Packages Log](#)
- [Subversion Log](#)
- [RSS Feeds](#)

Developer Resources

Package Development

- [Use Bioconductor 'devel'](#)
- [Use biocViews](#)
- [Use Git Mirrors](#)
- [Version Numbering](#)
- [Function Deprecation](#)
- [Package End-of-life](#)

Coding Practice

- [Robust and Efficient Code](#)
- [Unit Tests](#)
- [Coding Style](#)
- [Debug C/C++ Code](#)
- [C++ / Mavericks Best Practices](#)

Other

- [Creating Workflow Vignettes](#)

Source Code & Build Reports

Source code is stored in [svn](#) (user: [readonly](#), pass: [readonly](#)).

Software packages are built and checked nightly. Build reports:

- [All](#)
- [Release](#)
- [Development](#)
- [RSS feeds](#)
- [Package Download Statistics](#)

Development Version

Bioconductor packages under development:

- Analysis [software](#) packages.
- Illustrative [experiment data](#) packages


Developer Resources:

- [Subversion Log](#)
- [New Packages](#)

Contact us

For questions / concerns, please contact us at email: bioc-devel@r-project.org

Contact us: support.bioconductor.org
Copyright © 2003 - 2016, Bioconductor



<http://bioconductor.org/developers/>

From earlier ...

Strong recommend: use the “tidyverse”

- tibble + dplyr + tidyr

Now adding to that

- the pipe operator `%>%`
- the purrr package
- use of nested tibbles, list-columns



% > %

```
foo_foo <- little_bunny()

bop_on(
  scoop_up(
    hop_through(foo_foo, forest),
    field_mouse
  ),
  head
)
```

```
# vs
```

```
foo_foo %>%
  hop_through(forest) %>%
  scoop_up(field_mouse) %>%
  bop_on(head)
```

from Hadley Wickham

dplyr + tidyr + purrr **for data wrangling and aggregation**

`filter()` and `select()` for targeting specific rows or variables

`mutate()` for creating or mutating variables

`group_by()` for creating conceptual groups of rows

`summarize()` for computing on groups

“shock and awe” re: nested nibbles, list-columns

dplyr for data wrangling and aggregation

Gapminder example:

I have found friends and family love to ask seemingly innocuous questions like, “which country experienced the sharpest 5-year drop in life expectancy?”. In fact, that is a totally natural question to ask. But if you are using a language that doesn’t know about data, it’s an incredibly annoying question to answer.

http://stat545.com/block010_dplyr-end-single-table.html

go to coding demo

dplyr + tidyr + purrr for data wrangling and aggregation

nest() for creating meta-observations from groups of rows (nested tibbles, with data in list-columns)

complicated things, e.g., models, can go in a list-column

mutate() + purrr::map() functions can be used to post-process models

```
mutate(new_var = map(old_var, fun))
```

goal is always to get back to a “normal” data frame


```
df %>% group_by() %>% nest()
```

is not the only way to get list-columns

I often make list-columns directly

then use this same pattern

```
mutate(new_var = map(old_var, fun))
```

and work my way back to a “normal” data frame

go to coding demo

More examples of purrr usage

<https://github.com/jennybc/send-email-with-r#readme>

<https://github.com/jennybc/analyze-github-stuff-with-r#readme>

<https://github.com/jennybc/manipulate-xml-with-purrr-dplyr-tidyr#readme>

dplyr + tidyr + purrr

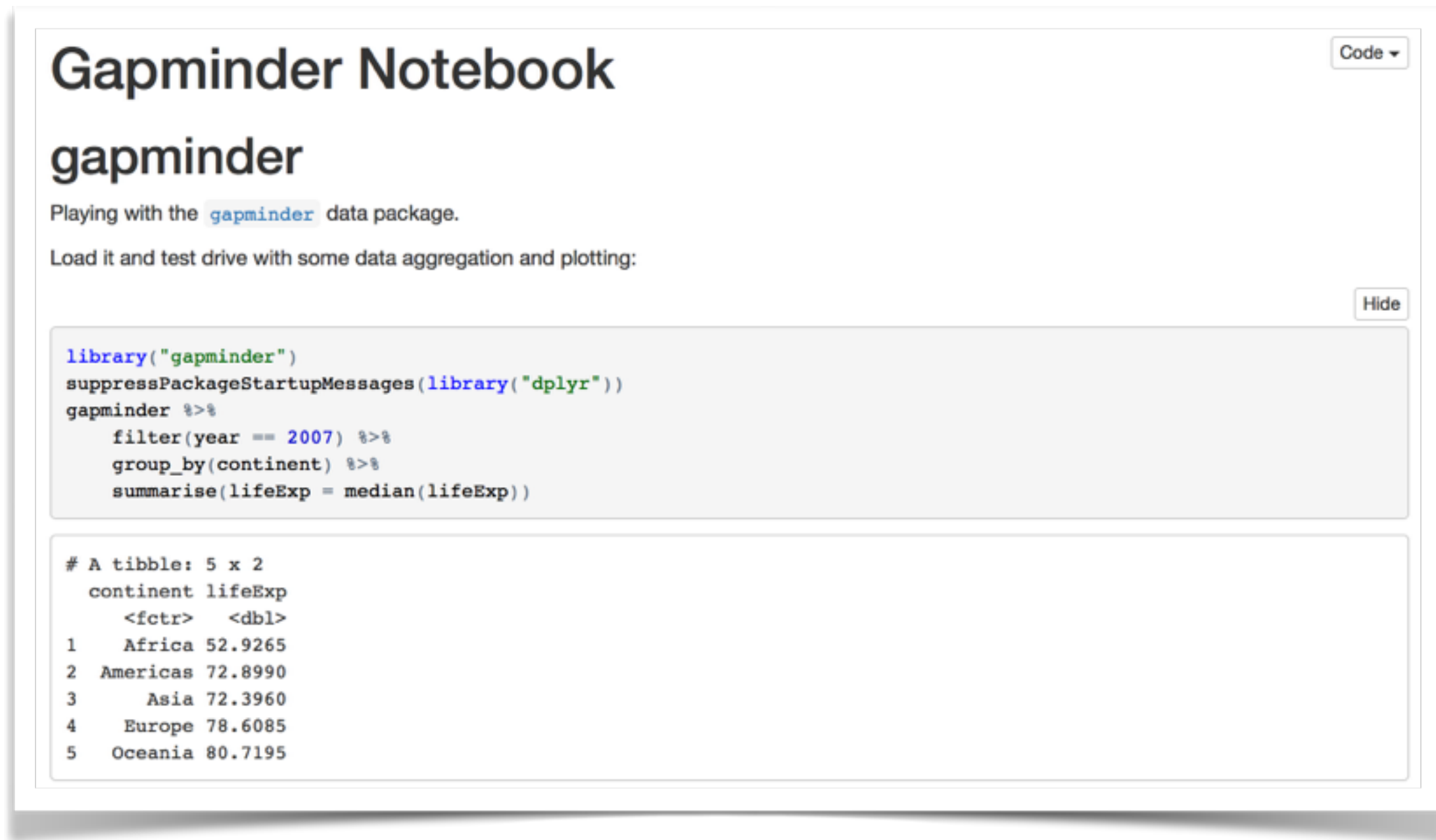
with heavy use of list-columns

is how I do all iterative tasks

i.e. has taken the place of

- **for loops** (ok I haven't used those in very long time)
- base "apply" family of functions
- **plyr** package

R Markdown Notebooks



The screenshot shows an R Markdown notebook interface. At the top, the title "Gapminder Notebook" is displayed in a large font, with a "Code" dropdown menu to its right. Below the title, the word "gapminder" is written in a slightly smaller font. A paragraph of text follows: "Playing with the `gapminder` data package. Load it and test drive with some data aggregation and plotting:". To the right of this text is a "Hide" button. Below the text is a code block containing R code for loading the `gapminder` package, suppressing startup messages, and using `dplyr` to filter for the year 2007, group by continent, and calculate the median life expectancy. The output of the code is a tibble with 5 rows and 2 columns: continent and lifeExp. The data is as follows:

```
library("gapminder")
suppressPackageStartupMessages(library("dplyr"))
gapminder %>%
  filter(year == 2007) %>%
  group_by(continent) %>%
  summarise(lifeExp = median(lifeExp))
```

```
# A tibble: 5 x 2
  continent lifeExp
  <fctr>     <dbl>
1 Africa    52.9265
2 Americas  72.8990
3 Asia      72.3960
4 Europe    78.6085
5 Oceania   80.7195
```

Talk from June 2016 useR! @ Stanford

<https://channel9.msdn.com/Events/useR-international-R-User-conference/useR2016/Notebooks-with-R-Markdown>

Documentation

http://rmarkdown.rstudio.com/r_notebook_format.html

go to coding demo



#rstats
#bioconductor

follow people
watch package repos



<https://support.bioconductor.org>

The screenshot shows the Bioconductor support forum interface. At the top, there is a blue header bar with the Bioconductor logo and navigation links: ASK QUESTION, LATEST (highlighted), NEWS, JOBS, TUTORIALS, and TAGS. Below the header, there are filters for Limit, Sort, and a search bar. The main content area displays a list of questions:

Limit	Sort	Search
1 vote	5 answers	50 views
0 votes	0 answers	10 views
0 votes	0 answers	16 views

Questions listed:

- Diffbind EdgeR and DESeq2 normalization**
normalized counts diffbind
written 17 hours ago by GFM • 10
- CHIPQC error: "if (nrow(pv\$binding) > 0) { : argument is of length zero"**
chipqc
written 2 hours ago by Danielle Denisko • 0
- dba.count error in DiffBind**
diffbind
written 3 hours ago by shohei.hori • 0