

A conversation on reproducibility and interactive notebooks with RMarkdown



Dana Miller

BEARS

January 3, 2019

Learning objectives

- What about “reproducibility”?
- What is Markdown?
- Hands on:
 - Create an Rmarkdown document and “knit” it
 - Working with Rmarkdown files and git + GitHub
- (As time allows) examples of other document types
 - Journal article templates (eg with “rticles” package)
 - Bookdown
 - Blogdown



Why reproducible research? Robustness

PHYSICS TODAY

22 Aug 2018 in **Research & Technology**

The war over supercooled water

How a hidden coding error fueled a seven-year dispute between two of condensed matter's top theorists.

“Limmer maintains that he and his mentor weren’t trying to hide anything. “I had and was very willing to share the code,” he says. What he didn’t have, he says, was the time or personnel to prepare the code in a form that could be useful to an outsider”

“Suddenly it made sense that the Berkeley researchers hadn’t seen a second liquid phase; they were effectively running their simulations at temperatures well above the critical point.”

<https://physicstoday.scitation.org/doi/10.1063/PT.6.1.20180822a/full/>

Why reproducible research? Records

Email from grad student to Berkeley professor in 2017:

“...The citation on the slide is [your report on a state government website].

I downloaded that PDF and do not see [important statistic] anywhere in it.
Table 3 seems to have [different value] , Table 15 has [related value] in California...

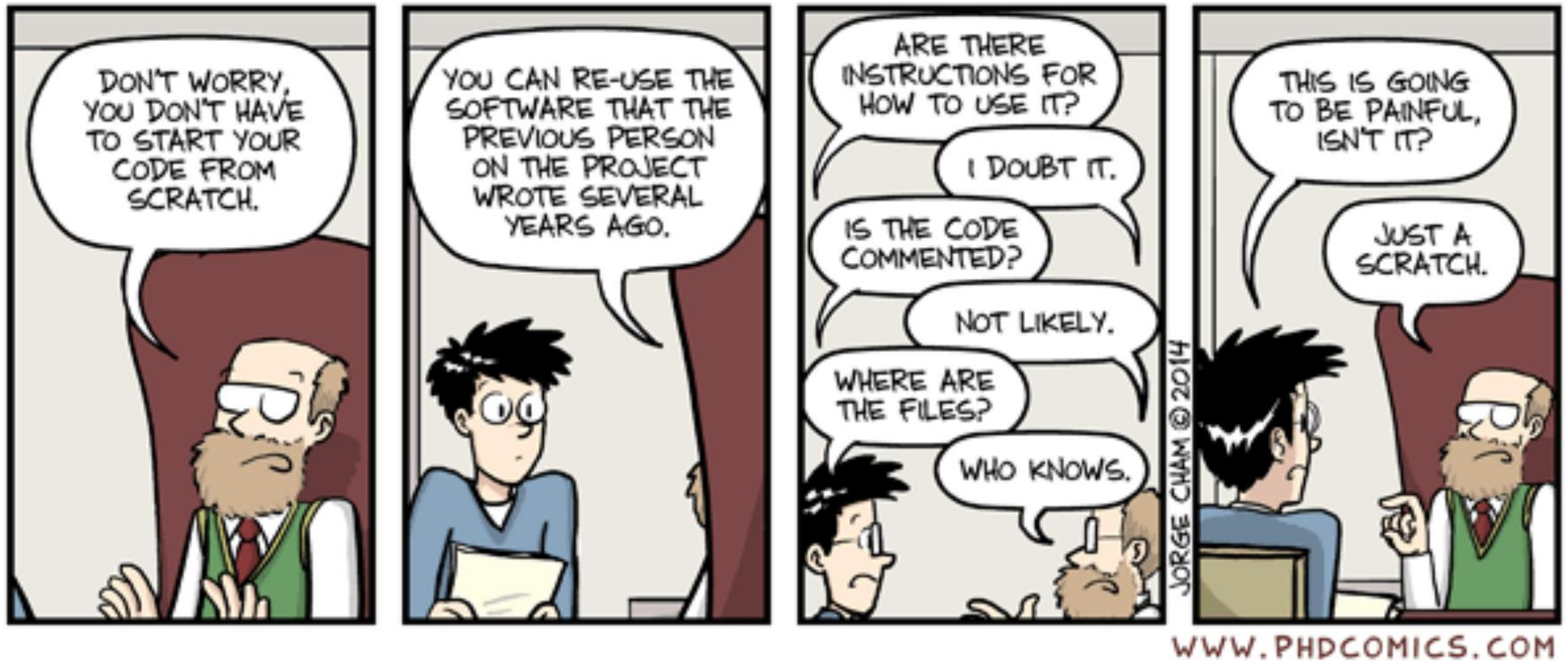
...The closest match that I've found is [other statistic] from Table 15...
Is it possible there was an error in transcription? If not, can you help me figure out what's going on here? What does [important statistic] on [slide] represent?”

Response from Berkeley professor:

“I don't remember the details. [Former student], do you? Thanks in advance.”

[Former student] never responded

Why reproducible research? Reusability



Why is this important?

- Show evidence of the correctness of our results
- Enable others to make use of our methods and results (including future you)
- Increasingly encouraged or required by funders and journals

<http://ropensci.github.io/reproducibility-guide/sections/introduction/>

<https://www.stat.berkeley.edu/~stark/Seminars/reproNE16.htm>



<https://cos.io/our-services/open-science-badges>

“reproducibility”

What does reproducibility mean? Three ideas to distinguish

1. *Reproducibility* (sometimes called repeatability or even ‘preproducibility’) [6]

- *Data, and/or statistical methods, code, software environment are available so data analysis can be repeated and get a similar result [1, 2, 3]*

2. *Replicability* (sometimes called reproducibility)

- *Another researcher is able to conduct the same experiment and get similar results*

3. *Correctness*

- *Is your analysis appropriate? Is your hypothesis correct?*
→ *Reproducible research can still be wrong! [4]*

1 - Peng *Science* 2011, DOI: 10.1126/science.1213847

2 - <http://ropensci.github.io/reproducibility-guide/sections/introduction/>

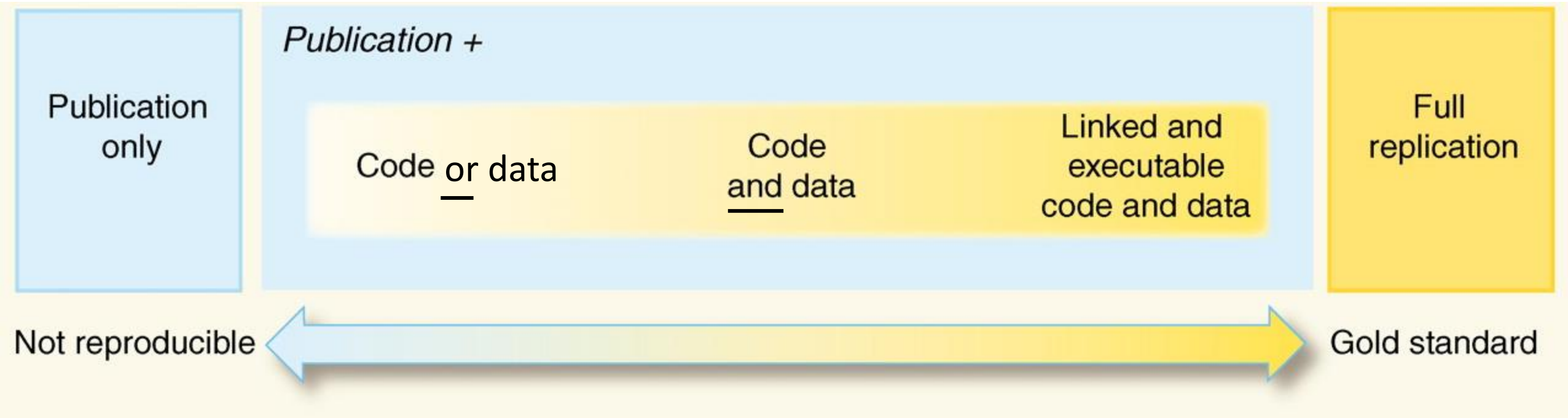
3 - <https://simplystatistics.org/2014/06/06/the-real-reason-reproducible-research-is-important/>, Peng 2014

4 - Reproducible research can still be wrong, Leek, Peng
PNAS 2015, DOI: 10.1073/pnas.1421412111

5- <https://www.nature.com/news/muddled-meanings-hamper-efforts-to-fix-reproducibility-crisis-1.20076#/b1>, citing <https://www.ascb.org/wp-content/uploads/2015/11/How-can-scientist-enhance-rigor.pdf> citing Schmidt, Stefan. (2009). Shall We Really Do It Again? The Powerful Concept of Replication Is Neglected in the Social Sciences. *Review of General Psychology*. 13. 90-100. 10.1037/a0015108.

6 – Barba 2018 <https://arxiv.org/pdf/1802.03311.pdf>

A spectrum of practice (depends on project)



Adopted from:
Reproducible Research in Computational Science
By Roger D. Peng
Science 02 Dec 2011 : 1226-1227
<http://science.sciencemag.org/content/334/6060/1226.full>

What are some examples
of good practices
we can start using now?

Data analysis

1. File organization

- Suggest one folder per project, eg:

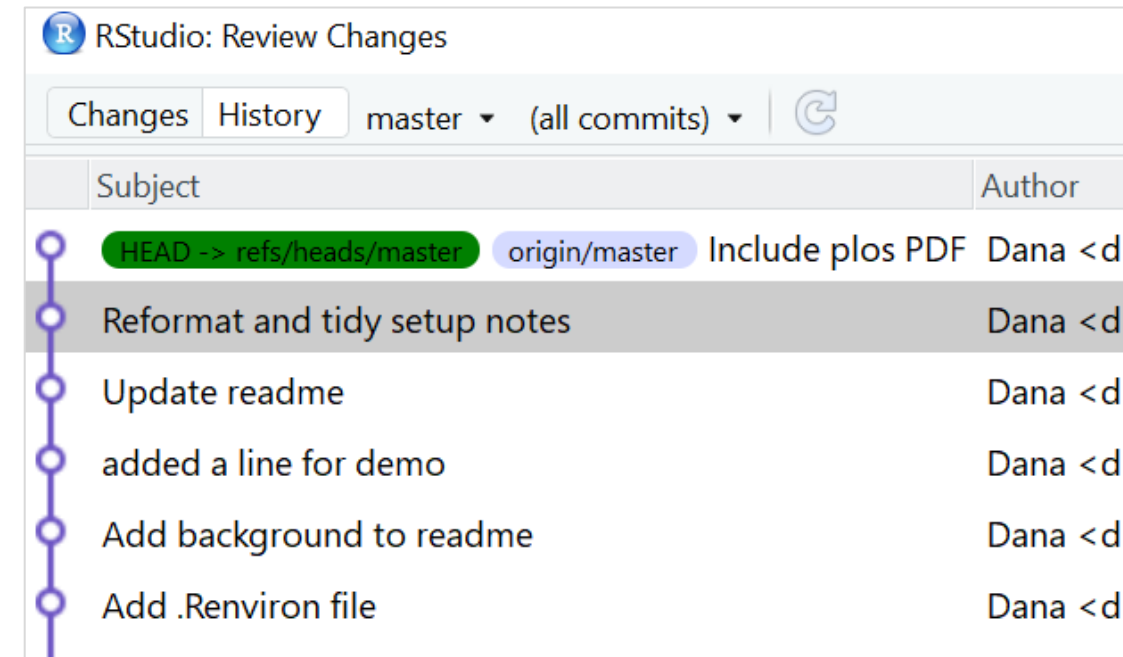
```
My_project
├── Data
│   └── raw_data.csv
├── Docs
│   └── Manuscript.doc
├── Results
│   └── Processed_results.csv
└── Scripts
    ├── clean_data.R
    └── fit_model_and_graph.R
```

Resources:

- [A Quick Guide to Organizing Computational Biology Projects](#)
- [Good enough practices in scientific computing](#)

2. Version control (for code)

- Like MS Word's Previous Versions → Restore, or unlimited undo for your R scripts



Resources:

- <http://happygitwithr.com/>
- [A quick introduction to version control with Git and GitHub](#)

Sharing + publication of results

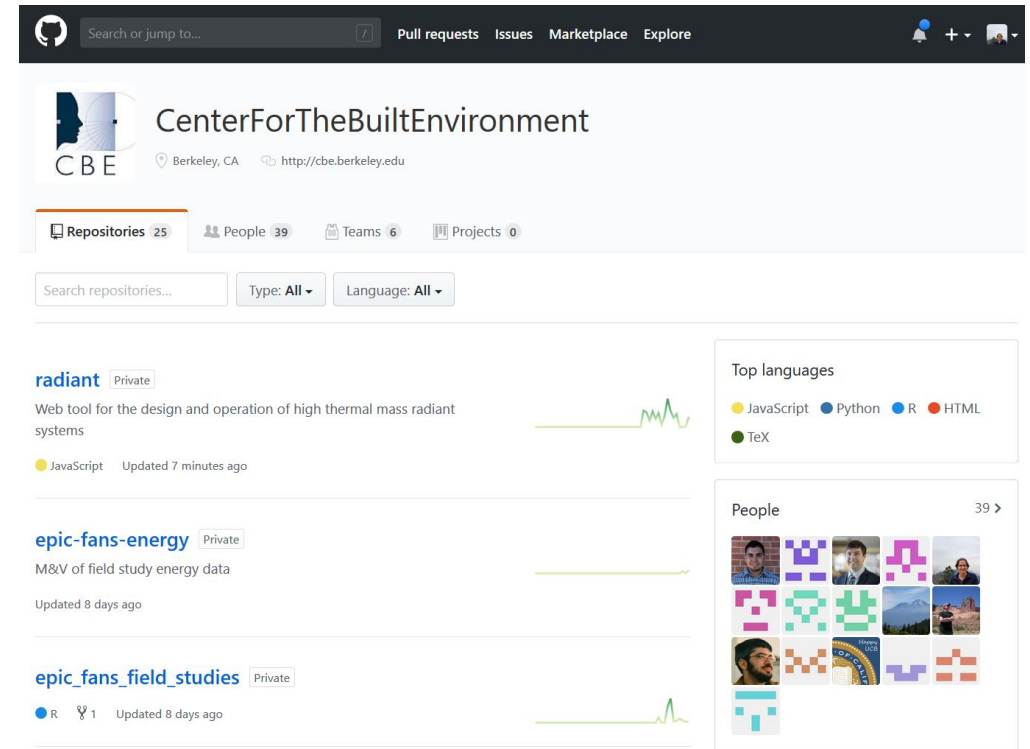
1. Data repositories

- UC system – [dash](#)
- Example: [ASHRAE Global Thermal Comfort Database II](#)

Resources:

- [Berkeley Research Data Management librarians!](#)
- Other repositories include Zenodo, Open Science Framework, DataDryad, Figshare

2. Code repositories



Resources:

- [Software Carpentry's Intro to Version Control for Novices](#)
- [A quick introduction to version control with Git and GitHub](#)

Two very common (and free) interactive notebooks:

Jupyter

Sampling from the generative model

In this notebook, we will use the generative model of the HDHP (Hierarchical Dirichlet-Hawkes Process) in order to sample ϵ . We will start with a predefined number of users, say 10, and we will attempt to model their behavior as they are posting questions on an online platform. For simplicity, our "vocabulary" will be dummy.

We start by importing all the libraries that will be required.

```
%matplotlib inline
import datetime
import string
import hdhp
import notebook_helpers
import seaborn as sns
```

Now, let us set some parameters for our model. These fall under two categories; the ones relevant to the content and then ϵ .

```
vocabulary = ['word' + str(i) for i in range(100)] # the "words" of our documents
doc_min_length = 5
doc_length = 10
words_per_pattern = 50

alpha_0 = (2.5, 0.75)
mu_0 = (2, 0.5)
```

```
overlap = notebook_helpers.compute_pattern_overlap(process)
sns.distplot(overlap, kde=True, norm_hist=True, axlabel='Content overlap')
```

Average overlap: 0.338826769742

```
<matplotlib.axes._subplots.AxesSubplot at 0x10de8ca50>
```



Other interactive notebooks:
Beaker (Two Sigma)
[Zeppelin](#) (Apache)
Databricks (Databricks)

R Markdown

Unit 3: Fisheries Collapse Module

This module will focus on understanding and replicating fisheries stock assessment data and fisheries collapse.

The Database

We will use data from the [RAM Legacy Stock Assessment Database](#)

First, load in the necessary libraries. Note that this time we need a package we haven't used before `readxl`. This package is useful for reading in .xls or .xlsx files. As always if you want more info on a package run `?readxl` after loading it.

```
library("tidyverse")
library("readxl")
library("scales") # For y-axis labels not in scientific notation - is there a better way to do this since 2012?
```

Reading in the tables

```
download.file("https://depts.washington.edu/ramlegac/wordpress/databaseVersions/RLSADB_v3.0_(assessment_data_only)_excel.zip",
              "ramlegacy.zip")
path <- unzip("ramlegacy.zip") # unzip the .xls files
sheets <- readxl::excel_sheets(path) # use the readxl package to identify sheet names
```

```
# purrr::map is the tidyverse version of lapply
ram <- lapply(sheets, readxl::read_excel, path = path) # read the data from all 3 sheets into a list
names(ram) <- sheets # give the list of datatables their assigned sheet names
```

```
## check the names
names(ram)
```

```
## [1] "area"                "assessment"
## [3] "assessmethod"        "assessor"
## [5] "biometrics"          "bioparams"
## [7] "bioparams_ids_views" "bioparams_units_views"
## [9] "bioparams_values_views" "management"
## [11] "stock"               "taxonomy"
## [13] "timeseries"          "timeseries_ids_views"
## [15] "timeseries_units_views" "timeseries_values_views"
## [17] "tsmetrics"
```

```
## check our data
```

Similarities of Jupyter and R notebooks – compatible with best practices

Sampling from the generative model

In this notebook, we will use the generative model of the HDHP (Hierarchical Dirichlet-Hawkes Process) in order to sample ϵ . We will start with a predefined number of users, say 10, and we will attempt to model their behavior as they are posting questions on a question-answer platform. For simplicity, our "vocabulary" will be drawn from a predefined set of words.

We start by importing all the libraries that will be required.

```
import matplotlib inline
import datetime
import string
import hdhp
import notebook_helpers
import seaborn as sns
```

```
vocabularies = ['word' + str(i) for i in range(100)] # the "words" of our documents
doc_min_length = 5
doc_length = 10
words_per_pattern = 10
```

```
alpha_0 = (2.5, 0.75)
mu_0 = (2, 0.5)
omega = 3.5
```

```
num_patterns = 10
```

```
process = hdhp.HDHPProcess(num_patterns=num_patterns, mu_0=mu_0, vocabulary=vocabularies,
                           omega=omega, words_per_pattern=words_per_pattern,
                           random_state=12)
```

```
overlap = notebook_helpers.compute_pattern_overlap(process)
sns.distplot(overlap, kde=True, norm_hist=True, axlabel='Content overlap')
```

Average overlap: 0.338826769742

```
<matplotlib.axes._subplots.AxesSubplot at 0x10de8ca50>
```

Unit 3: Fisheries Collapse Module

This module will focus on understanding and replicating fisheries stock assessment data and fisheries collapse.

The Database

First, load in the necessary libraries. Note that this time we need a package we haven't used before: `readxl`. This package is useful for reading in .xls or .xlsx files. As always if you want more info on a package run `?readxl` after loading it.

```
library("readxl")
library("scales") # For y-axis labels not in scientific notation - is there a better way to do this since
```

Reading in the tables

```
path <- "https://www.fishbase.org/ramlegac/wordpress/databaseVersions/RLSADB_v3.0_(assessment_data_only)_excel.zip"
path <- unzip("ramlegac.zip") # unzip the files
# purrr::map is the tidyverse version of lapply
ram <- lapply(sheets, readxl::read_excel, path = path) # read the data from all 3 sheets into a list
names(ram) # check the names of the sheets
# check the names of the sheets
names(ram)
```

```
## [3] "assessment"
## [5] "assessment"
## [7] "biometrics"
## [9] "biometrics"
## [11] "stock"
## [13] "timeseries"
## [15] "timeseries_units_views"
## [17] "tsmetrics"
```

Exercise 1: Investigating the North-Atlantic Cod

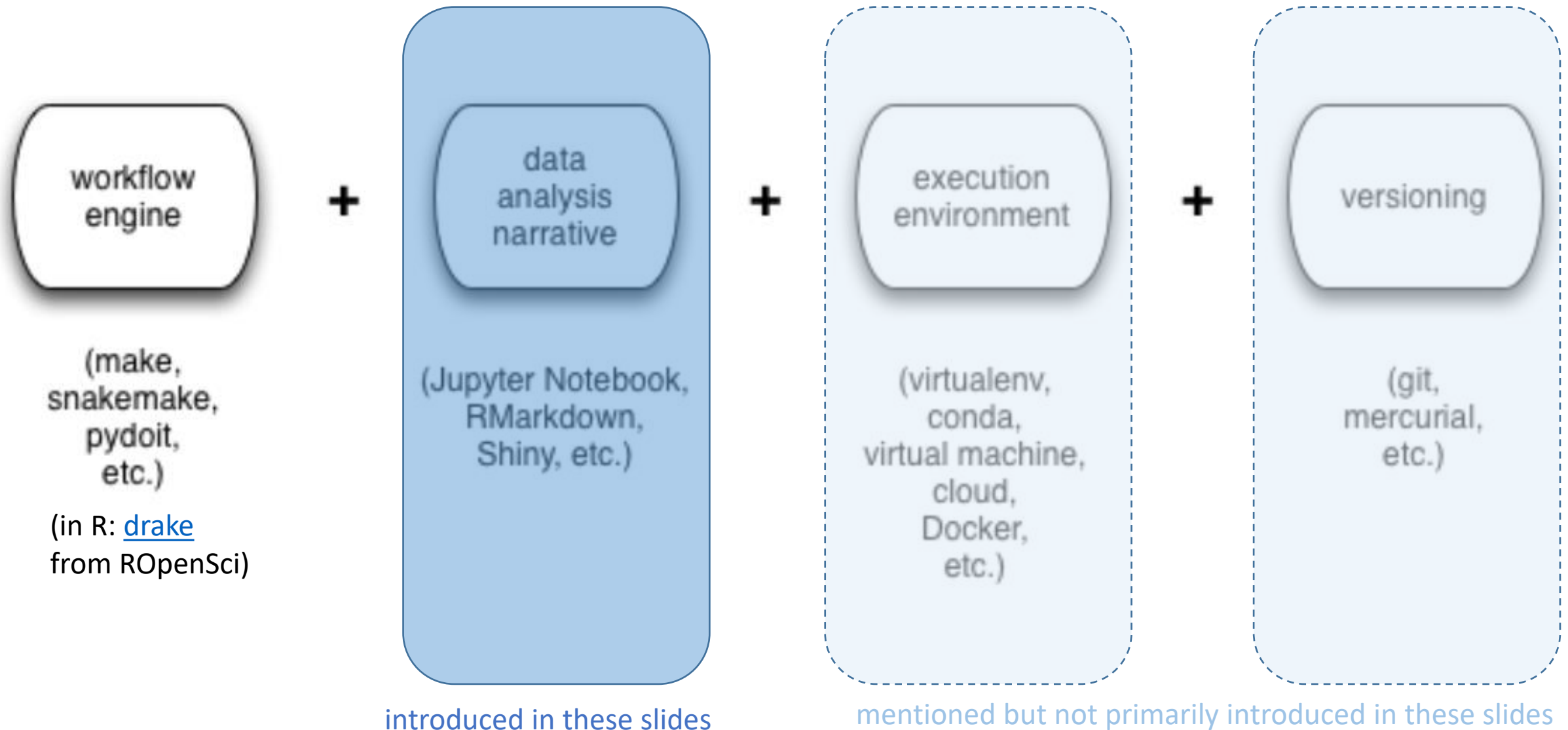
First, we seek to replicate the following figure from the Millennium Ecosystem Assessment Project using the RAM data.

Fish landings in tons

1900-2000

All but data visualization based on Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. (2014) Best Practices for Scientific Computing. PLoS Biol 12(1): e1001745. <https://doi.org/10.1371/journal.pbio.1001745>

Interactive notebooks are **one part** of a toolset for reproducibility



Adapted from slide by C Titus Brown, UC Davis

<https://2016-oslo-repeatability.readthedocs.io/en/latest/overview-and-agenda.html>

Background— what is literate programming?



https://en.wikipedia.org/wiki/Donald_Knuth#/media/File:KnuthAtOpenContentAlliance.jpg

D. E. Knuth; *Literate Programming*, *The Computer Journal*, Volume 27, Issue 2, 1 January 1984, Pages 97–111,
<https://doi.org/10.1093/comjnl/27.2.97>

“Instead of imagining that our main task is to instruct a *computer* what to do, let us concentrate rather on explaining to *human beings* what we want a computer to do.”

- Donald Knuth (1984)

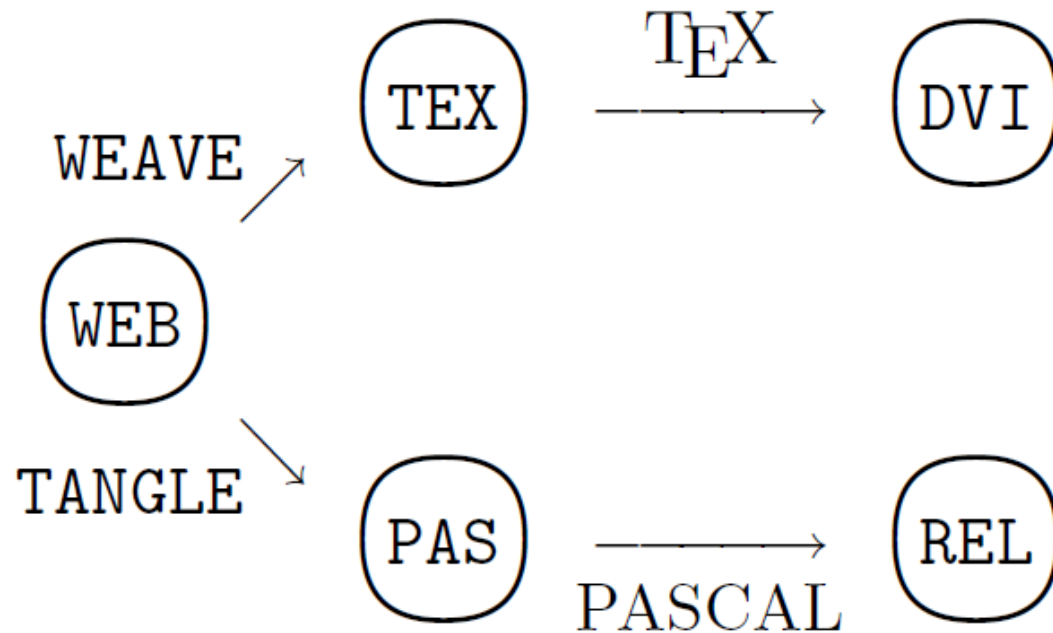
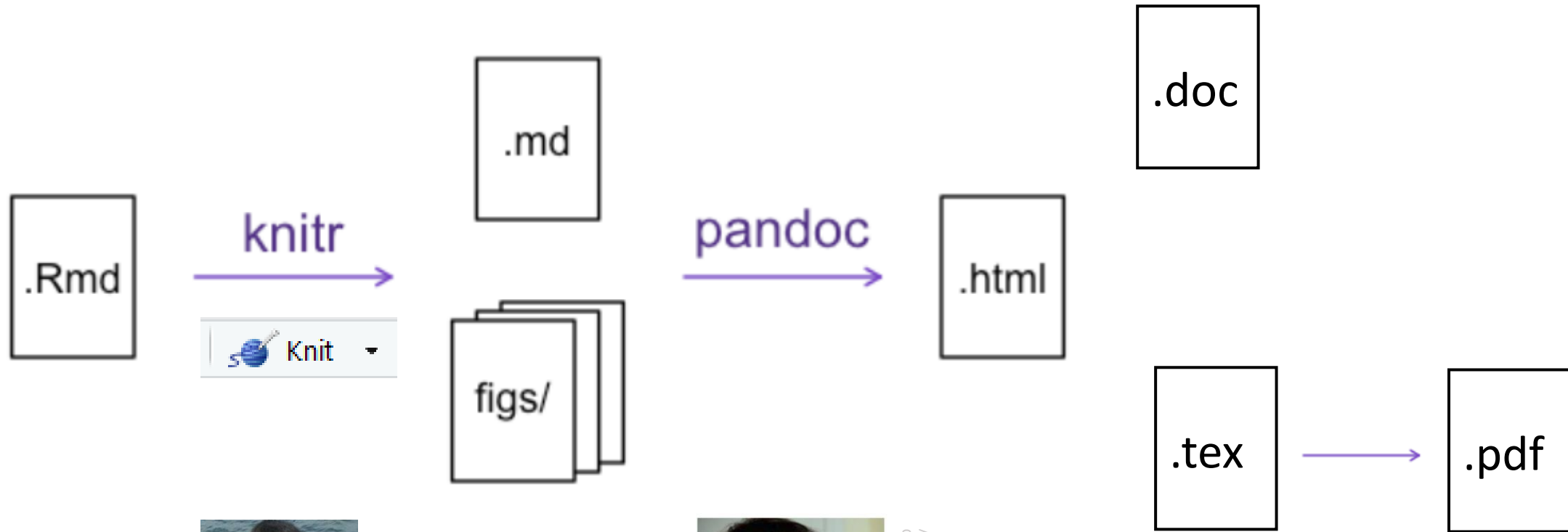


Figure 1. Dual usage of a WEB file.

What is markdown?

Text using Markdown syntax	Corresponding HTML produced by a Markdown processor	Text viewed in a browser
Heading ===== ## Sub-heading Paragraphs are separated by a blank line. Two spaces at the end of a line produces a line break. Text attributes <i>_italic_</i> , **bold** , <code>`monospace`</code> . Horizontal rule: --- Bullet list: * apples * oranges * pears	<h1>Heading</h1> <h2>Sub-heading</h2> <p>Paragraphs are separated by a blank line.</p> <p>Two spaces at the end of a line produces a line break.</p> <p>Text attributes italic, bold, <code>monospace</code>.</p> <p>Horizontal rule:</p> <hr /> <p>Bullet list:</p> apples oranges	Heading [edit] Sub-heading [edit] Paragraphs are separated by a blank line. Two spaces at the end of a line produces a line break. Text attributes <i>italic</i> , bold , <code>monospace</code> . Horizontal rule: <hr/> Bullet list: <ul style="list-style-type: none">• apples• oranges• pears Numbered list: <ol style="list-style-type: none">1. wash2. rinse3. repeat

.Rmd files combine code and text and convert to many formats



<https://usesthis.com/interviews/yihui.xie/>

Yihui Xie
RStudio
(Iowa State Statistics PhD)



<https://philosophy.berkeley.edu/people/detail/1>

John Macfarlane
Philosophy, UC Berkeley

[Try pandoc](#)

<https://swcarpentry.github.io/r-novice-gapminder/15-knitr-markdown/>

Literate programming example with R Markdown

This button or command:



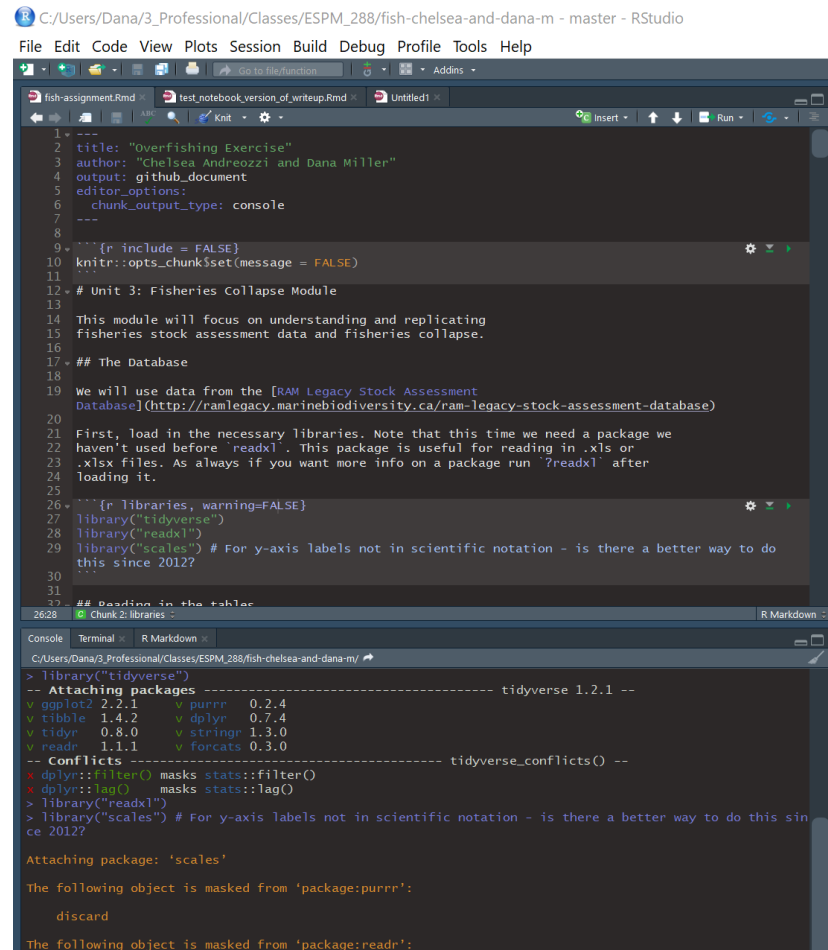
renders formatted document
with executed code and text
(.md, .pdf, .doc etc)

Starting with one
.Rmd file with both
R code and text

This button or
command:



executes all R
code
(output shown in
console)

A screenshot of the RStudio interface. The top pane shows an R Markdown file named 'fishassignment.Rmd'. The code includes a title 'Overfishing Exercise', author 'Chelsea Andreozzi and Dana Miller', and a unit title 'Unit 3: Fisheries Collapse Module'. It also contains R code for loading libraries (tidyverse, readxl, scales) and setting options. The bottom pane shows the console output, which includes the output of the library() function and the output of the readxl::read_excel() function. A blue arrow points from the 'Run' button icon to the console output.

Overfishing Exercise

Chelsea Andreozzi and Dana Miller

Unit 3: Fisheries Collapse Module

This module will focus on understanding and replicating fisheries stock assessment data and fisheries collapse.

The Database

We will use data from the [RAM Legacy Stock Assessment Database](https://ramlegacy.marinebiodiversity.ca/ram-legacy-stock-assessment-database)

First, load in the necessary libraries. Note that this time we need a package we haven't used before `readxl`. This package is useful for reading in .xls or .xlsx files. As always if you want more info on a package run `?readxl` after loading it.

```
library("tidyverse")
library("readxl")
library("scales") # For y-axis labels not in scientific notation - is there a better way to do this since 2012?
```

Reading in the tables

```
download.file("https://depts.washington.edu/ramlegac/wordpress/databaseVersions/RLSADB_v3.0_(assessment_data_only)_e_
"ramlegacy.zip")
path <- unzip("ramlegacy.zip") # unzip the .xls files
sheets <- readxl::excel_sheets(path) # use the readxl package to identify sheet names

# purrr::map is the tidyverse version of lapply
ram <- lapply(sheets, readxl::read_excel, path = path) # read the data from all 3 sheets into a list
names(ram) <- sheets # give the list of datatables their assigned sheet names

## check the names
names(ram)
```

R Markdown documents – key features

History

Rstudio ~ 2011

knitr ~ 2012

Markdown rendering ~ 2013

R Notebook ~ 2016

R Markdown (.Rmd)
document mixing
code and text

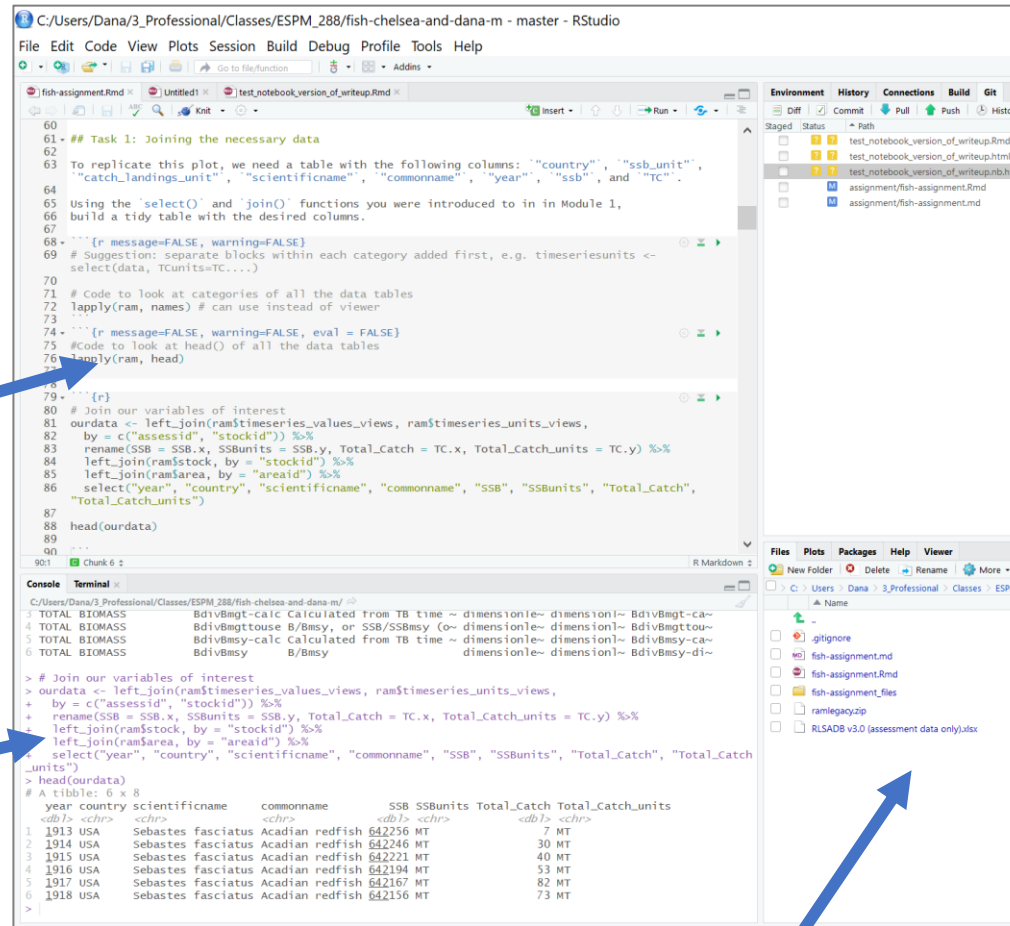
Code can be executed in-line
(above) or in console (below)

See file explorer, help documentation, plots, packages

Convert file to .md, .doc, .pdf, html, html notebook, slides,
book, blog, dashboards, Shiny app, etc

See R Markdown [output formats](#)

Version control
tracker
See [happygitwithr](#)



The screenshot shows the RStudio IDE. The main editor displays an R Markdown document with a task titled "Task 1: Joining the necessary data". The code includes comments and R functions like `left_join`, `select`, and `head`. The console at the bottom shows the output of the code, including a table of fish catch data for the USA from 1913 to 1918.

```
## Task 1: Joining the necessary data
To replicate this plot, we need a table with the following columns: "country", "ssb_unit",
"catch_landings_unit", "scientificname", "commonname", "year", "ssb", and "TC".

Using the 'select()' and 'join()' functions you were introduced to in in Module 1,
build a tidy table with the desired columns.

# Suggestion: separate blocks within each category added first, e.g. timeseriesunits <-
select(data, TCunits=TC....)

# Code to look at categories of all the data tables
library(ram, names) # can use instead of viewer

# Code to look at head() of all the data tables
library(ram, head)

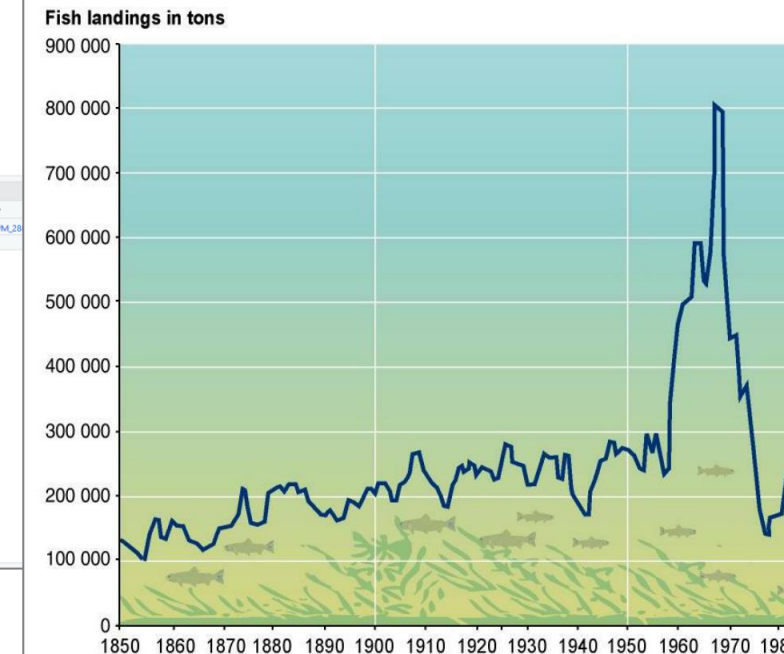
# Join our variables of interest
ourdata <- left_join(ram$timeseries_values_views, ram$timeseries_units_views,
  by = c("assessid", "stockid")) %>%
  rename(SSB = SSB.x, SSBunits = SSB.y, Total_Catch = TC.x, Total_Catch_units = TC.y) %>%
  left_join(ram$stock, by = "stockid") %>%
  left_join(ram$area, by = "areaid") %>%
  select("year", "country", "scientificname", "commonname", "SSB", "SSBunits", "Total_Catch",
    "Total_Catch_units")

head(ourdata)
```

year	country	scientificname	commonname	SSB	SSBunits	Total_Catch	Total_Catch_units
1913	USA	Sebastes fasciatus	Acadian redfish	642256	MT	30	MT
1914	USA	Sebastes fasciatus	Acadian redfish	642246	MT	40	MT
1915	USA	Sebastes fasciatus	Acadian redfish	642221	MT	53	MT
1916	USA	Sebastes fasciatus	Acadian redfish	642194	MT	82	MT
1917	USA	Sebastes fasciatus	Acadian redfish	642167	MT	73	MT
1918	USA	Sebastes fasciatus	Acadian redfish	642156	MT		

Exercise 1: Investigating the North-Atlantic Cod

First, we seek to replicate the following figure from the Millenium Ecosystem Assessment Project using the RAM data.



Source: Millennium Ecosystem Assessment

Task 1: Joining the necessary data

To replicate this plot, we need a table with the following columns: "country", "ssb_unit", "catch_landings_unit", "scientificname", "commonname", "year", "ssb", and "TC".

Using the `select()` and `join()` functions you were introduced to in in Module 1, build a tidy table with the desired columns.

```
# Suggestion: separate blocks within each category added first, e.g. timeseriesunits <- select(data, TCunits=TC....)
```

```
# Code to look at categories of all the data tables
```

Demo: R Markdown

Learning objectives:

1.

- How to create a new R Markdown document in Rstudio
- Make edits
- Knit it
- Share it somewhere
- Have someone else download and run it

2.

- Example of paper written in R Markdown

Reference to set up git w/ R: <http://happygitwithr.com>

Limitations of R + RMarkdown

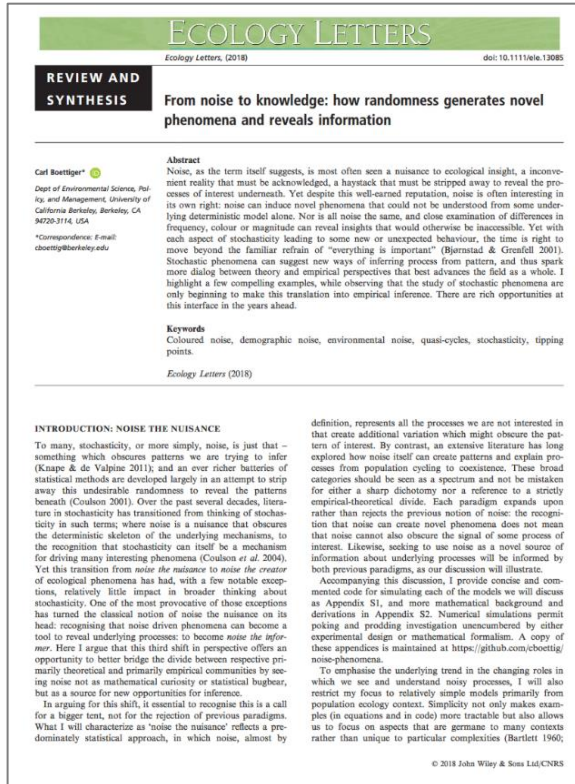
- Not a “general purpose programming language”
- By default, loads entire data into memory
- Less widely used in industry than python
- Possible to run chunks out of order (but can “run all”)

Interactive publications

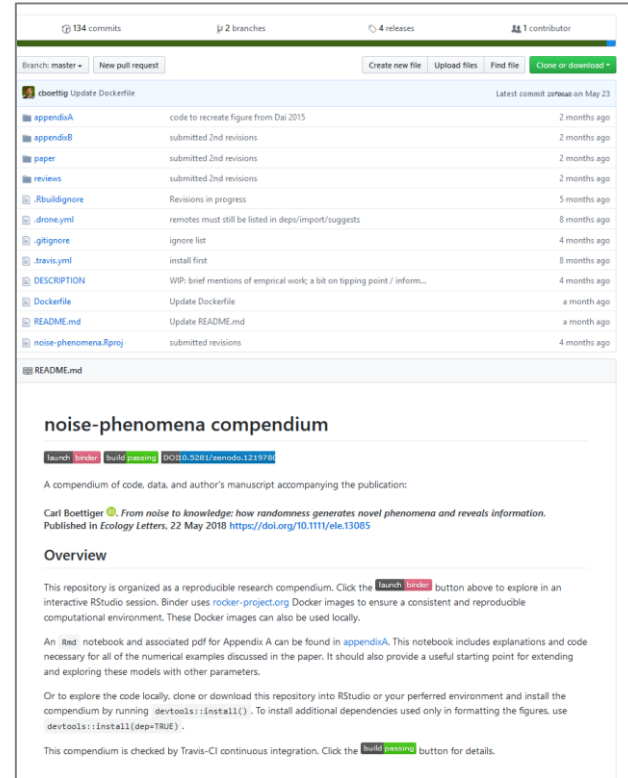
New paper
(open access)

Corresponding “compendium”
(with text, code, data)

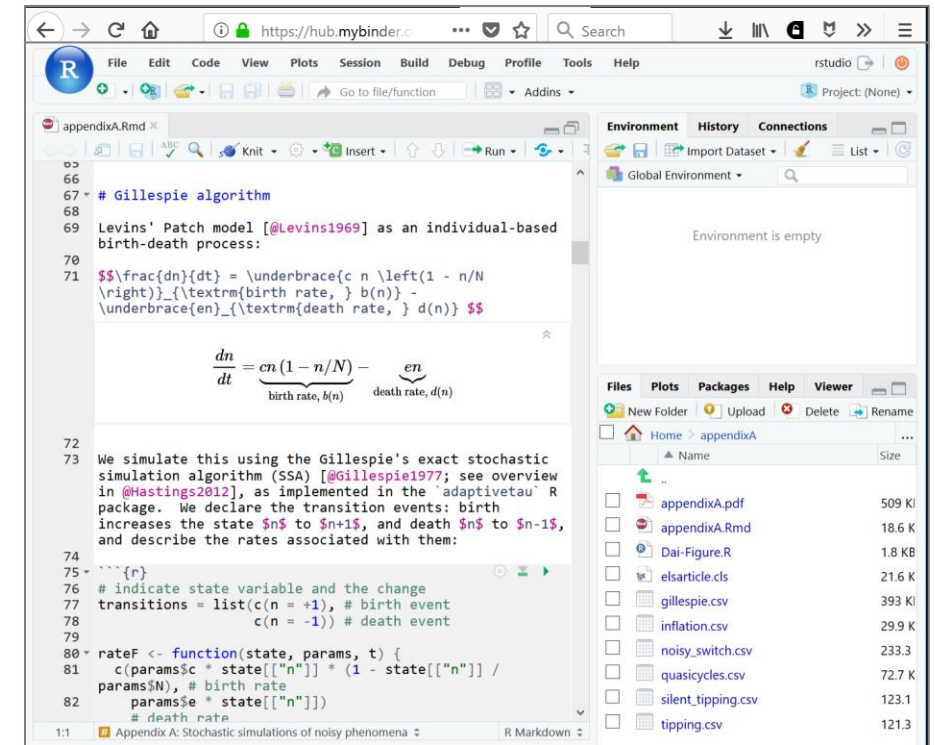
Interactive R Markdown notebook to explore code
(with all required files and software ready
to run in Rstudio in your browser with no
installation required, thanks to Binder)



<https://onlinelibrary.wiley.com/doi/epdf/10.1111/ele.13085>



<https://github.com/cboettig/noise-phenomena>



Demo – reproduce a figure from paper on the previous slide

See the paper: <https://onlinelibrary.wiley.com/doi/epdf/10.1111/ele.13085>

- *Note the PDF includes a link to a GitHub repo*

2. Inspect the code

- *Open the associated GitHub repository containing the code and data used to produce the figures for the paper*
 - *Repo: <https://github.com/cboettig/noise-phenomena>*
 - *Code for the paper: <https://github.com/cboettig/noise-phenomena/blob/master/paper/paper.Rmd>*

3. Reproduce a figure without installing anything on your local computer!

- *Click the  button on the main page of the repo*

Interactive publications – in python too!

New paper
(in this case, a preprint)

+

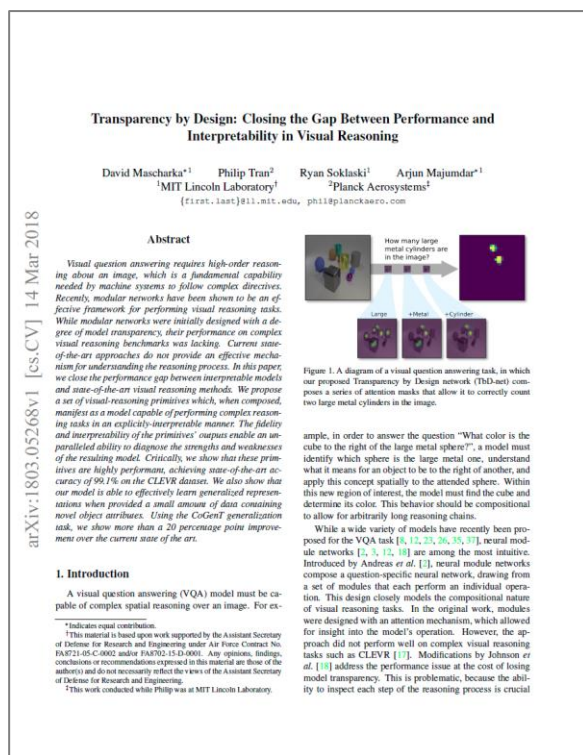
Website explaining paper

(has code to replicate experiments + plots and explicit software versions)

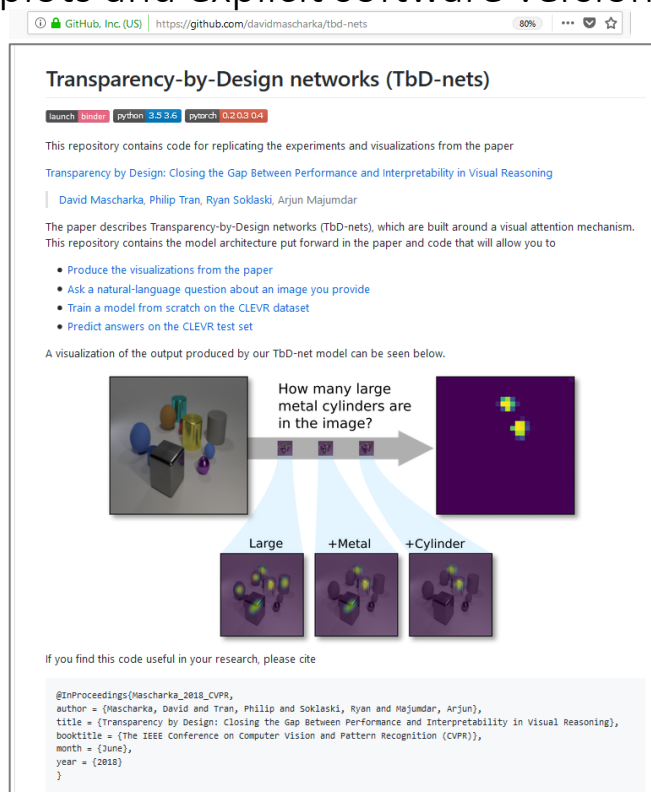
+

Interactive Jupyter notebook with code to apply their method to your own images

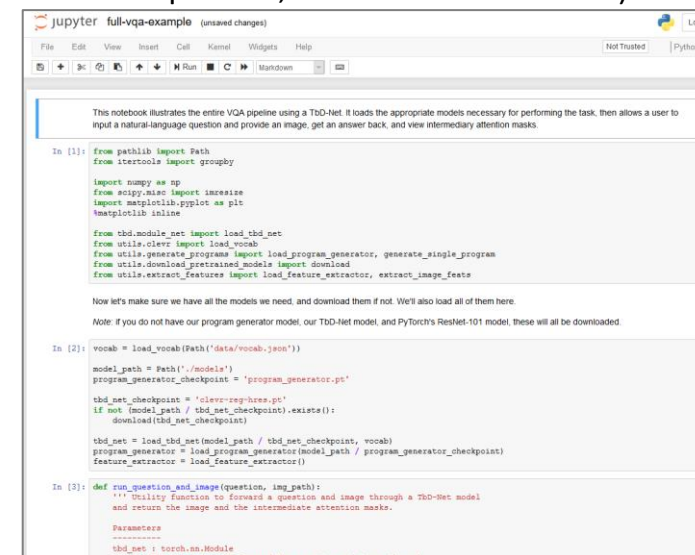
(with all required files and software ready to run in your browser with no installation required, thanks to Binder)



<https://arxiv.org/abs/1803.05268v1>



<https://github.com/davidmascharka/tbd-nets>



<https://mybinder.org/v2/gh/davidmascharka/tbd-nets/binder?filepath=full-vqa-example.ipynb>

Additional reference: [Link](#) to tweet by first author publicizing this work

What next?

UCB Prof Phillip Stark on reproducibility:

“Science is *show me*, not *trust me*”

“Perfection is impossible but improvement is easy”

There are many communities who can help!

[Software Carpentry](#), R for Data Analysis (R4DS), RBloggers, PyData (also has R videos), RLadies, #rstats...

And we can learn a lot from each other 😊

Some more R resources

R for Data Science:
<https://r4ds.had.co.nz>

From Excel to R:
<http://rpubs.com/acolumbus/how-to-use-r-with-excel>
(includes list of common Excel functions in R)

Markdown guide:
<https://bookdown.org/yihui/rmarkdown/>

Free resources aimed at beginners:

[1. introduction to programming in R](#)

[2. Reproducible scientific analysis using RStudio and R](#)

Useful but not fully free:

Datacamp - *no longer recommended due to poor treatment of employees, instructors, and user community in the aftermath of sexual assault by a member of the executive team:*
<https://www.buzzfeednews.com/article/daveyalba/datacamp-sexual-harassment-metoo-tech-startup>

Bonus slides!

(for reference)

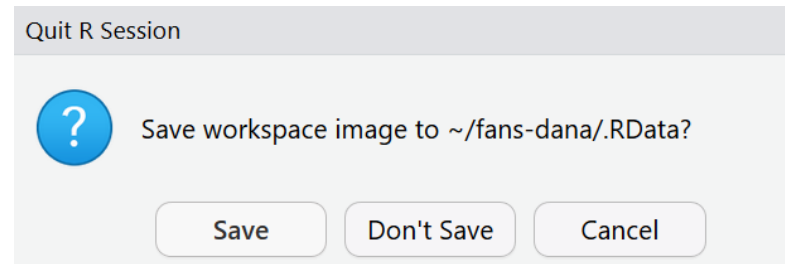
Three R tips you can start using today:

- [here::here](#) (use with RProject)
 - Freedom from hardcoded file paths!

```
library(here)
```

```
db <- read_csv(here("subfolder_name", "file_name.csv"))
```

- [read_csv](#)
 - tidyverse version of read.csv
 - won't coerce strings to factors, outputs a dataframe (tibble)
- “Save workspace image?”
 - Don't save!



Where do we go from here?

ROpenSci's perspective:

- **Train students** by putting homework, assignments & dissertations on the reproducible research spectrum
- **Publish examples** of reproducible research in our field
- **Request code & data** when reviewing
- **Submit to & review for journals** that support reproducible research
- **Critically review & audit** data management plans in grant proposals
- Consider reproducibility wherever possible in **hiring, promotion & reference letters**.

<http://ropensci.github.io/reproducibility-guide/sections/introduction/> - Leveque et al

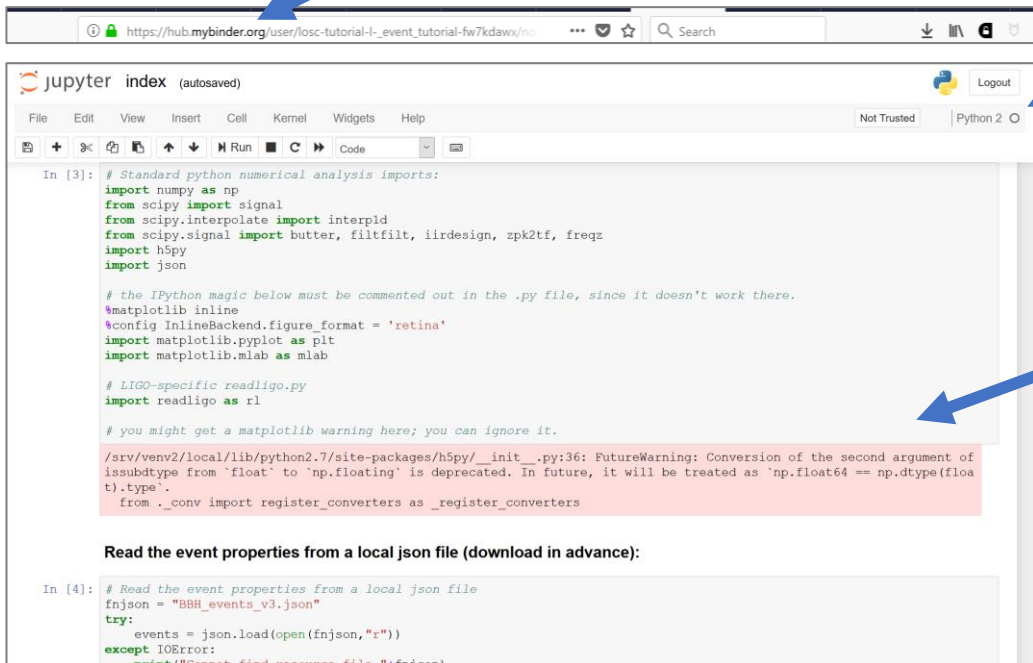
How does Binder work?

“Binder allows you to create custom computing environments that can be shared and used by many remote users”

<https://mybinder.readthedocs.io/en/latest/>

Visible in your browser,
but running on a remote machine

Your browser



```
In [3]: # Standard python numerical analysis imports:
import numpy as np
from scipy import signal
from scipy.interpolate import interp1d
from scipy.signal import butter, filtfilt, iirdesign, zp2tf, freqz
import h5py
import json

# the IPython magic below must be commented out in the .py file, since it doesn't work there.
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab

# LIGO-specific readligo.py
import readligo as rl

# you might get a matplotlib warning here; you can ignore it.

/srv/venv2/local/lib/python2.7/site-packages/h5py/_init_.py:36: FutureWarning: Conversion of the second argument of
issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(floa
t).type`.
  from ._conv import register_converters as _register_converters

Read the event properties from a local json file (download in advance):

In [4]: # Read the event properties from a local json file
fnjson = "BBH_events_v3.json"
try:
    events = json.load(open(fnjson, "r"))
except IOError:
    print("Cannot find resource file "+fnjson)
```

Specified version
of Python
(or other kernels eg R)

You see the code in
the .ipynb file

How does Binder work?

Code for notebook and list of dependencies and files available on GitHub (right now Binder only works with public repositories)

+

Your browser

The image illustrates the workflow for running a LIGO GW150914 tutorial using Binder. It shows the source code on GitHub, the local development environment in JupyterLab, and the final executable URL in a web browser.

Your browser

GitHub Repository: cranmer / ligo-binder
 Bindarize version of LIGO GW150914 tutorial
 7 commits | 1 branch | 0 releases | 3 contributors
[Clone or download](#)

JupyterLab Interface:

```
In [3]: # Standard python numerical analysis imports:
import numpy as np
from scipy import signal
from scipy.interpolate import interp1d
from scipy.signal import butter, filtfilt, iirdesign, zpk2tf, freqz
import h5py
import json

# the IPython magic below must be commented out in the .py file, since it doesn't work there.
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab

# LIGO-specific readligo.py
import readligo as rl

# you might get a matplotlib warning here; you can ignore it.

~/srv/venv2/local/lib/python2.7/site-packages/h5py/_lib/compat.py:36: FutureWarning: Conversion of the second argument of
issubdtype from 'float' to 'np.floating' is deprecated. In future, it will be treated as 'np.float64 == np.dtype(float).type'.
from ._conv import register_converters as _register_converters
```

Read the event properties from a local json file (download in advance):

```
In [4]: # Read the event properties from a local json file
fnjson = "BBH_events_v3.json"
try:
    events = json.load(open(fnjson,"r"))
except IOError:
    pass
```

Browser Window: https://hub.mybinder.org/user/losc-tutorial-l_event_tutorial-fw7kdw/mc...

How does Binder work?

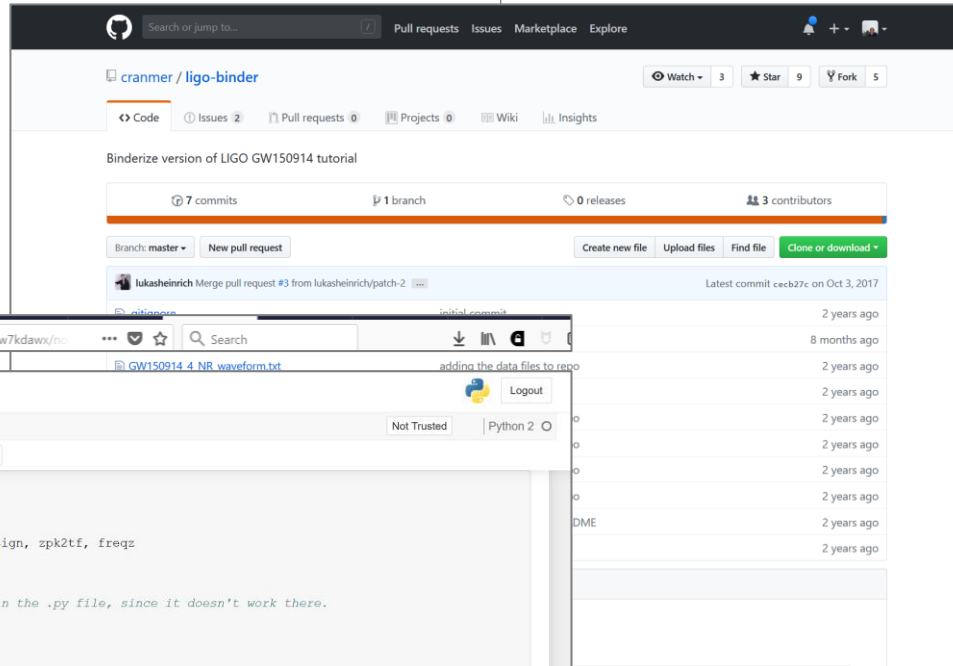


builds Docker image based on repo and generates URL for public access

Code and dependencies in repo

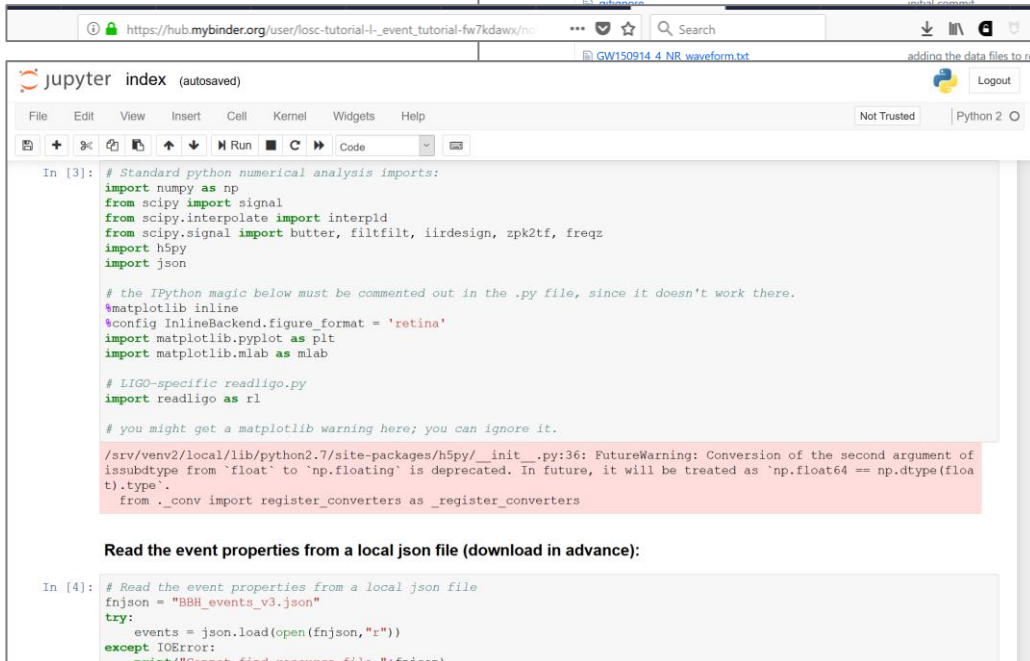


Your browser



required
software
+
packages

included
data



How does Binder work?

