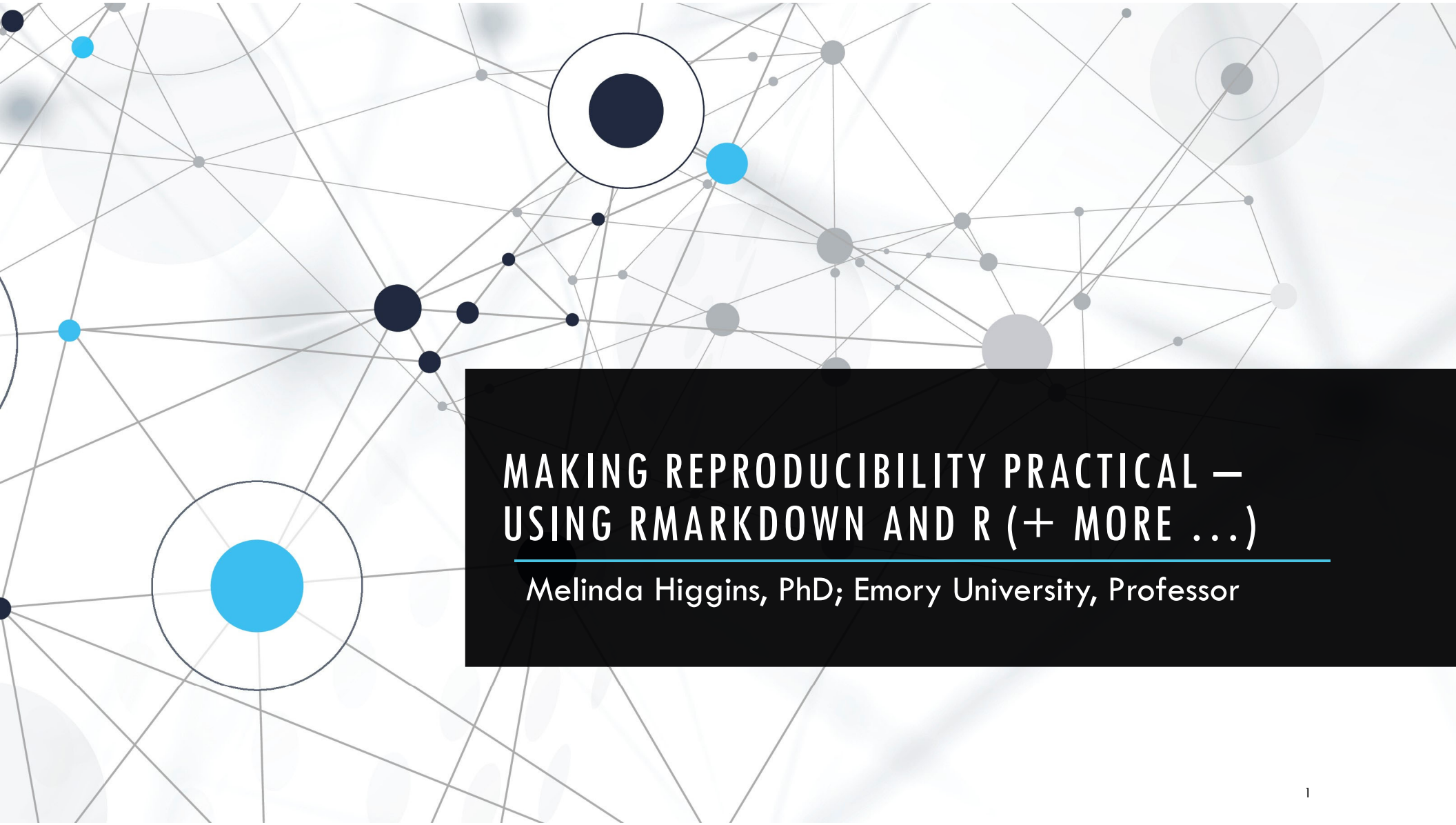
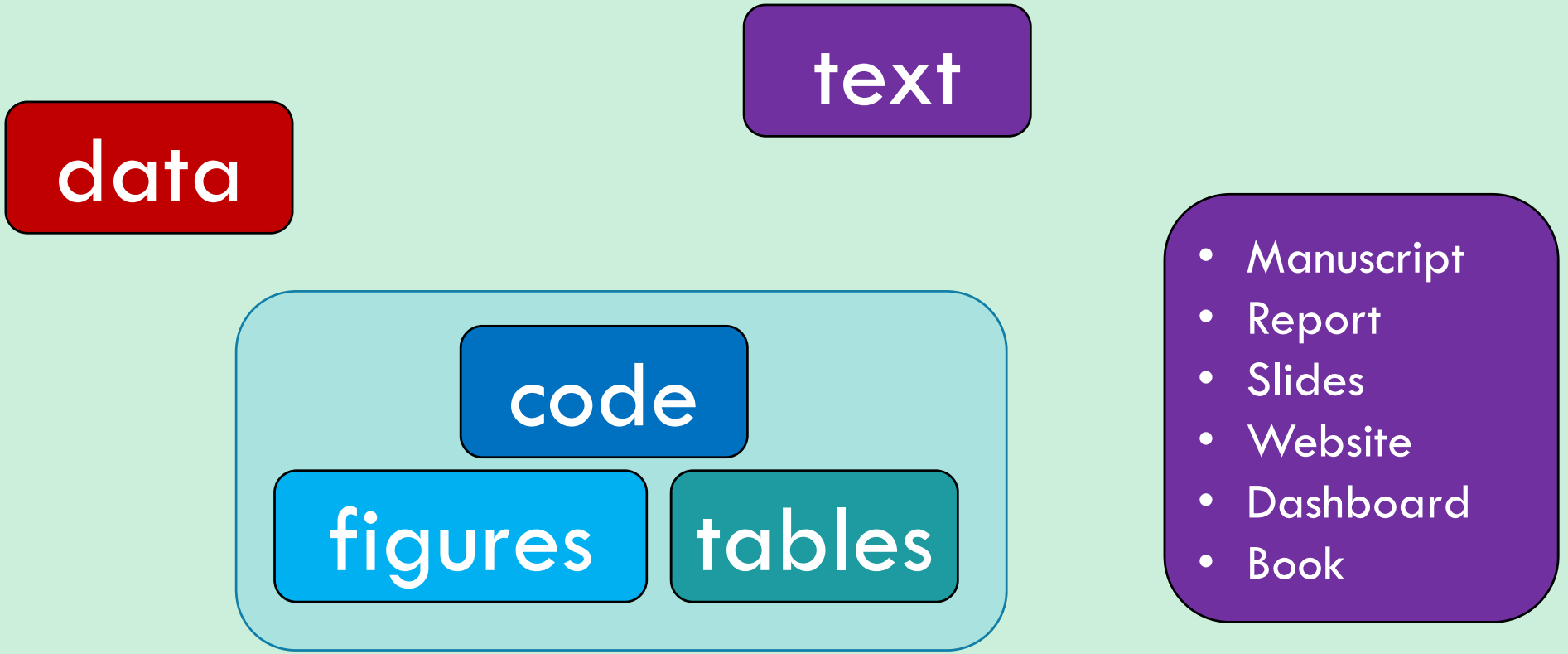


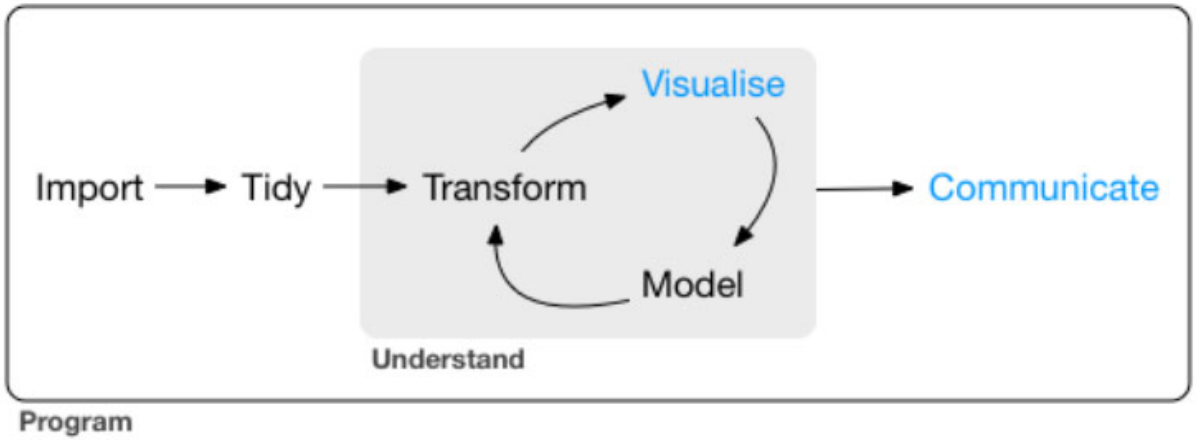
MAKING REPRODUCIBILITY PRACTICAL – USING RMARKDOWN AND R (+ MORE …)

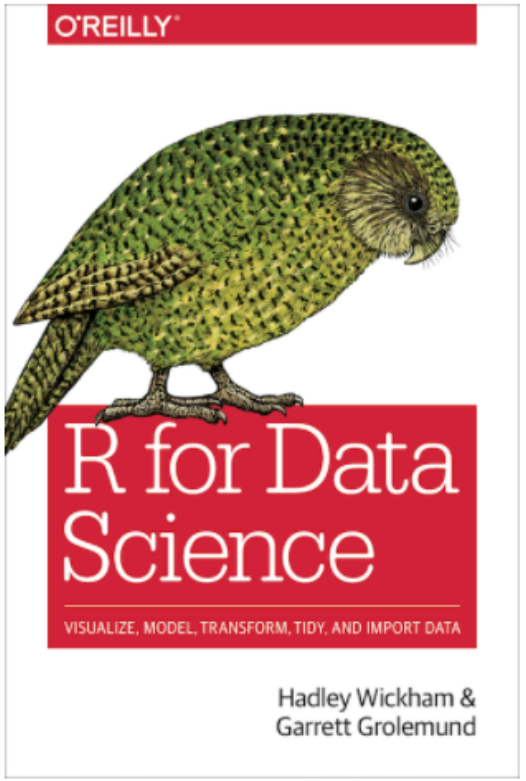
Melinda Higgins, PhD; Emory University, Professor

THE BIG PICTURE



"TIDYVERSE" WORKFLOW





https://r4ds.had.co.nz/communicate-intro.html

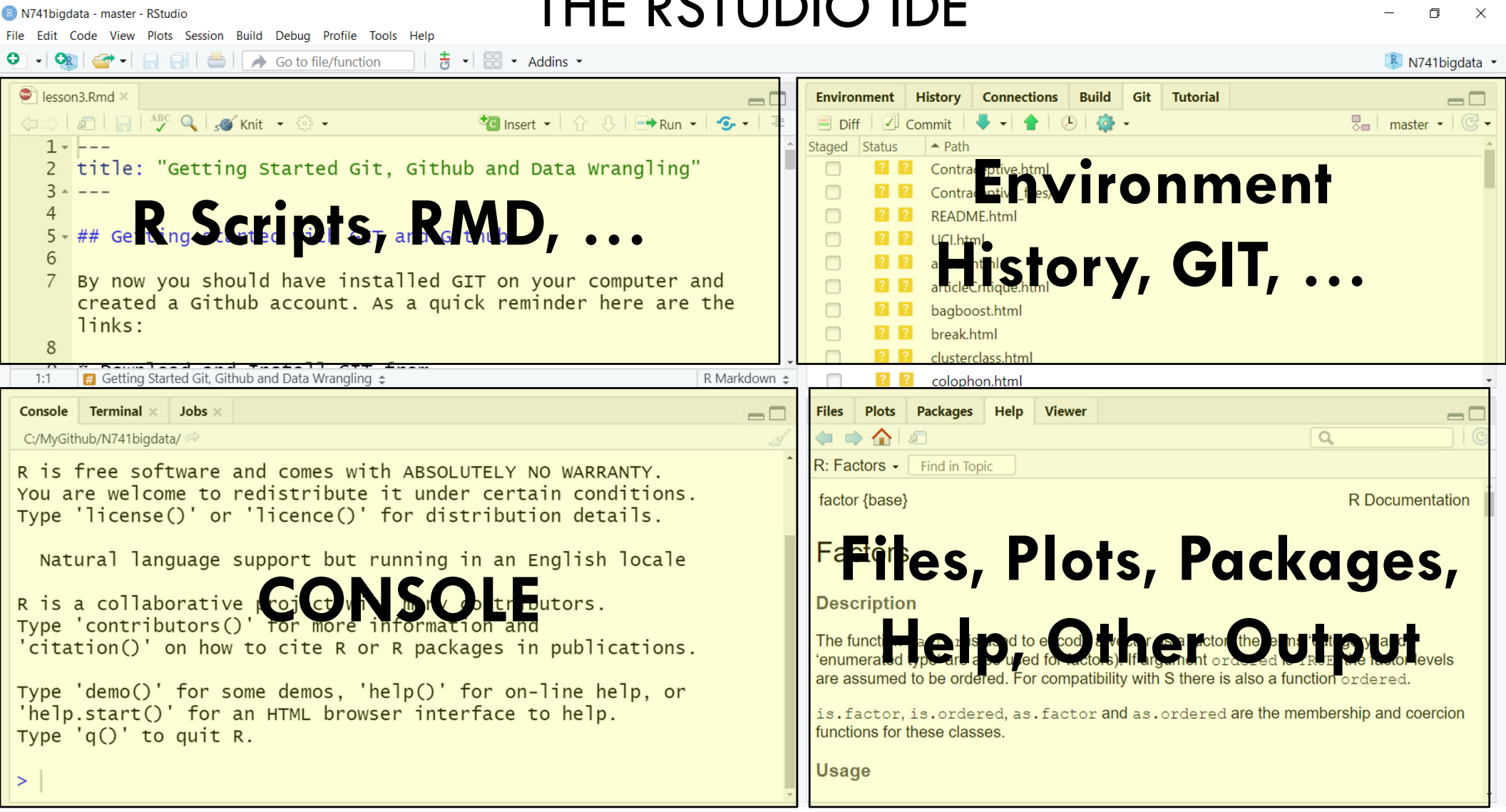
RMARKDOWN (+ PANDOC)

https://pandoc.org/

How it works

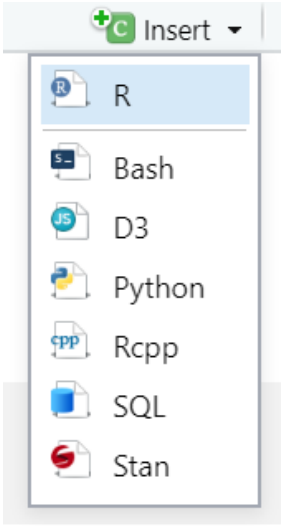
https://rmarkdown.rstudio.com/

THE RSTUDIO IDE



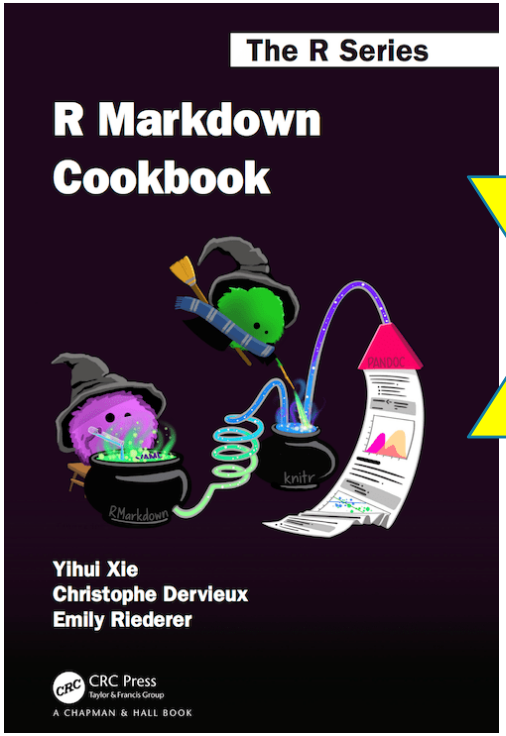
SHORT DEMO

NOT JUST FOR R ANYMORE…




```
> library(bookdown)
> names(knitr::knit_engines$get())
 [1] "awk"         "bash"        "coffee"      "gawk"        "groovy"
 [6] "haskell"     "lein"        "mysql"       "node"        "octave"
[11] "perl"        "psql"        "Rscript"     "ruby"        "sas"
[16] "scala"       "sed"         "sh"          "stata"       "zsh"
[21] "highlight"   "Rcpp"        "tikz"        "dot"         "c"
[26] "cc"          "fortran"     "fortran95"   "asy"         "cat"
[31] "asis"        "stan"        "block"       "block2"      "js"
[36] "css"         "sql"         "go"          "python"      "julia"
[41] "sass"        "scss"        "theorem"     "lemma"       "corollary"
[46] "proposition" "conjecture"  "definition"  "example"     "exercise"
[51] "proof"       "remark"      "solution"
```

MORE THAN R AND PYTHON…



NOVEMBER 2020

15 Other Languages

15.1 Register a custom language …

15.2 Run Python code and interac…

15.3 Execute content conditionall…

15.4 Execute Shell scripts

15.5 Visualization with D3

15.6 Write the chunk content to a …

15.7 Run SAS code

15.8 Run Stata code

15.9 Create graphics with Asympt…

15.10 Style HTML pages with Sas…

https://bookdown.org/yihui/rmarkdown-cookbook/other-languages.html

MORE THAN R AND PYTHON…

15.7 Run SAS code

You may run SAS (https://www.sas.com) code using the `sas` engine. You need to either make sure the SAS executable is in your environment variable `PATH`, or (if you do not know what `PATH` means) provide the full path to the SAS executable via the chunk option `engine.path`, e.g., `engine.path = "C:\\Program Files\\SASHome\\x86\\SASFoundation\\9.3\\sas.exe"`. Below is an example to print out "Hello World":

```
```{sas}
data _null_;
put 'Hello, world!';
run;
```
```

Also see
https://www.ssc.wisc.edu/~hemken/SASworkshops/Markdown/SASmarkdown.html
https://cran.r-project.org/web/packages/SASmarkdown/

MORE THAN R AND PYTHON...

15.8 Run Stata code

You can run Stata (https://www.stata.com) code with the `stata` engine if you have installed Stata. Unless the `stata` executable can be found via the environment variable `PATH`, you need to specify the full path to the executable via the chunk option `engine.path`, e.g., `engine.path = "C:/Program Files (x86)/Stata15/StataSE-64.exe"`. The following is a quick example:

````
```{stata}
sysuse auto
summarize
```
````

The `stata` engine in knitr is quite limited. Doug Hemken has substantially extended it in the Statamarkdown package, which is available on GitHub at https://github.com/Hemken/Statamarkdown. You may find tutorials about this package by searching online for "Stata R Markdown."

MODULARIZATION & AUTOMATION

Child Documents

Parameterized Reports

16.4 Child documents (*)

When you feel an R Markdown document is too long, you may consider splitting it into shorter documents, and include them as child documents of the main document via the chunk option `child` . The `child` option takes a character vector of paths to the child documents, e.g.,

````
```{r, child=c('one.Rmd', 'two.Rmd')}
```
````

Since **knitr** chunk options can take values from arbitrary R expressions, one application of the `child` option is the conditional inclusion of a document. For example, if your report has an appendix containing technical details that your boss may not be interested in, you may use a variable to control whether this appendix is included in the report:

````
Change `BOSS_MODE` to `TRUE` if this report is to be read
by the boss:

```{r, include=FALSE}
BOSS_MODE <- FALSE
```

Conditionally include the appendix:

```{r, child=if (!BOSS_MODE) 'appendix.Rmd'}
```
````

Customized based on Conditional Flags

https://bookdown.org/yihui/rmarkdown-cookbook/child-document.html

17.4 Parameterized reports

In Section 17.3, we mentioned one way to render a series of reports in a `for` -loop. In fact, `rmarkdown::render()` has an argument named `params` specifically designed for this task. You can parameterize your report through this argument. When you specify parameters for a report, you can use the variable `params` in your report. For example, if you call:

```
for (state in state.name) {
  rmarkdown::render('input.Rmd', params = list(state = state))
}
```

then in `input.Rmd` , the object `params` will be a list that contains the `state` variable:

```
---
title: "A report for `r params$state`"
output: html_document
---

The area of `r params$state` is
`r state.area[state.name == params$state]`
square miles.
```

Another way to specify parameters for a report is to use the YAML field `params` , e.g.,

```
---
title: Parameterized reports
output: html_document
params:
  state: Nebraska
  year: 2019
  midwest: true
---
```

https://bookdown.org/yihui/rmarkdown-cookbook/parameterized-reports.html#

CHECKLIST

- Software (R, …)
- Version Control
- Environment
- Workflow
- Reproducible Research
- Tidyverse vs/& Base R
- R Packages
- To GUI or not to GUI
- Datasets, Data Sources
- Data Sharing/Repositories
- Resources

SOFTWARE

R https://cran.r-project.org/

Rstudio https://rstudio.com/products/rstudio/download/

Git https://git-scm.com/

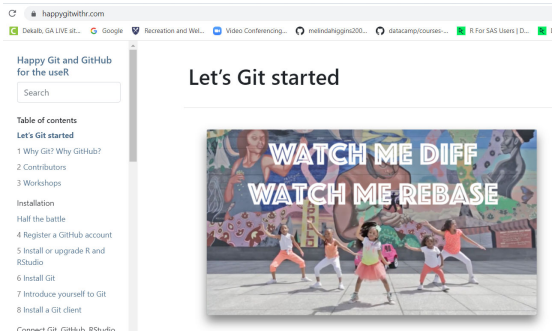
VERSION CONTROL

Github, https://github.com/

[Gitlab, https://about.gitlab.com/]

"Happy Git and GitHub for the UseR"

by Jenny Bryan, [https://happygitwithr.com/]



History for N741bigdata / _site.yml

Commits on Jan 15, 2020

update links to hmwk
melindahiggins2000 committed on Jan 15 ✓ — 97c868a

add files for 2020
melindahiggins2000 committed on Jan 15 ✓ — 4fffc4c

Commits on Apr 24, 2019

add files networks lecture
melindahiggins2000 committed on Apr 24, 2019 ✓ — 96ab8d1

Commits on Apr 17, 2019

add hmwk8 files
melindahiggins2000 committed on Apr 17, 2019 ✓ — 847b8c2

REPRODUCIBLE RESEARCH

- Start from day 1
- All files for a given project: Github ⟷ Rstudio project
- Rmarkdown: data, code, document immediately linked
- Use "knitr" and "Rmarkdown" https://rmarkdown.rstudio.com/
  - documents – HTML, PDF, DOC
  - slides – HTML (ioslides, slidy), PDF (Beamer)
  - others – e.g. dashboards

WORKFLOW

1. Create Github repo
2. Create Rstudio project – version control to Github
3. Create/Begin with Rmarkdown [https://rmarkdown.rstudio.com/]
4. Knit (check that everything is working)
5. Modify code and/or text in Rmarkdown, Knit
6. GIT: Add, Commit, Push
7. Refresh, check GIT and Github

HELPFUL R PACKAGES

- **tidyverse** – mainly **dplyr, ggplot2, readr**
- **foreign** – importing of SAS, SPSS, Stata
- **Hmisc** – lots of useful functions from Frank Harrell
- **arsenal** – making nice tables (HTML, PDF and **WORD**)
- **knitr, Rmarkdown, printr, kablextra**
- **tinytex** - create PDFs without full LaTeX installation!!

https://www.tidyverse.org/

Tidyverse

TIDYVERSE VS/& BASE R

- Tidyverse – packages that work well together
  - **dplyr** - pipe %>% workflow
  - **ggplot2** – build graphs with + layers
- Base R
  - tibble data frames ≠ data.frame
  - data import **haven** vs **foreign** (SAS, SPSS or Stata files)
  - "haven labeled" variables
  - factors (pros and cons – useful to have both)
  - selecting variables (dplyr::select() and dplyr::pull() versus $ versus [,2] – useful to know all of these)

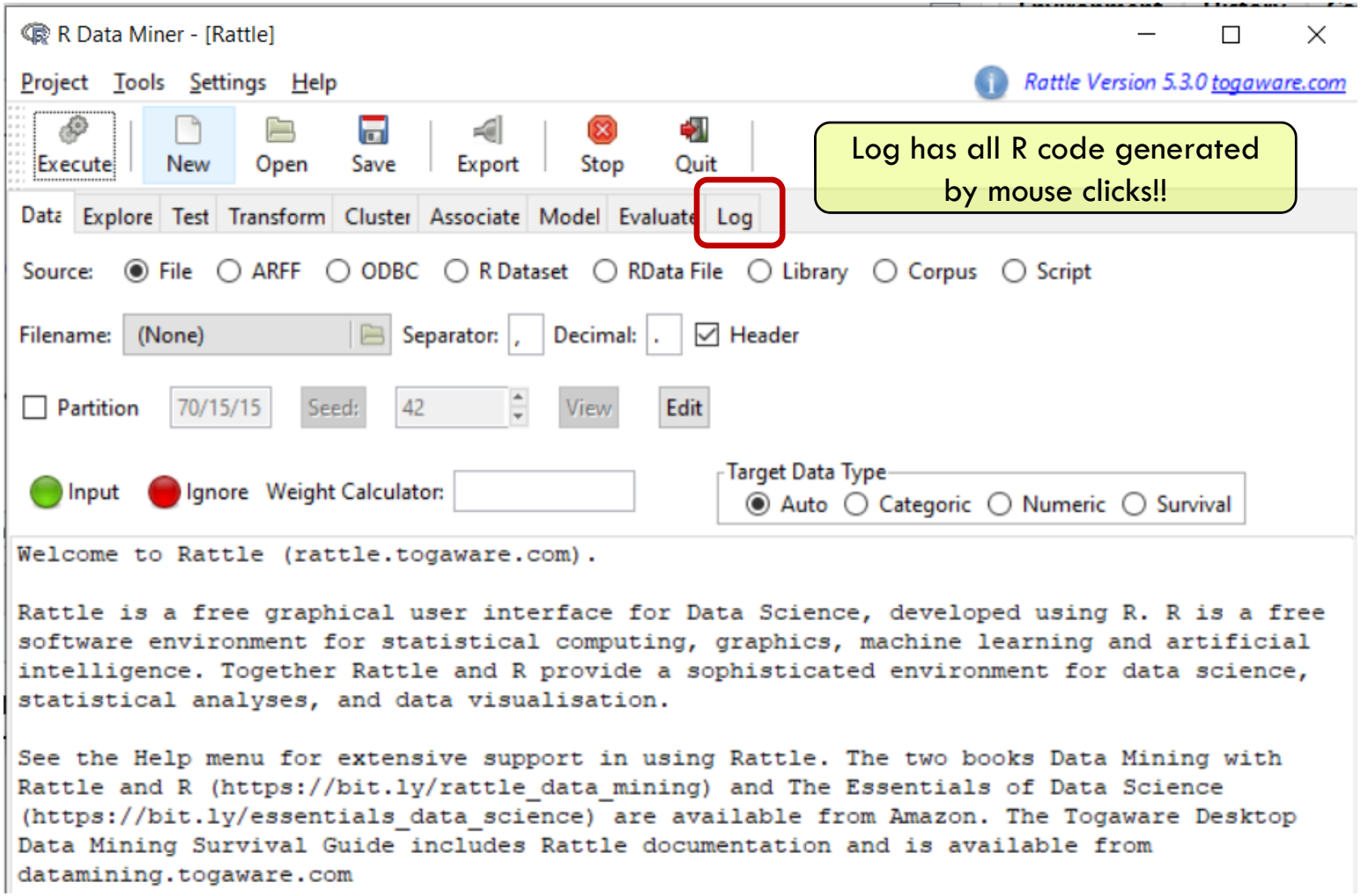
TO GUI OR NOT TO GUI

- no GUI – all code
- every step is captured and documented
- Rmarkdown always begins with clean environment supports reproducible research workflow

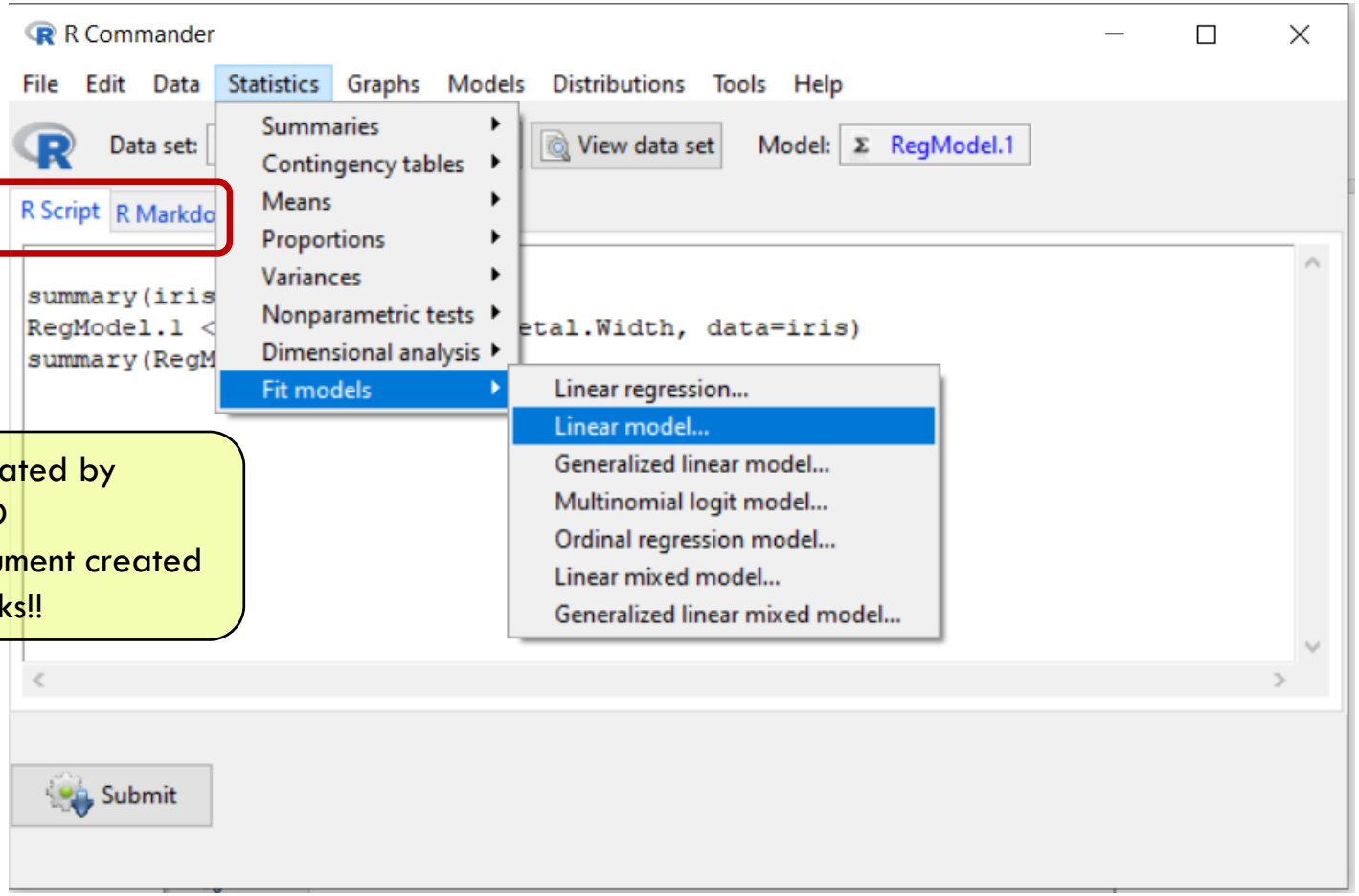
TO GUI OR NOT TO GUI

- GUIs - packages: **rattle** and **Rcmdr**
  - very helpful for beginners
  - provides insights into data mining
  - **rattle,** https://rattle.togaware.com/
    - saves all R code
  - **Rcmdr,** https://www.rcommander.com/
    - saves all R code
    - also creates a draft Rmarkdown file

https://rattle.togaware.com/



https://www.rcommander.com/



All R code generated by mouse clicks AND Rmarkdown document created with R code chunks!!

ENVIRONMENT(S)/CONTAINER(S)

PC & Macs (also Linux)

Rstudio.cloud, https://rstudio.cloud/

** no longer free, tiered pricing **

Local R/Rstudio server

https://rstudio.com/products/rstudio/#rstudio-server

AWS, Docker, …

OTHER CONSIDERATIONS

- Code testing (**testthat**)
- Package Management (**packrat**)
- Continuous Integration
- Data/Code Sharing - Repositories

Container &
Version "locking"

Travis CI

DATA RESOURCES, SECURITY, SHARING @ EMORY

- Data Resources @ Emory
  - https://researchdata.emory.edu/index.html
- Data Security Policies @ Emory
  - https://it.emory.edu/security/awareness/archive/encrypt.html
- Data Sharing/Publication
  - https://researchdata.emory.edu/share/repositories.html
- Data Management Plan (DMP Tool)
  - https://researchdata.emory.edu/plan/dm-planning/write-dmp.html
- Rigor and Reproducibility Lecture Series
  - https://guides.libraries.emory.edu/rigor-rep

RESOURCES

- Happy Git and Github for the UseR, https://happygitwithr.com/
- Stat 545, https://stat545.com/ and https://stat545.stat.ubc.ca/
- Quick R, https://www.statmethods.net/
- R Graphics Cookbook, https://r-graphics.org/ and http://www.cookbook-r.com/Graphs/

RESOURCES

- Rstudio education, https://education.rstudio.com/
- Datacamp for the classroom, https://www.datacamp.com/groups/education
- Github education, https://education.github.com/
- Gitlab for education, https://about.gitlab.com/solutions/education/
- Mine Cetinkaya-Rundel, https://mine-cetinkaya-rundel.github.io/teach-r-online/ - also see **ghclass** R package for managing students in Github

https://melindahiggins2000.github.io/N741bigdata/

COURSE NUMBER, TITLE:

COURSE DESCRIPTION

COURSE OBJECTIVES

TEACHING AND LEARNING

N741 Big Data Analytics

COURSE NUMBER, TITLE:

NRSG 741, Big Data Analytics for Healthcare

COURSE DESCRIPTION

This course will describe the concepts underlying the field of study identified as big data analytics along with its application in healthcare. The theoretical underpinnings of these concepts will be presented along with applications in healthcare, including knowledge discovery, precision medicine/nursing, and the development of targeted interventions to improve health outcomes. Commonly used methods in big data analytics will be reviewed, and the challenges related to gathering, analyzing, visualizing, and interpreting big data will be discussed. Hands-on computer laboratory experience with these techniques relevant to an identified area will be included.

QUESTIONS?

My contact info:

Melinda.higgins@emory.edu

https://melindahiggins.netlify.app/

http://nursing.emory.edu/faculty-and-research/directory/profile.html?id=980