Reproducible report generation using Markdown

Stas Kolenikov

June 29, 2020

BOLD
THINKERS
DRIVING
REAL-WORLD
IMPACT

Purposes

1. Introduce Markdown as a bare bones text-only format that allows formatting text, inserting links, inserting images.
2. Introduce RMarkdown as a tool to weave together text and analysis (tables, figures, plots, individual numbers).
3. Demonstrate other uses of Markdown (Stata, Python notebooks, diagrams).
4. Demonstrate flexible / reproducible report generation with parametric Markdown templates and input parameters.
5. Provide resources for further professional development.

Helpful upfront knowledge: some R; some HTML

Focus on reproducibility

Do, redo, and do over again:

- Produce reports for several waves of data.
- Produce reports for the data that is being collected / cleaned.
- Produce updated analyses / updated text for academic papers revisions.

Modern take: literate programming tools based on plain text

- Markdown to code and to write up,
- GitHub to store versions of the code and the writeup, plus optionally the data.

Adopt sensible file storage structures, code documentation practices.

Avoid point and click; avoid copy and paste.

Literate programming

> Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do.
>
> — Donald E. Knuth, Literate Programming, 1984

Copy-paste vs. markdown

KFF tracking poll

Case study: Kaiser Family Foundation Kaiser Health Tracking Poll

Simplified KFF report: source

Simplified KFF report: rendered

Text, tables and graphs version of KFF: rendered

Data sources:

March 2019: https://ropercenter.cornell.edu/ipoll/study/31116193, doi:10.25940/ROPER-31116193

April 2019: https://ropercenter.cornell.edu/ipoll/study/31116357, doi:10.25940/ROPER-31116357

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Extended examples

Books in markdown: http://bookdown.org, https://r4ds.had.co.nz/

Flexible parametric documents in markdown:

https://medium.com/@urban_institute/iterated-fact-sheets-with-r-markdown-d685eb4eafce (GitHub: https://github.com/UI-Research/rmarkdown-fact-pages)

Dashboards in markdown:

https://rmarkdown.rstudio.com/flexdashboard/examples.html

Diagrams: Typora diagrams

Markdowns with several languages: ICHPS 2020 Complex Surveys

Browser presentations: this webinar (Stas can show the source code)

Basics of Markdown

BOLD
THINKERS
DRIVING
REAL-WORLD
IMPACT

Markdown

When you make your text **bold**, or *italic*, or `paste(code, snippets)`, or create

- items in lists
- and more items
    - some of which may be nested

you are *marking* certain elements of your text to be formated in a special way. (The heading above is also a marked text.)

Markdown modifies this to a very bare bones, text-only, no-mouse-selection-needed process. The name is supposed to be a play on "markup" which is a technical term to describe languages like HTML, XML, or LaTeX.

https://daringfireball.net/projects/markdown/syntax

Markdown elements

```
Pieces of `code`

Text in _italics_

Text in **bold**

### Heading 3

- unnumbered item
    + nested list item

1. numbered item

[text of a link](http://some.url.com/with/path)

![](some_meme_file.jpg)
```

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Exercise

1. Go to SlackEdit.io online editor.
2. Open/create a new file (click 🗀 to open the file menu).
3. Enter the following in the editor with formatting, watch the rendered version in the right pane:

My first markdown file

Hi, I am **Stas** *Kolenikov*. I use Markdown to:

- create simple documents
    - take notes
    - make to-do lists

05:00

Solution

```
# My first markdown file

Hi, I am **FirstName** _LastName_. I am using Markdown to:

- create simple documents
    + take notes
    + make to-do lists
```

Feel free to carry on and do more of the basic markdown tutorials at https://www.markdowntutorial.com/.

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Markdown + code

BOLD THINKERS DRIVING REAL-WORLD IMPACT

R Markdown

Additionally, combination of Markdown with R and some other languages allows to

- incorporate source code
- incorporate the output, such as numbers, tables, and plots

... into data-driven documents. (You are explaining to other humans, including your future self, what you want to do with the data!)

Code chunks:

~~~
```r

# some code here

```
~~~

Inline code:

The total number of observatsions is `` `r nrow(mydata)` ``.

From source to the report

Your source markdown file contains

- your text describing the data, the analysis, and possibly the findings;
- the code to deal with the data.

How do you get the numbers into the document? How do you know the code actually runs?

Compilation stage: instruct your software to…

- run the underlying code;
- insert the numbers into output;
- present this in a human-readable form (HTML, Word, PDF, PPT, …)

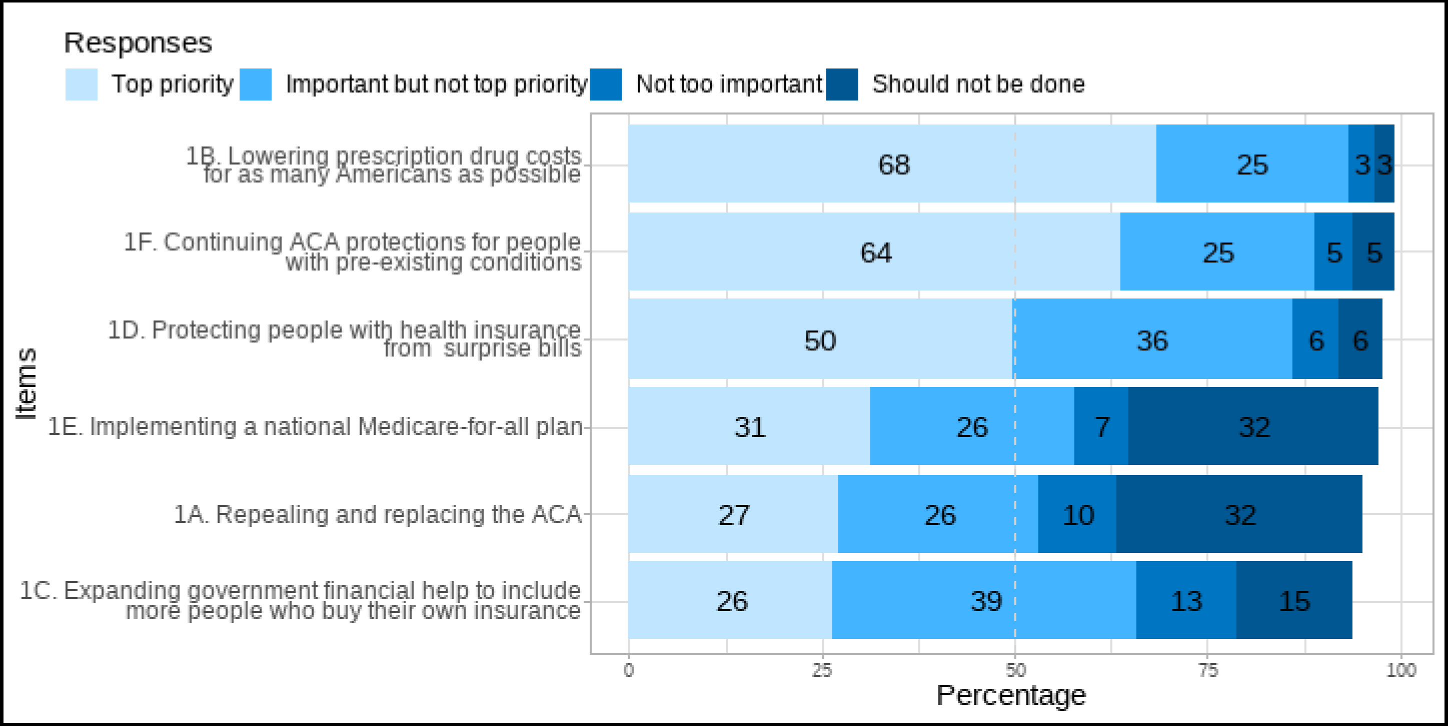
🧶 Knit button in RStudio

Almost everything you need to know

https://www.rstudio.com/resources/cheatsheets/

Markdown: https://github.com/rstudio/cheatsheets/raw/master/rmarkdown-2.0.pdf

Graphics



```
library(ggplot2)
```

and a myriad of extentions to it. Graphic output of a chunk is inserted.

Tables

Group	N (unweighted)	MOE
Democrats	342	+/- 6.2 percentage points
Republicans	273	+/- 6.9 percentage points
Independents	544	+/- 4.9 percentage points
Dem-leaning Indeps	199	+/- 8.3 percentage points
Rep-leaning Indeps	181	+/- 8.4 percentage points

A lot of output looks like a table already: regression results, cross-tabs. Other research results may be assembled as small data sets of summary statistics.

```
# aimed at HTML
library(kableExtra)
library(gt)
# aimed more at MS products
library(flextable)
```

BOLD THINKERS DRIVING REAL-WORLD IMPACT

library(kableExtra)

Compared to others, produces much nicer tables in HTML, but MS Word breaks down.

Group	N (unweighted)	MOE
Total	1,203	+/- 3.3 percentage points
Party ID		
Democrats	342	+/- 6.2 percentage points
Republicans	273	+/- 6.9 percentage points
Independents	544	+/- 4.9 percentage points
Dem-leaning Indeps	199	+/- 8.3 percentage points
Rep-leaning Indeps	181	+/- 8.4 percentage points

Documentation vignette here.

library(kableExtra)

Compared to others, produces much nicer tables in HTML, but MS Word breaks down.

~~~
```r MOE_kableExtra
readRDS(here('KFF_MOE.Rds')) %>%
  mutate(`N (unweighted)` = cell_spec(`N (unweighted)`, "html",
            font_size = spec_font_size(sqrt(neff), end=20),
            align = 'c') ) %>% # fancy sample size
  select(Group, `N (unweighted)`,MOE) %>%
  kable(escape=FALSE) %>%
  row_spec(0,color='white',background="#48A9C5") %>%  # header colors
  row_spec(2,align="center") %>%                      # center Party ID
  row_spec(c(2,4,6), background = '#3e4040') %>%      # alternate rows
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```
~~~

library(flextable)

Compared to others, produces reasonably nice table in both HTML and MS Word.

Group	N (unweighted)	MOE
Total	1,203	+/- 3.3 percentage points
Party ID		
Democrats	342	+/- 6.2 percentage points
Republicans	273	+/- 6.9 percentage points
Independents	544	+/- 4.9 percentage points
Dem-leaning Indeps	199	+/- 8.3 percentage points
Rep-leaning Indeps	181	+/- 8.4 percentage points

Overview here

library(flextable)

~~~
```r MOE_flextable
readRDS(here('KFF_MOE.Rds')) %>%
  select(Group, 'N (unweighted)', MOE) %>%
  flextable() %>%             # convert to flextable
  align(j=1,align="left") %>%         # change alignment for Group
  align(i=2,j=1,align="center") %>%   # put "Party ID" in the center
  bg(bg="#48A9C5", part="header") %>% # header colors
  color(color="white", part="header") %>%
  bg(bg="white", part="body") %>%
  autofit()                           # some magic in column widths
```
~~~

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Customizing markdown output

BOLD
THINKERS
DRIVING
REAL-WORLD
IMPACT

R chunk options

Chunk options modify how the code runs within the chunk, and how the output is being displayed.

~~~
```r ChunkName, echo=FALSE, fig.height=4, fig.width=6

# some graphics code here that I don't care to see in the final output

```
~~~

Review the chunk options in RStudio cheatsheet

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Parametric markdown documents

The more advanced forms of markdown documents use inputs/parameters.

- Output format: HTML (the most flexible, and the only interactive), Word, PDF (requires LaTeX)
- Task to perform: analysis vs. reporting the results
- Input data files
- Formatting specifications (e.g. colors to use)

Compilation of parametric markdown documents is typically done from command line / R scripts:

```
rmarkdown::render( ...,
    params=list(todo='analysis',
          input_data='KFF-April2019.sav'))
```

Parametric markdown: output

```
filename_stub <- 'MyReport'
rmarkdown::render(
  input=paste0(filename_stub,.'Rmd'),
         # input file
  output_file=I(paste0(filename_stub,'-',Sys.Date())),
         # output file with date embedded, no extension
  output_format='docx_document')
         # output format; the extension is inferred from it
  # output_format='html_document')
         # alternative output format; here, commented out
)
```

PDF output requires a LaTeX compiler; not presented in this talk.

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Parametric markdown: YAML header

```
---
title: "Reproducible report generation using Markdown"
author: "Stas Kolenikov"
date: 2020-06-29
output:
  html_output:
    code_folding: hide
    toc: TRUE
  word_document:
    reference_docx: Abt_2018_Report_Template.docx
---
```

Parametric markdown: YAML header

```
---
title: "Reproducible report genercation using Markdown"
author: "Stas Kolenikov"
date: 2020-06-29
params:
  todo: analysis
  # todo: reporting
  data_source: April2019.dta
  # data_source: Mar2019.dta
  digits: 4
---
```

Parametric markdown: tasks to do

~~~
```r long_analysis
if (params$todo=="analysis") {
  # mixed modeling code that takes hours to execute
  model1 <- ...
  saveRDS(model1, file='model1.Rds')
} else if (params$todo=="reporting") {
  model1 <- readRDS(file='model1.Rds')
}
```
~~~

Parametric markdown: varying data

Explicit naming:

~~~
---
params:
  data_source: KFF-April2019.sav
---

```r read_data
main_data <- read.sav( params$data_source )
```
~~~

Parametric markdown: varying data

Combining parts:

~~~
---
...
params:
  month: April
---

```r read_data_by_month
main_data <- read.sav( paste0( 'KFF-',params$month, '2019.sav' ) )
```
~~~

Which can be rendered with

```
filename_stub <- 'MyReport'
month         <- 'April'
rmarkdown::render(input=paste0(filename_stub,.'Rmd'),    # input file
  output_file=I(paste0(filename_stub,'-',month,'-',Sys.Date())),
  output_format='docx_document'),                        # output format
  params=list(month=month)                               # pass parameters
)
```

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Hands-on: edit the KFF report

Go back to the KFF report markdown file and make it parametric (take month and year as input)

???

Markdown presentations

BOLD
THINKERS
DRIVING
REAL-WORLD
IMPACT

Markdown presentations

Several libraries available, with slightly differing functionality (and ease of adjusting options):

- `ioslides` (probably the easiest to run)
  - presenter mode
  - custom CSS
- `slidy`
  - fixed timer per slide
- `shower`
  - easier to create self-contained presentation HTML files
  - not compatible with `flextable`
- `xaringan` (this presentation)
  - not compatible with `shiny` interactive elements
  - presenter mode

Slides are separated by `---` (separator line) or by `##` (Header 2).

Handouts: print from browser to PDF.

Markdown presentations

Abt colors:

```
style_solarized_dark()
style_xaringan(
  background_color = '#000000',
  text_color       = '#B7C9D3',
  header_color     = '#DA291C',
  title_slide_text_color          = '#DA291C',
  title_slide_background_color    = '#DFD1A7',
  title_slide_background_image    = 'AbtLogo2.png',
  title_slide_background_size        = '200px 100px',
  title_slide_background_position    = 'bottom 10px left 20px',
  background_image      = 'AbtLogo2.png',
  background_size          = '90px 45px',
  background_position      = 'bottom 10px left 10px',
  link_color            = '#7566A0',
  code_inline_color     = '#C3C6A8',
  code_highlight_color = '#E87722'
)
```

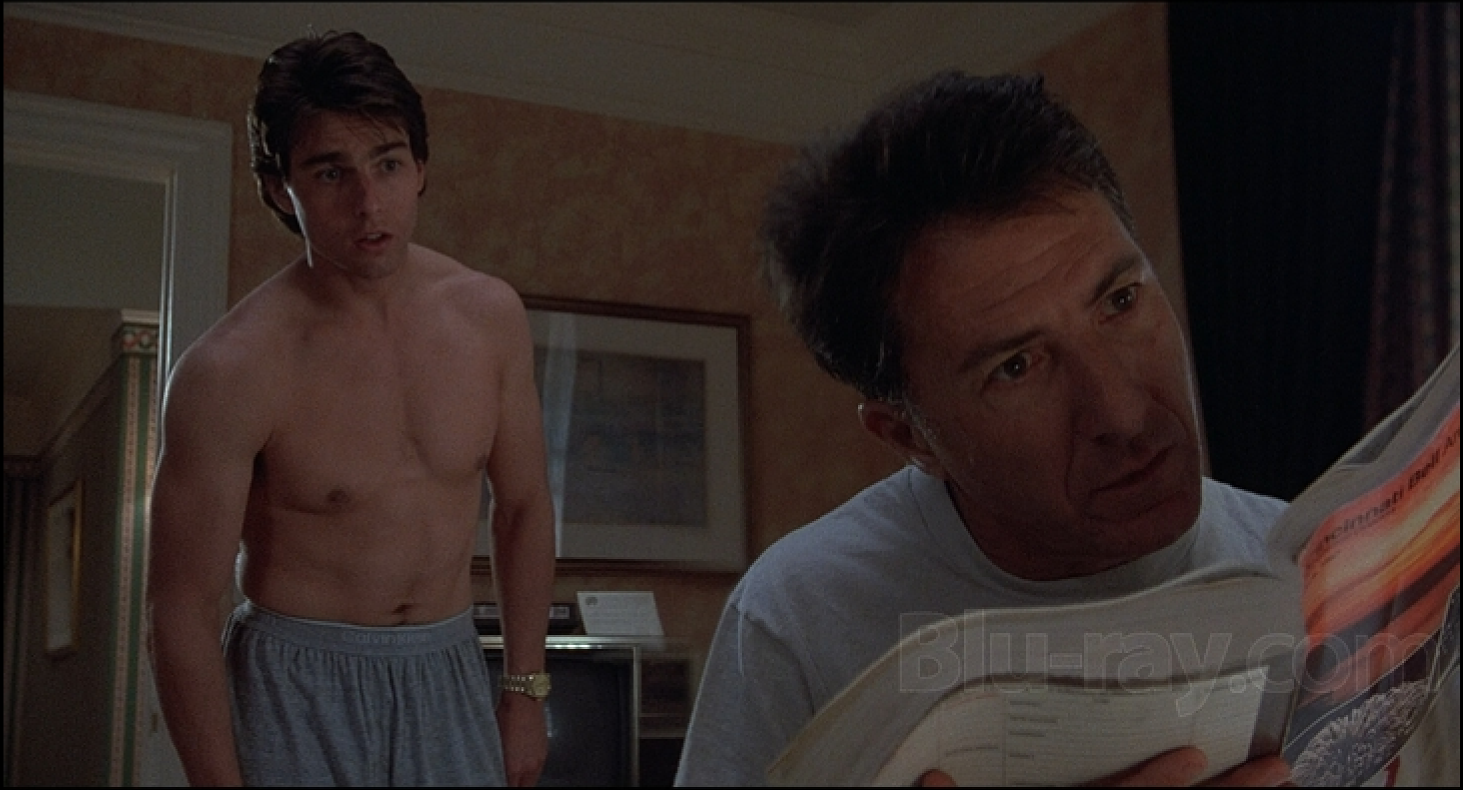
A few words about version control

BOLD
THINKERS
DRIVING
REAL-WORLD
IMPACT

Reasons for markdown

1. Combine narrative and data analysis
2. Reproduce your work (reports, revisions)
3. Reuse the code with different options, sources, formatting, ...
4. Version control over the plain text of markdown documents

Reasons for markdown: version control



Version control

This talk:

- https://github.com/skolenik/markdown2020

A complete introduction to using Git with RStudio

- https://happygitwithr.com/ by @JennyBryan

Markdown and other languages

BOLD
THINKERS
DRIVING
REAL-WORLD
IMPACT

Markdown

MARKDOWN, MARKDOWN
EVERYWHERE
imgflip.com
LIGHTYEAR
SPACE RANGER

Markdown in other languages

Stata:

- official Stata markdown `<<dd-syntax>>`
- user-contributed `markstat` (interface to Pandoc markdown) package, https://data.princeton.edu/stata/markdown

(See Stas' ICHPS 2020 Complex Surveys)

Markdown in other languages

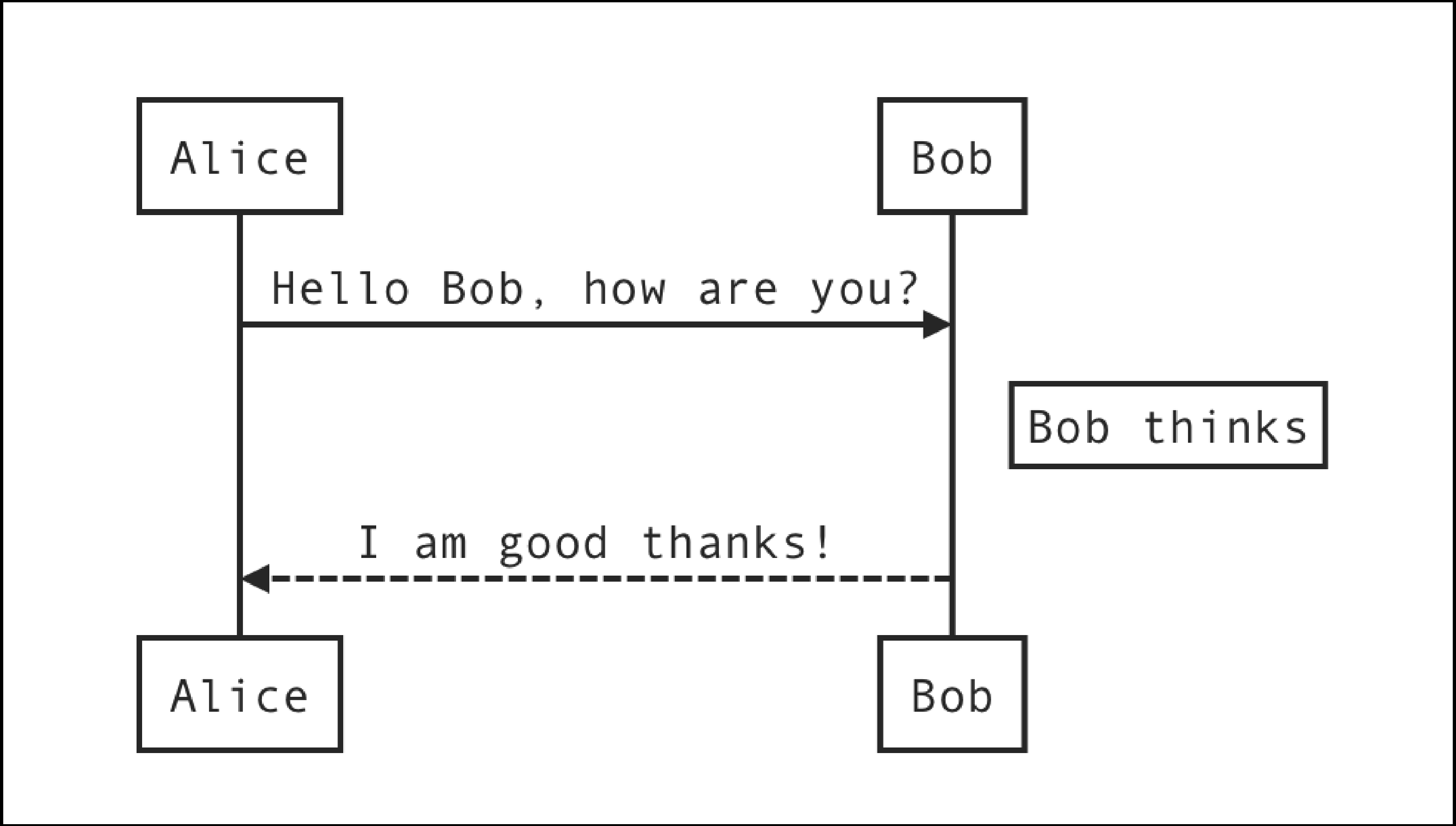
Python: markdown cells in notebooks

Hands-on part for Stas: launch a notebook to show

Markdown in other languages

Diagrams: `mermaid` in Typora

Sequence diagrams

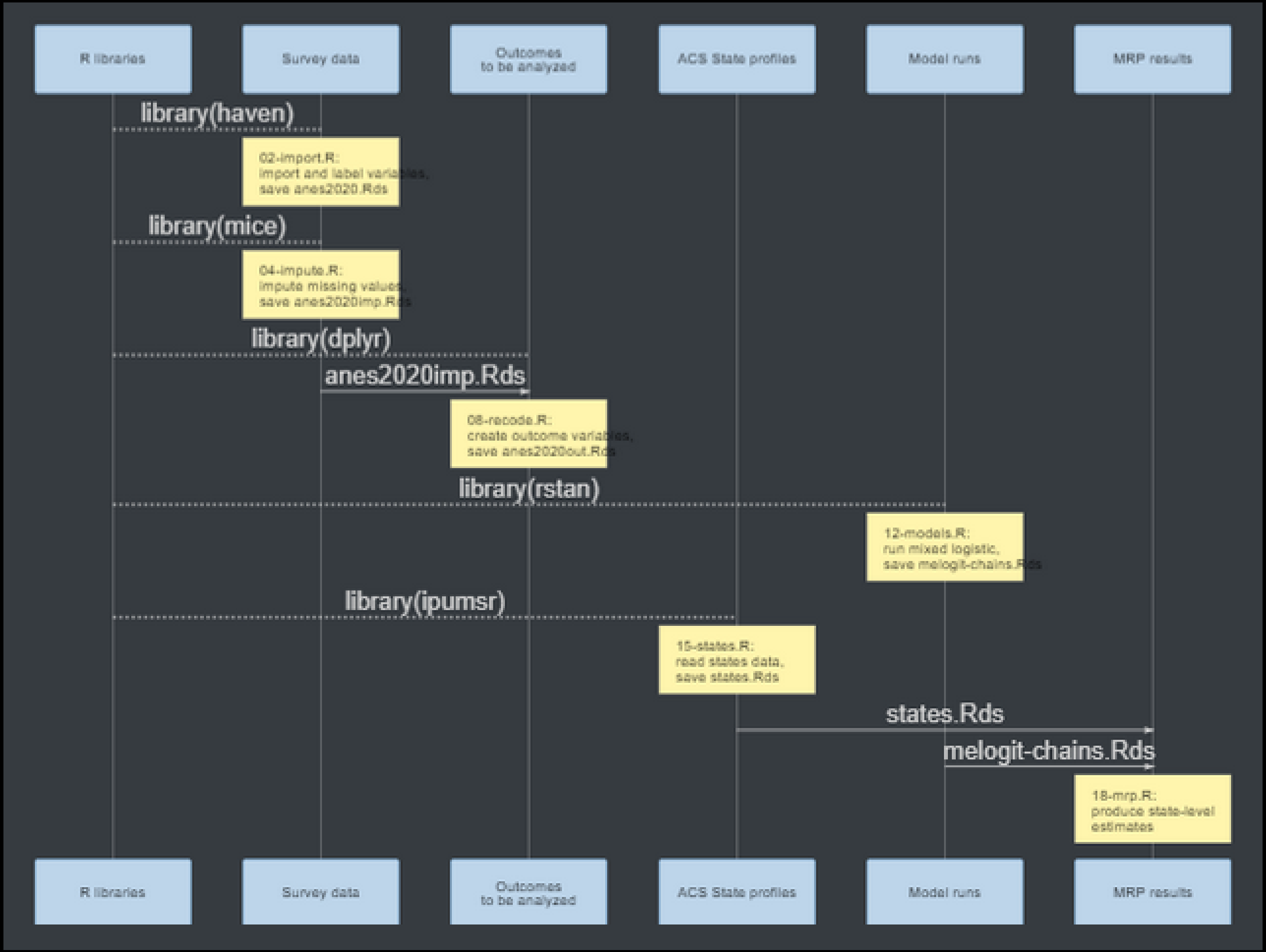


Sequence diagrams

~~~
```sequence
Alice -> Bob:      Hello Bob, how are you?
Note right of Bob: Bob thinks
Bob --> Alice:     I am good thanks!
```
~~~

Try more at https://mermaid-js.github.io/mermaid-live-editor/

Sequence diagrams


R libraries
Survey data
Outcomes to be analyzed
ACS State profiles
Model runs
MRP results
library(haven)
02-import.R: import and label variables, save anes2020.Rds
library(mice)
04-impute.R: impute missing values, save anes2020imp.Rds
library(dplyr)
anes2020imp.Rds
08-recode.R: create outcome variables, save anes2020out.Rds
library(rstan)
12-models.R: run mixed logistic, save melogit-chains.Rds
library(ipumsr)
15-states.R: read states data, save states.Rds
states.Rds
melogit-chains.Rds
18-mrp.R: produce state-level estimates
R libraries
Survey data
Outcomes to be analyzed
ACS State profiles
Model runs
MRP results

Other languages in RStudio

Python:

~~~
```r libraries
library(reticulate)
```

Do something in Python:

```python
import pandas as pd
flights = pd.read_csv('flights.csv')
```

Pass back to R:

```r
head( py$flights )
```
~~~

BOLD THINKERS DRIVING REAL-WORLD IMPACT

Other languages in RStudio

Stata:

~~~
```r stata_engine
statapath <- "C:/Program Files/Stata16/Stata-MP.exe"
library(Statamarkdown)
knitr::opts_chunks$set(engine.path=list(stata=statapath))
```
~~~

Do something in Stata (`collectcode` option makes sure results are cumulative):

~~~
```stata, collectcode=TRUE
sysuse auto, clear
```
~~~

More Stata:

~~~
```stata
scatter mpg price
```
~~~

In the spirit of reproducible reporting

BOLD
THINKERS
DRIVING
REAL-WORLD
IMPACT

Software citations

R version: R version 3.6.3 (2020-02-29).

Package versions:

- library(here): version 0.1
- library(knitr): version 1.28
- library(kableExtra): version 1.1.0
- library(flextable): version 0.5.9
- library(tidyverse): version 1.3.0
- library(xaringanthemer): version 0.3.0
- library(shiny): version 1.4.0
- library(countdown): version 0.3.3

Thanks

Stas Kolenikov (skolenik@gmail.com)

Twitter @StatStas