Reproducible Workflow: Coding Strategies and Software [75mins]

Introduction, Hands-on with Version Control (Github) and Dynamic Documents (RMarkdown)

Fernando Hoces de la Guardia
BITSS

-

Slides at https://goo.gl/aBQ3LR

Inter-American Development Bank Workshop, March 2018

File Management & Coding Suggestions (Christensen et al, 2018) [15 mins]

File Management [5 mins]

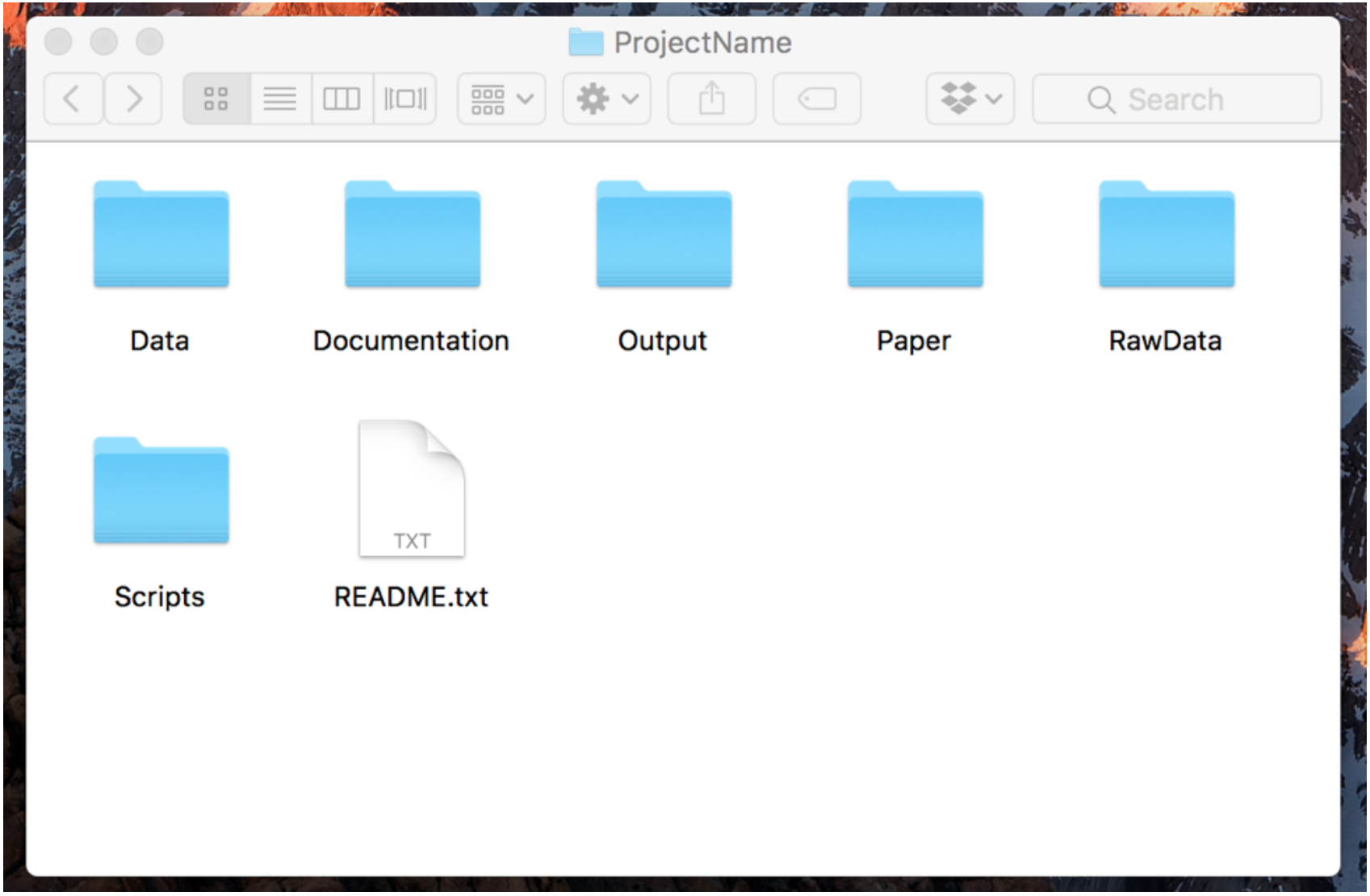


Figure 1:

Organizing Principles

1 - Use code (scripts), don't work by hand (GUI's or comand line).

2 - Consider not saving statistical output, and just saving the code and data that generates it.

3 - Reproducibility. Minimum: machine (laptop) independence. Ideal: analyst indepence.

Coding Suggestions [10 mins]

1. Include tests in your code.
2. You can never comment your code too much.
3. Indent your code.
4. Once posted, any changes at all require a new file name. Or a version control system in place.
5. Separate your data cleaning and analysis files
6. Never name a file "final" because it won't be.
7. Name binary variables "male" instead of "gender," (1=Male and 0=Not)
8. Don't leave clutter around-delete temporary or unnecessary intermediate objects.
9. Every variable should have a label.
10. Use relative directory paths (such as "./Data" and not "C:/Users/Fernando/Documents/Project/Data")

Coding Suggestions: Stata-specific

1. Accurately and concisely capture missing values. (`.` and `.a-.z`)
2. Make sure code always produces the same result, and that merging and sorting is reproducible. `duplicates report; isid; sort, stable`
3. Run simple tests to alert yourself when results change. Example:

```
count if _merge!=3
if r(N)!=74 {
display "Unmatched observations changed!"
there is an error here
}
```

4. Don't use abbreviations for variables or commands.
5. Use global macros to define directory paths so collaborators can readily work across different computers.
6. Use local macros for varlists.

Coding Suggestions: Stata-specific

7. Use computer-stored versions of numerical output (eg `r(mean)`). Use `return list` and `ereturn list`
8. If you have a master .do file that calls other .do files, which each have their own .log file capturing output, you can run multiple log files at the same time (so you end up with a master .log file)
9. Use the `label data` and `notes`.
10. Use the `notes` command for variables as well for identifying information that is too long for the variable label.
11. Validate data sources to ensure consistency. Use `datasignature` on auto data set (`sysuse auto.dta`, then `datasignature set` should give you this number: `74:12(71728):3831085005:1395876116`)
12. Use value labels for all categorical variables. `numlabel [lblname-list], add command.`
13. Don't use capital letters in variable names.
14. Make your files as non-proprietary as possible (use the `saveold` command)

Version Control [30 mins]

Problem to avoid

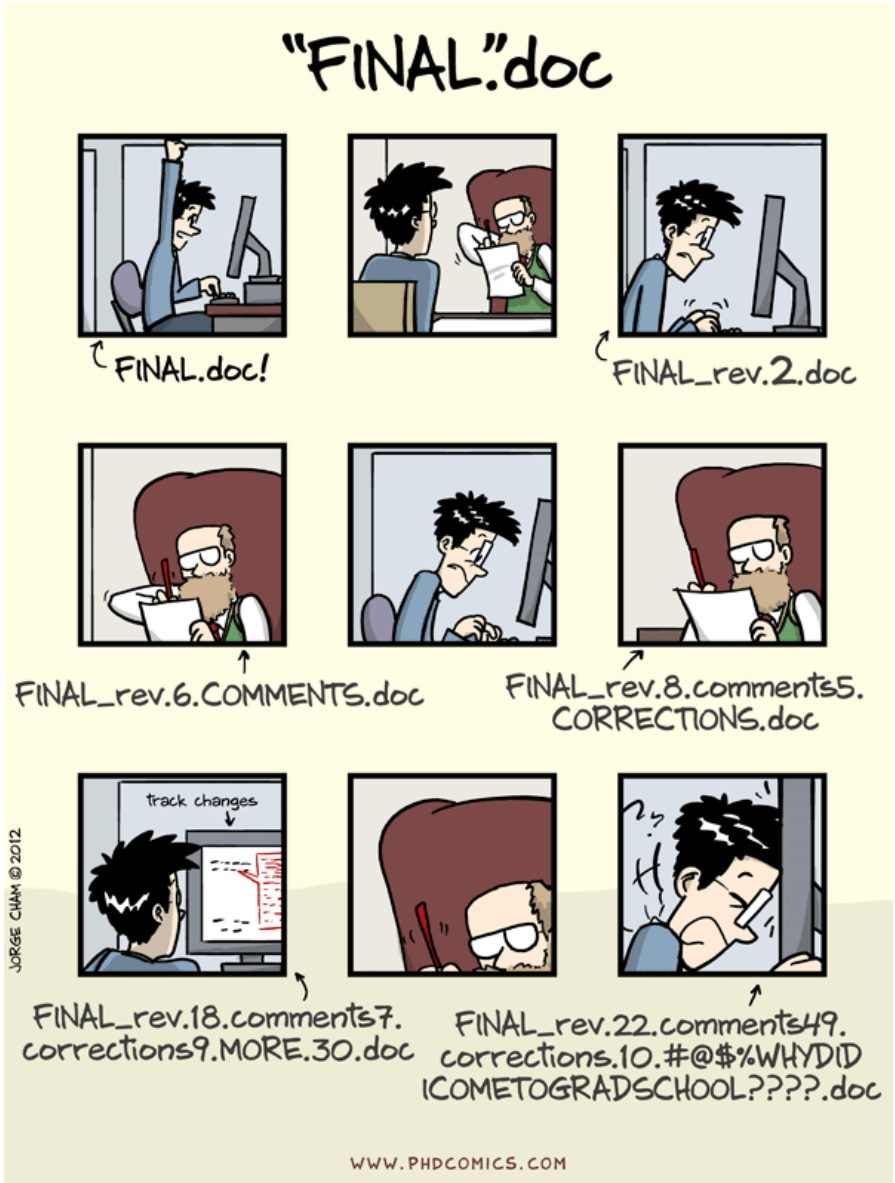


Figure 2: http://www.phdcomics.com/comics/archive/phd101212s.gif

Managing expectations

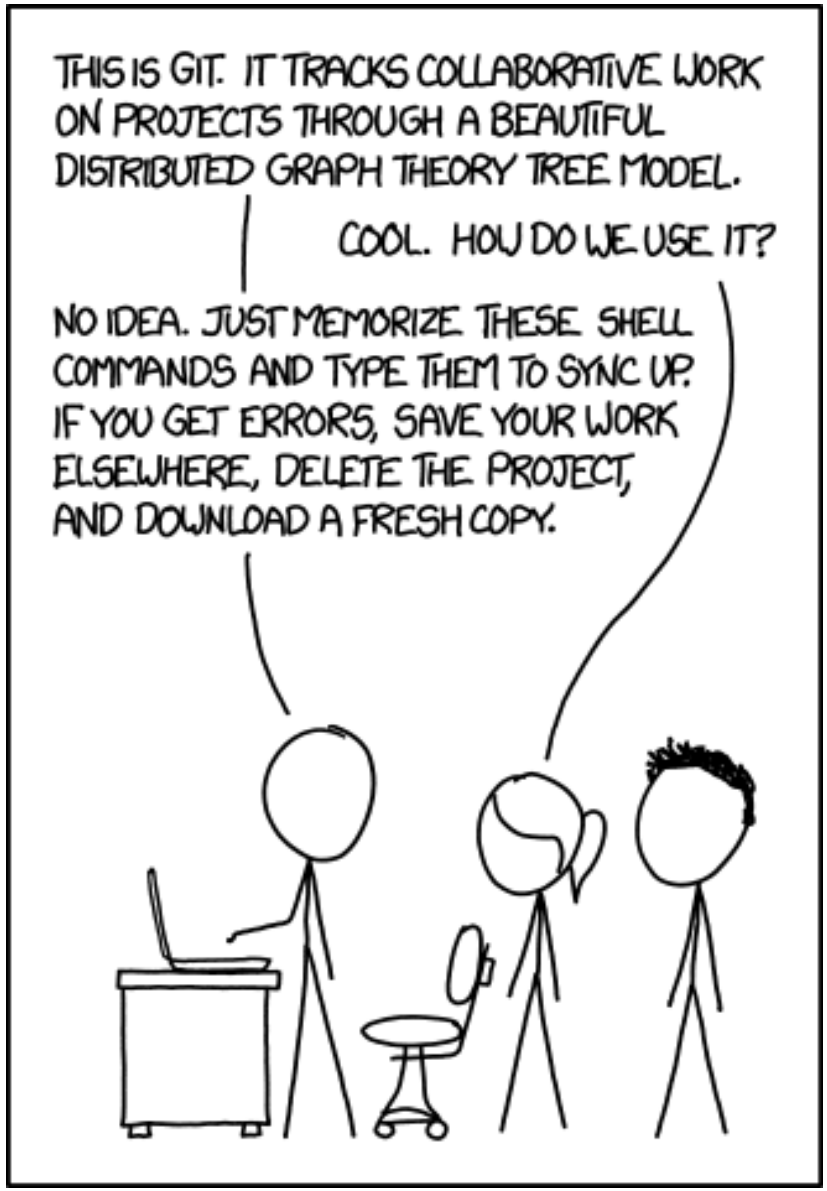


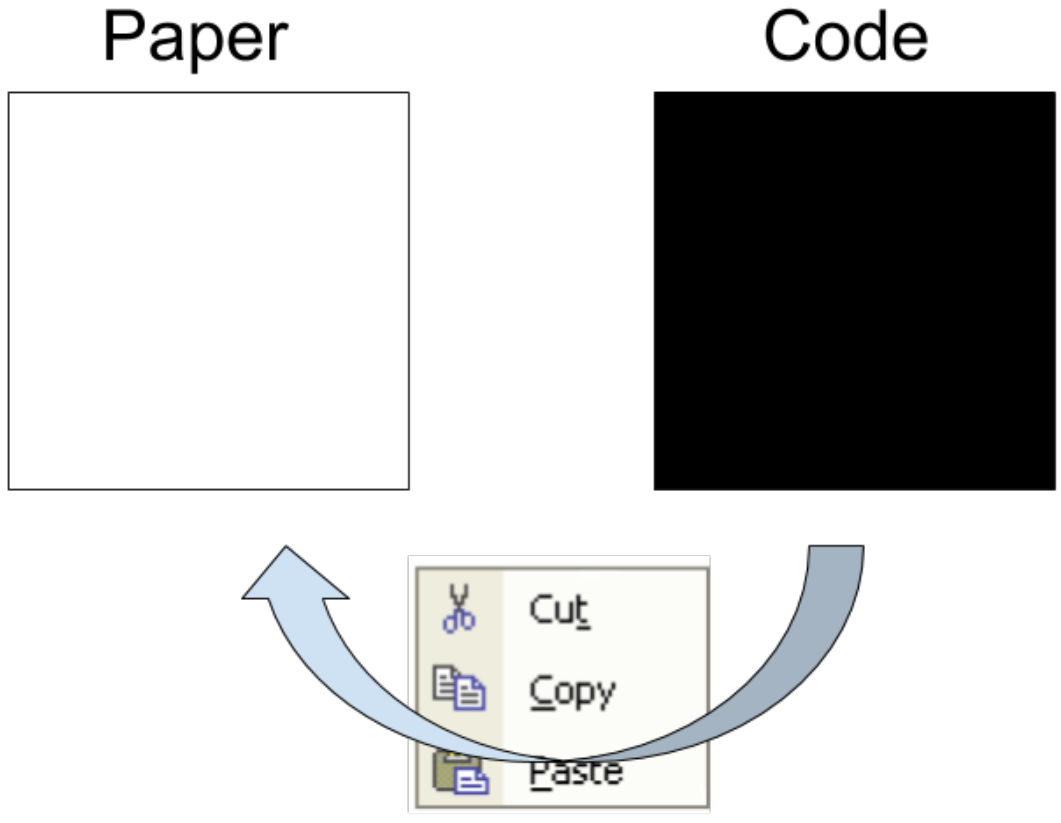
Figure 3: Git xkcd comic

Dynamic Documents [30 mins]

Dynamic Documents For Computational Reproducibility

- Based on principles of *literate programming* aims at combining code and paper in one single document
- Best framework to achieve the holy grail of **one-click reproducible workflow**
- Best two current implementations: `RMarkdown (R)` & `Jupyter (Python)`. `Stata` is catching up (more at the end)

Currently code and narrative components live in separate universes



Dynamic Documents: integrate the two universes!

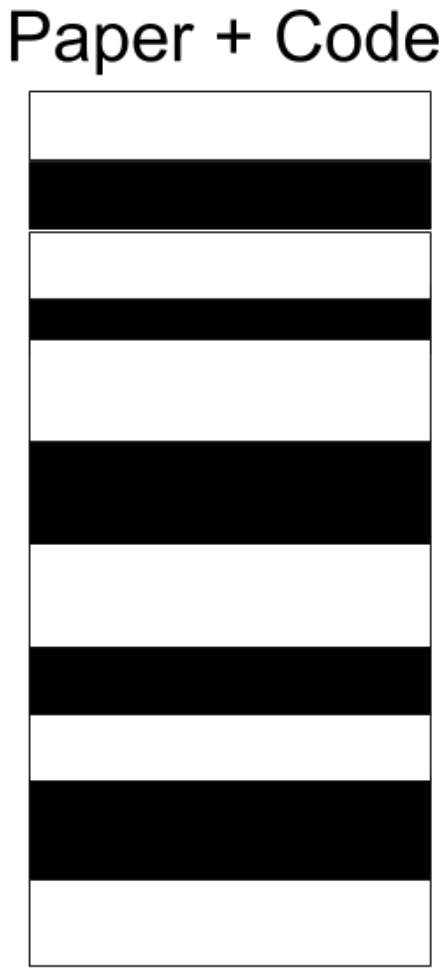


Figure 5:

Dynamic Documents: A Recipe

- 1 simple language that can combine text and code: `Markdown`
- 1 statistical package to do the analysis (`R`, `Python`, `3S's?`)
- 1 machinery to combine analysis and text to create a single output: `Pandoc`
- [Optional-but-not-really] 1 program to bring all the elements together: RStudio/RMarkdown, Jupyter

Markdown laguange/syntax in 60 seconds

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):
```

becomes

Plain text
End a line with two spaces to start a new para
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image: 

Figure 6:

One Type of Dynamic Document: R Markdown

For our excercise: R Markdown

- R: **open source** programming language design for statistical analysis.
- RStudio: free software that provides and Integrated Development Environment (IDE)
- RStudio combines all together: R + Markdown + Pandoc to produce multiple outputs

R Markdown

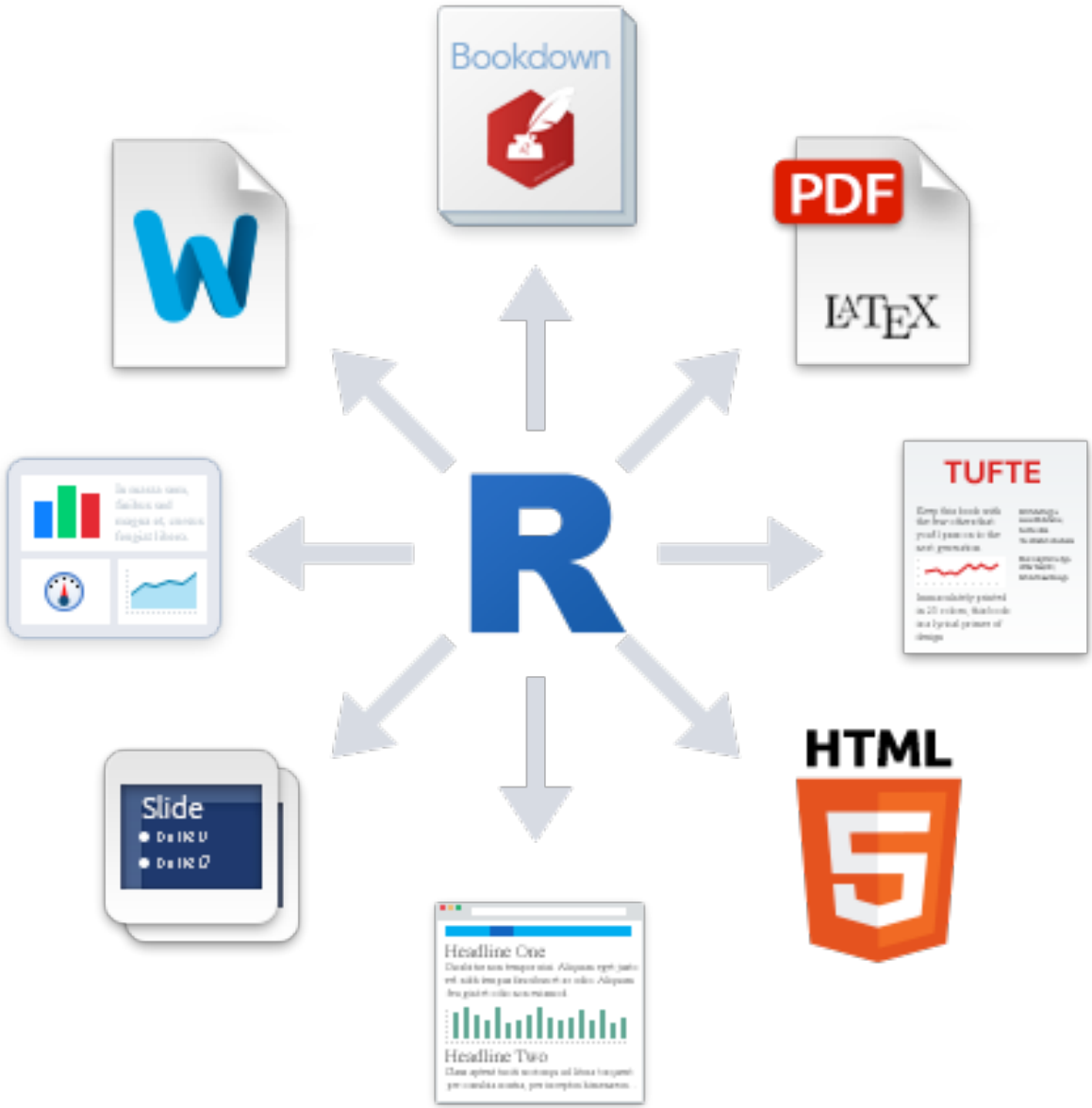


Figure 7:

Basic Structure

- A header
- Text
- Code: inline and chunks

Basic Structure: Header

```
---
title: "Sample Paper"
author: "Fernando Hoces de la Guardia"
output: html_document
---
```

Basic Structure: Body of Text

```
---
header
---
```

This is where you write your paper. Nothing much to add. You can check Markdown syntax here. And it can use can type equations using LaTex syntax!

Basic Structure: Code Chunks and Inline

```
---
header
---
```

Body of text.

To begin a piece of code ("code chunk"). Enclose them in the following expression (Ctrl/Cmd + shift/optn + i)

```
```{r, eval=TRUE}
here goes the code
```
```

To write inline use only one Backtick to open followed by an "r"" and one to close `r 1+1` in the output.

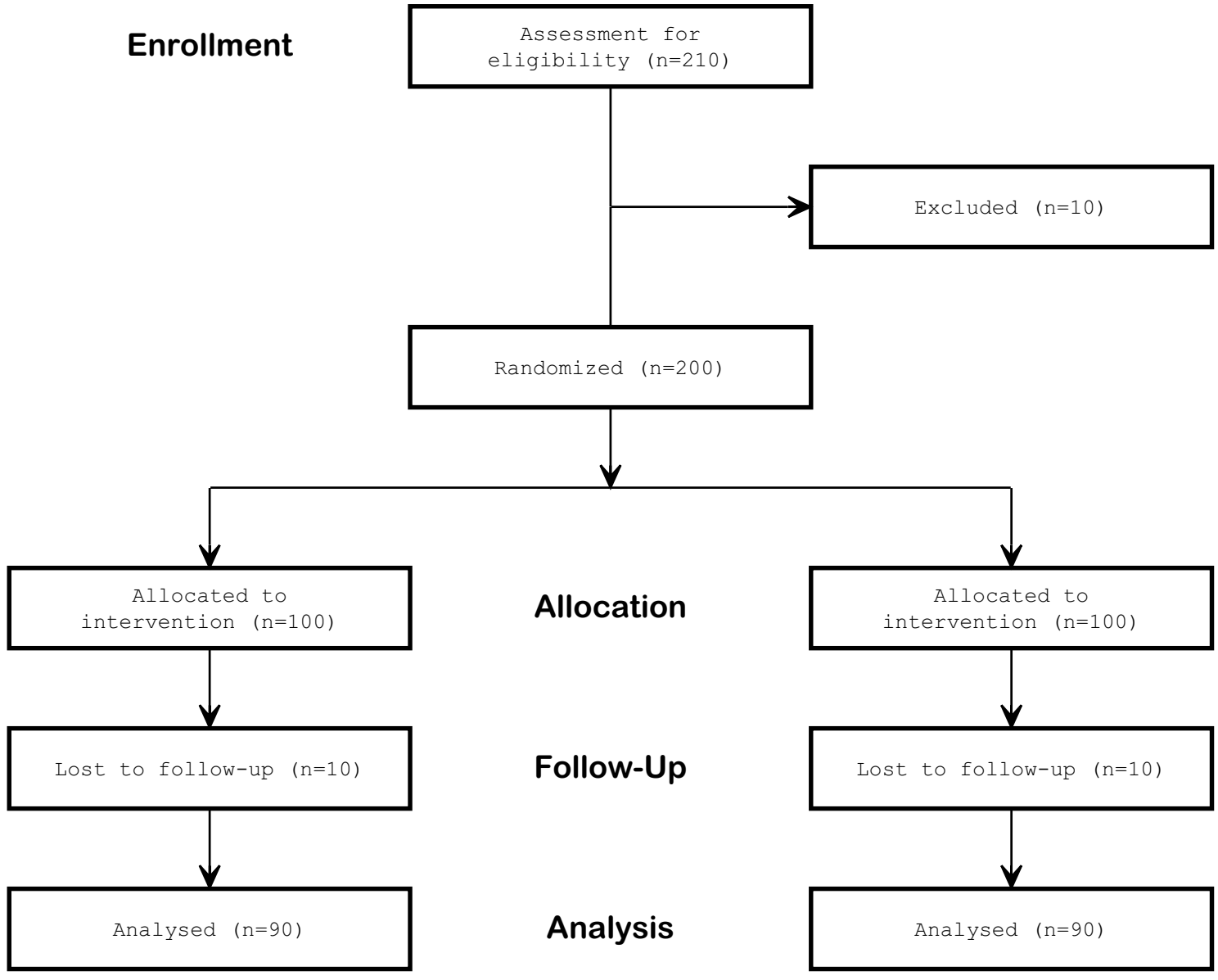
Hands-on!

```
## Parsed with column specification:
## cols(
##   Timestamp = col_character(),
##   `ID number` = col_integer(),
##   `Dollar value to question #1` = col_integer(),
##   Gender = col_character(),
##   Education = col_character(),
##   `Years of working experience` = col_character(),
##   `Dollar value to question #2` = col_integer()
## )

##
## 0 1
## 6 4

##
## Call:
## lm(formula = `Dollar value to question #1` ~ treatment,
```

CONSORT diagram of our little experiment



Balance of covariates

Estimated effect

The Stata version of all of the above:

Additional Resources

Garret, Ted and Jeremy's book

The Practice of Reproducible Research

Code and Data for the Social Sciences

The Workflow of Data Analysis Using Stata

Reproducible Research with R and R Studio

Project TIER