Dynamic Documents using R: Hands-On

CEGA EASST Fellows Training

Fernando Hoces de la Guardia, BITSS

24 September 2021 | slides

Before we begin

- Clone/download this repo
- If you are having issues of bandwith, clone/download this repo instead

Dynamic Documents For Computational Reproducibility

- Based on principles of *literate programming* aims at combining code and paper in one single document

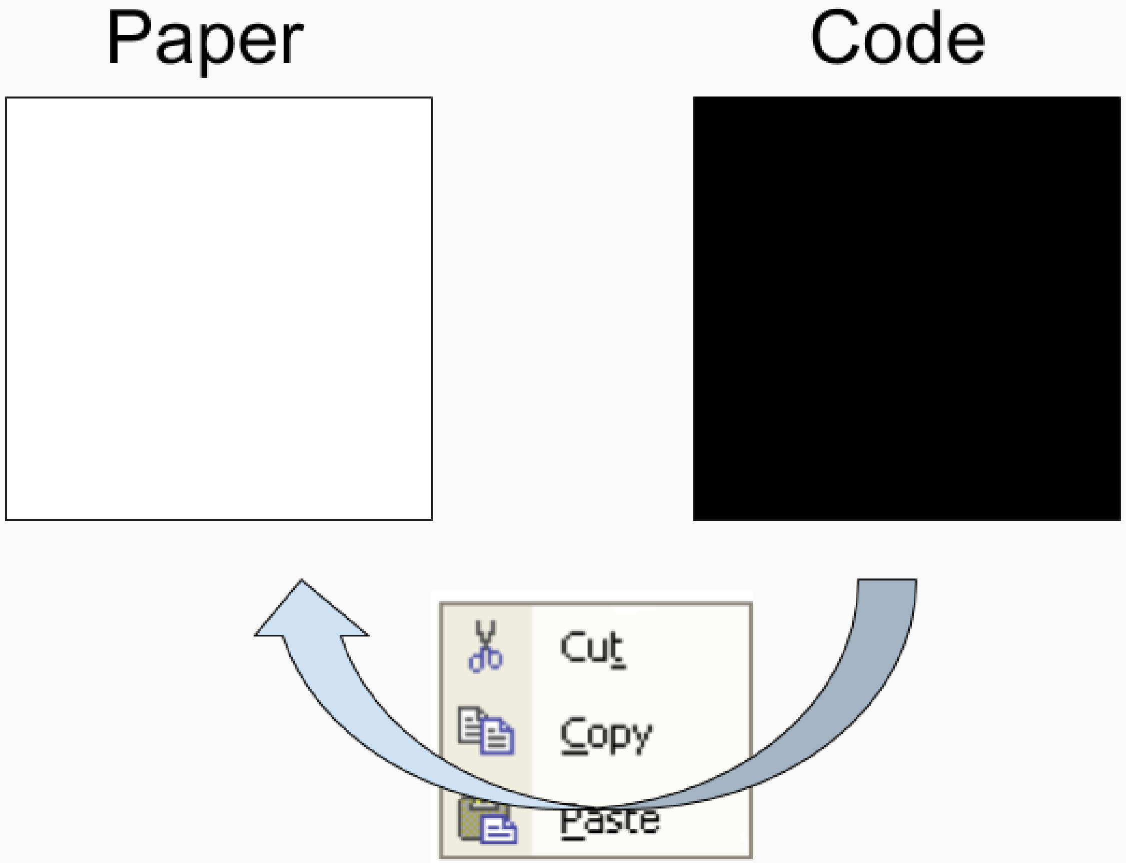
- Best framework to achieve the aspirational goal of **one-click reproducible workflow**

- Best two current implementations: `RMarkdown (R)` & `Jupyter (Julia, Python)`. `Stata` is catching up (dyndocs release here and reviews here and here)

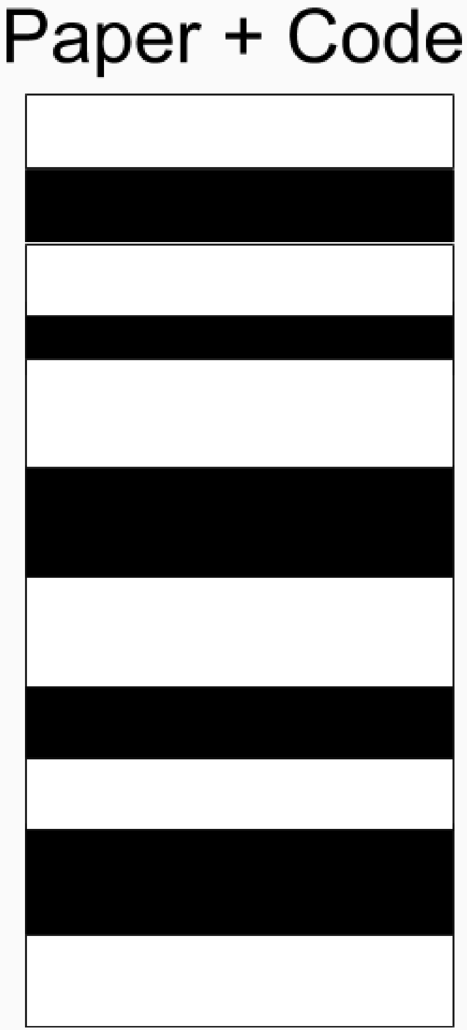
"Bro-pen" Science Warning Ahead!

- Dynamic docs are useful, but they are not essential for reproducibility
- They are pack with little quirks that takes a while to master
- Within DD 90% of the values comes from the most basic (default) output. All the other "cool and easy" feature can be not so easy and make you suffer from the "shiny new exit" phenomena

Currently code and narrative components live in separate universes



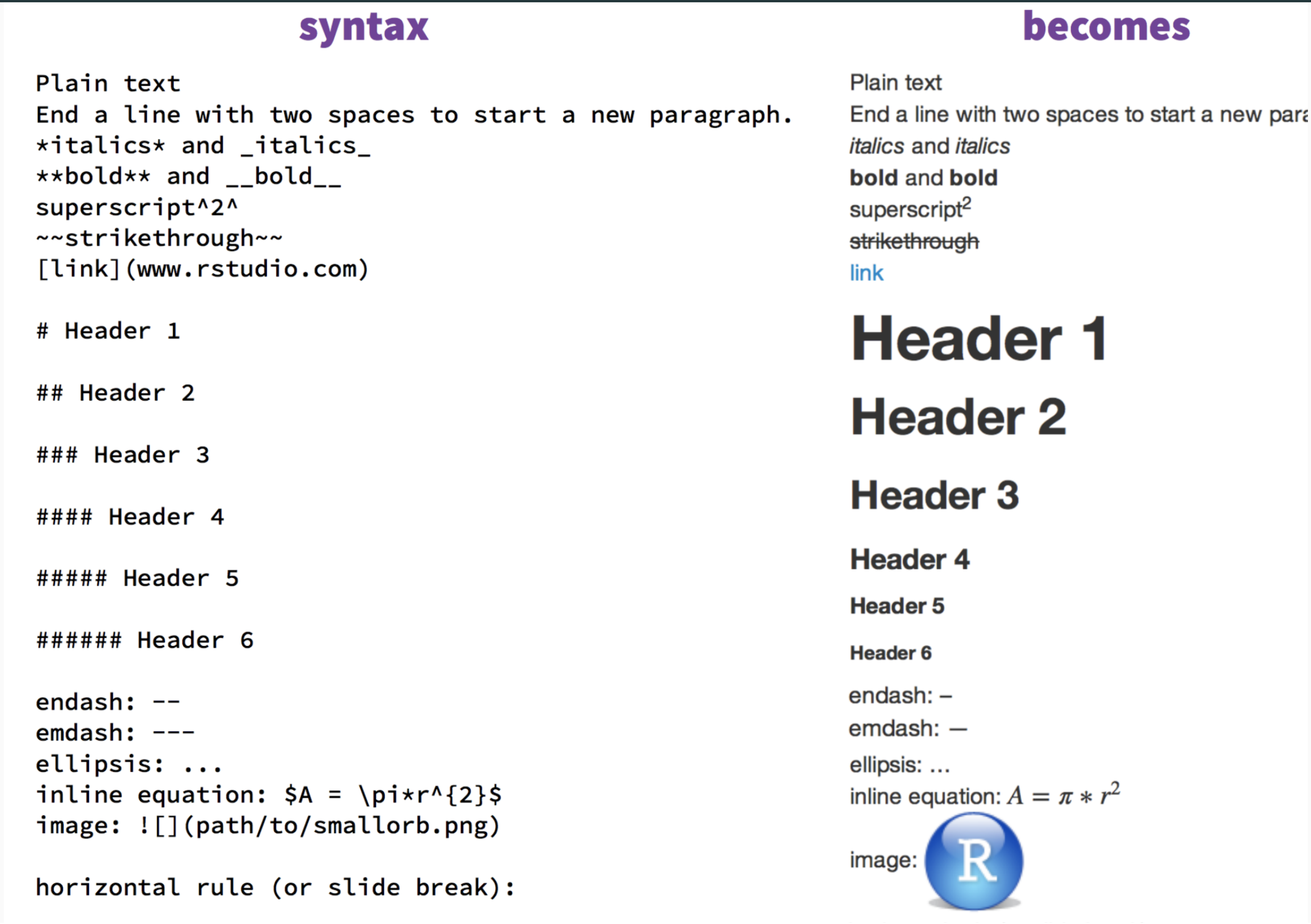
Dynamic Documents: integrate the two universes!



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

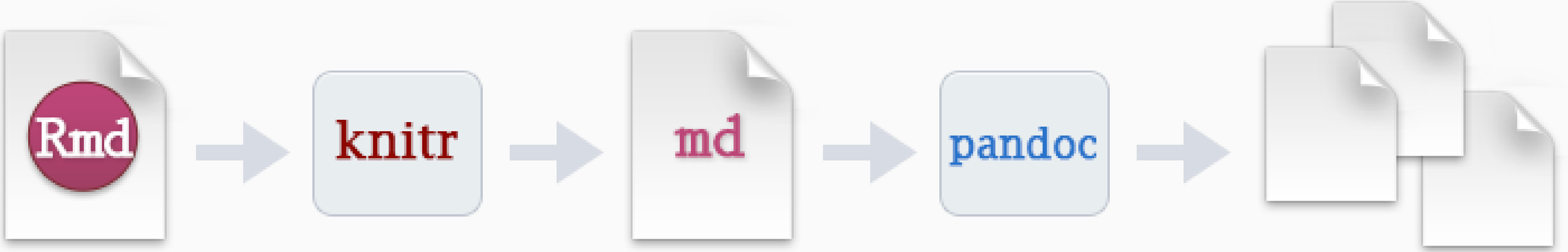
Header 6

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

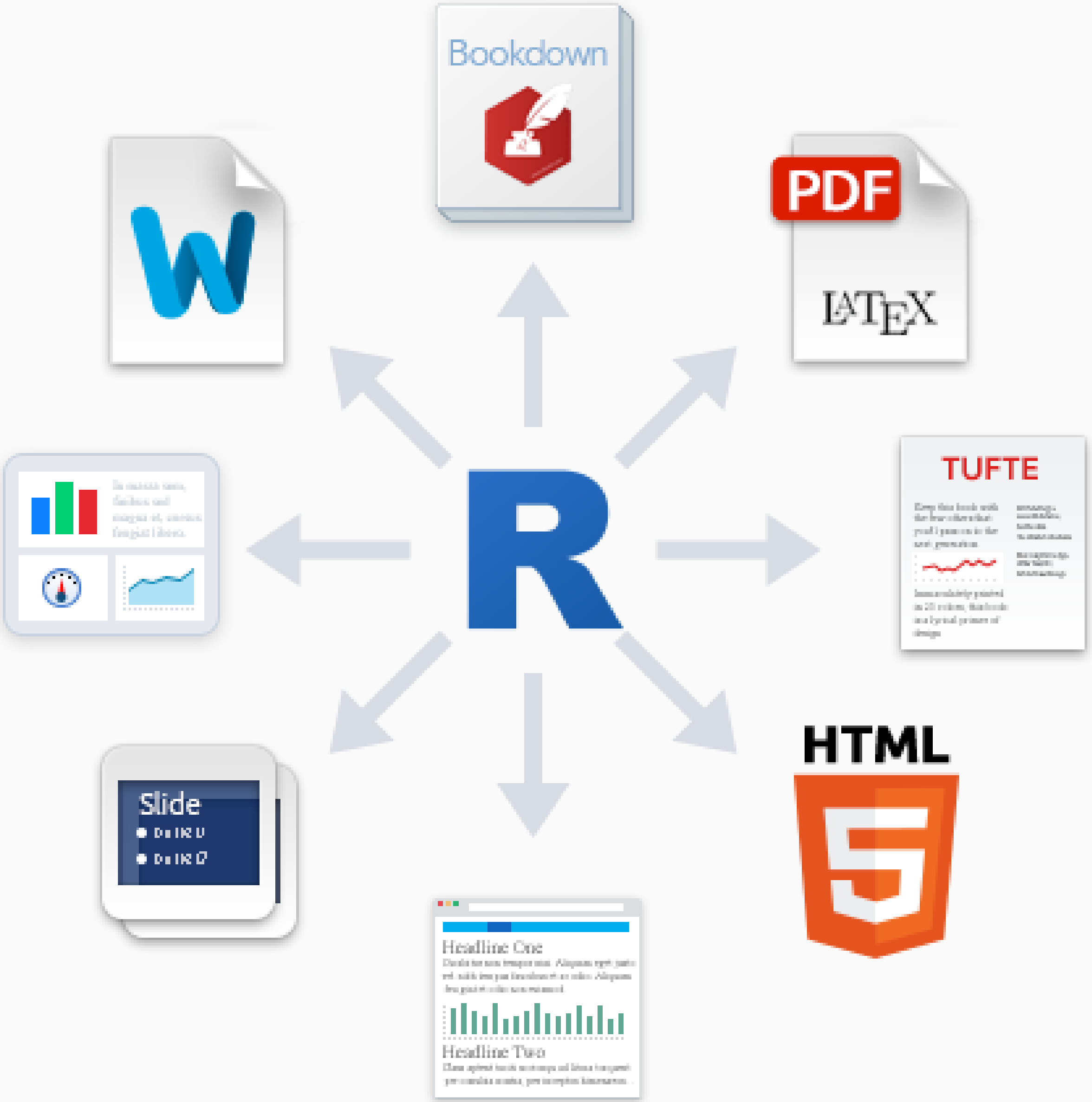
helpful editor
to get you
started

One Type of Dynamic Document: 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


Bookdown
W
PDF
LaTeX
R
TUFTE
HTML
5
Slide
Headline One
Headline Two

Basic Structure

- A header
- Text
- Code: inline and chunks

Basic Structure: Header

- Controls document-level charchterisitics:
  - type of output (.html, .pdf, etc)
  - links to other docs (appeareance, bibliography)
  - others
- YAML language
- Define by "`---`"

```
---
title: "Sample Paper"
author: "Fernando Hoces"
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!

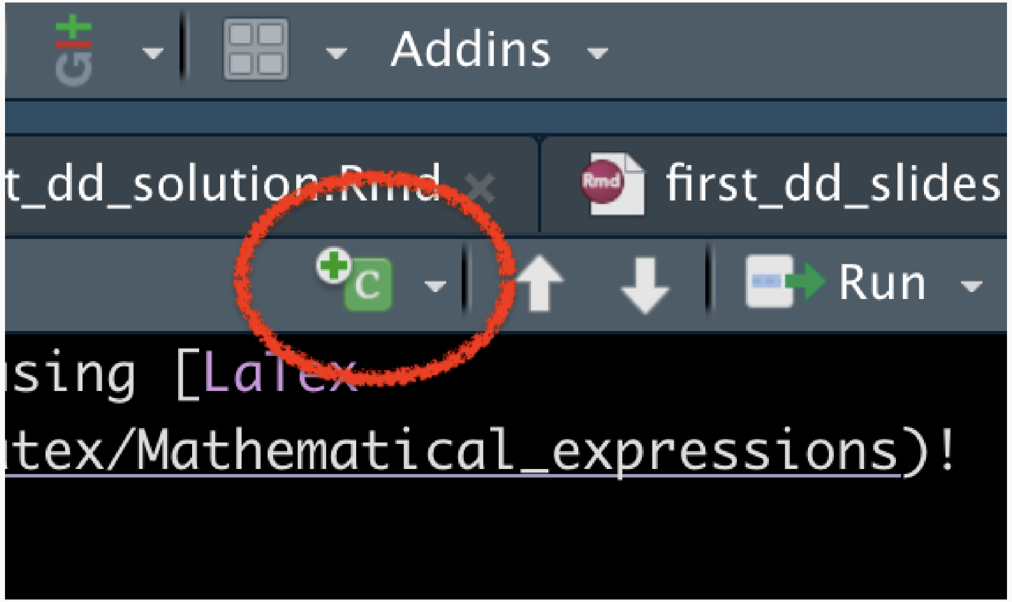
Basic Structure: Code Chunks and Inline

```
---
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
```
~~~



```
sing [LaTex
tex/Mathematical_expressions)!
```

To write inline use only one Back-tick to open followed by an "r" and one to close.

Practical Excercise Goals

Primary Goals:

1. Become familiar with your first DD.
2. Compile an empty (or default) DD into multiple formats.
3. Edit a DD with some narrative, some code (in R) and some math (optional).
4. Present all the results dynamically into multiple outputs.

Secondary Goal:

1. Expose you to some R programming.
2. Entertain you with a fun problem.

Hands-on excercise: the birthday problem!

As an illustration lets write a report using the participants in this workshop to illustrate the famous birthday problem.

> What is the probability that at least two people this room share the same birthday?
>
> There are 7 in this room.
>
> Is it something like $\frac{7}{365} = 0.019$?

Create a new RMarkdown File

1 - In RStudio: `File→ New File → RMarkdown ...`

2 - Name it, and save it as `/dynamicdocs/first_dd.Rmd`.

3 - Review/edit the header, and delete all the default body of text except for one code chunk.

4 - In that chunk define a seed (`set.seed(1234)`, number of people in the room (`n.pers = ?`), and this other variable:`n.sims = 10000` (will explain later)

5 - Below the first chunk, write down a title (using `#`) and a brief description.

The birthday problem: the math

Actually the math says otherwise. Define p(n) as the probability that at least one pair has the same birthday, then the $1 - p(n)$ is the probability that all are born in a different day. Which we can compute as:

$$
\begin{aligned}
1 - p(n) &= 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{n-1}{365}\right) \\
&= \frac{365 \times 364 \times \cdots \times (365 - n + 1)}{365^n} \\
&= \frac{365!}{365^n (365 - n)!} = \frac{n! \cdot \binom{365}{n}}{365^n}
\end{aligned}
$$

$$p(n = 7) = 0.056$$

Code for the math

(`/dynamicdocs/first_dd_solution.Rmd`)

Copy and paste lines below into your `first_dd.Rmd`

```
\begin{align}
 1 -  p(n) &= 1 \times \left(1-\frac{1}{365}\right) \times
              \left(1-\frac{2}{365}\right) \times \cdots \times
              \left(1-\frac{n-1}{365}\right) \nonumber  \\
           &= \frac{365 \times 364 \times \cdots \times (365-n+1) }{365^n} \nonumber \\
           &= \frac{ 365! }{ 365^n (365-n)!} = \frac{n!\cdot\binom{365}{n}}{365^n}\\
\p(n= `r n.pers`) &= `r  round(1 - factorial(n.pers) *
                           choose(365,n.pers)/ 365^n.pers, 3)`  \nonumber
\end{align}
```

Don't like math? Let's run a simple simulation!

1 - Simulate 10^{4} rooms with $n = 7$ random birthdays, and store the results in matrix where each row represents a room.

2 - For each room (row) compute the number of unique birthdays.

3 - Compute the average number of times a room has 7 unique birthdays, across 10^{4} simulations, and report the complement.

Code for the simulation

(`/dynamicdocs/first_dd_solution.Rmd`)

```r
birthday.prob = function(n.pers_var, n.sims_var) {
  # simulate birthdays
  birthdays = matrix(round(runif(n = n.pers_var * n.sims_var, min = 1, max = 365) ),
                     nrow = n.sims_var, ncol = n.pers_var)
  # for each room (row) get unique birthdays
  unique.birthdays = apply(birthdays, 1,
                           function(x)  length( unique(x) ) )
  # Indicator with 1 if all are unique birthdays
  all.different = 1 * (unique.birthdays==n.pers_var)
  # Compute average time all have different birthdays
  result = 1 - mean(all.different)
return(result)
}

bp_sim = birthday.prob(n.pers_var = 21, n.sims_var = 10000)
print(bp_sim)
```

```
## [1] 0.4365
```

Results

- Many people originally think of a prob ~ $\frac{1}{365} \times n = 0.019$
- However the true probability is of $p(n = 7) = 0.056$
- And the simulated probability is of 0.4365

Let's try some other outpus

Output	Additional Step?	Loss	Share	Others
Simple HTML	no	none	one click (RPubs) or "index method" in github	cheatsheet
Word	no	equations, interactivity	attachment	help

PDF	no	interactivity	attachment	requires LaTex
				`install.packages('tinytex')`
				`tinytex::install_tinytex()`

Slides	New File	some diff in syntax	one click / index method	HTML or PDF (with similar characteristics as above)
Book	many	-	index method	`install.packages('bookdown')` book

- Other outputs not covered: Apps (`shiny`), blogs (`blogdown`), packages (`pkgdown`), richer webpages (`pagedown`), and dashboards (`flexdashboard`)
- Explore templates and install `rticles` for more!

Additional Steps for Book Output

1 - Create a new project: `File→ New Porject ... → New Directory → Book Project using bookdown`

2 - Place project in independent folder and then convert into a git repo: `Tools→Project Options`, choose git, then use GitHub app to publish repo.

3 - Place each chapter into a new Rmd, and enumerate them

4 - Go to top-right pane click on `Build → Build`

5 - (preparation to share) In the file `_bookdown.yml`, add the following lines

- `repo: https://github.com/YOUR-USER-ID/first_book_solution`
- `output_dir: docs`
- `edit: https://github.com/YOUR-USER-ID/first_book_solution/edit/master/%s`

6 - Modify latex engine in code chunk (line 20 of chapter 2) from `latex` to `md` 6 - Build the book again.

7 - To share: push repo into github.com, then go to the repo's URL, then `settings`, `pages`, select `master` branch, select `docs` folder and save

Here is the repo with the solutions for the book on birthday problem

Practical Excercise #2: Mostly Harmless Econometrics!

There is a fantastic Github repo that is reproducing results from MHE

Lets use the of examples Figure 5.2.4 to show how dynamic docs can be used in data analysis.

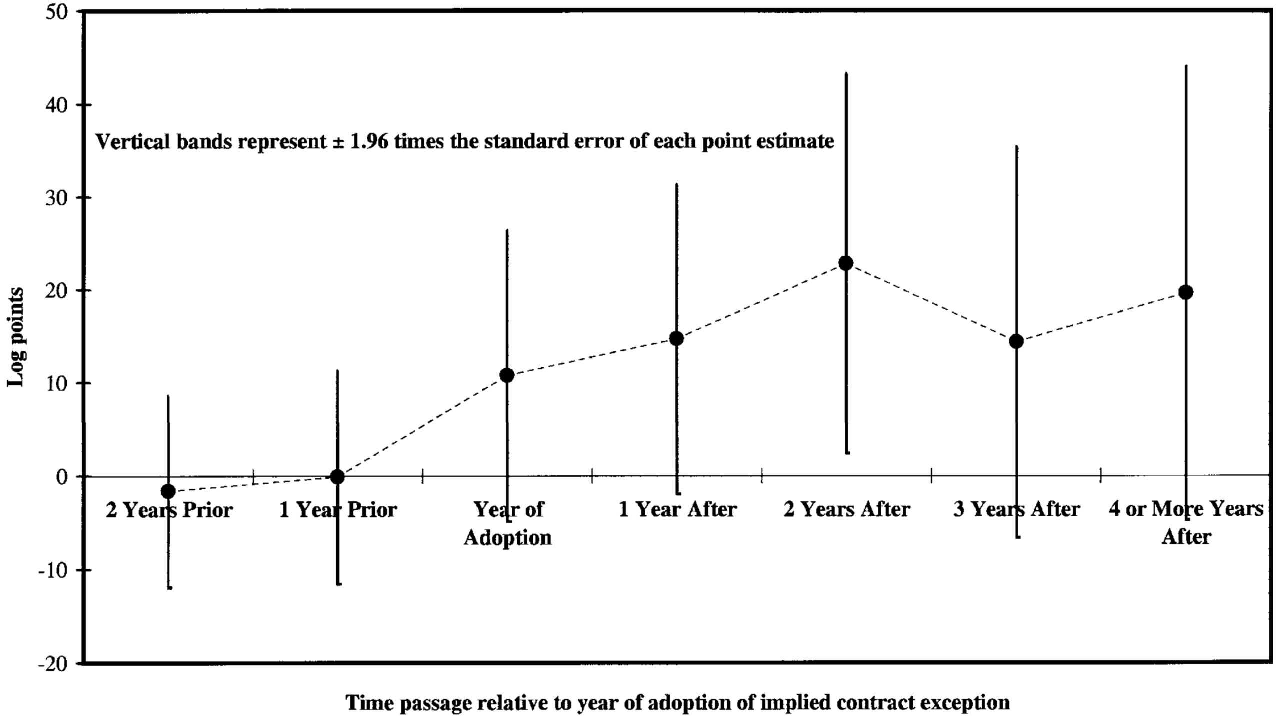


FIG. 3.—Estimated impact of implied contract exception on log state temporary help supply industry employment for years before, during, and after adoption, 1979–95.

Goals for excercise #2

Primary Goals:

1 - Demonstrate how the **entire workflow** of a study can fit into a DD.

2 - Show how to add options to the header.

3 - Demonstrate how a DD make code readable to non-coders.

Secondary Goal:

1 - Expose you to some R programming.

Instructions to get started with excercise #2:

1 - Create a new blank `.Rmd` file (steps 1 - 3 in from previous ex.)

2 - Save it as `/dynamicdocs/Figure 5-2-4.Rmd`

3 - Look at this code behind figure 5.2.4.

4 - Start building your own DD to describe what this code does.

We will go step by step using `/dynamicdocs/Figure 5-2-4_solutions.Rmd` as back-up.

Description

- Begin a new section (`##`), titled "Description"
- Write a brief description of our goal in the DD.
- You might want to insert a reference to the paper: link here.
- Specific content not so relevant, just refer to "a treatment" and "a outcome".

Getting the raw data

- Begin a new section (`##`), titled "Raw Data".
- Describe what you will do.
- Create two code chunks:

~~~
```{r download data, eval=FALSE, echo=TRUE,
warning=FALSE, results='hide', message=FALSE}
here goes the code
```
~~~

Cleaning the data

- Begin a new section (`##`), titled "Data Cleaning".
- Describe what you will do:
  - Restrict sample to years between 1979 and 1995 (inclusive)
  - Guam from the sample (state = 98).

- Create one code chunk:

~~~
```{r data cleaning, echo=TRUE}
here goes the code
```
~~~

- Add some description on the data (using dynamic reporting). See solutions (Figure `5-2-4_solutions.Rmd` line 58) for examples.

Build the analytic file

- Begin a new section (`##`), titled "Build analytic file".
- Describe what you will do.
- We need to construct the following variables:
  - Log of total employment
  - Normalize the year variable to 1978
- Insert a new code chunk:

~~~
```{r analytic file, echo=TRUE}
here goes the code
```
~~~

Describe the model to estimate (optional)

- Begin a new section (`##`), titled "Define model to estimate".
- One line describing what we want to estimate (i.e. "We want to estimate a fixed effect model with lead and lag treatment variables").
- A mathematical model that represents the equation to be estimated (look at solutions).

Vizualize the results (optional)

- Begin a new section (`##`), titled "Vizualize the results".
- One line describing what we want to estimate (i.e. "This estimates are then used to create figure 3 of the original paper, which is figure 5.2.4 in MHE.").

~~~
```{r viz}
here goes the code
```
~~~

Practical Excercise #2

- Run your version into multiple outputs.
- Run the solutions version into multiple outputs.
- Compare document with original version of the code.

Bonus truck: NBER Working Papers!

- Remeber the example from yesterday on the half bake analysis of NBER papers in github?
- Fork and clone the repo one more time: github.com/fhoces/nber_trends
- Now knit the `.Rmd` file instead.

Final Remarks & More Resources

- With DD we can get closer to a one-click reproducible workflow.
- Helps to generate 95% of final output, but if you cannot customize your output 100% do the last 5% in the final desired output (word, latex, etc).
- Large amount of great tutorials on the web (check out the most recent one from RStudio).
- Great examples: blog posts, and so, many, books (not many papers!).
- Want to learn more: great free books (can you guess how they were written?)