Dynamic Documents for Your Research Workflow

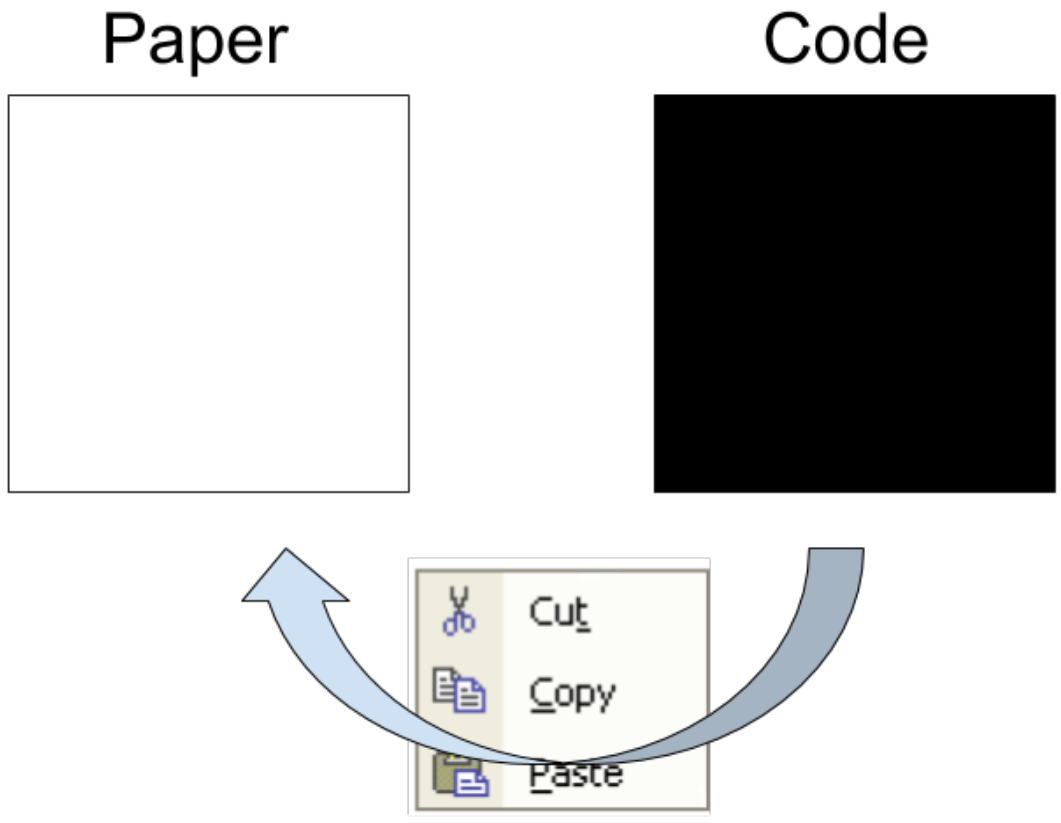
Fernando Hoces de la Guardia
BITSS

BITSS Annual Meeting, December 2017

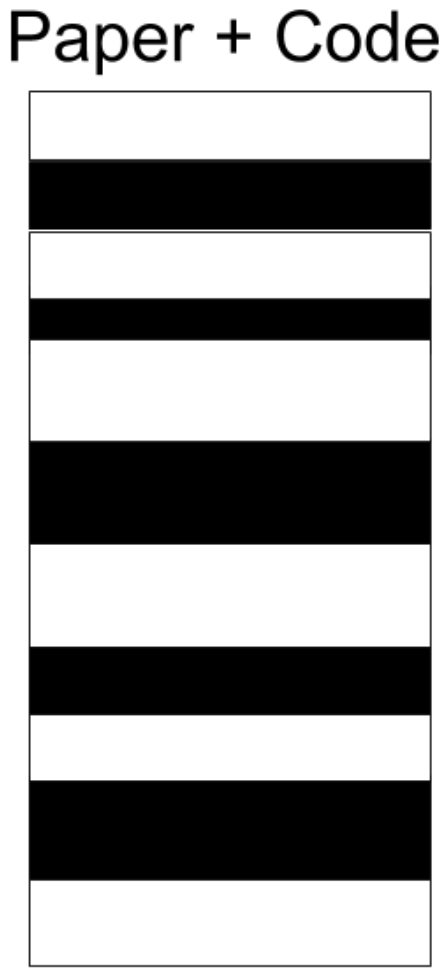
Dynamic Documents for computational reproducibility

- 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 cathching up (more at the end)

Currently code and narrative components live in separate universes



Dynamic Documents: integrate the two universes!



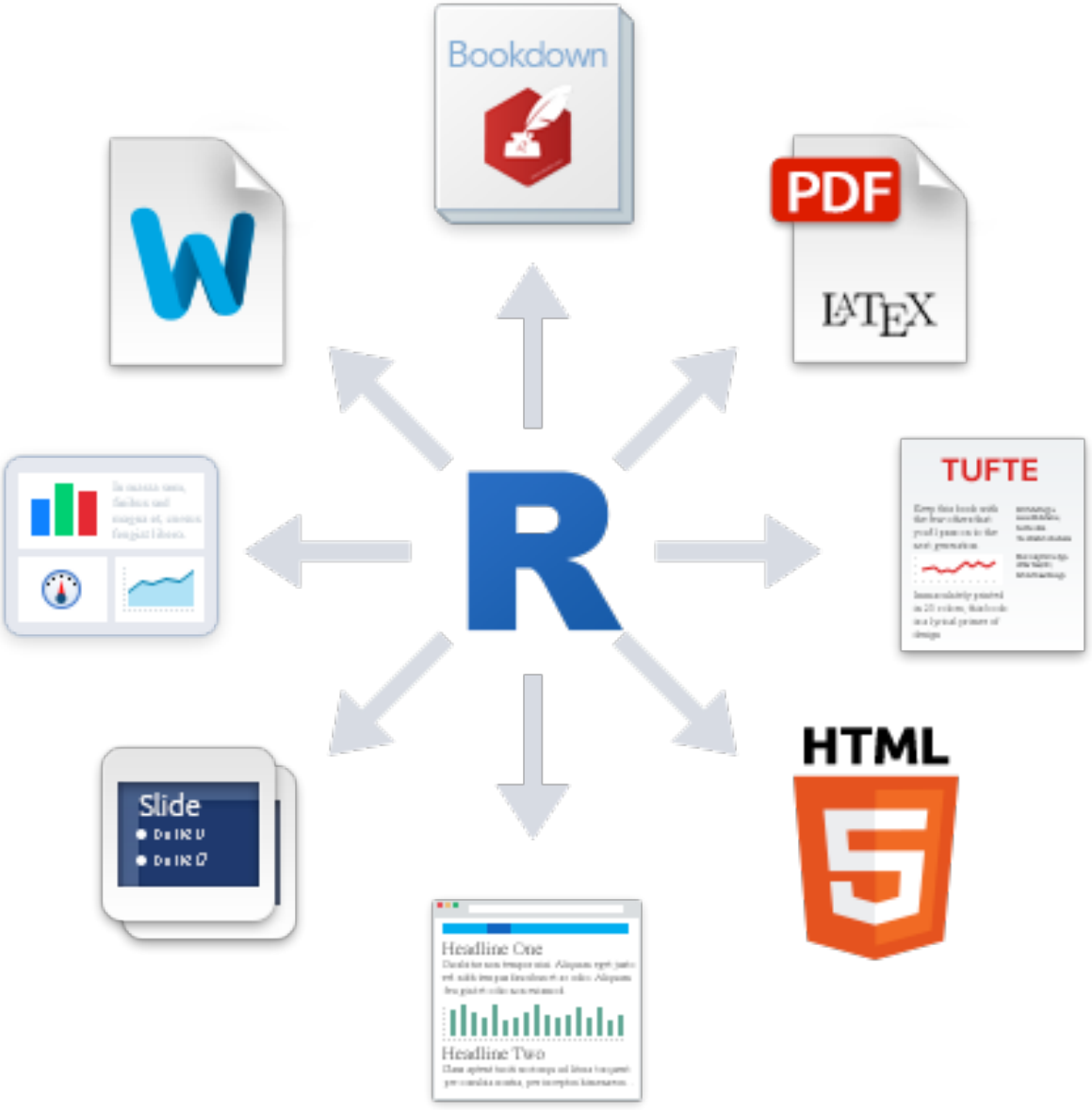
Dynamic Documents: A Reciepe

- 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

For our excercise: R Markdown

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

R Markdown


Bookdown
PDF
LATEX
TUFTE
R
HTML
Slide
Headline One
Headline Two

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. For example, we could use the people in this workshop to illustrate the famous birthday paradox. What is the probability that at least two people this room share the same birthday?

Is it something like $\frac{1}{365} \times N = 0.11$?

Basic Structure: Body of Text

```
---
The Birthday Problem
---
```

Actually the math says otherwise:

$$
\begin{aligned}
1 - \bar{p}(n) &= 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{n-1}{365}\right) \\
&= \frac{365 \times 364 \times \cdots \times (365 - n + 1)}{365^n} \\
&= \frac{365!}{365^n (365 - n)!} = \frac{n! \cdot \binom{365}{n}}{365^n} \qquad (1) \\
p(n = 40) &= 0.891
\end{aligned}
$$

Basic Structure: Code

Don't believe me? Let's run a simple simulation.

1 - Generate a 10,000 simulated rooms with $n = 40$ random birthdays, and store the results in matrix where each row represents a room.

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

3 - Compute the average number of times a room has 40 different birthdays, across 10,000 simulatinos, and report the complement.

```
n.pers = 40
birthdays = matrix(round(runif(n.pers * 1e4, 1, 365)), nrow
unique.birthdays = apply(birthdays, 1, unique)
all.different = (lapply(unique.birthdays, length) == 40)
result = 1 - mean(all.different)
print(result)
```

```
## [1] 0.8872
```

Slide with R Output

Slide with Plot