Escape the Land of LaTeX / Word for Statistical Reporting

Yihui Xie, RStudio Inc
2015/10/28

The Ecosystem of R Markdown

About me

- http://yihui.name
- first language: Chinese
- second language: R (10 years)
- third language: English
- graduated from Iowa State Univ (Stats, 2013)
- software engineer at RStudio
- (co-)author and maintainer of some R packages (e.g. animation, knitr, cranvas, formatR, testit, highr, Rd2roxygen, fun, servr, tikzDevice, shiny, evaluate, markdown, DT, leaflet)
- initiated the Chinese R Conference in 2008 (8th this year)
- Capital of Statistics (http://cos.name)

Statistical reporting

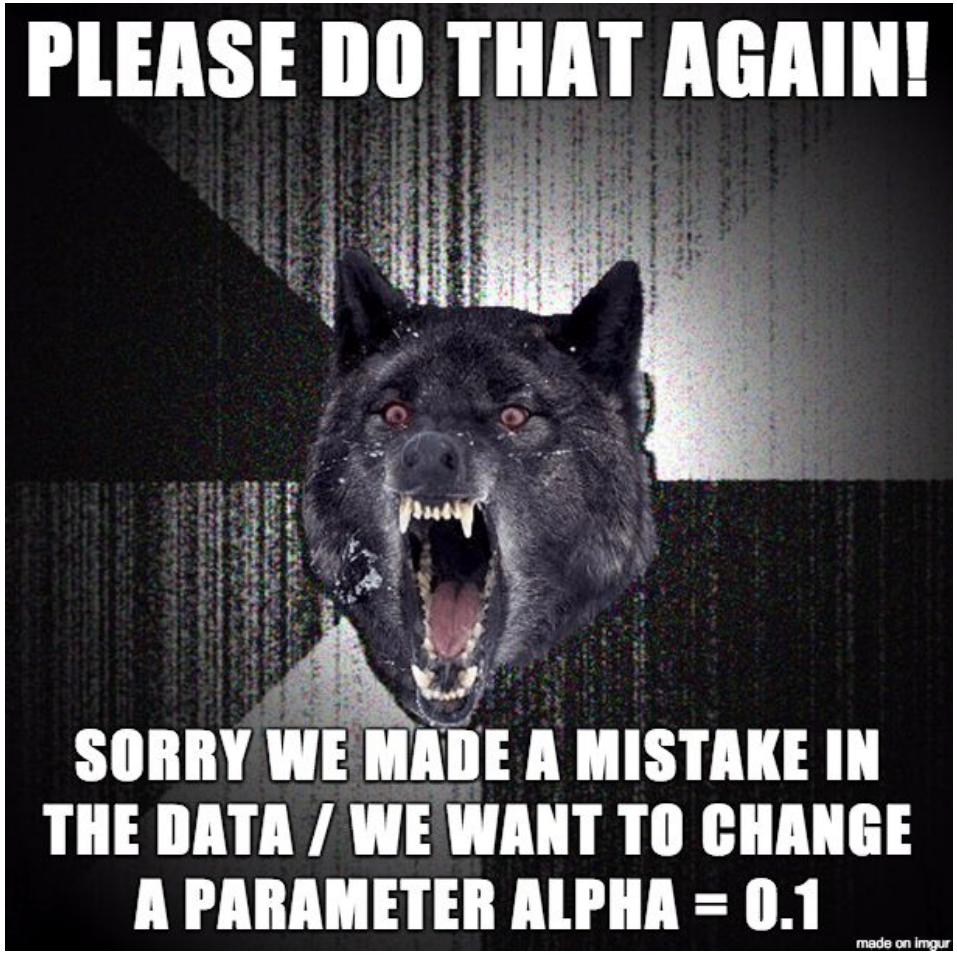
- Collect data
- Clean data
- Build / refine models / tune parameters
- Collect results from computing / graphics
- Write the report

Why escape the LaTeX / Word land

- Reproducible research
- R Markdown is much easier
- More fun/possibilities in the HTML/JavaScript world

I know you click, click, copy and paste

But imagine you hear these words after you finished a project



No cut-and-paste

- Dynamic Documents
- code + narratives = report
- i.e. computing languages + authoring languages

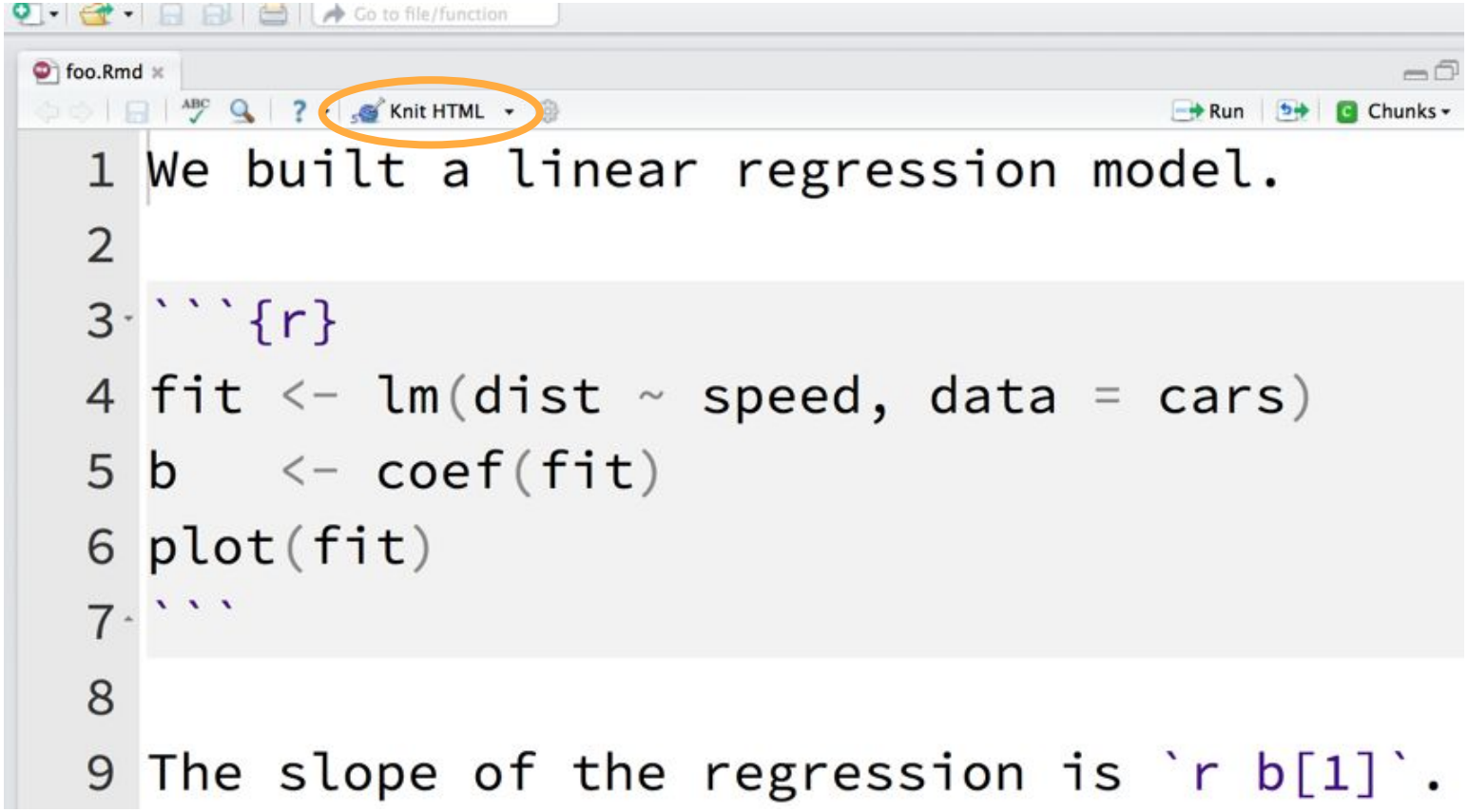
````
We built a linear regression model.

```{r}
fit <- lm(dist ~ speed, data = cars)
b   <- coef(fit)
plot(fit)
```

The slope of the regression is `r b[1]`.
````

RStudio (http://www.rstudio.com)

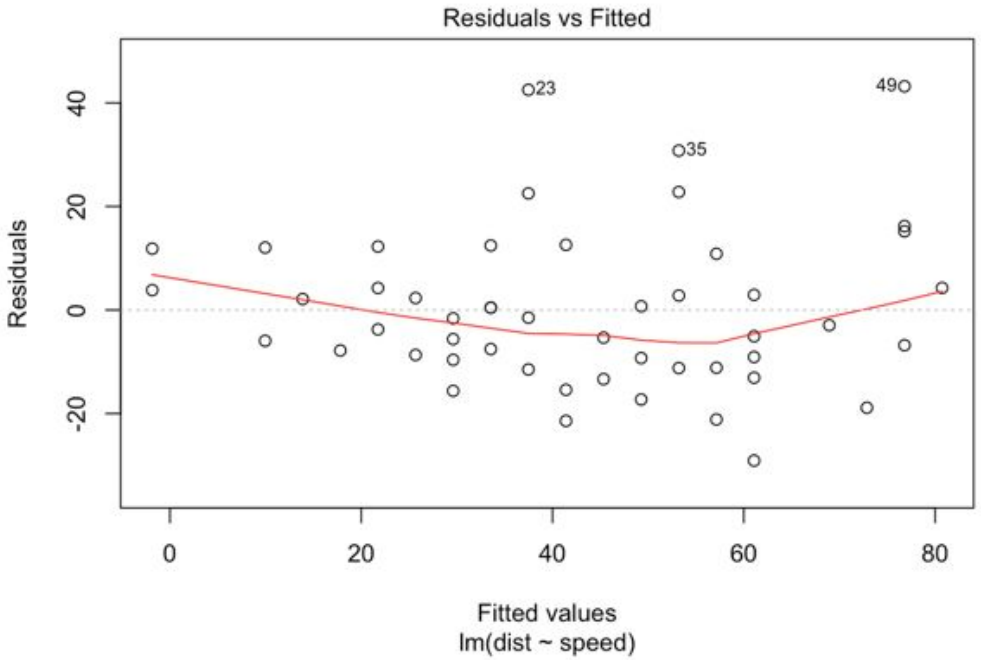
File ➔ New File ➔ R Markdown



Output

We built a linear regression model.

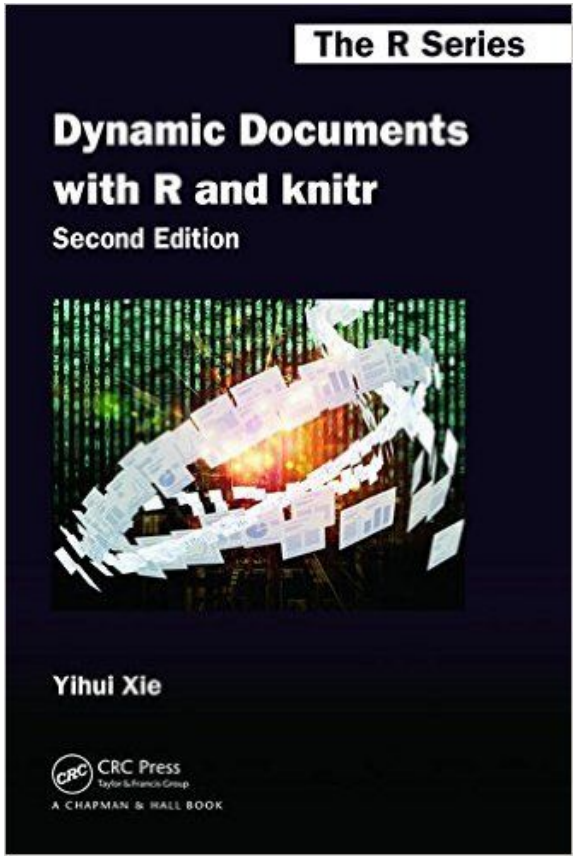
```
fit <- lm(dist ~ speed, data = cars)
b <- coef(fit)
plot(fit)
```



The slope of the regression is -17.5790949.

knitr

- an R package (`install.packages('knitr')`)
- document formats
  - .Rnw (R + LaTeX)
  - .Rmd (R + Markdown)
  - any computing language + any authoring language (in theory)
- editors
  - RStudio
  - LyX
  - Emacs/ESS
  - ...



What is possible with knitr

- text output

  ```{r, echo=FALSE}
  ```

- graphics

  ```{r, fig.width=5, fig.height=4}
  ```

- cache
- cross references (reuse code chunks)
- output hooks and chunk hooks (tweak output)
- language engines (C, C++, shell scripts, and many more)
- ...

Two extremes of documentation languages

LaTeX: precise control, full complexity, horrible readability

```
\section{Introduction}

We did a \emph{cool} study, and our main findings:

\begin{enumerate}
    \item You can never remember how to escape backslashes.
    \item A dollar sign is \$, an ampersand \&, and a
    \textbackslash{}.
    \item How about ~? Use $\sim$.
\end{enumerate}
```

Do you call that human-readable?

Two extremes of documentation languages

Markdown: simple, simple, simple

```
# Introduction

We did a _cool_ study, and our main findings:

1. You do not need to remember a lot of rules.
2. A dollar sign is $, an ampersand is &, and a backslash \.
3. A tilde is ~.

Write content instead of markup languages.
```

My personal award that nobody has ever been able to claim

If you are unable to learn pretty much *everything* about Markdown in 10 minutes, I will give you 10 dollars.

Use the quick reference in RStudio, or go to http://rmarkdown.rstudio.com for the comprehensive documentation.

If you are comfortable with LaTeX anyway

Use Rnw documents instead of R Markdown, e.g. in RStudio, File → New File → R Sweave

Markdown cannot do everything LaTeX does. If you write an extremely complicated document (e.g. lots of backslashes and custom commands), stay with LaTeX (i.e. use .Rnw documents instead of .Rmd).

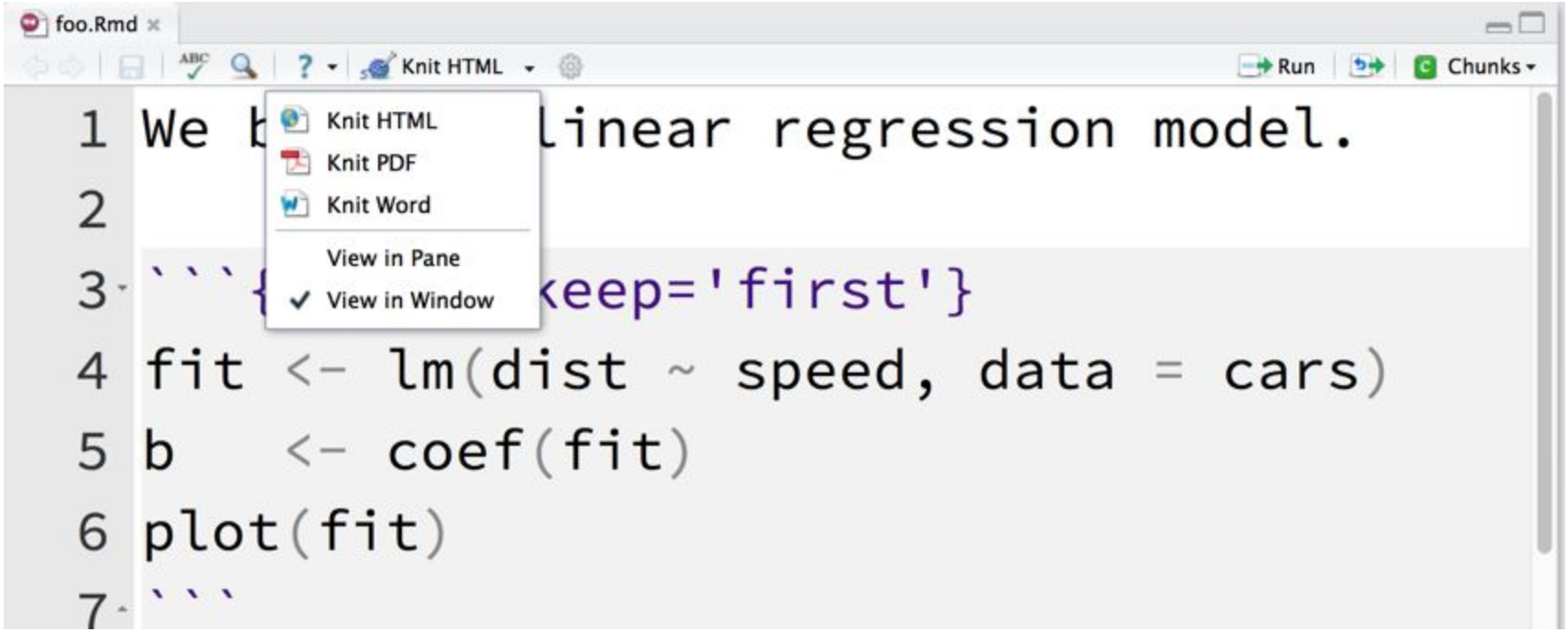
Word? http://noooooooooooooooo.com

Output formats from R Markdown

Thanks to Pandoc, you can convert Markdown to

- Documents
  - HTML
  - LaTeX / PDF (requires LaTeX, e.g. MikTeX on Windows, MacTeX on OS X, TeXLive on Linux)
  - MS Word (yes, Word)
- Presentations
  - LaTeX Beamer slides
  - HTML5 slides
    - Slidy presentation
    - ioslides
- Many, many other formats (e.g. E-books)

One click to rule them all



One syntax to rule them all

- `**text**` (think \textbf{text} in LaTeX, or text in HTML)
- `_text_` (\emph{text}, text)
- `# text` (\section{text}, <h1>text</h1>)
- `- item`
- `[text](url)`
- `![](image)`
- `$math$`
- `| table | column 1 | column 2 | ... |`
- `^[footnote]`
- `[@citation]`

Other applications of R Markdown

- Shiny
  - http://shiny.rstudio.com
  - Interactive computing with R on an HTML page
- HTML Widgets
  - http://www.htmlwidgets.org
  - Using JavaScript libraries in R without having to know JavaScript
  - Watch one of my previous talks here if you are interested: http://datascience.la/yihui-xie-presents-html-widgets/

Shiny + R Markdown ⇒ Interactive documents

File ➜ New File ➜ R Markdown ➜ Shiny

Open in Browser | Publish

A Shiny Document

Yihui Xie

September 2, 2015

You can embed Shiny inputs and outputs in your document. Outputs are automatically updated whenever inputs change. This demonstrates how a standard R plot can be made interactive by wrapping it in the Shiny `renderPlot` function. The `selectInput` and `sliderInput` functions create the input widgets used to drive the plot.

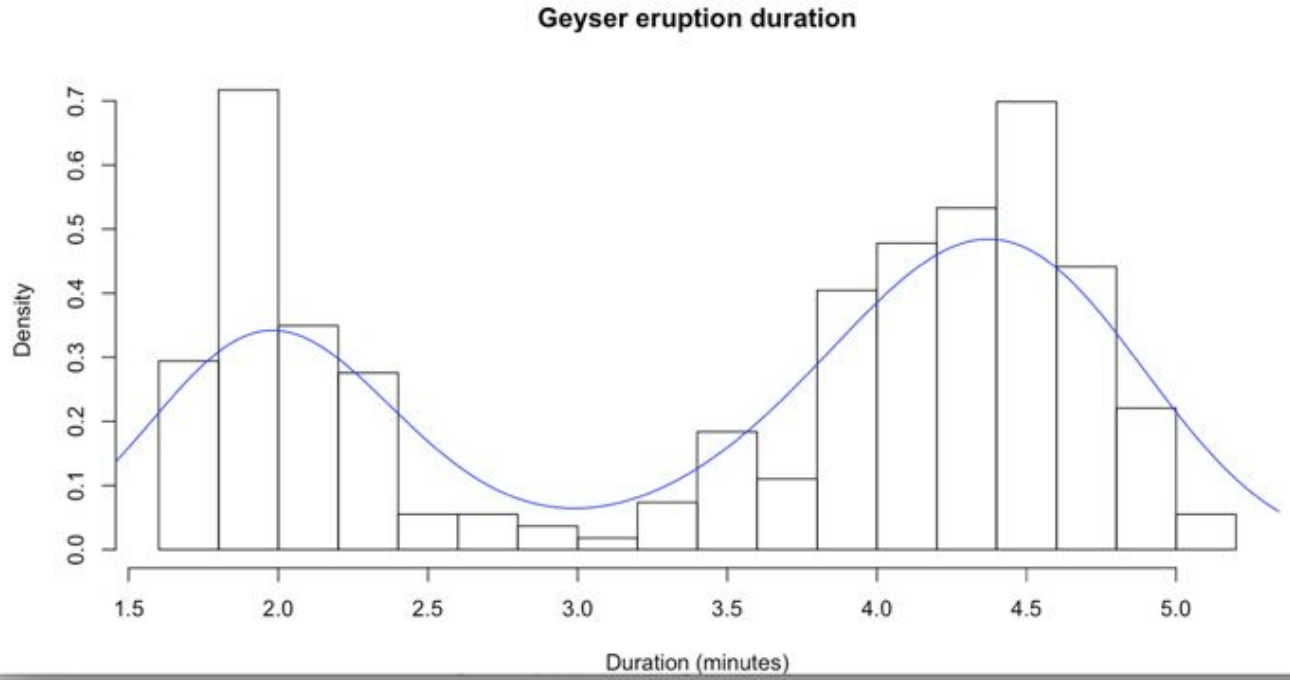
Number of bins:

20

Bandwidth adjustment:

0.2 1 2

0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2



HTML Widgets

- The R package htmlwidgets is the infrastructure package
  - you don't use it directly unless you are a widget developer
  - normally you use specific widget packages instead
- Widget packages
  - DT: http://rstudio.github.io/DT (tabular data display)
  - leaflet: http://rstudio.github.io/leaflet (geo-spatial mapping)
  - dygraphs: http://rstudio.github.io/dygraphs (time series charting)
  - many more...
- Widgets can be rendered in the R console, RStudio viewer, or R Markdown documents / Shiny apps

HTML Widgets + R Markdown

Just call the widget function(s) in a widget package in an R code chunk, e.g.

````
```{r}
if (!require('DT')) install.packages('DT')
library(DT)
datatable(iris)
```
````

Show 10 entries

Search:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries

Previous 1 2 3 4 5 … 15 Next

RPubs

Looking for examples? Go to http://rpubs.com

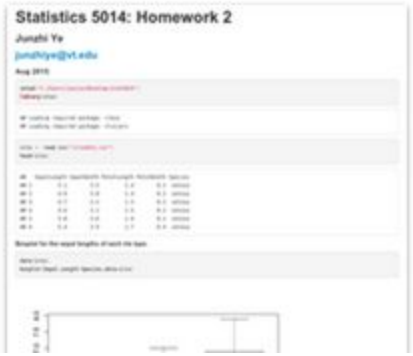
Easy web publishing from R

Write R Markdown documents in RStudio.

Share them here on RPubs. (It's free, and couldn't be simpler!)

Get Started

Recently Published



ShinyApps.io

To publish interactive Shiny documents, you may consider http://www.shinyapps.io unless you want to set up your own Shiny Server http://www.rstudio.com/products/shiny-server-pro

Since it is R, it is programmable

- To generate 1000 similar reports (e.g. same analysis/format, different data), you can just write a loop to do everything
  - click-cut-paste will be hopeless
- You can use program code to dynamically control the content of your report
  - e.g. show this part is P-value is < 0.05, otherwise show that part

PRINT IS DEAD

Ghostbusters movie, 1984 https://youtu.be/D3v_ogRaTf4

(image credit: Michael Carniello)

Thanks!

- Contact me
  - http://yihui.name
  - Twitter @xieyihui
  - Github @yihui
- RStudio
  - http://www.rstudio.com
  - Twitter @rstudio
  - Github @rstudio