

Работа с R и воспроизводимые исследования

Что обсуждаем

- Как поставить весь необходимый софт
- Как делать отчеты
- Как делать воспроизводимые отчеты
(работающие не только на устройствах автора отчета)

Установка R

(«стандартный» подход)

- Windows: скачать exe с <https://www.r-project.org>
- Linux: поставить из менеджера пакетов r-base (Пошаговая инструкция для ubuntu 14.04 <https://www.digitalocean.com/community/tutorials/how-to-set-up-r-on-ubuntu-14-04>)
- MacOS: Homebrew.

Пакеты в R

- Общий репозиторий CRAN
- Установка: `install.packages("packageName")`
- Устанавливается локально — доступно только пользователю

Пакеты с Github

- Ставим devtools `install.packages('devtools')`
- `devtools::install_github('IRkernel/IRkernel')`

Про пакеты на Windows

- Ставит бинарные пакеты. Все с CRAN должно быть доступно
- Если требуется компиляция, то «все плохо». Какой-то tutorial есть здесь: <https://www.biostat.wisc.edu/~kbroman/Rintro/Rwinpack.html>
- Есть шансы, что пакет под windows не соберется. Ставьте ubuntu в виртуалку.

Сборка пакетов на Mac

- Компилятор по-умолчанию — clang
- Не умеет openmp — модный xgboost будет работать в один поток
- brew install gcc (с фортраном) и прочие зависимости...
- Некоторые пакеты clang'м просто не собираются

Code editor, visualisation, etc

- RStudio: <https://www.rstudio.com>
- Jupyter notebooks: <http://jupyter.org>

RStudio и выбор интерпретатора R

- Windows: В настройках RStudio указать путь к нужному R
- Добавить в ~/.bash_profile: `launchctl setenv RSTUDIO_WHICH_R /anaconda/bin/R`
- Linux: `export RSTUDIO_WHICH_R=...`

Не воспроизводимые исследования

- Большая часть старых исследований, особенно в далеких от «математики и программирования» областях
- Невозможно проверить
- Невозможно развивать
- Нельзя передать другому человеку

Про психологов

- 2011г, 100 статей:
 - 61 — результаты не совпадают, в 15 случаях — совсем не совпадают.
 - 36 — повторяются, но уровень значимости завышен в 2 раза

<https://nplus1.ru/news/2015/08/28/psychology-studies-fail-reproducibility>

Частичный workflow исследователя

- Собираем «сырые» данные
- Код для обработки данных
- Код для анализа данных (графики, стат. тесты, etc)
- Отчет по анализу, выводы, презентации

R Markdown

- Cheatsheet: <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>
- Reference: <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>
- More: Google and <https://www.rstudio.com>

Пример RMarkdown- файла в Studio

R Markdown Cheat Sheet

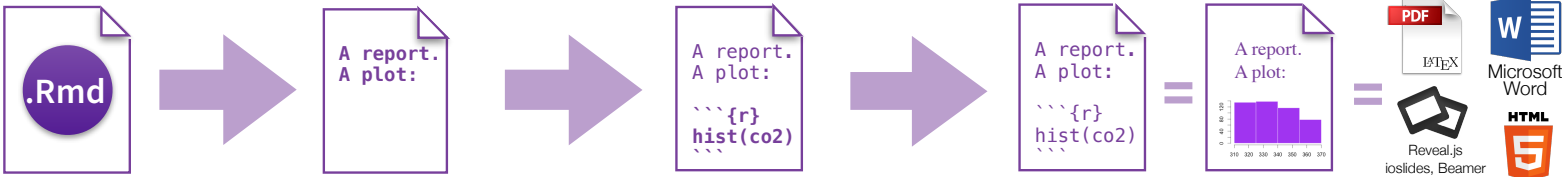
learn more at rmarkdown.rstudio.com

rmarkdown 0.2.50 Updated: 8/14



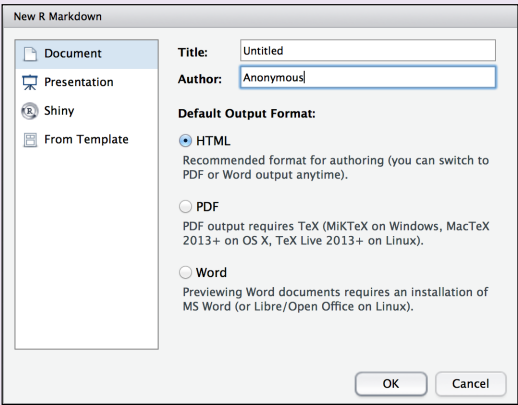
1. Workflow R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

- i. Open** - Open a file that uses the .Rmd extension.
- ii. Write** - Write content with the easy to use R Markdown syntax
- iii. Embed** - Embed R code that creates output to include in the report
- iv. Render** - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.



2. Open File Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

- In the menu bar, click **File ► New File ► R Markdown...**
- A window will open. Select the class of output you would like to make with your .Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK



3. Markdown Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

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: 

horizontal rule (or slide break):

\*\*\*

> block quote

\* unordered list  
\* item 2  
+ sub-item 1  
+ sub-item 2

1. ordered list  
2. item 2  
+ sub-item 1  
+ sub-item 2

| Table Header | Second Header |
|--------------|---------------|
| Table Cell   | Cell 2        |
| Cell 3       | Cell 4        |

### becomes

Plain text  
End a line with two spaces to start a new paragraph.  
*italics* and *italics*  
**bold** and **bold**  
<sup>superscript<sup>2</sup></sup>  
~~strikethrough~~  
[link](#)

Header 1  
Header 2  
Header 3  
Header 4  
Header 5  
Header 6

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



horizontal rule (or slide break):

block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

- ordered list
- item 2
  - sub-item 1
  - sub-item 2

| Table Header | Second Header |
|--------------|---------------|
| Table Cell   | Cell 2        |
| Cell 3       | Cell 4        |

**4. Choose Output** Write a YAML header that explains what type of document to build from your R Markdown file.

### YAML

A YAML header is a set of key: value pairs at the start of your file. Begin and end the header with a line of three dashes (---)

```
---  
title: "Untitled"  
author: "Anonymous"  
output: html_document  
---
```

This is the start of my report. The above is metadata saved in a YAML header.

The RStudio template writes the YAML header for you

The output value determines which type of file R will build from your .Rmd file (in Step 6)

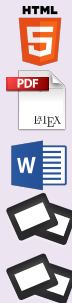
output: html\_document ..... html file (web page)

output: pdf\_document ..... pdf document

output: word\_document ..... Microsoft Word .docx

output: beamer\_presentation ..... beamer slideshow (pdf)

output: ioslides\_presentation ..... ioslides slideshow (html)



**5. Embed Code** Use knitr syntax to embed R code into your report. R will run the code and include the results when you render your report.

### inline code

Surround code with back ticks and `r`. R replaces inline code with its results.

Two plus two equals ``r 2 + 2``.  
Two plus two equals 4.

### code chunks

Start a chunk with ``{r}``.  
End a chunk with ``}``.

Here's some code  
``{r}``  
`dim(iris)`  
``}``  
Here's some code  
`dim(iris)`  
## [1] 150 5

### display options

Use knitr options to style the output of a chunk.  
Place options in brackets above the chunk.

Here's some code  
``{r eval=FALSE}``  
`dim(iris)`  
Here's some code  
`dim(iris)`  
Here's some code  
``{r echo=FALSE}``  
`dim(iris)`  
Here's some code  
`dim(iris)`  
## [1] 150 5

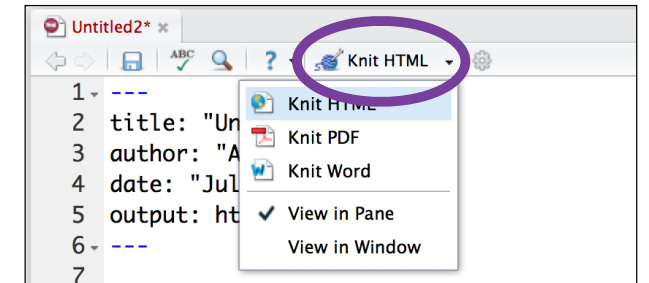
| option     | default  | effect                                                    |
|------------|----------|-----------------------------------------------------------|
| eval       | TRUE     | Whether to evaluate the code and include its results      |
| echo       | TRUE     | Whether to display code along with its results            |
| warning    | TRUE     | Whether to display warnings                               |
| error      | FALSE    | Whether to display errors                                 |
| message    | TRUE     | Whether to display messages                               |
| tidy       | FALSE    | Whether to reformat code in a tidy way when displaying it |
| results    | "markup" | "markup", "asis", "hold", or "hide"                       |
| cache      | FALSE    | Whether to cache results for future renders               |
| comment    | "##"     | Comment character to preface results with                 |
| fig.width  | 7        | Width in inches for plots created in chunk                |
| fig.height | 7        | Height in inches for plots created in chunk               |

For more details visit [yihui.name/knitr/](http://yihui.name/knitr/)

**6. Render** Use your .Rmd file as a blueprint to build a finished report.

Render your report in one of two ways

1. Run `rmarkdown::render("<file path>")`
2. Click the **knit HTML** button at the top of the RStudio scripts pane



When you render, R will

- execute each embedded code chunk and insert the results into your report
- build a new version of your report in the output file type
- open a preview of the output file in the viewer pane
- save the output file in your working directory

**7. Interactive Docs** Turn your report into an interactive Shiny document in 3 steps

1 Add **runtime: shiny** to the YAML header

```
---
title: "Line graph"
output: html_document
runtime: shiny
---
```

2 In the code chunks, add Shiny **input** functions to embed widgets. Add Shiny **render** functions to embed reactive output

```
---
title: "Line graph"
output: html_document
runtime: shiny
---
```

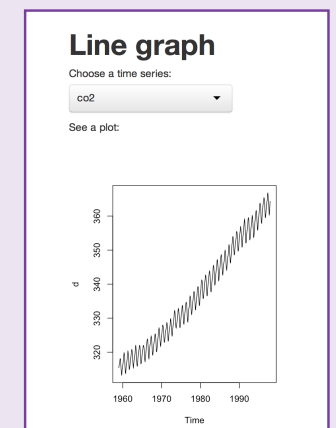
Choose a time series:

```
`{r echo = FALSE}`
selectInput("data", "",
  c("co2", "lh"))
```

See a plot:

```
`{r echo = FALSE}`
renderPlot({
  d <- get(input$data)
  plot(d)
})
```

3 Render with **rmarkdown::run** or click **Run Document** in RStudio



\* Note: your report will be a Shiny app, which means you must choose an html output format, like **html\_document** (for an interactive report) or **ioslides\_presentation** (for an interactive slideshow).

**8. Publish** Share your report where users can visit it online

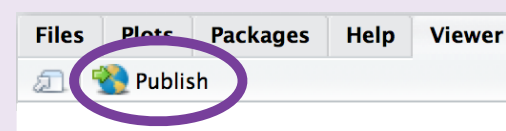
### Rpubs.com

Share non-interactive documents on RStudio's free R Markdown publishing site  
[www.rpubs.com](http://www.rpubs.com)

### ShinyApps.io

Host an interactive document on RStudio's server. Free and paid options  
[www.shinyapps.io](http://www.shinyapps.io)

Click the "Publish" button in the RStudio preview window to publish to [rpubs.com](http://rpubs.com) with one click.



**9. Learn More**

Documentation and examples - [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)

Further Articles - [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)

🌐 - [blog.rstudio.com](http://blog.rstudio.com)

🐦 - [@rstudio](https://twitter.com/rstudio)



RStudio® and Shiny™ are trademarks of RStudio, Inc.  
CC BY RStudio [info@rstudio.com](mailto:info@rstudio.com)  
844-448-1212 [rstudio.com](http://rstudio.com)

# Форматы отчетов

- html
- docx
- pdf — с русским «из коробки» скорее всего не заработает:

```
1 ---
2 title: "Examples"
3 mainfont: "Helvetica"
4 output:
5   pdf_document:
6     latex_engine: xelatex
```

# RStudio preview

- Notebooks (аналог jupyter, из плюсов — version control friendly)
- Поддержка не только R (Python, Bash, SQL, Rcpp, Stan)

# Anaconda

- Набор пактов для data science
- Изначально ориентирована на python
- Включает менеджер пакетов conda
- Для многих задач достаточно поставить anaconda и забыть про установку каких-либо пакетов

# conda

- Менеджер пакетов
- Воспроизводимость — позволяет создавать среду с наборами необходимых пакетов (и фиксированными версиями) и «поделиться» ей

# conda environment

- `conda create -n env_name packages`
- `conda remove -n env_name --all`
- `source activate env_name` (или `activate env_name` на Windows)
- `conda info --envs` — все envs
- `conda env export > filename.yml`
- `conda env create -f filename.yml`

# Conda and R

- Живет в channel r
- Метапакет r-essentials — набор популярных пакетов для r
- `conda install -c r r-essentials`

# conda build

- В r-channel anaconda только популярные пакеты
- Rssa там нету (а вам когда-нибудь понадобится)
- Можно устанавливать через `install.packages`, но не «воспроизводимо»

# conda build from cran

- conda позволяет легко создавать свои пакеты, учитывающие зависимости
- `conda install conda-build`
- `conda skeleton cran package_name`
- Создастся папка с мета-информацией про пакет
- Далее может хватить `cd ... && conda build .`
- Квест: собрать Rssa под вашу систему

# Собственные метапакеты

- `conda metapackage bundle_name version — dependencies ... —summary “some text”`

# Jupyter

- С `anaconda` доступен «из коробки».  
Рекомендуемый и самый быстрый способ установки
- На `unix-like` системах можно установить с использованием системного `python`.
- Запуск: `jupyter notebook`

# ВОЗМОЖНОСТИ

- Аналогичный RStudio markdown
- Вся работа через web-browser
- Интерактивный
- Поддержка большого кол-ва языков (python, r, julia, scala, ...)
- Генерация слайдов

# IRKernel

- С anaconda и r-essentials доступен «из коробки»
- Без anaconda: устанавливаем отсюда <https://github.com/IRkernel/IRkernel>

# Экспорт

- «Тетрадку» можно сохранить в множестве разных форматов (чистый код, markdown, html, pdf)
- Часть форматов — доступно File->Download As
- Остальное nbconvert
- Всегда можно сохранить markdown и вызвать pandoc

# Пример jupyter, обзор «КНОПОК»

# Microsoft R

- Бывший Revolution R
- «Оптимизированная» версия R под многопоточные процессоры
- Для анаконды живут в channel MRO
- За счет env — легко «попробовать» и откатить, когда что-нибудь сломается

# Еще немного полезной информации

- Интерактивные js-графики из R: <https://plot.ly>
- Пакет xtable — создание tex-таблиц из data.frame
- И RStudio, и jupyter могут работать на удаленном сервере