

Generating dynamic reports using R Markdown in RStudio

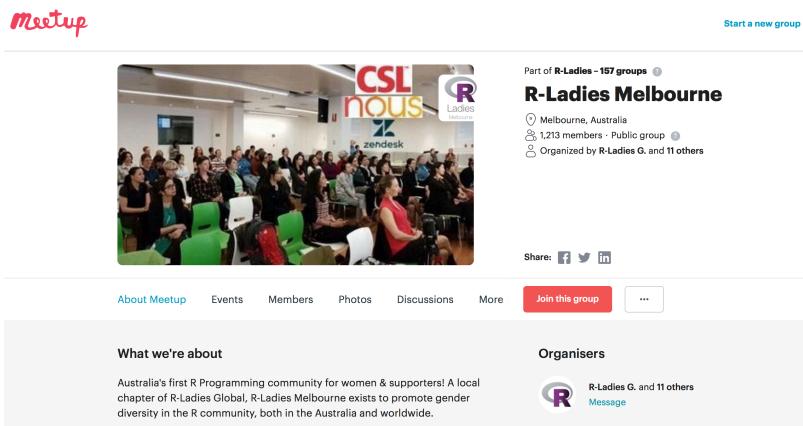
Momeneh (Sepideh) Foroutan

@S_Foroutan

June 2019 (updated: 01 July, 2019)

About me and R

- Background in Molecular Genetics (BSc and MSc)
- PhD in Computational Cancer Biology
- Self teaching R through on-line courses and forums
- Co-founder of **R-Ladies Melbourne** in Sep 2016
- President of R-Ladies Melbourne Inc.



The image shows a screenshot of the R-Ladies Melbourne group page on Meetup.com. At the top, there's a red 'meetup' logo. To the right, a blue button says 'Start a new group'. Below the logo is a large photo of a group of people in a meeting room. To the right of the photo, it says 'Part of R-Ladies - 157 groups'. The group name 'R-Ladies Melbourne' is in bold black text. Below that, it says 'Melbourne, Australia', '1,213 members - Public group', and 'Organized by R-Ladies G. and 11 others'. There are social sharing icons for Facebook, Twitter, and LinkedIn. Below the photo, there are navigation links: 'About Meetup', 'Events', 'Members', 'Photos', 'Discussions', 'More', 'Join this group', and a 'More' button. A grey sidebar on the left contains 'What we're about' (describing it as Australia's first R Programming community for women & supporters) and 'Organisers' (listing R-Ladies G. and 11 others). A 'Message' button is also present in the sidebar.

<https://www.meetup.com/en-AU/r-ladies-melbourne/>

R-Ladies



<https://blog.revolutionanalytics.com/2018/12/women-and-r.html>

Motivation

Imagine a project ...

You are given:

- Gene expression data for some cancer samples
- A gene expression signature

You are asked to:

- Find out which samples are more concordant with that signature?
- Communicate the analyses with your colleagues

How I used to do these without R Markdown

I would have:

- a folder with several analysis **scripts**
- a folder with several **figures**
- a folder with several **tables** (e.g. .csv, txt, tsv, etc)
- a **notebook** storing all the notes (rationale of the analyses, methodology, interpretation and descriptions)
- a folder of **papers** that have relevant figures and information.
- **It was always pretty hectic to *communicate* all these information with colleagues, *reproduce* all the results, and *share* my analyses.**

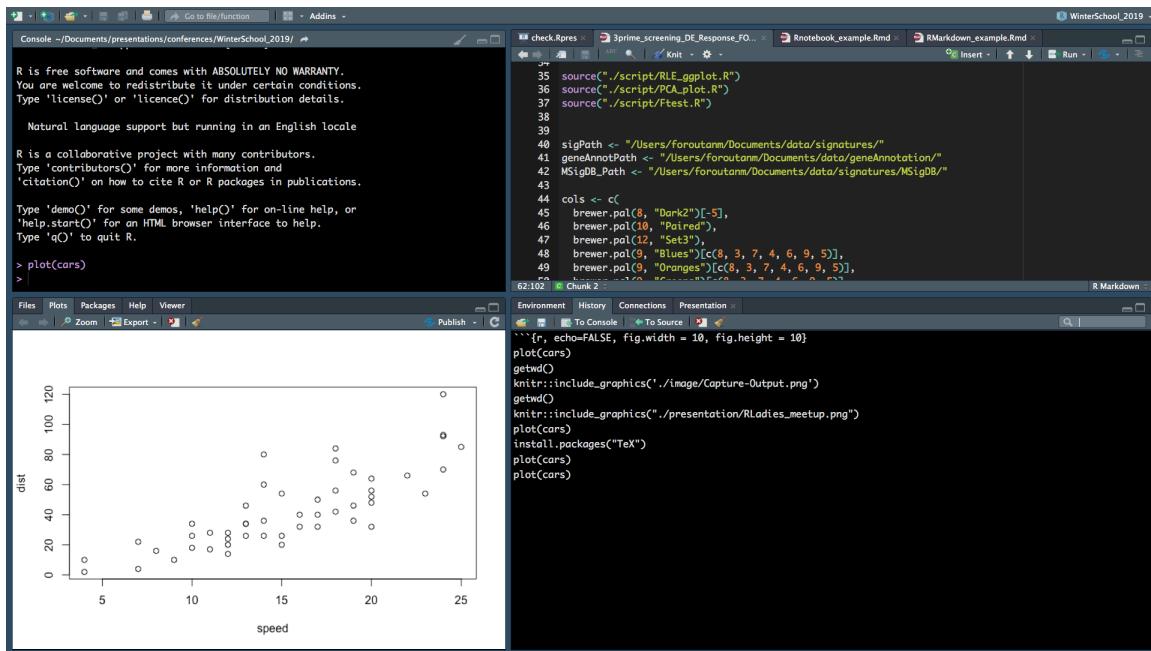
R Markdown was a game changer!

R Markdown is an authoring framework provided by RStudio, which can keep all steps of the analyses together:

- Codes (save and execute)
- Figures and tables
- Methodology, interpretations, and descriptions of the analyses
- Link to papers, and images from papers
- **It is now much easier to *communicate* all these information with colleagues, *reproduce* all the results, and *share* my analyses.**
- **Several output formats, and possibilities for static and dynamic (interactive) reports.**

RStudio

RStudio is an integrated development environment (IDE) for R which includes a console, syntax-highlighting editor, and tools for plotting, history, debugging, etc. Rstudio help you to interact with R more readily.



Things you can do using RStudio

- Write, save and run codes
- Generate interactive web application
- Generate high quality reports and documents
- Making presentation slides
- Version control (Git/Github)
- Many more
- [RStudio website](#)
- [Introduction to R and RStudio](#)
- [RStudio cheatsheet](#)

Online courses

Data Science Specialisation by JHU on Coursera

R Markdown and knitr

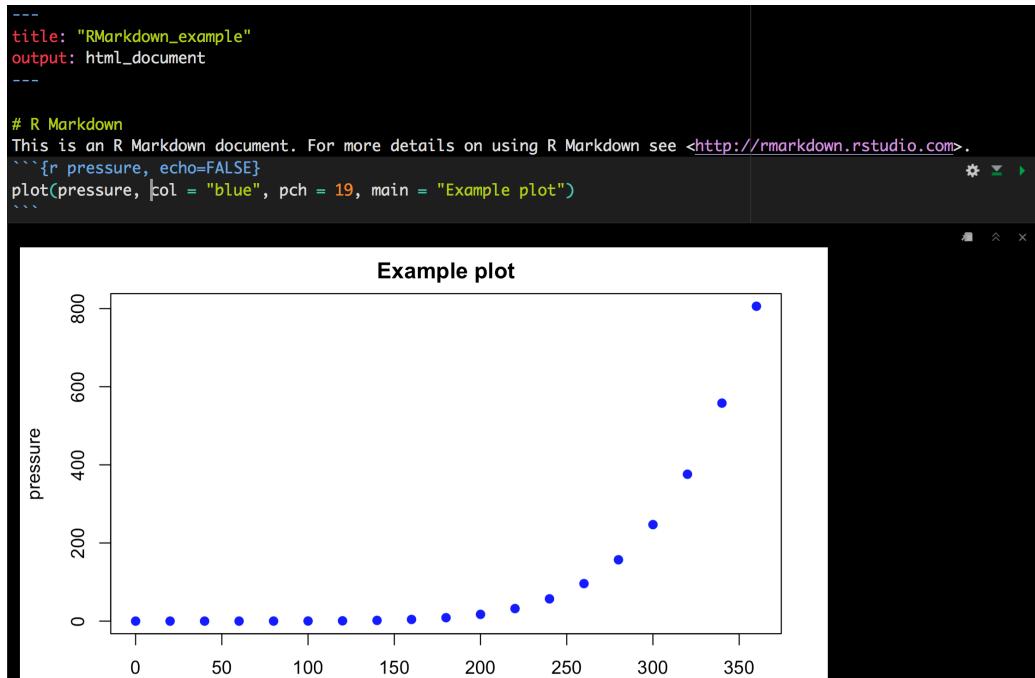
R Markdown (.Rmd) combines **R codes** (.R) and **documentatiuon language** (.md) using **knitr**.

knitr ...

- inspired by Sweave
- is an R package which works as an engine for dynamic report generation in R.
- helps us to integrate R codes into other documents (e.g. Markdown, LaTeX, HTML, etc).
- can generate HTML, PDF or Word documents.
- supports other languages, such as Python, Perl, C++, Shell scripts, etc.
- enables reproducible research

R Markdown main sections

YAML header, Markdown text, and code chunks (with outputs).



The screenshot shows the RStudio interface with an R Markdown document. The YAML header at the top defines the title and output type. The code chunk below it contains R code to generate a scatter plot titled "Example plot". The plot shows a positive linear relationship between two variables, with the x-axis labeled "X" and the y-axis labeled "pressure".

```
---
```

```
title: "RMarkdown_example"
```

```
output: html_document
```

```
---
```

```
# R Markdown
```

```
This is an R Markdown document. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
```

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

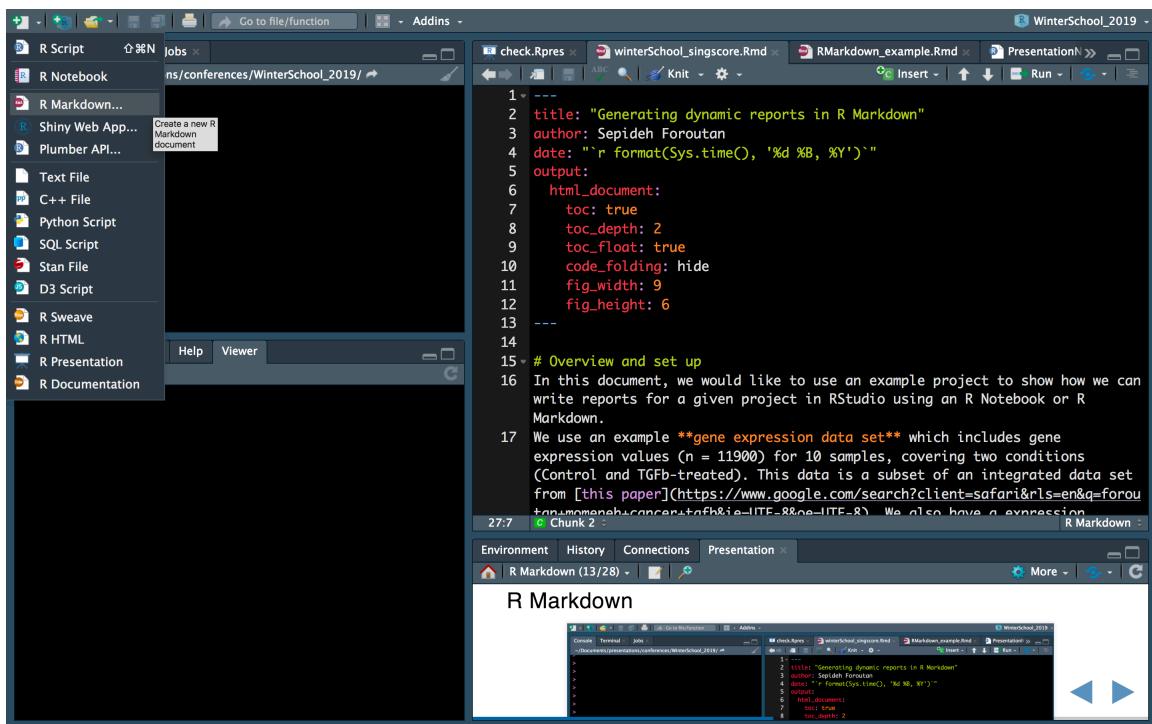
```
plot(pressure, col = "blue", pch = 19, main = "Example plot")
```

```
```
```

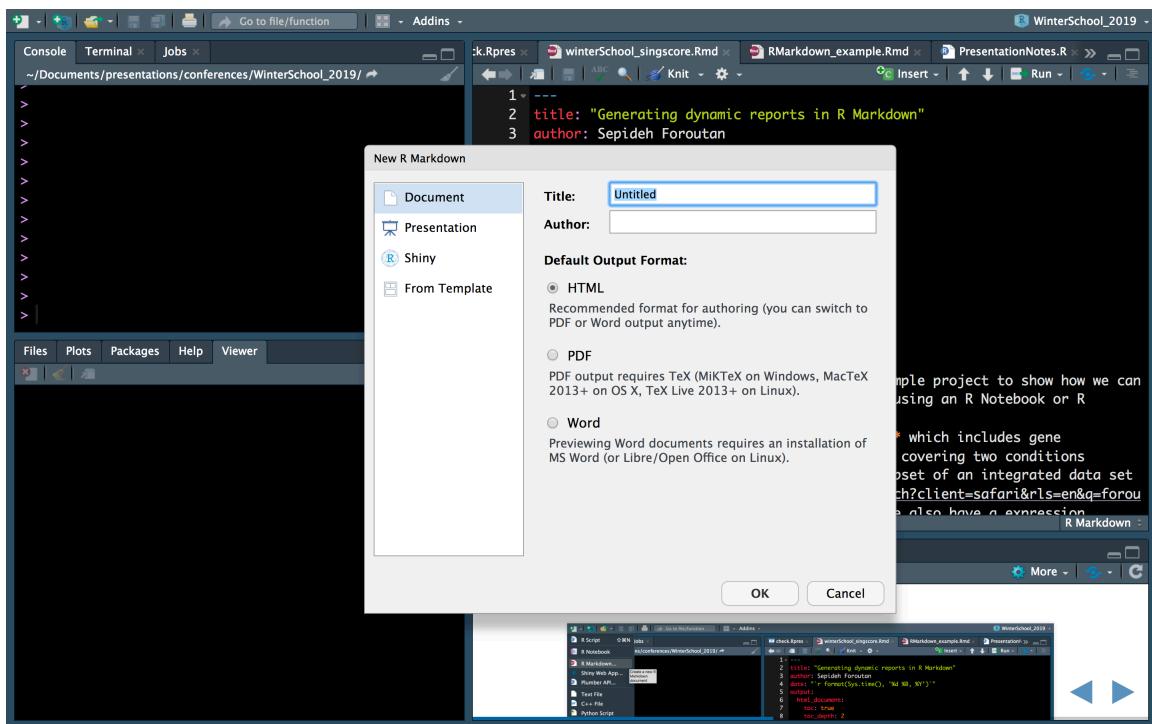
Example plot

| X | pressure |
|-----|----------|
| 0 | 0 |
| 50 | 0 |
| 100 | 0 |
| 150 | 0 |
| 200 | 0 |
| 225 | 50 |
| 250 | 100 |
| 275 | 150 |
| 300 | 250 |
| 325 | 400 |
| 350 | 600 |
| 350 | 800 |

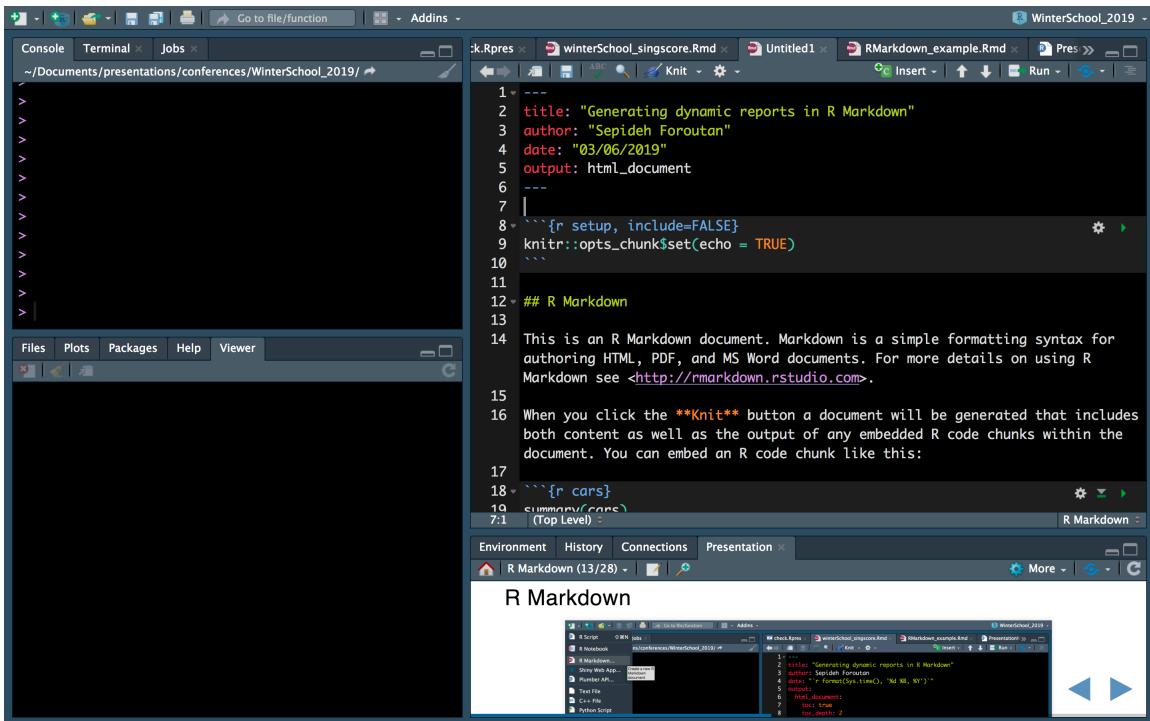
Open a new R Markdown



Open a new R Markdown



Open a new R Markdown

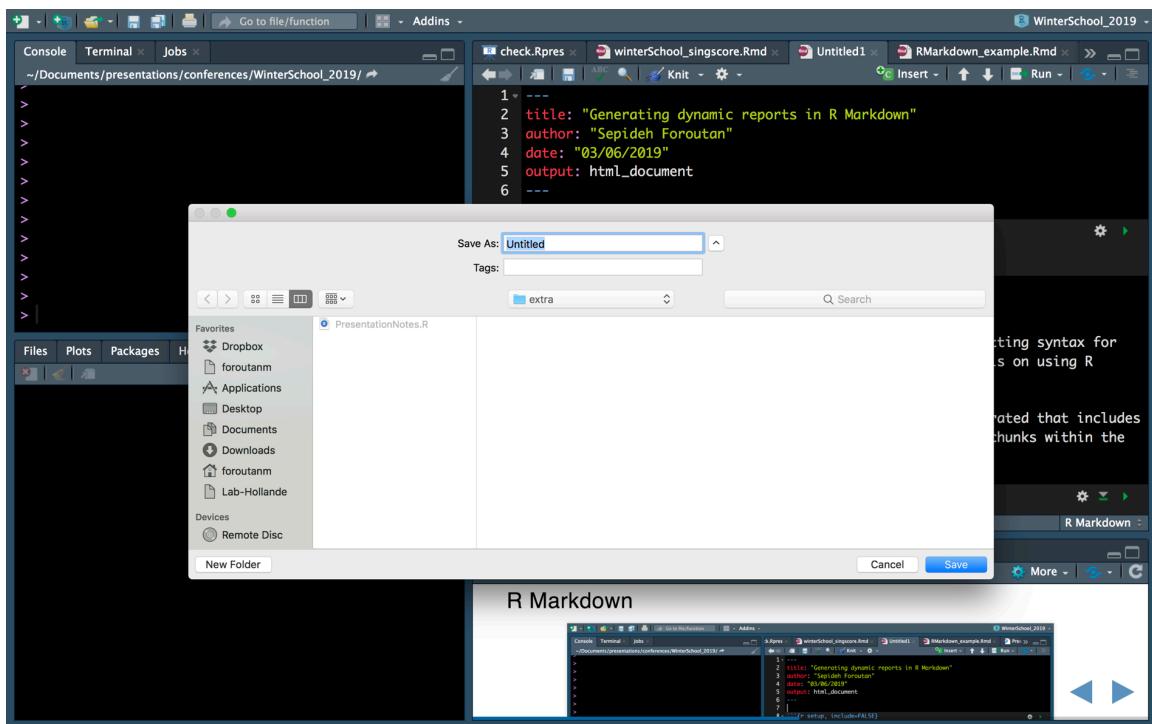


The screenshot shows the RStudio interface with a new R Markdown file open. The code editor displays the following content:

```
1 ---  
2 title: "Generating dynamic reports in R Markdown"  
3 author: "Sepideh Foroutan"  
4 date: "03/06/2019"  
5 output: html_document  
6 ---  
7 |  
8 ````{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ````  
11 ## R Markdown  
12  
13 This is an R Markdown document. Markdown is a simple formatting syntax for  
authoring HTML, PDF, and MS Word documents. For more details on using R  
Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that includes  
both content as well as the output of any embedded R code chunks within the  
document. You can embed an R code chunk like this:  
17  
18 ````{r cars}  
19 summary(cars)  
7:1 (Top Level) R Markdown
```

The R Markdown pane shows the rendered content of the document, which includes the R code and its output.

Open a new R Markdown



Open a new R Markdown

example.html | Open in Browser | Find | Publish |

Generating dynamic reports in R Markdown

Sepideh Foroutan

03/06/2019

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed         dist
##  Min.   :4.0   Min.   : 2.00
##  1st Qu.:12.0  1st Qu.:26.00
##  Median :15.0  Median :36.00
##  Mean   :15.4  Mean   :42.98
##  3rd Qu.:19.0  3rd Qu.:56.00
##  Max.   :25.0  Max.   :120.00
```

Including Plots

You can also embed plots, for example:



R Notebook

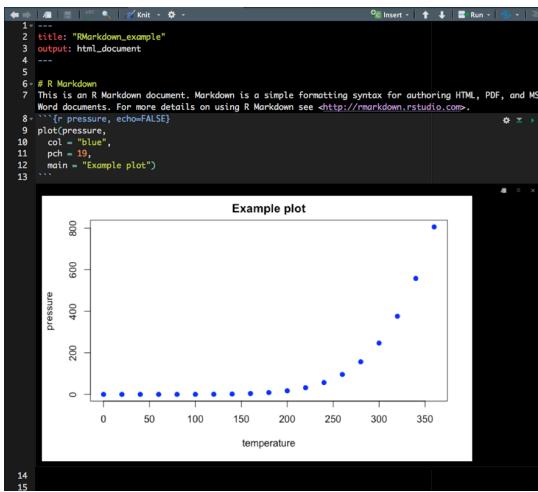
An **R Notebook** is an R Markdown document with chunks that can be executed independently and interactively, with output visible immediately beneath the input.

- **R Markdown** (knitr button): executes and evaluates all code in one go. It can be very time-consuming when we have heavy processing
- **R Notebook** (preview button): has caching behaviour; it evaluates a code chunk and save it.

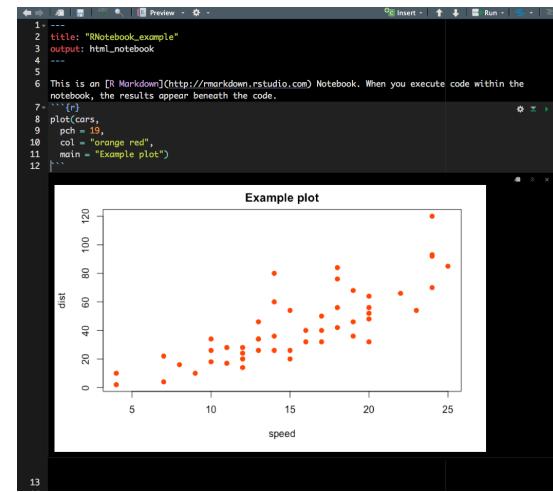
"The immediacy of notebook mode makes it a good choice while authoring the R Markdown document and iterating on code. When you are ready to publish the document, you can share the notebook directly, or render it to a publication format with the Knit button."

R Markdown and R Notebook

- R Markdown
 - **knit** button
 - output: **html_document**



- R Notebook
 - **preview** button
 - output: **html_notebook**



How does it look when we knit?

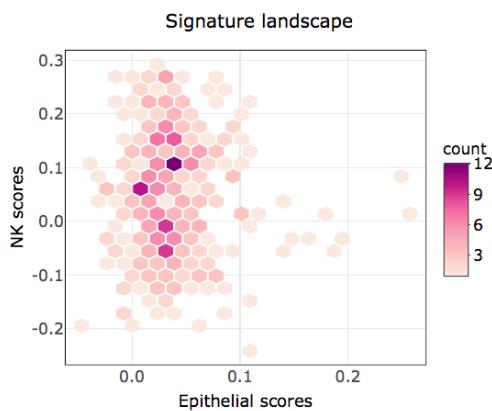
| |
|---------------------------------------|
| 1 Overview and set up |
| 2 Score samples |
| 3 Explore the scores |
| 3.1 Landscape plots |
| 3.2 Signature genes in single samples |
| 4 Survival analysis |
| 5 Session information |

3 Explore the scores

3.1 Landscape plots

We would like to plot landscape of NK scores versus epithelial scores; to do this, we use the `plotScoreLandscape` function from `singscore` package.

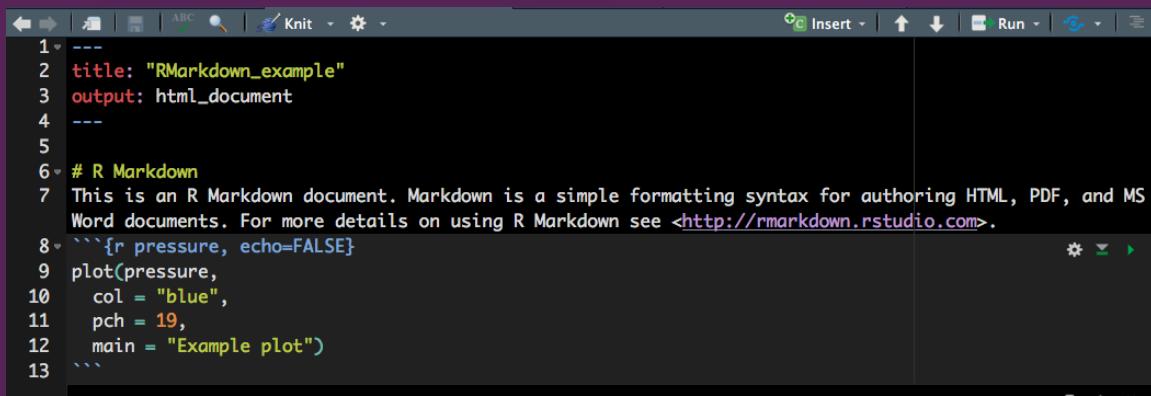
```
plotScoreLandscape(scoredf1 = epiScore_tcga,
                    scoredf2 = nkScore_tcga,
                    scorenames = c("Epithelial scores", "NK scores"),
                    textSize = 1,
                    isInteractive = T,
                    hexMin = 100)
```



https://github.com/DavisLaboratory/NK_scoring/

Creating the document

- Modify YAML header
- Structure and format the text
- Insert and modify code chunks
- Generate interactive tables
- Generate interactive figures



A screenshot of the RStudio interface showing an R Markdown file. The code editor contains the following content:

```
1 ---  
2 title: "RMarkdown_example"  
3 output: html_document  
4 ---  
5  
6 # R Markdown  
7 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
8 ```{r pressure, echo=FALSE}  
9 plot(pressure,  
10   col = "blue",  
11   pch = 19,  
12   main = "Example plot")  
13 ```
```

YAML header

"YAML (YAML Ain't Markup Language) is a human-friendly data serialization standard for all programming languages."

YAML header allows us to modify the output of the document; YAML section can evaluate R expressions. By modifying the YAML header, we can add/change:

- Output format
- Table of content (toc)
- Tabbed sections
- Global figure options (width and height)
- Custom CSS
- a lot more

YAML header

```
---
title: "Generating dynamic reports in R Markdown - an example using singscore"
author: Sepideh Foroutan
date: "`r format(Sys.time(), '%d %B, %Y')`"
output:
  BiocStyle::html_document:
    toc: true
    toc_depth: 2
    toc_float: true
    fig_caption: true
    number_sections: true
    code_folding: hide
    fig_width: 9
    fig_height: 6
params:
  output_dir: "./reports"
---
```

| | |
|----------|--|
| 1 | Overview and set up |
| 2 | Data and signature |
| 3 | Score samples |
| 4 | Insert EMT landscape from the literature |
| 5 | Interactive plot independent of singscore method |

1 Overview and set up

In this document, we would like to use an example project to show how we can write reports in RStudio using R Markdown. We use an example gene expression data set which includes gene expression values ($n = 11900$) for 10 samples, covering two conditions (Control and TGFb-treated). This data is a subset of an integrated data set from [this paper](#). We also have a expression signatures, called TGFb-EMT signature generated in the same paper. Both the data subset and the signature are available from the singscore R/Bioconductor package.

The purpose of this project is to find samples that are more concordant with the TGFb-EMT signature. To do this, we need to use a gene-set scoring method and samples' transcriptional profiles to score samples against this signature, and then compare their scores.

To score samples, we use the [singscore method](#), which is available as an R/Bioconductor package. If you are interested in the method, you can check the workflow paper by Bhuvu et al., [Using singscore to predict mutations in acute myeloid leukemia from transcriptomic signatures](#).

[Code](#)

2 Data and signature

YAML header

```
---
title: "Generating dynamic reports in R Markdown - an example using singscore"
author: Sepideh Foroutan
date: "`r format(Sys.time(), '%d %B, %Y')`"
output:
  BiocStyle::html_document:
    toc: true
    toc_depth: 2
    toc_float: true
    fig_caption: true
    number_sections: true
    code_folding: hide
    fig_width: 9
    fig_height: 6
params:
  output_dir: "./reports"
---
```

| | |
|----------|--|
| 1 | Overview and set up |
| 2 | Data and signature |
| 3 | Score samples |
| 4 | Insert EMT landscape from the literature |
| 5 | Interactive plot independent of singscore method |

1 Overview and set up

In this document, we would like to use an example project to show how we can write reports in RStudio using R Markdown. We use an example gene expression data set which includes gene expression values ($n = 11900$) for 10 samples, covering two conditions (Control and TGFb-treated). This data is a subset of an integrated data set from [this paper](#). We also have a expression signatures, called TGFb-EMT signature generated in the same paper. Both the data subset and the signature are available from the [singscore](#) R/Biocductor package.

The purpose of this project is to find samples that are more concordant with the TGFb-EMT signature. To do this, we need to use a gene-scoring method and samples' transcriptional profiles to score samples against this signature, and then compare their scores.

To score samples, we use the [singscore](#) method, which is available as an R/Biocductor package. If you are interested in the method, you can check the workflow paper by Bhuvu et al. [Using singscore to predict mutations in acute myeloid leukemia from transcriptomic signatures](#).

```
library(knitr)
library(DT)
library(gplots)
library(singscore)
library(SummarizedExperiment)
library(GSEABase)
```

[Hide](#)

2 Data and signature

Structure and modify the text

- Headings and sub-headings are generated using #:

```
# Heading 1
```

Heading 1

```
## Heading 2
```

Heading 2

- Italic texts are generated using :

This is italic!

This is italic!

- Bold texts are generated using :

This is bold!

This is bold!

Structure and modify the text

- Add hyperlinks using [your_text](your_url); for example:

[R Markdown documentation]

(<https://rmarkdown.rstudio.com>) will make R Markdown documentation clickable, which opens up the corresponding webpage for the documentation.

- Use **single backticks** as wrappers to change the font to make the code, package names, etc different from other plain texts.
- Add bullet points using **minus** and **plus** signs as well as tab.

For example:

- First point

Tab + class A

Tab + class B

- Second point

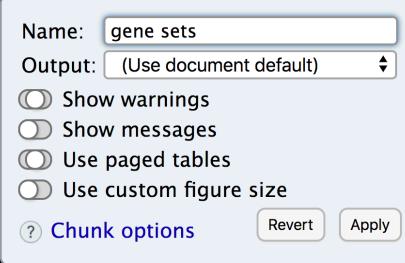
Will result in:

- First point
 - class A
 - class B
- Second point

Code chunks

```
```{r gene sets, message = FALSE}
head of gene sets
head(geneIds(tgfb_gs_up))
head(geneIds(tgfb_gs_dn))
```
```

```
```{r gene sets, message=FALSE}
head of gene sets
head(geneIds(tgfb_gs_up))
head(geneIds(tgfb_gs_dn))
```
[1] "19"   "87"   "182"  "
[1] "136"  "220"  "224"  "
```



The screenshot shows the RStudio interface with a code chunk in the editor. A context menu is open over the code, displaying options for the chunk. The 'Name:' field is set to 'gene sets'. The 'Output:' dropdown is set to '(Use document default)'. There are four radio buttons for 'Show warnings', 'Show messages', 'Use paged tables', and 'Use custom figure size', all of which are unselected. At the bottom of the menu are 'Chunk options', 'Revert', and 'Apply' buttons.

```
```{r set-up}
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
```
```

Insert/run code chunks

A screenshot of the RStudio interface. The top panel shows an R script with the following code:

```
1 ---  
2 title: "RNotebook_example"  
3 output: html_notebook  
4 ---  
5 This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the  
notebook, the results appear beneath the code.  
6 ---{r}  
7 plot(cars,  
8 pch = 19,  
9 col = "orange red",  
10 main = "Example plot")  
11 ---  
12 ---
```

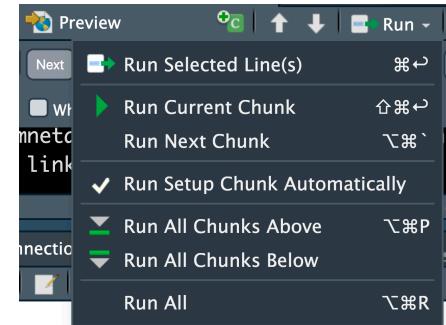
The bottom panel displays a scatter plot titled "Example plot". The x-axis is labeled "speed" and ranges from 5 to 25. The y-axis is labeled "dist" and ranges from 0 to 120. The plot contains several orange-red data points.



- In Mac: Command + Option + i
- In Windows: Ctrl + Alt + i

Insert/run code chunks

A screenshot of the RStudio interface. On the left, there is a code editor window containing R Markdown code. The code includes a title, output type, and a note about the notebook. It then contains a code chunk that plots a scatter of cars' speed versus distance. The resulting scatter plot is titled "Example plot" and shows a positive correlation between speed and distance. The x-axis is labeled "speed" and ranges from 5 to 25. The y-axis is labeled "dist" and ranges from 0 to 120. The data points are orange-red. At the bottom of the code editor, the number "13" is visible.



Interactive tables using DT package

- Interface to the DataTable javascript library
- Very easy-to-use: `datatable(df)`
- filtering, paging, sorting, formatting the tables, etc.
- DT documentation

```
library(rn)
library(rshape2)
data("tips")

datatable(tips, filter = "top", options = list(pageLength = 12)) %>%
  formatStyle("total_bill",
    fontWeight = styleInterval(18, c('normal', 'bold')))%>% ## bold some numbers!
  formatStyle('tip',
    backgroundColor = styleColorBar(tips$tip, 'mediumpurple'),
    backgroundRepeat = 'no-repeat',
    backgroundPosition = 'center'
  )%>% ## transform values
  formatStyle('size',
    transform = 'rotateX(-45deg) rotateY(-30deg) rotateZ(-50deg)',
    backgroundColor = styleEqual(unique(tips$sex), c('lightblue', 'lightseagreen')))%>% ## colour value/background
  formatStyle('size',
    color = styleInterval(c(2, 4), c('blue', 'black', 'red')),
    backgroundColor = styleInterval(c(2, 4), c('white', 'gray', 'gray50')))
```

| Show 12 entries | | | | | | | Search: |
|-----------------|------------|------|--------|--------|-----|--------|---------|
| | total_bill | tip | sex | smoker | day | time | size |
| 1 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| 2 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| 3 | 21.01 | 3.5 | Male | No | Sun | Dinner | 3 |
| 4 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| 5 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |
| 6 | 25.29 | 4.71 | Male | No | Sun | Dinner | 4 |
| 7 | 8.77 | 2 | Male | No | Sun | Dinner | 2 |
| 8 | 26.88 | 3.12 | Male | No | Sun | Dinner | 4 |
| 9 | 15.04 | 1.96 | Male | No | Sun | Dinner | 2 |
| 10 | 14.78 | 3.23 | Male | No | Sun | Dinner | 2 |
| 11 | 10.27 | 1.71 | Male | No | Sun | Dinner | 2 |
| 12 | 35.26 | 5 | Female | No | Sun | Dinner | 4 |

Showing 1 to 12 of 244 entries

Previous 1 2 3 4 5 ... 21 Next

Interactive plots using plotly package

- Plotly can generate 2D and 3D plots, as well as animations
- It is possible to zoom, pan, label, and toggle between items in the legend
- Save static image functionality
- Configurable tooltips
- Very easy-to-use with **ggplot**: `ggplotly(ggplot_object)`
- Plotly documentation

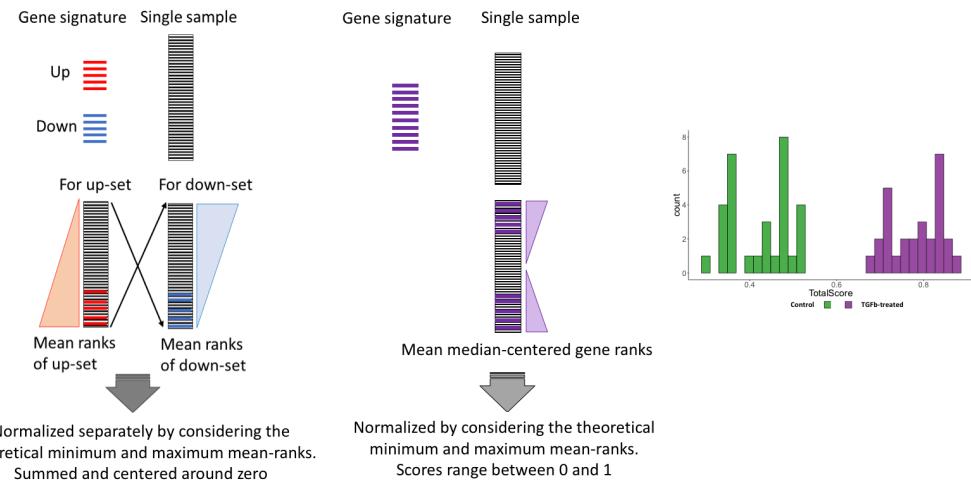
Back to our theoretical project...

Use samples' gene expression data and a gene set scoring method to score samples against gene sets and identify those that are more concordant with a given signature.



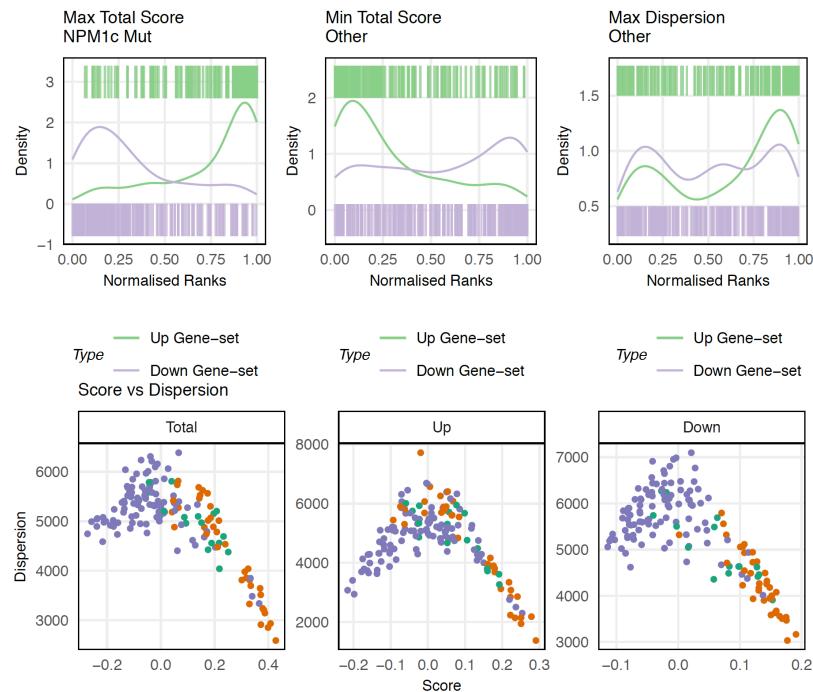
- Rank-based method
- Single-sample approach
- R/Bioconductor package
- Interactive plots

The singscore method



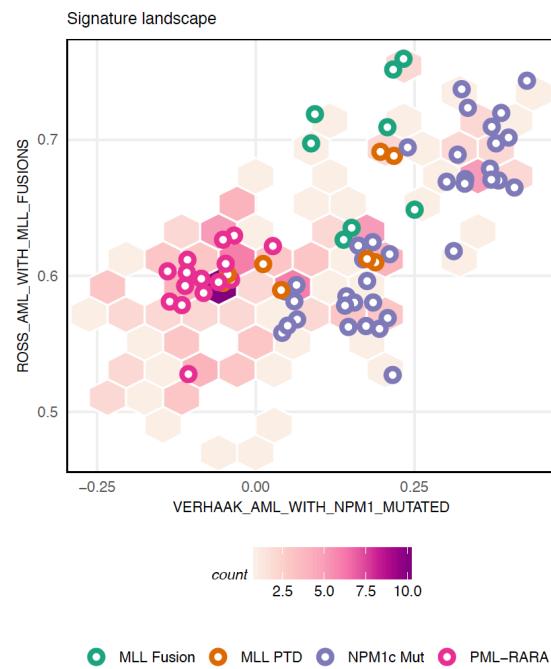
Single sample scoring of molecular phenotypes. Foroutan M, Bhuva D, et al. *BMC Bioinformatics*

Visualisations by singscore



Using singscore to predict mutations in AML from transcriptomic signatures. Bhava D. et al. *f1000 research*.

Visualisations by singscore



Using singscore to predict mutations in AML from transcriptomic signatures. Bhuva D. et al. *f1000 research*.

Let's look at our R Markdown report

Generating dynamic reports in R Markdown - an example using singscore

Resources I used for this presentation

- **RStudio website**
- **R Markdown documentation**
- **R Notebook documentation**
- **Baby one more time - Reproducibility in R and when to pull in the big guns** by *Lavinia Gordon*
- **RLadies presentation Ninja** by *Alison Presmanes Hill*
- **Making slides in R Markdown** by *Alison Hill*

Thank you!

38 / 38