

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.

 [Start a new group](#)



Part of **R-Ladies** – 157 groups

R-Ladies Melbourne

Melbourne, Australia
1,213 members · Public group
Organized by R-Ladies G. and 11 others

Share: [f](#) [t](#) [in](#)

[About Meetup](#) [Events](#) [Members](#) [Photos](#) [Discussions](#) [More](#) [Join this group](#) [...](#)

What we're about
Australia's first R Programming community for women & supporters! A local chapter of R-Ladies Global, R-Ladies Melbourne exists to promote gender diversity in the R community, both in the Australia and worldwide.

Organisers
 R-Ladies G. and 11 others
[Message](#)

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.

The screenshot shows the RStudio interface with the following components:

- Console:** Displays R startup messages and a command-line session:

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> plot(cars)  
>
```
- Code Editor:** Shows R code in a script file named "check.Rpres":

```
35 source("./script/RLE_ggplot.R")  
36 source("./script/PCA_plot.R")  
37 source("./script/Ftest.R")  
38  
39  
40 sigPath <- "/Users/foroutanm/Documents/data/signatures/"  
41 geneAnnotPath <- "/Users/foroutanm/Documents/data/geneAnnotation/"  
42 MSigDB_Path <- "/Users/foroutanm/Documents/data/signatures/MSigDB/"  
43  
44 cols <- c(  
45 brewer.pal(8, "Dark2")[-5],  
46 brewer.pal(10, "Paired"),  
47 brewer.pal(12, "Set3"),  
48 brewer.pal(9, "Blues")[c(8, 3, 7, 4, 6, 9, 5)],  
49 brewer.pal(9, "Oranges")[c(8, 3, 7, 4, 6, 9, 5)],  
50 brewer.pal(10, "Greens")[c(8, 3, 7, 4, 6, 9, 5)]  
62:102 C Chunk 2 : R Markdown
```
- Plot Viewer:** Displays a scatter plot of "dist" versus "speed" for the "cars" dataset.
- Bottom Navigation:** Includes tabs for Environment, History, Connections, and Presentation, along with a search bar.

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 ...

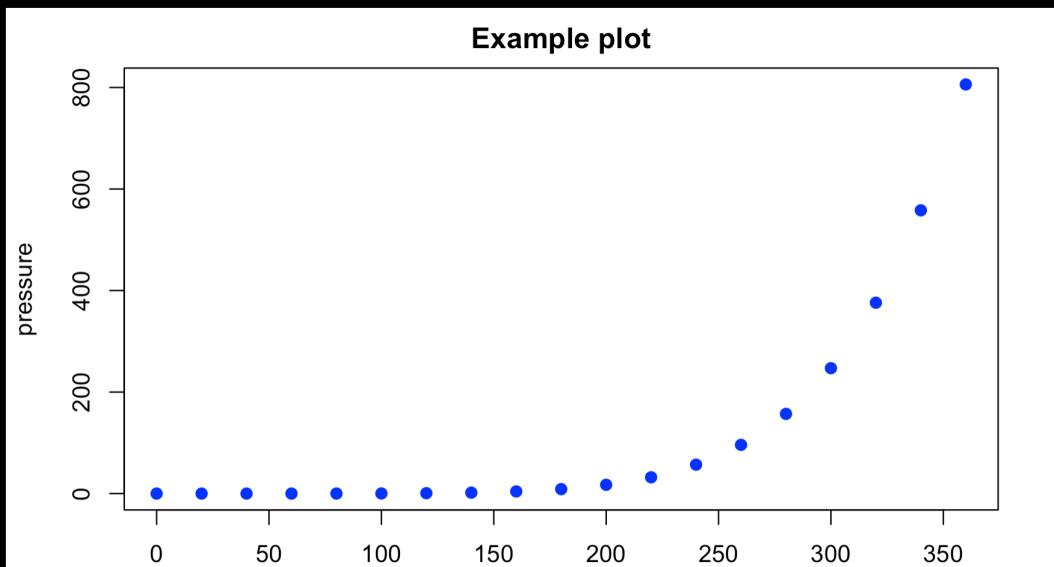
- inspired by Sweave
- is an R package which works as an engine for dynamic report generation in R.
- helps us to integrate R codes into other documents (e.g. Markdown, LaTeX, HTML, etc).
- can generate HTML, PDF or Word documents.
- supports other languages, such as Python, Perl, C++, Shell scripts,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

R Markdown main sections

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

```
---
```

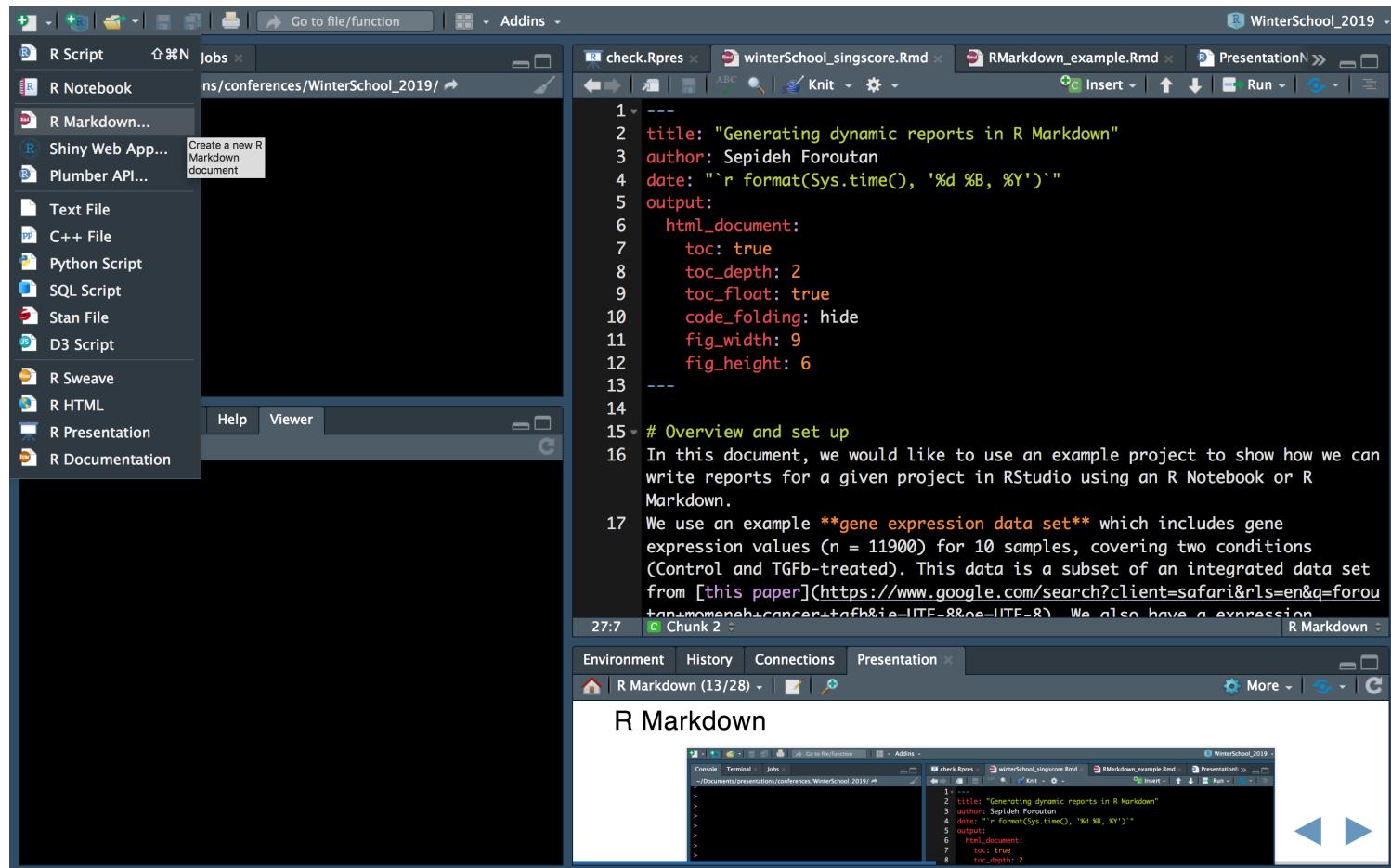
```
title: "RMarkdown_example"
output: html_document
---
```

```
# 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")
```


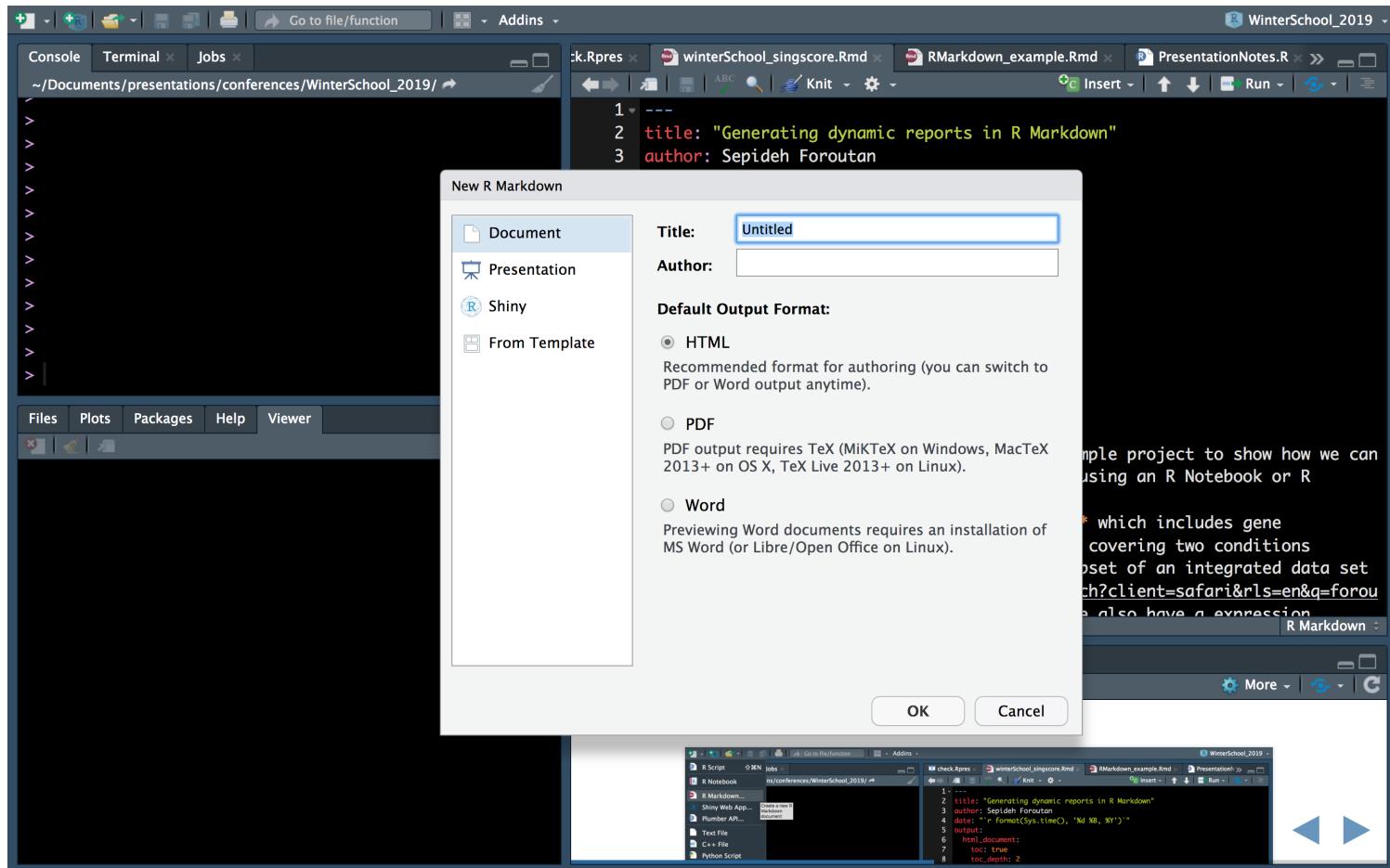
distance	pressure
0	0
20	0
40	0
60	0
80	0
100	0
120	0
140	0
160	0
180	0
200	20
220	40
240	60
260	100
280	180
300	250
320	400
340	580
350	820


```

Open a new R Markdown



Open a new R Markdown



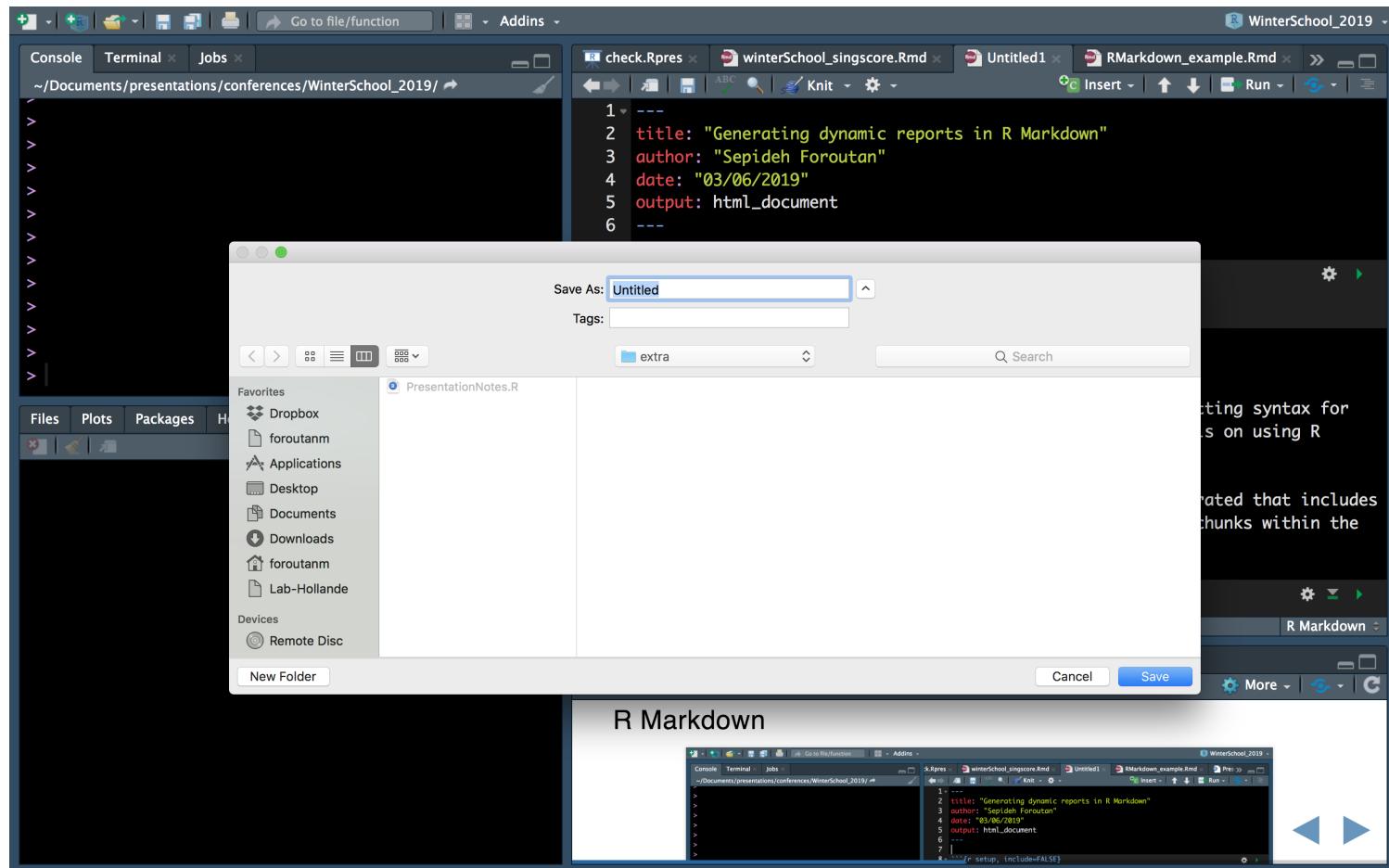
Open a new R Markdown

The screenshot shows the RStudio interface with a new R Markdown document open. The top menu bar includes 'Console', 'Terminal', 'Jobs', 'Go to file/function', 'Addins', and a tab for 'WinterSchool_2019'. The main workspace displays the following R Markdown code:

```
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  
12 ## R Markdown  
13  
14 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 bottom pane shows the 'Presentation' view with the title 'R Markdown' and the rendered code from the document.

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`

A screenshot of the RStudio interface for R Markdown. The code editor shows the following R code:

```
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 ...
```

The plot area displays a scatter plot titled "Example plot" with "pressure" on the y-axis (0 to 800) and "temperature" on the x-axis (0 to 350). The data points are blue circles with a radius of 19.

- R Notebook

- preview button
- output: `html_notebook`

A screenshot of the RStudio interface for R Notebook. The code editor shows the following R code:

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

The plot area displays a scatter plot titled "Example plot" with "dist" on the y-axis (0 to 120) and "speed" on the x-axis (5 to 25). The data points are orange-red circles with a radius of 19.

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

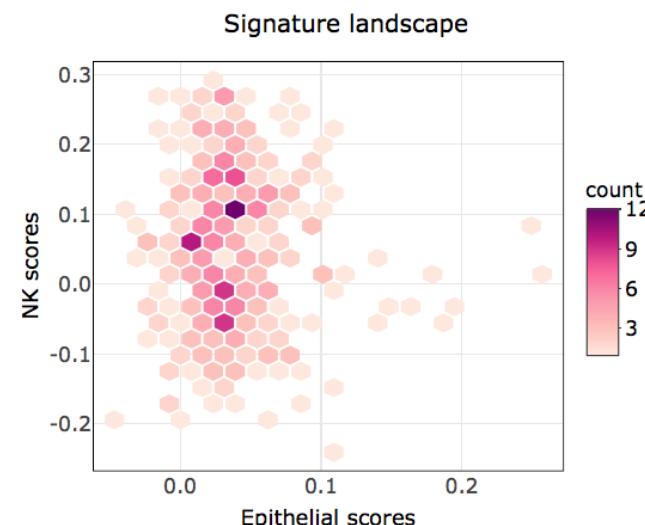
5 Session information

3 Explore the scores

3.1 Landscape plots

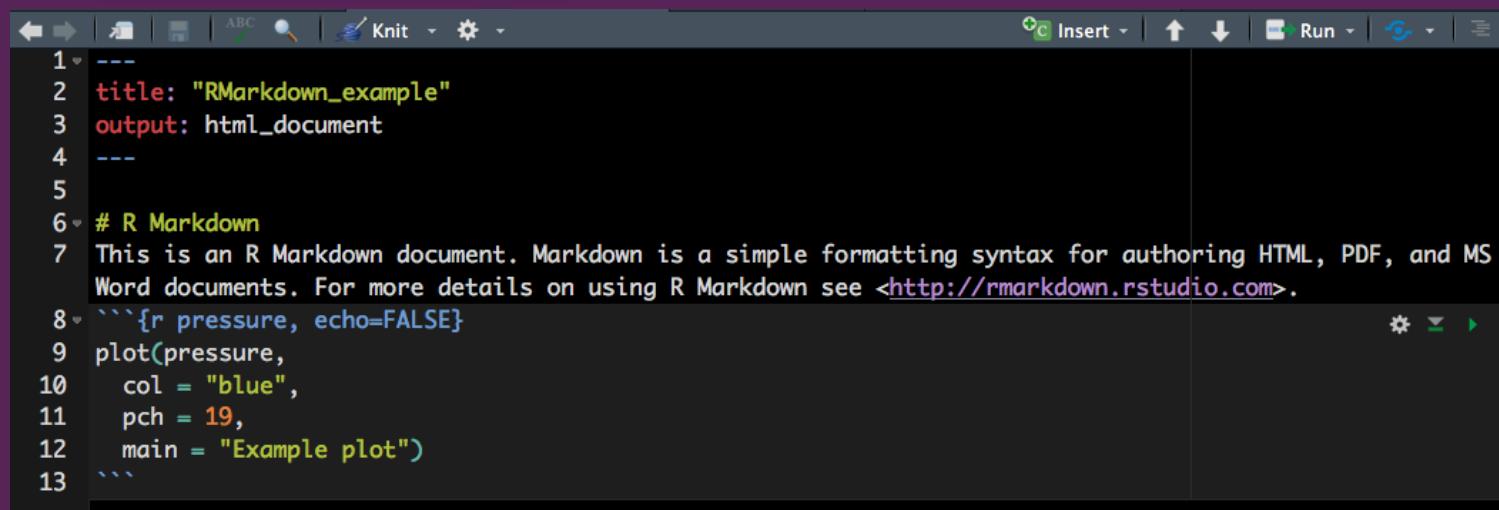
We would like to plot landscape of NK scores versus epithelial scores; to do this, we use the `plotScoreLandscape` function from `singscore` package.

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



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 IDE interface. The top bar shows various icons and buttons, including 'Knit' and 'Run'. The main workspace is a dark-themed code editor displaying an R Markdown file. The code includes a YAML header, a section titled '# R Markdown', and an R code chunk for plotting. Lines 1 through 13 are visible.

```
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 TGF β -treated). This data is a subset of an integrated data set from [this paper](#). We also have a expression signatures, called TGF β -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 TGF β -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 Bhuvan 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"
---
```

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

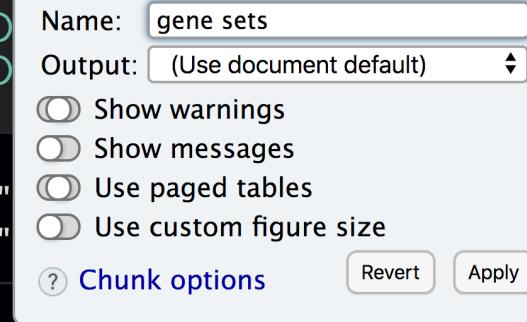
Code chunks

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



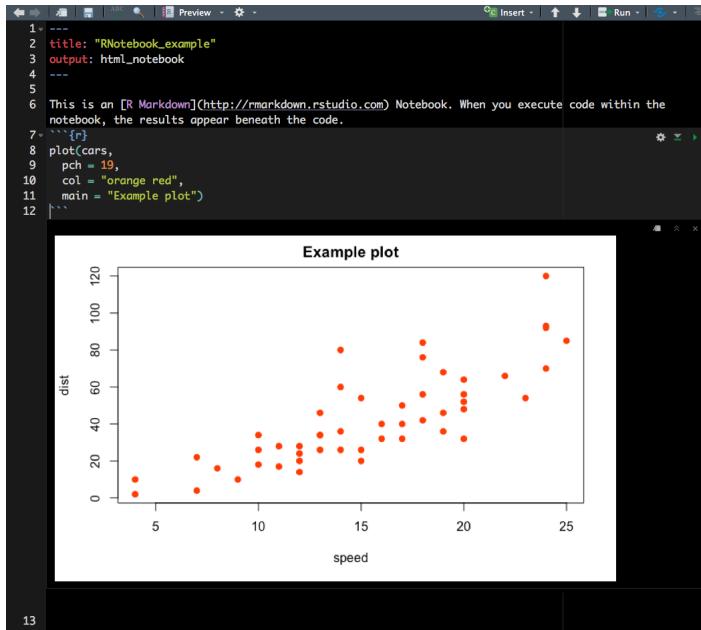
```
```{r gene sets, message=FALSE}
head of gene sets
head(geneIds(tgfb_gs_up))
head(geneIds(tgfb_gs_dn))
````
```

```
[1] "19"   "87"   "182"  "
[1] "136"  "220"  "224"  "
```



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

Insert/run code chunks



A screenshot of the RStudio interface. The top bar shows tabs for 'Preview' and 'Insert'. Below the tabs, there are buttons for 'Run' and navigation. The main area contains R code and a plot. The code is:

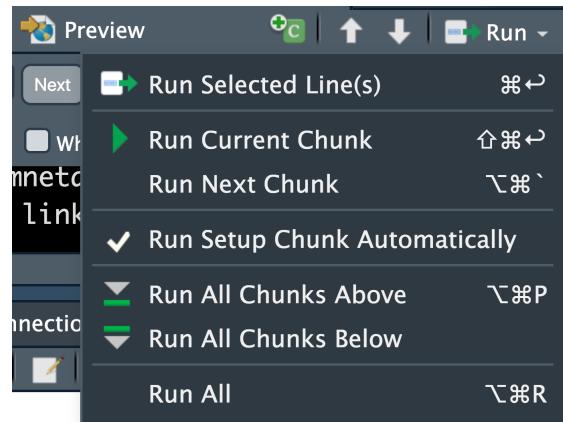
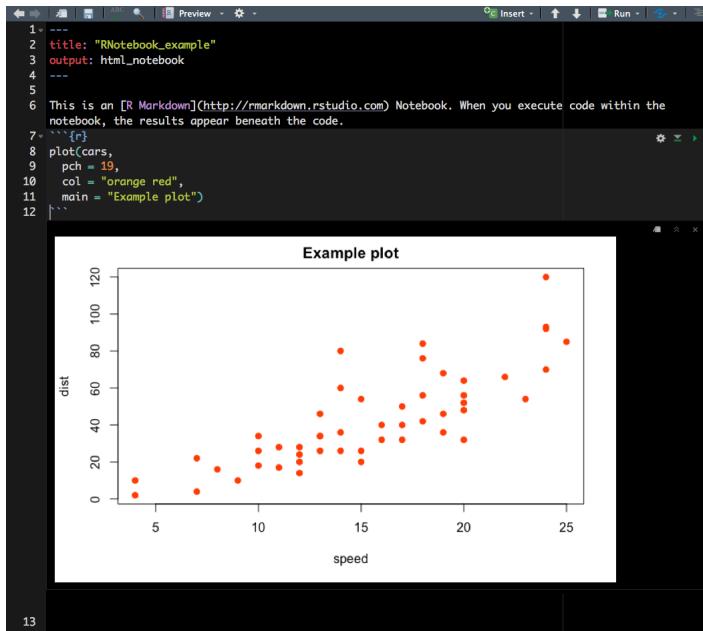
```
1 ---  
2 title: "RNotebook_example"  
3 output: html_notebook  
4 ---  
5  
6 This is an [R Notebook](http://rmarkdown.rstudio.com) Notebook. When you execute code within the  
notebook, the results appear beneath the code.  
7 ---  
8 plot(cars,  
9   pch = 19,  
10  col = "orange red",  
11  main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

The plot below the code is titled 'Example plot'. It is a scatter plot of 'dist' versus 'speed'. 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 circles.



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

Insert/run code chunks



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(DT)
library(reshape2)
data("tips")

datatable(tips, filter = "top", options = list(pageLength = 12)) %>%
  formatStyle('total_bill',
    fontWeight = styleInterval(18, c('normal', 'bold'))) %>%
  ## show colour bar
  formatStyle('tip',
    background = styleColorBar(tips$tip, 'mediumpurple'),
    backgroundColor = '100% 95%',
    # backgroundRepeat = 'no-repeat',
    # backgroundPosition = 'centre'
  ) %>%
  ## transform values
  formatStyle('sex',
    transform = "rotateX(-45deg) rotateY(-30deg) rotateZ(-50deg)",
    backgroundColor = styleEqual(unique(tips$sex), c('lightblue', 'lightseagreen'))) %>%
  ## colour value/background
  nd
  formatStyle('size',
    color = styleInterval(c(2, 4), c('blue', 'black', 'red')),
    backgroundColor = styleInterval(c(2, 4), c('white', 'gray', 'gray50')))
```

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

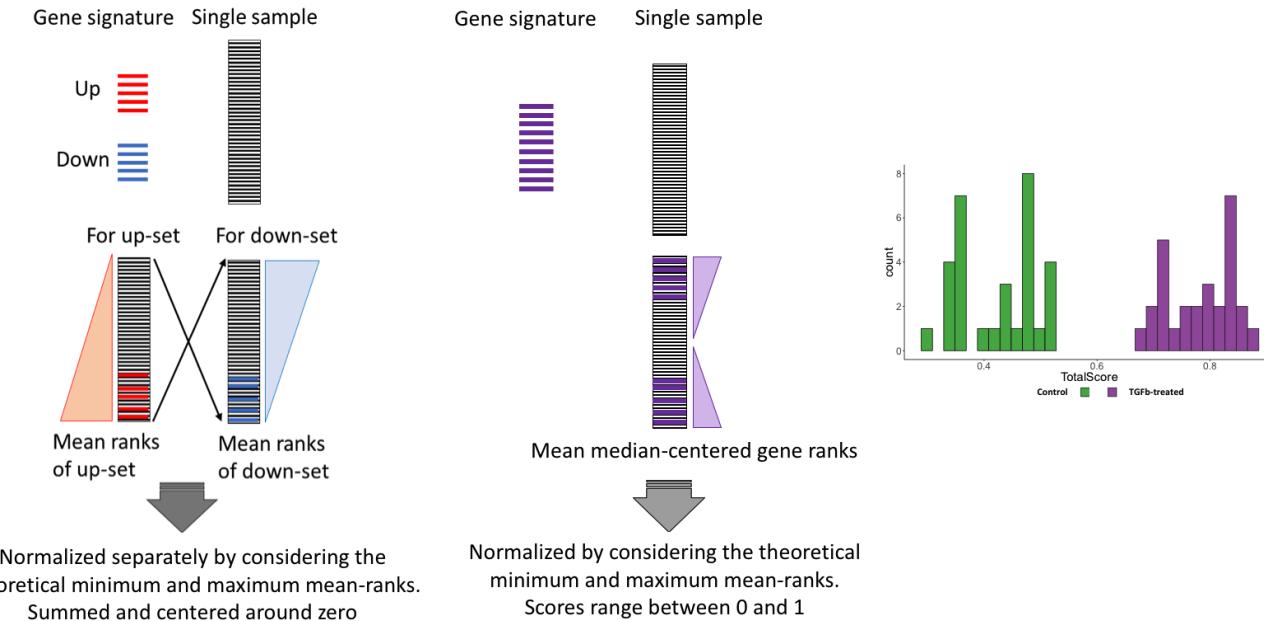
Back to our theoretical project...

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.



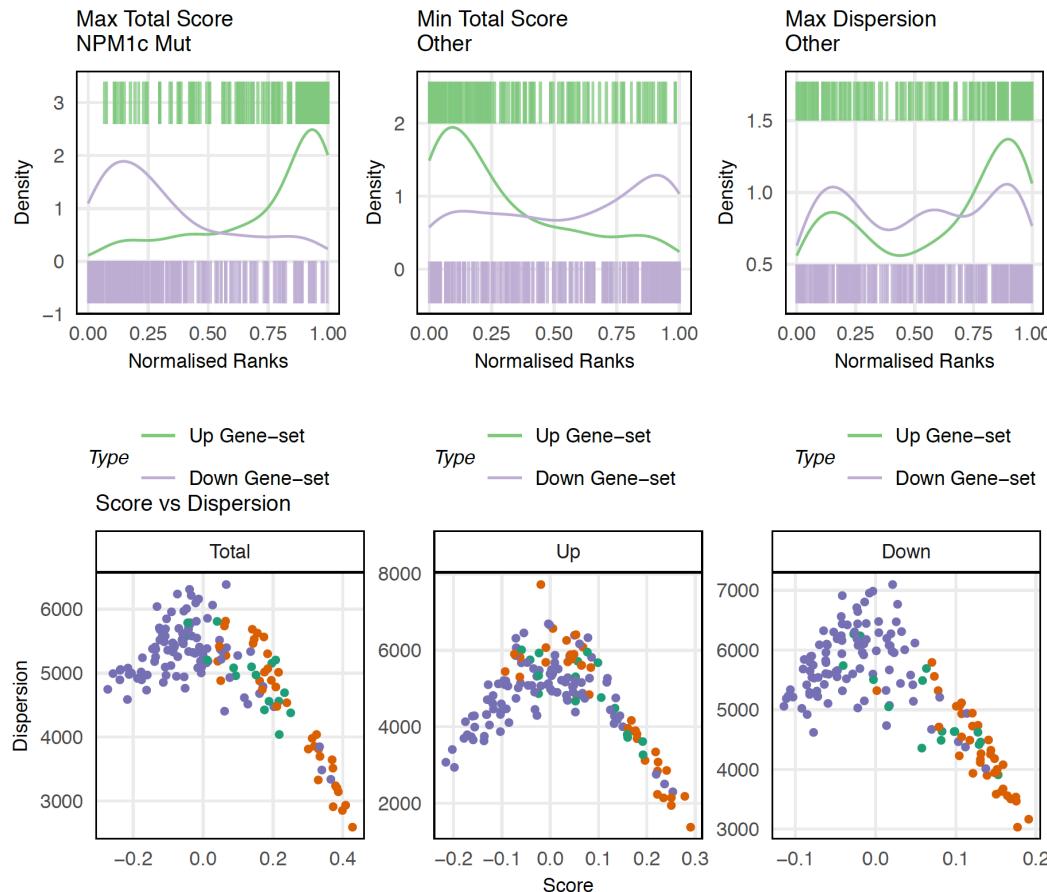
- Rank-based method
- Single-sample approach
- R/Bioconductor package
- Interactive plots

The singscore method



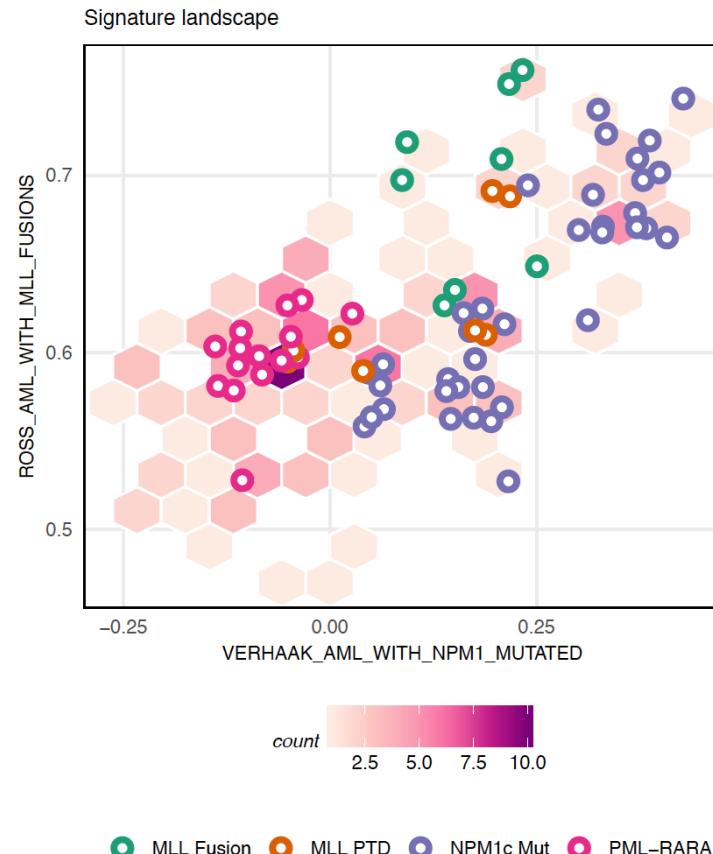
Single sample scoring of molecular phenotypes. Foroutan M, Bhuva D, et al. *BMC Bioinformatics*

Visualisations by singscore



Using singscore to predict mutations in AML from transcriptomic signatures. Bhuva 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!