Why would you care about mathematics as a data scientist? & An introduction to R Markdown

MATH 2740 – Mathematics of Data Science – Lecture 02

Julien Arino
julien.arino@umanitoba.ca

Department of Mathematics @ University of Manitoba

Fall 202X

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis. We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

Outline

Mathematics of data science?

Introduction to R Markdown

Mathematics of data science?

Introduction to R Markdown

In days of yore (circa 2010)

> *Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it...*

Attributed to Dan Ariely (Duke University)

The vocabulary has evolved, big data $\rightarrow$ complex data $\rightarrow$ data science, but Data Science remains a loosely defined concept, although things are becoming better

Data Science (according to Wikipedia)

Data science is an **interdisciplinary field** that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from **structured** and **unstructured data**, and apply knowledge and actionable insights from data across a broad range of application domains.

[..] It uses techniques and theories drawn from many fields within the context of **mathematics**, **statistics**, **computer science**, **information science**, and **domain knowledge**.

The data deluge

► Data science is nothing new (some statisticians argue it is just another name for statistics), but it has become prominent in recent years as a consequence of the unprecedented mass of information generated and collected by our modern societies

► One speaks of *information explosion* or *data deluge*. See some considerations, e.g., here

A wide variety of jobs

We have absolutely insane amounts of data and we try to make sense of it

$\Longrightarrow$ data science

However, except for the name, the situation has not improved significantly since the days of yore of Ariely's quote: data science is a hodge-podge that contains everything but the kitchen sink

To caricature

- two main types of data: structured and unstructured
- two main branches: statistics and computer science
- two main types of jobs: users and developpers

Math of Data Science?

- DS has two main branches: statistics and computer science
- DS has two main types of jobs: users and developpers

So why a course on Math of Data Science?

If you plan to be a user and are not curious about *the how* and *the why* and can tolerate errors due to misuse of methods, then you probably don't care about this course

In other cases, many of the concepts used have their roots in math and to understand where the methods are coming from and, even more importantly, to develop new methods, math is often required

Warning! We barely brush the surface

- Some techniques from linear algebra
- Some graph theory ideas
- A little bit of multivariable calculus

There is a lot more to see!!!

Prerequisites / What you will learn

► (MATH 1210 or MATH 1220 or MATH 1300) and (MATH 1232 or MATH 1700 or MATH 1710)

⇒ You **must know and be comfortable** with 1st year linear algebra (3 CH) and 1st year calculus (6 CH)

► We need more: some stuff you would learn in 2090 (Linear Algebra 2), some stuff from 2130, 2150 or 2720 (Multivariable Calculus) and some stuff from 2070 (Graph Theory)

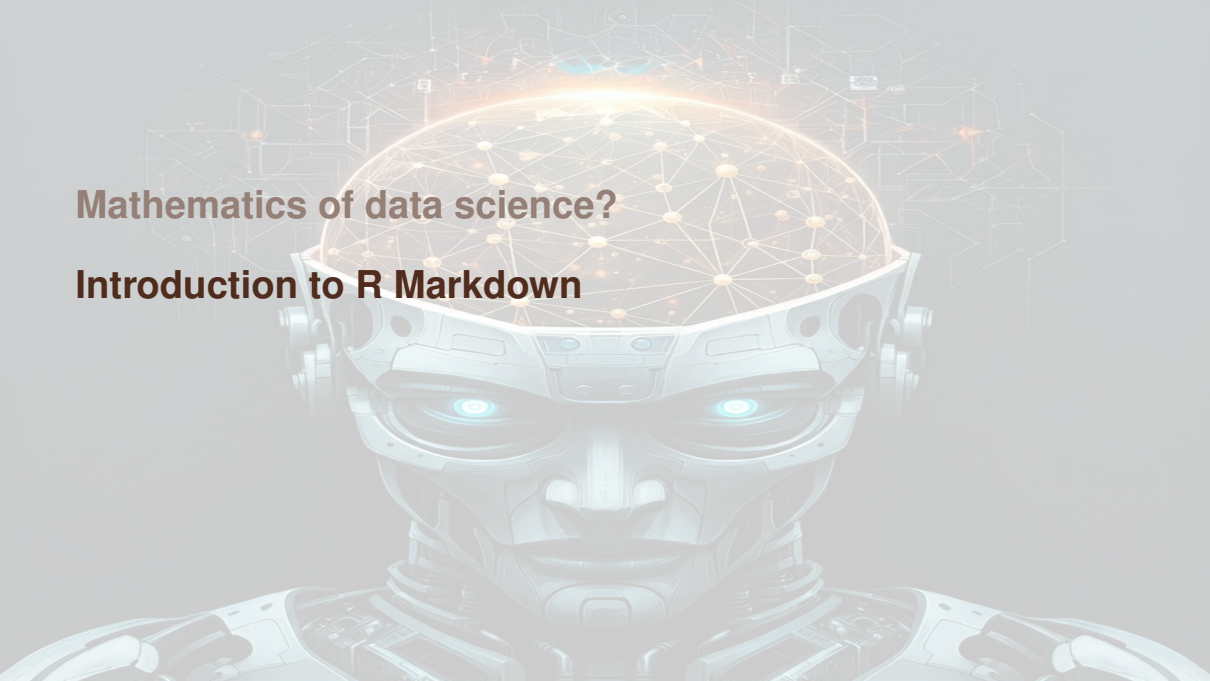
Focus here is not on mathematical “precision”

- We won’t do complicated proofs. I will show some when they are useful in understanding *why* something works

- We will cover just enough of the mentioned math topics that you can understand *how* to do things

- In classic math courses, we work on “small” examples so we can work them by hand and are able to do them in tests

Here, we will do small examples by hand, indeed. But you will do regularly sized examples in computer assignments



Mathematics of data science?

Introduction to R Markdown

Computer work (reminder from Lecture 01)

- Being able to use computers is an integral part of being a data scientist, so in this course, we use computers a lot
- The two main languages in data science are `R` and `Python`. Typically, `R` is used more by people in Stats, while `Python` is more CS
- There is great value in both and knowing both is a plus, but for simplicity, here we use `R`

Computer assignments (reminder from Lecture 01)

- Use R Markdown to generate a **notebook**
- Notebooks mix formatted text and code. They are executable and should be submitted as source, not as pdf or html or whatever. Only files in .Rmd are accepted for the computer part of the assignments
- Notebooks are not straight code. Submitting straight R code in a notebook with commented code $\Rightarrow$ 0)

R Markdown?

- File format for making dynamic documents with R
- Combines code, its results and narrative text in a single document
- Uses simple `Markdown` syntax for text and `R` code chunks for analysis
- From one '.Rmd' file, you can create HTML, PDF, Word documents, presentations, ...

Introduction to R Markdown

The YAML Header

Markdown 101

R code chunks

Instructions for the computer assignments

The YAML header

Every R Markdown document starts with a YAML header, enclosed by `---`. This controls the overall properties of the document

Example YAML for a PDF document

```
---
title: "My Report"
author: "My Name"
date: "2025-08-17"
output:
  pdf_document:
    toc: true
    number_sections: true
---
```

Introduction to R Markdown

- The YAML Header
- **Markdown 101**
- R code chunks
- Instructions for the computer assignments

Basic Text Formatting

Markdown provides a simple syntax for formatting text

- `# Header` creates a section title, `## Sub-header` creates a subsection title, etc.
- `*italic text*` produces *italic text*
- `**bold text**` produces **bold text**
- `` `code font` `` produces `code font`

Lists

Unordered lists start with * or -

```
* Item 1
* Item 2
```

Ordered lists use numbers

```
1. First Item
2. Second Item
```

(Once the numbered list is “initiated”, the number doesn’t matter, you can write 1., 1., 1., etc., provided you have a new line each time)

Introduction to R Markdown

- The YAML Header
- Markdown 101
- **R code chunks**
- Instructions for the computer assignments

Chunks: the core of R Markdown

R code is placed in “chunks”, which start with ```` ```{r chunk-name} ```` and end with ```` ``` ````

````
```{r cars-summary}
# Your R code goes here
summary(cars)
```
````

Chunk names (`cars-summary` here) are not mandatory (you could write ‘‘‘`{r}` but are very useful, in particular when debugging

Chunk output

When the document is "knit," the R code is executed and its output is embedded in the final document

````
```{r cars-summary}
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
```

Controlling chunks with options

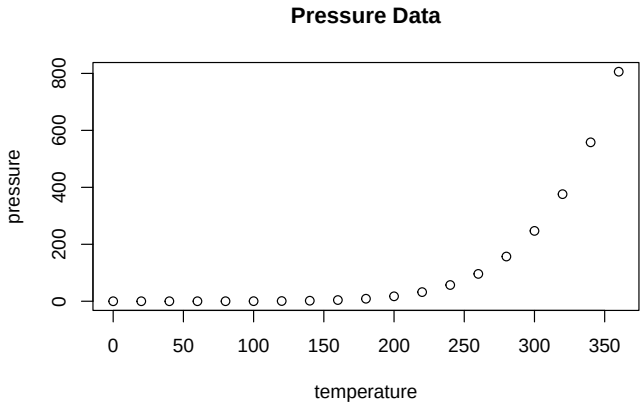
Chunk behavior is controlled by options inside the curly braces

- `echo=FALSE` hides the `R` code, but shows the output
- `eval=FALSE` shows the code, but does not execute it
- `include=FALSE` executes the code, but hides both code and output
- `warning=FALSE`, `message=FALSE` hides warnings or messages
- `fig.width=5`, `fig.height=4` sets figure dimensions

Figure chunks

Plots are automatically embedded

```
```{r pressure-plot, echo=FALSE, fig.cap="Pressure Data"}
plot(pressure)
```
```



Inline R Code

You can embed R code directly into text with backticks: `` `r ...` ``

```
The `cars` dataset has `r nrow(cars)` rows.
```

The cars dataset has 50 rows.

Introduction to R Markdown

- The YAML Header
- Markdown 101
- R code chunks
- **Instructions for the computer assignments**

Sample code for the assignments

Please compare the results obtained when knit-ing these two files

- Proper notebook: CODE & output
- Comment heavy notebook: CODE & output

The first one will get you good marks. The second one will get you in hot water with the marker: you *might* get one warning salvo, but afterwards, a lot of marks will be deducted

The chunk option that will avoid issues

Compare the first option chunk in the two iris files

````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE, fig.width = 10,
     fig.height = 6)
```
````

````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE, fig.width =
    10, fig.height = 6)
```
````

I recommend that you play with the `echo =` option. Set it to `FALSE` and judge your verbosity... Are you explaining enough with the code (and your comments) not shown?