INTRODUCTION TO R MARKDOWN

ACTS-DBDA

Sep 2021, CDAC Bangalore

WHAT IS R MARKDOWN?

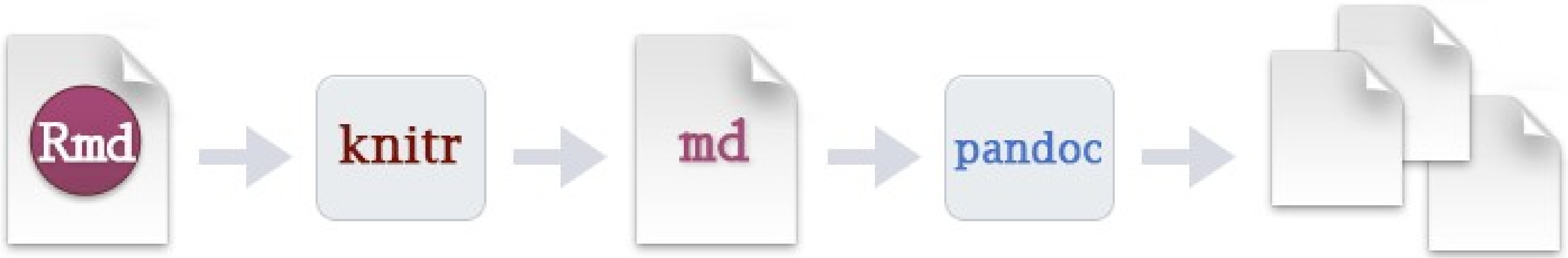
- **R Markdown allows you to create documents which has detailed record of your analysis.**
- It also serves other researches/developers to understand the what we did in our analysis.
- You can also upload your document to github or publish to Rpubs and R open source community.
- It presents your code alongside its output (graphs, tables, etc.) with conventional text to explain it, a bit like a **notebook**.

WHAT IS R MARKDOWN?

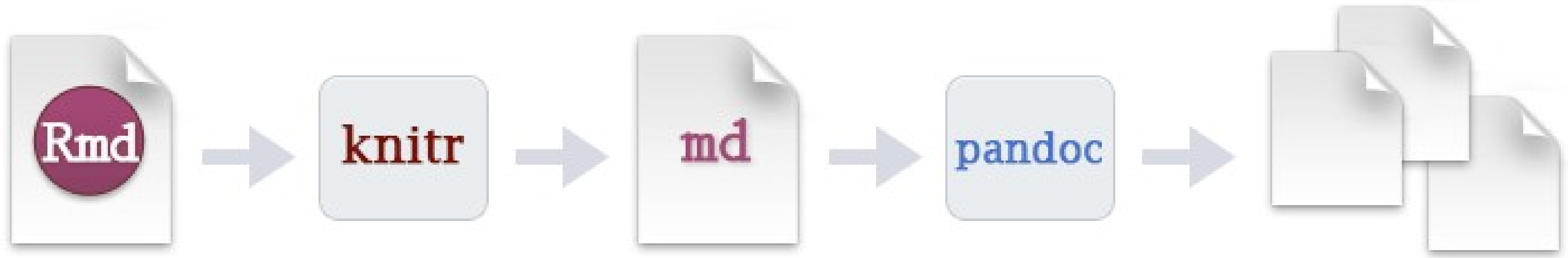
- R Markdown uses Markdown syntax like Markup language.
  - Example: headers, images, links etc
- You can convert Markdown documents to many other file types like .html or .pdf to display the headers, images etc..

HOW IT WORKS?

- R Markdown generates a new file that contains selected text, code, and results from the .Rmd file.
- The new file can be a finished **web page**, **PDF**, **MS Word**, **document**, **slide show**



HOW IT WORKS?



- **Knitr** executes all code chunks and creates .md document which includes code and its output.
- The markdown file generated by knitr is then processed by **pandoc** which is responsible for creating the finished format.
- It is done internally using `render()` function

DOWNLOAD R MARKDOWN

- To get RMarkdown working in RStudio, the first thing you need is the rmarkdown package, which you can get from CRAN by running the following commands in R or Rstudio:

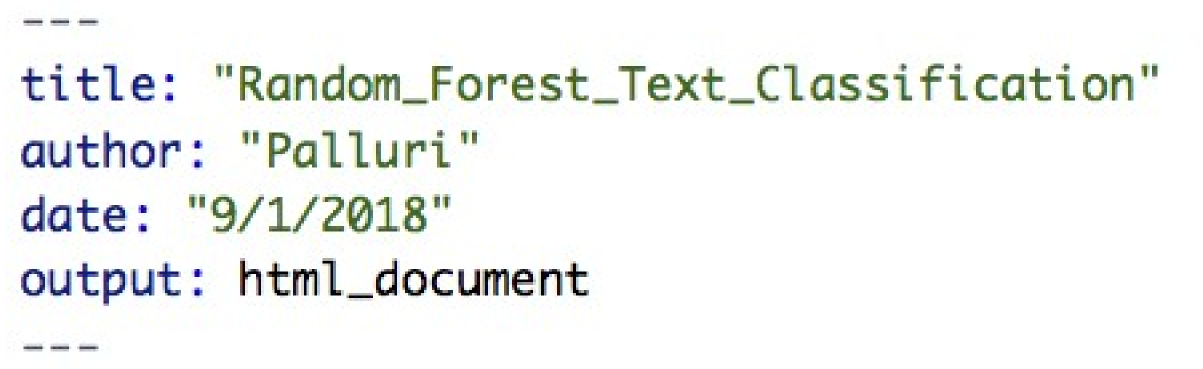
  - `install.packages("rmarkdown")`
  - `library(rmarkdown)`

CREATE AN RMARKDOWN FILE

- To create a new RMarkdown file (.Rmd), select File -> New File -> R Markdown in Rstudio.
- Then choose file type you want to create. (We used “.html” to create markdown)
- By default, newly created file will have certain template which has basic instructions.

PARTS OF MARKDOWN FILE

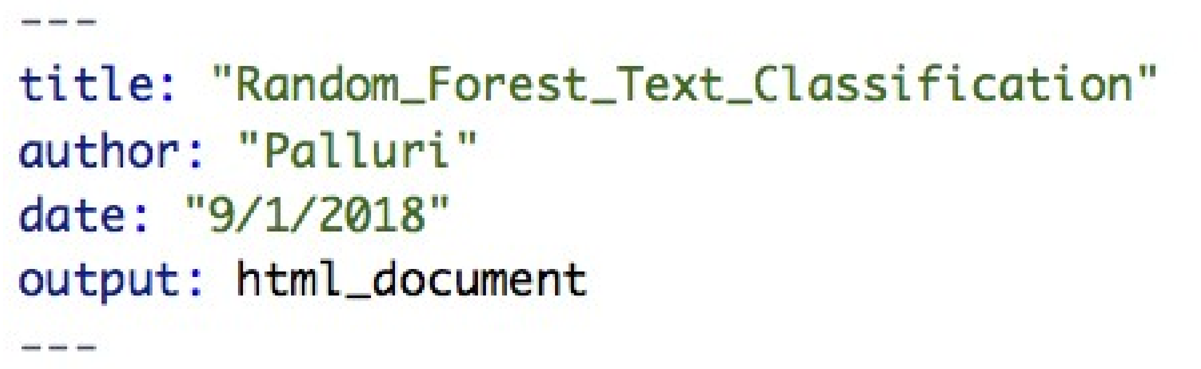
- At the top of any RMarkdown script is a YAML header section enclosed by ---
  - YAML is a kind of configuration file
- By default, the **title**, **author**, **date** and **output** format are printed at the top of your .html document.



```
---
title: "Random_Forest_Text_Classification"
author: "Palluri"
date: "9/1/2018"
output: html_document
---
```

PARTS OF MARKDOWN FILE

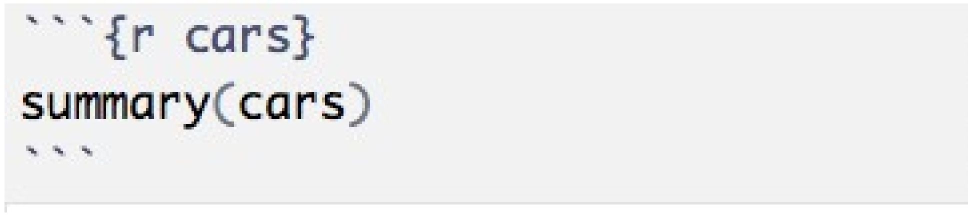
- At the top of any RMarkdown script is a YAML header section enclosed by ---
  - YAML is a kind of configuration file
- By default, the **title**, **author**, **date** and **output** format are printed at the top of your .html document.
- Save and **Knit** it



```
---
title: "Random_Forest_Text_Classification"
author: "Palluri"
date: "9/1/2018"
output: html_document
---
```

PARTS OF MARKDOWN FILE

- Code Chunks
  - Below the YAML header is the space where you will write your code.
  - Code that is included in your .Rmd document should be enclosed by three backwards apostrophes ``` . These are known as code chunks and look like this:

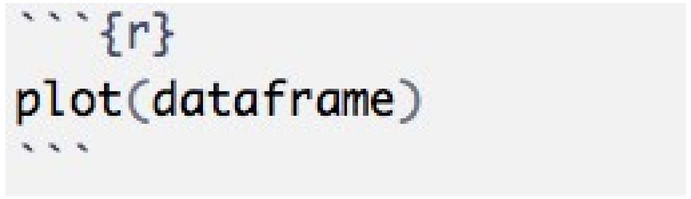


~~~
```{r cars}
summary(cars)
```
~~~

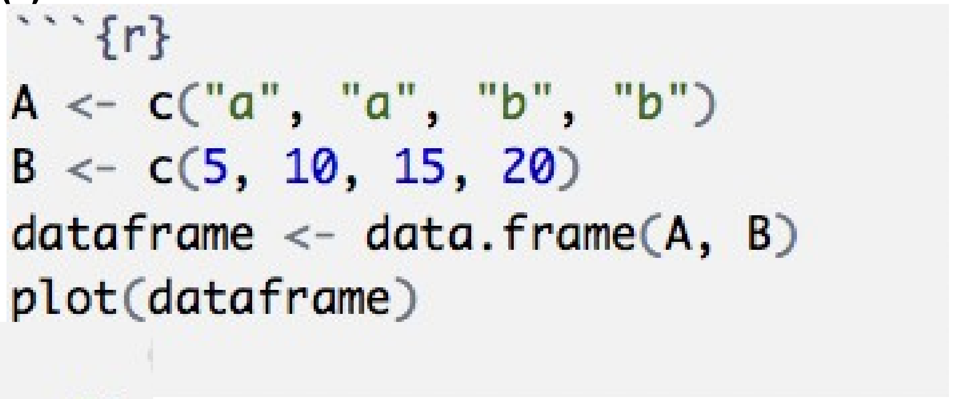
  - Run -> Run Current Chunk (Individual Chunk of code to RUN)

PARTS OF MARKDOWN FILE

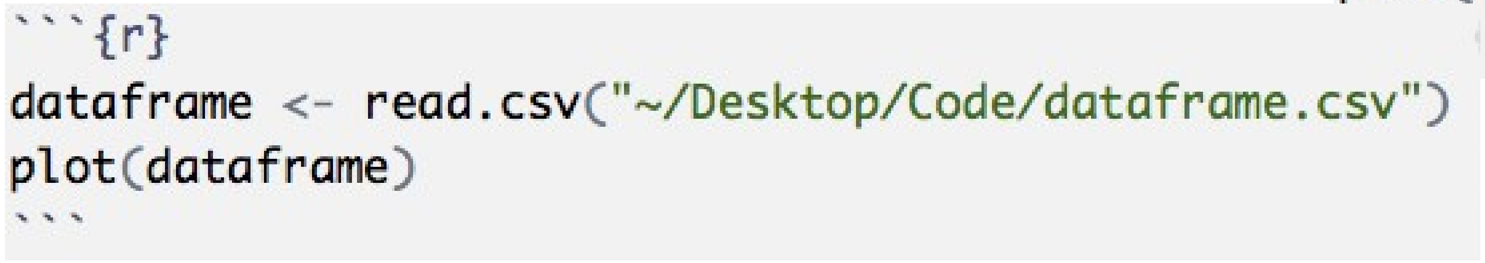
- Code Chunks
  - In R Markdown, if you want to run code that refers to an object, you must include instructions showing what dataframe is
  - Similarly, if you are using any packages in your analysis, you will have to load them in the .Rmd file using library()



~~~
```{r}
plot(dataframe)
```
~~~



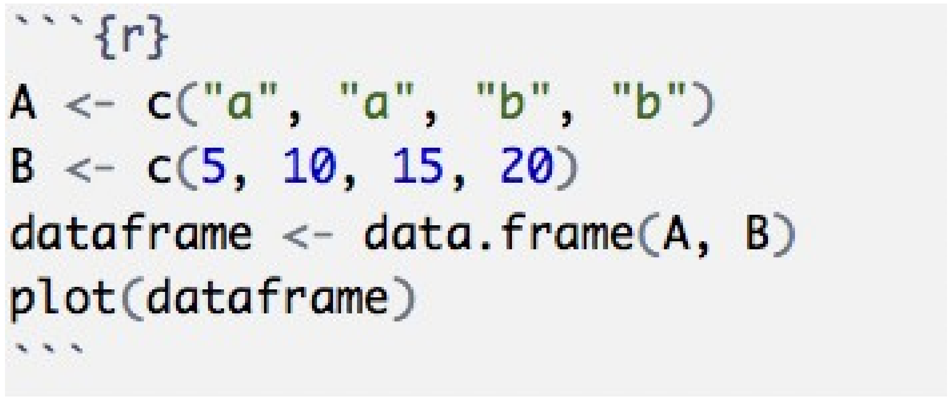
~~~
```{r}
A <- c("a", "a", "b", "b")
B <- c(5, 10, 15, 20)
dataframe <- data.frame(A, B)
plot(dataframe)
~~~



~~~
```{r}
dataframe <- read.csv("~/Desktop/Code/dataframe.csv")
plot(dataframe)
```
~~~

PARTS OF MARKDOWN FILE

- Hiding code chunks



~~~
```{r}
A <- c("a", "a", "b", "b")
B <- c(5, 10, 15, 20)
dataframe <- data.frame(A, B)
plot(dataframe)
```
~~~

~~~
```{r, echo = FALSE}
A <- c("a", "a", "b", "b")
B <- c(5, 10, 15, 20)
dataframe <- data.frame(A, B)
plot(dataframe)
```
~~~

- **You MUST load all objects and packages in the R Markdown script**.

PARTS OF MARKDOWN FILE

- code chunks

Rule	Example (default)	Function
eval	eval=TRUE	Is the code run and the results included in the output?
include	include=TRUE	Are the code and the results included in the output?
echo	echo=TRUE	Is the code displayed alongside the results?
warning	warning=TRUE	Are warning messages displayed?
fig.width, fig.height	fig.width=7	What width/height (in inches) are the plots?
fig.align	fig.align="left"	"left" "right" "center"

INSERTING CONTENT

- By default, RMarkdown will place graphs by maximizing their height, and keeps them within the margins of the page.
- You can change the default values for the image dimensions

~~~
```{r, fig.width = 2.5, fig.height = 7.5}
ggplot(df, aes(x = x, y = y) + geom_point()
~~~

```{r}
dataframe
```
```
```

INLINE CODE

- Code results can be inserted directly into the text of a .Rmd file by enclosing the code with `r `

~~~
```{r}
colorFunc <- "heat.colors”
```
~~~

~~~
```{r, fig.cap="Figure Caption", echo=TRUE}
image(volcano, col=get(colorFunc)(200))
```
~~~

CODE LANGUAGES

- Knitr can execute code in many languages besides R. Some of the available language engines include:
  - Python
  - SQL
  - Bash
  - Rcpp
  - Stan
  - JavaScript
  - CSS

TABLES

- R Markdown displays data frames and matrixes as they would be in the R terminal.
- For additional formatting of tables, you can use `knit::kable` function.
- Several other packages are also there for making beautiful tables
  - xtables
  - Stargazer
  - Pander
  - Tables

~~~
```{r, echo=FALSE
library(knitr)
Kable(mtcars[1:5, ], caption = ”A knit
table”
```
~~~

INTERACTIVE DOCUMENTS

- R Markdown documents are a perfect platform for interactive content. To make your documents interactive.

  - Interactive JavaScript visualizations based on **htmlwidgets**
  - Reactive components made with **Shiny**

- HTML Widgets are R functions that return javaScript visualizations
- HTML Widgets used **leaflet** Package in R for interactive visualizations

- Output can be seen on any web browser

REFERENCES

- R Tutorials
- Beginning R – Dr. Mark Gardener
- Relevant information from public domain

THANK YOU