

Building LaTeX Templates for R Markdown to Produce Branded PDF Reports

CSP 2020 Short Course

Ben Barnard

February 20, 2019

Contents

Course Outline	3
Abstract	3
Learning Objectives	3
About the Instructor	3
Relevance to Conference Goals	3
Getting Started	4
Introduction to R Markdown - 45 mins	5
What is R Markdown	5
Why should I use R Markdown	5
How to start an R Markdown document	5
Structure of an R Markdown document	5
PDF documents	5
Table of contents	7
Figure options	8
Data frame printing	8
Syntax highlighting	8
Latex options	8
Latex packages for citations	8
Advanced customization	8
Other features	8
Lets Start building a report	8
Building the header	8
Running R code in the yaml	9
.	9
Rendering R Markdown documents	9
Knit	9
Render function	9
Engines	9
Python	9
Shell Scripts	9
SQL	9
Rcpp	9
Stan	9
JavaScript	9
CSS	9
Julia	9
C	9
Fortran	9

Introduction to R packages - 35 mins	10
What is an R package	10
Why should I build R packages	10
Distribution	10
Testing	10
Reproducibility	10
Documentation	10
How to build your first R package	10
GitHub	10
usethis	10
Ways to distribute R packages	10
Source and binary	10
GitHub	10
drat	10
CRAN	10
Introduction to LaTeX - 60 mins	11
What is LaTeX	11
Why should I use LaTeX	11
Reproducibility	11
Automation	11
Distribution	11
Not a Microsoft product	11
How to start your first LaTeX document	11
Bringing It all together - 30 mins	12
Possible extensions - 35 mins	13

Course Outline

Abstract

Branded reports give a clean, clear and consistent message for data science teams in an organization. We walk through the process of building a latex template distributed through an R package. We begin with a short introduction to rmarkdown and some motivating examples for using branded reports. Then, we demonstrate from scratch how one can build a minimal latex template, and distribute in a R package. We describe some best practices for branding and highlight use of ggplot2 themes to match document branding. Finally, we walk through some further uses such as parameterized reports, using the template for bookdown, and recommendation for deploying the R package at your company.

Learning Objectives

The student should be able to walk away from this class with:

1. a general understanding of rmarkdown,
2. why it is important to have branded reports,
3. a R package with a latex template that uses their companies branding,
4. understanding of best practices in branding,
5. use of ggplot2 themes,
6. and some possible further uses for the using and distributing the template.

About the Instructor

Ben Barnard is a Data Scientist at Wells Fargo in the Team Member Insights group. Ben has a PhD from Baylor University in Statistics.

Jeff Idle is an Analytic Manager at Wells Fargo in the Team Member Insights group. Jeff leads the HR Advanced Analytics & Architecture team. Jeff is currently pursuing a MBA from the University of Minnesota's Carlson School of Management.

Relevance to Conference Goals

We stress using branded reports to communicate clean, clear and consistent messages to your audience. Communication is the most important part of Data Science since decision makers are rarely analytic experts. Branded reports bring a certain professionalism that will be greatly appreciated by administration. Building the latex templates saves time and makes sure every report comes out looking the same. Consistently branded reports allows your team to be recognized immediately by your work product.

Getting Started

Introduction to R Markdown - 45 mins

We discuss the R Markdown document format developed by RStudio. R Markdown is a document format, and `rmarkdown` is an R package. There can be confusion when talking about the two, and in this course we are usually talking about the R Markdown document format. Most of this portion of the class uses references from the Rstudio R Markdown website, the `rmarkdown` package website and the R Markdown book. We will walk through creating an R Markdown document that we will use for the initial template of our branded document.

What is R Markdown

R Markdown is a document format built to embed R code chunks in Markdown documents. R Markdown documents remove most of the formatting aspects of report generation to a “backend.” R Markdown can be used to generate many different output some of which are websites, pdfs, presentations, shiny, and Word documents. In this course we are going to focus on pdf documents, but in general the same R Markdown document can be used to produce all of the above with minimal changes.

Why should I use R Markdown

R markdown provides a medium to attach and run code inside the content for dashboards, articles, reports, websites, powerpoints and books. This medium provides a reproducible and portable format that works with text based version control systems. These characteristics generally make R markdown documents easy to put into production systems

How to start an R Markdown document

As previously mentioned R markdown plays well with text based version control systems, and that is because at its core an R markdown document is just a text file. R markdown files have `.rmd` extensions but you could potentially render any text file no matter the extension. If you expect the RStudio IDE to recognize an R markdown document it needs the `.rmd` extension. Let us take the easy route and create an R markdown document from the RStudio IDE menu.

Structure of an R Markdown document

```
---
title: "Untitled"
author: "Name"
date: "Date"
output: pdf_document
---
```

```
1 + 1
```

PDF documents

R markdown can be used to build pdf documents using latex. We can usually tell what R markdown documents are going to render as pdfs by output specified in the yaml header.

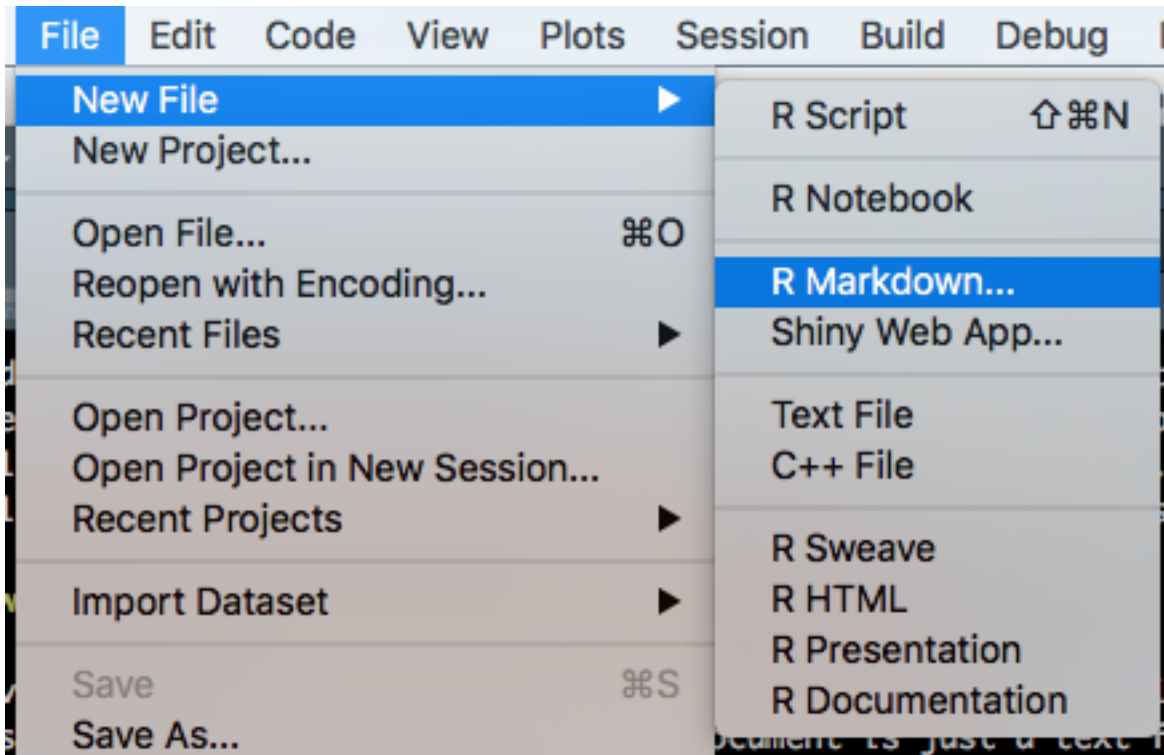


Figure 1:

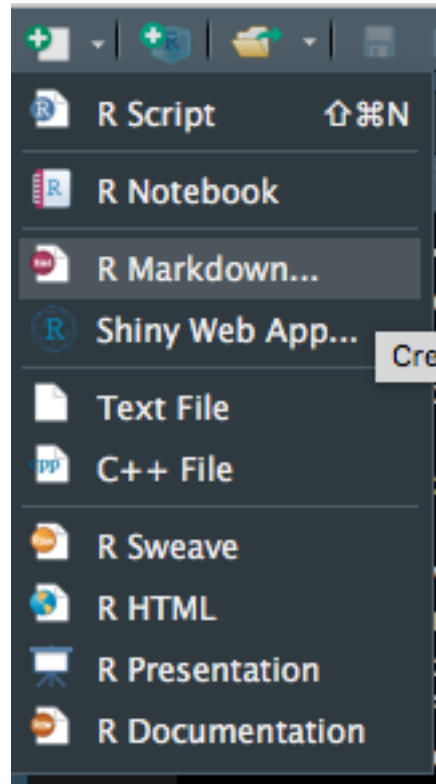


Figure 2:

```
---
title: "Untitled"
author: "Name"
date: "Date"
output: pdf_document
---
```

The `pdf_document` output option actually references a function. We are going to create our own output function that wraps the `pdf_document` function. This will end up being one of the few functions in the R package we will build. We will discuss further in the R package portion of the course. The `pdf_document` function has some options that we can specify in the yaml of our R markdown document. These next subsections discuss some of these options.

Table of contents

For longer reports and documents it is useful to have a Table of Contents. The pdf documents generated from R markdown can automatically build the table of contents for you. We can build the table of contents by setting `toc: true` in the yaml options under the `pdf_document` output. You might also notice that `toc` is a function call option in `pdf_document`, and we should expect it to be listed under `pdf_document` in the yaml.

```
---
title: "Untitled"
author: "Name"
date: "Date"
output:
  pdf_document:
    toc: true
    toc_depth: 3
---
```

Another option under `pdf_document` was added with `toc`. The yaml `toc_depth` should always be added with `toc: true`. It isn't that you can't specify it, but there really isn't much of a reason to specify it. The `toc_depth` or table of contents depth sets the depth of sections/subsections listed in the table of contents. The number specified by `toc_depth` corresponds to the number of `#` preceeding your section/subsection name in the R markdown document. If you wonder why the subsection isn't showing up in the table of contents then you should count the `#`s and see if that count matches the `toc_depth`.

Figure options

Data frame printing

Syntax highlighting

Latex options

Latex packages for citations

Advanced customization

Other features

Lets Start building a report

Building the header

```
---  
title: "Iris Petal and Sepal Length"  
author: "Ben Barnard"  
date: !r Sys.Date()  
output: pdf_document  
---
```


Running R code in the yaml

Rendering R Markdown documents

Knit

Render function

Engines

Python

Shell Scripts

SQL

Rcpp

Stan

JavaScript

CSS

Julia

C

Fortran

Introduction to R packages - 35 mins

What is an R package

Why should I build R packages

Distribution

Testing

Reproducibility

Documentation

How to build your first R package

GitHub

usethis

Ways to distribute R packages

Source and binary

GitHub

drat

CRAN

Introduction to LaTeX - 60 mins

What is LaTeX

Why should I use LaTeX

Reproducibility

Automation

Distribution

Not a Microsoft product

How to start your first LaTeX document

Bringing It all together - 30 mins

Possible extensions - 35 mins