

# Creating Dynamic Documents with RMarkdown and Knitr

*Marian L. Schmidt*

*May 11th, 2016*

## Contents

Information that is Relevant to this Tutorial . . . . .	1
<b>Dynamic Documents</b>	<b>1</b>
<b>Markdown</b>	<b>2</b>
<b>RMarkdown</b>	<b>2</b>
Why R Markdown? . . . . .	4
<b>Getting Started</b>	<b>8</b>
YAML Headers . . . . .	8
<b>Knitr Themes</b>	<b>9</b>

---

## Information that is Relevant to this Tutorial

This tutorial was constructed as a part of Dr. C Titus Brown's Data Intensive Biology (DIB) training program at the University of California, Davis. The DIB training program is hosting several local+remote workshops.

Today, there is are two remote classrooms tuning in:

1. University of California, Davis
2. Simon Fraser University in British Columbia, Canada.

In addition, the audio of this tutorial and the screen of the main instructor will be shared on YouTube.

The Github repository for this website can be found [here](#).

---

## Dynamic Documents

Literate programming is the basic idea behind dynamic documents and was proposed by Donald Knuth in 1984. Originally, it was for mixing the source code and documentation of software development together. Then we could:

1. **Tangle**: Extract the source code out of the document.
2. **Weave**: Execute the code to get the compiled results.

In dynamic documents, program or analysis code is run to produce output (e.g. tables, plots, models, etc) and then are explained through narrative writing.

---

## Markdown

To fully understand RMarkdown, we first need to cover Markdown, which is a system for writing simple, readable text that is easily converted to html. Markdown essentially is two things:

1. A plain text formatting syntax
2. A software tool written in Perl.
  - Converts the plain text formatting into HTML.

### Main goal of Markdown:

Make the syntax of the raw (pre-html) document as readable possible.

Would you rather read this?

```
<body>
  <section>
    <h1>Rock Climbing Packing List</h1>
    <ul>
      <li>Climbing Shoes</li>
      <li>Harness</li>
      <li>Backpack</li>
      <li>Rope</li>
      <li>Belay</li>
    </ul>
  </section>
</body>
```

The above code is html.

Or this?

```
# Rock Climbing Packing List

* Climbing Shoes
* Harness
* Backpack
* Rope
* Belay
```

The above code is Markdown and it is clear that this option is definitely much easier to read!

We will talk more about the syntax of Markdown after we introduce RMarkdown.

---

## RMarkdown

RMarkdown is a variant of Markdown that makes it easy to create dynamic documents, presentations and reports from R. It has embedded R code chunks to be used with **knitr** to make it easy to create reproducible (web-based) reports in the sense that they can be automatically regenerated when the underlying code is modified.

- RMarkdown lets you combine **Markdown** with images, links, tables, LaTeX, and actual R code.

- **RStudio makes creating documents from RMarkdown easy** but you can use Pandoc (more on that later) instead.
- RStudio (like R) is free and runs on any operating system.

RMarkdown renders many different types of files including:

- HTML
- PDF
- Markdown
- Microsoft Word
- Presentations:
  - Fancy HTML5 presentations:
    - \* ioslides
    - \* Slidy
    - \* Slidify
  - PDF Presentations:
    - \* Beamer
  - Handouts:
    - \* Tufte Handouts
- HTML R Package Vignettes
- Even Entire Websites!



Figure 1:

While there are a lot of different types of rendered documents in RMarkdown, today we will focus primarily on HTML output files, as I have found these files to be the most useful and flexible for my research.

## Why R Markdown?

A convenient tool for reproducible and dynamic reports with R!

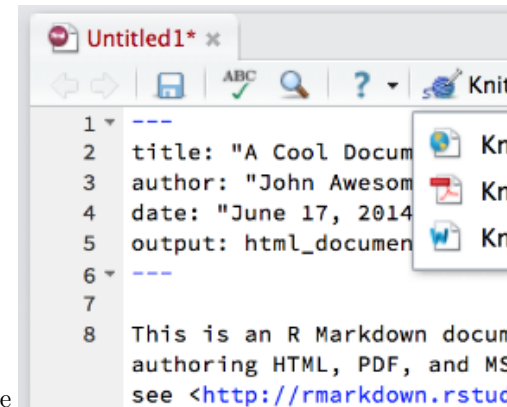
- Execute R code in a few ways:
  1. **Code chunks**
  2. **Inline code**
- Easy to:
  - Embed images.
  - Learn Markdown syntax.
  - Include LaTeX equations.
  - Include interactive tables.
  - Use version control with **Git**.
    - \* Even easier to share and collaborate on analyses, projects and publications!
  - Add external links - Rmarkdown even understands html code!
  - Make beautifully formatted documents.
- Do not need to worry about page breaks or figure placement.
- Consolidate your code and write up into a single file:
  - Slideshows, pdfs, html documents, word files

## Simple Workflow

1. Create `.Rmd` file that includes R code chunks and and markdown narratives.
2. Give the `.Rmd` file to `knitr` to execute the R code chunks and create a new `.md` file.
3. Give the `.md` file to pandoc, which will create the final rendered document (e.g. html, microsoft word, pdf, etc.).



Figure 2:



While this may seem complicated, we can hit the “Knit” button at the top of the page or we can run the following code:

```
rmarkdown::render("RMarkdown_Lesson.Rmd", "html_document")
```

### Creating an .Rmd File

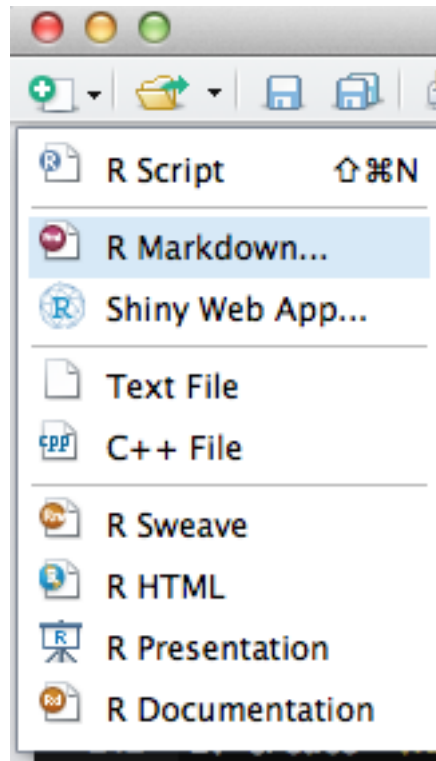


Figure 3:

### Installation

#### Load Packages

The following packages are necessary for today’s workshop.

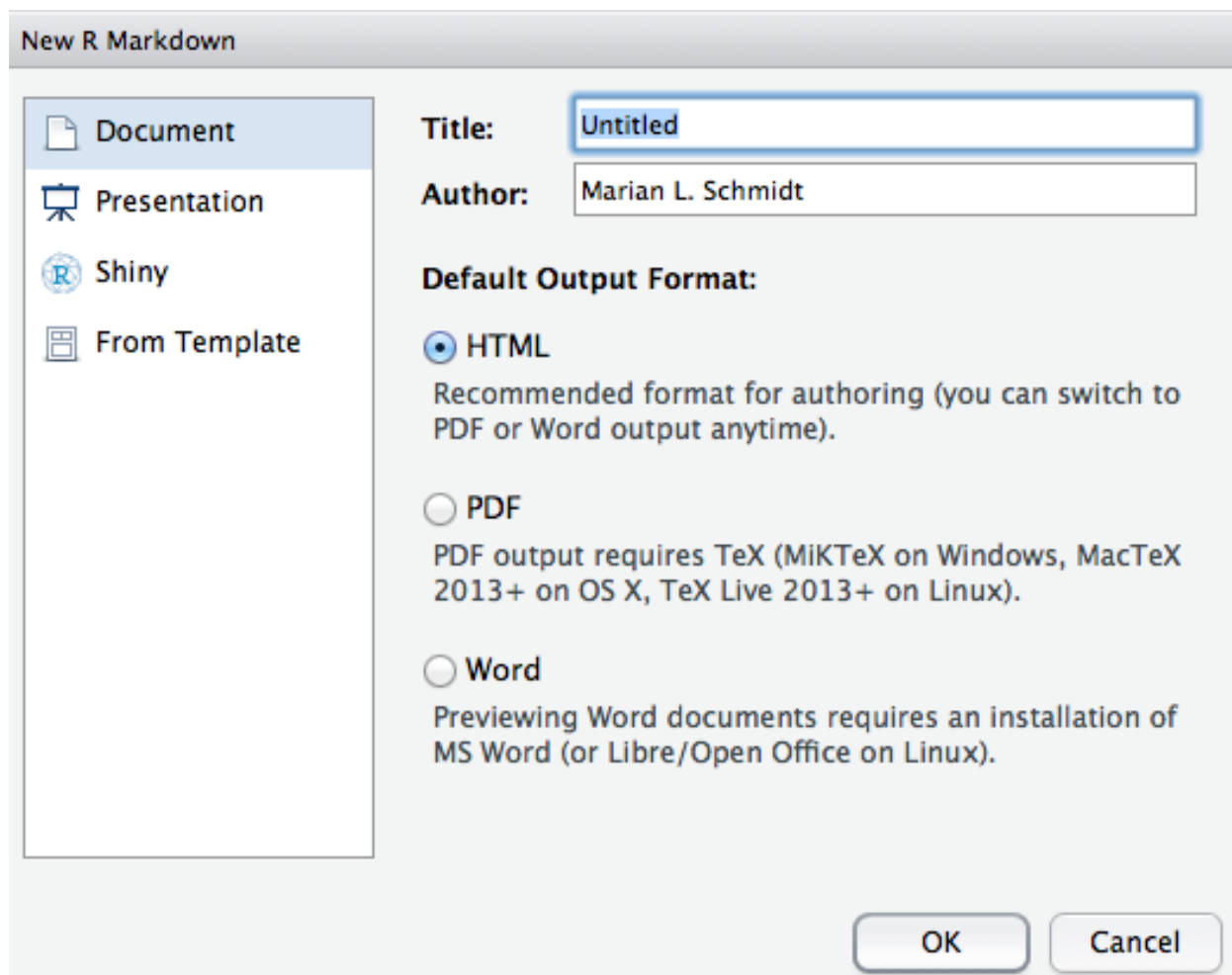


Figure 4:

```
#install.packages("knitr")
library(knitr)
#install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
#install.packages("ggplot2")
library(ggplot2)
```

```
knitr::opts_chunk$set(echo = TRUE, eval = TRUE)
```

## Markdown Basics

Check out the RMarkdown Reference Guide

**Something to remember:** End a line with two spaces to start a new paragraph!

Time to think!

What happens when you run the following code?

```
*honey* is very _sweet_. Yum^!!!^
bold and __bolder__
R2 values are the informative!
R2 describe the variance explained in the model. W
~~today is not my day~~
[link](www.rstudio.com)
# RMarkdown is the best
## R is the best
### Knitr is the best
#### Pandoc is the best
##### You're the best
 $\sqrt{b^2 - 4ac}$ 
 $X_{i,j}$ 
```

**Fun Fact!** The table of contents of this website was created from headers with 1-3 pound symbols!

## R Code Chunks

### Inline R Code

### Embedding Equations with LaTeX

Thank you to

James H. Steiger for this awesome website.

What's the difference between these two lines of code?

```
$$X_{i,j}$$
$X_{i,j}$
```

## Rendering Output

## Output Options

# Getting Started

## YAML Headers

### RMarkdown Appearance and Syle

Rmarkdown has several options that control the appearance of HTML documents. Some aruments to choose from are:

- **theme**
- **highlight**
- **smart**

The HTML output themes are drawn from the Bootswatch library. Valid **HTML themes** include the following:

- cerulean, cosmo,flatly, journal, lumen, paper, readable, sandstone, spacelab, simplex, united, and yeti.
  - For example, the theme of this page is **lumen**.
- Pass null for no theme (in this case you can use the `css` parameter to add your own styles).

**Highlight** specifies the syntax highlighting style. Supported styles include the following:

- default, espresso, haddock, kate, monochrome, pygments, tango, textmate, and zenburn.
- Pass null to prevent syntax highlighting.

**Smart** indicates whether to produce typographically correct output, converting straight quotes to curly quotes, — to em-dashes, – to en-dashes, and ... to ellipses. **Smart** is enabled by default.

For example:

```
---
output:
  html_document:
    theme: slate
    highlight: tango
---
```

If you felt inclined, you could also produce and use your own theme. If you did so, the output section of your YAML header would like like:

```
output:
  html_document:
    css: styles.css
```

If you wanted to go the extra mile and write your own theme in addition to highlight, the YAML header would look like:



```

---
output:
  html_document:
    theme: null
    highlight: null
    css: styles.css
---

```

Here's a link to learn more about the Appearance and Style in HTML output.

## Knitr Themes

The knitr syntax theme can be adjusted or completely customized. If you do not prefer the default themes, use the object `knit_theme` to change it. There are **80 themes** contained within `knitr` and we can view the names of them by `knit_theme$get()`.

What are the first 30 knitr themes?

```
head(knit_theme$get(), 30)
```

```
## [1] "acid"          "aiseered"      "andes"         "anotherdark"
## [5] "autumn"        "baycomb"       "bclear"        "biogoo"
## [9] "bipolar"       "blacknblue"    "bluegreen"     "breeze"
## [13] "bright"        "camo"          "candy"         "clarity"
## [17] "dante"         "darkblue"      "darkbone"      "darkness"
## [21] "darkslategray" "darkspectrum"  "default"       "denim"
## [25] "dusk"          "earendel"      "easter"        "edit-anjuta"
## [29] "edit-eclipse"  "edit-emacs"
```

We can use `knit_theme$set()` to set the theme. For example, to set the theme to *fruit* we could run the following code:

```
knit_theme$set("fruit")
```

Here's the link to find your favorite theme of all the 80 knitr highlight themes.