

MY THESIS

by

YOUR R. NAME

A thesis submitted to the
Department of Excellence
in conformity with the requirements for
the degree of Winner of Science

Queen's University
Kingston, Ontario, Canada

May 20xx

Copyright © Your R. Name, 20xx

Abstract

This is a placeholder for including an abstract.

Second paragraph of abstract starts here.

Acknowledgements

This package was possible thanks to the work of Chester Ismay in developing **thesisdown** and Yihui Xie and other contributors in developing **bookdown**. Further acknowledgements for the development of the LaTeX template can be found in the `template.tex` and `queensuthesis.cls` files.

Table of Contents

Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vi
Introduction	1
Welcome	1
Motivation	2
Structure	3
Chapter 1: Getting started	6
1.1 Writing in markdown	6
1.2 Figures and tables in R markdown	7
1.3 Math equations and formulas	9
1.4 Assembling your thesis	10
1.4.1 YAML Ain't Markup Language	10
1.4.2 Folders and files	12
1.5 Summary	14

Chapter 2: Workflow	15
2.1 Data files and scripts	15
2.1.1 <code>targets</code> package	16
2.2 Version control	16
2.3 Troubleshooting	17
2.3.1 R errors	17
2.3.2 Markdown errors	17
2.3.3 \LaTeX errors	17
2.3.4 Where to find help	18
2.4 Writing environment (the fun part)	18
2.5 Getting feedback for your thesis	19
Chapter 3: Additional functions	20
3.1 <code>kableExtra</code>	20
3.2 Writing and editing TeX	20
3.3 Formatting figure captions	21
Conclusion	22
3.4 More info	22
References	23
Appendix A: Further resources	24
Appendix B: The second appendix	25

List of Figures

1 Illustration of document generation in gaeltdown, with inputs in red/orange rectangles, software in yellow ellipses, intermediate products in purple hexagons, and outputs a green rectangles. 5

Introduction

Welcome

Welcome to the **gaelsdown** guide! This document is intended to get you up and running with writing your thesis in R markdown so that it can be published in a variety of formats, most importantly a PDF with formatting specifications in the TeX language.

In this **introduction chapter**, I'll run through the benefits of adopting **gaelsdown** into your writing process and what to expect, then explain the basics of how you get from a R markdown to a formatted pdf. Then, in the **first chapter** of the guide, I'll provide you with all the basic tools to put together a document with formatting, figures and tables, and a structure appropriate to your thesis. In the **second chapter**, I'll get into some of the more advanced methods for figures and tables, specifying formatting inline, and more. Finally, in the **third chapter**, I'll go over some common issues and how you can address them, as well as providing more resources and approaches for troubleshooting.

Motivation

The purpose of **gaelsdown** is to enable you to write your thesis in R markdown so that you can focus on the content of your thesis. With R markdown, you can insert code for figures, tables, and references at ease, then transform the markdown file into a finished, formatted document that meets the specifications provided by the School of Graduate Studies. This is a departure from what may be a more familiar approach of using R to generate tables and figures separately, then inserting those into a “What-You-See-Is-What-You-Get” (WYSIWYG) editor like Microsoft Word.

There’s a number of **advantages** to this approach over using a WYSIWYG editor, including:

- Your R code and writing are streamlined into one document, so that changes to figures and tables are automatically integrated whenever you knit your document, and the bibliography and table of contents are automatically generated.
- R markdown is a much more lightweight format to work in, which reduces the load on your computer significantly if your thesis has hundreds of pages and includes many figures.
- The files making up your thesis are in plain text formats that can be easily transferred between platforms and require only free, open source software.
- By using LaTeX to generate your document, you have access to a wide range of packages to support different types of mathematical and scientific formulas and a professional-quality typesetting engine, superior to Microsoft Word.
- Many journals make .tex templates available for authors, which can help with submission. There are also several options for converting from a LaTeX document to a Word-compatible format.

- If you're writing a thesis for Queen's University, the formatting of the PDF output already conforms to the specifications that the school has published. If you're writing for another school, or if you'd like to change some of the formatting, changes can be made in one place that apply to the whole document.

However, there are some potential challenges with writing your thesis in R markdown, especially if this is a new approach for you. This guide will help with avoiding or anticipating some of these challenges, but be aware of the following:

- Introducing formatting changes into a LaTeX document is a different process than in a WYSIWYG editor, and involves a bit of a learning curve. However, by using a template such as **gaelsdown**, you can avoid much of the LaTeX editing and focus on the content of your thesis.
- **gaelsdown** relies on a few different programs when it knits together your document, which means there's a few different steps at which an error can be introduced. I'll explain what these steps are, and tips for addressing any issues encountered at each step.

Structure

To understand how **gaelsdown** and similar template packages work, you need to understand the different software involved:

- **knitr** converts your R markdown file (.Rmd) into a markdown file (.md). The difference between a R markdown file is that it includes R code chunks that need to be rendered as images for them to be markdown compatible.

- **pandoc** is included with your RStudio install and is capable of converting between a large number of different text formats, including markdown, Word (.docx), web browser compatible (.html), and LaTeX (.tex).
- \LaTeX renders a typeset format such as PDF from .tex files—if you knit into another format such as gitbook or Word, LaTeX will not have been involved.

Here's a visualization of how it all flows together: If you knitted this document locally from the **gaelsdown** package and watched the console carefully, then you've already seen all these steps in action! You can run the knit again if you want to review it.

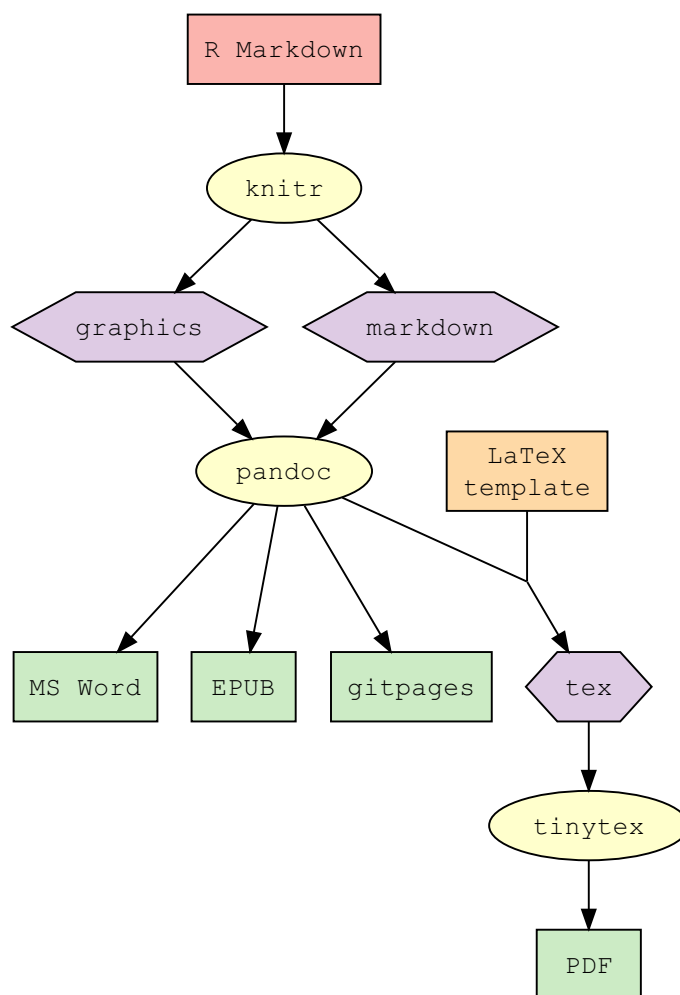


Figure 1. Illustration of document generation in gaelstown, with inputs in red/orange rectangles, software in yellow ellipses, intermediate products in purple hexagons, and outputs a green rectangles.

Chapter 1

Getting started

If you've reached this step, you should have already installed the **gaelsdown** and **tinytex** packages and knit your first document (this one!). If not, first take a look at the README for instructions explaining these steps.

1.1 Writing in markdown

In **RStudio**, you write in **Pandoc Markdown** to generate publishable documents. That means this is a valuable skill even if you're not using a LaTeX template, and it's also an easy skill to pick up. The basic principle behind markdown is that it should be easy to read even in plain text, and it accomplishes this by using special characters in intuitive ways.

For example, to *italicize* (emphasis) a word or line of text, you place it inside a pair of underscores or asterisks like so: `_italicize_`. To make a line **boldface** (strong emphasis), you put with two underscores or asterisks: `__boldface__`.

Where you start to get into real power of markdown is the ability to structure a document, such as by using headers. I mark the beginning of this chapter with a level-one header, `# Getting started`, and the beginning of this section with a level-two header, `## Writing in markdown`. If, say, we wanted to jump to another chapter, I could include a link using the header name `[Introduction]: Introduction`.

For a full primer on the features available in Pandoc Markdown, including the ability to insert lists, blockquotes, code chunks, and math, read this tutorial.

1.2 Figures and tables in R markdown

If you looked through the Pandoc Markdown tutorial, you may have noticed that there are features to include tables and images. However, the beauty of R markdown is that you can insert tables and figures using R code, then allow **knitr** to convert this code into a markdown format. To do this, I'll first need to load some R packages:

```
# List of packages required for this analysis
pkg <- c("dplyr", "ggplot2", "knitr", "bookdown", "devtools",
        "DiagrammeR")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
if (length(new.pkg))
  install.packages(new.pkg, repos = "http://cran.rstudio.com")
# Load packages (This can be accomplished by just loading gaelsdown)
library(gaelsdown)
```

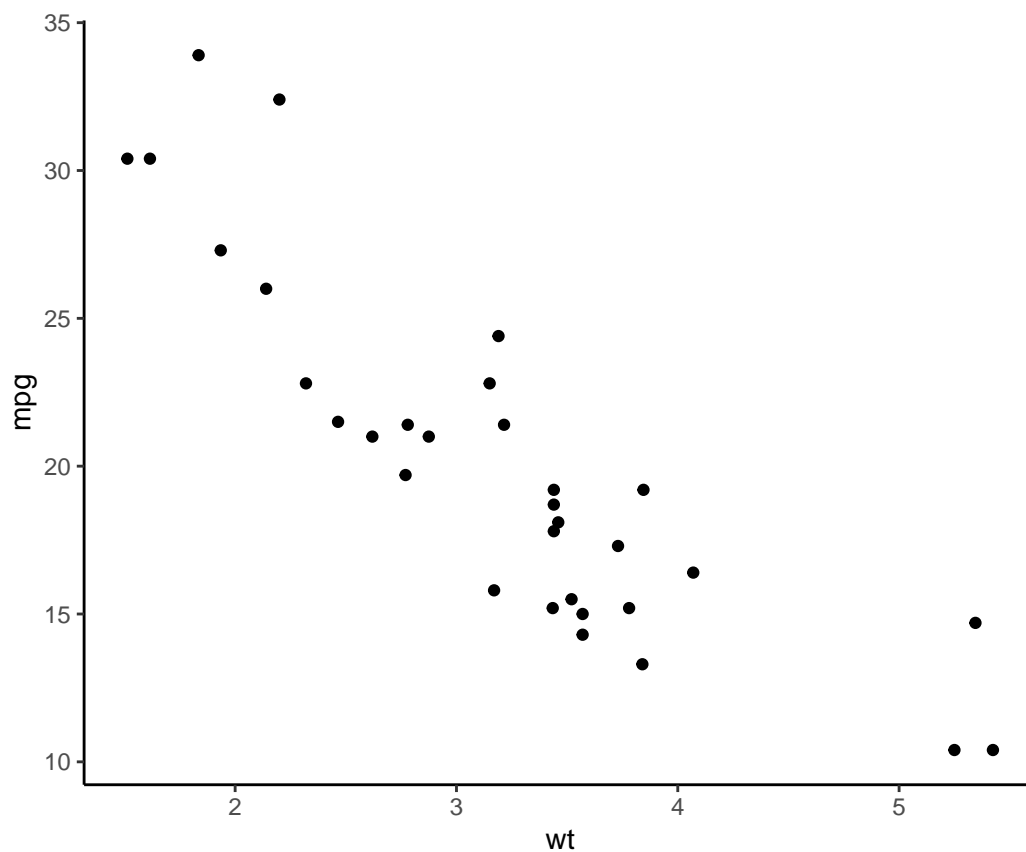
Once the packages are loaded, I can assemble a basic table with R like so:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

```
mtcars %>% head() %>% kable() %>% kable_styling(latex_options =  
  "scale_down")
```

Alternatively, I may want to produce a plot:

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +  
  geom_point() +  
  theme_classic()
```



I'll have some tips on how to make your tables and figures look great in the next chapter on Additional functions.

1.3 Math equations and formulas

One of the unique areas of strength of LaTeX is its support for writing equations and other scientific expressions, such as chemical formulas. For example, if you wanted to show the equation for the rate of population change, you would enter the following:

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

Which produces:

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

Equations can also be numbered and referred to, like figures or tables, by using the LaTeX command `\begin{equation}`:

```
\begin{equation}
S(t) = \frac{S_{\max}}{1 + e^{-(t-T_i) * r}} \label{eq:logistic}
\end{equation}
```

$$S(t) = \frac{S_{max}}{1 + e^{-(t-T_i)*r}} \quad (1.1)$$

1.4 Assembling your thesis

Although the final product you obtain using **gaelsdown** will likely be a single document contained in one file, there are multiple files that go into creating this document. There's a simplicity to this in that each individual file has a specific function within the larger document, and I'll explain how you can customize the structure of your thesis by adding or removing certain files or sections.

1.4.1 YAML Ain't Markup Language

Every thesis generated using this template has an **index.Rmd** file. This is the file you use to knit the document, and it contains your **YAML** section (recursively named, "YAML Ain't Markup Language"). Information from the **YAML** is passed onto **Pandoc**, and may include specifications for the output format and other options. You can also include *frontmatter* material here, such as your abstract.

Open up **index.Rmd** in RStudio to see what the YAML looks like. It's the section at the beginning bracketed by three dashes:


```
---  
---
```

Note: You need to be particularly attentive to the spaces in the YAML, so make sure when uncommenting lines you only remove the # and not the spaces!

At the top of the YAML are some pieces of information that get passed along to the title page, such as the title and author name. Since the format for title pages are set for Queen's University, you'll want to change the answers to these entries, but not which lines are included (To change the format of the title page, you'd have to go into the .tex template).

Following this you'll find your output options. Here, you have the option of either PDF, gitbook (web page), Word, and EPUB; to change the format, just comment out the active line and uncomment the appropriate line. After this are lines for your introductory material: abstract, acknowledgements, glossary, etc. These sections are *frontmatter*, will appear before the Table of Contents, are numbered with Roman instead of Arabic numerals, and won't be labelled as chapters.

Frontmatter is a bit quirky in that it needs to be read directly into LaTeX, so the frontmatter sections have to be defined in the .tex template and the YAML. I've defined all of the required and optional frontmatter sections specified for Queen's University theses, with some of them commented out. If you want to include frontmatter in another output format such as Word, then it has to be defined in the body of the .Rmd file, not the YAML.

After this are instructions for your references section. There's some defaults here for this document but you'll want to specify your own **bib** and **csl** file. The process for using these are the same as in **bookdown**, so for instructions on how to specify the **bibliography** and **csl** sections, follow Yihui Xie's tutorial [here](#).

1.4.2 Folders and files

Upon creating your *gaelsdown* project in RStudio, your folder will have the following files and folders:

bib/

The `.bib` file lists the references that you use in your thesis, and the name of your `.bib` should be specified in the YAML.

csl/

The `.csl` file defines the formatting for references in your thesis, and `.csl` files matching the format for specific journals can be found online. The file name should also be specified in the YAML.

data/ and figure/

These are just folders for example data and figures; you can set the folders as you wish for other files (e.g. a `R/` folder for scripts).

_bookdown.yml

Configuration file for **bookdown**, the package underlying **gaelsdown**. You can change the output file and chapter prefixes here; see the bookdown reference for more information.

01-chap1.Rmd, 02-chap2.Rmd...

These are the main chapters of your thesis, which **gaelsdown** will join into one document. The .Rmd files in your main directory will all be knit together, so you can add as many files as you need and it's not necessary to follow this naming format (it just makes things easier to read).

98-references.Rmd

Your references come after every section apart from the appendices.

99-appendix.Rmd

The appendices come after the references. You can add as many appendix sections as necessary.

prelims/

Files for your abstract, acknowledgements, and other files can go in here so that they are not automatically knit like chapters but instead inserted ahead of the table of contents according to YAML instructions. I've written in if statements to check whether the output is LaTeX or not to decide how to insert these files.

1.5 Summary

In this section, I've covered markdown basics including text formatting, headers, and links. I've introduced how to add figures and tables to your thesis, how YAML is used to configure your output, and the different contents of your *gaelsdown* project.

Chapter 2

Workflow

In the previous chapter, I introduced the necessary basics for using **gaelsdown** to write your thesis: markdown language, YAML, and the document structure. In this chapter, I'll describe some recommended practices to integrate **gaelsdown** neatly into your thesis writing and some basic troubleshooting advice.

2.1 Data files and scripts

While your R markdown files will contain the text, tables, and figures for your thesis, you will likely have a workflow set up to process your raw data and generate your analyses. I recommend keeping this workflow separate as R scripts (.R file extension), generating data that will be used by the R markdown scripts read by **gaelsdown**. You can then tell your R scripts to produce data in the form of .csv or .RData, then read in these files in the R markdown along with code to generate figures and tables.

2.1.1 `targets` package

An approach you may want to consider adopting is using `targets` (formerly `drake`) to organize your R workflow. This package works well with piped commands from `magrittr` used in tidyverse.

The advantage of using `targets` is that you can modify sections of your pipeline, and the function will automatically identify which computation needs to be run and which does not. Objects generated inside a `targets` function are written into a cache, and this works well with `gaelsdown` because those cached objects can be read into your R markdown files. You can do your data processing analysis using `.R` scripts, and only need to load that data and include the code in the R markdown files for generating your figures and tables.

You can find an interactive tutorial on how to use `targets` [here](#).

2.2 Version control

With your `gaelsdown` thesis set up as a project, it's easy to integrate git into your workflow through RStudio. This will help you to identify errors when they're introduced (see troubleshooting) and can be used for collaboration. The easiest way is through GitHub.

With a `gaelsdown` project, you may not want to include all of the automatically-generated files in your git. Instead, you should set these files to be ignored in the `.gitignore` (more about setting `gitignore`):

- `_book/`
- `_bookdown_files/`

2.3 Troubleshooting

Perhaps the biggest challenge of working with **gaelsdown** comes from the fact that involves different software working together (see Figure 1). This means that errors can be introduced at different stages of your document assembly.

Thankfully, it's relatively easy to determine where the error is introduced, which helps a lot with resolving these errors. When you knit the `index.Rmd`, the rendering will follow a set sequence, so the source of the error will be apparent in the Render panel in RStudio (see Figure 1). You may need to check that your installations for requisite packages are up to date (such as `rmarkdown`, `tinytex`).

2.3.1 R errors

These are errors that prevent your R code from running, such as syntax errors missing functions.

2.3.2 Markdown errors

Markdown is not especially sensitive to errors, so faults are more likely to appear in your document.

2.3.3 \LaTeX errors

LaTeX errors may be introduced if you have modified the underlying `.tex` or `.cls` files, if you have inserted LaTeX commands directly into your markdown, or if you use functions that pass instructions onto LaTeX. If you encounter errors at this stage, a `.tex` file will have been generated which will help with identifying the source of the error.

2.3.4 Where to find help

There's a couple places you can look for help with **gaelsdown**. If you have identified a particular error, you can try to enter the error message or keywords into a web search and see if there are any entries. Some of these may point you to stackoverflow, a useful resource for tapping the expertise that is out there.

Additionally, you can seek help directly from the creators of this package (gaelsdown) and underlying packages (bookdown, thesisdown). Visit the github pages for these packages and take a look through existing issues. If you cannot find the issue described, submit an issue of your own!

2.4 Writing environment (the fun part)

Although being able to write your entire thesis using R markdown means you could write your whole thesis in RStudio, that doesn't mean you have to! However, a traditional document editor like Microsoft Word is not suited to editing markdown files (although there are now plugins available that you can try) since it will either leave your document unformatted or insert its own formatting code into your files. Instead, you can use a markdown editor to open your .Rmd files, which will recognize the formatting and provide a visually pleasing and focused environment for writing. These programs are available for free or a low cost, and I used Typora for my own thesis.

2.5 Getting feedback for your thesis

Because much of the formatting for the **gaelsdown** output relies on the TeX engine, HTML and Microsoft Word outputs will not reflect the finished product. However, Word documents are useful for tracking changes and if you are getting feedback from your supervisor on your thesis this way, you will have to manually enter the changes into your underlying .Rmd files. Additionally, knitting directly to a .doc file often results in issues with tables and figures. If you do need a .doc output, you may want to try knitting to a gitbook (HTML), then opening and saving the respective HTML files as word files.

Chapter 3

Additional functions

This section is still in the works, and will describe some additional functions you may want to use when writing your thesis.

3.1 `kableExtra`

`kableExtra` adds functions to further customize tables that `knitr`'s `kable()` function generates in R markdown. Some useful functions include being able to add footnotes (for example, to specific significance thresholds) or set formatting for individual columns or rows in a table.

3.2 Writing and editing TeX

You can include expressions that would be rendered by TeX in your R markdown document; just be aware that they will not be rendered if knitting into other formats such as a gitbook or Word document.

Manual formatting commands such as `\newpage`, `break`, or `\textbf` can be inserted in this way. However, if you find yourself needing to manually adjust the position of page or line breaks, something may need to be reconfigured in the underlying TeX template!

3.3 Formatting figure captions

Instead of writing the full figure caption in the R code chunk, you can enter your caption in the body of the R markdown document following the expression `(ref:caption-name)`, then substitute the caption with `(ref:caption-name)`. By doing so, you can include code in the caption that you would otherwise not be able to include, such as inline code or links to other sections of the document.

Conclusion

This conclusion is a placeholder. If we don't want Conclusion to have a chapter number next to it, we can add the `{-}` attribute.

3.4 More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

References

- Angel, E. (2000). *Interactive computer graphics : A top-down approach with OpenGL*. Boston, MA: Addison Wesley Longman.
- Angel, E. (2001a). *Batch-file computer graphics : A bottom-up approach with Quick-Time*. Boston, MA: Wesley Addison Longman.
- Angel, E. (2001b). *Test second book by angel*. Boston, MA: Wesley Addison Longman.

Appendix A

Further resources

For information on using R markdown with LaTeX, Yihui Xie's bookdown guide is fairly comprehensive.

If you want to integrate more TeX commands into your thesis, the Comprehensive TeX Archive Network is where you will find documentation on all the TeX packages available.

Chester Ismay's thesisdown included a guide (I've written this present guide in place of that one) that you can find here. You may also find Rosanna van Hespén's series of blogposts on writing a thesis in R markdown useful, although Rosanna used limited LaTeX code and wrote it directly into the markdown files.

Appendix B

The second appendix

This second appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readability and/or setup.

```
# By default tinytex only compiles once; compiling multiple  
# times is necessary to ensure the List of Figures and List of  
Tables  
# page numbers are correct if listed in the Table of Contents  
options(tinytex.compile.min.times = 3)  
  
# This chunk ensures that the gaelsdown package is  
# installed and loaded. This gaelsdown package includes  
# the template files for the thesis.  
if(!require(devtools))  
  install.packages("devtools", repos = "http://cran.rstudio.com")  
if(!require(gaelsdown))  
  devtools::install_github("eugenesit/gaelsdown")  
library(gaelsdown)  
if(!require(DiagrammeR))  
  install.packages("DiagrammeR", repos = "http://cran.rstudio.com")  
library(DiagrammeR)  
if(!require(webshot))
```

```
install.packages("webshot", repos = "http://cran.rstudio.com")
library(webshot)
if(!webshot::is_phantomjs_installed())
  webshot::install_phantomjs()
if(!require(kableExtra))
  install.packages("kableExtra", repos = "http://cran.rstudio.com")
library(kableExtra)
```

```
DiagrammeR::grViz("
  digraph{
  graph []
  node [shape = rectangle, fontname = 'Courier New',
fontface = bold, style = filled]
    rmd [label = 'R Markdown', fillcolor = '#fbb4ae']
    knitr [label = 'knitr', shape = ellipse, fillcolor
= '#ffffcc']
    pandoc [label = 'pandoc', shape = ellipse,
fillcolor = '#ffffcc']
    word [label = 'MS Word', fillcolor = '#ccebc5']
    epub [label = 'EPUB', fillcolor = '#ccebc5']
    html [label = 'gitpages', fillcolor = '#ccebc5']
    blank [shape = point, width = 0, height = 0]
    img [label = 'graphics', shape = polygon, sides =
6, fillcolor = '#decbe4']
    md [label = 'markdown', shape = polygon, sides =
6, fillcolor = '#decbe4']
    tex [label = 'tex', shape = polygon, sides = 6,
fillcolor = '#decbe4']
    engine [label = 'tinytex', shape = ellipse,
fillcolor = '#ffffcc']
    pdf [label = 'PDF', fillcolor = '#ccebc5']
    style [label = 'LaTeX\ntemplate', fillcolor =
'#fed9a6']
  edge []
    rmd -> knitr
    knitr -> md
    knitr -> img
    md -> pandoc
```



```
img -> pandoc
pandoc -> blank [arrowhead = none]
pandoc -> word
pandoc -> epub
pandoc -> html
tex -> engine
style -> blank [arrowhead = none]
blank -> tex
engine -> pdf
{ rank = same; word, epub, html, tex }
}" , width = 500)
```