Writing a reproducible paper in Pagedown[3]

Paul C. Bauer
Mannheim Centre for European Social Research

Camille Landesvatter
Mannheim Centre for European Social Research

First version: 20 June, 2021
This version: 21 Juni, 2021

Abstract

The present paper provides a template for a reproducible scientific paper written in R Markdown. Below I outline some of the "tricks"/code (e.g., referencing tables, sections etc.) I had to figure out to produce this document. The underlying files which produce this document can be downloaded here[1]. I think I got pretty far but there is always room for improvement and more automatization, in parallel to the incredible developments in R and Rstudio (bookdown etc.). I intend to update this file when I discover more convenient code (you can follow any updates through the corresponding github repo[2]).

Keywords: open science, transparency, replication, R, markdown, pagedown.

[3]Corresponding adress: mail@paulcbauer.eu
[1]https://drive.google.com/drive/folders/1zJP3cNPrHN-gj0rcmbHQgg-XA0hqDXdd?usp=sharing
[2]https://github.com/paulcbauer/Writing_a_reproducable_paper_in_rmarkdown/

1. Why reproducible research (in R)?

Some arguments...

- **Access**: Research is normally funded by taxpayers (researchers are also taxpayers). Hence, it should be freely accessible to everyone without any barriers, e.g., without requiring commercial software. Importantly, researchers from developing countries are even more dependent on free access to knowledge (**Kirsop2005-ro?**).
- **Reproducability**: Even if you have written a study and analyzed the data yourself you will forget what you did after a few months. A fully reproducible setup will help you to trace back your own steps. Obviously, the same is true for other researchers who may want to understand your work and built on it. It may sound like a joke but why not aim for a document that can be used to reproduce your findings in 500 years.
- **Errors**: Manual steps in data analysis (e.g., manually copy/pasting values into a table etc.) may introduce errors. R Markdown allows you to **automatize** such steps and/or avoid them.
- **Revisions**: Revising a paper takes much less time if you have all the code you need in one place, i.e., one `.rmd` file. For instance, if you decide to exclude a subset of your data you simply need to insert one line of your code at the beginning and everything is rebuilt/re-estimated automatically.

2. Why Pagedown?

3. Prerequesites

We assume that you are using R on a day-to-day basis and you may have even started to work in R Markdown. If you don't know what R Markdown is watch this short video[4]. Also, there is an overview and template for a scientific paper written in R markdown by one of us two (Paul), which follows the same idea and structure just as this review (github repo[5]). The template compiles information, operations, tips and tricks you will very likely need to create a well-formatted scientific paper with R markdown.

Based on R Markdown, Pagedown allows you to create custom and well-formatted (paged) HTML Documents. For a comprehensive overview watch this video[6] which is a record of a talk introducing R's pagedown given by Yihui Xie (who in addition to Romain Lesur developed R's pagedown package). If you are not in a video watching mood find the slides here[7].

Then...

- ...install R[8] and Rstudio[9] (most recent versions) (**R2017?**; **Rstudio2015?**).
- ...install the "pagedown"-package from github using the code below.

```
R> remotes::install_github('rstudio/pagedown')
```

- ...also install the packages below using the code below (**bookdown1?**; **bookdown2?**; **knitr1?**; **knitr2?**; **knitr3?**; **kableextra?**; **plotly?**).

```
R> install.packages(c("rmarkdown", "knitr", "kableExtra",
R+           "stargazer", "plotly", "knitr"))
```

- ...download the 4 input files we created — `paper.rmd`, `references.bib`, `data.csv` and `american-sociological-association.csl` — from this folder[10]. Ignore the other files.
- ...also download the 4 styling files we created: `wp_paged.html`, `wp.css`, `wp-fonts.css` and `wp-pages.css`.
- ...store all 8 files from above together in one folder (and use this folder as your working directory later on)
- ...learn R and read about the other underlying components namely Markdown[11], R Markdown[12] and Latex[13].

[4] https://vimeo.com/178485416
[5] https://github.com/paulcbauer/Writing_a_reproducable_paper_in_rmarkdown/
[6] https://www.rstudio.com/resources/rstudioconf-2019/pagedown-creating-beautiful-pdfs-with-r-markdown-and-css/
[7] https://slides.yihui.org/2019-rstudio-conf-pagedown.html#1
[8] https://www.r-project.org/
[9] https://www.rstudio.com/
[10] https://drive.google.com/drive/folders/1zJP3cNPrHN-gj0rcmbHQgg-XA0hqDXdd?usp=sharing
[11] https://en.wikipedia.org/wiki/Markdown
[12] https://rmarkdown.rstudio.com/lesson-1.html
[13] https://en.wikipedia.org/wiki/LaTeX

- . . .pagedown comes with several Rmd-templates (presentations, poster, thesis, etc.) and via this review we provide another template for a working paper style. If however you want to modify single aspects or create your own template, you will need at least some basic skills in CSS[14] and HTML[15].

[14] https://www.w3schools.com/css/
[15] https://www.w3schools.com/html/

4. Basics: Input and output files

All the files you need to produce the present PDF file are:

1. the input files:

- `paper.rmd` (the underlying R Markdown file).
- `references.bib` (the bibliography).
    - I use paperpile to manage my references and export the `.bib` file into the folder that contains my `.rmd` file.
- `data.csv` (some raw data).
- `american-sociological-association.csl` (defines the style of your bibliography).[16]

2. the "styling" files:

Basically, these are files you will need to specify in the YAML of your rmd-file, so that R and ultimately pagedown recognizes the certain style you want to achieve for your document. With using our templates, you will create a document that has the "look" of a working paper.

- `wp_paged.html`
- `wp.css`
- `wp-fonts.css`
- `wp-pages.css`

Take `paper.rmd` (the underlying R Markdown file of this pdf) and have a look at the YAML (line 18-22) to see how to specifiy these files. Basically, what happens here is that within the jss_paged function[17] we additionally specify that we want to use custom css and html.

Download these files[18] and save them into a folder. Close R/Rstudio and directly open `paper.rmd` with RStudio. Doing so assures that the working directory is set to the folder that contains `paper.rmd` and the other files.[19]

Once you run/compile the `paper.rmd` file in Rstudio it creates mainly one output files:

- `paper_pagedown.html` (the one you are reading right now)

By using pagedown's `chrome_print` function in the YAML (uncomment the line) your html based web page will be printed to be PDF (the PDF will be stored in your working directory)

- `paper_pagedown.pdf`

[16] You can download various citation style files from this webpage: https://github.com/citation-style-language/styles.
[17] https://rdrr.io/cran/pagedown/man/jss_paged.html
[18] https://drive.google.com/drive/folders/1zJP3cNPrHN-gj0rcmbHQgg-XA0hqDXdd?usp=sharing
[19] You can always check your working directory in R with `getwd()`.

5. Referencing within your document

To see how referencing works simply see the different examples for figures, tables and sections below. For instance in Section 8 you can find different ways of referencing tables. The code of the underlying `paper.rmd` will show you how I referenced Section 8 right here namely with '`Section \@ref(sec:tables)`.'

6. Software versioning

Software changes and gets updated, especially with an active developer community like that of R. Luckily you can always access old versions of R[20] and old version of R packages in the archive[21]. In the archive you need to choose a particular package, e.g dplyr and search for the right version, e.g., `dplyr_0.2.tar.gz`. Then insert the path in the following function: `install.packages("https://...../dplyr_0.2.tar.gz", repos=NULL, type="source")`. Ideally, however, results will be simply reproducible in the most current R and package versions.

I would recommend to use the command below and simply add it to the appendix as I did here in Appendix 15.1. This will make sure you always provide the package versions that you used in the last compilation of your paper. For more advanced tools see packrat[22].

```
R> cat(paste("#", capture.output(sessionInfo()), "\n", collapse =""))
R>  # or use message() instead of cat()
```

[20] https://cran.r-project.org/bin/windows/base/old/
[21] https://cran.r-project.org/src/contrib/Archive/
[22] https://rstudio.github.io/packrat/

7. Data

7.1. Import

Generally, code is evaluated by inserting regular R Markdown blocks.

```
R> x <- 1:10
R> x

 [1] 1 2 3 4 5 6 7 8 9 10
```

Below we import an exemplary dataset (download[23]).

```
R> data <- read.csv("data.csv")
R> head(data)

  X speed dist
1 1     4    2
2 2     4   10
3 3     7    4
4 4     7   22
5 5     8   16
6 6     9   10
```

8. Tables

Producing good tables and referencing these tables within a R Markdown PDF has been a hassle but got much better. Examples that you may use are shown below. The way you reference tables is slightly different, e.g., for `stargazer` the label is contained in the function, for `kable` it's contained in the chunk name.

8.1. stargazer(): Summary and regression tables

8.1.1. Summary table with stargazer

Table **??** shows summary stats of your data.[24] I normally use `stargazer()` (**hlavac2013stargazer?**) which offers extreme flexibility regarding table output (see `?stargazer`).

```
R> library(stargazer)
R> stargazer(cars,
R+       label="tab1",
R+       table.placement = "H",
R+       type="html",
R+       header=FALSE)
```

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
speed	50	15.400	5.288	4	12	19	25
dist	50	42.980	25.769	2	26	56	120

8.1.2. Regression table with stargazer

[23] https://drive.google.com/drive/folders/1zJP3cNPrHN-gj0rcmbHQgg-XA0hqDXdd?usp=sharing
[24] To reference the table where you set the identifier in the stargazer function you only need to use the actual label, i.e., ´tab1´.

Table ?? shows the output for a regression table. Make sure you name all your models and explicitly refer to model names (M1, M2 etc.) in the text.

```
R> library(stargazer)
R> model1 <- lm(speed ~ dist, data = cars)
R> model2 <- lm(speed ~ dist, data = cars)
R> model3 <- lm(dist ~ speed, data = cars)
R>
R> stargazer(model1, model2, model3,
R+      label="tab2",
R+      table.placement = "H",
R+      column.labels = c("M1", "M2", "M3"),
R+      type="html",
R+      model.numbers = FALSE,
R+      header=FALSE)
```

	Dependent variable:		
	speed		dist
	M1	M2	M3
dist	0.166***	0.166***	
	(0.017)	(0.017)	
speed			3.932***
			(0.416)
Constant	8.284***	8.284***	-17.579**
	(0.874)	(0.874)	(6.758)
Observations	50	50	50
R^2	0.651	0.651	0.651
Adjusted R^2	0.644	0.644	0.644
Residual Std. Error (df = 48)	3.156	3.156	15.380
F Statistic (df = 1; 48)	89.567***	89.567***	89.567***
Note:			*p<0.1;* ***p<0.05;*** p<0.01

8.2. kable() and kable_styling()

Another great function is `kable()` (`knitr` package) in combination with `kableExtra`. Table 8.1 provides an example.[25]

```
R> library(knitr)
R> library(kableExtra)
R>
R> kable(mtcars[1:10,], row.names = TRUE,
R+    caption = 'Table with kable() and kablestyling()',
R+    format = "html", booktabs = T) %>%
R+     kable_styling(full_width = T,
R+            latex_options = c("striped",
R+                      "scale_down",
R+                      "HOLD_position"),
R+            font_size = 10)
```

[25] To reference the table produced by the chunk you need to add ´tab:´ to the chunk name, i.e., ´tab:tab3´.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Table 8.1: Table with kable() and kablestyling()

8.3. modelsummary

https://vincentarelbundock.github.io/modelsummary/articles/datasummary.html

9. Inline code & results

10. Figures

10.1. R base graphs

Inserting figures can be slightly more complicated. Ideally, we would produce and insert them directly in the `.rmd` file. It's relatively simple to insert R base graphs as you can see in Figure 10.1.

```
R> plot(cars$speed, cars$dist)
```

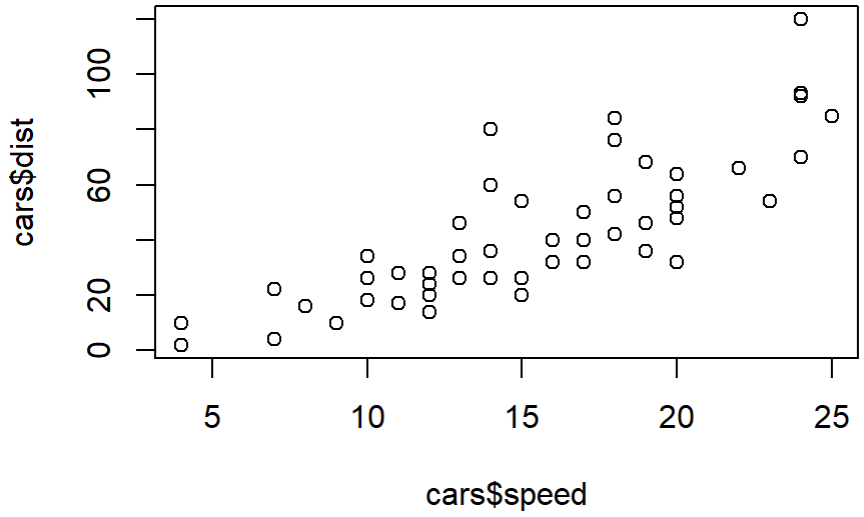

Figure 10.1: Scatterplot of Speed and Distance

But it turns out that it doesn't always work so well.

10.2. ggplot2 graphs

Same is true for ggplot2 as you can see in Figure 10.2.

```
R> mtcars$cyl <- as.factor(mtcars$cyl) # Convert cyl to factor
R> library(ggplot2)
R> ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl)) + geom_point() +
R+  labs(x="Weight (lb/1000)", y = "Miles/(US) gallon",
R+     shape="Number of \n Cylinders") + theme_classic()
```

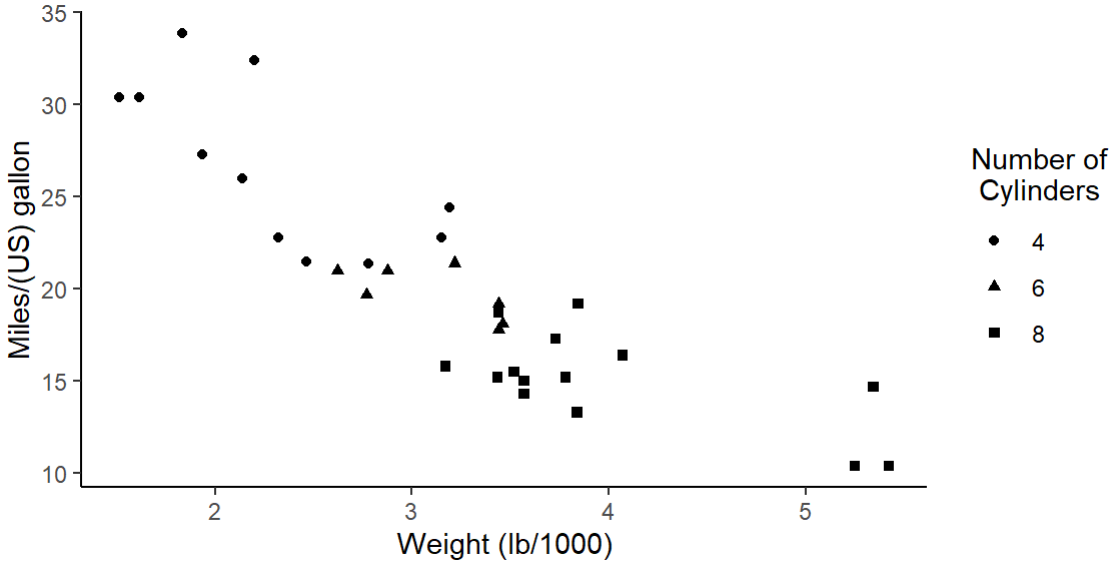

Figure 10.2: Miles per gallon according to the weight

10.3. Interactive graphs

html!

11. Compiling

To view your paper, pagedown requires a web server (since it is based von paged.js)ˆ(open-source library to paginate content in the browser). By compiling a document, R Studio will display your HTML page through a local web server, i.e., paged.js will work in RStudio Viewer.

There are several options, depending on your intention.

- click on the `Knit` button which by default will provide a html document in the RStudio viewer pane (the html will be stored in your working directory)
- use pagedown's `chrome_print` function in the YAML (uncomment line #24 of this `Rmd` file) if you want your html based web page to be (additionally to the html!) printed to be PDF (the PDF will be stored in your working directory)
- to "live"-preview your pages do not click on the `Knit` button but use the **xaringan** (Xie 2021) RStudio add-in *Infinite Moon Reader*. You can simply call the function `xaringan::inf_mr()` (within your console). This will launch a local web server via the **servr** package (Xie 2021a) and display your pages in the RStudio viewer. Each time you save your document (*Ctrl+S*) xaringan updates your pages in the viewer.
-

If you use the option `self_contained: false` (see line #21 of this `Rmd` file) (change to true for a self-contained document, but it'll be a litte slower for Pandoc to render), don't click on the `Knit` button in RStudio. Use instead the **xaringan** (Xie 2021) RStudio add-in *Infinite Moon Reader*.

12. Good practices

Every researcher has his own optimized setup. Currently I would recommend the following:

- Keep all files of your project (that matter for producing the PDF) in one folder without subfolders. You can zip and directly upload that folder to the Harvard dataverse[26].
- Make sure that filenames have a logic to them.
  - Main file with text/code: "paper.rmd," "report.rmd"
  - Data files: "data_xxxxxx.*"
  - Image files: "fig_xxxxxx.*"
  - Tables files: "table_xxxx.*"
  - etc.
  - Ideally, your filenames will correspond to the names in the paper. For instance, Figure 1 in the paper may have a corresponding file called `fig_1_xxxxx.pdf`.
- Use the document outline in R studio (Ctrl + Shift + O) when you work with R Markdown.
- Name rchunks according to what they do or produce:
  - "`fig-...`" for chunks producing figures
  - "`table-...`" for chunks producing tables
  - "`model-...`" for chunks producing model estimates
  - "`import-...`" for chunks importing data
  - "`recoding-...`" for chunks in which data is recoded
- Use "really" informative variable names:
  - Q: What do you think does the variable *trstep* measure? It actually measures trust in the European parliament.
    - How could we call this variable instead? Yes, `trust.european.parliament` which is longer but will probably be understood by another researcher.
  - If your setup is truly reproducible you will probably re-use the variable names that you generate as variable names in the tables you produce. Hence, there is an incentive to use good names.
- Use unique identifiers in the final document:
  - e.g., name the models you estimate "M1," "M2" etc.
  - These unique names should also appear in the published paper.
  - Think of someone who wants to produce Figure 1/Model 1 in your paper but doesn't find it in your code. . .

13. Additional tricks for publishing

- Make your script anonymous
  - Simply put a `<!-- ... -->` around any identifying information, e.g., author names, title footnote etc.
- Counting words
  - Use adobe acrobat (commerical software) to convert your file to a word file. Then open in word and delete all the parts that shouldn't go into the word count. The word count is displayed in the lower right.
  - Use an one of the online services to count your words (search for "pdf word count")

[26] https://dataverse.harvard.edu/

- Appendix: You can change the numbering format for the appendix in the rmd file
    - What is still not possible in this document is to automatically have separate reference sections for paper and appendix.
- Journals may require you to use their tex style: Sometimes you can simply use their template in your rmarkdown file. See here[27] for a PLOS one example.

14. Citation styles

If your study needs to follow a particular citation style, you can set the corresponding style in the header of your `.rmd` document. To do so you have to download the corresponding `.csl` file.

In the present document we use the style of the American Sociological Association and set it in the preamble with `csl: american-sociological-association.csl`. However, you also need to download the respective `.csl` file from the following github page: https://github.com/citation-style-language/styles and copy it into your working directory for it to work.

The github directory contains a wide variety of citation style files depending on what discipline you work in.

References

15. Online appendix

15.1. Attach R session info in appendix

Since R and R packages are constantly evolving you might want to add the R session info that contains information on the R version as well as the packages that are loaded.

```
R version 4.0.4 (2021-02-15)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19041)

Matrix products: default

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] ggplot2_3.3.3     kableExtra_1.3.4  knitr_1.33        stargazer_5.2.2

loaded via a namespace (and not attached):
 [1] tidyselect_1.1.1 xfun_0.23        bslib_0.2.5      purrr_0.3.4
 [5] generics_0.1.0   colorspace_2.0-1 vctrs_0.3.8      htmltools_0.5.1.1
 [9] viridisLite_0.4.0 yaml_2.2.1      utf8_1.2.1       rlang_0.4.11
[13] jquerylib_0.1.4  later_1.2.0      pillar_1.6.1     withr_2.4.2
[17] DBI_1.1.1        glue_1.4.2       lifecycle_1.0.0  stringr_1.4.0
[21] munsell_0.5.0    pagedown_0.14    gtable_0.3.0     rvest_1.0.0
[25] websocket_1.4.0  evaluate_0.14    labeling_0.4.2   httpuv_1.6.1
[29] ps_1.6.0         fansi_0.4.2      highr_0.9        Rcpp_1.0.6
[33] promises_1.2.0.1 scales_1.1.1     webshot_0.5.2    jsonlite_1.7.2
[37] farver_2.1.0     systemfonts_1.0.2 servr_0.22      digest_0.6.27
[41] stringi_1.5.3    bookdown_0.22    processx_3.5.2   dplyr_1.0.6
[45] grid_4.0.4       tools_4.0.4      magrittr_2.0.1   sass_0.4.0
```

[27] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/LDUMNY

```
[49] tibble_3.1.1   crayon_1.4.1   pkgconfig_2.0.3 ellipsis_0.3.2
[53] xml2_1.3.2    assertthat_0.2.1 rmarkdown_2.8.6 svglite_2.0.0
[57] httr_1.4.2    rstudioapi_0.13 R6_2.5.0     compiler_4.0.4
```

15.2. All the code in the paper

To simply attach all the code you used in the PDF file in the appendix see the R chunk in the underlying `.rmd` file:

```
R> remotes::install_github('rstudio/pagedown')
R> install.packages(c("rmarkdown", "knitr", "kableExtra",
R+           "stargazer", "plotly", "knitr"))
R> cat(paste("#", capture.output(sessionInfo()), "\n", collapse =""))
R>  # or use message() instead of cat()
R> x <- 1:10
R> x
R> data <- read.csv("data.csv")
R> head(data)
R> library(stargazer)
R> stargazer(cars,
R+      label="tab1",
R+      table.placement = "H",
R+      type="html",
R+      header=FALSE)
R> library(stargazer)
R> model1 <- lm(speed ~ dist, data = cars)
R> model2 <- lm(speed ~ dist, data = cars)
R> model3 <- lm(dist ~ speed, data = cars)
R>
R> stargazer(model1, model2, model3,
R+      label="tab2",
R+      table.placement = "H",
R+      column.labels = c("M1", "M2", "M3"),
R+      type="html",
R+      model.numbers = FALSE,
R+      header=FALSE)
R> library(knitr)
R> library(kableExtra)
R>
R> kable(mtcars[1:10,], row.names = TRUE,
R+    caption = 'Table with kable() and kablestyling()',
R+    format = "html", booktabs = T) %>%
R+     kable_styling(full_width = T,
R+             latex_options = c("striped",
R+                       "scale_down",
R+                       "HOLD_position"),
R+             font_size = 10)
R>
R>
R> plot(cars$speed, cars$dist)
R> mtcars$cyl <- as.factor(mtcars$cyl) # Convert cyl to factor
R> library(ggplot2)
R> ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl)) + geom_point() +
R+  labs(x="Weight (lb/1000)", y = "Miles/(US) gallon",
R+     shape="Number of \n Cylinders") + theme_classic()
R> knitr::write_bib(c(.packages(), 'pagedown', 'xaringan'), 'index.bib')
R> print(sessionInfo(), local = FALSE)
```

Xie Y (2021). *Xaringan: Presentation Ninja*. Retrieved from https://github.com/yihui/xaringan