

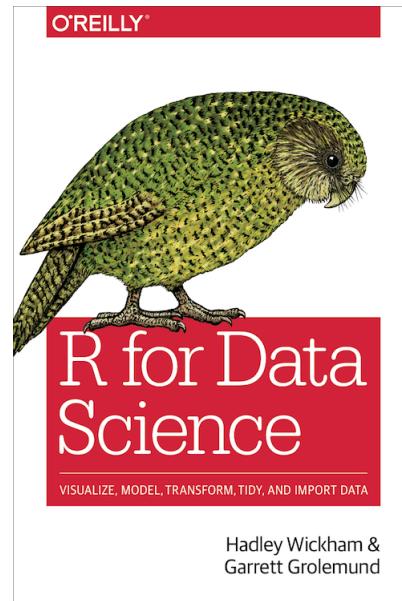
Reproducible Research(2)

문건웅

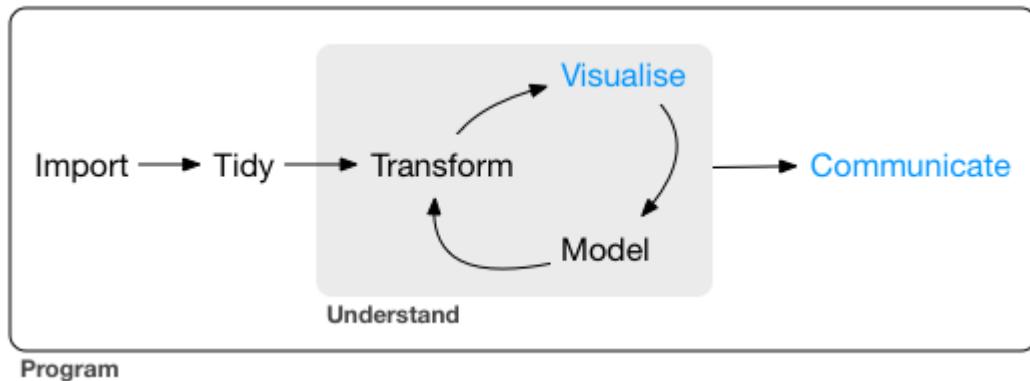
2018/9/4

References

이 강의 슬라이드의 내용 중 많은 부분은 Hadley Wickham의 **R for Data Science** 를 참조하여 만들어졌음을 밝힌다.



Work flow of your study



예제 파일 다운로드

이 강의에 사용되는 예제 파일들은 저자의 [github](#)에서 다운로드 받으실 수 있습니다.

- <https://github.com/cardiomoon/RRworkshop>

다음 예제 파일들이 사용됩니다.

- RmdEx1.Rmd
- RmdEx2.Rmd
- figureSizing.Rmd
- rmdexample.Rmd
- fuel-economy.Rmd

File > Open File.. > RmdEx1.Rmd

```
---
```

```
title: "다이아몬드의 크기"
output:
  html_document: default
  word_document: default
date: '2018-09-04'
```

```
--
```

```
```{r setup, include = FALSE}
library(ggplot2)
library(dplyr)

smaller <- diamonds %>%
 filter(carat <= 2.5)
```
``
```

(참고) 위에 보이는 R코드 한줄을 실행시키려면 R 코드 위에 커서를 위치시킨후 Cmd/Ctrl + Enter 를 우리가 가지고 있는 데이터에는 모두 `r nrow(diamonds)` 개의 다이아몬드가 있다. 이 중 2.5 캐럿

```
```{r plot1, echo = FALSE}
smaller %>%
 ggplot(aes(carat)) +
 geom_freqpoly()
```
```

~/Documents/ownCloud/Documents/tidyverse2 - RStudio

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays an R Markdown document with code chunks and YAML header information.
- File Browser:** Shows the directory structure of the tidyverse2 project, including files like `pev.sav`, `r4ds.R`, and various Rmd and png files.
- Global Environment:** Lists objects such as `class` (62 obs. of 11 variables), `reports` (7 obs. of 3 variables), and `smaller` (48944 obs. of 10 variables).
- Console:** Shows the command used to generate the HTML output: `/Applications/RStudio.app/Contents/MacOS/pandoc/pandoc --RTS -K512m -RTS fuel-economy.utf8.md --to html --from markdown+autolink_bare_uris+ascii_identifiers+tex_math_single_backslash --output fuel-economy-subcompact.html --smart --email-obfuscation none --self-contained --standalone --section-divs --template /Library/Frameworks/R.framework/Versions/3.4/Resources/library/rmarkdown/rmd/h/default.html --no-highlight --variable hid=highlights1 --variable 'theme:bootstrap' --include-in-header /var/folders/ft/_w6lfirs4mz4f8n_r5w_hvh0000gn/T/RtmpbpcQbh/rmarkdown-str157aa2bcc45ef.html --mathjax --variable 'mathjax-url:https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML'.
- Output:** Shows the generated file: Output created: fuel-economy-subcompact.html

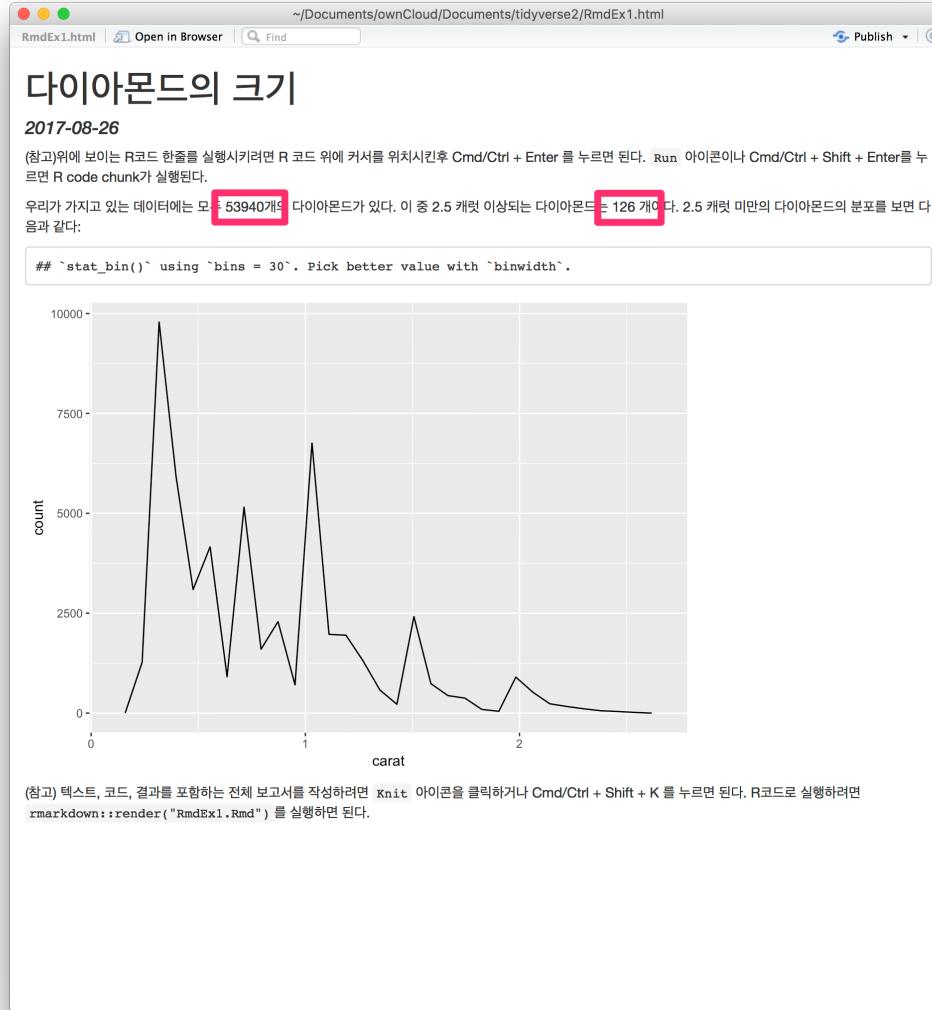
인라인 코드(Inline code)

R markdown 문서의 `text`에 R 코드를 삽입할 때에는 '`r` 코드' 와 같이 사용한다.

- 우리가 가지고 있는 데이터에는 모두 '`r nrow(diamonds)`' 개의 다이아몬드가 있다.
- 이 중 2.5 캐럿 이상되는 다이아몬드는 '`r nrow(diamonds) - nrow(smaller)`' 개이다.

보고서가 만들어질 때에는 다음과 같이 계산 결과가 텍스트에 삽입된다:

- 우리가 가지고 있는 데이터에는 모두 53940개의 다이아몬드가 있다.
- 이 중 2.5 캐럿 이상되는 다이아몬드는 126 개이다.



Knit to Word, PDF, HTML



연습문제

1. *File > New File > R Notebook* 메뉴에서 새 Notebook 을 만들어보자.

연습문제

1. *File > New File > R Notebook* 메뉴에서 새 Notebook 을 만들어보자.
 - 만들어진 Notebook을 읽어보고 R code chunk를 실행시켜보자.

연습문제

1. *File > New File > R Notebook* 메뉴에서 새 Notebook 을 만들어보자.

- 만들어진 Notebook을 읽어보고 R code chunk를 실행시켜보자.
- 코드의 내용을 변경해보고 다시 실행시켜 output의 변화를 보자

Rmd파일에서 텍스트 포맷 (rmdexample.Rmd)

The screenshot shows the RStudio interface with the following details:

- Left Panel (Code Editor):** An Rmd file titled "rmdexample.Rmd" containing code for text formatting, headings, and lists.
- Right Panel (File Browser):** A file browser titled "tidyverse2" showing the directory structure of the project. It includes files like tidyverse.Rproj, Untitled.html, and various R and CSV files.
- Bottom Panel (Console):** Shows the command used to run the Rmd file and the output file generated.

```
~Documents/ownCloud/Documents/tidyverse2 - master - RStudio
RR.Rmd R RmdEx1.Rmd rmdexample.Rmd Go to file/function Addins
1 Text formatting
2
3
4 *italic* or _italic_
5 **bold** __bold__
6 `code`
7 superscript^2^ and subscript~2~
8
9 Headings
10 -----
11
12 # 1st Level Header
13
14 ## 2nd Level Header
15
16 ### 3rd Level Header
17
18 Lists
19 -----
20
21 * Bulleted list item 1
22
23 * Item 2
24
25 * Item 2a
33:1 Lists R Markdown
Console R Markdown ...
~/tidyverse2/rmdexample.Rmd
//mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML'

processing file: rmdexample.Rmd
output file: rmdexample.knit.md

Output created: rmdexample.html

Environment Environment is empty
Files Plots Packages Help Viewer
New Folder Delete Rename More ...
Home Documents ownCloud Documents tidyverse2 ...
Name Size Modified
tidyverse.Rproj 205 B Jul 19, 2017, 8:55
Untitled.html 776.4 KB Jul 28, 2017, 11:1
Untitled.nb.html 833.9 KB Jul 27, 2017, 2:36
Untitled.Rmd 845 B Jul 28, 2017, 11:1
Untitled2.html 794.5 KB Jul 27, 2017, 2:36
Untitled2.nb.html 1.2 MB Jul 28, 2017, 11:2
Untitled2.Rmd 2.9 KB Jul 28, 2017, 11:2
Untitled3.html 791.1 KB Jul 28, 2017, 12:0
Untitled3.Rmd 538 B Jul 28, 2017, 12:0
v59110-data 10 MB Jul 19, 2017, 8:55
v59110-data.zip 359.6 KB Jul 19, 2017, 9:00
v59110.pdf 13.3 KB Jul 19, 2017, 9:00
v59110.R 2.8 KB Jul 19, 2017, 9:00
weather.csv 451.9 KB Jul 19, 2017, 9:00
weather.txt 472 B Aug 17, 2017, 3:2
RmdEx1.Rmd 801.3 KB Aug 17, 2017, 3:2
RmdEx1.html 1.3 MB Aug 17, 2017, 3:2
rmdex1.png
```

Cheatsheets

Help > Cheatsheets > R markdown Cheat Sheet

R Markdown :: CHEAT SHEET

What is R Markdown?

Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist used to produce their work along with the narration that a reader needs to understand the results.

Reproducible Research - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to automatically update the results as a finished report.

Dynamic Documents - You can choose to export the finished report in a variety of formats including HTML, PDF, MS Word, or RTF documents; html or pdf based slides; Notebooks, and more.

Workflow

Open a new .Rmd file | File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.

Write document by editing template

Knit document to create report; use knit button or knit command

Preview Output in IDE window

Publish (optional) to web server

Examine build log in R Markdown console

Or use output file that is saved along side .Rmd

Embed code with knitr syntax

INLINE CODE
Insert with `r`code``. Results appear as text without code.
Built with `getVersion()` Built with `3.2.3`

CODE CHUNKS
One or more lines surrounded with ````{r}` and `````. Place chunk options with curly braces after R. Insert with `knitr:::knitWithCachefile`
`echo=TRUE`
`getVersion()`

GLOBAL OPTIONS
Set with `knitr:::knitWithCachefile`, e.g.
`knitr:::knitWithCachefile(cache = TRUE)`

IMPORTANT CHUNK OPTIONS

| | |
|---|---|
| <code>cache</code> - cache results for future knits (default = <code>FALSE</code>) | <code>dependson</code> - chunk dependencies for caching (default = <code>NULL</code>) |
| <code>path</code> - directory to save cached results in (default = <code>"cache"</code>) | <code>display</code> - display code in output document (default = <code>TRUE</code>) |
| <code>child</code> - Rfle(s) to knit and then include (default = <code>NULL</code>) | <code>engine</code> - code language used in chunk (default = <code>"R"</code>) |
| <code>collapse</code> - collapses all output into single block (default = <code>FALSE</code>) | <code>error</code> - display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = <code>TRUE</code>) |
| <code>comment</code> - prefix for each line of results (default = <code>"#"</code>) | <code>echo</code> - Run code in chunk (default = <code>TRUE</code>) |
| Options not listed above: <code>R.options</code> , <code>auto.opts</code> , <code>autoplot</code> , <code>background</code> , <code>cache.comments</code> , <code>cache.lazycache</code> , <code>cache.rebuild</code> , <code>cache.ways</code> , <code>dpi</code> , <code>engine.opts</code> , <code>engine.path</code> , <code>extra.opts</code> , <code>fig_ex</code> , <code>fig_ip</code> , <code>fig_pa</code> , <code>fig_pp</code> , <code>fig_psp</code> , <code>fig_scrap</code> , <code>fig_shew</code> , <code>fig_shew_box</code> , <code>fig_subcap</code> , <code>interval</code> , <code>out.extra</code> , <code>out.height</code> , <code>out.width</code> , <code>prompt</code> , <code>ref</code> , <code>rlab</code> , <code>render</code> , <code>split</code> , <code>tidy.opts</code> | <code>message</code> - display code messages in document (default = <code>TRUE</code>) |
| | <code>fig.cap</code> - figure caption as character string (default = <code>NULL</code>) |
| | <code>fig.height</code> , <code>fig.width</code> - Dimensions of plots in inches (default = <code>6</code> , <code>4</code>) |
| | <code>highlight</code> - highlight source code (default = <code>TRUE</code>) |
| | <code>include</code> - include chunk in doc after running (default = <code>TRUE</code>) |
| | <code>tidy</code> - tidy code for display (default = <code>FALSE</code>) |
| | <code>warning</code> - display code warnings in document (default = <code>TRUE</code>) |

R Studio

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```
summary(cars)
```

| # | mpg | speed | dist |
|----|------|-------|------|
| ## | High | High | Low |
| 1 | 18.0 | 5 | 20.0 |
| 2 | 18.0 | 6 | 21.0 |
| 3 | 18.0 | 7 | 22.0 |
| 4 | 18.0 | 8 | 22.0 |
| 5 | 18.0 | 9 | 23.0 |
| 6 | 18.0 | 10 | 23.0 |
| 7 | 18.0 | 11 | 24.0 |
| 8 | 18.0 | 12 | 24.0 |
| 9 | 18.0 | 13 | 24.0 |
| 10 | 18.0 | 14 | 24.0 |
| 11 | 18.0 | 15 | 24.0 |
| 12 | 18.0 | 16 | 24.0 |
| 13 | 18.0 | 17 | 24.0 |
| 14 | 18.0 | 18 | 24.0 |
| 15 | 18.0 | 19 | 24.0 |
| 16 | 18.0 | 20 | 24.0 |
| 17 | 18.0 | 21 | 24.0 |
| 18 | 18.0 | 22 | 24.0 |
| 19 | 18.0 | 23 | 24.0 |
| 20 | 18.0 | 24 | 24.0 |
| 21 | 18.0 | 25 | 24.0 |
| 22 | 18.0 | 26 | 24.0 |
| 23 | 18.0 | 27 | 24.0 |
| 24 | 18.0 | 28 | 24.0 |
| 25 | 18.0 | 29 | 24.0 |
| 26 | 18.0 | 30 | 24.0 |
| 27 | 18.0 | 31 | 24.0 |
| 28 | 18.0 | 32 | 24.0 |
| 29 | 18.0 | 33 | 24.0 |
| 30 | 18.0 | 34 | 24.0 |
| 31 | 18.0 | 35 | 24.0 |
| 32 | 18.0 | 36 | 24.0 |
| 33 | 18.0 | 37 | 24.0 |
| 34 | 18.0 | 38 | 24.0 |
| 35 | 18.0 | 39 | 24.0 |
| 36 | 18.0 | 40 | 24.0 |
| 37 | 18.0 | 41 | 24.0 |
| 38 | 18.0 | 42 | 24.0 |
| 39 | 18.0 | 43 | 24.0 |
| 40 | 18.0 | 44 | 24.0 |
| 41 | 18.0 | 45 | 24.0 |
| 42 | 18.0 | 46 | 24.0 |
| 43 | 18.0 | 47 | 24.0 |
| 44 | 18.0 | 48 | 24.0 |
| 45 | 18.0 | 49 | 24.0 |
| 46 | 18.0 | 50 | 24.0 |
| 47 | 18.0 | 51 | 24.0 |
| 48 | 18.0 | 52 | 24.0 |
| 49 | 18.0 | 53 | 24.0 |
| 50 | 18.0 | 54 | 24.0 |
| 51 | 18.0 | 55 | 24.0 |
| 52 | 18.0 | 56 | 24.0 |
| 53 | 18.0 | 57 | 24.0 |
| 54 | 18.0 | 58 | 24.0 |
| 55 | 18.0 | 59 | 24.0 |
| 56 | 18.0 | 60 | 24.0 |
| 57 | 18.0 | 61 | 24.0 |
| 58 | 18.0 | 62 | 24.0 |
| 59 | 18.0 | 63 | 24.0 |
| 60 | 18.0 | 64 | 24.0 |
| 61 | 18.0 | 65 | 24.0 |
| 62 | 18.0 | 66 | 24.0 |
| 63 | 18.0 | 67 | 24.0 |
| 64 | 18.0 | 68 | 24.0 |
| 65 | 18.0 | 69 | 24.0 |
| 66 | 18.0 | 70 | 24.0 |
| 67 | 18.0 | 71 | 24.0 |
| 68 | 18.0 | 72 | 24.0 |
| 69 | 18.0 | 73 | 24.0 |
| 70 | 18.0 | 74 | 24.0 |
| 71 | 18.0 | 75 | 24.0 |
| 72 | 18.0 | 76 | 24.0 |
| 73 | 18.0 | 77 | 24.0 |
| 74 | 18.0 | 78 | 24.0 |
| 75 | 18.0 | 79 | 24.0 |
| 76 | 18.0 | 80 | 24.0 |
| 77 | 18.0 | 81 | 24.0 |
| 78 | 18.0 | 82 | 24.0 |
| 79 | 18.0 | 83 | 24.0 |
| 80 | 18.0 | 84 | 24.0 |
| 81 | 18.0 | 85 | 24.0 |
| 82 | 18.0 | 86 | 24.0 |
| 83 | 18.0 | 87 | 24.0 |
| 84 | 18.0 | 88 | 24.0 |
| 85 | 18.0 | 89 | 24.0 |
| 86 | 18.0 | 90 | 24.0 |
| 87 | 18.0 | 91 | 24.0 |
| 88 | 18.0 | 92 | 24.0 |
| 89 | 18.0 | 93 | 24.0 |
| 90 | 18.0 | 94 | 24.0 |
| 91 | 18.0 | 95 | 24.0 |
| 92 | 18.0 | 96 | 24.0 |
| 93 | 18.0 | 97 | 24.0 |
| 94 | 18.0 | 98 | 24.0 |
| 95 | 18.0 | 99 | 24.0 |
| 96 | 18.0 | 100 | 24.0 |

More details on using R Markdown see <http://markdown.rstudio.com>

.rmd Structure

YAML header
Optional section of render (e.g. `yaml`) options written as key/value pairs (YAML).
At start of file
Between lines of code
Narration formed with markdown, mixed with:

Code Chunks
Any code needed code. Each chunk:
Begins with ````{r}`
ends with `````
R Markdown will run the code and append the results to the doc.
It will use the location of the `.Rmd` file as the **working directory**.

Parameters

Parameterize your documents to reuse with different values (e.g., parameters...)

- Add `params` - Create and set parameters in the header as `params` values of parameters
- Call `params` - Call parameter values in code as `params$name`
- Set `parameters` - Set values with `parameters` argument and the `render` argument of render(`...`rmd`, params = list(n = 1, d = as.date("2015-01-01"))`)`

Today's date is `"`r Sys.Date()`"`

Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

- Add `runif` functions to the YAML header.
- Call `Shiny` functions to embed inputs.
- Call `Shiny` functions to embed reactive output.
- Render with `markdown::runr` or click Run Document in RStudio IDE

How many cars?

| output | html_document | runtime | shiny |
|--------|---------------|---------|-------|
| 5 | 1 | 1 | 1 |
| 6 | 2 | 2 | 2 |
| 7 | 3 | 3 | 3 |
| 8 | 4 | 4 | 4 |
| 9 | 5 | 5 | 5 |
| 10 | 6 | 6 | 6 |

I made a complete app into your document with `shinyApp(ui, server)`

NOTE: All output will render as `html_document`, meaning you must choose an `output` format like `html_document`, and serve it with an active `runApp`.

Markdown Quick Reference

Help > Markdown Quick Reference

The screenshot shows the RStudio interface with several windows open:

- File Explorer:** Shows the file structure of the current project, including files like `tidyverse.Rmd` and `rr.Rmd`.
- Console:** Displays R code and its output, including the creation of an R Markdown document and the opening of a help viewer.
- Help Viewer:** A modal window titled "Markdown Quick Reference" containing sections on Emphasis, Headers, and Lists.

```

~/.Documents/ownCloud/Documents/tidyverse2 - master - RStudio

RmdEx1.Rmd * RR.Rmd * rmarkdown.Rmd * rmdeexample.Rmd *
Addins

Environment History Git
Import Dataset List
Global Environment

Environment is empty

Files Plots Packages Help Viewer
Markdown Quick Reference Find in Topic

Markdown Quick Reference

R Markdown is an easy-to-write plain text format for creating dynamic documents and reports. See Using R Markdown to learn more.

Emphasis
*italic* **bold**
_italic_ bold

Headers
# Header 1
## Header 2
### Header 3

Lists
Unordered List
* Item 1
* Item 2
  + Item 2a
  + Item 2b

Ordered List

```

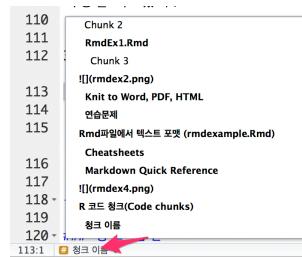
R 코드 청크(Code chunks)

Rmd 파일에 코드 청크를 삽입하는 세 가지 방법

1. 키보드 단축키: Cmd/Ctrl + Alt + I
2. 에디터 툴바의 Insert 버튼
3. 청크 경계기호인 '```{r}'과 '```' 삽입

청크 이름

코드 청크에 `{'{r by-name}`과 같이 이름을 붙일 수 있다. 이름을 붙이면 다음과 같은 장점이 있다.



1. 에디터 왼쪽 아래의 drop-down code navigator로 코드 청크를 쉽게 찾을 수 있다.
2. 코드 청크에 의해 만들어지는 그래프에 이름이 붙기 때문에 그림을 다른 곳에서 이용할 수 있다.
3. 코드 청크에 의해 생기는 연산 결과를 캐쉬에 넣어 다른 곳에서 연산을 다시 수행하지 않고 사용할 수 있다.

setup 청크

청크 이름 중 *setup*은 특별한 의미가 있다. `setup` 코드 청크는 다른 코드가 수행되기 전 한번만 수행된다.

청크 옵션

청크 헤드에 옵션을 주어 청크의 결과물을 조절할 수 있다. 약 60개의 옵션이 제공된다. (<http://yihui.name/knitr/options/>). 이 중 중요한 옵션은 다음과 같다.

- eval = FALSE
- include = FALSE
- echo = FALSE
- message = FALSE, warning =FALSE
- results = 'hide', fig.show='hide'
- error = TRUE

청크 옵션

| Option | Run code | Show code | Output | Plots | Messages | Warnings |
|-------------------|----------|-----------|--------|-------|----------|----------|
| eval = FALSE | - | | - | - | - | - |
| include = FALSE | | - | - | - | - | - |
| echo = FALSE | | - | | | | |
| results = 'hide' | | | - | | | |
| fig.show = 'hide' | | | | - | | |
| message = FALSE | | | | | - | |
| warning = FALSE | | | | | | - |

테이블

디폴트로 R markdown 에서는 데이터 프레임을 콘솔에서 보는 것과 같이 인쇄해준다.

```
mtcars[1:5, ]
```

| | 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 |

```
knitr::kable(mtcars[1:5,], caption = "A knitr kable",format="html")
```

| | 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 |

그 외에 xtable, stargazer, pander, tables, ascii 패키지 등을 이용하면 표 출력을 보다 세밀하게 조절 할 수 있다. 다음은 ztable의 예이다.

```
require(ztable)
ztable(mtcars[1:5], type="html", caption = "Table: A ztable")
```

| Table: A ztable | | | | | | | | | | | |
|-------------------|-------|------|--------|--------|------|------|-------|------|------|------|------|
| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
| Mazda RX4 | 21.00 | 6.00 | 160.00 | 110.00 | 3.90 | 2.62 | 16.46 | 0.00 | 1.00 | 4.00 | 4.00 |
| Mazda RX4 Wag | 21.00 | 6.00 | 160.00 | 110.00 | 3.90 | 2.88 | 17.02 | 0.00 | 1.00 | 4.00 | 4.00 |
| Datsun 710 | 22.80 | 4.00 | 108.00 | 93.00 | 3.85 | 2.32 | 18.61 | 1.00 | 1.00 | 4.00 | 1.00 |
| Hornet 4 Drive | 21.40 | 6.00 | 258.00 | 110.00 | 3.08 | 3.21 | 19.44 | 1.00 | 0.00 | 3.00 | 1.00 |
| Hornet Sportabout | 18.70 | 8.00 | 360.00 | 175.00 | 3.15 | 3.44 | 17.02 | 0.00 | 0.00 | 3.00 | 2.00 |

ztable을 이용하면 row color, column color cell color 등을 쉽게 바꿀수 있다.

```
ztable(mtcars[1:5,]) %>%
  addRowColor(4,"yellow") %>%
  addColColor(5,"orange") %>%
  addCellColor(4,5,"red") %>%
  print(type="html")
```

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|-------------------|-------|------|--------|--------|------|------|-------|------|------|------|------|
| Mazda RX4 | 21.00 | 6.00 | 160.00 | 110.00 | 3.90 | 2.62 | 16.46 | 0.00 | 1.00 | 4.00 | 4.00 |
| Mazda RX4 Wag | 21.00 | 6.00 | 160.00 | 110.00 | 3.90 | 2.88 | 17.02 | 0.00 | 1.00 | 4.00 | 4.00 |
| Datsun 710 | 22.80 | 4.00 | 108.00 | 93.00 | 3.85 | 2.32 | 18.61 | 1.00 | 1.00 | 4.00 | 1.00 |
| Hornet 4 Drive | 21.40 | 6.00 | 258.00 | 110.00 | 3.08 | 3.21 | 19.44 | 1.00 | 0.00 | 3.00 | 1.00 |
| Hornet Sportabout | 18.70 | 8.00 | 360.00 | 175.00 | 3.15 | 3.44 | 17.02 | 0.00 | 0.00 | 3.00 | 2.00 |

```
DT::datatable(mtcars[,1:5])
```

Caching

- 정상적으로 Rmd 파일에서 문서가 만들어질 때 아무것도 없는 상태에서 시작한다.
- 코드에서 모든 연산을 수행하기 때문에 재현성 측면에서 유리하다.
- 계산에 시간이 많이 걸리는 경우 *cache = TRUE* 옵션을 주면 코드 청크의 결과물을 디스크에 저장했다가 코드가 바뀌지 않은 경우 결과를 다시 사용한다.

Cache 사용시 주의할 점

디폴트로 캐쉬는 dependency에 의존하지 않는다. 예를 들어 다음 코드를 보자.

```
```{r raw_data}
raw_data <- readr::read_csv("a_very_large_file.csv")
```

```{r processed_data, cache = TRUE}
processed_data <- raw_data %>%
 filter(!is.na(import_var)) %>%
 mutate(new_variable = complicated_transformation(x, y, z))
```
```

```

processed\_data 청크는 raw\_data 청크에 의존한다. 하지만 이때 process\_data 청크는 dplyr 파이프라인이 바뀐 경우에는 다시 수행되나 read\_csv() 함수가 바뀐 경우에는 다시 수행되지 않는다. 이를 해결하기 위해서는 *dependson* 청크옵션을 사용하면 된다.

```
```{r processed_data, cache = TRUE, dependson = "raw_data"}
processed_data <- raw_data %>%
  filter(!is.na(import_var)) %>%
  mutate(new_variable = complicated_transformation(x, y, z))
```
```

```

- *dependson* 청크옵션에는 의존성이 있는 모든 청크의 문자형벡터가 포함되어야 한다. 그래야만 의존성이 있는 청크들 중 하나라도 바뀐 경우 결과가 업데이트 된다.

- *dependson* 청크옵션에는 의존성이 있는 모든 청크의 문자형벡터가 포함되어야 한다. 그래야만 의존성이 있는 청크들 중 하나라도 바뀐 경우 결과가 업데이트 된다.
- 이 경우 `a_very_large_file.csv`이 바뀐 경우에는 수행되지 않는다. `cache.extra` 옵션을 사용하면 파일의 변화를 추적할 수 있다.

```
```{r raw_data, cache.extra = file.info("a_very_large_file.csv")}  
rawdata <- readr::read_csv("a_very_large_file.csv")
```
```

- *dependson* 청크옵션에는 의존성이 있는 모든 청크의 문자형벡터가 포함되어야 한다. 그래야만 의존성이 있는 청크들 중 하나라도 바뀐 경우 결과가 업데이트 된다.
- 이 경우 `a_very_large_file.csv`이 바뀐 경우에는 수행되지 않는다. `cache.extra` 옵션을 사용하면 파일의 변화를 추적할 수 있다.

```
```{r raw_data, cache.extra = file.info("a_very_large_file.csv")}  
rawdata <- readr::read_csv("a_very_large_file.csv")
```
```

- 캐쉬의 사용이 점점 복잡해지다면 `knitr::clean_cache()`를 사용해 cache를 주기적으로 청소해주는 것이 좋다.

- *dependson* 청크옵션에는 의존성이 있는 모든 청크의 문자형벡터가 포함되어야 한다. 그래야만 의존성이 있는 청크들 중 하나라도 바뀐 경우 결과가 업데이트 된다.
- 이 경우 `a_very_large_file.csv`이 바뀐 경우에는 수행되지 않는다. `cache.extra` 옵션을 사용하면 파일의 변화를 추적할 수 있다.

```
```{r raw_data, cache.extra = file.info("a_very_large_file.csv")}  
rawdata <- readr::read_csv("a_very_large_file.csv")
```
```

- 캐쉬의 사용이 점점 복잡해지다면 `knitr::clean_cache()`를 사용해 `cache`를 주기적으로 청소해주는 것이 좋다.
- 청크의 이름을 붙일 때에는 **David Robinson**의 조언대로 그 청크에서 만들어지는 가장 중요한 객체의 이름을 따서 짓는 것을 권유한다. 그럴 경우 *dependson* 옵션을 이해하기 쉬워진다.

Global Options

디폴트 청크 옵션이 원하는 것과 맞지 않는 경우 `knitr::opts_chunks$set()` 함수를 써서 한꺼번에 바꿀 수 있다.

```
knitr::opts_chunks$set(  
  comment = NA,  
  collapse = TRUE,  
  message = FALSE,  
  warning = FALSE  
)
```

만일 보고서에서 R code를 숨기고 싶은 경우에는 다음과 같이 할 수 있다.

```
knitr::opts_chunks$set(  
  echo = FALSE  
)
```

숫자의 출력

텍스트에 숫자를 넣을 경우 `format()` 함수를 사용하면 소숫점 아래 자릿수나 1000단위에 맞추어 숫자를 보기 좋게 출력할 수 있다.

```
comma <- function(x) format(x, digits = 2, big.mark = ",")  
comma(3452345)
```

```
[1] "3,452,345"
```

```
comma(.12358124331)
```

```
[1] "0.12"
```

연습문제

1. diamonds 데이터를 사용하여 cut, colour, clarity에 따른 다이아몬드의 크기를 탐색하는 section을 추가하라. 보고서를 보는 사람이 R을 모른다는 가정하에 global option을 사용하여 R code를 출력하지 않게 하라.
2. RmdEx1.Rmd 파일 끝에 가장 큰 20개의 다이아몬드를 기술하는 섹션을 추가하라.
3. RmdEx1.Rmd 파일에 comma() 함수를 사용하여 숫자를 보기좋게 출력하고 2.5 캐럿보다 큰 다이아몬드의 백분율을 추가하라.
4. R markdown 문서를 하나 만들고 d가 b 와 c에 의존하고 b와 c는 a에 의존하는 코드 청크를 만들고 각 청크에 lubridate::now() 를 출력하고 cache = TRUE 로 지정하라. 이 문서를 통해 캐쉬를 이해하고 있는지 확인하라.

YAML 헤더

문서 전체의 셋팅을 조절하려면 YAML 헤더의 인수를 조절하면 된다. YAML은 원래 *yet another markup language* 의 약자였으나 요즘은 *YAML Ain't Markup Language*로 사용되고 있다. R markdown 문서에서는 출력을 세부적으로 조절하기 위해 YAML 헤더를 사용한다.

```
---
```

```
title: "다이아몬드의 크기"
output:
  html_document: default
  word_document: default
date: '2017-08-26'
params:
  size: 2.5


```

```


```

인수(Parameters)

R 마크다운 문서에는 보고서를 출력할 때 값을 정할 수 있는 인수들을 사용할 수 있다. 인수들은 중요한 값을 달리하여 같은 보고서를 출력할 때 유용하다. 예를 들어 지점별로 판매 보고서를 출력하거나 학생별로 시험 결과를 출력할 때, 또는 나라별로 인구통계를 출력할 때 사용할 수 있다. 인수를 정의할 때는 *params* 를 사용한다.

다음 예는 2.5 캐럿 이하의 다이아몬드의 분포를 보는 RmdEx1.Rmd 파일에서 2.5를 인수로 지정한 예이다.(RmdEx2.Rmd)

```
---
```

```
title: "다이아몬드의 크기"
output:
  html_document: default
  word_document: default
date: '2017-08-26'
params:
  size: 2.5
---
```

```
```{r setup, include = FALSE}
library(ggplot2)
library(dplyr)

smaller <- diamonds %>%
 filter(carat <= params$size)
```
```

우리가 가지고 있는 데이터에는 모두 `r nrow(diamonds)`개의 다이아몬드가 있다. 이 중 `r param`

```
```{r, echo = FALSE}
smaller %>%
 ggplot(aes(carat)) +
 geom_freqpoly()
```
```

연습문제

2.5 캐럿이 아닌 3.0 캐럿 이하의 다이아몬드의 분포를 보는 문서를 작성해 보자.

명령행에서 Rmd 파일의 인수를 지정

- 사용파일 : fuel-economy.Rmd

fuel-economy.Rmd

```
---
```

```
output: html_document
```

```
params:
```

```
  my_class: "suv"
```

```
---
```

```
```{r setup, include = FALSE}
```

```
library(ggplot2)
```

```
library(dplyr)
```

```
class <- mpg %>% filter(class == params$my_class)
```

```
```
```

```
# Fuel economy for `r params$my_class`s
```

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

```
knitr::kable(class[1:6,])
```

```
```
```

```
```{r, message = FALSE}
```

```
ggplot(class, aes(displ, hwy)) +
```

```
 geom_point() +
```

```
 geom_smooth(se = FALSE)
```

*suv*가 아닌 *pickup* 차종에 대해 같은 보고서를 출력하려면 다음과 같이 할 수 있다.

```
rmarkdown::render("fuel-economy.Rmd", params=list(my_class="pickup"))
```

mpg 데이터의 차의 종류는 다음과 같다.

```
unique(mpg$class)
```

```
[1] "compact" "midsize" "suv" "2seater" "minivan"
[6] "pickup" "subcompact"
```

모든 차의 종류에 대해 같은 보고서를 출력하려면 어떻게 해야 할까?

```
reports <- tibble(
 class = unique(mpg$class),
 filename = stringr::str_c("fuel-economy-", class, ".html"),
 params = purrr::map(class, ~ list(my_class = .))
)
reports
```

```
A tibble: 7 x 3
 class filename params
 <chr> <chr> <list>
1 compact fuel-economy-compact.html <list [1]>
2 midsize fuel-economy-midsize.html <list [1]>
3 suv fuel-economy-suv.html <list [1]>
4 2seater fuel-economy-2seater.html <list [1]>
5 minivan fuel-economy-minivan.html <list [1]>
6 pickup fuel-economy-pickup.html <list [1]>
7 subcompact fuel-economy-subcompact.html <list [1]>
```

```
reports %>%
 select(output_file = filename, params) %>%
 purrr::pwalk(rmarkdown::render, input = "fuel-economy.Rmd")
```

# 그림의 사용

figureSizing.Rmd

```

```

```
title: "Figure sizing"
output:
 html_document:
 code_folding: hide
 toc: true
 toc_float: true
 pdf_document: default
 word_document: default

```

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, comment=NA, message=FALSE,
                      warning=FALSE,
                      fig.width=6, fig.asp=0.618,
                      out.width="70%", fig.align="center")
```

One figure
```{r one fig, fig.cap="Figure 1.bula bula"}
```

웹 R의 reproducible research

Web-R.org

Data Table Descriptive ExploPlots Compare Categorical Multiple Group ▾ Regression ▾ survival Tree Analysis ▾

Diagnosis ROC curve R command Report/Plot Options Pubmed Wordcloud Propensity Score Matching Principal component

Validation citation information PPT List

Saved PowerPoint List

	type	title	code
1	ggplot	Exploratory plot for data acs	ggDot(acs,aes(x=sex,y=age,color=sex),binwidth=0.5)
2	table	Summary of Statistics	numSummaryTable('age','sex',data=acs, vanilla=FALSE,method=3,lang='en',addstat=TRUE,paired=FALSE,id='None')
3	Rcode	Independent two-sample test	out=lm(age~sex,data=acs) shapiro.test(resid(out)) wilcox.test(age~sex, data=acs)
4	ggplot	Exploratory plot for data aacs	ggDot(acs,aes(x=Dx,y=age,color=Dx),binwidth=0.5)
5	table	Summary of Statistics	numSummaryTable('age','Dx',data=acs, vanilla=FALSE,method=3,lang='en',addstat=TRUE,paired=FALSE,id='None')
6	Rcode	Independent test for multiple groups	out=lm(age~Dx,data=acs) shapiro.test(resid(out)) bartlett.test(age~Dx,data=acs) cat(' Bonferroni Outlier test') car::outlierTest(out) kruskal.test(age~ factor(Dx),data=acs) varA <-acs\$age varB <-acs\$Dx data <- data.frame(varA=varA,varB=varB) summary(npcomp::mctp(varA~varB,data=data))

Upload PPTList(*.csv) Browse... No file selected

Reset PPT List

[download as csv](#) [download as HTML](#) [download as PDF](#) [download as PPTx](#) [download as Word](#)