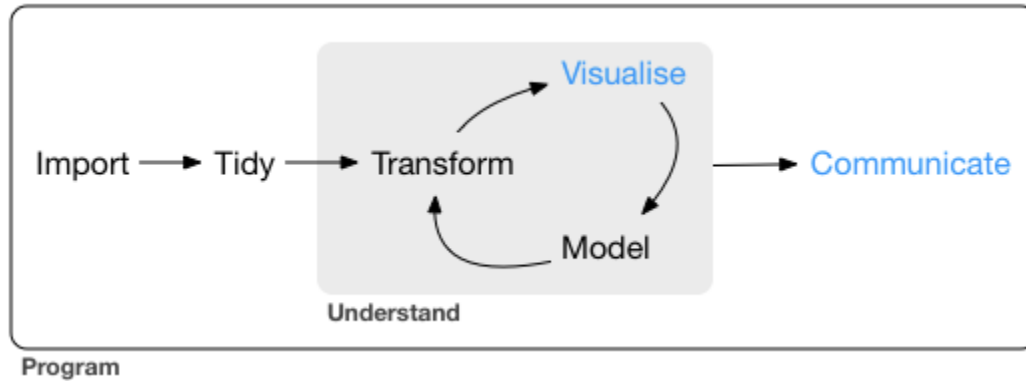


Reproducible Research

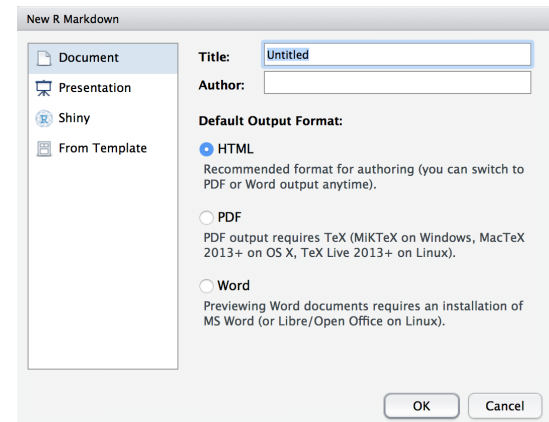
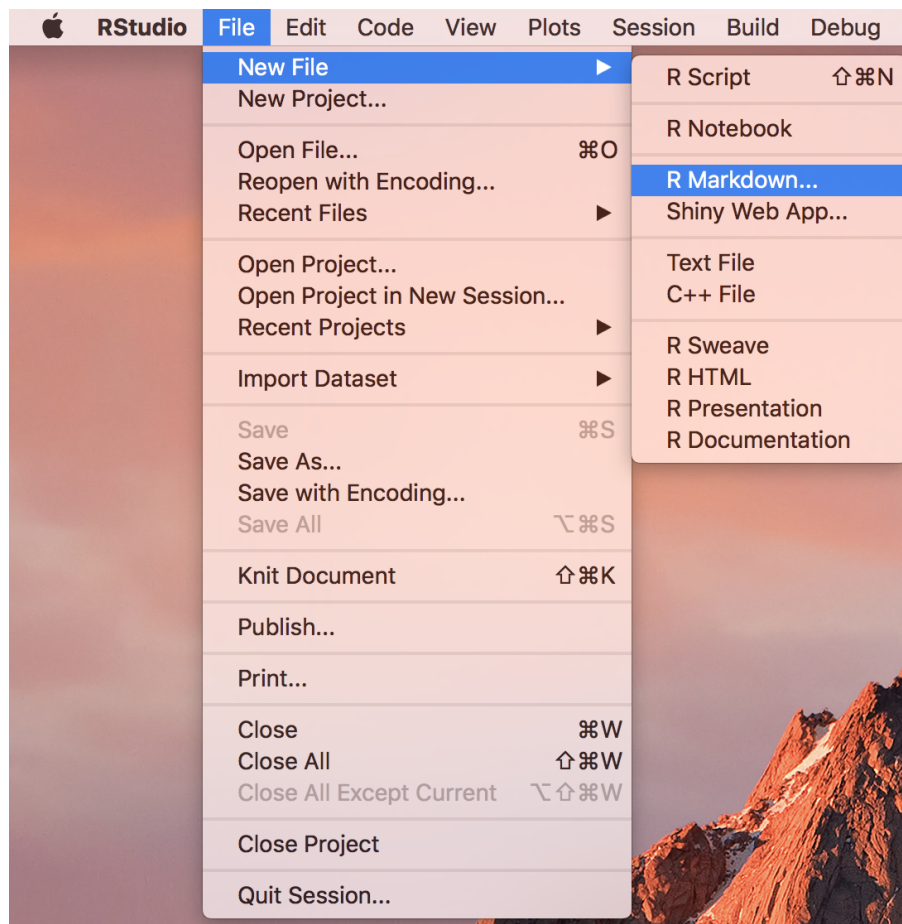
문건웅

2017/8/26

Work flow of your study



R markdown의 기초



RmdEx1.Rmd

```
---  
title: "다이아몬드의 크기"  
output:  
  html_document: default  
  word_document: default  
date: '2017-08-26'  
---  
  
```${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, echo = FALSE}  
smaller %>%
 ggplot(aes(carat)) +
 geom_freqpoly()
```,
```

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

Go to file/function Addins

RmdEx2.Rmd x RmdEx1.Rmd x RR.Rmd x rmdexample.Rmd x fuel-economy.Rmd x Untitled1* x

Knit Run

1 ---
 2 title: "다이아몬드의 크기"
 3 output:
 4 html_document: default
 5 word_document: default
 6 date: '2017-08-26'
 7 ---

YAML header

9- ```{r setup, include = FALSE}
 10 library(ggplot2)
 11 library(dplyr)
 12
 13 smaller <- diamonds %>%
 14 filter(carat <= 2.5)
 15 ```

R code chunk

17 (참고)위에 보이는 R코드 한줄을 실행시키려면 R 코드 위에 커서를 위치시킨 후
 Cmd/Ctrl + Enter 를 누르면 된다. 'Run' 아이콘이나 Cmd/Ctrl + Shift + Enter를
 누르면 R code chunk가 실행된다.

Text(Human Readable)

18 우리가 가지고 있는 데이터에는 모두 `r nrow(diamonds)`개의 다이아몬드가 있다. 이
 중 2.5 캐럿 이상되는 다이아몬드는 `r nrow(diamonds) - nrow(smaller)`개이다.
 2.5 캐럿 미만의 다이아몬드의 분포를 보면 다음과 같다:

20
 21- ```{r, echo = FALSE}
 22 smaller %>%
 23 ggplot(aes(carat)) +
 24 geom_freqpoly()
 25 ```

20:1 (Top Level) R Markdown

Console R Markdown x

~/Documents/ownCloud/Documents/tidyverse2/

/Applications/RStudio.app/Contents/MacOS/pandoc/pandoc +RTS -K512m -RTS fuel-economy.u
 tf8.md --to html --from markdown+autolink_bare_uris+ascii_identifiers+tex_math_single_
 backslash --output fuel-economy-subcompact.html --smart --email-obfuscation none --sel
 f-contained --standalone --section-divs --template /Library/Frameworks/R.framework/Ver
 sions/3.4/Resources/Library/rmarkdown/rmd/h/default.html --no-highlight --variable hig
 hlightjs=1 --variable 'theme:bootstrap' --include-in-header /var/folders/ft/_w6lflrs4m
 z4f8n_r5w_h7v00000gn/T/~/RtmpbpcQbh/rmarkdown-str157aa2bcc45ef.html --mathjax --variabl
 e 'mathjax-url:https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLor
 MML'

Output created: fuel-economy-subcompact.html

>

Environment History Git

Global Environment

Data

class	62 obs. of 11 variables
reports	7 obs. of 3 variables
smaller	48944 obs. of 10 variables

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home Documents ownCloud Documents tidyverse2

Name	Size	Modified
pew.sav	6.3 MB	Jul 19, 2017, 8:59 AM
r4ds.R	12 KB	Jul 19, 2017, 8:59 AM
rmarkdown-formats.Rmd	12.8 KB	Aug 17, 2017, 4:07 PM
rmarkdown-workflow.Rmd	3.9 KB	Aug 17, 2017, 4:07 PM
rmarkdown.bib	801 B	Jul 28, 2017, 12:04 PM
rmarkdown.Rmd	24 KB	Aug 17, 2017, 4:07 PM
RmdEx1.docx	32.9 KB	Aug 18, 2017, 8:44 AM
rmdex1.png	1.3 MB	Aug 17, 2017, 3:24 PM
RmdEx1.Rmd	1.1 KB	Aug 18, 2017, 10:13 AM
rmdex2.png	471.6 KB	Aug 17, 2017, 3:26 PM
rmdex3.png	1.1 MB	Aug 17, 2017, 3:36 PM
rmdex4.png	1.2 MB	Aug 17, 2017, 4:34 PM
rmdexample.html	695.1 KB	Aug 17, 2017, 3:35 PM
rmdexample.Rmd	715 B	Aug 17, 2017, 3:35 PM
RR.html	2.9 KB	Aug 18, 2017, 8:15 AM
RR.Rmd	1.3 KB	Aug 18, 2017, 8:15 AM
rr1.png	1.1 MB	Aug 17, 2017, 2:30 PM
rr2.png	266.7 KB	Aug 17, 2017, 2:31 PM
string.R	2.9 KB	Jul 19, 2017, 9:01 AM
tb-clean-1.tex	553 B	Jul 19, 2017, 8:59 AM
tb-raw.tex	866 B	Jul 19, 2017, 8:59 AM
tb.csv	265.1 KB	Jul 19, 2017, 8:59 AM
tb_raw.csv	11.9 KB	Jul 19, 2017, 8:59 AM

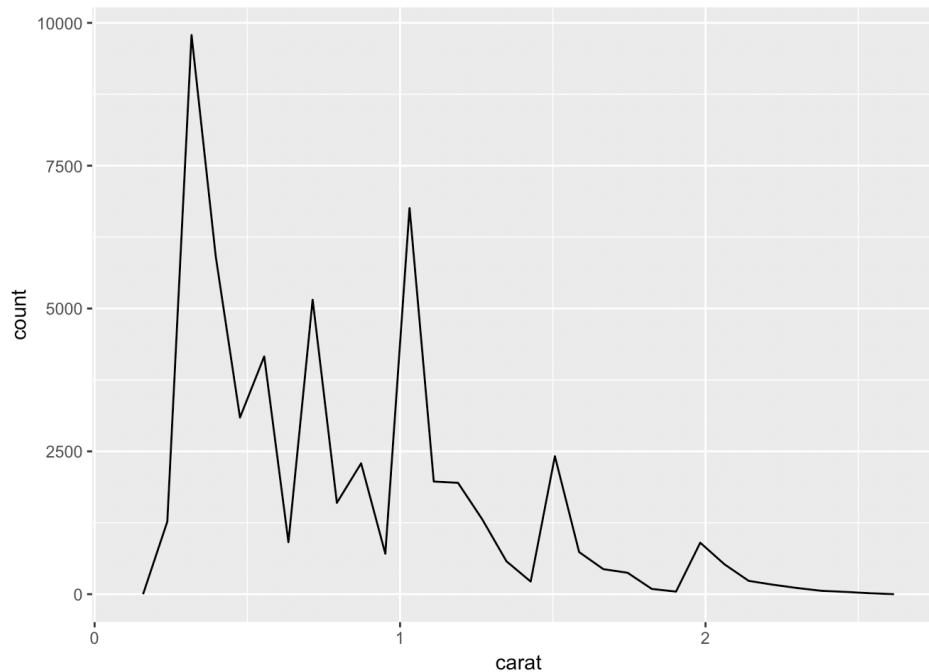
다이아몬드의 크기

2017-08-26

(참고)위에 보이는 R코드 한줄을 실행시키려면 R 코드 위에 커서를 위치시킨후 Cmd/Ctrl + Enter 를 누르면 된다. Run 아이콘이나 Cmd/Ctrl + Shift + Enter를 누르면 R code chunk가 실행된다.

우리가 가지고 있는 데이터에는 모 53940개의 다이아몬드가 있다. 이 중 2.5 캐럿 이상되는 다이아몬드는 126 개이다. 2.5 캐럿 미만의 다이아몬드의 분포를 보면 다음과 같다:

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



(참고) 텍스트, 코드, 결과를 포함하는 전체 보고서를 작성하려면 Knit 아이콘을 클릭하거나 Cmd/Ctrl + Shift + K 를 누르면 된다. R코드로 실행하려면 `rmarkdown::render("RmdEx1.Rmd")` 를 실행하면 된다.

Knit to Word, PDF, HTML



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)

Cheatsheets

Help > Cheatsheets > R markdown Cheat Sheet

Markdown Quick Reference

Help > Markdown Quick Reference

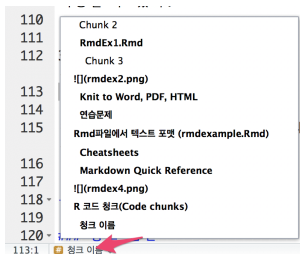
R 코드 청크(Code chunks)

Rmd 파일에 코드 청크를 삽입하는 세가지 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 navigator**로 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`

체크 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")
```

A knitr kable

	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.

```
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

```
ztable::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])
```

Show entriesSearch:

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.85
Hornet 4 Drive	21.4	6	258	110	3.08
Hornet Sportabout	18.7	8	360	175	3.15
Valiant	18.1	6	225	105	2.76
Duster 360	14.3	8	360	245	3.21
Merc 240D	24.4	4	146.7	62	3.69
Merc 230	22.8	4	140.8	95	3.92
Merc 280	19.2	6	167.6	123	3.92

Showing 1 to 10 of 32 entries

Previous

1

2

3

4

Next

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 코드청크의 결과물을 디스크에 저장했다가 코드가 바뀌지 않은 경우 결과를 다시 사용한다.

Cache 사용시 주의할 점

디폴트로 캐쉬는 `dependency` 에 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 바뀔 경우 결과가 업데이트 된다.
- 이 경우 `a_very_large_file.csv` 이 바뀔 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 code**를 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 1000단 위에 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 보기 좋게 출력하고 2.5 캐럿보다 큰 다이아몬드의 백분율을 추가하라.
4. R markdown 문서를 하나 만들고 d가 b 와 c에 의존하고 b와 c는 a에 의존하는 코드청크를 만들고 각 청크에 lubraidate::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()  
` `
```

fuel-economy.Rmd

```
---  
output: html_document  
params:  
  my_class: "suv"  
---  
  
```${r} setup, include = FALSE}  
library(ggplot2)
library(dplyr)

class <- mpg %>% filter(class == params$my_class)
```${r}  
knitr::kable(class[1:6,])  
```${r}  

Fuel economy for `${r} params$my_class`

```${r, message = FALSE}  
ggplot(class, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(se = FALSE)  
```${r}
```

*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 <chr>	filename <chr>	params <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.with=6,fig.asp=0.618,
```

```
                      out.width="70%",fig.align="center")
```

```
```
```

```
One figure
```

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


인라인 코드(Inline code)

R markdown 문서의 text에 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.