

한국한의학연구원, 구본초

통계 패키지 활용

2020년도 2학기 충남대학교 정보통계학과 강의 노트



Contents

List of Tables	v
List of Figures	vii
Course Overview	ix
1 R Markdown	1
1.1 R Markdown의 구성	2
1.2 R Markdown 문서 시작하기	6
1.3 R Markdown 기본 문법(syntax)	9
1.3.1 텍스트 문법	9
1.3.2 Block-level elements	12
1.3.3 수식표현(math expression)	14
1.4 R Code Chunks	15
1.5 인라인(inline) R 코드	31
1.6 YAML	32
1.7 참고문헌 인용	34



List of Tables

1.1 코드 실행 관련 청크	17
1.2 소스 코드 출력 결과 관련 청크	19
1.3 코드 서식 관련 청크	25
1.4 Plot 출력 관련 청크	27



List of Figures

1.1 R markdown 세계 (https://ulyngs.github.io/rmarkdown-workshop-2019 에서 발췌)	1
1.2 R markdown structure	4
1.3 R Markdown 의 최종 결과물 산출과정 (http://applied-r.com/project-reporting-template/)	5
1.4 test.html 문서 화면 (저장 폴더 내 ‘test.html’을 크롬 브라우저로 실행)	8
1.5 장난꾸러기	11
1.6 Chunk anatomy (https://ulyngs.github.io/rmarkdown-workshop-2019 에서 발췌)	16
1.7 청크 옵션 results = ‘markup’인 경우 rmd vs. md 파일 비교	22
1.8 청크 옵션 results = ‘asis’인 경우 rmd vs. md 파일 비교	22
1.9 Taj Mahal	29
1.10 Scatterplot of the car dataset	30



Course Overview

R을 이용한 데이터 분석 시 CRAN에 등록된 패키지를 활용한다. 적절한 패키지의 활용은 데이터 분석의 효율을 증대할 뿐 아니라 분석의 재현성을 향상할 수 있다. 본 강의는 지난학기에 학습한 통계프로그래밍언어 강의 내용의 연속선상에서 진행할 예정이며, 해당 강의에서 학습한 내용들을 기반으로 데이터 분석 및 그 결과에 대한 보고서 작성, 그리고 R 생성 파일에 대한 버전 관리 방법에 대해 알아보려고 한다.

교과 목표

- R Markdown의 이해와 활용
- R 프로그래밍 능력 향상 및 통계 시뮬레이션의 이해
- R을 이용한 데이터 분석 실습
- R을 이용한 기초 통계분석
- 텍스트 마이닝에 대한 이해
- Shiny, plotly 를 활용한 동적 문서 및 시각화 이해
- RStudio + Github을 이용한 버전관리 이해

선수과목

통계학 개론 통계 프로그래밍 언어

수업 방법

- 강의: 30 %
- 실험/실습: 70 %

평가방법

- 중간고사: 35 %
- 기말고사: 35 %
- 출석: 10 %
- 과제: 20 %

교재

별도의 교재 없이 본 강의 노트로 수업을 진행할 예정이며, 수업의 이해도 향상을 위해 아래 소개할 도서 및 웹 문서 등을 참고할 것을 권장함.

참고문헌

- R Markdown Cookbook¹ (Xie et al., 2020)
- bookdown: Authoring Books and Technical Documents with R Markdown² (Xie, 2016)
- R과 knitr를 활용한 데이터 연동형 문서 만들기 (고석범, 2014)
- R for data science³ (Wickham and Grolemund, 2016)
- Statistical Computing with R (Rizzo, 2019)

¹<https://bookdown.org/yihui/rmarkdown-cookbook/>

²<https://bookdown.org/yihui/bookdown/>

³<https://r4ds.had.co.nz/>

- R programming for data science⁴ (Peng, 2016)
- Text mining with R⁵ (Silge and Robinson, 2017)

⁴<https://bookdown.org/rdpeng/rprogdatascience/>

⁵<https://www.tidytextmining.com/>



1

R Markdown

Sketch

- 동일한 문서에 코드, 결과, 텍스트가 동시에 있을 수 있을까?
- 만약 결과와 도표가 자동으로 생성된 경우 데이터가 변경 되더라도 자동으로 문서를 업데이트 할 수 있을까?
- 최종 완료한 문서가 미래에도 열 수 있을까?
- 이러한 모든 과정이 매우 쉽다면??

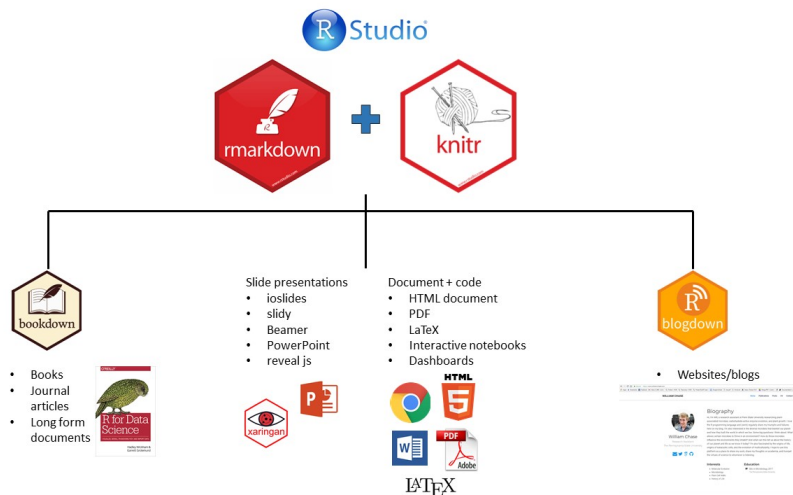


FIGURE 1.1: R markdown 세계 (<https://ulyngs.github.io/rmarkdown-workshop-2019> 에서 발췌)

1.1 R Markdown의 구성



본 절의 내용 중 일부는 지난 학기 강의노트 1.7절과 중복되거나 재구성한 내용이 포함됨.

1. R Markdown은 R 코드와 분석 결과(표, 그림 등)을 포함한 문서 또는 콘텐츠를 제작하는 도구로 일반적으로 아래 열거한 형태로 활용함
 - 문서 또는 논문(pdf, html, docx)
 - 프리젠테이션(pdf, html, pptx)
 - 웹 또는 블로그
2. 재현가능(reproducible)한 분석 및 연구¹ 가능
 - 신뢰성 있는 문서 작성
 - Copy & paste를 하지 않고 효율적 작업 가능

R 마크다운 파일 = .Rmd 확장자를 가진 일반 텍스트 파일

```
---
title: "Untitled.Rmd"
date: "2020-09-11"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

¹과학적 연구의 결과물을 오픈소스로 내놓고 누구라도 검증 가능

R Markdown

Markdown은 HTML, PDF 및 MS Word 문서를 작성하기 위한 간단한 형식 지정 구문입니다.

R Markdown 사용에 대한 자세한 내용은 <<http://rmarkdown.rstudio.com>>을 참조하십시오.

****Knit**** 버튼을 클릭하면 두 가지를 모두 포함하는 문서가 생성됩니다.

문서에 포함된 R 코드 청크의 출력 내용뿐 아니라

다음과 같이 R 코드 청크를 포함할 수 있습니다.

```
```${r cars}
summary(cars)
```
```

Including Plots

You can also embed plots, for example:

```
```${r pressure, echo=FALSE}
plot(pressure)
```
```

`echo = FALSE` 매개 변수가 코드 청크에 추가되었습니다.

플롯을 생성한 R 코드의 인쇄를 방지합니다.

위 R Markdown 문서는 아래 그림과 같이 **YAML**, **Markdown 텍스트**, **Code Chunk** 세 부분으로 구성됨.

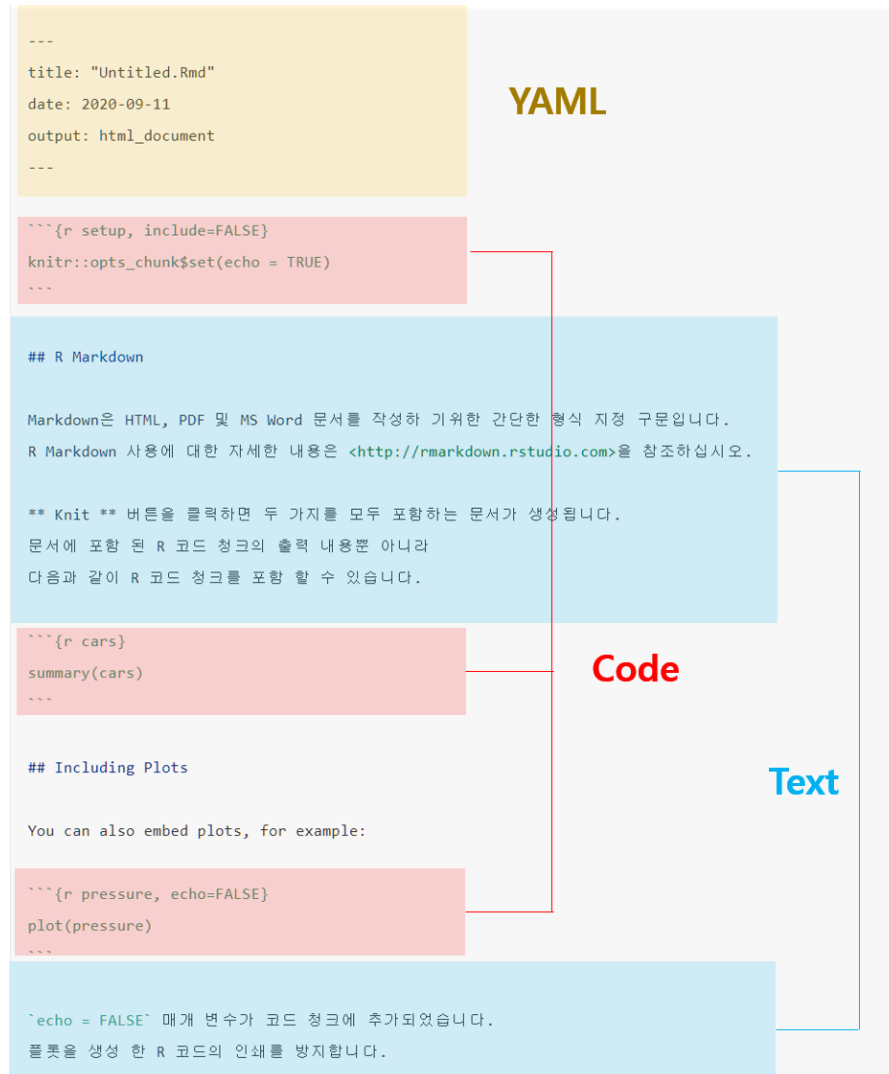


FIGURE 1.2: R markdown structure

YAML (YAML Ain't Markup Language)

- R Markdown 문서의 metadata로 문서의 맨 처음에 항상 포함(header) 되어야 함.
- R Markdown 문서의 최종 출력 형태(html, pdf, docx, pptx 등), 제목, 저자, 날짜 등의 정보 등을 포함

최종 문서 생성 과정

- Rmd 파일을 knitr 을 통해 .md 파일로 변환 후 pandoc 이라는 문서 변환기를 통해 원하는 문서 포맷으로 출력

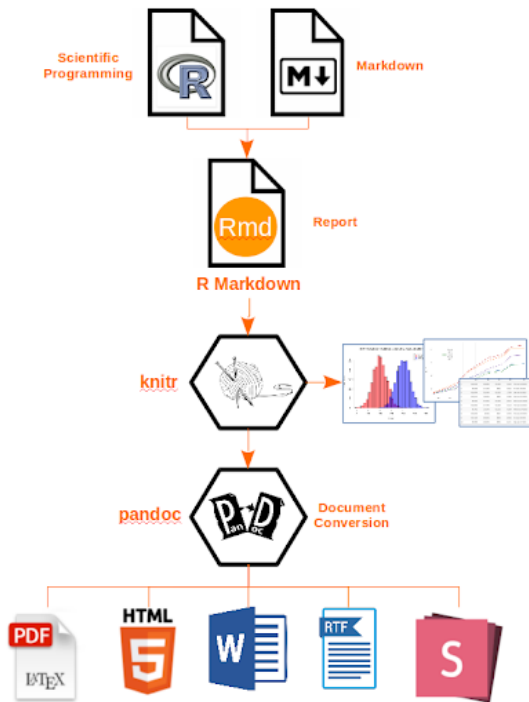


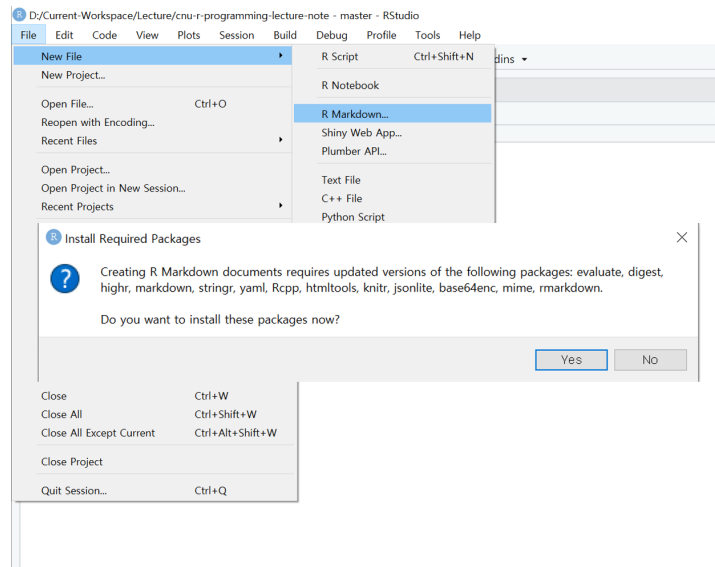
FIGURE 1.3: R Markdown의 최종 결과물 산출과정 (<http://applied-r.com/project-reporting-template/>)

1.2 R Markdown 문서 시작하기

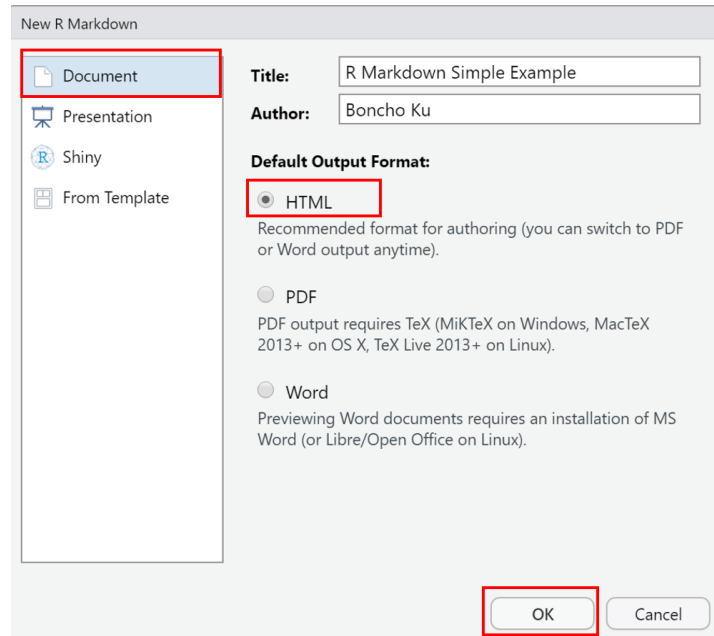
- R Markdown 문서 생성 : [File] -> [New File] -> [R Markdown...]을 선택



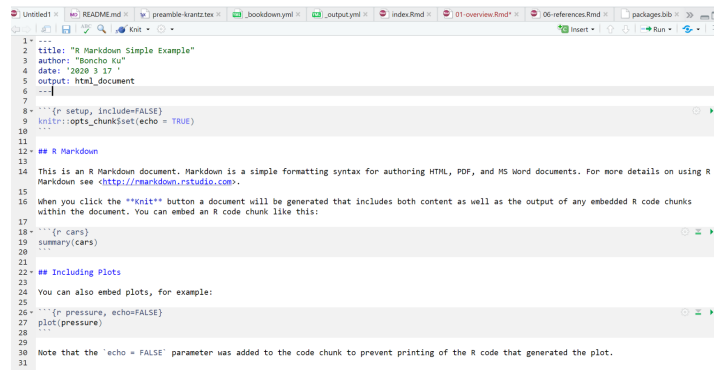
RStudio를 처음 설치하고 위와 같이 진행할 경우 아래와 같은 패키지 설치 여부를 묻는 팝업 창이 나타남. 패키지 설치 여부에 [Yes]를 클릭하면 R Markdown 문서 생성을 위해 필요한 패키지들이 자동으로 설치



- 설치 완료 후 R Markdown으로 생성할 최종 문서 유형 선택 질의 창이 나타남. 아래 창에서 제목(Title)과 저자(Author) 이름 입력 후 [OK] 버튼 클릭 (Document, html 문서 선택)

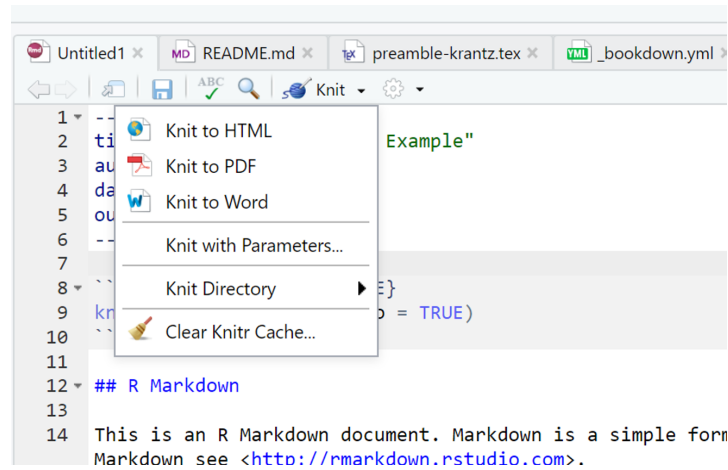


- 아래 그림과 같이 새로운 문서 창이 생성되고 `test.Rmd` 파일로 저장²



- 문서 상단에 Knit 아이콘을 클릭 후 Knit to HTML 클릭 또는 문서 아무 곳에 커서를 위치하고 단축키 [Ctrl] + [Shift] + [K] 입력

²[RStudio 프로젝트]에서 생성한 폴더 내에 파일 저장



- knitr + R Markdown + pandoc → html 파일 생성 결과

R Markdown Simple Example

Boncho Ku
2020 3 17

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0   Min.   : 2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean    : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.    :120.00
```

Including Plots

You can also embed plots, for example:

FIGURE 1.4: test.html 문서 화면 (저장 폴더 내 ‘test.html’을 크롬 브라우저로 실행)

1.3 R Markdown 기본 문법(syntax)

R Markdown의 기본 문법은 Rstudio 풀다운 메뉴 [Help] → [Markdown Quick Reference] 에서 확인 가능

1.3.1 텍스트 문법

강조(emphasis)

- 이탤릭체: **italic1**, *__italic2__* → *italic1*, *italic2*
- 볼드(굵은)체: ***bold1***, **___bold2___** → **bold1**, **bold2**

Inline code

- ‘inline code’ → `inline code`

아래/위 첨자(sub/superscript)

- subscript~2~ → subscript₂
- superscript^2^ → superscript²

삭제표시(strike through)

- ~~strikethrough~~ → ~~strikethrough~~

생략표시(ellipsis)

- ... → ⋯

긴/짧은 대쉬(en/emd-dash)

- 짧은 대쉬: -- → –
- 긴 대쉬: --- → —

특수문자 탈출 지정자

- `*`, `_`, `\~`, `\\` → `*`, `_`, `~`, `\`

하이퍼링크

`-[text](link)` → 통계프로그래밍언어³

외부그림 삽입

- `![image title](path/to/image):` ![장난꾸러기](figures/son-02.jpg)

강제 줄바꿈 (line breaks)

- 하나의 줄에서 공백(space) 두 개 이상 또는 백슬레시(\) 입력 후 [Enter]

```
End a line with two spaces to start
a new paragraph
```

End a line with two spaces to start a new paragraph

```
End a line with two spaces to start\
a new paragraph
```

```
End a line with two spaces to start
a new paragraph
```

각주 (footnote)

- `A footnote^[주석내용]` → A footnote⁴

³<https://zorba78.github.io/cnu-r-programming-lecture-note>

⁴주석내용



FIGURE 1.5: 장난꾸러기

주석(comment)

- `<!-- this is a comment that won't be shown -->` →



RStudio에서 단축키 `[Ctrl] + [Shift] + [C]`를 통해 전체 line 에 대해 주석처리 가능

1.3.2 Block-level elements

장/절(header)

- # Header 1 (chapter, 장)
- ## Header 2 (section, 절)
- ### Header 3 (subsection, 관)

목록(list)

- 비순서(unordered) 목록: -, *, + 중 어느 하나로 입력 가능

```
- one item
* two item
  + sub-item 1
  + sub-item 2
    - subsub-item 1
    - subsub-item 2
```

- one item
- two item
 - sub-item 1
 - sub-item 2
 - * subsub-item 1
 - * subsub-item 2
- 순서(ordered) 목록: 비순서 목록의 기호 대신 숫자로 리스트 생성

```
1. the first item
  - sub-item 1
```



```
2. the second item
3. the third item
```

1. the first item
 - sub-item 1
 2. the second item
 3. the third item
- 같은 숫자로 적어도 순서대로 목록 생성

```
1. the first item
  - sub-item 1
1. the second item
1. the third item
```

1. the first item
 - sub-item 1
2. the second item
3. the third item

인용구(blockquote): >로 시작

```
> "There are three kinds of lies: lies, damn lies, and statistics"
>
> --- Benjamin Disraeli
```

“There are three kinds of lies: lies, damn lies, and statistics”

— Benjamin Disraeli

1.3.3 수식표현 (math expression)

- 줄 안에 수식 입력 시 `$수식표현$` 으로 입력
- 수식 display style (보통 교과서에 정리 및 정의에 기술된 수식들) 적용 시 `$$ ~ $$` 안에 수식 입력
- 수식 표현은 LaTeX 의 수식 표현을 동일하게 준용 (<https://www.latex4technics.com/>, <https://latex.codecogs.com/legacy/eqneditor/editor.php> 에서 수식 입력 명령어 학습 가능)
- LaTeX 수식 입력 코드는
- 예시

$$P(X = x) = f(x; n, p) = \binom{n}{x} p^x (1 - p)^{n-x}$$

- Inline equation: `$P(X = x) = f(x; n, p) = \{n \choose x\} p^x (1-p)^{n-x}$` $\rightarrow P(X = x) = f(x; n, p) = \binom{n}{x} p^x (1 - p)^{n-x}$
- Math block: `$$P(X = x) = f(x; n, p) = \{n \choose x\} p^x (1-p)^{n-x}$$`

$$P(X = x) = f(x; n, p) = \binom{n}{x} p^x (1 - p)^{n-x}$$

- `$ $` 또는 `$$ $$` 안에 LaTeX에서 제공하는 수식 함수 사용 가능

```

$$\begin{array}{ccc}
x_{11} & x_{12} & x_{13} \\
\end{array}

```

$$\begin{array}{ccc} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{array}$$
$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$
$$\begin{aligned} g(X_n) &= g(\theta) + g'(\tilde{\theta})(X_n - \theta) \\ \sqrt{n}[g(X_n) - g(\theta)] &= g'(\tilde{\theta}) \sqrt{n}[X_n - \theta] \end{aligned}$$

- 실제 R code가 실행되는 부분임

- Code chunk 실행 시 다양한 옵션 존재(본 강의에서는 몇 개의 옵션만 다를 것이며, 더 자세한 내용은 <https://yihui.org/knitr/options/> 또는 R Markdown 레퍼런스 가이드⁵ 참조
- Code chunk는 ````${r}````로 시작되며 `r`은 code 언어 이름을 나타냄.
- Code chunk는 `````로 종료
- R Markdown 문서 작성 시 단축키 [Ctrl] + [Alt] + [I]를 입력하면 Chunk 입력창이 자동 생성됨
- Code chunk의 옵션 조정을 통해 코드의 출력여부, 코드 출력 시 코드의 출력 형태, 코드의 결과물 출력 조정 가능

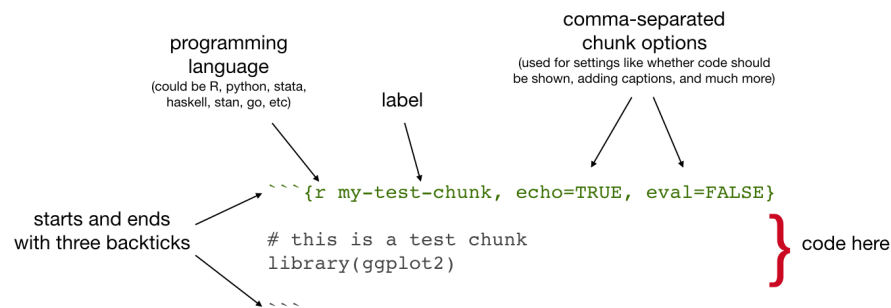


FIGURE 1.6: Chunk anatomy (<https://ulyngs.github.io/rmarkdown-workshop-2019> 에서 발췌)

자주 활용하는 chunk 옵션

코드 실행 관련 청크

⁵<https://rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>

TABLE 1.1: 코드 실행 관련 청크

| Chunk 옵션 | Default | 설명 |
|----------|---------|--------------------------------|
| eval | TRUE | R 실행 (코드 실행 결과)에 대응하는 결과 출력 여부 |
| include | TRUE | 출력 문서에 코드 청크의 내용을 포함할지 여부 |

```
```{r ex01-1, eval=TRUE}
summary(iris)
hist(iris$Sepal.Length)
```
```

```
```{r ex01-2, eval=FALSE}
summary(iris)
hist(iris$Sepal.Length)
```
```

```
#청크 옵션 eval=TRUE
summary(iris)
```

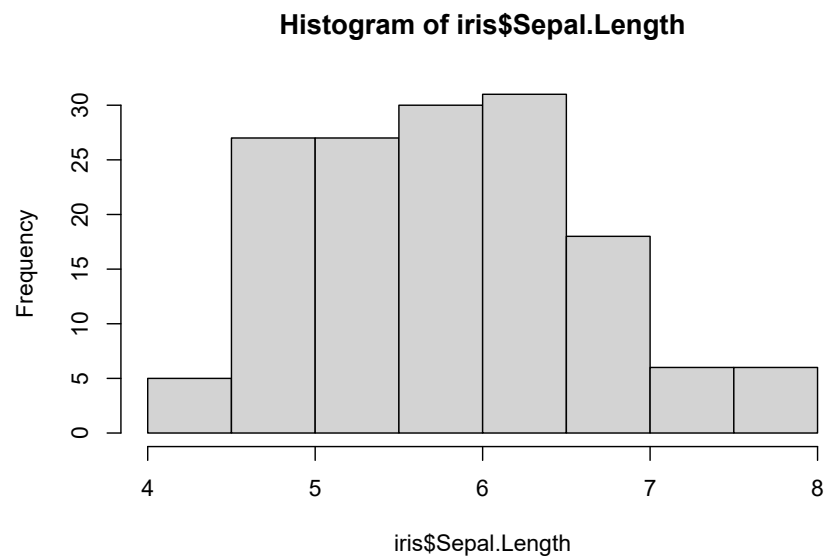
```

      Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
Min.    :4.300   Min.    :2.000   Min.    :1.000   Min.    :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean    :5.843   Mean    :3.057   Mean    :3.758   Mean    :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.    :7.900   Max.    :4.400   Max.    :6.900   Max.    :2.500

      Species
setosa    :50
versicolor:50
```

```
virginica :50
```

```
hist(iris$Sepal.Length)
```



```
#체크 옵션 eval=FALSE  
summary(iris)  
hist(iris$Sepal.Length)
```

소스 코드 출력 (텍스트) 결과 관련 링크

TABLE 1.2: 소스 코드 출력 결과 관련 청크

| Chunk 옵션 | Default | 설명 |
|----------|---------|---|
| echo | TRUE | R 실행 결과에 대응하는 코드 출력 여부 |
| results | markup | 출력 결과 포맷 지정에 위한 옵션으로 추가적으로 3가지 옵션 선택 가능: 'hide', 'asis', 'hold', 'markup' |
| error | TRUE | 코드 또는 스크립트에 구문오류 메시지 출력 여부 |
| message | TRUE | 코드로부터 생성된 메시지 출력 여부 |
| warning | TRUE | 경고 메시지 출력 여부 |

- **echo:** 코드 청크에 작성한 R-script 출력 여부 결정
 - `echo = FALSE` 이면 소스 코드 출력 없이 그림 결과만 출력

```

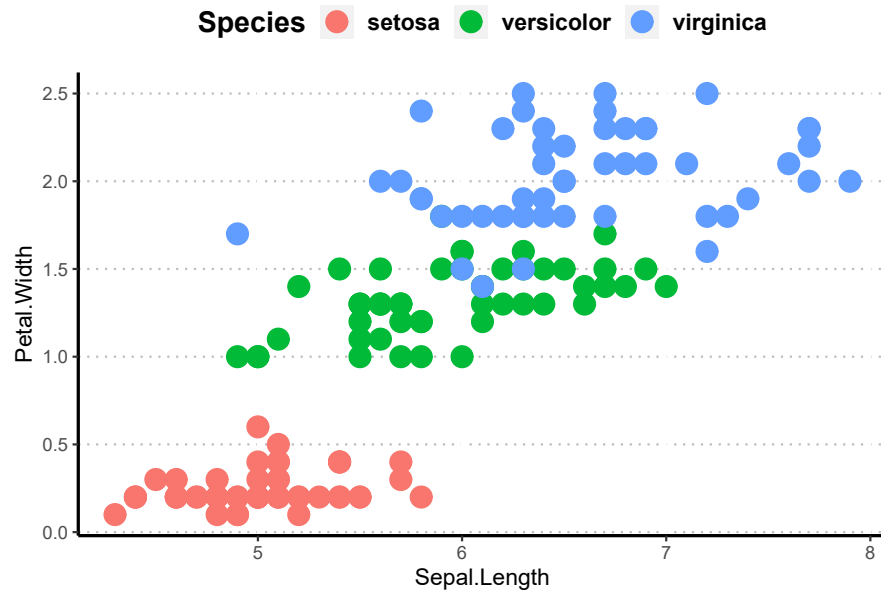
```{r ex01-2, echo=TRUE}
require(ggthemes) # ggtheme 패키지 불러오기
require(ggpubr) # ggpubr 패키지 불러오기
iris %>%
 ggplot(aes(x = Sepal.Length, y = Petal.Width, color = Species)) +
 geom_point(size = 5) +
 theme_pubclean() +
 theme(axis.line = element_line(size = 0.8),
 legend.title = element_text(face = "bold", size = 15),
 legend.text = element_text(face = "bold", size = 12))
```

```

```
```{r ex01-3, echo=FALSE}
require(ggthemes) # ggtheme 패키지 불러오기
require(ggpubr) # ggpubr 패키지 불러오기
iris %>%
 ggplot(aes(x = Sepal.Length, y = Petal.Width, color = Species)) +
 geom_point(size = 5) +
 theme_pubclean() +
 theme(axis.line = element_line(size = 0.8),
 legend.title = element_text(face = "bold", size = 15),
 legend.text = element_text(face = "bold", size = 12))

```
```

```
# echo = TRUE
require(ggthemes) # ggtheme 패키지 불러오기
require(ggpubr) # ggpubr 패키지 불러오기
iris %>%
  ggplot(aes(x = Sepal.Length, y = Petal.Width, color = Species)) +
  geom_point(size = 5) +
  theme_pubclean() +
  theme(axis.line = element_line(size = 0.8),
        legend.title = element_text(face = "bold", size = 15),
        legend.text = element_text(face = "bold", size = 12))
```

- **results:** 코드의 텍스트 출력 결과 포맷 지정
 - **markup** (default): 코드 청크 내 스크립트의 출력 형태에 따라 텍스트 출력 결과를 mark-up
 - **asis:** 변환하지 않은 원래 R 출력 결과 그대로 (as is) 출력
 - **hide:** R 스크립트로 생성된 텍스트 출력을 보여주지 않음 (warning, message 출력 제외)
 - **hold:** 코드 청크로 생성된 모든 소스 및 출력을 단일 블록으로 축소

```
# results = 'markup'인 경우 아래 텍스트를 mark-up
# (이 경우 아래 텍스트는 "" "" 블록 처리)한 결과를 md 파일로 전송
cat("I'm raw **Markdown** content.\n")
```

```
I'm raw **Markdown** content.
```

```
# results = 'asis' 인 경우 텍스트를 그대로 md 파일에 입력
cat("I'm raw **Markdown** content.\n")
```

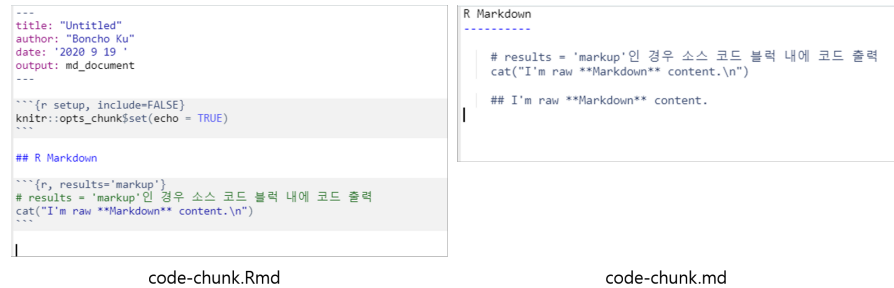


FIGURE 1.7: 청크 옵션 `results = 'markup'`인 경우 `rmd` vs. `md` 파일 비교

I'm raw **Markdown** content.



FIGURE 1.8: 청크 옵션 `results = 'asis'`인 경우 `rmd` vs. `md` 파일 비교

```
# results = 'hide'
cat("I'm raw **Markdown** content.\n")
```

텍스트 결과를 출력하지 않음

```
# results = 'hold'가 아닌 경우 한 라인 별 출력 결과 생성
x <- rnorm(10)
x
```

```
[1] 0.11397949 -1.11975525 -0.39546848 -1.48319594 0.86714230 -0.65739039
[7] -0.94070684 -0.06004431 -0.84747143 0.11357520
```

```
y <- rnorm(10, 1, 2)
y
```

```
[1]  2.6074654  2.2176548 -1.4640987  1.3247515  1.2189210 -0.5720802
[7] -0.8809272 -1.9154795 -1.5656312  0.8735466
```

```
x + y
```

```
[1]  2.7214449  1.0978996 -1.8595671 -0.1584444  2.0860633 -1.2294706
[7] -1.8216340 -1.9755238 -2.4131027  0.9871218
```

```
# results = 'hold'인 경우 코드 부분과 출력 부분이 따로 블록 처리
```

```
x <- rnorm(10)
```

```
x
```

```
y <- rnorm(10, 1, 2)
```

```
y
```

```
x + y
```

```
[1] -1.0223661  0.2992861 -1.1081556 -0.0126041 -1.1982391 -1.5091321
[7] -0.8100077 -0.9633556  0.6873936  0.1792799
[1] -3.37265758  0.97343197 -0.02659187  1.06986991  5.90701592  0.33479442
[7]  0.74497495 -1.38770572  2.16830646  2.78405177
[1] -4.39502367  1.27271807 -1.13474743  1.05726582  4.70877684 -1.17433764
[7] -0.06503275 -2.35106137  2.85570008  2.96333162
```

- **error**: 코드 청크 내 스크립트에 오류에 대한 보존 여부(`stop()`)
 - 기본적으로 Rmarkdown 컴파일 시 **error**에 대한 옵션이 **FALSE**이기 때문에 스크립트(코드)에 오류가 포함되면 컴파일이 정지됨.
 - **error = TRUE** 이면 오류 메시지를 포함한 텍스트 결과를 출력

```
3x <- 3
```

```
x <- 25 # 위 행이 구문 오류를 포함하고 있기 때문에
```

```
# 오류 이후의 코드는 실행되지 않음
x
```

Error: <text>:1:2: 예상하지 못한 기호(symbol)입니다.

```
1: 3x
```

- `message/warning`: 텍스트 출력물 중 경고(`warning`, `warning()` 함수의 출력 결과) 메시지 출력 여부 결정

```
# message = TRUE 인 경우 함수 message 출력
testit <- function() {
  message("testing package startup messages")
  packageStartupMessage("initializing ...", appendLF = FALSE)
  Sys.sleep(1)
  packageStartupMessage(" done")
} # help(message) 예시 중 발췌

testit()
```

```
testing package startup messages
```

```
initializing ... done
```

```
# message=FALSE -> 메시지 출력하지 않음
testit()
```

```
# 경고 메시지 출력
x <- c(1, 2, "new", 4:10)
x <- as.numeric(x)
```

Warning: 강제형변환에 의해 생성된 NA 입니다

코드 서식 관련 링크 옵션

TABLE 1.3: 코드 서식 관련 청크

| Chunk 옵션 | Default | 설명 |
|-----------|---------|---|
| comment | TRUE | 소스 코드 실행 출력의 각 줄 앞에 붙는 표시문자 출력 여부: 기본 값은 '##' 임 |
| highlight | TRUE | 구문 강조 여부 |
| prompt | FALSE | R 프롬프트 출력 여부 |
| tidy | FALSE | R 소스 코드 출력 정리 여부 |

- **comment**: 텍스트 출력물에 주석 표시(default)를 함으로써 소스 코드와 출력 결과를 동시 선택과 복사를 가능(##는 주석 표시이기 때문에 실행되지 않음)
 - 주석 표시를 제거하고 싶다면 `comment = NA` 또는 `comment = ''`

```
# 디폴트 comment 사용
summary(iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300
## Median :5.800 Median :3.000 Median :4.350 Median :1.300
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500
## Species
## setosa :50
## versicolor:50
## virginica :50
##
##
##
```

- highlight: 구문 강조 표시 여부
 - highlight=FALSE 일 때 소스 코드 출력 결과

```
# highlight=FALSE
```

```
iris %>%
  ggplot(aes(x = Sepal.Length, y = Petal.Width, color = Species)) +
  geom_point(size = 5) +
  theme_pubclean() +
  theme(axis.line = element_line(size = 0.8),
        legend.title = element_text(face = "bold", size = 15),
        legend.text = element_text(face = "bold", size = 12))
```

- prompt: R 콘솔 상 프롬프트 >, + 출력 여부

```
> # prompt = TRUE 인 경우 코드 출력 결과
> require(ggthemes) # ggtheme 패키지 불러오기
> require(ggpubr) # ggpubr 패키지 불러오기
> iris %>%
+   ggplot(aes(x = Sepal.Length, y = Petal.Width, color = Species)) +
+   geom_point(size = 5) +
+   theme_pubclean() +
+   theme(axis.line = element_line(size = 0.8),
+         legend.title = element_text(face = "bold", size = 15),
+         legend.text = element_text(face = "bold", size = 12))
```

- tidy: 코드를 사용자가 지정 (혹은 `formatR::tidy_source()` 함수에 초기 값으로 지정된 코드 정리 값) 한 줄 당 문자 길이 등을 반영해 코드를 정리
 - tidy=TRUE 인 경우 자동으로 줄 바꿈

```
> # tidy = FALSE 인 경우 코드 출력 결과
> require(ggthemes) # ggtheme 패키지 불러오기
```

```

> require(ggpubr) # ggpubr 패키지 불러오기
> iris %>% ggplot(aes(x = Sepal.Length, y = Petal.Width, color = Species)) + geom_point(size = 5) +

> # tidy = TRUE 인 경우 코드 출력 결과
> require(ggthemes) # ggtheme 패키지 불러오기
> require(ggpubr) # ggpubr 패키지 불러오기
> iris %>% ggplot(aes(x = Sepal.Length, y = Petal.Width, color = Species)) + geom_point(size = 5) +
+   theme_pubclean() + theme(axis.line = element_line(size = 0.8), legend.title = element_text(face
+   size = 15), legend.text = element_text(face = "bold", size = 12))

```

그림(plot) 출력 관련 청크 옵션

TABLE 1.4: Plot 출력 관련 청크

| Chunk 옵션 | Default | 설명 |
|----------------------|---------|--|
| fig.align | default | 최종 문서에 plot 정렬 방식 결정 (center, left, right) |
| fig.height/fig.width | 7 | 그림 크기 (단위: 인치) |
| fig.cap | NULL | 그림 캡션 (문자열 입력) |
| dpi | 72 | dot per inch: 출력 그림 해상도 |

알아두면 좋은 청크 형태

Setup 청크

- 일반적으로 Rmarkdown 문서는 YAML 헤더 뒤에 전역적 청크 옵션 지정과 R 패키지를 불러오는 것으로 시작
- 청크 옵션은 `knitr::opts_chunk$set(청크 옵션 지정)` 형태로 지정 가능
- 다음은 RStudio 에서 Rmd 문서 생성 시 맨 처음 나오는 코드 청크 예시임

```
```{r ex01-2, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

- 일반적 활용 예시

```
```{r option-init, include=FALSE}
knitr::opts_chunk$set(root.dir = '../..', # 프로젝트 폴더 지정
 eval = TRUE,
 echo = FALSE,
 cache = FALSE,
 include = TRUE,
 tidy = TRUE,
 tidy.opts = list(blank=FALSE, width.cutoff=120), # 소스 출력길이
 message = FALSE,
 warning = FALSE,
 engine = "R", # Chunks will always have R code, unless noted
 error = TRUE,
 fig.path="Figures/", # Set the figure options
 fig.align = "center",
 fig.width = 7,
 fig.height = 7,
 fig.keep='all', fig.retina=2)
```
```

이미지 불러오기


```
```{r, fig.cap = "Taj Mahal"}
knitr::include_graphics("figures/taj.JPG", dpi = NA)
```
```



FIGURE 1.9: Taj Mahal

```
```{r, fig.cap = "Taj Mahal"}
cars %>%
 ggplot(aes(x = speed, y = dist)) +
 geom_point(size = 5) +
 theme_tufte(base_size = 15) # ggtheme::theme_tufte()
```
```

R 생성 도표 포함

테이블 삽입

- 가장 간단한 테이블은 `knitr::kable()` 함수를 통해 생성 가능

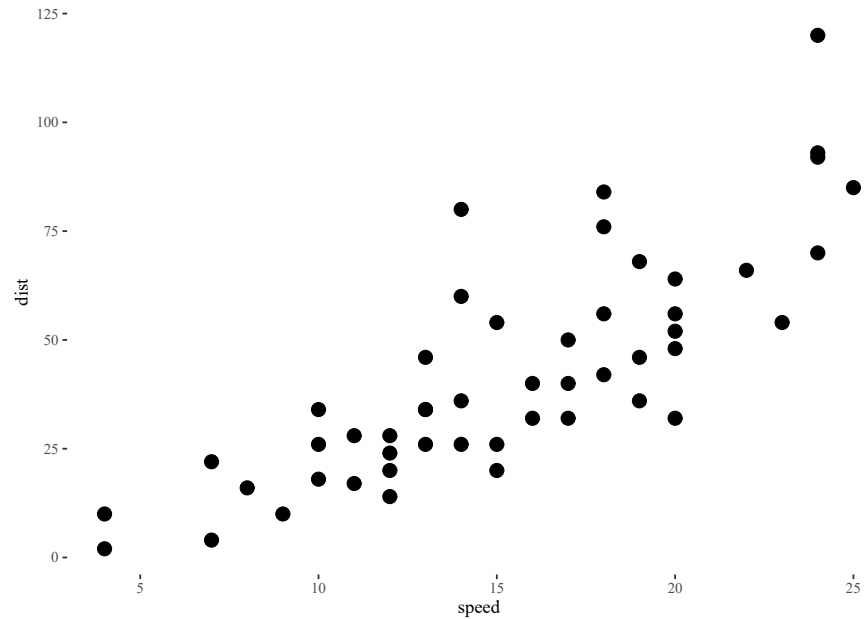


FIGURE 1.10: Scatterplot of the car dataset

- `kable()` 함수는 가장 단순한 형태의 표만 생성하기 때문에 복잡한 표를 만들기에는 한계가 존재함
- 이를 보완하기 위해 다음과 같은 패키지 활용
 - `kableExtra`: HTML 또는 LaTeX 용 표 생성
 - * https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html
 - * https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_pdf.pdf
 - `flextable` + `officer`: HTML, 워드 문서 표 작성
 - * <https://davidgohel.github.io/flextable/>

```
```${r}

knitr::kable(head(iris))

```
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

1.5 인라인 (inline) R 코드

- 문서의 모든 숫자를 인라인 R 코드를 통해 재현가능하게 생성 가능
- 인라인 R 코드는 ``r`` 과 ``` 사이에 변수 계산 스크립트를 입력해 작성 가능
- 예를 들어 ``r 10 + 4`` 는 14 출력
- 활용 예시

```
head(mtcars, 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 |

```
N <- nrow(mtcars)
```

`mtcars` 데이터셋에 포함된 자동차는 ‘r N ‘ 개다.

→

`mtcars` 데이터셋에 포함된 자동차는 32 개다.

1.6 YAML

- R Markdown 문서의 가장 처음에 정의하는 metadata
- .Rmd 파일을 .md 파일로 변환 후 최종 출력문서 생성 시 필요한 pandoc의 옵션을 설정하는 것과 같은 의미임
- 일반적으로 문서 형태 및 생성을 위해 사용하는 R package (예: bookdown, officedown, rticles 등)에 따라 YAML 구성요소가 달라짐

기본 문법

- `/#`: 주석 처리
- YAML 문서의 시작과 끝은 `---` 로 정의함
- 기본적으로 콜론(`:`)으로 구분된 태그(키): 값 쌍으로 구성됨 → `key: value`
 - 여기서 콜론 바로 다음에는 반드시 공백문자가 있어야 함
- 한 `key`의 하위 키는 리스트 형태로 표현하고, 하위 키는 두 개 이상의 스페이스로 공백을 주어 표현

```
---  
key : value  
  subkey1: value1  
  subkey2: value2  
    subsubkey1: value3  
---
```

R Markdown 기본 YAML 구조

```
---  
title: "문서 제목" # 일반적으로 따옴표 사용  
subtitle: "문서 부제목"  
author: "문서 작성자"  
date: "문서 작성일자"  
output:  
  - "html_document"  
  - "word_document"  
  - "pdf_document"  
  - "md_document"  
  - "isoslides_presentation"  
  - "slidy_presentation"  
  - "beamer_presentation"  
bibliography: 참고문헌.bib # bibtex 서식 활용  
.  
.  
.  
---
```

- <https://bookdown.org/yihui/rmarkdown/documents.html> 에
자세한 예시 참고

1.7 참고문헌 인용

- 참고문헌 정보가 BibTeX 포맷으로 저장된 .bib 파일을 YAML에 선언 후
인용 가능

Bibtex 참고문헌 입력 형태

```
@article{Shea2014,
  author = {Shea, Nicholas and Boldt, Annika},
  journal = {Trends in Cognitive Sciences},
  pages = {186--193},
  title = {{Supra-personal cognitive control}},
  volume = {18},
  year = {2014},
  doi = {10.1016/j.tics.2014.01.006},
}
```

YAML에 bib 파일 지정

```
---
title: "Citation test"
bibliography: example.bib
output: html_document
---
```

- 참 고 문 헌 표 현 : `[@citation-identifier]` 또 는
`@citation-identifier`

This...

Blah blah [@Shea2014; @Lottridge2012].

Shea et al. says blah [-@Shea2014].

@Shea2014 says blah.

Blah blah [see @Shea2014, pp. 33-35; also @Wu2016, ch. 1].

turns into this...

Blah blah (Shea et al. 2014; Lottridge et al. 2012).

Shea et al. says blah (2014).

Shea et al. (2014) says blah.

Blah blah (see Shea et al. 2014, 33-35; also Wu 2016, ch. 1).

- BibTeX 포맷은 Google Scholar 에서 쉽게 획득 가능
- Citation 스타일은 YAML 헤더에 `cl: style.cs1`로 변경 가능하며 Zotero⁶ 에서 .cs1 파일 다운로드 가능

⁶<https://www.zotero.org>



Bibliography

Peng, R. D. (2016). *R programming for data science*. Learnpub.

Rizzo, M. L. (2019). *Statistical computing with R*. CRC Press.

Silge, J. and Robinson, D. (2017). *Text mining with R*. " O'Reilly Media, Inc."

Wickham, H. and Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data*. " O'Reilly Media, Inc."

Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1138700109.

Xie, Y., Dervieux, C., and Riederer, E. (2020). *R Markdown Cookbook*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1000290806.

고석범 (2014). *R과 knitr를 활용한 데이터 연동형 문서 만들기*. 에이콘 출판사, 1st edition. ISBN 978-8960775510.