

R + Python + Latex in one editor

Preparation for literate programming

1	Preface	3
2	R, RStudio, Rtools의 설치	4
2.1	사용자 계정에 관련된 환경 설정	4
2.2	R 설치	6
2.3	R Studio 설치	6
2.4	Rtools 설치	6
2.5	R Studio의 실행 전 권한 셋팅	6
2.6	R Studio의 설치 확인	7
2.7	R Studio의 추가 설정	8
2.8	진짜 초보를 위한 R Studio 사용	9
3	R Markdown의 설치 및 사용법	11
3.1	Motivation	11
3.2	실습	11
3.3	Source (.rmd) vs Document (.html)	13
3.4	Documents	13
3.5	Slides	15
3.6	.Rmd 파일의 구성	18
3.7	Literature Programming	20
3.8	R Markdown Cheatsheet	21
4	R Studio에서 Python 사용하기	23
4.1	라이브러리 reticulate 설치	23
4.2	라이브러리 reticulate 설치 확인	23
4.3	Trouble-shooting	24
4.4	R에서의 python 객체 접근	25
4.5	(Optional) Tensorflow의 설치	26
5	TinyTex 설치 (pdf 조판)	27
5.1	Tinytex 설치	27
5.2	extrafont 패키지 설치	27
5.3	설치 테스트	27
5.4	폰트 설치 troubleshoot	27
5.5	일반 tex 파일 컴파일	28

5.6 주의 사항 및 기타	29
--------------------------	----

1 Preface

데이터과학자는 절대로 혼자 일하지 않는다. 직무가 어떤 것이던 간에 분석과 연구의 결과는 타인과 공유되어야만 가치를 창출한다. 그러므로 데이터과학자는 커뮤니케이션에 능해야 한다. 현업에 종사하는 경우에는 커뮤니케이션 능력이 커리어의 포텐셜을 결정하며, 연구에 종사하는 경우에도 마찬가지로 효과적인 Written Communication을 바탕으로 본인의 work를 타인에게 인정받을 수 있다.

Literate Programming이란 하나의 통합 환경에서 프로그래밍 언어를 돌리는 것 뿐만 아니라 Documentation을 동시에 하는 것을 의미한다. 이를 통해 아래와 같은 효과를 얻을 수 있다.

- Integrated Environment: 분석을 위한 S/W와 문서화를 위한 S/W가 하나로 통합된다.
- Automatic Documentation: 분석의 결과가 자동으로 문서에 반영되게 하여 자동으로 문서를 제작할 수 있으며, 정기적인 보고서 등을 자동으로 만들어 낼 수 있다.
- Reproducible Research: 하나의 환경에서 코드와 서술이 함께있기 때문에 주석을 달 필요가 줄어들며, 타인 혹은 미래의 자신이 review할 때에도 훨씬 효과적이다.

대표적인 Literate Programming의 장치로서 잘 알려진 것은 Python Notebook이다. 하지만, 현재 2021년에 기술적으로 더 잘 통합된 환경을 제공하는 것은 RStudio의 R Markdown이다. R과 Python은 모두 오픈소스이지만, RStudio는 사기업으로서 운영되기에 더 잘 통합된 환경을 제공하는데 일원화된 노력을 하는 것이 그 이유라 생각된다.

본 문서는 아래의 과정을 담고 있다.

1. R, R Studio, R Tools의 설치
2. 통합된 환경의 중심이 되는 R Markdown의 설치 및 사용법
3. R Studio에서 사용할 수 있도록 Python 설치
4. R Studio에서 사용할 수 있도록 Academic 조판 S/W인 TexLive의 설치

이를 통해 하나의 통합된 환경에서 R과 Python을 이용한 분석을 수행하고, 그 결과를 Technical Note, Presentation, Academic Paper 등의 포맷으로 생산해 낼 수 있다.

본 문서는 본인 심민규의 대학원 수업 Data Visualization 등에서 사용할 목적으로 작성하였습니다. 본 문서는 2023-05-05에 마지막으로 수정되었습니다. 사용하면서 에러가 발생하는 경우에는 mksim@seoultech.ac.kr으로 신고 부탁드립니다.

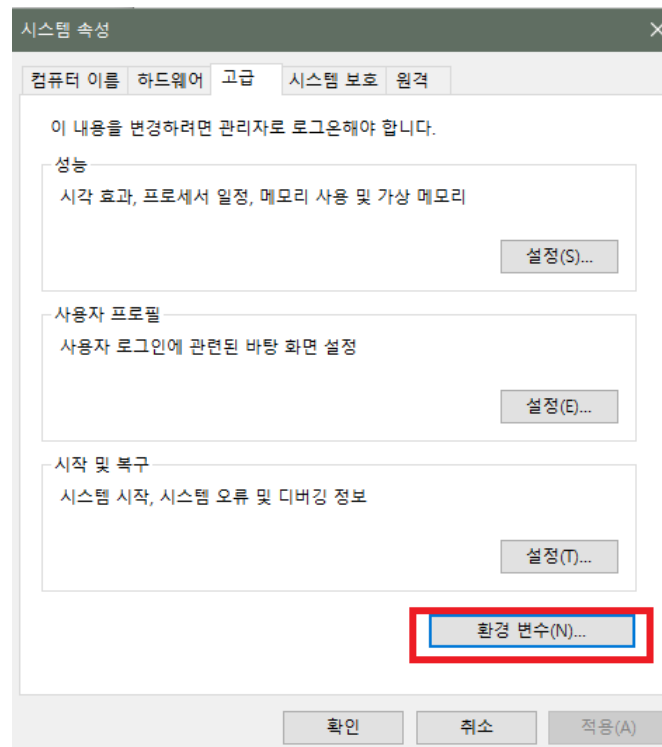
2 R, RStudio, Rtools의 설치

2.1 사용자 계정에 관련된 환경 설정

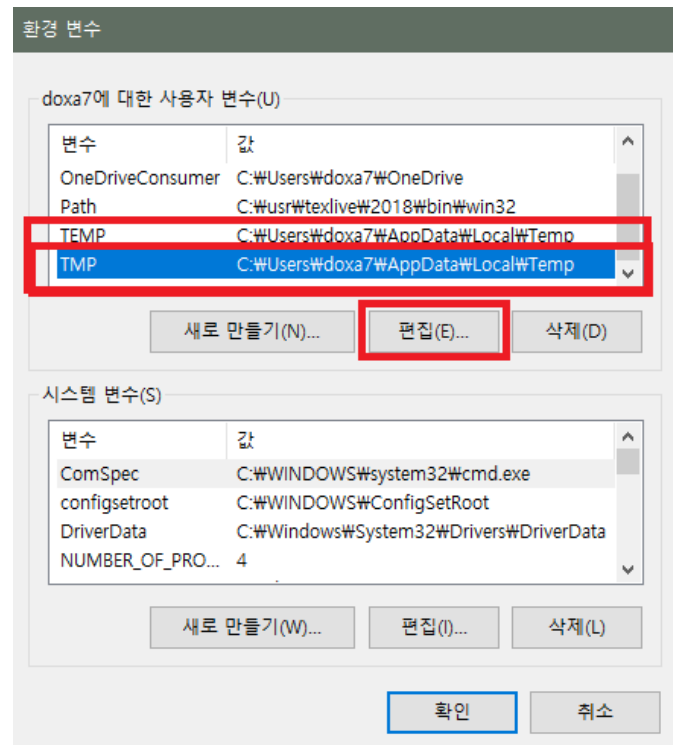
- R/R Studio의 경우에 많은 한글 관련 이슈가 해결되고 있는 추세이지만 아직 완벽하지 않다. 그러므로 윈도우 사용자 계정이 한글인 경우에는 아래의 과정을 거쳐서 각종 이슈를 미연에 방지해야 한다.
- 윈도우 사용자 계정이 영어로 되어 있는 경우라면 이번 단계를 skip해도 무방하다.
- 참고로 프로그래밍을 할 예정이라면 폴더이름/파일이름등을 빈칸없이 영어로 하는 습관이 좋다. 빈칸이 필요한 경우에는 south_korea과 같이 “underscore” 기호를 사용하면 된다.

만약에 현재 사용자 계정이 한글로 되어있다면 아래의 과정을 수행해야 한다.

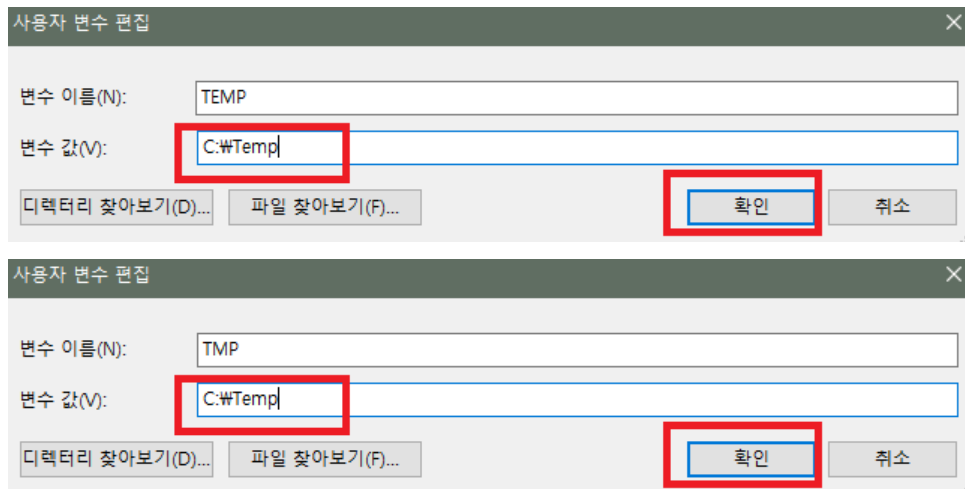
1. 제어판 → 시스템 및 보안 클릭 → 시스템 클릭 → 고급 시스템 속성 클릭 → 시스템 속성 창을 아래와 같이 띄운다. (윈도우 버전이나 시스템 설정에 따라서 과정이 다를 수는 있으나 어쨌든 아래 화면을 띄워야 함)



2. 환경변수를 클릭하고 아래의 TEMP와 TMP항목의 값이 영어로된 경로의 폴더를 가르키게 바꾸어야 한다. (어떤 폴더라도 상관없지만, 경로가 영어로 되어있어야 함)



3. 간단한 방법으로, 탐색기 프로그램을 이용해서 c:\Temp 폴더를 생성하고, 위의 그림에서 편집을 선택하여 TEMP와 TMP 항목의 값에 아래와 같이 모두 c:\Temp를 입력해 주면 된다.



이로서 사용자 계정이 한글인 경우에 관련된 환경 설정을 마친다.

2.2 R 설치

1. google에서 “R download” 검색
2. OS에 맞는 최신 버전 설치

2.3 R Studio 설치

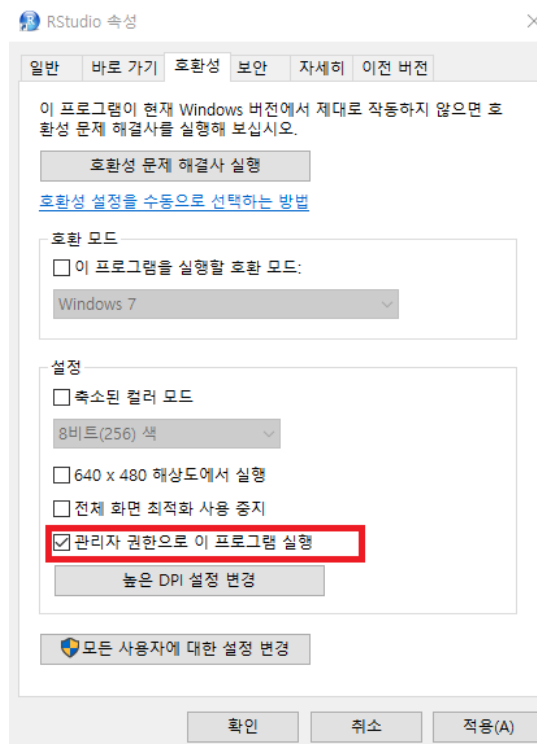
1. (R의 설치가 완전히 완료된 이후에 설치하는 것이 좋음)
2. 이 문서는 2023-05-05에 마지막으로 업데이트 되었고, 현재 R Studio의 최신버전은 1.4이다. 그런데 파이썬 사용과 관련해서 1.4 버전에 문제가 있는 것으로 보인다. 따라서 현재로서는 1.3 버전 설치를 권장한다.
3. google에서 “R Studio previous version” 검색
4. 마찬가지로 OS에 맞게 다운로드후 설치

2.4 Rtools 설치

1. goole에서 “R Studio download” 검색
2. 마찬가지로 OS에 맞는 최신 버전 설치

2.5 R Studio의 실행 전 권한 셋팅

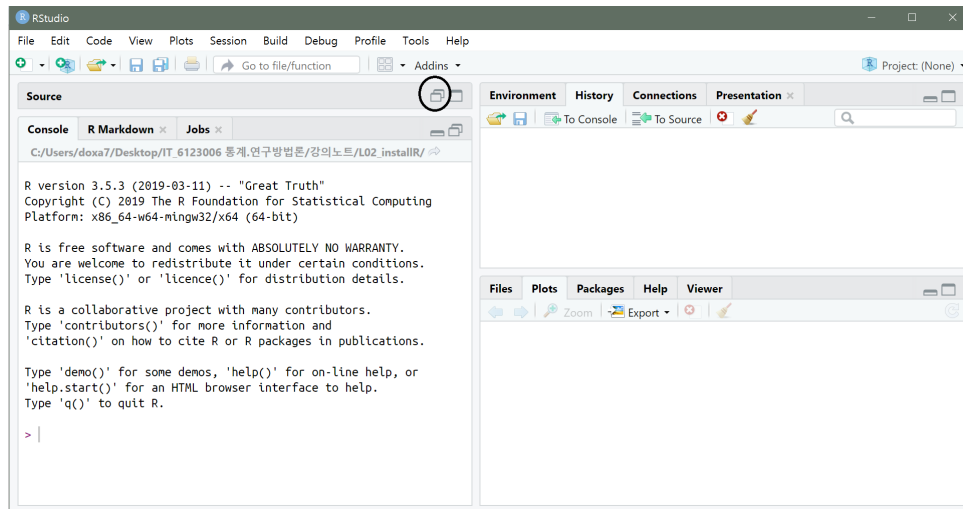
- 바탕화면에 R Studio 아이콘이 생성되지 않았다면, 시작버튼을 눌러서 R Studio 아이콘을 찾음
- 바탕화면에 바로가기를 만들거나 작업표시줄에 추가하는 것을 추천
- R Studio 아이콘을 우클릭 → 호환성 탭 → 관리자 권한으로 이 프로그램 실행을 체크



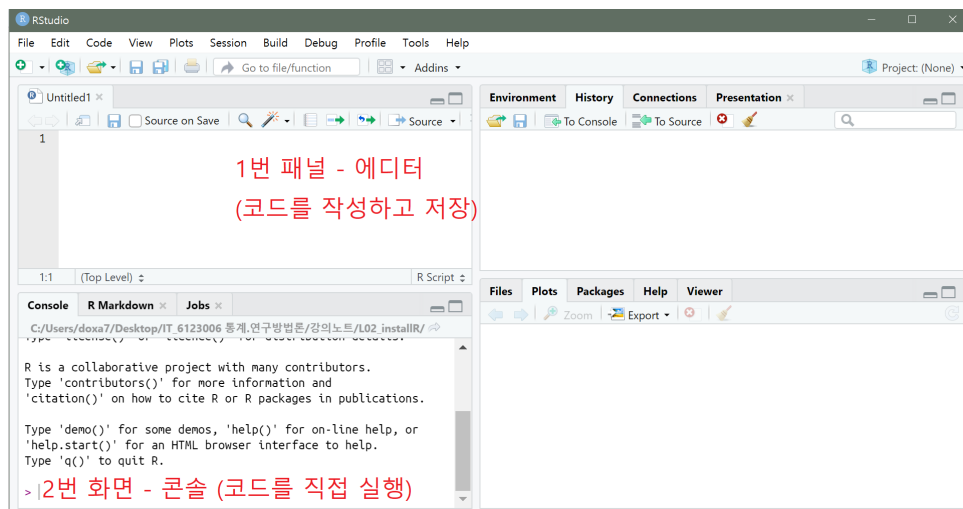
2.6 R Studio의 설치 확인

2.6.1 실행

- 아이콘을 더블클릭하여 실행하면 아래와 같은 화면이 나옵니다.



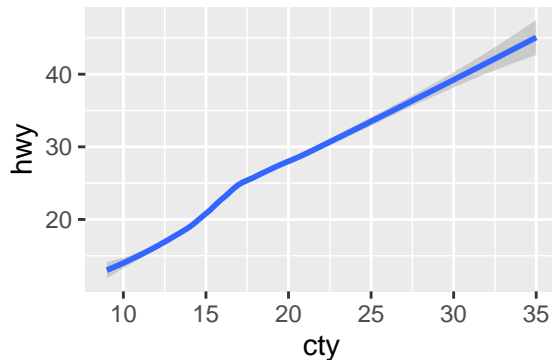
- 위 화면에서 검정색 동그라미로 표시된 버튼을 누르면 아래와 같이 4분할된 화면이 등장합니다.



2.6.2 설치 테스트

- 2번 화면에 프롬프트 표시(>)가 보입니다.
- 여기에 아래의 명령어를 차례로 입력하여 아래와 같은 그림이 나오는지 확인하세요.
- 그림이 잘 나온다면 설치가 완료된 것입니다.

```
install.packages("ggplot2")
library(ggplot2)
ggplot(mpg, aes(cty, hwy)) + geom_smooth()
```



2.7 R Studio의 추가 설정

- 한글 관련 이슈를 방지하기 위해서 아래의 설정을 합니다.
 - Tool → Global Option → CODE → Saving → Default Text Encoding... → UTF-8
 - 향후에 소스파일에서 한글이 깨질때에는
 1. File → Reopen with Encoding → UTF-8
 2. 그래도 안되면 UTF-8이 아닌 CP-949 선택
 3. 그래도 안되면 Rstudio를 닫았다가 재실행해서 1과 2를 다시 시도
 4. 그래도 안되면 질문!
- 편안한 작업을 위해서 폰트와 색상을 변경합니다.
 - Tool → Global Option → Appearance
 - 내가 일반적으로 사용하는 설정
 - * Editor Font Size: 14
 - * Editor Theme: "Tomorrow Midnight Blue"

2.8 진짜 초보를 위한 R Studio 사용

2.8.1 코드 입력

- 콘솔 패널에 아래와 같이 입력하세요.

```
"Hello"
a <- 10
a <- a + 5
a
```

- 아래의 결과가 나오시나요?

```
## [1] 15
```

- 우상단 패널의 History 탭을 누르면 지금까지 실행한 명령이 보입니다.
- Shift키와 Ctrl키를 이용해서 이들을 선택하고 To Editor 버튼을 눌러서 에디터 패널로 옮기세요.
- File → Save As버튼을 눌러서 (단축키 Ctrl + S) 파일을 first.R로 저장하세요.
- R Studio를 종료하고 윈도우 탐색기에서 first.R를 더블클릭하면 R Studio가 실행됩니다.
- (아이콘이 아니라 소스 파일을 더블클릭해도 R Studio를 실행할 수 있습니다. - 엑셀처럼)

2.8.2 몇 개의 단축키

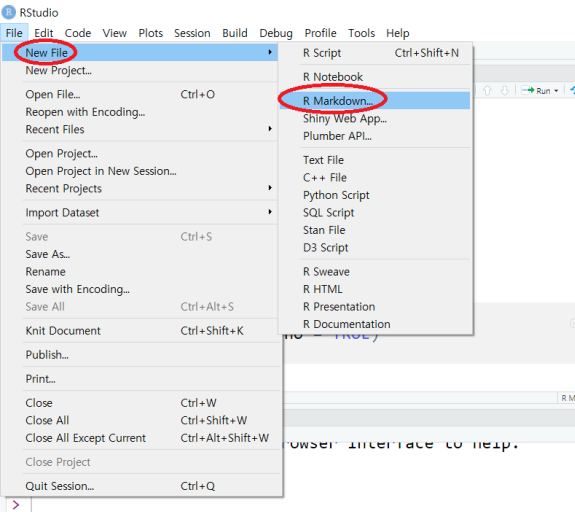
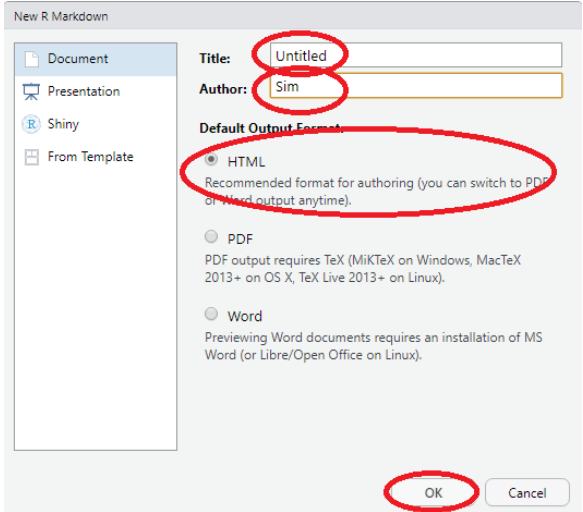
- 1번 화면으로 커서를 이동하려면 Ctrl + 1
- 2번 화면으로 커서를 이동하려면 Ctrl + 2
- 1번 화면을 전체화면으로 만들려면 Ctrl + Shift + 1
- 2번 화면을 전체화면으로 만들려면 Ctrl + Shift + 2
- 1번 화면에 있는 코드를 2번 화면으로 Copy & Paste 하지 않고 바로 실행하려면 Ctrl + Enter

3 R Markdown의 설치 및 사용법

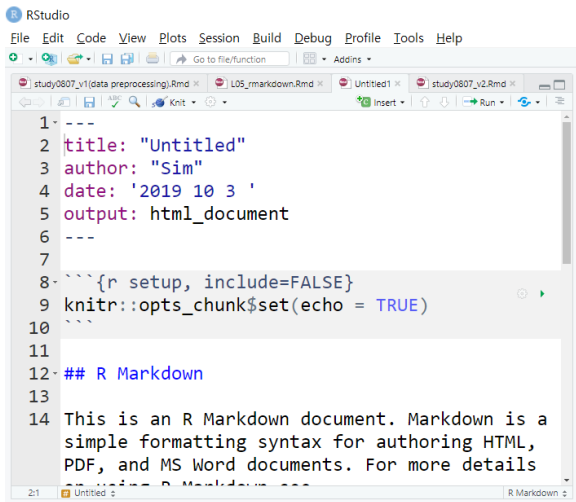
3.1 Motivation

- Workflow
 - Old: 계산 → 표, 그림 정리 → 워드프로세서에 정리 → 다시 계산 → ...
 - New: rmarkdown in R Studio
- Tools
 - Old: 에디터, 콘솔, 아웃풋 윈도우, 엑셀, 워드프로세서
 - New: rmarkdown in R Studio

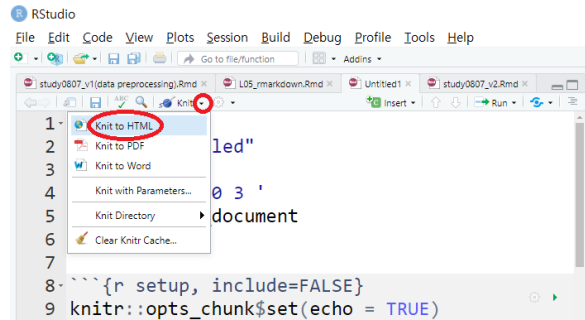
3.2 실습

Step 1	Step 2
<p>File → New File → R Markdown</p> 	<p>Document → Title과 Author 입력 → html</p> 
<p>Step 3</p> <p>템플릿 소스 코드 등장!</p>	<p>Step 4</p> <p>Knit → Knit to html</p>

Step 3

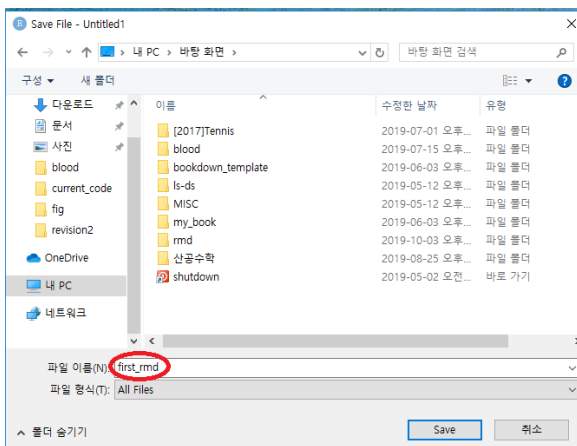


Step 4



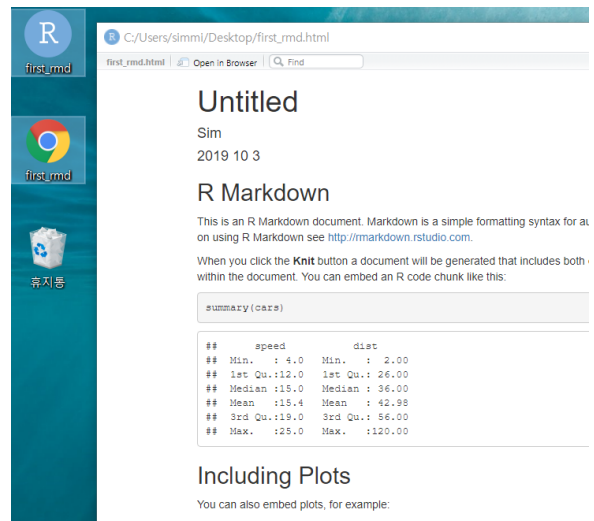
Step 5

파일 이름 지정



Step 6

rmd 파일과 그 결과로 html 파일이 생성됨!



3.3 Source (.rmd) vs Document (.html)

The first screenshot shows the RStudio Source Editor with the following code and annotations:

```

1 ---
2 title: "Untitled"
3 author: "Sim"
4 date: '2019 10 3'
5 output: html_document
6 ---
7
8 {r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10
11 ## R Markdown
12
13 This is an R Markdown document. Markdown is a
14 simple formatting syntax for authoring HTML,
15 PDF, and MS Word documents. For more details
16 on using R Markdown see
17 <http://rmarkdown.rstudio.com>.
18
19 When you click the Knit button a document
20 will be generated that includes both content
21 as well as the output of any embedded R code
22 chunks within the document. You can embed an R
23 code chunk like this:
24
25 {r cars}
26 summary(cars)
27
28 
```

Annotations for the first screenshot:

- 1. 제목, 저자, 날짜
- 2. 섹션의 제목을 정할 때에는 #기호를 사용함. 두번을 쓴 경우를 "level 2 헤더"라고 함
- 3. <>기호를 이용해 하이퍼링크 생성
- 4. **안쪽은 굵은 글씨
- 5. R chunk 부분 cars는 해당 r chunk의 이름. 코드와 결과가 같이 문서에 등장함. R chunk를 시작할 때에는 백틱 (') 3회와 종괄호 안에 r을 넣고, 종료할 때에는 백틱 (')를 다시 3회 적어줌

The second screenshot shows the rendered HTML document with the following content and annotations:

```

<h1>Untitled</h1>
<p>Sim</p>
<p>2019 10 3</p>
<h2>R Markdown</h2>
<p>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 <a href="http://rmarkdown.rstudio.com">http://rmarkdown.rstudio.com</a>.</p>
<p>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:</p>
<pre>summary(cars)</pre>
<table>
<tr>
<th>##</th>
<th>speed</th>
<th>dist</th>
</tr>
<tr>
<td>##</td>
<td>Min.</td>
<td>4.0</td>
<td>Min.</td>
<td>2.00</td>
</tr>
<tr>
<td>##</td>
<td>1st Qu.</td>
<td>12.0</td>
<td>1st Qu.</td>
<td>26.00</td>
</tr>
<tr>
<td>##</td>
<td>Median</td>
<td>15.0</td>
<td>Median</td>
<td>36.00</td>
</tr>
<tr>
<td>##</td>
<td>Mean</td>
<td>15.4</td>
<td>Mean</td>
<td>42.98</td>
</tr>
<tr>
<td>##</td>
<td>3rd Qu.</td>
<td>19.0</td>
<td>3rd Qu.</td>
<td>56.00</td>
</tr>
</table>

```

Annotations for the second screenshot:

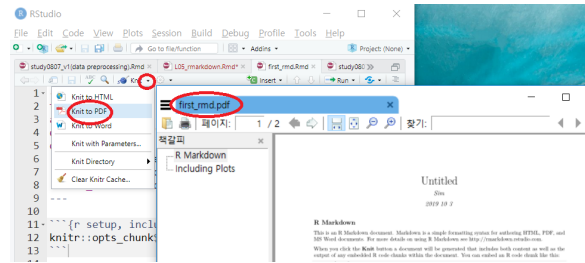
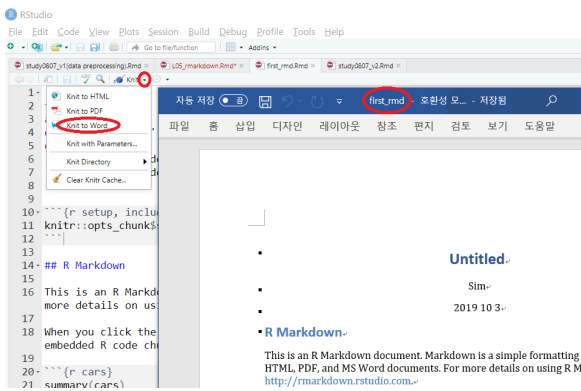
- 6. r chunk의 시작에 echo=FALSE를 넣어주면 문서에 코드가 보이지 않음
- 7. R chunk의 코드 결과가 그림이므로 문서에 그림이 자동으로 삽입됨
- 8. 백틱 (')으로 감싸준 글자는 typesetting font (monospace)로 바뀌어서 코드임을 구별할 수 있게 해줌

3.4 Documents

- Knit 버튼에서 포맷 선택 가능
 - html: 가장 빠르게 컴파일 됨
 - docx: MS Office 프로그램이 설치되어 있지 않아도 사용이 가능함
 - pdf: texlive라는 프로그램이 설치되어 있어야만 가능하며, 설치에 관해서는 본 문서의 iv. TexLive의 설치를 참조

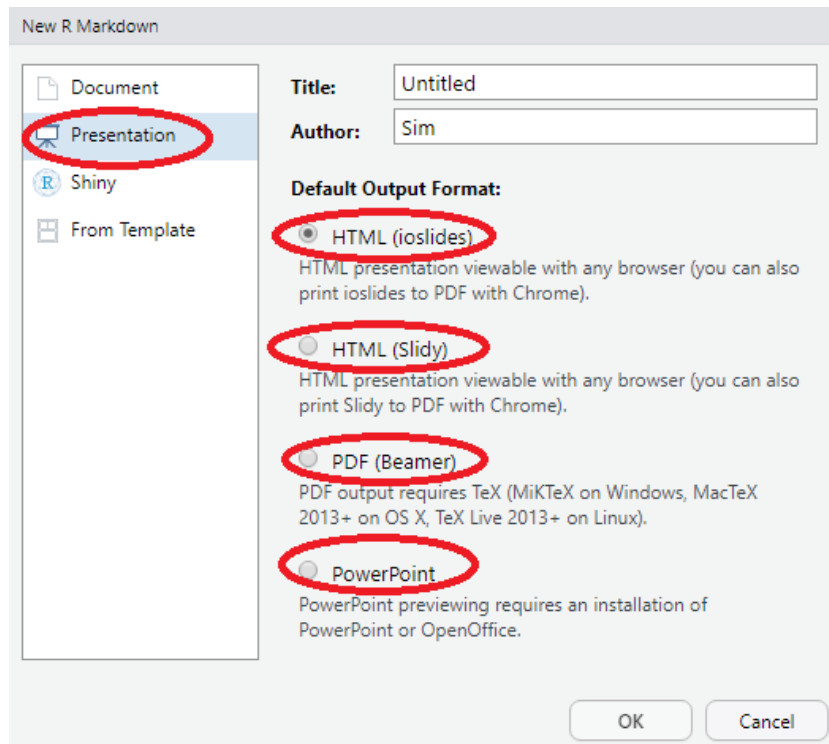
docx

pdf



3.5 Slides

- 위의 첫 번째 html 예제의 Step 1 단계에서 Document가 아닌 Presentation을 선택
- Level-II Header (##)로 page를 구분



- io_slides와 slidy
 - File → New File → R Markdown → Presentation → io_slides나 slidy로 시작하면 됨
 - html파일 형식의 슬라이드, 모바일 기기 등에서 접근하기 편한 슬라이드를 만들 수 있다.
- pptx
 - File → New File → R Markdown → Presentation → power point로 시작하면 됨
- beamer
 - pdf 파일이므로 texlive 프로그램이 필요하다.
 - 본인 심민규 교수의 강의노트 포맷
 - 수식의 표현이 용이해서 학회발표/강의노트 제작에 유용
 - 다음 링크에서 템플릿을 받아서 사용하는 것을 권장 <https://github.com/aceMKSIm/templates/tree/master/rmarkdown/beamer>

io_slides

Untitled

Sim
2019 10 3

R Markdown

This is an R Markdown presentation. 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.

2/3

Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

3/3

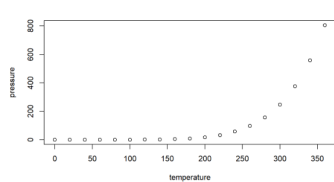
Slide with R Output

```
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
```

4/3

Slide with Plot



5/3

slidy

Untitled

Sim
2019 10 3

R Markdown

This is an R Markdown presentation. 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.

slide 1/3

Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

slide 2/3

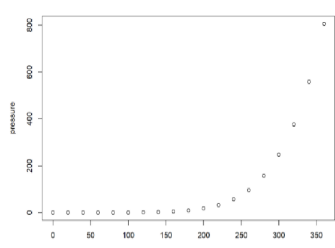
Slide with R Output

```
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
```

slide 3/3

Slide with Plot



slide 3/3

beamer (pdf)	pptx																																										
	<p>1</p> 																																										
<p>R Markdown</p> <p>This is an R Markdown presentation. 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.</p> <p>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.</p>	<p>2</p> <p>R Markdown</p> <p>This is an R Markdown presentation. 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.</p> <p>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.</p>																																										
<p>Slide with Bullets</p> <ul style="list-style-type: none"> Bullet 1 Bullet 2 Bullet 3 	<p>3</p> <p>Slide with Bullets</p> <ul style="list-style-type: none"> Bullet 1 Bullet 2 Bullet 3 																																										
<p>Slide with R Output</p> <pre>##> summary(mtcars)</pre> <table border="1"> <thead> <tr> <th></th> <th>mpg</th> <th>wt</th> </tr> </thead> <tbody> <tr> <td>## Min.</td> <td>10.4</td> <td>1.615</td> </tr> <tr> <td>## 1st Qu.</td> <td>15.4</td> <td>2.620</td> </tr> <tr> <td>## Median</td> <td>19.2</td> <td>3.440</td> </tr> <tr> <td>## Mean</td> <td>20.09</td> <td>3.217</td> </tr> <tr> <td>## 3rd Qu.</td> <td>22.8</td> <td>3.570</td> </tr> <tr> <td>## Max.</td> <td>33.4</td> <td>5.263</td> </tr> </tbody> </table>		mpg	wt	## Min.	10.4	1.615	## 1st Qu.	15.4	2.620	## Median	19.2	3.440	## Mean	20.09	3.217	## 3rd Qu.	22.8	3.570	## Max.	33.4	5.263	<p>4</p> <p>Slide with R Output</p> <pre>##> summary(mtcars)</pre> <table border="1"> <thead> <tr> <th></th> <th>mpg</th> <th>wt</th> </tr> </thead> <tbody> <tr> <td>## Min.</td> <td>10.4</td> <td>1.615</td> </tr> <tr> <td>## 1st Qu.</td> <td>15.4</td> <td>2.620</td> </tr> <tr> <td>## Median</td> <td>19.2</td> <td>3.440</td> </tr> <tr> <td>## Mean</td> <td>20.09</td> <td>3.217</td> </tr> <tr> <td>## 3rd Qu.</td> <td>22.8</td> <td>3.570</td> </tr> <tr> <td>## Max.</td> <td>33.4</td> <td>5.263</td> </tr> </tbody> </table>		mpg	wt	## Min.	10.4	1.615	## 1st Qu.	15.4	2.620	## Median	19.2	3.440	## Mean	20.09	3.217	## 3rd Qu.	22.8	3.570	## Max.	33.4	5.263
	mpg	wt																																									
## Min.	10.4	1.615																																									
## 1st Qu.	15.4	2.620																																									
## Median	19.2	3.440																																									
## Mean	20.09	3.217																																									
## 3rd Qu.	22.8	3.570																																									
## Max.	33.4	5.263																																									
	mpg	wt																																									
## Min.	10.4	1.615																																									
## 1st Qu.	15.4	2.620																																									
## Median	19.2	3.440																																									
## Mean	20.09	3.217																																									
## 3rd Qu.	22.8	3.570																																									
## Max.	33.4	5.263																																									
<p>Slide with Plot</p> 	<p>5</p> <p>Slide with Plot</p> 																																										

3.6 .Rmd 파일의 구성

.Rmd 파일은 아래의 3가지 부분으로 구성되어 있다.

- 포맷과 메타정보를 담는 Yaml Header
- R, Python, SQL 등을 실행할 수 있는 Code Chunk
- 보고서의 서술을 위해서 Markdown언어를 사용하는 MD Chunk

3.6.1 Yaml Header

- 코드의 맨 앞에 ---로 둘러싸인 영역
- title, author, date 등을 입력할 수 있다.
- 파일 포맷 지정
 - 다양한 파일 포맷 및 옵션 지정 가능
 - 하위 항목은 빈칸 두 번으로 표시함

Documents	Slides	pptx
<pre> 1 --- 2 title: "Untitled" 3 author: "Sim" 4 date: '2019 10 3 ' 5 output: 6 pdf_document: default 7 html_document: default 8 word_document: default 9 --- </pre>	<pre> 1 --- 2 title: "Untitled" 3 author: "Sim" 4 date: '2019 10 3 ' 5 output: 6 beamer_presentation: default 7 ioslides_presentation: default 8 slidy_presentation: default 9 --- </pre>	<pre> 1 --- 2 title: "Untitled" 3 author: "Sim" 4 date: '2019 10 3 ' 5 output: powerpoint_presentation 6 --- </pre>

3.6.2 R chunks

- R 언어로 작성하며 Color로도 구분됨
- R Chunk를 만들 때에는
 - 백틱을 3회하고 중괄호 안에 r을 넣어줌
 - pressure는 해당 R chunk의 고유한 이름이지만, 없어도 상관없음
 - echo=FALSE는 코드를 보이지 않게 한다는 것이며, 다른 여러 종류의 chunk option이 존재함 (마지막 섹션의 Cheatsheet 참조)
- 오른쪽에 3개의 버튼이 있다.
 - Setting
 - Run all chunks above
 - Run this chunk

```

25 ## Including Plots
26
27 You can also embed plots, for example:
28
29 ```{r pressure, echo=FALSE}
30 plot(pressure)
31 ```
32
33 Note that the `echo = FALSE` parameter was
    added to the code chunk to prevent printing of
    the R code that generated the plot.
  
```

3.6.3 Markdown chunks

- 사람의 언어 입력
- Inline R code를 사용할 수 있으며, 이것이 automatic documentation의 핵심요소라 할 수 있다.

The screenshot shows the RStudio interface. The left pane displays a Markdown document with the following content:

```

20
21 ```{r cars}
22 summary(cars)
23 ```
24
25 + 조사된 차종의 평균 최대 속력은 mean(cars$speed)이다.
26
  
```

The right pane shows the output of the R code chunk, which is the summary of the 'cars' dataset:

```

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
  
```

Below the output, a bullet point states: "조사된 차종의 평균 최대 속력은 15.4이다." (The average maximum speed of the surveyed car models is 15.4).

3.7 Literature Programming

Markdown은 HTML(HyperText Markup Language) 보다 간결한 Markup 언어로서 github 문서 등 기술 문서 작성의 대세라고 할 수 있다. R Markdown에서는 아래와 같은 문서 변환 과정을 거친다. Yaml 헤더의 약간의 수정만으로 여러가지 형식의 문서를 만들 수 있다.

1. 일반적인 경우: Rmd \rightarrow md \rightarrow html, docx, pptx
2. 최종 파일이 pdf인 경우: Rmd \rightarrow md \rightarrow tex \rightarrow pdf

Literate Programming에서는 프로그래밍 보다 글쓰기가 초점이 된다. 글을 쓰는 것과 프로그래밍을 하는 인간으로서 가장 수준 높은 지적인 활동을 한꺼번에 할 수 있는 환경을 제공한다. 코드에 주석을 조금만 달아도 되고 하나의 파일로 관리할 수 있다. 코드와 보고서가 어울려져서 의사소통이 용이하다. 본인은 모든 반복적인 엑셀 행정 업무를 하는 회사원에게 이를 가르쳐야 한다고 생각한다.

You don't know what you will can get away with until you try – Colin Powell

3.8 R Markdown Cheatsheet



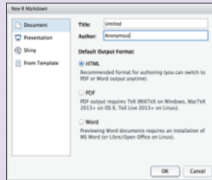
1. 작업흐름 R 마크다운은 R로 재현가능하고, 동적인 보고서를 작성하는 서식이다. R 마크다운을 사용해서 R 코드와 실행결과를 발표자료, pdf, html, 워드 문서 등에 삽입할 수 있다. 보고서를 작성하려면:

- i. 파일열기 - Rmd 확장자를 갖는 텍스트 파일로 저장해서 시작하거나, Studio Rmd 템플릿을 열어 시작한다.
- ii. 작성하기 - 문서를 작성하기 쉬운 R 마크다운 구문을 사용해서 작성한다.
- iii. 내장하기 - 리포트에 포함될 출력 결과를 생성하는 R 코드를 내장한다.
- iv. 렌더링(Render) - R 코드를 출력형식으로 치환하고 보고서를 발표자료, pdf, html, MS 워드 파일 형식으로 변환한다.



2. 파일 열기 .Rmd 확장자를 갖는 텍스트 파일로 저장해서 시작하거나, Studio Rmd 템플릿을 열어 시작한다.

- 메뉴막대에서, 다음순으로 클릭한다.
File ▶ New File ▶ R Markdown...
- 원도우가 열리면, .Rmd 파일로 작성하려는 출력유형을 선택한다.
- 라디오 버튼으로 출력형식을 선택한다 (나중에 출력형식은 변경할 수 있다)
- OK 버튼을 클릭한다.



4. 출력형식 선정 R 마크다운 파일에서 생성할 문서 유형을 기술하는 YAML 헤더정보를 작성한다.

YAML

YAML 헤더는 키(key) 집합; 파일 시작지점에 나오는 키-값 쌍, 헤더 시작과 끝은 3개 대쉬를 갖는 라인(---)

```

---
title: "xwMOOC 보고서"
author: "정민석"
output: html_document
---
  
```

RStudio 템플릿이 자동으로 YAML 헤더 정보를 작성해 넣는다.

출력값이 .Rmd 파일에서 어떤 형식 파일을 생성할 것인지 결정한다(단계 6)


output: html_document html 파일 (웹페이지)
output: pdf_document pdf 문서
output: word_document MS 워드문서 .docx
output: beamer_presentation Beamer 발표자료 (pdf)
output: ioslides_presentation 맥 발표자료 (html)

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • joh@rstudio.com • 844-449-1212 • rstudio.com
한국 번역: 이광준 ig@xwmooc.net xwMOOC 홈페이지: <http://www.xwmooc.net>, CC BY-SA xwMOOC

3. 마크다운 다음으로, 일반 텍스트로 보고서를 작성한다. 마크다운 구문을 사용해서 최종 보고서에 적용할 텍스트 서식을 기술한다.

입력 구문

일반 텍스트
새로운 단락을 시작하려면 줄 마지막을 공백 2개로 끝낸다.
기울임 글씨* and 기울임 글씨, *굵은 글씨* and *굵은 글씨*
원형사각?
~ 취소선 ~
[링크](www.rstudio.com)

제목 1
제목 2
제목 3
제목 4
제목 5
제목 6
N자크기 대쉬 부호: ---
n자크기 대쉬 부호: ---
상위:
중시 처리하는 수식: $A = \frac{1}{2} \pi r^2$
이미지: 
수생선 (혹은 슬라이드 번호):

...

> 인용 블록

* 순서없는 목록
* 항목 2
* 상위 항목 1
* 상위 항목 2


1. 순서있는 목록
2. 항목 2
* 상위 항목 1
* 상위 항목 2

표 제목 | 두번째 제목

표 1 | 칸 2
칸 3 | 칸 4
칸 3 | 칸 4

출력 결과

일반 텍스트
새로운 단락을 시작하려면 줄 마지막을 공백 2개로 끝낸다.
기울임 글씨* and 기울임 글씨, *굵은 글씨* and *굵은 글씨*
원형사각?
~ 취소선 ~
[링크](www.rstudio.com)

제목 1
제목 2
제목 3
제목 4
제목 5
제목 6
N자크기 대쉬 부호: ---
n자크기 대쉬 부호: ---
상위: ...
중시 처리하는 수식: $A = \frac{1}{2} \pi r^2$
이미지: 
수생선 (혹은 슬라이드 번호):

...

> 인용 블록

* 순서없는 목록
* 항목 2
* 상위 항목 1
* 상위 항목 2

1. 순서있는 목록
2. 항목 2
* 상위 항목 1
* 상위 항목 2

표 제목 | 두번째 제목

표 1 | 칸 2
칸 3 | 칸 4

5. 코드내장하기 knitr 구문을 사용해서 R 코드를 보고서에 내장한다. R이 코드를 실행하고, 보고서를 렌더링할 때 결과를 포함시킨다.

인라인 코드

R 코드를 백틱(`)으로 감싼다. R이 인라인 코드를 실행된 결과로 대체한다.

```
2. 터키가 2는 `r 2`  
2와 같다.
```

Two plus two
equals 4.

코드 덩어리

R코드 덩어리를 ```` (4) 으로
사작하고, ```` 으로 마무리한다.

```
실행결과를 다음과 같다  
````  
dim(iris)
```

```
Here's some code
dim(iris)
[1] 150 5
```

#### 화면 출력 선택옵션

knitr 선택옵션을 사용해서 코드 덩어리 출력 스타일을 적용한다.  
코드 상단 괄호 내부에 선택옵션을 지정한다.

```
Here's some code
````{r eval=FALSE}  
dim(iris)
```

```
Here's some code  
dim(iris)
```

```
Here's some code  
````{r echo=FALSE}  
dim(iris)
```

```
Here's some code
[1] 150 5
```

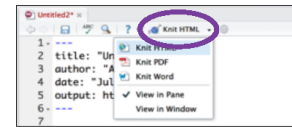
| 선택옵션       | 기본설정     | 효과                               |
|------------|----------|----------------------------------|
| eval       | TRUE     | 코드를 평가하고 실행결과를 포함한다.             |
| echo       | TRUE     | 실행결과와 함께 코드를 출력한다.               |
| warning    | TRUE     | 경고메시지를 출력한다.                     |
| error      | FALSE    | 오류메시지를 출력한다.                     |
| message    | TRUE     | 메시지를 출력한다.                       |
| tidy       | FALSE    | 깔끔한 방식으로 코드 형태를 변형한다.            |
| results    | "markup" | "markup", "asis", "hold", "hide" |
| cache      | FALSE    | 결과값을 캐쉬해서 향후 실행시 건너뛰게 설정한다.      |
| comment    | "##"     | 주석문자로 출력결과에 서두를 붙인다.             |
| fig.width  | 7        | 덩어리로 생성되는 그래프에 대한 폭을 인치로 지정한다.   |
| fig.height | 7        | 덩어리로 생성되는 그래프에 대한 높이를 인치로 지정한다.  |

보다 자세한 사항은 웹사이트를 참조: [yihui.name/knitr/](http://yihui.name/knitr/)

**6. 렌더링** 최종보고서를 생성하는데 .Rmd 파일을 사용하여 청사진을 제작한다.

두가지 방식으로 보고서를 렌더링한다.

1. `rmarkdown::render("<파일 경로>")` 명령어를 실행한다.
2. RStudio 스크립트 작성창 상단에 **knit HTML** 버튼을 클릭한다.



렌더링 명령을 실행시키면, R은 다음을 수행한다.

- 내장된 코드 덩어리를 각각 실행시키고, 실행결과를 보고서에 삽입한다.
- 출력 파일형식에 맞춰 신규 보고서를 생성한다.
- 미리보기로 뷰어창에 출력파일을 연다.
- 작업디렉토리에 출력파일을 저장한다.

**7. 인터랙티브 문서** 작성한 보고서를 3단계를 거쳐 인터랙티브 Shiny 문서변환.

1. YAML 헤더에 `runtime: shiny` 을  
추가한다.

```

title: "Line graph"
output: html_document
runtime: shiny

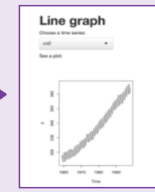
```

2. 코드 덩어리에, 위젯을 내장하는  
Shiny input 함수를 추가한다. Shiny  
render 함수를 추가해서 반응형 출력  
결과를 내장한다.

```
Choose a time series:
selectInput("data", "",
 c("ca2", "lh7"))

See a plot:
{r echo = FALSE}
renderPlot({
 d <- get(input$data)
 plot(d)
})
```

3. `rmarkdown::run` 명령어  
로 렌더링하거나 RStudio  
Run Document 버튼을  
클릭한다.



\* 주목: 보고서는 Shiny 앱이 된다. 따라서, (인터랙티브 보고서를 위해) `html_document` 혹은 (인터랙티브 발표자료) `ioslides.presentation` 출력형식을 선택한다.

**8. Publish** 온라인으로 접속하는 사용자와 보고서를 공유한다.

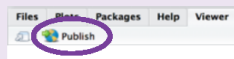
#### Rpubs.com

RStudio 무료 R 마크다운 게시 사  
이트를 통해 정적 문서를 공유한다.  
[www.rpubs.com](http://www.rpubs.com)

#### ShinyApps.io

Studio 서버에 인터랙티브 문서  
를 올려 호스팅한다. 무료와 유료  
선택옵션이 있다.  
[www.shinyapps.io](http://www.shinyapps.io)

RStudio 미리보기 창에 "Publish" 버튼을 클  
여, rpubs.com 사이트에 버튼 한번 클릭으로  
바로 올린다.



**9. 추가 학습**

문서와 예제 - [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)  
추가 기사 - [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)  
- [blog.rstudio.com](http://blog.rstudio.com)  
- @rstudio



RStudio® and Shiny™ are trademarks of RStudio, Inc.  
CC BY RStudio info@rstudio.com  
844-448-1212 rstudio.com

## 4 R Studio에서 Python 사용하기

### 4.1 라이브러리 reticulate 설치

앞의 R Studio 설치 부분에서 언급한 바와 같이, 이 문서는 2023-05-05에 마지막으로 업데이트 되었고, 현재 R Studio의 최신버전은 1.4이다. 그런데 파이썬 사용과 관련해서 1.4 버전에 문제가 있는 것으로 보인다. 따라서 현재로서는 “R Studio previous version”로 검색하여 1.3 버전 설치를 권장한다. 그리고 R Tools도 설치되어 있어야 한다.

R Studio에서 python을 사용하는 것은 R의 라이브러리인 `reticulate`를 사용한다. 다음의 명령을 실행하면 `reticulate`를 설치한다.

```
install.packages("reticulate")
```

### 4.2 라이브러리 reticulate 설치 확인

설치를 확인하기 위해서

1. 먼저 `library(reticulate)`를 실행한다.
2. 아래 그림처럼 R Markdown에서 python 코드 블록을 만들어서 입력을 하고 실행을 해서 정상적으로 출력이 되는지 확인한다.



### 4.3 Trouble-shooting

앞의 스텝에서 정상적으로 출력이 되지 않고 R Studio가 Abort되는 현상이 종종 생긴다. 이 경우에 대한 트러블 슈팅 과정을 설명한다.

reticulate 패키지는 R의 r-miniconda라는 디렉토리와 r-reticulate라는 environment를 생성하고, 해당 python을 사용해야지 정상적으로 R Studio에서 작업을 할 수 있다. 그런데 때로는 r-reticulate가 아닌, 컴퓨터에 미리 인스톨이 되어있는 Anaconda로 사용하게 되어 있는 경우에 해당 에러가 발생한다.

우선 아래의 코드 블록을 실행한다. 결과로서 시스템 내에서 default로 사용하기로 되어있는 python 경로가 보인다.

```
Sys.which("python")
```

```
python
"C:\\Users\\kenjin\\AppData\\Local\\R-MINI~1\\envs\\R-RETI~1\\python.exe"
```

혹은 아래의 코드 블록을 실행해본다. 결과로서 reticulate 라이브러리에서 사용하기로 설정되어 있는 python의 경로가 보인다.

```
library(reticulate)
py_config()
```

```
python: C:/Users/kenjin/AppData/Local/r-miniconda/envs/r-reticulate/python.exe
libpython: C:/Users/kenjin/AppData/Local/r-miniconda/envs/r-reticulate/python38.dll
pythonhome: C:/Users/kenjin/AppData/Local/r-miniconda/envs/r-reticulate
version: 3.8.16 | packaged by conda-forge | (default, Feb 1 2023, 15:53:35) [MSC v.1929 64 bit (AMD64)]
Architecture: 64bit
numpy: C:/Users/kenjin/AppData/Local/r-miniconda/envs/r-reticulate/Lib/site-packages/numpy
numpy_version: 1.24.3
##
NOTE: Python version was forced by RETICULATE_PYTHON_FALLBACK
```

위의 경로가 /r-miniconda/envs/r-reticulate로 끝나지 않는 경우에 에러가 발생하는 것으로 보인다. 이를 고쳐주기 위해서는 경로를 정확하게 찾아서 예를 들어 아래와 같이 입력한다.

```
use_python("C:/Users/MinKyuSim/AppData/Local/r-miniconda/envs/r-reticulate")
```

그리고 나서 다시 아래의 그림과 같이 python이 정상작동하는지 확인한다. 만약에 이렇게 해도 안된다면 연락주세요. 교수님과 함께 해결책을 찾아봅시다.



#### 4.4 R에서의 python 객체 접근

R Studio에서 python 코드를 사용할 수 있을 뿐만 아니라, python에서 생성된 객체를 R로 불러올 수도 있다. 아래의 그림을 확인하라. 라이브러리 `reticulate`에 대해서 더 자세히 알고 싶으면 다음 링크를 참조하라. <https://rstudio.github.io/reticulate/>

```

434 + Step 1: R block에서 `reticulate`를 로드한다.
435
436 ▾ ```{r}
437 library(reticulate)
438 ▸ ```
439
440 + Step 2: Python block에서 데이터를 만들었다고 가정한다면...
441
442 ▾ ```{python}
443 import numpy as np
444 a = np.zeros(3)
445 a
446 ▸ ```

```

array([0., 0., 0.])

```

447
448 + Step 3: R block에서 `py$` 뒤에 python 객체 이름을 넣으면 python
 객체에 접근 가능하다.
449
450 ▾ ```{r}
451 py$a
452 ▸ ```

```

[1] 0 0 0

## 4.5 (Optional) Tensorflow의 설치

Tensorflow를 R Studio에서 실행할 수 있다. 그러나 속도가 조금 더 느리다고 알려져 있기에 간단한 프로젝트가 아니라면 python에서 하는것을 추천한다. 다만, R Markdown 환경과의 연동 등의 목적으로 R에서 작업하고 싶을 수도 있을 것이다. 혹은 python tensorflow에서 작성된 brain file을 R에서 시각화 하고자 하는 경우에도 유용하다.

- Step 1. Windows의 경우라면 Anaconda의 설치가 필요하다. 모든 프로그램을 닫고, Anaconda를 설치하고, 그리고 나서 재부팅을 한 번 한 후에 다시 이곳으로 돌아온다.
- Step 2. 아래의 코드를 실행한다.

```
install.packages("tensorflow")
```

- Step 3. 아래의 코드를 실행한다.

```
library(tensorflow)
install_tensorflow()
```

- Step 4. 위의 Step 1에서 설치한 Anaconda3의 경로를 아래 처럼 입력해준다.

```
library(reticulate)
use_python("E:/ProgramData/Anaconda3")
```

- Step 5. 아래의 코드를 입력해서 결과가 정상적으로 나오는지 확인한다. 정상적으로 나온다면 설치에 성공한 것이다.

```
library(tensorflow)
tf$constant("Hello Tensorflow")
```

- Step 6. 위의 Step 5에서 에러가 나는 경우에는 tensorflow가 제대로 설치되지 않았을 가능성이 높다. 이 때에는 아래의 코드를 실행한다. 그리고 나서 Step 3부터 Step 5까지 다시 해본다.

```
reticulate::py_install("tensorflow")
```

만약에 이렇게 해도 안된다면 연락주세요. 교수님과 함께 해결책을 찾아봅시다.

## 5 TinyTex 설치 (pdf 조판)

### 5.1 Tinytex 설치

- 먼저 R과 R Studio를 최신 버전(2023년 이후)으로 업데이트를 한다.
- 혹시 컴퓨터에 texlive 라는 프로그램을 설치되어 있다면 이를 깨끗이 삭제하자.
- 다음의 명령을 RStudio 콘솔에서 실행하면 된다.

```
install.packages('tinytex')
tinytex::install_tinytex()
```

이제 필요한 패키지를 RStudio 가 자동으로 설치해 줄 것이다. 시스템에 따라 다르지만 설치는 5분 이내에 완료될 것이다.

### 5.2 extrafont 패키지 설치

영어나 한글의 예쁜 글꼴을 사용하기 위해서 extrafont라는 R 패키지를 설치해야 한다. 이는 일반 텍스트 뿐만 아니라 그래프 등의 제목이나 축에도 사용된다.

```
install.packages('extrafont')
library(extrafont)
fonts()
```

RStudio 는 현재 시스템에 설치되어 있는 폰트를 RStudio 에 등록하는 작업을 자동으로 수행한다.

위 명령의 마지막 줄 fonts() 는 현재 설치되어 있는 폰트들을 보여준다. 이제 제대로 작동하는지 확인하기 위하여 인기 있는 한글 폰트인 네이버 나눔글꼴을 다음의 링크를 통해 설치해보자

<https://hangeul.naver.com/share>

### 5.3 설치 테스트

아래의 링크를 방문하여 다운로드하고 압축을 푼다.

[https://github.com/aceMKSim/templates/blob/master/rmarkdown/pdf/pdf\\_template.zip](https://github.com/aceMKSim/templates/blob/master/rmarkdown/pdf/pdf_template.zip)

그리고 pdf\_template.Rmd을 RStudio에서 열어 Knit → Knit to PDF 버튼을 누른다. pdf 파일이 정상적으로 생성된다면 설치에 성공한 것이다. 만약에 폰트가 없다는 에러가 뜬다면 앞의 Step에서 나눔폰트가 정상적으로 설치되지 않은 것이다. 이 경우는 아래의 내용을 수행한다.

### 5.4 폰트 설치 troubleshoot

이상의 작업에서 대부분은 별다른 문제를 일으키지 않는다. 하지만 일부 시스템에서 다소 귀찮은 문제가 생기는 경우가 있는데, 네이버 나눔글꼴을 설치할 때 “C:/Windows/fonts” 가 아닌 “C:/Users/여러분의사용자계정명/AppData/Local/Microsoft/Windows/Fonts” 에 설치하는 것이 그것이다. 다른 프로그램에서는 사용자계정 폴더에 설치해도 문제가 없으나 R에서는 경로를 인식하지 못하고 에러를 낸다.

먼저 해당 문제가 발생했는지를 확인하기 위하여 다음의 명령을 콘솔에 실행하여 폰트가 제 위치에 설치되어 있는지를 확인한다.

```
library(extrafont)
fonttable()
```

나눔글꼴이 제대로 설치되어 있다면 아래와 같이 “C:/Windows/fonts/” 폴더 아래에 폰트들이 설치되어 있는 모습을 볼 수 있을 것이다.

```
1 library(extrafont)
2 fonttable()
3 package afmfile fontfile
4 1 NA NanumGothic.afm.gz C:\\Windows\\Fonts\\NanumGothic.ttf
5 2 NA NanumGothicBold.afm.gz C:\\Windows\\Fonts\\NanumGothicBold.ttf
6 3 NA NanumGothicExtraBold.afm.gz C:\\Windows\\Fonts\\NanumGothicExtraBold.ttf
7 4 NA NanumGothicLight.afm.gz C:\\Windows\\Fonts\\NanumGothicLight.ttf
8 5 NA NanumBarunGothic.afm.gz C:\\Windows\\Fonts\\NanumBarunGothic.ttf
9 6 NA NanumBarunGothicBold.afm.gz C:\\Windows\\Fonts\\NanumBarunGothicBold.ttf
10 7 NA NanumBarunGothicLight.afm.gz C:\\Windows\\Fonts\\NanumBarunGothicLight.ttf
11 8 NA NanumBarunGothicUltraLight.afm.gz C:\\Windows\\Fonts\\NanumBarunGothicUltraLight.ttf
```

폰트가 “C:/Windows/fonts/”가 아닌 사용자계정 폴더에 설치된 것으로 나타난다면 우선 기존의 폰트 데이터 베이스를 다음의 명령으로 제거한다. 설치 명령이지만 폰트 데이터 베이스를 삭제하는 역할을 우선 수행하므로 폰트를 재설치해야 할 때 실행할 수 있다.

```
install.packages('extrafontdb')
```

이후 다음의 링크에서 나눔글꼴 전체를 내려 받은 후 압축을 풀고 필요한 글꼴을 “C:/Windows/fonts” 폴더에 붙여 넣는다.

<https://hangeul.naver.com/font>

이제 RStudio 를 재실행하고 knit 버튼을 누르면 정상적으로 pdf 파일을 생성할 것이다.

## 5.5 일반 tex 파일 컴파일

확장자가 .Rmd 인 파일 뿐만 아니라 .tex 인 파일을 pdf 파일로 컴파일 해야 할 경우 다음의 스텝을 따른다.

Tex 명령 tlmgr 명령을 cmd 명령 프롬프트에서 실행할 수 있도록 경로를 설정하기 위하여 다음의 명령을 콘솔에 입력한다.

```
tinytex::tlmgr_path()
```

이제 윈도우 시작버튼을 누르고 명령프롬프트 cmd 를 실행한후 다음과 같이 tlmgr 명령을 이용해 tex 저장소를 설정한다.

```
C:\> tlmgr option repository http://mirror.kakao.com/CTAN/systems/texlive/tlnet
```

또는 다음과 같이 네이버 저장소를 이용하는 것도 가능하다.

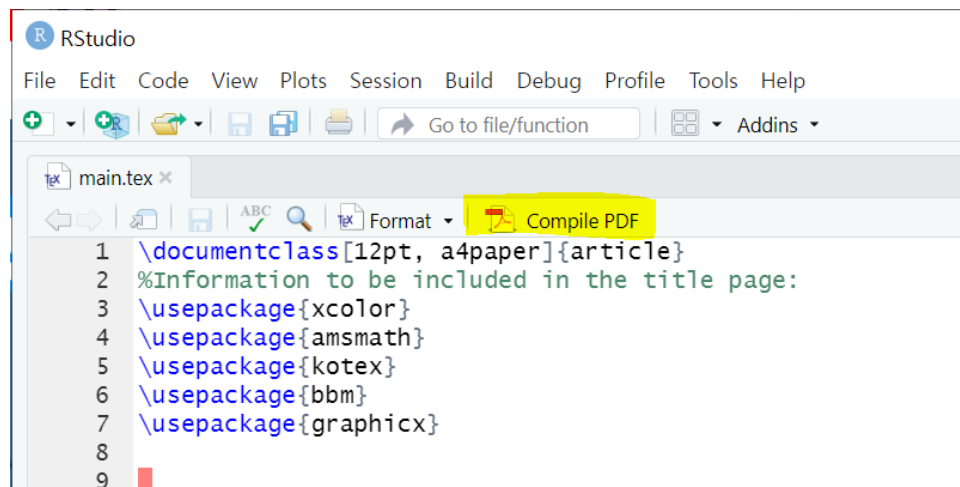
```
C:\> tlmgr option repository http://mirror.navercorp.com/CTAN/systems/texlive/tlnet/
```

마지막으로 다음의 명령으로 tex 을 전체설치할 수 있다.

```
C:\> tlmgr install scheme-full
```

위의 작업은 시스템에 따라 다르나 보통 20-30분 정도 걸리므로 여유를 갖고 기다려야 한다.

이제 Rstudio 를 재실행하고 여러분이 작성한 tex 파일을 열면 다음과 같은 compile pdf 버튼이 생기는 것을 확인할 수 있다.



## 5.6 주의 사항 및 기타

- .Rmd를 이용해서 pdf 문서를 만드는 경우에는 반드시 코드 블록이 1개 이상 포함되어야 한다. 예를 들어서 이 문서의 맨 마지막에 있는 아인슈타인 quote가 포함된 코드블록을 제거하면 컴파일이 되지 않는다.
- 다음의 링크에 데이터시각화 과목에서 다루는 Rmarkdown에 관련된 전반적인 사용법이 강의노트 3번과 4번에 포함되어 있다. 상당 내용은 이 문서와 중복된다. <https://github.com/aceMKSIm/teaching/tree/master/Data%20Visualization/Lecture%20Notes>
- 다음의 링크에는 Latex를 이용한 몇 가지 pdf 파일 포맷 템플릿이 포함되어 있다. 본인이 직접 한글화 작업을 수행하였기에 github에 공유한다. <https://github.com/aceMKSIm/templates/tree/master/rmarkdown>
  - 1. beamer: 슬라이드 형태로써 발표나 강의슬라이드 제작에 사용된다.
  - 2. pdf: 일반적인 pdf 문서 형태이다. 앞에서 설치확인을 위해서도 사용했었다.
  - 3. taylor: Taylor & Francis라는 논문 출판사의 템플릿이다.
  - 4. tufte: Tufte라는 사람은 데이터시각화와 관련해서 꽤 입지가 있는 사람이다. 이 사람이 제안한 Technical note 형식이다. 가독성이 매우 좋다.

"I never teach my pupils, I only provide the conditions in which they can learn - Albert Einstein"