

# Carseat Sales

LearningSpoonsR

2018-07-01

## Contents

- Part 1. 수집 - from ISLR package
- Part 2. 전처리 - dplyr
- Part 3. 분석 & 시각화 - ggplot2
- Part 4. 문서화 - rmarkdown

- 간단한 프로젝트를 진행하면서 R의 기능과 관련 문법을 알아봅니다.
- ISLR은 Introduction to Statistical Learning in R이라는 도서에서 제공하는 패키지입니다.
- dplyr은 R에서 데이터 사이언스의 근간이 되는 `data.frame`을 효율적으로 다루는 패키지입니다.
- rmarkdown은 R의 결과물을 바로 문서시키는 패키지입니다.

Elements in  
Stat Learning

## Part 1. 수집 - from "ISLR" package

## 패키지 (package, 응용프로그램, library)

- R을 처음에 설치하면 `base`라는 패키지가 인스톨 되어있음.
- `base`외의 확장 기능을 제공하는 패키지를 설치하여 사용하게 됨.
- R은 오픈 소스 기반 언어로써, 누구나 패키지를 만들고 R프로그램의 발전에 기여할 수 있음.
- R과 Python과 같은 오픈소스 언어에서는 많은 사람들이 많은 패키지를 만들고 많은 사람들이 사용하면서 발전이 일어남.
- 본 수업에서는 R에서 데이터 사이언스의 과정을 support하는 강력한 패키지 위주로 진행.
- 각각의 패키지는 특정 데이터나 함수를 제공함.

- base외의 패키지를 사용하려면 먼저 설치를 해주어야 함.
- 따옴표를 넣어서 문자열로 입력

```
install.packages("packageName")
```

- 설치이후에는 패키지 사용 전에 아래와 같이 패키지 사용을 선언해주어야 함.
- 따옴표 없이 입력

```
library(packageName)
```

## 파일 (files)

### 1. 소스 파일

- 명령어와 주석을 기록해둔 파일
- 한번에 실행이 가능한 단위로 구성
- 저장/불러오기/편집 등이 가능함 (Rstudio에서 마치 오피스 처럼 다룰 수 있음)
- .r : R 소스 파일
- .Rmd : R Markdown 소스파일
- 서식(폰트의 종류나 크기)이 없는 파일이기 때문에 메모장과 같은 텍스트 에디터에서도 저장/불러오기/편집이 가능함.

## 2. 데이터 파일

- 데이터가 기록된 파일
- .csv: 콤마로 구분된 파일 (Comma Separated Values)
- .txt: 더 일반적인 텍스트 파일. \t(Tab)이나 \n(줄바꿈) 등의 기호로 구분되어 있음
- .xls, .xlsx: 엑셀파일. 서식 등이 포함되어 있는 경우에는 [다른 이름으로 저장 - csv로 저장]하여 csv 파일로 저장하면 용량이 줄어들어서 다루기 좋은 경우가 많음. 바로 불러오기도 특화 패키지를 이용해서 가능함.

## 3. R 데이터 파일

- R에서 작업중인 데이터를 저장하고 불러오는 데 유용한 파일.
- .Rdata: R 작업시에 현재 메모리의 상태를 저장하고 나중에 불러올 수 있어서 작업을 이어서 하는데 유용
- .Rda: Data Structure 개체 (object) 1<sup>||</sup>를 저장하는 파일

## 데이터 파일 불러오기/저장하기

- 파일을 불러오면 R에서는 기본적으로 `data.frame`으로 받아들임

### 1. csv파일

- 데이터 프레임을 생성하는 명령이기에 `stringsAsFactors = FALSE`의 옵션의 사용을 추천!
- 만약에 데이터 파일의 첫 번째 line이 제목인 경우 이를 column name으로 만들 수 있음.  
(이게 default임, 즉 header=TRUE를 입력하지 않더라도 `header=TRUE`로 동작함)

```
dataset <- read.csv("fileName.csv", header = TRUE, stringsAsFactors = FALSE)
```

- 만약에 첫 번째 line이 제목이 아니라면 `header=FALSE`를 사용해야 겠죠?

```
dataset <- read.csv("fileName.csv", header = FALSE, stringsAsFactors = FALSE)
```

- `data.frame`을 파일로 저장할 때에는 아래의 명령으로…

```
write.csv(dataset, "fileName.csv", header=TRUE)  
write.csv(dataset, "fileName.csv", header=FALSE)
```

## 2. txt파일

- csv파일과 매우 유사함. csv파일은 comma로 나누어져 있기에 별도의 입력이 필요하지 않지만, txt파일에서는 무엇으로 나누어져있는가를 입력해주어야 함. 이를 separator(sep)라고 함.
- `read.csv` → `read.table ("filename.txt", header = F, sep = "\t")`
- `write.csv` → `write.table`      *sep = "\n")*

## 3. xls, xlsx파일

```
library(readxl)
dataset <- read_excel("fileName.xlsx")
dataset <- read_excel("fileName.xlsx", col_names = FALSE)
# (same as `header=FALSE`)
```

## 4. R 데이터 파일

- 데이터셋 한개

```
save(dataset, "fileName.rda")
load("fileName.rda")
```

- 현재 메모리의 모든 객체 (변수와 함수)

```
save("fileName.rda") Printed
load("fileName.rda")  
Printed
```

## 경로 (디렉토리, directory)

- Rstudio를 어떻게 실행시켰냐에 따라서 현재 작업 디렉토리가 설정됨.
- Rstudio 아이콘을 더블클릭하여 실행하였을 때에는 시스템의 기본 설정대로 현재 작업 디렉토리가 됨.  
 이미 존재하는 .R이나 .Rmd 파일을 더블클릭하여 Rstudio를 실행했을 때에는 해당 파일의 경로가 현재 작업 디렉토리가 됨.
- 현재 작업 디렉토리를 확인하려면 getwd()를 실행.
- 작업 디렉토리를 바꾸려면 setwd("C:/LS-R")을 실행.
- 디렉토리의 입력은 "C:/LS-R", "C://LS-R", "C:\\LS-R"의 방식이 가능함.

### Working Directory

#### getwd()

Find the current working directory (where inputs are found and outputs are sent).

#### setwd('C://file/path')

Change the current working directory.

**Use projects In RStudio to set the working directory to the folder you are working in.**

- 작업 디렉토리

- 현재 작업 디렉토리를 조회  
(파일 저장시 여기에 저장됨)

- 작업 디렉토리를 설정

- Rstudio에서는 프로젝트 단위로의 설정도 제공  
(진행하는 작업이 많아지면 유용함)

## 도시별 카시트 세일즈 데이터 불러오기 (ISLR 패키지)

```
install.package("ISLR")
library(ISLR)
class(Carseats)

## [1] "data.frame"
head(Carseats)

##   Sales CompPrice Income Advertising Population Price ShelveLoc Age
## 1  9.50      138     73          11       276    120      Bad  42
## 2 11.22      111     48          16       260     83      Good  65
## 3 10.06      113     35          10       269     80  Medium  59
## 4  7.40      117    100           4       466     97  Medium  55
## 5  4.15      141     64           3       340    128      Bad  38
## 6 10.81      124    113           13       501     72      Bad  78
##   Education Urban US
## 1            17 Yes Yes
## 2            10 Yes Yes
## 3            12 Yes Yes
## 4            14 Yes Yes
## 5            13 Yes  No
## 6            16 No Yes
```

```
tail(Carseats, 5)

##      Sales CompPrice Income Advertising Population Price ShelveLoc Age
## 396 12.57        138     108        17       203    128      Good  33
## 397 6.14         139      23         3        37    120     Medium 55
## 398 7.41         162      26        12       368    159     Medium 40
## 399 5.94         100      79         7       284     95      Bad   50
## 400 9.71         134      37         0        27    120      Good  49
##      Education Urban US
## 396          14 Yes Yes
## 397          11 No Yes
## 398          18 Yes Yes
## 399          12 Yes Yes
## 400          16 Yes Yes

View(Carseats)
dim(Carseats)

## [1] 400 11

str(Carseats)

## 'data.frame': 400 obs. of 11 variables:
## $ Sales : num 9.5 11.22 10.06 7.4 4.15 ...
## $ CompPrice : num 138 111 113 117 141 124 115 136 132 132 ...
## $ Income : num 73 48 35 100 64 113 105 81 110 113 ...
## $ Advertising: num 11 16 10 4 3 13 0 15 0 0 ...
## $ Population : num 276 260 269 466 340 501 45 425 108 131 ...
## $ Price : num 120 83 80 97 128 72 108 120 124 124 ...
## $ ShelveLoc : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...
## $ Age : num 42 65 59 55 38 78 71 67 76 76 ...
## $ Education : num 17 10 12 14 13 16 15 10 10 17 ...
## $ Urban : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
## $ US : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

```
summary(Carseats)
```

```
##      Sales      CompPrice      Income      Advertising
##  Min.   : 0.000  Min.   : 77  Min.   :21.00  Min.   : 0.000
##  1st Qu.: 5.390  1st Qu.:115  1st Qu.:42.75  1st Qu.: 0.000
##  Median : 7.490  Median :125  Median :69.00  Median : 5.000
##  Mean   : 7.496  Mean   :125  Mean   :68.66  Mean   : 6.635
##  3rd Qu.: 9.320  3rd Qu.:135  3rd Qu.:91.00  3rd Qu.:12.000
##  Max.   :16.270  Max.   :175  Max.   :120.00  Max.   :29.000
##      Population      Price      ShelveLoc      Age
##  Min.   : 10.0  Min.   :24.0  Bad    : 96  Min.   :25.00
##  1st Qu.:139.0  1st Qu.:100.0 Good   : 85  1st Qu.:39.75
##  Median :272.0  Median :117.0 Medium :219  Median :54.50
##  Mean   :264.8  Mean   :115.8          Mean   :53.32
##  3rd Qu.:398.5  3rd Qu.:131.0          3rd Qu.:66.00
##  Max.   :509.0  Max.   :191.0          Max.   :80.00
##      Education      Urban      US
##  Min.   :10.0  No   :118  No  :142
##  1st Qu.:12.0  Yes  :282  Yes :258
##  Median :14.0
##  Mean   :13.9
##  3rd Qu.:16.0
##  Max.   :18.0
```

## Part 2. 전처리 - "dplyr"

## dplyr?

### 1. 빠르고 직관적인 데이터를 다루는 패키지

- 가장 빠른 언어인 C를 기반으로 만들어서 빠름
- 데이터 처리에 있어서 가장 직관적인 SQL(Standard Query Language)과 유사하게 만들어져서 직관적임!
- 직관적이어서 코드 가독성도 높음
- ✓ • 그러나 base 명령어도 같이 알아 두면 타인의 코드 참조와 파이썬, SQL등 다른 언어를 배울 때 도움이 됨

### 2. Hadley Wickham

- Head Scientist, Rstudio
- 통계학 박사 후 R에서 다수의 사용하기 좋은 패키지 개발
- youtube.com에 keynote등 좋은 동영상 많아요
- 누나도 통계학 전공 교수
- ggplot2, dplyr, reshape2, stringr 등의 패키지를 묶어서 tidyverse라는 패키지를 만듬.

## Basic Manipulation

### 1. rename (이름 바꾸기)

```
1 | Carseats <- rename(Carseats, Sales = Revenue)  
2 | names(Carseats)[names(Carseats) == "Sales"] <- "Revenue"
```

deyr  
base

### 2. filter (관찰값 추출, Row 추출)

```
1 | temp <- filter(Carseats, Income > 100)  
2 | temp <- Carseats %>% filter(Income > 100)  
3 | temp <- Carseats[Carseats$Income > 100,]
```

```
temp <- filter(Carseats, Age >= 30 & Age < 40)  
temp <- Carseats %>% filter(Age >= 30 & Age < 40)  
temp <- Carseats[((Carseats$Age >= 30) & (Carseats$Age < 40)),]
```

2 : and  
| : or

### 3. select (변수 추출, Column 선택)

```
a { temp <- select(Carseats, Income, Population)
b temp <- Carseats %>% select(Income, Population)
b temp <- Carseats[,c("Income", "Population")]
```

### 4. arrange (정렬)

```
a { Carseats <- arrange(Carseats, Price)
b Carseats <- Carseats %>% arrange(Price)
b Carseats <- Carseats[order(Carseats$Price),]

a { Carseats <- arrange(Carseats, desc(Price))
b Carseats <- Carseats %>% arrange(desc(Price))
b Carseats <- Carseats[order(Carseats$Price, decreasing = TRUE),]
```

## 5. mutate (새로운 변수)

```
b Carseats$Revenue <- Carseats$Sales  
d } Carseats <- mutate(Carseats,  
d }     AdvPerCapita = Advertising/Population,  
d }     RevPerCapita = Revenue/Population)  
d } Carseats <- Carseats %>%  
d }     mutate(AdvPerCapita = Advertising/Population,  
d }             RevPerCapita = Revenue/Population)  
b } Carseats$AdvPerCapita <- Carseats$Advertising/Carseats$Population  
Carseats$RevPerCapita <- Carseats$Revenue/Carseats$Population  
  
d } Carseats <- mutate(Carseats, AgeClass = ifelse(Age>=60, "Silver", "non-Silver"))  
d } Carseats <- Carseats %>% mutate(AgeClass = ifelse(Age>=60, "Silver", "non-Silver"))  
d } Carseats$AgeClass <- ifelse(Carseats$Age >= 60, "Silver", "non-Silver")
```

## Carseats

- 해당 Carseat 회사의 주요 구매층은 아이를 낳아서 키울 나이인 30대 연령층입니다.
- 소득이 높으면서 도시의 평균 연령이 30대인 도시에 충분한 광고비를 지출하고 있는지 검토해보고 싶습니다.
- We all can think in what we can speak of!
- Think and speak in logical, sequential, (and more often, reverse-sequential), and codable ways!
- Learning how to program makes your brain more logical.

```

# successive treatments
focusCity <- Carseats %>%
  filter(Income > 100) %>%
  filter(Age >= 30 & Age < 40) %>%
  mutate(AdvPerCapita = Advertising/Population) %>%
  select(Sales, Income, Age, Population, Education, AdvPerCapita) %>%
  arrange(Sales)
print(focusCity)

```

	Sales	Income	Age	Population	Education	AdvPerCapita
## 1	5.04	114	34	298	16	0.00000000
## 2	5.32	116	39	170	16	0.00000000
## 3	6.80	117	38	337	10	0.01483680
## 4	7.49	119	35	178	13	0.03370787
## 5	7.67	117	36	400	10	0.02000000
## 6	8.55	111	36	480	16	0.04791667
## 7	8.97	107	33	144	13	0.00000000
## 8	9.03	102	35	123	16	0.10569106
## 9	9.39	118	32	445	15	0.03146067
## 10	9.58	104	37	353	17	0.06515581
## 11	10.36	105	34	428	12	0.04205607
## 12	10.59	120	30	262	10	0.05725191
## 13	12.57	108	33	203	14	0.08374384

- 도시의 평균 나이가 20대, 30대, 40대 이상인 경우에 Revenue에 차이가 있을까요?

```
Carseats %>%
  mutate(AgeClass =
    ifelse(Age < 30, "Twenties",
          ifelse(Age < 40, "Thirties", "FourtyAbove")))) %>%
  group_by(AgeClass) %>%
  summarise(avgRevenue = mean(Sales))

## # A tibble: 3 x 2
##   AgeClass     avgRevenue
##   <chr>         <dbl>
## 1 FourtyAbove   7.30
## 2 Thirties      8.26
## 3 Twenties      7.76
```

## Discussion

- 평균 나이에 따른 평균 Revenue의 차이는 무엇을 말해주나요?
- 평균 나이가 포함하지 못하고 있는 정보는 어떤 것이 있나요?
- 어떤 데이터가 있으면 더 좋을까요?
- 그 데이터가 있으면 어떻게 하실 건가요?

## Discussion

dplyr	base
<i>Everyday Language</i>	<i>Classic Programming language</i>
사용성	타 언어와 범용적
SQL	<i>Pandas package in Python</i>

- SQL (Standard Query Language)
  1. 대용량 데이터베이스와 통신하는 언어
  2. R에서도 `sqldf`라는 패키지를 사용해서 SQL명령어로 작업할 수 있음.
  3. R을 할 줄 아는 사람이 SQL을 배우는데 걸리는 시간 < 1 day  
(얇은 책 하나만 사서 보시면 됩니다.)
  4. 데이터를 보는 눈을 키워 줍니다.
- 데이터를 전처리하는 작업의 소요시간은 전체 프로젝트에서 80% 이상입니다.

# cheatsheet for dplyr (1)

## Data Transformation with dplyr :: CHEAT SHEET

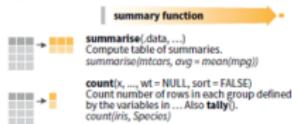


dplyr functions work with pipes and expect tidy data. In tidy data:



### Summarise Cases

These apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

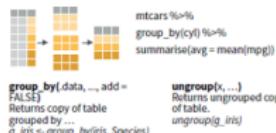


#### VARIATIONS

`summarise_all()` - Apply funs to every column.  
`summarise_at()` - Apply funs to specific columns.  
`summarise_if()` - Apply fns to all cols of one type.

### Group Cases

Use `group_by()` to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.



### Manipulate Cases

#### EXTRACT CASES

Row functions return a subset of rows as a new table.



#### Logical and boolean operators to use with filter()

<    <=    is.na()    %in%    |    &    xor()

See `?base::logic` and `?Comparison` for help.

#### ARRANGE CASES

`arrange(data, ...)` Order rows by values of a column or columns (low to high), use with `desc()` to order from high to low.  
`arrange(mtcars, mpg)`  
`arrange(mtcars, desc(mpg))`

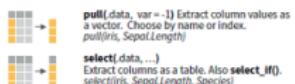
#### ADD CASES

`add_row(data, ..., before = NULL, after = NULL)`  
`add_one or more rows to a table.`  
`add_row_if(is, eruptions = 1, waiting = 1)`

### Manipulate Variables

#### EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

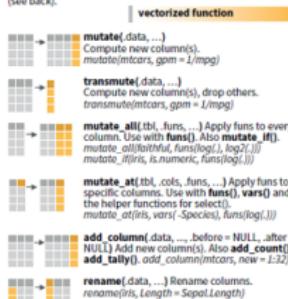


`use these helpers with select ...`  
e.g. `select_if(is, starts_with("Sepal"))`

`contains(match)`    `num_range(prefix, range)`    t, e.g. `mpg:cyl`  
`ends_with(match)`    `one_of(...)`    -, e.g. `-Species`  
`matches(match)`    `starts_with(match)`

#### MAKE NEW VARIABLES

These apply **vectorized functions** to columns. Vectorized fns take vectors as input and return vectors of the same length as output (see back).



RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-440-2121 • rstudio.com • Learn more with [browntroutnettes/package-c\("dplyr", "tidyverse"\)](#) • dplyr 0.7.0 • RStudio 1.2.0 • Updated: 2017-03

# Cheatsheet for dplyr (2)

## Vector Functions

### TO USE WITH MUTATE ()

**mutate()** and **transmute()** apply vectorized functions to columns to create new columns. Vectorized functions take vectors as input and return vectors of the same length as output.

### vectorized function

#### OFFSETS

dplyr::lag() - Offset elements by 1  
dplyr::lead() - Offset elements by -1

#### CUMULATIVE AGGREGATES

dplyr::cumall() - Cumulative all()  
dplyr::cumany() - Cumulative any()  
dplyr::cummax() - Cumulative max()  
dplyr::cummean() - Cumulative mean()  
dplyr::cummin() - Cumulative min()  
dplyr::cumprod() - Cumulative prod()  
dplyr::cumsum() - Cumulative sum()

#### RANKINGS

dplyr::cume\_dist() - Proportion of all values <=  
dplyr::dense\_rank() - rank with ties = min, no gaps  
dplyr::rank() - rank with ties = min  
dplyr::dense\_rank() - rank with ties = min  
dplyr::rank() - rank with ties = min  
dplyr::percent\_rank() - min rank scaled to [0,1]  
dplyr::row\_number() - rank with ties = "first"

#### MATH

\* - \* / ^ / ^ %/% %% - arithmetic ops  
log(), log2(), log10(), log() - logarithmic comparisons  
<->, >->, ><->, == - logical comparisons  
dplyr::between() - test if x is between right and left  
dplyr::near() - safe for floating point numbers

#### MISC

dplyr::case\_when() - multi-case if\_ else()  
dplyr::coalesce() - first non-NA values by element across a set of vectors  
dplyr::if\_(else()) - element-wise if() + else()  
dplyr::na\_if() - replace specific values with NA  
pmax() - element-wise max()  
pmin() - element-wise min()  
dplyr::model() - Vectorized switch()  
dplyr::recode\_factor() - Vectorized switch() for factors

## Summary Functions

### TO USE WITH SUMMARISE ()

**summarise()** applies summary functions to columns to create a new table. Summary functions take vectors as input and return single values as output.

### summary function

#### COUNTS

dplyr::n() - number of values/rows  
dplyr::n\_distinct() - # of uniques  
sum(is.na) - # of non-NA's

#### LOCATION

mean() - mean, also mean(is.na)  
median() - median

#### LOGICALS

mean() - proportion of TRUE's  
sum() - # of TRUE's

#### POSITION/ORDER

dplyr::first() - first value  
dplyr::last() - last value  
dplyr::nth() - value in nth location of vector

#### RANK

quantile() - nth quantile  
min() - minimum value  
max() - maximum value

#### SPREAD

IQR() - Inter-Quartile Range  
mad() - median absolute deviation  
sd() - standard deviation  
var() - variance

## Row Names

Tidy data does not use rownames, which store a variable outside of the columns. To work with the rownames, first move them into a column.

rownames\_to\_column() - Move row names into col.  
a <- rownames\_to\_column(mtcars, vor = "C")  
  
column\_to\_rownames() - Move col in row names.  
column\_to\_rownames(a, vor = "C")

Also has\_rownames(), remove\_rownames()

## Combine Tables

### COMBINE VARIABLES

x                    y  
 =                    

Use bind\_cols() to paste tables beside each other as they are.

bind\_cols(..., .) - Returns tables placed side by side as a single table.  
BE SURE THAT ROWS ARE THE SAME!

Use a "Mutating Join" to join one table to another. It creates a new column with the rows that don't correspond to. Each join retains a different combination of values from the tables.

left\_join(x, y, by = NULL,  
copy = FALSE, suffix = c("x", "y"), ...)  
Join matching values from y to x.

right\_join(x, y, by = NULL, copy =  
FALSE, suffix = c("x", "y"), ...)  
Join matching values from x to y.  
  
inner\_join(x, y, by = NULL, copy =  
FALSE, suffix = c("x", "y"), ...)  
Join data. Retain only rows with matches.

full\_join(x, y, by = NULL,  
copy = FALSE, suffix = c("x", "y"), ...)  
Join data. Retain all values, all rows.

Use by = c("col1", "col2") to specify the column(s) to match on.  
left\_join(x, y, by = "A")

Use a named vector. by = c("col1" =  
"col2"), to match on columns with different names in each data set.  
left\_join(x, y, by = c("C" = "D"))

Use suffix to specify suffix to give to duplicate column names.  
left\_join(x, y, by = c("C" = "D"), suffix =  
c("1", "2"))

Use a "Filtering Join" to filter one table against the rows of another.

semi\_join(x, y, by = NULL, ...)  
Return rows of x that have a match in y.  
USEFUL TO SEE WHAT WILL BE JOINED.

anti\_join(x, y, by = NULL, ...)  
Return rows of x that do not have a match in y.  
USEFUL TO SEE WHAT WILL NOT BE JOINED.



## Part 3. 분석 & 시각화 - 'ggplot2'

"The simple graph has brought more information to the data analyst's mind than any other device."

(간단한 그래프는 다른 어떤 장치보다 데이터 분석가의 마음에 더 많은 정보를 제공합니다.)

- John Tukey

## mpg

- ggplot2 패키지에 내장된 데이터셋
- 1999년 부터 2008년 까지의 38개 차종 연비 데이터
- 데이터셋 diamond와 함께 ggplot 관련 문서에서 가장 많이 예제로 쓰임.

변수 이름	변수 설명
<i>manufacturer</i>	
<i>model</i>	차종 <i>Model name</i>
<i>displ</i>	엔진 크기 <i>engine displacement, in litres</i>
<i>year</i>	연식 <i>year of manufacture</i>
<i>cyl</i>	기통 <i>number of cylinders</i>
<i>trans</i>	트랜스미션 <i>type of transmission</i>
<i>drv</i>	전륜/후륜/사륜 $f = \text{front-wheel drive}$ , $r = \text{rear wheel drive}$ , $4 = 4\text{wd}$
<i>cty</i>	도심마일리지 <i>city miles per gallon</i>
<i>hwy</i>	고속도로마일리지 <i>highway miles per gallon</i>
<i>fl</i>	연료 종류 <i>fuel type</i>
<i>class</i>	(세단, SUV...) "type" of car

```

library(ggplot2)
? mpg

## starting httpd help server ... done
mpg

## # A tibble: 234 x 11
##   manufacturer model     displ  year   cyl trans   drv   cty   hwy fl
##   <chr>        <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr>
## 1 audi         a4       1.8  1999     4 auto(l~ f      18    29 p
## 2 audi         a4       1.8  1999     4 manual~ f     21    29 p
## 3 audi         a4        2   2008     4 manual~ f     20    31 p
## 4 audi         a4        2   2008     4 auto(a~ f      21    30 p
## 5 audi         a4       2.8  1999     6 auto(l~ f      16    26 p
## 6 audi         a4       2.8  1999     6 manual~ f     18    26 p
## 7 audi         a4       3.1  2008     6 auto(a~ f      18    27 p
## 8 audi         a4 quat~  1.8  1999     4 manual~ 4     18    26 p
## 9 audi         a4 quat~  1.8  1999     4 auto(l~ 4      16    25 p
## 10 audi        a4 quat~   2   2008     4 manual~ 4     20    28 p
## # ... with 224 more rows, and 1 more variable: class <chr>
class(mpg)

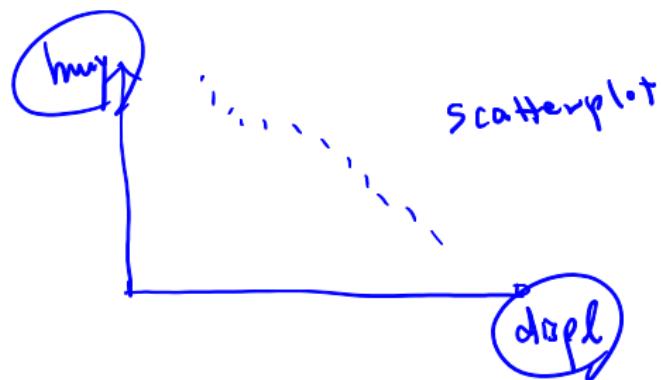
## [1] "tbl_df"     "tbl"        "data.frame"

```

- mpg는 여러개의 class의 특징을 가지고 있습니다.
- tbl\_df는 data.frame의 업그레이드 버전입니다. (몇 가지 기능 추가)

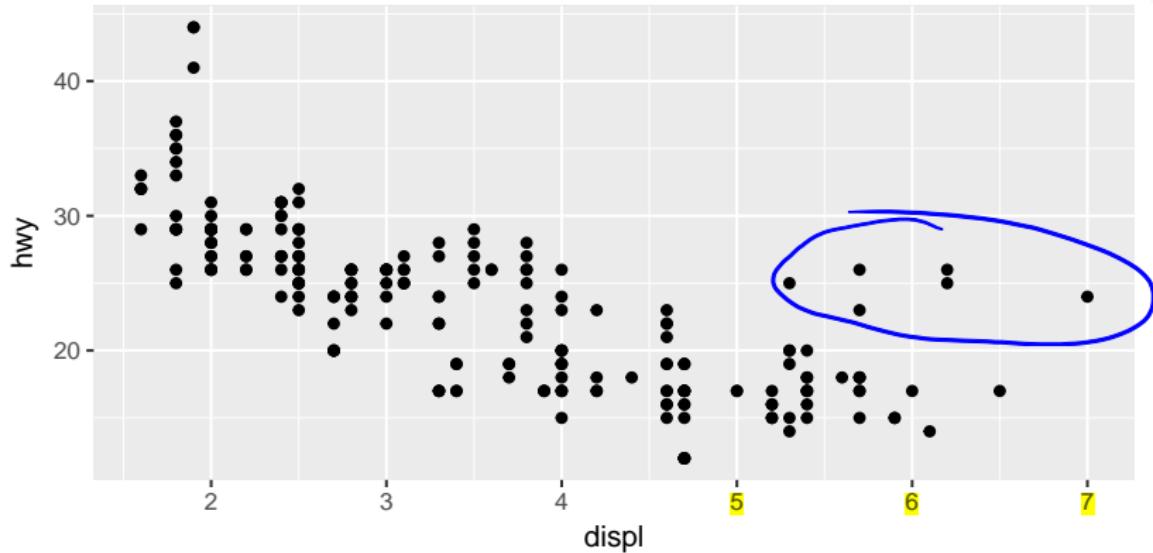
엔진이 크면 연비가 안 좋을까요?

↓      ↓  
dsgn      hwy  
 $X \rightarrow Y$



- 엔진 크기를 x축, 고속도로 마일리지를 y축으로 하는 산점도 (scatterplot)을 그려봅시다.

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```



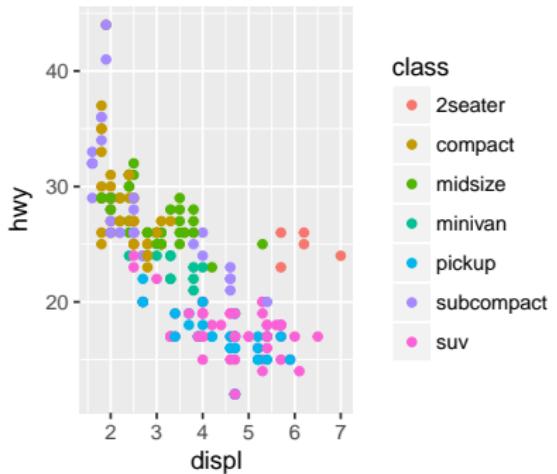
- 엔진이 크면 연비가 안 좋나요?
- 추가적으로 궁금한게 있나요?

```
# basic ggplot command  
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPING>))
```

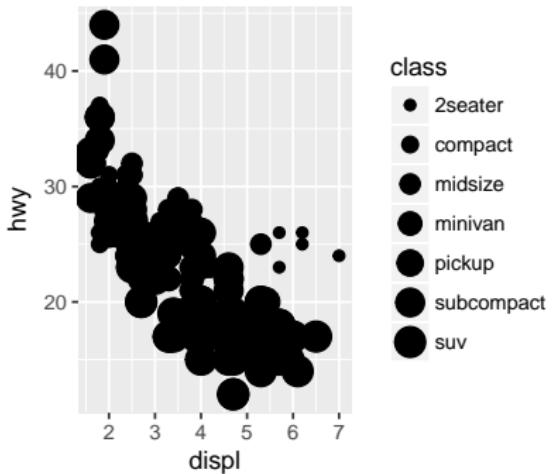
```

suppressMessages(library(gridExtra)) # grid.arrange()
a <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
b <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = class))
grid.arrange(a, b, nrow=1, ncol=2)

```



a



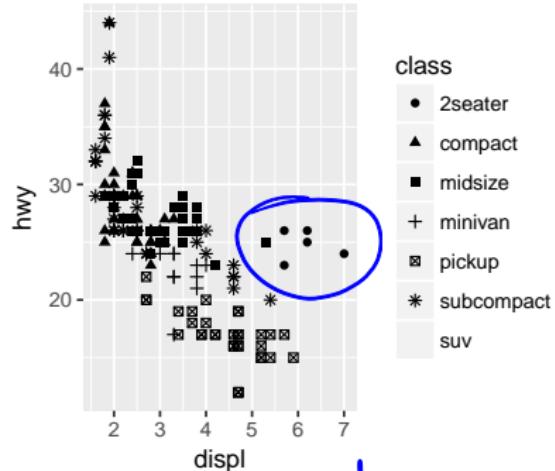
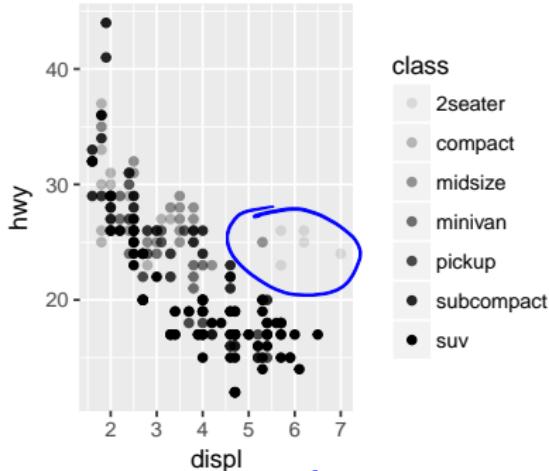
b

```

c <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
d <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))
grid.arrange(c, d, nrow=1, ncol=2)

## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 7.
## Consider specifying shapes manually if you must have them.

```



- a, b, c, d 중에 어느 그림이 가장 효과적인가요?
- 일반화 시키실 수 있나요? 즉, size, alpha, color, shape의 사용 기준은?

## Aesthetics (aes) - size, alpha, color, shape

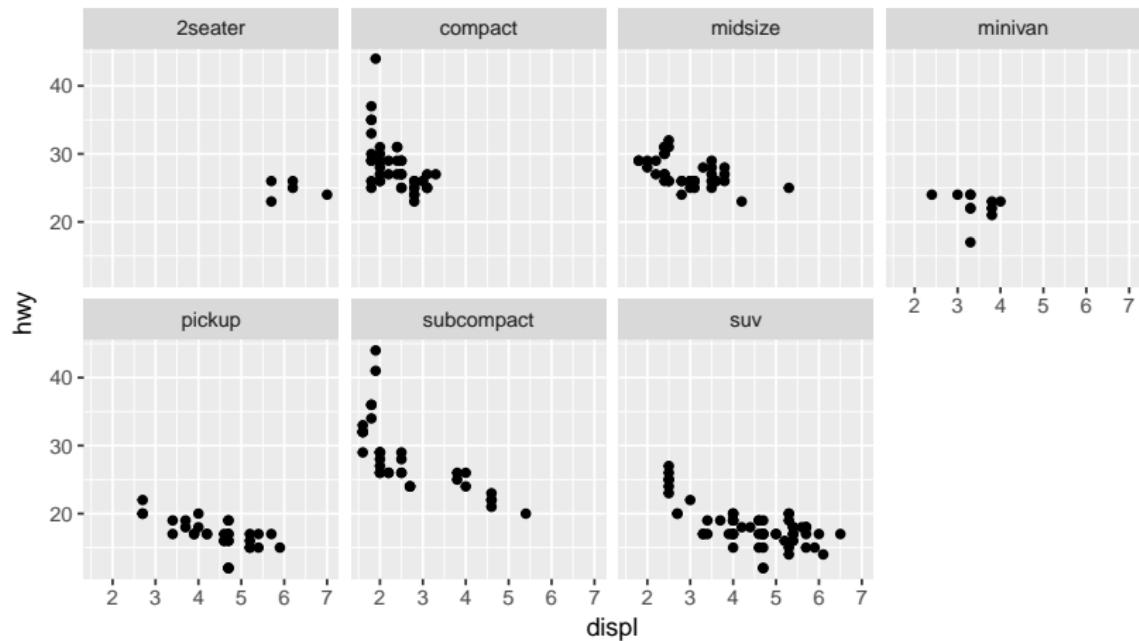
$\alpha$

변수 타입	ggplot aes	연속성	특징
Numeric (quantitative, 정량적)	<code>size</code> <code>alpha</code>	continuous (연속) or discrete (이산)	더할 수 있음 크고 작음
Factor (qualitative, 정량적)	<code>color</code> (<10) <code>shape</code> (<7)	discrete (이산)	더할 수 없음 집합 연산

## 자주 쓰이는 ggplot의 추가 기능들

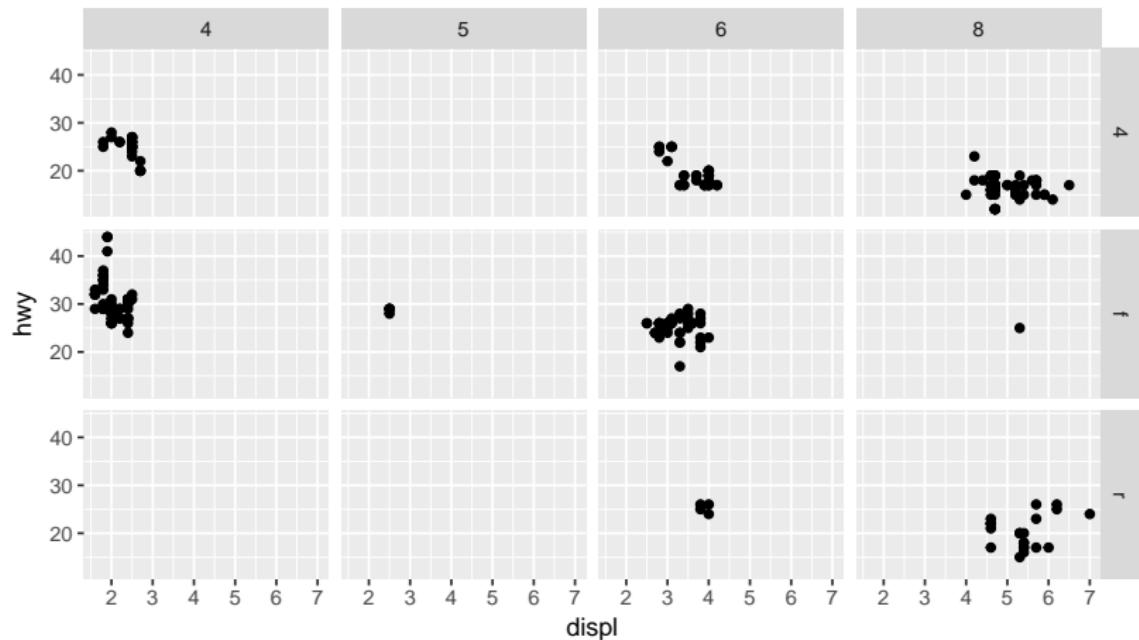
## Facets (1 variable) - facet\_wrap

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_wrap(~ class, nrow = 2)
```



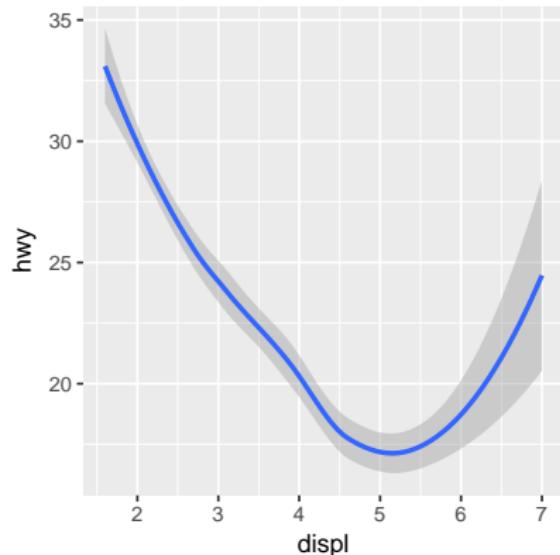
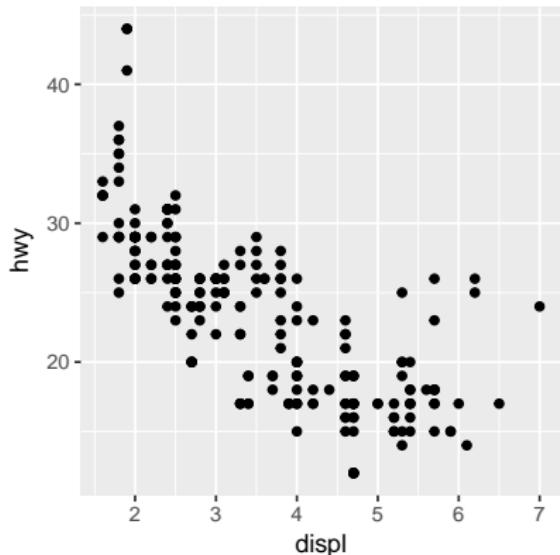
## Facets (2 variables) - facet\_grid

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_grid(drv ~ cyl)
```



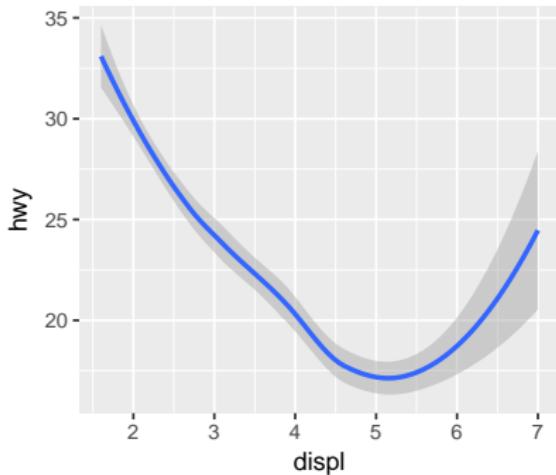
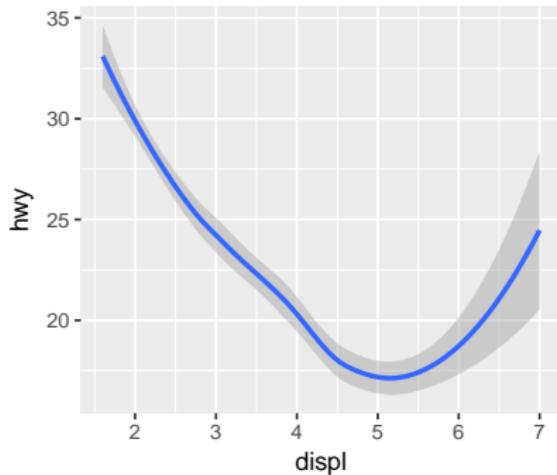
## Point vs Smooth

```
e <- ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))  
f <- ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))  
grid.arrange(e, f, nrow = 1, ncol = 2)
```



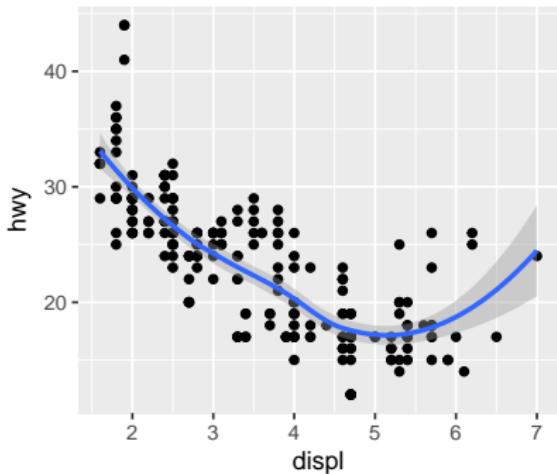
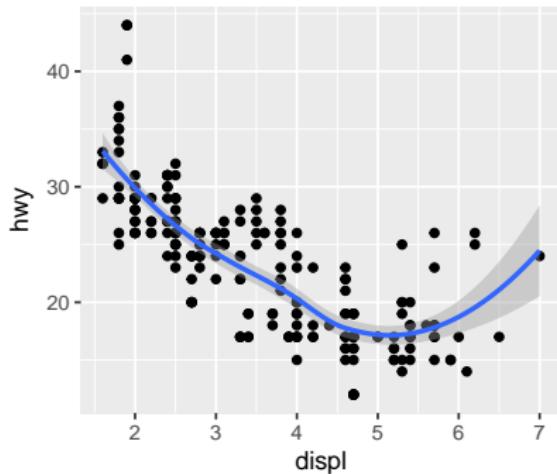
## AES mapping의 중첩

```
g <- ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
h <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth()
grid.arrange(g, h, nrow = 1, ncol = 2)
## `geom_smooth()` using method = 'loess'
```



## Point + Smooth

```
i <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
j <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
grid.arrange(i, j, nrow = 1, ncol = 2)
## `geom_smooth()` using method = 'loess'
```

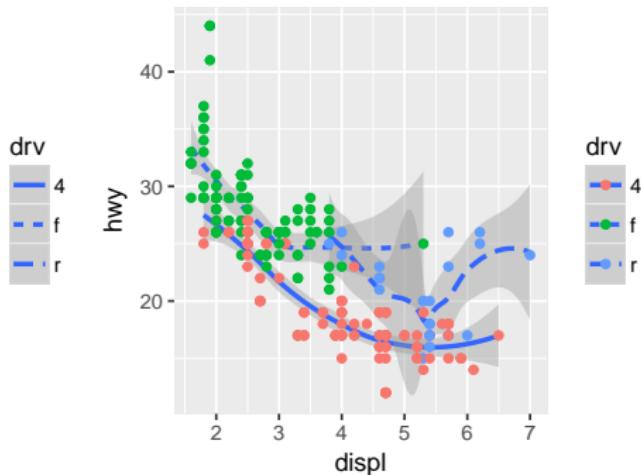
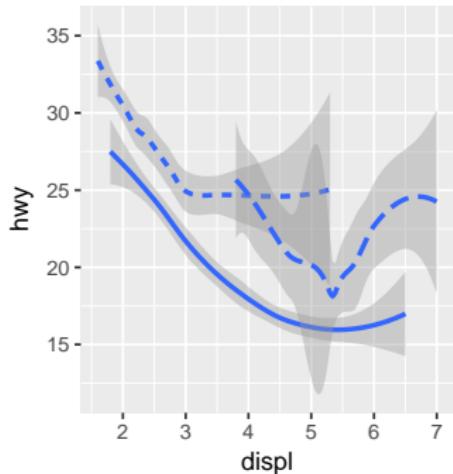


```

k <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth(aes(linetype = drv))
l <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth(aes(linetype = drv)) +
  geom_point(aes(color = drv))
grid.arrange(k, l, nrow = 1, ncol = 2)

## `geom_smooth()` using method = 'loess'

```



## Summary - ggplot

- **Data, Aesthetic, Geometric Object**를 명시하여 그림을 rendering.
- `gridExtra` 패키지의 `grid.arrange` 함수를 이용하여 `grid` 형태로 위치시킬 수 있음
- `Size > Alpha > Color >= Shape` 의 특성을 이해

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPING>))
```

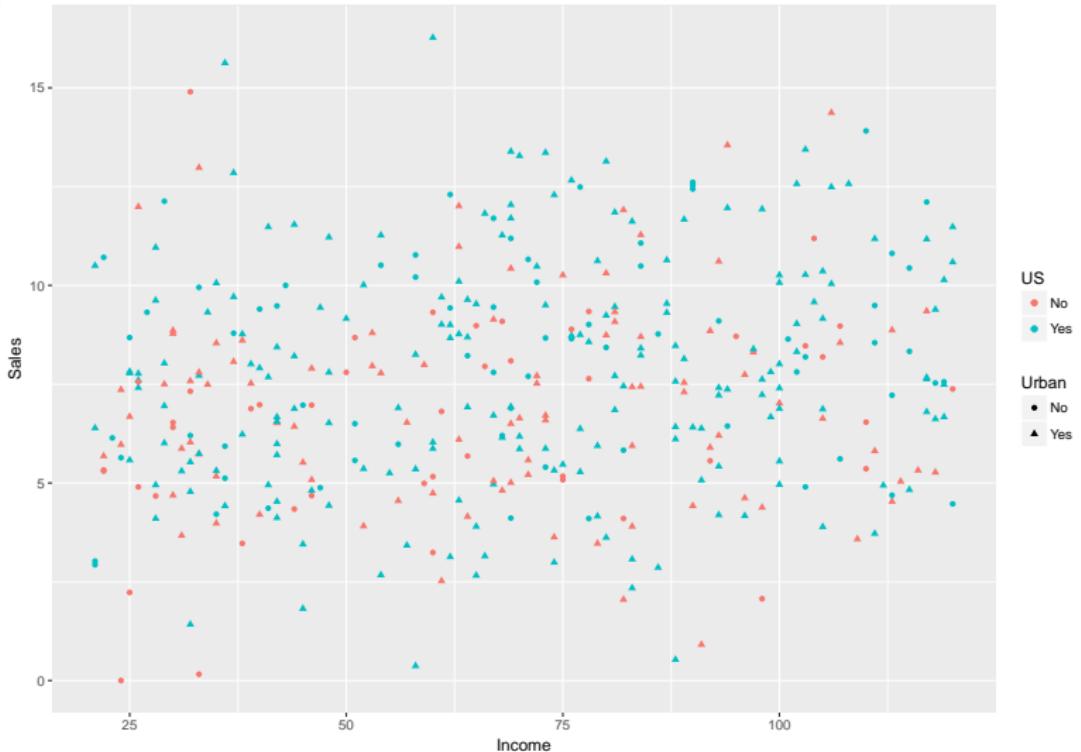
1. 그림의 목적을 정하고
2. X (explanatory variable - 설명 변수)의 개수와 각 변수의 중요도와 순서
3. Y (dependent variable)를 명시
4. ggplot 객체를 rendering

## Carseat

```
str(Carseats)

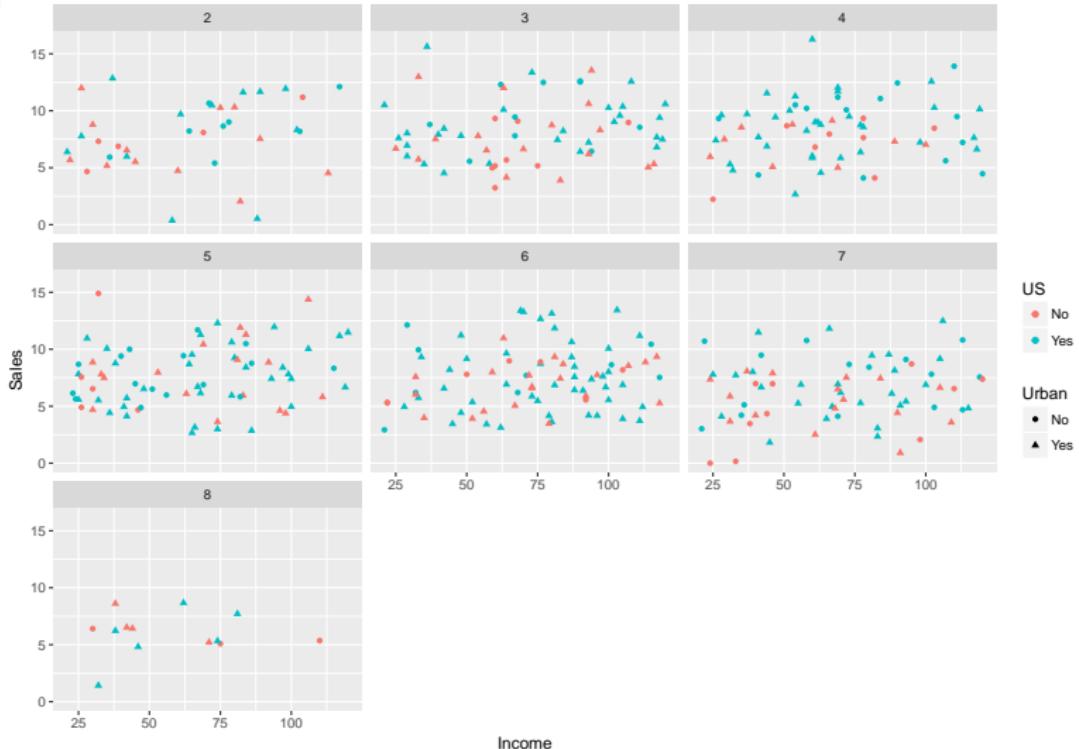
## 'data.frame': 400 obs. of 15 variables:
## $ Sales      : num  0.37 0 6.67 6.39 5.01 9.53 3.58 5.4 6.53 2.93 ...
## $ CompPrice   : num  147 139 156 131 159 175 142 149 154 143 ...
## $ Income      : num  58 24 42 21 69 65 109 73 30 21 ...
## $ Advertising : num  7 0 13 8 0 29 0 13 0 5 ...
## $ Population   : num  100 358 170 220 438 419 111 381 122 81 ...
## $ Price        : num  191 185 173 171 166 166 164 163 162 160 ...
## $ ShelveLoc    : Factor w/ 3 levels "Bad","Good","Medium": 1 3 2 2 3 3 2 1 3 3 ...
## $ Age          : num  27 79 74 29 46 53 72 26 57 67 ...
## $ Education    : num  15 15 14 14 17 12 12 11 17 12 ...
## $ Urban         : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 1 1 1 ...
## $ US            : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 2 1 2 1 2 ...
## $ Revenue       : num  0.37 0 6.67 6.39 5.01 9.53 3.58 5.4 6.53 2.93 ...
## $ AdvPerCapita  : num  0.07 0 0.0765 0.0364 0 ...
## $ RevPerCapita  : num  0.0037 0 0.0392 0.029 0.0114 ...
## $ AgeClass     : chr  "non-Silver" "Silver" "Silver" "non-Silver" ...
```

```
a <- ggplot(data = Carseats, aes(x = Income, y = Sales)) +  
  geom_point(aes(shape = Urban, color = US))  
print(a)
```



## Amazingly Additive

```
a <- a + facet_wrap(~ floor(Age/10))  
print(a)
```



- 2 slides above…

```
a <- ggplot(data = Carseats, aes(x = Income, y = Sales)) +  
  geom_point(aes(shape = Urban, color = US))
```

- 1 slide above…

```
a <- a + facet_wrap(~ floor(Age/10))
```

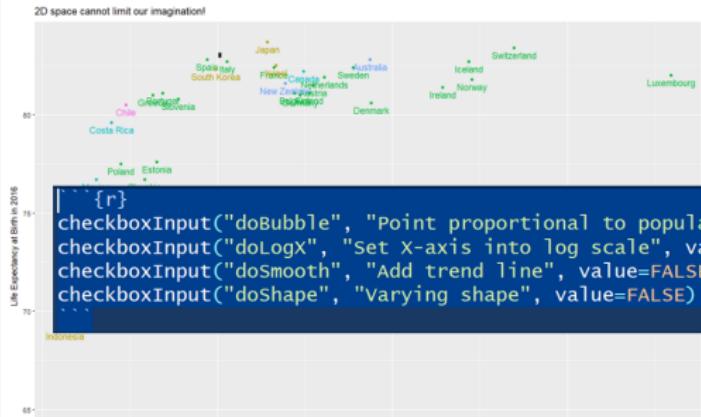
- 두 가지 기능을 합하고 사용자가 선택할 수 있게 합니다.
- Aggregate these and give a choice!

```
doAgeFacet <- TRUE  
a <- ggplot(data = Carseats, aes(x = Income, y = Sales)) +  
  geom_point(aes(shape = Urban, color = US))  
if (doAgeFacet) {  
  a <- a + facet_wrap(~ floor(Age/10))  
}  
print(a)
```

OECD

- Continents
- Asia
- Europe
- Africa
- North America
- South America
- Oceania

### Range of Country GDP (USD)



```
| ``{r}  
checkboxInput("doBubble", "Point proportional to population", value=FALSE)  
checkboxInput("doLogX", "Set X-axis into log scale", value=FALSE)  
checkboxInput("doSmooth", "Add trend line", value=FALSE)  
checkboxInput("doShape", "Varying shape", value=FALSE)
```

```
myGgplot <- ggplot(lifeCountry,
                    aes(x=GDP_per_Capita, y=Life.Expectancy, color=Continent)) +
  geom_text(aes(label=Country), size=4, vjust=1.5) +
  geom_point() +
  xlab("GDP per Capita in 2017 (in US Dollars)") +
  ylab("Life Expectancy at Birth in 2016") +
  ggtitle("2D space cannot limit our imagination!")
if (input$doBubble) { myGgplot <- myGgplot + geom_point(aes(size=Population)) }
if (input$doLogX) { myGgplot <- myGgplot + scale_x_log10() + xlab("GDP per Capita (log-scale)") }
if (input$doSmooth) { myGgplot <- myGgplot + geom_smooth(method='lm', se=TRUE, size=1) }
if (input$doShape) { myGgplot <- myGgplot + geom_point(aes(shape=Continent)) }
myGgplot
```

# Data Visualization with ggplot2 :: CHEAT SHEET

## Basics

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same components: a data set, a coordinate system, and geoms—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (aesthetics) like size, color, and x and y locations.



Complete the template below to build a graph.

```
ggplot(data = DATA) +
  GEOGRAPHICAL_FUNCTIONS(mapping = aes(MAPPINGS),
  stat = STAT, position = POSITION) +
  COORDINATE_FUNCTIONS+
  FACET_FUNCTIONS+
  SCALE_FUNCTIONS+
  THEME_FUNCTIONS
```

ggplot(data = mpg, aes(x = cyl, y = hwy)) begins a plot that you finish by adding layers to. Add one geom function per layer.

```
+ aesthetic mappings + data + geom  
ggplot(x = cyl, y = hwy, data = mpg, geom = "point")
```

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last\_plot() Returns the last plot.

```
ggplot("plot.png", width = 5, height = 5) Saves last plot as 5 x 5 and named "plot.png" in working directory.  
Matches file type to file extension.
```



## Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

### GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))  
b <- ggplot(bais, aes(x = long, y = lat))
```

a \* geom\_\*

Useful for expanding limits

```
b * geom_curve(aes(yend = lat + 1,  
xend = long), curvature = 0) x, vend, y, vend,  
alpha, angle, color, curvature, line_type, size
```

```
a * geom_pathline(path = "butt", linejoin = "round",  
x, y, alpha, color, group, line_type, size)
```

```
a * geom_polygon(aes(group = group))  
x, y, alpha, color, fill, group, line_type, size
```

```
b * geom_rect(aes(xmin = long, ymin = lat,  
xmax = long + 1, ymax = lat + 1), x_min, x_max,  
y_min, y_max, alpha, color, fill, group, line_type, size)
```

```
a * geom_ribbed(aes(min = unemploy - 900,  
max = unemploy + 900)) x, ymax, ymin,  
alpha, color, fill, group, line_type, size
```

### LINE SEGMENTS

common aesthetics: x, y, alpha, color, line\_type, size

```
b * geom_abline(aes(intercept = 0, slope = 1))  
b * geom_hline(aes(intercept = lat))
```

```
Not translatable  
discrete x, discrete y
```

```
b * geom_segment(aes(yend = lat + 1), vend = long + 1)  
b * geom_speaking(angle = 115.5, radius = 11)
```

continuous x, continuous y

!> ggplot(mpg, aes(class, hwy))

```
+ geom_line(mapping = aes(class, hwy))
```

```
+ geom_hline(mapping = aes(class, hwy))
```

```
+ geom_rect(mapping = aes(class, hwy))
```

```
+ geom_vline(mapping = aes(class, hwy))
```

```
+ geom_point(mapping = aes(class, hwy))
```

```
+ geom_bar(mapping = aes(class, hwy))
```

```
+ geom_text(mapping = aes(class, hwy))
```

```
+ geom_label(mapping = aes(class, hwy))
```

```
+ geom_raster(mapping = aes(class, hwy))
```

```
+ geom_map(mapping = aes(class, hwy))
```

```
+ geom_hex(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```

```
+ geom_hexstroke(mapping = aes(class, hwy))
```

```
+ geom_hexbin(mapping = aes(class, hwy))
```

```
+ geom_hexgrid(mapping = aes(class, hwy))
```



## More Reference

### 1. ggplot 관련 서적

- [Ref-external/books/ggplot2.pdf](#)

### 2. ggplot의 50개 예제

- [Ref-internal/M91-ggplot2-50examples](#)

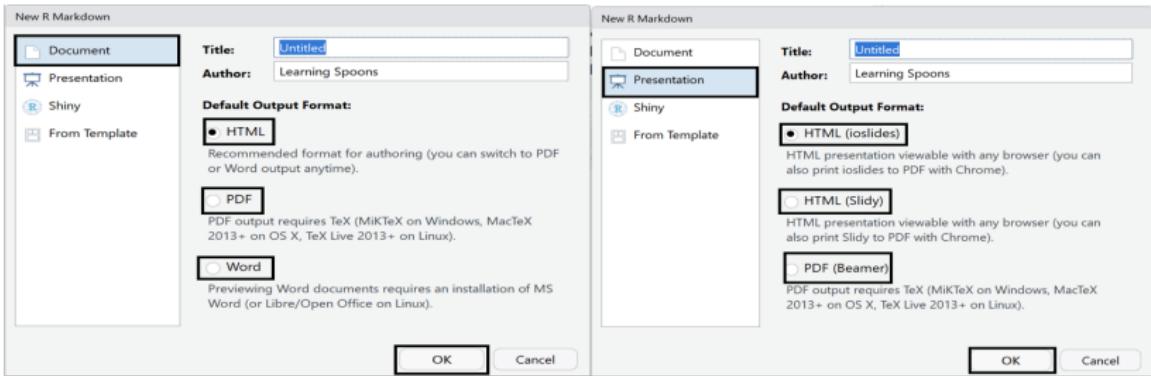
### 3. ggplot의 10주년을 기념하는 article

- [Ref-internal/M92-GGplotHappyBday](#)
- <https://qz.com/1007328/all-hail-ggplot2-the-code-powering-all-those-excellent-charts-is-10-years-old/>
- 크롤링과 구글 번역 API를 이용하여 자동 번역 및 문서화

## Part 4. 문서화 - "rmarkdown"

New Generation of Computer Programming  
Literate Programming

# 파일 -> 새파일 -> R Markdown



- *html*
  - *Interactive Feature 가능*
- *PDF*
  - *Professional 문서*
  - *Texlive가 설치되어 있어야 함.*
  - *한글을 위해서 별도의 template 사용*
- *Word*
  - *Office 없이도 문서 만들 수 있음*
- *ioslide, slidy*
  - *Interactive Feature 가능*
- *PDF (beamer)*
  - *Scientific Presentation*
  - *Texlive가 설치되어 있어야 함.*
  - *한글을 위해서 별도의 template 사용*

# html

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Untitled1 R Markdown Knit Run Insert Addins

```
1 title: "untitled"
2 author: "Learning Spoons"
3 date: "r Sys.Date()"
4 output:
5   html_document:
6     toc: TRUE
7
8
9
10 ---{r setup, include=FALSE}
11 knitr::opts_chunk$set(echo = TRUE)
12
13
14 ## R Markdown
15
16 For more details on using R Markdown see
<http://rmarkdown.rstudio.com>. When you click the **Knit** button a
document will be generated.
17
18 ---{r cars}
19 summary(cars)
20
21
22 만들기 R코드는 r ncol(cars) 와 같이 사용.
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ---{r pressure, echo=FALSE}
29 plot(pressure)
30
31
32
33
34
```

Console Terminal R Markdown

Untitled

file:///C:/Users/doxa7/Desktop/a.html

Untitled

Learning Spoons  
2018-04-17

R Markdown

For more details on using R Markdown see <http://rmarkdown.rstudio.com>. When you click the Knit button a document will be generated.

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

만들기 R코드는 2과 같이 사용.

Including Plots

You can also embed plots, for example:

# pdf (/Ref-rmdTemplates/M81-pdf)

```
1 + ---
2   title: "rmd pdf template"
3   author: "Learning Spoons"
4   date: "r Sys.Date()"
5   output:
6     pdf_document:
7       latex_engine: xelatex
8       highlight: haddock
9       keep_tex: true
10      # pandoc_args: [
11        # "-V", "classoption=twocolumn"
12      ]
13      smaller: true
14    mainfont: NanumGothic
15    classoption: a4paper
16    ...
17
18 - ``{r setup, include=FALSE}
19 knitr::opts_chunk$set(echo = TRUE)
20 ...
21
22 + ## R Markdown 한글
23
24 Code의 9,10,11 번째 라인의 pound sign(#)은 제거하고 위의 줄과 indent를
25 <http://rmarkdown.rstudio.com>.
26 When you click the **Knit** button a document will be generated.
27
28 + ``{r cars}
29 summary(cars)
30 ...
31
32 + ## Including Plots
33
34 + ``{r pressure, echo=FALSE, fig.height = 3}
35 plot(pressure)
36 ...
37
38 + ## Line Numbering
39
40 + ``{#numCode .R .numberLines}
41 + ``{r - 1-50}
42 rmd pdf template :
```

페이지: 1 / 1

rmd pdf template

Learning Spoons  
2018-06-30

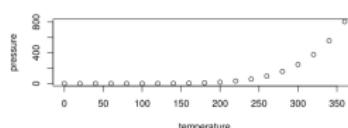
R Markdown 한글

Code의 9,10,11 번째 라인의 pound sign(#)을 제거하고 위의 줄과 indent를  
<http://rmarkdown.rstudio.com>.  
When you click the Knit button a document will be generated.

summary(cars)

	speed	dist
## Min	4.0	2.00
## 1st Qu.	12.0	1st Qu.: 26.00
## Median	15.0	Median : 36.00
## Mean	14.3	Mean : 36.77
## 3rd Qu.	19.0	3rd Qu.: 46.00
## Max.	28.0	Max. : 120.00

Including Plots



Line Numbering

```
x <- 1:10
y <- x^2
plot(x,y)
"Hello"
## [1] 1 2 3 4 5 6 7 8 9 10
## [1] "Hello"
```

# docx

The image shows a dual-pane view in RStudio. On the left, the R Markdown editor displays the following code:

```
1 ---  
2 title: "untitled"  
3 author: "Learning Spoons"  
4 date: "2018년 4월 17일"  
5 output: word_document  
6 ---  
7  
8 ## [r setup, include=FALSE]  
9 knitr:opts_chunk$set(echo = TRUE)  
10  
11  
12 ## R Markdown  
13  
14 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>.  
15  
16 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:  
17  
18 ##[r cars]  
19 summary(cars)  
20  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ##[r pressure, echo=FALSE]  
27 plot(pressure)  
28  
29  
30  
31
```

The right pane shows the generated Microsoft Word document titled "Untitled". The document contains the following text:

Learning Spoons.  
2018년 4월 17일.

## 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
```

At the bottom of the Word document, there is a status bar showing "1/2 페이지 151개 단어" and "영어(미국)".

ioslides

```
title: "Untitled"
author: "Learning Spoons"
date: "2018년 4월 17일"
output: ioslides_presentation
---

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

## R Markdown

This is an R Markdown presentation.
http://markdown.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 with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

## Slide with R Output

```{r cars, echo = TRUE}
summary(cars)
```

## Slide with plots
```

R Markdown

This is an R Markdown presentation.  
<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.

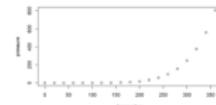
## Slide with Bullets

- Bullet
  - Bullet
  - Bullet

Slide with R Output

```
summary(cars)
```

Slide with Plot



# slidy

```
---
title: "Untitled"
author: "Learning Spoons"
date: "`r Sys.Date()`"
output:
  slidy_presentation: default
  ioslides_presentation: default
---

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

## R Markdown

This is an R Markdown presentation.
<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.

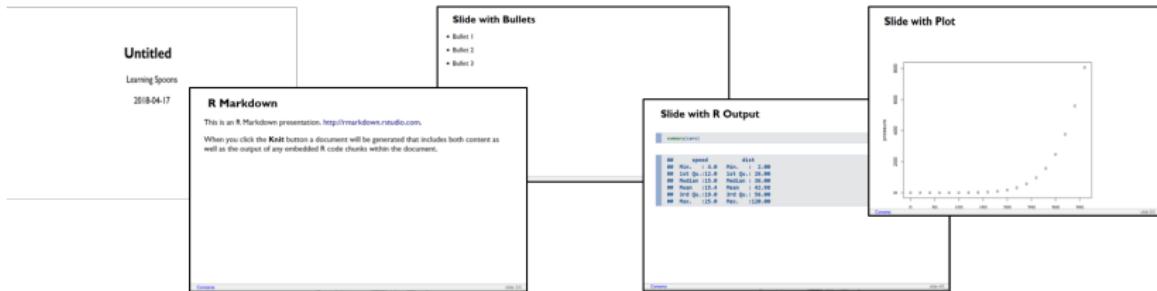
## Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

## Slide with R Output

```{r cars, echo = TRUE}
summary(cars)
```

```



# beamer (/Ref-rmdTemplates/M82-beamer)

The screenshot shows the RStudio interface with the 'rmd-beamer-template.Rmd' file open in the left pane. The code is written in R Markdown, defining a beamer presentation template. The right pane shows the Beamer presentation preview with three slides:

- Slide 1: Design**
  - 1. 헤더 부분의 아래 명령어를 검색해서 바꾸시면 다른 색상
  - + 'theme: "Singapore"'
  - + 'colortheme: "beaver"'
  - + 'knitr::opts\_chunk\$set(background = '71B0CA')'
- Slide 2: 구조**
  - 1. Left Column
    - 이 템플릿은 2단 구성이 가능합니다. \br
    - 'MSX-tidyR'의 강의노트가 beamer로 만든 좋은 예제입니다
  - 2. Right Column
- Slide 3: Code line number**
  - 1. Left Column
  - 2. Right Column

The code in the Rmd file includes:

```
1* ---  
2 title: "rnd - beamer - template"  
3 author: "LearningSpoonsR"  
4 date: `r Sys.Date()`"  
5 fontsize: 9pt  
6 output:  
7   beamer_presentation:  
8     theme: "Singapore"  
9     # For code line number, choose among  
10    # ["Antibes", "Montpellier", "Singapore", "Szeged"]  
11    colortheme: "beaver"  
12    # For Singapore : {beaver: print-friendly, "beetle": grey}  
13    latex_engine: xelatex  
14    # keep_tex: true  
15    # template: mytemplate.tex  
16    includes:  
17      in_header: myRndBeamerStyle/latex-topmatter.tex  
18    classoption: t |  
19    mainfont: NanumGothic  
20 ---  
21  
22  ***{r setup, include=FALSE}  
23  library(rmarkdown)  
24  knitr::opts_chunk$set(echo = TRUE)  
25  knitr::opts_chunk$set(background = '71B0CA')  
26  ***  
27  
28  ## beamer@Rmarkdown  
29  
30  - 이 템플릿은 한글 및 twocolumn layout으로 beamer 문서를 제작할 수 있는 템플릿입니다. \br  
31  - RMarkdown ('M2X-rnd' 를 참조) \br  
32  - pdf 조판을 위한 texlive 엔진 ('M2X-rnd' 를 참조) \br  
33  - slide 형태의 pdf를 만드는 beamer 폐키지 (r 폐키지가 아니라 tex 폐키지) \br  
34  - 'tex' 설정들의 작업이 되었어야 이 템플릿을 사용할 수 있습니다. \br  
35  - 하위 폴더인 'myRndBeamerStyle'이 있고, 폴더안에는 'latex-topmatter.tex' 와 'markdown_cus...  
36
```

# Example - 1주차 숙제

The image shows two overlapping windows. The left window is RStudio with the following R code:

```
1 --
2 title: "Review - week1"
3 author: "learningspoonsR"
4 date: "2018년 4월 15일"
5 output: word_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 knitr::opts_chunk$set(results = 'hide')
11 ```
12
13 ## Module 1 - Hello world
14
15 `r problem <- 1`
16
17 **Problem** `r problem` . `r problem <- problem + 1`
18 ```{r}
19 a <- "Hello"
20 a
21 ```
22 ans:
23
24
25 **Problem** `r problem` . `r problem <- problem + 1`
26 ```{r}
27 a <- "Hello"
28 b <- "world"
29 paste(a,b)
30 ```
31 ans:
32
```

The right window is a Korean application with tabs like 자동 저장, 파일, 흘, Review... (저장됨), 등. It displays the following output:

Review - Week1  
learningSpoonsR  
2018년 4월 15일

Module 1 - Hello World

Problem 1.  
a <- "Hello";  
a;  
ans;  
ans;

Problem 2.  
a <- "Hello";  
b <- "World";  
paste(a,b);  
ans;

Problem 3.  
paste(a,b);  
ans;

Problem 4.  
paste(a,b,sep="");  
ans;

```
**Problem** `r problem`.  
`r problem <- problem + 1`  
```{r}  
grep("ui", font)  
```  
```{r, echo=FALSE}  
ans0 <- "ANS: The function grep somehow returns 0 or 1, which is  
equivalent to TRUE or FALSE!"  
ans0  
```  
ans:
```

---

### Problem 3.

```
grep("ui", font)  
## [1] 1  
## [1] "ANS: The function grep somehow returns 0 or 1, which is equivalent to  
TRUE or FALSE!"  
ans:
```

# R Markdown :: CHEAT SHEET

## What is R Markdown?

**Readable File**: An R Markdown (.Rmd) file is a record of your research. It contains the code that a researcher used to produce the work along with the narration that a reader can reuse.

**Reproducible Research**: At the click of a button, or the type of a command, you can reuse the code in an R Markdown file to generate a report and export the results as a finished report.

**Dynamic Documents**: You can choose to export your R Markdown document in many formats, including HTML, pdf, MS Word, or RTF documents; html and pdf based slides; Notebooks, and more.

## Workflow



- ① Open a new .Rmd file > New File > R Markdown. Use the template that opens to prompt the user to enter a title.
- ② Write document by editing template
- ③ Knit document to create report; use knit button or render() to knit
- ④ Preview Output in IDE window
- ⑤ Publish (optional) to web server
- ⑥ Generate build log; R Markdown console
- ⑦ 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 `knitr(knitr)` Built with 3.3.3

### IMPORTANT CHUNK OPTIONS

`cache` - cache results for future knits (default = FALSE)  
`cache.path` - directory to save cached results in (default = "cache")  
`child` - file to knit and then include (default = NULL)  
`engine` - engine to use (default = "R")  
`collapse` - collapse all output into single block (default = FALSE)  
`comment` - prefix for each line of results (default = "#")

`dependsOn` - chunk dependencies for caching (default = NULL)

`eval` - display code in output document (default = TRUE)

`engine` - code language used in chunk (default = "R")

`error` - display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)

`eval` - run code in chunk (default = TRUE)

`options` - options not listed above (e.g., args, autodev, background, cache.comments, cache.laz, cache.rebuild, cache.vars, dev, dev.args, dpo, engine.opts, engine.path, fig.asp, fig.eval, fig.height, fig.out, fig.path, fig.proc, fig.refin, fig.scap, fig.show, fig.showtext, fig.subcap, interval, nfc.attrs, out.height, out.width, print, print.pdf, reflabel, render, tidy, split, tidy.opts)

`fig.align` - left, right, or center (default = "center")

`fig.cap` - figure caption as character string (default = NULL)

`fig.height` - height of plots in inches (default = 1)

`fig.width` - dimensions of plots in inches (default = 3)

`highlight` - highlight source code (default = TRUE)

`include` - include chunk in doc after running (default = TRUE)

`message` - display code messages in document (default = TRUE)

`options` - (default = "warn")

`asis` - maintains results

`asis` - put all results in one code block

`tidy` - code for display (default = FALSE)

`warning` - display code warnings in document (default = TRUE)

**R Markdown**  
RStudio  
+ R Markdown

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

**summary(cars)**

```
## #> #> speed dist
## #> 1 5.0 4.0
## #> 2 4.0 4.0
## #> 3 12.0 11.0
## #> 4 11.0 14.0
## #> 5 15.0 12.0
## #> 6 13.0 10.0
## #> 7 15.0 16.0
## #> 8 14.0 10.0
## #> 9 16.0 15.0
## #> 10 17.0 12.0
## #> 11 15.0 14.0
## #> 12 14.0 13.0
## #> 13 17.0 15.0
## #> 14 15.0 12.0
## #> 15 14.0 13.0
## #> 16 17.0 15.0
## #> 17 15.0 12.0
## #> 18 14.0 13.0
## #> 19 17.0 15.0
## #> 20 15.0 12.0
## #> 21 14.0 13.0
## #> 22 17.0 15.0
## #> 23 15.0 12.0
## #> 24 14.0 13.0
## #> 25 17.0 15.0
## #> 26 15.0 12.0
## #> 27 14.0 13.0
## #> 28 17.0 15.0
## #> 29 15.0 12.0
## #> 30 14.0 13.0
## #> 31 17.0 15.0
## #> 32 15.0 12.0
## #> 33 14.0 13.0
## #> 34 17.0 15.0
## #> 35 15.0 12.0
## #> 36 14.0 13.0
## #> 37 17.0 15.0
## #> 38 15.0 12.0
## #> 39 14.0 13.0
## #> 40 17.0 15.0
## #> 41 15.0 12.0
## #> 42 14.0 13.0
## #> 43 17.0 15.0
## #> 44 15.0 12.0
## #> 45 14.0 13.0
## #> 46 17.0 15.0
## #> 47 15.0 12.0
## #> 48 14.0 13.0
## #> 49 17.0 15.0
## #> 50 15.0 12.0
## #> 51 14.0 13.0
## #> 52 17.0 15.0
## #> 53 15.0 12.0
## #> 54 14.0 13.0
## #> 55 17.0 15.0
## #> 56 15.0 12.0
## #> 57 14.0 13.0
## #> 58 17.0 15.0
## #> 59 15.0 12.0
## #> 60 14.0 13.0
## #> 61 17.0 15.0
## #> 62 15.0 12.0
## #> 63 14.0 13.0
## #> 64 17.0 15.0
## #> 65 15.0 12.0
## #> 66 14.0 13.0
## #> 67 17.0 15.0
## #> 68 15.0 12.0
## #> 69 14.0 13.0
## #> 70 17.0 15.0
## #> 71 15.0 12.0
## #> 72 14.0 13.0
## #> 73 17.0 15.0
## #> 74 15.0 12.0
## #> 75 14.0 13.0
## #> 76 17.0 15.0
## #> 77 15.0 12.0
## #> 78 14.0 13.0
## #> 79 17.0 15.0
## #> 80 15.0 12.0
## #> 81 14.0 13.0
## #> 82 17.0 15.0
## #> 83 15.0 12.0
## #> 84 14.0 13.0
## #> 85 17.0 15.0
## #> 86 15.0 12.0
## #> 87 14.0 13.0
## #> 88 17.0 15.0
## #> 89 15.0 12.0
## #> 90 14.0 13.0
## #> 91 17.0 15.0
## #> 92 15.0 12.0
## #> 93 14.0 13.0
## #> 94 17.0 15.0
## #> 95 15.0 12.0
## #> 96 14.0 13.0
## #> 97 17.0 15.0
## #> 98 15.0 12.0
## #> 99 14.0 13.0
## #> 100 17.0 15.0
## #> 101 15.0 12.0
## #> 102 14.0 13.0
## #> 103 17.0 15.0
## #> 104 15.0 12.0
## #> 105 14.0 13.0
## #> 106 17.0 15.0
## #> 107 15.0 12.0
## #> 108 14.0 13.0
## #> 109 17.0 15.0
## #> 110 15.0 12.0
## #> 111 14.0 13.0
## #> 112 17.0 15.0
## #> 113 15.0 12.0
## #> 114 14.0 13.0
## #> 115 17.0 15.0
## #> 116 15.0 12.0
## #> 117 14.0 13.0
## #> 118 17.0 15.0
## #> 119 15.0 12.0
## #> 120 14.0 13.0
## #> 121 17.0 15.0
## #> 122 15.0 12.0
## #> 123 14.0 13.0
## #> 124 17.0 15.0
## #> 125 15.0 12.0
## #> 126 14.0 13.0
## #> 127 17.0 15.0
## #> 128 15.0 12.0
## #> 129 14.0 13.0
## #> 130 17.0 15.0
## #> 131 15.0 12.0
## #> 132 14.0 13.0
## #> 133 17.0 15.0
## #> 134 15.0 12.0
## #> 135 14.0 13.0
## #> 136 17.0 15.0
## #> 137 15.0 12.0
## #> 138 14.0 13.0
## #> 139 17.0 15.0
## #> 140 15.0 12.0
## #> 141 14.0 13.0
## #> 142 17.0 15.0
## #> 143 15.0 12.0
## #> 144 14.0 13.0
## #> 145 17.0 15.0
## #> 146 15.0 12.0
## #> 147 14.0 13.0
## #> 148 17.0 15.0
## #> 149 15.0 12.0
## #> 150 14.0 13.0
## #> 151 17.0 15.0
## #> 152 15.0 12.0
## #> 153 14.0 13.0
## #> 154 17.0 15.0
## #> 155 15.0 12.0
## #> 156 14.0 13.0
## #> 157 17.0 15.0
## #> 158 15.0 12.0
## #> 159 14.0 13.0
## #> 160 17.0 15.0
## #> 161 15.0 12.0
## #> 162 14.0 13.0
## #> 163 17.0 15.0
## #> 164 15.0 12.0
## #> 165 14.0 13.0
## #> 166 17.0 15.0
## #> 167 15.0 12.0
## #> 168 14.0 13.0
## #> 169 17.0 15.0
## #> 170 15.0 12.0
## #> 171 14.0 13.0
## #> 172 17.0 15.0
## #> 173 15.0 12.0
## #> 174 14.0 13.0
## #> 175 17.0 15.0
## #> 176 15.0 12.0
## #> 177 14.0 13.0
## #> 178 17.0 15.0
## #> 179 15.0 12.0
## #> 180 14.0 13.0
## #> 181 17.0 15.0
## #> 182 15.0 12.0
## #> 183 14.0 13.0
## #> 184 17.0 15.0
## #> 185 15.0 12.0
## #> 186 14.0 13.0
## #> 187 17.0 15.0
## #> 188 15.0 12.0
## #> 189 14.0 13.0
## #> 190 17.0 15.0
## #> 191 15.0 12.0
## #> 192 14.0 13.0
## #> 193 17.0 15.0
## #> 194 15.0 12.0
## #> 195 14.0 13.0
## #> 196 17.0 15.0
## #> 197 15.0 12.0
## #> 198 14.0 13.0
## #> 199 17.0 15.0
## #> 200 15.0 12.0
## #> 201 14.0 13.0
## #> 202 17.0 15.0
## #> 203 15.0 12.0
## #> 204 14.0 13.0
## #> 205 17.0 15.0
## #> 206 15.0 12.0
## #> 207 14.0 13.0
## #> 208 17.0 15.0
## #> 209 15.0 12.0
## #> 210 14.0 13.0
## #> 211 17.0 15.0
## #> 212 15.0 12.0
## #> 213 14.0 13.0
## #> 214 17.0 15.0
## #> 215 15.0 12.0
## #> 216 14.0 13.0
## #> 217 17.0 15.0
## #> 218 15.0 12.0
## #> 219 14.0 13.0
## #> 220 17.0 15.0
## #> 221 15.0 12.0
## #> 222 14.0 13.0
## #> 223 17.0 15.0
## #> 224 15.0 12.0
## #> 225 14.0 13.0
## #> 226 17.0 15.0
## #> 227 15.0 12.0
## #> 228 14.0 13.0
## #> 229 17.0 15.0
## #> 230 15.0 12.0
## #> 231 14.0 13.0
## #> 232 17.0 15.0
## #> 233 15.0 12.0
## #> 234 14.0 13.0
## #> 235 17.0 15.0
## #> 236 15.0 12.0
## #> 237 14.0 13.0
## #> 238 17.0 15.0
## #> 239 15.0 12.0
## #> 240 14.0 13.0
## #> 241 17.0 15.0
## #> 242 15.0 12.0
## #> 243 14.0 13.0
## #> 244 17.0 15.0
## #> 245 15.0 12.0
## #> 246 14.0 13.0
## #> 247 17.0 15.0
## #> 248 15.0 12.0
## #> 249 14.0 13.0
## #> 250 17.0 15.0
## #> 251 15.0 12.0
## #> 252 14.0 13.0
## #> 253 17.0 15.0
## #> 254 15.0 12.0
## #> 255 14.0 13.0
## #> 256 17.0 15.0
## #> 257 15.0 12.0
## #> 258 14.0 13.0
## #> 259 17.0 15.0
## #> 260 15.0 12.0
## #> 261 14.0 13.0
## #> 262 17.0 15.0
## #> 263 15.0 12.0
## #> 264 14.0 13.0
## #> 265 17.0 15.0
## #> 266 15.0 12.0
## #> 267 14.0 13.0
## #> 268 17.0 15.0
## #> 269 15.0 12.0
## #> 270 14.0 13.0
## #> 271 17.0 15.0
## #> 272 15.0 12.0
## #> 273 14.0 13.0
## #> 274 17.0 15.0
## #> 275 15.0 12.0
## #> 276 14.0 13.0
## #> 277 17.0 15.0
## #> 278 15.0 12.0
## #> 279 14.0 13.0
## #> 280 17.0 15.0
## #> 281 15.0 12.0
## #> 282 14.0 13.0
## #> 283 17.0 15.0
## #> 284 15.0 12.0
## #> 285 14.0 13.0
## #> 286 17.0 15.0
## #> 287 15.0 12.0
## #> 288 14.0 13.0
## #> 289 17.0 15.0
## #> 290 15.0 12.0
## #> 291 14.0 13.0
## #> 292 17.0 15.0
## #> 293 15.0 12.0
## #> 294 14.0 13.0
## #> 295 17.0 15.0
## #> 296 15.0 12.0
## #> 297 14.0 13.0
## #> 298 17.0 15.0
## #> 299 15.0 12.0
## #> 300 14.0 13.0
## #> 301 17.0 15.0
## #> 302 15.0 12.0
## #> 303 14.0 13.0
## #> 304 17.0 15.0
## #> 305 15.0 12.0
## #> 306 14.0 13.0
## #> 307 17.0 15.0
## #> 308 15.0 12.0
## #> 309 14.0 13.0
## #> 310 17.0 15.0
## #> 311 15.0 12.0
## #> 312 14.0 13.0
## #> 313 17.0 15.0
## #> 314 15.0 12.0
## #> 315 14.0 13.0
## #> 316 17.0 15.0
## #> 317 15.0 12.0
## #> 318 14.0 13.0
## #> 319 17.0 15.0
## #> 320 15.0 12.0
## #> 321 14.0 13.0
## #> 322 17.0 15.0
## #> 323 15.0 12.0
## #> 324 14.0 13.0
## #> 325 17.0 15.0
## #> 326 15.0 12.0
## #> 327 14.0 13.0
## #> 328 17.0 15.0
## #> 329 15.0 12.0
## #> 330 14.0 13.0
## #> 331 17.0 15.0
## #> 332 15.0 12.0
## #> 333 14.0 13.0
## #> 334 17.0 15.0
## #> 335 15.0 12.0
## #> 336 14.0 13.0
## #> 337 17.0 15.0
## #> 338 15.0 12.0
## #> 339 14.0 13.0
## #> 340 17.0 15.0
## #> 341 15.0 12.0
## #> 342 14.0 13.0
## #> 343 17.0 15.0
## #> 344 15.0 12.0
## #> 345 14.0 13.0
## #> 346 17.0 15.0
## #> 347 15.0 12.0
## #> 348 14.0 13.0
## #> 349 17.0 15.0
## #> 350 15.0 12.0
## #> 351 14.0 13.0
## #> 352 17.0 15.0
## #> 353 15.0 12.0
## #> 354 14.0 13.0
## #> 355 17.0 15.0
## #> 356 15.0 12.0
## #> 357 14.0 13.0
## #> 358 17.0 15.0
## #> 359 15.0 12.0
## #> 360 14.0 13.0
## #> 361 17.0 15.0
## #> 362 15.0 12.0
## #> 363 14.0 13.0
## #> 364 17.0 15.0
## #> 365 15.0 12.0
## #> 366 14.0 13.0
## #> 367 17.0 15.0
## #> 368 15.0 12.0
## #> 369 14.0 13.0
## #> 370 17.0 15.0
## #> 371 15.0 12.0
## #> 372 14.0 13.0
## #> 373 17.0 15.0
## #> 374 15.0 12.0
## #> 375 14.0 13.0
## #> 376 17.0 15.0
## #> 377 15.0 12.0
## #> 378 14.0 13.0
## #> 379 17.0 15.0
## #> 380 15.0 12.0
## #> 381 14.0 13.0
## #> 382 17.0 15.0
## #> 383 15.0 12.0
## #> 384 14.0 13.0
## #> 385 17.0 15.0
## #> 386 15.0 12.0
## #> 387 14.0 13.0
## #> 388 17.0 15.0
## #> 389 15.0 12.0
## #> 390 14.0 13.0
## #> 391 17.0 15.0
## #> 392 15.0 12.0
## #> 393 14.0 13.0
## #> 394 17.0 15.0
## #> 395 15.0 12.0
## #> 396 14.0 13.0
## #> 397 17.0 15.0
## #> 398 15.0 12.0
## #> 399 14.0 13.0
## #> 400 17.0 15.0
## #> 401 15.0 12.0
## #> 402 14.0 13.0
## #> 403 17.0 15.0
## #> 404 15.0 12.0
## #> 405 14.0 13.0
## #> 406 17.0 15.0
## #> 407 15.0 12.0
## #> 408 14.0 13.0
## #> 409 17.0 15.0
## #> 410 15.0 12.0
## #> 411 14.0 13.0
## #> 412 17.0 15.0
## #> 413 15.0 12.0
## #> 414 14.0 13.0
## #> 415 17.0 15.0
## #> 416 15.0 12.0
## #> 417 14.0 13.0
## #> 418 17.0 15.0
## #> 419 15.0 12.0
## #> 420 14.0 13.0
## #> 421 17.0 15.0
## #> 422 15.0 12.0
## #> 423 14.0 13.0
## #> 424 17.0 15.0
## #> 425 15.0 12.0
## #> 426 14.0 13.0
## #> 427 17.0 15.0
## #> 428 15.0 12.0
## #> 429 14.0 13.0
## #> 430 17.0 15.0
## #> 431 15.0 12.0
## #> 432 14.0 13.0
## #> 433 17.0 15.0
## #> 434 15.0 12.0
## #> 435 14.0 13.0
## #> 436 17.0 15.0
## #> 437 15.0 12.0
## #> 438 14.0 13.0
## #> 439 17.0 15.0
## #> 440 15.0 12.0
## #> 441 14.0 13.0
## #> 442 17.0 15.0
## #> 443 15.0 12.0
## #> 444 14.0 13.0
## #> 445 17.0 15.0
## #> 446 15.0 12.0
## #> 447 14.0 13.0
## #> 448 17.0 15.0
## #> 449 15.0 12.0
## #> 450 14.0 13.0
## #> 451 17.0 15.0
## #> 452 15.0 12.0
## #> 453 14.0 13.0
## #> 454 17.0 15.0
## #> 455 15.0 12.0
## #> 456 14.0 13.0
## #> 457 17.0 15.0
## #> 458 15.0 12.0
## #> 459 14.0 13.0
## #> 460 17.0 15.0
## #> 461 15.0 12.0
## #> 462 14.0 13.0
## #> 463 17.0 15.0
## #> 464 15.0 12.0
## #> 465 14.0 13.0
## #> 466 17.0 15.0
## #> 467 15.0 12.0
## #> 468 14.0 13.0
## #> 469 17.0 15.0
## #> 470 15.0 12.0
## #> 471 14.0 13.0
## #> 472 17.0 15.0
## #> 473 15.0 12.0
## #> 474 14.0 13.0
## #> 475 17.0 15.0
## #> 476 15.0 12.0
## #> 477 14.0 13.0
## #> 478 17.0 15.0
## #> 479 15.0 12.0
## #> 480 14.0 13.0
## #> 481 17.0 15.0
## #> 482 15.0 12.0
## #> 483 14.0 13.0
## #> 484 17.0 15.0
## #> 485 15.0 12.0
## #> 486 14.0 13.0
## #> 487 17.0 15.0
## #> 488 15.0 12.0
## #> 489 14.0 13.0
## #> 490 17.0 15.0
## #> 491 15.0 12.0
## #> 492 14.0 13.0
## #> 493 17.0 15.0
## #> 494 15.0 12.0
## #> 495 14.0 13.0
## #> 496 17.0 15.0
## #> 497 15.0 12.0
## #> 498 14.0 13.0
## #> 499 17.0 15.0
## #> 500 15.0 12.0
## #> 501 14.0 13.0
## #> 502 17.0 15.0
## #> 503 15.0 12.0
## #> 504 14.0 13.0
## #> 505 17.0 15.0
## #> 506 15.0 12.0
## #> 507 14.0 13.0
## #> 508 17.0 15.0
## #> 509 15.0 12.0
## #> 510 14.0 13.0
## #> 511 17.0 15.0
## #> 512 15.0 12.0
## #> 513 14.0 13.0
## #> 514 17.0 15.0
## #> 515 15.0 12.0
## #> 516 14.0 13.0
## #> 517 17.0 15.0
## #> 518 15.0 12.0
## #> 519 14.0 13.0
## #> 520 17.0 15.0
## #> 521 15.0 12.0
## #> 522 14.0 13.0
## #> 5
```



# R 마크다운

初心者

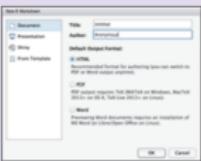
추가 학습 정보 [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)

rmarkdown 0.2.50 최종 갱신일: 8/14



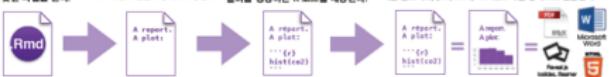
## 2. 파일 열기 .Rmd 확장자를 갖는 텍스트 파일로 저장해서 시작하거나, Studio Rmd 템플릿을 열어 시작한다.

- 메뉴에서, 다음순으로 클릭한다.  
**File ▶ New File ▶ R Markdown...**
- 윈도우가 열리면, .Rmd 파일로 작성하는 출력 형식을 선택한다.
- 라디오 버튼으로 출력 형식을 선택한다 (나중에 출력 형식은 변경할 수 있다)
- OK 버튼을 클릭한다.



**1. 작업흐름** R 마크다운은 R로 재현 가능하고, 동적인 보고서를 작성하는 서식이다. R 마크다운을 사용해서 R 코드와 실행 결과를 발표자료, pdf, html, 워드 문서 등에 삽입할 수 있다. 보고서를 작성하려면:

- I. 파일열기 - Rmd 확장자를 갖는 파일을 연다.
- II. 작성하기 - 본문을 작성하기 위해 R 마크다운 구문을 사용해서 작성한다.
- III. 내장하기 - 리포트에 포함될 출력 결과를 생성하는 R 코드를 내장한다.
- IV. 렌더링(Render) - R 코드를 출력 형식으로 치환하고 보고서를 발표자료, pdf, html, MS 워드 파일 형식으로 변환한다.



## 4. 출력 형식 설정 R 마크다운 파일에서 생성할 문서 유형을 기술하는 YAML 헤더정보를 작성한다.

### YAML

YAML 헤더는 키(key) 집합:  
파일 시작지점에 나오는 키-값 쌍.  
제작자와 문서의 글은 3개 대쉬 를 갖는 라인 (- - -)

YAML 헤더는 키-값 쌍으로  
제작자와 문서의 글은 3개 대쉬 를 갖는 라인 (- - -)

제작자와 문서의 글은 3개 대쉬 를 갖는 라인 (- - -)

출력: html\_document ..... html 파일 (웹페이지)  
output: pdf\_document ..... pdf 문서  
output: word\_document ..... MS 워드 문서 .docx  
output: beamer\_presentation ..... Beamer 발표자료 (pdf)  
output: ioslides\_presentation ..... 멕 발표자료 (html)



RStudio is a trademark of RStudio, Inc. • CC-BY\_RStudio • info@rstudio.com • 844-448-1212 • rstudio.com  
한글 번역: 이광수 [wwwsooc.net](http://wwwsooc.net) xvMOOC 웹페이지: <http://www.xvmooc.net> CC-BY-SA xvMOOC

## 3. 마크다운

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

### 입력 구문

일반 텍스트  
새로운 단락을 시작하면 줄 미지막을 뒤에 2개로 끝낸다.  
- \*문자 글씨\* and \*기울인 글씨\*  
- \*\*굵은 글씨\*\* and \*\*굵은 글씨\*\*  
-着重선  
-\*加重선\*  
[링크](http://www.rstudio.com)

# 제목 1

# # 제목 2

# # # 제목 3

# # # # 제목 4

# # # # # 제목 5

# # # # # # 제목 6

N자 크기 대시 부호: --

m자 크기 대시 부호: ---

줄이

혹시 사용하는 수식: \$A = \sqrt{r^2 + r^2}

이미지: 

수평선 (혹은 슬라이드 영역):

--

> 인용 편집

\*는 사용하는 목록

\* 항목 2

\* 항목 항목 1

\* 항목 항목 2

1. 순서 있는 목록

2. 항목 2

\* 항목 항목 1

\* 항목 항목 2

표 제작 | 무언가에 제작

표 1 | 표 2

간 3 | 간 4

### 출력 결과

일반 텍스트  
새로운 단락을 시작하면 줄 미지막을 뒤에 2개로 끝낸다.  
- \*문자 글씨\* and \*기울인 글씨\*  
- \*\*굵은 글씨\*\* and \*\*굵은 글씨\*\*  
-着重선  
-\*加重선\*

### 제목 1

### 제목 2

### 제목 3

### 제목 4

### 제목 5

### 제목 6

N자 크기 대시 부호: --

m자 크기 대시 부호: ---

줄이

혹시 사용하는 수식: \$A = \sqrt{r^2 + r^2}

이미지: 

수평선 (혹은 슬라이드 영역):

### 언어 편집

\*는 사용하는 목록

\* 항목 2

\* 항목 항목 1

\* 항목 항목 2

1. 순서 있는 목록

2. 항목 2

\* 항목 항목 1

\* 항목 항목 2

표 제작 | 무언가에 제작

표 1 | 표 2

간 3 | 간 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 더하기 2는 `r 2`  
2 뒤 같다.

Two plus two equals 4.

#### 코드 렁여리

R코드 링여리를 ```(r)으로 시작하고, ```으로 마무리한다.

실행결과는 다음과 같다  
```(r)  
dimiris)  
```

Here's some code  
dimiris)

## [1] 150 5

#### 화면 출력 선택옵션

knitr 선택옵션을 사용해서 R 코드 명령어 출력 스타일을 적용한다.  
코드 상단 글자 내부에 선택옵션을 지정한다.

Here's some code  
(r eval=FALSE)  
dimiris)

Here's some code  
dimiris)

Here's some code  
(r echo=FALSE)  
dimiris)

Here's some code  
dimiris)

## [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 변환한다.
	results	"markup"	"markup", "asis", "hold", "hide"
	cache	FALSE	결과값을 캐시해서 향후 실행시 긴너무게 설정한다.
	comment	"#"	주석문자로 출력결과에 서두를 붙인다.
	fig.width	7	영어로 생성되는 그래프에 대한 폭을 인치로 지정한다.
	fig.height	7	영어로 생성되는 그래프에 대한 높이를 인치로 지정한다.

보다 자세한 사용은 웹사이트를 참조: [yutu.name/knitr](#)

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

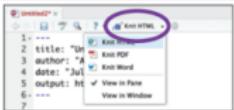
두가지 방식으로 보고서를 렌더링한다.

- 1 rmarkdown::render("(<파일 경로>)") 명령어를 실행한다.

- 2 RStudio 스크립트 작성창 상단에 knit HTML 버튼을 클릭한다.

렌더링 명령을 실행시키면 R은 다음을 수행한다

- 내장된 코드 렁여리를 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 렁여리에, 위젯을 내장하는  
Shiny input 함수를 추가한다. Shiny  
render 함수를 추가해서 반응형 출력  
결과를 내장한다.

```
title: "Line graph"  
output: html_document  
runtime: shiny  
  
Chose a time series:  
```{r echo = FALSE}  
shiny::input("ts", "ts")  
c("c100", "l10")  
```  
See a plot:  
```{r echo = FALSE}  
shiny::renderPlot({  
  d <- getinputdata()  
  plot(d)  
})  
```
```

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

Line graph  
Choose a time series:  
See a plot:

\* 주목: 보고서는 Shiny 웹이 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](#)

##### ShinyApps.io

Studio 서버에 인터랙티브 문서  
를 올려 호스팅한다. 무료와 유료  
선택옵션이 있다.

[www.shinyapps.io](#)



#### 9. 추가 학습

문서와 예제 - [rmarkdown.rstudio.com](#)

주기 기사 - [shiny.rstudio.com/articles](#)

blog.rstudio.com

@rstudio



RStudio® and Shiny™ are trademarks of RStudio, Inc.

CC BY-NC-SA

RStudio info@rstudio.com

## More Reference

1. Rstudio
  - <http://rmarkdown.rstudio.com>
2. Reference (5 pages)
  - <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>
3. LaTex (pdf)
  - [https://www.nyu.edu/projects/beber/files/Chang\\_LaTeX\\_sheet.pdf](https://www.nyu.edu/projects/beber/files/Chang_LaTeX_sheet.pdf)

## Carseats (M22-Carseats 파일 참조)

```
 1  ---  
 2  title: 'Carseat 판매량 '  
 3  author: "Learning Spoons"  
 4  date: "'`r Sys.Date()`'"  
 5  output:  
 6    pdf_document:  
 7      latex_engine: xelatex  
 8      keep_tex: true  
 9      # pandoc_args: [  
10        # "-v", "classoption=twocolumn"  
11        # ]  
12      smaller: true  
13      mainfont: NanumGothic  
14      classoption: a4paper  
15  ---  
16  
17  ```{r setup, include=FALSE}  
18  knitr::opts_chunk$set(echo = TRUE)  
19  ...  
20  
21  ## Carseat 소개  
22  
23  ```{r}  
24  library(ISLR)  
25  library(dplyr)  
26  library(ggplot2)  
27  str(Carseats)  
28  
29  
30  ## Focus city  
31  
32  ```{r}  
33  focuscity <- Carseats %>%  
34  filter(Income > 100) %>%  
35  filter(Age >= 30 & Age < 40) %>%  
36  mutate(AdvPerCapita = Advertising/Population) %>%  
37  select(Sales, Income, Age, Population, Education, AdvPerCapita) %>%  
38  arrange(Sales)  
39  print(focuscity)  
40  
41  ## Income vs Sales  
42  doFacetWrap <- FALSE  
43  a <- ggplot(data = Carseats, aes(x = Income, y = Sales)) +  
44  geom_point(aes(shape = Urban, color = US))  
45  if (doFacetwrap) {  
46    a <- a + facet_wrap(~ floor(Age/10))  
47  }  
48  print(a)  
49  
50  Your comment!  
51  
52  
53  
54  ## Income vs sales  
55  
56  doFacetWrap <- TRUE  
57  a <- ggplot(data = Carseats, aes(x = Income, y = Sales)) +  
58  geom_point(aes(shape = Urban, color = US))  
59  if (doFacetwrap) {  
60    a <- a + facet_wrap(~ floor(Age/10))  
61  }  
62  print(a)  
63  
64  
65  
66
```



## Templates (Ref-rmdTemplates/M8X)

- M81-pdf: 2 columns, line numbering, page numbering
- M82-beamer: Singapore theme, line numbering, page numbering
- M83-slidy
- 주의: 하위폴더(rmd-pdf-template\_files, myRmdBeamerStyle)까지 데리고 다녀야 됨

## Discussion - rmarkdown

- Markdown이라는 Markup Language로 변환하여 사용성 높음
  - 변환 과정
1. Rmd -> md -> docx, html
  2. Rmd -> md -> tex -> pdf
- 약간의 수정으로 여러가지 형식의 문서를 만들 수 있음
  - 자연스럽게 문법에 맞는 Color Coding
  - R output에 가장 비슷한 quality의 output을 얻을 수 있음.
  - 분석 -> 정리의 반복적인 작업을 하나로 줄여줌.

## Discussion - Literature Programming

- Features

1. 프로그래밍이 아닌 글쓰기가 초점
2. 데이터 분석 업무에 연관성이 높음!
3. 글을 쓴다는 것과 프로그래밍을 하는 것은 자기 자신을 표현하는 가장 높은 수준의 지적인 활동
4. 이것을 한꺼번에 할 수 있게 해주는 도구
5. !Express Yourself

- Advantages

1. 코드와 문서가 하나의 파일이라서 관리가 편함
2. 코드에서 주석을 조금만 달아도 됨
3. 코드만 있는 것에 비해서 의사소통이 용이함
4. 데이터나 분석의 결과가 달라지는 것이 문서에 즉각적으로 반영됨
5. 반복적으로 작성하는 문서 작업과 엑셀 작업을 안해도 됨
6. Colin Powell: “You don’t know what you will can get away with until you try”

## pdf 파일 제작을 위한 tex 엔진 설치

### 1. Texlive 2017 설치 (수백메가)

- <http://www.ktug.org/xe/index.php?mid=install>

### 2. 한글 폰트 설치

- google “nanumgothic download”
- google “nanummyeongjo download”
- google “nanumgothiccoding download”

### 3. 참고: google “latex beamer에서 한글 쓰기!”

## 다른 programming language의 문법

```
# C, C++ ↓
for(int i=1; i <=10, i=i+1) { ↓
    printf("value of i: %d\n", i); ↓
} ↓
↓
# MATLAB ↓
for i=1:10 { ↓
    disp("value of i: %d", i); ↓
} ↓
↓
# Python ↓
for i in range(1, 11): ↓
    print "value of i: %d" % (i) ↓
↓
## R ↓
for (i in 1:10) { ↓
    print(paste("value of i:", i)) ↓
}
```

