

# Carseat Sales

LearningSpoonsR

2018-10-13

## 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 책에 포함된 함수와 데이터를 포함한 패키지입니다.
  - dplyr은 R에서 데이터 사이언스의 근간이 되는 `data.frame`을 효율적으로 다루는 패키지입니다.
  - rmarkdown은 R의 결과물을 포함한 문서를 만들어 내는 패키지입니다.

## 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. 엑셀 파일을 바로 불러오는 것도 특정 패키지를 이용하면 가능하고 점점 발전하고 있음.

## 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 객체 (object) 1개를 저장하는 파일

## 데이터 파일 불러오기/저장하기

0. 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=TRUE`를 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 = FALSE, sep = "\t", stringsAsFactors = FALSE)`
- `write.csv → write.table(dataset, "fileName.txt")`

## 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.Rdata")
load("fileName.Rdata")
```

## 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-DS")`을 실행.
- 디렉토리의 입력은 “C:/LS-DS”, “C://LS-DS”, “C:\\LS-DS”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
##  3rd Qu.:398.5  3rd Qu.:131.0
##  Max.   :509.0  Max.   :191.0  Mean   :53.32
##          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
- youtube.com에 keynote등 좋은 동영상 많아요!
- 통계학 박사 후 R에서 다수의 사용하기 좋은 패키지 개발
- 누나도 통계학 전공 교수
- ggplot2, dplyr, reshape2, stringr 등의 본인이 작성한 패키지를 묶어서 tidyverse라는 패키지를 제작

## Basic Manipulation

- tidy data.frame!
  1. 개체 타입은 `data.frame`
  2. 각각의 row는 관찰값을 의미
  3. 각각의 column은 변수를 의미
- Basic Manipulation
  1. `rename`: 변수 이름 바꾸기. column의 이름을 바꿈
  2. `filter`: 관찰값 추출. row를 선택함
  3. `select`: 변수 추출. column을 선택함
  4. `arrange`: 관찰값 정렬. row를 재정렬함
  5. `mutate`: 변수 생성. column을 만듬

## 1. rename (이름 바꾸기)

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

d v  
b

## 2. filter (관찰값 추출, Row 추출)

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

d a > "pipe"  
b

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

- & 는 “and”
- | 는 “or”

### 3. select (변수 추출, Column 선택)

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

d  
d  
b

### 4. arrange (정렬)

```
Carseats <- arrange(Carseats, Price) v d 92%+5
Carseats <- Carseats %>% arrange(Price) v d
Carseats <- Carseats[order(Carseats$Price), ] b
```

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

## 5. mutate (새로운 변수)

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

- 많은 데이터 분석은 “연속 변수”를 “이산 변수”로 변환하여, “이산 변수”의 group별 차이를 파악하는 접근을 사용합니다.
  - Later...
1. Age를 factor로 잡아 Sales의 Boxplot.
  2. Age를 factor로 잡아 scatterplot for Population & Sales.
  3. 변수 3개를 사용하는 습관
  4. 공간과 시간에 대한 동질성과 이질성을 생각하는 습관

## 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 sqldf라는 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

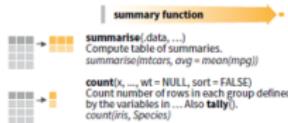


dplyr functions work with pipes and expect tidy data. In tidy data:



### 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

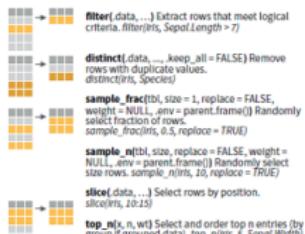
Use `group_by()` to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.



### 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))`

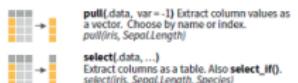
#### ADD CASES

`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.

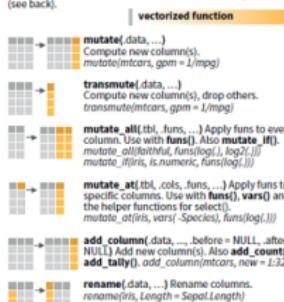


`use these helpers with select(...), e.g. select_if(is, starts_with("Sepal"))`

`contains(match)`    `num_range(prefix, range)`    `ends_with(match)`    `one_of(...)`    `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::ranked() - rank into n bins  
dplyr::percent\_rank() - min rank scaled to [0,1]  
dplyr::row\_number() - rank with ties = "first"

#### MATH

\* - \* / ^ / ^ %/% %% - arithmetic ops  
log(), log2(), log10(), log() --> logical comparisons  
<->, >->, ><->, == - logical comparisons  
dplyr::between() - test if x is between right  
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  
+                 =        
grid          grid      grid

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 ALIKE.

Use a "Mutating Join" to join one table to another. It has two matching columns with the rows that correspond to each. 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 columns 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.

anti\_join(x, y, by = NULL, ...)  
Return rows of x that do not have a match in y.



### COMBINE CASES

x                   y  
+                 =        
grid          grid      grid

Use bind\_rows() to paste tables below each other as they are.

bind\_rows(..., id = NULL)  
Retain table headers and add the other as a single table. Set id to a column name to add a column of the original table names (as pictured).

intersect(x, y, ...)  
Rows that appear in both x and y.

setdiff(x, y, ...)  
Rows that appear in x but not y.

union(x, y, ...)  
Rows that appear in x or y.  
(Duplicates removed). union\_all() retains duplicates.

Use setequal() to test whether two data sets contain the exact same rows (in any order).

### EXTRACT ROWS

x                   y  
+                 =        
grid          grid      grid

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.

anti\_join(x, y, by = NULL, ...)  
Return rows of x that do not have a match in y.



## 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   cla~
##   <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <ch~
## 1 audi         a4     1.8  1999     4 auto~ f      18    29 p   com~
## 2 audi         a4     1.8  1999     4 manu~ f      21    29 p   com~
## 3 audi         a4      2   2008     4 manu~ f      20    31 p   com~
## 4 audi         a4      2   2008     4 auto~ f      21    30 p   com~
## 5 audi         a4     2.8  1999     6 auto~ f      16    26 p   com~
## 6 audi         a4     2.8  1999     6 manu~ f      18    26 p   com~
## 7 audi         a4     3.1  2008     6 auto~ f      18    27 p   com~
## 8 audi         a4 q~  1.8  1999     4 manu~ 4     18    26 p   com~
## 9 audi         a4 q~  1.8  1999     4 auto~ 4     16    25 p   com~
## 10 audi        a4 q~   2   2008     4 manu~ 4    20    28 p   com~
## # ... with 224 more rows
class(mpg)

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

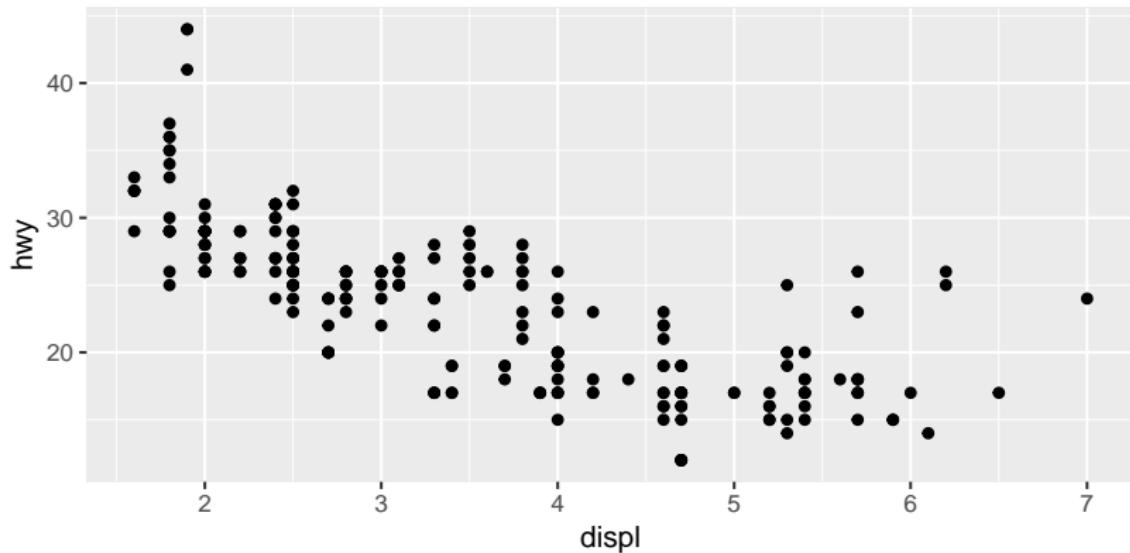
```

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

엔진이 크면 연비가 안 좋을까요?

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

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



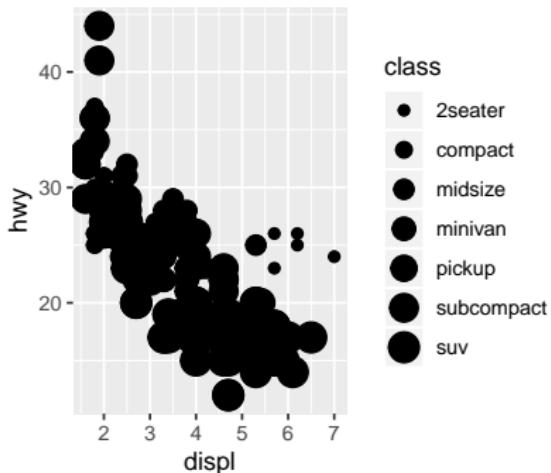
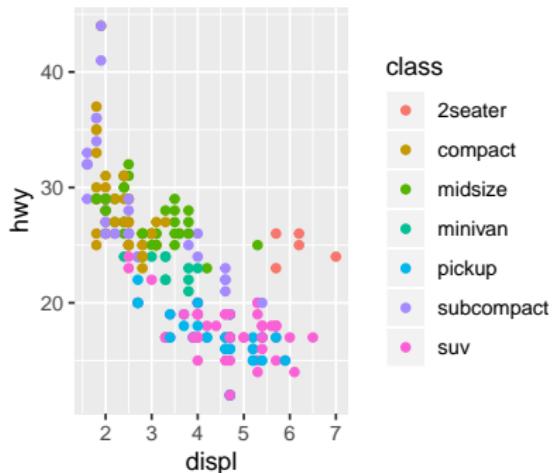
- 엔진이 크면 연비가 안 좋나요?
- 추가적으로 궁금한게 있나요?

```
# 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)

```

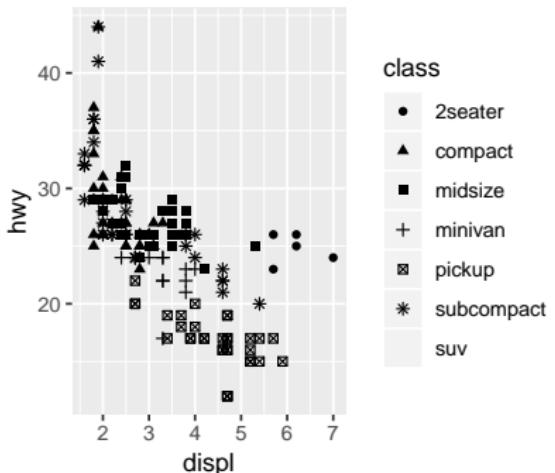
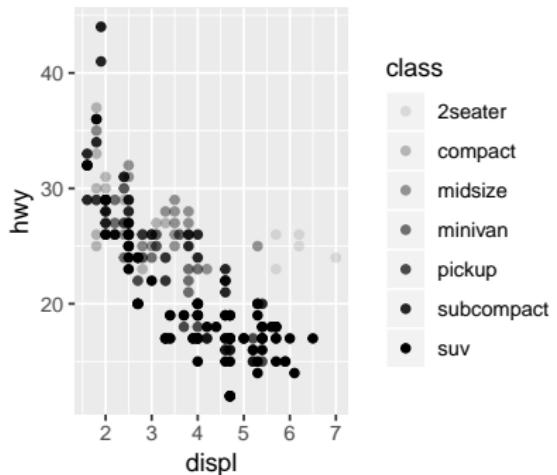


```

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: Using alpha for a discrete variable is not advised.
## 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?

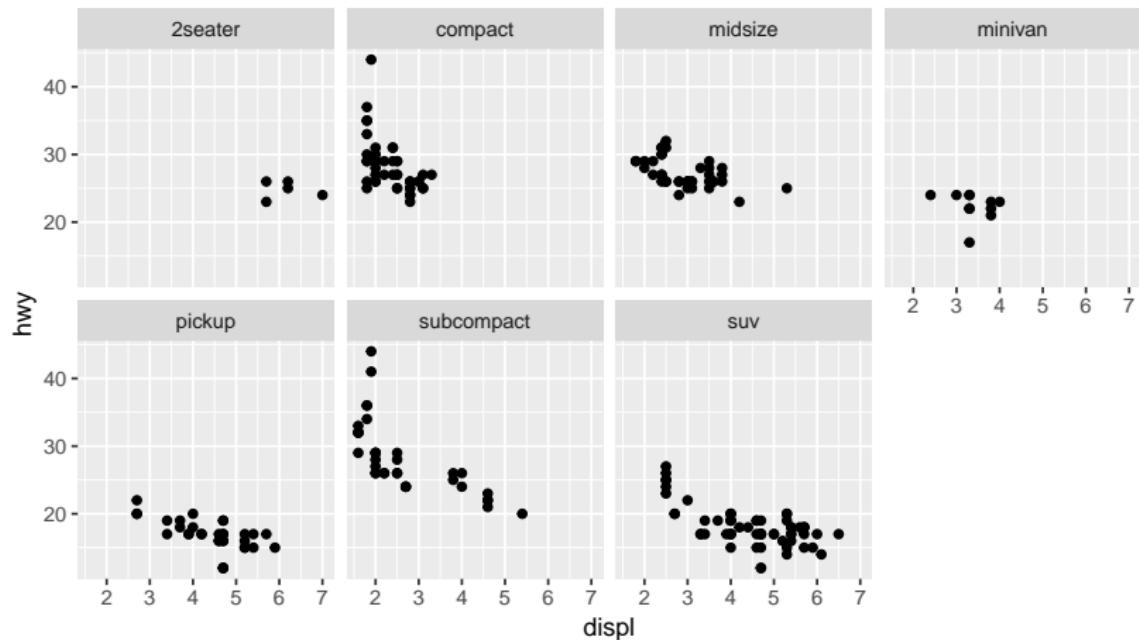
## Aesthetics (aes) - size, alpha, color, shape

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

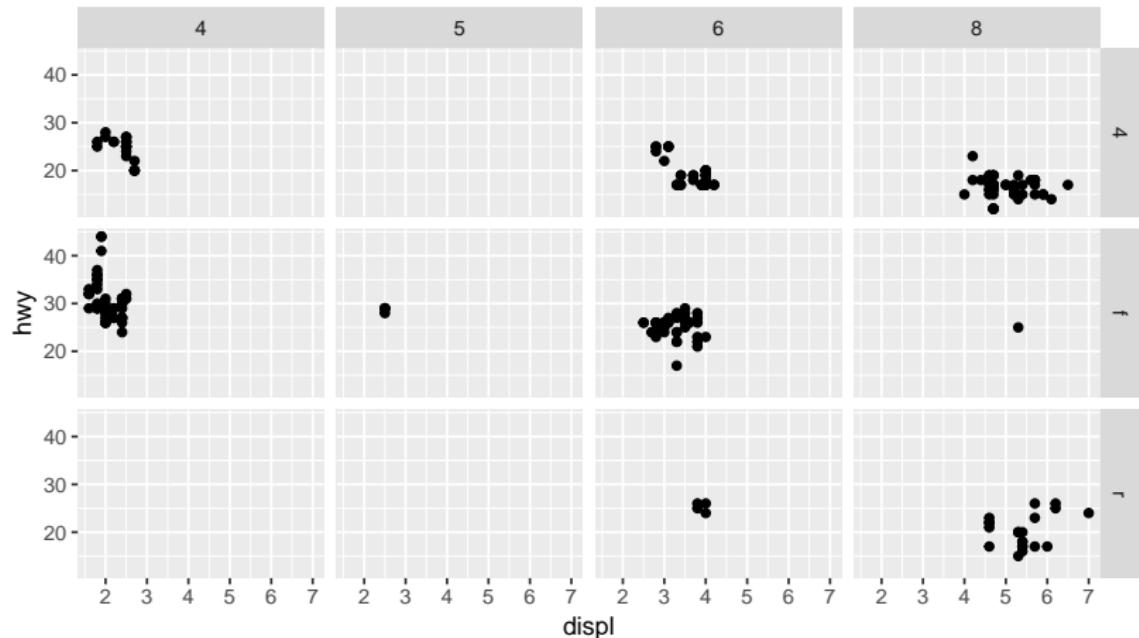
## Facets (1 variable) - facet\_wrap

```
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)
```

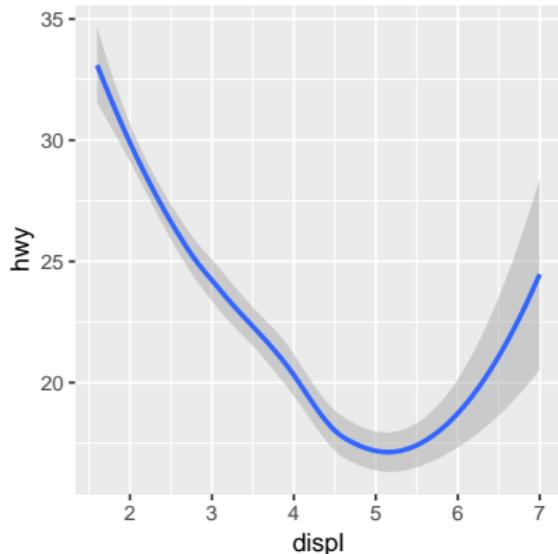
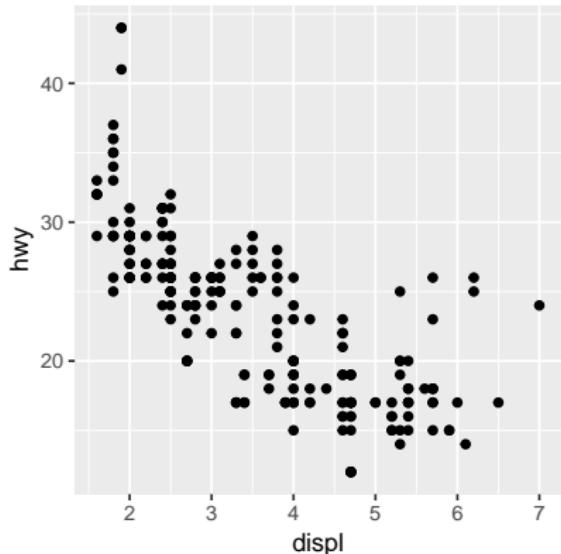


## Discussion

- When or not to facet?
1. Data가 충분히 많을 때 -> do facet
  2. Distribution 자체를 보고 싶을 때 -> do facet
  3. 근접 비교를 하고 싶을 때 -> don't facet
  4. factor 변수의 갯수를 고려해야 함 -> **magic number**

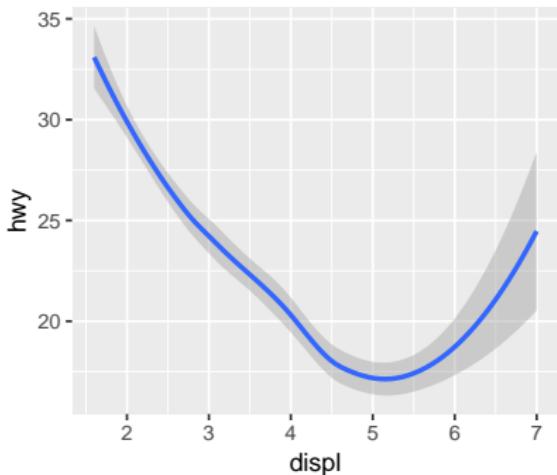
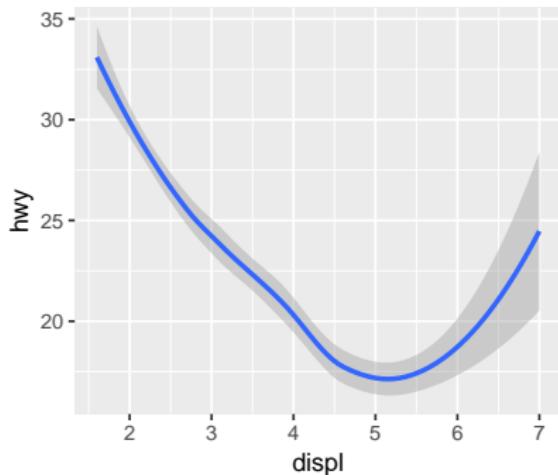
## Point vs Smooth

```
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

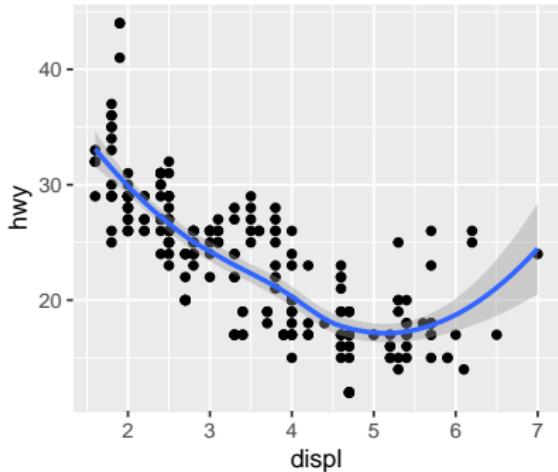
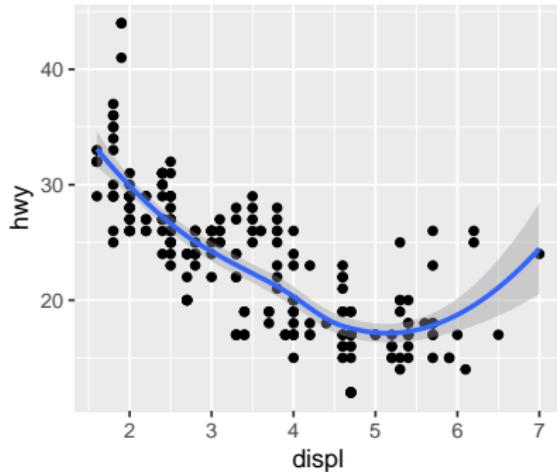
```
g <- ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
h <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth()
grid.arrange(g, h, nrow = 1, ncol = 2)
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



## 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' and formula 'y ~ x'
```

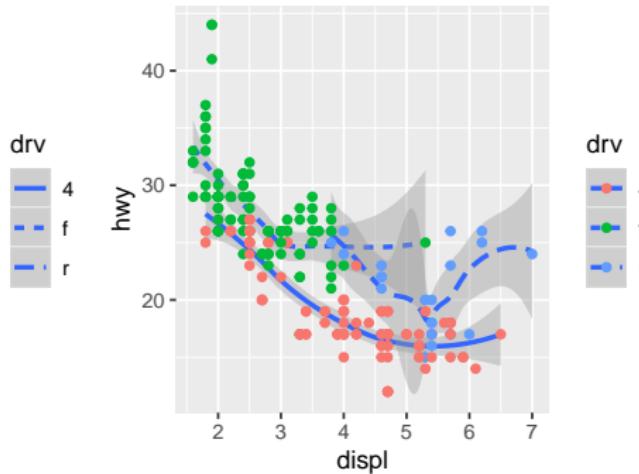
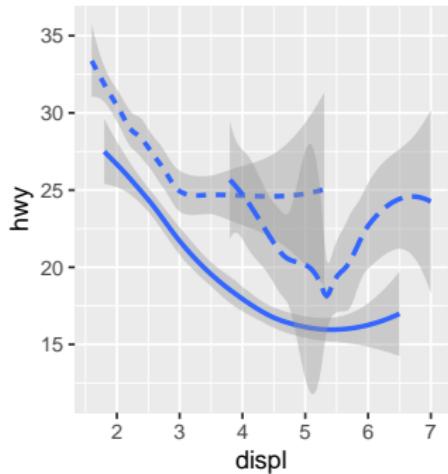


```

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' and formula 'y ~ x'

```



## Summary - ggplot

- Data, Aesthetic, Geometric Object를 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>))
```

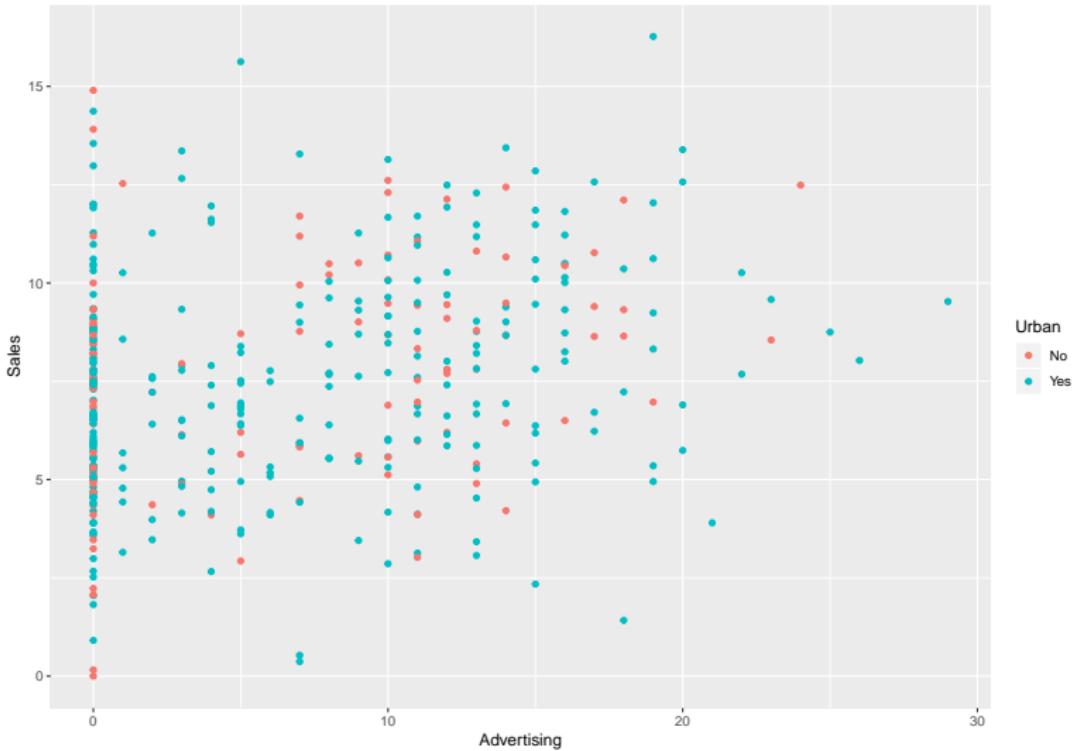
1. 그림의 목적을 정하고
2. X (explanatory variable - 설명 변수)의 개수와 각 변수의 중요도와 순서
3. Y (dependent variable)를 명시
4. ggplot 객체를 rendering

## Carseat

```
str(Carseats)

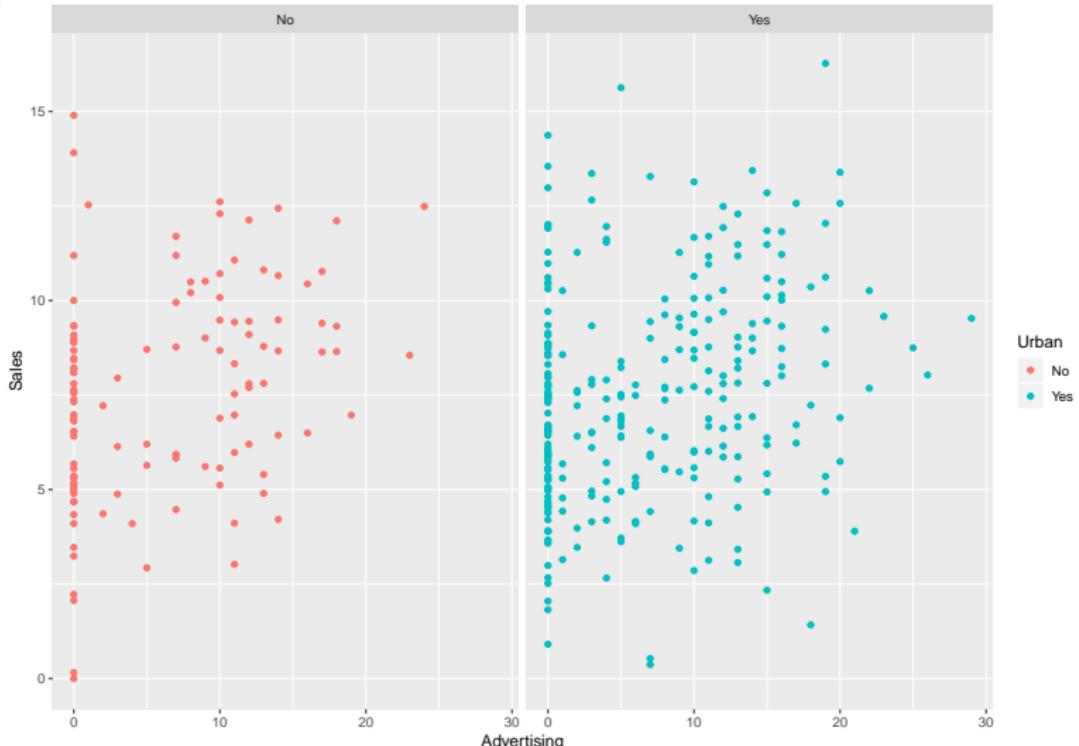
## 'data.frame': 400 obs. of 15 variables:
## $ Sales      : num  0.37 0 6.67 6.39 5.01 9.53 3.58 5.4 6.53 2.93 ...
## $ CompPrice   : num  147 139 156 131 159 175 142 149 154 143 ...
## $ Income      : num  58 24 42 21 69 65 109 73 30 21 ...
## $ Advertising : num  7 0 13 8 0 29 0 13 0 5 ...
## $ Population   : 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 = Advertising, y = Sales)) +  
  geom_point(aes(color = Urban))  
print(a)
```



## Amazingly Additive

```
a <- a + facet_wrap(~ Urban)  
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)
```

```
doAgeFacet <- FALSE  
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

 OECD non-OECD

Continents

 Asia Europe Africa North America South America Oceania

Range of Country GDP (USD)

 10,000

100,000

1,000,000

10,000,000

100,000,000

1,000,000,000

10,000,000,000

100,000,000,000

Life Expectancy at Birth in 2016

Indonesia

Costa Rica

Poland

Estonia

Spain

South Korea

Japan

France

Croatia

New Zealand

Brazil

Iceland

Denmark

Australia

Netherlands

Sweden

Norway

Switzerland

Luxembourg

Ireland

Denmark

Norway

Switzerland

Luxembourg





## More Reference

### 1. ggplot 관련 서적

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

### 2. ggplot의 50개 예제

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

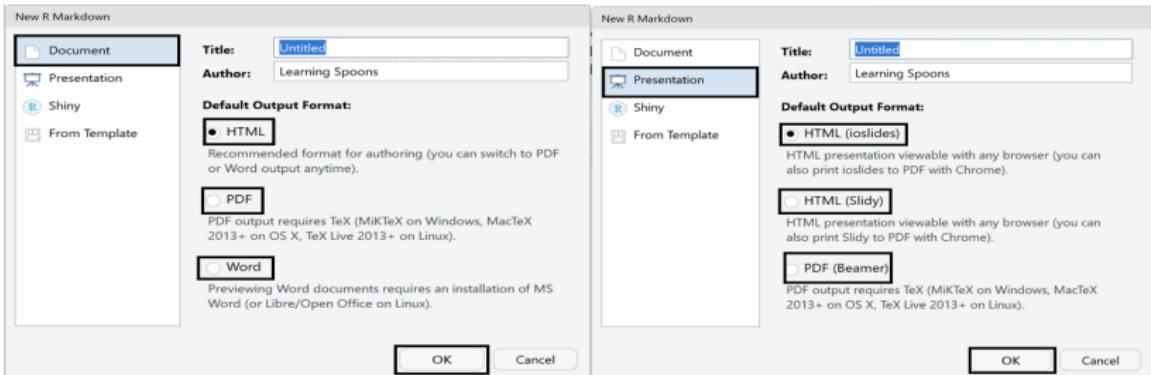
### 3. ggplot의 10주년을 기념하는 article

- [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 a.Rmd 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 #  
언제 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 R Markdown
```

Console Terminal R Markdown

Untitled

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

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. :129.00

언제 R코드는 2회 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 - 1-10}
42 rmd pdf template :
```

페이지: 1 / 1

rmd pdf template

Learning Spoons  
2018-06-30

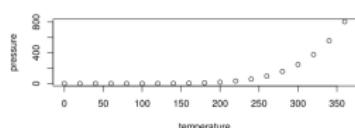
R Markdown 한글

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.	5.0
1st Qu.	12.0
Median	15.0
3rd Qu.	19.0
Max.	28.0
2nd Qu.	16.0
Mean	17.8
St. Dev.	7.58

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 screenshot shows the RStudio interface with an R Markdown file open. The code editor on the left contains the following R Markdown 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 R Markdown tab in the bottom navigation bar is selected. To the right, the Microsoft Word application is open, displaying the generated Word document. The document has a title "Untitled", a subtitle "Learning Spoons", and a date "2018년 4월 17일". It includes a section titled "R Markdown" which contains the text: "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>.". Below this, there is a code block with the R code from line 18 and its 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
```

The Microsoft Word ribbon at the top includes tabs for 파일 (File), 품 (Format), 삽입 (Insert), 그리기 (Drawing), 디자인 (Design), 레이아웃 (Layout), 참조 (References), 편집 (Editing), 검토 (Review), 보기 (View), 개발 도구 (Developer Tools), Acrobat, and 입력하세 (Input Method). The status bar at the bottom shows "1/2 페이지", "151개 단어", "영어(미국)", and "81%".

# ioslides

title: "Untitled"  
author: "Learning Spoons"  
date: "2018! 4월 17일"  
output: ioslides\_presentation  
---

```
```{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)
```
## slide with plot
```

R Markdown

Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

Slide with R Output

```
summary(cars)
```

| car | speed | dist |
|-----|-------|------|
| mpg | 1.8   | 40   |
| mpg | 1.8   | 111  |
| mpg | 1.8   | 120  |
| mpg | 1.8   | 130  |
| mpg | 1.8   | 140  |
| mpg | 1.8   | 150  |
| mpg | 1.8   | 160  |
| mpg | 1.8   | 170  |
| mpg | 1.8   | 180  |
| mpg | 1.8   | 190  |
| mpg | 1.8   | 200  |
| mpg | 1.8   | 210  |
| mpg | 1.8   | 220  |
| mpg | 1.8   | 230  |
| mpg | 1.8   | 240  |
| mpg | 1.8   | 250  |
| mpg | 1.8   | 260  |
| mpg | 1.8   | 270  |
| mpg | 1.8   | 280  |
| mpg | 1.8   | 290  |
| mpg | 1.8   | 300  |
| mpg | 1.8   | 310  |
| mpg | 1.8   | 320  |
| mpg | 1.8   | 330  |
| mpg | 1.8   | 340  |
| mpg | 1.8   | 350  |
| mpg | 1.8   | 360  |
| mpg | 1.8   | 370  |
| mpg | 1.8   | 380  |
| mpg | 1.8   | 390  |
| mpg | 1.8   | 400  |
| mpg | 1.8   | 410  |
| mpg | 1.8   | 420  |
| mpg | 1.8   | 430  |
| mpg | 1.8   | 440  |
| mpg | 1.8   | 450  |
| mpg | 1.8   | 460  |
| mpg | 1.8   | 470  |
| mpg | 1.8   | 480  |
| mpg | 1.8   | 490  |
| mpg | 1.8   | 500  |
| mpg | 1.8   | 510  |
| mpg | 1.8   | 520  |
| mpg | 1.8   | 530  |
| mpg | 1.8   | 540  |
| mpg | 1.8   | 550  |
| mpg | 1.8   | 560  |
| mpg | 1.8   | 570  |
| mpg | 1.8   | 580  |
| mpg | 1.8   | 590  |
| mpg | 1.8   | 600  |
| mpg | 1.8   | 610  |
| mpg | 1.8   | 620  |
| mpg | 1.8   | 630  |
| mpg | 1.8   | 640  |
| mpg | 1.8   | 650  |
| mpg | 1.8   | 660  |
| mpg | 1.8   | 670  |
| mpg | 1.8   | 680  |
| mpg | 1.8   | 690  |
| mpg | 1.8   | 700  |
| mpg | 1.8   | 710  |
| mpg | 1.8   | 720  |
| mpg | 1.8   | 730  |
| mpg | 1.8   | 740  |
| mpg | 1.8   | 750  |
| mpg | 1.8   | 760  |
| mpg | 1.8   | 770  |
| mpg | 1.8   | 780  |
| mpg | 1.8   | 790  |
| mpg | 1.8   | 800  |
| mpg | 1.8   | 810  |
| mpg | 1.8   | 820  |
| mpg | 1.8   | 830  |
| mpg | 1.8   | 840  |
| mpg | 1.8   | 850  |
| mpg | 1.8   | 860  |
| mpg | 1.8   | 870  |
| mpg | 1.8   | 880  |
| mpg | 1.8   | 890  |
| mpg | 1.8   | 900  |
| mpg | 1.8   | 910  |
| mpg | 1.8   | 920  |
| mpg | 1.8   | 930  |
| mpg | 1.8   | 940  |
| mpg | 1.8   | 950  |
| mpg | 1.8   | 960  |
| mpg | 1.8   | 970  |
| mpg | 1.8   | 980  |
| mpg | 1.8   | 990  |
| mpg | 1.8   | 1000 |
| mpg | 1.8   | 1010 |
| mpg | 1.8   | 1020 |
| mpg | 1.8   | 1030 |
| mpg | 1.8   | 1040 |
| mpg | 1.8   | 1050 |
| mpg | 1.8   | 1060 |
| mpg | 1.8   | 1070 |
| mpg | 1.8   | 1080 |
| mpg | 1.8   | 1090 |
| mpg | 1.8   | 1100 |
| mpg | 1.8   | 1110 |
| mpg | 1.8   | 1120 |
| mpg | 1.8   | 1130 |
| mpg | 1.8   | 1140 |
| mpg | 1.8   | 1150 |
| mpg | 1.8   | 1160 |
| mpg | 1.8   | 1170 |
| mpg | 1.8   | 1180 |
| mpg | 1.8   | 1190 |
| mpg | 1.8   | 1200 |
| mpg | 1.8   | 1210 |
| mpg | 1.8   | 1220 |
| mpg | 1.8   | 1230 |
| mpg | 1.8   | 1240 |
| mpg | 1.8   | 1250 |
| mpg | 1.8   | 1260 |
| mpg | 1.8   | 1270 |
| mpg | 1.8   | 1280 |
| mpg | 1.8   | 1290 |
| mpg | 1.8   | 1300 |
| mpg | 1.8   | 1310 |
| mpg | 1.8   | 1320 |
| mpg | 1.8   | 1330 |
| mpg | 1.8   | 1340 |
| mpg | 1.8   | 1350 |
| mpg | 1.8   | 1360 |
| mpg | 1.8   | 1370 |
| mpg | 1.8   | 1380 |
| mpg | 1.8   | 1390 |
| mpg | 1.8   | 1400 |
| mpg | 1.8   | 1410 |
| mpg | 1.8   | 1420 |
| mpg | 1.8   | 1430 |
| mpg | 1.8   | 1440 |
| mpg | 1.8   | 1450 |
| mpg | 1.8   | 1460 |
| mpg | 1.8   | 1470 |
| mpg | 1.8   | 1480 |
| mpg | 1.8   | 1490 |
| mpg | 1.8   | 1500 |
| mpg | 1.8   | 1510 |
| mpg | 1.8   | 1520 |
| mpg | 1.8   | 1530 |
| mpg | 1.8   | 1540 |
| mpg | 1.8   | 1550 |
| mpg | 1.8   | 1560 |
| mpg | 1.8   | 1570 |
| mpg | 1.8   | 1580 |
| mpg | 1.8   | 1590 |
| mpg | 1.8   | 1600 |
| mpg | 1.8   | 1610 |
| mpg | 1.8   | 1620 |
| mpg | 1.8   | 1630 |
| mpg | 1.8   | 1640 |
| mpg | 1.8   | 1650 |
| mpg | 1.8   | 1660 |
| mpg | 1.8   | 1670 |
| mpg | 1.8   | 1680 |
| mpg | 1.8   | 1690 |
| mpg | 1.8   | 1700 |
| mpg | 1.8   | 1710 |
| mpg | 1.8   | 1720 |
| mpg | 1.8   | 1730 |
| mpg | 1.8   | 1740 |
| mpg | 1.8   | 1750 |
| mpg | 1.8   | 1760 |
| mpg | 1.8   | 1770 |
| mpg | 1.8   | 1780 |
| mpg | 1.8   | 1790 |
| mpg | 1.8   | 1800 |
| mpg | 1.8   | 1810 |
| mpg | 1.8   | 1820 |
| mpg | 1.8   | 1830 |
| mpg | 1.8   | 1840 |
| mpg | 1.8   | 1850 |
| mpg | 1.8   | 1860 |
| mpg | 1.8   | 1870 |
| mpg | 1.8   | 1880 |
| mpg | 1.8   | 1890 |
| mpg | 1.8   | 1900 |
| mpg | 1.8   | 1910 |
| mpg | 1.8   | 1920 |
| mpg | 1.8   | 1930 |
| mpg | 1.8   | 1940 |
| mpg | 1.8   | 1950 |
| mpg | 1.8   | 1960 |
| mpg | 1.8   | 1970 |
| mpg | 1.8   | 1980 |
| mpg | 1.8   | 1990 |
| mpg | 1.8   | 2000 |
| mpg | 1.8   | 2010 |
| mpg | 1.8   | 2020 |
| mpg | 1.8   | 2030 |
| mpg | 1.8   | 2040 |
| mpg | 1.8   | 2050 |
| mpg | 1.8   | 2060 |
| mpg | 1.8   | 2070 |
| mpg | 1.8   | 2080 |
| mpg | 1.8   | 2090 |
| mpg | 1.8   | 2100 |
| mpg | 1.8   | 2110 |
| mpg | 1.8   | 2120 |
| mpg | 1.8   | 2130 |
| mpg | 1.8   | 2140 |
| mpg | 1.8   | 2150 |
| mpg | 1.8   | 2160 |
| mpg | 1.8   | 2170 |
| mpg | 1.8   | 2180 |
| mpg | 1.8   | 2190 |
| mpg | 1.8   | 2200 |
| mpg | 1.8   | 2210 |
| mpg | 1.8   | 2220 |
| mpg | 1.8   | 2230 |
| mpg | 1.8   | 2240 |
| mpg | 1.8   | 2250 |
| mpg | 1.8   | 2260 |
| mpg | 1.8   | 2270 |
| mpg | 1.8   | 2280 |
| mpg | 1.8   | 2290 |
| mpg | 1.8   | 2300 |
| mpg | 1.8   | 2310 |
| mpg | 1.8   | 2320 |
| mpg | 1.8   | 2330 |
| mpg | 1.8   | 2340 |
| mpg | 1.8   | 2350 |
| mpg | 1.8   | 2360 |
| mpg | 1.8   | 2370 |
| mpg | 1.8   | 2380 |
| mpg | 1.8   | 2390 |
| mpg | 1.8   | 2400 |
| mpg | 1.8   | 2410 |
| mpg | 1.8   | 2420 |
| mpg | 1.8   | 2430 |
| mpg | 1.8   | 2440 |
| mpg | 1.8   | 2450 |
| mpg | 1.8   | 2460 |
| mpg | 1.8   | 2470 |
| mpg | 1.8   | 2480 |
| mpg | 1.8   | 2490 |
| mpg | 1.8   | 2500 |
| mpg | 1.8   | 2510 |
| mpg | 1.8   | 2520 |
| mpg | 1.8   | 2530 |
| mpg | 1.8   | 2540 |
| mpg | 1.8   | 2550 |
| mpg | 1.8   | 2560 |
| mpg | 1.8   | 2570 |
| mpg | 1.8   | 2580 |
| mpg | 1.8   | 2590 |
| mpg | 1.8   | 2600 |
| mpg | 1.8   | 2610 |
| mpg | 1.8   | 2620 |
| mpg | 1.8   | 2630 |
| mpg | 1.8   | 2640 |
| mpg | 1.8   | 2650 |
| mpg | 1.8   | 2660 |
| mpg | 1.8   | 2670 |
| mpg | 1.8   | 2680 |
| mpg | 1.8   | 2690 |
| mpg | 1.8   | 2700 |
| mpg | 1.8   | 2710 |
| mpg | 1.8   | 2720 |
| mpg | 1.8   | 2730 |
| mpg | 1.8   | 2740 |
| mpg | 1.8   | 2750 |
| mpg | 1.8   | 2760 |
| mpg | 1.8   | 2770 |
| mpg | 1.8   | 2780 |
| mpg | 1.8   | 2790 |
| mpg | 1.8   | 2800 |
| mpg | 1.8   | 2810 |
| mpg | 1.8   | 2820 |
| mpg | 1.8   | 2830 |
| mpg | 1.8   | 2840 |
| mpg | 1.8   | 2850 |
| mpg | 1.8   | 2860 |
| mpg | 1.8   | 2870 |
| mpg | 1.8   | 2880 |
| mpg | 1.8   | 2890 |
| mpg | 1.8   | 2900 |
| mpg | 1.8   | 2910 |
| mpg | 1.8   | 2920 |
| mpg | 1.8   | 2930 |
| mpg | 1.8   | 2940 |
| mpg | 1.8   | 2950 |
| mpg | 1.8   | 2960 |
| mpg | 1.8   | 2970 |
| mpg | 1.8   | 2980 |
| mpg | 1.8   | 2990 |
| mpg | 1.8   | 3000 |
| mpg | 1.8   | 3010 |
| mpg | 1.8   | 3020 |
| mpg | 1.8   | 3030 |
| mpg | 1.8   | 3040 |
| mpg | 1.8   | 3050 |
| mpg | 1.8   | 3060 |
| mpg | 1.8   | 3070 |
| mpg | 1.8   | 3080 |
| mpg | 1.8   | 3090 |
| mpg | 1.8   | 3100 |
| mpg | 1.8   | 3110 |
| mpg | 1.8   | 3120 |
| mpg | 1.8   | 3130 |
| mpg | 1.8   | 3140 |
| mpg | 1.8   | 3150 |
| mpg | 1.8   | 3160 |
| mpg | 1.8   | 3170 |
| mpg | 1.8   | 3180 |
| mpg | 1.8   | 3190 |
| mpg | 1.8   | 3200 |
| mpg | 1.8   | 3210 |
| mpg | 1.8   | 3220 |
| mpg | 1.8   | 3230 |
| mpg | 1.8   | 3240 |
| mpg | 1.8   | 3250 |
| mpg | 1.8   | 3260 |
| mpg | 1.8   | 3270 |
| mpg | 1.8   | 3280 |
| mpg | 1.8   | 3290 |
| mpg | 1.8   | 3300 |
| mpg | 1.8   | 3310 |
| mpg | 1.8   | 3320 |
| mpg | 1.8   | 3330 |
| mpg | 1.8   | 3340 |
| mpg | 1.8   | 3350 |
| mpg | 1.8   | 3360 |
| mpg | 1.8   | 3370 |
| mpg | 1.8   | 3380 |
| mpg | 1.8   | 3390 |
| mpg | 1.8   | 3400 |
| mpg | 1.8   | 3410 |
| mpg | 1.8   | 3420 |
| mpg | 1.8   | 3430 |
| mpg | 1.8   | 3440 |
| mpg | 1.8   | 3450 |
| mpg | 1.8   | 3460 |
| mpg | 1.8   | 3470 |
| mpg | 1.8   | 3480 |
| mpg | 1.8   | 3490 |
| mpg | 1.8   | 3500 |
| mpg | 1.8   | 3510 |
| mpg | 1.8   | 3520 |
| mpg | 1.8   | 3530 |
| mpg | 1.8   | 3540 |
| mpg | 1.8   | 3550 |
| mpg | 1.8   | 3560 |
| mpg | 1.8   | 3570 |
| mpg | 1.8   | 3580 |
| mpg | 1.8   | 3590 |
| mpg | 1.8   | 3600 |
| mpg | 1.8   | 3610 |
| mpg | 1.8   | 3620 |
| mpg | 1.8   | 3630 |
| mpg | 1.8   | 3640 |
| mpg | 1.8   | 3650 |
| mpg | 1.8   | 3660 |
| mpg | 1.8   | 3670 |
| mpg | 1.8   | 3680 |
| mpg | 1.8   | 3690 |
| mpg | 1.8   | 3700 |
| mpg | 1.8   | 3710 |
| mpg | 1.8   | 3720 |
| mpg | 1.8   | 3730 |
| mpg | 1.8   | 3740 |
| mpg | 1.8   | 3750 |
| mpg | 1.8   | 3760 |
| mpg | 1.8   | 3770 |
| mpg | 1.8   | 3780 |
| mpg | 1.8   | 3790 |
| mpg | 1.8   | 3800 |
| mpg | 1.8   | 3810 |
| mpg | 1.8   | 3820 |
| mpg | 1.8   | 3830 |
| mpg | 1.8   | 3840 |
| mpg | 1.8   | 3850 |
| mpg | 1.8   | 3860 |
| mpg | 1.8   | 3870 |
| mpg | 1.8   | 3880 |
| mpg | 1.8   | 3890 |
| mpg | 1.8   | 3900 |
| mpg | 1.8   | 3910 |
| mpg | 1.8   | 3920 |
| mpg | 1.8   | 3930 |
| mpg | 1.8   | 3940 |
| mpg | 1.8   | 3950 |
| mpg | 1.8   | 3960 |
| mpg | 1.8   | 3970 |
| mpg | 1.8   | 3980 |
| mpg | 1.8   | 3990 |
| mpg | 1.8   | 4000 |
| mpg | 1.8   | 4010 |
| mpg | 1.8   | 4020 |
| mpg | 1.8   | 4030 |
| mpg | 1.8   | 4040 |
| mpg | 1.8   | 4050 |
| mpg | 1.8   | 4060 |
| mpg | 1.8   | 4070 |
| mpg | 1.8   | 4080 |
| mpg | 1.8   | 4090 |
| mpg | 1.8   | 4100 |
| mpg | 1.8   | 4110 |
| mpg | 1.8   | 4120 |
| mpg | 1.8   | 4130 |
| mpg | 1.8   | 4140 |
| mpg | 1.8   | 4150 |
| mpg | 1.8   | 4160 |
| mpg | 1.8   | 4170 |
| mpg | 1.8   | 4180 |
| mpg | 1.8   | 4190 |
| mpg | 1.8   | 4200 |
| mpg | 1.8   | 4210 |
| mpg | 1.8   | 4220 |
| mpg | 1.8   | 4230 |
| mpg | 1.8   | 4240 |
| mpg | 1.8   | 4250 |
| mpg | 1.8   | 4260 |
| mpg | 1.8   | 4270 |
| mpg | 1.8   | 4280 |
| mpg | 1.8   | 4290 |
| mpg | 1.8   | 4300 |
| mpg | 1.8   | 4310 |
| mpg | 1.8   | 4320 |
| mpg | 1.8   | 4330 |
| mpg | 1.8   | 4340 |
| mpg | 1.8   | 4350 |
| mpg | 1.8   | 4360 |
| mpg | 1.8   | 4370 |
| mpg | 1.8   | 4380 |
| mpg | 1.8   | 4390 |
| mpg | 1.8   | 4400 |
| mpg | 1.8   | 4410 |

# 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. The code defines a Beamer presentation template with various options like theme, font size, and LaTeX engine. It also includes a header section and a code chunk for generating plots. The Beamer viewer on the right displays the presentation's design, showing two columns and a plot.

```
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: myRmdBeamerStyle/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 페키지 (r 페키지가 아니라 tex 페키지) \br  
34  - 'tex' 설정들의 작업이 되었어야 이 템플릿을 사용할 수 있습니다. \br  
35  - 하위 폴더인 'myRmdBeamerStyle' 이 있고, 폴더안에는 'latex-topmatter.tex' 와 'markdown_cus...  
36
```

37 \* ## Design  
38  
39 1. 헤더 부분의 아래 명령어를 검색해서 바꾸시면 다른 색상  
40 + 'theme: "Singapore"'  
41 + 'colortheme: "beaver"'  
42 + 'knitr::opts\_chunk\$set(background = '71B0CA')'  
43  
44 \* ## 2단 구성  
45  
46 \lc  
47  
48 1. Left Column  
49 - 이 템플릿은 2단 구성이 가능합니다. \br  
50 - 'MSX-tidyR'의 강의노트가 beamer로 만든 좋은 예제입니다  
51  
52 \rc  
53  
54 2. Right Column  
55  
56 \*\*\*{r}  
57 plot(1:10)  
58  
59  
60 \rc  
61  
62 \* ## Code line number  
63  
64 \*\*\*{#numCode .R .numberLines}  
65 x <- 1:10  
66 y <- x^2  
67 plot(x,y)  
68  
69  
70 \*\*\*{r results='asis', echo=FALSE}  
71 x <- 1:10  
72 y <- x^2  
73 plot(x,y)  
74

# Example - 1주차 숙제

The screenshot shows the RStudio interface with two panes. The left pane displays the R Markdown code, and the right pane shows the rendered HTML output.

**R Markdown 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
```

**R Markdown Output:**

Review - Week1

learningSpoonsR

2018년 4월 15일

• **Module 1 - Hello World**

**Problem 1.**

```
a <- "Hello";
a;
ans;
```

**Problem 2.**

```
a <- "Hello";
b <- "World";
paste(a,b);
ans;
```

**Problem 3.**

```
ans;
```

**Problem 4.**

```
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 to understand the project.

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

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

## Workflow



- Open a new .Rmd file
- Run the code that opens to preview the document
- Write document by editing template
- Knit document to create report; use knit button or render() to knit
- Preview Output in IDE window
- Publish (optional) to web server
- Execute build log: If R Markdown console
- Save 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 `r(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)  
`child.path` - collapse all output into single block (default = FALSE)  
`comment` - prefix for each line of results (default = "#")

**CODE CHUNKS**  
One or more lines surrounded with ````{r}` and `````. Place chunk options with curly braces, after r. Insert with `knitr::knitWithCaching()` or `knitr::knitWithCaching()`.

| arg                       | description  |
|---------------------------|--|
| <code>fig_align</code>    | align: left, right, or center (default = "center")   |
| <code>fig_captions</code> | display code in output document (default = TRUE)   |
| <code>engine</code>       | code language used in chunk (default = NULL)   |
| <code>error</code>        | display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = TRUE)   |
| <code>eval</code>         | run code in chunk (default = TRUE)   |
| <code>options</code>      | options not listed above (e.g., args, engine, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpo, engine.opts, engine.path, fig.align, fig.cap, fig.height, fig.out, fig.path, fig.procsize, fig.refin, fig.scap, fig.show, fig.size, fig.subcap, interval, out.extra, out.height, out.width, print, plot, pdf, rtf, rmarkdown, render, rmd, split, tidy.opts) |

**GLOBAL OPTIONS**  
Set with `knitr::opts_chunkSet()`, e.g.  
````{r include=FALSE}`  
`knitr::opts_chunkSet( echo=TRUE )`



RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844 448 2322 • rstudio.com • Learn more at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com) • rmarkdown 2.6 • Updated: 2016-02

Annotations:

- Non all previous chunks: A callout pointing to the "non all previous chunks" link in the "Run Previous Chunks" dropdown.
- Run current chunk: A callout pointing to the "Run current chunk" link in the "Run Previous Chunks" dropdown.
- Run previous chunk: A callout pointing to the "Run previous chunk" link in the "Run Previous Chunks" dropdown.
- Run all chunks: A callout pointing to the "Run all chunks" link in the "Run Previous Chunks" dropdown.
- Run all previous chunks: A callout pointing to the "Run all previous chunks" link in the "Run Previous Chunks" dropdown.

## .rmd Structure



### .rmd Header

Options for creation of render (e.g. predict) options written as keyvalue pairs (YAML).

At start of file

Between lines of ...

Narration formatted with markdown, mixed with:

**Code Chunks**

Code blocks embedded code. Each chunk:

Begins with ````{r}`

ends with `````

R Markdown will run the code and append the results to the doc.

It will use the location of the .Rmd file as the working directory

## Parameters

Parameterize your documents to reuse with different inputs (e.g. data, values, etc.)

**Add parameters**: Create and set parameters in the header as sub-values of `params`:

**2. Call parameter**: Call parameter value in code as `param$parameter`

**3. Set parameters**: Set values with `Knit with parameters` or the `params` argument in `knitWithCaching()` or `knitWithR()`:

````{r, params = list(param1 = 1, param2 = 2)}`

**4. Render with rmarkdowncrafn** or click Run Document in RStudio IDE

## Interactive Documents

Turn your report into an interactive Shiny document in 4 steps:

1. Add `library(shiny)` to the `www` header.

2. Call `Shiny input functions` to embed input objects.

3. Call `Shiny render functions` to embed reactive output.

4. Render with `rmarkdowncrafn` or click Run Document in RStudio IDE

How many cars?

| spend | dist       |
|-------|------------|
| 1     | 4.00 3.00  |
| 2     | 4.00 10.00 |
| 3     | 9.00 4.00  |
| 4     | 7.00 22.00 |
| 5     | 8.00 16.00 |



Embed a complete app into your document with `shiny::shinyApp()`

NOTE: Your report will render as a Shiny app, which means you must choose an `html` output format (like `html_document`) and serve it with an active R session.







## 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 Carseat.pdfRmd x
 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 doFacetWrap <- TRUE
56 a <- ggplot(data = Carseats, aes(x = Income, y = Sales)) +
57   geom_point(aes(shape = Urban, color = US))
58 if (doFacetWrap) {
59   a <- a + facet_wrap(~ floor(Age/10))
60 }
61 print(a)
62
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)) ↓
}
```

## 실습과제

1. C:/LS-DS/classProject라는 폴더를 만드세요.
2. Rstudio를 열어서 rmarkdown의 html형식에 해당하는 classProject1.Rmd파일을 만들어 위 폴더에 저장하세요.
3. Rstudio를 닫고 classProject1.Rmd를 더블클릭하여 실행하면 working directory가 자동으로 위의 폴더가 됩니다.
4. github에서 lifeCountry.csv파일을 다운받아서 위 폴더에 넣으세요.
5. 이제 분석을 위한 모든 준비가 끝났습니다.
6. Infile/Preprocessing을 해야합니다. classProject1.Rmd에서 lifeCountry.csv를 불러와서 data.frame으로 저장합니다.
7. 각 나라의 GDP와 기대수명에 대해서 산점도를 그리고 대륙에 따라서 점의 색깔이 달라지게 해보세요.
8. 자유롭게 분석을 시작해보세요.