

M21 - Flow

LearningSpoonsR

2019-01-20

The Flow

- 해당업체는 미국과 미국 외의 오프라인 지점에서 유아용 카시트를 판매하는 업체입니다.
- 도시별 판매 집계 데이터를 이용해 insight를 얻어봅니다.
- Part 1. 데이터 수집
 - ISLR package에서 Carseat 데이터 셋을 구합니다.
 - ISLR은 Introduction to Statistical Learning in R이라는 머신러닝 학술서적에 포함된 함수와 데이터를 포함한 패키지입니다.
 - 기초 통계에 대한 지식이 있으면 읽을 수 있는 좋은 서적입니다.
 - 영문판은 무료로 책이 배포되어 있어서 github에도 공유하였고, 번역서도 출판되어 있습니다.
- Part 2. 전처리
 - dplyr은 R에서 데이터 사이언스의 근간이 되는 data.frame을 효율적으로 다루는 패키지입니다.
 - Rstudio의 Scientist가 만들었기에 체계적이고 간결합니다.
- Part 3. 분석 & 시각화
 - ggplot2은 R을 다른 언어에 비해서 가장 돋보이게 하는 체계적이고 효과적인 시각화 도구입니다.
- Part 4. 문서화
 - rmarkdown은 분석 결과를 다양한 포맷의 문서로 만들어주는 도구입니다.
 - 데이터 분석 환경을 제공합니다.

Part 1. 수집 - from "ISLR" package

패키지 (package, 응용프로그램, library)

- 각각의 패키지는 특정 데이터나 함수를 제공함.
- 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.
- 저희 수업에서는 R에서 데이터 사이언스의 과정을 support하는 강력한 패키지 위주로 진행합니다.

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

```
install.packages("packageName") # playstore, Appstore
```

- 설치이후에는 패키지 사용 전에 아래와 같이 패키지 사용을 선언해주어야 함.
- 따옴표 없이 입력
- #declare, #load, #import

```
library(packageName) # declaring "I will use this package!"
```

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, linebreak) 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 패키지(`read_excel`, `readxl`)를 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 행이 제목인 경우 이를 `column name`으로 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) # load
```

- `data.frame`을 파일로 저장할 때에는 아래의 명령으로…

```
write.csv(dataset, "fileName.csv") # save
```

2. txt파일

- csv파일과 매우 유사함.
- csv파일은 comma로 나누어져 있기에 별도의 입력이 필요하지 않지만, txt파일에서는 무엇으로 나누어져있는가를 입력해주어야 함. 이를 separator(sep)라고 함.

```
dataset <- read.table("fileName.txt", header = FALSE, sep = "\t", stringsAsFactors = FALSE)
write.table(dataset, "fileName.txt") # save
```

3. xls, xlsx파일

- 불러오기는 가능, 저장은 불가능

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

- 현재 메모리의 모든 객체 (변수와 함수)

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

경로 (디렉토리, directory)

- Rstudio를 어떻게 실행시켰냐에 따라서 현재 작업 디렉토리가 설정됨.
 - Rstudio 아이콘을 더블클릭하여 실행하였을 때에는
 ⇒ 시스템의 기본 설정대로 현재 작업 디렉토리가 정해짐.
 - 이미 존재하는 .R이나 .Rmd 파일을 더블클릭하여 Rstudio를 실행했을 때에는
 ⇒ 해당 파일의 경로가 현재 작업 디렉토리가 됨.
- 확인 및 변경
 - 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") # installation  
library(ISLR) # load package  
class(Carseats) # name of data structure  
  
## [1] "data.frame"  
head(Carseats) # the first 6 observations  
  
##   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) # the last 5 observations

##   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) # pop-up windows
dim(Carseats) # dimensions

## [1] 400 11

str(Carseats) # structural view

## '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) # summary statistics
```

```
##      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 직관적으로 `data.frame`을 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, Ph.D.

- 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 `tidyverse`라는 패키지를 제작

Basic Manipulation

- tidy data.frame!

1. 개체 타입은 `data.frame`
2. 각각의 row는 관찰값을 의미
3. 각각의 column은 변수를 의미

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



Figure 1: from `dplyr` Cheatsheet

- Basic Manipulations

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

6. group_by + summarise

- Categorical 변수로 집계

1. rename (이름 바꾸기)

```
# rename Sales to Revenue
Carseats <- rename(Carseats, Revenue = Sales) # dplyr
names(Carseats)[names(Carseats)=="Sales"] <- "Revenue" # base
```

2. filter (관찰값 추출, Row 추출)

```
# takes only if Income > 100
temp <- filter(Carseats, Income > 100) # dplyr
temp <- Carseats %>% filter(Income > 100) # dplyr, pipe
temp <- Carseats[Carseats$Income > 100,] # base
# takes only if Age between 30 and 40
temp <- filter(Carseats, Age >= 30 & Age < 40) # dplyr
temp <- Carseats %>% filter(Age >= 30 & Age < 40) # dplyr, pipe
temp <- Carseats[((Carseats$Age >= 30) & (Carseats$Age < 40)),] # base
```

- &는 “and”
- |는 “or”
- pipe는 가독성을 높여줍니다.

- $y = \tanh(\sqrt{(x_1 + x_2)^2 + x_3}) - x_4$
- $y <- x_1 \%>% \text{add}(x_2) \%>% \text{square}() \%>% \text{add}(x_3) \%>% \text{sqrt}() \%>% \tanh() \%>% \text{subtract}(x_4)$

3. select (변수 추출, Column 선택)

```
# Choose the variablse Income and Population
temp <- select(Carseats, Income, Population) # dplyr
temp <- Carseats %>% select(Income, Population) # dplyr
temp <- Carseats[,c("Income", "Population")] # base
```

4. arrange (정렬)

```
# Ascending (1-2-3)
Carseats <- arrange(Carseats, Price) # dplyr
Carseats <- Carseats %>% arrange(Price) # dplyr
Carseats <- Carseats[order(Carseats$Price),] # base
# Descending (3-2-1)
Carseats <- arrange(Carseats, desc(Price)) # dplyr
Carseats <- Carseats %>% arrange(desc(Price)) # dplyr
Carseats <- Carseats[order(Carseats$Price, decreasing = TRUE),] # base
```

5. mutate (새로운 변수)

```
# base
Carseats$Revenue <- Carseats$Sales

# dplyr
Carseats <- mutate(Carseats,
                    AdvPerCapita = Advertising/Population,
                    RevPerCapita = Revenue/Population)

# dplyr (with pipe)
Carseats <- Carseats %>%
  mutate(AdvPerCapita = Advertising/Population,
        RevPerCapita = Revenue/Population)

# base
Carseats$AdvPerCapita <- Carseats$Advertising/Carseats$Population
Carseats$RevPerCapita <- Carseats$Revenue/Carseats$Population

# mutate with ifelse
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 있나요?

```

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

- Process
 - AgeClass라는 factor 변수 생성 (mutate)
 - AgeClass로 묶어서(group_by)
 - mean(Sales)를 집계(summarise)
 - 새로운 data.frame이 만들어짐

Discussion

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 데이터 베이스와 통신하는 언어
 2. R에서도 `sqldf`라는 패키지를 사용해서 SQL명령어로 작업할 수 있음
 3. R을 할 줄 아는 사람이 SQL을 배우는데 걸리는 시간 < 1 day
 4. 데이터를 보는 눈을 키워 줍니다.
- 데이터를 전처리하는 작업의 소요시간은 전체 프로젝트에서 80% 이상입니다.

cheatsheet for dplyr (1)

Data Transformation with dplyr :: CHEAT SHEET

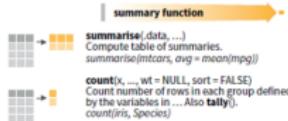


dplyr functions work with pipes and expect tidy data. In tidy data:



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).

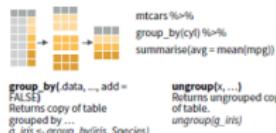


VARIATIONS

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

Group Cases

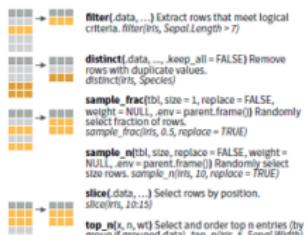
Use `group_by()` to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.



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()
 > >= is.na() ! &

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(iris, desc(Sepal.Length))

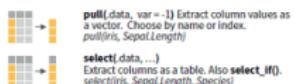
ADD CASES

add_row(data, ..., before = NULL, after = NULL)
Add one or more rows to a table.
add_row(firethorn, eruptions = 1, waiting = 1)

Manipulate Variables

EXTRACT VARIABLES

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

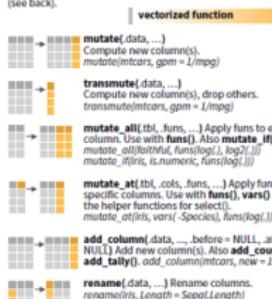


Use these helpers with `select()`,
e.g. `select_if(iris, starts_with("Sepal"))`

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 funs 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

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 diamond, iris와 함께 R에서 가장 많이 예제로 쓰임

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

```

library(ggplot2)
? mpg # getting help

## starting httpd help server ... done
mpg

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

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

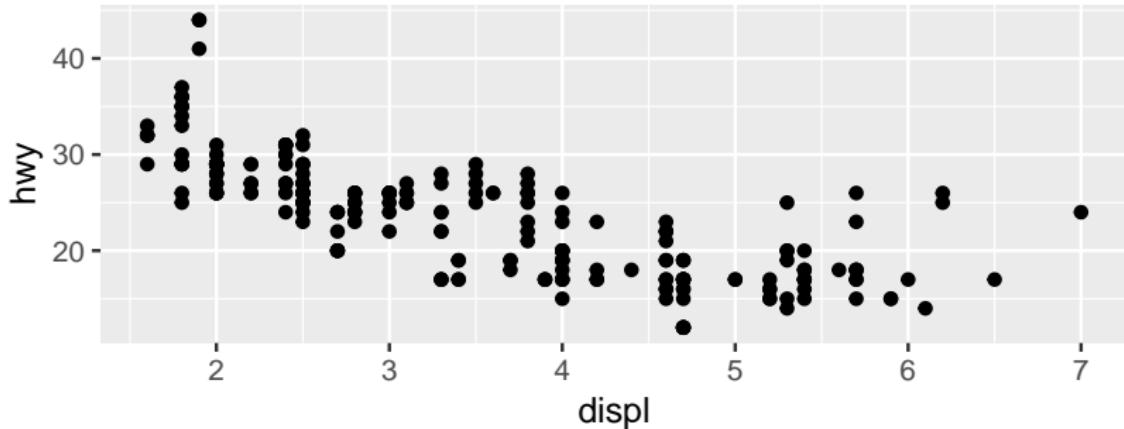
```

- mpg는 여러개의 class의 특징을 가지고 있습니다.
- tbl_df는 data.frame의 업그레이드 버전입니다.
 - data.frame의 모든 기능 사용
 - 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))
```



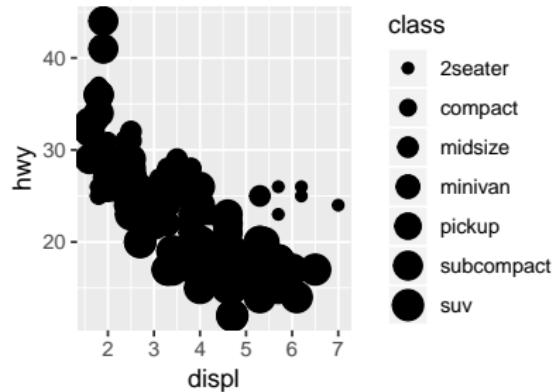
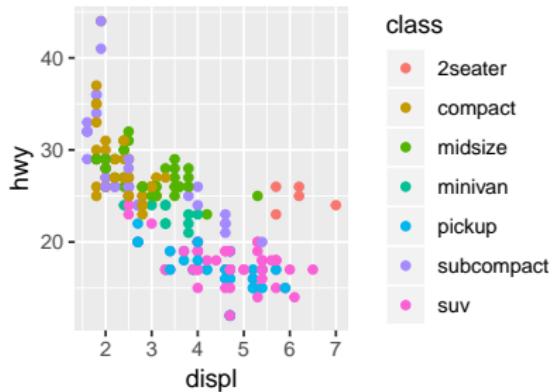
```
# basic ggplot command
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPING>))
```

- 엔진이 크면 연비가 안 좋나요?
- 추가적으로 궁금한게 있나요?

```

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)

```



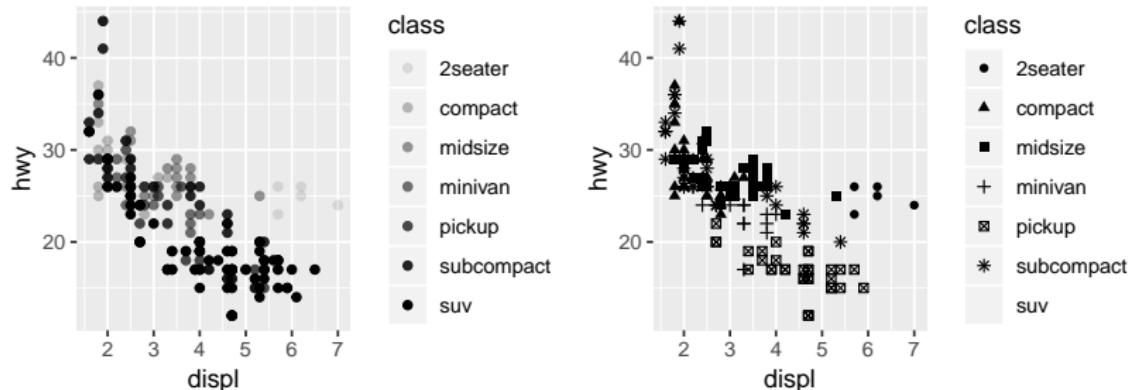
- `color=class`: class 변수의 값에 따라 color가 다르게
- `size=class`: class 변수의 값에 따라 size가 다르게
- a와 b는 각각 graph 객체
- `gridExtra`는 `grid.arrange` 함수를 사용하게 해 줌
- `suppressMessages()`는 “로딩되었습니다.”와 같은 메시지를 보기 싫을 때 선택적으로 사용
- `grid.arrange(a, b, nrow=1, ncol=2)`: a와 b를 1개행, 2개열에 배치

```

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.

```



- `alpha=class`: class 변수의 값에 따라 alpha(진하기)가 다르게
- `shape=class`: class 변수의 값에 따라 shape이 다르게
- a, b, c, d 중에 어느 그림이 가장 효과적인가요?
- 일반화 시키실 수 있나요? 즉, size, alpha, color, shape의 사용 기준은?

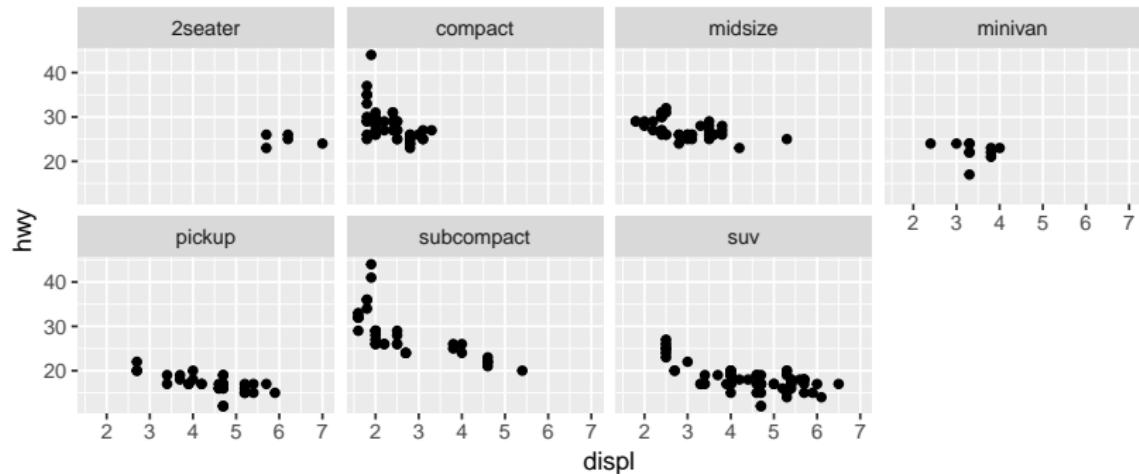
Aesthetics (aes) - size, alpha, color, shape

변수 타입	ggplot aes	특징
Numeric (quantitative, 정량적)	<code>size</code> <code>alpha</code>	수치 더할 수 있음 크고 작음
Factor (qualitative, 정량적)	<code>color (<10)</code> <code>shape (<7)</code>	분류 더할 수 없음 집합 연산

자주 쓰이는 ggplot의 추가 기능들

Facets (1 variable) - facet_wrap

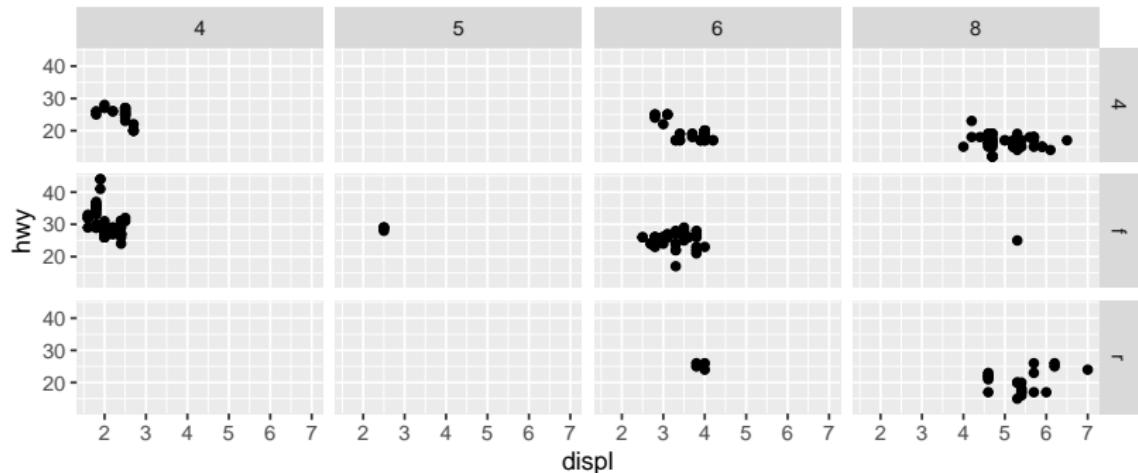
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_wrap(~ class, nrow = 2)
```



- `facet_wrap(~ class, nrow = 2)`
 - class를 x축으로 하여
 - 2행으로 배열

Facets (2 variables) - facet_grid

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



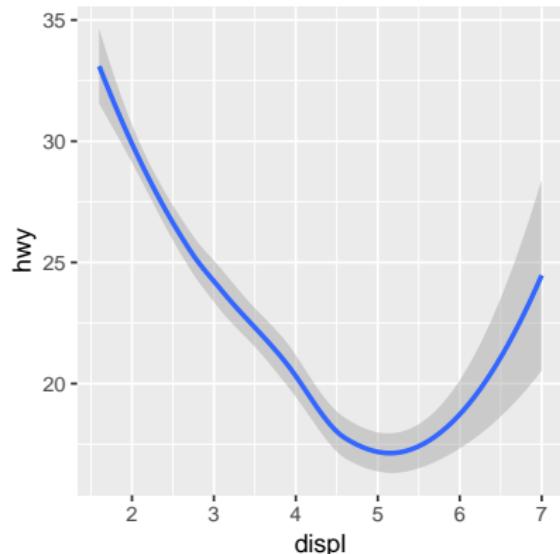
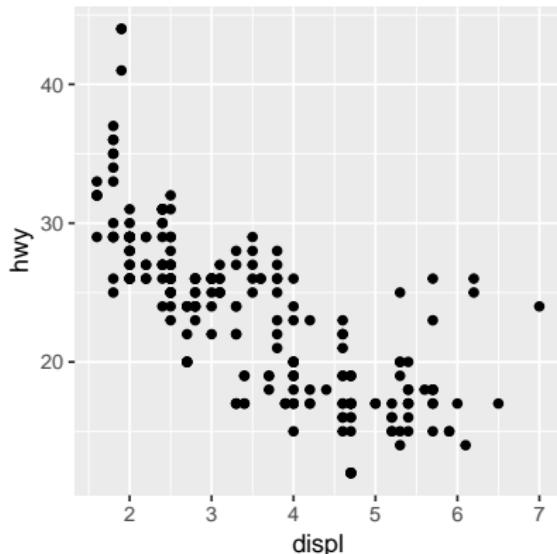
- `facet_grid(drv ~ cyl)`
 - cyl을 x축으로
 - drv를 y축으로
 - grid 모양으로 배열

Discussion

- When or not to facet?
1. Data가 충분히 많을 때 -> do facet
 2. 각각의 Distribution을 보고 싶을 때 -> do facet
 3. 근접 비교를 하고 싶을 때 -> don't facet
 4. 결과물의 reader가 content에 대한 사전 지식 수준이 높을 때 -> may facet
 5. factor 변수의 갯수를 고려해야 함 -> **magic number**를 염두

Point vs Smooth

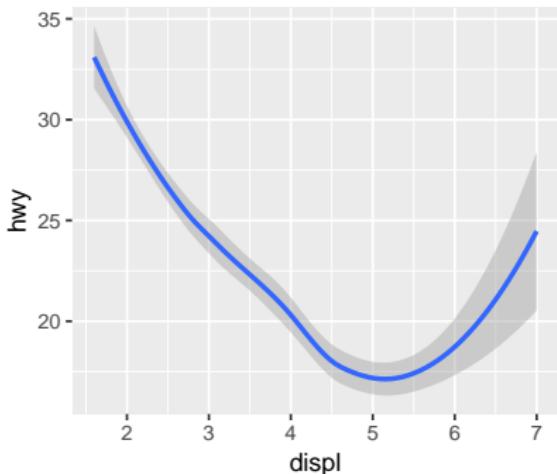
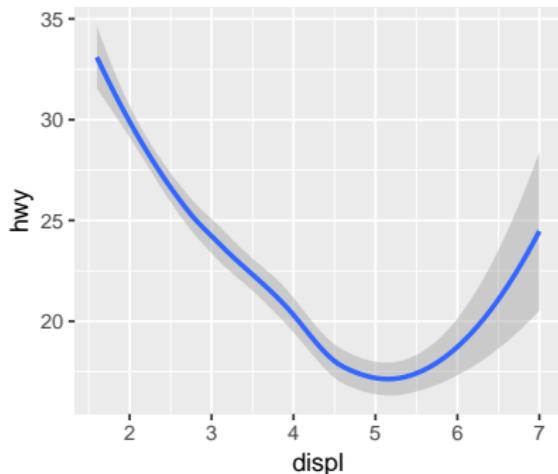
```
e <- ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))  
f <- ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))  
grid.arrange(e, f, nrow = 1, ncol = 2)
```



- `geom_point`: 점으로 표시
- `geom_smooth`: curve로 표시

AES mapping의 중첩

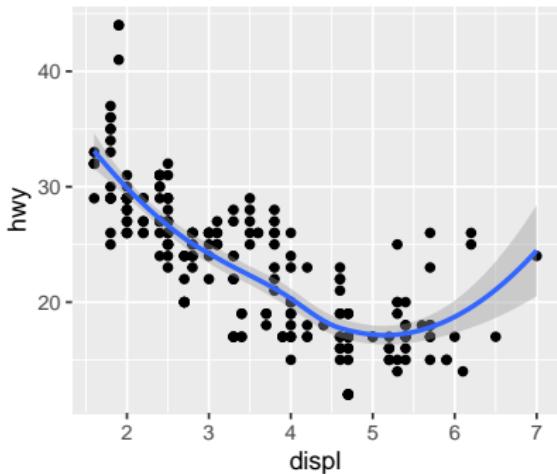
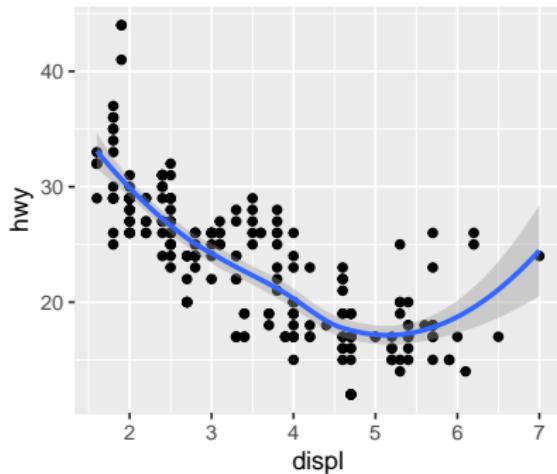
```
g <- ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
h <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth()
grid.arrange(g, h, nrow = 1, ncol = 2)
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



- g와 h는 동일하지만, 명령어의 variation을 주었음
- 즉, h의 `geom_smooth()`에서는 `ggplot`함수의 `mapping` 값을 가져옴

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'
```



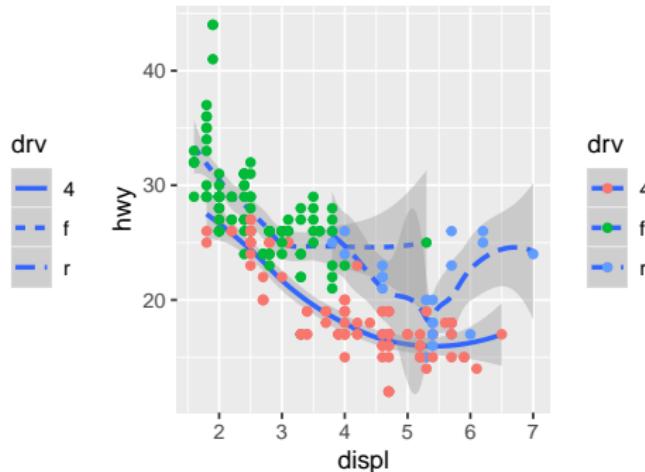
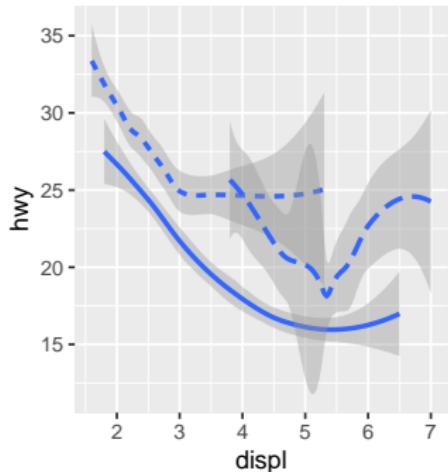
- j의 _point()와 _smooth()가 앞의 mapping을 상속받음
- 코드가 간결해짐

```

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'

```



- 1에서 `mapping`을 상속받으면서 `_smooth()`와 `_point()`가 더욱 구체화 됨

Summary - ggplot

- Data, Aesthetic, Geometric Object를 명시하여 그림을 rendering.
- gridExtra 패키지의 grid.arrange 함수를 이용하여 grid 형태로 위치시킬 수 있음
- 변수의 성격을 이해하여 size, alpha, color, shape와 facet을 적절히 사용

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

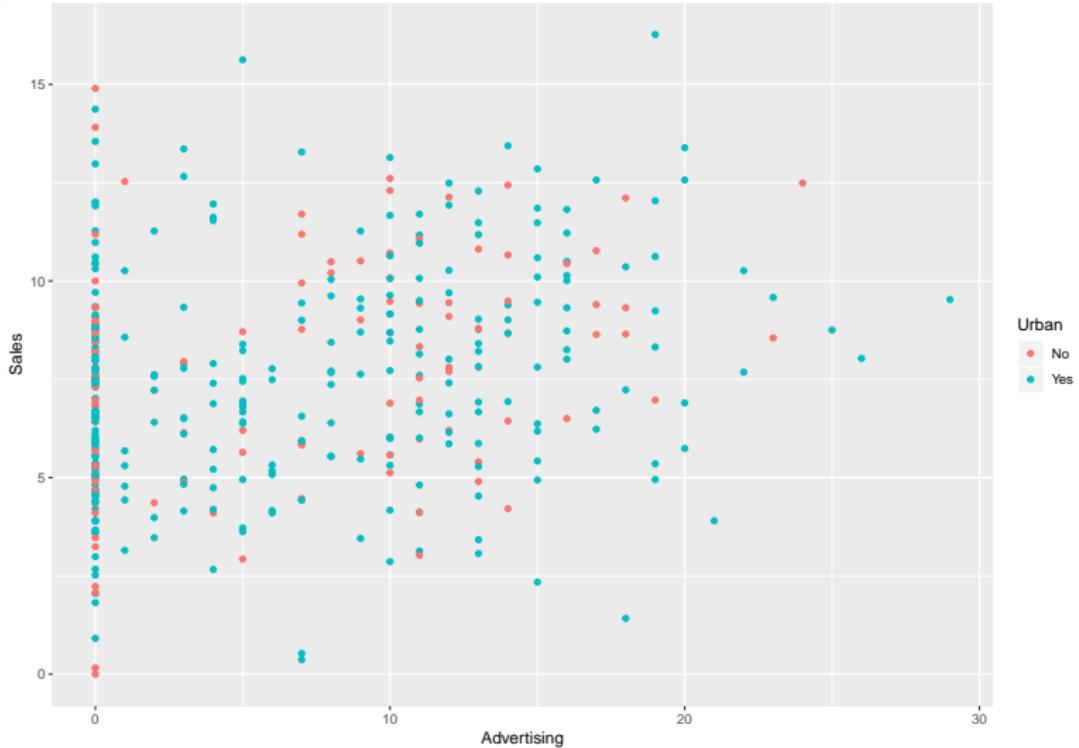
1. 그림의 목적을 정하고
2. X (explanatory variable - 설명 변수)의 개수와 각 변수의 중요도와 순서
3. Y (dependent variable)를 명시
4. ggplot 객체를 rendering

Carseat

```
str(Carseats)

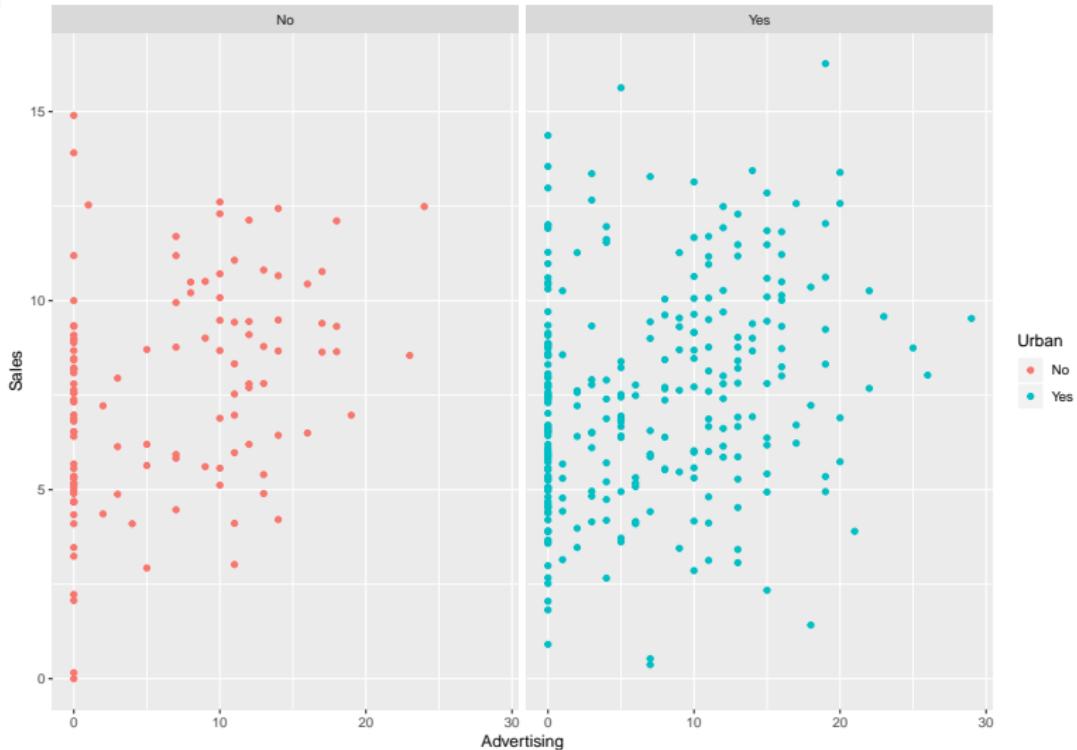
## 'data.frame': 400 obs. of 15 variables:
## $ Sales      : num  0.37 0 6.67 6.39 5.01 9.53 3.58 5.4 6.53 2.93 ...
## $ CompPrice   : num  147 139 156 131 159 175 142 149 154 143 ...
## $ Income      : num  58 24 42 21 69 65 109 73 30 21 ...
## $ Advertising : num  7 0 13 8 0 29 0 13 0 5 ...
## $ Population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

```
# Add features to existing ggplot object!
a <- a + facet_wrap(~ Urban)
print(a)
```



- 2 slides above…

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

- 1 slide above…

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

- 두 가지 기능을 합하고 사용자가 선택할 수 있게 한다.
- May allow user to choose as well!
- Why markdown?

```
doAgeFacet <- TRUE
a <- ggplot(data = Carseats, aes(x = Advertising, y = Sales)) +
  geom_point(aes(color = Urban))
if (doAgeFacet) {
  a <- a + facet_wrap(~ Urban)
}
print(a)

doAgeFacet <- FALSE
a <- ggplot(data = Carseats, aes(x = Advertising, y = Sales)) +
  geom_point(aes(color = Urban))
if (doAgeFacet) {
  a <- a + facet_wrap(~ Urban)
}
print(a)
```

Example: M41-longevity

Rich Country = Live Longer?? Final Project: [ggplot](#) + [shiny](#) + [flexdashboard](#) Source Code

OECD
 OECD
 non-OECD

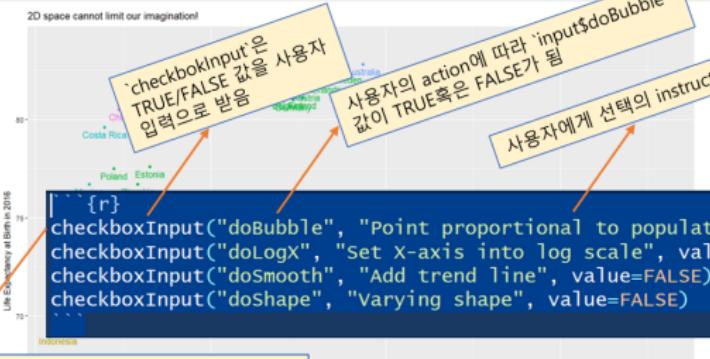
Continents
 Asia
 Europe
 Africa
 North America
 South America
 Oceania

Range of Country GDP (USD)
\$10,000 to \$100,000

Point proportional to population
Set X-axis into log scale
Add trend line
Varying shape

Life Expectancy at Birth in 2016

2D space cannot limit our imagination!



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)

Input\$doBubble 값이 TRUE인 경우에 기존 객체 myGgplot에 size=Population이라는 feature 추가

```
| 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
```

checkboxInput은 TRUE/FALSE 값을 사용자 입력으로 받음
사용자의 action에 따라 `input\$doBubble` 값이 TRUE 혹은 FALSE가 됨
사용자에게 선택의 instruction 제공
기본값은 FALSE로 설정했음

Stats

An alternative way to build a layer

A stat builds new variables to plot (e.g., count, prop).



Visualize a stat by changing the default stat of a geom function, `geom_bar(stat="count")` or by using a stat function, `stat_count(geom="bar")`, which calls a default function, `stat_summary(fun="count")`. You can also use `..name..` syntax to map stat variables to aesthetics.



```
c + stat_bin(binwidth=1, origin=10)
x + y | .count ..density.. ..density..
c + stat_count(count=1, n=100)
x + y | ..count.. ..density.. ..scaled..
```

```
x + binwidth(2) | x, y, z | ..level..
c + stat_boxplot(0.05, segments=5, type="T")
l + stat_contour(level=0.95, segments=51, type="T")
l + stat_summary(bins=2, bins=30, fun=max)
x, y, z, Rf | ..value..
c + stat_ellipse(level=0.95, segments=51, type="T")
l + stat_hexbin(30, x, y, z, order=..level..)
```

```
l + stat_hexagon(2, x, y, z, order=..level..)
```

```
l + stat_hexagon_hexbin(30, x, y, z, order=..level..)
```

```
x + stat_hexagon_hexbin_hexbin(30, x, y, z, Rf, order=..level..)
```

```
c + stat_hexagon_hexbin_hexbin_hexbin_hexbin(30, x, y, z, Rf, order=..level..)
```

```
c + stat_ecdf(n=40) | x...y...
c + stat_quantile(quantiles=c(0.1, 0.9), formula=y~x, alpha=0.95, color="red", size=1.5)
c + stat_smooth(method="lm", formula=y~x, se=T, level=0.95, x.y, ...=x...y..., ymin=yhat, ymax=yhat)
```

```
ggplot() + stat_function(fun=cos, n=99, fun=norm, args=list(d=0.5) | x...y...
c + stat_identities(name=TRUE)
```

```
ggplot() + stat_qq(q=qnorm, n=100, dist="t", sample=x, ...=y, ...=sample, theoretical=x, ...=summaryfun, data="mean_ci_boot")
```

```
b + stat_summary(fun.y="mean", geom="bar")
```

```
e + stat_unique()
```



Scales

Scales map data values to the visual values of an aesthetic. To change a mapping, add a new scale.



GENERAL PURPOSE SCALES

Use x/y with aesthetics

- `scale_x_continuous` - map cont_val values to visual ones
- `scale_y_continuous` - map discrete values to visual ones
- `scale_x_discrete` - map discrete values to visual ones
- `scale_y_discrete` - map discrete values to visual ones
- `scale_x_manual` - map discrete values to visual ones
- `scale_y_manual` - map discrete values to visual ones
- `scale_x_date` - labels in "mm/dm/yy", date_breaks = "2 weeks", date_labels = "date"
- `scale_y_date` - treat data values as date times.
- `scale_x_datetime` - treat data values as date times. (the same arguments as scale_x_date). See `?strptime` for label format.

X & Y LOCATION SCALES

Use x/y with aesthetics (x shown here)

- `scale_x_log10` - Plot x on log10 scale
- `scale_x_reverse` - reverse direction of x axis
- `scale_x_sqrt` - Put x on square root scale

COLOR AND FILL SCALES (DISCRETE)

- `n + d + geom_bar(aes(fill=..fill..))`
- `n + scale_fill_brewer(palette="Blues")`
- `n + scale_fill_brewer(display=brewer.all)`
- `n + scale_fill_grey(start=0.2, end=0.8, na.value="red")`

COLOR AND FILL SCALES (CONTINUOUS)

- `o + c + geom_dotplot(aes(fill=..color..))`
- `o + scale_fill_distiller(palette="Blues")`
- `o + scale_fill_gradient("red", "high"="yellow")`
- `o + scale_fill_gradient2("low"="red", "high"="blue", mid="white", midpoint = 25)`

- `o + scale_fill_gradients(colors=topo.colors[6])`
- Also: `rainbow()`, `heat.colors()`, `terrain.colors()`, `cm.colors()`, `ColorBrewer::brewer.pal`

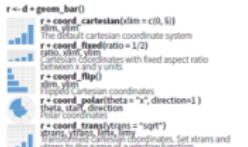
SHAPE AND SIZE SCALES

- `p + e + geom_point(aes(shape=0, size=cy0))`
- `p + scale_shape() + scale_size()`
- `p + scale_size_continuous(limits=c(10, 7))`
- `p + scale_size_area(max_size=8)`

THEMES

RStudio is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-440-1212 | rstudio.com • Learn more at <http://ggplot2.tidyverse.org> • ggplot2 2.1.0 • Updated: 2016-11

Coordinate Systems



Use x/y with aesthetics

- `coord_polar(theta="y", direction=1)`
- `coord_flip(theta="x", direction=1)`
- `coord_rect(xmin=-1, xmax=1, ymin=-1, ymax=1)`
- `coord_trans(x="sqrt", y="sqrt")`

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

- `s + ggplot(mpg, aes(flf, flf, flf))`
- `flf + position_dodge("group")`
- Average elements side by side
- `s + geom_bar(position="dodge")`
- `s + geom_bar(position="stack")`
- Normalizes height of top of one another, avoiding overlap
- `s + geom_bar(position="stack")`
- `s + geom_bar(position="stack")`
- Stack elements on top of one another

Each position adjustment can be recast as a function with manual width and height arguments

`s + geom_bar(position = position_dodge(width = 1))`

`s + geom_bar(position = position_stack(height = 1))`

`s + geom_bar(position = position_stack())`

More References

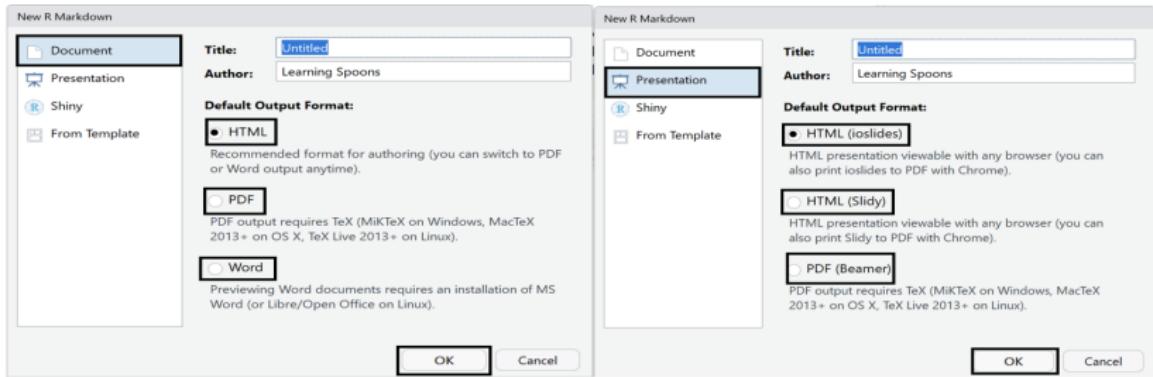
1. M23-More on ggplot2 - 다음주!
2. M24-ggplot2-50examples - ggplot의 갤러리 (50개 예제)
3. M34-Happy B-day, ggplot
 - ggplot의 10주년을 기념하는 article
 - 크롤링과 구글 번역 API를 이용하여 자동 번역 및 문서화
 - <https://qz.com/1007328/all-hail-ggplot2-the-code-powering-all-those-excellent-charts-is-10-years-old/>
4. M6X References/books/ggplot2.pdf
 - 무료 배포된 ggplot관련 영문 서적
 - 번역본도 출간되어 있음

Part 4. 문서화 - "rmarkdown"

New Generation of Computer Programming
Literate Programming

- Motivation
 - 1. Workflow 관점
 - old: 계산 -> 표, 그림 정리 -> 워드프로세서에 정리 -> 다시 계산 -> 표, 그림 정리 -> 워드프로세서 -> 반복 반복
 - new: **rmarkdown**
 - 2. Display 관점
 - old: 에디터, 콘솔, 아웃풋 윈도우, 엑셀, 워드프로세서…
 - new: **rmarkdown** under Rstudio

파일 -> 새파일 -> R Markdown



- **.html**
 - interactive feature 가능
- **.pdf**
 - professional 문서
 - 조판을 위해 texlive가 설치되어 있어야 함 (slide 76)
 - 한글 사용은 별도의 템플릿 사용 (M25-.pdf)
- **.docx**
 - MS office 프로그래 언어도 제작 가능
- **ioslides, slidy**
 - interactive feature 가능
 - **.html** 형식의 슬라이드
 - slidy 템플릿 (M27-slidy)
- **beamer (.pdf)**
 - Scientific presentation
 - 조판을 위해 texlive가 설치되어 있어야 함 (slide 76)
 - 한글 사용은 별도의 템플릿 사용 (M26-beamer)
 - 성실하 선생님들의 강의노트

- 시작: File -> New File -> R Markdown -> Document/Presentation + 포맷 선택
- 렌더링: Knit -> Knit to XXX

The screenshot illustrates the RStudio interface for generating an HTML document from R Markdown code.

Code Editor: The left pane shows the R Markdown code:

```
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
17 <http://rmarkdown.rstudio.com>. When you click the **Knit** button a document will be generated.
18
19 ``{r cars}
20 summary(cars)
21
22 #> R 코드는 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
```

Annotations highlight specific parts of the code:

- A red box surrounds the first few lines of the code, with a callout pointing to the rendered header in the browser.
- A red box surrounds the summary(cars) code, with a callout pointing to the rendered table in the browser.
- A red box surrounds the note about using R code, with a callout pointing to the rendered note in the browser.
- A red box surrounds the plot(pressure) code, with a callout pointing to the rendered plot in the browser.

Browser Preview: The right pane shows the generated HTML document:

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.

	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. :128.00

#> R 코드는 r ncol(cars) 와 같이 사용.

Including Plots

You can also embed plots, for example:

.pdf (템플릿: M25-.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 <- 1-10
42 rmd pdf template #
```

rmd pdf template
Learning Spoons
2018-06-30

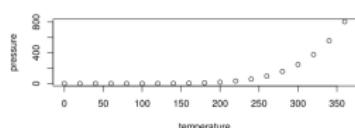
R Markdown 한글

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 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  
15 for authoring HTML, PDF, and MS Word documents. For more details on  
16 using R Markdown see <http://rmarkdown.rstudio.com>.  
17  
18 when you click the Knit button a document will be generated that  
19 includes both content as well as the output of any embedded R code  
20 chunks within the document. You can embed an R code chunk like this:  
21  
22 ## [r cars]  
23 summary(cars)  
24  
25  
26 ## Including Plots  
27  
28 You can also embed plots, for example:  
29  
30 ## [r pressure, echo=FALSE]  
31 plot(pressure)  
32  
33  
34
```

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

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 :35.4 Mean : 42.98  
## #> #> 3rd Qu.:39.0 3rd Qu.: 56.00  
## #> #> Max. :25.0 Max. :120.00
```

At the bottom of the Word document, there is a status bar with the text "1/2 페이지 151개 단어 영어(미국)" and a page number "58/77".

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://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

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

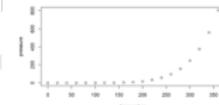
Slide with R Output

```

    speed      dist
    n.s. <= 4.0  n.s. <= 2.1
    1st Qu. 12.0  1st Qu. 26.8
    Median 15.0  Median 38.0
    Mean 15.6  Mean 42.1
    3rd Qu. 19.0  3rd Qu. 56.0
    n.s. <= 26.0  n.s. <= 128.0

```

Slide with Plot



slidy (템플릿: M27-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)
```

```



Untitled

Learning Spoons

2018-04-17

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

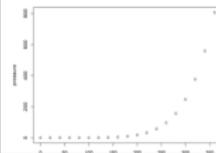
- Bullet 1
- Bullet 2
- Bullet 3

Slide with R Output

summary(cars)

| | speed | dist |
|---------|-------|-------|
| Min. | 5.0 | 1.0 |
| 1st Qu. | 12.0 | 9.0 |
| Median | 19.0 | 43.0 |
| 3rd Qu. | 24.0 | 62.0 |
| Max. | 29.0 | 219.0 |

Slide with Plot



beamer (템플릿: M26-beamer)

The screenshot shows the RStudio interface with the 'rmd-beamer-template.Rmd' file open in the left pane. The code is a template for a beamer presentation, including setup, document class, and various chunk options for different output formats like LaTeX, PDF, and Beamer. The right pane displays the Beamer presentation preview, which includes a title slide and a slide with a plot. The Beamer navigation bar at the top indicates the current slide number.

```
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(rmarkdowm)  
24  knitr::opts_chunk$set(echo = TRUE)  
25  knitr::opts_chunk$set(background = '71BABA')  
26  ***  
27  
28  ## beamer@Rmarkdown  
29  
30  - 이 템플릿은 한글 및 twocolumn layout으로 beamer 문서를 제작할 수 있는 템플릿입니다. \br  
31  - RMarkdown ('RMX-rnd' 를 참조) \br  
32  - pdf 조판을 위한 texlive 엔진 ('RMX-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_custo  
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 = '71BABA')'
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

.Rmd 파일의 구성

1. Header

- yaml, pandoc
- title, author, date 입력
- 파일 포맷 컨트롤

2. R chunks

- R Language 입력
- 오른쪽에 3개의 단추 (setting, run all chunks above, run this chunk)
- `knitr::opts_chunks$set()`으로 global setting 가능

3. MD chunks

- Human Language 입력
- Inline R 사용 가능

Example - 1주차 숙제

The image shows two side-by-side windows. On the left is RStudio with an R Markdown file named 'Review-Week1.Rmd'. The code includes global settings for R chunks and inline R code examples. On the right is a Korean web browser displaying the rendered output of the R code, which consists of several 'Problem' sections with their corresponding R code and results.

RStudio (Left):

```
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`.  
18+ ``{r}  
19+ a <- "Hello"  
20+ a  
21+  
22+ ans:  
23+  
24+  
25+ **Problem** `r problem`.  
26+ ``{r}  
27+ a <- "Hello"  
28+ b <- "world"  
29+ paste(a,b)  
30+  
31+ ans:  
32+
```

Global setting: R chunks - Global setting

Inline R code: `ans`

Korean Web Browser (Right):

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
ans

Problem 3
paste(a,b)
ans
ans

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

Problem에 1을 더해 자동으로 문제번호 컨트롤

```
**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  
`r`
```

```
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:
```

참고: 그림크기 조정

1. R chunk에서 rendering 되는 그림의 크기 조정
 - Chunk control 부분에서 크기 지정 (인치 단위로)
 - `{r, fig.width = 7, fig.height = 2}`
2. Markdown chunk에서 그림을 첨부하면서 크기 조정
 - 아래와 같이 입력 (`rmarkdown Cheatsheet` 참고하세요!)
 - `![] (fig.png){width=400px, height=300px}\``

Pandoc's Markdown

Write with syntax on the left to create effect on right (after render)

Plain text

End a line with two spaces

“Header” and “Body”

autobreak (split 2+ lines)

autolink (links)

escaped (` `)

emphasis (`` ``)

mention (`a < a >`)

equation block:

$$S(t) = m/(t^2 + S_0)$$

equation block:

$$S(t) = m/(t^2 + S_0)$$

• block quote

Header1 [Header]

Header 2 [From_H]

Header 3 [cnn_dan]

Header 4

Header 5

Header 6

– (List item)

Unordered list in HTML

Unordered list in pdf/docx

<http://www.rstudio.com/>

<http://www.rstudio.com/>

image:

(Caption) (bolded) (img)

• ordered list

◦ sub-item 1

◦ sub-item 2

◦ sub-item 3

Item 1

Continued (indented 4 spaces)

1. ordered list

2. item 1

◦ sub-item 1

◦ sub-item 2

◦ sub-item 3

Continued (indented 4 spaces)

1. ordered list

2. item 1

◦ sub-item 1

◦ sub-item 2

◦ sub-item 3

(iii) At the numbering

continues after

(iii) An interruption

Term 1

◦ Definition 1

◦ (Right | Left | Default | Center)

◦ 12 12 12 12

◦ 120 120 120 120

◦ 1 1 1 1

◦ white bullet 1

◦ white bullet 2

◦ white bullet 3

(– To have bullets appear on click)

horizontal rule/line break:

A footnote [*]

[*] Here is the footnote.

Plain text

Start a new paragraph

“Header” and “Body”

autobreak (split 2+ lines)

autolink (links)

escaped (` `)

emphasis (`` ``)

mention (`a < a >`)

equation block:

$$S(t) = m/(t^2 + S_0)$$

equation block:

$$S(t) = m/(t^2 + S_0)$$

• block quote

Header1 [Header]

Header 2 [From_H]

Header 3 [cnn_dan]

Header 4

Header 5

Header 6

– (List item)

Unordered list in HTML

Unordered list in pdf/docx

<http://www.rstudio.com/>

<http://www.rstudio.com/>

image:

(Caption) (bolded) (img)

• ordered list

◦ sub-item 1

◦ sub-item 2

◦ sub-item 3

Item 1

Continued (indented 4 spaces)

1. ordered list

2. item 1

◦ sub-item 1

◦ sub-item 2

◦ sub-item 3

Continued (indented 4 spaces)

1. ordered list

2. item 1

◦ sub-item 1

◦ sub-item 2

◦ sub-item 3

(iii) At the numbering

continues after

(iii) An interruption

Term 1

◦ Definition 1

◦ (Right | Left | Default | Center)

◦ 12 12 12 12

◦ 120 120 120 120

◦ 1 1 1 1

◦ white bullet 1

◦ white bullet 2

◦ white bullet 3

(– To have bullets appear on click)

horizontal rule/line break:

A footnote [*]

[*] Here is the footnote.

Set render options with YAML

When you render, R Markdown

- runs the R code, embeds results and text into .md file with knitr
- then converts the .md file into the finished format with pandoc



Set a document's default output in the YAML header:



Customize output with sub-options (listed to the right):



Use table class to place sub-headers into tabs



Reusable Template

Create a reusable template

1. Create a new package with a `inst/markdown/templates` directory

2. In the directory, Place a folder that contains `template.yaml` (see below) and all other files that are part of the template (any supporting files, etc.)

3. Install the package

4. Access template in wizard at File > New File > R Markdown

template.yaml



name: My Template

... Here is the template

Table Suggestions

Several functions format R data into tables

1. `table`: Create a new package with a `inst/markdown/templates` directory

2. In the directory, Place a folder that contains

`template.yaml` (see below)

and all other files that are part of the template (any supporting files, etc.)

3. Install the package

4. Access template in wizard at File > New File > R Markdown

template.yaml

... Here is the template

name: My Template

... Here is the template



rmardown

| | html | pdf | word | rtf | tex | beamer | slidy | beamer_presentation | beamer_beamer | beamer_minimal | beamer_minimal_presentation | beamer_minimal_beamer | beamer_minimal_minimal | beamer_minimal_minimal_presentation | beamer_minimal_minimal_beamer |
|--|------|-----|------|-----|-----|--------|-------|---------------------|---------------|----------------|-----------------------------|-----------------------|------------------------|-------------------------------------|-------------------------------|
| | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

bibliography.nib

style.css

smith.cit

smith.pdf

smith.doc

smith.ppt

smith.html

smith.rtf

smith.tex

smith.beamer

smith_minimal.cit

smith_minimal.pdf

smith_minimal.doc

smith_minimal.ppt

smith_minimal.html

smith_minimal.rtf

smith_minimal.tex

smith_minimal.beamer

smith_minimal_minimal.cit

smith_minimal_minimal.pdf

smith_minimal_minimal.doc

smith_minimal_minimal.ppt

smith_minimal_minimal.html

smith_minimal_minimal.rtf

smith_minimal_minimal.tex

smith_minimal_minimal.beamer

smith_minimal_minimal_minimal.cit

smith_minimal_minimal_minimal.pdf

smith_minimal_minimal_minimal.doc

smith_minimal_minimal_minimal.ppt

smith_minimal_minimal_minimal.html

smith_minimal_minimal_minimal.rtf

smith_minimal_minimal_minimal.tex

smith_minimal_minimal_minimal.beamer

smith_minimal_minimal_minimal_minimal.cit

smith_minimal_minimal_minimal_minimal.pdf

smith_minimal_minimal_minimal_minimal.doc

smith_minimal_minimal_minimal_minimal.ppt

smith_minimal_minimal_minimal_minimal.html

smith_minimal_minimal_minimal_minimal.rtf

smith_minimal_minimal_minimal_minimal.tex

smith_minimal_minimal_minimal_minimal.beamer

smith_minimal_minimal_minimal_minimal_minimal.cit

smith_minimal_minimal_minimal_minimal_minimal.pdf

smith_minimal_minimal_minimal_minimal_minimal.doc

smith_minimal_minimal_minimal_minimal_minimal.ppt

smith_minimal_minimal_minimal_minimal_minimal.html

smith_minimal_minimal_minimal_minimal_minimal.rtf

smith_minimal_minimal_minimal_minimal_minimal.tex

smith_minimal_minimal_minimal_minimal_minimal.beamer

smith_minimal_minimal_minimal_minimal_minimal_minimal.cit

smith_minimal_minimal_minimal_minimal_minimal_minimal.pdf

smith_minimal_minimal_minimal_minimal_minimal_minimal.doc

smith_minimal_minimal_minimal_minimal_minimal_minimal.ppt

smith_minimal_minimal_minimal_minimal_minimal_minimal.html

smith_minimal_minimal_minimal_minimal_minimal_minimal.rtf

smith_minimal_minimal_minimal_minimal_minimal_minimal.tex

smith_minimal_minimal_minimal_minimal_minimal_minimal.beamer

smith_minimal_minimal_minimal_minimal_minimal_minimal_minimal.cit

smith_minimal_minimal_minimal_minimal_minimal_minimal_minimal.pdf

smith_minimal_minimal_minimal_minimal_minimal_minimal_minimal.doc

smith_minimal_minimal_minimal_minimal_minimal_minimal_minimal.ppt

smith_min

R 마크다운

初心者

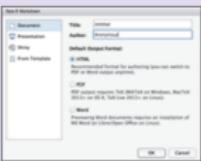
추가 학습 정보 rmarkdown.rstudio.com

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.



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.
글씨는 굵기 and *기울인 글씨*
글씨 굵기 and *굵은 글씨*
着重선
着重선+
(문서: www.rstudio.com)

제목 1

제목 2

제목 3

제목 4

제목 5

제목 6

N자 크기 대시 부호: --

mN자 크기 대시 부호: ---

줄바꿈

혹시 사용하는 수식: \$A = \sqrt{r^2 + r^2}

이미지:

수평선 (혹은 슬라이드 영역):

--

> 인용 틀

*는 사용하는 목록

* 항목 2

* 하위 항목 1

* 하위 항목 2

1. 순서 있는 목록

2. 항목 2

* 하위 항목 1

* 하위 항목 2

표 제작 | 무단계 제작

표 1 | 표 2

간 1 | 간 2

간 3 | 간 4

출력 결과

일반 텍스트
새로운 단락을 시작하면 줄 미지막을 뒤에 2개로 끝낸다.
글씨는 굵기 and *기울인 글씨*
글씨 굵기 and *굵은 글씨*
着重선
着重선+
(문서: www.rstudio.com)

제목 1

제목 2

제목 3

제목 4

제목 5

제목 6

N자 크기 대시 부호: --

mN자 크기 대시 부호: ---

줄바꿈

혹시 사용하는 수식: \$A = \sqrt{r^2 + r^2}

이미지:

수평선 (혹은 슬라이드 영역):

--

> 인용 틀

*는 사용하는 목록

* 항목 2

* 하위 항목 1

* 하위 항목 2

1. 순서 있는 목록

2. 항목 2

* 하위 항목 1

* 하위 항목 2

표 제작 | 무단계 제작

표 1 | 표 2

간 1 | 간 2

간 3 | 간 4

4. 출력 형식 설정

R 마크다운 파일에서 생성할 문서 유형을 기술하는 YAML 헤더정보를 작성한다.

YAML

YAML 헤더는 키(key) 집합:
파일 시작지점에 나오는 키값
쌍, 헤더와 같은 3개 대쉬
를 갖는 라인 (- - -)

R Markdown 편집기
마크다운 편집기
마크다운 편집기

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

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



RStudio is a trademark of RStudio, Inc. • [CC-BY](http://cc-by.rstudio.com) RStudio • <http://rstudio.com> • 844-448-1212 • rstudio.com

한국 번역 : 이광수 www.ksooc.net 웹페이지: <http://www.ksooc.net> CC-BY-SA 웹MOOC

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 더 같다.

Two plus two equals 4.

코드 렁여리

R코드 양여리를 ```(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 상단에抉择을 선택하는 옵션을 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

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

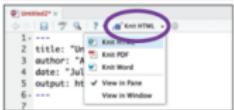
두가지 방식으로 보고서를 렌더링한다.

- 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.

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 건너뛰게 설정한다.
comment	"##"	주석문자로 출력결과에 서두를 붙인다.
fig.width	7	덩어리로 생성되는 그림에 대한 폭을 인치로 지정한다.
fig.height	7	덩어리로 생성되는 그림에 대한 높이를 인치로 지정한다.

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

- 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  
  
choose a time series:  
  (r echo = FALSE)  
  (r eval = FALSE)  
  (r hide = TRUE)  
  c1(cod1, "l1a1")  
  
See a plot:  
  (r echo = FALSE)  
  (r eval = FALSE)  
  (r hide = TRUE)  
  d <- getinputdata()  
  plot(d)
```

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

Line graph
Choose a time series:
See a plot:

* 주제: 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 - [blog.rstudio.com](#)

@rstudio



RStudio® and Shiny™ are trademarks of RStudio, Inc.
© 2012 RStudio, Inc. info@rstudio.com
844-448-1212 rstudio.com

References

1. Cheatsheets

- 영문 (2페이지)
 - M6X References/cheatsheets/08. rmarkdown-2.0.pdf
- 한글 (2페이지)
 - M6X References/cheatsheets/09. rmarkdown-cheatsheet-kr
- 영문 (5페이지)
 - <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>

2. rmarkdown 공식 페이지

- <http://rmarkdown.rstudio.com>

3. LaTex

- pdf 조판 언어
- 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 (M25, M26, M27)

- M25-.pdf: 2 columns, line numbering, page numbering
- M27-beamer: 2 columns, Singapore theme, line numbering, page numbering
- M26-slidy: 각종 컬러 삽입
- 하위폴더(rmd-pdf-template_files, myRmdBeamerStyle)등을 데리고 다녀야 됨

Discussion - rmarkdown

- Markdown이라는 Markup Language로 변환하여 사용성 높음
- 변환 과정
 1. Rmd -> md -> docx, html
 2. Rmd -> md -> tex -> pdf
- Advantages
 - 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 분석하고 insight를 공유하는 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!!!
6. ex) Python Notebook

- 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

“You don’t know what you will can get away with until you try” - Colin Powell

pdf 파일 제작에 필요한 tex 엔진 설치

1. Texlive 2017 설치 (< 1Gb)

- <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!”

실습과제 1

1. C:/LS-DS/classProject라는 폴더를 만드세요.
2. Rstudio를 열어서 rmarkdown의 html형식에 해당하는 .Rmd파일을 만드세요.
3. classProject1.Rmd로 위 폴더에 저장하세요.
4. Rstudio를 닫고 classProject1.Rmd를 더블클릭하여 실행하면 working directory가 자동으로 위의 폴더가 됩니다.
5. github M41-longevity에서 lifeCountry.csv파일을 다운받아서 위 폴더에 넣으세요.
6. 이제 분석을 위한 모든 준비가 끝났습니다.
7. Infile/Preprocessing을 해야합니다. classProject1.Rmd에서 lifeCountry.csv를 불러와서 data.frame으로 저장합니다.
8. 각 나라의 GDP와 기대수명에 대해서 산점도를 그리고 대륙에 따라서 점의 색깔이 달라지게 해보세요.
9. 자유롭게 분석을 시작해보세요.
10. html로 제작해 이메일을 보내주세요. learningSpoonsR@gmail.com