



# Lecture 3

*ggplot2 & wordcloud &  
flexdashboard  
(I have a dream)*

러닝스폰즈

2018년 4월

# 프로젝트 개요 (*"I have a dream"*)

1. 수집
2. 전처리 – *[text-mining]*
3. 분석, 시각화 – *ggplot2 + wordcloud*
4. 문서화, 공유 – *flexdashboard (advanced html for rmd)*



# *Module 1*

## *More on ggplot2*

# Grammar of Graphics

- *Motivation*

- 그래픽스에 대한 원리가 없다면, 그래픽 관련 패키지와 함수는 단지 특수 경우의 모음일 뿐
- 요리 백과사전을 다 읽는 것보다 물과 기름과 불의 작용에 대해서 익히고 백과사전을 찾아가면서 요리하는 것이 더 다양한 요리를 빠르게 할 수 있는 요리사가 되는 길

- 효과

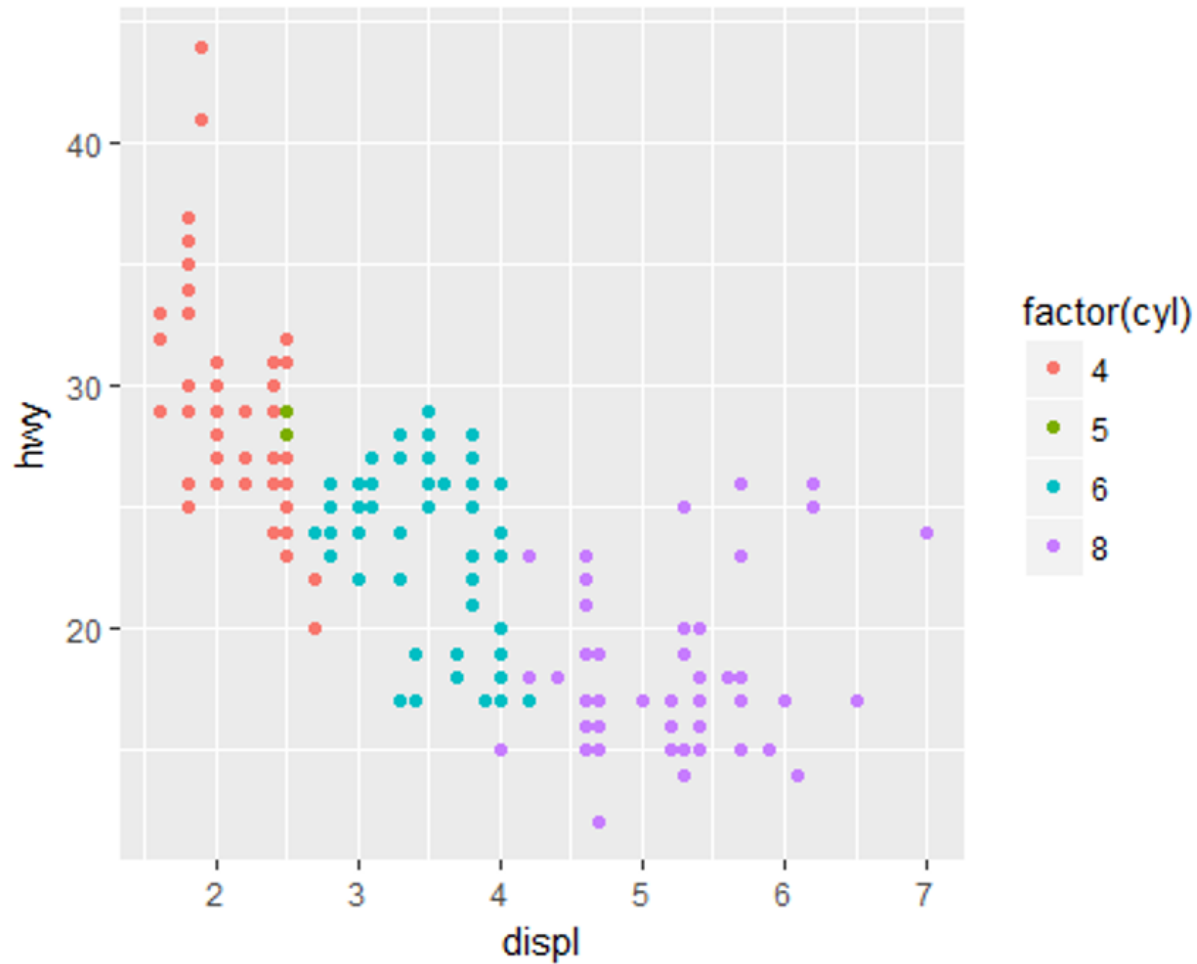
- 새로운 *package*나 함수의 등장을 빠르게 흡수
- 새로운 *graphics*를 만들어 내는 아이디어가 체계적이 됨

# Grammar of Graphics

- 특징

- 독립적이고 더할 수 있는 구성 요소들로 그래픽을 표현
- 개발과정에서 그래프의 특징을 한 가지 씩, 반복적으로 바꾸면서 그래프를 만들어 감
- 단계별로 반복적으로 바꾸어 가는 것은 생각의 흐름, 스토리텔링의 흐름과 연계시킬 수 있기에 *interactive graphics*와 잘 조화됨.

```
library(ggplot2) +  
ggplot(data = mpg, aes(x = displ, y = hwy, color = factor(cyl))) +  
  geom_point()
```



# Behind the scenes...

## data

```
head(mpg[,c("displ", "hwy", "cyl")],10)
```

```
## # A tibble: 10 x 3+  
##   displ  hwy  cyl+  
##   <dbl> <int> <int>+  
## 1  1.80    29    4+  
## 2  1.80    29    4+  
## 3  2.00    31    4+  
## 4  2.00    30    4+  
## 5  2.80    26    6+  
## 6  2.80    26    6+  
## 7  3.10    27    6+  
## 8  1.80    26    4+  
## 9  1.80    25    4+  
## 10 2.00    28    4+
```

## aes

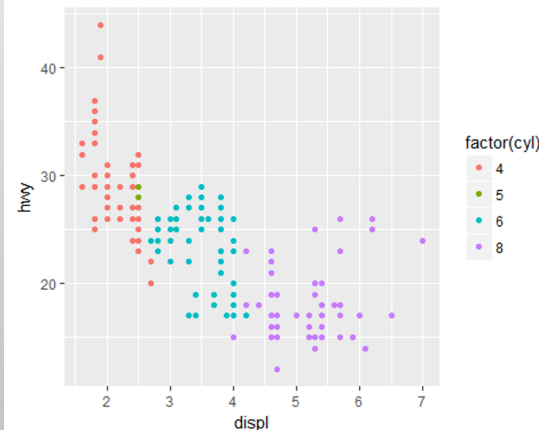
x	y	colour
1.8	29	4
1.8	29	4
2.0	31	4
2.0	30	4
2.8	26	6
2.8	26	6
3.1	27	6
1.8	26	4

## geom

x	y	colour	size	shape
0.037	0.531	#F8766D	1	19
0.037	0.531	#F8766D	1	19
0.074	0.594	#F8766D	1	19
0.074	0.562	#F8766D	1	19
0.222	0.438	#00BFC4	1	19
0.222	0.438	#00BFC4	1	19
0.278	0.469	#00BFC4	1	19
0.037	0.438	#F8766D	1	19

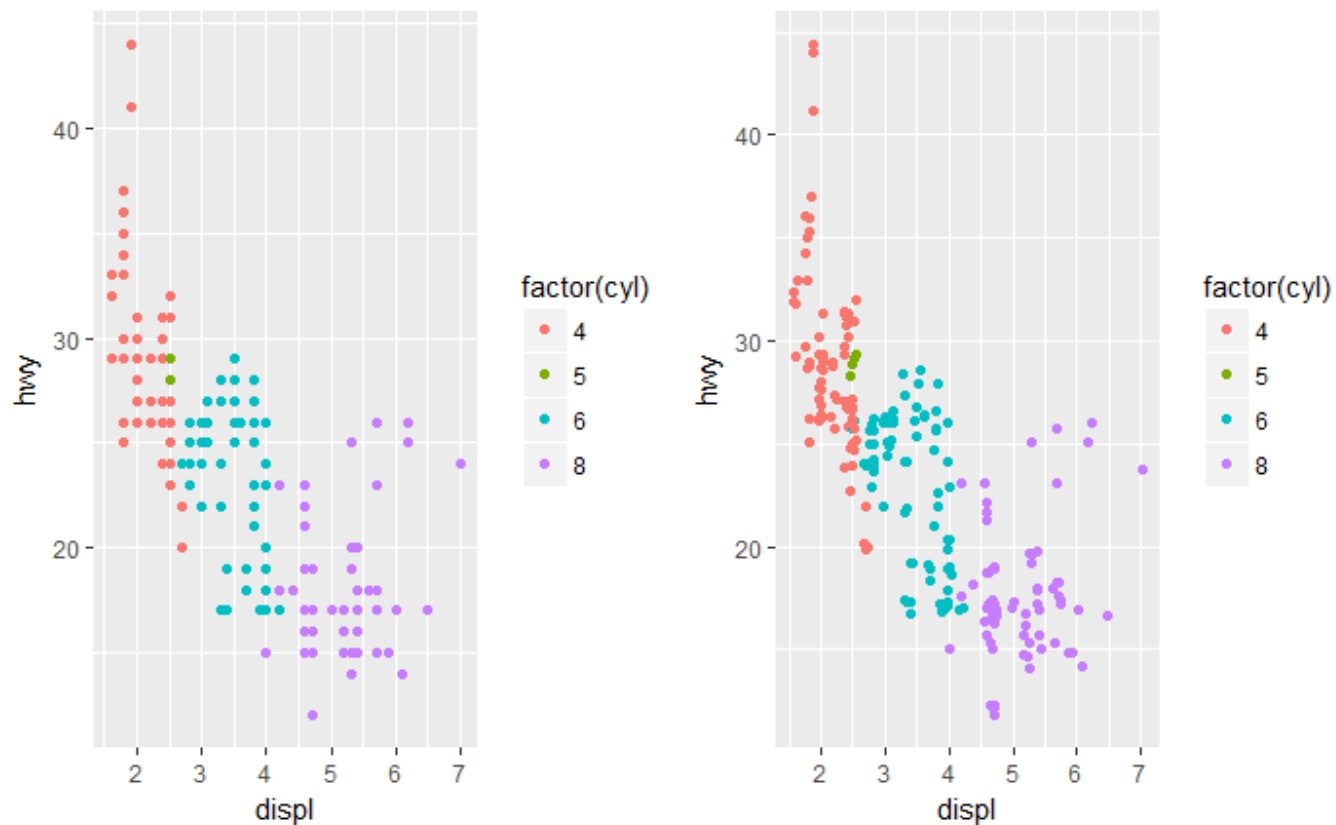
## plot

```
library(ggplot2)+  
ggplot(data = mpg, aes(x = displ, y = hwy, color = factor(cyl))) +  
geom_point()+
```



# *position = "jitter"*

```
library(ggplot2) ↓  
library(gridExtra) ↓  
a <- ggplot(data = mpg) + ↓  
  geom_point(aes(x = displ, y = hwy, color = factor(cyl)))  
b <- ggplot(data = mpg) + ↓  
  geom_point(aes(x = displ, y = hwy, color = factor(cyl)),  
            position = "jitter") ↓  
grid.arrange(a, b, nrow=1, ncol=2) ↓
```





# aesthetics and geometric\_object

- Aesthetics

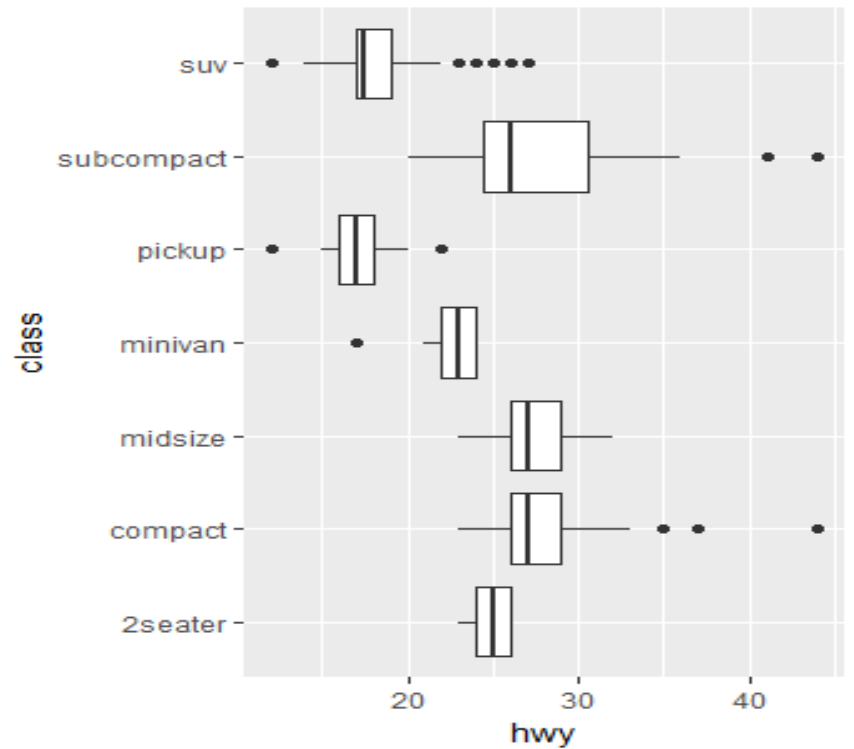
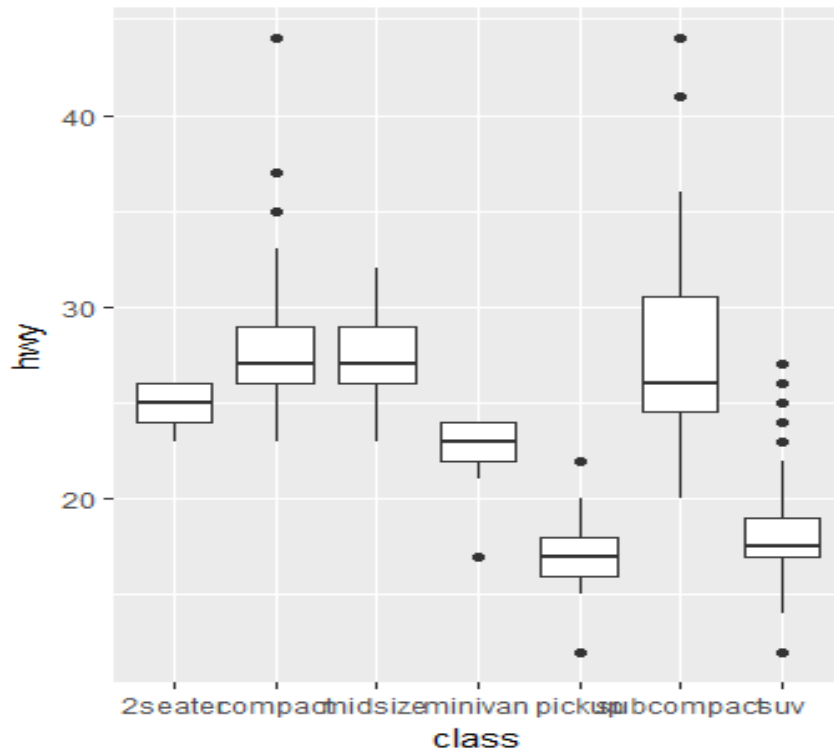
- Position  
(Horizontal & Vertical)
- Size
- Color
- Shape

- Geometric objects

이름	Geom	특징
Scatterplot (산점도)	Point	
Bubblechart	Point	변수가 Size를 결정
Bar chart	Bar	
Box-and-whisker plot	Boxplot	
Line chart	Line	

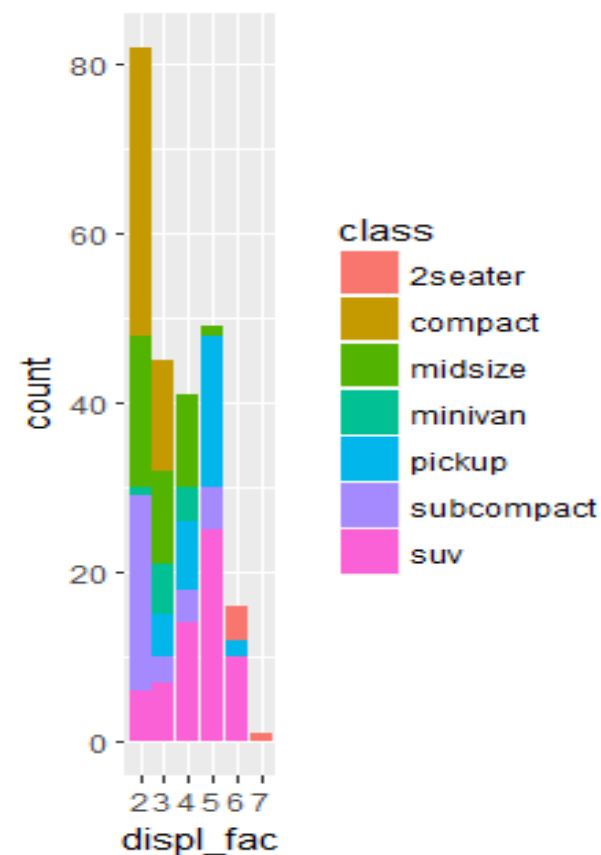
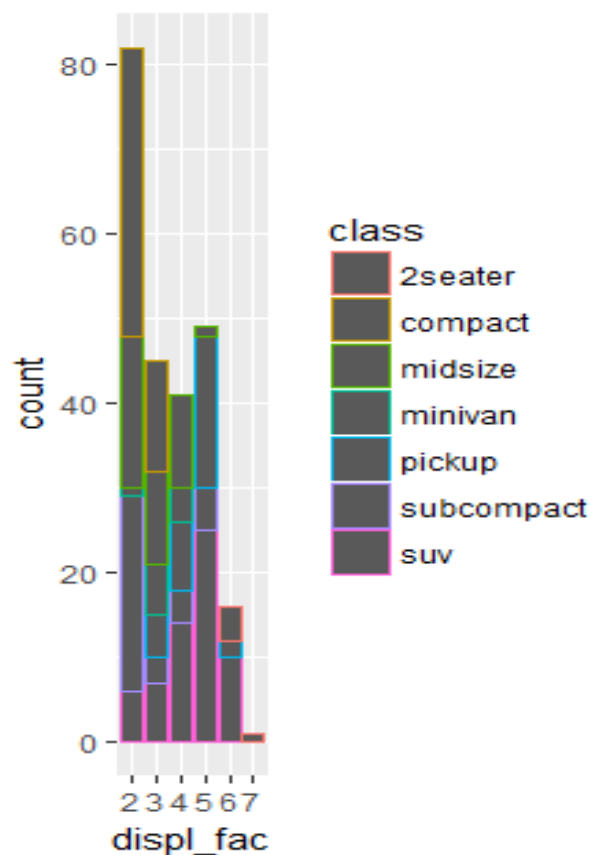
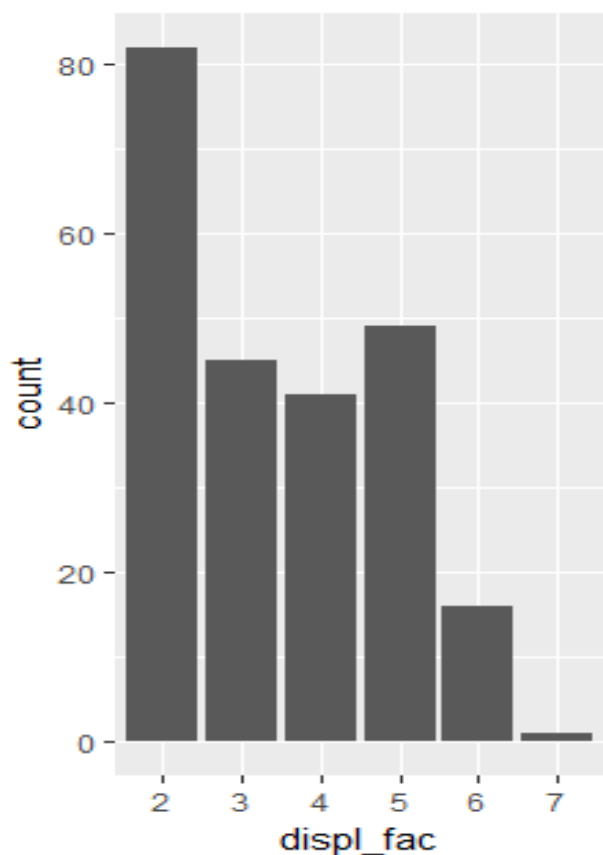
# coord\_flip()

```
a <- ggplot(data = mpg) +  
  geom_boxplot(aes(x = class, y = hwy))  
b <- ggplot(data = mpg) +  
  geom_boxplot(aes(x = class, y = hwy)) +  
  coord_flip() +  
  grid.arrange(a, b, nrow=1, ncol=2)
```



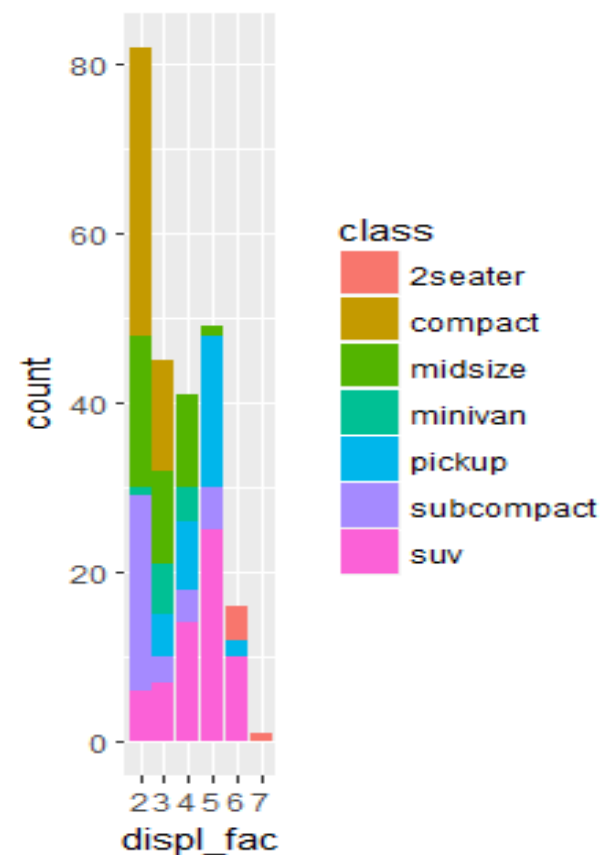
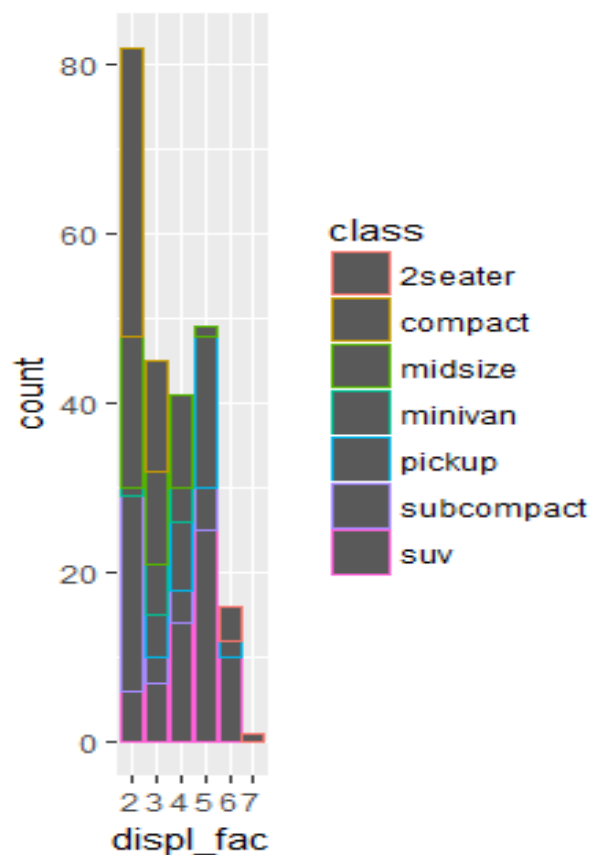
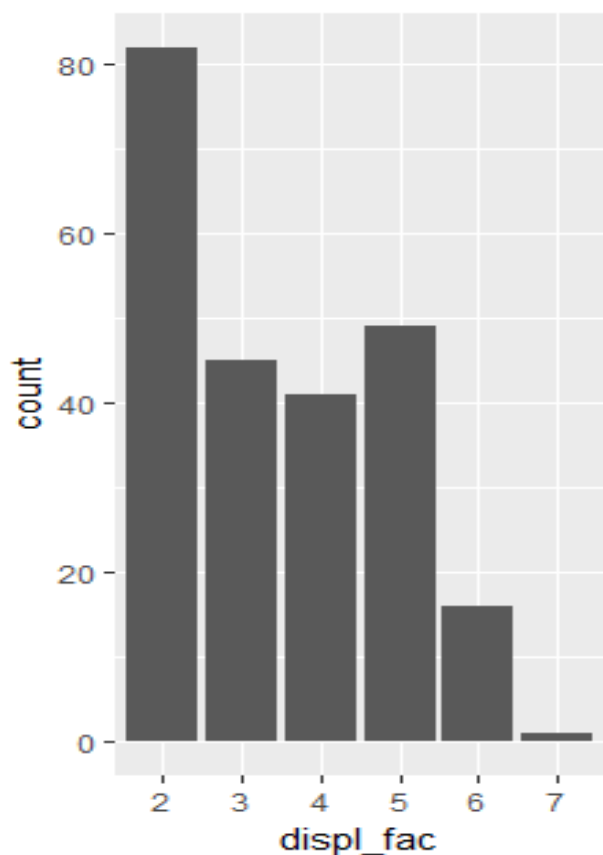
# Barchart (color and fill)

```
mpg$displ_fac <- factor(round(mpg$displ,0))  
a <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac))  
b <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac, color = class))  
c <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac, fill = class))  
grid.arrange(a, b, c, nrow=1, ncol=3)
```



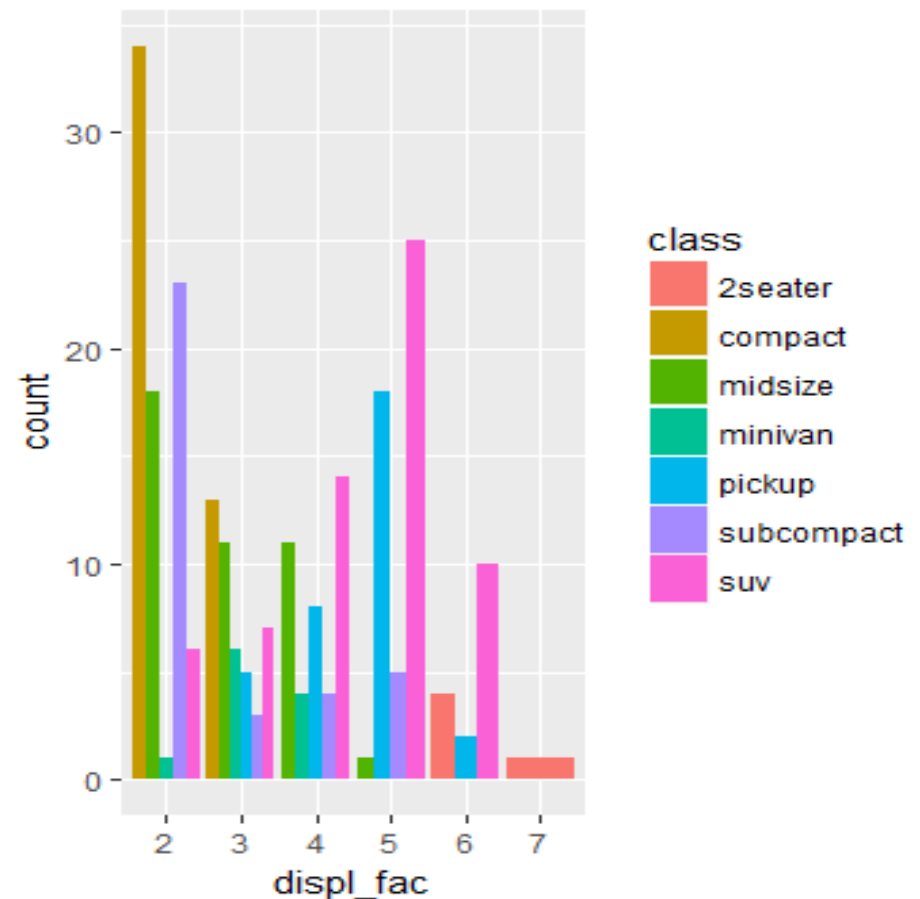
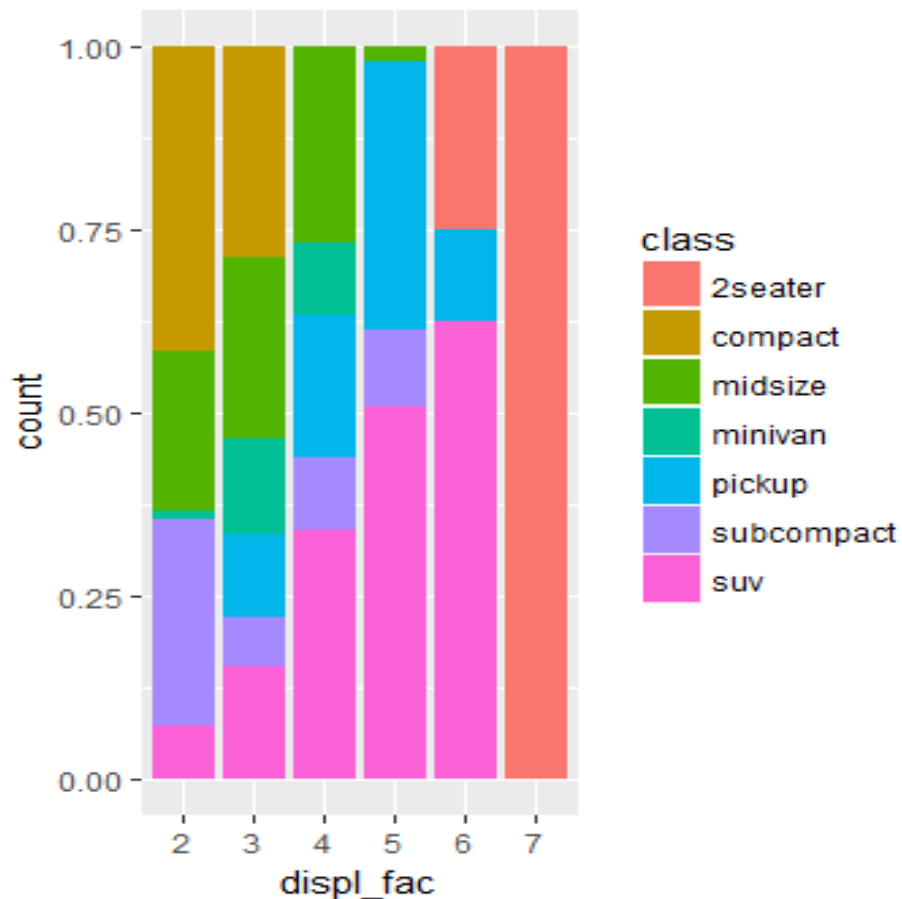
# Barchart (color and fill)

```
mpg$displ_fac <- factor(round(mpg$displ,0))  
a <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac))  
b <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac, color = class))  
c <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac, fill = class))  
grid.arrange(a, b, c, nrow=1, ncol=3)
```



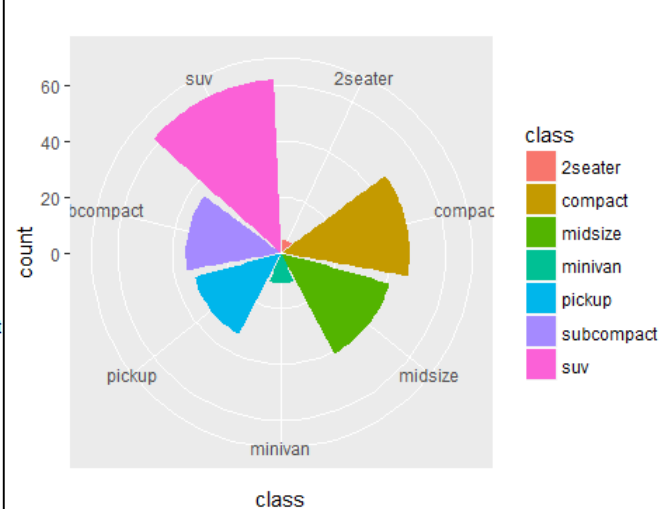
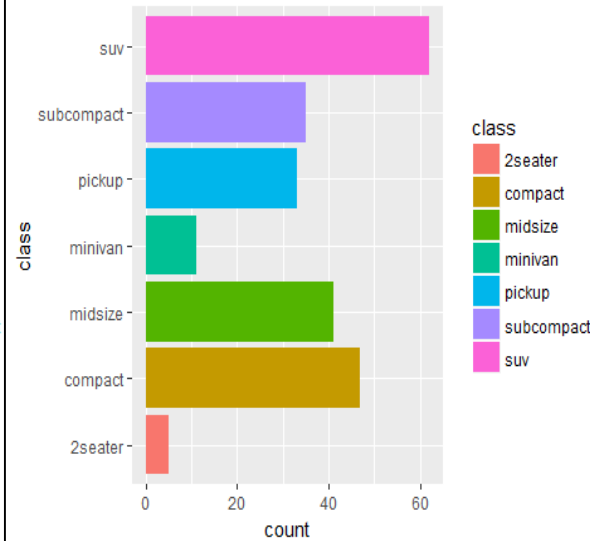
# Position in barchart

```
a <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac, fill = class), position = "fill") +  
b <- ggplot(data = mpg) +  
  geom_bar(aes(x = displ_fac, fill = class), position = "dodge")  
grid.arrange(a, b, nrow=1, ncol=2)
```



# coordinate

```
a <- ggplot(data = mpg) +  
  geom_bar(aes(x = class, fill = class)) +  
b <- a + coord_flip() +  
c <- a + coord_polar() +  
a ↵  
b ↵  
c ↵
```



## *ggplotly (interactive)*

```
library(plotly) ↓  
ggplotly(a) ↓  
ggplotly(b) ↓
```

## *save a plot*

```
# open -> save -> close ↓  
png("output_file.png") ↓  
ggplot(data = mpg) + ↓  
  geom_bar(aes(x = class, fill = class)) +  
  coord_flip() ↓  
dev.off() ↓
```







# *Module 2*

## *text-mining*



## *Module 2-1*

*“I have a dream” – Martin Luther King, Jr.*

## 0. Setup ↵

```
library(tm) # for text mining ↵  
library(SnowballC) # for text stemming ↵  
library(wordcloud) # word-cloud generator ↵  
library(RColorBrewer) # color palettes ↵  
library(ggplot2) ↵  
library(dplyr) ↵
```

## 1. Infile and prepare text (docs) ↵

```
filePath <- paste0("http://www.sthda.com/sthda/RDoc/example-files/",  
                  "martin-luther-king-i-have-a-dream-speech.txt") ↵  
text <- readLines(filePath) ↵  
docs <- Corpus(VectorSource(text)) ↵  
# inspect(docs) ↵
```

## 2. Cleanup the text (docs) ↵

# `dplyr` way ↵

```
toSpace <- content_transformer(↵  
  function (x , pattern) gsub(pattern, " ", x)) ↵  
docs <- docs %>% ↵  
  tm_map(toSpace, "/" ) %>% ↵  
  tm_map(toSpace, "@" ) %>% ↵  
  tm_map(toSpace, "\\|") ↵
```

# `base` way ↵

```
toSpace <- content_transformer(↵  
  function (x , pattern) gsub(pattern, " ", x)) ↵  
docs <- tm_map(docs, toSpace, "/") ↵  
docs <- tm_map(docs, toSpace, "@") ↵  
docs <- tm_map(docs, toSpace, "\\|") ↵
```

```
docs <- docs %>% ↵
```

```
  tm_map(content_transformer(tolower)) %>% # Convert it to Lower case ↵  
  tm_map(removeNumbers) %>% # Remove numbers ↵  
  tm_map(removeWords, stopwords("english")) %>% # Remove english common stopwords ↵  
  tm_map(removeWords, c("blabla1", "blabla2")) %>% # Remove your own stop word ↵  
  tm_map(removePunctuation) %>% # Remove punctuations ↵  
  tm_map(stripWhitespace) # Eliminate extra white spaces ↵
```

### 3. Arriving to frequency table! (docs to freqTable) ↵

```
termMat <- TermDocumentMatrix(docs) ↵
```

```
termTable <- as.matrix(termMat) ↵
```

```
head(termTable, 2) ↵
```

```
##           Docs ↵
```

```
## Terms      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 ↵
```

```
##   american 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ↵
```

```
##   deeply   0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ↵
```

```
##           Docs ↵
```

```
## Terms      25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 ↵
```

```
##   american 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ↵
```

```
##   deeply   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ↵
```

```
##           Docs ↵
```

```
## Terms      46 ↵
```

```
##   american 0 ↵
```

```
##   deeply   0 ↵
```

```
freqTable <- data.frame(word = rownames(termTable), ↵
```

```
                        freq = rowSums(termTable)) ↵
```

```
freqTable$word <- rownames(freqTable) ↵
```

```
freqTable <- freqTable %>% arrange(desc(freq)) ↵
```

```
head(freqTable, 2) ↵
```

```
##      word freq ↵
```

```
## 1    will  17 ↵
```

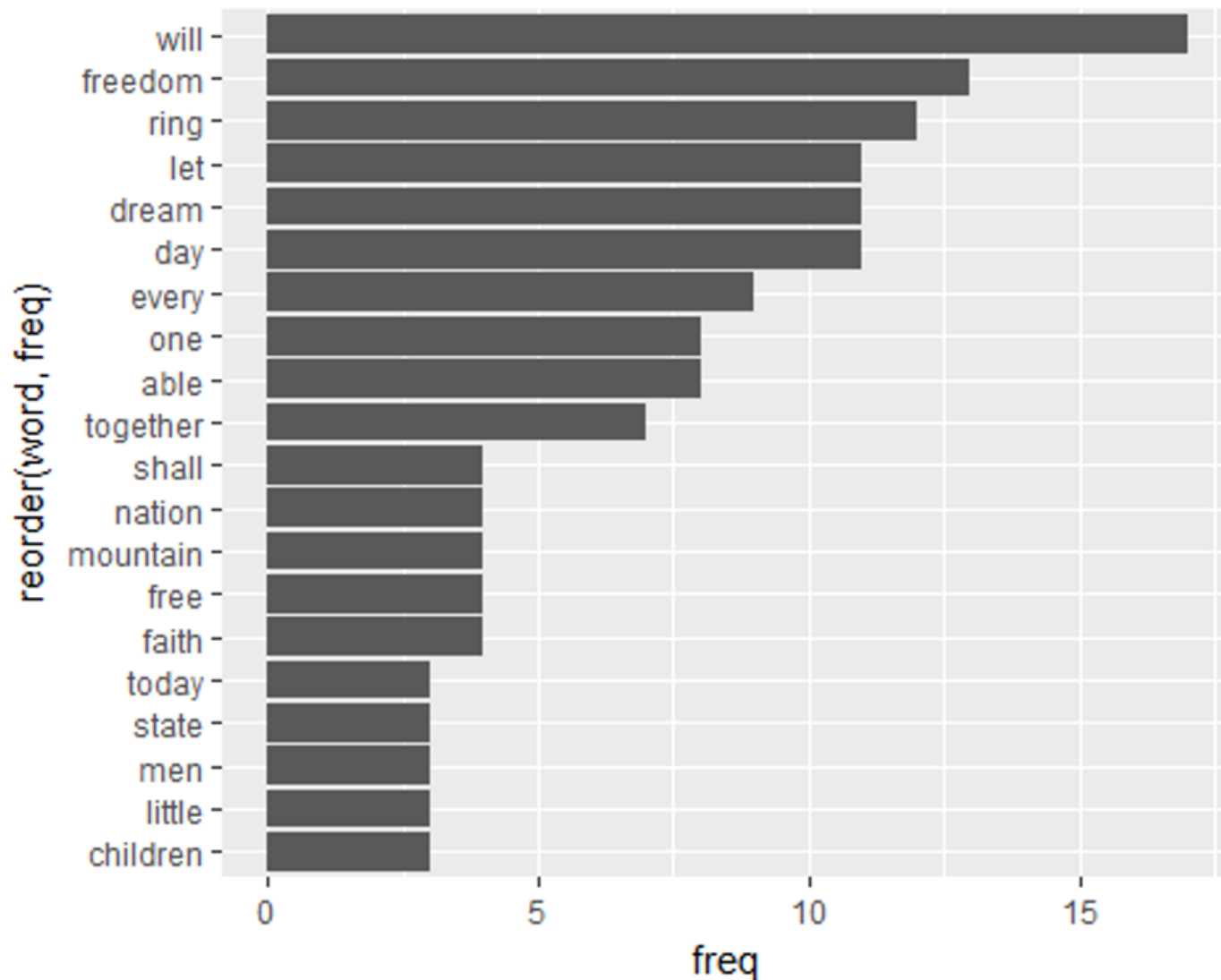
```
## 2 freedom  13 ↵
```

## 4. Render Bar Chart (play with `freqTable`) ↵

```
ggplot(head(freqTable,20)) + ↵
```

```
  geom_bar(aes(x=reorder(word, freq), y=freq), stat="identity") + ↵
```

```
  coord_flip() ↵
```

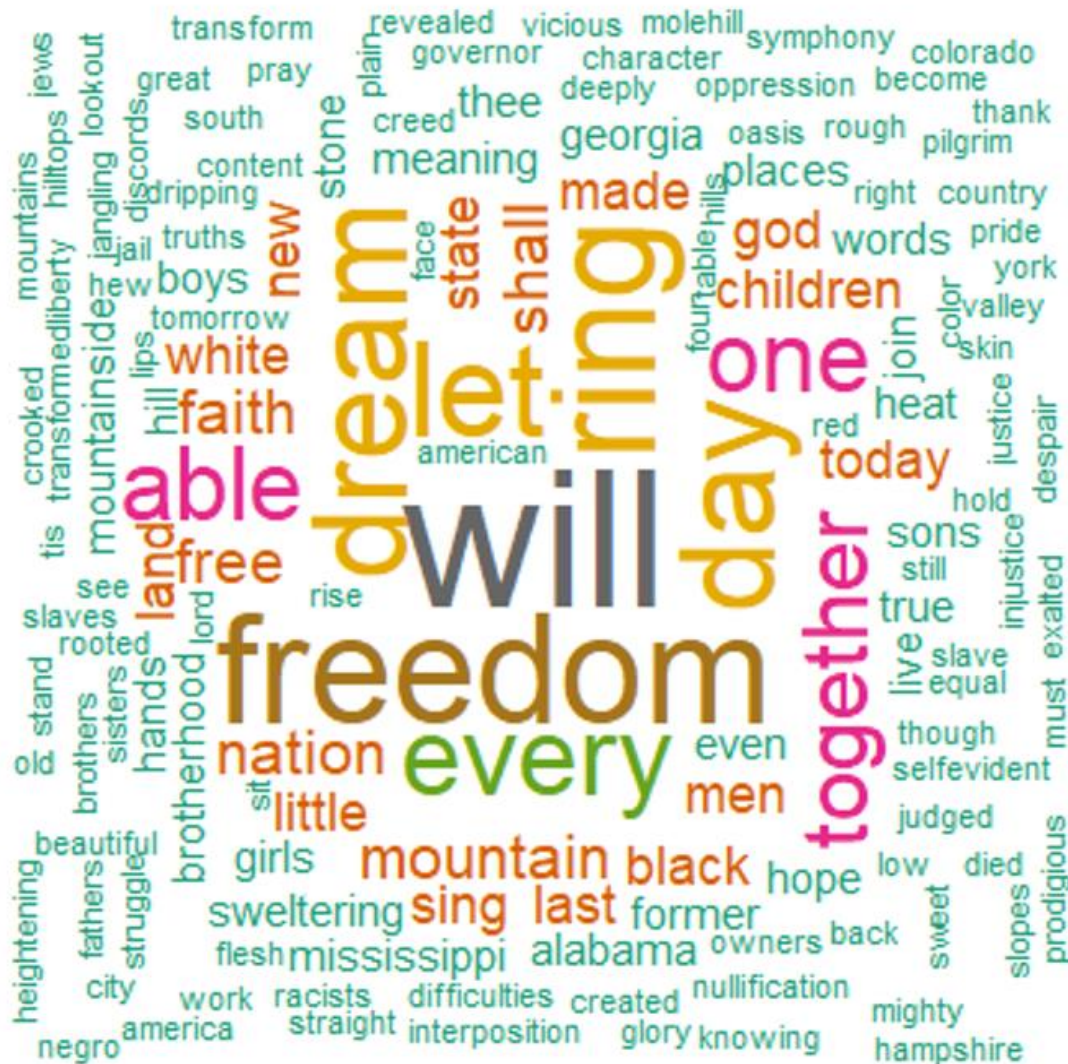




## 5. Render Word Cloud (play with freqTable)

```
wordcloud(words = freqTable$word, freq = freqTable$freq,  
min.freq = 1, max.words=200, random.order=FALSE, rot.per=0.35,  
colors=brewer.pal(8, "Dark2")))
```

↳ 1/3 0.35



# SUMMARY

## Text-mining

- *Text* 데이터에서 정보를 채굴(*mining*)하는 것으로 *Data-mining*의 한 기법
- 신문, 뉴스, *SNS*, 사용자 입력 등의 *article*이 분석 대상이 됨
- 딥러닝기술로 전처리 기술의 발전

## 전처리 기술 (영어)

- 문장을 띄어쓰기 단위로 분리
- *go vs going vs went vs be going to*를 구분
- *the, a, an* 제거



# Summary

- Text-mining Process (*plain vanilla*)

html  
ppl  
text

- 0. Setup

- 1. Infile and prepare text to "docs"

- 2. Cleanup the text ("docs")

- 3. Arriving to frequency table ("docs" to "freqTable")

- 4. Render barplot

- 5. Render wordcloud

- Question

- 한/영에는 어떤 공통점이 있나요?
  - 한/영에는 어떤 차이가 있나요?

## *Module 2-2*

# “소나기” – 황순원

## 0. Setup ↴

```
library(tm) # for text mining ↴
library(SnowballC) # for text stemming ↴
library(wordcloud) # word-cloud generator ↴
library(RColorBrewer) # color palettes ↴
library(ggplot2) ↴
library(dplyr) ↴
library(KoNLP) # Korean Natural Language Processing ↴
library(pdftools) # Extract text from pdf ↴
```

## 1. Infile and prepare text (docs) ↴

## 2. Cleanup the text (docs) ↴

```
text <- pdf_text("sonaki.pdf") ↴
docs <- sapply(text, extractNoun, USE.NAMES = F) # Apply extract Noun ↴
docs <- unlist(docs) ↴
docs <- Filter(function(x) {nchar(x) >= 2}, docs) # Character Length >= 2 ↴
head(docs, 10) ↴
```

##	[1]	"소나기"	"황순원"	↓
##	[3]	"소년"	"개울가"	↓
##	[5]	"소녀"	"증손녀(曾孫女)딸이라는"	↓
##	[7]	"소녀"	"개울"	↓
##	[9]	"물장난"	"서울서는"	↴

freq table  
my

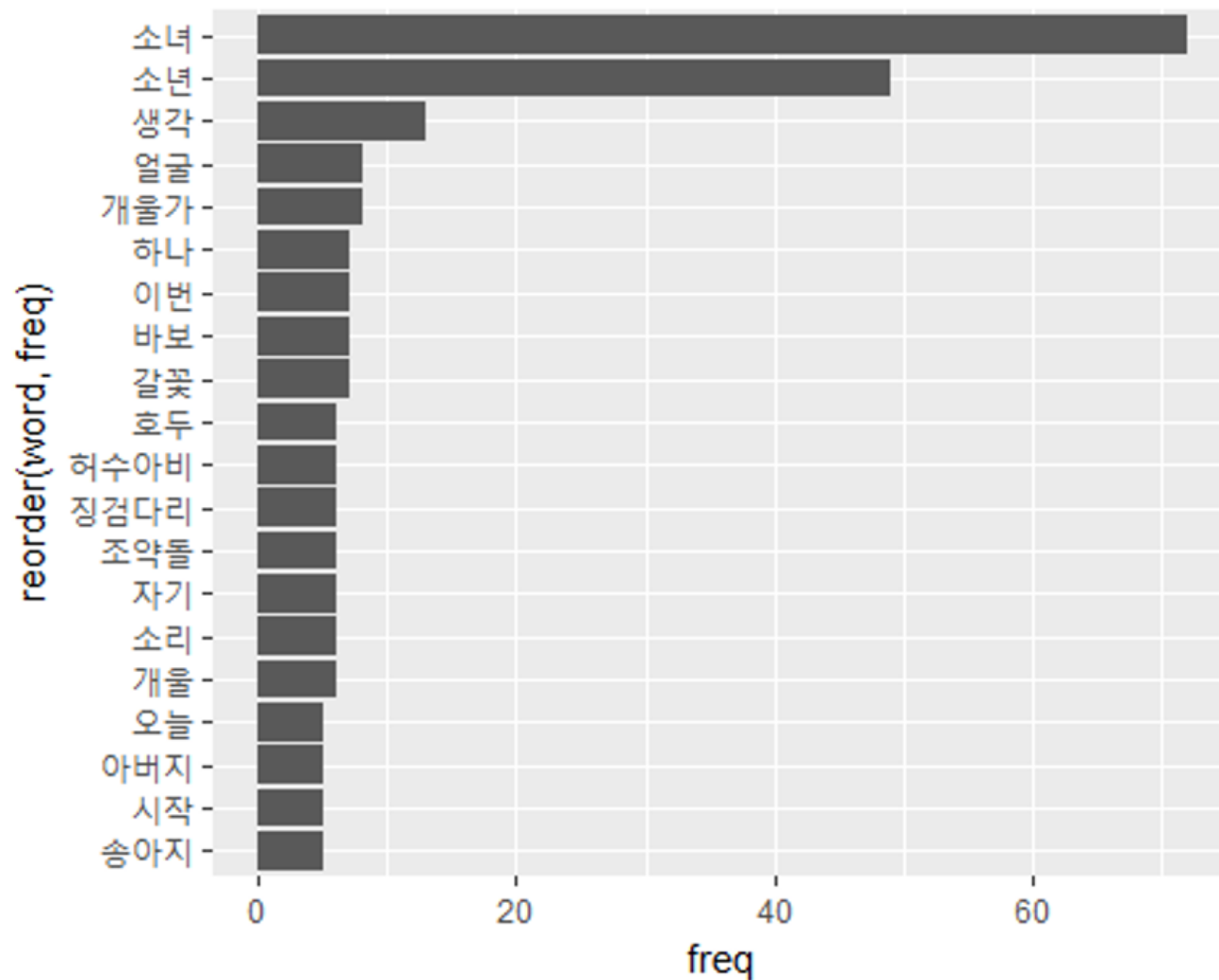
### 3. Arriving to frequency table! (docs to freqTable)

```
freqTable <- data.frame(table(docs)) ↓  
names(freqTable) <- c("word", "freq") ↓  
freqTable <- freqTable %>% arrange(desc(freq)) ↓  
head(freqTable) ↓
```

##		word	freq ↓
## 1		소녀	72 ↓
## 2		소년	49 ↓
## 3		생각	13 ↓
## 4		개울가	8 ↓
## 5		얼굴	8 ↓
## 6		갈꽃	7 ↓

## 4. Render Bar Chart (play with freqTable)

```
ggplot(head(freqTable, 20)) +  
  geom_bar(aes(x=reorder(word, freq), y=freq), stat="identity") +  
  coord_flip()
```



## 5. Render Word Cloud (play with `freqTable`)

```
wordcloud(words = freqTable$word, freq = freqTable$freq,
  min.freq = 1, max.words=100, random.order=FALSE, rot.per=0.35,
  colors=brewer.pal(8, "Dark2"))
```



# Summary

- *Text-mining Process (plain vanilla)*

0. Setup

1. Infile and prepare text to “docs”

2. Cleanup the text (“docs”)

3. Arriving to frequency table (“docs” to “freqTable”)

4. Render barplot

5. Render wordcloud

- *Question*

- 한/영에는 어떤 공통점이 있나요?

- Step (0), 3, 4, 5, 6

- 한/영에는 어떤 차이가 있나요?

- Step 1, 2

- 다음 모듈에서는 어떤 프로그램을 작성해 볼까요?



## *Module 3*

### *Writing a function*

(길고, 전형적이고, 반복적인 코드는  
함수로 묶는 것이 바람직!)



## 2. Cleanup the text (docs)

```
docs <- unlist(docs)
docs <- Filter(function(x) {nchar(x) >= 2}, docs) # Character Length >= 2
```

## 3. Arriving to frequency table! (docs to freqTable)

```
freqTable <- data.frame(table(docs))
names(freqTable) <- c("word", "freq")
freqTable <- freqTable %>% arrange(desc(freq))
```

I have a dream  
long = "er"

## 2. Cleanup the text (docs)

```
toSpace <- content_transformer(
  function (x , pattern) gsub(pattern, " ", x))
docs <- docs %>%
  tm_map(toSpace, "/") %>% tm_map(toSpace, "@") %>% tm_map(toSpace, "\\|")
docs <- docs %>%
  tm_map(content_transformer(tolower)) %>% # Convert it to lower case
  tm_map(removeNumbers) %>% # Remove numbers
  tm_map(removeWords, stopwords("english")) %>% # Remove english common stopwords
  tm_map(removeWords, c("blabla1", "blabla2")) %>% # Remove your own stop word
  tm_map(removePunctuation) %>% # Remove punctuations
  tm_map(stripWhitespace) # Eliminate extra white spaces
```

## 3. Arriving to frequency table! (docs to freqTable)

```
termMat <- TermDocumentMatrix(docs)
termTable <- as.matrix(termMat)
freqTable <- data.frame(word = rownames(termTable),
                        freq = rowSums(termTable))
freqTable$word <- rownames(freqTable)
freqTable <- freqTable %>% arrange(desc(freq))
```

I have a dream  
long = "er"

```

cleanDocsGenerateFreqTable <- function(docs, lang) {
  activate(c("tm", "SnowballC", "wordcloud", "konlp", "pdftools"))
  activate(c("ggplot2", "dplyr", "RColorBrewer"))
  if (lang == "kr") {
    docs <- unlist(docs)
    docs <- Filter(function(x) {nchar(x) >= 2}, docs) # character length >= 2
    freqTable <- data.frame(table(docs))
    names(freqTable) <- c("word", "freq")
    freqTable <- freqTable %>% arrange(desc(freq))
  } else { # lang == "en"
    toSpace <- content_transformer(
      function(x, pattern) gsub(pattern, " ", x))
    docs <- docs %>%
      tm_map(toSpace, "/") %>%
      tm_map(toSpace, "@") %>%
      tm_map(toSpace, "\\|")
    docs <- docs %>%
      tm_map(content_transformer(tolower)) %>% # Convert it to lower case
      tm_map(removeNumbers) %>% # Remove numbers
      tm_map(removeWords, stopwords("english")) %>% # Remove english common stopwords
      tm_map(removeWords, c("blabla1", "blabla2")) %>% # Remove your own stop word
      tm_map(removePunctuation) %>% # Remove punctuations
      tm_map(stripWhitespace) # Eliminate extra white spaces
    termMat <- TermDocumentMatrix(docs)
    termTable <- as.matrix(termMat)
    freqTable <- data.frame(word = rownames(termTable),
                           freq = rowSums(termTable))
    freqTable$word <- rownames(freqTable)
    freqTable <- freqTable %>% arrange(desc(freq))
  }
  return(freqTable)
}

```

```
1 activate <- function(package_name) {  
2   needsInstall <- !package_name %in% installed.packages()[,"Package"]  
3   if (sum(needsInstall)) {  
4     install.packages(package_name[needsInstall])  
5   }  
6   dummy <- sapply(package_name, function(x) require(x, character.only=TRUE))  
7 }  
8  
9 cleanDocsGenerateFreqTable <- function(docs, lang) {  
10   activate(c("tm", "SnowballC", "wordcloud", "konLP", "pdftools"))  
11   activate(c("ggplot2", "dplyr", "RColorBrewer"))  
12   if (lang == "kr") {  
13     docs <- unlist(docs)  
14     docs <- Filter(function(x) {nchar(x) >= 2}, docs) # character length >= 2  
15     freqTable <- data.frame(table(docs))  
16     names(freqTable) <- c("word", "freq")  
17     freqTable <- freqTable %>% arrange(desc(freq))  
18   } else { # lang == "en"  
19     toSpace <- content_transformer(  
20       function(x, pattern) gsub(pattern, " ", x))  
21     docs <- docs %>%  
22       tm_map(toSpace, "/") %>%  
23       tm_map(toSpace, "@") %>%
```

- 다른 .R 파일에 저장!
- 이 함수를 사용하기 위해서는 `source("LSR.R")`을 실행하면 됨!



# *Module 4*

## *flexdashboard*

```

week3 - word cloud - dream + sonaki.R... * x
[Icons] [Knit]

1 ---
2 title: "word Cloud - dream & 소나기 "
3 author: "Sim"
4 date: "`r Sys.Date()`"
5 output: word_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = FALSE)
10 knitr::opts_chunk$set(message = FALSE)
11 knitr::opts_chunk$set(warning = FALSE)
12 ```
13
14 ```{r}
15 source("LSR.R")
16 activate(c("tm", "SnowballC", "wordcloud", "koNLP", "pdftools"))
17 activate(c("ggplot2", "dplyr", "RColorBrewer"))
18 ```
19
20 ## 1. I have a dream
21
22 ```{r}
23 filePath <- paste0("http://www.sthda.com/sthda/RDoc/example-files/",
24                   "martin-luther-king-i-have-a-dream-speech.txt")
25 text <- readLines(filePath)
26 docs <- Corpus(VectorSource(text))
27 lang <- "en"
28 ```
29

```

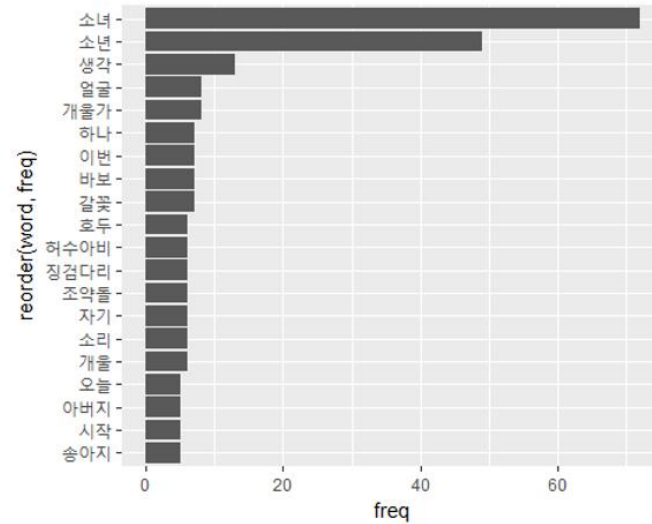
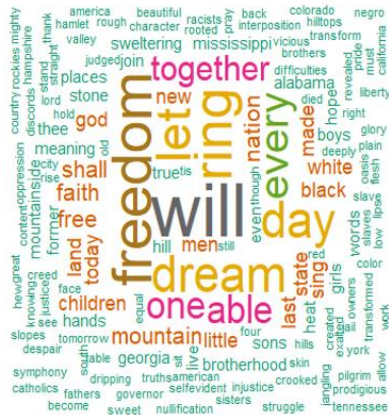
```

30 ```{r}
31 freqTable <- cleanDocsGenerateFreqTable(docs, lang)
32 ggplot(head(freqTable, 20)) +
33   geom_bar(aes(x=reorder(word, freq), y=freq), stat="identity") +
34   coord_flip()
35 wordcloud(words = freqTable$word, freq = freqTable$freq,
36           min.freq = 1, max.words=200, random.order=FALSE, rot.per=0.35,
37           colors=brewer.pal(8, "Dark2"))
38 ```
39
40 ## 2. 소나기
41
42 ```{r}
43 text <- pdf_text("sonaki.pdf")
44 docs <- sapply(text, extractNoun, USE.NAMES = F) # Apply extract Noun
45 docs <- unlist(docs)
46 docs <- Filter(function(x) {nchar(x) >= 2}, docs) # character length >= 2
47 lang <- "kr"
48 ```
49
50 ```{r}
51 freqTable <- cleanDocsGenerateFreqTable(docs, lang)
52 ggplot(head(freqTable, 20)) +
53   geom_bar(aes(x=reorder(word, freq), y=freq), stat="identity") +
54   coord_flip()
55 wordcloud(words = freqTable$word, freq = freqTable$freq,
56           min.freq = 1, max.words=200, random.order=FALSE, rot.per=0.35,
57           colors=brewer.pal(8, "Dark2"))
58 ```

```

## 2. 소나기

word	freq
will	17
freedom	13
ring	12.5
let	11
dream	11
day	10.5
every	9
one	8.5
able	8
together	7
shall	4.5
nation	4.5
mountain	4.5
free	4
faith	4
today	3.5
state	3.5
men	3.5
little	3.5
children	3





# Flexdashboard – easy interactive dashboards for r

- R MARKDOWN을 사용하여 관련 데이터 시각화 그룹을 대시 보드로 게시하십시오.
- HTMLWIDGETS를 포함한 다양한 구성 요소 지원; 기본, 격자 및 격자 그래픽; 표 형식의 데이터; 게이지 및 가치 상자; 및 텍스트 주석.
- 유연하고 쉽게 행 및 열 기반 레이아웃을 지정할 수 있습니다. 구성 요소는 브라우저를 채우고 모바일 장치에 표시되도록 지능적으로 크기 조정됩니다.
- 시각화 시퀀스 및 관련 해설을 제시하는 스토리 보드 레이아웃.
- SHINY를 사용하여 시각화를 동적으로 구동 할 수 있습니다.
- [HTTPS://RMARKDOWN.RSTUDIO.COM/FLEXDASHBOARD/INDEX.HTML](https://rmarkdown.rstudio.com/flexdashboard/index.html)
- 파일 – 새파일 – RMARKDOWN – TEMPLATE – FLEXDASHBOARD!

# Sample flexdashboard Layouts

## Overview

This page includes a variety of sample layouts which you can use as a starting point for your own dashboards.

When creating a layout, it's important to decide up front whether you want your charts to fill the web page vertically (changing in height as the browser changes) or if you want the charts to maintain their original height (with the page scrolling as necessary to display all of the charts).

This behavior is controlled via the `vertical_layout` output option, which defaults to `vertical_layout: fill`. Filling the page is generally a good choice when you have only one or two charts vertically stacked. Alternatively you can use `vertical_layout: scroll` to specify a scrolling layout, which is generally a better choice for three or more charts vertically stacked.

- 개요
- 이 페이지에는 자신의 대시 보드의 시작점으로 사용할 수 있는 다양한 샘플 레이아웃이 포함되어 있습니다.
- 레이아웃을 만들 때 차트가 웹 페이지를 수직으로 채우고 싶을 때 (브라우저가 바뀌면 높이가 변경됨) 차트가 원래 높이를 유지하게 할 것인지 결정할 필요가 있습니다. 모든 차트를 표시하십시오.)
- 이 동작은 `vertical_layout` 출력 옵션을 통해 제어되며, 기본값은 `vertical_layout: fill`입니다. 일반적으로 하나 또는 두 개의 차트 만 세로로 쌓아두면 페이지 채우기가 좋습니다. 또는 `vertical_layout: scroll`을 사용하여 스크롤 레이아웃을 지정할 수 있습니다. 이 레이아웃은 일반적으로 세 개 이상의 세로로 겹쳐진 차트에 더 좋습니다.



# Chart Stack (Fill)

This layout is a simple stack of two charts. Note that one chart or the other could be made vertically taller by specifying the `data-height` attribute.

```
1 ---
2 title: "Chart Stack"
3 output: flexdashboard::flex_dashboard
4 ---
5
6 ### Chart 1
7 ```${r}```
8 ```
9
10
11
12 ### Chart 2
13 ```${r}```
14 ```
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
```

Chart 1

Chart 2

- 이 레이아웃은 두 차트의 간단한 스택입니다. 하나의 차트 또는 다른<sup>47</sup> 차트는 데이터 높이 특성을 지정하여 세로로 더 크게 만들 수 있습니다.

# Chart Stack (Scrolling)

This layout is a simple stack of three charts. To provide enough room to display all the charts a scrolling layout is used (`vertical_layout: scroll`). Note that because of its ability to scroll this layout could easily accommodate many more charts (although for large numbers of charts you might consider organizing them into [Multiple Pages](#)).

```
1 ---
2 title: "Chart Stack (Scrolling)"
3 output:
4   flexdashboard::flex_dashboard:
5     vertical_layout: scroll
6 ---
7
8 ### Chart 1
9
10 ```{r}
11 ```
12
13 ### Chart 2
14
15 ```{r}
16 ```
17
18 ### Chart 3
19
20 ```{r}
21 ```
22
23
24
25
```

Chart 1

Chart 2

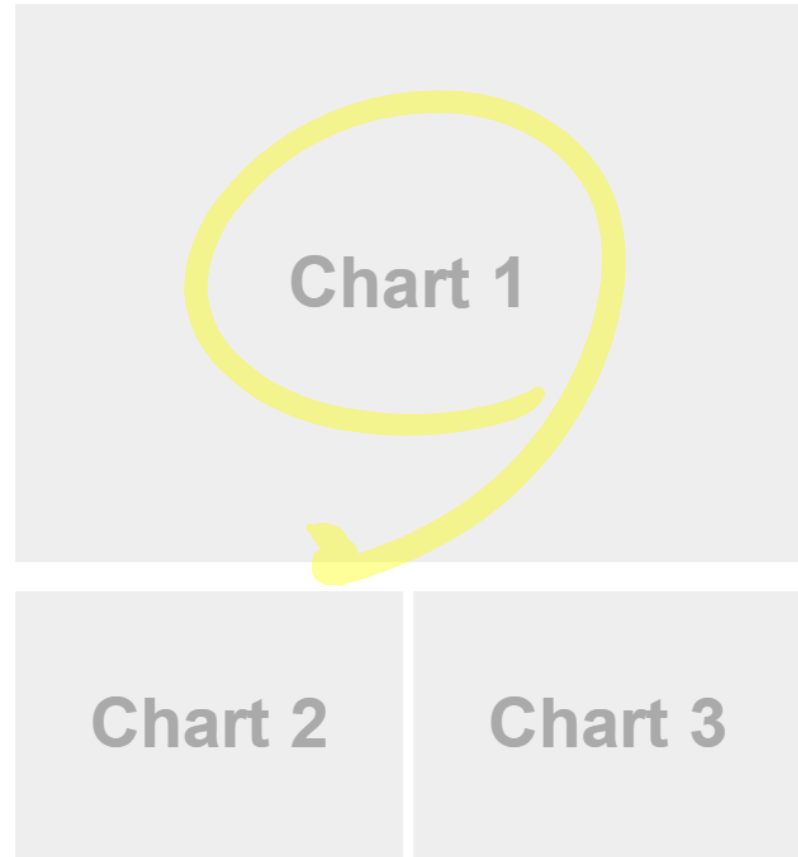
Chart 3

- 이 레이아웃은 세 개의 차트로 구성된 간단한 스택입니다. 모든 차트를 표시 할 수있는 충분한 공간을 제공하기 위해 스크롤 레이아웃이 사용됩니다 (`vertical_layout : 스크롤`). 이 레이아웃을 스크롤 할 수 있기 때문에 많은 수의 차트를 여러 페이지로 구성 할 수 있지만 차트를 더 쉽게 수용 할 수 있습니다.

## Focal Chart (Top)

This layout fills the page completely and gives prominence to a single chart at the top (with two secondary charts included below). To achieve this layout it uses `orientation: rows` and specifies `data-height` attributes on each row to establish their relative sizes.

```
1 ---
2 title: "Focal Chart (Top)"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: rows
6 ---
7
8 Row {data-height=650}
9 -----
10
11 ### Chart 1
12
13 ```{r}
14 ```
15
16 Row {data-height=350}
17 -----
18
19 ### Chart 2
20
21 ```{r}
22 ```
23
24 ### Chart 3
25
26 ```{r}
27 ```
28
```



- 이 레이아웃은 페이지를 완전히 채우고 상단의 단일 차트 (두 개의 보조 차트 포함)에 중요성을 부여합니다. 이 레이아웃을 달성하기 위해 `orientation: rows`를 사용하고 각 행의 데이터 - 높이 속성을 지정하여 상대 크기를 설정합니다.

# Focal Chart (Left)

This layout fills the page completely and gives prominence to a single chart on the left (with two secondary charts included to the right). Note that `data-width` attributes are specified on each column to establish their relative sizes.

```
1 ---
2 title: "Focal Chart (Left)"
3 output: flexdashboard::flex_dashboard
4 ---
5
6 Column {data-width=600}
7 -----
8
9 ### Chart 1
10
11 ```{r}
12 ```
13
14 Column {data-width=400}
15 -----
16
17 ### Chart 2
18
19 ```{r}
20 ```
21
22 ### Chart 3
23
24 ```{r}
25 ```
26
```



- 이 레이아웃은 페이지를 완전히 채우고 왼쪽의 단일 차트에 두드러진 보조 차트를 제공합니다 (오른쪽에 두 개의 보조 차트 포함). 데이터 너비 특성은 각 열에 지정되어 상대 크기를 설정합니다.

## Chart Grid (2x2)

This layout is a 2x2 grid of charts. This layout uses the default `vertical_scroll: fill` behavior however depending on the ideal display size for the charts it might be preferable to allow the page to scroll ( `vertical_layout: scroll` ). Note also that `orientation: rows` is used to ensure that the chart baselines line up horizontally.

```
1 |---
2 |title: "Row Orientation"
3 |output:
4 |  flexdashboard::flex_dashboard:
5 |    orientation: rows
6 |---
7 |
8 |Row
9 |-----
10|
11|### Chart 1
12|```{r}
13|
14|
15|
16|### Chart 2
17|```{r}
18|
19|
20|
21|Row
22|-----
23|
24|### Chart 3
25|```{r}
26|
27|
28|
29|### Chart 4
30|```{r}
31|
32|
33|
```

Chart 1

Chart 2

Chart 3

Chart 4

- 이 레이아웃은 2x2 그리드 차트입니다. 이 레이아웃에서는 기본 `vertical_scroll : fill` 비헤이비어를 사용하지만 차트의 이상적인 표시 크기에 따라 페이지를 스크롤하는 것이 바람직할 수 있습니다 (`vertical_layout : scroll`). 오리엔테이션 : 행은 차트 기준선이 수평으로 정렬되도록 하는 데 사용됩니다.

# Tabset Column

This layout displays the right column as a set of two tabs. Tabs are especially useful when you have a large number of components to display and prefer not to require the user to scroll to access everything.

```
1 |---
2 title: "Tabset Column"
3 output: flexdashboard::flex_dashboard
4 |---
5
6 Column
7 -----
8
9 ### Chart 1
10
11 ```{r}
12 ```
13
14 Column {.tabset}
15 -----
16
17 ### Chart 2
18
19 ```{r}
20 ```
21
22 ### Chart 3
23
24 ```{r}
25 ```
26
```

Chart 1

Chart 2

Chart 3

- 이 레이아웃은 오른쪽 열을 두 개의 탭으로 표시합니다. 탭은 표시 할 구성 요소 수가 많고 사용자가 모든 항목에 액세스 할 때 스크롤하지 않아도 되는 경우에 특히 유용합니다.<sup>46</sup>

# Tabset Row

This layout displays the bottom row as a set of two tabs. Note that the `{.tabset-fade}` attribute is also used to enable a fade in/out effect when switching tabs.

```
1 |---
2 title: "Tabset Row"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: rows
6   ---
7
8 Row
9 -----
10
11 ### Chart 1
12
13 ```{r}
14 ```
15
16 Row {.tabset .tabset-fade}
17 -----
18
19 ### Chart 2
20
21 ```{r}
22 ```
23
24 ### Chart 3
25
26 ```{r}
27 ```
28
```



- 이 레이아웃은 하단 행을 두 개의 탭으로 표시합니다. `{.tabset-fade}` 속성은 탭을 전환 할 때 페이드 인 / 아웃 효과를 사용하는 데에도 사용됩니다.

# Multiple Pages

This layout defines multiple pages using a level 1 markdown header ( ===== ). Each page has its own top-level navigation tab. Further, the second page uses a distinct orientation via the `data-orientation` attribute. The use of multiple columns and rows with custom `data-width` and `data-height` attributes is also demonstrated.

```
1 |---
2 |title: "Multiple Pages"
3 |output: flexdashboard::flex_dashboard
4 |---
5 |
6 |Page 1
7 |=====
8 |
9 |Column {data-width=600}
10 |-----
11 |
12 |### Chart 1
13 |
14 |~~~{r}
15 |
16 |
17 |Column {data-width=400}
18 |-----
19 |
20 |### Chart 2
21 |
22 |~~~{r}
23 |
24 |
25 |### Chart 3
26 |
27 |~~~{r}
28 |
29 |
30 |Page 2 {data-orientation=rows}
31 |=====
32 |
33 |Row {data-height=600}
34 |-----
35 |
36 |### Chart 1
37 |
38 |~~~{r}
39 |
40 |
41 |Row {data-height=400}
42 |-----
43 |
44 |### Chart 2
45 |
46 |~~~{r}
47 |
48 |
49 |### Chart 3
50 |
51 |~~~{r}
52 |
53 |
```

- 이 레이아웃은 레벨 1 마크 다운 헤더 (=====)를 사용하여 여러 페이지를 정의합니다. 각 페이지에는 고유 한 최상위 탐색 탭이 있습니다.
- 또한 두 번째 페이지는 데이터 방향 특성을 통해 고유 한 방향을 사용합니다. 사용자 정의 데이터 너비 및 데이터 높이 특성이있는 여러 열과 행을 사용하는 방법도 설명합니다.



# Storyboard

This layout provides an alternative to the row and column based layout schemes described above that is well suited to presenting a sequence of data visualizations and related commentary.

```
1 ---
2 title: "Storyboard Commentary"
3 output:
4   flexdashboard::flex_dashboard:
5     storyboard: true
6 ---
7
8 ### Frame 1
9
10 ````{r}
11 ````
12
13 ***
14
15 Some commentary about Frame 1.
16
17 ### Frame 2 {data-commentary-width=400}
18
19 ````{r}
20 ````
21
22 ***
23
24 Some commentary about Frame 2.
25
```

Note that the `storyboard: true` option is specified and that additional commentary is included alongside the storyboard frames (the content after the `***` separator in each section).

- 이 레이아웃은 위에 설명된 행 및 열 기반 레이아웃 스키마 대신 데이터 시각화 및 관련 해설 시퀀스를 표시하는 데 적합합니다.
- `storyboard: true` 옵션이 지정되고 추가 해설이 스토리보드 프레임 (각 섹션의 `***` 구분 기호 뒤의 내용)과 함께 포함됩니다.

# Input Sidebar

This layout demonstrates how to add a sidebar to a flexdashboard page (Shiny-based dashboards will often present user input controls in a sidebar). To include a sidebar you add the `.sidebar` class to a level 2 header ( ----- ):

```
1 |---
2 title: "Sidebar"
3 output: flexdashboard::flex_dashboard
4 runtime: shiny
5 ---
6
7 Inputs {.sidebar}
8 -----
9
10 ```{r}
11 # shiny inputs defined here
12 ```
13
14 Column
15 -----
16
17 ### Chart 1
18
19 ```{r}
20 ```
21
22 ### Chart 2
23
24 ```{r}
25 ```
26
```

Chart 1

Chart 2

- 이 레이아웃은 플렉스 대시 페이지에 사이드 바를 추가하는 방법을 보여줍니다 (샤이니 기반 대시 보드는 종종 사용자 입력 컨트롤을 사이드 바에 표시합니다). 사이드 바를 포함 시키려면 `.sidebar` 클래스를 레벨 2 헤더에 추가하십시오 (-----).

# Input Sidebar (Global)

If you have a layout that uses [Multiple Pages](#) you may want the sidebar to be global (i.e. present for all pages). To include a global sidebar you add the `.sidebar` class to a level 1 header ( ===== ):

```
1 ---
2 title: "Sidebar for Multiple Pages"
3 output: flexdashboard::flex_dashboard
4 runtime: shiny
5 ---
6
7 Sidebar {.sidebar}
8 =====
9
10 ```{r}
11 # shiny inputs defined here
12 ```
13
14 Page 1
15 =====
16
17 ### Chart 1
18
19 ```{r}
20 ```
21
22 Page 2
23 =====
24
25 ### Chart 2
26
27 ```{r}
28 ```
29
```

- 여러 페이지를 사용하는 레이아웃이있는 경우 사이드 바를 전역으로 (즉, 모든 페이지에 대해) 표시 할 수 있습니다.
- 전역 사이드 바를 포함 시키려면 레벨 1 헤더 (=====)에 `.sidebar`를 추가하세요.

# Mobile Specific

To customize your dashboard for display on small mobile screens you can either exclude selected components entirely or create mobile-specific variations of components. To exclude components you apply the `{.no-mobile}` class attribute. To use a mobile-specific rendering you create two identically titled components and apply the `{.mobile}` attribute to one of them.

For example, the following dashboard has a “Chart 1” that is included in mobile and desktop layouts, a “Chart 2” that is excluded from mobile layouts, and a “Chart 3” that has a custom variation for mobile:

```
1 |---
2 | title: "Mobile Specific"
3 | output: flexdashboard::flex_dashboard
4 | ---
5 |
6 | ### Chart 1
7 |
8 | ```{r}
9 | plot(cars)
10 | ```
11 |
12 | ### Chart 2 {.no-mobile}
13 |
14 | ```{r}
15 | plot(pressure)
16 | ```
17 |
18 | ## Chart 3
19 |
20 | ```{r}
21 | plot(mtcars)
22 | ```
23 |
24 | ## Chart 3 {.mobile}
25 |
26 | ```{r}
27 | plot(mtcars)
28 | ```
29 |
```

- 소형 모바일 화면에 표시 할 대시 보드를 사용자 정의하려면 선택한 구성 요소를 완전히 제외하거나 모바일 전용 변형을 만들 수 있습니다. 구성 요소를 제외하려면 `{.no-mobile}` 클래스 속성을 적용하십시오. 모바일 전용 렌더링을 사용하려면 동일한 제목의 두 개의 구성 요소를 만들고 그 중 하나에 `{.mobile}` 특성을 적용하십시오.
- 예를 들어, 다음 대시 보드에는 모바일 및 데스크톱 레이아웃에 포함 된 "차트 1", 모바일 레이아웃에서 제외 된 "차트 2" 및 모바일 용 사용자 지정 변형이있는 "차트 3"이 있습니다.

# Your Turn!

- 이번 모듈은 *I have a dream*과 소나기 두 개의 *text*를 불러와서, 분석하고, *barplot*과 *wordcloud*를 만드는 *rmarkdown* 예제로 시작합니다.
- 이를 *Flexdashboard*로 구현해 보세요.
- *Text* 파일 두 개를 더 제공해 드리니 *Flexdashdashboard* 레이아웃인 “*Multiple Pages*”를 참조해서 총 네 개의 *text*를 분석하는 대쉬보드를 만들어 보세요.



## IMPORT, TIDY, TRANSFORM, VISUALIZE, AND MODEL DATA

Hadley Wickham &  
Garrett Grolemund

Copyrighted Material

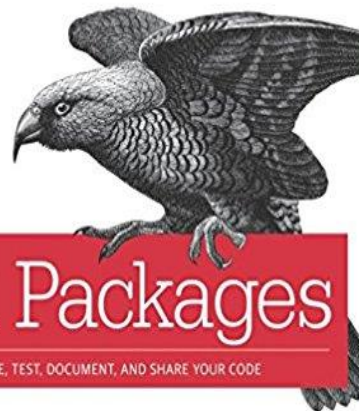
Use R!

Hadley Wickham

# ggplot2

## Elegant Graphics for Data Analysis

Second Edition



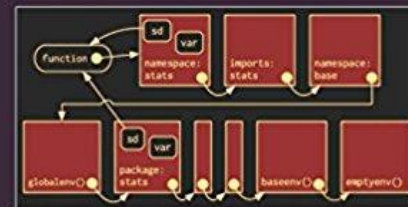
# R Packages

ORGANIZE, TEST, DOCUMENT, AND SHARE YOUR CODE

Copyrighted Material

Hadley Wickham

## Advanced R



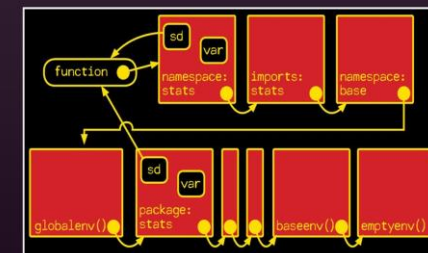
**Hadley Wickham**



Copyrighted Material

R을 R답게 사용하는 최고의 리소스!

# 해들리 위컴의 Advanced R



Hadley Wickham 지음 | 정우준 옮김







