L17. Dashboard (1)

Sim, Min Kyu, Ph.D., mksim@seoultech.ac.kr

서울과학기술대학교 데이터사이언스학과

The first step

flexdashboard

특성

- 과정보다는 시각화 객체의 결과물에 집중
- Advanced version of `rmarkdown`

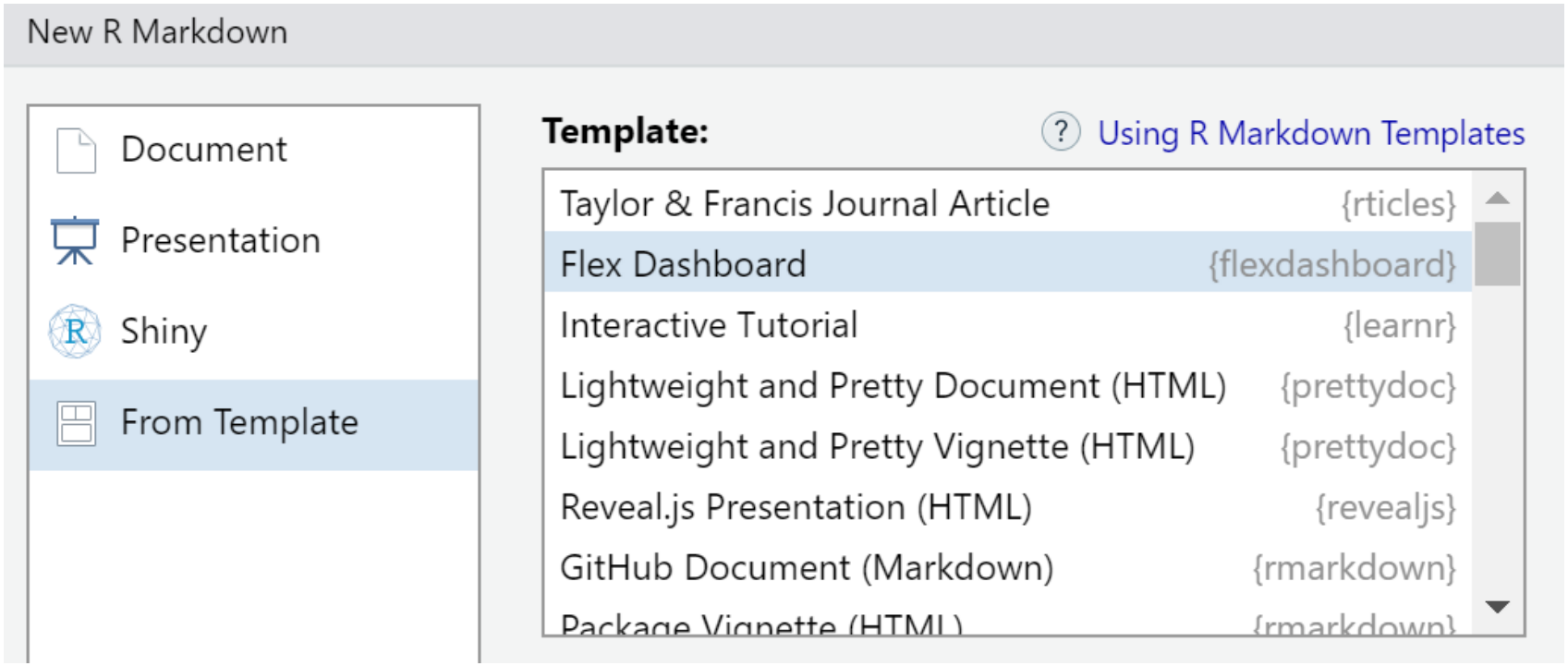
Header	in `rmarkdown`	in `flexdashboard`	component
Level-1	#	`==========`	page
Level-2	##	`----------`	a column or row
Level-3	###	###	an object

장점

- `markdown`의 문법이 그대로 적용되는 유연한 레이아웃
- html 포맷으로 제작
  - `htmlwidget` 객체 포함 가능
  - 모바일 장치 표현 지원
  - portability가 높음
- Automatic documentation (모니터링)

Getting Started

1. `install.packages("flexdashboard")`
2. 파일 – 새파일 – Rmarkdown – **Template** – Flexdashboard - Knit



ex1.Rmd

Knit | Insert | Run

```
---
title: "Untitled"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
---

```{r setup, include=FALSE}
library(flexdashboard)
```

Column {data-width=650}

Chart A

```{r}

```

Column {data-width=350}

Chart B

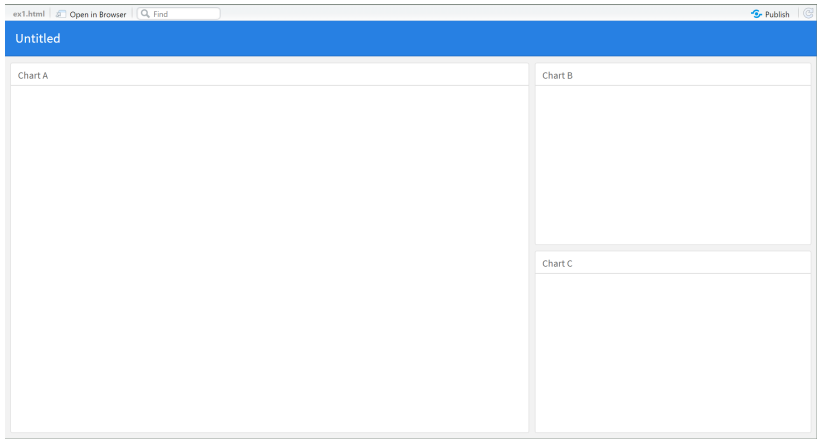
```{r}

```

Chart C

```{r}

```
```



Sample Layouts

`flexdashboard` 제작 프로세스

1. https://rmarkdown.rstudio.com/flexdashboard/layouts.html 에서 레이아웃 결정
2. Copy & Paste to your rstudio
3. Rendering 확인
4. Fill in the blanks

`flexdashboard/layouts.html`의 레이아웃들

	레이아웃	구성	Orientation
1	Chart Stack (Fill)	상하 fill	`column`
2	Chart Stack (Scrolling)	상하 scroll	`column`
3	Focal Chart (Top)	main-subs 상하	`row`
4	Focal Chart (Left)	main-subs 좌우	`column`
5	Chart Grid (2 by 2)	2 by 2	`row`
6	Tabset Column	main-subs 좌우 (tab)	`column`
7	Tabset Row	main-subs 상하 (tab)	`row`
8	Multiple Pages		
9	Input Sidebar		
10	Input Sidebar (Global)	8 + 9	

1. Chart Stack (Fill) - 상하 *fill* - `column`

- 두 차트의 간단한 스택입니다.
- 데이터 높이 특성을 지정하여 세로로 더 크게 만들 수 있습니다.

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.

````
---
title: "Chart Stack"
output: flexdashboard::flex_dashboard
---

### Chart 1

```{r}

```

### Chart 2

```{r}

```
````

2. Chart Stack (Scrolling) - 상하 scroll - column

- 공간을 확보하기 제공하기 위해 스택 (스크롤 레이아웃) 사용 (`vertical_layout: scroll`)
- 많은 수의 차트 포함 (더 많은 차트는 `Multiple Pages` 사용)

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).

````
---
title: "Chart Stack (Scrolling)"
output:
  flexdashboard::flex_dashboard:
    vertical_layout: scroll
---

### Chart 1

```{r}
```

### Chart 2

```{r}
```

### Chart 3

```{r}
```
````

3. Focal Chart (Top) - main-subs 상하 - row

- 상단의 차트에 중요성을 부여
- `orientation: rows`
- 각 행의 `data-height` 속성을 지정하여 상대 크기를 설정

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.

```
---
title: "Focal Chart (Top)"
output:
  flexdashboard::flex_dashboard:
    orientation: rows
---

Row {data-height=650}
-------------------------------------

### Chart 1

```{r}
```

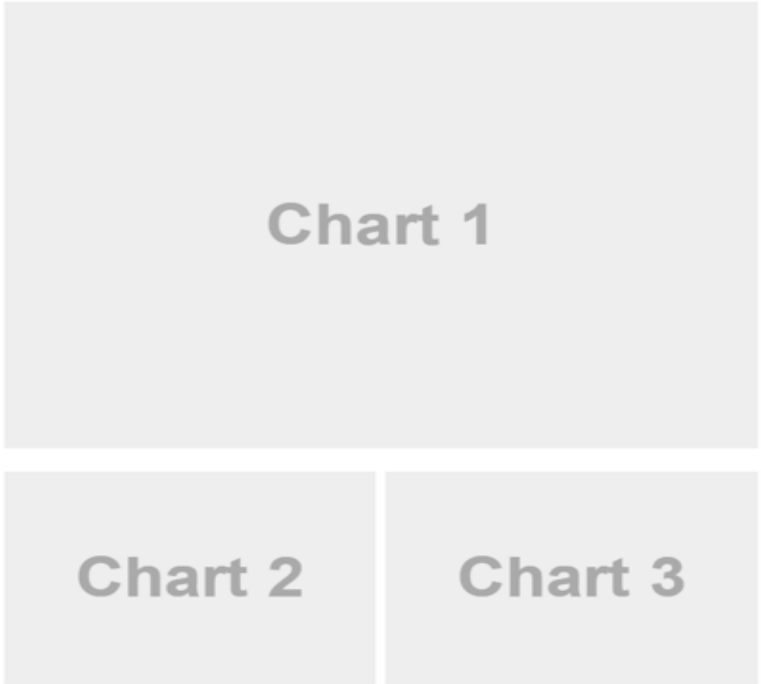
Row {data-height=350}

Chart 2

```{r}
```

Chart 3

```{r}
```
```



4. Focal Chart (Left) - main-subs 좌우 - column

- 왼쪽의 차트에 중요성을 부여
- `orientation: columns`는 default 이므로 입력없음
- 각 열의 `data-width` 속성을 지정하여 상대 크기를 설정

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.

```
---
title: "Focal Chart (Left)"
output: flexdashboard::flex_dashboard
---

Column {data-width=600}
-------------------------------------

### Chart 1

```{r}
```

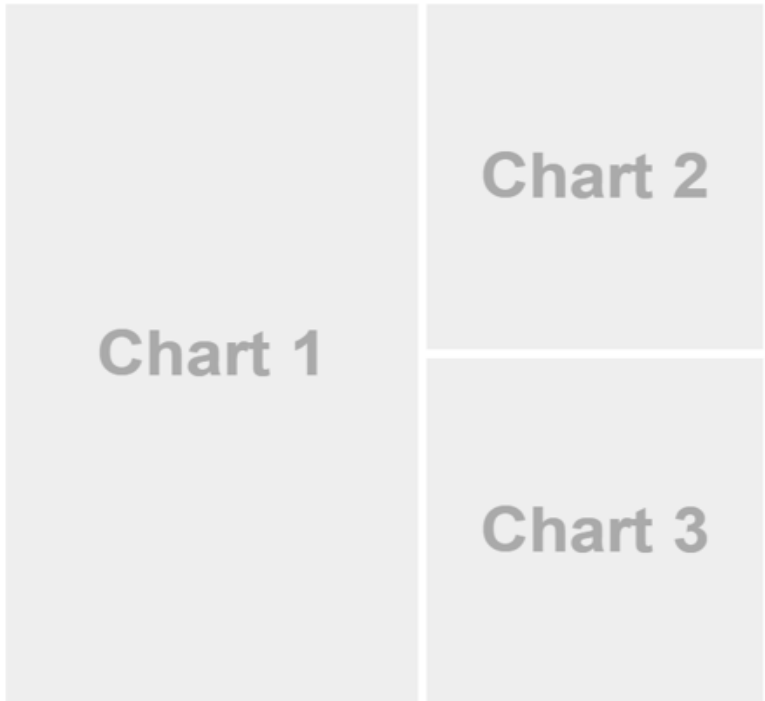
Column {data-width=400}

Chart 2

```{r}
```

Chart 3

```{r}
```
```



5. Chart Grid (2 by 2) - 2 by 2 - row

- `vertical_scroll: fill`이 default 이지만, 때로는 페이지를 스크롤하는 것(`vertical_layout: scroll`)이 바람직함
- `orientation: rows`는 차트 기준선을 수평으로 정렬시킴

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.

````
---
title: "Row Orientation"
output:
  flexdashboard::flex_dashboard:
    orientation: rows
---

Row
-------------------------------------

### Chart 1

```{r}
```

### Chart 2

```{r}
```

Row
-------------------------------------

### Chart 3

```{r}
```

### Chart 4

```{r}
```
````

6. Tabset Column - main-subs 좌우 (tab) - `column`

- 오른쪽 열을 두 개의 탭으로 표시
- 구성요소의 수가 많고 스크롤 않게 하는 경우에 유용
- 공간이 부족할 때 - 1) `scroll` vs 2) `tabset` vs 3) multiple pages

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.

```
---
title: "Tabset Column"
output: flexdashboard::flex_dashboard
---

Column
-------------------------------------

### Chart 1

```{r}
```

Column {.tabset}

Chart 2

```{r}
```

Chart 3

```{r}
```
```

7. Tabset Row - main-subs 상하 (tab) - row

- 하단 행을 두 개의 탭으로 표시
- {.tabset-fade}는 탭 전환시 페이드 인/아웃 효과를 줌

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.

```
---
title: "Tabset Row"
output:
  flexdashboard::flex_dashboard:
    orientation: rows
---

Row
-------------------------------------

### Chart 1

```{r}
```

Row {.tabset .tabset-fade}

Chart 2

```{r}
```

Chart 3

```{r}
```
```

8. Multiple Pages

- **Level-I 마크 다운 헤더(==========)를 사용하여 페이지 정의**
- `data-orientation`, `data-width`, `data-height` 사용을 확인

```
---
title: "Multiple Pages"
output: flexdashboard::flex_dashboard
---

Page 1
==================================

Column {data-width=600}
----------------------------------

### Chart 1

```{r}
```

Column {data-width=400}

Chart 2

```{r}
```

Chart 3

```{r}
```

Page 2 {data-orientation=rows}
==================================

Row {data-height=600}

Chart 1

```{r}
```

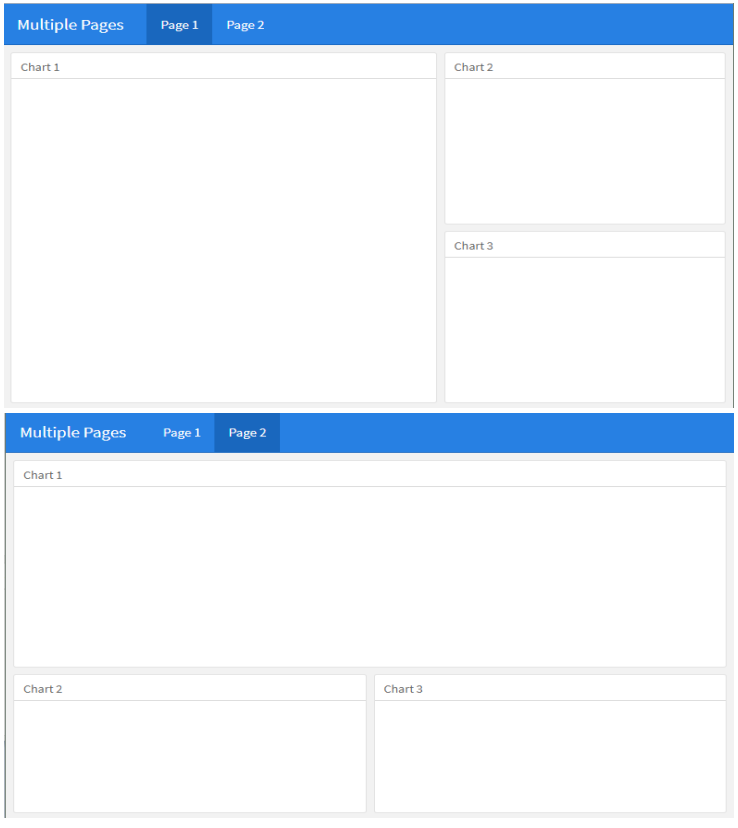
Row {data-height=400}

Chart 2

```{r}
```

Chart 3

```{r}
```
```



9. Input Sidebar

- 입력 컨트롤을 제공하는 사이드 바를 추가
- `.sidebar` 클래스를 레벨 2 헤더(`----------`)에 추가

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 (`----------`):

```
---
title: "Sidebar"
output: flexdashboard::flex_dashboard
runtime: shiny
---

Inputs {.sidebar}
-------------------------------------

```{r}
# shiny inputs defined here
```

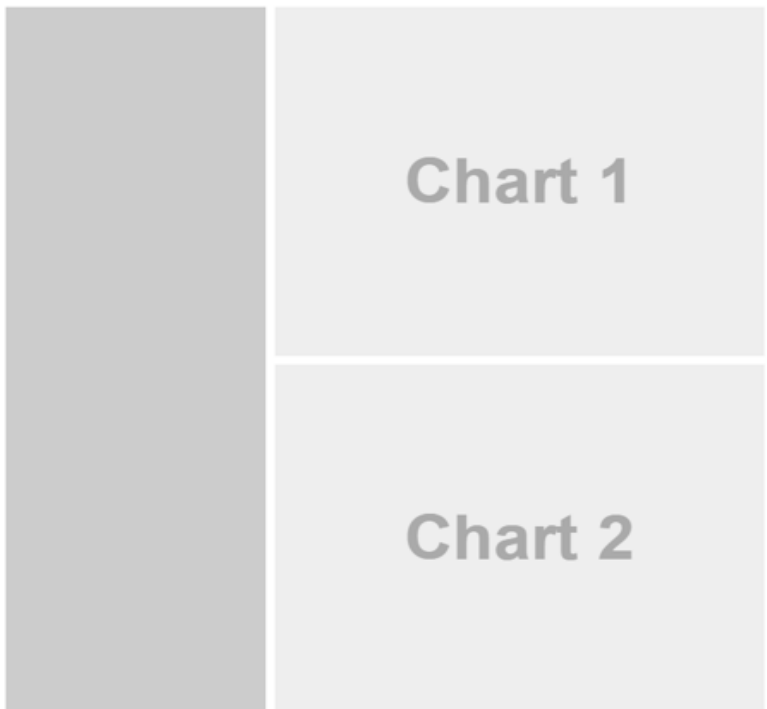
Column

Chart 1

```{r}
```

Chart 2

```{r}
```
```



10. Input Sidebar (Global) - Multiple Pages + Sidebar

- 전역(Global) 사이드 바를 포함하려면 레벨 1 헤더(==========)에 `.sidebar`를 추가
- 모든 페이지에서 사이드바 표시됨

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 (`==========`):

```
---
title: "Sidebar for Multiple Pages"
output: flexdashboard::flex_dashboard
runtime: shiny
---

Sidebar {.sidebar}
=====================================

```{r}
# shiny inputs defined here
```

Page 1
=====================================

Chart 1

```{r}
```

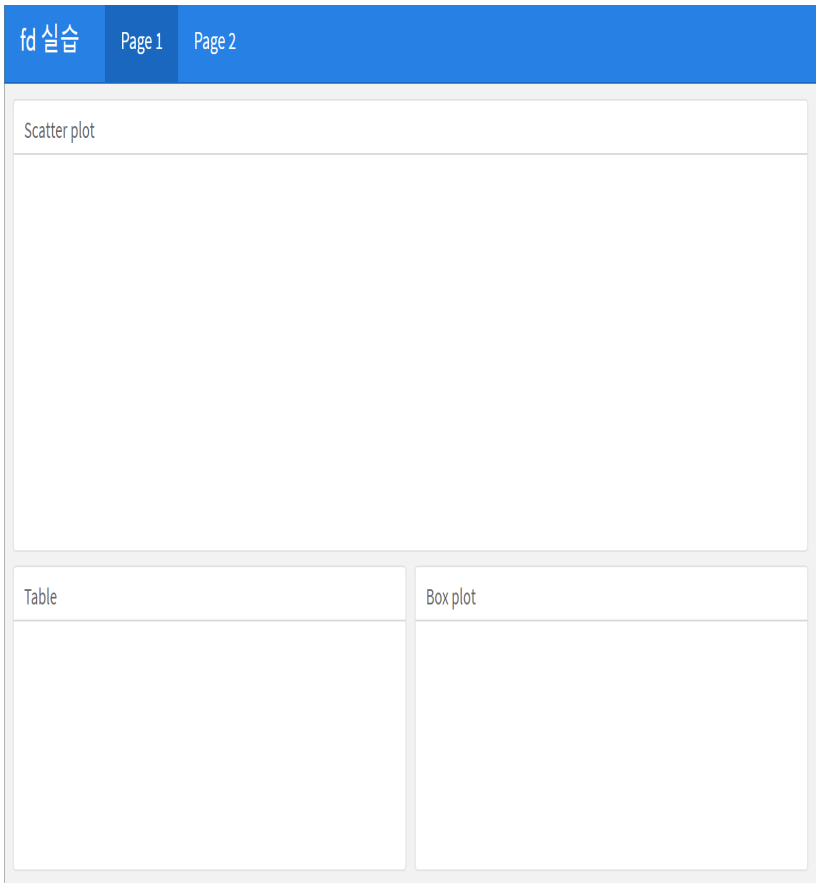
Page 2
=====================================

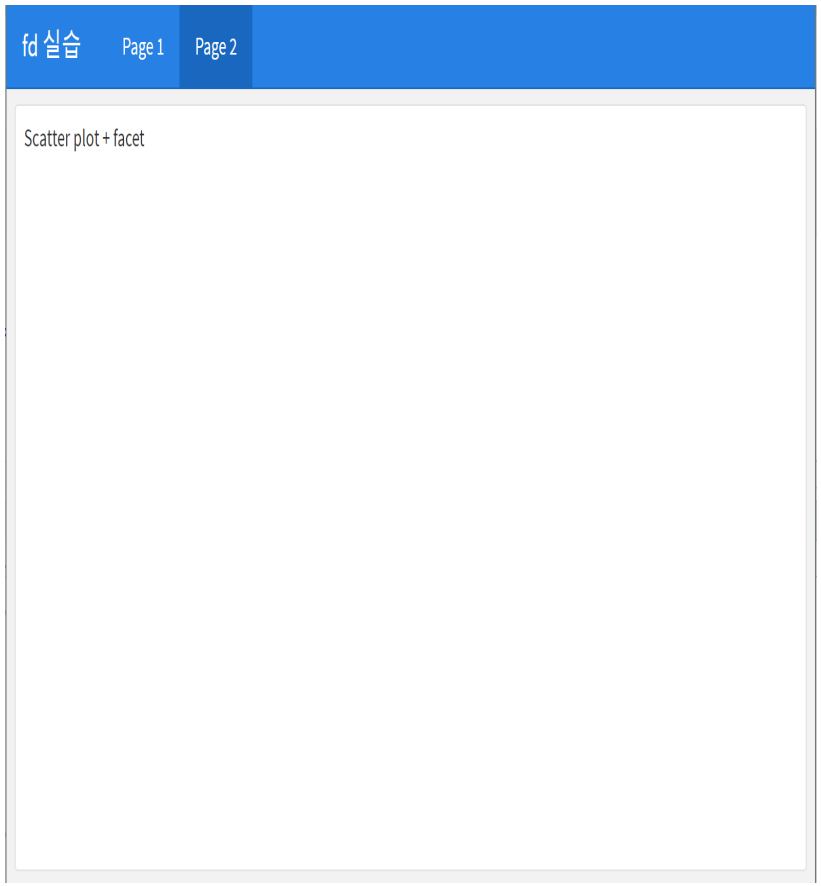
Chart 2

```{r}
```
```

실습!

Step 1. 레이아웃

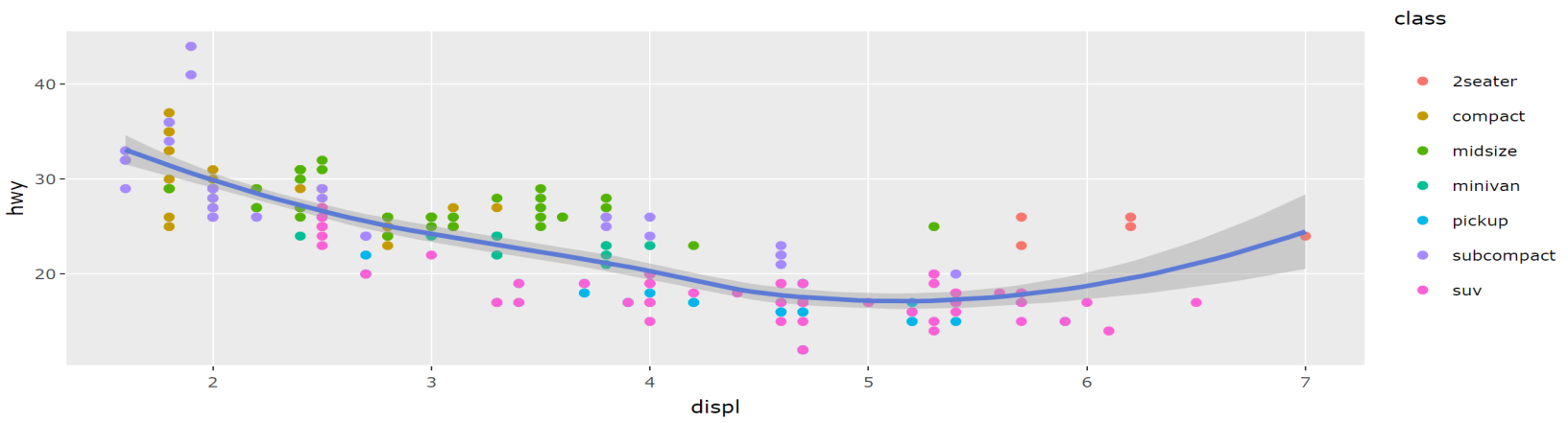

fd 실습
Page 1
Page 2
Scatter plot
Table
Box plot


fd 실습
Page 1
Page 2
Scatter plot + facet

Step 2. Fill in the blanks

fd 실습 | Page 1 | Page 2

Scatter plot

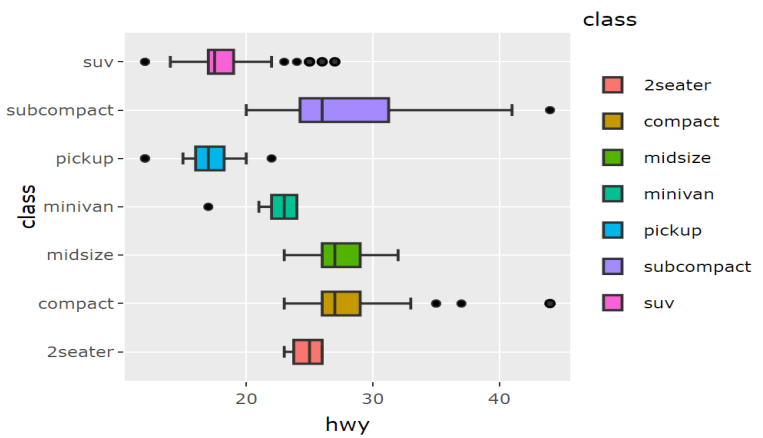


Table

class	2seater	compact	midsize	minivan	pickup	subcompact	suv
hwy	24.8	28.3	27.3	22.4	16.9	28.1	18.1
cty	15.4	20.1	18.8	15.8	13.0	20.4	13.5

Ref: https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html

Box plot




fd 실습
Page 1
Page 2
2seater
compact
midsize
minivan
pickup
subcompact
suv
class
2seater
compact
midsize
minivan
pickup
subcompact
suv
nwy
displ
40
30
20
2
3
4
5
6
7

Code

```
---
title: "fd 실습"
output: flexdashboard::flex_dashboard
---

Page 1 {data-orientation=rows}
=====================================

Row {data-height=500}
-------------------------------------

### Scatter plot

```{r}
library(ggplot2)
library(plotly)
a1 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color = class))
a2 <- a1 + geom_smooth()
ggplotly(a2)
```

Row {data-height=500}

Table

```{r}
library(dplyr)
library(kableExtra)

b <- mpg %>%
  group_by(class) %>%
  summarise(hwy = round(mean(hwy),1), cty = round(mean(cty),1)) %>%
  t()

kable(b) %>%
  kable_styling(bootstrap_options = "striped", full_width = FALSE)
```

Ref: <https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html>

Box plot

```{r}
c <- ggplot(mpg) + geom_boxplot(aes(x = class, y = hwy, fill = class)) + coord_flip()
ggplotly(c)
```

Page 2
=====================================

```{r}
a3 <- a1 + facet_wrap(~ class)
ggplotly(a3)
```
```

More examples

https://rmarkdown.rstudio.com/flexdashboard/examples.html

```
"Non scholae sed vitae discimus."
"우리는 학교를 위해서가 아니라 인생을 위해서 배운다."
```