

flexdashboard의 레이아웃

flexdashboard의 레이아웃에 대해 알아보기 위해,
flexdashboard의 default 템플릿에 레이아웃에 ggplot2 차트들만 추가하였다.

- 1) title, output 등이 설정 되어 있는 YAML 헤더 부분이 먼저 나오는데,
output 설정 부분에는 **orientation : columns**와 **vertical_layout: fill**이라는 부분이 보이고,

```
1 ---
2 title: "Dashboard Example"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7 ---
```

- 2) 이어 백틱 3 개(`````)로 둘러싸인 r 코드청크가 나오며,
* 아래의 예시에서는 이 부분에서 library() 함수로 flexdashboard와 ggplot2 패키지를 불러왔음

```
9 ```{r setup, include=FALSE}
10 library(flexdashboard)
11 library(ggplot2)
12 
13 ```
```

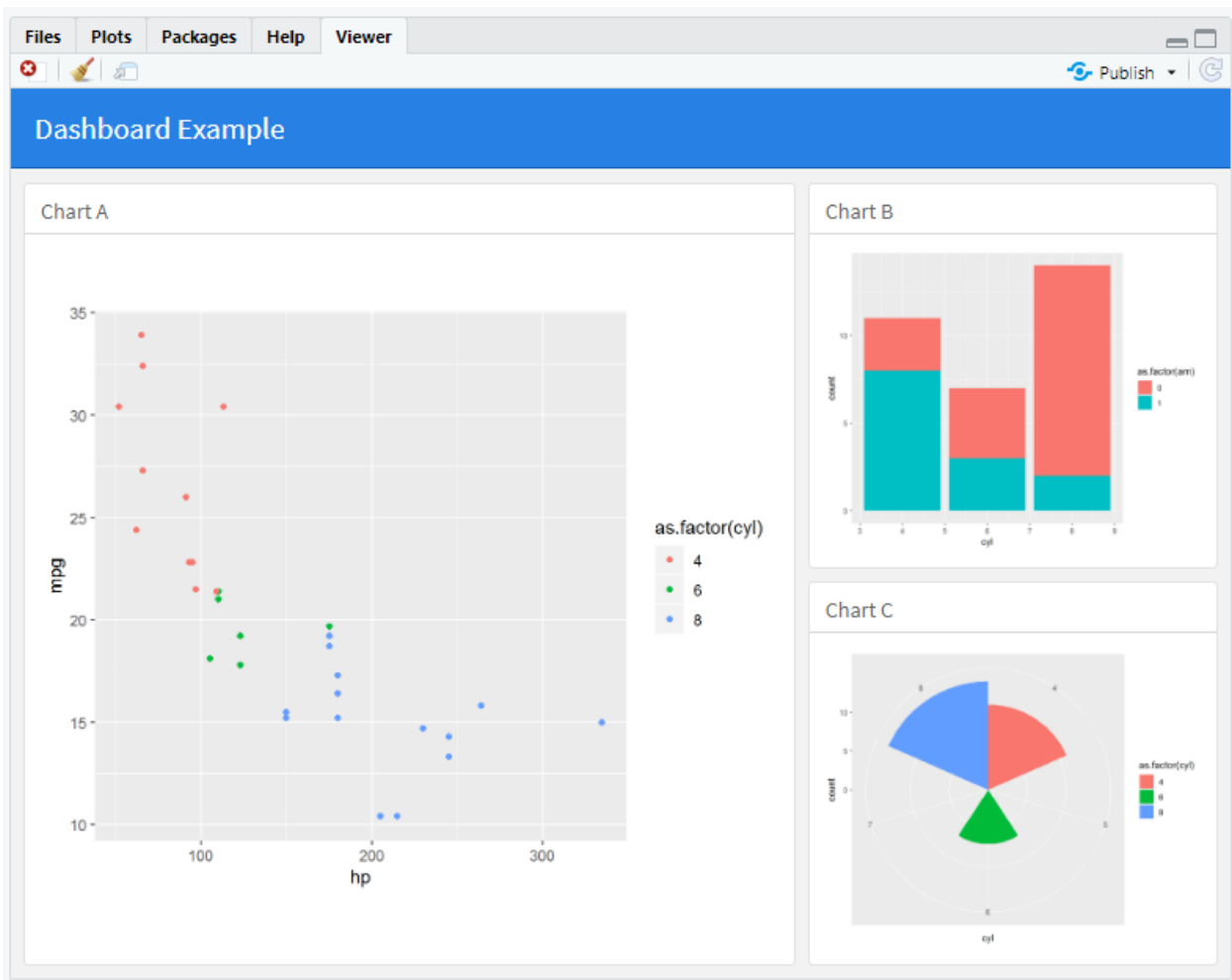
- 3) 그 아래에는 **Column** 이라고 기재된 부분이 2 군데 나온다.
그리고 각각의 Column 부분은 -----로 구분되어 있고, 그 밑에는 ###로 표시된 chart 부분이 나온다.

```
15 Column {data-width=650}
16 -----
17 
18 ### Chart A
19 
20 ```{r}
21 ggplot(data=mtcars, aes(x=hp, y=mpg, color=as.factor(cyl))) +
22   geom_point()
23 
24 ```
25 
26 Column {data-width=350}
27 -----
28 
29 ### Chart B
30 
31 ```{r}
32 ggplot(data=mtcars) +
33   geom_bar(mapping = aes(x=cyl, fill=as.factor(am)))
34 
35 ```
36 
37 ### Chart C
38 
39 ```{r}
40 ggplot(data=mtcars) +
41   geom_bar(mapping = aes(x=cyl, fill=as.factor(cyl)), position="dodge") +
42   coord_polar()
43 
44 ```
```

그리고 이 default 템플릿을 실행하면
다음과 같은 구조의 output 이 표시된다는 것도 이미 살펴보았다.

- 2 개의 열로 구분
- 1 열은 한 개의 행, 2 열은 두 개의 행으로 구성
- 그리고 열과 행의 내용들은 사용자의 브라우저 화면 크기에 맞게 자동으로
넓이와 높이가 조정됨

이를 실행하면 다음과 같이 대시보드가 생성되어 보여진다.



flexdashboard의 레이아웃 부분에 대해 세부적으로 알아보도록 하자.
설명은 Rstudio의 flexdashboard 홈페이지에 게시된 내용을 참고하였다.



Sample flexdashboard Layouts

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 window is resized) or horizontally (changing in width as the browser window is resized).

<https://rmarkdown.rstudio.com/flexdashboard/layouts.html>

이미 살펴봤듯이,
flexdashboard의 기본 레이아웃은 YAML 헤더에서 설정하는 부분이 많다.

1. Orientation 레이아웃 : 행과 열의 방향(순서)

가. default는 열 -> 행

flexdashboard에서는 default로 열 -> 행의 방향으로 설정이 이루어진다.

이를 YAML 헤더에서 확인할 수 있는데,

output: 아래 부분의 orientation: columns 부분이다. 이 부분은 default이기 때문에, 삭제해도 무관하다.

```
1 ---
2 title: "Dashboard Example"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7 ---
```

열과 행의 구분은 ---- 와 ### 를 이용하는데, 사실 이건 markdown의 문법이다.
바로 markdown 헤더.

default인 orientation: columns 옵션 :

열은 ---, 행은 ###로 구분된다.

여기서 유의할 것은,

열은 항상 ----이고 행은 항상 ###로 구분되는 것이 아니라는 것이다.

orientation: columns 로 설정했기 때문이라는 것에 유의해야 한다.

* 아래에서 다시 살펴보겠지만,

orientation: rows 로 하면, 행은 ----, 열은 ###로 구분되게 됨.

구 분	내 용
열(column)	level 2 markdown 헤더인 -----(개수 무관)로 구분
행(row)	level 3 markdown 헤더인 ###(개수 3 개)로 구분

나. 행 -> 열 방향으로 바꾸려면 orientation: rows 로 변경

만약 행 -> 열의 순서로 설정이 이루어지기를 원한다면,

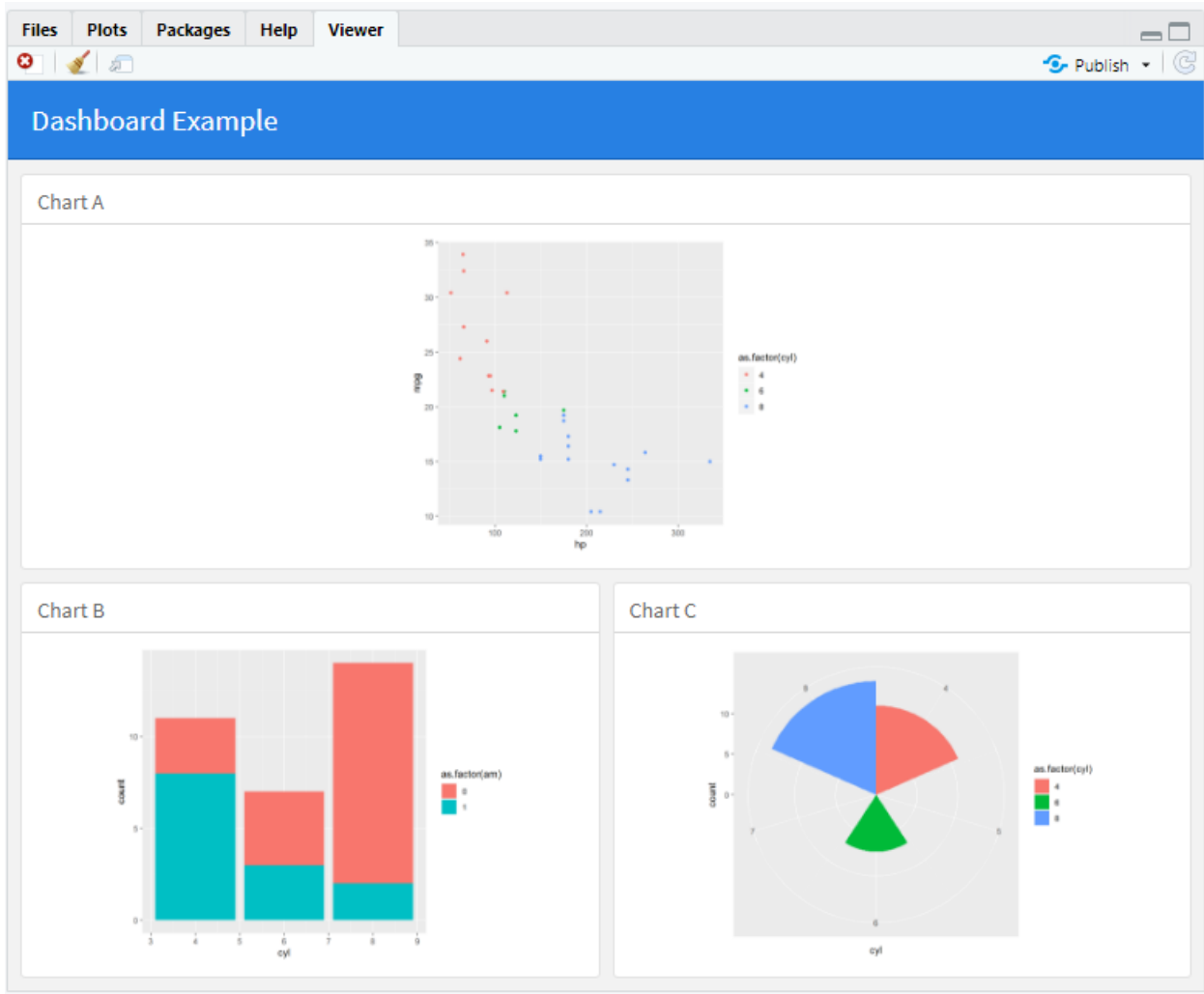
YAML 헤더의 orientation: 부분만 columns 를 rows 로 변경하면 된다.

```
1 ---
2 title: "Dashboard Example"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: rows
6     vertical_layout: fill
7 ---
```

그리고 knit 버튼을 눌러 실행해보면,

아래와 같이 2 개의 행이 먼저 구분되었으며

1 행은 1 개의 열, 2 행은 2 개의 행으로 구분되었음을 확인할 수 있다.



2. Scrolling 레이아웃

가. **default** 는 **fill** : 화면이 꽉 채워지도록 자동조정됨

flexdashboard 의 Scrolling 레이아웃은

생성되는 대시보드가 사용자의 화면에 맞게 리사이징되게 할지,
아니면 화면에 스크롤 기능을 넣을지를 설정하는 것이다.

이 역시 YAML 헤더에서 설정하며, **vertical_layout** 이다.

default 는 **vertical_layout: fill** 로 되어 있으며,

브라우저 크기를 조절하면 자동으로 화면 크기에 맞게 대시보드의 크기가 조정된다.

```
1 ---
2 title: "Dashboard Example"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7 ---
```

default 이니, vertical_layout:fill 은 삭제해도 무관,
역시 default 옵션인 orientation: columns 과 동시에 삭제하면
아래와 같이 error 가 뜨며 실행이 안됨에 유의

```
Console Terminal x R Markdown x
.../myFirstShinyApplication/myThirdReport_flexdashboard.Rmd
에러: Invalid YAML front matter (ends with ':')
실행이 중지되었습니다
```

제대로 나오게 하려면,
아래와 처럼 flexdashboard::flex_dashboard 바로 뒤에 있는 콜론(:)까지 삭제해야 한다.
그러면 정상적으로 실행된다.

```
1 ---
2 title: "Dashboard Example"
3 output:
4   flexdashboard::flex_dashboard
5 ---
```

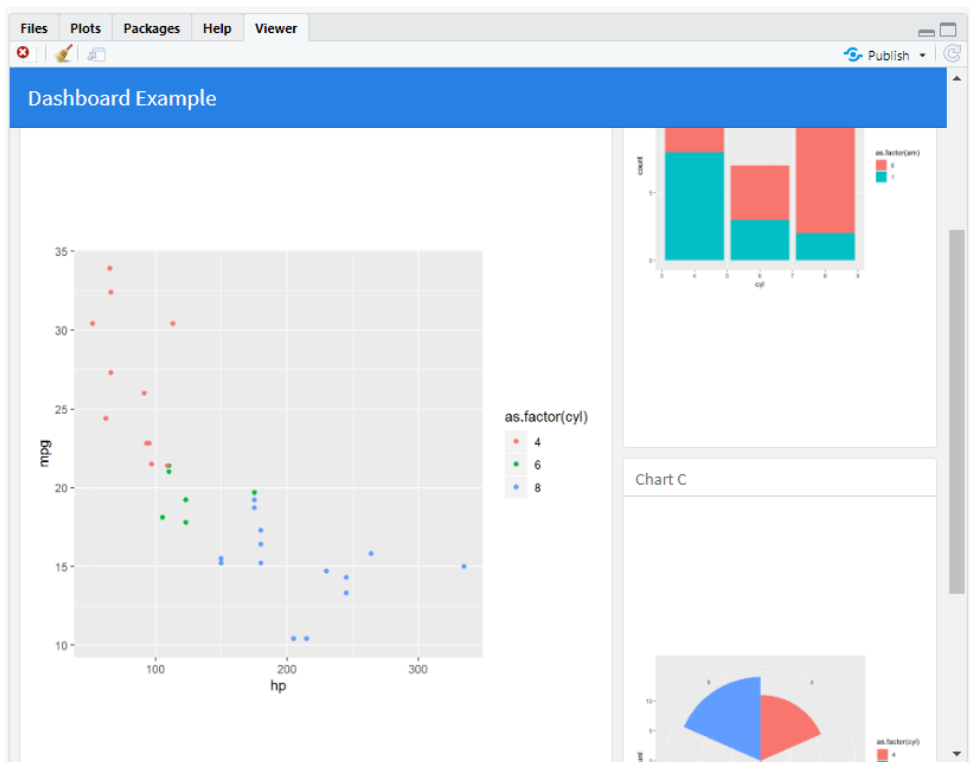
나. 스크롤바를 추가하려면 vertical_layout: scroll 로 변경

경우에 따라 브라우저 크기에 맞게 대시보드를 조정하지 않고,
스크롤바를 옆에 두고
위아래로 스크롤하며 대시보드를 보는 것을 원할 때도 있다.

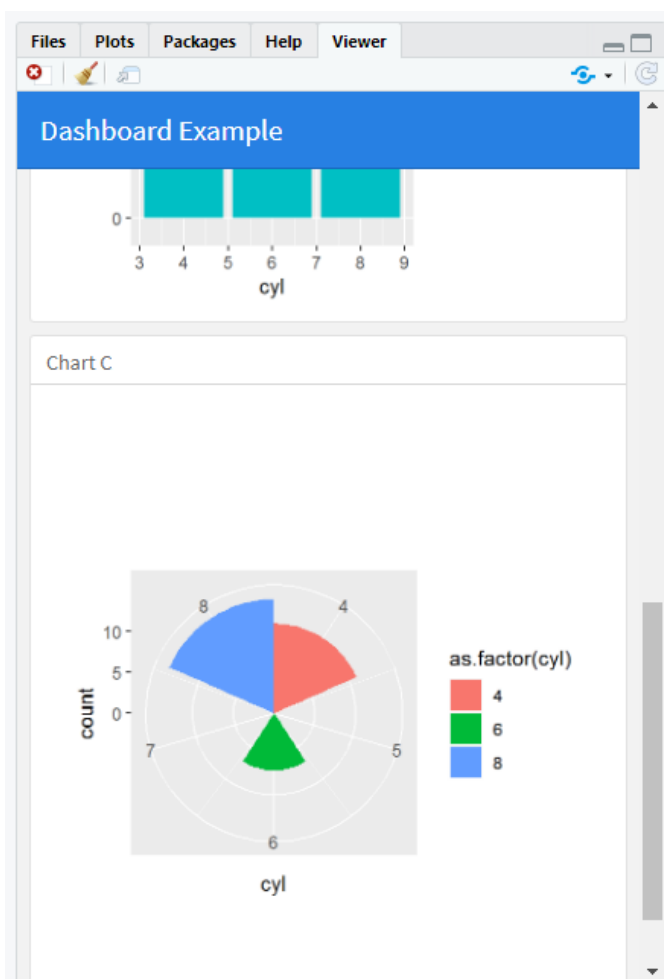
이 경우에는 YAML 헤더의 vertical_layout 을 fill 에서 scroll 로 변경하면 된다.
이 때, YAML 헤더에서는 들여쓰기(indentation)에 유의하자.

```
1 ---
2 title: "Dashboard Example"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: scroll
7 ---
```

vertical_layout 만 fill 에서 scroll 로 변경하여 실행해보면,
아래와 같이 대시보드의 오른쪽에 스크롤바가 생겼음을 확인할 수 있다.



이 옵션은 열이 2 개로 구분되어 있는 경우보다,
아래처럼 1 개의 열로 대시보드를 구성 후,
원래의 크기로 그래프 등을 나열하여 보여주고 싶을 때 유용.



3. tabset 레이아웃

대시보드의 한 행이나 열에
여러 개의 그래프 등의 자료를 보여주고 싶을 때에는,
한 화면에 모두 뿌려주는 것보다는 **tabset** 을 이용하는 것이 더 나을 수도 있다.

이는 특히,
어떤 컴포넌트는 메인 정보여서 모든 사용자에게 보여지길 원하고,
다른 컴포넌트는 부가적인 정보라서
관심 있는 사용자들만 볼 수 있게 하기를 원하는 경우에 유용할 수 있는데,

flexdashboard 홈페이지의 설명에 따르면,
많은 경우에 이 화면에서 정보를 찾아 들어가기 편한 tabset 이
그래프 등의 컴포넌트들을 상하로 나열하는 vertical_layout: scroll 옵션보다도
정보 전달 입장에서는 훨씬 더 효율적인 것이라고 제시한다.

flexdashboard 에서의 **tabset** 적용을 위해서는,
아래와 같이 **{.tabset}** 속성을 열 또는 행 구분할 때 기재하면 된다.

```
24 Column {.tabset}  
25 ▾ -----
```

아울러, **.tabset-fade** 를 **{.tabset}** 안에 기재하면,
탭 전환 시, 부드럽게 처리되는 효과까지 넣을 수 있다.

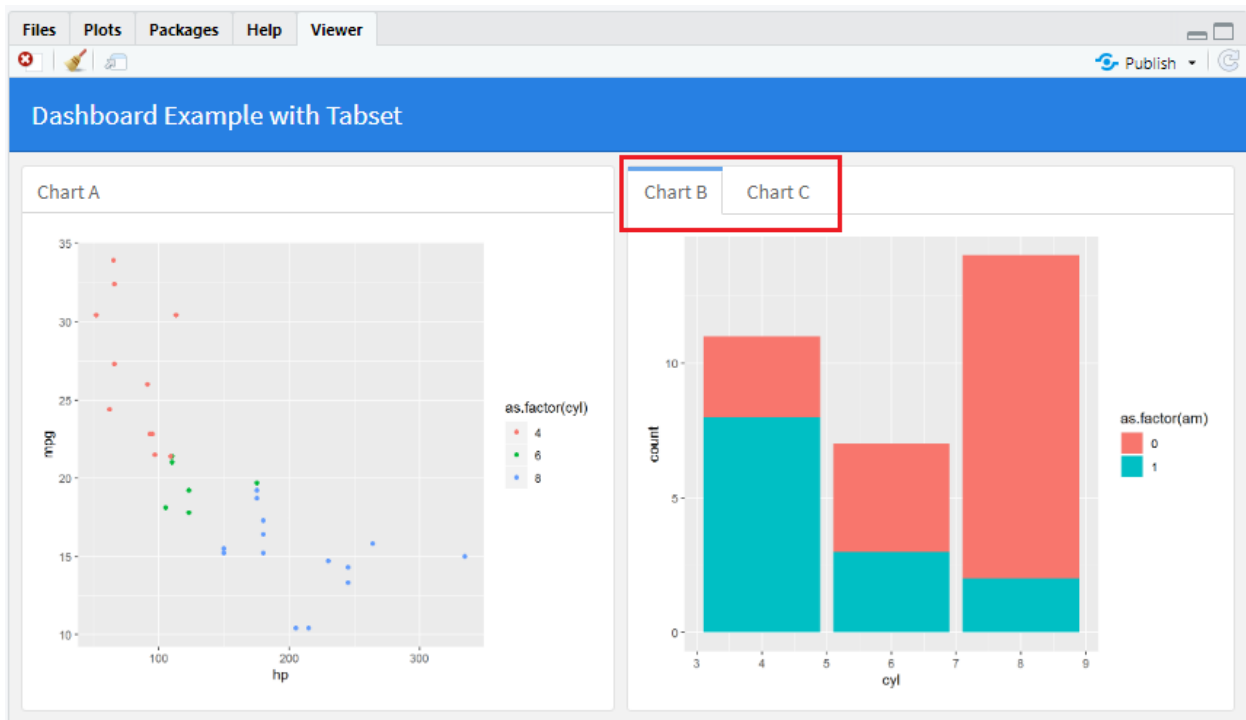
```
23  
24 Column {.tabset .tabset-fade}  
25 ▾ -----  
26
```


실습을 위해, 코드에 **tabset** 적용.

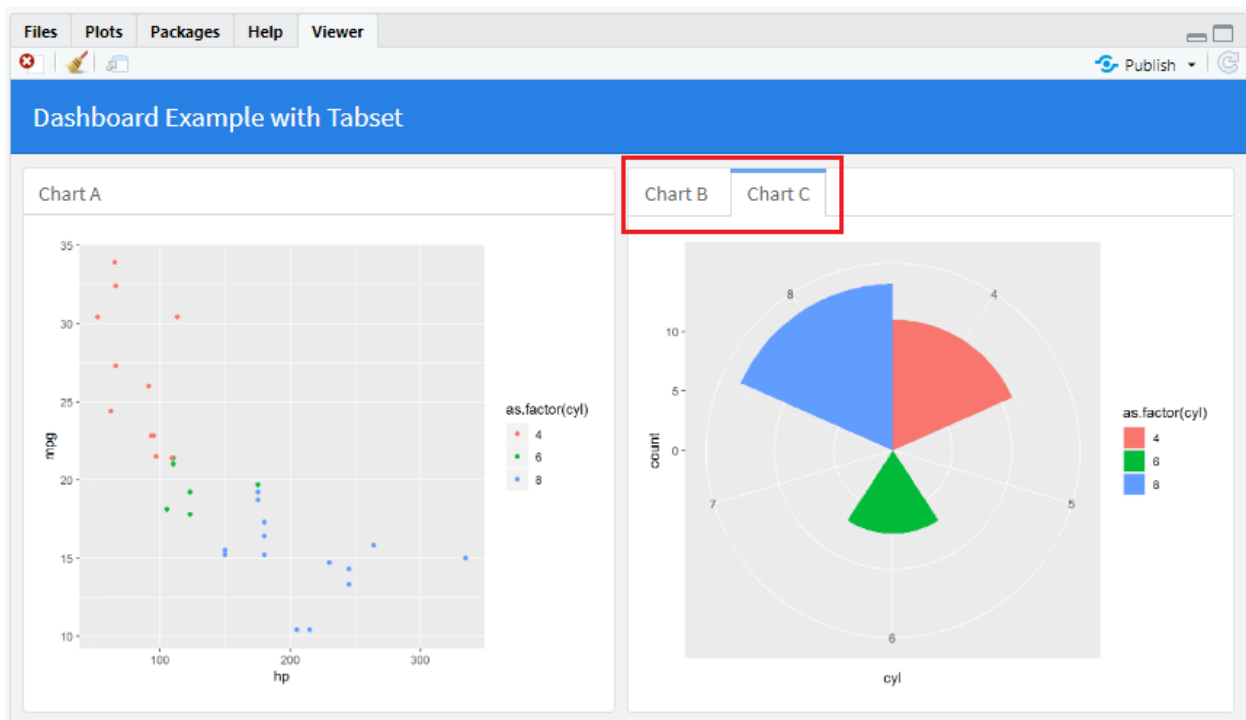
- YAML 헤더에서는 orientation 과 vertical_layout 부분을 삭제.
default 로 column(열) 방향 순서로 그려지며,
대시보드의 크기는 화면의 크기에 맞게 자동조정(fill)된다.
- 열의 크기는 별도로 설정하지 않았으므로 같은 크기가 되며,
두 번째 열에 **{.tabset .tabset-fade}**을 기재하여 tab 으로 구분될 수 있게 하였다.

```
myThirdReport_flexdashboard_tabset.R... * x
---
1 title: "Dashboard Example with Tabset"
2 output:
3   flexdashboard::flex_dashboard
4 ---
5
6
7 {r setup, include=FALSE}
8 library(flexdashboard)
9 library(ggplot2)
10
11
12
13 Column
14 -----
15
16 ### Chart A
17
18 {r}
19 ggplot(data=mtcars, aes(x=hp, y=mpg, color=as.factor(cyl))) +
20   geom_point()
21
22
23
24 Column {.tabset .tabset-fade}
25 -----
26
27 ### Chart B
28
29 {r}
30 ggplot(data=mtcars) +
31   geom_bar(mapping = aes(x=cyl, fill=as.factor(am)))
32
33
34
35 ### Chart C
36
37 {r}
38 ggplot(data=mtcars) +
39   geom_bar(mapping = aes(x=cyl, fill=as.factor(cyl)), position="dodge") +
40   coord_polar()
41
42
```

코드를 실행한 결과는 아래와 같다. 두 번째 열에 tab 이 생성되었음



두 번째 열에 있는 탭을 클릭해보면 부드럽게 탭 전환이 이루어지는 것도 확인 가능하고,



YAML 헤더의 orientation 설정을 변경하면,
아래와 같이 열이 아닌 행 단위로 tabset 을 적용할 수도 있고,
scroll 기능을 넣을 수도 있다.

```
1 ---  
2 title: "Dashboard Example with Tabset"  
3 output:  
4   flexdashboard::flex_dashboard:  
5     orientation: rows  
6     vertical_layout: scroll  
7 ---
```

