

## flexdashboard 의 Sizing(너비, 높이, 여백-padding) 설정

flexdashboard 의 레이아웃과 밀접한 관계가 있는 Sizing(크기 설정).

### 1. Width(너비)

flexdashboard 로 생성한 대시보드의 너비는,  
사용자가 쓰고 있는 브라우저의 너비에 의해 결정된다.

대시보드의 열이 1 개라면, 브라우저 너비를 모두 차지하게 되며,  
대시보드의 열이 여러 개라면 열마다 균등한 크기를 가지게 된다.

단, data-width 옵션을 적용 시 오버라이드하게 되는데,  
세부 내용은 아래의 "3. Size Attributes" 부분에서 더 자세히 알아보게 될 것이다.

### 2. Height(높이)

대시보드의 높이는 YAML 헤더의 vertical\_layout 이 fill 인지, scroll 인지에 따라 달라진다.

가. vertical\_layout: fill 의 경우

나.

- 대시보드의 크기는 사용자의 브라우저의 크기에 딱 맞게 조절됨
- 별도 data-height 옵션으로 크기를 설정하지 않는 한,  
세로로 적재되는(행 방향) chart 들은 같은 크기를 가지게 됨

나. vertical\_layout: scroll 의 경우

- 대시보드에 scroll 기능이 생기며,
- 이 때 chart 의 크기는 fig.height 옵션으로 설정하게 됨  
(default 는 5 인치(480 픽셀))

### 3. Size Attributes(크기 속성)

아무런 크기 속성도 설정되어 있지 않은 경우,  
개별 chart 의 크기는 default 로 6 x 4.8 인치(576 x 460 픽셀)인  
knitr figure size 에 비례하여 적용되는데,

data-width 와 data-height 속성을  
행(row), 열(column) 및 개별 chart 의 크기를 별도 설정할 수 있다.

이 속성은 { } 안에 기재하면 되는데,  
아래의 예시는 첫번째 열(column)의 너비를 300 픽셀로 정한 경우이다.

```
Column1 {data-width=300}
-----

### Gauge ex1. Contact Rate

```{r}
gauge(45, min = 0, max = 100, symbol = '%', sectors = gaugeSectors(
  success = c(80, 100), warning = c(40, 79), danger = c(0, 39)
))
```
```

두 번째 열의 크기까지 700 픽셀로 설정하고 knit 해보자.  
실행한 코드는 아래와 같다.

```
---
title: "Gauge, Navigation Bar & Text Annotation"
output:
  flexdashboard::flex_dashboard:
    orientations: columns
    source_code: embed
    navbar:
      - { title: "About", href: "https://example.com/about", align: right }
      - { icon: "fa-pencil", href: "https://www.naver.com", align: right }
---

This is an example. As you can see, flexdashboard can have text annotations.

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

```
library(ggplot2)
```

```
library(knitr)
```

```
```\n
```

```
Column1 {data-width=300}
```

```
-----\n
```

```
### Gauge ex1. Contact Rate
```

```
```\n{r}\n
```

```
gauge(45, min = 0, max = 100, symbol = '%', sectors = gaugeSectors(\n
```

```
  success = c(80, 100), warning = c(40, 79), danger = c(0, 39)\n
```

```
))\n
```

```
```\n
```

```
> Example from flexdashboard homepage
```

```
### Gauge ex2. Average Rating
```

```
```\n{r}\n
```

```
rating = 42
```

```
gauge(rating, 0, 50, label = 'Test', gaugeSectors(\n
```

```
  success = c(41, 50), warning = c(21, 40), danger = c(0, 20)\n
```

```
))\n
```

```
```\n
```

```
### Text Annotation
```

One of the dashboard section can be used as a text area.

Markdown Grammers can be helpful here.

It is not that difficult.

Just have a try.

Column2 {data-width=700}

### ggplot2 graph

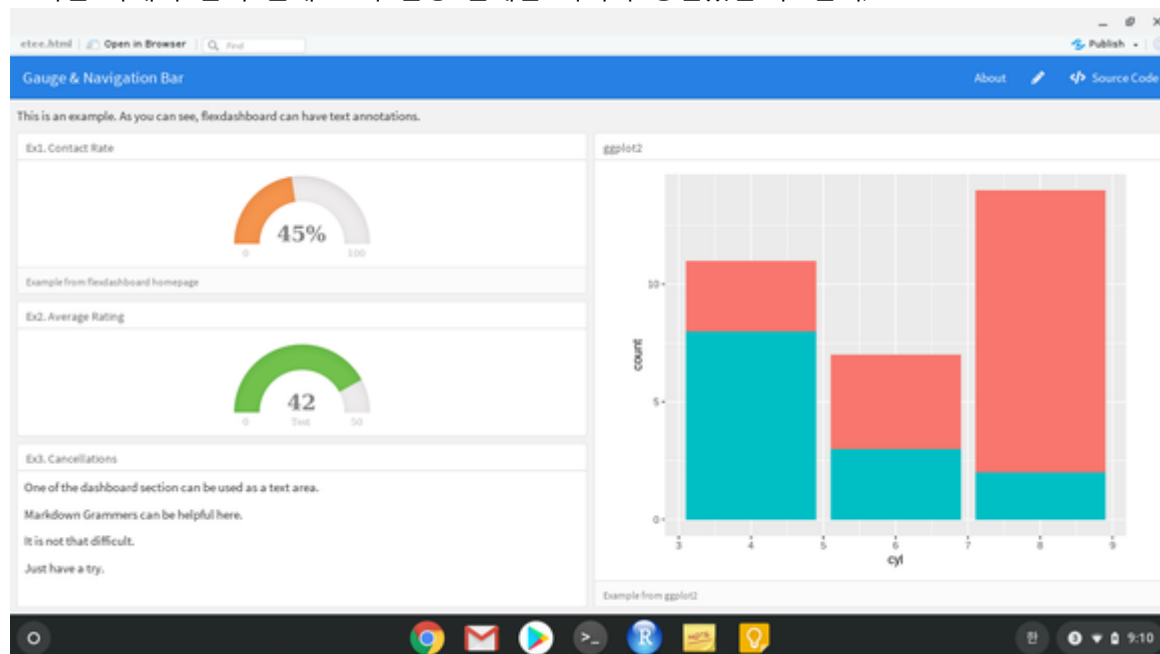
```
```{r}
```

```
ggplot(data=mtcars) +  
  geom_bar(mapping = aes(x=cyl, fill=as.factor(am))) +  
  theme(legend.position = "blank")
```

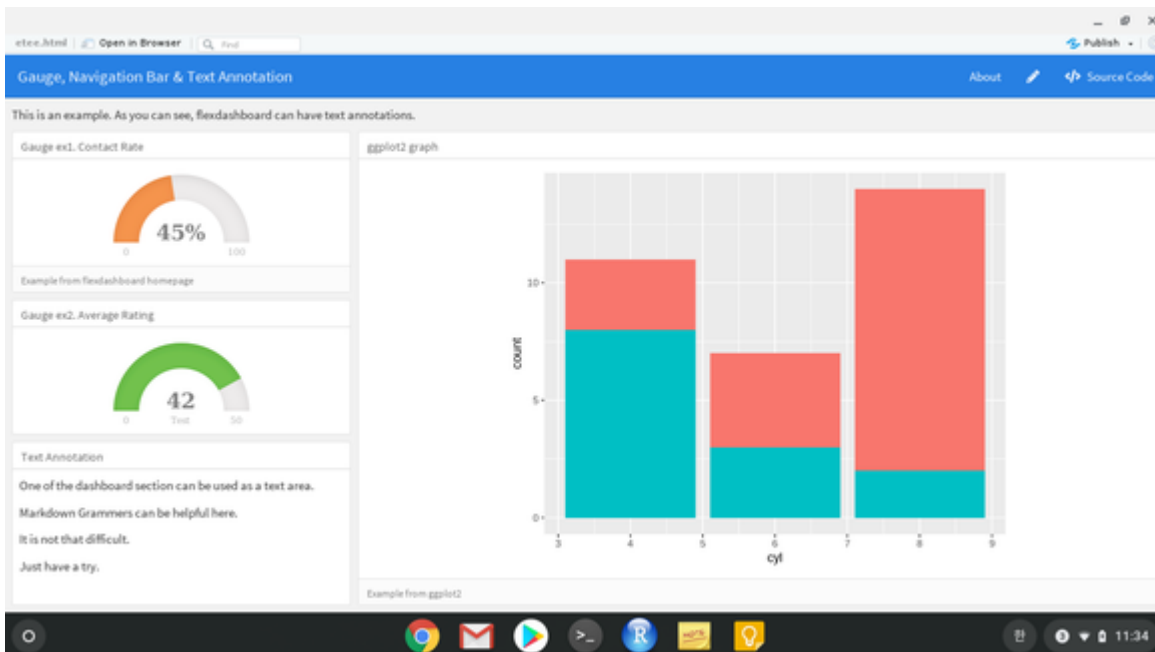
```
```
```

> Example from ggplot2

그러면 아래와 같이 원래 크기 설정 전에는 너비가 동일했던 두 열이,



아래와 같이 1 열은 300 픽셀, 2 열은 700 픽셀로 너비가 조정되었음을 확인할 수 있다.



#### 4. Chart Padding(여백)

flexdashboard 는 default 로  
대시보드 섹션 안에서 차트의 가장자리에 8 픽셀의 여백을 둔다.

이를 변경하기를 원하는 경우,  
차트의 r 청크코드에 해당하는 Level 3 마크다운 헤더(###)의 뒤에  
{no-padding}나 {data-padding=30} 등을 기재하면 된다.

아래와 같이 {no-padding}을 기재하는 경우  
여백공간 없이 차트가 대시보드 섹션을 가득 채우게 되며,

```
### No padding {no-padding}

```{r}
ggplot(data=mtcars) +
  geom_bar(mapping = aes(x=cyl, fill=as.factor(am))) +
  theme(legend.position = "blank")
```
```

{data-padding=30}와 같이  
기본 8 픽셀인 padding 수치를 원하는 크기로 변경하여 적용할 수도 있다.

```
### Padding 30 pixels {data-padding=30}

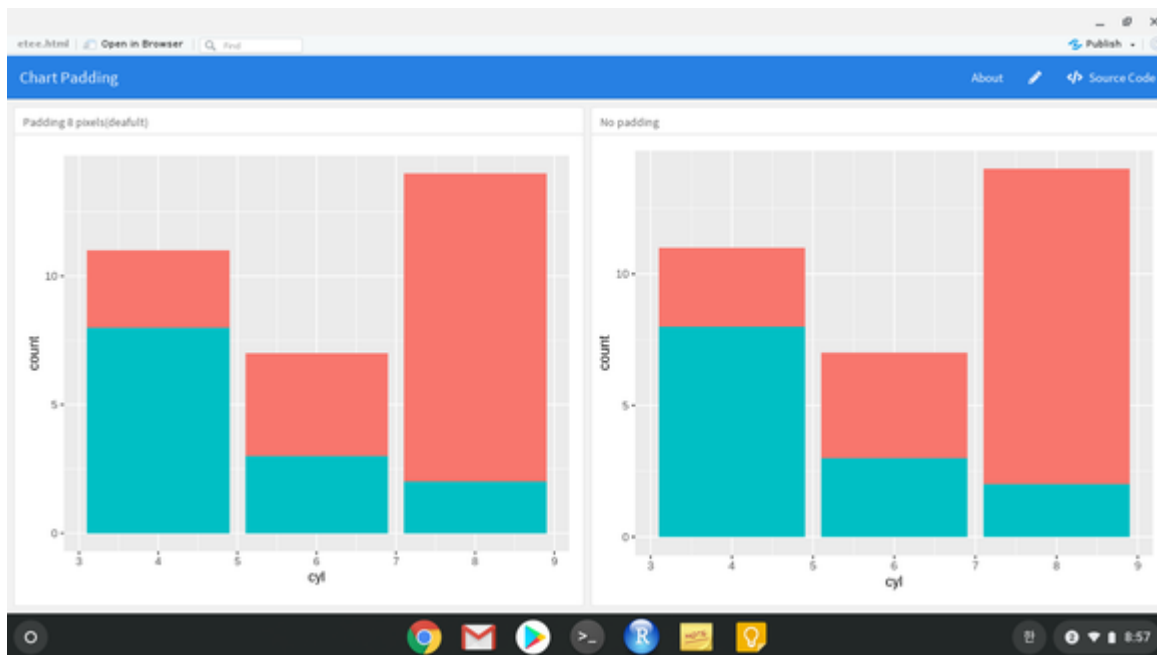
```{r}
```

```
ggplot(data=mtcars) +
  geom_bar(mapping = aes(x=cyl, fill=as.factor(am))) +
  theme(legend.position = "blank")
```

```

아래의 예시는 padding 을 조정하며 차트를 그려본 결과이다.

먼저 default 8 픽셀 padding 와 No padding 을 비교해보면 다음과 같다.  
 큰 차이가 없다고 할 수도 있으나,  
 그래프 오른쪽 좌우 및 위 쪽의 여백을 보면,  
 No padding 인 경우에 확실히 여백이 줄어있음을 확인할 수 있다.



그 다음, default 와 30 픽셀 padding 을 비교한 결과.  
 확실히 대시보드 섹션 안의 여백이 훨씬 많아졌다.

