

M32 - Flexdashboard

LearningSpoonsR

2018-07-07

Contents

Previously...

- Part 2. 분석 & 시각화 - `ggplot2`
- Part 1. Text 전처리 - `tm`, `KONLP`
- Part 3. 일반화 - Writing a function

Now...

- Part. 일반화 - `M32-flexdashboard`

Flexdashboard - easy & interactive dashboards for R

- R Markdown을 사용하여 관련 데이터 시각화 그룹을 대시 보드로 게시하십시오.
 - 1. `htmlwidgets`를 포함한 다양한 구성 요소 지원
 - 2. 기본, 격자 및 격자 그래픽
 - 3. 표 형식의 데이터
 - 4. 게이지 및 가치 상자
 - 5. 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>
 - **파일 - 새파일 - Rmarkdown - Template - Flexdashboard!**

개요

- 이 페이지에는 자신의 대시 보드의 시작점으로 사용할 수있는 다양한 샘플 레이아웃이 포함되어 있습니다.
- 레이아웃을 만들 때 차트가 웹 페이지를 수직으로 채우고 싶을 때 (브라우저가 바뀌면 높이가 변경됨), 혹은 차트가 원래 높이를 유지하게 할 것인지 결정할 필요가 있습니다. (모든 차트를 표시)
- 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.

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.

Chart Stack (Fill)

- 이 레이아웃은 두 차트의 간단한 스택입니다.
- 하나의 차트 또는 다른 차트는 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.

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

Chart 1

Chart 2

Chart Stack (Scrolling)

- 이 레이아웃은 세 개의 차트로 구성된 간단한 스택입니다.
- 모든 차트를 표시 할 수있는 충분한 공간을 제공하기 위해 스크롤 레이아웃이 사용됩니다.
(`vertical_layout: scroll`)
- 이 레이아웃을 스크롤 할 수 있기 때문에 많은 수의 차트를 여러 페이지로 구성 할 수 있습니다.
- 많은 수의 차트의 경우에는 `Multiple Pages`를 참조하세요.

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

Focal Chart (Top)

- 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 **data-height** 속성을 지정하여 상대 크기를 설정합니다.

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 |
```

Chart 1

Chart 2

Chart 3

Focal Chart (Left)

- 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.

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 |
```



Chart Grid (2 by 2)

- 이 레이아웃은 2x2 그리드 차트입니다.
- 이 레이아웃에서는 기본 `vertical_scroll: fill`를 사용하지만 차트의 이상적인 표시 크기에 따라 페이지를 스크롤하는 것이 바람직 할 수 있습니다 (`vertical_layout: scroll`).
- `orientation: rows`는 차트 기준선이 수평으로 정렬되도록하는 데 사용됩니다.

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 |  ### Chart 2
16 |  ```{r}
17 |  ```
18 |
19 |  Row
20 |  -----
21 |
22 |  ### Chart 3
23 |  ```{r}
24 |  ```
25 |
26 |  ### Chart 4
27 |  ```{r}
28 |  ```
29 |
30 |  Row
31 |  -----
32 |
33 |
```

Chart 1

Chart 2

Chart 3

Chart 4

Tabset Column

- 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.

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

Tabset Row

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

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
```

Chart 1

Chart 2

Chart 3

Multiple Pages

- 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 `data-width` 및 `data-height` 특성이있는 여러 열과 행을 사용하는 방법도 설명합니다.

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 |
```

Storyboard

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

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).

Input Sidebar

- 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

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

Input Sidebar (Global)

- 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.

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
```

Mobile Specific

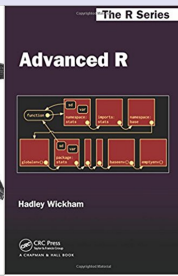
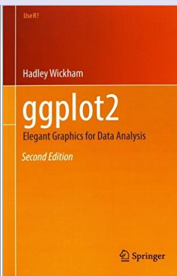
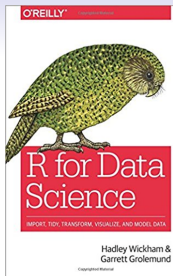
- 소형 모바일 화면에 표시 할 대시 보드를 사용자 정의하려면 선택한 구성 요소를 완전히 제외하거나 모바일 전용 변형을 만들 수 있습니다. 구성 요소를 제외하려면 `{.no-mobile}` 클래스 속성을 적용하십시오.
- 모바일 전용 렌더링을 사용하려면 동일한 제목의 두 개의 구성 요소를 만들고 그 중 하나에 `{.mobile}` 특성을 적용하십시오.
- 예를 들어, 다음 대시 보드에는 모바일 및 데스크톱 레이아웃에 포함 된 “Chart 1”, 모바일 레이아웃에서 제외 된 “Chart 2” 및 모바일 용 사용자 지정 변형이 있는 “Chart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

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 |
```



```
"Express Yourself"
```

```
## [1] "Express Yourself"
```