Flexdashboard

The goal of **flexdashboard** is to make it easy to create interactive dashboards for R, using R Markdown.

• Use R Markdown to publish a group of related data visualizations as a dashboard.
• Support for a wide variety of components including htmlwidgets; base, lattice, and grid graphics; tabular data; gauges and value boxes; and text annotations.

• Flexible and easy to specify row and column-based layouts. Components are intelligently re-sized to fill the browser and adapted for display on mobile devices.

• Storyboard layouts for presenting sequences of visualizations and related commentary.

• Optionally use Shiny to drive visualizations dynamically.

• Optionally use bslib to easily customize main colors, fonts, and more.

Learn more about **flexdashboard**: https://pkgs.rstudio.com/flexdashboard

Installation

Install the **flexdashboard** package from CRAN as follows:

```
install.packages("flexdashboard")
```

If you want to use the development version of the **flexdashboard** package, you can install the package from GitHub via the **remotes** package:

```
remotes::install_github('rstudio/flexdashboard')
```

Dashboard basics

Components

You can use flexdashboard to publish groups of related data visualizations as a dashboard. A flexdashboard can either be static (a standard web page) or dynamic (a Shiny interactive document). A wide variety of components can be included in flexdashboard layouts, including:

1. Interactive JavaScript data visualizations based on htmlwidgets.
2. R graphical output including base, lattice, and grid graphics.
3. Tabular data (with optional sorting, filtering, and paging).
4. Value boxes for highlighting important summary data.
5. Gauges for displaying values on a meter within a specified range.
6. Text annotations of various kinds.

SINGLE COLUMN (FILL)

Dashboards are divided into columns and rows, with output components delineated
using level 3 markdown headers (###).

By default, dashboards are laid out within a single column, with charts stacked vertically
within a column and sized to fill available browser height. For example, this layout defines a single column with two charts that fills available browser space:

````
---
title: "Single Column (Fill)"
output:
  flexdashboard::flex_dashboard:
    vertical_layout: fill
---

### Chart 1

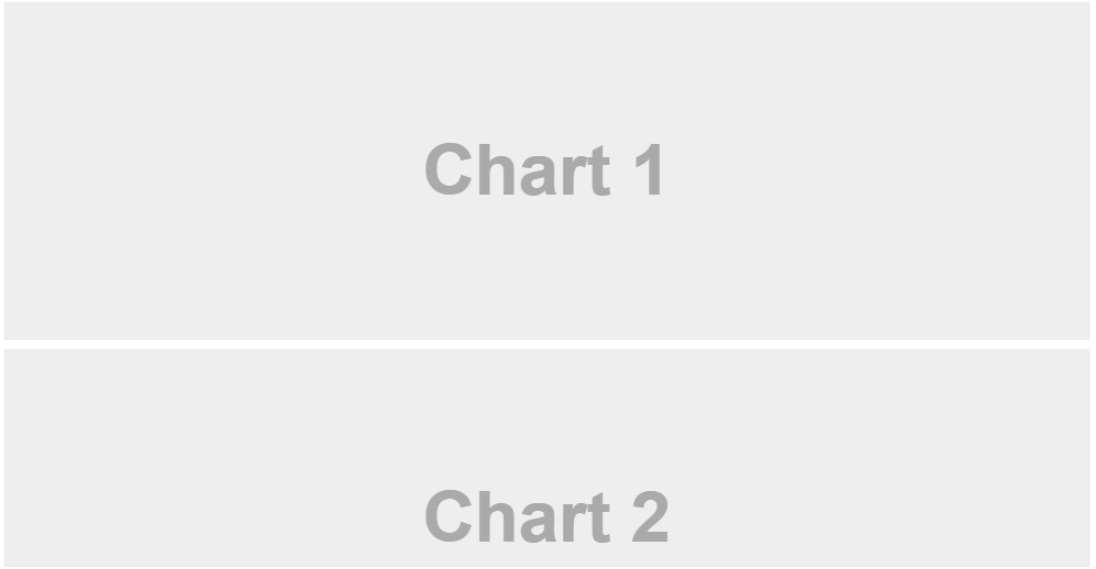
```{r}

```

### Chart 2

```{r}

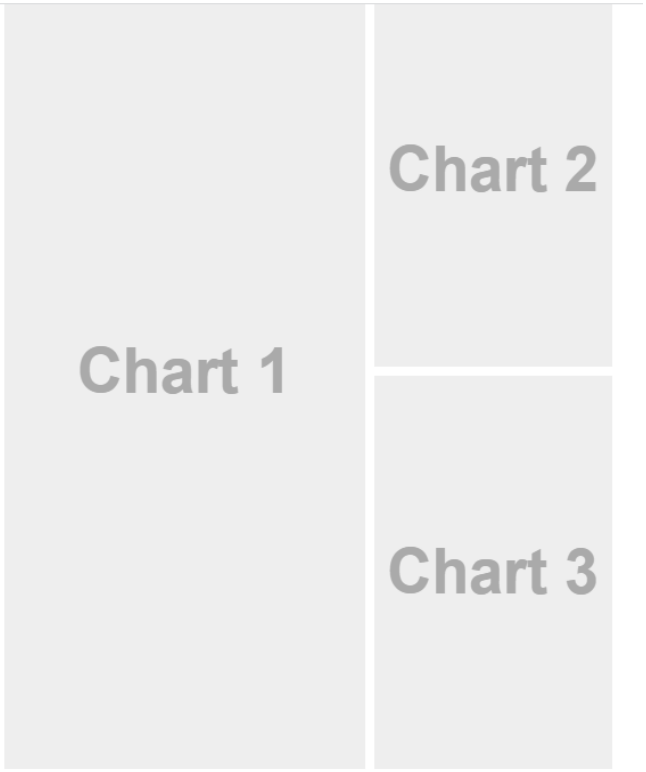
```
````



MULTIPLE COLUMNS

To lay out charts using multiple columns you introduce a level 2 markdown header (----------) for each column.
For example, this dashboard displays 3 charts split across two columns:

```
2  title: "Multiple Columns"
3  output: flexdashboard::flex_da
4  ---
5
6  Column {data-width=600}
7  -----------------------------
8
9  ### Chart 1
10
11 ```{r}
12
13 ```
14
15 Column {data-width=400}
16 -----------------------------
17
18 ### Chart 2
19
20 ```{r}
21
22 ```
23
24 ### Chart 3
25
26 ```{r}
27
28 ```
29
```



ROW ORIENTATION

You can also choose to orient dashboards row-wise rather than column-wise by specifying the orientation: rows option. For example, this layout defines two rows, the first of which has a single chart and the second of which has two charts:

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

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

### Chart 1

```{r}

```

Row

Chart 2

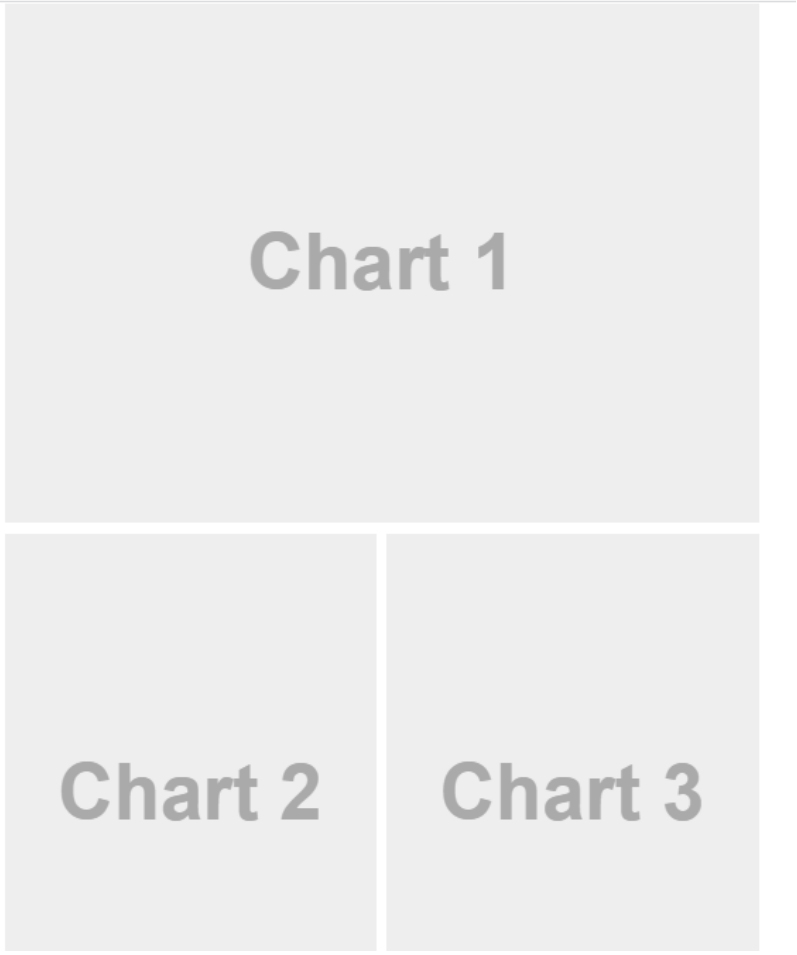
```{r}

```

Chart 3

```{r}

```
```



THEMES

A variety of themes are available to modify the base appearance of flexdashboard. Available themes include:

• Cosmo (The “cosmo” theme is used when “default” is specified. These themes are all based on the themes available in the R Markdown package.)

- bootstrap
- cerulean
- journal
- flatly
- readable
- spacelab
- united
- lumen
- paper
- sandstone
- simplex
- Yeti