Exploratory Data Analysis with R

Interactive Data Visualization with Flexdashboard

Xuemao Zhang
East Stroudsburg University

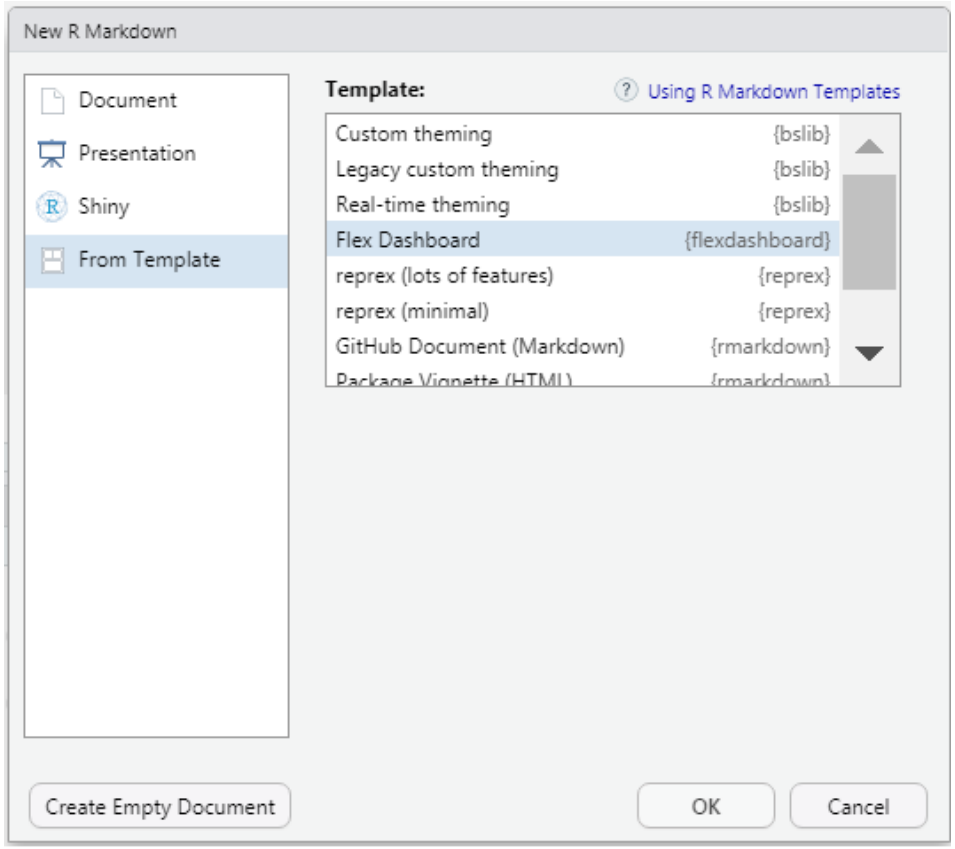
November 14, 2022

Overview

- A `flexdashboard` https://rmarkdown.rstudio.com/flexdashboard/ allows you to create an **interactive interface** to a data set, that allows communicating a lot of information visually and numerically, and the flexibility for the user to explore their own choice of aspects of the data.
- 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.

Basic dashboard

- To get started, make sure you have the `flexdashboard` package installed on your computer.
- Then create a "New R Markdown" document, "From Template", "Flex Dashboard".



Basic dashboard

- Check that the document compiles, by clicking `Knit`
- Modify the title and author

```
---
title: "An example of dashboard"
author: "by Xuemao Zhang"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
---
```

Flex Dashboard Components

- This creates a box or a pane for a plot or results

```
### Chart A
```

- This sets up columns

```
Column {data-width=650}
-------------------------------------
```

Flex Dashboard Components

- Let's fill in the three charts: A, B and C.

```
library(gapminder);
library(dplyr);
library(plotly);
p1 <- gapminder %>% filter(continent == "Asia")%>%
  filter(country != "Kuwait")%>% #outlier
  ggplot() +
   geom_point(aes(x=gdpPercap, y=lifeExp,
        size = pop, color =country, frame=year));
ggplotly(p1);
```

Flex Dashboard Components

```
p2 <- gapminder %>% filter(continent == "Europe") %>%
  ggplot() +
   geom_point(aes(x=gdpPercap, y=lifeExp,
        size = pop, color =country, frame=year));
ggplotly(p2);
```

```
p3 <- gapminder %>% filter(continent == "Americas") %>%
  ggplot() +
   geom_point(aes(x=gdpPercap, y=lifeExp,
        size = pop, color =country, frame=year));
ggplotly(p3);
```

Flex Dashboard Lab (to be continued)

- Make each column equal width
- Change the tab titles to reflect what continent is displayed

Flex Dashboard Lab

Adding pages and tabs

- A new page (tab) can be added using

```
Page 1
=====================================
```

- For details, see Multiple Pages: https://pkgs.rstudio.com/flexdashboard/articles/layouts.html

Flex Dashboard Lab

- Add a second page to your flexdashboard that shows gapminder data on the two remaining continents: `Africa` and `Oceania`
- Add a third page to your flexdashboard with
  - Gapminder data animation. You can copy the gganimate code from `Lecture20_animation.html`. Use `knitr::include_graphics("your animated graph")` to render a gganimate graph to the dashboard.
  - A data table showing mean `lifeExp`, `pop` and `gdpPercap` for each continent by year. Use `knitr::kable()` add the table

Flex Dashboard Lab

License