Introduction to R + Flexdashboard

Aaron Geller

Northwestern | INFORMATION TECHNOLOGY

What is Flexdashboard and why should I use it?

flexdashboard is an open-source R library that uses R Markdown to simplify the creation of dashboards.

Why use flexdashboard?

- easily generate row- and column-based layouts of figures, tables, widgets, etc. from the large ecosystem of data visualization libraries in R.
- many available themes (from Bootwatch via bslib) and options for custom CSS
- works well with Shiny (for user interactions)

A note on flexdashboard + Shiny

- Shiny is R's "go-to" tool for creating interactive data visualizations, but ...
  - it can be challenging to get started with
  - creating complicated (and good looking) dashboard-like layouts is not always easy
- flexdashboard actually simplifies the code needed for Shiny interactivity

Basic Code Structure

app.Rmd

```
---
# yaml header
---
```{r global}
# R global code chunk without visualization
```
Page 1 name
=====================================

Row

Figure 1 title
```{r}
# R code chunk with visualization
```
```

Basic Code Structure

app.Rmd

```
---
# yaml header
---
```{r global}
# R global code chunk without visualization
```
Page 1 name
=====================================

Row

Figure 1 title
```{r}
# R code chunk with visualization
```
```

- Include document-wide settings

```yaml
title: "Flexdashboard mtcars example"
runtime: shiny
output:
  flexdashboard::flex_dashboard :
    orientation: rows
    vertical_layout: fill
    source_code: embed
    theme:
      version: 4
      bootswatch: darkly
```

Basic Code Structure

app.Rmd

```
---
# yaml header
---
```{r global}
# R global code chunk without visualization
```
Page 1 name
=====================================

Row

Figure 1 title
```{r}
# R code chunk with visualization
```
```

- Include document-wide settings
- Add any code that will be used throughout the document (e.g., libraries, functions to produce plots/tables/etc.)

Basic Code Structure

app.Rmd

```
---
# yaml header
---
```{r global}
# R global code chunk without visualization
```
Page 1 name
=====================================

Row

Figure 1 title
```{r}
# R code chunk with visualization
```
```

- Include document-wide settings
- Add any code that will be used throughout the document (e.g., libraries, functions to produce plots/tables/etc.)
- layout tag for a tab/page in your app (replace “Page 1 name”); can have multiple tabs/pages

Basic Code Structure

app.Rmd

```
---
# yaml header
---
```{r global}
# R global code chunk without visualization
```
Page 1 name
=====================================

Row

Figure 1 title
```{r}
# R code chunk with visualization
```
```

- Include document-wide settings
- Add any code that will be used throughout the document (e.g., libraries, functions to produce plots/tables/etc.)
- layout tag for a tab/page in your app (replace “Page 1 name”); can have multiple tabs/pages
- layout tag for a row within this current page

Basic Code Structure

app.Rmd

```
---
# yaml header
---
```{r global}
# R global code chunk without visualization
```
Page 1 name
=====================================

Row

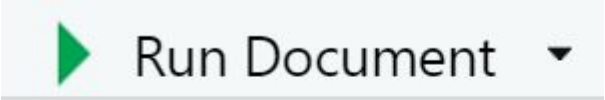
Figure 1 title
```{r}
# R code chunk with visualization
```
```

- Include document-wide settings
- Add any code that will be used throughout the document (e.g., libraries, functions to produce plots/tables/etc.)
- layout tag for a tab/page in your app (replace “Page 1 name”); can have multiple tabs/pages
- layout tag for a row within this current page
- Title and code to produce a visual (replace “Figure 1 title”)

How to compile the app

Use R Studio

- Launch R Studio and then open your application file (`"app.Rmd"`)
- Click the Run Document button.



- This should automatically open a window showing your local version of the app.

Examples

1. A few “gauge” plots to demonstrate how to work with layouts
2. Add scatter plots
3. Add a second tab/page with a table
4. Add a sidebar with a description of the content
5. Include Shiny elements to enable user control over plots

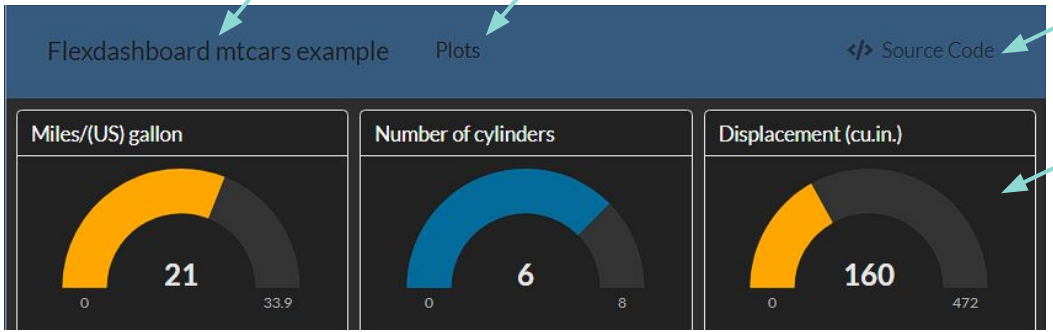
Example 1

A few “gauge” plots to demonstrate how to work with layouts

Row layout

app title

tab/page name

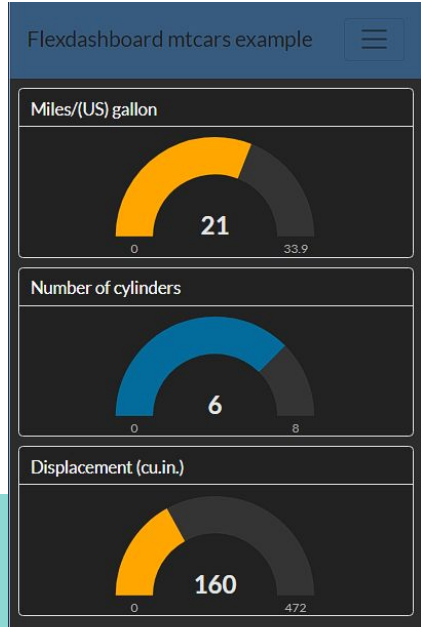


click here to view code

gauge plots for the first car in the dataset

try resizing your browser window

Column layout



- **Exercise 1.1:** change the layout from rows to columns
- **Exercise 1.2:** add more gauge plots in a 3x3 grid

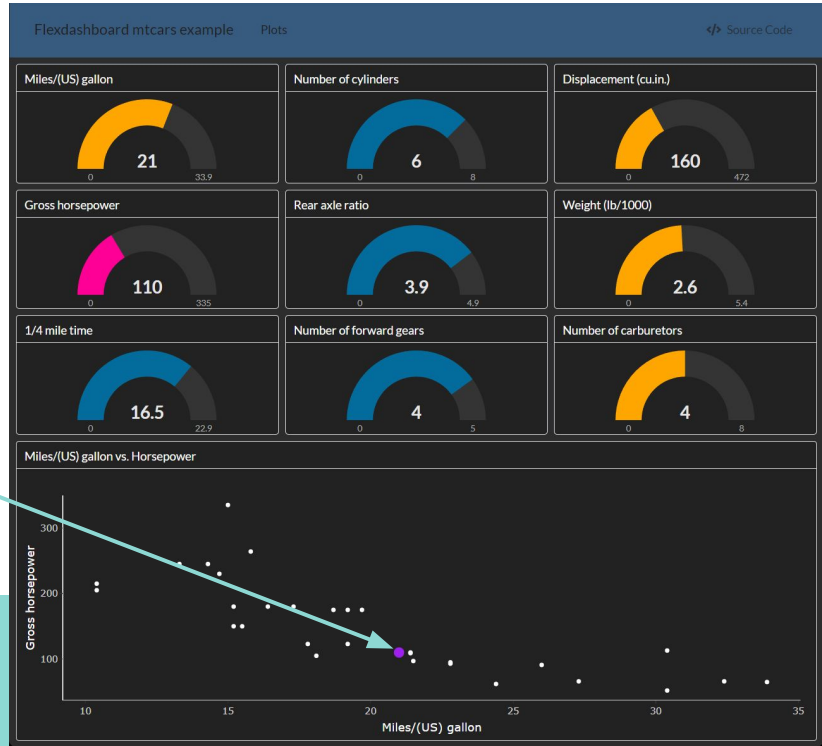
Example 2

Add scatter plot(s)

- I created a reusable function to generate the scatter plots
- `plotly` is a really nice library to enable interactivity with ggplot figures

try hovering over any point for a tooltip

highlights car shown in gauge plots



- **Exercise 2.1:** add a second scatter plot next to the current one in the same row (plotting different attributes)

Example 3

Add a second page with a table

- I'm using the `kableExtra` library for the table
  - it is scrollable, and keeps the header line at the top

new tab/page name

Flexdashboard mtcars example Plots Table Source Code

Table

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4

highlights car shown in gauge plots

- **Exercise 3.1:** test out adding another tab that contains some other data visualization (e.g., table or figure)

Example 4

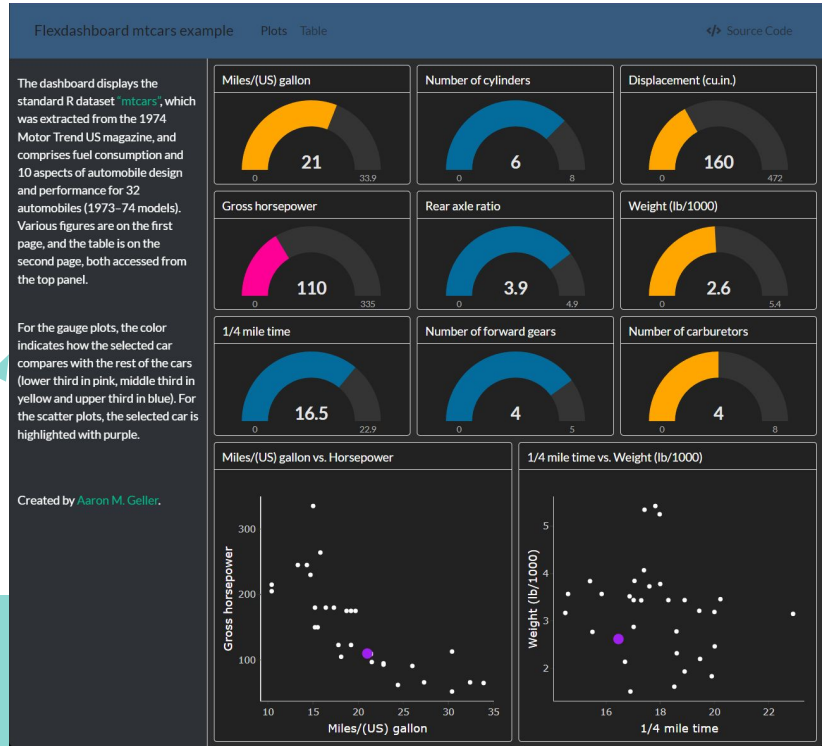
Add a sidebar

- Sidebars are created with the layout tag:

```
Sidebar {.sidebar}
=====================================
```

- Text can be formatted using standard html and/or markdown

the sidebar is visible on all tabs



- **Exercise 4.1:** experiment with changing the text in the sidebar

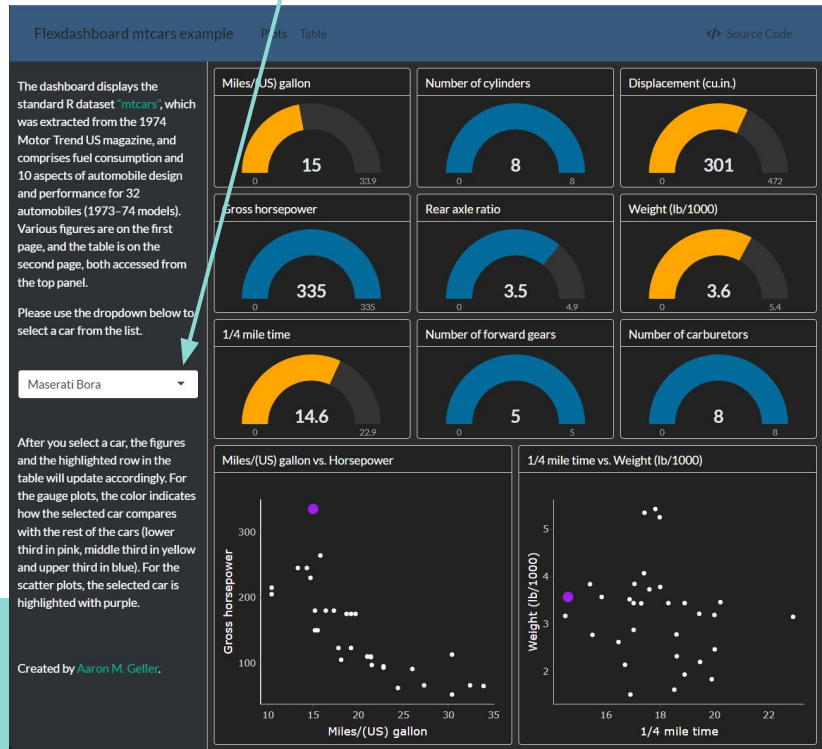
Example 5

Include Shiny element(s)

- I included a dropdown menu using `selectInput` from `Shiny`
- As with `Shiny` apps, the result from the dropdown is stored in the `input` object which I use to define the visualizations
- I do NOT need the usual `Shiny ui` or `server` functions (or to define reactive components explicitly)!!

- **Exercise 5.1:** use the dropdown to change the car and see the updated figures and table

dropdown menu shows available cars and is linked to plots and table



Hosting on shinyapps.io

First, sign up for an account on shinyapps.io

From R Studio

- Launch R Studio and then open your application file
- Click the publish button. 
- This will open a GUI window to select the file(s) to publish and other options, and will create the URL