

Introduction to Dashboards

bmRn CSM: Building dashboards with R Markdown

Martin Frigaard

2020-12-13

flexdashboard = Dashboards using R Markdown (and Shiny)



Load the packages



```
install.packages(c("tidyverse", "inspectdf",  
                  "flexdashboard", "reactable"))  
  
library(tidyverse)  
library(inspectdf)  
library(flexdashboard)  
library(reactable)
```



Outline (1)

Recap `rmarkdown`

What belongs in a dashboard?

Layouts

- *Sidebars, Columns, and Rows*
- *Multiple Pages, Tabs*

Themes

- *Bootstrap themes*



Outline (2)

inspectdf package

- *graphs, syntax*

reactable package

- *table displays*

Examples with shiny

- *shiny reactivity*



Materials

Slides

<https://mjfrigaard.github.io/intro-to-dashboards/Index.html>

Exercises

RStudio Project

<https://rstudio.cloud/project/2000287>

rmarkdown = **YAML** + **Markdown**
+ **R** (or other languages)



What is RMarkdown?



Three technologies:

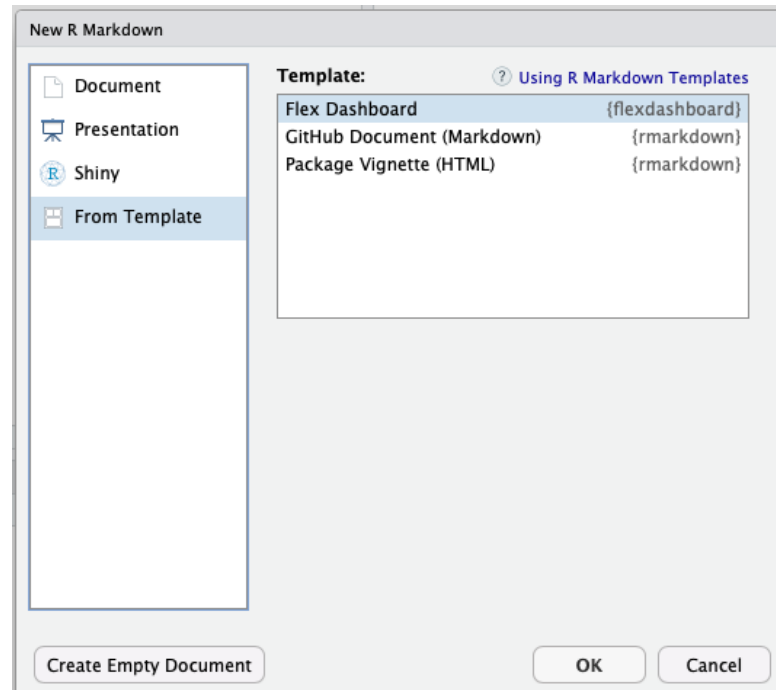
- 1) Markdown is a plain text markup language for capturing *human-readable* prose
- 2) Data manipulation/graphing/statistical language engines for computing *machine-readable* code
- 3) Multiple *output options* for creating PDFs, Word docs, PowerPoints, HTML, etc.

Your Turn 1



Open a new R Markdown file

*file > New File > R Markdown > From Template > **flexdashboard***



Your Turn 2



Add title and save R Markdown file

The screenshot shows the RStudio interface. The top pane displays an R Markdown file with the following content:

```
1 ---
2 title: "My Dashboard"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7 ---
8
9 ```{r setup, include=FALSE}
10 library(flexdashboard)
11 ```
12
13 Column {data-width=650}
14 -----
15
```

An orange box highlights the title "My Dashboard" on line 2, with an arrow pointing to the text "Add a title".

The bottom pane shows a "Save File - Untitled1" dialog box. The "File name:" field contains "my-dashboard.Rmd", which is highlighted by a red box and an arrow pointing to the text "Save your file!". The dialog also shows a file explorer view with the following files:

File Name	Size	Modified
..		
.Rhistory	0 B	Dec 6, 2020, 11:27 PM
install.R	175 B	Dec 6, 2020, 11:28 PM
project.Rproj	205 B	Dec 7, 2020, 12:08 AM

The dialog box has buttons for "New Folder", "Save", and "Cancel".

Your Turn 3



knit!

A screenshot of the RStudio interface demonstrating the R Markdown workflow. The left pane shows the source code for a dashboard. The right pane shows the rendered output, a dashboard titled "My Dashboard" with three charts. Colored arrows connect the code in the source pane to the corresponding elements in the rendered dashboard: an orange arrow from the title, a purple arrow from the first column/chart, and a red arrow from the second column/chart. The source code includes a title, output type, flexdashboard options, R setup code, and three chart sections with R code placeholders. The rendered dashboard shows the title, three columns, and three charts labeled Chart A, Chart B, and Chart C.

```
1 ---
2 title: "My Dashboard"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7 ---
8
9 ```{r setup, include=FALSE}
10 library(flexdashboard)
11 ```
12
13 Column {data-width=650}
14 -----
15
16 ## Chart A
17
18 ```{r}
19
20 ```
21
22 Column {data-width=350}
23 -----
24
25 ## Chart B
26
27 ```{r}
28
29 ```
30
31 ## Chart C
32
33 ```{r}
34
35 ```
36
37
```

What belongs in a dashboard?

Dashboards are particularly common in **business-style reports**. They can be used to **highlight brief and key summaries of a report**. The layout of a dashboard is often grid-based, with components arranged in boxes of various sizes.



Dashboard Anatomy



The YAML header setting creates the dashboard:

```
output:  
  flexdashboard::flex_dashboard:
```

The layout is determined by the **orientation** and **vertical_layout** options.

```
orientation: columns  
vertical_layout: fill
```

Column Widths

Column Widths must add up to **1000**

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

```
### Chart A
```

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

```
Column {data-width=350}
```

```
### Chart B
```

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

```
### Chart C
```

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

Sidebars



Include a sidebar with `{.sidebar data-width=200}`

```
Inputs {.sidebar data-width=200}
```

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

Adjust the column widths (set both to `{data-width=400}`)

```
Column {data-width=400}
```

```
### Chart A
```

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

```
Column {data-width=400}
```

```
### Chart B
```

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

```
### Chart C
```

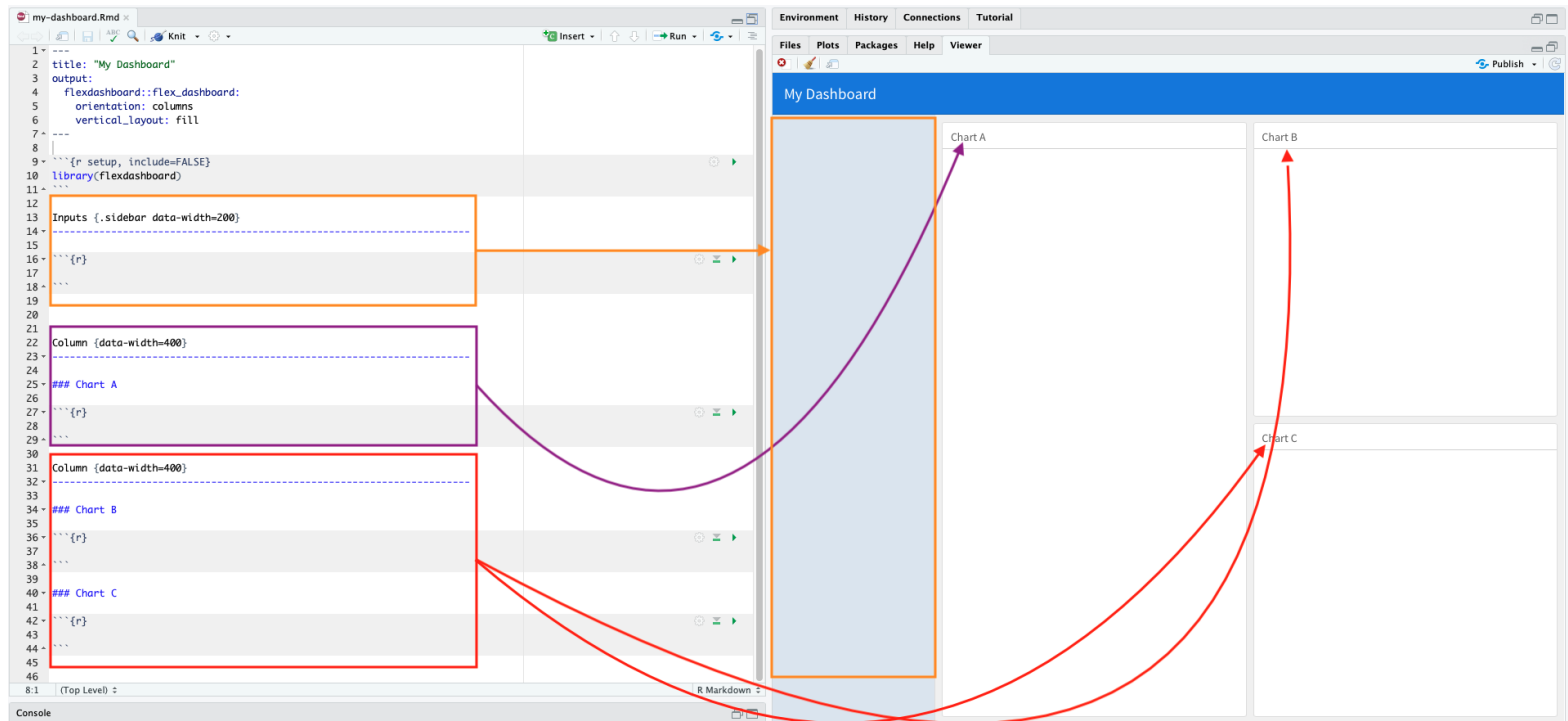
```
```${r}```  
```\n`
```

Knit!

Sidebars



Sidebars are typically used for data inputs and user-interface controls





Row Layout

We can also orient by rows

Change the `orientation` of the dashboard

```
output:
  flexdashboard::flex_dashboard:
    orientation: columns
```

Re-knit!

Rows Layout



The screenshot displays the RStudio interface with a flexdashboard R Markdown file named "my-dashboard.Rmd" open. The code is divided into sections, each highlighted with a colored box and an arrow pointing to the corresponding part of the rendered dashboard.

Code Sections:

- Output:** A blue box highlights the `output:` section with `flexdashboard::flex_dashboard:`, `orientation: rows`, and `vertical_layout: fill`. Arrows point from this box to the top of the dashboard and the "Row 1" and "Row 2" labels.
- Inputs:** A purple box highlights the `Inputs` section with `{.sidebar data-width=200}` and a `{r}` code chunk. An arrow points from this box to the sidebar area of the dashboard.
- Chart A:** A red box highlights the `Column` section for "Chart A" with `{data-width=400}` and a `{r}` code chunk. An arrow points from this box to the "Chart A" area of the dashboard.
- Chart B:** An orange box highlights the `Column` section for "Chart B" with `{data-width=400}` and a `{r}` code chunk. An arrow points from this box to the "Chart B" area of the dashboard.
- Chart C:** An orange box highlights the `Column` section for "Chart C" with `{data-width=400}` and a `{r}` code chunk. An arrow points from this box to the "Chart C" area of the dashboard.

Rendered Dashboard:

- The dashboard has a blue header titled "My Dashboard".
- It is organized into two rows, labeled "Row 1" and "Row 2" in green text.
- A sidebar on the left contains the "Inputs" section.
- The main content area contains three charts: "Chart A" (top), "Chart B" (bottom left), and "Chart C" (bottom right).

Scrolling



Change the [YAML](#) header back to `orientation: columns` and `vertical_layout: scroll`

```
orientation: columns  
vertical_layout: scroll
```

Re-knit!

Scrolling



Now we can scroll past the end of the column.

The screenshot shows the RStudio interface. On the left, the R Markdown editor displays code for a dashboard titled "My Dashboard". The code includes a flexdashboard header with orientation set to "columns" and vertical layout set to "scroll". It defines three columns: a sidebar with width 200, and two main content columns with width 400. The first main column contains "Chart A", the second contains "Chart B", and the third contains "Chart C". On the right, the rendered dashboard is shown. It has a blue header bar with the title "My Dashboard". Below the header, there is a sidebar on the left and two main content areas on the right. The top content area is empty, and the bottom content area is labeled "Chart C". A red arrow originates from the "vertical layout: scroll" line in the code and points to the vertical scrollbar on the right side of the rendered dashboard, indicating that the content can be scrolled past the end of the column.

Tabsets



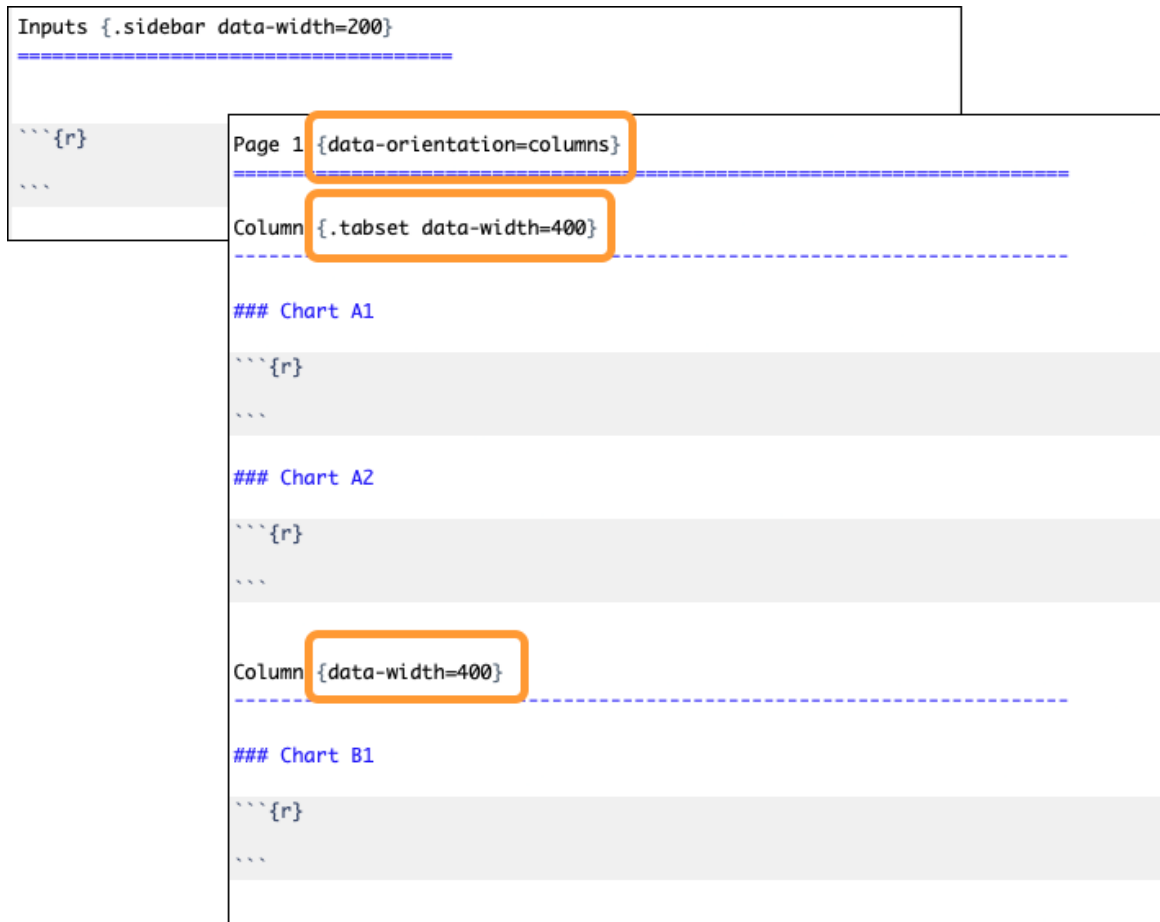
Add tabsets with `{.tabset}`

The screenshot shows the RStudio interface. On the left, the R Markdown source file 'my-dashboard.Rmd' is open. It contains a title 'My Dashboard', output settings for flexdashboard, and a column layout with three tabsets: 'Chart A1', 'Chart A2', and 'Chart A3' (all with data-width=400), followed by 'Chart B' (data-width=400). A red box highlights the first column of code, and a red arrow points from it to the corresponding tabset in the preview. On the right, the flexdashboard preview is shown. It has a blue header 'My Dashboard' and a sidebar. The main content area displays three tabs: 'Chart A1', 'Chart A2', and 'Chart A3'. 'Chart A1' is selected and shows a large empty white box. 'Chart B' is visible in a separate section below the tabs. The bottom of the RStudio window shows the console and R Markdown output.

Global Sidebar and Pages



For global settings, we use ===== instead of -----



Global Sidebar and Pages



data-orientation=columns

.tabset

The screenshot displays a flexdashboard interface. On the left is a global sidebar for "Page 1 (data-orientation=columns)". It contains three sections: "Chart A1", "Chart A2", and "Chart B1", each with a plot area and a legend. The main content area on the right is titled "My Dashboard" and features a tabset with three tabs: "Page 1", "Page 2", and "Page 3". The "Page 1" tab is active, showing a layout with three columns: "Chart A1", "Chart A2", and "Chart B1". The "Page 2" and "Page 3" tabs are visible but not active.

Global Sidebar and Pages



For global settings, we use ===== instead of -----

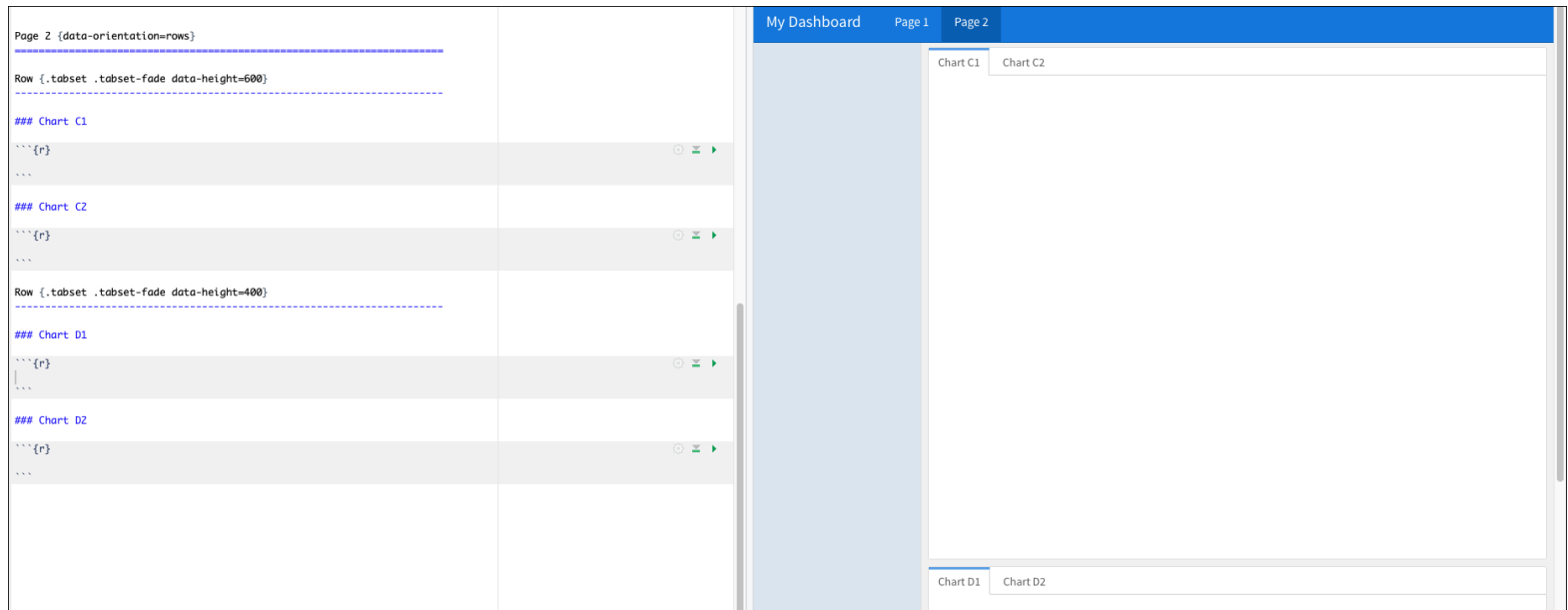
```
Page 2 {data-orientation=rows}  
-----  
Row {,tabset .tabset-fade data-height=600}  
-----  
  
### Chart C1  
  
```${r}```  
```\n\n### Chart C2  
  
```${r}```  
```\n\nRow {,tabset .tabset-fade data-height=400}  
-----  
  
### Chart D1  
  
```${r}```  
```\n\n### Chart D2  
  
```${r}```  
```\n\n
```


Global Sidebar and Pages



data-orientation=rows

.tabset-fade



Menus



data-navmenu=More

Page 3 {data-navmenu='More'}

Column

Chart E

```\${r}```

```\n\n

Page 4 {data-navmenu='More'}

Column

|

Chart F

```\${r}```

```\n\n

Menus



The screenshot displays a flexdashboard interface. On the left, a sidebar contains three chart sections: "## Chart D2", "## Chart E", and "## Chart F". Each section includes a header, a body with a "Page 3" or "Page 4" label, and a "Column" label. The main dashboard area on the right has a blue header with "My Dashboard" and navigation links "Page 1", "Page 2", and "More". A dropdown menu is open under "More", showing "Page 3" and "Page 4". The dashboard content area is mostly empty, with a "Chart C1" label at the top left. At the bottom, there are tabs for "Chart D1" and "Chart D2", with "Chart D2" being the active tab.

Themes



Change themes (just like `html_document()`!)

```
title: "My Dashboard"  
output:  
  flexdashboard::flex_dashboard:  
    theme: spacelab
```

See the website for more information

inspectdf = quickly examine
datasets



Previous Slides: Apple Mobility Data



<https://mjfrigaard.github.io/data-viz-as-comm/Index.html>

Import Data

```
AppleMobRaw <- readr::read_csv("https://bit.ly/36tTVpe")
```

Previous Slides: Apple Mobility Data



Don't Forget Wrangling Steps!

```
AppleMobRaw %>%  
  # transpose data  
  tidyr::pivot_longer(cols = -c(geo_type:country),  
    names_to = "date", values_to = "dir_request") %>%  
  # remove missing country data  
  dplyr::filter(!is.na(country) & !is.na(`sub-region`)) %>%  
  # clean names  
  janitor::clean_names() %>%  
  # date  
  mutate(date = lubridate::ymd(date)) %>%  
  # create trans_type  
  rename(trans_type = transportation_type) -> TidyApple
```

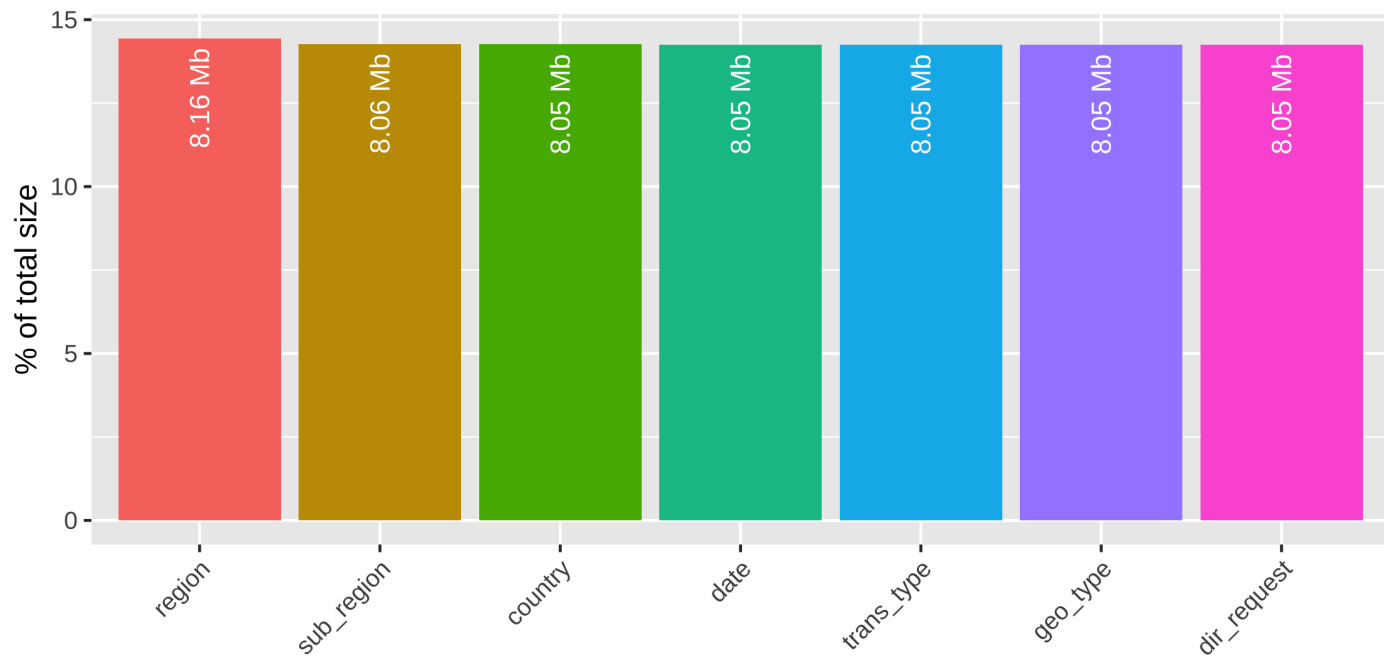
Dataset size in memory



```
TidyApple %>%  
  inspectdf::inspect_mem() %>%  
  inspectdf::show_plot(text_labels = TRUE)
```

Column sizes in df::.

df::. has 7 columns, 1055293 rows & total size of 56.48 Mb





Sidebar

Add the data to the `.sidebar`

Add the `import` and `wrangle` code to the sidebar in the dashboard.

```
# import ----
AppleMobRaw <- readr::read_csv("https://bit.ly/36tTVpe")
# wrangle ----
AppleMobRaw %>%
  # transpose data
  tidyr::pivot_longer(cols = -c(geo_type:country),
    names_to = "date", values_to = "dir_request") %>%
  # remove missing country data
  dplyr::filter(!is.na(country) & !is.na(`sub-region`)) %>%
  # clean names
  janitor::clean_names() %>%
  # date
  mutate(date = lubridate::ymd(date)) %>%
  # create trans_type
  rename(trans_type = transportation_type) -> TidyApple
```

Page 1, Column 1, Tab 1



Add the 'Memory Size' Graph

Add this code to [A1](#)

```
TidyApple %>%  
  inspectdf::inspect_mem() %>%  
  inspectdf::show_plot(text_labels = TRUE)
```

Knit--how does it look?

Page 1, Column 1, Tab 2



Add the Missing Data Graph

Add this code to [A2](#)

```
TidyApple %>%  
  inspectdf::inspect_na() %>%  
  inspectdf::show_plot(text_labels = TRUE)
```

Knit--how does it look?

Page 1, Column 2, Tab 1



Add the Categorical Data Graph

Add this code to [B1](#)

```
TidyApple %>%  
  select_if(is.character) %>%  
  inspectdf::inspect_cat() %>%  
  inspectdf::show_plot(text_labels = TRUE)
```

Knit--how does it look?

Page 1, Column 2, Tab 2



Add the **Data Imbalances** Graph

Add this code to [B2](#)

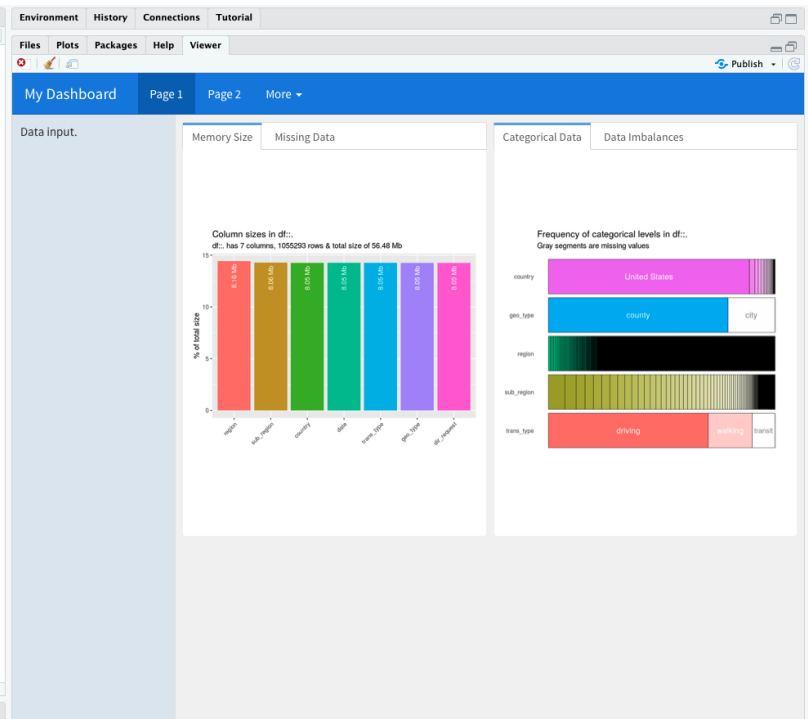
```
TidyApple %>%  
  inspectdf::inspect_imb() %>%  
  inspectdf::show_plot(text_labels = TRUE)
```

Knit--how does it look?

Page 1



```
install.R x solutions.Rmd x exercises.Rmd x
Insert Run
38
39
40 Page 1 {data-orientation=columns}
41 -----
42
43 Column {.tabset data-width=400}
44 -----
45
46 ## Memory Size
47
48 ```{r inspect_mem}
49 TidyApple %>%
50   inspectdf::inspect_mem() %>%
51   inspectdf::show_plot(text_labels = TRUE)
52 ```
53
54 ## Missing Data
55
56 ```{r inspect_na}
57 TidyApple %>%
58   inspectdf::inspect_na() %>%
59   inspectdf::show_plot(text_labels = TRUE)
60 ```
61
62
63 Column {.tabset data-width=400}
64 -----
65
66 ## Categorical Data
67
68 ```{r inspect_cat}
69 TidyApple %>%
70   select_if(is.character) %>%
71   inspectdf::inspect_cat() %>%
72   inspectdf::show_plot(text_labels = TRUE)
73 ```
74
75 ## Data Imbalances
76
77 ```{r inspect_imb}
78 TidyApple %>%
79   inspectdf::inspect_imb() %>%
80   inspectdf::show_plot(text_labels = TRUE)
81 ```
82
105.58 [x] Chunk 8: ggridges [x] R Markdown [x] Console
```



Page 2 (Rows)



Page 2 {data-orientation=rows}

=====



Page 2, Row 1, Tab 1

Create a `.tabset/.tabset-fade` Row

```
Row {.tabset .tabset-fade data-height=600}
```

Add Numeric Data Graph

```
TidyApple %>%  
  select_if(is.numeric) %>%  
  inspectdf::inspect_num() %>%  
  inspectdf::show_plot(text_labels = TRUE)
```

Knit--how does it look?

Page 2, Row 1, Tab 2



Add 'Distributions ggridges' Graph

```
library(ggribes)
lab_ridges <- labs(x = "Apple directions requests",
                  y = "Transportation Types",
                  title = "Direction Requests by Transportation Type",
                  subtitle = "source: https://covid19.apple.com/mobility")
```

```
TidyApple %>%
  ggplot() +
  geom_density_ridges(aes(x = dir_request,
                        y = trans_type,
                        fill = trans_type),
                    alpha = 1/5) +
  lab_ridges
```

Knit--how does it look?

Page 2, Row 2, Tab 1



Create Another `.tabset/.tabset-fade` Row

```
Row {.tabset .tabset-fade data-height=400}
```

In tab 1, add `TopUSCities` as `paged_table`

```
TopUSCities <- TidyApple %>%  
  filter(country == "United States" &  
         region %in% c("New York City", "Los Angeles",  
                       "Chicago", "Houston", "Phoenix"))  
rmarkdown::paged_table(TopUSCities)
```

Knit--how does it look?



Page 2, Row 2, Tab 2

In tab 2, add **MaxUSCitiesDriving** as **reactable**

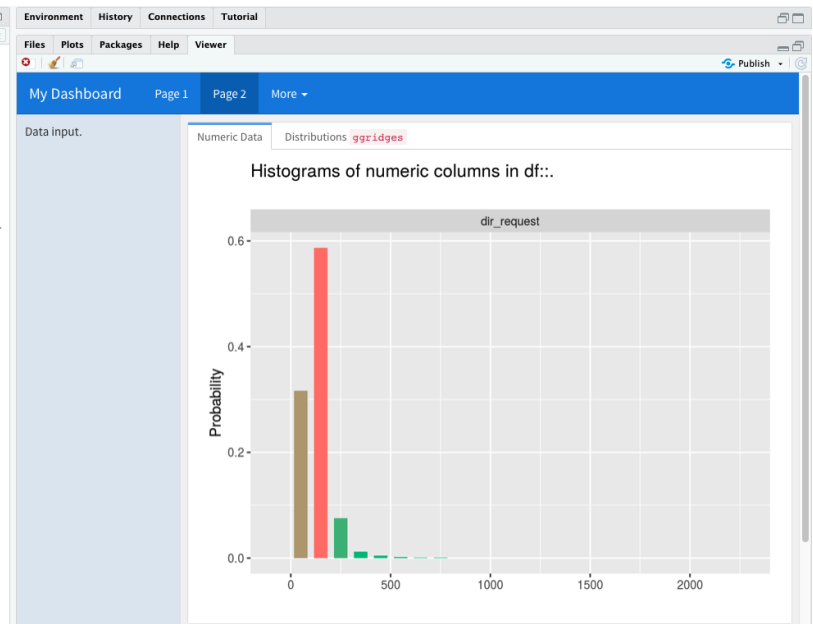
```
TopUSCities %>%  
  filter(trans_type == "driving") %>%  
  group_by(region) %>%  
  slice_max(dir_request) %>%  
  ungroup() -> MaxUSCitiesDriving  
reactable(MaxUSCitiesDriving,  
           resizable = TRUE, showPageSizeOptions = TRUE,  
           selection = "multiple", onClick = "select")
```

Knit--how does it look?

Page 2



```
83 Page 2 {data-orientation=rows}
84
85
86 Row {,tabset,tabset-fade data-height=600}
87
88
89
90 ## Numeric Data
91
92 ```{r}
93 TidyApple %>%
94   select_if(is.numeric) %>%
95   inspectdf::inspect_num() %>%
96   inspectdf::show_plot(text_labels = TRUE)
97
98
99 ## Distributions 'ggridges'
100
101 ```{r ggridges, message=FALSE, warning=FALSE}
102 library(ggridges)
103 lab_ridges <- labs(x = "Apple directions requests",
104                   y = "Transportation Types",
105                   title = "Direction Requests by Transportation Type",
106                   subtitle = "source: https://covid19.apple.com/mobility")
107
108
109 TidyApple %>%
110   ggplot() +
111   geom_density_ridges(aes(x = dir_request,
112                           y = trans_type,
113                           fill = trans_type,
114                           alpha = 1/5)) +
115   lab_ridges
116
117
118 Row {,tabset,tabset-fade data-height=400}
119
120
121 ## 'TopUSCities' as 'paged_table'
122
123 ```{r TopUSCities}
124 TopUSCities <- TidyApple %>%
```



Page 2



```
99 - ## Distributions 'ggridges'
100
101 - ```{r ggridges, message=FALSE, warning=FALSE}
102   library(ggridges)
103   lab_ridges <- labs(x = "Apple directions requests",
104                     y = "Transportation Types",
105                     title = "Direction Requests by Transportation Type",
106                     subtitle = "source: https://covid19.apple.com/mobility")
107
108
109   TidyApple %>%
110     ggplot() +
111     geom_density_ridges(aes(x = dir_request,
112                           y = trans_type,
113                           fill = trans_type,
114                           alpha = 1/5) +
115     lab_ridges
116 - ```
117
118   Row {.tabset .tabset-fade data-height=400}
119 -
120
121 - ## 'TopUSCities' as 'paged_table'
122
123 - ```{r TopUSCities}
124   TopUSCities <- TidyApple %>%
125     filter(country == "United States" &
126            region %in% c("New York City", "Los Angeles",
127                          "Chicago", "Houston", "Phoenix"))
128   rmarkdown::paged_table(TopUSCities)
129 - ```
130
131
132 - ## 'MaxUSCitiesDriving' as 'reactable'
133
134 - ```{r MaxUSCitiesDriving}
135   TopUSCities %>%
136     filter(trans_type == "driving") %>%
137     group_by(region) %>%
138     slice_max(dir_request) %>%
139     ungroup() -> MaxUSCitiesDriving
140   reactable(MaxUSCitiesDriving,
141             resizable = TRUE, showPageSizeOptions = TRUE,
142             selection = "multiple", onClick = "select")
143 - ```
144
60:4 Missing Data ↕ R Markdown ↕
```

Environment History Connections Tutorial

Files Plots Packages Help Viewer

My Dashboard Page 1 Page 2 More ▾

Data input.

TopUSCities as paged_table

MaxUSCitiesDriving as reactable

| <input checked="" type="checkbox"/> | geo_type | region | trans_type | sub_region | country | date | dir_reques |
|-------------------------------------|----------|---------------|------------|------------|---------------|------------|------------|
| <input checked="" type="checkbox"/> | city | Chicago | driving | Illinois | United States | 2020-07-17 | 166.1 |
| <input checked="" type="checkbox"/> | city | Houston | driving | Texas | United States | 2020-02-14 | 146. |
| <input checked="" type="checkbox"/> | city | Los Angeles | driving | California | United States | 2020-02-14 | 152.0 |
| <input checked="" type="checkbox"/> | city | New York City | driving | New York | United States | 2020-09-04 | 152.0 |
| <input checked="" type="checkbox"/> | city | Phoenix | driving | Arizona | United States | 2020-02-29 | 142.6 |



More Examples

Check out the package website and gallery

<https://rmarkdown.rstudio.com/flexdashboard/examples.html>