

# FlexDashboards for Clinical & Translational Research

David Miller M.D., Ph.D., and Sophia Shalhout Ph.D.

First Published on <https://www.themillerlab.io/post/flexdashboards> on 1/11/2020.  
Updated 06/05/2020

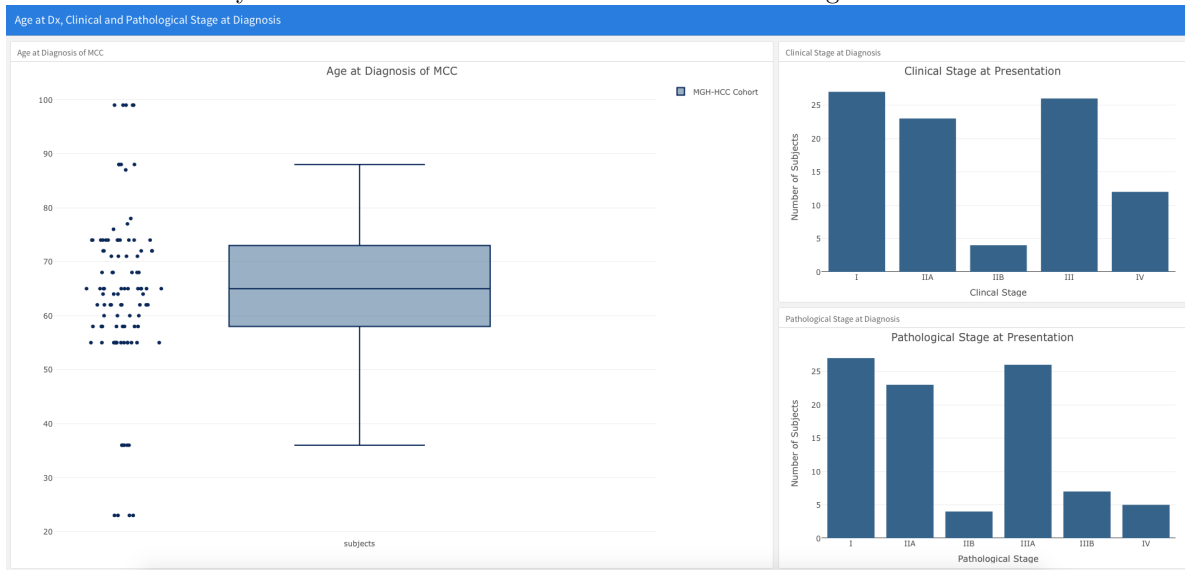
## Contents

Key Points . . . . .	1
Load Packages . . . . .	2
Format of Flexdashboard . . . . .	2
Load The Data . . . . .	3
View Dataset . . . . .	3
Data Visualizations . . . . .	5
BoxPlot of Age at Diagnosis . . . . .	5
Wrangle Data for Age at Diagnosis Box Plot . . . . .	5
View Dataset Age at Dx . . . . .	5
Plotly Graph of Age at Diagnosis . . . . .	7
Bar Chart of Stage of Diagnosis . . . . .	8
Bar Chart of Pathological Staging at Diagnosis . . . . .	9
Recapitulate the RMarkdown with the Flexdashboard Template . . . . .	10
Additional FlexDashboard Formatting to Enhance the User Experience (UX) . . . . .	12
Add Pages to A FlexDashboard . . . . .	12
Add Tabs to A FlexDashboard . . . . .	15
Take Home Points . . . . .	19
Session Info . . . . .	19

## Key Points

- This monograph provides a roadmap to create a **FlexDashboard** for Clinical and Translational Research
- The purpose of a **Dashboard** is to present data in a way that enhances its interpretation
- There are many excellent tutorials on using the package Flexdashboard(<https://cran.r-project.org/web/packages/flexdashboard/flexdashboard.pdf>) to create a dashboard; however, in this monograph we aim to highlight a few pearls that we have learned along the way that we hope you'll find useful

- What we will show you next is the code we used to create the following Flexdashboard



- Of note, the data here is completely fabricated, any relation to real subjects is completely coincidental
- Skill Level: Intermediate
  - Assumption made by this post is that readers will have basic familiarity with R

## Load Packages

```
library(flexdashboard)
library(RCurl)
library(REDCapR)
library(httr)
library(tidyverse)
library(knitr)
library(plotly)
library(readxl)
library(scales)
```

## Format of Flexdashboard

- The Flexdashboard package is built around RMarkdown
- To start, you can open a new RMarkdown by:
  - File -> New File -> RMarkdown
  - Then go to “From Template” on the left hand column
  - If you have installed Flexdashboard then Flexdashboard will be in the window that opens
  - Once you have selected Flexdashboard a new RMarkdown will open
  - What is specific here is the YAML

\* The content between the three dashes in beginning of the RMarkdown

- This is the YAML for the Flexdashboard

```
title: "Untitled"
output:
flexdashboard::flex_dashboard:
orientation: columns
vertical_layout: fill
```

- Then after the YAML, the Flexdashboard template is quite basic
- You will generate code for the the three charts that you want displayed
- The template calls them Chart A, Chart B, Chart C.
- Each Chart is preceded by three #s and is flanked by a long series of dashes
- You can then change the title from `Chart A` to the title you desire
- Then following the Title, you have an R chunk that you use to generate your charts

## Load The Data

- This is a fabricated dataset in Excel, but has many features that resemble an actual rare disease cohort dataset. The dataset can be accessed at `mcc_cohort_fake.xlsx`. Then just click on **View raw**.

```
dt <- read_excel("mcc_cohort_fake.xlsx")
```

## View Dataset

```
dt %>% kable
```

record_id	redcap_repeat_instrument	redcap_repeat_instance	redcap_data_access_group	...5	age_at_dx	ma
1	NA	NA	NA	NA	60	
2	NA	NA	NA	NA	65	
3	NA	NA	NA	NA	64	
4	NA	NA	NA	NA	71	
5	NA	NA	NA	NA	72	
6	NA	NA	NA	NA	74	
7	NA	NA	NA	NA	58	
8	NA	NA	NA	NA	23	
9	NA	NA	NA	NA	55	
10	NA	NA	NA	NA	99	
11	NA	NA	NA	NA	36	
12	NA	NA	NA	NA	62	
13	NA	NA	NA	NA	65	
14	NA	NA	NA	NA	68	
15	NA	NA	NA	NA	74	
16	NA	NA	NA	NA	58	
17	NA	NA	NA	NA	88	
18	NA	NA	NA	NA	55	

record_id	redcap_repeat_instrument	redcap_repeat_instance	redcap_data_access_group	...5	age_at_dx	ma
19	NA	NA	NA	NA	55	
20	NA	NA	NA	NA	74	
21	NA	NA	NA	NA	62	
22	NA	NA	NA	NA	65	
23	NA	NA	NA	NA	60	
24	NA	NA	NA	NA	65	
25	NA	NA	NA	NA	64	
26	NA	NA	NA	NA	71	
27	NA	NA	NA	NA	72	
28	NA	NA	NA	NA	74	
29	NA	NA	NA	NA	58	
30	NA	NA	NA	NA	23	
31	NA	NA	NA	NA	55	
32	NA	NA	NA	NA	99	
33	NA	NA	NA	NA	36	
34	NA	NA	NA	NA	62	
35	NA	NA	NA	NA	65	
36	NA	NA	NA	NA	68	
37	NA	NA	NA	NA	74	
38	NA	NA	NA	NA	58	
39	NA	NA	NA	NA	88	
40	NA	NA	NA	NA	55	
41	NA	NA	NA	NA	55	
42	NA	NA	NA	NA	74	
43	NA	NA	NA	NA	62	
44	NA	NA	NA	NA	65	
45	NA	NA	NA	NA	68	
46	NA	NA	NA	NA	55	
47	NA	NA	NA	NA	60	
48	NA	NA	NA	NA	65	
49	NA	NA	NA	NA	64	
50	NA	NA	NA	NA	71	
51	NA	NA	NA	NA	72	
52	NA	NA	NA	NA	74	
53	NA	NA	NA	NA	58	
54	NA	NA	NA	NA	23	
55	NA	NA	NA	NA	55	
56	NA	NA	NA	NA	99	
57	NA	NA	NA	NA	36	
58	NA	NA	NA	NA	62	
59	NA	NA	NA	NA	65	
60	NA	NA	NA	NA	60	
61	NA	NA	NA	NA	65	
62	NA	NA	NA	NA	64	
63	NA	NA	NA	NA	71	
64	NA	NA	NA	NA	72	
65	NA	NA	NA	NA	74	
66	NA	NA	NA	NA	58	
67	NA	NA	NA	NA	23	
68	NA	NA	NA	NA	55	
69	NA	NA	NA	NA	99	
70	NA	NA	NA	NA	74	

record_id	redcap_repeat_instrument	redcap_repeat_instance	redcap_data_access_group	...5	age_at_dx	ma
71	NA	NA	NA	NA	62	
72	NA	NA	NA	NA	65	
73	NA	NA	NA	NA	68	
74	NA	NA	NA	NA	74	
75	NA	NA	NA	NA	58	
76	NA	NA	NA	NA	88	
77	NA	NA	NA	NA	55	
78	NA	NA	NA	NA	55	
79	NA	NA	NA	NA	36	
80	NA	NA	NA	NA	62	
81	NA	NA	NA	NA	65	
82	NA	NA	NA	NA	68	
83	NA	NA	NA	NA	55	
84	NA	NA	NA	NA	99	
85	NA	NA	NA	NA	36	
86	NA	NA	NA	NA	62	
87	NA	NA	NA	NA	65	
88	NA	NA	NA	NA	68	
89	NA	NA	NA	NA	74	
90	NA	NA	NA	NA	58	
91	NA	NA	NA	NA	88	
92	NA	NA	NA	NA	55	
93	NA	NA	NA	NA	55	
94	NA	NA	NA	NA	74	
95	NA	NA	NA	NA	78	
96	NA	NA	NA	NA	72	
97	NA	NA	NA	NA	77	
98	NA	NA	NA	NA	76	
99	NA	NA	NA	NA	87	
100	NA	NA	NA	NA	62	

## Data Visualizations

### BoxPlot of Age at Diagnosis

#### Wrangle Data for Age at Diagnosis Box Plot

```
Age_at_Dx <- dt %>% select(record_id, age_at_dx) %>% drop_na(age_at_dx) # drop_na is a good function
# to eliminate rows that have missing values
Age_at_Dx$subjects <- "subjects" # add a column that unifies all the data (helpful for plotly)
```

#### View Dataset Age at Dx

```
Age_at_Dx %>% kable
```

record_id	age_at_dx	subjects
1	60	subjects
2	65	subjects
3	64	subjects

record_id	age_at_dx	subjects
4	71	subjects
5	72	subjects
6	74	subjects
7	58	subjects
8	23	subjects
9	55	subjects
10	99	subjects
11	36	subjects
12	62	subjects
13	65	subjects
14	68	subjects
15	74	subjects
16	58	subjects
17	88	subjects
18	55	subjects
19	55	subjects
20	74	subjects
21	62	subjects
22	65	subjects
23	60	subjects
24	65	subjects
25	64	subjects
26	71	subjects
27	72	subjects
28	74	subjects
29	58	subjects
30	23	subjects
31	55	subjects
32	99	subjects
33	36	subjects
34	62	subjects
35	65	subjects
36	68	subjects
37	74	subjects
38	58	subjects
39	88	subjects
40	55	subjects
41	55	subjects
42	74	subjects
43	62	subjects
44	65	subjects
45	68	subjects
46	55	subjects
47	60	subjects
48	65	subjects
49	64	subjects
50	71	subjects
51	72	subjects
52	74	subjects
53	58	subjects
54	23	subjects
55	55	subjects

record_id	age_at_dx	subjects
56	99	subjects
57	36	subjects
58	62	subjects
59	65	subjects
60	60	subjects
61	65	subjects
62	64	subjects
63	71	subjects
64	72	subjects
65	74	subjects
66	58	subjects
67	23	subjects
68	55	subjects
69	99	subjects
70	74	subjects
71	62	subjects
72	65	subjects
73	68	subjects
74	74	subjects
75	58	subjects
76	88	subjects
77	55	subjects
78	55	subjects
79	36	subjects
80	62	subjects
81	65	subjects
82	68	subjects
83	55	subjects
84	99	subjects
85	36	subjects
86	62	subjects
87	65	subjects
88	68	subjects
89	74	subjects
90	58	subjects
91	88	subjects
92	55	subjects
93	55	subjects
94	74	subjects
95	78	subjects
96	72	subjects
97	77	subjects
98	76	subjects
99	87	subjects
100	62	subjects

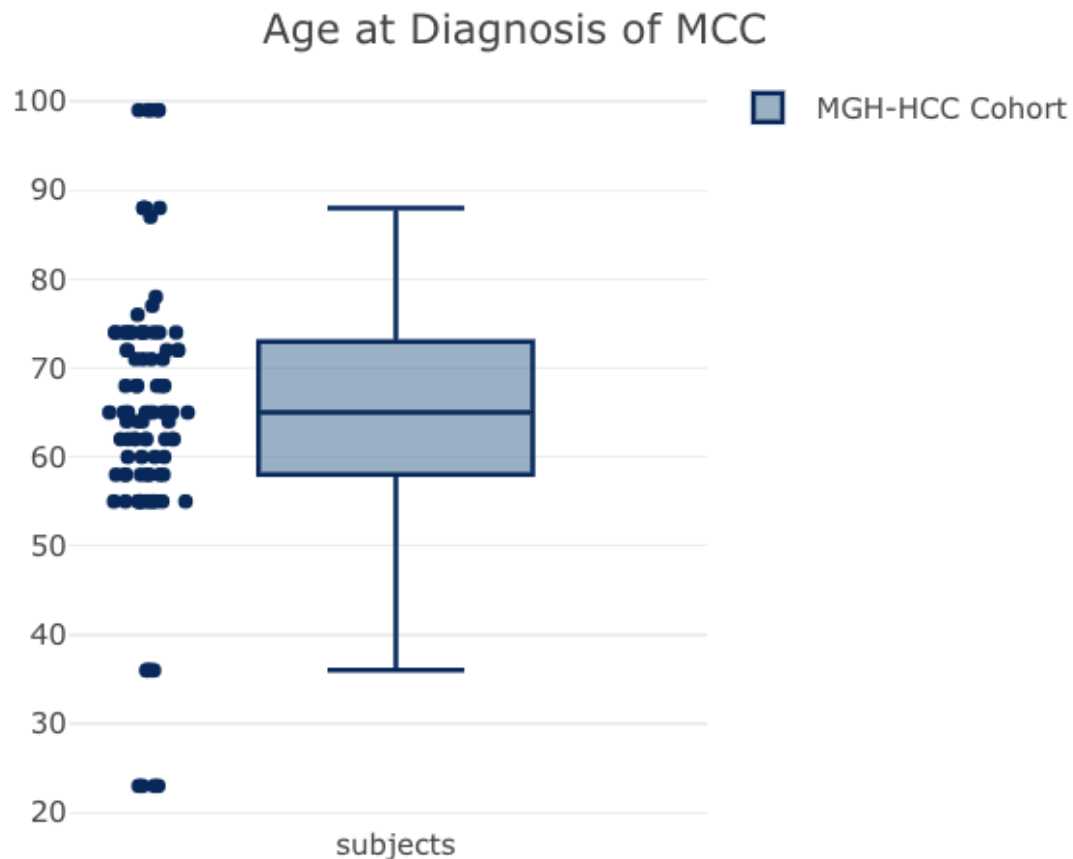
### Plotly Graph of Age at Diagnosis

```
plot_ly(data = Age_at_Dx, type = "box") %>%
  add_boxplot(x = Age_at_Dx$subjects, y = Age_at_Dx$age_at_dx,
             boxpoints = "all", jitter = 0.3, pointpos = -1.8,
```

```

marker = list(color = 'rgb(7,40,89)'),
line = list(color = 'rgb(7,40,89)'),
color = I("steelblue4"),
name = "MGH-HCC Cohort" %>%
layout(title = "Age at Diagnosis of MCC")

```



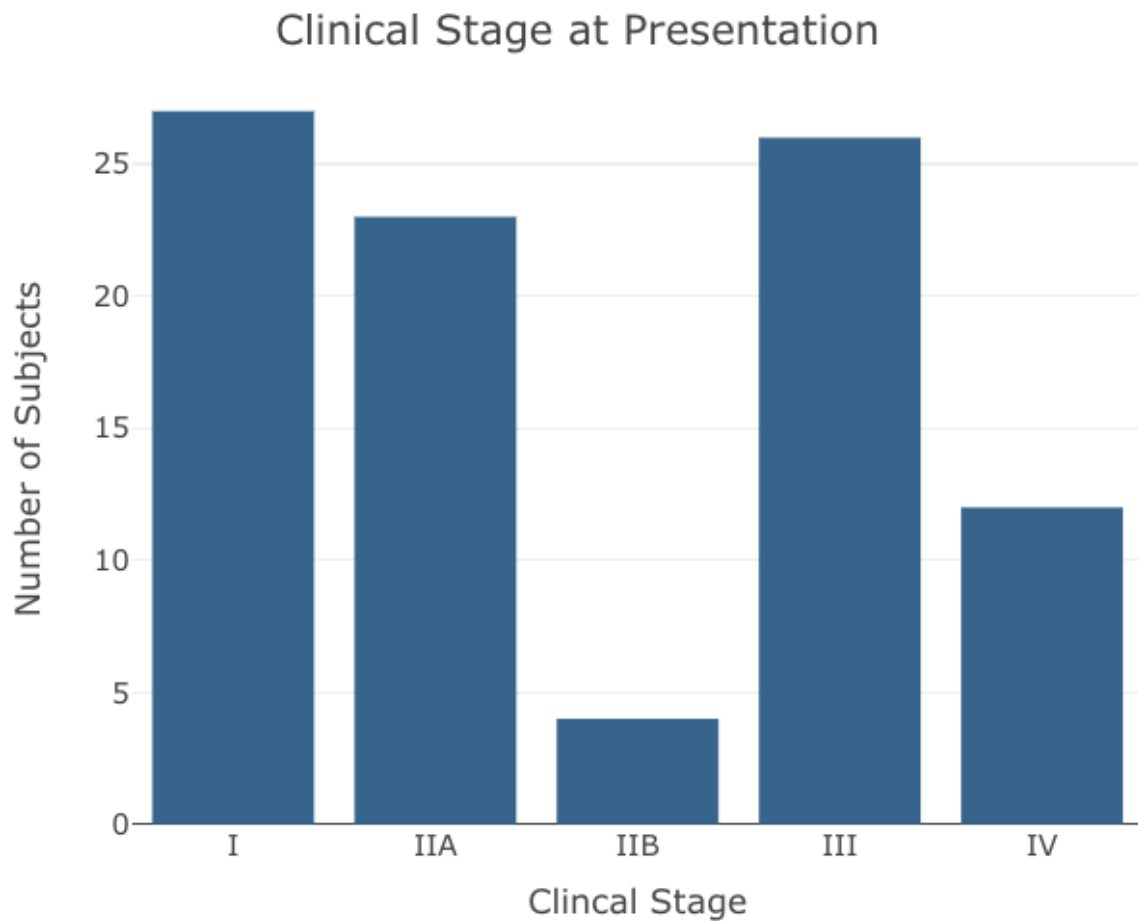
### Bar Chart of Stage of Diagnosis

```

cStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_clinstage) %>%
  filter(man_clinstage < 98)
cStageDF <- cStage %>% group_by(man_clinstage) %>% tally()
plot_ly(data = cStageDF) %>%
  add_bars(x = cStageDF$man_clinstage, y = cStageDF$n,
           color = I("steelblue4")) %>%
  layout(
    title = "Clinical Stage at Presentation",
    yaxis = list(title = "Number of Subjects"),
    xaxis = list(title = "Clinical Stage", ticktext = list("I", "IIA", "IIB", "III", "IV"),
                 tickvals = list(0, 1, 2, 3, 4))
  )

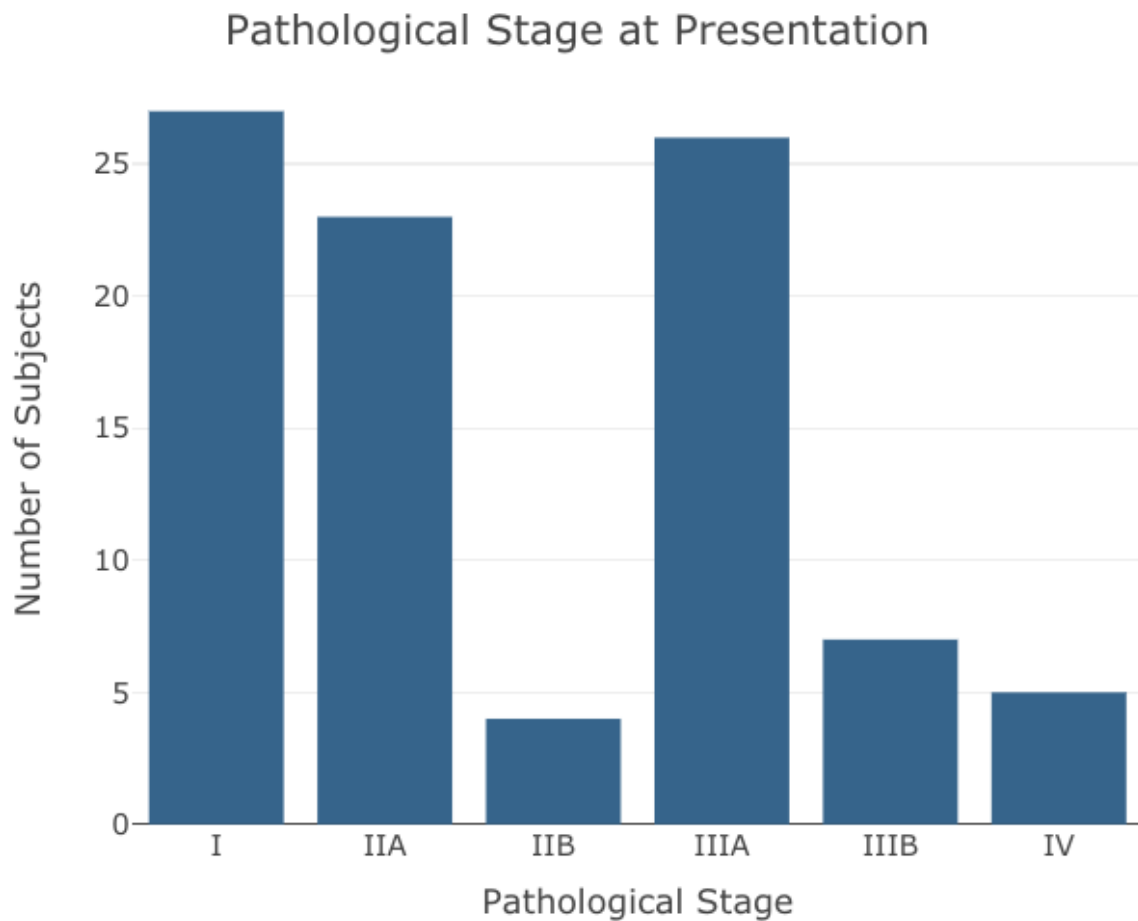
```





Bar Chart of Pathological Staging at Diagnosis

```
pStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_pathstage) %>%
  filter(man_pathstage < 6)
pStageDF <- pStage %>% group_by(man_pathstage) %>% tally()
plot_ly(data = pStageDF) %>%
  add_bars(x = pStageDF$man_pathstage, y = pStageDF$n,
           color = I("steelblue4")) %>%
  layout(
    title = "Pathological Stage at Presentation",
    yaxis = list(title = "Number of Subjects"),
    xaxis = list(title = "Pathological Stage", ticktext = list("I", "IIA", "IIB", "IIIA", "IIIB", "IV"),
                 tickvals = list(0, 1, 2, 3, 4, 5)))
```



### Recapitulate the RMarkdown with the Flexdashboard Template

- What we have above is a step-by-step of all of the code used to generate the figures that will be contained by the Dashboard, as if you were just trying to create those figures separately in R without a Dashboard per se
- What we have below is a more direct representation of where you would put that code to generate the following Flexdashboard



- Below are 3 chunks of code - referred to as R chunks in RMarkdown. This is all of the code used to generate the above FlexDashboard. These chunks also include code to load the data (here an excel file), which we've placed in the first code chunk under the `### Chart A` section.
- Immediately below is the code, and importantly the format, that you would enter in an RMarkdown.
- A screen shot of the RMarkdown appears just afterwards.

Column {data-width=650}

-----  
**### Chart A - Age at Diagnosis**

```
dt <- read_excel("mcc_cohort_fake.xlsx") # Load the data
Age_at_Dx <- dt %>% select(record_id, age_at_dx) %>% drop_na(age_at_dx)
Age_at_Dx$subjects <- "subjects" # add a column that unifies all the data (helpful for plotly)
Age_at_Dx %>% kable
plot_ly(data = Age_at_Dx, type = "box") %>%
  add_boxplot(x = Age_at_Dx$subjects, y = Age_at_Dx$age_at_dx,
             boxpoints = "all", jitter = 0.3, pointpos = -1.8,
             marker = list(color = 'rgb(7,40,89)'),
             line = list(color = 'rgb(7,40,89)'),
             color = I("steelblue4"),
             name = "MGH-HCC Cohort") %>%
  layout(title = "Age at Diagnosis of MCC")
```

Column {data-width=350}

-----  
**### Chart B - Clinical Stage at Diagnosis**

```
cStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_clinstage) %>%
  filter(man_clinstage < 98)
cStageDF <- cStage %>% group_by(man_clinstage) %>% tally()
```

```

plot_ly(data = cStageDF) %>%
  add_bars(x = cStageDF$man_clinstage, y = cStageDF$n,
           color = I("steelblue4")) %>%
  layout(
    title = "Clinical Stage at Presentation",
    yaxis = list(title = "Number of Subjects"),
    xaxis = list(title = "Clinical Stage", ticktext = list("I", "IIA", "IIB", "III", "IV"),
                 tickvals = list(0, 1, 2, 3, 4))
  )

```

### ### Chart C - Pathological Stage at Diagnosis

```

pStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_pathstage) %>%
  filter(man_pathstage < 6)
pStageDF <- pStage %>% group_by(man_pathstage) %>% tally()
plot_ly(data = pStageDF) %>%
  add_bars(x = pStageDF$man_pathstage, y = pStageDF$n,
           color = I("steelblue4")) %>%
  layout(
    title = "Pathological Stage at Presentation",
    yaxis = list(title = "Number of Subjects"),
    xaxis = list(title = "Pathological Stage", ticktext = list("I", "IIA", "IIB", "IIIA", "IIIB", "IV"),
                 tickvals = list(0, 1, 2, 3, 4, 5))
  )

```

## This is what that RMarkdown looks like with the FlexDashboard Template

```

1 ---
2 title: "Age at Dx, Clinical and Pathological Stage at Diagnosis"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7   ---
8
9   {r setup, include=FALSE}
10  library(flexdashboard)
11  library(tidyverse)
12  library(xlsx)
13  library(plotly)
14  library(readxl)
15  library(scales)
16  ---
17
18  Column (data-width=50)
19  -----
20
21  ## Age at Diagnosis of MCC
22  {r}
23
24  dt <- read_excel("mcc_cohort_fake.xlsx")
25  Age_at_Dx <- dt %>% select(record_id, age_at_dx) %>% drop_na(age_at_dx) # drop_na is a good function to eliminate rows that have missing values
26  Age_at_Dx$subjects <- "subjects" # add a column that unifies all the dots (helpful for plotly)
27  # graph
28  plot_ly(data = Age_at_Dx, type = "box") %>%
29  add_barpLOT(x = Age_at_Dx$subjects, y = Age_at_Dx$age_at_dx,
30            boxpoints = "all", jitter = 0.3, pointsize = 1.8,
31            marker = list(color = "rgb(7,48,83)",
32                          line = list(color = "rgb(7,48,83)",
33                                      color = I("steelblue4")),
34            name = "MCC Cohort") %>%
35  layout(title = "Age at Diagnosis of MCC")
36  ---
37
38  Column (data-width=50)
39  -----
40
41  ## Clinical Stage at Diagnosis
42  {r}
43
44  cStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_clinstage) %>% filter(man_clinstage < 98)
45  cStageDF <- cStage %>% group_by(man_clinstage) %>% tally()
46  plot_ly(data = cStageDF) %>%
47  add_bars(x = cStageDF$man_clinstage, y = cStageDF$n,
48          color = I("steelblue4")) %>%
49  layout(
50    title = "Clinical Stage at Presentation",
51    yaxis = list(title = "Number of Subjects"),
52    xaxis = list(title = "Clinical Stage", ticktext = list("I", "IIA", "IIB", "III", "IV"), tickvals = list(0, 1, 2, 3, 4))
53  )
54  ---
55  ## Pathological Stage at Diagnosis
56  {r}
57
58  pStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_pathstage) %>% filter(man_pathstage < 6)
59  pStageDF <- pStage %>% group_by(man_pathstage) %>% tally()
60  plot_ly(data = pStageDF) %>%
61  add_bars(x = pStageDF$man_pathstage, y = pStageDF$n,
62          color = I("steelblue4")) %>%
63  layout(
64    title = "Pathological Stage at Presentation",
65    yaxis = list(title = "Number of Subjects"),
66    xaxis = list(title = "Pathological Stage", ticktext = list("I", "IIA", "IIB", "IIIA", "IIIB", "IV"), tickvals = list(0, 1, 2, 3, 4, 5))
67  )

```

## Additional FlexDashboard Formatting to Enhance the User Experience (UX)

### Add Pages to A FlexDashboard

- If you have numerous data visualizations in your dataset that you want to include in your FlexDashboard, dividing the dashboard into multiple pages can improve the overall UX

- Each page is defined by a level 1 markdown header, (=====), and will have an individual navigation tab
  - For example, in the data shown above, we may want to have the **Age of Diagnosis of MCC** boxplot on a page by itself, and the **Clinical** and **Pathological Staging** bar charts on a second page in the dashboard
- Below is the code and formatting to produce that data visualization

PAGE 1: Age at Diagnosis of MCC

=====

### Chart A - Age at Diagnosis of MCC

```
dt <- read_excel("mcc_cohort_fake.xlsx") # Load the data
Age_at_Dx <- dt %>% select(record_id, age_at_dx) %>% drop_na(age_at_dx)
Age_at_Dx$subjects <- "subjects" # add a column that unifies all the data (helpful for plotly)
Age_at_Dx %>% kable
plot_ly(data = Age_at_Dx, type = "box") %>%
  add_boxplot(x = Age_at_Dx$subjects, y = Age_at_Dx$age_at_dx,
             boxpoints = "all", jitter = 0.3, pointpos = -1.8,
             marker = list(color = 'rgb(7,40,89)'),
             line = list(color = 'rgb(7,40,89)'),
             color = I("steelblue4"),
             name = "MGH-HCC Cohort") %>%
  layout(title = "Age at Diagnosis of MCC")
```

PAGE 2: Staging of MCC

=====

Column {data-width=350}

-----

### Chart B - Clinical Stage at Diagnosis of MCC

```
cStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_clinstage) %>%
  filter(man_clinstage < 98)
cStageDF <- cStage %>% group_by(man_clinstage) %>% tally()
plot_ly(data = cStageDF) %>%
  add_bars(x = cStageDF$man_clinstage, y = cStageDF$n,
          color = I("steelblue4")) %>%
  layout(
    title = "Clinical Stage at Presentation",
    yaxis = list(title = "Number of Subjects"),
    xaxis = list(title = "Clinical Stage", ticktext = list("I", "IIA", "IIB", "III", "IV"),
                tickvals = list(0, 1, 2, 3, 4)))
```

### Chart C - Pathological Stage at Diagnosis of MCC

```
pStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_pathstage) %>%
  filter(man_pathstage < 6)
pStageDF <- pStage %>% group_by(man_pathstage) %>% tally()
plot_ly(data = pStageDF) %>%
```

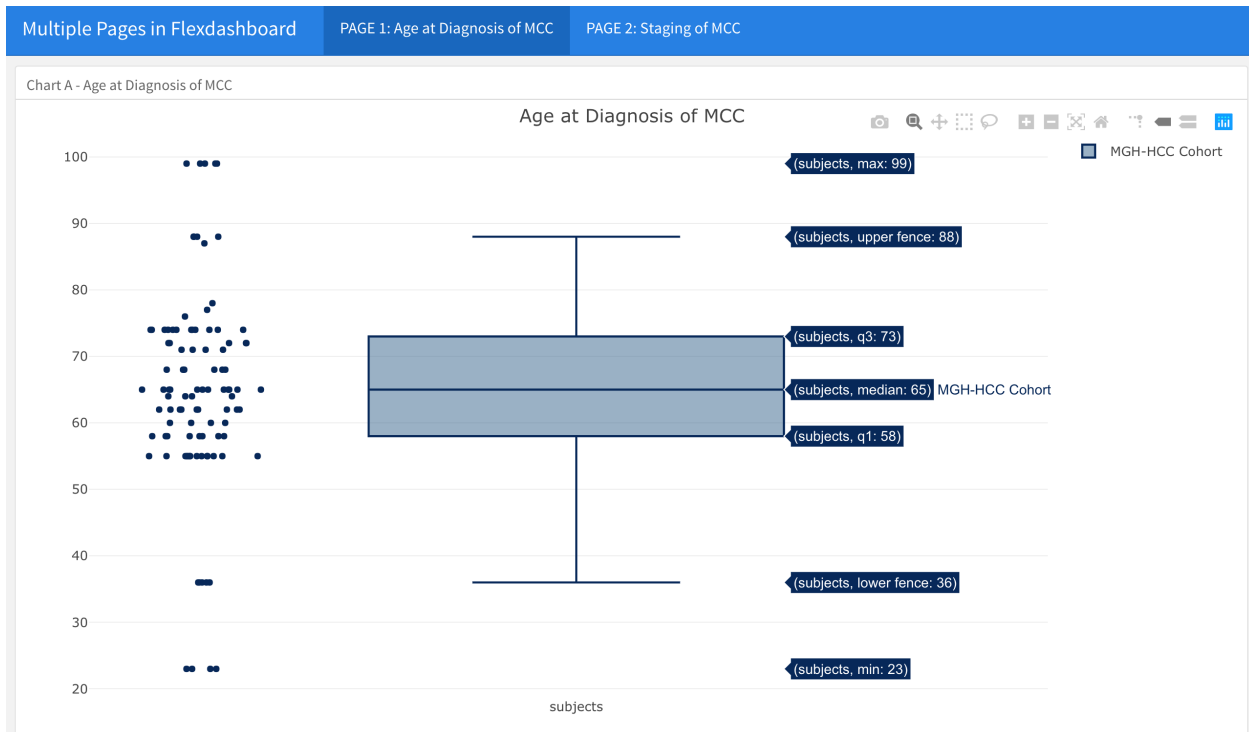
```

add_bars(x = pStageDF$man_pathstage, y = pStageDF$n,
         color = I("steelblue4")) %>%
layout(
  title = "Pathological Stage at Presentation",
  yaxis = list(title = "Number of Subjects"),
  xaxis = list(title = "Pathological Stage", ticktext = list("I", "IIA", "IIB", "IIIA", "IIIB", "IV"),
              tickvals = list(0, 1, 2, 3, 4, 5))

```

Again, the “=====” below the Page Header on the first page “PAGE 1: Age at Diagnosis” and below the Page Header of the second page “PAGE 2: Staging of MCC” are critical pieces of the formatting. This encodes for the subsequent code chunks to be placed on individual pages.

**This is what page 1 of the dashboard now looks like**



**And this is what page 2 of the dashboard now looks like**

Chart B - Clinical Stage at Diagnosis of MCC

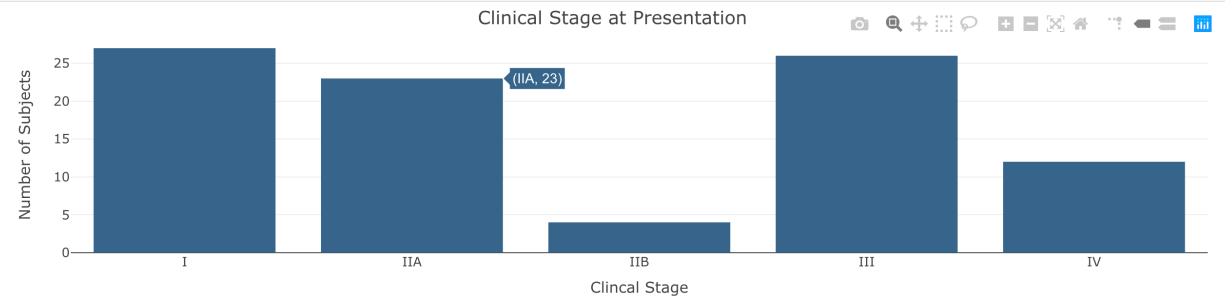
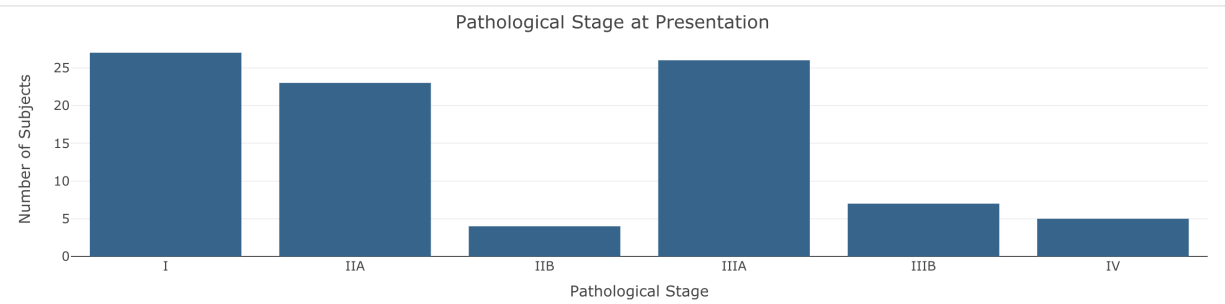


Chart C - Pathological Stage at Diagnosis of MCC



## Add Tabs to A FlexDashboard

- If your Dashboard has a lot of content, you may also want to add tabs within pages to further layer the data presentation
- Instead of using `Column {data-width=350}` above the dotted lines in your FlexDashboard, use `Column {.tabset}`

### PAGE 1: Age at Diagnosis of MCC

```
=====  
### Chart A - Age at Diagnosis of MCC
```

```
dt <- read_excel("mcc_cohort_fake.xlsx") # Load the data  
Age_at_Dx <- dt %>% select(record_id, age_at_dx) %>% drop_na(age_at_dx)  
Age_at_Dx$subjects <- "subjects" # add a column that unifies all the data (helpful for plotly)  
Age_at_Dx %>% kable  
plot_ly(data = Age_at_Dx, type = "box") %>%  
  add_boxplot(x = Age_at_Dx$subjects, y = Age_at_Dx$age_at_dx,  
             boxpoints = "all", jitter = 0.3, pointpos = -1.8,  
             marker = list(color = 'rgb(7,40,89)'),  
             line = list(color = 'rgb(7,40,89)'),  
             color = I("steelblue4"),  
             name = "MGH-HCC Cohort") %>%  
  layout(title = "Age at Diagnosis of MCC")
```

### PAGE 2: Staging of MCC

```
=====
```

Column {.tabset}

-----  
### Chart B - Clinical Stage at Diagnosis of MCC

```
cStage <-dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_clinstage) %>%
  filter(man_clinstage < 98)
cStageDF <- cStage %>% group_by(man_clinstage) %>% tally()
plot_ly(data = cStageDF) %>%
  add_bars(x = cStageDF$man_clinstage, y = cStageDF$n,
           color = I("steelblue4")) %>%
  layout(
    title = "Clinical Stage at Presentation",
    yaxis = list(title = "Number of Subjects"),
    xaxis = list(title = "Clinical Stage", ticktext = list("I", "IIA", "IIB", "III", "IV"),
                 tickvals = list(0, 1, 2, 3, 4))
```

### Chart C - Pathological Stage at Diagnosis of MCC

```
pStage <-dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_pathstage) %>%
  filter(man_pathstage < 6)
pStageDF <- pStage %>% group_by(man_pathstage) %>% tally()
plot_ly(data = pStageDF) %>%
  add_bars(x = pStageDF$man_pathstage, y = pStageDF$n,
           color = I("steelblue4")) %>%
  layout(
    title = "Pathological Stage at Presentation",
    yaxis = list(title = "Number of Subjects"),
    xaxis = list(title = "Pathological Stage", ticktext = list("I", "IIA", "IIB", "IIIA", "IIIB", "IV"),
```

This is the rmd of the FlexDashboard with the green arrow highlighting this critical line of code

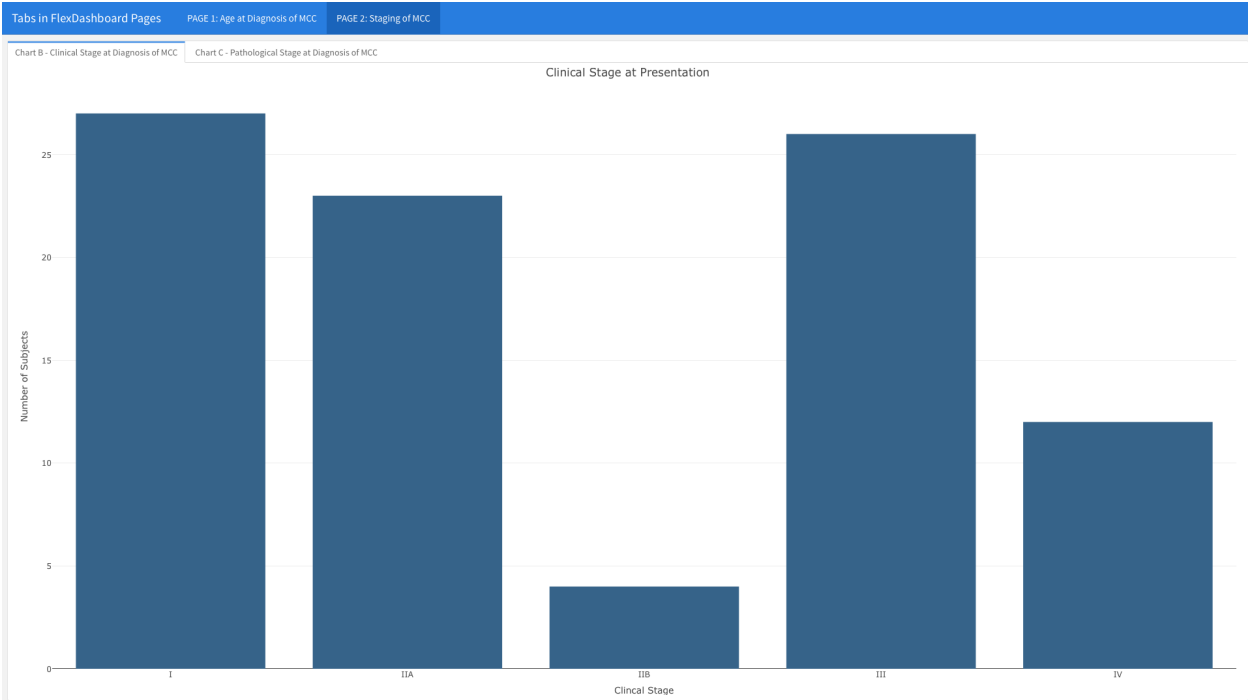


```

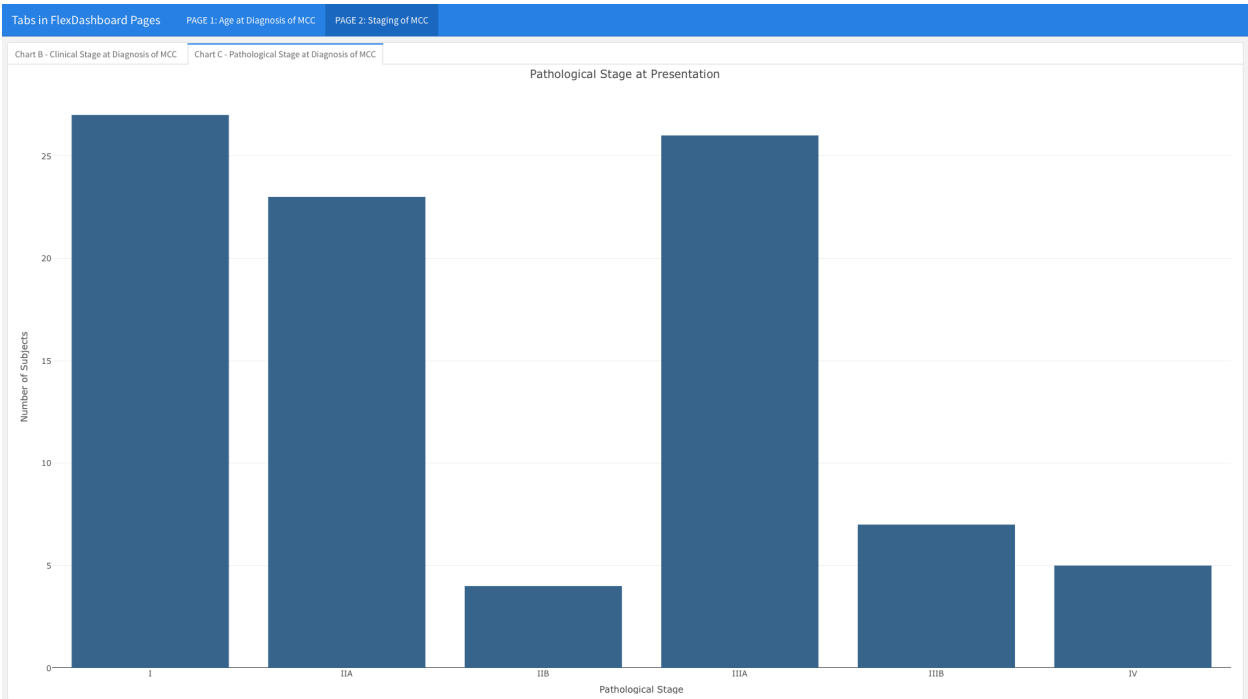
2 title: "Tabs in FlexDashboard Pages"
3 output:
4 flexdashboard::flex_dashboard:
5 orientation: columns
6 vertical_layout: fill
7 ---
8
9 ```{r setup, include=FALSE}
10 library(flexdashboard)
11 library(RCurl)
12 library(REDCapR)
13 library(httr)
14 library(tidyverse)
15 library(knitr)
16 library(plotly)
17 library(readxl)
18 library(scales)
19 library(plotly)
20 library(viridis)
21 library(tidyr)
22 library(scales)
23 library(surminer)
24 library(survival)
25 library(reshape2)
26 library(dplyr, warn.conflicts = FALSE)
27 ---
28
29 PAGE 1: Age at Diagnosis of MCC
30 =====
31 ## Chart A - Age At Diagnosis
32
33 ```{r}
34 dt <- read_excel("mcc_cohort_fake.xlsx") # Load the data
35 Age_at_Dx <- dt %>% select(record_id, age_at_dx) %>% drop_na(age_at_dx) # drop_na is a good function to eliminate rows that have missing values
36 Age_at_Dx$subjects <- "subjects" # add a column that unifies all the data (helpful for plotly)
37
38 plot_ly(data = Age_at_Dx, type = "box") %>%
39   add_boxplot(x = Age_at_Dx$subjects, y = Age_at_Dx$age_at_dx,
40             boxpoints = "all", jitter = 0.3, pointpos = -1.8,
41             marker = list(color = "rgb(7,40,89)"),
42             line = list(color = "rgb(7,40,89)"),
43             color = I("steelblue4"),
44             name = "MGH-HCC Cohort") %>%
45   layout(title = "Age at Diagnosis of MCC")
46 ---
47
48 PAGE 2: Staging of MCC
49 =====
50 Column {, tabset}
51 ## Chart B - Clinical Stage at Diagnosis of MCC
52
53 ```{r}
54 cStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_clinstage) %>% filter(man_clinstage < 98)
55 cStageDF <- cStage %>% group_by(man_clinstage) %>% tally()
56 plot_ly(data = cStageDF) %>%
57   add_bars(x = cStageDF$man_clinstage, y = cStageDF$n,
58           color = I("steelblue4")) %>%
59   layout(
60     title = "Clinical Stage at Presentation",
61     yaxis = list(title = "Number of Subjects"),
62     xaxis = list(title = "Clinical Stage", ticktext = list("I", "IIA", "IIB", "III", "IV"), tickvals = list(0, 1, 2, 3, 4))
63   )
64 ---
65 ## Chart C - Pathological Stage at Diagnosis of MCC
66
67 ```{r}
68 pStage <- dt %>% select(record_id, man_clinstage, man_pathstage) %>% drop_na(man_pathstage) %>% filter(man_pathstage < 6)
69 pStageDF <- pStage %>% group_by(man_pathstage) %>% tally()
70 plot_ly(data = pStageDF) %>%
71   add_bars(x = pStageDF$man_pathstage, y = pStageDF$n,
72           color = I("steelblue4")) %>%
73   layout(
74     title = "Pathological Stage at Presentation",
75     yaxis = list(title = "Number of Subjects"),
76     xaxis = list(title = "Pathological Stage", ticktext = list("I", "IIA", "IIB", "IIIA", "IIIB", "IV"), tickvals = list(0, 1, 2, 3, 4, 5))
77   )
78 ---
79
80 Tabs in FlexDashboard Pages

```

This is now what page 2 of the dashboard looks like



You will notice that the **Clinical** and **Pathological Staging** bar charts are no longer stacked in two rows on the same page. In contrast, you only see the **Clinical Stage at Presentation** graph, which takes up the entire page. You can see the tab in the upper left corner; if you click on the second tab, the chart for **Pathological Stage at Presentation** will emerge (see below).



## Take Home Points

- Dashboards are an excellent data visualization tool for Clinical and Translational Research
- Flexdashboard is a great package to develop dashboards with R
- Adding pages and tabs to your dashboard can create a richer user experience for your intended audience

As always, please reach out to us with thoughts and feedback

## Session Info

```
sessionInfo()
```

```
## R version 4.0.0 (2020-04-24)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
##
## other attached packages:
## [1] scales_1.1.0 readxl_1.3.1 plotly_4.9.2.1
## [4] knitr_1.28 forcats_0.5.0 stringr_1.4.0
## [7] dplyr_0.8.5 purrr_0.3.4 readr_1.3.1
## [10] tidyr_1.0.3 tibble_3.0.1 ggplot2_3.3.0
## [13] tidyverse_1.3.0 httr_1.4.1 REDCapR_0.11.0
## [16] RCurl_1.98-1.2 flexdashboard_0.5.1.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.4.6 lubridate_1.7.8 lattice_0.20-41 assertthat_0.2.1
## [5] digest_0.6.25 R6_2.4.1 cellranger_1.1.0 backports_1.1.6
## [9] reprex_0.3.0 evaluate_0.14 highr_0.8 pillar_1.4.4
## [13] rlang_0.4.6 lazyeval_0.2.2 rstudioapi_0.11 data.table_1.12.8
## [17] rmarkdown_2.1 htmlwidgets_1.5.1 munsell_0.5.0 broom_0.5.6
## [21] compiler_4.0.0 modelr_0.1.7 xfun_0.13 pkgconfig_2.0.3
## [25] htmltools_0.4.0 tidyselect_1.0.0 fansi_0.4.1 viridisLite_0.3.0
## [29] crayon_1.3.4 dbplyr_1.4.3 withr_2.2.0 bitops_1.0-6
## [33] grid_4.0.0 nlme_3.1-147 jsonlite_1.6.1 gtable_0.3.0
## [37] lifecycle_0.2.0 DBI_1.1.0 magrittr_1.5 cli_2.0.2
## [41] stringi_1.4.6 fs_1.4.1 xml2_1.3.2 ellipsis_0.3.0
## [45] generics_0.0.2 vctrs_0.2.4 tools_4.0.0 glue_1.4.0
## [49] crosstalk_1.1.0.1 hms_0.5.3 yaml_2.2.1 colorspace_1.4-1
## [53] rvest_0.3.5 haven_2.2.0
```