I took advantage of this course’s final project to apply the skills I’ve learned to a workplace need for an accessible data dashboard. I work at a small nonprofit that uses Salesforce to track our over 20 years worth of donor data. This database is only accessible to a handful of staff, and none of our board members, due to licensing and costs. Thus, I was recently asked if I could find a way to display some of our metrics on a password protected page of our website. I chose to learn flexdashboard in R to create a Shiny-based dashboard for this purpose.

I used the *keyring* library to store the necessary API access information. I have a *dataprep* .Rmd file that uses a *salesforcer* package to query our Salesforce database using SQL statements, in order to pull in data from the donation object and the account object. This is cleaned and joined together, and then I created a variety of new variables to manage my boss’ request for all data to be in reference to our fiscal year, which runs April 1 – March 31, instead of the calendar year. It also took some time to create variables for donor retention in this year, last year, and two years ago to look and see if the donor had been retained from year to year. The resulting dataframe is saved as a .Rdata file to be imported into the app.Rmd file. The app.Rmd file is structured similarly to an R Markdown file, with a few ways to denote row/column sections and sizes for the adaptive app it creates.

The dashboard includes a Notes section, as I know specific users will want this information though I don’t want it cluttering my actual data displays. After that, the first data tab displays Individual Donor data – the distinction is that these are donations from Household accounts, not Business, Nonprofit, Government, or Foundation (our other account types in Salesforce). All data is display in comparison to the past 3 fiscal years so the users can see if we are doing better/worse on each measure. The next tab, Donor Retention, is very important as knowing if a large pool of donors gave last year, but not yet again this year, can spark action from the marketing team to hopefully re-engage those donors. I did make an active choice on these two tabs to *not* include filters or interactivity, with my team I find it’s best not to let them get overwhelmed by those features, and instead if they have a question on how the data looks when filtered a certain way, I’d rather make an additional tab or chart.

In terms of the code, my biggest learning in this project was how easy it really easy to get a dashboard up and running with flexdashboard. As I didn’t need the reactivity, a full-blown Shiny dashboard wasn’t necessary for this project. Further, the ggplot graphs weren’t sizing very well but I found a fantastic *ggplotly()* function that wraps a standard ggplot call and outputs a plotly graph. This also gave me the hover-text which I think greatly reinforces how users interpret the graphs, and eliminated the need to clutter these relatively small graph tiles with excessive labels and legend.

Finally, I expect this dashboard to eventually have 6+ tabs as I work through other staff requests such as mapping donor locations, monthly donor trends, mail campaign analysis, new donor acquisition, and more.