TOPIC: Why would someone use Python for EDA over R or another language? What are the pros and cons of the languages you have selected to compare? What are some out of the box libraries or packages that are available for EDA?

I picked this topic to compare Python vs R for exploratory data analysis (EDA) because 1) I finally have experience with both after taking DSC 510 and 520 and 2) I was interested in learning more about the differences in these tools, especially for preparing for analysis. To be honest, going into the program I thought I would like Python more, but R syntax actually seemed more intuitive to me. But hoping learning more about Python this quarter will increase my understanding on how to use it.

For R, the main package for EDA is ggplot2 and dplyr. I used this last quarter for my final project, as ggplot made it easy for me to visualize patterns in my data and spot any outliers that needed to be reviewed more closely. The dplyr package is also great for data analysis, as it allows you to group, summarize or filter your data.

For Python, there are also several packages available that help you with EDA and statistical analysis. Matplotlib is similar to ggplot in R, as it gives you the ability to plot your data on a scatterchart or create a bar chart to show you trends in your data visually. While I haven’t used it yet, the scripting does look a little more complicated than ggplot, although practice and real-life examples will help with the learning curve. It also doesn’t look as polished as some of the ggplot charts I’ve seen using markdown, but I am sure there are ways to improve the visuals as well as you learn more advanced techniques.

Another popular package for EDA and statistical analysis in Python is pandas, which can connect to a dataset and create a data frame, similar to R or other analysis software. You can view top N rows of the data to see what the overall structure of the dataset is like, and you can also do basic summary statistics on the data like mean, standard deviation and min/max values to see high level trends in your data. In addition to doing similar functions as dplyr with sorting and filtering the data, you can also use data cleaning techniques such as checking for nulls, as well as replacing or renaming certain variables in your data.

In my brief understanding of both languages, Python was developed as a more all-around language where R was built primarily for analysis rather than data manipulation. Both languages have evolved over time to fill in their weaknesses, as packages have been developed to improve data manipulation in R (like dplyr, mentioned above), while Python has had improvements in its ability to do statistical analysis, with packages like pandas. Since Python is a more powerful programming tool than R, it makes sense to Python over other languages if you have more complex data manipulation needed or a large amount of data to analyze. Since Python seems to be more flexible since it was designed as a programming language, it may make getting the data in the shape you want for analysis before you create your data frame easier than R or other languages.

References

Siddiqi, A. (2018, March 3). *Introduction to Explatory Data Analysis in Python.* Retrieved from: https://analyticsindiamag.com/exploratory-data-analysis-in-python-vs-r/

Kodali, T. (2015, August 20). *Data Manipulation with dplyr.* Retrieved from: https://www.r-bloggers.com/data-manipulation-with-dplyr/

https://www.geeksforgeeks.org/python-introduction-matplotlib/

Bronshtein, A. (2017, April 17). *A Quick Introduction to the "Pandas" Python Library*. Retrieved from: https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library-f1b678f34673