4.1 Lesson Summary - Introduction to Pandas & Jupyter

Python is a powerful programming language well-suited to data analytics. There are a number of different programs that you can use to write Python code and Jupyter Notebook is one of the most popular. Jupyter Notebook provides a free editor that allows you to break your code up into manageable pieces and view the results. Pandas elevates Python by providing objects like tables (DataFrames) and arrays (DataSeries) that contain additional functionality to aid in data processing and do so more efficiently than is possible with standard Python objects.

Concept: **Jupyter Notebook** organizes your executable Python code into cells. This allows you to edit and run manageable sets of Python instructions in order to methodically build to your desired result. The output of your code, including charts, is displayed after the cells allowing you to more easily visualize your code’s result.

* Activity: 01-Ins\_JupyterIntro, 02-Stu\_NetflixRemix
* Suppl link: <https://jupyter.org/>

Concept: **Pandas** provides a number of tools to allow a Python programmer to process data more efficiently. A Pandas **DataSeries** is a one dimensional array of data similar to a Python list. A Pandas DataSeries can be created using the following code:

*data\_series = pd.Series(["First Row", "Second Row", "Third Row"])*

Pandas **DataFrames** provide the benefit of a two-dimensional, table style data object for data processing. This allows you to organize more complex data sets in an intuitive manner with the option to describe the type of data, label the columns of your data, and organize the contents into rows. You can create a Pandas DataFrame using the following code:

*customer\_dicts = [{"CUSTOMER": "Jonathan"}, {"CUSTOMER": "Susan"}]*

*customer\_df = pd.DataFrame(customer\_dicts)*

* Activity: 03-Ins\_IntroToPandas, 04-Stu\_DataFrameShop
* Suppl link: <https://pandas.pydata.org/>

Concept: You can get a statistical overview of a Pandas DataFrame by using the *describe* method illustrated in the following code:

*data\_csv\_file\_df.describe()*

The columns of a DataFrame can be accessed by specifying them in a square bracketed list following the DataFrame object. You can target just the first rows of the DataFrame using the *head* method. For example:

*data\_file\_df[["FirstColumn", "SecondColumn"]].head()*

Pandas allows you to compute statistics like **sum**, **mean**, count, and display the unique values of a DataFrame column using the following code:

*data\_file\_df["FirstColumn"].sum()*

Pandas provides you with the functionality to rename and reorganize the columns of your DataFrame.

* Activity: 05-Ins\_DataFunctions, 06-Stu\_TrainingGrounds, 07-Ins\_ColumnManipulation, 08-Stu\_Hey\_Arnold

Concept: **CSV** files provide a portable and versatile way to store table data. The following code can be used to read data from a CSV file into a Pandas DataFrame:

*data\_csv\_file\_df = pd.read\_csv(“Resources/mydata.csv”)*

*data\_csv\_file\_df.head()*

Pandas also provides functionality to export data directly from a data frame into a CSV file. For example:

*data\_to\_export\_df.to\_csv("Output/fileOne.csv", index=False, header=True)*

* Activity: 09-Ins\_ReadingWritingCSV, 10-Stu\_GoodReads, 11-Stu\_GoodReadsSummary

Jupyter Notebook provides an elegant interface to write Python code and view its results. Pandas provides functionality to organize and process data.