* Python Packages for Data Science.
  + Scientifics Computing Libraries
    - Pandas -Data Structures & tools
    - Numpy – Arrays & Matrices
    - Scipy – Integrals, solving differential equations, optimization.
  + Visualization Libraries
    - Matplotlib – Plots & Graphs, most popular
    - Seaborn – Plots : heat maps, time series, violin plots.
  + Algorithmic Libraries
    - Scikit-learn – Machine Learning : regression, Classification
    - Statsmodels – Explore data, estimate statistical models, and perform statistical tests.
* Importing and Exporting Data in Python
  + Format
    - .csv, .json, .xlsx, .hdf
  + File path of dataset
    - /Desktop/mydata.csv
  + Reading data using pandas with header
    - **Import pandas as pd**
    - **url = “hdalsjkhflasjsdhflakjsdhf’**
    - **df = pd.read\_csv(url, names = headers) \*add the column headers as a list.**
  + Reading data without header
    - **Import pandas as pd**
    - **url = “dskjdalsdkjfalskdjf’**
    - **df = pd.read\_csv(url, header = None)**
  + Printing dataframe
    - **df.head(n) –** first number of rows
    - **df.tail(n) –** last number of rows.
  + Applying Column names (or just to print column names)
    - **df.columns = headers. (headers is a list ex: headers=[“ “])**
  + Replacing data in a Dataframe
    - **df.replace(‘ ‘, np.NaN)**
  + Dropping columns
    - **df.drop('aspiration', axis = 1, inplace=True)**
    - **df1.dropna(subset=["price"], axis=0)**
  + Exporting dataframe to CSV
    - set a path**: path = “C:/Windows/…/sdjshadjf.csv”**
    - **Table

      Description automatically generateddf.to\_csv(path, index = False) \*The index in this case is used to indicate that row names will not be written.**
  + Basic Analysis
    - Methods
    - Understand the data before you begin any analysis.
      * Check
        + Data Types

**df.dtypes**

returns a table with the list of data attributes and their type.

* + - * + Data Distribution

**df.describe()**

returns a basic statistical summary of the data.

**dataframe[[' column 1 ',column 2', 'column 3'] ].describe()**

This can be used to specify which columns to describe rather than all.

**df.describe(include=”all”)**

provides a full summary statistic.

**df.info()**

returns the top 30 rows and the bottom 30 rows of the dataframe.

* + - * + **Graphical user interface

          Description automatically generated with low confidence**Locate potential issues with the data.
* Graphical user interface, application

  Description automatically generatedAccessing databases using Python
  + DB-API
    - Python standard api for databases that works with python.
    - Concepts
      * Connection Objects
        + Database connections
        + Manage Transactions
      * Cursor Objects
        + Data Queries.
      * Connection methods
        + Cursor()
        + Commit()
        + Rollback()
        + Close()

Graphical user interface, text, application, email

Description automatically generated**Graphical user interface, text, application

Description automatically generated**