**Pandas Glossary**

To open Pandas: terminal 🡪 jupyter notebook

Start Pandas module: **import pandas as pd**

CREATING DATA FRAMES:

* *List of dictionaries:* 
  + **variable = pd.DataFrame([{“Name of Column1 ”: “value”, “Col2”: “value”},**

**{“Name of Column1 ”: “value”, “Col2”: “value”},**

**“Name of Column1 ”: “value”, “Col2”: “value”}])**

* *Dictionary of lists:*
  + **variable1 = pd.DataFrame({“Column1”: [“value”, “value”, “value”],**

**“Column2”: [“value”, “value”, “value”],**

**“Column2”: [“value”, “value”, “value”]})**

OPEN/READ/WRITE FILES:

* *Open/read:*
  + **variable1 = “../Resources/sample.csv”**
  + **df = pd.read\_csv(variable1)**
  + **df.head()**
* *Write/save:*
  + *CSV*: **df.to\_csv(“../Resources/file.csv”, index=False, header=True)**
  + *EXCEL*: **df.to\_excel(“../Resources/file.csv”, index=False, header=True)**

PRINTING:

* *To print, just type the variable (don’t need “’print()”)*
* *Print first x amount of rows:* **variable.head(x)** *(if no x, defaults to showing 5 rows)*

DATA OPERATIONS:

* *Describe/summarize data:* **df.describe()**
* *Average a column*: **df[“Column”].mean()**
* *Find the max value in a column*: **df[“Column”].max()**
* *Find the minimum value in a column*: **df[“Column”].min()**
* *Find the unique values in a column:* **df.[“Column”].unique()**
* *Show how many times each value occurs in a column:* **df[“column”].value\_counts()**
* *Find the length of a dataframe:* **variable = len(df)**

MANIPULATE COLUMNS:

* *Show one column:* ***df[“Column”].head()***
* *Show multiple columns:* **df[[“Column1”, “Column2”]].head()**
* *Reorganize*: **newdf = df[[“Col1”, “Col2”, “Col3”]]**
* *Remove*: **newdf = df[[“”, “”, “”]]** *(enter only the columns you want to keep)*
* *Rename:* **newdf = df.rename(columns = {“old name col 1”: “new name col 1”,**

**“old name col 2”: “new name col 2”,**

**“old name col 3”: “new name col 3”})**

* *List of columns:* **df.columns**
* *Add column:* **df[“name of new column”] = df[“column”] – 100** *(add calculation for new column)*
* *Delete a column:* **del df[“Column”]**

INDEXES:

* *Change index*: **newdf = df.set\_index(“name of col to use as index”)**
* *Print one row:* **variable = df.loc[“index”]**
* *Grab info from single cell:* **variable = df.loc[“index”, “column1”]**
* *Grab info from multiple cells:* **variable = df.loc[“index”, [“column1”, “column2”]]**
* *First “” is index, second “”s are columns*
* *Grab info from default index:* **variable =** **df.iloc[numberofindex, numberofcolumn]**

FILTERS (LOC/ILOC):

* *Show all rows and certain columns:* **df.loc[:, [“column1”, “column2”]]**
* *Show all columns and certain rows:* **df.loc[[“rowvalue”, “rowvalue”], :]**
* *Show rows with one condition:* **variable = df.loc[df[“row”] == “value”, :]**
* *Show rows with multiple conditions:* **variable = df.loc[(df[“row1”] == “value”) | (df[“row2”] =**

**“value”), :]**

SORTING:

* *Sort a column:* **df.sort\_values([“column1”, “column2”], ascending=False/True)** *(ascending true = lowest to highest; ascending false = highest to lowest; sorts by first column first, then next, etc*
* *Reset the index values for the newly-sorted dataframe:* **new\_index = df.reset\_index(drop=True)** *(if you don’t do drop=Trues, then the old index values become a new column)*

CLEANING DATA:

* *Count how much data is in each row:* **df.count()**
* *Drop rows with missing information:* **df = df.dropna(how=’any’)**
* *Show data types for each column:* **df.dtypes**
* *Change data type to numeric:* **df[“column”] = pd.to\_numeric(df[“column])**
* *Edit data values:* **df[“column”] = df[“column”].replace({“old value”: “new value”,**

**“old value: “new value”})**

MERGING DATA:

* *Inner join (only rows that have matching data):* **merge\_table = pd.merge(first\_df, second\_df, on=”column”)**
* *Outer join (all rows):* **merge\_table = pd.merge(first\_df, second\_df, on=”column”, how=”outer”)**
* *Left join (all from first df and matching from second):* **merge\_table = pd.merge(first\_df, second\_df, on=”column”, how=”left”)**
* *Right join (all from second df and matching from first):* **merge\_table = pd.merge(first\_df, second\_df, on=”column”, how=”right”)**

GROUPBY:

* *GroupBy:* **df2 = df1.groupby([“column”])**
* *To visualize the GroupBy:* **df2.count()**
* *GroupBy more than one thing:* **df2 = df1.groupby([“column1”, “column2”])**