Common Imports

import numpy as np

import pandas ps pd

import matplotlib.pyplot as plt

import seaborn as sns

Vectorized Operations

xs + ys : : : : : : : : : : : : : : : : : : : : : : : : Element-wise addition

xs + z : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : Adding a scalar

xs & ys : : : : : : : : : : : : : : : : : : : : : : : : Bitwise (boolean) and

xs | ys : : : : : : : : : : : : : : : : : : : : : : : : : : bitwise (boolean) or

xs : : : : : : : : : : : : : : : : : : : : : : : : : : : : : Bitwise (boolean) not

xs < ys : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : Less than

Subtraction (-), multiplication (\*), division (/), expo-

nentiation (\*\*), and other comparison operators (<=,

>, >=, ==, !=) work similarly.

“+” with df and series adds along rows by default. The .add() method (and other named vectorized operations) can add along an axis = [“rows”, “columns”, “index”]. Index adds along the columns, columns and rows do the opposite of what you’d expect—rows adds the series to df columns, and vice-versa.

matplotlib plotting

plt.hist(xs): : : : : : : : : : : : : : : : : : : : : : : : : : : : : :Histogram

plt.scatter(xs, ys): : : : : : : : : : : : : : : : : : : : :Scatterplot

plt.plot(xs, ys) : : : : : : : : : : : : : : : : : : : : : : : : : : Line plot

Array / Series functions

max() : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : Maximum

min() : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : Minumum

mean(): : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :Mean (average)

median() : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : Median

sum(): : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :Sum (total)

Accessing Data in a Series

s.iloc[i]: : : : : : : : : : : : : : : : : : : :Get element by position

s.loc[x] : : : : : : : : : : : : : : : : : : : : : : : Get element by index

s.values: : : : : : : : : : : : : : : : : : : : : : : : : : :Get NumPy array

Plotting for Series

s.hist() : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : Histogram

s.plot(): : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :Line plot

Apply Functions

s.apply(value -> value) : : : : : : : : : : : returns a Series

df.applymap(value -> value) : returns a DataFrame

df.apply(series -> value) : : : : : : : : : returns a Series

df.apply(series -> series): : :returns a DataFrame

Accessing Data in a DataFrame

df['col']: : : : : : : : : : : : : : : : : : : : : : :Get column by name

df.iloc[i] : : : : : : : : : : : : : : : : : : : : : : Get row by position

df.loc[x] : : : : : : : : : : : : : : : : : : : : : : : : : : Get row by index

df.iloc[i, j] : : : : : : : : : : : : : : : Get element by position

df.loc[x, y]: : : : : : : : : : : : : : : : : : :Get element by index

df.values : : : : : : : : : : : : : : : : : : : : : : Get 2D NumPy array

DataFrame Summarization

df.describe() : : : : : : : : : : : : : : Stats about each column

df.head(n) : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : First n rows

df.tail(n): : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :Last n rows

df.columns : : : : : : : : : : : : : : : : : : : : : List of column names

Axis Argument

df.mean(axis=0) : : : : : : : : : : : : : : : mean of each column

df.mean(axis=1) : : : : : : : : : : : : : : : : : : : mean of each row

df.mean(axis='index') : : : : : : : : mean of each column

df.mean(axis='columns' : : : : : : : : : : : : : : : : : : : : : : : : : : : )

mean of each row

Plotting for DataFrames

df.plot() : : : : : : : : : Line plot with one line per column

**DataFrame Alteration**

df.qcut([list of values to be used for ranges, i.e. 0, 1, 3, 4 will assign df elements to label 1, if they are between o and 1], labels = [new values/names to be given if elements are in the corresponding range])

**DataFrame Aggregation**

df.group\_by([“column name”, optional “column 2”]) has attribute .groups, which returns a dictionary with the index mapped to a list containing iloc positions corresponding matching that index.

df.group\_by([“column name”].sum()[“Column name”].mean() would conveniently sum all rows under a common index, then take the mean of the specific column.