Credit Risk L&D:

Beginner’s Python Training Exercises

# Task 1: importing files / libraries / functions

1. Import library “numpy” and “pandas” and the testing files from the link
2. Import function “timedelta” from library “datetime”

# Task 2: create copies of data

1. Make a copy of the imported data in the python environment
2. Make a copy of the imported data as local CSV file on your desktop

# Task 3: basic column operations

1. Extract the column names of one imported “AAPL.csv” file, and remove the following two columns: “High”, “Low”
2. Remove the 2nd column from the result of previous step
3. Keep the following columns from the result of previous step: “Date”, “Adj Close”, “Volume”, “Close”
4. Keep only the first 3 columns from the previous step
5. Check if the remaining column names are in this given list: “Date”, “Volume”, “banana”, “apple”, “orange”
6. Sort the dataframe by column name, alphabetically
7. Rename the data frame as “Date”, “AAPL”
8. Replace the column name “Date” by “new\_date”, and “AAPL” by “apple”

# Task 4: merging data and processing missing values

1. Convert the dates from “dates.csv” into %Y-%m-%d format, and use it as master index to align AAPL and IBM data, by using the “merge” function from pandas
2. Count the missing values from the merged data and fill them by the following method:
3. Replace missing values by 0, then replace all 0 by 5
4. Replace missing values by forward filling
5. Replace missing values by mean / mod

# Task 5: data rearrangement / change type and calculations

1. Sort the data by values of AAPL stock price, from both high to low and low to high
2. Convert AAPL stock price from numeric to string, and then onvert it back
3. Add new columns as: the sum of AAPL and IBM price / the difference between AAPL and IBM price
4. Use “np.vectorize” function to calculate the result of 0.5\*(x+y)/sqrt(x\*y), x = AAPL stock price, y = IBM stock price

# Task 6: shifting data and create lags / returns

1. Shift both AAPL and IBM stock price down for 10 days (i.e. creating a 10-day lag) while keeping the date column untouched
2. Shift these two columns back 5 days
3. Create one-day absolute / relative returns for both stocks

# Task 6: date operation

1. Generate all dates between 2011-05-03 and 2012-05-03
2. Keep only business days in the generated date series
3. Perform the following date operations: Subtract 5 days from the date series / add 1 month to the date series / add 1 year to the date series
4. Form the date series, extract the date, month, and year

# Task7: basic matrix operation

1. Convert a pd dataframe to matrix, and then convert it back
2. For a matrix, add / multiply it with a constant
3. Transpose the matrix
4. Produce the following matrix: 5\*5 matrix of zeros / 15\*5 matrix of ones
5. Perform the following matrix operation: A+B / A-B / A\*B
6. Create a square matrix: [1,2,3],[4,6,7],[17,19,23]
7. Use the created square matrix, calculate its inverse / determinant / eigenvalue / eigenvector

# Some helpful Links:

1. Package documents (detailed information about the functions / variables of the library)

Numpy:

<https://docs.scipy.org/doc/>

Pandas:

<https://pandas.pydata.org/pandas-docs/stable/>

Datetime:

<https://docs.python.org/2/library/datetime.html>

A list of Python libraries, based on their speciality

<https://github.com/vinta/awesome-python>

1. Python basics

Strings

<https://github.com/Akuli/python-tutorial/blob/master/basics/handy-stuff-strings.md>

List and tuples

<https://github.com/Akuli/python-tutorial/blob/master/basics/lists-and-tuples.md>

Dictionary

<https://github.com/Akuli/python-tutorial/blob/master/basics/dicts.md>

1. Conditional statements / functions

Conditional statements (if, else, etc.)

<https://github.com/Akuli/python-tutorial/blob/master/basics/lists-and-tuples.md>

Loop

<https://github.com/Akuli/python-tutorial/blob/master/basics/loops.md>

Functions

<https://github.com/Akuli/python-tutorial/blob/master/basics/defining-functions.md>

Modules

<https://github.com/Akuli/python-tutorial/blob/master/basics/modules.md>

Exceptions

<https://github.com/Akuli/python-tutorial/blob/master/basics/exceptions.md>

1. Classes / large programmes

Class

<https://github.com/Akuli/python-tutorial/blob/master/basics/classes.md>

Docstrings

<https://github.com/Akuli/python-tutorial/blob/master/basics/docstrings.md>

Write large programmes

<https://github.com/Akuli/python-tutorial/blob/master/basics/larger-program.md>

1. Python tutorial with YouTube video:

<https://github.com/vprusso/youtube_tutorials>