QF627 Programming and Computational Finance

HWS0305: Data Manipulation and Visualization

(part 2)

**In all the Python programs, we assume students will import Matplotlib, Pandas, Numpy and Scipy as the following:**

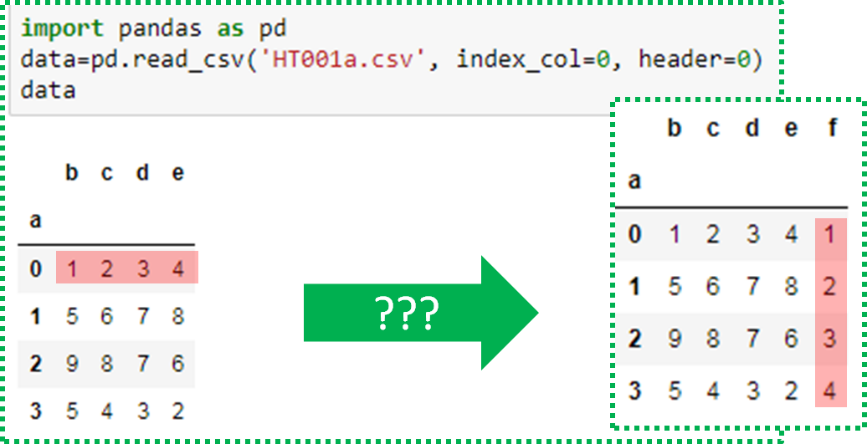
**import matplotlib.pyplot as plt**

**import pandas as pd**

**import numpy as np**

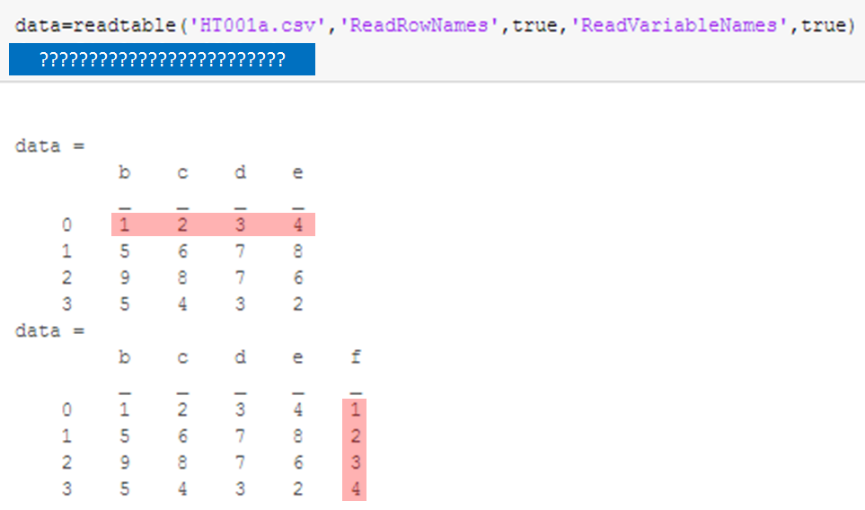
**import scipy.stats as ss**

Q1. (Python) Complete the following code **with one command** to add a column to **data** using the first row of **data**, and name this column **f**.



**data['f']= pd.DataFrame(data.values[0, :])**

Q2. (MATLAB) Complete the following code **with one command** to add a column to **data** using the first row of **data**, and name this column **f**.



**data.f=data{1,:}'**

Q3. (Python) Follow the instructions to complete the computation.

|  |
| --- |
| 1. **Use one command** with the Pandas library function **pandas.read\_csv** to load data from the CSV file, **dataset01.csv**, using the first row as column names. Name the data as **data**. |
| **data = pd.read\_csv('dataset01.csv', header=0)** |
| 2. Define a function, **option\_BS**, which computes and returns the European call option price using the following formula:  where  and |
| **def option\_BS(S,K,T,r,q,sigma):**  **d1=(np.log(S/K)+(r-q+sigma\*\*2/2)\*T)/(sigma\*np.sqrt(T))**  **d2 = d1 - sigma\*np.sqrt(T)**  **return S\*np.exp(-q\*T)\*ss.norm.cdf(**  **d1)-K\*np.exp(-r\*T)\*ss.norm.cdf(d2)** |
| 3. **Use one command** with the Pandas library function **pandas.DataFrame.apply** to compute the European call option price for each row of **data** and add the results to **data** as a new column, and name this column as **BS**. |
| **data['BS'] = data.apply(lambda x: option\_BS(x['S'],x['K'],x['T']**  **,x['r'],x['q'], x['sigma']), axis=1)** |

Q4. (MATLAB) Follow the instructions to complete the computation.

|  |
| --- |
| 1. **Use one command** to load data from the CSV file, **dataset01.csv**, using the first row as column names. Name the data as **data**. |
| **data=readtable('dataset01.csv','ReadVariableNames',true)** |
| 2. Define a function, **option\_BS**, which computes and returns the European call option price using the following formula:  where  and |
| **function [V]=option\_bs(S, K, r,q,sigma,T,c)**  **d1=(log(S/K)+(r-q+sigma^2)\*T)/(sigma\*sqrt(T));**  **d2=d1-sigma\*sqrt(T);**  **V=S\*exp(-q\*T)\*normcdf(d1)-K\*exp(-r\*T)\*normcdf(d2);**  **end** |
| 3. **Use one command** with library function **rowfun** to compute the European call option price for each row of **data** and add the results to **data** as a new column, and name this column as **BS**. |
| **BS=rowfun(@option\_bs,data)**  **data{:,'BS'}=BS{:,1}** |