Muhammad Shas K T
EMP ID: 281775

Assignment 1 - Data Analysis with NumPy (Arrays)

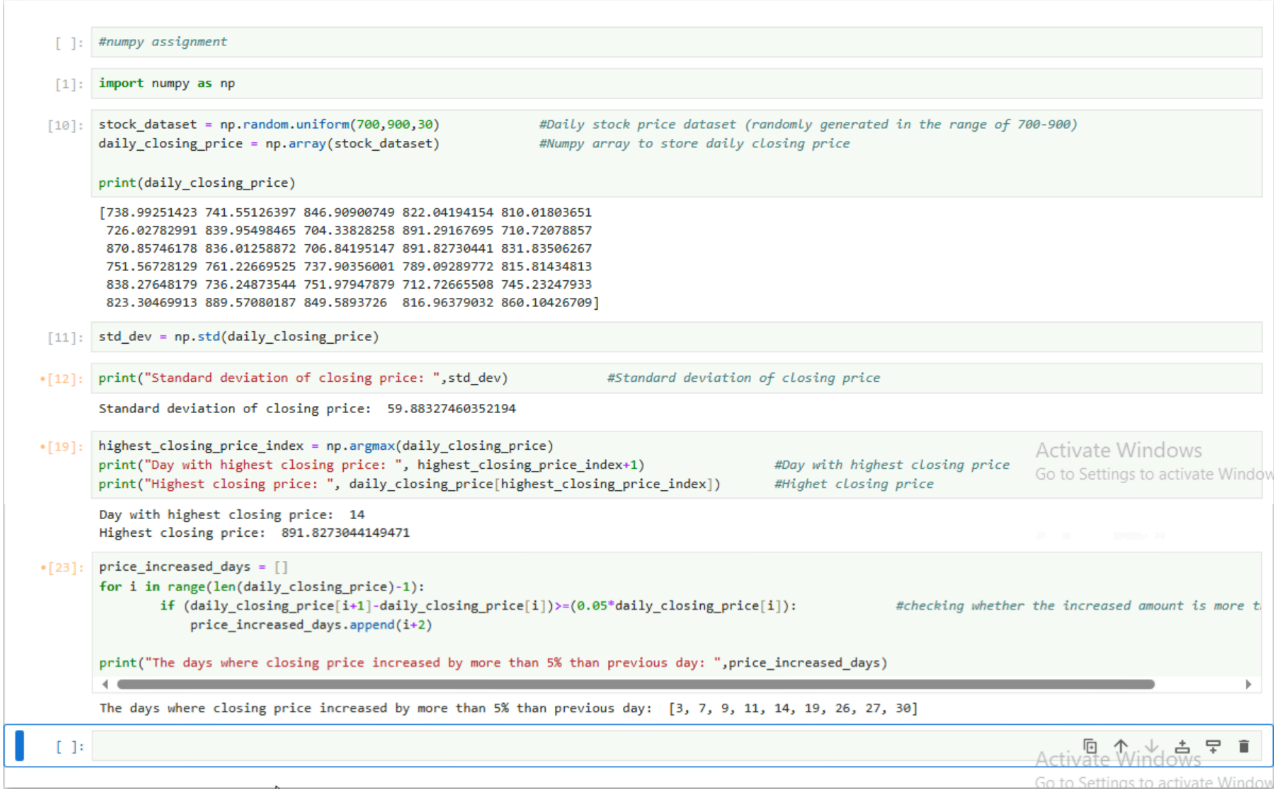
Task:-

- Simulate a dataset containing daily stock prices for a company over the past month.
- Each day's price will be a random number within a reasonable range.
- Use NumPy to create an array to store the daily closing prices.
- Analyze the stock price data using NumPy functions:
    - ✓ Calculate the standard deviation of the closing prices
    - ✓ Identify the day with the highest closing price and its value
    - ✓ Find the days where the closing price increased by more than 5% compared to the previous day
- Display the results in a clear and informative manner

Code:-

```
import numpy as np

#Daily stock price dataset (randomly generated in the range of 700-900)
stock_dataset = np.random.uniform(700,900,30)

#Numpy array to store daily closing price
daily_closing_price = np.array(stock_dataset)

print(daily_closing_price)

#Standard deviation of closing price
std_dev = np.std(daily_closing_price)
print(“Standard deviation of closing price: ”,std_dev)

#Day with highest closing price and highest closing price
highest_closing_price_index = np.argmax(daily_closing_price)
print(“Day with highest closing price: ”, highest_closing_price_index+1)
print(“Highest closing price: ”, daily_closing_price[highest_closing_price_index])

#Collecting days where more than 5% increase than previous day
price_increased_days = []
for i in range(len(daily_closing_price)-1):
    If (daily_closing_price[i+1]-daily_closing_price[i])>=(0.05*daily_closing_price[i]):
        price_increased_days.append(i+2)

print(“The days where closing price increased by more than 5%:“, price_increased_days)
```

Result:



```
[ ]: #numpy assignment

[1]: import numpy as np

[10]: stock_dataset = np.random.uniform(700,900,30)            #Daily stock price dataset (randomly generated in the range of 700-900)
      daily_closing_price = np.array(stock_dataset)           #Numpy array to store daily closing price

      print(daily_closing_price)

      [738.99251423 741.55126397 846.90900749 822.04194154 810.01803651
       726.02782991 839.95498465 704.33828258 891.29167695 710.72078857
       870.85746178 836.01258872 706.84195147 891.82730441 831.83506267
       751.56728129 761.22669525 737.90356001 789.09289772 815.81434813
       838.27648179 736.24873544 751.97947879 712.72665508 745.23247933
       823.30469913 889.57080187 849.5893726  816.96379032 860.10426709]

[11]: std_dev = np.std(daily_closing_price)

•[12]: print("Standard deviation of closing price: ",std_dev)            #Standard deviation of closing price

      Standard deviation of closing price:  59.88327460352194

•[19]: highest_closing_price_index = np.argmax(daily_closing_price)
      print("Day with highest closing price: ", highest_closing_price_index+1)            #Day with highest closing price
      print("Highest closing price: ", daily_closing_price[highest_closing_price_index])      #Highet closing price

      Day with highest closing price:  14
      Highest closing price:  891.8273044149471

•[23]: price_increased_days = []
      for i in range(len(daily_closing_price)-1):
              if (daily_closing_price[i+1]-daily_closing_price[i])>=(0.05*daily_closing_price[i]):            #checking whether the increased amount is more t
                  price_increased_days.append(i+2)

      print("The days where closing price increased by more than 5% than previous day: ",price_increased_days)

      The days where closing price increased by more than 5% than previous day:  [3, 7, 9, 11, 14, 19, 26, 27, 30]

[ ]:
```

Activate Windows
Go to Settings to activate Window