**Assignment – 25**

Q1. What is the distinction between a numpy array and a pandas data frame? Is there a way to convert between the two if there is?

Ans: A numpy array is a multidimensional array of homogeneous data types, commonly used for mathematical operations and numerical computations. In contrast, a pandas DataFrame is a two-dimensional labeled data structure with columns of potentially different types, resembling a table or spreadsheet. Numpy arrays are more suitable for numerical computations, while pandas DataFrames offer more functionality for data manipulation, analysis, and integration with tabular data.

Yes, there is a way to convert between the two. You can convert a numpy array to a pandas DataFrame using the pd.DataFrame() constructor, passing the numpy array as an argument. Conversely, you can convert a pandas DataFrame to a numpy array using the values attribute of the DataFrame.

Q2. What can go wrong when an user enters in a stock-ticker symbol, and how do you handle it?

Ans: When a user enters a stock-ticker symbol, potential issues include incorrect or non-existent symbols, missing data, or data format inconsistencies. To handle these issues, implement error handling mechanisms such as data validation to ensure the entered symbol is valid and exists in the database. Additionally, utilize exception handling to catch errors and provide informative messages to the user, guiding them on how to rectify the issue.

Q3. Identify some of the plotting techniques that are used to produce a stock-market chart.

Ans: Some plotting techniques used for producing stock-market charts include line plots, candlestick charts, bar charts, and scatter plots. Line plots are commonly used to visualize price trends over time, while candlestick charts provide information on price movements within a specified time frame. Bar charts are useful for comparing stock performance across different categories or time periods, and scatter plots can reveal relationships between different stocks or market indicators.

Q4. Why is it essential to print a legend on a stock market chart?

Ans: Printing a legend on a stock market chart is essential to provide context and interpretation for the plotted data. The legend identifies the different elements or series present in the chart, such as stock prices, moving averages, or trading volumes. By including a legend, viewers can easily understand the meaning of each element and make informed decisions based on the chart's information.

Q5. What is the best way to limit the length of a pandas data frame to less than a year?

Ans: The best way to limit the length of a pandas DataFrame to less than a year is to filter the DataFrame based on the date column. You can use datetime indexing or the pd.Timestamp() function to specify a date range and select only the rows within that range. Additionally, you can use the loc[] accessor or boolean indexing to filter rows based on specific conditions, such as dates falling within the desired time frame.

Q6. What is the definition of a 180-day moving average?

Ans: A 180-day moving average is a technical indicator used in financial analysis to smooth out short-term fluctuations and highlight long-term trends in a stock's price. It calculates the average closing price of a stock over the past 180 trading days (or other specified period) and updates the average as new data becomes available. By plotting the 180-day moving average alongside the stock's actual price, traders and analysts can identify potential trend reversals, support and resistance levels, and overall market sentiment.

Q7. Did the chapter's final example use "indirect" importing? If so, how exactly do you do it?

Ans: Yes, the chapter's final example likely used "indirect" importing by importing a module under a different name using the as keyword. For example, instead of importing a module directly with import module\_name, you can import it indirectly with import module\_name as alias. This allows you to refer to the module using the specified alias throughout your code. Indirect importing can help improve code readability and avoid naming conflicts, especially when working with multiple modules or libraries.