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?

The main distinction between a NumPy array and a Pandas DataFrame lies in their functionality and the types of data they can handle.

Structure

Indexing

Data Types

Missing Data Handling

Additional Functionality

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

When a user enters a stock ticker symbol, several issues can arise, and it's important to handle them properly. Here are some common problems and ways to address them.

Invalid Ticker Symbol: The user may enter an invalid or non-existent ticker symbol. To handle this, you can validate the ticker symbol against a reliable source, such as a stock exchange or financial data provider.

Case Sensitivity: Ticker symbols are sometimes case sensitive. For example, "AAPL" and "aapl" may represent different stocks

Data Availability: Stock data may not be available for certain ticker symbols or at specific times. It's important to handle situations where the requested data is not present or cannot be retrieved.

Data Quality and Accuracy: Stock data can occasionally have errors or inaccuracies. It's important to have proper data validation mechanisms in place to verify the integrity and accuracy of the retrieved data

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

Line Chart

Candlestick Chart

Moving Averages

Bollinger Bands

Volume Bars

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

Printing a legend on a stock market chart is essential because it provides key information about the elements represented in the chart. Here are a few reasons why including a legend is important.

Interpretation of Data

Contextual Understanding:

Visual Clarity

Documentation and Presentation

Accessibility

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

Filtering by Date Range

Resampling

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

A 180-day moving average is a statistical calculation used in financial analysis to smooth out price data and identify trends over a specific time period. It is a type of moving average that calculates the average of a stock or asset's price over the past 180 trading days

Take the closing prices of a stock or asset for each trading day over the past 180 trading days.

Sum up the closing prices for those 180 days.

Divide the sum by 180 to calculate the average

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

In the context of Python, "indirect" importing typically refers to importing a module indirectly through another module. It allows you to access the functionality of a module without explicitly importing it in your current module.

# module1.py

def function1():

print("This is function 1.")

# module2.py

import module1

def function2():

module1.function1()

# main.py

import module2

module2.function2()

This is a basic illustration of how "indirect" importing can be achieved. The actual implementation and structure of modules can vary based on the specific requirements of the code. "Indirect" importing can be useful in situations where you want to organize your code into multiple modules and control the access and visibility of certain functionalities across different modules.