**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?**

**ANSWER:** Numpy arrays are multidimensional arrays that can store homogeneous data (i.e., data of the same type). They provide fast numerical operations and support mathematical operations like linear algebra and statistical computations. Numpy arrays are efficient for performing mathematical operations on large datasets and are commonly used in scientific computing, machine learning, and data analysis.

On the other hand, Pandas DataFrames are tabular data structures that can store heterogeneous data (i.e., data of different types). They are built on top of Numpy arrays and provide additional functionalities like indexing, labeling, and handling missing data. Pandas DataFrames are designed to work with tabular or structured data, and they are commonly used for data manipulation, data cleaning, and data analysis.

To convert a Numpy array to a Pandas DataFrame, you can use the pd.DataFrame() function, which takes a Numpy array as input and creates a DataFrame with the same data. While to convert a Pandas DataFrame to a Numpy array, you can use the .values attribute of the DataFrame.

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

**ANSWER:** When a user enters a stock ticker symbol, there are several issues that can arise, including:

1. Invalid symbol: The user might enter a symbol that does not exist or is not recognized by the stock exchange. In this case, you should provide an error message to the user and prompt them to enter a valid symbol.
2. Multiple matches: Some stock exchanges may have multiple securities listed with the same ticker symbol. For example, different classes of shares, options, or warrants. In this case, you should provide a list of matching securities and ask the user to select the appropriate one.
3. Missing data: There may be missing or incomplete data for the specified symbol, which can result in errors or inaccurate information. You should have a strategy to handle missing data, such as using default values or estimates.
4. Outdated data: The stock data may not be up-to-date, which can lead to inaccuracies. You should ensure that the stock data is regularly updated and provide a warning to the user if the data is stale.

* To handle these issues, you can implement some error handling and validation techniques in your application. For example:

1. Check the validity of the symbol: Before processing the user input, you can use a stock exchange API or library to verify that the symbol is valid.
2. Handle multiple matches: If there are multiple securities with the same symbol, you can prompt the user to select the appropriate security using a dropdown menu or a list of choices.
3. Handle missing data: You can use default values or estimates when data is missing. For example, if the stock price is missing for a certain date, you can use the previous day's closing price as an estimate.

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

**ANSWER:** There are various plotting techniques that can be used to produce a stock market chart. Here are some of the commonly used techniques:

1. Line chart: This is the most basic type of stock market chart, where the closing prices are plotted as a line graph over a specified period of time. The x-axis represents the time period, and the y-axis represents the stock price.
2. Candlestick chart: This chart displays the high, low, open, and close prices of a security for a specific period of time. It uses a rectangle to represent the difference between the open and close price (the body of the candle) and a line to represent the range between the high and low price (the shadow).
3. Area chart: This chart shows the volume and price movements of a stock over time by filling in the area under the line chart. It is useful for visualizing trends and changes in the volume of trading.
4. Bar chart: This chart shows the opening, high, low, and closing prices of a security for a specified period of time using vertical bars. The top of the bar represents the highest price of the day, the bottom represents the lowest price, and the horizontal line represents the opening and closing prices.

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

**ANSWER:** Including a legend on a stock market chart is essential because it provides clarity, context, and consistency to the viewer. It helps the viewer understand the data being displayed and make informed decisions based on the information provided.

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

**ANSWER:** You can limit the length of a Pandas DataFrame to less than a year by filtering the data based on a date range using the loc[] method.

For example, to select data from January 1st, 2022 to December 31st, 2022, you can use:

df = df.loc['2022-01-01':'2022-12-31']

This will select all rows with dates within the specified range.

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

**ANSWER:** A 180-day moving average is a technical indicator that calculates the average price of a security over the past 180 trading days. It is also known as a 6-month moving average because it represents the average price over a period of six months.

To calculate the 180-day moving average, you take the sum of the closing prices of a security for the past 180 trading days and divide it by 180. This calculation is repeated every trading day, taking into account the closing price of the most recent trading day and dropping the closing price from 180 trading days ago.

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

**ANSWER:** Indirect importing is a way to import modules or functions without explicitly importing them in your code. Instead, you import a module that has imported the desired modules or functions and use them through that module.