**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?**

* A NumPy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. In contrast, a Pandas DataFrame is a 2-dimensional size-mutable, tabular data structure with rows and columns that can have different types.

Yes, one can convert a NumPy array to a Pandas DataFrame using the pandas.DataFrame() constructor, which takes the NumPy array as its argument.

Example :

import pandas as pd

import numpy as np

***# Create a NumPy array***

***arr = np.array([[1, 2, 3], [4, 5, 6]])***

***# Convert the array to a DataFrame***

***df = pd.DataFrame(arr)***

***# Print the resulting DataFrame***

***print(df)***

* You can also convert a Pandas DataFrame to a NumPy array using the DataFrame.to\_numpy() method. This method returns a NumPy array representation of the DataFrame.

For example:

***import pandas as pd***

***import numpy as np***

***# Create a DataFrame***

***df = pd.DataFrame([[1, 2, 3], [4, 5, 6]])***

***# Convert the DataFrame to a NumPy array***

***arr = df.to\_numpy()***

***# Print the resulting array***

***print(arr)***

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

There are several things that can go wrong when a user enters a stock ticker symbol. For example:

The user may enter an invalid or unrecognized ticker symbol. In this case, it is important to handle the error gracefully and inform the user that the ticker symbol they entered is not valid.

The user may enter a valid ticker symbol, but the data for that particular stock may not be available. This could be due to a temporary outage or other issue with the data source. In this case, it is important to handle the error gracefully and inform the user that the data is not currently available.

The user may enter a ticker symbol that is valid, but the data for that particular stock may be out of date or otherwise inaccurate. In this case, it is important to inform the user that the data may not be current and should be used with caution.

To handle these situations, it is important to implement error handling and validation logic in your code. For example, you could use regular expressions to validate the format of the ticker symbol and check if it matches the expected pattern for a stock ticker symbol. You could also use try-except blocks to handle any errors that may occur when trying to retrieve data for the ticker symbol from a data source. Additionally, you could use the datetime module to check if the data for the ticker symbol is up to date and within a certain time frame.

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

There are several plotting techniques that can be used to produce a stock-market chart in python. Some common ones include:

* **Line plots:** These are used to show the changes in a stock's price over time. A line is drawn between each data point, connecting the stock's price at different points in time. This makes it easy to see trends and patterns in the stock's price.
* **Scatter plots:** These are similar to line plots, but instead of connecting the data points with lines, each data point is plotted individually. This can be useful for identifying outliers or unusual patterns in the data.
* **Bar plots:** These are used to show the difference between the opening and closing prices of a stock over time. A vertical bar is drawn for each data point, with the length of the bar representing the difference between the opening and closing prices. This can be useful for quickly visualizing the overall performance of a stock.
* **Candlestick plots:** These are a type of bar plot that show the range of a stock's prices over a given time period. Each data point is represented by a "candlestick" with a body representing the range of prices and "wicks" representing the highest and lowest prices. Candlestick plots can be useful for identifying trends and patterns in a stock's price.
* **OHLC plots:** These are similar to candlestick plots, but instead of showing the range of prices with a body, they show the opening, high, low, and closing prices with separate bars. This can be useful for seeing the full range of a stock's prices over time.

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

It is essential to print a legend on a stock market chart because it helps to identify and label the different data series that are being plotted. A legend is a key or guide that explains the meaning of the symbols, colors, or patterns used in the chart. Without a legend, it can be difficult or impossible for viewers to understand the information being presented in the chart.

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

One way to limit the length of a pandas data frame to less than a year would be to use the pandas.DataFrame.tail() method. This method returns the last n rows of the data frame, where n is the number of rows you want to include in the resulting data frame. For example, if you want to limit the data frame to the last 365 rows (i.e., less than a year's worth of data), you could use the following code:

***# create a new data frame with only the last 365 rows of the original data frame***

***df\_limited = df.tail(365)***

Alternatively, you could use the **pandas.DataFrame.truncate()** method, which allows you to specify a specific range of rows to include in the resulting data frame. For example, if your original data frame has a column called date that contains timestamps for each row, you could use the following code to keep only the rows from the last year:

***# create a new data frame with only the rows from the last year***

***start\_date = pd.to\_datetime('1 year ago')***

***end\_date = pd.to\_datetime('today')***

***df\_limited = df.truncate(before=start\_date, after=end\_date)***

Both of these methods allow you to easily limit the length of a pandas data frame to less than a year.

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

A 180-day moving average is a type of technical analysis indicator that shows the average value of a stock's price over the past 180 days. To calculate the 180-day moving average, you take the sum of the stock's closing prices for the past 180 days, and then divide that sum by 180. The result is the average price of the stock over the past 180 days. This value is then plotted on a stock chart, along with the stock's actual price.

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