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?

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

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

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

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

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

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

Q1. The main distinction between a NumPy array and a Pandas DataFrame is that a NumPy array represents a homogeneous array of elements with a fixed size, whereas a Pandas DataFrame represents a heterogeneous tabular data structure with labeled axes. A NumPy array is typically used for numerical operations, while a Pandas DataFrame is used for data analysis and manipulation.

Yes, it is possible to convert between the two data structures. To convert a NumPy array to a Pandas DataFrame, you can use the `pd.DataFrame()` function, which takes a NumPy array as input. To convert a Pandas DataFrame to a NumPy array, you can use the `.to_numpy()` method.

Q2. When a user enters a stock-ticker symbol, several things can go wrong, including:

- The user may enter an invalid symbol or misspell the symbol.
- The symbol may refer to a company that is not publicly traded.
- The symbol may be associated with a different exchange than the one the user is interested in.
- The symbol may be associated with a different security type than the one the user is interested in.

To handle these issues, you can implement error handling techniques such as using try-except blocks to catch errors and prompt the user to enter a valid symbol. You can also provide a dropdown menu or a search bar to assist the user in selecting the correct symbol.

Q3. Some of the plotting techniques used to produce a stock-market chart include:

- Line charts: to plot the daily closing prices of a stock over time.
- Candlestick charts: to show the daily high, low, open, and close prices of a stock over time.
- Bar charts: to show the daily volume of trading activity for a stock over time.

- Moving averages: to show the trend of a stock's price over time.

Q4. It is essential to print a legend on a stock market chart because it provides context for the data being presented. The legend explains the meaning of the different colors, lines, and markers used in the chart, making it easier for the viewer to interpret the data accurately.

Q5. To limit the length of a Pandas DataFrame to less than a year, you can use the `.loc[]` method to select the rows that fall within a specific date range. For example, to select all rows that fall between January 1, 2022, and December 31, 2022, you can use the following code:

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

Q6. A 180-day moving average is a technical analysis indicator that calculates the average price of a stock over the past 180 days. It is a lagging indicator that smooths out short-term fluctuations in the stock's price and helps to identify longer-term trends.

Q7. The answer to this question depends on the specific chapter and example being referred to. In general, indirect importing refers to the practice of importing a module or function indirectly by importing another module that depends on it. For example, if module A depends on module B, you can indirectly import module B by importing module A. To do this, you would include the following line in your code:

python
```
from module_a import *
```

This would import all functions and classes from module A, as well as any modules that module A depends on.