ASSIGNMENT 25

# Q1

The main distinction between a NumPy array and a pandas DataFrame is that a NumPy array is a multidimensional homogeneous data structure, while a pandas DataFrame is a two-dimensional labeled data structure with potentially heterogeneous data types for each column. NumPy arrays provide efficient storage and operations for numerical data, while pandas DataFrames offer additional functionalities for data manipulation, analysis, and integration with other data sources.

Yes, it is possible to convert between a NumPy array and a pandas DataFrame. We can create a DataFrame from a NumPy array by using the pd.DataFrame() constructor, passing the NumPy array as the data parameter. Conversely, we can convert a pandas DataFrame to a NumPy array using the .values attribute of the DataFrame.

# Q2

Invalid symbol: The user may enter a symbol that does not exist in the stock market database.

Case sensitivity: The symbol may be case sensitive, and the user's input may not match the expected format.

Data availability: The symbol entered may not have data available or may not be traded on the chosen stock market.

Input validation: The user's input may contain special characters, spaces, or invalid characters that need to be handled appropriately.

To handle these issues, it is important to implement proper input validation and error handling. This can include checking the validity of the symbol, handling exceptions or errors if the symbol is not found or if data retrieval fails, providing informative error messages to the user, and guiding them to enter valid symbols.

# Q3

Line plots: Used to display the historical price data of a stock over time.

Candlestick charts: Represent the open, high, low, and close prices of a stock for each trading period, providing a visual representation of price movements.

Moving averages: Plots the average price of a stock over a specific period, highlighting trends and smoothing out short-term fluctuations.

# Q4

It is essential to print a legend on a stock market chart because it provides a key to understanding the various elements displayed in the chart. The legend typically explains the meaning of different colors, lines, or markers used in the chart, such as the price line, moving average lines, volume bars, or other technical indicators. By including a legend, viewers can easily interpret the information presented in the chart without ambiguity.

# Q5

To limit the length of a pandas DataFrame to less than a year, we can filter the DataFrame based on the date column using datetime operations.

import pandas as pd

# Assuming 'df' is the DataFrame with a 'date' column

df['date'] = pd.to\_datetime(df['date']) # Convert 'date' column to datetime if not already in datetime format

start\_date = pd.to\_datetime('2022-01-01') # Specify the start date

end\_date = start\_date + pd.DateOffset(years=1) # Calculate the end date as one year from the start date

filtered\_df = df[(df['date'] >= start\_date) & (df['date'] < end\_date)]

# Q6

A 180-day moving average is a technical analysis indicator that calculates the average price of a stock over the previous 180 trading days. It provides a smoothed line that helps identify the long-term trend by reducing short-term price fluctuations. Each data point in the moving average is calculated by taking the average of the stock's closing prices over the preceding 180 days.

# Q7

Yes, the chapter's final example did use "indirect" importing. In Python, we can import modules indirectly by using the importlib.import\_module() function from the importlib module. This allows us to import a module using a string representing its name.

import importlib

module\_name = 'example\_module' # The name of the module we want to import

module = importlib.import\_module(module\_name)