Firstly, the code defines a function 'extracting_data' to extract historical stock data using the 'yfinance' library. It handles any exceptions that may occur during data extraction and returns the extracted data.

The code then defines the symbol, start date, and end date for the data extraction. It calls the 'extracting_data' function to retrieve the stock data and stores it in the 'stock_data' variable. The first 5 rows of the extracted data are printed.

Next, the extracted data is converted into a pandas DataFrame named 'df'. The code checks the shape of the DataFrame and examines the columns. It also checks for any missing values using the 'isna().sum()' function and provides descriptive statistics using the 'describe()' function.

The code defines a function 'average_volume' to calculate the average daily trading volume using pandas. It calculates the mean of the 'Volume' column in the stock data. The function handles any exceptions that may occur during the calculation.

The code then calls the 'average_volume' function and prints the average daily trading volume for the selected period.

For data visualization, the code creates a candlestick chart using matplotlib. It separates the data into bullish and bearish candlesticks based on the closing and opening prices. The bullish candlesticks are plotted as green bars, indicating prices ready to move high, while the bearish candlesticks are plotted as red bars, indicating prices ready to move low.

The chart is customized with labels, titles, legends, and gridlines. The x-axis ticks are rotated for better readability. Finally, the chart is displayed using 'plt.show()'.