

"Robo Advisor" Further Exploration Challenges

Functionality Challenges

Handling Multiple Stocks

Instead of processing a single stock at a time, allow the user to process multiple stocks at the same time.

If you decide to implement this approach, pay attention to the "Information Output Requirements" about using multiple CSV files instead of a single file.

Calculating 52-Week Highs and Lows

Instead of calculating and printing the stock's **recent average high** and **recent average low**, calculate and print the stock's **52-week high** and **52-week low**, respectively.

The **52-week high** should be equal to the maximum daily "high" price over approximately the past year of trading data. For example, if the last available day of trading data is June 1st, 2018, the program should find the maximum of all the "high" prices between around June 1st, 2017 and June 1st, 2018.

The **52-week low** should be calculated in a similar manner as the 52-week high, but it should instead be equal to the minimum of all "low" prices within the past year.

HINT: If you were previously requesting daily data, you will have to change your approach to either request more of it (i.e. by specifying the URL parameter `&outputsize=full`), or to instead request weekly data. Theoretically the maximum weekly "high" price should be the same as the maximum daily "high" price within the same period.

NOTE: If you do request "full" responses of daily data, the size of the response may drastically increase, and the speed of your requests may slow down noticeably. If you think these changes negatively impact user experience, you might want to consider requesting weekly data instead.

Plotting Prices over Time

Prerequisite: The ["Executive Dashboard" Exercise](#)

In addition to writing the historical stock prices to a CSV file, your program should also display a line graph of the stock prices over time. 🐱

Optionally include any other data visualizations as desired, to add context to and increase confidence in the final recommendations.

Sending Alerts via Email

Prerequisite: The email-sending component of the Notification Service app, referenced in the ["Interface Capabilities" Exercise](#)

Modify the logic of your application such that if it detects the stock's price has moved past a given threshold within a given time period (e.g. the price has increased or decreased by more than 5% within the past day), it will send the user a "Price Movement Alert" message via email.

HINT: leverage the email-sending capabilities of [the sendgrid package](#), and optionally use [Sendgrid email templates](#) to further control the formatting of email contents

Sending Alerts via SMS

Prerequisite: The SMS-sending component of the Notification Service app, referenced in the ["Interface Capabilities" Exercise](#)

Modify the logic of your application such that if it detects the stock's price has moved past a given threshold within a given time period (e.g. the price has increased or decreased by more than 5% within the past day), it will send the user a "Price Movement Alert" message via SMS.

HINT: leverage the SMS-sending capabilities of [the twilio package](#)