

This manual testing plan will cover all plots created by the application. Note that below steps can be done for any stock, but we will use VOD (Vodafone) for this example.

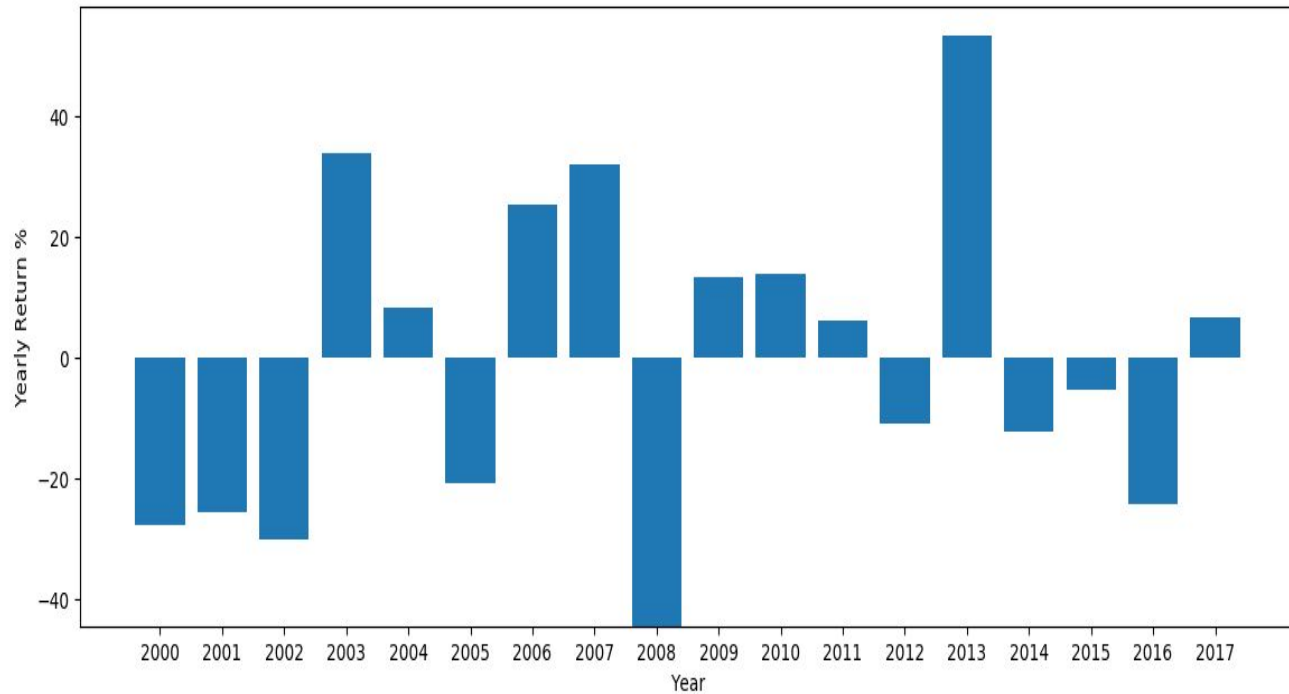
- 1.) Run application with command “python main.py” in project directory
- 2.) Open Postmaster (alternatively can make requests using ‘curl’ on unix shell)
- 3.) Make a GET request to: [http://127.0.0.1:5000/api/stocks/VOD/price\\_plot](http://127.0.0.1:5000/api/stocks/VOD/price_plot)
- Visit the Plotly URL returned by request. Make the following checks:
  - X-axis is labeled “Year”
  - Y-axis is labeled “Share Price”
  - Labels on x-axis should range from 2000 to 2018
  - Data should end slightly after 2017 ticker, since 2017 in progress



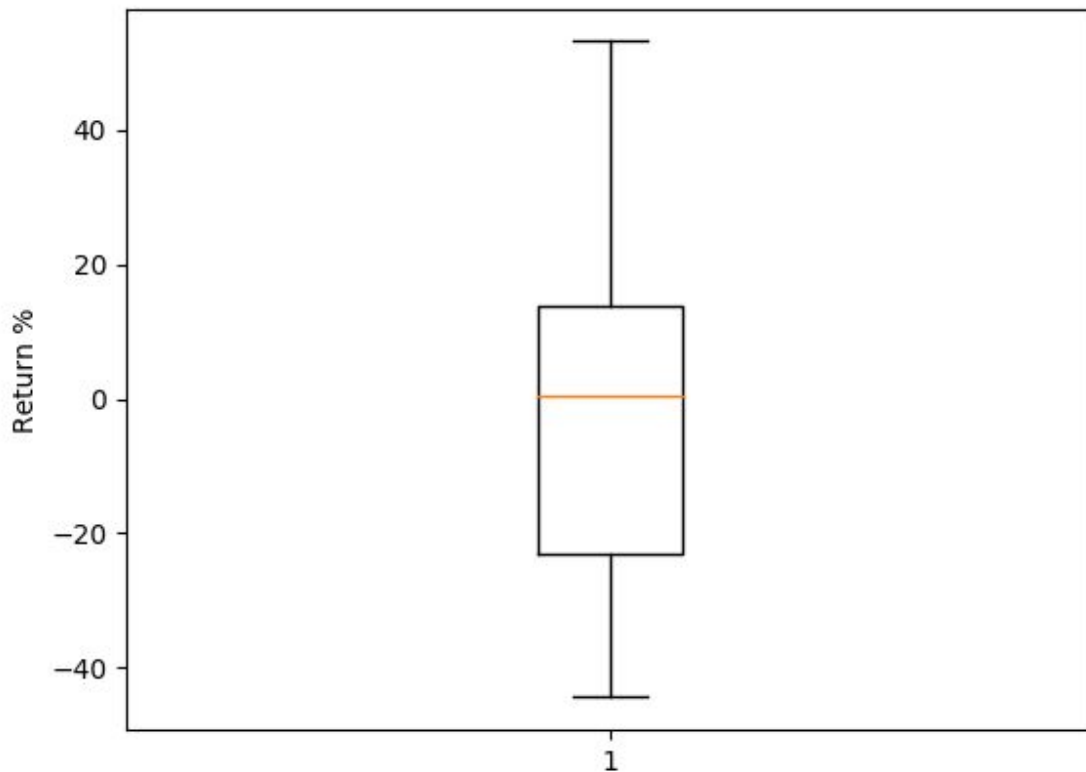
4.) Make a GET request to [http://127.0.0.1:5000/api/stocks/VOD/yearly\\_returns\\_plot](http://127.0.0.1:5000/api/stocks/VOD/yearly_returns_plot)

- Visit the Plotly URL returned by request. Make the following checks:
  - X-axis is labeled “Year”
  - Y-axis is labeled “Yearly return %”
  - Labels on x-axis should range from 200 to 2017

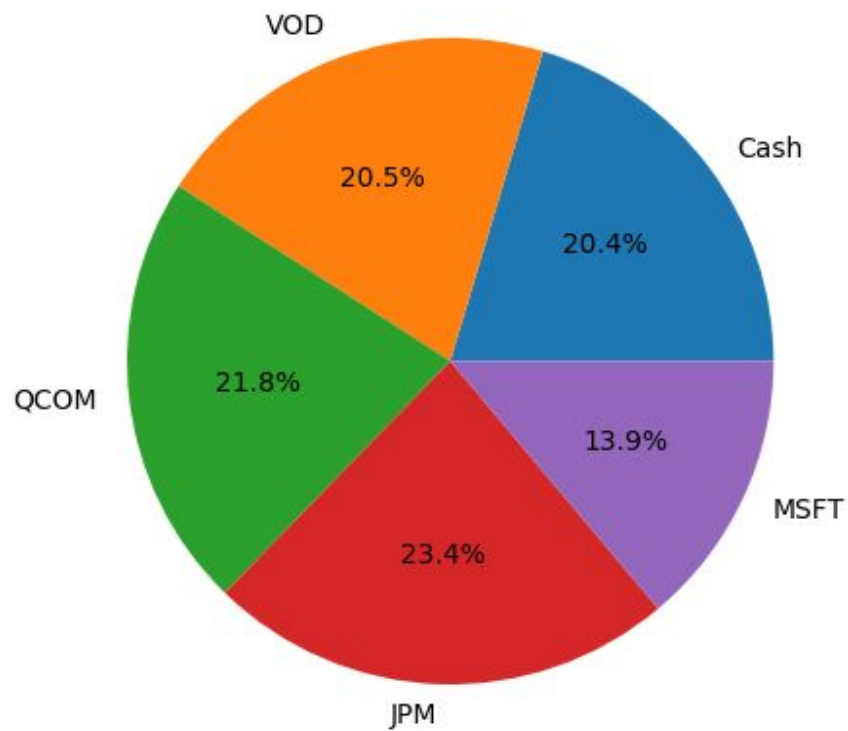
- All boxes should originate along line  $y=0$ . Each box should be either entirely below or above this line



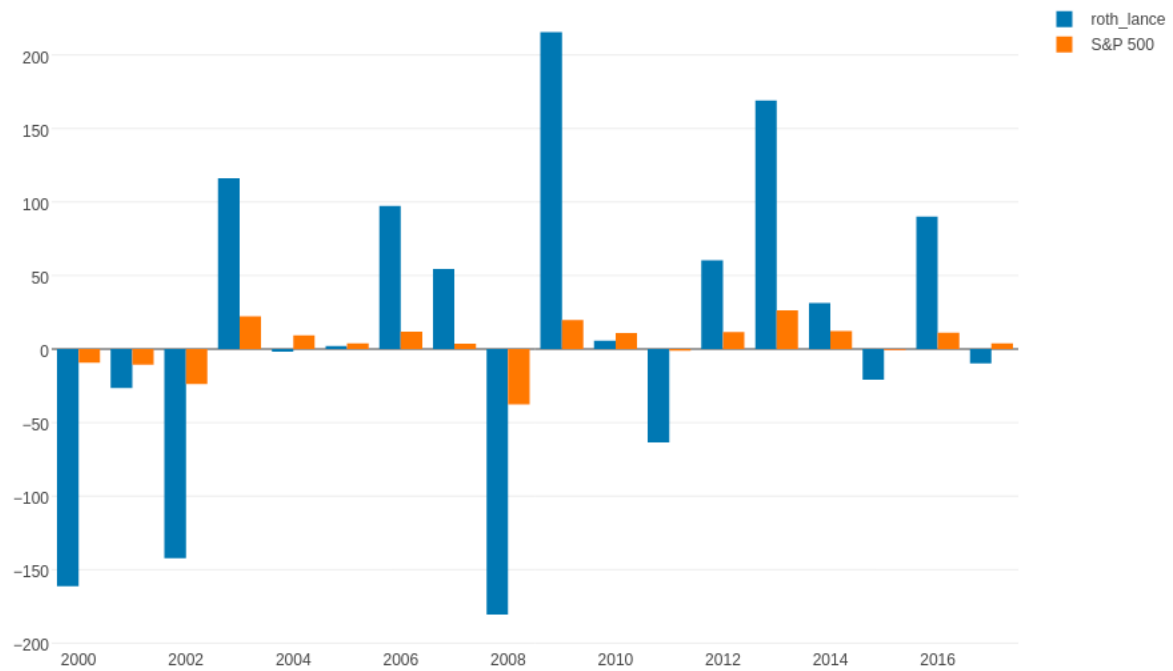
- 5.) Make a GET request to [http://127.0.0.1:5000/api/stocks/VOD/returns\\_boxplot](http://127.0.0.1:5000/api/stocks/VOD/returns_boxplot)
- Visit the Plotly URL returned by request. Make the following checks:
    - Y-Axis should be labeled (Return %)
    - X-Axis should be unlabeled
    - Plot should be a valid Boxplot. That is, it should have a ticker for max and min at the top and bottom respectively. An orange line indicating the median, and two edges of the rectangle indicating the 75th and 25th percentile.



- 6.) Make a GET request to [http://127.0.0.1:5000/api/portfolio/roth\\_lance/breakdown\\_plot](http://127.0.0.1:5000/api/portfolio/roth_lance/breakdown_plot)
- Visit the Plotly URL returned by request. Make the following checks:
    - Each section of pie chart should be a different color
    - Each section of pie chart should have a label on the outer edge (e.g. JPM)
    - Percentage should be displayed within each section
      - These percentages should sum to 100%
      - Percentage should correspond to area of plot that section takes up
      - Sections with larger percentages should visibility take up more space (see above)



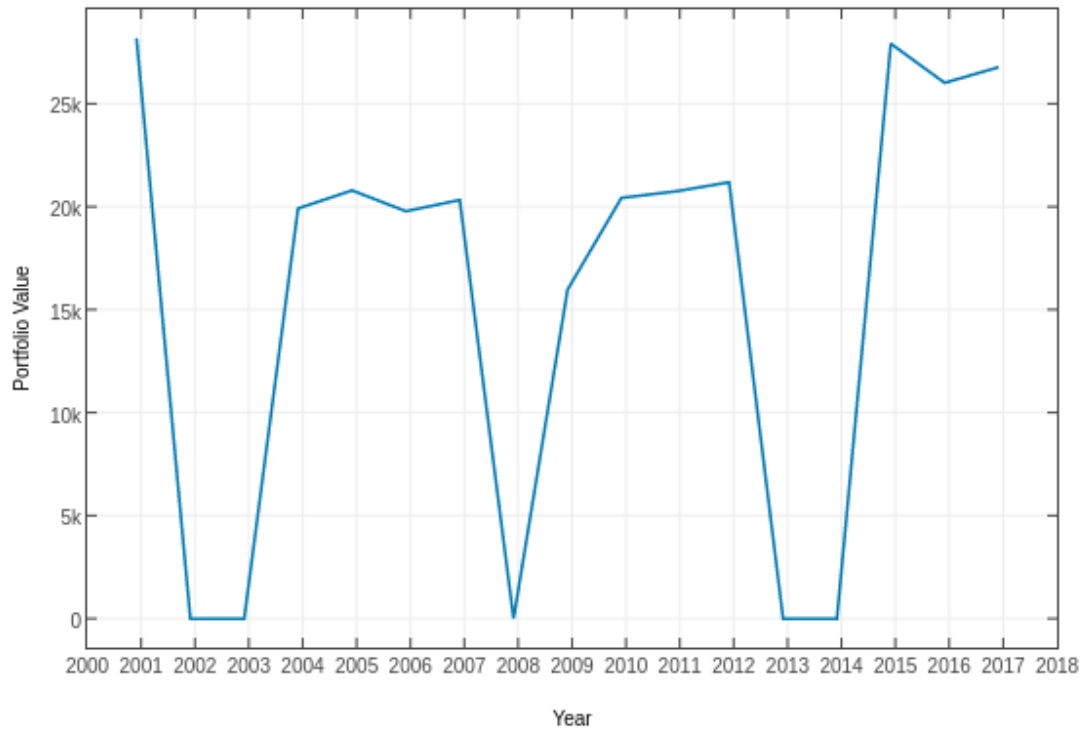
- 7.) Make a GET request to [http://127.0.0.1:5000/api/portfolio/roth\\_lance/returns\\_plot](http://127.0.0.1:5000/api/portfolio/roth_lance/returns_plot)
- Visit the Plotly URL returned by request. Make the following checks:
    - One bar graph should be colored blue, other orange
    - One bar should be labeled "roth\_lance" the other "S&P500"
    - Percent returns should be the y-axis
    - Years should be the x-axis, should see 2000 to 2018



8.) Make a GET request to [http://127.0.0.1:5000/api/portfolio/roth\\_lance/value\\_plot](http://127.0.0.1:5000/api/portfolio/roth_lance/value_plot)

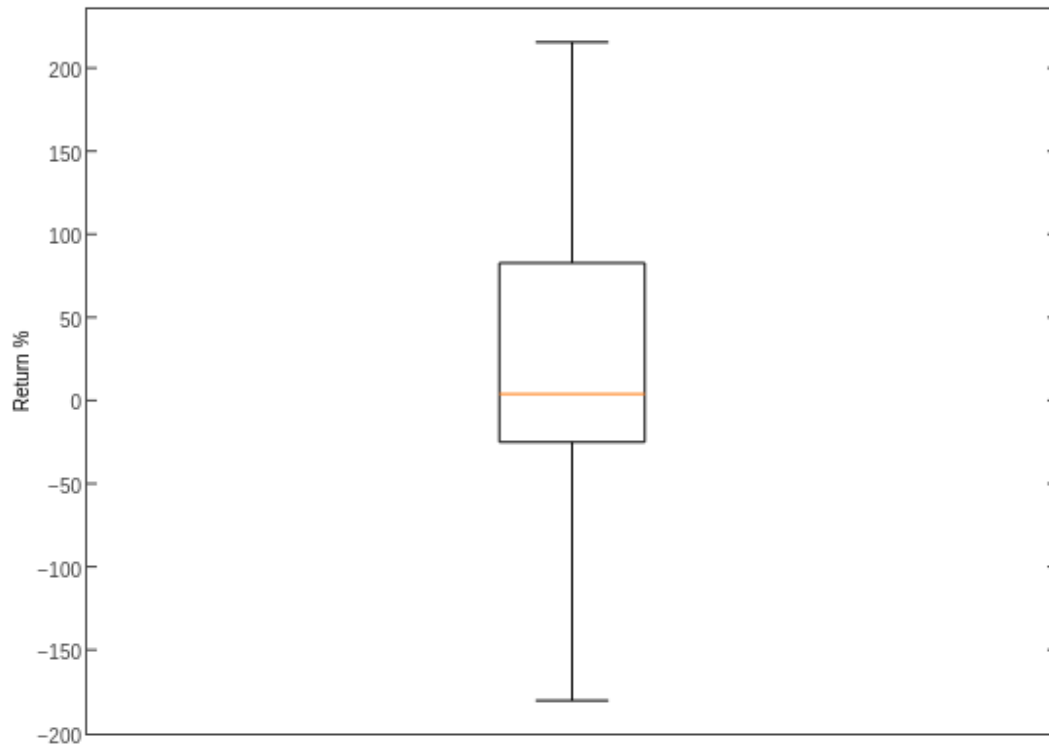
- Visit the Plotly URL returned by request. Make the following checks:

- One bar graph should be colored blue, other orange
- One bar should be labeled "roth\_lance" the other "S&P500"
- Portfolio value should be the y-axis
- Years should be the x-axis, should see 2000 to 2018

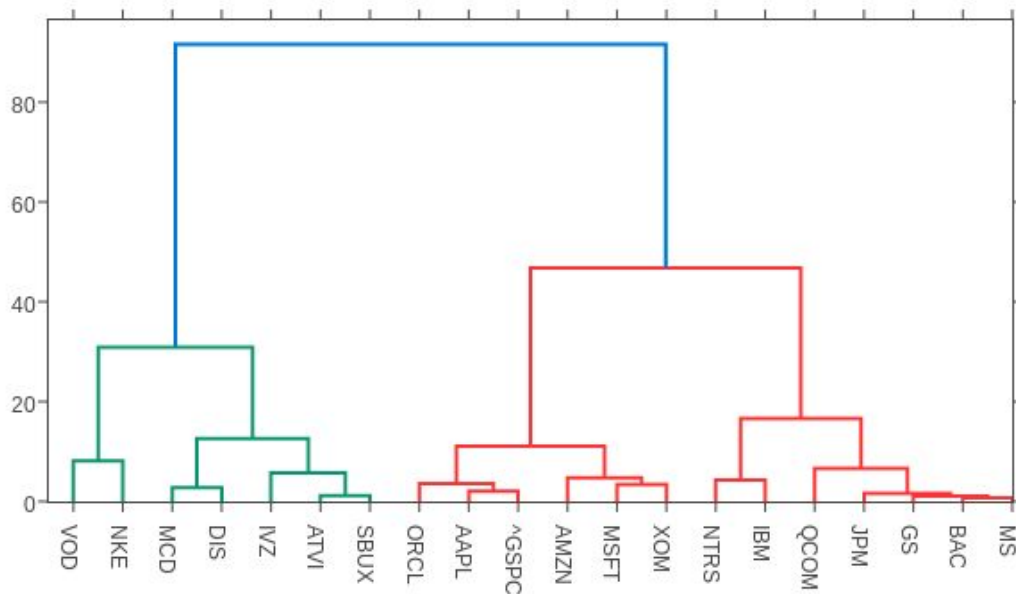


9.) Make a GET request to [http://127.0.0.1:5000/api/portfolios/roth\\_lance/boxplot](http://127.0.0.1:5000/api/portfolios/roth_lance/boxplot)

- Visit the Plotly URL returned by request. Make the following checks:
  - Y-Axis should be labeled (Return %)
  - X-Axis should be unlabeled
  - Plot should be a valid Boxplot. That is, it should have a ticker for max and min at the top and bottom respectively. An orange line indicating the median, and two edges of the rectangle indicating the 75th and 25th percentile.



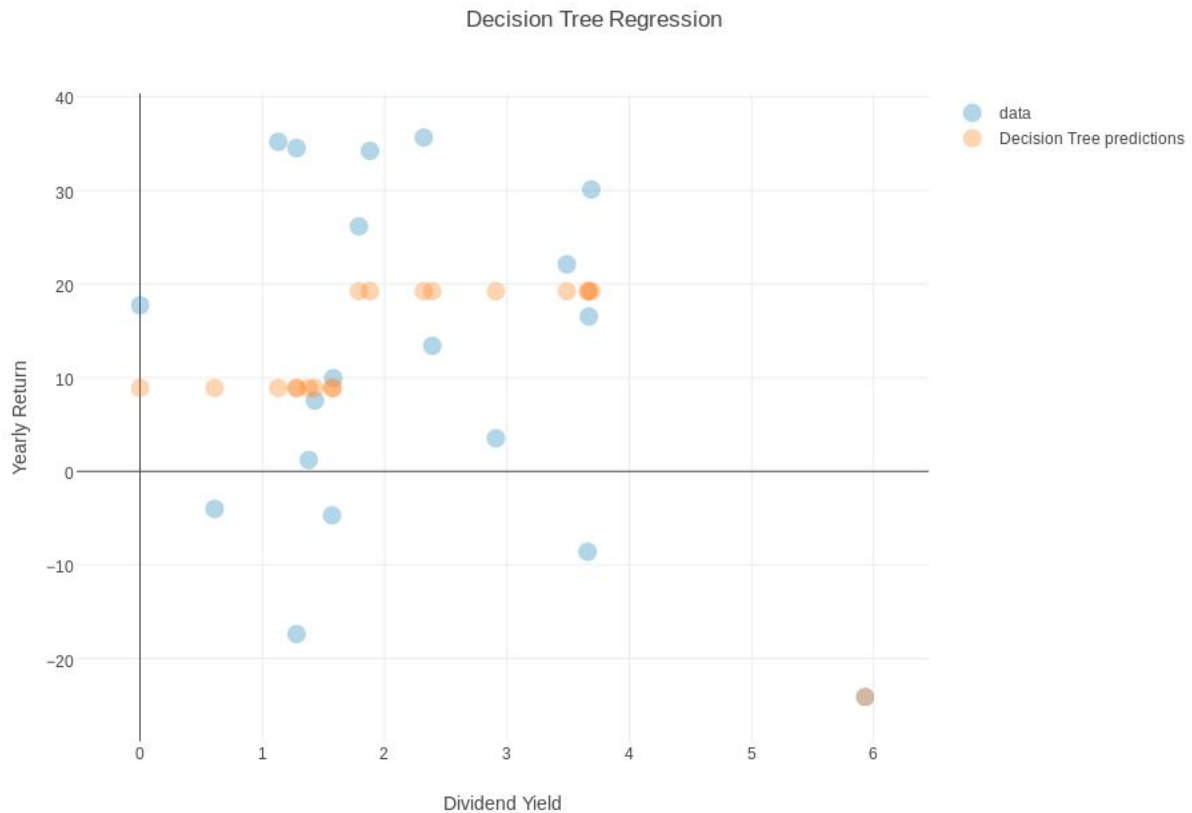
- 10.) Make a GET request to [http://127.0.0.1:5000/api/stocks/hierarchical\\_clustering/plot](http://127.0.0.1:5000/api/stocks/hierarchical_clustering/plot)
- Visit the Plotly URL returned by request. Make the following checks:
    - Plot should be a valid hierarchical clustering, specifically:
    - Distance (as measured on y-axis) should be larger for clusters lower in the plot (closer to the x-axis)
    - Each cluster should be labeled with a different color, with the top level cluster having a completely unique color



11.) Make a GET request to [http://127.0.0.1:5000/api/stocks/decision\\_tree](http://127.0.0.1:5000/api/stocks/decision_tree)

- Visit the Plotly URL returned by request. Make the following checks:
  - Plot should be titled Decision Tree Regression
  - Y-axis should be labeled Yearly Return
  - X-axis should be labeled Dividend Yield
  - Blue points should be labeled data in legend
  - Orange points should be labeled Decision Tree predictions in legend
  - Both orange and blue points should appear on plot





12.) Make a GET request to <http://127.0.0.1:5000/api/stocks/svm>

- Visit the Plotly URL returned by request. Make the following checks:
  - Plot should be titled Support Vector Machine Regression
  - Y-axis should be labeled Yearly Return
  - X-axis should be labeled Dividend Yield
  - Blue points should be labeled data in legend
  - Orange points should be labeled svm in legend
  - Both orange and blue points should appear on plot
  - An orange line should appear, but not a blue line

Support Vector Machine Regression

