
```

% Stock Analysis and Visualization Script
% This script loads and analyzes processed stock data for multiple stock
% symbols. It generates heatmaps to visualize the daily returns of each
% stock symbol, overlays volatility information, and includes 3D plots and
% interactive plots for further analysis. The script utilizes CSV files
% containing processed data for each stock symbol and provides colorful and
% interactive visualizations to aid in understanding and exploring the
% stock performance.

% Instructions:
% 1. Ensure the processed data CSV files for each stock symbol are present
% in the same directory as this script.
% 2. Modify the 'symbols' variable to include the desired stock symbols.
% 3. Run the script to generate heatmaps, 3D plots, and interactive plots
% for the stock symbols.

% Note: The script assumes the second-to-last column in each processed data
% file contains the daily returns.

% Author: Abraham reines
% Date: May 23, 2023

% Load the processed data for each stock symbol
symbols = {'AAPL', 'GOOGL', 'MSFT'};
for i = 1:numel(symbols)
    symbol = symbols{i};
    filename = sprintf('%s_processed_data.csv', symbol);
    data = csvread(filename, 1, 0); % Skip the header row

    % Extract the relevant column for daily returns
    dailyReturns = data(:, end-1); % Assuming the second-to-last column
    contains the daily returns

    % Determine the number of data points
    numDataPoints = size(dailyReturns, 1);

    % Determine the number of trading days
    numTradingDays = ceil(numDataPoints / 5);

    % Determine the number of data points per day
    numDataPointsPerDay = ceil(numDataPoints / numTradingDays);

    % Calculate the number of missing data points
    missingDataPoints = numTradingDays * numDataPointsPerDay - numDataPoints;

    % Add NaN values for missing data points
    dailyReturns = [dailyReturns; NaN(missingDataPoints, 1)];

    % Create a matrix for daily returns
    dailyReturnsMatrix = reshape(dailyReturns, numDataPointsPerDay,
    numTradingDays); % Reshape to a matrix

```

```

% Calculate volatility (standard deviation)
volatility = std(dailyReturnsMatrix);

% Data visualization - Heatmap of daily returns with volatility overlay
figure;
imagesc(dailyReturnsMatrix);
colormap(jet); % Use 'jet' colormap for colorful representation
colorbar;
hold on;
plot(1:numTradingDays, volatility, 'w-', 'LineWidth', 2); % Overlay
volatility
hold off;
title(sprintf('%s Stock Daily Returns Heatmap with Volatility', symbol));
xlabel('Trading Days');
ylabel('Data Points');
legend('Volatility');

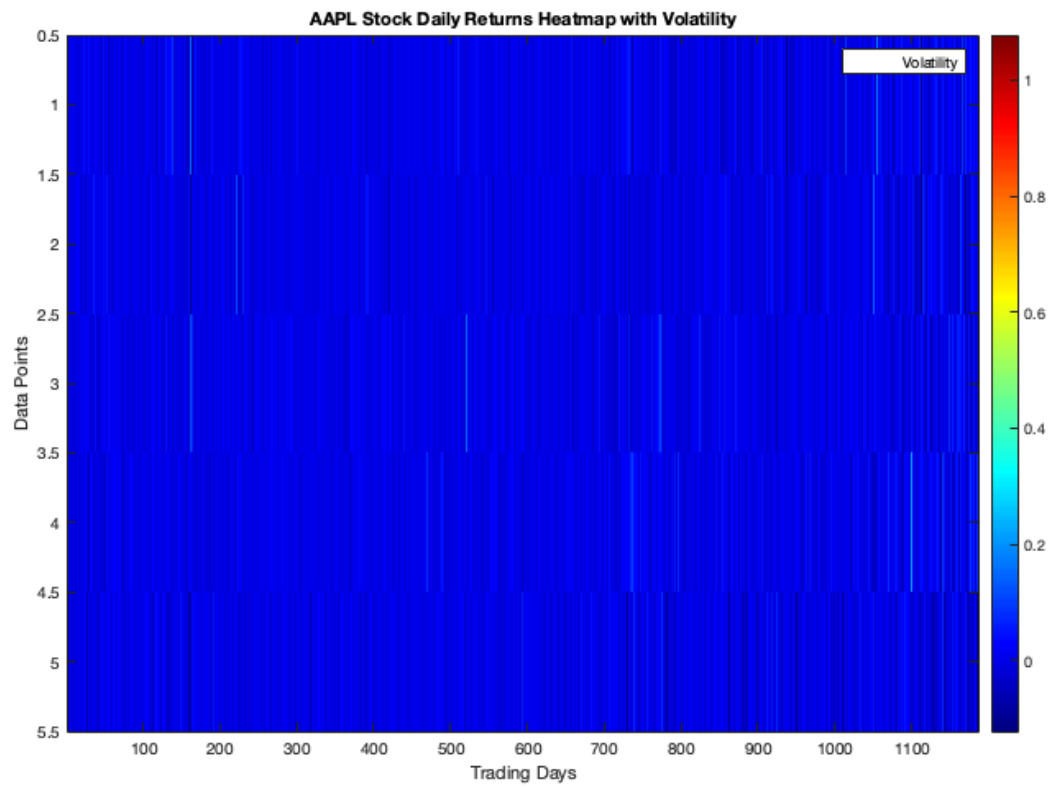
% Adjust the figure size for better display
fig = gcf;
fig.Position(3) = fig.Position(3) + 200;
fig.Position(4) = fig.Position(4) + 100;

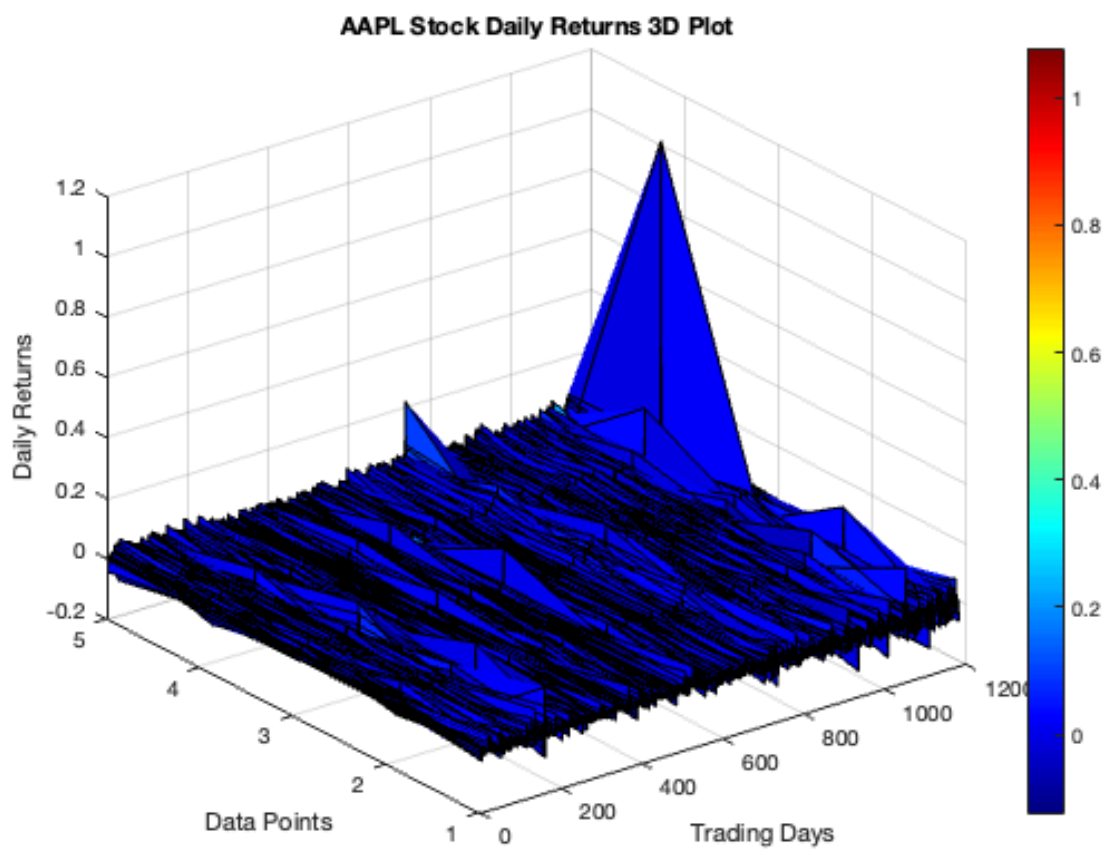
% Data visualization - 3D plot of daily returns
figure;
surf(dailyReturnsMatrix);
colormap(jet); % Use 'jet' colormap for colorful representation
title(sprintf('%s Stock Daily Returns 3D Plot', symbol));
xlabel('Trading Days');
ylabel('Data Points');
zlabel('Daily Returns');
colorbar;

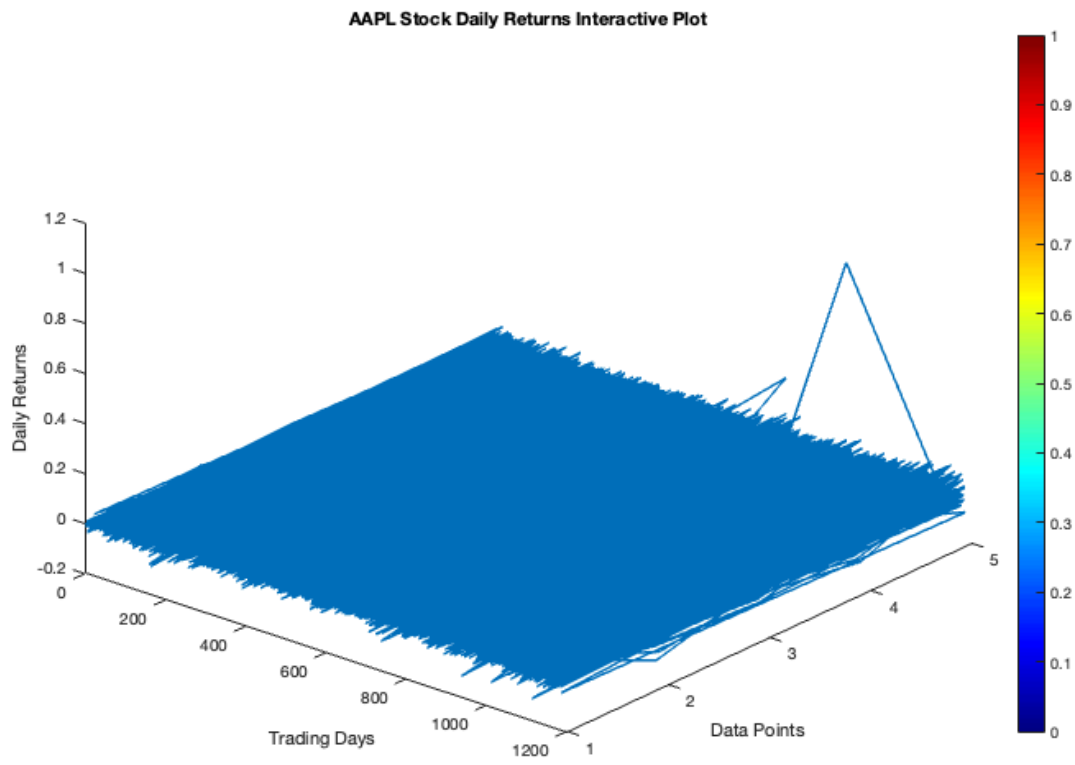
% Data visualization - Interactive plot of daily returns
figure;
[X, Y] = meshgrid(1:numTradingDays, 1:numDataPointsPerDay);
plot3(X(:), Y(:), dailyReturnsMatrix(:), 'LineWidth', 1.5);
colormap(jet); % Use 'jet' colormap for colorful representation
title(sprintf('%s Stock Daily Returns Interactive Plot', symbol));
xlabel('Trading Days');
ylabel('Data Points');
zlabel('Daily Returns');
colorbar;
view(40, 35); % Set the initial viewing angle

% Adjust the figure size for better display
fig = gcf;
fig.Position(3) = fig.Position(3) + 200;
fig.Position(4) = fig.Position(4) + 100;
end

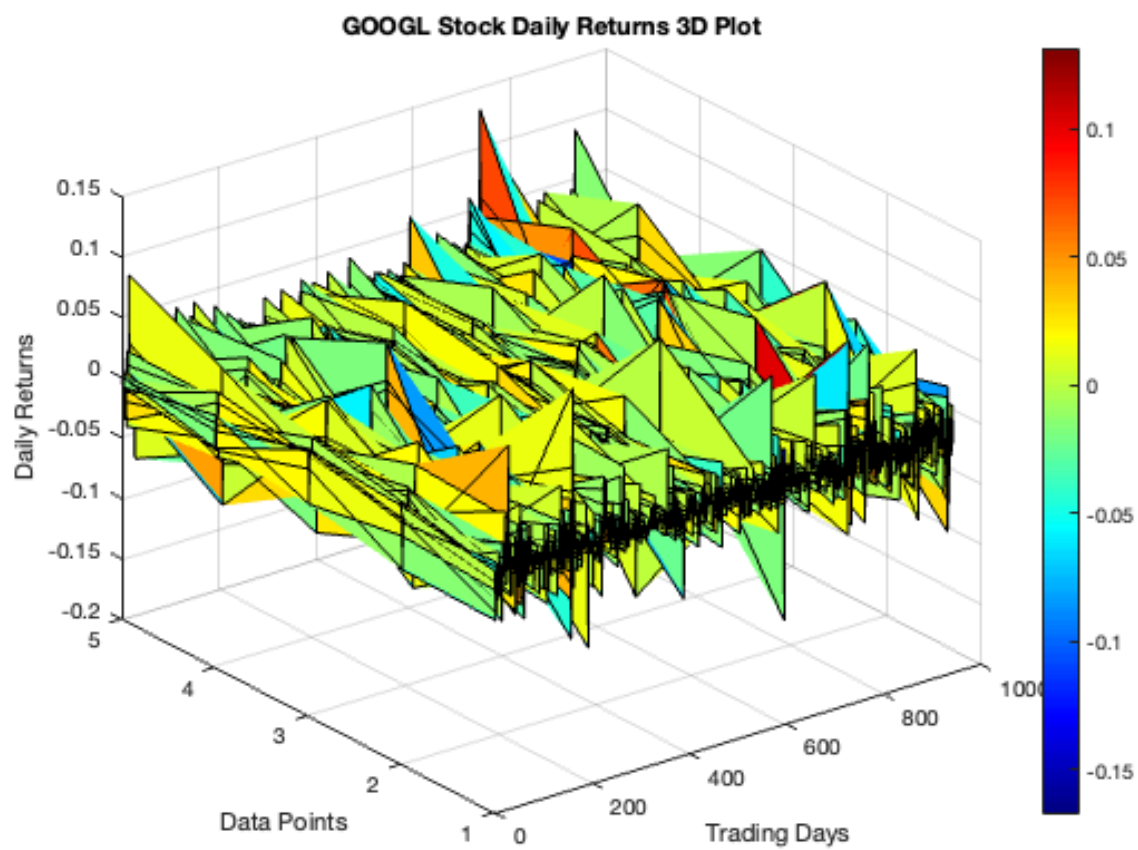
```

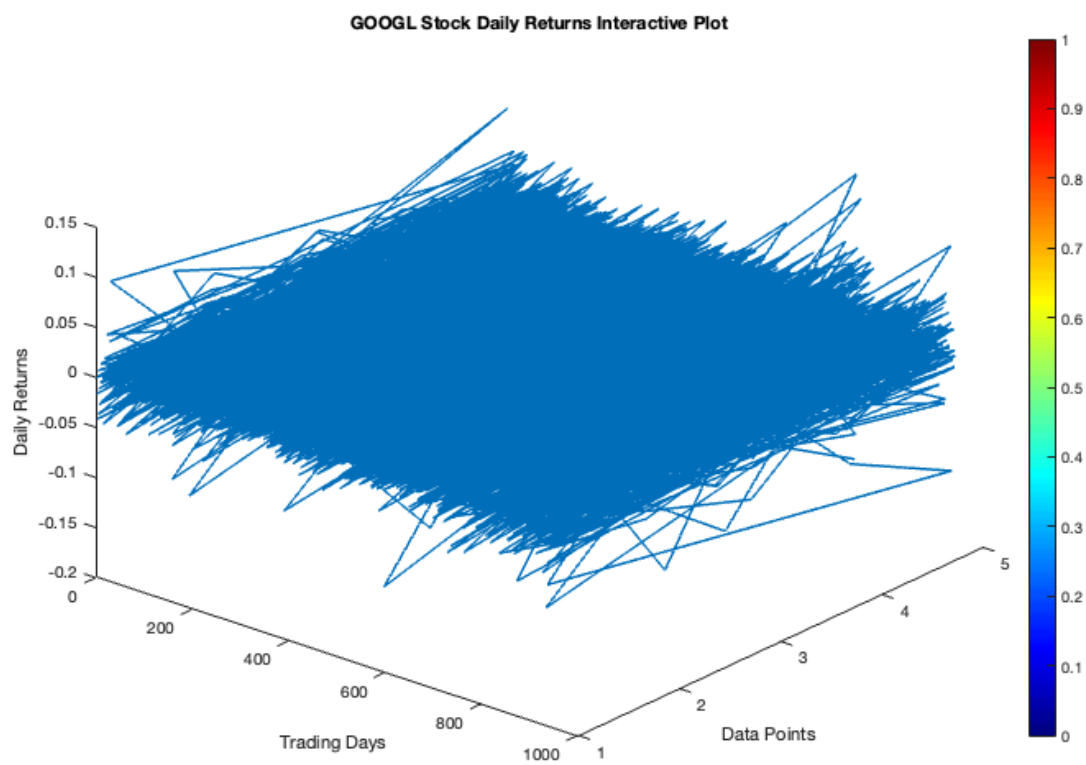


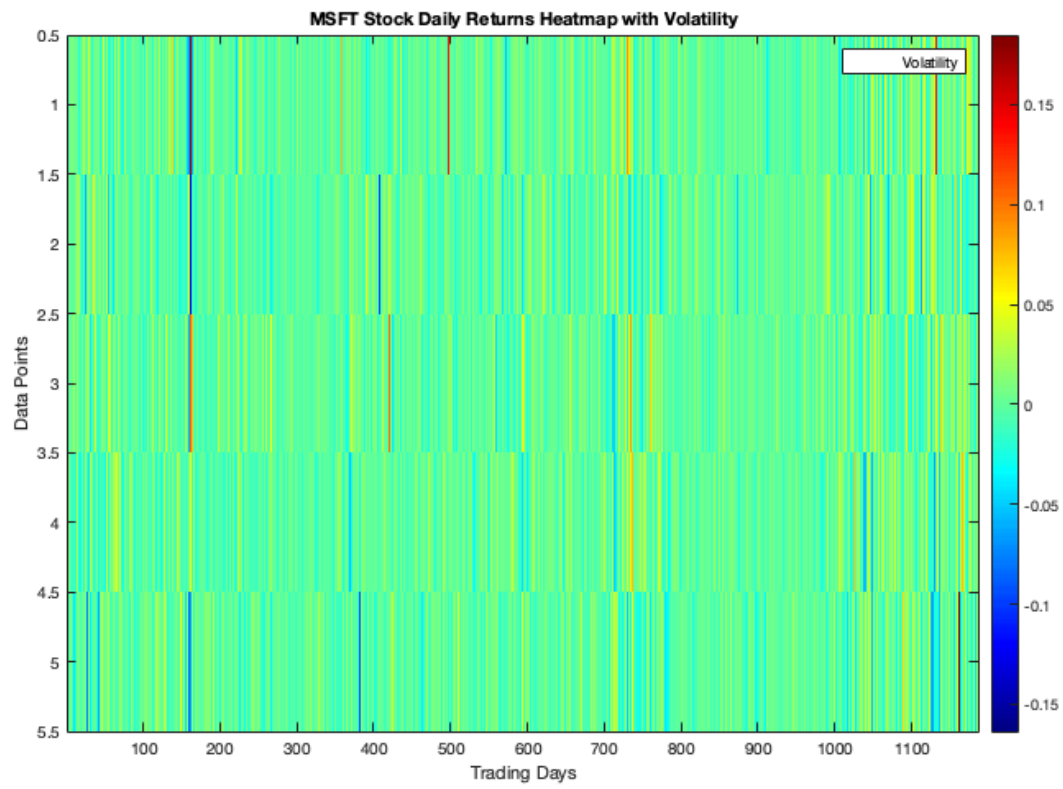


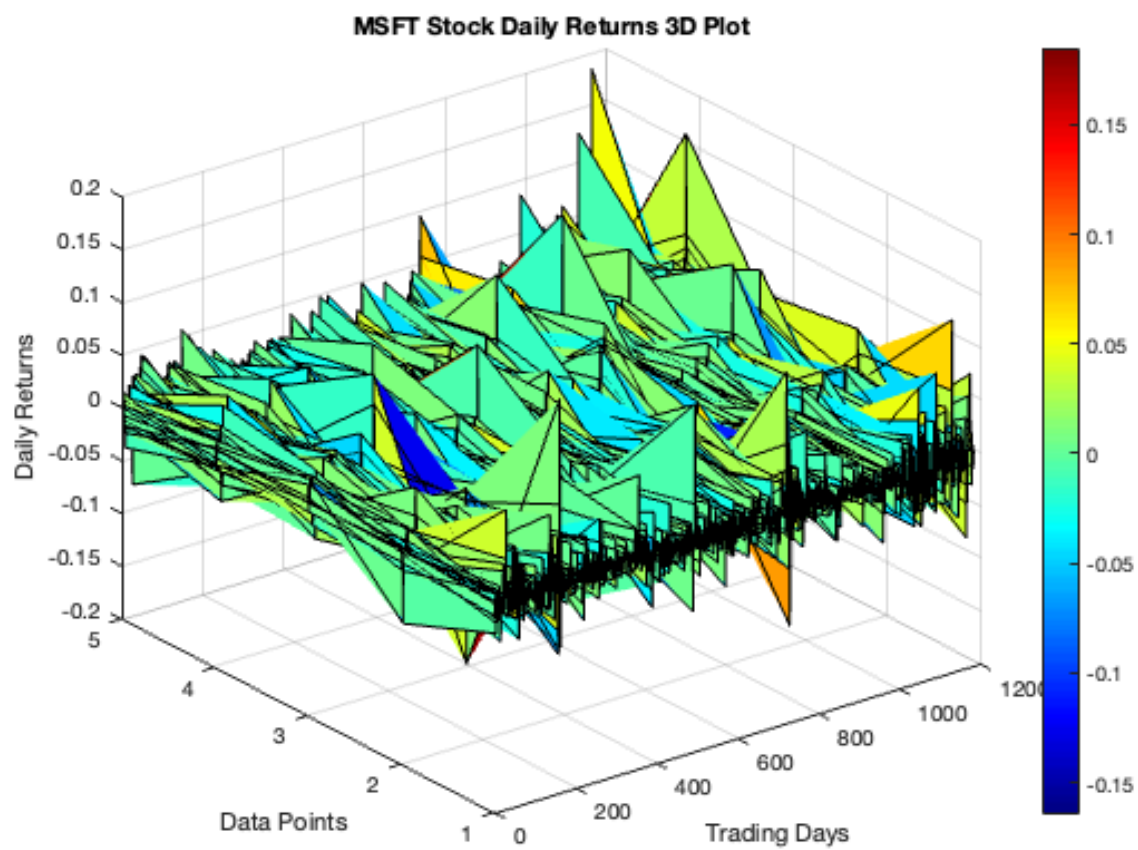


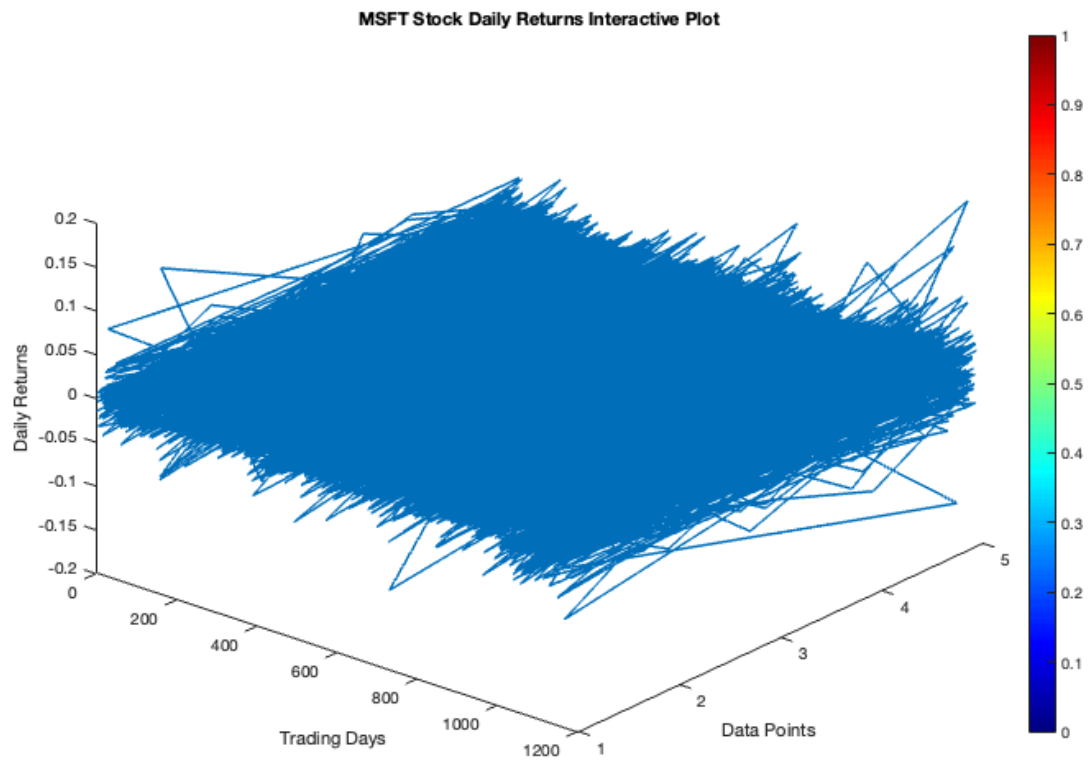












Published with MATLAB® R2021b