

Unit 10—A Yen for the Future

Yen Photo

Background

The financial departments of large companies often deal with foreign currency transactions while doing international business. As a result, they are always looking for anything that can help them better understand the future direction and risk of various currencies. Hedge funds, too, are keenly interested in anything that will give them a consistent edge in predicting currency movements.

In this assignment, you will test the many time-series tools that you have learned in order to predict future movements in the value of the Japanese yen versus the U.S. dollar.

You will gain proficiency in the following tasks:

1. Time Series Forecasting
 2. Linear Regression Modeling
-

Files

Use the following starter code to complete this assignment.

Note: The starter code shows example calculations and figures to use as a guide. However, your actual output may differ depending on the code and data used.

Time-Series Starter Notebook

Linear Regression Starter Notebook

Yen Data CSV File

Instructions

Time-Series Forecasting

In this notebook, you will load historical Dollar-Yen exchange rate futures data and apply time series analysis and modeling to determine whether there is any predictable behavior.

Follow the steps outlined in the time-series starter notebook to complete the following:

1. Decomposition using a Hodrick-Prescott Filter (Decompose the Settle price into trend and noise).
2. Forecasting Returns using an ARMA Model.

3. Forecasting the Settle Price using an ARIMA Model.
4. Forecasting Volatility with GARCH.

Use the results of the time series analysis and modeling to answer the following questions:

1. Based on your time series analysis, would you buy the yen now?
2. Is the risk of the yen expected to increase or decrease?
3. Based on the model evaluation, would you feel confident in using these models for trading?

Linear Regression Forecasting

In this notebook, you will build a Scikit-Learn linear regression model to predict Yen futures (“settle”) returns with *lagged* Yen futures returns and categorical calendar seasonal effects (e.g., day-of-week or week-of-year seasonal effects).

Follow the steps outlined in the `regression_analysis` starter notebook to complete the following:

1. Data Preparation (Creating Returns and Lagged Returns and splitting the data into training and testing data)
2. Fitting a Linear Regression Model.
3. Making predictions using the testing data.
4. Out-of-sample performance.
5. In-sample performance.

Use the results of the linear regression analysis and modeling to answer the following question:

- Does this model perform better or worse on out-of-sample data compared to in-sample data?
-

Hints and Considerations

- Out-of-sample data is data that the model hasn’t seen before (Testing data).
- In-sample data is data that the model was trained on (Training data).

Submission

- Create Jupyter Notebooks for the analysis and host the notebooks on GitHub.
 - Include a Markdown that summarizes your models and findings and include this report in your GitHub repo.
 - Submit the link to your GitHub project to Bootcampspot.
-

Requirements

Time Series Forecasting (20 points)

To receive all points, your code must:

- Utilize the Hodrick-Prescott Filter to decompose the settle price into trend and noise. (5 points)
- Use the ARMA Model to forecast returns. (5 points)
- Use the ARIMA Model to forecast returns. (5 points)
- Use the GARCH Model to forecast returns. (5 points) ##### Time Series Analysis (15 points)

To receive all points, your code must:

- Analyze the yen to decide whether to make a purchase. (5 points)
- Analyze the risk of the yen. (5 points)
- Analyze the confidence of models as a basis for trading. (5 points)

Linear Regression Forecasting (30 points)

To receive all points, your code must:

- Prepare the data, create returns and lagged returns, then split the data into training and testing sets. (5 points)
- fit a linear Regression model to the data. (5 points)
- Make predictions using the testing data. (6 points)
- Evaluate the out-of-sample performance. (7 points)
- Evaluate the In-sample performance. (7 points)

Linear Regression Analysis (5 points)

To receive all points, your code must:

- Analyze the model performance for out-of-sample and in-sample data and write a conclusion. (5 points)

Coding Conventions and Formatting (10 points)

To receive all points, your code must:

- Place imports at the beginning of the file, just after any module comments and docstrings and before module globals and constants. (3 points)
- Name functions and variables with lowercase characters and with words separated by underscores. (2 points)
- Follow Don't Repeat Yourself (DRY) principles by creating maintainable and reusable code. (3 points)

- Use concise logic and creative engineering where possible. (2 points)

Deployment and Submission (10 points)

To receive all points, you must:

- Submit a link to a GitHub repository that's cloned to your local machine and contains your files. (5 points)
- Include appropriate commit messages in your files. (5 points)

Code Comments (10 points)

To receive all points, your code must:

- Be well commented with concise, relevant notes that other developers can understand. (10 points)

© 2021 Trilogy Education Services