Learning Objectives

Retrieving Data

1. FRED
2. WRDS
3. yfinance

Linear algebra

1. Matrix operations
2. Covariance & correlation matrices
3. Implied NxN covariance matrix from KxK factor matrix.

Stationarity Measures

1. Skewness
2. Kurtosis

Time series analyse.

4. Autocorrelation
5. Autoregressions
    a. GARCH
6. Heteroscedasticity
7. Endogeneity
8. Seasonality

Stationarity Transformations

9. Standardization (Z-Values)
10. Min-Max Normalization
11. Windsorization
12. Stationarization
13. Fractional differentiation

Hand-written Matrix Quiz:

1. Refer to the THIS worksheet for steps to calculate implied covariance matrices.
    a. To better understand matrix operations, check out THIS video series.
    b. If you are not feeling up to this, don't worry, Michael and Oliver are the math experts, and they will show us how to do this in a future working session.

Some Context for a Career in Quant Research:

Marcos Lopez De Prado is a leader in the space of quant finance. I highly recommend watching THIS presentation. In this presentation, he mentions a very important understanding of stationarity and how we can do this effectively. Ask yourself if we should use returns or some other version of the raw data? What is the trade off between precision and recall.

Python Coding Exercise Part 1: *Adjusting Time Series*

In this assignment, we are going to try predicting the price of the raw materials index by using the manufacturing orders index as the independent variable. We will analyse the distribution characteristics, and then transform our features to hopefully improve the model.

1. **Retrieve 3 Datasets from FRED**.
    a. From FRED, retrieve:
        i. 20 years of Manufacturing Orders Index (MOI). Use THIS for API reference, or download the CSV from THIS site.
        ii. 20 years of Raw Materials Index (RMI). Download the CSV from THIS link.
        iii. 20 years of Producer Price Index (PPI). Download CSV from THIS link.
2. **First Attempt:** Raw Data
    a. Using monthly observations, Calculate the spearman autocorrelation between the raw materials index, PMI, and PPI data with a lag of 1 through 30 months. For this, just start by using the raw values.

Compare your results and determine which lag and which relationship is the strongest. Produce a scatter plot using the optimal lag.
    i. Documentation HERE and video description HERE.
b. Next, create a new column as the percent change of the MOI and RMI data. Run the autocorrelation test again and interpret the results. Are the correlations more significant with percent change values? Does a stationary time series work better?
    i. To do this properly, you should iterate over the different lags and plot the correlation values.

3. **Second Attempt:** Features Transformations
    a. Calculate the stationarity of both time series using Augmented Dickey Fuller test. Documentation HERE. Interpret the statistical strength of stationary.
        i. You will find that the PMI data is highly seasonal and up-trending (non-stationary).
    b. Adjust the RMI for PPI inflation. This is very simple, just subtract the PPI% change from the RMI% change.
    c. Once adjusted for inflation, run the Augmented Dickey Fuller test once again and interpret the results. What are the distribution parameters for the adjusted data? When plotting the RMI, does it look more realistic when adjusted for inflation?
    d. Next, we need to decompose the trend and seasonality from the adjusted MOI data. Refer to THIS for documentation.
4. **Run a Regression.**
    a. Using the residual values form the de-seasonalized as the independent variable, and the RMI percent change as the dependent variable. Run regressions with lag 1 through 30 and plot the R-squared values for each. Interpret your results. Which lag performs best for predicting changes in RMI?
    b. Now, using the raw (unadjusted for inflation) percent changes for RMI, Run multiple regression using the PPI% change and MOI residual% change as X values and the RMI% change as Y. How much additional variation are we able to explain by adding the PPI?
    c. Should we use this factor to predict prices in our model?

Python Coding Exercise Part 2: *Trading Indicators & Machine Learning*

In this exercise, the goal is to build a feature that we think will predict profitable trades and visualize it's performance. The focus of this exercise is not the transformations but rather the back-testing process, as well as testing the algorithm out-of-sample.

Building a Trading Signal

Simply building factors that aim to predict prices in the future might be a fool's game. Just think about it: when we use regressions to test a strategy, we are restricting ourselves to a strict investment horizon. What if the trading signals work well but the time horizon for each trade is not known when we execute the trade?

Instead of predicting the price, we can build indicators that identify entry and exit points (i.e., when to buy and when to sell). W are going to start by building an algorithm that trades the S&P500 based on 2 factors: the relative strength index (RSI) of the S&P500 and VIX inflection.

1. Using yfinance API, import 30 years of historical data for the VIX and the S&P500 (just ask ChatGPT to do this for you)

a. Create a new column called SPY_ret as the percent change of S&P500 prices.

Building the optimal RSI indicator

2. Next, built the relative strength index (RSI) for the S&P500
   a. Refer to THIS for instruction on building the relative strength index. (or just ask ChatGPT). Make note of any assumptions you used for this.
   b. Plot the S&P500 price with the RSI below it.
3. Test-Train split
   a. Spit the data into a test set (most recent 5 years) and a training set (first 25 years). Use the training data for the following steps.
4. For the RSI, notice that there is a high and low threshold at which point the S&P500 changes direction. Use your data science skills to find these upper (sell) and lower (buy) thresholds that offer most profitable trading strategy.
   a. For this, you will need to create a new column that assigns a 1 or -1 to all values, call this column "signal". The 1 or -1 will indicate whether you are long or short the S&P500. For example, let's say you are going to buy if the RSI is below 20 and sell if the RSI is above 80. Create a dummy flag variable and set it to 0. The flag indicates if you bought or sold in the past. Bought; flag = 1, sold; flag = -1, no position; flag = 0. Now loop through the RSI values from oldest to newest. Set a condition that when true changes the flag based on the RSI value, and then all row values for the "signal" column are set to the flag.
   b. Then create another column by taking the cumulative sum product of the S&P500 % returns and the "signal" column and call it "cumulative profit". Now, plot the S&P500 and the "cumulative profit" together and see how the trading strategy works. You will need to scale the S&P500 by dividing it by 100 or by making the y axis independent.
   c. Finally, chose a range of values for the upper and lower levels, then iterate over all combinations of the levels and find the levels that maximize the information ratio. Information ratio measures the performance of your trading strategy compared to a simple buy-and-hold strategy of the S&P500 .
      i. $IR = \frac{(R_{CumulativeProfit} - R_{S\&P500})}{STD(R_{CumulativeProfit} - R_{S\&P500})}$
      ii. For more documentation on the information ratios, refer to THIS link.
      iii. The end goal of this is to find the RSI levels that produce the best risk adjusted returns.
5. Now use the optimal levels on the test data to see how well your strategy performs out of sample.

Building the Optimal VIX indicator

6. Next, plot the VIX below the S&P500 and perform a similar analysis. Do you see any thresholds or ranges that might make good trading signals? This one is harder to spot. Instead of looking for levels or thresholds directly from the VIX price, transform the VIX into its 2nd derivative. The number of observations you use for this is your choice, but somewhere in the range of 3 – 10 seems reasonable.
   a. Using a rolling window of 3-10 days, calculate the 2nd derivative at time t. For example, using 3 observations: $F''(t) = F''([t_{-3}, t_{-2}, t_{-1}, t])$. Call this new column "VIX inflection".
   b. Now, plot the VIX inflection values below the S&P500 and see if you can spot the relationship. Feel free to test this with a regression. Do you think that an inflection in the VIX leads price action in the S&P500? You may want to plot only 100 – 300 observations for this because plotting the whole time series will be impossible to interpret.
   c. Repeat the steps from part 2 to find the optimal levels for the VIX indicator that return the maximum information ratio.

   d. Make sure you split the data into training and testing data before optimizing for the levels.
7. Test the optimal levels out-of-sample using the test data. Plot your results.

Putting The Two Indicators Together.

You might say at this point "what's the points of combining these indicators if neither work in isolation?" This is a fair point, but we should recall that all indicators contain some information, and we shouldn't expect any single indicator to work all the time. The point of combining indicators is because we need to consider the possibility that the indicators are conditional on each other. One indicator cannot produce results on its own, but perhaps one indicator compliments the other.

When I go outside and ask myself if its going to be a great day, I could ask myself two exclusive questions: 1) if the sky is blue, I have a great day 30% of the time. And 2) If I ate breakfast, I have a great day 50% of the time. In the case of exclusive questions, I am omitting the possibility that a great day is even more likely when the sky is blue and I ate breakfast, which is obviously going to result in a better day.

Let's see how it works.

1. Now that you have the optimal levels for both RSI and VIX, use those levels to create the "signal" column for the RSI and VIX. (this should already be completed in the last steps).
   a. Now you need to create a target column that represents if those trades were profitable. Make a new column that calculates the future profit/loss of each trade. For example, every time the signal column changes from 1 to -1, or vise versa, you want to find the profit between that trade and the next trade. All other observations can be set to 0. Create this "profit" column for the RSI and VIX signals.
   b. Next, create one last column as the "total profit" which is the sum of the RSI profit and VIX profit.
   c. If you've made it this far, you are doing very well. The data science required to do this was tricky.
2. The final step to this process is to use a logistic regression machine learning algorithm to improve this model once more.
   a. If you're new to ML, refer to THIS video tutorial for simple logistic regression models.
   b. The feature columns are "RSI signal" and "VIX signal" and the target column is "total profit"
   c. You might want to scale the total profit column using MinMaxScalar.
   d. Fit the ML model on the training data.
   e. Test the model out-of-sample on the test data and then interpret the confusion matrix.
      i. Using the predicted values output as the signal, any predicted value below 0 was a short, and any predicted value above 0 was a long.
      ii. Using those new trading signals, recreate the cumulative profit column and then plot the performance of the strategy.
      iii. If you want to take this to the next level, you can use the probability weighted predictions to adjust the allocation per trade.
      iv. Interpret the confusion matrix and compare the precision of the combined indicators to the precision of the individual indicators. By how much did you improve the probability of a successful trade by combining the indicators?

Congratulations, you just did an advanced coding project. You're ready to start market research. Remember to be creative when building factors, while also aiming for simplicity.