Code Readme

This document serves as a guide to the scripts used for the thesis “Market and Volatility Timing Using Gamma”. These scripts cover all analysis and results included in the paper. They are separated in to 5 different groups based on their function.

1) Data cleaning and aggregation
2) Data analysis
3) Regressions
4) Trading strategies
5) Modules

The scripts, their function and their output result will be described briefly here. At the top of all scripts, some input parameters can be specified. Some scripts contain lists of the assets and will loop through the list to do all the analysis automatically. Some scripts require changing these inputs to get the results for the desired asset.

1. Data Cleaning and Aggregation

These scripts serve to load and clean raw option data, and aggregate the data to the needed format for analysis. They will save the new data into excel files and store them in separate folders, to be used later for the analysis. As the zip-folder contains all readily cleaned and aggregated data, these scripts will not have to be run for replication. The following scripts are included in this section.

- Pepareoptiondata.py
    - Loads raw option and spot data, removes unwanted columns, adds columns such as moneyness flags, mid-price and spot price to option data. Stores prepared data in *Cleandata*-folder.
- Aggregateoptiondata.py
    - Loads option data and spot data from *Cleandata*-folder, and aggregates the important measures needed for further analysis. It consolidates option measurements such as open interest, gamma, volume across all contracts and stores them in a daily time-series.

2. Data Analysis

datanalysisfinal.py

- This script covers all the plots displayed in the empirical analysis section of the paper. One need only to adjust the index and ETF at the top and it will output all the plots used (and some unused)**.**

3. Regressions

These scripts cover all the regressions used in the paper. There are quite many of them, with different variations. For simplicity, I separated the variants by script. They are categorized as follows.

- regressions.py
    - Produces the results from the original regression framework, control variables and liquidity.
    - Prints results for tables: 4, 5, 9 in main body.
    - Prints results for tables in appendix C.2 and C.3.
    - Lags must be set manually, default is one day of lag.
- regressionsvolindex.py
    - Similar to regressions.py, but this script uses implied volatility indices as control variables.
    - Script produces tables displayed in appendix. C.3 and C.4.
- regressionsaggregate.py
    - Run aggregate regression where gamma for index and etf is combined.
    - Produces table 6 in main body and table in appendix C.5.
- regressionstotal.py
    - Run total regression where gamma for all indices is combined.
    - Run regressions where total gamma for other assets is used as control variable.
    - Computes return and gamma correlation.
    - Produces table 7 and 8 and figure 10 in main body.
- regressionsreversals.py
    - Run reversal regressions described by equations 13 and 15 in the main body.
    - Produce table 10 and
- regressionslagloop.py
    - Runs original regression repeatedly for different lags and stores the results.
    - Generates figure 9 displayed in the main body.

4. Trading Strategies

These scripts contains the scripts for backtesting the trading strategies described in the paper. The straddle and reversal strategies are covered individually.

- straddlestrategiesfinal.py
    - This script contains the backtest for the straddle strategy.
    - It computes the return for the long straddle and short straddle strategies separately.
    - Both the benchmark and the realistic strategies are computed here
    - For transaction costs, one should toggle include_tc = True at top of script. Returns will then be computed on bid-ask prices.
    - The plot generates all plots and tables related to the straddle strategies in the paper.

- reversalstrategies.py
    - This script contains the backtest for the reversal strategies.
    - It computes all results associated with this section of the paper.

5. Modules

The folder also contains two modules. These are scripts containing functions used at various points in the analysis. They have been fully programmed from scratch by the author.

- Backtest.py
    - This module contains useful functions for backtesting strategies, such as computing returns, performance and turnover.
- gammafunctions.py
    - This module contains functions used in the dataanalysis script, for easy computation of different statistics.