DEGREE OF MASTER OF SCIENCE IN FINANCIAL ECONOMICS

FINANCIAL ECONOMETRICS

HILARY TERM 2020 COMPUTATIONAL ASSIGNMENT 1 (PRACTICAL WORK 3)

November 2020.

Assignment must be submitted before noon (12:00) Friday 15 January 2020 (0th Week) by uploading to SAMS.

This is group work. Groups of three or four are permitted.
Groups with less than three or more than four are not permitted.
All solutions must be submitted by the due date and time.
Do not write the names of members of your group on your submission.

*Candidates should answer **all** questions.*
Answers will be assessed on the quality of the answer, not the quality of the code.

Suggested Length: 10 pages; limit lesser of 15 pages or 3,750 words.
Material on pages 16+ will not be assessed.
The limit does not include a cover sheet or academic honesty declaration. All material, including figures, equations, explanatory text, and code, must fit within 15/3,750 words pages.

Assessment

This assignment is assessed in 2 parts:

- 67% - Report. This report should focus on analysis and synthesis and not code or the numerical values of the problems. Tell a convincing story detailing the regularities you have documented in your analysis.

- 33% - Autograder. 2 functions must be submitted to compute the required outputs using the inputs. The signature of each function is provided as part of the problem. You must ***exactly*** match the function name. Submissions must be in Python and must be in a single Python file (*some_filename*.py) containing all functions. IPython notebooks are not accepted. If using Python, pay close attention to the input and output dimensions. All data inputs will be pandas DataFrames or Series, as indicated in the program description. **Note:** *Please* run your code in the function you submit to ensure that it does not produce an error. A function that errors when run is given a mark of 0. The autograder uses loose criteria when judging correctness, and so any value within about 1% of the reference will be marked as correct. See the example file solutions-pw2.py for the structure of the file expected by the autograder.

Tips for Autograded Code

- Your submissions **MUST** be a Python file (.py). IPython notebooks are not accepted, and submitting your solution in a notebook will result in a mark of 0.

- Your submission will not have access to any data files. You must not read or write anything from your program.

- Your submission will be run in a random directory. You cannot assume anything about data files existing in any particular location. Your code does not need to access data. ALl required data is passed through the inputs.

- You can use demo-autograder-pw2.py to check that your solution accepts the required arguments and returns values with the correct type and shape. You should run the program along with your code *in an empty directory*, which is how the autograder runs it. Code similar to

```
mkdir empty
cd empty
<copy demo-autograder.py to empty along with your submission>
python demo-autograder-pw2.py
```

  can be used to check that your code will run correctly.

- The autograder submission should not normally have code that you wrote as part of the analysis. It should only contain the required functions, imports, and code necessary to run the functions. In *ideal* submission would not execute any code when imported, and instead would only contain import statements and functions.

```
import numpy as np
import pandas as pd

def first_function(a, b):
    # Do something
    return "something"

def used_by_second_function(x, y):
    # Do something
    return "something", "else"

def used_by_second_function(x, y, z):
    w = used_by_second_function(x, y)
    return w[0], w[1], z
```

Problem 1

Data

You will need the following data to complete this assignment:

1. The momentum factor from Ken French's website (monthly).
2. The 1, 5, and 10-year constant maturity series from FRED.
3. The AAA and BAA (Moody's) from FRED.
4. The monthly returns on the VWM from Ken French's site.
5. Monthly data on core CPI from FRED.
6. Monthly data in the unemployment rate from FRED.
7. Monthly data on Industrial Productivity from FRED.

Variable Construction

1. TERM - The difference between the 10-year and the 1-year

$$(Y_{10} - Y_1).$$

2. CURVE - The 10-year yield plus the 1-year yield minus 2 times the 5-year yield

$$Y_{10} - 2Y_5 + Y_1 = (Y_{10} - Y_5) - (Y_5 - Y_1).$$

3. DEFAULT - The difference between the AAA and the BAA yields

$$(Y_{AAA} - Y_{BAA}).$$

4. INFLATION - The Year-over-Year difference of log CPI

$$\ln CPI_t - \ln CPI_{t-12}$$

In-sample and Out-of-sample

In-sample exercises should use all available data. Out-of-sample exercises should split the data in half. You will need to construct the common sample of the relevant variables (this may differ between problems 1 and 2). You should construct a single sample where all data is available and only use this common sample.

Analysis

Assess the predictability of the monthly returns of the VWM and the Momentum portfolio, both in-sample and out-of-sample. You should examine the range of available predictors (in the data set constructed).

1. Can a predictive model improve the Sharpe ratio in-sample?
2. Is this reproducible out-of-sample?
3. How are in-sample and out-of-sample R^2 related?

The "Out-of-sample R^2" is defined as

$$R^2_{OOS} = 1 - \frac{\sum_{t=\tau}^{T} \left(Y_t - \hat{Y}_{t|t-1}\right)^2}{\sum_{t=\tau}^{T} \left(Y_t - \hat{\mu}_{t|t-1}\right)^2}$$

where $\hat{Y}_{t|t-1}$ is the model-based forecast and $\hat{\mu}_{t|t-1}$ is the sample mean using data up to $t-1$ (that is, it is the simple mean only forecast). Finally, τ is the starting point of the forecasting exercise. You should consider both univariate regressions using each of the predictors, model-selection-based models (e.g., as selected by the AIC or BIC), and a "kitchen sink" model that uses all predictors. When implementing a model-selection procedure and assessing out-of-sample fit, the selection procedure should be implemented using only the in-sample period.

***Note**: You may need to time-align observations so that the data for January 2000 is aligned (e.g., the January 2000 unemployment is aligned with the January 2000 return). This is unrealistic since many of these measures are not available in real-time, but it will dramatically simplify the problem.*

Notes

The basic regression is[1]

$$R_{t+1} = \beta_0 + \beta_1 X_t + \varepsilon_{t+1}.$$

This is the same whether in-sample or out-of-sample. In-sample means that the same data is used to estimate model parameters as is used to evaluate the model. In this problem, this means that you use the full-sample estimates of $\hat{\beta}_0$ and $\hat{\beta}_1$. Out-of-sample means that estimates of $\hat{\beta}_0$ and $\hat{\beta}_1$ use data from $1,\ldots,\tau$ in estimation and then $\hat{R}_{\tau+1} = \hat{\beta}_{0,\tau} + \hat{\beta}_{1,\tau} X_\tau$ where the notation $\hat{\beta}_{j,\tau}$ is used to indicate that is estimate makes use of data until period τ.

[1] You might also fit more complex models such as the Kitchen Sink or one selected by a model selection procedure. This main idea is that same since the input to the next step is $\hat{R}_{t+1}$.

Once you have a series of expected returns, either in sample, $\left\{\hat{R}_t^{\text{IS}}\right\}_{t=\tau+1}^{T}$ or $\left\{\hat{R}_t^{\text{OOS}}\right\}_{t=\tau+1}^{T}$ where τ is the point where the sample is split (e.g., 50% of the sample size), these must be turned into portfolio weights using some function. One example function is

$$w(\hat{r}) = \begin{cases} 1.5 & \hat{r}_{t+1} > \bar{r} \\ 0.5 & \hat{r}_{t+1} \leq \bar{r} \end{cases}$$

so that when the expected return is above the historical return one would invest 50% more (using leverage) and when it is below one would invest 50% and hold 50% in a risk free asset. Choosing a weighting function is up to your group. The ultimate quantities to calculate are the portfolio returns

$$R_{p,t+1} = w\left(\hat{R}_{t+1}\right) r_{t+1} + \left(1 - w\left(\hat{R}_{t+1}\right)\right) R_{f,t+1}.$$

Importantly here $w\left(\hat{R}_{t+1}\right)$ is known at time t when using out-of-sample values.

Finally, the portfolio returns can be used to compute the Sharpe ratio in the usual manner. Note that when comparing in-sample and out-of-sample quantities, you should only compare the observation in the common sample (the second half of the sample). It doesn't make sense to compare a portfolio return or R^2 using different time periods.

Problem 2

Data

- Factor returns on the Value and Momentum portfolios from Ken French's site
- The 6 Portfolios Formed on Size and Momentum
- The 6 Portfolios Formed on Size and Value
- The 17 Industry Portfolios

Note the the the Value return is constructed from the 6 Size-Value portfolios as

$$HML = 1/2(SV + BV) - 1/2(SG + BG)$$

where S is for small, B is for big, V is for value, and G is for growth. Momentum is similarly defined from the 6 Size-Momentum portfolios as

$$MOM = 1/2(SH + BH) - 1/2(SL + BL)$$

where H is high momentum and L is low momentum.

Analysis

1. Critically assess the ability of different model selection procedures to construct accurate out-of-sample tracking portfolios for the Value and Momentum factors using the industry portfolios, assuming

   (a) You train your models using 5-years of data and hold for 5 years;

   (b) You train using 10-years of data and then hold for 5 years; and

   (c) You train using 20-years of data and then hold for 5 years.

   You should implement these using a 5-year rolling scheme (i.e., estimate using data in 1940-1944, then hold 1945-1949, re-estimate using data in 1945-49, then hold 1950-54, and so on), and only compare samples where all three have predictions available.

2. For each of the two factors considered, examine the ability to track the long-side (positive weights) and the short-side (negative weights) of each factor separately. Comment on the ease or difficulty of replicating the components as opposed to the entire return.

3. Does a combination of your two or three best procedures outperform the components? Note that a combination forecast of n methods is just

$$\hat{Y}_{t+1} = n^{-1} \sum_{i=1}^{n} \hat{Y}_{i,t+1}.$$

Notes

A tracking regression has the form

$$R_{p,t} = \sum_{i=1}^{k} \beta_i R_{i,t} + \varepsilon_t$$

where $R_{p,t}$ is the return on the portfolio being tracked and $R_{i,t}$ are the returns assets used to track. There is no constant in this regression.

You can assess the out-of-sample tracking performance by examining the variance of the out-of-sample tracking residuals and comparing these to the out-of-sample variance of the tracked portfolio.

You might want to consider some of GtS and StG (using either p-values or an Information Criteria), Forward Stepwise, Backward Stepwise, or Hybrid Stepwise Procedures, Ridge Regression and LASSO or Random Forests. It is not expected that you will try them all. A good effort will try a variety of distinct approaches but may have only a few (possibly 1) of each major type.

Code Problems

Out-of-Sample R^2

Produce a function that will compute the out-of-sample R^2

```
r2 = oos_rsquared(y,yhat,mu)
```

Outputs

- `r2` - The out-of-sample R^2 (float)

Inputs

- `y` - A pandas Series with shape n. The out-of-sample realized value.
- `yhat` - A pandas Series with shape n. The forecasts of `y`. The index of `yhat` will match that of `y`, so that observation i of `yhat` will be the forecast of `y` in position i.
- `mu` - A float. The in-sample mean of the time-series. Essentially a forecast of `y` assuming that the correct model has a constant mean.

Out-of-Sample Residual Construction

Compute out-of-sample residuals for values stored in a Series where regressors are a DataFrame and parameters are a Series.

```
resid = oos_residuals(y, x, beta, first, last)
```

Outputs

- `resid` - A pandas `Series` with shape n. The value of $Y - \mathbf{X}\hat{\beta}$ for the relevant sample.

Inputs

- `y` - A pandas `Series` with shape T. `y` will have a `DatetimeIndex`.
- `x` - A pandas `DataFrame` with shape T by k. The index of `x` will match `y`.
- `beta` - A pandas Series with shape k. The regression coefficients. The index of `beta` will match the column names of `x`.
- `first` - A date in string format, e.g., `"1970"` or `"1974-03-01"`.
- `last` - A date in string format, e.g., `"1980"` or `"1979-03-01"`.

Notes

You should return the residuals only for the sample bracketed by first and last (inclusive).