DEGREE OF MASTER OF SCIENCE IN FINANCIAL ECONOMICS

FINANCIAL ECONOMETRICS

HILARY TERM 2020 COMPUTATIONAL ASSIGNMENT 2
PRACTICAL WORK 4

February 2020.

Assignment must be submitted before
noon (12:00) March 20 2019 (9th Week)
by uploading to SAMS**.**

This is group work. Groups of 3 or 4 are permitted.
Groups with less than 3 or more than 4 are not permitted without explicit permission.
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.*
Suggested Length: 7 pages; limit 10 pages.
Material on pages 11+ will not be assessed.
Limit does not include the cover sheet, academic honesty declaration or submitted code files.
All material, including figures, equations, and explanatory text must fit within 10 pages.

Note: Text in red differs from that hard copy distributed in class.

Assessment

This assignment is assessed in 3 parts:

- 67% - Report. This report should focus on the analysis of the results and not code or the numerical values of the problems.
- 33% - Autograder. 4 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.

  Tips:

  - Verify that the output shapes exactly match the expected output shapes.
  - If using MATLAB, ensure that the m-file names exactly match the function name (e.g., `weighted_hs` is in weighted_hs.m)
  - When submitting your work, *do not* further zip your code files. They should be part of your assignment just like your PDF or word document.

Data

You will need the following data to complete this assignment:

- Gold Futures (GC) (precious metal)
- Crude Futures (SL) (energy)
- Copper Futures (HG) (industrial metal)
- Wheat Futures (W) (agricultural)

All of these data sets are available on the assignment website,

`https://www.kevinsheppard.com/teaching/mfe/practical_work_4`

All series are daily with different start dates.

Problem 1

1. Using in-sample estimates, build and test Value-at-Risk models for 1-day, 5-day and 10-day holding periods for Copper, Crude, Wheat, and Gold. You should consider alternative conditional and unconditional methods. You should use the maximum sample in each market and drop days where returns are not available.

2. Take your models out-of-sample, using a 50% split. How do the models perform out-of-sample?

3. Can you identify periods where different models perform well/poorly?

Code Problems

Weighted Historical Simulation

Implement Weighted Historical Simulation 5% VaR using a recursive generation scheme.

$$w_i = \lambda^{t-i}(1-\lambda)/\left(1-\lambda^t\right), \quad i = 1, 2, \ldots, t$$
$$\hat{G}_t(r) = \sum_{i=1}^{t} w_i I_{[r<r_i]}$$

```
[var_forecast] = weighted_hs(returns, lambda, sample_size)
```

Outputs

- `var_forecast` - T by 1 vector of 1-step ahead VaR forecasts. The forecast produced using data from observations $1, 2, \ldots, t$ for day $t+1$ must appear in location $t+1$. The first `sample_size` values should be `nan`.

Inputs

- `returns` - T by 1 vector of returns.
- `lambda` - smoothing parameter $\in (0, 1)$.
- `sample_size` - the size of the window to use when computing the weighted cdf, e.g., 252 days.

Unconditional Bernoulli Test

Implement the Bernoulli Test for VaR models.

```
[decision, p_hat, llf0, llf1] = bernoulli_test(returns, forecasts, ...
                                                alpha, test_size)
```

Inputs

- `returns` - T by 1 vector of returns.
- `forecasts` - T by 1 vector of VaR forecasts where the forecast in position t is for the VaR on observation t made using data up to time $t-1$.
- `alpha` - The VaR threshold used, e.g., 5%.
- `test_size` - The size of the test, e.g. 5%.

Outputs

- `decision` - 1 if the null is rejected, 0 if not.
- `p_hat` - Estimate of the MLE value of p, the parameter of the Bernoulli distribution, computed from the inputs.
- `llf0` - scalar log-likelihood evaluated using the data when the null is true.
- `llf1` - scalar log-likelihood evaluated using the data when the alternative may be true.

Problem 2

1. Download the high-frequency return data on my website. This file contains 10 years of 5-second data. Import this into your analysis software.

2. What is the optimal sampling time for realized variance?

3. What is the optimal sampling time for RV^{AC_1}?

4. Using the optimal RV sampling scheme you arrived at in 2, estimate a HAR with 1-, 5- and 22-lags using 50% of the sample.

5. Estimate a GARCH(1,1) or GJR-GARCH(1,1) (as appropriate) using the same 50% of the sample using end-of-day data.

6. Estimate the same model you produced in 5 using the pseudo returns constructed from the realized variance and the end-of-day returns covered in the lecture slides (use 50% of the sample).

7. Compare the models in 4, 5, and 6 across 1, 5 and 22-day horizons. Use the $RV + r^2_{CtO}$ to evaluate your model. Note: The models in 4 and 6 only use open-to-close data, and so you should consider whether you need to adjust the forecasts to match the variance over the entire day, which is what the evaluation measure captures.

Code Problems

HAR

Implement a function that estimates a HAR model using 1-, 5- and 22-day lags (and a constant).

```
[params] = heterogeneous_ar(y)
```

Outputs

- `params` - 4 by 1 vector of coefficients containing the constant, 1-lag, 5-lag and 22-lag coefficients in order.

Inputs

- `y` - T by 1 vector of observed data.

Subsampled Realized Variance

Implement a function that estimates subsampled realized variance defined as

$$RV_{SS}^{(k)} = \frac{n/k}{n-k} \sum_{j=1}^{n-k} \left(\sum_{i=1}^{k} r_{j+i-1} \right)^2$$

where k is the desired number of highest-frequency returns in a single lower-frequency return and n is the total number of returns available. This estimator makes use of all consecutive blocks of return to estimate RV rather than just using disjoint (non-overlapping) blocks. The leading ratio corrects for the fact that we have too many returns in the day when we overlap.

```
[rv] = rv_ss(ret, k)
```

Outputs

- `rv` - A scalar value containing the sub-sampled realized variance.

Inputs

- `ret` - n by 1 vector of observed high-frequency returns data.
- `k` - A scalar integer containing the block size.