Ac.F633 - Python Programming for Data Analysis

Manh Pham

Final Individual Project

20 March 2024 noon/12pm to 10 April 2024 noon/12pm (UK time)

This assignment contains one question worth 100 marks and constitutes 60% of the total marks for this course.

You are required to submit to Moodle **a SINGLE .zip folder** containing a **single** Jupyter Notebook .ipynb file **OR** a **single** Python script .py file, together with any supporting .csv files (e.g. input data files. However, do **NOT** include the 'IBM_202001.csv.gz' data file as it is large and may slow down the upload and submission) AND a signed coursework coversheet. The name of this folder should be your student ID or library card number (e.g. 12345678.zip, where 12345678 is your student ID).

In your answer script, either Jupyter Notebook .ipynb file or Python .py file, you do **not** have to retype the question for each task. However, you must clearly label which task (e.g. 1.1, 1.2, etc) your subsequent code is related to, either by using a markdown cell (for .ipynb file) or by using the comments (e.g. `#1.1` or `'''1.1'''` for .py file). Provide only **ONE** answer to each task. If you have more than one method to answer a task, choose one that you think is best and most efficient. If multiple answers are provided for a task, only the first answer will be marked.

Your submission .zip folder **MUST** be submitted electronically via Moodle by the **10 April 2024 noon/12pm (UK time)**. Email submissions will **NOT** be considered. If you have any issues with uploading and submitting your work to Moodle, please email Carole Holroyd at c.holroyd@lancaster.ac.uk **BEFORE the deadline** for assistance with your submission.

The following penalties will be applied to all coursework that is submitted **after** the specified submission date:
Up to 3 days late - deduction of 10 marks
Beyond 3 days late - no marks awarded

Good Luck!

Question 1:

Task 1: High-frequency Finance (Σ = **30 marks**)

The data file 'IBM_202001.csv.gz' contains the tick-by-tick transaction data for stock IBM in January 2020, with the following information:

Fields	**Definitions**
DATE	Date of transaction
TIME_M	Time of transaction (seconds since mid-night)
SYM_ROOT	Security symbol root
EX	Exchange where the transaction was executed
SIZE	Transaction size
PRICE	Transaction price
NBO	Ask price (National Best Offer)
NBB	Bid price (National Best Bid)
NBOqty	Ask size
NBBqty	Bid size
BuySell	Buy/Sell indicator (1 for buys, -1 for sells)

Import the data file into Python and perform the following tasks:

1.1: Write code to perform the filtering steps below in the following order: (**15 marks**)

F1: Remove entries with either transaction price, transaction size, ask price, ask size, bid price or bid size ≤ 0

F2: Remove entries with bid-ask spread (i.e. ask price - bid price) ≤ 0

F3: Aggregate entries that are (a) executed at the same date time (i.e. same 'DATE' and 'TIME_M'), (b) executed on the same exchange, and (c) of the same buy/sell indicator, into a single transaction with the median transaction price, median ask price, median bid price, sum transaction size, sum ask size and sum bid size.

F4: Remove entries for which the bid-ask spread is more that 50 times the median bid-ask spread on each day

F5: Remove entries with the transaction price that is either above the ask price plus the bid-ask spread, or below the bid price minus the bid-ask spread

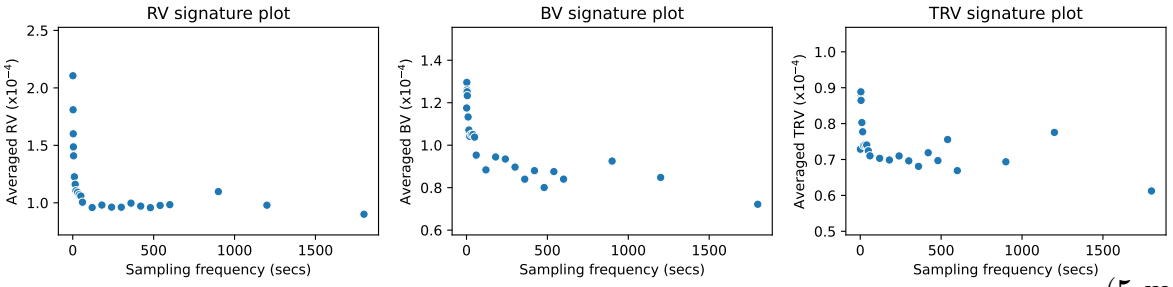
Create a data frame called `summary` of the following format that shows the number and proportion of entries removed by each of the above filtering steps. The proportions (in %) are calculated as the number of entries removed divided by the original number of entries (**before any filtering**).

	F1	F2	F3	F4	F5
Number					
Proportion					

Here, F1, F2, F3, F4 and F5 are the columns corresponding to the above 5 filtering rules, and Number and Proportion are the row indices of the data frame.

1.2: Using the cleaned data from Task **1.1**, write code to compute Realized Volatility (RV), Bipower Variation (BV) and Truncated Realized Volatility (TRV) measures (defined in the lectures) for each trading day in the sample using different sampling frequencies including 1 second (1s), 2s, 3s, 4s, 5s, 10s, 15s, 20s, 30s, 40s, 50s, 1 minute (1min), 2min, 3min, 4min, 5min, 6min, 7min, 8min, 9min, 10min, 15min, 20min and 30min. The required outputs are 3 data frames `RVdf`, `BVdf` and `TRVdf` (for Realized Volatility, Bipower Variation and Truncated Realized Volatility respectively), each having columns being the above sampling frequencies and row index being the unique dates in the sample. (**10 marks**)

1.3: Use results in Task **1.2**, write code to produce a 1-by-3 subplot figure that shows the 'volatility signature plot' for RV, BV and TRV. Scale (i.e. multiply) the RVs, BVs and TRVs by 10^4 when making the plots. Your figure should look similar to the following.



(**5 marks**)

Task 2: Return-Volatility Modelling (Σ = **25 marks**)

Refer back to the csv data file 'DowJones-Feb2022.csv' that lists the constituents of the Dow Jones Industrial Average (DJIA) index as of 9 February 2022 that was investigated in the group project. Import the data file into Python.

Using your student ID or library card number (e.g. 12345678) as a random seed, draw a random sample of **2 stocks** (i.e. tickers) from the DJIA index excluding stock DOW.[1] Import daily Adjusted Close (Adj Close) prices for both stocks between 01/01/2010 and 31/12/2023 from Yahoo Finance. Compute the **log** daily returns **(in %)** for both stocks and drop days with NaN returns. Perform the following tasks.

2.1: Using data between 01/01/2010 and 31/12/2020 as in-sample data, write code to find the best-fitted ARMA(p, q) model for returns of each stock that minimizes AIC, with p and q no greater than 3. Print the best-fitted ARMA(p, q) output and a statement similar to the following for your stock sample.

```
Best-fitted ARMA model for WBA: ARMA(2,2) - AIC = 11036.8642
Best-fitted ARMA model for WMT: ARMA(2,3) - AIC = 8810.4277
```

(**5 marks**)

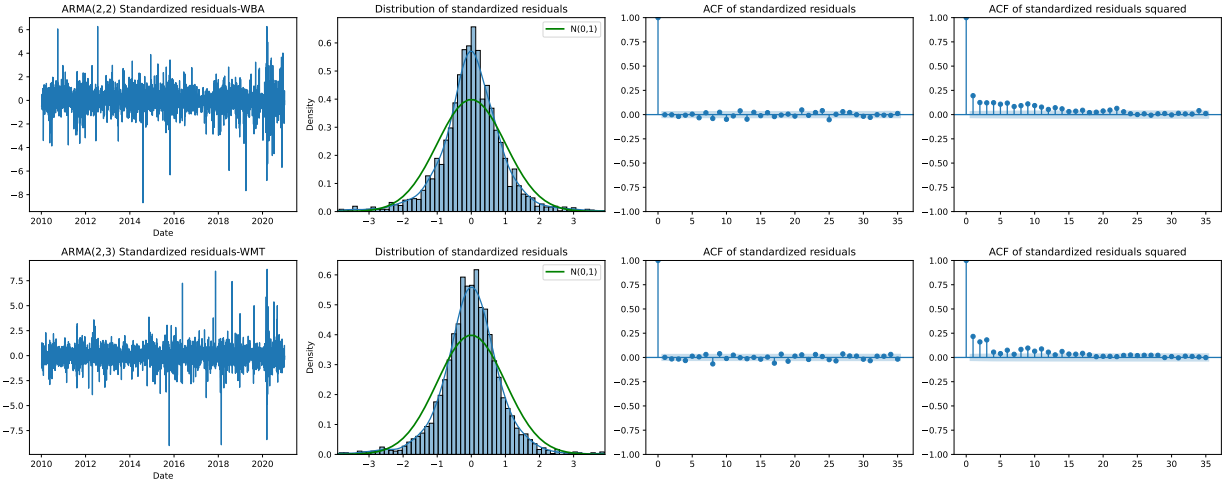
[1] DOW only started trading on 20/03/2019

2.2: Write code to plot a 2-by-4 subplot figure that includes the following diagnostics for the best-fitted ARMA model found in Task **2.1**:

Row 1: (i) Time series plot of the standardized residuals, (ii) histogram of the standardized residuals, fitted with a kernel density estimate and the density of a standard normal distribution, (iii) ACF of the standardized residuals, and (iv) ACF of the squared standardized residuals.

Row 2: The same subplots for the second stock.

Your figure should look similar to the following for your sample of stocks. Comment on what you observe from the plots. (**6 marks**)



2.3: Use the same in-sample data as in Task **2.1**, write code to find the best-fitted AR(p)-GARCH(p^*, q^*) model with Student's t errors for returns of each stock that minimizes AIC, where p is fixed at the AR lag order found in Task **2.1**, and p^* and q^* are no greater than 3. Print the best-fitted AR(p)-GARCH(p^*, q^*) output and a statement similar to the following for your stock sample.

```
Best-fitted AR(p)-GARCH(p*,q*) model for WBA: AR(2)-GARCH(1,1) - AIC
= 10137.8509
Best-fitted AR(p)-GARCH(p*,q*) model for WMT: AR(2)-GARCH(3,0) - AIC
= 7743.4547
```

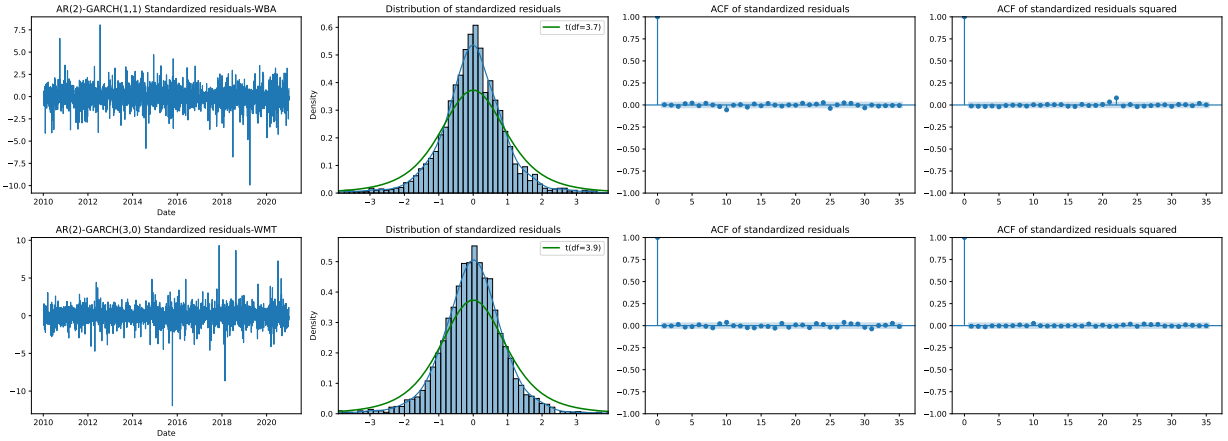
(**5 marks**)

2.4: Write code to plot a 2-by-4 subplot figure that includes the following diagnostics for the best-fitted AR-GARCH model found in Task **2.3**:

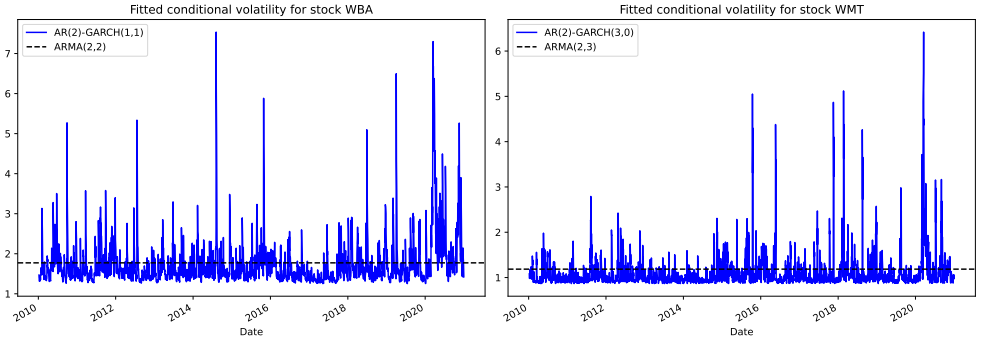
Row 1: (i) Time series plot of the standardized residuals, (ii) histogram of the standardized residuals, fitted with a kernel density estimate and the density of a fitted Student's t distribution, (iii) ACF of the standardized residuals, and (iv) ACF of the squared standardized residuals.

Row 2: The same subplots for the second stock.

Your figure should look similar to the following for your sample of stocks. Comment on what you observe from the plots. (**6 marks**)



2.5: Write code to plot a 1-by-2 subplot figure that shows the fitted conditional volatility implied by the best-fitted AR(p)-GARCH(p^*, q^*) model found in Task **2.3** against that implied by the best-fitted ARMA(p, q) model found in Task **2.1** for each stock in your sample. Your figure should look similar to the following.



(**3 marks**)

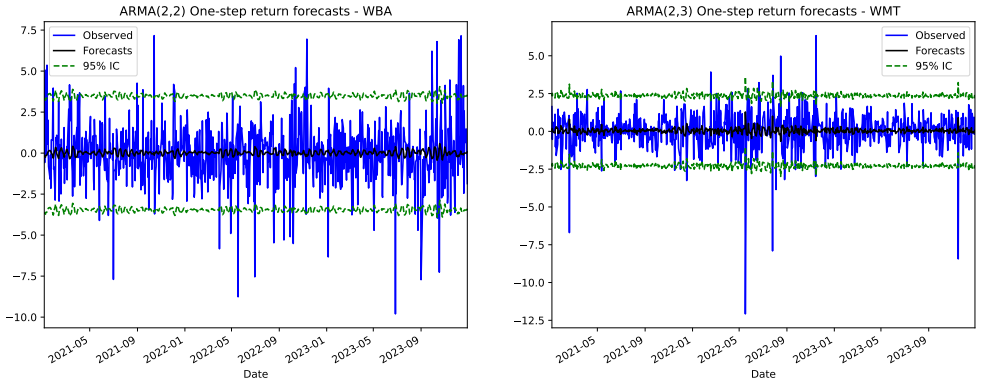
Task 3: Return-Volatility Forecasting (Σ = **25 marks**)

3.1: Use data between 01/01/2021 and 31/12/2023 as out-of-sample data, write code to compute **one-step forecasts**, together with 95% confidence interval (CI), for the returns of each stock using the respective best-fitted ARMA(p, q) model found in Task **2.1**. You should extend the in-sample data by one observation each time it becomes available and apply the fitted ARMA(p, q) model to the extended sample to produce one-step forecasts. Do **NOT** refit the ARMA(p, q) model for each extending window.[2] For each stock, the forecast output is a data frame with 3 columns `f`, `fl` and `fu` corresponding to the one-step forecasts, 95% CI lower bounds, and 95% CI upper bounds. (**5 marks**)

3.2: Write code to plot a 1-by-2 subplot figure showing the one-step return forecasts found in Task **3.1** against the true values during the out-of-sample

[2] Refitting the model each time a new observation comes generally gives better forecasts. However, it slows down the program considerably so we do **not** pursue it here.

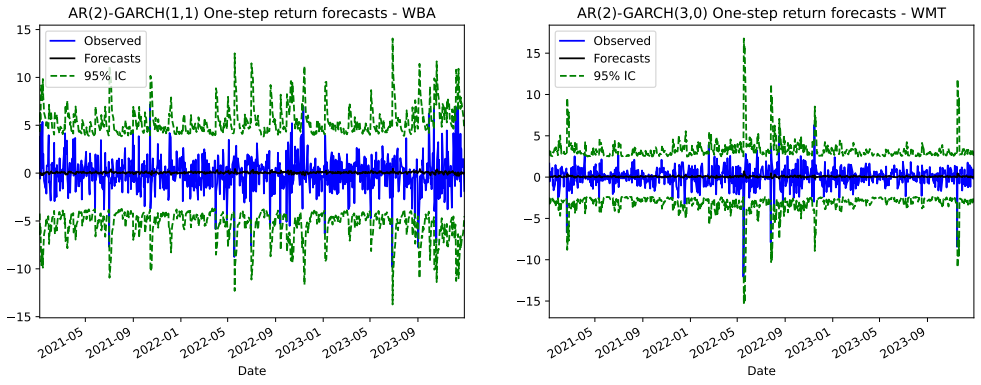
period for both stocks in your sample. Also show the 95% confidence interval of the return forecasts. Your figure should look similar to the following.



(**3 marks**)

3.3: Write code to produce **one-step analytic forecasts**, together with 95% confidence interval, for the returns of each stock using respective best-fitted AR(p)-GARCH(p^*, q^*) model found in Task **2.3**. For each stock, the forecast output is a data frame with 3 columns `f`, `fl` and `fu` corresponding to the one-step forecasts, 95% CI lower bounds, and 95% CI upper bounds. (**4 marks**)

3.4: Write code to plot a 1-by-2 subplot figure showing the one-step return forecasts found in Task **3.3** against the true values during the out-of-sample period for both stocks in your sample. Also show the 95% confidence interval of the return forecasts. Your figure should look similar to the following.



(**3 marks**)

3.5: Denote by $e_{t+h|t} = y_{t+h} - \widehat{y}_{t+h|t}$ the h-step forecast error at time t, which is the difference between the observed value y_{t+h} and an h-step forecast $\widehat{y}_{t+h|t}$ produced by a forecast model. Four popular metrics to quantify the accuracy of the forecasts in an out-of-sample period with T' observations are:

1. Mean Absolute Error: MAE $= \dfrac{1}{T'}\sum_{t=1}^{T'} |e_{t+h|t}|$
2. Mean Square Error: MSE $= \dfrac{1}{T'}\sum_{t=1}^{T'} e_{t+h|t}^2$
3. Mean Absolute Percentage Error: MAPE $= \dfrac{1}{T'}\sum_{t=1}^{T'} |e_{t+h|t}/y_{t+h}|$
4. Mean Absolute Scaled Error: MASE $= \dfrac{1}{T'}\sum_{t=1}^{T'} \left| \dfrac{e_{t+h|t}}{\frac{1}{T'-1}\sum_{t=2}^{T'} |y_t - y_{t-1}|} \right|$.

The closer the above measures are to zero, the more accurate the forecasts. Now, write code to compute the four above forecast accuracy measures for one-step return forecasts produced by the best-fitted ARMA(p,q) and AR(p)-GARCH(p^*,q^*) models for each stock in your sample. For each stock, produce a data frame containing the forecast accuracy measures of a similar format to the following, with columns being the names of the above four accuracy measures and index being the names of the best-fitted ARMA and AR-GARCH model:

	MAE	MSE	MAPE	MASE
ARMA(2,2)				
AR(2)-GARCH(1,1)				

Print a statement similar to the following for your stock sample:

```
For WBA:
Measures that ARMA(2,2) model produces smaller than AR(2)-GARCH(1,1)
model:
Measures that AR(2)-GARCH(1,1) model produces smaller than ARMA(2,2)
model:  MAE, MSE, MAPE, MASE.
```

(**5 marks**)

3.6: Using a 5% significance level, conduct the Diebold-Mariano test for each stock in your sample to test if the one-step return forecasts produced by the best-fitted ARMA(p,q) and AR(p)-GARCH(p^*,q^*) models are equally accurate based on the three accuracy measures in Task **3.5**. For each stock, produce a data frame containing the forecast accuracy measures of a similar format to the following:

	MAE	MSE	MAPE	MASE
ARMA(2,2)				
AR(2)-GARCH(1,1)				
DMm				
pvalue				

where 'DMm' is the Harvey, Leybourne & Newbold (1997) modified Diebold-Mariano test statistic (defined in the lecture), and 'pvalue' is the p-value associated with the DMm statistic. Draw and print conclusions whether the best-fitted ARMA(p,q) model produces equally accurate, significantly less accurate or significantly more accurate one-step return forecasts than the best-fitted AR(p)-GARCH(p^*,q^*) model based on each accuracy measure for your stock sample.

Your printed conclusions should look similar to the following:

```
For WBA:
Model ARMA(2,2) produces significantly less accurate one-step return
forecasts than model AR(2)-GARCH(1,1) based on MAE.
Model ARMA(2,2) produces significantly less accurate one-step return
forecasts than model AR(2)-GARCH(1,1) based on MSE.
Model ARMA(2,2) produces significantly less accurate one-step return
forecasts than model AR(2)-GARCH(1,1) based on MAPE.
Model ARMA(2,2) produces significantly less accurate one-step return
forecasts than model AR(2)-GARCH(1,1) based on MASE.
```

(**5 marks**)

Task 4: (Σ = **20 marks**)

These marks will go to programs that are well structured, intuitive to use (i.e. provide sufficient comments for me to follow and are straightforward for me to run your code), generalisable (i.e. they can be applied to different sets of stocks (2 or more)) and elegant (i.e. code is neat and shows some degree of efficiency).