QR Development Test

Please create a github private repository to upload your code, answers and discussions, and kindly share your repository to our company account sfshipping for submission.

1. The ARMA (AutoRegressive Moving Average) model is a simple yet popolar econometric model for stationary discrete time series. In this question, you will be asked to implement the simulation and analytics of a few basic ARMA models in Python 3.9+.

   Submission requirements:

   a. Please do create a conda environment, install all packages needed, and export your enviroment configurations to a .yml file, and export your package dependences to a requirements.txt file.

   b. Please show your code and answers in one Jupyter notebook, while dependencies on any local files are acceptable.

   (1) Consider an AR(1) model:

   $$X_t = \varphi X_{t-1} + \varepsilon_t$$

   where ε_t is a white noise process with zero mean and constant variance σ_ε^2.

   Your first task is to simulate *N*-timestep time series of *X*, with your own choices of the parameter values (*N* is not less than 5000). Please create a .json file to pre-specify your parameters, and your code should load parameter values from the .json file. The output is a one-column pandas.DataFrame where the index is datetime objects and the column is your simulated time series. Suppose the initial time is pandas.Timestamp("2000-03-06 09:30:00"), you should create the following *N - 1* timestamps. We require that consecutive timestamps should have 1-minute intervals, and take values only from 9:30am to 3:00pm on buisness days. In this task, you should only use pandas and numpy for computation.

   (2) Simulate 20 time series as above, make a plot of these time series with the horizontal axis labelled by datetime.

   (3) Using the ARIMA class by statsmodels (or any packages you are familiar with) to estimate parameters from your simulated time series. Compare with ground truth values which you pre-specified in the .json file. What would happen if the timestep *N* increases? Can you should some empirical evidences?

2. The attached `pv.csv` file contains daily price and volume time series data for an asset from 1st Jan, 2020 to 10th Jan, 2023.

   (1) Pick your desired libraries to properly load the daily data and summarise the data provided by calculating analytics of the asset performance in tables, summary statistics or charts, etc.

   (2) Predict the prices for the last 20% of the days by developing THREE apporaches with the data, test the accuracy and repeat the tests with different (hyper-)parameters if necessary to improve the result.

   The three approaches are:

   a. Moving average - set the current closing price as the mean of the closing prices of the previous *N* days.

   b. Linear regression - fit a linear regression model to the previous N values, and use the model to predict the value for the test days.

   c. A machine learning or deep learning model of your own choice.

   You are required to visualize the price prediction process. Please show your code and answers in one Jupyter notebook.