BEMM458 - Programming for Data Analytics Final Assignment

December 2022

Assignment Submission Deadline: 9th January 2023.

INSTRUCTIONS – PLEASE READ CAREFULLY

Software required: A working Python interpreter with the Pandas, Numpy, Seaborn, Matplotlib and Statsmodels code libraries.

Your task is to estimate and analyse a simple model of the US economy. The necessary data is supplied to you in three csv files, as follows:

1. The file '`2021-12.csv`'. The file containing 127 monthly US economic time series statistics from Mathew McCraken's FRED-MD project (https://research.stlouisfed.org/econ/mccracken/fred-databases/).
2. The file '`fred_md_desc.csv`' – a file containing a variety of summary data for the FRED-MD data.
3. '`NBER_DATES.csv`' – a table containing the dates of US economic recessions and expansions as identified by the US National Bureau of Economic Research (https://www.nber.org).

In addition, you are given a 'template.py' file which you can use as a template to develop your code. This file contains a function which you will need to call to complete your analysis.

You are asked to estimate a simple economic model using Principal Components Analysis (PCA). You DO NOT need to understand or explain the workings of the PCA algorithm for the purposes of the assignment, but you will need to format and prepare the data in a suitable way (as specified below), pass the formatted data to the algorithm, and extract the output. PCA is a data compression technique which produces a factor or 'index' at each point in time, summarising several input variables. This approach is based on the insight that changes in many economic time series are primarily driven by changes in overall economic conditions. This technique is widely used in economic analysis and forecasting (See for example: Stock, James H., and Mark W. Watson. 1998. "Business Cycle Fluctuations in U.S. Macroeconomic Time Series." https://doi.org/10.3386/w6528.).

The output of your model is then used to build a regression model for five economic time series, also extracted from the database. Once more you do not need to explain the regression algorithm for the purpose of this test, but you do need to be able to call it correctly with the appropriate 'x' and 'y' data, and perform a basic assessment of the adequacy of the model (how well do the model values correspond to the data?)

You are asked to produce a pdf file, summarising:

1. Your brief assessment of the performance of the model on each of the five series.

2. Your approach to the problem and a description of the process you adopted to complete the task.

Consult the online documentation for each package if you need to clarify any of the commands mentioned below.

Please write a Python script to load the data file and complete the calculations set out below. The questions specify the filename and contents required for your output, and your script should generate each file as described. Write your script, test it and then upload as '`final_exam.py`'.

PLEASE NOTE VERY CAREFULLY:

1. The script should output files in the directory from which it is run (i.e. default Python behaviour). DO NOT include operating system file paths when reading and writing files in your script. These will cause your code to crash when run on another computer and you will gain very limited marks.

2. PLEASE BE SURE TO SUBMIT A ZIPPED PYTHON SCRIPT ('final_exam.py') FILE AND NOT AN IPYTHON NOTEBOOK FILE ('.ipynb')

3. Your work will be assessed chiefly on the accuracy and completeness of the output generated by the script you submit. It is your responsibility to make sure that your script files run without error and produce the output requested, and with the specified file names. Scripts which do not run correctly are unlikely to achieve sufficient marks to pass the assignment. You have access to the development tools covered in the course to develop and test your scripts. No attempt will be made to fix non-functioning code during the marking process.

QUESTIONS

1. Load the data from the '2021-12.csv' file and set the file index to an appropriate Pandas date format. All the variables of interest are stored in the columns of the table.
2. Select the data up to and including December 2019 and perform the following analysis on this subset of the data.
3. Load the data description table: '`fred_md_desc.csv`' file. The keys in this table should match the columns of (1).
4. In order to fit the model, the data needs to be mathematically transformed in several ways. The transformation required for each series and the corresponding code to compute these are set out below. The appropriate transformation for each variable is supplied in the '`tfcode`' field in the description table. The table below gives the code required to transform the data for any given Pandas Series stored in the variable '`srs`':

Tfcode	**Description**	**Pandas Code**
0	No transformation	`srs`

1	1st differences	`srs.diff()`
2	2nd differences	`srs.diff().diff()`
3	Log	`np.log(srs)`
4	Log 1st differences	`np.log(srs).diff()`
5	Log 2nd differences	`np.log(srs).diff().diff()`
6	Percent Change	`(srs/srs.shift(1) – 1)`

5. Produce a new data frame (with a correct pandas time series index and the original column names) by applying the specified mathematical transformation to each variable in the data.
6. Standardise each transformed variable from (Q5) above by subtracting its own time series mean and then dividing by its own time series standard deviation. Fill any missing values in the standardised data with a zero. Write this table to file (‘`transformed_data.csv`’) with a correct Pandas time series index and using the original variable names as column names.
7. Using the standardised data from (Q6) produce a PCA analysis. Call the function (provided for you in the template.py file) `pca_function()`, passing in your standardised dataset from Q6.
8. Produce a histogram of the distribution of the factor, and a plot of the time series of the factor (both in one figure) and save this to a pdf file (‘`factor.pdf`’). Note that the command `plt.tight_layout()` can be called after you have drawn your plot and added a title so as to neatly arrange the axis/ titles so that they do not overlap.
9. Produce a new data frame of 1st lags of the data by shifting your transformed (but not standardised) data from (Q5) forward in time by one time period using `df.shift(1).`
10. You are asked to analyse the following five variables `['INDPRO', 'S&P 500', 'PAYEMS', 'CPIAUCSL', 'BUSINVx'].` For each variable produce a regression model using Statsmodels (`sma.OLS()`), regressing:

    $$y(t) = a + y(t-1) + f(t-1) + e(t)$$

    Where a is a constant, y is the transformed series at time t, y(t-1) is the transformed series at time t-1 (i.e. the first lag produced in Q9) and f(t-1) is the factor at time t-1, and e(t) is a zero mean error term. (You will need to shift the factor time series forward by one time period). Fit the model and capture the fitted values.
11. Produce a data frame of the fitted values from your 5 models, with their variable names as columns, indexed by your Pandas time series index from above and save to file as ‘`fitted_values.csv`’.
12. Produce for each of the variables a Seaborn plot (for example use `lmplot`) showing a scatter plot of the transformed variable and its fitted value. Differentiate the plot between periods of time identified by the NBER as recessions and expansions (obtain this data from the ‘`NBER_DATES.csv`’ file) (use the ‘cols’ argument of `lmplot` to produce two plots for each variable, one for recessions and one for expansions. Use Matplotlib to give your figure a title (use `fig.suptitle(‘my title’)`) containing the description of each variable from the descriptions table, and a description of the transform applied from the table above. Save each to a file

‘`srs.pdf`’ where `srs` is the key for each series. (Note that if `plot = sns.lmplot(…)` then `plot.figure` returns a Matplotlib figure.)

13. Produce a single pdf file (using software of your choice) containing the following two sections:
    a. Analytical Results - The charts your software has generated. Comment (briefly) on each of the charts for the variables, noting whether the model provides a reasonable fit for the data (does the model output correlate positively with the data?) and whether it is appropriate in recessionary and/ or expansionary periods. Save your summary document as ‘`Summary.pdf`’ and upload this pdf file as part of your answer (Maximum 250 words).
    b. Development Process - In part 2 of the same pdf document, describe the process you adopted to write, debug and check your code. Mention the tools you used and summarise briefly the stages of your analysis. Please add screenshots to illustrate this where you think this is helpful, and reference / document any external resources you have used (software documentation, academic research / books etc). How might you develop / improve your code so that it might be used in future by other analysts? (Maximum 750 words).

END OF PAPER