Take Home Exam - Applied Time Series Analysis 2024

- This exam consists of 2 problems. Please do them all. The grade you get for the take home counts 30% of the final grade.
- Please upload your solutions to the 'Übungsmodul' on ILIAS no later than June 24, 2024, 11.30 am. Your solutions should contain **one** pdf-file and **one** zip-file containing your code (Python or Matlab) used for the solutions. Late submissions will not be considered!
- The pdf with your solutions should be in form of a short research report. Hand in a text with your graphs, outputs and answers. Your solutions may contain scanned versions of handwritten notes (no need to typeset). Make sure that we understand what you have done, that is, briefly explain your answers and modeling decisions and include all relevant computer output. Consistently number all tables and graphs and refer to these results accordingly. When presenting results from statistical tests, make sure to state the null hypothesis, the appropriate distribution of the test statistic (under H_0), the value of the test statistic, the critical value (or p-value) and the test decision. Be as concise as possible nonetheless. You are allowed to hand in at most 10 pages (including graphs and tables).
- Note: Submitting only Python/Matlab code with answers to questions as text comments within the code is not acceptable.
- We recommend to use the programming language Python for the computations in all problems.
- Unless specified otherwise, for standard methods you may use functions readily available in Python/Matlab. You may, of course, reuse program code that you have written for the programming assignments.
- On your solutions and code, please give your student number not your name.
- Policy with regard to academic dishonesty: The grade for your take home exam will be a part of your overall grade for this course. Students who wish to work together on the take home material may do so. However, **each student must formulate and hand in the solutions independently**. This means that students who have worked as a group may **not** simply hand in the same solution.
- Good luck!

---

1) In this problem you explore the properties of different time series and evaluate some time series models.

   a) The file `NGDPRSAXDCGBQ.csv` contains data on seasonally adjusted real UK GDP for a period from 1983Q1 - 2023Q3.[1] Compute and plot the logarithm, the annualized quarter-over-quarter growth rates, and the year-over-year (annual) growth rates of the series in %. Describe (using words) the main features of the time series.

   Use data from 1983Q1 to 2019Q4 for the following problems (unless noted otherwise).

[1] Data source: FRED, https://fred.stlouisfed.org

b) Let y_t denote the annualized quarter-over-quarter growth rate of real GDP. Plot SACFs and SPACFs for y_t together with $\pm 2/\sqrt{T}$ bounds. Based on your graphs, suggest a pure MA model. Explain your choice briefly.

c) Apply information criteria to select a pure AR model for y_t (use models with an intercept, $p_{max} = 8$). What are the suggested lag orders?

d) Estimate an AR(2) with intercept for y_t using the LS method. Report the parameter estimates together with asymptotic standard errors.

e) For the estimated AR(2) in d) give plots of standardized residuals and residual ACFs and PACFs. Conduct diagnostic tests for remaining residual autocorrelation (use modified Portmanteau tests, LM and LMF tests). Also perform a non-normality test for the estimated model. Describe your results and comment on the adequacy of the model.

f) Estimate the AR(2) model from d) using exact ML and report the MLE estimates of c, α_1, α_2 and σ^2 together with their asymptotic standard errors. **Note: For this part you have to program the ML estimator on your own**. This involves setting up the exact log-likelihood function and using an optimizer.[2] What is the value of the maximized log-likelihood function? Do you get the same parameter estimates as from Python's statsmodels ARIMA function? What is the estimated mean based on the AR(2)?

Use data from 1983Q1 to 2023Q3 for the following problem.

g) Estimate the AR(2) model from d) over the sample period ending in 2023Q3. How do the estimated autoregressive parameters change compared to those from d)? Also provide a residual plot. Your fellow researcher things that the results are driven by unusually small and large growth rate observations in the extented sample. Therefore, she suggests to create 'impulse dummy variables' for the quarters 2020Q1, 2020Q2, 2020Q3, 2021Q1, 2021Q2. Each impulse dummy takes on value 1 in the respective quarter and is zero elsewhere. Reestimate the AR(2) with the impulse dummies and report estimates and a standardized residual plot. Interpret the sign of the estimated coefficients on the dummy variables. How does the inclusion of the dummy variables affect the estimates of the AR coefficients?

2) In this problem you analyze the properties of unit root tests by a Monte Carlo simulation. **The KPSS test in this task must be programmed on your own**. However, for the Augmented Dickey-Fuller test you may use the built-in Python function.

a) Consider the data generating process $y_t = c + \varepsilon_t, \quad t = 1, \ldots, T, \quad \varepsilon_t \overset{iid}{\sim} N(0, 1)$.

   i. Explain why simulating from this process can be thought of as simulating a time series under the H_0 of a KPPS test.

   ii. Generate $M = 2000$ sets of time series of length $T = 5000$ using $c = 1$. For each time series, conduct a KPPS test with a constant and lag truncation $l_{12} = [12(T/100)^{1/4}]$ and store the value of the test statistic. Report the 0.9, 0.95, and 0.99 quantiles of the test statistic distribution.

   iii. Relate the reported quantiles to the asymptotic critical values of the KPPS test.

b) Repeat problems 2c) to 2f) for the following values of α_1, α_2 and T

$$(\alpha_1, \alpha_2) \in \{(0.4, 0.2), (0.5, 0.2), (0.6, 0.2), (0.7, 0.2), (0.8, 0.2)\}$$
$$T \in \{50, 100, 200, 500\}$$

[2] The log-likelihood function for a Gaussian AR(2) can be found in equation 5.3.8 of Hamilton's (2004) textbook. For this part, the Python functions for ARMA models should only be used for checking your results.

c) Simulate for a given α_1, α_2 and T, $M = 2000$ sets of time series from the DGP

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \varepsilon_t, \tag{1}$$

where $y_{-1} = y_0 = 0$, and $\varepsilon_t \overset{iid}{\sim} N(0,1)$.

d) For each of your generated M time series, conduct a KPSS test with a constant at the 5% level. Use the lag truncation parameter l_q from the lecture. For both $q = 4$ and $q = 12$, record how often the H_0 is rejected in the M replications and compute the empirical rejection frequency (number of rejections/M).

e) For each of your generated M time series, conduct an Augmented Dickey Fuller test at the 5% level with a constant (Case 2) and one lagged difference. Record how often the unit root hypothesis is rejected in the M replications and compute the empirical rejection frequency (number of rejections/M).

f) For each of your generated M time series, conduct an Augmented Dickey Fuller test at the 5% level using the test regression $\Delta y_t = c + \phi y_{t-1} + \sum_{i=1}^{p-1} \alpha_i^* \Delta y_{t-i} + \varepsilon_t$, where the number of lagged differences $p - 1$ is determined using the information criterion BIC. Allow between 0 and 4 lagged differences. Record how often the unit root hypothesis is rejected in the M replications and compute the empirical rejection frequency (number of rejections/M).

g) Report for each T and different values of α_1 the rejection frequencies of the two KPSS test variants and of the two ADF test variants in a table. For the KPPS and the ADF tests, explain which of the rejection frequencies correspond to the empirical size and power of the test. What size values would you expect in an ideal test? How does the empirical size values compare against this?

h) For each T provide a plot showing different values of α_1 on the x-axis and the rejection frequencies of each of the 4 tests on the y-axis. Interpret your results: Which of the KPSS test variants is doing better? Which variant of the ADF test is doing better? Briefly describe any differences between the ADF test and the KPSS test in terms of correct unit root detection.