Midterm Exam

Econ 6376 — Time Series Econometrics

Parameters

What this is. An expanded problem set crossed with a short paper. It is *not* a research paper — I am not grading the originality of your question, and you are not expected to produce a novel finding. I am grading whether you can execute this course's methods correctly on data you chose yourself, and explain what you did to a reader who does not know time series analysis. A correct, plainly written analysis of a boring question beats a muddled analysis of an exciting one.

This midterm sets up the rest of your semester. The dataset you build, the split you fix, and the forecasts you produce here are inputs to Problem Sets 3, 4 and 5 — you will reload the exact files you submit. Choose your series with that in mind. You will live with them until December.

Treat this like a professional report for someone who is not familiar with time series analysis. All answers should be concise but as informative as possible. All graphs and tables should be legible and labeled. If you wouldn't turn it in to your boss, don't turn it in to me.

- **Limit your report to 15 pages of body.** Your reference list and any code appendix do not count toward that. Do not paste code into the report — it belongs in `analysis.R`. A figure that is not referenced in the text costs you space and earns nothing.
- Submit through Blackboard by 11:59pm on the due date listed there. Late submissions **WILL NOT** be accepted unless previously arranged.
- I am happy to answer questions at any point.

Use of AI

AI assistance is permitted on this exam, as it is on every graded deliverable in this course. What gets graded is understanding you can defend.

End your report with a short, free-form paragraph headed **AI Use Disclosure** describing how you used it — which tools, for what, and where you checked its work. It does not count toward your 15 pages and it is not graded on its content. A submission without it is incomplete, in the same way a submission without `analysis.R` is incomplete.

What to submit

Four files, through Blackboard.

File	What it must contain
Report (PDF or Word)	The write-up. Fifteen pages of body, ending with the **AI Use Disclosure** paragraph.
`data.csv`	Your four series, one row per date, plus a `block` column taking the values `train`, `weights`, and `holdout`.
`forecasts.csv`	One row for each of the twelve dates in the `weights` block — `(T+1):(T+12)`. Twelve rows, not twenty-four: the forecasts in Question 7 are one step ahead, and a one-step forecast for $T+14$ would require you to have observed y_{T+13}, which sits inside the holdout. The weight window is the most this exam can legitimately produce. Columns: `date`, `arima_1`, `arima_2`, `arima_3`, `locf`, `avg_k`, `meanf`. **Forecasts only — no actuals.**
`analysis.R`	Organized, commented, and reproducible top to bottom. It must produce every number and every figure in your report.

`forecasts.csv` carries no actuals on purpose. The actuals are already in `data.csv`, and the two files get joined at scoring time — by you when you answer Question 8, and by me when I grade it. Storing the answers next to the forecasts is how a forecast file quietly stops being a forecast file.

Grading

Total: 100 points. Every row below is one numbered question, plus two rows that cut across the whole report.

Row	Points
Question 1 — Question and motivation	5
Question 2 — Literature review	8
Question 3 — Data	19
Question 4 — Stationarity	8
Question 5 — Model selection	22
Question 6 — Serial correlation	6
Question 7 — Forecasting	12
Question 8 — Forecast evaluation	7
Report craft	7
Code and reproducibility	6

Every graded row ends with a blockquote headed **What earns the points here**. Read them. They are the difference between full marks and half marks, and they are the most useful thing in this document.

The Midterm

Think of a question you would like to explore that involves four or more time series. For example, you may want to explore the relationship between GDP, unemployment, inflation, and the money supply. Alternatively you could think of something less traditional, such as the relationship between search history for a particular term, sales of a particular product or products, and advertising expenditures. Be creative!

Once you have your question, work through the eight questions below in order. They build on each other, and the files you produce along the way are the files you will reload in Problem Sets 3, 4 and 5.

Question 1 — Question and motivation

Write a few paragraphs explaining your question and why you find it interesting. What do you expect to learn from it? Name the four or more time series you will use and say, in a sentence each, why that series belongs in this question.

> **What earns the points here.** Full marks: the question is stated in one sentence that a reader could repeat back to you, the four series are obviously the series that question needs, and you have said what you expect to find and why you expect it. Half marks: two pages of background on the topic with the question itself never actually stated, or four series that are really a list of whatever was easy to download, retrofitted to a question afterward. I am not grading whether your question is important. I am grading whether it is a question, and whether your data can speak to it.

Question 2 — Literature review

Search for academic articles that have already explored your question, or a related question, or that use the data you are using.

- Write a brief literature review of the articles you found. One paragraph per article is plenty.
- Cite your articles properly. Pick a style — APA, Chicago, whatever you like — and use it consistently.

> **What earns the points here.** Full marks: each article gets a paragraph that says what the authors did, what they found, and what it has to do with *your* question — including the case where a paper reached a conclusion you do not expect to reproduce. The reference list is complete and consistently formatted. Half marks: paragraphs that restate abstracts and never connect back to your analysis, a bare list of URLs, or a "literature" consisting of blog posts and the FRED series description page. If nothing in this section would change how you read your own results, you have written a bibliography, not a review.

Question 3 — Data

Identify your data sources and obtain each series. Before you download anything, three requirements:

- You need **four or more** time series.
- Each series must have **at least 124 observations** — 100 to train on and 24 held out. If you are planning a seasonal model, more is better: 124 is a floor, not a target. A seasonal ARIMA fit on 100 monthly observations is fitting eight and a third cycles.
- All series must be at the **same frequency**, and you should **avoid stocks and indices**, as they are typically random walks.
- Where a source offers both, **prefer the series in levels** over a pre-differenced or growth-rate version. You can always difference a series yourself; you cannot un-difference one.

Part A — The data table

Produce one table with one row per series and exactly these columns:

Series	Source & series ID	Frequency	Span	N	Units	SA?	Transformation	Accessed
one row per series								

Nine columns is wide, and it will not sit comfortably in portrait at body font size. Set the page landscape, or split it into two stacked tables, or shrink the font — but **do not drop columns**.

The **Accessed** column is not bookkeeping. FRED and every other statistical agency revise published numbers, sometimes years after the fact. The file you submit with this midterm is frozen, and Problem Sets 3, 4 and 5 reload it. The access date is what makes the midterm dataset and the problem-set dataset the same dataset.

So: do not re-pull your data for those problem sets. Your `data.csv` *is* a vintage — a specific snapshot of what the agency believed on a specific day. If you download the same series ID again in December you will get a different set of numbers, and every result in your midterm will stop reproducing.

Part B — What the numbers are

Write one short paragraph per series covering the following, then a closing paragraph.

1. **The measurement regime.** Who produces this number, under what framework or standard, using what instrument. "The BEA publishes GDP" is not an answer. GDP is constructed under the System of National Accounts, and that framework is what decides what counts as output — which is why household production sits outside the boundary and imputed rent sits inside it. Your reader should come away knowing what your series is a number *of*, and what rulebook decided that.

2. **Revision and vintage.** On what schedule is this series revised? Have the definitions or classifications changed *inside your sample* — a rebasing, a new NAICS vintage, a methodology change, a redesigned survey instrument? A definitional break inside the sample means the early part and the late part measure different things, and your model has no way to know that.
3. **Documented limitations.** Imputation, suppression, top-coding, sampling error, coverage gaps. Cite these with a URL **plus** a section or page number — I want to see that you opened the technical documentation, not the landing page.
4. **Commensurability.** Frequency, seasonal-adjustment status, calendar alignment, units. Are your four series actually comparable to each other, and if you had to convert one to make them comparable, what did the conversion do?
5. **Closing paragraph — consequence.** Pick one specific item from above and say the specific thing it does to your analysis. Not "data quality matters." One item, one consequence.

> **What earns the points here.** Full marks: the table is complete — every cell filled, every series ID exact enough that I could pull the identical series myself, every access date real. The prose tells me what each number is a number *of* and which rulebook decided that, and it cites the technical documentation by section or page. The closing paragraph names one specific limitation and one specific thing it does to *your* analysis. Half marks: a table with "FRED" in the source column and blanks under Units and SA, plus a paragraph per series that names the publishing agency and stops. Vaguely gesturing at "possible measurement error" earns nothing; naming the 2018 NAICS revision that sits in the middle of your sample earns everything. If you cannot tell me what your series would do differently under a different rulebook, you have not answered this question. Nineteen points sit here because this is the question most students skip, and it is the one that decides whether the next five questions are about anything.

Question 4 — Stationarity

Define **strict stationarity** and **weak stationarity**. Then determine whether each of your time series is stationary. Use all available evidence to support your conclusion.

Characterize the persistence of each series while you are here — how fast the autocorrelation decays, how far back the memory runs. You will need that characterization again in Question 7.

> **What earns the points here.** Full marks: both definitions stated correctly and in your own words, with the relationship between them made explicit. A call — stationary or not — for each of your four series, each supported by more than one kind of evidence: the time plot, the correlogram, and a formal test. Where those disagree, you say so, and you say how you weighted them. Half marks: correct definitions followed by four ADF p-values and nothing else, or four time plots with no test at all, or a stationarity call here that quietly contradicts the differencing you use in Question 5. A test result is evidence. It is not a verdict, and the correlogram is not decoration.

Question 5 — Model selection

The split

Fix the following three-block split of your series and use it for the rest of this exam and for Problem Sets 3, 4 and 5. Let N be the number of observations in your series — the same N you reported in the Part A table — and let

$$T = N - 24$$

be the **forecast origin**: the last observation any model on this exam is permitted to be *estimated* on, and the first date you forecast from. Every T in this document means the origin, never the length of your file.

```
|<---------- train ---------->|<-- weights -->|<-- holdout -->|
1                            T  T+1       T+12 T+13        T+24
     fit every model here        score here     DO NOT TOUCH
                             ^
                  forecast origin T = N-24
```

- **train** — observations `1:T`. Every model on this exam is estimated here and nowhere else.
- **weights** — observations `(T+1):(T+12)`, twelve observations. You will score your forecasts on this window in Question 8. On Problem Set 3, you will fit forecast-combination weights on it.
- **holdout** — observations `(T+13):(T+24)`, twelve observations, ending at N. **Do not touch this window on this midterm.**

The holdout actuals are necessarily sitting in your `data.csv`, so on the face of it this instruction is on the honor system. Under the forecasting design in Question 7 it is close to self-enforcing instead. Every forecast you produce there is one step ahead, which means the only way to forecast across the holdout is to feed the model holdout values as they arrive — there is no legitimate route to a holdout number that does not begin with opening the window. You are not being asked to resist a temptation so much as to not go looking for one.

The stated price stands as a backstop. **Reporting any loss, error, or accuracy figure computed on the holdout window costs you points under Question 8.** The holdout is Problem Set 3's scoring window. Spending it now means that in October you will be evaluating combined forecasts against data you have already seen, and neither of us will be able to trust the answer.

The modeling

Choose **one** of your time series. Model the training block using the Box-Jenkins methodology (ARIMA).

- Evaluate the ACF and PACF of the **stationary** data to determine what model to start your analysis with. Be specific about why you choose your starting point.
- Estimate the starting model and move from **general to specific** within the ARIMA framework.

- Provide a table of **at least three different models**, including your "best" model as indicated by your information criteria.
- Define the model selection criteria you use and describe their relative strengths and weaknesses.
- Be detailed in explaining your decision process.

Keep all three models. You will forecast with every one of them in Question 7.

> **What earns the points here.** Full marks: the split is stated and honored, the ACF/PACF reading produces a specific starting point for a specific stated reason, the table has at least three fitted models scored on criteria you have defined, and the prose walks me from the starting point to the winner one decision at a time. I want to see a model you rejected and the reason you rejected it. I want to know whether your criteria agreed, and what you did when they did not. Half marks: `auto.arima()` output pasted in with a sentence reporting that it picked (1,1,1), or a tidy three-row table with no account of how those three were chosen out of everything you could have fit, or information criteria reported as numbers with no statement of what they penalize. This is the largest question on the exam because the reasoning *is* the answer here — the winning specification is almost incidental.

Question 6 — Serial correlation

Define serial correlation and explain how to test for it. Include the appropriate maths in your explanation. Test your models for serial correlation and discuss the results and how they bear on the choice of "best" model.

> **What earns the points here.** Full marks: a definition, the test statistic written out with its null hypothesis, its distribution, and its degrees of freedom — including how the degrees of freedom change once the residuals come from a fitted ARIMA rather than raw data. The test is then run on **every** model in your Question 5 table, and you say what you do when the IC winner has correlated residuals and a runner-up does not. Half marks: `checkresiduals()` output dropped in with no maths and no reading of it, or the maths written out correctly and never applied to your own models, or a passing test reported as though it proved the model correct. The point of running a test is that the result can change your answer. If it could not have, you did not really test anything.

Question 7 — Forecasting

Every forecast in this question is **one step ahead**. You will produce **six forecast tracks**, each covering the twelve dates of the weight window.

The rolling one-step-ahead design

Read this section carefully. It is short, and it describes a design we have not drilled in class.

A one-step-ahead forecast is written $\hat{y}_{t+1|t}$: the forecast *for* period $t+1$, made *at* period t, using every actual observation up to and including y_t and nothing after it. Roll the origin t forward one period at a time,

$$t = T,\ T+1,\ \ldots,\ T+11,$$

and you get twelve forecasts, one for each of the twelve dates $(T+1) : (T+12)$ — the weight window. That is the whole design.

Two things about it, and the second is the one people miss:

- **The information set rolls.** The forecast for $T+5$ is entitled to use y_{T+4}. By the time you are standing at $T+4$ that value has arrived, and a forecaster working in real time would have had it.
- **The parameters do not.** Every model on this exam is estimated once, on the training block `1:T`, and never again. Nothing is re-fit as the origin moves. Re-estimating at every origin is a different exercise — a rolling *refit* — and it is not what this question asks for.

This is why the exam stops at the weight window. A one-step forecast for $T+14$ needs y_{T+13} in hand, and y_{T+13} is inside the holdout — so you cannot forecast your way across the holdout without first opening it. The weight window is the most this midterm can honestly produce, and it is exactly what Question 8 scores. Problem Set 3 produces the holdout forecasts in October, when it is entitled to the data they require.

The three ARIMA tracks

Use all three models from your Question 5 table — not just the winner. These become the columns `arima_1`, `arima_2`, and `arima_3`. Say in your report which specification is which.

You do not have to write the recursion yourself. `forecast::Arima()` will apply an already-estimated set of coefficients to a longer series without estimating anything new:

```
fit   <- Arima(train, order = c(p, d, q))   # train = y[1:T]; estimated once
refit <- Arima(y[1:(T + 12)], model = fit)  # re-uses fit's coefficients; fits NOTHING
os    <- fitted(refit)                      # one-step-ahead forecasts throughout
```

The `model = fit` argument is doing all the work: it re-uses `fit`'s coefficients verbatim, so `coef(fit)` and `coef(refit)` are the same numbers. What it recomputes is the *recursion* — feeding it actuals one period at a time — so `fitted(refit)` is the sequence of one-step-ahead forecasts you want. Because the series was truncated at $T+12$, the **last twelve** fitted values are exactly your twelve weight-window forecasts, and no holdout observation has been anywhere near your session.

Note the alignment: `fitted(refit)` at date t is the forecast **for** t, made at $t-1$. It is already sitting on the date it predicts. Hold on to that, because the next family is not.

If you find yourself typing `Arima(y, order = c(p, d, q))` on the full series without `model = fit`, stop. You have just re-estimated the model on data this exam does not let you see.

The averaging family

The remaining three tracks come from a single formula, indexed by a window length k. Standing at date t, the forecast for $t+1$ is the average of the last k observations you have actually seen:

$$\hat{y}_{t+1|t} = \frac{1}{k} \sum_{j=0}^{k-1} y_{t-j}$$

As t rolls forward the window rolls with it, so for the later origins the average is taken over weight-window observations. That is correct and intended — those values have arrived by then.

- $k = 1$ — $\hat{y}_{t+1|t} = y_t$, the last observation carried forward. Column `locf`.
- k — one intermediate value that you choose and **justify from the persistence you characterized in Question 4**. Column `avg_k`. At the early origins this window reaches back into the training block, which is exactly right: the average is over k actual observations at every origin, never fewer. If your first forecast is an average of one or two numbers, you started the window at $T+1$ instead of letting it run back.
- **Expanding** — k grows with the origin, so the average runs over everything observed so far: $\hat{y}_{t+1|t} = \frac{1}{t} \sum_{s=1}^{t} y_s$. Column `meanf`.

The first and last are the one-step versions of `forecast::naive()` and `forecast::meanf()`, which is where the column names come from. Both are also one line of base R, and writing them yourself is a reasonable way to be certain what you built.

Three things about this family, stated plainly because every one of them has tripped students up before.

1. **We call this a k-period average forecast, not a moving average.** In this course "moving average" means the MA(q) component of an ARIMA model, which is a completely different object — a weighted sum of past *innovations*, not of past observations. Do not use the phrase.
2. **None of these is a flat line, with one qualified exception.** `locf` tracks your series with a one-period lag. `avg_k` tracks it, smoothed — the larger the k, the smoother and the more sluggish. The expanding mean will look *approximately* flat across twelve periods if your series is stationary, because by then each new observation moves an average of many terms by very little. That is a fact about your series, not a property of the method: on a trending series the expanding mean climbs, permanently behind. If your `meanf` column comes out flat, say why it did.
3. **Six tracks, twelve rows**, written to `forecasts.csv`: `arima_1`, `arima_2`, `arima_3`, `locf`, `avg_k`, `meanf`, keyed by `date`. One row for each date in the weight window, `(T+1):(T+12)`. Twelve rows, not twenty-four.

The mistake that will cost you this question

A correctly lagged rolling average is now exactly the right answer, so `zoo::rollmean` is a perfectly good tool here. The danger is not the function. It is the off-by-one.

`zoo::rollmean(y, k, align = "right")` at position t returns the average of y_{t-k+1} through y_t. That is the forecast **for** $t+1$. It comes back sitting on row t, so it must be shifted forward one period before you compare it to anything.

Fail to shift it and the number you are calling a forecast for date t has already seen y_t — the value it is supposed to be predicting. **It will look extraordinarily good and it will be worthless.**

This is look-ahead bias, and it is the single most likely way to hand in a confidently wrong answer on this exam. It does not announce itself. It shows up as a suspiciously excellent scorecard in Question 8, which is precisely when you are least inclined to go looking for a bug.

Check it before you submit, on every column. The cheapest check there is: the entry in your `locf` column on date t must equal the actual value of your series on date $t-1$, exactly. If it equals the actual on date t, the column is shifted wrong — and if `locf` is shifted wrong, everything you built the same way is too.

Plot your six tracks against the actuals over the weight window. Do not plot them against the holdout. There is no fan to draw here: all twelve forecasts are made at the same horizon, so forecast uncertainty does not widen across the window the way it does for a multi-step path from a fixed origin.

What earns the points here. Full marks: `forecasts.csv` has six forecast columns and twelve rows, the dates match the weight-window dates in `data.csv` exactly, every ARIMA path comes from coefficients estimated on train and never re-estimated, and every column is aligned so that the entry on date t was computable at $t-1$. Your k is justified from something you actually measured in Question 4 — the decay rate of your correlogram, the lag at which autocorrelation dies — and you say what that choice implies about how quickly your forecast can respond. Half marks: a column that is an unshifted `zoo::rollmean`, which scores beautifully and means nothing; a k chosen "because four quarters is a year" with no reference to the persistence in your own data; ARIMA paths from `Arima()` refit on the full sample because that was the object already in memory; or a `locf` column that is a constant, which means you produced a flat forecast from a frozen origin and answered a question this exam is not asking. Verify the alignment. It is the cheapest work on this exam and the most expensive thing to get wrong.

Question 8 — Forecast evaluation

Choose a loss function and define it. Then score **all six tracks on the weight window only** — the twelve dates `(T+1):(T+12)`.

Every one of those twelve numbers is a one-step-ahead forecast, so what you have is twelve *origins*, not twelve horizons. Nothing on this exam is scored at a horizon greater than one, and your scorecard therefore has no horizon dimension to break out.

Report a scorecard and answer three things:

- Which track wins?

- Does the model your information criteria selected in Question 5 also forecast best? If not, what do you make of that?
- Does **any** of your ARIMA models beat the naive baselines — `locf`, `avg_k`, `meanf`?

Your available loss functions are **MSE, MAE, and RMSE**. MASE, Theil's U, and the Diebold-Mariano test are Module 7 material; they belong to Problem Set 3, not to this exam.

On that last question, recall Module 6 §6.8, where the naive random walk beat a diagnostic-clean SARIMA on UNRATE across every metric we computed. A model can pass every diagnostic in Module 5 and still add nothing over "carry the last number forward" on a calm stretch with no turning points. **A loss to the baseline is information, not failure.** It is a finding about your series and your window, and it belongs in the report — discussed, not hidden.

> **What earns the points here.** Full marks: one loss function, defined with its formula, chosen because of what it costs you to be wrong on *this* series, and applied identically to all six tracks across the twelve weight-window dates. A clean ranking, and an honest paragraph about what it means when the IC winner is not the forecast winner — or when nothing you built beats carrying the last number forward. Half marks: three loss functions reported so that something wins under one of them, a scorecard printed with no interpretation, a loss function selected after the results came in, a comparison across loss functions treated as though it settled anything, or the twelve one-step forecasts written up as though they were a horizon profile. And to be explicit: **any number in your report computed on the holdout window costs you points in this row** — and under Question 7's design, arriving at one at all means you went and got data you were told to leave alone. Losing to `locf` is a finding worth two sentences. Quietly dropping the baselines from the scorecard is a different thing entirely, and I will notice which columns are missing.

Report craft

This row is scored across the whole document, not at any one question.

> **What earns the points here.** Full marks: a reader who has never taken this course can follow the report from the first page to the last. Every figure is referenced in the text, legible at printed size, and labeled in words rather than variable names. Every table has units. The report is inside 15 pages, is organized, and is free of the kind of error that a second read would have caught. Half marks: a wall of R console output with prose wrapped around it, axis labels left at `sp$train`, six pages of estimation tables against one paragraph of interpretation, or figures that appear and are never mentioned. Length is not the measure of effort here. I would rather read nine tight pages than fifteen padded ones, and a figure that is not referenced in the text costs you space and earns nothing.

Code and reproducibility

This row is scored on `analysis.R`, not on the report.

What earns the points here. Full marks: I open `analysis.R`, run it top to bottom in a clean session, and it writes `data.csv` and `forecasts.csv` and produces every number and every figure I read in your report. It is sectioned, it is commented in a way that explains *why* rather than restating the function name, and its paths are relative. Half marks: a script that runs but produces numbers that do not match the report, a `setwd("C:/Users/...")` at the top, code that silently depends on objects left in your workspace from an earlier session, or a file that is a transcript of everything you tried rather than a record of what you did. You will open this script again in October and build Problem Sets 3, 4 and 5 on top of it. Write it for that person.