Discrete Time Series Analysis

Course Code	Instructor	Lectures		
STAT 464/864	Skye P. Griffith	Jeffery Hall	Monday	10:30
Fall 2024	griffith.skye@queensu.ca	Room 118	Wednesday	9:30
			Friday	8:30

Office Hours Wednesdays, 15:00-16:00 **Zoom Link:** ♥ **Password:** See OnQ course page

TA Name: Yangsai Lyu Email: 21yl36@queensu.ca

Textbook Introduction to Time Series and Forecasting (third edition)
by Peter J. Brockwell and Richard A. Davis

Prerequisites STAT 361 or ECON 351, or permission of the Department.

Course Description Time series are data collected over time, e.g., daily temperature. The interesting structures in time series are, well, *temporal,* whether they be probabilistic or deterministic (independent observations over time do not make interesting time series. In fact, that's called noise). This course will spend most of its time looking at probabilistic models for time series in which the observations are collected at regular intervals. We will see how we can use these models to do forecasting. We will also look at some ways we can deal with deterministic temporal structure in the data.

Online Resources

OnQ
Link: ♥

This will act as one online portal for the course. It's where you'll find:
— Large files (class recordings, other video lectures)
— Administrative stuff (announcements, grades)

Skye's Github
Link: ♥

The other online portal for the course. Here you'll find:
— Frequently updated files (homework, slides)
— Coding resources (workshop files, datasets)

Crowdmark
Link: ♥

You'll submit all assignments here, including your final project.
This is also where you can view the grader's feedback on your submissions.

Learning Outcomes

By the end of the course, students should know:

1. Methods to estimate and eliminate trends and seasonal components.
2. The concepts of stationarity, autocovariance, and autocorrelation.
3. What linear filters are, and their properties.
4. Relevant hypothesis tests and confidence intervals.
5. Optimal linear forecasting for a variety of time series models.
6. AR(p), MA(q), and ARMA(p,q) models.
7. Estimation techniques for ARMA models.
8. The use of R for working with time series.
9. The use of Quarto for rendering PDF reports with dynamic output.
10. How to apply the course's techniques to real data, and interpret the results.

Lectures

Lectures will mostly consist of (flamboyant) slides, with the occasional handwritten derivation using a tablet and projector. Barring technical difficulties, these components of the lectures **will be recorded** (video screen capture), and regularly **uploaded to OnQ.** That being said, I intend to make time for higher level discussions of the content during class, and I will encourage you to ask questions and participate in such conversations. Thus, the better your attendance, the more you will get out of this course.

Printable copies of the slides, complete with full derivations, will be made available each week on **my Github,** in a folder called **Printable_Files**. My slideshows include long equations and proofs that would normally be written out by hand, and it's up to you how you want to engage with them in class and on your own time. Here are some options:

1. **Tablet (Recommended):** Bring a tablet of some kind if you have one, so you can draw on the PDF. The printable files include blank pages to provide space for extra notes. (Alternatively, you can mark up PDFs using a text tool in programs like OneNote, if you don't have a touchscreen.)
2. **Handwritten Notes:** If you don't have access to a tablet/printer, or if you prefer to take notes by hand, you may still do so. The math on screen will mostly be presented character by character, and we will walk through proofs at a transcribable pace. Don't be afraid to ask me to slow down.
3. **Printed Slides:** I gave the PDF slides white backgsrounds so they don't use as much ink, but like... they're gonna use a lot of ink, bro. But you *can* print them out and write on them in class.

Links to full Google Slides presentations are provided in the **README on my Github.** Unlike the printable slides, these presentations include GIFs and video files. I use a lot of animations when I teach, so you may want to take advantage of these links.

Workshops

Some lectures will take the form of interactive coding workshops. You will be expected to **bring your laptop for these workshops, with RStudio installed.** Depending on how quickly we traverse each topic, these workshops may be irregularly spaced over the course – I will let you know ahead of time if we are running the class as a workshop, via **OnQ announcements.**

Workshop Files

Prior each workshop, a Quarto Markdown Document (.qmd) file will be made available on **my Github**, in a folder called **Workshop_Files**. You will edit this file in RStudio, filling in the missing code as we work through the lesson. **Download it before coming to workshop.**

At the end of each workshop, I'll upload the **completed workshop document** (a PDF rendered by Quarto) to that same **Workshop_Files** folder. It will contain the "solutions" to the workshop, in case you miss something and don't want to search through a 50min video recording.

Software

Computing is critical to analysing time series. Computing will be a major component of the assignments, but won't be included on the midterm nor the final exam. The required computing environment for this course is the R statistical environment, however, we will be extending this environment to use a variety of special tools throughout the course. All such software is **mandatory,** and also **free.**

I'll provide a number of resources to help you with the installation process, including short video tutorials provided by some of colleagues at Trent University. You can find these on **both OnQ and my Github.**

Rstudio

This program runs on windows, mac and linux operating systems. It is an interface that allows you work with your R console, scripts, data, and plots all in the same place. You **must** download and install RStudio to do the computing in this course.

Quarto

This program allows you to produce PDF documents from within the RStudio environment. You'll be expected to submit fully rendered Quarto documents for all assignments/reports in this course.

LaTeX

Math notation included in your final project must be typed using LaTeX. We will cover the basics of this skill, and use it consistently during the coding workshops.

Course Outline

Tentative. Very tentative... What my old math reasoning prof might call a "polite fiction."

Introduction to Time Series	Chapter 1
Definition and Classical Decomposition	
Eliminating Trends	Chapter 1
Polynomial Regression	
Moving Average (MA) Smoothing Filters	
Exponential Smoothing Filters	
Eliminating Seasonality	Chapter 1
Harmonic Regression	
S1 Method (See text)	Section 1.5.2.1
The Difference Operator	
Stationarity and Autocovariance	Chapter 2
Covariance / Correlation	
Functions of lag (ACVFs / ACFs)	
Matrix forms and Sample Estimates	
Linear Filters	Chapter 1/2
Causality and Proposition 2.2.1	
Concatenation of Multiple Filters	
The Backshift Operator	
ARMA models (Part I)	Chapter 2
Autoregressive (AR) Processes	
Moving average (MA) Processes	
ARMA(1,1) Processes	
ARMA models (Part II)	Chapter 3
ARMA(p,q) Processes	
Causality and Invertibility	
Linear Prediction	Chapter 2
Best Linear Predictor (LP)	
(Minimum) Mean Squared Error (MSE)	
Properties of the LP operator	
Forecasting	Chapter 2
1-step ahead + MSE AR(1), AR(p), MA(1), ARMA(1,1)	
h-step ahead + MSE AR(1), AR(p)	
LP Approximation using the Infinite Past	
The Innovations Algorithm & Beyond	Chapter 5
Application to MA(q); h-step ahead + MSE	
Prediction Intervals	
Partial ACFs (PACFs)	

Assessment

Grading Scheme

Component	Weight	Due Date (Fridays at Midnight)
Homework 1	15%	Sep. 20
Homework 2	15%	Oct. 4
Midterm	15%	Oct. 21 (Monday, in class)
Homework 3	15%	Nov. 1
Project (proposal)	5%	Nov. 15
Homework 4	15%	Nov. 29
Project (report)	20%	Dec. 7-21 (feedback cutoff: Dec. 13)

Homework *The homework in this course follows a contemporary grading scheme. It incorporates elements of **Mastery Grading,** also known as Standards-Based Grading or Specifications Grading (Nilson 2014). The goal of this method is to measure your understanding of major topics without restricting you from demonstrating that understanding over multiple attempts throughout the course.*

Think of the homework in terms of individual problems, rather than assignments. The assignments are just themed groupings of problems. All problems are weighted equally, each worth exactly 5 points, with no part marks: $(0/5, 1/5, \ldots, 5/5)$. Each assignment contains 6 problems.

Choose 4 problems to complete per assignment (yes, you can skip the other 2). These are your "Original Problems." Hand these in by their assignment's due date.

When the TA is finished grading an assignment, you may view your results on Crowdmark. If you want to shoot for a higher grade, you can attempt one of the problems you skipped — these are your "Alternate Problems." You are permitted a total of 2 Alternate Problems per assignment.

If you score higher on an Alternate Problem than you did on an Original Problem from the same assignment, I will swap in the better grade, no penalty. All Alternate Problems are due the last day of class.

In other words: you get a free second attempt for up to half of the homework problems, but that second attempt requires solving a slightly different problem based on the same topic.

Bad news: I will not post full assignment solutions.

Good news: I'll go over the solutions to your submitted problems in office hours. Additionally, the TA will provide basic feedback on Crowdmark.

Formatting Expectations The assignments and reports you submit in this course **must** be pdf documents rendered using Quarto: a tool included with RStudio. We will cover how to do this in class. This is a transferable skill that will allow you to produce professional quality reports within the RStudio environment – moreover, the plots and computations rendered in these documents will be the direct output of your true code **(dynamic output).** All equations included in your final project must be typed using LaTeX. On the assignments, all coding and discussion of the material must be typed – however, theoretical proofs may be written by hand.

Final Project You will be required to write a final report as a full analysis of a data set, using the methods and theory developed throughout the semester. This project will take the place of the final examination, and is broken down into 2 components:

1. A **proposal** (1-2 pages) pitching your choice of dataset and your analysis plan. This will be due some time after the midterm.
2. A **report** discussing your analysis and its results. Due at the end of the semester.

Graduate Students Each assignment will have 2 additional **Graduate Problems** (I will make clear which these are). In addition to the undergraduate's tasks, graduate students must choose 1 Original Graduate Problem to complete per assignment, leaving the other as an Alternate Graduate Problem. Anyone may attempt Graduate Problems as alternates to Original Undergrad Problems, but the converse is not permitted.

The midterm will include one graduate-level question – this is only mandatory for graduate students to complete. Moreover, graduate students will be held to higher expectations than undergraduates in regards to the quality of their final projects.

Copyright: Course materials created by the course instructor, including all slides, presentations, synchronous and asynchronous course recordings, handouts, tests, exams, and other similar course materials, are the intellectual property of the instructor. It is a departure from academic integrity to distribute, publicly post, sell or otherwise disseminate an instructor's course materials or to provide an instructor's course materials to anyone else for distribution, posting, sale or other means of dissemination, without the instructor's express consent. A student who engages in such conduct may be subject to penalty for a departure from academic integrity and may also face adverse legal consequences for infringement of intellectual property rights and, with respect to recordings, potentially privacy violations of other students.

Academic Integrity: Information on policies concerning academic integrity is available in the Queen's University Code of Conduct, in the Senate Academic Integrity Policy Statement, on the Faculty of Engineering and Applied Science website, and from your instructor. Departures from academic integrity include plagiarism, use of unauthorized materials or services, facilitation, forgery, falsification, unauthorized use of intellectual property, and collaboration, and are antithetical to the development of an academic community at Queen's. Given the seriousness of these matters, actions which contravene the regulation on academic integrity carry sanctions that can range from a warning or the loss of grades on an assignment to the failure of a course to a requirement to withdraw from the University.

TLDR

This course will be organized using a mixture of OnQ and my Github (link on page 1).

Submit your work on Crowdmark.

You'll need a bunch of software, some of which you might already have.
I will post installation tutorials to both Github and OnQ.

The lectures will mostly be slideshows, and I'll provide the slides beforehand (Github). In class, you can: bring a tablet to write on the PDFs, bring copies you've printed on paper, or copy the slides by hand.

Some lectures will be coding workshops (I'll let you know, prior). You need to bring your laptop to these.

Workshop resources are all on my Github.

You have 4 assignments, a midterm, and a final project broken into 2 parts.

All submitted assignments and reports must be rendered by Quarto.

Each assignment has 6 problems. You choose 4 to complete, and 2 of those get an "extra life" which doesn't expire until the last day of class. Grad students are given 2 additional "expert" problems, one of which gets an extra life.

I don't post full assignment solutions, but I will walk you through specific solutions during office hours.