Modeling and Analysis of Time Series Data
Chapter 1: Introduction

STATS 531, Winter 2026

Edward Ionides

Objectives for this chapter

- Discuss some basic motivations for the topic of time series analysis.
- Introduce some fundamental concepts for time series analysis: stationarity, autocorrelation, autoregressive models, moving average models, autoregressive-moving average (ARMA) models, state-space models. These will be covered in more detail later.
- Introduce some of the computational tools we will be using.

Overview

- Time series data are, simply, data collected at many different times.
- This is a common type of data! Observations at similar time points are often more similar than more distant observations.
- This immediately forces us to think beyond the independent, identically distributed assumptions fundamental to much basic statistical theory and practice.
- Time series dependence is an introduction to more complicated dependence structures: space, space/time, networks (social/economic/communication), ...

Looking for trends and relationships in dependent data

The first half of this course focuses on:

1. Quantifying dependence in time series data.

2. Finding statistical arguments for the presence or absence of associations that are valid in situations with dependence.

Example questions: Does Michigan show evidence for global warming? Does Michigan follow global trends, or is there evidence for regional variation? What is a good prediction interval for weather in the next year or two?

Modeling and statistical inference for dynamic systems

The second half of this course focuses on:

1. Building models for dynamic systems, which may or may not be linear and Gaussian.
2. Using time series data to carry out statistical inference on these models.

Example questions: Can we develop a better model for understanding variability of financial markets (known in finance as volatility)? How do we assess our model and decide whether it is indeed an improvement?

Example: Winter in Michigan

There is a temptation to attribute a warm winter to global warming. You can then struggle to explain a subsequent cold winter. Is a trend in fact noticeable at individual locations in the presence of variability? Let's look at some data, downloaded from www.usclimatedata.com and put in `ann_arbor_weather.csv`.

- This file is in the course git repository on GitHub. Make a local clone of this repo to get an up-to-date copy of all data, notes, code, homeworks and solutions for this course.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf

y = pd.read_csv("ann_arbor_weather.csv",
    sep='\t', comment='#')
```

Reproducible documents with Quarto

The notes combine source code with text, to generate statistical analysis that is:

- Reproducible
- Easily modified or extended

These two properties are useful for developing your own statistical research projects. Also, they are useful for teaching and learning statistical methodology, since they make it easy for you to replicate and adapt analysis presented in class.

- Many of you will already know Jupyter notebooks and/or Rmarkdown.
- Quarto (qmd format) is a newer system that works with both Python and R, and naturally produces both html and pdf outputs.

Some basic investigation using Python

To get a first look at our dataset, we can examine its structure:

```
print(y.head(4))
```

```
   Year  Low  High  Hi_min  Lo_max  Avg_min  Avg_max  Mean  Precip  Snow  \
0  1900 -7.0  50.0    36.0    12.0     18.0     34.7  26.3    1.06   4.0
1  1901 -7.0  48.0    37.0    20.0     17.0     31.8  24.4    1.45  10.1
2  1902 -4.0  41.0    27.0    11.0     15.0     30.4  22.7    0.60   6.0
3  1903 -7.0  50.0    36.0    12.0     15.1     29.6  22.4    1.27   7.3

   Hi_Pricip  Hi_Snow
0       0.28      1.1
1       0.40      3.2
2       0.25      2.5
3       0.40      3.2
```

We focus on `Low`, which is the lowest temperature, in Fahrenheit, for January.

Statistical uncertainty

As statisticians, we want an uncertainty estimate. We want to know how reliable our estimate is, since it is based on only a limited amount of data.

- The data are $y_1, \ldots, y_N$, which we also write as $y_{1:N}$.

- Basic estimates of the mean and standard deviation are

$$\hat{\mu}_1 = \frac{1}{N}\sum_{n=1}^{N} y_n, \qquad \hat{\sigma}_1 = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(y_n - \hat{\mu}_1)^2}.$$

- This suggests an approximate confidence interval for μ of $\hat{\mu}_1 \pm 1.96\,\hat{\sigma}_1/\sqrt{N}$.

Question: What are the assumptions behind this confidence interval?

Computing the estimates

1955 has missing data, requiring a minor modification. So, we compute $\hat{\mu}_1$ and $SE_1 = \hat{\sigma}_1/\sqrt{N}$ as:

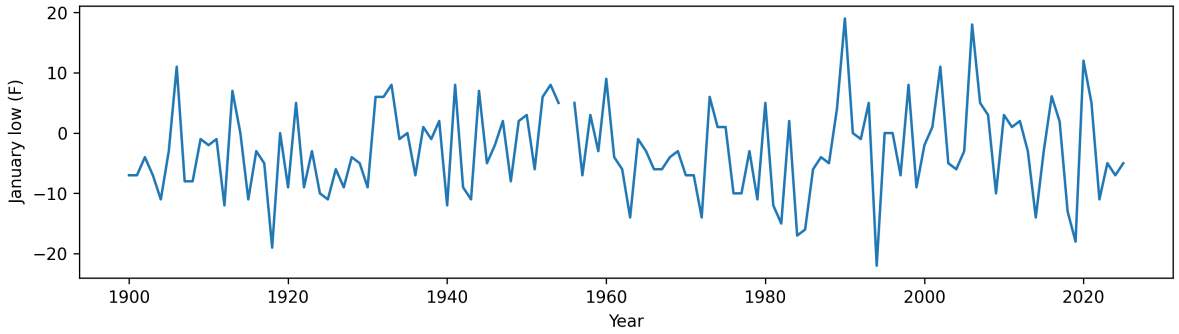
```
mu1 = y['Low'].mean()
se1 = y['Low'].std() / np.sqrt(y['Low'].count())
print(f"mu1 = {mu1:.2f}, se1 = {se1:.2f}")
```

```
mu1 = -2.88, se1 = 0.67
```

Question: If you had to give an uncertainty estimate on the mean, is it reasonable to present the confidence interval, -2.88 $\pm$ 1.31? Do you have ideas of a better alternative?

The first rule of data analysis is to plot the data in as many ways as you can think of. For time series, we usually start with a time plot.

```
plt.figure(figsize=(10, 3))
plt.plot(y['Year'], y['Low'], '-')
plt.xlabel('Year'); plt.ylabel('January low (F)')
plt.tight_layout(); plt.show()
```



Question: Do you see any patterns in the data?

The auto-regressive moving average (ARMA) model

A basic way to look for dependence is to fit an **auto-regressive moving average** (ARMA) model. We'll see ARMA models in more detail later in the course. Let's fit an ARMA model given by

$$Y_n = \mu + \alpha(Y_{n-1} - \mu) + \epsilon_n + \beta\epsilon_{n-1}.$$

This has a one-lag autoregressive term, $\alpha(Y_{n-1} - \mu)$, and a one-lag moving average term, $\beta\epsilon_{n-1}$. It is therefore called an ARMA(1,1) model. These lags give the model some time dependence.

- $\alpha = \beta = 0$ is the basic independent model, $Y_n = \mu + \epsilon_n$.
- $\alpha = 0$ is a moving average model with one lag, MA(1).
- $\beta = 0$ is an autoregressive model with one lag, AR(1).

We model $\epsilon_1 \ldots, \epsilon_N$ to be an independent, identically distributed (iid) sequence. To be concrete, let's specify a model where they are normally distributed with mean zero and variance σ^2.

A note on notation

- In this course, capital Roman letters, e.g., X, Y, Z, denote random variables. We may also use ϵ, η, ξ, ζ for random noise processes. Thus, these symbols are used to build models.
- We use lower case Roman letters (x, y, z, ...) to denote numbers. These are not random variables. We use y to denote a data point.

- "We must be careful not to confuse data with the abstractions we use to analyze them." (William James, 1842-1910).
- Other Greek letters will usually be parameters, i.e., real numbers that form part of the model.

ARMA(1,1) maximum likelihood estimation in Python

```
arma11 = ARIMA(y['Low'], order=(1, 0, 1)).fit()
print(arma11.summary())
```

```
/Users/ionides/git/531w26/.venv/lib/python3.12/site-packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary starting autoregressive parameters
  warn('Non-stationary starting autoregressive parameters'
/Users/ionides/git/531w26/.venv/lib/python3.12/site-packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible starting MA parameters found. Using
  warn('Non-invertible starting MA parameters found.'


                               SARIMAX Results
==============================================================================
Dep. Variable:                    Low   No. Observations:                  126
Model:                 ARIMA(1, 0, 1)   Log Likelihood                -427.884
Date:                Tue, 06 Jan 2026   AIC                            863.768
Time:                        12:08:42   BIC                            875.113
Sample:                             0   HQIC                           868.377
                                  - 126
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.8744      0.697     -4.125      0.000      -4.240      -1.509
ar.L1         -0.5974      1.334     -0.448      0.654      -3.212       2.017
ma.L1          0.6322      1.298      0.487      0.626      -1.911       3.176
sigma2        55.0492      6.923      7.952      0.000      41.481      68.617
===================================================================================
Ljung-Box (L1) (Q):                   0.05   Jarque-Bera (JB):                 0.95
Prob(Q):                              0.82   Prob(JB):                         0.62
Heteroskedasticity (H):               1.83   Skew:                             0.21
Prob(H) (two-sided):                  0.05   Kurtosis:                         3.11
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Extracting key parameters

- The full model summary printout is too much information. You or your reader might not understand everything. Some parts might be best ignored.
- We can extract the quantities of primary interest from the fitted ARMA model in Python.

```
mu2 = arma11.params['const']
se2 = arma11.bse['const']
print(f"mu2 = {mu2:.2f}, se2 = {se2:.2f}")
```

```
mu2 = -2.87, se2 = 0.70
```

- We will write the ARMA(1,1) estimate of μ as $\hat{\mu}_2$, and its standard error as SE_2.

Comparing the iid estimate with the ARMA estimate

```
/Users/ionides/git/531w26/.venv/lib/python3.12/site-packages/statsmodels/tsa/statespace/sarim
  warn('Non-stationary starting autoregressive parameters'
/Users/ionides/git/531w26/.venv/lib/python3.12/site-packages/statsmodels/tsa/statespace/sarim
  warn('Non-invertible starting MA parameters found.'
```

- The two estimates, $\hat{\mu}_1$ = -2.88 and $\hat{\mu}_2$ = -2.87, and their standard errors, $\mathrm{SE}_1 = 0.67$ and $\mathrm{SE}_2 = 0.70$, are close.
- For data up to 2015, $\hat{\mu}_1^{2015}$ = -2.83 and $\hat{\mu}_2^{2015}$ = -2.85, with standard errors, $\mathrm{SE}_1^{2015} = 0.68$ and $\mathrm{SE}_2^{2015} = 0.85$.
- In this case, the standard error for the simpler model is 20.3% smaller.

Exactly how the ARMA(1,1) model is fitted and the standard errors computed will be covered later.

Question: When standard errors for two methods differ, which is more trustworthy? Or are they both equally valid for their distinct estimators?

Model diagnostic analysis

- We should do **diagnostic analysis**. The first thing to do is to look at the residuals.
- For an ARMA model, the residual r_n at time t_n is defined to be the difference between the data, y_n, and a one-step ahead prediction of y_n based on $y_{1:n-1}$, written as $\hat{y}_n$.

From the ARMA(1,1) definition,

$$Y_n = \mu + \alpha(Y_{n-1} - \mu) + \epsilon_n + \beta\epsilon_{n-1},$$

a basic one-step-ahead predicted value corresponding to parameter estimates $\hat{\mu}$ and $\hat{\alpha}$ could be

$$\hat{y}_n = \hat{\mu} + \hat{\alpha}(y_{n-1} - \hat{\mu}).$$

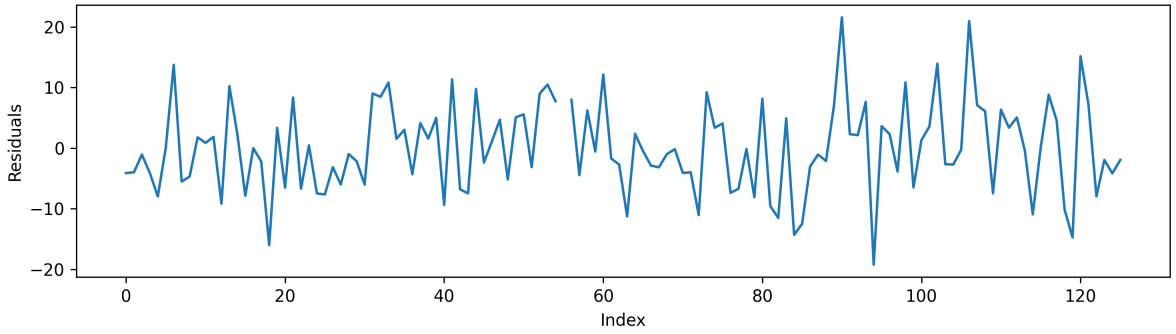
A **residual time series**, $r_{1:N}$, is then given by

$$r_n = y_n - \hat{y}_n.$$

In fact, Python's statsmodels does something slightly more sophisticated.

Residual plot

```
plt.figure(figsize=(10, 3))
plt.plot(arma11.resid)
plt.xlabel('Index')
plt.ylabel('Residuals')
plt.tight_layout()
plt.show()
```



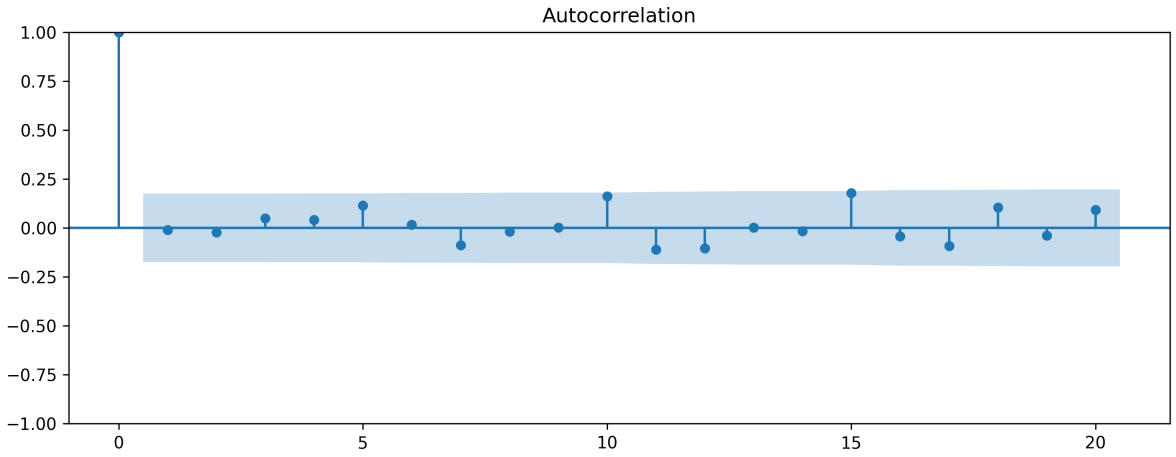
What patterns do you see?

- The residual plot may suggest slow variation in the residuals, over a decadal time scale.
- Is there short-term dependence? We study this by plotting the pairwise sample correlations at a range of lags. This is the **sample autocorrelation function**, or **sample ACF**, written for each lag h as

$$\hat{\rho}_h = \frac{\frac{1}{N}\sum_{n=1}^{N-h} r_n\, r_{n+h}}{\frac{1}{N}\sum_{n=1}^{N} {r_n}^2}.$$

ACF plot

```
fig, ax = plt.subplots(figsize=(10, 4))
plot_acf(arma11.resid.dropna(), ax=ax, lags=20)
plt.tight_layout()
plt.show()
```



- This shows no substantial autocorrelation. An ARMA model may not be a good way to describe the slow variation present in the residuals of the ARMA(1,1) model.

Quantifying uncertainty for scientific reproducibility

Usually, omitted dependency in the model will give overconfident (too small) standard errors.

- This leads to scientific reproducibility problems, where chance variation is too often assigned statistical significance.
- It can also lead to improper pricing of risk in financial markets, a factor in the US financial crisis of 2007-2008.

Artificial neural network (ANN) methods vs parametric models

- Training an ANN requires extensive data, and is complicated by dependence.
- ANN methods, "deep learning," are better adapted to prediction problems than testing hypotheses about causal mechanisms.
- Long short-term memory (LSTM) ANN architectures have been used for time series forecasting.
- This has only a small role in the course.

AI for data analysis

- Methodological choices in data analysis are sometimes limited by practical difficulties coding up the analysis.
- Methods for dependent data involve extra computational complexities.
- AI can provide effective coding support, if it is carefully used.
- We focus on quality over quantity to avoid being overwhelmed by excessive low-quality AI analysis.
- We learn AI data analysis by sharing successful techniques. Please share!
- These notes were translated from R to Python using claude code, with the context set by https://github.com/ionides/531w25/tree/main/CLAUDE.md

A first look at a state-space model

Scientists and engineers often have equations in mind to describe a system they're interested in. Often, we have a model for how the state of a stochastic dynamic system evolves through time, and another model for how imperfect measurements on this system gives rise to a time series of observations.

This is called a **state-space model**. The **state** models the quantities that we think determine how the system changes with time. However, these idealized state variables are not usually directly and perfectly measurable.

Statistical analysis of time series data on a system aims to:

1. Help scientists choose between rival hypotheses.
2. Estimate unknown parameters in the model equations.

We will look at examples from a wide range of scientific applications. The dynamic model may be linear or nonlinear, Gaussian or non-Gaussian.

A finance example: fitting a model for volatility of a stock market index

- Let $\{y_n, n = 1, \ldots, N\}$ be the daily returns on a stock market index, such as the S&P 500.

- The **return** is the difference of the log of the index. If z_n is the index value for day n, then $y_n = \log(z_n) - \log(z_{n-1})$.
- Since the stock market is notoriously unpredictable, it is often unproductive to predict the mean of the returns and instead there is much emphasis on predicting the variability of the returns, known as the **volatility**.
- Volatility is critical to assessing the risk of financial investments.

A volatility model

Financial mathematicians have postulated the following model. We do not need to understand it in detail right now. The point is that we may want to fit and evaluate time series models of interest to scientists.

$$
\begin{aligned}
Y_n &= \exp\left\{\frac{H_n}{2}\right\}\epsilon_n, \quad G_n = G_{n-1} + \nu_n, \\
H_n &= \mu_h(1-\phi) + \phi H_{n-1} \\
&\qquad + Y_{n-1}\sigma_\eta\sqrt{1-\phi^2}\tanh(G_{n-1}+\nu_n)\exp\left\{\frac{-H_{n-1}}{2}\right\} + \omega_n.
\end{aligned}
$$

- $\{\epsilon_n\}$ is iid $N(0,1)$, $\{\nu_n\}$ is iid $N(0,\sigma_\nu^2)$, $\{\omega_n\}$ is iid $N(0,\sigma_\omega^2)$.
- H_n is unobserved volatility at time t_n, with auto-regressive behavior, having dependence on Y_{n-1} and a slowly varying process G_n. We only observe the return, modeled by Y_n.

Questions to be addressed later in the course

This is an example of a **mechanistic model**, where scientific or engineering considerations lead to a proposed model. Now there is data and a model, it is time to recruit a statistician!

1. How can we get good estimates of the parameters, μ_h, ϕ, σ_ν, σ_ω, together with their uncertainties?
2. Does this model fit better than alternative models? So far as it does, what have we learned?
3. Does the data analysis suggest new models, or the collection of new data?

Likelihood-based inference for this partially observed stochastic dynamic system is possible, and enables addressing these questions [1].

By the end of this course, you will be able to carry out data analysis developing complex models and fitting them to time series. See past final projects for 2018, 2020, 2021, 2022, 2024.

References

[1] Carles Bretó. "On idiosyncratic stochasticity of financial leverage effects". In: *Statistics & Probability Letters* 91 (2014), pp. 20–26. DOI: 10.1016/j.spl.2014.04.003.