DATA X

Introduction to Time Series Forecasting

Chad Wakamiya
Spring 2020

Berkeley SCET

Agenda

Time Series Definition

What is a time series and why are they useful?

Forecasting Applications

Examples where forecasting is often used.

Practical Considerations

Forecast pitfalls and how to avoid them.

Time Series Features

Identifying trends, seasonality and holidays.

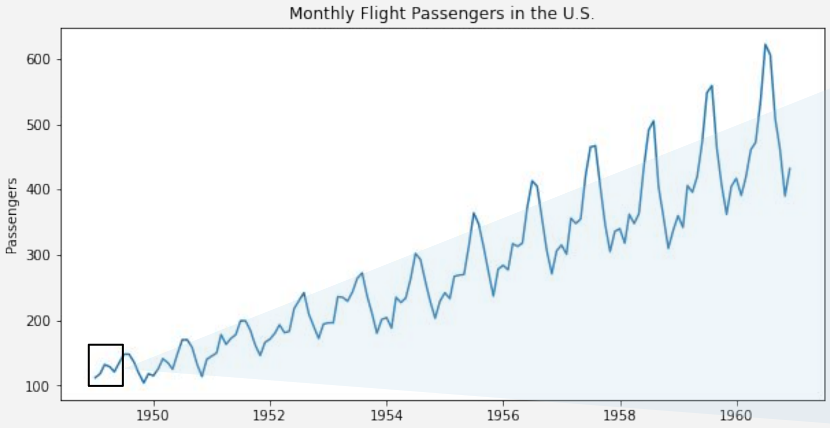
What is a time series?

Time Series

A ***time series*** dataset measures a quantity over time.

- The interval between consecutive observations is constant (daily, weekly, monthly, etc.)
- Time series are used to **forecast** some measured quantity at points in the future.

Measured Quantity



	passengers
Date	
1949-01-01	112
1949-02-01	118
1949-03-01	132
1949-04-01	129
1949-05-01	121

Time

Forecasting Applications

Forecasting time series applications naturally arise in many business and environmental settings.

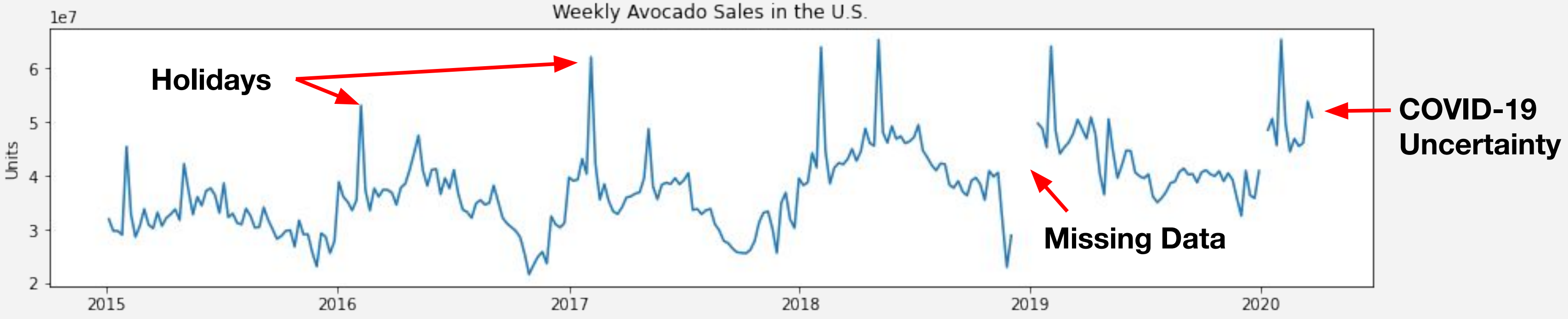
Examples:

- Demand planning for consumer goods
- Infrastructure capacity planning
- Environmental data (temperature, pollution)
- Financial data (prices, stock value)

Image by Pexels from Pixabay

Practical Considerations

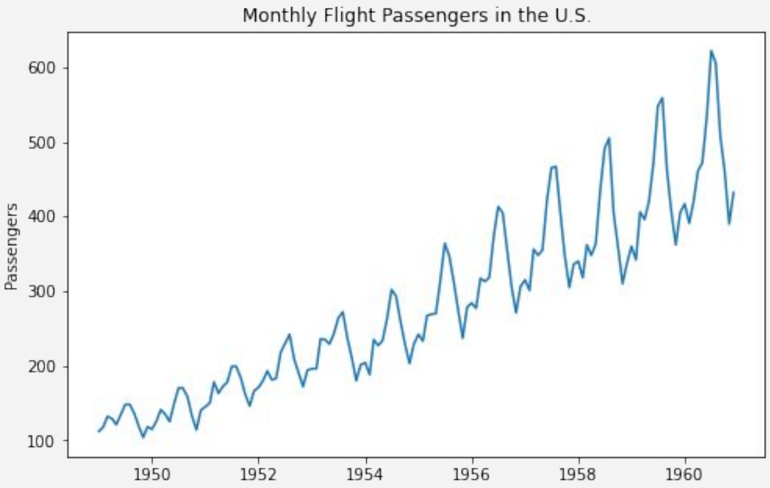
- Historical patterns don't necessarily inform future behavior
- Good forecasts often involve some mix of **domain expertise** and **statistics**
- Understand your time series data. (Are there outliers, missing data, holidays, or seasonal patterns?)
- Machine learning methods (random forests, neural nets, etc.) can be applied to time series, however, they may be difficult to explain to stakeholders.



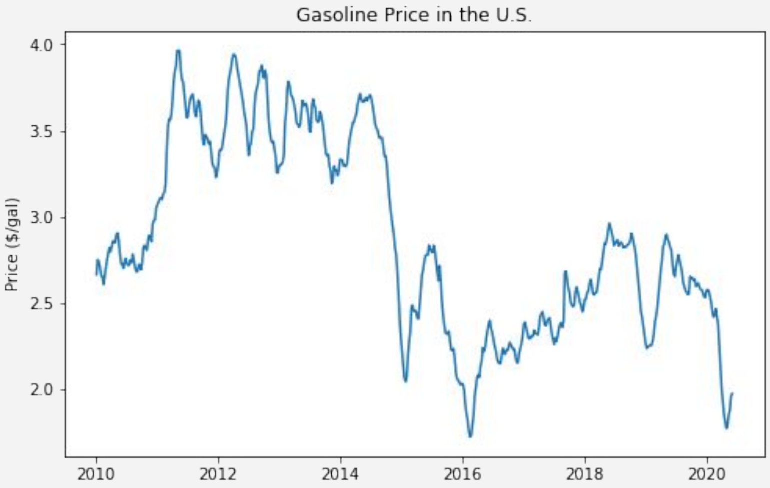
Time Series Features

Seasonality & Trend

Seasonality & Trend



No Seasonality & (Weak) Trend

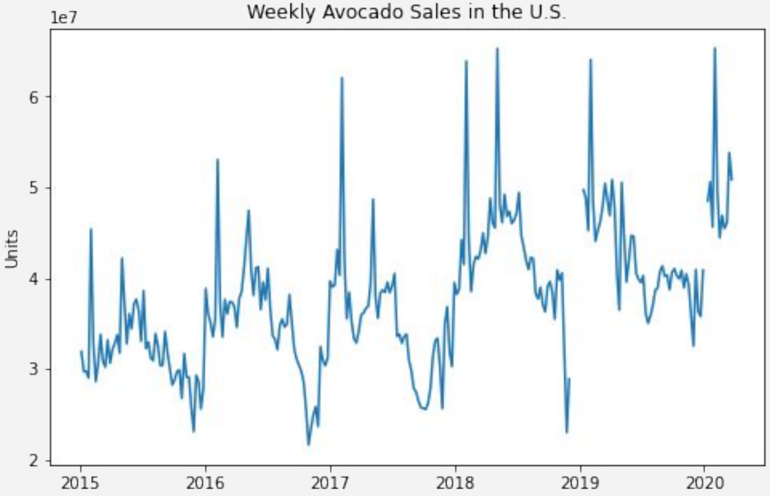


- **Seasonality** is a pattern that repeats at equal time intervals (daily, weekly, monthly, etc.)
- **Trend** a long-term increase or decrease throughout the time series.

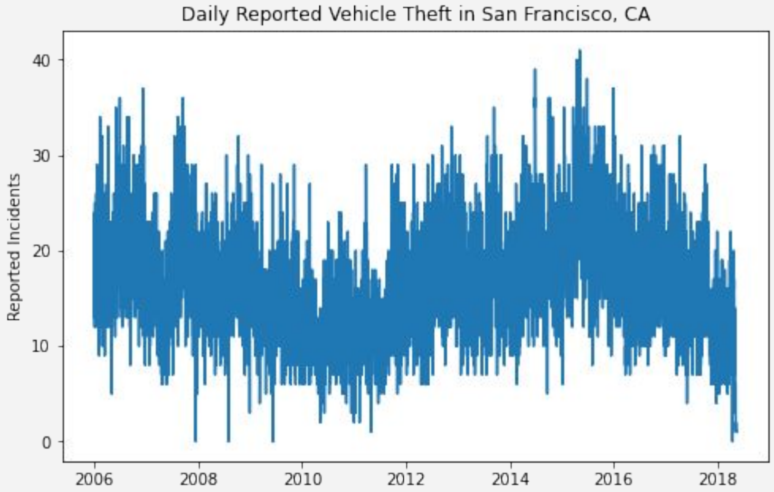
Holiday Spikes & Changing Trends

On top of seasonality and trend, there may be holiday spikes and the trend may change multiple times throughout the dataset.

Holiday Spikes



Changing Trends



Berkeley SCET

Selecting A Model

Agenda

No Seasonality & (Weak) Trend

- Simple Exponential Smoothing
- ARIMA

Seasonality and Trend

- Triple Exponential Smoothing
- SARIMA

Complex Models

- Prophet Library for holiday spikes and changing trends

No Seasonality & (Weak) Trend

Simple Exponential Smoothing

Simple Exponential Smoothing uses a weighted average where the weights increase exponentially with time.

S_t : Smoothed average at time t (the forecasted value)
$y_0, \ldots y_{t-2}, y_{t-1}$: Time series observations at time 0, … t-2, t-1

Parameters
α : A smoothing constant between 0 and 1 that controls the forecast's responsiveness.

$$S_t = (\alpha)\, y_{t-1} + (1-\alpha)\, S_{t-1}$$

This recursive formula can be expanded as: $S_t = \alpha\, [(\alpha-1)^0\, y_{t-1} + (\alpha-1)^1\, y_{t-2} + (\alpha-1)^2\, y_{t-3} + \ldots]$

Examples: $\alpha = 0.5$: $S_t = 0.5\, y_{t-1} + 0.25\, y_{t-2} + 0.125\, y_{t-3} + \ldots$

$\alpha = 0.01$: $S_t = 0.01\, y_{t-1} + 0.0099\, y_{t-2} + 0.00009\, y_{t-3} + \ldots$

When **α** increases the weights placed on more recent observations increase resulting in a more responsive forecast.

ARIMA (**A**utoregressive **I**ntegrated **M**oving **A**verage)

ARIMA forecasting fits a regression model on previous observations, error terms, and differencing.

$\hat{y}_t$: Time series value at time t (the forecasted value)
$y_0', \ldots y_{t-2}', y_{t-1}'$: *Differenced* time series observations at time 0, … t-2, t-1 (series may be differenced multiple times†)
c, β_i, ϕ_i : Constant term and model coefficients selected by regression

Parameters: order = (p, d, q)
p : The number of past observations to include (autoregressive order)
d : The number of times to difference the series to remove autocorrelation before fitting (differencing order)
q : The number of error terms to include (moving average order)

† Differencing means subtracting the consecutive values of the original series $y_0, y_1, \ldots y_{t-2}, y_{t-1}$.

- 1st Order (**d** =1): $y_i' = y_i - y_{i-1}$ for $i \geq 1$
- 2nd Order (**d**=2): $y_i'' = y_i' - y_{i-1}'$ for $i \geq 2$

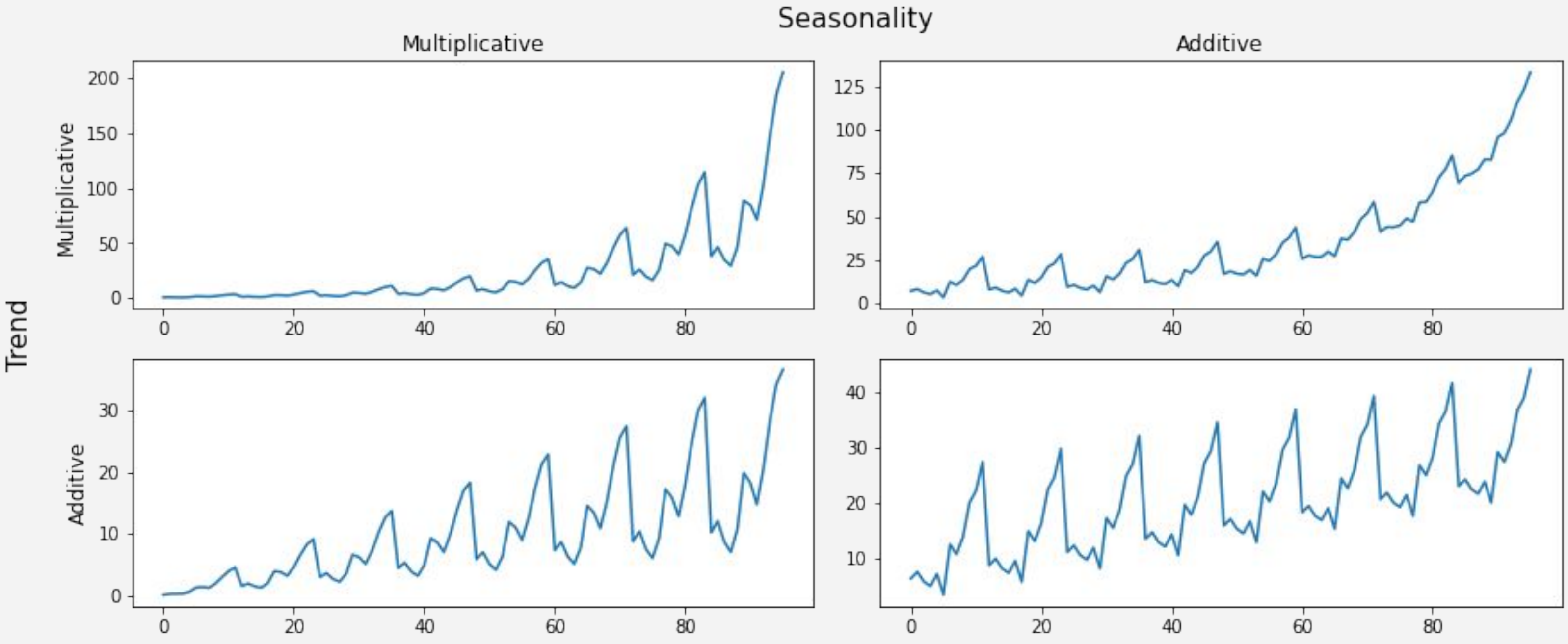
$$\hat{y}_t = c + \underbrace{\beta_1 y_{t-1}' + \beta_2 y_{t-2}' + \ldots + \beta_p y_{t-p}'}_{\text{Autoregressive Terms}} + \varepsilon_t + \underbrace{\phi_1 \varepsilon_{t-1} + \phi_2 \varepsilon_{t-2} + \ldots + \phi_d \varepsilon_{t-q}}_{\text{Moving Average Terms}}$$

where $\varepsilon_i = y_i - \hat{y}_i$ for $t \geq i \geq t-q$

Berkeley SCET

Seasonality and Trend

Types of Trend & Seasonality



Berkeley SCET

Triple Exponential Smoothing (Holt-Winters)

Triple Exponential Smoothing sums level, trend, and seasonality components where each component is a weighted average with weights that increase exponentially with time.

$\hat{y}_{t+h}$: Time series value at time t+h (the forecasted value), where h is the time between now and period being forecasted
$y_0, \dots y_{t-2}, y_{t-1}$: Time series observations at time 0, … t-2, t-1

Parameters

α : A smoothing constant between 0 and 1 that controls the level.
β : A smoothing constant between 0 and 1 that controls the trend
γ : A smoothing constant between 0 and 1 that controls the seasonality.

When α, β, and γ increase, the weights placed on more recent observations increase resulting in a more responsive forecast.

Additive Triple Exponential Smoothing

$$\hat{y}_{t+h} = S_t + hG_t + c_{t+h-m(k+1)}$$

Level: $S_t = \alpha (y_t - S_{t-m}) + (1-\alpha) (S_{t-1} + G_{t-1})$

Trend: $G_t = \beta (S_t - S_{t-1}) + (1-\beta) G_{t-1}$

Seasonality: $c_t = \gamma (y_t - S_{t-1} - G_t) + (1-\gamma) c_{t-m}$

m = number of periods in a year (i.e. m = 12 for monthly data); *k* = (h-1) mod m

- The level, trend, and seasonality components are calculated recursively as in simple exponential smoothing.
- S_0, G_0, c_0 are initialized using the beginning terms in the time series y0, y1,...
- The *multiplicative* triple exponential smoothing formula (very similar, but not shown) should be used when the trend and seasonality are multiplicative.

Berkeley SCET

SARIMA (Seasonal ARIMA)

SARIMA is the seasonal extension of the ARIMA model. It fits a regression model on previous observations, error terms, and differencing just like the ARIMA model, but adds terms to account for the seasonal components. These seasonal terms are controlled by the seasonal order (P, D, Q) which are analogous to (p, d, q) in ARIMA.

Parameters: order = (p, d, q), **seasonal order =** $\mathbf{(P, D, Q)_m}$

Original ARIMA parameters:

- p : The number of past observations to include (autoregressive order)
- d : The number of times to difference the series to remove autocorrelation before fitting (differencing order)
- q : The number of error terms to include (moving average order)

SARIMA parameters:

- **P**: Autoregressive order for the seasonal component
- **D**: Differencing order for the seasonal component
- **Q**: Moving average order for the seasonal component
- **m**: The number of periods in a year (i.e. m = 12 for monthly data, 52 for weekly data)

> $\mathbf{(P, D, Q)_m}$ allow observations from previous years to be added to the model. For example to forecast next December, the SARIMA model might use the value from the last few Decembers, perform differencing over past December observations, and use historical December errors.

Complex Models (Prophet)

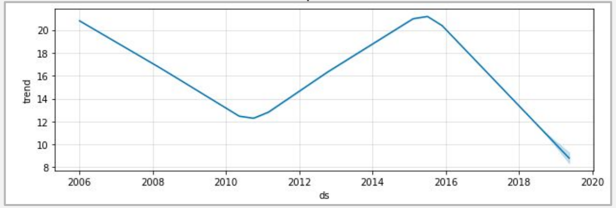
Holiday Spikes and Changing Trends

Prophet

Prophet is open source library for time series forecasting developed by Facebook. A Prophet model is composed of trend, seasonality, and holiday components fitted through the time series data points.

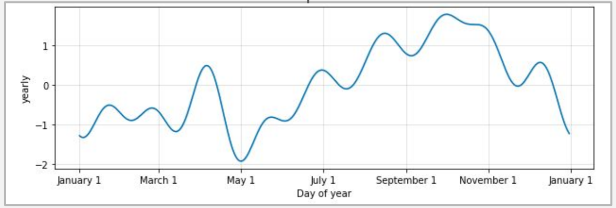
Noise that cannot be captured by the model

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$



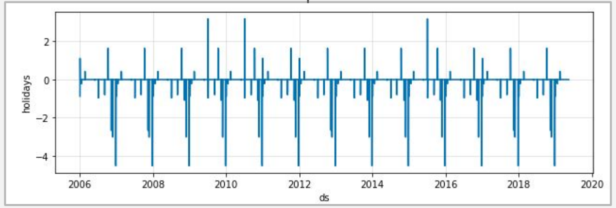
Trend Function

Models the non-periodic changes in the time series. Captures changing trends through identifying changepoints.



Seasonality Function

Captures yearly, monthly, week, and daily patterns. Estimated through Fourier Series.



Holiday Function

Adds peaks and drops that can be explained by holidays. Can account for holidays with dates that change year to year (Thanksgiving).

Parameters

- Add custom trend changepoints
- Specify Fourier series order (the smoothness of the seasonality function)
- Add custom holidays or events (Super Bowl or new product launches)

Examples of how to adjust these and other parameters can be found at Prophet Docs.

References/Future Reading

Exponential Smoothing:

- https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc431.htm

ARIMA

- https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
- https://people.duke.edu/~rnau/411arim.htm

Triple Exponential Smoothing

- https://otexts.com/fpp2/holt-winters.html
- *Production and Operations Analysis 7th Edition* by Steven Nahmias and Tava Lemon Olsen

SARIMA

- https://otexts.com/fpp2/seasonal-arima.html
- https://people.duke.edu/~rnau/seasarim.htm

Prophet

- https://peerj.com/preprints/3190v2/
- https://facebook.github.io/prophet/docs/quick_start.html#python-api