Time Series Analysis

ECE 6563, Electrical and Computer Engineering
University of New Brunswick

Recap

In the last section, we introduced methods to estimate, de-trend, and model time series

- Estimated/removed trend and seasonal components of time series
  Parametric, non-parametric, differencing, etc…
- Evaluated the residuals to assess stationarity
  WGN, IID, Random Walk, ACF, Lag plots, PACF, ADF test…
- Introduced ARIMA modeling of the residuals
  AR, Differencing, MA, SARIMA, etc…

In this section, we'll explore how we can

- Make design choices for our model parameters
- Use them to predict/forecast future values
- Evaluate their performance

Model Fitting

To build an ARIMA or SARIMA model, we need to estimate the associated parameters.

- We previously discussed fitting the AR and MA models based on the ACF and PACF

For AR, we can also quantify the fit using Least Squares, just as we did for linear trend models and seasonality.

But, for the MA component, there is no direct formula because there is no regression.

- Instead, we estimate iteratively until the sum of squared errors is minimized.

Another common method is Maximum Likelihood Estimation…

Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) measures the plausibility of observing our actual samples given a model

- The conditional probability of observing the data (X) given a specific probability distribution and parameters
- For most ARIMA models, MLE is similar to least squares estimation we used for regression that minimized $\sum_{t=1}^{T} \varepsilon_t^2$
- Assumes that x_t is Gaussian (often reasonable)

$$L(\beta, \sigma^2) = \frac{exp\left(-\frac{1}{2} Y^T \Gamma_n^{-1} Y\right)}{\sqrt{(2\pi)^n \det(\Gamma_n)}}$$

- Output is typically given as the log of the likelihood function (convenient)

$$LL = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln\det(\Gamma_n) - \frac{1}{2} Y^T \Gamma_n^{-1} Y$$

- No direct solution - found iteratively (subject to the starting point)
- The 'best' model is the one that maximizes the likelihood

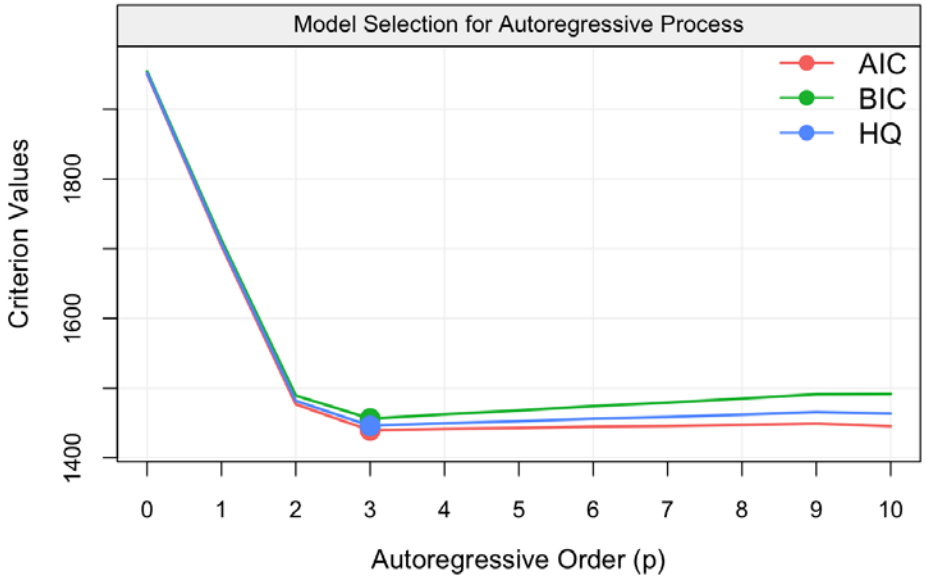
Information Criteria

As with many cases in modeling and machine learning, adding more hyperparameters can lead to a better fit of the data

- But this can often lead to overfitting/poor generalization

Information criteria combines a measure of the goodness-of-fit (like MLE) with a penalty for larger numbers of parameters

- Akaiki Information Criterion (AIC)
- Bayesian Information Criterion (BIC)
- Hannan-Quinn Information Criterion (HQIC)



Information Criteria

Akaike's Information Criterion (AIC)

$$\mathrm{AIC} = -2\log(L) + 2(p+q+k+1),$$

where *L* is the likelihood of the data, and *k* denotes the presence of a constant term in the model (so is either 1 or 0)

- You may see all parameters lumped together as *k* in some notations

$$\mathrm{AIC} \ = \ 2k - 2\ln(\hat{L})$$

Sometimes, when the sample size is small (short time series), a corrected AIC is used to avoid a bias towards too many parameters

$$\mathrm{AICc} = \mathrm{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2},$$

Information Criteria

Bayesian Information Criterion (BIC):

$$\mathrm{BIC} = k\ln(n) - 2\ln(\widehat{L}). \qquad \mathrm{AIC} = 2k - 2\ln(\hat{L})$$

Very similar to AIC

- Both introduce a penalty term for the number of parameters in the model
- The penalty term is larger in BIC than in AIC.

$$\mathrm{BIC} = \mathrm{AIC} + [\log(T) - 2](p + q + k + 1). \qquad \mathrm{AIC} = -2\log(L) + 2(p + q + k + 1),$$

Hannan-Quinn Information Criterion (HQIC):

$$\mathrm{HQC} = -2L_{max} + 2k\ln(\ln(n)),$$

None of these are good for determining the order of *d*

- Differencing changes the data that the likelihoods are computed from

Many packages will "automate" the selection of model parameters empirically, based on search methods that minimize these functions

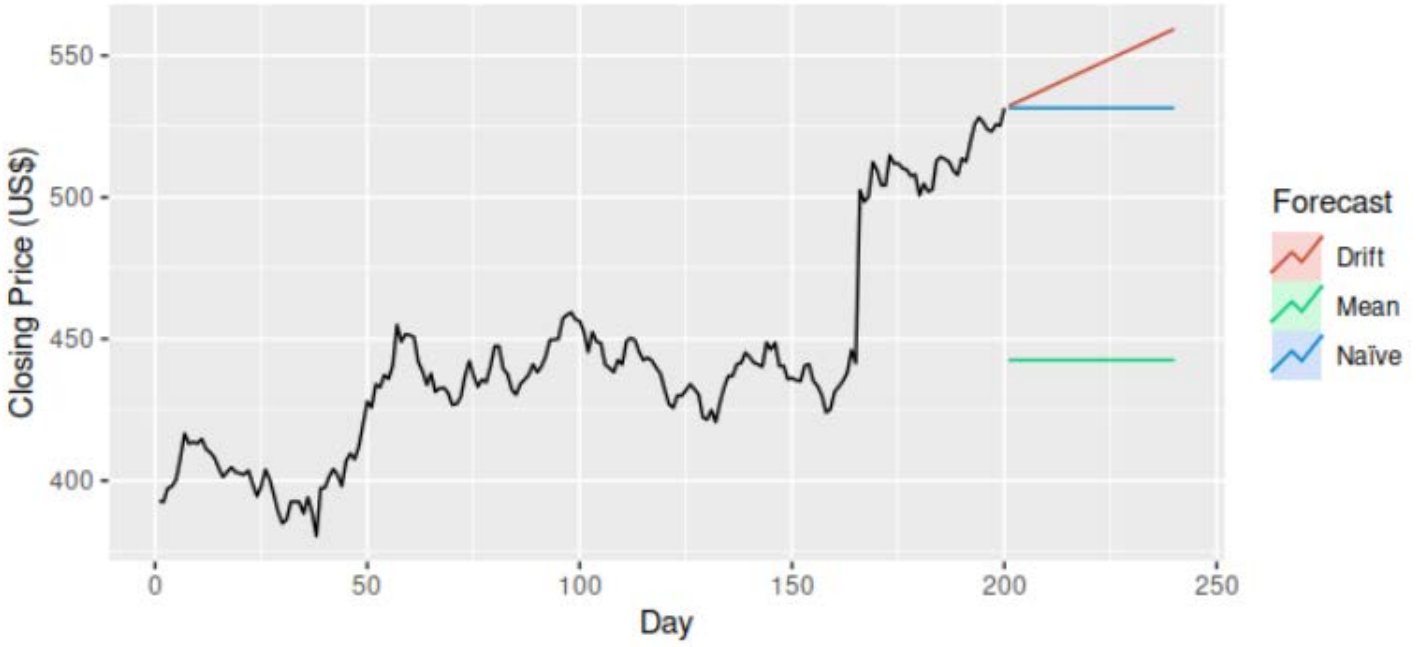
Recap & Recipe for ARIMA

- Visualize the data
- Identify if you need a model that is multiplicative or additive
- Transform data to make it linear (if it isn't already)
- Identify potential components: trend, seasonal, cyclical, residuals
- Make data stationary (if it isn't already)
- Choose ARIMA or SARIMA model
- Use ACF/PACF or other to select model orders
- Identify optimal model based on Information Criteria, like AIC
- Evaluate the residuals using statistical test, ACF/PACF, etc.
- Forecast new data points
- Calculate forecasting error

Forecasting

A main goal of time series modeling is to predict future values.

- Average Method: All future values are equal to the mean of the historical data.
- Naïve Method: All future values are equal to the last known value
- Drift Method: Forecast values follow the identified trend



Forecasting

Another method is exponential smoothing.

- In the naïve method, only the last known value matters, sort of like a weighted average of all points, but with all of the weight placed on the last point.
- The average method effectively does the same, but with all points contributing equally to this weighted average.

- Simple exponential smoothing lands somewhere in between.

- Weights newer points more heavily, with exponentially decreasing weightings for older data.

$$x_{T+1} = \alpha x_T + \alpha(1-a)x_{T-1} + \alpha(1-a)^2 x_{T-2} + \cdots$$

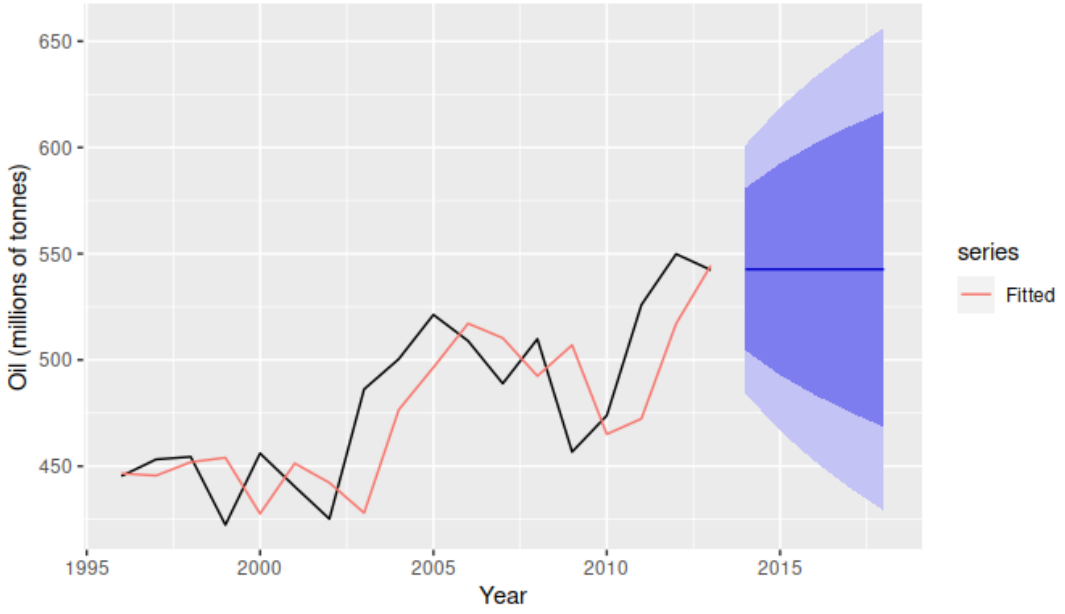
(α is a smoothing parameter, $0 \leq \alpha \leq 1$)

Forecasting

Another method is exponential smoothing .

- Weights newer points more heavily, with exponentially decreasing weightings for older data.
- Many extensions that incorporate trend, seasonality
  - Holt's method
  - Holt-Winters' methods
  - Damped method

- Requires the selection of the smoothing parameter α
- Part of a family of techniques known as ETS modeling
- Has a "flat" projection, unless applied iteratively



Forecasting

We can do the same with our ARIMA models

- Looking at our equation for the ARMA model, we can simply predict the next point, as

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots \phi_p x_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots \theta_q \varepsilon_{t-q} + \varepsilon_t$$

- Assume that ε_t, which is the best we can do, is white noise
- Repeat iteratively for the remaining prediction period

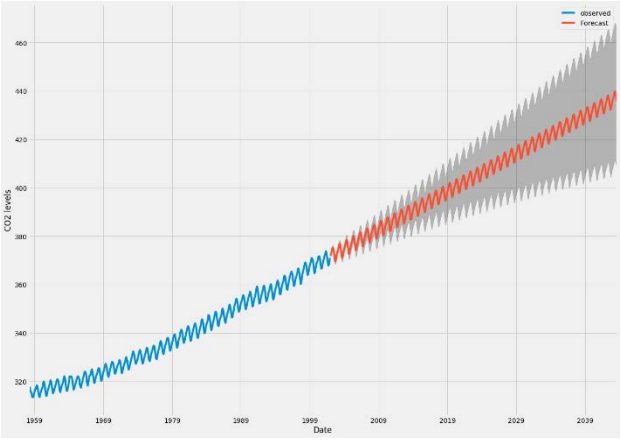
This can also be thought of as replacing t with $t + h$, and then repeating for $h = 1,2,3, \ldots$ until all predictions have been made.

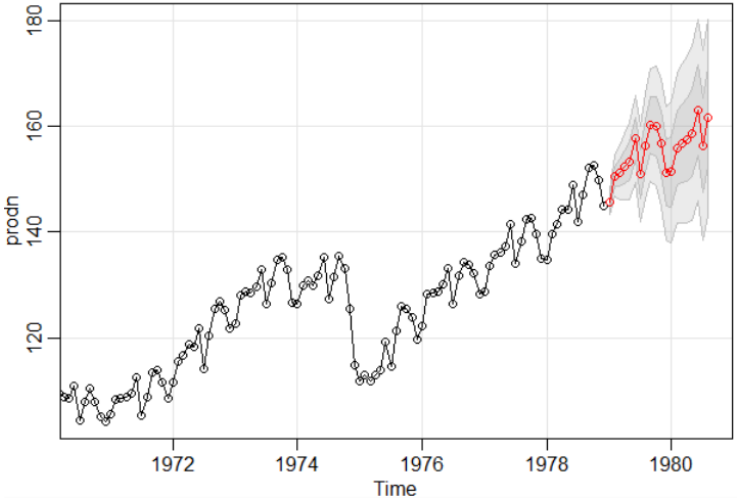
Note that, as h increases, more and more of the values used to predict x_{t+h} become estimates themselves.

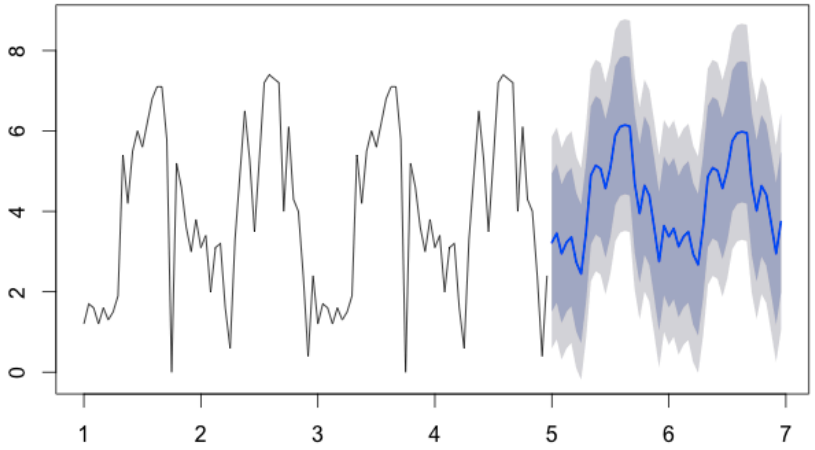
- This leads to a potential accumulation of error, and thus, reduced confidence

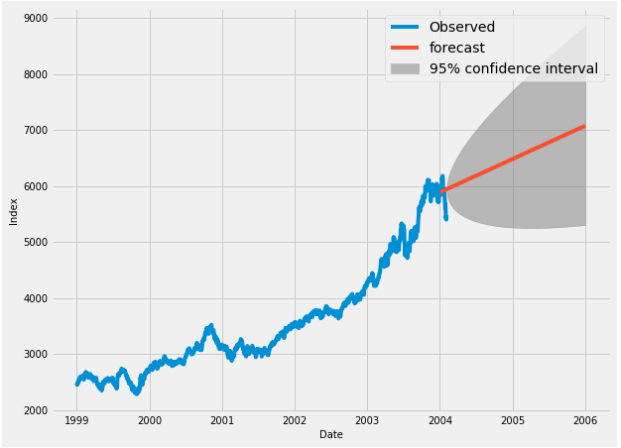
Prediction Intervals

You may have noticed that when we see plots of predictions, we often see a band of color around the prediction points.









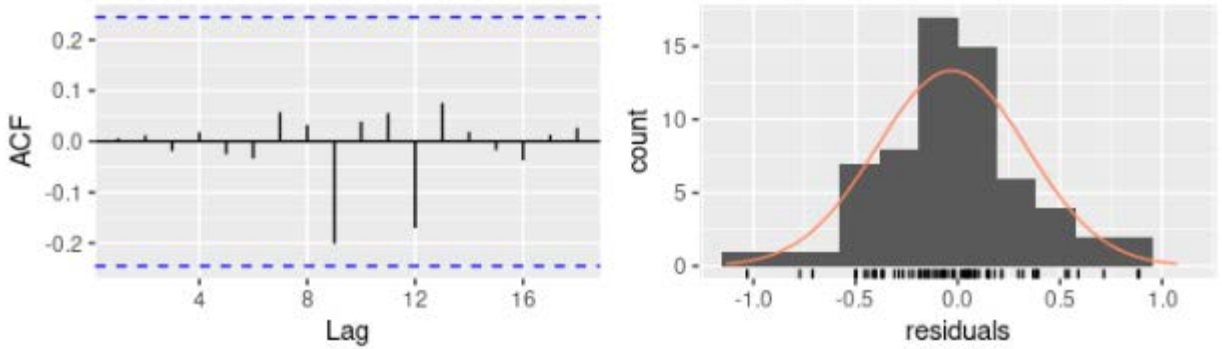
Prediction Intervals

Remember that our time series data are only observations of sequences of random variables

- We never know exactly what the future values will be.
- So, even our point forecasts have some probability of error

It is therefore advantageous to be able to report, not only the predicted value, but our confidence in that value.

- The bands you see are typically the 95% confidence interval around the prediction
- Assumes that the residuals, ε_t, are uncorrelated and normally distributed
- Always check your ACF (and histogram of the residuals) before evaluating the prediction intervals



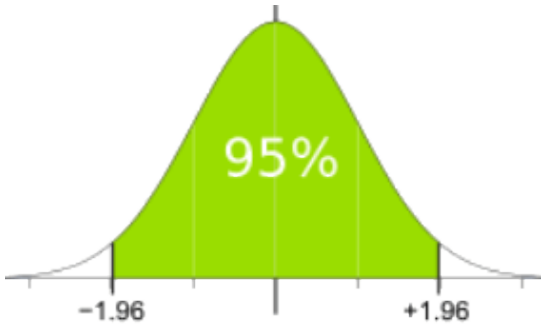
Prediction Intervals

Remember that our time series data are only observations of sequences of random variables

- We never know exactly what the future values will be.
- So, even our point forecasts have some probability of error

It is therefore advantageous to be able to report, not only the predicted value, but our confidence in that value.

- The bands you see are typically the 95% confidence interval around the prediction
- Assumes that the residuals, ε_t, are uncorrelated and normally distributed
- Always check your ACF (and histogram of the residuals) before evaluating your prediction intervals
- If $\hat{\sigma}$ is the standard deviation of the residuals, then the 95% prediction interval of the 1st prediction will always be $\hat{x}_{t+1|t} \pm 1.96\hat{\sigma}$



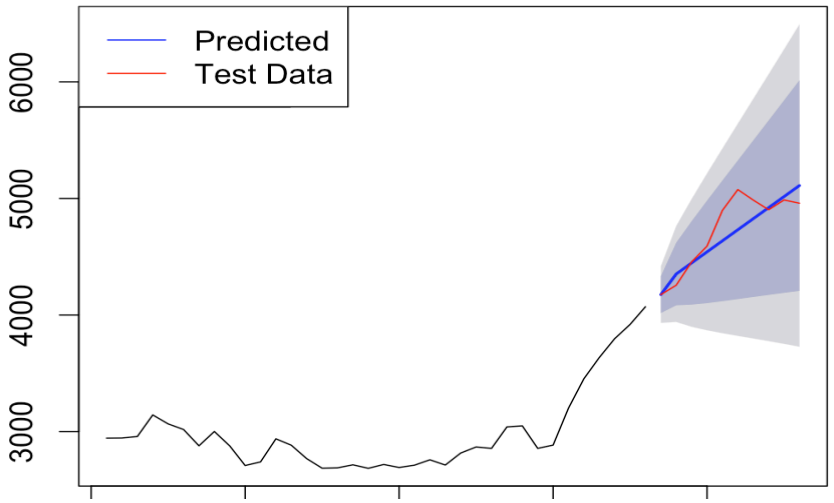
Prediction Intervals

The prediction interval grows for subsequent predictions, compounding the probabilities in the prediction errors

- For stationary models, it eventually converges for long horizons
- Intervals tends to be overly optimistic (narrow) for ARIMA, because errors in parameter estimates and model order are not considered

Confidence intervals can be calculated for any %

- Sometimes bands are shown for more than one %
- Higher the confidence intervals - larger bands



Confidence Interval	Multiplier *C*
80%	1.282
85%	1.440
90%	1.645
95%	1.960
99%	2.576
99.5%	2.807
99.9%	3.291

$$\hat{x}_{t+h|t} \pm c\hat{\sigma}_h$$

Performance Testing

Once we've built a model, we will want to evaluate its performance.

- Like in machine learning, we can evaluate how well a model works given the data used to train it, but that doesn't guarantee that it will generalize.

Fitted Values: The one-step ahead forecasts made during the estimation period. This is the same as the training error in other forms of ML. Gives an indication of how well the model was "fit" to the training data.

- Usually overly optimistic, especially if the model is overfit.
- Errors are the *residuals*

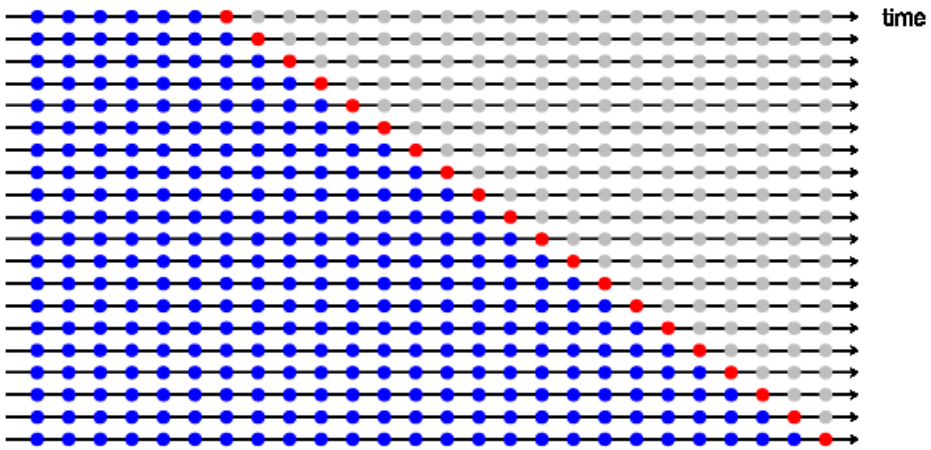
Out-of-Sample/Hold-Out Validation: Holding out part of your data from model estimation to be used later for testing. These data can give you an indication of how your model may perform in the future.

- The amount you hold out depends on the amount of available data (~20% is common, should be at least as long as anticipated forecast length)
- Error is the *forecast error*

Performance Testing

The selection of the test set may be somewhat arbitrary and may not be a great indicator of future performance. It also implies predicting ahead for a long period (with growing uncertainty).

- We may want to evaluate our ability to always predict *h* steps ahead.
- Conduct a series of predictions, always using only the data up to *h* less than the prediction points.
- Similar to *k*-fold cross-validation in ML, but must be careful never to show the model data from *after* the prediction point
- The performance is reported as the average across all predictions

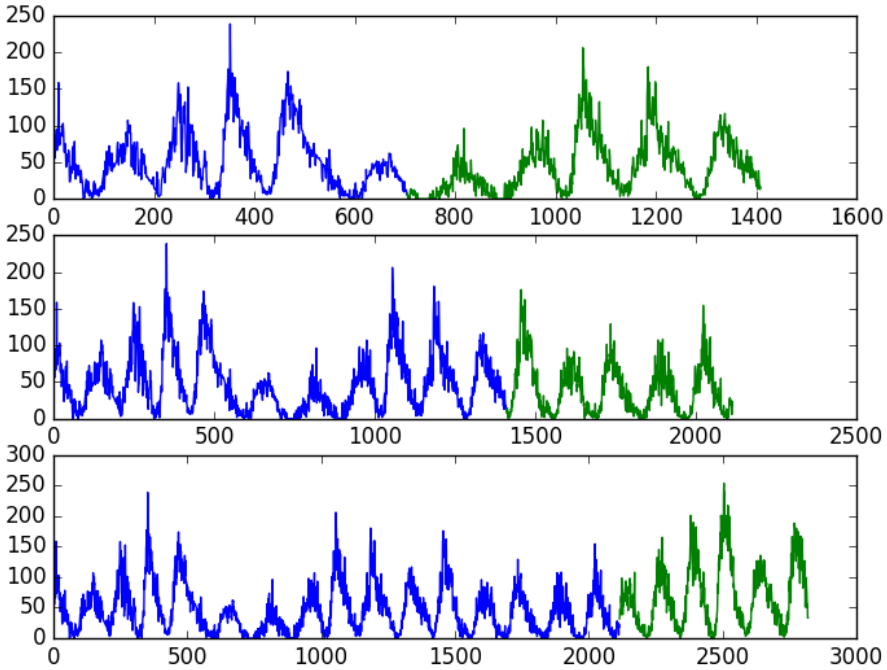


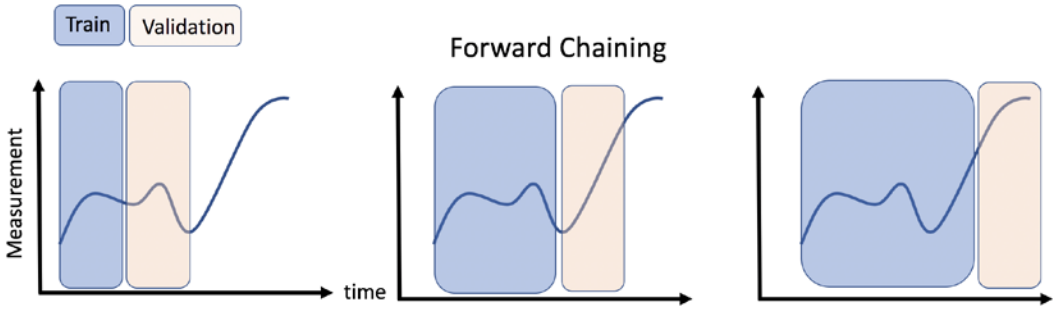
Note: we can still have multiple predictions in each “fold”

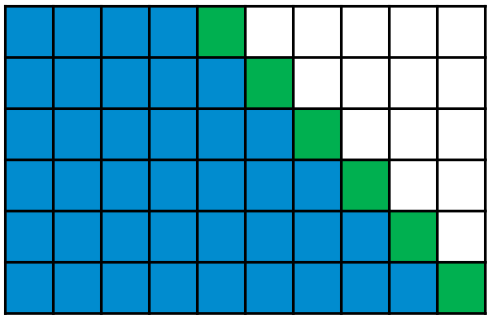
Performance Testing

Expanding Window Forecasts

- To begin, the first N (here 705) are used to train (determine the model parameters & coefficients)
- The model is tested using the next M (here also 705), and the results are saved.
- Those M are then grouped with the training data, and the model is re-trained.
- The next M are tested. Repeat.







https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

Performance Testing

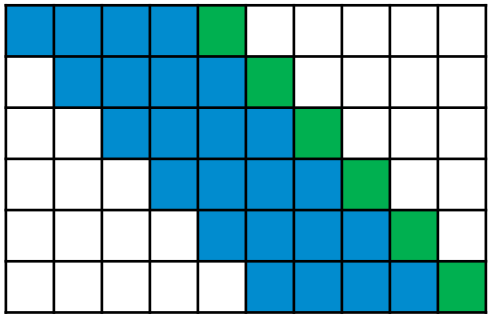
Walking/Rolling Window Forecasts

- Similar to expanding windows, but the size of the training (coefficient re-estimation) window is held constant
- The training window walks/rolls forward, along with the test sample(s)

- Report the $mean \pm std\ dev$ of the prediction error across folds

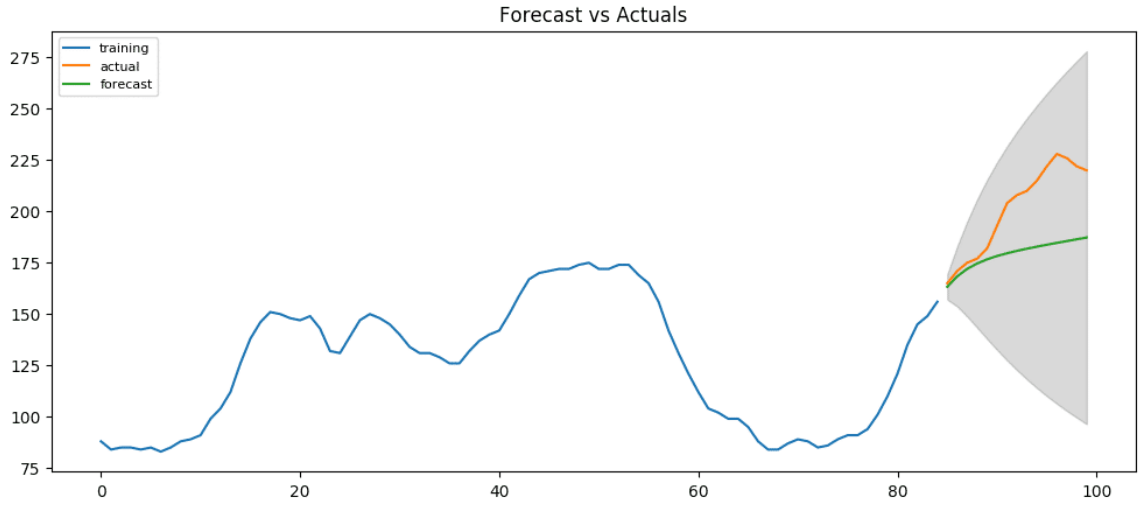
Walking/Rolling Window Forecasts

- Similar to expanding windows, but the size of the training (coefficie
  re-estimation) window is held constant



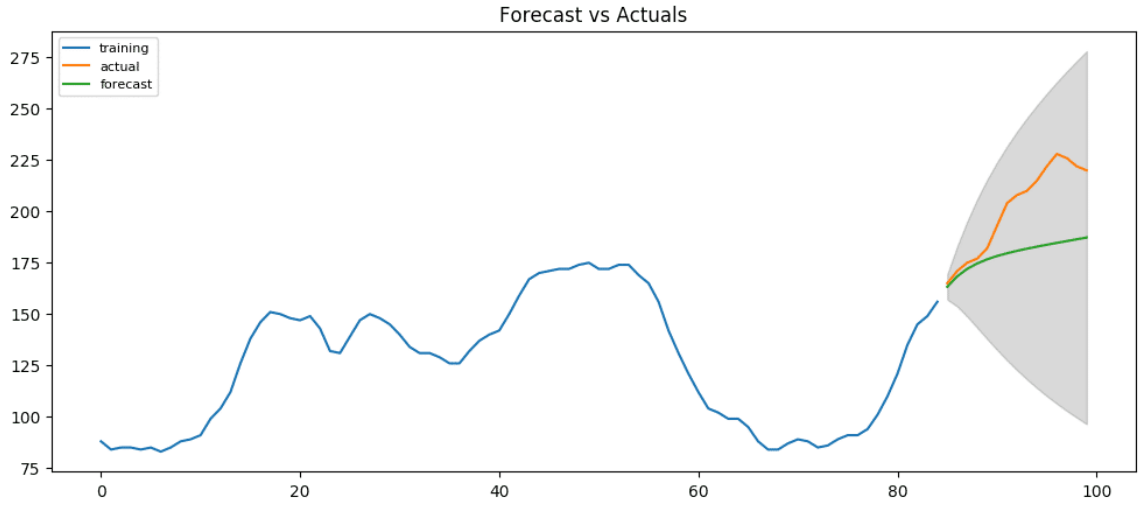
https://medium.com/@pradip.samuel/cross-validation-in-time-series-model-b07fbba65db7

Performance Testing

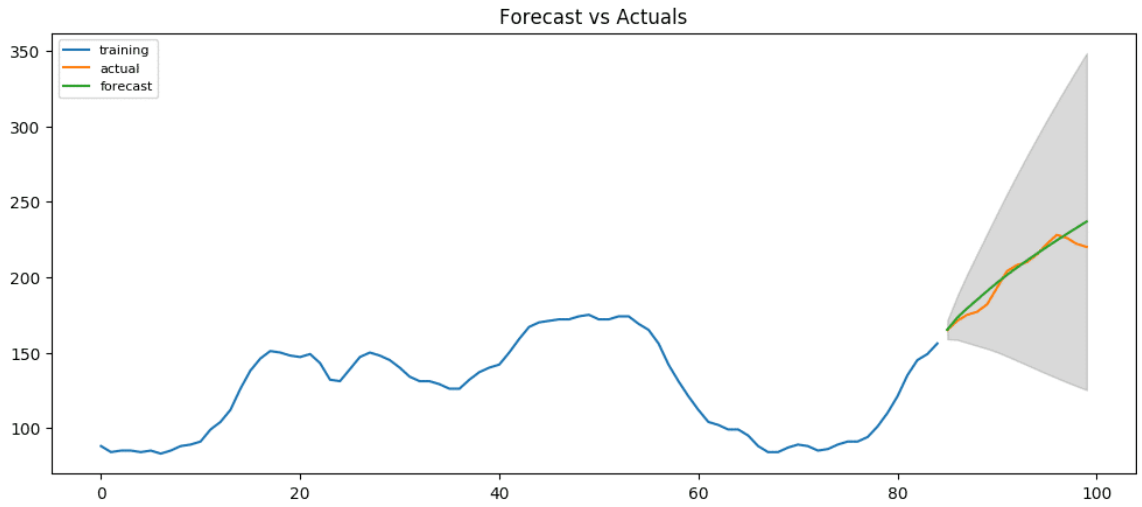


Here, we see a held-out forecast that under-shoots the actual, but still falls within the 95% confidence interval.

Performance Testing



Here, we see a held-out forecast that under-shoots the actual, but still falls within the 95% confidence interval.



Here, we see an updated model that seems to predict much more "closely" to the actual value.

- Let's quantify "closely"

Performance Metrics

Forecast Error (or Residual Forecast Error)

- The forecast error is calculated as the real value minus the predicted value (real – predicted)
- This is called the residual error of the prediction.

$$RE = x_t - \hat{x}_t$$

Mean Error (or Forecast Bias)

- The forecast error values can be +ve or –ve.
- The mean of the forecast error can signify if a forecast model tends to over or underestimate

$$ME = \frac{1}{n}\sum_{t=1}^{T} x_t - \hat{x}_t$$

Mean Absolute Error (MAE)

- Gives a measure of the average amount of error per prediction, regardless of sign

$$MAE = \frac{1}{n}\sum_{t=1}^{T} |x_t - \hat{x}_t|$$

Median Absolute Error (MedAE)

- Similar to MAE, but reduces the influence of outliers
- Can also handle missing values

$$MedAE = median\left\{|x_t - \hat{x}_t|\right\}_{t=1}^{T}$$

Performance Metrics

Mean Absolute Percentage Error (MAPE)

- Same as MAE, but normalized to yield a percentage

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{T}\left|\frac{x_t - \hat{x}_t}{x_t}\right|$$

Root Mean Squared Error (RMSE)

- Similar to MAE, but penalizes larger errors more

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2}$$

Coefficient of Determination (R^2)

- The fraction of the total sum of squares that is explained by the model
- That is, it measures the amount of error, relative to the normal amount of deviation
- Scaled between 0 and 1 (bigger is better)

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$SS_{res} = \sum_{t=1}^{T}(x_t - \hat{x}_t)^2$$

$$SS_{tot} = \sum_{t=1}^{T}(x_t - \bar{x}_t)^2$$

Also, Correlation between signals, using the Lag 1 of the ACF, and many more…

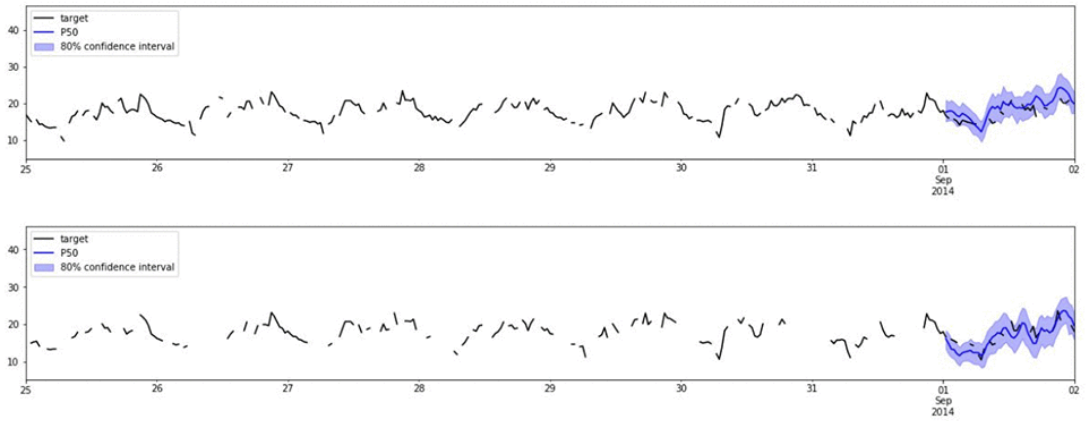
Missing Values

Sometimes, your data may have missing values at certain times.

- Or you've removed some points identified as outliers
- Usually not good to replace the missing values with 0, or the mean of the data (especially if the series isn't stationary)
- Many different approaches, which vary in effectiveness.
- Try out a few of them before making a choice

Some common options include:

- Forward- or backward Fill
- Polynomial Interpolation (Linear, Quad)
- K Nearest Neighbors
- Seasonal Mean



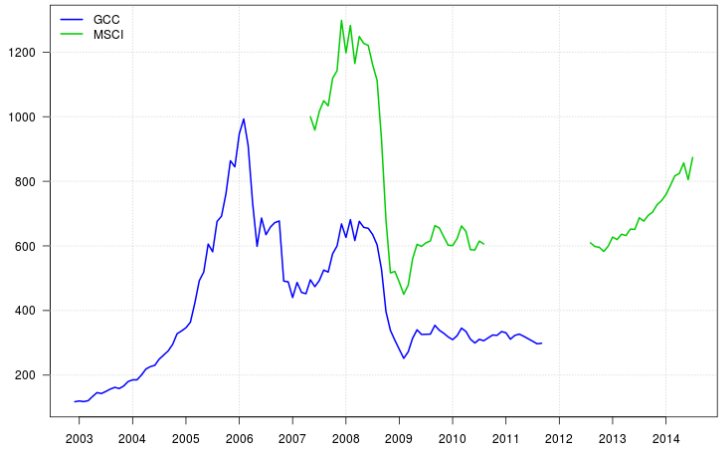
Missing Values

Sometimes, your data may have missing values at certain times.

- Or you've removed some points identified as outliers
- Usually not good to replace the missing values with 0, or the mean of the data (especially if the series isn't stationary)
- Many different approaches, which vary in effectiveness.
- Try out a few of them before making a choice

Some common options include:

- Forward- or backward Fill
- Polynomial Interpolation (Linear, Quad)
- K Nearest Neighbors
- Seasonal Mean



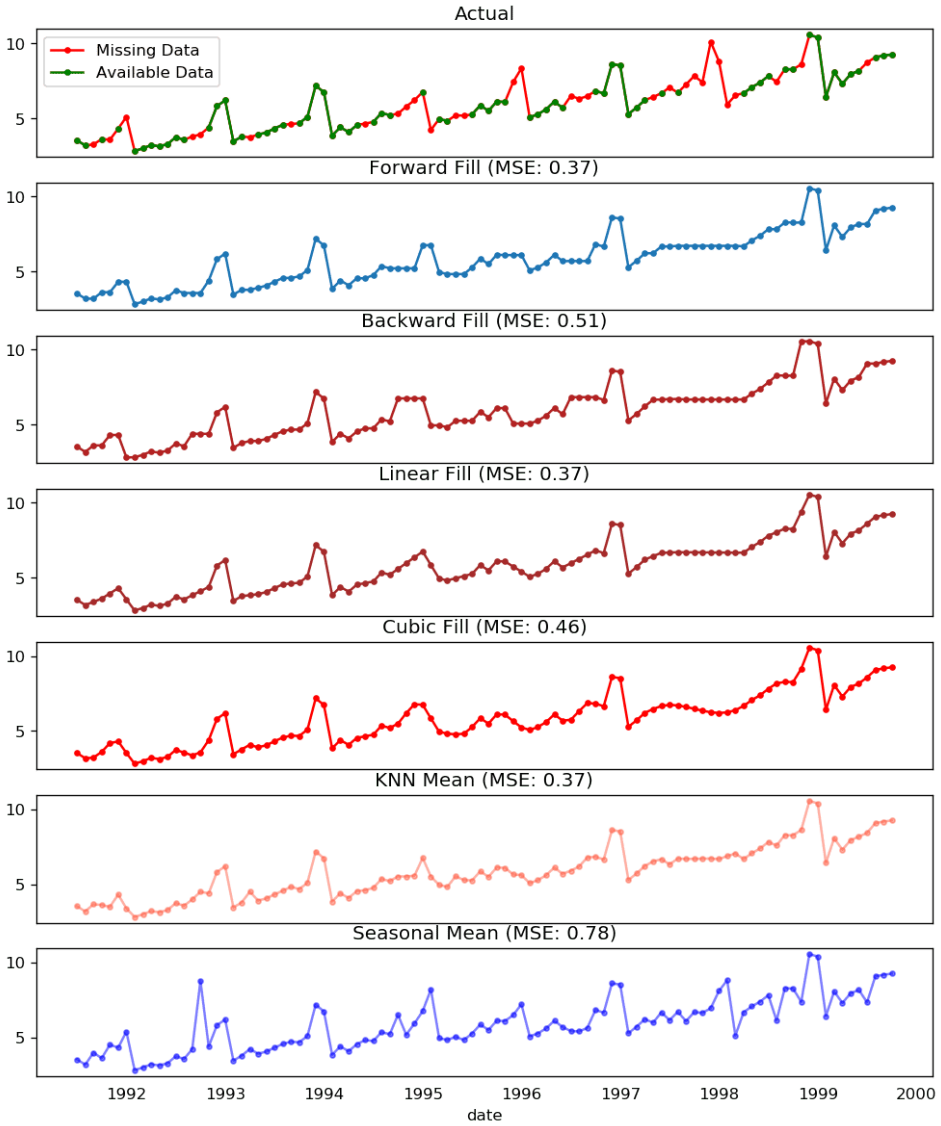
Some more advanced options include:

- Forecast them using past values (if you have enough)
- Forecast them backwards using future values
- Leverage other variables in an ML model

Missing Values

Here, missing data were simulated from an original (full) time series.

- Several different methods were used to replace the missing values
- Performance is shown relative to the original signal using the MSE

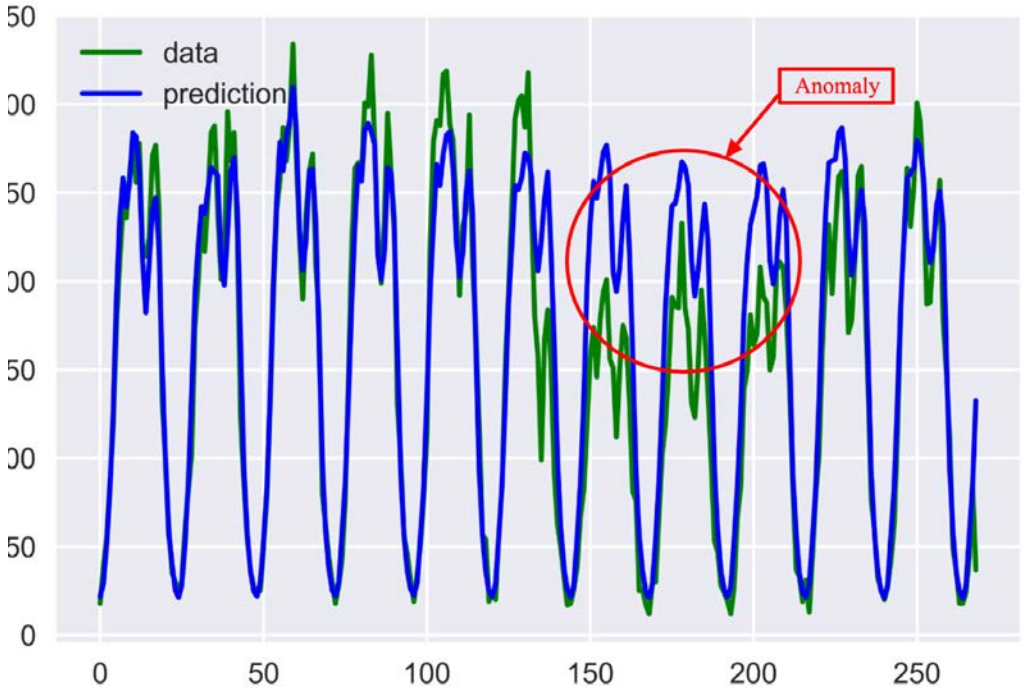


https://www.machinelearningplus.com/wp-content/uploads/2019/02/11_Missing_Value_Treatments-min.png

Anomaly Detection

The concept of anomaly detection has meaningful and practical applications implications

- Detecting unwanted or unexpected activity (e.g. credit card fraud)
- Changes in the underlying state of the world (e.g. a change in health, stock markets, or human machine interface control)



https://blog.statsbot.co/time-series-anomaly-detection-algorithms-1cef5519aef2

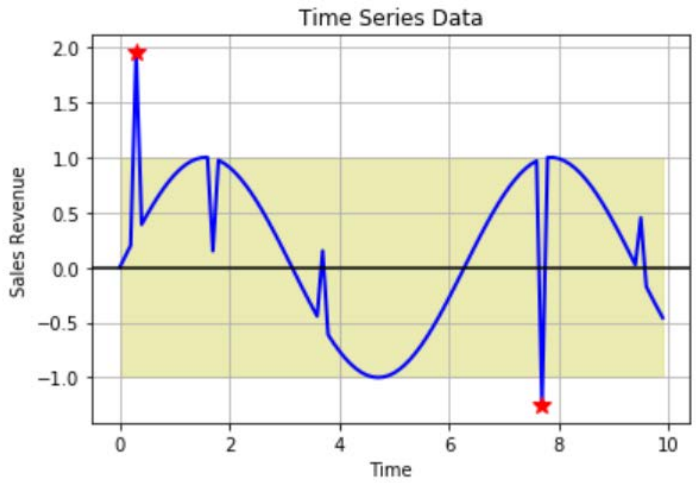
Anomaly Detection

The concept of anomaly detection has meaningful and practical applications implications

- Detecting unwanted or unexpected activity (e.g. credit card fraud)
- Changes in the underlying state of the world (e.g. a change in health, stock markets, or human machine interface control)

Sometimes, in we can simply look for statistical outliers

- If we know the distribution of our data, we can set thresholds
  e.g. a systolic blood pressure over 140 is bad
- Or we can use classifiers with distance measures and rejection options
  e.g. Not like one of my known classes



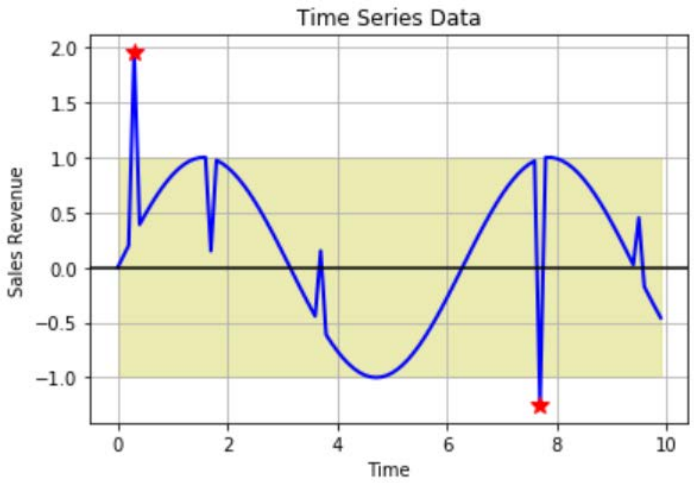
Anomaly Detection

The concept of anomaly detection has meaningful and practical applications implications

- Detecting unwanted or unexpected activity (e.g. credit card fraud)
- Changes in the underlying state of the world (e.g. a change in health, stock markets, or human machine interface control)

Sometimes, in we can simply look for statistical outliers

- If we know the distribution of our data, we can set thresholds
  e.g. a systolic blood pressure over 140 is bad
- Or we can use classifiers with distance measures and rejection options
  e.g. Not like one of my known classes



But, if the data are *autocorrelated*, then we are wasting valuable information
e.g. a spike among lower values, may still be within the “normal” range

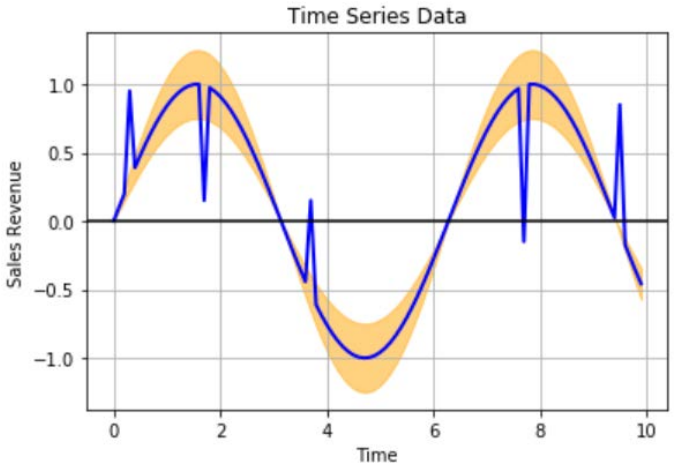
Anomaly Detection

A core goal of ARIMA modeling is to explain the structure in the time series

- This leaves residuals that are random white noise
- When building our models, we look at the residuals to make sure that they are normally distributed
- If they aren't, we assume that there is remaining structure, and we improve our model

But, if we are using a known model and it is working well, we can continue to evaluate the residuals

- If our model is good, our prediction errors should be small, and normally distributed with known variance.
- If a prediction (or series of predictions) suddenly lies outside of these expected ranges, it may signify an anomaly
- It lies outside some confidence interval!



Q&A