**Introduction to Forecasting and Time Series Structure**

Time series models are meant for extrapolation/forecasting

A time series can have the following trends

* Trend 🡪 overall pattern to the data (can be linear, quadratic, positive, negative)
* Season 🡪 Trend by season. A systematic up and down pattern that can follows seasons
* Cycle 🡪 different than season, and is a trend with varying lengths
  + Time series can also have multiple patterns within a dataset, which can be viewed as nested time series

Time series = Signal + noise

* Signal 🡪 explained variation (trend, cycle, seasonality)
  + Forecasts extrapolate signal portion of model
* Noise 🡪 unexplained variation (error)
  + Confidence intervals account for uncertainty

Time Series Decomposition

* If a time series only has trend/cycle patterns, then there is no need to decompose
* If a time series has both trend/cycle patterns and seasonal variation, then we decompose series into three parts
  + Tt 🡪 trend/cycle patters
  + St 🡪 seasonal variation
  + Rt 🡪error
* Additive 🡪 Yt = Tt + St + Rt
  + Magnitude of variation around the trend/cycle will remain constant
* Multiplicative 🡪 Yt = Tt \* St \* Rt
  + Magnitude of the variation around the trend/cycle proportionally changes (cornucopia shape)
  + To transform times series from multiplicative to additive, do a log transformation
    - Log(Yt) =log( Tt ) + log(St) + log(Rt)
    - Log transformation will stabilize the variance

STL graphs the original data, trend, seasonality, and remainder

* If you add trend + seasonality + remainder you should get the original data point

Seasonally adjusted data

* One advantage of time series decomposition is that we are able to create seasonally adjusted data
  + i.e. remove the “effect” of seasonality
* This allows us to understand the trend of the series
* Seasonally adjusted additive 🡪 Yt - St = Tt + Rt
* Seasonally adjusted multiplicative 🡪 Yt/St = Tt \* Rt

Decomposition Techniques

1. Classical Decomposition
   1. Trend 🡪 uses moving/rolling average smoothing
   2. Seasonal 🡪 average de-trended values across seasons(assumed to be constant throughout series)
   3. Can do additive or multiplicative time series
2. X-11 ARIMA Decomposition
   1. Trend 🡪 uses moving/rolling average smoothing
   2. Seasonal 🡪 uses moving/rolling average smoothing
   3. Iteratively repeats above methods and ARIMA modeling
   4. Can handle outliers
   5. Automatic (will chose better of either additive or multiplicative)
   6. Is what the government uses
3. STL ( Seasonal and Trend using LOESS estimation) Decomposition
   1. Default of STL function in R
   2. Using LOcal regrESSion techniques to estimate trend and seasonality
   3. Allows changing effects for trend and season
   4. Adapted to handle outliers

Cautions on decomposition

* You must have more than one observation per year in order to decompose a dataset
* Decomposition will not tell you if you have seasonal data (nor the length of the seasonality)

Measures for Strength of trend/season

* Values of F close to 1 indicate high strength and values close to 0 indicate low strength
* Ft 🡪 strength of trend
* Fs🡪 strength of seasonality component
  + Better to use Fs to determine seasonality component versus looking at the graph

Types of Imputation

* More than 20% of data missing could be an issue
* Mean 🡪 uses the mean of the time series (not a good imputation method for time series data)
* Locf 🡪 last observation carried forward
  + NOCB 🡪 next observation carried backwards
* Spline 🡪 uses spline to impute missing values (could also be linear, quadratic, etc.)
  + Uses previous and next point to male a curve to predict missing value
* Seadec 🡪Seasonally decomposed imputation
  + Does a decomposition and removes the seasonal component, imputes data, and then puts seasonal component back in)
  + Good for seasonal data

**Evaluating Forecasts**

Accuracy of forecast depends on your definition of accuracy, as well as it is different across fields/industries

Good forecasts should have the following characteristics

* Be highly correlated with the actual series values
* Exhibit small forecast errors
* Capture the important features of the original time series

Judgement forecasting

* When using data, forecasts are found using a quantitative approach. However, there are instances where models are not available and a qualitative or judgement forecast is used instead
* Occasionally a qualitative and quantitative approach are used together

Accuracy versus goodness of fit

* A diagnostic statistic calculated using the same sample that was used to build the model is a goodness-of-fit statistic (i.e. AIC, BIC, sometimes MAPE, MAE, etc.)
* A diagnostic statistic calculated using a hold out sample that was not used in the model building is an accuracy statistic

Hold-Out Sample 🡪 a hold out sample in time series analysis is different than cross-sectional analysis

* Hold out sample is always at the end of the time series, and doesn’t typically go beyond 25% of the data
* If you have a seasonal time series, then at least an entire season should be captured in the hold-out sample

1. Divide the time series into two or three segments – training and validation and/or test
2. Derive a set of candidate models
3. Calculate the chosen accuracy statistic by forecasting the validation set
   1. Only report accuracy on the test dataset
4. Pick the model with the best accuracy statistic
5. Provide the accuracy of the model on the test dataset (recommend combining train and validation to get the most updated parameters, and at this point you will not change the model

Model Diagnostic Statistics

* Mean Absolute Percent Error (MAPE)
  + \*(1/n)
  + Resulting number is a percent
  + Difference between actual and forecast
  + Percent off on average
  + Problems
    - Overweight of overpredictions
    - If there is an observation of 0 ( as 0 is in the denominator of the equation)
* Mean Absolute Error (MAE)
  + Abs(observed – predicted) \*(1/n)
  + How far off are you on average
  + Problems
    - Not scale invariant
    - Should also report S.D. (as 10 units could mean different things for different datasets/scales)
* Square root of Mean Square Error (RMSE)
  + Not sure what to put here
  + Problems
    - Overweight of large errors
    - Not scale invariant
      * Make sure to give S.D. when using RMSE
* Symmetric Mean Absolute Percent Error
  + Not sure what to put here

**Exponential Smoothing Models (ESM)**

Time series data relies on the assumption that the observations at a certain point of time depend on the previous observations in time

Naïve model 🡪predicting the next point by using the most recent point of data

Average Model 🡪 predicting the next point by taking the average of all previous points

Exponential Smoothing 🡪a weighted average of previous data points to predict a future data point

* Models only require a few parameters
* Equations are simple and easy to interpret
* Good for “one-step ahead” forecasting
* Types of ESM models we will learn about
  + Single Exponential Smoothing (SES)
    - No Trend component
    - No Season Component
  + Linear/Holt Exponential Smoothing
    - Has Trend component
    - No Seasonality component
  + Holt-Winters
    - Has Trend Component
    - Has Seasonality Component

Single Exponential Smoothing (SES)

* Equates the predictions at time t equal to the weighted values of the previous time period along with the previous time periods prediction
* We can apply a weighting scheme that decreases exponentially the further back in time we go
  + As we go further back in time, the weights decrease exponentially (more weight is put on the more recent observations)
* The typical method for calculating the optimal value of alpha in the Exponential smoothing model is through one step ahead forecasts
* The value of alpha that minimizes the one step ahead forecast errors is considered the optimal value
* For SES, we have no season, no trend, and we will start with an additive error
  + Error is allowed to be additive or multiplicative, and most older models assume additive error)

Trending Exponential Smoothing (an overall category)

* The Single Exponential Smoothing models are better used for short term forecasts
* The SES model cannot adequately handle data that is trending up or down
* There are different ways to incorporate a trend in the Exponential Smoothing Model
  + Linear / Holt Exponential Smoothing
  + Damped Trend Exponential Smoothing

Linear /Holt Exponential Smoothing (incorporates trend)

* The Linear Exponential Smoothing Model has two components
  + First component is from the SES model
  + Second component incorporates trending into the model
  + Both parameters are smoothing or weight parameters
* Damped Trend Exponential Smoothing
  + From my understanding this is a subset of Linear/Holt Exponential Smoothing

Holt-Winters / Triple Exponential Smoothing (incorporates trend and season)

* Can be adapted to account for seasonal factors
* Can be additive or multiplicative in the seasonal effect in the Exponential Smoothing Model
  + Holt Winters Additive Exponential Smoothing (includes trend)
  + Holt Winters Multiplicative Exponential Smoothing (incudes trend)
* In season exponential smoothing, weights decay with respect to the seasonal factor
* The Linear Exponential Smoothing model has three components
  + Level, trend, and season (and a different weight for each component)

**More ETS**

* ETS 🡪 Error, Trend, Season
* ETS can automatically select best model (based on certain information criteria)
  + For “Error”, the choices are Additive (A) or Multiplicative (M)
  + For “Trend”, the choices are None (N), Additive (A) or Additive Damped (Ad)
  + For “Season”, the choices are None (N), Additive (A) or Multiplicative (M)
  + You can choose which one you want, or you can let the computer choose
  + Default is using the AICc to choose the best model
* Random notes
  + In the fable package, the dampening parameter is restricted to be between 0.8 < phi < 0.98
  + You are not restricted to the default of AICc (and could use AIC or BIC)
  + Some models may create division by 0 and are not included in the search algorithm
    - ETS(A,N,M), ETS(A,A,M), and ETS(A,Ad,M)
  + If data contains any negative values, multiplicative errors will not be considered

**ARIMA**

ARIMA 🡪 AutoRegressive Integrated Moving Averages

Signal

* Can have signal due to seasonal pattern (visible)
* Can have signal due to trend (visible)
* Can have signal due to “correlation structure” which can be in the form of AR and MA

ARIMA(p,d,q)

* p = number of AR terms
* d = number of differences in the data taken
* q = number of MA terms

Stationary 🡪 to model the AR and MA terms, we must first have stationary

* in other words, the statistical properties do not depend upon time
  + constant mean
  + constant variance
  + constant correlation structure
* this means no visible trend and no visible seasonality

Random Walk 🡪 The next value of Y only depends on the previous value of Y

* If a random walk exists, we need to take the differences of the series

Unit Route 🡪 Used to tell if the data is stationary, and if so how many differences need to be taken

* Augmented Dickey-Fuller Unit Route test
  + Used less often by the IAA
  + Ho: there exists a random walk
  + Ha: the series is stationary
* KPSS test (the one we usually use)
  + Ho: the series is stationary
  + Ha: there exists a random walk
  + Ndiffs 🡪 tells how many differences need to be taken, and if 0 then the data is stationary

Over-differencing 🡪 when you take a difference and you didn’t need to. This introduces more dependence on error terms in the model

Autocorrelation function 🡪 is the correlation between two sets of observations, from the same series, that are separated by K points in time

* The autocorrelation function (ACF) is the function of all autocorrelations (between two sets of observations Yt and Yt-k) across time (for all values of K)
* Suppose that the first autocorrelation value (ACF(1)) is significant
  + This implies that two consecutive time points are related to one another
  + March is related to April, April is related to May, etc....
* Autocorrelation function 🡪 can be in both a positive and negative direction
  + Positive 🡪 High Mondays imply high Tuesday
  + Negative 🡪 High Monday imply low Tuesday

Partial Autocorrelation Function 🡪 is the correlation between two sets of observations, from the same series, that are separated by K points in time, after adjusting for all the previous (1,2, k-1) autocorrelations

* Partial autocorrelations are conditional correlations
* The partial autocorrelation function (PACF) is the function of all partial autocorrelations (between two sets of observations Yt and Yt-k) across time (for all values of k)
* The partial autocorrelation function tries to measure the direct relationship between two sets of observations, without the influence of other sets of time in between
* If you difference the values, then make sure to use the differenced values in the correlation graphs
* Partial autocorrelation at lag = 1 is equal to the autocorrelation, as there is no timepoint in between to remove

**AR and MA Models and White Noise**

PACF 🡪 AR: wherever the last significant spike is, that is the AR term to try first

* The ACF decreases exponentially as the number of lags increases
* The PACF has a significant spike at the first lag, followed by nothing after

ACF 🡪 MA wherever the last significant spike is, that is the MA term to try first

* The ACF has a significant spike at the first lag, followed by nothing after
* The PACF decreases exponentially as the number of lags increases

AR 🡪 autoregressive model that forecasts a series solely based on the past values of Yt

* AR(p) 🡪 A time series that is a linear function of p past values plus error is called an autoregressive process of order p
  + No consistent pattern (potentially see spikes up to lag p in the PACF)
  + Restrictions on φ
* AR(1)
  + Yt = ω + φYt-1 + et
    - ω = intercept
    - φ = AR coefficient (not interpretable), but can tell correlation for AR(1) model
* AR(p) model
  + Yt = ω + φ1Yt-1 + φ2Yt-2 + … + φpYt-p + et
* Data must be stationary to fit AR() terms
* Correlation functions? Page 2
* If φ= 1, then it is a random walk
* There are some restrictions on φ

MA 🡪 Moving average model that forecasts a series solely based on the past error values

* MA(q) 🡪 a time series that is a linear function of q past errors is called a moving average process of order q
  + The ACF for an MA(q) has significant spikes at lags up to lag q, followed by nothing after (kind of exponentially decreasing for PACF)
* MA(1)
  + Yt = ω + et + ϴet-1
  + Needs to be invertible
  + MA(1) is going back 1 error term

General notes about MA/AR models

* Any AR(p) model can be re-written as MA(∞)
* If the MA(q) model is invertible, then this MA(q) model can be rewritten as an AR(∞)

White Noise

* The goal of modeling a time series is to be left with white noise residuals in the time series
  + If there are still spikes in the ACF/PACF, then more modeling should be done
* If we are successful in removing all the “correlation” signals, we are left with independent errors
* A white noise time series has errors that follow a normal distribution with a mean of 0 and a positive, constant variance in which all observations are independent of each other
  + The autocorrelation and partial autocorrelation functions of the residuals from these models have a value close to zero at every time point (should not see any significant spikes)
* There are two ways to check that there is not any significant correlation left in the model
  + Graphs: look at the ACF/PACF plots of the residuals, and there should be no significant spikes
  + Formal: run a Ljung-box test on the residuals

Ljung-Box chi square test for white noise

* Can be applied to the original data or to the residuals after fitting a model
* Ho: no autocorrelation
* Ha: one or more autocorrelations up to lag m are not zero