General Forecasting Models

# Introduction

* The basic steps in Forecasting Time Series are:
  + Choose a model
  + Split the data (train-test)
  + Fit model on train set
  + Evaluate on test set
  + Refit on entire set
  + Forecast for future data
* What covered in this document:
  + ACF and PACF (Autocorrelation functions)
  + Autocorrelation (AR)
  + Descriptive statistics and Tests
  + Choose ARIMA orders
  + ARIMA based models

# Introduction to Forecasting

* Typically it is a good practice to have your test set towards the end of your timeseries data (i.e. the most recent)  
  
* Typically the test set is 20% of the total sample
* One important thing to consider is that your test set should be at LEAST as large as the maximum forecast horizon required
  + So if you need to forecast 1 month ahead then your Test Set should be at least 1 month in duration. If your split is 80-20 the this means that you need at least 5 months of data to train and test
  + Of course it is expected that the longer the test horizon the less accurate your prediction will become
* Since we have this peculiarity of sequential data needed for time series then we manually have to define the borders of our train and test set (i.e. we must grab the first 80% of the data for training and the last sequential 20% for testing)
  + Find the total number of entries by  
    df.info()
  + Find the location of 80% of your rows and say:  
    train\_data=df.iloc[:EndLocation] where EndLocation is an integer of 80% of the total length of your dataset
  + Assign your test dataset:  
    test\_data=df.iloc[EndLocation-1:]

# Holt Winters forecasting

* We use the statsmodels.tsa.holtwinters package to fit the model  
  from statsmodels.tsa.holtwinters import ExponentialSmoothing
* Apply your fit on the train data via:  
  fitted\_model=ExponentialSmoothing(train\_data[ColumnToFit], trend= ‘mul’, seasonal=’mul’,seasonal\_periods=12).fit()  
  where ColumnToFit is the column in the variable we wish to forecast its value. (i.e. if multivariate make it univariate fitting)
* The above line essentially takes the multiplicative trend and since it is a year season it specifies the seasonal period length as 12 months
* Evaluate your fitted model on the test data we specified before:  
  test\_predictions=fitted\_model.forecast(NumberOfPeriods)  
  where NumberOfPeriods is the number of units ahead we wish to forecast. If our model represents one month per row then this means that NumberOfPeriods is essentially how many months ahead do we wish to forecast.
* NB: If you get warnings ahead of evaluating your fitted model on the test data it is fine. Statsmodels updates that information internally and you should not worry

# Evaluating Forecast Predictions

* Up until this point we have applied fitting on the train data and evaluated the fitted model on the test data. The fitted model created predictions on unseen data
* After the evaluation we can compare the accuracy visually by plotting our train data, test data and predictions of test data
* However, visually we can only briefly see the accuracy of the model thus we need to find evaluation matrices that quantify the accuracy
* Accuracy, recall and precision evaluations are not useful for time series data because we need metrics designs for continuous values
* There are three metrics for continuous values available:
  + Mean Absolute Error (MAE)
  + Mean Squared Error (MSE)
  + Root Mean Square Error (RMSE)
* Notation:
  + represents the real value of the set (i.e. the one that exists in the data frame)
  + represents the predicted value from our forecast model