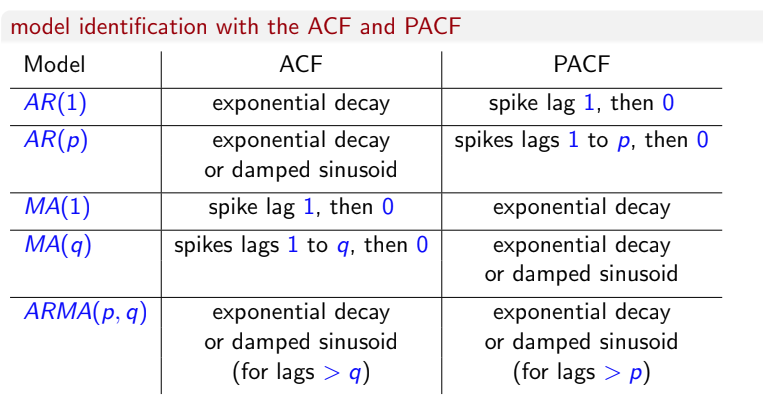
Time Series Model Outline

* Data Pre-processing:
  + Read table
    - Make sure each feature name in table is the same as shown in the code
    - Do not change any feature name!
    - Make sure all order quantity >0
  + Select MasterSKU
    - Remove MasterSKU that is less than Two years old
      * At least One-year data for testing
    - Find start time of each MasterSKU
  + Replace strange Requested\_Date by OrderDate
    - To find why this case happened
    - May have more ‘Date’ problems, fix them before modeling
  + Combine data to extend Product History
    - To add history, use the order history of other product
    - Before filling, make sure the product has similar trend
  + Combine data
    - To add more data, combine products from the same line
      * E.g. Tundra 105 – Tundra 350
  + Select Stable STD
    - Determine the period of analysis
    - Definition of Stable STD: order products in each year (at least once per year)
    - If the last year is incomplete, Stable STD also need to have order history in this year unless the year takes less than three months into account
    - Warning: we should not use any information from testing set, so the time interval we use to select stable STD should from ‘2014-01-01’ to ‘2017-05-31’ (Not ‘2018-05-31’)
* Segment:
  + Segment Method One: by all products
    - Calculate average order quantity for each stable STD
      * overall quantity / months they stay
    - Segment STD into two group: L and S
    - Re - Segment STD in S into two group: M and S
  + Segment Method Two: by Product Category
    - Some STD only order Hard Cooler and Drinkware, so it is incorrect to use overall quantity
    - For each product Category, segment STD using average order quantity
      * E.g. Hard Cooler-Class S, M, L; Soft Cooler-Class S, M, L; Drinkware-Class S, M, L
  + Check the correlation between order quantity and class size
    - Calculate the order quantity of each class every month and calculate the correlation
    - In our assumption, the order quantity should be significant different in different class
* Modeling:
  + Data select: according to start time of each MasterSKU, we use different time interval (Monthly / Weekly)
    - ‘Monthly’ MasterSKU: the product has been launched for more than 3.5 years
      * Change specific date into month, set ‘cut time’ to separate data into training and testing
    - ‘Weekly’ MasterSKU: the product has been launched less than 3.5 years
      * Change specific date into week, set ‘cut time’ to separate data into training and testing
    - Fill nan: time series contains missing order quantity in some month or week because the order quantity is zero during that time interval. We need to fill 0 into these missing values.
      * Using ‘Roadie 20’ in Class S to find these missing value (need to make sure ‘Roadie 20, Class S’ has order every week)
    - Use function: testing\_Monthly () and testing\_Weekly () to help you know which MasterSKU has missing value
  + Date explore: function basic\_Monthly () and basic\_Weekly () show all detailed information from time series
    - White noise testing: in this hypothesis test, we assume this time series is white noise, in other word, there is no information contained in this time series. If P-value is smaller than 5%, we can reject this hypothesis, this time series has information.
    - Dickey\_Fuller Test: test this time series is stationary or not. If p-value is less than 5%, we can say this time series is stationary.

(If p-value above is large, we need to take difference to make time series stationary)

* + - ACF & PACF (Weekly MasterSKU has too many data points, only the top 60 data are shown):



* + - Order Value select
      * ARIMA has 3 order values; SARIMA has 7. We need to determine the order value first.
      * Combination of different (p, d, q)/ (p, d, q, P, D, Q, S) has been considered
      * Compare the AIC, BIC and the symmetric mean absolute percentage error for each combination
      * Choose the combination with lowest AIC or BIC value (We select the model that minimizes the information loss among the candidate models)
        + AICc (When the sample size is small (n/K<40; size n, # of parameter K), there is a substantial probability that AIC will select models that have too many parameters, i.e. that AIC will overfit)
        + ;  (<https://en.wikipedia.org/wiki/Akaike_information_criterion>)
        + When fitting models, it si possible to increase the likelihood by adding parameters, may result in overfitting. Both AIC and BIC attempt to resolve this problem by introducing a penalty term for the number of parameters in the model; the penalty term is larger in BIC than in AIC
        +  (<https://en.wikipedia.org/wiki/Bayesian_information_criterion>)
    - ARIMA/ SARIMA output
      * Residual check
        + QQ plot of residuals should looks like a straight line, in other word, the residual is normally distributed.
        + White noise check: p-value should be large (at least larger than 5%)
        + Prediction result plot and 95% confidence interval (Only in ARIMA model)
        + RMSE, SMAPE and MAPE are used to measure the accuracy of modeling
        + 
    - Baseline: Prophet and Naïve model
      * Using result from Facebook time series package: prophet and Naïve model as baseline
      * Output MAPE and SMAPE and prediction result plot