# DATA 624
# Time Series Decomposition

Corey Arnouts, Humberto Hernandez, Chester Poon, Mike Silva

# Introduction

- Time Series analysis looks to separate underlying patterns in the data series from randomness. Such patterns can then be projected into the future (Forecast)

- **Decomposition** is the process of breaking down into sub patterns to identify the component factors that influence each of the values in a series

# Introduction

The two most significant components of an underlying pattern, characterizing economics and business series, are:

- **Trend Cycle** - represents long term changes in the level of series
- **Seasonal Factor** - is the periodic fluctuations of constant length that is usually caused by known factors such as weather seasons, month of the year, temperature, holidays, etc.

# Decomposition Model

Mathematical representation of the decomposition approach is:

$$Y_t = f\,(S_t,\ T_t,\ E_t)$$

$Y_t$ is the time series value (actual data) at period t.

$S_t$ is the seasonal component ( index) at period t.

$T_t$ is the trend cycle component at period t.

$E_t$ is the irregular (remainder) component at period t.

# Decomposition Models - Classical

The exact functional form depends on the decomposition model actually used. Two simple approaches that are considered classical decomposition are:

Additive Model

$$Y_t = S_t + T_t + E_t$$

Multiplicative Model

$$Y_t = S_t \text{ x } T_t \text{ x } E_t$$

Seasonal Adjustment

$$Y_t - S_t = T_t + E_t$$

$$Y_t / S_t = T_t \text{ x } E_t$$

Most other forms of decomposition stem from classical decomposition

# Decomposition Models - Classical

Disadvantages:

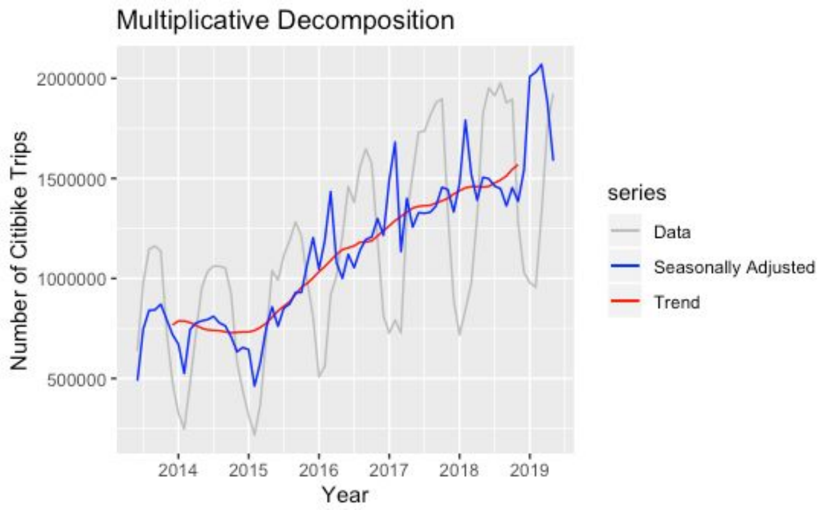
- Trend-cycle estimates for the first and last few observations are unavailable
- Trend-cycle over-smooths sudden drops or rises in the time series data
- Assumptions that the seasonal pattern remains static year after year. Can be a disadvantage as trends change over time
- Not robust to sudden random events

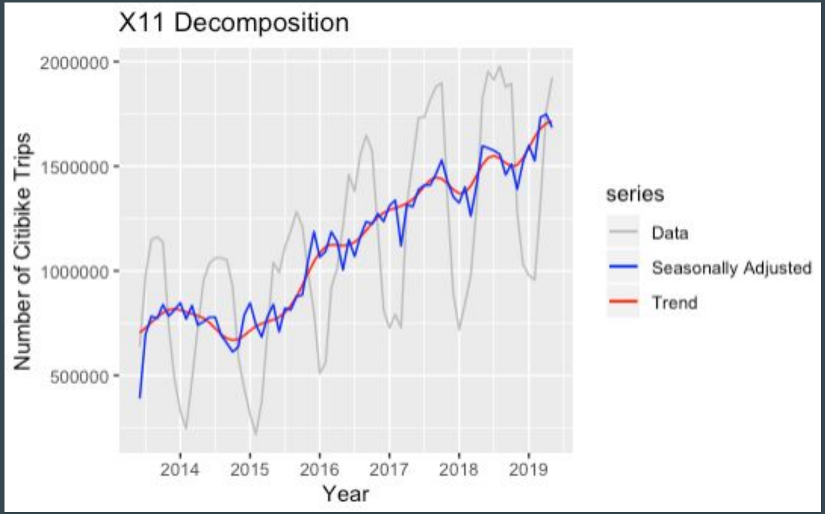
# Decomposition Models - X11 & SEATS

- X11 Decomposition
  - Based on classical decomposition, X11 decomposition can handle its shortcomings
  - Trend-cycle estimates are available for all observations, including first and last few observations
  - Where classical decomposition does not do well with outliers, X11 tends to be highly robust to outliers and sudden swings in the time series
  - Commonly used to create seasonally adjusted series

- SEATS Decomposition
  - Works only with monthly or quarterly data. For more granular analysis of seasonality, a different approach should be utilized

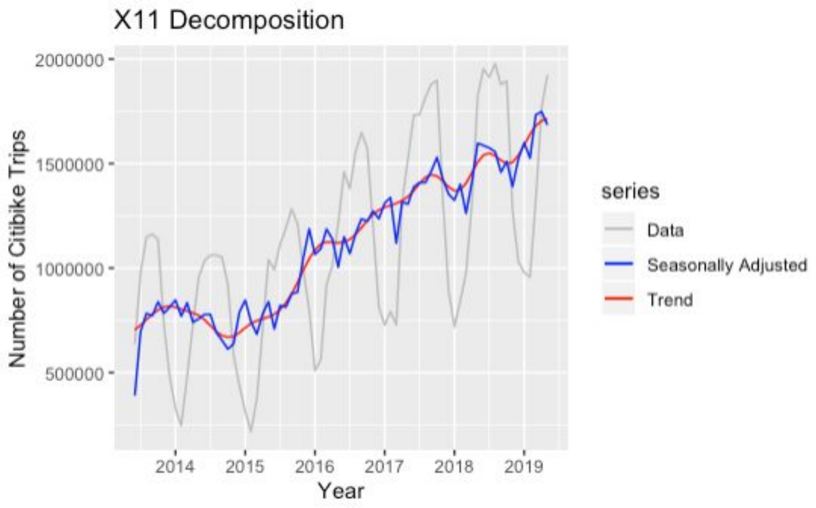
# Comparing Multiplicative against X11

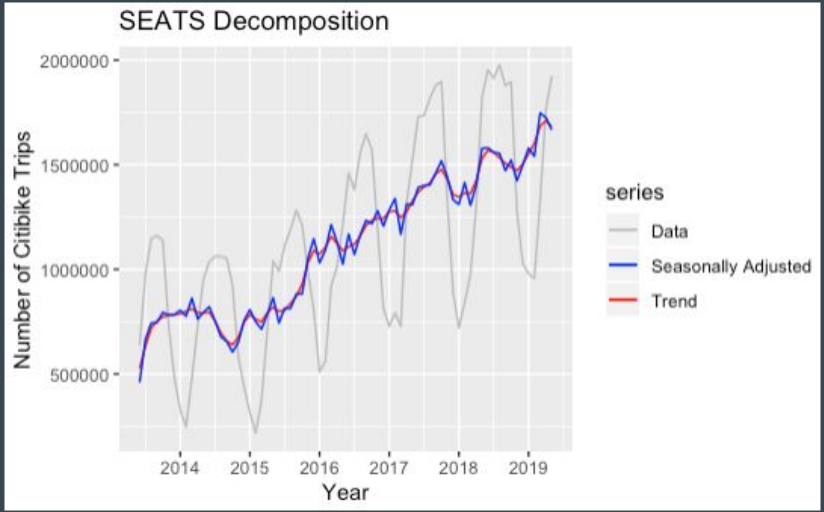



- Multiplicative appears to over-smooth the trend-cycle, obscuring that detail that is offered in X11..
- First and last few observations are available in X11
- X11 appears more robust in handling the sudden drops and rises from the winter and summer months.

# Comparing X11 against SEATS




- Appears very similar to each other
- Seasonal adjustment in SEATS appears smoother than the X11 decomposition.
- Trend cycle is slightly smoother in X11 than in SEATS.

# Decomposition Models - STL

STL (“Seasonal & Trend Decomposition using Loess”)

Advantages:

- Unlike SEATS and X11, STL will handle any type of seasonality, not only monthly and quarterly data
- The seasonal component is allowed to change over time, the rate of change can be controlled by the user
- The smoothness of the trend-cycle can be controlled by the user
- It can be robust to outliers (robust decomp) so unusual obs. Will not affect the estimates of trend-cycle and seasonal components. This will, however, affect the remainder

Disadvantage - Does not handle trading day or calendar variation automatically & only facilitates for additive decomp (Log transform necessary for multiplicative decomp)

# Exponential Smoothing Models - Simple Exponential Smoothing

This is a method that uses all of the prior data in the time series to predict the next observation. The most recent data gets a higher weighting. The forecast at time t depends on the forecast at t-1 and the error of that forecast. Does not capture trend or seasonality. ses function in the forecast package

$\hat{y}_t$ = Prediction at time t

$\hat{y}_{t+1}$ = Prediction at time t+1

$y_t$ = Actual at Time t

$\alpha$ = Smoothing Parameter, between 0 and 1, zero means past values have equal influence (under-smoothing), one means past values have no influence (over-smoothing), can use trial and error to determine optimal parameter

**$\hat{y}_{t+1} = \hat{y}_t + \alpha(y_t - \hat{y}_t)$ update by the error term**

$\hat{y}_{t+1} = \alpha(y_t) + (1-\alpha)(\hat{y}_t)$

$\hat{y}_{t+1} = \alpha(y_t) + (1-\alpha)(\alpha(y_{t-1}) + (1-\alpha)(\hat{y}_{t-1}))$

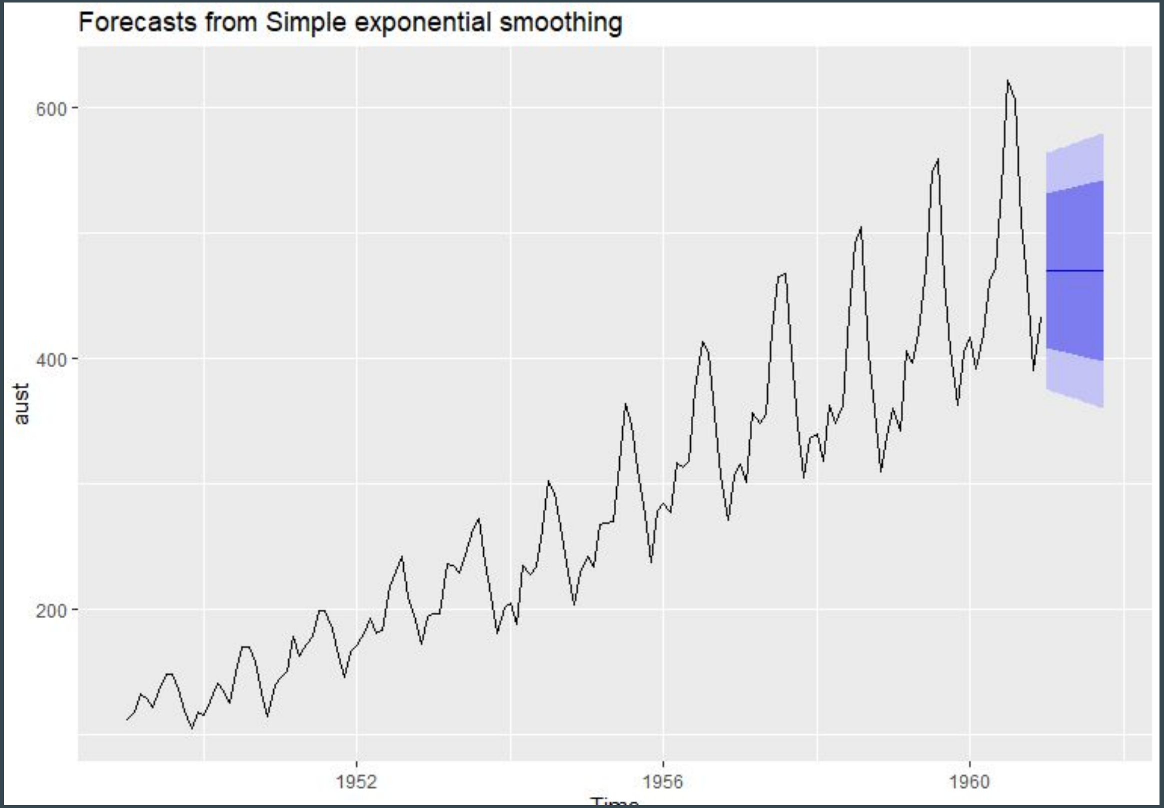
$\hat{y}_{t+1} = \alpha(y_t) + (1-\alpha)(\alpha(y_{t-1}) + (1-\alpha)(\hat{y}_{t-1}))$

This piece is why it is called exponential smoothing because this is like an exponential decay function with examples further in the past getting less and less weighting

$\hat{y}_{t+1} = \alpha(y_t) + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \alpha(1-\alpha)^3 y_{t-3}$..... Can also be written as $\hat{y}_{t+1} = \hat{y}_t + \alpha(y_t - \hat{y}_t)$ OR

$\hat{y}_{t+1} = L_t = L_{t-1} + \alpha(y_t - L_{t-1})$

# Simple Exponential Smoothing on Australia Airlines Passengers



# Exponential Smoothing - Holt's Exponential Smoothing

Also known as double exponential smoothing, because there are two smoothing parameters $\alpha$ and $\beta$, for the level and trend components respectively. This takes into account trends but not seasonality. In R we can use the ets function for this

The level is adjusted by the previous trend component. Essientially level would increase according to what the trend is

$T_t$ is the trend at time t

$L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + T_{t-1})$ adjust the previous level by adding trend

$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$ updated the trend by difference in the most recent level values

Using Holt's with Additive Trend:

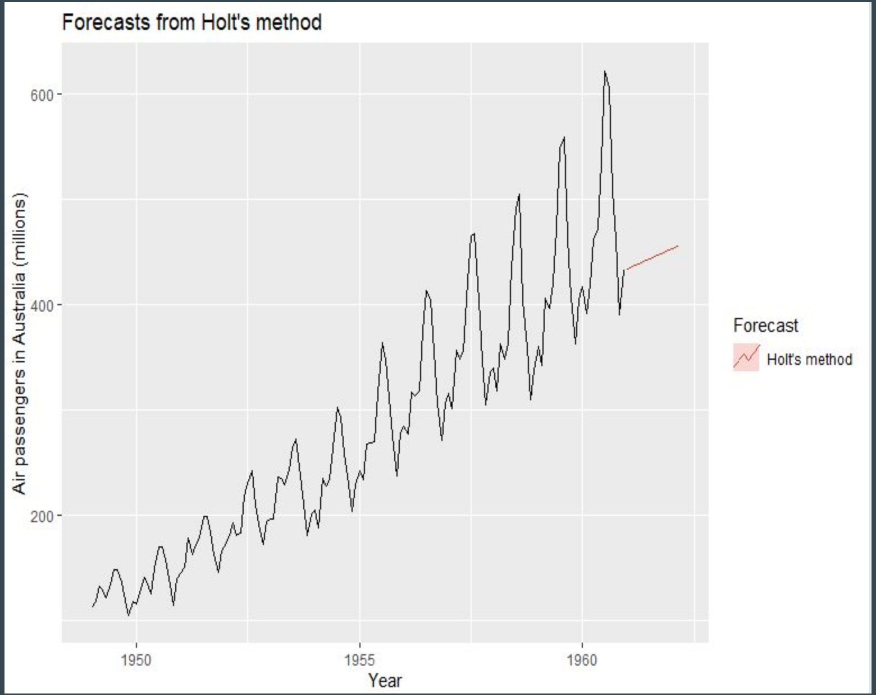
The new trend is weighted average of the old trend and the most previous change in levels

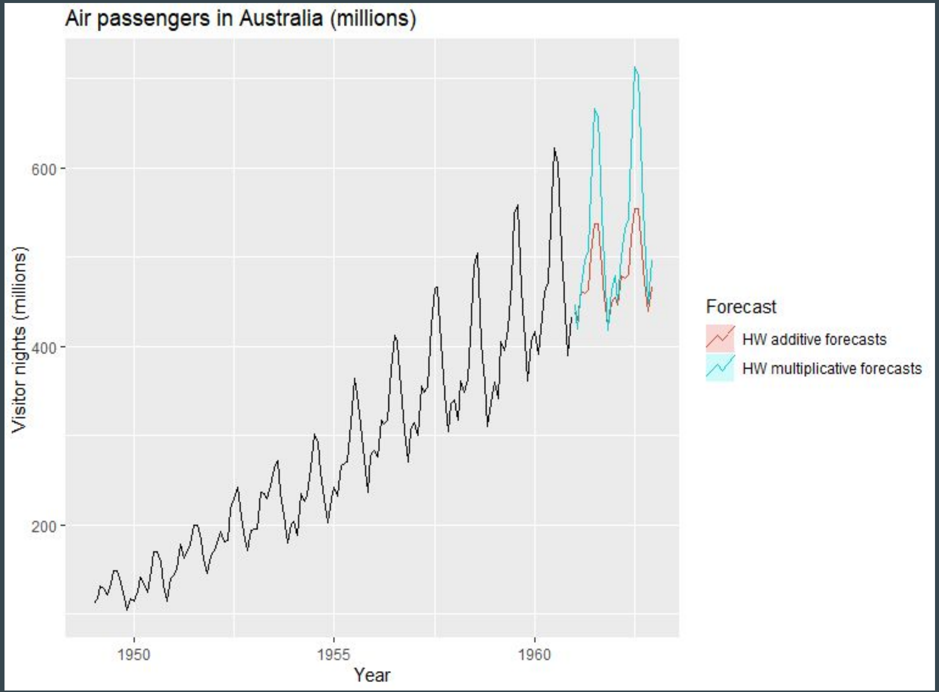
$Y_{t+k} = L_t + kT_t$

Using Holt's with Multiplicative Trend:

$Y_{t+k} = L_t * (T_t)^k$

# Comparing Holts and Holt Winters - forecast package holts and hw function




Picks up on the seasonal trend and the growing volatility of the series as a whole, this is the advantage of the multiplicative forecasts

# Exponential Smoothing Holt-Winter's Exponential Smoothing

Three smoothing parameters $\alpha$ and $\beta$ and $\gamma$, Forecast = estimated level + trend + seasonality at most recent time

All three of the smoothing parameters work the same way just on different parameters

These equations are for projecting the outcome in k periods, M is the number of seasons

Holt winters Additive Model = $Y_{t+k} = L_t + kT_t + S_{t+k-M}$

Holt winters Multiplicative Model = $Y_{t+k} = (L_t + kT_t)S_{t+k-M}$

The level is almost the exact same equation except now the seasonality component is also factored in, the level depends on the season

## Multiplicative equations

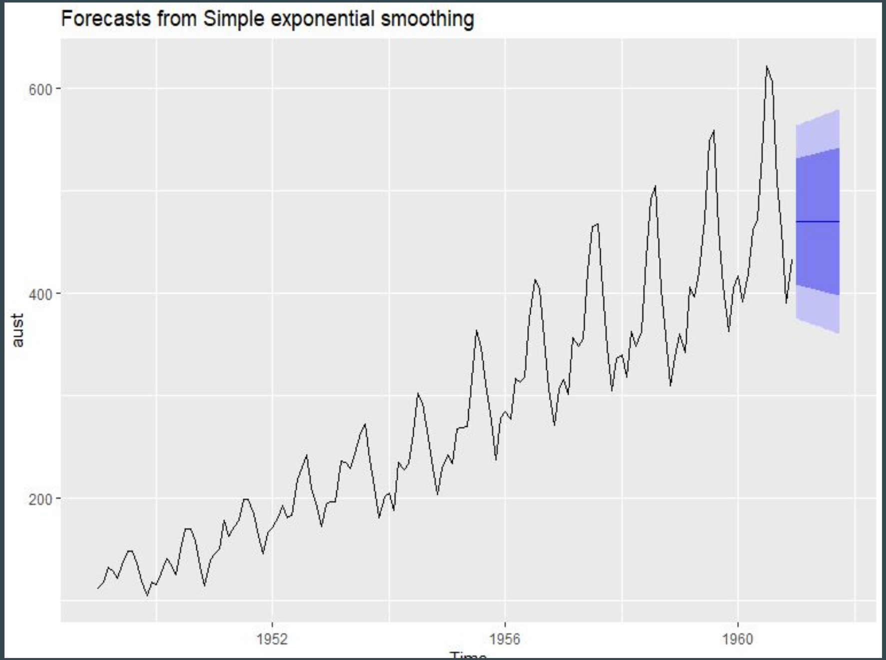
Level $L_t = \alpha\, (Y_t / S_{t-M}) + (1-\alpha)(L_{t-1} + T_{t-1})$ -- weighted average of previous season and most recent level and trend

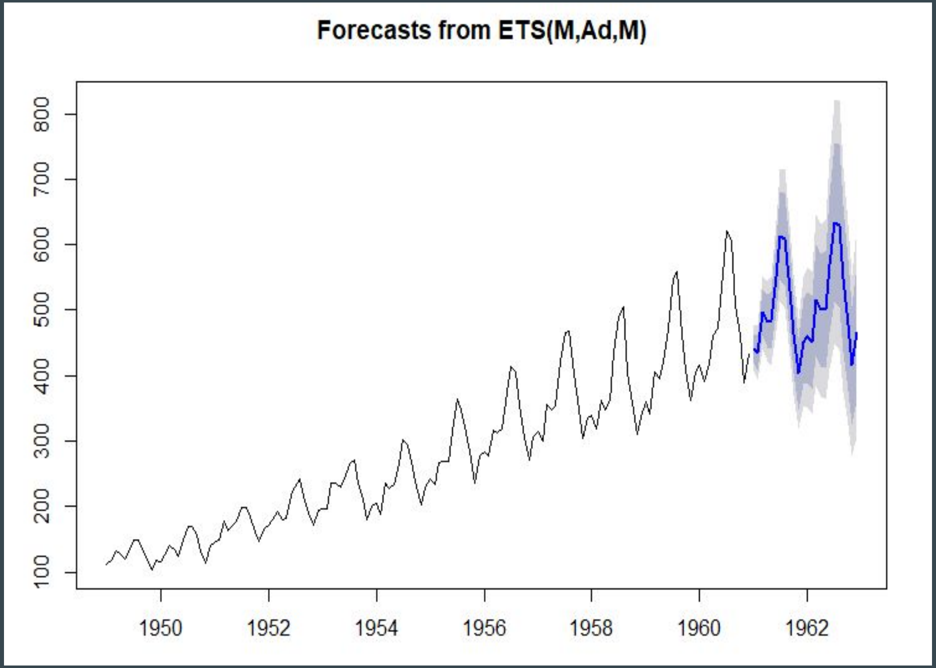
Trend $T_t = \beta\, (L_t - L_{t-1}) + (1 - \beta)T_{t-1}$ -- weighted average of most recent level change and the most recent trend

The most recent season prior to the current one

Seasonality: $S_t = \gamma\, (Y_t/(L_t+T_t)) + (1-\gamma)S_{t-M}$ - weighted average of most recent season and also the most recent difference between between the prediction and the level and trend - this helps adjust the growing volatility in the seasons in the previous example, how much different was the observation than expected

# Holt Winters with Confidence Interval ets function




Shows the difference between the simplest model and the most complex model and their forecasting abilities. Exponential smoothing is able to be broken up into three components: level, trend, and seasonality

# Time Series Metrics

$e_t = (y_t - \hat{y}_t)$ *(at error at time t),* $\hat{y}_t$ = Prediction at time t, $y_t$ = Actual at Time t

Mean Error = 1 / n * Σ $e_t$

Mean Percentage Error = 100 * 1 / n * Σ $e_t / y_t$

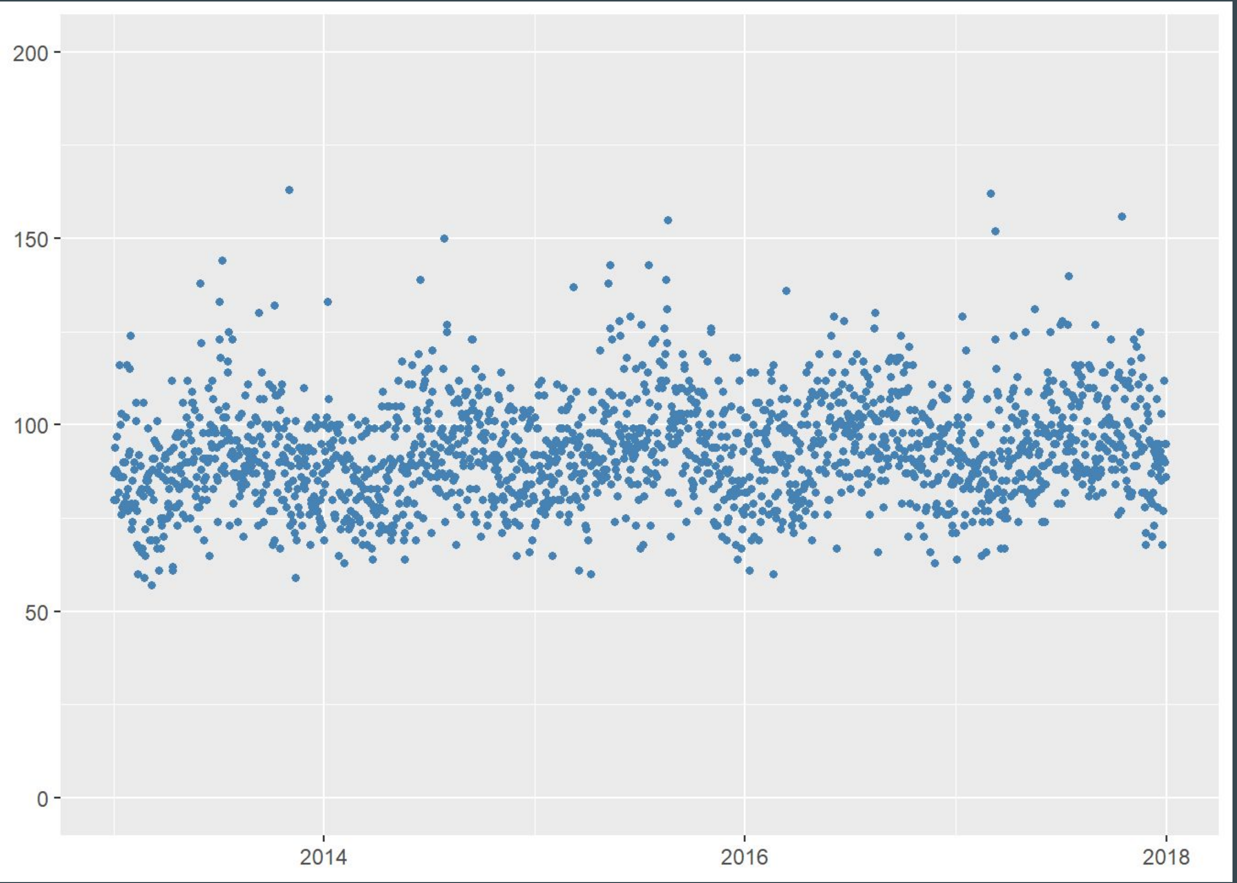
Mean Squared Error = 1 / n * Σ $e^2_t$

Mean Absolute Error = 1 / n * Σ | $e_t$ |

Mean Absolute Percentage Error = 100 * 1 / n * Σ | $e_t / y_t$ |

# Case Study: Facebook Prophet Model of Rochester Fire Department Fire Incidents

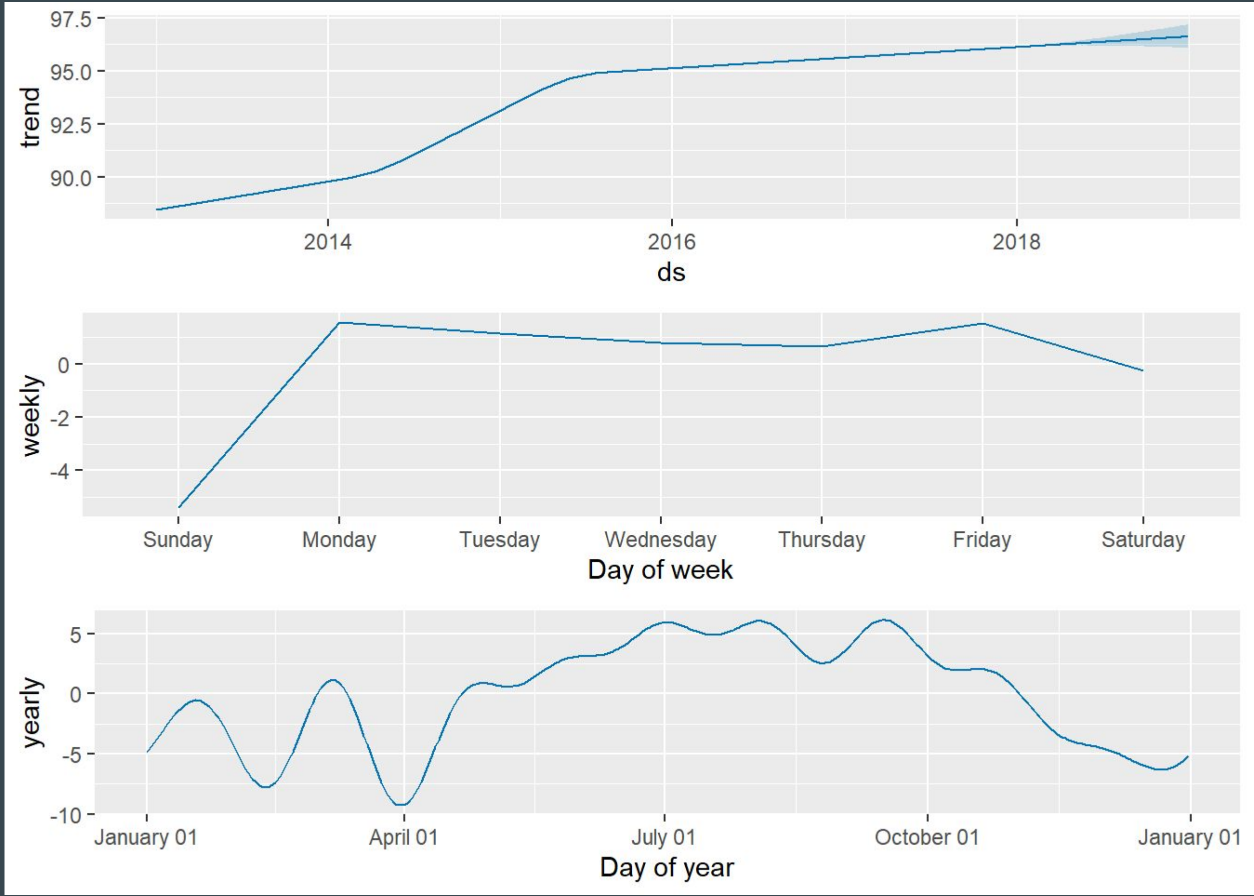
# Rochester New York Fire Department Fire Incidents



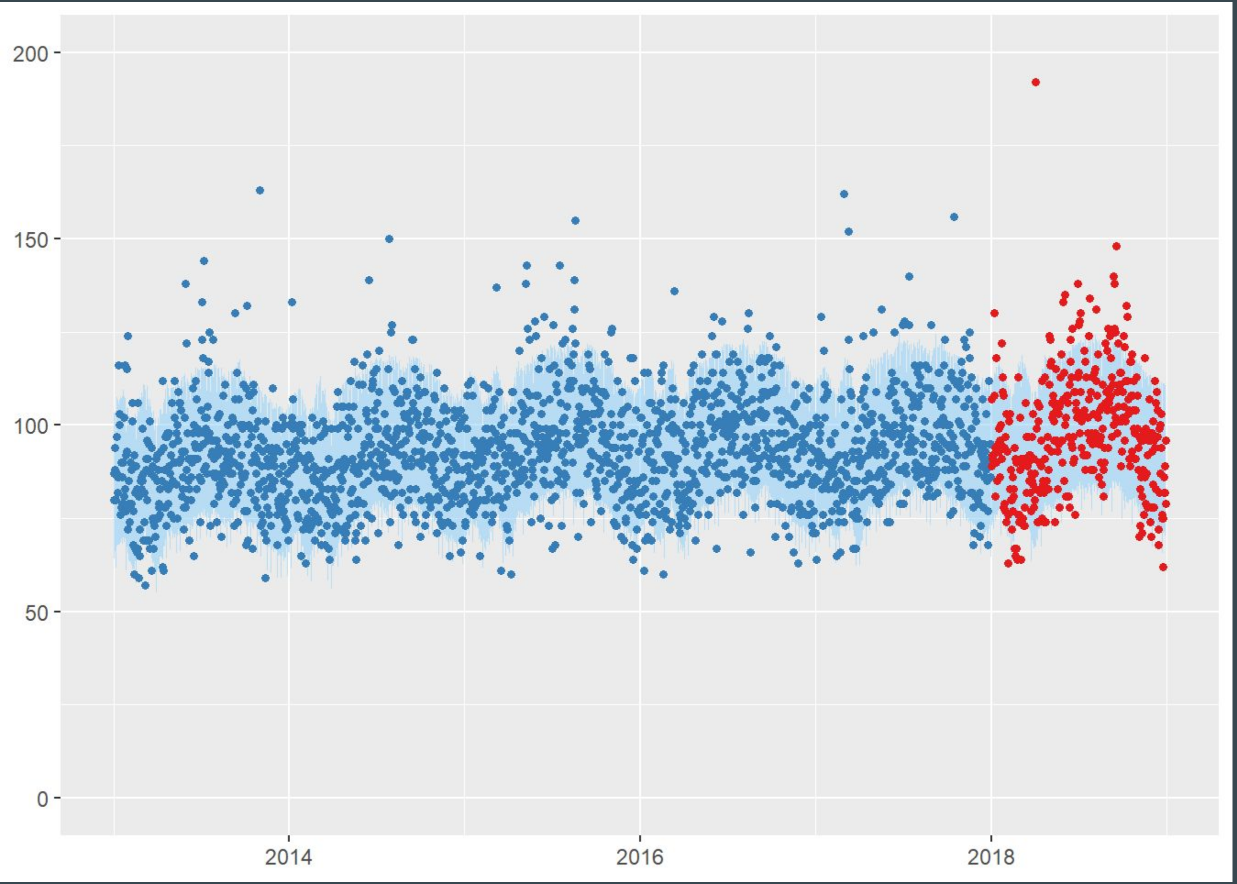

2013 through 2018

Min: 57
1st Qu.: 83.25
Median: 92
Mean: 93.21
3rd Qu.: 101
Max: 433 (not shown)

# Rochester New York Fire Department Fire Incidents



# Rochester New York Fire Department Fire Incidents




Predictions Accuracy

In Bands: 304 times
Outside Bands: 61 times

# Questions?

# Resources and Further Readings

# Resources & Further Readings

- General Time Series Decomposition
  - https://machinelearningmastery.com/decompose-time-series-data-trend-seasonality/
  - https://medium.com/better-programming/a-visual-guide-to-time-series-decomposition-analysis-a1472bb9c930
- Facebook Prophet
  - Homepage - https://facebook.github.io/prophet/
  - https://cran.r-project.org/web/packages/prophet/vignettes/quick_start.html
  - Example R Code: https://github.com/mikeasilva/CUNY-SPS/blob/master/DATA624/Prophet%20Example.R
- X11
  - https://www.abs.gov.au/websitedbs/d3310114.nsf/4a256353001af3ed4b2562bb00121564/c890aa8e65957397ca256ce10018c9d8!OpenDocument
  - https://support.sas.com/documentation/onlinedoc/ets/132/x11.pdf
  - Data source: https://www.citibikenyc.com/system-data
- Exponential Smoothing
  - https://www.youtube.com/user/ProfGalitShmueli/videos