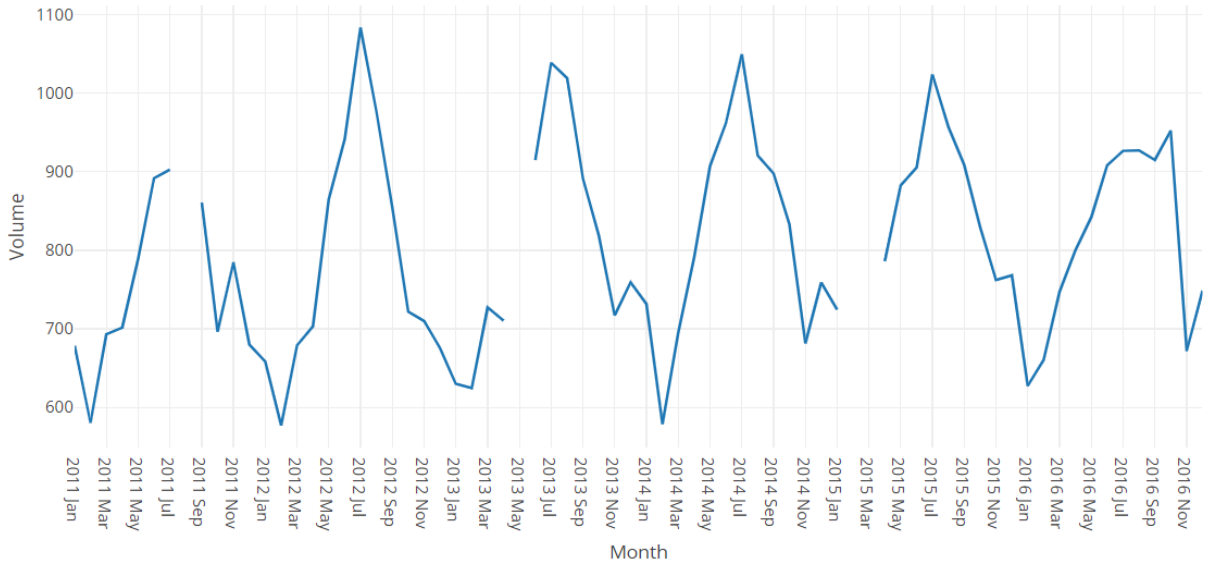
Missing Values

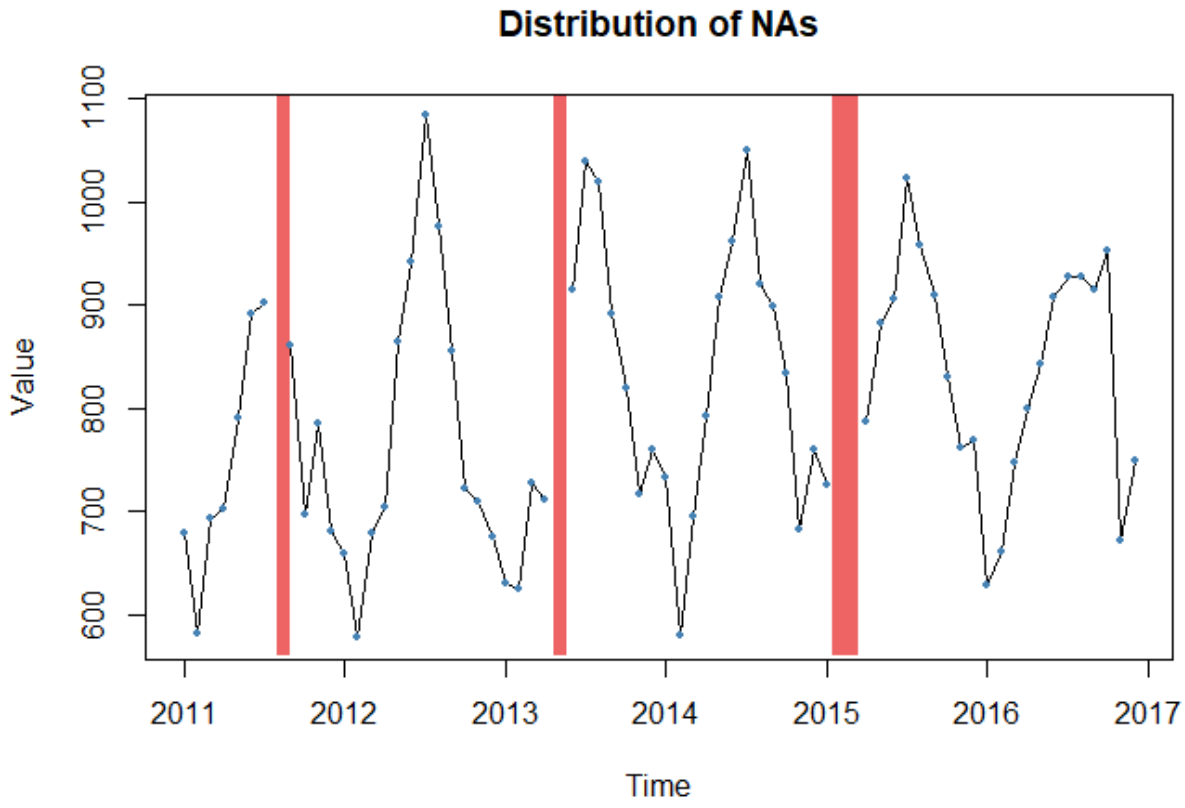


How do we handle them?

- Excluding rows that have missing values is not an option in time series. There are multiple ways to impute missing values:
    - Imputation by Interpolation (linear/spline)
    - Imputation by Last Observation Carried Forward
    - Missing Value Imputation by Weighted Moving Average
    - Imputation by Mean Value
    - Imputation by Random Sample
- The above techniques can be used in conjunction with the below:
    - Seasonally Decomposed Missing Value Imputation
    - Seasonally Splitted Missing Value Imputation
- Let's characterize the missing values in the above beer volume series:

```
# Using a function from imputeTS package
statsNA(volume_ts, bins = 6, printOnly = TRUE)
## [1] "Length of time series:"
## [1] 72
## [1] "-------------------------"
## [1] "Number of Missing Values:"
## [1] 4
## [1] "-------------------------"
## [1] "Percentage of Missing Values:"
## [1] "5.56%"
## [1] "-------------------------"
## [1] "Stats for Bins"
## [1] "  Bin 1 (12 values from 1 to 12) :      1 NAs (8.33%)"
## [1] "  Bin 2 (12 values from 13 to 24) :      0 NAs (0%)"
```

```
## [1] "  Bin 3 (12 values from 25 to 36) :      1 NAs (8.33%)"
## [1] "  Bin 4 (12 values from 37 to 48) :      0 NAs (0%)"
## [1] "  Bin 5 (12 values from 49 to 60) :      2 NAs (16.7%)"
## [1] "  Bin 6 (12 values from 61 to 72) :      0 NAs (0%)"
## [1] "-------------------------"
## [1] "Longest NA gap (series of consecutive NAs)"
## [1] "2 in a row"
## [1] "-------------------------"
## [1] "Most frequent gap size (series of consecutive NA series)"
## [1] "1 NA in a row (occuring 2 times)"
## [1] "-------------------------"
## [1] "Gap size accounting for most NAs"
## [1] "2 NA in a row (occuring 1 times, making up for overall 2 NAs)"
## [1] "-------------------------"
## [1] "Overview NA series"
## [1] "  1 NA in a row: 2 times"
## [1] "  2 NA in a row: 1 times"
plotNA.distribution(volume_ts)
```

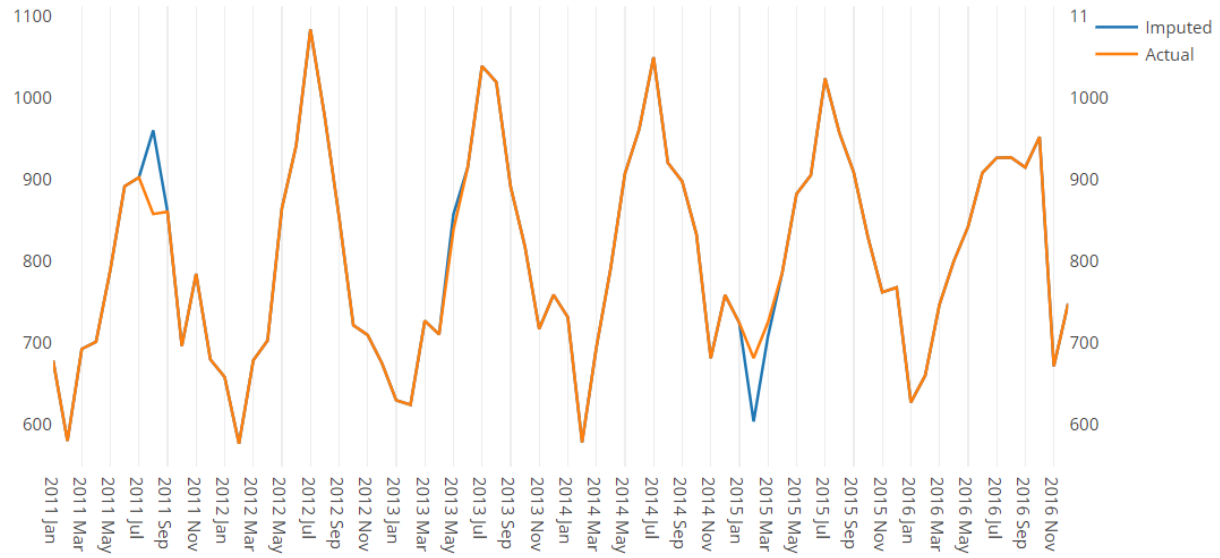


Imputation

- From a shipments point of view, the volume in a month of February should be fairly identical to the behavior of previous Februaries
- Let's try seasonally splitted imputation by mean value to replicate the above logic :

```
# Using a function from imputeTS package
data$Volume_imp <- na.seasplit(volume_ts, algorithm = "mean")

# Plotting imputed vs. actual to check if approximations are fit for use
plot_ly(data, x = ~Month, y = ~Volume_imp, type = 'scatter', mode = 'lines',
yaxis='y', name='Imputed') %>%
  add_trace(data_a, x = ~Month, y= ~volume_ts_a, type = 'scatter', mode='lines',
name='Actual', yaxis='y2') %>% layout(autosize=F, width=900, height = 450, margin =
list(b = 100),
         xaxis = list(title = ""),
         yaxis = list(side = 'left', title = '', showgrid = FALSE, zeroline = FALSE),
         yaxis2 = list(side = 'right', overlaying = "y", title = '', showgrid =
FALSE, zeroline = FALSE))
```

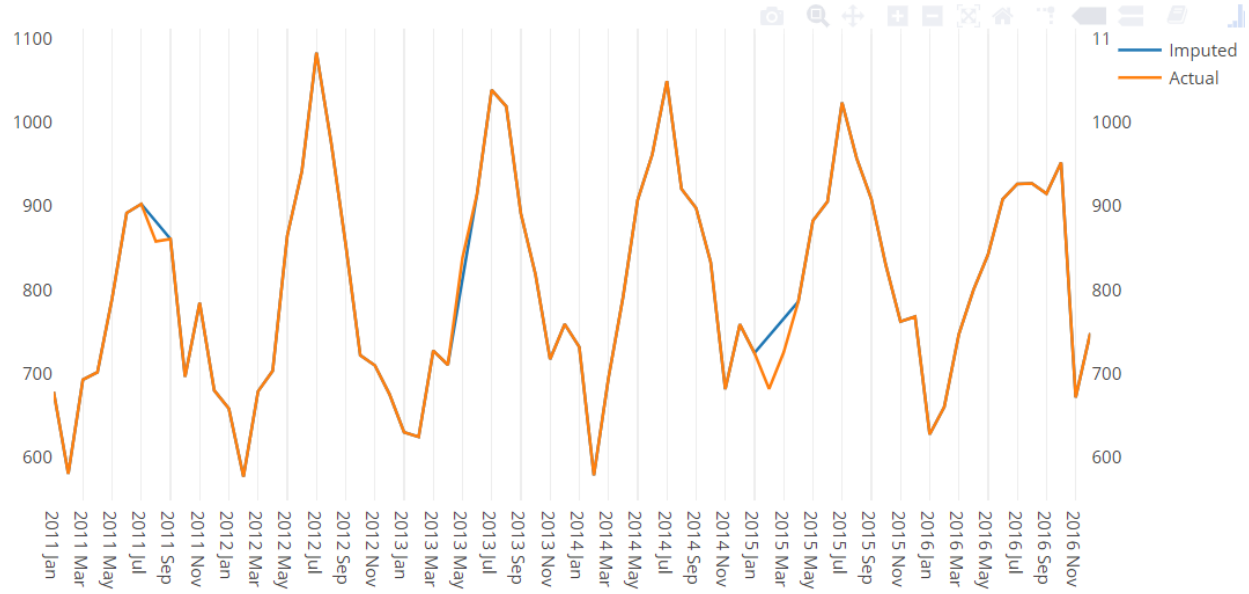


- Observe that the algorithm imputed the average value of that month across seasons. Coming very close in just 2013, the algorithm is not able to impute well for other years due to high variation
- This tells us that the seasonally split mean imputation technique is suited for patterns which have less variance during the same period across seasons
- We will now attempt to use linear interpolation to impute the values :

```
# Using a function from imputeTS package
data$Volume_imp <- na.interpolation(volume_ts, option = "linear")

# Plotting imputed vs. actual to check if approximations are fit for use
plot_ly(data, x = ~Month, y = ~Volume_imp, type = 'scatter', mode = 'lines',
yaxis='y', name='Imputed') %>%
  add_trace(data_a, x = ~Month, y= ~volume_ts_a, type = 'scatter', mode='lines',
name='Actual', yaxis='y2') %>% layout(autosize=F, width=900, height = 450, margin =
list(b = 100),
         xaxis = list(title = ""),
         yaxis = list(side = 'left', title = '', showgrid = FALSE, zeroline = FALSE),
```

```
        yaxis2 = list(side = 'right', overlaying = "y", title = '', showgrid =
FALSE, zeroline = FALSE))
```



- Although a little off in imputing the 2015 February and March values, the linear interpolation technique comes pretty close in 2011 and 2013