Capestone Project

James Strayer

02/01/2020

Introduction

I was asked by a manager of Restaurant that does retail, to have a model to predict his sales for a given week, he gave me a file with the sales since end of March 2015. He started as the manager on 01-10-2018

Data import and cleaning

```
## Loading required package: tidyverse

## -- Attaching packages --------------------------------------------------- tidyverse 1.2.1 --

## <U+2713> ggplot2 3.2.1     <U+2713> purrr   0.3.2
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.3
## <U+2713> tidyr   0.8.3     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0

## -- Conflicts ------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Weekday = col_character(),
##   Year = col_double(),
##   Month = col_character(),
##   Day = col_double(),
##   `annee fiscale` = col_character(),
##   Sales = col_number(),
##   Retail = col_number(),
##   TakeOutSales = col_number(),
##   Bar_Sales = col_number(),
##   Sales_Restaurant = col_number()
## )

## Observations: 1,715
## Variables: 11
## $ Date             <date> 2015-03-29, 2015-03-30, 2015-03-31, 2015-04-01, 201...
## $ Weekday          <chr> "SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDA...
## $ Year             <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015...
## $ Month            <chr> "March'15", "March'15", "March'15", "April'15", "Apr...
## $ Day              <dbl> 29, 30, 31, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1...
## $ `annee fiscale`  <chr> "2014-15", "2014-15", "2014-15", "2014-15", "2014-15...
## $ Sales            <dbl> 1792.00, NA, 1526.30, 2250.26, 2077.57, 2357.48, 150...
## $ Retail           <dbl> 373.00, NA, 380.95, 363.72, 268.80, 394.23, 476.17, ...
```

```
## $ TakeOutSales    <dbl> 0.00, 0.00, 99.25, 402.75, 0.00, 32.50, 93.73, 0.00,...
## $ Bar_Sales       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 227....
## $ Sales_Restaurant <dbl> 1419.00, 0.00, 1046.10, 1483.79, 1808.77, 1930.75, 9...
```

The manager is using the data to have a day to day idea of the health of his buisness and therefore he added an accounting variable, annee fiscale (fiscal year) ,that we don't need and that we can remove.The Month variable is in a format not pratical for analysis (Month, YY) we want to have just the Full moth written without the year. So for simplicity and because we have the full date in the Date column, we are going to remove the month column as it is and create a new one based on the date column, we will do the same for the weekdays column. Furthermore we don't need the day number in our analysis so we will remove it too:

```r
#Remove unncessary column for the analysis
DataForAnalysis<-DailySales%>%select(-Weekday,-Month,-Day,-`annee fiscale`)
#Add a column for weekday and month
DataForAnalysis<-DataForAnalysis%>%mutate(Weekday=weekdays(Date),Month=months(Date))
```

If we look again at the data we see that we have a Sales column, representing the Total Sales for the day and then each column after it, is the total sales for each day for each component of the restaurant possible sales revenu, so Retail, Take-out, Bar and Restaurant. We can see that there is NAs in all of those data

```r
#looking for NAs in the sales data
sum(is.na(DataForAnalysis$Sales))
```

```
## [1] 175
```

```r
sum(is.na(DataForAnalysis$Bar_Sales))
```

```
## [1] 208
```

```r
sum(is.na(DataForAnalysis$Retail))
```

```
## [1] 188
```

```r
sum(is.na(DataForAnalysis$TakeOutSales))
```

```
## [1] 1
```

```r
sum(is.na(DataForAnalysis$Sales_Restaurant))
```

```
## [1] 1
```

we will change those NAs to 0, considering a $0 CAD sales for that day and variable

```r
#Chaning NAs to 0 in the sales data
DataForAnalysis[is.na(DataForAnalysis$Sales),]$Sales<-0
DataForAnalysis[is.na(DataForAnalysis$Bar_Sales),]$Bar_Sales<-0
DataForAnalysis[is.na(DataForAnalysis$Retail),]$Retail<-0
DataForAnalysis[is.na(DataForAnalysis$TakeOutSales),]$TakeOutSales<-0
DataForAnalysis[is.na(DataForAnalysis$Sales_Restaurant),]$Sales_Restaurant<-0
```

Once we Cleaned the data. We are interested to add some classification for the days, specialy considering that holidays and special event should have an impact on the sales of a restaurant, to test this hypothesis we create a data frame event for all the bank holidays and events in the province of Quebec during the year:

here is an example of the data used to create that data frame: https://www.statutoryholidays.com/2017.php, all dates with observance National, QC and event such as Mother's day and Valentine's day. Once this vector is created we can create a new variable called EventDay which is true if the date equals one of the date in the vector

```r
#Add a column for EventDay
DataForAnalysis<-DataForAnalysis%>%mutate(EventDay=ifelse(Date %in% Event,TRUE,FALSE))
```

Data exploration

```r
#look at the structure of the data
glimpse(DataForAnalysis)
```

```
## Observations: 1,715
## Variables: 10
## $ Date            <date> 2015-03-29, 2015-03-30, 2015-03-31, 2015-04-01, 201...
## $ Year            <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015...
## $ Sales           <dbl> 1792.00, 0.00, 1526.30, 2250.26, 2077.57, 2357.48, 1...
## $ Retail          <dbl> 373.00, 0.00, 380.95, 363.72, 268.80, 394.23, 476.17...
## $ TakeOutSales    <dbl> 0.00, 0.00, 99.25, 402.75, 0.00, 32.50, 93.73, 0.00,...
## $ Bar_Sales       <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00...
## $ Sales_Restaurant <dbl> 1419.00, 0.00, 1046.10, 1483.79, 1808.77, 1930.75, 9...
## $ Weekday         <chr> "Sunday", "Monday", "Tuesday", "Wednesday", "Thursda...
## $ Month           <chr> "March", "March", "March", "April", "April", "April"...
## $ EventDay        <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...
```

We want to have full years to have the same thing amount of days in the year to predict 2019 weeks. So we filter for 2016 and more

```r
#Summary statistics for the variable
summary(DataForAnalysis)
```

```
##       Date                  Year          Sales          Retail
##  Min.   :2016-01-01   Min.   :2016   Min.   :   0.0   Min.   :   0.0
##  1st Qu.:2016-12-25   1st Qu.:2016   1st Qu.: 756.8   1st Qu.: 160.2
##  Median :2017-12-19   Median :2017   Median :1212.2   Median : 257.7
##  Mean   :2017-12-19   Mean   :2017   Mean   :1229.4   Mean   : 266.5
##  3rd Qu.:2018-12-13   3rd Qu.:2018   3rd Qu.:1703.3   3rd Qu.: 367.9
##  Max.   :2019-12-29   Max.   :2019   Max.   :4360.8   Max.   :1436.9
##   TakeOutSales        Bar_Sales      Sales_Restaurant    Weekday
##  Min.   :   0.00   Min.   :   0.0   Min.   :-2299.8   Length:1437
##  1st Qu.:   0.00   1st Qu.:  70.0   1st Qu.:  358.0   Class :character
##  Median :  64.00   Median : 153.8   Median :  665.0   Mode  :character
##  Mean   :  83.85   Mean   : 177.0   Mean   :  702.0
##  3rd Qu.: 120.00   3rd Qu.: 257.5   3rd Qu.:  985.2
##  Max.   :2965.72   Max.   :1045.0   Max.   : 4284.2
##     Month             EventDay
##  Length:1437        Mode :logical
##  Class :character   FALSE:1437
##  Mode  :character
##
##
##
```
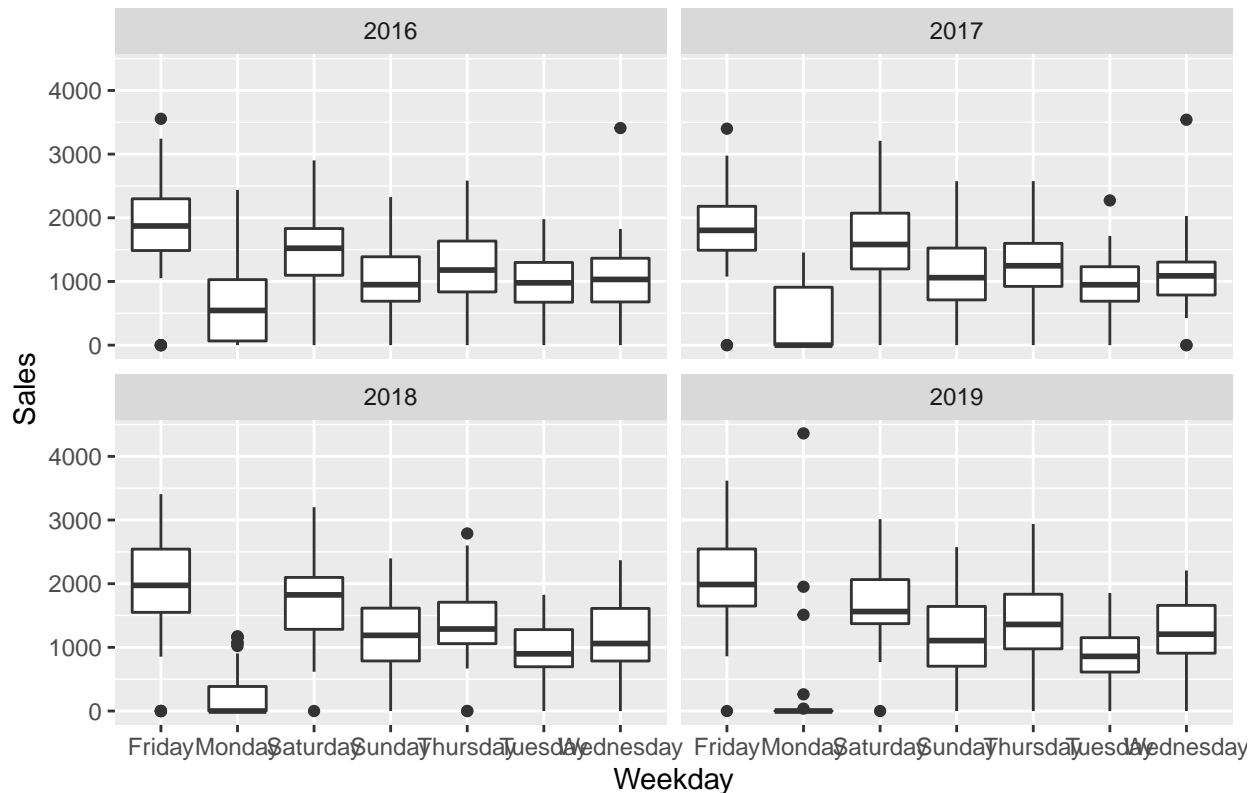
In our data set we have 10 variables for 1715 observations corresponding to the number of days the restaurant was open since the 29-03-2015.

Sales Variable

Sales distribution by Year
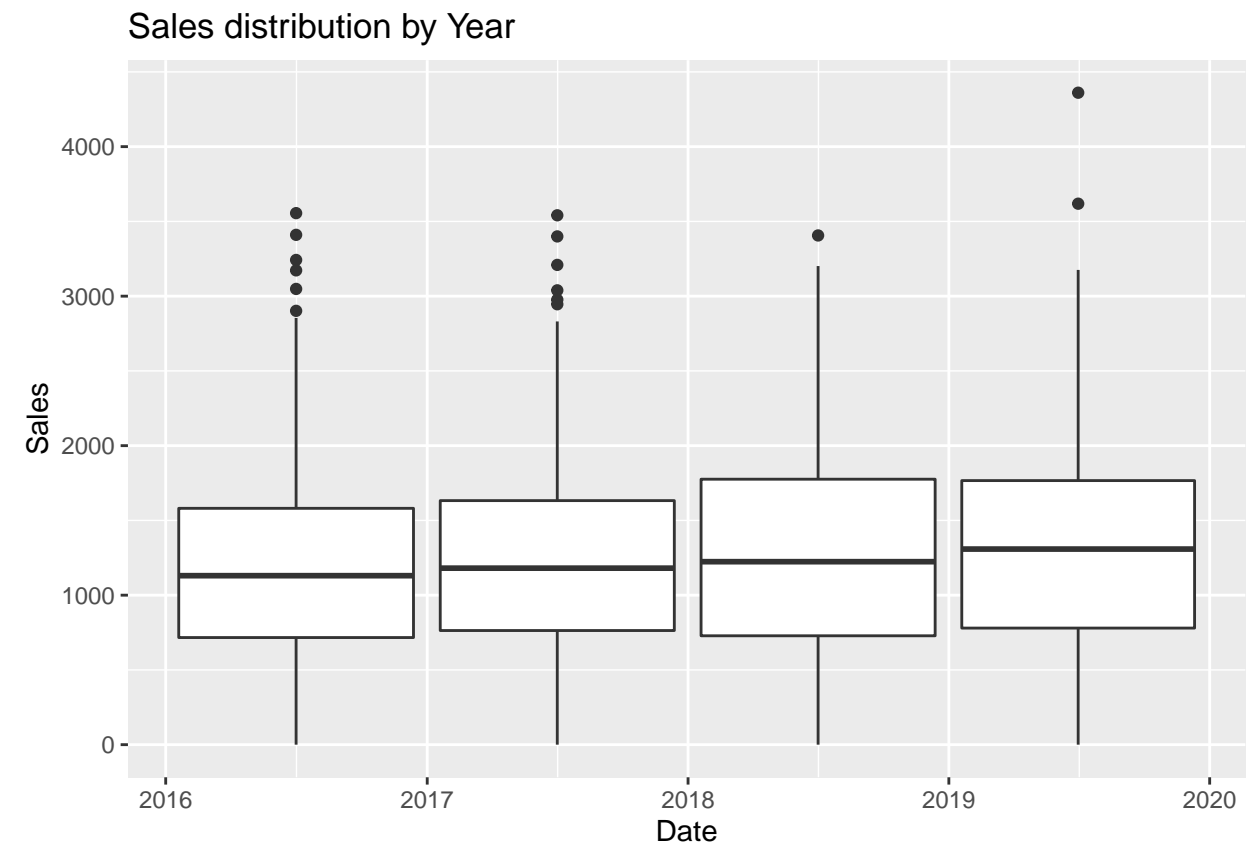
We see In the box plot above that 2019 was similar to 2018 in term of sales but with 2 outliers

Sales distribution by Day of the Week for each year

In the box plot above we see that in 2019 they were very few Sales on Mondays and that the biggest are through the years were Thursday, Friday and Saturday. ##Preparing Data for Modeling I will be considering this data as a time series and such I will use a library made to handle them

```
## Loading required package: timetk
```

```
## Warning: package 'timetk' was built under R version 3.6.2
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Loading required package: tibbletime
```

```
## Warning: package 'tibbletime' was built under R version 3.6.2
```

```
##
## Attaching package: 'tibbletime'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

We then transform the data to train and test data and transform the train data into a time serie

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.6.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Modeling Data

ARIMA Model

We are going to test to 2 models to predict sales at a given week knowing the previous weeks. We will wan to try to predict the first week of 2019. We going to compare Linear Regression and ARIMA (Auto Regressive Integrated Moving Average) model (you can have a brief summary of ARIMA model here:https://machinelearningmastery.com/gentle-introduction-box-jenkins-method-time-series-forecasting/)

We use the forecast packages to use it:

```
## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.6.2

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method            from
##   fitted.fracdiff   fracdiff
##   residuals.fracdiff fracdiff

## Loading required package: sweep

## Warning: package 'sweep' was built under R version 3.6.2

## Series: ts_train
## ARIMA(5,0,1)(0,0,1)[52] with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5      ma1     sma1       mean
##       1.1903  -0.4182  0.0356  -0.0956  0.2718  -0.8345  -0.0506  1129.7874
## s.e.  0.0360   0.0476  0.0488   0.0467  0.0325   0.0262   0.0359   158.8204
##
## sigma^2 estimated as 322605:  log likelihood=-8503.07
## AIC=17024.14   AICc=17024.3   BIC=17069.13
##
## Training set error measures:
##                    ME     RMSE      MAE  MPE MAPE      MASE        ACF1
## Training set 6.959659 565.9062 439.6488 -Inf  Inf 0.4948046 -0.04969305

## # A tibble: 1,096 x 4
##    index .actual .fitted .resid
##    <dbl>   <dbl>   <dbl>  <dbl>
## 1 2016        0    863.  -863.
## 2 2016.       0    494.  -494.
## 3 2016.       0    532.  -532.
## 4 2016.       0    534.  -534.
## 5 2016.     478.   561.  -82.7
## 6 2016.     683.   652.   31.8
## 7 2016.     920.   621.  299.
## 8 2016.    1496.   630.  866.
```

```
##  9 2016.    1365.    702.  662.
## 10 2016.     822.    574.  247.
## # ... with 1,086 more rows
```

We see that as we go forward in time the residual get corrected by the mean of the previous days we can visualize the evolution of the residuals:
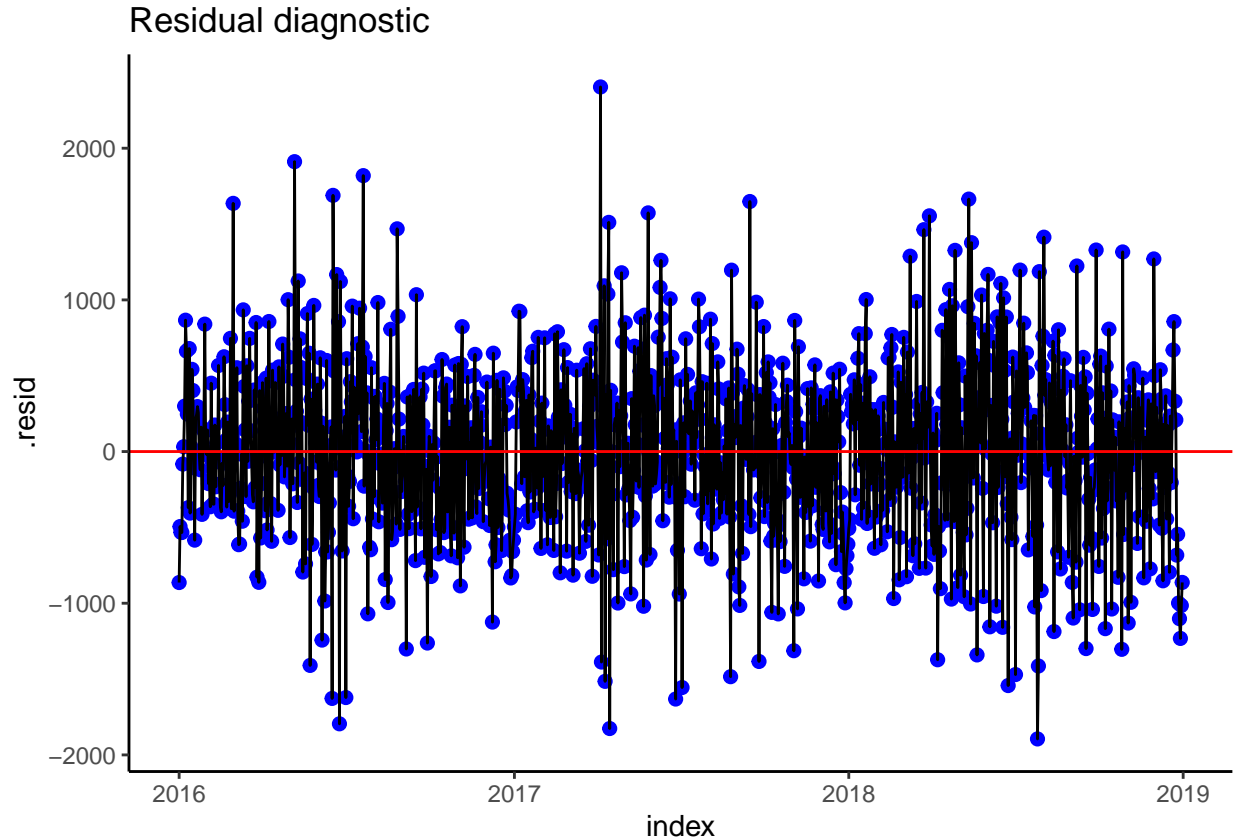
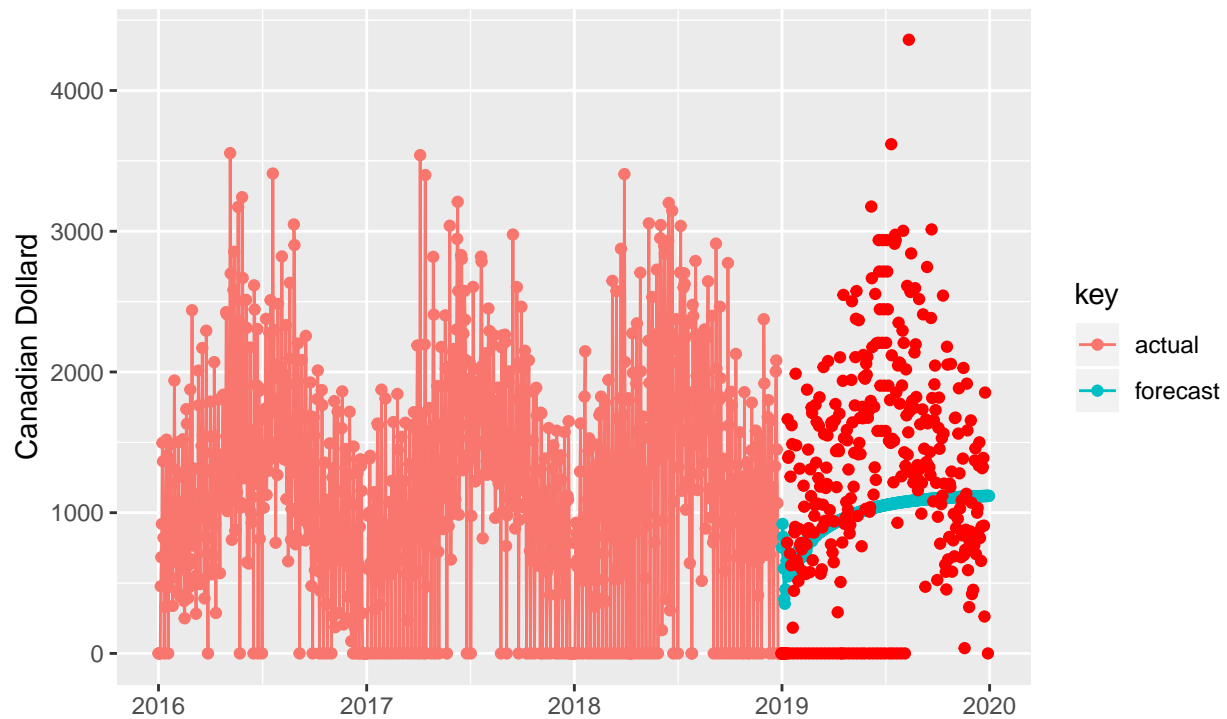We then forecast the first week of 2019

```
#try to forecast the next 365 days (2019)
predict_arima<-forecast(fit_arima,h=365)
#Get the forecast value in a table
fcast_tbl <- sw_sweep(predict_arima, timetk_idx = TRUE)
fcast_tbl
```

```
## # A tibble: 1,461 x 7
##    index      key    value lo.80 lo.95 hi.80 hi.95
##    <date>     <chr>  <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2016-01-01 actual     0    NA    NA    NA    NA
##  2 2016-01-02 actual     0    NA    NA    NA    NA
##  3 2016-01-03 actual     0    NA    NA    NA    NA
##  4 2016-01-04 actual     0    NA    NA    NA    NA
##  5 2016-01-05 actual   478.   NA    NA    NA    NA
##  6 2016-01-06 actual   683.   NA    NA    NA    NA
##  7 2016-01-07 actual   920.   NA    NA    NA    NA
##  8 2016-01-08 actual  1496.   NA    NA    NA    NA
##  9 2016-01-09 actual  1365.   NA    NA    NA    NA
## 10 2016-01-10 actual   822.   NA    NA    NA    NA
```

```
## # ... with 1,451 more rows
actual_tbl <- DataForAnalysis %>% filter(Date >= '2018-12-31')
#Visualize with actual value
fcast_tbl %>%
    ggplot(aes(x = index, y = value, color = key)) +
    geom_line() +
    geom_point() +
#Actual data
    geom_point(data = actual_tbl, aes(x = Date, y = Sales), color = 'red') +
    labs(title = "Sales Forecast: ARIMA", x = "", y = "Canadian Dollard",
            subtitle = "sw_sweep tidies the auto.arima() forecast output") +
    scale_x_date(date_breaks = "1 year", date_labels = "%Y")
```

Sales Forecast: ARIMA

sw_sweep tidies the auto.arima() forecast output

Discussion