Anomaly Detection in R

Karimi Gichunge

11/14/2020

Research Question Identify any anomalies using the sales dataset given

1. Reading data and loading libraries

```
#Loading dataset
Supermarket_Sales_Forecasting...Sales <- read.csv("C:/Users/Karimi/Downloads/Supermarket_Sales_Forecasting - Sales.csv")
data <- Supermarket_Sales_Forecasting...Sales

#Previewing head
head(data)
```

```
##        Date    Sales
## 1  1/5/2019 548.9715
## 2  3/8/2019  80.2200
## 3  3/3/2019 340.5255
## 4 1/27/2019 489.0480
## 5  2/8/2019 634.3785
## 6 3/25/2019 627.6165
```

```
#Installing packages
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------ tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.2     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

```
library(anomalize)
```

```
## == Use anomalize to improve your Forecasts by 50%! ==================================================
## Business Science offers a 1-hour course - Lab #18: Time Series Anomaly Detection!
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

2. Tidying dataset

```
#Checking for missing values
colSums(is.na(data))
```

```
##  Date Sales
##     0     0
```

```
#Changing table to tibble
data$Date <- as.Date(data$Date, format = "%m/%d/%Y")
df <- as.tibble(data)
```

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
is_tibble(df)
```

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

```
#aggregating sales values to get daily records
df.anomaly <- aggregate(df["Sales"], by=df["Date"],sum)
head(df.anomaly)
```

```
##         Date    Sales
## 1 2019-01-01 4745.181
## 2 2019-01-02 1945.503
## 3 2019-01-03 2078.128
## 4 2019-01-04 1623.688
## 5 2019-01-05 3536.684
## 6 2019-01-06 3614.205
```

```
df.anomaly <- as.tibble(df.anomaly)
is_tibble(df.anomaly)
```

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

3.Anomaly detection

```
anomaly.detect <- df.anomaly %>%
  time_decompose(Sales, method = "stl", frequency = "auto", trend = "auto") %>%
  anomalize(remainder, method = "gesd", alpha = 0.05, max_anoms = 0.2) %>%
  plot_anomaly_decomposition()
```

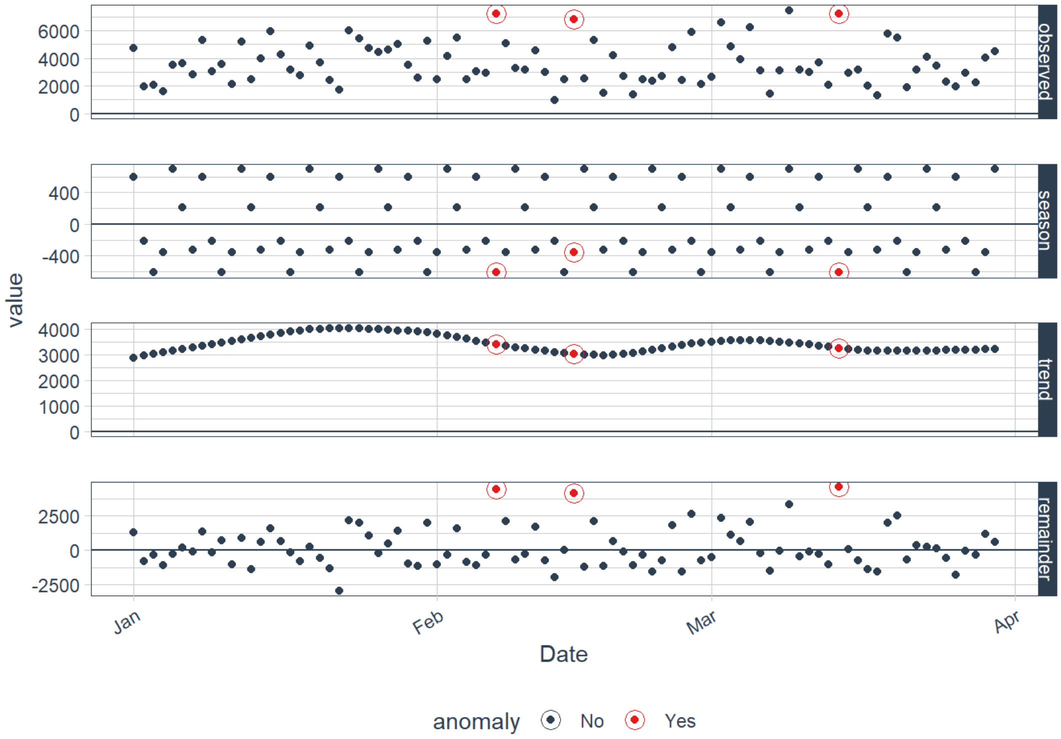
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date
```

```
## frequency = 7 days
```

```
## trend = 30 days
```

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

```
anomaly.detect
```



4. Conclusions

The sales data seems to contain some anomalies as shown by the red points on the graph above It would be important for the marketing team to check them out to ascertain they are not fraud.