Part 4 Anomaly Detection

Noah Kandie

9/11/2021

Part 4: Anomaly Detection

You have also been requested to check whether there are any anomalies in the given sales dataset. The objective of this task being fraud detection

```
# Install required package

library(tinytex)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

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

```
library(tibbletime)
```

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

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

```
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 </>
```

```
library(timetk)
```

Load and preview the dataset

```
df4<-read.csv('http://bit.ly/CarreFourSalesDataset')
head(df4)
```

```
##        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
```

```
# We preview
class(df4)
```

```
## [1] "data.frame"
```

```
dim(df4)
```

```
## [1] 1000    2
```

```
str(df4)
```

```
## 'data.frame':    1000 obs. of  2 variables:
##  $ Date : chr  "1/5/2019" "3/8/2019" "3/3/2019" "1/27/2019" ...
##  $ Sales: num  549 80.2 340.5 489 634.4 ...
```

The dataset has 1000 rows and 2 variables with datetime and interger datatypes

Exploratory Data Analysis

```
# check for missing values
colSums(is.na(df4))
```

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

No missing values

```
# Changing table to tibble
df4$Date<-as.Date(df4$Date,format='%m/%d/%Y')
df_t<-as.tibble(df4)
```

```
## Warning: ‘as.tibble()‘ was deprecated in tibble 2.0.0.
## Please use ‘as_tibble()‘ instead.
## The signature and semantics have changed, see ‘?as_tibble‘.
```

```
is_tibble(df_t)
```

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

```
# totalling the sales based on their common shared dates
sales_agg <- aggregate(df_t['Sales'], by = df_t['Date'],sum)

head(sales_agg)
```

```
##         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
```

```
sales_agg<-as.tibble(sales_agg)
is.tibble(sales_agg)
```

```
## Warning: ‘is.tibble()‘ was deprecated in tibble 2.0.0.
## Please use ‘is_tibble()‘ instead.
```

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

Anomaly Detection

```
sales_agg %>%
  time_decompose(Sales,method = 'stl',frequency = 'auto',trend = 'auto') %>%
  anomalize(remainder,method='gesd',alpha=0.05,max_anoms = 0.2) %>%
  plot_anomaly_decomposition()
```

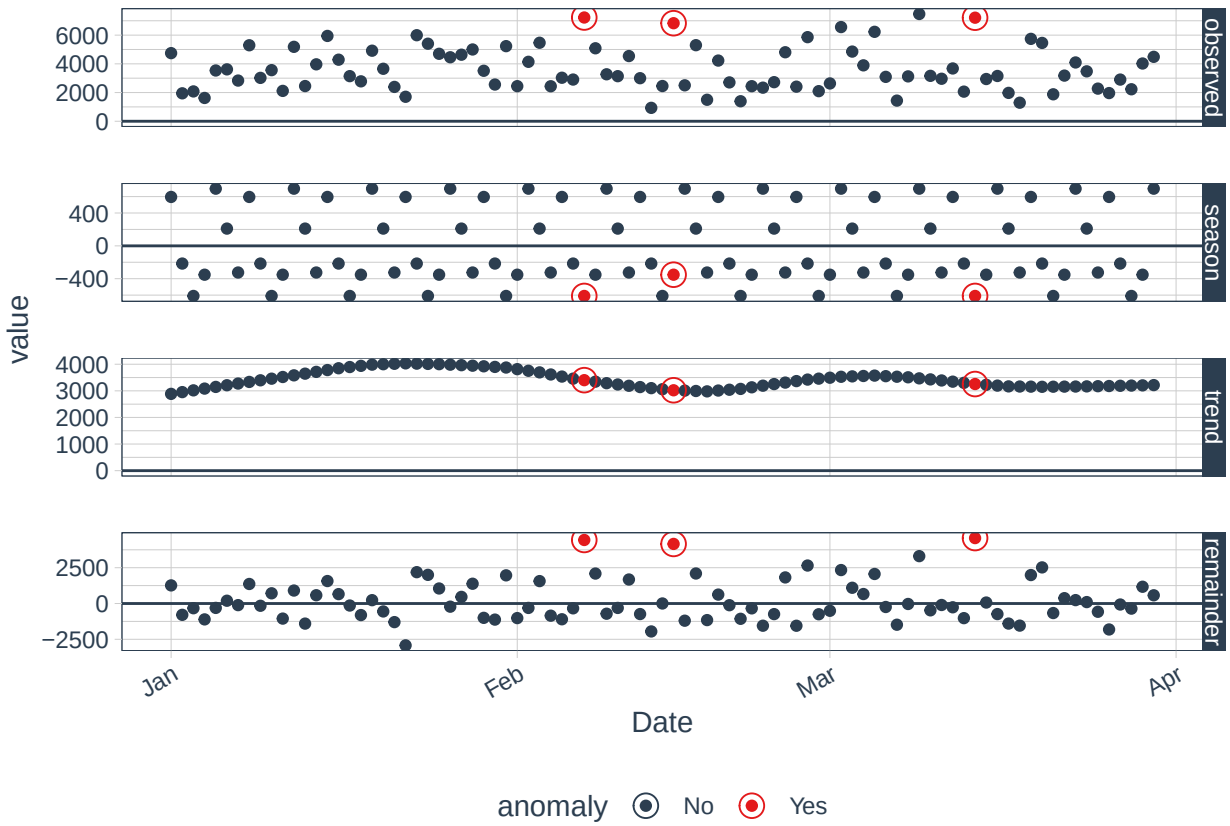
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date
```

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

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

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

```
## Warning: ‘type_convert()‘ only converts columns of type 'character'.
## - ‘df‘ has no columns of type 'character'
```



Conclusion The sales data seems to contain some anomalies shown by the red points on the graph. It is imperative the marketing team should check on the m to ascertain their status