Week14-Part 4: Anomaly Detection

Brendah

2022-04-01

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.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
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(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
# loading the dataset
df <- read.csv("http://bit.ly/CarreFourSalesDataset")
head(df)
```

```
##        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
```

```
# checking the shape of the dataset
dim(df)
```

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

The dataset has 1000 records and 2 variables

```
#changing date to date time.
df$Date <- as.Date(df$Date, "%m/%d/%y")
head(df)
```

```
##         Date    Sales
## 1 2020-01-05 548.9715
## 2 2020-03-08  80.2200
## 3 2020-03-03 340.5255
## 4 2020-01-27 489.0480
## 5 2020-02-08 634.3785
## 6 2020-03-25 627.6165
```

```
# check the number of transactions per day
carrefour_df <- df %>% group_by(Date) %>% tally()
colnames(carrefour_df) <- c('Data_of_Transaction', 'Total_Count')
head(carrefour_df)
```

```
## # A tibble: 6 x 2
##   Data_of_Transaction Total_Count
##   <date>                    <int>
## 1 2020-01-01                   12
## 2 2020-01-02                    8
## 3 2020-01-03                    8
## 4 2020-01-04                    6
## 5 2020-01-05                   12
## 6 2020-01-06                    9
```

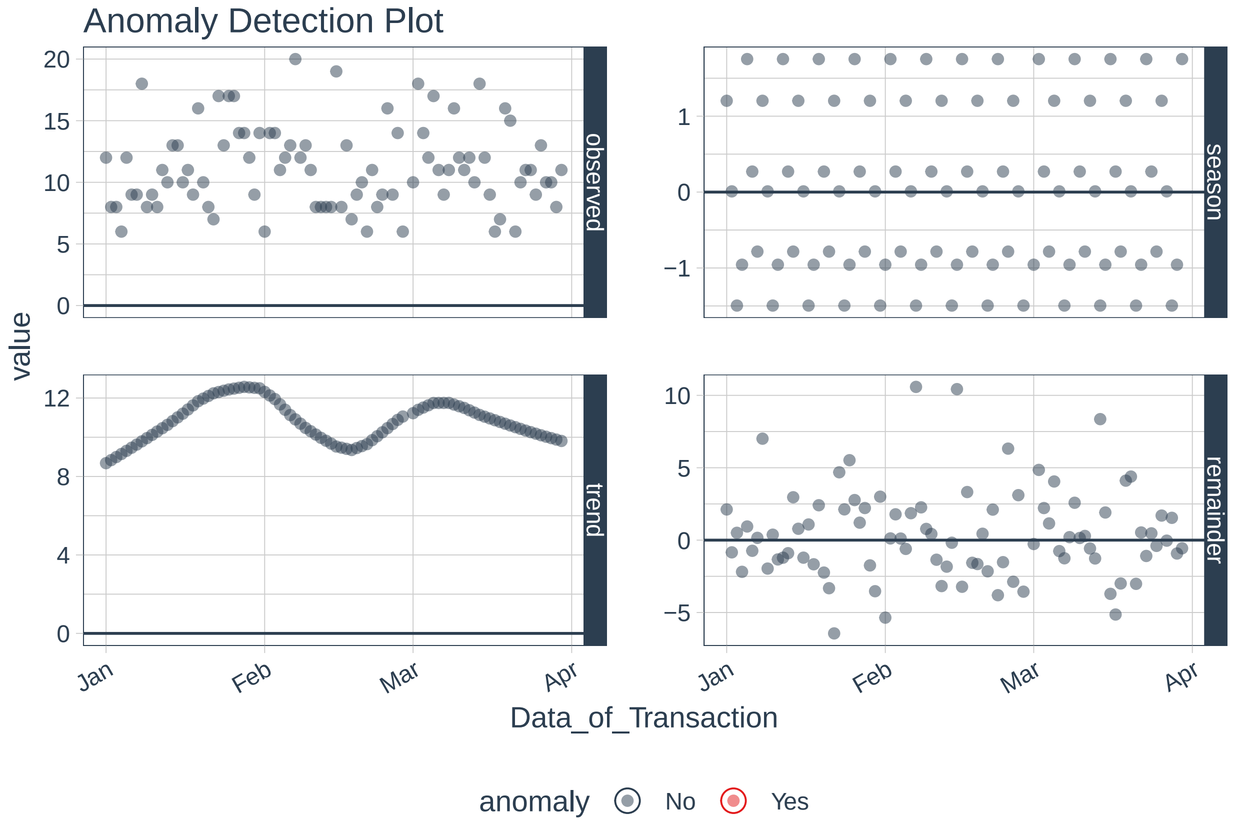
The above list shows the number of customers from 1st January to 6th January

```
# we now plot using plot_anomaly_decomposition() to visualize out data.
carrefour_df %>%
    time_decompose(Total_Count) %>%
    anomalize(remainder) %>%
    plot_anomaly_decomposition(ncol = 2, alpha_dots = 0.5) +
    ggtitle("Anomaly Detection Plot")
```

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

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

```
## trend = 30 days
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```



There are no 'red' which means these transactions have no anomalies