Anomaly detection

kelvin njunge

9/10/2021

Defining the Question

a) Specifying the Question

The objective of this project fraud detection by checking whether there are any anomalies in the given sales dataset that could point out potential fraud activity. ## b) Defining the Metric for Success Exhaustively performing anomaly detection without any errors.

c) Understanding the context

Working as a consultant Data analyst at Carrefour Kenya and are currently undertaking a project that will inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax).This project endeavors to explore a recent marketing dataset and check whether there are any anomalies in the given sales dataset that could point out potential fraud activity.

d) Recording the experimental design

Importing and reading the data Data Cleaning Anomalies detection Conclusions and recommendations

e) Data Relevance

The data was provided by the company (http://bit.ly/CarreFourSalesDataset).

```
library(future)
```

```
## Warning: package 'future' was built under R version 4.1.1
```

```
library(fracdiff)
```

```
## Warning: package 'fracdiff' was built under R version 4.1.1
```

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.1.1
```

```
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 4.1.1
```

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

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

```r
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.1.1
```

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

```
##
## Attaching package: 'tseries'
```

```
## The following object is masked from 'package:future':
##
##     value
```

```r
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
```

```
## Warning: package 'tibble' was built under R version 4.1.1
```

```
## Warning: package 'readr' was built under R version 4.1.1
```

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

```r
library(anomalize)
```

```
## Warning: package 'anomalize' was built under R version 4.1.1
```

```
## == 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(tibbletime)
```

```
## Warning: package 'tibbletime' was built under R version 4.1.1
```

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

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

```
getwd()
```

```
## [1] "C:/Users/Ricky/Documents"
```

```
df <- read.csv("C:\\Users\\Ricky\\Documents\\Supermarket_Sales_Forecasting - Sales.csv")
```

```
# preview the dataset
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
```

```
#stucture of the dataset
str(df)
```

```
## '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 ...
```

```
# reformatting the dates and sortings
df$Date <- as.Date(df$Date, format = "%m/%d/%Y")
df$Date <- sort(df$Date,  decreasing = FALSE)
```

```
# casting as a tibble
data <- as_tbl_time(df, index = Date)
# getting unique daily entries without multiple entries
data <- data %>%
  as_period(period = "daily")
# dimensions of data
dim(data)
```

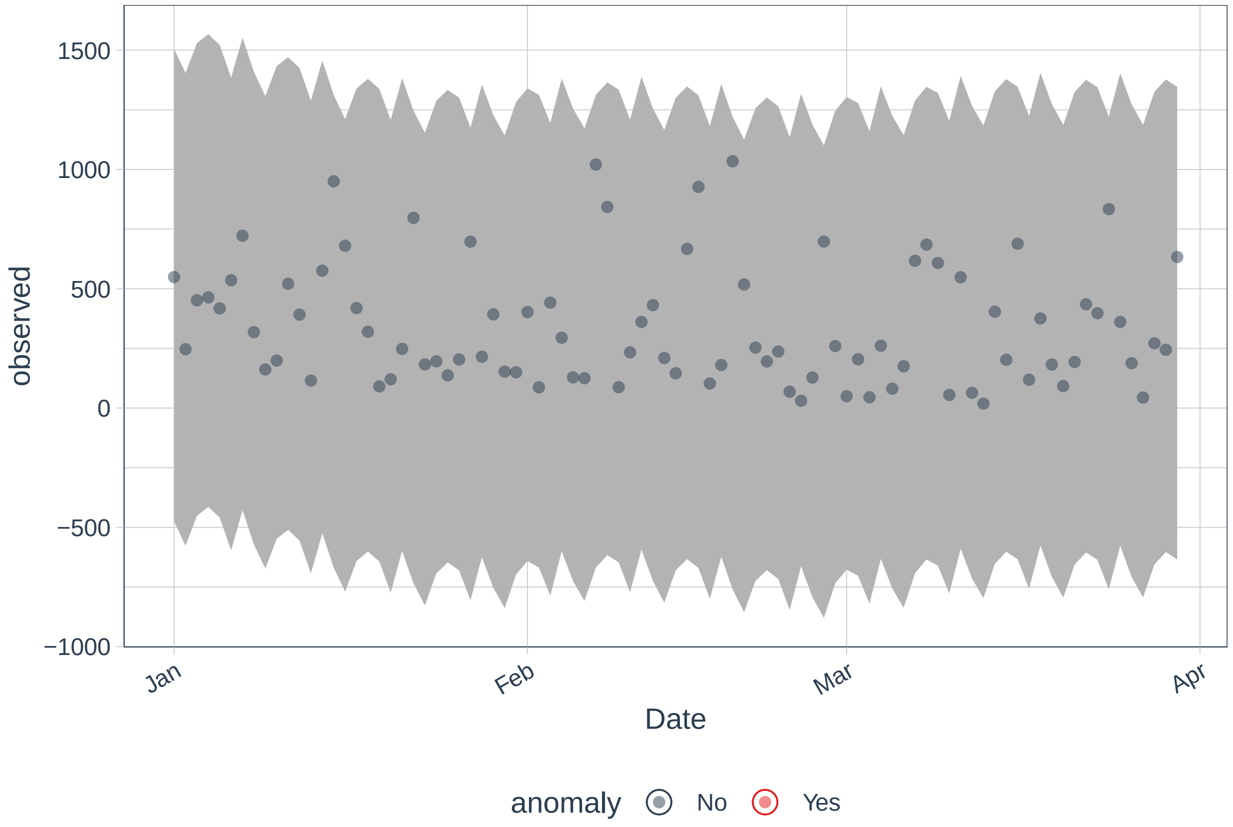
```
## [1] 89  2
```

```
# getting and plotting data for anomaly detection
data %>%
  time_decompose(Sales) %>%
  anomalize(remainder) %>%
  time_recompose() %>%
  plot_anomalies(time_recomposed = TRUE, ncol = 3, alpha_dots = 0.5)
```

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

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

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



Conclusion

There were no anomalies detected in the data.