Anomaly detection

Kennedy Muriuki

22/09/2020

Anomaly detection in R

```
# loading the dataset
df <- read.csv(file.choose())
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
```

```
# converting into date tome format
df$Date<- as.Date(df$Date,format = "%m/%d/%Y")

df[["Date"]] <- as.POSIXct(df$Date)
```

```
# converting into a tibble
library(tibbletime)
```

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

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

```
#df = df%>% as_tibble()
```

```
# loading the required libraries

library(tidyverse)
```

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

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------------ tidyverse_conflic
## x dplyr::filter() masks tibbletime::filter(), 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 </>
```

```
library(dplyr)
```

```
df <- df %>%
tibbletime::as_tbl_time(index = Date)
```

```
df %>%
    time_decompose(Sales, method = "stl", frequency = "auto", trend = "auto") %>%
    anomalize(remainder, method = "gesd") %>%
    time_recompose() %>%
    plot_anomalies(time_recomposed = TRUE, ncol = 3, alpha_dots = 0.5)
```

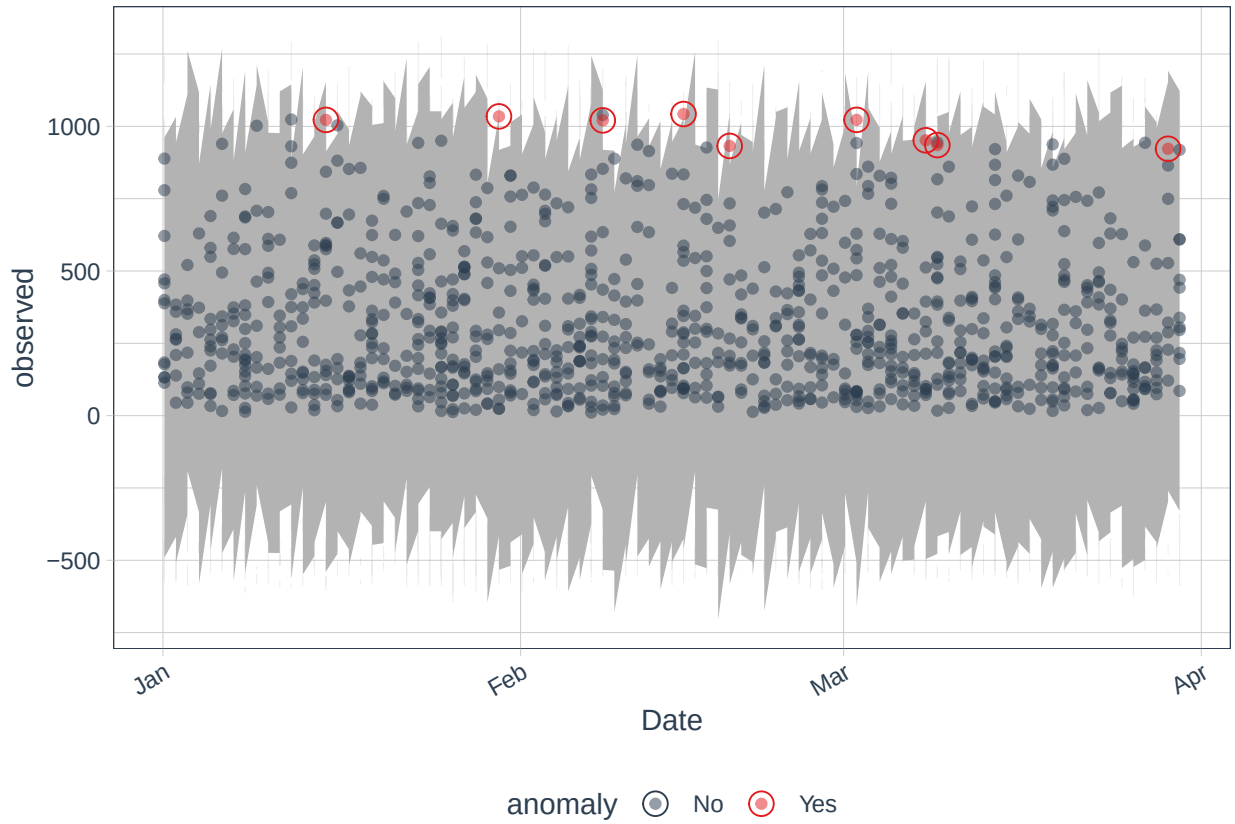
```
## Note: Index not ordered. tibbletime assumes index is in ascending order. Results may not be as desire

## frequency = 12 seconds

## Note: Index not ordered. tibbletime assumes index is in ascending order. Results may not be as desire

## trend = 12 seconds

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



In the analysis a few anomalies were detected. Most of them were centered between the month of Feb and mid March