Anomaly detection

Daisy Lynn

2022-04-02

Objective

fraud detection

```
library(readr)
library(caret)
```

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

```
## Loading required package: lattice
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggplot2)
library(tidyverse)
```

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

```
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1
```

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

```
library(dplyr)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
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(tibbletime)
```

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

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

#Loading Dataset

```
sales <- read_csv("http://bit.ly/CarreFourSalesDataset")
```

```
## Rows: 1000 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Date
## dbl (1): Sales
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(sales)
```

```
## # A tibble: 6 x 2
##   Date      Sales
##   <chr>     <dbl>
## 1 1/5/2019  549.
## 2 3/8/2019   80.2
## 3 3/3/2019  341.
## 4 1/27/2019 489.
## 5 2/8/2019  634.
## 6 3/25/2019 628.
```

```
#checking missing values
colSums(is.na(sales))
```

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

no missing values

changing Date data type

```
sales$Date <- as.Date(sales$Date, "%m/%d/%Y")
```

```
sales$Date <- as.POSIXct(sales$Date)
```

```
sales.tb <- as_tibble(sales)
sales.tb
```

```
## # A tibble: 1,000 x 2
##    Date                Sales
##    <dttm>              <dbl>
##  1 2019-01-04 16:00:00 549.
##  2 2019-03-07 16:00:00  80.2
##  3 2019-03-02 16:00:00 341.
##  4 2019-01-26 16:00:00 489.
##  5 2019-02-07 16:00:00 634.
##  6 2019-03-24 17:00:00 628.
##  7 2019-02-24 16:00:00 434.
##  8 2019-02-23 16:00:00 772.
##  9 2019-01-09 16:00:00  76.1
## 10 2019-02-19 16:00:00 173.
## # ... with 990 more rows
```

Anomalize

```
anomalies <- sales.tb %>%
 time_decompose(Sales, merge = TRUE) %>%
 anomalize(remainder) %>%
 time_recompose()
```

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

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

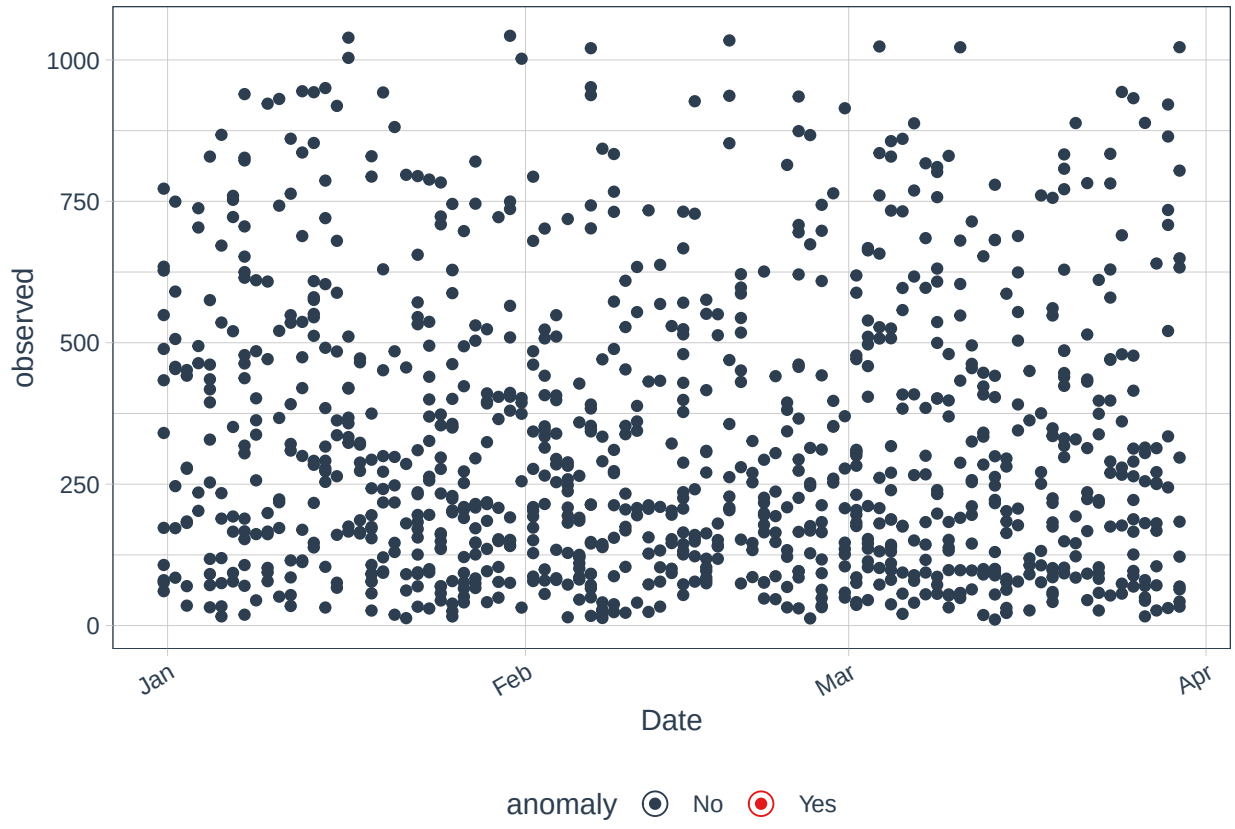
```
## frequency = 12 seconds
```

```
## 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
```

```
anomalies %>%
  plot_anomalies(ncol = 3)
```



No anomalies detected

Using IQR

```
sales.tb %>%
 time_decompose(Sales, method = "stl",frequency = "auto", trend = "auto") %>%
 anomalize(remainder, method = "iqr", alpha = 0.05, max_anoms = 0.2) %>%
 plot_anomaly_decomposition()
```

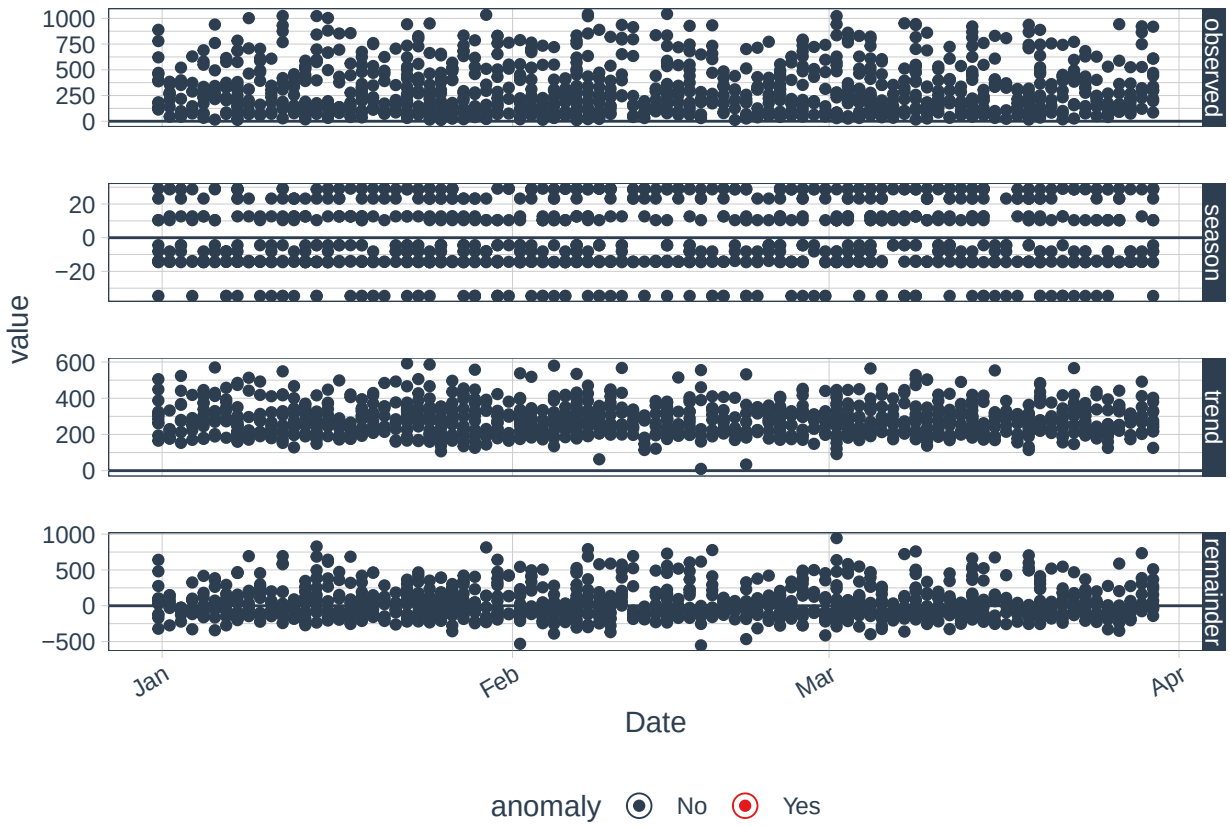
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date

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

## frequency = 12 seconds

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

## trend = 12 seconds
```



IQR method found no anomalies

Using GESD

```
sales.tb %>%
 time_decompose(Sales, method = "stl",frequency = "auto", trend = "auto") %>%
 anomalize(remainder, method = "gesd", alpha = 0.05, max_anoms = 0.2) %>%
 plot_anomaly_decomposition()
```

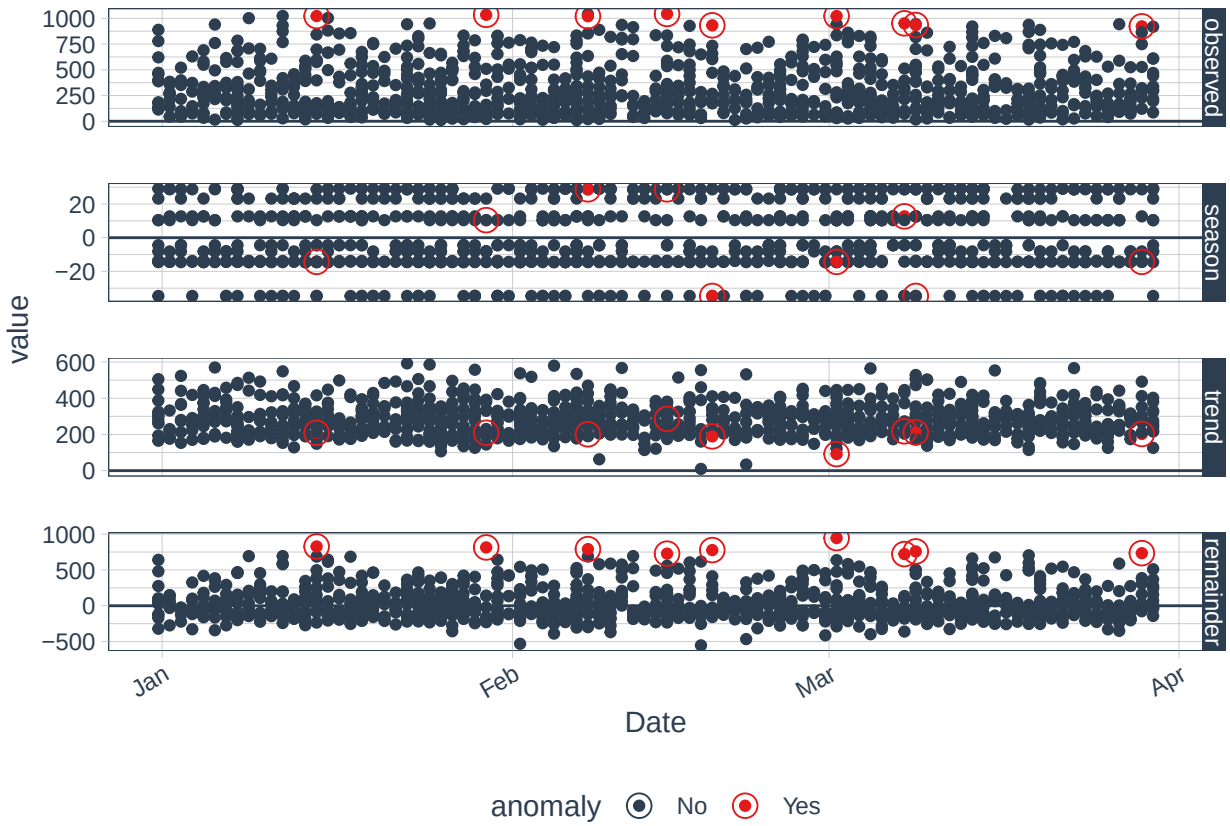
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date

## 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
```



GESD detected a few anomalies

Conclusion

the anomalies could suggest presence of fraud or they were caused due to a spike in sales that differed from the normal

Marketing team could look into these months to identify exact cause