Week 14 IP- Anomaly Detection

Billiah

2022-04-01

```
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(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(dplyr)
library(tibble)
library(tsibble)
```

```
##
## Attaching package: 'tsibble'
```

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

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
# Loading the data
data <- read.csv("http://bit.ly/CarreFourSalesDataset")
head(data)
```

```
##        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
```

```
dim(data)
```

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

The data has 1000 rows and 2 columns.

```
str(data)
```

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

```
Date = as.Date(data$Date)
```

```
# Convert df to a tibble
data_1 <- as_tibble(data)
class(data_1)
```

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

```
# Converting a data.frame to a `tbl_time`
data_1$Date <- as.Date(data_1$Date, format = "%m/%d/%y")

#Convertion to POCIXct type
data_1$Date <- as.POSIXct(data_1$Date)
```

```
anomalized <- data_1 %>%
 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
```

```
anomalized %>% glimpse()
```

```
## Rows: 1,000
## Columns: 11
## $ Date          <dttm> 2019-12-31 16:00:00, 2019-12-31 16:00:00, 2019-12-31 16~
## $ Sales         <dbl> 457.4430, 399.7560, 470.6730, 388.2900, 132.7620, 132.02~
## $ observed      <dbl> 548.9715, 80.2200, 340.5255, 489.0480, 634.3785, 627.616~
## $ season        <dbl> -14.359190, -4.462252, 28.744495, 23.243172, -13.844108,~
## $ trend         <dbl> 445.2248, 445.5012, 445.7776, 435.9271, 426.0767, 416.19~
## $ remainder     <dbl> 118.105886, -360.818930, -133.996555, 29.877684, 222.145~
## $ remainder_l1  <dbl> -917.358, -917.358, -917.358, -917.358, -917.358, -917.3~
## $ remainder_l2  <dbl> 946.1539, 946.1539, 946.1539, 946.1539, 946.1539, 946.15~
## $ anomaly       <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", "N~
## $ recomposed_l1 <dbl> -486.4924, -476.3191, -442.8360, -458.1877, -505.1254, -~
## $ recomposed_l2 <dbl> 1377.020, 1387.193, 1420.676, 1405.324, 1358.387, 1372.7~
```

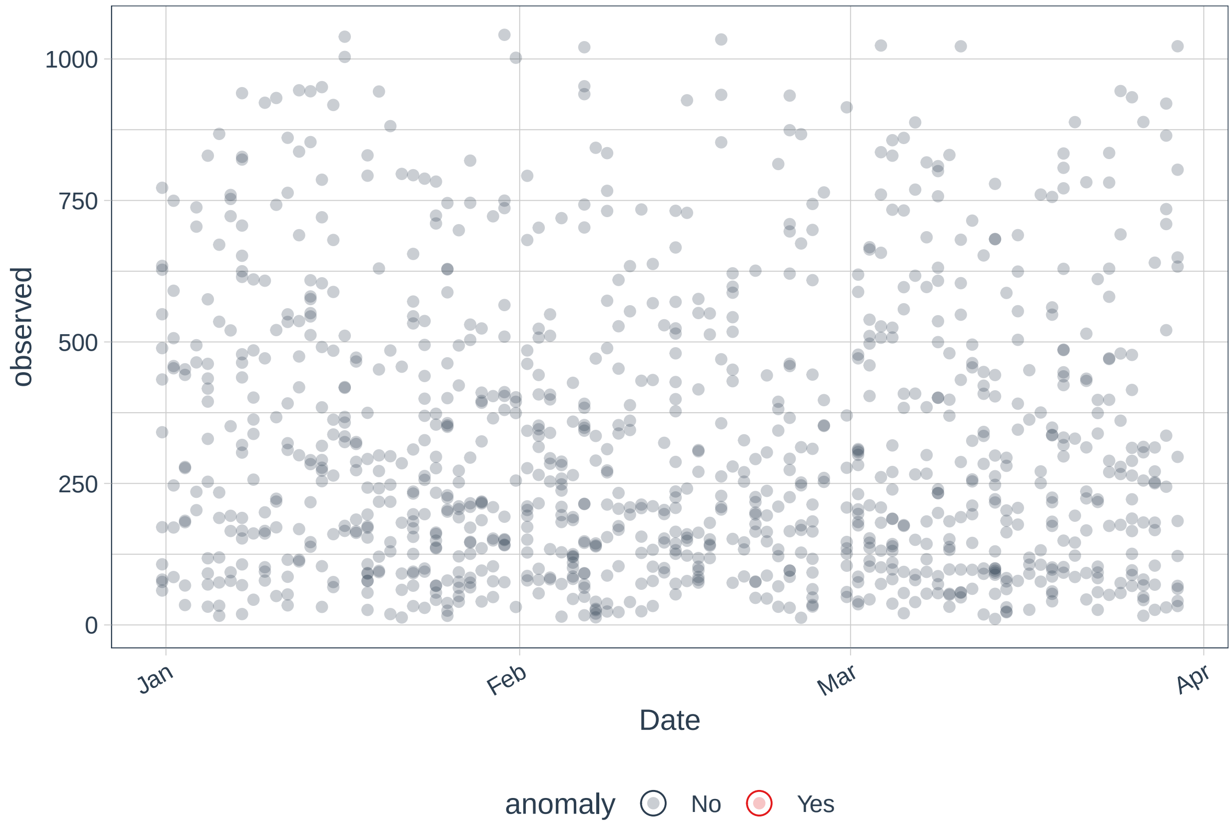
observed represents the observed values

season is the seasonal or cyclic trend. The default for daily data is a weekly seasonality.

Remainder: This is what we want to analyze for outliers. It is simply the observed minus both the season and trend.

Setting merge = TRUE keeps the original data with the newly created columns.

```
anomalized %>%
plot_anomalies(ncol = 3, alpha_dots = 0.25)
```



Majority of the observed are not anomalies.

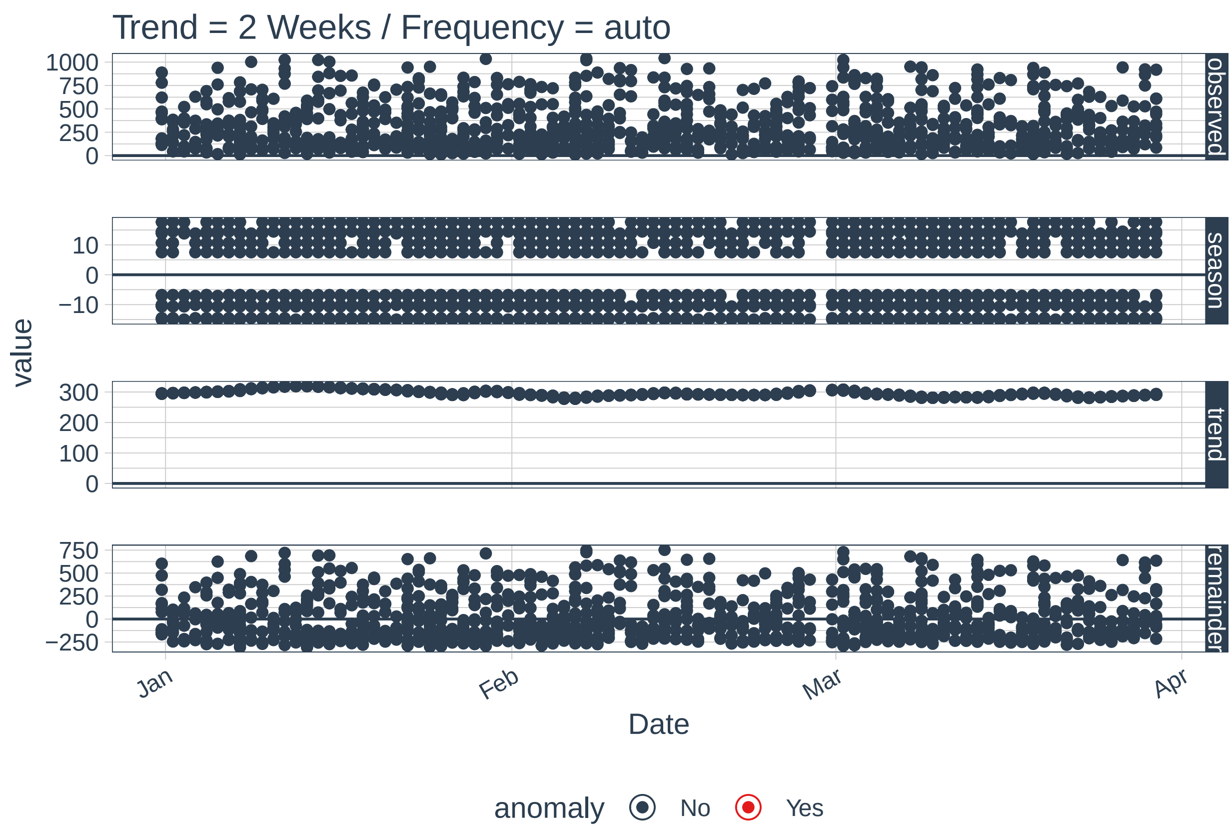
```
anomalized %>%
  time_decompose(Sales, frequency = "auto", trend = "2 weeks")%>%
  anomalize(remainder)%>%
  plot_anomaly_decomposition()+
  ggtitle("Trend = 2 Weeks / Frequency = auto ")
```

```
## frequency = 11 seconds
```

```
## Warning in lubridate::floor_date(x, unit): Multi-unit not supported for weeks.
## Ignoring.
```

```
## Warning in lubridate::ceiling_date(x, unit): Multi-unit not supported for weeks.
## Ignoring.
```

```
## trend = 156 seconds
```



There are little to no anomalies. Adjusting the parameters further will give more clarity.

```
data_1%>%
  time_decompose(Sales)%>%
  anomalize(remainder, alpha = 0.05)%>%
  time_recompose()%>%
  plot_anomalies(time_recompose = T)+
  ggtitle("alpha = 0.05")
```

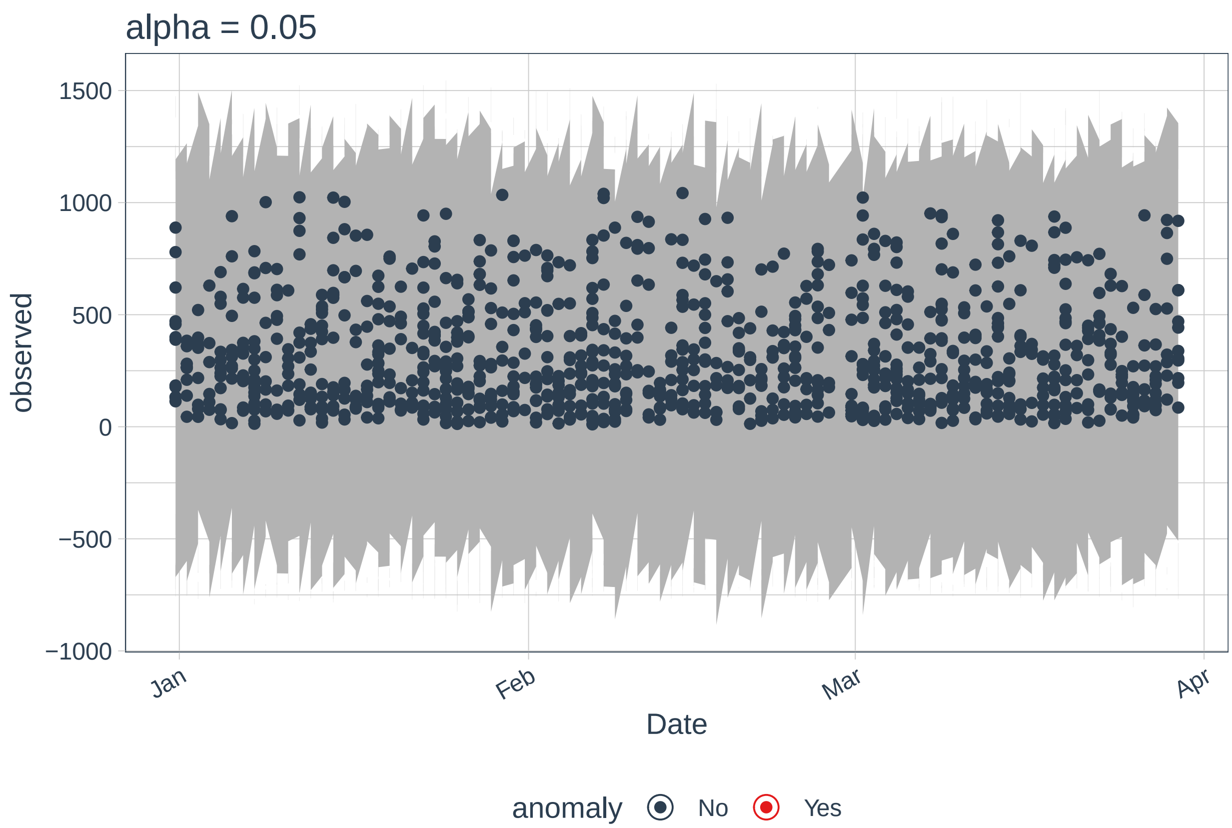
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date

## 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
```




```
data_1%>%
  time_decompose(Sales)%>%
  anomalize(remainder, alpha = 0.025)%>%
  time_recompose()%>%
  plot_anomalies(time_recompose = T)+
  ggtitle("alpha = 0.025")
```

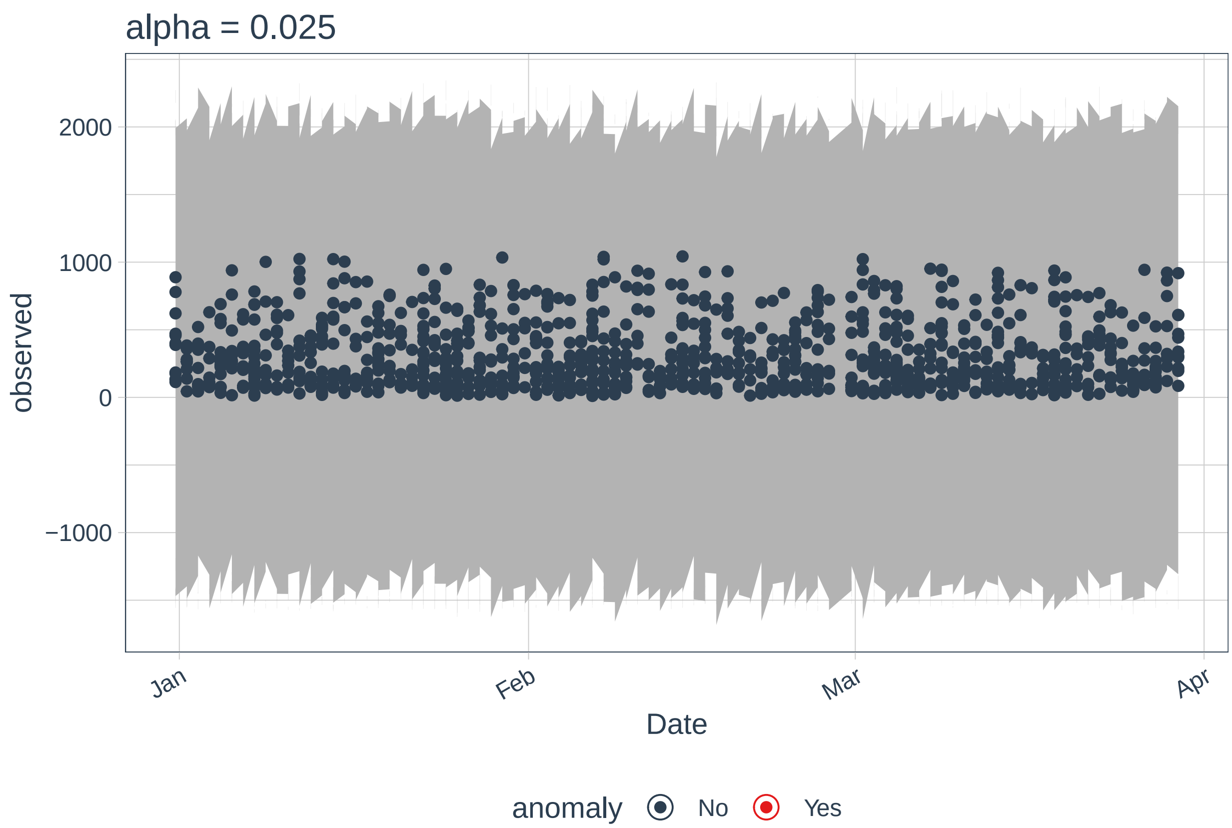
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date

## 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
```




```
data_1%>%
  time_decompose(Sales)%>%
  anomalize(remainder, alpha = 0.2, max_anoms = 0.2)%>%
  time_recompose()%>%
  plot_anomalies(time_recompose = T)+
  ggtitle("Anomaly limit = 20%")
```

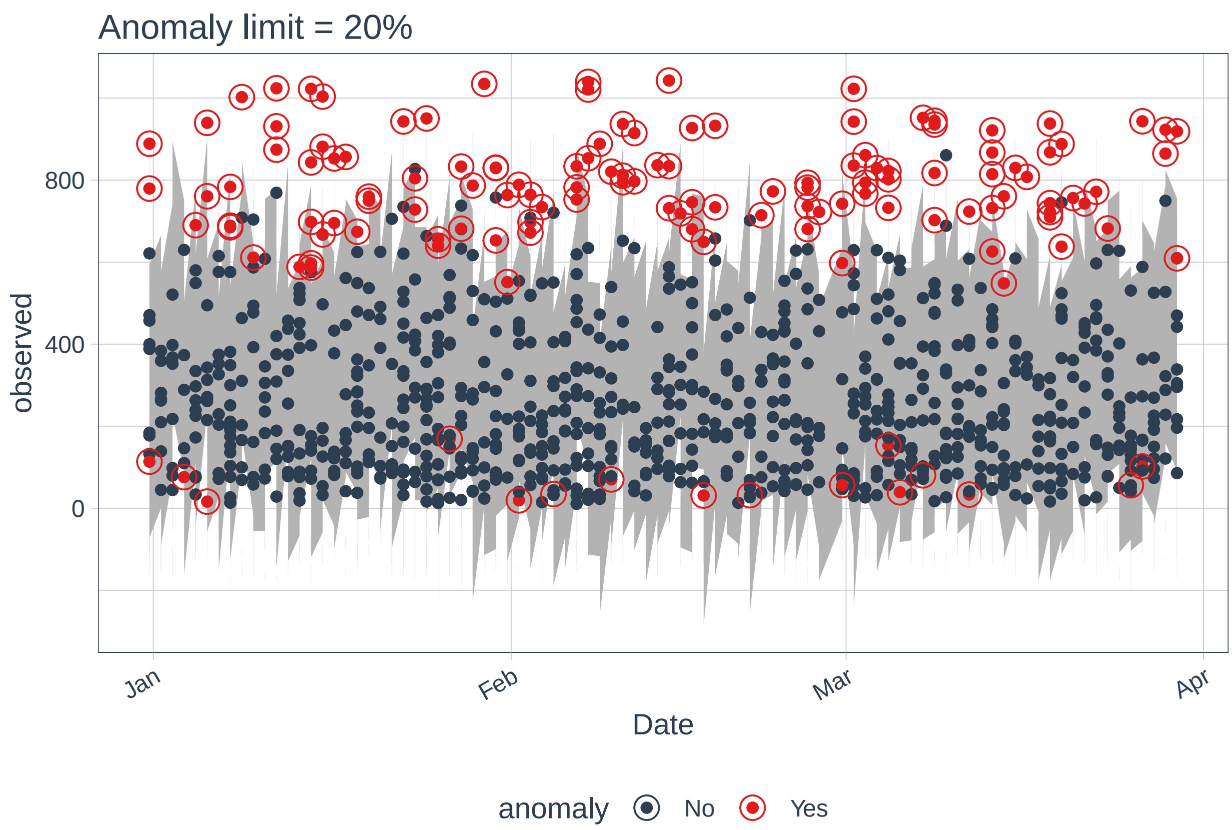
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date

## 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
```



The anomalies can be clearly observed now.

```
data_1%>%
  time_decompose(Sales)%>%
  anomalize(remainder, alpha = 0.2, max_anoms = 0.05)%>%
  time_recompose()%>%
  plot_anomalies(time_recompose = T)+
  ggtitle("Anomaly limit = 5%")
```

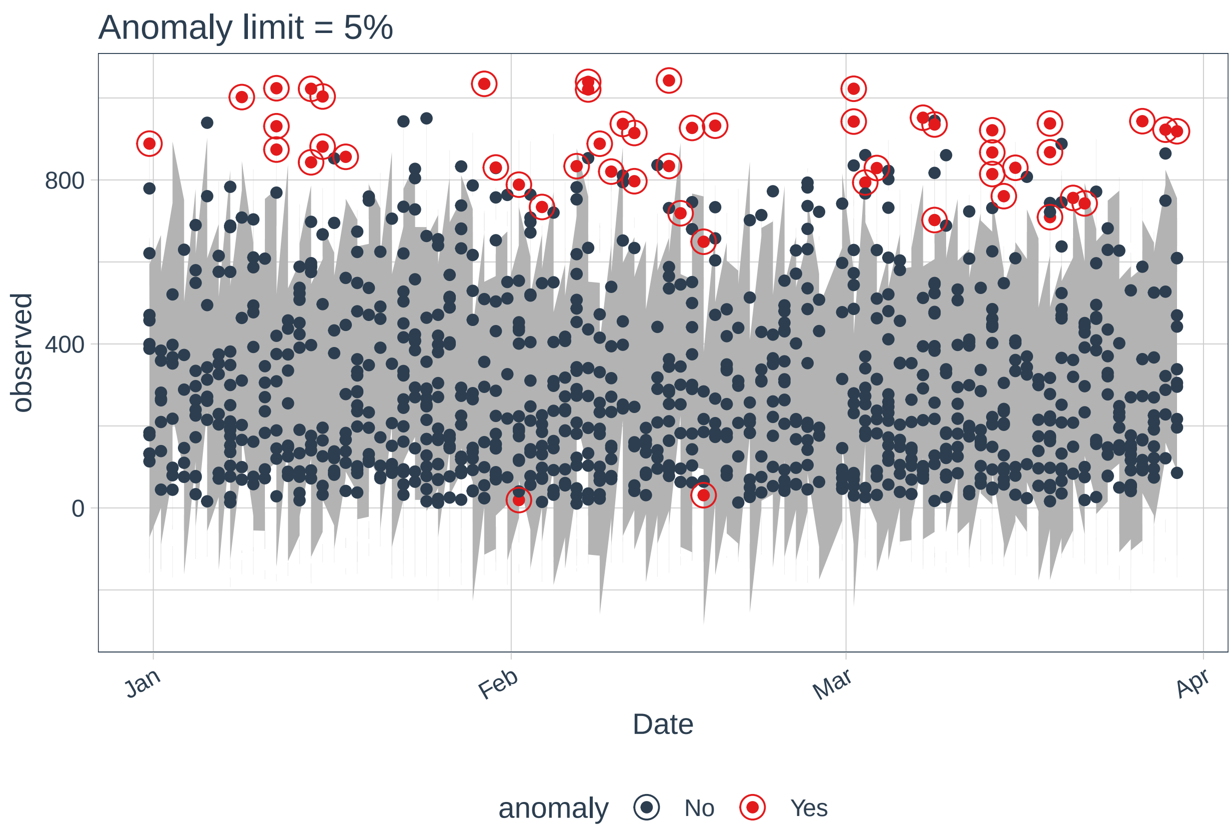
```
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date

## 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
```



From the above observations, sales extremely increased in January, February all the way to April, with very minimal decreases between February and March.

This may be an indication of the success of the measures put in place.

This indicates that the marketing campaigns have been effective, the customers have been retained and probably increased. Also, it may be as a result of increased app performance, and also implies improved product quality.

RECOMMENDATION

The measures put in place to increase sales should be practised more and also improve on the current measures.