# CarrefourAnomalyDetection

Joy Machuka

9/11/2021

## Problem Definition

As a Data analyst at Carrefour Kenya working for the marketing department Carrefour you are to come up with relevant marketing strategies that will result in the higher no. of sales. Amongst the processes is checking for anomalies in the data.

## Specifying the Question

Identify Anomalies in the Sales of Carrefour

## Metric of Success

When we are able to detect anomalies in the sales trend.

#Context Anomalies could be used in sales data to detect fraud or to detect abnormal trends in sales made hence further investigation. In our case we are going to check for both.

## Experimental Design

Load the Data Check the Data Data cleaning Exploratory Data Analysis Implement the Solution Challenge the Solution

## Data Sourcing

```r
# loaddata
sales <- read.csv("http://bit.ly/CarreFourSalesDataset")
head(sales)
```

```
##        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(sales)
```

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

Data has 1000 columns and 2 columns

```
class(sales)
```

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

```
str(sales)
```

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

Our columns are date and sales.

```
#checking for missing values
anyNA(sales)
```

```
## [1] FALSE
```

No missing values

```
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

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

```
library(tibbletime)
```

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

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

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

sales_ts = sales %>%
  as_tibble() %>%
  as_tbl_time(Date) %>%
  arrange(Date) %>%
  as_period("daily")
```

```r
anyNA(sales_ts)
```

```
## [1] FALSE
```

Implementing solution

```r
# install.packages("anomalize")
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 </>
```

```r
library(dplyr)
```

```r
pkg <- c('timetk')
# install.packages(pkg)
```

```r
library(timetk)
```

```r
#function
sales_anomaly <- sales_ts %>%
  time_decompose(Sales) %>%
  anomalize(remainder,max_anoms = 0.2, alpha=0.05) %>%
  time_recompose() %>% glimpse()
```

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

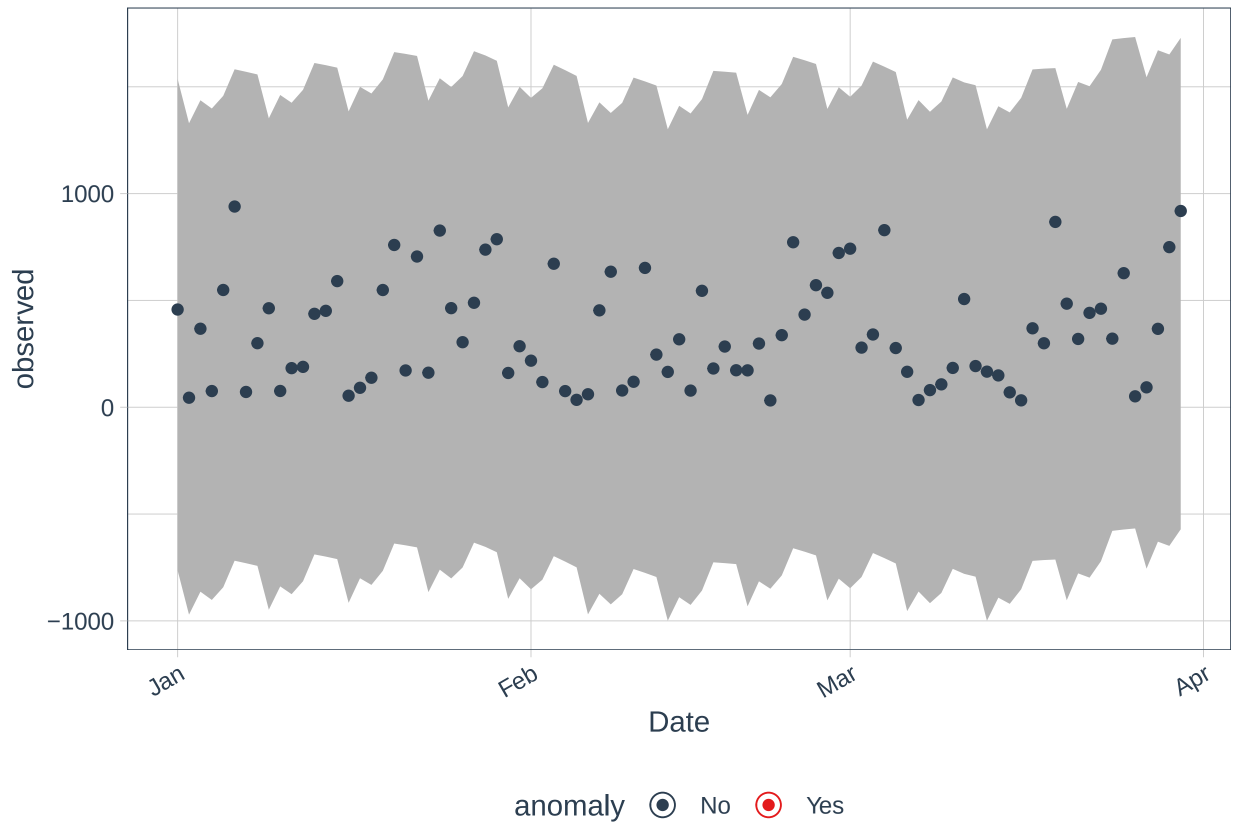
```
## trend = 30 days
```

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

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

```
## Rows: 89
## Columns: 10
## $ Date          <date> 2019-01-01, 2019-01-02, 2019-01-03, 2019-01-04, 2019-01~
## $ observed      <dbl> 457.4430, 44.5935, 367.5525, 75.7785, 548.9715, 939.5400~
## $ season          <dbl> 71.60220, -137.93561, -32.93877, -73.88690, -17.91742, 1~
## $ trend           <dbl> 296.3521, 298.8125, 301.2728, 303.7331, 307.0337, 310.33~
## $ remainder       <dbl> 89.488658, -116.283358, 99.218460, -154.067738, 259.8551~
## $ remainder_l1    <dbl> -1131.823, -1131.823, -1131.823, -1131.823, -1131.823, --~
## $ remainder_l2    <dbl> 1168.64, 1168.64, 1168.64, 1168.64, 1168.64, 1168.64, 11~
## $ anomaly         <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", "N~
## $ recomposed_l1 <dbl> -763.8688, -970.9462, -863.4891, -901.9769, -842.7068, --~
## $ recomposed_l2 <dbl> 1536.594, 1329.517, 1436.974, 1398.486, 1457.756, 1582.0~
```

```
# Plot anomalies
sales_anomaly %>% plot_anomalies(time_recomposed = TRUE)
```



Our data has no anomalies.