IP - Anomaly Detection

Data Analysis Objectives

Perform anomaly detection analysis on sales data and report any inconsistencies.

Understanding context

Carrefour Kenya seeks to undertake 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). Anomaly detection analysis on sales data would be helpful in identifying if there is occurrence of any fraudulent transactions and when they occured to perform further investigation.

Experimental Design

- Business Understanding
- Loading Data
- EDA
- Implementation of the solution
- Findings

```
# Load packages
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
## Warning: package 'tibble' was built under R version 4.0.4
```

```
## Warning: package 'tidyr' was built under R version 4.0.4
```

```
## Warning: package 'readr' was built under R version 4.0.4
```

```
## Warning: package 'purrr' was built under R version 4.0.4
```

```
## Warning: package 'dplyr' was built under R version 4.0.4
## Warning: package ’stringr’ was built under R version 4.0.4
```

```
## Warning: package ’forcats’ was built under R version 4.0.4
```

```
## -- 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 4.0.5
```

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

```
## Warning: package ’tibbletime’ was built under R version 4.0.5
```

```
##
## Attaching package: ’tibbletime’
```

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

```r
library(lubridate)
```

```
## Warning: package ’lubridate’ was built under R version 4.0.4
```

```
##
## Attaching package: ’lubridate’
```

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

```r
# Load data
sales_data <- read.csv(’Supermarket_Sales_Forecasting - Sales.csv’, stringsAsFactors = FALSE)
head(sales_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
```

```
summary(sales_data$Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   10.68  124.42  253.85  322.97  471.35 1042.65
```

```
# Convertieng date column into date object
sales_data$Date <- as_date(mdy(sales_data$Date))
```

```
# Get the range of dates in the data
paste(c('Start:'), min(sales_data$Date))
```

```
## [1] "Start: 2019-01-01"
```

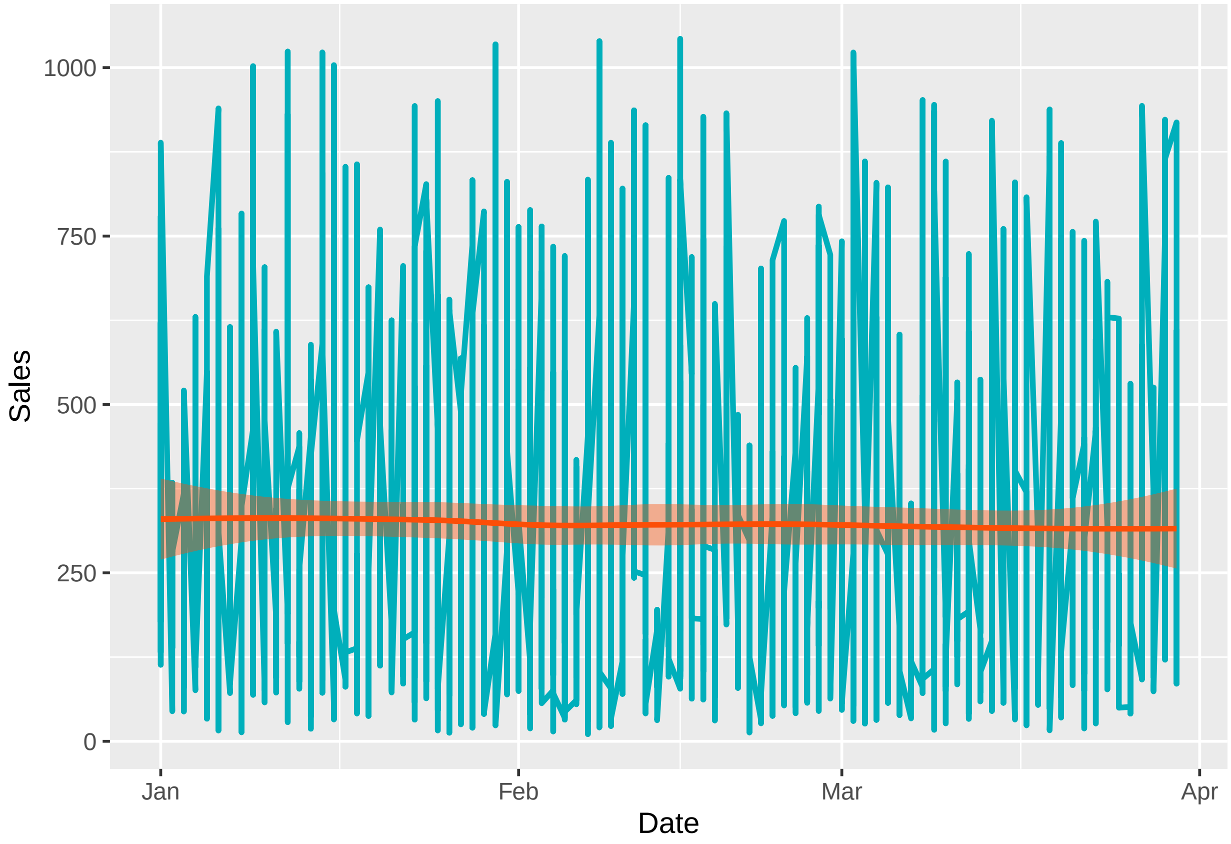
```
paste(c('End:'), max(sales_data$Date))
```

```
## [1] "End: 2019-03-30"
```

Sales data ranges over a period of 3 months.

```
# Visualize sales data
ggplot(data = sales_data, aes(x=Date, y = Sales))+
  geom_line(color = "#00AFBB", size = 1)+
  stat_smooth(color = "#FC4E07", fill = "#FC4E07",
  method = "loess")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```r
# Get total sales and average sales made per day
summary_sales <- sales_data %>% group_by(Date) %>% summarise_all(list(mean = mean, sum = sum))
head(summary_sales,  3)
```

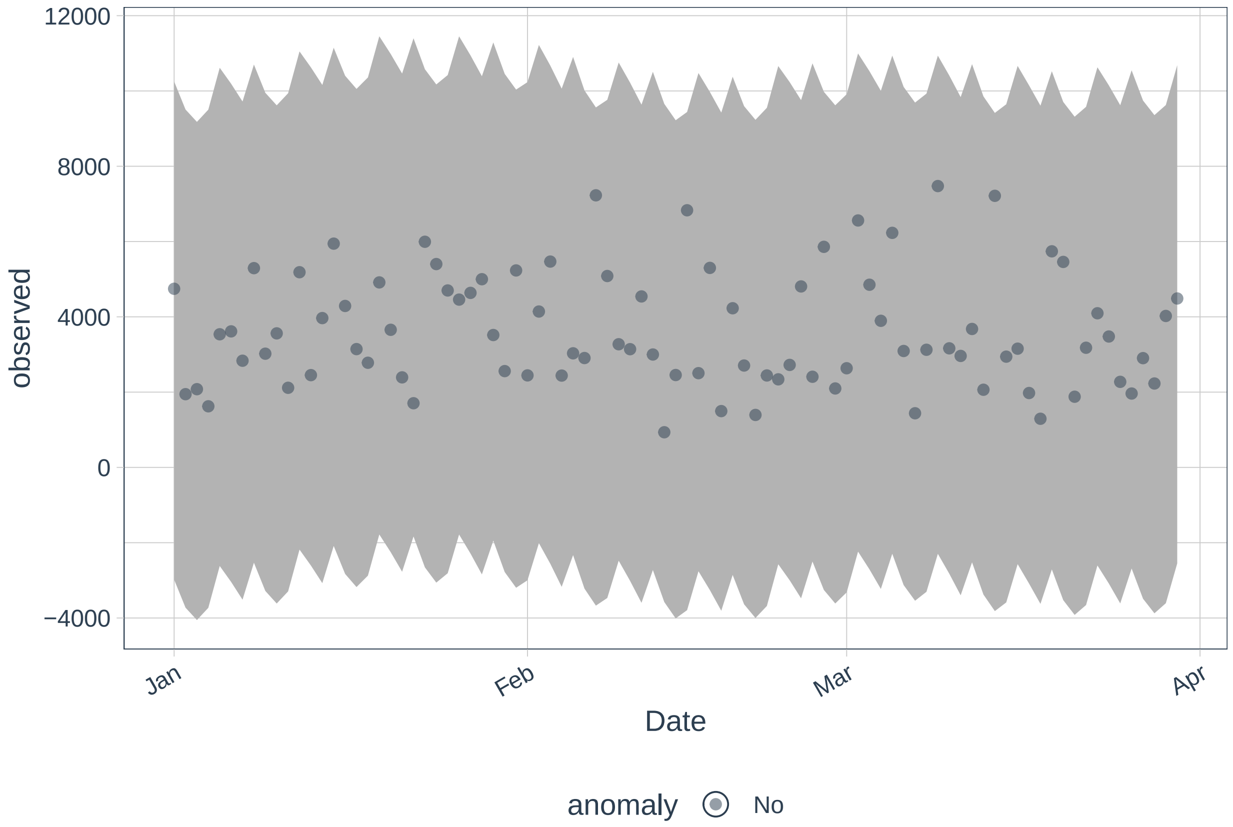
```
## # A tibble: 3 x 3
##   Date        mean   sum
##   <date>     <dbl> <dbl>
## 1 2019-01-01  395. 4745.
## 2 2019-01-02  243. 1946.
## 3 2019-01-03  260. 2078.
```

```r
# Anomaly detection using total sales per day
summary_sales %>%
  as_tbl_time(Date) %>%
  time_decompose(sum) %>%
  anomalize(remainder) %>%
  time_recompose()%>%
  plot_anomalies(time_recomposed = TRUE, ncol = 3, alpha_dots = 0.5)
```

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

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

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

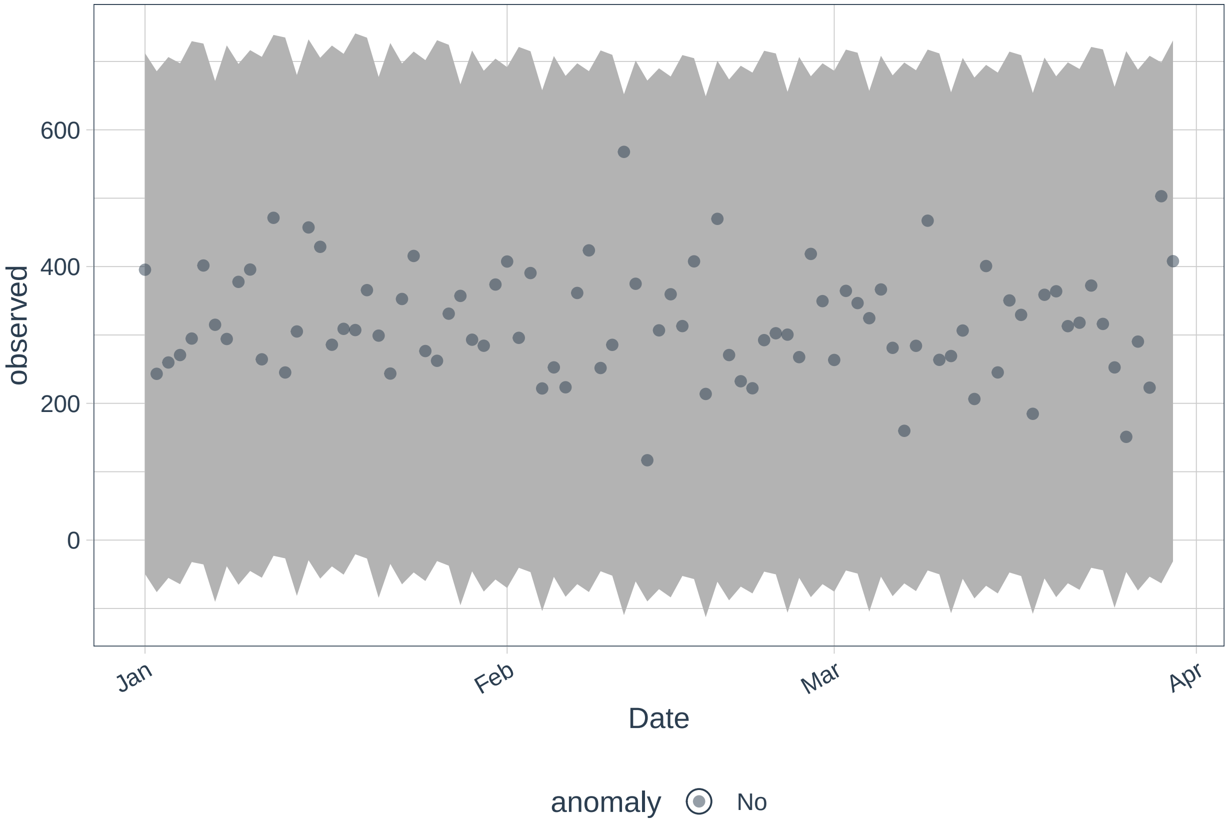


NO anomalies present in the data

```r
# Anomaly detection using average sales per day
summary_sales %>%
  as_tbl_time(Date) %>%
  time_decompose(mean) %>%
  anomalize(remainder) %>%
  time_recompose()%>%
  plot_anomalies(time_recomposed = TRUE, ncol = 3, alpha_dots = 0.5)
```

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

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



NO anomalies detected in the data

Summary of Findings.

Analysis of the data indicates no fraudulent activity in the data