Fraud Detection

Cynthia Mwadime

2022-06-10

Anomaly Detection

Overview

We are tasked with checking whether there are any anomalies in the given sales dataset for the purpose of fraud detection.

Loading the Data and Libraries

```r
# Loading tidyverse and anomalize
library(tidyverse)
```

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

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## 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()
```

```r
#devtools::install_github("JesseVent/crypto")
library(dplyr)
library(crypto2)
library(anomalize,warn.conflicts = FALSE)
```

```
## == 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)
```

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

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

```r
# reading the data
sales <- read.csv('http://bit.ly/CarreFourSalesDataset')
View(sales)
```

```r
# checking the structure of our data
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 ...
```

- We have 1000 observations and 2 variables.
- We'll have to change the date datatype

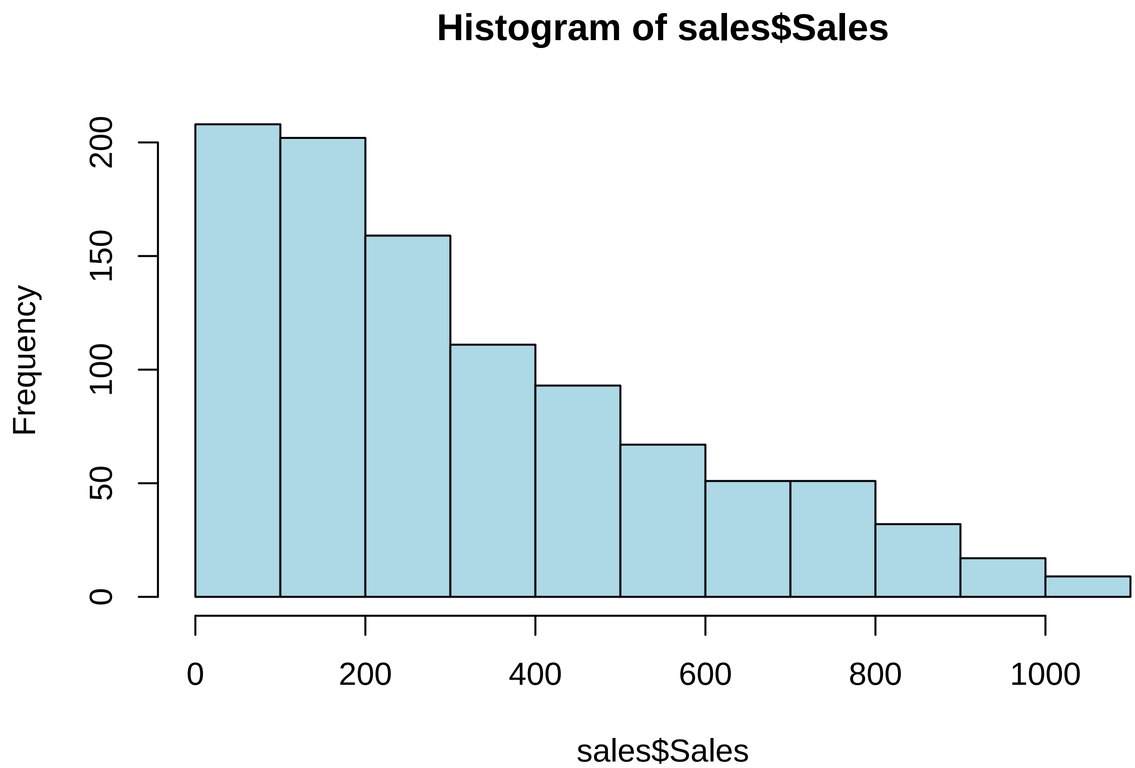
```r
# converting variables to our preferred format
sales$Date <- as.Date(sales$Date, "%m/%d/%Y")
```

```r
# confirming change
str(sales)
```

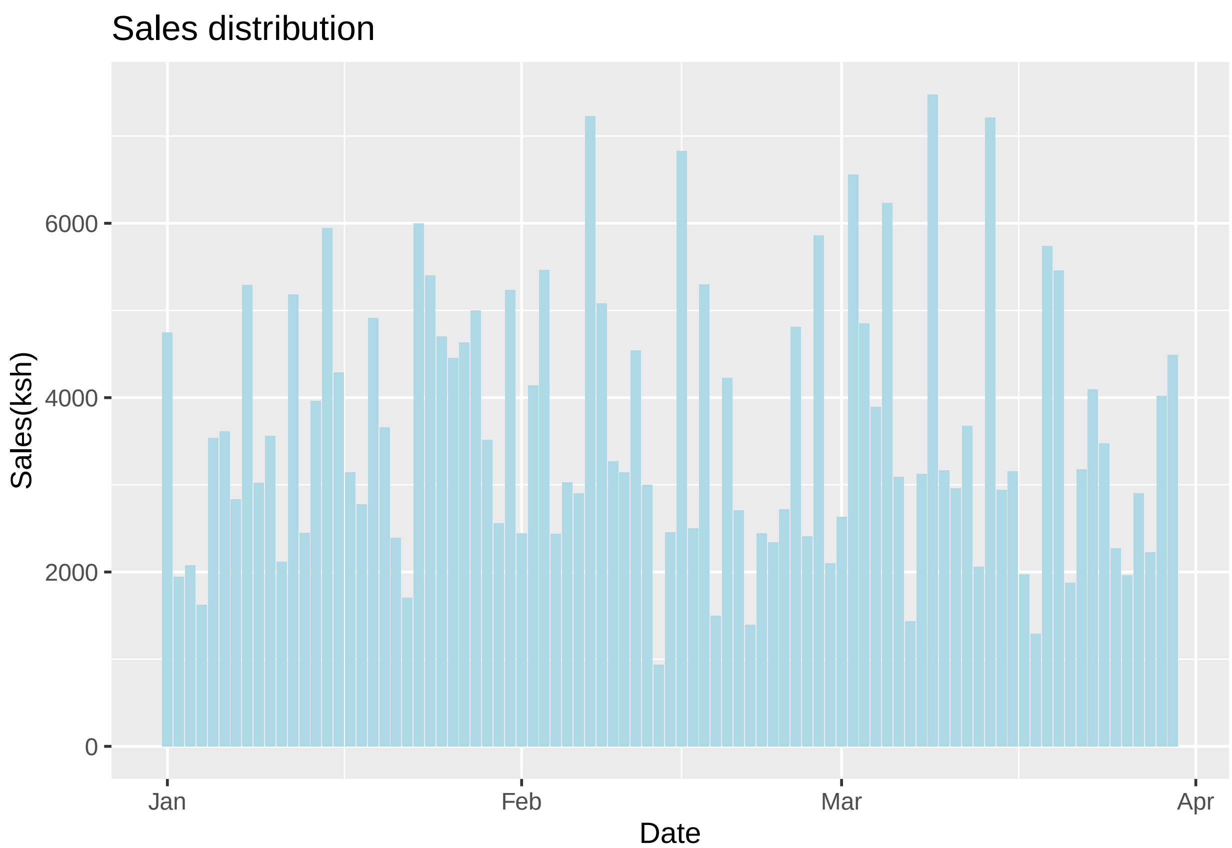
```
## 'data.frame':    1000 obs. of  2 variables:
##  $ Date : Date, format: "2019-01-05" "2019-03-08" ...
##  $ Sales: num  549 80.2 340.5 489 634.4 ...
```

Visualizing our Sales

```r
# frequency of sales
hist(sales$Sales,col="lightblue")
```




```r
#Checking the distribution over time
library(ggplot2)
ggplot(data = sales, aes(x = Date, y = Sales)) +
      geom_bar(stat = "identity", fill = "lightblue") +
      labs(title = "Sales distribution",
           x = "Date", y = "Sales(ksh)")
```




```
# Ordering the data by Date
sales = sales %>% arrange(Date)
head(sales)
```

```
##         Date   Sales
## 1 2019-01-01 457.443
## 2 2019-01-01 399.756
## 3 2019-01-01 470.673
## 4 2019-01-01 388.290
## 5 2019-01-01 132.762
## 6 2019-01-01 132.027
```

‘Since our data consists of daily records, let's get the average per day so we have more compactdata to work with

```
forecast <- aggregate(Sales ~ Date , sales , FUN="mean")
head(forecast)
```

```
##         Date    Sales
## 1 2019-01-01 395.4318
## 2 2019-01-02 243.1879
## 3 2019-01-03 259.7661
## 4 2019-01-04 270.6148
## 5 2019-01-05 294.7236
## 6 2019-01-06 401.5783
```

```
# Converting data frame into a tibble time (tbl_time) tbl_time have a time index that contains informat
pred= tbl_time(forecast, Date)
class(pred)
```

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

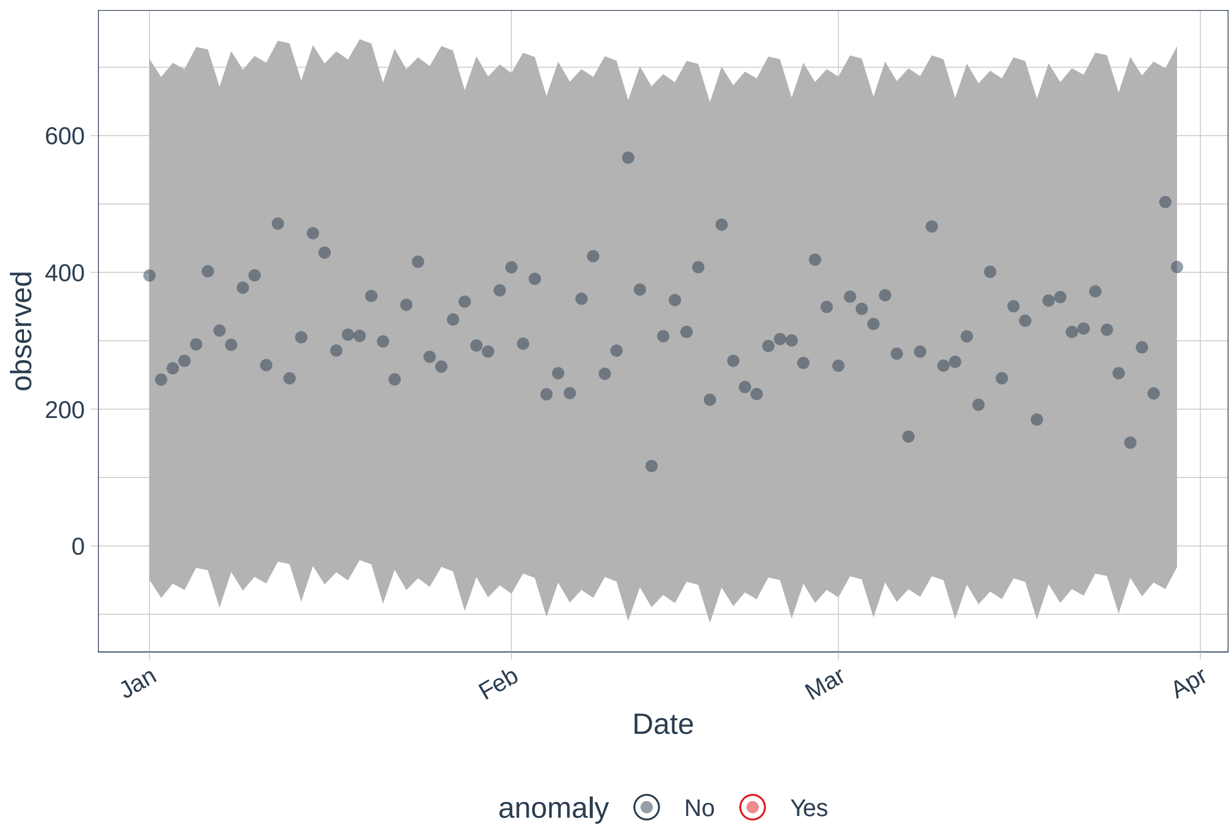
We now use the following functions to detect and visualize anomalies;

```
pred %>%
    time_decompose(Sales) %>%
    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
```



Confirming that there aren't anyanomalies

```r
skew <- sum(as.numeric(sales$Class))/nrow(sales)
sprintf('Percentage of fraudulent transactions %f', skew*100)
```

```
## [1] "Percentage of fraudulent transactions 0.000000"
```

Conclusion

There were no anomalies detected in the data.