ANOMALY DETECTION

Objective

To check whether there are any anomalies in the given sales dataset.

Context

The dataset contains sales data collected over a period of four months in 2019.The task is to detect if there are any anomalies in the sales totals

Loading libraries

```
library(tidyverse)
## -- Attaching packages ---------------------------------------
tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
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 </>
```

##Loading and Previewing data

```
sales_df <- read.csv("http://bit.ly/CarreFourSalesDataset")
```

First 6 records

```
rmarkdown::paged_table(head(sales_df, n=5))
```

Last 6 records

```
rmarkdown::paged_table(tail(sales_df, n=5))
```

Dataset Dimension

```
#The data has 1000 records and 2 features
dim(sales_df)
## [1] 1000    2
```

Data types

```
sapply(sales_df,class)
##        Date       Sales
```

```
## "character"   "numeric"
sales_df$Date <- as.Date(sales_df$Date, "%m/%d/%y")
sapply(sales_df,class)
##      Date     Sales
##    "Date" "numeric"
```

Column names

```
colnames(sales_df)
## [1] "Date"  "Sales"
```

Data Cleaning

Duplicates

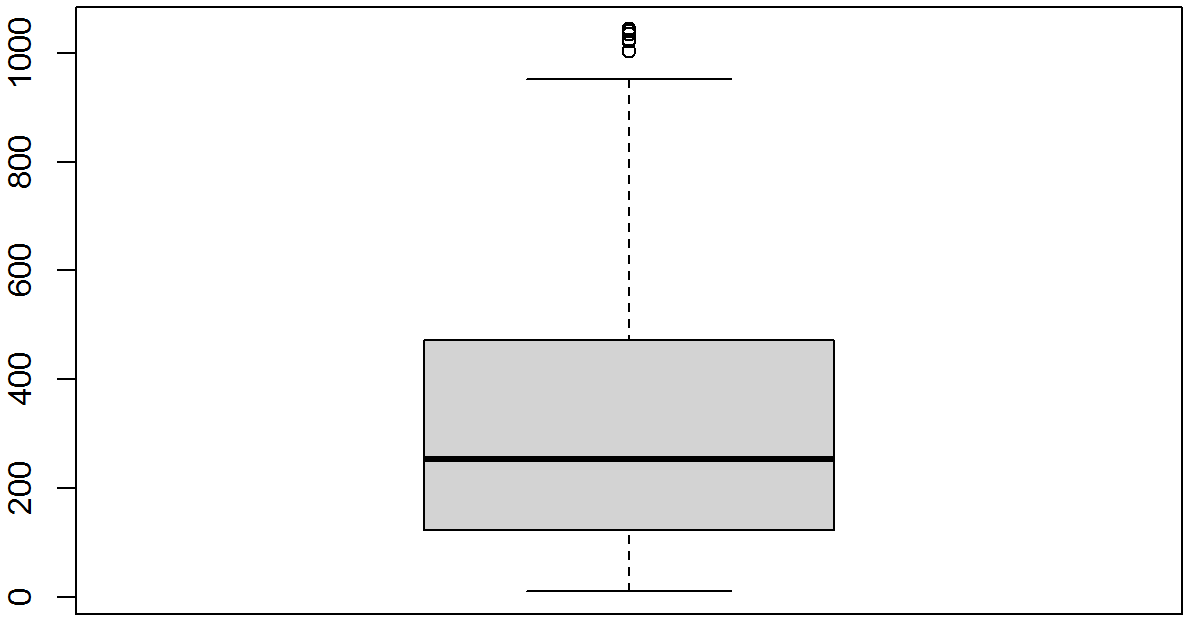
```
#Checking for duplicated records
sum(duplicated(sales_df))
## [1] 0
#The dataset has no duplicates
```

Missing Values

```
#Checking for missing values
colSums(is.na(sales_df))
##  Date Sales
##     0     0
#The dataset has no missing values
```

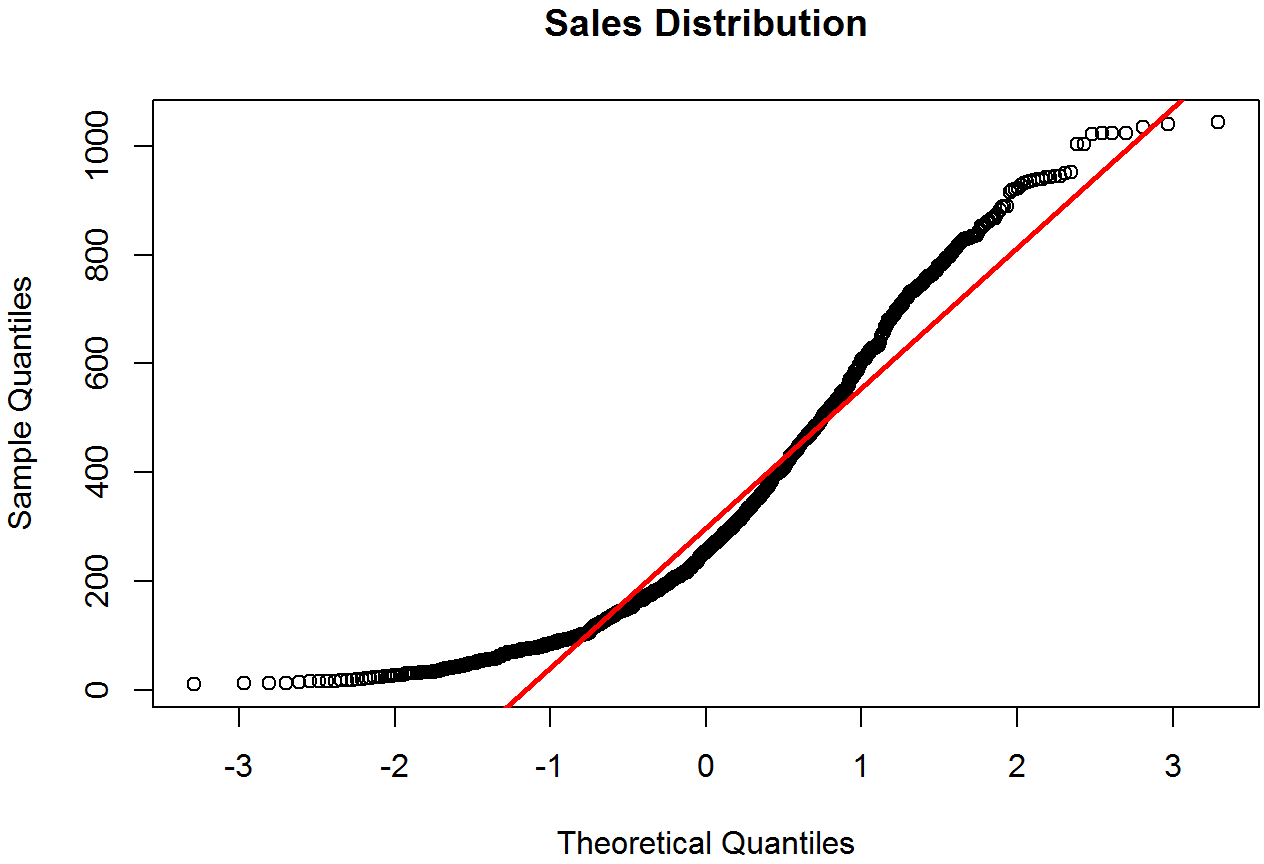
Outliers

```
#Checking for outliers
boxplot(sales_df$Sales)
```



Distribution

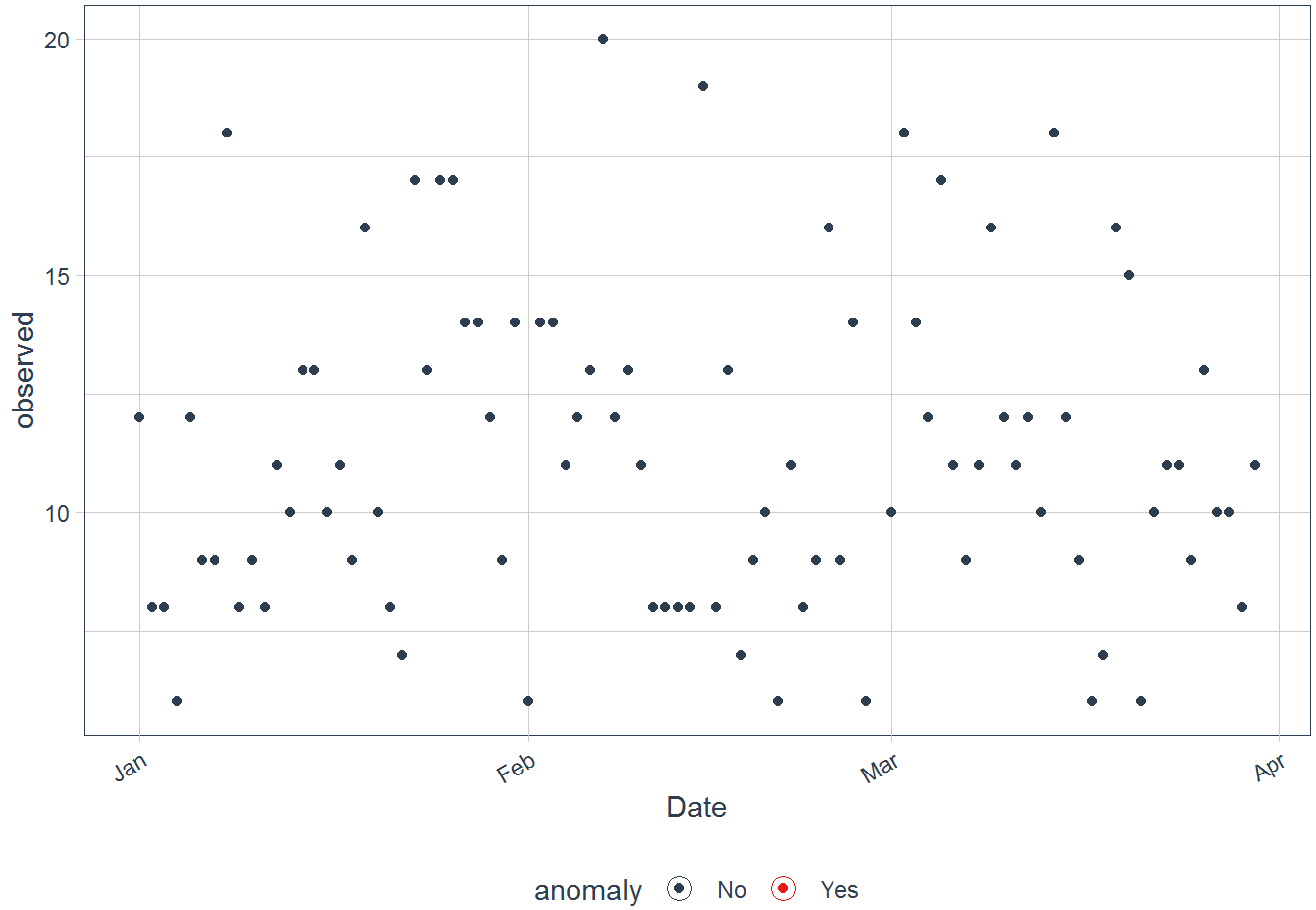
```
#Checking the distribution of the sales column
qqnorm(sales_df$Sales,main = "Sales Distribution")
qqline(sales_df$Sales, lwd=2.5,col="red")
```




```
#The sales column datapoint are normaly distributed
sales_df <- sales_df %>%
  group_by(Date) %>%
  tally()
colnames(sales_df) <- c('Date','Total_Sales')
head(sales_df)
## # A tibble: 6 x 2
##   Date       Total_Sales
##   <date>           <int>
## 1 2020-01-01          12
## 2 2020-01-02           8
## 3 2020-01-03           8
## 4 2020-01-04           6
## 5 2020-01-05          12
## 6 2020-01-06           9
```

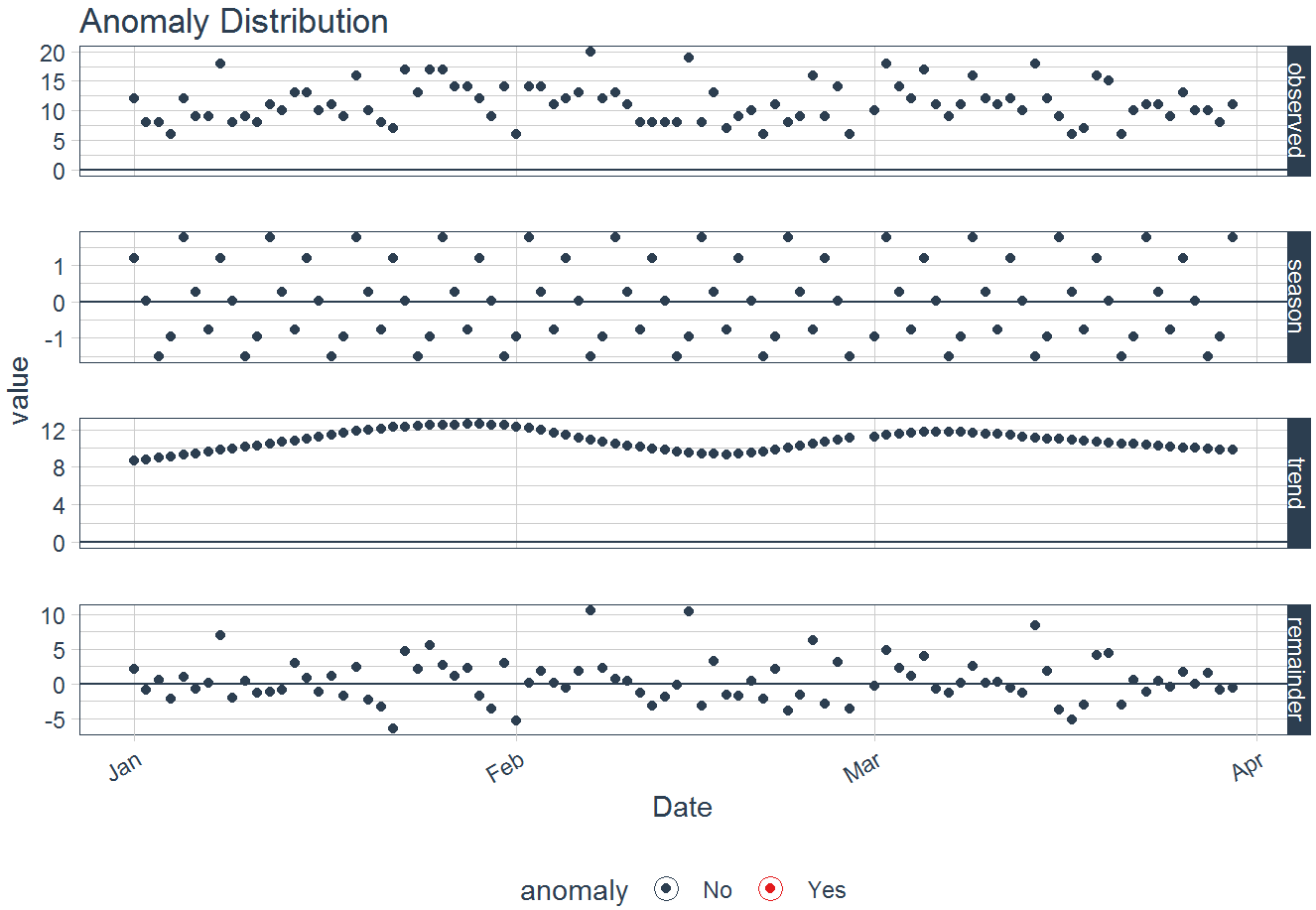
Detecting Anomalies

```
anomalized<-sales_df %>%
    time_decompose(Total_Sales) %>%
    anomalize(remainder) %>%
    time_recompose() %>%
    plot_anomalies(ncol = 3, alpha_dots = 2.5)
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date
## frequency = 7 days
## trend = 30 days
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
anomalized
```




```
anomalized<-sales_df %>%
    time_decompose(Total_Sales, merge = TRUE) %>%
    anomalize(remainder) %>%
    time_recompose() %>%
    plot_anomaly_decomposition() +
  ggtitle("Anomaly Distribution")
## Converting from tbl_df to tbl_time.
## Auto-index message: index = Date
```

```
## frequency = 7 days
## trend = 30 days
anomalized
```




```

**Summary:**
There are no anomalies detected
```