Cybersecurity

Alan Gabriel Paredes Cetina
00279105
Universidad Anáhuac Mayab
Dr. Alberto Muñoz Ubando

Final Project

November 27th, 2019

Resume

A very important concept in Machine Learning in Time Series Analysis. This concept plays a crucial role in Cyber Security for Anomaly Detection. Historic data is collected, analyzed and compared with current data, to detect deviations from regular behavior. A time series as the name suggests is a series of data points with respect to time. The data points are indicators of some activity that takes place in a given period of time. One popular example of this in Cybersecurity is a time series on the number of cyber attacks in a year in order to predict the future intensity of attacks. For this project, I am doing an analysis of Time Series in R with the Anomalize package.

Tools

1. **RStudio:** is an integrated development environment (IDE) for R**.** It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

2. **Anomalize:** is a R package that enables a tidy workflow for detecting anomalies in data. The main functions are time_decompose(), anomalize(), and time_recompose(). When combined, it's quite simple to decompose time series, detect anomalies, and create bands separating the "normal" data from the anomalous data.

The Algorithm & Analysis

Getting Started

In order to realize this project, I uploaded three libraries: `anomalize`, `tidyverse`, and `conindeskr`. The first one will help me analyze time series, the second one is for speedy data processing and the third one is for downloading the bitcoin data that I will employ for this example.

```
library(anomalize)
library(tidyverse)
library(coindeskr)
```

Then I obtained the bitcoin data from 1st January, 2007. This data contains the price per date.

```
bitcoin_data <- get_historic_price(start = "2017-01-01")
```

Transform into Time Series

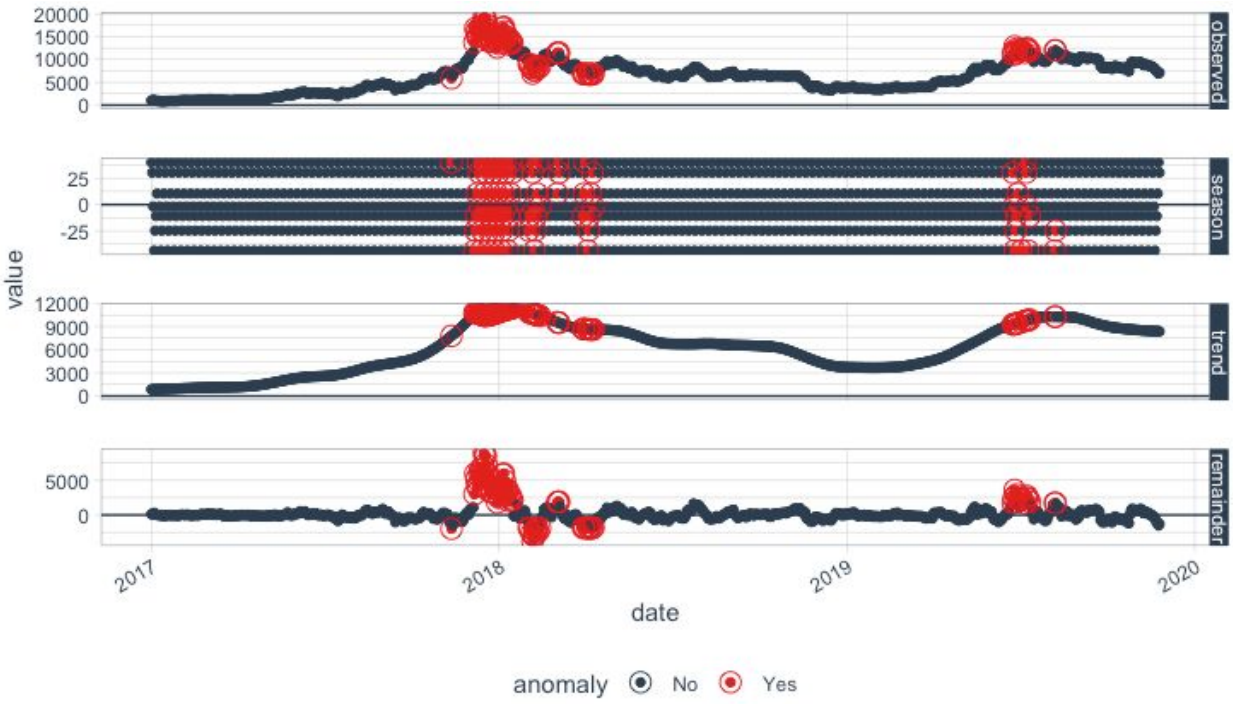
The next step now that I have the data, is to convert it to time series. In the time series conversion, I am actually converting the data to a tibble_df which the package requires.

```
bitcoin_data_ts = bitcoin_data %>% rownames_to_column() %>%
as_tibble() %>% mutate(date = as.Date(rowname)) %>%
select(-one_of('rowname'))
```

First Plot

Since it is a time series now, I should also see the seasonality and trend patterns in the data. It is important to remove them so that anomaly detection is not affected. I decomposed the series and also plotted the series. This was done with `time_decompose()` function in anomalize package. I used `stl` method which extracts seasonality.

```
bitcoin_data_ts %>% time_decompose(Price, method = "stl",
frequency = "auto", trend = "auto") %>%
anomalize(remainder, method = "gesd", alpha = 0.05,
max_anoms = 0.1) %>% plot_anomaly_decomposition()
# > Converting from tbl_df to tbl_time.
# > Auto-index message: index = date
# > frequency = 7 days
# > trend = 90.5 days
```

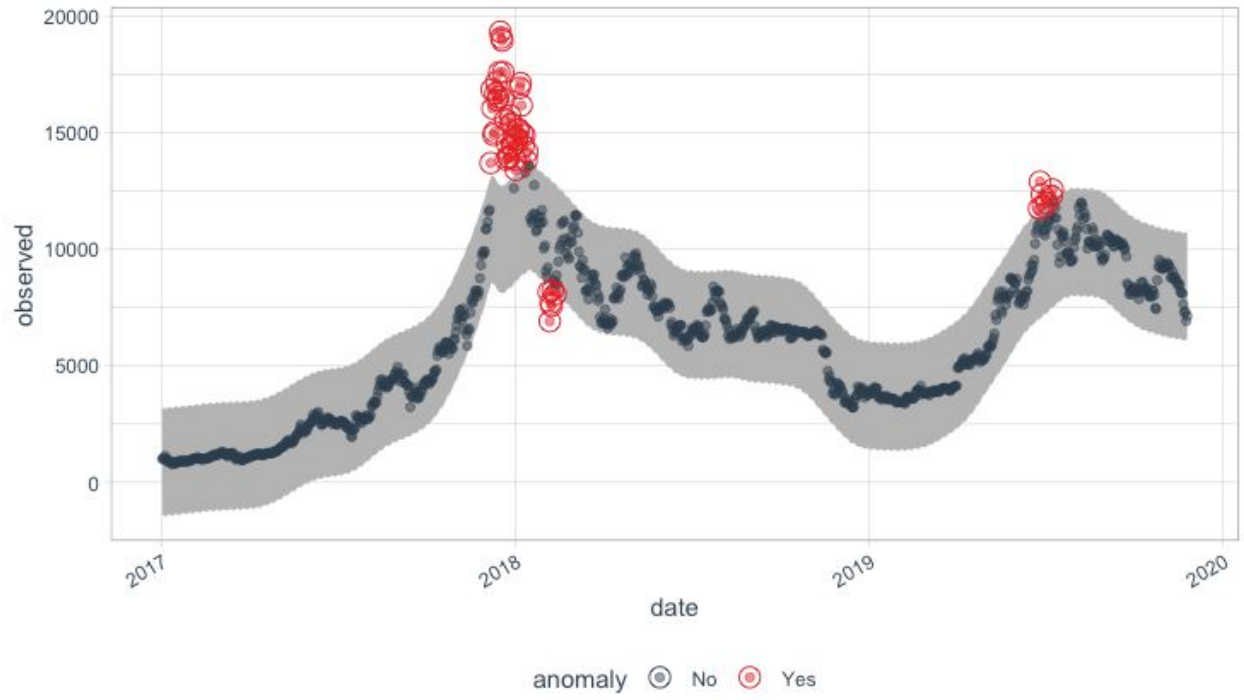


Plot 1. Anomalies Detected Within the Series.

I got some beautiful plots with the first plot being overall observed data, second being season, third being trend and the final plot analyzed for anomalies. The red points indicate anomalies according to the anomalize function. However, I am not looking for this plot. I only want the anomalies plot with trend and seasonality removed. Let's plot the data again with recomposed data.

Recomposing Data

```
bitcoin_data_ts %>% time_decompose(Price) %>%
anomalize(remainder) %>%  time_recompose() %>%
plot_anomalies(time_recomposed = TRUE, ncol = 3, alpha_dots =
0.5)
# > Converting from tbl_df to tbl_time.
# > Auto-index message: index = date
# > frequency = 7 days
# > trend = 90.5 days
```



Plot 2. Anomalies Detected During Time

This is a better plot and shows the anomalies. We all know how bitcoin prices shot up in 2018. The grey portion explains the expected trend.

Finally, to obtain the data points of anomalies and extract it out of the dataset, we only have to write the following code:

```
anomalies = bitcoin_data_ts %>% time_decompose(Price) %>%
anomalize(remainder) %>%  time_recompose() %>%
filter(anomaly == 'Yes')
```

Conclusion

We used CRAN's anomaly detection package based on factor analysis, Mahalanobis distance, Horn's parallel analysis or Principal component analysis. Hence, one can get a general idea from all such packages: anomalies are data points which do not follow the general trend or do not lie under the expected behavior of the rest of the data. The next question which is raised is the criteria for a data point to be following expected behavior. The rest of the data points are all anomalies.

References

[1] (2018). "Anomaly Detection in R". Perceptive Analytics. R-Bloggers.