Anomaly detection in Time Series using R [*]

Cesar Conejo Villalobos *Data Scientist*

Currently, a lot of processes cause information that can be gathered as time series. Predict anomalies in the observations also is crucial for determining changes in business patterns and decision making. Identifying those events in time series is usually complicated, so package like Anomalize in R brings effective solutions for identifying outliers observations. In this exercise, we use this package for detecting anomalies in the price of Tesla's share from January 2019 to March 2020.

Introduction

In this project, there is involved two fundamental concepts:

- Time series
- Anomaly detection

A time series is a set of observations x_t, each one being recorded at a specific time t. P. Brockwell (2002). More generally, a Time Series is a materialization of the stochastic process, that is, it is an observation of T random variables ordered in time. On the other hand, an anomaly it is just another way of calling an outlier is defined as *"an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism"* Hawkins (1980). Moreover, there is a lot of application of outlier analysis, but we focus on anomalies in the behavior of the data across time.

As a branch of machine learning, there are both supervised and unsupervised techniques in temporal outlier analysis. These topics are well developed by Aggarwal (2016). However, in most cases and patterns, time series are no marked with anomalies, so algorithms must be able to recognize an anomaly without being given any samples of what an anomaly is. For this reason, this document focus on one unsupervised approach.

In this way, before focusing on abnormalities: first, we focus on the normal behavior of a time series. Then, we study how to decompose a time series and study its anomalies. Finally, we apply the anomalies package in some time series for detecting abnormalities in the temporal series of Tesla shares.

What is normal?

One of the first approaches for detecting anomalies is to determine the normal behavior, represented by finding a mean μ and variance σ^2 for the normal distribution associated with the process. Then, we apply a statistical test to determine whether any data point for the same time series is normal or abnormal. A statistical test for an observation x can be:

$$|x - \mu| > 3\sigma$$

This under the assumption that given a large number of data points, 99.7% of the data points should fall within the mean plus/minus three times the standard deviation. (See figure 1).

[*] Template taken from (http://github.com/svmiller). **Corresponding author**: svmille@clemson.edu.

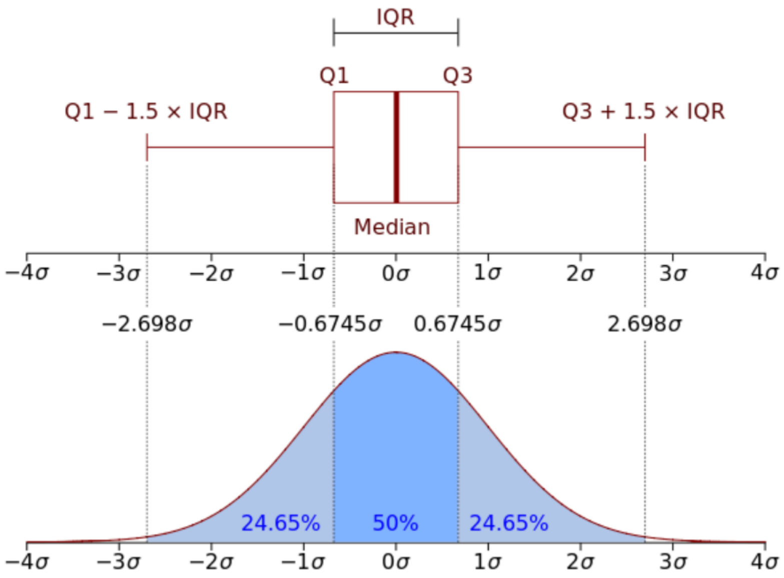


Figure 1: Normal distribution and Boxplot

Making this assumption means that if the data comes from a known distribution and if a data point is outside these bounds, it can be called an anomaly.

This strategy is easy to apply, but it has a huge problem, assuming any type of data will behave like the normal distribution is naive. We could extend this approach using other forms of parametric distributions; but again, give a large data set there are most likely many different types of behavior in the data. For this reason, we need another approach for detecting anomalies. The strategy of the anomalize package in R is based on the decomposition of the time series and detect anomalies in the remainder of the observations. As a result, we review some details about time series decomposition.

Time Series Decomposition

For time series decomposition, R. Hyndman (2008) gives us a description of the components of a time series:

- Trend: The long-term direction of the series.
- Seasonal: Pattern that repeats with a known periodicity.
- Cycle: Pattern that repeats with some regularity but with unknown and changing periodicity.
- Error/Remainder: The unpredictable component of the series.

Several techniques can be used for decomposing a time series and producing residuals. For example, some methodologies are:

1. ARIMA
2. Regression

3. Seasonal decomposition

In most series, ARIMA and Regression usually provide poor fit to the series and so it can be difficult to detect anomalies using these techniques. On the other hand, seasonal decomposition has the advantage of removing the seasonal and trend components and preserving the anomalies in the residuals. However, we must be careful using this technique because we need a smooth time series. We know that not everything is smooth and stationary, so some time is necessary to do transformations for avoiding the heteroscedasticity in the remainders. Some popular techniques are:

1. Differentiate the time series enough times to achieve stationarity.
2. Apply logarithms or power to the time series.

Once the series has been smoothed, the seasonality of the series must be determined. Again, there are various techniques that allow us to determine the frequency of the series, for example:

1. Fourier transform of signals. Takes a signal, transforms it into the frequency domain and finds frequencies that are local maximums (peaks) in the power of the Fourier transform.
2. Another technique is the autocorrelation of signals, also known as serial correlation (ACF), it is the correlation of a signal with itself at different points in time.

To sum up, there is some previous validation that is necessary to do on the time series before the detection of the anomalies. Once we resolve this step, we can use the anomalies package. For this reason, we will review some fundamental aspect of this package.

Anomalize

Business-science the creators of the packages give a review of the options of the packages. We focus on the two aspects. First, the methodologies for seasonal decomposition:

1. Seasonal decomposition of time series by Loess (STL): This can be used when there is a long term trend in the time series.
2. Twitter: Seasonal decomposition of time Series by Median. It can be used when a long term trend is less dominant that the short-term seasonal component.

Second, the anomalies in the remainders are analyzed using two strategies:

1. Inner quantile Range (IQR):Uses the 25% and 75% inner quantile range to establish the distribution. By default, the limit for being considered as an outlier is 3X. If we want the traditional 1.5X, we can use the alpha parameter for adjusting the 3X factor (Use alpha = 0.10).
2. Generalized Extrem Studentized test (GESD). Uses the GESD test, which is is an iterative test until the outlier observations have been removed. Details can be seen in the Engineering Statistics Handbook.

With these in mind, we are going to apply the anomalies package in the univariate time series of the shares prices of Tesla's actions.

Financial Application: Tesla shares prices

First, we proceed with loading the necessary libraries.

```r
library("tidyverse")
library("tibbletime")
library("anomalize")
library("lubridate")
library("quantmod")  #getsymbols
library("tbl2xts")
```

Second, we can proceed to load the price of the actions using the *getSymbols()* function. Then, we use the adjusted value of Tesla share prices in 2019/2020 for detecting anomalies in the price.

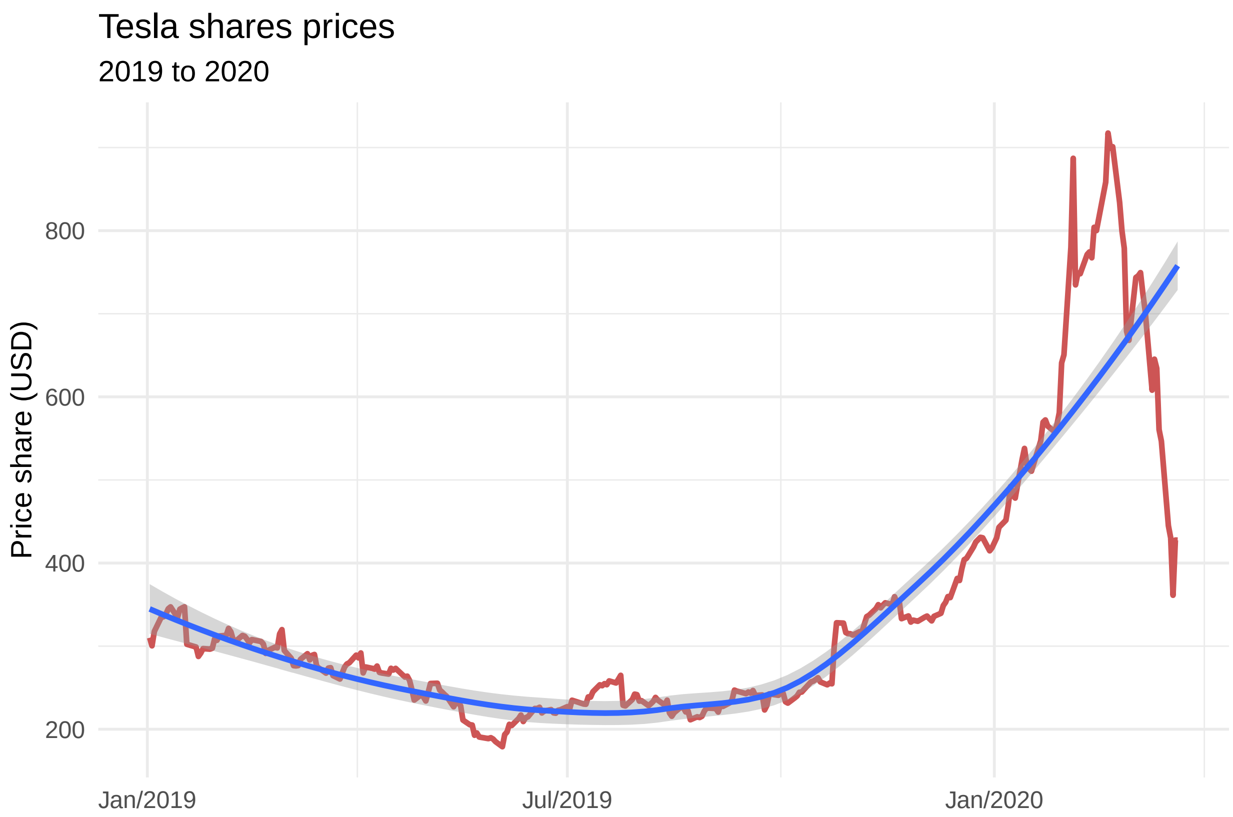
```r
getSymbols("TSLA", from = "2019-01-01", src=’yahoo’)
```

```
## [1] "TSLA"
```

```r
tbl_TSLA_Adj <- TSLA$TSLA.Adjusted %>%
                  xts_tbl()

# Index of the number of observations
tbl_TSLA_Adj$index <- as.numeric(rownames(tbl_TSLA_Adj))
```

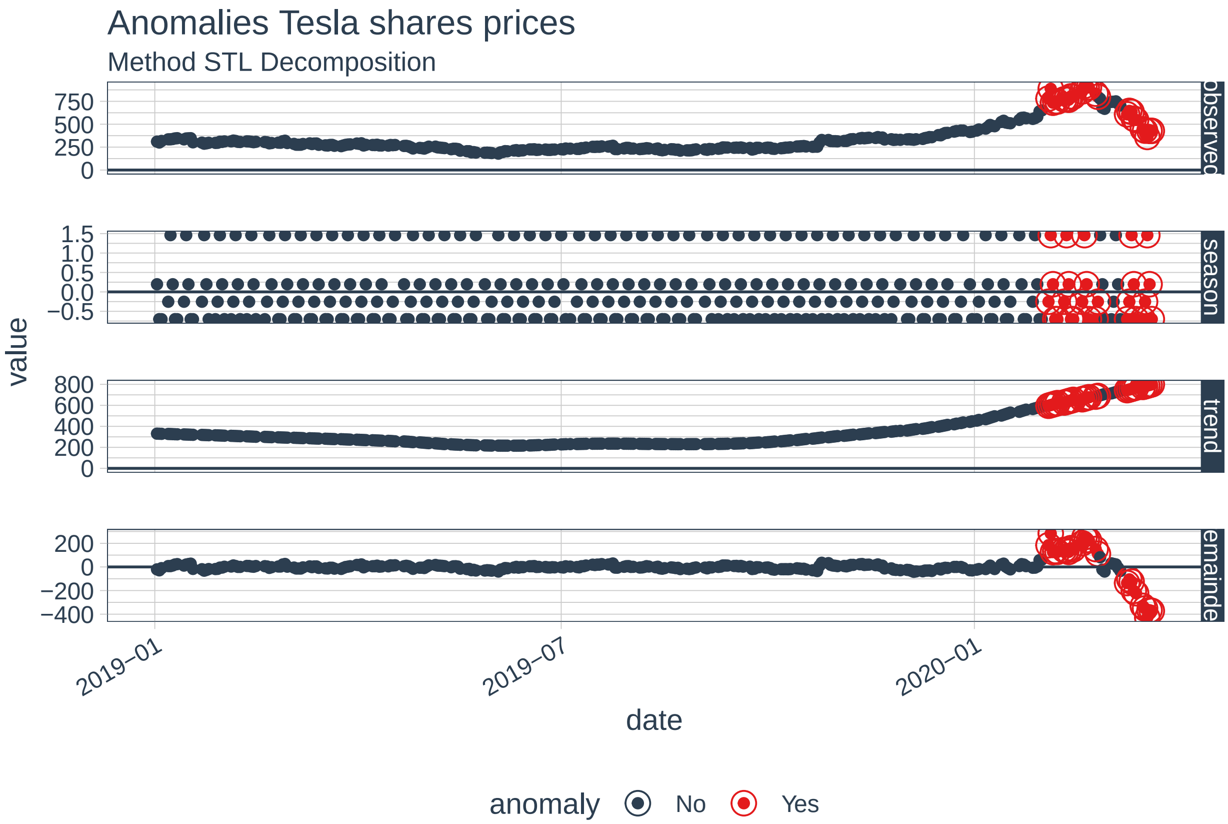
Third, we can see our time series using *ggplot()*:

```r
ggplot(data = tbl_TSLA_Adj, aes(x = date, y = TSLA.Adjusted)) +
  geom_line(color = "indianred3",
            size  = 1 ) +
  geom_smooth(method = "loess" ) +
  scale_x_date(date_labels = "%b/%Y") +
  labs(title = "Tesla shares prices",
       subtitle = "2019 to 2020",
       x = "",
       y = "Price share (USD)") +
  theme_minimal()
```



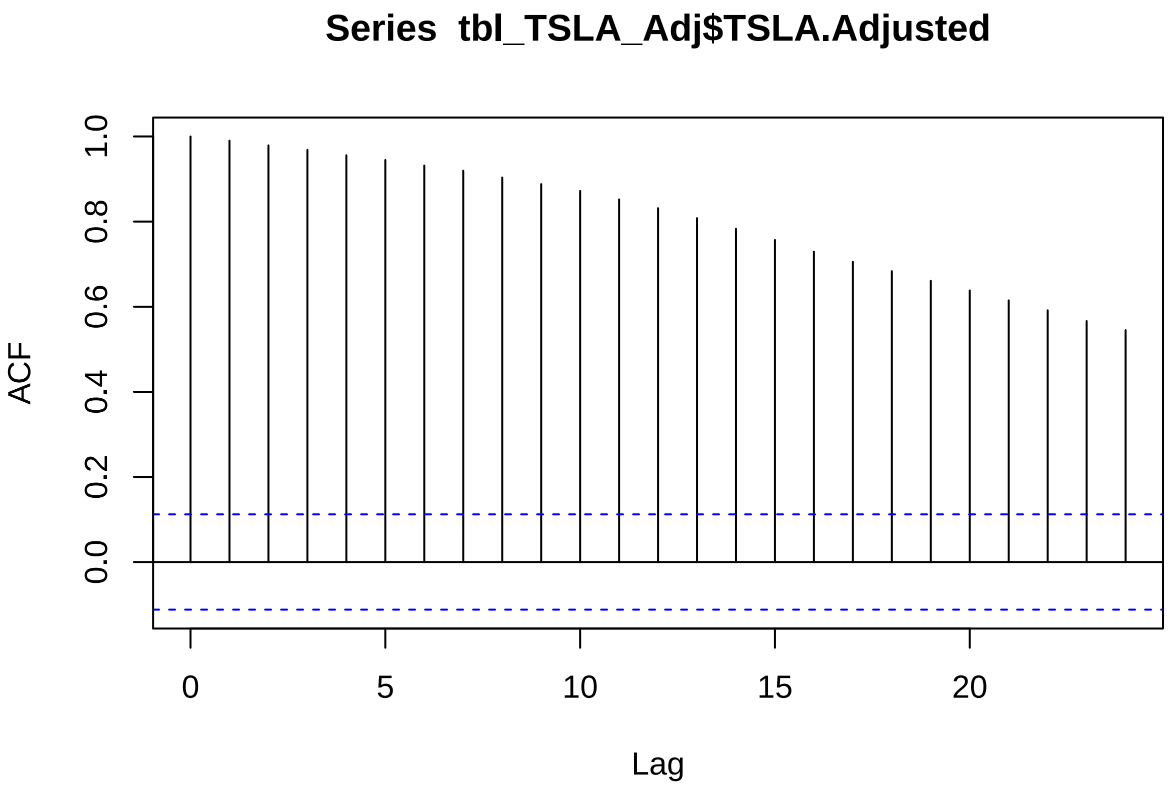
If we applied directly the functions *time_decompose()* and *anomalize()* we obtain the following result:

```
tbl_TSLA_Adj %>%
  time_decompose(TSLA.Adjusted,
                 method    = "stl",
                 frequency = "1 week") %>%
  anomalize(remainder) %>%
  plot_anomaly_decomposition() +
  labs(title = "Anomalies Tesla shares prices",
       subtitle = "Method STL Decomposition")
```



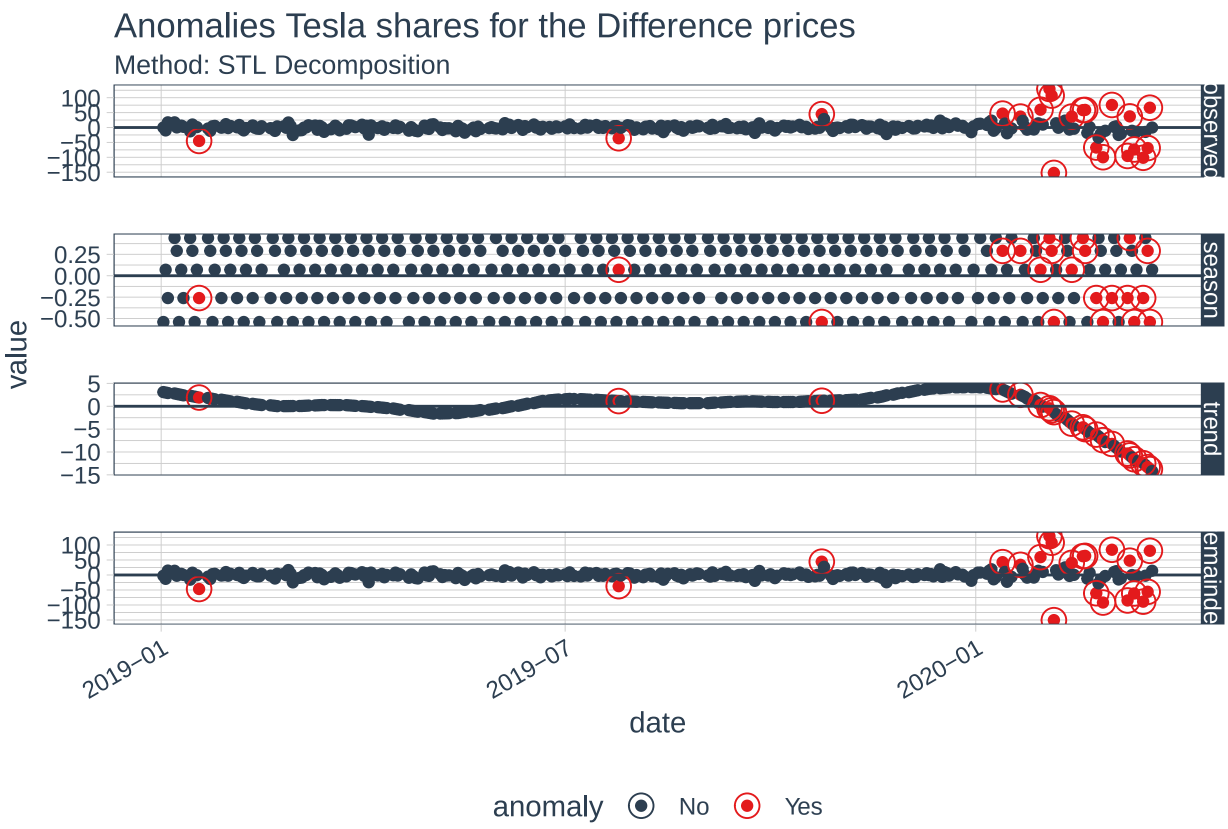
As we can confirm, this time series display considerable variation, especially since 2020. Applying *anomalize* model does not seem to be reasonable, because all the last observations will be considered an outlier. The *acf()* confirms our suspects showing all the values are significantly far from zero.

```
acf(tbl_TSLA_Adj$TSLA.Adjusted)
```



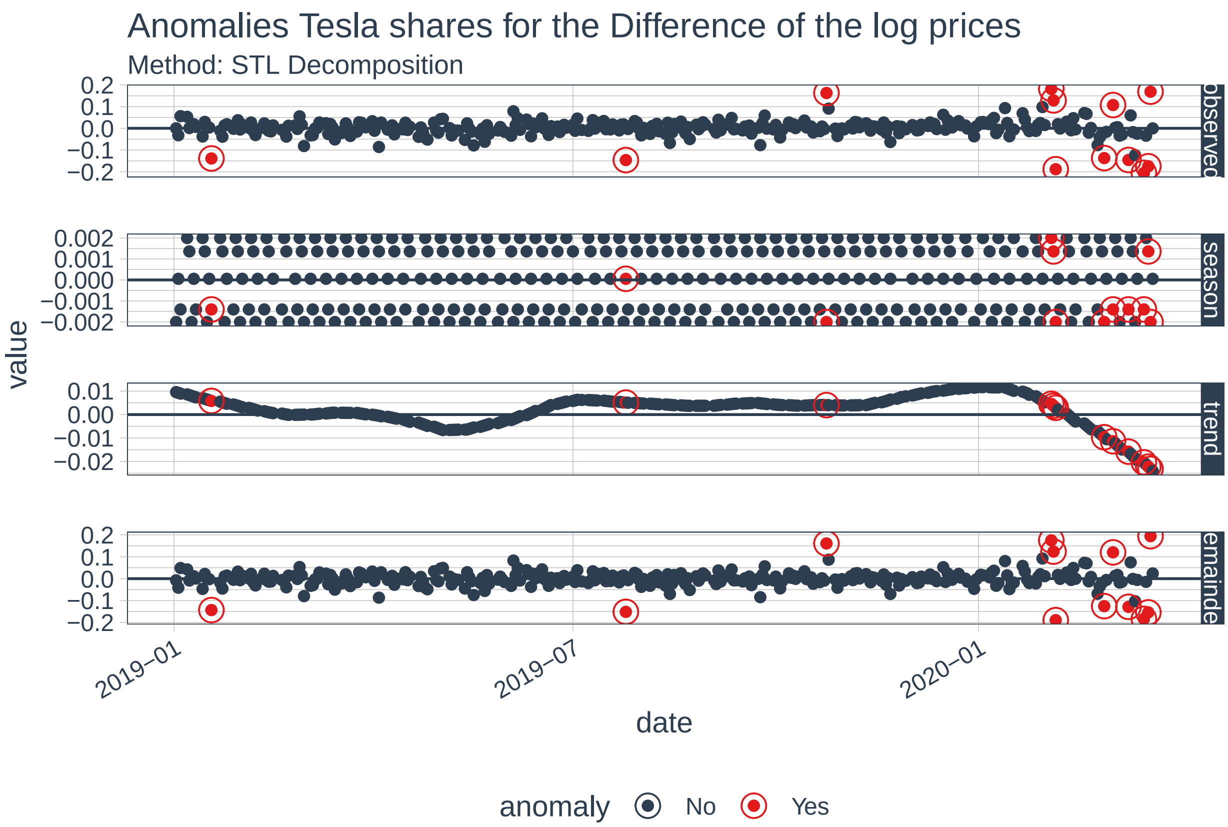
As a result, we can add a fourth step: We can do transformations and differences in the time series. In this case, we compute the first difference in the series and apply the functions *time_decompose()* and *anomalize()*.

```
# First difference
tbl_TSLA_Adj <- tbl_TSLA_Adj %>%
  mutate( TSLA.Adjusted.Diff = c(0,diff(TSLA.Adjusted)))

#Plot
tbl_TSLA_Adj %>%
  time_decompose(TSLA.Adjusted.Diff,
                 method    = "stl",
                 frequency = "1 week") %>%
  anomalize(remainder) %>%
  plot_anomaly_decomposition() +
  labs(title =  "Anomalies Tesla shares for the Difference prices",
       subtitle = "Method: STL Decomposition")
```



This graph reveals some important information. In fact, it detects three anomalies during 2019 that the first graph does not detect. But, we continue with the problem of the considerable number of outliers at the end of the series. Also, we can see some more variation at the end of the time series. It is a signal of Heteroscedasticity that we can solve using logarithms.

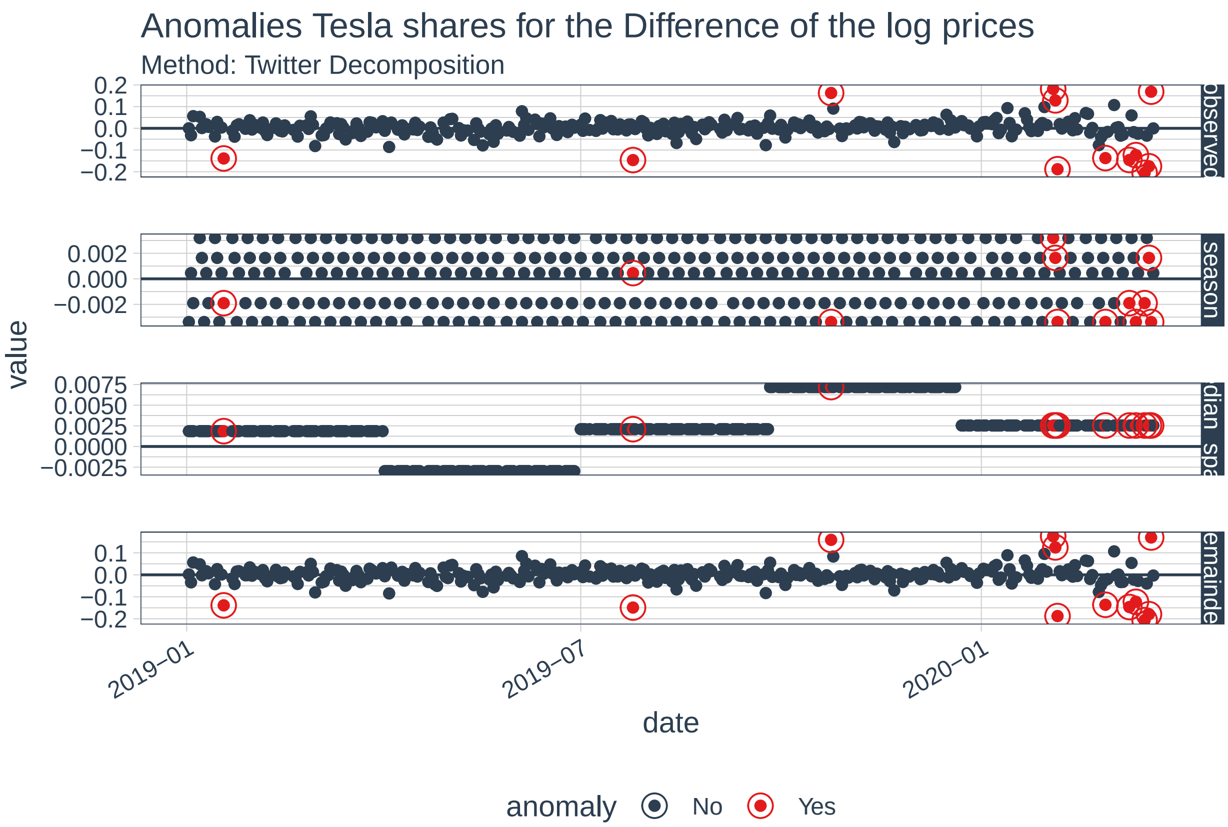
```
# Difference for logarithms for Tesla shares prices
tbl_TSLA_Adj <- tbl_TSLA_Adj %>%
  mutate( TSLA.Adjusted.Diff.Log = c(0,diff(log(TSLA.Adjusted))))

#Plot
tbl_TSLA_Adj %>%
  time_decompose(TSLA.Adjusted.Diff.Log,
                 method    = "stl",
                 frequency = "1 week") %>%
  anomalize(remainder) %>%
  plot_anomaly_decomposition() +
  labs(title =  "Anomalies Tesla shares for the Difference of the log prices",
       subtitle = "Method: STL Decomposition")
```



The Loess algorithm usually detects the trend and we can review that the three anomalies of 2019 remain and the number of outliers at the end of the series are fewer than previous graphs. This decomposition of the difference of log series also improves the identification of the change in the series. In 2020, Tesla shares price increases significantly, so the algorithms must be adaptive for detecting true changes in business patterns. As a result, the rate of change has implications on the learning algorithms that an anomaly detection system should use.

In the case of the Twitter method, it performs the trend decomposition removing the median of the data rather than fitting a smoother as STL does. The media works well in a long-term trend because the smoother tends to overfit the irregularities.

```
tbl_TSLA_Adj %>%
  time_decompose(TSLA.Adjusted.Diff.Log,
                 method    = "twitter",
                 frequency = "1 week") %>%
  anomalize(remainder) %>%
  plot_anomaly_decomposition() +
  labs(title =  "Anomalies Tesla shares for the Difference of the log prices",
       subtitle = "Method: Twitter Decomposition")
```



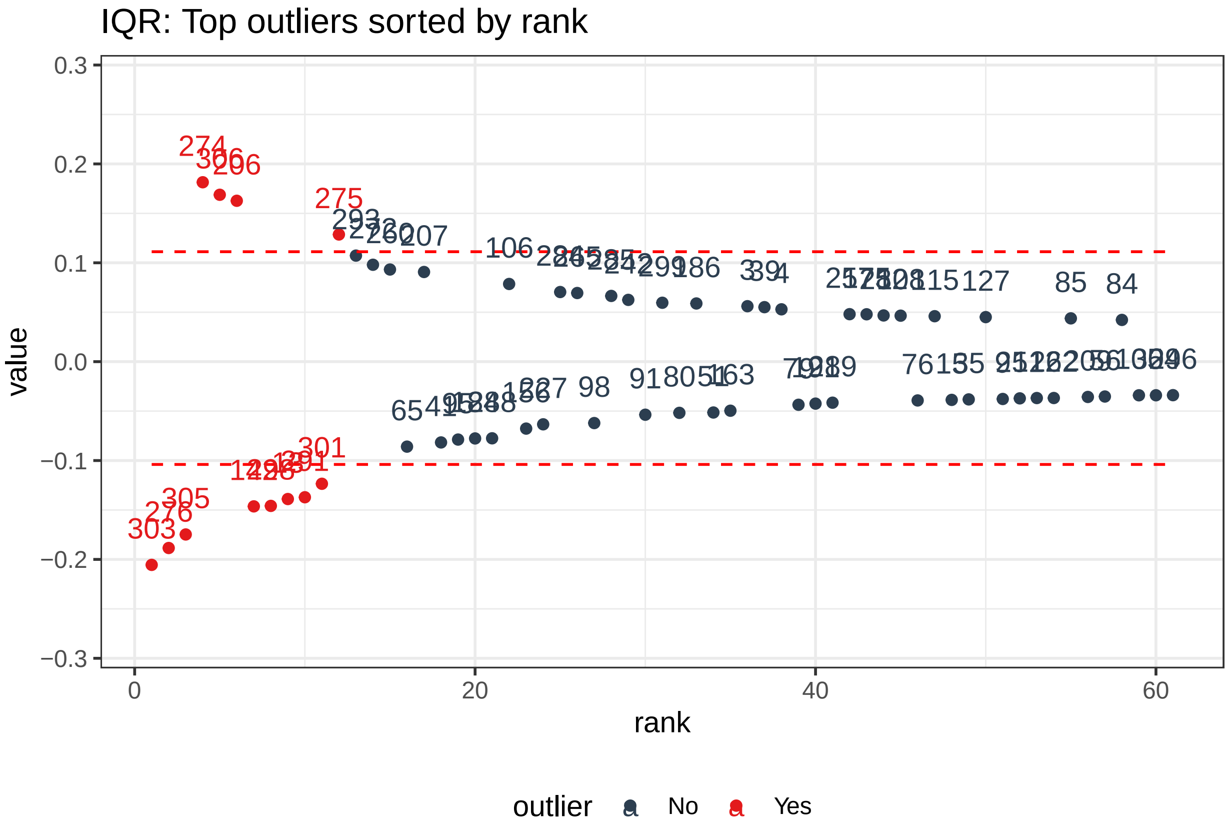
In both graphs, we can see there is no big difference between the methods of decomposition of the series and the detection of the anomalies in the Tesla shares series. On the other hand, we can compare also the methods for detecting anomalies in the remainders.

In the case of the IQR method using the default limit 3X and the parameter $\alpha = 0.05$ for consistency with GESD, we obtain the following results:

```
TSLA_iqr_outliers <- iqr(tbl_TSLA_Adj$TSLA.Adjusted.Diff.Log,
                         alpha = 0.05,
                         max_anoms = 0.2,
                         verbose = TRUE)$outlier_report
```

```
# Ploting function for anomaly observations
# Basis taken from:
# https://business-science.github.io/anomalize/articles/anomalize_methods.html

ggsetup <- function(data) {

  #Scale y axis
  y_axis <- max(abs(data$value)) + mean(abs(data$value))

  data %>%
    ggplot(aes(rank, value, color = outlier)) +
    geom_point() +
    geom_line(aes(y = limit_upper), color = "red", linetype = 2) +
    geom_line(aes(y = limit_lower), color = "red", linetype = 2) +
    geom_text(aes(label = index), vjust = -1.25) +
    theme_bw() +
    scale_color_manual(values = c("No" = "#2c3e50", "Yes" = "#e31a1c")) +
    ylim(- y_axis, y_axis) +
    theme(legend.position = "bottom")
}
```

```
TSLA_iqr_outliers %>%
  ggsetup() +
  ggtitle("IQR: Top outliers sorted by rank")
```



```
# View the date of the outlier
report_outlier_iqr <-  tbl_TSLA_Adj %>% inner_join(TSLA_iqr_outliers)


report_outlier_iqr %>%
  select(date,index, TSLA.Adjusted, TSLA.Adjusted.Diff.Log, rank, outlier) %>%
  filter(outlier == "Yes") %>%
  arrange(rank) %>%
  knitr::kable()
```

date	index	TSLA.Adjusted	TSLA.Adjusted.Diff.Log	rank	outlier
2020-03-16	303	445.07	-0.2055223	1	Yes
2020-02-05	276	734.70	-0.1884504	2	Yes
2020-03-18	305	361.22	-0.1747631	3	Yes
2020-02-03	274	780.00	0.1814450	4	Yes
2020-03-19	306	427.64	0.1687946	5	Yes
2019-10-24	206	299.68	0.1627074	6	Yes
2019-07-25	142	228.82	-0.1463412	7	Yes
2020-03-09	298	608.00	-0.1458645	8	Yes
2019-01-18	13	302.26	-0.1389301	9	Yes
2020-02-27	291	679.00	-0.1371331	10	Yes
2020-03-12	301	560.55	-0.1234932	11	Yes
2020-02-04	275	887.06	0.1286187	12	Yes

In conclusion, there are 12 observations that can be considered as outliers. These events can be gathered in the following groups:

1. March 20.
2. February 20.
3. October 19.
4. July 19.
5. January 19.

The purpose of detecting anomalies is trying to detect events that provoke the anomalies. In this case, we have the following explanations:

1. March 20: The descending of the Tesla share is provoked by the coronavirus outbreak and an abrupt drop in oil prices. Reference: cnbc.com

2. February 19: Tesla's quarterly earnings report and the opinion of influence investors provoke the rise in the price of Tesla's actions. Reference: cnbc.com

3. October 19: Tesla announced results that are better than the expected in the third quarter of 2019. Reference: nasdaq.com

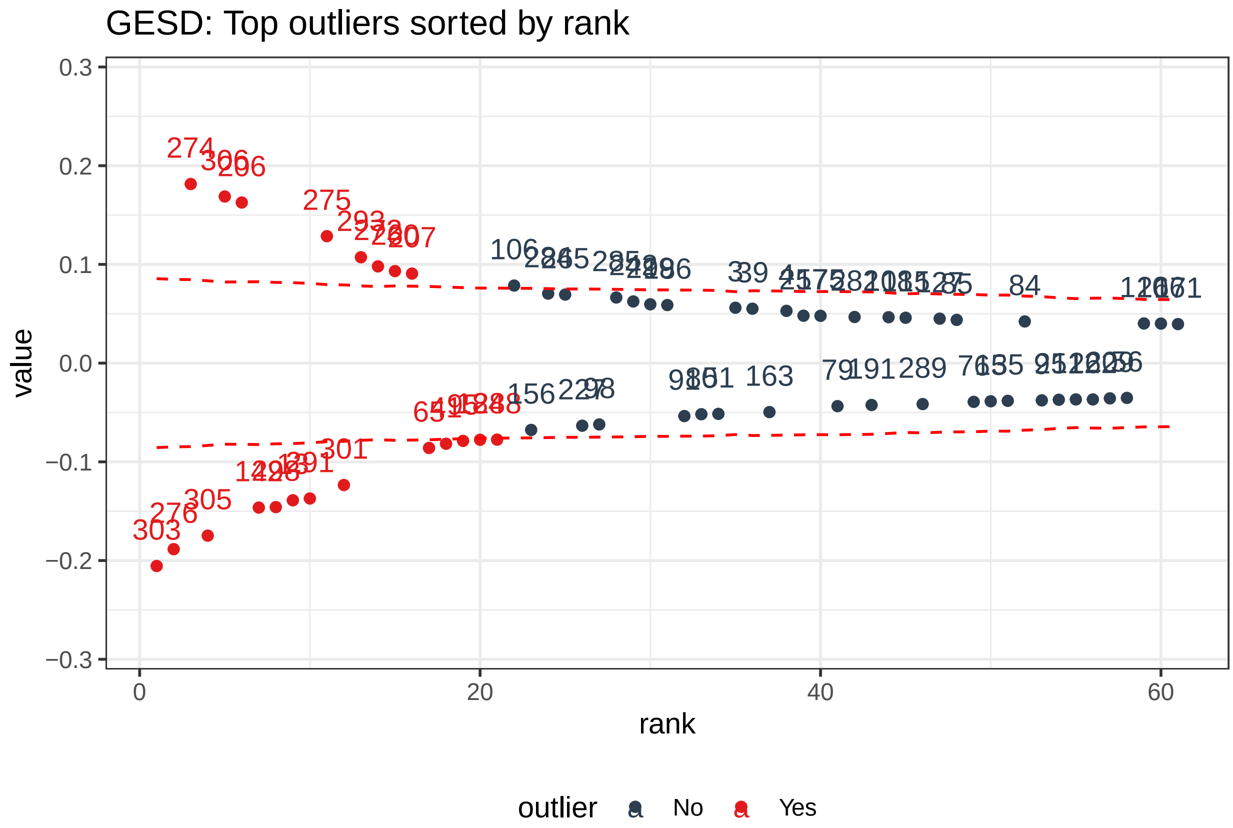
4. July 19: The fall down of the prices is because Tesla reported expected losses. It also announced the departure of one of the cofounders of the company. Reference: cnbc.com

5. January 19: Tesla cut 7 percent of its full-time jobs and faced some problems in the production of Model 3 sedans. Reference: cnbc.com

In the case of GESD, the number of outliers detected in the same series is 21, showing the advantage of this algorithm. GESD analyzes the data as anomalies are excluded. As a result, it is less resistant to high leverage points.

```r
TSLA_gesd_outliers <- gesd(tbl_TSLA_Adj$TSLA.Adjusted.Diff.Log,
                          alpha = 0.05,
                          max_anoms = 0.2,
                          verbose = TRUE)$outlier_report
```

```
TSLA_gesd_outliers %>%
  ggsetup() +
  ggtitle("GESD: Top outliers sorted by rank")
```



In the following table, we see the outliers for GESD method. We observe the presence of the same events. But, GESD adds some other dates that IQR does not identify. For example:

1. January 20.
2. September 19.
3. May 19.
4. April 19.
5. March 19.

```
# View the date of the outlier
report_outlier_gesd <-  tbl_TSLA_Adj %>% inner_join(TSLA_gesd_outliers)


report_outlier_gesd %>%
  select(date,index, TSLA.Adjusted, TSLA.Adjusted.Diff.Log, rank, outlier) %>%
  filter(outlier == "Yes") %>%
  arrange(rank) %>%
  knitr::kable()
```

date	index	TSLA.Adjusted	TSLA.Adjusted.Diff.Log	rank	outlier
2020-03-16	303	445.07	-0.2055223	1	Yes
2020-02-05	276	734.70	-0.1884504	2	Yes
2020-02-03	274	780.00	0.1814450	3	Yes
2020-03-18	305	361.22	-0.1747631	4	Yes
2020-03-19	306	427.64	0.1687946	5	Yes
2019-10-24	206	299.68	0.1627074	6	Yes
2019-07-25	142	228.82	-0.1463412	7	Yes
2020-03-09	298	608.00	-0.1458645	8	Yes
2019-01-18	13	302.26	-0.1389301	9	Yes
2020-02-27	291	679.00	-0.1371331	10	Yes
2020-02-04	275	887.06	0.1286187	11	Yes
2020-03-12	301	560.55	-0.1234932	12	Yes
2020-03-02	293	743.62	0.1072570	13	Yes
2020-01-30	272	640.81	0.0979995	14	Yes
2020-01-13	260	524.86	0.0932071	15	Yes
2019-10-25	207	328.13	0.0906947	16	Yes
2019-04-04	65	267.78	-0.0859372	17	Yes
2019-03-01	41	294.79	-0.0816827	18	Yes
2019-05-17	95	211.03	-0.0787917	19	Yes
2019-09-24	184	223.21	-0.0776378	20	Yes
2020-02-24	288	833.79	-0.0775237	21	Yes

References

Aggarwal, C. 2016. *Outlier Analysis*. Springer.

Hawkins, D. 1980. *Identification of Outliers*. Chapman and Hall.

P. Brockwell, R. Davis. 2002. *Introduction to Time Series and Forecasting*. Springer.

R. Hyndman, A. Koehler, J Ord. 2008. *Forecasting with Exponential Smoothing*. Springer.