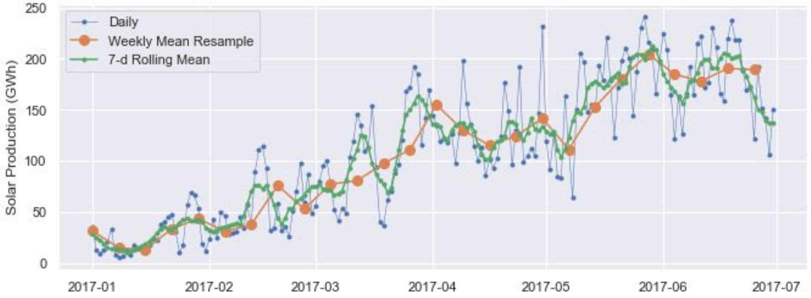
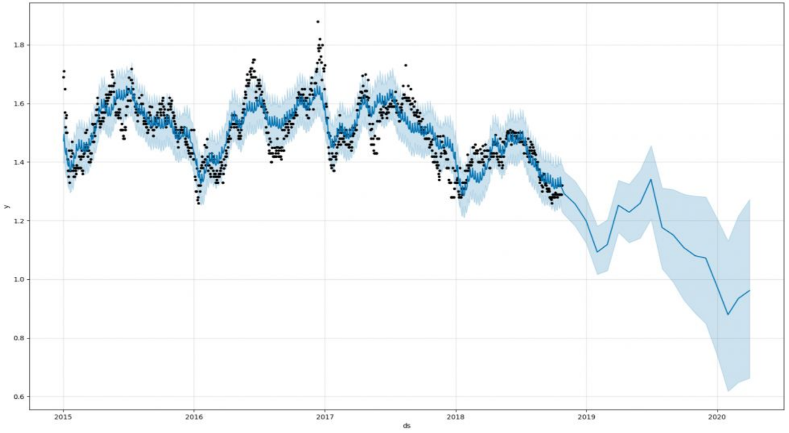
Time Series Anomaly Detection

Tokopedia Analytics Team

Oky Mauludany (Analytics Manager)
Farhan Reza Gumay (Senior Data Analyst)





Speakers

Oky Mauludany

Analytics Manager

Farhan Reza Gumay

Sr. Data Analyst

Contents

01

Intro to Time Series

Short introduction about the basic theory of Time Series

02

Anomaly Detection

Explanation about what is anomaly and some approaches to detect anomaly

03

QnA

QnA about the topics today

04

Hands-on

Hands-on in detecting Time Series Anomaly using R

1.

Introduction to Time Series

What is Time Series?

Time Series is a series of data points indexed (or listed or graph) in time order (Wikipedia)

Commonly use for:

- Monitor transaction data
- Monitor traffic
- Monitor population growth
- Closing Stock price
- Monitor employment rate
- Monitor demand and supply

Time Series Analysis: Analyzing time series data in order to extract meaningful statistic and other characteristic data

Time Series Forecasting: The use of a model to predict future values based on previously observed value

Time Series Component (1 / 3)

- Systematic parts (characterize the underlying series): level, trend, and seasonality
- Non-systematic part: noise

(1)….. = The baseline value for the series if it were a straight line.
(2)….. = The optional and often linear increasing or decreasing behavior of the series over time.
(3)….. = The optional repeating patterns or cycles of behavior over time.
(4)….. = The optional variability in the observations that cannot be explained by the model

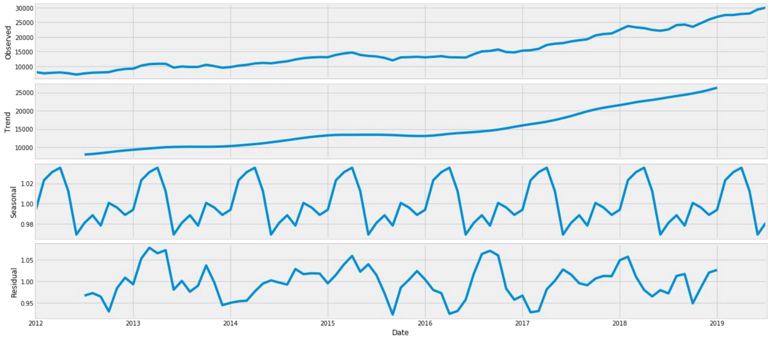
All time series have a (5)….., most have (6)….., and the (7)….. and (8)….. are optional.

Time Series Component (2 / 3)

- Systematic parts (characterize the underlying series): level, trend, and seasonality
- Non-systematic part: noise

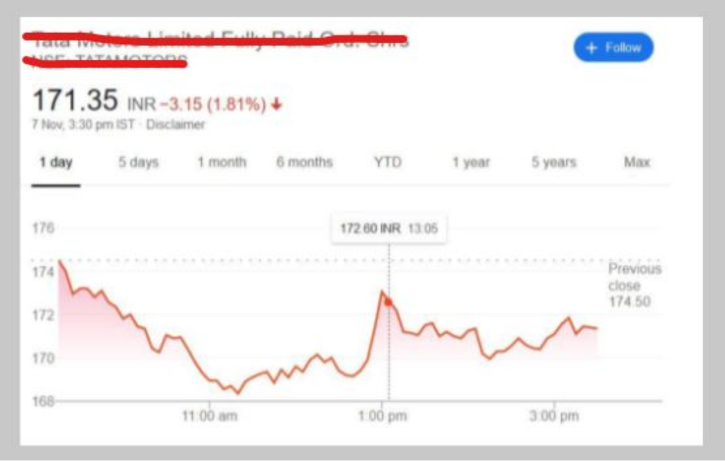
Level = The baseline value for the series if it were a straight line.
Trend = The optional and often linear increasing or decreasing behavior of the series over time.
Seasonality = The optional repeating patterns or cycles of behavior over time.
Noise = The optional variability in the observations that cannot be explained by the model

All time series have a **Level**, most have **Noise**, and the **Trend** and **Seasonality** are optional.



Time Series Component (3 / 3)

1. Trend gives us a general direction of the overall data. For example for image below, we can see that from 9AM to 11AM there is a downtrend, from 11AM to 1PM there is an uptrend, and after 1PM the trend is sideways or constant.
2. Seasonality is a regular and predictable pattern that recur at a fixed interval of time.
3. Randomness, Noise, or Residual is the random fluctuation or unpredictable changes, more like something that we can't guess.



[redacted]

171.35 INR −3.15 (1.81%) ↓

7 Nov, 3:30 pm IST · Disclaimer

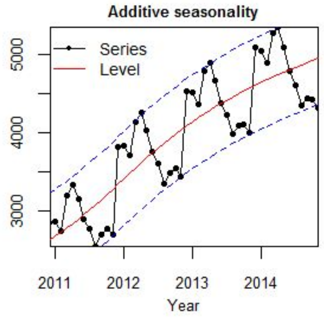
Forecasting - Additive or Multiplicative

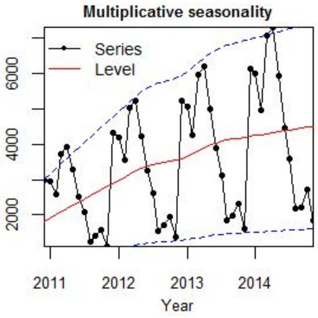
Additive (y(t) = Level + Trend + Seasonality + Noise)

- The behavior is linear where changes over time are consistently made by the same amount, like a linear trend
- The linear seasonality has the same amplitude and frequency

Multiplicative (y(t) = Level x Trend x Seasonality x Noise)

- Trend and seasonal components are multiplied and then added to the error component
- The multiplicative model has an increasing or decreasing amplitude and/or frequency over time





Forecasting Concern

1. **How much data do you have available and are you able to gather it all together?** More data is often more helpful, offering greater opportunity for exploratory data analysis, model testing and tuning, and model fidelity.

2. **What is the time horizon of predictions that is required? Short, medium or long term?** Shorter time horizons are often easier to predict with higher confidence.

3. **Can forecasts be updated frequently over time or must they be made once and remain static?** Updating forecasts as new information becomes available often results in more accurate predictions.

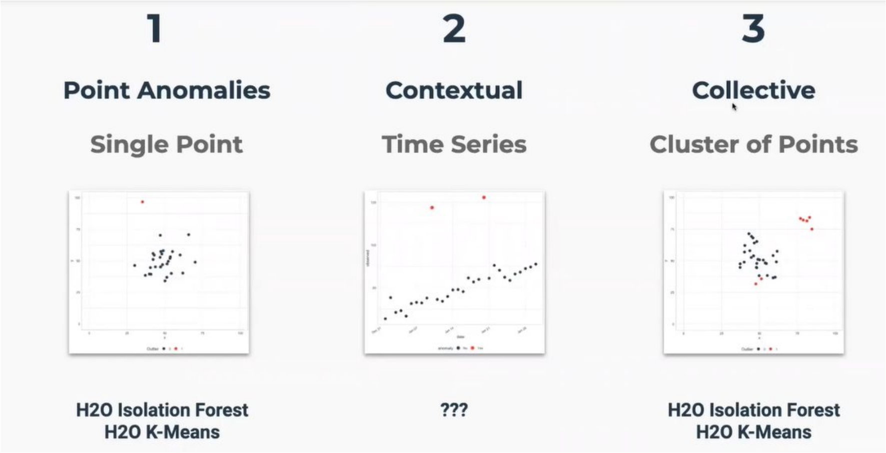
4. **At what temporal frequency are forecasts required?** Often forecasts can be made at a lower or higher frequencies, allowing you to harness down-sampling, and up-sampling of data, which in turn can offer benefits while modeling.

2.

Time Series Anomaly Detection

What is Anomaly?

- **Anomaly =** Data points that are outliers or an exceptional events.
- In small data sets, it can be identified easily with some simple analysis graphs like boxplots. But the cases will simultaneously get complicated when switched to large data sets, especially in the case of time series.
- There are 3 types of anomaly:



1	2	3
Point Anomalies	Contextual	Collective
Single Point	Time Series	Cluster of Points
H2O Isolation Forest H2O K-Means	???	H2O Isolation Forest H2O K-Means

Some Approaches of Detecting Anomaly

There are some approaches to detect anomaly:

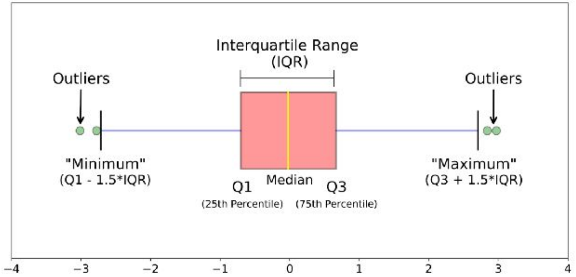
No	Approach	Method
1	Boxplot	Descriptive Statistic and Distribution Analysis
2	Simple Moving Average	
3	Moving Average + Standard Deviation	
4	Analyze the residuals (**anomalize** package in R)	Advanced Statistic by using Time Series Decomposition

Later in hands-on, we will focus more on approaches 2, 3, and 4.

Boxplot

A standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”).

Name	Description
Median (Q2/50th Percentile)	the middle value of the dataset
First Quartile (Q1/25th Percentile)	the middle number between the smallest number (not the “minimum”) and the median of the dataset.
Third Quartile (Q3/75th Percentile)	the middle value between the median and the highest value (not the “maximum”) of the dataset
Interquartile Range (IQR)	25th to the 75th percentile
Whiskers	(shown in blue)
Outliers	(shown as green circles)
“Maximum”	Q3 + 1.5*IQR
“Minimum”	Q1 -1.5*IQR



Simple Moving Average (1 / 6)

- One of the best parameters to see the today’s performance is to compare with historical performance.

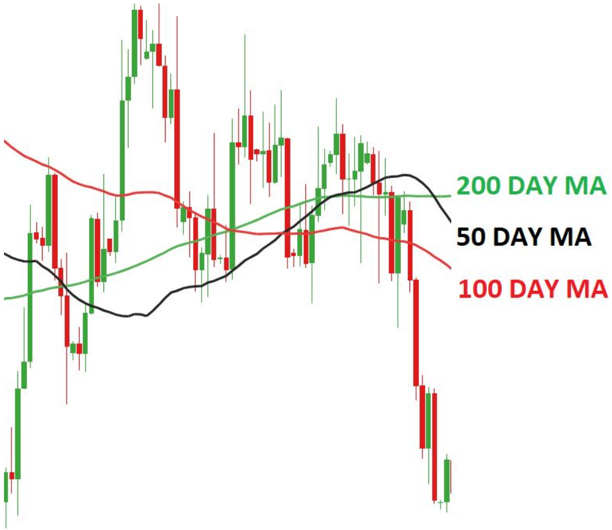
- In Time Series, generally known as **Moving Average**. If in the last 1 year our traffic is around 100k - 200k, of course we expect our traffic tomorrow not far off from that.

- But how long of past data do we have to look for into? There are some pros and cons, such as the longer the data timeframe, the more stable the data (for example 1 month data compare to yesterday data). However, too much data will impact to harder computation.

Simple Moving Average (2 / 6)

- There are some types of Moving Average (exponential, weighted, smoothed, simple, etc.), but today we are going to focus on **Simple Moving Average**.

- **Simple Moving Average** = **average of last-*x* days** data.
    - For example: **MA30** is calculated by taking the average of **last 30 days** data.



Simple Moving Average (3 / 6)

How Long to Look for Moving Average Data?

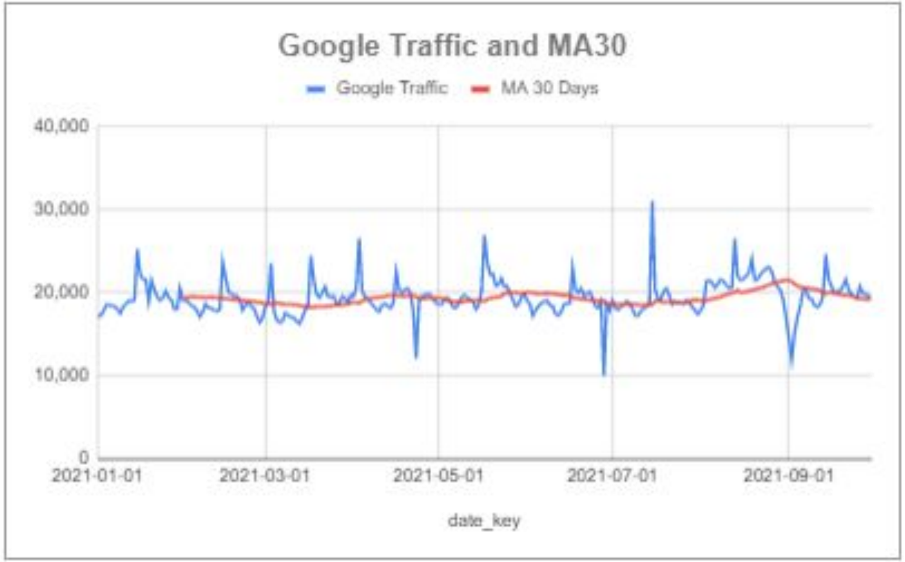
- There is no right answer. **The longer period** you look, that means you paid **more attention on major trend**.

- Some things that can be considered are :
  - How big is the data?
  - Is there any seasonality? For example if there is monthly seasonality but we only take MA7, we might not captured the seasonality trend.

Simple Moving Average (4 / 6)

What's next?

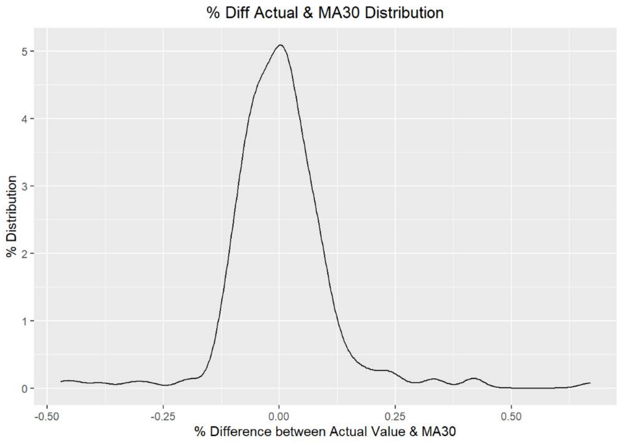
- Let say we take MA30 and have data like below. How do we decide which one is anomaly? How far the data from MA30 that we can conclude they are anomaly?



Simple Moving Average (5 / 6)

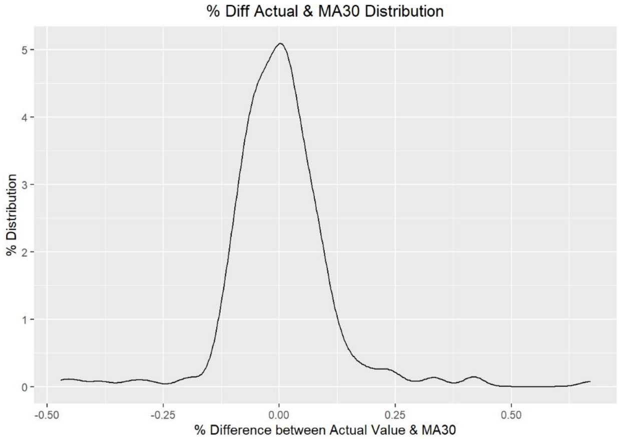
- Again, there is no right or wrong answer. If you decide that **% diff >= 25%** need to be classified as anomaly, then go with it.
  **% differences = (actual value - MA30) / MA30)*

- One of the possible approach that we can do is to look into the distribution of the %differences. Let say we have the distribution of %differences like below.



Simple Moving Average (6 / 6)

- We can see that the distribution is following Normal Distribution.
- One of the rules in Normal Distribution is **anomaly** falls outside of **3 standard deviation**.
- So we can use that rules to get how big % differences that we will classified as anomaly.



Moving Average + Standard Deviation (1 / 3)

- Another approach that we can use is to represent the differences in term of standard deviation instead of % of differences. This is very similar with previous approach, except that the differences is presented in standard deviation unit.

- So not only we take MA30, but we also calculated the **Standard Deviation** of the last 30 days data.

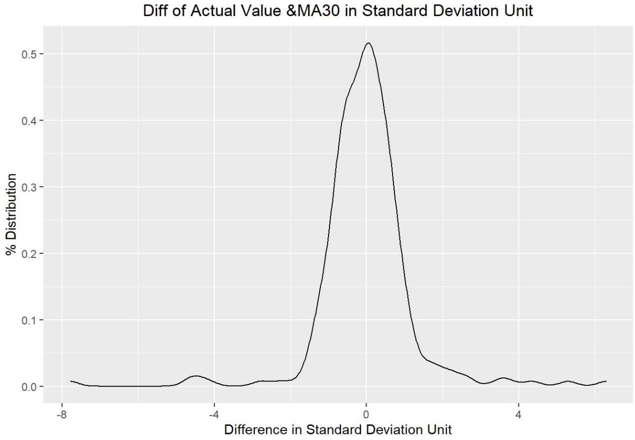
- This way we can have a **dynamic threshold**, that the threshold also changes if there is some occasions that make our data more volatile.

- So the steps will be like this :
    a. Calculate MA30
    b. Calculate Standard Deviation of Last 30 Days data
    c. Calculate the differences between actual data and MA30 in Standard Deviation unit
    (Actual Data - MA30) / SD30

Moving Average + Standard Deviation (2/3)

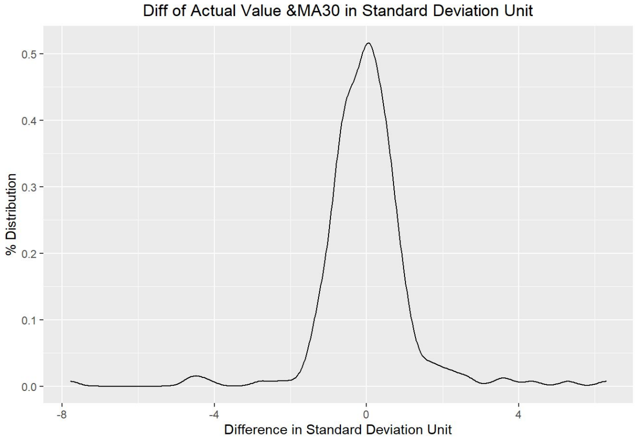
But then again the question, how big is the differences that we can classified as anomaly?

- AGAIN, there is no right and wrong answer. If in your case **difference > 1 standard deviation** is already big enough to be classified as anomaly, then go with it.
- Here, we can use the same approach as before that will see the distribution.



Moving Average + Standard Deviation (3 / 3)

- For this case, we will also use the rules of anomaly detection for Normal Distribution, that data **outside 3 standard deviation** will be classified as **anomaly**



anomalize Package - Introduction

The entire process of Anomaly Detection using ***anomalize*** takes place across 3 steps:

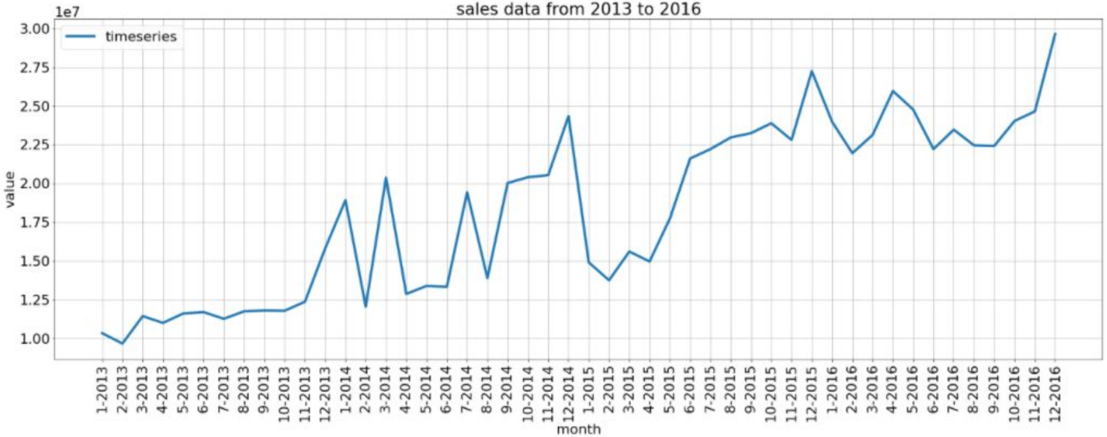
1. Decompose the time-series into the underlying variables; Trend, Seasonality, Residuals.
2. Create upper and lower residuals thresholds with some threshold value.
3. Identify the data points which are outside the thresholds as anomalies.

anomalize Package - Time Series Decomposition (1 / 7)

- Time Series Decompose using STL (Seasonal and Trend Decomposition using Loess).

- Loess is a regression technique that uses local weighted regression to fit a smooth curve through points in sequence (for our case, it is time series).

- Let's take a look step by step on how to decompose time series. We will use this image (from *Corporacion Favorita Groceries Sales Forecasting)* for example



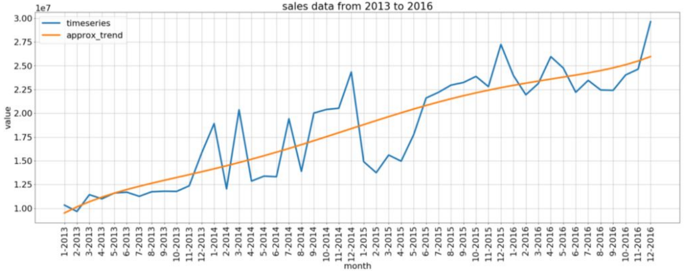
anomalize Package - Time Series Decomposition (2 / 7)

Why we need decomposition in Time Series when we can see pretty much everything in the original plot?

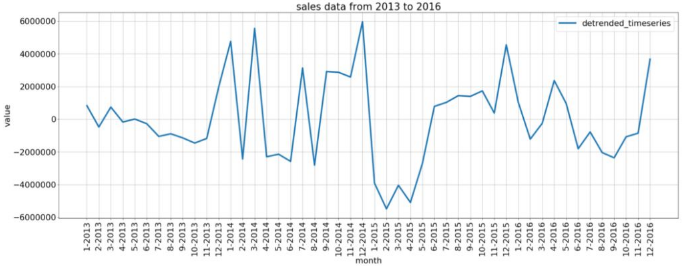
1. Teasing things apart make our observations easier. For instance, there might be spikes that are due to some drivers, but the spikes may not be visible due to seasonality. Decomposing the seasonality out will make the spikes stand out more visibly in the residual component.
2. By decomposing the time series, we can treat each component separately, then recombine them if needed.
3. For forecasting purpose, some models are assuming stationarity. This can be fulfilled using a decomposition.

anomalize Package - Time Series Decomposition (3 / 7)

1. Find the approximate trend line which fits the above time series

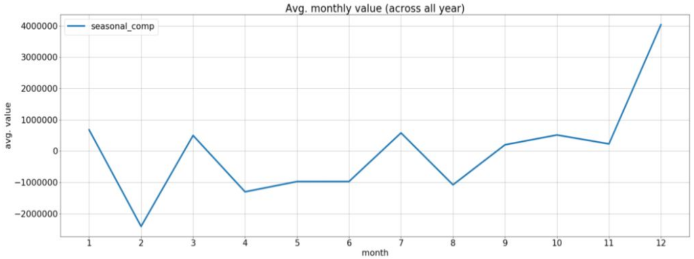


2. Find the De-trended series by subtracting time series data with approximate trend line. By this step, we remove the trend from the time series.

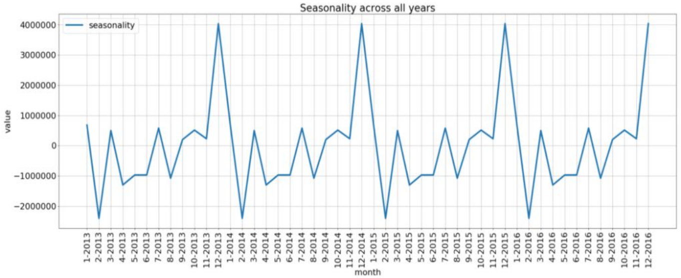


anomalize Package - Time Series Decomposition (4 / 7)

3. Find the seasonal component. For this example, it is on a month level (period=12)

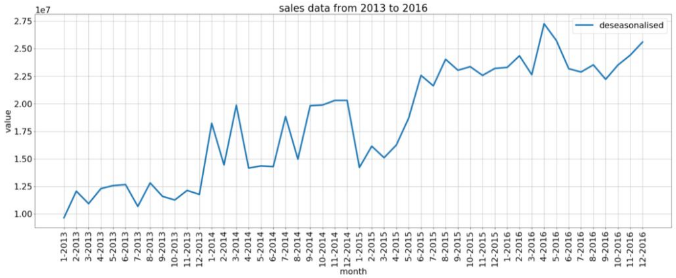


4. Populate this component across the whole data (for 48 months) *(later this will be the **seasonal value**)*

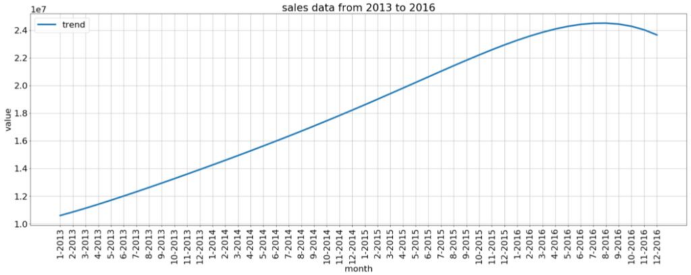


anomalize Package - Time Series Decomposition (5 / 7)

5. Find the De-seasonalised time series by subtracting time series data with seasonal data. By this step, we remove the seasonality of the time series.

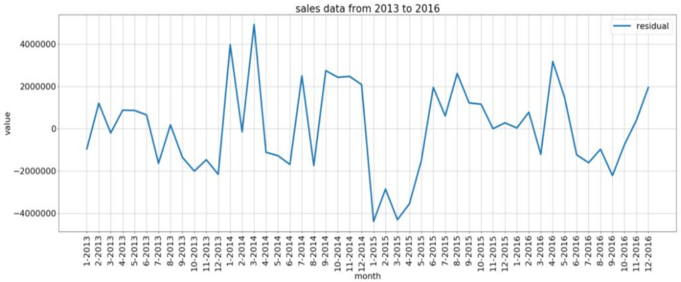


6. Find the trend by fitting the De-seasonalised data to a polynomial model. *(later this will be the **trend value**)*



anomalize Package - Time Series Decomposition (6 / 7)

7. Find the residual by subtracting time series data with seasonal and trend. *(this is the **residuals / remainders value**)*

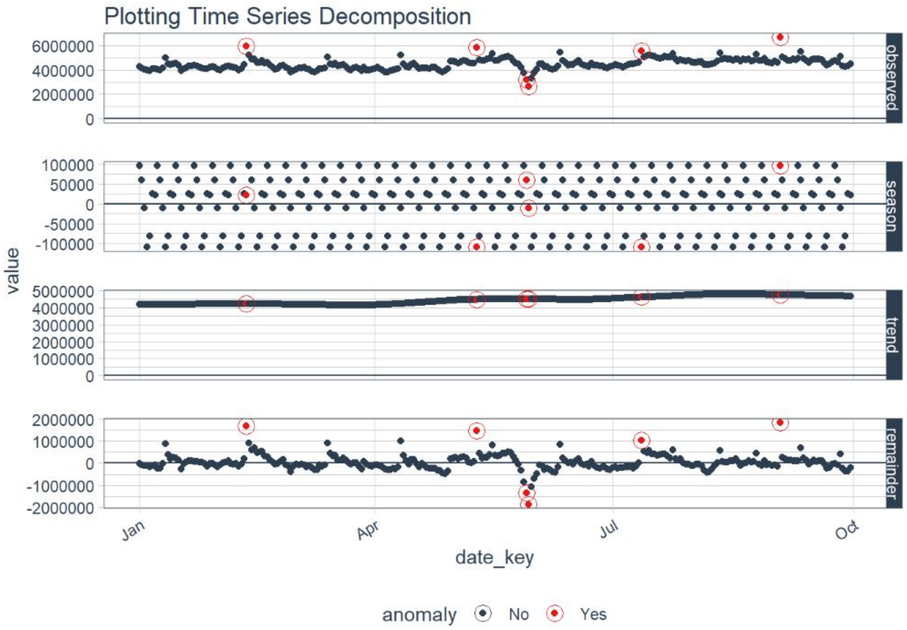


Key Concept of anomalize package : Outliers have abnormal residuals (remainders)

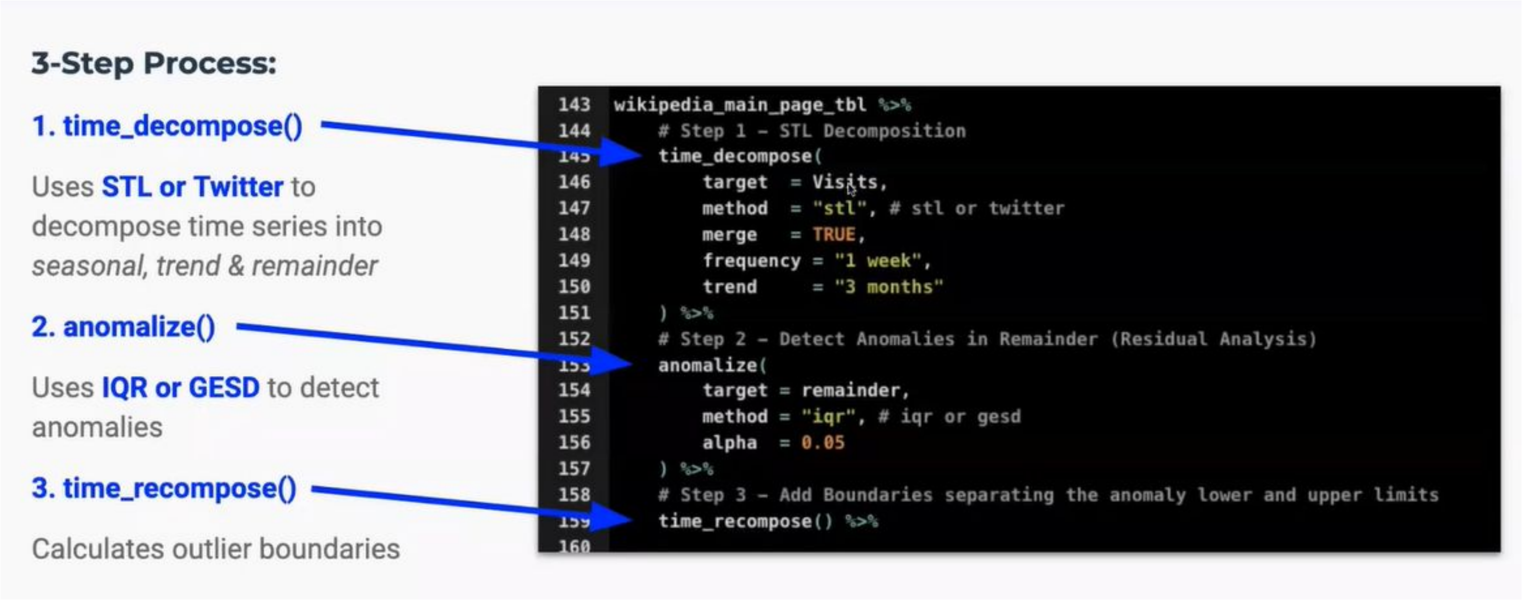
There is other Decomposition Methods that called “twitter decomposition”, we won’t discuss it further here, but the difference is only on the trend-decomposition part.

anomalize Package - Time Series Decomposition (7 / 7)

The normal data point should have uniform remainders (residuals) as we see in the 3rd picture.
So we can detect anomaly by having some threshold to detect residuals that behave beyond normal, and then classify data points outside the threshold as anomaly.



anomalize Package - Code Structure



3-Step Process:

1. time_decompose()

Uses **STL or Twitter** to decompose time series into *seasonal, trend & remainder*

2. anomalize()

Uses **IQR or GESD** to detect anomalies

3. time_recompose()

Calculates outlier boundaries

```
wikipedia_main_page_tbl %>%
    # Step 1 - STL Decomposition
    time_decompose(
        target    = Visits,
        method    = "stl", # stl or twitter
        merge     = TRUE,
        frequency = "1 week",
        trend     = "3 months"
    ) %>%
    # Step 2 - Detect Anomalies in Remainder (Residual Analysis)
    anomalize(
        target = remainder,
        method = "iqr", # iqr or gesd
        alpha  = 0.05
    ) %>%
    # Step 3 - Add Boundaries separating the anomaly lower and upper limits
    time_recompose() %>%
```

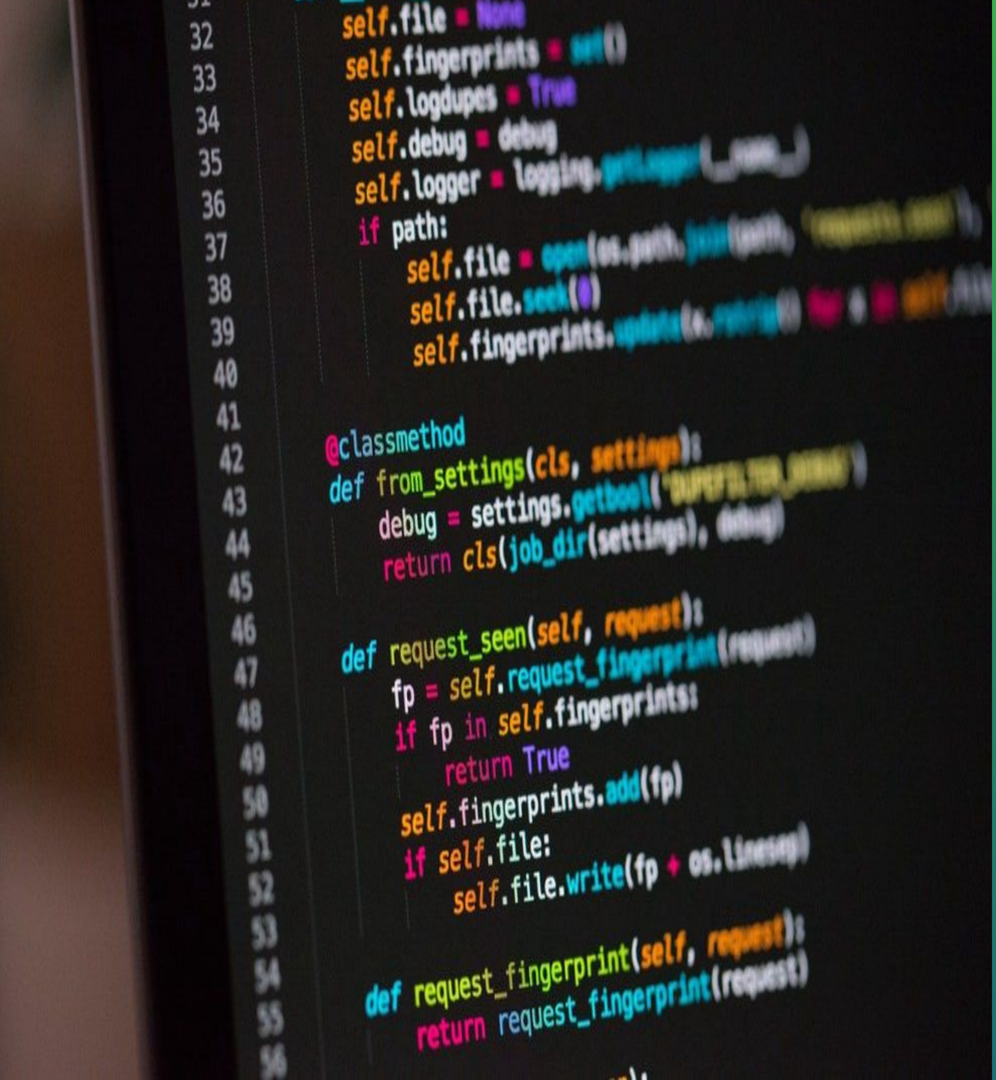
Time Series Anomaly Detection Implementation

Time series anomaly detection implementation

a. Detect traffic anomaly
b. Alert for declining sales
c. Decision for cut loss or take profit in stocks, future, or crypto
d. Preparing and cleaning the data by removing the anomaly for forecasting purpose

3.

QnA



```
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, 'requests.seen'),
            self.file.seek(0)
            self.fingerprints.update(x.rstrip() for x in self.fi

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('DUPEFILTER_DEBUG')
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```

4.

Hands-On with R

Hands-On with R

Packages needed :
Make sure you install RTools first, follow the steps below

1. For Windows : https://cran.r-project.org/bin/windows/Rtools/
2. Mac: https://github.com/rmacoslib/r-macos-rtools

Because sometimes if you don’t install RTools, the ***anomalize*** package is not working properly.

- zoo (for calculating moving average)
- tidyverse
- anomalize (package that will be used to detect anomaly)
- tibbletime (anomalize package need to work with tibble format)
- timetk (for working with time on time series data)
- tidyquant
- Rcpp

Thank you!

Do you have any questions?

da@tokopedia.com

tokopedia.com