**Shewhart Individuals Control Chart**

According to Shewart [5], a phenomenon is said to be within control, when through the use of past experience, we can predict, at least within limits, how the phenomenon may be expected to vary in the future. Control is not defined as the complete absence of variation, but simply a state where all variation is predictable. A controlled process is not necessarily good nor is an-out of control process bad. Variation can be attributed to common cause (or natural variation in a process) and special cause (or unexpected variation which cause can be assignable). Out of control points that are not random in pattern indicates the presence of special cause variation.

Control chart (also known as Shewhart charts) is a statistical process control tool which can be used to assess if an industrial process is in a state of control. [1] Given the nature of the project time series auto correlated data, we would be utilizing the Individual-X /Moving Range pair chart (XmR) as well as Time Series Model (TSM) to monitor the readings for the individual sensors.

The XmR consist of a pair of charts that enable us to monitor a process for shifts in the process that changes the mean or variance of the measured statistic. The Individual-X chart is used to analyses central location while the Moving Range chart shows the difference between consecutive readings. Together, the XmR chart is used to study system variability.

* Individual-X chart which displays the individual measured values (mean ± 3sigma); and
* Moving Range chart which displays the difference from one point to the next.

The control limits (or natural process limits) are defined by ± 3 sigma of the measured statistic. These limits indicate the levels by which the process will fall within if there are no significant changes to the process. When any given point exceeds the 3 sigma control limits, it signals that some assignable cause may have resulted a change in the process. Likewise, continuous run of points on one side of the mean line should also be interpreted as sign of change in the process. [1] These signals support the occurrence of anomalies for a component where ratification action could be taken prevent further deterioration to the component system, within and adjacent.

**Time Series Analysis**

Given that the time-series data is auto correlated and may be subjected to certain trend and seasonal patterns, the control chart may be inadequate in considering these patterns when assessing anomalies. Hence we sought to use time series methods of forecasting, via the Anomaly package, to breakdown the time series data into its seasonal, trend, and remainder components. Observation where its remainder component falls outside of the normal remainder component limits is deemed to be an anomaly.

In generating the remainder component, Anomalize utilized 2 time series decomposition techniques:

1. Seasonal Decomposition of Time Series by Loess (STL); and
2. Seasonal Decomposition of Time Series by Median (Twitter)

STL works better where long term trend is present. While Twitter works better when short term seasonal pattern is more dominant than the long term trend.

After the time series decomposition is complete, the remainder component is assessed via 2 methods:

1. Inter Quartile Range (IQR) uses 25-75% inner quartile range to establish the distribution of the remainder. Limits are set to a factor of 3X (alpha=0.05) above and below the IQR; alpha parameter adjusts the 3X factor (alpha=0.1 correspond to a factor of 1.5X, making it easier for a point to be considered an anomaly)
2. Generalized Extreme Studentized Deviate Test (GESD) is an interactive evaluation which progressively evaluates anomalies, and re-computes the test statistics and critical values after the removing the worst offender. The alpha (default=0.05) parameter adjusts the width of the critical values.

As IQR does not depend on any loops, it is computational faster but is not as accurate in detecting anomalies as high leverage anomalies can skew the IQR median. GESD on the other hand is less resistant to high leverage anomalies but is an iterative process hence slower.

Individual-X chart: red line (3 sigma limits) vs blue line (3 standard deviation limits)

mR (Moving Range) = Absolute difference between points

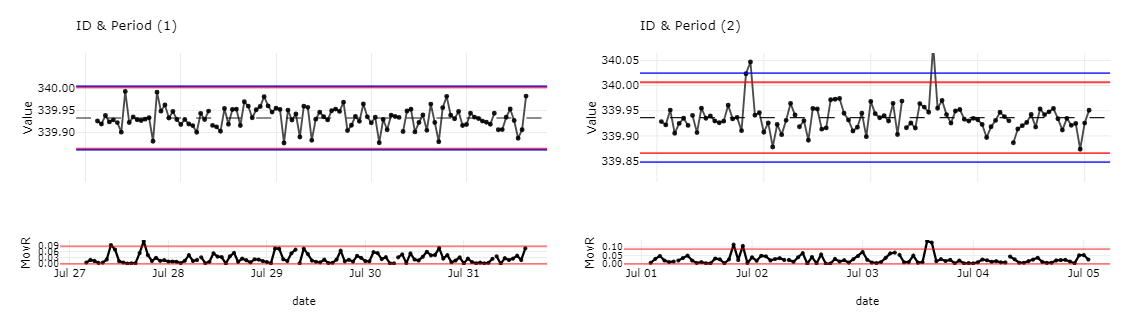
SigmaX (Sequential Standard DeviationX) = MeanmR/1.128

Upper Control Limit for XmR (UCLX) = MeanX + 3\*Sigma

Lower Control Limit for XmR (LCLX) = MeanX - 3\*Sigma

Upper Control Limit for mR (UCLmR) = 3.267\*MeanmR

Lower Control Limit for mR (LCLmR) = 0



Standard Deviation (S.D) measures total variation (systematic & random variation)

Sigma is sequence sensitive and deemphasizes systematic variation (less affected by systematic variation) which allowing us to measure the inherent random variation more clearly.

**PURPOSE OF Chart**

* XmR chart: Plot individual sensor readings against time with use control limits to detect systematic and random variation
* Sigma Violation chart: Highlight points where sigma threshold are breached
* Anomaly chart: detect anomaly after trend and seasonality factors are removed
* Compare between different Periods and/or Sensors

**HOW TO CREATE THE PLOT**

**R Packages**

|  |  |
| --- | --- |
| lubridate | For functions to work with date-times especially extraction of components of months, days, hours, etc. |
| ggQC | To initiate method under ggplot() + stat\_qc\_violations(method = “XmR”) to derive Sigma violation chart |
| anomalize | To breakdown data into its seasonal, trend, and remainder components so as to detect anomaly within remainder component, and display findings into an anomaly chart |
| ggplot2 | To create XmR (Individual+ Moving Range) and Sigma Violation charts |
| plotly | Enable interactivity with ggplot2 and anomalize charts |

**Methodology**

Create data tables using DT and data.table package.

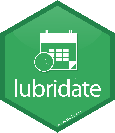
Create R-Shiny Dashboard using Shiny package

Select only required columns for horizon graph to reduce lag

Calculate percentage of changes within the group of readings by each sensor using change() function in DataCombine package.

Take sensor ID as keys and percentage of changes as values, and spreads into multiple columns.

Create horizon graph using horizonplot() function in latticeExtra package.

Data Processing

Parallel

Filter data based on selection:

1. Time interval
2. Period, exclusion (if any)
3. Sensor ID

Compute variables

1. Mean / Sigma / S.D
2. mR and its control limits
3. Trend and Seasonality

Create charts

1. XmR chart using ggplot2
2. Sigma violation chart via ggQC package
3. Anomaly chart via anomalize package

**How to Interpret XmR and Sigma Violation Charts**

XmR

* If the sigma line is close to standard deviation line, it signals the lack of systematic variation in the process.
* If the sigma line deviates from the standard deviation line, it signals significant systematic variation in the process
* Look at mR chart first, if mR chart is out of control, then control limits on Individual-X chart are meaningless.
* After reviewing the mR chart, look for point that exceed the 3 sigma limits

Sigma Violation Chart [6]

* In addition to looking for points exceeding the 3 sigma limit, we can apply additional rules to provide an earlier warning that process are out of control

1. Violation same side: 8 or more consecutive points on the same side
2. Violation same side: 4 or more consecutive points on the same side exceeding 1 sigma
3. Violation same side: 2 or more consecutive points on the same side exceeding 2 sigma
4. Violation same side: any points exceeding 3 sigma

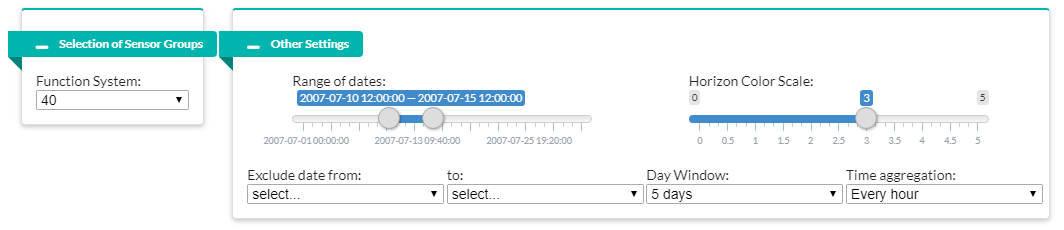
For points that are deemed out of control, consider the context, possible reasons for failure, and if it warrants further actions

**How to Interpret Anomaly Chart**

**HOW TO USE THE PLOT**

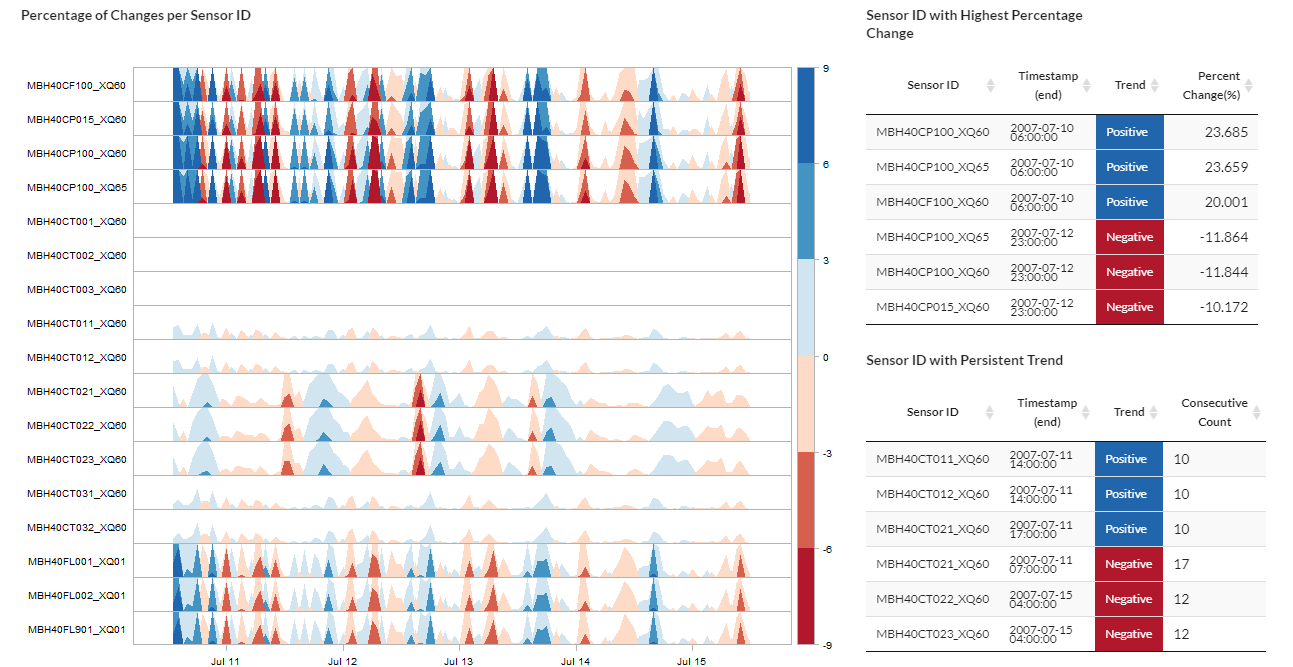
**Allowable selections:**

* Function system
* Range of dates (minimum, maximum)
* Exclude dates
* Day window
* Time aggregation
* Horizon color scale



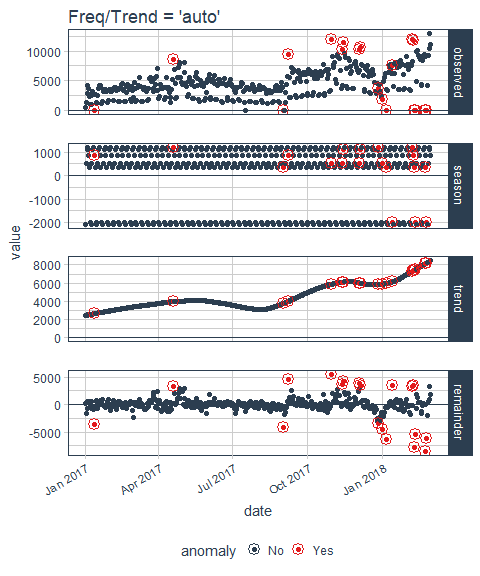
**EDA:**

1. The range of dates slider can be adjusted accordingly to achieve the desired observation period. Upon selecting full range, percentage of changes can be observed holistically in attempt to pick up sensors with prominent positive or negative value or trend within the same function system.
2. The day window input specifies the number of days for the observation period in order to zoom in on a portion of the graph. The range of dates slider will be reactively updated and can be drag left or right to observe anomalies with greater details for across the full range. (Horizon color scale slider is to make adjustment to the scale of each color segment to enhance the readability.)
3. Within the observation period, top few sensors with highest percentage of changes and with persistent trends, from both positive and negative sections, are displayed together with their timestamp on data tables. They are potentially abnormal sensor readings.
4. These abnormalities suggest that potentially abnormal sensor readings and timestamp that should go for more in-depth investigation.



**Sensors spotted with unusual peak**

**Sensors spotted with persistent trend**

****

**LESSON LEARNT**

There are few approaches to create horizon graph in R shiny. Here is the overview of the approaches, in which our team tried, along with their respective pros and cons.

1. **Horizon graph via ggTimeSeries package**

(source: <https://cran.r-project.org/web/packages/ggTimeSeries/vignettes/ggTimeSeries.html>)

This horizon graph package is based on ggplot2.

|  |  |
| --- | --- |
| **Cons** | **Cons** |
| **Convenient coding with layering system** | **Bad readability at both big and small scales** |

1. **Horizon graph via base graphics**

(source: <https://www.r-bloggers.com/horizon-plots-in-base-graphics/>)

The original code for this horizon plot was taken from R-bloggers.com by klr, and our team further modified it so that each sensor’s horizon graph can be stacked vertically.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| **Good readability at small scale** | **Bad readability at big scale** |

1. **Horizon graph via cubism.js**

(source: <https://github.com/kbroman/horizon>)

This R/horizon package is a htmlwidgets-based package using cubism.js but modified by Karl W Broman to handle static data set instead of real-time. The difference with the usual horizon graph is that negative values are offset instead of mirrored. The positive values in blue are placed at bottom axis, while negative values in red are at top axis.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| **Good interactive overlay** | **Unable to zoom in to a portion of graph as each pixel encodes a distinct point in time** |

1. **Horizon chart via latticeExtra package**

(source: <https://www.rdocumentation.org/packages/latticeExtra/versions/0.6-28/topics/horizonplot>)

This horizon graph package is the final one used by our team. The horizonplot() function is taken from latticeExtra package which was built on the infrastructure provided by the lattice package. While there is a slight bug in the package that causes some overlap between positive and negative areas, it provides good readability at both large and small scales to estimate the percentage of changes between sensors.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| **Good readability at both big and small scales** | **Positive and negative areas overlap** |