1 Module 3: Service Utilization

This module is used to:

- Detect disruptions or surpluses in service use.
- Compare current service use volumes to historical trends and seasonality, adjusting for data quality.

1.1 Background

Service utilization refers to the volume of health services delivered and reported through routine health information systems (i.e. DHIS2). It reflects how populations access and use essential healthcare services over time and across different regions.

However, service utilization can fluctuate due to various factors, including seasonal trends, policy changes, pandemics, or other external shocks. Identifying whether these variations are part of normal patterns or signal significant disruptions is important for health system monitoring and decision-making.

The control chart analysis helps determine whether deviations in service volumes are part of normal fluctuations or indicate significant disruptions.

The disruption analysis helps quantify the impact of these disruptions by measuring how service volumes changed during flagged periods.

1.2 Overview

The Service Utilization module is designed to evaluate trends in health service usage by identifying disruptions and surpluses in service delivery over time. The module consists of two key components:

1. **Control chart analysis**
2. **Disruption analysis**

1.2.1 Control Chart Analysis

Service volumes are aggregated at the specified geographic level (configurable via `CONTROL_CHART_LEVEL`). The pipeline removes outliers (`outlier_flag == 1`), fills in missing months, and filters low-volume months (<50% of global mean volume).

A robust regression model estimates expected service volumes per indicator × geographic area (`panelvar`). A centered rolling median is applied to smooth the predicted values. Residuals (actual - smoothed) are standardized using MAD. Disruptions are identified using a rule-based tagging system. Each rule is controlled by user-defined parameters, allowing customization of the sensitivity and behavior of the detection logic:

Sharp Disruptions Flags a single month when the standardized residual (residual divided by MAD) exceeds a threshold:

$$\left|\frac{\text{residual}}{\text{MAD}}\right| \geq \text{THRESHOLD}$$

- **Parameter:** `THRESHOLD` (default: `1.5`)
- Lower values make the detection more sensitive to sudden spikes or dips.

Sustained Drops Flags a sustained drop if: - Three consecutive months show mild deviations (standardized residual ≥ 1), and - The final month also exceeds the `THRESHOLD`.

This captures slower, compounding declines.

Sustained Dips Flags periods where the actual volume falls consistently below a defined proportion of expected volume (smoothed prediction):

$$\text{count\_original} < \text{DIP\_THRESHOLD} \times \text{count\_smooth}$$

- **Parameter:** `DIP_THRESHOLD` (default: `0.90`)

- Users can adjust this to detect deeper or shallower dips (e.g., `0.80` for a 20% drop).

Sustained Rises Symmetric to dips, flags periods of consistent overperformance:

$$\text{count\_original} > \text{RISE\_THRESHOLD} \times \text{count\_smooth}$$

- **Parameter:** `RISE_THRESHOLD` (default: `1 / DIP_THRESHOLD`, e.g., `1.11`)
- Users can adjust this to detect upward surges in volume.

Missing Data Flags when 2 or more of the past 3 months have missing (`NA`) or zero service volume. - **Fixed rule**.

Recent Tail Override Automatically flags all months in the last 6 months of data to ensure recent trends are reviewed, even if model-based tagging is not conclusive. - **Fixed rule**.

These parameters can be adjusted to make the detection stricter or more lenient depending on the use case.

A final `tagged` flag is assigned where any of the above conditions are met. Results are saved in `M3_chartout.csv`.

1.2.2 Disruption Analysis

Once anomalies are identified and saved in `M3_chartout.csv`, the disruption analysis quantifies their impact using regression models. These models estimate how much service utilization changed during the flagged disruption periods by adjusting for long-term trends and seasonal variations.

For each indicator, we estimate:

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{date} + \sum_{m=1}^{12} \gamma_m \cdot \text{month}_m + \beta_2 \cdot \text{tagged} + \epsilon_{it}$$

where: - Y_{it} is the observed service volume, - date captures time trends, - month_m controls for seasonality, - tagged is the disruption dummy (from the control chart analysis), - ϵ_{it} is the error term.

The coefficient on `tagged` (β_2) measures the relative change in service utilization during flagged disruptions. Separate regressions are run at the national, province and district levels to assess the impact across different geographic scales.

1.3 Detailed Analysis Steps

1.3.1 PART 1 - Control Chart Analysis

Step 1: Prepare the Data

- Load raw HMIS service utilization data.
- Merge in outlier flags (`outlier_flag`) by facility × indicator × month.
- Remove rows flagged as outliers (`outlier_flag == 1`).
- Create a `date` variable from `period_id` and extract `year` and `month`.
- Create a unique `panelvar` for each geographic area-indicator combination.
- Aggregate data to the specified geographic level by summing `count_model` (based on `SELECTEDCOUNT`) by date.
- Fill in missing months within each panel to ensure continuity.
- Fill missing metadata using forward and backward fill.

Step 2: Filter Out Low-Volume Months

- Compute the global mean service volume for each `panelvar`.
- If `count_original` is <50% of the global mean, drop the value by setting it to `NA`.

Step 3: Apply Regression and Smoothing

Estimate expected service volume using robust regression, then smooth the predicted trend.

Model fitting: - If $ $12 observations and >12 unique dates:

$$Y_{it} = \beta_0 + \sum \gamma_m \cdot \text{month}_m + \beta_1 \cdot \text{date} + \epsilon_{it}$$

- If only $ $12 observations:

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{date} + \epsilon_{it}$$

- If insufficient data: use the median of observed values.

- Fit robust regression (`rlm`) for each panel.
- **Apply rolling median smoothing** to predictions:

$$\text{count\_smooth}_{it} = \text{Median}(\text{count\_predict}_{t-k}, \ldots, \text{count\_predict}_{t}, \ldots, \text{count\_predict}_{t+k})$$

- **Parameter:** `SMOOTH_K` (default: 7, must be odd)
- Larger `SMOOTH_K` smooths more; smaller retains more variation.
- If smoothing is not possible (e.g., at series edges), fallback to model predictions.
- **Calculate residuals:**

$$\text{residual}_{it} = \text{count\_original}_{it} - \text{count\_smooth}_{it}$$

- **Standardize residuals using MAD:**

$$\text{robust\_control}_{it} = \text{residual}_{it} / \text{MAD}_i$$

This standardized control variable is used to detect anomalies in Step 4.

Step 4: Tag Disruptions

Apply rule-based tagging to identify potential disruptions. Each rule is governed by user-defined parameters that can be tuned for sensitivity.

Sharp Disruptions - Condition: `|robust_control|  THRESHOLD` - **Parameter:** `THRESHOLD` (default: 1.5) - Tags the individual month.

Sustained Drops - Condition: 3 consecutive months with mild deviations (`|robust_control|  1`), where the final month also exceeds `THRESHOLD`. - Only the **final month** in the sequence is tagged (`tag_sustained == 1`). - Helps identify gradual declines that culminate in a significant deviation.

Sustained Dips - Condition: `count_original < DIP_THRESHOLD × count_smooth` for 3 or more consecutive months - **Parameter:** `DIP_THRESHOLD` (default: 0.90) - If the condition holds for 3 or more months in a row, the entire sequence is tagged.

Sustained Rises - Condition: `count_original > RISE_THRESHOLD × count_smooth` for 3 or more consecutive months - **Parameter:** `RISE_THRESHOLD` (default: 1 / DIP_THRESHOLD, e.g., 1.11) - If the condition holds for 3 or more months in a row, the entire sequence is tagged.

Missing Data - Condition: 2 or more of the past 3 months are missing (`NA`) or zero - Tags the final month in the flagged sequence.

Recent Tail Override - Automatically tags **all months in the last 6 months** of data to ensure recent trends are reviewed, even if model-based tagging is not conclusive.

Final Flag: A month is assigned `tagged = 1` if **any** of the following conditions are met: - `tag_sharp == 1` - `tag_sustained == 1` - `tag_sustained_dip == 1` - `tag_sustained_rise == 1` - `tag_missing == 1` - It falls within the most recent 6 months (`last_6_months == 1`)

Tagged records are saved in `M3_chartout.csv` and passed to the disruption analysis.

1.3.2 PART 2 - Disruption Analysis

Step 1: Data preparation

- The `M3_chartout` dataset is merged with the main dataset to integrate the `tagged` variable, which identifies flagged disruptions.
- The lowest available geographic level (`lowest_geo_level`) is identified for clustering, based on the highest-resolution `admin_area_*` column available.

The regression pipeline follows a structured, multi-level approach, starting from the broadest level (country-wide) and moving to more granular levels (province, then district).

Step 2: Country-wide regression

The country-wide regression estimates how service utilization changes at the national level when a disruption occurs. Instead of analyzing individual provinces or districts separately, this model considers the entire country's data in a single regression. Errors are clustered at the lowest available geographic level (`lowest_geo_level`), which can be districts or wards.

- A panel regression model is applied at the country-wide level, estimating the expected service volume (`expect_admin_area_1`) for each indicator (`indicator_common_id`).
- The model adjusts for historical trends and seasonal variations, ensuring that deviations are measured against expected patterns.
- If a disruption (`tagged` = 1) is detected, the predicted service volume is adjusted by subtracting the estimated effect of the disruption to isolate its impact.

Model Specification:

For each `indicator_common_id`, we estimate:

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{date} + \sum_{m=1}^{12} \gamma_m \cdot \text{month} + \beta_2 \cdot \text{tagged} + \epsilon_{it}$$

Where: - Y_{it} = volume (e.g., number of deliveries) - date = time trend - month_m = controls for seasonality (factor variable) - tagged = dummy for disruption period - ϵ_{it} = error term, clustered at the district level (`admin_area_3`)

Step 3: Province-level regression

The province-level disruption regression estimates how service utilization changes at the province level when a disruption occurs. Unlike the country-wide model, which treats the entire country as a single unit, this approach runs separate regressions for each province to capture regional variations in service utilization during disruptions. Errors are clustered at the lowest available geographic level, districts or wards, to account for local variation within each province.

- A fixed effects panel regression model is applied at the province level, estimating expected service volume (`expect_admin_area_2`) while controlling for province-specific factors.
- The model adjusts for historical trends and seasonal variations, ensuring deviations are compared against expected patterns.
- If a disruption (`tagged` = 1) is detected, the predicted service volume is adjusted by subtracting the estimated effect of the disruption to isolate its impact.

Model specification:

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{date} + \sum_{m=1}^{12} \gamma_m \cdot \text{month} + \beta_2 \cdot \text{tagged} + \alpha_{\text{province}} + \epsilon_{it}$$

Where: - Y_{it} = volume (e.g., number of deliveries) - date = time trend - month_m = controls for seasonality (factor variable) - tagged = dummy for disruption period - α_{province} = province fixed effects - ϵ_{it} = error term, clustered at the district level (`admin_area_3`)

Step 4: District-level regression

The district-level disruption regression estimates how service utilization changes at the district level when a disruption occurs. This approach runs separate regressions for each district to capture localized variations in service utilization during disruptions. Errors are clustered at the lowest available geographic level, typically wards or districts, to account for variations within each district.

- A fixed effects panel regression model is applied at the district level, estimating expected service volume (`expect_admin_area_3`) while controlling for district-specific factors.
- The model adjusts for historical trends and seasonal variations, ensuring deviations are compared against expected patterns.
- If a disruption (`tagged` = 1) is detected, the predicted service volume is adjusted by subtracting the estimated effect of the disruption to isolate its impact.

Model specification:

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{date} + \sum_{m=1}^{12} \gamma_m \cdot \text{month} + \beta_2 \cdot \text{tagged} + \alpha_{\text{district}} + \epsilon_{it}$$

Where: - Y_{it} = volume (e.g., number of deliveries) - date = time trend - month_m = controls for seasonality (factor variable) - tagged = dummy for disruption period - α_{district} = district fixed effects - ϵ_{it} = error term

Regression Outputs:

Each regression level produces the following outputs:

- **Expected values (`expect_admin_area_*`)**: Predicted service volume adjusted for seasonality and trends.
- **Disruption effect (`b_admin_area_*`)**: Estimated relative change during disruptions:

$$b_{\text{admin\_area\_*}} = -\frac{\text{diff mean}}{\text{predict mean}}$$

- **Trend coefficient (`b_trend_admin_area_*`)**: Reflects long-term trend.
  - Positive = increasing service use
  - Negative = declining service use
  - Near zero = stable trend
- **P-value (`p_admin_area_*`)**: Measures statistical significance of the disruption effect.
  - Lower values = stronger evidence of true disruption

Step 5: Prepare Outputs for Visualization

Once expected values have been calculated for each level (country, province, district), the pipeline compares predicted and actual values to assess the magnitude of disruption.

For each month and indicator, the pipeline calculates:

- **Absolute and percentage difference** between predicted and actual values:

$$\text{diff\_percent} = 100 \times \frac{\text{predicted} - \text{actual}}{\text{predicted}}$$

- A configurable threshold parameter `DIFFPERCENT` (default: `10`) is used to determine when a disruption is significant.

  If the percentage difference exceeds $±$10%, the expected (predicted) value is retained and used for plotting and summary statistics. Otherwise, the actual observed value is used.

This ensures that minor fluctuations do not lead to artificial disruptions in the visualization, while meaningful deviations are preserved.

- The final adjusted value for plotting is stored in a field such as `count_expected_if_above_diff_threshold`.

  This value reflects either:

  - The predicted count (if deviation > threshold), or
  - The actual count (if within acceptable range).

This logic is applied consistently across admin level 1 (national), admin level 2 (province), and admin level 3 (district).

These adjusted values are then exported as part of the final output files for each level.

1.4 Configuration Parameters

The module behavior is controlled by several key parameters:

Parameter	Default	Description
`SELECTEDCOUNT`	"count_final_outliers"	Data column used for analysis
`VISUALIZATIONCOUNT`	"count_final_both"	Data column used for visualization
`SMOOTH_K`	7	Rolling median window size (must be odd)
`THRESHOLD`	1.5	MAD units threshold for sharp disruptions
`DIP_THRESHOLD`	0.90	Proportion threshold for sustained dips
`DIFFPERCENT`	10	Percentage threshold for using predicted vs actual values
`CONTROL_CHART_LEVEL`	"admin_area_3"	Geographic level for control chart analysis
`RUN_DISTRICT_MODEL`	TRUE	Whether to run district-level regressions
`RUN_ADMIN_AREA_4_ANALYSIS`	FALSE	Whether to run finest-level analysis

1.5 Outputs

The Service Utilization module generates several key output files:

Control Chart Results: - `M3_chartout.csv` - Contains flagged disruptions with period identifiers, geographic areas, indicators, and tagging status - `M3_service_utilization.csv` - Copy of adjusted data for service utilization analysis

Disruption Analysis Results: - `M3_disruptions_analysis_admin_area_1.csv` - National-level disruption estimates - `M3_disruptions_analysis_admin_area_2.csv` - Province/regional-level disruption estimates - `M3_disruptions_analysis_admin_area_3.csv` - District/state-level disruption estimates (if `RUN_DISTRICT_MODEL = TRUE`) - `M3_disruptions_analysis_admin_area_4.csv` - Finest geographic level estimates (if `RUN_ADMIN_AREA_4_ANALYSIS = TRUE`)

Key Messages Dataset: - `M3_all_indicators_shortfalls.csv` - Summary of shortfalls and surpluses by indicator and time period for external analysis

1.5.1 Output File Structure

Each disruption analysis file contains: - **Geographic identifier** (`admin_area_*`) - **indicator_common_id** - Health service indicator - **period_id** - Time period (YYYYMM format) - **quarter_id** - Quarter identifier - **year** - Year - **count_sum** - Actual service volume - **count_expect_sum** - Expected service volume (adjusted for disruptions) - **count_expected_if_above_diff_threshold** - Plotting value (expected if disruption $\geq$ DIFFPERCENT, otherwise actual)

1.6 Interpretation Guidelines

Disruption Effects (b_admin_area_*): - Negative values indicate service volume shortfalls during disrupted periods - Positive values indicate service volume surpluses during disrupted periods - Values closer to zero indicate smaller disruption impacts

P-values (p_admin_area_*): - Values < 0.05 suggest statistically significant disruptions - Values > 0.05 may indicate normal variation rather than true disruptions

Trend Coefficients (b_trend_admin_area_*): - Positive values indicate increasing service utilization over time - Negative values indicate declining service utilization over time - Values near zero indicate stable utilization patterns

1.7 Technical Implementation Notes

Geographic Clustering: - Regressions use clustered standard errors at the lowest available geographic level - This accounts for within-area correlation in service delivery patterns

Data Requirements: - Minimum 12 observations with >12 unique dates for full seasonal models - Minimum 12 observations for trend-only models - Below thresholds trigger fallback to median imputation

Performance Considerations: - District-level analysis can be computationally intensive for large datasets - Set `RUN_DISTRICT_MODEL = FALSE` for faster execution - Admin_area_4 analysis is disabled by default due to computational overhead

Quality Assurance: - Outliers are removed prior to control chart analysis based on Module 1 flags - Low-volume months (<50% of mean) are excluded to improve model stability - Recent months are automatically flagged to ensure current disruptions are captured

---

Last edit 2025 September 3