USAID GLOBAL HEALTH SUPPLY CHAIN PROGRAM

Procurement and Supply Management

|  |
| --- |
| LaDATA (Logistic and Data Analysis for Targeted Action)  Using big data for targeted actions for the supply of family planning commodities  User Guide (GitHub Guide)  November 2022 |

The USAID Global Health Supply Chain Program-Procurement and Supply Management (GHSC-PSM) project is funded under USAID Contract No. AID-OAA-I-15-0004. GHSC-PSM connects technical solutions and proven commercial processes to promote efficient and cost-effective health supply chains worldwide. Our goal is to ensure uninterrupted supplies of health commodities to save lives and create a healthier future for all. The project purchases and delivers health commodities, offers comprehensive technical assistance to strengthen national supply chain systems, and provides global supply chain leadership.

GHSC-PSM is implemented by Chemonics International, in collaboration with Arbola Inc., Axios International Inc., IDA Foundation, IBM, IntraHealth International, Kuehne + Nagel Inc., McKinsey & Company, Panagora Group, Population Services International, SGS Nederland B.V., and University Research Co., LLC. To learn more, visit [ghsupplychain.org](http://www.ghsupplychain.org/)

DISCLAIMER:  
The views expressed in this publication do not necessarily reflect the views of the U.S. Agency for International Development or the U.S. government.

Table of Contents

[Acronyms 2](#_Toc82773252)

[Background and Purpose 6](#_Toc82773253)

[Access and Setup 6](#_Toc82773254)

[Data Requirements 6](#_Toc82773255)

[Data Coverage 6](#_Toc82773256)

[Data Fields 7](#_Toc82773257)

[Data Cleaning 7](#_Toc82773258)

[Running the Inventory Analysis 7](#_Toc82773259)

[Tool Startup 7](#_Toc82773260)

[Initial runs 7](#_Toc82773261)

[Subsequent runs 9](#_Toc82773262)

[Settings Configuration 9](#_Toc82773263)

[Running the Tool 9](#_Toc82773264)

[Using the Report 9](#_Toc82773265)

[Creating a Custom Report 9](#_Toc82773266)

[Troubleshooting 9](#_Toc82773267)

[Annex 1: Understanding the Code Structure 10](#_Toc82773268)

[Notes for Developers 10](#_Toc82773269)

Acronyms

|  |  |
| --- | --- |
| COV  GHSC-PSM  eLMIS  LMIS | Coefficient of variance  Global Health Supply Chain Program – Procurement and Supply Management  Electronic Logistics Management Information System  Logistics Management Information System |

Background and Purpose

The Global Health Supply Chain Program – Procurement and Supply Management project (GHSC-PSM) has developed several analyses with the purpose of using country Logistics Management Information Systems (LMIS) data to analyze inventory stock patterns over time. While many of these analyses were very similar in nature, they required a developer or analyst to re-write or repurpose scripts to fit an individual country’s data. In 2021, GHSC-PSM Task Order 3 undertook an activity to explore refactoring of code for such commonly used analyses, using an existing inventory analysis implemented in Ghana and Nepal as a proof of concept. The purpose was to create a version of the analysis that could be easily scaled and repeated for other countries with little to no need for script adaptation.

While the purpose of this inventory analysis is to use several key supply chain metrics to provide recommendations for intervention within a country’s supply chain, there is a particular emphasis on the methods used for adapting the analysis for broader use by non-technical staff.

This document serves as a user guide for a new user running the analysis on their own data.

Access and Setup

The tool consists of data files and scripts that can be downloaded and run on the user’s desktop.

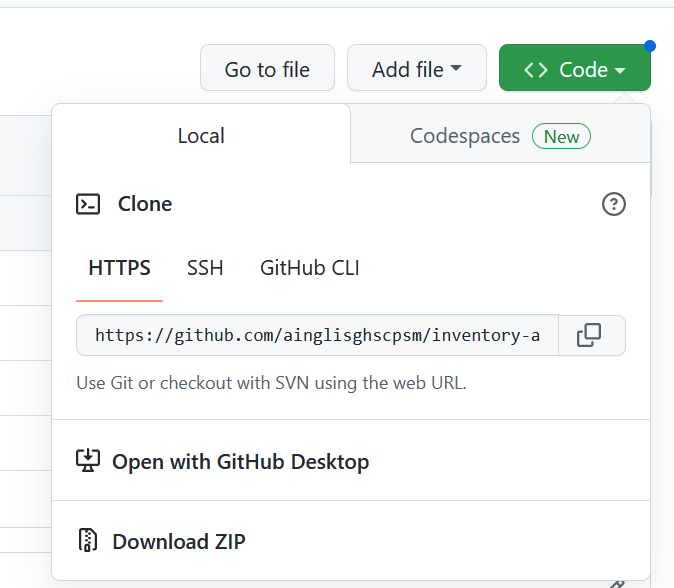
The scripts and files for the tool can be accessed GitHub:

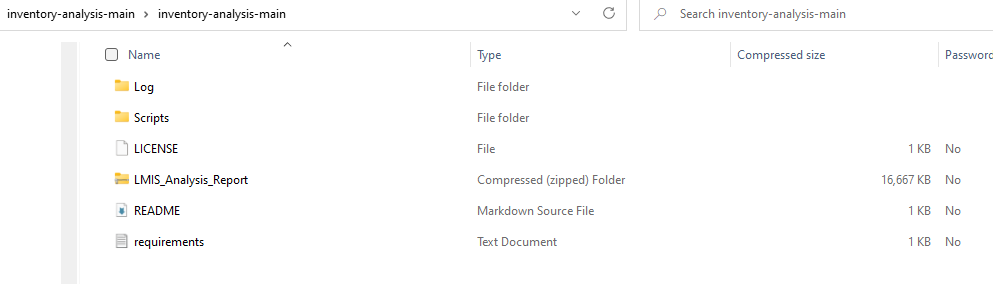
GitHub

For dissemination purposes, the scripts and files have been stored on GitHub, a file storage and version control platform, so that they can be easily downloaded, and so developers may create their own versions (forks) of the tool.

Goto: <https://github.com/ainglisghscpsm/inventory-analysis>

Depending if you are wanting look at code or implement the code, either generate a clone or Download the Zip.



If downloaded the the file system should look like this:

NOTE: The version excel dashboard has been zipped up (too large to uploaded), it unzipped into the main folder to be used.

Data Requirements

The tool can be run on data extracted from an electronic (eLMIS) system. This section will list the expectations and requirements for the data, including required fields, data types and required structure.

Data Coverage

It is recommended to have a minimum of 18 months of continuous historical inventory data for this tool. The tool can be run on a shorter span of data; however, meaningful results cannot be guaranteed.

Data Fields

Data for the tool may be either cumulative or transactional. Cumulative, meaning a single data entry represents one whole month, and transactional meaning there is a data entry each time a change in stock occurs. The required and optional fields for the tool are listed below.

Required fields for the tool:

* Facility identifier
* Product identifier
* Date
* Consumption
* Stock on hand

Optional fields:

* Additional info tied to the product (product name if not the unique identifier, product category)
* Additional info tied to the facility (facility name if not the unique identifier, region, district)

Data Cleaning

The tool expects reasonably clean data, and the expectation is that most relevant cleaning will be completed before the data is run through the tool. This is because data is expected to come from an LMIS system that already upholds a certain standard of data. The tool does implement a few common data cleaning measures which include:

* The unique identifier for the product and facility are put in all-caps.
* Additional whitespace is removed from the product and facility identifiers (e.g. leading and trailing spaces, more than one space character between words).
* The specified date field is cast to a date type in the script (provided the date field can be interpreted as a date).

A developer can add additional cleaning modules to the scripts to suit the needs of the specific dataset.

Running the Inventory Analysis

Tool Startup

Initial run

Launch the tool by running the following script inventory\_analysis\_gui.py

Graphical user interface, text, application, chat or text message

Description automatically generated

Since this is the initial run, select **New configurations** at the top of the prompt (it should already be selected). Select **Browse Files**. A file navigation dialog will open:

Graphical user interface, text, application

Description automatically generated

Select the data file (.xlsx or .csv) that contains the inventory data. Select **Open.**

Graphical user interface, application

Description automatically generated

The window will fill in with dropdowns for the user to specify each of the required (and additional) fields. The left-hand column contains the required value

1. Select a **Data Type** of transactional or cumulative (see Data Requirements above for a description of each).
2. Select the fields for remaining values. The corresponding dropdowns contain a list of all field names from the loaded file. Fill in the corresponding field name for each drop down. Any value labeled (Additional) is optional and can be used to attach relevant information about the product or facility (e.g. Product Category, Region, District) that the user wishes to carry through to the output data. Note that field names cannot be repeated.
   1. **Consumption** and **Stock on Hand** have radio buttons next to them for specifying how blank values should be handled. Select either “blank fields = 0” or “blank fields are missing” depending on the data set.
3. Select the **Time Window for Rates** for the key metrics to be calculated on. The default is 12 continuous records and is recommended. Note that the time window should not exceed the span of records in the loaded inventory data (e.g. if there are only 8 months of data, the **Time Window for Rates** should be <= 8).

Once complete, the window should look like this (dropdown values will vary by data set):

Graphical user interface, application

Description automatically generated

The tool is ready to be run.

Running the Tool

Select **Run Analysis** at the bottom of the window. This triggers the script to begin running the analysis. Run times will vary depending on the size of the input data. When the analysis is complete the following will appear at the bottom of the window:

Graphical user interface, text, application, chat or text message

Description automatically generated

The total number of products and facilities are listed, along with buttons to save the current settings or open the Excel report.

Settings Configuration

Save My Settings

After a successful run, the given settings and field selections can be saved for future analysis runs. This is recommended for situations where the input data have the same format each time it’s updated (e.g. a recurring LMIS extract).

To do this, select **Save My Settings**. This will save all of the current settings, but not the current file. If configurations change, they can be saved again. Only one set of configurations can be saved at a given time.

Running the tool with saved configurations

Once settings have been saved the configurations can be used to quickly perform repeat analyses.

Prepare the new data set that has the same fields as those saved in the configuration. Launch the tool by double clicking **inventory\_analysis.exe**. This time toggle **Use saved settings** before loading the file. Select **Browse Files**. Use the file dialogue to select the new file. Your formerly saved configurations will auto-populate in the dropdown fields:

Graphical user interface, application

Description automatically generated

Change any configurations that may have changed, then select **Run Analysis**.

Viewing the Log

The tool logs information and errors in the **inventory\_analysis.log** file are included in the tool file system under the **Log** folder. Each log is timestamped with the date and time that the update and/or error occurred.

Graphical user interface, application, Word

Description automatically generated

Note: Upon first download the log file will not exist. It will be populated after the first run of the tool.

Using the Report

The tool contains a default dashboard in Excel for viewing the results of the analysis. The underlying data outputs can be used with another dashboarding tool for those looking to create a custom report (details below).

In the file system for the tool select **LMIS Analysis Report.xlsx**

The report should open with a pop-up stating the data connections have been updated. These are the connections to the text files within the tool directory (i.e the output tables from running the analysis).

Note: Report connections may need to be reset upon the first use of the Excel dashboard to ensure they point to the correct directory. This will only need to be done once. This can be done by going to **Data > Queries & Connections > Connection Properties.** For each table listed, go to the Definition tab and update the current connection by selecting **Browse**. Navigate to the **Scripts** folder of the inventory\_analysis directory and select the .txt file with the name corresponding to the table to be updated. Walk through the prompts, set the delimiter as space, and leave the rest of the settings as the default. Once this is complete for all tables, the data should flow automatically through the Excel report.

Primary Dashboard

Analysis tab of dashboard

**Chart

Description automatically generated**

The first tab **Analysis** contains a dashboard comparing the inventory turnover to the coefficient of variance of the consumption for each facility and product combination.

Select a single product at the top of the page to view the patterns for that product.

Select a date period at the top of the page.

Press the **Refresh** button to update visuals.

The scatterplot on the righthand side of the screen shows where each facility falls for that period. Facilities are divided into different categories based on where they fall on the scatterplot, each corresponding to a risk category (left of scatterplot) and recommended action (lower left of scatterplot).

Scatterplot showing Inventory Turnover Rate vs Coefficient of Variance (COV) Consumption for Microlut

Chart, scatter chart

Description automatically generated

Facility categories and corresponding actions

A picture containing diagram

Description automatically generated

The inventory turnover over time can be seen in the lower right of the dashboard. Details on the metrics for each facility for the selected product and period are listed in the bottom right.

Facility Analysis

The \_ tab provides an overview of facilities across all products.

Table

Description automatically generated

Stock Status

The dashboard contains details on the recent stock status of products.

Table

Description automatically generated

Controls

To fit different distribution schedules, thresholds for inventory turnover and COV Consumption can, as well as scoring thresholds, be adjusted in the **Controls** tab.

Table

Description automatically generated

Creating a Custom Report

The tool exports data into standalone text files independent of the default Excel dashboard. This enables the user to build their own report, using these files as the backend, either in Excel or another dashboarding tool (e.g. Power BI). The files containing data can be found in the **Script** folder of the tool directory with the following names:

|  |  |
| --- | --- |
| File Name | Contains info on |
| dates\_tables.txt | Complete lists of dates contained in dataset |
| facility\_list.txt | Complete list of facilities in dataset |
| plot\_data.txt | Special data table for Excel dashboard graph |
| product\_list.txt | Complete list of products in dataset |
| rates\_data.txt | Inventory turnover and COV consumption |
| stock\_data.txt | Most recent stock status of each product |

Troubleshooting

For issues and errors with the tool or for questions please contact Clare Isaacson (cisaacson@ghsc-psm.org)

Annex 1: Understanding the Code Structure

The Python components that run the tool consist of two Python files, each containing a single class:

|  |  |  |
| --- | --- | --- |
| File Name | Class Name | Functions |
| inventory\_analysis.py | InventoryRates | * Read input data * Perform basic data cleaning * Calculate metrics for analysis * Generate output tables (input tables for report) |
| inventory\_analysis\_gui.py | IAGui | * Run the graphical user interface * Gather user inputs * Use user inputs to instantiate and run the InventoryRates class * Trigger data output |

Developers can add analytic functionality to the tool by creating new functions within the InventoryRates class. They can expand the user interface and operational flow of the tool by adding functions to the IAGui class. Note that since the IAGui controls the process flow of the instance of InventoryRates, added functions in InventoryRates will likely need to be called within the IAGui class to be executed.

Notes for Developers

* If updates are made to the code, the developer will need to re-make the standalone executable.
* If a new report is created the **Open Report** button will need to be updated to launch the new report (see function **openReport** in file **inventory\_analysis\_gui.py**)

Annex 2: Inventory Turnover and COV Consumption



