# Data Ingestion Pipeline using R

The Dashboard Automation project is designed to automate the ingestion and uploading of data to the Sandpit as part of a reporting pipeline and create an output in the form of a Dashboard and a PowerPoint Presentation. This document guides users through the process, explains the code, and provides detailed instructions on configuring and running the data ingestion part of the pipeline.

**Prerequisites**

Before you begin, ensure you have the following software installed:

* [R 4.4.1](https://gbr01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcran.ma.imperial.ac.uk%2Fbin%2Fwindows%2Fbase%2FR-4.4.1-win.exe&data=05%7C02%7Carush.mohan%40nhs.net%7Ccb378cc3af3e40bc4c2c08dcac9080db%7C37c354b285b047f5b22207b48d774ee3%7C0%7C0%7C638574983830048359%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C0%7C%7C%7C&sdata=T61KK4vP7YTE5RigVTXARbNTFUHAvCs8tNvZH9ZOfII%3D&reserved=0)
* [R tools 44](https://gbr01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcran.r-project.org%2Fbin%2Fwindows%2FRtools%2Frtools44%2Ffiles%2Frtools44-6104-6039.exe&data=05%7C02%7Carush.mohan%40nhs.net%7Ccb378cc3af3e40bc4c2c08dcac9080db%7C37c354b285b047f5b22207b48d774ee3%7C0%7C0%7C638574983830059391%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C0%7C%7C%7C&sdata=XCnG9%2BAkjxNjOVICEfxcaKTfaczeK1P66HCp2txfrCk%3D&reserved=0)
* [R studio](https://gbr01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdownload1.rstudio.org%2Felectron%2Fwindows%2FRStudio-2024.04.2-764.exe&data=05%7C02%7Carush.mohan%40nhs.net%7Ccb378cc3af3e40bc4c2c08dcac9080db%7C37c354b285b047f5b22207b48d774ee3%7C0%7C0%7C638574983830067454%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C0%7C%7C%7C&sdata=IDlBy1a4FZc5wGBnGUTnVGYb6fJWCXc%2BmTm0ohjyoCc%3D&reserved=0)
* [Git – Downloading Package (git-scm.com)](https://gbr01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgit-scm.com%2Fdownload%2Fwin&data=05%7C02%7Carush.mohan%40nhs.net%7Ccb378cc3af3e40bc4c2c08dcac9080db%7C37c354b285b047f5b22207b48d774ee3%7C0%7C0%7C638574983830074441%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C0%7C%7C%7C&sdata=7mEIIImmecmURx4aPfYmuJa9vas1LHnNH7X6Oryt5dU%3D&reserved=0)
* Create a [github](https://gbr01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2F&data=05%7C02%7Carush.mohan%40nhs.net%7Ccb378cc3af3e40bc4c2c08dcac9080db%7C37c354b285b047f5b22207b48d774ee3%7C0%7C0%7C638574983830080792%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C0%7C%7C%7C&sdata=tEoZOQnH2tpU1ujO3a8JbU6x8MZVMNaZ91p%2FBrqCBEY%3D&reserved=0) account

**Clone the GitHub Repository**

--

To ensure all necessary packages and dependencies are installed, follow these steps:

**Open the R project**:

* Launch RStudio.
* Go to **File** > **Open Project**.
* Navigate to the cloned repository directory and open the R project file (Dashboard\_Automation).

**Data Files**: Ensure that the necessary Excel data files are present in the data directory, organized by date into Flex and Freeze subdirectories.

**Database Access**: Ensure that you have the necessary permissions and access to the SQL Server database. The script uses an ODBC connection, so you may need to configure your ODBC settings to connect to the database.

**Input CSV Files**: Ensure that the necessary input CSV files (Provider\_indicator\_list.csv, Provider\_inconsistency\_adjustment.csv, and Sheet\_parameters.csv) are present in the input directory. These files can also be used to modify and update the parameters without the need to alter the R code.

**Initialise the R environment**

* In the R console, run the following command: renv::init()

This will install the required packages for the pipeline.

**Set Parameters**:

Update the parameters at the top of the R Markdown file to match your needs:

A screenshot of a computer

Description automatically generated

report\_date: YYYYMMDD # Update with the desired report date

freeze\_window: 7 # Update with the desired freeze window

**Source the Functions**:

The project uses a custom functions script located in the src directory. Ensure it is sourced correctly in the R Markdown file:

A black rectangular object with a white border

Description automatically generated

Ensure that you have the necessary input files in the input directory (Provider\_indicator\_list.csv, Provider\_inconsistency\_adjustment.csv, Sheet\_parameters.csv).

**Execute the Data Processing Pipeline**:

Run the R Markdown document to perform the data ingestion and transformation processes.

Click the green arrow button at the top right corner of each chunk to execute it. Alternatively, you can click the ‘Run’ button at the top right of the r window and select ‘Run All’.

A screenshot of a computer

Description automatically generated

**Upload Data**:

The script automatically handles the data upload to the database, including checking for duplicates and handling transactions.

**Explanations for the different chunks of code within the RMD file**

**1. Setup**

This section loads the necessary functions from the functions.R script and sets up the R Markdown chunk options.

**2. Defining Folder Directories**

This section calculates the main folder date and freeze date based on the report date and freeze window. It defines the directory paths for flex and freeze files.

**3. Listing the Files in the Directory**

This section lists all the Excel files in the Flex and Freeze directories, storing them in separate vectors.

This section combines the file lists for flex and freeze data, adds a column to differentiate between flex and freeze data, and extracts the provider name from the file name.

**5. Defining Columns and Sheet Parameters**

This section reads the input CSV files for indicators and sheet parameters. It creates a list of indicators and handles inconsistencies by removing missing indicators and adding new ones as specified in the adjustment file.

**6. Extracting the Data**

This section reads and combines all Excel files into a single data frame using a custom function ingest\_raw\_data\_file from the functions.R script.

**7. Pivoting Data to Wide Format**

This section pivots the long-format data to a wide-format table, making it suitable for certain types of analyses or reporting. It also calculates the week number based on the fiscal year starting in April.

**8. Processing for Upload**

This section processes the data for upload, separating the indicator column into two separate columns: Indicator and Description.

**9. Adding Column for Week Number**

This step was integrated into the pivoting step to calculate the fiscal week number.

**10. Creating Intermediate Excel File for QA**

This section creates an intermediate Excel file from the combined data for quality assurance (QA) purposes, saving it to the output directory.

**11. Connecting to the Sandpit**

This section establishes a connection to the SQL Server database using ODBC. Ensure the ODBC Data Source Name (DSN) is correctly configured.

**12. Checking for Duplicates and Uploading to Sandpit - Long Data**

This section uploads long-format data (flex and freeze) to the database, checking for duplicates and handling them as needed.

**13. Checking for Duplicates and Uploading to Sandpit - Wide Data**

This section uploads wide-format data (flex and freeze) to the database, checking for duplicates and handling them as needed.

**14. Disconnecting from the Sandpit**

This section closes the connection to the Sandpit.