How to Go with the (Data)Flow

User Guide for `arcedfold` and `arceddataflow`

`arcedfold` and `arceddataflow` are two separate Stata commands to create folders and do files for projects.

arcedfold

The `arcedfold` command in Stata is designed to help users quickly set up standardized folder structures for their projects. Originally developed for ARCED projects, this command can be used by anyone needing to create organized and consistent folder layouts for data analysis or research projects. Whether you are just starting a project or archiving files at the end of one, `arcedfold` simplifies the process.

Key Features

1. **Automatic Folder Creation:** The command generates a predefined folder structure, ensuring that all necessary directories are in place from the start.
2. **Flexible Use:** `arcedfold` can be run at the beginning of a project to establish the directory layout, or at the end of a project to organize files for archiving.
3. **Consistency Across Projects**: By using this command, users can maintain a consistent file organization standard, making it easier to manage multiple projects over time and to handover a project work.
4. **Automatically Integrates `arceddataflow`:** This command automatically integrates `arceddataflow`, uses and features are explained below.

Benefits of Using `arcedfold`

1. **Efficiency**: Quickly set up or archive projects without manually creating folders.
2. **Organization**: Keep all project files organized in a logical and consistent manner.
3. **Ease of Use**: Simple command syntax makes it accessible even for those new to Stata.

arceddataflow

The `arceddataflow` command in Stata is designed to automate the creation of essential do-files for managing the data flow in your projects. While initially developed for ARCED projects, this command is versatile and can be used by anyone needing a streamlined, organized approach to handling data—from importing it from a server to preparing, cleaning, and generating critical check reports.

Key Features

1. **Automatic Do-File Creation:** The command generates four key do-files: setup.do, import.do, prep.do, and clean.do, covering all critical stages of data processing.
2. **End-to-End Data Management:** Once set up, the dataflow ensures that your project runs smoothly from data import to the generation of detailed, high-frequency check reports.
3. **Consistent Workflow:** By using `arceddataflow`, you maintain a standardized, repeatable workflow across projects, making data management more efficient and reliable.
4. **Flexible Usage:** `arcedfold` is not a prerequisite for `arceddataflow`. This can be installed in any defined folder path. But it is highly recommended to use **arcedfold** for seamless execution of the project.

Benefits of `arceddataflow`

1. **Efficiency:** Automate the creation of essential do-files, saving time and ensuring consistency.
2. **Comprehensive Workflow:** Manage your data from import to final cleaning and validation with a clear, structured process.
3. **Quality Control:** Generate detailed check reports to ensure the integrity of your data, including high-frequency checks and backchecks.
4. **Reproducibility:** Ensure that your analyses can be easily reproduced by maintaining a clear and consistent data flow.

Installation

`arcedfold`

```
* arcedfold can be installed from github

net install arcedfold, all replace ///
from("https://raw.githubusercontent.com/ARCED-Foundation/arcedfold/master")

help arcedfold
```

Syntax

```
arcedfold foldername, path() [options]
arcedsubfold foldername, path() [options]
```

Write desired folder name in place of `foldername.` Write the full path of the directory in path() where you want the folder structure to be created.

Options

Path specifies the path where the folders should be created.

rounds specifies the list of folders to be created inside the 02_DataWork folder. If surveys are not specified, the dataflow will be created inside each round.

surveys specifies the list of folders to be created inside each of the rounds folder. If rounds are not specified, the list of folders will be created inside 02_DataFlow folder and data flow will be created inside each survey.

final specifies that the folders already exist and to create the file list. Not valid for Not valid for `arcedsubfold` subcommand.

author Your name and affiliation.

email Your email address.

Command

```
arcedfold project ABC, path(X:\Projects 2023) author(Mehrab Ali, ARCED Foundation)
email(mehrab.ali@arced.foundation)

arcedfold project ABC, path(X:\Projects 2023) rounds(01_Pilot 02_Baseline)
surveys(Camp Host) author(Mehrab Ali, ARCED Foundation)
email(mehrab.ali@arced.foundation)

arcedsubfold project ABC, path(X:\Projects 2023) rounds(02_Baseline)
surveys(Schools) author(Mehrab Ali, ARCED Foundation)
email(mehrab.ali@arced.foundation)

arcedsubfold project ABC, path(X:\Projects 2023) rounds(03_Endline) surveys(Camp
Host Schools) author(Mehrab Ali, ARCED Foundation)
email(mehrab.ali@arced.foundation)

**# Generate a list of all files and export in an excel file
arcedfold project ABC, path(X:\Projects 2017) final
```

`arceddataflow` Installation

1. Open Stata.
2. Open the .do file.
3. Run the following command

Command

```
net install arceddataflow, all replace ///
from("https://raw.githubusercontent.com/ARCED-Foundation/arceddataflow/master")
```

Syntax

```
arceddataflow, dofiles(string) correction(string)
```

Example Syntax

Copy the following example syntax and adjust the path in the first line to your desired path:

```
arceddataflow, do("C:\Users\Mehrab Ali\Projects\New Project") /// set your folder path
correction("C:\Users\Mehrab Ali\Projects\New Project\Data\Corrections") /// set your path
author("Mehrab Ali, ARCED Foundation") /// set the author name
email("mehrab.ali@arced.foundation") /// your email
project("Project ABC") //// your project name
```

VeraCrypt

VeraCrypt is a free, open-source disk encryption software available for Windows, macOS, and Linux. It adds enhanced security to the algorithms used for system and partition encryption, making it resistant to new developments in brute-force attacks.For download and installation instructions check **Download and Installation of veracrypt** (or here) Guideline provided by the Abdul Latif Jameel Poverty Action Lab (J-PAL). If you cannot find the file from the provided link, you can access it from the **J-PAL** website.

SurveyCTO Desktop

SurveyCTO Desktop is a tool within the SurveyCTO platform used for data collection, management, and analysis. SurveyCTO is widely employed for conducting surveys, research studies, and other data-driven projects, particularly in fields such as international development, public health, and social sciences.

Download

To download SurveyCTO Desktop, follow these steps:

1. Visit the SurveyCTO Desktop download page.
2. On this page, navigate to the "Installing and using SurveyCTO Desktop" section, where you will find the download options.
3. Click on the download option for Windows (Universals). This action will redirect you to the download page.
4. Download the file for SurveyCTO Desktop. Once the download is complete, the installation will start automatically.

Description of Folder Tree

The **`arcedfold`** command creates the following folders in the destination. For detailed description of each of the folders can be found here or here.

```
01_Admin
 |-- 01_Management
          |-- 01_Design
          |-- 02_Project_log
          |-- 03_Gantt_Chart
          |-- 04_Hiring_and_Onboarding
 |-- 02_Funding
          |-- 01_Proposal
          |-- 02_Agreement
          |-- 03_Reporting
          |-- 04_Invoice
          |-- 05_Deliverables
 |-- 03_Budget
 |-- 04_IRB
          |-- 01_Applications
          |-- 02_Amendments
          |-- 03_Approvals
          |-- 04_Certificates
 |-- 05_Communications
          |-- 01_Letters_and_Permissions
          |-- 02_Email
          |-- 03_Reports_and_Presentations
          |-- 04_Outreach
          |-- 05_Meeting_Minutes
02_DataFlow
 |-- Round
          |-- Survey
                    |-- 01_Instruments
                              |-- 01_Paper
                              |-- 02_XLSForm
                                        |-- 01_Attachments
                    |-- 02_Codes
                              |-- 01_Ado
                    |-- 03_Data
                              |-- 01_Samples
                              |-- 02_Raw
                              |-- 03_Corrections
                              |-- 04_Intermediate
                              |-- 05_Clean
                    |-- 04_Output
                              |-- 01_Logs
                              |-- 02_Checks
                              |-- 03_Analysis
                                        |-- 01_Table
                                        |-- 02_Figure
                                        |-- 03_Illustration
                                        |-- 04_Report
                                        |-- 05_Presenatation
03_FieldWork
 |-- Round
          |-- Survey
                    |-- 01_Protocol_and_Manual
                    |-- 02_Field_Materials
                    |-- 03_Training_Materials
                    |-- 04_Field_Team_Hiring
```

Description of Dataflow's DO-files

- **Master:** The master.do file is a one stop to run all the DO-files mentioned below. As a RA, you should run only this do file everyday to run the entire dataflow.
- **Setup:** The setup.do file configures the environment, including setting global macros and paths that will be used throughout the project. You should prepare this file once when you set up your dataflow. You can exercise it in the Test Pack.
- **Import:** The import.do file handles the importation of raw data directly from your server into Stata, ensuring that all data files are correctly loaded and stored in a structured format.the data usable for every feature we have in this dataflow e,g. `odksplit`, `encryption` etc. You do not need to edit this file.
- **Preparation:** The prep.do file takes over to prepare the imported (deidentified) data. This might include merging datasets, transforming variables, or reshaping data as necessary and to make manual corrections.
    - The corrections are done using an excel file stored in `02_Dataflow/03_Data/03_Corrections/Correction_sheet.xlsx`
    - All the corrections are logged in log file `02_Dataflow/04_Output/01_Logs/Correction_log_ `date'.xlsx` that is saved everyday (with date in suffix) the dataflow is run.
    - The raw data is stored in the container in `02_Dataflow/03_Data/02_Raw/rawdata`
    - The deidentified(excluding all the PIIs) raw data is saved in `02_Dataflow/03_Data/04_Intermediate/`filename`.dt`
- **Cleaning:** Finally, the clean.do file focuses on cleaning the data, identifying and addressing inconsistencies, missing values, and outliers. This stage also includes generating a check report, which covers critical aspects such as high-frequency checks, backchecks, and other data validation processes.
    - The cleaned data is stored in `02_Dataflow/03_Data/05_Clean/`filename`_clean.dta`

Customizing Do-Files

While the generated do-files provide a robust framework, you can customize them to suit the specific needs of your project. This might involve adding additional data processing steps or modifying existing ones. Other than that you will only be needing this customizations given below:

1. In setup.do, subject to the project variable naming, XLSX sheet naming, setting switches etc are needed.

2. In prep.do,for any additional data processing or preparation **e.g**. generating variables, ID generation, sorting & ordering, dropping unnecessary variables, media file naming, merging & appending with other dataset etc you can write your codes under the subheading "Additional preparation". In a nutshell, anything and everything you want to do with data which has not already been done with these 4 DO-files instead of creating a separate DO-file you write the codes in prep.do.

   For instance, if you have media files such as images in a unencrypted dataset then you can do the naming of those files using this following Stata code:

Syntax

```
sctomedia varname [if] [in], [by(varlist)] id(varname) vars(varlist)
media(folder_path) output(folder_path) resolve(varname)

Example:
**# Additional preparation
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

**#Naming respondents' photos
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

sctomedia photo, id(uid_survey) vars(district vil) by(respondentid)
media("../03_Data/02_Raw/media") output("../03_Data/02_Raw/media")
```

   For encrypted dataset how to unlock the container and do the naming that is shown in Advanced Guideline.

3. In checks.do,we usually don't do any customization but if you need to add an extra layer of check for data quality management you may add it in this DO-file. You can find examples in the Advanced Guideline.

Setting up Dataflow

Dataflow Practice

Now that you know what the components of `arcedfold` and `arceddataflow` are, it's time for some hands-on practice. By this point, you should have Stata, Veracrypt and SurveyCTO Desktop downloaded and installed on your desktop. Otherwise, install that and then come back to this section.

Done with the installation? Now, let's tell you about our sample project.

ARCED Foundation is going to conduct a baseline survey of brick kiln managers and owners all across the country and collect data on brick production, brick price, production cost, brick sales, kiln technology etc. There will be three rounds of survey: ***baseline, midline and endline****.*

Typically, you will be the one who programmes such a form and will have a clear idea about the survey beforehand. For the time being, however, here is the link to the surveyCTO xlsform for the manager survey: Dataflow Practice. Go through the form to get an idea of the survey.

The purpose of this exercise is to download the data for this form from the surveyCTO server, clean the data [make corrections, separate personally identifiable information (PII)], run high frequency checks (enumerator progress, performance, time use, outliers, date check, form version conflict, text audit etc.) and prepare check report.

Before any of that, the first thing you need to do is create the project folder.

Creating Project Folder

Step-01: Open Stata

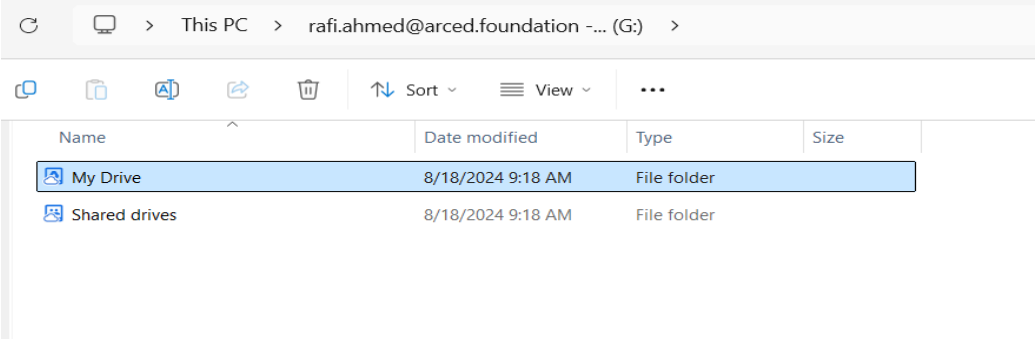
Step-02: Install `arcedfold`

In the command window, write the following command

```
net install arcedfold, all replace
from("https://raw.githubusercontent.com/ARCED-Foundation/arcedfold/master")
```

Step-03: Creating Project Folder using `arcedfold`

Now we want to create the project folder. To do so, go to the folder in which you want to create the project folder. Select the folder and copy the path (Ctrl+Shift+C). Then, go back to the Stata command window.



Write the following command in the command box

```
arcedfold [Project Name], path([The path you copied]) rounds([Name of separate
folders for each round]) surveys([Name of separate surveys in each round])
author([Name of author/RA]) email([your email)
```

For example, the practice project is on brick kilns. So, we want to create a project folder named Brick Kiln. It has three (03) rounds: Baseline, Midline and Endline. There are two surveys in each round: Manager Survey and Owner Survey. The main RA of the project is Mehrab Ali from ARCED Foundation and his email address is mehrab.ali@arced.foundation. In such a case, the command would look like this:

```
arcedfold Brick Kiln, path("G:\My Drive") rounds(01_Baseline 02_Midline 03_Endline)
surveys(Manager Owner) author(Mehrab Ali, ARCED Foundation)
email(mehrab.ali@arced.foundation)
```

The output for the command would look like this.

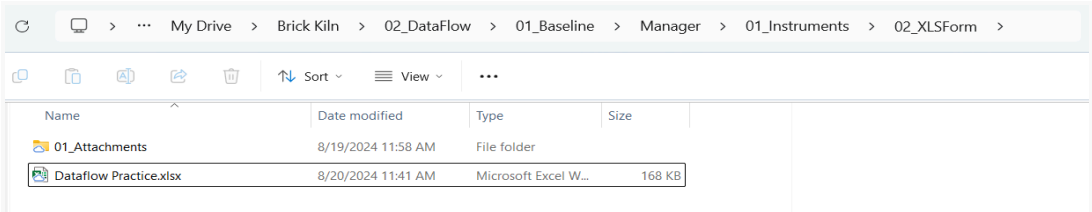
```
. arcedfold Brick Kiln, path("G:\My Drive") rounds(01_Baseline 02_
> Midline 03_Endline) surveys(Manager Owner) author(Mehrab Ali, AR
> CED Foundation) email(mehrab.ali@arced.foundation)
note: ado update updates community-contributed files; type
      update to check for updates to official Stata.

Checking status of specified packages:

   [30] arceddataflow at
        https://raw.githubusercontent.com/ARCED-Foundation/arced
        > dataflow/master:
        installed package is up to date

(no packages require updating)
All the folders are created. To browse click here:  Brick Kiln
```

Step-5: Learn the Folder Structure

Click on the Brick Kiln (or your preferred Project Name) folder. You can take a look around and corroborate your learning of the folder structure. Close Stata.

A short exercise:

Download the surveyCTO xlsform for the manager survey: Dataflow Practice. Save it in `~\Brick Kiln\02_DataFlow\01_Baseline\Manager\01_Instruments\02_XLSForm`



Setting up Dataflow

Step-01: Go to the Program Folder

Once you are done playing around with the folders, go to this folder: ~\Brick Kiln\02_DataFlow\01_Baseline\Manager\02_Codes.

Since the exercise survey is a baseline survey of managers, this 02_Codes folder is where we do most of our work associated with dataflow.

Step-02: Open *00_master.do*

Open the `00_master.do` file from the folder. Remember to open the do file by double-clicking on it.

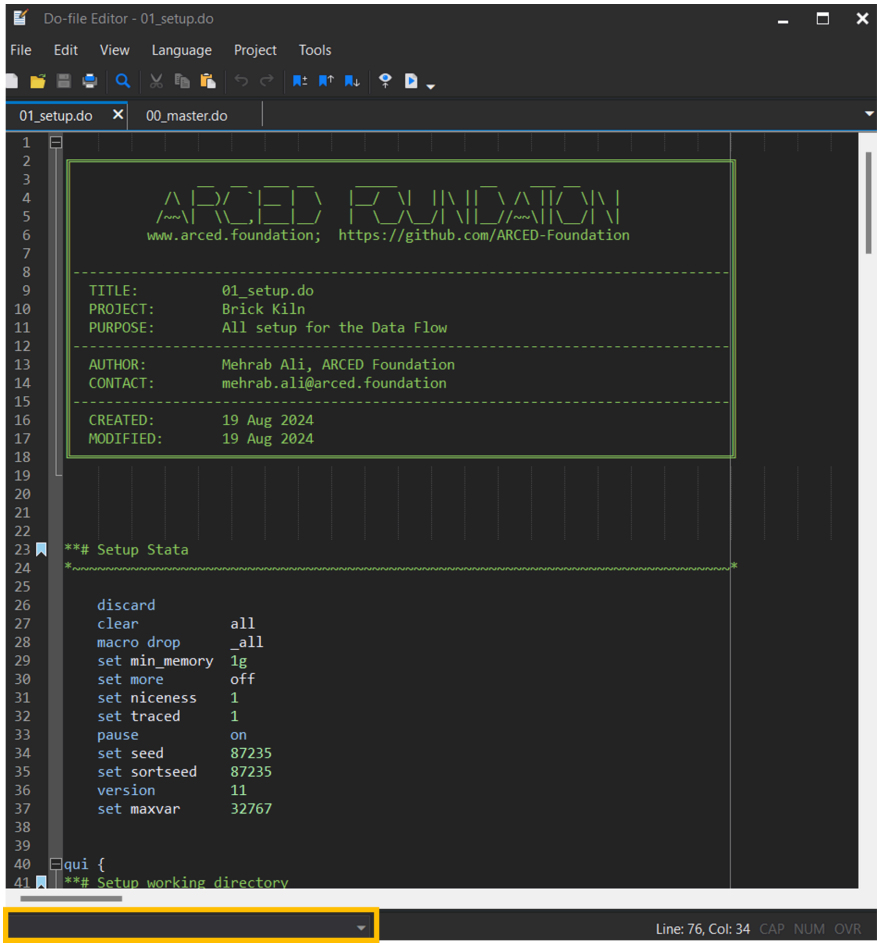
```
**# Run do files
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    **# 01_setup.do
    /*----------------------------------------------------

        01_setup.do congiures the Stata environment
        and set necessary globals for the data flow.

    ----------------------------------------------------*/

    do "01_setup.do"


    **# 02_import.do
    /*----------------------------------------------------

        02_import.do does data download, encryption,
        PII splitting, data labeling.

    ----------------------------------------------------*/

    do "02_import.do"


    **# 03_prep.do
    /*----------------------------------------------------

        03_prep.do includes all the preparatory works
        for data, data labeling, variable renaming,
        manual data corrections and basic cleaning.

    ----------------------------------------------------*/

    do "03_prep.do"


    **# 04_checks.do
    /*----------------------------------------------------

        04_checks.do includes all the logical checks
        and data quality checks.

    ----------------------------------------------------*/

    do "04_checks.do"



**# End
```

You will see that the do file contains commands to run four (04) different do files and their description: `01_setup.do`, `02_import.do`, `03_prep.do`, `04_checks.do`

You have already learnt about the function of these do files. If you need to take a look again, go to

Step-03: **Open *01_setup.do***

Remember to open the do file by double-clicking on it.

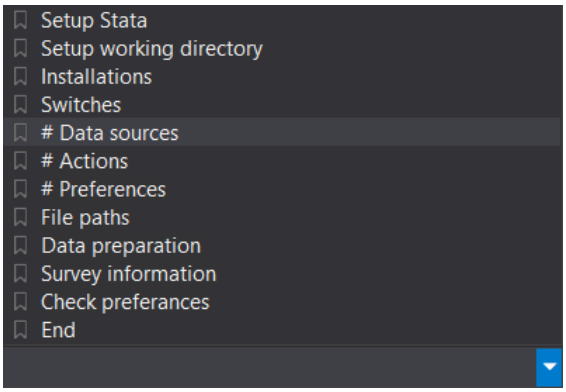


The `01_setup.do` is the most important do file for an RA. As you guessed it, its job is to set up the parameters for the entire dataflow.

In this do file you will show Stata the directories to save the, downloaded, de-identified, clean data, how you want to download these datasets (API, Manual Download), how you want to label the datasets (using *odksplit* or *sctoimport.do*), whether you want your raw data

to be encrypted, your target sample size, your parameters for an outlier, PII variables that should be excluded from the clean dataset etc.

For navigation through the do file, click on the yellow rectangle. The following list of bookmarks will appear. You can click on any of them to navigate through the do file.



Step-04: Switches

Click on Switches bookmark. It will take you to the following section of the do file.

```
**# Switches
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

    **# Data sources
    *--------------------------

    /*---------------------------------------------------------------

        The sctodownload and odkdownload prompts for the username
        and password. If you want not to type the username and
        password on the command window, you can setup profile.do
        to set the username and password. Read more:
        https://www.stata.com/support/faqs/programming/profile-do-file
        https://www.techtips.surveydesign.com.au/post/the-profile-and-sysprofile-do-files-automating-your-stata-start-up

        In the profile.do set the following globals:

        * SurveyCTO credentials
        gl suser = "xxxxx"
        gl spass = "yyyyy"

        * ODK credentials
        gl Ouser = "xxxxx"
        gl Opass = "yyyyy"

    ---------------------------------------------------------------*/
```

Here, you can see a lot of global macros. These global macros need to be changed to properly set up the dataflow. If you don't know what macros are, click here to learn more.

Step-05: Setting up Data Download

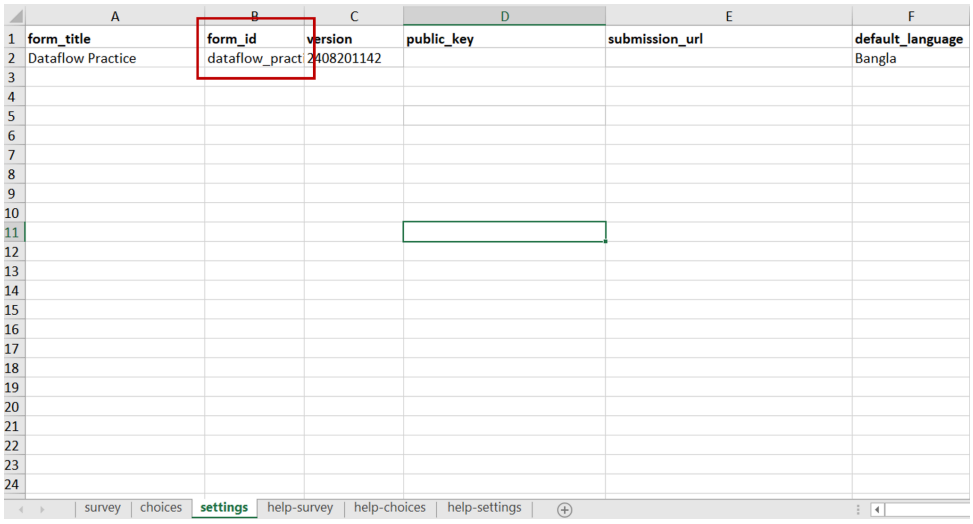
Under the Switches bookmark, The first few global macros mainly determine how you want to download the raw data from the survey. More precisely, do you want to download using surveyCTO API, ODK server API or do you want to download manually using SurveyCTO Desktop.

```
gl  sctodownload            0       // Download data from SurveyCTO
    * Globals for SurveyCTO api download
    *-----------------------------------
    gl formid           "ssc_bd_23"
    gl sctodataloc      "X:"
    gl timeshift        "6"             /*  When data downloaded through API, the time is UTC,
                                            so for Bangladesh the default is UTC+6 to shift to local time */

gl  odkdownload             1       // Download data from ARCED ODK server
    * Globals for ARCED ODK server api download
    *-------------------------------------------
    gl OData        "https://sotlab.eastus.cloudapp.azure.com/v1/projects/17/forms/skills_for_growth_listing.svc"
    gl odkapi       `"`=regexr("$OData", ".svc", "/submissions.csv.zip")'"'


gl  manualdownload          0       // Download data manually
    * Globals for manual download
    *----------------------------
    gl  sctodesktoploc  "C:\Users\\`=c(username)'\AppData\local\Programs\SurveyCTODesktop\SurveyCTO Desktop.exe"
```

SurveyCTO API Download

If you want the data to be downloaded automatically using surveyCTO API, change the value of the global macro *sctodownload* to 1.

Go to the surveyCTO xlsform you saved earlier and go to the settings tab. Copy the form_id (dataflow_practice) shown within the red rectangle.



Replace the global formid with what you just copied. Then set global macros *odkdownload* and *manualdownload* to 0.

```
gl  sctodownload            1       // Download data from SurveyCTO
    * Globals for SurveyCTO api download
    *----------------------------------
    gl formid           "dataflow_practice"
    gl sctodataloc      "X:"
    gl timeshift        "6"         /*  When data downloaded through API, the time is UTC,
                                        so for Bangladesh the default is UTC+6 to shift to local time *

gl  odkdownload             0       // Download data from ARCED ODK server
    * Globals for ARCED ODK server api download
    *-----------------------------------------
    gl OData        "https://sotlab.eastus.cloudapp.azure.com/v1/projects/17/forms/skills_for_growth_li
    gl odkapi       `"`=regexr("$OData", ".svc", "/submissions.csv.zip")'"'


gl  manualdownload          0       // Download data manually
    * Globals for manual download
    *---------------------------
    gl  sctodesktoploc  "C:\Users\\`=c(username)'\AppData\local\Programs\SurveyCTODesktop\SurveyCTO Des
```

At this point, your do file should look like this.

Since the exercise survey is based on SurveyCTO, we will ignore the odk download part of the setup for now. But what if you face some problem with the SurveyCTO API and want to download data manually using SurveyCTO desktop?

Manual Download

For manual download simply set the global macros *sctodownload* and *odkdownload* to 0 and *manualdownload* to 1.

```
gl  sctodownload            0       // Download data from SurveyCTO
    * Globals for SurveyCTO api download
    *----------------------------------
    gl formid           "ssc_bd_23"
    gl sctodataloc      "X:"
    gl timeshift        "6"         /*  When data downloaded through API, the time is UTC,
                                        so for Bangladesh the default is UTC+6 to shift to local time *

gl  odkdownload             0       // Download data from ARCED ODK server
    * Globals for ARCED ODK server api download
    *-----------------------------------------
    gl OData        "https://sotlab.eastus.cloudapp.azure.com/v1/projects/17/forms/skills_for_growth_li
    gl odkapi       `"`=regexr("$OData", ".svc", "/submissions.csv.zip")'"'

gl  manualdownload          1       // Download data manually
    * Globals for manual download
    *---------------------------
    gl  sctodesktoploc  "C:\Users\\`=c(username)'\AppData\local\Programs\SurveyCTODesktop\SurveyCTO Des
```

Step-06: Setting up Data Correction and Labeling

```
**# Actions
*--------------------------
gl  data_correction         1
gl  pii_correction          0
gl  odksplit                1
    gl  language        "English"
    gl  sctoimportdo    "02b_datalabel.do"
```

Given we almost always want corrections to be made in our data, the global macro data_correction is always set to 1. If you want to make corrections to personally identifiable information, you should set the global macro pii_correction to 1 as well. But for this exercise and in most cases, this should be set to 0.

odksplit is a Stata command developed by ARCED Foundation that is used for labeling variables, values, datasets etc. after importing the rawdata. Typically, the global *odksplit should be set to 1.*

Most xlsforms you make will have two languages: Bangla and English. However, when we analyze the data, we typically want the English labels. That's why the global *language* is typically set to English.

Step-07: Setting File Paths

Now we need to show Stata where the relevant files (datasets, xlsforms etc.) are or should be located.

```
**# File paths
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

    /*------------------------------------------------------------------------

        container       The name and path of the encrypted container.
                        If the mentioned container does not exist, the
                        program will notify and automatically create the
                        container using VeraCrypt.

        rawdata         The name of the raw csv data. The path is by default
                        X:/ because the encrypted container will be mounted
                        on X:/ drive. Don't change that.

        deidentified    The name and path of the deidentified dataset. This
                        dataset will not be encrypted.

    ------------------------------------------------------------------------*/

    ** Data files
    gl  rawpath             "${cwd}/../03_Data/02_Raw"
    gl  container           "${cwd}/../03_Data/02_Raw/rawdata"
    gl  rawdata             "X:/Social Contact Survey 2023_WIDE.csv"
    gl  rawdatadta      `"`=regexr("${rawdata}", ".csv", ".dta")'"'
    gl  mediafolder         "X:/media"

    gl  deidentified        "${cwd}/../03_Data/04_Intermediate/Social Contact Survey 2023.dta"
    gl  cleandata           "${cwd}/../03_Data/05_Clean/Social Contact Survey 2023_clean.dta"
    gl  textauditdata       "${cwd}/../03_Data/02_Raw/Text_audit_data.dta"
    gl  commentsdata        "${cwd}/../03_Data/02_Raw/Comments_data.dta"


    ** Correction files
    gl  correctionsheet     "${cwd}/../03_Data/03_Corrections/Correction_sheet.xlsx"
    gl  pii_correction_file "X:/pii_correction.xlsx"
    gl  correction_log      "${cwd}/../04_Output/01_Logs/Correction_log_`c(current_date)'.xlsx"

    ** Outfile
    gl  outfile_hfc         "${cwd}/../04_Output/02_Checks/Check_report.xlsx"

    **  audio_audit folder
    gl  audit_folder        "../../../03_FieldWork/02_Phone_Call/05_Audio_audit"
```

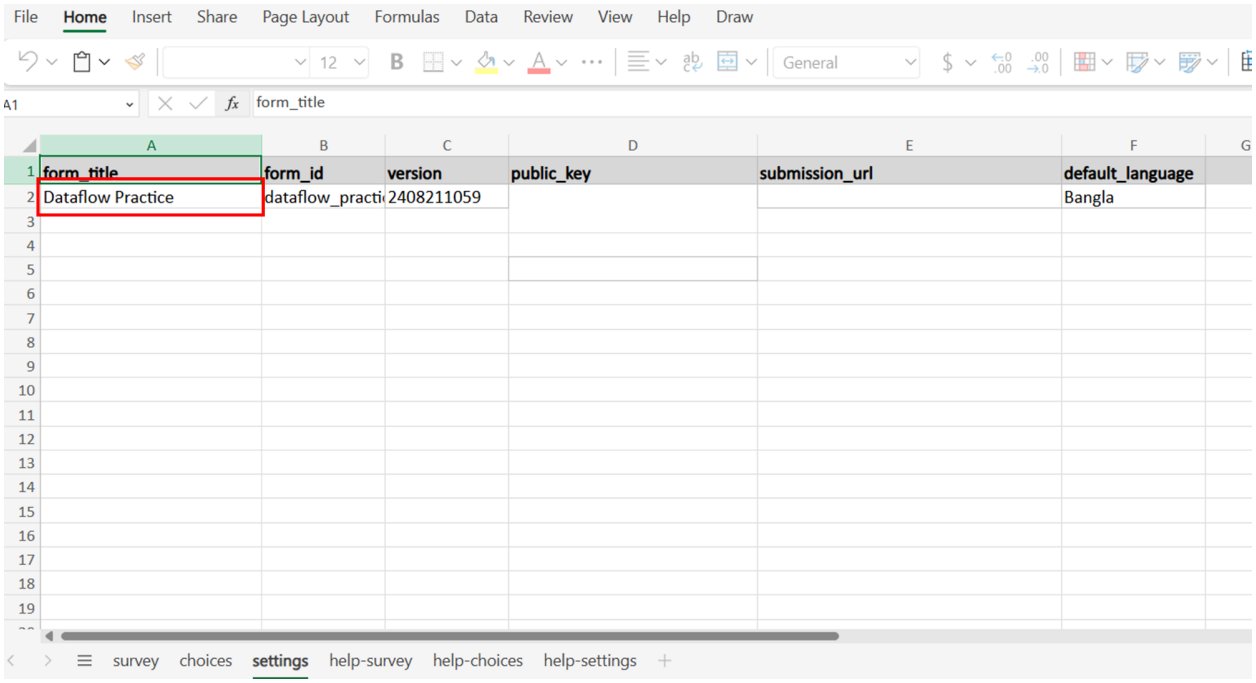
Leave the globals *rawpath* and *container* as it is.

The rawpath is similar to what we learned in the section before and the container is created by Veracrypt (an encryption software you should already have installed). The container is password protected. The container is also called the X drive. You may use other letters as well. But as convention, we always use X drive to denote the container.

We will talk about creating the container in greater detail soon.

In the global rawdata, Social Contact Survey 2023 is the sample form title and _WIDE is used to represent the format of the data. In Wide datasets, one variable is enough to uniquely identify the data.

Go to the Dataflow Practice form and copy the form title. The form title is shown in the red rectangle below.



Replace the previous form title (Social Contact Survey 2023) with the previously copied form title (Dataflow Practice). Keep the suffix _WIDE. At this point, your do file should look like this.

```
** Data files
gl  rawpath          "${cwd}/../03_Data/02_Raw"
gl  container        "${cwd}/../03_Data/02_Raw/rawdata"
gl  rawdata          "X:/Dataflow Practice_WIDE.csv"
gl  rawdatadta       `"`=regexr("${rawdata}", ".csv", ".dta")'"'
gl  mediafolder      "X:/media"
```

Leave the globals *rawdatadta* and *mediafolder*. The global rawdatadta simply replaces the .csv extension at the end of your downloaded rawdata and copies it as a .dta file. The global *mediafolder* just shows that the text_audit and audio_audit would be saved in a folder named media within the X drive.

Replace the copied form title in globals *deidentified* and *cleandata. Keep the suffix _clean.* At this point the do file should look like this.

```
** Data files
gl  rawpath             "${cwd}/../03_Data/02_Raw"
gl  container           "${cwd}/../03_Data/02_Raw/rawdata"
gl  rawdata             "X:/Dataflow Practice_WIDE.csv"
gl  rawdatadta          `"`=regexr("${rawdata}", ".csv", ".dta")'"'
gl  mediafolder         "X:/media"

gl  deidentified        "${cwd}/../03_Data/04_Intermediate/Dataflow Practice.dta"
gl  cleandata           "${cwd}/../03_Data/05_Clean/Dataflow Practice_clean.dta"
gl  textauditdata       "${cwd}/../03_Data/02_Raw/Text_audit_data.dta"
gl  commentsdata        "${cwd}/../03_Data/02_Raw/Comments_data.dta"
```

Leave all of the global macros under the subsection Correction files. Outfile and audit_folder.

Correction files: The global *correctionsheet* shows the directory of the Correction Sheet. The function of the Correction sheet is explained here.

If you make corrections to PIIs, you will use the pii_correction.xlsx file. The global pii_correction_file shows where its located.

The global *correction_log* shows where the Correction Log is located. The Correction Log basically keeps a record of every change made to the data using the Correction Sheet, every day such changes are made.

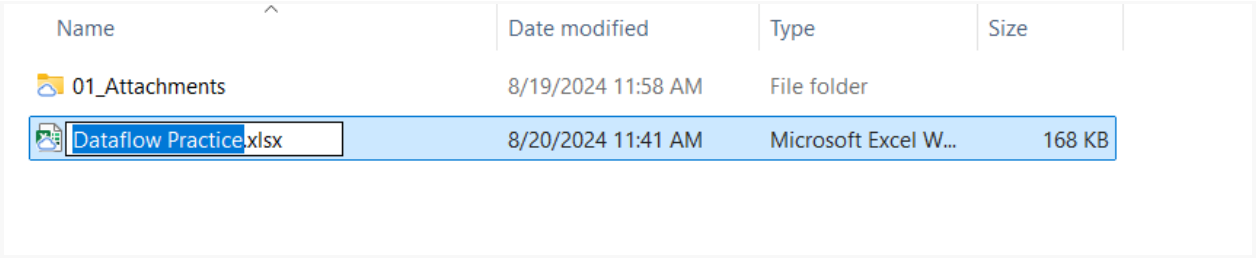
Note: `c(current_date)' refers to the date of the day on which a do file is run.

Step-08: Data Preparation

The first global *xlsform* is used to set the directory of the xlsform. You have saved the xlsform before in the following folder:

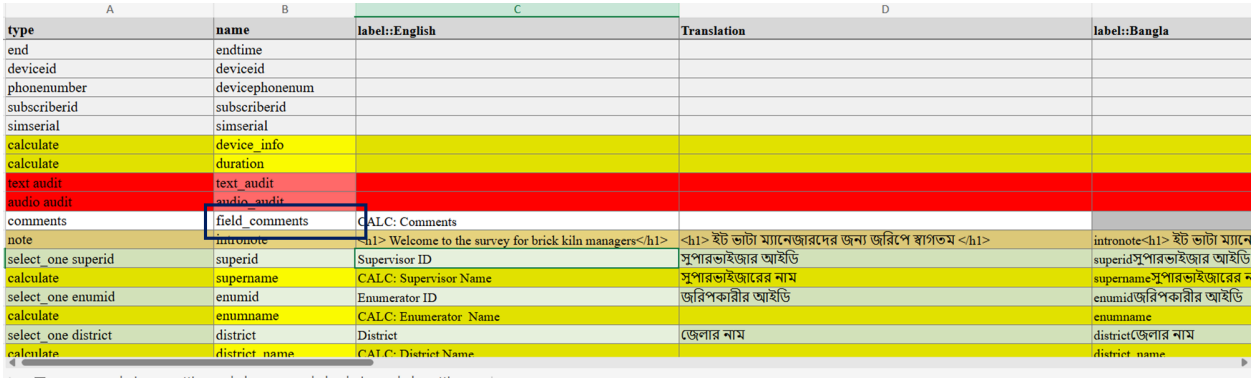
```
~\Brick Kiln\02_DataFlow\01_Baseline\Manager\01_Instruments\02_XLSForm
```

Go to this folder and copy the name of the file and replace it in the global..



Leave the globals *media*, *text_audit*. Make sure the xlsform has named the audio audit and text audit variables similarly (as audio_audit and text_audit). Otherwise, you will have to edit these.

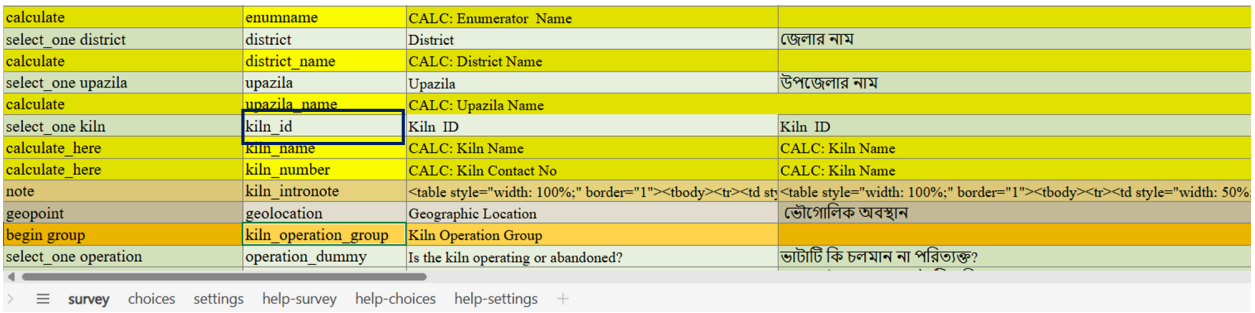
Global *sctocomments* refers to the variable in the xlsform used to capture comments made using the SurveyCTO comments feature. Go to the practice xlsform and find the field_comments variable. Copy the name of the variable and place it in the global.



type	name	label::English	Translation	label::Bangla
end	endtime			
deviceid	deviceid			
phonenumber	devicephonenum			
subscriberid	subscriberid			
simserial	simserial			
calculate	device_info			
calculate	duration			
text audit	text_audit			
audio audit	audio_audit			
comments	field_comments	CALC: Comments		
note	intronote	Welcome to the survey for brick kiln managers	ইট ভাটা ম্যানেজারদের জন্য জরিপে স্বাগতম	intronote<h1> ইট ভাটা ম্যানে
select_one superid	superid	Supervisor ID	সুপারভাইজার আইডি	superidসুপারভাইজার আইডি
calculate	supername	CALC: Supervisor Name	সুপারভাইজারের নাম	supernameসুপারভাইজারের ন
select_one enumid	enumid	Enumerator ID	জরিপকারীর আইডি	enumidজরিপকারীর আইডি
calculate	enumname	CALC: Enumerator Name		enumname
select_one district	district	District	জেলার নাম	districtজেলার নাম
calculate	district_name	CALC: District Name		district_name

The global *uid* refers to the unique identifier of each dataset. Almost always it is going to be the variable key as it contains a unique identifier for each submission made to surveyCTO.

The global *sid* contains the unique identifier in the sample that is being surveyed. For instance, for a household survey that would be the household ID. In the practice survey on managers, it is the id of the kilns.



calculate	enumname	CALC: Enumerator Name	
select_one district	district	District	জেলার নাম
calculate	district_name	CALC: District Name	
select_one upazila	upazila	Upazila	উপজেলার নাম
calculate	upazila_name	CALC: Upazila Name	
select_one kiln	kiln_id	Kiln ID	Kiln ID
calculate_here	kiln_name	CALC: Kiln Name	CALC: Kiln Name
calculate_here	kiln_number	CALC: Kiln Contact No	CALC: Kiln Name
note	kiln_intronote	<table style="width: 100%;" border="1"><tbody><tr><td st	<table style="width: 100%;" border="1"><tbody><tr><td style="width: 50%
geopoint	geolocation	Geographic Location	ভৌগোলিক অবস্থান
begin group	kiln_operation_group	Kiln Operation Group	
select_one operation	operation_dummy	Is the kiln operating or abandoned?	ভাটাটি কি চলমান না পরিত্যক্ত?

survey choices settings help-survey help-choices help-settings

Find the id variable of the kilns and paste it in the global *sid*.

Finally, you have to find the PII variables from the xlsform. And list them under the global PIIs. Try to do this on your own and check with the results below.

At this point, your do file should look like this.

```
gl  xlsform             "${cwd}/../01_Instruments/02_XLSForm/Dataflow Practice.xlsx"
gl  media               "audio_audit text_audit"
gl  text_audit          "text_audit"
gl  sctocomments        "field_comments"
gl  uid                 "key"
gl  sid                 "kiln_id"

                        #d ;
gl  PIIs                "    kiln_name kiln_number geolocation*
                             name phone_num kiln_name_txt owner_name
                             owner_phone_num bkash_num
                        " ;
                        #d cr
```

Step-09: Survey Information

Here is a brief description of all of the globals.

global	Description
surveystart	Start Date of the survey. For this survey
targetsample	Target Sample Size
startdate	*startdate*: Variable that records the starting date of each submission
starttime	*starttime*: Variable that records the starting time and date of each submission.
enddate	*enddate*: Variable that records the end date of each submission
endtime	*endtime*: Variable that records the ending time and date of each submission.
duration	*duration*: Duration in minutes for each submission
consent	*Consent*: Variable that contains whether a respondent contented to the survey or not
enumid	Variable that contains Enumerator ID, typically written as *enumid*
enumname	Name of the Enumeratortypically written as *enumname*
dk	Value of the option "don't know" in xlsform. Typically, it should be -99. This should be consistent across the form. If it's -99 for one question, it should be the case for all questions.

ref	Value of the option "refused" in xlsform. Typically, it should be -98. This should be consistent across the form. If it's -98 for one question, it should be the case for all questions.
skip	Value of the option "skipped" in xlsform. Typically, it should be -97. This should be consistent across the form. If it's -97 for one question, it should be the case for all questions.
NA	Value of the option "Not Applicable" in xlsform. Typically, it should be -97. This should be consistent across the form. If it's -95 for one question, it should be the case for all questions.
other	Value of the option "Others (Please, Specify)" in the xlsform. Typically, it should be -96. This should be consistent across the form. If it's -96 for one question, it should be the case for all questions.
enumcomments	Variable that contains enumerator comments.

Exercise: Given that the survey started on August 18, 2024 with a target sample size of 692, fill out all of the globals in your do file and check the results.

Once you are done, your do file should look like this.

```
gl   surveystart          "18aug2024"
gl   targetsample         "692"

gl   startdate            "startdate"
gl   starttime            "starttime"
gl   enddate              "enddate"
gl   endtime              "endtime"
gl   duration             "duration"

gl   consent              "consent"
gl   enumid               "enumid"
gl   enumname             "enumname"

gl   dk                   "-99"
gl   ref                  "-98"
gl   skip                 "-97"
gl   NA                   "-95"
gl   other                "-96"

gl   enumcomments         "enum_comment"
```

Step-10: Check Preferences

Leave the globals *otherdupvars*, *outkeepvars and comboutvars* blank for the time being.

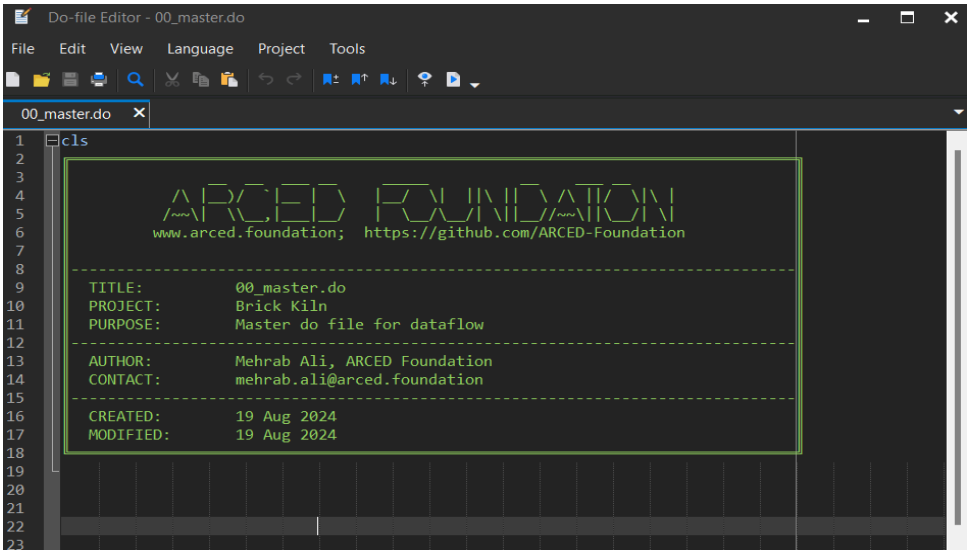
Outliers are defined as observations that fall outside two or three standard deviations of the mean. Some PIs choose 2 while others choose 3. For this example, let's keep it as 3. So, set the global *sd* to 3.
Global outexclude includes numeric variables that should be excluded from outlier considerations. This may include numeric phone numbers. Otherwise, leave this empty.

Finally, the global missper shows the percentage of missing values in each variable for each enumerator that we can allow. The standard convention is to allow up to 70%. This part of your do file should look something like this.

```
**# Check preferances
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*

    * Duplicates from other variables
    gl  otherdupvars        ""

    * Outliers
    gl  outkeepvars         ""
    gl  sd                  3
    gl  comboutvars         ""

                            #d ;
    gl  outexclude          "
                            " ;
                            #d cr

    * Missing max %
    gl  missper             "70"
```

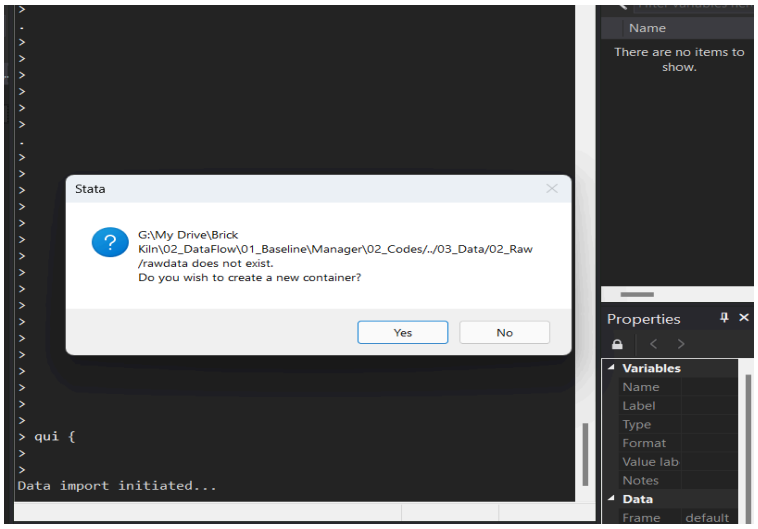
Running the Dataflow

Step-01: Open the 00_master.do do file by double clicking on it.



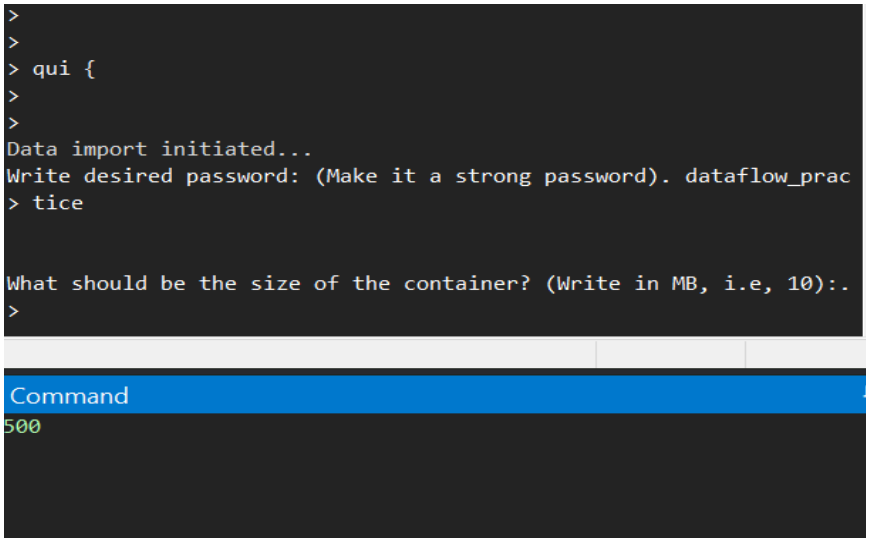
Step-02: Run it (Ctrl+D)

Step-03: If you have set up everything, the following will pop-up on your result window. Click Yes.

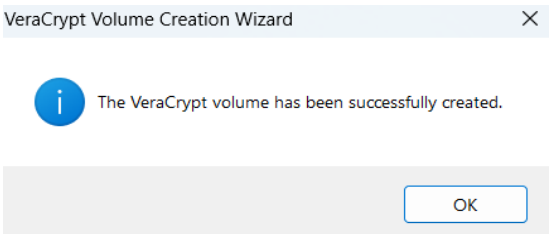


Step-04: Enter a strong password in your command window. Also, share this password with the Executive Director. Do not write this password elsewhere.

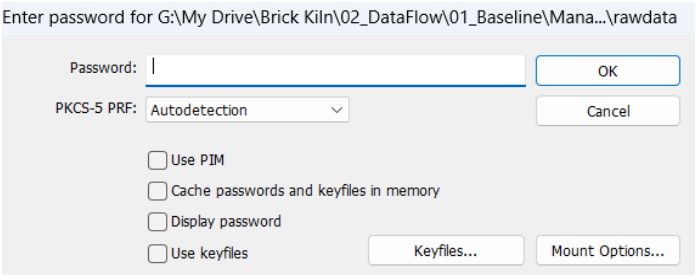
Step-05: Enter your preferred size of the container. Typically we use 500 MB.



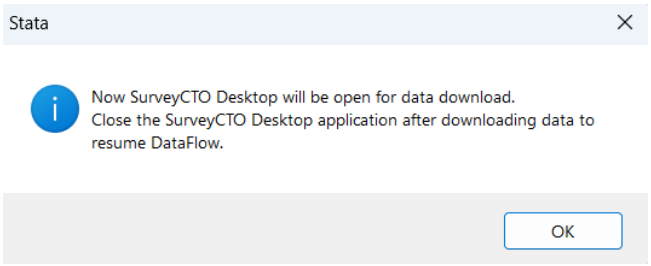
Step-06: Once you enter, Veracrypt will take some time to create the folder and then you will see the following pop-up on screen. Click OK.



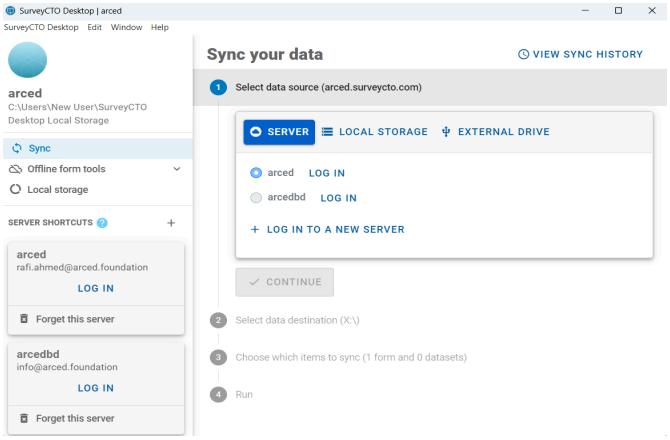
Step-07: The following screen will pop up. Enter the password you created to open the container or X drive.



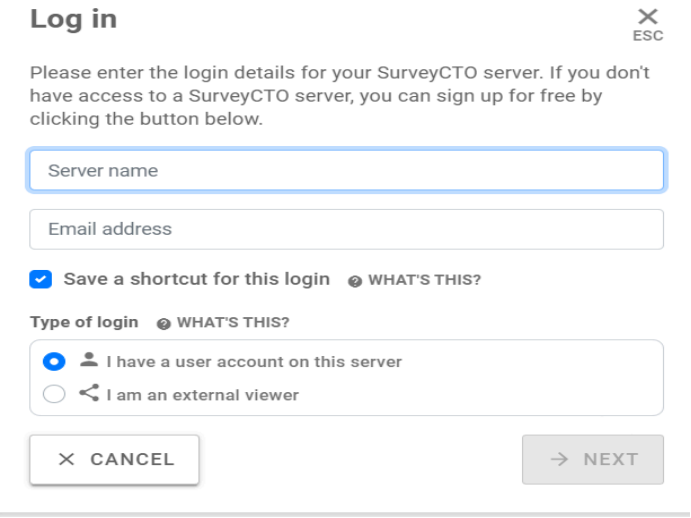
Step-08: For the time being we are using SurveyCTO Desktop as we had set the global manualdownload to 1. So, this should pop-up. Click OK.



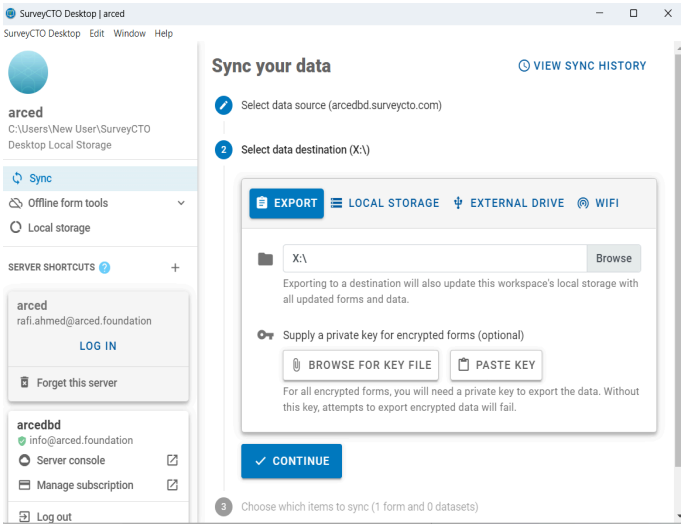
Step-09: This screen will pop-up. If you are a new user, Click on "LOG INTO A NEW SERVER". Otherwise, click on "LOG IN" next to a pre-loaded server (Here arced and arcedbd are pre-loaded).



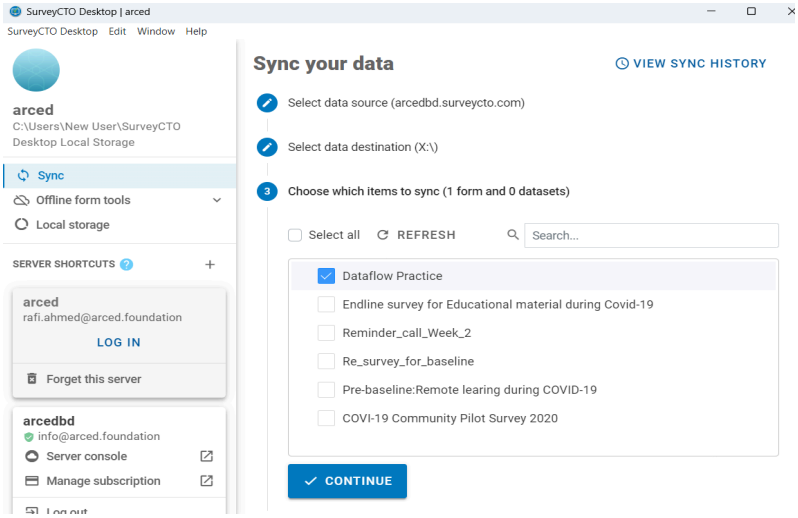
Use the following credentials to log in to surveyCTO Desktop.
Email: info@arced.foundation
Server: arcedbd
Pass: Arced12345@



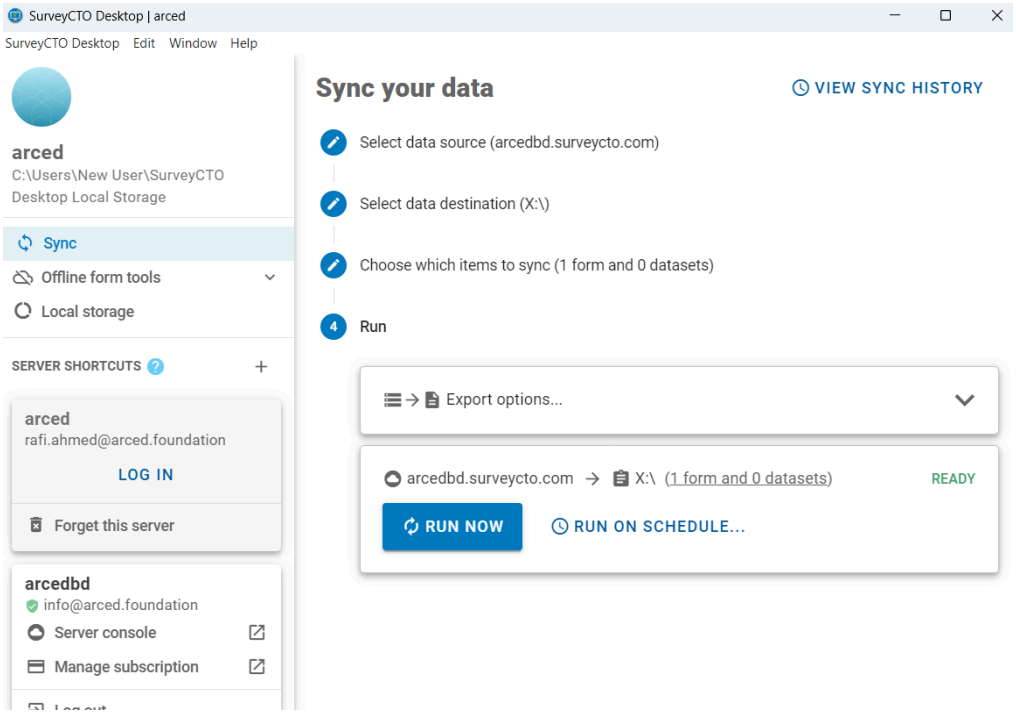
Step-10: Once you've logged in, the following screen will appear. Click on "CONTINUE".



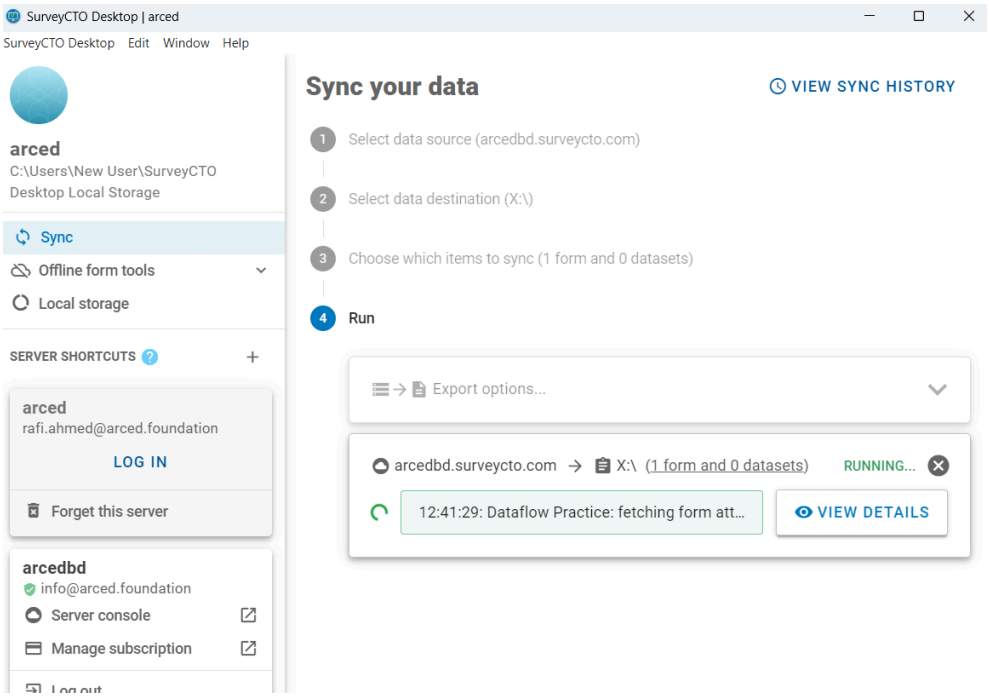
Step-11: Select the form "Dataflow Practice". Click "CONTINUE".



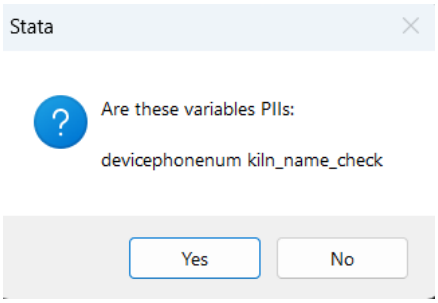
Step-12: Click on "RUN NOW"



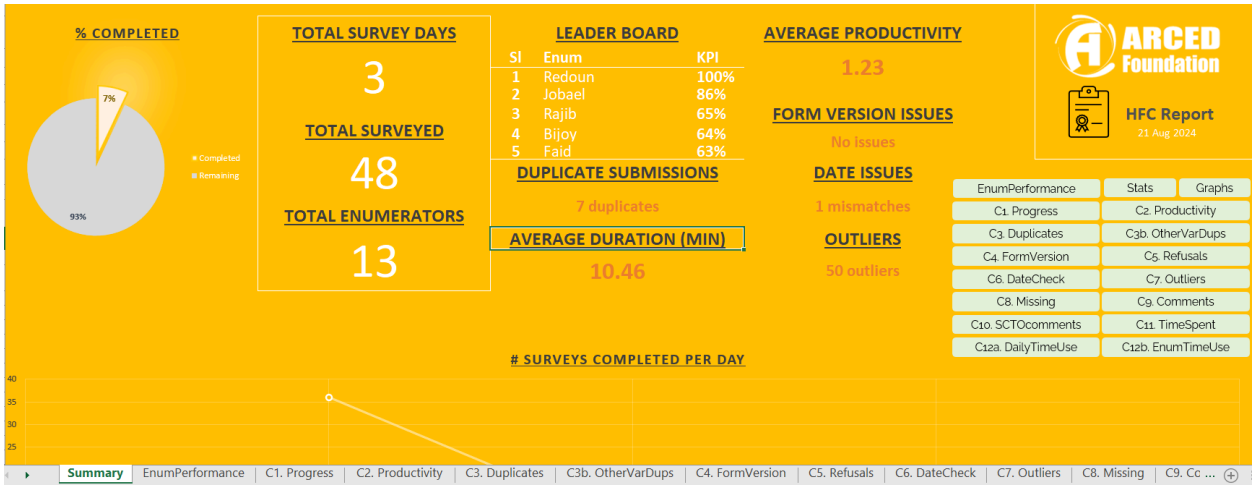
Then, wait for the data to be downloaded. Close the SurveyCTO Desktop app once the download is completed.



Step-13: PII Prompt. Since we set the global *warning* to 1, the dataflow will send you a warning message by identifying some variables that it may consider to be PII. If they are indeed PIIs, you will have to go back to the setup and include them in the global *pii.* Otherwise, click "No"



Step-14: The rest of the code should run without any intervention. Once the entire do file runs, Go to ~`\Brick Kiln\02_DataFlow\01_Baseline\Manager\04_Output\02`_Checks and go through the Check Report.



Correction Sheet & Log

Correction sheet is used to make any changes that you may want to make in the data for any observation. This may become necessary for various reasons, suppose the surveyor has given a wrong entry and you were later informed or there was an error in SCTO coding or there were duplicate entries. In such cases we use a correction sheet and the log of this changes are stored in the correction log.

You can find the correction sheet here:

```
02_Dataflow/03_Data/03_Corrections/Correction_sheet.xlsx
```

The logs will be saved everyday here:

```
02_Dataflow/04_Output/01_Logs/Correction_log_ `date'.xlsx
```

There are two sheets in this excel file; the first sheet corrections is used if we want to change any particular observation, As for the second sheet droplist is used to drop particular observations by using the key. An example is given below,

	key	variable	current_value	correction	remarks
195	uuid:07d2e2b5-912f-49fc-8fd0-72f724f74eb7	life_std	.	-66	Shopper didn't ask this particular question that is why we are labelling it
196	uuid:07d2e2b5-912f-49fc-8fd0-72f724f74eb7	mileage_std_up	.	-66	Shopper didn't ask this particular question that is why we are labelling it
197	uuid:07d2e2b5-912f-49fc-8fd0-72f724f74eb7	mileage_std_low	.	-66	Shopper didn't ask this particular question that is why we are labelling it
198	uuid:07d2e2b5-912f-49fc-8fd0-72f724f74eb7	std_probing	.	-66	Shopper didn't ask this particular question that is why we are labelling it
202	uuid:91e58405-4780-4271-a230-a5517cb3ae49	std_availibility	0	1	According to the protocol recommended battery should be considered as standard battery if any of the listed batterries was not found in the shop.
203	uuid:9759f9b3-b60d-43ac-be19-5d904e24ff5c	std_availibility	0	1	According to the protocol recommended battery should be considered as standard battery if any of the listed batterries was not found in the shop.
204	uuid:8a3ed27f-1402-4997-9e07-a272a336f22d	pp	.	1	Auditor reconciled it to complete since we have audio proof of it otherwise it was submitted as incomplete survey by the enumerator
207	uuid:c40de356-6967-46ae-bd40-c2ddebe79e80	shop_id	MS_093	MS_077	Same shop had two different ID so FMO filled it with MS_077 so we are changing it here and dropping MS_093 from the assignment

corrections droplist +

key	reason
uuid:f081f928-2655-433f-b6c7-79774869cff7	Dropping it since Shahin Alam played the wrong character
uuid:056d03f6-a24e-46ed-959b-a863265877b6	Dropping it since Shahin Alam played the wrong character
uuid:6f12bf6a-fe57-479a-b366-a29a8a8fe18b	Technical issue, dropping this as duplicated - Sifat verified from field
uuid:5a80ebf2-81f9-4a87-be27-a76d7a13b7a7	Zieul couldn't do it on the first day but got the same shop opened
uuid:0752db34-40e3-4d58-8b91-d34e99d74035	Jesan did this shop. The shop was closed at first then it was opened that's why dropping the incomplete one.
uuid:356b1b2e-6b39-4aff-b2b5-d5ec4d3cecea	This was done by Hannan bhai dropping this since because of ID duplicates (MS_130 & MS_133)

corrections | droplist

This sheet works just like the simple Stata `replace` command. That is why note that, string variables can only be corrected with string values. Similarly numeric or categorical variables can be corrected with numeric values. For missing values keep the values blank and for numeric or categorical use " . " (dot).

Never ever miss writing the remarks or reasons in the correction sheet otherwise the dataflow will not run. Secondly, don't be short sighted, never write it in short and flesh it out to the level max. Since you are neither *Chacha Chowdhury* nor a computer. So you will forget why you made these changes. This document should work as an answer sheet to all the questions regarding the changes made in data. This is mandatory for data integrity.

Variable Naming & Commonly Used Variable Names

We want to make sure the variable names and variable labels are internally and universally consistent so that it makes everything easier in the long run. Ideally, you should balance your PI's preferences and the best practices. In this document you will have some guidance on the best practices. For variable naming there are no hard and fast rules, but read this blog to understand the best practices. These best practices are sketched keeping in mind that we use electronic data collection tools and statistical software like Stata. These best practices keep in mind either the limitations of Stata or SurveyCTO/ODK or easiness of use (such as use of wildcard). Therefore, it is wise to follow these rules to avoid any issue in the future that can almost break the data flow and make everyone's life easier.

PIs might have different preferences. In that case, discuss it with your supervisor. here are some guidelines:

1. All variables should be labeled, and all multiple-choice variables should have value labels.

2. The labeling system should be internally consistent for example Calculated fields should begin with the prefix CALC.

Here is a list of commonly used variables.

Variable name	Discription
audio_audit	Audio audit field
text_audit	Text audit field
upazilaname	Upazila name
upazilaid	Upazila ID/code
zilaname	Zila name
zilaid	Zila ID/code
unionname	Union name
unionid	Union ID/code
gps	GPS coordinate
startdate	Start date (Automatically created by dataflow)
starttime	Start time

enddate	End date (Automatically created by dataflow)
endtime	End time
duration	Duration of the survey
consent	Consent variable
enumid	Enumerator ID (Cannot contain any space and must be alpha-numeric)
enumname	Enumeartor name
enumcomments	Enumerator comments open text field
sctocomments	SurveyCTO comment field type

Special codes

Option	Code
Others	-96
Refused	-98
Don't know	-99
Skipped	-97
Not applicable	-95

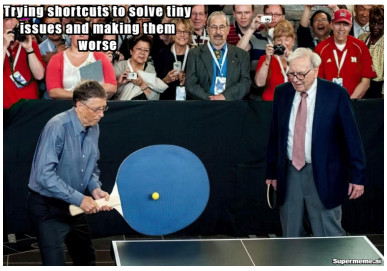
No need to get creative—just stick to the script when naming these variables!

Common Mistakes

The first common mistake that everybody makes is considering Dataflow as a mobile phone app and it will run automatically. No, it isn't, it's a series of Stata codes so the mechanism is like tools; everything that is necessary to run this tool has to be in place. Second mistake that you are making is thinking everybody got it the first time, No, nobody could. While you are reading this guide, maybe the ones who have written it are stuck at some point running the Dataflow right now.

Let's mention a few common mistakes which can be avoided easily and will save you from those error messages in red.

1. Copying folders from other projects and not running the `arcedfold`



2. Running the dataflow without submitting any forms.

3. Not purging the test data from SurveyCTO server before starting the survey. Note that the dta file can be deleted anytime it will not create any issue but it may take longer to download.

4. Not using the adjusting global `surveystart` and `targetsample` according to the survey

```
gl  surveystart          "12may2024"
gl  targetsample         "1250"
```

5. Naming the files such as xlsx form, Form ID according to the survey. For example,

```
gl formid "mystery_shopping_2nd"

gl xlsform
"${cwd}/../01_Instruments/02_XLSForm/Mystery_Shopping_Shopper.xlsx"
```

6. Changing variable names or groups after setting up dataflow.

7. Too long variable names. Duplicate variable names.


8. Not having a Unique ID for the survey (Other than the automatically generated variable 'key')

9. Unexpected character in question labels, for example double quotes or single quotes.

10. Not closing the dialogue box or SurveyCTO desktop after finishing download to resume the dataflow. It automatically restarts when the dialogue box or app is closed. If minimized, the do file will not continue

11. Keeping Check_report.xlsx open while running the dataflow.
12. Switch 'encrypt' turned on but no PIIs specified.

```
**# Preferences
*---------------------------
gl  warning                 0
gl  encrypt                 1
```

```
gl  PIIs                    #d ;
                            "
                            " ;
                            #d cr
```

13. Numeric values in `enumid` or space in alphanumeric values e.g. "3258" or "AFFT 3258". The correct way is "AFFT-3268" .

14. While defining the container(commonly named as “rawdata”) size, not speculating the amount of data to be stored from the survey properly. If the container is full the Dataflow will not run, since it will have no space to accommodate any more raw data.

    Similarly, too large container for a small amount of data will do the same as it will have trouble syncing.

15. As we use Google drive & dropbox for shareability, these are also commonly known issues of syncing. The way to get rid of it is given below,

    `Open veracrypt>Settings>Preference and uncheck "Preserve modification timestamp of file container"`