

Data Preparation for Tax Gap toolkit



UNU
WIDER

Dr Maria Jouste
Research Associate, UNU-WIDER

2024 ATI Tax Gap Estimation Toolkit Training | 7 October 2024

Required data for bottom-up approach

Tax return data

- Corporate income tax, Value-added tax, Personal income tax (PAYE, individual businesses)
- Full population at least for one year
- Variables: reported income, deductions, exemptions, and tax liability
- Detailed variable requirements in Readme file

Required data for bottom-up approach

Tax return data

- Corporate income tax, Value-added tax, Personal income tax (PAYE, individual businesses)
- Full population at least for one year
- Variables: reported income, deductions, exemptions, and tax liability
- Detailed variable requirements in Readme file

Audit data

- Detailed audit data
- All audits at least for one year
- Variables: detected errors, adjustments, and penalties
- Detailed variable requirement in Readme file

Required data for bottom-up approach

Tax return data

- Corporate income tax, Value-added tax, Personal income tax (PAYE, individual businesses)
- Full population at least for one year
- Variables: reported income, deductions, exemptions, and tax liability
- Detailed variable requirements in Readme file

Audit data

- Detailed audit data
- All audits at least for one year
- Variables: detected errors, adjustments, and penalties
- Detailed variable requirement in Readme file

Supporting data

- Taxpayer register, payments
- Variables: firm size, location, payments, industry
- Detailed variables requirements in Readme file

Readme file

README and Guidance for Bottom-Up Tax Gap Analysis Toolkit

- 1 Overview
- 2 Usage Guidelines
- 3 First Stage
- 4 Second Stage: Data Management
Process for VAT, CIT, and PIT
- 5 Third Stage: Data Management
Process for Audit Return Data
- 6 Fourth Stage: Merging CIT, PIT, and
VAT with the Audit Data

README and Guidance for Bottom-Up Tax Gap Analysis Toolkit

Part 1: Data Management

1 Overview

This toolkit, developed by UNU-WIDER/ATI, provides a comprehensive framework for conducting bottom-up tax gap analyses. It utilizes machine learning algorithms to estimate Value Added Tax (VAT), Personal Income Tax (PIT), and Corporate Income Tax (CIT) gaps, specifically designed for Revenue Authorities.

The toolkit consists of two code files: the first part is data management, which is used to clean the tax return and audit data and merge them, so it is ready for machine learning analysis using the second code file.

Version:

Date: 15/07/24

Prerequisites:

To use this toolkit, we must have [Stata software](#) installed on your computer. Additionally, the data requirements depend on the tax gap you want to analyze, such as CIT, PIT, and/or VAT return data, in addition to audit data.

Variable list: example

Table 1:

Variable Name	Type/Category	Definition
tpin	VAT, CIT, PIT, Audit / O, Ness	A Taxpayer Identification Number (TPIN) is a unique identifier assigned to each taxpayer by tax authorities for identification and administrative purposes, used to track and manage tax obligations. TPIN ensures accurate processing of tax returns, payments, and related activities. Different countries may refer to TPINs using various terms.
rtn_year	VAT, CIT, PIT, Audit / O, Ness	Return year, the specific year in which the tax return was filed by the taxpayer.
month	VAT, PIT / O, Ness	The month during which the VAT/PIT return was filed, important for tracking monthly returns.
isic_code	VAT, CIT, PIT / O, Ness	International Standard Industrial Classification (ISIC) Code: A numeric code, used to classify the economic activities of the taxpayer. This code may also consist of 2 or 4 digits if available.
broad_sector	VAT, CIT, PIT / O, Unn	The broad sector classification of the taxpayer's economic activities is indicated by a single-digit code, providing a general overview of the sector.
regdate	VAT, CIT, PIT / O, Unn	Registration date, the date when the taxpayer registered for VAT/CIT/PIT.
town_name	VAT, CIT, PIT, Audit / O, Ness	Name of the town, city, or locality where the taxpayer is located. This is used for geographical analysis and audit planning, taking into account the administrative structure of the specific country or region
province	VAT, CIT, PIT, Audit / O, Ness	Name of the administrative division where the taxpayer is located. This could refer to a province, state, region, district, or any other relevant administrative unit, depending on the country.
returnid	VAT, CIT, PIT / O, Unn	Return ID, a unique identifier for each tax return.
returnstatus	VAT, CIT, PIT / O, Unn	Status of the tax return, indicating whether the return was filed, processed, or pending.
returntype	VAT, CIT, PIT / O, Unn	Type of tax return filed, indicating whether it is a regular return, amendment, etc.
invat	VAT / O, Ness	Input VAT, the amount of VAT the taxpayer paid on their purchases, expressed in local currency.
outvat	VAT / O, Ness	Output VAT, the amount of VAT the taxpayer collected on their sales, expressed in local currency.
vat_due	VAT / O, Ness	VAT due, the net amount of VAT the taxpayer needs to pay (output VAT minus input VAT), expressed in local currency.

Data preparation

First Stage: Requirements

- **Software:** STATA (installed on your computer).
- **Directory Structure:** Create a main "*Cleaning*" folder with the following subfolders:
 - **Excel:** Store raw data files.
 - **Data:** Save cleaned and processed data.
 - **Graphs:** Save generated graphs.
 - **Dofile:** Store STATA dofiles.
 - **Results:** Save analytical results.
 - **Log:** Store log files.

First Stage: Steps

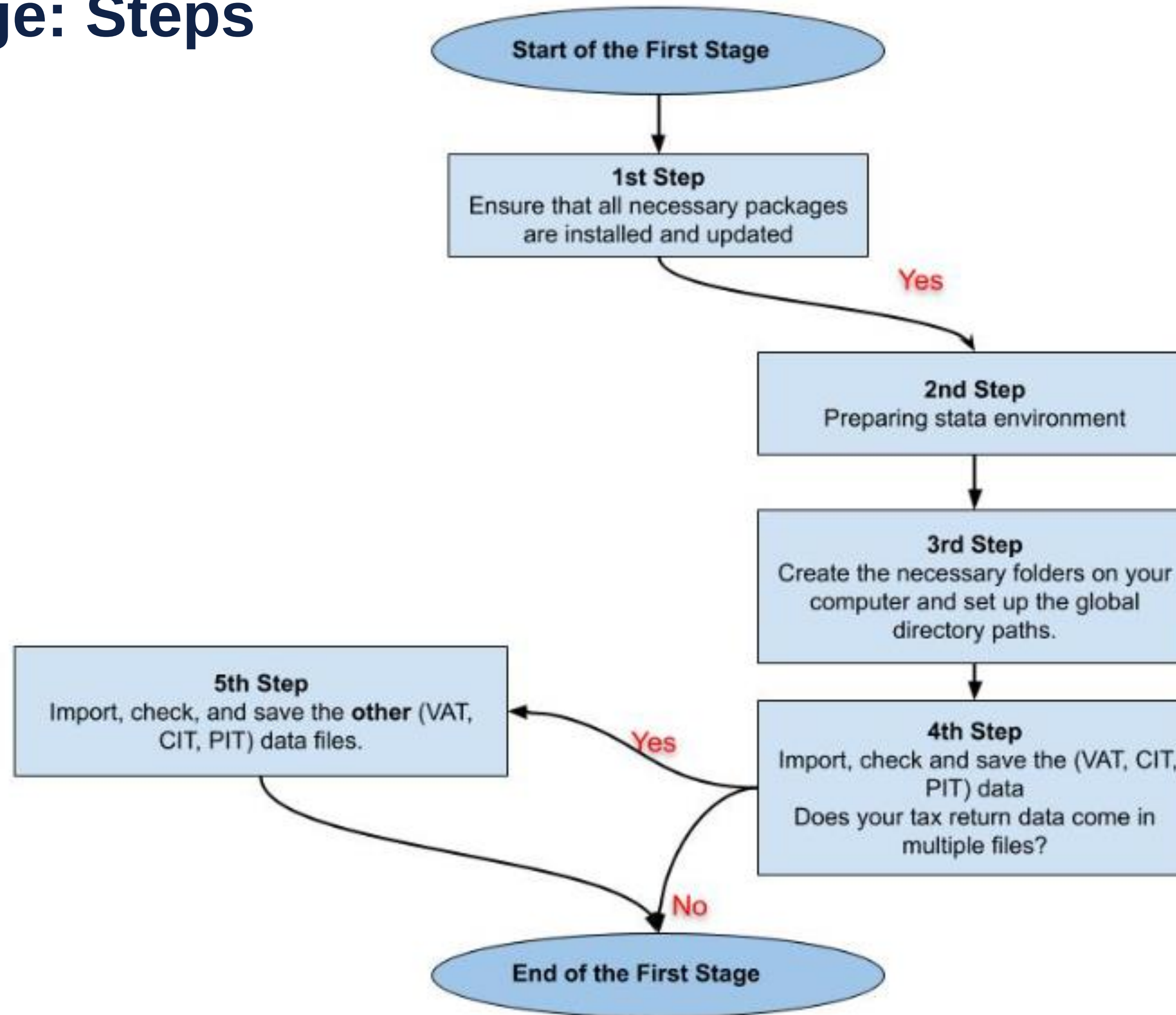


Figure 1: Workflow for the First Stage

First Stage: Codes

1. Install and Update Necessary STATA Packages:

```
capture ssc install usepackage  
usepackage unique mdesc savesome xls2dta rmfiles, near
```

2. Prepare Stata Environment

```
version 14      /* Use Stata Version 14 */
```

```
set more off  
clear matrix  
set linesize 200  
set matsize 800  
cap log close
```

3. Set Up Global Directory Paths:

```
global dir "<YOUR PATH>"  
  
global data      "$dir\Data"  
global graphs    "$dir\Graphs"  
global xlsout     "$dir\Excel"  
global dopath     "$dir\Dofile"  
global results    "$dir\Results"  
global log        "$dir\Log"  
  
log using "$log\Tax Gaps data prep.log", replace
```

4. Importing Data, Formatting, Renaming Variables, and Saving the Data

```
import delimited "$xlsout\<Your_CIT,PIT,or VAT_returns_Data_1st_File>.dsv",clear  
  
format tpin %15.0g  
  
rename old_var_name new_var_name //Example  
  
rename actvty_isic_code isic_code  
rename provincename province  
rename rtn_id returnid  
rename form_name returtype  
  
save "$data\<Your_CIT,PIT,or VAT_returns_Data_1st_File>.dta", replace
```

Second Stage: Steps

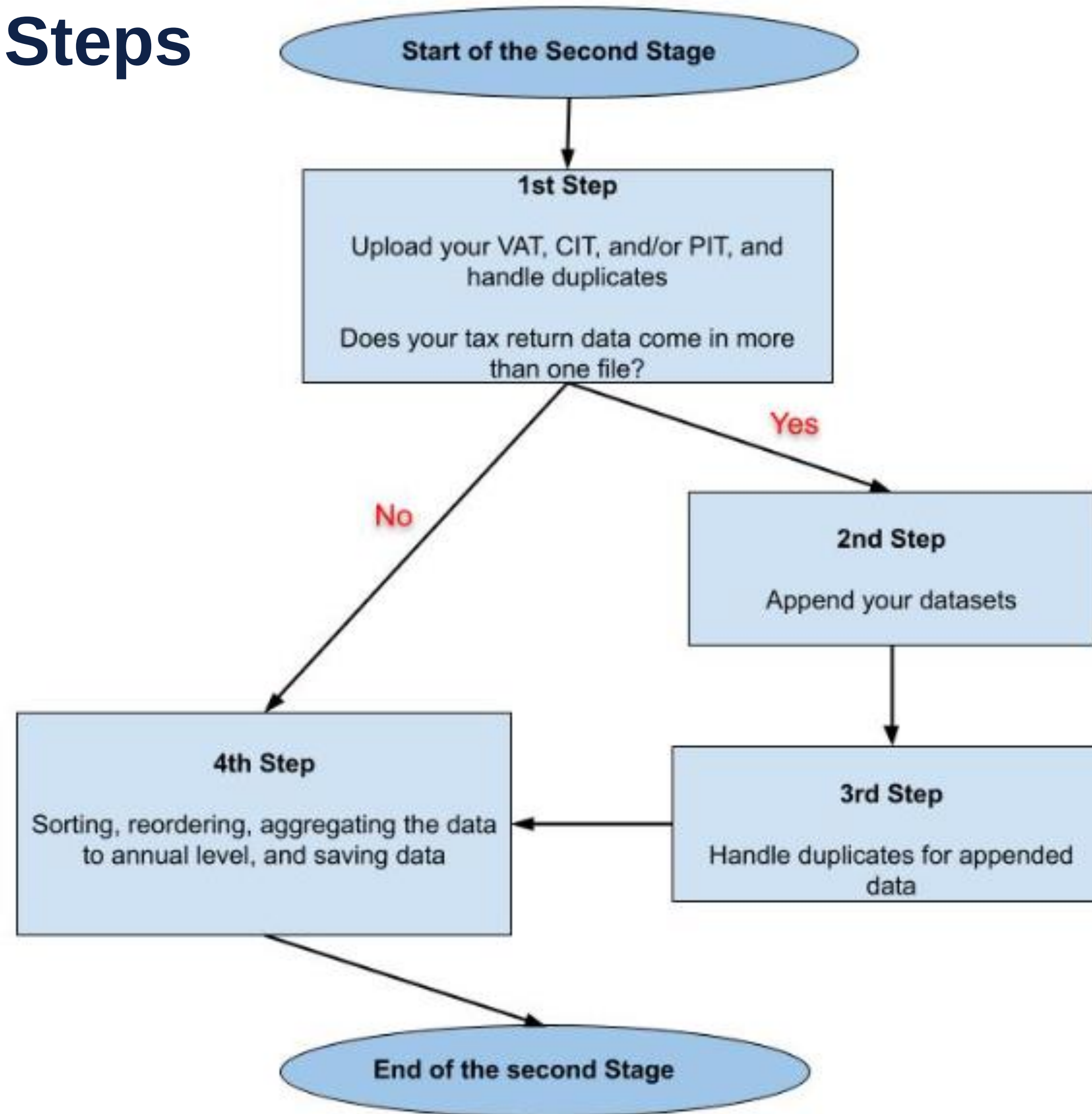


Figure 2: Workflow for the Second Stage

Second Stage: 1st Step - Dealing with Duplicates

What is a Duplicate in Tax Return Data?

- In this context, a **duplicate** refers to **multiple entries** that share the same combination of key identifiers. These identifiers typically include variables such as the return year, taxpayer identification number (TPIN), and sometimes additional identifiers like the type of tax or filing period.
- Tax authorities sometimes make multiple reviews or assessments, resulting in multiple records for the same period (e.g., month, year) and taxpayer. In this scenario, a duplicate refers to these **multiple records** that pertain to the same tax return.
- Our objective is to identify and keep **the most recent tax return** (self-reported by the taxpayer) and discard older versions or reviewed/assessed returns.

Second Stage: 1st Step - Dealing with Duplicates

- This step begins by uploading the data and then implementing the code required to handle duplicates, as described previously.

```
use "$data\<Your_VAT,CIT,or PIT_returns_Data_1st_File>.dta", clear

duplicates tag rtn_year tpin month, gen(dup0)
tabulate dup0
browse rtn_year tpin month broad_sector outvat dup0 if dup0 == 1

bys tpin month: egen latest0 = max(rtn_year) if dup0 != 0
drop if rtn_year != latest0 & dup0 != 0

duplicates drop rtn_year tpin month, force
duplicates tag rtn_year tpin month, gen(dup0)
assert dup0 == 0
drop dup0 latest0
```

Second Stage: 2nd & 3rd Steps – Case of Multiple Data Files

- The [2nd step](#) aims to append multiple data files into a single dataset.

```
append using "$data\<Your_CIT, PIT, or VAT_returns_Data_2nd_File>.dta"  
browse
```

- The [3rd step](#) repeats the same process of the [1st step](#) regarding dealing with duplicates but now for the combined dataset.

Second Stage: 4th Step - Sorting, Reordering, Collapsing, and Saving Data

- This step involves sorting and reordering the dataset by key variables, such as return year and taxpayer ID.
- For VAT and PIT, variables are aggregated to an annual level to reduce the dataset to one observation per taxpayer per year. The cleaned data is then saved for further analysis.

```
///Sort Variables:
sort rtn_year tpin

///Reorder Variables:
order rtn_year tpin isic_code broad_sector regdate town_name province returnid returnstatus
returntype

///Collapse VAT and PIT to Annual Level:

//For VAT:
collapse (sum) invat outvat vat_due vat_vars, by(tpin rtn_year)
save "$data\vat_ann.dta", replace

//For PIT:
collapse (sum) pit_due pit_vars, by(tpin rtn_year)
save "$data\pit_ann.dta", replace

//For CIT:
save "$data\cit.dta", replace
```

Third Stage: Steps

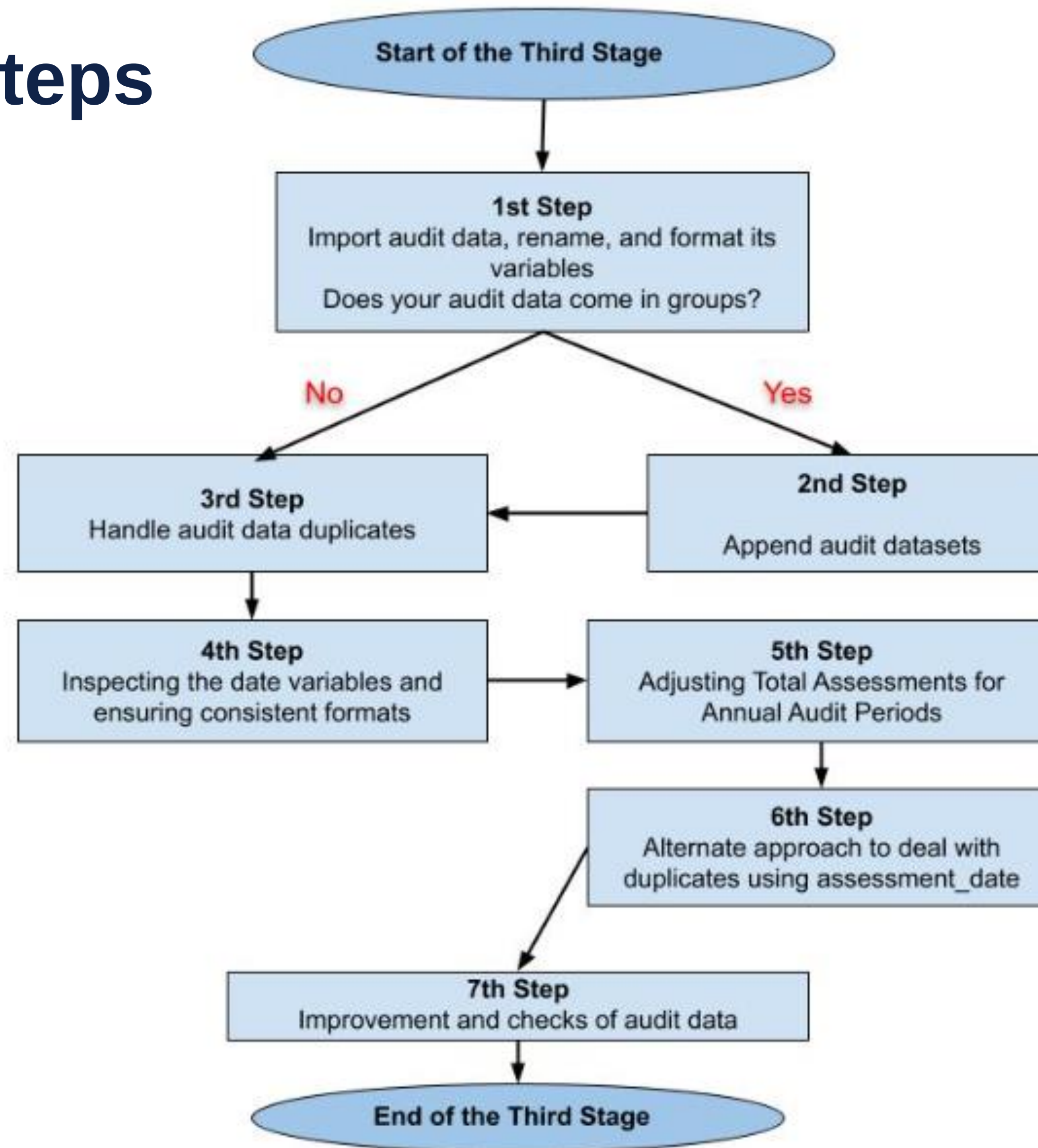


Figure 3: Workflow for the Third Stage

Third Stage: 1st & 2nd Steps – Import and Append Audit Data

- In the [1st step](#), we import the audit data, rename and format its variables, and then save the data.

```
import excel "$xlsout\<Your_Audits_Data_1st_File>.xlsx", sheet("Sheet1") firstrow clear  
rename old_var_name new_var_name  
format tpin %15.0g  
save "$data\<Your_Audits_Data_1st_File>.dta", replace
```

- If the data is in more than one file, we implement the [2nd step](#) of appending the data after applying the same process from the 1st step to the second file.

```
append using "$data\<Your_Audits_Data_2nd_File>.dta"  
save "$data\<combined_audit_data>.dta", replace
```

Third Stage: 3rd Step – Dealing with Duplicate in Audit Data

What is a Duplicate in Audit Data?

- In the context of audit data, a duplicate refers to multiple entries that have the same combination of key identifiers. These identifiers can vary depending on the dataset but typically include variables such as the return year, taxpayer identification number, and sometimes additional identifiers like month or specific audit-related fields.
- A duplicate occurs when more than one record has the same combination of these identifiers, indicating that the same audit entry has been recorded multiple times.
- Our objective here is to identify and keep the most recent post-audit return (reported by the audit department) and discard older versions or reviewed tax assessments.

Third Stage: 3rd Step – Dealing with Duplicate in Audit Data

- The following code handles duplicates as described previously.

- Note:

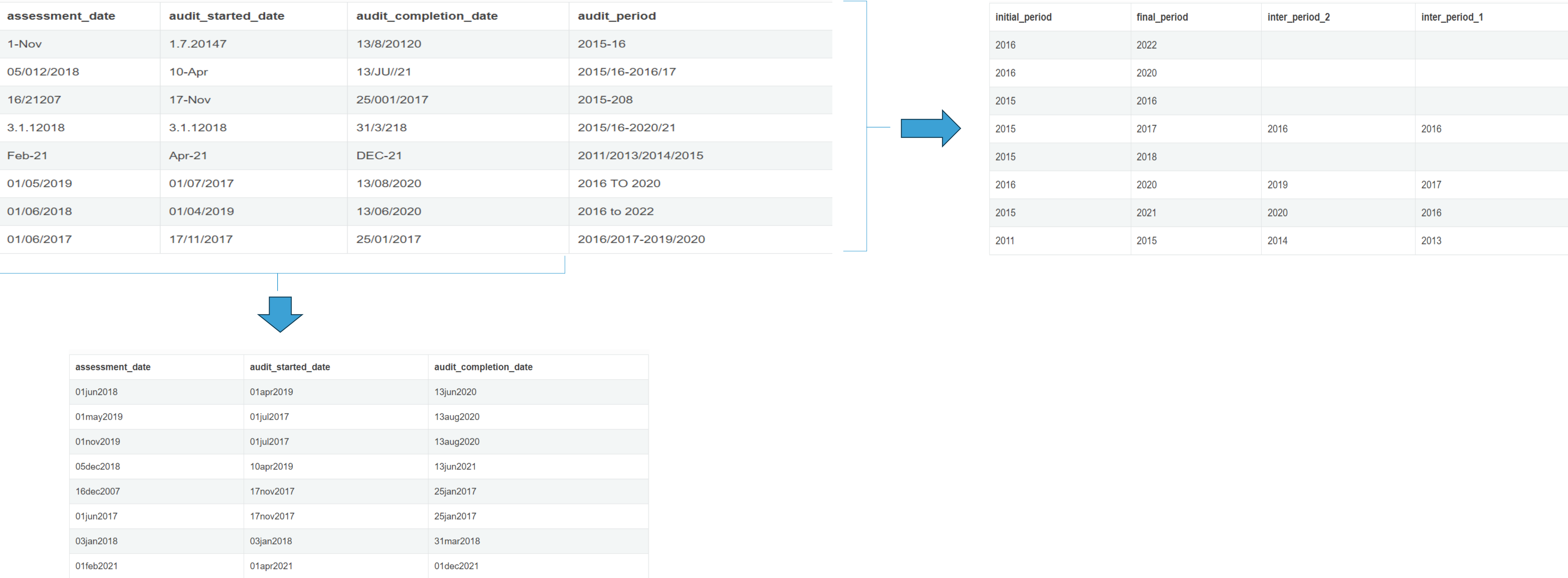
In 6th step, we use an alternative approach to deal with duplicates using assessment_date.

```
use "$data\<Your_Audits_Data_2nd_File or combined_audit_data>.dta", clear
unique rtn_year tpin

duplicates tag rtn_year tpin , gen(dup0)
duplicates drop rtn_year tpin , force
drop dup0
```

Third Stage: 4th Step – Inspecting the date variables and ensuring consistent formats

In the README file we provide instructions and example cases of how to transform date variables to desirable formats to facilitate the analysis as audit data might contain multiple dates such as assessment_date, audit_started_date, audit_completion_date, and audit_period.



Third Stage: 5th Step – Adjusting Total Assessments for Annual Audit Periods

- In this step, negative values in total assessments are set to zero, and the total assessments are allocated across the audit period to reflect annual assessments.

```
replace totalassessments = 0 if totalassessments < 0  
  
egen total_audit_period = rownonmiss(initial_period final_period inter_period_1 inter_period_2)  
  
egen totalassessments_annual = totalassessments/total_audit_period
```

Third Stage: 6th Step – Alternate Approach to Deal with Duplicates

- This approach uses the assessment_date to retain only the latest entry for each return year and tax category.

```
// Sort and Drop Duplicates Based on Assessment Date
gsort +tpin +rtn_year +tax_head +initial_period +final_period +inter_period_1 +inter_period_2 -assessment_date
duplicates drop tpin rtn_year tax_head initial_period final_period inter_period_1 inter_period_2, force

// 6.2. Sort and Organize Variables
sort rtn_year tpin initial_period final_period inter_period_1 inter_period_2
order rtn_year tpin initial_period final_period inter_period_1 inter_period_2 isic_code broad_sector town_name province
format tpin %15.0g

// 6.3. Check for Duplicates
unique rtn_year tpin initial_period final_period inter_period_1 inter_period_2
duplicates tag rtn_year tpin initial_period final_period inter_period_1 inter_period_2, gen(dup1)
tabulate dup1
browse rtn_year tpin totalassessments dup1 if dup1 == 1

// Dropping Clear Duplicates
duplicates drop rtn_year tpin initial_period final_period inter_period_1 inter_period_2, force
unique rtn_year tpin initial_period final_period inter_period_1 inter_period_2
drop dup1
```

Third Stage: 7th Step – Improvement and Checks

- This step simplifies and categorizes audit types by converting text descriptions into numerical codes, consolidating them into broader categories, and labeling them for easier analysis.

```
encode AuditType, gen(auditcode)
recode auditcode (2/6 13=1) (8/12 14=2) (7=3), gen(audittype)
drop auditcode
lab define type_lbl 1"1.Comprehensive audit" 2"2.Issue audit" 3"3.Credibility check" 4"4.CIT Audit" 5"5.PIT Audit" 6"6.VAT audit" 7"7.Desk audit"
label val audittype type_lbl
tabulate audittype
order tpin initial_period final_period inter_period_1 inter_period_2 audittype started_date
audit_completion TotalTax

*save as STATA data
save "$data\audit.dta", replace
```

Fourth Stage: Steps

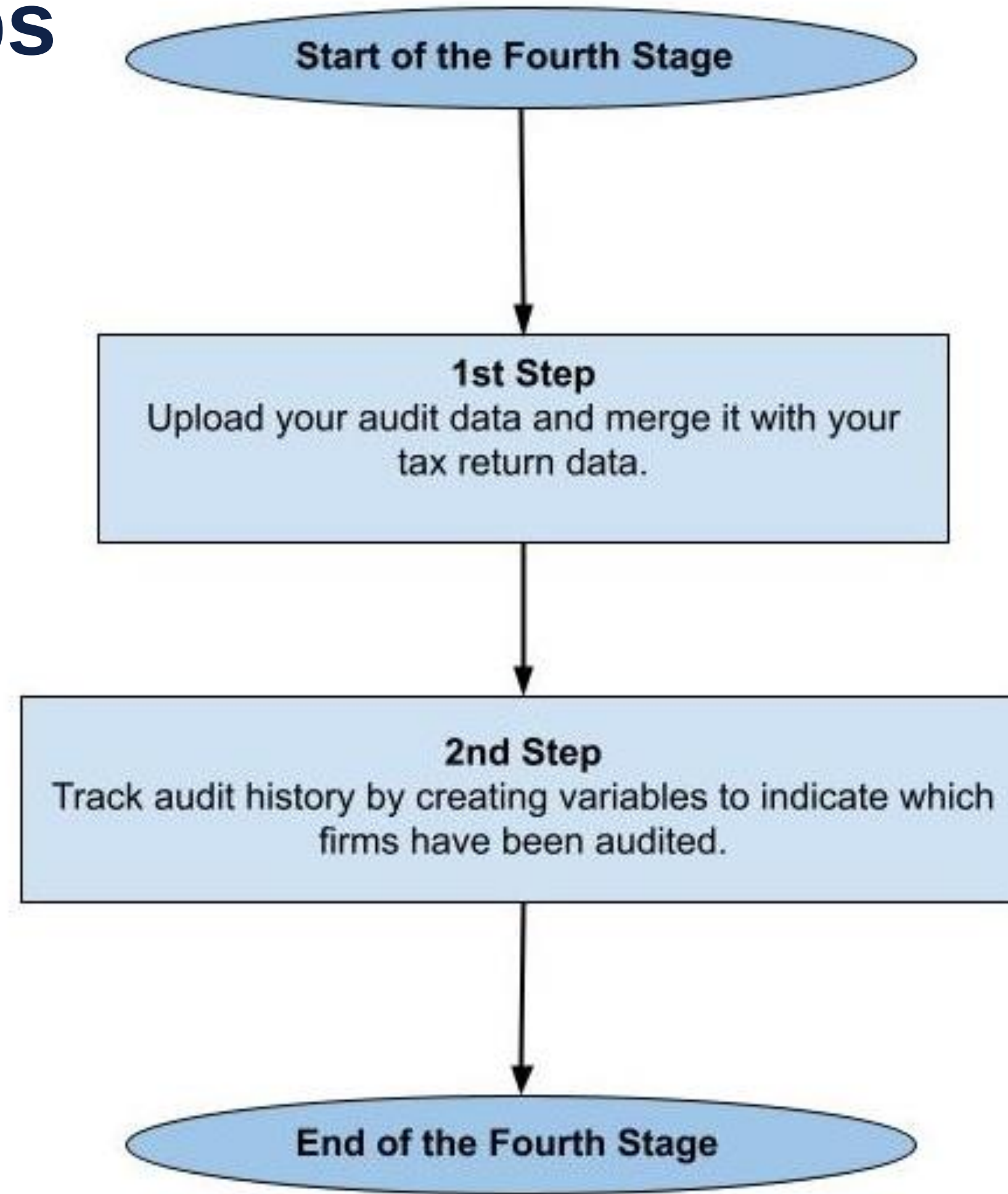


Figure 5: Workflow for the Fourth Stage

Fourth Stage: 1st Step – Upload and Merge Audit and Tax Return Data

- This step uploads and merges audit data with annual VAT data to identify firms with matching VAT returns.
- Firms are then categorized based on their audit status, and a new variable is created to flag audited firms.
- Note: The following code is for VAT data, but the same process can be applied to PIT and CIT.

```
use "$data\audit.dta", clear

merge 1:1 tpin rtn_year using "$data\vat.dta"
recode _merge (3=1) (2 =0) (1 =2), gen(vataudstat)
drop _merge
label define vatstat_lbl 1"1.Match & Audited" 0"0.No Match not Audited" 2"2.No Match & Audited"
label val vataudstat vatstat_lbl
tabulate vataudstat
order tpin rtn_year vataudstat

generate vataudited=1 if vataudstat==1|vataudstat==2
replace vataudited=0 if vataudstat==0|vataudstat==.
```

Fourth Stage: 2nd Step – Tracking Audit History

- This step tracks audit history by calculating the total number of audits and creating variables to indicate whether a firm has ever been audited.
- Descriptive labels are then assigned to the variables for clarity, and the merged dataset is saved.

```
egen nr_audits=total(audit), by(tpin)
generate ever_vataudited = (nr_audits>0)

tabulate          vataudited
label var rtn_year "Return Year"
label var vataudited "VAT Audit Status"
label values vataudited          vataudited
label define vataudited 0  "Unaudited", replace
label define vataudited 1  "VAT Audited", add
label define ever_vataudited 0  "never VAT Audited", replace
label define ever_vataudited 1  "ever VAT Audited", add

*save final data
save "$data\vataudit.dta", replace

log close
```

Q&A

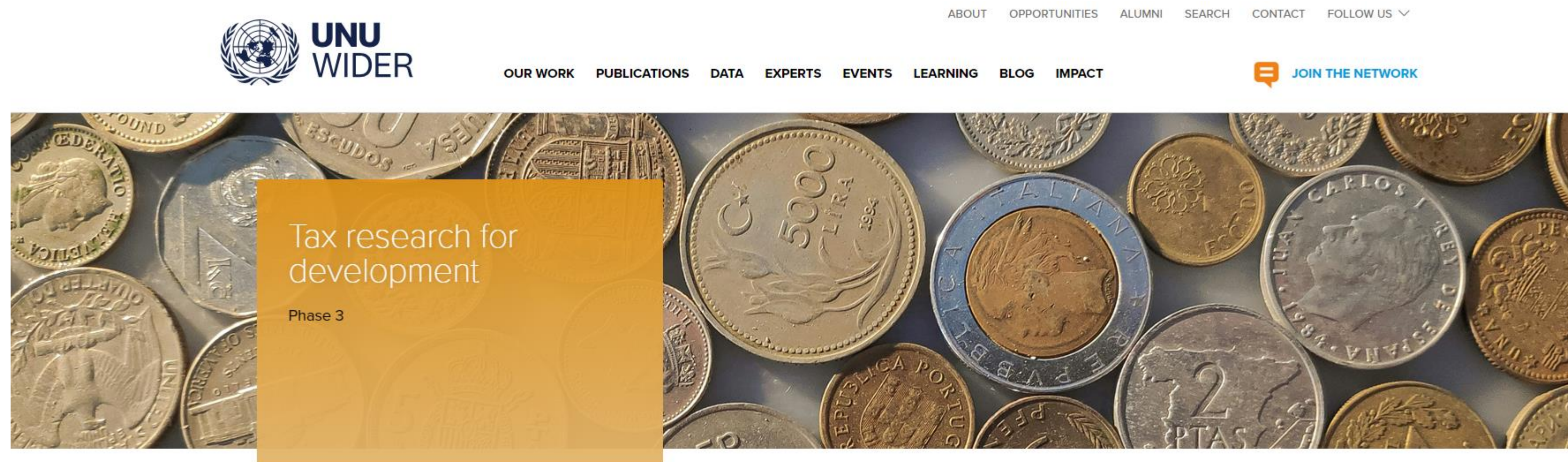
Summary

Key Learnings So Far:

- What is a tax gap
- Why estimate the tax gap
- Benefits and limitations of the Bottom-up approach
- Importance of Random Audits
- What are the data requirements for the tax gap estimation
- How to prepare data for tax gap estimation

Bottom-up approach materials

- This workshop trains to use the Bottom-up approach. Providing technical sessions with examples of estimating the **Tax gap** using **dummy data**
- Tax Gap Toolkit **Technical Note**, **Readme files** and example **Stata codes** will be available on the UNU-WIDER's webpage later this year



Thank you!

Maria Jouste

jouste@wider.unu.edu



**UNU
WIDER**

More information available online

wider.unu.edu

Follow us on LinkedIn, Facebook, and Twitter **@UNUWIDER**