World Bank-ICAR Funded
National Advanced Higher Education Project
Centre for Advanced Agricultural Science and Technology (CAAST)

On
"Genomics-Assisted Crop Improvement and Management"

Training Manual

Data Analysis with Stata

January 25 – 29, 2021

Division of Agricultural Economics
ICAR – Indian Agricultural Research Institute
New Delhi – 110012
www.nahep-caast.iari.res.in

NAHEP Sponsored
Online Workshop Programme
On
Data Analysis with Stata
Course Director

Alka Singh

Professor and Head

Division of Agricultural Economics

ICAR-Indian Agricultural Research Institute

Pusa Campus, Delhi – 110012

E-mail: asingh.eco@gmail.com

Phone No. 9871198527

Course Coordinators

Aditya K S

Scientist

Division of Agricultural Economics

ICAR-Indian Agricultural Research Institute

New Delhi 110 012

E-mail: adityaag68@gmail.com

Nithyashree M L

Scientist

Division of Agricultural Economics

ICAR-Indian Agricultural Research Institute

New Delhi 110 012

E-mail: nithya.econ@gmail.com

Division of Agriculture Economics
ICAR-Indian Agricultural Research Institute
New Delhi- 110 012

About NAHEP-CAAST at IARI, New Delhi

Centre for Advanced Agricultural Science and Technology (CAAST) is a new initiative and student-centric subcomponent of the World Bank-sponsored **National Agricultural Higher Education Project (NAHEP)** granted to the Indian Council of Agricultural Research, New Delhi, to provide a platform for strengthening education and research activities of postgraduate and doctoral students. The ICAR-Indian Agricultural Research Institute, New Delhi, was selected by the NAHEP-CAAST programme. NAHEP sanctioned Rs 19.99 crores for the project on "**Genomic assisted crop improvement and management**" under CAAST programme. The project at IARI specifically aims at inculcating genomics education and skills among the students and enhancing the expertise of the faculty of IARI in the area of genomics.

Objectives:

1. **To develop online teaching facility and online courses for enhancing the teaching and learning efficiency, and scientific communication skills**
2. **To develop and/or strengthen state-of-the-art next-generation genomics and phenomics facilities for producing quality PG and Ph.D.students**
3. **To develop collaborative research programmes with institutes of international repute and industries in the area of genomics and phenomics**
4. **To enhance the skills of faculty and PG students of IARI and NARES**
5. **To generate and analyze big data in genomics and phenomics of crops, microbes and pests for genomics augmentation of crop improvement and management**

IARI's CAAST project is unique as it aims to provide funding and training support to the M.Sc. and Ph.D. students from different disciplines who are working in the area of genomics. It will organize lectures and training programmes, send IARI students for training at expert laboratories and research institutions abroad, and cover students from several disciplines. It will provide opportunities to the students and faculty to gain international exposure. Further, the project envisages developing a modern lab named **Discovery Centre** that will serve as a common facility for students' research at IARI.

Core-Team Members

S.No.	Name of the Faculty	Discipline	Institute
1.	Dr. Ashok K. Singh	Genetics	ICAR-IARI
2.	Dr. Vinod	Genetics	ICAR-IARI
3.	Dr. Gopala Krishnan S	Genetics	ICAR-IARI
4.	Dr. A. Kumar	Plant Pathology	ICAR-IARI
5.	Dr. T.K. Behera	Vegetable Science	ICAR-IARI
6.	Dr. R.N. Sahoo	Agricultural Physics	ICAR-IARI

7.	Dr. Alka Singh	Agricultural Economics	ICAR-IARI
8.	Dr. A.R. Rao	Bioinformatics	ICAR-IASRI
9.	Dr. R.C. Bhattacharya	Molecular Biology & Biotechnology	ICAR-NIPB
10.	Dr. K. Annapurna	Microbiology **Nodal officer, Grievance Redressal, CAAST**	ICAR-IARI
11.	Dr. R. Roy Burman	Agricultural Extension **Nodal officer, Equity Action Plan, CAAST**	ICAR-IARI
12.	Dr. K.M. Manjaiah	Soil Science & Agri. Chemistry **Nodal officer, CAAST**	ICAR-IARI
13.	Dr.Viswanathan Chinnusamy	Plant Physiology **PI, CAAST**	ICAR-IARI

Associate Team

S.No.	Name of the Faculty	Discipline	Institute
14.	Dr. Kumar Durgesh	Genetics	ICAR-IARI
15.	Dr. Ranjith K. Ellur	Genetics	ICAR-IARI
16.	Dr. N. Saini	Genetics	ICAR-IARI
17.	Dr. D. Vijay	Seed Science & Technology	ICAR-IARI
18.	Dr. Kishor Gaikwad	Molecular Biology & Biotechnology	ICAR-NIPB
19.	Dr. Mahesh Rao	Genetics	ICAR-NIPB
20.	Dr. Veena Gupta	Economic Botany	ICAR-NBPGR
21.	Dr. Era V. Malhotra	Molecular Biology & Biotechnology	ICAR-NBPGR
22.	Dr. Sudhir Kumar	Plant Physiology	ICAR-IARI
23.	Dr. Dhandapani R	Plant Physiology	ICAR-IARI
24.	Dr. Lekshmy S	Plant Physiology	ICAR-IARI
25.	Dr. Madan Pal	Plant Physiology	ICAR-IARI
26.	Dr. Shelly Praveen	Biochemistry	ICAR-IARI
27.	Dr. Suresh Kumar	Biochemistry	ICAR-IARI
28.	Dr. Ranjeet R. Kumar	Biochemistry	ICAR-IARI
29.	Dr. S.K. Singh	Fruits & Horticultural Technology	ICAR-IARI
30.	Dr. Manish Srivastava	Fruits & Horticultural Technology	ICAR-IARI
31.	Dr. Amit Kumar Goswami	Fruits & Horticulture Technology	ICAR-IARI
32.	Dr. Srawan Singh	Vegetable Science	ICAR-IARI
33.	Dr. Gograj S Jat	Vegetable Science	ICAR-IARI
34.	D. Praveen Kumar Singh	Vegetable Science	ICAR-IARI
35.	Dr. V.K. Baranwal	Plant Pathology	ICAR-IARI
36.	Dr. Deeba Kamil	Plant Pathology	ICAR-IARI
37.	Dr. Vaibhav K. Singh	Plant Pathology	ICAR-IARI
38.	Dr. Uma Rao	Nematology	ICAR-IARI
39.	Dr. S. Subramanium	Entomology	ICAR-IARI
40.	Dr. M.K. Dhillon	Entomology	ICAR-IARI
41.	Dr. B. Ramakrishnan	Microbiology	ICAR-IARI
42.	Dr. V. Govindasamy	Microbiology	ICAR-IARI
43.	Dr. S.P. Datta	Soil Science & Agricultural Chemistry	ICAR-IARI
44.	Dr. R.N. Padaria	Agricultural Extension	ICAR-IARI
45.	Dr. Satyapriya	Agricultural Extension	ICAR-IARI
46.	Dr. Sudeep Marwaha	Computer Application	ICAR-IASRI
47.	Dr. Seema Jaggi	Agricultural Statistics	ICAR-IASRI
48.	Dr. Anindita Datta	Agricultural Statistics	ICAR-IASRI
49.	Dr. Soumen Pal	Computer Application	ICAR-IASRI
50.	Dr. Sanjeev Kumar	Bioinformatics	ICAR-IASRI
51.	Dr. S.K. Jha	Food Science & Post Harvest Technology	ICAR-IARI
52.	Dr. Shiv Dhar Mishra	Agronomy	ICAR-IARI
53.	Dr. D.K. Singh	Agricultural Engineering	ICAR-IARI
54.	Dr. S. Naresh Kumar	Environmental Sciences; **Nodal officer, Environmental Management Framework**	ICAR-IARI

Foreword

The Division of Agricultural Economics, a constituent of the School of Social Sciences of ICAR-Indian Agricultural Research Institute, was established in 1960. The division's mandate is to conduct research in frontier areas and serve as a center for academic excellence in postgraduate education. Since its inception, the division has contributed basic and applied research with significant implications for agricultural policy. The division has achieved excellence in postgraduate education and research as an ICAR-UNDP Centre of Excellence through a faculty exchange program for human resources development and infrastructure facilities. Since 1995 it has been functioning as an ICAR Centre of Advanced Faculty Training (CAFT) to strengthen agricultural economics and policy research in the national agricultural research system.

The division's research contributions have been globally recognized, and many of the alumni occupy positions of repute in national and international organizations. The division has maintained good academic liaison with other divisions at IARI and other national and international agricultural research institutions. The research focus of the division has been continuously reoriented to address contemporary development challenges. Current research thrust areas of the division include investment in agriculture, inclusive growth, and poverty alleviation, the impact of agricultural technologies and policies, price forecasting and market outlooks, natural resource use in agriculture and ecosystem services, climate change effects, mitigation and adaptations, and food and nutritional security.

Statistical analysis of data is at the heart of research in social science. With advances in data collection tools and several large scales, nationally representative datasets are now available. However, handling large datasets requires knowledge of data analysis software like Stata and R. Considering the need to build the capacity of students in using Stata software and in management of large datasets, an online workshop on "Data Analysis with Stata" is being organized by the division. The workshop was sponsored by the Centre for Advanced Agricultural Science and Technology (CAAST) component of the World Bank-funded National Agricultural Higher Education Project (NAHEP). The workshop covers basic Stata commands for data management, statistical analysis and data visualization. The sessions also cover the extraction of structured text data, reshaping and merging large datasets like different rounds of survey data from NSSO, Annual Survey of Industries and Periodic Labor Force Survey. I am sure that the workshop participants will immensely benefit from exposure to statistical software and nuances of handling large datasets.

Rashmi Aggarwal

Dean and Joint Director(Edn)
ICAR-IARI,New Delhi

Date: 27.01.2021

Preface

Data management and Statistical analysis of data are integral parts of research. With the availability of different survey datasets, expertise in data management software is essential, particularly in social science research. Stata is one of the very popular data analysis software used by social scientists across the globe. Stata offers an easy way of managing the data, performing advanced econometric techniques. Use specialized Stata file types like 'dofiles' and 'logfiles' to help reproducible research. Students of social sciences need to equip themselves with data handling and statistical analysis skills, and the knowledge of Stata software will be very useful. In this direction, the division of Agricultural Economics is organizing a five-day workshop on 'Data Analysis with Stata.' Apart from introducing data management with stata, we will also impart training on handling large survey datasets like the dataset of National Sample Survey Office, Annual Survey of Industries and Periodic Labour Force Surveys. This teaching manual consists of basic Stata commands used for the analysis covered in the session. The sessions will also be live-streamed on YouTube.

We take this opportunity to sincerely acknowledge all the authors' contributions in the preparation of this manual. Considering the diversity and comprehensive nature of the topics covered, the manual can act as a quick and effective reference source for the students in their future research endeavours. The Stata commands for basic analysis are given in this manual, which will help students as a ready reference.

Alka Singh

Aditya K S

Nithyashree M L

Date: 27.01.2021

Acknowledgments

1. Secretary DARE and Director General ICAR, New Delhi
2. Deputy Director General (Education), ICAR, New Delhi
3. Assistant Director General (HRD), ICAR, New Delhi
4. National Coordinator, NAHEP, ICAR, New Delhi
5. CAAST Team, ICAR-IARI, New Delhi
6. PG School, ICAR-IARI, New Delhi
7. Director, ICAR-IARI, New Delhi
8. Dean & Joint Director (Education), ICAR-IARI, New Delhi
9. Joint Director (Research), ICAR-IARI, New Delhi
10. Head, Division of Agriculture Economics, ICAR-IARI, New Delhi
11. Professor, Division of Agriculture Economics, ICAR-IARI, New Delhi
12. AKMU, ICAR-IARI, New Delhi
13. Staff & Students, Division of Agriculture Economics, ICAR-IARI, New Delhi

Contents

Lecture notes on Data Analysis with Stata

Aditya K S

Scientist, Division of Agricultural Economics, ICAR-IARI, New Delhi

This chapter is intended to introduce Stata software to the beginners and used it as teaching material in the workshop on 'Data Analysis with Stata.' Those who don't have access to Stata software can request a short-term student license by filling out this form https://www.stata.com/customer-service/short-term-license/.

Why learning Stata?

For social science research, mostly observational, statistical analysis is at the center of scientific and reproducible research. Researchers often have to use varied data sources to answer the research question and employ suitable up-to-date statistical methods. Knowledge of Stata can be handy in data cleaning, management and handling, particularly if the datasets are large. Another advantage of Stata is that it helps in 'reproducible research' by making it easy to share the results with other researchers or referees of journals through 'do files and log files.' Further, since Stata is one of the most commonly used software by social scientists worldwide, its knowledge will be extremely helpful in collaborative works.

Learning Stata with Stata

The best way to learn stata is by accessing the help menu. The easy way to access help is to use the 'help' command in stata. In the command window of stata, type help followed by the name of the analysis for which you need help. If the system is connected to the internet, the help menu will open in the new window, where the command structure, description and options are explained. One can download the dataset used in the help menu (by using sysuse or webuse

commands, which will be given in the help file as well as PDF documentation) and try the various examples detailed in the help.

Further, stata allows us to type the command to perform any required analysis directly. Alternatively, there is also an interactive UI, a drop-down menu. For example, by typing summarize variable list, we can get the result to summarize a data. One can also do it by using the drop-down menu by going to current statistics. Nevertheless, it is better to use the commands than menu-based UI, as different commands can be saved in a one file called "do file" that enables single click execution. Few basic stata commands and functions are discussed below in detail.

The help command: Help command is probably the most useful command that every Stata user should know. It makes it easier to get help on various commands. For example, one wants to know about summarize command. In the command window, type

help summarize

All the details regarding the summarize command are listed, along with the examples. However, there is every possibility that I may not know the command name to use the help command. In such cases, I can use the search command. For example, one wants to run factor analysis and has no clue about the command to use for this analysis.

help factoranalysis

There is every possibility that stata returns with error message. Instead,

search factor analysis

Findit command: stata has many user-written commands, which are made available as packages. These are verified by stata, and published in stata journal. These packages can

simplify the complex analysis and enable us to use the methodology with a single click. To find and install such user-written packages, one can use findit command.

Findit psmatch2

The result sheet will include the stata journal articles and packages. Find the suitable one from the package list and install it.

Simple Mathematical Operators

Mathematical operators are very similar to those used in MS excel

display 2+3

display (2/3)^2

Rational And Logical Operators

Rational operators include- >, <, >=,<=

Logical operators are '|' ***for 'or', '&' for 'and'***

Use of do files and log files

Dofiles in stata

Dofiles and log files are extremely useful for researchers. Dofile is a stata file, which is used to store the commands. Usually, analysis involves many steps, with each step requiring the use of a certain set of commands. Storing them in the form of dofiles will help to reduce the time. Dofiles will also help us to revisit the steps followed and treatment of variables. Dofiles are also used to share the commands with others.

- Open the dofile from the stata drop down menu 
- Open new dofile
- Copy all the used commands and save dofile
- Commands in the dofile are directly executable

Log files in stata

These days, many international journals demand log files when we submit an article for publication. Log files essentially record all our activities in a particular session. It includes the record of commands used and results obtained. It is a good practice to open a log file before starting an analysis.

Log using filename, format

Example: Log using demonstration.smcl

Browsing the data files: We can use the browse command to see the data set we are using. To refine our results browse can be combined with 'if' or 'or' or 'and' command.

Example : br if sex==1 & age==18 | age==17

String and numeric variables: Commonly, variables are saved in stata in either string or numeric format. Usually, the qualitative variables are saved in string format such as name, village name etc. Many commands are not compatible with string variables. In such cases, the destring command can be used. One should note that the values of string variables are enclosed in "".

Example: destring Household_id, generate (hhid_new)

Handling the data set

Creating a new variable: Usually, gen or egen command is used to create the new variable. These commands can be combined with arithmetic operators or logical operators

Example : gen illiterate=1 if edu==1

egen gross_return_rank=rank(gross_return)

egen stdev_age= std(age)

The replace command: Replace command generally helps in editing the value of already existing or generated variable; this command can be used.

Example: replace sex_dummy=0 if missing(sex_dummy)

Sort command- To sort the data in ascending order. To order the data in descending order, use gsort command

Tabulate command is used to make tables from the data.

Tabstat command: To tabulate the description of the variables.

Keep or drop command can be used to delete the unwanted variable/ dropping a few observations.

Tostring and destring : Command that can convert string to numeric variables

Xi command: Automatically creates dummy variables for the specified categorical variable.

Few Commands for Basic Statistical analysis

1. Regression: reg or probit or logit
2. Marginal Effects after probit or logit: mfx
3. Correlation: corr or pwcorr
4. Student T test: ttest
5. Chisquare test: tab, all
6. Principal Component Analysis: pca
7. Factor analysis: factor

Few useful user-written commands

1. Esttab; for making regression tables.
2. tatable2: Calculate group-wise mean value and test the significance
3. orth_out: Perform t-test for any number of variables at once
4. pscore: Estimates Propensity Scores

5. psmatch2: Perform Propensity Score Matching
6. cem: Perform Coarsened Exact Matching
7. doubleb: Perform Double Bound Contingent Valuation.
8. clustersampsi: perform power calculations for RCT

Exercise -1

1. Open Stata and open a log file and a do file.
2. Use ‘help reg’ command
    a) Click on the ‘view complete PDF Manual Entry.’
    b) Go to page 2252 in the manual – Ordinary Least Squares- Example 1. Follow the commands in Example 1.
3. Use ‘help corr,’ Scroll down to examples.
    a) Follow the commands.
    b) Click on the ‘View complete PDF Manual Entry.’
    c) Follow the examples on Page 505 in the manual.
4. Save the log and do files.

Submit the dofile and logfile using this google form

https://forms.gle/z5XBWA19JzDvsHRW7

Hands-on Session 1- Basic stata commands for data management and analysis

Use the 'Stata example for class' file for executing these commands. The data file is extracted data from visit 1, block 4 of NSSO's situational assessment survey of farmers (70^{th} round). Kindly e-mail 'adityaag68@gmail.com' if you need the datasets for this hands-on session. Commands are given in italic font.

- **Creating the log files- Always start the session by opening the log file**

log using demonstration.smcl, replace

- **Summary statistics with 'summarize' or 'tabstat' or 'tabulate' or inspect commands**

summarize age landowned

tabstat age, stat(mean median sk ku)

inspect age edu

- **Using 'by' prefix.**

bysort sgroup: sum operatedland

bysort sgroup pension: sum operatedland

sort state

by state: summarize edu training

sort sex

by sex: tab edu training

(here 'sum' is the abbreviated version of summarizing, and the tab is the abbreviation for tabulating)

- **Converting the string data to numeric**

destring Household_id, generate (hhid_new)

- **Data Management with 'gen', 'egen', 'replace', 'keep' and 'drop'.**

gen dummy_sex=1 if sex==1

replace dummy_sex=0 if sex==2

keep if age<=40

drop if age>40

egen mean_age=mean(age), by(sex)

- **'ttest' and chi square test in stata**

ttest landowned, by(sex)

tab age edu, al

- **Installing user-written commands**

findit orth_out (and then browse, select the orth_out package and then install)

findit ttable2

*ssc install orth_out (*ssc install is the alternative command for installing user written packages)

ssc install ttable2

- **'orth_out' and 'ttable2' user written package**

orth_out age landowned landleasedin , by(sex) pcompare

ttable2 age landowned landleasedin, by(sex)

- **Labeling the variables and values for easy interpretation**

label variable non_agric "Recieve non agricultural income"

label define yes_no 1 "Yes" 2 "No"

label values non_agric yes_no

label values wage yes_no

- **Regression using stata- OLS and Probit/ logit**

reg dummy_sex edu training religion sgroup

probit dummy_sex edu training religion sgroup

- **Plotting the distributions**

kdensity landowned, normal

histogram landowned if sex==1, color(green) width(1) normal

histogram religion, discrete xlab (1 (1) 10)

histogram religion,color (red) discrete xlab (1 (1) 10)

histogram religion,color (black) discrete xlab (1 (1) 10)

histogram religion,fcolor (black) lcolor(yellow) discrete xlab (1 (1) 10)

histogram religion,fcolor (brown) lcolor(blue) discrete xlab (1 (1) 10)

histogram religion,fcolor (none) lcolor(blue) discrete xlab (1 (1) 10)

(Note: Use the graph recorder function to record the editing)

- **Two way graphs for better visuals**

twoway (histogram landowned if sex==1 & landowned<15, color(green) width(0.1)) (histogram landowned if sex==2 & landowned<15, fcolor(none) lcolor(yellow) width(0.1)), legend(order(1 "Male" 2 "Female"))

- **Automatically creating dummy variables from categorical variable**

xi i.religion

probit dummy_sex edu training sgroup i.religion

or

xi:probit dummy_sex edu training sgroup i.religion

- **At the end of the session, don't forget to close the log file.**

log close

Hands-on Session 2- Basic stata commands for data management and analysis

Answer the following questions using Stata

1. Import the dataset "prod_data_12", which has been shared with you already.
2. Summarize the data so that you see the means, standard deviation, min, and max of each variable
3. Find the correlation between Area and Yield, then create a simple plot of this relationship
4. Create a kernel density (kdensity) plot of yield per acre
5. Generate a new variable that indicates (i.e., 1 or 0) yields above 3,000 kg per acre.
6. Create a simple histogram of fertilizer expenditure
7. Create a more complex histogram of fertilizer expenditure by sex (i.e., overlapping histograms, one for males the other for females).
8. Calculate the Net returns from the data. Summarize the net profit
9. Create the following new variables:
    a. Total expenditure on fertilizer and pesticide
    b. The log of yield
    c. Numeric variable for sex called "male."
    d. Variable called "farmer" that indicates that the respondent's primary occupation is farming
    e. Variable called "ownland" to indicate that the respondent owns the plots they farmed
10. Create dummy variables for a.) each soil type, and b.) each land type
11. Create a single variable that contains the average fertilizer expenditure in each village
12. Generate a dummy variable indicating if the farmer received more than the average price received by the sample farmers

13. What are the correlates of farmers receiving a higher price?

The commands for all the questions are given below

- Open the logfile

log using exercise_1

- Summarise

describe

sum

summarize age area prodloc price fertilizer pesticide hiredlabor

tab sex

tab primoccup

tab irrigsource

- Correlation

corr age educyear

plot age educyear

twoway (scatter age educ)

- Plotting the yield

histogram prodloc

kdensity prodloc, normal

- bar diagram to depict expenditure

graph hbar fertilizer pesticide hiredlabor others , by(sex)

- Histogram can depict the distribution**

graph histogram fertilizer, by(sex)

- Histogram for better visualization

twoway (histogram fertilizer if male==1, color(green) width(0.1)) (histogram fertilizer if male==0, fcolor(none) lcolor(yellow) width(0.1)), legend(order(1 "Male" 2 "Female"))

- Violin chart

ssc install vioplot

vioplot fertilizer, over(sex)

- Creating new variable for higher yield

gen yield_above5=1 if prodloc>3000

replace yield_above5=0 if missing(yield_above5)

tab yield_above5

sum prodloc

- Estimating cost of cultivation, Gross return and Net return

gen cost1= seedplant+ fertilizer+ pesticide+ hiredlabor+ others

egen cost2= rowtotal(seedplant fertilizer pesticide hiredlabor others)

br cost1 cost2

gen nr= gr-cost1

- Identifying extreme values**

graph box age, marker(1, mlabel(farmerid))

extremes age

- Kdensity plots

kdensity prodloc, normal

graph save Graph "C:\Users\adity\Dropbox\Redgram work\kdensity.gph"

kdensity fertilizer if sex=="Male"

graph save Graph "C:\Users\adity\Dropbox\Redgram work\Kdensity- male.gph"

kdensity fertilizer if sex=="Female", normal

graph save Graph "C:\Users\adity\Dropbox\Redgram work\Kdensity-female.gph"

- Gen expenditure command and taking log

gen expenditure= fertilizer+ pesticide

gen log_yield= log(prodloc)

- Generating dummy variables

gen male=1 if sex=="Male"

replace male=0 if missing(male)

tab primoccup

gen Farmer=1 if primoccup=="Farmer" | primoccup=="farming"

replace Farmer=0 if missing(Farmer)

tab tenure

gen ownland=1 if tenure=="Owned"

replace ownland=0 if missing(ownland)

tab soiltype

gen clay=1 if soiltype=="Clay"

replace clay=0 if missing(clay)

gen loam=1 if soiltype=="Loam"

replace loam=0 if missing(loam)

gen sandy=1 if soiltype=="Sandy"

replace sandy=0 if missing(sandy)

tab landtype

gen lowland=1 if landtype=="Lowland"

replace lowland=0 if missing(lowland)

- The ‘egen’ command

egen fertiliser_ave=mean(fertilizer), by (village)

tab fertiliser_ave

- Alternative way of achieving same results

sort village

by village : egen fert_vil= mean(fertilizer)

gen diff= fert_vil-fertiliser_ave

sum diff

- Ttest to see if male farmers produce a higher yield

ttest prodloc, by(male)

- Examining the characters of farmers who gets a higher price

gen above_avergae= 1 if price>13.27

replace above_avergae=0 if missing(above_avergae)

reg above_avergae expenditure hiredlabor Farmer ownland clay loam sandy lowland age sex educyear primoccup secoccup area

- To automatically generate dummies for villages**

xi i.village

- Using village fix effects in regression

*reg log_yield expenditure hiredlabor Farmer ownland clay loam sandy lowland _Ivillage_**

- Alternate way to use dummies

xi: reg log_yield expenditure hiredlabor Farmer ownland clay loam sandy lowland i.village

Exercise-II: Attend the following questions and submit the dofles and logfiles

1. Use the command “sysuse auto” in stata command window to download the dataset
2. Which command will you use to examine the data? List the variables and description
3. Summarise the data and make descriptive tables from the data.
4. Generate a new variable better mileage for all the cars having a mileage of more than 20.
5. Generate the kdensity of variable ‘displacement.’
6. For the variables mpg, weight, draw box diagram and violin plots and interpret the results.
7. Does the car fetch a higher price if it is manufactured abroad?
8. Draw a scatter graph of ‘mpg’ and ‘weight,’ and use different colors or symbols to depict the place of manufacturing (foreign).
9. Draw the same graph, but only for the data points for which mpg> 20
10. Properly label the graphs, set the background color and record the changes.
11. Draw a scatter graph of mpg and weight and add a trend line to it.
12. Run the following three regressions
    a) Price = F (mpg length turn)
    b) Price = F (weight length displacement)
    c) Price = F (weight length gear_ratio foreign)
    d) Present the result in a single table using esttab/ estout
13. Generate a dummy variable if the price is less than 5000. Run the following models using the dummy variable. (
    a) Dummy_ Price = F (weight length gear_ratio foreign) – Linear Probability Model
    b) Dummy_ Price = F (weight length gear_ratio foreign) – Probit model
    c) Dummy_ Price = F (weight length gear_ratio foreign) – Logit model

d) Present the result in a single table using esttab

Extraction and handling of unit-level data sets of NSSO and ASI

Nithyashree M L

Scientist, Division of Agricultural Economics, ICAR-IARI, New Delhi

This chapter is aimed to introduce the different unit-level data sets available and the way of handling them. As a beginner, before getting the hands-on unit-level data, it is important to go through the key indicators or summary results. This helps to understand the purpose of the survey and the data coverage in the particular dataset. Besides, this will also help the user comprehend the sample size, sampling design, and estimation procedure necessary for further analysis and interpretation of the results. Since the key indicators/summary results, by and large, provides the aggregate estimate, the use of unit-level data enables the researchers to work with the basic unit of the survey, i.e., the household in case of agriculture and firm about the industry sector. The unit-level data sets are generally available in text format. To convert these files into a usable format, some of the steps need to be understood, and they are discussed by using two unit-level datasets *viz.,* Key Indicators of Situation of Agricultural Households in India of National Sample Survey Organization (NSSO) and Annual Survey of Industries (ASI). For handling any unit-level data, it is essential to use suitable statistical software. Here we use STATA (version 15) software for the illustration, and the summary of the steps to be followed is shown in Figure 1.

The first and most important step in using unit-level data is to get a though understanding of the supporting documents; they mainly comprise the layout, which instructs how the data arranges in the text file and other details such as tabulation programme, code list, concept and definition and schedule, etc. For extracting data from the text file, one has to

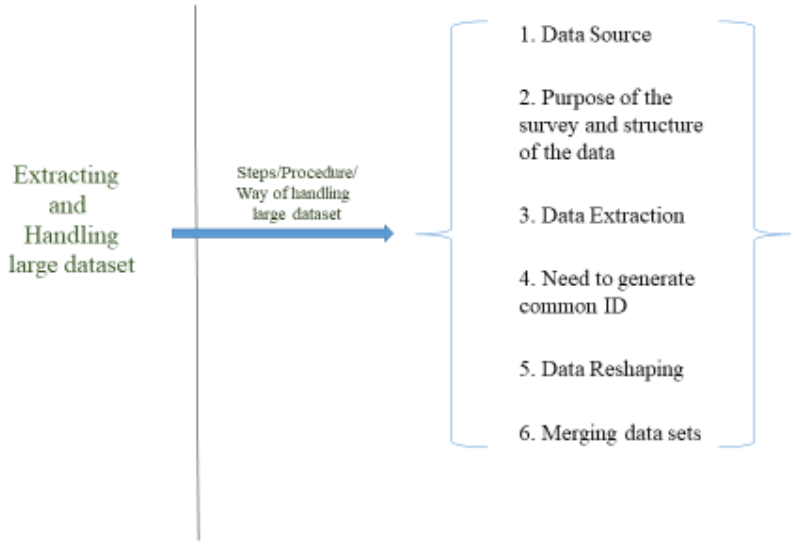


Figure 1. Steps to extract and handle the unit-level data

write the commands in *STATA* by using the information given in the layout. For example, for the NSSO data set it can be written as:

```
infix Centercode 1-3 FSU 4-8 Round 9-10 Schedule 11-13 Sample 14 Sector 15 NSS_Region 16-18 District 19-20 Stratum
21-22 Sub_Stratum 23-24 Sub_Round 25 Sub_sample 26 FOD_subRegion 27-30 Hamlet 31 SecondStageStratum 32 ///
SampleHHno 33-34 VisitNo 35 Level 36-37 Filler 38-40 HouseholdSize 43-44 Religion 45 Social_Group 46 Dwelling 47
StructureType 48 WaterSource 49 HH_OwnLand_YN 50 LandType 51 PossesLandOutsideVillage 52 LandOperated 53
Land_Own_ha 54-62 Land_Leasedin_ha 63-71 Land_neitherLeased 72-80 Land_LeasedOut_ha 81-89
Land_total_possessed 90-98 Cultivation_whetherPerformed 99 Cultivation_incomesource 100
Livestock_whetherPerformed 101 Livestock_incomesource 102 OtherAgr_whetherPerformed 103
OtherAgr_incomesource 104 NonAgr_whetherPerformed 105 NonAgr_incomesource 106 Wage_whetherPerformed 107
Wage_incomesource 108 Pension_whetherPerformed 109 Pension_incomesource 110 Remittance_whetherPerformed 111
Remittance_incomesource 112 Others_whetherPerformed 113 Others_incomesource 114 MGNREGACard_YN 115
RationCard_YN 116 RationCard_Type 117 NSS 127-129 NSC 130-132 MLT 133-142 using "C:\Users\Nithyashree M
L\Desktop\Stata_Workshop\Unit level data\AH0233V1.TXT"
```

where the command infix is written to extract the text file into *STATA* file and at the end data path has been specified. Similarly, for the ASI data it can be written as:

```
infix year 1-4 Factory 5-9 State 10-11 str C 12 str Block 13 Scheme_code 14 NIC4digit 15-18 NIC5digit 19-23 R_U 24
RO_SRO 25-29 noofunits 30-32 Statusofunit 33-34 Manufacturingdays 35-37 Non_Manufacturingdays 38-40 Total 41-43
Costofproduction 44-57 directlyexported 58-60 Multiplierfactor 61-73 using "C:\Users\Nithyashree M
L\Desktop\Stata_Workshop\Unit level data\ASI\OASBLA16.TXT"
```

After extracting the data files, it is important to create the common ID or unique ID in each data file. This will help to identify the household across the different data sets and also enables

to combine the information available in different data files. For creating a common id, the following command in *STATA* is written as:

NSSO

egen id=concat (FSU Hamlet SecondStageStratum SampleHHno)

ASI

egen id=concat(Factory State C)

To combine the information of different data files, the researcher must ensure that the observations are uniquely identified across the data sets; if not, the data needs to be rearranged. By observing the data structure, it can be grouped as wide-format or long format. For example, in Figure 2, students' ids and results are written in the wide-format in Table 1, and the same students' subject-wise grades are arranged in the long format in Table 2. To combine the information of grade and results, Table 2 has to be reshaped/rearranged into a wide format, which can be done using the command reshape in STATA, i.e., reshaping wide Grade, i (ID) j(J).

Table 1. student id and results (Wide format)

ID	Results
R20D151	P
K20D162	P
S20D170	F
R20D165	F

Table 2. student id and grade (long format)

ID	Subject (J)	Grade
R20D151	1	A
R20D151	2	A
R20D151	3	A
K20D162	1	A
K20D162	2	A
K20D162	3	A
S20D170	1	C
S20D170	2	C
S20D170	3	C
R20D165	1	C
R20D165	2	C
R20D165	3	C

Table 2 converted as Wide format

ID	Grade1	Grade2	Grade3
R20D151	A	A	A
K20D162	A	A	A
S20D170	C	C	C
R20D165	C	C	C

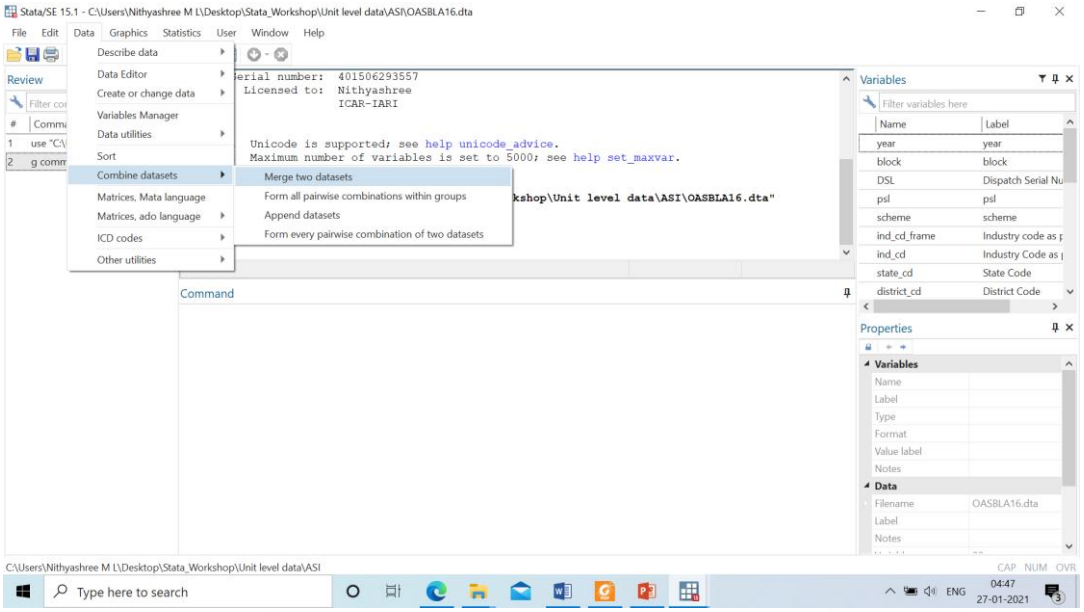
Figure 2. Visualization of wide and long format data

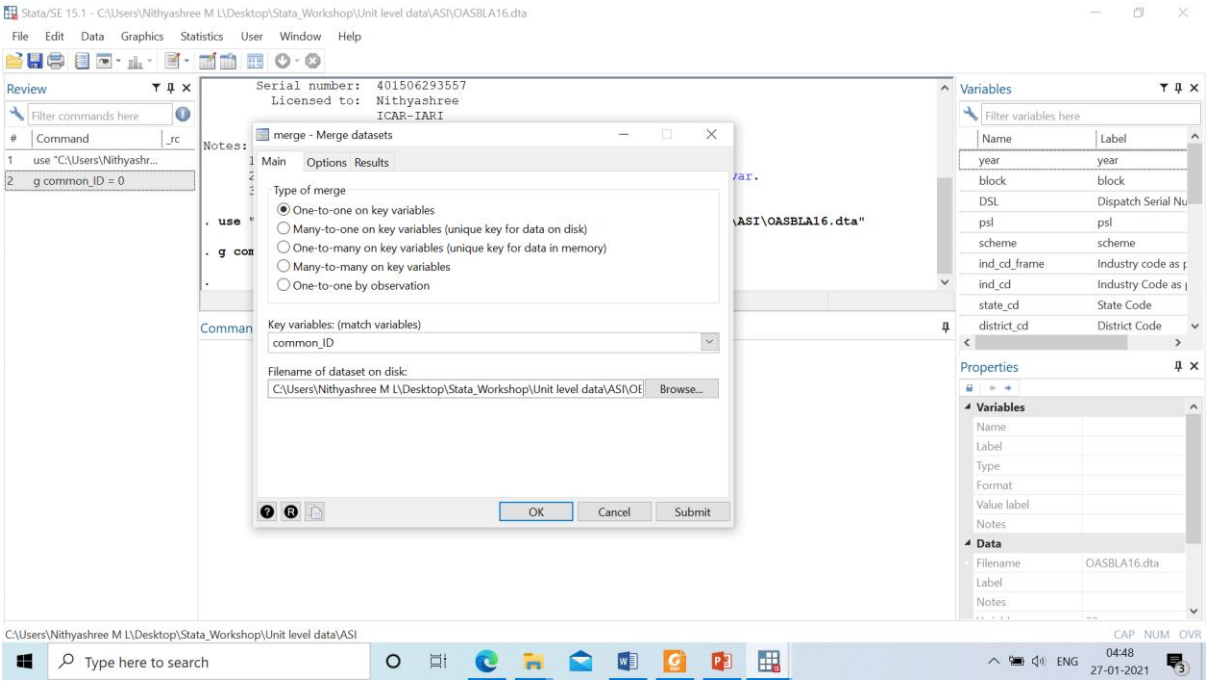
Similarly, to the unit-level data set command can be written as:

NSSO
reshape wide CropCode- PreHarvestSaleValue, i(id) j(SerialNo)

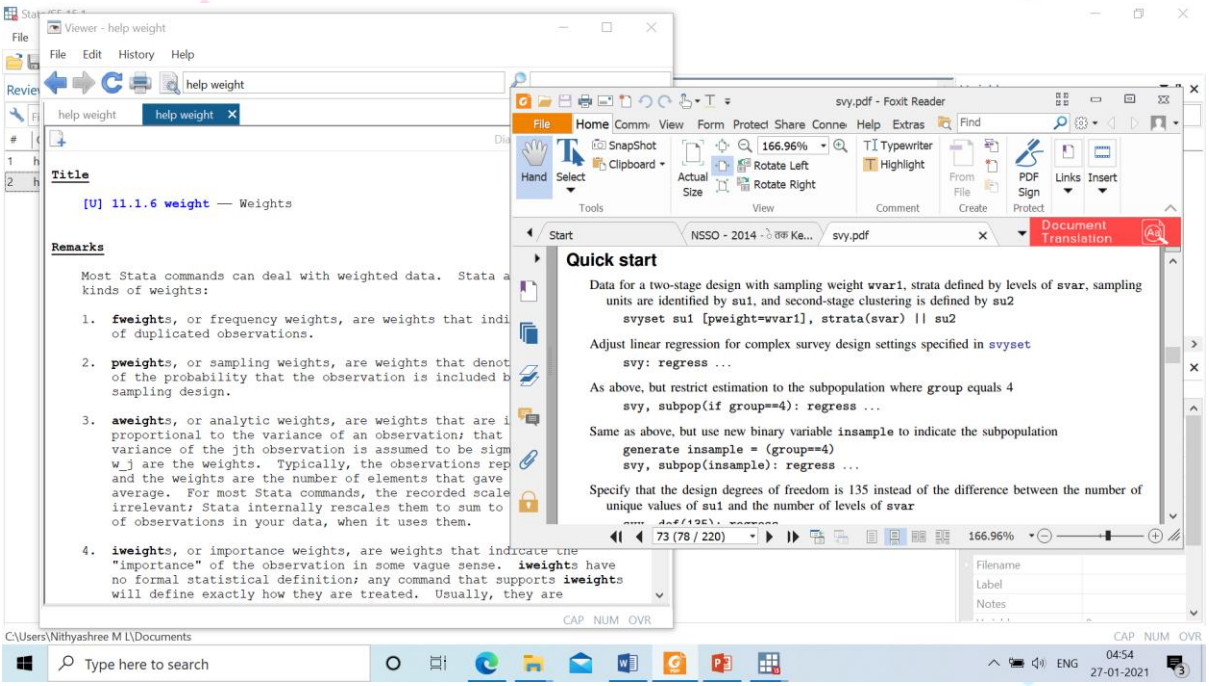
ASI
reshape wide GrossValueOpening - NVC , i(DSL) j(S_no)

In the next step, data files can be merged by using the common ID with the merge option one-to-one on key variable, as shown below





Based on the need for interpretation, multiplier options can be further explored by using the help command as shown below.



The detailed syntax for the ASI survey is given below, keeping students in mind, which will be useful to practice reshaping and merging.

```
      name:  <unnamed>
       log:  F:\MOSPI_ASI_UNIT\2015-16\Data\M_1516.smcl
  log type:  smcl
 opened on:  21 Oct 2019, 12:54:11

. save "F:\MOSPI_ASI_UNIT\2015-16\Data\B.dta"
file F:\MOSPI_ASI_UNIT\2015-16\Data\B.dta saved

. use "F:\MOSPI_ASI_UNIT\2015-16\Data\blkC201516.dta"

. drop Year

. tab S_no
```

S.no	Freq.	Percent	Cum.
1	36,930	9.37	9.37
2	46,763	11.87	21.24
3	52,954	13.44	34.69
4	42,085	10.68	45.37
5	40,209	10.21	55.57
6	3,532	0.90	56.47
7	51,758	13.14	69.61
8	54,993	13.96	83.57
9	9,008	2.29	85.85
10	55,727	14.15	100.00
Total	393,959	100.00	

```
. reshape wide GrossValueOpening Gross_ValueaddduetoRevaluation G_ValueActuala
> ddition G_Valuedepadj G_Valueclose Depuotobeginning Depprovideduringtheyear
> Depadjustment Depuptoyearend N_V_O NVC , i( DSL ) j( S_no )
(note: j = 1 2 3 4 5 6 7 8 9 10)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                   393959   ->   55727
Number of variables                  14   ->     112
j variable (10 values)             S_no   ->   (dropped)
xij variables:
                     GrossValueOpening   ->   GrossValueOpening1 GrossValueOp
> ening2 ... GrossValueOpening10
          Gross_ValueaddduetoRevaluation   ->   Gross_ValueaddduetoRevaluation1
>  Gross_ValueaddduetoRevaluation2 ... Gross_ValueaddduetoRevaluation10
                  G_ValueActualaddition   ->   G_ValueActualaddition1 G_ValueA
> ctualaddition2 ... G_ValueActualaddition10
                       G_Valuedepadj   ->   G_Valuedepadj1 G_Valuedepadj2 .
> .. G_Valuedepadj10
                        G_Valueclose   ->   G_Valueclose1 G_Valueclose2 ...
>  G_Valueclose10
                    Depuotobeginning   ->   Depuotobeginning1 Depuotobeginn
> ing2 ... Depuotobeginning10
                Depprovideduringtheyear   ->   Depprovideduringtheyear1 Deppro
> videduringtheyear2 ... Depprovideduringtheyear10
                       Depadjustment   ->   Depadjustment1 Depadjustment2 .
> .. Depadjustment10
                      Depuptoyearend   ->   Depuptoyearend1 Depuptoyearend2
>  ... Depuptoyearend10
                                N_V_O   ->   N_V_O1 N_V_O2 ... N_V_O10
                                  NVC   ->   NVC1 NVC2 ... NVC10
-----------------------------------------------------------------------------

. save "F:\MOSPI_ASI_UNIT\2015-16\Data\C.dta"
\ file F:\MOSPI_ASI_UNIT\2015-16\Data\C.dta saved
```

```
. use "F:\MOSPI_ASI_UNIT\2015-16\Data\blkD201516.dta"

. drop Year

. tab S_No
```

S.No	Freq.	Percent	Cum.
1	43,434	6.24	6.24
2	7,693	1.10	7.34
3	21,317	3.06	10.40
4	46,962	6.74	17.14
5	21,202	3.04	20.19
6	34,421	4.94	25.13
7	48,998	7.03	32.16
8	53,187	7.64	39.80
9	48,309	6.94	46.73
10	47,374	6.80	53.54
11	54,010	7.75	61.29
12	47,454	6.81	68.10
13	29,978	4.30	72.41
14	48,651	6.98	79.39
15	51,354	7.37	86.76
16	54,031	7.76	94.52
17	38,157	5.48	100.00
Total	696,532	100.00	

```
. reshape wide OpenungRs ClosingRs , i( DSL ) j( S_No )
(note: j = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17)

Data                                long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                    696532   ->   54047
Number of variables                    5   ->      36
j variable (17 values)             S_No   ->   (dropped)
xij variables:
                                OpenungRs   ->   OpenungRs1 OpenungRs2 ... Openu
> ngRs17
                                ClosingRs   ->   ClosingRs1 ClosingRs2 ... Closi
> ngRs17
-----------------------------------------------------------------------------

. save "F:\MOSPI_ASI_UNIT\2015-16\Data\D.dta"
file F:\MOSPI_ASI_UNIT\2015-16\Data\D.dta saved
```

```
. use "F:\MOSPI_ASI_UNIT\2015-16\Data\blkE201516.dta"

. br

. drop Year

. reshape wide MandaysWorkedManuf MandaysWorkedNonManuf MandaysWorkedTotal Ave
> NumberPersonwork NoofMandayspaid WagessalariesRs , i( DSL ) j( S_No )
(note: j = 1 2 3 4 5 6 7 8 9)

Data                                  long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                      338475   ->   54346
Number of variables                      9   ->      56
j variable (9 values)                S_No   ->   (dropped)
xij variables:
                       MandaysWorkedManuf   ->   MandaysWorkedManuf1 MandaysWork
> edManuf2 ... MandaysWorkedManuf9
                    MandaysWorkedNonManuf   ->   MandaysWorkedNonManuf1 MandaysW
> orkedNonManuf2 ... MandaysWorkedNonManuf9
                       MandaysWorkedTotal   ->   MandaysWorkedTotal1 MandaysWork
> edTotal2 ... MandaysWorkedTotal9
                     AveNumberPersonwork   ->   AveNumberPersonwork1 AveNumberP
> ersonwork2 ... AveNumberPersonwork9
                          NoofMandayspaid   ->   NoofMandayspaid1 NoofMandayspai
> d2 ... NoofMandayspaid9
                          WagessalariesRs   ->   WagessalariesRs1 WagessalariesR
> s2 ... WagessalariesRs9
-----------------------------------------------------------------------------
```

```
. reshape wide ItemCode Unit_Quantity_code QtyCons Purchase_Value Rate_PerUnit
>  , i( DSL ) j( Sno )
(note: j = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 2
> 6 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
>  52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
> 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101
>  102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
>  121 122 123 124 125 126 127 128 129 130 131 132 133)

Data                                  long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                      541009   ->   53406
Number of variables                      9   ->     668
j variable (133 values)                Sno   ->   (dropped)
xij variables:
                                  ItemCode   ->   ItemCode1 ItemCode2 ... ItemCod
> e133
                      Unit_Quantity_code   ->   Unit_Quantity_code1 Unit_Quanti
> ty_code2 ... Unit_Quantity_code133
                                 QtyCons   ->   QtyCons1 QtyCons2 ... QtyCons13
> 3
                           Purchase_Value   ->   Purchase_Value1 Purchase_Value2
>  ... Purchase_Value133
                             Rate_PerUnit   ->   Rate_PerUnit1 Rate_PerUnit2 ...
>  Rate_PerUnit133
-----------------------------------------------------------------------------
```

```
. reshape wide ItemCode Unit_Qty QtyCons Pur_value R_Perunit , i( DSL ) j( Sno
>  )
(note: j = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 2
> 6 27 28 29 30 31 32 33 34 35 36 37 38 39)

Data                                long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                     29442   ->    8763
Number of variables                    8   ->     197
j variable (39 values)               Sno   ->   (dropped)
xij variables:
                                ItemCode   ->   ItemCode1 ItemCode2 ... ItemCod
> e39
                                Unit_Qty   ->   Unit_Qty1 Unit_Qty2 ... Unit_Qt
> y39
                                 QtyCons   ->   QtyCons1 QtyCons2 ... QtyCons39
                               Pur_value   ->   Pur_value1 Pur_value2 ... Pur_v
> alue39
                               R_Perunit   ->   R_Perunit1 R_Perunit2 ... R_Per
> unit39
-----------------------------------------------------------------------------
```

```
. reshape wide Item_code Unit_Qty Qty_Manuf Qty_Sold Gross_salevalue Excise_du
> ty Sales_taxVAT Others Subsidy Per_unit_Netsale_value Ex_FactvalOutput , i(
> DSL ) j( Sno )
(note: j = 1 2 3 4 5 6 7 8 9 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 2
> 7 28 29 30 31 32 33 34 35 36 37 38 39)

Data                                  long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                      127906   ->   43768
Number of variables                     15   ->     421
j variable (38 values)                 Sno   ->   (dropped)
xij variables:
                                 Item_code   ->   Item_code1 Item_code2 ... Item_
> code39
                                  Unit_Qty   ->   Unit_Qty1 Unit_Qty2 ... Unit_Qt
> y39
                                 Qty_Manuf   ->   Qty_Manuf1 Qty_Manuf2 ... Qty_M
> anuf39
                                  Qty_Sold   ->   Qty_Sold1 Qty_Sold2 ... Qty_Sol
> d39
                            Gross_salevalue   ->   Gross_salevalue1 Gross_salevalu
> e2 ... Gross_salevalue39
                               Excise_duty   ->   Excise_duty1 Excise_duty2 ... E
> xcise_duty39
                              Sales_taxVAT   ->   Sales_taxVAT1 Sales_taxVAT2 ...
>  Sales_taxVAT39
                                    Others   ->   Others1 Others2 ... Others39
                                   Subsidy   ->   Subsidy1 Subsidy2 ... Subsidy39
                    Per_unit_Netsale_value   ->   Per_unit_Netsale_value1 Per_uni
> t_Netsale_value2 ... Per_unit_Netsale_value39
                           Ex_FactvalOutput   ->   Ex_FactvalOutput1 Ex_FactvalOut
> put2 ... Ex_FactvalOutput39
-----------------------------------------------------------------------------

. save "F:\MOSPI_ASI_UNIT\2015-16\Data\J.dta"
file F:\MOSPI_ASI_UNIT\2015-16\Data\J.dta saved

. log close
      name:  <unnamed>
       log:  F:\MOSPI_ASI_UNIT\2015-16\Data\M_1516.smcl
  log type:  smcl
 closed on:  21 Oct 2019, 13:01:11
-----------------------------------------------------------------------------
```

Extraction and reshaping of unit-level data

For NSSO dataset

1. Extract the unit level data for the given dataset **level1** using the command given in the dofile.
2. From the given extracted Stata data file **level4**, identify i and j variable and reshape the data into a wide format.
3. Merge the reshaped file **level4** with **level1**

For ASI dataset

1. Extract the unit level data for the given dataset **OASBLA16** by using the information given in the layout.
2. Using the given extracted Stata data file **OBSBLA16**, identify i and j variable and reshape the data into a wide format.
3. Merge the reshaped file **OBSBLA16** with **OASBLA16**

References

National Sample Survey Organization. (2014). *Key Indicators of Situation of Agricultural Households in India*, January – December 2013. NSS 70th Round. Ministry of Statistics and Programme Implementation, Government of India, New Delhi.

Annual Survey of Industries. (2015-16). *Summary results*, Ministry of Statistics and Programme Implementation, Government of India, New Delhi.

Period Labor Force Survey data extraction using Stata

Subhash S P

Scientist, ICAR-National Institute of Agricultural Economics and Policy Research, New Delhi

1. Where to download Periodic Labour Force Survey what are the files u get.

PLFS report :
http://mospi.nic.in/sites/default/files/publication_reports/Annual_Report_PLFS_2018_19_HL.pdf

PLFS data:

http://mospi.nic.in/download-tables-data

http://mospi.nic.in/sites/default/files/reports_and_publication/PLFS_2018_2019/README_demo.pdf

2. Extracting data from text file

Set directories: define the path of the directories

global PLFS "C:\Users\pc 1\Desktop\PLFS Class"

cd "$PLFS"

Extraction layout: Data_LayoutPLFS.XLS

infix str file_id 1-4 schedule 5-7 str quarter 8-9 str visit 10-11 sector 12 state 13-14 district 15-16 nss_region 17-19 stratum 20-21 sub_stratum 22-23 sub_sample 24 fod_sub_region 25-28 fsu 29-33 sample_no 34 SSS_no 35 sample_hh_no 36-37 month_survey 38-39 response_code 40 survey_code 41 reason 42 hh_size 43-44 hh_type 45 religion 46 social_group 47 month_exp 48-55 informant_no 56-57 survey_date 58-65 time 66-69 nss 70-72 nsc 73-75 mult 76-85 occurance 86 using "FHH_FV.txt"

3. Labelling variables and from where do we get different codes

Labelling variables

```
label var file_id "File Identification"
label var schedule "Schdule"
label var quarter "Quarter"
```

4. Defining label values

label define sector 1 "rural" sector 2 "urban"

label value sector sector

5. How to combine household and unit level datasets (when to reshape and when to use one to many merge)?

Generating unique id

egen hhid = concat (fsu sample_no sss_no sample_hh_no)

Merging household file with individual file

merge 1:m hhid using "C:\Users\pc 1\Desktop\PLFS Class\FPER_FV_extracted.dta"

6. Using sampling weights

gen weight = mult/200 if nss!=nsc

gen weight = mult/100 if nss== nsc

tab hh_type [aweight=weight]

Practical Manual

Data Analysis with Stata

January 25-29, 2021

Division of Agricultural Economics,
ICAR-IARI, New Delhi

Course Director

Alka Singh
Professor and Head
Division of Agricultural Economics
ICAR-Indian Agricultural Research Institute
Pusa Campus, Delhi – 110 012
Email: asingh.eco@gmail.com

Course Coordinators

Aditya K.S.
Scientist
Division of Agricultural Economics
ICAR-Indian Agricultural Research Institute
New Delhi 110 012
E-mail: adityaag68@gmail.com

Nithyashree M L
Scientist
Division of Agricultural Economics
ICAR- Indian Agricultural Research Institute
New Delhi-110012
Email: nithya.econ@gmail.com

Published By

NAHEP- Centre for Advanced Agriculture Science and Technology

ICAR- Indian Agricultural Research Institute, New Delhi

Website: www.nahep-caast.iari.res.in