PEA Stata Code Manual

I. Introduction

The new PEA 3.0 framework provides an integrated analysis of countries' progress in advancing on the updated World Bank mission statement: *ending extreme poverty and boosting shared prosperity on a livable planet.* Key to the PEA revamp is a standardization effort to help to streamline the analysis.[1]

This manual provides step-by-step guidance to World Bank staff for using a newly developed PEA Stata package, which automates the production of trends in poverty, shared prosperity and inequality, as well as their drivers and profiles. The package generates a full set of tables (`pea tables`) and figures (`pea figures`) to aide the core analytics in PEAs (shown in Annex 1: Full list of tables and figures). A smaller set of tables and figures are for inclusion in a standardized data annex (`pea core`) to ensure consistency and comparability across countries.

All outputs are automatically created by running the three commands, with minimum manual inputs needed from users. Parts of the codes can nevertheless be tailored to county contexts, for instance by specifying country-specific education or industry groups. The codes facilitate standardization of core analysis across countries and reproducibility of results. This guide explains how to install, set up, and execute the codes in Stata by following the below steps:

- Step 1: Installing and setting up the pea Stata package
- Step 2: Preparing and harmonizing survey data
- Step 3: Create additional variables
- Step 4: Run the pea commands

II. Installation and Setup

Step 1: Install the pea Stata Package

The PEA package can be downloaded from Github (click on "Code" on the top right and "Download ZIP"). Proceed to store the unzipped file in your folder of choice. The file location needs to be specified in the code later on (see Annex 2: Example code (adopath +)).[2]

The package includes the three main commands (`pea core`, `pea tables`, `pea figures`) which produce one Excel output file each. Additional ado-files allow each table and figure to be created separately. The whole package needs to be downloaded in order to produce the full output. This is because some of the ado-files run calculations and prepare and download data in the background. Ensure internet access is enabled, as additional data files are downloaded automatically from pip or datalibweb.

A data sub-folder needs to be extracted from the package and stored in the correct location on your device. A few additional datasets are required to run the full package, which are included in the pea

[1] See the PEA website for resources and tools for PEAs.
[2] An integration to be downloaded directly from Stata for automatic installation is planned.

package.[4,5] The sub-folder within the package can be found here: "/pea/Stata/personal/**pea**". Importantly, this **pea** sub-folder (including the Scorecard_Summary_Vision folder) needs to be moved into the personal folder of your Stata system directory (which can be found by entering "sysdir" in Stata): e.g. "c:/ado/personal/**pea**".

Step 2: Access and Prepare Survey Data

The pea Stata package runs on survey data harmonized according to the Global Monitoring Database (GMD).[7] Proceed to one of the following next steps.

(2a) Using pre-harmonized GMD data

If the survey data is already included in the GMD, access it in Stata using the `datalibweb` command (see Step 2 in Annex 2: Example code).[10] Using `datalibweb` has the advantage of retrieving multiple years of data easily and consistently. You may proceed to Step 3.

(2b) If survey data needs to be harmonized and added to GMD

If the latest survey is not already harmonized in the GMD, country teams should work with the regional stats team to add the latest survey data into the database. If using multiple rounds of data for analysis, the new data needs to be appended to previous datasets (which can be accessed through `datalibweb`) before running the pea command, to include trends and changes in the tables and figures.

Step 3: Create additional variables and check data

Once all surveys have been GMD-harmonized (step 2), only the poverty lines and PPP-adjusted welfare aggregates need to be created manually. The specific variables that need to be created are: welfareppp, pline215, pline365, pline685, natline (and natline2—in case that there is a second national poverty line).[11] Step 3 in Annex 2: Example code provides an example code to construct these additional variables. The

---

[4] The following are included in the data folder: CLASS.dta, which includes country classifications, updated in July every year (https://github.com/GPID-WB/Class/tree/master/OutputData); exposure_vulnerability_2021.dta, which contains data needed to calculate exposure and vulnerability to climate-related hazards (accessed from the reproducibility package of the Poverty, Prosperity and Planet Report 2024); UNESCO.dta, which is used in the calculation for the multidimensional poverty measure (accessed through datalibweb); Scorecard_Summary_Vision, which contains excel files for each corporate scorecard indicator (accessed from World Bank Scorecard website.)

[5] The package also installs several additional Stata commands. The following Stata packages are installed automatically when running the pea code for the first time (and don't need to be installed separately): `apoverty, ineqdeco, svylorenz, fastgini, glcurve, alorenz, povdeco, fs, groupfunction, drdecomp, adecomp, pip, schemepack, colorpalette, geoplot, palettes, colrspace, moremata`. The command `sedecomposition` cannot be installed through Stata's SSC and is therefore included in the pea package (pea/Stata/plus/s). The sedecomposition package has been accessed from here: https://github.com/vavalomi/stata_tools/blob/master/sedecomposition/sedecomposition.ado

[7] For more information, visit the GMD website.

[10] If this is your first time accessing datalibweb, visit https://datalibweb2.worldbank.org/home to create a token to access `datalibweb` through Stata. See here for datalibweb guidelines, and the D4G website for more instructions.

[11] Importantly, verify that the correct deflators are used.

PEA package also includes example do-files that can be used as templates to run the codes ("pea/Stata/Example").

The pea commands have an option "`setting(GMD)`", which creates an additional set of variables. If the option `setting(GMD)` is not specified, variables "married" and "nowork" need to be created. Table 1 displays code to create these, and the other additional variables (such as "welfppp").

Table 1 lists all variables that are needed to run the pea commands in Stata. It is critically important to check and confirm that all variables in the GMD-harmonized dataset take on the correct variable type, value labels, and only the allowed values. Specifically, all variables should be numeric and variable labels and value labels need to be defined. This is to ensure proper presentation and minimal post-editing in tables. For example, the value label of the subnational region variable should be in the format in which region names should be displayed in the table output (Annex 2: Example code shows an example in Step 3).

The type, value labels and values of all variables need to be consistent across all surveys. This should be carefully checked, in particular, when appending new survey data to pre-harmonized GMD data. Household identifiers might be repeated between surveys over time, even though it is not panel data. In this case, please create year-specific identifiers.

Note that even when not all variables are in the dataset or correctly defined, the pea commands will still run and produce all tables and figures where these variables are not required.

Table 1: List of variables used by the PEA command

Description	Name	Values	Code
1) Variables that need to be created			
Welfare aggregate (divided by CPI (2017) and ICP (2017), and 365 days)	welfppp	*integer*	gen welfppp = welfare/cpi2017/icp2017/365 (note: welfare should be in per-capita terms)
National poverty line (2017 LCU)	natline	*integer*	gen natline = X if year==
Second national poverty line (2017 LCU)	natline2	*integer*	gen natline2 = X if year==
Poverty line: $2.15 per day (2017 PPP)	pline215	2.15	gen pline215 = 2.15
Poverty line: $3.65 per day (2017 PPP)	pline365	3.65	gen pline365 = 3.65
Poverty line: $6.85 per day (2017 PPP)	pline685	6.85	gen pline685 = 6.85
Subnational ID. Make sure the variable is numeric and harmonized over all years.	subnatvar	integer (labeled)	e.g. encode subnatid, gen(subnatvar)

2) *Additional variables that are needed if setting(GMD) is not specified (see next section for more details on the code)*			
Household head dummy	head	0=Not household head; 1=Household head	gen head = relationharm==1 if relationharm~=.
Married dummy	married	0=not married; 1=married	gen married = marital==1 if marital~=.
Not working dummy	nowork	0=working; 1=not working	gen nowork = lstatus==2 | lstatus==3 if lstatus~=.
3) Variables are needed to obtain the full set of outputs			
Household identifier	hhid	*integer*	
Individual identifier	pid	*integer*	
Year of survey	year	*integer (4 digits)*	
Poverty specific weights	weight_p	*integer*	
Welfare aggregate (LCU)	welfare	*integer*	
Relation to household head	relationharm	1=Household head; 2=Spouse; 3=Child; 4=Parents; 5=Other relative; 6=Non-relative	
Age of individual	age	*integer*	
Sex of individual	male	0=female; 1=male	
Household size	hsize	*integer*	
Marital status of individual	marital	1=married; 2=never married; 3=living together; 4=divorced/separated; 5=widowed	
Highest level of education completed (4 categories)	educat4	1=No education; 2=Primary (complete or incomplete); 3=Secondary (complete or incomplete); 4=Tertiary (complete or incomplete)	
Urban indicator	urban	1=urban; 2=rural	
Currently in school dummy	school	0=not currently in school; 1=currently in school	
Labor force Status	lstatus	1=employed ; 2=unemployed; 3=not in labor force	

Sector/industry of employment (4 categories) [NOTE: For some countries this is industrycat4_year]	industrycat4	1=Agriculture; 2=Industry; 3=Services; 4=Other	
Employment status	empstat	1=Paid employee; 2=Non-paid employee; 3=Employer; 4=Self-employed	
Improved water access	imp_wat_rec	0=No; 1=Yes	
Improved sanitation facility	imp_san_rec	0=No; 1=Yes	
Access to electricity	electricity	0=No; 1=Yes	
Ownership of a television	tv	0=No; 1=Yes	
Ownership of a car	car	0=No; 1=Yes	
Ownership of a cellphone	cellphone	0=No; 1=Yes	
Ownership of a computer	computer	0=No; 1=Yes	
Ownership of a fridge	fridge	0=No; 1=Yes	
Comparability between survey rounds	Comparability	*Integer*	

Step 4: Run pea commands

Once the pea package, data and additional variables have been prepared, users can follow the example in Annex 2: Example code under Step 4 to run the commands. The PEA package includes example do-files that can be used as templates to run the codes ("pea/Stata/Example"). Note that the prepared data and the adopath to the pea package need to be called every time before a pea code is run (as in the example).

The syntax of the three main programs inside the `pea` code is outlined below. Stata help files are available for all commands. Each table or figure can also be produced separately, using the table/figure specific command (see Step 4 in Annex 2: Example code). All tables and figures include standardized tables notes which can be customized as needed.

Table 2 provides details on the main options that need to be specified to produce output. Option `setting(GMD)`, when specified, automatically creates all the input variables needed to produce the entire core analytics output. Furthermore, when specified, a number of additional options do not need to be specified. These options are listed in Table 3. The `setting(GMD)` option works only in conjunction with the main pea commands, and not the table or figure specific commands. For the table or figure specific commands, those options in Table 3 need to be specified that relate to the specific table or figure.

Which options are required for which table or figure is specified in the help files (accessible through `help pea core`, `help pea tables`, `help pea figures`, `help pea table10`, `help pea figure2`, etc.). Not specifying `setting(GMD)` can be useful when users want to use categories and values for certain variables that differ from the harmonized default values listed in Table 1.

1. `pea core`

This first code (in Stata: `pea core`) generates a standardized data annex which is expected to be included in all PEAs. The annex includes main poverty and shared prosperity indicators, as well as multidimensional and sub-group (e.g. by age or education) poverty rates and poverty profiles. The code also benchmarks core statistics for a selected group of peer countries, the country's region and income group averages. Peer countries can be specified by the user and a benchmarking tool is available to support the selection of countries. Growth incidence curves and the Datt-Ravallion decomposition complement the core outputs.[13]

```
pea core [weight] [if exp] [in exp] [,Year(varname numeric)
NATWelfare(varname numeric) NATPovlines(varlist numeric)
PPPWelfare(varname numeric) PPPPovlines(varlist numeric)
SETting(string) BYInd(varlist numeric) ONELine(varname numeric)
ONEWelfare(varname numeric) missing latest within3 BENCHmark(string)
spells(string) excel(string) save(string)]
```

2. `pea tables`

The second set of outputs is produced using the `pea tables` command. These expand on the core outputs, providing more detailed statistics on poverty and shared prosperity over the last years, the extent of vulnerability to poverty, poverty profiles as well as drivers of welfare changes. The latter examines how inclusive economic growth has been to those in the bottom of the income distribution, as well as decompositions of welfare changes into within and between group shifts.

```
pea tables [weight] [if exp] [in exp] [, Country(string) Year(varname
numeric) natwelfare(varname numeric) natpovlines(varlist numeric)
pppwelfare(varname numeric) ppppovlines(varlist numeric)
oneline(varname numeric) onewelfare(varname numeric) setting(string)
byind(varlist numeric) latest within3 missing benchmark(string)
spells(string) excel(string) save(string)]
```

3. `pea figures`

The third output includes a series of figures that are produced using the `pea figures` command. Figures include a comparison of poverty rates and GDP per capita with other countries, changes in poverty

[13] See benchmarking tool and guidance note to identify benchmarking countries.

and inequality over time, population composition by income decile, profiles, and exposure to risk from climate-related hazards.[14] Options that are specific to the `pea figures` command are listed in Table 4.

```
pea figures [if] [in] [weight], [natwelfare(varname)
natpovlines(varlist) pppwelfare(varname) ppppovlines(varlist)
year(varname) setting(string) byind(varlist) oneline(varname)
onewelfare(varname) missing country(string) within(integer) combine
comparability(varname) benchmark(string) spells(string) noequalspacing
scheme(string) palette(string) welfaretype(string) YRange0
excel(string) save(string)]
```

Table 2: Main options

Setting	Description	*Entry* or `variable name`
1) Required options		
Country()	PEA country	*3-letter country code*
natwelfare()	Welfare aggregate in current or constant LCU	`welfare`
natpovlines()	National poverty line, multiple entries allowed	`natline`
pppwelfare()	Welfare aggregate in PPP terms	`welfareppp`
ppppovlines()	International poverty lines in PPP terms	`pline215` `pline365` `pline685`
year()	Year variable in survey dataset	`year`
onewelfare()	Main welfare variable used for comparisons (i.e. Table 1, Table 7, Table 14, Table A1, Table A2, Table A4, Table A4, Figure A1)	`welfareppp`
oneline()	Main poverty line used for comparisons (i.e. Table 1, Table 7, Table 14, Table A1, Table A2, Table A4, Table A4, Figure A1)	`pline685`
2) Optional options		
byind()	Geographic units for disaggregation, multiple entries allowed. Note that while optional, it is used in many tables.	`urban subnatvar`
urban()	Variable that indicates the urban indicator	`urban`
benchmark()	Countries to be benchmarked against, multiple entries allowed	*3-letter country codes*
setting()	If GMD option is specified, harmonized variables are created, and additional options (Table 3) in do not need to be specified.	*GMD*
spells()	Years to be used when calculating or showing changes across periods (such as Growth Incidence Curves), multiple entries allowed separated by semicolon	*year1 year2; year2 year3*

[14] See the Corporate scorecard website and CCG dashboard on the risk from climate-related hazards indicator.

missing	Missing values are reported separately. Otherwise, they are treated as non-existent.	
latest	Includes only the most recent available data.	
Within3	Limits analysis to data from benchmark countries within 3 years of the target year.	
excel()	Path of directory where and name under which output file is saved. If it is not specified, the file is saved in "Temp". Note that if the option is specified, the excel file needs to exist before the code is run (empty).	*Path and file name*
save	Specifies the file path for saving results.	

Additional options, if `setting(GMD)` is not specified. Note that these are relevant for the `pea core`, `pea tables` and `pea figures` commands.

```
hhhead(varname numeric) edu(varname numeric) married(varname numeric)
school(varname numeric) services(varlist numeric) assets(varlist
numeric) hhsize(varname numeric)  hhid(string) pid(string)
industrycat4(varname numeric) lstatus(varname numeric) empstat(varname
numeric)
```

Table 3: Additional options if setting(GMD) is not specified

Setting	Description	Default under setting(GMD)
age()	Age of individual	`age`
male()	Sex of individual	`male`
married()	Marital status (variable is based on marital)	`married`
services()	Services to be included in the analysis. Default are access to improved water, access to improved sanitation, access to electricity.	`imp_wat_rec imp_san_rec electricity`
assets()	Assets to be included in the analysis	`tv car cellphone computer fridge`
hhsize()	Household size	`hsize`
head()	Household head indicator (Variable is based on variable relationharm)	`head`
hhid()	Household identifier	`hhid`
pid()	Individual identifier	`pid`
edu()	Education level	`educat4`
school()	Currently in school dummy	`school`
lstatus()	Dummy if respondent does not work (Variable is based on variable lstat)	`nowork`
industrycat4()	Industry categories	`industrycat4`
empstat	Employment status of respondent	`empstat`

Table 4: Additional optional figure options

Setting	Description	Entry or variable name
within()	Specifies the number of years before and after the pea survey year, to define which surveys from other countries should be used (e.g. in scatter plots on inequality). Default is 3, and value should be less than 10.	*integer*
combine	When specified, figures with multiple panels are combined to one figure with only one legend.	
comparability()	This variable denotes which survey rounds are comparable over time. Non-comparable survey rounds are not connected in figures.	`comparability`
noequalspacing	When specified, figures show gaps between years on the x-axis proportional to their distance. Default is to display constant gaps between years, regardless of how far away years are. The option can be useful if gaps between survey-years are large.	
scheme()	Sets the scheme, specifying the overall look of the figures. Default is "white_tableau".	*string*
palette()	Sets the color palette for figures. Default is "tab20". See Annex 2: Example code for an example of a custom set of colors.	Either *string* (e.g. *vividis)* or list of colors.
welfaretype()	Can be used to specify whether the survey uses income or consumption to calculate welfare. Figures showing scatters of inequality display different symbols for countries with consumption or income aggregates.	*CONS* or *INC*
yrange0	When specified, the y-axis in figures 1, 9a and 10a will be forced to start at 0.	

III. Caveats and FAQ

- Importantly, when the nominal welfare aggregate is passed into the onewelfare() option, growth incidence curves will show growth in welfare in nominal, not in in real terms. In this case, it is advised to redo the growth incidence curves separately (e.g. `pea table11, welfare(welfppp) spells(2018 2022) year(year) setting(GMD) graph`).
- When there are multiple survey periods, make sure that welfare in LCU and national poverty lines are both expressed in current prices or both in constant prices.
- Note that when national welfare in current prices is specified in the `oneline()` option, growth incidence curves will not reflect changes in real-terms. Therefore, specifying welfare in constant terms in the `oneline()` is encouraged, or re-doing the growth incidence curves separately in constant terms (`pea table11, pea figure3`) is recommended.
- It is important that the correct temporal and spatial deflators are used.
- Poverty numbers are based on population numbers from the respective surveys. Note that this may cause slight discrepancies between poverty numbers reported in PIP and in PEBs, which are based on population from WDI.

- Labor force related variables can have different recall periods. For example, in GMD, some survey data use the variable *lstatus* (which has a 7-day recall period) and some use *lstatus_year* (which has a 1-year recall period). It is up to country teams to decide whether variables are comparable over time (e.g. even when recall periods are different). For the example of labor force status, if different recall periods are deemed to be comparable, one variable would need to be defined which has both variables' values (e.g. `replace lstatus = lstatus_year if lstatus==.`)
- The code produces output, regardless of the number of observations for each cell. It may be that a subgroup indicator is based on very few observations. Country teams need to decide if these are to be shown or not.
- The MPM is retrieved from the GMI. New surveys in the GMD might not be immediately included in the GMI (updated usually in April and October).
- For growth incidence curves disaggregated by urban and rural areas, the calculation is performed for the welfare distribution within that area only.
- The "Other" sector in the variable industrycat4 is based on a country-specific harmonization from the GMD which can vary across countries.
- Note that not all surveys of a country in the GMD are comparable over time, due to updated methodologies. The variable "comparability" indicates which surveys are comparable. When specified, the `comparability` option in the `pea figures` commands shows breaks between non-comparable survey spells.

FAQ

Q1: Why are my tables missing some indicators?

✓ Check if all required variables (Step 3) are present in the dataset.

✓ Ensure numerical variables are correctly formatted (encode string variables if needed).

Q2: Can I use country-specific variables instead of GMD harmonized variables for education, sector, etc?

✓ Yes, not specifying the option `setting(GMD)` allows to insert country-specific variables in the options specified in Table 3. For example, the globally harmonized education variable in the GMD database takes on only 4 values (educat4), but a different, country-specific variable can be inserted in option `edu()`.[15] See example 2b under Step 4 in Annex 2: Example code.

Q3: How do I define a custom color palette for figures?

[15] In the GMD, education by default follows the International Standard Classification of Education (ISCED) mappings.

✓ Use the `palette()` option, see example under Step 4 in Annex 2: Example code.

Q4: How do I include multiple years for analysis?

✓ Make sure that all variables are consistently specified over all years

✓ Options `years()` and `spells()` specify which years are used in the analysis. Use `spells(year1 year2; year3 year4)`, to select which pairs of years should be used to display growth rates (for example in growth incidence curves).

Annex 1: Full list of tables and figures

Standardized data annex (pea core):

- Table A1: Core poverty and equity indicators
- Table A2: Poverty indicators by subgroup
- Table A3: Benchmarking of poverty and inequality
- Table A4: Poverty profiles
- Figure A1: Growth Incidence Curves
- Figure A2: Datt-Ravallion decomposition

Full set of tables (pea tables):

- Table 1: core poverty indicators.
- Table 2: poverty rate and share of poor by area and region.
- Table 3: subgroup poverty rates.
- Table 6: Multidimensional Poverty Measure (World Bank)
- Table 7: vulnerability for poverty.
- Table 8: core inequality indicators.
- Table 9: vision indicators (corporate scorecard).
- Table 10: benchmarking of poverty and inequality.
- Table 11_1 / Figure 11_1: Growth Incidence Curve (All sample).
- Table 11_2 / Figure 11_2: Growth Incidence Curve (Rural).
- Table 11_3 / Figure 11_3: Growth Incidence Curve (Urban).
- Table 12b: decomposition of poverty changes: growth and redistribution - Datt-Ravallion decomposition
- Table 12c: decomposition of poverty changes: growth and redistribution - Shorrocks-Kolenikov decomposition
- Table 13: decomposition of poverty changes: Huppi-Ravallion decomposition.
- Table 14: profiles of the poor
- Table 15: gender typology (to come).

Full set of figures (pea figures):

- Figure 1: Poverty rates by year lines.
- Figure 2: Poverty and GDP per capita scatter.
- Figure 3: Growth Incidence Curve.
- Figure 4: Decomposition of poverty changes: growth and redistribution: Datt-Ravallion and Shorrocks-Kolenikov.
- Figure 5: Decomposition of poverty changes: growth and redistribution: Huppi-Ravallion .
- Figure 6: GDP per capita GDP - Poverty elasticity.
- Figure 7: Welfare Figure with poverty line breakdowns.
- Figure 9a: Inequality by year lines.

- Figure 9b: GINI and GDP per capita scatter.
- Figure 10a: Prosperity gap by year lines.
- Figure 10b: Prosperity gap scatter (line-up).
- Figure 10c: PG (survey) and GDP per capita scatter.
- Figure 12: Decomposition of growth in prosperity gap.
- Figure 13: Distribution of welfare by deciles
- Figure 14: Multidimensional poverty: Multidimensional Poverty Measure (World Bank).
- Figure 15: Climate risk and vulnerability.
- Figure 16: Gender typology (to come)

Annex 2: Example code

The following code produces the main tables, the standardized data annex and figure outputs for Guinea-Bissau.

```
// Set path where the survey data is stored, or should be stored if accessing from datalibweb
global pea_path "C:/Users/data/ "

// Step 2: Access data (either from survey, or from datalibweb)
datalibweb, country(GNB) year(2018 2021) type(gmd) mod(all) clear

// Step 3: Preparation of additional variables
// Welfare aggregate to compute international poverty rates
gen welfppp = welfare/cpi2017/icp2017/365
gen pline215 = 2.15
gen pline365 = 3.65
gen pline685 = 6.85
// National poverty line: Adjust according to the PEA country
gen natline = 298083.5 if year == 2021
replace natline = 271071.8 if year == 2018
la var pline215 "Poverty line: $2.15 per day (2017 PPP)"
la var pline365 "Poverty line: $3.65 per day (2017 PPP)"
la var pline685 "Poverty line: $6.85 per day (2017 PPP)"
la var natline "Poverty line: 298,083.5 per year (2017 LCU)"
// Cleaning of region variable: Adjust according to the PEA country
split subnatid, parse("-") gen(tmp)
replace tmp2 = ustrlower( ustrregexra( ustrnormalize( tmp2, "nfd" ) ,
"\p{Mark}", "" ) )
replace tmp2 = " bolama/bijagos" if tmp2 == " bolama_bijagos"
replace tmp2 = proper(tmp2)
encode tmp2, gen(subnatvar)

save "${pea_path}/data/GNB_GMD_clean.dta", replace

// Step 4: Run pea commands
// 1. Main tables code:
//Set ado path
adopath + "${pea_path}/ado/pea/Stata/plus"
//Call data
use "${pea_path}/data/GNB_GMD_clean", clear
//Run code with options
pea tables [aw=weight_p], c(GNB)  year(year)                    ///
          natw(welfare) natp(natline)                           ///
          pppw(welfppp) pppp(pline365 pline215 pline685)        ///
          onew(welfppp) oneline(pline365)                       ///
          byind(urban subnatvar)                                ///
```

```
           benchmark(CIV GHA GMB SEN)                                  ///
           setting(GMD) missing                                        ///
           spells(2018 2021)

// 2a. Appendix tables code:
//Set ado path
adopath + "${pea_path}/ado/pea/Stata/plus"
//Call data
use "${pea_path}/data/GNB_GMD_clean", clear
//Run code with options

pea core [aw=weight_p], c(GNB) year(year)                              ///
           natw(welfare) natp(natline)                                 ///
           pppw(welfppp) pppp(pline365 pline215 pline685)              ///
           onew(welfppp) oneline(pline365)                             ///
           byind(urban subnatvar)                                      ///
           benchmark(CIV GHA GMB SEN)                                  ///
           setting(GMD) missing                                        ///
           spells(2018 2021)


// 2b. Appendix tables code without setting(GMD):
//Set ado path
adopath + "${pea_path}/ado/pea/Stata/plus"
//Call data
use "${pea_path}/data/GNB_GMD_clean", clear
//Run code with options
pea core [aw=weight_p], c(GNB) year(year)                              ///
           natw(welfare) natp(natline)                                 ///
           pppw(welfppp) pppp(pline365 pline215 pline685)              ///
           onew(welfppp) oneline(pline365)                             ///
           byind(urban subnatvar)                                      ///
           benchmark(CIV GHA GMB SEN)                                  ///
           missing spells(2018 2021)                                   ///
           age(age) male(male) hhhead(head) edu(educat4)               ///
           urban(urban) married(married) school(school)                ///
           services(imp_wat_rec imp_san_rec electricity)               ///
           assets(tv car cellphone computer fridge)                    ///
           hhsize(hsize) hhid(hhid) pid(pid)                           ///
           industrycat4(industrycat4)                                  ///
           lstatus(nowork) empstat(empstat)

// 3. Figures code:
//Set ado path
adopath + "${pea_path}/ado/pea/Stata/plus"
//Call data
use "${pea_path}/data/GNB_GMD_clean", clear
//Run code with options
```

```
pea figures [aw=weight_p], c(GNB) year(year)                          ///
           natw(welfare) natp(natline)                                ///
           pppw(welfppp) pppp(pline365 pline215 pline685)             ///
           onew(welfppp) oneline(pline215)                            ///
           byind(urban) benchmark(CIV GHA GMB SEN AGA)                ///
           spells(2010 2018; 2018 2021)                               ///
           setting(GMD) urban(urban)  within(3)                       ///
           comparability(comparability) welfaretype(CONS)             ///
           combine nonotes

// 4. Running a single figure and defining own color palette
use "${pea_path}/data/GNB_GMD_clean", clear
adopath + "${pea_path}/ado/pea/Stata/plus"
local custom_palette = "#337ab7 #5cb85c #5bc0de #f0ad4e #d9534f #e6e6e6
#286090 #449d44 #31b0d5 #ec971f #c9302c"

pea figure2 [aw=weight_p], c(GNB) year(year)                          ///
           onew(welfppp) onel(pline215)                               ///
           benchmark(CIV GHA GMB SEN) palette(`custom_palette')


// 5. Running a single table with LCU poverty line
use "${pea_path}/data/GNB_GMD_clean", clear
adopath + "${pea_path}/ado/pea/Stata/plus"

// When running single tables, setting(GMD) is not called, and additional
variables need to be created
gen head = relationharm==1 if relationharm~=.
la def head 1 "HH head"
la val head head
gen nowork = lstatus==2|lstatus==3 if lstatus~=.
gen married = marital==1 if marital~=.

pea table14 [aw=weight_p], welfare(welfare) povlines(natline)         ///
           year(year) missing urban(urban)                            ///
           age(age) male(male) hhhead(head)                           ///
           edu(educat4) married(married) school(school)               ///
           services(imp_wat_rec imp_san_rec electricity)              ///
           assets(tv car cellphone computer fridge)                   ///
           hhsize(hsize) hhid(hhid) pid(pid)                          ///
           industrycat4(industrycat4) lstatus(nowork)                 ///
           empstat(empstat)
```