Chapter 7

Analysing PIAAC Data with Stata

François Keslair

Abstract This chapter explains the basics of analysing data from the Programme for the International Assessment of Adult Competencies (PIAAC) with Stata. It describes how to import the PIAAC datasets into Stata, gives an overview of the different categories of variables available in these datasets, and mentions a number of features of some types of variables about which users should be aware. The different types of missing values are explained. Routines frequently used with PIAAC datasets are presented using examples. Furthermore, the chapter is devoted to the use of plausible values variables and to the computation of imputation errors and sampling errors. In particular, it presents *repest*, a Stata ado file written to facilitate the analysis of international skills assessments, such as PIAAC.

Stata is an integrated statistical analysis package designed for research professionals. It is particularly well suited for analysing the Organisation for Economic Co-operation and Development's (OECD) Programme for the International Assessment of Adult Competencies (PIAAC) survey (OECD 2013, 2016b, c). Among existing statistical software packages, Stata stands out as it is designed to operate on one dataset at a time, using a dataset that has been previously loaded in memory. With a one-dataset survey such as PIAAC, it brings a simplicity of use and computation speeds difficult to find elsewhere. Moreover, Stata users can benefit from *repest*, a Stata *ado* file developed at the OECD and designed to facilitate the analysis of international skills assessments.

Stata works as a command-line-driven software. It also includes a graphic user interface. Commands can be run—one command at a time—from a prompt located below the results window. This makes the preliminary exploration of a dataset both simple and interactive, in particular, because another window is dedicated to displaying the list of all variables. Commands can be regrouped and saved in

F. Keslair (✉)

Organisation for Economic Co-operation and Development, OECD, Paris, France
e-mail: francois.keslair@oecd.org

© The Author(s) 2020

D. B. Maehler, B. Rammstedt (eds.), *Large-Scale Cognitive Assessment*,
Methodology of Educational Measurement and Assessment,
https://doi.org/10.1007/978-3-030-47515-4_7

149

a do-file in a separated window. Do-files include files that create new functions that can enrich the Stata library. These files are called ado files, and *repest*, which is described below, is one of them. All native Stata functions can be accessed through the graphic interface. Any action from the graphic interface is translated into the equivalent command inside the prompt. This latter feature is of great help in generating commands with a complex syntax, especially the commands used to generate charts.

It should be noted that, the brief description of the interface aside, this chapter is not intended as an introduction to Stata (therefore, see, e.g. Kohler and Kreuter 2012). Rather, it will focus on how to use Stata with PIAAC. It is assumed that readers have at least a basic knowledge of Stata and know how to perform simple procedures such as loading a dataset and creating new variables. Users unfamiliar with Stata can obtain a good introduction to the package from the Stata help files. The Stata help files deserve a word of praise. They are well written, comprehensive, and represent one of the main assets of the software.¹ They are accompanied by documentation in PDF format that offers detailed examples for each Stata function, as well as by YouTube tutorial videos. These resources are relevant for all Stata users—from beginners to experts—and make working with Stata a very pleasant experience.

This chapter is structured as follows: The first section will describe basic management of the PIAAC dataset using Stata, commonly used procedures, and some pitfalls to avoid. The second section will present *repest*, a Stata macro designed to help perform data analysis in international skills assessments. The third section will describe in detail all of *repest*'s features. Examples are provided to illustrate the commands described and to present Stata routines useful in analyses of PIAAC data.

7.1 Basic Manipulation of PIAAC Data Using Stata

7.1.1 Importing the Datasets

PIAAC data are accessible in public use files (PUFs; see Chap. 4 in this volume) that can be downloaded from the data section of the [PIAAC website](https://www.oecd.org/skills/piaac/data/).² There is one single file for each participating country containing all publicly available variables. Unfortunately, PIAAC PUFs are not available in the Stata native *dta* format and have to be imported into STATA from the generic comma-separated-values (csv) versions. Because of the complexity of the datasets—most notably the encoding of missing values—this procedure is not straightforward without any loss

¹New users can simply type *help resources* in the command prompt in order to have access to all available help resources.

²<https://www.oecd.org/skills/piaac/data/>

of information. In order to simplify the access to the datasets, a Stata do-file is also available in the data section of the PIAAC website. Once all CSV files have been downloaded into a directory, this do-file imports and appends all PUFs and then formats all variables into a unique *dta* dataset ready to be used. This do-file will also work if the target directory contains only one country dataset.³

7.1.2 *Different Types of Variables*

The PUFs contain more than 1300 variables divided into three broad categories. With a few exceptions, most of these variables are either numeric or categorical variables. Note that categorical variables in Stata are numeric variables that accept value labels. The *lookfor* command provides an easy way to search for variables; it displays all variables containing the desired keywords in its name or label. The reader can also consult questionnaires and codebooks available online (OECD 2014, 2016a).

7.1.2.1 Respondent and Interviewer Inputs

This category covers respondents' answers to the background questionnaire and the cognitive instruments. Variables relating to the background questionnaire are named according to the position of the question to which they relate. For instance, *b_q01b* refers to Question 1b in Section B of the background questionnaire. Most of these variables were collected for the purpose of building the derived variables introduced before. Background questions might be useful for more detailed analysis, but data users should consider using them only when derived variables cannot be used in their place (OECD 2011). Users should be careful when employing them—and should be particularly careful to handle missing values correctly. For each of the cognitive test items, the PUFs also include the respondents' answer status, supplemented with a set of variables on item timing for those respondents who were assessed on the computer. These variables are named after the item identifier (a six-digit number), with the last letter in the variable name standing for the type of variable. For instance, E644002S is the scored response for item 644002. The cognitive variables should not be used in analysis without a very good knowledge of the design of the cognitive assessment. In addition, the PUFs also include answers to the observation module that was filled in by the interviewer following completion of the interview.

³For the rest of the chapter, it will be assumed that the loaded dataset includes all participating countries.

7.1.2.2 Variables Derived from the Survey Instruments

These variables are designed to be used directly in analysis, and data users should focus principally on this set. Derived variables include proficiency scores, education levels, earning variables, indexes, or occupation variables. The process of derivation can take many forms: from the very simple creation of gender and age interval variables to the human coding of occupation descriptions into ISCO categories, or the computation of plausible values of individual proficiency scores through an IRT and population model (see Chap. 3 in this volume). Derived variables have generally meaningful names such as *gender_r* or *readytolearn*. However, some derived variables were created by the survey software and are named following the same convention as the background questionnaire variables. *c_d05* is thus a derived variable created in Section C of the questionnaire after its fifth question. All derived variables bear variable labels including the word ‘derived’.

7.1.2.3 Auxiliary Variables

Auxiliary variables represent information about the survey workflow and the survey design, including the set of final weights and balanced repeated replicated (BRR) weights. Other than the replicate weights, auxiliary variables are not used in most analyses. Other auxiliary variables of interest include *pbroute*, which indicates whether the respondent took the paper-based or computer-based assessment, and the disposition codes, which indicate whether the observation is considered to be complete, or the reason why it is considered incomplete.

7.1.3 The Correct Handling of Missing Values

The handling of missing values is one of the main sources of mistakes in the analysis of PIAAC data. There are a number of different types of missing values that must be clearly distinguished. In Stata, missing values for numeric variables (including categorical variables) are coded with a letter preceded by a dot character. The function *missing(myvarname)* will yield 1 if and only if the observation has one of these missing value codes recorded in *myvarname*. The different types of missing values codes used in PIAAC are as follows:

- *.d* : the respondent *didn't know* how to answer.
- *.r* : the respondent *refused* to answer.
- *.n*: the answer cannot be interpreted and is considered as *non stated*.
- *.u*, *.a* , *.z* , *.m* : these codes are rarer and specific to some derived variables.
- *.v* : the respondent did not have to answer the question and a *valid skip* was recorded.
- *.* : the value is missing for other reasons.

All these codes except *valid skip* (and sometimes ‘.’) refer to values that are missing due to nonresponse or errors. In contrast, missing values coded as *valid skip* do not represent forms of nonresponse or other response errors. These values are missing by design (see also Chap. 2 in this volume). Variables with *valid skip* must be interpreted in the context of the questionnaire and assessment. The background questionnaire is a complex set of questions, and in order to prevent nonsensical or redundant questions, many questions are administered only when some specific conditions are satisfied.

First, questions are administered only to the population for which the question has a meaning. For instance, the set of questions in Section D in the background questionnaire (OECD 2014) collects information on the respondent’s current occupation, including income, and only respondents who are currently working are presented with the questions in this section. For all other respondents, the response to the variables in this section is imputed as *valid skip*. Thus, respondents who refused to answer the set of questions about their work income and respondents who are not working both have missing values in these questions. However, as they should not be confused with each other, they have missing values with different codes.

Second, questions are administered only if the information requested cannot be inferred from answers to previous questions. For instance, respondents are asked about their household composition only if they previously reported that there was more than one member in their household. Households with only one member are, by definition, single-person households. Respondents who live in single-person households are assigned *valid skips* for the question about household composition. In analysis, however, these *valid skips* must be assigned their true value—that is, the value code for single-person households.

In order to avoid problems with missing values, data users should, as a rule, check for the occurrence of missing values and should tabulate all categorical variables they use with the following command:

```
tabulate myvarname , missing
```

If *valid skips* are present, data users should consult the question section in the background questionnaires to determine whether observations with these missing codes result from the redundancy of the question or from it not being relevant to the particular respondent. In the former case, a new variable with the correct values should be created. It is important to note that derived variables can sometimes feature *valid skips*. In these cases, however, this coding is always due to the lack of relevance of the question and not from its redundancy.

The coding of missing values in the PIAAC dataset has nonetheless one exception. All ISCO and ISIC variables describing occupations and industry sectors, respectively, are coded as string variables. For these variables, all strings starting with 999 indicate codes for missing values.

7.1.4 Working with Plausible Values

PIAAC includes proficiency measures in three domains: literacy, numeracy, and problem solving in technology-rich environments. Proficiency scores in PIAAC proceed from a complex computation. One of the consequences of this model is the presence of imputation error, which requires the use of plausible values variables in order to account for it. Scores in each of the three assessment domains, *literacy*, *numeracy*, and *problem solving in technology-rich environments*, are described by ten different plausible values variables, numbered from 1 to 10. Any one of these variables will give an unbiased estimate of individual proficiency, but the full set is required in order to compute accurate standard errors of population estimates (for more details see Chap. 3 in this volume).

The PUFs include proficiency scores only in the form of point estimates. Other variables derived from proficiency scores must be created by data users. This is the case, for example, with the proficiency levels, which are often used to describe the distribution of proficiency scores. Categorical variables for the proficiency levels in literacy and numeracy have to be created using the following loop over the ten plausible values (the example relates to the creation of proficiency levels categorical variables for literacy):

```
forvalues i=1/10 {
    generate litlev`i' = (pvlit`i'>176) + (pvlit`i'>226) + ///
    (pvlit`i'>276) + (pvlit`i'>326) + (pvlit`i'>376) ///
    if missing(pvlit`i')==0
}
```

This short string of code provides the opportunity to discuss a common mistake. The brackets are used to create a function that yields 1 if and only if the predicate inside the brackets is true and 0 for all other observations in the dataset. Importantly, the predicate remains defined for observations in which *pvlit* is missing. In this situation, missing values are considered by Stata to be larger than any number, and, as a result, each inequality is true for observations with missing literacy scores. In this example the value 0 would have been created for observations in which the literacy score is missing, were it not for the *if* statement at the end that causes these observations to have a missing value. The ten *litlev* variables will then be defined based on each *pvlit* variable and will cover the five different proficiency levels. Respondents with their i^{th} plausible value scoring below the Level 1 threshold category (176) would have their i^{th} plausible value level assigned to 0.

7.1.5 Computing Point Estimates in Stata

One of the main advantages of working with Stata is the simplicity of its syntax. Once the dataset has been properly imported, most statistics can be computed using

a short command, with results being displayed in the Stata results window. While the computation of accurate standard errors requires using the *repest* macro, the computation of accurate point estimates requires only the name of the final weight *spfw0* to be mentioned in the command.⁴ Importantly, these computations are much faster than those required for accurate standard errors and are thus more convenient for preliminary work and for obtaining an overview of the PIAAC dataset. The following examples cover some frequent cases of data exploration:

```
(1) tabulate agegl0lfs if centry_e=="ENG" [aw= spfw0]
```

tabulate returns the frequencies of categorical or string variables. This command will give the age distribution of the English target population displayed in 10-year age bands. Importantly, in the absence of an *if* statement, the command would give the age distribution for all countries appearing in the dataset, with each country weighted according to the size of its target population.

```
(2) bysort centry_e: summarize pvlit1 [aw= spfw0]
```

summarize returns common statistics (average, standard deviation, minimum, and maximum) for a continuous variable. This command will describe the literacy distribution for the target population of each country in the dataset based on the first set of plausible values. The *bysort* prefix will loop the *summarize* command over all groups defined by the *centry_e* variable. To obtain unbiased estimates, it is sufficient to use only *pvlit1*. However, for reasons of efficiency, the average statistics published by the OECD represent the averages of each of the ten statistics associated with the ten sets of plausible values.

```
(3) bysort centry_e edlevel3: regress pvlit1 ib1.gender_r  
[aw= spfw0]
```

regress will give OLS regression coefficients of *pvlit1* on the *gender_r* variable, taking 1 as a reference. In plain language, it will estimate gender gaps in literacy proficiency. *bysort* accepts several variables, and in this case the regression will be computed for each country identified by *centry_e* and, within each country, for each of the education levels identified by *edlevel3*.

⁴It should be kept in mind that *spfw0* represents the number of persons that each observation represents in their country's target population. As a result, each country contributes to the statistics in proportion to its size. If the user works with pooled samples and wishes that all countries contribute equally, he or she will have to rescale all weights.

7.2 The *Repest* Command: Computing Correct Standard Errors

The main purpose of the *repest* ado file is the correct computation of standard errors of statistics in international skills surveys such as PIAAC.

Computing standard errors in PIAAC is not as straightforward as computing point estimates. Since the sampling structure in each PIAAC country is not purely random, but rather involves complex random draws from the sampling frame performed in several stages, it has to be taken into account in a particular way. To do so, PIAAC uses a replication method with 80 replicate weights to account for the resulting sampling error. These weights simulate alternative samples, and the comparison of all these alternative samples yields an estimation of sampling errors. All population statistics are affected by sampling error. In the case of proficiency scores, as mentioned above, imputation error also needs to be taken into account. The ten plausible values are all different imputations of the proficiency scores using the same model. Following the same principle underlying the BRR, the comparison of these ten plausible values with each other allows estimation of the magnitude of the imputation error.

The operating principle of BRR and plausible values is simple: an empirical distribution of a statistic is estimated by computing it as many times as there are BRR weights and plausible values and then drawing a standard error from this distribution. The core of *repest* is thus a loop going over all 80 BRR weights and the final weights (and the ten plausible values if applicable). This method is extremely flexible, as it can be applied to any statistics, but it is also slow: if there are no plausible values, 81 statistics must be computed, and when plausible values are present, 810 statistics must be computed.

7.2.1 Installing *Repest*

Repest can be installed from within Stata by using the following simple command:

```
ssc install repest, replace
```

The command will download the *repest* ado file, along with its help file, and install it into the user's ado directory.⁵ Once the installation is completed, the *repest* command can be used in the same way as any other command. The help file covers all options in detail: it can be accessed via the following command:

⁵*Repest* can also be downloaded from the data section of the PIAAC website. However, in this case, the user will have to install it in Stata manually.


```
help repest
```

7.2.2 General Syntax

Repest general syntax is framed as a meta-command surrounding the respective Stata command for the desired statistics. This section will cover only the properties of its main syntax and of its mandatory components. The description of options, many of them aimed at facilitating its use, will be addressed in the next section.

When using PIAAC data, the *repest* syntax is as follows:

```
repest PIAAC [if] [in] , estimate([stata:]cmd) [repest_options]
```

The left-hand side includes only the PIAAC keyword, along with any *if* or *in* statements. For efficient computations, *if* and *in* statements must be mentioned here rather than in the estimate argument. The PIAAC keyword instructs the program to load parameters associated with PIAAC:

- Final weight: *spfw0*
- Replicate weights: *spfw1-spfw80*
- Variance method: Jackknife 1 or 2, depending on variable *vemethodn*
- Number of replications: 80
- Number of plausible values: 10

It is important to note that *repest* will not work if any of these variables are missing or have been renamed.

The *estimate* main option is mandatory and will contain the full Stata command associated with the desired statistic and any specific option associated to this command. Any *eclass* Stata command will work. *eclass* commands are Stata estimation commands; they are characterised by the presence in the output of an estimation vector $e(b)$ and an associated variance matrix $e(V)$.⁶ By default, *repest* will compute the simulated empirical distribution of the $e(b)$ vector. Importantly, as some usual Stata commands are not *eclass* and do not return a $e(b)$ vector, *repest* includes built-in commands designed to replace them. As a result, the *estimate* argument can take two forms.

The first one is dedicated to built-in commands:

```
estimate(built_in_cmd [,cmd_options])
```

Available built-in commands are *means* for computing averages, *freq* for frequencies tables, *corr* for correlations, and *summarize* for describing univariate

⁶Which, incidentally, does not account for sampling or imputation error.

distributions. The syntax for these commands is kept simple, as shown in the examples below:

```
repest PIAAC, estimate(means pvlit@ ictwork)
repest PIAAC if ageg10lfs==4, estimate(freq edlevels3)
repest PIAAC, estimate(summarize pvlit@ pvnum@, stats(sd p50))
repest PIAAC, estimate(corr pvlit@ pvnum@)
```

The commands *means*, *summarize*, and *corr* can take any number of variables, while *freq* will accept only one variable at a time. The *stats* option in *summarize* is mandatory and contains keywords for the desired univariate statistics: in this case, the median and the standard deviation. The full list of possible statistics is available in the help file. Please note that the *if* statement in the *freq* example, which constrained the frequencies to be computed for respondents aged between 45 and 54 years, is mentioned before the comma, as mentioned above. Weights do not have to be mentioned, as they are automatically added once the PIAAC keyword is specified.

The second form of the *estimate* argument is the general syntax dedicated to Stata *eclass* commands. It simply adds the prefix *stata*:

```
estimate(stata: e_cmd [,e_cmd_options])
```

Regression commands in Stata are all *eclass*, with regression coefficients stored in the *e(b)* vector. As a consequence, this syntax should be used for computing any regression—for instance, a regression of literacy scores on education levels and including country fixed effects:

```
repest PIAAC, estimate(stata: xi: areg pvlit@ i.edlevels3,
absorb (cny_e))
```

Without any further options, all statistics in the Stata command *e(b)* vector will be computed, using the full sample, potentially conditioned with an *if* statement. The program will return these statistics with their standard errors in the results window.

One important remark: any plausible values variables (native or user-built)—be it in the *if/in* statement, inside the *estimate* argument or inside one of the option arguments—must be included using an @ character in place of the plausible value number. For example, plausible values for literacy scores that appear in the database as *pvlit1*, *pvlit2*, etc. have to be indicated only once as *pvlit@*. The *repest* program recognises the @ character as indicating the presence of a variable with plausible values and will automatically include a loop over the right number of plausible values.

7.2.3 *Repest Options*

Available options are as follows:

- *by*: produces separate estimates by levels of the specified variable (e.g. countries)
- *over*: joint estimates across the different levels of a variable list
- *outfile*: creates a Stata dataset recording all results
- *display*: displays results in output window
- *results*: keep, add, and combine estimation results
- *svypar*: change survey parameters
- *store*: saves the estimation results stored in `e()`
- *fast*: when a plausible value variable is specified, computes sampling variance only for the first plausible value

7.2.3.1 *By Option*

This option allows the desired statistics to be computed separately for each group defined by *varname*. Without any *by* option, *repest* computes the desired statistics once, using the complete dataset. Akin to Stata *bysort*, the *by* option will instruct *repest* to compute the desired statistics for each value of *varname*. In contrast to the *over* option described below, *by* is not intended to split a sample into subsamples of interest, but rather to isolate samples from one another. In practice, *varname* will always be a country indicator. We recommend that the *cntry_e* variable be used to identify countries. The *cntry_e* variable contains ISO3 country codes and remains short, without space characters, and readable.

by accepts two different options. *average(list_of_countries)* will compute the simple average of statistics for countries appearing in its argument. It will also compute the standard error of this average, with the assumption that all samples are independent of each other. *levels(list_of_countries)* will restrain the computation over the countries. By default, *repest* will loop over all countries present in the dataset. Results tables will be displayed for each country, as they are computed, and the results for the average, if requested, will be displayed at the end. The following examples cover different uses of *by*:

```
repest PIAAC, estimate(means pvlit@ pvnum@) by(cntry_e)
```

The above command will compute literacy and numeracy averages in all countries present in the sample, as identified by the *cntry_e* variable.

```
repest PIAAC if ageg10lfs==5, estimate(freq c_d05) by(cntry_e,
average(USA FRA DEU))
```

The above command will compute the labour force status of the population aged between 55 and 65 years for each country in the dataset and then display the simple average of statistics for the United States, France, and Germany.

```
repest PIAAC if litlev@==4, estimate(freq gender_r) by
(cntry_e, levels(USA FRA))
```

This command will compute gender frequencies of the target population scoring at Level 4 in literacy, but only for the United States and France.

7.2.3.2 *Over* Option

Like *by*, *over* splits the sample into different groups in which the desired statistics will be computed. However, these groups are intended to be categories of interest (such as gender, age groups, or education levels) within a country rather than countries. This choice of two different options is justified by the possibility provided in *over* to compute differences of statistics across categories of interest. In contrast to the *by* option, the simulated distribution of the desired statistics is jointly computed for all categories, so that the simulated distribution of a difference can be generated as well. In contrast to *by*, which accepts both string and numeric variables, *over* accepts only numerical variables. *over* includes also a *test* option, which will compute the difference (and its standard error) between the statistics in the groups associated with the two extreme values and the smallest values, along with its standard error. *over* accepts several variables in its argument. In such a case, statistics will be computed in every possible combination of categories.

```
repest PIAAC, estimate(means pvlit@) over(gender, test) by
(cntry_e)
```

The above command will compute in each country average literacy for men and for women, and their difference.

```
repest PIAAC, estimate(freq c_d05) over(gender litlev@) by
(cntry_e)
```

This command will compute labour force status frequencies in each country, for every combination of gender and literacy levels.

7.2.3.3 *Outfile* Option

A large number of statistics can be produced by *repest* (particularly if they are computed by country), and a simple display of the results is not enough to obtain

easy access to and reuse these numbers. For this purpose, *outfile* will export all statistics into a Stata dataset called *filename*, with one observation per country and point estimates and standard errors as variables. The file will be saved in the current Stata working directory. Statistics bear the same names as in the *e(b)* vector, with suffixes *_b* and *_se* identifying point estimates and standard errors. In the presence of an *over* variable, a prefix *_x_* will be added in order to identify statistics for category *x*.

outfile accepts two different options: (1) *pvalue* will add the p-value for every statistic on top of standard errors and point estimates, using the *_pv* suffix. These p-values are computed assuming that the point estimates follow a normal distribution. *long_over* is to be used only if *over* was specified. It will create one observation per country and per combination of *over* variables, in order to have a more readable output. (2) *long_over* is turned on automatically if there is more than one *over* variable. When *outfile* is specified, results are not displayed in the output window.

```
repest PIAAC, estimate(freq c_d05) over(gender) outfile(myfile)
```

7.2.3.4 Display Option

display causes *repest* to display the results in the output window when *outfile* is specified. In this case, results will be displayed both in a *dta* file and in the output window.

```
repest PIAAC, estimate(freq c_d05) display outfile(myfile)
```

7.2.3.5 Results Option

By default, *repest* output consists of the contents of the *e(b)* vector. The *results* option manipulates the vector of desired statistics. It requires one (or several) of the following suboptions:

keep will instruct *repest* to keep only statistics of the *e(b)* vector mentioned in its argument.

add will extend the output to scalar statistics stored in *e()* but not in the *e(b)* vector.

For most *eclass* commands, the help file provides a list of stored results.

combine will take functions of the *e(b)* vector and create new statistics.

Importantly, *results* cannot manipulate the output across subsamples created by the *over* or the *by* options. When using *results*, knowing which exact names the statistics bear can be difficult. The set of examples below will cover potential difficulties.

```
repest PIAAC, estimate(stata: xi: reg pvlit@ ib2.edlevels3
readytolearn ) by(cntry_e) results(add(r2 N))
```

The above command will run the desired regression for each country and add the R-squared coefficient and the number of observations to the output. Note that only the 'r2' keyword is required rather than *e(r2)*. Standard errors will be computed for these two new outputs despite their difficult interpretation.

```
repest PIAAC, estimate(stata: xi: reg readytolearn ib2.
edlevel3 ib2.litlev@ ) by(cntry_e) results(keep(1_litlev@
2b_litlev@ 3_litlev@ 4_litlev@ 5_litlev@))
```

This command will run the desired regression for each country and retain in the output only coefficients associated with the literacy levels. The '@' indicating the presence of a plausible value variable is required. This example shows that the names of statistics to be mentioned in this option might differ from those written in the *outfile* dataset. The starting '_' appearing in the *outfile* dataset for dummy variables must be dropped, while the *b* character indicating the reference must be maintained.

```
repest PIAAC, estimate(summarize pvlit@, stats(p75 p25) ) by
(cntry_e) results(combine( pvlit@_iqr : _b[pvlit@_p75]-
_b[pvlit@_p25] ))
```

While the *summarize* built-in command allows some selected percentiles to be produced, it lacks keywords for interquartile ranges. However they can be computed using the *combine* suboption. Each derivation starts with the name of the new statistics, including the @ character in case of the presence of plausible value, followed by '_iqr', a colon, and a formula definition. The name of each statistic in the formula must be enclosed in *_b[...]*. Additional new statistics can be added after a comma.

7.2.3.6 *Svyvar* Option

The *svyvar* option allows the survey parameters to be manipulated and directly specified. This option is designed to adapt *repest* to currently unfeatured surveys. As such, it is not directly useful to PIAAC users. However, as *svyvar* allows the number of replicated weights used by the program to be manipulated, this option can help to considerably reduce computing time at the expense of incorrect standard errors. This can be useful in case of debugging, and it works as follows:

```
repest PIAAC, estimate(summarize pvlit@) by(cntry_e) svypar  
(NREP(3))
```

7.2.3.7 *Store Option*

The *store* option provides another way of saving results using Stata's estimates store tool. If the option *store* is active, results for each country will be saved using *string* as a prefix and the country identifier as a suffix. Every estimates vector can then be recollected and potentially reused during the rest of the Stata session.

```
repest PIAAC, estimate(freq c_d05) store(freq_c_d05)
```

7.2.3.8 *Fast Option*

The computation of standard errors for statistics that use plausible values variables normally requires an empirical distribution with 810 points. However, the sampling error can be computed using one plausible value instead of all of them without introducing any bias. The *fast* option uses this in order to greatly decrease computation time by almost factor 10. Nonetheless, even though the standard error will be unbiased, it will not be numerically the same.

```
repest PIAAC, estimate(freq c_d05) fast
```

7.2.4 *Conclusion*

This chapter provided an overview of how to use and analyse PIAAC data with Stata. Other statistical software packages, such as SAS, SPSS, or R (see Chap. 9 in this volume), are also well suited for this task, and the user should first consider using the software with which he or she is more familiar. Nonetheless, the availability of the *repest* command is a great asset—all the more so because it is also designed to work with other international skills surveys created by the OECD, such as the Programme for International Student Assessment (PISA) or the Teaching and Learning International Survey (TALIS), or by other institutions (Trends in International Mathematics and Science Study, TIMMS).

References

- Kohler, U., & Kreuter, F. (2012). *Data analysis using Stata*. College Station: Stata Press.
- Organisation for Economic Co-operation and Development (OECD). (2011). *PIAAC conceptual framework of the background questionnaire main survey*. Paris: OECD Publishing.
- Organisation for Economic Co-operation and Development (OECD). (2013). *OECD skills outlook 2013: First results from the Survey of Adult Skills*. Paris: OECD Publishing. <https://doi.org/10.1787/9789264204256-en>.
- Organisation for Economic Co-operation and Development (OECD). (2014). *Technical report of the Survey of Adult Skills (PIAAC)*. Paris: OECD. http://www.oecd.org/skills/piaac/_Technical%20Report_17OCT13.pdf. Accessed 5 May, 2019.
- Organisation for Economic Co-operation and Development (OECD). (2016a). *Survey of Adult Skills technical report (2nd ed.)*. Paris: OECD. http://www.oecd.org/skills/piaac/PIAAC_Technical_Report_2nd_Edition_Full_Report.pdf. Accessed 5 May, 2019.
- Organisation for Economic Co-operation and Development (OECD). (2016b). *Skills matter: Further results from the survey of adult skills*. Paris: OECD Publishing. <https://doi.org/10.1787/9789264258051-en>.
- Organisation for Economic Co-operation and Development (OECD). (2016c). *The survey of adult skills – Reader's companion (2nd. ed.)*. Paris: OECD Publishing. <https://doi.org/10.1787/9789264258075-en>.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.