

Stata and R PIP clients

Accessing the World Bank's estimates of poverty and inequality

The **pip** Stata client (wrapper)

Installation

SSC

```
1 ssc install pip
```

Github

```
1 net install github, from("https://haghish.github.io/github/")  
2 github install worldbank/pip
```

Installation

SSC

```
1 ssc install pip
```

Github

```
1 net install github, from("https://haghish.github.io/github/")  
2 github install worldbank/pip, version(0.9.5.9003)
```

- Official website <https://worldbank.github.io/pip/>
- Report issues at <https://github.com/worldbank/pip/issues>

Country-level estimates

```

1 /* To query poverty at $2.15-a-day poverty line for
2 all countries in all survey years*/
3
4 pip, clear

```

Version in use: 20230328_2017_01_02_PROD

first observation

country_code	year	poverty_line	headcount	mean	median	welfare_type
AGO	2000	2.15	0.2141	7.3744	4.6645291	consumption

[Click here to display how to cite](#)

Country-level estimates

```

1 /* To query poverty at \ $2.15-a-day poverty line
2 for Morocco in 2013 */
3
4 pip, country(mar) year(2013) clear

```

Version in use: 20230328_2017_01_02_PROD

first observation

country_code	year	poverty_line	headcount	mean	median	welfare_type
MAR	2013	2.15	0.0144	10.5652	7.7121489	consumption

[Click here to display how to cite](#)

Country-level estimates

```

1 /* For extrapolated and interpolated data that
2 underpin the global and regional poverty numbers, use
3 `fillgaps` option */
4
5 pip, country(mar) year(2019) clear fillgaps

```

Version in use: 20230328_2017_01_02_PROD

first observation

country_code	year	poverty_line	headcount	mean	welfare_type
MAR	2019	2.15	0.0067	12.4621	consumption

[Click here to display how to cite](#)

Global and regional poverty estimates

```
1 pip wb, clear
```

```
Version in use: 20230328_2017_01_02_PROD
```

```
first observation
```

region_code	year	poverty_line	headcount	mean
WLD	2019	2.15	0.0851	18.0426

```
Click here to display how to cite
```


Global and regional poverty estimates

```

1 pip wb, clear
2
3 /* Query a particular region or global values with `region()` */
4
5 pip wb, clear region(wld)
6 pip wb, clear region(lac)

```

Version in use: 20230328_2017_01_02_PROD

first observation

region_code	year	poverty_line	headcount	mean
LAC	2021	2.15	0.0469	17.3542

[Click here to display how to cite](#)

Poverty lines

```

1 /*Query poverty at other thresholds*/
2
3 pip, country(mar) year(2019) clear ///
4 fillgaps povline(6.85)

```

Version in use: 20230328_2017_01_02_PROD

first observation

country_code	year	poverty_line	headcount	mean	welfare_type
MAR	2019	6.85	0.3144	12.4621	consumption

[Click here to display how to cite](#)

Poverty lines

```

1 /*Or multiple thresholds*/
2
3 pip, country(mar) year(2019) clear ///
4 fillgaps povline(2.15 3.65 6.85 10)

```

Version in use: 20230328_2017_01_02_PROD

first 4 observations

country_code	year	poverty_line	headcount	mean	welfare_type
MAR	2019	10.00	0.5601	12.4621	consumption
MAR	2019	6.85	0.3144	12.4621	consumption
MAR	2019	3.65	0.0578	12.4621	consumption
MAR	2019	2.15	0.0067	12.4621	consumption

[Click here to display how to cite](#)

Data availability

```
1 pip info, clear
```

pip requires the packages below from SSC:

Version in use: 20230328_2017_01_02_PROD

Available Surveys: Select a country or region

Countries

AGO	ALB	ARE	ARG	ARM	AUS	AUT	AZE
BDI	BEL	BEN	BFA	BGD	BGR	BIH	BLR
BLZ	BOL	BRA	BTN	BWA	CAF	CAN	CHE
CHL	CHN	CIV	CMR	COD	COG	COL	COM
CPV	CRI	CYP	CZE	DEU	DJI	DNK	DOM
DZA	ECU	EGY	ESP	EST	ETH	FIN	FJI
FRA	FSM	GAB	GBR	GEO	GHA	GIN	GMB
GNB	GRC	GTM	GUY	HND	HRV	HTI	HUN
IDN	IND	IRL	IRN	IRQ	ISL	ISR	ITA
JAM	JOR	JPN	KAZ	KEN	KGZ	KIR	KOR
LAO	LBN	LBR	LCA	LKA	LSO	LTU	LUX
LVA	MAR	MDA	MDG	MDV	MEX	MHL	MKD
MLI	MLT	MMR	MNE	MNG	MOZ	MRT	MUS
MWI	MYS	NAM	NER	NGA	NIC	NLD	NOR
NPL	NRU	PAK	PAN	PER	PHL	PNG	POL
PRT	PRY	PSE	ROU	RUS	RWA	SDN	SEN
SLB	SLE	SLV	SRB	SSD	STP	SUR	SVK
SVN	SWE	SWZ	SYC	SYR	TCD	TGO	THA
TJK	TKM	TLS	TON	TTO	TUN	TUR	TUV
TWN	TZA	UGA	UKR	URY	USA	UZB	VEN
VNM	VUT	WSM	XKX	YEM	ZAF	ZMB	ZWE

Regions

EAP	ECA	LAC	MNA	NAC	SAR	SSA
-----	-----	-----	-----	-----	-----	-----

World Bank regions by year

Data availability

```
1 pip, country(mar) year(2019) clear
```

```
Warning: years selected for MAR do not match any survey year.  
You could type pip_info, country(MAR) version(20230328_2017_01_02_PROD) clear to check availability.
```

```
the countries and years selected do not match any year available.  
invalid syntax  
r(197); t=0.40 17:34:04
```

```
. pip_info, country(MAR) version(20230328_2017_01_02_PROD) clear
```

Available Surveys for MAR

Detailed information (browser)

Morocco-national

survey year

1984 1990 1998 2000 2006 2013 All

Replicability

```
1 pip version, clear
```

	version
1.	20230328_2017_01_02_PROD
2.	20230328_2011_02_02_PROD
3.	20220909_2017_01_02_PROD
4.	20220909_2011_02_02_PROD

Towards distributional analysis

```
1 pip, country(mar) year(2019) clear fillgaps popshare(0.5)
```

Version in use: 20230328_2017_01_02_PROD

first observation

country_code	year	poverty_line	headcount	mean	welfare_type
MAR	2019	9.10	0.5000	12.4621	consumption

[Click here to display how to cite](#)

Towards distributional analysis

```

1 numlist ".3(.1).8"
2 pip, country(mar) year(2019) clear fillgaps ///
3     popshare(`r(numlist)') n2disp(6)

```

Version in use: 20230328_2017_01_02_PROD

first 6 observations

country_code	year	poverty_line	headcount	mean	welfare_type
MAR	2019	16.01	0.8000	12.4621	consumption
MAR	2019	12.82	0.6999	12.4621	consumption
MAR	2019	10.71	0.5999	12.4621	consumption
MAR	2019	9.10	0.5000	12.4621	consumption
MAR	2019	7.82	0.3999	12.4621	consumption
MAR	2019	6.70	0.3000	12.4621	consumption

[Click here to display how to cite](#)

Towards distributional analysis

```

1 numlist ".3(.1).8"
2 pip, country(mar) year(2019) clear fillgaps ///
3     popshare(`r(numlist)') n2disp(6)

```

Version in use: 20230328_2017_01_02_PROD

first 6 observations

country_code	year	poverty_line	headcount	mean	welfare_type
MAR	2019	16.01	0.8000	12.4621	consumption
MAR	2019	12.82	0.6999	12.4621	consumption
MAR	2019	10.71	0.5999	12.4621	consumption
MAR	2019	9.10	0.5000	12.4621	consumption
MAR	2019	7.82	0.3999	12.4621	consumption
MAR	2019	6.70	0.3000	12.4621	consumption

[Click here to display how to cite](#)

PIP Percentiles data

<https://pip.worldbank.org>

Auxiliary Data

```
1 pip tables, clear
```

```
Auxiliary tables available for 20230328_2017_01_02_PROD:
01 aux_versions
02 countries
03 country_coverage
04 country_list
05 cpi
06 decomposition
07 dictionary
08 framework
09 gdp
10 incgrp_coverage
11 indicators
12 interpolated_means
13 missing_data
14 national_poverty_lines
15 pce
16 pop
17 pop_region
18 poverty_lines
19 ppp
20 region_coverage
21 regions
22 spl
23 survey_means
```

Alternatively

```
1 pip tables, table(cpi) clear
2
3 pip tables, table(interpolated_means) clear
```

<https://pip.worldbank.org>

Help file

Sections

2. General troubleshooting (Go up to Sections menu)

In case **pip** is not working correctly, try the following steps in order

1. Uninstall **pip** by typing **pip uninstall**

Installing the same Stata command from two different sources may result in conflicting

2. Execute **which pip**. If **pip** is still installed, delete all the **pip** files from wherever they are in your computer until the command above returns error. The idea is to leave no trace of **pip** in your computer in the regular way and then you install the stable version from SSC. By doing that, you are creating two entries in the *stata.trk* file, making Stata believe that
3. Install **pip** again with the following code and check the version number. It should be the same as the most recent release confirm this by typing the following,

Selecting countries from the menu loads the survey-year variable

github install worldbank/pip pip development the variable

ssc install github install worldbank/pip

discard

* You can **which pip** tall pip directly pip variable

option **clean** is not included, pip preserves data in memory

interface of survey availability in the results window.

criteria matches more than one package country_code

4. Try to run it again and see if **pip** fails.

* This is because you have two versions of **pip** installed

5. If it's still failing, open a new issue in the GitHub issues page, making sure you're adding all the necessary steps to reproduce the problem.

[318] package pip from <https://raw.githubusercontent.com/worldbank/pip/master>

6. Once the issue is created, run the code below, making sure you replace the commented line--and send the test.log file, along with the issue number created in the previous step,

to pip@worldbank.org <http://fmwww.bc.edu/repec/bocode/p>

PIP : module to access poverty and inequality data from the World Bank's Poverty and Inequality Platform (PIP) <https://pip.worldbank.org>

log using "test.log", name(pip test) text replace // this is in your cd

Examples

4. Analyze data by country/years

4.1 Graph of trend in poverty headcount ratio and number of changes from one year to another (national coverage is

```

pip wb, clear ilable survey_year estimates for Colombia
pip, clear
* Keep if year > 1989 level variable
* Keep if region code == "WB.Dha" 2 "urban" 1 "rural"
egen poorpop = headcount(population/1000000_2) label(lev
egen hcpopround = round(headcount*100, 0.1)
egen poorpopround = round(poorpop, 1)
bycountry, code(welfare_type year: egen _ncover = co
twoway (sc hcpopround year, yaxis(1) mlab(hcpopround)
mlabpos(7) mlabsize(vsmall) c(1))
keep if (sc poorpopround year, yaxis(2) mlab(poorpopround
mlabsize(vsmall) mlabpos(1) c(1)),
* Keep louti("Poverty Rate(%)" try ", size(small) axis(1)
by counylab(0(10)40, labs(small) rtogrid angle(0) axis(1)
by counyti("Number of Poor (million)", size(small) axis(1)
by counylab(0(400)2000, labs(small) angle(0) axis(2))
duplicate graph region(c(white)) ysize(5) xsize(5) // duplica
legend(order(
bysort cl PovertyRate f(% of people living below $2.15)"
bysort cl "Number of people who live below $2.15" si(vsm
replace row(2)) scheme(s2color) == g_type_length) and movir
reference years.
(click to see) same length in series, keep consumption
by country code type_max, sort!(2gen) _ntmax = fill_n == s1
cap of trends in poverty the PIP platform for regions
b. 3. 65 (16.25) : replace _ntmax = s_ntmax[_N] ab//) number

```

Also see * Jump to * (by country/years menu)

4.1 Graph of trend in poverty headcount ratio and number of changes from one year to another (national coverage is

pip wb, clear ilable survey_year estimates for Colombia

pip, clear

* Keep if year > 1989 level variable

* Keep if region code == "WB.Dha" 2 "urban" 1 "rural"

egen poorpop = headcount(population/1000000_2) label(lev

egen hcpopround = round(headcount*100, 0.1)

egen poorpopround = round(poorpop, 1)

bycountry, code(welfare_type year: egen _ncover = co

twoway (sc hcpopround year, yaxis(1) mlab(hcpopround)

mlabpos(7) mlabsize(vsmall) c(1))

keep if (sc poorpopround year, yaxis(2) mlab(poorpopround

mlabsize(vsmall) mlabpos(1) c(1)),

* Keep louti("Poverty Rate(%)" try ", size(small) axis(1)

by counylab(0(10)40, labs(small) rtogrid angle(0) axis(1)

by counyti("Number of Poor (million)", size(small) axis(1)

by counylab(0(400)2000, labs(small) angle(0) axis(2))

duplicate graph region(c(white)) ysize(5) xsize(5) // duplica

legend(order(

bysort cl PovertyRate f(% of people living below \$2.15)"

bysort cl "Number of people who live below \$2.15" si(vsm

replace row(2)) scheme(s2color) == g_type_length) and movir

reference years.

(click to see) same length in series, keep consumption

by country code type_max, sort!(2gen) _ntmax = fill_n == s1

cap of trends in poverty the PIP platform for regions

b. 3. 65 (16.25) : replace _ntmax = s_ntmax[_N] ab//) number

By using `clear all` subcommand, `pip` makes sure all the conflicting installations are solved. `clear all` install `pip` from SSC and from GitHub, one after the other, and you won't have conflicting installations. Be aware that if you have more than one version installed in your `set tracedepth 4` is going to request you to confirm that you want to uninstall both versions `set traceexpand on` on console and hitting enter.

`install` installs the stable version of `pip` from SSC (`pip install` from GitHub (`pip install gh`), the `install` subcommand prevents potentially conflicting installations of the command. Use `set trace on` to install `pip` from SSC and from GitHub, one after the other. There is `/*` for the `pip` command that is failing to find the same search path, `PLUS`. You need `cap noinstall region(EAP) year(last) clear /*` path with option `path()`. Further details are provided on the examples section help. Type `yes` `set trace off` and hit enter to confirm you agree to uninstall `pip`.

`log close pip_test`

`uninstall` Uninstalls any version of `pip` in the installation path. These styles are often rendered as color. In the Results window, on a white background, use `pip install` again because you won't have any version of `pip` installed locally. You will need to install `pip` directly from either SSC (`ssc install pip`) or from GitHub (`github install worldbank/pip`)

`update` This subcommand makes sure the `pip` version is up-to-date. By default, the first time that `pip` is used in a session, it will search for any new versions available from either SSC or GitHub. These styles are often rendered as color. In the Results window, on a white background

Parameters	Description
	These styles are often rendered as color. In the Results window, on a white background,
<code>country</code> (3-letter code)	List of <i>country code</i> (accepts multiples) or <i>all</i> . Default "all". Cannot be used with option <i>region</i> ()
<code>region</code> (WB code)	List of <i>region code</i> (accepts multiple) or <i>all</i> . Default "all". Cannot be used with option <i>country</i> ()
<code>coverage</code> (string)	Coverage level ("national", "urban", "rural", "all"). Default "all".
<code>year</code> (numlist string)	List of years (accepts up to 10), or <i>all</i> , or <i>last</i> . Default "all".
<code>povline</code> (#)	List of poverty lines (in PPP specified, see option <code>ppp_year</code> (#)) to calculate poverty measures (accepts up to 5). Default is 2.15 and 2017 PPPs.
<code>popshare</code> (#)	List of quantiles. No default. Cannot be used with option <code>povline</code>
<code>fillgaps</code>	Loads country-level estimates (including extrapolations and interpolations) used to create regional and global aggregates.

R Client (wrapper)

Installation

```
1 # install.packages("devtools")  
2 devtools::install_github("worldbank/pipr")
```

Webpage

<https://worldbank.github.io/pipr/>

Country-level estimates

```
1 pipr::get_stats() |>
2   select(c(country_code, year, poverty_line, headcount))
```

```
# A tibble: 2,320 × 4
  country_code year poverty_line headcount
  <chr>         <dbl> <dbl>         <dbl>
1 AGO          2000    2.15 0.2140885
2 AGO          2008    2.15 0.1463249
3 AGO          2018    2.15 0.3112201
4 ALB          1996    2.15 0.005348460
5 ALB          2002    2.15 0.01092647
6 ALB          2005    2.15 0.005910857
7 ALB          2008    2.15 0.001999251
8 ALB          2012    2.15 0.006206909
9 ALB          2014    2.15 0.01023101
10 ALB         2015    2.15 0.001206211
# i 2,310 more rows
```

Country-level estimates

```
1 pipr::get_stats(country = "MAR") |>  
2   select(c(country_code, year, poverty_line, headcount))
```

```
# A tibble: 6 × 4  
  country_code year poverty_line headcount  
  <chr>        <dbl>    <dbl>      <dbl>  
1 MAR          1984      2.15 0.1319537  
2 MAR          1990      2.15 0.04728802  
3 MAR          1998      2.15 0.08375334  
4 MAR          2000      2.15 0.07934032  
5 MAR          2006      2.15 0.04242266  
6 MAR          2013      2.15 0.01443110
```

Country-level estimates

```
1 pipr::get_stats(country = c("COL", "MAR")) |>
2   arrange(year) |>
3   select(c(country_code, year, poverty_line, headcount))
```

A tibble: 33 × 4

	country_code	year	poverty_line	headcount
	<chr>	<dbl>	<dbl>	<dbl>
1	COL	1980	2.15	0.2075780
2	MAR	1984	2.15	0.1319537
3	COL	1988	2.15	0.1524133
4	COL	1989	2.15	0.1416732
5	MAR	1990	2.15	0.04728802
6	COL	1991	2.15	0.1341064
7	COL	1992	2.15	0.09557514
8	COL	1996	2.15	0.1757369
9	MAR	1998	2.15	0.08375334
10	COL	1999	2.15	0.2135428

i 23 more rows

Country-level estimates

```
1 pipr::get_stats(country = "COL", year = c(2012, 2018)) |>  
2   select(c(country_code, year, poverty_line, headcount))
```

```
# A tibble: 2 × 4
```

	country_code	year	poverty_line	headcount
	<chr>	<dbl>	<dbl>	<dbl>
1	COL	2012	2.15	0.06715117
2	COL	2018	2.15	0.04498261

Interporlated and Extrapolated values

```
1 pipr::get_stats(country = "HTI") |>  
2   select(c(country_code, year, poverty_line, headcount))
```

```
# A tibble: 3 × 4  
  country_code  year poverty_line headcount  
  <chr>         <dbl>      <dbl>      <dbl>  
1 HTI          2001        2.15  0.6783363  
2 HTI          2012        2.15  0.2919219  
3 HTI          2012        2.15  0.5787614
```

Interporlated and Extrapolated values

```
1 pipr::get_stats(country = "HTI", fill_gaps = TRUE) |>
2   select(c(country_code, year, poverty_line, headcount))
```

```
# A tibble: 41 × 4
  country_code year poverty_line headcount
  <chr>         <dbl>   <dbl>     <dbl>
1 HTI          1981     2.15 0.5127917
2 HTI          1982     2.15 0.5320211
3 HTI          1983     2.15 0.5377601
4 HTI          1984     2.15 0.5445521
5 HTI          1985     2.15 0.5504935
6 HTI          1986     2.15 0.5627245
7 HTI          1987     2.15 0.5717675
8 HTI          1988     2.15 0.7630265
9 HTI          1989     2.15 0.7783534
10 HTI         1990     2.15 0.7747321
# i 31 more rows
```

Poverty line

```
1 pipr::get_stats(country = "COL", povline = 3) |>
2   select(c(country_code, year, poverty_line, headcount))
```

```
# A tibble: 27 × 4
  country_code year poverty_line headcount
  <chr>         <dbl> <dbl>         <dbl>
1 COL          1980     3 0.2994786
2 COL          1988     3 0.2344281
3 COL          1989     3 0.2310108
4 COL          1991     3 0.2139244
5 COL          1992     3 0.1753281
6 COL          1996     3 0.2701443
7 COL          1999     3 0.3089631
8 COL          2000     3 0.2578116
9 COL          2001     3 0.3186980
10 COL         2002     3 0.2312075
# i 17 more rows
```

Multiple Poverty lines

```

1 povlines <- c(2, 4, 6, 10)
2 map_df(.x      = povlines,
3       .f      = pipr::get_stats,
4       country = "COL",
5       year    = 2012) |>
6   select(c(country_code, year, poverty_line, headcount))

```

```

# A tibble: 4 × 4
  country_code year poverty_line headcount
  <chr>         <dbl> <dbl>         <dbl>
1 COL          2012     2 0.05875268
2 COL          2012     4 0.1939661
3 COL          2012     6 0.3436242
4 COL          2012    10 0.5632525

```


Global estimates

```
1 pipr::get_wb() |>
2   arrange(year) |>
3   select(c(region_code, year, poverty_line, headcount))
```

```
# A tibble: 347 × 4
```

	region_code	year	poverty_line	headcount
	<chr>	<dbl>	<dbl>	<dbl>
1	OHI	1981	1.9	0.005183592
2	LAC	1981	1.9	0.1262799
3	SAS	1981	1.9	0.4871674
4	EAP	1981	1.9	0.7870897
5	WLD	1981	1.9	0.3933288
6	OHI	1982	1.9	0.005643049
7	LAC	1982	1.9	0.1316380
8	SAS	1982	1.9	0.4828071
9	EAP	1982	1.9	0.7630886
10	WLD	1982	1.9	0.3875048

```
# i 337 more rows
```

Many more functions

pipr **1.0.0** [Get started](#) [Reference](#) [Articles](#) [Changelog](#)

Reference

All functions

<code>args_to_string()</code>	convert arguments and values of a function to a string to parse into other functions
<code>call_aux()</code>	call a table from .pip env
<code>check_api()</code>	Check internet connection and API status
<code>display_aux()</code>	Display available auxiliary tables
<code>get_aux()</code>	Get auxiliary data
<code>get_countries()</code>	get_countries
<code>get_cpi()</code>	get_cpi()
<code>get_dictionary()</code>	get_dictionary
<code>get_gdp()</code>	get_gdp()
<code>get_hfce()</code>	get_hfce
<code>get_incgrp_coverage()</code>	get_incgrp_coverage
<code>get_interpolated_means()</code>	get_interpolated_means
<code>get_pip_info()</code>	Get PIP info
<code>get_pop()</code>	get_pop
<code>get_pop_region()</code>	get_pop_region
<code>get_ppp()</code>	get_ppp
<code>get_region_coverage()</code>	get_region_coverage
<code>get_regions()</code>	get_regions
<code>get_stats()</code> <code>get_wb()</code>	Get poverty and inequality statistics
<code>get_survey_means()</code>	get_survey_means
<code>get_versions()</code>	Get versions

Code

plot

```

1 # Global poverty trends 1990-2019
2 df <- get_wb() |>
3   filter(year > 1989, region_code == "WLD") |>
4   mutate(
5     pop_in_poverty = round(pop_in_poverty / 1000000, 0),
6     headcount = round(headcount, 3)
7   )
8
9 headcount_col <- "#E69F00"
10
11 gr <- ggplot(df, aes(x = year)) +
12   geom_text(aes(label = headcount * 100,
13               y = headcount),
14             vjust = 1,
215            nudge_y = -,
16            color = headcount_col) +
17   geom_text(aes(label = pop_in_poverty,
18               y = pop_in_poverty / 5000),

```

Code

plot

```

1 # Number of poor by region
2 df <- get_wb() |>
3   filter(year > 1989 & year < 2019) |>
4   mutate(
5     pop_in_poverty = round(pop_in_poverty / 1000000, 0),
6     headcount = round(headcount, 3)
7   )
8
9 regions <- df |>
10  filter(!region_code %in% c("WLD", "AFE", "AFW")) |>
11  mutate(
12    region_name = fct_relevel(region_name,
13                              c("Other high Income",
14                                "Europe and Central Asia",
15                                "Middle East and North Africa",
16                                "Latin America and the Caribbean",
17                                "East Asia and Pacific",
18                                "South Asia",

```

Thanks.

<https://pip.worldbank.org>