PovcalNet API

Beta September 2018
(Updated September 2019)

Qinghua Zhao

The World Bank

Washington DC, USA

Abstract PovcalNet is a product of the World Bank's Global Poverty Monitoring project. It has an interactive computational front end that allows users to replicate the calculations made by the World Bank's researchers in estimating the extent of absolute poverty in the world. PovcalNet also allows users to calculate poverty under different assumptions and to assemble the estimates using alternative country groupings, or for any set of individual countries of the user's choice. The release of the PovcalNet API will bring more flexibility to users. The PovcalNet API can be called directly from a web browser and statistical packages like Stata and R.

This report was prepared by Qinghua Zhao (DECRG) with contributions from Shaohua Chen (DECPI) and colleagues of the Global Poverty Working Group. Ruoxuan Wu provided assistance to the production of the report. Updated by Tony Fujs in September 2019

1. Introduction

PovcalNet is a product of the World Bank's Global Poverty Monitoring project. It is the source of the World Bank's official global, regional and internationally comparable country level poverty estimates published in the World Development Indicators. It has an interactive computational front end that allows users to replicate the calculations made by the World Bank's researchers in estimating the extent of absolute poverty in the world. PovcalNet also allows users to calculate poverty under different assumptions and to assemble the estimates using alternative country groupings or for any set of individual countries of the user's choice.

1.1 History

In the early 90s, a PC executable file for poverty analysis was released by researchers from the World Bank Development Research Group. The name "Povcal" stood for *poverty calculator*. The calculation was based on parametric analysis with grouped data. Two types of Lorenz curves, the general quadratic and beta Lorenz curves, were used to estimate a given distribution. By 2003, a new version of Povcal was under consideration. The original plan of turning it into a Windows application was replaced by having an online version of Povcal. The strength of this framework was clear: since many datasets were not publicly available, users could now have access to country-level distributional data that the World Bank used for Global Poverty Monitoring. With the distributional data and computation tool, the new PovcalNet allowed users to perform poverty analysis just like researchers within the World Bank.

1.2 Major release

The first edition of PovcalNet was released in April 2004. All computations used grouped distributions. The poverty estimates were calculated using the Purchasing Power Parity (PPP) exchange rates for household consumption from the 1993 Penn World Table.

In September 2008, PovcalNet adopted an updated set of PPP exchange rates from the 2005 International Comparison Program.

In March 2013, PovcalNet started to use unit-record data whenever possible.

In October 2014, the World Bank released estimates of global poverty for reference year 2011. PovcalNet encompassed more than 1,000 household surveys across 128 developing countries, and 21 high-income countries.

On January 29, 2015, a new design of PovcalNet was released, with shared prosperity indices added to the site.[1] This release marked a complete redesign of PovcalNet: the computing core was about 10 times faster than the old one, which enabled the public to use the PovcalNet API directly.

[1] The world bank defines shared prosperity as the growth rate of the poorest 40 percent within a country.

Disclaimer

PovcalNet was developed for the sole purpose of public replication of the World Bank's poverty measures for its widely used international poverty lines. The methods built into PovcalNet are considered reliable for that purpose. However, we cannot be confident that the methods work well for other purposes, including tracing out the entire distribution of income. We would especially warn that estimates of the densities near the bottom and top tails of the distribution could be quite unreliable, and no attempt has been made by the Bank's staff to validate the tool for such purposes. PovcalNet team reserves the right to change the syntax of API in the future.

2. PovcalNet API specification

A simple API specification may look like the following:

> http://iResearch.worldbank.org/PovcalNet/PovcalNetAPI.ashx?C0=ALB_3&PL0=1.90&Y0=2002&format=csv

It asks PovcalNet in the server iResearch.worldbank.org to provide poverty estimates for Albania (C0=ALB), using 2002 survey data (Y0=2002), and a poverty line of $1.90 (PL0=1.90). The output is in CSV format:

```
"isInterpolated","useMicroData","CountryCode","CountryName","RegionCode",
"CoverageType","RequestYear","DataYear","DataType","PPP","PovertyLine","M
ean","HeadCount","PovGap","PovGapSqr","Watts","Gini","Median","pr.mld","P
olarization","ReqYearPopulation","SvyInfoID","Decile1","Decile2","Decile3
","Decile4","Decile5","Decile6","Decile7","Decile8","Decile9","Decile10"

0,1,"ALB","Albania","ECA","N","2002","2002","X",58.16801,1.90,192.0995,0
.
002198555,0.0004519817,0.000213323,0.0006470717,0.3173898,158.4549,0.164
8 116,-
1,3.05101,"ALB_N2002X",0.03494,0.04859,0.05842,0.06738,0.07653,0.08839,0.
1023,0.1198,0.1493,0.2544
```

The full API specification consists of

- the protocol: http://,
- the server name: iResearch.worldbank.org,
- the site name: PovcalNet,
- the action handler: PovcalNetAPI.ashx,
- the query string: C0=ALB_3&PL0=1.90&Y0=2002,
- as well as the requested response format: format=csv.

For notational convenience, we will omit all the specification before the question mark and focus on the syntax of the query string.

We need to define a few concepts to describe the syntax of the API calls.

1.Survey year and Reference year

While computing the poverty estimate in the year when survey data are available seems straightforward, aggregating poverty estimates at the regional level could be very tricky because surveys were conducted in different years across countries. The so-called '*Reference year'* was introduced to solve this problem. It referred to three-year intervals starting in 1981. This rule changed in the 2013 release when 2010 was added as a reference year. Then in the 2014 release, 2011 was also included as a reference year. In the September 2018 release, the reference year became biennial. The following is the list of all reference years (as of September 2018)

```
1981,1984,1987,1990,1993,1996,1999,2002,2005,2008,2010,2011,2012,2013,2015
```

All multi-countries aggregations must be performed in the reference year. The syntaxes for survey year and reference year are quite different, even though they share some common notation.

2.Data Coverage

The data coverage describes the area which the survey data is representative of. It takes value

```
1-Rural,
2-Urban,
3-National,
4-Special CPI,
5-Aggregated distribution,
6-Invalid PPP
```

The values reflect different situations. For example, users should adopt code 5 for countries like China, India and Indonesia where a national representative distribution is not used. In this case, the national estimate is computed by weighting the estimates of rural and urban accordingly. On the other hand, only a handful of countries have separate rural and urban distributions. The data coverage code is closely associated with the country code. For this reason, a full specification of country and coverage is ALB_3 in the example above. In most cases, the datasets have national coverage so the coverage code 3 can be omitted, i.e. ALB is the same as ALB_3. Likewise, since coverage code 5 applies only to China, India and Indonesia, CHN is equivalent to CHN_5. The same applies to India and Indonesia.

3.Format

Five possible output formats can be used in any request. They are

```
CSV (default)
Js or JsavaScript
R
Json
Detailed
```

The *detailed* output format applies to single estimates, but all other formats can be applied to any estimates, regardless of the estimation type (single or multiple) or year (survey or reference). When there is no key phrase format, the default format is CSV.

4.PPP

The default option is the PPP rates for consumption in 2011, estimated by the World Bank's Development Data Group.

5.Poverty line

Poverty line is specified in PPP$ per day. In the past, PovcalNet used month as the basic time unit even though the commonly used poverty line was *per day.* This was changed in the April 2015 release which started to use the daily number in the poverty line box. Nowadays, the

mean expenditure (or income) in the output still reports the monthly figure. The conversion of daily to monthly estimate is to multiply the daily figure by 30.4 (=365/12).

2.1 Basic API syntax for survey year estimation

The query string for estimating survey data is very simple:

```
C0=ALB_3&PL0=1.90&Y0=2002&format=csv
```

It requests an estimation using Albania 2002 data with the poverty line set to 1.90 PPP$/day and displays its output in CSV format. The integer suffix of each parameter is a dataset sequence number (must start from 0). For each dataset, the parameters are country(C), year (Y), poverty line (PL) and optional PPP (PPP). The following example shows a query string that estimates Albania 2002 and Zambia 2010.

```
C0=ALB_3&PL0=1.90&Y0=2002&C1=ZMB_3&PL1=1.90&Y1=2010
```

The dataset sequence number should not have any gap. Otherwise only datasets with consecutive numbers will be used. The next query string will only produce two estimates because the sequence number is not consecutive.

```
C0=ALB_3&PL0=1.90&Y0=2002&C1=ZMB_3&PL1=1.90&Y1=2010&C3=BOL_3&PL3=1.90&
Y 3=2011
```

All query strings we have seen so far explicitly specify the national figure. This can be omitted like the following example:

```
C0=ALB&PL0=1.90&Y0=2002&C1=ZMB&PL1=1.90&Y1=2010
```

If the survey year is not present in the query string, all datasets of the corresponding country will be used. The next query string will use all Albania's survey data in years 1996, 2002, 2005, 2008, 2012.

```
C0=ALB&PL0=1.90&Y0=all
```

When country level observations are needed for multiple years, the API allows for another shorthand notation. Users can put in multiple years, separated by a comma.

```
C0=ALB&PL0=1.90&Y0=2012,2002
```

2.2 Alternative syntax for survey year estimation

To improve the usability, the PovcalNet API also allows users to specify the parameters in three different dimensions: PovertyLine, Countries and surveyYears. The following query strings are all valid:

```
PovertyLine=1.9&C0=ALB&Y0=2002&C1=ZMB&Y1=2010

PovertyLine=1.9&Countries=ALB,ZMB&SurveyYears=all

PovertyLine=1.9&Countries=all&SurveyYears=2000,2002

PovertyLine=1.9&Countries=all&SurveyYears=all
```

Please note that `SurveyYears=all` will yield data in different years for different countries, depending on their availability.

2.3 API syntax for reference year estimation

In reference year estimation, a common year will be applied to all selected countries. The parameter to specify the common year selection is *YearSelected=*. Once this parameter appears on the query string, PovcalNet will switch to the mode for reference year computation. Since all estimates for the reference year are done with the same poverty line, the specification of poverty line is defined by the phrase *PovertyLine*=nnn. Similarly, users can use *Countries*=country1, country2 to specify multiple countries in the reference year. The years in the *YearSelected=* clause should only contains reference years. Any non-reference year will be removed. The next query string will calculate the 2010 poverty estimates for all South Asian countries using poverty line PPP$ 1.9.

```
YearSelected=2010&PovertyLine=1.9&Countries=IND,PAK,CHN,IDN,USA
```

Please note that the keyword *YearSelected* is the trigger for reference year calculations. Without it the computation will be done in all specified data years.

Every reference year estimation needs two output tables: the aggregation table and the country table. The aggregation table of above query looks like:

```
 "requestYear","regionTitle","regionCID","povertyLine","mean","hc","pg","p2","population
 " 2010,"XX","XX",1.9,452.4778,0.1005217,0.02315777,0.008427505,2060.137498
```

and the country (or economy) table is:

```
"isInterpolated","useMicroData","CountryCode","CountryName","RegionCode","CoverageT ...
0,1,"IND","India","XX","A","2010","","X",14.97516,1.9,100.5388,0.2785799,0.0610189, ...
1,1,"PAK","Pakistan","XX","N","2010","","X",25.41425,1.9,119.3856,0.09534192,0.0143 ...
0,0,"CHN","China","XX","A","2010","2010","X",3.696112,1.9,218.5439,0.1118215,0.0265 ...
0,1,"IDN","Indonesia","XX","A","2010","2010","X",4091.939,1.9,132.1081,0.1573373,0. ...
0,1,"USA","United States","XX","N","2010","2010","Y",1,1.9,1898.887,0.009971488,0.0 ...
```

The aggregation table gives the aggregation for all countries (or economies) listed in the query. The code XX represents the un-named group total. The country table has the same structure as the survey year estimation but all the estimations are done in the reference year, possibly with interpolation.

Structure of the two tables may cause problem in CSV format when importing the CSV output directly into Excel, Stata or R because these software expect one header row for the whole CSV file.

To avoid the confusion, the additional key phrase *Display=* should be included to identify which part users want. The options for this key phrase are:

```
Regional (or "R")
Country (or "C")
Both (or "B")
```

Here are a few examples of the query strings:

```
YearSelected=2010&PovertyLine=1.9&Countries=IND,PAK,CHN,IDN,USA&display=R

YearSelected=2010&PovertyLine=1.9&Countries=IND,PAK,CHN,IDN,USA&display=C

YearSelected=2010&PovertyLine=1.9&Countries=IND,PAK,CHN,IDN,USA&display=B
```

The first two examples will display regional aggregation table in CSV format. The third query string will show only the country table while the forth will display both regional and country tables.

2.4 Invalid request

If an API call has an invalid parameter, the computation may be terminated before its completion. In this case, users will get nothing or null object. If the query string is in correct format but there is no data available for the country-year combination, then no result will appear in the output.

2.5 Missing value

Missing value may appear when it is not applicable or there is lack of data. It is presented as -1. In the context of PovcalNet, it will not cause confusion.

2.6 Case sensitivity issue

The query string is not case sensitive so the key phrases and value can either be in capital or lower cases.

3. Using API in statistical packages

3.1 Using API in Excel

Excel users can use the command **Data->Get Data->From Web** or **Data->Get Data->From Other Sources->From Web** and then fill in the query URL to load PovcalNet output to Excel worksheet. If the user's Excel does not support this function, simply copy the CSV output from web browser and then paste it into a worksheet using **Data->Text to Columns** command.

3.2 Using API in Stata

Using the insheet command in Stata as follows will load the PovcalNet output (in CSV format) to Stata as a Stata dataset

```
insheet using
"http://iresearch.worldbank.org/povcalnet/povcalnetapi.ashx?C0=ALB&PL0=1
.90&Y0=all", clear
```

The columns in the Stata dataset are converted directly from the CSV file. If user encounter a checksum error, then use

```
set checksum off
```

before the **insheet** command to disable the checksum check.

3.3 Using API in R

Creating a data frame object in R from a web source can be done with

```
df<-read.csv( "the url that returns a csv file")
```

Thus, to load Povcalnet output to a data frame:

```
df<-
read.csv("http://iresearch.worldbank.org/povcalnet/povcalnetapi.ashx
?C0=ALB&PL0=1.90&Y0=all")
```

The data columns in the data frame **df** in the above example will be converted directly from the CSV file.

Appendix: Economy code

The following table shows the economies (regions) available in PovcalNet as of September 1, 2018, sorted by their name.

Code	Name of Economy	WB Region Code	UN Region Code	Income Region Code
AFG	Afghanistan	SAS	ASC	LIC
ALB	Albania	ECA	EUS	UMC
DZA	Algeria	MNA	AFN	UMC
ASM	American Samoa	EAP	OCP	UMC
AND	Andorra	OHI	EUS	HIC
AGO	Angola	SSA	AFM	LMC
ATG	Antigua and Barbuda	OHI	LAB	HIC
ARG	Argentina	LAC	LAS	UMC
ARM	Armenia	ECA	ASW	LMC
ABW	Aruba	OHI	LAB	HIC
AUS	Australia	OHI	OCA	HIC
AUT	Austria	OHI	EUW	HIC
AZE	Azerbaijan	ECA	ASW	UMC
BHS	Bahamas, The	OHI	LAB	HIC
BHR	Bahrain	OHI	ASW	HIC
BGD	Bangladesh	SAS	ASC	LMC
BRB	Barbados	LAC	LAB	HIC
BLR	Belarus	ECA	EUE	UMC
BEL	Belgium	OHI	EUW	HIC
BLZ	Belize	LAC	LAC	UMC
BEN	Benin	SSA	AFW	LIC
BMU	Bermuda	OHI	LAN	HIC
BTN	Bhutan	SAS	ASC	LMC
BOL	Bolivia	LAC	LAS	LMC
BIH	Bosnia and Herzegovina	ECA	EUS	UMC
BWA	Botswana	SSA	AFS	UMC
BRA	Brazil	LAC	LAS	UMC
VGB	British Virgin Islands	OHI	LAB	HIC
BRN	Brunei Darussalam	OHI	ASS	HIC
BGR	Bulgaria	ECA	EUE	UMC
BFA	Burkina Faso	SSA	AFW	LIC
BDI	Burundi	SSA	AFE	LIC
CPV	Cabo Verde	SSA	AFW	LMC
KHM	Cambodia	EAP	ASS	LMC

CMR	Cameroon	SSA	AFM	LMC
CAN	Canada	OHI	LAN	HIC
CYM	Cayman Islands	OHI	LAB	HIC
CAF	Central African Republic	SSA	AFM	LIC
TCD	Chad	SSA	AFM	LIC
CHI	Channel Islands	OHI	EUN	HIC
CHL	Chile	LAC	LAS	HIC
CHN	China	EAP	ASE	UMC
COL	Colombia	LAC	LAS	UMC
COM	Comoros	SSA	AFE	LIC
ZAR	Congo, Democratic Republic of	SSA	AFM	LIC
COG	Congo, Republic of	SSA	AFM	LMC
CRI	Costa Rica	LAC	LAC	UMC
CIV	Cote d'Ivoire	SSA	AFW	LMC
HRV	Croatia	ECA	EUS	UMC
CUB	Cuba	LAC	LAB	UMC
CUW	Curacao	OHI	LAB	HIC
CYP	Cyprus	OHI	ASW	HIC
CZE	Czech Republic	ECA	EUE	HIC
DNK	Denmark	OHI	EUN	HIC
DJI	Djibouti	MNA	AFE	LMC
DMA	Dominica	LAC	LAB	UMC
DOM	Dominican Republic	LAC	LAB	UMC
ECU	Ecuador	LAC	LAS	UMC
EGY	Egypt, Arab Republic of	MNA	AFN	LMC
SLV	El Salvador	LAC	LAC	LMC
GNQ	Equatorial Guinea	SSA	AFM	UMC
ERI	Eritrea	SSA	AFE	LIC
EST	Estonia	ECA	EUN	HIC
SWZ	Eswatini	SSA	AFS	LMC
ETH	Ethiopia	SSA	AFE	LIC
FRO	Faroe Islands	OHI	EUN	HIC
FJI	Fiji	EAP	OCE	UMC
FIN	Finland	OHI	EUN	HIC
FRA	France	OHI	EUW	HIC
PYF	French Polynesia	OHI	OCP	HIC
GAB	Gabon	SSA	AFM	UMC
GMB	Gambia, The	SSA	AFW	LIC
GEO	Georgia	ECA	ASW	LMC
DEU	Germany	OHI	EUW	HIC
GHA	Ghana	SSA	AFW	LMC
GIB	Gibraltar	OHI	EUS	HIC

GRC	Greece	OHI	EUS	HIC
GRL	Greenland	OHI	LAN	HIC
GRD	Grenada	LAC	LAB	UMC
GUM	Guam	OHI	OCM	HIC
GTM	Guatemala	LAC	LAC	LMC
GIN	Guinea	SSA	AFW	LIC
GNB	Guinea-Bissau	SSA	AFW	LIC
GUY	Guyana	LAC	LAS	UMC
HTI	Haiti	LAC	LAB	LIC
HND	Honduras	LAC	LAC	LMC
HKG	Hong Kong SAR, China	OHI	ASE	HIC
HUN	Hungary	ECA	EUE	HIC
ISL	Iceland	OHI	EUN	HIC
IND	India	SAS	ASC	LMC
IDN	Indonesia	EAP	ASS	LMC
IRN	Iran, Islamic Republic of	MNA	ASC	UMC
IRQ	Iraq	MNA	ASW	UMC
IRL	Ireland	OHI	EUN	HIC
IMN	Isle of Man	OHI	EUN	HIC
ISR	Israel	OHI	ASW	HIC
ITA	Italy	OHI	EUS	HIC
JAM	Jamaica	LAC	LAB	UMC
JPN	Japan	OHI	ASE	HIC
JOR	Jordan	MNA	ASW	LMC
KAZ	Kazakhstan	ECA	ASC	UMC
KEN	Kenya	SSA	AFE	LMC
KIR	Kiribati	EAP	OCM	LMC
PRK	Korea, Democratic People's Republic of	EAP	ASE	LIC
KOR	Korea, Republic of	OHI	ASE	HIC
KSV	Kosovo	ECA	EUS	LMC
KWT	Kuwait	OHI	ASW	HIC
KGZ	Kyrgyz Republic	ECA	ASC	LMC
LAO	Lao People's Democratic Republic	EAP	ASS	LMC
LVA	Latvia	ECA	EUN	HIC
LBN	Lebanon	MNA	ASW	UMC
LSO	Lesotho	SSA	AFS	LMC
LBR	Liberia	SSA	AFW	LIC
LBY	Libya	MNA	AFN	UMC
LIE	Liechtenstein	OHI	EUW	HIC
LTU	Lithuania	ECA	EUN	HIC
LUX	Luxembourg	OHI	EUW	HIC
MAC	Macao SAR, China	OHI	ASE	HIC

MKD	Macedonia, former Yugoslav Republic of	ECA	EUS	UMC
MDG	Madagascar	SSA	AFE	LIC
MWI	Malawi	SSA	AFE	LIC
MYS	Malaysia	EAP	ASS	UMC
MDV	Maldives	SAS	ASC	UMC
MLI	Mali	SSA	AFW	LIC
MLT	Malta	OHI	EUS	HIC
MHL	Marshall Islands	EAP	OCM	UMC
MRT	Mauritania	SSA	AFW	LMC
MUS	Mauritius	SSA	AFE	UMC
MYT	Mayotte	SSA	AFE	UMC
MEX	Mexico	LAC	LAC	UMC
FSM	Micronesia, Federated States of	EAP	OCM	LMC
MDA	Moldova	ECA	EUE	LMC
MCO	Monaco	OHI	EUW	HIC
MNG	Mongolia	EAP	ASE	LMC
MNE	Montenegro	ECA	EUN	UMC
MAR	Morocco	MNA	AFN	LMC
MOZ	Mozambique	SSA	AFE	LIC
MMR	Myanmar	EAP	ASS	LMC
NAM	Namibia	SSA	AFS	UMC
NRU	Nauru	OHI	OCM	UMC
NPL	Nepal	SAS	ASC	LIC
NLD	Netherlands	OHI	EUW	HIC
ANT	Netherlands Antilles	OHI	LAB	HIC
NCL	New Caledonia	OHI	OCE	HIC
NZL	New Zealand	OHI	OCA	HIC
NIC	Nicaragua	LAC	LAC	LMC
NER	Niger	SSA	AFW	LIC
NGA	Nigeria	SSA	AFW	LMC
MNP	Northern Mariana Islands	EAP	OCM	HIC
NOR	Norway	OHI	EUN	HIC
OMN	Oman	MNA	ASW	HIC
PAK	Pakistan	SAS	ASC	LMC
PLW	Palau	EAP	OCM	HIC
PAN	Panama	LAC	LAC	UMC
PNG	Papua New Guinea	EAP	OCE	LMC
PRY	Paraguay	LAC	LAS	UMC
PER	Peru	LAC	LAS	UMC
PHL	Philippines	EAP	ASS	LMC
POL	Poland	ECA	EUE	HIC
PRT	Portugal	OHI	EUS	HIC

PRI	Puerto Rico	OHI	LAB	HIC
QAT	Qatar	OHI	ASW	HIC
ROU	Romania	ECA	EUE	UMC
RUS	Russian Federation	ECA	EUE	UMC
RWA	Rwanda	SSA	AFE	LIC
WSM	Samoa	EAP	OCP	UMC
SMR	San Marino	OHI	EUS	HIC
STP	Sao Tome and Principe	SSA	AFM	LMC
SAU	Saudi Arabia	OHI	ASW	HIC
SEN	Senegal	SSA	AFW	LIC
SRB	Serbia	ECA	EUN	UMC
SYC	Seychelles	SSA	AFE	HIC
SLE	Sierra Leone	SSA	AFW	LIC
SGP	Singapore	OHI	ASS	HIC
SXM	Sint Maarten (Dutch part)	OHI	LAS	HIC
SVK	Slovak Republic	ECA	EUE	HIC
SVN	Slovenia	ECA	EUS	HIC
SLB	Solomon Islands	EAP	OCE	LMC
SOM	Somalia	SSA	AFE	LIC
ZAF	South Africa	SSA	AFS	UMC
SSD	South Sudan	SSA	AFE	LIC
ESP	Spain	OHI	EUS	HIC
LKA	Sri Lanka	SAS	ASC	LMC
KNA	St. Kitts and Nevis	LAC	LAB	HIC
LCA	St. Lucia	LAC	LAB	UMC
MAF	St. Martin (French part)	LAC	LAB	HIC
VCT	St. Vincent and the Grenadines	LAC	LAB	UMC
SDN	Sudan	SSA	AFN	LMC
SUR	Suriname	LAC	LAS	UMC
SWE	Sweden	OHI	EUN	HIC
CHE	Switzerland	OHI	EUW	HIC
SYR	Syrian Arab Republic	MNA	ASW	LMC
TWN	Taiwan, China	OHI	ASE	HIC
TJK	Tajikistan	ECA	ASC	LMC
TZA	Tanzania	SSA	AFE	LIC
THA	Thailand	EAP	ASS	UMC
TMP	Timor-Leste	EAP	ASS	LMC
TGO	Togo	SSA	AFW	LIC
TON	Tonga	EAP	OCP	UMC
TTO	Trinidad and Tobago	LAC	LAB	HIC
TUN	Tunisia	MNA	AFN	LMC
TUR	Turkey	ECA	ASW	UMC

TKM	Turkmenistan	ECA	ASC	UMC
TCA	Turks and Caicos Islands	OHI	LAB	HIC
TUV	Tuvalu	EAP	OCP	UMC
UGA	Uganda	SSA	AFE	LIC
UKR	Ukraine	ECA	EUE	LMC
ARE	United Arab Emirates	OHI	ASW	HIC
GBR	United Kingdom	OHI	EUN	HIC
USA	United States	OHI	LAN	HIC
URY	Uruguay	LAC	LAS	HIC
UZB	Uzbekistan	ECA	ASC	LMC
VUT	Vanuatu	EAP	OCE	LMC
VEN	Venezuela, Republica Bolivariana de	LAC	LAS	UMC
VNM	Vietnam	EAP	ASS	LMC
VIR	Virgin Islands, US	OHI	LAB	HIC
WBG	West Bank and Gaza	MNA	ASW	LMC
YEM	Yemen, Republic of	MNA	ASW	LMC
ZMB	Zambia	SSA	AFE	LMC
ZWE	Zimbabwe	SSA	AFE	LIC

The following table provides details on the abbreviation for the grouping categories:

Group	Region Code	Region Title
Income	HIC	High Income
	LIC	Low Income
	LMC	Lower middle Income
	UMC	Upper middle Income
UN	AF	Africa
	AFE	Eastern Africa
	AFM	Middle Africa
	AFN	Northern Africa
	AFS	Southern Africa
	AFW	Western Africa
	AS	Asia
	ASC	South-central Asia
	ASE	Eastern Asia
	ASS	South-eastern Asia
	ASW	Western Asia
	EU	Europe
	EUE	Eastern Europe
	EUN	Northern Europe
	EUS	Southern Europe

	EUW	Western Europe
	LA	Latin America and the Caribbean
	LAB	Caribbean
	LAC	Central America
	LAN	Northern America
	LAS	South America
	OC	Oceania
	OCA	Australia and New Zealand
	OCE	Melanesia
	OCM	Micronesia
	OCP	Polynesia
WB	EAP	East Asia and Pacific
	ECA	Europe and Central Asia
	LAC	Latin America and the Caribbean
	MNA	Middle East and North Africa
	OHI	Other High Income
	SAS	South Asia
	SSA	Sub-Saharan Africa