

# Explore the data with manydata: : CHEAT SHEET

**manydata** provides the portal to access the packages in the many universe of packages and the data contained in those. **manydata** also contains functions to help users to collect, correct and connect data from different datasets and databases across issue-domains of global governance.



## 1) Collecting

`get_packages(manypackage)`

`get_packages()` is a quick and easy way to access and download data from the many packages universe. The function allows users to interactively select which branch (main/develop).



`data_source(manypackage, database, dataset)`

Dataset	Reference
Dataset_A	"Name Surname of authors, year, paper title using the data, publisher, place"
Dataset_B	"Name Surname of authors, year, paper title using the data, publisher, place"
Dataset_C	"Name Surname of authors, year, paper title using the data, publisher, place"

`data_contrast(manypackage, database, dataset)`

Dataset	Unique ID	Missing data	Rows	Columns	Beg	End	URL
Dataset_A	0	3%	3666	8	1351-08-01	NA	<a href="https://sourceidatasetA.com">https://sourceidatasetA.com</a>
Dataset_B	2765	13%	2765	10	1351-08-01	2020-09-12	<a href="https://sourcedatasetB.com">https://sourcedatasetB.com</a>
Dataset_C	2390	19%	2390	9	1868-10-17	9999-12-31	<a href="https://sourcedatasetC.com">https://sourcedatasetC.com</a>

## 2) Correcting

`retrieve_bilaterals()` and `retrieve_multilaterals()` extracts bilateral or multilateral treaties from a dataset of treaties.

manyID	Title	Beg
TFJXKC_1999O	B	1999-02-28
ECE_2003A	M	2003-07-13
AGEJKL_1947O	A	1947-09-19
BALTTT_1966O	T	1966-05-08

`retrieve_bilaterals(dataset)`

manyID	stateID1	stateID2	Title	Beg
TFJXKC_1999O	SIN	BRA	B	1999-02-28
BALTTT_1966O	NZL	MEX	T	1966-05-08

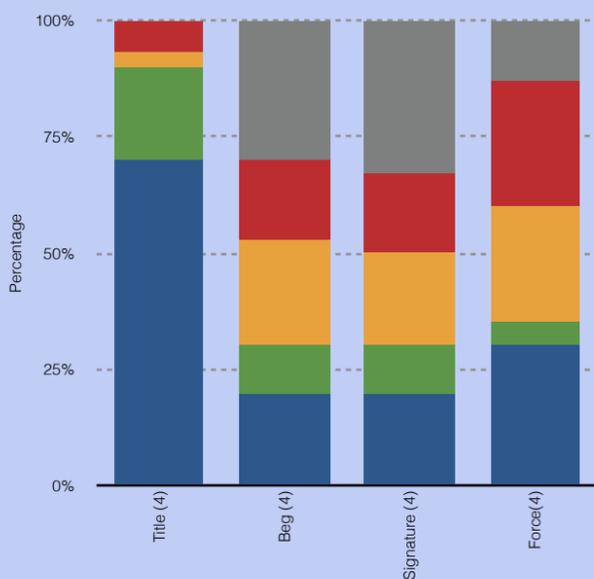
`retrieve_multilaterals(dataset)`

manyID	Title	Beg
ECE_2003A	M	2003-07-13
AGEJKL_1947O	A	1947-09-19

## 3) Connecting

`db_plot(database, key, variable, category)`

`db_plot()` visualises the profile of variables across all datasets in a database.



**'confirmed'**: same across all datasets  
**'majority'**: present in most datasets  
**'unique'**: only in one dataset  
**'conflict'**: different observations in each dataset  
**'missing'**: no observations

Use `db_comp()` to get a tibble that compares the variables across all datasets in the database according to the specified category/categories.

`consolidate()` allows users to produce a dataset from different datasets within the database. It resolves conflicts in data using the 'resolve' argument, which allows users to specify the value they would like to retain in the dataset.

`consolidate(database, rows, cols, resolve, key)`

Database from a package in the many packages universe

For rows and cols, select either any (all units are retained), or every (only those observations appearing in all parent datasets)

Specify value using 'resolve'

manyID	DatasetA\$Beg	DatasetB\$Beg	DatasetC\$Beg	DatasetD\$Beg
ABCEFG_1995O	1995-01-01	1995-03-04	1995-12-07	NA

resolve = min

manyID	Beg
ABCEFG_1995O	1995-01-01

resolve = max

manyID	Beg
ABCEFG_1995O	1995-12-07

resolve = median

manyID	Beg
ABCEFG_1995O	1995-03-04

resolve = coalesce

manyID	Beg
ABCEFG_1995O	1995-01-01

resolve = random

manyID	Beg
ABCEFG_1995O	1995-03-04

resolve = mean

manyID	Beg
ABCEFG_1995O	1995-05-15

if rows and cols = any



if rows and cols = every



if rows = any & cols = every



Use `favour()` to specify the reference dataset for the first NA value before consolidating.