crosswalk

A command for fast and flexible bulk recoding

Ben Jann

University of Bern

2026 German Stata Conference
Munich, June 19, 2026

Outline

Introduction

- Some years ago I wrote command `iscogen` (Jann 2019), a tool to translate ISCO codes (International Standard Classification of Occupations) to scales such as ISEI scores or EGP classes.
- By integrating a comprehensive set of translators in a single package, `iscogen` has quickly become popular among applied researchers.
- However, `iscogen` has limitations.
  - The architecture of `iscogen` makes it tedious to add additional translators.
  - The expected input is restricted to numeric 4-digit ISCO-08, ISCO-88, or ISCO-68 codes.
  - `iscogen` only supports *unique* translations.
  - Documentation is not very user-friendly.

Introduction

- This is why I now wrote `crosswalk`, a general command for table-based bulk recoding.
- `crosswalk` is much more general than `iscogen`.
  - `crosswalk` is not restricted to ISCO codes; any type of input is allowed (including string).
  - New translators are very simple to add; custom translators can be defined on the fly.
  - Translators are self-documenting.
  - Support for non-unique translators is provided.
- If possible, `crosswalk` uses an approach based on indexing, which is very fast. Otherwise, `crosswalk` uses hash tables.
  - In both cases, no data sorting is required (so there is no performance loss in large datasets).
  - The heavy lifting is done by Mata function `mm_crosswalk()` from the `moremata` package (Jann 2005).

Syntax

Main command: translate a variable using *fcn*()

`crosswalk` *newvar* `=` *fcn*`(`*varname* [*case*]`)` [*if*] [*in*] [`,` *options*]

Example: generate ISEI scores from variable `job`, which is assumed to contain ISCO-88 codes

```
crosswalk isei = isco88_to_isei(job)
```

Some options:

- `replace`: allow overwriting an existing variable
- `duplicates(`*method*`)`: how to handle non-unique translations (expand data vs. taking averages, first, last, min, or max)
- `label(`*lblset*`)`: apply custom set of value labels
- `missing`: treat missing values like other values
- `copyrest`: pass unmatched values through
- `copymissing`: pass (extended) missing values through

Translators

fcn() specifies the translator to be used. Various translators related to occupational classifications are provided as part of the `crosswalk` package.

fcn()	Description
Scales for ISCO-08	
isco08_to_isei()	ISCO-08 to ISEI scores
isco08_to_iseisps()	alternative to **isco08_to_isei()**
isco08_to_siops()	ISCO-08 to SIOPS scores; **isco08_to_treiman()** is a synonym
isco08_to_oep()	ISCO-08 to OEP scores
isco08_to_esec()	ISCO-08 to ESEC classes; also see **isco08_3_to_esec()**
isco08_to_oesch()	ISCO-08 to OESCH classes; also see **isco08_to_oesch8()**, **isco08_to_oesch5()**, **oesch_to_oesch8()**, **oesch_to_oesch5()**
Scales for ISCO-88	
isco88_to_isei()	ISCO-88 to ISEI scores
isco88_to_siops()	ISCO-88 to SIOPS scores; **isco88_to_treiman()** is a synonym
isco88_to_oep()	ISCO-88 to OEP scores
isco88_to_mps()	ISCO-88 to MPS scores
isco88_to_esec()	ISCO-88 to ESEC classes; also see

etc . . .

Translators

Clicking on a translator's name will view its source and documentation.

isco88_to_isei() —— Translate 4-digit ISCO-88 to ISEI scores

Syntax

isco88_to_isei(*varname*)

where *varname* contains 4-digit ISCO-88 codes.

Description

crosswalk table translating 4-digit ISCO-88 codes to ISEI scores (International Socio-economic Index of Occupational Status; Ganzeboom et al. 1992).

Source

File **scaleapp.htm** provided by Harry Ganzeboom at http://www.harryganzeboom.nl/isco88/ (equivalent to the Appendix in Ganzeboom/Treiman 1996). Apart from two exceptions, the mapping is consistent with Stata script **iskoisei.ado** (version 1.0 15jun2001) by Hendrickx (2002), which is based on SPSS script **iskoisei.sps** provided by Harry Ganzeboom at http://www.harryganzeboom.nl/isco88/. The two exceptions are ISCO codes 6134 (score 23 in **scaleapp.htm** vs. score 28 in **iskoisei.ado/iskoisei.sps**) and 7520 (score 38 in **scaleapp.htm** vs. score 39 in **iskoisei.ado/iskoisei.sps**); the scores used in **isco88_to_isei()** are consistent with the values printed in Ganzeboom/Treiman (1996).

References

Ganzeboom, H.B.G., P.M. De Graaf, D.J. Treiman. 1992. A Standard International Socio-Economic Index of Occupational Status. Social Science Research 21: 1–56.

Ganzeboom, H.B.G., D.J. Treiman. 1996. Internationally Comparable Measures of Occupational Status for the 1988 International Standard Classification of Occupations. Social Science Research 25: 201–239.

Hendrickx, J. 2002. isko: Stata module to recode 4 digit ISCO-88 occupational codes. Available from http://ideas.repec.org/c/boc/bocode/s425802.html.

```
1000 55
1100 70
1110 77
1120 77
1130 66
```

Making additional translators available

- To make translator *myname*() available, simply store the translator's definition in file `_cwfcn_`*myname*`.sthlp` in the working directory (or somewhere else along the ado-path).
  - Each line of a translator defines a recoding rule consisting of an origin value and one or several destination values.
  - To add documentation, you can include a SMCL header.
  - Example:

— *_cwfcn_myrecode.sthlp* —

```
{smcl}
{helpb crosswalk} translator to flip 1-3 and recode .a to .b
{hline}
{asis}
 1  3
 2  2
 3  1
.a .b
```

— *end of file* —

- At least two third-party add-on packages already exist (`kldbrecode` by Gerhardt/Kappes 2025, `crosswalk_countries` by Kohler 2025).

Defining translators on the fly

- You can also define custom translators on the fly.
- For example, you could type

```
crosswalk define myrecode()
 1  3
 2  2
 3  1
.a .b
end
```

  and then apply the new translator in subsequent code:

```
crosswalk Y = myrecode(X), missing
```

- See `help crosswalk` for several additional commands for managing and manipulating translators.

Meta translators

- A meta translator is defined as a chain of existing translators. In the code of the meta translator, type *.fcn* to apply *fcn*`()`.
- For example, ESeC classes (European Socio-economic Classification) are based on 3-digit ISCO codes. This is why translator `isco08_to_esec()` is defined as follows:

```
crosswalk define isco08_to_esec()
.isco08_to_isco08_3
.isco08_3_to_esec
end
```

  The 4-digit ISCO codes are first translated to 3-digit ISCO codes, then the 3-digit ISCO codes are translated to ESeC classes.
- Limitation: Meta translators cannot be nested, that is, you cannot include a meta translator in the definition of a meta translator.

Multi-column translators and the *case* argument

- Translators can have multiple destination columns; use argument *case* in *fcn*(*varname* *case*) to select the column.
- Argument *case* can be a variable, an expression, or a special case function called `case.`*name*`()`.
- See the next slide for an example of a multi-column translator.
- The syntax appears a bit clumsy, but I did not have a better idea.

isco88_to_oesch() —— Translate 4-digit ISCO-88 to 16 OESCH classes

Syntax

isco88_to_oesch(*varname* [*case*]**)**

where *varname* contains 4-digit ISCO-88 codes and *case* selects the destination column.

Typical usage:

isco88_to_oesch(*varname* **case.oesch(***sempl supvis***))**

with *sempl* and *supvis* as described in **case.oesch()**.

Description

crosswalk table translating 4-digit ISCO-88 codes to 16 OESCH classes (Oesch 2006a,b). Also see **isco88_to_oesch8()** and **isco88_to_oesch5()**.

Cases (destination columns):

1 = employed
2 = self-employed, no employees
3 = self-employed, 1-9 employees
4 = self-employed, 10 or more employees

Case 2 includes helping family members.

Source

Stata script **iskooesch.ado** (version May 2018) by Kaiser (2018). The mapping has been generated automatically by applying **iskooesch.ado** to all relevant combinations of ISCO codes, self-employment status, and number of subordinates or employees.

References

Kaiser, S. 2018. oesch: Stata module to recode ISCO codes into Oesch class scheme. Available from http://ideas.repec.org/c/boc/bocode/s458490.html.
Oesch, D. 2006a. Coming to Grips with a Changing Class Structure. An Analysis of Employment Stratification in Britain, Germany, Sweden and Switzerland. International Sociology 21(2): 263-288
Oesch, D. 2006b. Redrawing the Class Map. Stratification and Institutions in Britain, Germany, Sweden and Switzerland. Palgrave Macmillan.

```
1000 9 4 3 1
1100 9 4 3 1
1110 9 4 3 1
1120 9 4 3 1
1130 9 4 3 1
1140 9 4 3 1
1141 9 4 3 1
```

Value labels

- By default, `crosswalk` looks for value labels and applies them to the generated variable.
  - To be precise, when applying translator *origin*`_to_`*destination*`()`, `crosswalk` will apply label-set *destination* if such a label-set is found.
- For more flexibility there is also command `crosswalk label` that applies value labels to existing variables.
- A label-set is simply a translator whose name starts with `label_`. See the example on the next slide.

help _cwfcn_labels_egp

Dialog Also see Jump to

labels_egp() —— EGP labels (Hendrickx variant)

Description

crosswalk table providing labels for EGP classes (Hendrickx variant).

Source

Stata script **iskoegp.ado** (version 1.2 14Oct2004) by Hendrickx (2002).

References

Hendrickx, J. 2002. isko: Stata module to recode 4 digit ISCO-88 occupational codes. Available from https://ideas.repec.org/c/boc/bocode/s425802.html.

```
1  "higher controllers"
2  "lo controllers"
3  "routine nonmanual"
4  "sempl with emp"
5  "sempl without empl"
7  "manual supervisor"
8  "skilled manual"
9  "semi-unskilld manual"
10 "farm labor"
11 "selfempl farm"
```

Example

- Data from the Swiss Labor Market Survey 1998 (Diekmann et al. 1999).
- Information on occupational characteristics from the O*NET 5.0 database (April 2003); see www.onetcenter.org/db_releases.html.
- Goal: Analyze relations between occupational prestige, job characteristics, and education.

Load main data

```
. use slms98.dta, clear
(Excerpt from the Swiss Labor Market Survey 1998)
. describe
Contains data from slms98.dta
 Observations:         2,130                  Excerpt from the Swiss Labor Market Survey
                                                1998
    Variables:             5                  10 Sep 2025 11:28
```

Variable name	Storage type	Display format	Value label	Variable label
id	float	%9.0g		Respondent ID
female	double	%10.0g		Gender (1=female)
degree	byte	%27.0g	degree	Highest educational degree
isco	int	%8.0g		ISCO-88 code of current job
wt	double	%10.0g		Sampling weight

```
Sorted by: id
. list in 1/5
```

	id	female	degree	isco	wt
1.	10001	0	full-time vocational school	4121	1.0605954
2.	10002	0	university	2321	1.0605954
3.	10003	0	apprenticeship	3400	1.5908932
4.	10006	0	apprenticeship	3430	1.0605954
5.	10007	0	apprenticeship	7212	1.0605954

Add labels to ISCO codes

```
. crosswalk label isco88 isco
(lblset isco88 assigned to isco)
. fre isco, t(5) /* ssc install fre */
isco ── ISCO-88 code of current job
```

			Freq.	Percent	Valid	Cum.
Valid	0	Armed forces	1	0.05	0.05	0.05
	110	Armed forces	1	0.05	0.05	0.09
	1000	Legislators, senior officials and managers	132	6.20	6.20	6.29
	1120	Senior government officials	5	0.23	0.23	6.53
	2113	Chemists	9	0.42	0.42	6.95
	:		:	:	:	:
	9161	Garbage collectors	2	0.09	0.09	99.25
	9300	Labourers in mining, construction, manufacturing and transport	1	0.05	0.05	99.30
	9312	Construction and maintenance labourers: roads, dams and similar constructions	1	0.05	0.05	99.34
	9320	Manufacturing labourers	13	0.61	0.61	99.95
	9999		1	0.05	0.05	100.00
	Total		2130	100.00	100.00	

Generate prestige scores

```
. crosswalk prestige = isco88_to_treiman(isco)
(4 levels of isco not matched)
(variable prestige generated)
. di r(levels_out) /* codes in -isco- that were not translated */
0 110 7139 9999
. list id isco prestige in 1/5, nolabel
```

	id	isco	prestige
1.	10001	4121	45
2.	10002	2321	60
3.	10003	3400	48
4.	10006	3430	49
5.	10007	7212	39

```
. summarize prestige
```

Variable	Obs	Mean	Std. dev.	Min	Max
prestige	2,126	44.31891	11.74677	13	78

Years of education

```
. fre degree
degree ── Highest educational degree
```

			Freq.	Percent	Valid	Cum.
Valid	1	none	49	2.30	2.30	2.30
	2	still in compulsory school	10	0.47	0.47	2.77
	3	compulsory school	144	6.76	6.77	9.54
	4	pre-apprenticeship	38	1.78	1.79	11.33
	5	apprenticeship	990	46.48	46.54	57.87
	6	full-time vocational school	87	4.08	4.09	61.97
	7	vocational baccalaureate	20	0.94	0.94	62.91
	8	diploma middle school	26	1.22	1.22	64.13
	9	high school	124	5.82	5.83	69.96
	10	1-2 year commercial school	54	2.54	2.54	72.50
	11	master craftsman	179	8.40	8.42	80.91
	12	technical college	43	2.02	2.02	82.93
	13	advanced technical college	107	5.02	5.03	87.96
	14	university	209	9.81	9.83	97.79
	15	other	47	2.21	2.21	100.00
	Total		2127	99.86	100.00	
Missing	.y	don't know	2	0.09		
	.z	no answer	1	0.05		
	Total		3	0.14		
Total			2130	100.00		

Years of education

- Transforming educational degrees to years of education could easily be done using the `recode` command. For purpose of illustration I use `crosswalk`.
- Step 1: define the translator

```
crosswalk define educyrs()
    1  5    // none
    2  5    // still in compulsory school
    3  9    // compulsory school
    4  9.75 // pre-apprenticeship
    5  10.5 // apprenticeship
    6  12   // full-time vocational school
    7  12   // vocational baccalaureate
    8  11.5 // diploma middle school
    9  12.5 // high school
    10 10   // 1-2 year commercial school
    11 12   // master craftsman
    12 12.5 // technical college
    13 15   // advanced technical college
    14 17.5 // university
    15 9    // other
end
```

Years of education

- Step 2: apply the translator

```
. crosswalk educyrs = educyrs(degree), copymissing
(variable educyrs generated)
. fre educyrs
educyrs
```

		Freq.	Percent	Valid	Cum.
Valid	5	59	2.77	2.77	2.77
	9	191	8.97	8.98	11.75
	9.75	38	1.78	1.79	13.54
	10	54	2.54	2.54	16.08
	10.5	990	46.48	46.54	62.62
	11.5	26	1.22	1.22	63.85
	12	286	13.43	13.45	77.29
	12.5	167	7.84	7.85	85.14
	15	107	5.02	5.03	90.17
	17.5	209	9.81	9.83	100.00
	Total	2127	99.86	100.00	
Missing	.y don't know	2	0.09		
	.z no answer	1	0.05		
	Total	3	0.14		
Total		2130	100.00		

O*NET job characteristics

- O*NET 5.0 is based on the 2000 SOC (US Standard Occupational Classification). Hence we need to translate ISCO-88 to SOC-2000 and then merge the O*NET data.

```
. crosswalk soc00 = isco88_to_soc00(isco)
crosswalk table isco88_to_soc00() contains duplicate origin values;
origin values must be unique unless option duplicates() is specified
r(498);
```

- That didn't work because `isco88_to_soc00()` is a non-unique translator (single ISCO codes have multiple matches in SOC 2000).
- We could specify option `duplicates(expand)` to keep all matches, merge the O*NET data, and then apply `collapse` by respondent ID to average job characteristics across matches.
- An alternative is to define a SOC-to-characteristic translator and then apply a meta translator with option `duplicates(mean)`.

O*NET job characteristics

- Step 1: define the SOC-to-characteristic translator using `crosswalk post`

```
. preserve
. import delimited onet5.0/WorkActivity.txt, clear
(encoding automatically selected: ISO-8859-1)
(7 vars, 73,964 obs)
. keep if elementname=="Thinking Creatively" & scaleid=="IM"
(73,062 observations deleted)
. keep onetsoccode datavalue
. replace onetsoccode = substr(onetsoccode, 1, 7)
(902 real changes made)
. replace onetsoccode = subinstr(onetsoccode, "-", "", 1)
(902 real changes made)
. crosswalk post soc00_to_creativity()
(crosswalk table soc00_to_creativity() defined)
. restore
```

- Step 2: define the meta translator

```
. crosswalk define isco88_to_creativity()
  1. .isco88_to_soc00
  2. .soc00_to_creativity
  3. end
(crosswalk table isco88_to_creativity() defined)
```

O*NET job characteristics

- Step 3: apply the meta translator with option `duplicates(mean)`

```
. crosswalk creativity = isco88_to_creativity(isco), duplicates(mean)
(56 levels of isco not matched)
(variable creativity generated)
. summarize creativity
```

Variable	Obs	Mean	Std. dev.	Min	Max
creativity	1,599	2.394206	.8334291	1	4.718571

- It seems we lost quite a few observations along the way (`creativity` has 531 missing values). We could look into this and try to find a fix, but I'll ignore the problem for now.

Analysis: compute correlations

- Official `correlate` does not allow sampling weights, so I use command `dstat` (Jann 2020).

```
. /* ssc install dstat */
. dstat pw prestige creativity educyrs [pw=wt], stat(corr) lower pvalues
corr                                            Number of obs     =      1,598
```

	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
prestige						
creativity	.5738194	.0171436	33.47	0.000	.540193	.6074457
educyrs	.5258997	.0232568	22.61	0.000	.4802826	.5715168
creativity						
educyrs	.4070814	.0222446	18.30	0.000	.3634497	.4507131

  - Jobs with higher prestige tend to require more creative thinking.
  - Respondents with higher education tend to work in jobs with higher prestige.
  - Jobs requiring more creative thinking are more likely to be held by respondents with higher education.

Analysis: compute correlations

- Could also use `sem`:

```
. sem (prestige) (creativity) (educyrs) [pw=wt], nolog nodescribe standardized
(532 observations with missing values excluded)
Structural equation model                                   Number of obs = 1,598
Estimation method: ml
Log pseudolikelihood = -11233.33
```

Standardized	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
mean(prestige)	3.69298	.0693281	53.27	0.000	3.5571	3.828861
mean(creativity)	2.882423	.0449269	64.16	0.000	2.794368	2.970478
mean(educyrs)	4.569829	.110038	41.53	0.000	4.354159	4.785499
var(prestige)	1	.			.	.
var(creativity)	1	.			.	.
var(educyrs)	1	.			.	.
cov(prestige, creativity)	.5738194	.0171436	33.47	0.000	.5402185	.6074202
cov(prestige, educyrs)	.5258997	.0232568	22.61	0.000	.4803172	.5714822
cov(creativity, educyrs)	.4070814	.0222446	18.30	0.000	.3634828	.45068

Conclusions

- `crosswalk` is a general, fast, and flexible utility for table-based recoding.
- `crosswalk` supersedes `iscogen` and provides additional functionality such as custom translators or support for non-unique translations.
- To install the command from SSC, type

```
. ssc install crosswalk, replace
. ssc install moremata, replace
```

- Alternatively, install the command from GitHub, see github.com/benjann/crosswalk.
- Check out the following add-on packages:
  - `kldbrecode` (on SSC): translate German occupational classifications (Gerhardt/Kappes 2025)
  - `crosswalk_countries`: translate country codes (Kohler 2025)

References

- Diekmann, A., H. Engelhardt, B. Jann, K. Armingeon, S. Geissbühler. 1999. Der Schweizer Arbeitsmarktsurvey 1998 [Dataset]. DOI: 10.48573/ywf8-hp15
- Gerhardt, H., A. Kappes. 2025. kldbrecode: Stata module that provides crosswalk tables for translations between German KldB and ISCO. Available from https://ideas.repec.org/c/boc/bocode/s459442.html.
- Jann, B. 2005. moremata: Stata module (Mata) to provide various functions. Available from ideas.repec.org/c/boc/bocode/s455001.html.
- Jann, B. 2019. iscogen: Stata module to translate ISCO codes. Available from ideas.repec.org/c/boc/bocode/s458665.html.
- Jann, B. 2020. dstat: Stata module to compute summary statistics and distribution functions including standard errors and optional covariate balancing. Available from ideas.repec.org/c/boc/bocode/s458874.html.
- Kohler, U. 2025. crosswalk-countries. Available from https://gitup.uni-potsdam.de/ukohler/crosswalk-countries.