

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/374556686>

# DIGCLASS: A package to translate between occupational classes in R

Technical Report · September 2023

DOI: 10.13140/RG.2.2.33940.07046

CITATION

1

READS

402

4 authors, including:



**Guillem Vidal-Lorda**  
European Commission

21 PUBLICATIONS 307 CITATIONS

SEE PROFILE



**Carlos J. Gil-Hernández**  
University of Florence

33 PUBLICATIONS 286 CITATIONS

SEE PROFILE



**Oscar Smallembroek**  
European Commission

13 PUBLICATIONS 112 CITATIONS

SEE PROFILE



JRC VALIDATED METHODS, REFERENCE METHODS AND MEASUREMENTS REPORT

# DIGCLASS R package

Cimentada, J., Vidal Lorda, G., Gil Hernandez, C.,  
Smallembroek, O.

2023

This publication is a Validated Methods, Reference Methods and Measurements report by the Joint Research Centre (JRC), the European Commission's science and knowledge service. It aims to provide evidence-based scientific support to the European policymaking process. The contents of this publication do not necessarily reflect the position or opinion of the European Commission. Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use that might be made of this publication. For information on the methodology and quality underlying the data used in this publication for which the source is neither Eurostat nor other Commission services, users should contact the referenced source. The designations employed and the presentation of material on the maps do not imply the expression of any opinion whatsoever on the part of the European Union concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries.

#### Contact information

Name: Guillem Vidal Lorda  
Address: Joint Research Centre, European Commission (Seville, Spain)  
Email: Guillem.Vidal-Lorda@ec.europa.eu  
Tel.: +34 954489005

#### EU Science Hub

<https://joint-research-centre.ec.europa.eu>

JRC133935

Seville: European Commission, 2023

© European Union, 2023



The reuse policy of the European Commission documents is implemented by the Commission Decision 2011/833/EU of 12 December 2011 on the reuse of Commission documents (OJ L 330, 14.12.2011, p. 39). Unless otherwise noted, the reuse of this document is authorised under the Creative Commons Attribution 4.0 International (CC BY 4.0) licence (<https://creativecommons.org/licenses/by/4.0/>). This means that reuse is allowed provided appropriate credit is given and any changes are indicated.

For any use or reproduction of photos or other material that is not owned by the European Union permission must be sought directly from the copyright holders. The European Union does not own the copyright in relation to the following elements:

- Cover page illustration, © krass99 / stock.adobe.com n. [175461355](#)

How to cite this report: Cimentada, J., Vidal Lorda, G., Gil Hernandez, C. and Smallembroek, O., *DIGCLASS R package*, European Commission, Seville, 2023, JRC133935.

# Package ‘DIGCLASS’

August 30, 2023

**Title** A package to translate between occupational classes in R. Currently translate ISCO68/88/08 to dozens of occupational classes

**Version** 0.0.1

**Description** Translating between ISCO class schemas to other schemas is a difficult problem since most translations are scattered around the internet. This packages implements most available translations available for ISCO08/88/68. The package has extra functionalities to check whether ISCO variable are well formatted as well as for converting to major/minor units.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Depends** R (>= 2.10)

**LazyData** true

**Imports** cli,  
dplyr,  
magrittr,  
tibble

**Suggests** glue,  
haven,  
knitr,  
readr,  
rmarkdown,  
testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://digclass.pages.code.europa.eu/digclass/>

**VignetteBuilder** knitr

## R topics documented:

all\_labels . . . . . 2

all_schemas . . . . .	3
esco_to_isco08 . . . . .	3
ess . . . . .	4
isco08_swap . . . . .	5
isco08_to_esec . . . . .	7
isco08_to_esec_mp . . . . .	10
isco08_to_eseg . . . . .	13
isco08_to_ipics . . . . .	15
isco08_to_isco88 . . . . .	17
isco08_to_isei . . . . .	18
isco08_to_microclass . . . . .	19
isco08_to_msec . . . . .	20
isco08_to_oesch . . . . .	22
isco08_to_siops . . . . .	24
isco08_two_to_esec . . . . .	26
isco88com_to_wright . . . . .	27
isco88_to_egp . . . . .	30
isco88_to_egp_mp . . . . .	32
isco88_to_isco08 . . . . .	35
isco88_to_isco68 . . . . .	36
isco88_to_isco88com . . . . .	37
isco88_to_mps . . . . .	38
isco88_to_ordc . . . . .	39
issp . . . . .	40
repair_isco . . . . .	41

**Index** **42**

---

all_labels	<i>A list with all labels available for all occupational schemas</i>
------------	----------------------------------------------------------------------

---

**Description**

Each slot contains a data frame for each of the schemas. The name of each slot in the list shows the current schema.

**Format**

list

---

all_schemas	<i>A list with all available translations between occupational schemas.</i>
-------------	-----------------------------------------------------------------------------

---

### Description

Each slot contains a data frame for each of the translation between two schemas. Additionally, it contains the translation between major/submajor/minor/unit for all ISCO schemas. The name of each slot in the list shows which schemas being translated. Always, the first column is the origin schema and the second column is the destination schema.

### Format

list

---

esco_to_isco08	<i>Translate 4-digit ESCO to ISCO08</i>
----------------	-----------------------------------------

---

### Description

This function translates a vector of 4-digit ESCO codes to ISCO08 codes using the translation table stored in `all_schema$esco_to_isco08`.

### Usage

```
esco_to_isco08(x, label = FALSE, to_factor = FALSE)
```

### Arguments

x	A character vector of 4-digit ESCO codes.
label	A logical value indicating whether to return the labels of the translated ISCO08 codes (default is FALSE).
to_factor	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ISCO08 found in <code>all_labels</code> (default is FALSE).

### Details

This translation converts the ESCO classification of the European Commission to ISCO08. This translations uses ESCO v1.1.1 and was downloaded from the dataset <https://esco.ec.europa.eu/en/use-esco/download>.

For more information on this class schema, please check the references below:

- The ESCO Classification: <https://esco.ec.europa.eu/en/classification>

**Value**

A character vector of ISCO08 codes.

**Examples**

```
esco_to_isco08(c("11401", "1101", "11301"), label = FALSE)
```

```
esco_to_isco08(c("11401", "1101", "11301"), label = TRUE)
```

```
esco_to_isco08(c("11401", "1101", "11301"), label = TRUE, to_factor = TRUE)
```

---

ess

*An example data frame with data from the European Social Survey, round 6.*

---

**Description**

Just an example data frame to show how the package can be used to translate between schemas.

**Format**

Data Frame

**Details**

The codebook for the columns is:

- `isco68`: The ISCO68 class schema in 4-digits.
- `isco88`: The ISCO88 class schema in 4-digits.
- `isco88com`: The ISCO88COM class schema in 4-digits.
- `isco08`: The ISCO08 class schema in 4-digits.
- `emplno`: Number of subordinates, if there are any. If 0 employees, the number is 0 and not an NA, as it should be in your data for using DIGCLASS.
- `self_employed`: A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
- `is_supervisor`: A numeric vector indicating whether each individual is a supervisor (1, e.g. responsible for other employees) or not (0).
- `control_work`: A likert-scale type question from 0 to 10 where 0 is whether an individual has no control over their work/organisation decisions and 10 is complete control over work/organization decisions. For an example, see the variable `iorgact` in the European Social Survey.
- `control_daily`: A likert-scale type question from 1 to 4 where 1 means complete control to decide how their own daily work is/was organised and 4 means no control to decide how their own daily work is/was organised. For an example, see the variable `orgwrk` in the European Social Survey. Another example is recoding the variable `wkdcorga` from the European Social Survey such that 8-10 is 1, 5-7 is 2, 2-4 is 3 and 0-1 is 4.

- `work_status`: This is the same as `self_employed` but has non-employed respondents coded as 2. The unique values are 1 = self\_employed, 0 = employee and 2 = non employed. This variable was constructed using the variable `emplrel` and `mainact` for the unemployed.
- `main_activity`: A numeric vector where 1 = respondent is working, 2 = respondent is in education, 3 = respondent is disabled, 4 = respondent has no paid work (household work, taking care of children, and 5 = respondent is retired. This variable can be recoded using the `mainact` variable from the European Social Survey.

---

isco08\_swap

*Swap ISCO08/ISCO88/ISCO88 between 1, 2, 3 and 4 digit groups*


---

### Description

This function translates a vector of ISCO08/ISCO88/ISCO88 codes between different digits. For most surveys, this will be translating between the 4 digit occupations to more general groups, such as three, two and one digit groups.

### Usage

```
isco08_swap(x, from, to)
```

```
isco88_swap(x, from, to)
```

```
isco68_swap(x, from, to)
```

### Arguments

<code>x</code>	A character vector of 4-digit ISCO08/ISCO88/ISCO88 codes. By 4 digit it means that even though the function could be 3-digits (code 131 for example), the code should be 1310, which is the 4-digit version of ISCO.
<code>from</code>	a numeric specifying the occupation digits of the input vector. Possible values are only 1, 2, 3 or 4.
<code>to</code>	a numeric specifying the desired occupation digits. Possible values are only 1, 2, 3 or 4.

### Details

Note that to translate using `isco*_swap` you'll need to provide the `from` and `to` arguments. The first one specifies the current number of digits of the input variable. If your variable is 4-digit occupations, then `from` should be 4. If you want to translate 4-digit occupations to 3-digits then the arguments should be `from = 4` and `to = 3`. See the argument description of `from` and `to` for all possible values. As well as examples on how this works.

This function will accept 4 digit codes as 4 digits. This means that if the 3-digit code is 131 then it should be 1310. All codes should be 4 digits, even though the code is represented as 3-digits (1310, 1320, etc..)



Note that translation can only be done from higher to smaller digits (4 to 3, 3 to 2, 3 to 1) and never the other way around (1 to 2, 2 to 3, 3 to 4).

ISCO68 might return some NAs depending on the occupation code as it does not have 4 digits for the groups 0000 and 1000. Any translation from 4 digit codes to 1 digit codes within those groups will return an NA for those major groups. See the ILO website: <https://www.ilo.org/public/english/bureau/stat/isco/isco68/major.htm>.

## Value

A character vector of ISCO08/ISCO88/ISCO88 codes.

## Examples

```
library(dplyr)

# isco08
ess %>%
  transmute(
    isco08,
    isco08_one = isco08_swap(isco08, from = 4, to = 1),
    isco08_two = isco08_swap(isco08, from = 4, to = 2),
    isco08_three = isco08_swap(isco08, from = 4, to = 3),
    isco08_four = isco08_swap(isco08, from = 4, to = 4)
  )

# isco88
ess %>%
  transmute(
    isco88,
    isco88_one = isco88_swap(isco88, from = 4, to = 1),
    isco88_two = isco88_swap(isco88, from = 4, to = 2),
    isco88_three = isco88_swap(isco88, from = 4, to = 3),
    isco88_four = isco88_swap(isco88, from = 4, to = 4)
  )

# isco68
# Note that for certain four digit groups, isco68 does not have a
# major group (0000, 1000). That means that Some NAs might be present,
# such as for occupations that are between 1000 and 200. Remember to
# check well the result.
ess %>%
  transmute(
    isco68,
    isco68_one = isco68_swap(isco68, from = 4, to = 1),
    isco68_two = isco68_swap(isco68, from = 4, to = 2),
    isco68_three = isco68_swap(isco68, from = 4, to = 3),
    isco68_four = isco68_swap(isco68, from = 4, to = 4)
  )
```

---

isco08\_to\_esec      *Translate 3-digit ISCO08/ISCO08COM to ESEC*

---

### Description

This function translates a vector of 3-digit ISCO08/ISCO08COM codes to ESEC codes using the translation tables stored in `all_schema$isco08_to_esec / all_schema$isco88com_to_esec_three`.

### Usage

```
isco08_to_esec(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)

isco88com_to_esec(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  full_method = TRUE,
  label = FALSE,
  to_factor = FALSE
)
```

### Arguments

<code>x</code>	A character vector of 3-digit ISCO codes. This should be the 4-digit equivalent so instead of 131, the code should be 1310, which is the 4-digit version of the 3-digit ISCO.
<code>is_supervisor</code>	A numeric vector indicating whether each individual is a supervisor (1, e.g. responsible for other employees) or not (0).
<code>self_employed</code>	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
<code>n_employees</code>	A numeric vector indicating the number of employees under each respondent.
<code>label</code>	A logical value indicating whether to return the labels of the translated ESEC codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ESEC found in <code>all_labels</code> (default is FALSE).
<code>full_method</code>	a boolean on whether to apply the full method or the simple method.

## Details

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for `ISCO08/ISCO88COM -> ESEC`.

This function will accept 3 digit codes as 4 digits. This means that if the 3-digit code is 131 then it should be 1310. All codes should be 4 digits, even though the code is represented as 3-digits (1310, 1320, etc..)

ISCO88COM has two types of translations: simple and full method. The full method uses information on whether the respondent is a supervisor, self-employed and the number of subordinates of the employee. In contrast, the simple method matches directly the ISCO code to an ESEC code.

For more info, please see page 17 of the European Socio-economic Classification (ESeC) User Guide (2006) by Rode, D. and Harrison, E.

The translation for ISCO88 is done from ISCO88COM which is not ISCO88. If you have ISCO88, you can translate it to ISCO88COM using the function `DIGCLASS: : isco88_to_isco88com` before translating to ESEC.

Contrary to ISCO88COM-ESEC, ISCO08 does not have a simplified method and the translation is done from ISCO08 directly to ESEC.

For more information on this class schema, please check the references below:

- Resource website of the European Socio-economic Classification (ESeC): <https://www.iser.essex.ac.uk/archives/esec>
- Derivation material: <https://www.iser.essex.ac.uk/archives/esec/user-guide/derivation-material>
- Rose, D. and Harrison, E. (2007) 'The European Socio-economic Classification: A New Social Class Schema for European Research', *European Societies*, 9, 3: 459-490. <https://doi.org/10.1080/14616690701336518>
- Rose D, Harrison E (2010) *Social Class in Europe. An Introduction to the European Socio-economic Classification*. London: Routledge.
- Wirth, H. (2023). *EU-SILC Tools: European Socioeconomic Classification - ESeC88 and ESeC08*. (GESIS Papers, 2023/01). Köln: GESIS - Leibniz-Institut für Sozialwissenschaften. <https://doi.org/10.21241/ssoar.83962>

## Value

A character vector of ESEC codes.

## Examples

```
library(dplyr)

# convert isco08 to three digits
ess$isco08_three <- isco08_swap(ess$isco08, from = 4, to = 3)

ess %>%
  transmute(
    isco08_three,
    esec = isco08_to_esec(
      isco08_three,
```

```
        is_supervisor,
        self_employed,
        emplno,
        label = FALSE
    ),
    esec_label = isco08_to_esec(
        isco08_three,
        is_supervisor,
        self_employed,
        emplno,
        label = TRUE
    )
)

# convert isco88 to three digits
ess$isco88com_three <- isco88_swap(ess$isco88com, from = 4, to = 3)

# Using the full method
ess %>%
  transmute(
    isco88com_three,
    esec_label = isco88com_to_esec(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      label = TRUE,
      full_method = TRUE
    ),
    esec_no_label = isco88com_to_esec(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      label = FALSE,
      full_method = TRUE
    )
  )

# Using the simple method
ess %>%
  transmute(
    isco88com_three,
    esec_simple = isco88com_to_esec(
      isco88com_three,
      label = FALSE,
      full_method = FALSE
    ),
    esec_simple_label = isco88com_to_esec(
      isco88com_three,
      label = TRUE,
      full_method = FALSE
    )
  )
```

```
)
```

---

```
isco08_to_esec_mp      Translate 3-digit ISCO08/ISCO88COM to ESEC-MP
```

---

### Description

This function translates a vector of 3-digit ISCO08/ISCO88COM codes to ESEC-MP codes using the translation tables stored in `all_schema$isco08_to_esec / all_schema$isco88com_to_esec_three`. After translating to ESEC using these tables, this function reassigns managers and professionals (ISCO08/ISCO88COM codes 1 and 2) to have both high/low managers and professionals

### Usage

```
isco08_to_esec_mp(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)

isco88com_to_esec_mp(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  full_method = TRUE,
  label = FALSE,
  to_factor = FALSE
)
```

### Arguments

<code>x</code>	A character vector of 3-digit ISCO codes. This should be the 4-digit equivalent so instead of 131, the code should be 1310, which is the 4-digit version of of the 3-digit ISCO.
<code>is_supervisor</code>	A numeric vector indicating whether each individual is a supervisor (1, e.g. responsible for other employees) or not (0).
<code>self_employed</code>	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
<code>n_employees</code>	A numeric vector indicating the number of employees under each respondent.
<code>label</code>	A logical value indicating whether to return the labels of the translated ESEC-MP codes (default is FALSE).

to_factor	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the sorted codes of ESEC-MP which can be found in the source code of each function.
full_method	a boolean on whether to apply the full method or the simple method.

### Details

This function translates a vector of 3-digit ISCO08/ISCO88COM codes to ESEC-MP codes.

ESEC-MP is a class schema similar to ESEC but reassigns managers and professionals (ISCO08/ISCO88COM codes 1 and 2) to have both high/low managers and professionals. Similarly to DIGCLASS::isco88com\_to\_esec, isco88com\_to\_esec\_mp allows to translate using the simple or full method. isco08\_to\_esec\_mp does not allow to translate using two different methods and uses the full method by default.

This schema is a slight variation of the original ESEC and the logic used to build this is like this:

- All occupations with ESEC digit 1 and ISCO 1-digit 0 or 1 or has subordinates, **is a high manager**
- All occupations with ESEC digit 1 and is self-employed with more than 1 employee, **is a high manager**
- All occupations with ESEC digit 1 and has a 1-digit ISCO higher than 1 and is either an employee or a self-employed with no subordinates, is a **high professional**
- All occupations with ESEC digit 2 and ISCO 1-digit 0 or 1 or has subordinates, is a **lower manager**
- All occupations with ESEC digit 2 and is self-employed with more than 1 employee, is a **lower manager**
- All occupations with ESEC digit 2 and has a 1-digit ISCO higher than 1 and is either an employee or a self-employed with no subordinates, is a **lower professional**

This translation was created from the Stata do file shared by Oscar Smallenbroek called "ESEC-MP.do". For more info, please contact the author.

This function will accept 3 digit codes as 4 digits. This means that if the 3-digit code is 131 then it should be 1310. All codes should be 4 digits, even though the code is represented as 3-digits (1310, 1320, etc..)

For more information on this class schema, please check the references below:

- Smallenbroek O, Hertel F, Barone C (2022) Measuring class hierarchies in post-industrial societies: a criterion and construct validation of EGP and ESEC across 31 countries. *Sociological Methods & Research*. Epub ahead of print 11 November. <https://doi.org/10.1177/00491241221134522>

### Value

A character vector of ESEC-MP codes.

**Examples**

```

library(dplyr)

# convert to three digits
ess$isco08_three <- isco08_swap(ess$isco08, from = 4, to = 3)

ess %>%
  transmute(
    isco08_three,
    esec = isco08_to_esec(
      isco08_three,
      is_supervisor,
      self_employed,
      emplno,
      label = FALSE
    ),
    esec_label = isco08_to_esec(
      isco08_three,
      is_supervisor,
      self_employed,
      emplno,
      label = TRUE
    )
  )

# Convert isco88com to three digits
ess$isco88com_three <- isco88_swap(ess$isco88com, from = 4, to = 3)

# Using the full method
ess %>%
  transmute(
    isco88com_three,
    esec = isco88com_to_esec_mp(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      full_method = TRUE,
      label = FALSE
    ),
    esec_label = isco88com_to_esec_mp(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      full_method = TRUE,
      label = TRUE
    )
  )

# Using the simple method. For esec_mp
# we need all variables (is_supervisor, self_employed, etc..)

```

```

# because we need to assign the manager/professionals depending
# these variables.
ess %>%
  transmute(
    isco88com_three,
    esec_simple = isco88com_to_esec_mp(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      full_method = FALSE,
      label = FALSE
    ),
    esec_simple_label = isco88com_to_esec_mp(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      full_method = FALSE,
      label = TRUE
    )
  )
)

```

---

isco08\_to\_eseg

*Translates 2-digit ISCO08 to ESEG.*


---

### Description

This function translates a vector of 2-digit ISCO88COM codes to the ESEG class schema.

### Usage

```

isco08_to_eseg(
  x,
  work_status,
  main_activity,
  age,
  type,
  label = FALSE,
  to_factor = FALSE
)

```

### Arguments

x	A character vector of 2-digit ISCO codes. This should be the 4-digit equivalent so instead of 13, the code should be 1300, which is the 4-digit version of the 2-digit ISCO.
work_status	A numeric vector of values from 0 to 3 where 1 = self_employed, 0 = employee and 2 = non employed.



main_activity	A numeric vector of values from 1 to 5 where 1 = respondent is working, 2 = respondent is in education, 3 = respondent is disabled, 4 = respondent has no paid work (household work, taking care of children, etc..) and 5 = respondent is retired. For an example, see the variable mainact from the European Social Survey.
age	A numeric vector of ages of the respondent.
type	The type of translation to make. Possible values are "one-digit" and "two-digit". The "one-digit" translation returns a broad summary based translation of only 9 categories, whereas the "two-digit" translation returns a much bigger ESEG translation of more than 25 categories.
label	A logical value indicating whether to return the labels of the translated ESEG codes (default is FALSE).
to_factor	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the sorted codes of ESEG which can be found in the source code of each function.

### Details

The translation was implemented following the work of Kea Tijdens in the document "ESEG-2014 coding scheme + explanatory note". For more details, see the last table in the document [here](#).

For more details on this class schema, please check the references below:

- Börlin, S., & Zettl, L. (2020). Mikrozensus Tools: Die Klassifikation European Socio-economic Groups (ESeG) im Mikrozensus Scientific Use File. (GESIS Papers, 2020/08). Köln: GESIS - Leibniz-Institut für Sozialwissenschaften. <https://doi.org/10.21241/ssoar.68449>
- Bohr, J. (2018). EU-AES Tools: Implementation of the European Socioeconomic Groups Classification (ESeG) using Adult Education Survey Microdata. (GESIS Papers, 2018/14). Köln: GESIS - Leibniz-Institut für Sozialwissenschaften. <https://doi.org/10.21241/ssoar.57622>
- Meron M, and all ESSnet members (2014) ESSnet ESeG Final Report. Paris, INSEE, Direction des Statistiques Démographiques et Sociales ESSnet project
- Meron, M. et al. (2014): Final Report of the ESSnet on the harmonisation and implementation of a European socio-economic classification: European Socio-economic Groups (ESeG)
- Tijdens, K.G. (2016) ESEG-2014 coding scheme + explanatory note. Deliverable 8.13 of the SERISS project funded under the European Union's Horizon 2020 research and innovation programme GA No: 654221. Available at: <https://seriss.eu/resources/deliverables>

Resource websites of the European Socio-economic Groups (ESeG):

- [https://cros-legacy.ec.europa.eu/content/eseg\\_en](https://cros-legacy.ec.europa.eu/content/eseg_en)
- [https://ec.europa.eu/eurostat/cros/content/eseg-report-technicaldocuments\\_en](https://ec.europa.eu/eurostat/cros/content/eseg-report-technicaldocuments_en)
- [https://ec.europa.eu/eurostat/cros/system/files/ESEG-Report-TechnicalAnnexes\\_0.zip\\_en](https://ec.europa.eu/eurostat/cros/system/files/ESEG-Report-TechnicalAnnexes_0.zip_en)

**Examples**

```
library(dplyr)

# Convert isco08 to two digits
ess$isco08_two <- isco08_swap(ess$isco08, from = 4, to = 2)

# Using the two-digit translation
ess %>%
  transmute(
    isco08_two,
    eseg = isco08_to_eseg(
      isco08_two,
      work_status,
      main_activity,
      agea,
      type = "two-digit"
    ),
    eseg_label = isco08_to_eseg(
      isco08_two,
      work_status,
      main_activity,
      agea,
      type = "two-digit",
      label = TRUE
    )
  )

# Using the one-digit translation
ess %>%
  transmute(
    isco08_two,
    eseg = isco08_to_eseg(
      isco08_two,
      work_status,
      main_activity,
      agea,
      type = "one-digit"
    ),
    eseg_label = isco08_to_eseg(
      isco08_two,
      work_status,
      main_activity,
      agea,
      type = "one-digit",
      label = TRUE
    )
  )
```

**Description**

This function translates a vector of 4-digit ISCO08/ISCO88 codes to IPICS codes using the translation tables stored in `all_schema$isco08_to_ipics` / `all_schema$isco88_to_ipics`.

**Usage**

```
isco08_to_ipics(
  x,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)

isco88_to_ipics(
  x,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)
```

**Arguments**

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>self_employed</code>	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
<code>n_employees</code>	A numeric vector indicating the number of employees under each respondent.
<code>label</code>	A logical value indicating whether to return the labels of the translated IPICS codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for IPICS found in <code>all_labels</code> (default is FALSE).

**Details**

These translation were created from the CSV files shared by Oscar Smallembroek name "ISCO08 to IPICS.csv" and "ISCO88 to IPICS.csv". For more info, please contact the author.

For more details on the class schema please check the references below:

- Hertel, Florian R. 2017. *Social Mobility in the 20th Century: Class Mobility and Occupational Change in the United States and Germany*. Springer VS.

**Value**

A character vector of IPICS codes.

**Examples**

```

library(dplyr)

# isco08
ess %>% transmute(
  isco08,
  ipics = isco08_to_ipics(isco08, self_employed, emplno),
  ipics_label = isco08_to_ipics(isco08, self_employed, emplno, label = TRUE)
)

# isco88
ess %>% transmute(
  isco88,
  ipics = isco88_to_ipics(isco88, self_employed, emplno),
  ipics_label = isco88_to_ipics(isco88, self_employed, emplno, label = TRUE)
)

```

---

isco08_to_isco88	<i>Translate 4-digit ISCO08/ISCO68 to ISCO88</i>
------------------	--------------------------------------------------

---

**Description**

This function translates a vector of 4-digit ISCO08/ISCO68 codes to ISCO88 codes using the translation tables stored in `all_schema$isco08_to_isco88` / `all_schema$isco68_to_isco88`.

**Usage**

```

isco08_to_isco88(x, label = FALSE, to_factor = FALSE)

isco68_to_isco88(x, label = FALSE, to_factor = FALSE)

```

**Arguments**

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>label</code>	A logical value indicating whether to return the labels of the translated ISCO88 codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ISCO88 found in <code>all_labels</code> (default is FALSE).

**Details**

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for ISCO08/ISCO68 -> ISCO88.

For more information on this class schema, please check the references below:

- International Standard Classification of Occupations: ISCO-08 / International Labour Office, - Geneva: ILO, 2012. <https://www.ilo.org/public/english/bureau/stat/isco/index.htm>

### Value

A character vector of ISCO88 codes.

### Examples

```
library(dplyr)

# isco08
ess %>% transmute(
  isco08,
  isco88 = isco08_to_isco88(isco08, label = FALSE),
  isco88_label = isco08_to_isco88(isco08, label = TRUE)
)

# isco68
ess %>% transmute(
  isco68,
  isco88 = isco68_to_isco88(isco68, label = FALSE),
  isco88_label = isco68_to_isco88(isco68, label = TRUE)
)
```

---

<code>isco08_to_isei</code>	<i>Translate 4-digit ISCO08/ISCO88/ISCO68 to ISEI</i>
-----------------------------	-------------------------------------------------------

---

### Description

This function translates a vector of 4-digit ISCO08/ISCO88/ISCO68 codes to ISEI codes using the translation tables stored in `all_schema$isco08_to_isei` / `all_schema$isco88_to_isei` / `all_schema$isco68_to_isei`.

### Usage

```
isco08_to_isei(x, to_factor = FALSE)
```

```
isco88_to_isei(x, to_factor = FALSE)
```

```
isco68_to_isei(x, to_factor = FALSE)
```

### Arguments

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the sorted codes of ISEI in <code>all_schemas\$isco08_to_isei</code> , <code>all_schemas\$isco88_to_isei</code> and <code>all_schemas\$isco68_to_isei</code> .

## Details

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for ISCO08/ISCO88/ISCO88 -> ISEI.

Since ISEI doesn't have any labels, the `labels` is not available in this function.

For more information on this class schema, please check the references below:

- Ganzeboom, H.B.G. (2010) International Standard Classification of Occupations. ISCO-08. With ISEI-08 scores. Last revised: July 27 2010. Available from <http://www.harryganzeboom.nl/isco08/>.
- 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.

Resource websites of the International Socio-economic Index (ISEI):

- <http://www.harryganzeboom.nl/isco08/qa-isei-08.htm>
- <http://www.harryganzeboom.nl/isco88/>

## Value

A character vector of ISEI codes.

## Examples

```
library(dplyr)

ess %>%
  transmute(
    isco08,
    isco88,
    isco68,
    isei_08 = isco08_to_isei(isco08),
    isei_88 = isco88_to_isei(isco88),
    isei_68 = isco68_to_isei(isco68)
  )
```

---

`isco08_to_microclass` *Translate 4-digit ISCO08 to microclass*

---

## Description

This function translates a vector of 4-digit ISCO08 codes to microclass codes using the translation table stored in `all_schema$isco08_to_microclass`.

## Usage

```
isco08_to_microclass(x, label = FALSE, to_factor = FALSE)
```

**Arguments**

x	A character vector of 4-digit ISCO codes.
label	A logical value indicating whether to return the labels of the translated microclass codes (default is FALSE).
to_factor	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for microclass found in <code>all_labels</code> (default is FALSE).

**Details**

This translation was created from the Excel file shared by Oscar Smallembroek called "isco08 to micro with numeric labels.xlsx". For more info, please contact the author.

For more details on this class schema, please check the references below:

- Weeden, Kim A., and David B. Grusky. 2005. "The Case for a New Class Map." *American Journal of Sociology* 111(1):141-212.
- —. 2012. "The Three Worlds of Inequality." *American Journal of Sociology* 117(6):1723-85.

**Value**

A character vector of microclass codes.

**Examples**

```
library(dplyr)

ess %>% transmute(
  isco08,
  microclasses = isco08_to_microclass(isco08),
  microclasses_label = isco08_to_microclass(isco08, label = TRUE)
)
```

---

isco08\_to\_msec

*Translate 3-digit ISCO08/ISCO88COM to MSEC*


---

**Description**

This function translates a vector of 3-digit ISCO08/ISCO88COM codes to MSEC codes using the translation tables stored in `all_schema$isco08_to_msec` / `all_schema$isco88com_to_msec`.

**Usage**

```

isco08_to_msec(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)

isco88com_to_msec(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)

```

**Arguments**

x	A character vector of 3-digit ISCO codes. This should be the 4-digit equivalent so instead of 131, the code should be 1310, which is the 4-digit version of the 3-digit ISCO.
is_supervisor	A numeric vector indicating whether each individual is a supervisor (1, e.g. responsible for other employees) or not (0).
self_employed	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
n_employees	A numeric vector indicating the number of employees under each respondent.
label	A logical value indicating whether to return the labels of the translated MSEC codes (default is FALSE).
to_factor	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for MSEC found in <code>all_labels</code> (default is FALSE).

**Details**

These translations were created from the document "Allocation rules of ISCO-08 and ISCO-88 (COM) 3-digit codes to ESEG-Revised" from Oscar Smallenbroek, Florian Hertel and Carlo Barone. For more info, please contact the authors. Although originally called ESEG-Revised, the class schema has been formally called MSEC.

This function will accept 3 digit codes as 4 digits. This means that if the 3-digit code is 131 then it should be 1310. All codes should be 4 digits, even though the code is represented as 3-digits (1310, 1320, etc..)

**Value**

A character vector of MSEC codes.



**Examples**

```

library(dplyr)

# convert to three digits
ess$isco08_three <- isco08_swap(ess$isco08, from = 4, to = 3)
ess$isco88com_three <- isco88_swap(ess$isco88com, from = 4, to = 3)

# isco08
ess %>%
  transmute(
    isco08_three,
    msec_label = isco08_to_msec(
      isco08_three,
      is_supervisor,
      self_employed,
      emplno,
      label = TRUE
    ),
    msec = isco08_to_msec(
      isco08_three,
      is_supervisor,
      self_employed,
      emplno,
      label = FALSE
    )
  )

# isco88com
ess %>%
  transmute(
    isco88com_three,
    msec_label = isco88com_to_msec(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      label = TRUE
    ),
    msec = isco88com_to_msec(
      isco88com_three,
      is_supervisor,
      self_employed,
      emplno,
      label = FALSE
    )
  )

```

## Description

This function translates a vector of 4-digit ISCO08/ISCO88 codes to OESCH codes using the translation tables stored in `all_schema$isco08_to_oesch16` / `all_schema$isco88_to_oesch16` / `all_schema$oesch16_to_oesch5` / `all_schema$oesch16_to_oesch5`.

## Usage

```
isco08_to_oesch(
  x,
  self_employed,
  n_employees,
  n_classes = 16,
  label = FALSE,
  to_factor = FALSE
)

isco88_to_oesch(
  x,
  self_employed,
  n_employees,
  n_classes = 16,
  label = FALSE,
  to_factor = FALSE
)
```

## Arguments

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>self_employed</code>	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
<code>n_employees</code>	A numeric vector indicating the number of employees under each respondent.
<code>n_classes</code>	a numeric value indicating the number of OESCH classes to obtain. Default is 16 OESCH classes. The possible values are 16 classes, 8 classes and 5 classes. For more information, see the details section.
<code>label</code>	A logical value indicating whether to return the labels of the translated OESCH codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for OESCH found in <code>all_labels</code> (default is FALSE).

## Details

This function works by first translating to OESCH16 and then translating to other OESCH variants, if the user has requested this through the `n_classes` argument.

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for ISCO88/ISCO88 -> OESCH. For translations between OESCH16 and OESCH8/OESCH5, see the source of the Stata package [oesch](#) [here](#).

Users can see the translation used in this package in `all_schema$oesch16_to_oesch8` and `all_schema$oesch16_to_oesch5`. Moreover, the labels used can be found in `all_labels$oesch16`, `all_labels$oesch8` and `all_labels$oesch5`.

For more details on this class schema, please check the references below:

- 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.

Resource websites of the OESCH Social Class Schema:

- <https://people.unil.ch/danieloesch/scripts/>

### Value

A character vector of OESCH codes.

### Examples

```
library(dplyr)

# isco08
ess %>%
  transmute(
    isco08,
    oesch16 = isco08_to_oesch(isco08, self_employed, emplno, label = FALSE),
    oesch8 = isco08_to_oesch(isco08, self_employed, emplno, n_classes = 8, label = FALSE),
    oesch5 = isco08_to_oesch(isco08, self_employed, emplno, n_classes = 5, label = FALSE),
  )

# isco88
ess %>%
  transmute(
    isco88,
    oesch16 = isco88_to_oesch(isco88, self_employed, emplno, label = FALSE),
    oesch8 = isco88_to_oesch(isco88, self_employed, emplno, n_classes = 8, label = FALSE),
    oesch5 = isco88_to_oesch(isco88, self_employed, emplno, n_classes = 5, label = FALSE),
  )
```

---

isco08\_to\_siops

*Translate 4-digit ISCO08/ISCO88/ISCO68 to SIOPS*

---

### Description

This function translates a vector of 4-digit ISCO08/ISCO88/ISCO68 codes to SIOPS codes using the translation tables stored in `all_schema$isco08_to_siops` / `all_schema$isco88_to_siops` / `all_schema$isco68_to_siops`.

## Usage

```
isco08_to_siops(x, to_factor = FALSE)
```

```
isco88_to_siops(x, to_factor = FALSE)
```

```
isco68_to_siops(x, to_factor = FALSE)
```

## Arguments

x	A character vector of 4-digit ISCO codes.
to_factor	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the sorted codes of SIOPS in <code>all_schemas\$isco08_to_siops</code> , <code>all_schemas\$isco88_to_siops</code> and <code>all_schemas\$isco68_to_siops</code> .

## Details

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for `ISCO08/ISCO88/ISCO88 -> SIOPS`.

Since SIOPS doesn't have any labels, the `labels` is not available in this function.

For more information on this class schema, please check the references below:

- 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.

## Value

A character vector of SIOPS codes.

## Examples

```
library(dplyr)

ess %>%
  transmute(
    isco08,
    isco88,
    isco68,
    siops_08 = isco08_to_siops(isco08),
    siops_88 = isco88_to_siops(isco88),
    siops_68 = isco68_to_siops(isco68)
  )
```

---

isco08\_two\_to\_esec      *Translate 2-digit ISCO08 to ESEC*

---

### Description

This function translates a vector of 2-digit ISCO08 codes to ESEC codes using the translation table stored in `all_schema$isco08_two_to_esec`.

### Usage

```
isco08_two_to_esec(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)
```

### Arguments

<code>x</code>	A character vector of 2-digit ISCO codes. This should be the 4-digit equivalent so instead of 13, the code should be 1300, which is the 4-digit version of the 2-digit ISCO.
<code>is_supervisor</code>	A numeric vector indicating whether each individual is a supervisor (1, e.g. responsible for other employees) or not (0).
<code>self_employed</code>	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
<code>n_employees</code>	A numeric vector indicating the number of employees under each respondent.
<code>label</code>	A logical value indicating whether to return the labels of the translated ESEC codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ESEC found in <code>all_labels</code> (default is FALSE).

### Details

This function will accept 2 digit codes as 4 digits. This means that if the 2-digit code is 13 then it should be 1300. All codes should be 4 digits, even though the code is represented as 2-digits (1300, 1400, etc..)

This is exactly the same as `DIGCLASS::isco08_to_esec` but for two digit ISCO.

For more information on this class schema, please check the references below:

- Resource website of the European Socio-economic Classification (ESeC): <https://www.iser.essex.ac.uk/archives/esec>

- Derivation material: <https://www.iser.essex.ac.uk/archives/esecc/user-guide/derivation-material>
- Rose, D. and Harrison, E. (2007) 'The European Socio-economic Classification: A New Social Class Schema for European Research', *European Societies*, 9, 3: 459-490. <https://doi.org/10.1080/14616690701336518>
- Rose D, Harrison E (2010) *Social Class in Europe. An Introduction to the European Socio-economic Classification*. London: Routledge.
- Wirth, H. (2023). *EU-SILC Tools: European Socioeconomic Classification - ESeC88 and ESeC08*. (GESIS Papers, 2023/01). Köln: GESIS - Leibniz-Institut für Sozialwissenschaften. <https://doi.org/10.21241/ssoar.83962>

## Examples

```
library(dplyr)

# convert to two digits
ess$isco08_two <- isco08_swap(ess$isco08, from = 4, to = 2)

ess %>%
  transmute(
    isco08_two,
    esec = isco08_two_to_esec(
      isco08_two,
      is_supervisor,
      self_employed,
      emplno,
      label = FALSE
    ),
    esec_label = isco08_two_to_esec(
      isco08_two,
      is_supervisor,
      self_employed,
      emplno,
      label = TRUE
    )
  )
```

---

isco88com\_to\_wright     *Translates 4-digit ISCO88COM to WRIGHT.*

---

## Description

This function translates a vector of 4-digit ISCO88COM codes to the E.O Wright class schema.

## Usage

```
isco88com_to_wright(
  x,
```

```

    is_supervisor,
    self_employed,
    n_employees,
    control_work,
    control_daily,
    type,
    label = FALSE,
    to_factor = FALSE
  )

```

### Arguments

x	A character vector of 4-digit ISCO codes. This should be the 4-digit equivalent so instead of 131, the code should be 1310, which is the 4-digit version of the 4-digit ISCO.
is_supervisor	A numeric vector indicating whether each individual is a supervisor (1, e.g. responsible for other employees) or not (0).
self_employed	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
n_employees	A numeric vector indicating the number of employees under each respondent.
control_work	A likert-scale type question from 0 to 10 where 0 is whether an individual has no control over their work/organisation decisions and 10 is complete control over work/organization decisions. For an example, see the variable <code>iorgact</code> in the European Social Survey.
control_daily	A likert-scale type question from 1 to 4 where 1 means complete control to decide how their own daily work is/was organised and 4 means no control to decide how their own daily work is/was organised. For an example, see the variable <code>orgwrk</code> in the European Social Survey. Another example is recoding the variable <code>wkdcorga</code> from the European Social Survey such that 8-10 is 1, 5-7 is 2, 2-4 is 3 and 0-1 is 4.
type	The type of translation to make. Possible values are "simple", "decision-making" and "power-class".
label	A logical value indicating whether to return the labels of the translated WRIGHT codes (default is FALSE).
to_factor	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the sorted codes of WRIGHT which can be found in the source code of each function.

### Details

The translation implemented in this function was originally developed by Erik Olin Wright. There are three possible types of translations: the "simple" version, the "decision-making" version and the "power-class" version. This translation was implemented following the SPSS script from Håkon Leiulfstrud and Heidi Jensberg. For more information, please contact the authors.

For more information on this class schema, please check the references below:

- Leifsrud, H., I. Bison & H. Jensberg (2005) Social Class in Europe: European Social Survey 2002/3. Official ESS Report. NTNU Social Research in Trondheim & Department of Social Research, Trento University. [https://www.europeansocialsurvey.org/docs/methodology/ESS1\\_social\\_class.pdf](https://www.europeansocialsurvey.org/docs/methodology/ESS1_social_class.pdf)
- Leifsrud, H., Bison, I. and Solheim, E. (2010), Social Class in Europe II: The European Social Survey 2002–2008, Trondheim: NTNU. [https://www.researchgate.net/publication/317624268\\_Social\\_Class\\_in\\_Europe\\_II](https://www.researchgate.net/publication/317624268_Social_Class_in_Europe_II)
- Wright, E. (1978): Class, Crises and the State. London: New Left Books.
- Wright, E. O. (1985): Classes. London: New Left Books.
- Wright, E. O. (1997): Class Counts: Comparative studies in class analyses. Cambridge: Cambridge University Press.
- Wright, E. O. (2005) Approaches to Class Analysis. Cambridge: Cambridge University Press.

## Examples

```
library(dplyr)

# E.O Wright - Simple translation
ess %>%
  transmute(
    isco88com,
    wr_simple = isco88com_to_wright(
      isco88com,
      is_supervisor,
      self_employed,
      emplno,
      control_work,
      control_daily,
      type = "simple"
    ),
    wr_simple_label = isco88com_to_wright(
      isco88com,
      is_supervisor,
      self_employed,
      emplno,
      control_work,
      control_daily,
      type = "simple",
      label = TRUE
    )
  )

# E.O Wright - Decision-making translation
ess %>%
  transmute(
    isco88com,
    wr_decision = isco88com_to_wright(
      isco88com,
      is_supervisor,
      self_employed,
```



```

    emplno,
    control_work,
    control_daily,
    type = "decision-making"
  ),
  wr_decision_label = isco88com_to_wright(
    isco88com,
    is_supervisor,
    self_employed,
    emplno,
    control_work,
    control_daily,
    type = "decision-making",
    label = TRUE
  )
)

# E.O Wright - Power-class translation
ess %>%
  transmute(
    isco88com,
    wr_power = isco88com_to_wright(
      isco88com,
      is_supervisor,
      self_employed,
      emplno,
      control_work,
      control_daily,
      type = "power-class"
    ),
    wr_power_label = isco88com_to_wright(
      isco88com,
      is_supervisor,
      self_employed,
      emplno,
      control_work,
      control_daily,
      type = "power-class",
      label = TRUE
    )
  )
)

```

---

isco88\_to\_egp

*Translate 4-digit ISCO88/ISCO68 to EGP11/EGP7/EGP5/EGP3*


---

### Description

This function translates a vector of 4-digit ISCO88/ISCO68 codes to EGP11/EGP7/EGP5/EGP3 codes using the translation tables stored in `all_schema$isco88_to_egp11` / `all_schema$isco68_to_egp11` / `all_schema$egp11_to_egp7` / `all_schema$egp11_to_egp5` / `all_schema$egp11_to_egp3`.

**Usage**

```

isco88_to_egp(
  x,
  self_employed,
  n_employees,
  n_classes = 11,
  label = FALSE,
  to_factor = FALSE
)

isco68_to_egp(
  x,
  self_employed,
  n_employees,
  n_classes = 11,
  label = FALSE,
  to_factor = FALSE
)

```

**Arguments**

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>self_employed</code>	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
<code>n_employees</code>	A numeric vector indicating the number of employees under each respondent.
<code>n_classes</code>	a numeric value indicating the number of EGP classes to obtain. Default is 11 EGP classes. The possible values are 11 classes, 7 classes, 5 classes and 3 classes. For more information, see the details section.
<code>label</code>	A logical value indicating whether to return the labels of the translated EGP codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for EGP found in <code>all_labels</code> (default is FALSE).

**Details**

This function works by first translating to EGP11 and then translating to other EGP variants, if the user has requested this through the `n_classes` argument.

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for `ISCO88/IS68 -> EGP11`. For translations between EGP11 and EGP7/EGP5/EGP3, see the source of the `occupar` R package [here](#).

Users can see the translation used in this package in `all_schema$egp11_to_egp7`, `all_schema$egp11_to_egp5` and `all_schema$egp11_to_egp3`. Moreover, the labels used can be found in `all_labels$egp11`, `all_labels$egp7`, `all_labels$egp5` and `all_labels$egp3`.

For more details on this class schema, please check the references below:

- Erikson, R., Goldthorpe, J.H. and Portocarero, L (1979) Intergenerational Class Mobility in Three Western European Societies: England, France and Sweden', *British Journal of Sociology*, 30.
- Erikson, R., J.H. Goldthorpe, L. Portocarero (1983) Intergenerational Class Mobility and the Convergence Thesis: England, France and Sweden. *British Journal of Sociology*, 34(3): 303-343.
- Erikson R, Goldthorpe JH (1992) *The Constant Flux: A Study of Class Mobility in Industrial Societies*. Oxford: Clarendon Press.
- Goldthorpe class scheme. Oxford Reference. Retrieved 19 May. 2023, from <https://www.oxfordreference.com/view/10.1093/oi/authority.20110803095858703>.
- Goldthorpe JH (2007) Social class and the differentiation of employment contracts. In: Goldthorpe JH (ed.) *On Sociology: Volume Two*. Stanford: Stanford University Press, 101–124.

### Value

A character vector of EGP codes.

### Examples

```
library(dplyr)

# isco88
ess %>% transmute(
  isco88,
  egp11 = isco88_to_egp(isco88, self_employed, emplno, label = FALSE),
  egp7 = isco88_to_egp(isco88, self_employed, emplno, n_classes = 7, label = FALSE),
  egp5 = isco88_to_egp(isco88, self_employed, emplno, n_classes = 5, label = FALSE),
  egp3 = isco88_to_egp(isco88, self_employed, emplno, n_classes = 3, label = FALSE),
)

# isco68
ess %>% transmute(
  isco68,
  egp11 = isco68_to_egp(isco68, self_employed, emplno, label = FALSE),
  egp7 = isco68_to_egp(isco68, self_employed, emplno, n_classes = 7, label = FALSE),
  egp5 = isco68_to_egp(isco68, self_employed, emplno, n_classes = 5, label = FALSE),
  egp3 = isco68_to_egp(isco68, self_employed, emplno, n_classes = 3, label = FALSE),
)
```

## Description

This function translates a vector of 4-digit ISCO88/ISCO68 codes to EGP-MP codes using the translation tables stored in `all_schema$isco88_to_egp11` / `all_schema$isco68_to_egp11`. After translating to EGP using these tables, this function reassigns managers and professionals (ISCO88/ISCO68 codes 1 and 2) to have both high/low managers and professionals. Note that this function translates to EGP11 (not EGP7/EGP5/EGP3) and then reassigns categories to have both high/low managers and professionals. **Note that this translation uses EGP11.**

## Usage

```
isco88_to_egp_mp(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)

isco68_to_egp_mp(
  x,
  is_supervisor,
  self_employed,
  n_employees,
  label = FALSE,
  to_factor = FALSE
)
```

## Arguments

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>is_supervisor</code>	A numeric vector indicating whether each individual is a supervisor (1, e.g. responsible for other employees) or not (0).
<code>self_employed</code>	A numeric vector indicating whether each individual is self-employed (1) or an employee (0).
<code>n_employees</code>	A numeric vector indicating the number of employees under each respondent.
<code>label</code>	A logical value indicating whether to return the labels of the translated EGP-MP codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the sorted codes of EGP-MP which can be found in the source code of each function.

## Details

EGP-MP is a class schema similar to EGP but reassigns managers and professionals (ISCO88/ISCO68 codes 1 and 2) to have both high/low managers and professionals.

This schema is a slight variation of the original EGP and the logic used to build this is like this:

- All occupations with EGP digit 1 and ISCO 1-digit 0 or 1 or has subordinates, **is a high manager**
- All occupations with EGP digit 1 and is self-employed with more than 1 employee, **is a high manager**
- All occupations with EGP digit 1 and has a 1-digit ISCO higher than 1 and is either an employee or a self-employed with no subordinates, is a **high professional**
- All occupations with EGP digit 2 and ISCO 1-digit 0 or 1 or has subordinates, is a **lower manager**
- All occupations with EGP digit 2 and is self-employed with more than 1 employee, is a **lower manager**
- All occupations with EGP digit 2 and has a 1-digit ISCO higher than 1 and is either an employee or a self-employed with no subordinates, is a **lower professional**

This translation was created from the Stata do file shared by Oscar Smallenbroek called "EGP-MP.do". For more info, please contact the author.

For more information on this class schema, please check the references below:

- Smallenbroek O, Hertel F, Barone C (2022) Measuring class hierarchies in post-industrial societies: a criterion and construct validation of EGP and ESEC across 31 countries. *Sociological Methods & Research*. Epub ahead of print 11 November. <https://doi.org/10.1177/00491241221134522>

## Value

A character vector of EGP-MP codes.

## Examples

```
library(dplyr)

# isco88
ess %>%
  transmute(
    isco88,
    egp_mp = isco88_to_egp_mp(
      isco88,
      is_supervisor,
      self_employed,
      emplno,
      label = FALSE
    ),
    egp_mp_label = isco88_to_egp_mp(
      isco88,
      is_supervisor,
      self_employed,
      emplno,
      label = TRUE
    )
  )
```

```

# isco68
ess %>%
  transmute(
    isco68,
    egp_mp = isco68_to_egp_mp(
      isco68,
      is_supervisor,
      self_employed,
      emplno,
      label = FALSE
    ),
    egp_mp_label = isco68_to_egp_mp(
      isco68,
      is_supervisor,
      self_employed,
      emplno,
      label = TRUE
    )
  )

```

---

isco88\_to\_isco08

*Translate 4-digit ISCO88/ISCO68 to ISCO08*


---

### Description

This function translates a vector of 4-digit ISCO88/ISCO68 codes to ISCO08 codes using the translation tables stored in `all_schema$isco88_to_isco08` / `all_schema$isco68_to_isco08`.

### Usage

```
isco88_to_isco08(x, label = FALSE, to_factor = FALSE)
```

```
isco68_to_isco08(x, label = FALSE, to_factor = FALSE)
```

### Arguments

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>label</code>	A logical value indicating whether to return the labels of the translated ISCO08 codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ISCO08 found in <code>all_labels</code> (default is FALSE).

### Details

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for ISCO88/ISCO68 -> ISCO08.

**Value**

A character vector of ISCO08 codes.

**Examples**

```
library(dplyr)

# isco88
ess %>%
  transmute(
    isco88,
    isco08 = isco88_to_isco08(isco88, label = FALSE),
    isco08_label = isco88_to_isco08(isco88, label = TRUE)
  )

# isco68
ess %>%
  transmute(
    isco68,
    isco08 = isco68_to_isco08(isco68, label = FALSE),
    isco08_label = isco68_to_isco08(isco68, label = TRUE)
  )
```

---

isco88\_to\_isco68

*Translate 4-digit ISCO88 to ISCO68*

---

**Description**

This function translates a vector of 4-digit ISCO88 codes to ISCO68 codes using the translation table stored in `all_schema$isco88_to_isco68`.

**Usage**

```
isco88_to_isco68(x, label = FALSE, to_factor = FALSE)
```

**Arguments**

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>label</code>	A logical value indicating whether to return the labels of the translated ISCO68 codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ISCO68 found in <code>all_labels</code> (default is FALSE).

**Details**

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for ISCO88 -> ISCO68.

**Value**

A character vector of ISCO68 codes.

**Examples**

```
library(dplyr)

# isco88
ess %>%
  transmute(
    isco88,
    isco68 = isco88_to_isco68(isco88, label = FALSE),
    isco68_label = isco88_to_isco68(isco88, label = TRUE)
  )
```

---

isco88\_to\_isco88com     *Translate 4-digit ISCO88 to ISCO88COM*

---

**Description**

This function translates a vector of 4-digit ISCO88 codes to ISCO88COM codes using the translation table stored in `all_schema$isco88_to_isco88com`.

**Usage**

```
isco88_to_isco88com(x, label = FALSE, to_factor = FALSE)
```

**Arguments**

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>label</code>	A logical value indicating whether to return the labels of the translated ISCO88COM codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ISCO88COM found in <code>all_labels</code> (default is FALSE).

**Details**

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for ISCO88 -> ISCO88COM.

**Value**

A character vector of ISCO88COM codes.



**Examples**

```
library(dplyr)

ess %>%
  transmute(
    isco88,
    isco88com = isco88_to_isco88com(isco88, label = FALSE),
    isco88com_label = isco88_to_isco88com(isco88, label = TRUE)
  )
```

---

isco88\_to\_mps

*Translate 4-digit ISCO88 to MPS*


---

**Description**

This function translates a vector of 4-digit ISCO88 codes to MPS codes using the translation table stored in `all_schema$isco88_to_mps`.

**Usage**

```
isco88_to_mps(x, to_factor = FALSE)
```

**Arguments**

`x` A character vector of 4-digit ISCO codes.

`to_factor` A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the sorted codes of MPS in `all_schemas$isco88_to_mps`.

**Details**

This translation was taken from the `iscogen` Stata package. For more details, check out the package documentation and search for ISCO88 -> MPS.

**Value**

A character vector of MPS codes.

**Examples**

```
library(dplyr)

ess %>%
  transmute(
    isco88,
    mps = isco88_to_mps(isco88),
  )
```

---

isco88_to_ordc	<i>Translate 4-digit ISCO88 to ORDC</i>
----------------	-----------------------------------------

---

### Description

This function translates a vector of 4-digit ISCO88 codes to ORDC codes using the translation table stored in `all_schema$isco88_to_ordc`. ORDC stands for Oslo Register Data Class.

### Usage

```
isco88_to_ordc(x, income = NULL, label = FALSE, to_factor = FALSE)
```

### Arguments

<code>x</code>	A character vector of 4-digit ISCO codes.
<code>income</code>	A numeric vector with the income corresponding to each respondent. See the details section for more information on how this is used.
<code>label</code>	A logical value indicating whether to return the labels of the translated ORDC codes (default is FALSE).
<code>to_factor</code>	A logical value indicating whether to return a factor instead of a character. The order of the labels is taken from the labels for ORDC found in <code>all_labels</code> (default is FALSE).

### Details

The translation implemented in this function comes from the tables found in [https://journals.sagepub.com/doi/suppl/10.1177/00031224211020012/suppl\\_file/sj-pdf-1-asr-10.1177\\_00031224211020012.pdf](https://journals.sagepub.com/doi/suppl/10.1177/00031224211020012/suppl_file/sj-pdf-1-asr-10.1177_00031224211020012.pdf). That table is the appendix of the paper "Wealth Accumulation and Opportunity Hoarding: Class-Origin Wealth Gaps over a Quarter of a Century in a Scandinavian Country" from Nordli and Toft.

If `income` is specified, occupations in the economic upper class, the economic upper-middle class and the economic lower-middle class are grouped according to their income. The top 10 percent are assigned to the upper class, the next 40 percent to the upper-middle class, and the lowest 50 percent of earners to the lower-middle class. If `income` is not specified, a direct match to the ORDC class schema is performed.

For more information on this class schema, please check the references below:

- Hansen, M. N., & Toft, M. (2021). Wealth Accumulation and Opportunity Hoarding: Class-Origin Wealth Gaps over a Quarter of a Century in a Scandinavian Country. *American Sociological Review*, 86(4), 603–638. <https://doi.org/10.1177/00031224211020012>

### Value

A character vector of ORDC codes.

## Examples

```
library(dplyr)

# isco88
ess %>%
  transmute(
    isco88,
    ordc = isco88_to_ordc(isco88, label = FALSE),
    ordc_label = isco88_to_ordc(isco88, label = TRUE)
  )
```

---

issp

*An example data frame with data from the International Social Survey Programme (ISSP).*

---

## Description

Just an example data frame to show how the package can be used to translate between schemas. The source can be found at <https://issp.org/data-download/by-year/>.

## Format

Data Frame

## Details

The columns in the data frame refer to:

- `isco08`: The ISCO08 class schema. Currently in 4-digits.
- `empr1`: The employee relationship of the respondent. Could be self-employed, employee or in a family business.
- `nsup`: Number of subordinates, if there are any.
- `wrksup`: Whether the respondent has subordinates, or in other words, whether the respondent is a supervisor.

For how to do a proper analysis of this data in DIGCLASS and how to recode this data set, see the vignette "Case study: Exploring Occupations Using The International Social Survey Programme (ISSP)" in the package website.

---

repair_isco	<i>Repair an ISCO variable</i>
-------------	--------------------------------

---

### Description

ISCO variables need have certain properties. In particular, they must contain occupations of a single digit. This means that all occupations need to be either 4 digits, 3 digits, 2 digits or 1 digit. The ISCO variable can't have 3 digits and 2 digits at the same time.

### Usage

```
repair_isco(x, digits = 4)
```

### Arguments

x	ISCO variable to repair
digits	The baseline digits that the function should expect. This is whether the variable is 4/3/2/1 digits.

### Details

When reading datasets, it might happen that the ISCO column is read as a numeric column and occupations such as 0140 are converted to 140. What this function does is a bit of sanity check:

- Checks if the provided ISCO variable is a character vector, and if not, warns the user that numeric ISCO variables may contain lost data.
- Checks that all occupations have the same number of digits and warns if not
- Converts all occupations with digits less than digits to have the same number of digits by appending 0 from the left until all have the same number of digits.

Even if you're working with 3/2/1 digits, your ISCO codes should have 4 digits. So instead of having an ISCO code 241, DIGCLASS expects 2410, which is a 3-digit code.

### Value

Repaired ISCO variable

### Examples

```
repair_isco(c("1234", "5678", "9012"))  
  
## Not run:  
repair_isco(c("123", "5678", "012"))  
  
## End(Not run)
```

# Index

all\_labels, [2](#)  
all\_schemas, [3](#)

esco\_to\_isco08, [3](#)  
ess, [4](#)

isco08\_swap, [5](#)  
isco08\_to\_esec, [7](#)  
isco08\_to\_esec\_mp, [10](#)  
isco08\_to\_eseg, [13](#)  
isco08\_to\_ipics, [15](#)  
isco08\_to\_isco88, [17](#)  
isco08\_to\_isei, [18](#)  
isco08\_to\_microclass, [19](#)  
isco08\_to\_msec, [20](#)  
isco08\_to\_oesch, [22](#)  
isco08\_to\_siops, [24](#)  
isco08\_two\_to\_esec, [26](#)  
isco68\_swap (isco08\_swap), [5](#)  
isco68\_to\_egp (isco88\_to\_egp), [30](#)  
isco68\_to\_egp\_mp (isco88\_to\_egp\_mp), [32](#)  
isco68\_to\_isco08 (isco88\_to\_isco08), [35](#)  
isco68\_to\_isco88 (isco08\_to\_isco88), [17](#)  
isco68\_to\_isei (isco08\_to\_isei), [18](#)  
isco68\_to\_siops (isco08\_to\_siops), [24](#)  
isco88\_swap (isco08\_swap), [5](#)  
isco88\_to\_egp, [30](#)  
isco88\_to\_egp\_mp, [32](#)  
isco88\_to\_ipics (isco08\_to\_ipics), [15](#)  
isco88\_to\_isco08, [35](#)  
isco88\_to\_isco68, [36](#)  
isco88\_to\_isco88com, [37](#)  
isco88\_to\_isei (isco08\_to\_isei), [18](#)  
isco88\_to\_mps, [38](#)  
isco88\_to\_oesch (isco08\_to\_oesch), [22](#)  
isco88\_to\_ordc, [39](#)  
isco88\_to\_siops (isco08\_to\_siops), [24](#)  
isco88com\_to\_esec (isco08\_to\_esec), [7](#)  
isco88com\_to\_esec\_mp  
    (isco08\_to\_esec\_mp), [10](#)  
isco88com\_to\_msec (isco08\_to\_msec), [20](#)  
isco88com\_to\_wright, [27](#)  
issp, [40](#)

repair\_isco, [41](#)

## **GETTING IN TOUCH WITH THE EU**

### **In person**

All over the European Union there are hundreds of Europe Direct centres. You can find the address of the centre nearest you online ([european-union.europa.eu/contact-eu/meet-us\\_en](https://european-union.europa.eu/contact-eu/meet-us_en)).

### **On the phone or in writing**

Europe Direct is a service that answers your questions about the European Union. You can contact this service:

- by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),
- at the following standard number: +32 22999696,
- via the following form: [european-union.europa.eu/contact-eu/write-us\\_en](https://european-union.europa.eu/contact-eu/write-us_en).

## **FINDING INFORMATION ABOUT THE EU**

### **Online**

Information about the European Union in all the official languages of the EU is available on the Europa website ([european-union.europa.eu](https://european-union.europa.eu)).

### **EU publications**

You can view or order EU publications at [op.europa.eu/en/publications](https://op.europa.eu/en/publications). Multiple copies of free publications can be obtained by contacting Europe Direct or your local documentation centre ([european-union.europa.eu/contact-eu/meet-us\\_en](https://european-union.europa.eu/contact-eu/meet-us_en)).

### **EU law and related documents**

For access to legal information from the EU, including all EU law since 1951 in all the official language versions, go to EUR-Lex ([eur-lex.europa.eu](https://eur-lex.europa.eu)).

### **Open data from the EU**

The portal [data.europa.eu](https://data.europa.eu) provides access to open datasets from the EU institutions, bodies and agencies. These can be downloaded and reused for free, for both commercial and non-commercial purposes. The portal also provides access to a wealth of datasets from European countries.

# Science for policy

The Joint Research Centre (JRC) provides independent, evidence-based knowledge and science, supporting EU policies to positively impact society



**EU Science Hub**

[joint-research-centre.ec.europa.eu](https://joint-research-centre.ec.europa.eu)



[@EU\\_ScienceHub](https://twitter.com/EU_ScienceHub)



[EU Science Hub - Joint Research Centre](https://www.facebook.com/EU_Science_Hub_-_Joint_Research_Centre)



[EU Science, Research and Innovation](https://www.linkedin.com/company/eu-science-research-and-innovation)



[EU Science Hub](https://www.youtube.com/EU_Science_Hub)



[@eu\\_science](https://www.instagram.com/eu_science)