

Generation dupID_gen

=====

This variable indicates how a spell relates within an ObjektID to the prior spell. It is defined as follows:

0: ObjektID only once or 1st spell of ObjektID

1: Probably belongs together, e.g. interruption due to delivery to RWI or slight adaptation of the advertisement

2: As (1), but gap between advertisements >6 months, i.e. it could also be a new rental of the apartment e.g.

3: Advertisements close together in time (≤ 6 months), but differentiation in important variables

4: As (3), but gap between advertisements >6 months

5: Clear differences in important variables, so there are doubts that it is the same object.

For the generation the ado file GenDupID is used, which can also be changed by the data end user to adapt the definitions.

prerequisites

The following prerequisites must be fulfilled so that the ado file runs correctly:

1. Missings of numerical variables used in the definition of similar dwellings must be coded as missing in Stata (not as negative values).

Alternatively, there is the option to pass a variable list of variables to the program in the option negmissings(varlist), which must be transcoded accordingly. In this case and if type 5 is calculated, negative values of the under 3. listed variables are also replaced to missings.

2. It needs variables for the start and end date of a spell that are formatted as a date format on a monthly basis.

e.g. gen adat=ym(ayear, amonth)

If type 5 is to be calculated by the program, the following variables must exist:

- kaufpreis,
- mietekalt,
- mietewarm,
- wohnflaeche,
- grundstuecksflaeche,
- zimmeranzahl.

Alternatively, the classification of type 5 can also be done manually after running the program, as described under Input/Optional.

Inputs

The following syntax must be maintained:

GenDupID varname, gen(string) duplist1(varlist) adat(varname numeric) edat(varname numeric)

where

- varname denotes the ObjektID,
- is the name of the classification to be generated (e.g. dupID_gen),
- duplist1 a list of existing variables used to characterize similar Spells (Type 1 and Type 2)
- adat and edat denote the start and end date of the spell, as described above.

optional

The following parameters can be passed optionally:

- duplist2() and duplist3(), which are defined like duplist1(). A spell is then classified as type 1 or 2 if one of the definitions matches (OR-connection).
- notyfive: If the data user prefers to define type 5 himself, these spells are defined as type 3 or type 4 based on the gap to the start spell. Type 5 can then easily be adjusted manually after running the program.

The default classification is described under Defaults.

- months(integer): The default value is 6 months, i.e. the gaps used to distinguish between type 1 and type 2 or type 3 and type 4 are more than 6 months. Any integer value can be entered here.
- nosum: The program outputs smaller descriptive statistics during the run. If this is not desired, it can be turned off with this option.

defaults

months

A default value of >6 months is given for differentiating between type 1 and type 2 or type 3 and type 4.

Type 5

A spell is classified as type 5 if one of the conditions below is met (OR-connection):

- Purchase price (Kaufpreis) varies by more than 20%,
- Hot or cold rent(Warm- oder Kaltmiete) deviates by more than 20%,
- Living space (Wohnfläche) deviates by more than 10%,
- Plot area (Grundstücksfläche) deviates by more than 10%,
- Number of rooms (Zimmeranzahl) varies by more than 1.

Type 1/2

The following three definitions are used to identify 'similar' dwellings:

* Def 1: Generally very similar features

mietekalt mietewarm heizkosten_in_wm_enthalten /// rent only

kaufpreis grundstuecksflaeche /// purchase only

etage /// apartment only

baujahr letzte_modernisierung wohnflaeche anzahltagen zimmeranzahl ///

schlafzimmer badezimmer energieeffizienzklasse ausstattung

* Def 2: Without manipulation

Here we calculate dup without manipulation variables, e.g. rent, if apartment was not rented for a longer period of time. Note that we do not allow the manipulation of a postal code, even if it's plausible. But this is allowed in dup1.

The following variables are included:

plz ///

grundstuecksflaeche /// purchase only

etage /// apartment only

baujahr wohnflaeche anzahltagen zimmeranzahl ///

schlafzimmer badezimmer