

[Description](#)
Remarks and examples

[Quick start](#)
Stored results

[Menu](#)
References

[Syntax](#)
Also see

[Options](#)

Description

`icd9p` is a suite of commands for working with ICD-9-CM procedure codes from the 16th version (effective October 1998) to the 32nd version. To see the current version of the ICD-9-CM procedure codes and any changes that have been applied, type `icd9p query`.

`icd9p check`, `icd9p clean`, and `icd9p generate` are data management commands. `icd9p check` verifies that a variable contains defined ICD-9-CM procedure codes and provides a summary of any problems encountered. `icd9p clean` standardizes the format of the codes. `icd9p generate` can create a binary indicator variable for whether the code is in a specified set of codes, a variable containing a corresponding higher-level code, or a variable containing the description of the code.

`icd9p lookup` and `icd9p search` are interactive utilities. `icd9p lookup` displays descriptions of the codes specified on the command line. `icd9p search` looks for relevant ICD-9-CM procedure codes from keywords given on the command line.

Quick start

Determine whether ICD-9-CM procedure codes in `proc1` are invalid, and store reasons in `invalid`

```
icd9p check proc1, generate(invalid)
```

Standardize display of codes in `proc2` to remove all periods

```
icd9p clean proc2
```

Create `descr3` as the procedure code prepended to short description of procedure code in `proc3`

```
icd9p generate descr3 = proc3, description long
```

Create `eye` as an indicator for eye surgery in `proc4` using ICD-9-CM procedure codes 16.1 through 16.99

```
icd9p generate eye = proc4, range(16*)
```

Look up descriptions for ICD-9-CM procedure codes 25.0 through 25.4 and 25.9 through 25.99

```
icd9p lookup 25.0/25.4 25.9*
```

Menu

Data > ICD codes > ICD-9

Syntax

Verify that variable contains defined codes

```
icd9p check varname [ if ] [ in ] [ , any list generate(newvar) ]
```

Clean variable and verify format of codes

```
icd9p clean varname [ if ] [ in ] [ , dots pad ]
```

Generate new variable from existing variable

```
icd9p generate newvar = varname [ if ] [ in ] , category
```

```
icd9p generate newvar = varname [ if ] [ in ] , description [ long end ]
```

```
icd9p generate newvar = varname [ if ] [ in ] , range(codelist)
```

Display code descriptions

```
icd9p lookup codelist
```

Search for codes from descriptions

```
icd9p search [ " ] text [ " ] [ [ " ] text [ " ] ... ] [ , or ]
```

Display ICD-9 code source

```
icd9p query
```

codelist is

<i>icd9code</i>	(the particular code)
<i>icd9code*</i>	(all codes starting with)
<i>icd9code/icd9code</i>	(the code range)

or any combination of the above, such as 50.21 37.7* 88.71/88.79. *icd9codes* must be typed with leading 0s. For example, type 01; typing 1 will result in an error.

collect is allowed with icd9p check, icd9p clean, and icd9p lookup; see [\[U\] 11.1.10 Prefix commands](#).

The icd9p suite of commands does not allow alias variables; see [\[D\] frunalias](#) for advice on how to get around this restriction.

Options

Options are presented under the following headings:

Options for icd9p check
Options for icd9p clean
Options for icd9p generate
Option for icd9p search

Options for icd9p check

`any` tells `icd9p check` to verify that the codes fit the format of ICD-9-CM procedure codes but not to check whether the codes are defined.

`list` specifies that `icd9p check` list the observation number, the invalid or undefined ICD-9-CM procedure code, and the reason the code is invalid or whether it is an undefined code.

`generate`(*newvar*) specifies that `icd9p check` create a new variable containing, for each observation, 0 if the observation contains a defined code or is missing. Otherwise, it contains a number from 1 to 10. The positive numbers indicate the kind of problem and correspond to the listing produced by `icd9p check`.

Options for icd9p clean

`dots` specifies that the period be included in the final format. If `dots` is not specified, then all periods are removed.

`pad` specifies that `icd9p clean` pad the codes with spaces, front and back, to make the (implied) dots align vertically in listings. Specifying `pad` makes the resulting codes look better when used with most other Stata commands.

Options for icd9p generate

`category`, `description`, and `range`(*codelist*) specify the contents of the new variable that `icd9p generate` is to create. You do not need to `icd9p clean varname` before using `icd9p generate`; it will accept any supported format or combination of formats.

`category` creates a new variable that contains ICD-9-CM procedure category codes. The resulting variable may be used with the other `icd9p` subcommands. For procedure codes, the category code is the first two characters.

`description` creates *newvar* containing descriptions of the ICD-9-CM procedure codes.

`long` is for use with `description`. It specifies that the code be prepended to the text describing the code.

`end` modifies `long` (specifying `end` implies `long`) and places the code at the end of the string.

`range`(*codelist*) creates a new indicator variable equal to 1 when the ICD-9-CM procedure code is in the range specified, equal to 0 when the ICD-9-CM procedure code is not in the range, and equal to missing when *varname* is missing.

Option for icd9p search

`or` specifies that ICD-9-CM procedure codes be searched for descriptions that contain any word specified with `icd9p search`. The default is to list only descriptions that contain all the words specified.

Remarks and examples

Remarks are presented under the following headings:

Verifying and cleaning variables
Interactive utilities
Creating new variables

If you have not yet read *Introduction to ICD coding* in [D] **icd**, please do so before using the `icd9p` commands. Please also see *Using icd9 and icd9p* in [D] **icd9** for information about Stata’s implementation of the ICD-9 coding system.

Throughout the remainder of this entry, we use `nhds2010.dta`, an extract of adult same-day discharges from the 2010 National Hospital Discharge Survey (NHDS). Below we *describe* the data.

```
. use https://www.stata-press.com/data/r19/nhds2010
(Adult same-day discharges, 2010)

. describe

Contains data from https://www.stata-press.com/data/r19/nhds2010.dta
Observations:      2,210      Adult same-day discharges, 2010
Variables:         15        30 Jan 2024 15:03
                        (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
ageu	byte	%8.0g	ageu	Units for age
age	byte	%8.0g		Age
sex	byte	%8.0g	sex	Sex
race	byte	%8.0g	race	Race
month	byte	%8.0g		Discharge month
status	byte	%8.0g	status	Discharge status
region	byte	%8.0g	region	Region
atype	byte	%8.0g	atype	Type of admission
dx1	str5	%9s		Diagnosis 1
dx2	str5	%9s		Diagnosis 2
dx3	str5	%9s		Diagnosis 3 (imported incorrectly)
dx3corr	str5	%9s		Diagnosis 3 (corrected)
pr1	str4	%9s		Procedure 1
wgt	int	%12.0g		Frequency weight
recid	float	%9.0g		Order of record (raw data)

Sorted by: `recid`

Verifying and cleaning variables

`icd9p check` verifies that *varname* contains defined ICD-9-CM procedure codes and, if not, provides a full report on the problems. It is a good idea to begin with this command and fix any potential problems before proceeding to other `icd9p` commands. However, the `check` subcommand is also useful for tracking down problems when any of the other `icd9p` commands tell you that the “variable does not contain ICD-9 codes”.

`icd9p clean` modifies the variable to ensure consistency and to make subsequent output look better. This is not strictly necessary because all `icd9p` commands work equally well with cleaned or uncleaned codes. `icd9p clean` also can be used to verify that the codes in a variable conform with the ICD-9-CM procedure format, without checking to see whether the codes are defined.

➤ Example 1: Standardizing the format of codes

If we plan to do any reporting with the codes in our data, we may want to make them more readable. Suppose we want to report the primary procedure for each discharge. We can use `icd9p clean` with the `dots` option to add the period between the category code and any subsequent digits.

```
. icd9p clean pr1, dots pad
(821 changes made)

. list recid pr1 in 1/5
```

	recid	pr1
1.	84	38.34
2.	105	
3.	255	
4.	651	
5.	696	73.59

Using `icd9p clean` with undefined codes will not result in an error message. So if you are using codes from a country other than the United States, the `clean` subcommand can still be used to standardize the format of your codes and check for correct placement of the period.



Interactive utilities

`icd9p search` looks for relevant ICD-9-CM procedure codes from the description given on the command line, and `icd9p lookup` lists the descriptions of codes given on the command line. The two commands complement each other.

➤ Example 2: Finding procedure code descriptions

If we wanted to find the corresponding abbreviated description for procedure code 38.34, we would type

```
. icd9p lookup 38.34
1 match found:
  38.34    aorta resection & anast
```

If you are curious, the cryptic result translates into resection with anastomosis of the aorta.

To find a list of other procedure codes for resection with anastomosis and their descriptions, we could type `icd9p lookup 38.3*`. Or if we were interested in finding codes for procedures on the aorta, we could type

```
. icd9p search aorta
(output omitted)
```



Creating new variables

`icd9p generate` produces new variables based on existing variables containing (cleaned or uncleaned) ICD-9-CM procedure codes. `icd9p generate, category` creates *newvar* containing the category code that corresponds to the code in the existing variable. `icd9p generate, description`

creates *newvar* containing the abbreviated textual description of the ICD-9-CM procedure code. `icd9p generate, range()` produces numeric *newvar* containing 1 if *varname* records an ICD-9-CM procedure code in the range listed and containing 0 otherwise.

► Example 3: Adding descriptions to codes

In [example 4](#) of [\[D\] icd9](#), we created an indicator variable for whether a patient had congestive heart failure (CHF). We may want to know what procedures were performed for patients with CHF. We check the procedure codes in `pr1` and then generate a new variable with their descriptions. We include the `long` option so that we can see the ICD-9-CM procedure code as well.

```
. icd9p check pr1
(pr1 contains defined ICD-9-CM procedure codes; 1389 missing values)
. icd9p generate pr1descr = pr1, description long
. tabulate pr1descr [fweight=wgt] if chf==1, missing sort
```

label for pr1	Freq.	Percent	Cum.
	7,185	58.93	58.93
37.22 left heart cardiac cath	1,906	15.63	74.57
92.05 c-vasc scan/isotop funct	1,027	8.42	82.99
88.72 dx ultrasound-heart	776	6.36	89.35
03.31 spinal tap	498	4.08	93.44
39.95 hemodialysis	388	3.18	96.62
34.91 thoracentesis	138	1.13	97.75
99.60 cardiopulm resuscita nos	112	0.92	98.67
37.94 implt/repl carddefib tot	110	0.90	99.57
89.44 cardiac stress test nec	52	0.43	100.00
Total	12,192	100.00	

We see that the majority of same-day discharges (58.9%) did not involve any procedure. When a procedure was performed, the most common was left heart cardiac catheterization (15.6%).



□ Technical note

The dataset that supports `icd9p` includes all codes that were added or deleted between V16 and the last version (V32). However, the descriptions were updated with each new version. If you are using `icd9p generate` with option `description` for codes from a version other than 32, please review the `icd9p` query log for any changes to descriptions between the version you are using and version 32.



Stored results

`icd9p check` stores the following in `r()`:

Scalars

<code>r(e#)</code>	number of errors of type #
<code>r(esum)</code>	total number of errors

`icd9p clean` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of changes
-------------------	-------------------

`icd9p lookup` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of codes found
-------------------	-----------------------

References

- National Center for Health Statistics. 2011. International Classification of Diseases, Ninth Revision, Clinical Modification. https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Publications/ICD9-CM/2011/.
- . 2012. National Hospital Discharge Survey: 2010 Public Use Data File Documentation. https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NHDS/NHDS_2010_Documentation.pdf.

Also see

- [D] [icd](#) — Introduction to ICD commands
- [D] [icd9](#) — ICD-9-CM diagnosis codes
- [D] [icd10pcs](#) — ICD-10-PCS procedure codes
- [D] [frunalias](#) — Change storage type of alias variables

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).