icd10cm — ICD-10-CM diagnosis codes

Description | Quick start | Menu | Syntax | Options
Remarks and examples | Stored results | Acknowledgments | Reference | Also see

Description

`icd10cm` is a suite of commands for working with ICD-10-CM diagnosis codes from US federal fiscal year 2016 to the present. To see the current version of the ICD-10-CM diagnosis codes and any changes that have been applied, type `icd10cm query`.

`icd10cm check`, `icd10cm clean`, and `icd10cm generate` are data management commands. `icd10cm check` verifies that a variable contains defined ICD-10-CM diagnosis codes and provides a summary of any problems encountered. `icd10cm clean` standardizes the format of the codes. `icd10cm 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.

`icd10cm lookup` and `icd10cm search` are interactive utilities. `icd10cm lookup` displays descriptions of the codes specified on the command line. `icd10cm search` looks for relevant ICD-10-CM diagnosis codes from keywords given on the command line.

Quick start

Determine whether ICD-10-CM diagnosis codes in `diag1` are invalid, and store reasons in `invalid`

```
icd10cm check diag1, generate(invalid)
```

Standardize display of codes in `diag2` to add a period and left-align codes

```
icd10cm clean diag2, replace
```

Generate `descr3` as the diagnosis code prepended to the short description of diagnosis code in `diag3`

```
icd10cm generate descr3 = diag3, description addcode(begin)
```

Generate `mhypertn` as an indicator for a maternal hypertension diagnosis in `diag4` using ICD-10-CM codes O16.1 through O16.5 or O16.9

```
icd10cm generate mhypertn = diag4, range(O161/O165 O169)
```

Look up descriptions for ICD-10-CM diagnosis codes T46.1X1, T46.1X1A, T46.1X1D, and T46.1X1S

```
icd10cm lookup T46.1X1*
```

Look up codes where the description contains the words “delivery” or “birth”

```
icd10cm search delivery birth, or
```

Menu

Data > ICD codes > ICD-10-CM

Syntax

Verify that variable contains defined codes

```
icd10cm check varname [if] [in] [, checkopts]
```

Clean variable and verify format of codes

```
icd10cm clean varname [if] [in], {generate(newvar)|replace} [cleanopts]
```

Generate new variable from existing variable

```
icd10cm generate newvar = varname [if] [in], category [check]
icd10cm generate newvar = varname [if] [in], description [genopts]
icd10cm generate newvar = varname [if] [in], range(codelist) [check]
```

Display code descriptions

```
icd10cm lookup codelist [, version(#)]
```

Search for codes from descriptions

```
icd10cm search ["]text["] [["]text["] ...] [, searchopts]
```

Display ICD-10-CM version

```
icd10cm query
```

codelist is one of the following:

icd10code	(the particular code)
icd10code`*`	(all codes starting with)
icd10code`/`*icd10code*	(the code range)

or any combination of the above, such as `A27.0 G40* Y60/Y69.9`.

checkopts	Description
`fmtonly`	check only format of the codes
`summary`	frequency of each invalid or undefined code
`list`	list observations with invalid or undefined ICD-10-CM codes
`generate(`*newvar*`)`	create new variable marking invalid codes
`version(#)`	fiscal year to check codes against; default is the current year

cleanopts	Description
* generate(*newvar*)	create new variable containing cleaned codes
* replace	replace existing codes with the cleaned codes
check	check that variable contains ICD-10-CM codes before cleaning
nodots	format codes without a period
pad	add space to the right of codes shorter than seven characters

* Either generate() or replace is required.

genopts	Description
addcode(begin | end)	add code to the beginning or end of the description
pad	add spaces to the right of the code; must specify addcode(begin)
nodots	format codes without a period; must specify addcode()
check	check that variable contains ICD-10-CM codes before generating new variable
long	use long description rather than short
version(#)	select description from fiscal year #; default is the current year

searchopts	Description
or	match any keyword
matchcase	match case of keywords
version(#)	search description from fiscal year #; default is all

collect is allowed with icd10cm check and icd10cm clean; see **[U] 11.1.10 Prefix commands**.

The icd10cm 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 icd10cm check
Options for icd10cm clean
Options for icd10cm generate
Option for icd10cm lookup
Options for icd10cm search

Options for icd10cm check

fmtonly tells icd10cm check to verify that the codes fit the format of ICD-10-CM diagnosis codes but not to check whether the codes are defined.

summary specifies that icd10cm check should report the frequency of each invalid or undefined code that was found in the data. Codes are displayed in descending order by frequency. summary may not be combined with list.

list specifies that icd10cm check list the observation number, the invalid or undefined ICD-10-CM diagnosis code, and the reason the code is invalid or whether it is an undefined code. list may not be combined with summary.

`generate(`*newvar*`)` specifies that `icd10cm check` create a new variable containing, for each observation, 0 if the observation contains a defined code. Otherwise, it contains a number from 1 to 11 if the code is invalid, 77 if the code is valid only for a previous version, 88 if the code is valid only for a later version, 99 if the code is undefined, or missing if *varname* is missing.. The positive numbers indicate the kind of problem and correspond to the listing produced by `icd10cm check`.

`version(#)` specifies the version of the codes that `icd10cm check` should reference. # indicates the federal fiscal year for the codes. For example, use `2016` for federal fiscal year 2016 (FFY-2016), which is October 1, 2015 to September 30, 2016. `icd10cm` supports all years after the United States officially adopted ICD-10-CM. The appropriate value of # should be determined from the data source. The default is the current year.

Warning: The default value of `version()` will change over time so that the most recent codes are used. Using the default value rather than specifying a specific version may change results after a new version of the codes is introduced.

Options for icd10cm clean

`generate(`*newvar*`)` and `replace` specify how the formatted values of *varname* are to be handled. You must specify either `generate()` or `replace`.

`generate()` specifies that the cleaned values be placed in the new variable specified in *newvar*.

`replace` specifies that the existing values of *varname* be replaced with the formatted values.

`check` specifies that `icd10cm clean` should first check that *varname* contains codes that fit the format of ICD-10-CM diagnosis codes. Specifying the `check` option will slow down `icd10cm clean`.

`nodots` specifies that the period be removed in the final format.

`pad` specifies that spaces be added to the end of the codes to make the (implied) dots align vertically in listings. The default is to left-align codes without adding spaces.

Options for icd10cm generate

`category`, `description`, and `range(`*codelist*`)` specify the contents of the new variable that `icd10cm generate` is to create. You do not need to `icd10cm clean` *varname* before using `icd10cm generate`; it will accept any supported format or combination of formats.

`category` specifies to extract the three-character category code from the ICD-10-CM diagnosis code. The resulting variable may be used with the other `icd10cm` subcommands.

`description` creates *newvar* containing descriptions of the ICD-10-CM diagnosis codes.

`range(`*codelist*`)` creates a new indicator variable equal to 1 when the ICD-10-CM diagnosis code is in the range specified, equal to 0 when the ICD-10-CM diagnosis code is not in the range, and equal to missing when *varname* is missing.

`addcode(begin | end)` specifies that the code should be included with the text describing the code. Specifying `addcode(begin)` will prepend the code to the text. Specifying `addcode(end)` will append the code to the text.

`pad` specifies that the code that is to be added to the description should be padded spaces to the right of the code so that the start of description text is aligned for all codes. `pad` may be specified only with `addcode(begin)`.

nodots specifies that the code that is added to the description should be formatted without a period. nodots may be specified only if addcode() is also specified.

check specifies that icd10cm generate should first check that *varname* contains codes that fit the format of ICD-10-CM diagnosis codes. Specifying the check option will slow down the generate subcommand.

long specifies that the long description of the code be used rather than the short (abbreviated) description.

version(#) specifies the version of the codes that icd10cm generate should reference. # indicates the federal fiscal year for the codes. For example, use 2016 for federal fiscal year 2016 (FFY-2016), which is October 1, 2015 to September 30, 2016. icd10cm supports all years after the United States officially adopted ICD-10-CM. The appropriate value of # should be determined from the data source. The default is the current year.

Warning: The default value of version() will change over time so that the most recent codes are used. Using the default value rather than specifying a specific version may change results after a new version of the codes is introduced.

Option for icd10cm lookup

version(#) specifies the version of the codes that icd10cm lookup should reference. # indicates the federal fiscal year for the codes. For example, use 2016 for federal fiscal year 2016 (FFY-2016), which is October 1, 2015 to September 30, 2016. icd10cm supports all years after the United States officially adopted ICD-10-CM. The appropriate value of # should be determined from the data source. The default is the current year.

Warning: The default value of version() will change over time so that the most recent codes are used. Using the default value rather than specifying a specific version may change results after a new version of the codes is introduced.

Options for icd10cm search

or specifies that ICD-10-CM diagnosis codes be searched for descriptions that contain any word specified with icd10cm search. The default is to list only descriptions that contain all the words specified.

matchcase specifies that icd10cm search should match the case of the keywords given on the command line. The default is to perform a case-insensitive search.

version(#) specifies the version of the codes that icd10cm search should reference. # indicates the federal fiscal year for the codes. For example, use 2016 for federal fiscal year 2016 (FFY-2016), which is October 1, 2015 to September 30, 2016. icd10cm supports all years after the United States officially adopted ICD-10-CM.

By default, descriptions for all versions are searched, meaning that codes that changed descriptions and that have descriptions in multiple versions that contain the search terms will be duplicated. To ensure a list of unique code values, specify the version number.

Remarks and examples

Remarks are presented under the following headings:

Introduction
Managing datasets with ICD-10-CM codes
Interactive utilities

If you have not yet read *Introduction to ICD coding* in [D] **icd**, please do so before using the `icd10cm` commands.

Introduction

The general format of an ICD-10-CM diagnosis code is a three-character category code followed by up to four characters after an (implied) period. The first character is always a letter and the second character is always a number, but the remaining characters may be any combination of letters and numbers.

Some examples of ICD-10-CM diagnosis codes are B69 (cysticercosis) and W20.0XXA (struck by falling object in cave-in, initial encounter). Many datasets record (and some people write) codes without the period; for example, the code I74.3 may appear as I743. The `icd10cm` commands understand both ways of recording codes. The commands are also insensitive to codes recorded with or without leading and trailing blanks and are case insensitive.

All the following are acceptable formats to record codes in Stata:

```
    T37.0X3A
A25.1
    C52
       a80.0
       z8261
```

Important note: What constitutes a valid code changes between versions. For the rest of this entry, a defined code is any code that is currently valid, was valid at some point since the ICD-10-CM coding system was introduced, or has a meaning as a grouping of codes. The list of valid codes and their associated descriptions is from the US Centers for Disease Control and Prevention's National Center for Health Statistics (Centers for Disease Control and Prevention 2013). The ICD-10-CM is a licensed adaptation of the ICD-10, which is copyrighted by the World Health Organization (WHO); see [R] **Copyright ICD-10**.

To view the current version of the ICD-10-CM diagnosis codes in Stata, its source, and a log of changes that have been made to the list of ICD-10-CM diagnosis codes since the `icd10cm` commands were implemented, type

```
. icd10cm query
ICD-10-CM Diagnosis Code Version and Change Log
  Note
    The ICD-10 coding system is copyrighted by the World Health Organization.
        The ICD-10-CM is the WHO's authorized adaptation for use in the
        United States.  It is maintained by the National Center for Health
        Statistics (NCHS), at the Center for Disease Control and Prevention.
        Stata obtains the ICD-10-CM data from the NCHS website.
    See copyright icd10 for the ICD-10 copyright notification.
  (output omitted)
```

Managing datasets with ICD-10-CM codes

The `icd10cm` suite of commands has three data management commands. `icd10cm check` verifies that the ICD-10-CM diagnosis codes in *varname* are valid. `icd10cm clean` standardizes the format of ICD-10-CM diagnosis codes in *varname*. And `icd10cm generate` produces a new variable from an existing variable containing ICD-10-CM diagnosis codes.

Examples in this section use `hosp2015.dta`, a fictional sample of inpatient hospital discharges in Washington State from July 2015 to December 2015. The data were simulated based on the Comprehensive Hospital Abstract Reporting System (CHARS); see https://www.doh.wa.gov/DataandStatisticalReports/HealthcareinWashington/HospitalandPatientData/HospitalDischargeDataCHARS. Examples analyzing the procedure codes for this dataset may be found in [D] **icd10pcs**.

```
. use https://www.stata-press.com/data/r19/hosp2015
(Fictional WA hospital discharges)

. describe

Contains data from https://www.stata-press.com/data/r19/hosp2015.dta
 Observations:         3,935                  Fictional WA hospital discharges
    Variables:            18                  6 Apr 2024 13:10
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
hospid          str5    %9s                   Hospital ID
age             byte    %11.0g     age        Age (years)
sex             byte    %8.0g      sex        Sex
ins             byte    %9.0g      ins        Insurance type
los             byte    %19.0g     los        Length of stay (days)
atype           byte    %9.0g      admtype    Admission type
asource         byte    %18.0g     admsrc     Admission source
aday            byte    %8.0g      day        Admission day of week
dmonth          int     %tm                   Discharge month
dstatus         byte    %22.0g     status     Discharge status
died            byte    %8.0g                 Patient died (1=yes)
diag1           str7    %9s                   Diagnosis 1
diag2           str7    %9s                   Diagnosis 2
diag3           str7    %9s                   Diagnosis 3
proc1           str7    %9s                   Procedure 1
proc2           str7    %9s                   Procedure 2
proc3           str7    %9s                   Procedure 3
billed          float   %8.2fc                Amount billed ($1,000s)
-------------------------------------------------------------------------------
Sorted by: hospid  dmonth
```

Although not necessary, it is a good idea to begin with `icd10cm check` and fix any potential problems before proceeding to other `icd10cm` commands. By default, it verifies that *varname* contains defined ICD-10-CM diagnosis codes and, if not, tabulates the type of problems encountered.

▷ Example 1: Checking the validity of a variable

We want to verify that the primary diagnosis code (`diag1`) contains only valid ICD-10-CM diagnosis codes. Because any discharges that use ICD-10-CM diagnosis codes in our data will be from October 1, 2015 to December 31, 2015, we use `version(2016)` to specify the FFY-2016 version of ICD-10-CM. If there are invalid or undefined codes in our data, we want to see what the codes are, their frequency, and the reason they were not valid, so we add the `summary` option.

```
. icd10cm check diag1, version(2016) summary
(diag1 contains no missing values)

diag1 contains invalid codes:

    1.  Invalid placement of period                    0
    2.  Too many periods                               0
    3.  Code too short                                 0
    4.  Code too long                                  0
    5.  Invalid 1st char (not A-Z)                 1,916
    6.  Invalid 2nd char (not 0-9)                     0
    7.  Invalid 3rd char (not 0-9 A or B)              0
    8.  Invalid 4th char (not 0-9 or A-Z)              0
    9.  Invalid 5th char (not 0-9 or A-Z)              0
   10.  Invalid 6th char (not 0-9 or A-Z)              0
   11.  Invalid 7th char (not 0-9 or A-Z)              0
   77.  Valid only for previous versions               0
   88.  Valid only for later versions                  0
   99.  Code not defined                              32
                                                  ------
        Total                                     1,948

Summary of invalid and undefined codes

+-------------------------------------+
| diag1   Count   Problem             |
|-------------------------------------|
|  0389      91   Invalid 1st char    |
| 65421      57   Invalid 1st char    |
| 64511      45   Invalid 1st char    |
| 71536      33   Invalid 1st char    |
| 66411      31   Invalid 1st char    |
(output omitted)
|  4940       1   Invalid 1st char    |
|  4270       1   Invalid 1st char    |
|  1570       1   Invalid 1st char    |
| 53550       1   Invalid 1st char    |
| 64413       1   Invalid 1st char    |
+-------------------------------------+
```

It looks like the records with problems used ICD-9-CM codes instead of ICD-10-CM codes. We could confirm our suspicion by using `icd9 check` or `icd9 lookup` to see whether the codes are defined in the ICD-9-CM coding system.

Because our data span the date the US switched to ICD-10-CM (October 1, 2015), we create an indicator for whether the record should use ICD-10-CM based on the date of discharge (`dmonth`). We then run `icd10cm check` again for only these records.

```
. generate use10 = (dmonth>=tm(2015m10))

. icd10cm check diag1 if use10==1, version(2016)
(diag1 contains defined codes; no missing values)
```

All the problems in `diag1` are before the switch, so we proceed without concern about our data.

In the `generate` command above, we used the `tm()` function, which lets us easily provide date values to Stata in string form; see [D] **Datetime** for more information about working with dates.

◁

If we wanted to check codes in more than one diagnosis variable, we could use a `foreach` loop or `reshape` our data; see *Working with multiple codes* in [D] **icd**. Also, additional options for `icd10cm check` help you identify the source of any errors. For example, you can obtain a list of observations that have invalid codes. See *Options for icd10cm check*.

icd10cm clean formats the variable to ensure consistency and to make subsequent output from other commands such as list and tabulate look better. icd10cm clean also can be used to verify that the codes in a variable conform to the ICD-10-CM format, without checking to see whether the codes are defined.

Example 2: Creating a variable with standardized codes

We would like to find the frequency of each primary diagnosis in our dataset. We can use tabulate with the sort option to see the most common primary diagnoses first.

So that the codes in diag1 are more readable in the tabulate output, we first use icd10cm clean. This adds a period after the three-character category code. We specify the pad option to make sure our codes align and store the result in the new variable pdx.

```
. icd10cm clean diag1 if use10==1, pad generate(pdx)
(1,955 missing values generated)
. tabulate pdx, sort
```

pdx	Freq.	Percent	Cum.
A41.9	105	5.30	5.30
O48.0	40	2.02	7.32
I21.4	37	1.87	9.19
O70.1	36	1.82	11.01
M17.11	33	1.67	12.68
O34.21	28	1.41	14.09
J96.01	21	1.06	15.15
M16.11	21	1.06	16.21
J18.9	20	1.01	17.22
O70.0	20	1.01	18.23
(output omitted)			
Total	1,980	100.00	

Notice that we used if with the use10 variable we created in example 1 to restrict icd10cm clean to just those diagnosis codes where the ICD-10-CM coding system should have been applied.

Aside from validating values of codes, the icd10cm command is primarily used to create inputs for other Stata commands. For example, in example 5 of [D] **icd9**, we show how to graph the frequency of category codes with descriptions, and in example 3 of [D] **icd10pcs**, we calculate average billed amounts over different procedures.

Example 3: Creating a variable indicating diagnosis

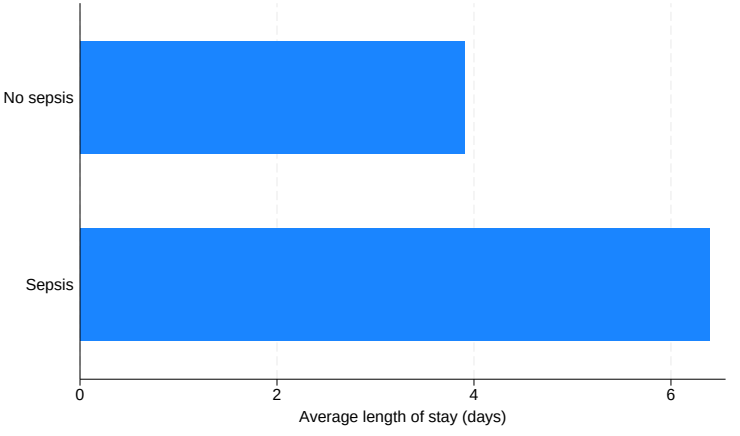
In example 2, we found that the most common primary diagnosis code in our data is A41.9, a code for a type of sepsis (a complication of infection).

Suppose we are interested in differences in length of stay (los) for discharges with and without a primary diagnosis of sepsis. We can use icd10cm generate with the range() option to search records for other diagnosis codes starting with A40, A41, and A42, which also indicate a sepsis diagnosis.

```
. icd10cm generate sepsis=diag1 if use10==1, range(A40* A41* A42*)
```

An informal way to examine differences is to plot the average length of stay for discharges with and without a sepsis diagnosis. We first label the values of our sepsis variable so that it displays nicely in the graph.

```
. label define sepsis 0 "No sepsis" 1 "Sepsis"
. label values sepsis sepsis
. graph hbar los, over(sepsis) ytitle("Average length of stay (days)")
```



More formally, we could include the new `sepsis` indicator as a factor variable in a regression model.

◁

Interactive utilities

`icd10cm lookup` and `icd10cm search` are interactive tools. You can use them without having any ICD-10-CM diagnosis data in memory.

`icd10cm lookup` lists the descriptions of codes given on the command line, and `icd10cm search` looks for relevant ICD-10-CM diagnosis codes from the specified keywords. The two commands complement each other.

▷ Example 4: Finding diagnosis codes from descriptions

In example 3, we specified codes for sepsis as any code starting with A40, A41, or A42. Suppose we want to look for other relevant codes. We can search the descriptions of the ICD-10-CM codes to locate codes of interest.

```
. icd10cm search sepsis, version(2016)
    A02.1   Salmonella sepsis
    A22.7   Anthrax sepsis
    A26.7   Erysipelothrix sepsis
    A32.7   Listerial sepsis
      (output omitted)
```

Note that `icd10cm search` is case insensitive. If you want `icd10cm search` to respect the case of the search terms you type, specify the `matchcase` option.

◁

Using `icd10cm lookup` is similar to `icd10pcs lookup`. See example 4 in [D] **icd10pcs**.

Stored results

`icd10cm check` stores the following in `r()`:

Scalars

`r(e#)`	number of errors of type #
`r(esum)`	total number of errors
`r(miss)`	number of missing values
`r(N)`	number of nonmissing values

`icd10cm clean` stores the following in `r()`:

Scalars

`r(N)`	number of changes

`icd10cm lookup` and `icd10cm search` store the following in `r()`:

Scalars

`r(N_codes)`	number of codes found

Acknowledgments

We thank the Washington State Department of Health's Center for Health Statistics for providing us with access to its 2015 Comprehensive Hospital Abstract Reporting System (CHARS) inpatient dataset. The `hosp2015` dataset used here was partially simulated based on information from the 2015 limited use CHARS. We also thank Jeanne M. Sears (retired) of the University of Washington for bringing the CHARS to our attention.

We thank Joe Canner, who while at Yale University School of Medicine, wrote `mycd10` and `mycd10p`, which provide many utilities for ICD-10 diagnosis and procedure codes. The commands rely on a user-supplied ICD-10 lookup dataset for diagnosis codes and ICD-10-PCS codes from the US Centers for Medicare and Medicaid Services for procedure codes.

Reference

Centers for Disease Control and Prevention. 2013. International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM). https://www.cdc.gov/nchs/icd/icd9cm.htm.

Also see

[D] **icd** — Introduction to ICD commands

[D] **icd9** — ICD-9-CM diagnosis codes

[D] **icd10** — ICD-10 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.