The `icd9` and `icd9p` commands are a suite of commands for working with ICD-9-CM diagnosis and ICD-9-CM procedure codes from the 16th version (effective October 1998) and later. To see the current version of the ICD-9-CM diagnosis codes and any changes that have been applied, type `icd9 query`.

`icd9 check`, `icd9 clean`, and `icd9 generate` are data management commands. `icd9 check` verifies that a variable contains defined ICD-9-CM diagnosis codes and provides a summary of any problems encountered. `icd9 clean` standardizes the format of the codes. `icd9 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.

`icd9 lookup` and `icd9 search` are interactive utilities. `icd9 lookup` displays descriptions of the diagnosis codes specified on the command line. `icd9 search` looks for relevant ICD-9-CM diagnosis codes from key words given on the command line.

`icd9p` may be used in place of `icd9` for any command above to obtain results for procedure codes.

**Quick start**

Determine if ICD-9-CM diagnosis codes in `diag1` are invalid and store reasons in `invalid`

```
icd9 check diag1, generate(invalid)
```

Standardize display of codes in `diag2` to remove all periods and align codes by padding with spaces

```
icd9 clean diag2, pad
```

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

```
icd9 generate descr3 = diag3, description long
```

Create `diabetes` as an indicator for a diabetes diagnosis in `diag4` using ICD-9-CM codes 250.xx

```
icd9 generate diabetes = diag4, range(25000/25093)
```

Look up descriptions for ICD-9-CM diagnosis codes E827.0 to E828.9

```
icd9 lookup E8270/E8289
```

Note: For any command above, `icd9p` may be used in place of `icd9` to obtain results for procedure codes.
Menu

Data > ICD codes > ICD-9

Syntax

Verify that variable contains defined codes

\{icd9|icd9p\} check varname [if] [in] [, any list generate(newvar)]

Clean variable and verify format of codes

\{icd9|icd9p\} clean varname [if] [in] [, dots pad]

Generate new variable from existing variable

\{icd9|icd9p\} generate newvar = varname [if] [in], category
\{icd9|icd9p\} generate newvar = varname [if] [in], description [long end]
\{icd9|icd9p\} generate newvar = varname [if] [in], range(codelist)

Display code descriptions

\{icd9|icd9p\} lookup codelist

Search for codes from descriptions

\{icd9|icd9p\} search ["]text["] [["]text["] ...] [, or]

Display ICD-9 code source

\{icd9|icd9p\} query

codelist is

\texttt{icd9code} \hspace{1cm} (the particular code)
\texttt{icd9code*} \hspace{1cm} (all codes starting with)
\texttt{icd9code/icd9code} \hspace{1cm} (the code range)

or any combination of the above, such as 001* 018/019 E* 018.02. \texttt{icd9codes} must be typed with leading 0s. For example, type 001 (diagnosis code) or 01 (procedure code); typing 1 will result in an error.
Options

Options are presented under the following headings:

- Options for icd9[p] check
- Options for icd9[p] clean
- Options for icd9[p] generate
- Option for icd9[p] search

Warning: The option descriptions are brief and use jargon. Please read the Remarks and examples before using the icd9 or icd9p command.

Options for icd9[p] check

any tells icd9[p] check to verify that the codes fit the format of ICD-9-CM codes but not to check whether the codes are defined.

list specifies that icd9 check list the observation number, the invalid or undefined ICD-9-CM code, and the reason the code is invalid or if it is undefined code. When specified with icd9p check, the list option references ICD-9-CM procedure codes.

generate(newvar) specifies that icd9 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 icd9 check. The values are labeled with the Stata-defined value label __icd_9.

Options for icd9[p] 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 icd9[p] 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 icd9[p] generate

category, description, and range(codelist) specify the contents of the new variable that icd9[p] generate is to create. You do not need to icd9[p] clean varname before using icd9[p] generate; it will accept any ICD-9-CM format or combination of formats.

category generates a new variable that also contains ICD-9-CM codes. The resulting variable may be used with the other icd9[p] subcommands. For procedure codes, the category code is the first two characters. For diagnostic codes, the category code is the first three characters, except for E-codes when it is the first four characters.

description creates newvar containing descriptions of the ICD-9-CM 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 code is in the range specified and equal to 0 otherwise.
Option for icd9[p] search

or specifies that ICD-9-CM codes be searched for descriptions that contain any word specified with icd9[p] search. The default is to list only descriptions that contain all the words specified.

Remarks and examples

Remarks are presented under the following headings:

Using icd9 and icd9p
Verifying and cleaning variables
Interactive utilities
Creating new variables

If you have not yet read Introduction to ICD coding in [D] icd before using the icd9 and icd9p command.

Using icd9 and icd9p

The ICD-9-CM coding system includes diagnosis and procedure codes. Some examples of diagnosis codes are 552.3 (Diaphragmatic hernia with obstruction) and E871.0 (Foreign object left in body during surgical operation). Some example of procedure codes are 01.2 (Craniotomy and craniectomy) and 55.23 (Closed renal biopsy).

Many datasets record (and some people write) codes without the period, for example, diagnosis code 550.1 may appear as 5501. The icd9 and icd9p commands understand both ways of recording codes. The commands are also insensitive to codes recorded with or without leading and trailing blanks. For E-codes and V-codes, the icd9 commands are case-insensitive. All the following codes are acceptable formats.

<table>
<thead>
<tr>
<th>Diagnosis</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>27.62</td>
</tr>
<tr>
<td>001.</td>
<td>72</td>
</tr>
<tr>
<td>000581</td>
<td>32.6</td>
</tr>
<tr>
<td>552.3</td>
<td>97.11</td>
</tr>
<tr>
<td>E800.2</td>
<td>872</td>
</tr>
<tr>
<td>e8002</td>
<td>5523</td>
</tr>
<tr>
<td>v82.2</td>
<td>08.51</td>
</tr>
</tbody>
</table>

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 version 16 (V16, effective 1 October 1998), or has meaning as a grouping of codes. The list of valid codes and their associated descriptions is from the U.S. Centers for Medicare and Medicaid Services (CMS). These codes are jointly maintained and distributed by the U.S. Centers for Disease Control and Prevention’s National Center for Health Statistics and by CMS (Centers for Disease Control and Prevention 2013).

In icd9 and icd9p, descriptions that end with an asterisk (*) are used to denote codes that are invalid for medical coding purposes but are defined as a category code or a subcategory code that has been further subdivided. For example, diagnosis code 001 (Cholera) is invalid without a fourth digit but is defined as a category code, so its description appears as cholera*. CMS does not distribute short descriptions of category and subcategory codes that are defined but not valid for coding. To ensure that Stata reports that these codes are defined, we added them to the datasets that icd9 and icd9p use with a description of *.
Codes that were valid in the past, but no longer are, have descriptions that end with a hash mark (#). For example, the diagnosis code 645.01 was deleted between V16 and V18. It remains a defined code, and its description appears as prolonged preg-delivered#.

To view the current version of ICD-9-CM diagnosis codes in Stata, its source, and a log of changes that have been made to the list of ICD-9-CM codes since the icd9 commands were implemented, type

```
.icd9 query
```

**ICD9 Diagnostic Code Mapping Data for use with Stata, History**

(output omitted)

**V32**

Dataset obtained 26aug2014 from

```text
http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/
>`
codes.html`, by selecting the 'Version 32...' file. Can be gotten directly via

```text
http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/
>`
Downloads/ICD-9-CM-v32-master-descriptions.zip`. After unzipping, the useful file name is "CMS32_DESC_SHORT_DX.txt (there are other files we did not use)."

09oct2014: V32 put into Stata distribution
BETWEEN V31 and V32: There were no additional codes.
BETWEEN V31 and V32: 0 codes were deleted.
BETWEEN V31 and V32: There were no description changes.
```

(output omitted)

icd9p query provides information about changes to procedure codes.

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 and list the first five observations for the diagnosis and procedure code variables.

```
.use http://www.stata-press.com/data/r14/nhds2010
(Adult same-day discharges, 2010 NHDS)
.describe
obs: 2,210 Adult same-day discharges, 2010
vars: 15 30 Jan 2015 15:03
size: 83,980 (_dta has notes)
```

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

Sorted by: recid
Verifying and cleaning variables

icd9[p] check verifies that varname contains defined ICD-9-CM 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 icd9[p] commands. However, the check subcommand is also useful for tracking down problems when any of the other icd9[p] commands tell you that the “variable does not contain ICD-9 codes”.

icd9[p] clean modifies the variable to ensure consistency and to make subsequent output look better. This is not strictly necessary because all icd9[p] commands work equally well with cleaned or uncleaned codes. icd9[p] clean also can be used to verify that the codes in a variable conform with the ICD-9-CM format, without checking to see if the codes are defined.

Example 1: Checking the validity of a variable

We noticed when we listed our data that dx3 appears to be padded with dashes instead of spaces. As a preemptive step, we replace the dashes with spaces by using the subinstr() function because the icd9[p] commands ignore spaces.

```
. replace dx3=subinstr(dx3,"-"," ",.)
(1,009 real changes made)
. list recid dx1 dx2 dx3 pr1 in 1/5
```

<table>
<thead>
<tr>
<th>recid</th>
<th>dx1</th>
<th>dx2</th>
<th>dx3</th>
<th>pr1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>84</td>
<td>4414</td>
<td>99811</td>
<td>14275</td>
</tr>
<tr>
<td>2.</td>
<td>105</td>
<td>25013</td>
<td>3572</td>
<td>-2506</td>
</tr>
<tr>
<td>3.</td>
<td>255</td>
<td>51909</td>
<td>1489</td>
<td>V146</td>
</tr>
<tr>
<td>4.</td>
<td>651</td>
<td>9678</td>
<td>E8528</td>
<td>8</td>
</tr>
<tr>
<td>5.</td>
<td>696</td>
<td>V271</td>
<td>64421</td>
<td>16564</td>
</tr>
</tbody>
</table>

Now that we have replaced the characters we know will be a problem, we can icd9 check the diagnosis variables. We add the generate() option so that we can identify any observations with invalid codes.
We see that all codes in \texttt{dx1} are valid and all discharges have a primary diagnosis recorded. Likewise, all codes in \texttt{dx2} are defined, and we see that 179 observations did not have a second diagnosis.

However, \texttt{icd9 check} reports that 1,000 of the 2,210 observations on \texttt{dx3} have some sort of problem: 79 codes are too short, 128 have an invalid second character, and 793 are undefined. After some investigation, we discover that when we imported the data, we started reading from the wrong position in the file. Hereafter, we use the correctly imported variable, \texttt{dx3corr}.

```
. icd9 check dx3corr
(dx3corr contains defined ICD-9-CM codes; 356 missing values)
```

Rather than typing the \texttt{icd9 check} command once for each variable, we could have checked all three simultaneously. See \textit{Working with multiple codes} in \texttt{[D] icd}.

\section*{Example 2: Standardizing the format of codes}

If we plan to do any reporting with these codes later, we may want to make them more readable. Suppose we want to report the primary diagnosis and procedure for each discharge. We can use \texttt{icd9 clean} with the \texttt{dots} and \texttt{pad} options to add the period between the category code and any subsequent digits and to align the periods.

```
. icd9 clean dx1, dots pad
(2210 changes made)
```

To standardize the procedure codes, we simply use the \texttt{icd9p} analogue to \texttt{icd9}.
. icd9p clean pr1, dots pad
(821 changes made)
. list recid dx1 pr1 in 1/5

<table>
<thead>
<tr>
<th>recid</th>
<th>dx1</th>
<th>pr1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>84</td>
<td>441.4</td>
</tr>
<tr>
<td>2.</td>
<td>105</td>
<td>250.13</td>
</tr>
<tr>
<td>3.</td>
<td>255</td>
<td>519.09</td>
</tr>
<tr>
<td>4.</td>
<td>651</td>
<td>967.8</td>
</tr>
<tr>
<td>5.</td>
<td>696</td>
<td>V27.1</td>
</tr>
</tbody>
</table>

Using `icd9[p] 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

`icd9[p] search` looks for relevant ICD-9-CM codes from the description given on the command line, and `icd9[p] lookup` lists the descriptions of codes given on the command line. The two commands complement each other.

Example 3: Finding diagnosis codes

Suppose that we want to identify the observations for which the primary diagnosis is congestive heart failure (CHF). As part of a quick exploratory analysis, we can use `icd9 search` to find ICD-9-CM codes that we may want to use to define our study population. We use the terms “heart failure” and “chf”. We enclose “heart failure” in quotation marks and use option `or` so that `icd9 search` looks for either term.

. icd9 search "heart failure" chf, or
5 matches found:
  398.91  rheumatic heart failure
  428    heart failure*
  428.0   chf nos
  428.1   left heart failure
  428.9   heart failure nos

Because the descriptions are abbreviated, we are concerned that some of the 428 codes may be left out. So, we use `icd9 lookup` to look up a range of codes.
The same result could be found by typing

```
.icd9 lookup 428/4289
```

if we knew that 428.9 was the last code in the 428 category.

### Creating new variables

`icd9[p] generate` produces new variables based on existing variables containing (cleaned or uncleaned) ICD-9-CM codes. `icd9[p] generate, category` creates `newvar` containing the category code that corresponds to the code in the existing variable. `icd9[p] generate, description` creates `newvar` containing the abbreviated textual description of the ICD-9-CM code. `icd9[p] generate, range()` produces numeric `newvar` containing 1 if `varname` records an ICD-9-CM code in the range listed and containing 0 otherwise.

#### Example 4: Creating an indicator variable

We review the list of codes we found in example 3 and decide that we will use 398.91 and all of the 428 codes in our definition of a CHF diagnosis. Now we can use `icd9 generate` with the `range()` option to create an indicator variable.

```
.icd9 generate chf = dx1, range(398.91 428*)
.tabulate chf [fweight=wgt]
```

<table>
<thead>
<tr>
<th>chf</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>563,048</td>
<td>97.88</td>
<td>97.88</td>
</tr>
<tr>
<td>1</td>
<td>12,192</td>
<td>2.12</td>
<td>100.00</td>
</tr>
</tbody>
</table>

After tabulating the results, we see that about 2.1% of all same-day discharges were for CHF in 2010.
Example 5: Adding descriptions to codes

Continuing example 4, we may wish 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 option *long* 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
```

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

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 icd9 and icd9p includes all codes that were added or deleted between V16 and the current version (V32). However, the descriptions are updated with each new version. If you are using *icd9[p] generate* with option *description* for codes other than the current version, please review the *icd9[p] query* log for any changes to descriptions between the version you are using and the current version.

Example 6: Combining commands for reporting

The *icd9[p] generate* commands are useful in isolation, but their real power comes when they are combined. For example, suppose that we want to make a graph showing the number of discharges in each diagnosis category for ICD-9-CM chapter 4, “Diseases of Blood and Blood Forming Organs”. We could use several *generate* commands and string functions, but *icd9 generate* greatly reduces our work.

First, we extract the category code from the detailed diagnosis code. Then, because the *icd9[p]* commands work equally well with complete codes or category codes, we can use *icd9 generate* with the *range(280/289)* option to create an indicator variable for whether the discharge had a primary diagnosis in chapter 4.

```
. icd9 generate dx1cat = dx1, main
. icd9 generate ch4 = dx1cat, range(280/289)
```

Next, we create a variable with the descriptions of the category codes in chapter 4.

```
. icd9 generate ch4des = dx1cat if ch4==1, description long
```
Finally, we use `graph hbar` to make a horizontal bar graph showing the frequencies of same-day discharges by diagnosis category.

```
. graph hbar (count) [fweight=wgt], over(ch4des) ytitle(Discharges)
   > title(Diseases of Blood and Blood Forming Organs, span)
   > subtitle(Same-day Discharges (2010), span)
```

![Graph showing diseases of blood and blood forming organs](image)

See [G-2] `graph bar` for information about customizing the graph above. For more information about graphing results, see [G-2] `graph`.

---

### Stored results

`icd9[p] check` stores the following in `r()`:

Scalars

- `r(e#)` number of errors of type 
- `r(esum)` total number of errors

`icd9[p] clean` stores the following in `r()`:

Scalars

- `r(N)` number of changes

`icd9[p] lookup` stores the following in `r()`:

Scalars

- `r(N)` number of codes found

### References


Also see

[D] icd — Introduction to ICD commands