

Introducing icd9: working with ICD-9 codes and comorbidities in R

Jack O. Wasey

January 26, 2015

1 Introduction

This package is designed to be used with a variety of input data, including multiple possible formats of ICD-9 codes, but some assumptions are made. There are many ways of misinterpreting ICD-9 codes, especially when dealing with ranges. The code in this package carefully considers a wide range of possibilities. **ICD-9 codes are not numeric.** Using numeric values for either decimal or non-decimal form will cause serious problems, hence the predominantly string-based processing here, and a robust set of unit tests.

When calculating which patients have which comorbidities, the input data is typically structured as follows:

```
patientData

##   visitId   icd9 poa
## 1    1000 402010   Y
## 2    1000 27801   Y
## 3    1000 7208    N
## 4    1000 25001   Y
## 5    1001 34400   N
## 6    1001 4011    Y
## 7    1002 4011    N
```

Only the visitId column is propagated to the results. The poa field indicates whether the code was determined to be present on arrival. The implicit default is to ignore it, and give ICD-9 code regardless of POA status, but filtering functions are provided and demonstrated later in this vignette.

The comorbidities can be determined as follows (showing the first ten columns for brevity):

```
icd9ComorbidAhrq(patientData)[, 1:10]

##   visitId   CHF Valvular   PHTN   PVD   HTN Paralysis NeuroOther Pulmonary   DM
## 1    1000 FALSE    FALSE FALSE FALSE FALSE    FALSE    FALSE    FALSE TRUE
## 2    1001 FALSE    FALSE FALSE FALSE TRUE     TRUE    FALSE    FALSE FALSE
## 3    1002 FALSE    FALSE FALSE FALSE TRUE     FALSE    FALSE    FALSE FALSE
```

or

```
icd9ComorbidQuanDeyo(patientData)[, 1:10]
```

##	visitId	MI	CHF	PVD	Stroke	Dementia	Pulmonary	Rheumatic	PUD	LiverMild
## 1	1000	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 2	1001	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 3	1002	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

or using magrittr

```
\%>\%
```

to chain functions together:

```
patientData %>%
  icd9FilterPoaYes() %>%
  icd9ComorbidAhrq() %>%
  extract(1:5)
```

##	visitId	CHF	Valvular	PHTN	PVD
## 1	1000	FALSE	FALSE	FALSE	FALSE
## 2	1001	FALSE	FALSE	FALSE	FALSE

The following code shows gets the same result with default options written out. Note that if

`isShort`

is not provided, the value is guessed using an internal function.

```
icd9Comorbid(icd9df = icdFilterPoaYes(patientData),
  icd9Mapping = ahrqComorbid,
  visitId = "visitId",
  icd9Field = "icd9",
  isShort = icd9:::icd9GuessIsShort(icd9df[[icd9Field]]),
  shortMapping = TRUE)[1:5]
```

2 Converting ICD-9 codes between types

These functions were designed with the common problem of incorrectly formatted ICD-9 codes in mind. These functions make the assumption that short codes of three or fewer characters are describing only the 'major' part. In most cases, when *icd9* works on ICD-9 codes, it will convert any codes of fewer than three characters into zero-padded three-digit codes.

```
icd9DecimalToShort(c("1", "10.20", "100", "123.45"))
```

##	[1]	"001"	"01020"	"100"	"12345"
----	-----	-------	---------	-------	---------

```
icd9ShortToDecimal(icd9DecimalToShort(c("1", "10.20", "100", "123.45")))
```

##	[1]	"001"	"010.20"	"100"	"123.45"
----	-----	-------	----------	-------	----------

```
icd9DecimalToShort(c("1", "22", "22.44", "1005"))
```

##	[1]	"001"	"022"	"02244"	NA
----	-----	-------	-------	---------	----

```
icd9ShortToDecimal(c("1", "22", "2244", "1005"))
```

```
## [1] "001" "022" "224.4" "100.5"

# similar with magrittr, also showing invalid codes
codes <- c("9", "87.65", "100.5", "9999", "Aesop", NA)
codes %>% icd9DecimalToShort

## [1] "009" "08765" "1005" NA NA NA

codes %>% icd9DecimalToShort %>% icd9ShortToDecimal

## [1] "009" "087.65" "100.5" NA NA NA
```

3 Validation of ICD-9 codes

```
icd9IsValidDecimal("V10.2")

## [1] TRUE

icd9IsValidShort(c("099.17", "-1"))

## [1] FALSE FALSE

icd9IsValidDecimal(c("099.17", "-1"))

## [1] TRUE FALSE

icd9IsValidShort(c("1", "001", "100", "123456", "003.21"))

## [1] TRUE TRUE TRUE FALSE FALSE
```

Validation forces the package user to provide character format ICD-9 codes. If great care is taken, passing some integers could be valid, but given the high chance of mistakes, and the simplicity of dealing entirely with character input, character is enforced:

```
icd9IsValidShort(100) # warns
```

4 Ranges of ICD-9 codes

These functions generate syntactically valid ICD-9 codes, without including parent codes when the range limit would subset the parent. E.g. "100.99" %i9d% "101.01" does not include "100" or "100.0", both of which imply large subsets than requested by the range command.

```
"10099" %i9sa% "10101"

## [1] "10099" "101" "1010" "10100" "10101"

"V10" %i9da% "V10.02"

## [1] "V10" "V10.0" "V10.00" "V10.01" "V10.02"
```

```

"E987" %i9da% "E988.1"

## [1] "E987" "E987.0" "E987.1" "E987.2" "E987.3" "E987.4" "E987.5" "E987.6" "E987.7"
## [10] "E987.8" "E987.9" "E988" "E988.0" "E988.1"

# get all possible codes
"003" %i9sa% "0033" %>% head(9) # returns 111 values

## [1] "003" "0030" "00300" "00301" "00302" "00303" "00304" "00305" "00306"

# just get the ones which correspond to billable codes (but keep the 3-digit chapters)
"003" %i9s% "0033"

## [1] "003" "0030" "0031" "00320" "00321" "00322" "00323" "00324" "00329"

```

Another way of specifying ranges are to use function calls. These are exactly equivalent to the %i9s% and %i9d% range operators. This example shows the result when the user specifies a range which would include parents but not all their children:

```

icd9ExpandRangeShort("V100", "V1002", onlyReal = TRUE) # default

## [1] "V1000" "V1001" "V1002"

icd9ExpandRangeShort("V100", "V1002", onlyReal = FALSE) # V10.0 is not billable

## [1] "V100" "V1000" "V1001" "V1002"

```

Although V100 would include ten children, the range only returns 4 values. In all other cases, parents are omitted to avoid the range returning broader classifications than intended. A planned feature is to optionally enable returning these parent codes, which would then follow a more numerical pattern (although still distinguishing trailing zeroes).

When calculating comorbidities, codes are compared to all possible children of the ranges specified in the publications. Therefore, some non-billable codes could be included which would otherwise be discarded.

We can easily find the children of a given higher-level ICD-9 code:

```

icd9Children("391", onlyReal = TRUE)

## [1] "3910" "3911" "3912" "3918" "3919"

```

Without the onlyReal flag, all syntactically correct ICD-9 codes are returned, even if not defined. This is relevant because of minor coding errors, or coding in a different year to the master list. A planned feature is to allow testing of an ICD-9 code against the valid codes for the year it was entered, but at present only the 2014 master list is used. This means that some older valid codes may not longer be on the list.

```

icd9ChildrenShort("391", onlyReal = FALSE)[1:10]

## [1] "391" "3910" "39100" "39101" "39102" "39103" "39104" "39105" "39106" "39107"

```

5 Human-readable ICD-9

There are various ways of extracting the description of the condition described by an ICD-9 code. the `icd9Explain...` functions return a data frame with a column for the ICD-9 code, a column for the full length Diagnosis, and a column for the short Description.

```
icd9ExplainDecimal("1.0")

## [1] "Cholera due to vibrio cholerae"

icd9ExplainShort("0019")

## [1] "Cholera, unspecified"


icd9Explain("1.0", isShort = FALSE)

## [1] "Cholera due to vibrio cholerae"

icd9Explain(c("0010", "4131"), isShort = TRUE)

## [1] "Cholera due to vibrio cholerae" "Prinzmetal angina"

# combine with some conversions
icd9ExplainDecimal(icd9ShortToDecimal("0019"))

## [1] "Cholera, unspecified"

"4139" %>% icd9ShortToDecimal() %>% icd9ExplainDecimal()

## [1] "Other and unspecified angina pectoris"

"413.1" %>% icd9DecimalToShort() %>% icd9ExplainShort()

## [1] "Prinzmetal angina"

# explain top level code with children
"391" %>% icd9ChildrenShort(onlyReal = TRUE)

## [1] "3910" "3911" "3912" "3918" "3919"

"391" %>% icd9ExplainShort()

## [1] "Rheumatic fever with heart involvement"

# default is to condense down to three-digit "major" level
"391" %>% icd9ChildrenShort() %>% icd9ExplainShort()

## [1] "Rheumatic fever with heart involvement"

"391" %>% icd9ChildrenShort() %>% icd9ExplainShort(doCondense = FALSE)

## [1] "Acute rheumatic pericarditis"
## [2] "Acute rheumatic endocarditis"
## [3] "Acute rheumatic myocarditis"
## [4] "Other acute rheumatic heart disease"
## [5] "Acute rheumatic heart disease, unspecified"
```

Arbitrary named list(s) of codes:

```
icd9ExplainDecimal(list(cholera = c("001", "001.0", "001.1", "001.9")))

## $cholera
## [1] "Cholera"

# same using decimal codes without a list
icd9ExplainDecimal(c("001", "001.0", "001.1", "001.9"))

## [1] "Cholera"
```

Now try to explain on a non-existent (but 'valid') ICD-9 code:

```
icd9ExplainDecimal("001.5") # gives warning
```

6 Chaining commands

With the

`magrittr`

package installed, commands can be chained together in a convenient and readable manner. This is not a dependency for this package, but is recommended because of the frequent need to chain together `icd9` commands.

For example, let's find all ICD-9 codes matching 'heart' or 'cardiac' in the short or long descriptions, first without `magrittr`:

```
icd9Hierarchy[
  grepl(
    pattern = "(heart)|(cardiac)",
    x = c(icd9Hierarchy[["descLong"]],
          icd9Hierarchy[["descShort"]]),
    ignore.case = TRUE),
  "icd9"] %>% unique -> cardiac
```

then explain the list, just showing the first ten:

```
cardiac %>% icd9ExplainShort %>% head(10)

## Warning: Some ICD codes are not 'real', e.g. NA

## [1] "Rheumatic fever without mention of heart involvement"
## [2] "Chagas' disease with heart involvement"
## [3] "Other gonococcal heart disease"
## [4] "Malignant neoplasm of heart"
## [5] "Benign neoplasm of heart"
## [6] "Other acute rheumatic heart disease"
## [7] "Acute rheumatic heart disease, unspecified"
## [8] "Rheumatic chorea with heart involvement"
## [9] "Rheumatic chorea without mention of heart involvement"
## [10] "Rheumatic heart disease, unspecified"
```

7 Explaining codes

As we have just seen, `icd9ExplainShort` (and its siblings) can convert lists of ICD-9 codes to a human-readable format. If every single member of a chapter

is quoted, the default is to report just the chapter title. This can make comprehending a complicated list much easier. E.g. in the Quan Deyo Charlson system, let's see what codes went into the *Dementia* category:

```
quanDeyoComorbid[["Dementia"]] %>% icd9ExplainShort

## Warning: Some ICD codes are not 'real', e.g. 29000 29001 29002 29003 29004

## [1] "Dementias" "Senile degeneration of brain"

icd9GetInvalidShort(quanDeyoComorbid[["Dementia"]])

## character(0)
```

Use a range with more than two hundred ICD-9 codes (most of them not real):

```
length("390" %i9da% "392.1")

## [1] 244

"390" %i9da% "392.1" %>%
  icd9DecimalToShort() %>%
  icd9ExplainShort()

## Warning: Some ICD codes are not 'real', e.g. 3900 39000 39001 39002 39003

## [1] "Rheumatic fever without mention of heart involvement"
## [2] "Rheumatic fever with heart involvement"
## [3] "Rheumatic chorea"
## [4] "Rheumatic chorea with heart involvement"

"390" %i9da% "392.1" %>% icd9ExplainDecimal()

## Warning: Some ICD codes are not 'real', e.g. 3900 39000 39001 39002 39003

## [1] "Rheumatic fever without mention of heart involvement"
## [2] "Rheumatic fever with heart involvement"
## [3] "Rheumatic chorea"
## [4] "Rheumatic chorea with heart involvement"
```

8 Filtering from Present-on-Arrival

Present-on-arrival (POA) is typically a factor, or vector of values such as "Y", "N", "X", "E", or NA. Intermediate codes, such as "exempt", "unknown" and NA mean that "yes" is not the same as "not no." This requires four functions to cover the possibilities stored in `icd9PoaChoices`:

```
icd9PoaChoices

## [1] "yes" "no" "notYes" "notNo"
```

Magrittr allows chaining of the filtering very nicely. Set up some data to demonstrate this:

```
myData <- data.frame(
  visitId = c("v1", "v2", "v3", "v4"),
  icd9 = c("39891", "39790", "41791", "4401"),
  poa = c("Y", "N", NA, "Y"),
  stringsAsFactors = FALSE)
```

Filter for present-on-arrival being "Y"

```
myData %>% icd9FilterPoaYes()
```

```
##   visitId icd9
## 1      v1 39891
## 4      v4 4401
```

Show that 'yes' is not equal to 'not no':

```
myData %>% icd9FilterPoaNotNo()
```

```
##   visitId icd9
## 1      v1 39891
## 3      v3 41791
## 4      v4 4401
```

Chain commands together to get a few columns of comorbidities. This is showing off how magrittr works in the context of this package.

```
myData %>%
  icd9FilterPoaNotNo() %>%
  icd9ComorbidAhrq(isShort = TRUE) %>%
  extract(1:9)
```

```
##   visitId CHF Valvular PHTN PVD HTN Paralysis NeuroOther Pulmonary
## 1      v1 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2      v3 FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE
## 3      v4 FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE
```

Can fill out some named fields, if we want:

```
myData %>% icd9FilterPoaYes(poaField = "poa") %>%
  icd9ComorbidAhrq(visitId = "visitId", isShort = TRUE) %>%
  extract(1:9)
```

```
##   visitId CHF Valvular PHTN PVD HTN Paralysis NeuroOther Pulmonary
## 1      v1 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2      v4 FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE
```

Call the core icd9Comorbid function with a named mapping:

```
myData %>%
  icd9FilterPoaYes() %>%
  icd9Comorbid(icd9Mapping = quanElixComorbid,
               icd9Field = "icd9", visitId = "visitId",
               isShort = TRUE, isShortMapping = TRUE
  ) %>%
  extract(1:9)
```

```
##   visitId CHF Arrhythmia Valvular PHTN PVD HTN HTNcx Paralysis
## 1      v1 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2      v4 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE
```

9 Arbitrary ICD-9 mapping

The user can provide any ICD-9 mapping they wish. Included in this package is a small data set called icd9Chapters, which lists the ICD-9-CM (and indeed ICD-9) Chapters. These can easily be expanded out and used as a mapping

```

icd9Chapters[1:5]

## `$Infectious And Parasitic Diseases`
## start end
## "001" "139"
##
## $Neoplasms
## start end
## "140" "239"
##
## `$Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders`
## start end
## "240" "279"
##
## `$Diseases Of The Blood And Blood-Forming Organs`
## start end
## "280" "289"
##
## `$Mental Disorders`
## start end
## "290" "319"

```

The next expression is obsolete since all children are now included in the packaged mappings. The user may provide their own mapping which needs this operation.

```

myMap <- icd9:::icd9ChaptersToMap(icd9Chapters[c(1,2:4)])

system.time(
  patientChapters <- icd9Comorbid(
    icd9df = patientData,
    isShort = TRUE,
    icd9Mapping = myMap)
)

##      user  system elapsed
## 0.060   0.000   0.059

# much faster the second time because the internal lookup is memoised:
system.time(
  patientChapters <- icd9Comorbid(
    icd9df = patientData,
    isShort = TRUE,
    icd9Mapping = myMap)
)

##      user  system elapsed
## 0.059   0.000   0.059

patientChapters

## visitId Infectious And Parasitic Diseases Neoplasms
## 1 1000 FALSE FALSE
## 2 1001 FALSE FALSE
## 3 1002 FALSE FALSE
## Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders
## 1 TRUE
## 2 FALSE
## 3 FALSE
## Diseases Of The Blood And Blood-Forming Organs
## 1 FALSE
## 2 FALSE
## 3 FALSE

```

10 Putting things together

10.1 Example 1

For the next example, we will find the differences between some categories of the original Elixhauser and the updated version by Quan.

```
in_both <- intersect(elixComorbid$Pulmonary, quanElixComorbid$Pulmonary) # ~1800
only_in_elix <- setdiff(elixComorbid$Pulmonary, quanElixComorbid$Pulmonary) # none
only_in_quanElix <- setdiff(quanElixComorbid$Pulmonary, elixComorbid$Pulmonary) #about 50

in_both %>% icd9GetReal %>% icd9Explain

## [1] "Bronchitis, not specified as acute or chronic"
## [2] "Chronic bronchitis"
## [3] "Emphysema"
## [4] "Asthma"
## [5] "Bronchiectasis"
## [6] "Extrinsic allergic alveolitis"
## [7] "Chronic airway obstruction, not elsewhere classified"
## [8] "Coal workers' pneumoconiosis"
## [9] "Asbestosis"
## [10] "Pneumoconiosis due to other silica or silicates"
## [11] "Pneumoconiosis due to other inorganic dust"
## [12] "Pneumonopathy due to inhalation of other dust"
## [13] "Pneumoconiosis, unspecified"
## [14] "Chronic respiratory conditions due to fumes and vapors"

only_in_quanElix %>% icd9GetReal %>% icd9Explain

## [1] "Asthma"
## [2] "Extrinsic allergic alveolitis"
## [3] "Other chronic pulmonary heart diseases"
## [4] "Chronic pulmonary heart disease, unspecified"
## [5] "Chronic and other pulmonary manifestations due to radiation"
## [6] "Respiratory conditions due to other specified external agents"
```

Passing through `icd9GetReal` stops `icd9Explain` complaining that some of the input codes don't exist. This is because the comorbidity mappings have every possible numerical ICD-9 code, not just the real (official billable) ones.

10.2 Example 2

Suppose we want to exact match only real ICD-9 codes when looking up comorbidities for some patients. E.g. if the coder accidentally omitted a trailing zero, e.g. code 003.20 (*Localized salmonella infection, unspecified*) might have been written as 003.2 which has a heading (*Localized salmonella infections*) but is not itself billable. Use of ICD-9 codes for comorbidities generally assumes the codes are either right or wrong. How do we match only real codes, for a strict interpretation of comorbidities? It's one line or R code:

```
ahrqStrict <- lapply(ahrqComorbid, icd9GetReal)

# first five of the original:
str(ahrqComorbid[1:5])

## List of 5
## $ CHF      : chr [1:120] "39891" "40201" "40211" "40291" ...
```

```
## $ Valvular: chr [1:624] "09320" "09321" "09322" "09323" ...
## $ PHTN : chr [1:130] "41511" "41512" "41513" "41514" ...
## $ PVD : chr [1:585] "4400" "44000" "44001" "44002" ...
## $ HTN : chr [1:27] "4011" "40110" "40111" "40112" ...

# and we see the first five of the updated list is much shorter:
str(ahrqStrict[1:5])

## List of 5
## $ CHF : chr [1:25] "39891" "40201" "40211" "40291" ...
## $ Valvular: chr [1:38] "09320" "09321" "09322" "09323" ...
## $ PHTN : chr [1:10] "41511" "41512" "41513" "41519" ...
## $ PVD : chr [1:53] "4400" "4401" "44020" "44021" ...
## $ HTN : chr [1:7] "4011" "4019" "64200" "64201" ...
```

11 Comorbidities

The comorbidities from different sources are provided as lists. At present only the most recent mapping of ICD-9 codes to comorbidities is provided. See <https://github.com/jackwasey/icd9/issues/6> and <https://github.com/jackwasey/icd9/issues/7>.

This package contains ICD-9-CM to co-morbidity mappings from several sources, based on either the Charlson or Elixhauser lists of co-morbidities. Updated versions of these lists from [AHRQ](<http://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity>) and [Quan et al](<http://www.ncbi.nlm.nih.gov/pubmed/16224307>) are included, along with the original Elixhauser mapping . Since some data is provided in SAS source code format, this package contains functions to parse this SAS source code and generate R data structures. This processing is limited to what is needed for this purpose, although may be generalizable and useful in other contexts. Other lists are transcribed directly from the published articles, but interpretation of SAS code from the publication author is preferred.

A function is provided to compare comorbidity mappings, and show the differences.

```
# compare the first five comorbidities on these mappings
icd9DiffComorbid(ahrqComorbid[1:5], elixComorbid[1:5])

## Comorbidity CHF:
## ahrqComorbid[1:5] has 3 codes not in elixComorbid[1:5]. First few are: 'Mal hypert hrt dis w hf' 'Mal hyp ht/kd I-IV w hf' 'Ma
## Comorbidity Valvular: match.
## Comorbidity PHTN:
## ahrqComorbid[1:5] has 9 codes not in elixComorbid[1:5]. First few are: 'Iatrogen pulm emb/infarc' 'Septic pulmonary embolism' '
## Comorbidity PVD:
## ahrqComorbid[1:5] has 288 codes not in elixComorbid[1:5]. First few are: 'Other aneurysm' 'Septic arterial embolism' 'Dsct of
## Comorbidities only defined in ahrqComorbid[1:5] are: HTN
## Comorbidities only defined in elixComorbid[1:5] are: Arrhythmia
```

11.1 AHRQ comorbidity classification

The AHRQ keeps an updated version of the Elixhauser classification of ICD-9-CM codes into comorbidities, useful for research. They provide the data in the form of SAS code. This package provides just enough code to parse the SAS

source code provided by the AHRQ (but probably not much other SAS code), and generate a list of ICD-9 codes for each comorbidity.

```
ahrqComorbid <- parseAhrqSas()
```

```
head(summary(ahrqComorbid))

##           Length Class      Mode
## CHF          " 120" "-none-" "character"
## Valvular      " 624" "-none-" "character"
## PHTN          " 130" "-none-" "character"
## PVD           " 585" "-none-" "character"
## HTN           "  27" "-none-" "character"
## HTNcx         "  72" "-none-" "character"
```

SAS source code has a strong whiff of the 1970s about it. A fragment of a recent AHRQ SAS comorbidity mapping SAS FORMAT is as follows. Note the mix of character and numeric-only ranges, isolated values, all in 'short' ICD-9 code form.

```
PROC FORMAT LIB=library fmtlib;
VALUE $RCOMFMT
"2780 ",
"27800",
"27801",
"27803",
"64910"-"64914",
"V8530"-"V8539",
"V8541"-"V8545",
"V8554",
"79391"          = "OBESE"      /* Obesity      */

"3004 ",
"30112",
"3090 ",
"3091 ",
"311  "          = "DEPRESS"
```

This is parsed using:

```
parseAhrqSas()
```

resulting in a named list. Here is an extract.

```
ahrqComorbid[c("Obesity", "Depression")]

## $Obesity
## [1] "2780" "27800" "27801" "27802" "27803" "27804" "27805" "27806" "27807" "27808"
## [11] "27809" "64910" "64911" "64912" "64913" "64914" "79391" "V8530" "V8531" "V8532"
## [21] "V8533" "V8534" "V8535" "V8536" "V8537" "V8538" "V8539" "V8541" "V8542" "V8543"
## [31] "V8544" "V8545" "V8554"
##
## $Depression
```

```
## [1] "3004" "30040" "30041" "30042" "30043" "30044" "30045" "30046" "30047" "30048"
## [11] "30049" "30112" "3090" "30900" "30901" "30902" "30903" "30904" "30905" "30906"
## [21] "30907" "30908" "30909" "3091" "30910" "30911" "30912" "30913" "30914" "30915"
## [31] "30916" "30917" "30918" "30919" "311" "3110" "31100" "31101" "31102" "31103"
## [41] "31104" "31105" "31106" "31107" "31108" "31109" "3111" "31110" "31111" "31112"
## [51] "31113" "31114" "31115" "31116" "31117" "31118" "31119" "3112" "31120" "31121"
## [61] "31122" "31123" "31124" "31125" "31126" "31127" "31128" "31129" "3113" "31130"
## [71] "31131" "31132" "31133" "31134" "31135" "31136" "31137" "31138" "31139" "3114"
## [81] "31140" "31141" "31142" "31143" "31144" "31145" "31146" "31147" "31148" "31149"
## [91] "3115" "31150" "31151" "31152" "31153" "31154" "31155" "31156" "31157" "31158"
## [101] "31159" "3116" "31160" "31161" "31162" "31163" "31164" "31165" "31166" "31167"
## [111] "31168" "31169" "3117" "31170" "31171" "31172" "31173" "31174" "31175" "31176"
## [121] "31177" "31178" "31179" "3118" "31180" "31181" "31182" "31183" "31184" "31185"
## [131] "31186" "31187" "31188" "31189" "3119" "31190" "31191" "31192" "31193" "31194"
## [141] "31195" "31196" "31197" "31198" "31199"
```

The `icd9Condense` functions can be used to make a minimal set of parent codes which describes each group:

```
lapply(ahrqComorbid[c("Obesity", "Depression")], icd9CondenseToMajorShort, onlyReal = TRUE)

## Warning in FUN(X[[2L]], ...): only real values requested, but unreal ICD-9 code(s) given.
## Warning in FUN(X[[2L]], ...): only real values requested, but unreal ICD-9 code(s) given.

## $Obesity
## [1] "27800" "27801" "27802" "27803" "64910" "64911" "64912" "64913" "64914" "79391"
## [11] "V8530" "V8531" "V8532" "V8533" "V8534" "V8535" "V8536" "V8537" "V8538" "V8539"
## [21] "V8541" "V8542" "V8543" "V8544" "V8545" "V8554"
##
## $Depression
## [1] "311" "3004" "30112" "3090" "3091"

ahrqComorbid[c("Obesity", "Depression")] %>% icd9ExplainShort(doCondense = FALSE)

## $Obesity
## [1] "Obesity, unspecified"
## [2] "Morbid obesity"
## [3] "Overweight"
## [4] "Obesity hypoventilation syndrome"
## [5] "Obesity complicating pregnancy, childbirth, or the puerperium, unspecified as to episode of care or not applicable"
## [6] "Obesity complicating pregnancy, childbirth, or the puerperium, delivered, with or without mention of antepartum condition"
## [7] "Obesity complicating pregnancy, childbirth, or the puerperium, delivered, with mention of postpartum complication"
## [8] "Obesity complicating pregnancy, childbirth, or the puerperium, antepartum condition or complication"
## [9] "Obesity complicating pregnancy, childbirth, or the puerperium, postpartum condition or complication"
## [10] "Image test inconclusive due to excess body fat"
## [11] "Body Mass Index 30.0-30.9, adult"
## [12] "Body Mass Index 31.0-31.9, adult"
## [13] "Body Mass Index 32.0-32.9, adult"
## [14] "Body Mass Index 33.0-33.9, adult"
## [15] "Body Mass Index 34.0-34.9, adult"
## [16] "Body Mass Index 35.0-35.9, adult"
## [17] "Body Mass Index 36.0-36.9, adult"
## [18] "Body Mass Index 37.0-37.9, adult"
## [19] "Body Mass Index 38.0-38.9, adult"
## [20] "Body Mass Index 39.0-39.9, adult"
## [21] "Body Mass Index 40.0-44.9, adult"
## [22] "Body Mass Index 45.0-49.9, adult"
## [23] "Body Mass Index 50.0-59.9, adult"
## [24] "Body Mass Index 60.0-69.9, adult"
## [25] "Body Mass Index 70 and over, adult"
## [26] "Body Mass Index, pediatric, greater than or equal to 95th percentile for age"
##
## $Depression
## [1] "Depressive disorder, not elsewhere classified"
## [2] "Dysthymic disorder"
## [3] "Chronic depressive personality disorder"
## [4] "Adjustment disorder with depressed mood"
## [5] "Prolonged depressive reaction"
```

12 Elixhauser co-morbidities

Elixhauser originally developed this set of co-morbidities to predict long term mortality based on hospital ICD-9-CM coding records. The AHRQ comorbidities are an updated version of this, however the original Elixhauser have been used in many publications. The ICD-9-CM codes have changed slightly over the years.

```
names(elixComorbid)

## [1] "CHF"          "Arrhythmia"   "Valvular"     "PHTN"         "PVD"
## [6] "HTN"          "HTNcx"       "Paralysis"    "NeuroOther"   "Pulmonary"
## [11] "DM"          "DMcx"        "Hypothyroid"  "Renal"        "Liver"
## [16] "PUD"         "HIV"         "Lymphoma"     "Mets"         "Tumor"
## [21] "Rheumatic"    "Coagulopathy" "Obesity"      "WeightLoss"   "FluidsLytes"
## [26] "BloodLoss"    "Anemia"      "Alcohol"      "Drugs"        "Psychoses"
## [31] "Depression"
```

13 Quan

Quan's paper looked at indices using both ICD-10 and ICD-9-CM. Quan generated updated ICD-9-CM codes for all 30 of Elixhauser and all 17 of Charlson/Deyo's co-morbidities. Thus there are two 'Quan' comorbidity mappings.

```
names(quanDeyoComorbid)

## [1] "MI"          "CHF"          "PVD"          "Stroke"       "Dementia"     "Pulmonary"
## [7] "Rheumatic"   "PUD"          "LiverMild"    "DM"           "DMcx"         "Paralysis"
## [13] "Renal"       "Cancer"       "LiverSevere"  "Mets"         "HIV"

names(quanElixComorbid)

## [1] "CHF"          "Arrhythmia"   "Valvular"     "PHTN"         "PVD"
## [6] "HTN"          "HTNcx"       "Paralysis"    "NeuroOther"   "Pulmonary"
## [11] "DM"          "DMcx"        "Hypothyroid"  "Renal"        "Liver"
## [16] "PUD"         "HIV"         "Lymphoma"     "Mets"         "Tumor"
## [21] "Rheumatic"    "Coagulopathy" "Obesity"      "WeightLoss"   "FluidsLytes"
## [26] "BloodLoss"    "Anemia"      "Alcohol"      "Drugs"        "Psychoses"
## [31] "Depression"
```