

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

Jack O. Wasey

June 23, 2014

1 ICD-9 interpretation and processing

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 27801  Y
## 2    1000  7208  N
## 3    1000 25001  Y
## 4    1001 34400  N
## 5    1001  4011  Y
## 6    1002  4011  N
```

Only the visitId column is propagated to the results. If 'present-on-arrival' is needed, it must be separated out first. The implicit default, therefore, is to ignore it, and give ICD-9 code regardless of POA status.

The comorbidities can be determined as follows:

```
icd9Comorbidities(icd9df = patientData)[, 1:5]
##   visitId   CHF VALVE PULMCIRC PERIVASC
## 1    1000 FALSE FALSE    FALSE    FALSE
## 2    1001 FALSE FALSE    FALSE    FALSE
## 3    1002 FALSE FALSE    FALSE    FALSE
```

The following shows the same code with default options written out:

```
icd9Comorbidities(icd9df = patientData,
                  visitId = "visitId",
                  icd9Field = "icd9",
                  icd9Mapping = ahrqComorbid,
                  validateMapping = FALSE,
                  shortMapping = TRUE)
```

2 validation of ICD-9 codes

```
icd9ValidDecimal("V10.2")
## [1] TRUE
icd9ValidShort(c("099.17", "-1"))
## [1] 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:

```
icd9ValidShort(100) # throws an error
```

3 human-readable ICD-9

```
icd9Explain("001")
## [1] ICD9          Diagnosis  Description
## <0 rows> (or 0-length row.names)
icd9Explain("001.1")
## [1] ICD9          Diagnosis  Description
## <0 rows> (or 0-length row.names)
icd9Explain(list(cholera = c("001", "001.0", "001.1", "001.9")))
## $cholera
## [1] ICD9          Diagnosis  Description
## <0 rows> (or 0-length row.names)
```

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

```
icd9Explain("001.5")
```

```
## [1] ICD9          Diagnosis  Description
## <0 rows> (or 0-length row.names)
```

```
icd9Explain(list(cholera=c("001.0", "001.1", "001.9"))) @
```

4 AHRQ comorbidity classification

The AHRQ makes an annual 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(save = F)
```

```
head(summary(ahrqComorbid))
```

```
##          Length Class      Mode
## CHF          " 120" "-none-" "character"
## VALVE        " 515" "-none-" "character"
## PULMCIRC     " 120" "-none-" "character"
## PERIVASC     " 435" "-none-" "character"
## HTN           "   7" "-none-" "character"
## HTNCX        "  52" "-none-" "character"
```

Here are a couple of the shorter ICD-9 groups listed in ahrqComorbid:

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("OBESE", "DEPRESS")]
## $OBESE
## [1] "2780" "27800" "27801" "27803" "V8554" "79391"
## [7] "64910" "64911" "64912" "64913" "64914" "V8530"
## [13] "V8531" "V8532" "V8533" "V8534" "V8535" "V8536"
## [19] "V8537" "V8538" "V8539" "V8541" "V8542" "V8543"
## [25] "V8544" "V8545"
##
## $DEPRESS
## [1] "3004" "30112" "3090" "3091" "311"

```