

# Package ‘icd9’

July 21, 2014

**Title** Tools for Working with ICD-9 Codes, and Finding Comorbidities

**Description** This package allows accurate manipulation, conversion, validation and comparison of ICD-9-CM (clinical modification) codes. Although ICD-9 codes appear numeric, this is not the case, since trailing zeroes, and a mixture of decimal and non-decimal “short” format codes exist. Tools to map ICD-9 codes to co-morbidities are also provided, with mappings from Quan (Deyo and Elixhauser versions), Elixhauser and AHRQ included.

**Version** 0.5

**Date** 2014-07-14

**Maintainer** Jack O. Wasey <jack@jackwasey.com>

**URL** <https://github.com/jackwasey/icd9>

**Depends** R (>= 3.0.3)

**Imports** memoise

**Suggests** testthat, roxygen2, devtools, knitr, microbenchmark, XML, magrittr

**LazyData** true

**LazyDataCompression** xz

**BugReports** <https://github.com/jackwasey/icd9/issues>

**License** GPL-3

**VignetteBuilder** knitr

**Author** Jack O. Wasey [aut, cre]

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-07-21 20:07:52

**R topics documented:**

icd9-package . . . . .	3
ahrqComorbid . . . . .	4
ahrqComorbidAll . . . . .	4
elixhauserComorbid . . . . .	5
elixhauserComorbidNames . . . . .	5
icd9Benchmark . . . . .	6
icd9Chapters . . . . .	6
icd9ChildrenDecimal . . . . .	7
icd9Comorbidities . . . . .	8
icd9CondenseToExplain . . . . .	9
icd9CondenseToMajor . . . . .	10
icd9DecimalToParts . . . . .	11
icd9DecimalToShort . . . . .	12
icd9ExpandRangeMajor . . . . .	13
icd9ExpandRangeShort . . . . .	14
icd9Explain . . . . .	15
icd9FilterPoa . . . . .	17
icd9GetMajor . . . . .	18
icd9Hierarchy . . . . .	19
icd9InvalidActions . . . . .	20
icd9InvalidDecimal . . . . .	20
icd9PoaChoices . . . . .	21
icd9Real . . . . .	21
icd9ShortToDecimal . . . . .	22
icd9ShortToParts . . . . .	23
icd9SortShort . . . . .	24
icd9Valid . . . . .	24
icd9ValidDecimal . . . . .	25
icd9ValidMajor . . . . .	26
icd9ValidMapping . . . . .	27
icd9ValidShort . . . . .	28
quanDeyoComorbid . . . . .	29
quanElixhauserComorbid . . . . .	29
sasFormatExtract . . . . .	30
saveSourceTreeData . . . . .	30
stopIfInvalidIcd9 . . . . .	31
strMultiMatch . . . . .	31

## Description

icd9 package allows accurate manipulation, validation and comparison of ICD-9-CM (clinical modification) codes. Although ICD-9 codes appear numeric, this is not the case, since trailing zeroes, and a mixture of decimal and non-decimal "short" format codes exist. Tools to map ICD-9 codes to co-morbidities are also provided, with mappings from Quan, Elixhauser and AHRQ included.

\* [icd9Validicd9ValidDecimal](#), [icd9ValidShort](#) and related functions ([icd9ValidDecimalN](#), [icd9ValidDecimalV](#), [icd9ValidDecimalE](#), [icd9ValidShortN](#), [icd9ValidShortV](#)), [icd9ValidShortE](#)) check whether given ICD-9 codes are syntactically valid (although not necessarily genuine ICD-9 codes) \* [icd9Comorbidities](#) determines co-morbidities for a set of patients with one or more ICD-9 codes each.

\* Functions to convert vectors (not lists) of ICD-9 codes include: [icd9DecimalToShort](#), [icd9ShortToDecimal](#). [icd9ShortToParts](#) and [icd9DecimalToParts](#) split a vector of ICD-9 codes into the 'major' part (before decimal point), and 'minor' part (after decimal). The inverse operations are [icd9PartsToShort](#) and [icd9PartsToDecimal](#).

\* Several functions interpret or generate ranges of ICD-9 codes. ICD-9 codes in publications are often specified in ranges, and this requires careful interpretation to avoid including overly broad intermediate codes that might appear in a simple numerical range. See the test suite for some example cases. [icd9ChildrenDecimal](#) and [icd9ChildrenShort](#) generate the full range of child ICD-9 codes. The reverse operation is [icd9CondenseToMajor](#), which takes a mass of codes and finds the smallest set of parent 'major' codes with any of the original codes, which exactly describes the original set. [icd9CondenseToExplain](#)) does the same as [icd9CondenseToMajor](#), but only finds parents which have a human-readable name. Many top-level codes, e.g. "391" are not themselves defined, but have description for their children. [%i9mj%](#) creates a range of icd9 major codes, whereas [%i9s%](#) ([icd9ExpandRangeShort](#)) and [%i9d%](#) ([icd9ExpandRangeDecimal](#)) create correct ranges of short and decimal ICD-9 codes, respectively. The only limitation is that E code ranges are not implemented (not ever seen in the wild by the author).

\* [icd9SortShort](#) sorts in heirarchical, then numerical order, so 100.0 comes before 100.00, for example.

\* AHRQ comorbidity mapping is provided, and a function to read the raw SAS code from AHRQ into R data structures. The pre-processed data is available by lazy-loading in [ahrqComorbid](#). AHRQ releases new mappings annually.

\* Quan revised both Deyo/Charlson and Elixhauser ICD-9 to comorbidity mappings. These are presented as: `link{quanDeyoComorbid}` (which is also derived from the original SAS code used in his publication, referenced in the data documentation), and [quanElixhauserComorbid](#) which was transcribed directly from the same paper.

\* The original Elixhauser mapping is provided, with codes transcribed from the original publication.

\* [parseAhrqSas](#) interprets an SAS format file, returning R data, intended currently only for the FORMAT code provided by AHRQ for comorbidities.

**References**

<http://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp>

**See Also**

rClinicalCodes

---

ahrqComorbid	<i>AHRQ comorbidities</i>
--------------	---------------------------

---

**Description**

This mapping of comorbidities to ICD-9 codes is derived directly from SAS code provided by AHRQ, and translated into this R data structure. This is a revision of the Elixhauser system, notably excluding cardiac arrhythmia.

**Format**

list of character vectors

**Source**

<http://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp>

---

ahrqComorbidAll	<i>AHRQ comorbidities, with HTN, CHF and renal failure subgroups</i>
-----------------	--

---

**Description**

This mapping of comorbidities to ICD-9 codes is derived directly from SAS code provided by AHRQ, and translated into this R data structure. Beyond ahrqComorbid, this includes all the HTN, CHF and renal subgroups, not rolled into their parent categories.

**Format**

list of character vectors, each named by co-morbidity

**Source**

<http://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp>

---

elixhauserComorbid      *Elixhauser comorbidities*

---

**Description**

The original mapping of Elixhauser's ICD-9-CM to 30 comorbidities. According to Sharabiani et al, this mapping provides the best long-term mortality prediction. The weaknesses of this mapping are that it is based on slightly out-dated ICD-9 codes. I have not yet verified what changes to the ICD-9-CM specification between 1998 and now would impact this mapping.

**Format**

list of character vectors, each named by co-morbidity

**References**

Sharabiani, Mansour T. A., Paul Aylin, and Alex Bottle. "Systematic Review of Comorbidity Indices for Administrative Data." *Medical Care* December 2012 50, no. 12 (2012): 1109-18. doi:10.1097/MLR.0b013e31825f64d0. <http://www.ncbi.nlm.nih.gov/pubmed/22929993>

Elixhauser, Anne, Claudia Steiner, D. Robert Harris, and Rosanna M. Coffey. "Comorbidity Measures for Use with Administrative Data." *Medical Care* January 1998 36, no. 1 (1998): 8-27.

---

elixhauserComorbidNames  
*Comorbidity names*

---

**Description**

In the Elixhauser derived mappings, uncomplicated and complicated hypertension are listed separately, but are always combined in the final analyses. Uncomplicated and complicated hypertension are list separately and as "Hypertension, combined." Abbrev suffix indicates a very short space-free description. Quan's version of Elixhauser is identical. AHRQ's update drops the arrythmia field. The Naming convention with neither/either/both suffixes Htn and Abbrev. The Charlson derived mappings do not include hypertension.

**Format**

list, with character/numeric code. HTN numbered 6a and 6b. DM, cancer, mets are counted as in the original paper giving the original 30 groups. "01" to "30"

---

icd9Benchmark	<i>benchmark and profile major functions with larger data sets</i>
---------------	--

---

### Description

icd9 package is intended to be used with large data sets, with millions or rows. Performance of core functions is therefore of some importance, after correctness. R package test code is for correctness, whereas this script stresses the core functions, and looks for bottlenecks.

### Usage

```
icd9Benchmark()
```

---

icd9Chapters	<i>ICD-9-CM chapters</i>
--------------	--------------------------

---

### Description

icd9Chapters, icd9ChaptersSub and icd9ChaptersMajor contain mappings from the higher level descriptions of ICD-9 codes to the ranges of ICD-9 codes they describe. Helpful in summarizing codes or grouping for human-readable output. These can easily be converted to a co-morbidity mapping, as shown in the vignette.

- 001-139 Infectious And Parasitic Diseases
- 140-239 Neoplasms
- 240-279 Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders
- 280-289 Diseases Of The Blood And Blood-Forming Organs
- 290-319 Mental Disorders
- 320-389 Diseases Of The Nervous System And Sense Organs
- 390-459 Diseases Of The Circulatory System
- 460-519 Diseases Of The Respiratory System
- 520-579 Diseases Of The Digestive System
- 580-629 Diseases Of The Genitourinary System
- 630-679 Complications Of Pregnancy, Childbirth, And The Puerperium
- 680-709 Diseases Of The Skin And Subcutaneous Tissue
- 710-739 Diseases Of The Musculoskeletal System And Connective Tissue
- 740-759 Congenital Anomalies
- 760-779 Certain Conditions Originating In The Perinatal Period
- 780-799 Symptoms, Signs, And Ill-Defined Conditions
- 800-999 Injury And Poisoning
- V01-V91 Supplementary Classification Of Factors Influencing Health Status And Contact With Health Services
- E000-E999 Supplementary Classification Of External Causes Of Injury And Poisoning

**Format**

list with chapter/usb-chapter or major names stored in list names, each with two element named character vector with start and end codes.

---

icd9ChildrenDecimal *generate all child codes for given decimal ICD9 codes*

---

**Description**

take ICD9 codes in decimal form and lists of all possible sub-classification codes: e.g. 1.1 returns 1.11, 1.12, 1.13 etc. There are no codes like 1.10 which are distinct from 1.1, so this can be purely numeric. Also, note that expanding "100.0" (100.00 to 100.09) is different from expanding "100.00" (no expansion)  $0.1 == .3/3$  is a problem... [http://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-doesn\\_0027t-R-think-these-numbers-are-equal\\_003f](http://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-doesn_0027t-R-think-these-numbers-are-equal_003f)

**Usage**

```
icd9ChildrenDecimal(icd9Decimal, onlyReal = FALSE,
  invalidAction = icd9InvalidActions)
```

**Arguments**

icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
onlyReal	single logical value, if TRUE, will limit the search to those codes which appear in the master list, not just syntactically valid codes. Since nearly valid, outdated or new codes may be missed, not limiting to 'real' values will be useful. Ultimately, there will need to be annual (and all-time) master lists of codes and the ability to test against a given master list given the year of the ICD-9 coding.
invalidAction	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" - ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code> , then the behavior is the same as silent. - silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code> ). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned - warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s). - stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

**Value**

unsorted vector of ICD9 codes for all subsections of the provided code.

**See Also**

Other ICD-9 ranges: [%i9d%](#), [icd9ExpandRangeDecimal](#); [%i9mj%](#), [icd9ExpandRangeMajor](#); [%i9s%](#), [icd9ExpandRangeShort](#); [icd9Children](#), [icd9ChildrenShort](#); [icd9CondenseToExplain](#); [icd9CondenseToMajor](#); [icd9ExpandMinor](#), [icd9ExpandMinorE](#), [icd9ExpandMinorNV](#); [icd9PrecedingMinors](#), [icd9SubsequentMinors](#)

**Examples**

```
#icd9ChildrenDecimal("100.1")
#icd9ChildrenDecimal("2.34")
```

---

```
icd9Comorbidities      find comorbidities from ICD-9 codes.
```

---

**Description**

This is the main function which extracts co-morbidities from a set of ICD-9 codes. This is when some trivial post-processing of the comorbidity data is done, e.g. renaming to human-friendly field names, and updating fields according to rules. The exact fields from the original mappings can be obtained using `applyHierarchy = FALSE`, but for comorbidity counting, Charlson Score, etc., the rules should be applied.

For Charlson-based comorbidities, strictly speaking, there is no dropping of more e.g. uncomplicated DM if complicated DM exists, however, this is probably useful, in general and is essential when calculating the Charlson score.

**Usage**

```
icd9Comorbidities(icd9df, visitId = "visitId", icd9Field = "icd9", isShort,
  icd9Mapping, validateMapping = FALSE, isShortMapping = TRUE)
```

```
icd9ComorbiditiesAhrq(icd9df, visitId = "visitId", icd9Field = "icd9",
  isShort, validateMapping = FALSE, abbrevNames = TRUE,
  applyHierarchy = TRUE)
```

```
icd9ComorbiditiesQuanDeyo(icd9df, visitId = "visitId", icd9Field = "icd9",
  isShort, validateMapping = FALSE, abbrevNames = TRUE,
  applyHierarchy = TRUE)
```

```
icd9ComorbiditiesQuanElixhauser(icd9df, visitId = "visitId",
  icd9Field = "icd9", isShort, validateMapping = FALSE,
  abbrevNames = TRUE, applyHierarchy = TRUE)
```

```
icd9ComorbiditiesElixhauser(icd9df, visitId = "visitId", icd9Field = "icd9",
  isShort, validateMapping = FALSE, abbrevNames = TRUE,
  applyHierarchy = TRUE)
```



**Arguments**

icd9df	data frame containing columns for visitId (which is the feault name), icd9 (default for the icd9 code), and maybe also a POA flag.
visitId	The name of the column in the data frame which contains the patient or visit identifier. Typically this is the visit identifier, since patients come leave and enter hospital with different ICD-9 codes. It is a character vector of length one. Defaults to "visitId"
icd9Field	The column in the data frame which contains the ICD-9 codes. This is a character vector of length one.
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form.
icd9Mapping	list (or name of a list if character vector of length one is given as argument) of the comorbidities with each top-level list item containing a vector of decimal ICD9 codes. This is in the form of a list, with the names of the items corresponding to the comorbidities (e.g. "HTN", or "diabetes") and the contents of each list item being a character vector of short-form (no decimal place but ideally zero left-padded) ICD-9 codes. No default: user should prefer to use the derivative functions, e.g. icd9ComorbiditiesAhrq, since these also provide appropriate naming for the fields, and squashing the hierarchy (see applyHierarchy below)
validateMapping	logical, whether to validate all the ICD-9 codes in the mapping list. Default is not to check. If validation fails, stop with an error. This is probably worth doing at least once for each mapping used, since there should never be an error in mapping.
isShortMapping	logical, whether the mapping is defined with short ICD-9 codes (TRUE, the default), or decimal if set to FALSE.
abbrevNames	single logical value that defaults to TRUE, in which case the ishorter human-readable names stored in e.g. ahrqComorbidNamesAbbrev are applied to the data frame column names.
applyHierarchy	single logical value that defaults to TRUE, in which case the hierarchy defined for the mapping is applied. E.g. in Elixhauser, you can't have uncomplicated and complicated diabetes both flagged.

---

icd9CondenseToExplain *condense list of short ICD-9 code into minimal set of parent descriptions*

---

**Description**

This can be thought of as the inverse operation to expanding a range. The list given must already contain the parents, because this function will never add a parent ICD-9 which, although may have all children present, may itself have an additional clinical meaning. In addition, in contrast to icd9CondenseToMajor, this function only walks back up to parents which have descriptions in icd9Hierarchy, so it is useful for generating a minimal textual description of a set of ICD-9 codes.

**Usage**

```
icd9CondenseToExplain(icd9Short, invalidAction = c("stop", "ignore", "silent",
"warn"))
```

**Arguments**

`icd9Short` is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

`invalidAction` character string (unit length) containing one of: "ignore", "silent", "warn", "stop"

- ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. `icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")`, then the behavior is the same as `silent`.
- silent invalid ICD-9 codes are replaced silently by NA (actually `NA_character_()`). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned
- warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s).
- stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

**See Also**

Other ICD-9 ranges: [%i9d%](#), [icd9ExpandRangeDecimal](#); [%i9mj%](#), [icd9ExpandRangeMajor](#); [%i9s%](#), [icd9ExpandRangeShort](#); [icd9ChildrenDecimal](#); [icd9Children](#), [icd9ChildrenShort](#); [icd9CondenseToMajor](#); [icd9ExpandMinor](#), [icd9ExpandMinorE](#), [icd9ExpandMinorNV](#); [icd9PrecedingMinors](#), [icd9SubsequentMinors](#)

---

<code>icd9CondenseToMajor</code>	<i>condense list of short ICD-9 code into minimal set of major-part-only codes</i>
----------------------------------	--

---

**Description**

This can be thought of as the inverse operation to `icd9Children("123")`. The list given must already contain the parents, because this function will never add a parent ICD-9 which, although may have all children present, may itself have an additional clinical meaning.

**Usage**

```
icd9CondenseToMajor(icd9Short, onlyReal, dropNonReal = TRUE,
invalidAction = c("stop", "ignore", "silent", "warn"))
```

**Arguments**

icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
onlyReal	single logical value, if TRUE, will limit the search to those codes which appear in the master list, not just syntactically valid codes. Since nearly valid, outdated or new codes may be missed, not limiting to 'real' values will be useful. Ultimately, there will need to be annual (and all-time) master lists of codes and the ability to test against a given master list given the year of the ICD-9 coding.
dropNonReal	single logical, if TRUE, and onlyReal is TRUE, then codes not found in the master list are dropped; otherwise they are included in the output.
invalidAction	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" - ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code> , then the behavior is the same as <code>silent</code> . - silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code> ). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned - warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s). - stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

**See Also**

Other ICD-9 ranges: `%i9d%`, `icd9ExpandRangeDecimal`; `%i9mj%`, `icd9ExpandRangeMajor`; `%i9s%`, `icd9ExpandRangeShort`; `icd9ChildrenDecimal`; `icd9Children`, `icd9ChildrenShort`; `icd9CondenseToExplain`; `icd9ExpandMinor`, `icd9ExpandMinorE`, `icd9ExpandMinorNV`; `icd9PrecedingMinors`, `icd9SubsequentMinors`

---

`icd9DecimalToParts`      *convert decimal-form ICD-9 code to major and minor parts*

---

**Description**

convert decimal-form ICD-9 code to major and minor parts

**Usage**

```
icd9DecimalToParts(icd9Decimal, minorEmpty = "",
  invalidAction = icd9InvalidActions)
```

**Arguments**

icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
minorEmpty	vector of length one, to be used in place of minor part of zero. Defaults to ""
invalidAction	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" - ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code> , then the behavior is the same as <code>silent</code> . - silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code> ). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned - warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s). - stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

---

icd9DecimalToShort     *convert between icd9 decimal and short formats*

---

**Description**

converted decimal ICD9 code, e.g. 123.45 to 'short' e.g. 12345 non-decimal format

**Usage**

```
icd9DecimalToShort(icd9Decimal, invalidAction = icd9InvalidActions)
```

**Arguments**

icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
invalidAction	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" - ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code> , then the behavior is the same as <code>silent</code> . - silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code> ). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned - warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s). - stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

**Value**

character vector of converted ICD-9 codes

**See Also**

Other ICD-9 convert: [icd9DropLeadingZeroes](#), [icd9DropLeadingZeroesDecimal](#), [icd9DropLeadingZeroesMajor](#), [icd9DropLeadingZeroesShort](#); [icd9MajMinToDecimal](#), [icd9MajMinToParts](#), [icd9MajMinToShort](#), [icd9PartsRecompose](#), [icd9PartsToDecimal](#), [icd9PartsToDecimal](#), [icd9PartsToShort](#), [icd9PartsToShort](#); [icd9ShortToDecimal](#)

---

`icd9ExpandRangeMajor` *create range of icd9 major parts*

---

**Description**

accepts V, E or numeric codes. Does not validate codes beyond ensuring that the start and end of the range are of the same type. Will add leading zeroes when appropriate. User can strip them out with [icd9DropLeadingZeroes](#) if they wish.

**Usage**

```
icd9ExpandRangeMajor(start, end, invalidAction = icd9InvalidActions)
```

```
start %i9mj% end
```

**Arguments**

<code>start,end</code>	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
<code>invalidAction</code>	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" - ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code> , then the behavior is the same as <code>silent</code> . - silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code> ). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned - warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s). - stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

**Value**

character vector with range inclusive of start and end

**See Also**

Other ICD-9 ranges: [%i9d%](#), [icd9ExpandRangeDecimal](#); [%i9s%](#), [icd9ExpandRangeShort](#); [icd9ChildrenDecimal](#); [icd9Children](#), [icd9ChildrenShort](#); [icd9CondenseToExplain](#); [icd9CondenseToMajor](#); [icd9ExpandMinor](#), [icd9ExpandMinorE](#), [icd9ExpandMinorNV](#); [icd9PrecedingMinors](#), [icd9SubsequentMinors](#)

---

`icd9ExpandRangeShort` *take two ICD-9 codes and expand range to include all child codes*

---

**Description**

this is cumbersome code, covering a whole load of edge cases relating to the fact that icd9 codes are **not** in numeric order. An alternative strategy would be to list all the ICD9 codes, then a range would just pick out start and finish positions, and return subset of the list. Not all ICD-9 codes are valid, including some parent codes which have valid children. However, I expect at least some of these have been used in some billing databases.

**Usage**

```
icd9ExpandRangeShort(start, end, inferParents = TRUE,
  invalidAction = icd9InvalidActions)
```

```
start %i9s% end
```

**Arguments**

<code>start,end</code>	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
<code>inferParents</code>	single logical value, if TRUE, will infer and include a parent code if a range ends with a value which terminates a higher-level code. E.g. "043" the codes from 0440 to 04499, are covered, but "044" is not explicit. If <code>inferParents</code> is TRUE, "044" would be added, otherwise omitted.
<code>invalidAction</code>	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" - ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code> , then the behavior is the same as silent. - silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code> ). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned - warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s). - stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

**See Also**

Other ICD-9 ranges: [%i9d%](#), [icd9ExpandRangeDecimal](#); [%i9mj%](#), [icd9ExpandRangeMajor](#); [icd9ChildrenDecimal](#); [icd9Children](#), [icd9ChildrenShort](#); [icd9CondenseToExplain](#); [icd9CondenseToMajor](#); [icd9ExpandMinor](#), [icd9ExpandMinorE](#), [icd9ExpandMinorNV](#); [icd9PrecedingMinors](#), [icd9SubsequentMinors](#)

**Examples**

```
"4280 " %i9s% "4289 "
"V80 " %i9s% " V8210 "
```

```
# the following should give all codes in 428 EXCEPT "428",
# and all codes upto 43014 EXCEPT 430 and 4301
icd9ExpandRangeShort("4280 ", "43014")
```

---

icd9Explain	<i>explain ICD9 codes</i>
-------------	---------------------------

---

**Description**

convert full format (123.45 style) ICD9 codes into the name and description for human review there are official ICD9-CM data tables, not with conversion to decimal notation, but to the textual format.

**Usage**

```
icd9Explain(icd9, isShort, doCondense = TRUE)

icd9ExplainShort(icd9Short, doCondense = TRUE)

icd9ExplainDecimal(icd9Decimal, doCondense = TRUE)

## S3 method for class 'list'
icd9Explain(icd9, isShort, doCondense = TRUE)

## S3 method for class 'character'
icd9Explain(icd9, isShort, doCondense = TRUE)

## S3 method for class 'numeric'
icd9Explain(icd9, isShort, doCondense = TRUE)
```

**Arguments**

icd9	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form.
doCondense	single logical value which indicates whether to condense the given set of ICD-9 codes by replacing subsets of codes with 'parent' codes which exactly encompass certain subsets. E.g. If all cholera diagnoses are provided, only '001 - Cholera' needs to be displayed, not all subtypes. This is currently partially implemented. See issue #3 in github.

**Value**

data frame, or list of data frames, with fields for ICD9 code, name and description, derived from datamart lookup table

**Methods (by class)**

- list: explain all ICD-9 codes in a list of vectors
- character: explain character vector of ICD-9 codes
- numeric: explain numeric vector of ICD-9 codes, with warning

**Note**

TODO: it will be useful to have S3 ICD-9 short and long classes for situations like this where we could easily dispatch on short or long type, or even use a `print.icd9decimal` or `print.icd9Short` S3 method to display ICD-9 codes.

**References**

<http://www.stata.com/help.cgi?icd9>

**See Also**

package comorbidities

**Examples**

```
icd9ExplainShort(ahrqComorbid[[1]][1:3])
```



---

icd9FilterPoa	<i>filters data frame based on present-on-arrival flag</i>
---------------	--

---

### Description

this is not a simple binary, since many codes are exempt, unspecified, or unknown. Therefore, two options are given: get all the comorbidities where the POA flag was definitely -ve, coded as "N" or definitely +ve and coded as "Y". Negating one set won't give the other set unless all codes were either Y or N. #describeIn icd9Comorbidities

### Usage

```
icd9FilterPoa(icd9df, poaField = "poa", poa = icd9PoaChoices)

icd9FilterPoaYes(icd9df, poaField = "poa")

icd9FilterPoaNo(icd9df, poaField = "poa")

icd9FilterPoaNotNo(icd9df, poaField = "poa")

icd9FilterPoaNotYes(icd9df, poaField = "poa")
```

### Arguments

icd9df	data frame containing columns for visitId (which is the feault name), icd9 (default for the icd9 code), and maybe also a POA flag.
poaField	The name of column in the data frame which contains the Present On Arrival flag. The flag itself is a single character, typically one of "Y", "N", "E", "X", "U" or empty. The poaField is a character vector of length one.
poa	single character value, being one of poaChoices whether to account for comorbidities flagged as present-on-arrival. This is not a simple binary, since many codes are exempt, unspecified, or unknown. poaField gives the choices: yes, not no, no, not yes. The intermediate codes, such as "exempt", "unknown" and NA mean that "yes" is not the same as "not no."

### Examples

```
## Not run:
# using magrittr is beautiful:
library("magrittr", quietly = TRUE, warn.conflicts = FALSE)
myData <- data.frame(
  visitId = c("v1", "v2", "v3", "v4"),
  icd9 = c("39891", "39790", "41791", "4401"),
  poa = c("Y", "N", NA, "Y"),
  stringsAsFactors = FALSE
)
myData %>% icd9FilterPoaNotNo() %>% icd9ComorbiditiesAhrq(isShort = TRUE)
```

```
# can fill out named fields also:
myData %>% icd9FilterPoaYes(poaField="poa") %>%
  icd9ComorbiditiesAhrq(icd9Field = "icd9", visitId = "visitId")
# can call the core icd9Comorbidities function with an arbitrary mapping
myData %>%
  icd9FilterPoaYes() %>%
  icd9Comorbidities(icd9Field = "icd9", visitId = "visitId",
    icd9Mapping = quanElixhauserComorbid,
    validateMapping = TRUE,
    isShortMapping = TRUE)

## End(Not run)
```

---

 icd9GetMajor

*extract major part from short or decimal ICD-9 code*


---

## Description

Simply extracts parts, then returns only the major part in a character vector

## Usage

```
icd9GetMajor(icd9, isShort, invalidAction = icd9InvalidActions)
```

```
icd9DecimalToMajor(icd9Decimal, invalidAction = icd9InvalidActions)
```

```
icd9ShortToMajor(icd9Short, invalidAction = icd9InvalidActions)
```

## Arguments

icd9	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form.
invalidAction	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" - ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code> , then the behavior is the same as <code>silent</code> . - silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code> ). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned - warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s). - stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

**Value**

character vector

---

icd9Hierarchy	<i>ICD9-CM diagnosis code lookup</i>
---------------	--------------------------------------

---

**Description**

short-form ICD-9 codes with short and long descriptions, and description of each hierarchy level containing each code.

**Format**

data frame

**Source**

[http://wonder.cdc.gov/wonder/sci\\_data/codes/icd9/type\\_txt/icd9cm.asp](http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9cm.asp)

Rich text descriptions here: <http://www.cdc.gov/nchs/icd/icd9cm.htm> <http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html> This page has versions 23 to 32 (2005 to 2014). At present, only the 2014 data is included in this package.

<http://wonder.cdc.gov/wonder/help/icd.html>

[http://wonder.cdc.gov/wonder/sci\\_data/codes/icd9/type\\_txt/icd9abb.asp](http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9abb.asp)

[http://wonder.cdc.gov/wonder/sci\\_data/codes/icd9/type\\_txt/icd9cm.asp](http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9cm.asp)

[http://wonder.cdc.gov/wonder/sci\\_data/codes/icd9/type\\_txt/icdcm.asp](http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icdcm.asp)

[http://wonder.cdc.gov/wonder/sci\\_data/codes/icd9/type\\_txt/icd9abb.asp](http://wonder.cdc.gov/wonder/sci_data/codes/icd9/type_txt/icd9abb.asp)

---

icd9InvalidActions     *invalid actions, default is first item*

---

### Description

Some functions accept `invalidAction` argument, but require non-default validation, but the vast majority will honor the default of 'ignore.' Ignore may lead to downstream errors, since bad data may arrive in internal functions, however, this is the fastest option. A typical use-case of this package would involve validation and cleaning steps (with validation), followed by repeated analysis (as fast as possible, with minimal validation).

### Usage

```
icd9InvalidActions
```

### Format

```
chr [1:4] "ignore" "silent" "warn" "stop"
```

---

icd9InvalidDecimal     *invalid subset of decimal or short ICD-9 codes*

---

### Description

given vector of short or decimal ICD-9 codes in `icd9Decimal` or `icd9Short`, return (in the same format) those codes which are invalid. Useful for generating error messages with the faulty codes if validation fails.

### Usage

```
icd9InvalidDecimal(icd9Decimal)
```

```
icd9InvalidShort(icd9Short)
```

### Arguments

`icd9Decimal`     character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

`icd9Short`     is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

---

icd9PoaChoices	<i>present-on-admission flags</i>
----------------	-----------------------------------

---

### Description

Present-on-admission (POA) is not simply true or false. It can be one of a number of indeterminate values, including NA, or "Y" or "N". "Present-on-arrival" in this context will mean a positive "Y" flag and nothing else. Other interpretations are to include all ICD-9 codes not flagged 'N': but this would include many unknowns. Conversely, when looking for definite new diagnoses, we should only find 'N' flagged codes, and ignore anything marked "Y" or indeterminate. This gives four options: `poa == "Y"` , `poa == "N"` , `poa != "N"` , `poa != "Y"`.

### Usage

```
icd9PoaChoices
```

### Format

```
chr [1:4] "yes" "no" "notYes" "notNo"
```

---

icd9Real	<i>Does ICD-9 code exist</i>
----------	------------------------------

---

### Description

This is different from syntactic validity: it looks it up in the list of icd9 codes. This may have been easier all along, but checking syntactic validity still very useful, with a changing list of icd-9 codes over time, and possibly imperfections in the master list derived from CMS.

### Usage

```
icd9Real(icd9, isShort, invalidAction = icd9InvalidActions)
```

```
icd9RealShort(icd9Short, invalidAction = icd9InvalidActions)
```

```
icd9RealDecimal(icd9Decimal, invalidAction = icd9InvalidActions)
```

### Arguments

icd9	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form.

invalidAction	<p>character string (unit length) containing one of: "ignore", "silent", "warn", "stop"</p> <ul style="list-style-type: none"> <li>- ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code>, then the behavior is the same as <code>silent</code>.</li> <li>- silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code>). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned</li> <li>- warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s).</li> <li>- stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.</li> </ul>
icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
icd9Decimal	character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

**Value**

logical vector

---

`icd9ShortToDecimal`     *convert short-form ICD-9 code to decimal form*

---

**Description**

converts ICD-9 'short' form to decimal form

**Usage**

```
icd9ShortToDecimal(icd9Short, invalidAction = icd9InvalidActions)
```

**Arguments**

icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
invalidAction	<p>character string (unit length) containing one of: "ignore", "silent", "warn", "stop"</p> <ul style="list-style-type: none"> <li>- ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code>, then the behavior is the same as <code>silent</code>.</li> <li>- silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code>). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned</li> </ul>

- warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s).
- stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.

### See Also

Other ICD-9 convert: [icd9DecimalToShort](#); [icd9DropLeadingZeroes](#), [icd9DropLeadingZeroesDecimal](#), [icd9DropLeadingZeroesMajor](#), [icd9DropLeadingZeroesShort](#); [icd9MajMinToDecimal](#), [icd9MajMinToParts](#), [icd9MajMinToShort](#), [icd9PartsRecompose](#), [icd9PartsToDecimal](#), [icd9PartsToDecimal](#), [icd9PartsToShort](#), [icd9PartsToShort](#)

---

<code>icd9ShortToParts</code>	<i>extract major and minor parts of a decimal ICD-9 code</i>
-------------------------------	--

---

### Description

accepts Vxxxx Exxxx or xxxxx

### Usage

```
icd9ShortToParts(icd9Short, minorEmpty = "",
  invalidAction = icd9InvalidActions)
```

### Arguments

<code>icd9Short</code>	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
<code>minorEmpty</code>	vector of length one, to be used in place of minor part of zero. Defaults to ""
<code>invalidAction</code>	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" <ul style="list-style-type: none"> <li>- ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code>, then the behavior is the same as <code>silent</code>.</li> <li>- silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code>). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned</li> <li>- warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s).</li> <li>- stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.</li> </ul>

### Value

data.frame with two columns. At least the minor part must be character, because "03" is different to "3", but "30" is the same as "3"

---

icd9SortShort	<i>sort short-form icd9 codes</i>
---------------	-----------------------------------

---

### Description

should work with numeric only, V or E codes. Note that a numeric sort does not work for ICD-9 codes, since "162" > "1620" TODO: write tests. TODO: reply with the actual items given (not trimmed, etc.) TODO: need to be able to compare a pair of codes quickly, then use built-in sort. This becomes easier when I move to S3 classes for ICD-9.

### Usage

```
icd9SortShort(icd9Short, invalidAction = icd9InvalidActions)
```

### Arguments

icd9Short	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
invalidAction	character string (unit length) containing one of: "ignore", "silent", "warn", "stop" <ul style="list-style-type: none"> <li>- ignore no check for validity of ICD-9 code is made implicitly. This is the fastest option. If the function call is explicitly to check validity, e.g. <code>icd9ValidShort(icd9Short = "12345", invalidAction = "ignore")</code>, then the behavior is the same as <code>silent</code>.</li> <li>- silent invalid ICD-9 codes are replaced silently by NA (actually <code>NA_character_()</code>). If the function returns TRUE or FALSE rather than the ICD-9 code itself, then FALSE is returned</li> <li>- warn invalid ICD-9 codes are replaced by NA with a warning revealing the invalid code(s).</li> <li>- stop stop with error if any invalid codes are found in input. This is the preferred method when doing one-time validation, e.g. of the co-morbidity mapping lists.</li> </ul>

### Value

sorted vector of ICD-9 codes

---

icd9Valid	<i>check whether any ICD-9 code is syntactically valid</i>
-----------	--

---

### Description

check whether any ICD-9 code is syntactically valid



**Usage**

```
icd9Valid(icd9, isShort)
```

**Arguments**

`icd9` is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

`isShort` single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form.

**See Also**

[icd9ValidDecimal](#) and [icd9ValidShort](#)

Other ICD9 validation: [icd9GetInvalidMappingDecimal](#), [icd9GetInvalidMappingShort](#), [icd9ValidMapping](#), [icd9ValidMappingDecimal](#), [icd9ValidMappingShort](#); [icd9ValidDecimal](#), [icd9ValidDecimalE](#), [icd9ValidDecimalN](#), [icd9ValidDecimalV](#); [icd9ValidMajor](#), [icd9ValidMajorE](#), [icd9ValidMajorN](#), [icd9ValidMajorV](#); [icd9ValidNaWarnStop](#), [icd9ValidNaWarnStopDecimal](#), [icd9ValidNaWarnStopMajor](#), [icd9ValidNaWarnStopShort](#); [icd9ValidShort](#), [icd9ValidShortE](#), [icd9ValidShortN](#), [icd9ValidShortV](#); [stopIfInvalidIcd9](#), [warnIfInvalidIcd9](#)

---

<code>icd9ValidDecimal</code>	<i>check whether decimal icd9 codes are valid</i>
-------------------------------	---

---

**Description**

Check validity of 'long' (i.e. decimal form) ICD9 codes. The codes may be numeric disease descriptors or V or E prefixed.

**Usage**

```
icd9ValidDecimal(icd9Decimal)
```

```
icd9ValidDecimalV(icd9Decimal)
```

```
icd9ValidDecimalE(icd9Decimal)
```

```
icd9ValidDecimalN(icd9Decimal)
```

**Arguments**

`icd9Decimal` character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

**Details**

Long form is not ambiguous so additional zeroes are tolerated. Although integer ICD-9 codes could be correct, there is a difference between 100 and 100.0, and indeed 100.00. Therefore, character class is enforced to avoid this problem.

**Value**

logical vector with T or F for each icd9 code provided according to validity

**Note**

TODO: icd9ValidDecimalN not quite right, since it would validate 0.12

**See Also**

<http://www.stata.com/users/wgould/icd9/icd9.hlp> [urlhttp://www.sascommunity.org/wiki/Validate\\_the\\_format\\_of\\_ICD9\\_codes](http://www.sascommunity.org/wiki/Validate_the_format_of_ICD9_codes)

Other ICD9 validation: [icd9GetInvalidMappingDecimal](#), [icd9GetInvalidMappingShort](#), [icd9ValidMapping](#), [icd9ValidMappingDecimal](#), [icd9ValidMappingShort](#); [icd9ValidMajor](#), [icd9ValidMajorE](#), [icd9ValidMajorN](#), [icd9ValidMajorV](#); [icd9ValidNaWarnStop](#), [icd9ValidNaWarnStopDecimal](#), [icd9ValidNaWarnStopMajor](#), [icd9ValidNaWarnStopShort](#); [icd9ValidShort](#), [icd9ValidShortE](#), [icd9ValidShortN](#), [icd9ValidShortV](#); [icd9Valid](#); [stopIfInvalidIcd9](#), [warnIfInvalidIcd9](#)

---

icd9ValidMajor	<i>validate a major part</i>
----------------	------------------------------

---

**Description**

validation for just the 'major' part of an ICD-9 code. This can in fact be provided as a numeric, since there is no ambiguity. Numeric-only codes should be one to three digits, V codes are followed by one or two digits, and E codes always by three digits between 800 and 999.

**Usage**

`icd9ValidMajor(major)`

`icd9ValidMajorN(major)`

`icd9ValidMajorV(major)`

`icd9ValidMajorE(major)`

**Arguments**

major	character vector of 'major' part of ICD-9 codes, i.e. that part which falls before the decimal point, in decimal notation. (In 5 digit notation, the 'major' part is the first three characters (with leading zeroes), and includes V or E prefix. xyz
-------	--

**See Also**

Other ICD9 validation: [icd9GetInvalidMappingDecimal](#), [icd9GetInvalidMappingShort](#), [icd9ValidMapping](#), [icd9ValidMappingDecimal](#), [icd9ValidMappingShort](#); [icd9ValidDecimal](#), [icd9ValidDecimalE](#), [icd9ValidDecimalN](#), [icd9ValidDecimalV](#); [icd9ValidNaWarnStop](#), [icd9ValidNaWarnStopDecimal](#), [icd9ValidNaWarnStopMajor](#), [icd9ValidNaWarnStopShort](#); [icd9ValidShort](#), [icd9ValidShortE](#), [icd9ValidShortN](#), [icd9ValidShortV](#); [icd9Valid](#); [stopIfInvalidIcd9](#), [warnIfInvalidIcd9](#)

---

icd9ValidMapping	<i>validate an icd9 mapping to comorbidities</i>
------------------	--

---

**Description**

just takes each item in each vector of the list of vectors and checks validity

**Usage**

```
icd9ValidMapping(icd9Mapping, isShort)
```

```
icd9ValidMappingDecimal(icd9Mapping)
```

```
icd9ValidMappingShort(icd9Mapping)
```

```
icd9GetInvalidMappingShort(icd9Mapping)
```

```
icd9GetInvalidMappingDecimal(icd9Mapping)
```

**Arguments**

**icd9Mapping**      named list containing vectors of icd9 codes. E.g. the AHRQ comorbidities, contains `list(OBESE = c("2780", "27800", "27801", "27803", "V8554", "79391", "64910", "6491"), DEPRESS = c("3004", "30112", "3090", "3091", "311"))` amongst other longer groups.

**isShort**          single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form.

**See Also**

Other ICD9 validation: [icd9ValidDecimal](#), [icd9ValidDecimalE](#), [icd9ValidDecimalN](#), [icd9ValidDecimalV](#); [icd9ValidMajor](#), [icd9ValidMajorE](#), [icd9ValidMajorN](#), [icd9ValidMajorV](#); [icd9ValidNaWarnStop](#), [icd9ValidNaWarnStopDecimal](#), [icd9ValidNaWarnStopMajor](#), [icd9ValidNaWarnStopShort](#); [icd9ValidShort](#), [icd9ValidShortE](#), [icd9ValidShortN](#), [icd9ValidShortV](#); [icd9Valid](#); [stopIfInvalidIcd9](#), [warnIfInvalidIcd9](#)

---

icd9ValidShort	<i>validate ICD-9 short form code</i>
----------------	---------------------------------------

---

### Description

As in `icd9ValidDecimal`, character type of the input is enforced. An integer 100 could correspond to decimal ICD-9 codes 1.00 10.0 or 100.

### Usage

```
icd9ValidShort(icd9Short)
```

```
icd9ValidShortV(icd9Short)
```

```
icd9ValidShortE(icd9Short)
```

```
icd9ValidShortN(icd9Short)
```

### Arguments

`icd9Short` is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"

### Value

logical vector with T or F for each icd9 code provided according to validity

### See Also

<http://www.stata.com/users/wgould/icd9/icd9.hlp> [http://www.sascommunity.org/wiki/Validate\\_the\\_format\\_of\\_ICD-9\\_codes](http://www.sascommunity.org/wiki/Validate_the_format_of_ICD-9_codes)

Other ICD9 validation: `icd9GetInvalidMappingDecimal`, `icd9GetInvalidMappingShort`, `icd9ValidMapping`, `icd9ValidMappingDecimal`, `icd9ValidMappingShort`; `icd9ValidDecimal`, `icd9ValidDecimalE`, `icd9ValidDecimalN`, `icd9ValidDecimalV`; `icd9ValidMajor`, `icd9ValidMajorE`, `icd9ValidMajorN`, `icd9ValidMajorV`; `icd9ValidNaWarnStop`, `icd9ValidNaWarnStopDecimal`, `icd9ValidNaWarnStopMajor`, `icd9ValidNaWarnStopShort`; `icd9Valid`; `stopIfInvalidIcd9`, `warnIfInvalidIcd9`

---

quanDeyoComorbid

*Quan adaptation of Deyo/Charlson comorbidities*

---

**Description**

Derived programmatically from the SAS code used in the original publication. According to the referenced study, this provides the best predictor of in-patient to <30d mortality. Of note, Deyo drops the distinction between leukemia, lymphoma and non-metastatic cancer. As far as I have looked into this, in the rare cases where someone had two or three of leukemia, lymphoma and non-metastatic cancer, the Quan adaptation would give a lower Charlson score than the original scheme. The Deyo original Charlson to ICD-9-CM groups does include distinct categories for these things.

**Format**

list of character vectors, each named by co-morbidity

**References**

Quan, Hude, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L. Duncan Saunders, Cynthia A. Beck, Thomas E. Feasby, and William A. Ghali. "Coding Algorithms for Defining Comorbidities in ICD-9-CM and ICD-10 Administrative Data." *Medical Care* 43, no. 11 (November 1, 2005): 1130-39. <http://www.ncbi.nlm.nih.gov/pubmed/16224307> unavailable: <http://web.archive.org/web/20110225042437/http://www.chaps.ucalgary.ca/sas>

---

quanElixhauserComorbid

*Quan adaptation of Elixhauser comorbidities*

---

**Description**

These were transcribed directly from the Quan paper referenced.

**Format**

list of character vectors, each named by co-morbidity

**References**

Quan, Hude, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L. Duncan Saunders, Cynthia A. Beck, Thomas E. Feasby, and William A. Ghali. "Coding Algorithms for Defining Comorbidities in ICD-9-CM and ICD-10 Administrative Data." *Medical Care* 43, no. 11 (November 1, 2005): 1130-39. <http://www.ncbi.nlm.nih.gov/pubmed/16224307> unavailable: <http://web.archive.org/web/20110225042437/http://www.chaps.ucalgary.ca/sas>

---

sasFormatExtract	<i>extract assignments from a SAS FORMAT definition.</i>
------------------	--

---

**Description**

this is modelled entirely on a single chunk of SAS code, but hopefully will have some generalizability. It relies heavily on lists and regex, but, as you will see from the code, R is not a great language with which to write a SAS parser.

```
#example #sasFormatExtract(readLines('inst/extdata//comformat2012-2013.txt'))
```

**Usage**

```
sasFormatExtract(sasTxt)
```

**Arguments**

sasTxt is a character vector, with one item per line, e.g. from readLines

**Value**

list (of lists)

**References**

<http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#a002473474.htm>  
<https://communities.sas.com/thread/47571?start=0&tstart=0>  
<https://communities.sas.com/message/165945>

---

saveSourceTreeData	<i>save data in source tree</i>
--------------------	---------------------------------

---

**Description**

attempt to write the data from the source file to RData in the package source tree.

**Usage**

```
saveSourceTreeData(varName, path = "~/icd9/data")
```

**Arguments**

varName is the variable name and the part of the filename which will be saved, followed by ".RData"

path is a path name to destination folder for the data: no trailing slash.

---

stopIfInvalidIcd9      *warn or stop with invalid ICD-9 codes*

---

### Description

In the case of warning, execution continues, and the logical vector containing the *\*invalid\** codes is returned invisibly. Returning the invalids for consistency with the name of the function. The warn version of this function only uses one call to check validity, making execution as fast as possible.

### Usage

```
stopIfInvalidIcd9(icd9, isShort)
```

```
warnIfInvalidIcd9(icd9, isShort)
```

### Arguments

icd9	is a character vector of ICD-9 codes. If fewer than five characters is given in a code, then the digits are greedily assigned to hundreds, then tens, then units, before the decimal parts. E.g. "10" becomes "010", not "0010"
isShort	single logical value which determines whether the ICD-9 code provided is in short (TRUE) or decimal (FALSE) form.

### See Also

Other ICD9 validation: [icd9GetInvalidMappingDecimal](#), [icd9GetInvalidMappingShort](#), [icd9ValidMapping](#), [icd9ValidMappingDecimal](#), [icd9ValidMappingShort](#); [icd9ValidDecimal](#), [icd9ValidDecimalE](#), [icd9ValidDecimalN](#), [icd9ValidDecimalV](#); [icd9ValidMajor](#), [icd9ValidMajorE](#), [icd9ValidMajorN](#), [icd9ValidMajorV](#); [icd9ValidNaWarnStop](#), [icd9ValidNaWarnStopDecimal](#), [icd9ValidNaWarnStopMajor](#), [icd9ValidNaWarnStopShort](#); [icd9ValidShort](#), [icd9ValidShortE](#), [icd9ValidShortN](#), [icd9ValidShortV](#); [icd9Valid](#)

---

strMultiMatch      *return the actual matches from a bracketed regex*

---

### Description

Be careful: this may throw funny results for exotic regex, but so far, it seems okay. it also drops the first result which always seems to be a duplicate or whole-string match

strPairMatch differs in that there should only be two pairs of parenthesis, then the first (by default) becomes the name, and the second the value.

**Usage**

```
strMultiMatch(pattern, text, dropEmpty = FALSE, ...)
```

```
strPairMatch(pattern, text, swap = FALSE, dropEmpty = FALSE, ...)
```

**Arguments**

pattern	regular expression: if it has bracketed sections, these submatches are returned
text	is the string to match against. This vector should be the same length as the pattern vector, or the pattern vector should be length one.
...	are additional parameters passed to <code>regexec</code> and <code>regmatches</code> . I haven't tried this: it may need two separate variables containing lists of params, since this will send everything to both functions.
dropEmpty	logical whether to drop rows with no matches
swap	logical scalar, whether to swap the names and values. Default is not to swap, so the first match becomes the name.

**Value**

list of character vectors, list length being the length of the input text vector.



# Index

- \*Topic **category**
  - icd9Chapters, 6
- \*Topic **character**
  - icd9InvalidActions, 20
  - icd9PoaChoices, 21
- \*Topic **datasets**
  - ahrqComorbid, 4
  - ahrqComorbidAll, 4
  - elixhauserComorbid, 5
  - elixhauserComorbidNames, 5
  - icd9Chapters, 6
  - icd9Hierarchy, 19
  - quanDeyoComorbid, 29
  - quanElixhauserComorbid, 29
- \*Topic **error**
  - stopIfInvalidIcd9, 31
- \*Topic **list**
  - icd9Chapters, 6
  - sasFormatExtract, 30
- \*Topic **manip**
  - icd9ChildrenDecimal, 7
  - icd9CondenseToExplain, 9
  - icd9DecimalToParts, 11
  - icd9DecimalToShort, 12
  - icd9InvalidDecimal, 20
  - icd9ShortToDecimal, 22
  - icd9ShortToParts, 23
  - icd9SortShort, 24
- \*Topic **misc**
  - icd9-package, 3
- \*Topic **programming**
  - sasFormatExtract, 30
- \*Topic **utilities**
  - icd9-package, 3
- %i9mj% (icd9ExpandRangeMajor), 13
- %i9s% (icd9ExpandRangeShort), 14
- %i9d%, 3, 8, 10, 11, 14, 15
- %i9mj%, 3, 8, 10, 11, 15
- %i9s%, 3, 8, 10, 11, 14
- ahrqComorbid, 3, 4
- ahrqComorbidAll, 4
- ahrqComorbidNames
  - (elixhauserComorbidNames), 5
- ahrqComorbidNamesAbbrev
  - (elixhauserComorbidNames), 5
- ahrqComorbidNamesHtn
  - (elixhauserComorbidNames), 5
- ahrqComorbidNamesHtnAbbrev
  - (elixhauserComorbidNames), 5
- charlsonComorbidNames
  - (elixhauserComorbidNames), 5
- charlsonComorbidNamesAbbrev
  - (elixhauserComorbidNames), 5
- elixhauserComorbid, 5
- elixhauserComorbidNames, 5
- elixhauserComorbidNamesAbbrev
  - (elixhauserComorbidNames), 5
- elixhauserComorbidNamesHtn
  - (elixhauserComorbidNames), 5
- elixhauserComorbidNamesHtnAbbrev
  - (elixhauserComorbidNames), 5
- icd9 (icd9-package), 3
- icd9-package, 3
- icd9Benchmark, 6
- icd9Chapters, 6
- icd9ChaptersMajor (icd9Chapters), 6
- icd9ChaptersSub (icd9Chapters), 6
- icd9Children, 8, 10, 11, 14, 15
- icd9ChildrenDecimal, 3, 7, 10, 11, 14, 15
- icd9ChildrenShort, 3, 8, 10, 11, 14, 15
- icd9Comorbidities, 3, 8
- icd9ComorbiditiesAhrq
  - (icd9Comorbidities), 8
- icd9ComorbiditiesElixhauser
  - (icd9Comorbidities), 8

- icd9ComorbiditiesQuanDeyo  
    (icd9Comorbidities), 8
- icd9ComorbiditiesQuanElixhauser  
    (icd9Comorbidities), 8
- icd9CondenseToExplain, 3, 8, 9, 11, 14, 15
- icd9CondenseToMajor, 3, 8, 10, 10, 14, 15
- icd9DecimalToMajor (icd9GetMajor), 18
- icd9DecimalToParts, 3, 11
- icd9DecimalToShort, 3, 12, 23
- icd9DropLeadingZeroes, 13, 23
- icd9DropLeadingZeroesDecimal, 13, 23
- icd9DropLeadingZeroesMajor, 13, 23
- icd9DropLeadingZeroesShort, 13, 23
- icd9ExpandMinor, 8, 10, 11, 14, 15
- icd9ExpandMinorE, 8, 10, 11, 14, 15
- icd9ExpandMinorNV, 8, 10, 11, 14, 15
- icd9ExpandRangeDecimal, 3, 8, 10, 11, 14, 15
- icd9ExpandRangeMajor, 8, 10, 11, 13, 15
- icd9ExpandRangeShort, 3, 8, 10, 11, 14, 14
- icd9Explain, 15
- icd9ExplainDecimal (icd9Explain), 15
- icd9ExplainShort (icd9Explain), 15
- icd9FilterPoa, 17
- icd9FilterPoaNo (icd9FilterPoa), 17
- icd9FilterPoaNotNo (icd9FilterPoa), 17
- icd9FilterPoaNotYes (icd9FilterPoa), 17
- icd9FilterPoaYes (icd9FilterPoa), 17
- icd9GetInvalidMappingDecimal, 25–28, 31
- icd9GetInvalidMappingDecimal  
    (icd9ValidMapping), 27
- icd9GetInvalidMappingShort, 25–28, 31
- icd9GetInvalidMappingShort  
    (icd9ValidMapping), 27
- icd9GetMajor, 18
- icd9Hierarchy, 19
- icd9InvalidActions, 20
- icd9InvalidDecimal, 20
- icd9InvalidShort (icd9InvalidDecimal),  
    20
- icd9MajMinToDecimal, 13, 23
- icd9MajMinToParts, 13, 23
- icd9MajMinToShort, 13, 23
- icd9PartsRecompose, 13, 23
- icd9PartsToDecimal, 3, 13, 23
- icd9PartsToShort, 3, 13, 23
- icd9PoaChoices, 21
- icd9PrecedingMinors, 8, 10, 11, 14, 15
- icd9Real, 21
- icd9RealDecimal (icd9Real), 21
- icd9RealShort (icd9Real), 21
- icd9ShortToDecimal, 3, 13, 22
- icd9ShortToMajor (icd9GetMajor), 18
- icd9ShortToParts, 3, 23
- icd9SortShort, 3, 24
- icd9SubsequentMinors, 8, 10, 11, 14, 15
- icd9Valid, 3, 24, 26–28, 31
- icd9ValidDecimal, 3, 25, 25, 27, 28, 31
- icd9ValidDecimalE, 3, 25, 27, 28, 31
- icd9ValidDecimalE (icd9ValidDecimal), 25
- icd9ValidDecimalN, 3, 25, 27, 28, 31
- icd9ValidDecimalN (icd9ValidDecimal), 25
- icd9ValidDecimalV, 3, 25, 27, 28, 31
- icd9ValidDecimalV (icd9ValidDecimal), 25
- icd9ValidMajor, 25, 26, 26, 27, 28, 31
- icd9ValidMajorE, 25–28, 31
- icd9ValidMajorE (icd9ValidMajor), 26
- icd9ValidMajorN, 25–28, 31
- icd9ValidMajorN (icd9ValidMajor), 26
- icd9ValidMajorV, 25–28, 31
- icd9ValidMajorV (icd9ValidMajor), 26
- icd9ValidMapping, 25–27, 27, 28, 31
- icd9ValidMappingDecimal, 25–28, 31
- icd9ValidMappingDecimal  
    (icd9ValidMapping), 27
- icd9ValidMappingShort, 25–28, 31
- icd9ValidMappingShort  
    (icd9ValidMapping), 27
- icd9ValidNaWarnStop, 25–28, 31
- icd9ValidNaWarnStopDecimal, 25–28, 31
- icd9ValidNaWarnStopMajor, 25–28, 31
- icd9ValidNaWarnStopShort, 25–28, 31
- icd9ValidShort, 3, 25–27, 28, 31
- icd9ValidShortE, 3, 25–27, 31
- icd9ValidShortE (icd9ValidShort), 28
- icd9ValidShortN, 3, 25–27, 31
- icd9ValidShortN (icd9ValidShort), 28
- icd9ValidShortV, 3, 25–27, 31
- icd9ValidShortV (icd9ValidShort), 28
  
- package-icd9 (icd9-package), 3
- parseAhrqSas, 3
  
- quanDeyoComorbid, 29
- quanElixhauserComorbid, 3, 29
- quanElixhauserComorbidNames  
    (elixhauserComorbidNames), 5

quanElixhauserComorbidNamesAbbrev  
    (elixhauserComorbidNames), 5  
quanElixhauserComorbidNamesHtn  
    (elixhauserComorbidNames), 5  
quanElixhauserComorbidNamesHtnAbbrev  
    (elixhauserComorbidNames), 5  
  
sasFormatExtract, 30  
saveSourceTreeData, 30  
stopIfInvalidIcd9, 25–28, 31  
strMultiMatch, 31  
strPairMatch (strMultiMatch), 31  
  
warnIfInvalidIcd9, 25–28  
warnIfInvalidIcd9 (stopIfInvalidIcd9),  
    31