multimorbidity R Package

Wyatt P. Bensken
PhD Candidate, Epidemiology and Biostatistics
Case Western Reserve University
wyattbensken.com
wpb27 [at] case.edu

Agenda

- Overview of the Package
- Installing the Package
- Included Data
- Included Data Cleaning Functions
- Included Indices
- The package can be found at:
  - https://CRAN.R-project.org/package=multimorbidity

Overview of the Package

Why Do We Need The `multimorbidity` package?

- Identifying comorbidities, multimorbidity, and frailty in large healthcare data is often a duplicative process
- Each algorithm may have its own slight variation of the input
  - Such as wide vs long data
  - Some are only built for ICD-9, others are built for ICD-9 and ICD-10
- Different measures may also use different components of the data (i.e. ICD codes, CPT codes, HCPCS codes)
- By harmonizing the algorithms, and eliminating this variation, we can more efficiently and effectively implement these measures

What Does The Package Do?

- The `multimorbidity` package includes functions to:
  - Clean the data
  - Limit the data (to match a specific time frame)
  - Run any number of published indices
- The package also includes a number of sample datasets which can help demonstrate the package's functions

Installing the Package

Installing `multimorbidity`

- `multimorbidity` is available on CRAN
  - https://CRAN.R-project.org/package=multimorbidity
  - It can be installed with the following code:

```
install.packages("multimorbidity")
```

- The development version of `multimorbidity` can be found on my GitHub
  - https://github.com/WYATTBENSKEN/multimorbidity

Included Data

Claims Data (i9_i10_comb)

- A dataset with fake patient data for 5 patients
  - Both inpatient and outpatient data, as well as HCPCS codes, and ICD9 and ICD10
- 58 rows, and 11 variables

```
 [1] "date_of_serv" "dx1"          "dx2"
 [4] "dx3"          "dx4"          "dx5"
 [7] "hcpcs"        "icd_version"  "patient_id"
[10] "sex"          "visit_type"
```

Claims Data (i9_i10_comb)

patient_id	sex	date_of_serv	visit_type	dx1	dx2	dx3
1001	male	2012-02-14	ip	2768	4019	3310
1001	male	2013-05-15	ip	486	2768	99591
1001	male	2013-01-10	ot	40290	29620	4019

dx4	dx5	hcpcs	icd_version
29620	2630	E2201	9
4019	3310	E2201	9
NA	NA	E2201	9

ID Data (id)

A dataset with fake patient data, to match the diagnoses, that includes a date of interest - Such as a date of diagnosis with some condition of interest - This is used to limit claims around this time frame

- 5 rows and 3 variables

patient_id	date_of_interest10	date_of_interest9
1001	2021-06-04	2013-06-04
1002	2021-03-11	2013-03-11
1003	2021-08-02	2013-08-02
1004	2021-01-20	2013-01-20
1005	2021-02-14	2013-02-14

Included Data Cleaning Functions

prepare_data()

- Takes our raw claims data, in a number of different forms, and prepares it in a way which allows the other functions in this package to easily work with it.
  - It is recommended to run this function on all data

```
prepare_data(dat = i9_i10_comb,
             id = patient_id,
             style = "wide",
             prefix_dx = "dx",
             hcpcs = "yes",
             prefix_hcpcs = "hcpcs",
             version_var = icd_version,
             type_name = visit_type,
             date = date_of_serv)
```

prepare_data()

- dat, our dataset
- id, the variable name of our our patient id
- style, “wide” or “long”
- prefix_dx, the prefix for all of our diagnoses variables
- hcpcs, “yes” or “no” if our data contains HCPCS codes
- prefix_hcpcs, if our data contains HCPCS codes, the prefix or variable name for these codes
- version_var, the variable which denotes which version of ICD our data are
  - 9 = ICD-9, 10 = ICD-10
- type_name, the variable which denotes inpatient (“ip”) or outpatient (“ot”) for the code(s)
- data, the variable which contains the date of the claim

comorbidity_window()

- Takes prepared data, using the 'prepare_data' function, along with an identification dataset to limit the claims of interest to a specific time window

```
comorbidity_window(id_dat = id,
                   dat = prepared_data,
                   id = patient_id,
                   id_date = date_of_interest9,
                   claims_date = claim_date,
                   time_pre = 60,
                   time_post = 60)
```

comorbidity_window()

- id_dat, our ID dataset
- dat, our claims data
- id, the variable with the unique patient ID
- id_date, the date of interest (around which claims will be limited)
- time_pre, time before the date of interest to limit
- time_post, time after the date of interest to limit

Included Indices

elixhauser()

- The Elixhauser Comorbidities and Comorbidity Index are a widely-used set of comorbidities
  - Developed in 1998 by Elixhauser
  - Indices for mortality and readmission created in 2017 by Moore et al.
- The ICD-10 data contain a larger set of comorbidities and, as of this writing, no calculator for the indices has been released
  - When data contain both ICD-9 and ICD-10, we will use the ICD-9 comorbidities with the Beta code.
- Finally, the original algorithm takes into account DRG, which this package currently does not accommodate.

elixhauser()

```
elixhauser(dat = limit_data,
           id = patient_id,
           dx = dx,
           version = 19,
           version_var = version,
           outpatient_two = "yes")
```

- dat, our cleaned dataset
- id, our patient ID
- dx, the variable which contains our diagnoses
- version, the version of our claims
  - 9 = ICD-9, 10 = ICD-10, 19 = ICD-9 and ICD-10
- version_var, the variable denoting the ICD version of the diagnosis
- outpatient_two, whether or not we want two outpatient claims for a patient to be coded with a comorbidity

elixhauser()

```
 [1] "aids"          "alcohol"      "anemdef"
 [4] "arth"          "bldloss"      "chf"
 [7] "chrnlung"      "coag"         "depress"
[10] "dm"            "dmcx"         "drug"
[13] "elix_death"    "elix_htn_c"   "elix_htn_uc"
[16] "elix_readmit"  "htn_c"        "hypothy"
[19] "id"            "liver"        "lymph"
[22] "lytes"         "mets"         "neuro"
[25] "obese"         "para"         "perivasc"
[28] "psych"         "pulmcirc"     "renlfail"
[31] "tumor"         "ulcer"        "valve"
[34] "wghtloss"
```

charlson()

- The Charlson Comorbidities and Index are, similarly, a widely-used set of comorbidities
  - First developed in 1987 by Charlson et al.
  - This algorithm employs the Deyo et al. list of 17 comorbidities, with the adaptations included in Quan et al.

charlson()

```
charlson(dat = limit_data,
         id = patient_id,
         dx = dx,
         version = 19,
         version_var = version,
         outpatient_two = "yes")
```

- dat, our cleaned dataset
- id, our patient ID
- dx, the variable which contains our diagnoses
- version, the version of our claims
  - 9 = ICD-9, 10 = ICD-10, 19 = ICD-9 and ICD-10
- version_var, the variable denoting the ICD version of the diagnosis
- outpatient_two, whether or not we want two outpatient claims for a patient to be coded with a comorbidity

charlson()

```
 [1] "charlson_cerebro"      "charlson_chf"
 [3] "charlson_chronic_pulm" "charlson_dementia"
 [5] "charlson_diab_c"       "charlson_diab_uc"
 [7] "charlson_hemi_para"    "charlson_hiv"
 [9] "charlson_malig"        "charlson_met_solid"
[11] "charlson_mild_liv"     "charlson_mod_sev_liv"
[13] "charlson_myocar"       "charlson_peptic_ulcer"
[15] "charlson_periph_vasc"  "charlson_renal"
[17] "charlson_rheum"        "charlson_score"
[19] "id"
```

cfi()

- Claims Frailty Index (CFI) is based off of work by Kim et al. in 2018
  - This algorithm uses ICD-9, ICD-10, and procedure codes to establish the frailty score for each patient
  - As the original algorithms included HCPCS/CPT procedure codes, so does this

cfi()

```
cfi(dat = limit_data,
      id = patient_id,
      dx = dx,
      version = 19,
      version_var = version)
```

- dat, our cleaned dataset
- id, our patient ID
- dx, the variable which contains our diagnoses
- version, the version of our claims
  - 9 = ICD-9, 10 = ICD-10, 19 = ICD-9 and ICD-10
- version_var, the variable denoting the ICD version of the diagnosis

cfi()

id	frailty_index
1001	0.36494
1002	0.27935
1003	0.31326
1004	0.27249
1005	0.33721

mwi()

- Multimorbidity Weighted Index (MWI) was created by Wei et al. in 2020
  - This uses ICD-9 codes (note: ICD-10 is not yet available for MWI) to establish a multimorbidity index for each individual

mwi()

```
mwi(dat = limit_data,
      id = patient_id,
      dx = dx,
      version = 9,
      version_var = version)
```

- dat, our cleaned dataset
- id, our patient ID
- dx, the variable which contains our diagnoses
- version, the version of our claims
  - the version must be set to 9 here, as ICD-10 weights are not available yet
- version_var, the variable denoting the ICD version of the diagnosis

mwi()

id	mwi
1001	20.958
1002	3.910
1003	3.511
1004	3.546
1005	0.614

nicholsonfortin()

- Nicholson and Fortin Conditions were first published in 2015
  - Updated to ICD-10 in 2017
- 20 chronic conditions are a standardized list used for multimorbidity research, and developed from a community-based primary healthcare project.

nicholsonfortin()

```
nicholsonfortin(dat = limit_data,
                id = patient_id,
                dx = dx,
                version = 19,
                version_var = version,
                outpatient_two = "yes")
```

- dat, our cleaned dataset
- id, our patient ID
- dx, the variable which contains our diagnoses
- version, the version of our claims
  - 9 = ICD-9, 10 = ICD-10, 19 = ICD-9 and ICD-10
- version_var, the variable denoting the ICD version of the diagnosis
- outpatient_two, whether or not we want two outpatient claims for a patient to be coded with a comorbidity

nicholsonfortin()

```
 [1] "anxietydepress" "arthritis"      "cancer"
 [4] "ckd"            "clrd"           "colon"
 [7] "cvd"            "dementia"       "diabetes"
[10] "heartfail"      "htn"            "hyperlipid"
[13] "id"             "liver"          "musculo"
[16] "obesity"        "osteo"          "stomach"
[19] "stroketia"      "thyroid"        "urinary"
```

Questions?

Contact Information

- Wyatt P. Bensken
- wyattbensken.com
- wpb27 [at] case.edu