dxpr: An R package for generating analysis-ready data from electronic health records—diagnoses and procedures

Yi-Ju Tseng [Corresp., 1, 2], **Hsiang-Ju Chiu** [1], **Chun Ju Chen** [1, 3]

[1] Department of Information Management, Chang Gung University, Taoyuan, Taiwan

[2] Department of Laboratory Medicine, Chang Gung Memorial Hospital at Linkou, Taoyuan, Taiwan

[3] Department of Information Management, National Taiwan University, Taipei, Taiwan

Corresponding Author: Yi-Ju Tseng
Email address: yjtseng.info@gmail.com

Background

Enriched electronic health records (EHRs) contain crucial information related to disease progression, and this information can help with decision-making in the health care field. Data analytics in health care is deemed as one of the essential processes that help accelerate the progress of clinical research. However, processing and analyzing EHR data are common bottlenecks in health care data analytics.

Methods

The *dxpr* R package provides mechanisms for integration, wrangling, and visualization of clinical data, including diagnosis and procedure records. First, the *dxpr* package helps users transform International Classification of Diseases (ICD) codes to a uniform format. After code format transformation, the *dxpr* package supports four strategies for grouping clinical diagnostic data. For clinical procedure data, two grouping methods can be chosen. After EHRs are integrated, users can employ a set of flexible built-in querying functions for dividing data into case and control groups by using specified criteria and splitting the data into before and after an event based on the record date. Subsequently, the structure of integrated long data can be converted into wide, analysis-ready data that are suitable for statistical analysis and visualization.

Results

We conducted comorbidity analyses based on a cohort of newborns from Medical Information Mart for Intensive Care-III (n = 7,833) by using the *dxpr* package. We first defined patent ductus arteriosus (PDA) cases as patients who had at least one PDA diagnosis (ICD, Ninth Revision, Clinical Modification [ICD-9-CM] 7470*). Controls were defined as patients who never had PDA diagnosis. In total, 381 and 7,452 patients with and without PDA, respectively, were included in our study population. Then, we grouped the diagnoses into defined comorbidities. Finally, we observed a statistically significant difference in 8 of the 16 comorbidities among patients with and without PDA, including fluid and electrolyte disorders, valvular disease, and others.

Conclusions

This *dxpr* package helps clinical data analysts combat the common bottleneck caused by clinical data characteristics such as heterogeneity and sparseness.

dxpr: An R package for generating analysis-ready data from electronic health records—diagnoses and procedures

Yi-Ju Tseng[1,2,*], Hsiang-Ju Chiu[1#], Chun Ju Chen[1,3#]

[1] Department of Information Management, Chang Gung University, Taoyuan, Taiwan

[2] Department of Laboratory Medicine, Chang Gung Memorial Hospital at Linkou, Taoyuan, Taiwan

[3] Department of Information Management, National Taiwan University, Taipei, Taiwan

Short Title: An R package for EHR analysis

***Corresponding Author**

Yi-Ju Tseng, Ph.D.

Department of Information Management, Chang Gung University

259 Wen-Hwa 1st Road, Guishan, Taoyuan, Taiwan 333

Yjtseng.info@gmail.com

[#] Hsiang-Ju Chiu and Chun-Ju Chen contributed equally to this manuscript.

Keywords: electronic health records; analysis-ready data, exploratory data analysis, R package

Abstract

Background

Enriched electronic health records (EHRs) contain crucial information related to disease progression, and this information can help with decision-making in the health care field. Data analytics in health care is deemed as one of the essential processes that help accelerate the progress of clinical research. However, processing and analyzing EHR data are common bottlenecks in health care data analytics.

Methods

The *dxpr* R package provides mechanisms for integration, wrangling, and visualization of clinical data, including diagnosis and procedure records. First, the *dxpr* package helps users transform International Classification of Diseases (ICD) codes to a uniform format. After code format transformation, the *dxpr* package supports four strategies for grouping clinical diagnostic data. For clinical procedure data, two grouping methods can be chosen. After EHRs are integrated, users can employ a set of flexible built-in querying functions for dividing data into case and control groups by using specified criteria and splitting the data into before and after an event based on the record date. Subsequently, the structure of integrated long data can be converted into wide, analysis-ready data that are suitable for statistical analysis and visualization.

Results

We conducted comorbidity analyses based on a cohort of newborns from Medical Information Mart for Intensive Care-III (n = 7,833) by using the *dxpr* package. We first defined patent ductus arteriosus (PDA) cases as patients who had at least one PDA diagnosis (ICD, Ninth Revision, Clinical Modification [ICD-9-CM] 7470*). Controls were defined as patients who never had PDA diagnosis. In total, 381 and 7,452 patients with and without PDA, respectively, were included in our study population. Then, we grouped the diagnoses into defined comorbidities.

Finally, we observed a statistically significant difference in 8 of the 16 comorbidities among patients with and without PDA, including fluid and electrolyte disorders, valvular disease, and others.

Conclusions

This *dxpr* package helps clinical data analysts combat the common bottleneck caused by clinical data characteristics such as heterogeneity and sparseness.

1 Introduction

On the basis of the development of electronic health records (EHRs), data analytics in health care is deemed as an essential process for accelerating the progress of clinical research (Hersh, 2007; Jensen, Jensen & Brunak, 2012; Miotto & Weng, 2015). Enriched EHRs contain crucial information related to disease progression, and this information can help with decision making in the health care field including for treatment selection and disease diagnosis (Jensen, Jensen & Brunak, 2012; Raghupathi & Raghupathi, 2014). However, processing and analyzing EHR data are usually challenging because of their heterogeneity and sparsity. These inherent characteristics create a common bottleneck in health care big data analytics (Wu, Roy & Stewart, 2010; Hripcsak & Albers, 2013; Weiskopf & Weng, 2013). Moreover, executing clinical data analysis project across different departments or institutes is difficult because clinical data formats and terminologies used to describe clinical conditions may vary across departments. A method that can standardize and facilitate the sharing of data or analysis pipelines from multiple sources is needed in research on clinical data analysis. Several common data models (CDMs) have been developed for eliminating clinical data format barriers, including the National Patient-Centered Clinical Research Network (PCORnet) (Fleurence et al., 2014; PCORnet, 2020) and Observational Medical Outcomes Partnership (OMOP) CDM (Observational Health Data Sciences and Informatics, 2020). The concept of CDM is to transform data into a CDM and terminology and then allow users to perform systematic analyses by using various sources. Although a CDM can help perform systematic analyses across different sources, the integration of clinical data and the preparation of analysis-ready data are unsolved issues.

The proposed open-source *dxpr* R package is a software tool aimed at expediting general EHR or claims data analyses through incorporating several functions that enable users to standardize, integrate, wrangle, and visualize clinical diagnosis and procedure records. Preparing

an analysis-ready dataset from EHRs or claims data is a complex task that requires both medical knowledge and data science skills. The proposed *dxpr* package simplifies and accelerates the workflow for EHR data extraction and helps clinical data analysts generate simple and clean scripts that can easily be shared and reproduced. The *dxpr* package enables researchers to explore EHRs or claims data to acquire crucial information, understand disease progression, and analyze outcomes without writing complicated data preprocessing scripts. Moreover, the proposed package can support collaborative research across multiple data sources as long as the data include general diagnosis- or procedure-related information.

The *dxpr* package has three phases to process and analyze diagnosis codes in EHRs (Fig. 1). In the first phase, namely data integration, we transform diagnosis codes into a uniform format and provide four strategies to group diagnoses into clinically meaningful categories before the wrangling process. In the second phase, namely, data wrangling, users can use provided functions to query eligible cases, split data based on the index date, and calculate condition era according to the grouped diagnostic categories of each patients. Furthermore, exploratory data analysis preparation can be performed in this phase. Moreover, the *dxpr* package provides a function to convert a long format of grouped data into a wide format, which fits other analytical and plotting functions from other packages better. In the last phase, namely visualization, we provide overviews for diagnosis standardization and data integration, such as comorbidity distribution in the study population, comorbidity differences between case and control groups, and the most common diagnoses that failed to be grouped or transformed. For processing and analyzing procedure codes, the concept is similar to diagnosis.

2 Materials and methods

The *dxpr* R package provides mechanisms for integrating, wrangling, and visualizing

clinical data, including diagnosis and procedure records. First, the *dxpr* package helps users to transform International Classification of Diseases (ICD) codes to a uniform format. After code format transformation, the *dxpr* package supports four strategies for grouping clinical diagnostic data. For clinical procedure data, two grouping methods can be chosen. After EHRs are integrated, users can use a set of flexible built-in querying functions for dividing data into case and control groups based on specified criteria and splitting the data into before and after an event groups according to the record date. Then, the structure of integrated long data can be converted into wide, analysis-ready data that are more suitable for statistical analysis and visualization. The usage details are presented in the Supplementary Data S1 and S2.

Preparation for analysis

The current version of the package is available at Github (https://github.com/DHLab-TSENG/dxpr, Supplementary Data S3) and is accessible through the devtools package (Wickham H). To install the *dxpr* R package, users can type the following commands in an R session:

```
devtools::install_github("DHLab-TSENG/dxpr")
library(dxpr)
```

The imported EHR dataset must contain at least three columns as indicated below:

- Member ID: a patient identifier, which can be numeric, alphanumeric, or a list of characters.
- Diagnosis/procedure code: ICD-9 or ICD-10 code assigned to a visit or an admission.
- Visit or admission date: the date of the visit, admission, or clinical service provided. The date has to be recorded in year–month–day format (YYYY/MM/DD or YYYY-MM-DD).

Column names can be passed in each function by using function arguments.

We illustrate the use of the *dxpr* package with a diagnostic sample dataset of 10-year admissions of 38 patients, sampleDxFile, and the first five records are shown in Table 1.

Data integration

Code format transformation

The *dxpr* package first transforms ICD diagnostic codes into a uniform format before code grouping. ICD-9 and ICD-10 diagnostic codes (U.S. Centers for Medicare & Medicaid Services, b) have two formats, namely decimal (with a decimal place separating the code) and short formats. Different hospitals, grouping methods, or standards coded ICD into different formats. For example, studies using Clinical Classifications Software (CCS) (Healthcare Cost and Utilization Project (HCUP), 2017, 2019a) and comorbidity measures, such as Elixhauser and Charlson (Elixhauser et al., 1998; Menendez et al., 2014; Moore et al., 2017), have coded the ICD in a short format, and a phenome-wide association study (PheWAS) (Denny et al., 2010) coded the ICD in a decimal format. Therefore, format transformation is required before code grouping, and the transformation type is decided by the chosen grouping method.

The transformation function (icdDxShortToDecimal) converts ICD-9 and ICD-10 codes into a uniform decimal format because a decimal format is needed for grouping diagnostic codes in PheWAS classification. Similar to icdDxShortToDecimal, icdDxDecimalToShort function converts diagnostic codes into a uniform short format, which can be used for grouping to CCS, Elixhauser, or other classifications. These transformative functions not only convert ICD codes into uniform format codes but also check for potential coding errors. We provide two types of warning messages: wrong ICD format and wrong ICD version. Additional suggestions are generated to help users adjust potential incorrect ICD codes if available.

```
ICD_Decimal <- icdDxShortToDecimal(dxDataFile = sampleDxFile,
                                   icdColName = ICD,
                                   dateColName = Date,
                                   icd10usingDate = "2015/10/01")
sampleDxFile$Decimal <- ICD_Decimal$ICD
head(sampleDxFile)
   ID  ICD       Date Decimal
1:  A2 Z992 2020-05-22   Z99.2
2:  A5 Z992 2020-01-24   Z99.2
3:  A8 Z992 2015-10-27   Z99.2
4: A13 Z992 2020-04-26   Z99.2
5: A13 Z992 2025-02-02   Z99.2
6: A15 Z992 2023-05-12   Z99.2
tail(ICD_Decimal$Error)
     ICD count IcdVersionInFile     WrongType Suggestion
1:  75.52     4           ICD 9  Wrong format
2:  E03.0     4           ICD 9 Wrong version
3:    650     4          ICD 10 Wrong version
4: 123.45     3          ICD 10  Wrong format
5:  755.2     3           ICD 9  Wrong format     755.29
6:   7552     2           ICD 9  Wrong format      75529
```

Code grouping

The code grouping functions collapse clinical diagnostic data (ICD-9/ICD-10 codes) (U.S. Centers for Medicare & Medicaid Services, b) into a smaller number of clinically meaningful categories that are more useful for presenting descriptive statistics than using individual diagnostic codes (Healthcare Cost and Utilization Project (HCUP), 2019b). The *dxpr* package supports four strategies to group EHR diagnosis codes, namely CCS (Healthcare Cost and Utilization Project (HCUP), 2017, 2019a), PheWAS (Denny et al., 2010) (icdDxToPheWAS), comorbidity measures (Elixhauser et al., 1998; Menendez et al., 2014; Moore et al., 2017), and self-defining grouping methods. The CCS grouping strategies includes single-level CCS (icdDxToCCS) and multiple-level CCS (icdDxToCCSLvl) (Healthcare Cost and Utilization Project

(HCUP), 2017, 2019a), comorbidity measures (icdDxToComorbid) includes Elixhauser, Agency for Healthcare Research and Quality (AHRQ) and Charlson (Elixhauser et al., 1998; Menendez et al., 2014; Moore et al., 2017), and self-defining grouping methods includes precise matching (icdDxToCustom) and searching for lines containing a match (icdDxToCustomGrep). The grouping functions return two tables of the dataset, one is data with the corresponding grouping categories of each ICD (Table 2), and the other is summarized data exhibiting the earliest/latest record date and diagnosis counts in the same grouping category for each patient (Table 3). For example, after executing function icdDxToCCS for the records of patients A and B, two output types are shown in Tables 2 and 3, respectively. Patient A has three diagnosis records (ICD codes: 78550, 78552, and 785.59), which are all in the "shock" category of the CCS classification, with the earliest record on September 1, 2013 and the latest one on October 1, 2014. The icdDxToCCS function mapped corresponding CCS categories for these ICD codes and returned the grouping results (Table 2). Similarly, patient B has two diagnosis records (ICD codes: 78552 and 250.00) in the "shock" category and "Diabetes mellitus without complication" category of CCS classification, and the grouping results are also shown in Table 2. According to these diagnosis records shown in Table 2, Table 3 shows that icdDxToCCS function can summarize the first and last dates of diagnosis, the total number of diagnoses, and the period between the first and last diagnoses for each category, which can be used for designing the analysis strategy. While icdDxToCCS groups codes into single-level CCS, icdDxToCCSLvl groups codes into multi-level CCS. Multi-level CCS expands single-level CCS into a four-level hierarchical system for diagnoses, which provide the opportunity to examine general aggregations or to assess specific conditions ("HCUP-US Tools & Software Page"). For instance, if a user wishes to group codes into the second level of multi-level CCS, then this task can be performed through simply entering "`ccslvl2`" as the assigned grouping type. These

grouping functions not only facilitate users to convert original diagnosis records from detailed levels into clinically meaningful diagnostic groups for further analysis but also provide aggregated information of each diagnostic group that can help research design and hypothesis generation, such as filtering out data based on specified criteria (e.g., first diagnosis dates of specific chronic disease).

The usage of code classification function for CCS is as follows:

```
## ICD to CCS description
CCS_description <- icdDxToCCS(dxDataFile = sampleDxFile,
                                idColName = ID,
                               icdColName = ICD,
                               dateColName = Date,
                               icd10usingDate = "2015-10-01",
                               isDescription = TRUE)
CCS_description$groupedDT[CCS_description$groupedDT$ID=="A0",]
   Short ID   ICD       Date CCS_CATEGORY_DESCRIPTION
1:  5855 A0  5855 2013-12-20   Chronic kidney disease
2: V4511 A0 V4511 2012-04-05   Chronic kidney disease
3:  V560 A0  V560 2010-03-28   Chronic kidney disease
4:  5853 A0  5853 2010-10-29   Chronic kidney disease
5:  5856 A0  5856 2009-07-25   Chronic kidney disease
6:   001 A0   001 2014-11-05                    <NA>
7: A0.11 A0 A0.11 2017-01-31                    <NA>
8: A0.11 A0 A0.11 2023-08-12                    <NA>
head(CCS_description$summarised_groupedDT, 5)
    ID CCS_CATEGORY_DESCRIPTION firstCaseDate endCaseDate count    period
1:  A0   Chronic kidney disease    2009-07-25  2013-12-20     5 1609 days
2:  A1   Chronic kidney disease    2006-11-29  2014-09-24     5 2856 days
3: A10   Chronic kidney disease    2007-11-04  2012-07-30     5 1730 days
4: A11   Chronic kidney disease    2008-03-09  2011-09-03     5 1273 days
5: A12   Chronic kidney disease    2006-05-14  2015-06-29     5 3333 days
```

Data wrangling

Case selection

In clinical data analysis projects, the most crucial step is case definition and selection, such as defining Lyme disease cases from claims data (Tseng et al., 2015) or defining acute ischemic stroke from EHR (Tseng et al., 2020). The analysis results could change based on case definition and lead to a different conclusion. The query function selectCases can select cases matching case definitions. Users can select cases based on diagnosis (ICD) or diagnostic categories (CCS, PheWAS, comorbidities, or self-defined diagnostic categories). Moreover, the function provides an option to set the minimum number of diagnoses within a specific duration. For example, users can extract diabetes cases by assigning at least two diagnoses in ICD codes “250.xx” or “E10.x-E14.x” within 730 days when a user applies the validated diabetes case definition: “two physician claims within 2 years with diagnosis codes 250.xx or E10.x-E14.x” (Chen et al., 2010). The output dataset of this function provides the start and end dates of the cases, the number of days between them, and the most common ICD codes used in the case definition. Furthermore, a list of people who did not satisfy the required case conditions or practically match the case definition is appended in the returned output table, and these individuals can be defined as a control group or be removed.

```
Case <- selectCases(dxDataFile = sampleDxFile,
                    idColName = ID,
                    icdColName = ICD,
                    dateColName = Date,
                    icd10usingDate = "2015/10/01",
                    groupDataType = ccslvl2,
                    caseCondition = "Diseases of the urinary system",
                    isDescription = TRUE,
                    caseCount = 1,
                    periodRange = c(30, 365))
```

```
head(Case)
    ID selectedCase count firstCaseDate endCaseDate    period MostCommonICD MostCommonICDCount
1:  A3     Selected     5    2008-07-08  2014-02-24 2057 days         V420                3
2:  A1     Selected     5    2006-11-29  2014-09-24 2856 days         5855                2
3: A10     Selected     5    2007-11-04  2012-07-30 1730 days        V5631                2
4: A12     Selected     5    2006-05-14  2015-06-29 3333 days         5859                2
5: A13     Selected     5    2006-04-29  2025-02-02 6854 days         5855                2
6: A15     Selected     5    2007-05-25  2023-05-12 5831 days        V5631                2
tail(Case)
   ID selectedCase count firstCaseDate endCaseDate  period MostCommonICD MostCommonICDCount
1: D3 non-Selected    NA         <NA>       <NA> NA days         <NA>                NA
2: D4 non-Selected    NA         <NA>       <NA> NA days         <NA>                NA
3: D5 non-Selected    NA         <NA>       <NA> NA days         <NA>                NA
4: D6 non-Selected    NA         <NA>       <NA> NA days         <NA>                NA
5: D7 non-Selected    NA         <NA>       <NA> NA days         <NA>                NA
6: D8 non-Selected    NA         <NA>       <NA> NA days         <NA>                NA
```

Eligible period identification

In some clinical data, such as claims data, individuals can join or leave the program on different dates, and the length of available records might affect the analysis completeness. The *dxpr* package provides a function `getEligiblePeriod` for researchers to identify the first/last record date for each patient. These outputs can be used as an index date for case exclusion, such as cases without at least 6 months washout or follow-up period, or further data splitting.

```
admissionDate <- getEligiblePeriod(dxDataFile = sampleDxFile,
                                   idColName = ID,
                                   dateColName = Date)
head(admissionDate)
    ID firstRecordDate endRecordDate
1:  D6      2005-10-09    2025-01-05
2: A12      2006-01-12    2022-06-12
3:  D1      2006-02-12    2024-04-04
4: A13      2006-04-29    2025-02-02
5:  A9      2006-06-30    2023-12-10
6:  D2      2006-09-01    2025-08-11
```

Data splitting based on index date and moving window

In clinical data analysis projects, users usually need to extract data based on a specific clinical event (e.g., extracting data before the first Lyme disease diagnosis in the records (Tseng et al., 2017)). The date of the specific event (index date) can be the first/last record date of the events or patient record, and the table of the index date for each individual can be generated using selectCases or getEligiblePeriod function, respectively. The *dxpr* package provides a convenient function splitDataByDate that can split data through classifying the data recorded before or after the defined index date and calculating the period between the record date and index date based on a self-defined window. For example, if a user needs to aggregate the data by using a 30-day window, the data recorded on 15 and 45 days after the index date will be defined as window 1 and window 2, respectively. The output of splitDataByDate function helps users to split the data based on the study design, and this can be applied to further time-series multiple-measurement analysis with period information.

```
indexDateTable <- data.frame (ID = c("A0","B0","C0","D0"),
                              indexDate = c("2023-08-12", "2015-12-26",
                                            "2015-12-05", "2017-01-29"))
Data <- splitDataByDate(dxDataFile = sampleDxFile[grepl("A0|B0|C0|D0",ID),],
                      idColName = ID,
                      icdColName = ICD,
                      dateColName = Date,
                         indexDateFile = indexDateTable,
                         gap = 30)
Data[6:11,]
   ID   ICD       Date  indexDate timeTag window
1: A0   001 2014-11-05 2023-08-12       B    107
2: A0 A0.11 2017-01-31 2023-08-12       B     80
3: A0 A0.11 2023-08-12 2023-08-12       A      1
4: B0  N185 2015-12-26 2015-12-26       A      1
5: B0  N189 2017-11-27 2015-12-26       A     24
6: B0 A0.11 2017-12-19 2015-12-26       A     25
```

Condition era generation

Condition era is a means to apply consistent rules for medical conditions to infer distinct episodes in care, generated through integrating distributed clinical records into a single progression record (Ryan, 2010). The concept of condition era is committed to the length of the persistence gap: when the time interval of any two consecutive admissions for certain conditions is smaller than the length of the persistence gap, then these two admission events will be aggregated into the same condition era. Each condition era consists of one or many events, and differences between any two consecutive admission events are all within the persistence gap. For example, an episode of influenza may include single or multiple outpatient visits, and the length of the influenza course should be the period between the first and last visits of the episode. getConditionEra function calculates condition era by using the grouped categories or self-defining groups of each patient and then generates a table with individual IDs, the first and last record of an era, and the sequence number of each episode. Users can easily convert scattered diagnoses into an episode of condition based on the chararistics of target disease progression with the proposed function.

```
Era <- getConditionEra(dxDataFile = sampleDxFile,
                       idColName = ID,
                       icdColName = ICD,
                       dateColName = Date,
                       icd10usingDate = "2015/10/01",
                       groupDataType = CCS,
                       isDescription = FALSE,
                       gapDate = 360)
head(Era)
   ID CCS_CATEGORY era firstCaseDate endCaseDate count   period
1: A0         158   1    2009-07-25  2010-10-29     3 461 days
2: A0         158   2    2012-04-05  2012-04-05     1   0 days
```

```
3: A0         158   3    2013-12-20  2013-12-20    1   0 days
4: A1         158   1    2006-11-29  2006-11-29    1   0 days
5: A1         158   2    2008-06-25  2008-06-25    1   0 days
6: A1         158   3    2012-06-19  2013-04-28    2 313 days
```

Analysis-ready data generation

After data integration and wrangling, researchers often need to further analyze these processed data, and function groupedDataLongToWide converts the long format of grouped data into a wide format, which is fit for other analytical and plotting packages, such as tableone (Yoshida K).

```
CHARLSON <- icdDxToComorbid(dxDataFile = sampleDxFile,
                            idColName = ID,
                            icdColName = ICD,
                            dateColName = Date,
                            icd10usingDate = "2015-10-01",
                            comorbidMethod = CHARLSON)
groupedData_Wide <- groupedDataLongToWide(dxDataFile  = CHARLSON$groupedDT,
                                          idColName = ID,
                                          categoryColName = Comorbidity,
                                          dateColName = Date,
                                          reDup = TRUE,
                                          numericOrBinary = B,
                                          count = 2)
head(groupedData_Wide, 5)
   ID CANCER  CEVD  COPD DIAB_C  MSLD  PARA   PUD   PVD   RD Rheum
1  A0  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
2  A1  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
3 A10  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
4 A11  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
5 A12  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
```

Visualization

Pareto chart of error ICD

When code transformation is implemented in the *dxpr* package, it generates unified data of

diagnosis codes with potential errors. Function plotICDError visualizes codes with potential error by using the Pareto chart containing a bar plot where error ICD codes are arranged in descending order, and the cumulative total is represented by the line. Users can sort based on the counts of error ICD codes and set the top selected number of the ordered dataset. For instance, if a user chooses the top 10 ordinal rankings, then the Pareto chart shows a plot of the top 10 common error ICD codes and a list with details of these 10 and other error ICD codes.

```
error <- icdDxDecimalToShort(dxDataFile = sampleDxFile,
                             icdColName = ICD,
  dateColName = Date,
icd10usingDate = "2015/10/01")
plot1 <- plotICDError(errorFile = error$Error,
           icdVersion = all,
           wrongICDType = all,
           groupICD = FALSE,
           others = TRUE,
                        topN = 10)

plot1$ICD
       ICD count CumCountPerc IcdVersionInFile     WrongType Suggestion
 1:  A0.11    20       18.35%          ICD 10  Wrong format
 2:  V27.0    18       34.86%          ICD 10 Wrong version
 3:   E114     8        42.2%          ICD 10  Wrong format
 4: A01.05     8       49.54%           ICD 9 Wrong version
 5:  42761     7       55.96%          ICD 10 Wrong version
 6:  Z9.90     6       61.47%          ICD 10  Wrong format
 7:    F42     6       66.97%          ICD 10  Wrong format
 8:  V24.1     6       72.48%          ICD 10 Wrong version
 9:  A0105     5       77.06%           ICD 9 Wrong version
10:    001     5       81.65%           ICD 9  Wrong format      0019
11: others    20         100%           ICD 9  Wrong format
```

Bar chart of diagnostic categories

Function plotDiagCat provides an overview of the grouping categories of the diagnoses and summarizes the proportion of individuals diagnosed with grouped diagnostic categories in the

whole study population or case and control groups in a bar chart. Users can observe the number and percentage of diagnostic categories in their dataset through this function. Furthermore, this function compares the usage of significantly different diagnostic categories between case and control groups by using the chi-square test or Fisher's exact test when the data does not match the assumptions of the chi-square test. The default level of statistical significance is considered at 5% ($p = 0.05$). Researchers can set a threshold of the top N significant grouped categories and the minimum prevalence of the diagnostic groups in the case or control group.

The "`percentage`" column shows the proportion of individuals diagnosed with the diagnostic category in the group. For example, there are 38 patients in the sample file, and "Renal Failure" defined in Elixhauser comorbidity accounts for 63.16% of the population (24/38).

```
ELIX <- icdDxToComorbid(dxDataFile  = sampleDxFile,
                        idColName = ID,
                        icdColName = ICD,
                        dateColName = Date,
                        icd10usingDate = "2015-10-01",
                        comorbidMethod = ELIX)
groupedDataWide <- groupedDataLongToWide(dxDataFile  = ELIX$groupedDT,
                                         idColName = ID,
                                         categoryColName = Comorbidity,
                                         dateColName = Date,
                                         reDup = TRUE,
                                         numericOrBinary = B)
plot2 <- plotDiagCat(groupedDataWide = groupedDataWide,
                           idColName = ID,
                             topN = 10,
                          limitFreq = 0.01)

plot2$sigCate
   DiagnosticCategory  N Percentage
1:           RENLFAIL 24     63.16%
2:              TUMOR  6     15.79%
3:               ARTH  5     13.16%
4:              LYMPH  4     10.53%
5:              PSYCH  4     10.53%
```

```
 6:             DRUG  3     7.89%
 7:            NEURO  3     7.89%
 8:             PARA  2     5.26%
 9:         PERIVASC  2     5.26%
10:            VALVE  2     5.26%
```

Clinical procedure analysis

As diagnosis codes, ICD-9-Procedure Coding System (PCS) code also has two formats, namely decimal and short, whereas ICD-10-PCS code only has a short format. The functions (icdPrToCCS and icdPrToProcedureClass) provide two strategies (CCS and procedure class) to collapse ICD procedure codes into clinically meaningful categories for further analysis. This procedure has two CCS classifications: single and multiple levels. The usage is similar to the diagnostic CCS classification. A sample file (samplePrFile) is provided with procedure records, including three patients and 170 records.

The procedure classes (Healthcare Cost and Utilization Project (HCUP), 2016) are created to facilitate health services research on hospital procedures by using administrative data. The procedure classes provide a standard to categorize individual procedure codes into one of the four broad categories: minor diagnostic, minor therapeutic, major diagnostic, and major therapeutic. The aforementioned classification functions mentioned allow the researcher to readily determine whether a procedure is diagnostic or therapeutic and whether a procedure is minor or major in terms of invasiveness, resource use, or both.

Use case

To illustrate the main features in the *dxpr* package and the typical workflow, we demonstrated an analysis using the package among newborns who were diagnosed with patent ductus arteriosus (PDA) from Medical Information Mart for Intensive Care-III (MIMIC-III) (L.

et al., 2000; Johnson et al., 2016). MMIC-III is a publicly available database comprising deidentified health-related data associated with the admissions of approximately 60,000 patients who stayed in the critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012.

We provided a sample file sampleFile_MIMIC obtained from MIMIC-III (L. et al., 2000; Johnson et al., 2016), a medical dataset of 7,833 newborn patients with 45,674 admissions. This dataset is used for verifying the comorbidity difference between patients with and without PDA based on the *dxpr* package. In this study, we defined PDA cases as patients who had at least one PDA diagnosis (ICD-9-CM 7470*). The controls are defined as patients who never had PDA diagnosis.

Performance analysis

The *dxpr* package is designated to accelerate the process of large EHR data integration and provide the ready-for-analysis dataset from the integrated EHR data. We verified the running time 100 times with a simulated dataset of 953,294 unique patients and 7,948,418 distinct diagnosis records in a standard personal computer with 64 GB DDR4 2133GHz RAM and an Intel® Core™ i7-6700 (CPU @3.40GHz), using Windows 10 (1809), R 4.0.1 (64 bits), and RStudio 1.2.5033.

3 Result

An example of analysis—patients with PDA

We conducted comorbidity analyses based on a cohort of newborns from MIMIC-III (n = 7,833) by using *dxpr* and tableone (Yoshida K) packages. In the *dxpr* package, we first use selectCases function to define case (PDA) and control (non-PDA) groups. In total, 381 and 7,452

patients with and without PAD were included in our study, respectively. Then, icdDxToComorbid function was applied to group diagnoses into AHRQ-defined comorbidities. Finally, we analyzed and graphed the AHRQ-defined comorbidities based on plot_groupedData function (Fig. 2) by using the chi-square test and Fisher's exact test. To focus on comorbidities that were essential and recorded in adequate individuals in our study population, we excluded comorbidities recorded in <1% of the patients in the PDA or non-PDA group. The analysis-ready data generated by groupedDataLongToWide can be passed to the tableone (Yoshida K) package to create objects summarizing all comorbidities stratified by patients with and without PDA and by performing the statistical chi-square tests. The AHRQ comorbidity table revealed 8 of the 16 statistically significant comorbidities ($p < 0.05$, Table 4) among patients with and without PDA, and the comorbidities are visualized in Fig. 2.

Performance

For a simulated dataset of 953,294 unique patients and 7,948,418 admission records, code grouping with CCS-defined comorbidities required 149±2.48 seconds (including code transformation). Case selection required 238±3.05 seconds to query patients with diseases of the urinary system, eligible period identification required 1.12±0.22 seconds to find the first and last admission date for each patient, data splitting with the first admission date for each patient required 6.50±0.42 seconds, condition era generation required 372±6.39 seconds, and analysis-ready data generation required 3.75±0.27 seconds.

4 Discussion and conclusions

The *dxpr* package considerably simplifies the extraction, accelerates the processing of clinical data research, and enables researchers to prepare analysis-ready data with a standard

workflow. The package had been developed and tested using structured clinical data, such as critical care data (MIMIC-III (Johnson et al., 2016)), a multi-institutional medical care database (Chang Gung Research Database (Tsai et al., 2017; Tseng et al., 2020)), and claims data (National Health Insurance Research Database (Hsieh et al., 2019)), indicating that the package can be applied to data from different countries, institutions, and data structures. The available functions are summarized in Table 5.

Several software and packages were developed to facilitate clinical data analysis. rEHR (Springate et al., 2017) established a clinical data analysis workflow to simplify the processing of EHR. The rEHR package simplifies the process of extracting data from EHR databases. It used the database backend that can accelerate data access and process times. However, this design needs database backend, which might not be suitable in many circumstances. Furthermore, the international diagnosis coding standard, such as ICD, were not used in the package. The ICD (Wasey & Lang, 2020) package is designed for calculating comorbidities and medical risk scores with ICD-9 and ICD-10 codes. It is helpful to group ICD codes according to comorbidities. However, in clinical data analysis, eligible case selection, data split based on the defined index date, and visualization are also essential. Therefore, we designed and developed the *dxpr* package to facilitate diagnosis data analysis.

The proposed package has limitations, which come from either the data or package itself. For analyzing clinical data, the *dxpr* package highly depends on diagnosis and procedure codes, but these codes may vary in accuracy across different institutions [10]. Furthermore, the effect of switching diagnosis codes from ICD-9 to ICD-10 should be considered if the analysis period is across the switching date. In addition to diagnosis and procedure data, the other data not included in proposed packages, such as medication data, are important in clinical data analysis projects. In the R ecosystem, the AdhereR (Dima & Dediu, 2017) package implements a set of functions that

564 are consistent with current adherence guidelines and definitions. Finally, the *dxpr* package is
565 focused on analysis-ready data generation so that the statistic method incorporation may be
566 insufficient. However, the R ecosystem's most significant advantage is that many well-
567 developed packages were developed to facilitate statistical analysis. In the use case
568 demonstration, our package can be used with other packages, such as tableone. The tableone
569 (Yoshida K) package is developed to ease the construction of the common "Table 1" in research
570 papers, providing patient baseline characteristics table with summary statistics and hypothesis
571 tests.

572 We demonstrated that the *dxpr* package can play an essential role in complex clinical data
573 preprocessing and analysis-ready data generation through integrating the international standard
574 of clinical data. This package helps clinical data analysts combat the common bottleneck caused
575 by certain clinical data characteristics, such as heterogeneity and sparseness.
576

577 ## 5 Supporting information

578 **Supplementary Data S1.** Sample code of the *dxpr* package provided in the manuscript
579 https://github.com/DHLab-TSENG/dxpr-paper/blob/main/SampleCode.md
580

581 **Supplementary Data S2.** A detailed example of the usage of the *dxpr* package
582 https://github.com/DHLab-TSENG/dxpr-paper/blob/main/Supplement.md
583

584 **Supplementary Data S3.** The source code of the *dxpr* package
585 https://github.com/DHLab-TSENG/dxpr
586
587

Acknowledgements

We thank Ru-Fang Hu, Yi-An Zhu, and Chia-Wei Chang, from Department of Information Management, Chang Gung University, for testing the *dxpr* package. This manuscript was edited by Wallace Academic Editing.

Funding

This research was supported by Chang Gung Memorial Hospital (CMRPD3K0011), the Ministry of Science and Technology, Taiwan (MOST 109-2636-E-182-001). The funders had no role in the design and conduct of the study; collection, management, analysis, and interpretation of the data; preparation, review, or approval of the manuscript; and decision to submit the manuscript for publication.

Conflicts of Interest

The authors declare no conflicts of interest. The funders had no role in the study design; collection, analyses, or interpretation of data; manuscript writing, or decision to publish the results.

References

Chen G, Khan N, Walker R, Quan H. 2010. Validating ICD coding algorithms for diabetes mellitus from administrative data. *Diabetes research and clinical practice* 89:189–195.

Denny JC, Ritchie MD, Basford MA, Pulley JM, Bastarache L, Brown-Gentry K, Wang D, Masys DR, Roden DM, Crawford DC. 2010. PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene-disease associations. *Bioinformatics* 26:1205–1210.

Dima AL, Dediu D. 2017. Computation of adherence to medication and visualization of medication

612 histories in R with AdhereR: Towards transparent and reproducible use of electronic healthcare data.
613 *PLOS ONE* 12:e0174426. DOI: 10.1371/journal.pone.0174426.
614 Elixhauser A, Steiner C, Harris DR, Coffey RM. 1998. Comorbidity Measures for Use with
615 Administrative Data. *Medical Care* 36:8–27. DOI: 10.1097/00005650-199801000-00004.
616 Fleurence RL, Curtis LH, Califf RM, Platt R, Selby J V., Brown JS. 2014. Launching PCORnet, a
617 national patient-centered clinical research network. *Journal of the American Medical Informatics*
618 *Association* 21:578–582. DOI: 10.1136/amiajnl-2014-002747.
619 HCUP-US Tools & Software Page.
620 Healthcare Cost and Utilization Project (HCUP). 2016.HCUP Procedure Classes. *Available at*
621 *https://www.hcup-us.ahrq.gov/toolssoftware/procedure/procedure.jsp*
622 Healthcare Cost and Utilization Project (HCUP). 2017.Clinical Classifications Software (CCS) for ICD-
623 9-CM. *Available at https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp* (accessed October 8,
624 2017).
625 Healthcare Cost and Utilization Project (HCUP). 2019a.Beta Elixhauser Comorbidity Software for ICD-
626 10-CM
627 Healthcare Cost and Utilization Project (HCUP). 2019b.Beta Clinical Classifications Software (CCS) for
628 ICD-10-CM/PCS
629 Hersh WR. 2007. Adding value to the electronic health record through secondary use of data for quality
630 assurance, research, and surveillance. *American Journal of Managed Care* 13:277–278. DOI:
631 10.1001/jama.229.4.457.
632 Hripcsak G, Albers DJ. 2013. Next-generation phenotyping of electronic health records. *Journal of the*
633 *American Medical Informatics Association* 20:117–121. DOI: 10.1136/amiajnl-2012-001145.
634 Hsieh C-Y, Su C-C, Shao S-C, Sung S-F, Lin S-J, Yang Kao Y-H, Lai EC-C. 2019. Taiwan's National
635 Health Insurance Research Database: past and future. *Clinical Epidemiology* Volume 11:349–358.

636 DOI: 10.2147/CLEP.S196293.

637 Jensen PB, Jensen LJ, Brunak S. 2012. Mining electronic health records: towards better research
638 applications and clinical care. *Nature Reviews Genetics* 13:395–405. DOI: 10.1038/nrg3208.

639 Johnson AEW, Pollard TJ, Shen L, Lehman LH, Feng M, Ghassemi M, Moody B, Szolovits P, Anthony
640 Celi L, Mark RG. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*
641 3:160035. DOI: 10.1038/sdata.2016.35.

642 L. GA, N. ALA, Leon G, M. HJ, Ch. IP, G. MR, E. MJ, B. MG, Chung-Kang P, Eugene SH. 2000.
643 PhysioBank, PhysioToolkit, and PhysioNet . *Circulation* 101:e215–e220. DOI:
644 10.1161/01.CIR.101.23.e215.

645 Menendez ME, Neuhaus V, Van Dijk CN, Ring D. 2014. The Elixhauser comorbidity method
646 outperforms the Charlson index in predicting inpatient death after orthopaedic surgery. *Clinical*
647 *Orthopaedics and Related Research* 472:2878–2886. DOI: 10.1007/s11999-014-3686-7.

648 Miotto R, Weng C. 2015. Case-based reasoning using electronic health records efficiently identifies
649 eligible patients for clinical trials. *Journal of the American Medical Informatics Association*
650 22:e141–e150. DOI: 10.1093/jamia/ocu050.

651 Moore BJ, White S, Washington R, Coenen N, Elixhauser A. 2017. Identifying Increased Risk of
652 Readmission and In-hospital Mortality Using Hospital Administrative Data. *Medical Care* 55:698–
653 705. DOI: 10.1097/MLR.0000000000000735.

654 Observational Health Data Sciences and Informatics. 2020.OMOP Common Data Model. *Available at*
655 *https://www.ohdsi.org/data-standardization/the-common-data-model/* (accessed July 28, 2020).

656 PCORnet. 2020.PCORnet Common Data Model (CDM). *Available at https://pcornet.org/data-driven-*
657 *common-model/* (accessed July 28, 2020).

658 Raghupathi W, Raghupathi V. 2014. Big data analytics in healthcare: promise and potential. *Health*
659 *Information Science and Systems* 2:3. DOI: 10.1186/2047-2501-2-3.

660 Ryan P. 2010. Establishing a Condition Era Persistence Window for Active Surveillance. *Foundation for*
661 *the National Institutes of Health*:1–13.
662 Springate DA, Parisi R, Olier I, Reeves D, Kontopantelis E. 2017. rEHR: An R package for manipulating
663 and analysing electronic health record data. *PLoS ONE* 12:e0171784. DOI:
664 10.1371/journal.pone.0171784.
665 Tsai MS, Lin MH, Lee CP, Yang YH, Chen WC, Chang GH, Tsai Y Te, Chen PC, Tsai YH. 2017. Chang
666 Gung Research Database: A multi-institutional database consisting of original medical records.
667 *Biomedical Journal* 40:263–269. DOI: 10.1016/j.bj.2017.08.002.
668 Tseng Y-J, Cami A, Goldmann DA, DeMaria A, Mandl KD. 2015. Incidence and Patterns of Extended-
669 Course Antibiotic Therapy in Patients Evaluated for Lyme Disease. *Clinical Infectious Diseases*
670 61:1536–1542. DOI: 10.1093/cid/civ636.
671 Tseng Y-J, DeMaria A, Goldmann DA, Mandl KD. 2017. Claims-Based Diagnostic Patterns of Patients
672 Evaluated for Lyme Disease and Given Extended Antibiotic Therapy. *Vector-Borne and Zoonotic*
673 *Diseases* 17:116–122. DOI: 10.1089/vbz.2016.1991.
674 Tseng Y-J, Hu R-F, Lee S-T, Lin Y-L, Hsu C-L, Lin S-W, Liou C-W, Lee J-D, Peng T-I, Lee T-H. 2020.
675 Risk Factors Associated with Outcomes of Recombinant Tissue Plasminogen Activator Therapy in
676 Patients with Acute Ischemic Stroke. *International Journal of Environmental Research and Public*
677 *Health* 17:618. DOI: 10.3390/ijerph17020618.
678 U.S. Centers for Medicare & Medicaid Services.ICD-9. *Available at*
679 *https://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html* (accessed
680 October 8, 2017a).
681 U.S. Centers for Medicare & Medicaid Services.ICD-10. *Available at*
682 *https://www.cms.gov/Medicare/Coding/ICD10/index.html* (accessed October 8, 2017b).
683 Wasey JO, Lang M. 2020.icd: Comorbidity Calculations and Tools for ICD-9 and ICD-10 Codes.

Available at https://cran.r-project.org/web/packages/icd/index.html (accessed July 28, 2020).

Weiskopf NG, Weng C. 2013. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association* 20:144–151. DOI: 10.1136/amiajnl-2011-000681.

Wickham H CW.devtools: Tools to make developing R code easier

Wu J, Roy J, Stewart WF. 2010. Prediction Modeling Using EHR Data: Challenges, Strategies, and a Comparison of Machine Learning Approaches. *Medical Care* 48.

Yoshida K BJ. Package 'tableone' for R.

Figure 1

Overview of the *dxpr* package

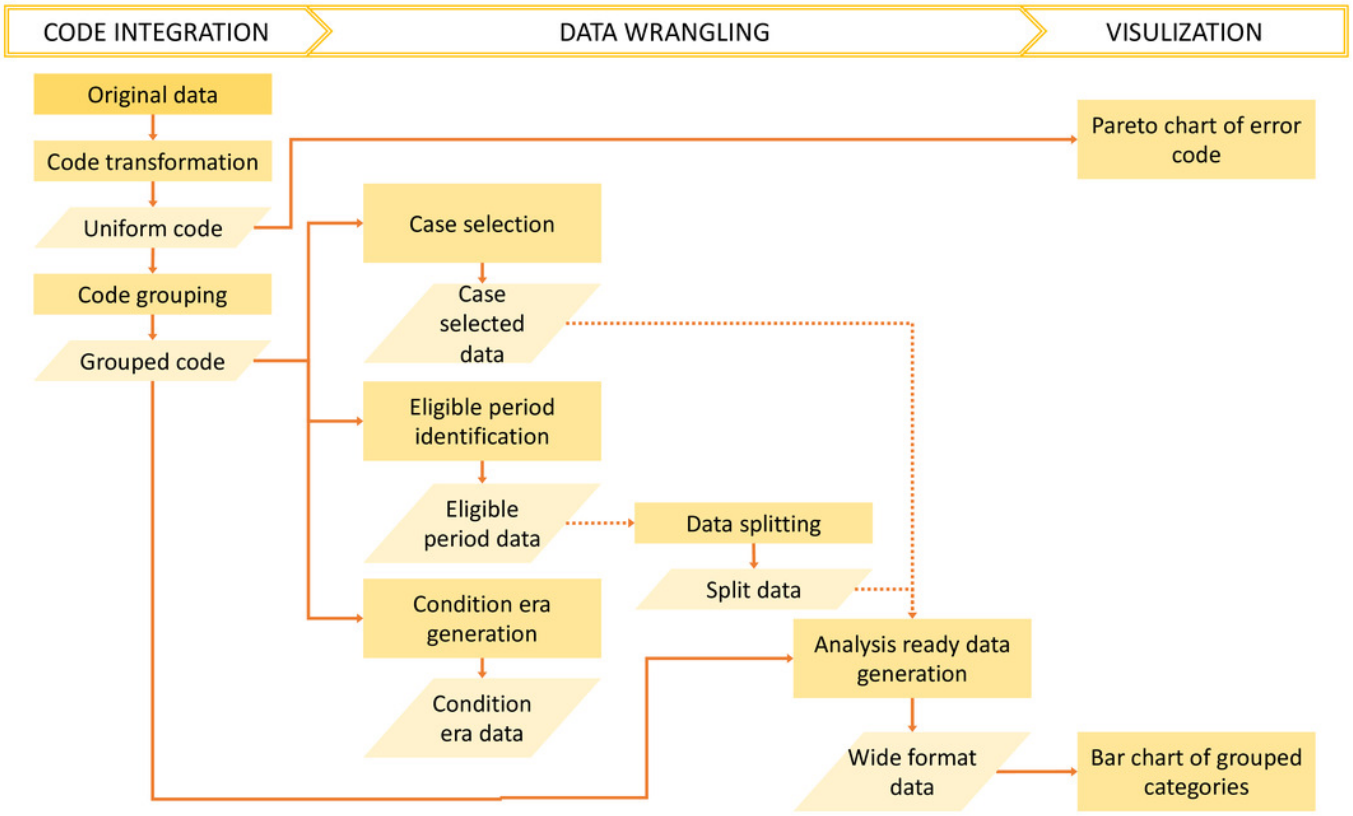

Figure 2

Bar chart to visualize the statistically significant difference of diagnostic categories between patients with and without PDA, grouped by the AHRQ-defined comorbidities.

PDA: patent ductus arteriosus; AHRQ: Agency for Healthcare Research and Quality; FluidsLytes: Fluid and electrolyte disorders; Valvular: valvular disease; CHF: congestive heart failure; HTN: hypertension, uncomplicated; Hypothyroid: hypothyroidism; NeuroOther: other neurological disorders; PHTN: pulmonary circulation disorders.

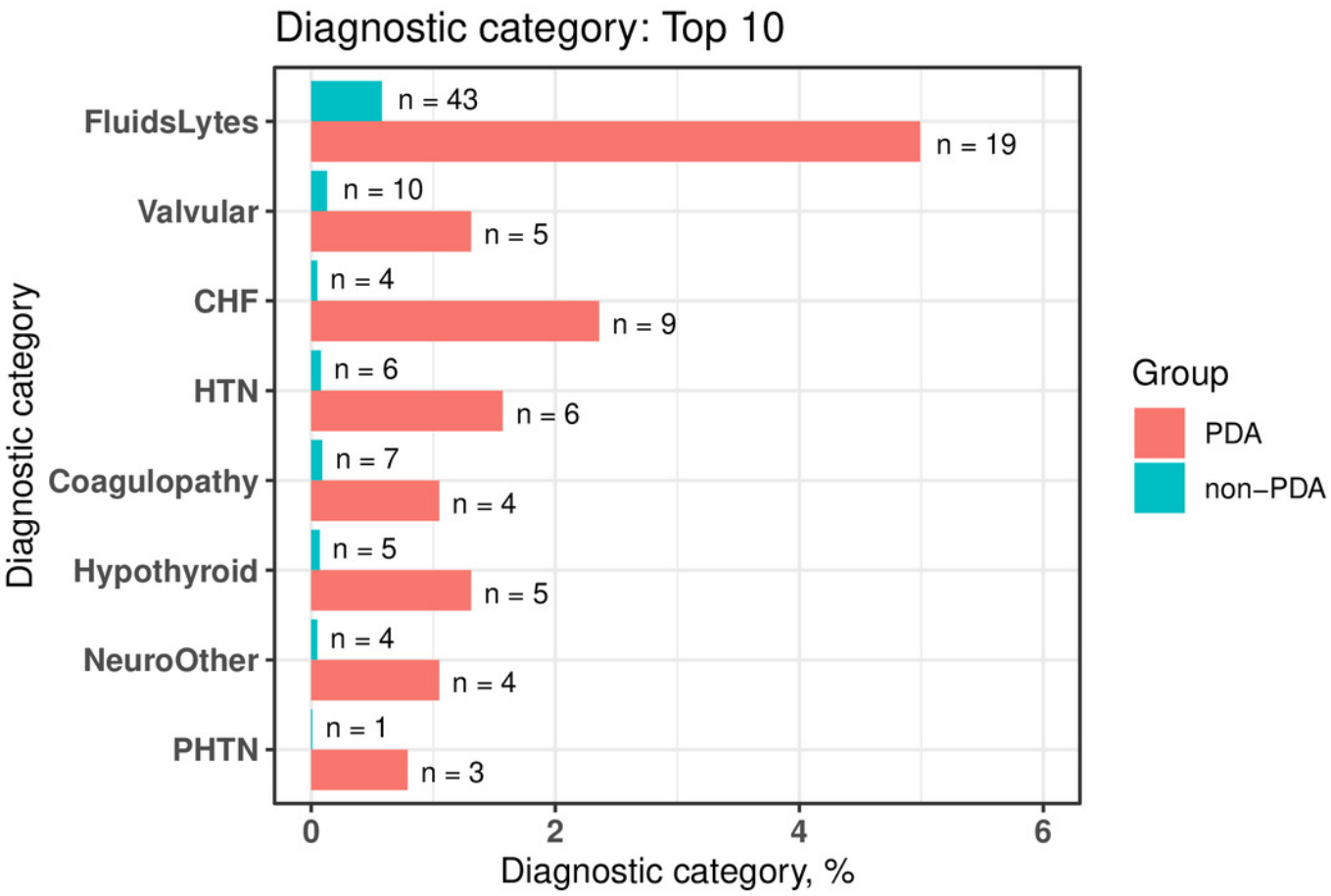

Table 1(on next page)

The first five diagnosis records of the sample dataset

1 **Table 1.** The first five diagnosis records of the sample dataset

ID	ICD	Date
A2	Z992	2020-05-22
A5	Z992	2020-01-24
A8	Z992	2015-10-27
A13	Z992	2020-04-26
A13	Z992	2025-02-02

2

Table 2(on next page)

Grouping results from grouping functions—icdDxToCCS

1 **Table 2.** Grouping results from grouping functions—icdDxToCCS

Short	ID	ICD	date	ccs_categories_description
78550	A	78550	2014/10/01	Shock
78552	A	78552	2013/10/01	Shock
78559	A	785.59	2013/09/01	Shock
78552	B	78552	2013/09/01	Shock
25000	B	250.00	2012/07/01	Diabetes mellitus without complication
25000	B	250.00	2012/05/01	Diabetes mellitus without complication

2

3

Table 3(on next page)

Summarized results from grouping functions—icdDxToCCS

1 **Table 3.** Summarized results from grouping functions—icdDxToCCS

ID	categories	firstCaseDate	endCaseDate	count	period
A	Shock	2013/09/01	2014/10/01	3	395 days
B	Diabetes mellitus without complication	2012/05/01	2012/05/01	2	62 days
B	Shock	2013/09/01	2013/09/01	1	0 days

2

3

Table 4(on next page)

Summary of AHRQ-defined comorbidities based on the tableone package using the integrated data generated by the *dxpr* package

Table 4. Summary of AHRQ-defined comorbidities based on the tableone package using the integrated data generated by the *dxpr* package

AHRQ comorbidities	Non-PDA	PDA[a]	p
n	7452	381	
Coagulopathy (%)	7 (0.1)	4 (1.0)	<0.001
Congestive heart failure (%)	4 (0.1)	9 (2.4)	<0.001
Deficiency anemias (%)	2 (0.0)	1 (0.3)	0.342
Depression (%)	1 (0.0)	0 (0.0)	1
Diabetes, complicated (%)	2 (0.0)	0 (0.0)	1
Fluid and electrolye disorders (%)	43 (0.6)	19 (5.0)	<0.001
Hypertension, complicated (%)	2 (0.0)	0 (0.0)	1
Hypertension, uncomplicated (%)	6 (0.1)	6 (1.6)	<0.001
Hypothyroidism (%)	5 (0.1)	5 (1.3)	<0.001
Other neurological disorders (%)	4 (0.1)	4 (1.0)	<0.001
Peripheral vascular disorders (%)	1 (0.0)	0 (0.0)	1
Pulmonary circulation disorders (%)	1 (0.0)	3 (0.8)	<0.001
Renal failure (%)	1 (0.0)	0 (0.0)	1
Solid tumor without metastasis (%)	1 (0.0)	0 (0.0)	1
Valvular disease (%)	10 (0.1)	5 (1.3)	<0.001
Weight loss (%)	2 (0.0)	0 (0.0)	1

[a] PDA: patent ductus arteriosus

AHRQ: Agency for Healthcare Research and Quality

Table 5(on next page)

Functions in the *dxpr* package

1 **Table 5.** Functions in the *dxpr* package

Functions	Descriptions
I. Data integration	
icdDxShortToDecimal	Transform ICD[a] diagnostic codes into decimal format
icdDxDecimalToShort	Transform ICD diagnostic codes into short format
icdDxToCCS	Group ICD diagnostic codes into single CCS[b] category
icdDxToCCSLvl	Group ICD diagnostic codes into multiple CCS category
icdDxToComorbid	Group ICD diagnostic codes into comorbidity category (Elixhauser, Charlson, and AHRQ)
icdDxToPheWAS	Group ICD diagnostic codes into PheWAS[c] category
icdDxToCustom	Group ICD diagnostic codes into customized grouping category based on precise method
icdDxToCustomGrep	Group ICD diagnostic codes into customized grouping category based on fuzzy method
II. Data Wrangling	
selectCases	Query matching cases in the EHR[d] data
splitDataByDate	Query data by a clinical event
patientRecordDate	Query the earliest/latest admission date for each patient.
getConditionEra	Calculate condition era by grouped categories of each patient.
groupedDataLongToWide	Convert long format of grouped data into wide format

	for analytical and plotting functions
III. Visualization	
plotICDError	Pareto chart of error ICD list
plotDiagCat	Bar chart of diagnostic categories
Procedure	
icdPrToCCS	Group ICD procedure codes into single CCS category
icdPrToCCSLvl	Group ICD procedure codes into multiple CCS category
icdPrToProcedureClass	Group ICD procedure codes into procedure class category

[a]ICD, International Classification of Diseases; [b]CCS, Clinical Classifications Software; [c]PheWAS, Phenome Wide Association Studies; and [d]EHR, Electronic Health Record.