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 [3] , Chun Ju Chen [3, 4]

[1] Department of Information Management, National Central University, Taoyuan, Taiwan

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

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

[4] 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 data processes 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 address the common bottleneck caused by clinical data characteristics such as heterogeneity and sparseness.

1   *dxpr*: **An R package for generating analysis-ready data from**

2   **electronic health records—diagnoses and procedures**

4   Yi-Ju Tseng[1,2,*], Hsiang-Ju Chiu[3#], Chun Ju Chen[3,4#]

5   [1] Department of Information Management, National Central University, Taoyuan, Taiwan

6   [2] Department of Laboratory Medicine, Chang Gung Memorial Hospital at Linkou, Taoyuan,

7   Taiwan

8   [3] Department of Information Management, Chang Gung University, Taoyuan, Taiwan

9   [4] Department of Information Management, National Taiwan University, Taipei, Taiwan

11   **Short Title**: An R package for EHR processing

12   ***Corresponding Author***

13   Yi-Ju Tseng, Ph.D.

14   Department of Information Management, National Central University

15   300, Zhongda Rd., Zhongli District, Taoyuan City, Taiwan 320317

16   Yjtseng.info@gmail.com

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

21   **Abstract**

22   **Background**

23   Enriched electronic health records (EHRs) contain crucial information related to disease

24   progression, and this information can help with decision-making in the health care field. Data

25   analytics in health care is deemed as one of the essential processes that help accelerate the

26   progress of clinical research. However, processing and analyzing EHR data are common

27   bottlenecks in health care data analytics.

28   **Methods**

29   The *dxpr* R package provides mechanisms for integration, wrangling, and visualization of

30   clinical data, including diagnosis and procedure records. First, the *dxpr* package helps users

31   transform International Classification of Diseases (ICD) codes to a uniform format. After code

32   format transformation, the *dxpr* package supports four strategies for grouping clinical diagnostic

33   data. For clinical procedure data, two grouping methods can be chosen. After EHRs are

34   integrated, users can employ a set of flexible built-in querying functions for dividing data into

35   case and control groups by using specified criteria and splitting the data into before and after an

36   event based on the record date. Subsequently, the structure of integrated long data can be

37   converted into wide, analysis-ready data that are suitable for statistical analysis and visualization.

38   **Results**

39   We conducted comorbidity data processes based on a cohort of newborns from Medical

40   Information Mart for Intensive Care-III (n = 7,833) by using the *dxpr* package. We first defined

41   patent ductus arteriosus (PDA) cases as patients who had at least one PDA diagnosis (ICD, Ninth

42   Revision, Clinical Modification [ICD-9-CM] 7470*). Controls were defined as patients who

43   never had PDA diagnosis. In total, 381 and 7,452 patients with and without PDA, respectively,

44   were included in our study population. Then, we grouped the diagnoses into defined

45   comorbidities. Finally, we observed a statistically significant difference in 8 of the 16

46   comorbidities among patients with and without PDA, including fluid and electrolyte disorders,

47   valvular disease, and others.

48   **Conclusions**

49   This *dxpr* package helps clinical data analysts address the common bottleneck caused by clinical

50   data characteristics such as heterogeneity and sparseness.

51

1   Introduction

52

53   On the basis of the development of electronic health records (EHRs), data analytics in

54 health care is deemed as an essential process for accelerating the progress of clinical research

55 (Hersh, 2007; Jensen, Jensen & Brunak, 2012; Miotto & Weng, 2015). Enriched EHRs contain

56 crucial information related to disease progression, and this information can help with decision

57 making in the health care field including for treatment selection and disease diagnosis (Jensen,

58 Jensen & Brunak, 2012; Raghupathi & Raghupathi, 2014). However, processing and analyzing

59 EHR data are usually challenging because of their heterogeneity and sparsity. These inherent

60 characteristics create a common bottleneck in health care big data analytics (Wu, Roy & Stewart,

61 2010; Hripcsak & Albers, 2013; Weiskopf & Weng, 2013). Moreover, executing clinical data

62 analysis project across different departments or institutes is difficult because clinical data formats

63 and terminologies used to describe clinical conditions may vary across departments. A method

64 that can standardize and facilitate the sharing of data or analysis pipelines from multiple sources

65 is needed in research on clinical data analysis. Several common data models (CDMs) have been

66 developed for eliminating clinical data format barriers, including the National Patient-Centered

67 Clinical Research Network (PCORnet) (Fleurence et al., 2014; PCORnet, 2020) and

68 Observational Medical Outcomes Partnership (OMOP) CDM (Observational Health Data

69 Sciences and Informatics, 2020). The concept of CDM is to transform data into a CDM and

70 terminology and then allow users to perform systematic analyses by using various sources.

71 Although a CDM can help perform systematic analyses across different sources, the integration

72 of clinical data and the preparation of analysis-ready data are unsolved issues.

73   The proposed open-source *dxpr* R package is a software tool aimed at expediting general

74 EHR or claims data analyses through incorporating several functions that enable users to

75 standardize, integrate, wrangle, and visualize clinical diagnosis and procedure records. Preparing

76   an analysis-ready dataset from EHRs or claims data is a complex task that requires both medical

77   knowledge and data science skills. The proposed *dxpr* package simplifies and accelerates the

78   workflow for EHR data extraction and helps clinical data analysts generate simple and clean

79   scripts that can easily be shared and reproduced. The *dxpr* package enables researchers to

80   explore EHRs or claims data to acquire crucial information, understand disease progression, and

81   analyze outcomes without writing complicated data preprocessing scripts. Moreover, the

82   proposed package can support collaborative research across multiple data sources as long as the

83   data include general diagnosis- or procedure-related information.

84   The *dxpr* package has three phases to process and analyze diagnosis codes in EHRs (Fig. 1).

85   In the first phase, namely data integration, we transform diagnosis codes into a uniform format

86   and provide four strategies to group diagnoses into clinically meaningful categories before the

87   wrangling process. In the second phase, namely, data wrangling, users can use provided

88   functions to query eligible cases, split data based on the index date, and calculate condition era

89   according to the grouped diagnostic categories of each patients. Furthermore, exploratory data

90   analysis preparation can be performed in this phase. Moreover, the *dxpr* package provides a

91   function to convert a long format of grouped data into a wide format, which fits other analytical

92   and plotting functions from other packages better. In the last phase, namely visualization, we

93   provide overviews for diagnosis standardization and data integration, such as comorbidity

94   distribution in the study population, comorbidity differences between case and control groups,

95   and the most common diagnoses that failed to be grouped or transformed. The usage details are

96   presented in the Supplementary Data S1 and S2. For processing and analyzing procedure codes,

97   the concept is similar to diagnosis.

98

99   **2   Materials and methods**

100

101 **Preparation**

102 The current version of the package is available at Github (https://github.com/DHLab-

103 TSENG/dxpr, Supplementary Data S3) and is accessible through the devtools package that

104 enables installing packages from GitHub (Wickham, Hester & Chang, 2020). To install the *dxpr*

105 R package, users can type the following commands in an R session:

106

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

109

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

111 ·   Member ID: a patient identifier, which can be numeric, alphanumeric, or a list of characters.

112 ·   Diagnosis/procedure code: ICD-9 or ICD-10 code assigned to a visit or an admission.

113 ·   Visit or admission date: the date of the visit, admission, or clinical service provided. The

114   date should be in date format. If the date is recorded in a string format, it has to be recorded

115   in year–month–day format (YYYY/MM/DD or YYYY-MM-DD).

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

117 The data can be imported from files or databases, with packages provide access to databases

118 within R, such as DBI (R Special Interest Group on Databases (R-SIG-DB), Wickham & Müller,

119 2021) and odbc (Hester & Wickham, 2021). We illustrate the use of the *dxpr* package with a

120 diagnostic sample dataset of 10-year admissions of 38 patients, sampleDxFile, and the first five

121 records are shown in Table 1.

122

123   **Data integration**

124   **Code format transformation**

125   The *dxpr* package first transforms ICD diagnostic codes into a uniform format before code

126   grouping. ICD-9 and ICD-10 diagnostic codes (U.S. Centers for Medicare & Medicaid Services,

127   b) have two formats, namely decimal (with a decimal place separating the code) and short

128   formats. Different hospitals, grouping methods, or standards coded ICD into different formats.

129   For example, studies using Clinical Classifications Software (CCS) (Healthcare Cost and

130   Utilization Project (HCUP), 2017, 2019a) and comorbidity measures, such as Elixhauser and

131   Charlson (Elixhauser et al., 1998; Menendez et al., 2014; Moore et al., 2017), have coded the

132   ICD in a short format, and a phenome-wide association study (PheWAS) (Denny et al., 2010)

133   coded the ICD in a decimal format. Therefore, format transformation is required before code

134   grouping, and the transformation type is decided by the chosen grouping method.

135   The transformation function (icdDxShortToDecimal) converts ICD-9 and ICD-10 codes into

136   a uniform decimal format because a decimal format is needed for grouping diagnostic codes in

137   PheWAS classification. Similar to icdDxShortToDecimal, icdDxDecimalToShort function converts

138   diagnostic codes into a uniform short format, which can be used for grouping to CCS,

139   Elixhauser, or other classifications. These transformative functions not only convert ICD codes

140   into uniform format codes but also check for potential coding errors. We provide two types of

141   warning messages: wrong ICD format and wrong ICD version. Additional suggestions are

142   generated to help users adjust potential incorrect ICD codes if available.

143

```
144    ICD_Decimal <- icdDxShortToDecimal(dxDataFile = sampleDxFile,
145                                       icdColName = ICD,
146                                       dateColName = Date,
147                                       icd10usingDate = "2015/10/01")
```

```
148  sampleDxFile$Decimal <- ICD_Decimal$ICD
149  head(sampleDxFile)
150      ID  ICD        Date Decimal
151  1:  A2 Z992 2020-05-22  Z99.2
152  2:  A5 Z992 2020-01-24  Z99.2
153  3:  A8 Z992 2015-10-27  Z99.2
154  4: A13 Z992 2020-04-26  Z99.2
155  5: A13 Z992 2025-02-02  Z99.2
156  6: A15 Z992 2023-05-12  Z99.2
157  tail(ICD_Decimal$Error)
158       ICD count IcdVersionInFile    WrongType Suggestion
159  1:  75.52    4            ICD  9  Wrong format
160  2:  E03.0    4            ICD  9 Wrong version
161  3:    650    4            ICD 10 Wrong version
162  4: 123.45    3            ICD 10  Wrong format
163  5:  755.2    3            ICD  9  Wrong format     755.29
164  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

179   (icdDxToCustom) and searching for lines containing a match (icdDxToCustomGrep). The

180   grouping functions return two tables of the dataset, one is data with the corresponding grouping

181   categories of each ICD (Table 2), and the other is summarized data exhibiting the earliest/latest

182   record date and diagnosis counts in the same grouping category for each patient (Table 3). For

183   example, after executing function icdDxToCCS for the records of patients A and B, two output

184   types are shown in Tables 2 and 3, respectively. Patient A has three diagnosis records (ICD

185   codes: 78550, 78552, and 785.59), which are all in the "shock" category of the CCS

186   classification, with the earliest record on September 1, 2013 and the latest one on October 1,

187   2014. The icdDxToCCS function mapped corresponding CCS categories for these ICD codes and

188   returned the grouping results (Table 2). Similarly, patient B has two diagnosis records (ICD

189   codes: 78552 and 250.00) in the "shock" category and "Diabetes mellitus without complication"

190   category of CCS classification, and the grouping results are also shown in Table 2. According to

191   these diagnosis records shown in Table 2, Table 3 shows that icdDxToCCS function can

192   summarize the first and last dates of diagnosis, the total number of diagnoses, and the period

193   between the first and last diagnoses for each category, which can be used for designing the

194   analysis strategy. While icdDxToCCS groups codes into single-level CCS, icdDxToCCSLvl groups

195   codes into multi-level CCS. Multi-level CCS expands single-level CCS into a four-level

196   hierarchical system for diagnoses, which provide the opportunity to examine general

197   aggregations or to assess specific conditions ("HCUP-US Tools & Software Page"). For

198   instance, if a user wishes to group codes into the second level of multi-level CCS, then this task

199   can be performed through simply entering "ccslvl2" as the assigned grouping type. These

200   grouping functions not only facilitate users to convert original diagnosis records from detailed

201   levels into clinically meaningful diagnostic groups for further analysis but also provide

202   aggregated information of each diagnostic group that can help research design and hypothesis

203   generation, such as filtering out data based on specified criteria (e.g., first diagnosis dates of

204   specific chronic disease).

205

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

207

208
```
## 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
```

232

Data wrangling

Case selection

235   In clinical data analysis projects, the most crucial step is case definition and selection, such

236   as defining Lyme disease cases from claims data (Tseng et al., 2015) or defining acute ischemic

237   stroke from EHR (Tseng et al., 2020). The analysis results could change based on case definition

238   and lead to a different conclusion. The query function selectCases can select cases matching case

239   definitions. Users can select cases based on diagnosis (ICD) or diagnostic categories (CCS,

240   PheWAS, comorbidities, or self-defined diagnostic categories). Moreover, the function provides

241   an option to set the minimum number of diagnoses within a specific duration. For example, users

242   can extract diabetes cases by assigning at least two diagnoses in ICD codes "250.xx" or "E10.x-

243   E14.x" within 730 days when a user applies the validated diabetes case definition: "two

244   physician claims within 2 years with diagnosis codes 250.xx or E10.x-E14.x" (Chen et al., 2010).

245   The output dataset of this function provides the start and end dates of the cases, the number of

246   days between them, and the most common ICD codes used in the case definition. Furthermore, a

247   list of people who did not satisfy the required case conditions or practically match the case

248   definition is appended in the returned output table, and these individuals can be defined as a

249   control group or be removed.

250

```
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
```

```
271   1: D3 non-Selected    NA       <NA>      <NA> NA days      <NA>           NA
272   2: D4 non-Selected    NA       <NA>      <NA> NA days      <NA>           NA
273   3: D5 non-Selected    NA       <NA>      <NA> NA days      <NA>           NA
274   4: D6 non-Selected    NA       <NA>      <NA> NA days      <NA>           NA
275   5: D7 non-Selected    NA       <NA>      <NA> NA days      <NA>           NA
276   6: D8 non-Selected    NA       <NA>      <NA> NA days      <NA>           NA
```

277

278 **Eligible period identification**

279   In some clinical data, such as claims data, individuals can join or leave the program on

280 different dates, and the length of available records might affect the analysis completeness. The

281 *dxpr* package provides a function getEligiblePeriod for researchers to identify the first/last record

282 date for each patient. These outputs can be used as an index date for case exclusion, such as

283 cases without at least 6 months washout or follow-up period, or further data splitting.

284

```
285   admissionDate <- getEligiblePeriod(dxDataFile = sampleDxFile,
286                               idColName = ID,
287                               dateColName = Date)
288   head(admissionDate)
289       ID firstRecordDate endRecordDate
290   1:  D6      2005-10-09    2025-01-05
291   2: A12      2006-01-12    2022-06-12
292   3:  D1      2006-02-12    2024-04-04
293   4: A13      2006-04-29    2025-02-02
294   5:  A9      2006-06-30    2023-12-10
295   6:  D2      2006-09-01    2025-08-11
```

296

297 **Data splitting based on index date and moving window**

298   In clinical data analysis projects, users usually need to extract data based on a specific

299 clinical event (e.g., extracting data before the first Lyme disease diagnosis in the records (Tseng

300 et al., 2017)). The date of the specific event (index date) can be the first/last record date of the

301 events or patient record, and the table of the index date for each individual can be generated

302 using selectCases or getEligiblePeriod function, respectively. The *dxpr* package provides a

303 convenient function splitDataByDate that can split data through classifying the data recorded

304 before or after the defined index date and calculating the period between the record date and

305 index date based on a self-defined window. For example, if a user needs to aggregate the data by

306 using a 30-day window, the data recorded on 15 and 45 days after the index date will be defined

307 as window 1 and window 2, respectively. The output of splitDataByDate function helps users to

308 split the data based on the study design, and this can be applied to further time-series multiple-

309 measurement analysis with period information.

310

```
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
```

328

329 **Condition era generation**

330   Condition era is a means to apply consistent rules for medical conditions to infer distinct

331 episodes in care, generated through integrating distributed clinical records into a single

332 progression record (Ryan, 2010). The concept of condition era is committed to the length of the

333   persistence gap: when the time interval of any two consecutive admissions for certain conditions

334   is smaller than the length of the persistence gap, then these two admission events will be

335   aggregated into the same condition era. Each condition era consists of one or many events, and

336   differences between any two consecutive admission events are all within the persistence gap. For

337   example, an episode of influenza may include single or multiple outpatient visits, and the length

338   of the influenza course should be the period between the first and last visits of the episode.

339   getConditionEra function calculates condition era by using the grouped categories or self-

340   defining groups of each patient and then generates a table with individual IDs, the first and last

341   record of an era, and the sequence number of each episode. Users can easily convert scattered

342   diagnoses into an episode of condition based on the chararistics of target disease progression

343   with the proposed function.

344

```
345    Era <- getConditionEra(dxDataFile = sampleDxFile,
346                           idColName = ID,
347                           icdColName = ICD,
348                           dateColName = Date,
349                           icd10usingDate = "2015/10/01",
350                           groupDataType = CCS,
351                           isDescription = FALSE,
352                           gapDate = 360)
353    head(Era)
354       ID CCS_CATEGORY era firstCaseDate endCaseDate count   period
355    1: A0          158   1    2009-07-25  2010-10-29    3 461 days
356    2: A0          158   2    2012-04-05  2012-04-05    1   0 days
357    3: A0          158   3    2013-12-20  2013-12-20    1   0 days
358    4: A1          158   1    2006-11-29  2006-11-29    1   0 days
359    5: A1          158   2    2008-06-25  2008-06-25    1   0 days
360    6: A1          158   3    2012-06-19  2013-04-28    2 313 days
```

361   **Analysis-ready data generation**

362   After data integration and wrangling, researchers often need to further analyze these

363   processed data, and function groupedDataLongToWide converts the long format of grouped data

364   into a wide format, which is fit for other analytical and plotting packages, such as tableone

365   (Yoshida & Bartel, 2020) package.

366

```
367   CHARLSON <- icdDxToComorbid(dxDataFile = sampleDxFile,
368                               idColName = ID,
369                               icdColName = ICD,
370                               dateColName = Date,
371                               icd10usingDate = "2015-10-01",
372                               comorbidMethod = CHARLSON)
373   groupedData_Wide <- groupedDataLongToWide(dxDataFile  = CHARLSON$groupedDT,
374                                   idColName = ID,
375                                   categoryColName = Comorbidity,
376                                   dateColName = Date,
377                                   reDup = TRUE,
378                                   numericOrBinary = B,
379                                   count = 2)
380   head(groupedData_Wide, 5)
381      ID CANCER  CEVD  COPD DIAB_C  MSLD  PARA   PUD   PVD   RD Rheum
382   1  A0  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
383   2  A1  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
384   3 A10  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
385   4 A11  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
386   5 A12  FALSE FALSE FALSE  FALSE FALSE FALSE FALSE FALSE TRUE FALSE
```

387

388   **Visualization**

389   **Pareto chart of error ICD**

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

391   diagnosis codes with potential errors. Function plotICDError visualizes codes with potential error

392   by using the Pareto chart containing a bar plot where error ICD codes are arranged in descending

393   order, and the cumulative total is represented by the line. Users can sort based on the counts of

394   error ICD codes and set the top selected number of the ordered dataset. For instance, if a user

395   chooses the top 10 ordinal rankings, then the Pareto chart shows a plot of the top 10 common

396   error ICD codes and a list with details of these 10 and other error ICD codes.

397

```
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
```

422

Bar chart of diagnostic categories

424   Function plotDiagCat provides an overview of the grouping categories of the diagnoses and

425 summarizes the proportion of individuals diagnosed with grouped diagnostic categories in the

426 whole study population or case and control groups in a bar chart. Users can observe the number

427 and percentage of diagnostic categories in their dataset through this function. Furthermore, this

428 function compares the usage of significantly different diagnostic categories between case and

429 control groups by using the chi-square test or Fisher's exact test when the data does not match

430 the assumptions of the chi-square test. The default level of statistical significance is considered at

431 5% (p = 0.05). Researchers can set a threshold of the top N significant grouped categories and

432   the minimum prevalence of the diagnostic groups in the case or control group.

433   The "percentage" column shows the proportion of individuals diagnosed with the diagnostic

434   category in the group. For example, there are 38 patients in the sample file, and "Renal Failure"

435   defined in Elixhauser comorbidity accounts for 63.16% of the population (24/38).

436

```
437  ELIX <- icdDxToComorbid(dxDataFile  = sampleDxFile,
438                          idColName = ID,
439                          icdColName = ICD,
440                          dateColName = Date,
441                          icd10usingDate = "2015-10-01",
442                          comorbidMethod = ELIX)
443  groupedDataWide <- groupedDataLongToWide(dxDataFile  = ELIX$groupedDT,
444                                  idColName = ID,
445                                  categoryColName = Comorbidity,
446                                  dateColName = Date,
447                                  reDup = TRUE,
448                                  numericOrBinary = B)
449  plot2 <- plotDiagCat(groupedDataWide = groupedDataWide,
450                       idColName = ID,
451                          topN = 10,
452                       limitFreq = 0.01)
453
454  plot2$sigCate
455      DiagnosticCategory  N Percentage
456   1:          RENLFAIL 24     63.16%
457   2:             TUMOR  6     15.79%
458   3:              ARTH  5     13.16%
459   4:             LYMPH  4     10.53%
460   5:             PSYCH  4     10.53%
461   6:              DRUG  3      7.89%
462   7:             NEURO  3      7.89%
463   8:              PARA  2      5.26%
464   9:          PERIVASC  2      5.26%
465  10:             VALVE  2      5.26%
```

466

467 **Clinical procedure data processing**

468   As diagnosis codes, ICD-9-Procedure Coding System (PCS) code also has two formats,

469   namely decimal and short, whereas ICD-10-PCS code only has a short format. The functions

470   (icdPrToCCS and icdPrToProcedureClass) provide two strategies (CCS and procedure class) to

471   collapse ICD procedure codes into clinically meaningful categories for further analysis. This

472   procedure has two CCS classifications: single and multiple levels. The usage is similar to the

473   diagnostic CCS classification. A sample file (samplePrFile) is provided with procedure records,

474   including three patients and 170 records.

475   The procedure classes (Healthcare Cost and Utilization Project (HCUP), 2016) are created

476   to facilitate health services research on hospital procedures by using administrative data. The

477   procedure classes provide a standard to categorize individual procedure codes into one of the

478   four broad categories: minor diagnostic, minor therapeutic, major diagnostic, and major

479   therapeutic. The aforementioned classification functions mentioned allow the researcher to

480   readily determine whether a procedure is diagnostic or therapeutic and whether a procedure is

481   minor or major in terms of invasiveness, resource use, or both.

482

483   **Use case**

484   To illustrate the main features in the *dxpr* package and the typical workflow, we

485   demonstrated an analysis using the package among newborns who were diagnosed with patent

486   ductus arteriosus (PDA) from Medical Information Mart for Intensive Care-III (MIMIC-III)

487   (Johnson et al., 2016). MMIC-III is a publicly available database comprising deidentified health-

488   related data associated with the admissions of approximately 60,000 patients who stayed in the

489   critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012.

490   We provided a sample file sampleFile_MIMIC obtained from MIMIC-III (Johnson et al.,

491   2016), a medical dataset of 7,833 newborn patients with 45,674 admissions. This dataset is used

492   for verifying the comorbidity difference between patients with and without PDA based on the

493   *dxpr* package. In this example, we defined PDA cases as patients who had at least one PDA

494   diagnosis (ICD-9-CM 7470*). The controls are defined as patients who never had PDA

495   diagnosis.

496

497   **Performance analysis**

498   The *dxpr* package is designated to accelerate the process of large EHR data integration and

499   provide the ready-for-analysis dataset from the integrated EHR data. We verified the running

500   time 100 times with a simulated dataset of 953,294 unique patients and 7,948,418 distinct

501   diagnosis records in a standard personal computer with 64 GB DDR4 2133GHz RAM and an

502   Intel® Core™ i7-6700 (CPU @3.40GHz), using Windows 10 (1809), R 4.0.1 (64 bits), and

503   RStudio 1.2.5033.

504

505   **3   Result**

506   **A use case — patients with PDA**

507   We conducted comorbidity analyses based on a cohort of newborns from MIMIC-III (n =

508   7,833) by using *dxpr* and tableone (Yoshida & Bartel, 2020) packages. In the *dxpr* package, we

509   first use selectCases function to define case (PDA) and control (non-PDA) groups. In total, 381

510   and 7,452 patients with and without PAD were included in our study, respectively. Then,

511   icdDxToComorbid function was applied to group diagnoses into AHRQ-defined comorbidities.

512   Finally, we analyzed and graphed the AHRQ-defined comorbidities based on plot_groupedData

513   function (Fig. 2) by using the chi-square test and Fisher's exact test. To focus on comorbidities

514   that were essential and recorded in adequate individuals in our study population, we excluded

515   comorbidities recorded in <1% of the patients in the PDA or non-PDA group. The analysis-ready

516   data generated by groupedDataLongToWide can be passed to the tableone (Yoshida & Bartel,

517   2020) package to create objects summarizing all comorbidities stratified by patients with and

518   without PDA and by performing the statistical chi-square tests. The AHRQ comorbidity table

519   revealed 8 of the 16 statistically significant comorbidities (p < 0.05, Table 4) among patients

520   with and without PDA, and the comorbidities are visualized in Fig. 2.

521

522   **Performance**

523   For a simulated dataset of 953,294 unique patients and 7,948,418 admission records, code

524   grouping with CCS-defined comorbidities required 149±2.48 seconds (including code

525   transformation). Case selection required 238±3.05 seconds to query patients with diseases of the

526   urinary system, eligible period identification required 1.12±0.22 seconds to find the first and last

527   admission date for each patient, data splitting with the first admission date for each patient

528   required 6.50±0.42 seconds, condition era generation required 372±6.39 seconds, and analysis-

529   ready data generation required 3.75±0.27 seconds.

530

531   **4   Discussion and conclusions**

532   The *dxpr* package considerably simplifies the extraction, accelerates the processing of

533   clinical data research, and enables researchers to prepare analysis-ready data with a standard

534   workflow. The package had been developed and tested using structured clinical data, such as

535   critical care data (MIMIC-III (Johnson et al., 2016)), a multi-institutional medical care database

536   (Chang Gung Research Database (Tsai et al., 2017; Tseng et al., 2020)), and claims data

537   (National Health Insurance Research Database (Hsieh et al., 2019)), indicating that the package

538   can be applied to data from different countries, institutions, and data structures. The available

539   functions are summarized in Table 5.

540   Several software and packages were developed to facilitate clinical data analysis. rEHR

541   (Springate et al., 2017) established a clinical data analysis workflow to simplify the processing of

542   EHR. The rEHR package simplifies the process of extracting data from EHR databases. It used

543   the database backend that can accelerate data access and process times. However, this design

544   needs database backend, which might not be suitable in many circumstances. Furthermore, the

545   international diagnosis coding standard, such as ICD, were not used in the package. The ICD

546   (Wasey & Lang, 2020) package is designed for calculating comorbidities and medical risk scores

547   with ICD-9 and ICD-10 codes. It is helpful to group ICD codes according to comorbidities.

548   However, in clinical data analysis, eligible case selection, data split based on the defined index

549   date, and visualization are also essential. Therefore, we designed and developed the *dxpr*

550   package to facilitate diagnosis data analysis.

551   The proposed package has limitations, which come from either the data or package itself.

552   For analyzing clinical data, the *dxpr* package highly depends on diagnosis and procedure codes,

553   but these codes may vary in accuracy across different institutions. Furthermore, the effect of

554   switching diagnosis codes from ICD-9 to ICD-10 should be considered if the analysis period is

555   across the switching date. In addition to diagnosis and procedure data, the other data not included

556   in proposed packages, such as medication data, are important in clinical data analysis projects. In

557   the R ecosystem, the AdhereR (Dima & Dediu, 2017) package implements a set of functions that

558   are consistent with current adherence guidelines and definitions. Fourth, we provide an easy-to-

559   use package that will help analysts process raw data and notify them when potential coding

560   errors exist. However, even with this package, analysts should understand their data precisely.

561   This easy-to-use package will help analysts process clinical data with its coding error–checking

562   functions, but may also lead naïve analysts to miss opportunities to find other errors in the data.

563   Finally, the *dxpr* package is focused on analysis-ready data generation so that the statistic

564   method incorporation may be insufficient. However, the R ecosystem's most significant

565 advantage is that many well-developed packages were developed to facilitate statistical analysis.

566 In the use case demonstration, our package can be used with other packages, such as tableone

567 package. The tableone (Yoshida & Bartel, 2020) package is developed to ease the construction of

568 the common "Table 1" in research papers, providing patient baseline characteristics table with

569 summary statistics and hypothesis tests.

570   We demonstrated that the *dxpr* package can play an essential role in complex clinical data

571 preprocessing and analysis-ready data generation through integrating the international standard

572 of clinical data. This package helps clinical data analysts combat the common bottleneck caused

573 by certain clinical data characteristics, such as heterogeneity and sparseness.

574

575 **Acknowledgements**

576 We thank Ru-Fang Hu, Yi-An Zhu, and Chia-Wei Chang, from Department of Information

577 Management, Chang Gung University, for testing the *dxpr* package. This manuscript was edited

578 by Wallace Academic Editing.

579

580 **References**

581 Chen G, Khan N, Walker R, Quan H. 2010. Validating ICD coding algorithms for diabetes mellitus from

582   administrative data. *Diabetes research and clinical practice* 89:189–195.

583 Denny JC, Ritchie MD, Basford MA, Pulley JM, Bastarache L, Brown-Gentry K, Wang D, Masys DR,

584   Roden DM, Crawford DC. 2010. PheWAS: demonstrating the feasibility of a phenome-wide scan to

585   discover gene-disease associations. *Bioinformatics* 26:1205–1210.

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

587   histories in R with AdhereR: Towards transparent and reproducible use of electronic healthcare data.

588   *PLOS ONE* 12:e0174426. DOI: 10.1371/journal.pone.0174426.

589   Elixhauser A, Steiner C, Harris DR, Coffey RM. 1998. Comorbidity Measures for Use with

590   Administrative Data. *Medical Care* 36:8–27. DOI: 10.1097/00005650-199801000-00004.

591   Fleurence RL, Curtis LH, Califf RM, Platt R, Selby J V., Brown JS. 2014. Launching PCORnet, a

592   national patient-centered clinical research network. *Journal of the American Medical Informatics*

593   *Association* 21:578–582. DOI: 10.1136/amiajnl-2014-002747.

594   HCUP-US Tools & Software Page.

595   Healthcare Cost and Utilization Project (HCUP). 2016.HCUP Procedure Classes. *Available at*

596   *https://www.hcup-us.ahrq.gov/toolssoftware/procedure/procedure.jsp*

597   Healthcare Cost and Utilization Project (HCUP). 2017.Clinical Classifications Software (CCS) for ICD-

598   9-CM. *Available at https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp* (accessed October 8,

599   2017).

600   Healthcare Cost and Utilization Project (HCUP). 2019a.Beta Elixhauser Comorbidity Software for ICD-

601   10-CM

602   Healthcare Cost and Utilization Project (HCUP). 2019b.Beta Clinical Classifications Software (CCS) for

603   ICD-10-CM/PCS

604   Hersh WR. 2007. Adding value to the electronic health record through secondary use of data for quality

605   assurance, research, and surveillance. *American Journal of Managed Care* 13:277–278. DOI:

606   10.1001/jama.229.4.457.

607   Hester J, Wickham H. 2021.odbc: Connect to ODBC Compatible Databases (using the DBI Interface).

608   *Available at https://cran.r-project.org/package=odbc* (accessed February 23, 2021).

609   Hripcsak G, Albers DJ. 2013. Next-generation phenotyping of electronic health records. *Journal of the*

610   *American Medical Informatics Association* 20:117–121. DOI: 10.1136/amiajnl-2012-001145.

611   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

612   Health Insurance Research Database: past and future. *Clinical Epidemiology* Volume 11:349–358.

613   DOI: 10.2147/CLEP.S196293.

614   Jensen PB, Jensen LJ, Brunak S. 2012. Mining electronic health records: towards better research

615   applications and clinical care. *Nature Reviews Genetics* 13:395–405. DOI: 10.1038/nrg3208.

616   Johnson AEW, Pollard TJ, Shen L, Lehman LH, Feng M, Ghassemi M, Moody B, Szolovits P, Anthony

617   Celi L, Mark RG. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*

618   3:160035. DOI: 10.1038/sdata.2016.35.

619   Menendez ME, Neuhaus V, Van Dijk CN, Ring D. 2014. The Elixhauser comorbidity method

620   outperforms the Charlson index in predicting inpatient death after orthopaedic surgery. *Clinical*

621   *Orthopaedics and Related Research* 472:2878–2886. DOI: 10.1007/s11999-014-3686-7.

622   Miotto R, Weng C. 2015. Case-based reasoning using electronic health records efficiently identifies

623   eligible patients for clinical trials. *Journal of the American Medical Informatics Association*

624   22:e141–e150. DOI: 10.1093/jamia/ocu050.

625   Moore BJ, White S, Washington R, Coenen N, Elixhauser A. 2017. Identifying Increased Risk of

626   Readmission and In-hospital Mortality Using Hospital Administrative Data. *Medical Care* 55:698–

627   705. DOI: 10.1097/MLR.0000000000000735.

628   Observational Health Data Sciences and Informatics. 2020.OMOP Common Data Model. *Available at*

629   *https://www.ohdsi.org/data-standardization/the-common-data-model/* (accessed July 28, 2020).

630   PCORnet. 2020.PCORnet Common Data Model (CDM). *Available at https://pcornet.org/data-driven-*

631   *common-model/* (accessed July 28, 2020).

632   R Special Interest Group on Databases (R-SIG-DB), Wickham H, Müller K. 2021.DBI: R Database

633   Interface. *Available at https://cran.r-project.org/package=DBI* (accessed February 23, 2021).

634   Raghupathi W, Raghupathi V. 2014. Big data analytics in healthcare: promise and potential. *Health*

635   *Information Science and Systems* 2:3. DOI: 10.1186/2047-2501-2-3.

636   Ryan P. 2010. Establishing a Condition Era Persistence Window for Active Surveillance. *Foundation for*

637   *the National Institutes of Health*:1–13.

638   Springate DA, Parisi R, Olier I, Reeves D, Kontopantelis E. 2017. rEHR: An R package for manipulating

639   and analysing electronic health record data. *PLoS ONE* 12:e0171784. DOI:

640   10.1371/journal.pone.0171784.

641   Tsai MS, Lin MH, Lee CP, Yang YH, Chen WC, Chang GH, Tsai Y Te, Chen PC, Tsai YH. 2017. Chang

642   Gung Research Database: A multi-institutional database consisting of original medical records.

643   *Biomedical Journal* 40:263–269. DOI: 10.1016/j.bj.2017.08.002.

644   Tseng Y-J, Cami A, Goldmann DA, DeMaria A, Mandl KD. 2015. Incidence and Patterns of Extended-

645   Course Antibiotic Therapy in Patients Evaluated for Lyme Disease. *Clinical Infectious Diseases*

646   61:1536–1542. DOI: 10.1093/cid/civ636.

647   Tseng Y-J, DeMaria A, Goldmann DA, Mandl KD. 2017. Claims-Based Diagnostic Patterns of Patients

648   Evaluated for Lyme Disease and Given Extended Antibiotic Therapy. *Vector-Borne and Zoonotic*

649   *Diseases* 17:116–122. DOI: 10.1089/vbz.2016.1991.

650   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.

651   Risk Factors Associated with Outcomes of Recombinant Tissue Plasminogen Activator Therapy in

652   Patients with Acute Ischemic Stroke. *International Journal of Environmental Research and Public*

653   *Health* 17:618. DOI: 10.3390/ijerph17020618.

654   U.S. Centers for Medicare & Medicaid Services.ICD-9. *Available at*

655   *https://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html* (accessed

656   October 8, 2017a).

657   U.S. Centers for Medicare & Medicaid Services.ICD-10. *Available at*

658   *https://www.cms.gov/Medicare/Coding/ICD10/index.html* (accessed October 8, 2017b).

659   Wasey JO, Lang M. 2020.icd: Comorbidity Calculations and Tools for ICD-9 and ICD-10 Codes.

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

661   Weiskopf NG, Weng C. 2013. Methods and dimensions of electronic health record data quality

662   assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics*

663   *Association* 20:144–151. DOI: 10.1136/amiajnl-2011-000681.

664   Wickham H, Hester J, Chang W. 2020.devtools: Tools to Make Developing R Packages Easier. *Available*

665   *at https://cran.r-project.org/package=devtools*

666   Wu J, Roy J, Stewart WF. 2010. Prediction Modeling Using EHR Data: Challenges, Strategies, and a

667   Comparison of Machine Learning Approaches. *Medical Care* 48.

668   Yoshida K, Bartel A. 2020.tableone: Create "Table 1" to Describe Baseline Characteristics with or

669   without Propensity Score Weights. *Available at https://cran.r-project.org/package=tableone*

670

Figure 1

Overview of the *dxpr* package

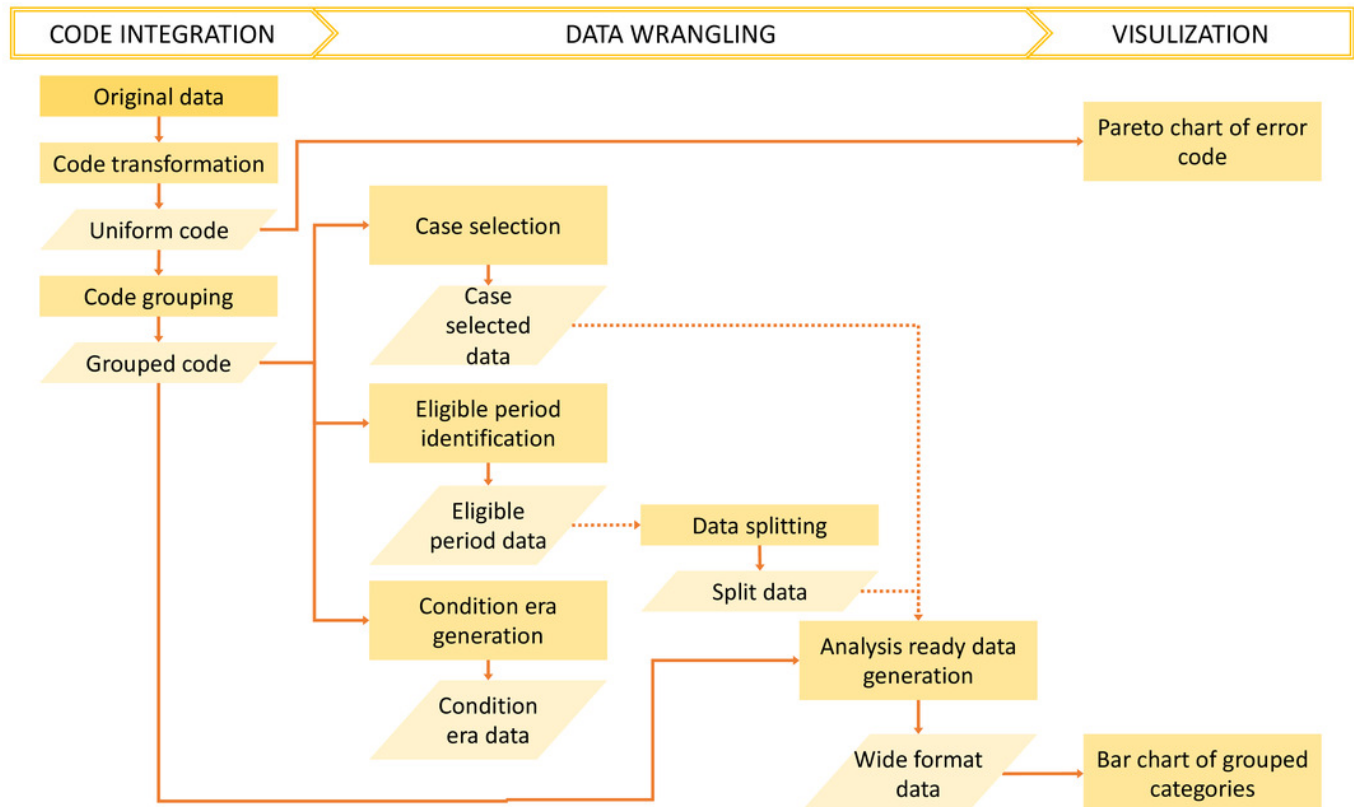
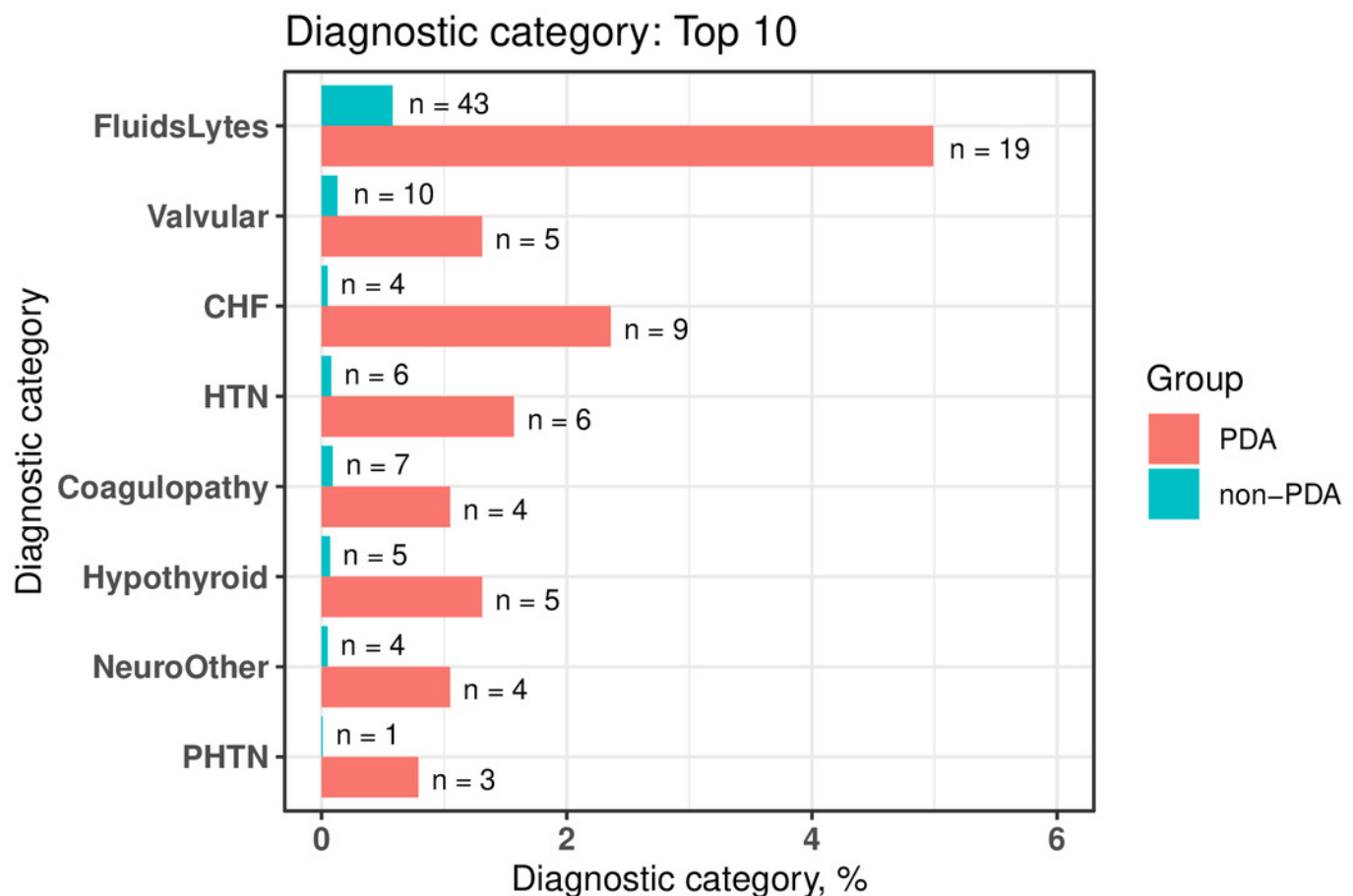



Figure 2

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.

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

Grouping results from grouping functions—icdDxToCCS

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

Summarized results from grouping functions—icdDxToCCS

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

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

1   **Table 4.** Summary of AHRQ-defined comorbidities based on the tableone package using the

2   integrated data generated by the *dxpr* package

AHRQ[a] comorbidities	Non-PDA	PDA[b]	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

3   [a] AHRQ: Agency for Healthcare Research and Quality

4   [b] PDA: patent ductus arteriosus

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

2   [a]ICD, International Classification of Diseases; [b]CCS, Clinical Classifications Software;

3   [c]PheWAS, Phenome Wide Association Studies; and [d]EHR, Electronic Health Record.