Supplementary File. Main R functions of the Queralt DxS tool and Queralt System user's guide.

Performance of Three Measures of Comorbidity in Predicting Critical COVID-19: A Retrospective Analysis of 4,607 Hospitalized Patients

David Monterde, Gerard Carot-Sans, Miguel Cainzos-Achirica, Sònia Abilleira, Marc Coca, Emili Vela, Montse Clèries, Damià Valero, Josep Comin-Colet, Luis Garcia-Eroles, Pol Perez-Sust, Miquel Arrufat, Yolanda Lejardi, and Jordi Piera

Contents

```
Queralt.System.Dx = function(Data.Dx, id = "ID", dx = "ICD", type = "Type", poa = "POA") {

  if (!require("dplyr")) install.packages('package')
  if (!require("tidyr")) install.packages('package')
  library(dplyr)
  library(tidyr)


  Data = Data.Dx[,c(id,dx,type,poa)]
  names(Data) = c("ID", "Dx", "Type", "POA")
  Data$ID = as.character(Data$ID)
  PD = Data[Data$Type=="P",c("ID", "Dx")]
  PD = unique(PD)
  names(PD) = c("ID", "PD")

  Data =

    suppressMessages(

      Data%>%
        left_join(Table_Dx_QI, by = c("Dx" = "ICD"))%>%
        select(ID, ACS, Type, POA, W_global, W_risk, W_resources, W_severity)%>%
        unique()%>%
        mutate(ACS3 = substr(ACS, 1, 3))%>%
        mutate(W_global = coalesce(W_global, 0),
               W_risk = coalesce(W_risk, 0),
               W_resources = coalesce(W_resources, 0),
               W_severity = coalesce(W_severity, 0))%>%
        mutate(B = ifelse(Type == "P",
                          "IQDP",
                          ifelse(Type == "S" & POA == 1,
                                 "IQDA",
                                 "IQDC")))%>%
        filter(!is.na(ID) & !is.na(B) & !is.na(ACS3))%>%
        group_by(ID, B, ACS3)%>%
        summarise(W_global = max(W_global),
                  W_risk = max(W_risk),
                  W_resources = max(W_resources),
                  W_severity = max(W_severity))%>%
        ungroup()%>%
        group_by(ID, B)%>%
        summarise(W_global = sum(W_global),
                  W_risk = sum(W_risk),
                  W_resources = sum(W_resources),
                  W_severity = sum(W_severity))%>%
        ungroup()%>%
        pivot_longer(3:6, names_to = "W", values_to = "V")%>%
        mutate(Label = paste0(B, "_", W))%>%
        select(ID, Label, V)%>%
        pivot_wider(names_from = Label, values_from = V)%>%
        replace(., is.na(.), 0)%>%
        mutate(IQD_global = IQDA_W_global + IQDC_W_global + IQDP_W_global,
               IQD_risk = IQDA_W_risk + IQDC_W_risk + IQDP_W_risk,
               IQD_resources = IQDA_W_resources + IQDC_W_resources + IQDP_W_resources,
               IQD_severity = IQDA_W_severity + IQDC_W_severity + IQDP_W_severity)%>%
        arrange(ID)%>%
        left_join(PD, by = "ID")%>%
        left_join(Table_AD_2020, by = c("PD" = "ICD"))%>%
        mutate(CxD_Global = 1,
               CxD_Risk = 1,
               CxD_Resources = 1,
               CxD_Severity = 1)%>%
        arrange(ID))

  Data$CxD_Global = ifelse(Data$pt2_Global != 0 & Data$IQD_global >= Data$pt2_Global,
                           2,
                           Data$CxD_Global)
  Data$CxD_Global = ifelse(Data$pt3_Global != 0 & Data$IQD_global >= Data$pt3_Global,
                         3,
                         Data$CxD_Global)
Data$CxD_Global = ifelse(Data$pt4_Global != 0 & Data$IQD_global>= Data$pt4_Global,
                         4,
                         Data$CxD_Global)
Data$CxD_Global = ifelse(Data$pt5_Global != 0 & Data$IQD_global>= Data$pt5_Global,
                         5,
                         Data$CxD_Global)

Data$CxD_Risk = ifelse(Data$pt2_Risk != 0 & Data$IQD_risk >= Data$pt2_Risk,
                       2,
                       Data$CxD_Risk)
Data$CxD_Risk = ifelse(Data$pt3_Risk != 0 & Data$IQD_risk >= Data$pt3_Risk,
                       3,
                       Data$CxD_Risk)
Data$CxD_Risk = ifelse(Data$pt4_Risk != 0 & Data$IQD_risk >= Data$pt4_Risk,
                       4,
                       Data$CxD_Risk)
Data$CxD_Risk = ifelse(Data$pt5_Risk != 0 & Data$IQD_risk >= Data$pt5_Risk,
                       5,
                       Data$CxD_Risk)

Data$CxD_Resources = ifelse(Data$pt2_Resources != 0 & Data$IQD_resources >= Data$pt2_Resources,
                            2,
                            Data$CxD_Risk)
Data$CxD_Resources = ifelse(Data$pt3_Resources != 0 & Data$IQD_resources >= Data$pt3_Resources,
                            3,
                            Data$CxD_Risk)
Data$CxD_Resources = ifelse(Data$pt4_Resources != 0 & Data$IQD_resources >= Data$pt4_Resources,
                            4,
                            Data$CxD_Risk)
Data$CxD_Resources = ifelse(Data$pt5_Resources != 0 & Data$IQD_resources >= Data$pt5_Resources,
                            5,
                            Data$CxD_Risk)

Data$CxD_Severity = ifelse(Data$pt2_Severity != 0 & Data$IQD_severity >= Data$pt2_Severity,
                           2,
                           Data$CxD_Severity)
Data$CxD_Severity = ifelse(Data$pt3_Severity != 0 & Data$IQD_severity >= Data$pt3_Severity,
                           3,
                           Data$CxD_Severity)
Data$CxD_Severity = ifelse(Data$pt4_Severity != 0 & Data$IQD_severity >= Data$pt4_Severity,
                           4,
                           Data$CxD_Severity)
Data$CxD_Severity = ifelse(Data$pt5_Severity != 0 & Data$IQD_severity >= Data$pt5_Severity,
                           5,
                           Data$CxD_Severity)

Data = Data%>%
  select(ID,
         AD,
         ADdesc,
         ADM,
         ADMdesc,
         ##Queralt Groups
         CxD_Global,
         CxD_Risk,
         CxD_Resources,
         CxD_Severity,
         ##Queralt Index sum up!
         IQD_global,
         IQD_risk,
         IQD_resources,
         IQD_severity,
         ##Queralt Index Weight's for principal diagnoses
         IQDP_W_global,
         IQDP_W_risk,
         IQDP_W_resources,
           IQDP_W_severity,
           ##Queralt Index Weight's for secondary diagnosis
           IQDA_W_global,
           IQDA_W_risk,
           IQDA_W_resources,
           IQDA_W_severity,
           ##Queralt Index Weight's for complications, aka not present on admission
           IQDC_W_global,
           IQDC_W_risk,
           IQDC_W_resources,
           IQDC_W_severity)

  return(Data)


}
```

Queralt System:

A tool for hospital discharges risk stratification in R

User’s Guide

Table of contents

1 First things first: what is Queralt System and what kind of data is expected?

1.1 Queralt System: a brief introduction

Stratification tools are widely used in healthcare data world. Queralt System aims to **capture and quantify the case-mix behind hospital discharges**.

Queralt System is similar to Charlson or Elixhauser indices or Diagnosis Related Groups (DRGs): the main goal is to assign a —risk- category to a patient, based on its personal health history and the hospitalization process.

While Charlson or Elixhauser indices focuses on comorbidities and DRGs stresses on the economic dimension, Queralt System integrates both worlds and can produce **different results according to the perspective of interest**: risk, resource consumption, severity and a global score.

Also, Queralt System **can handle diagnoses and procedures**, depending on the focus of the exercise: risk adjustment or resource consumption. For the purpose of this paper, **we just provide the functions and data for diagnoses**.

The main output generated are: **groups and indices**. Groups consists of an ordinal value ranged from 1 to 5, from lowest to highest risk. **Queralt Groups** (CxD) are computed based on diagnoses groups (AD) and, thus, their aim is to stratify the risk within AD.

Indices are calculated as the sum of weights of subindices associated to registered ICD-10 codes, differentiating their **position and context**: principal (P), secondary (S) or present on admission (POA).

As can be seen in figure 1, subindices are also provided by the function Queralt.System.Dx(). The paper submitted uses the **IQDA_risk**, that considers POA codes and focuses on the risk of death.

The **weights** are the key component of Queralt System and are based on the data (and, hence, the characteristics) of Catalan Healthcare System users.

The algorithm considers three basic dimensions of every hospital admission: **principal diagnosis** (leading cause of the hospitalization), **secondary diagnosis** present on admission and identified thereafter and possible **hospital-associated complications**.

All in all, Queralt System considers the main elements of a hospital discharge that can be found in any administrative database, similar to DRGs, with the addition that is **open and freely distributed in R** (table 1).

Table 1. Inputs considered by stratification algorithms

Inputs	Charlson	Elixhauser	DRGs	Queralt
Comorbidity	x	x	x	x
Complications			x	x
Main cause			x	x
Diagnoses			x	x
Procedures			x	x
Outputs				
Categorical			x	x
Numerical	x	x		x

1.2 The data: inputs

Queralt System for diagnoses requires a **dataframe structured in long format** with the following variables:

Table 2. Queralt System's functions and arguments

`Queralt.System.Dx(Data.Dx, id = "ID", dx = "ICD", type = "Type", poa = "POA")`

Variable	Function's argument	Description
Data.Dx	Data.Dx	The input dataframe
ID	id =	Individual Id of the patient or hospital discharge
Diagnoses	dx =	Diagnosis codes in ICD-10
Type	type =	Principal or secondary classification for the record: P or S values are expected
POA	poa =	Present on Admission: 0 or 1

It is usual that, sometimes, this kind of data is structured in wide format, that is to say, different diagnoses codes are organized through columns. When this is the case, it is possible to reshape the table via `tidyr::pivot_longer()` function.

1.3 From now on

From now on it is going to be explained the main steps behind the algorithm through practical examples using the provided functions and, finally, before proceeding further, it must be made quite clear that Queralt System is for **research purposes only**.

2 Queralt System: diagnoses, procedures and outcomes

2.1 `Queralt.System.Dx()`

As stated above, the input data must be a dataframe composed by the following columns: an Id, diagnoses in ICD-10 and without points, the type of the diagnosis (P -for principal- or S -for secondary-) and 0 or 1 if present on admission. An example data set can be found in the workspace:

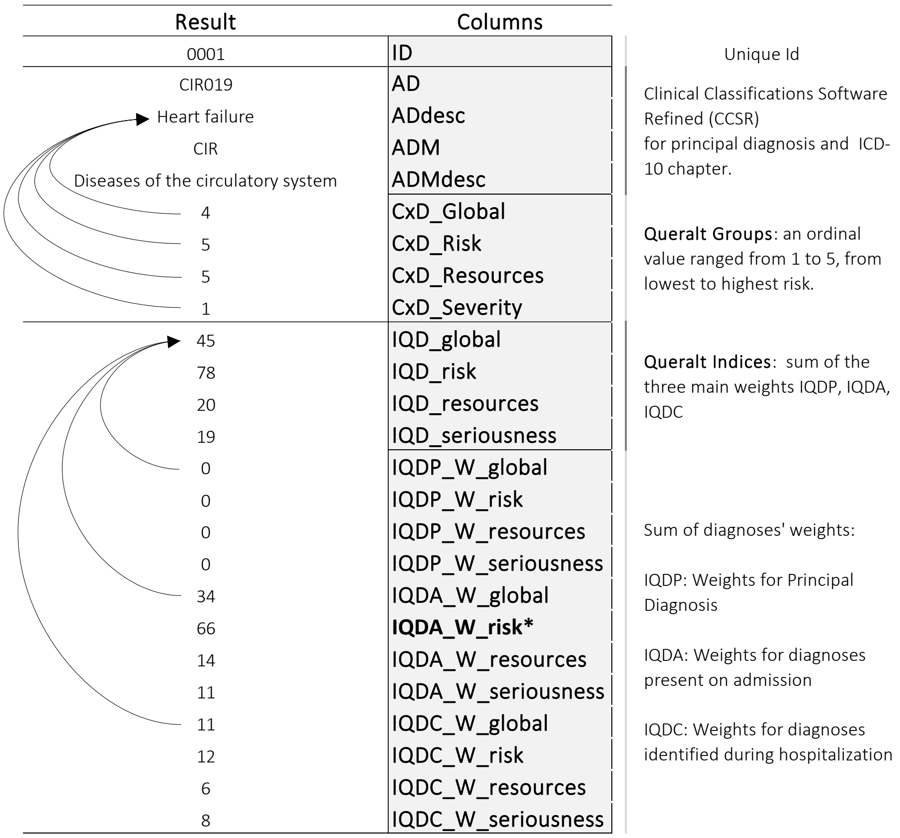
```
head(Data.Dx)

    ID     ICD    Type  POA
1 0872  O4292     P    1
2 0872    O669    S    1
3 0872 O99214     S    1
4 0872   E6609    S    1
5 0872   Z6830    S    0
6 0872   O9081    S    0
:  :       :      :    :
:  :       :      :    :
```

The function for the diagnoses is `Queralt.System.Dx()` and takes the following arguments (based on the example data):

```
Queralt.System.Dx(
  Data.Dx = Data.Dx,
  id = "ID",
  dx = "ICD",
  type = "Type",
  poa = "POA"
)
```

The output consists in a new dataframe of unique identifications (ID) and joined with the main components of Queralt System; an example is shown in figure 1.

Figure 1. Queralt System for diagnoses: outcomes



Result	Columns	
0001	ID	Unique Id
CIR019	AD	Clinical Classifications Software Refined (CCSR) for principal diagnosis and ICD-10 chapter.
Heart failure	ADdesc	
CIR	ADM	
Diseases of the circulatory system	ADMdesc	
4	CxD_Global	**Queralt Groups**: an ordinal value ranged from 1 to 5, from lowest to highest risk.
5	CxD_Risk	
5	CxD_Resources	
1	CxD_Severity	
45	IQD_global	**Queralt Indices**: sum of the three main weights IQDP, IQDA, IQDC
78	IQD_risk	
20	IQD_resources	
19	IQD_seriousness	
0	IQDP_W_global	Sum of diagnoses' weights: IQDP: Weights for Principal Diagnosis IQDA: Weights for diagnoses present on admission IQDC: Weights for diagnoses identified during hospitalization
0	IQDP_W_risk	
0	IQDP_W_resources	
0	IQDP_W_seriousness	
34	IQDA_W_global	
66	**IQDA_W_risk***	
14	IQDA_W_resources	
11	IQDA_W_seriousness	
11	IQDC_W_global	
12	IQDC_W_risk	
6	IQDC_W_resources	
8	IQDC_W_seriousness	

*IQDA_W_risk is the input considered in the submitted paper