2020

RDataXMan User Guide

Mark K Salloway[1]

Ning Yilin[2,3]

Tan Chuen Seng[1]

1: Saw Swee Hock School of Public Health, National University of Singapore (NUS) and National University Health System (NUHS); 2: NUS Graduate School for Integrative Sciences and Engineering, NUS; 3: Department of Surgery, Yong Loo Lin School of Medicine, NUS and NUHS.

Table of Contents

1. Overview and Installation

RDataXMan (**R Data** e**X**traction **Man**agement) is an Open Source tool built using the R language, with the capability to assist users perform reproducible extractions of datasets using a simple to use template approach. The R package is a used in conjunction with a user-friendly graphical user interface (GUI) based on the R Commander framework that assists the user from the identification of data or columns, to the full extraction of research data. Our aim in the development of this tool was to lower the barrier of entry and speed up efforts to access a variety of data sources for research, while promoting reproducibility and minimizing the risk of data extraction variation.

The RDataXMan package (available from https://github.com/nyilin/RDataXMan) and the R Commander plug-in (available from https://github.com/nyilin/RcmdrPlugin.RDataXMan) are free under an academic non-commercial license, and operates on Windows and Mac operating systems. Installation of this application is described in detail in the following sections.

1.1. Install R and RStudio

RDataXMan runs on R version 3.2.0 or later and the GUI runs on R version 3.5.0 or later, but users should avoid R version 4.0.0 because it has been reported to have issues with a dependency of this package. Users are recommended to install the latest version of R by following the instructions below.

Install R from the installer downloadable from the official website of the R Project.

- For Windows: download the Setup Wizard from https://cran.r-project.org/bin/windows/base/. Follow through the installation steps and keep the default options.
- For macOS: download the installer (a pkg file) from https://cran.r-project.org/bin/macosx/. Follow through the installation steps and keep the default options.

Install RStudio Desktop from the installer downloadable from the official website, https://rstudio.com/products/rstudio/download/#download.

1.2. Install RDataXMan and R Commander Plug-in

Installation of RDataXMan requires the installation of Java JDK:

- Go to https://www.oracle.com/technetwork/java/javase/downloads/index.html.
- Go to the download page for the installer of the latest Java JDK by following the “JDK Download” link.
- Download the appropriate installer.
    - Windows users should choose “Windows x64 Installer”.
    - Mac users should choose “macOS Installer”.

After installing Java JDK, macOS users need to open the terminal and execute the following commands to configure the path to Java:

```
sudo R CMD javareconf –n
sudo ln –s $(/usr/libexec/java_home)/jre/lib/server/libjvm.dylib /usr/local/lib
```

Installation and configuration of Java is successful if users are able to install and load the rJava package, by executing the following commands in RStudio without error:

```
install.packages("rJava")
library(rJava)
```

The RDataXMan package can be installed by opening RStudio and executing the following commands:

```
# First install package devtools if you have not done so:
# install.packages("devtools")
# To install the RDataXMan package:
devtools::install_github("nyilin/RDataXMan")
```

To use the RDataXMan package via the GUI, mac OS users need to first install an additional application, XQuarts, from its official website: https://www.xquartz.org. The GUI can be installed by executing the following command after installing the RDataXMan package:

```
# To install the R Commander plug-in:
devtools ::install_github("nyilin/RcmdrPlugin.RDataXMan")
```

1.3. Additional Requirements

RDataXMan uses Excel 97-2003 Workbook (which has a .xls extension) as request forms in data extraction tasks, therefore the use of RDataXMan requires the presence of software that is able to edit and save such files, e.g., Excel.

2. Illustrative example

2.1. Overview

We illustrate the usage of RDataXMan with an example based on real-life extraction requirements using simulated data. A quality of life (QoL) survey was conducted in 2005 in a hospital among 300 patients newly diagnosed with cancer, and the aim of the research study is to investigate whether their QoL has an effect on their inpatient admissions in 2006.

Information collected in the QoL survey is stored in an Excel file named "QoL survey data.xlsx", which include deidentified patient identifiers (*PATIENT_NRIC*) and their overall QoL (*Global QoL*). The deidentified patient identifiers in the QoL data will be linked to the electronic medical record (EMR) of the hospital (stored in a MySQL databased named "emr") to extract additional information on the inpatient admissions of these 300 patients in 2006. Specifically, the length of stay (*LOS*) of each inpatient admission in the year 2006 will be extracted from the inpatient movement table ("v2m_c_movement_pc_3yr"). These inpatient admissions will be linked to the diagnosis table ("v2m_c_diagnosis_p_3yr") using deidentified patient identifiers (*PATIENT_NRIC*) and deidentified case numbers (*CASE_NO*) to extract the International Classification of Diseases (ICD) code (*DIAGNOSIS_CD*), the corresponding name of disease (*DIAGNOSIS_DESC*) and the version of the ICD code (*ICD_VERSION*). We will also extract information on the race (*RACE*) of each patient from the demographics table ("v2m_c_patient_basic_3yr"), using *PATIENT_NRIC* as the identifier variable. The QoL survey data and the three EMR tables (available as CSV files) can be downloaded from: https://github.com/nyilin/RDataXMan_example_data.

The data request described above involves two inclusion criteria:

- an inclusion criterion based on *PATIENT_NRIC* of the QoL data to include all the 300 patients in the survey, and
- an inclusion criterion based on the year admission (*AYEAR*) of the inpatient movement table to include only inpatient admissions in 2006.

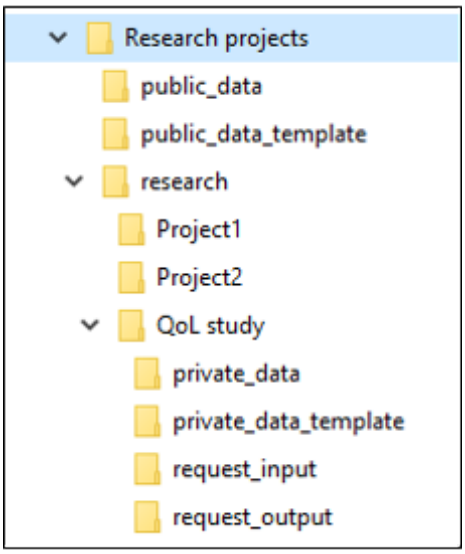
Four variable lists need to be generated for this data request, each corresponding to one of the four data sources:

- a variable list from the QoL data to extract the overall QoL (*Global QoL*),
- a variable list from the inpatient movement table to extract the length of stay (*LOS*),
- a variable list from the diagnosis table to extract information on diagnosis (*DIAGNOSIS_CD*, *DIAGNOSIS_DESC* and *ICD_VERSION*), and
- a variable list from the demographics table to extract the race (*RACE*) of each patient.

In Section 2.2, we will provide an overall introduction to the workflow of RDataXMan and the folder structure that facilitates this workflow. In Section 2.3, we will provide a detailed instruction on the use of RDataXMan via the R script and the GUI on a Windows operating system using this illustrative example.

2.2. Folder Structure Employed by RDataXMan

RDataXMan is able to extract data from a MySQL database as well as from local flat files, including TXT, CSV and Excel files (with either .xls or .xlsx extension) and data exported from R (with either .RData or .RDS extension), Stata (with .dta extension) and SPSS (with .sav extension). In each data extraction, RDataXMan generates Excel request forms for users to specify the data extraction requirements, and subsequently performs the data extraction and/or generates summary statistics on the data requested for. To manage the files associated with data extraction requests, RDataXMan employs a disciplined workflow that is integrated with an organized folder structure. This folder structure, as well as commands to create this folder structure, is described in detail below in the context of our illustrative QoL study example.



A working directory needs to be identified at the beginning of the workflow, where all input and output of data extraction will be organised into its subfolders. In our illustrative example we name the working directory as “Research projects”, in the folder “D:/Documents”. The working directory must include three subfolders: “research”, “public_data” and “public_data_template”. The working directory and these three subfolders can be created by first loading the RDataXMan package and then executing the following command:

```
# Load RDataXMan package:
library(RDataXMan)
# Create working directory and the three subfolders:
initWkdir(wkdir = "D:/Documents/Research projects")
```

The subfolder “research” contains the dedicated folders (which we will refer to as research folders) for each data request. The folder dedicated to our illustrative example is “QoL study”, and the other two folders, “Project1” and “Project2”, are research folder for other two data requests that are not described in this document. The research folder and its four essential subfolders, i.e., “private_data”, “private_data_template”, “request_input” and “request_output”, can be created by executing the following command (after loading the RDataXMan package):

```
initResearchFolder(wkdir = " D:/Documents/Research projects",
                   research.folder = "QoL study")
```

The “private_data” subfolder within the research folder contains flat files that is specific to this project, in our example the Excel file “QoL survey data.xlsx” that contains the survey data. Users need to manually move this Excel file to the “private_data” folder after initialising the research folder. Any request forms generated from this private data will be stored in the “private_data_template” subfolder. If there is sensitive information in private data files, users may consider encrypting the research folder to restrict access to authorised personals only. If a flat file will be the data source for more than one data requests, it should be saved in the “public_data” subfolder of the working directory to avoid duplication, but since the only flat file involved in our illustrative example is the QoL data, this subfolder will be left empty. Request forms generated from public data or from data on the server will be saved in the “public_data_template” subfolder of the working directory. Annotated request forms (which will be described in detail in the next subsection) should be saved to the “input” subfolder of the research folder, and data extraction output will be stored in the “output” subfolder.

In this section, we described how to initialise the working directory and research folder by using the RDataXMan package via the R script. These steps can also be done using the RDataXMan GUI, which are integrated with the selection of data sources and will be described in Section 2.3.2.1 and 2.3.4.1.

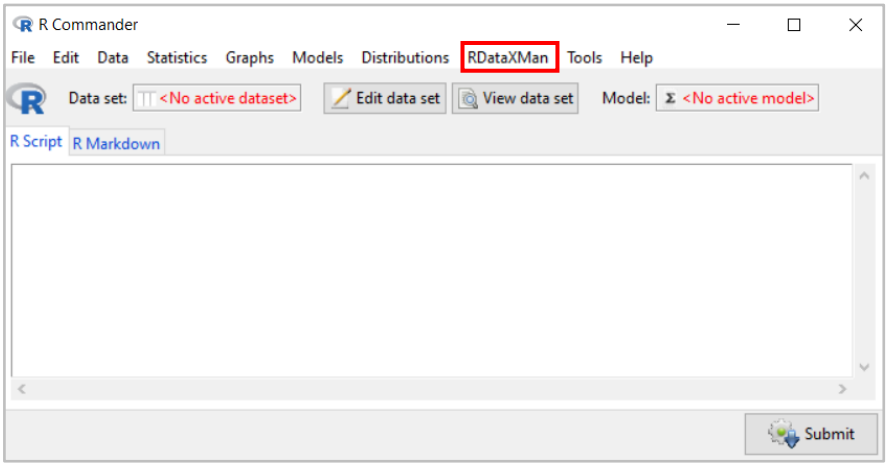
2.3. Using RDataXMan via Script and GUI

2.3.1. Launch RDataXMan GUI

To use RDataXMan via the R Commander GUI, open RStudio and execute the following command:

```
library(RcmdrPlugin.RDataXMan)
```

The following window should pop up, and all the features of RDataXMan is available from the “RDataXMan” drop-down menu:

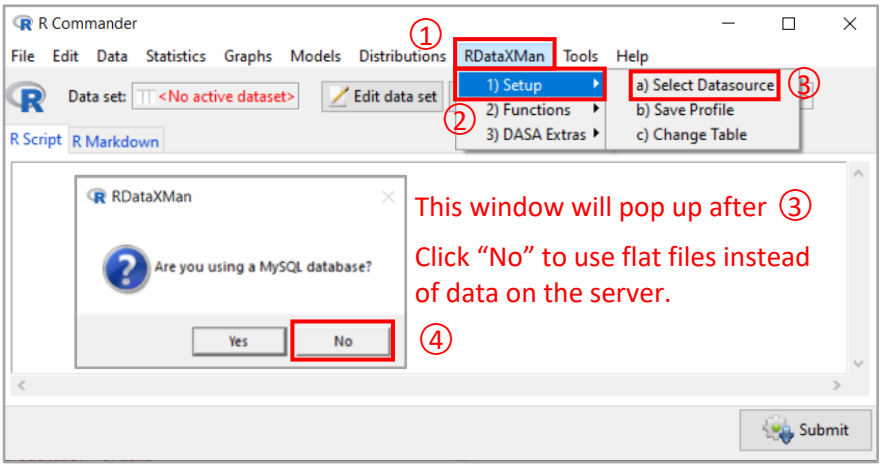


2.3.2. Work with Private Data

2.3.2.1. Select Data

Initialisation of the folder structure using the GUI requires users to first create the working directory, “D:/Documents/Research projects”, manually. Subsequently, GUI users can use the drop-down menu to need to select the data source to initialise the working directory

and research folder and then specify a data source (e.g., a private flat file, a public flat file, or a table on the server) to work with. Steps described in this subsection are not necessary for users who use RDataXMan via R scripts. We recommend GUI users to begin with flat files and then proceed to data on the server. Follow the instructions below to initialise the working directory and research folder and then select the private QoL data:

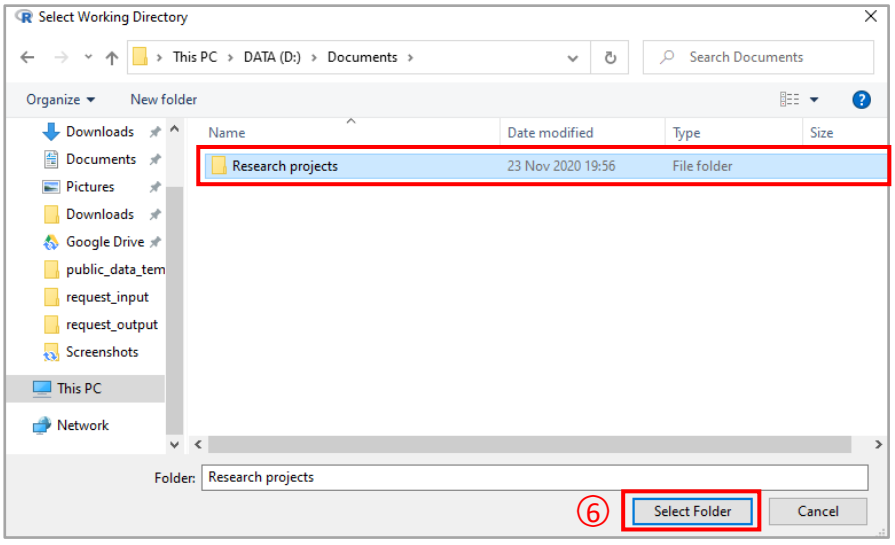


This window will pop up after ③

Click "No" to use flat files instead of data on the server.

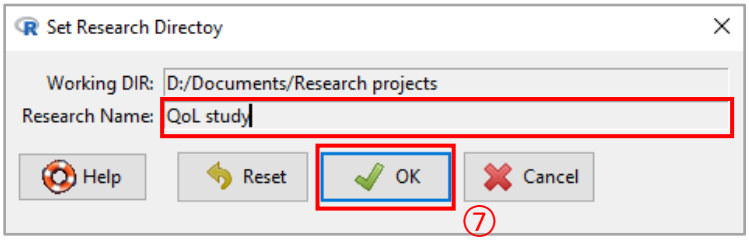
This window will pop up after ④

Click "Yes" to use private data.



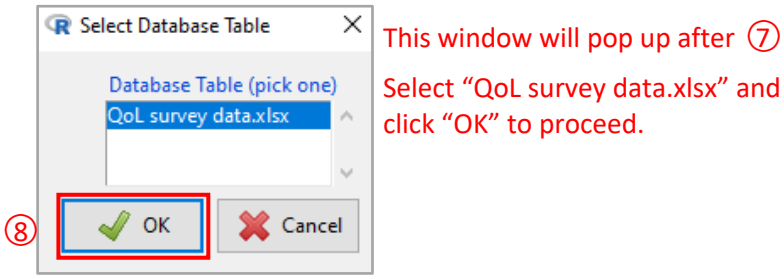
This window will pop up after ⑤

Use this window to allocate the working directory, and click "Select Folder" to proceed.

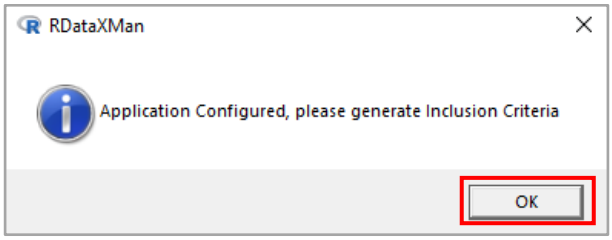


This window will pop up after ⑥

Use this window to specify the research folder, i.e., "QoL study", and then click "OK" to proceed.

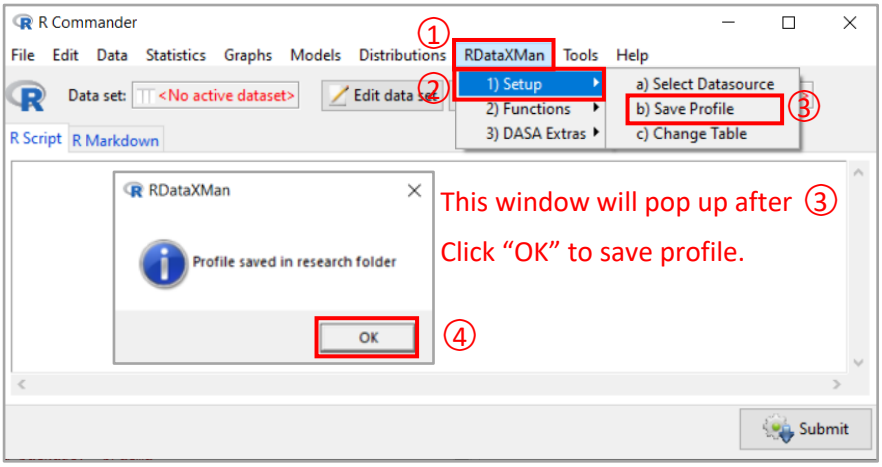


The following window will pop up after ⑧ to indicate the successful selection of data source. Click “OK” to proceed to the next step, i.e., to specify the inclusion criteria and/or the variables to select from the private data selected.

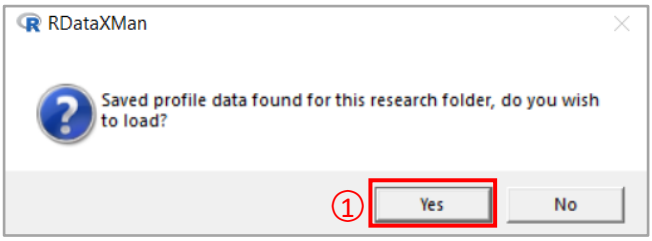


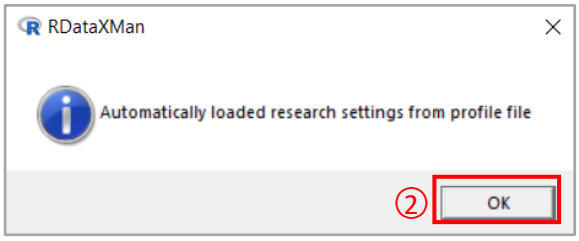
2.3.2.2. Save Profile

After specifying a data source, in this example the private QoL data, GUI users may choose to save this configuration as a file named “profile” in the current research folder:



When GUI users go through steps ① to ⑦ in Section 2.3.2.1 again to select the same research folder (i.e., folder “QoL study” within the working directory “Research projects”), the GUI will prompt following windows to ask users whether the existing “profile” file should be loaded, if any:



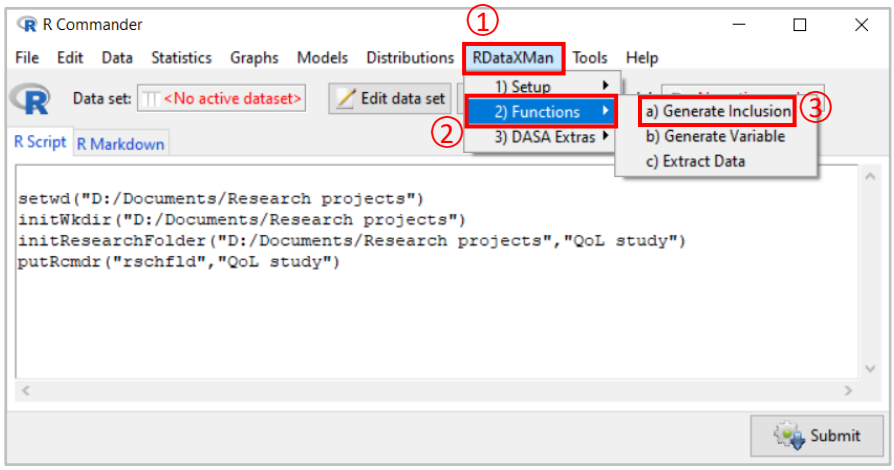


The two steps above specify the private QoL data as the current data source without going through step ⑧. The usefulness of saving the current data source configuration becomes prominent when working with data on the server (see Section 2.3.4).

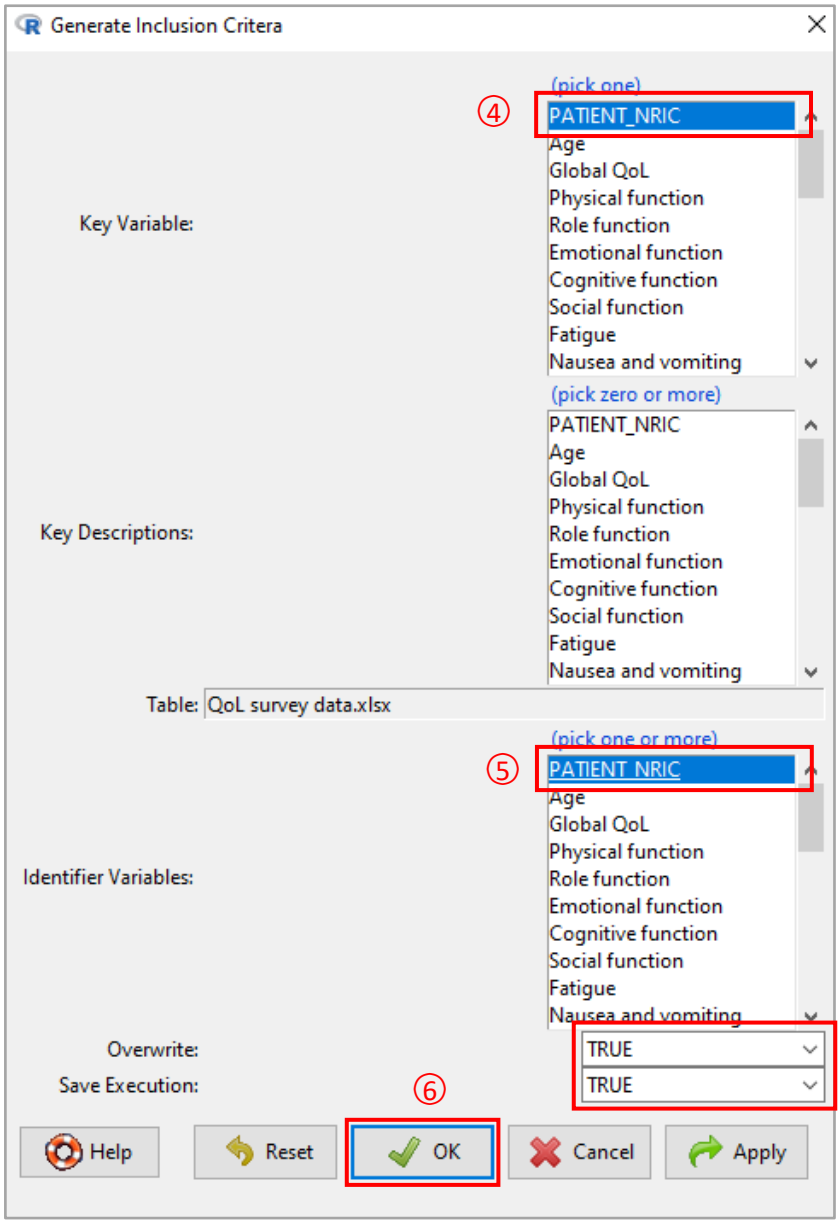
2.3.2.3. Generate Inclusion Criteria

Two essential information is required to specify an inclusion criterion: a single key variable that defines the inclusion criterion, and one or more identifier variables that uniquely identifies each subject in the data source. In our illustrative example, it is desirable to extract information for all the 300 patients recruited in the QoL study. This inclusion criterion can be specified by selecting *PATIENT_NRIC* as the key variable, which is also the identifier variable for this data source.

To generate inclusion criterion for the QoL data using the GUI, follow the instructions below:

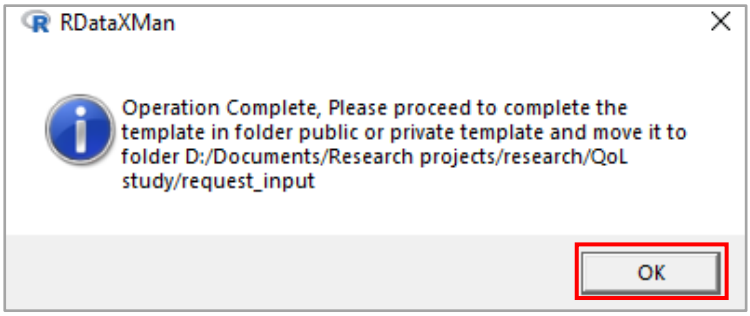


The following window will pop up after ③ for GUI users to select the key variable and the identifier variable for the QoL data:



Request form previously generated for the same data source with the same identifier variable will be overwritten, and the corresponding command will be saved to a TXT file in the research folder.

Select *PATIENT_NRIC* in "Key Variable" and "Identifier Variables", and click "OK" to proceed. The following window will pop up when the Excel request form for this inclusion criterion is generated:



To facilitate easier management of request forms, the file name of each request form for an inclusion criterion from a private flat file has the following structure: “inclusion.[name of private flat file]_[key variable]_[extension of private flat file].xls”. Hence, the request form generated above is named “inclusion.QoL survey data.xlsx_PATIENT_NRIC_xlsx.xls”. By default, existing request form that have the same file name will be overwritten to avoid duplication.

Steps ① to ⑥ correspond to the following R command:

```
genInclusion(wkdir = "Research projects", research.folder = "QoL study",
             table_name = "QoL survey data.xlsx",
             key.var = "PATIENT_NRIC", identifier.var = c('PATIENT_NRIC'),
             data.type = "flat", database = "private")
```

By default, the R command corresponding to the steps above is saved as a TXT file in the research folder, where file name has the following structure: “genInc-[date and time stamp].txt”, e.g., “genInc-2020-11-24 15-42-17.txt”.

The following Excel request form is generated and saved in the “private_data_template” subfolder in the research folder, which has 300 rows corresponding to the 300 unique NRIC:

	A	B	C	D	E
1	sno	PATIENT_NRIC	remarks	selection	logic
2	1	PX30511569074904790654068249463476548889917190161065767827261371 45			
3	2	PX30715488999767998253060170577395701695786374416087790435540585 48			
4	3	PX30797601078846223723000903898225761474397865224236369208859956 49			
5	4	PX31001724138466799705403832341260748001639839213053713138283740 07			
6	5	PX31684681101337527088961613295539517850291429484880668994849481 80			

To select all 300 subjects in this data, users can either put “x” in all the 300 rows under column “selection”:

	A	B	C	D	E
1	sno	PATIENT_NRIC	remarks	selection	logic
2	1	PX30511569074904790654068249463476548889917190161065767827261371 45		x	
3	2	PX30715488999767998253060170577395701695786374416087790435540585 48		x	
4	3	PX30797601078846223723000903898225761474397865224236369208859956 49		x	
5	4	PX31001724138466799705403832341260748001639839213053713138283740 07		x	
6	5	PX31684681101337527088961613295539517850291429484880668994849481 80		x	

Alternatively, users can write the R logical statement “!is.na(PATIENT_NRIC)” in the first row under column “logic” to select any row in the QoL data that has a valid value for *PATIENT_NRIC*:

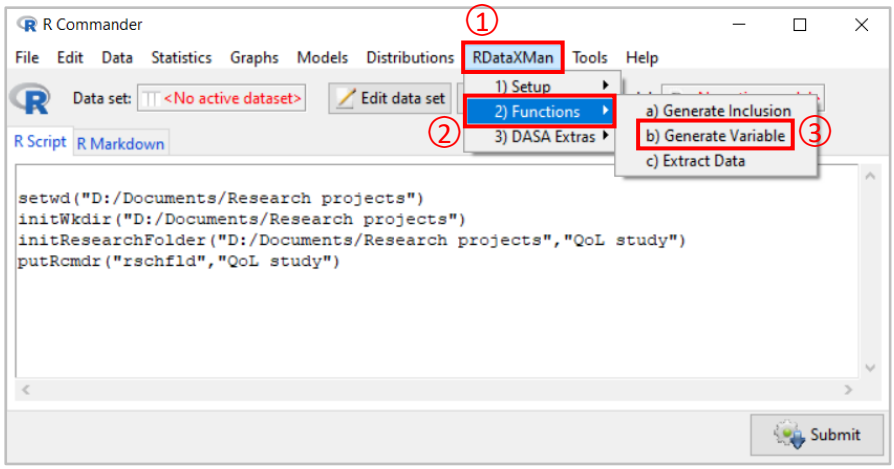
	A	B	C	D	E
1	sno	PATIENT_NRIC	remarks	selection	logic
2	1	PX30511569074904790654068249463476548889917190161065767827261371 45			!is.na(PATIENT_NRIC)
3	2	PX30715488999767998253060170577395701695786374416087790435540585 48			
4	3	PX30797601078846223723000903898225761474397865224236369208859956 49			
5	4	PX31001724138466799705403832341260748001639839213053713138283740 07			
6	5	PX31684681101337527088961613295539517850291429484880668994849481 80			

The presence of this “logic” column gives users who are familiar with R expressions much more flexibility to perform more complex filtering of data according to their needs.

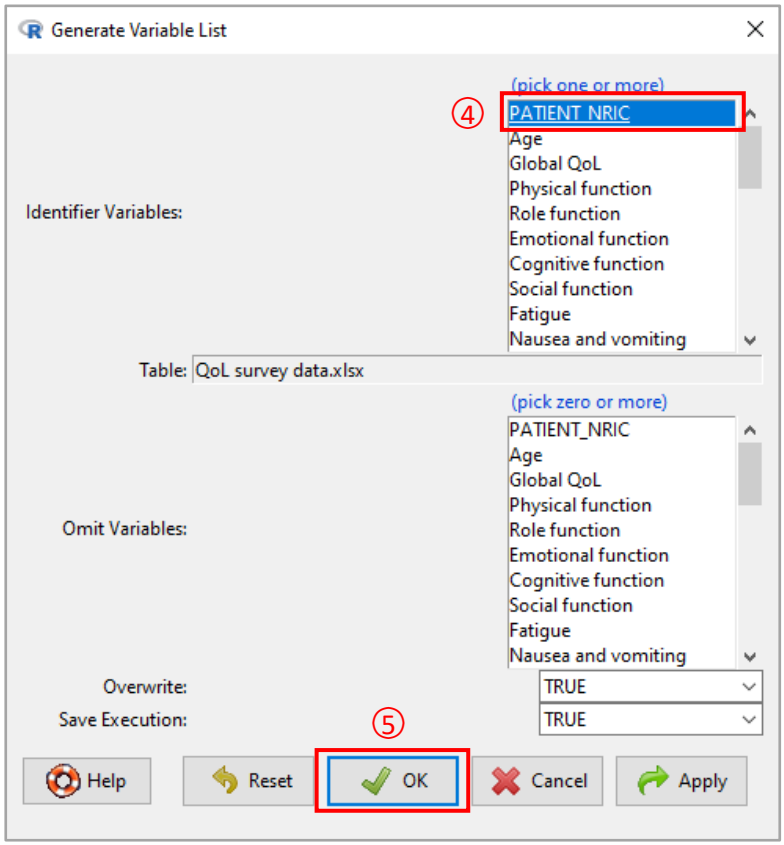
After specifying the selection, either by using the “selection” or “logic” column, users should use the “Save As” option of Excel to save the annotated request form to the “request _input” subfolder in the research folder, using the same file name.

2.3.2.4. Select Variable

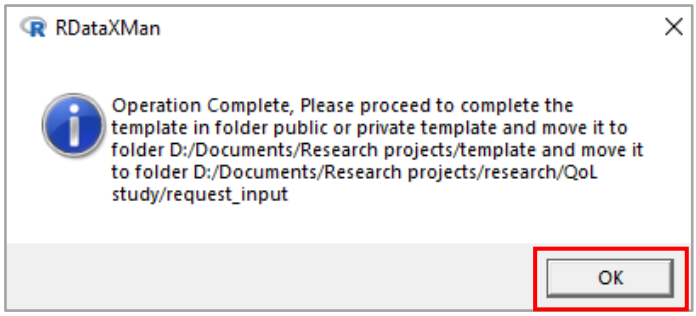
To specify the variable(s) to extract from a data source, users only need to specify the identifier variable(s) of this data source. In our illustrative example, we select a single variable *Global QoL* from the QoL data by following the steps below:



The following window will pop up after ③ for GUI users to select the identifier variable for the QoL data:



Select *PATIENT_NRIC* in “Identifier Variables” and click “OK” to proceed. The following window will pop up when the Excel request form for this variable selection is generated:



Steps ① to ⑤ correspond to the following R command:

```
genVariable(wkdir = "Research projects", research.folder = "QoL study",
            table_name  = "QoL survey data.xlsx",
            identifier.var = c('PATIENT_NRIC'),
            data.type = "flat", database = "private")
```

which is saved as a TXT file in the research folder named “genVar-2020-11-24 15-03-02.txt”.

The following Excel request form is generated and saved in the “private_data_template” subfolder in the research folder with file name “variable.QoL survey data.xlsx(PATIENT_NRIC)_xlsx.xls”, which follows the naming convention: “variable.[name of private flat file]_([identifier variables])_[extension of private flat file].xls”, e.g., “variable.QoL survey data.xlsx(PATIENT_NRIC)_xlsx.xls”. This request form lists the names of all variables in the QoL data except for the identifier variable (i.e., *PATIENT_NRIC*), which will always be selected. Users can indicate the variable “Global QoL” to extract by putting “x” in the corresponding row in column “selection”:

	A	B	C	D
1	sno	variable	remarks	selection
2	1	Age		
3	2	Global QoL		x
4	3	Physical function		
5	4	Role function		
6	5	Emotional function		

After specifying the selection, users should use the “Save As” option of Excel to save the annotated request form to the “request_input” subfolder in the research folder, using the same file name.

2.3.3. Work with Public Data

If a data request involves flat files shared among research projects (which is not present in this example), users can configure the GUI by selecting “No” instead of “Yes” in step ⑤ in Section 2.3.2.1, where files in the “public_data” subfolder of the working directory will be listed in the window after step ⑦. Otherwise, the steps to work with public data are the same as those to work the private data.

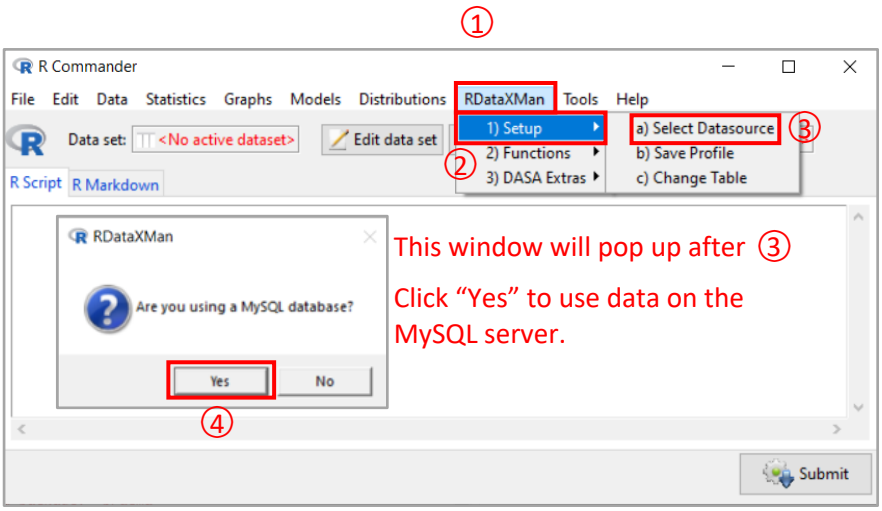
As mentioned in Section 2.2, request forms generated from public flat files will be saved in folder “public_data_template”. To avoid unintended overwriting of existing request forms, the file name of each request form ends with the date and time stamp.

2.3.4. Work with Data on MySQL server

To illustrate how to work with data on a MyQoL server using RDataXMan, we generate inclusion criterion and variable list from the movement table of EMR to include only inpatient admission in 2006. This table is saved in database "emr" with name "v2m_c_movement_pc_3yr", and contains the time of admission and discharge of each inpatient stay (*ADATE* and *AYEAR* for the date and year of admission, and *DDATE* and *DYEAR* for the date and year of discharge), and the length of each inpatient stay (*LOS*). Each entry is jointly defined by deidentified patient NRIC (*PATIENT_NRIC*) and deidentified case number (*CASE_NO*).

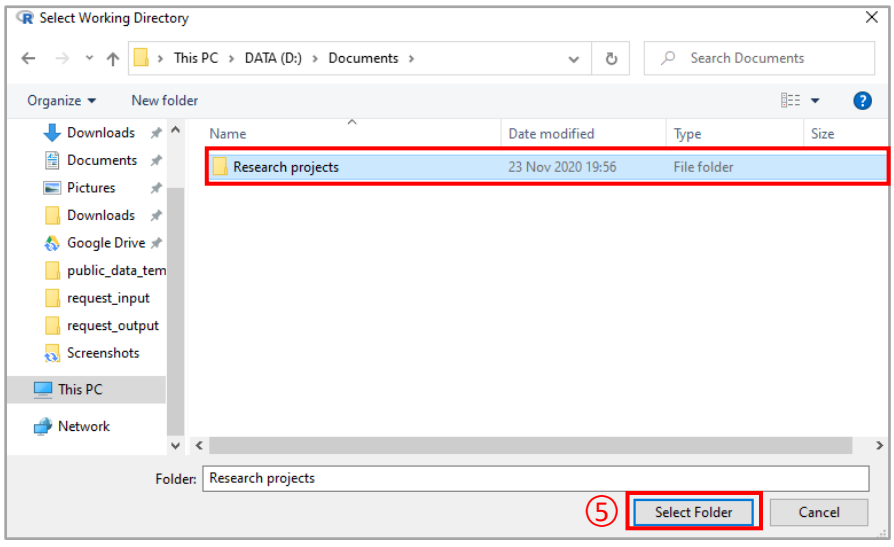
2.3.4.1. Select Data

Firstly, we need to switch data source from the private QoL data to the movement table by following the instruction below, which are not necessary if users are using RDataXMan via R scripts:



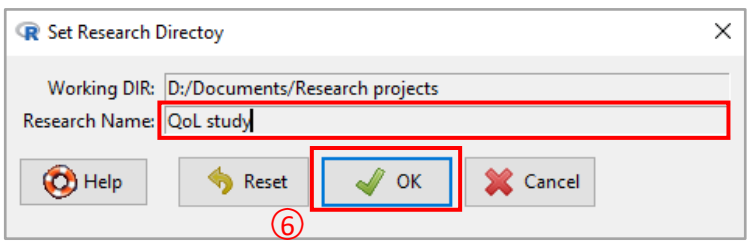
This window will pop up after ③

Click "Yes" to use data on the MySQL server.



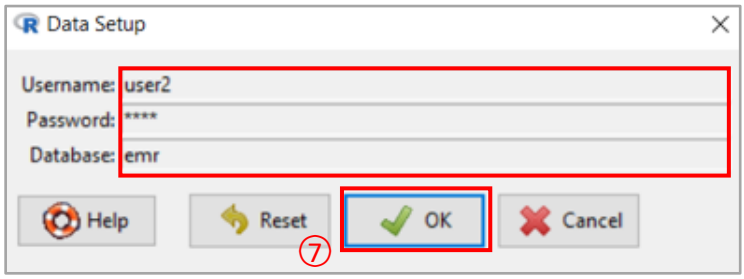
This window will pop up after ④

Use this window to allocate the working directory, and click "Select Folder" to proceed.



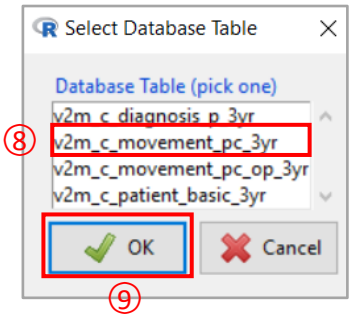
This window will pop up after ⑤

Use this window to specify the research folder, i.e., "QoL study", and then click "OK" to proceed.



This window will pop up after ⑥

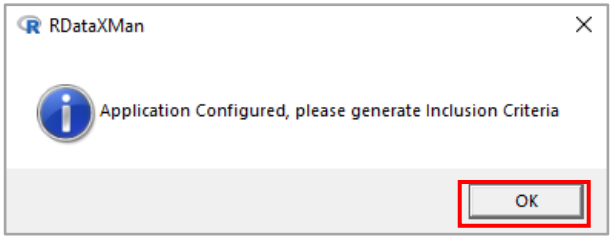
Use this window to enter the username, password and the name of database (i.e., "emr"), and then click "OK" to proceed.



This window will pop up after ⑦

Use this window to select the movement table, and then click "OK" to proceed.

The following window will pop up after ⑨ to indicate the successful change of data source. Click "OK" to proceed to the next step, i.e., to specify the inclusion criteria and/or the variables to select from the movement table.



Once the movement table is selected, the steps to generate inclusion criteria and variable list are the same as those described in Section 2.3.2.3 and 2.3.2.4 respectively.

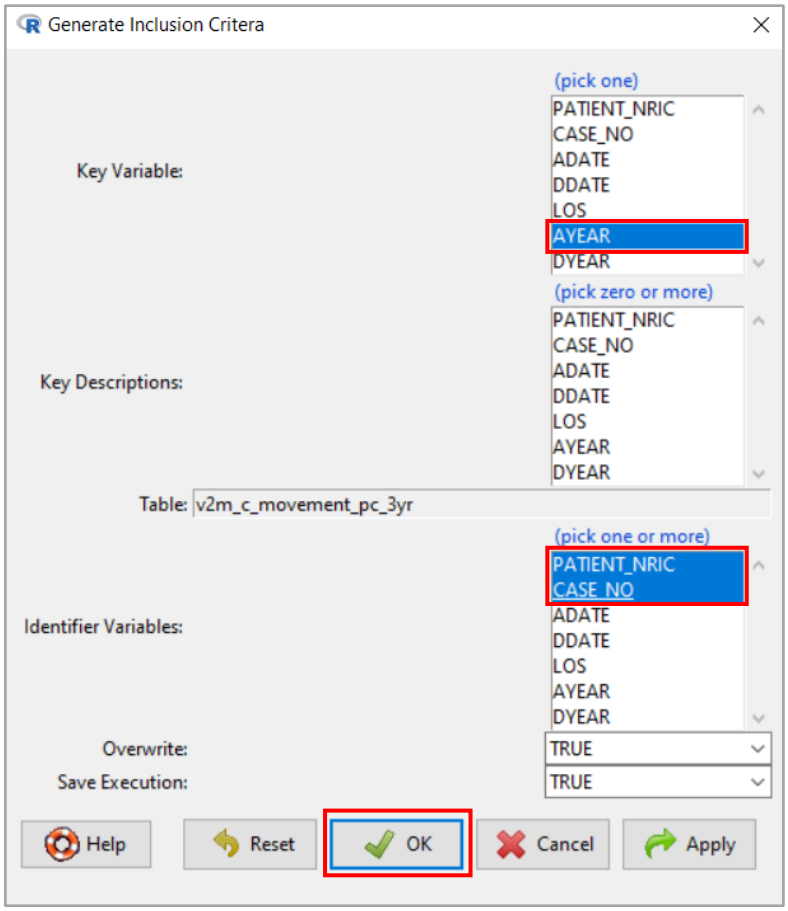
2.3.4.2. Save Profile

As described in Section 2.3.2.2, GUI users can save the current configuration of data source, i.e., the name of the database, the username and password to connect to the database, and the selection of the movement table. Note that this will always overwrite the existing profile saved in the research folder.

After saving the profile for connecting to the movement table, when GUI users specify the working directory as "Research projects" and the research folder as "QoL study" in a new data selection step, users may choose to load the existing profile, which gives them access to the movement table without the need to go through steps ⑦ to ⑨ in the previous section. After this, users can easily switch to other tables in the same database (see Section 2.3.4.5) as long as the account loaded from the data profile has access to them. Since loading previously saved data profile can give users direct access to tables on a database without the need to enter username and password, project managers are strongly advised to encrypt the project folder to prevent unauthorised access.

2.3.4.3. Generate Inclusion Criteria

In the illustrative example, we are interested in the inpatient admissions in year 2006, which can be selected by specifying *AYEAR* as the key variable, and using *PATIENT_NRIC* and *CASE_NO* as identifier variables:



Select multiple variables by holding "Ctrl" while clicking.

The corresponding R command is:

```
genInclusion(wkdir = "Research projects", research.folder = "QoL study",
             table_name = "v2m_c_movement_pc_3yr",
             key.var = "AYEAR", identifier.var = c('PATIENT_NRIC','CASE_NO'),
             data.type = "sql", username = "username", password = "password",
             database = "emr")
```

where "username" and "password" should be replaced by the actual username and password. The command is saved to the research folder as a TXT file.

Note that the request form generated is now saved to the "public_template" subfolder within the working directory, which by default overwrite any existing file that has the same file name. To avoid unintended overwriting of request forms, an inclusion request form generated from a table on the server follows a specific format: "inclusion.[table name]_([key variable])_sql_[username]_[date and time stamp].xls". For example, the request form generated from the steps above is "inclusion.v2m_c_movement_pc_3yr_AYEAR_sql_user2_20201124_150145.xls".

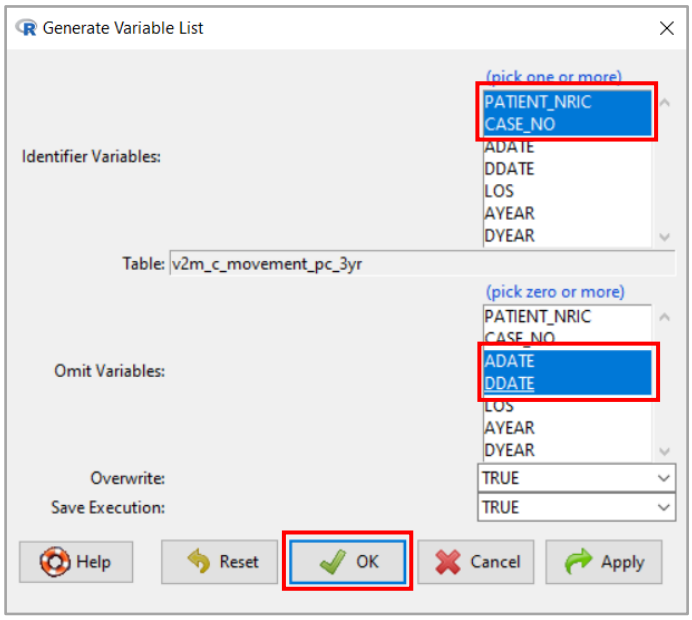
Select admission in 2006 by filling the request form:

	A	B	C	D	E
1	sno	AYEAR	remarks	selection	logic
2	1	2006		x	
3	2	2007			
4	3	2008			

and save the annotated request form (with the same file name) in the "request_input" folder within the research folder.

2.3.4.4. Select variable

The variable of interest in this movement table is *LOS*, which quantifies the length of stay of each inpatient stay. However, in addition to specifying the identifier variables as instructed in Section 2.3.2.4, it is advisable to omit the dates of admission and discharge (i.e., variables *ADATE* and *DDATE*) from the request form to prevent these two variables from being extracted to be compliant with data privacy and security regulations:



Variables selected here are excluded from the request form, and hence will not be extracted.

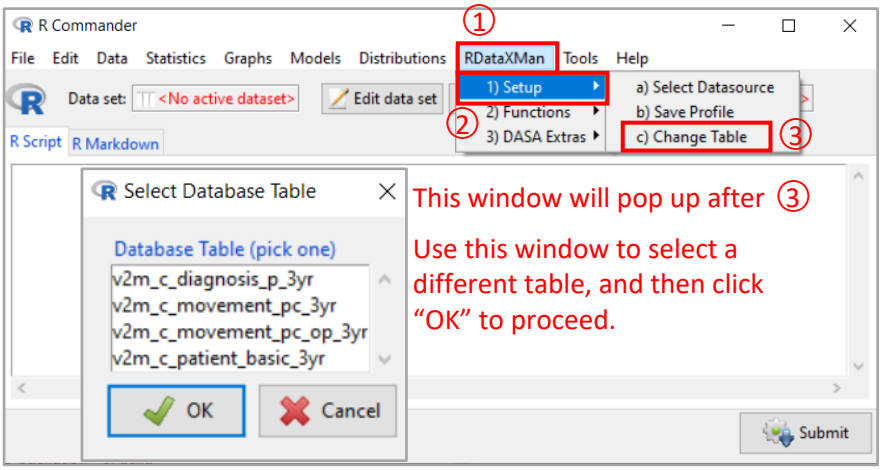
Select *LOS* in the request form generated, which is saved to the "public_template" subfolder within the working directory:

	A	B	C	D
1	sno	variable	remarks	selection
2	1	LOS		x
3	2	AYEAR		
4	3	DYEAR		

and save the annotated request form (with the same file name) in the "request_input" folder within the research folder. The request form for variable list follows the naming convention: "variable.[table name]_([identifier variables])_sql_[username]_[date and time stamp].xls", e.g., "variable.v2m_c_movement_pc_3yr_(PATIENT_NRIC_CASE_NO)_sql_user2_20201124_150154.xls".

2.3.4.5. Change Table

After generating inclusion criterion and variable list from the EMR movement table, GUI users can switch to another EMR table by following the instructions below:

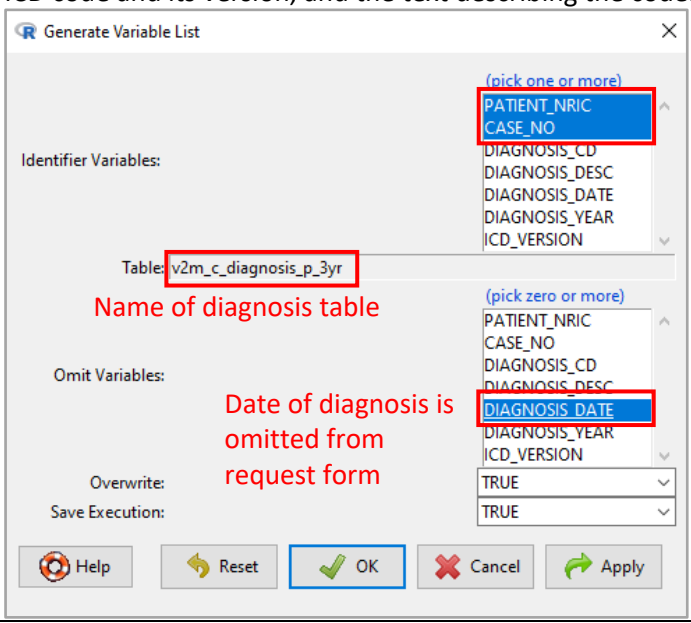


Although illustrated for data on the MySQL server, the same steps apply to public and private flat files.

2.3.5. Extract Data

After specifying inclusion criteria and generating variable lists from the private QoL data and the movement table on the MySQL server, generate variable lists from the diagnosis and demographics tables:

From diagnosis table, extract information on diagnosis associated with each inpatient admission, including the ICD code and its version, and the text describing the code.



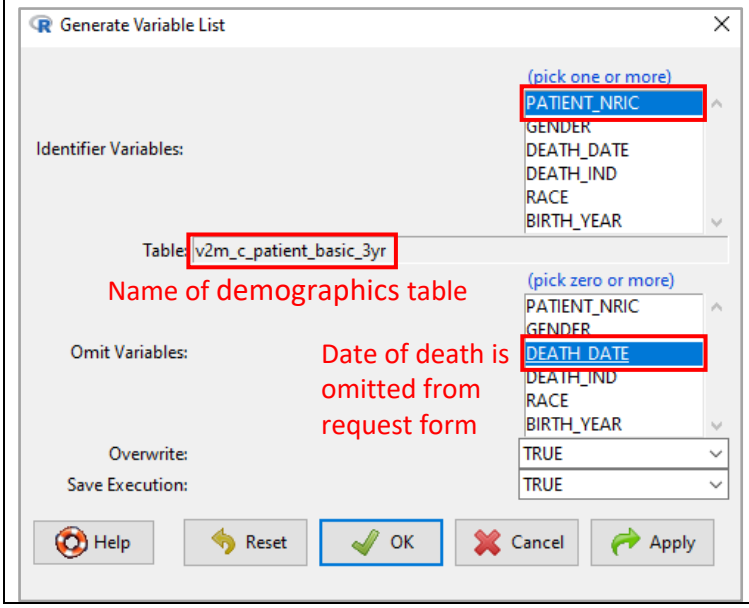
Corresponding R command:

```
genVariable(
  wkdir = "Research projects",
  research.folder = "QoL study",
  table_name  = "v2m_c_diagnosis_p_3yr",
  identifier.var = c('PATIENT_NRIC',
                     'CASE_NO'),
  omit.var = c('DIAGNOSIS_DATE'),
  data.type = "sql",
  username = "username",
  password = "password",
  database = "emr"
)
```

Annotated request form:

	A	B	C	D
1	sno	variable	remarks	selection
2	1	DIAGNOSIS_CD		x
3	2	DIAGNOSIS_DESC		x
4	3	DIAGNOSIS_YEAR		
5	4	ICD_VERSION		x

From demographics table, extract the race of each patient.



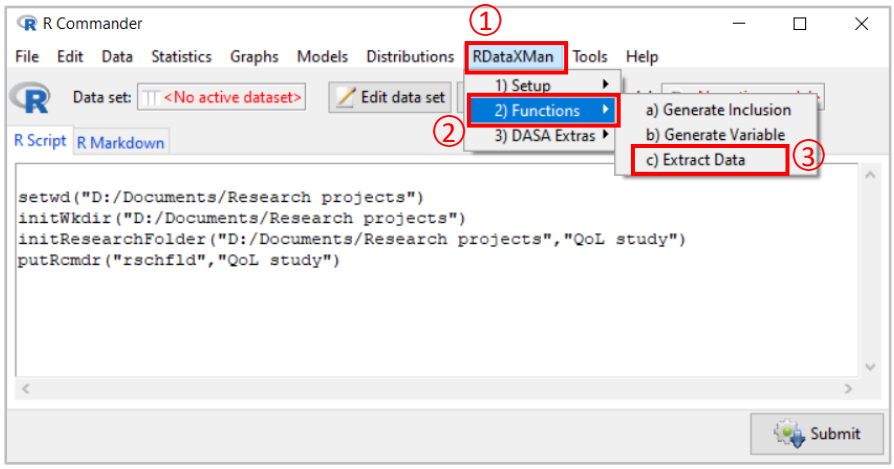
Corresponding R command:

```
genVariable(
  wkdir = "Research projects",
  research.folder = "QoL study",
  table_name = "v2m_c_patient_basic_3yr",
  identifier.var = c('PATIENT_NRIC'),
  omit.var = c('DEATH_DATE'),
  data.type = "sql",
  username = "username",
  password = "password",
  database = "emr"
)
```

Annotated request form:

	A	B	C	D
1	sno	variable	remarks	selection
2	1	GENDER		x
3	2	DEATH_IND		
4	3	RACE		
5	4	BIRTH_YEAR		

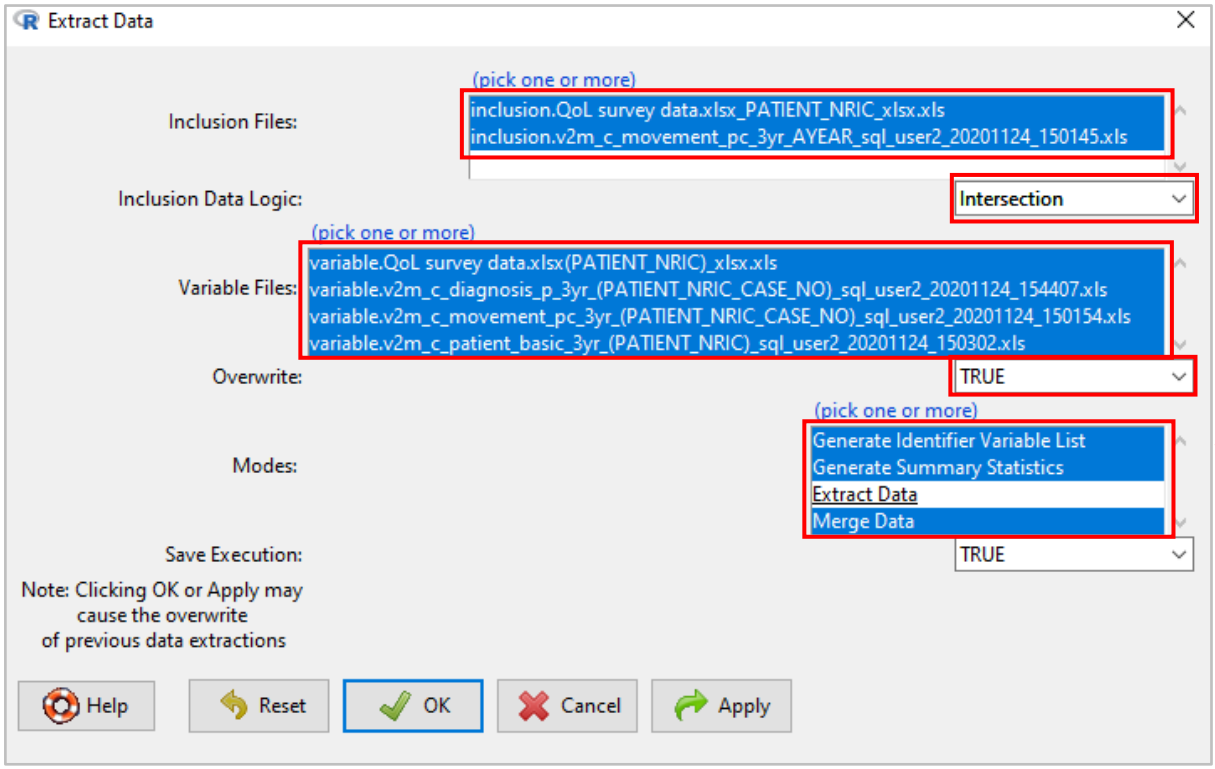
Hence, this data request involves 2 request forms on inclusion criteria (based on QoL data and movement table), and 4 request forms on variables to extract (from QoL data, movement table, diagnosis table and demographics table). Follow the instructions bellow to extract data:



The following window pops up after ③ for GUI users to select the request forms relevant to this data extraction. When more than 1 inclusion criteria are involved, users need to specify whether extracted data should satisfy all the inclusion criteria (where inclusion data logic is “Intersection”), or whether it is sufficient to satisfy any of the inclusion criterion (where inclusion data logic is “Union”).

For the output of data extraction, the package offers four modes. Mode 1 generates a list of identifier variables from the given inclusion criteria, which is also useful for defining base inclusion criteria for advanced extractions. Mode 2 produces summary statistics based on the request form(s), which is useful to indicate if the proposed inclusion criteria would yield sufficient number of participants for the study. Mode 3 extracts data based on the request form(s) provided, without merging them into a single dataset. Mode 4 produces a dataset that merges extracted data together and a dataset with merged inclusion variables.

In this illustrative example, we select modes 1, 2 and 4, and extract data that satisfy both inclusion criteria:



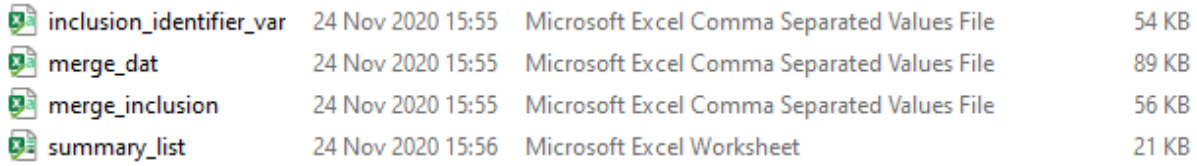
Existing output files will be overwritten

The corresponding R command is:

```
rdataxman_result <- extract_data(
  wkdir = "Research projects", research.folder = "QoL study",
  inclusion.xls.file = c(
    'inclusion.QoL survey data.xlsx_PATIENT_NRIC_xlsx.xls',
    'inclusion.v2m_c_movement_pc_3yr_AYEAR_sql_user2_20201124_150145.xls'
  ),
  variable.xls.file = c(
    'variable.QoL survey data.xlsx(PATIENT_NRIC)_xlsx.xls',
    'variable.v2m_c_diagnosis_p_3yr_(PATIENT_NRIC_CASE_NO)_sql_user2_20201124_154407.xls',
    'variable.v2m_c_movement_pc_3yr_(PATIENT_NRIC_CASE_NO)_sql_user2_20201124_150154.xls',
    'variable.v2m_c_patient_basic_3yr_(PATIENT_NRIC)_sql_user2_20201124_150302.xls'
  ),
  dataLogic = "Intersection", select.output = c('1','2','4'), overwrite = TRUE,
  username = "user2", password = "password", database = "emr"
)
```

and is saved to the research folder as a TXT file, where the file name starts with “exData” and is followed by the data and time stamp, e.g., “exData-2020-11-24 15-55-58.txt”.

The following output files are saved to the “request_output” folder:



which are described in detail in the next page.

Mode 1: generate a file named “inclusion_identifier_var.csv” that contains identifier variable list.

	A	B
1	CASE_NO	PATIENT_NRIC
2	CN78522960840063775 57592	PX30511569074904790654068 24
3	CN66362568674606926 45812	PX30715488999767998253060 17
4	CN21303315079743618 62141	PX30797601078846223723000 90
5	CN74986665491769234 76755	PX31001724138466799705403 83

Mode 2: generate a file named “summary_list.xlsx” with multiple sheets that contains summary statistics, including sample size (top) and summary statistics for variables extracted (bottom).

	A	B
1	Item	Summary
2	Total unique CASE_NO	395
3	Total unique PATIENT_NRIC	300

	A	B	C	D	E
1	Variable	N	Group	Summary	Type
2	Global QoL	395		75.79 (33.77)	Mean(S.D.)
3	DIAGNOSIS_CD	395	174.0	52 (13.16%)	N(%)
4			174.1	37 (9.37%)	N(%)
5			174.2	35 (8.86%)	N(%)
6			174.3	40 (10.13%)	N(%)
7			174.4	41 (10.38%)	N(%)
8			174.5	48 (12.15%)	N(%)
9			174.6	48 (12.15%)	N(%)
10			174.8	43 (10.89%)	N(%)
11			174.9	51 (12.91%)	N(%)
12	DIAGNOSIS_DESC	395	Malignant neoplasm of axillary tail of female breast	48 (12.15%)	N(%)
13			Malignant neoplasm of breast (female), unspecified	51 (12.91%)	N(%)
14			Malignant neoplasm of central portion of female breast	37 (9.37%)	N(%)
15			Malignant neoplasm of lower-inner quadrant of female breast	40 (10.13%)	N(%)
16			Malignant neoplasm of lower-outer quadrant of female breast	48 (12.15%)	N(%)
17			Malignant neoplasm of nipple and areola of female breast	52 (13.16%)	N(%)
18			Malignant neoplasm of other specified sites of female breast	43 (10.89%)	N(%)
19			Malignant neoplasm of upper-inner quadrant of female breast	35 (8.86%)	N(%)
20			Malignant neoplasm of upper-outer quadrant of female breast	41 (10.38%)	N(%)
21	ICD_VERSION	395	9CM	395 (100%)	N(%)
22	LOS	395		6.86 (2.71)	Mean(S.D.)
23	RACE	395	Chinese	323 (81.77%)	N(%)

Mode 3: extract data, with one file corresponding to each request form.

Mode 4: merge data, with one file for variables from inclusion criteria (“merge_inclusion.csv”, top) and one file for variables requested (“merge_dat.csv”, bottom).

	A	B	C
1	PATIENT_NRIC	CASE_NO	AYEAR
2	PX305115690749047	CN7852296084006377	2006
3	PX307154889997679	CN6636256867460692	2006
4	PX307976010788462	CN2130331507974361	2006
5	PX310017241384667	CN7498666549176923	2006

	A	B	C	D	E	F	G	H
1	PATIENT_NRIC	CASE_NO	Global QoL	DIAGNOSIS_CD	DIAGNOSIS_DESC	ICD_VERSION	LOS	RACE
2	PX30511569074904	CN78522960840063	100	174.1	Malignant neoplasn	9CM	5	Malay
3	PX30715488999767	CN66362568674606	62.5	174.6	Malignant neoplasn	9CM	7	Chinese
4	PX30797601078846	CN21303315079743	150	174.3	Malignant neoplasn	9CM	7	Indian
5	PX31001724138466	CN74986665491769	87.5	174.9	Malignant neoplasn	9CM	5	Malay

3. DASA Extra

The RDataXMan GUI includes another tool created by the DASA team, named Ezy De-identifier, that de-identifies text-based datasets. Launching Ezy De-identifier requires internet access, and additional R packages and software may need to be installed and configured to use the tool. Interested users can refer to this webpage for detailed introduction and instructions: http://blog.nus.edu.sg/dasa/ezy-de-identifier/.