

RDataXMan User Guide

Mark K Salloway¹

Ning Yilin^{2,3}

Tan Chuen Seng¹

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	3
1.1. <i>Install R and RStudio</i>	3
1.2. <i>Install RDataXMan and R Commander Plug-in</i>	3
1.3. <i>Additional Requirements</i>	7
2. Illustrative example	8
2.1. <i>Overview</i>	8
2.2. <i>Folder Structure Employed by RDataXMan.....</i>	9
3. Using RDataXMan via Script and GUI.....	11
3.1. <i>Launch RDataXMan GUI</i>	11
3.2. <i>Work with Private Data</i>	12
3.2.1. <i>Select Data</i>	12
3.2.2. <i>Save Profile (optional).....</i>	13
3.2.3. <i>Generate Inclusion Criteria</i>	14
3.2.4. <i>Select Variable</i>	17
3.3. <i>Work with Public Data</i>	19
3.3.1. <i>Select Data</i>	19
3.3.2. <i>Save Profile (optional).....</i>	20
3.3.3. <i>Generate Inclusion Criteria</i>	21
3.3.4. <i>Select variable</i>	22
3.3.5. <i>Change Table.....</i>	23
3.4. <i>Work with Data on MySQL server.....</i>	25
3.4.1. <i>Select Data</i>	25
3.4.2. <i>Save Profile (optional).....</i>	26
3.4.3. <i>Generate Inclusion Criteria</i>	27
3.4.4. <i>Select variable</i>	28
3.4.5. <i>Change Table.....</i>	29
3.5. <i>Extract Data</i>	31
4. DASA Extra.....	35

1. Overview and Installation

RDataXMan (**R Data eXtraction Management**) 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 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 of interest, to the full extraction of research data. The aim of developing this tool is to: (1) lower the barrier of entry, (2) speed up the process of accessing data from various source, and (3) promote a reproducible data extraction workflow that requires minimal edits in case of a variation to extraction requirements.

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 this particular version 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:

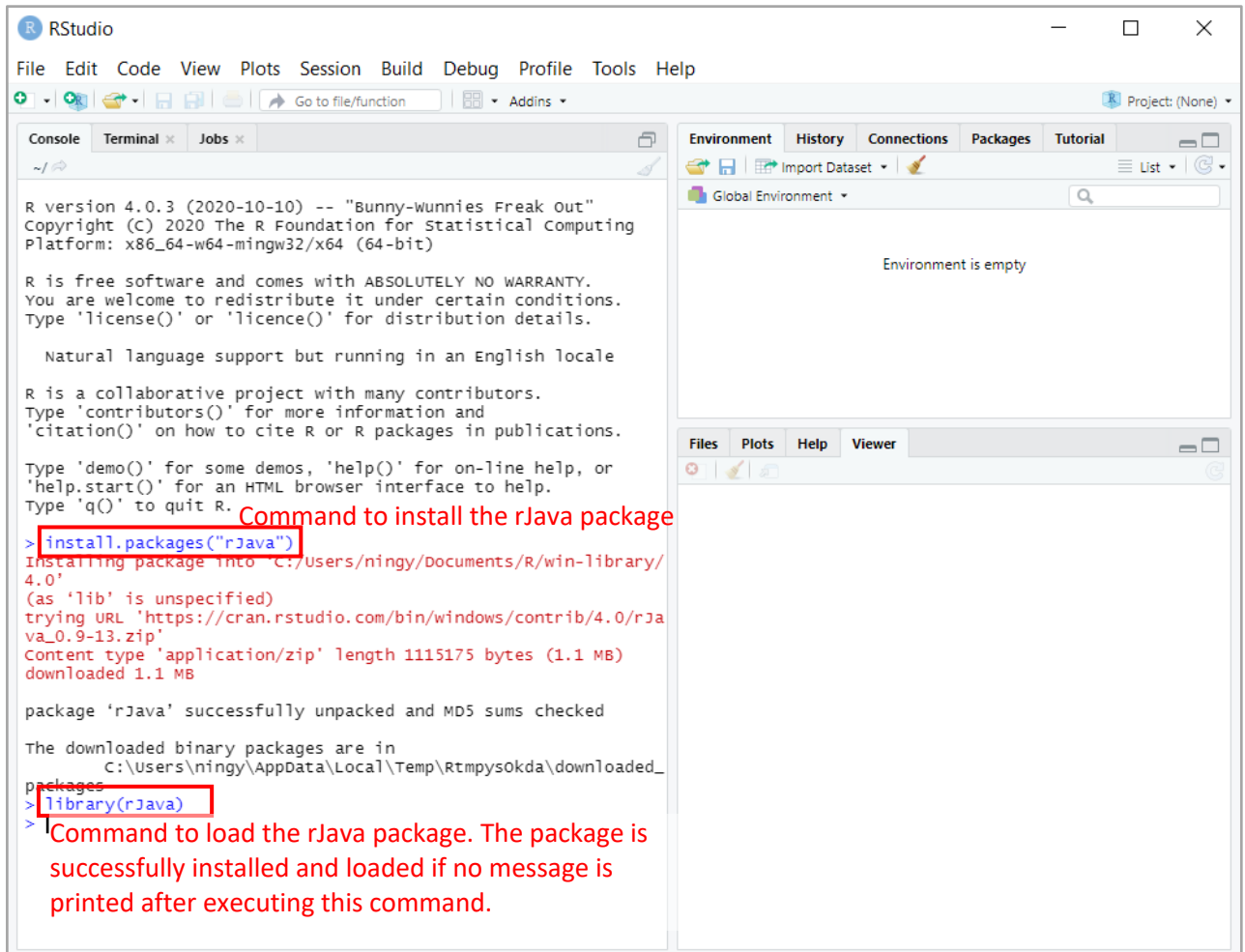
- Go to <https://www.oracle.com/technetwork/java/javase/downloads/index.html>.
- Go to the download page for the installer of the latest Java JDK by following the “JDK Download” link.
- Download the appropriate installer.
 - Windows users should choose “Windows x64 Installer”.
 - Mac users should choose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

Windows users do not need to go through this step.

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:



The screenshot shows the RStudio interface with the console pane active. The console displays the R startup message and the execution of two commands. The first command, `install.packages("rJava")`, is highlighted with a red box. The output shows the package being downloaded from CRAN and successfully installed. The second command, `library(rJava)`, is also highlighted with a red box. The output for this command is empty, indicating successful loading. Red text annotations provide additional context for each command.

```
R version 4.0.3 (2020-10-10) -- "Bunny-wunnies Freak out"
Copyright (c) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("rJava")
Installing package into 'C:/Users/ningy/Documents/R/win-library/
4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/rj
va_0.9-13.zip'
Content type 'application/zip' length 1115175 bytes (1.1 MB)
downloaded 1.1 MB

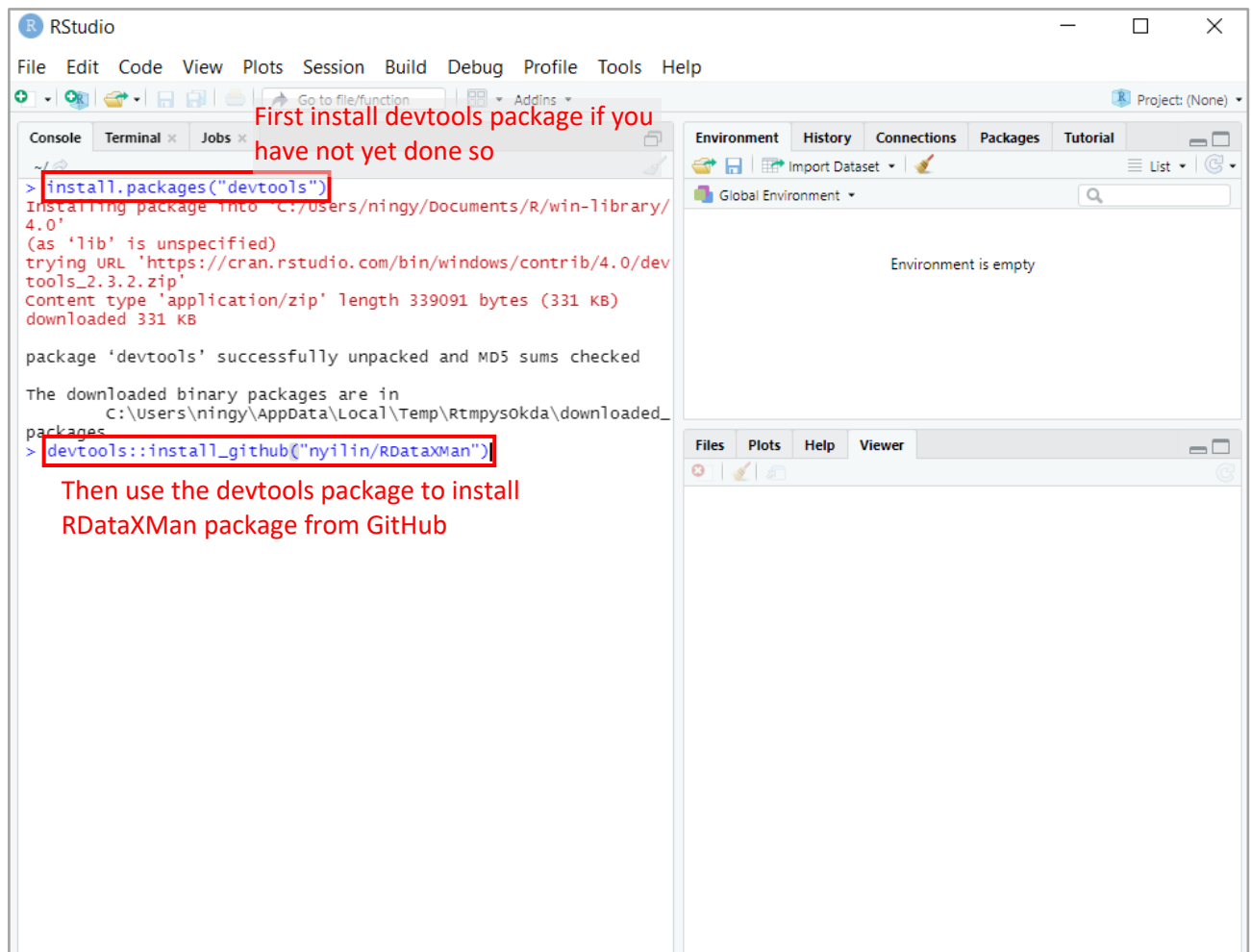
package 'rJava' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\ningy\AppData\Local\Temp\Rtmpysokda\downloaded_
packages
> library(rJava)
```

Command to install the rJava package

Command to load the rJava package. The package is successfully installed and loaded if no message is printed after executing this command.

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

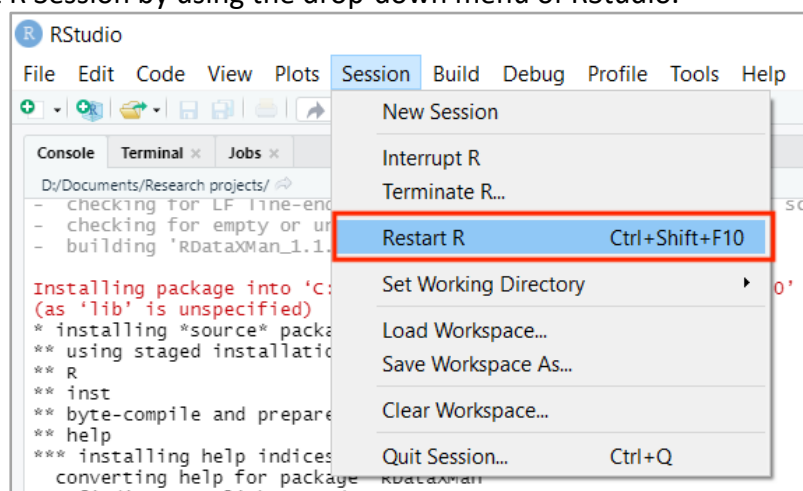


Note:

When installing the RDataXMan package from GitHub, users may be prompted to update a list of packages. Users may choose to update all or some of the packages, but this is not required by RDataXMan.

If users choose to update some or all of the packages listed, and an error occurs where R failed to remove prior installation of some packages, this error may be resolved by following the instructions below:

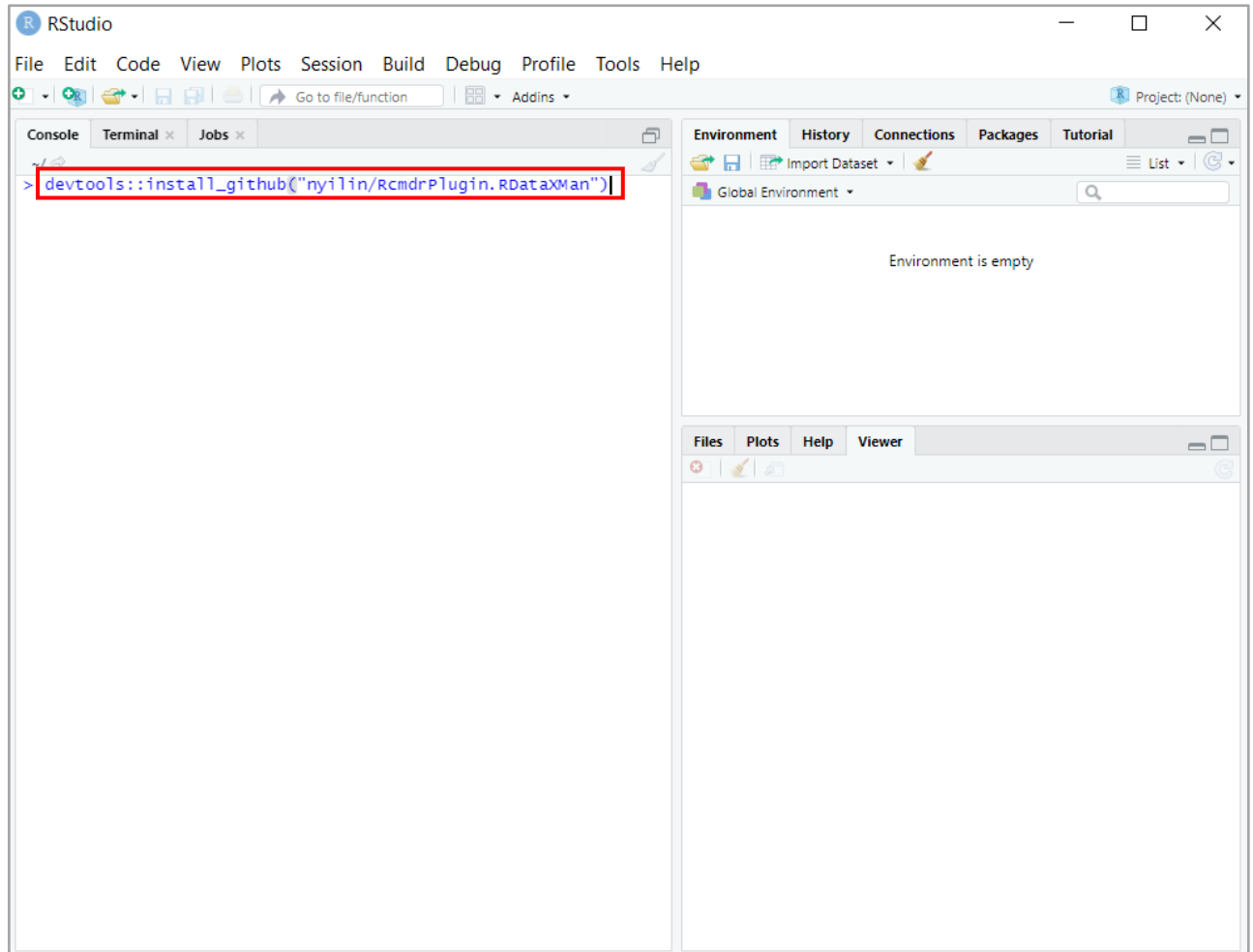
1. Restart R Session by using the drop-down menu of RStudio.



2. Manually install the package that caused the error. For example, if the error occurred because R failed to remove prior installation of package 'rlang', execute the following command in RStudio:
`install.packages("rlang")`
3. Retry installing RDataXMan from GitHub.

To use the RDataXMan package via the GUI, macOS users need to first install an additional application, XQuartz, from its official website: <https://www.xquartz.org>. This is not necessary for Windows users.

The GUI can be installed by executing the following command after installing the RDataXMan package:



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., Microsoft Excel.

2. Illustrative example

2.1. Overview

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) or data exported from R (with either .RData or .RDS extension), Stata (with .dta extension) and SPSS (with .sav extension). 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 patients newly diagnosed with cancer, and valid data was collected from 300 patients. The aim of the research study was to investigate whether the QoL of these 300 patients has an effect on their inpatient admissions in 2006.

Information collected in the QoL survey includes 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 to extract additional information on the inpatient admissions of these 300 patients in 2006.

Specifically, the length of stay (*LOS*) of each inpatient admission will be extracted from the inpatient movement table. These inpatient admissions will be linked to the diagnosis table using deidentified patient identifiers (*PATIENT_NRIC*) and deidentified case numbers (*CASE_NO*) to extract the diagnosis for each admission, including 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 demographics information on each patient from the demographics table, using race (*RACE*) as an example, where *PATIENT_NRIC* is the identifier variable.

The data request described above involves two inclusion criteria:

- all the 300 patients who participated in the survey and had valid QoL data, and
- inpatient admission in the year 2006.

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

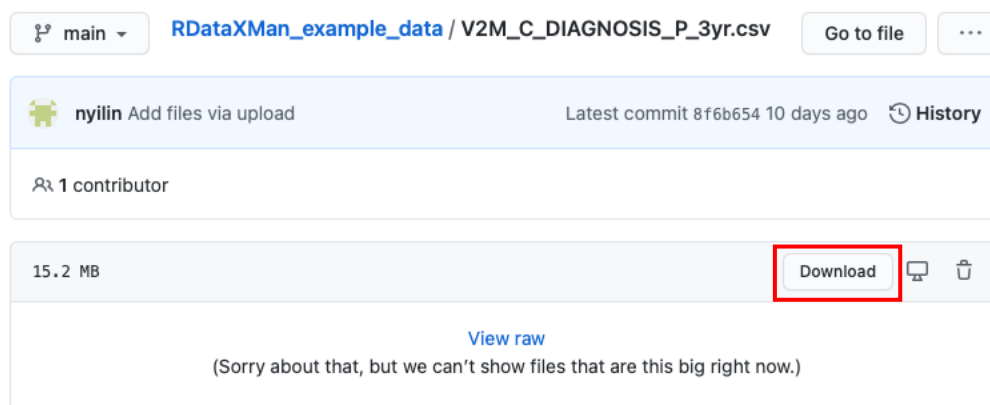
- QoL data to extract the overall QoL (*Global QoL*),
- Inpatient movement table to extract the length of stay (*LOS*),
- Diagnosis table to extract information on diagnosis (*DIAGNOSIS_CD*, *DIAGNOSIS_DESC* and *ICD_VERSION*), and
- Demographics table to extract the race (*RACE*) of each patient.

The simulated QoL survey data is saved in the first sheet of an Excel file named “QoL survey data.xlsx”, downloadable from https://github.com/nyilin/RDataXMan_example_data. Note that when extracting input data from an Excel file RDataXMan only reads the first sheet, therefore when preparing data sources users should avoid storing input data as multiple sheets in the same Excel file.

The three simulated EMR tables mentioned above are made available to users from the same link as three CSV files:

- “V2M_C_MOVEMENT_PC_3yr.csv” for the inpatient movement table,
- “V2M_C_DIAGNOSIS_P_3yr.csv” for the diagnosis table, and
- “V2M_C_PATIENT_BASIC_3yr.csv” for the demographics table.

Users can download each file by first clicking on the file name, and then clicking on the “Download” button (using the diagnosis table as an example):



In this example, we illustrate two application scenarios of RDataXMan when working with the EMR tables. In the first application scenario, we assume the EMR tables are provided to users as local CSV files for this data extraction. While this data access strategy is not an issue in our illustrative example, where all data are simulated, when working with real data users must make sure the data access strategy selected is in line with ethical and data security regulations. More information regarding this data access strategy is provided in Section 2.2.

In practice, EMR tables are often stored on a designated server to facilitate data security. In the second application scenario of this example, we assume the IT team have imported the three EMR tables into a MySQL database named “emr”, where the movement, diagnosis and demographics tables are named “v2m_c_movement_pc_3yr”, “v2m_c_diagnosis_p_3yr” and “v2m_c_patient_basic_3yr” respectively, and users have a valid account to access these tables.

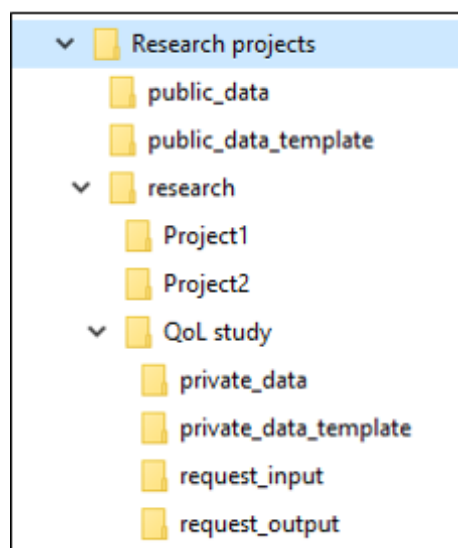
In Section 2.2, we will provide an overall introduction to the workflow of RDataXMan and the folder structure that facilitates this workflow. In Section 3, we will provide a detailed instruction on the use of RDataXMan via the GUI on a Windows operating system and the corresponding R script by using this illustrative example. Section 3.1 will illustrate how to launch the RDataXMan GUI and Section 0 will describe how to specify data requests when working with the QoL data. The next two subsections will describe how to specify data requests when accessing the three EMR tables as local files (see Section 3.3) or tables on a MySQL database (see Section 3.4), which will lead to the same data extraction output (see Section 3.5). Section 4 will introduce an extra tool developed by our team that facilitates data de-identification for users’ information.

2.2. Folder Structure Employed by RDataXMan

In each data extraction, RDataXMan generates Excel request forms for users to specify the data extraction requirements (e.g., the data source, the inclusion criteria and the list of variables to extract), 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, including local data sources and request forms, 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” (see the screenshot in the next page), but users can choose any appropriate name and location for the working directory on their computers.

The working directory must include three subfolders: “research”, “public_data” and “public_data_template”. Users may choose to manually create these folders, but RDataXMan provides functions to automatically create the required folder structure for users’ convenience. This subsection provides the R commands for creating the folder structure, and the corresponding point-and-click approach using the GUI will be introduced in Section 3.2.1.



The working directory and these three subfolders can be created by first loading the RDataXMan package and then executing the following commands:

```
library(RDataXMan)
initWkdir(wkdir = "D:/Documents/Research projects")
```

Users should replace "D:/Documents/Research projects" in the command above with the actual location of the working directory they have selected.

The subfolder “research” contains the dedicated folders (which we will refer to as research project folders) for each data request. The folder dedicated to our illustrative example is “QoL study”, and the other two folders, “Project1” and “Project2”, are placeholders for other research projects. Each specific research project folder (e.g., “QoL study”, or “Project1” or “Project2”) has four essential subfolders, i.e., “private_data”, “private_data_template”, “request_input” and “request_output”. These subfolders can be created by executing the following command (after loading the RDataXMan package):

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

Users should replace "D:/Documents/Research projects" in the command above with the actual location of the working directory they have selected. There is no need to specify the full path to the research project folder (“QoL study”).

The “private_data” subfolder within the research folder contains data files that is specific to this project, in our example the survey data (“QoL survey data.xlsx”). 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 specific research project folder to restrict access to authorised persons only, but users should avoid encrypting data files (e.g., password-protecting Excel files), as this is not yet supported in the current version of RDataXMan.

If users choose to access the three EMR tables (which contain simulated data for all patients in a hospital) as local flat files, they can move the three CSV files to the “public_data” subfolder of the working directory to avoid duplication, as these files will also be used by other projects that involve EMR data. In real-life applications, users should be careful when storing flat files in this “public_data” subfolder, as the current version of RDataXMan does not support encryption of this subfolder or any file within the subfolder. Users who have a MySQL database set up for the EMR tables can leave the “public_data” subfolder 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 later sections) should be saved to the “request_input” subfolder of the research project folder, and data extraction output will be stored in the “request_output” subfolder.

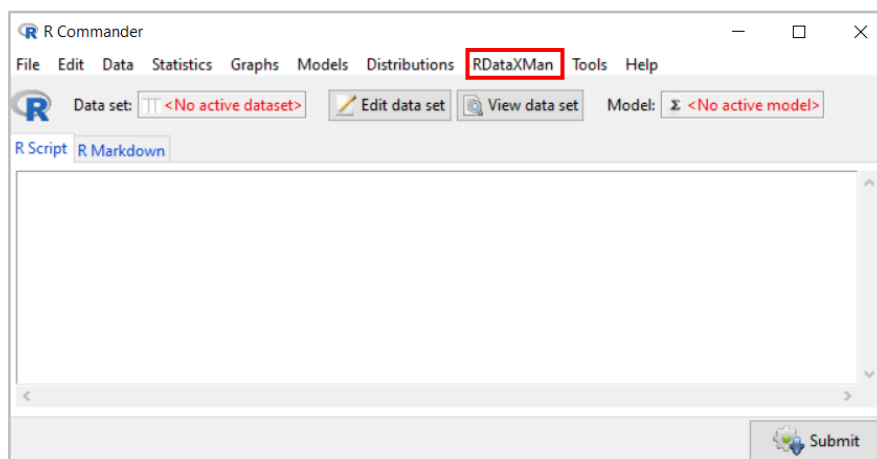
3. Using RDataXMan via Script and GUI

3.1. Launch RDataXMan GUI

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

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

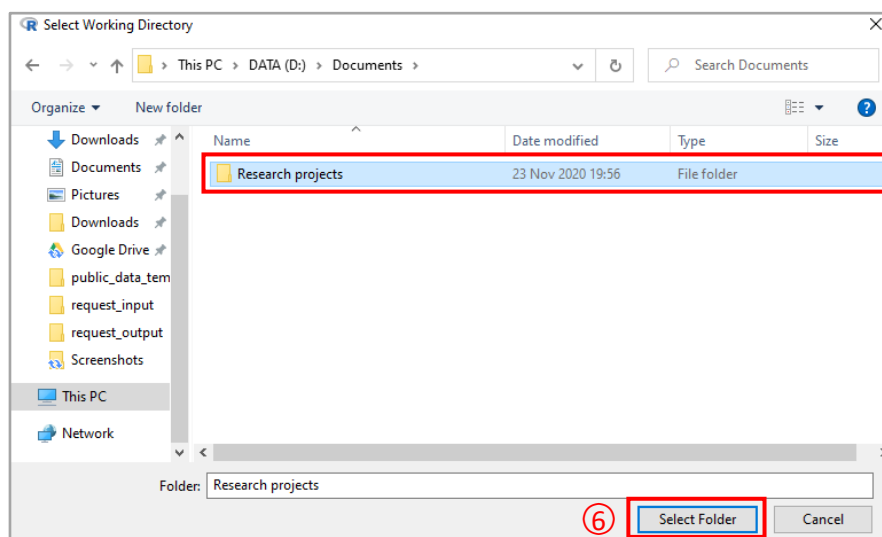
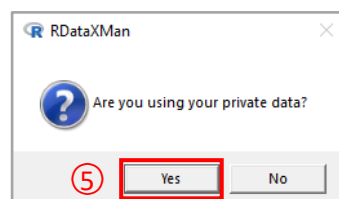
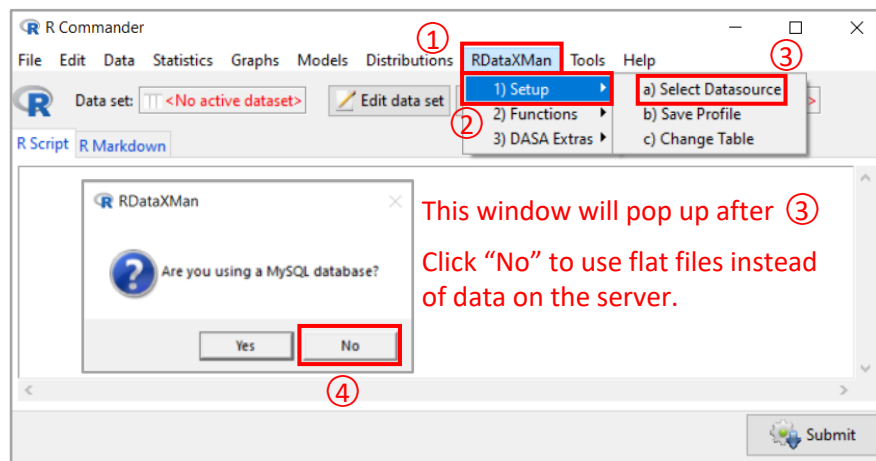
The following window should pop up, and all the features of RDataXMan is available from the “RDataXMan” drop-down menu:

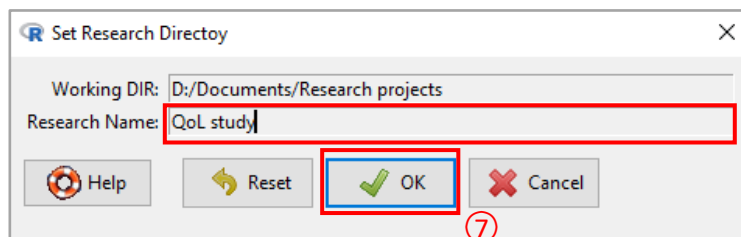


3.2. Work with Private Data

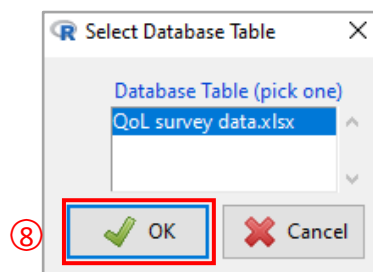
3.2.1. Select Data

Initialisation of the folder structure using the GUI requires users to first create the working directory (“D:/Documents/Research projects” in our example) manually. Subsequently, GUI users can use the drop-down menu to initialise the working directory and the research project 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. Follow the instructions below to initialise the working directory and research project folder, and then select the private QoL data:





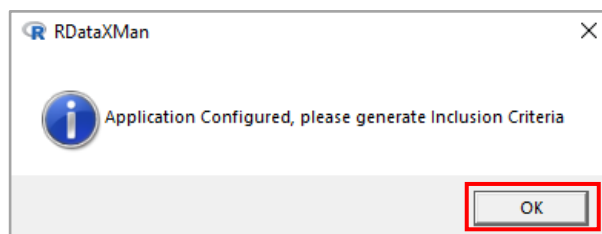
This window will pop up after ⑥
Use this window to specify the research folder, i.e., “QoL study”, and then click “OK” to proceed.



This window will pop up after ⑦
Select “QoL survey data.xlsx” and click “OK” to proceed.

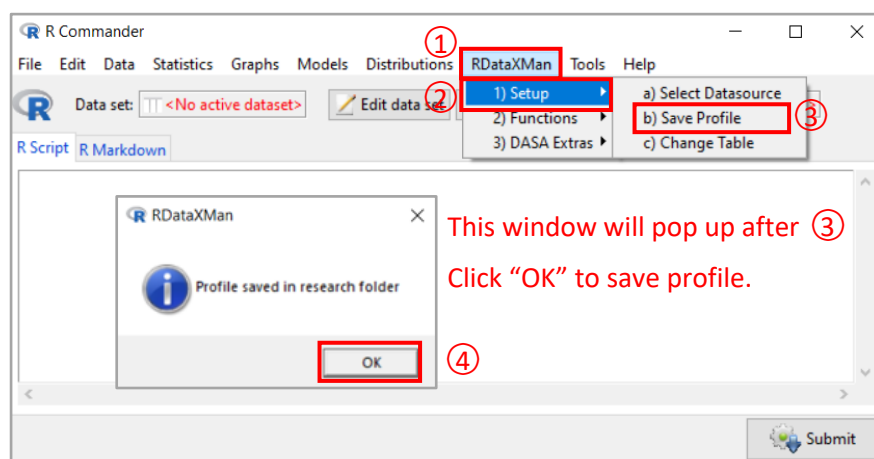
If this is the first time the research folder “QoL study” is initialised, the window that pops up after step ⑦ will be empty. Users need to close this window, move “QoL survey data.xlsx” to the “private_data” subfolder of “QoL study”, and then repeat steps ① to ⑧ to select the QoL data.

The following window will pop up after step 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.



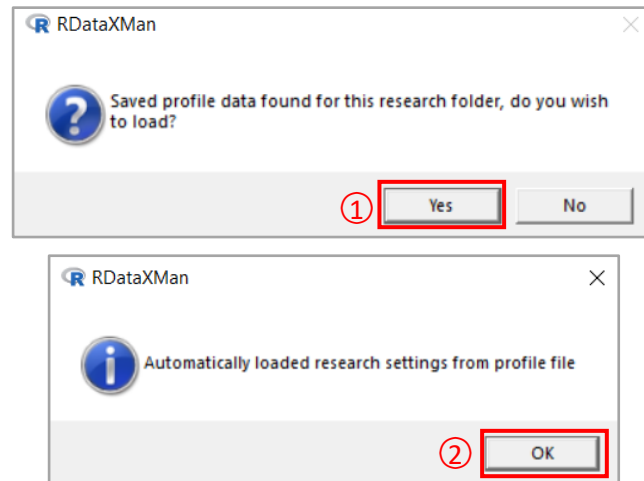
3.2.2. Save Profile (optional)

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:



This window will pop up after ③
Click “OK” to save profile.

In a different project, when GUI users go through steps ① to ⑦ in Section 3.2.1 again to select the same research project 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:

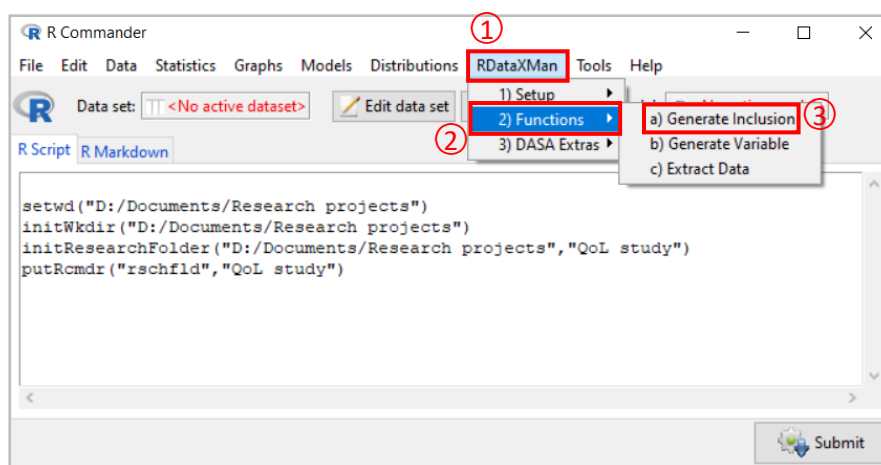


The two steps above specify the private QoL data as the current data source without going through step ⑧ in Section 3.2.1. The usefulness of saving the current data source configuration becomes prominent when working with data on the server (see Section 3.4).

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 entry 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 step ③ for GUI users to select the key variable and the identifier variable for the QoL data:

Generate Inclusion Criteria

Key Variable: (pick one) PATIENT_NRIC

Key Descriptions: (pick zero or more)

Table: QoL survey data.xlsx

Identifier Variables: (pick one or more) PATIENT_NRIC

Overwrite: TRUE

Save Execution: TRUE

Help Reset OK Cancel Apply

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:

RDataXMan

Operation Complete, Please proceed to complete the template in folder public or private template and move it to folder D:/Documents/Research projects/research/QoL study/request_input

OK

Steps ① to ⑥ correspond to the following R command:

```
genInclusion(wkdir = "D:/Documents/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 the file name begins with “genInc” and is followed by the date and time stamp.

The Excel request form generated is named “inclusion.QoL survey data.xlsx_PATIENT_NRIC.xlsx.xls” and is saved in the “private_data_template” subfolder of the research project folder. The file name indicates the name of the private flat file, the key variable and the extension of the input file to facilitate easier management of request forms. By default, existing request form that have the same file name will be overwritten to avoid duplication.

The request form has 300 rows corresponding to the 300 unique NRIC:

	A	B	C	D	E
1	sno	PATIENT_NRIC	remarks	selection	logic
2	1	PX3051156907490479065406824946347654888991719016106576782726137145			
3	2	PX3071548899976799825306017057739570169578637441608779043554058548			
4	3	PX3079760107884622372300090389822576147439786522423636920885995649			
5	4	PX3100172413846679970540383234126074800163983921305371313828374007			
6	5	PX3168468110133752708896161329553951785029142948488066899484948180			

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

	A	B	C	D	E
1	sno	PATIENT_NRIC	remarks	selection	logic
2	1	PX3051156907490479065406824946347654888991719016106576782726137145		x	
3	2	PX3071548899976799825306017057739570169578637441608779043554058548		x	
4	3	PX3079760107884622372300090389822576147439786522423636920885995649		x	
5	4	PX3100172413846679970540383234126074800163983921305371313828374007		x	
6	5	PX3168468110133752708896161329553951785029142948488066899484948180		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*:

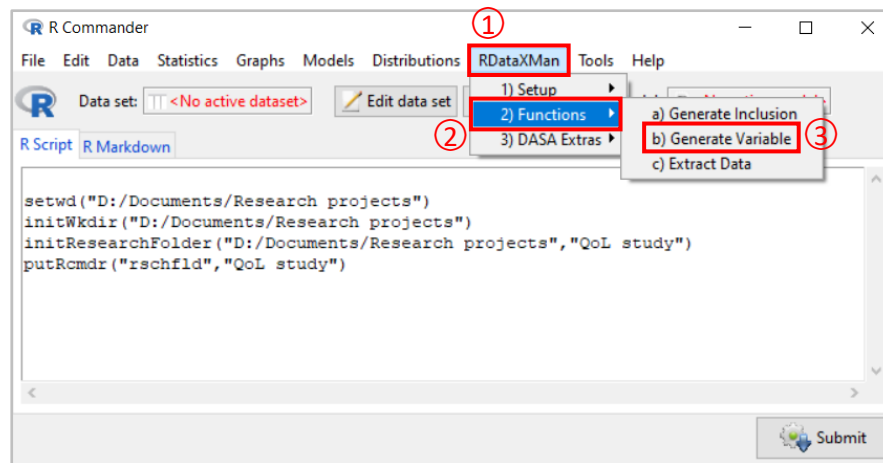
	A	B	C	D	E
1	sno	PATIENT_NRIC	remarks	selection	logic
2	1	PX3051156907490479065406824946347654888991719016106576782726137145			!is.na(PATIENT_NRIC)
3	2	PX3071548899976799825306017057739570169578637441608779043554058548			
4	3	PX3079760107884622372300090389822576147439786522423636920885995649			
5	4	PX3100172413846679970540383234126074800163983921305371313828374007			
6	5	PX3168468110133752708896161329553951785029142948488066899484948180			

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.

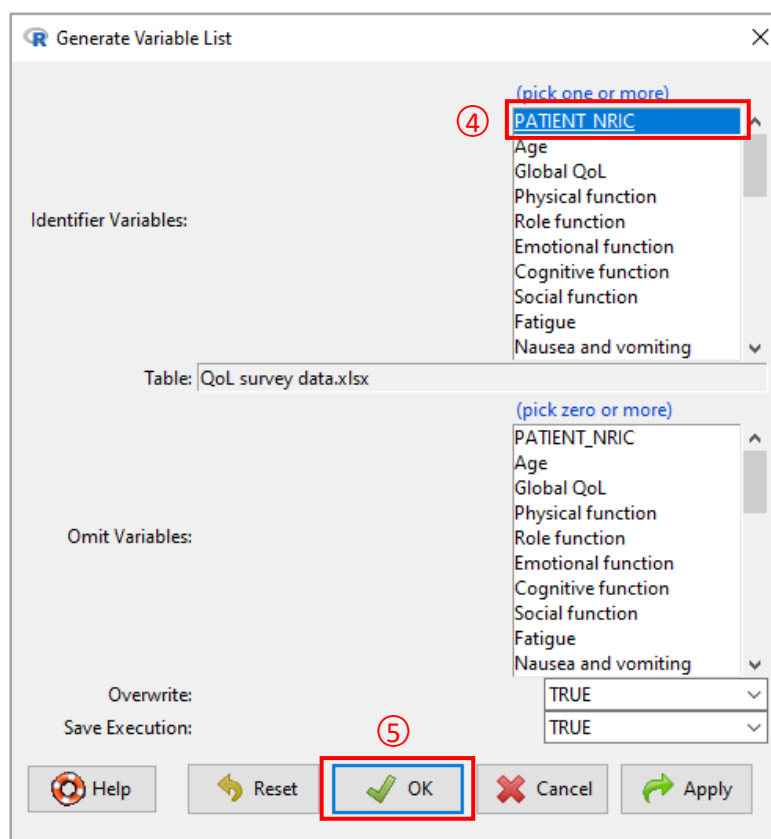
After specifying the selection, either by using the “selection” or “logic” column, users can use the “Save As” option of Excel to save the annotated request form to the “request_input” subfolder in the research folder, using the same file name.

3.2.4. Select Variable

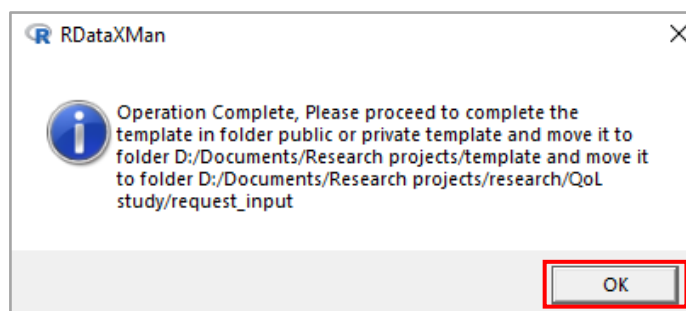
To generate a request form and indicate the variable(s) to extract from a data source, users only need to specify the identifier variable(s) of this data source. In our illustrative example, we select a single variable *Global QoL* from the QoL data by following the steps below:



The following window will pop up after step 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 = "D:/Documents/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 Excel request form generated is named “variable.QoL survey data.xlsx(PATIENT_NRIC)_xlsx.xls” and is saved in the “private_data_template” subfolder of the research project folder. The file name indicates the name of the private flat file, the identifier variable (in brackets) and the extension of the input file to facilitate easier management of request forms. By default, existing request form with the same file name will be overwritten.

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 extracted. 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 can 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.

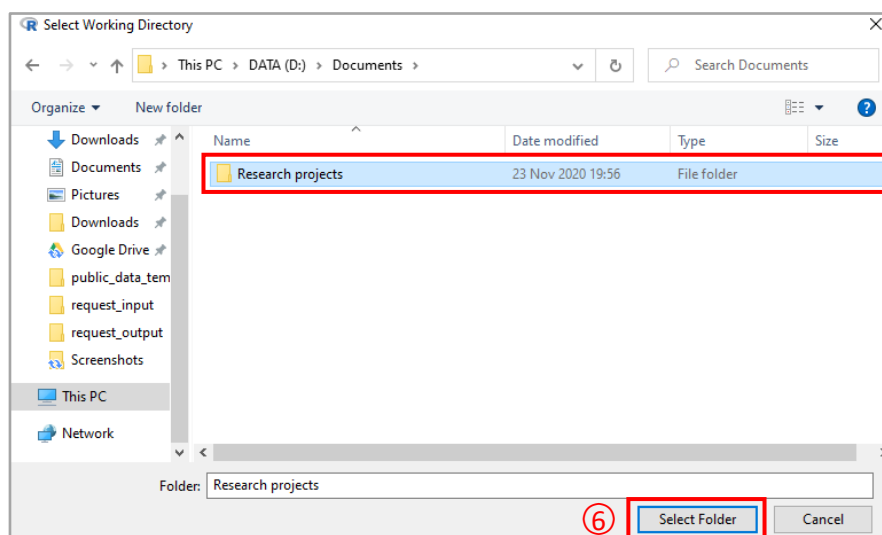
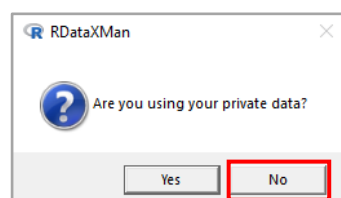
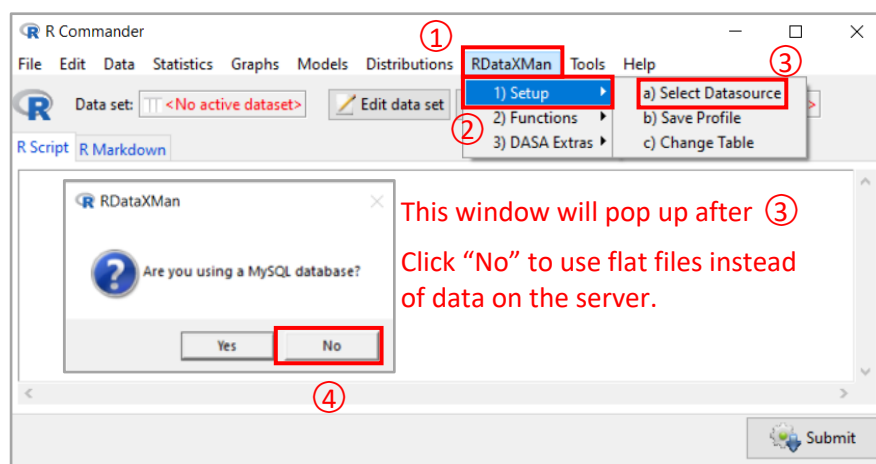
After generating the inclusion criterion and variable list from the private QoL data, users who choose to save the three EMR tables as public flat tables should proceed to Section 3.3. Users who choose to save the EMR tables on a MySQL server can skip Section 3.3 and proceed to Section 3.4 instead.

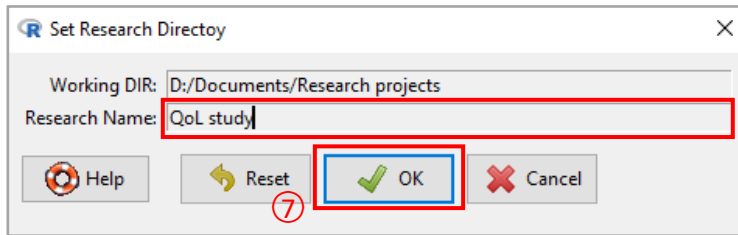
3.3. Work with Public Data

To illustrate how to work with public flat files that are shared among research projects, we generate inclusion criterion and variable list from the movement table of EMR to include only inpatient admission in 2006, which we assume is saved as a CSV file named “V2M_C_MOVEMENT_PC_3yr.csv” in the “public_data” subfolder of the working directory. The movement table 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*).

3.3.1. 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:

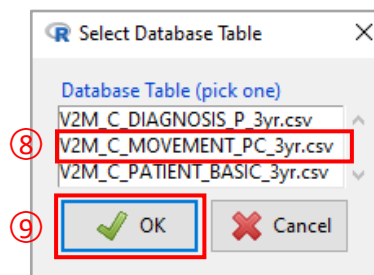




This window will pop up after ⑥. Use this window to specify the research folder, i.e., “QoL study”, and then click “OK” to proceed.

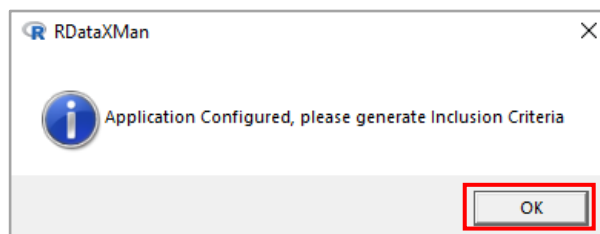
If users have saved data profile in Section 3.2.2, a window will pop up after step ⑦ to ask users whether to load the existing data profile in this research project folder. Select “No”, and the window below will pop up for users to select from public data.

If users did not save data profile, the following window will pop up immediately after step ⑦.



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

The following window will pop up after step 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.



3.3.2. Save Profile (optional)

As described in Section 3.2.2, GUI users can save the current configuration of data source, i.e., the selection of the movement table in the “public_data” subfolder. Note that this will overwrite the existing profile saved in the research folder. After saving the profile for selecting the public 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 ⑧ and ⑨ in the previous section. Again, as mentioned in Section 3.2.2, the ability to load previously saved profile is more useful when working with data on a server.

3.3.3. 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:

The screenshot shows the 'Generate Inclusion Criteria' dialog box. It has several sections: 'Key Variable' with a dropdown menu showing 'AYEAR' selected; 'Key Descriptions' with an empty dropdown; 'Table' with the text 'V2M_C_MOVEMENT_PC_3yr.csv'; 'Identifier Variables' with a dropdown menu showing 'PATIENT_NRIC' and 'CASE_NO' selected; 'Overwrite' and 'Save Execution' checkboxes, both of which are checked; and a row of buttons at the bottom: 'Help', 'Reset', 'OK' (highlighted with a red box), 'Cancel', and 'Apply'.

Select multiple variables by holding "Ctrl" while clicking.

The corresponding R command is:

```
genInclusion(wkdir = "D:/Documents/Research projects",
            research.folder = "QoL study",
            table_name = "V2M_C_MOVEMENT_PC_3yr.csv",
            key.var = "AYEAR", identifier.var = c('PATIENT_NRIC', 'CASE_NO'),
            data.type = "flat", database = "public")
```

and 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 with file name "inclusion.V2M_C_MOVEMENT_PC_3yr.csv_AYEAR_csv_20201124_110522.xls". In addition to the name of the public flat file, the key variable and the file extension of the input file, the file name now includes the date and time stamp (which will be different in each request) to avoid unintended overwriting of request forms.

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.

3.3.4. 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 in this table, 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.

The corresponding R command is:

```
genVariable(wkdir = "D:/Documents/Research projects",
            research.folder = "QoL study",
            table_name = "V2M_C_MOVEMENT_PC_3yr.csv",
            identifier.var = c('PATIENT_NRIC', 'CASE_NO'),
            omit.var = c('ADATE', 'DDATE'),
            data.type = "flat", database = "public")
```

and is saved as a TXT file in the research folder.

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

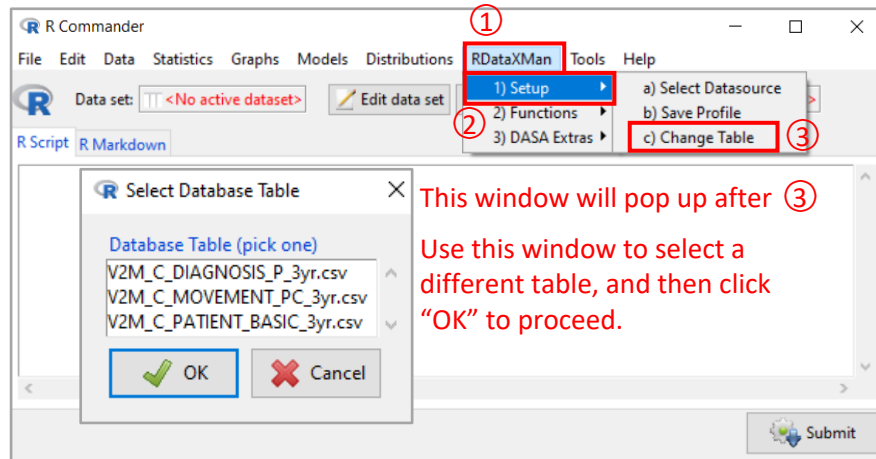
	A	B	C	D
1	sno	variable	remarks	selection
2		1 LOS		x
3		2 AYEAR		
4		3 DYEAR		

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 is named “variable.V2M_C_MOVEMENT_PC_3yr.csv(PATIENT_NRIC_CASE_NO)_csv_20201124_11055

5.xls”, where the date and time stamp at the end of the file name will be different in each request to avoid unintended overwriting of request forms.

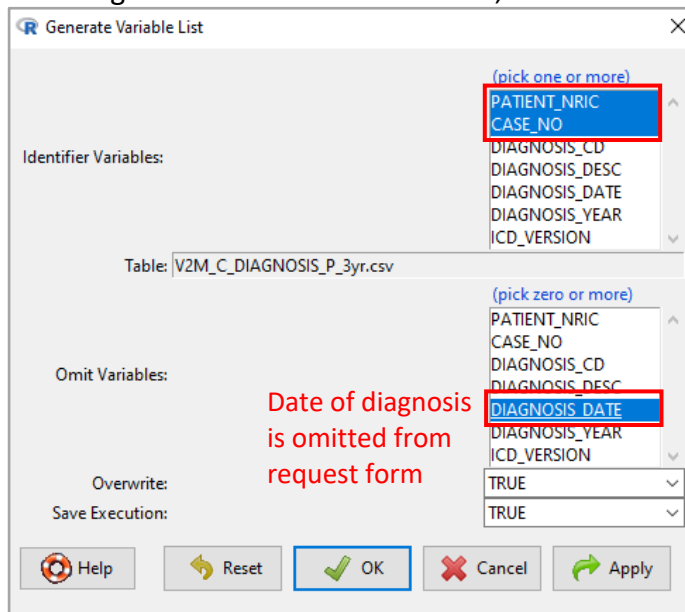
3.3.5. Change Table

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:



Following the instructions in Section 3.3.4, users can generate variables 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.csv",  
  identifier.var = c('PATIENT_NRIC',  
                    'CASE_NO'),  
  omit.var = c('DIAGNOSIS_DATE'),  
  data.type = "flat",  
  database = "public"  
)
```

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.

Generate Variable List

Identifier Variables: (pick one or more)

PATIENT_NRIC
GENDER
DEATH_DATE
DEATH_IND
RACE
BIRTH_YEAR

Table: V2M_C_PATIENT_BASIC_3yr.csv

Omit Variables: (pick zero or more)

PATIENT_NRIC
GENDER
DEATH_DATE
DEATH_IND
RACE
BIRTH_YEAR

Date of death is omitted from request form

Overwrite: TRUE

Save Execution: TRUE

Help Reset OK Cancel Apply

Corresponding R command:

```
genVariable(  
  wkdir = "Research projects",  
  research.folder = "QoL study",  
  table_name = "V2M_C_PATIENT_BASIC_3yr.csv",  
  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		

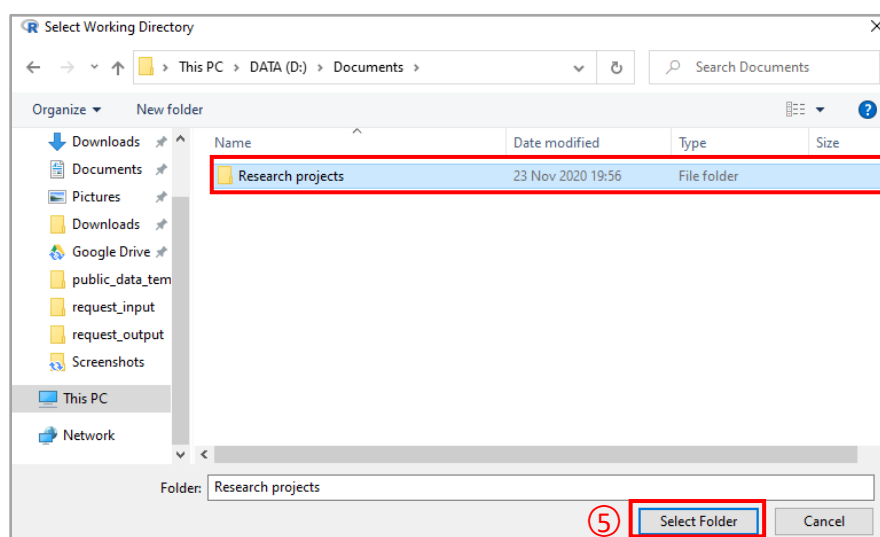
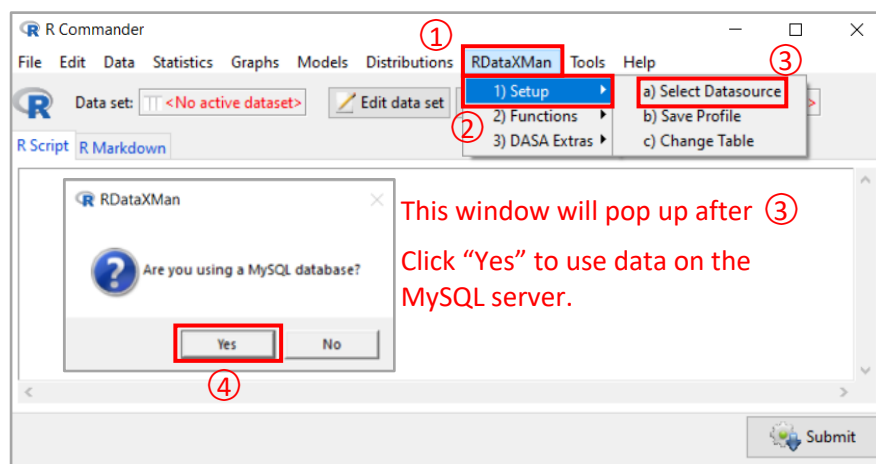
Users can proceed to Section 3.5 for instructions on how to extract data given the extraction requirements specified in this section.

3.4. Work with Data on MySQL server

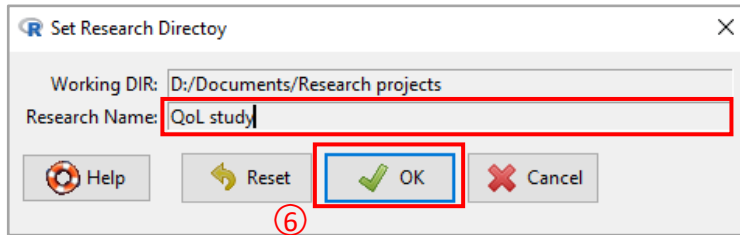
To illustrate how to work with data on a MySQL server using RDataXMan, we generate inclusion criterion and variable list from the movement table of EMR to include only inpatient admission in 2006, which we assume is saved in database “emr” with name “v2m_c_movement_pc_3yr”. The movement table contains 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*).

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:



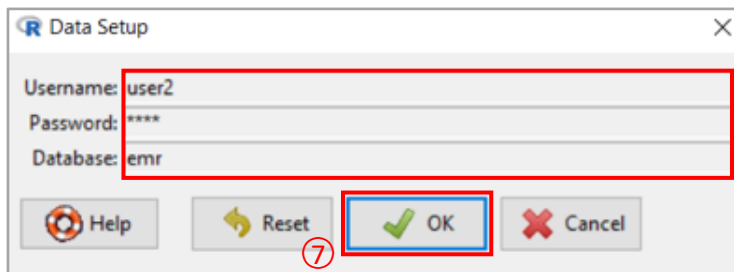
This window will pop up after ④
Use this window to allocate the working directory, and click “Select Folder” to proceed.



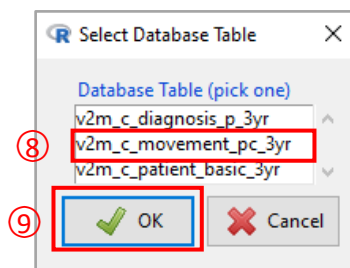
This window will pop up after ⑤
Use this window to specify the research folder, i.e., “QoL study”, and then click “OK” to proceed.

If users have saved data profile in Section 3.2.2, a window will pop up after step ⑥ to ask users whether to load the existing data profile in this research project folder. Select “No”, and the window below will pop up for users to enter the username, password and name of database to connect to the MySQL server.

If users did not save data profile, the following window will pop up immediately after step ⑥.

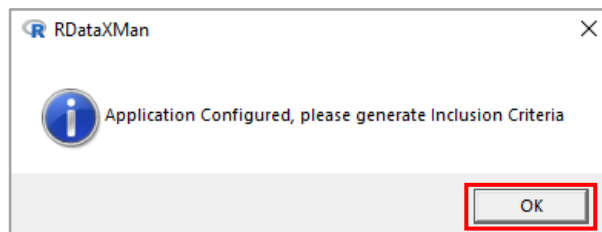


This window will pop up after ⑥
Use this window to enter the username, password and the name of database (i.e., “emr”), and then click “OK” to proceed.



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 step 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.



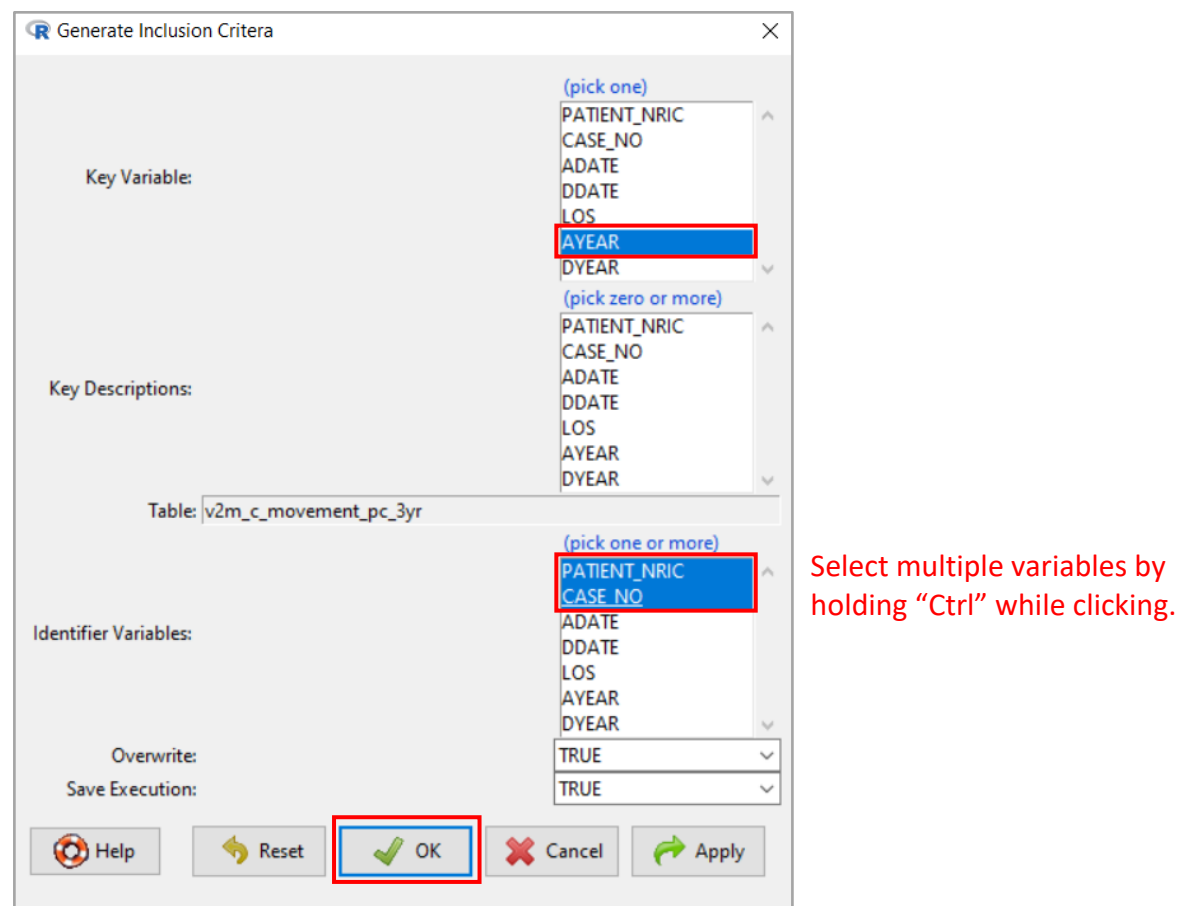
3.4.2. Save Profile (optional)

As described in Section 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 overwrite the existing profile saved in the research project 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 (details will be described in Section 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.

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:



The screenshot shows the 'Generate Inclusion Criteria' dialog box. It has several sections: 'Key Variable' with a dropdown menu showing 'AYEAR' selected; 'Key Descriptions' with a dropdown menu showing 'PATIENT_NRIC' selected; 'Table' with a text field containing 'v2m_c_movement_pc_3yr'; 'Identifier Variables' with a dropdown menu showing 'PATIENT_NRIC' and 'CASE_NO' selected; 'Overwrite' and 'Save Execution' checkboxes, both of which are checked; and a row of buttons at the bottom: 'Help', 'Reset', 'OK' (highlighted with a red box), 'Cancel', and 'Apply'. A red text annotation points to the 'PATIENT_NRIC' and 'CASE_NO' selections in the 'Identifier Variables' dropdown, stating: 'Select multiple variables by holding "Ctrl" while clicking.'

The corresponding R command is:

```
genInclusion(wkdir = "D:/Documents/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 with file name “inclusion.v2m_c_movement_pc_3yr_AYEAR_sql_user2_20201124_150145.xls”. In addition to the table name, the key variable and the type of server (“sql”), the file name now includes the date and time stamp (which will be different in each request) to avoid unintended overwriting of request forms.

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.

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 in this table, 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:

The screenshot shows the 'Generate Variable List' dialog box. The 'Table' field is set to 'v2m_c_movement_pc_3yr'. Under 'Identifier Variables', 'PATIENT_NRIC' and 'CASE_NO' are selected. Under 'Omit Variables', 'ADATE' and 'DDATE' are selected. The 'Overwrite' and 'Save Execution' options are both set to 'TRUE'. The 'OK' button is highlighted with a red box.

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

The corresponding R command is:

```
genVariable(wkdir = "D:/Documents/Research projects",
            research.folder = "QoL study",
            table_name = "v2m_c_movement_pc_3yr",
            identifier.var = c('PATIENT_NRIC', 'CASE_NO'),
            omit.var = c('ADATE', 'DDATE'),
            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.

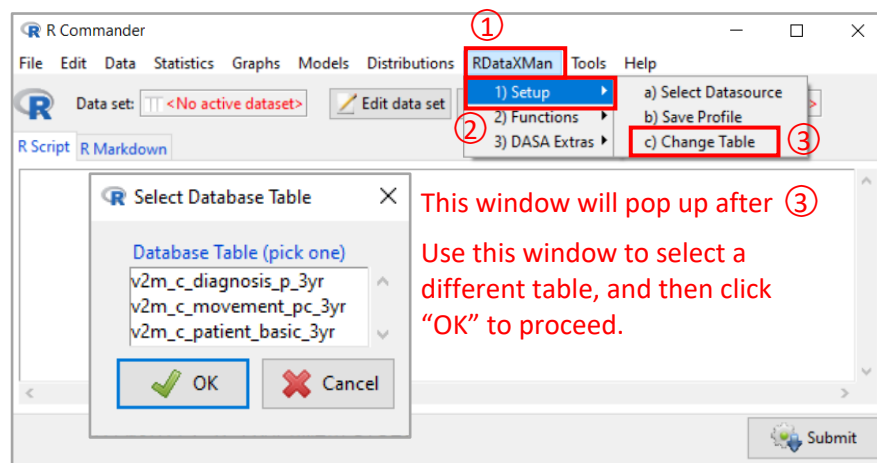
Select *LOS* in the request form generated, which is saved to the “public_template” subfolder within the working directory:

	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 is named “variable.v2m_c_movement_pc_3yr_(PATIENT_NRIC_CASE_NO)_sql_user2_20201124_150154.xls”, where the date and time stamp at the end of the file name will be different in each request to avoid unintended overwriting of request forms.

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:



Following the instructions in Section 3.4.4, users can generate variables 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:

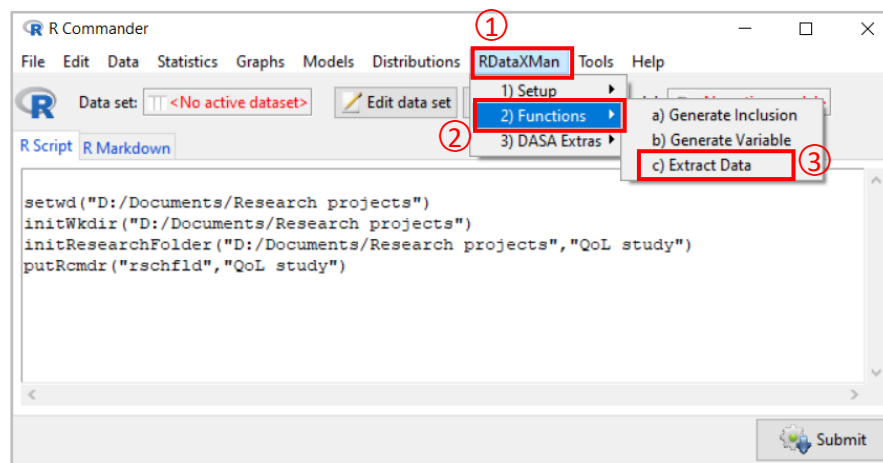
	A	B	C	D
1	sno	variable	remarks	selection
2	1	GENDER		x
3	2	DEATH_IND		
4	3	RACE		
5	4	BIRTH_YEAR		

Users can proceed to Section 3.5 for instructions on how to extract data given the extraction requirements specified in this section.

3.5. Extract Data

In the previous sections, we have provided detailed instruction on how to generate and annotate the two request forms on inclusion criteria (based on QoL data and movement table) and the four request forms on variables to extract (from QoL data, movement table, diagnosis table and demographics table) described in Section 2.1. In this section, we provide instructions on how to extract data based on the requirements specified in the annotated request forms.

Follow the instructions bellow to extract data:



A window will pop up after step ③ for GUI users to select the request forms relevant to this data extraction. As RDataXMan includes essential information (e.g., name of data source, key and identifier variables) in the name of each request form, the appropriate request forms to use in a data extraction can be easily identified by inspecting their file names. In the next page, we show the pop-up windows for users who choose to save the three EMR tables as public flat files and users who choose to save the EMR tables on a MySQL server (where the request forms display will be different due to the different data sources selected), and explain the choices available for data extraction.

Pop up window for data extraction when EMR tables are available as public flat files:

The 'Extract Data' window is shown with the following settings:

- Inclusion Files:** (pick one or more)
 - inclusion.QoL_survey_data.xlsx_PATIENT_NRIC.xlsx.xls
 - inclusion.V2M_C_MOVEMENT_PC_3yr.csv_AYEAR_csv_20201124_110522.xls
- Inclusion Data Logic:** Intersection
- Variable Files:** (pick one or more)
 - variable.QoL_survey_data.xlsx(PATIENT_NRIC)_xlsx.xls
 - variable.V2M_C_DIAGNOSIS_P_3yr.csv(PATIENT_NRIC_CASE_NO)_csv_20201124_110806.xls
 - variable.V2M_C_MOVEMENT_PC_3yr.csv(PATIENT_NRIC_CASE_NO)_csv_20201124_110555.xls
 - variable.V2M_C_PATIENT_BASIC_3yr.csv(PATIENT_NRIC)_csv_20201124_110826.xls
- Overwrite:** TRUE
- Modes:** (pick one or more)
 - Generate Identifier Variable List
 - Generate Summary Statistics
 - Extract Data
 - Merge Data
- Save Execution:** TRUE

Note: Clicking OK or Apply may cause the overwrite of previous data extractions

Buttons: Help, Reset, OK, Cancel, Apply

Existing output files will be overwritten

Pop up window for data extraction when EMR tables are available as public flat files:

The 'Extract Data' window is shown with the following settings:

- Inclusion Files:** (pick one or more)
 - inclusion.QoL_survey_data.xlsx_PATIENT_NRIC.xlsx.xls
 - inclusion.v2m_c_movement_pc_3yr_AYEAR_sql_user2_20201124_150145.xls
- Inclusion Data Logic:** Intersection
- Variable Files:** (pick one or more)
 - 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
- Overwrite:** TRUE
- Modes:** (pick one or more)
 - Generate Identifier Variable List
 - Generate Summary Statistics
 - Extract Data
 - Merge Data
- Save Execution:** TRUE

Note: Clicking OK or Apply may cause the overwrite of previous data extractions

Buttons: Help, Reset, OK, Cancel, Apply

Existing output files will be overwritten

When more than one inclusion criteria are involved (e.g., in our illustrative example), 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.

For users who have EMR tables as public flat tables, the corresponding R command is:





```
rdataxman_result <- extract_data(
  wkdir = "D:/Documents/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.csv_AYEAR_csv_20201124_110522.xls'
  ),
  variable.xls.file = c(
    'variable.QoL survey data.xlsx(PATIENT_NRIC)_xlsx.xls',
    'variable.V2M_C_DIAGNOSIS_P_3yr.csv(PATIENT_NRIC_CASE_NO)_csv_20201124_110806.xls',
    'variable.V2M_C_MOVEMENT_PC_3yr.csv(PATIENT_NRIC_CASE_NO)_csv_20201124_110555.xls',
    'variable.V2M_C_PATIENT_BASIC_3yr.csv(PATIENT_NRIC)_csv_20201124_110826.xls'
  ),
  dataLogic = "Intersection", select.output = c('1','2','4'), overwrite = TRUE,
  database = "public"
)
```

For users who have EMR tables on a MySQL server, 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"
)
```

The command 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.

The same four output files, listed below, are extracted and saved to the “request_output” folder, regardless of where the EMR tables are saved:

 inclusion_identifier_var	24 Nov 2020 15:55	Microsoft Excel Comma Separated Values File	54 KB
 merge_dat	24 Nov 2020 15:55	Microsoft Excel Comma Separated Values File	89 KB
 merge_inclusion	24 Nov 2020 15:55	Microsoft Excel Comma Separated Values File	56 KB
 summary_list	24 Nov 2020 15:56	Microsoft Excel Worksheet	21 KB

The output files 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	CN7852296084006377057592	PX3051156907490479065406824
3	CN6636256867460692645812	PX3071548899976799825306017
4	CN2130331507974361862141	PX3079760107884622372300090
5	CN7498666549176923476755	PX3100172413846679970540383

Mode 2: generate a file named “summary_list.xlsx” with multiple sheets that contain summary statistics on data extracted: sheet “inclusion_count_overall” (top) summarises the sample size and sheet “variable_summary” (bottom) contains summary statistics for each variable extracted.

	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		174.44 (0.30)	Mean(S.D.)
4	DIAGNOSIS_DESC	395	Malignant neoplasm of axillary tail of female breast	48 (12.15%)	N(%)
5			Malignant neoplasm of breast (female), unspecified	51 (12.91%)	N(%)
6			Malignant neoplasm of central portion of female breast	37 (9.37%)	N(%)
7			Malignant neoplasm of lower-inner quadrant of female breast	40 (10.13%)	N(%)
8			Malignant neoplasm of lower-outer quadrant of female breast	48 (12.15%)	N(%)
9			Malignant neoplasm of nipple and areola of female breast	52 (13.16%)	N(%)
10			Malignant neoplasm of other specified sites of female breast	43 (10.89%)	N(%)
11			Malignant neoplasm of upper-inner quadrant of female breast	35 (8.86%)	N(%)
12			Malignant neoplasm of upper-outer quadrant of female breast	41 (10.38%)	N(%)
13	ICD_VERSION	395	9CM	395 (100%)	N(%)
14	LOS	395		6.86 (2.71)	Mean(S.D.)
15	GENDER	395	Female	395 (100%)	N(%)

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

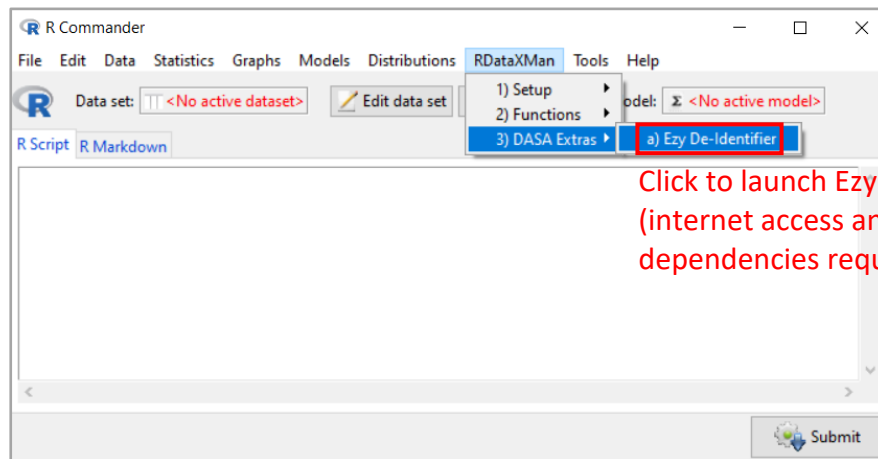
Mode 4: two files for merged data, with one file for variables from inclusion criteria (“merge_inclusion.csv”, top) and one file for variables requested (“merge_dat.csv”, bottom).

	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 neoplas	9CM	5	Malay
3	PX30715488999767	CN66362568674606	62.5	174.6	Malignant neoplas	9CM	7	Chinese
4	PX30797601078846	CN21303315079743	150	174.3	Malignant neoplas	9CM	7	Indian
5	PX31001724138466	CN74986665491769	87.5	174.9	Malignant neoplas	9CM	5	Malay

4. 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/>.



Click to launch Ezy De-identifier
(internet access and additional
dependencies required)