

Introduction to VA Analysis using the openVA R Shiny App in Windows 10

Jason Thomas, Sam Clark, Richard Li, & Tyler McCormick
in collaboration with the Bloomberg Data for Health Initiative

October 11, 2017

1 Introduction

There are several methods for analyzing verbal autopsy (VA) data with the goal of assigning a cause of death. **openVA** for the R free software environment consolidates many of these tools into a single package. This document serves to introduce the **openVA** package through the installation and use of a user-friendly R Shiny app. After an initial setup, the user can analyze VA data with different algorithms and access results without having to learn the R programming language. Part of the focus in the following sections relates to data collected and managed using the Open Data Kit (ODK). Data collected using ODK is prepared for analysis using another R package, namely, CrossVA. The information presented here provides all the necessary steps for an introduction to **openVA**, but more information can be found at the following sites:

- R software environment <https://cran.r-project.org/>
- **openVA** <https://cran.r-project.org/package=openVA>
- R Shiny <https://shiny.rstudio.com/>
- Open Data Kit (ODK) <https://opendatakit.org/>

2 Installing and Starting R Shiny App for Windows 10

2.1 Quick Start-up Guide (for a more detailed description see below)

Download the R Shiny zip file, shiny-openVA.zip, at

<https://github.com/jarathomas/openVA-Pipeline/blob/master/ShinyDownload/>

and unzip it – the contents should include three files: (1) shiny-openVA.R (R Shiny script file); (2) CrossVA_0.9.3.zip (an R package); and (3) who_va_output.csv (an example data file from ODK).

1. Install the Java SE 7 (or higher) development kit (JDK) (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
2. Download the R executable (with same architecture as Java, e.g., 64-bit) from <https://cran.r-project.org/bin/windows/base/> and double-click the file to install R
3. Start R and install the necessary packages by typing the following in the R Console window at the prompt `>` and then pressing the enter key

```
install.packages(c("openVA", "shinyjs", "lubridate"), dependencies=TRUE)
```


A new window should appear asking the user to select a Secure CRAN mirror (select the mirror geographically closest to you), and the installation will begin after clicking the OK button.

4. Install the CrossVA package (included in the R Shiny zip file) by clicking, in the R menu bar, **Packages > Install package(s) from local files . . .** then select the CrossVA_0.9.3.zip file (you may need to navigate to the folder where the R Shiny zip file was downloaded) and click the open button at the bottom-right of the Select files window.
5. Open the shiny-openVA.R script (included in the R Shiny zip file) through the R menu bar **File > Open Script . . .** then select the shiny-openVA.R script file and click the Open button at the bottom-right of the Open script window. This step should open a new R Editor window.
6. Make sure the focus is on the R Editor window (by clicking on it) and run the shiny-openVA.R script through the menu bar **Edit > Run all** which will execute the code (in the R Console) and open your web browser with the R Shiny app.

2.2 More Details on Installation

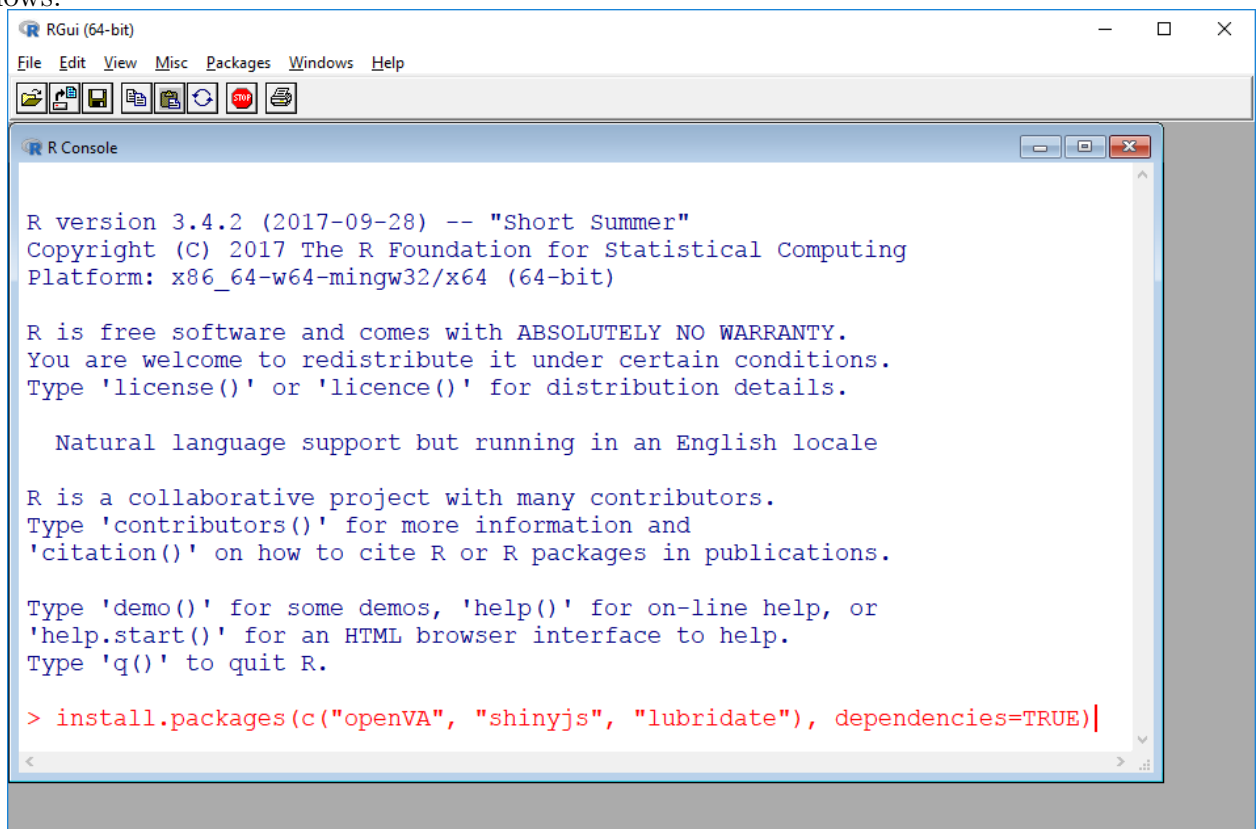
The R software is available for download from the Comprehensive R Archive Network (CRAN) <https://cran.r-project.org/bin/windows/base/>. Installation instructions are also available from CRAN, but the process should be familiar: download the executable file (ending in .exe and double-click the file). The R Shiny app uses several packages, one of which requires the Java SE 7 (or higher) development kit (JDK) (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

com/technetwork/java/javase/downloads/index.html). Be sure to install the same architecture versions (i.e., 32-bit or 64-bit) of R and the Java Platform. It is not unusual to experience difficulties while trying to configure R to work with Java. For more details about the configuration procedure, see the InSilicoVA manual (<https://github.com/richardli/InSilicoVA/blob/master/Documents/Insilico-manual.pdf>).

After Java and R have been installed, the next step is to install R packages that contain the tools needed for the Shiny app to run. Start the R Gui by double-clicking the icon  and install the necessary packages with the command:

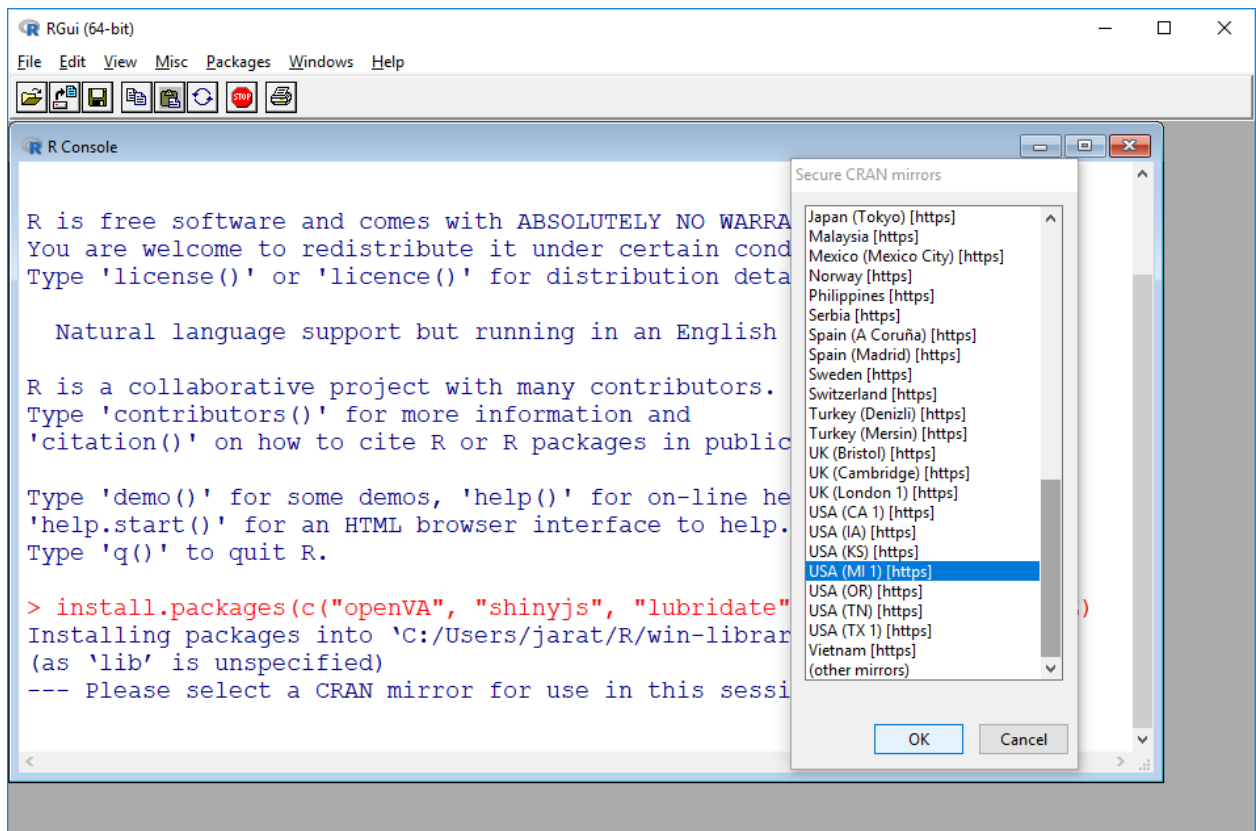
```
install.packages(c("openVA", "shinyjs", "lubridate"), dependencies=TRUE)
```

which should be typed (or copy and pasted) in the R Console window at the prompt as follows:



The `dependencies=TRUE` option will tell R to install many more packages than just the three packages included in the command. For example, the following packages (along with many more) will also be installed: InSilicoVA, InterVA4, nbc4va, Tariff, rJava, and shiny.

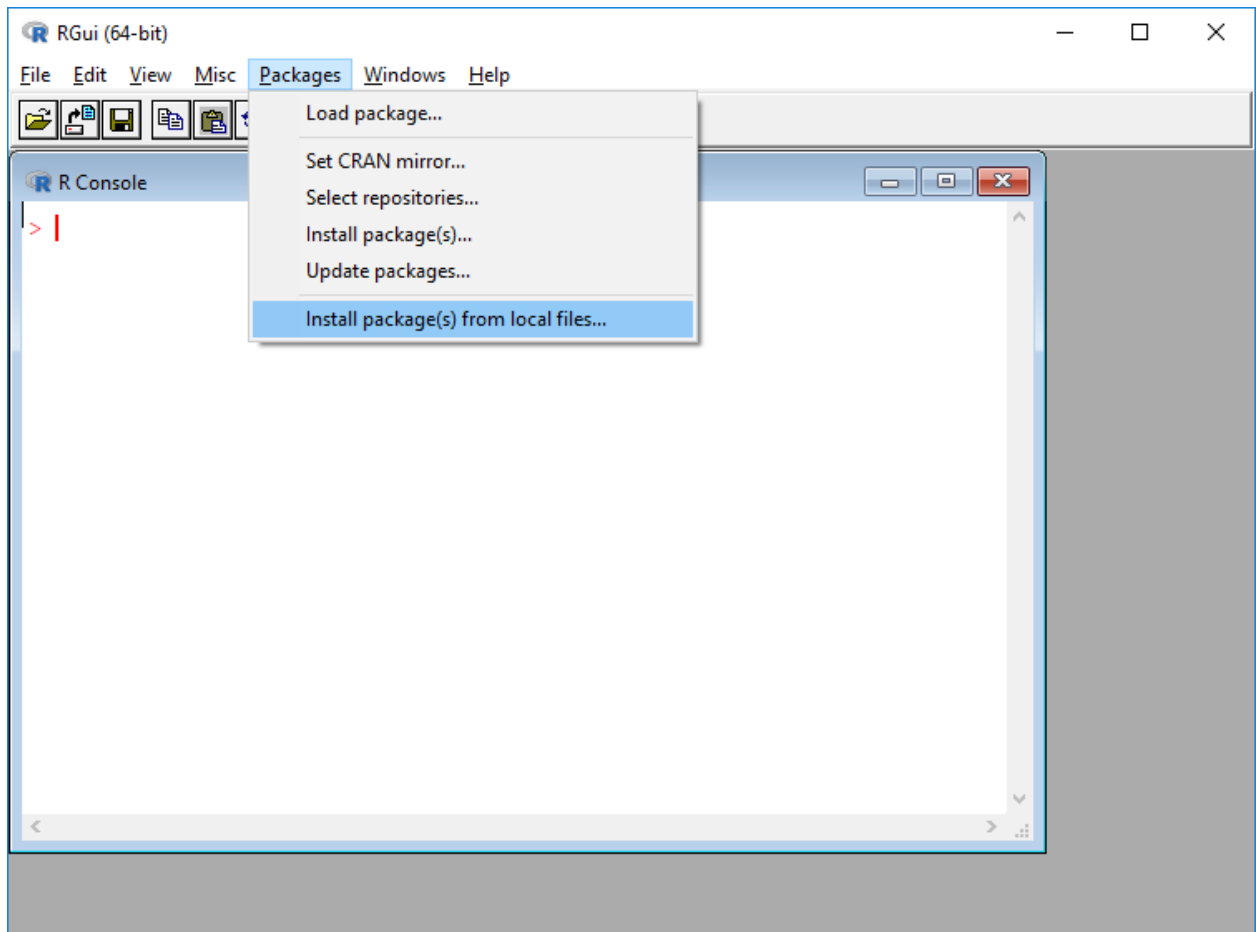
After pressing the enter key in the R console, a new window should appear (as shown in the following figure) asking the user to select a Secure CRAN mirror (choose a CRAN mirror near your location), and the installation will begin after clicking the OK button.



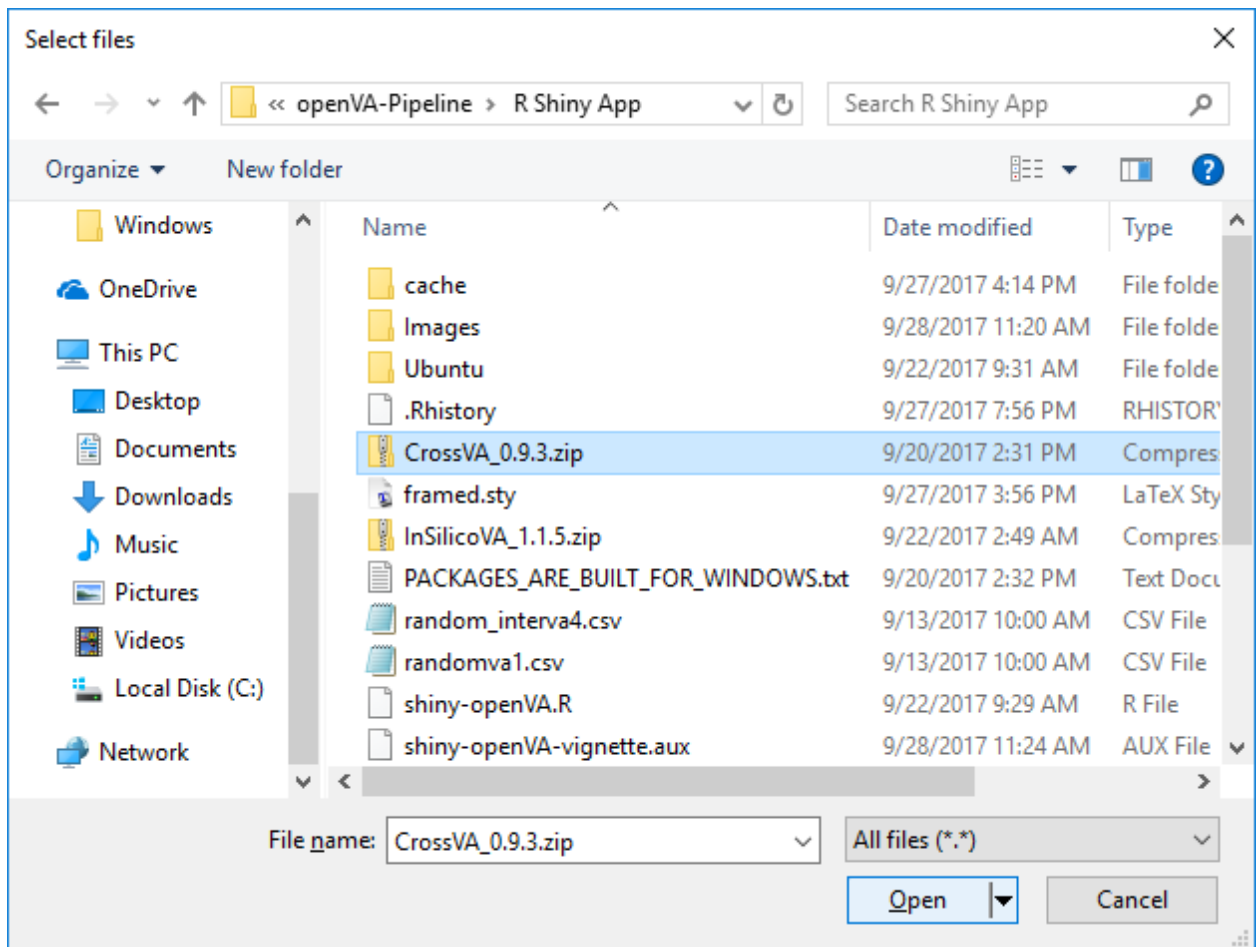
You will see A LOT of text scrolling in the R Console which describes the installation of the numerous packages the Shiny app needs.

A modified version of the CrossVA package also needs to be installed with the following three steps: (1) select the R menu bar options

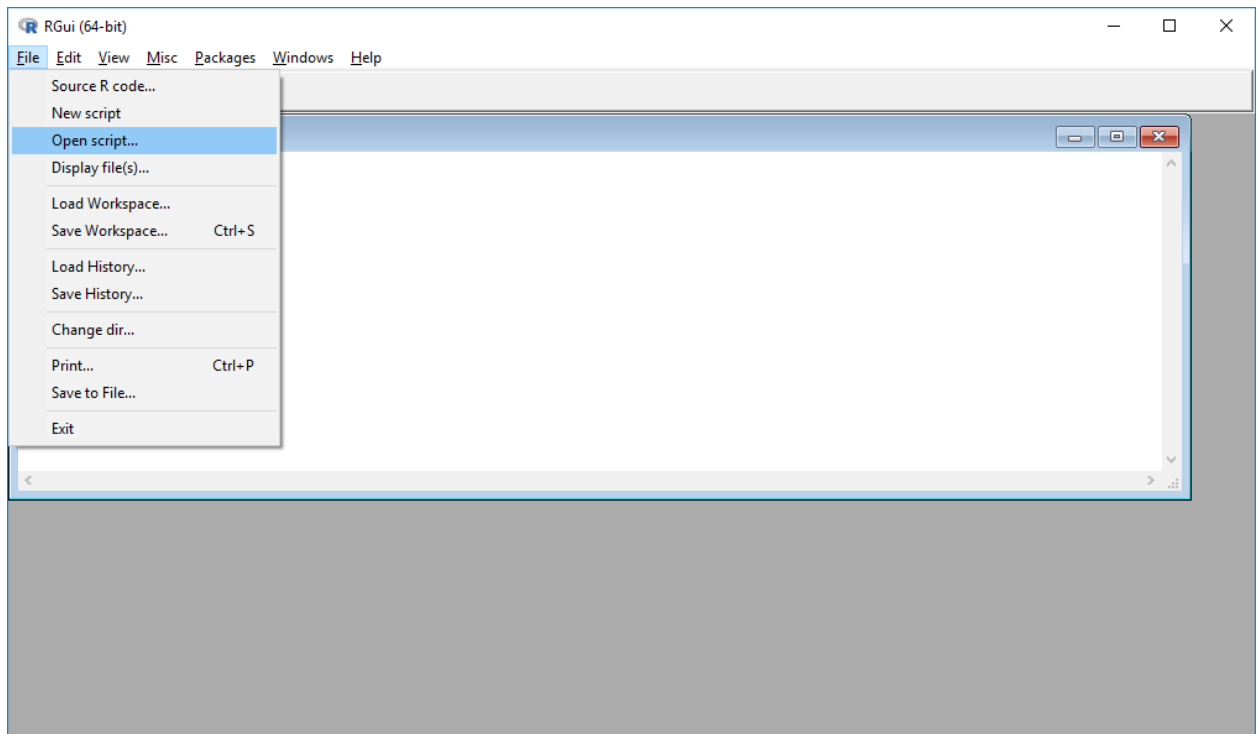
Packages > Install package(s) from local files . . .



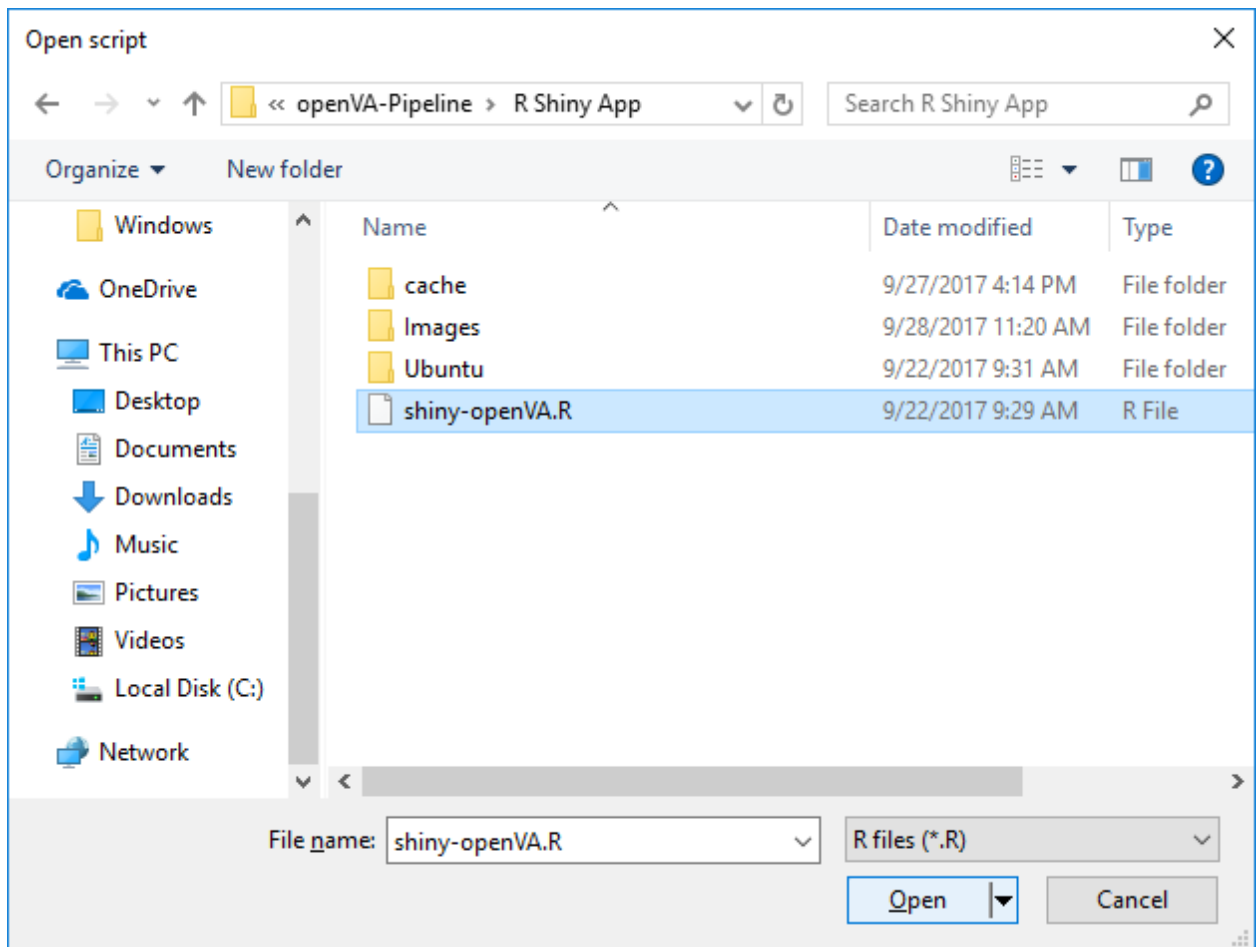
(2) select the CrossVA_0.9.3.zip file (as shown in the following figure; you may need to navigate to the folder where the R Shiny zip file was downloaded); and (3) click the Open button at the bottom-right of the Select files window.



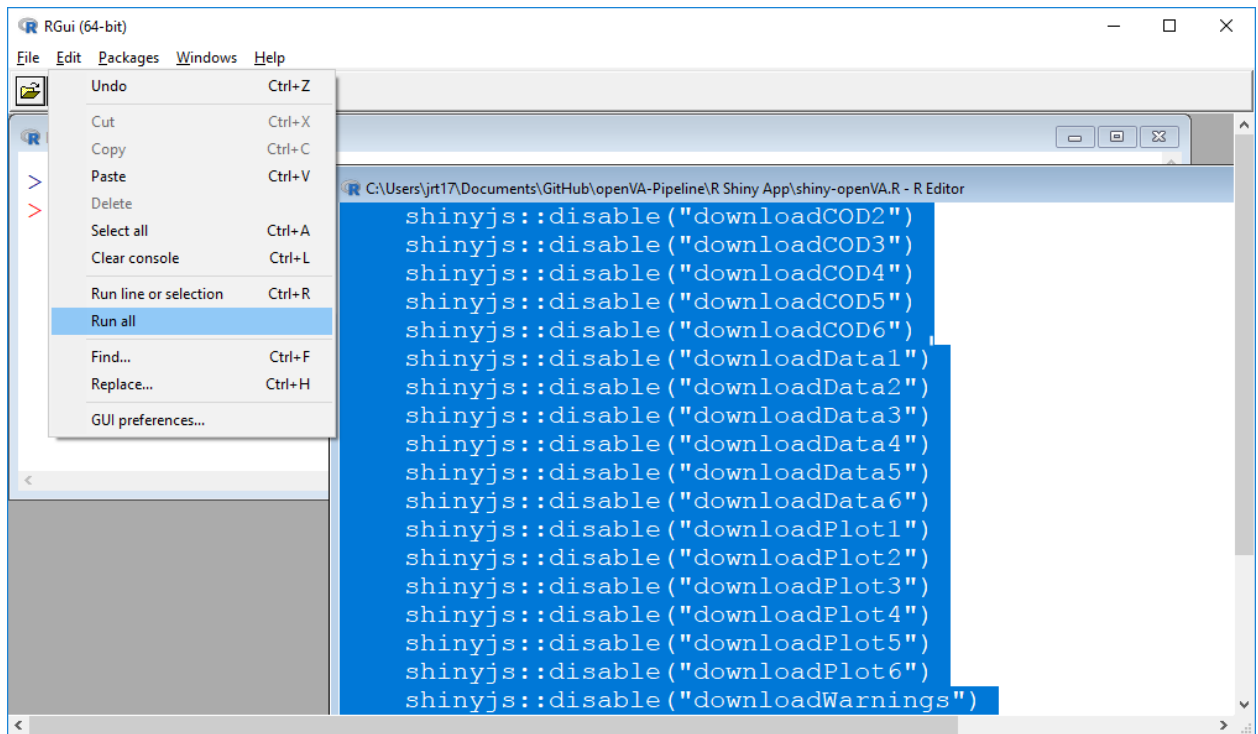
Next open the shiny-openVA.R script (included in the R Shiny zip file) through the R menu bar **File** > **Open Script...**

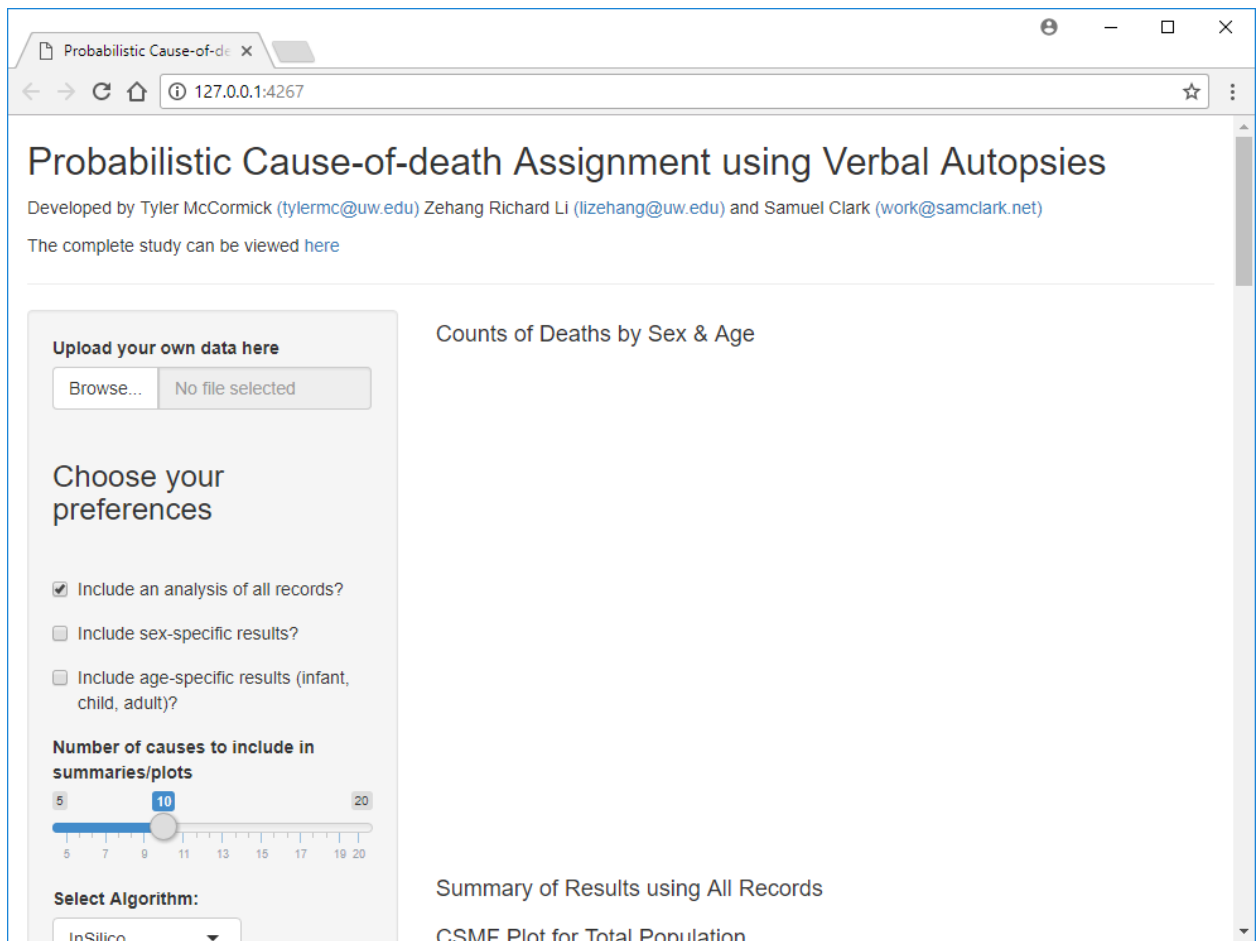


then select the shiny-openVA.R script file and click the Open button at the bottom-right of the Open script window.

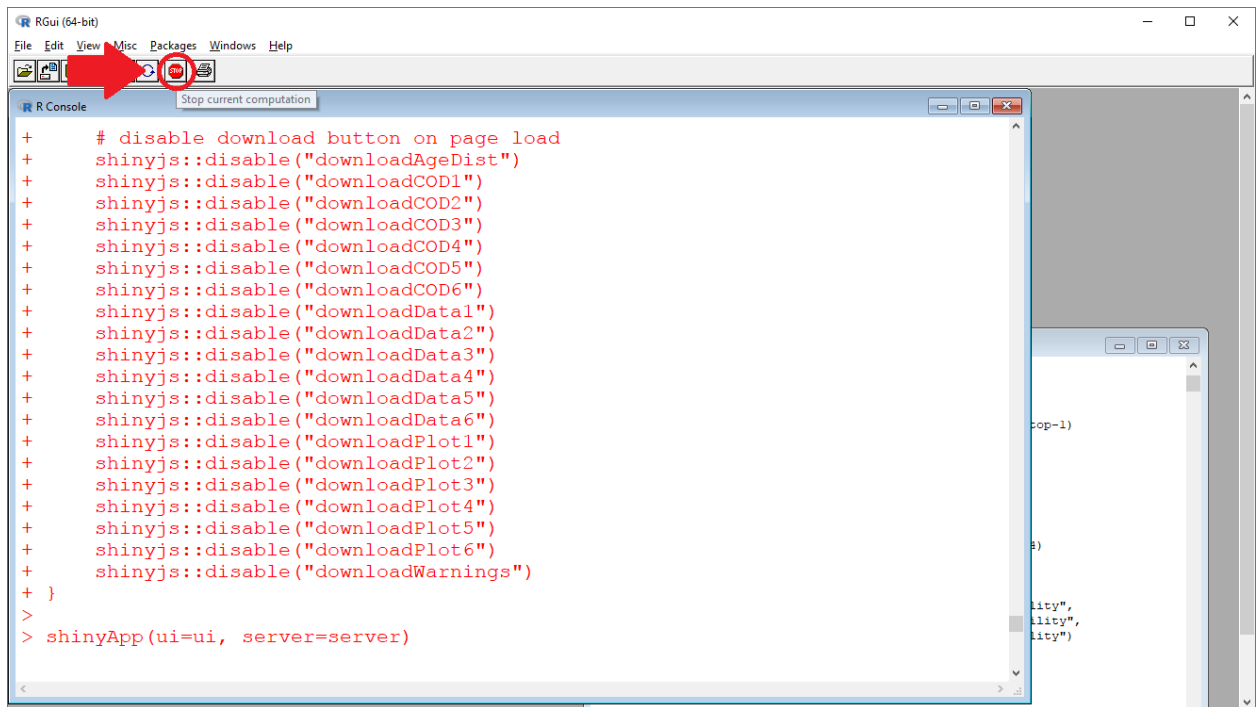


This step should open a new R Editor window. Make sure the focus is on the R Editor window (by clicking in that window) and run the shiny-openVA.R script through the menu bar **Edit > Run all** which will execute the code (in the R Console) and open a web browser with the R Shiny app. It may take a few seconds for the app to appear in the web browser as the code is processed in the R Console window.





Finally, to stop the Shiny app, return to the R Gui and click on the red stop sign button in the tool bar.



3 Using the R Shiny app

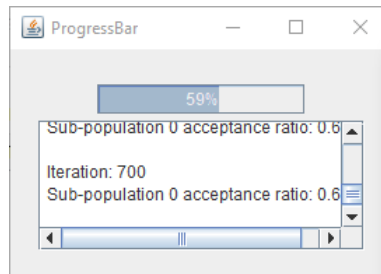
Once the R Shiny app is running in a web browser there are is a panel on the left side of the web page where the data and analysis options are entered.

1. **Upload your own data here** – The first input is a data file in Comma Separated Values (CSV) format (the file should originate from an ODK Briefcase export). After clicking the Browse... button a window will appear allowing you to select your data file. Once the data have been loaded the app produces the message “Upload complete”. There is a sample data file, who_va_output.csv, that is included in the R Shiny zip file and used in the examples below. Note that once a data file has been uploaded a new button will appear **Analyze my data!**.
2. **Choose your preferences** – The data are assumed to be in a particular format with fields identifying the age and gender of the deceased. There are several options for selecting group-specific results. Note that the time it takes to process the data increases with the number of groups selected.
 - (a) **Include an analysis of all records?** – This option is selected by default, but it is not necessary (e.g., you could solely request sex-specific results).
 - (b) **Include sex-specific results?** – Simply click in the empty box to produce separate results for males and females.

- (c) **Include age-specific results?** – Simply click in the empty box to produce separate results for infants (less than 1 year old), children (aged 1-4 years old), and adults (ages 5 and above).
- (d) **Number of causes to include in summaries/plots** – The Shiny app will display the VA results in several tables and plots showing a list of the top (i.e., most prevalent) causes of death. Slide the bar right (or left) to include more (less) causes in the summaries of the results. The default is to include 10 causes, with a minimum and maximum of 5 and 20, respectively – this was simply a design choice for the app, **openVA** can include any number of causes the user wants.

3. Select Algorithm¹

- (a) **InSilico** – The algorithm computes the causes of death (COD) by performing a specific number of iterations, which can be chosen with the slider input (the default is 5,000). Choosing a larger number of iterations will result in a longer waiting time for the algorithm to finish, but will also help improve the approximate results. If you are simply exploring/testing out the app, you may want to choose a small number of iterations (and thus reduce the processing time). Part of the InSilico algorithm is implemented in Java and thus a separate windows will appear with a Java Progress Bar indicating the percentage of iterations that have been completed.



Technical note: if the algorithm does not converge in the specified number of iterations, it will automatically add additional iterations to the simulation. For more details, see <https://cran.r-project.org/package=InSilicoVA>.

- (b) **InterVA4** – The user can specify the level of HIV and malaria prevalence; which are the same parameters required by the InterVA-4 software developed by by Peter Byass (<http://www.interva.net/>). Here is the description from the InterVA-4 User Guide (version 4.04, 2017-06-06): “These settings do not directly determine the cause of death, but conceptually they are similar to a physician knowing that (s)he is working in a high or low malaria or HIV population, irrespective of the details of a specific case currently under consideration.” (p.6)

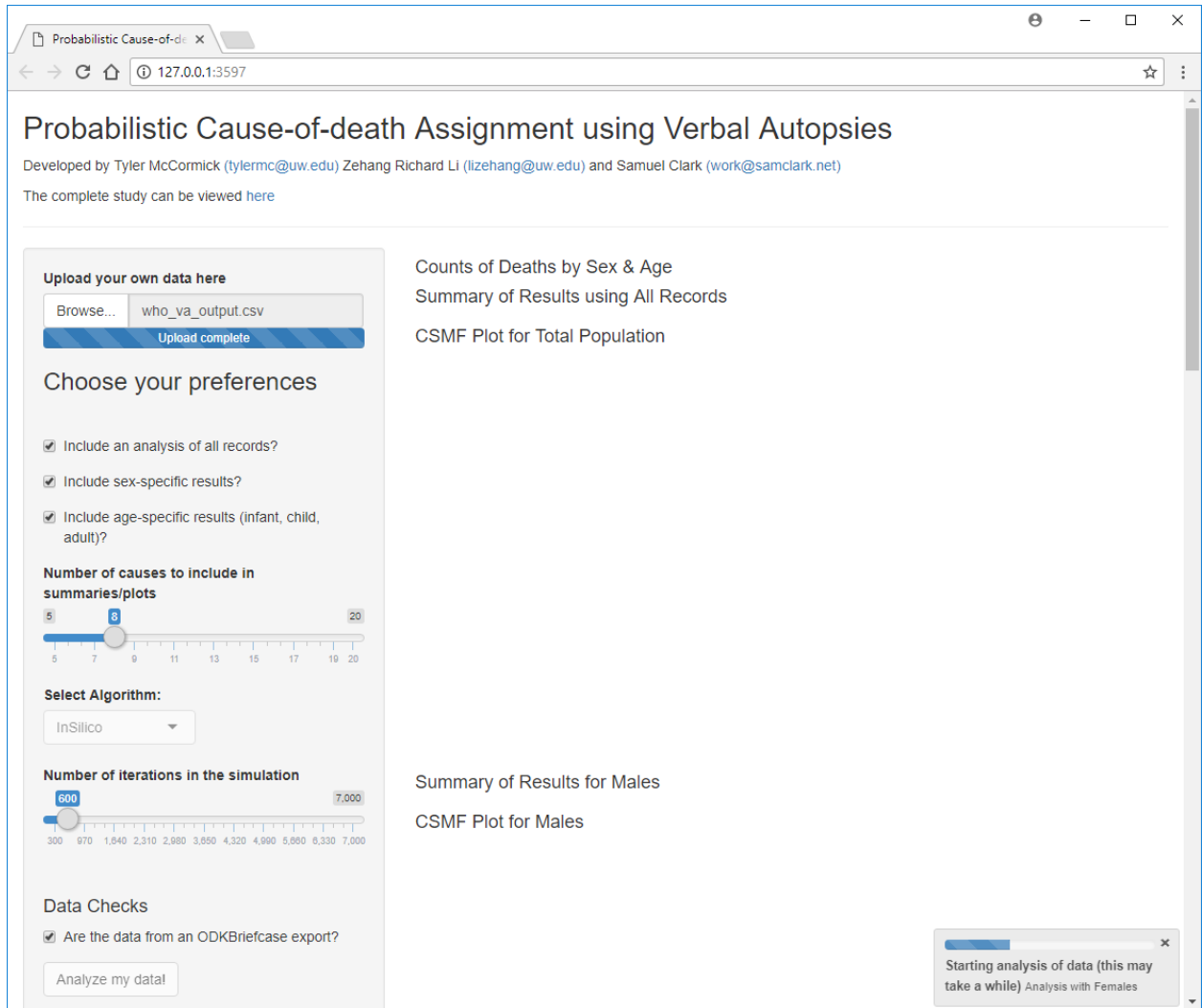
¹Technical Note: **openVA** includes additional algorithms, namely Tariff (<https://cran.r-project.org/package=Tariff>) and the Bayes Classifier for Verbal Autopsy Data (<https://cran.r-project.org/package=nbc4va>), but these options require several additional steps to work with ODK data exports of the WHO VA instruments and thus are not included in the app.

4. **Data Checks:** *Are the data from an ODKBriefcase export?* – This option is checked by default and refers to the CSV file generated by the ODKBriefcase command line interface with arguments to the options “-export_filename” and “-export_directory” (for more details, see <https://opendatakit.org/use/briefcase/>). To analyze data from a different source, see the InSilicoVA documentation for the required data format (<https://cran.r-project.org/package=InSilicoVA>)
5. **Analyze my data!** – Processing the data can take a while with a large amount of data. With 500 cases it may take anywhere from 5 to 30 minutes depending on the algorithm (InSilico typically takes longer, and the waiting time will obviously increase with the number of iterations for the simulation). Once pressed, a progress bar will appear. Again, with InSilico a separate Java ProgressBar window may appear (see above).

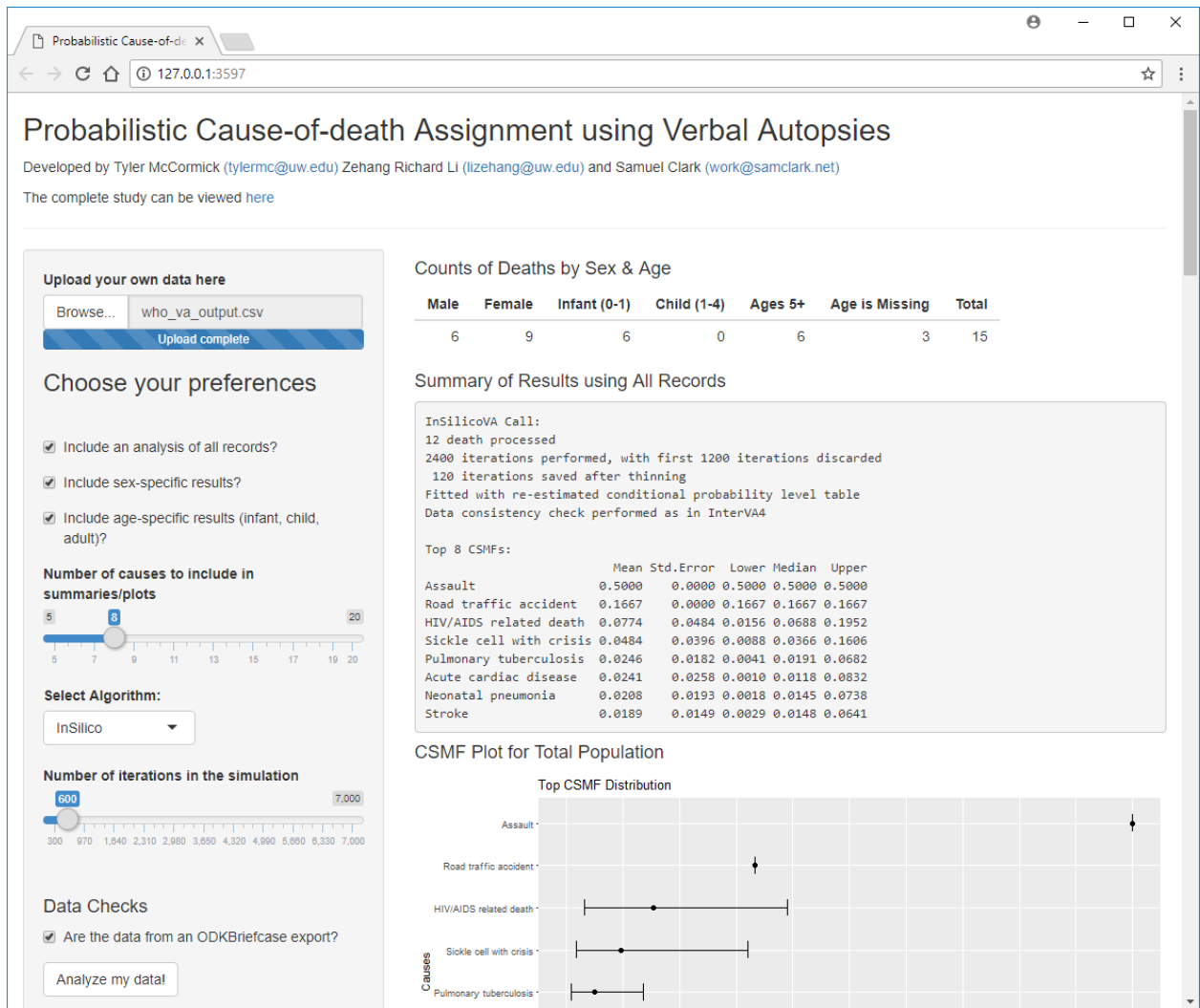
This side panel also includes several additional buttons that are inactive when no data have been analyzed. These buttons allow you to download results in the form of CSV and PDF files, as well as additional text files that document error and warning messages. After providing the necessary data and pressing the **Analyze my data!** the data will be processed and these buttons will become active (i.e., you may click on them) as the results are obtained.

3.1 InSilicoVA Example

In our first example, we analyze the sample data set `who_va_output.csv` (15 records/deaths) generated from an ODKBriefcase export and included in the R Shiny zip file. Results are requested for all records, as well as separately by gender and age groups. The top 8 causes will be included in the summaries of the results (as chosen with the slider input). InSilico is the algorithm being used with the number of iterations set to 600. This took about one minute on a laptop with a 2.7GHz Intel Core i-7 processor.



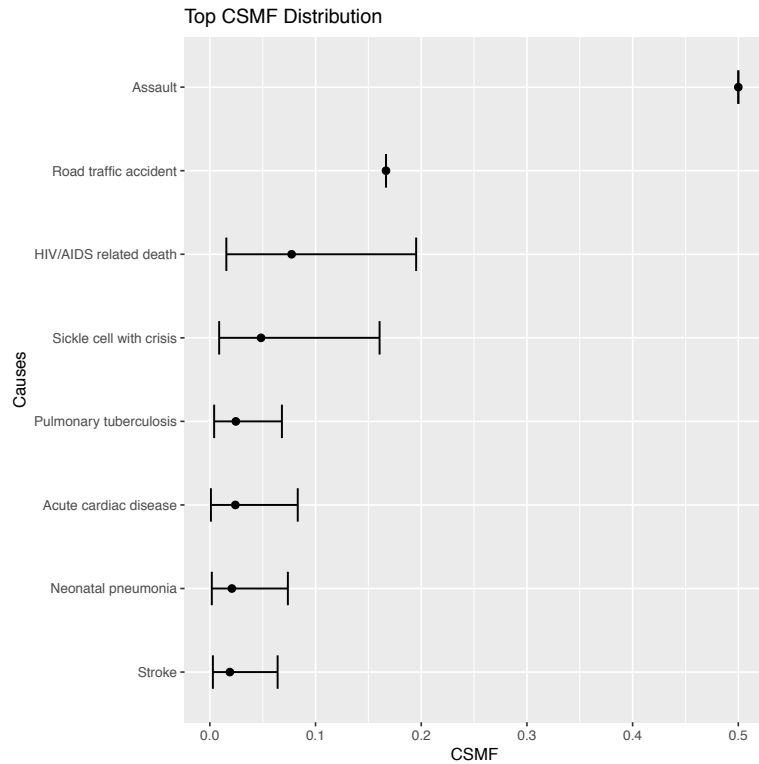
The results are displayed on the right side of the web browser and begin with a table that lists the number of deaths by age and sex that are included in the data. Summaries for the top 8 causes of death and plots of the cause-specific mortality fraction (CSMF) for the top 8 causes are also included for each group selected – results are presented in the following order: total population, males, females, infants, children (results are empty since there are no child deaths in this data set), and adults.



Finally, the results are available for download by clicking on the buttons appearing in the panel on the left side of the browser, with the options described in the following list:

- **Download Plot of Age Distribution as .pdf** – This file contains a histogram of the number of deaths in the three age groups (infants, children, and adults). The corresponding counts are displayed in the table (appearing in the right side of the browser) entitled “Counts of Deaths by Sex & Age”.
- **Download Causes for All Records as .csv** – This file includes the record ID (which is simply the row number in the original data file) and the three most likely causes of death and the associated probabilities; note that some records (5, 9, and 15) are excluded due to data errors (i.e., missing necessary data/indicators).
- **Download Summary for All Records as .csv** – This file contains the information appearing in the web browser under the title “Summary of Results using All Records”.

- **Download Plot for All Records as .pdf** – This file contains the CSMF displayed on the right side of the web browser under the title “CSMF Plot for Total Population” and is reproduced here:

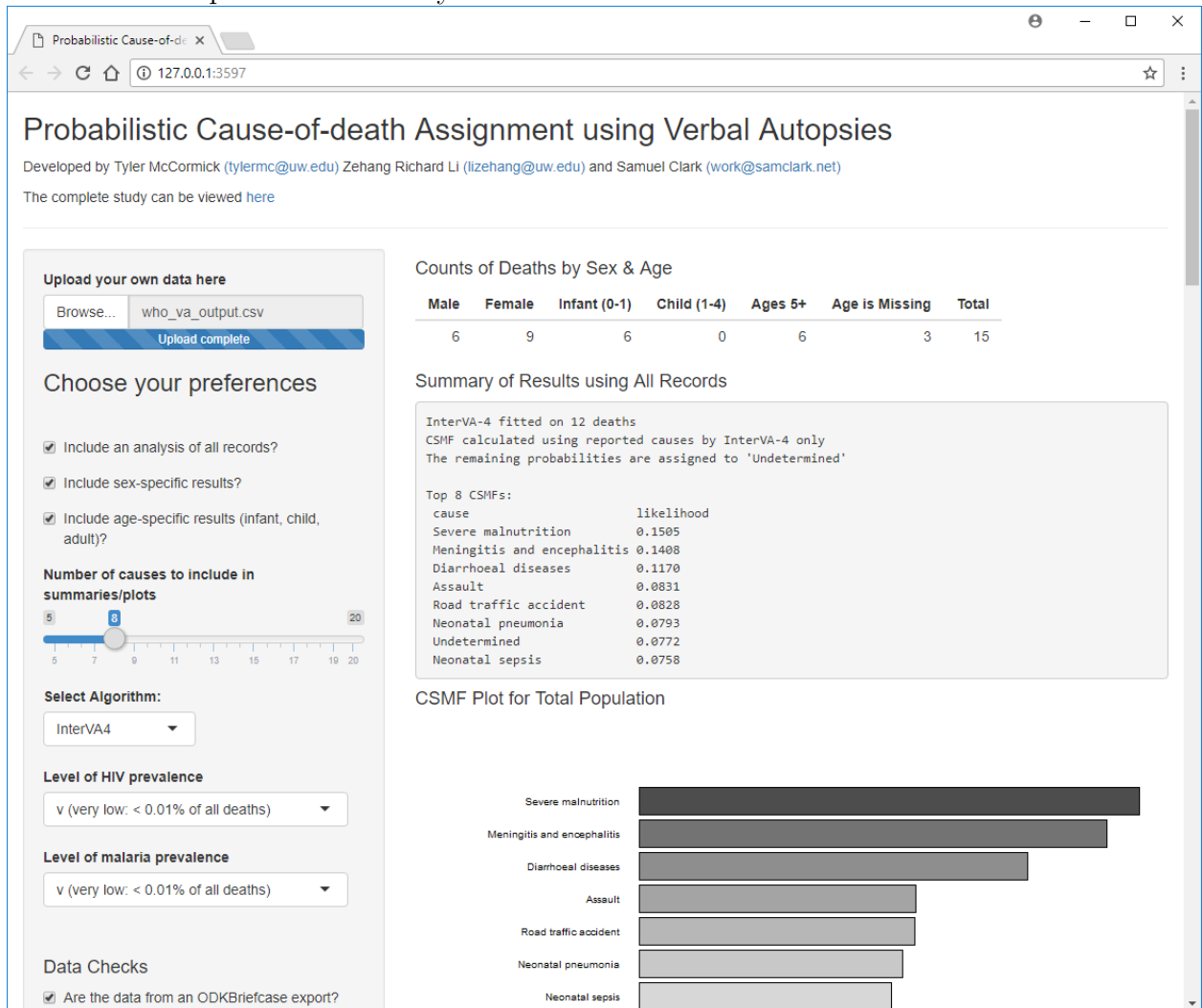


InSilicoVA uses a Bayesian approach which yields results in the form of probability distributions for the parameters of interest. The above figure summarizes these distributions of the CSMF for the top causes with small, solid rectangles – indicating the medians of the probability distributions – and black lines stretching from the 2.5 percentile to the 97.5 percentile (similar to a 95% confidence interval).

- The options for downloading causes and summaries (as a .csv files) and the plots (as a .pdf) are also available for each group included in the analysis. The corresponding buttons also appear in the panel on the left side of the browser, and will become active after the algorithms have calculated the results.
- **Download warnings as .txt** – This option is located at the very bottom of the panel on the left side of the web browser. The associated text file includes the warning and error messages generated by InSilico and the issues encountered and, in some cases, the steps taken to reconcile the inconsistencies in the data.

3.2 InterVA Example

For our second example, we analyze the same data but use the InterVA4 algorithm with the HIV and malaria options set to “very low”.



The results are slightly different because InSilico includes uncertainty (similar to confidence intervals) around the point estimates, whereas InterVA4 does not. However, the number of deaths tabulated by age and sex as well as the top causes of death are still included. The corresponding results are also available for download by clicking on the buttons in the panel on the left side of the browser. An example of the CSMF for all records is included below.

