Forest Service
U.S. DEPARTMENT OF AGRICULTURE

# Quickstart Guide to the vegClass R Package

July 2022

## Table of Contents

## 1.0 Introduction

This user guide provides an overview of the vegClass R package. The vegClass package consists of a suite of R functions that are used to derive vegetation classifications and conditions defined by the US Forest Service (Vandendriesche 2013). Currently, the package only supports classifications defined by Region 3 of the US Forest Service. The outputs that are produced from the functions in this package can be used as input for state and transition models and for other forest planning applications. The vegClass package is implemented through the R statistical programming language and environment (R) and depends on output produced from the Forest Vegetation Simulator (FVS) software. The contents of this guide describe the software you will need to run the vegClass R package, the workflow for producing output with the vegClass package, and documentation on the underlying R functions included in the vegClass package.

## 2.0 Installing Required Software

Three software components are required to use the vegClass R package: FVS software, R, and the vegClass R package itself. Instructions for installing each of these are provided in the sections below. Each software component should be installed in the order that they are presented in this guide.

Note: If you are an employee in the USDA Forest Service, then you will likely need to open a ticket with CHD before installing RStudio Desktop IDE (optional) or R (when assuming Option 2 described below) on your computer.

### 2.1 FVS Software

FVS is a nationally supported growth and yield modelling framework that is maintained by the USDA Forest Service. The following link will take you to the webpage where the FVS software can be downloaded (instructions for installing and uninstalling FVS can be found here as well) https://www.fs.fed.us/fvs/software/index.shtml. Please ensure that you install the latest version of the FVS software or a version that does not predate version 20210930. Users are encouraged to stay up to date with the latest version of the FVS software. Users also have the option to use the online configuration of FVS that is hosted by Virginia Tech University: https://charcoal2.cnre.vt.edu/FVSOnline/.

### 2.2 R

The vegClass R package can currently only be implemented through the R statistical programming language and environment. As such, a version of R needs to be installed on the user's PC. There are two options available for installing a version of R that can be used by the vegClass package.

Option 1 **(Recommended)**
The Forest Vegetation Simulator complete package installation (refer to FVS software section above) includes a version of R that is compatible with the vegClass package. This version of R is

recommended since it comes included with several R packages that the vegClass package depends on. No further action is required if the FVS complete package is already installed on PC.

Option 2
Install a version of R through the r-project: https://www.r-project.org/. Instructions for downloading R can be found in the getting started section of the r-project webpage. Please ensure that a version of R greater than or equal to 4.1.1 is downloaded and installed.

## 2.3 vegClassInstall Folder

The vegClassInstall folder can be downloaded through the following link: TO BE DETERMINED. Once the files have been downloaded, unzip the folder to a desired location on your PC. the vegClassInstall folder contains the following items which will be described in more depth further along in this guide:

**Data_Create.R:** R script that illustrates how to combine the FIA2FVS datasets Arizona and New Mexico and create grouping codes corresponding to ERUs.

**FVSOut.db:** Example FVS output database that is used to illustrate the use of the vegClass package in vegClass_Example.R.

**Install.R:** R script that can be used to install the vegClass R package (vegClass_0.1.0.tar.gz). This guide does not describe how to use this script but provides directions for installing the vegClass package from the RGUI and RStudio Desktop interfaces.

**vegClass_0.1.0.tar.gz:** Source R package that will be installed in Install.R.

**vegClass_Example.R:** R Script that presents an example of how to attach vegClass R package and call main function to derive vegetation classification output. The Installation of the vegClass R package will be described in section 3.0 of the guide.

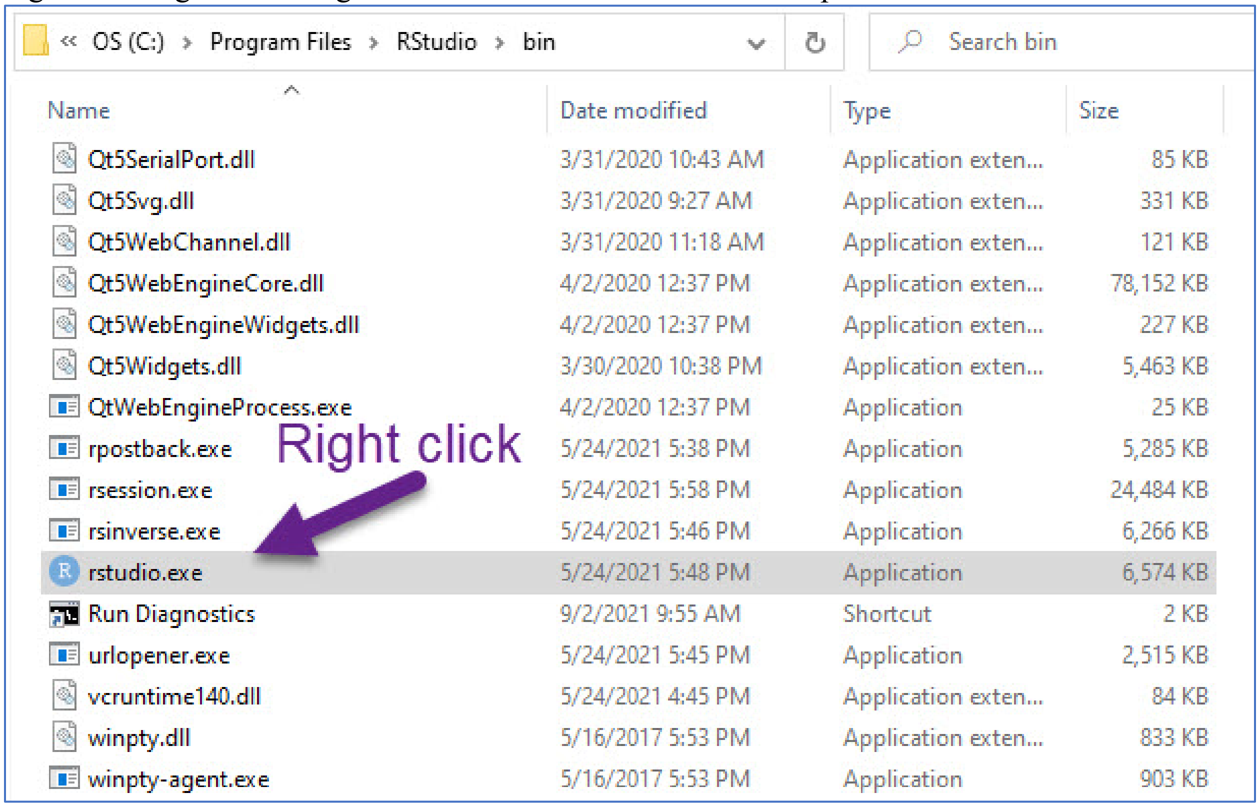
## 2.4 RStudio Desktop IDE (Optional)

Although you can run vegClass directly through the RGui which comes included with the installation of R, it is recommended that you install the RStudio Desktop IDE as well. The RStudio Desktop IDE provides a more intuitive and flexible environment for interacting with R when compared to the default RGui. The following link will take you to the webpage for installing the RStudio IDE: https://www.rstudio.com/products/rstudio/download/.

Note: A desktop icon is not automatically created for RStudio once the application is installed. To easily access RStudio from your desktop, you will need to create a shortcut for the application or pin it to the Windows taskbar. For either option, you will need to navigate to the directory where RStudio was installed to. By default, RStudio is installed to the following directory: C:\Program Files\RStudio. In the RStudio folder, navigate to the Bin folder. To pin RStudio to the taskbar, right click on the rstudio.exe file and select **Pin to taskbar** option (Figure

1). Alternatively, if you would like to create a shortcut, you can select the **Create shortcut** option.

Figure 1. Image illustrating how to access rstudio.exe in File Explorer.



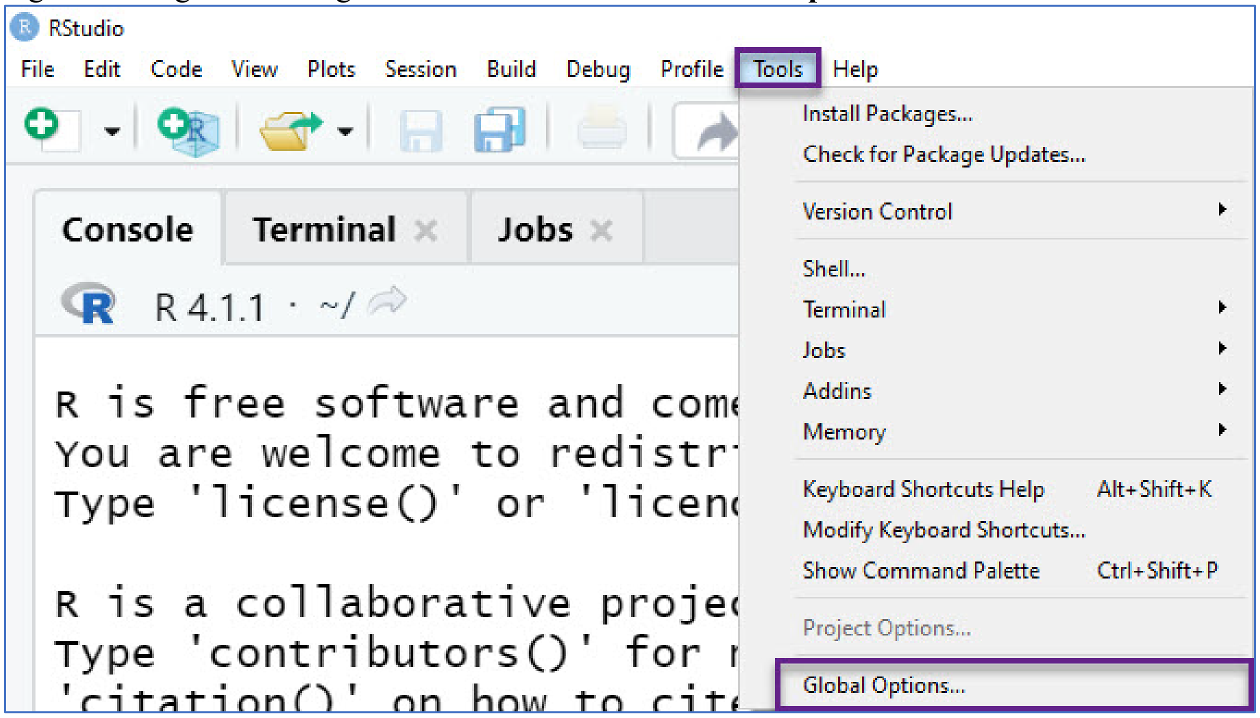
| Name | Date modified | Type | Size |
|---|---|---|---|
| Qt5SerialPort.dll | 3/31/2020 10:43 AM | Application exten... | 85 KB |
| Qt5Svg.dll | 3/31/2020 9:27 AM | Application exten... | 331 KB |
| Qt5WebChannel.dll | 3/31/2020 11:18 AM | Application exten... | 121 KB |
| Qt5WebEngineCore.dll | 4/2/2020 12:37 PM | Application exten... | 78,152 KB |
| Qt5WebEngineWidgets.dll | 4/2/2020 12:37 PM | Application exten... | 227 KB |
| Qt5Widgets.dll | 3/30/2020 10:38 PM | Application exten... | 5,463 KB |
| QtWebEngineProcess.exe | 4/2/2020 12:37 PM | Application | 25 KB |
| rpostback.exe | 5/24/2021 5:38 PM | Application | 5,285 KB |
| rsession.exe | 5/24/2021 5:58 PM | Application | 24,484 KB |
| rsinverse.exe | 5/24/2021 5:46 PM | Application | 6,266 KB |
| rstudio.exe | 5/24/2021 5:48 PM | Application | 6,574 KB |
| Run Diagnostics | 9/2/2021 9:55 AM | Shortcut | 2 KB |
| urlopener.exe | 5/24/2021 5:45 PM | Application | 2,515 KB |
| vcruntime140.dll | 5/24/2021 4:45 PM | Application exten... | 84 KB |
| winpty.dll | 5/16/2017 5:53 PM | Application exten... | 833 KB |
| winpty-agent.exe | 5/16/2017 5:53 PM | Application | 903 KB |

### 2.4.1 Linking RStudio Desktop IDE to R

If you are planning to use the vegClass package in RStudio, it is necessary to ensure that it is linked to the correct version of R. The following steps detail how to link RStudio to your desired version of R.

1. Navigate to the **Tools** menu in RStudio and select **Global Options…** from the dropdown list (Figure 2).

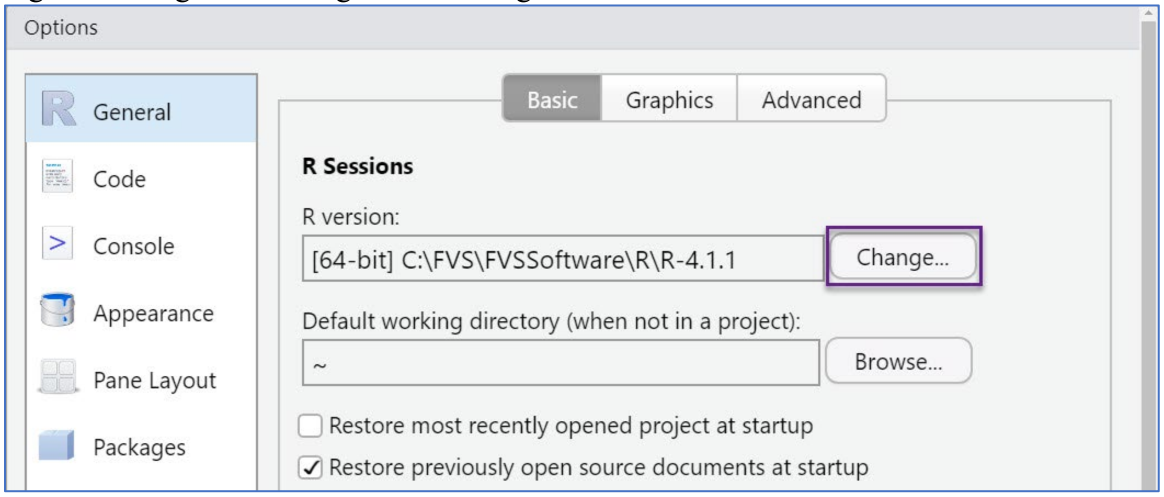
Figure 2. Image illustrating how to access **Tools** and **Global Options…** in RStudio.



```
R is free software and come
You are welcome to redistr
Type 'license()' or 'licen

R is a collaborative projec
Type 'contributors()' for m
'citation()' on how to cite
```

2. Under R session, click **Change…** next to R version (Figure 3).

Figure 3. Image illustrating how to change version of R that RStudio uses.



**R Sessions**

R version:

[64-bit] C:\FVS\FVSSoftware\R\R-4.1.1

Default working directory (when not in a project):

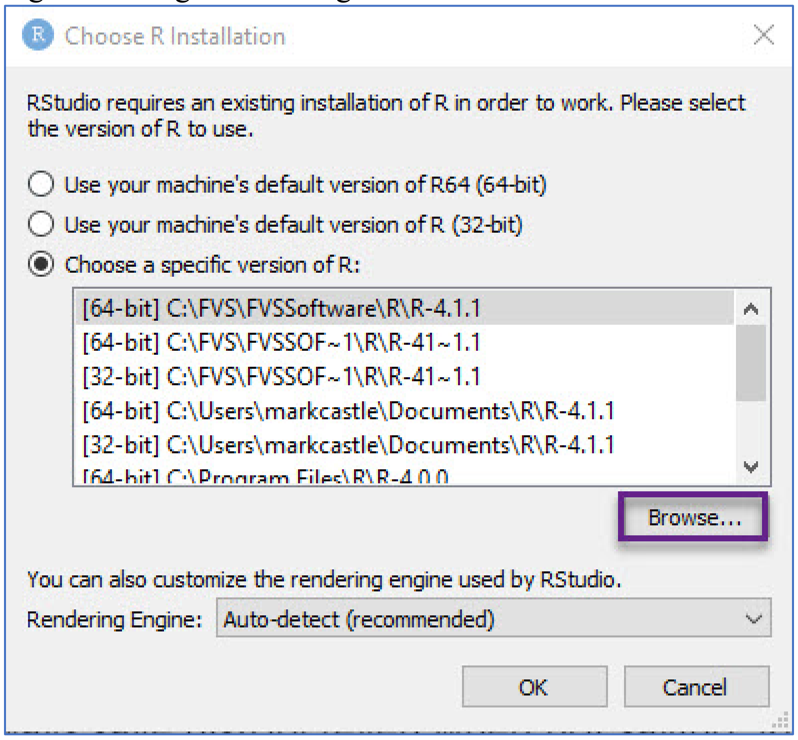
~

☐ Restore most recently opened project at startup

☑ Restore previously open source documents at startup

3. Browse for the desired version of R. To do so, click the **Browse…** button under the *Change a specific version of R* field (Figure 4). **Note: If the desired version of R is already active in the *R version* field, then skip the remaining steps in section 2.3.1.**

Figure 4. Image illustrating how to browse for version of R on PC in RStudio.


Choose R Installation

RStudio requires an existing installation of R in order to work. Please select the version of R to use.

( ) Use your machine's default version of R64 (64-bit)
( ) Use your machine's default version of R (32-bit)
[✓ Choose a specific version of R:]

[64-bit] C:\FVS\FVSSoftware\R\R-4.1.1
[64-bit] C:\FVS\FVSSOF~1\R\R-41~1.1
[32-bit] C:\FVS\FVSSOF~1\R\R-41~1.1
[64-bit] C:\Users\markcastle\Documents\R\R-4.1.1
[32-bit] C:\Users\markcastle\Documents\R\R-4.1.1
[64-bit] C:\Program Files\R\R-4.0.0

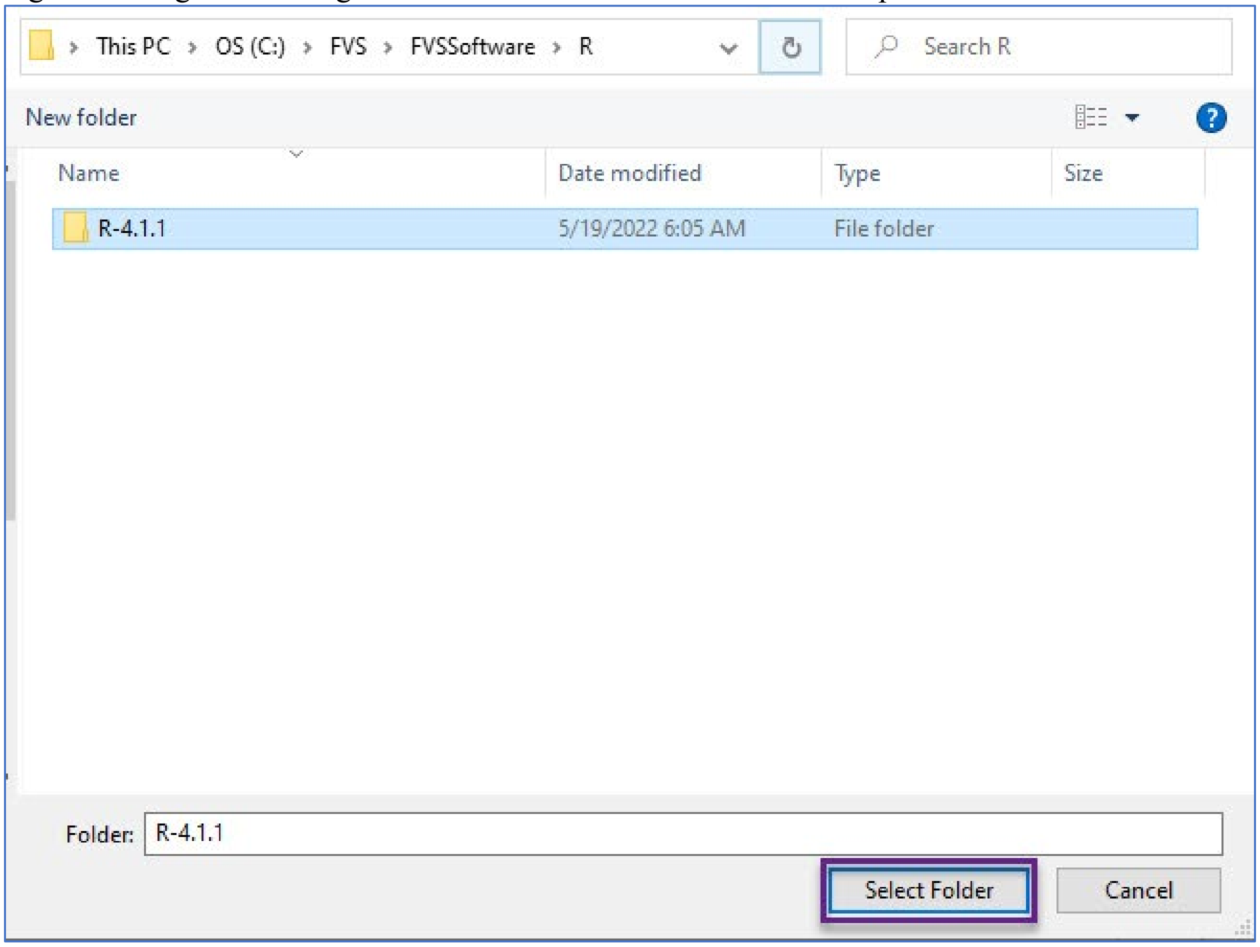
You can also customize the rendering engine used by RStudio.

Rendering Engine: Auto-detect (recommended)

Note: It is recommended that you use the version of R that is distributed with the Forest Vegetation Simulator (FVS) software. By default, this version of R can be found in the following directory on your PC: C:\FVS\FVSSoftware\R\R-4.1.1. If you installed FVS to location other than the C drive, then browse for the version of R in the installation directory.

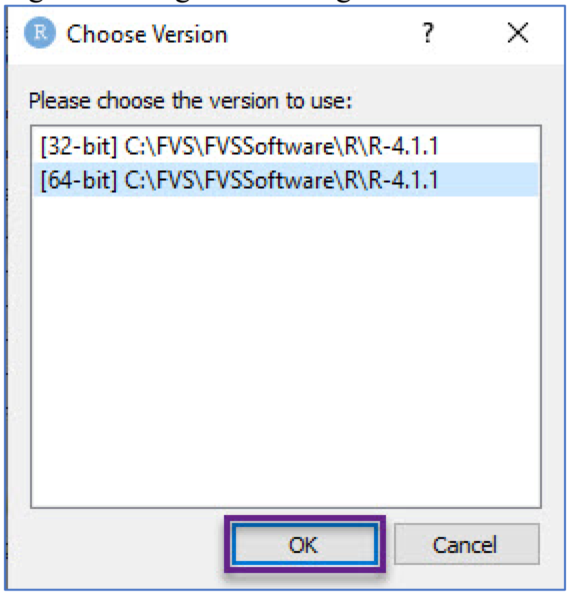
4. Once you have found the desired version of R, click the Select Folder button in file explorer (Figure 5).

Figure 5. Image illustrating how to select R version folder in File Explorer.



5. You will then be prompted to select either a 32- or 64-bit (Recommended) version of R. Once the version of R is selected, press the **OK** button in the *Choose Version* window (Figure 6).

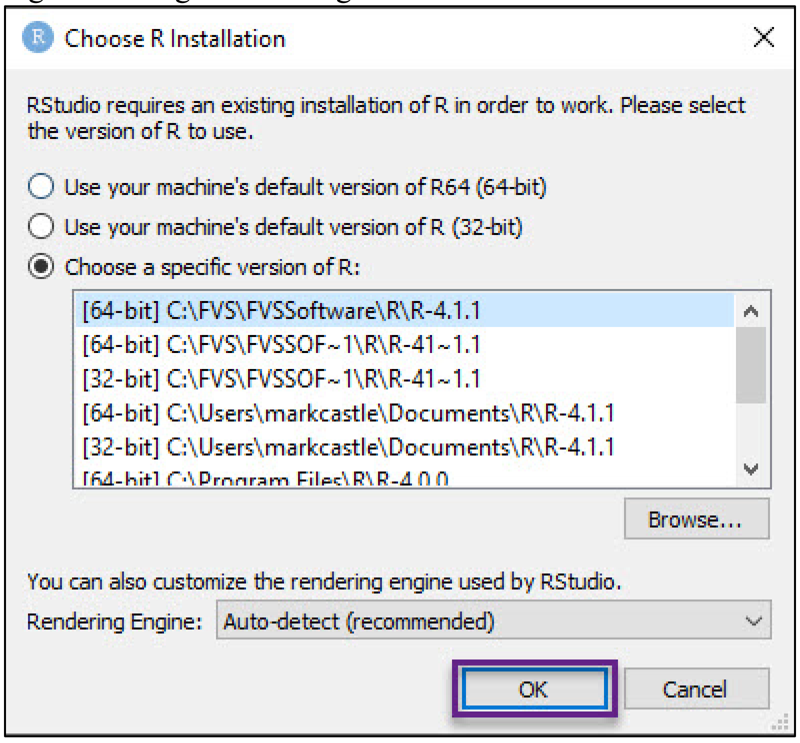
Figure 6. Image illustrating how to select build of R to use in *Choose Version* window.



[32-bit] C:\FVS\FVSSoftware\R\R-4.1.1
[64-bit] C:\FVS\FVSSoftware\R\R-4.1.1

6. Press the **OK** button in the *Choose R Installation* window. You will then be prompted to close and reopen RStudio, so the change takes effect (Figure 7).

Figure 7. Image illustrating how to select version of R to use in *Choose R Installation* window.



RStudio requires an existing installation of R in order to work. Please select the version of R to use.

- Use your machine's default version of R64 (64-bit)
- Use your machine's default version of R (32-bit)
- [✓ Choose a specific version of R:]

[64-bit] C:\FVS\FVSSoftware\R\R-4.1.1
[64-bit] C:\FVS\FVSSOF~1\R\R-41~1.1
[32-bit] C:\FVS\FVSSOF~1\R\R-41~1.1
[64-bit] C:\Users\markcastle\Documents\R\R-4.1.1
[32-bit] C:\Users\markcastle\Documents\R\R-4.1.1
[64-bit] C:\Program Files\R\R-4.0.0

You can also customize the rendering engine used by RStudio.

Rendering Engine: Auto-detect (recommended)

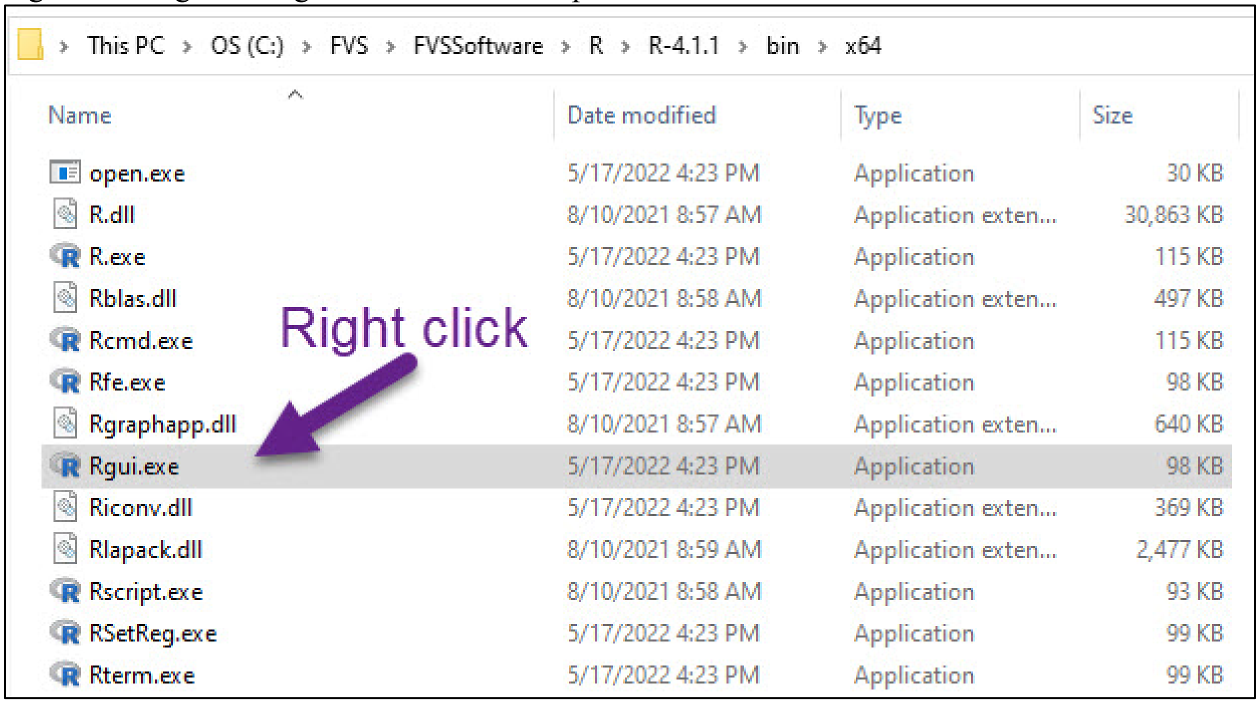
## 3.0 Installing vegClass R Package

This guide provides instructions for installing the vegClass R package in two environments: RGUI (Section 4.1) and RStudio Desktop IDE (Section 4.2). The decision for what instructions to follow depend on the user's preference for the environment that the vegClass package will be implemented in.

### 3.1 RGUI

The RGUI interface comes with the version of R that is included with the Forest Vegetation Simulator software. By default, the RGUI installation can be found in the C:\FVS\FVSSoftware\R\R-4.1.1\bin directory. It is recommended to use the 64-bit version of RGUI which can be found in C:\FVS\FVSSoftware\R\R-4.1.1\bin\x64. For ease of use, you can pin RGUI to the taskbar or create a shortcut. To pin RGUI to the taskbar, right click on the Rgui.exe file and select **Pin to taskbar** option (8). Alternatively, if you would like to create a shortcut, you can select the **Create shortcut** option (Figure 8).

Figure 8. Imagine of Rgui.exe file in File Explorer.


This PC › OS (C:) › FVS › FVSSoftware › R › R-4.1.1 › bin › x64
Right click

| Name | Date modified | Type | Size |
|---|---|---|---|
| open.exe | 5/17/2022 4:23 PM | Application | 30 KB |
| R.dll | 8/10/2021 8:57 AM | Application exten... | 30,863 KB |
| R.exe | 5/17/2022 4:23 PM | Application | 115 KB |
| Rblas.dll | 8/10/2021 8:58 AM | Application exten... | 497 KB |
| Rcmd.exe | 5/17/2022 4:23 PM | Application | 115 KB |
| Rfe.exe | 5/17/2022 4:23 PM | Application | 98 KB |
| Rgraphapp.dll | 8/10/2021 8:57 AM | Application exten... | 640 KB |
| Rgui.exe | 5/17/2022 4:23 PM | Application | 98 KB |
| Riconv.dll | 5/17/2022 4:23 PM | Application exten... | 369 KB |
| Rlapack.dll | 8/10/2021 8:59 AM | Application exten... | 2,477 KB |
| Rscript.exe | 8/10/2021 8:58 AM | Application | 93 KB |
| RSetReg.exe | 5/17/2022 4:23 PM | Application | 99 KB |
| Rterm.exe | 5/17/2022 4:23 PM | Application | 99 KB |

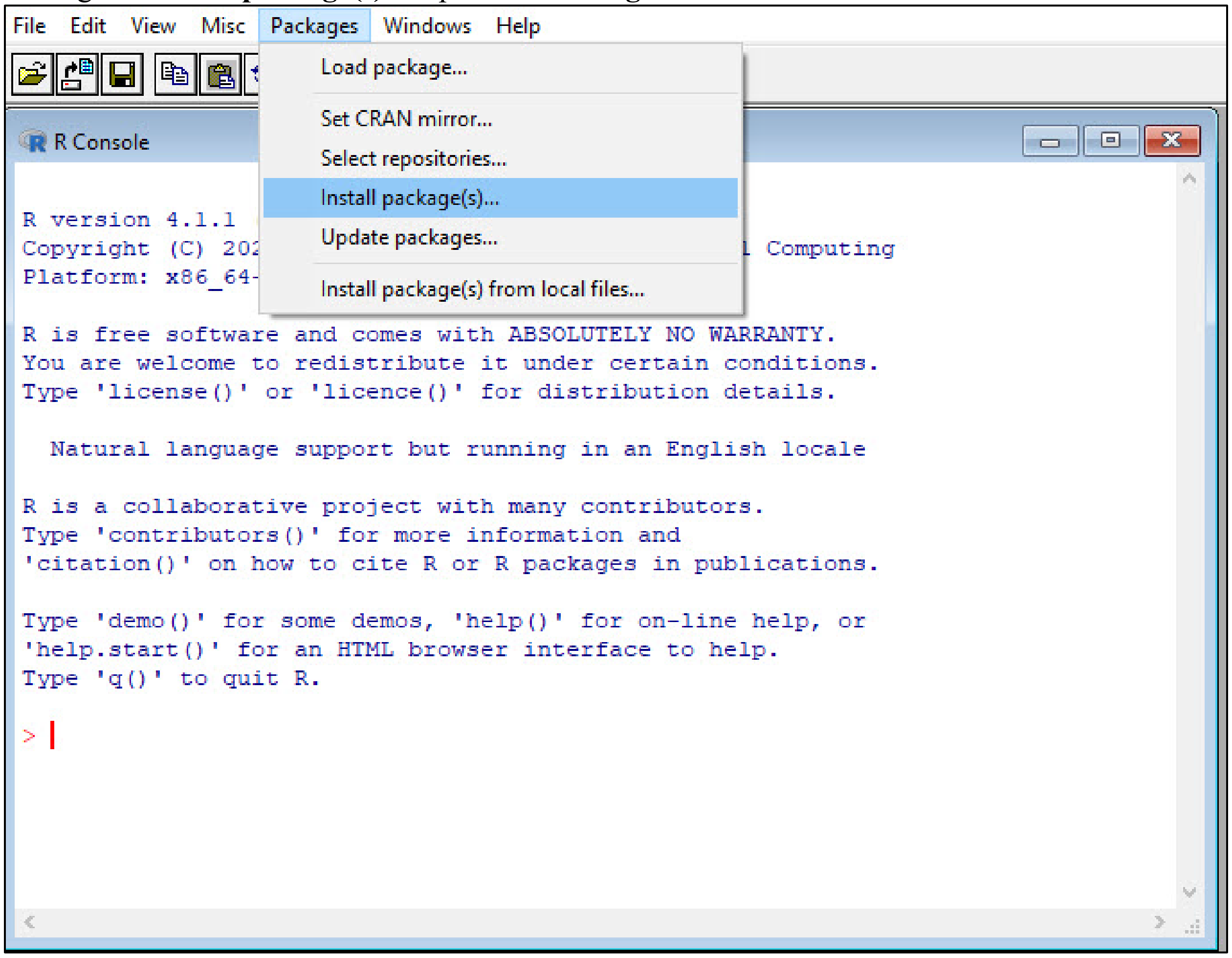
The follow steps describe how to install the vegClass R package in the RGUI interface.

1. Open the RGUI interface.

2. If you are NOT using the version of R that is distributed with FVS and you don't have the RSQLite package installed, then you will first need to install these packages now. Steps 3 – 5 will outline how to do this. If you already have these packages, skip to step 6

3. To install these packages, go to the **Packages** dropdown list and select the **Install package(s)…** option (Figure 9).

Figure 9. Image of **Install package(s)…** option in **Packages** menu of RGUI.



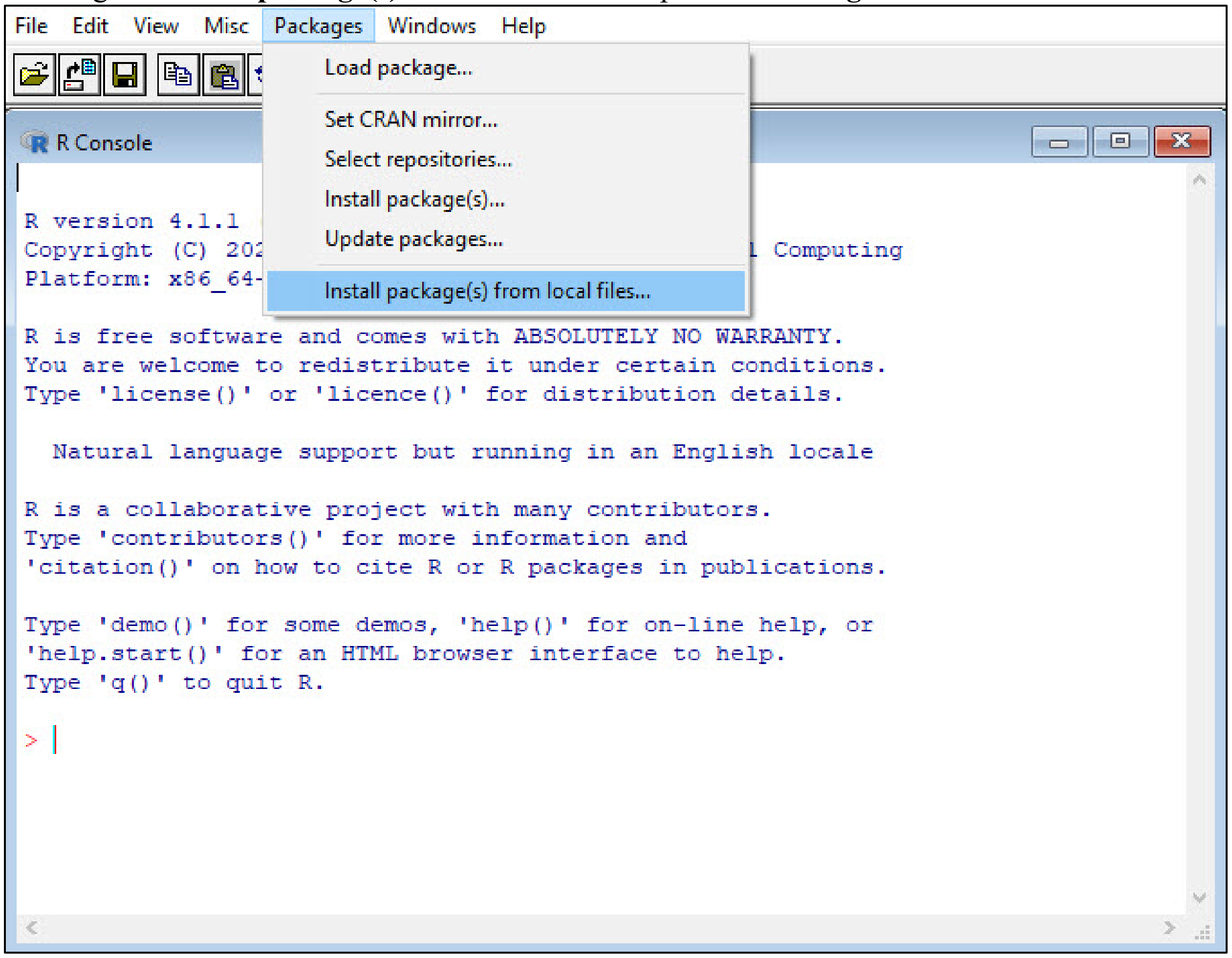
```
R version 4.1.1
Copyright (C) 202                                   l Computing
Platform: x86_64-

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.

>
```

4. You will be prompted to select a CRAN mirror where these packages will be download from. It is recommended that you select a CRAN mirror that is hosted in the US. Once you have selected a CRAN mirror, click the **OK** button.

5. From the *Packages* window, select the RSQLite package and then click the **OK** button. You can select multiple packages by pressing **Ctrl + mouse click**.

6. Now the vegClass package will need to be installed. You can install this by navigating to the Packages dropdown list and selecting the **Install package(s) from local file** option (Figure 10).

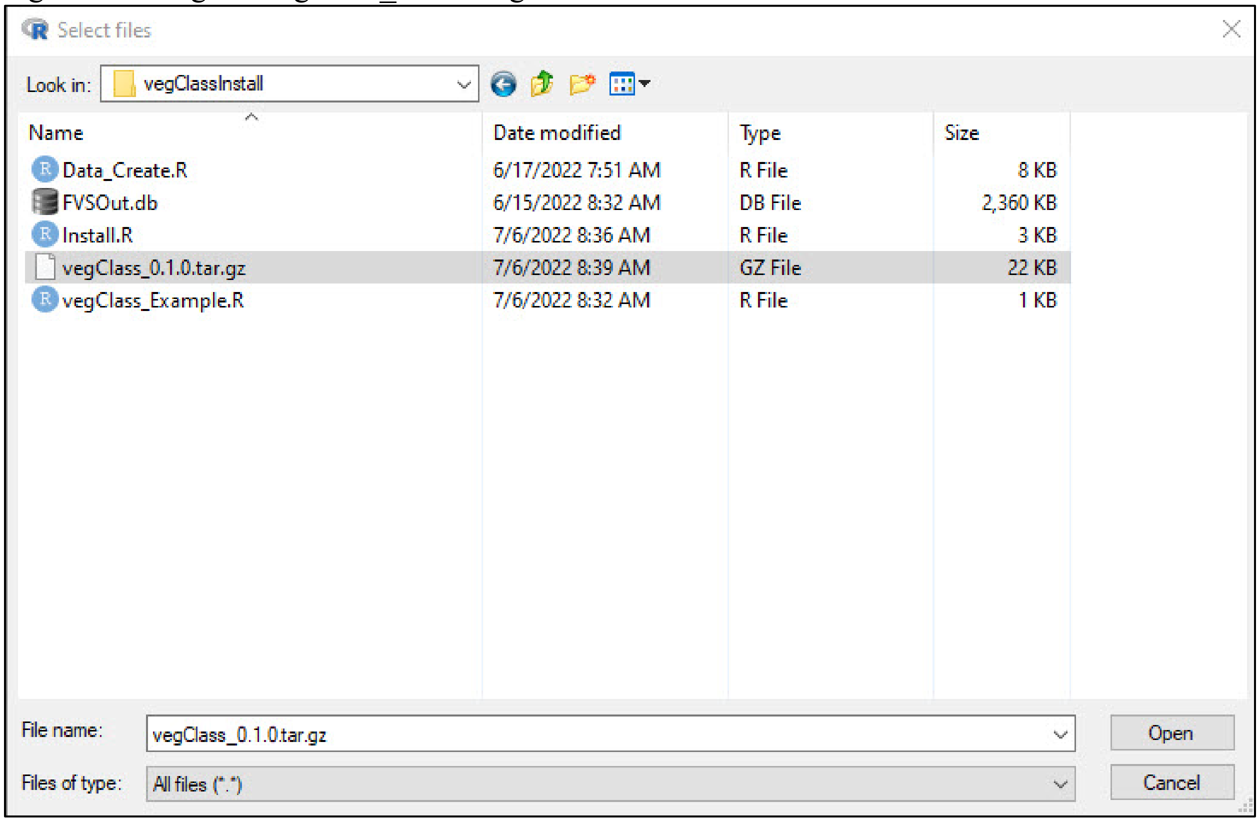
Figure 10. Image of **Install package(s) from local files…** option in **Packages** menu of RGUI.



```
R version 4.1.1
Copyright (C) 202                                  l Computing
Platform: x86_64-

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.

> 
```

7. You will be prompted to navigate to the directory where vegClass_0.1.0.tar.gz is located. Navigate to this location, select the vegClass_0.1.0.tar.gz file, and then click the **Open** button (Figure 11).

Figure 11. Image of vegClass_0.1.0.tar.gz file in *Select Files* window.



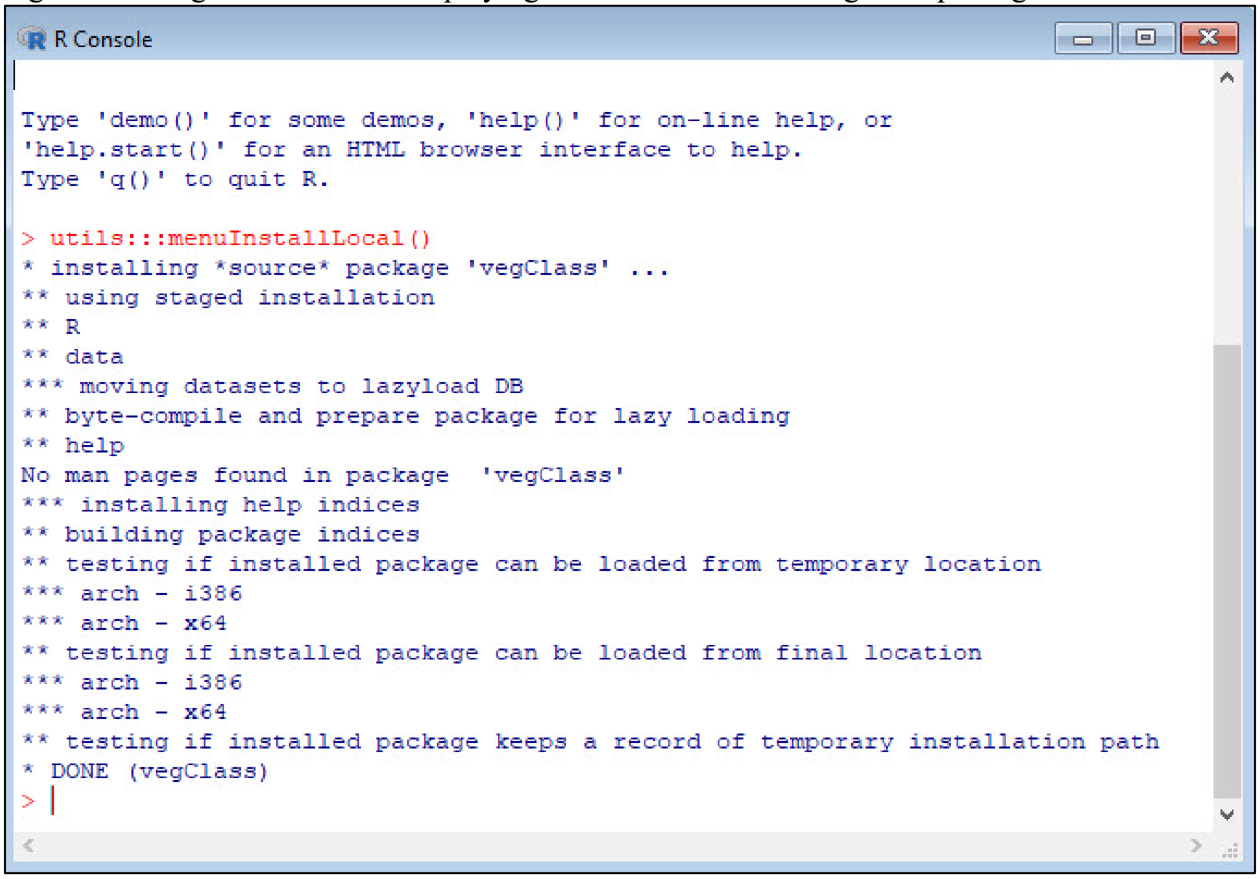
| Name | Date modified | Type | Size |
|---|---|---|---|
| Data_Create.R | 6/17/2022 7:51 AM | R File | 8 KB |
| FVSOut.db | 6/15/2022 8:32 AM | DB File | 2,360 KB |
| Install.R | 7/6/2022 8:36 AM | R File | 3 KB |
| vegClass_0.1.0.tar.gz | 7/6/2022 8:39 AM | GZ File | 22 KB |
| vegClass_Example.R | 7/6/2022 8:32 AM | R File | 1 KB |

File name: vegClass_0.1.0.tar.gz

Files of type: All files (*.*)

The information in the following screenshot should appear in the R console if the installation of the vegClass package is successful (Figure 12).

Figure 12. Image of R Console displaying installation status of vegClass package in RGUI.



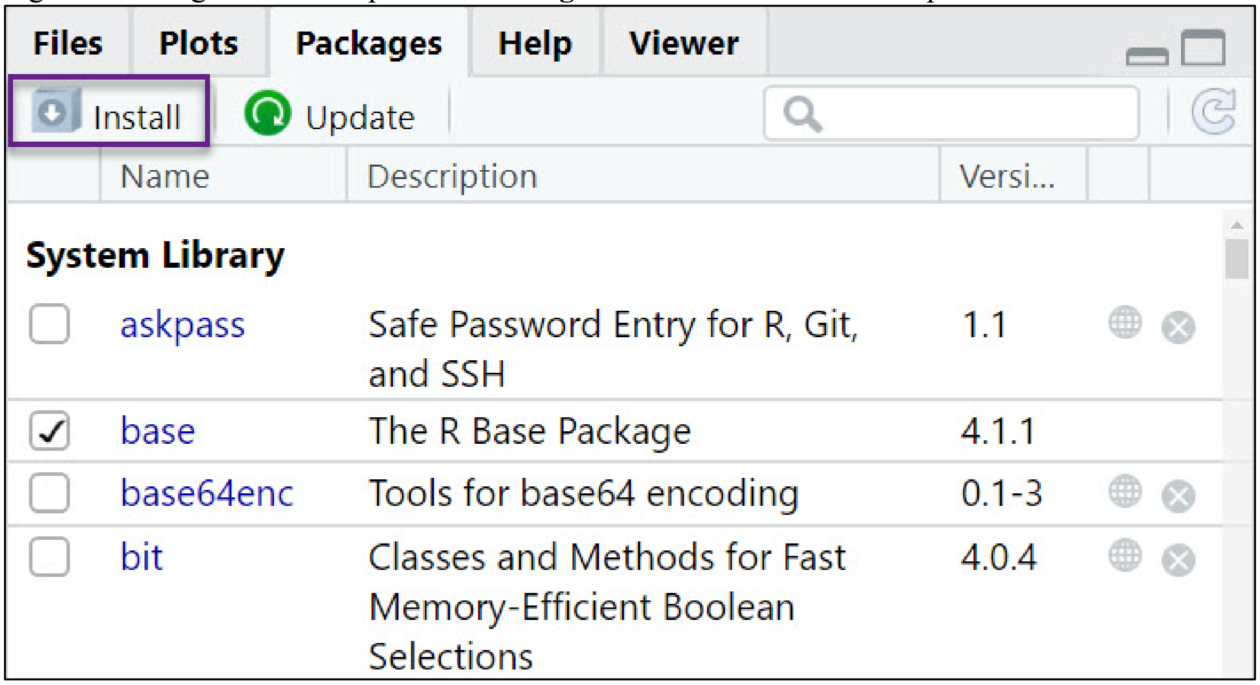
```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> utils:::menuInstallLocal()
* installing *source* package 'vegClass' ...
** using staged installation
** R
** data
*** moving datasets to lazyload DB
** byte-compile and prepare package for lazy loading
** help
No man pages found in package  'vegClass'
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
*** arch - i386
*** arch - x64
** testing if installed package can be loaded from final location
*** arch - i386
*** arch - x64
** testing if installed package keeps a record of temporary installation path
* DONE (vegClass)
> 
```

## 3.2 RStudio Desktop IDE

Once RStudio has been configured (refer to section 2.4), the vegClass R package will need to be installed so it can be accessed in R sessions. The follow steps describe how to install the vegClass R package.

1. Open RStudio and navigate to the **Packages** menu and then click the **Install** button (Figure 13).

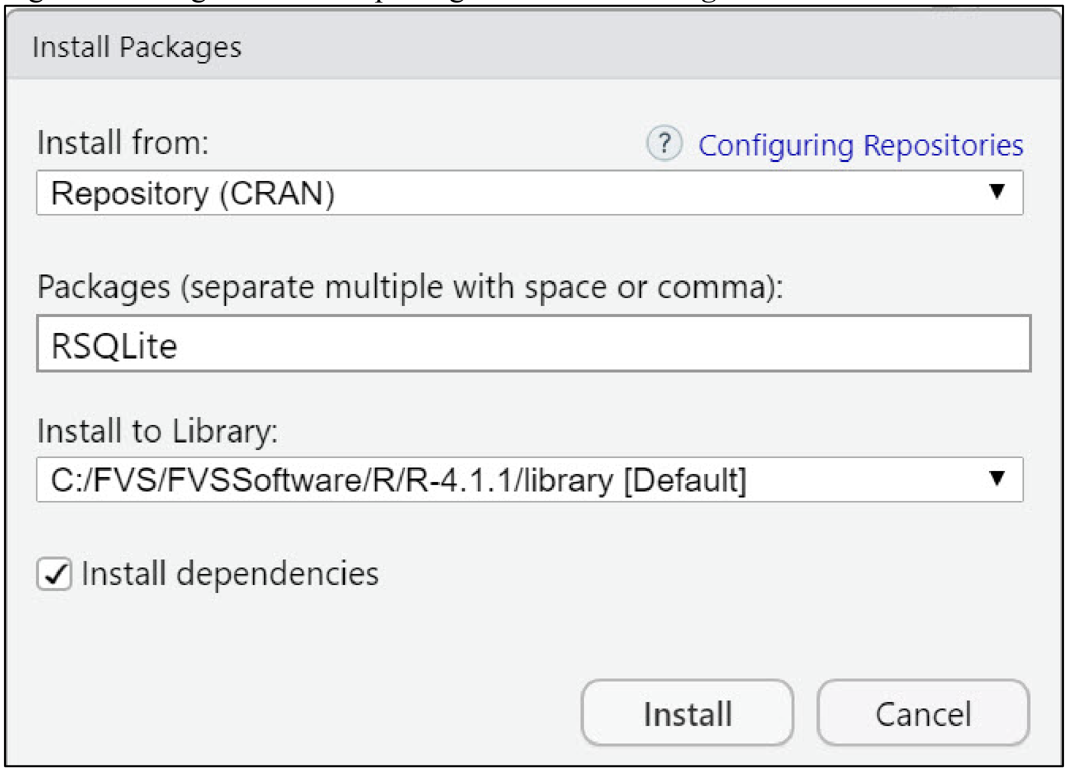
Figure 13. Image of Install option in **Packages** menu of RStudio Desktop.



| Name | Description | Versi... |
|---|---|---|
| **System Library** | | |
| ☐ askpass | Safe Password Entry for R, Git, and SSH | 1.1 |
| ✓ base | The R Base Package | 4.1.1 |
| ☐ base64enc | Tools for base64 encoding | 0.1-3 |
| ☐ bit | Classes and Methods for Fast Memory-Efficient Boolean Selections | 4.0.4 |

2. If you are NOT using the version of R that is distributed with FVS and you don't have the RSQLite package installed, then you will first need to install these packages now. You can search for packages in the **Packages (separate multiple with space or column)** field and then click the install button (Figure 14).

Figure 14. Image of selected packages in *Install Packages* window of RStudio Desktop.


Install Packages

Install from: Repository (CRAN)

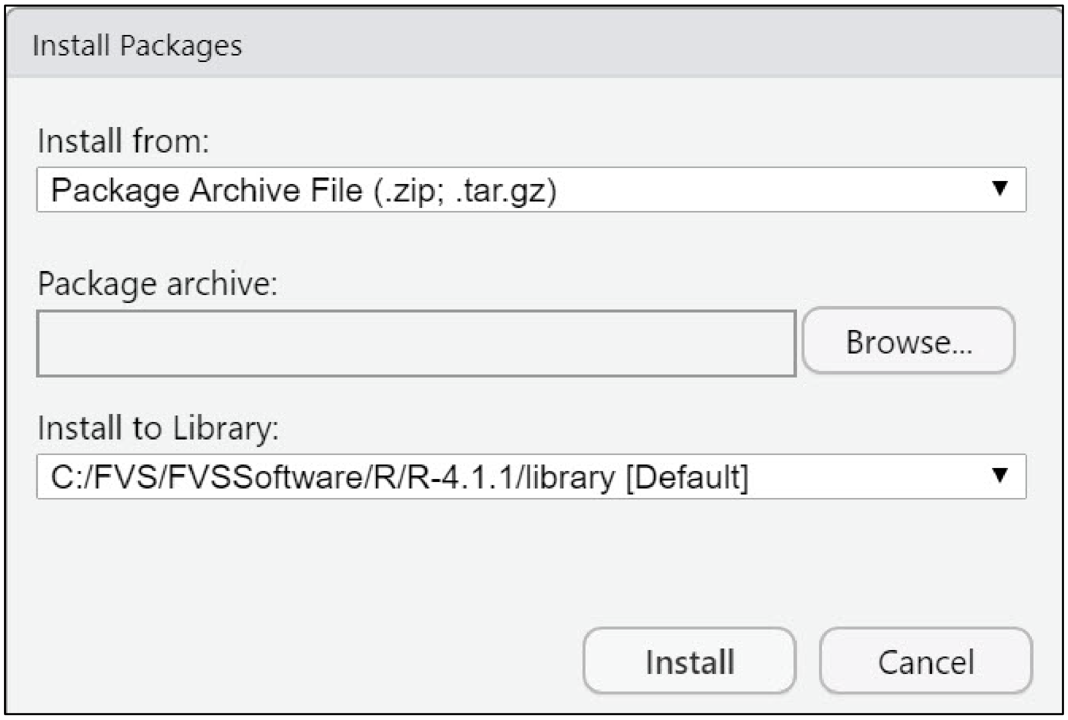
Packages (separate multiple with space or comma): RSQLite

Install to Library: C:/FVS/FVSSoftware/R/R-4.1.1/library [Default]

✓ Install dependencies

3. Now the vegClass package will need to be installed (vegClass_0.1.0.tar.gz). This package can be installed by selecting the **Package Archive File (.zip; .tar)** option from the **Install from:** dropdown list (Figure 15).

Figure 15. Image of **Package Archive File (.zip; .tar.gz)** option in *Install Packages* Window of RStudio.


Install Packages

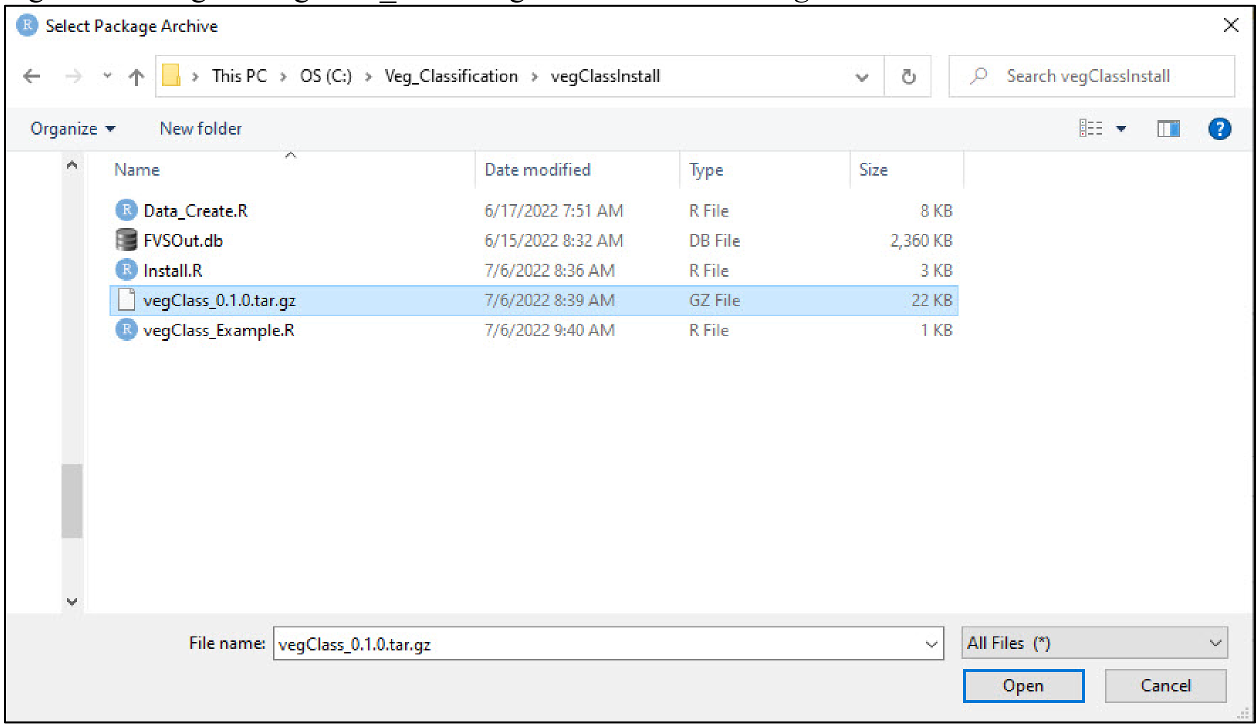
Install from:
Package Archive File (.zip; .tar.gz)

Package archive:

Install to Library:
C:/FVS/FVSSoftware/R/R-4.1.1/library [Default]

4. You will be prompted to navigate to the directory where vegClass_0.1.0.tar.gz is located. If you are not automatically prompted to browse for the installation file, then click the **Browse** button. Navigate to this location, select the vegClass_0.1.0.tar.gz file, and then click the **Open** button (Figure 16).

Figure 16. Image of vegClass_0.1.0.tar.gz file in *Select Package Archive* window.



Select Package Archive

This PC › OS (C:) › Veg_Classification › vegClassInstall

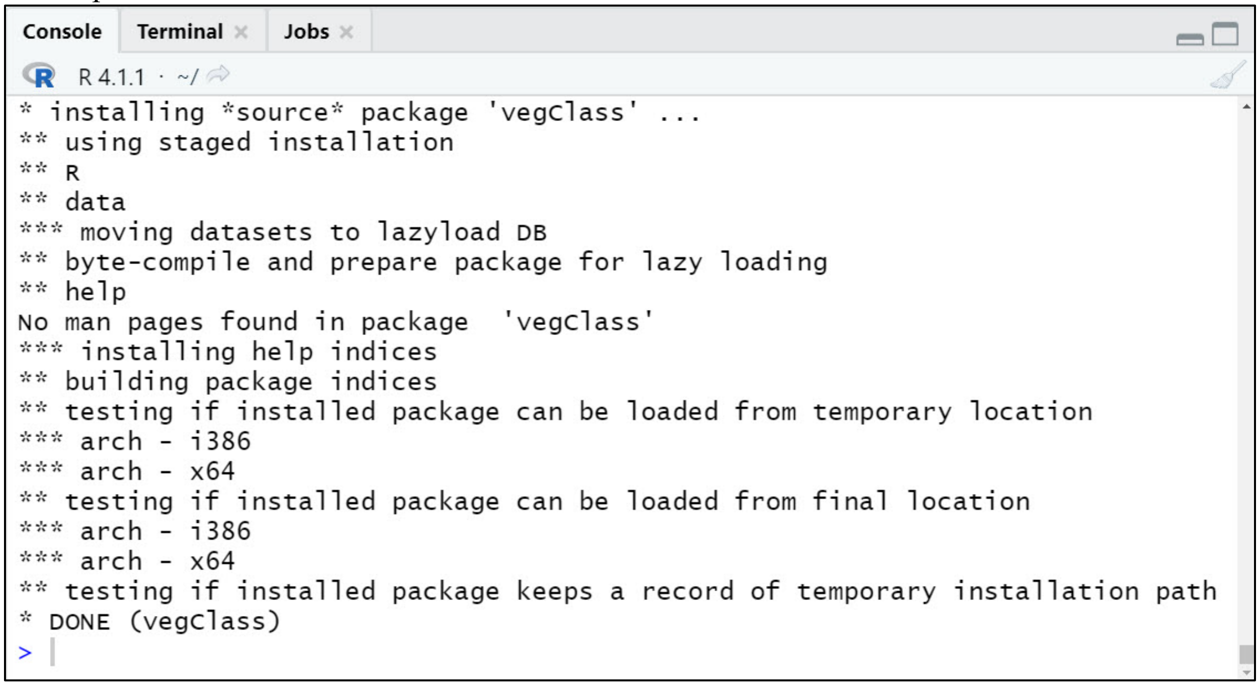
| Name | Date modified | Type | Size |
|---|---|---|---|
| Data_Create.R | 6/17/2022 7:51 AM | R File | 8 KB |
| FVSOut.db | 6/15/2022 8:32 AM | DB File | 2,360 KB |
| Install.R | 7/6/2022 8:36 AM | R File | 3 KB |
| vegClass_0.1.0.tar.gz | 7/6/2022 8:39 AM | GZ File | 22 KB |
| vegClass_Example.R | 7/6/2022 9:40 AM | R File | 1 KB |

File name: vegClass_0.1.0.tar.gz

All Files (*)

The information in Figure 17 should appear in the R console if the installation of the vegClass package is successful.

Figure 17. Image of R console output following installation of vegClass R package in RStudio Desktop.



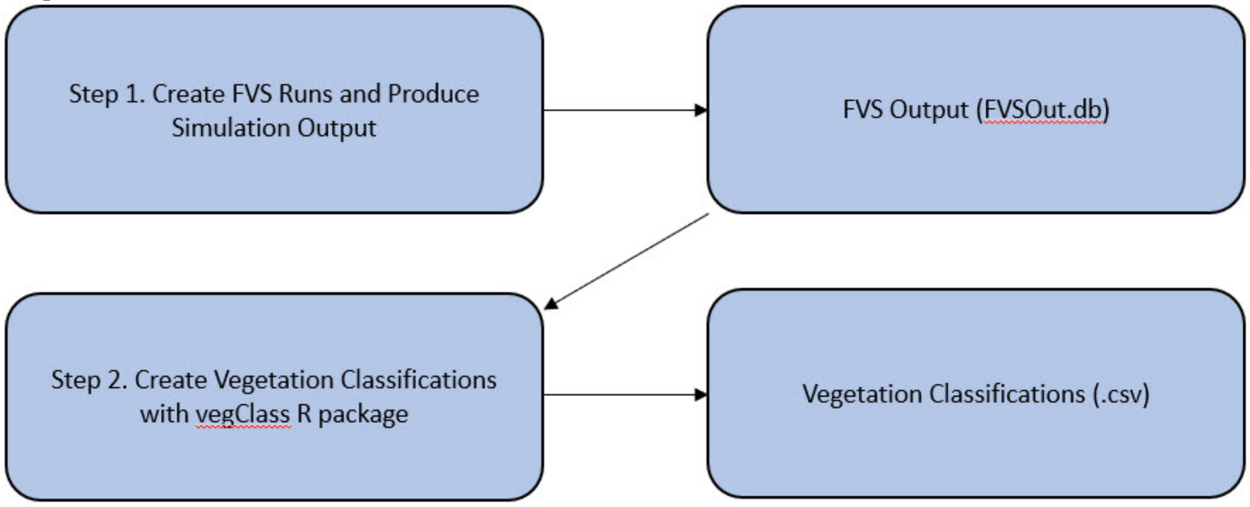
```
* installing *source* package 'vegClass' ...
** using staged installation
** R
** data
*** moving datasets to lazyload DB
** byte-compile and prepare package for lazy loading
** help
No man pages found in package  'vegClass'
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
*** arch - i386
*** arch - x64
** testing if installed package can be loaded from final location
*** arch - i386
*** arch - x64
** testing if installed package keeps a record of temporary installation path
* DONE (vegClass)
>
```

# 4.0 Workflow for Deriving Vegetation Classifications

## 4.1 General Processing Sequence

Deriving vegetation classifications with the vegClass package is a two-step process. First, the user will parameterize and run FVS simulations to derive stand and tree level output. The output derived in FVS simulations is stored in SQLite databases. The vegClass package is then implemented in R to derive vegetation classifications using the output SQLite database (.db) produced by FVS. The vegetation classifications will be sent to a comma separated value files (.csv). Figure 18 provides a conceptual diagram of the general processing sequence used for producing state and transition model input information.

Figure 18. Conceptual illustration of general processing sequence for state and transition model output.


Step 1. Create FVS Runs and Produce Simulation Output
FVS Output (FVSOut.db)
Step 2. Create Vegetation Classifications with vegClass R package
Vegetation Classifications (.csv)

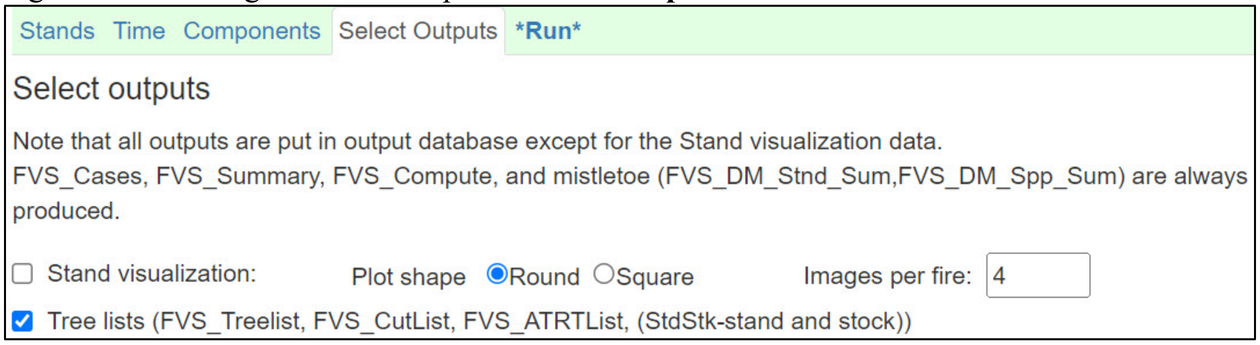
## 4.2 Producing Simulation Output with FVS

This section of the guide details how to setup projects in FVS and describes how output from FVS simulations is stored. It is assumed that users understand how to construct and run FVS simulations. Resources for learning more about the FVS software can be found on the Forest Vegetation Simulator training webpage: https://www.fs.fed.us/fvs/training/index.shtml and user guides webpage: https://www.fs.fed.us/fvs/documents/guides.shtml. A step-by-step FVS training guide (FVSTrainingGuide.pdf) can also be found in the Training Guides folder that comes installed with the FVS software (C:\FVS\Training Guides).

After FVS is installed, users should create a new FVS project. A project is simply a workspace for creating FVS runs that are linked to a single input database. The steps for creating a project are described on page 9 of the FVS training guide. Users can create as many projects as they would like for their work. Once a project has been created, the user should upload the inventory database that they want to produce state and transition input for. The steps for uploading an inventory database are also presented on page 9 of the FVS training guide.

Once a FVS project has been created and an FVS-ready input database has been uploaded, the user can begin to create FVS simulations and produce simulation output. There are no restrictions on what keywords and components can be included in simulations. However, certain output must be generated in simulations to produce inputs for state and transition model with the vegClass R package. The FVS_Summary2, FVS_Cases, and FVS_Treelists tables output database tables are required. By default, the FVS_Summary2 and FVS_Cases tables are generated for every FVS simulation that is run. The FVS_TreeList can be created by selecting the **Tree lists** options in the **Select Outputs** tab (Figure 19) or by invoking the TREELIBDB keyword through the **Keywords** tab (Figure 20) or in a keyword component file (.kcp file).

Figure 19. Selecting **Tree lists** output in **Select Outputs** tab in FVS interface.



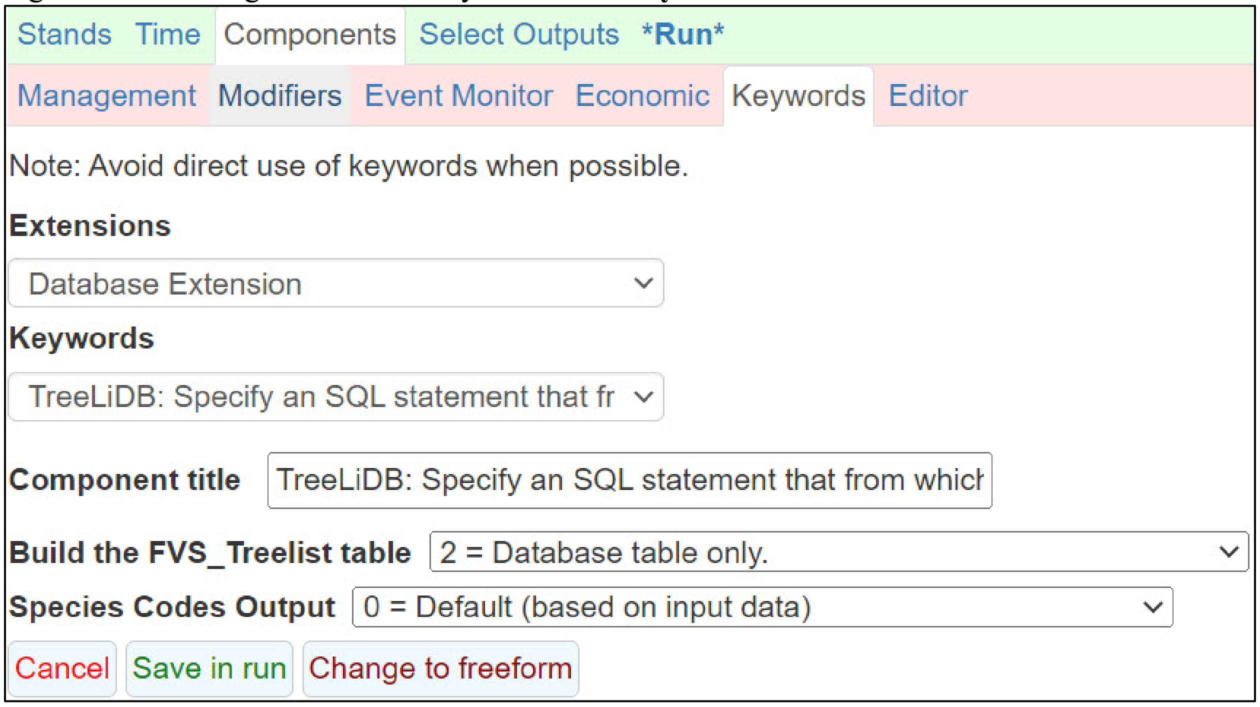
Select outputs

Note that all outputs are put in output database except for the Stand visualization data.
FVS_Cases, FVS_Summary, FVS_Compute, and mistletoe (FVS_DM_Stnd_Sum,FVS_DM_Spp_Sum) are always produced.

☐ Stand visualization: Plot shape [✓ Round] Square Images per fire: 4

☑ Tree lists (FVS_Treelist, FVS_CutList, FVS_ATRTList, (StdStk-stand and stock))

Figure 20. Selecting TREELIDB keyword from **Keywords** tab in FVS interface.



Note: Avoid direct use of keywords when possible.

Extensions: Database Extension

Keywords: TreeLiDB: Specify an SQL statement that fr

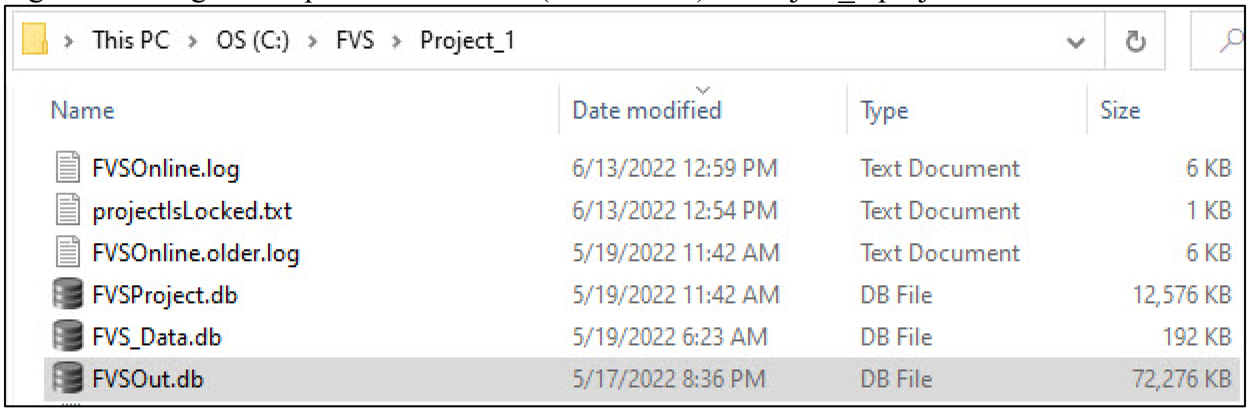
Component title: TreeLiDB: Specify an SQL statement that from which

Build the FVS_Treelist table: 2 = Database table only.

Species Codes Output: 0 = Default (based on input data)

Simulation output is stored in an SQLite database entitled FVSOut.db that is specific to the project the user is working from. By default, FVSOut.db files are stored in project folders that are located in the FVS folder that is created from the FVS installation. Project folders share the same name as the project being assumed in the FVS interface. The screenshot in Figure 21 shows an example FVSOut.db file that is stored in the Project_1, project folder.

Figure 21. Image of output FVS database (FVSOut.db) in Project_1 project folder.

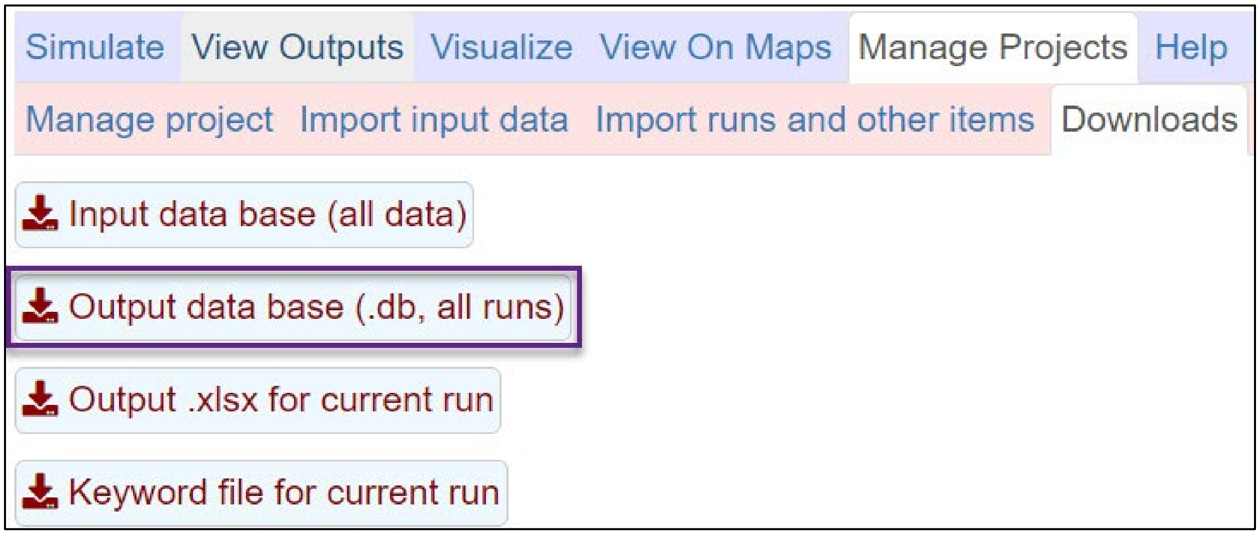


This PC › OS (C:) › FVS › Project_1

| Name | Date modified | Type | Size |
|---|---|---|---|
| FVSOnline.log | 6/13/2022 12:59 PM | Text Document | 6 KB |
| projectIsLocked.txt | 6/13/2022 12:54 PM | Text Document | 1 KB |
| FVSOnline.older.log | 5/19/2022 11:42 AM | Text Document | 6 KB |
| FVSProject.db | 5/19/2022 11:42 AM | DB File | 12,576 KB |
| FVS_Data.db | 5/19/2022 6:23 AM | DB File | 192 KB |
| FVSOut.db | 5/17/2022 8:36 PM | DB File | 72,276 KB |

When using the local configuration of FVS, users can make a copy of the FVSOut.db database by right clicking, selecting the **Copy** option, and then use the **Paste** option to store the database in a desired directory. In both the location and online configuration of FVS, the FVSOut.db can be downloaded and saved to a desired directory from within in the FVS interface as shown in Figure 22.

Figure 22. Image showing button used to download output FVS database in **Downloads** tab of FVS interface.


Simulate
View Outputs
Visualize
View On Maps
Manage Projects
Help
Manage project
Import input data
Import runs and other items
Downloads
Input data base (all data)
Output data base (.db, all runs)
Output .xlsx for current run
Keyword file for current run

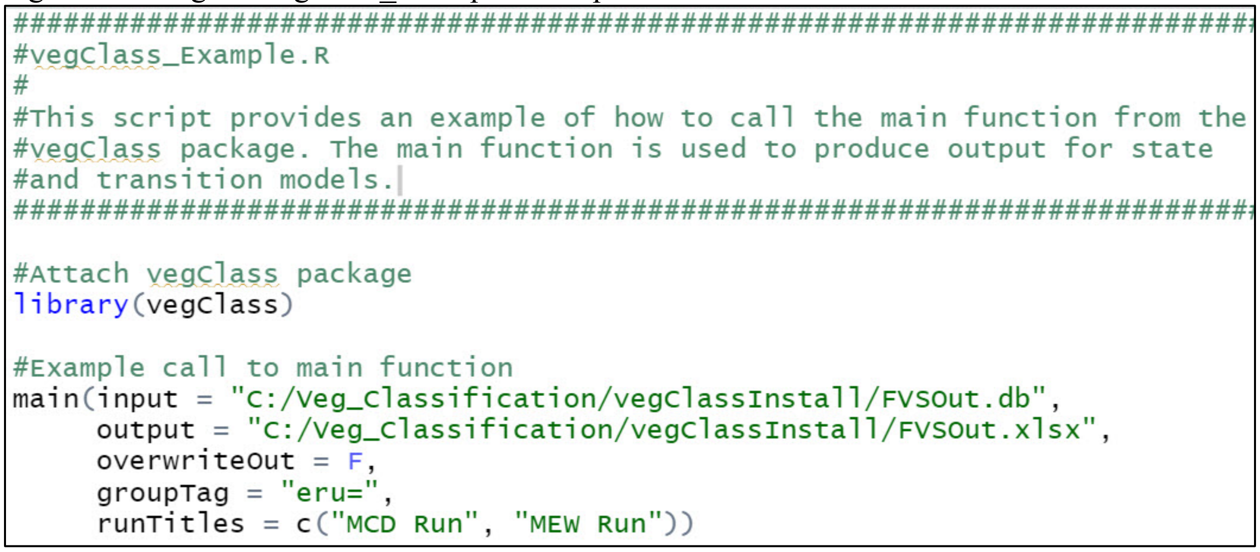
## 4.3 Using the vegClass Package

Once the desired number of FVS simulations have been run (refer to section 4.2), the vegClass R package will be used to derive information for state and transition models. Users should invoke the vegClass package through a R script (.R file). First time users should refer to the example script, vegClass_Example.R, located in the vegClassInstall folder. The vegClass_Example.R file can be opened by right clicking the file in File Explorer and selecting the **Open With** option followed by the selection RStudio or RGUI. Alternatively, users can first open either RStudio or RGUI and then click the **File** and **Open File…** (**Open script…** button in RGUI) buttons to browse for the vegClass_Example.R file**.** The components in the vegClass_Example.R file are described below.

All the lines that start with the # symbol are comments for the user and are not interpreted by R. The first line of code library(vegClass) is used to attach the vegClass package to the current R session (Figure 23). This line of code needs to be included in any R script that a user creates when the invocation of the vegClass package is desired.

Vegetation classification output is created by calling function main (Figure 23). There are 6 arguments that are used in this function that should be considered, and each are described below (Function descriptions are also available in Section 6.0). The input argument should be a directory path and file name to an output FVS SQLite database. The main function will use the information in this database to produce vegetation classifications. The output argument is a directory path and file name to a comma separated file (.csv). The information produced by the main function is sent to this file. The overWriteOut argument is a logical variable which controls how output is written to the file specified in the output argument. If the value of this variable is TRUE (T), then previously existing information in the output argument will be overwritten by information produced in the most recent call to the main function. If the value of this argument is FALSE (F), then information from the most recent call to the main function will be appended to previously existing information in the output argument. The groupTag argument is used to extract a group identifier (such as an ERU) from a set of FVS grouping codes. For instance, if you have a grouping code such as "ERU=MCD", then the groupTag would be "eru=" and you would have the group identifier MCD returned in the file specified in the output argument. The runTitles is a character vector of FVS run titles that will be processed by the main function. The run titles specified in this argument correspond to run titles that exist in the FVS project you are working with. Individual and multiple runs can be processed by a single call to the main function. Finally, the allRuns arguments tells the main function to process all the runs from the database specified the input argument. If the value of this argument is set to TRUE(T) then all runs will be processed and the runTitles argument will be ignored.

Figure 23. Image of vegClass_Example R script.

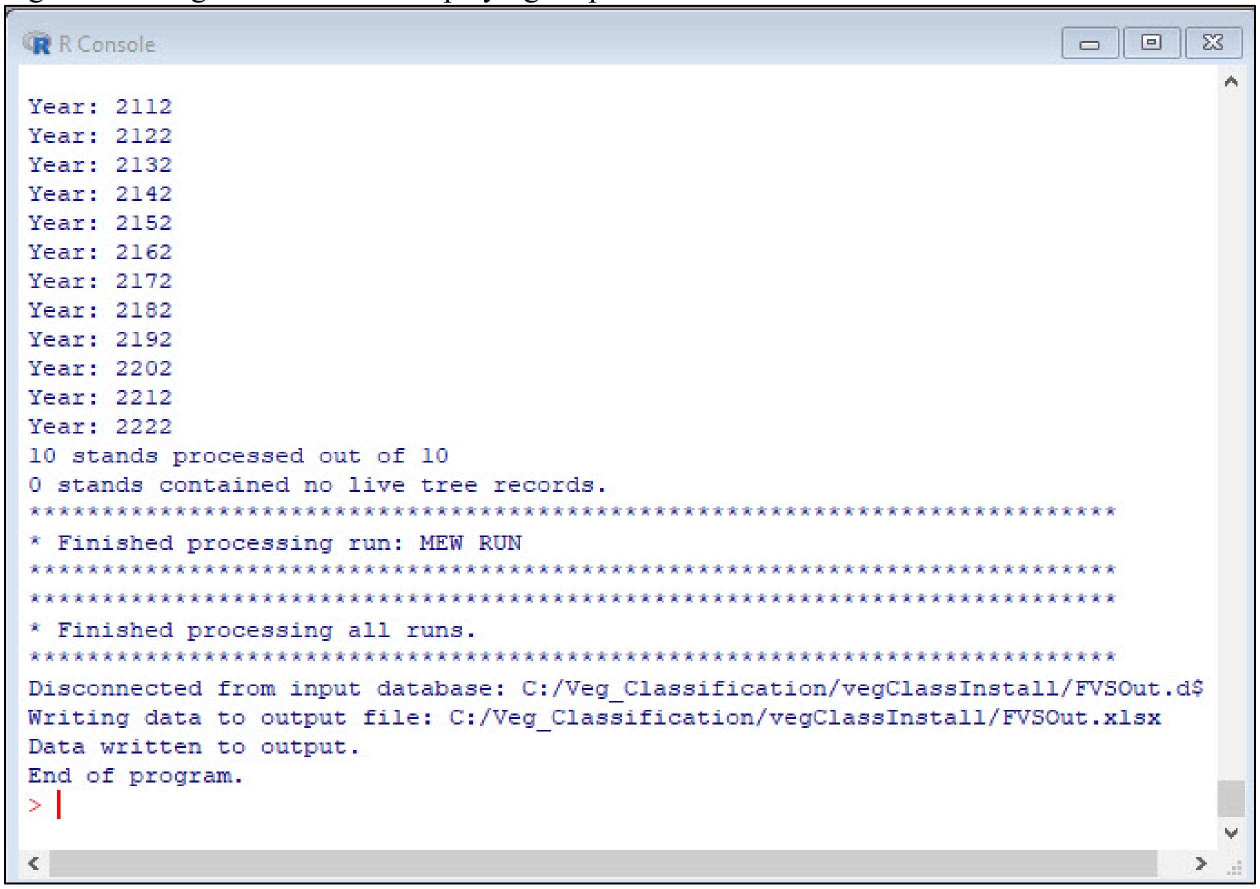


```
############################################################################
#vegClass_Example.R
#
#This script provides an example of how to call the main function from the
#vegClass package. The main function is used to produce output for state
#and transition models.
############################################################################

#Attach vegClass package
library(vegClass)

#Example call to main function
main(input = "C:/Veg_Classification/vegClassInstall/FVSOut.db",
     output = "C:/Veg_Classification/vegClassInstall/FVSOut.xlsx",
     overwriteOut = F,
     groupTag = "eru=",
     runTitles = c("MCD Run", "MEW Run"))
```

### 4.3.1 RGUI

The code in vegClass_Example.R can be run by highlighting the desired lines of code in the RGUI editor window and then pressing the **Ctrl + R** keys. Users can also run lines of code individually by placing the mouse cursor at the beginning of a line of code and pressing the **Ctrl + R** keys. When the main function is called, messages will be printed to the R console window. These messages report the runs and stands that have been processed by the main function and primarily serve as mechanism to inform the user that the main function is running. “End of program.” will be printed in the R Console once the main function completed its processing (Figure 24).

Figure 24. Image of R Console displaying output from main function in RGUI.

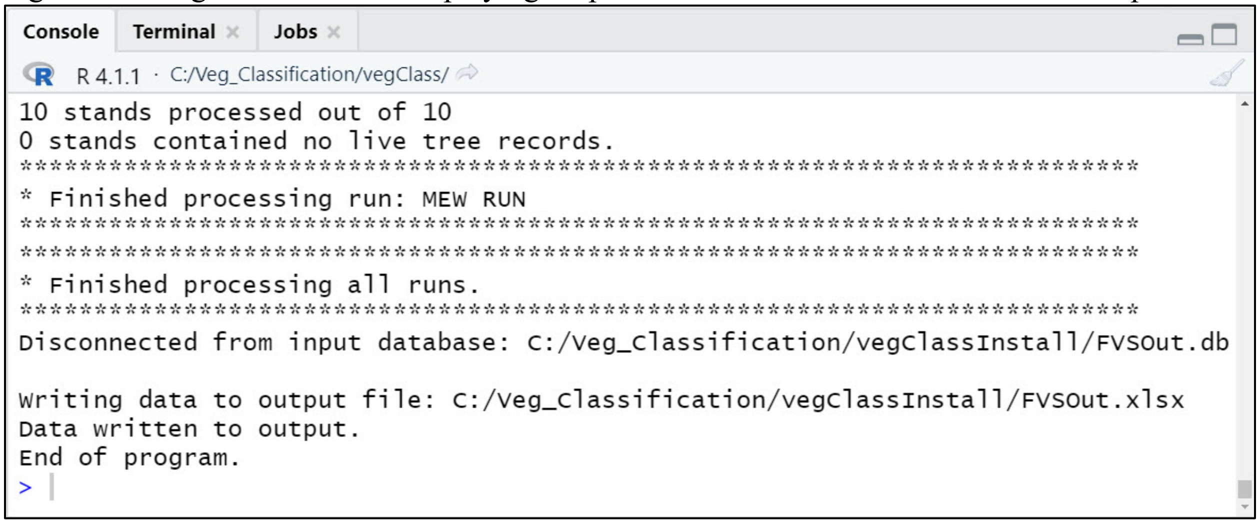


```
Year: 2112
Year: 2122
Year: 2132
Year: 2142
Year: 2152
Year: 2162
Year: 2172
Year: 2182
Year: 2192
Year: 2202
Year: 2212
Year: 2222
10 stands processed out of 10
0 stands contained no live tree records.
***************************************************************************
* Finished processing run: MEW RUN
***************************************************************************
***************************************************************************
* Finished processing all runs.
***************************************************************************
Disconnected from input database: C:/Veg_Classification/vegClassInstall/FVSOut.d$
Writing data to output file: C:/Veg_Classification/vegClassInstall/FVSOut.xlsx
Data written to output.
End of program.
> 
```

**4.3.2 RStudio Desktop IDE**

The code in vegClass_Example.R can be run by highlighting the desired lines in the RStudio Desktop IDE window and then pressing the **Ctrl + Enter** keys. Users can also run lines of code individually by placing the mouse cursor at the beginning of a line of code and pressing the **Ctrl + Enter** keys. When the main function is called, messages will be printed to the R console window. These messages report the runs and stands that have been processed by the main function and primarily serve as mechanism to inform the user that the main function is running. “End of program.” will be printed in the R Console once the main function completed its processing (Figure 25).

Figure 25. Image of R Console displaying output from main function in RStudio Desktop.



```
10 stands processed out of 10
0 stands contained no live tree records.
****************************************************************************
* Finished processing run: MEW RUN
****************************************************************************
****************************************************************************
* Finished processing all runs.
****************************************************************************
Disconnected from input database: C:/Veg_Classification/vegClassInstall/FVSOut.db

Writing data to output file: C:/Veg_Classification/vegClassInstall/FVSOut.xlsx
Data written to output.
End of program.
> 
```

## 5.0 Output Produced by vegClass R Package.

This section of the guide provides a description of the information that is produced by the vegClass R package. Table 5.0.1 lists the variables and definitions of each produced from function main.

Table 5.0.1 Vegetation classification variables reported by vegClass package.

| Variable Name | Description |
|---|---|
| RUN | Name of the FVS run title |
| GROUP | Name of FVS group associated with PLOT_ID |
| PLOT_ID | Plot/Stand/Point Identification Number |
| CY | FVS Projection Cycle |
| PROJ_YEAR | FVS projection Year |
| ST_AGE | Age of Plot/Stand/Point |
| TPA | Trees per acre (including seedlings and stems) |
| BA | Basal area per acre (including seedlings and stems) |
| CAN_COV | Canopy cover corrected for overlap (including seedlings and stems) |
| DOM_TYPE | Dominance type |
| DCC1 | Primary attribute in DOM_TYPE |
| XDCC1 | Canopy cover corrected for overlap represented by DCC1 |
| DCC2 | Secondary attribute in DOM_TYPE (can be NA) |
| XDCC2 | Canopy cover corrected for overlap represented by DCC2 |
| CAN_SIZCL | Canopy cover dominant size class: R3 mid-scale mapping |
| CAN_SZTMB | Canopy cover dominant size class: R3 timberland types |
| CAN_SZWDL | Canopy cover dominant size class: R3 woodland types |
| BA_STORY | Canopy Layers/Stories: R3 ruleset; basal area per 8" sliding diameter range |

## 6.0 Functions Available to User in vegClass R package

**Function** buildQuery

---

**Description**

This function builds and returns a string of SQL statements that can be used to read in data from the FVS_TreeList, FVS_Summary2, and FVS_Cases tables of the FVS database. This function is primarily invoked outside of the main function for testing purposes and can be used in the dbGetQuery function of the RSQLite R package.

**Arguments**

| | |
|---|---|
| stands: | Character vector of stand IDs. |
| runTitle: | Character string pertaining to FVS run title. |

**Value**

Character string of SQL statements.

**Examples**

```
#Call buildQuery
buildQuery(
        stands = c("Stand1", "Stand2"),
        runTitle = "Run 1"
)
```

**Function** baStory

---

**Description**

Calculates storiedness for stand using USFS Region 3 rules in accordance with (Vandendriesche 2013 pg. R3-4 – R3-5)**.**

**Arguments**

| | |
|---|---|
| stdYrFrame: | Dataframe that contains tree records for stand. Must include DBH and TREEBA as columns. |
| totalCC: | Uncorrected percent canopy cover of stand. |
| tpa: | Trees per acre of the stand. |

| | |
|---|---|
| ba: | Basal area per acre of the stand. |
| debug: | Logical variable used to specify if debug output should be printed to R console. If value is TRUE, then debug output will printed to R console. |

**Value**

Storiedness value (integer value from 0 - 3)

**Examples**

```
#Calculate storiedness
baStory(stdYrFrame, totalCC)
```

**Function** canSizCl

---

**Description**

Calculates a canopy size dominance class for stand using USFS Region 3 rules in accordance with (Vandendriesche 2013 pg. R3-3 – R3-4).

**Arguments**

| | |
|---|---|
| dat: | Dataframe that contains tree records for stand. Must include DBH and TREECC as columns. |
| totalCC: | Uncorrected percent canopy cover of stand. |
| tpa: | Trees per acre of stand. |
| type: | Integer value (1 – 3) used to determine which type of diameter class to return<br>1 - Midscale mapping (default value)<br>2 - Timberland dominance type<br>3 - Woodland dominance type |
| debug: | Logical variable used to specify if debug output should be printed to R console. If value is TRUE, then debug output will printed to R console. |

**Value**

Canopy size class value (integer value from 0 – 5)

**Examples**

```
#Calculate canopy size class using Midscale mapping.
calcCanSizCl(dat, totalCC, type = 1)
```

**Function** domType

**Description**

Calculates dominance type for stand using USFS Region 3 rules in accordance with (Vandendriesche 2013 pg. R3-1 – R3-3).

**Arguments**

| | |
|---|---|
| stdYrFrame | Dataframe that contains tree records for stand. Must include DBH and TREEBA as a column. |
| totalCC | Uncorrected percent canopy cover of stand. |
| debug: | Logical variable used to specify if debug output should be printed to R console. If value is TRUE, then debug output will printed to R console. |

**Value**

List containing dominance type (DOMTYPE), primary attribute in dominance type (DCC1), percent canopy cover represented by DCC1 (XDCC1), secondary attribute in dominance type (DCC2), percent canopy cover represented by DCC2 (XDCC2).

**Examples**

```
#Calculate dominance type
domType(stdYrFrame, totalCC)
```

**Function** fvsGaak

**Description**

Creates an FVS_GroupAddfilesAndKeyword (GAAK) table in dataframe format. This function is primarily used when creating an FVS-ready database. An example of its usage can be found in Data_Create.R file that is included in the vegClassInstall folder.

**Arguments**

| | |
|---|---|
| dbName: | Character string pertaining to name of database to be referenced in GAAK table. Default value is "FVS_Data" |
| type: | Variable to determine what grouping codes are included in GAAK table.<br>1 = Standard FVS grouping codes (All_Stands, All_Plots)<br>2 = FIA grouping codes (All_FIA_Conditions, All_FIA_Plots, All_FIA_Subplots)<br>3 = Both standard FVS grouping codes and FIA grouping codes |

Default value for type argument is 2.

**Value**

Dataframe containing contents of FVS GAAK table.

**Examples**

#Create FVS GAAK table
fvsGaak()

**Function:** main

**Description**

Produces a comma separated values file (.csv) that contains vegetation classifications.

**Arguments**

| | |
|---|---|
| Input: | Character string for directory path and file name to an FVS output SQLite database (.db). Directory path and file name must be surrounded with double quotes "" and double back slashes or single forward slashes need to be used for specifying directory paths. Input database must contain the following tables: FVS_Summary2, FVS_TreeList, and FVS_Cases or main function will terminate with an error.<br><br>Examples of valid formatting for input argument:<br><br>"C:/FVS/R3_Work/FVSOut.db"<br>"C:\\FVS\\R3_Work\\FVSOut.db" |
| output: | Character string for directory path and file name to output .csv. Directory path and file name must be surrounded with double quotes "" and double back slashes or single forward slashes need to be used for specifying directory paths.<br><br>Examples of valid formatting for output argument:<br>"C:/FVS/R3_Work/FVSOut.csv"<br>"C:\\FVS\\R3_Work\\FVSOut.csv" |
| overwriteOut: | Logical variable used to determine if existing file specified in output argument should be overwritten or have new data appended to it. If value is TRUE (T), any information existing in output will be overwritten with new information. If value is FALSE (F), then the new information will be appended to existing file specified in output argument. The default value of this argument is FALSE. |

groupTag: This is a grouping tag used to extract a group identifier (such as an ERU) from a set of FVS grouping codes (as defined in FVS_StandInit table). For instance, if you have a grouping code such as ERU=MCD, then the groupTag would be ERU= and you would have the group identifier MCD returned in the output. The value for groupTag must be surrounded in double quotes. The groupTag argument is case insensitive but must be spelled correctly.

Example group tag:
"ERU="

runTitles: Character vector of FVS run titles that will be processed from database defined in input argument. If runTitles is left as NULL, execution of main function will terminate. Each run title in argument must be surrounded by double quotes. The values specified for runTitles argument are case insensitive but need to be spelled correctly. If any of the values in runTitles are spelled incorrectly, the execution of main function will terminate.

Example of how to specify single run title:
runTitles = "Run 1"

Example of how to specify multiple run titles:
runTitles = c("Run 1", "Run 2",...) Note the use of the c(...) when specifying multiple run titles

allRuns: Logical variable that is used to determine if all runs in argument input should processed. If value is TRUE (T), then all runs will be processed and any runs specified in argument runTitles will be ignored. By default, this variable is set to FALSE (F).

**Value**

Character string containing “End of program” message.

**Examples**

```
#Using main function to process MCW Run and MEW Run
main(
        input = "C:/State_Transition/vegClassInstall/FVSOut.db",
        output = "C:/Users/markcastle/Desktop/FVSOut.csv",
        overwriteOut = F,
        groupTag = "eru=",
        runTitles = c("MCW Run", "MEW Run"),
        allRuns = F
)
```

**Function** pvConvert

**Description**

This function takes in a PV_Code and returns an ERU code. This function is currently only relevant to PV Codes used by USFS Region 3.

**Arguments**

| | |
|---|---|
| pv: | PV code as a character string. |

**Value**

ERU code character string. If the input PV_Code is not recognized, then NA is returned.

**Examples**

```
#Determine ERU from input PV Code
pvConvert("012340")
```

## 7.0 References

Vandendriesche 2013. A Compendium of NFS Regional Vegetation Classification Algorithms. Internal Rep. Fort Collins, CO: U. S. Department of Agriculture, Forest Service, Forest Management Service Center. 75p.