

# **Forest Carbon Dashboard Documentation**

Jeremiah Groom and Nadia Tase

2025-05-16

# Table of contents

<b>Preface {#pref, .unnumbered}</b>	<b>3</b>
<b>Acknowledgements</b>	<b>4</b>
<b>1 Introduction</b>	<b>5</b>
<b>2 Dashboard Summary and Use</b>	<b>6</b>
<b>3 Preparing Your Data</b>	<b>7</b>
3.1 Your data folder . . . . .	7
3.2 Features of table and figure scripts . . . . .	8
3.3 <i>General data set and script control features</i> . . . . .	8
3.3.1 Data set features . . . . .	8
3.3.2 Script control features . . . . .	8
3.3.3 The Global Tab . . . . .	9
3.3.4 General Figure and Table Tab Settings . . . . .	10
<b>4 Uploading Your Data</b>	<b>17</b>
<b>5 Process Your Data Locally</b>	<b>18</b>
5.1 Running the standalone code . . . . .	18
5.1.1 <i>Code Outline</i> . . . . .	18
5.1.2 <i>Details for using the Standalone Code by Section</i> . . . . .	18
<b>6 Frequently Asked Questions and Troubleshooting</b>	<b>22</b>
6.1 Things to be Fixed . . . . .	22
<b>7 Version Information</b>	<b>23</b>
<b>References</b>	<b>24</b>

# Preface {#pref, .unnumbered}

Welcome to the documentation for the Forest Carbon Dashboard. The Forest Carbon Dashboard conveys the state of carbon and carbon change (flux) for regions of land. The United States Department of Agriculture's [Forest Inventory Analysis Program](#) generates the information used in the dashboard. The information is processed and provided by FIA staff to regional (i.e., state-level) parties such as [Cal Fire](#). Cal Fire and the Oregon Department of Forestry (another user of this information) produce reports that provide summaries of ecosystem carbon for their regions on a semi-annual basis. One example is the report by Christensen et al. (2021). The motivation for the dashboard is the effort and size of the report: automating the process to the extent possible provides both publishers and readers with an efficient way to approach the sizable amount of data these projects generate. The motivation for this document is to provide users with an understanding of how to use the dashboard and how they can upload and view their own information.

The goal of the dashboard is to make ecosystem carbon data accessible to all. This includes the ability of users to browse and specify which data tables and figures they would like to use, and download the figures and the source data. It also includes the ability of users to upload their own data.

# Acknowledgements

Many thanks to Nadia Tase (California Department of Forestry and Fire Protection - CAL FIRE) and Olaf Kuegler (USDA Forest Service, Retired) for supporting the development of this work through feedback, discussion, and data revision. Amanda Reynolds (CAL FIRE) provided administrative support. This work builds upon the many revisions and iterations of the AB 1504 California Forest Ecosystem and Harvested Wood Product Carbon Inventory reports. *Who at CalFire did the figure coding?*

This project was funded by the California Department of Forestry and Fire Protection Purchase Order 3540-0000678085.

# 1 Introduction

## 2 Dashboard Summary and Use

In summary, this book has no content whatsoever.

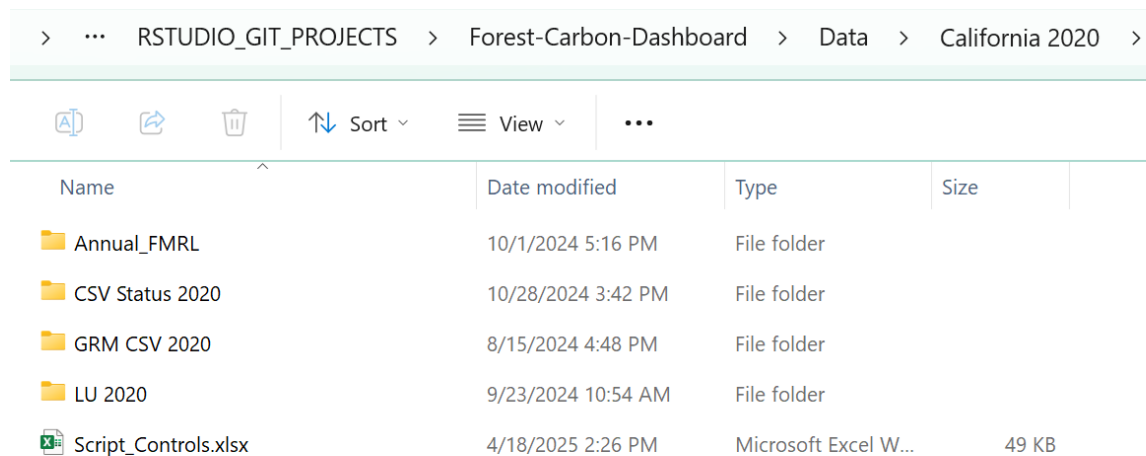
1 + 1

[1] 2

## 3 Preparing Your Data

### 3.1 Your data folder

Users provide a suite of folders that follow this format:



> ... RSTUDIO_GIT_PROJECTS > Forest-Carbon-Dashboard > Data > California 2020 >				
[Icons: Copy, Paste, Delete]   Sort ↑↓   View ≡   ...				
Name	Date modified	Type	Size	
Annual_FMRL	10/1/2024 5:16 PM	File folder		
CSV Status 2020	10/28/2024 3:42 PM	File folder		
GRM CSV 2020	8/15/2024 4:48 PM	File folder		
LU 2020	9/23/2024 10:54 AM	File folder		
Script_Controls.xlsx	4/18/2025 2:26 PM	Microsoft Excel W...	49 KB	

Figure 3.1: Required structure of data input folders and the script control Excel file.

Notice that the parent folder is called “California 2020”. The subfolders with years in them (CSV Status, GRM CSV, LU) need to have the same year in their name, which is the user’s year of interest. This is important! The script will look for these files with these year.

Be sure to include all of these folders! The code may get unhappy if they are missing. Include the folders but leave them empty if you have nothing to put in them (e.g., Annual\_FMRL and LU 2020 may be specific to California and a few other states).

The USDA Forest Service FIA program personnel can provide the CSV files that populate these folders. It should be evident which go where (they will give you CSV Status and GRM CSV folders, for instance).

We will now spend a great deal of verbiage describing what goes into Script\_Controls.xlsx.

## 3.2 Features of table and figure scripts

The table and figure scripts were built with flexibility in mind.

- CSV files provided by the FIA may change from year to year in column titles. The code finds key words or phrases in each target column and gives those columns specific names, which it then works with to create the figure or table.
- The Script\_Control.xlsx file allows users to control the terms that the table and figure scripts search for, use, and display. The scripts are built to be agnostic towards column values, relying entirely on what the user provides. The user must check columns to verify that the column values being sought exist and have not been altered.

## 3.3 *General data set and script control features*

### 3.3.1 Data set features

The CSV files provided by FIA staff may have extremely long file names. The names are so long that we recommend that users save the files close to their root directory. This hopefully allows users the ability of opening the CSV files. It is necessary for users to open these files so that they can verify that the items listed in Script\_Control.xlsx tabs are accurate or need updated. The code in the app is constructed to search for files using a unique string that is shorter than the actual file name.

### 3.3.2 Script control features

The file Script\_Control.xlsx (Script Control) provides specific information to the app on how to label files, where to find files, how to cope with specific features of CSV data files, and what processes to run. There are three distinct parts to Script Control: the Global tab, Fig tabs, and Tab tabs.

The Global tab contains general information on the region and years to be examined, ecoregions and ownerships specific to users' data, color palates to be used for ownerships, forest types, carbon stocks, Growth removal and mortality (GRM) fates, etc. There are also columns specific to running the Harvested Wood Products (HWP) model. These features help define aspects for all figures and tables, from caption text to figure colors and table categories.

The Fig tabs control all figures generated. Each Fig tab controls the creation of one or a few similar figures. The tabs provide information on what file names the code should search for, which folder to search in, specific figure text to provide, and figure control information.

The Tab tabs control all tables generated. Like the Fig tabs, the Tab tabs control the creation of one or a few tables. Many of the columns are the same and serve the same function as the Fig tabs. Some columns can offer specific information that is needed to generate a table. Some



Tab tabs are used repeatedly on different data set or require the combination of separate data sets to produce a single table.

Table and figure tabs were created around the tables and figures listed in Christensen et al. (2021). We recommend that users review that source (or the app output) to orient themselves regarding which data sets are relevant to which tables and figures.

### 3.3.3 The Global Tab

This section describes important aspects of the Script Control’s Global Tab. The user should review and alter all of these settings to align with their data and needs.

- **REGION:** This is the name of the region in question. Specifically, as can be seen in Figure 3.1, the parent folder is named “California 2020”. The REGION for this fold is ‘California’. The user should set their REGION similarly.
- **YEAR:** Like REGION, this should be set to the appropriate year for the data and it will/is be part of the parent folder’s title.
- **EARLY.YRS and CURRENT.YRS:** Years are broken into two sets. **CURRENT.YRS** are the most recent ten years ending in **YEAR**. **EARLY.YRS** are the ten years prior to **CURRENT.YRS**.
- **ECOREGIONS:** A list of ecoregions used within data sets and potentially in data CSV file names.
- **OWNERSHIPS:** This is a list of ownership names that show up repeatedly within CSV data files. Ensure that the spelling is consistent with data file values.
- **OWNERSHIP.COLORS:** The hexadecimal codes used for ownerships. Be sure the number of **OWNERSHIP.COLORS** matches **OWNERSHIPS**.
- **RESERVED.TYPES, RESERVED.COLORS:** Like **OWNERSHIPS**, these columns include forest type names by reserve status and the associated hexadecimal colors, respectively. Be sure the two columns are of equal length.
- **STOCKS, STOCK.COLORS:** Like **OWNERSHIPS**, these columns include carbon stock types by physical location on the landscape (e.g., roots, standing dead) and their associated hexadecimal colors, respectively. Be sure the two columns are of equal length.
- **GRM.TYPES, GRM.COLORS:** Like **OWNERSHIPS**, these columns include carbon associated with stand growth/removal/mortality categories (e.g., Fire, Gross Tree Growth) and their associated hexadecimal colors, respectively. Be sure the two columns are of equal length.

- **OWNERSHIP.RESERVED.TYPES, OWNERSHIP.RESERVED.COLORS:** These columns list general ownership types that include public reserved and unreserved lands and their associated hexadecimal colors, respectively. Be sure the two columns are of equal length.
- **RUN.HWP:** This column indicates whether the app upload (Chapter 4) or stand-alone (Chapter 5) code should attempt to run the [HWP model](#) (TRUE) or forego running the model (FALSE). It is primarily useful for the stand-alone script run on one's own computer. The app relies on users to select whether or not they want HWP data to be run, bypassing the importance of this setting.
- **SOURCE.GIT:** This setting is also only used in the stand-alone setting. Users may enter four options here: "OWN.DIR", "CA", "WA", and "OR". One of the three state abbreviations will cause the code to obtain the state's HWP data from the [HWP GitHub repository](#). "OWN.DIR" indicates that the user has their own HWP data they would like incorporated in the tables. The app by default selects "OWN.DIR".
- **OWN.HWP.DIR:** This option also is only relevant for users running the stand-alone script. Under this column heading the user lists is the folder location and file name for the user's HWP data when they entered "OWN.DIR" for the column "SOURCE.GIT". We recommend placing the Excel file in the repository's base file (wherever `getwd()` is for the project) and just listing the file's Excel name; e.g., 'Some\_Data.xlsx'. This simplifies what is entered. For instance, entering '/MyData/Home/Some\_Data.xlsx' opens oneself up to problems (was that initial backslash needed? Is the file path full and correct? Is the file path in relation to the working directory or the main directory?).

### 3.3.4 General Figure and Table Tab Settings

With the exceptions of Tab 6 and Tab 7, all table and figure tabs share certain columns.

- **DAT.FILE:** This is the unique string used to identify a data table used in a figure or table. For instance, Fig2.4 has a DAT.FILE listed as 'Figure\_2.4\_Percent'. The data file names are usually much longer than the DAT.FILE entry.
- **DAT.TABLE.LOC:** The location of the data table is stated here. It is used to identify the folder listed in Figure 3.1 generically. Users can enter 'CSV.STATUS', 'GRM.CSV', 'LU.CSV', or 'FMRL.CSV'. These generic versions of the folder names are used to facilitate code execution.

DAT.FILE and DAT.TABLE.LOC are used together for opening CSV files. The DAT.FILE string is searched for in file names found in the DAT.TABLE.LOC folders. There can be multiple DAT.FILE files listed for certain tables and figures with a single DAT.TABLE.LOC (see Tab4.12 and Figure 4.12). Two data tables may be needed from different folders, in which case the DAT.FILE and DAT.TABLE.LOC can have entries that correspond by row. For

Tab4.2a, DAT.FILE = ‘Table 4.2a\_’ and ‘\_LuseC’ while DAT.TABLE.LOC = ‘GRM.CSV’ and ‘LU.CSV’.

- FIG.NAME: This applies to figure tabs only. This value is the name of the figure that is to be produced and saved by the code. This value affects users running the stand-alone code, not the app.
- INCREMENT: Some figure tabs have this column. It represents the plotting increment to use on the y axis.
- IS\_CO2e: Yes/No. This applies to some figure tabs. The associated figure code has option for providing results in  $CO_2e$  or  $CO_2$ . The default setting is “Yes”,  $CO_2e$ . This option was developed because previous versions of the figure were produced in  $CO_2$  but current reporting metrics rely on  $CO_2e$ .
- TAB.CAP: This applies to some tables. There may be a reference to repeated tables in the appendix in which case this option controls the appendix reference.

Frequently for a table or figure tab there will be groups of three columns with a heading of `_.LISTED`, `_.USE`, `_.ORDER` (e.g., `OWNER.LISTED`, `OWNER.USE`, `OWNER.ORDER`) that are specific to the table or figure’s parent data set. One table or figure can have more than one of these sets of columns. The `_.LISTED` column indicates the values listed in a column that are of interest. The `_.USE` column allows the user to specify how the `_.LISTED` items are displayed (e.g., “Private Corporate” instead of “Corporate”). The `_.ORDER` column allows the user to specify the display order of the items in a figure; this column is not always present. Note that the `_.USE` and `_.ORDER` columns may repeat values for different `_.LISTED` values, as the table or figure script is combining these values.

### 3.3.4.1 Specific Figure Information

Below are descriptions of each figure and figure-specific settings.

#### *Fig2.4*

Percent of forest land base by landowner and reserve status.

- The columns ‘combine’ and ‘combine\_order’ are used to lump certain combinations of ‘Ownership’, ‘ReservedStatus’, and ‘Forest Type’ with the category ‘Other Forest Land’. Some ownerships/reserved status combinations combine ‘Productive’ and ‘Timberland’ with ‘Other Forest Land’ and other combinations do not. These may be adjusted to suit user needs.
- FOR.TYPE.LISTED: Note that there may be formatting issues with the provided CSV listed forest types. The code attempts to sidestep these peculiarities. The figure-producing script (Fig2.4.R) and associated functions (Functions.R) should be able to produce a working version of this figure without the `\xb9` text in this column.

*Fig4.1* Region's average annual net forestland  $CO_2e$  flux by owner, comparing findings from the initial 10 years with most recent 10 years.

- No other figure-specific columns.

*Fig4.2*

Region's average annual net forestland  $CO_2e$  flux by pool and owner, comparing findings from the initial 10 years with most recent 10 years.

- POOL.LISTED: For California 2020 and 2021 data, the listed category 'Foliage' is not found by the QA, nor is it in the data set. It was present in earlier versions of the figure for California. The SC will still work and the figure will be plotted.

*Fig4.3*

Region's average annual net timberland  $CO_2e$  flux by pool and owner, comparing findings from the initial 10 years with most recent 10 years.

- No other figure-specific columns.

*Fig4.4a*

Region's forestland average annual net  $CO_2e$  flux in above-ground live carbon by owner, comparing findings from the initial 10 years with most recent 10 years.

- No other figure-specific columns.

*Fig4.4b*

Region's timberland/reserved productive forestland average annual net  $CO_2e$  flux in above-ground live carbon by ownership, comparing findings from the initial 10 years with most recent 10 years.

- No other figure-specific columns.

*Fig4.6a*

Ecoregions' average annual net  $CO_2e$  flux by pool and owner, comparing findings from the initial 10 years with most recent 10 years.

- This figure relies on several different CSV files, one per ecoregion.
- Like Fig4.2, 'Foliage' does not appear in the data set but did appear in earlier versions of this figure for California.

*Fig4.6b*

Region forestland average annual net  $CO_2e$  flux by ecoregion, comparing findings from the initial 10 years with most recent 10 years.

- POOL.COLOR: This column provides hexcode color values for carbon pools.

*Fig4.8*

Region average forest carbon stock by owner,  $CO_2e$ .

- No other figure-specific columns.

*Fig4.9\_4.10* This tab provides information for the region's average carbon stock by pool and owner (Figure 4.9) and land status and owner (Figure 4.10).

- Both figures rely on the same CSV data set.
- The columns POOL.LISTED1, POOL.LISTED2, POOL.USE, and POOL.ORD are used

together but are not processed in the same fashion as other column sets (e.g., the OWNERS columns in this tab). POOL.LISTED1 and POOL.LISTED2 are used together to match with two columns in the CSV data in a `left_join()` function by the Fig4.9\_4.10.R code.

*Fig4.11*

Region average forest carbon stock by land status and ecoregion.

- No other figure-specific columns.

*Fig4.12*

Region average carbon stock by pool and forest type.

- This tab is used to inform/produce Figure 4.12 as well as the ecoregion figures for A1.x.1 (x = ecoregion number, the combination references Appendix section A1, figure A1.x.1 in Christensen et al. (2021)). Under DAT.FILE the first file listed is for the entire region, the others reference specific ecoregions. Be sure the Global tab ecoregion order matches the order of ecoregion files under DAT.FILE.
- The script produces two figures, one for  $CO_2e$  and the other for  $CO_2$ .

*Fig4.13*

Region average carbon stock by land status and forest type, metric tons  $CO_2$  and  $CO_2e$  per acre.

- This tab is used to inform/produce Figure 4.13 as well as the ecoregion figures for A1.x.2 (x = ecoregion number, the combination references Appendix section A1, figure A1.x.2 in Christensen et al. (2021)). Under DAT.FILE the first file listed is for the entire region, the others reference specific ecoregions. Be sure the Global tab ecoregion order matches the order of ecoregion files under DAT.FILE.
- The script produces two figures, one for  $CO_2e$  and the other for  $CO_2$ .

*Fig4.14* Region carbon density by pool and forest type, metric tons  $CO_2$  and  $CO_2e$  per acre.

- DAT.FILE: A carbon accounting CSV file and acreage data file are used to calculate metric tons per acre.
- The script produces two figures, one for  $CO_2e$  and the other for  $CO_2$ .

### 3.3.4.2 Specific Table Information

Below are descriptions of each table and table-specific settings.

*Tab4.1*

Region average annual net  $CO_2e$  flux from forest pools in forest land remaining forest land, comparing findings from the initial 10 years with most recent 10 years.

- NET.REMOVE: This value can be blank. In 2020 the California data set included a carbon pool value labeled 'NET FLUX'. That row was not needed and was removed. Therefore, this entry serves to remove up to one carbon pool.

*Tab4.2a*

Region average annual net  $CO_2e$  flux from forest pools, non- $CO_2$  emissions from forest fires in

forest remaining forest land, and changes due to forest land conversion. The table compares findings from the initial 10 years with most recent 10 years.

- DAT.FILE, DAT.TABLE.LOC: Two separate tables are required, one from the LU folder.

#### *Tab4.2b*

Average annual net  $CO_2e$  flux from forest ecosystem pools and non- $CO_2$  emissions from forest fires in Forest land remaining forest land.

- POOL4.2b.FPD.LEVELS: Note that the first entry here is “Forest Practice District”. See the associated table; this row will be blank. This column is not associated with the other POOL columns directly which is why it is OK that its length is or can be different.

#### *Tab4.3*

Region-wide estimate of average annual net carbon flux in  $CO_2e$  by pools and owner. Changes in  $CO_2e$  due to land-use and non- $CO_2$  greenhouse gas emissions are not included. The table compares findings from the initial 10 years with most recent 10 years.

- No other table-specific columns.

#### *Tab4.6*

This tab controls the creation of 4+ tables: Average Annual Net Change in Carbon Stocks for Aboveground Pools on Forest Land by Disturbance, Forest Land Status and Owner Group by:

\* The entire region

\* All counties

\* All national forests

\* All forest practice districts or the equivalent (e.g., Oregon has Forest Practice Areas). The number of these is flexible but must correspond to available CSV data.

- DAT.FILE: The order of data files is Region (State), County, National Forests, and then one per Forest Practice District or equivalent. - REG.COUNTIES.APP.TABLE.4.6 is a reference list that should probably have been listed as TAB.CAP. See TAB.CAP, above.

- \_\_.LEVELS columns are the ordered values from different columns. Users should be sure that their respective data column levels are captured here. There was not an effort to clean up displayed level names; names from the CSV file are used.

#### *Tab4.7*

Annual Net Emissions of Non- $CO_2$  Greenhouse Gases from Fire; actually includes  $CO_2$ .

- The script sources Tab4.6.R because it combines information from that table with the CSV file listed for this table.

- GENOWNER.\_\_, OWNER.\_\_, TYPE\_\_ columns: These four columns are used together. The table has two ‘Total’ columns, one for private forest and one for everything (in the script this is referred to as the ‘Gen.Total’). The blank cells are filled in by the script as ‘Gen.Total’. This could be confusing, but the user should examine Table 4.7, this tab, and their CSV table and ensure that everything lines up. If not, the user should make adjustments to the values in this tab accordingly.

#### *Tab4.8*

Annual change in forest land area to/from other IPCC land-use classes in the region. The

table compares findings from the initial 10 years with most recent 10 years.

- The pairs of columns for LAND.\_\_, TYPE.\_\_, and CHNG/FLC are not manipulated by a function but are used as-is. Ensure that both sets of columns for each pair, the \_\_.USE and -.LISTED, columns, properly capture the data set's listed labels and the desired labels.

#### *Tab4.9*

Annual change in carbon pools due to change in land use between forest and nonforest in the region. The table compares findings from the initial 10 years with most recent 10 years.

- The columns FTC.LISTED and CHNG.4.9.USE are used as a pair; see notes for Tab4.8. Otherwise, there are no other table-specific comments for this table.

#### *Tab4.10*

Area of forest land remaining forest land by ownership group and region.

- No other table-specific columns.

#### *Tab4.11*

Area of Forest Land by Forest Type, Owner Group and Forest Land Status for the entire region.

- No other table-specific columns.

#### *Tab4.12*

This tab controls the creation of 4+ tables: Forest land carbon stock by ownership and land status, where results in MMT  $CO_2e$  and thousand acres by:

- \* The entire region
- \* All counties
- \* All national forests
- \* All forest practice districts or the equivalent (e.g., Oregon has Forest Practice Areas). The number of these is flexible but must correspond to available CSV data.
- DAT.FILE: The order of data files is Region (State), County, National Forests, and then one per Forest Practice District or equivalent.
- The current script Tab4.12.R relies on the values within the CSV file column for carbon stock values (e.g., values include Live trees, Down wood, etc.). Hopefully the provided carbon stock value text in that column is similarly usable in future data sets provided by FIA.

#### *Tab4.31*

Forest carbon pools by 10-year inventory period. The table compares findings from the initial 10 years with most recent 10 years.

- No other table-specific columns.

#### *Tab4.32*

Live tree carbon stocks by ownership and land status. The table compares findings from the initial 10 years with most recent 10 years.

- No other table-specific columns.

#### *Tab4.33*

Forest carbon stocks by ownership and land status. The table compares findings from the

initial 10 years with most recent 10 years.  
- No other table-specific columns.

#### *Tab6*

This tab controls the creation of tables 6.5, 6.6, and 6.9. These three tables only summarize HWP values. Specifically, this tab is used to rename the HWP owner groups. The owner groups are column labels (not including the first column, “Year”) listed at the top of the ‘Harvest’ Excel tab for the HWP data input file. This table is described in the [HWP documentation](#). For example, if there is an owner column for ‘Private and Tribal’, the table will be processed by the HWP code that the Forest Carbon Dashboard sources and the ownership name is transformed to “Private.and.Tribal”. Tab6 offers a remedy for strange HWP output names.

#### *Tab7*

This tab controls some caption content for tables 7.1, 7.2, and 7.3. All three of these tables combine and summarize ecosystem carbon estimates with HWP estimates. The caption text is relevant to the reporting period of interest.



## 4 Uploading Your Data

In summary, this book has no content whatsoever.

1 + 1

[1] 2

## 5 Process Your Data Locally

### 5.1 Running the standalone code

The standalone code is provided so that users may generate their tables and figures without the aid of the Shiny app. We recommend reading this text alongside the opened `Stand_Alone.R` file. The user needs to make a few changes for the script to work.

#### 5.1.1 *Code Outline*

The Settings section in the standalone code relies on user input for reading data files correctly and sources the `Functions.R` script. If the user already has an RDS data file on hand, they can set options to skip the RDS-building step of ‘Create RDS File’.

Within the ‘Create RDS File’ step the code runs a QA of the provided data files. The `QA_Code.R` script reads the `Script_Control.xlsx` file for setting information to check against the CSV data files. The `Excel_to_RDS.R` builds the input RDS file from `Script_Control.xlsx` settings and the CSV data files. If the user provides information about an HWP data set, the `Excel_to_RDS.R` script will source the `HWP_Run.R` script to include HWP data in the RDS file. The RDS file is saved locally.

The ‘Read RDS file and load conditions’ section loads the RDS file and obtains settings from the RDS file. The standalone code can proceed with producing tables and figures. The ‘Produce figures and tables’ section sources individual figure and table scripts which each draw upon data from the RDS data file.

#### 5.1.2 *Details for using the Standalone Code by Section*

##### *Settings*

In the subsection “Enter values”, users must enter the `year.use` value which is the year of their dataset.

Users must also enter the `region.use` value, their region name. Note that this will produce text that matches the user’s folder/RDS name.

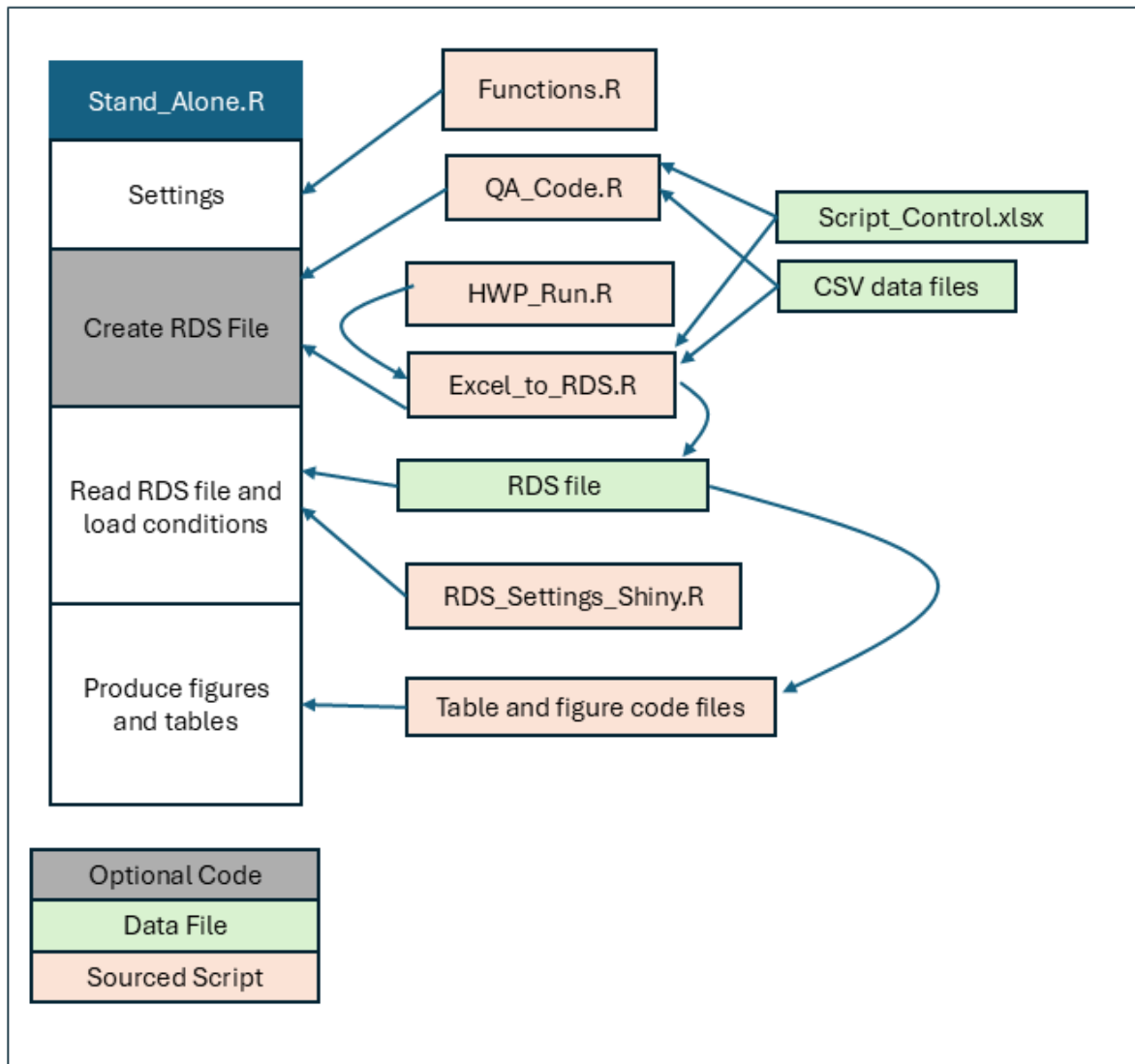


Figure 5.1: Data and code source files associated with `Stand_Alone.R`

If the user is loading existing data, the `EXISTING.RDS` needs to be set to `TRUE`. The script will look for the resulting RDS file. If `EXISTING.RDS` is set to `FALSE`, the script will attempt to read files and generate the RDS (giving it the name defined by `DATA.FOLDER.NAME`).

If the user is wanting to generate a fresh RDS from their parent data folder, we recommend placing that folder in the repository/project directory folder “Data/”. That will make things operate in a more straightforward manner.

Does the user want to have figure scripts save figures to a results folder? If so, `SAVE.PNG` option should be set to `TRUE`. This is an option because the Shiny app relies on the same figure code; there is no point for the figure code to internally save the images (users can download the figures from the app).

The ‘Other settings’ subsection provides R with information on where folders are located. If the user is relying on the project file structure provided by the GitHub repository (recommended) then the user need not alter anything in this sub section.

Leave the `SHINY.QA` setting at `FALSE`. This allows the Standalone code to save the RDS file. If `TRUE`, the RDS file will not be saved. The logic is that when the Shiny code compiles the data and performs the QA checks, there is no sense in having the code save the RDS file to the Shinyapps.io instance. The app user can instead click on a button and download the RDS file.

#### *Create RDS file*

This code may be run whether or not a new RDS file is desired. The code will be skipped if a new RDS is not wanted. If used, the code will rely on the information from above (file names) when it sources the `QA_Code.R` file. This will run the QA routine and write the `QA_Report_Table.csv` file. This file can be informative for determining process errors.

The `Excel_to_RDS.R` script compiles the RDS file from the `Script_Control.xlsx` file, the CSV files, and (if selected and available) the output HWP files. The settings in the control file, `Script_Controls.xlsx`, will determine whether the HWP code is run and provide information for processing the tables and figures. The `Excel_to_RDS.R` script will save the output RDS file to the local directory.

#### *Read RDS File and Load Conditions*

The script reads the RDS file that had been created earlier or was recently created by the code for the previous section. The script also checks to see if the FMRL and LU folders are empty. This information is used to determine which tables and figures to produce below.

#### *Produce Figures and Tables*

The object `tabfig.run` is important. It will list which tables and figures the QA deems available. If a particular table or figure is not being produced (or produced with a ‘no data’ message) type in `tabfig.run` to see if the QA prevented the figure’s production, then backtrack to the `QA_Report_Table.csv` to see why this might be the case.

Almost every figure and table is sourced from a separate file. The exceptions are tables and figures that rely on the same code or function to generate them. Users will likely select to save figure .png files (see above). The table code saves individual XLSX documents.

## **6 Frequently Asked Questions and Troubleshooting**

### **6.1 Things to be Fixed**

- For tables, allow for null values for TAB.CAP to trigger code that removes reference to the repeated Appendix table (i.e., remove “See also” from the caption text).

## 7 Version Information

In summary, this book has no content whatsoever.

1 + 1

[1] 2

## References

Christensen, Glenn A, Andrew N Gray, Olaf Kuegler, Nadia A Tase, Mark Rosenberg, and Jeremy Groom. 2021. *AB 1504 California Forest Ecosystem and Harvested Wood Product Carbon Inventory: 2019 Reporting Period Data Update*. Sacramento, CA: California Department of Forestry; Fire Protection; California Board of Forestry; Fire Protection. [https://bof.fire.ca.gov/media/beddx5bp/6-final\\_forest\\_ecosys\\_hwp\\_c\\_2019\\_feb2021\\_all\\_ada.pdf](https://bof.fire.ca.gov/media/beddx5bp/6-final_forest_ecosys_hwp_c_2019_feb2021_all_ada.pdf).