

Organizing Your Project Files

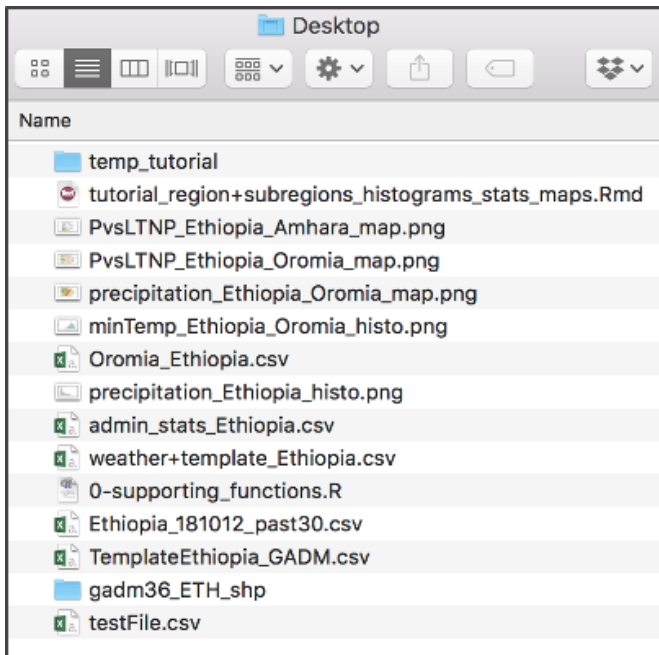
aWhere Training Tutorial

Introduction

Why is file structure important? The short answer is that it enables you to find different input and output files to efficiently generate data products, charts and maps to generate insights and produce reports to achieve your objectives. The structure proposed here has evolved over the years based on practical experience that we want to pass on to you to position you for success in the use of R and QGIS.

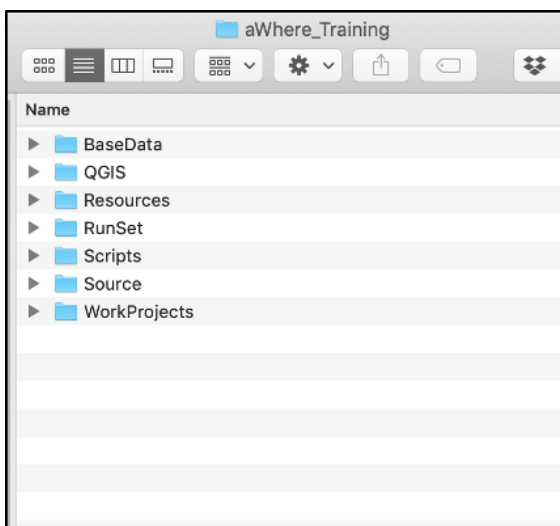
It's best practice to keep all of your project files grouped together in one place. Ideally, this place is a single folder on your computer with a descriptive name so you and your colleagues can easily locate this project in the future.

Over the years, we have found that the following folder and file structure has offered an effective way to manage your code, input files, and output files. While it is tempting to just store all your files on the desktop for a single session (**Option A below**), it does make it difficult to find outputs in the future when compared to **Option B** that organizes your work for future reference and re-use to make you more efficient in the use of R and QGIS.



Option A

Option A has all of the files saved onto the desktop. There is no clear organization; it is not clear which files are inputs, which are outputs, and where the code is. What does this project even do? Where to start? This structure is not ideal.

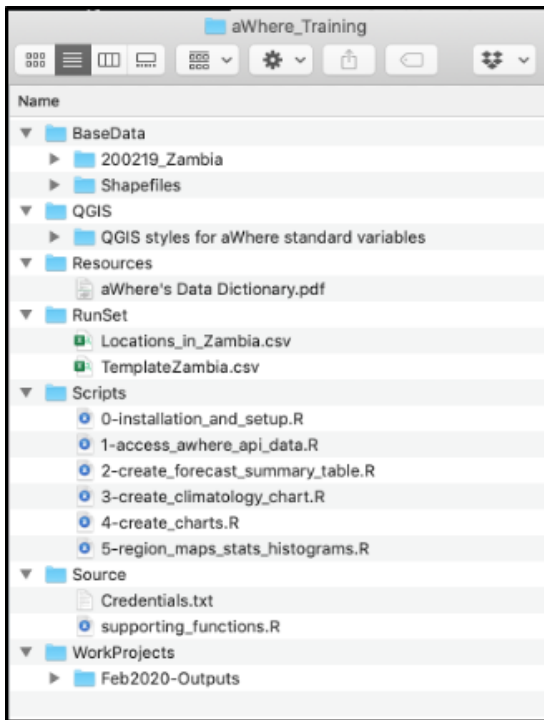


Option B

Option B appears more organized and easy to navigate. Files are grouped into clearly labeled folders based on their role in the project: input data, output data, and code.

The suggested structure and contents for each folder in **Option B** are described here along with where you can find these resources.

<p>BaseData</p>	<ul style="list-style-type: none"> ● Weather data files (Geospatial csv files downloaded from the adaptER Platform - refer to <i>Tutorial 2: Accessing Resources on aWhere's adaptER Platform</i>) <ul style="list-style-type: none"> ○ Example: <i>200219_past30.csv</i> ● Shapefiles <ul style="list-style-type: none"> ○ Example: Global Administrative Areas (GADM) shapefiles describing administrative region boundaries. You can download these shapefiles from https://gadm.org/ for the country of interest. These shapefiles will have a ".shp" file extension along with the country code and admin level number in their filename.
<p>QGIS</p>	<ul style="list-style-type: none"> ● QGIS styles for aWhere weather variables <ul style="list-style-type: none"> ○ These styles include predetermined color ramps for variables found in the Geospatial csv files and can be downloaded in the adaptER Platform refer to <i>Tutorial 2: Accessing Resources on aWhere's adaptER Platform</i>) ● GIS output maps: Save all QGIS maps and map layouts here
<p>Resources</p>	<ul style="list-style-type: none"> ● Tutorials and guides <ul style="list-style-type: none"> ○ Example: <i>aWhere Data Dictionary.pdf</i> (download in the adaptER Platform)
<p>RunSet</p>	<ul style="list-style-type: none"> ● Template data files containing information about each grid cell, which will have a filename similar to <i>TemplateZambia.csv</i> (<i>download it here</i>) ● Locations files (.txt or .csv) <ul style="list-style-type: none"> Example: <i>Locations_in_Zambia.csv</i> (<i>download it here</i>)
<p>Scripts</p>	<ul style="list-style-type: none"> ● Scripts: R-Training-Tutorials from aWhere Github
<p>Source</p>	<ul style="list-style-type: none"> ● Credentials text file <ul style="list-style-type: none"> ○ Example: <i>Credentials.txt</i> ● Supporting Functions file <ul style="list-style-type: none"> ○ Example: <i>supporting_functions.R</i> (found in aWhere Github)
<p>WorkProjects</p>	<ul style="list-style-type: none"> ● Outputs from R Scripts ● Note: create subfolders here to add additional organization to this folder. <ul style="list-style-type: none"> Examples include the date of the analysis or country of analysis ○ Example: <i>Feb2020-Outputs</i>



Here is a screenshot of this folder structure and contents. When it comes to organizing data and code for your projects, **Option B** will certainly be easier for you and your collaborators to utilize in the future. To optimize the efficiency and reproducibility of your projects, this tutorial describes some useful organizational techniques.

Tip: Avoid using spaces when naming folders and files; use underscores or dashes instead. This makes your filenames more machine-readable since spaces can cause issues when specifying the path to files

Working Directory

Within the R-training-tutorial scripts, you will set your working directory which is where all of your outputs will be saved from R. This will be the WorkProjects folder or subfolder within WorkProjects. This will be reviewed in more detail in the R tutorials that follow in this series.

In RStudio, you can set your working directory using the **setwd()** function and supplying the absolute path to the project folder as the input parameter. This absolute path must be in the form of a character string, which means it needs to be surrounded by double or single quotes: `setwd("C:/awhere_training/")`

Use the **getwd()** function without any input parameters and R will return the

current working directory:

```
getwd()  
[1] "C:/aWhere_training/"
```

By setting your working directory, you won't have to give R "absolute paths" to find each of the data files. "Absolute paths" start at the very top level of your computer and include the entire sequence of folders that lead to the file of interest. For example:

Absolute Paths

Example: "C:/aWhere_training"

Instead, by setting the working directory to be our project folder, R expects to find everything within this folder and we can just supply "relative paths" that point to each file of interest during our analysis. For example, if we set the working directory to be the absolute path of the project folder (i.e. C:/aWhere_training/WorkProjects/), then we can use much shorter paths to specify where files are located within our analysis.

Relative Paths

```
BaseData/countryName_YYMMDD_past30.csv  
RunSet/Template_countryName.csv
```

Additional Organization Tips

Introduce your project with a README

A README file is a form of documentation that introduces and explains a project. It is called a "README" file so the user reads it first! An effective README will include information such as:

- What is the purpose of this project?
- What data are required? Where did they come from?
- What outputs will be generated?
- How to get started with installations and running the code?
- Who can be contacted for help or more information about the project?

Here's an example of a very simple yet informative README file:

```
README.md
1 # Ag Observatory training tutorial
2
3 This repository holds Ag Observatory materials to generate histograms, statistics,
  and maps for a region and selected subregions.
4
5 Code
6
7 - All scripts and functions are located within the "code" folder.
8
9 Input_data
10
11 - Weather, template, and shapefiles used by the code are located within the
  "input_data" folder. These input files are grouped further into the following
  subdirectories:
12
13   - weather_data
14     .csv containing aWhere weather observations for a series of grid cells,
     each with a unique location id
15
16   - template_data
17     .csv containing geographic location data for each location id
18
19   - shapefiles
20     .shp and supporting files with administrative boundaries for the region
21     and subregions
22
23 Output_data
24
25 - Figures (histograms and maps with .png extensions) and spreadsheets (.csv files
  containing statistics) generated by the code are saved here.
26
27 For more information about this project, contact:
28 Drew Marticorena, drewmarticorena@aWhere.com
```

Naming Folders and Files

Here are some tips for creating file and folder names:

- Be descriptive! Instead of naming your file “test_v1.csv”, use more details so you and/or your colleagues will know that this file contains.
 - For example, “awhere_tutorial_countryName_stats_admin1.csv”.
- As mentioned earlier - avoid using spaces when naming folders and files; use underscores or dashes instead. This makes your filenames more machine-readable since spaces can cause issues when specifying the path to files.
- When using dates in a filename, consider using the ISO 8601 format: YYYY-MM-DD.
- If your scripts are meant to be run in sequence, consider numbering them in such a way that they can be easily alphabetized in order.
 - For example,
 - 0-setup.R
 - 1-pre_process_data.R
 - 2-run_analysis.R

- 3-summarize_results.R

What's next?

Review the next tutorial in the series to access the geospatial files using aWhere's Platform with the Tutorial: Accessing Resources on the adaptER Platform.

If you have any questions, please contact customersupport@awhere.com