Saguaro FFI QAQC Process

Eva Deegan 3/8/2024 [eva\_deegan@partner.nps.gov](mailto:eva_deegan@partner.nps.gov)

This document contains information of the process of QAQC for FFI data developed by Eva Deegan during her Scientists in Parks internship for Saguaro National Park in 2023 – 2024 as well as information on how to use the code that was developed. For any questions please reach out to Eva at her personal email (NPS discontinued post internship) [edeegan2@alumni.nd.edu](mailto:edeegan2@alumni.nd.edu), her supervisor Windy Bunn [windy\_bunn@nps.gov](mailto:windy_bunn@nps.gov), or the current Fire Ecologist at Saguaro, [gabriel\_dejong@nps.gov](mailto:gabriel_dejong@nps.gov).

# Running QAQC Code

## Information

* To run QAQC code on FFI data, data needs to be exported from FFI and downloaded to a location on the user’s computer (not the remote server). R will accept csv files which is the default export format for FFI
  + For the structure of the R markdown and functions data should be exported by protocol for each project unit e.g., Tree – Individual (Metric) for PSME plots, 1990 – 2023. For trees and seedlings, add preferred life form.
  + All code referenced in this document is currently available on Eva’s github at <https://github.com/edeegannps/FFIqaqc>, a publicly available repository. datacleaningfunctions.R is in the folder R.
    - Here is a short (watch 1 - 5 min) youtube tutorial on how to clone a repository from github. [Use Other People's Code on GitHub in 15 Min (THE EASY WAY) (youtube.com)](https://www.youtube.com/watch?v=CT1zTWyZX5w)
    - Here is a tutorial (1 min) for copying the code. Either option works. https://youtu.be/8XNU00VqQWo?feature=shared
* datacleaningfunctions.R is an R script with 24 functions defined to check FFI for common data issues in data from protocols for cover, fuels, post burn severity, seedlings, saplings, and trees
  + Optional: The script testing\_package.R in the FFIqaqc github repository lists each function with a ? before them. To learn about a specific function, run the line of code that lists the function with the ?, and a description should appear in the help file window of R studio.
* Results\_QC\_\*Mtype.Rmd is a formatted R markdown file designed to use these functions. It sources in the base code from datacleaningfunctions.R, reads in the FFI data csv files and runs the functions. It assigns each function to an object “flags” to export data issues to an excel file at the end.
* Results\_QC\_\*Mtype.html is the file produced when the Results\_QC\_\*Mtype.Rmd file is knitted to an html. The R markdown file includes code to format the html for readability. The html file will print the checks run on the data and whether they will pass those checks. There is a separate tab for each protocol to toggle through. It will also print if there are any errors in the code preventing the functions from being properly run.

## Instructions

1. Download data from desired monitoring type from FFI and save to local server
2. Make a copy of Results\_QC\_PSME.Rmd and save with desired monitoring type replacing PSME in file name
3. In the third code chunk, reading in data, insert the unique pathways to link to
   1. The working directory with FFI data (setwd function)
   2. The name of the vegetation type to be checked (mtype object)
   3. The location of the most up to date version of datacleaningfunctions.R on local server (source function)
   4. The location of the data from each protocol on the local server (samp, cover, fuel1000, duff, fine, saps, seeds, tree)
   5. For help: [How to Import a CSV File into R (example included) – Data to Fish](https://datatofish.com/import-csv-r/#:~:text=Here%20is%20the%20full%20code%20to%20import%20the,stored%20on%20your%20computer%29%3A%20df%20%3D%20read.csv%28%22C%3A%5CUsers%5CRon%5CDesktop%5CTest%5CProducts.csv%22%29%20print%28df%29)
4. Click the down arrow next to knit and click knit to html
5. Examine the html file for any errors (and useful QAQC info!)
   1. If errors are found – feel free to either contact me (Eva) or take a stab at editing the base code (datacleaningfunctions.R) to address the issue (push these changes to GitHub so I can check them out!). If it’s a warning message, usually all the code is still running fine, and it can be ignored. If lots of the checks seem to be missing, or there is an error with the format flags function so that the excel file is not being created, contact me so that I can help.

# Excel tracker sheet

## Information

* The last function in the Rmd files and datacleaningfunctions.R is format\_flags. This function exports a list of “flags” or issues with the data to an excel file. This excel file should appear after knitting the html in the set working directory under the name \*Mtype flags\_QAQC \*todays date.xlsx. If the file doesn’t show up, make sure that the format flags function hasn’t thrown an error in the html file.
* Data issues are separated by plot and each plot has a different sheet in the excel file. Each sheet should have 5 column headers – Issue, Resolved, Resolved\_by, Action\_needed, and Other\_notes
* The last sheet in the excel file includes any comments in the data that are atypical (excluding common comments such as “no data collected”, or “tag missing”)
* This is a workspace to explore and evaluate data issues and each user should feel free to approach it how best works for them. The excel sheet will be used to complete the QAQC report so whatever process make that the easiest is fine. The only “rule” per se is to only put Yes or No in the resolved column as the code in historicalvsyearly.R uses that data.

## Instructions

1. Open excel file and examine to make sure it looks correct, all plots are included, and preview which data issues are showing up.
   1. If a specific issue is being flagged hundreds of times, I would encourage some big picture data checking before getting into the minute. Make sure the relevant function is running properly, the data was downloaded correctly, issues are being correctly flagged.
2. Begin going through each issue, verifying data with original scans of datasheets and in FFI downloaded data. I do a lot of this in R, but it can be done however. The goal is to determine if it is an actual issue in the data that is relevant for analysis and can be fixed in FFI
   1. Each flag should include information about the sample event (monitoring status and macroplot) affected, as well as the tree tag or fuel transect as needed.
3. If the issue is not something that can be fixed, put Yes in the Resolved column, and put your initials and the data in the Resolved\_by column. For these types of issues, I put “note in QAQC report in the action needed column” and any other relevant information in the other notes column.
4. If the issue is something that can be fixed, put No in the Resolved column (I still date and initial in Resolved\_by) and briefly explain the action needed in the action\_needed column. Many categories in the QAQC report require evidence or support for the action needed and the other\_notes column is a great place to put this information.
5. Complete all sheets for each plot. I don’t follow the same process for comments – a skim is usually fine – but if its helpful go for it.

# QAQC report – Before FFI corrections

## Information

* The QAQC report is a non-automated document in word that summarizes data issues, assigns solutions and priorities for fixing. Descriptions of the types of issues (Repeat Issues (Conclusive), Repeat Issues (Inconclusive), Unique Issues, and Known Issues (No Solution)) are included in the template document under Introduction along with definitions of solution and priorities.
* The template document includes many common issues, places to put the affected monitoring status and macroplot sample event and tree tag as needed. Many solutions and priorities have been assigned but feel free to adjust as needed

## Instructions

1. Make a copy of the QAQC report template and save under the desired monitoring type.
2. Replace title with monitoring type and subtitle with author and date
3. Use excel tracker sheet to put information about data issues in the word document, noting the relevant sample event (monitoring status and macroplot) and tree tag or fuel transect if needed. Include evidence supporting a data issue as needed.
4. Insert new issues and adjust solutions and priorities as needed.
5. When all the data issues have been recorded in the document, go through, and delete any issues that don’t have any instances for this monitoring type. Update the table of contents.
6. Export the QAQC report to a pdf with the date and monitoring type in the file name.

# FFI corrections – Updating report

## Information

* Any changes made to data in FFI need to have a comment
* This section of the document will be updated when Eva and Windy complete this step of the process for the first time :)

## Instructions

1. Make a copy of the QAQC report, changing the title from before FFI corrections to FFI corrections.
2. Beginning with high priority, conclusive issues, make changes to data in FFI adding a comment in FFI each time with date and initials.
3. For each data issue that is corrected, document changes in QAQC report with a new line under priority labeled “Correction:”.

# Historical vs Yearly Data Issues comparison

## Information

* The data cleaning functions and Rmarkdown document is code that was designed to be run on an entire historical dataset. Many of the functions are designed to compare recent data with historical data, including the Lazarus tree function, consistency of fuel slope and azimuth over time, and changing DBH after death. For this reason, each year after data is collected, I recommend running the QAQC code on the entire data set for each protocol.
* The script historicalvsyearly.R is designed to compare two different excel tracker sheets, sort through the flagged issues, and assign one of three statuses; New, Previously flagged and unresolved, or Previously flagged and resolved. The excel sheet is sorted by this status so that the newest issues are at the top of each sheet and the new excel file is exported with the name \*mtypeflags\_wStatus\*todaysdate.xlsx.
* Optional: An additional script (filter\_for\_solving\_issues.R) was written to make resolving issues easier. This script reads in an excel tracker sheet and removes any data issues that have something written in the resolved column (\*YES AND NO). I.e., it only keeps the issues for which the resolved column is blank or na. This script exports a new excel file with the name \*mtypeflags\_newissues\*todaysdate.xlsx.

## Instructions

1. In the script historicalvsyearly.R, replace the pathway in setwd() with the pathway unique to the location of your excel tracker files.
2. Replace the pathway in both the excel\_sheets() function and the lapply() function assigning to the object historical with the pathway to the historical tracker sheet.
3. Repeat step 2 for the newest yearly tracker sheet. This tracker sheet is made by downloading the most recent data from FFI and running the QAQC code (see instructions 1.2)
4. Select all the code (shortcut Ctrl +A) and click run at the top (shortcut Ctrl + enter). This should export a new excel file with the assigned status of each issue.
5. Optional: To use the filter\_for\_solving\_issues.R script, replace the pathway in the excel\_sheets() function and lapply() function to the pathway to your desired excel tracker sheet. Run the code.
6. As always, if running into any errors please contact me (Eva) so that I can help.