Tabular Data Guide

Tetra Tech

March 27, 2024

## Introduction

The following user guide pertains to the [*summary\_AU\_table.R*](https://github.com/KateriSalk/Alaska_IR_Automation/blob/main/Code/7_Results/summary_au_table.R)script developed by Tetra Tech for the Alaska Department of Environmental Conservation (AK DEC). This guide assumes that the user has relatively recent versions of R and [RStudio](#X6589fc6ab0dc82cf12099d1c2d40ab994e8410c) installed and that they are familiar with the R coding language. Please direct any questions regarding the usage of these scripts to Amber Bethe Crawford, AK DEC (amber.crawford@alaska.gov).

## Purpose

The purpose of the *summary\_AU\_table.R* script is to summarize the analysis results for each assessment unit for the ArcGIS online hub.

## Required Packages

The only required packages for the *data\_processing.R* code is [*tidyverse*](https://www.tidyverse.org/) *.* Installation for the *tidyverse* package can be found in the Data Pull Guide. Below is a code chunk that demonstrates how to load these packages.

####Load Packages####  
library(tidyverse)

## Required Inputs

This code has specific exterior files that are required for it to run. The inputs are listed below. All input files are loaded at the beginning of the script.

| File Name | Purpose | Source |
| --- | --- | --- |
| AUID\_crosswalk.csv | Connects previous AU IDs to current AU IDs. Required to connect to previous year ATTAINS information | AK DEC |
| ATTAINS\_AK\_AsessmentUnits\_DataDownload\_20240126.xlsx | Provides information from previous years ATTAINS | ATTAINS |
| categorized\_aus\_20240222.csv | Provides necessary metadata for ATTAINS csv outputs | *categorize\_au.R* |

## Combine all AU IDs

In order to connect the AU metadata to the summary, we need to use the \*AUID\_crosswalk.csv\* information. The next section of code pulls previous ATTAINS information with the current AU IDs, previous AU IDs, and historical AU IDs. None of the AU ID numbers are overridden in this code. This information is then combined together to provide previous metadata to the current year’s upload.

#Join with previous AU ATTAINS by each AUID type  
data\_current\_AU <- data\_cat\_AUID\_added %>%  
 mutate(assessmentUnitId = AUID\_ATTNS) %>%  
 inner\_join(previous\_au\_attains, by = c('assessmentUnitId'))  
  
#Find retired AUs  
data\_retired\_AU <- data\_cat\_AUID\_added %>%  
 mutate(assessmentUnitId = Retired\_AUID) %>%  
 inner\_join(previous\_au\_attains, by = c('assessmentUnitId'))  
  
#Find historical AUs  
data\_historical\_AU <- data\_cat\_AUID\_added %>%  
 mutate(assessmentUnitId = Historical\_AUID) %>%  
 inner\_join(previous\_au\_attains, by = c('assessmentUnitId'))  
  
#Combine all together  
#This becomes the starting point for each csv export process  
data\_all\_AUs\_listed <- data\_current\_AU %>%  
 rbind(data\_retired\_AU) %>%  
 rbind(data\_historical\_AU)  
  
#Find AUs not in previous ATTAINS  
data\_current\_AU\_not\_listed <- data\_cat\_AUID\_added %>%  
 mutate(assessmentUnitId = AUID\_ATTNS) %>%  
 full\_join(previous\_au\_attains, by = c('assessmentUnitId')) %>%  
 filter(!assessmentUnitId %in% data\_current\_AU$assessmentUnitId) %>%  
 filter(!assessmentUnitId %in% data\_retired\_AU$assessmentUnitId) %>%  
 filter(!assessmentUnitId %in% data\_historical\_AU$assessmentUnitId)

## Summarize

After combining all AU information together, the code creates a new column that assigns a specific string value to the state’s numerical categories. The code then selects the most relevant columns for the online map and ensures each row is unique. The *final\_summary* variable is then exported as a csv.

#Summarize data  
final\_summary <- data\_all\_AUs %>%  
 mutate(overallStatus = case\_when(Use\_Category == 2 ~ 'Fully Supporting',  
 Use\_Category == 3 ~ 'Not Assessed',  
 Use\_Category == 5 ~ 'Not Supporting',  
 T ~ NA)) %>%   
 select(AUID\_ATTNS, `Waterbody Type`, Use, Use\_Category, overallStatus,   
 assessmentUnitName, locationDescription, waterSize, waterSizeUnits) %>%  
 unique()