

FNS-905 Summer Sites Report

Jared Walker

2022-07-27

Contents

1 - Set Up	1
2 - Data Import and Prep	2
2.1 - Merge Lat & Long	2
2.2 - Combine 1026 & 1027	3
2.3 - Format Dates	3
2.4 - Address Corrections	3
3 - Filter Inactive Sites	4
4 - Edit Checks	5
4.1 - Duplicate Site Numbers	5
4.2 - Missing Long and Lat	5
4.3 - Other Missing Values	5
5 - Added & Removed	6
5.1 - Edit Check	6
6 - Prepare to Save	7
7 - Generate Report	7
7.1 - Create Workbook	7
7.2 - Add Worksheets	7
7.3 - Write Data	8
7.4 - Save Sheets to Workbook	8
8 - Operational Changes	8

1 - Set Up

```
rm(list = ls(all.names = T)) #clear all objects from environment.
options(max.print=100) ##limit output.
start.time <- Sys.time()    ## Log start time.
options(scipen=999) ##prohibit use of scientific notation
gc() ## free up memory and report the memory usage.
curr_date <- as.numeric(format(Sys.Date(), "%Y%m%d")) # Format date as numeric
# date <- as.Date(date) # Use this code if date format is to be kept

library(janitor)
library(tidyverse)
```

```
library(openxlsx)
library(readr)
library(readxl)
```

2 - Data Import and Prep

2.1 - Merge Lat & Long

Note: When transitioning between school years, it may be necessary download both the new SY & the previous SY of Data1027 & Data1026.

```
## Import Previous submission, clean names, remove extraneous columns
previous_submission <- read_excel(paste0("P:/1_Internal/Reports/Summer/USDA 543a_FNS 905/2022_post COVID19/2022_FNS-905_Temporal_School_Year_2022-2023/2022_FNS-905_Temporal_School_Year_2022-2023/previous_submission.xlsx"),
  clean_names %>%
  select(-c(so_o:map_y))

## Import both sheets of the Geocode Master List, clean names, rename long and lat to match FNS-905 Temporal School Year 2022-2023
sso <- read_excel(
  "P:/1_Internal/Reports/Summer/USDA 543a_FNS 905/Geocode Master List_CURRENT.xlsx",
  sheet = 1) %>%
  clean_names %>%
  rename(soo_x = latitude,
    soo_y = longitude) %>%
  rename(state_site_id_number = site_number) %>% # Rename unique ID in order to merge()
  select(state_site_id_number, soo_x, soo_y)

sfsp <- read_excel(
  "P:/1_Internal/Reports/Summer/USDA 543a_FNS 905/Geocode Master List_CURRENT.xlsx",
  sheet = 2) %>%
  clean_names %>%
  rename(soo_x = latitude,
    soo_y = longitude) %>%
  rename(state_site_id_number = site_number) %>% # Rename unique ID in order to merge()
  select(state_site_id_number, soo_x, soo_y)

# Import Data1026 & Data1027, latin encoding, clean names, merge long and lat coordinates
data1026_sfsp <- read_csv(
  "H:/Downloads/Data1026 (70).csv", locale = locale(encoding = "Latin1")) %>%
  clean_names %>%
  select(site_name:state_sponsor_id_number, last_modified_date) %>%
  merge(sfsp, by = "state_site_id_number", all.x = T)

data1027_sso <- read_csv(
  "H:/Downloads/Data1027 - 2022-07-15T091136.119.csv", locale = locale(encoding = "Latin1")) %>%
  clean_names %>%
  select(site_name:state_sponsor_id_number, last_modified_date) %>%
  relocate(last_modified_date) %>%
  merge(sso, by = "state_site_id_number", all.x = T)

# Note: When transitioning between school years, it may be necessary download
# both the new SY & the previous SY of Data1027 & Data1026. If the previous SY
# has sites with end dates that are in the future, check to see if those sites
# are in the current year's data. If not, both schools years should be imported.
```

```
# After importing both years, combine the data. For Example:
```

```
# data1026 <- rbind(data1026 (1), data1026 (2))  
# data1027 <- rbind(data1027 (1), data1027 (2))
```

2.2 - Combine 1026 & 1027

```
# Combine Data1026 & Data1027  
all<-rbind(data1027_sso, data1026_sfsp)
```

2.3 - Format Dates

```
# Format Dates  
all$end_date_mm_dd_yy <- format(as.Date(all$end_date_mm_dd_yy, "%m/%d/%Y"))  
all$start_date_mm_dd_yy <- format(as.Date(all$start_date_mm_dd_yy, "%m/%d/%Y"))  
all$last_modified_date <- format(as.Date(all$last_modified_date, "%m/%d/%Y"))
```

2.4 - Address Corrections

```
# RICH JR HIGH/NRES
```

```
all[which(all$state_site_id_number == "24-304"), grep("site_name", colnames(all))]  
all[which(all$site_name == "RICH JR HIGH/NRES"), grep("state_site_id_number", colnames(all))]  
all[which(all$state_site_id_number == "24-304"), grep("site_address1", colnames(all))]  
all[which(all$state_site_id_number == "24-304"), grep("site_address1", colnames(all))] <- "54 E 100 S"  
all[which(all$state_site_id_number == "24-304"), grep("site_address1", colnames(all))]
```

```
# NORTH SUMMIT HIGH
```

```
all[which(all$state_site_id_number == "21-704"), grep("site_name", colnames(all))]  
all[which(all$site_name == "NORTH SUMMIT HIGH"), grep("state_site_id_number", colnames(all))]  
all[which(all$state_site_id_number == "21-704"), grep("site_address1", colnames(all))]  
all[which(all$state_site_id_number == "21-704"), grep("site_address1", colnames(all))] <- "53 S 100 E"  
all[which(all$state_site_id_number == "21-704"), grep("site_address1", colnames(all))]
```

```
# NORTH SUMMIT ELEMENTARY
```

```
all[which(all$state_site_id_number == "21-110"), grep("site_name", colnames(all))]  
all[which(all$site_name == "NORTH SUMMIT ELEMENTARY"), grep("state_site_id_number", colnames(all))]  
all[which(all$state_site_id_number == "21-110"), grep("site_address1", colnames(all))]  
all[which(all$state_site_id_number == "21-110"), grep("site_address1", colnames(all))] <- "240 S BEACON"  
all[which(all$state_site_id_number == "21-110"), grep("site_address1", colnames(all))]
```

```
# NORTH SUMMIT MIDDLE
```

```
all[which(all$state_site_id_number == "21-304"), grep("site_name", colnames(all))]  
all[which(all$site_name == "NORTH SUMMIT MIDDLE"), grep("state_site_id_number", colnames(all))]  
all[which(all$state_site_id_number == "21-304"), grep("site_address1", colnames(all))]  
all[which(all$state_site_id_number == "21-304"), grep("site_address1", colnames(all))] <- "76 S 100 E"  
all[which(all$state_site_id_number == "21-304"), grep("site_address1", colnames(all))]
```

```
# BLUE PEAK HIGH
```

```
all[which(all$state_site_id_number == "30-740"), grep("site_name", colnames(all))]  
all[which(all$site_name == "BLUE PEAK HIGH"), grep("state_site_id_number", colnames(all))]  
all[which(all$state_site_id_number == "30-740"), grep("site_address1", colnames(all))]
```

```

all[which(all$state_site_id_number == "30-740"), grep("site_address1", colnames(all))] <- "211 South Tooele"
all[which(all$state_site_id_number == "30-740"), grep("site_address1", colnames(all))]

# NAA TSIS' AAN COMMUNITY SCHOOL
all[which(all$state_site_id_number == "Q8-1"), grep("site_name", colnames(all))]
all[which(all$state_site_id_number == "Q8-1"), grep("site_address1", colnames(all))]
all[which(all$state_site_id_number == "Q8-1"), grep("site_city", colnames(all))]
all[which(all$state_site_id_number == "Q8-1"), grep("site_city", colnames(all))] <- "Tonalea"
all[which(all$state_site_id_number == "Q8-1"), grep("site_zip", colnames(all))]
all[which(all$state_site_id_number == "Q8-1"), grep("site_zip", colnames(all))] <- 86044
all[which(all$state_site_id_number == "Q8-1"), grep("site_address1", colnames(all))] <- "24 Highway 98"
all[which(all$state_site_id_number == "Q8-1"), grep("site_address1", colnames(all))]

# GROUSE CREEK SCHOOL
all[which(all$site_name == "GROUSE CREEK SCHOOL"), grep("state_site_id_number", colnames(all))]
all[which(all$state_site_id_number == "03-140"), grep("site_name", colnames(all))]
all[which(all$state_site_id_number == "03-140"), grep("site_address1", colnames(all))]
all[which(all$state_site_id_number == "03-140"), grep("site_address1", colnames(all))] <- "76785 W 11900"
all[which(all$state_site_id_number == "03-140"), grep("site_address1", colnames(all))]

# BELKNAP ELEMENTARY
all[which(all$state_site_id_number == "02-104"), grep("site_name", colnames(all))]
all[which(all$state_site_id_number == "02-104"), grep("site_address1", colnames(all))]
all[which(all$state_site_id_number == "02-104"), grep("site_address1", colnames(all))] <- "510 North 650"
all[which(all$state_site_id_number == "02-104"), grep("site_address1", colnames(all))]

# BEAVER HIGH
all[which(all$state_site_id_number == "02-704"), grep("site_name", colnames(all))]
all[which(all$state_site_id_number == "02-704"), grep("site_address1", colnames(all))]
all[which(all$state_site_id_number == "02-704"), grep("site_address1", colnames(all))] <- "195 E CENTER"
all[which(all$state_site_id_number == "02-704"), grep("site_address1", colnames(all))]

# WELLINGTON ELEMENTARY
all[which(all$state_site_id_number == "05-154"), grep("site_name", colnames(all))]
all[which(all$state_site_id_number == "05-154"), grep("site_address1", colnames(all))]
all[which(all$state_site_id_number == "05-154"), grep("site_address1", colnames(all))] <- "250 W 200 N"
all[which(all$state_site_id_number == "05-154"), grep("site_address1", colnames(all))]

# BLUFF ELEMENTARY
all[which(all$site_name == "BLUFF ELEMENTARY"), grep("state_site_id_number", colnames(all))]
all[which(all$state_site_id_number == "25-108"), grep("site_name", colnames(all))]
all[which(all$state_site_id_number == "25-108"), grep("site_address1", colnames(all))]
all[which(all$state_site_id_number == "25-108"), grep("site_address1", colnames(all))] <- "Old Main High"
all[which(all$state_site_id_number == "25-108"), grep("site_address1", colnames(all))]

```

3 - Filter Inactive Sites

```

active <- all %>%
  relocate(state_site_id_number, .before = state_sponsor_id_number) %>%
  filter(end_date_mm_dd_yy >= Sys.Date()+1) %>%
  filter(start_date_mm_dd_yy <= Sys.Date()+30)

```

```
refresh_active <- active
refresh_all <- all

nrow(all)
nrow(active)
```

4 - Edit Checks

4.1 - Duplicate Site Numbers

```
active$b<-duplicated(active$state_site_id_number, fromLast = F)
active$a<-duplicated(active$state_site_id_number, fromLast = T)
active$ab<-paste0(active$a, active$b)

all_dups<-active[active$ab != "FALSEFALSE", ]
active<-active[active$ab == "FALSEFALSE", ]

active <- active %>%
  select(-c(a, b, ab))

nrow(all_dups) # Number of duplicates
```

4.2 - Missing Long and Lat

Write to a workbook list of sites with missing long and lat coordinates, get coordinates from Google, add to Geocode Master List_CURRENT.

```
# If NAs are present, save to file to add to Geocode Master List
NAs2<-active[is.na(active$soo_x),]

NAs2<-NAs2 %>%
  select(site_program, sponsoring_organization, state_sponsor_id_number,
         site_name, state_site_id_number, site_address1, site_city,
         site_state, site_zip) %>%
  write.xlsx("H:/Documents/NAs2.xlsx", overwrite=T)

nrow(NAs2)
```

4.3 - Other Missing Values

Saves list of sites to workbook with NAs in other required fields.

```
# Identify NAs in the following columns

NAs1 <- active %>%
  filter(is.na(site_address1) | is.na(site_zip) | is.na(site_city)
         | is.na(site_state) | is.na(sponsoring_organization)
         | is.na(state_site_id_number) | is.na(state_sponsor_id_number)) %>%
  write.xlsx("H:/Documents/NAs1.xlsx", overwrite=T)

nrow(NAs1)

# NA Locations (used for correcting NAs with code)
which(is.na(active$site_address1))
```

```

which(is.na(active$site_zip))
which(is.na(active$site_state))
which(is.na(active$sponsoring_organization))
which(is.na(active$state_site_id_number))
which(is.na(active$state_sponsor_id_number))
which(is.na(active$soo_x))
which(is.na(active$soo_y))

```

5 - Added & Removed

Combine current and previous submissions and look for unique site numbers. Sites added are indicated by unique site numbers in the current submission, sites removed are indicated by unique site numbers in the previous submission. Filter unique site #'s by submission and save as separate data objects.

```

current_submission <- active %>%
  select(-last_modified_date) %>%
  mutate(submission = "current")

previous_submission <- active %>%
  select(-last_modified_date) %>%
  mutate(submission = "previous")

# Combine
added_removed<-rbind(previous_submission,current_submission)

# Identify unique values
added_removed$b<-duplicated(added_removed$state_site_id_number, fromLast = F)
added_removed$a<-duplicated(added_removed$state_site_id_number, fromLast = T)
added_removed$ab<-paste0(added_removed$a, added_removed$b)

unique<-added_removed[added_removed$ab == "FALSEFALSE", ]

nrow(unique)

added <- unique %>%
  select(site_program, sponsoring_organization, site_name,
         start_date_mm_dd_yy, end_date_mm_dd_yy, submission) %>%
  filter(submission == "current") %>%
  select(-submission)

removed <- unique %>%
  select(site_program, sponsoring_organization, site_name,
         start_date_mm_dd_yy, end_date_mm_dd_yy, submission) %>%
  filter(submission == "previous") %>%
  select(-submission)

```

5.1 - Edit Check

	<i>the # of sites in the previous submission</i>
+	<i>the # of sites added</i>
-	<i>the # of sites removed</i>
=	<i>the # of sites in the current submission</i>

```
# Make sure the number of sites corresponds with the number of sites added and removed
nrow(added)
nrow(removed)

nrow(previous_submission)+nrow(added)-nrow(removed)
nrow(current_submission)
```

6 - Prepare to Save

Add spaces to site numbers to prevent data from being converted to dates in Excel. Reformat column headers for viewing.

```
active <- refresh_active
all <- refresh_all

# Add spaces to site numbers to prevent them from turning into dates
all <- all %>%
  mutate(state_site_id_number = paste0(" ", state_site_id_number, " ")) %>%
  select(-last_modified_date)

active <- active %>%
  mutate(state_site_id_number = paste0(" ", state_site_id_number, " ")) %>%
  select(-last_modified_date)

# Column Names for Added / Removed Sheets
colnames(added) <- stringr::str_to_title(gsub("_", " ", colnames(added)))
colnames(removed) <- stringr::str_to_title(gsub("_", " ", colnames(removed)))

# Get the number of rows of each data frame and save it as a vector.
nrow_all<-as.vector(nrow(all))
nrow_active<-as.vector(nrow(active))
nrow_added<-as.vector(nrow(added))
nrow_removed<-as.vector(nrow(removed))
```

7 - Generate Report

7.1 - Create Workbook

```
# Create workbook
wb <- createWorkbook()
```

7.2 - Add Worksheets

```
# Add Worksheets to Workbook and Name
addWorksheet(wb, paste0("Active Sites, (",nrow_active,")"))
addWorksheet(wb, paste0("All Sites, (",nrow_all,")"))
```

```
addWorksheet(wb, paste0("Added,  (",nrow_added,""))
addWorksheet(wb, paste0("Removed,  (",nrow_removed,""))
```

7.3 - Write Data

```
# Write Data to Worksheets
writeData(wb, sheet = paste0("Active Sites,  (",nrow_active,""), x = active)
writeData(wb, sheet = paste0("All Sites,  (",nrow_all,""), x = all)
writeData(wb, sheet = paste0("Added,  (",nrow_added,""), x = added)
writeData(wb, sheet = paste0("Removed,  (",nrow_removed,""), x = removed)
```

7.4 - Save Sheets to Workbook

```
# Save
saveWorkbook(wb, paste0("H:/Documents/USDA FNS 905 Submission ", curr_date, ".xlsx"), overwrite = T)
```

8 - Operational Changes

```
active <- refresh_active

# Subset data
modifications <- active %>%
  filter(last_modified_date >= Sys.Date()-7) %>%
  mutate(state_site_id_number = paste0(" ", state_site_id_number, " "))

# Save to P Drive
write.csv(modifications, paste0("P:/1_Internal/Contracts, MOUs, State Agreements, Purchase Requests (no

end.time <- Sys.time() # Stop Timer
duration <- end.time - start.time
duration
```