Bin Review Data Management

This file uses the bin peer review data to make updates to the existing labels, create new bins and labels, and remove nonexistent bins and labels.

Data Formatting

1. Initialization

```
# Init
if(!requireNamespace("devtools")) install.packages("devtools")
devtools::install_github("dkahle/ggmap", ref = "tidyup", force=TRUE)
library(tidyverse)
library(geojsonio)
library(sp)
library(ggmap)
library(sf)
library(geojsonsf)

# This is an API Key to use Google Maps in R. You can get one yourself from
# Google or ask me for mine. I didn't include it in the file because it
# costs money to use and I don't know who will see this file in the future
# - Oliver
ggmap::register_google("...")
```

2. Import GeoJSON

```
# Parse GeoJSON
fileName = paste0(getwd(), '/geojson/review-combined.json')
geojsonText = readChar(fileName, file.info(fileName)$size)
sf <- geojson_sf(geojsonText)
```

3. Cleaning

a. Remove duplicate bins based on ID

```
# Original n features
nrow(sf)
```

```
## [1] 1534
```

```
unique_bin_indices <- split(seq_along(sf$name), sf$name) %>% map_int(~ .[1])
sf_new <- sf[unique_bin_indices, ]
# New n features
nrow(sf_new)
```

```
## [1] 556
```

b. Tag for nonexistent bin - 1084FL - 1085FR

```
non_ex_bin_indices <- which(sf_new$name %in% c('1084FL', '1085FR'))
sf_new <- sf_new[-non_ex_bin_indices, ]
# New n features
nrow(sf_new)
```

```
## [1] 554
```

c. Incorrect Bin Labels

```
# New n features
nrow(sf_new)
```

```
## [1] 554
```

```
# Zone 2
  # XL/XR -> IL/IR
sf_new[which(sf_new$name == '2007XR'), 'name'] <- '2xxxIR'
rm_1_df <- sf_new %>% filter(name == '2xxxIR')
rm_1_df[1, 'name'] <- '2xxxIL'
sf_new <- rbind(sf_new, rm_1_df)

sf_new[which(sf_new$name == '2010XL'), 'name'] <- '2010IL'
sf_new[which(sf_new$name == '2011XR'), 'name'] <- '2011IR'

sf_new[which(sf_new$name == '2009XL'), 'name'] <- '2009IL'
sf_new[which(sf_new$name == '2008XR'), 'name'] <- '2008IR'

sf_new[which(sf_new$name == '2003XL'), 'name'] <- '2003IL'
sf_new[which(sf_new$name == '2004XR'), 'name'] <- '2004IR'

  # F cluster -> D cluster
sf_new[which(sf_new$name == '2005FR'), 'name'] <- '2005DR'
sf_new[which(sf_new$name == '2006FL'), 'name'] <- '2006DL'
f_to_d_1 <- sf_new %>% filter(name == '2006DL')
f_to_d_1[1, 'name'] <- '2xxxDC'

sf_new[which(sf_new$name == '2066FR'), 'name'] <- '2066DR'
sf_new[which(sf_new$name == '2067FL'), 'name'] <- '2067DL'
f_to_d_2 <- sf_new %>% filter(name == '2067DL')
f_to_d_1[1, 'name'] <- '2xxxDC'

sf_new <- rbind(sf_new, f_to_d_1, f_to_d_2)

# Zone 3
sf_new[which(sf_new$name == '3027CC'), 'name'] <- '3027DC'
sf_new[which(sf_new$name == '3026CL'), 'name'] <- '3926DL'
sf_new[which(sf_new$name == '3025CR'), 'name'] <- '3025DR'

# New n features
nrow(sf_new)
```

```
## [1] 557
```

```r
# Extant Clusters
unique(substr(sf_new$name, 5, 5))
```

```
## [1] "F" "D" "X" "A" "I" "L" "H" "B"
```

d. No tag for existent bin

```r
# Split by Zone
zone1_bins <- sf_new %>% filter(substr(name, 1, 1) == "1")
zone2_bins <- sf_new %>% filter(substr(name, 1, 1) == "2")
zone3_bins <- sf_new %>% filter(substr(name, 1, 1) == "3")
# Get Lat
zone1_bins_lat <- zone1_bins$geometry %>% map_dbl(~ .[2])
zone2_bins_lat <- zone2_bins$geometry %>% map_dbl(~ .[2])
zone3_bins_lat <- zone3_bins$geometry %>% map_dbl(~ .[2])
# Generate Bin ID xxx nums
zone1_bins$bin_num[order(zone1_bins_lat)] <- seq_along(zone1_bins_lat)
zone2_bins$bin_num[order(zone2_bins_lat)] <- seq_along(zone2_bins_lat)
zone3_bins$bin_num[order(zone3_bins_lat)] <- seq_along(zone3_bins_lat)
zone1_bins$bin_num <- formatC(zone1_bins$bin_num, width=3, flag="0")
zone2_bins$bin_num <- formatC(zone2_bins$bin_num, width=3, flag="0")
zone3_bins$bin_num <- formatC(zone3_bins$bin_num, width=3, flag="0")
# Get Streams
zone1_bins$stream_char <- substr(zone1_bins$name, 6, 6)
zone2_bins$stream_char <- substr(zone2_bins$name, 6, 6)
zone3_bins$stream_char <- substr(zone3_bins$name, 6, 6)
zone1_bins$stream <- zone1_bins$stream_char %>% recode(R="Recycle", C="Compost", L="Landfill")
zone2_bins$stream <- zone2_bins$stream_char %>% recode(R="Recycle", C="Compost", L="Landfill")
zone3_bins$stream <- zone3_bins$stream_char %>% recode(R="Recycle", C="Compost", L="Landfill")
# Generate Full Bin IDs
zone1_bins <- zone1_bins %>% mutate(new_name = paste0('1', bin_num, substr(name, 5, 6)))
zone2_bins <- zone2_bins %>% mutate(new_name = paste0('2', bin_num, substr(name, 5, 6)))
zone3_bins <- zone3_bins %>% mutate(new_name = paste0('3', bin_num, substr(name, 5, 6)))
# Recombine
sf_old <- sf_new
sf_new <- rbind(zone1_bins, zone2_bins, zone3_bins)
# Extant Clusters
unique(substr(sf_new$name, 5, 5))
```

```
## [1] "F" "D" "X" "A" "I" "L" "H" "B"
```

e. Reassigning X clusters

```r
# Zone 1
  # Behind Music School
    # N
sf_new[which(sf_new$new_name == '1185XR'), 'new_name'] <- '1185NR'
sf_new[which(sf_new$new_name == '1186XL'), 'new_name'] <- '1186NL'

sf_new[which(sf_new$new_name == '1189XR'), 'new_name'] <- '1189NR'
sf_new[which(sf_new$new_name == '1190XL'), 'new_name'] <- '1190NL'
```

```
sf_new[which(sf_new$new_name == '1194XR'), 'new_name'] <- '1194NR'
sf_new[which(sf_new$new_name == '1195XL'), 'new_name'] <- '1195NL'
    # M
sf_new[which(sf_new$new_name == '1196XL'), 'new_name'] <- '1196MR'
sf_new[which(sf_new$new_name == '1199XL'), 'new_name'] <- '1199ML'
  # YRL
    # L
sf_new[which(sf_new$new_name == '1160XL'), 'new_name'] <- '1160LL'
sf_new[which(sf_new$new_name == '1161XL'), 'new_name'] <- '1161LL'
sf_new[which(sf_new$new_name == '1162XL'), 'new_name'] <- '1162LL'
sf_new[which(sf_new$new_name == '1163XL'), 'new_name'] <- '1163LL'
  # Behind Powell
    # L
sf_new[which(sf_new$new_name == '1006XL'), 'new_name'] <- '1006LL'
sf_new[which(sf_new$new_name == '1007XL'), 'new_name'] <- '1007LL'
sf_new[which(sf_new$new_name == '1008XL'), 'new_name'] <- '1008LL'
sf_new[which(sf_new$new_name == '1009XL'), 'new_name'] <- '1009LL'
sf_new[which(sf_new$new_name == '1012XL'), 'new_name'] <- '1012LL'
sf_new[which(sf_new$new_name == '1013XL'), 'new_name'] <- '1013LL'
sf_new[which(sf_new$new_name == '1014XL'), 'new_name'] <- '1014LL'
sf_new[which(sf_new$new_name == '1018XL'), 'new_name'] <- '1018LL'
sf_new[which(sf_new$new_name == '1020XL'), 'new_name'] <- '1020LL'
  # Road behind Law
    # N
sf_new[which(sf_new$new_name == '1109XR'), 'new_name'] <- '1109NR'
sf_new[which(sf_new$new_name == '1104XL'), 'new_name'] <- '1104NL'

sf_new[which(sf_new$new_name == '1091XR'), 'new_name'] <- '1091NR'
sf_new[which(sf_new$new_name == '1090XL'), 'new_name'] <- '1090NL'

# Zone 2
  # Bruin Plaza
    # P
sf_new[which(sf_new$new_name == '2054XC'), 'new_name'] <- '2054PC'
sf_new[which(sf_new$new_name == '2055XR'), 'new_name'] <- '2055PR'
sf_new[which(sf_new$new_name == '2056XL'), 'new_name'] <- '2056PL'
  # Behind IM Field
    # N
sf_new[which(sf_new$new_name == '2140XR'), 'new_name'] <- '2140NR'
sf_new[which(sf_new$new_name == '2141XL'), 'new_name'] <- '2141NL'
sf_new[which(sf_new$new_name == '2142XL'), 'new_name'] <- '2142NL'
sf_new[which(sf_new$new_name == '2143XR'), 'new_name'] <- '2143NR'
sf_new[which(sf_new$new_name == '2145XL'), 'new_name'] <- '2145NL'
sf_new[which(sf_new$new_name == '2148XR'), 'new_name'] <- '2148NR'
  # Lab School
    # M
sf_new[which(sf_new$new_name == '2160XL'), 'new_name'] <- '2160ML'
sf_new[which(sf_new$new_name == '2161XL'), 'new_name'] <- '2161ML'

# Zone 3
  # Kinsey
    # M
sf_new[which(sf_new$new_name == '3174XL'), 'new_name'] <- '3174ML'
```

```
  # Med
    # N
sf_new[which(sf_new$new_name == '3035XR'), 'new_name'] <- '3035NR'
sf_new[which(sf_new$new_name == '3036XL'), 'new_name'] <- '3036NL'
    # M
sf_new[which(sf_new$new_name == '3041XL'), 'new_name'] <- '3041ML'
sf_new[which(sf_new$new_name == '3081XL'), 'new_name'] <- '3081ML'
sf_new[which(sf_new$new_name == '3083XL'), 'new_name'] <- '3083ML'
sf_new[which(sf_new$new_name == '3084XL'), 'new_name'] <- '3084ML'
```

f. Small Landfill
It turns out the all landfills in triple clusters are small, so we will make the `A` and `B` cluster contain `C`, `R`, `S` streams instead of `C`, `R`, `L`, in lieu of the small landfills.

```
small_landfills <- (sf_new %>% filter(substr(new_name, 5, 6) %in% c('BL', 'AL')))$new_name
for(bin_name in small_landfills) {
  sf_new[which(sf_new$new_name == bin_name), 'stream_char'] <- 'S'
  sf_new[which(sf_new$new_name == bin_name), 'new_name'] <- paste0(substr(bin_name, 1, 5), 'S')
}
```

Export

GeoJSON

Format

```
exportable_sf <- sf_new %>% transmute(Name=new_name, Stream=stream, 'Bin Number'=bin_num, 'Stream Chara
```

Write File

```
st_write(exportable_sf, dsn = paste0(getwd(), "/geojson/peer_reviewed_labels6.json"), driver = "GeoJSON
```

CSV

Extra IDs

1. M for every L and L for every M because they may be confused
2. Extra bins in each zone in case of new bins

```
# Ms & Ls
m_bins <- exportable_sf %>% filter(substr(Name, 5, 5) == 'M')
fake_l_bins <- m_bins %>% mutate(Name=paste0(substr(Name, 1, 4), 'L', substr(Name, 6, 6)))
fake_l_bins <- data.frame(Name=fake_l_bins$Name)

l_bins <- exportable_sf %>% filter(substr(Name, 5, 5) == 'L')
fake_m_bins <- l_bins %>% mutate(Name=paste0(substr(Name, 1, 4), 'M', substr(Name, 6, 6)))
fake_m_bins <- data.frame(Name=fake_m_bins$Name)
```

```
# Extra
unique_bin_chars <- unique(substr(exportable_sf$Name, 5, 6))
z1_max <- max(as.integer(exportable_sf$`Bin Number`[exportable_sf$Zone == '1']))
z2_max <- max(as.integer(exportable_sf$`Bin Number`[exportable_sf$Zone == '2']))
z3_max <- max(as.integer(exportable_sf$`Bin Number`[exportable_sf$Zone == '3']))
z1_extra <- character()
z2_extra <- character()
z3_extra <- character()
for(n in seq_along(unique_bin_chars)) {
  z1_extra <- c(z1_extra, paste0('1', as.character(z1_max + n), unique_bin_chars[n]))
}
for(n in seq_along(unique_bin_chars)) {
  z2_extra <- c(z2_extra, paste0('2', as.character(z1_max + n), unique_bin_chars[n]))
}
for(n in seq_along(unique_bin_chars)) {
  z3_extra <- c(z3_extra, paste0('3', as.character(z1_max + n), unique_bin_chars[n]))
}
z_extra_df <- data.frame(Name=c(z1_extra, z2_extra, z3_extra))

# I <3 REA
csv_bins_w_extra <- rbind(data.frame(Name=exportable_sf$Name), fake_l_bins, fake_m_bins, z_extra_df, da
```

Write File

```
write_csv(csv_bins_w_extra, paste0(getwd(), "/label-order/FinalLabels2.csv"))
```

Mapping

1. Formatting for mapping

```
ilat <- sf_new$geometry %>% map_dbl(~ .[2])
ilong <- sf_new$geometry %>% map_dbl(~ .[1])
sf_new <- sf_new %>% mutate(lat=ilat, long=ilong)

x_clust <- sf_new %>% filter(substr(new_name, 5, 5) == 'X')
x_clust1 <- x_clust %>% filter(substr(new_name, 1,1) == '1')
x_clust2 <- x_clust %>% filter(substr(new_name, 1,1) == '2')
x_clust3 <- x_clust %>% filter(substr(new_name, 1,1) == '3')
```

```
# Base Terrain from Google
p <- ggmap(get_googlemap(center = c(Longitude = -118.442, Latitude = 34.06551),
                         zoom = 17, scale = 2,
                         maptype ='satellite',
                         color = 'color'))

p + geom_point(aes(x=long, y=lat, color=stream), data = x_clust3, size=2.5)
x_clust3 %>% select(new_name, stream, lat, long) %>% arrange(lat)
```