

# Little Free Library Analysis

Kaleb Crans

2023-04-17

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.1    v purrr   1.0.1
## v tibble  3.1.8    v dplyr   1.1.0
## v tidyr   1.3.0    v stringr 1.5.0
## v readr   2.1.4    v forcats 1.0.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(readr)
library(sf)

## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE

library(spData)

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`

rm(list = ls())

typeof(read_csv("libraries.csv"))

## Rows: 65598 Columns: 19
## -- Column specification -----
## Delimiter: ","
## chr  (8): Name, Street__c, City__c, State_Province_Region__c, Postal_Zip_Cod...
## dbl  (7): id, Count_of_Primary_Stewards__c, Latitude_MapAnything__c, Longitu...
## lgl  (2): Traveling_Library__c, Duplicate_Charter_Number__c
## date (2): First_Map_Date__c, Map_Date__c
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## [1] "list"

if (file.exists("lfl.RData")) {
  load("lfl.RData")
} else {
  libraries = as_tibble(read_csv("libraries.csv"))
  save(libraries, file = "lfl.RData")
}
```

## Data Cleaning and Preparation

```
lapply(libraries, typeof)
```

```
## $id
## [1] "double"
##
## $Name
## [1] "character"
##
## $Street__c
## [1] "character"
##
## $City__c
## [1] "character"
##
## $State_Province_Region__c
## [1] "character"
##
## $Postal_Zip_Code__c
## [1] "character"
##
## $Country__c
## [1] "character"
##
## $Traveling_Library__c
## [1] "logical"
##
## $Official_Charter_Number__c
## [1] "character"
##
## $First_Map_Date__c
## [1] "double"
##
## $Map_Me__c
## [1] "character"
##
## $Map_Date__c
## [1] "double"
##
## $Duplicate_Charter_Number__c
## [1] "logical"
##
## $Count_of_Primary_Stewards__c
## [1] "double"
##
## $Latitude_MapAnything__c
## [1] "double"
##
## $Longitude_MapAnything__c
## [1] "double"
##
## $Library_Geolocation__Latitude__s
## [1] "double"
```

```
##
## $Library_Geolocation__Longitude__s
## [1] "double"
##
## $check_in_count
## [1] "double"
```

Map\_Me\_\_c needs to be transformed into a logical variable, as “Taken Down Temporarily” and “Mapped” are the only two categories.

```
libraries <- libraries %>% mutate(Map_Me__c = Map_Me__c == "Mapped")
```

A quick look at check\_in\_counts shows that this feature is rarely used considering how many times a given library is actually visited:

```
print("Max:")
```

```
## [1] "Max:"
```

```
max(libraries$check_in_count)
```

```
## [1] 47
```

```
print("Summary Stats:")
```

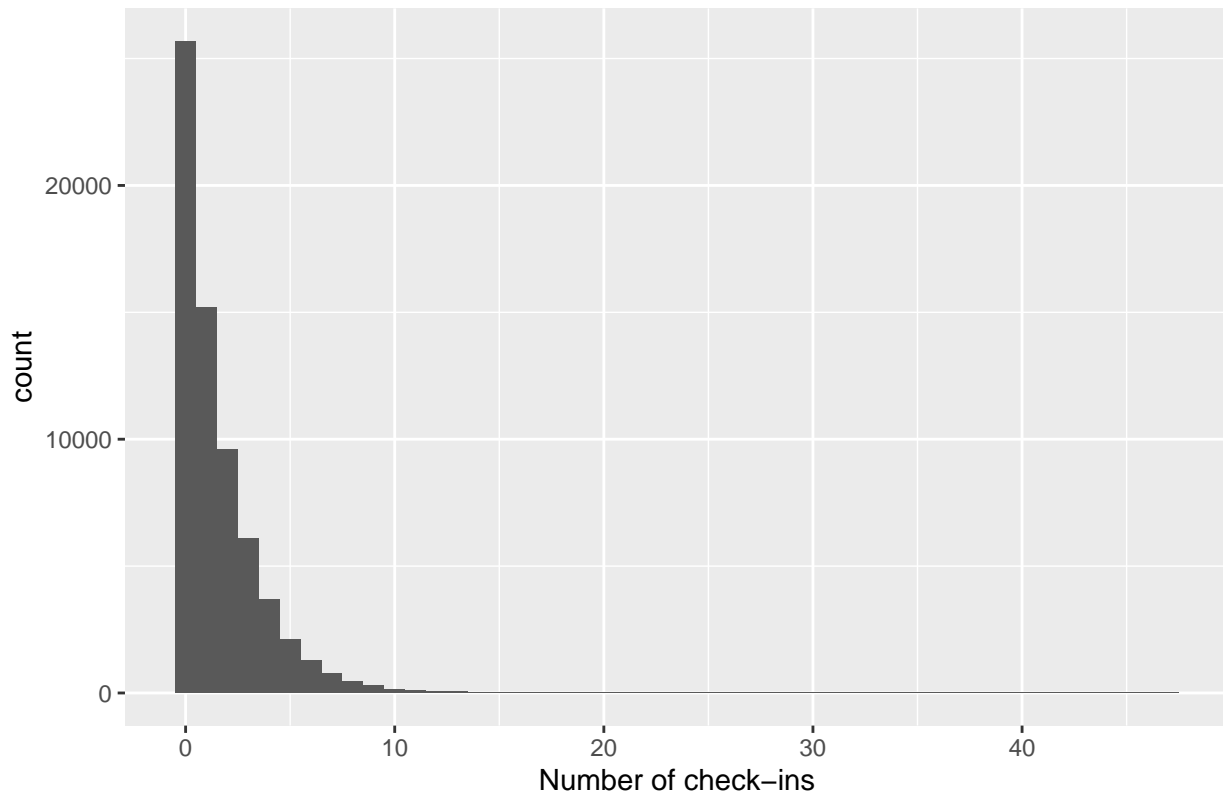
```
## [1] "Summary Stats:"
```

```
summary(libraries$check_in_count)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.000   0.000   1.000   1.567   2.000  47.000
```

```
check_in_dist <- libraries %>% count(check_in_count)
ggplot(data = libraries, aes(x = check_in_count)) +
  ggtitle("Check-in count distribution") +
  xlab("Number of check-ins") +
  geom_histogram(binwidth = 1)
```

Check-in count distribution



```
# Clean up alternative names for the same country:
libraries <- libraries %>% mutate(Country__c = replace(Country__c, Country__c %in% c("USA", "US", "U.S.", "United States"), "United States"))

# Ranking of countries by number of little free libraries:
libraries %>% count(Country__c) %>% arrange(desc(n))
```

##	Country__c	n
## 1	United States	59816
## 2	Canada	2871
## 3	<NA>	633
## 4	Italy	367
## 5	United Kingdom	278
## 6	Australia	229
## 7	France	202
## 8	Philippines	123
## 9	Netherlands	108
## 10	Belgium	90
## 11	Brazil	58
## 12	Malaysia	45
## 13	Armenia	41
## 14	New Zealand	39
## 15	Lithuania	37
## 16	Germany	34
## 17	Ukraine	33
## 18	Mexico	32
## 19	India	28
## 20	Japan	28

## 21	Norway	26
## 22	Denmark	23
## 23	Nigeria	23
## 24	China	22
## 25	Slovenia	21
## 26	Ireland	19
## 27	Finland	17
## 28	Spain	17
## 29	Romania	16
## 30	Colombia	15
## 31	Portugal	15
## 32	Switzerland	15
## 33	Croatia	13
## 34	Sweden	12
## 35	Lebanon	11
## 36	U.S. Virgin Islands	10
## 37	Israel	9
## 38	Luxembourg	9
## 39	Saint Kitts and Nevis	9
## 40	South Africa	9
## 41	Belize	7
## 42	Argentina	6
## 43	Czech Republic	6
## 44	Dominican Republic	6
## 45	Puerto Rico	6
## 46	United Arab Emirates	6
## 47	Honduras	5
## 48	Indonesia	5
## 49	Pakistan	5
## 50	Cabo Verde	4
## 51	Chile	4
## 52	Guyana	4
## 53	PAKISTAN	4
## 54	Slovakia	4
## 55	Taiwan	4
## 56	Barbados	3
## 57	Bermuda	3
## 58	Ghana	3
## 59	Guam	3
## 60	Hungary	3
## 61	Kyrgyzstan	3
## 62	Poland	3
## 63	Russia	3
## 64	Serbia	3
## 65	Trinidad and Tobago	3
## 66	Albania	2
## 67	Austria	2
## 68	Azerbaijan	2
## 69	Bahrain	2
## 70	Belarus	2
## 71	Bulgaria	2
## 72	Cyprus	2
## 73	Ecuador	2
## 74	Egypt	2

## 75	Greece	2
## 76	Haiti	2
## 77	Jamaica	2
## 78	Malta	2
## 79	Marshall Islands	2
## 80	Micronesia	2
## 81	Nicaragua	2
## 82	Oman	2
## 83	Qatar	2
## 84	Rwanda	2
## 85	Saint Vincent and the Grenadines	2
## 86	South Korea	2
## 87	Turkey	2
## 88	Venezuela	2
## 89	Vietnam	2
## 90	AUSTRALIA	1
## 91	Afghanistan	1
## 92	Bahamas	1
## 93	Bangladesh	1
## 94	Brunei Darussalam	1
## 95	Cambodia	1
## 96	Caribbean Netherlands	1
## 97	Cayman Islands	1
## 98	Costa Rica	1
## 99	El Salvador	1
## 100	FRANCE	1
## 101	Georgia	1
## 102	Grenada	1
## 103	Guatemala	1
## 104	Hong Kong	1
## 105	Hong Kong SAR	1
## 106	ITALY	1
## 107	Iceland	1
## 108	Iraq	1
## 109	Korea	1
## 110	Korea, Republic Of	1
## 111	Kosovo	1
## 112	Laos	1
## 113	Latvia	1
## 114	Malawi	1
## 115	Mauritius	1
## 116	Mongolia	1
## 117	North Korea	1
## 118	Panama	1
## 119	Peru	1
## 120	Republic of Kosovo	1
## 121	Saudi Arabia	1
## 122	Serbia and Montenegro	1
## 123	Singapore	1
## 124	Sudan	1
## 125	Tanzania	1
## 126	Thailand	1
## 127	Trinidad And Tobago	1
## 128	Uganda	1

```
## 129                Zambia      1
## 130                ghana       1
```

The U.S. by and large has the greatest amount of little free libraries (with alternate spellings outpacing many countries even). Canada is the only country with a somewhat comparable amount, specifically if you adjust for population size.

```
# US population
us_pop <- 331900000

# Canada population
can_pop <- 38250000

us_count <- libraries %>% filter(Country__c == "United States") %>% nrow

can_count <- libraries %>% filter(Country__c == "Canada") %>% nrow
```

So the per capita number of little free libraries in the US is:

```
format(us_count/us_pop, scientific = FALSE)
```

```
## [1] "0.000180223"
```

And in Canada is:

```
format(can_count/can_pop, scientific = FALSE)
```

```
## [1] "0.00007505882"
```

So we can conclude that Little Free Libraries are a predominately American phenomenon. For the purposes of further analysis, let's exclude all data points not in the US:

```
libraries <- libraries %>% filter(Country__c == "United States")
```

## Analysis by state

How about the distribution by state?

```
length(unique(libraries$State_Province_Region__c))
```

```
## [1] 120
```

But there's only 50 states! So we need to do some data cleaning first.

```
libraries %>% count(State_Province_Region__c) %>% arrange(desc(n))
```

```
##      State_Province_Region__c      n
## 1                CA 7278
## 2                TX 3352
## 3                MN 2747
## 4                WA 2588
## 5                WI 2572
## 6                IL 2547
## 7                MI 2400
## 8                FL 2309
## 9                PA 2200
## 10               NY 2123
## 11               OH 2069
## 12               NC 1948
## 13               CO 1863
```

## 14	VA 1795
## 15	MA 1555
## 16	OR 1471
## 17	GA 1435
## 18	MD 1335
## 19	IN 1292
## 20	AZ 1174
## 21	NJ 1119
## 22	IA 1003
## 23	TN 962
## 24	MO 845
## 25	SC 807
## 26	CT 700
## 27	UT 577
## 28	AL 539
## 29	LA 536
## 30	KY 485
## 31	KS 483
## 32	NE 406
## 33	ME 396
## 34	NM 372
## 35	OK 369
## 36	ID 357
## 37	AR 325
## 38	NH 325
## 39	MT 311
## 40	MS 293
## 41	DC 273
## 42	WV 242
## 43	NV 226
## 44	RI 224
## 45	ND 213
## 46	VT 203
## 47	DE 202
## 48	SD 194
## 49	AK 186
## 50	<NA> 124
## 51	WY 121
## 52	HI 75
## 53	Wi 29
## 54	Ca 25
## 55	ca 19
## 56	Wa 18
## 57	wi 14
## 58	Mn 11
## 59	Tx 11
## 60	Il 10
## 61	mn 10
## 62	wa 9
## 63	Ga 7
## 64	Co 6
## 65	Fl 6
## 66	Va 6
## 67	ga 6



## 68	Pa	4
## 69	Al	3
## 70	California	3
## 71	Mi	3
## 72	Michigan	3
## 73	Oh	3
## 74	Or	3
## 75	Pennsylvania	3
## 76	Illinois	2
## 77	Ks	2
## 78	Md	2
## 79	Minnesota	2
## 80	New York	2
## 81	Utah	2
## 82	ak	2
## 83	il	2
## 84	la	2
## 85	nc	2
## 86	or	2
## 87	sc	2
## 88	tx	2
## 89	Az	1
## 90	Delaware	1
## 91	Ia	1
## 92	Idaho	1
## 93	In	1
## 94	Indiana	1
## 95	Iowa	1
## 96	Ky	1
## 97	La	1
## 98	Ma	1
## 99	Missouri	1
## 100	Mo	1
## 101	ON	1
## 102	Ohio	1
## 103	Ok	1
## 104	Texas	1
## 105	Tn	1
## 106	USA	1
## 107	Ut	1
## 108	X	1
## 109	co	1
## 110	ct	1
## 111	ks	1
## 112	ma	1
## 113	md	1
## 114	mi	1
## 115	mo	1
## 116	ny	1
## 117	pa	1
## 118	tn	1
## 119	va	1
## 120	wv	1

There's a bunch of different spelling variations. Let's instead take the actual coordinates and then find the states ourselves. One point of interest in the dataset to note that there are two sets of coordinates for each row: Latitude\_MapAnything\_\_c and Longitude\_MapAnything\_\_c vs Library\_Geolocation\_\_Latitude\_\_s and Library\_Geolocation\_\_Longitude\_\_s.

We can make a dataframe with the differences as separate columns, and print out the mean difference in latitude and longitude respectively:

```
differences <- libraries %>% mutate(dif_lat = (abs(Latitude_MapAnything__c) - abs(Library_Geolocation__Latitude__s))
c(mean(differences$dif_lat), mean(differences$dif_long))
```

```
## [1] -1.473463 -3.669450
```

Unfortunately due to the curvature of the earth these values don't mean too much as-is.

To explore further, let's take one example with a latitude difference of 16.6 and then plug the coordinates into Google maps. We get two different locations, one in Lake Park Iowa and the other in Lake Park Florida:

```
libraries %>% filter(Latitude_MapAnything__c == 26.79489)
```

```
##      id      Name  Street__c  City__c State_Province_Region__c
## 1 41210 LIB-000004180 307 4th St. Lake Park                      FL
##   Postal_Zip_Code__c  Country__c Traveling_Library__c
## 1                33403 United States                FALSE
##   Official_Charter_Number__c First_Map_Date__c Map_Me__c Map_Date__c
## 1                11455      2014-10-03      TRUE  2019-01-22
##   Duplicate_Charter_Number__c Count_of_Primary_Stewards__c
## 1                FALSE                                1
##   Latitude_MapAnything__c Longitude_MapAnything__c
## 1                26.79489                -80.06038
##   Library_Geolocation__Latitude__s Library_Geolocation__Longitude__s
## 1                43.45611                -95.31709
##   check_in_count
## 1                0
```

The MapAnything location:

The geolocation:

This library is actually displayed incorrectly in Iowa on the official webapp.

Another example is a location with a 101 degree difference in longitude.

```
libraries %>% filter(Latitude_MapAnything__c == 37.33889)
```

```
##      id      Name                                Street__c
## 1 75881 LIB-000084125 North Campus Building, Perandori Dushan, Mitrovicë
##   City__c State_Province_Region__c Postal_Zip_Code__c  Country__c
## 1 Mitrovica                        KS                40000 United States
##   Traveling_Library__c Official_Charter_Number__c First_Map_Date__c Map_Me__c
## 1                FALSE                85849      2022-10-13      TRUE
##   Map_Date__c Duplicate_Charter_Number__c Count_of_Primary_Stewards__c
## 1  2022-10-13                FALSE                                1
##   Latitude_MapAnything__c Longitude_MapAnything__c
## 1                37.33889                -121.8825
##   Library_Geolocation__Latitude__s Library_Geolocation__Longitude__s
## 1                42.89542                20.86808
##   check_in_count
## 1                0
```

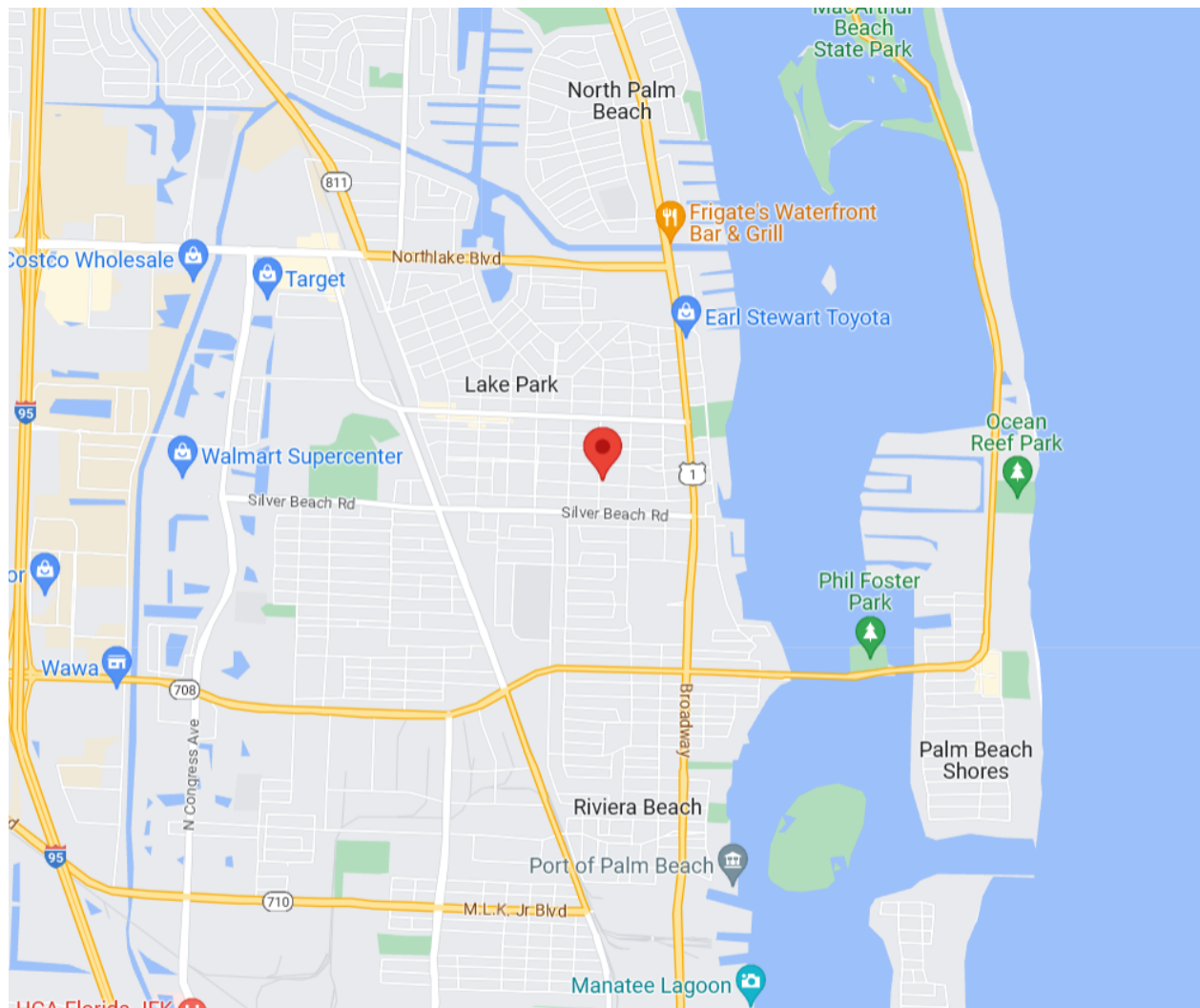


Figure 1: Lake Park in Florida

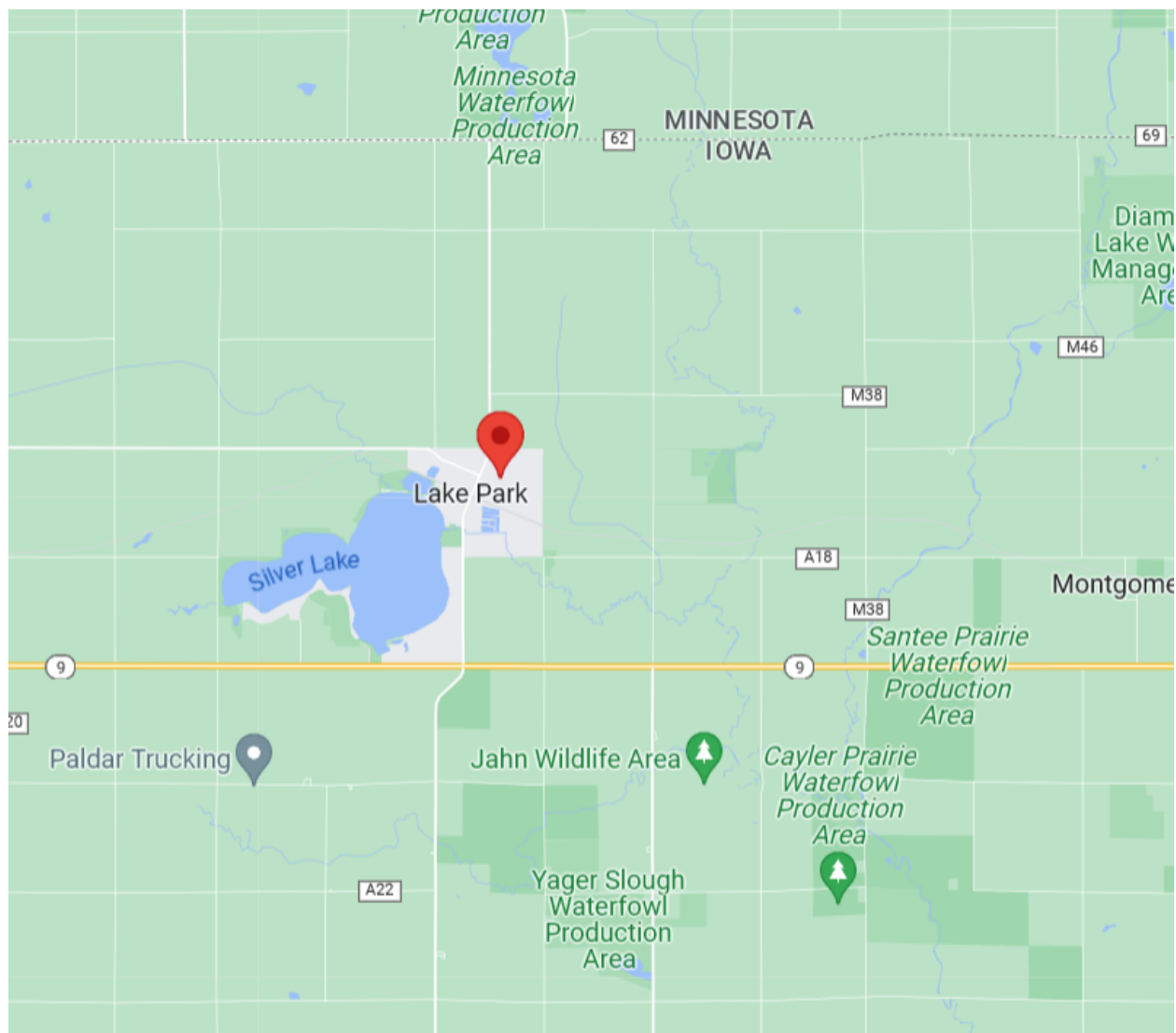


Figure 2: Lake Park in Iowa

The actual location is in Kosovo, but because they put “KS” as the state (which is Kansas, not Kosovo) this row was mistakenly assigned “United States” as its country.

The MapAnything location is in San Jose:

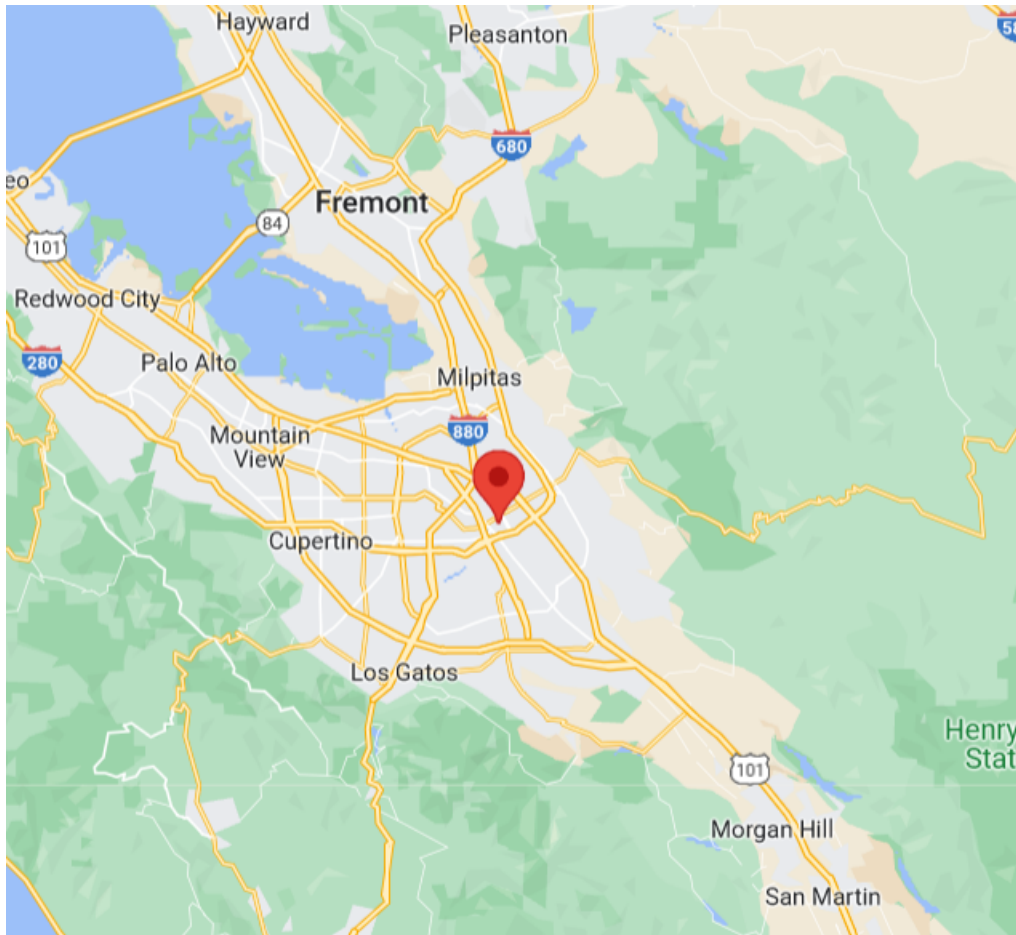


Figure 3: Location in San Jose

and the Geolocation is in Kosovo:

So we have two different examples where the correct coordinates are of different types. If we look at the distribution of coordinates we have:

```
libraries %>% select(Latitude_MapAnything__c, Library_Geolocation__Latitude__s, Longitude_MapAnything__c)

## Latitude_MapAnything__c Library_Geolocation__Latitude__s
## Min. : 0.00 Min. : -27.00
## 1st Qu.: 34.44 1st Qu.: 35.17
## Median : 39.57 Median : 39.73
## Mean : 37.54 Mean : 39.01
## 3rd Qu.: 42.42 3rd Qu.: 42.49
## Max. : 71.30 Max. : 86.95
## Longitude_MapAnything__c Library_Geolocation__Longitude__s
## Min. : -170.47 Min. : -170.49
## 1st Qu.: -105.07 1st Qu.: -106.09
## Median : -88.09 Median : -88.54
## Mean : -90.10 Mean : -93.60
```

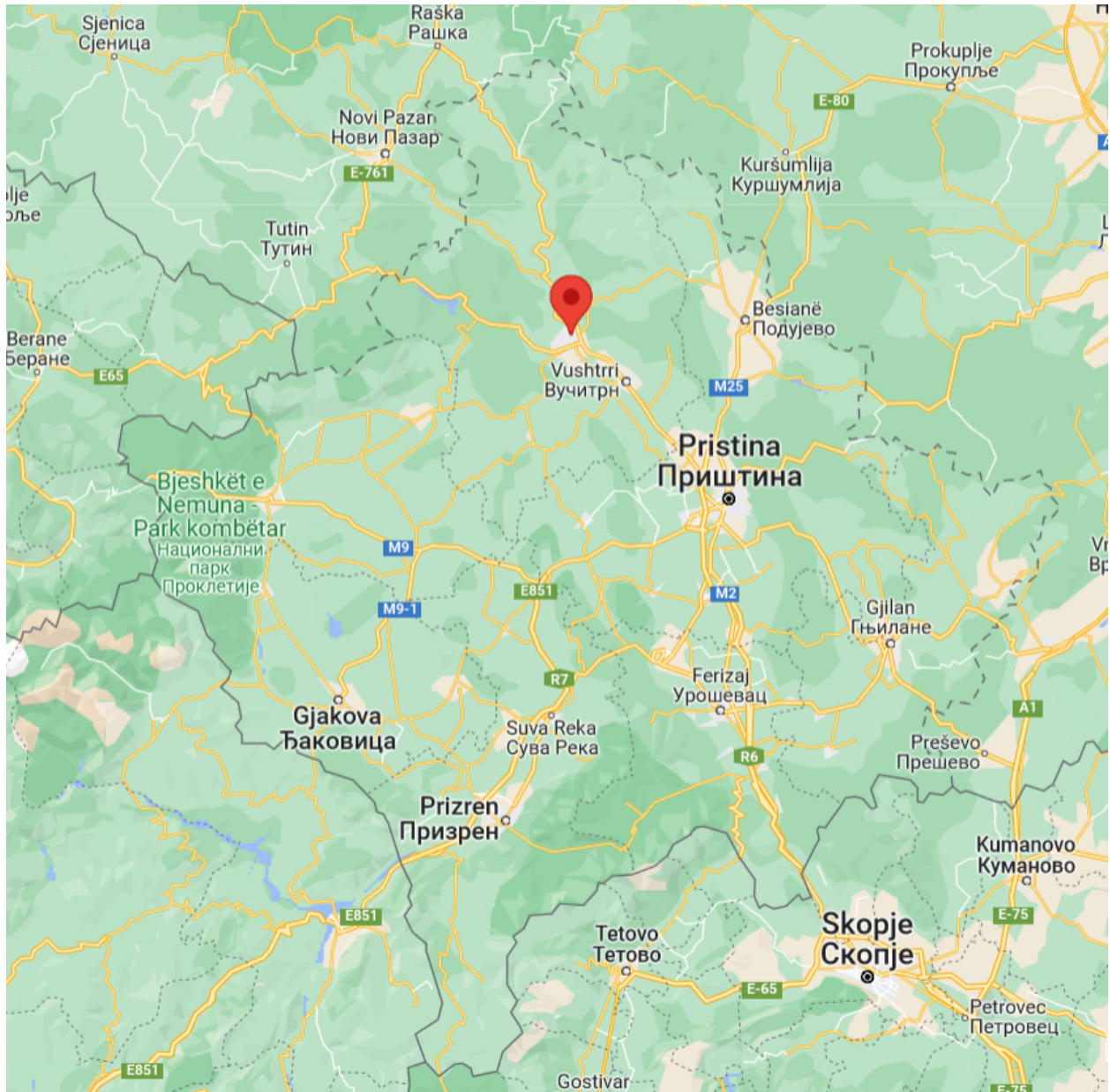


Figure 4: Location in Kosovo



```
## 3rd Qu.: -79.93      3rd Qu.: -80.81
## Max.    : 76.76      Max.    : 158.07
```

Thus summary statistics are similar, but there are enough differences to cause concern. Note that there are a decent amount of rows where the MapAnything coordinates are (0, 0):

```
libraries %>% filter(Latitude_MapAnything__c == 0 & Longitude_MapAnything__c == 0) %>% count()
```

```
##      n
## 1 2343
```

One example is the library with id 14180.

```
libraries %>% filter(id == 14180)
```

```
##      id      Name      Street__c      City__c      State_Province_Region__c
## 1 14180 LIB-000038331 11509 Kenny Dr Fort Worth TX
##      Postal_Zip_Code__c      Country__c      Traveling_Library__c
## 1      76244 United States FALSE
##      Official_Charter_Number__c      First_Map_Date__c      Map_Me__c      Map_Date__c
## 1      77029      2019-03-21      TRUE      2023-04-07
##      Duplicate_Charter_Number__c      Count_of_Primary_Stewards__c
## 1      FALSE      1
##      Latitude_MapAnything__c      Longitude_MapAnything__c
## 1      0      0
##      Library_Geolocation__Latitude__s      Library_Geolocation__Longitude__s
## 1      32.93978      -97.27757
##      check_in_count
## 1      1
```

None of the values look notable other than the (0, 0) MapAnything coordinates, and this library shows up on the official map.

These (0, 0) coordinates are basically missing values as all the libraries we are looking at are located in the US so (0, 0) is definitely an invalid coordinate. If we look at the webapp, it appears that the developers use the geolocation values on the interactive map:

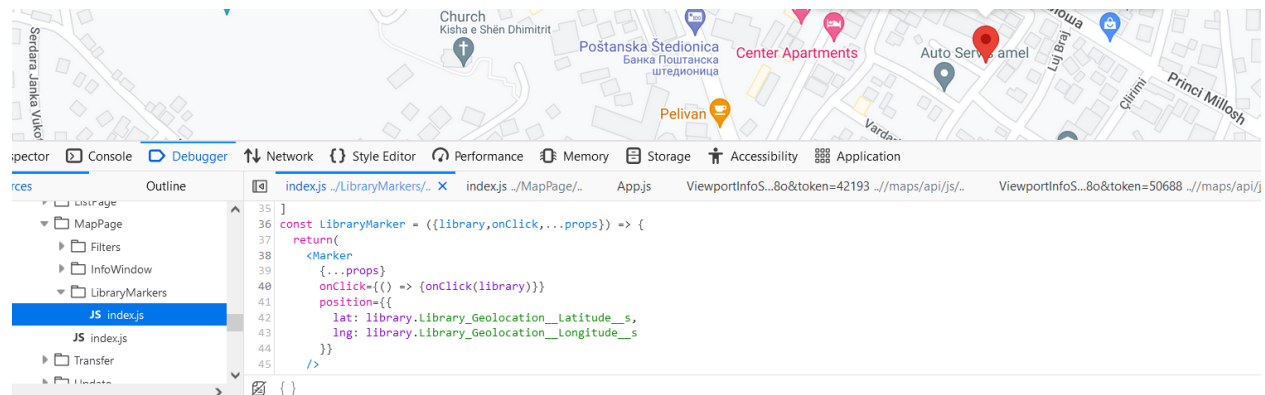


Figure 5: Screenshot of javascript snippet using Library\_Geolocation for the pins on the map

So let's just use the geolocation coordinates exclusively then.

```
libraries %>% select(Postal_Zip_Code__c) %>% unique %>% count
```

```
##      n
## 1 15001
```

Let's drop every data point with coordinates not located within the United States. We can look at what U.S. state a given library is in and then filter out the libraries with no state values.

```
# Convert the coordinates to a sf object
# Our coordinate reference system is the WGS84 standard which is what Google
# maps uses. Its EPSG Code is 4326. The format for a point is (longitude, latitude).
lib_pts <- libraries %>% st_as_sf(coords = c("Library_Geolocation__Longitude__s", "Library_Geolocation__Latitude__s"))

lib_pts_alt <- libraries %>% st_as_sf(coords = c("Longitude_MapAnything__c", "Latitude_MapAnything__c"))

GADM_data <- st_read(dsn = "gadm36_USA_gpkg/gadm36_USA.gpkg", layer = "gadm36_USA_1")

## Reading layer `gadm36_USA_1' from data source
##   `/Users/kcrans/Desktop/projects/little_free/gadm36_USA_gpkg/gadm36_USA.gpkg'
##   using driver `GPKG'
## Simple feature collection with 51 features and 10 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:   xmin: -179.1506 ymin: 18.90986 xmax: 179.7734 ymax: 72.6875
## Geodetic CRS:   WGS 84

state_pts <- st_transform(GADM_data, crs = 4326)

state_names <- state_pts$NAME_1
classifications <- as.integer(st_intersects(lib_pts, state_pts))
alt_classifications <- as.integer(st_intersects(lib_pts_alt, state_pts))

libraries <- libraries %>% mutate(state = state_names[classifications])
classifs <- data.frame(state_names[classifications], state_names[alt_classifications])
sum(is.na(libraries$state))

## [1] 123

#classifications != alt_classifications
```

There are 123 location with coordinates not in the U.S. for whatever reason. Let's take a look at them.

```
libraries %>% filter(is.na(state))
```

##	id	Name	Street__c
## 1	293	LIB-000024202	North Main St
## 2	2278	LIB-000026204	Perumkulam
## 3	2981	LIB-000026910	<NA>
## 4	3385	LIB-000027318	6862 Chico Way NW
## 5	3919	LIB-000027852	Keerkring 112
## 6	4141	LIB-000028075	Aljazar st
## 7	4515	LIB-000028452	Chequers Corner, Hurst Drive
## 8	4737	LIB-000034539	<NA>
## 9	4871	LIB-000034675	<NA>
## 10	5604	LIB-000028915	1 Stirrup Close
## 11	6601	LIB-000029925	1951 Miracle Mile Dr E
## 12	7355	LIB-000030684	Dorfstrasse 21
## 13	7764	LIB-000037639	<NA>
## 14	7915	LIB-000037791	<NA>
## 15	10689	LIB-000033964	Tautus 59
## 16	10894	LIB-000034171	1932 S. Oceanshore Blvd
## 17	11411	LIB-000035034	<NA>



## 18	11427 LIB-000035050	<NA>
## 19	11524 LIB-000035147	<NA>
## 20	11526 LIB-000035149	<NA>
## 21	11780 LIB-000035405	<NA>
## 22	11938 LIB-000035565	<NA>
## 23	13072 LIB-000036784	<NA>
## 24	13396 LIB-000037112	<NA>
## 25	13499 LIB-000037217	<NA>
## 26	14004 LIB-000038154	<NA>
## 27	14091 LIB-000038241	<NA>
## 28	14136 LIB-000038286	<NA>
## 29	14183 LIB-000038334	<NA>
## 30	14209 LIB-000038363	<NA>
## 31	14549 LIB-000038708	<NA>
## 32	14849 LIB-000039024	<NA>
## 33	14850 LIB-000039025	<NA>
## 34	16202 LIB-000039271	<NA>
## 35	16520 LIB-000039608	<NA>
## 36	16657 LIB-000039745	<NA>
## 37	16883 LIB-000039976	<NA>
## 38	17272 LIB-000040365	<NA>
## 39	17339 LIB-000040432	<NA>
## 40	17437 LIB-000040531	<NA>
## 41	17438 LIB-000040532	<NA>
## 42	17576 LIB-000040671	<NA>
## 43	18550 LIB-000042906	<NA>
## 44	19119 LIB-000046571	319 Dunham Point Road
## 45	19356 LIB-000043089	<NA>
## 46	19421 LIB-000043160	Nimitiz Rd.
## 47	20217 LIB-000044052	Co
## 48	22166 LIB-000049022	7223 South Virginia Dare Trail
## 49	23608 LIB-000050541	176 Cherry And Webb Lane
## 50	24116 LIB-000046066	<NA>
## 51	24174 LIB-000046128	<NA>
## 52	26050 LIB-000051223	<NA>
## 53	26622 LIB-000053055	1 Newport Ave
## 54	26770 LIB-000053225	<NA>
## 55	26896 LIB-000053363	<NA>
## 56	27042 LIB-000053562	<NA>
## 57	27158 LIB-000053845	<NA>
## 58	27201 LIB-000053934	<NA>
## 59	28106 LIB-000056674	<NA>
## 60	28458 LIB-000057066	<NA>
## 61	28974 LIB-000057661	<NA>
## 62	29083 LIB-000057791	<NA>
## 63	29252 LIB-000057995	Fisherman's Beach
## 64	29894 LIB-000058848	<NA>
## 65	30188 LIB-000059287	<NA>
## 66	32185 LIB-000061778	<NA>
## 67	33246 LIB-000063296	<NA>
## 68	34090 LIB-000064381	<NA>
## 69	34172 LIB-000064463	567 Angell Street
## 70	34415 LIB-000064714	<NA>
## 71	34531 LIB-000064831	<NA>

## 72	34890	LIB-000065194	<NA>
## 73	36568	LIB-000066887	<NA>
## 74	36967	LIB-000067287	<NA>
## 75	38204	LIB-000068525	<NA>
## 76	39648	LIB-000002433	<NA>
## 77	40434	LIB-000003393	<NA>
## 78	41452	LIB-000004422	KomunitnÃ© centrum HorÃ;reÃ~\nLesnÃ; 1
## 79	41682	LIB-000004654	Weststraat 73
## 80	41706	LIB-000004679	Ã...lykkevej 13
## 81	43194	LIB-000006199	615 W 10th St
## 82	46526	LIB-000009684	510 Shorewood Drive
## 83	49152	LIB-000012489	<NA>
## 84	49709	LIB-000013059	148 Marina Plaza
## 85	49984	LIB-000013334	#14 Plaza del Mercado, Ave. Rafael Cordero 1000
## 86	51495	LIB-000014854	86 Butts Rock Rd.
## 87	51556	LIB-000014915	Ocean Ave & Lincoln
## 88	56156	LIB-000019673	2013 Wildwood Lane
## 89	58958	LIB-000022491	1800 gulf road
## 90	60572	LIB-000068898	<NA>
## 91	60949	LIB-000069498	<NA>
## 92	61086	LIB-000069752	<NA>
## 93	61216	LIB-000069941	<NA>
## 94	63301	LIB-000071612	185 Ferry Road
## 95	63749	LIB-000072009	<NA>
## 96	64910	LIB-000073170	261 Hart Street
## 97	65309	LIB-000073569	<NA>
## 98	65780	LIB-000074020	<NA>
## 99	66004	LIB-000074265	<NA>
## 100	66571	LIB-000074821	131 Woodscape Dr
## 101	66663	LIB-000074911	<NA>
## 102	68872	LIB-000077117	116 S Santa Fe
## 103	69834	LIB-000078080	<NA>
## 104	70053	LIB-000078298	<NA>
## 105	70599	LIB-000078842	<NA>
## 106	71250	LIB-000079496	<NA>
## 107	71624	LIB-000079869	Bishop Lane (between Corner And #35)
## 108	71741	LIB-000079986	2494 Newcastle St
## 109	71951	LIB-000080196	18300 Justice Way (Across The Street From This Address)
## 110	72196	LIB-000080441	1198 Venice Ave
## 111	72535	LIB-000080779	1710 S Trenon Ave
## 112	72704	LIB-000080948	<NA>
## 113	73257	LIB-000081501	4 Shelter Cove LN
## 114	74679	LIB-000082920	<NA>
## 115	74893	LIB-000083136	3515 Mccauley Rd
## 116	74894	LIB-000083137	2776 Sanford Rd Se
## 117	75691	LIB-000083936	22 Beachwood Drive
## 118	75881	LIB-000084125	North Campus Building, Perandori Dushan, Mitrovicë
## 119	76243	LIB-000084486	<NA>
## 120	78630	LIB-000086854	<NA>
## 121	78786	LIB-000087009	<NA>
## 122	79239	LIB-000087459	114 W. Chicago Street
## 123	82395	LIB-000090612	64 Sleeper Street
##		City__c State_Province_Region__c Postal_Zip_Code__c	
## 1		Raynham	MA 02767

## 2	Kottarakara	<NA>	691566
## 3	Star	ID	83669
## 4	Bremerton	WA	98383
## 5	De Meern	<NA>	3454kz
## 6	Khartoum	<NA>	11112
## 7	Walton on the Hill	<NA>	KT20 7QT
## 8	Cove	OR	97824
## 9	TUPELO	MS	38804
## 10	Newbury	<NA>	RG14 7XD
## 11	Port Orchard	WA	98366
## 12	Bühler	<NA>	9055
## 13	Coeur D'Alene	ID	83814
## 14	Monroe	WA	98272
## 15	Daugavpils	<NA>	5404
## 16	Flagler Beach	FL	32136
## 17	Marthasville	MO	63357
## 18	Glen Carbon	IL	62034
## 19	Auburn	AL	36830
## 20	Auburn	AL	36830
## 21	Parker	CO	80134
## 22	Greenville	SC	29615
## 23	Tustin	CA	92780- _
## 24	Erie	PA	16563
## 25	Pottstown	PA	19465
## 26	Independence	KY	41051
## 27	Johnson Lake	NE	68937
## 28	Wilmington	NC	28409
## 29	Marysville	WA	98270
## 30	Friday Harbor	WA	98250
## 31	Nashville	TN	37212
## 32	Shippenville	PA	16254
## 33	Leeper	PA	16233
## 34	Milford	NH	03055
## 35	Old Lyme	CT	06371
## 36	Encino	CA	91316
## 37	Yardley	PA	19067
## 38	Pacific	WA	98047
## 39	North Olmsted	OH	44070
## 40	Alcoa	TN	37701
## 41	Alcoa	TN	37701
## 42	Tiverton	RI	02878
## 43	Forest Hills	NY	11375
## 44	Deer Isle	ME	04627
## 45	Alytus	AL	63210
## 46	Dover	DE	19903
## 47	Bremerton	WA	98312
## 48	Nag's Head	NC	27959
## 49	Westport	MA	02790
## 50	San Pedro	CA	90732
## 51	Charleston	SC	29414
## 52	Fort Dick	CA	95531
## 53	North Kingstown	CT	02852
## 54	Mililani	HI	96789
## 55	Minnetonka	MN	55345

## 56	Manalapan	NJ	07726
## 57	Lancaster	PA	17601
## 58	Guilford	CT	06437
## 59	Corrales	NM	87048
## 60	North Charleston	SC	29420
## 61	New Canaan	CT	06840
## 62	Berea	KY	40403
## 63	Swampscott	MA	01907
## 64	Prescott	WI	54021
## 65	Franklin	TN	37064
## 66	Chino	CA	91708
## 67	Swansboro	NC	28584
## 68	Mt Juliet	TN	37122
## 69	Providence	RI	02906
## 70	Nashville	TN	37205
## 71	Knoxville	TN	37920
## 72	Wilmette	IL	60091
## 73	Greenville	DE	19807
## 74	Powell	OH	43065
## 75	Barrington	RI	02777
## 76	Longmont	CO	80503
## 77	Corvallis	OR	97333
## 78	Bratislava	<NA>	81104
## 79	Den Helder	<NA>	1781 BW
## 80	Risskov	<NA>	8240
## 81	Rush City	MN	55069
## 82	International Falls	MN	56649
## 83	Cranberry Twp	PA	16066
## 84	Dunedin	FL	34698
## 85	Caguas	<NA>	00725
## 86	Little Compton	RI	02837
## 87	Avon by the Sea	NJ	07717
## 88	Anchorage	AK	99517
## 89	Tarpon springs	FL	34689
## 90	Detroit	MI	48221
## 91	North Topsail Beach	NC	28460
## 92	Crestview	FL	32539
## 93	Provincetown	MA	02657
## 94	Saunderstown	RI	02874
## 95	Monterey	CA	93940
## 96	Dighton	MA	02715
## 97	Lawrence	KS	66049
## 98	North Kingstown	RI	02852
## 99	Huntersville	NC	28078
## 100	Canton	MS	39046
## 101	Jefferson	GA	30549
## 102	Galva	KS	67460
## 103	Davenport	FL	33837
## 104	Truro	MA	02666
## 105	Homer	AK	99603
## 106	Seattle	WA	98115
## 107	Madison	CT	06443
## 108	Eugene	OR	97404
## 109	Lakeville	MN	55044

## 110	Mentone	CA	92359
## 111	Tulsa	OK	74120
## 112	Peru	ME	04290
## 113	Hilton Head Island	SC	29928
## 114	Barrington	RI	02806
## 115	Smyrna	GA	30080
## 116	Smyrna	GA	30080
## 117	Warwick	RI	02818
## 118	Mitrovica	KS	40000
## 119	Wilsonville	OR	97070
## 120	Sandbridge	VA	23456
## 121	Madison	MS	39110
## 122	Caldwell	ID	83605
## 123	Boston	MA	02210
##	Country__c	Traveling_Library__c	Official_Charter_Number__c
## 1	United States	FALSE	49077
## 2	United States	FALSE	57110
## 3	United States	FALSE	29203
## 4	United States	FALSE	54282
## 5	United States	FALSE	58388
## 6	United States	FALSE	46879
## 7	United States	FALSE	46255
## 8	United States	FALSE	75704
## 9	United States	FALSE	67533
## 10	United States	FALSE	60235
## 11	United States	FALSE	66881
## 12	United States	FALSE	M68171
## 13	United States	FALSE	21735
## 14	United States	FALSE	72070
## 15	United States	FALSE	69989
## 16	United States	FALSE	76585
## 17	United States	FALSE	64269
## 18	United States	FALSE	57369
## 19	United States	FALSE	77823
## 20	United States	FALSE	77824
## 21	United States	FALSE	77362
## 22	United States	FALSE	77773
## 23	United States	FALSE	80984
## 24	United States	FALSE	69514
## 25	United States	FALSE	56281
## 26	United States	FALSE	23531
## 27	United States	FALSE	85091
## 28	United States	FALSE	76217
## 29	United States	FALSE	85156
## 30	United States	FALSE	74788
## 31	United States	FALSE	86163
## 32	United States	FALSE	80878
## 33	United States	FALSE	80879
## 34	United States	FALSE	85751
## 35	United States	FALSE	77385
## 36	United States	FALSE	87863
## 37	United States	FALSE	88289
## 38	United States	FALSE	69842
## 39	United States	FALSE	89237

## 40	United States	FALSE	89186
## 41	United States	FALSE	89187
## 42	United States	FALSE	88753
## 43	United States	FALSE	80004
## 44	United States	FALSE	83568
## 45	United States	FALSE	13135
## 46	United States	FALSE	94044
## 47	United States	FALSE	94608
## 48	United States	FALSE	101283
## 49	United States	FALSE	72143
## 50	United States	FALSE	99380
## 51	United States	FALSE	98608
## 52	United States	FALSE	31814
## 53	United States	FALSE	107184
## 54	United States	FALSE	93143
## 55	United States	FALSE	103897
## 56	United States	FALSE	108025
## 57	United States	FALSE	108139
## 58	United States	FALSE	108742
## 59	United States	FALSE	111454
## 60	United States	FALSE	109197
## 61	United States	FALSE	102233
## 62	United States	FALSE	92579
## 63	United States	FALSE	95120
## 64	United States	FALSE	116190
## 65	United States	FALSE	112264
## 66	United States	FALSE	M107532
## 67	United States	FALSE	87496
## 68	United States	FALSE	80662
## 69	United States	FALSE	117921
## 70	United States	FALSE	117881
## 71	United States	FALSE	120972
## 72	United States	FALSE	107416
## 73	United States	FALSE	116922
## 74	United States	FALSE	124325
## 75	United States	FALSE	126314
## 76	United States	FALSE	14533
## 77	United States	FALSE	8765
## 78	United States	FALSE	12411
## 79	United States	FALSE	14158
## 80	United States	FALSE	14384
## 81	United States	FALSE	1826
## 82	United States	FALSE	6426
## 83	United States	FALSE	23735
## 84	United States	FALSE	G10014
## 85	United States	FALSE	26275
## 86	United States	FALSE	26399
## 87	United States	FALSE	29517
## 88	United States	FALSE	39053
## 89	United States	FALSE	23484
## 90	United States	FALSE	126557
## 91	United States	FALSE	58129
## 92	United States	FALSE	112883
## 93	United States	FALSE	124938

## 94	United States	FALSE	135342
## 95	United States	FALSE	130403
## 96	United States	FALSE	135518
## 97	United States	FALSE	137297
## 98	United States	FALSE	139911
## 99	United States	FALSE	136122
## 100	United States	FALSE	137195
## 101	United States	FALSE	141291
## 102	United States	FALSE	144262
## 103	United States	FALSE	22361
## 104	United States	FALSE	138629
## 105	United States	FALSE	138808
## 106	United States	FALSE	96689
## 107	United States	FALSE	148379
## 108	United States	FALSE	147285
## 109	United States	FALSE	147785
## 110	United States	FALSE	149811
## 111	United States	FALSE	150219
## 112	United States	FALSE	91287
## 113	United States	FALSE	151089
## 114	United States	FALSE	147452
## 115	United States	FALSE	141828
## 116	United States	FALSE	141829
## 117	United States	FALSE	152156
## 118	United States	FALSE	85849
## 119	United States	FALSE	37523
## 120	United States	FALSE	131731
## 121	United States	FALSE	157744
## 122	United States	FALSE	162828
## 123	United States	FALSE	160393
##	First_Map_Date__c	Map_Me__c	Map_Date__c Duplicate_Charter_Number__c
## 1	2017-03-24	TRUE	2017-03-24 FALSE
## 2	2017-06-28	TRUE	2017-06-28 FALSE
## 3	2017-07-20	TRUE	2022-08-11 FALSE
## 4	2017-08-07	TRUE	2017-08-07 FALSE
## 5	2017-08-29	TRUE	2017-08-29 FALSE
## 6	2017-09-07	TRUE	2017-09-07 FALSE
## 7	2017-09-28	TRUE	2017-09-28 FALSE
## 8	2018-08-17	TRUE	2018-08-17 FALSE
## 9	2018-08-21	TRUE	2018-08-21 FALSE
## 10	2017-10-16	TRUE	2017-10-16 FALSE
## 11	2017-12-13	TRUE	2017-12-13 FALSE
## 12	2018-01-29	TRUE	2018-02-19 FALSE
## 13	2019-01-25	TRUE	2019-01-25 FALSE
## 14	2019-02-05	TRUE	2019-02-05 FALSE
## 15	2018-07-27	TRUE	2018-07-27 FALSE
## 16	2018-08-03	TRUE	2018-08-03 FALSE
## 17	2018-09-04	TRUE	2018-09-04 FALSE
## 18	2018-09-04	TRUE	2018-09-04 FALSE
## 19	2018-09-10	TRUE	2018-09-10 FALSE
## 20	2018-09-10	TRUE	2018-09-10 FALSE
## 21	2018-09-21	TRUE	2018-09-21 FALSE
## 22	2018-10-02	TRUE	2018-10-02 FALSE
## 23	2018-11-27	TRUE	2018-11-27 FALSE

## 24	2018-12-17	TRUE	2018-12-17	FALSE
## 25	2018-12-27	TRUE	2018-12-27	FALSE
## 26	2019-03-08	TRUE	2019-03-08	FALSE
## 27	2019-03-15	TRUE	2019-03-15	FALSE
## 28	2019-03-18	TRUE	2019-03-19	FALSE
## 29	2019-03-21	TRUE	2020-10-01	FALSE
## 30	2019-03-25	TRUE	2019-03-25	FALSE
## 31	2019-04-12	TRUE	2019-04-12	FALSE
## 32	2019-04-26	TRUE	2019-04-26	FALSE
## 33	2019-04-26	TRUE	2019-04-26	FALSE
## 34	2019-05-06	TRUE	2019-05-06	FALSE
## 35	2019-05-20	TRUE	2019-05-20	FALSE
## 36	2019-05-24	TRUE	2019-05-24	FALSE
## 37	2019-06-04	TRUE	2019-06-04	FALSE
## 38	2019-06-17	TRUE	2019-06-17	FALSE
## 39	2019-06-24	TRUE	2019-06-24	FALSE
## 40	2019-06-24	TRUE	2019-06-24	FALSE
## 41	2019-06-24	TRUE	2019-06-24	FALSE
## 42	2019-06-27	TRUE	2022-06-14	FALSE
## 43	2019-09-23	TRUE	2019-09-23	FALSE
## 44	2020-04-13	TRUE	2020-04-13	FALSE
## 45	2019-10-01	TRUE	2019-10-01	FALSE
## 46	2019-10-07	TRUE	2019-10-07	FALSE
## 47	2019-11-15	TRUE	2019-11-15	FALSE
## 48	2020-06-01	TRUE	2020-06-01	FALSE
## 49	2020-07-07	TRUE	2020-07-07	FALSE
## 50	2020-04-01	TRUE	2020-04-01	FALSE
## 51	2020-04-06	TRUE	2020-04-06	FALSE
## 52	2020-07-23	TRUE	2020-08-03	FALSE
## 53	2020-09-02	TRUE	2020-09-02	FALSE
## 54	2020-09-09	TRUE	2020-09-09	FALSE
## 55	2020-09-09	TRUE	2020-09-09	FALSE
## 56	2020-09-10	TRUE	2020-09-10	FALSE
## 57	2020-09-14	TRUE	2020-09-14	FALSE
## 58	2020-09-15	TRUE	2020-09-15	FALSE
## 59	2020-10-14	TRUE	2020-10-14	FALSE
## 60	2020-10-20	TRUE	2020-10-20	FALSE
## 61	2020-11-09	TRUE	2020-11-09	FALSE
## 62	2020-11-12	TRUE	2020-11-12	FALSE
## 63	2020-11-18	TRUE	2020-11-18	FALSE
## 64	2020-12-10	TRUE	2020-12-10	FALSE
## 65	2020-12-23	TRUE	2020-12-23	FALSE
## 66	2021-03-05	TRUE	2021-03-05	FALSE
## 67	2021-04-12	TRUE	2021-04-12	FALSE
## 68	2021-05-03	TRUE	2021-05-03	FALSE
## 69	2021-05-03	TRUE	2021-05-03	FALSE
## 70	2021-05-12	TRUE	2021-05-12	FALSE
## 71	2021-05-13	TRUE	2021-05-13	FALSE
## 72	2021-05-17	TRUE	2021-05-17	FALSE
## 73	2021-06-14	TRUE	2021-06-15	FALSE
## 74	2021-06-18	TRUE	2021-06-18	FALSE
## 75	2021-07-15	TRUE	2021-07-15	FALSE
## 76	2018-09-04	TRUE	2018-09-04	FALSE
## 77	2015-03-28	TRUE	2019-01-22	FALSE



## 78	2014-10-03	TRUE	2019-01-22	FALSE
## 79	2014-10-03	TRUE	2019-01-22	FALSE
## 80	2014-10-03	TRUE	2019-01-22	FALSE
## 81	2013-02-09	TRUE	2013-02-09	FALSE
## 82	2014-10-10	TRUE	2019-01-22	FALSE
## 83	2019-01-17	TRUE	2019-01-17	FALSE
## 84	2015-06-15	TRUE	2015-06-15	FALSE
## 85	2015-06-24	TRUE	2015-06-24	FALSE
## 86	2015-09-10	TRUE	2020-04-01	FALSE
## 87	2015-09-10	TRUE	2015-09-10	FALSE
## 88	2016-07-05	TRUE	2016-07-05	FALSE
## 89	2016-11-14	TRUE	2016-11-14	FALSE
## 90	2021-07-26	TRUE	2021-07-26	FALSE
## 91	2021-08-11	TRUE	2021-08-11	FALSE
## 92	2021-08-17	TRUE	2021-08-17	FALSE
## 93	2021-08-23	TRUE	2021-08-23	FALSE
## 94	2021-10-13	TRUE	2021-10-18	FALSE
## 95	2021-10-27	TRUE	2021-10-27	FALSE
## 96	2021-12-13	TRUE	2021-12-13	FALSE
## 97	2022-01-04	TRUE	2022-01-04	FALSE
## 98	2022-01-24	TRUE	2022-01-26	FALSE
## 99	2022-02-07	TRUE	2022-02-07	FALSE
## 100	2022-03-07	TRUE	2022-03-07	FALSE
## 101	2022-03-14	TRUE	2022-03-14	FALSE
## 102	2022-05-03	TRUE	2022-05-03	FALSE
## 103	2022-05-24	TRUE	2022-05-24	FALSE
## 104	2022-05-26	TRUE	2022-05-26	FALSE
## 105	2022-06-07	TRUE	2022-06-07	FALSE
## 106	2022-06-21	TRUE	2022-06-21	FALSE
## 107	2022-07-03	TRUE	2022-07-03	FALSE
## 108	2022-06-28	TRUE	2022-06-28	FALSE
## 109	2022-07-05	TRUE	2022-07-05	FALSE
## 110	2022-07-26	TRUE	2022-07-26	FALSE
## 111	2022-07-26	TRUE	2022-07-26	FALSE
## 112	2022-07-25	TRUE	2022-07-25	FALSE
## 113	2022-10-24	TRUE	2022-10-24	FALSE
## 114	2022-09-08	TRUE	2022-09-08	FALSE
## 115	2022-09-16	TRUE	2022-09-16	FALSE
## 116	2022-09-12	TRUE	2022-09-12	FALSE
## 117	2022-10-04	TRUE	2022-10-04	FALSE
## 118	2022-10-13	TRUE	2022-10-13	FALSE
## 119	2022-10-20	TRUE	2022-10-20	FALSE
## 120	2023-02-21	TRUE	2023-02-21	FALSE
## 121	2023-03-01	TRUE	2023-03-01	FALSE
## 122	2023-03-20	TRUE	2023-03-20	FALSE
## 123	2023-04-10	TRUE	2023-04-10	FALSE
##	Count_of_Primary_Stewards__c		Latitude_MapAnything__c	
## 1		1	41.94253	
## 2		1	9.03882	
## 3		1	43.69603	
## 4		1	47.62469	
## 5		1	52.08682	
## 6		1	15.50065	
## 7		1	51.27481	

## 8	1	45.30151
## 9	1	34.24820
## 10	1	51.38518
## 11	1	47.55189
## 12	2	47.37292
## 13	1	47.67949
## 14	1	47.85537
## 15	1	55.87810
## 16	1	29.46048
## 17	1	38.62891
## 18	1	38.77548
## 19	1	32.60903
## 20	1	32.60903
## 21	1	39.52464
## 22	1	34.86674
## 23	1	33.73510
## 24	1	42.12067
## 25	1	40.23540
## 26	1	38.96558
## 27	1	40.40762
## 28	1	34.17041
## 29	1	48.05626
## 30	1	48.53449
## 31	1	36.13380
## 32	1	41.25121
## 33	1	41.37164
## 34	1	42.83471
## 35	1	41.32375
## 36	1	34.16536
## 37	1	40.24233
## 38	1	47.25008
## 39	1	41.43607
## 40	1	35.76489
## 41	1	35.76489
## 42	1	41.66056
## 43	1	40.72267
## 44	1	44.21357
## 45	1	54.40273
## 46	1	39.17734
## 47	1	47.57192
## 48	0	35.90963
## 49	1	41.65611
## 50	1	33.72969
## 51	1	32.82421
## 52	2	41.86655
## 53	1	41.57126
## 54	1	21.43769
## 55	1	44.90991
## 56	1	40.29804
## 57	1	40.06113
## 58	1	41.28370
## 59	1	35.25445
## 60	1	32.93033
## 61	1	41.14464

## 62	1	37.57095
## 63	1	42.47464
## 64	1	44.75178
## 65	1	35.92170
## 66	1	33.96430
## 67	1	34.70468
## 68	1	36.16360
## 69	1	41.83050
## 70	1	36.13297
## 71	1	35.92650
## 72	1	42.07568
## 73	1	39.76317
## 74	1	40.15559
## 75	1	41.77445
## 76	1	40.14814
## 77	1	44.56045
## 78	1	48.15755
## 79	1	52.95998
## 80	1	56.20131
## 81	1	45.67904
## 82	1	48.58890
## 83	1	40.68226
## 84	1	28.01277
## 85	1	18.23743
## 86	1	41.48505
## 87	1	40.18972
## 88	1	61.20429
## 89	1	28.14467
## 90	1	42.43593
## 91	1	34.54495
## 92	1	30.76327
## 93	1	42.05261
## 94	1	41.50778
## 95	1	36.59700
## 96	1	41.80444
## 97	1	38.97055
## 98	1	41.61362
## 99	1	35.41121
## 100	1	32.61198
## 101	1	34.11803
## 102	1	38.38292
## 103	1	28.21429
## 104	1	41.97150
## 105	1	59.64814
## 106	1	47.67951
## 107	1	41.28296
## 108	1	44.10012
## 109	1	44.68420
## 110	1	34.06994
## 111	1	36.13765
## 112	1	44.46699
## 113	1	32.18044
## 114	1	41.74405
## 115	1	33.87086

## 116	2	33.88523
## 117	1	41.66889
## 118	1	37.33889
## 119	1	45.30687
## 120	1	0.00000
## 121	1	0.00000
## 122	1	0.00000
## 123	1	0.00000
##	Longitude_MapAnything__c	Library_Geolocation__Latitude__s
## 1	-71.0516899	41.92650
## 2	76.7597700	9.03882
## 3	-116.4903600	43.71855
## 4	-122.7074200	47.62468
## 5	5.0297300	52.08682
## 6	32.5598994	15.50065
## 7	-0.2506241	51.27481
## 8	-117.8085500	45.17816
## 9	-88.5337400	34.26976
## 10	-1.3159300	51.38518
## 11	-122.5429400	47.55189
## 12	9.4252400	47.37292
## 13	-116.7794700	47.33500
## 14	-121.9776100	47.85546
## 15	26.5483600	55.87810
## 16	-81.1179200	29.46050
## 17	-91.0646200	38.62850
## 18	-89.9565000	38.75380
## 19	-85.4790500	32.33170
## 20	-85.4790500	32.33160
## 21	-104.7711800	39.56020
## 22	-82.3512000	34.88528
## 23	-117.8250200	33.70000
## 24	-79.9852400	49.11986
## 25	-75.6568300	40.21452
## 26	-84.5994100	38.95427
## 27	-95.9993300	40.68347
## 28	-77.8839100	34.16000
## 29	-122.1704200	48.06956
## 30	-123.0164800	48.31673
## 31	-86.7985400	36.08420
## 32	-79.4629400	41.23751
## 33	-79.3058100	41.36799
## 34	-71.6470400	42.98083
## 35	-72.3308000	41.28340
## 36	-118.5270600	34.10656
## 37	-74.8387900	40.14236
## 38	-122.2522000	47.26654
## 39	-81.8815400	41.40341
## 40	-83.9733800	35.79847
## 41	-83.9733800	35.79780
## 42	-71.1896600	41.62537
## 43	-73.8479100	40.42438
## 44	-68.7195743	44.21276
## 45	24.0316200	54.40512

## 46	-75.5145299	39.10372
## 47	-122.6522300	47.34160
## 48	-75.5963900	35.91011
## 49	-71.0791300	41.51308
## 50	-118.3119200	32.72057
## 51	-80.0463600	32.48223
## 52	-124.1447500	41.52093
## 53	-71.4416000	41.57137
## 54	-158.0193600	21.28590
## 55	-93.5031900	44.90135
## 56	-74.3583500	40.32285
## 57	-76.3004600	40.08727
## 58	-72.6829400	41.36700
## 59	-106.6110100	35.22483
## 60	-80.0951000	32.90992
## 61	-73.4890200	77.49770
## 62	-84.3009500	84.17340
## 63	-70.9251200	42.46767
## 64	-92.7959900	44.69810
## 65	-86.8721300	86.94875
## 66	-117.6243400	27.20460
## 67	-77.0713200	34.68940
## 68	-86.5043400	86.50000
## 69	-71.3852600	41.83050
## 70	-86.8261000	86.88802
## 71	-83.9214900	83.93900
## 72	-87.7083100	27.20460
## 73	-75.5873200	75.37368
## 74	-83.0878200	49.08361
## 75	-71.2983500	41.73706
## 76	-105.1576300	40.19278
## 77	-123.2675000	44.33524
## 78	17.0949088	48.15755
## 79	4.7675100	52.95998
## 80	10.2561600	56.20131
## 81	-92.9710800	45.67904
## 82	-93.4644200	48.58890
## 83	-80.1289400	40.71211
## 84	-82.7924076	28.01326
## 85	-66.0298581	18.23743
## 86	-71.1414100	41.48536
## 87	-74.0099700	40.18918
## 88	-149.9224700	61.20386
## 89	-82.7896231	28.14444
## 90	-83.1415600	42.24120
## 91	-77.3863100	34.27210
## 92	-86.5625200	30.42230
## 93	-70.1895100	42.36000
## 94	-71.4186000	41.50778
## 95	-121.8963400	36.80070
## 96	-71.1238300	41.80444
## 97	-95.2798200	38.96314
## 98	-71.4623100	41.58048
## 99	-80.8381800	24.85310

## 100	-90.0394100	32.50391
## 101	-83.5767400	83.57240
## 102	-97.5408900	27.20460
## 103	-81.6337200	36.12800
## 104	-70.0698600	70.03786
## 105	-151.5335500	59.63820
## 106	-122.3197000	47.68451
## 107	-72.5788900	41.28571
## 108	-123.1386000	44.10012
## 109	-93.3137000	44.68000
## 110	-117.1266000	34.06654
## 111	-95.9698900	-27.00000
## 112	-70.3983700	44.46333
## 113	-80.7346200	31.10570
## 114	-71.3382900	41.73965
## 115	-84.5230300	33.86809
## 116	-84.5272452	84.52580
## 117	-71.4234195	41.66889
## 118	-121.8825300	42.89542
## 119	-122.7645100	45.30928
## 120	0.0000000	36.42580
## 121	0.0000000	32.46640
## 122	0.0000000	43.67455
## 123	0.0000000	42.35282
##	Library_Geolocation__Longitude__s	check_in_count state
## 1	71.0514000	0 <NA>
## 2	76.7597700	0 <NA>
## 3	-166.4865100	0 <NA>
## 4	-122.7063340	3 <NA>
## 5	5.0297300	0 <NA>
## 6	32.5598994	0 <NA>
## 7	-0.2506241	0 <NA>
## 8	117.5060000	0 <NA>
## 9	88.7178333	0 <NA>
## 10	-1.3159300	0 <NA>
## 11	-122.5417569	2 <NA>
## 12	9.4252400	0 <NA>
## 13	116.4931000	0 <NA>
## 14	122.0019456	0 <NA>
## 15	26.5483600	0 <NA>
## 16	-81.1179484	2 <NA>
## 17	91.0616000	0 <NA>
## 18	89.9982800	0 <NA>
## 19	85.2843000	0 <NA>
## 20	85.2837000	0 <NA>
## 21	104.7926000	0 <NA>
## 22	82.3450000	0 <NA>
## 23	117.8000000	0 <NA>
## 24	-79.9797340	0 <NA>
## 25	-73.7091035	0 <NA>
## 26	84.5433220	0 <NA>
## 27	99.8288880	0 <NA>
## 28	77.8800000	0 <NA>
## 29	122.1551270	1 <NA>

## 30	123.0117100	0	<NA>
## 31	86.4751000	0	<NA>
## 32	79.4173600	0	<NA>
## 33	79.3087200	0	<NA>
## 34	71.9197222	0	<NA>
## 35	72.2834000	1	<NA>
## 36	118.3132600	0	<NA>
## 37	74.4953700	0	<NA>
## 38	122.2556297	0	<NA>
## 39	81.9465363	0	<NA>
## 40	83.9793260	0	<NA>
## 41	83.9776270	0	<NA>
## 42	-71.2140031	1	<NA>
## 43	-73.5108500	0	<NA>
## 44	-68.7187959	0	<NA>
## 45	24.0294800	0	<NA>
## 46	-75.3047800	1	<NA>
## 47	-122.3855630	0	<NA>
## 48	-75.5955743	2	<NA>
## 49	-71.0770756	1	<NA>
## 50	-118.3110490	0	<NA>
## 51	-80.0607100	0	<NA>
## 52	-124.0856000	0	<NA>
## 53	-71.4415782	4	<NA>
## 54	158.0700000	0	<NA>
## 55	93.4807080	0	<NA>
## 56	74.3285850	0	<NA>
## 57	76.3324000	0	<NA>
## 58	72.7170000	0	<NA>
## 59	106.6007110	0	<NA>
## 60	80.1288000	0	<NA>
## 61	-27.2046000	0	<NA>
## 62	-37.3412000	0	<NA>
## 63	-70.9106390	1	<NA>
## 64	-19.2695200	0	<NA>
## 65	-35.9535280	0	<NA>
## 66	-77.4977000	0	<NA>
## 67	-77.1169820	0	<NA>
## 68	36.2000000	0	<NA>
## 69	-71.3852600	17	<NA>
## 70	-36.0832400	0	<NA>
## 71	-35.9152000	0	<NA>
## 72	-77.4977000	0	<NA>
## 73	-39.4536000	0	<NA>
## 74	-83.0780560	0	<NA>
## 75	-71.2974662	3	<NA>
## 76	105.1372222	0	<NA>
## 77	123.1543400	0	<NA>
## 78	17.0949088	0	<NA>
## 79	4.7675100	0	<NA>
## 80	10.2561600	0	<NA>
## 81	-2.9709560	0	<NA>
## 82	-93.4644260	0	<NA>
## 83	80.1096700	0	<NA>

## 84	-82.7932962	2	<NA>
## 85	-66.0298581	0	<NA>
## 86	-71.1415070	1	<NA>
## 87	-74.0100272	0	<NA>
## 88	-149.9231257	0	<NA>
## 89	-82.7896022	1	<NA>
## 90	-2.1026500	0	<NA>
## 91	-77.2937000	0	<NA>
## 92	-86.3333000	0	<NA>
## 93	-70.1122000	1	<NA>
## 94	-71.4186000	4	<NA>
## 95	-121.9473000	0	<NA>
## 96	-71.1238300	2	<NA>
## 97	95.2731110	0	<NA>
## 98	-71.4356513	4	<NA>
## 99	-52.9851000	0	<NA>
## 100	90.0659700	0	<NA>
## 101	-34.1171000	0	<NA>
## 102	84.4977000	0	<NA>
## 103	9.5920000	0	<NA>
## 104	-42.0200700	0	<NA>
## 105	-151.5403420	0	<NA>
## 106	122.3018900	0	<NA>
## 107	72.5801900	0	<NA>
## 108	123.1385100	0	<NA>
## 109	93.3100000	0	<NA>
## 110	117.1111700	0	<NA>
## 111	0.0000000	0	<NA>
## 112	70.3919440	1	<NA>
## 113	80.4410000	0	<NA>
## 114	-71.3075090	0	<NA>
## 115	84.5202528	0	<NA>
## 116	32.8857000	0	<NA>
## 117	-71.4234195	2	<NA>
## 118	20.8680769	0	<NA>
## 119	122.7851040	0	<NA>
## 120	-75.5630000	0	<NA>
## 121	90.1502700	0	<NA>
## 122	116.6913200	0	<NA>
## 123	-71.0492320	0	<NA>

Some of the rows are for foreign libraries, but it look like the majority are libraries with no street entries. Some like charter number G10014(148 Marina Plaza Dunedin) are located very close to the ocean and hence were classified due to the resolution of the geography. A few like 150219 (1710 S Trenon Ave Tulsa) are mislabeled with coordinates not in the Unites States. Luckily there are only 123 entries, so we can filter them out without major concern.

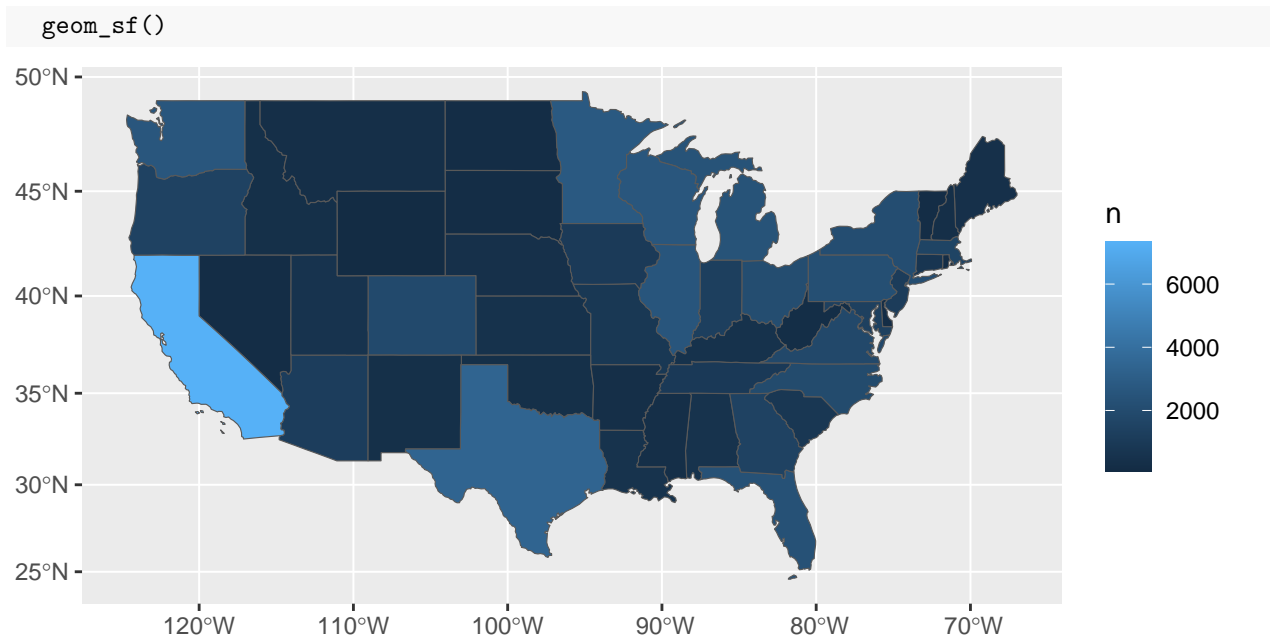
```
libraries <- libraries %>% drop_na(state)
```

Finally we can do some analysis with states:

```
state_counts <- libraries %>% group_by(state) %>% count %>% arrange(desc(n))
state_counts <- state_counts %>% rename("NAME" = "state")
```

```
st_transform(us_states, crs = 3857) %>% full_join(state_counts, by = join_by(NAME)) %>%
  ggplot(aes(fill = n)) +
```





Research Q's:

Is there a correlation between income in a given zip code and the number of little free libraries? What about political party preference? How about climate (i.e. colder areas might have less LFLs which are outdoors by design)?