

Preperation code for Mapbox lab (Solution)

Written by Raechel Portelli November 10, 2021

Add a line to import geopandas, pandas, and numpy

```
In [1]: import geopandas as gp
import pandas as pd
import numpy as np
```

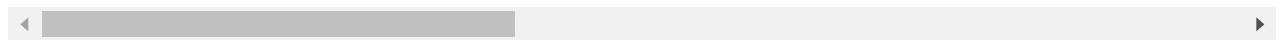
Add a line to read in the CSV file and call it tri2019

```
In [3]: tri2019 = gp.read_file("C:/Users/raechel/tri_2019_us.csv")
```

Out[3]:

	1. YEAR	2. TRIFD	3. FRS ID	4. FACILITY NAME	5. STREET ADDRESS	6. CITY	7. CC
0	2019	98421SLRFN3001M	1.1E+11	US OIL & REFINING CO	3001 MARSHALL AVE	TACOMA	
1	2019	46808DRLBR4611D	1.10001E+11	DR LUBRICANTS INC	4611-D NEWAYGO RD	FORT WAYNE	
2	2019	41105RMCSHUSROU	1.10067E+11	AK STEEL CORP	170 ARMCO ROAD	ASHLAND	
3	2019	41105RMCSHUSROU	1.10067E+11	AK STEEL CORP	170 ARMCO ROAD	ASHLAND	
4	2019	19934CMDLMRTE10	1.10042E+11	HANDYTUBE CORP	124 VEPCO BOULEVARD	CAMDEN	
...
79414	2019	22801WMPLR590MT	1.10002E+11	PILGRIM'S PRIDE CORP HARRISONBURG FEEDMILL	590 MT CLINTON PIKE	HARRISONBURG	HARRISON
79415	2019	50213PLMLL1000T	1.1E+11	PAUL MUELLER CO	1715 TIEKEN DR	OSCEOLA	C
79416	2019	24124HCHSTRT460	1.10057E+11	CELANESE ACETATE LLC - CELCO P LANT	3520 VIRGINIA AVE	NARROWS	
79417	2019	80916NCRMD1635A	1.1E+11	DPIX LLC	1635 AEROPLAZA DR	COLORADO SPRINGS	EI
79418	2019	54311FCNTR3913A	1.10002E+11	FEECO INTERNATIONAL	3913 ALGOMA RD	GREEN BAY	B

79419 rows × 119 columns



Obtain a list of column names for the TRI dataset

In [4]:

```
for col in tri2019.columns:  
    print(col)
```

```
1. YEAR  
2. TRIFD  
3. FRS ID  
4. FACILITY NAME  
5. STREET ADDRESS  
6. CITY  
7. COUNTY  
8. ST  
9. ZIP  
10. BIA  
11. TRIBE  
12. LATITUDE  
13. LONGITUDE  
14. HORIZONTAL DATUM  
15. PARENT CO NAME  
16. PARENT CO DB NUM  
17. STANDARD PARENT CO NAME  
18. FEDERAL FACILITY  
19. INDUSTRY SECTOR CODE  
20. INDUSTRY SECTOR  
21. PRIMARY SIC  
22. SIC 2  
23. SIC 3  
24. SIC 4  
25. SIC 5  
26. SIC 6  
27. PRIMARY NAICS  
28. NAICS 2  
29. NAICS 3  
30. NAICS 4  
31. NAICS 5  
32. NAICS 6  
33. DOC_CTRL_NUM  
34. CHEMICAL  
35. ELEMENTAL METAL INCLUDED  
36. TRI CHEMICAL/COMPOUND ID  
37. CAS#  
38. SRS ID  
39. CLEAN AIR ACT CHEMICAL  
40. CLASSIFICATION  
41. METAL  
42. METAL CATEGORY  
43. CARCINOGEN  
44. PFAS  
45. FORM TYPE  
46. UNIT OF MEASURE  
47. 5.1 - FUGITIVE AIR  
48. 5.2 - STACK AIR  
49. 5.3 - WATER  
50. 5.4 - UNDERGROUND  
51. 5.4.1 - UNDERGROUND CL I
```

52. 5.4.2 - UNDERGROUND C II-V
53. 5.5.1 - LANDFILLS
54. 5.5.1A - RCRA C LANDFILL
55. 5.5.1B - OTHER LANDFILLS
56. 5.5.2 - LAND TREATMENT
57. 5.5.3 - SURFACE IMPNDMNT
58. 5.5.3A - RCRA SURFACE IM
59. 5.5.3B - OTHER SURFACE I
60. 5.5.4 - OTHER DISPOSAL
61. ON-SITE RELEASE TOTAL
62. 6.1 - POTW - TRNS RLSE
63. 6.1 - POTW - TRNS TRT
64. POTW - TOTAL TRANSFERS
65. 6.2 - M10
66. 6.2 - M41
67. 6.2 - M62
68. 6.2 - M40 METAL
69. 6.2 - M61 METAL
70. 6.2 - M71
71. 6.2 - M81
72. 6.2 - M82
73. 6.2 - M72
74. 6.2 - M63
75. 6.2 - M66
76. 6.2 - M67
77. 6.2 - M64
78. 6.2 - M65
79. 6.2 - M73
80. 6.2 - M79
81. 6.2 - M90
82. 6.2 - M94
83. 6.2 - M99
84. OFF-SITE RELEASE TOTAL
85. 6.2 - M20
86. 6.2 - M24
87. 6.2 - M26
88. 6.2 - M28
89. 6.2 - M93
90. OFF-SITE RECYCLED TOTAL
91. 6.2 - M56
92. 6.2 - M92
93. OFF-SITE ENERGY RECOVERY T
94. 6.2 - M40 NON-METAL
95. 6.2 - M50
96. 6.2 - M54
97. 6.2 - M61 NON-METAL
98. 6.2 - M69
99. 6.2 - M95
100. OFF-SITE TREATED TOTAL
101. 6.2 - UNCLASSIFIED
102. 6.2 - TOTAL TRANSFER
103. TOTAL RELEASES
104. 8.1 - RELEASES
105. 8.1A - ON-SITE CONTAINED
106. 8.1B - ON-SITE OTHER
107. 8.1C - OFF-SITE CONTAIN
108. 8.1D - OFF-SITE OTHER R
109. 8.2 - ENERGY RECOVER ON
110. 8.3 - ENERGY RECOVER OF
111. 8.4 - RECYCLING ON SITE
112. 8.5 - RECYCLING OFF SIT

113. 8.6 - TREATMENT ON SITE
114. 8.7 - TREATMENT OFF SITE
115. PRODUCTION WSTE (8.1-8.7)
116. 8.8 - ONE-TIME RELEASE
117. PROD_RATIO_OR_ ACTIVITY
118. 8.9 - PRODUCTION RATIO
geometry

Examine the properties for attribute "100. OFF-SITE TREATED TOTAL." This value represents the total quantity of chemicals reported as transferred offsite for treatment. (See Data File Documentation on the website for details about other attributes).

Next, create a subset of the dataset using the following properties: Facility Name, City, ST, the OFF-SITE TREATED TOTAL and the geographic coordinates and call it tri2019_subset

```
In [25]: tri2019_subset = tri2019[["4. FACILITY NAME", "6. CITY", "8. ST", "12. LATITUDE", "13. L
```

How many rows does the dataframe have? 79419

```
In [26]: tri2019_subset
```

Out[26]:

	4. FACILITY NAME	6. CITY	8. ST	12. LATITUDE	13. LONGITUDE	100. OFF-SITE TREATED TOTAL
0	US OIL & REFINING CO	TACOMA	WA	47.257694	-122.395259	2.2
1	DR LUBRICANTS INC	FORT WAYNE	IN	41.11999	-85.15772	0
2	AK STEEL CORP	ASHLAND	KY	38.498333	-82.663611	0
3	AK STEEL CORP	ASHLAND	KY	38.498333	-82.663611	0
4	HANDYTUBE CORP	CAMDEN	DE	39.10974	-75.55563	0
...
79414	PILGRIM'S PRIDE CORP HARRISONBURG FEEDMILL	HARRISONBURG	VA	38.468592	-78.871151	0
79415	PAUL MUELLER CO	OSCEOLA	IA	41.02471	-93.79055	0
79416	CELANESE ACETATE LLC - CELCO P LANT	NARROWS	VA	37.3451	-80.76293	0
79417	DPIX LLC	COLORADO SPRINGS	CO	38.80957	-104.73373	0
79418	FEECO INTERNATIONAL	GREEN BAY	WI	44.5394	-87.886	0

79419 rows × 6 columns

Find the unique cities within the dataset and write this list to a csv file to open in excel. How many cities are represented in the dataset? 5259

```
In [23]: uniqueCities = pd.unique(tri2019_subset["6. CITY"])
```

```
In [24]: pd.DataFrame(uniqueCities).to_csv("file.csv")
```

Next, subset the data based on a state you are interested in. If you need help with subsetting based on rows, see this page. <https://stackoverflow.com/questions/17071871/how-do-i-select-rows-from-a-dataframe-based-on-column-values>

```
In [32]:
```

```
Out[32]:
```

	4. FACILITY NAME	6. CITY	8. ST	12. LATITUDE	13. LONGITUDE	100. OFF-SITE TREATED TOTAL
304	FEDERAL CARTRIDGE CO-CCI	LEWISTON	ID	46.396389	-117.035004	0
317	BIG TEX TRAILER MANUFACTURING INC-CALDWELL	CALDWELL	ID	43.670219	-116.701127	0
946	CS BEEF PACKERS	KUNA	ID	43.44488	-116.27435	0
979	POTLATCHDELTIC LAND & LUMBER LLC ST MARIES COM...	SAINT MARIES	ID	47.323441	-116.586195	0
1030	US SILVER - IDAHO INC	WALLACE	ID	47.48289	-115.961552	0
...
78494	TRINITY TRAILER MANUFACTURING INC	BOISE	ID	43.525979	-116.153474	0
78727	COLEMAN OIL CO	LEWISTON	ID	46.421089	-116.982239	0
78792	ROCKY MOUNTAIN HARDWARE INC (SHOSHONE)	SHOSHONE	ID	42.936491	-114.398042	0
78959	HART & COOLEY LLC	NAMPA	ID	43.60172	-116.54829	0
79102	BASIC AMERICAN FOODS REXBURG FACILITY	REXBURG	ID	43.838063	-111.783121	0

395 rows × 6 columns

Export the file as a CSV file. There is no reliable direct way to conver the CSV to a geojson in python that doesn't require somewhat difficult to understand code, so you will use a different tool to convert your CSV file to a geojson.

```
In [ ]:
```