In [261…

```
using Pkg
using DataFrames
using PyCall
# Pkg.add("CSV")
using CSV
```

In [262…

```
py"""
# for scraping call beautifulsoup
from bs4 import BeautifulSoup
import requests
import pandas as pd
import lxml


# https://www.huduser.gov/portal/datasets/fmr/fmrs/FY2023_code/2023summary.odn?fips=131219

URL = 'https://www.huduser.gov/portal/datasets/fmr/fmrs/FY2023_code/2023summary.odn?fips=1

page = requests.get(URL)

soup = BeautifulSoup(page.content, 'html.parser')

results = soup.find('table', class_='big_table')
df = pd.read_html(results.prettify())
# export it to a csv file
df[0].to_csv('fmr.csv')
"""
```

In [263…

```
# read in the csv file
df = CSV.read("fmr.csv", DataFrame)
```

302×7 DataFrame *277 rows omitted*

Row	Column1	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_1	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_2	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_3	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_4	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_5
	Int64?	String15	String15	String15	String15	String15	String15
1	*missing*	ZIP Code	Efficiency	One-Bedroom	Two-Bedroom	Three-Bedroom	Four-Bedroom
2	0	30002	$940	$960	$1,080	$1,310	$1,610
3	1	30003	$1,460	$1,500	$1,690	$2,060	$2,510
4	2	30004	$1,590	$1,630	$1,840	$2,240	$2,730
5	3	30005	$1,730	$1,770	$2,000	$2,430	$2,970
6	4	30006	$1,390	$1,430	$1,610	$1,960	$2,390
7	5	30007	$1,390	$1,430	$1,610	$1,960	$2,390
8	6	30008	$1,200	$1,220	$1,380	$1,680	$2,050
9	7	30009	$1,700	$1,740	$1,960	$2,390	$2,910
10	8	30010	$1,460	$1,500	$1,690	$2,060	$2,510

Row	Column1	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_1	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_2	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_3	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_4	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_5
	Int64?	String15	String15	String15	String15	String15	String15
11	9	30011	$1,330	$1,360	$1,540	$1,870	$2,290
12	10	30012	$1,110	$1,130	$1,280	$1,560	$1,900
13	11	30013	$1,260	$1,290	$1,460	$1,780	$2,170
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
291	289	31106	$1,410	$1,440	$1,630	$1,980	$2,420
292	290	31107	$1,350	$1,380	$1,560	$1,900	$2,320
293	291	31119	$1,350	$1,380	$1,560	$1,900	$2,320
294	292	31126	$1,410	$1,440	$1,630	$1,980	$2,420
295	293	31131	$1,410	$1,440	$1,630	$1,980	$2,420
296	294	31139	$1,390	$1,430	$1,610	$1,960	$2,390
297	295	31141	$1,350	$1,380	$1,560	$1,900	$2,320
298	296	31145	$1,350	$1,380	$1,560	$1,900	$2,320
299	297	31146	$1,350	$1,380	$1,560	$1,900	$2,320
300	298	31150	$1,410	$1,440	$1,630	$1,980	$2,420
301	299	31156	$1,410	$1,440	$1,630	$1,980	$2,420
302	300	39901	$1,350	$1,380	$1,560	$1,900	$2,320

In [264…

```
# drop the first column
df = df[:, 2:end]
```

302×6 DataFrame *277 rows omitted*

Row	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_1	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_2	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_3	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_4	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_5
	String15	String15	String15	String15	String15	String15
1	ZIP Code	Efficiency	One-Bedroom	Two-Bedroom	Three-Bedroom	Four-Bedroom
2	30002	$940	$960	$1,080	$1,310	$1,610
3	30003	$1,460	$1,500	$1,690	$2,060	$2,510
4	30004	$1,590	$1,630	$1,840	$2,240	$2,730
5	30005	$1,730	$1,770	$2,000	$2,430	$2,970
6	30006	$1,390	$1,430	$1,610	$1,960	$2,390

Row	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_1	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_2	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_3	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_4	Atlanta-Sandy Springs-Roswell, GA HUD Metro FMR Area Small Area FMRs By Unit Bedrooms_5
	String15	String15	String15	String15	String15	String15
7	30007	$1,390	$1,430	$1,610	$1,960	$2,390
8	30008	$1,200	$1,220	$1,380	$1,680	$2,050
9	30009	$1,700	$1,740	$1,960	$2,390	$2,910
10	30010	$1,460	$1,500	$1,690	$2,060	$2,510
11	30011	$1,330	$1,360	$1,540	$1,870	$2,290
12	30012	$1,110	$1,130	$1,280	$1,560	$1,900
13	30013	$1,260	$1,290	$1,460	$1,780	$2,170
⋮	⋮	⋮	⋮	⋮	⋮	⋮
291	31106	$1,410	$1,440	$1,630	$1,980	$2,420
292	31107	$1,350	$1,380	$1,560	$1,900	$2,320
293	31119	$1,350	$1,380	$1,560	$1,900	$2,320
294	31126	$1,410	$1,440	$1,630	$1,980	$2,420
295	31131	$1,410	$1,440	$1,630	$1,980	$2,420
296	31139	$1,390	$1,430	$1,610	$1,960	$2,390
297	31141	$1,350	$1,380	$1,560	$1,900	$2,320
298	31145	$1,350	$1,380	$1,560	$1,900	$2,320
299	31146	$1,350	$1,380	$1,560	$1,900	$2,320
300	31150	$1,410	$1,440	$1,630	$1,980	$2,420
301	31156	$1,410	$1,440	$1,630	$1,980	$2,420
302	39901	$1,350	$1,380	$1,560	$1,900	$2,320

In [265…

```
df
rename!(df, [:ZIP, :EFFICIENCY, :ONE_BEDROOM, :TWO_BEDROOM, :THREE_BEDROOM, :FOUR_BEDROOM,

# remove first row
df = df[2:end, :]
```

301×6 DataFrame *276 rows omitted*

Row	ZIP	EFFICIENCY	ONE_BEDROOM	TWO_BEDROOM	THREE_BEDROOM	FOUR_BEDROOM
	String15	String15	String15	String15	String15	String15
1	30002	$940	$960	$1,080	$1,310	$1,610
2	30003	$1,460	$1,500	$1,690	$2,060	$2,510
3	30004	$1,590	$1,630	$1,840	$2,240	$2,730
4	30005	$1,730	$1,770	$2,000	$2,430	$2,970
5	30006	$1,390	$1,430	$1,610	$1,960	$2,390

Row	ZIP	EFFICIENCY	ONE_BEDROOM	TWO_BEDROOM	THREE_BEDROOM	FOUR_BEDROOM
	String15	String15	String15	String15	String15	String15
6	30007	$1,390	$1,430	$1,610	$1,960	$2,390
7	30008	$1,200	$1,220	$1,380	$1,680	$2,050
8	30009	$1,700	$1,740	$1,960	$2,390	$2,910
9	30010	$1,460	$1,500	$1,690	$2,060	$2,510
10	30011	$1,330	$1,360	$1,540	$1,870	$2,290
11	30012	$1,110	$1,130	$1,280	$1,560	$1,900
12	30013	$1,260	$1,290	$1,460	$1,780	$2,170
13	30014	$1,060	$1,080	$1,220	$1,480	$1,810
⋮	⋮	⋮	⋮	⋮	⋮	⋮
290	31106	$1,410	$1,440	$1,630	$1,980	$2,420
291	31107	$1,350	$1,380	$1,560	$1,900	$2,320
292	31119	$1,350	$1,380	$1,560	$1,900	$2,320
293	31126	$1,410	$1,440	$1,630	$1,980	$2,420
294	31131	$1,410	$1,440	$1,630	$1,980	$2,420
295	31139	$1,390	$1,430	$1,610	$1,960	$2,390
296	31141	$1,350	$1,380	$1,560	$1,900	$2,320
297	31145	$1,350	$1,380	$1,560	$1,900	$2,320
298	31146	$1,350	$1,380	$1,560	$1,900	$2,320
299	31150	$1,410	$1,440	$1,630	$1,980	$2,420
300	31156	$1,410	$1,440	$1,630	$1,980	$2,420
301	39901	$1,350	$1,380	$1,560	$1,900	$2,320

In [266…

```
# convert each column to integers

df[!, :ZIP] = parse.(Int64, df[!, :ZIP])
# remove the $ sign from the strings

df

# replace $ and ,
columns_names = ["EFFICIENCY",  "ONE_BEDROOM",  "TWO_BEDROOM"   ,"THREE_BEDROOM"         ,'

for i in columns_names
    df[!, i] = replace.(df[!, i], r"\$" => "")
    df[!, i] = replace.(df[!, i], r"\," => "")
    df[!, i] = parse.(Int64, df[!, i])
end
df
```

301×6 DataFrame *276 rows omitted*

Row	ZIP	EFFICIENCY	ONE_BEDROOM	TWO_BEDROOM	THREE_BEDROOM	FOUR_BEDROOM
	Int64	Int64	Int64	Int64	Int64	Int64

Row	ZIP	EFFICIENCY	ONE_BEDROOM	TWO_BEDROOM	THREE_BEDROOM	FOUR_BEDROOM
	Int64	Int64	Int64	Int64	Int64	Int64
1	30002	940	960	1080	1310	1610
2	30003	1460	1500	1690	2060	2510
3	30004	1590	1630	1840	2240	2730
4	30005	1730	1770	2000	2430	2970
5	30006	1390	1430	1610	1960	2390
6	30007	1390	1430	1610	1960	2390
7	30008	1200	1220	1380	1680	2050
8	30009	1700	1740	1960	2390	2910
9	30010	1460	1500	1690	2060	2510
10	30011	1330	1360	1540	1870	2290
11	30012	1110	1130	1280	1560	1900
12	30013	1260	1290	1460	1780	2170
13	30014	1060	1080	1220	1480	1810
⋮	⋮	⋮	⋮	⋮	⋮	⋮
290	31106	1410	1440	1630	1980	2420
291	31107	1350	1380	1560	1900	2320
292	31119	1350	1380	1560	1900	2320
293	31126	1410	1440	1630	1980	2420
294	31131	1410	1440	1630	1980	2420
295	31139	1390	1430	1610	1960	2390
296	31141	1350	1380	1560	1900	2320
297	31145	1350	1380	1560	1900	2320
298	31146	1350	1380	1560	1900	2320
299	31150	1410	1440	1630	1980	2420
300	31156	1410	1440	1630	1980	2420
301	39901	1350	1380	1560	1900	2320

In [267…

```
# install pgeocode from pycall


pg = pyimport_conda("pgeocode", "pgeocode")
nomi = pg.Nominatim("us")

# get the latitude and longitude for each zip code
lat = []
long = []
for i in df[!, :ZIP]
    lat = push!(lat, nomi.query_postal_code(i).latitude)
    long = push!(long, nomi.query_postal_code(i).longitude)
end
```

```
# add the latitude and longitude to the dataframe
df[!, :LAT] = lat
df[!, :LONG] = long


df
```

301×8 DataFrame *276 rows omitted*

Row	ZIP	EFFICIENCY	ONE_BEDROOM	TWO_BEDROOM	THREE_BEDROOM	FOUR_BEDROOM	LAT	LONG
	Int64	Int64	Int64	Int64	Int64	Int64	Any	Any
1	30002	940	960	1080	1310	1610	33.7717	-84.260
2	30003	1460	1500	1690	2060	2510	33.9604	-84.037
3	30004	1590	1630	1840	2240	2730	34.1124	-84.302
4	30005	1730	1770	2000	2430	2970	34.0782	-84.228
5	30006	1390	1430	1610	1960	2390	33.9526	-84.549
6	30007	1390	1430	1610	1960	2390	33.9125	-84.557
7	30008	1200	1220	1380	1680	2050	33.8972	-84.592
8	30009	1700	1740	1960	2390	2910	34.077	-84.303
9	30010	1460	1500	1690	2060	2510	33.9604	-84.037
10	30011	1330	1360	1540	1870	2290	34.0191	-83.826
11	30012	1110	1130	1280	1560	1900	33.7192	-84.002
12	30013	1260	1290	1460	1780	2170	33.6436	-83.968
13	30014	1060	1080	1220	1480	1810	33.5293	-83.849
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
290	31106	1410	1440	1630	1980	2420	33.8444	-84.474
291	31107	1350	1380	1560	1900	2320	33.8444	-84.474
292	31119	1350	1380	1560	1900	2320	33.8913	-84.074
293	31126	1410	1440	1630	1980	2420	33.8444	-84.474
294	31131	1410	1440	1630	1980	2420	33.8444	-84.474
295	31139	1390	1430	1610	1960	2390	33.8444	-84.474
296	31141	1350	1380	1560	1900	2320	33.8913	-84.074
297	31145	1350	1380	1560	1900	2320	33.8913	-84.074
298	31146	1350	1380	1560	1900	2320	33.8913	-84.074
299	31150	1410	1440	1630	1980	2420	33.8444	-84.474
300	31156	1410	1440	1630	1980	2420	33.8444	-84.474
301	39901	1350	1380	1560	1900	2320	33.8913	-84.074

In [268…

```
# export the dataframe to a csv file
CSV.write("fmr_plottable.csv", df)

# call the plotty.py file from the command line
run(`python plotty.py`)
```

```
# remove the csv file
rm("fmr_plottable.csv")
```