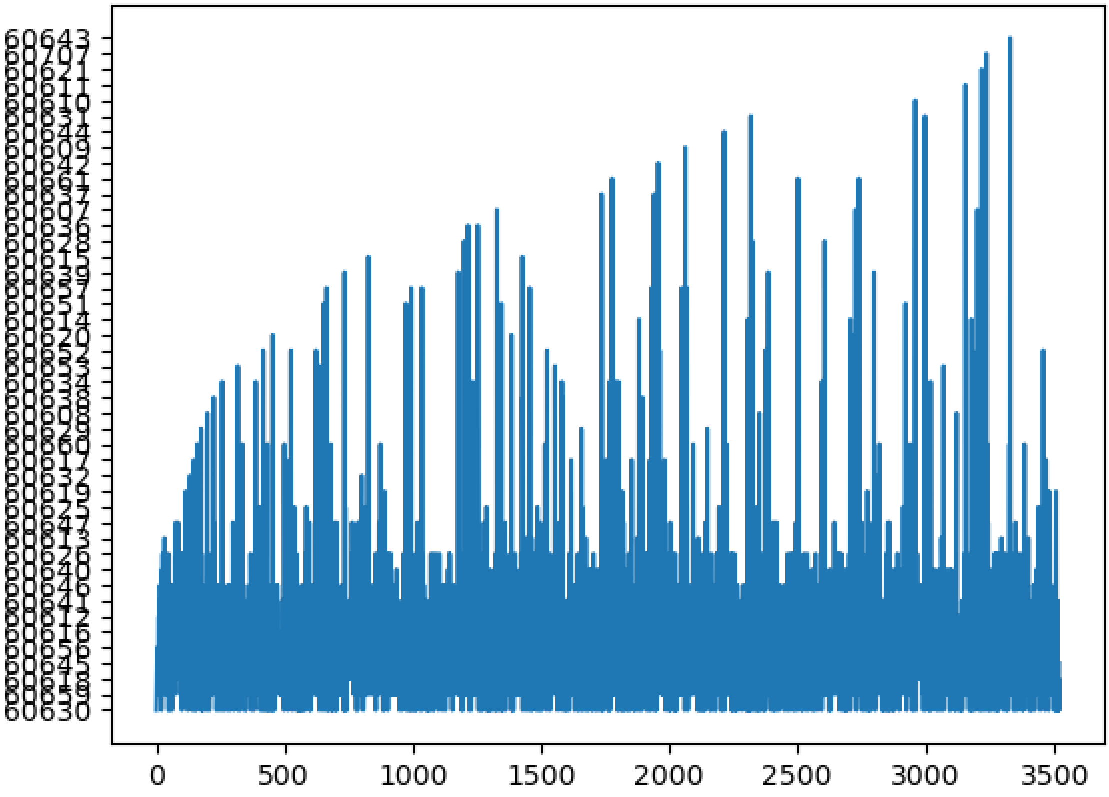
In [19]:
```
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
taxi_owner = pd.read_pickle('taxi_owners.p')
taxi_owner.head()
plt.plot(taxi_owner.zip)
```

Out[19]:
```
[<matplotlib.lines.Line2D at 0x7fc67ae36eb0>]
```



In [13]:
```
print(taxi_owner)
```

```
        rid   vid             owner                   address    zip
0     T6285  6285    AGEAN TAXI LLC     4536 N. ELSTON AVE.  60630
1     T4862  4862     MANGIB CORP.  5717 N. WASHTENAW AVE.  60659
2     T1495  1495     FUNRIDE, INC.     3351 W. ADDISON ST.  60618
3     T4231  4231     ALQUSH CORP.   6611 N. CAMPBELL AVE.  60645
4     T5971  5971    EUNIFFORD INC.     3351 W. ADDISON ST.  60618
...     ...   ...               ...                     ...    ...
3514  T4453  4453   IMAGIN CAB CORP     3351 W. ADDISON ST.  60618
3515   T121   121  TRIBECA CAB CORP     4536 N. ELSTON AVE.  60630
3516  T3465  3465  AMIR EXPRESS INC     3351 W. ADDISON ST.  60618
3517  T1962  1962  KARY CAB COMPANY     4707 N. KENTON AVE.  60630
3518  T1031  1031       NECT 42 LLC    6500 N. WESTERN AVE.  60645

[3519 rows x 5 columns]
```

In [7]:
```
taxi_owner.describe()
```

Out[7]:

	rid	vid	owner	address	zip
count	3519	3519	3519	3519	3519
unique	3519	3519	2375	317	44
top	T6285	6285	CHICAGO SEVEN INC	3351 W. ADDISON ST.	60618
freq	1	1	21	639	798

In [9]:

```
taxi_owner.shape
```

Out[9]:

```
(3519, 5)
```

In [10]:

```
taxi_owner.values
```

Out[10]:

```
array([['T6285', '6285', 'AGEAN TAXI LLC', '4536 N. ELSTON AVE.',
        '60630'],
       ['T4862', '4862', 'MANGIB CORP.', '5717 N. WASHTENAW AVE.',
        '60659'],
       ['T1495', '1495', 'FUNRIDE, INC.', '3351 W. ADDISON ST.', '60618'],
       ...,
       ['T3465', '3465', 'AMIR EXPRESS INC', '3351 W. ADDISON ST.',
        '60618'],
       ['T1962', '1962', 'KARY CAB COMPANY', '4707 N. KENTON AVE.',
        '60630'],
       ['T1031', '1031', 'NECT 42 LLC', '6500 N. WESTERN AVE.', '60645']],
      dtype=object)
```

In [11]:

```
taxi_owner.columns
```

Out[11]:

```
Index(['rid', 'vid', 'owner', 'address', 'zip'], dtype='object')
```

In [12]:

```
taxi_owner.index
```

Out[12]:

```
RangeIndex(start=0, stop=3519, step=1)
```

In [20]:

```
homeless = pd.read_csv("homelessness.csv")
homeless.head()
```

Out[20]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
0	0	East South Central	Alabama	2570.0	864.0	4887681
1	1	Pacific	Alaska	1434.0	582.0	735139
2	2	Mountain	Arizona	7259.0	2606.0	7158024
3	3	West South Central	Arkansas	2280.0	432.0	3009733
4	4	Pacific	California	109008.0	20964.0	39461588

In [21]:

```
homeless.describe() #compute some summary statics for numerical columns like
```

Out[21]:

	Unnamed: 0	individuals	family_members	state_pop
count	51.000000	51.000000	51.000000	5.100000e+01
mean	25.000000	7225.784314	3504.882353	6.405637e+06
std	14.866069	15991.025083	7805.411811	7.327258e+06
min	0.000000	434.000000	75.000000	5.776010e+05
25%	12.500000	1446.500000	592.000000	1.777414e+06
50%	25.000000	3082.000000	1482.000000	4.461153e+06
75%	37.500000	6781.500000	3196.000000	7.340946e+06
max	50.000000	109008.000000	52070.000000	3.946159e+07

In [22]:

```
homeless.shape     #the number of rows and columns
```

Out[22]:

```
(51, 6)
```

In [23]:

```
homeless.values       #the data value in 2-d Numpy array
```

```
Out[23]: array([[0, 'East South Central', 'Alabama', 2570.0, 864.0, 4887681],
                [1, 'Pacific', 'Alaska', 1434.0, 582.0, 735139],
                [2, 'Mountain', 'Arizona', 7259.0, 2606.0, 7158024],
                [3, 'West South Central', 'Arkansas', 2280.0, 432.0, 3009733],
                [4, 'Pacific', 'California', 109008.0, 20964.0, 39461588],
                [5, 'Mountain', 'Colorado', 7607.0, 3250.0, 5691287],
                [6, 'New England', 'Connecticut', 2280.0, 1696.0, 3571520],
                [7, 'South Atlantic', 'Delaware', 708.0, 374.0, 965479],
                [8, 'South Atlantic', 'District of Columbia', 3770.0, 3134.0,
                 701547],
                [9, 'South Atlantic', 'Florida', 21443.0, 9587.0, 21244317],
                [10, 'South Atlantic', 'Georgia', 6943.0, 2556.0, 10511131],
                [11, 'Pacific', 'Hawaii', 4131.0, 2399.0, 1420593],
                [12, 'Mountain', 'Idaho', 1297.0, 715.0, 1750536],
                [13, 'East North Central', 'Illinois', 6752.0, 3891.0, 12723071],
                [14, 'East North Central', 'Indiana', 3776.0, 1482.0, 6695497],
                [15, 'West North Central', 'Iowa', 1711.0, 1038.0, 3148618],
                [16, 'West North Central', 'Kansas', 1443.0, 773.0, 2911359],
                [17, 'East South Central', 'Kentucky', 2735.0, 953.0, 4461153],
                [18, 'West South Central', 'Louisiana', 2540.0, 519.0, 4659690],
                [19, 'New England', 'Maine', 1450.0, 1066.0, 1339057],
                [20, 'South Atlantic', 'Maryland', 4914.0, 2230.0, 6035802],
                [21, 'New England', 'Massachusetts', 6811.0, 13257.0, 6882635],
                [22, 'East North Central', 'Michigan', 5209.0, 3142.0, 9984072],
                [23, 'West North Central', 'Minnesota', 3993.0, 3250.0, 5606249],
                [24, 'East South Central', 'Mississippi', 1024.0, 328.0, 2981020],
                [25, 'West North Central', 'Missouri', 3776.0, 2107.0, 6121623],
                [26, 'Mountain', 'Montana', 983.0, 422.0, 1060665],
                [27, 'West North Central', 'Nebraska', 1745.0, 676.0, 1925614],
                [28, 'Mountain', 'Nevada', 7058.0, 486.0, 3027341],
                [29, 'New England', 'New Hampshire', 835.0, 615.0, 1353465],
                [30, 'Mid-Atlantic', 'New Jersey', 6048.0, 3350.0, 8886025],
                [31, 'Mountain', 'New Mexico', 1949.0, 602.0, 2092741],
                [32, 'Mid-Atlantic', 'New York', 39827.0, 52070.0, 19530351],
                [33, 'South Atlantic', 'North Carolina', 6451.0, 2817.0, 10381615],
                [34, 'West North Central', 'North Dakota', 467.0, 75.0, 758080],
                [35, 'East North Central', 'Ohio', 6929.0, 3320.0, 11676341],
                [36, 'West South Central', 'Oklahoma', 2823.0, 1048.0, 3940235],
                [37, 'Pacific', 'Oregon', 11139.0, 3337.0, 4181886],
                [38, 'Mid-Atlantic', 'Pennsylvania', 8163.0, 5349.0, 12800922],
                [39, 'New England', 'Rhode Island', 747.0, 354.0, 1058287],
                [40, 'South Atlantic', 'South Carolina', 3082.0, 851.0, 5084156],
                [41, 'West North Central', 'South Dakota', 836.0, 323.0, 878698],
                [42, 'East South Central', 'Tennessee', 6139.0, 1744.0, 6771631],
                [43, 'West South Central', 'Texas', 19199.0, 6111.0, 28628666],
                [44, 'Mountain', 'Utah', 1904.0, 972.0, 3153550],
                [45, 'New England', 'Vermont', 780.0, 511.0, 624358],
                [46, 'South Atlantic', 'Virginia', 3928.0, 2047.0, 8501286],
                [47, 'Pacific', 'Washington', 16424.0, 5880.0, 7523869],
                [48, 'South Atlantic', 'West Virginia', 1021.0, 222.0, 1804291],
                [49, 'East North Central', 'Wisconsin', 2740.0, 2167.0, 5807406],
                [50, 'Mountain', 'Wyoming', 434.0, 205.0, 577601]], dtype=object)
```

```
In [25]: homeless.columns    #contains column names
```

```
Out[25]: Index(['Unnamed: 0', 'region', 'state', 'individuals', 'family_members',
                'state_pop'],
               dtype='object')
```

```
In [26]: homeless.index    #row numbers or row names
```

```
Out[26]: RangeIndex(start=0, stop=51, step=1)
```

Sorting and Subnetting (Filtering)

In [34]:

```
homeless.sort_values("individuals", ascending = False)
```

Out[34]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
4	4	Pacific	California	109008.0	20964.0	39461588
32	32	Mid-Atlantic	New York	39827.0	52070.0	19530351
9	9	South Atlantic	Florida	21443.0	9587.0	21244317
43	43	West South Central	Texas	19199.0	6111.0	28628666
47	47	Pacific	Washington	16424.0	5880.0	7523869
37	37	Pacific	Oregon	11139.0	3337.0	4181886
38	38	Mid-Atlantic	Pennsylvania	8163.0	5349.0	12800922
5	5	Mountain	Colorado	7607.0	3250.0	5691287
2	2	Mountain	Arizona	7259.0	2606.0	7158024
28	28	Mountain	Nevada	7058.0	486.0	3027341
10	10	South Atlantic	Georgia	6943.0	2556.0	10511131
35	35	East North Central	Ohio	6929.0	3320.0	11676341
21	21	New England	Massachusetts	6811.0	13257.0	6882635
13	13	East North Central	Illinois	6752.0	3891.0	12723071
33	33	South Atlantic	North Carolina	6451.0	2817.0	10381615
42	42	East South Central	Tennessee	6139.0	1744.0	6771631
30	30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025
22	22	East North Central	Michigan	5209.0	3142.0	9984072
20	20	South Atlantic	Maryland	4914.0	2230.0	6035802
11	11	Pacific	Hawaii	4131.0	2399.0	1420593
23	23	West North Central	Minnesota	3993.0	3250.0	5606249
46	46	South Atlantic	Virginia	3928.0	2047.0	8501286
25	25	West North Central	Missouri	3776.0	2107.0	6121623
14	14	East North Central	Indiana	3776.0	1482.0	6695497
8	8	South Atlantic	District of Columbia	3770.0	3134.0	701547
40	40	South Atlantic	South Carolina	3082.0	851.0	5084156
36	36	West South Central	Oklahoma	2823.0	1048.0	3940235
49	49	East North Central	Wisconsin	2740.0	2167.0	5807406
17	17	East South Central	Kentucky	2735.0	953.0	4461153
0	0	East South Central	Alabama	2570.0	864.0	4887681

	Unnamed: 0	region	state	individuals	family_members	state_pop
18	18	West South Central	Louisiana	2540.0	519.0	4659690
6	6	New England	Connecticut	2280.0	1696.0	3571520
3	3	West South Central	Arkansas	2280.0	432.0	3009733
31	31	Mountain	New Mexico	1949.0	602.0	2092741
44	44	Mountain	Utah	1904.0	972.0	3153550
27	27	West North Central	Nebraska	1745.0	676.0	1925614
15	15	West North Central	Iowa	1711.0	1038.0	3148618
19	19	New England	Maine	1450.0	1066.0	1339057
16	16	West North Central	Kansas	1443.0	773.0	2911359
1	1	Pacific	Alaska	1434.0	582.0	735139
12	12	Mountain	Idaho	1297.0	715.0	1750536
24	24	East South Central	Mississippi	1024.0	328.0	2981020
48	48	South Atlantic	West Virginia	1021.0	222.0	1804291
26	26	Mountain	Montana	983.0	422.0	1060665
41	41	West North Central	South Dakota	836.0	323.0	878698
29	29	New England	New Hampshire	835.0	615.0	1353465
45	45	New England	Vermont	780.0	511.0	624358
39	39	New England	Rhode Island	747.0	354.0	1058287
7	7	South Atlantic	Delaware	708.0	374.0	965479
34	34	West North Central	North Dakota	467.0	75.0	758080
50	50	Mountain	Wyoming	434.0	205.0	577601

```
In [36]: homeless.sort_values(["individuals" , "state_pop"])
```

Out[36]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
50	50	Mountain	Wyoming	434.0	205.0	577601
34	34	West North Central	North Dakota	467.0	75.0	758080
7	7	South Atlantic	Delaware	708.0	374.0	965479
39	39	New England	Rhode Island	747.0	354.0	1058287
45	45	New England	Vermont	780.0	511.0	624358
29	29	New England	New Hampshire	835.0	615.0	1353465
41	41	West North Central	South Dakota	836.0	323.0	878698
26	26	Mountain	Montana	983.0	422.0	1060665
48	48	South Atlantic	West Virginia	1021.0	222.0	1804291
24	24	East South Central	Mississippi	1024.0	328.0	2981020
12	12	Mountain	Idaho	1297.0	715.0	1750536
1	1	Pacific	Alaska	1434.0	582.0	735139
16	16	West North Central	Kansas	1443.0	773.0	2911359
19	19	New England	Maine	1450.0	1066.0	1339057
15	15	West North Central	Iowa	1711.0	1038.0	3148618
27	27	West North Central	Nebraska	1745.0	676.0	1925614
44	44	Mountain	Utah	1904.0	972.0	3153550
31	31	Mountain	New Mexico	1949.0	602.0	2092741
3	3	West South Central	Arkansas	2280.0	432.0	3009733
6	6	New England	Connecticut	2280.0	1696.0	3571520
18	18	West South Central	Louisiana	2540.0	519.0	4659690
0	0	East South Central	Alabama	2570.0	864.0	4887681
17	17	East South Central	Kentucky	2735.0	953.0	4461153
49	49	East North Central	Wisconsin	2740.0	2167.0	5807406
36	36	West South Central	Oklahoma	2823.0	1048.0	3940235
40	40	South Atlantic	South Carolina	3082.0	851.0	5084156
8	8	South Atlantic	District of Columbia	3770.0	3134.0	701547
25	25	West North Central	Missouri	3776.0	2107.0	6121623
14	14	East North	Indiana	3776.0	1482.0	6695497

	Unnamed: 0	region	state	individuals	family_members	state_pop
		Central				
46	46	South Atlantic	Virginia	3928.0	2047.0	8501286
23	23	West North Central	Minnesota	3993.0	3250.0	5606249
11	11	Pacific	Hawaii	4131.0	2399.0	1420593
20	20	South Atlantic	Maryland	4914.0	2230.0	6035802
22	22	East North Central	Michigan	5209.0	3142.0	9984072
30	30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025
42	42	East South Central	Tennessee	6139.0	1744.0	6771631
33	33	South Atlantic	North Carolina	6451.0	2817.0	10381615
13	13	East North Central	Illinois	6752.0	3891.0	12723071
21	21	New England	Massachusetts	6811.0	13257.0	6882635
35	35	East North Central	Ohio	6929.0	3320.0	11676341
10	10	South Atlantic	Georgia	6943.0	2556.0	10511131
28	28	Mountain	Nevada	7058.0	486.0	3027341
2	2	Mountain	Arizona	7259.0	2606.0	7158024
5	5	Mountain	Colorado	7607.0	3250.0	5691287
38	38	Mid-Atlantic	Pennsylvania	8163.0	5349.0	12800922
37	37	Pacific	Oregon	11139.0	3337.0	4181886
47	47	Pacific	Washington	16424.0	5880.0	7523869
43	43	West South Central	Texas	19199.0	6111.0	28628666
9	9	South Atlantic	Florida	21443.0	9587.0	21244317
32	32	Mid-Atlantic	New York	39827.0	52070.0	19530351
4	4	Pacific	California	109008.0	20964.0	39461588

```
In [37]: homeless.sort_values(["individuals" , "state_pop"], ascending=[True, False])
```

Out[37]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
50	50	Mountain	Wyoming	434.0	205.0	577601
34	34	West North Central	North Dakota	467.0	75.0	758080
7	7	South Atlantic	Delaware	708.0	374.0	965479
39	39	New England	Rhode Island	747.0	354.0	1058287
45	45	New England	Vermont	780.0	511.0	624358
29	29	New England	New Hampshire	835.0	615.0	1353465
41	41	West North Central	South Dakota	836.0	323.0	878698
26	26	Mountain	Montana	983.0	422.0	1060665
48	48	South Atlantic	West Virginia	1021.0	222.0	1804291
24	24	East South Central	Mississippi	1024.0	328.0	2981020
12	12	Mountain	Idaho	1297.0	715.0	1750536
1	1	Pacific	Alaska	1434.0	582.0	735139
16	16	West North Central	Kansas	1443.0	773.0	2911359
19	19	New England	Maine	1450.0	1066.0	1339057
15	15	West North Central	Iowa	1711.0	1038.0	3148618
27	27	West North Central	Nebraska	1745.0	676.0	1925614
44	44	Mountain	Utah	1904.0	972.0	3153550
31	31	Mountain	New Mexico	1949.0	602.0	2092741
6	6	New England	Connecticut	2280.0	1696.0	3571520
3	3	West South Central	Arkansas	2280.0	432.0	3009733
18	18	West South Central	Louisiana	2540.0	519.0	4659690
0	0	East South Central	Alabama	2570.0	864.0	4887681
17	17	East South Central	Kentucky	2735.0	953.0	4461153
49	49	East North Central	Wisconsin	2740.0	2167.0	5807406
36	36	West South Central	Oklahoma	2823.0	1048.0	3940235
40	40	South Atlantic	South Carolina	3082.0	851.0	5084156
8	8	South Atlantic	District of Columbia	3770.0	3134.0	701547
14	14	East North Central	Indiana	3776.0	1482.0	6695497
25	25	West North	Missouri	3776.0	2107.0	6121623

	Unnamed: 0	region	state	individuals	family_members	state_pop
		Central				
46	46	South Atlantic	Virginia	3928.0	2047.0	8501286
23	23	West North Central	Minnesota	3993.0	3250.0	5606249
11	11	Pacific	Hawaii	4131.0	2399.0	1420593
20	20	South Atlantic	Maryland	4914.0	2230.0	6035802
22	22	East North Central	Michigan	5209.0	3142.0	9984072
30	30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025
42	42	East South Central	Tennessee	6139.0	1744.0	6771631
33	33	South Atlantic	North Carolina	6451.0	2817.0	10381615
13	13	East North Central	Illinois	6752.0	3891.0	12723071
21	21	New England	Massachusetts	6811.0	13257.0	6882635
35	35	East North Central	Ohio	6929.0	3320.0	11676341
10	10	South Atlantic	Georgia	6943.0	2556.0	10511131
28	28	Mountain	Nevada	7058.0	486.0	3027341
2	2	Mountain	Arizona	7259.0	2606.0	7158024
5	5	Mountain	Colorado	7607.0	3250.0	5691287
38	38	Mid-Atlantic	Pennsylvania	8163.0	5349.0	12800922
37	37	Pacific	Oregon	11139.0	3337.0	4181886
47	47	Pacific	Washington	16424.0	5880.0	7523869
43	43	West South Central	Texas	19199.0	6111.0	28628666
9	9	South Atlantic	Florida	21443.0	9587.0	21244317
32	32	Mid-Atlantic	New York	39827.0	52070.0	19530351
4	4	Pacific	California	109008.0	20964.0	39461588

PRACTICE

1

In [60]:

```
# Sorting the DataFrame
homelessness = pd.read_csv("homelessness.csv", index_col=0)
homelessness = homelessness.sort_values('individuals' , ascending=True)

homelessness.head()
```

Out[60]:

	region	state	individuals	family_members	state_pop
50	Mountain	Wyoming	434.0	205.0	577601
34	West North Central	North Dakota	467.0	75.0	758080
7	South Atlantic	Delaware	708.0	374.0	965479
39	New England	Rhode Island	747.0	354.0	1058287
45	New England	Vermont	780.0	511.0	624358

2

In [61]:

```
homelessness = homelessness.sort_values('family_members' , ascending=False)
homelessness.head()
```

Out[61]:

	region	state	individuals	family_members	state_pop
32	Mid-Atlantic	New York	39827.0	52070.0	19530351
4	Pacific	California	109008.0	20964.0	39461588
21	New England	Massachusetts	6811.0	13257.0	6882635
9	South Atlantic	Florida	21443.0	9587.0	21244317
43	West South Central	Texas	19199.0	6111.0	28628666

3

In [62]:

```
homelessness = homelessness.sort_values(["region" ,"family_members"] , ascen
homelessness.head()
```

Out[62]:

	region	state	individuals	family_members	state_pop
13	East North Central	Illinois	6752.0	3891.0	12723071
35	East North Central	Ohio	6929.0	3320.0	11676341
22	East North Central	Michigan	5209.0	3142.0	9984072
49	East North Central	Wisconsin	2740.0	2167.0	5807406
14	East North Central	Indiana	3776.0	1482.0	6695497

4

In [63]:

```
homelessness = pd.read_csv("homelessness.csv", index_col=0)
state_fam = homelessness[["state", "family_members"]]
state_fam.head()
```

Out[63]:

	state	family_members
0	Alabama	864.0
1	Alaska	582.0
2	Arizona	2606.0
3	Arkansas	432.0
4	California	20964.0

5

In [64]:

```
homelessness = pd.read_csv("homelessness.csv", index_col=0)
ind_gt_10k = homelessness[homelessness["individuals"] >= 10000]
ind_gt_10k
```

Out[64]:

	region	state	individuals	family_members	state_pop
4	Pacific	California	109008.0	20964.0	39461588
9	South Atlantic	Florida	21443.0	9587.0	21244317
32	Mid-Atlantic	New York	39827.0	52070.0	19530351
37	Pacific	Oregon	11139.0	3337.0	4181886
43	West South Central	Texas	19199.0	6111.0	28628666
47	Pacific	Washington	16424.0	5880.0	7523869

6

In [65]:

```
mountain_reg = homelessness[homelessness["region"] == "Mountain"]
mountain_reg
```

Out[65]:

	region	state	individuals	family_members	state_pop
2	Mountain	Arizona	7259.0	2606.0	7158024
5	Mountain	Colorado	7607.0	3250.0	5691287
12	Mountain	Idaho	1297.0	715.0	1750536
26	Mountain	Montana	983.0	422.0	1060665
28	Mountain	Nevada	7058.0	486.0	3027341
31	Mountain	New Mexico	1949.0	602.0	2092741
44	Mountain	Utah	1904.0	972.0	3153550
50	Mountain	Wyoming	434.0	205.0	577601

7

In [66]:

```
fam_It_1k_pac = homelessness[(homelessness["region"] == "Pacific") & (homele
fam_It_1k_pac
```

Out[66]:

	region	state	individuals	family_members	state_pop
1	Pacific	Alaska	1434.0	582.0	735139

8

In [67]:
```
south_mid_atlantic = homelessness["region"].isin(["South Atlantic", "Mid-Atl
homelessness[south_mid_atlantic]
```

Out[67]:

	region	state	individuals	family_members	state_pop
7	South Atlantic	Delaware	708.0	374.0	965479
8	South Atlantic	District of Columbia	3770.0	3134.0	701547
9	South Atlantic	Florida	21443.0	9587.0	21244317
10	South Atlantic	Georgia	6943.0	2556.0	10511131
20	South Atlantic	Maryland	4914.0	2230.0	6035802
30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025
32	Mid-Atlantic	New York	39827.0	52070.0	19530351
33	South Atlantic	North Carolina	6451.0	2817.0	10381615
38	Mid-Atlantic	Pennsylvania	8163.0	5349.0	12800922
40	South Atlantic	South Carolina	3082.0	851.0	5084156
46	South Atlantic	Virginia	3928.0	2047.0	8501286
48	South Atlantic	West Virginia	1021.0	222.0	1804291

9

In [69]:
```
mojave_state = ['Arizona', 'California', 'Nevada', 'Utah']
mojave_homelessness = homelessness[homelessness['state'].isin(mojave_state)]

mojave_homelessness
```

Out[69]:

	region	state	individuals	family_members	state_pop
2	Mountain	Arizona	7259.0	2606.0	7158024
4	Pacific	California	109008.0	20964.0	39461588
28	Mountain	Nevada	7058.0	486.0	3027341
44	Mountain	Utah	1904.0	972.0	3153550

In [73]:
```
homelessness = pd.read_csv("homelessness.csv", index_col=0)
```

10

In [75]:
```
homelessness["total"] = homelessness["family_members"] + homelessness["indiv
homelessness.head()
```

Out[75]:

	region	state	individuals	family_members	state_pop	total
0	East South Central	Alabama	2570.0	864.0	4887681	3434.0
1	Pacific	Alaska	1434.0	582.0	735139	2016.0
2	Mountain	Arizona	7259.0	2606.0	7158024	9865.0
3	West South Central	Arkansas	2280.0	432.0	3009733	2712.0
4	Pacific	California	109008.0	20964.0	39461588	129972.0

11

In [77]:

```
homelessness["p_individuals"] = homelessness['individuals'] / homelessness["
homelessness.head()
```

Out[77]:

	region	state	individuals	family_members	state_pop	total	p_individuals
0	East South Central	Alabama	2570.0	864.0	4887681	3434.0	0.748398
1	Pacific	Alaska	1434.0	582.0	735139	2016.0	0.711310
2	Mountain	Arizona	7259.0	2606.0	7158024	9865.0	0.735834
3	West South Central	Arkansas	2280.0	432.0	3009733	2712.0	0.840708
4	Pacific	California	109008.0	20964.0	39461588	129972.0	0.838704

12

In [80]:

```
homelessness = pd.read_csv("homelessness.csv", index_col=0)
homelessness["indiv_per_10k"] = 10000* homelessness["individuals"] / homeles
high_homelessness = homelessness[homelessness["indiv_per_10k"] >= 20]
high_homelessness_srt = high_homelessness.sort_values("indiv_per_10k" , asce
result = high_homelessness_srt[["state" , "indiv_per_10k"]]
result
```

Out[80]:

	state	indiv_per_10k
8	District of Columbia	53.738381
11	Hawaii	29.079406
4	California	27.623825
37	Oregon	26.636307
28	Nevada	23.314189
47	Washington	21.829195
32	New York	20.392363

Summarizing numerical data

In [84]:
```
homelessness['family_members'].mean()
homelessness['family_members'].median()
homelessness['family_members'].mode()
homelessness['family_members'].min()
homelessness['family_members'].max()
homelessness['family_members'].var()
homelessness['family_members'].std()
homelessness['family_members'].sum()
```

Out[84]:
```
178749.0
```

.agg() method

short for "aggregate" and is used to perform aggregation operations on DataFrame columns

allows you to apply one or more aggregation functions to one or more columns simultaneously

In [87]:
```
homelessness = pd.read_csv("homelessness.csv", index_col=0)
def pct30(column):
    return column.quantile(0.3)
homelessness["family_members"].agg(pct30)
```

Out[87]:
```
676.0
```

In [88]:
```
sales = pd.read_csv("sales_subset.csv", index_col = 0)
# Print the head of the sales DataFrame
print(sales.head())

# Print the info about the sales DataFrame
print(sales.info())

# Print the mean of weekly_sales
print(sales['weekly_sales'].mean())

# Print the median of weekly_sales
print(sales['weekly_sales'].median())

# Print the maximum of the date column
print(sales['date'].max())

# Print the minimum of the date column
print(sales['date'].min())
```

```
   store type  department        date  weekly_sales  is_holiday  \
0      1    A           1  2010-02-05      24924.50       False
1      1    A           1  2010-03-05      21827.90       False
2      1    A           1  2010-04-02      57258.43       False
3      1    A           1  2010-05-07      17413.94       False
4      1    A           1  2010-06-04      17558.09       False

   temperature_c  fuel_price_usd_per_l  unemployment
0       5.727778              0.679451         8.106
1       8.055556              0.693452         8.106
2      16.816667              0.718284         7.808
3      22.527778              0.748928         7.808
4      27.050000              0.714586         7.808
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10774 entries, 0 to 10773
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   store                 10774 non-null  int64
 1   type                  10774 non-null  object
 2   department            10774 non-null  int64
 3   date                  10774 non-null  object
 4   weekly_sales          10774 non-null  float64
 5   is_holiday            10774 non-null  bool
 6   temperature_c         10774 non-null  float64
 7   fuel_price_usd_per_l  10774 non-null  float64
 8   unemployment          10774 non-null  float64
dtypes: bool(1), float64(4), int64(2), object(2)
memory usage: 768.1+ KB
None
23843.950148505668
12049.064999999999
2012-10-26
2010-02-05
```

In [91]:

```
sales = pd.read_csv("sales_subset.csv", index_col = 0)
sales_1_1 = sales[(sales["department"] == 1) & (sales["store"] == 1) ]
# Sort sales_1_1 by date
sales_1_1 = sales_1_1.sort_values('date', ascending = True)

# Get the cumulative sum of weekly_sales, add as cum_weekly_sales col
sales_1_1['cum_weekly_sales'] = sales['weekly_sales'].cumsum()

# Get the cumulative max of weekly_sales, add as cum_max_sales col
sales_1_1['cum_max_sales'] = sales['weekly_sales'].cummax()

# See the columns you calculated
print(sales_1_1[["date", "weekly_sales", "cum_weekly_sales", "cum_max_sales"
```

```
          date  weekly_sales  cum_weekly_sales  cum_max_sales
0   2010-02-05      24924.50          24924.50       24924.50
1   2010-03-05      21827.90          46752.40       24924.50
2   2010-04-02      57258.43         104010.83       57258.43
3   2010-05-07      17413.94         121424.77       57258.43
4   2010-06-04      17558.09         138982.86       57258.43
5   2010-07-02      16333.14         155316.00       57258.43
6   2010-08-06      17508.41         172824.41       57258.43
7   2010-09-03      16241.78         189066.19       57258.43
8   2010-10-01      20094.19         209160.38       57258.43
9   2010-11-05      34238.88         243399.26       57258.43
10  2010-12-03      22517.56         265916.82       57258.43
11  2011-01-07      15984.24         281901.06       57258.43
```

13

In [127…

```
sales = pd.read_csv("sales_subset.csv", index_col = 0)
store_types = sales.drop_duplicates(subset=["store", "type"])
store_types.head()
```

Out[127]:

	store	type	department	date	weekly_sales	is_holiday	temperature_c	fuel_price_
0	1	A	1	2010-02-05	24924.50	False	5.727778	
901	2	A	1	2010-02-05	35034.06	False	4.550000	
1798	4	A	1	2010-02-05	38724.42	False	6.533333	
2699	6	A	1	2010-02-05	25619.00	False	4.683333	
3593	10	B	1	2010-02-05	40212.84	False	12.411111	

In [128…

```
sales = pd.read_csv("sales_subset.csv", index_col = 0)
store_depts = sales.drop_duplicates(subset=["store", "department"])
store_depts.head()
```

Out[128]:

	store	type	department	date	weekly_sales	is_holiday	temperature_c	fuel_price_us
0	1	A	1	2010-02-05	24924.50	False	5.727778	(
12	1	A	2	2010-02-05	50605.27	False	5.727778	(
24	1	A	3	2010-02-05	13740.12	False	5.727778	(
36	1	A	4	2010-02-05	39954.04	False	5.727778	(
48	1	A	5	2010-02-05	32229.38	False	5.727778	(

In [129…

```
sales = pd.read_csv("sales_subset.csv", index_col = 0)
holiday_df = sales[sales['is_holiday'] == True]
holiday_dates = holiday_df.drop_duplicates(subset=["date"])
print(holiday_dates["date"])
```

```
498     2010-09-10
691     2011-11-25
2315    2010-02-12
6735    2012-09-07
6810    2010-12-31
6815    2012-02-10
6820    2011-09-09
Name: date, dtype: object
```

14

In [130…

```
sales = pd.read_csv("sales_subset.csv", index_col = 0)
store_counts = store_types['type'].value_counts()
print(store_counts)
```

```
A    11
B     1
Name: type, dtype: int64
```

In [131…

```
store_types = store_types["type"].value_counts(normalize = True)
print(store_types)
```

```
A    0.916667
B    0.083333
Name: type, dtype: float64
```

In [137…

```
dept_counts_sort = store_depts["department"].value_counts(sort = True)
print(dept_counts_sort)
```

```
1     12
55    12
72    12
71    12
67    12
      ..
37    10
48     8
50     6
39     4
43     2
Name: department, Length: 80, dtype: int64
```

In [138…

```
dept_props_sorted = store_depts["department"].value_counts(sort = True, norm
dept_props_sorted
```

Out[138]:

```
1     12
55    12
72    12
71    12
67    12
      ..
37    10
48     8
50     6
39     4
43     2
Name: department, Length: 80, dtype: int64
```

15

In [139…

```
# Calc total weekly sales
sales_all = sales["weekly_sales"].sum()

# Subset for type A stores, calc total weekly sales
sales_A = sales[sales["type"] == "A"]["weekly_sales"].sum()

# Subset for type B stores, calc total weekly sales
sales_B = sales[sales["type"] == "B"]["weekly_sales"].sum()

# Subset for type C stores, calc total weekly sales
sales_C = sales[sales["type"] == "C"]["weekly_sales"].sum()

# Get proportion for each type
sales_propn_by_type = [sales_A, sales_B, sales_C] / sales_all
print(sales_propn_by_type)
```

```
[0.9097747 0.0902253 0.        ]
```

Store Type A: The proportion of sales for store type A is approximately 90.98%. This indicates that store type A has the highest contribution to the total weekly sales among the three store types. It implies that store type A is likely the most dominant or highest-performing store type in terms of sales.

Store Type B: The proportion of sales for store type B is approximately 9.02%. Although it is significantly lower than store type A, it still represents a notable portion of the total sales. Store type B likely represents a significant number of stores or has a reasonable level of sales performance, but it is not as dominant as store type A.

Store Type C: The proportion of sales for store type C is 0%. This suggests that there are either no stores or no recorded sales for store type C in the dataset. It could indicate that store type C is not present in the dataset or that it has not generated any sales during the recorded period.

In [140…

```
# Import numpy with the alias np
import numpy as np

# For each store type, aggregate weekly_sales: get min, max, mean, and media
sales_stats = sales.groupby('type')['weekly_sales'].agg([min, max, np.mean,

# Print sales_stats
print(sales_stats)

# For each store type, aggregate unemployment and fuel_price_usd_per_l: get
unemp_fuel_stats = sales.groupby('type')[['unemployment','fuel_price_usd_per

# Print unemp_fuel_stats
print(unemp_fuel_stats)
```

```
         min        max          mean     median
type
A    -1098.0  293966.05  23674.667242  11943.92
B     -798.0  232558.51  25696.678370  13336.08
     unemployment                         fuel_price_usd_per_l            \
              min    max      mean median                  min       max
type
A           3.879  8.992  7.972611  8.067             0.664129  1.107410
B           7.170  9.765  9.279323  9.199             0.760023  1.107674


          mean    median
type
A     0.744619  0.735455
B     0.805858  0.803348
```

In [142…

```
temperatures = pd.read_csv("temperatures.csv" , index_col = 0)
# Look at temperatures
print(temperatures)
# Set the index of temperatures to city
temperatures_ind = temperatures.set_index('city')

# Look at temperatures_ind
print(temperatures_ind)

# Reset the temperatures_ind index, keeping its contents
print(temperatures_ind.reset_index())

# Reset the temperatures_ind index, dropping its contents
print(temperatures_ind.reset_index(drop = True))
# Make a list of cities to subset on
cities = ["Moscow", "Saint Petersburg"]

# Subset temperatures using square brackets
print(temperatures[temperatures['city'].isin(cities)])

# Subset temperatures_ind using .loc[]
print(temperatures_ind.loc[cities])
```

```
             date     city        country  avg_temp_c
0      2000-01-01  Abidjan  Côte D'Ivoire      27.293
1      2000-02-01  Abidjan  Côte D'Ivoire      27.685
2      2000-03-01  Abidjan  Côte D'Ivoire      29.061
3      2000-04-01  Abidjan  Côte D'Ivoire      28.162
4      2000-05-01  Abidjan  Côte D'Ivoire      27.547
...           ...      ...            ...         ...
16495  2013-05-01     Xian          China      18.979
16496  2013-06-01     Xian          China      23.522
16497  2013-07-01     Xian          China      25.251
16498  2013-08-01     Xian          China      24.528
16499  2013-09-01     Xian          China         NaN

[16500 rows x 4 columns]
               date        country  avg_temp_c
city
Abidjan  2000-01-01  Côte D'Ivoire      27.293
Abidjan  2000-02-01  Côte D'Ivoire      27.685
Abidjan  2000-03-01  Côte D'Ivoire      29.061
Abidjan  2000-04-01  Côte D'Ivoire      28.162
Abidjan  2000-05-01  Côte D'Ivoire      27.547
...             ...            ...         ...
Xian     2013-05-01          China      18.979
Xian     2013-06-01          China      23.522
Xian     2013-07-01          China      25.251
Xian     2013-08-01          China      24.528
Xian     2013-09-01          China         NaN

[16500 rows x 3 columns]
          city        date        country  avg_temp_c
0      Abidjan  2000-01-01  Côte D'Ivoire      27.293
1      Abidjan  2000-02-01  Côte D'Ivoire      27.685
2      Abidjan  2000-03-01  Côte D'Ivoire      29.061
3      Abidjan  2000-04-01  Côte D'Ivoire      28.162
4      Abidjan  2000-05-01  Côte D'Ivoire      27.547
...        ...         ...            ...         ...
16495     Xian  2013-05-01          China      18.979
16496     Xian  2013-06-01          China      23.522
16497     Xian  2013-07-01          China      25.251
16498     Xian  2013-08-01          China      24.528
16499     Xian  2013-09-01          China         NaN

[16500 rows x 4 columns]
             date        country  avg_temp_c
0      2000-01-01  Côte D'Ivoire      27.293
1      2000-02-01  Côte D'Ivoire      27.685
2      2000-03-01  Côte D'Ivoire      29.061
3      2000-04-01  Côte D'Ivoire      28.162
4      2000-05-01  Côte D'Ivoire      27.547
...           ...            ...         ...
16495  2013-05-01          China      18.979
16496  2013-06-01          China      23.522
16497  2013-07-01          China      25.251
16498  2013-08-01          China      24.528
16499  2013-09-01          China         NaN

[16500 rows x 3 columns]
             date             city country  avg_temp_c
10725  2000-01-01           Moscow  Russia      -7.313
10726  2000-02-01           Moscow  Russia      -3.551
10727  2000-03-01           Moscow  Russia      -1.661
10728  2000-04-01           Moscow  Russia      10.096
10729  2000-05-01           Moscow  Russia      10.357
...           ...              ...     ...         ...
```

```
13360  2013-05-01  Saint Petersburg  Russia      12.355
13361  2013-06-01  Saint Petersburg  Russia      17.185
13362  2013-07-01  Saint Petersburg  Russia      17.234
13363  2013-08-01  Saint Petersburg  Russia      17.153
13364  2013-09-01  Saint Petersburg  Russia         NaN

[330 rows x 4 columns]
                        date country  avg_temp_c
city
Moscow            2000-01-01  Russia      -7.313
Moscow            2000-02-01  Russia      -3.551
Moscow            2000-03-01  Russia      -1.661
Moscow            2000-04-01  Russia      10.096
Moscow            2000-05-01  Russia      10.357
...                      ...     ...         ...
Saint Petersburg  2013-05-01  Russia      12.355
Saint Petersburg  2013-06-01  Russia      17.185
Saint Petersburg  2013-07-01  Russia      17.234
Saint Petersburg  2013-08-01  Russia      17.153
Saint Petersburg  2013-09-01  Russia         NaN

[330 rows x 3 columns]
```

16

In [144…

```
temperatures_ind = temperatures.set_index(["country", "city"])

rows_to_keep = [("Brazil", "Rio De Janeiro"), ("Pakistan", "Lahore")]

print(temperatures_ind.loc[rows_to_keep])
```

```
                               date  avg_temp_c
country  city
Brazil   Rio De Janeiro  2000-01-01      25.974
         Rio De Janeiro  2000-02-01      26.699
         Rio De Janeiro  2000-03-01      26.270
         Rio De Janeiro  2000-04-01      25.750
         Rio De Janeiro  2000-05-01      24.356
...                             ...         ...
Pakistan Lahore          2013-05-01      33.457
         Lahore          2013-06-01      34.456
         Lahore          2013-07-01      33.279
         Lahore          2013-08-01      31.511
         Lahore          2013-09-01         NaN

[330 rows x 2 columns]
```

In [ ]: