In [11]:

```
import pandas as pd

taxi_owner = pd.read_pickle('taxi_owners.p')
taxi_owner.head()
```

Out[11]:

	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

In [7]:

```
taxi_owner.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3519 entries, 0 to 3518
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   rid      3519 non-null   object
 1   vid      3519 non-null   object
 2   owner    3519 non-null   object
 3   address  3519 non-null   object
 4   zip      3519 non-null   object
dtypes: object(5)
memory usage: 137.6+ KB
```

In [9]:

```
taxi_owner.describe()
```

Out[9]:

	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 [14]:

```
taxi_owner.shape
```

Out[14]:

```
(3519, 5)
```

In [16]:
```
taxi_owner.values
```

Out[16]:
```
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 [17]:
```
taxi_owner.columns
```

Out[17]:
```
Index(['rid', 'vid', 'owner', 'address', 'zip'], dtype='object')
```

In [18]:
```
taxi_owner.index
```

Out[18]:
```
RangeIndex(start=0, stop=3519, step=1)
```

In [55]:
```
homelessnessdf = pd.read_csv('homelessness.csv')
homelessnessdf.head()
```

Out[55]:

	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 [29]:
```
print(homelessnessdf.columns)
```

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

In [ ]:
```

```

In [ ]:
```
#Ex1
```

In [33]:

```
homelessness_ind = homelessnessdf.sort_values("individuals", ascending = T
homelessness_ind.head()
```

Out[33]:

	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

In [ ]:

```
#Ex2
```

In [34]:

```
homelessness_fam = homelessnessdf.sort_values("family_members", ascending
homelessness_fam.head()
```

Out[34]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
32	32	Mid-Atlantic	New York	39827.0	52070.0	19530351
4	4	Pacific	California	109008.0	20964.0	39461588
21	21	New England	Massachusetts	6811.0	13257.0	6882635
9	9	South Atlantic	Florida	21443.0	9587.0	21244317
43	43	West South Central	Texas	19199.0	6111.0	28628666

In [ ]:

```
#Ex3
```

In [35]:

```
homelessness_reg_fam = homelessnessdf.sort_values(["region", "family_membe
homelessness_reg_fam.head()
```

Out[35]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
13	13	East North Central	Illinois	6752.0	3891.0	12723071
35	35	East North Central	Ohio	6929.0	3320.0	11676341
22	22	East North Central	Michigan	5209.0	3142.0	9984072
49	49	East North Central	Wisconsin	2740.0	2167.0	5807406
14	14	East North Central	Indiana	3776.0	1482.0	6695497

In [ ]:

```
#Ex4
```

In [40]:

```python
state_fam = homelessnessdf [["state", "family_members"]]
state_fam.head()
```

Out[40]:

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

In [ ]:

```python
#Ex5
```

In [49]:

```python
ind_gt_10k = homelessnessdf[homelessnessdf ["individuals"] > 10000]
ind_gt_10k.head()
```

Out[49]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
4	4	Pacific	California	109008.0	20964.0	39461588
9	9	South Atlantic	Florida	21443.0	9587.0	21244317
32	32	Mid-Atlantic	New York	39827.0	52070.0	19530351
37	37	Pacific	Oregon	11139.0	3337.0	4181886
43	43	West South Central	Texas	19199.0	6111.0	28628666

In [ ]:

```python
#Ex6
```

In [48]:

```python
mountain_reg = homelessnessdf[homelessnessdf["region"] == "Mountain"]
mountain_reg.head()
```

Out[48]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
2	2	Mountain	Arizona	7259.0	2606.0	7158024
5	5	Mountain	Colorado	7607.0	3250.0	5691287
12	12	Mountain	Idaho	1297.0	715.0	1750536
26	26	Mountain	Montana	983.0	422.0	1060665
28	28	Mountain	Nevada	7058.0	486.0	3027341

In [ ]:

```python
#Ex7
```

In [54]:

```
fam_lt_1k_pac = homelessnessdf[(homelessnessdf["family_members"] < 1000) &
fam_lt_1k_pac.head()
```

Out[54]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
1	1	Pacific	Alaska	1434.0	582.0	735139

In [56]:

```
#Ex8
```

In [58]:

```
south_mid_atlantic = homelessnessdf[(homelessnessdf["region"].isin(["South
south_mid_atlantic
```

Out[58]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
7	7	South Atlantic	Delaware	708.0	374.0	965479
8	8	South Atlantic	District of Columbia	3770.0	3134.0	701547
9	9	South Atlantic	Florida	21443.0	9587.0	21244317
10	10	South Atlantic	Georgia	6943.0	2556.0	10511131
20	20	South Atlantic	Maryland	4914.0	2230.0	6035802
30	30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025
32	32	Mid-Atlantic	New York	39827.0	52070.0	19530351
33	33	South Atlantic	North Carolina	6451.0	2817.0	10381615
38	38	Mid-Atlantic	Pennsylvania	8163.0	5349.0	12800922
40	40	South Atlantic	South Carolina	3082.0	851.0	5084156
46	46	South Atlantic	Virginia	3928.0	2047.0	8501286
48	48	South Atlantic	West Virginia	1021.0	222.0	1804291

In [59]:

```
#Ex9
```

In [64]:

```
canu= ["California", "Arizona", "Nevada", "Utah"]

mojave_homelessness = homelessnessdf[homelessnessdf["state"].isin(canu)]
mojave_homelessness.head()
```

Out[64]:

	Unnamed: 0	region	state	individuals	family_members	state_pop
2	2	Mountain	Arizona	7259.0	2606.0	7158024
4	4	Pacific	California	109008.0	20964.0	39461588
28	28	Mountain	Nevada	7058.0	486.0	3027341
44	44	Mountain	Utah	1904.0	972.0	3153550

In [65]:

```
#Ex10
```

```
In [66]:  homelessnessdf["total"] = homelessnessdf["individuals"] + homelessnessdf["
          homelessnessdf.head(50)
```

Out[66]:

	Unnamed: 0	region	state	individuals	family_members	state_pop	total
0	0	East South Central	Alabama	2570.0	864.0	4887681	3434.0
1	1	Pacific	Alaska	1434.0	582.0	735139	2016.0
2	2	Mountain	Arizona	7259.0	2606.0	7158024	9865.0
3	3	West South Central	Arkansas	2280.0	432.0	3009733	2712.0
4	4	Pacific	California	109008.0	20964.0	39461588	129972.0
5	5	Mountain	Colorado	7607.0	3250.0	5691287	10857.0
6	6	New England	Connecticut	2280.0	1696.0	3571520	3976.0
7	7	South Atlantic	Delaware	708.0	374.0	965479	1082.0
8	8	South Atlantic	District of Columbia	3770.0	3134.0	701547	6904.0
9	9	South Atlantic	Florida	21443.0	9587.0	21244317	31030.0
10	10	South Atlantic	Georgia	6943.0	2556.0	10511131	9499.0
11	11	Pacific	Hawaii	4131.0	2399.0	1420593	6530.0
12	12	Mountain	Idaho	1297.0	715.0	1750536	2012.0
13	13	East North Central	Illinois	6752.0	3891.0	12723071	10643.0
14	14	East North Central	Indiana	3776.0	1482.0	6695497	5258.0
15	15	West North Central	Iowa	1711.0	1038.0	3148618	2749.0
16	16	West North Central	Kansas	1443.0	773.0	2911359	2216.0
17	17	East South Central	Kentucky	2735.0	953.0	4461153	3688.0
18	18	West South Central	Louisiana	2540.0	519.0	4659690	3059.0
19	19	New England	Maine	1450.0	1066.0	1339057	2516.0

	Unnamed: 0	region	state	individuals	family_members	state_pop	total
20	20	South Atlantic	Maryland	4914.0	2230.0	6035802	7144.0
21	21	New England	Massachusetts	6811.0	13257.0	6882635	20068.0
22	22	East North Central	Michigan	5209.0	3142.0	9984072	8351.0
23	23	West North Central	Minnesota	3993.0	3250.0	5606249	7243.0
24	24	East South Central	Mississippi	1024.0	328.0	2981020	1352.0
25	25	West North Central	Missouri	3776.0	2107.0	6121623	5883.0
26	26	Mountain	Montana	983.0	422.0	1060665	1405.0
27	27	West North Central	Nebraska	1745.0	676.0	1925614	2421.0
28	28	Mountain	Nevada	7058.0	486.0	3027341	7544.0
29	29	New England	New Hampshire	835.0	615.0	1353465	1450.0
30	30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025	9398.0
31	31	Mountain	New Mexico	1949.0	602.0	2092741	2551.0
32	32	Mid-Atlantic	New York	39827.0	52070.0	19530351	91897.0
33	33	South Atlantic	North Carolina	6451.0	2817.0	10381615	9268.0
34	34	West North Central	North Dakota	467.0	75.0	758080	542.0
35	35	East North Central	Ohio	6929.0	3320.0	11676341	10249.0
36	36	West South Central	Oklahoma	2823.0	1048.0	3940235	3871.0
37	37	Pacific	Oregon	11139.0	3337.0	4181886	14476.0
38	38	Mid-Atlantic	Pennsylvania	8163.0	5349.0	12800922	13512.0
39	39	New England	Rhode Island	747.0	354.0	1058287	1101.0
40	40	South Atlantic	South Carolina	3082.0	851.0	5084156	3933.0

	Unnamed: 0	region	state	individuals	family_members	state_pop	total
41	41	West North Central	South Dakota	836.0	323.0	878698	1159.0
42	42	East South Central	Tennessee	6139.0	1744.0	6771631	7883.0
43	43	West South Central	Texas	19199.0	6111.0	28628666	25310.0
44	44	Mountain	Utah	1904.0	972.0	3153550	2876.0
45	45	New England	Vermont	780.0	511.0	624358	1291.0
46	46	South Atlantic	Virginia	3928.0	2047.0	8501286	5975.0

In [67]:

```
#Ex11
```

In [68]:

```
homelessnessdf["p_individuals"] = homelessnessdf["individuals"] / homelessn
homelessnessdf.head()
```

Out[68]:

	Unnamed: 0	region	state	individuals	family_members	state_pop	total	p_indiv
0	0	East South Central	Alabama	2570.0	864.0	4887681	3434.0	0.7
1	1	Pacific	Alaska	1434.0	582.0	735139	2016.0	0.7
2	2	Mountain	Arizona	7259.0	2606.0	7158024	9865.0	0.7
3	3	West South Central	Arkansas	2280.0	432.0	3009733	2712.0	0.8
4	4	Pacific	California	109008.0	20964.0	39461588	129972.0	0.8

In [ ]:

```
#Ex12
```

In [82]:

```
homelessnessdf["indiv_per_10k"] = 10000 * homelessnessdf['individuals'] / 

high_homelessness = homelessnessdf[homelessnessdf["indiv_per_10k"] > 20]

high_homelessness_srt = high_homelessness.sort_values("indiv_per_10k", asc


high_homelessness_srt [["state", "indiv_per_10k"]]
```

Out[82]:

	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

In [ ]:

In [85]:

```
def pct40(column):
    return column.quantile(0.5)

homelessnessdf["family_members"].agg(pct40)
```

Out[85]:

```
1482.0
```

In [89]:

```


sales = pd.read_csv('sales_subset.csv')
sales.head()
```

Out[89]:

	Unnamed: 0	store	type	department	date	weekly_sales	is_holiday	temperature_c
0	0	1	A	1	2010-02-05	24924.50	False	5.727778
1	1	1	A	1	2010-03-05	21827.90	False	8.055556
2	2	1	A	1	2010-04-02	57258.43	False	16.816667
3	3	1	A	1	2010-05-07	17413.94	False	22.527778
4	4	1	A	1	2010-06-04	17558.09	False	27.050000

In [92]:

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10774 entries, 0 to 10773
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10774 non-null  int64  
 1   store               10774 non-null  int64  
 2   type                10774 non-null  object 
 3   department          10774 non-null  int64  
 4   date                10774 non-null  object 
 5   weekly_sales        10774 non-null  float64
 6   is_holiday          10774 non-null  bool   
 7   temperature_c       10774 non-null  float64
 8   fuel_price_usd_per_l  10774 non-null  float64
 9   unemployment        10774 non-null  float64
dtypes: bool(1), float64(4), int64(3), object(2)
memory usage: 768.2+ KB
None

23843.950148505668
12049.064999999999
2012-10-26
2010-02-05
```

In [94]:

```
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_sale
```

```
          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
```

In [95]:

```
#13
```

In [96]:

```
store_types = sales.drop_duplicates(subset = ["store", "type"])
store_types.head()
```

Out[96]:

	Unnamed: 0	store	type	department	date	weekly_sales	is_holiday	temperature
0	0	1	A	1	2010-02-05	24924.50	False	5.7277
901	901	2	A	1	2010-02-05	35034.06	False	4.5500
1798	1798	4	A	1	2010-02-05	38724.42	False	6.5333
2699	2699	6	A	1	2010-02-05	25619.00	False	4.6833
3593	3593	10	B	1	2010-02-05	40212.84	False	12.4111

In [98]:

```
store_depts = sales.drop_duplicates(subset = ["store", "department"])
store_depts.head()
```

Out[98]:

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

In [118]:

```
holiday_dates = sales[sales["is_holiday"] == True].drop_duplicates(subset
holiday_dates['date']
```

Out[118]:

```
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
```

In [109]:

```
#Ex14
```

In [157]:

```
store_types = sales.drop_duplicates(subset = ['store', 'type'])
store_counts = store_types['type'].value_counts()
print(store_counts)

store_props = store_types['type'].value_counts(normalize= True)
print(store_props)

store_depts = sales.drop_duplicates(subset = ['store', 'department'])
dept_counts_sorted = store_depts [ 'department'].value_counts(sort=True)
print(dept_counts_sorted)

dept_props_sorted = store_depts['department'].value_counts(sort = True, no
print(dept_props_sorted)
```

```
A    11
B     1
Name: type, dtype: int64
A    0.916667
B    0.083333
Name: type, dtype: float64
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
1     0.012917
55    0.012917
72    0.012917
71    0.012917
67    0.012917
        ...
37    0.010764
48    0.008611
50    0.006459
39    0.004306
43    0.002153
Name: department, Length: 80, dtype: float64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
#Practise
```

In [145]:

```
# 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.        ]
```

In [ ]:

```
#15
The code first calculates the sum of all weekly sales using the sum() func
Then, it subsets the data for each store type (A, B, and C) and calculates
Finally, it computes the proportion of sales for each store type by dividi
The results are stored in the variable sales_propn_by_type as a list.
```

In [ ]:

```
#Practise
```

In [139]:

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

# For each store type, aggregate weekly_sales: get min, max, mean, and med
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: ge
unemp_fuel_stats = sales.groupby('type')[['unemployment','fuel_price_usd_p

# 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 [141]:

```
# Print mean weekly_sales by department and type; fill missing values with
print(sales.pivot_table(values = 'weekly_sales', index = 'department', col
```

```
type                     A              B
department
1             30961.725379   44050.626667
2             67600.158788  112958.526667
3             17160.002955   30580.655000
4             44285.399091   51219.654167
5             34821.011364   63236.875000
...                    ...            ...
95           123933.787121   77082.102500
96            21367.042857    9528.538333
97            28471.266970    5828.873333
98            12875.423182     217.428333
99              379.123659       0.000000

[80 rows x 2 columns]
```

In [143]:

```
temperatures = pd.read_csv('temperatures.csv')

# 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])
```

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

[16500 rows x 5 columns]
         Unnamed: 0        date        country  avg_temp_c
city
Abidjan           0  2000-01-01  Côte D'Ivoire      27.293
Abidjan           1  2000-02-01  Côte D'Ivoire      27.685
Abidjan           2  2000-03-01  Côte D'Ivoire      29.061
Abidjan           3  2000-04-01  Côte D'Ivoire      28.162
Abidjan           4  2000-05-01  Côte D'Ivoire      27.547
...             ...         ...            ...         ...
Xian          16495  2013-05-01          China      18.979
Xian          16496  2013-06-01          China      23.522
Xian          16497  2013-07-01          China      25.251
Xian          16498  2013-08-01          China      24.528
Xian          16499  2013-09-01          China         NaN

[16500 rows x 4 columns]
          city  Unnamed: 0        date        country  avg_temp_c
0      Abidjan           0  2000-01-01  Côte D'Ivoire      27.293
```

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

[16500 rows x 5 columns]
       Unnamed: 0        date        country  avg_temp_c
0               0  2000-01-01  Côte D'Ivoire      27.293
1               1  2000-02-01  Côte D'Ivoire      27.685
2               2  2000-03-01  Côte D'Ivoire      29.061
3               3  2000-04-01  Côte D'Ivoire      28.162
4               4  2000-05-01  Côte D'Ivoire      27.547
...           ...         ...            ...         ...
16495       16495  2013-05-01          China      18.979
16496       16496  2013-06-01          China      23.522
16497       16497  2013-07-01          China      25.251
16498       16498  2013-08-01          China      24.528
16499       16499  2013-09-01          China         NaN

[16500 rows x 4 columns]
       Unnamed: 0        date              city country  avg_temp_c
10725       10725  2000-01-01            Moscow  Russia      -7.313
10726       10726  2000-02-01            Moscow  Russia      -3.551
10727       10727  2000-03-01            Moscow  Russia      -1.661
10728       10728  2000-04-01            Moscow  Russia      10.096
10729       10729  2000-05-01            Moscow  Russia      10.357
...           ...         ...               ...     ...         ...
13360       13360  2013-05-01  Saint Petersburg  Russia      12.355
13361       13361  2013-06-01  Saint Petersburg  Russia      17.185
13362       13362  2013-07-01  Saint Petersburg  Russia      17.234
13363       13363  2013-08-01  Saint Petersburg  Russia      17.153
13364       13364  2013-09-01  Saint Petersburg  Russia         NaN

[330 rows x 5 columns]
                  Unnamed: 0        date country  avg_temp_c
city                                                        
Moscow                 10725  2000-01-01  Russia      -7.313
Moscow                 10726  2000-02-01  Russia      -3.551
Moscow                 10727  2000-03-01  Russia      -1.661
Moscow                 10728  2000-04-01  Russia      10.096
Moscow                 10729  2000-05-01  Russia      10.357
...                      ...         ...     ...         ...
Saint Petersburg       13360  2013-05-01  Russia      12.355
Saint Petersburg       13361  2013-06-01  Russia      17.185
Saint Petersburg       13362  2013-07-01  Russia      17.234
Saint Petersburg       13363  2013-08-01  Russia      17.153
Saint Petersburg       13364  2013-09-01  Russia         NaN

[330 rows x 4 columns]
```

In [144]:

```
#16
```

In [150]:

```
temperatures_ind = temperatures.set_index(["country", "city"])
rows_to_keep = [("Brazil", "Rio De Janeiro") , ("Pakistan" , "Lahore")]
subset = temperatures_ind.loc[rows_to_keep]

subset
```

Out[150]:

		Unnamed: 0	date	avg_temp_c
country	**city**			
Brazil	**Rio De Janeiro**	12540	2000-01-01	25.974
	Rio De Janeiro	12541	2000-02-01	26.699
	Rio De Janeiro	12542	2000-03-01	26.270
	Rio De Janeiro	12543	2000-04-01	25.750
	Rio De Janeiro	12544	2000-05-01	24.356
...	**...**	...	...	...
Pakistan	**Lahore**	8575	2013-05-01	33.457
	Lahore	8576	2013-06-01	34.456
	Lahore	8577	2013-07-01	33.279
	Lahore	8578	2013-08-01	31.511
	Lahore	8579	2013-09-01	NaN

330 rows × 3 columns

In [ ]: