```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
# taxi_owner = pd.read_pickle('taxi_owners.p')
# taxi_owner.head()
```

```
homelessness = pd.read_csv('homelessness.csv',index_col=0)
# homelessness
print(homelessness.head())

print('-'*50)
print(homelessness.info())

print('-'*50)
print(homelessness.shape)

print('-'*50)
print(homelessness.describe())
```

```
               region       state  individuals  family_members  state_pop
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
--------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
Index: 51 entries, 0 to 50
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   region          51 non-null     object 
 1   state           51 non-null     object 
 2   individuals     51 non-null     float64
 3   family_members  51 non-null     float64
 4   state_pop       51 non-null     int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 2.4+ KB
None
--------------------------------------------------
(51, 5)
--------------------------------------------------
         individuals  family_members     state_pop
count      51.000000       51.000000  5.100000e+01
mean     7225.784314     3504.882353  6.405637e+06
std     15991.025083     7805.411811  7.327258e+06
min       434.000000       75.000000  5.776010e+05
25%      1446.500000      592.000000  1.777414e+06
50%      3082.000000     1482.000000  4.461153e+06
75%      6781.500000     3196.000000  7.340946e+06
max    109008.000000    52070.000000  3.946159e+07
```

```
print(homelessness.values)
print("-"*50)

print(homelessness.columns)
print("-"*50)

print(homelessness.index)
```

```
[['East South Central' 'Alabama' 2570.0 864.0 4887681]
 ['Pacific' 'Alaska' 1434.0 582.0 735139]
 ['Mountain' 'Arizona' 7259.0 2606.0 7158024]
 ['West South Central' 'Arkansas' 2280.0 432.0 3009733]
 ['Pacific' 'California' 109008.0 20964.0 39461588]
 ['Mountain' 'Colorado' 7607.0 3250.0 5691287]
 ['New England' 'Connecticut' 2280.0 1696.0 3571520]
 ['South Atlantic' 'Delaware' 708.0 374.0 965479]
 ['South Atlantic' 'District of Columbia' 3770.0 3134.0 701547]
 ['South Atlantic' 'Florida' 21443.0 9587.0 21244317]
 ['South Atlantic' 'Georgia' 6943.0 2556.0 10511131]
 ['Pacific' 'Hawaii' 4131.0 2399.0 1420593]
 ['Mountain' 'Idaho' 1297.0 715.0 1750536]
 ['East North Central' 'Illinois' 6752.0 3891.0 12723071]
 ['East North Central' 'Indiana' 3776.0 1482.0 6695497]
 ['West North Central' 'Iowa' 1711.0 1038.0 3148618]
 ['West North Central' 'Kansas' 1443.0 773.0 2911359]
 ['East South Central' 'Kentucky' 2735.0 953.0 4461153]
 ['West South Central' 'Louisiana' 2540.0 519.0 4659690]
 ['New England' 'Maine' 1450.0 1066.0 1339057]
 ['South Atlantic' 'Maryland' 4914.0 2230.0 6035802]
 ['New England' 'Massachusetts' 6811.0 13257.0 6882635]
 ['East North Central' 'Michigan' 5209.0 3142.0 9984072]
 ['West North Central' 'Minnesota' 3993.0 3250.0 5606249]
 ['East South Central' 'Mississippi' 1024.0 328.0 2981020]
 ['West North Central' 'Missouri' 3776.0 2107.0 6121623]
 ['Mountain' 'Montana' 983.0 422.0 1060665]
 ['West North Central' 'Nebraska' 1745.0 676.0 1925614]
 ['Mountain' 'Nevada' 7058.0 486.0 3027341]
 ['New England' 'New Hampshire' 835.0 615.0 1353465]
 ['Mid-Atlantic' 'New Jersey' 6048.0 3350.0 8886025]
 ['Mountain' 'New Mexico' 1949.0 602.0 2092741]
 ['Mid-Atlantic' 'New York' 39827.0 52070.0 19530351]
 ['South Atlantic' 'North Carolina' 6451.0 2817.0 10381615]
 ['West North Central' 'North Dakota' 467.0 75.0 758080]
 ['East North Central' 'Ohio' 6929.0 3320.0 11676341]
 ['West South Central' 'Oklahoma' 2823.0 1048.0 3940235]
 ['Pacific' 'Oregon' 11139.0 3337.0 4181886]
 ['Mid-Atlantic' 'Pennsylvania' 8163.0 5349.0 12800922]
 ['New England' 'Rhode Island' 747.0 354.0 1058287]
 ['South Atlantic' 'South Carolina' 3082.0 851.0 5084156]
 ['West North Central' 'South Dakota' 836.0 323.0 878698]
 ['East South Central' 'Tennessee' 6139.0 1744.0 6771631]
 ['West South Central' 'Texas' 19199.0 6111.0 28628666]
 ['Mountain' 'Utah' 1904.0 972.0 3153550]
 ['New England' 'Vermont' 780.0 511.0 624358]
 ['South Atlantic' 'Virginia' 3928.0 2047.0 8501286]
 ['Pacific' 'Washington' 16424.0 5880.0 7523869]
 ['South Atlantic' 'West Virginia' 1021.0 222.0 1804291]
 ['East North Central' 'Wisconsin' 2740.0 2167.0 5807406]
 ['Mountain' 'Wyoming' 434.0 205.0 577601]]
--------------------------------------------------
Index(['region', 'state', 'individuals', 'family_members', 'state_pop'], dt
--------------------------------------------------
Index([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50],
      dtype='int64')
```

```
# dogs = pd.read_csv("dogs.csv")

# dogs.sort_values("weight_kg","height_cm")
# dogs

# #subsetting
# dogs[["breed","height_cm"]]

# #step by step subset
# cols_tosubset = ["breed", "height_cm"]
# dogs[cols_tosubset]

# dogs[dogs["breed"] == "Labrador"]
# dogs[dogs["date_of_birth"] < "2015-01-01"]

# #based on multiple conditions
# is_lab = dogs["breed"] == "Labrador"
# is_brown = dogs["color"] == "Brown"
# dogs[is_labs & is_brown]


#No.1
homelessness_ind = homelessness.sort_values("individuals")
print(homelessness_ind.head())
```

```
                region         state  individuals  family_members  state_po
50            Mountain       Wyoming        434.0           205.0     57760
34  West North Central  North Dakota        467.0            75.0     75808
7       South Atlantic      Delaware        708.0           374.0     96547
39         New England  Rhode Island        747.0           354.0    105828
45         New England       Vermont        780.0           511.0     62435
```

```
#No.2
homelessness_fam = homelessness.sort_values("family_members", ascending=False)
print(homelessness_fam.head())
```

```
                 region          state  individuals  family_members  state_p
32         Mid-Atlantic       New York      39827.0         52070.0   195303
4               Pacific     California     109008.0         20964.0   394615
21          New England  Massachusetts       6811.0         13257.0    68826
9        South Atlantic        Florida      21443.0          9587.0   212443
43   West South Central          Texas      19199.0          6111.0   286286
```

```
#No.3
homelessness_reg_fam = homelessness.sort_values(["region","family_members"], as
print(homelessness_reg_fam.head())
```

```
                 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
```

```
#No.4
state_fam = homelessness[["state","family_members"]]
print(state_fam.head())
```

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

```
#No.5
int_gt_10k = homelessness[homelessness["individuals"] > 10000]
print(int_gt_10k)
```

```
                 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
```

```
#No.6
mountain_reg = homelessness[homelessness["region"] == "Mountain"]
print(mountain_reg)
```

```
      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
```

```
#No7
fam_It_1k_pac = homelessness[(homelessness["family_members"] < 1000) & (homeles
print(fam_It_1k_pac)
```

```
    region   state  individuals  family_members  state_pop
1  Pacific  Alaska       1434.0           582.0     735139
```

```
#No8
is_SA = homelessness["region"] == "South Atlantic"
is_MA = homelessness["region"] == "Mid-Atlantic"
south_mid_atlantic = homelessness[is_SA | is_MA]
# south_mid_atlantic = homelessness[np.logical_or(is_SA , is_MA)]
(south_mid_atlantic)
```

	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

```
#No.9
mojave_homelessness = homelessness[homelessness["state"].isin(["California", "A
print(mojave_homelessness)
```

```
      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
```

```
#No.10
homelessness["total"] = homelessness["individuals"] + homelessness["family_memb
homelessness
```

	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
5	Mountain	Colorado	7607.0	3250.0	5691287	10857.0
6	New England	Connecticut	2280.0	1696.0	3571520	3976.0
7	South Atlantic	Delaware	708.0	374.0	965479	1082.0
8	South Atlantic	District of Columbia	3770.0	3134.0	701547	6904.0
9	South Atlantic	Florida	21443.0	9587.0	21244317	31030.0
10	South Atlantic	Georgia	6943.0	2556.0	10511131	9499.0
11	Pacific	Hawaii	4131.0	2399.0	1420593	6530.0
12	Mountain	Idaho	1297.0	715.0	1750536	2012.0
13	East North Central	Illinois	6752.0	3891.0	12723071	10643.0
14	East North Central	Indiana	3776.0	1482.0	6695497	5258.0
15	West North Central	Iowa	1711.0	1038.0	3148618	2749.0
16	West North Central	Kansas	1443.0	773.0	2911359	2216.0
17	East South Central	Kentucky	2735.0	953.0	4461153	3688.0
18	West South Central	Louisiana	2540.0	519.0	4659690	3059.0
19	New England	Maine	1450.0	1066.0	1339057	2516.0
20	South Atlantic	Maryland	4914.0	2230.0	6035802	7144.0
21	New England	Massachusetts	6811.0	13257.0	6882635	20068.0

22	East North Central	Michigan	5209.0	3142.0	9984072	8351.0
23	West North Central	Minnesota	3993.0	3250.0	5606249	7243.0
24	East South Central	Mississippi	1024.0	328.0	2981020	1352.0
25	West North Central	Missouri	3776.0	2107.0	6121623	5883.0
26	Mountain	Montana	983.0	422.0	1060665	1405.0
27	West North Central	Nebraska	1745.0	676.0	1925614	2421.0
28	Mountain	Nevada	7058.0	486.0	3027341	7544.0
29	New England	New Hampshire	835.0	615.0	1353465	1450.0
30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025	9398.0
31	Mountain	New Mexico	1949.0	602.0	2092741	2551.0
32	Mid-Atlantic	New York	39827.0	52070.0	19530351	91897.0
33	South Atlantic	North Carolina	6451.0	2817.0	10381615	9268.0
34	West North Central	North Dakota	467.0	75.0	758080	542.0
35	East North Central	Ohio	6929.0	3320.0	11676341	10249.0
36	West South Central	Oklahoma	2823.0	1048.0	3940235	3871.0

```
#No.11
homelessness["p_individuals"] = homelessness["individuals"] / homelessness["tot
homelessness
```

	region	state	individuals	family_members	state_pop	total	p_
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	
	West						

3	South Central	Arkansas	2280.0	432.0	3009733	2712.0
4	Pacific	California	109008.0	20964.0	39461588	129972.0
5	Mountain	Colorado	7607.0	3250.0	5691287	10857.0
6	New England	Connecticut	2280.0	1696.0	3571520	3976.0
7	South Atlantic	Delaware	708.0	374.0	965479	1082.0
8	South Atlantic	District of Columbia	3770.0	3134.0	701547	6904.0
9	South Atlantic	Florida	21443.0	9587.0	21244317	31030.0
10	South Atlantic	Georgia	6943.0	2556.0	10511131	9499.0
11	Pacific	Hawaii	4131.0	2399.0	1420593	6530.0
12	Mountain	Idaho	1297.0	715.0	1750536	2012.0
13	East North Central	Illinois	6752.0	3891.0	12723071	10643.0
14	East North Central	Indiana	3776.0	1482.0	6695497	5258.0
15	West North Central	Iowa	1711.0	1038.0	3148618	2749.0
16	West North Central	Kansas	1443.0	773.0	2911359	2216.0
17	East South Central	Kentucky	2735.0	953.0	4461153	3688.0
18	West South Central	Louisiana	2540.0	519.0	4659690	3059.0
19	New England	Maine	1450.0	1066.0	1339057	2516.0
20	South Atlantic	Maryland	4914.0	2230.0	6035802	7144.0
21	New	Massachusetts	6811.0	13257.0	6882635	20068.0

	England					
22	East North Central	Michigan	5209.0	3142.0	9984072	8351.0
23	West North Central	Minnesota	3993.0	3250.0	5606249	7243.0
24	East South Central	Mississippi	1024.0	328.0	2981020	1352.0
25	West North Central	Missouri	3776.0	2107.0	6121623	5883.0
26	Mountain	Montana	983.0	422.0	1060665	1405.0
27	West North Central	Nebraska	1745.0	676.0	1925614	2421.0
28	Mountain	Nevada	7058.0	486.0	3027341	7544.0
29	New England	New Hampshire	835.0	615.0	1353465	1450.0
30	Mid-Atlantic	New Jersey	6048.0	3350.0	8886025	9398.0
31	Mountain	New Mexico	1949.0	602.0	2092741	2551.0

```
#No.12
homelessness["indiv_per_10k"] = 10000*homelessness["individuals"] / homelessnes

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

high_homelessness_srt = high_homelessness.sort_values("indiv_per_10k", ascendir

result = high_homelessness_srt[["state","indiv_per_10k"]]

result
```

	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

```
sales = pd.read_csv("sales_subset.csv", index_col = 0)

print(sales.head)

print(sales.info())

print(sales["weekly_sales"].mean())

print(sales["weekly_sales"].median())

print(sales["date"].max())

print(sales["date"].min())
```

```
<bound method NDFrame.head of        store type  department        date  we
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
...      ...  ...         ...         ...           ...         ...
10769     39    A          99  2011-12-09        895.00       False
10770     39    A          99  2012-02-03        350.00       False
10771     39    A          99  2012-06-08        450.00       False
10772     39    A          99  2012-07-13          0.06       False
10773     39    A          99  2012-10-05        915.00       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
...              ...                   ...           ...
10769       9.644444              0.834256         7.716
10770      15.938889              0.887619         7.244
10771      27.288889              0.911922         6.989
10772      25.644444              0.860145         6.623
10773      22.250000              0.955511         6.228

[10774 rows x 9 columns]>
<class 'pandas.core.frame.DataFrame'>
Index: 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
23843.95014850566
12049.064999999999
2012-10-26
2010-02-05
```

```
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"]])

sales_1_1
```

```
          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
```

	store	type	department	date	weekly_sales	is_holiday	temperature_c	f
0	1	A	1	2010-02-05	24924.50	False	5.727778	
1	1	A	1	2010-03-05	21827.90	False	8.055556	
2	1	A	1	2010-04-02	57258.43	False	16.816667	
3	1	A	1	2010-05-07	17413.94	False	22.527778	
4	1	A	1	2010-06-04	17558.09	False	27.050000	
5	1	A	1	2010-07-02	16333.14	False	27.172222	
6	1	A	1	2010-08-06	17508.41	False	30.644444	
7	1	A	1	2010-09-03	16241.78	False	27.338889	

```
#No.13
store_types = sales.drop_duplicates(subset = ["store","type"])

(store_types.head())
```

	store	type	department	date	weekly_sales	is_holiday	temperature_c
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
[illegible]	[illegible]	[illegible]	[illegible]	2010-	[illegible]	[illegible]	[illegible]

```
#No.14
store_depts = sales.drop_duplicates(subset = ["store", "department"])

(store_depts.head())
```

	store	type	department	date	weekly_sales	is_holiday	temperature_c	f
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	
[illegible]	[illegible]	[illegible]	[illegible]	2010-	[illegible]	[illegible]	[illegible]	

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

print(dept_count_sorted)
```

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

```
dept_props_sorted = store_depts["department"].value_counts(sort = True, normali
print(dept_props_sorted)
```

```
department
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: proportion, Length: 80, dtype: float64
```

```
#No.15
holiday_dates = sales[sales["is_holiday"]==True].drop_duplicates("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
```

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

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

```
#Calculate 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.       ]
```

```
# For each store type, aggregate weekly_sales: get min, max, mean, and median
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 mir
unemp_fuel_stats = sales.groupby('type')[["unemployment", "fuel_price_usd_per_l

# Print unemp_fuel_stats
(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
/var/folders/0k/sxpkc5jn6336yf8n1_lhlwv00000gn/T/ipykernel_23470/1344652713
  sales_stats = sales.groupby('type')['weekly_sales'].agg(["min", "max", np
/var/folders/0k/sxpkc5jn6336yf8n1_lhlwv00000gn/T/ipykernel_23470/1344652713
  sales_stats = sales.groupby('type')['weekly_sales'].agg(["min", "max", np
/var/folders/0k/sxpkc5jn6336yf8n1_lhlwv00000gn/T/ipykernel_23470/1344652713
  unemp_fuel_stats = sales.groupby('type')[["unemployment", "fuel_price_usd
/var/folders/0k/sxpkc5jn6336yf8n1_lhlwv00000gn/T/ipykernel_23470/1344652713
  unemp_fuel_stats = sales.groupby('type')[["unemployment", "fuel_price_usd
```

	unemployment				**fuel_price_usd_per_l**			
	min	**max**	**mean**	**median**	**min**	**max**	**mean**	**median**
type								
A	3.879	8.992	7.972611	8.067	0.664129	1.107410	0.744619	0.735455
B	7.170	9.765	9.279323	9.199	0.760023	1.107674	0.805858	0.803348

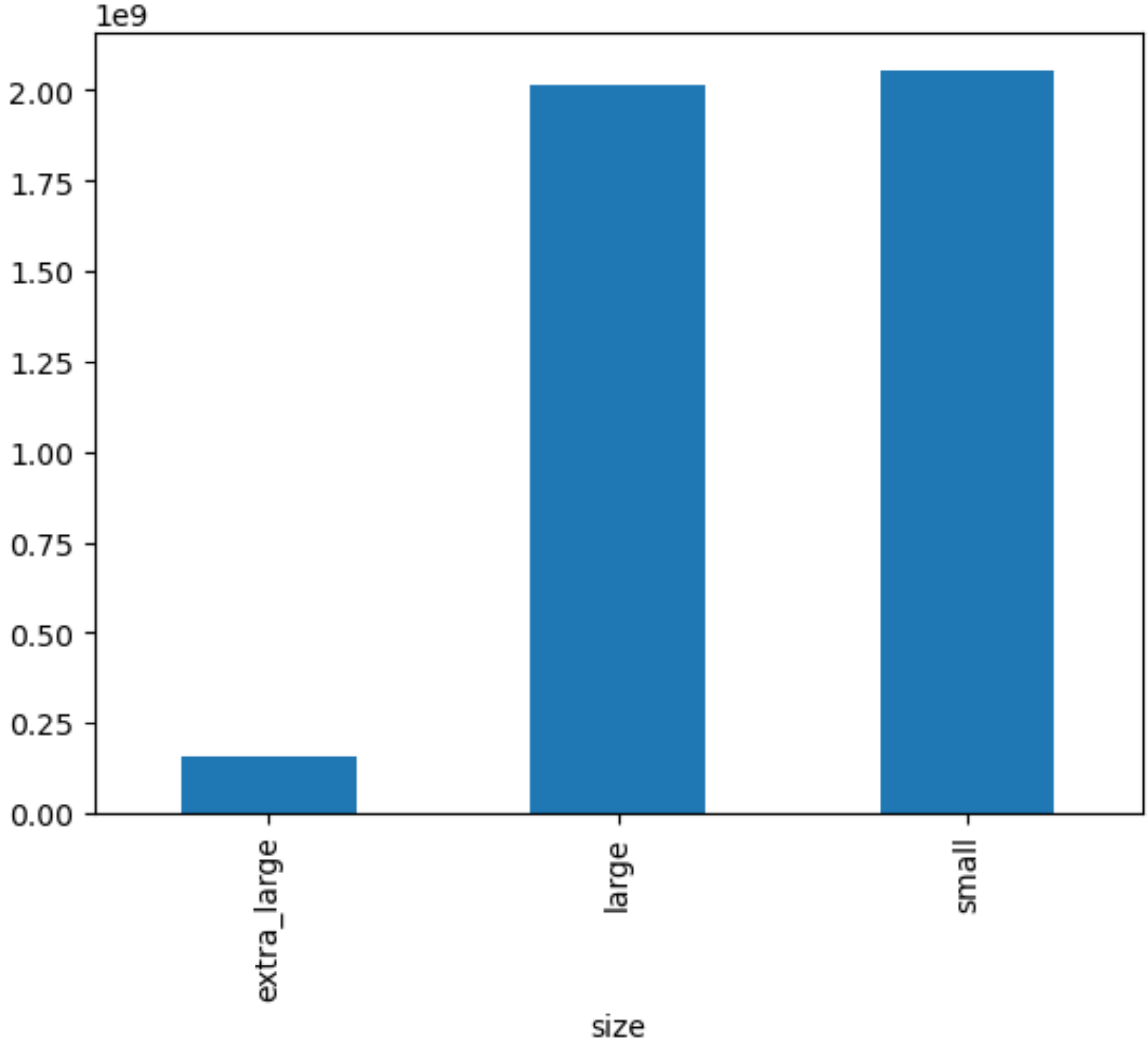
```
temperatures = pd.read_csv("temperatures.csv", index_col= 0)

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])
```

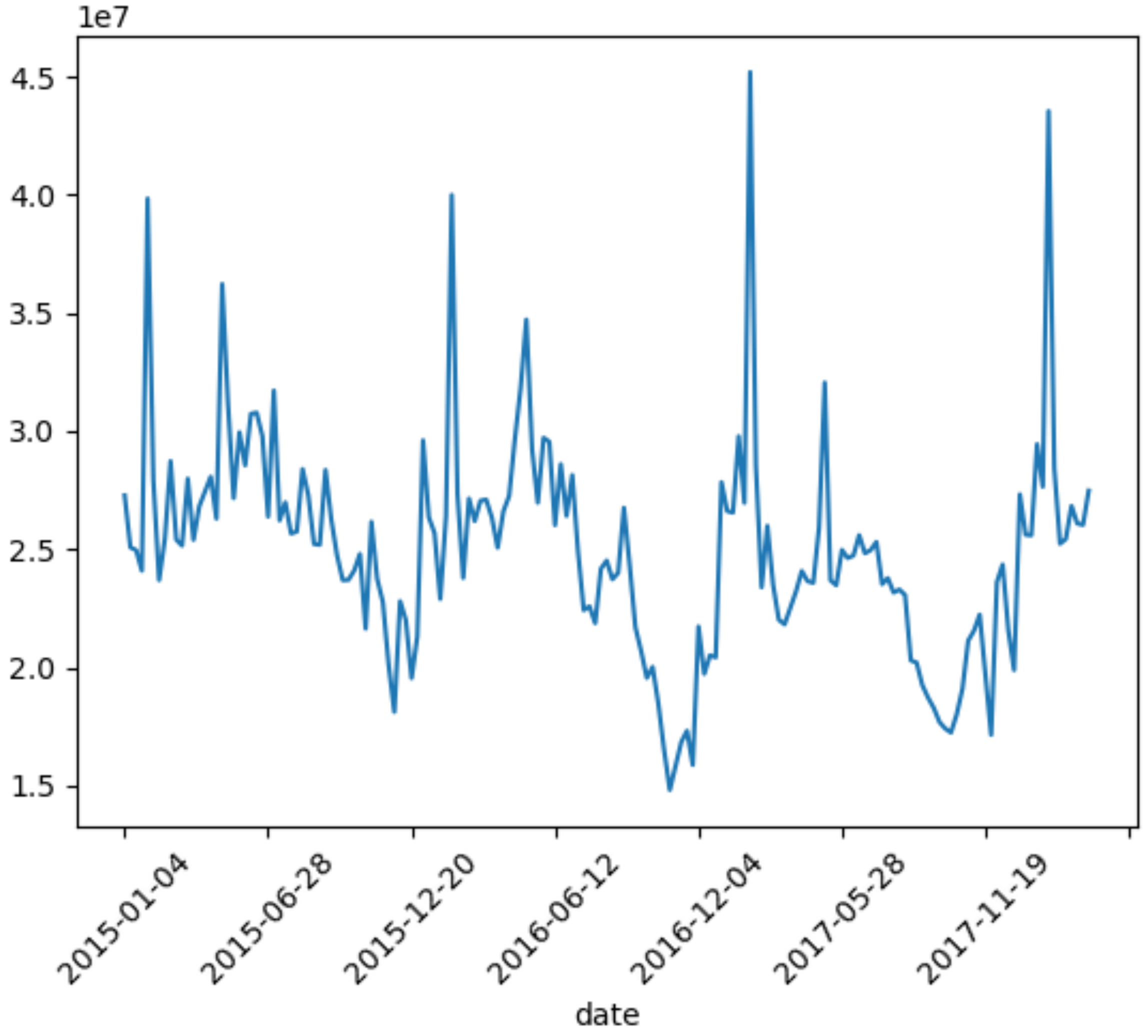
```
Xian     2013-09-01          China         NaN

[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
```

```
avocados = pd.read_csv("avocados.csv")
print(avocados.head())
# Get the total number of avocados sold of each size
nb_sold_by_size = avocados.groupby('size')['nb_sold'].sum()
# Create a bar plot of the number of avocados sold by size
nb_sold_by_size.plot(kind = 'bar')
# Show the plot
plt.show()
```

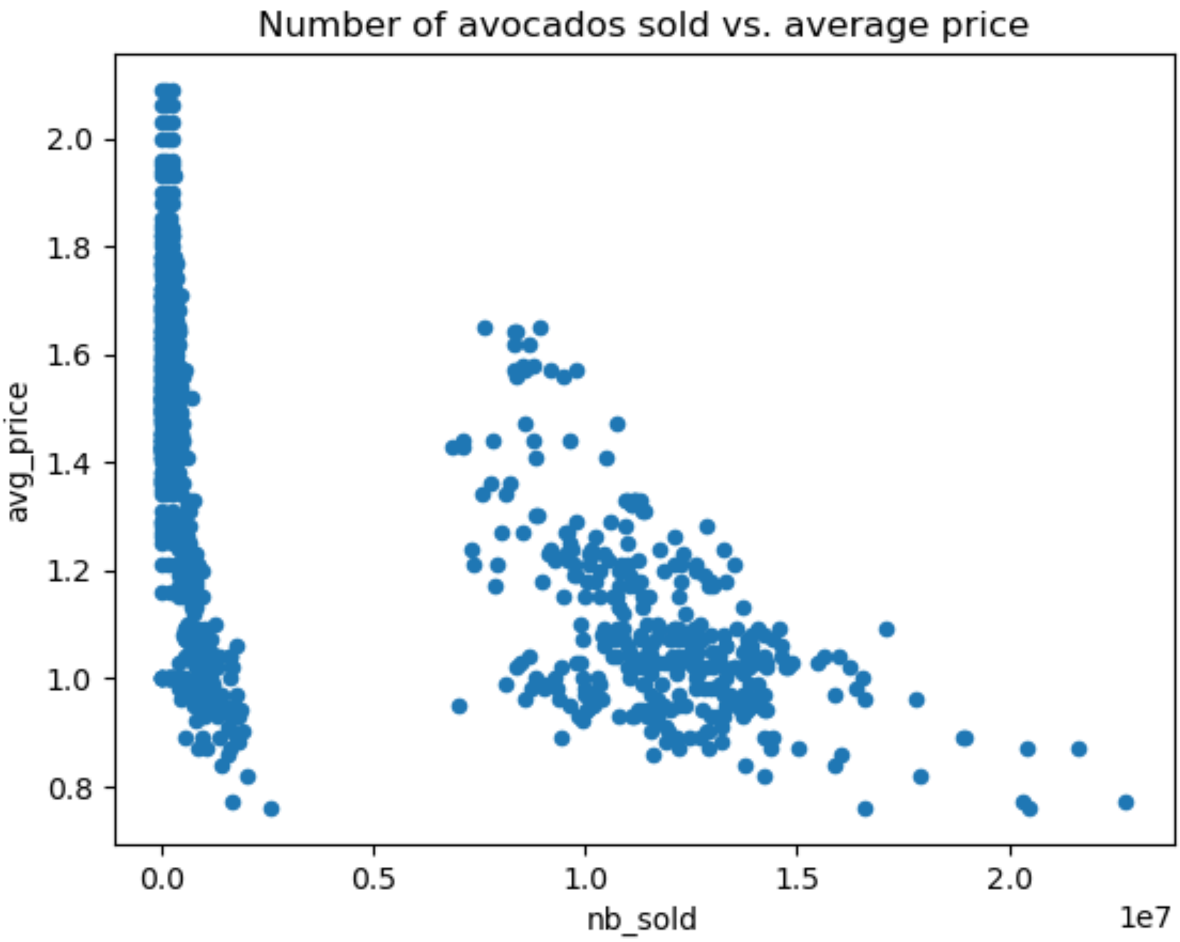
```
   Unnamed: 0        date          type  year  avg_price   size     nb_sold
0           0  2015-12-27  conventional  2015       0.95  small  9626901.09
1           1  2015-12-20  conventional  2015       0.98  small  8710021.76
2           2  2015-12-13  conventional  2015       0.93  small  9855053.66
3           3  2015-12-06  conventional  2015       0.89  small  9405464.36
4           4  2015-11-29  conventional  2015       0.99  small  8094803.56
```



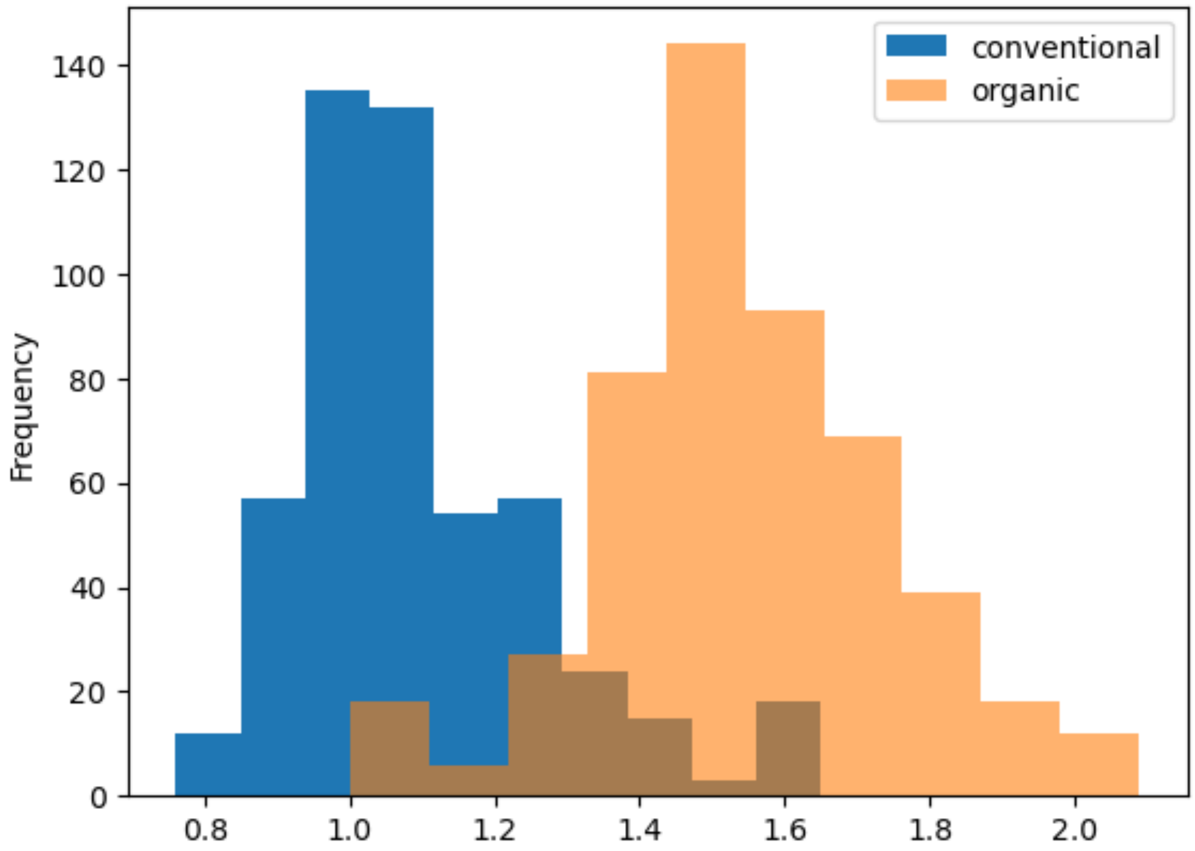
```
# Get the total number of avocados sold on each date
nb_sold_by_date = avocados.groupby('date')['nb_sold'].sum()
# Create a line plot of the number of avocados sold by date
nb_sold_by_date.plot(kind='line', rot = 45)
# Show the plot
plt.show()
```



```
# Scatter plot of avg_price vs. nb_sold with title
avocados.plot(x='nb_sold', y='avg_price',kind = "scatter", title = "Number of a
# Show the plot
plt.show()
```



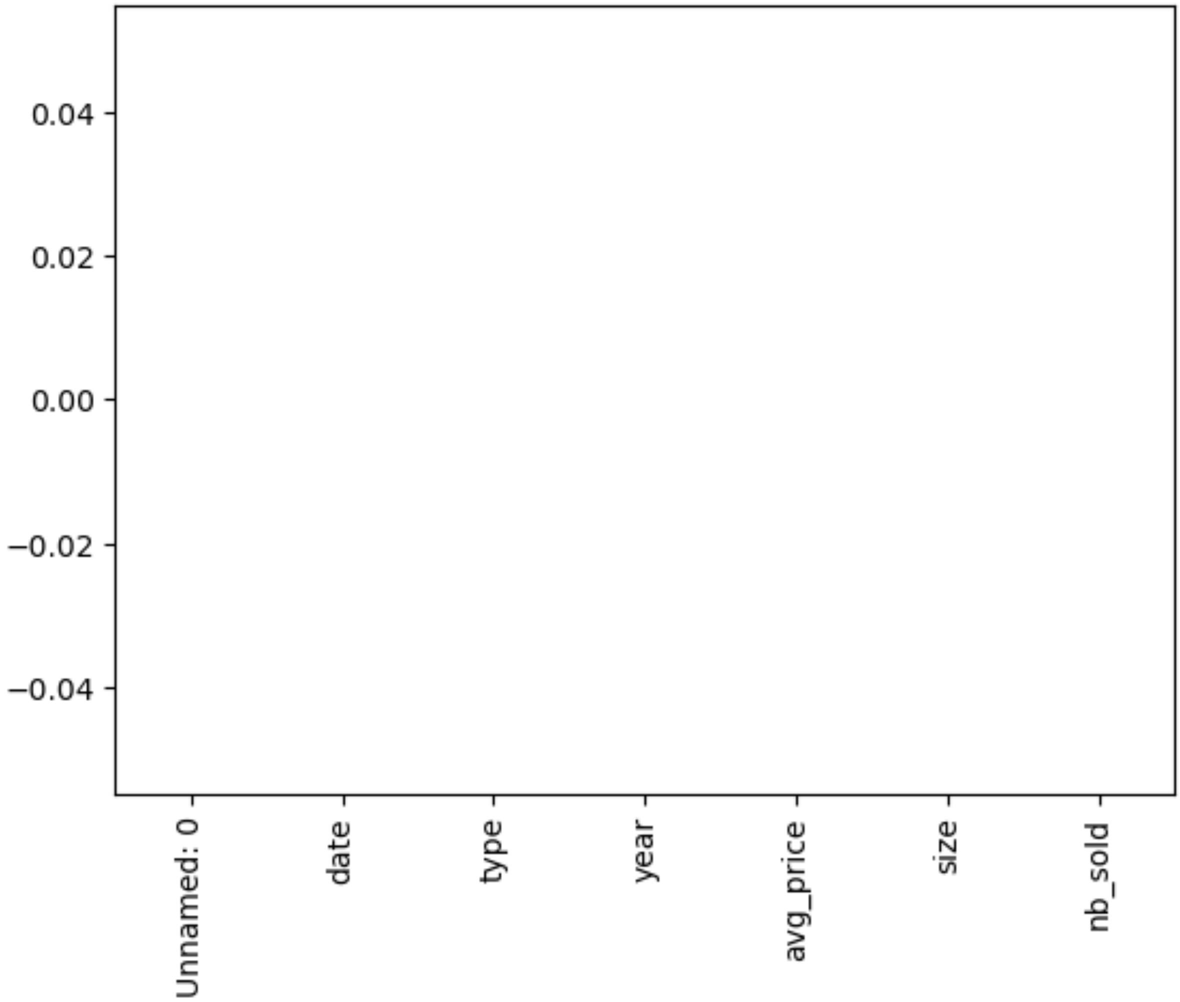
```
# Histogram of conventional avg_price
avocados[avocados['type'] == 'conventional']['avg_price'].plot(kind = 'hist')
# Histogram of organic avg_price
avocados[avocados['type'] == 'organic']['avg_price'].plot(kind = 'hist', alpha
# Add a legend
plt.legend(['conventional','organic'])
# Show the plot
plt.show()
```



```
avocados_2016 = pd.read_csv("avocados_2016.csv", index_col= 0)
# Check individual values for missing values
print(avocados_2016.isna())
# Check each column for missing values
print(avocados_2016.isna().any())
# Bar plot of missing values by variable
avocados_2016.isna().sum().plot(kind = 'bar')
# Show plot
plt.show()
# Remove rows with missing values
avocados_complete = avocados_2016.dropna()
# Check if any columns contain missing values
print(avocados_complete.isna().any())
```

```
        Unnamed: 0   date   type   year  avg_price   size  nb_sold
    52       False  False  False  False      False  False    False
```

```
52        False  False  False  False      False  False    False
53        False  False  False  False      False  False    False
54        False  False  False  False      False  False    False
55        False  False  False  False      False  False    False
56        False  False  False  False      False  False    False
..          ...    ...    ...    ...        ...    ...      ...
944       False  False  False  False      False  False    False
945       False  False  False  False      False  False    False
946       False  False  False  False      False  False    False
947       False  False  False  False      False  False    False
948       False  False  False  False      False  False    False

[312 rows x 7 columns]
Unnamed: 0    False
date          False
type          False
year          False
avg_price     False
size          False
nb_sold       False
dtype: bool
```



```
Unnamed: 0    False
date          False
type          False
year          False
avg_price     False
```

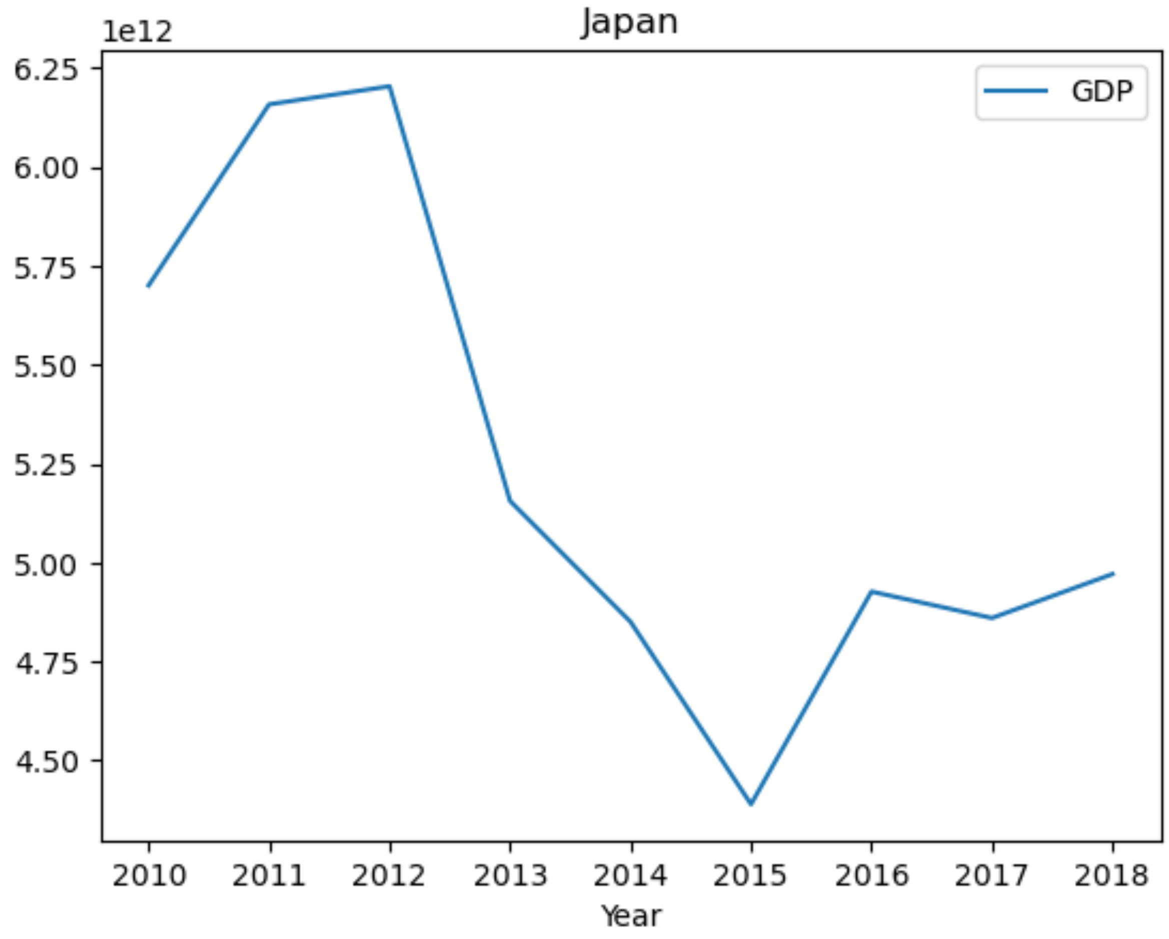
```
size          False
nb_sold       False
dtype: bool
```

```
gdp = pd.read_csv("WorldBank_GDP.csv", index_col = 0)

# worldBankGDP[(worldBankGDP['Year'] == 2010) | worldBankGDP['Year'] == 2018]['

Country = gdp.groupby("Country Name")
gdp
```

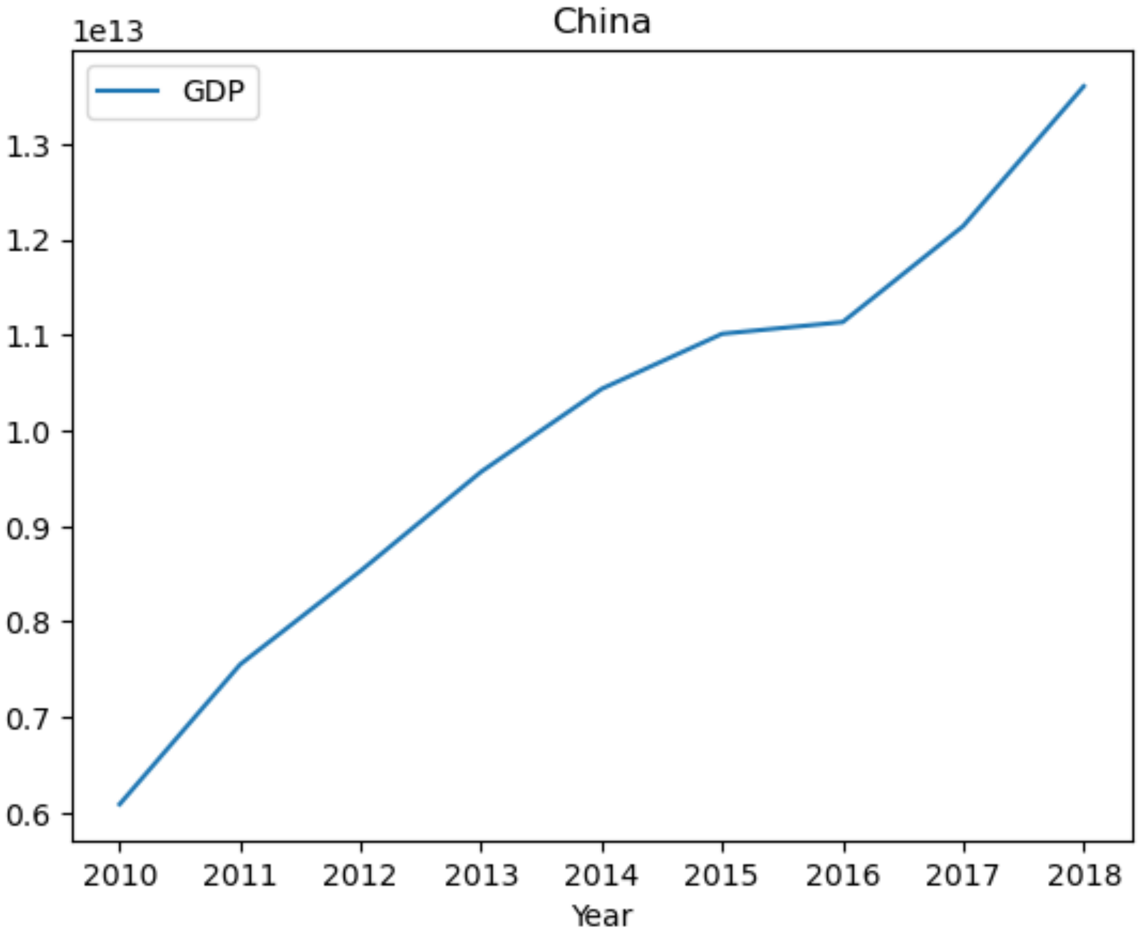
	Country Code	Indicator Name	Year	GDP
Country Name				
China	CHN	GDP (current US$)	2010	6.087160e+12
Germany	DEU	GDP (current US$)	2010	3.417090e+12
Japan	JPN	GDP (current US$)	2010	5.700100e+12
United States	USA	GDP (current US$)	2010	1.499210e+13
China	CHN	GDP (current US$)	2011	7.551500e+12
Germany	DEU	GDP (current US$)	2011	3.757700e+12
Japan	JPN	GDP (current US$)	2011	6.157460e+12
United States	USA	GDP (current US$)	2011	1.554260e+13
China	CHN	GDP (current US$)	2012	8.532230e+12
Germany	DEU	GDP (current US$)	2012	3.543980e+12
Japan	JPN	GDP (current US$)	2012	6.203210e+12
United States	USA	GDP (current US$)	2012	1.619700e+13
China	CHN	GDP (current US$)	2012	8.532230e+12
Germany	DEU	GDP (current US$)	2012	3.543980e+12
Japan	JPN	GDP (current US$)	2012	6.203210e+12
United States	USA	GDP (current US$)	2012	1.619700e+13
China	CHN	GDP (current US$)	2013	9.570410e+12
Germany	DEU	GDP (current US$)	2013	3.752510e+12
Japan	JPN	GDP (current US$)	2013	5.155720e+12
United States	USA	GDP (current US$)	2013	1.678480e+13
China	CHN	GDP (current US$)	2014	1.043850e+13

```
         China      CHN   GDP (current US$)   2014   [illegible]
       Germany      DEU   GDP (current US$)   2014   3.898730e+12
         Japan      JPN   GDP (current US$)   2014   4.850410e+12
 United States      USA   GDP (current US$)   2014   1.752170e+13
         China      CHN   GDP (current US$)   2015   1.101550e+13
       Germany      DEU   GDP (current US$)   2015   3.381390e+12
         Japan      JPN   GDP (current US$)   2015   4.389480e+12
 United States      USA   GDP (current US$)   2015   1.821930e+13
         China      CHN   GDP (current US$)   2016   1.113790e+13
       Germany      DEU   GDP (current US$)   2016   3.495160e+12
         Japan      JPN   GDP (current US$)   2016   4.926670e+12
 United States      USA   GDP (current US$)   2016   1.870720e+13
         China      CHN   GDP (current US$)   2017   1.214350e+13
       Germany      DEU   GDP (current US$)   2017   3.693200e+12
         Japan      JPN   GDP (current US$)   2017   4.859950e+12
 United States      USA   GDP (current US$)   2017   1.948540e+13
         China      CHN   GDP (current US$)   2018   1.360820e+13
       Germany      DEU   GDP (current US$)   2018   3.996760e+12
         Japan      JPN   GDP (current US$)   2018   4.970920e+12
 United States      USA   GDP (current US$)   2018   2.049410e+13
```

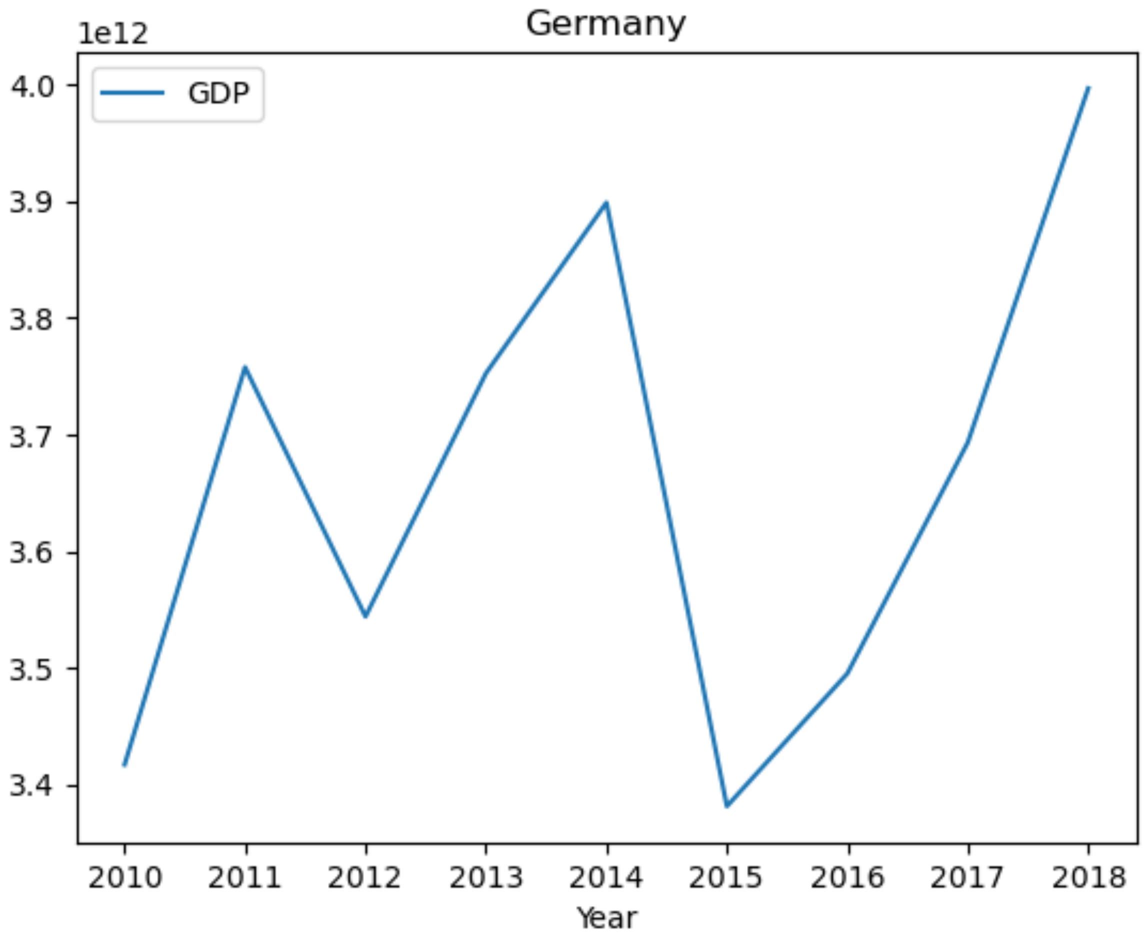
```
gdp_japan = gdp[gdp["Country Code"] == "JPN"]

gdp_japan.plot(x="Year", y="GDP", kind="line")
plt.title("Japan")
plt.show()
```



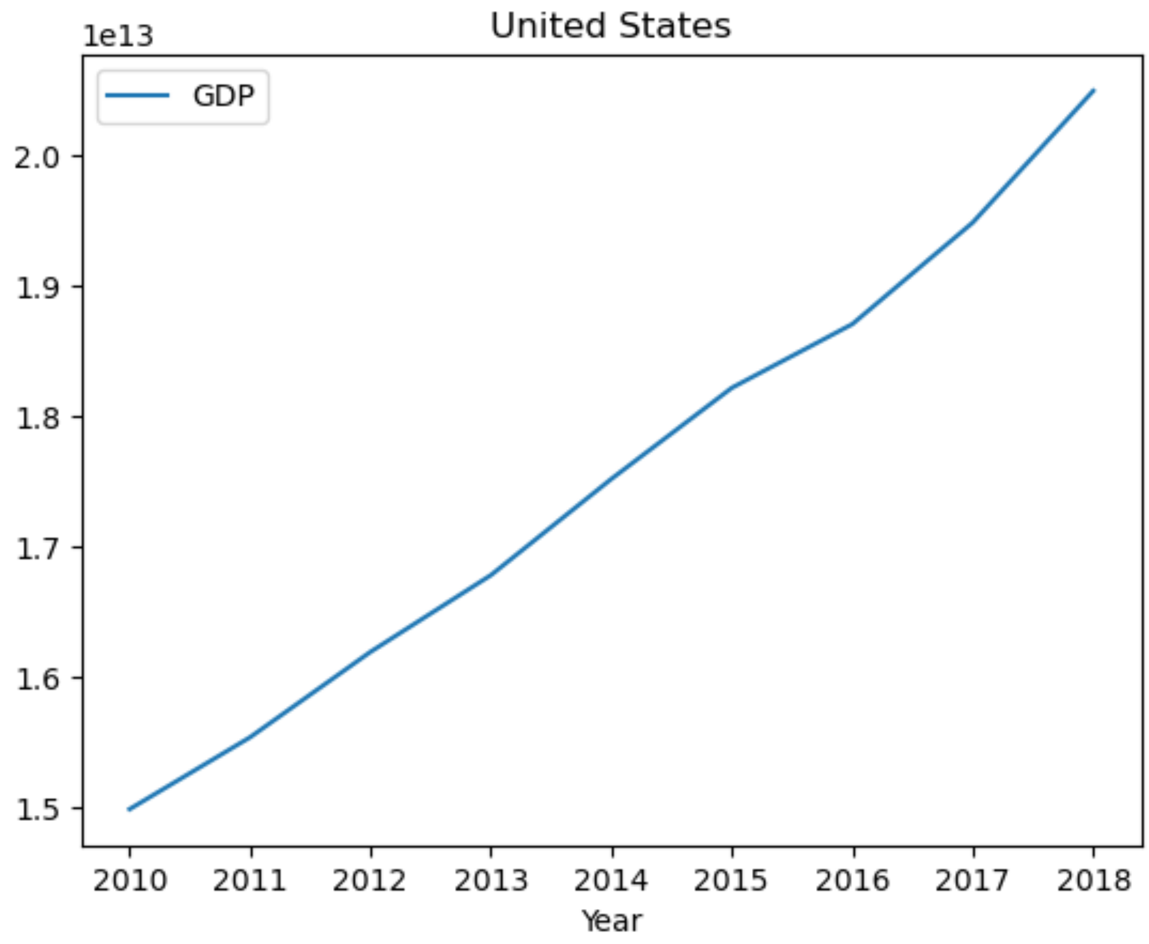
```
gdp_japan = gdp[gdp["Country Code"] == "CHN"]

gdp_japan.plot(x="Year", y="GDP", kind="line")
plt.title("China")
plt.show()
```



```
gdp_japan = gdp[gdp["Country Code"] == "DEU"]

gdp_japan.plot(x="Year", y="GDP", kind="line")
plt.title("Germany")
plt.show()
```



```
gdp_japan = gdp[gdp["Country Code"] == "USA"]

gdp_japan.plot(x="Year", y="GDP", kind="line")
plt.title("United States")
plt.show()
```



```
temp = pd.read_csv('temperatures.csv', index_col = 0)
```

```
temp
```

	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 × 4 columns

```
temp_avg = temp.groupby('country')["avg_temp_c"].mean().reset_index()

print(temp_avg)
```

```
                               country  avg_temp_c
0                          Afghanistan   15.525756
1                               Angola   24.387659
2                            Australia   16.028104
3                           Bangladesh   26.164378
4                               Brazil   23.906030
5                                Burma   27.514213
6                               Canada    6.637158
7                                Chile    6.345768
8                                China   12.983107
9                             Colombia   21.649607
10  Congo (Democratic Republic Of The)   24.504963
11                       Côte D'Ivoire   26.971024
12                  Dominican Republic   26.852800
13                               Egypt   22.044807
14                            Ethiopia   18.425378
15                              France   11.514274
16                             Germany   10.152421
17                               India   26.633255
18                           Indonesia   27.408634
19                                Iran   14.228701
20                                Iraq   24.074841
21                               Italy   13.127646
22                               Japan   14.526165
23                               Kenya   16.817134
24                              Mexico   16.406630
25                             Morocco   18.336195
26                             Nigeria   27.176191
27                            Pakistan   25.824654
28                                Peru   17.203762
29                         Philippines   27.153518
30                              Russia    5.557576
31                        Saudi Arabia   27.635610
32                             Senegal   25.425994
33                           Singapore   27.323165
34                             Somalia   27.963183
35                        South Africa   18.913680
36                         South Korea   11.693262
37                               Spain   12.460860
38                               Sudan   29.981780
39                               Syria   18.501244
40                              Taiwan   23.078829
41                            Tanzania   26.481774
42                            Thailand   27.929518
43                              Turkey   14.799793
44                             Ukraine    8.701683
45                      United Kingdom   10.523585
46                       United States   12.954515
47                             Vietnam   27.909878
48                            Zimbabwe   20.721988
```

```
#Ex.1 temp
print(temp_avg.max())
```

```
country       Zimbabwe
avg_temp_c    29.98178
dtype: object
```

```
#Ex.2 temp
temp_country2030 = temp_avg[(temp_avg['avg_temp_c'] >= 20) & (temp_avg["avg_tem

print(temp_country2030)

print(temp_country2030.count())
```

```
                               country  avg_temp_c
1                               Angola   24.387659
3                           Bangladesh   26.164378
4                               Brazil   23.906030
5                                Burma   27.514213
9                             Colombia   21.649607
10  Congo (Democratic Republic Of The)   24.504963
11                       Côte D'Ivoire   26.971024
12                  Dominican Republic   26.852800
13                               Egypt   22.044807
17                               India   26.633255
18                           Indonesia   27.408634
20                                Iraq   24.074841
26                             Nigeria   27.176191
27                            Pakistan   25.824654
29                         Philippines   27.153518
31                        Saudi Arabia   27.635610
32                             Senegal   25.425994
33                           Singapore   27.323165
34                             Somalia   27.963183
38                               Sudan   29.981780
40                              Taiwan   23.078829
41                            Tanzania   26.481774
42                            Thailand   27.929518
47                             Vietnam   27.909878
48                            Zimbabwe   20.721988
country       25
avg_temp_c    25
dtype: int64
```

```
#No.3
temp_thailand = temp[temp["country"]=="Thailand"]
temp_thailand_20052010 = temp_thailand[(temp_thailand['date'] >= "2005-01-01")
temp_thailand_20052010
```

	date	city	country	avg_temp_c
1380	2005-01-01	Bangkok	Thailand	25.323
1381	2005-02-01	Bangkok	Thailand	28.225
1382	2005-03-01	Bangkok	Thailand	28.825
1383	2005-04-01	Bangkok	Thailand	30.210
1384	2005-05-01	Bangkok	Thailand	30.023
...	...	...	...	...
1436	2009-09-01	Bangkok	Thailand	28.308
1437	2009-10-01	Bangkok	Thailand	27.564
1438	2009-11-01	Bangkok	Thailand	26.533
1439	2009-12-01	Bangkok	Thailand	25.973
1440	2010-01-01	Bangkok	Thailand	26.615

61 rows × 4 columns

```
avg_temp_thailand = temp_thailand_20052010["avg_temp_c"].mean()

print(f"The average temp of thailand during 2005-2010 is {avg_temp_thailand: .2
```

```
The average temp of thailand during 2005-2010 is  27.76 Celcius
```