Intermediate Pandas Python Library for Data Science

Instructor: Vinita Silaparasetty

Import Libraries

In [1]:
```
import numpy as np                  #import numpy as np for convenience.
import pandas as pd                 #import pandas as pd for convenience.
```

Load Data

In [2]:
```
df=pd.read_csv("https://raw.githubusercontent.com/VinitaSilaparasetty/Coursera
-Intermediate-Pandas/master/What_does_aid_to_Africa_finance_1.csv ")

df.head(3) #Ensure data has loaded correctly.
```

Out[2]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp	
0	Burkina Faso	1970	35.44188862	5633000.0	141.3999939	13	1	13.3182802200317	1
1	Burkina Faso	1970	35.44188862	5633000.0	141.3999939	13	1	13.3182802200317	1
2	Burkina Faso	1971	36.16739069	5740700.0	139.1999969	13.6	1.2	16.7043991088867	0.6

3 rows × 50 columns

Splitting Data

In [6]:
```
#splitting our data into 4 subsets

df_new = df.copy()
#extract 25% of our data not at random
df1 = df_new.sample(frac=0.25, random_state=0)
#drop values that have been assigned to df1
df_new = df_new.drop(df1.index)

df2 = df_new.sample(frac=0.25, random_state=0)
df_new = df_new.drop(df2.index)

df3 = df_new.sample(frac=0.25, random_state=0)
#since this is the only remaining data, not necessary to sample
df4 = df_new.drop(df3.index)
```

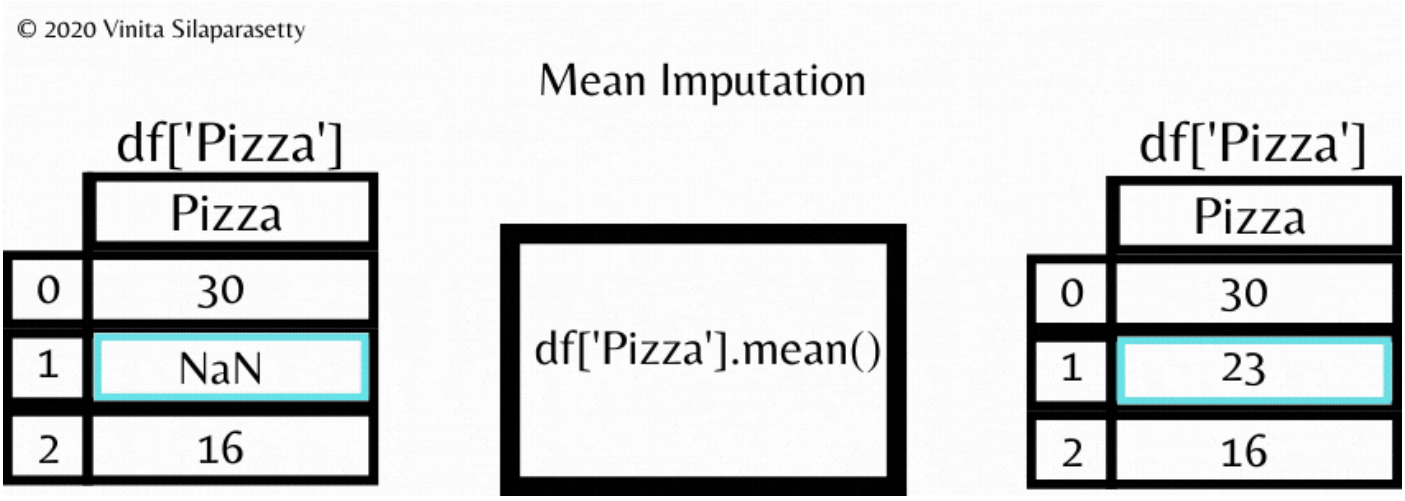
Handle Missing Values

Detect Missing Values

In [8]:
```
print(df3.isnull().sum())
```

```
countryc     0
year         0
agrgdp       0
popn         1
infmort      0
schprim      0
schsec       0
grtdsbp      0
grlndsbp     0
aiddsbp      0
totexpp      0
agexpp       0
enexpp       0
indexpp      0
tacexpp      0
eduexpp      0
hthexpp      0
pirepp       0
curexpp      0
capexpp      0
gdnpp        0
d0           0
cnlnagp      0
cnlnenp      0
cnlninp      0
cnlntacp     0
cnlnedup     0
cnlnhthp     0
cnlnothp     0
dgrtdsbp     0
dgrlndsbp    0
daiddsbp     0
dtotexpp     0
dagexpp      0
denexpp      0
dindexpp     0
dtacexpp     0
deduexpp     0
dhthexpp     0
dothexpp     0
dcurexpp     0
dcapexpp     0
dpirepp      0
dcnlnagp     0
dcnlnenp     0
dcnlninp     0
dcnlntacp    0
dcnlnedup    0
dcnlnhthp    0
dcnlnothp    0
dtype: int64
```

Impute Missing Values



In [9]:
```
df3['popn'].mean()
```

Out[9]:
```
12570541.658536585
```

In [10]:
```
df3.isnull()
```

Out[10]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp	grlndsbp	aiddsbp	...
237	False	False	False	False	False	False	False	False	False	False	...
244	False	False	False	False	False	False	False	False	False	False	...
299	False	False	False	False	False	False	False	False	False	False	...
87	False	False	False	False	False	False	False	False	False	False	...
91	False	False	False	False	False	False	False	False	False	False	...
260	False	False	False	False	False	False	False	False	False	False	...
14	False	False	False	False	False	False	False	False	False	False	...
157	False	False	False	False	False	False	False	False	False	False	...
207	False	False	False	False	False	False	False	False	False	False	...
160	False	False	False	False	False	False	False	False	False	False	...
255	False	False	False	False	False	False	False	False	False	False	...
9	False	False	False	False	False	False	False	False	False	False	...
138	False	False	False	False	False	False	False	False	False	False	...
181	False	False	False	False	False	False	False	False	False	False	...
88	False	False	False	False	False	False	False	False	False	False	...
33	False	False	False	False	False	False	False	False	False	False	...
102	False	False	False	False	False	False	False	False	False	False	...
164	False	False	False	False	False	False	False	False	False	False	...
100	False	False	False	False	False	False	False	False	False	False	...
104	False	False	False	False	False	False	False	False	False	False	...
275	False	False	False	False	False	False	False	False	False	False	...
4	False	False	False	False	False	False	False	False	False	False	...
171	False	False	False	False	False	False	False	False	False	False	...
289	False	False	False	True	False	False	False	False	False	False	...
180	False	False	False	False	False	False	False	False	False	False	...
61	False	False	False	False	False	False	False	False	False	False	...
110	False	False	False	False	False	False	False	False	False	False	...
281	False	False	False	False	False	False	False	False	False	False	...
75	False	False	False	False	False	False	False	False	False	False	...
43	False	False	False	False	False	False	False	False	False	False	...
54	False	False	False	False	False	False	False	False	False	False	...
213	False	False	False	False	False	False	False	False	False	False	...
71	False	False	False	False	False	False	False	False	False	False	...
16	False	False	False	False	False	False	False	False	False	False	...
203	False	False	False	False	False	False	False	False	False	False	...

	countryc	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp	grlndsbp	aiddsbp	...
153	False	False	False	False	False	False	False	False	False	False	...
161	False	False	False	False	False	False	False	False	False	False	...
273	False	False	False	False	False	False	False	False	False	False	...
279	False	False	False	False	False	False	False	False	False	False	...
40	False	False	False	False	False	False	False	False	False	False	...
50	False	False	False	False	False	False	False	False	False	False	...
193	False	False	False	False	False	False	False	False	False	False	...

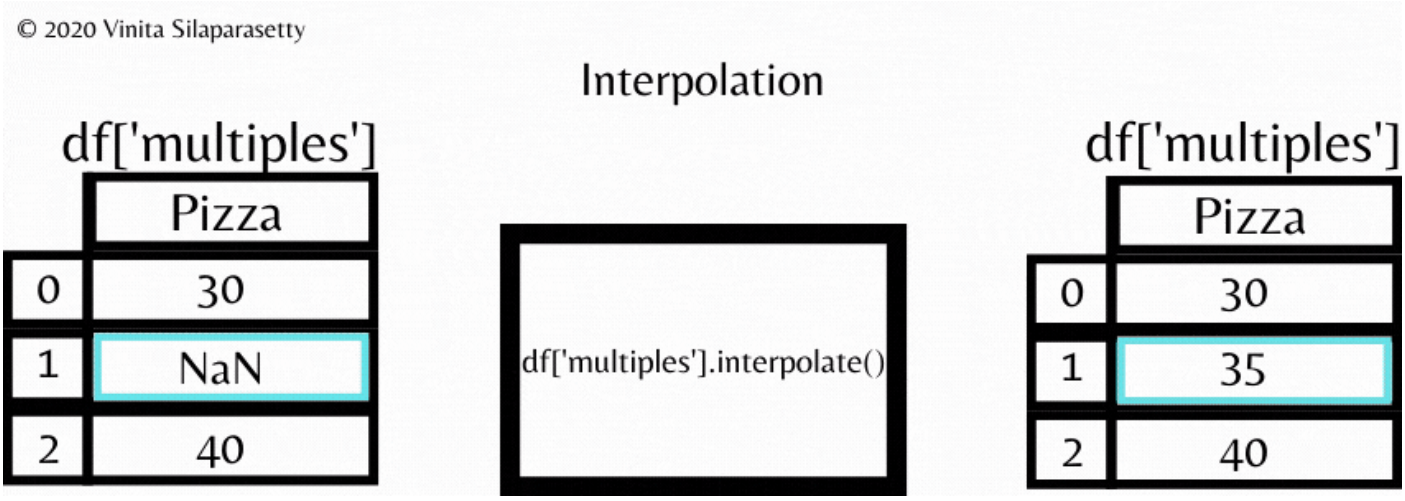
42 rows × 50 columns

In [11]:
```
#replaces the missing values of popn with the mean
df3['popn'].fillna(df3['popn'].mean(), inplace=True)
```

In [12]:
```
print(df3.isnull().sum())
```

```
countryc     0
year         0
agrgdp       0
popn         0
infmort      0
schprim      0
schsec       0
grtdsbp      0
grlndsbp     0
aiddsbp      0
totexpp      0
agexpp       0
enexpp       0
indexpp      0
tacexpp      0
eduexpp      0
hthexpp      0
pirepp       0
curexpp      0
capexpp      0
gdnpp        0
d0           0
cnlnagp      0
cnlnenp      0
cnlninp      0
cnlntacp     0
cnlnedup     0
cnlnhthp     0
cnlnothp     0
dgrtdsbp     0
dgrlndsbp    0
daiddsbp     0
dtotexpp     0
dagexpp      0
denexpp      0
dindexpp     0
dtacexpp     0
deduexpp     0
dhthexpp     0
dothexpp     0
dcurexpp     0
dcapexpp     0
dpirepp      0
dcnlnagp     0
dcnlnenp     0
dcnlninp     0
dcnlntacp    0
dcnlnedup    0
dcnlnhthp    0
dcnlnothp    0
dtype: int64
```

Interpolate Missing Values



In [13]:
```
#interpolation refers to the multiples of 5, not average
print(df1.isnull().sum())
```

```
countryc     0
year         0
agrgdp       0
popn         1
infmort      0
schprim      0
schsec       0
grtdsbp      0
grlndsbp     0
aiddsbp      0
totexpp      0
agexpp       0
enexpp       0
indexpp      0
tacexpp      0
eduexpp      0
hthexpp      0
pririepp     0
curexpp      0
capexpp      0
gdnpp        0
d0           0
cnlnagp      0
cnlnenp      0
cnlninp      0
cnlntacp     0
cnlnedup     0
cnlnhthp     0
cnlnothp     0
dgrtdsbp     0
dgrlndsbp    0
daiddsbp     0
dtotexpp     0
dagexpp      0
denexpp      0
dindexpp     0
dtacexpp     0
deduexpp     0
dhthexpp     0
dothexpp     0
dcurexpp     0
dcapexpp     0
dprirepp     0
dcnlnagp     0
dcnlnenp     0
dcnlninp     0
dcnlntacp    0
dcnlnedup    0
dcnlnhthp    0
dcnlnothp    0
dtype: int64
```

In [15]:
```
df1['popn'].fillna(df1['popn'].interpolate(), inplace=True)
```

In [16]:
```
df1.isnull().sum()
```

Out[16]:
```
countryc     0
year         0
agrgdp       0
popn         0
infmort      0
schprim      0
schsec       0
grtdsbp      0
grlndsbp     0
aiddsbp      0
totexpp      0
agexpp       0
enexpp       0
indexpp      0
tacexpp      0
eduexpp      0
hthexpp      0
prirepp      0
curexpp      0
capexpp      0
gdnpp        0
d0           0
cnlnagp      0
cnlnenp      0
cnlninp      0
cnlntacp     0
cnlnedup     0
cnlnhthp     0
cnlnothp     0
dgrtdsbp     0
dgrlndsbp    0
daiddsbp     0
dtotexpp     0
dagexpp      0
denexpp      0
dindexpp     0
dtacexpp     0
deduexpp     0
dhthexpp     0
dothexpp     0
dcurexpp     0
dcapexpp     0
dprirepp     0
dcnlnagp     0
dcnlnenp     0
dcnlninp     0
dcnlntacp    0
dcnlnedup    0
dcnlnhthp    0
dcnlnothp    0
dtype: int64
```

In [ ]:
```
#interpolation is used when there's some linear relationship in the data
#If there isn't , imputation is used
```

🏆 Challenge

Detect missing values in df2 and decide on the best method to handle them with respect to infant mortality rate.

In [18]:
```
#detect missing values
#infant mortality rate has a linear relationship w population according to the
data
df2.isnull().sum()
```

Out[18]:
```
countryc     0
year         0
agrgdp       0
popn         1
infmort      0
schprim      0
schsec       0
grtdsbp      0
grlndsbp     0
aiddsbp      0
totexpp      0
agexpp       0
enexpp       0
indexpp      0
tacexpp      0
eduexpp      0
hthexpp      0
pririepp     0
curexpp      0
capexpp      0
gdnpp        0
d0           0
cnlnagp      0
cnlnenp      0
cnlninp      0
cnlntacp     0
cnlnedup     0
cnlnhthp     0
cnlnothp     0
dgrtdsbp     0
dgrlndsbp    0
daiddsbp     0
dtotexpp     0
dagexpp      0
denexpp      0
dindexpp     0
dtacexpp     0
deduexpp     0
dhthexpp     0
dothexpp     0
dcurexpp     0
dcapexpp     0
dprirepp     0
dcnlnagp     0
dcnlnenp     0
dcnlninp     0
dcnlntacp    0
dcnlnedup    0
dcnlnhthp    0
dcnlnothp    0
dtype: int64
```

In [19]:
```
#handle missing values
df2['popn'].fillna(df2['popn'].interpolate(), inplace=True)
```

In [20]:
```
df2.isnull().sum()
```

Out[20]:
```
countryc     0
year         0
agrgdp       0
popn         0
infmort      0
schprim      0
schsec       0
grtdsbp      0
grlndsbp     0
aiddsbp      0
totexpp      0
agexpp       0
enexpp       0
indexpp      0
tacexpp      0
eduexpp      0
hthexpp      0
prirepp      0
curexpp      0
capexpp      0
gdnpp        0
d0           0
cnlnagp      0
cnlnenp      0
cnlninp      0
cnlntacp     0
cnlnedup     0
cnlnhthp     0
cnlnothp     0
dgrtdsbp     0
dgrlndsbp    0
daiddsbp     0
dtotexpp     0
dagexpp      0
denexpp      0
dindexpp     0
dtacexpp     0
deduexpp     0
dhthexpp     0
dothexpp     0
dcurexpp     0
dcapexpp     0
dprirepp     0
dcnlnagp     0
dcnlnenp     0
dcnlninp     0
dcnlntacp    0
dcnlnedup    0
dcnlnhthp    0
dcnlnothp    0
dtype: int64
```

Combining Data

Joining

In [22]:
```
df5 = df1.join(df2, lsuffix="_left")
#_left added to columns that're on the left handside dataframe
#the NaN values simply cuz df1 is larger than df2
df5

#join operates only on the columns
```

Out[22]:

	countryc_left	year_left	agrgdp_left	popn_left	infmort_left	schprim_left	schsec_left	
223	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87
150	Gambia, The	1988	31.22936246	841250.0	140.7799988	65	15.25	11
226	Lesotho	1986	21.14252061	1603960.0	92	109	23.4	66
296	Malawi	1976	39.20110669	5409980.0	179.8	56	4	1
52	Botswana	1994	5.199306759	1420270.0	55.39999898	117	53	58
...	...	...	...	...	...	...	...	
20	Burkina Faso	1988	48.94457166	8534390.0	107.8	33.5	6.5	35
46	Botswana	1988	7.060807251	1195140.0	56.6	111.75	35.75	13
158	Kenya	1970	33.29286623	11498000.0	102	58	9	9.
230	Lesotho	1990	19.96355858	1783000.0	84.6	105	25	67
179	Kenya	1991	28.14106137	24015140.0	61.4	93	28	2

75 rows × 100 columns

Concatenation

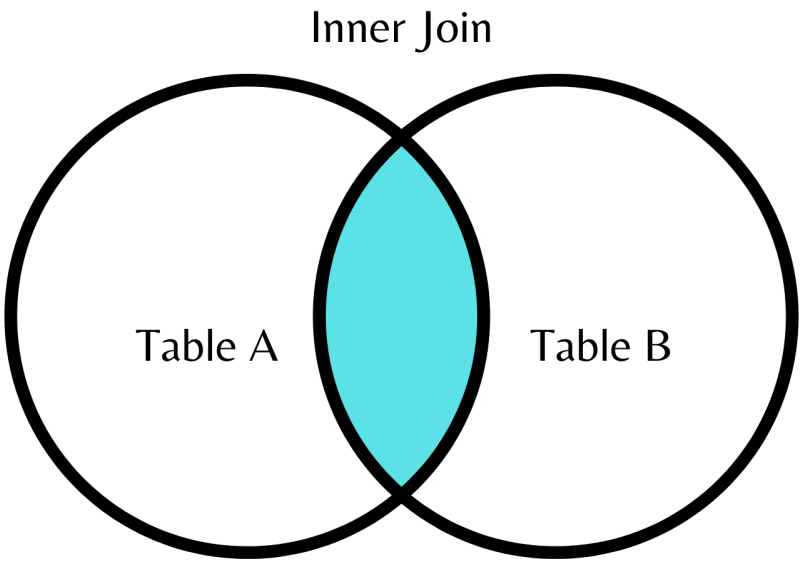
In [27]:
```
df6 = pd.concat([df1, df2], axis=0)
df6
#concat operates on either, depending on axis specified
#0 - rows; 1 - columns
```

Out[27]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	
223	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.476
150	Gambia, The	1988	31.22936246	841250.0	140.7799988	65	15.25	113.24
226	Lesotho	1986	21.14252061	1603960.0	92	109	23.4	66.716
296	Malawi	1976	39.20110669	5409980.0	179.8	56	4	14.33
52	Botswana	1994	5.199306759	1420270.0	55.39999898	117	53	58.704
...	...	...	...	...	...	...	...	
240	Madagascar	1974	34.22234966	7408570.0	163.2	94	12	16.473
256	Madagascar	1990	32.30721538	11672000.0	101.1600006	87	17	42.646
98	Ethiopia	1988	49.15748278	47643232.0	129.4	34.25	13.5	20.359
23	Burkina Faso	1991	34.66403162	9269910.0	104.2	37	8	38.100
191	Liberia	1977	31.87008374	1708160.0	167	60.66666667	18.33333333	20.350

131 rows × 50 columns

Advanced Joins

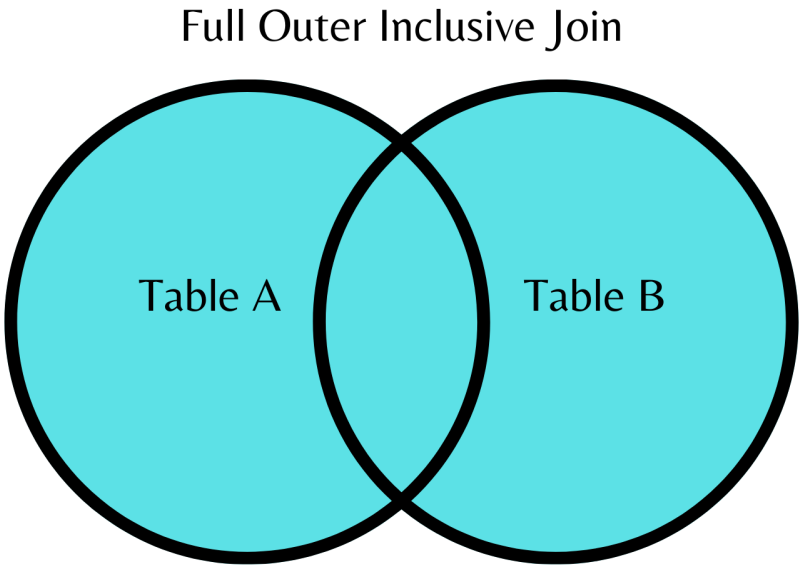


In [30]:
```
#the on attribute is the attribute in which we join, IOW the common attribute
df7 = pd.merge(df1, df2, on="countryc")
#df1 cols - suffix _x; df2 _y
df7
```

Out[30]:

	countryc	year_x	agrgdp_x	popn_x	infmort_x	schprim_x	schsec_x	grtdsl
0	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.476272583
1	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.476272583
2	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.476272583
3	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.476272583
4	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.476272583
...	...	...	...	...	...	...	...	
356	Ghana	1970	46.51883327	8614000.0	110.5999985	64	14	8.4712486267
357	Ghana	1970	46.51883327	8614000.0	110.5999985	64	14	8.4712486267
358	Ghana	1970	46.51883327	8614000.0	110.5999985	64	14	8.4712486267
359	Ghana	1970	46.51883327	8614000.0	110.5999985	64	14	8.4712486267
360	Ghana	1970	46.51883327	8614000.0	110.5999985	64	14	8.4712486267

361 rows × 99 columns

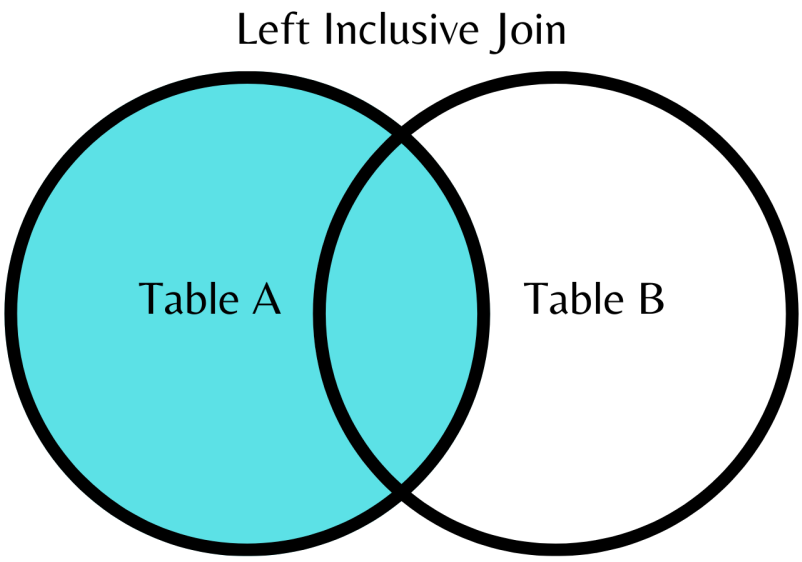


In [31]:
```
#how - type of join
df8 = pd.merge(df1, df2, how='outer')
df8
```

Out[31]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	
0	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.476
1	Gambia, The	1988	31.22936246	841250.0	140.7799988	65	15.25	113.24
2	Lesotho	1986	21.14252061	1603960.0	92	109	23.4	66.716
3	Malawi	1976	39.20110669	5409980.0	179.8	56	4	14.33
4	Botswana	1994	5.199306759	1420270.0	55.39999898	117	53	58.704
...	...	...	...	...	...	...	...	
126	Madagascar	1974	34.22234966	7408570.0	163.2	94	12	16.473
127	Madagascar	1990	32.30721538	11672000.0	101.1600006	87	17	42.646
128	Ethiopia	1988	49.15748278	47643232.0	129.4	34.25	13.5	20.359
129	Burkina Faso	1991	34.66403162	9269910.0	104.2	37	8	38.100
130	Liberia	1977	31.87008374	1708160.0	167	60.66666667	18.33333333	20.350

131 rows × 50 columns

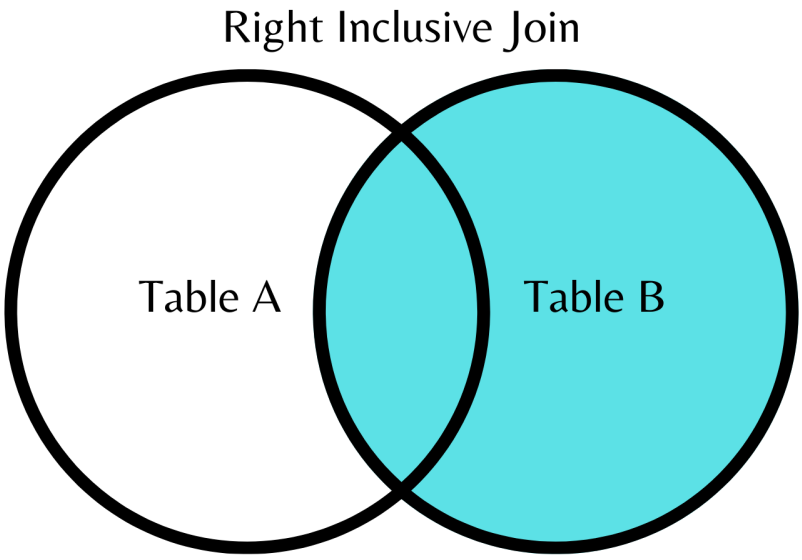


In [32]:
```
df9 = pd.merge(df1, df2, how='left')
df9
```

Out[32]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp
0	Lesotho	1983	23.91304348	1483270.0	98	106.8	21	87.4762725830078
1	Gambia, The	1988	31.22936246	841250.0	140.7799988	65	15.25	113.245697021484
2	Lesotho	1986	21.14252061	1603960.0	92	109	23.4	66.7168197631836
3	Malawi	1976	39.20110669	5409980.0	179.8	56	4	14.338809967041
4	Botswana	1994	5.199306759	1420270.0	55.39999898	117	53	58.7042083740234
...	...	...	...	...	...	...	...	...
70	Burkina Faso	1988	48.94457166	8534390.0	107.8	33.5	6.5	35.4728813171387
71	Botswana	1988	7.060807251	1195140.0	56.6	111.75	35.75	135.776702880859
72	Kenya	1970	33.29286623	11498000.0	102	58	9	9.42126178741455
73	Lesotho	1990	19.96355858	1783000.0	84.6	105	25	67.9856872558594
74	Kenya	1991	28.14106137	24015140.0	61.4	93	28	29.552059173584

75 rows × 50 columns



In [33]:
```
df10 = pd.merge(df1, df2, how='right')
df10
```

Out[33]:

	countryc	year	agrgdp	popn	infmort	schprim
0	Kenya	1980	32.59223808	16560000.0	72.40000153	115
1	Ghana	1973	48.97463727	9388140.0	106	68.2
2	Liberia	1990	1.79769313486232e+308	2435000.0	176.8	1.79769313486232e+308
3	Madagascar	1981	33.07584521	8951460.0	134	1.79769313486232e+308
4	Ethiopia	1989	48.50468489	49337260.0	126.8	34
5	Lesotho	1988	24.40058125	1689570.0	88.2	107
6	Kenya	1984	33.91489131	19302140.0	64.8	102.2
7	Ethiopia	1987	49.64547916	46087060.0	132	34.5
8	Mauritius	1984	14.406639	1011330.0	26.4	106.6
9	Lesotho	1980	23.58414239	1367000.0	108.4000015	102
10	Liberia	1974	32.28125548	1560820.0	175.4	60.8
11	Ghana	1990	47.85769483	14870000.0	82.74000092	77
12	Ethiopia	1974	56.27994714	32098120.0	152.6	22.4
13	Burkina Faso	1979	34.21393892	6797540.0	123	18
14	Liberia	1984	35.45335614	2140800.0	148.6	48
15	Madagascar	1986	36.78584035	10287560.0	116.7200012	1.79769313486232e+308
16	Liberia	1971	24.425	1426840.0	179.4000015	57.2
17	Burkina Faso	1974	36.48014145	6075700.0	133	15.4
18	Cameroon	1973	30.79282681	7021850.0	115.6	93.8
19	Mauritius	1983	13.8038255	1003930.0	27.2	103.2
20	Madagascar	1991	32.98051479	12054150.0	97.08000031	79
21	Liberia	1973	29.92056487	1514530.0	178.2	59.6
22	Lesotho	1979	30.48723898	1328910.0	112.600001	104
23	Lesotho	1995	10.08687856	1980000.0	75.59999847	1.79769313486232e+308
24	Gambia, The	1979	31.23783032	622460.0	161.799998	42
25	Ethiopia	1973	56.27994714	31273540.0	153.8	20.8
26	Mauritius	1973	19.9317554	859470.0	51.6	101.8
27	Botswana	1990	5.456680968	4370235.0	55.8	114
28	Burkina Faso	1985	37.93019376	7881000.0	112.2	29
29	Kenya	1971	31.37739887	11903370.0	100	65.4
30	Liberia	1995	1.79769313486232e+308	2733000.0	171.8000031	1.79769313486232e+308
31	Kenya	1986	33.04255174	20683850.0	63.6	98.2
32	Gambia, The	1977	33.64667747	585370.0	167	33

	countryc	year	agrgdp	popn	infmort	schprim
33	Cameroon	1988	23.9429277	10835870.0	71.4	102.75
34	Liberia	1985	36.52958877	2199000.0	146.4	48
35	Burkina Faso	1993	34.90956072	9804460.0	101.8000005	38
36	Burkina Faso	1989	31.72235372	8770560.0	106.6	35
37	Liberia	1981	31.56142828	1941970.0	155.8000031	48
38	Burkina Faso	1978	36.08594394	6639120.0	125	17
39	Ghana	1989	48.96733723	14425360.0	85.16000061	74
40	Lesotho	1991	12.096718	1820770.0	82.8	105
41	Mauritius	1981	14.34112949	981170.0	30	96.4
42	Liberia	1975	26.59668835	1609000.0	172.6	62
43	Gambia, The	1978	30.6446491	603960.0	164.399999	37
44	Ghana	1984	49.24249984	12167740.0	94.8	76.8
45	Ethiopia	1984	48.82879875	42152320.0	148.2	34.8
46	Botswana	1984	7.441225106	1037290.0	61.2	102.2
47	Mauritius	1990	12.10414774	1057000.0	20.4	109
48	Kenya	1990	29.51903784	23354000.0	61.8	95
49	Mauritius	1977	19.66080402	913710.0	38	109
50	Ghana	1995	46.27585233	17075000.0	72.73999786	80
51	Madagascar	1974	34.22234966	7408570.0	163.2	94
52	Madagascar	1990	32.30721538	11672000.0	101.1600006	87
53	Ethiopia	1988	49.15748278	47643232.0	129.4	34.25
54	Burkina Faso	1991	34.66403162	9269910.0	104.2	37
55	Liberia	1977	31.87008374	1708160.0	167	60.66666667

56 rows × 50 columns

🏆 Challenge

Generate a new data frame which contains information only from those African countries with the same 'Agriculture as a Share of GDP'.

In [34]:
```
df11 = pd.merge(df1, df2, on='agrgdp')
df11
```

Out[34]:

	countryc_x	year_x	agrgdp	popn_x	infmort_x	schprim_x
0	Ethiopia	1979	56.27994714	36696848.0	153	37
1	Ethiopia	1979	56.27994714	36696848.0	153	37
2	Ethiopia	1980	56.27994714	37717000.0	155	34
3	Ethiopia	1980	56.27994714	37717000.0	155	34
4	Liberia	1991	1.79769313486232e+308	2483450.0	188.4	1.79769313486232e+308
5	Liberia	1991	1.79769313486232e+308	2483450.0	188.4	1.79769313486232e+308
6	Ethiopia	1971	1.79769313486232e+308	29698260.0	156.4000015	17.6
7	Ethiopia	1971	1.79769313486232e+308	29698260.0	156.4000015	17.6

8 rows × 99 columns

Sorting

Sort values by a single column

In [36]:
```
#sort column=wise based on agrgdp in ascending order
df1.sort_values(by=['agrgdp'], ascending=True)
```

Out[36]:

	countryc	year	agrgdp	popn	infmort	schprim
81	Ethiopia	1971	1.79769313486232e+308	29698260.0	156.4000015	17.6
205	Liberia	1991	1.79769313486232e+308	2483450.0	188.4	1.79769313486232e+308
285	Mauritius	1992	10.82725922	1081000.0	18	107
282	Mauritius	1989	12.32418189	1048560.0	21.6	109.2
234	Lesotho	1994	13.69297806	1938930.0	77.39999898	1.79769313486232e+308
...	...	...	...	...	...	...
90	Ethiopia	1980	56.27994714	37717000.0	155	34
118	Ghana	1982	57.34115279	11366410.0	98	78.4
101	Ethiopia	1991	59.13332578	52954000.0	121.6	25
46	Botswana	1988	7.060807251	1195140.0	56.6	111.75
286	Mauritius	1993	9.715403179	1097000.0	17.35999997	106

75 rows × 50 columns

Sort values by row labels

In [37]:
```
#sorts row wise
df1.sort_index(axis=0, ascending=True)
```

Out[37]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp
5	Burkina Faso	1973	34.83428571	5958700.0	135	14.8	1.6	25.9791507720947
7	Burkina Faso	1975	34.27100776	6202000.0	131	16	2	30.4486293792725
8	Burkina Faso	1976	34.80431988	804215.0	129	15	2	24.3181304931641
12	Burkina Faso	1980	33.24267254	6962000.0	121	18	3	41.1754417419434
15	Burkina Faso	1983	31.9042673	7490710.0	115.4	24.6	4.2	27.2162609100342
...	...	...	...	...	...	...	...	...
285	Mauritius	1992	10.82725922	1081000.0	18	107	57	32.6669311523438
286	Mauritius	1993	9.715403179	1097000.0	17.35999997	106	59	31.8793106079102
294	Malawi	1974	41.17239788	5087140.0	185.4	51.6	4	8.73302841186524
295	Malawi	1975	37.23468769	5244000.0	182.6	53.8	4	10.5895004272461
296	Malawi	1976	39.20110669	5409980.0	179.8	56	4	14.338809967041

75 rows × 50 columns

🏆 Challenge

Sort the values in df3 according to the least 'Secondary School Enrolment Rate'. Select the best method for sorting to solve this challenge.

In [40]:
```
df3.sort_values(by=['schsec'], ascending=True)
```

Out[40]:

	countryc	year	agrgdp	popn	infmort	
4	Burkina Faso	1973	34.83428571	5.958700e+06	135	
244	Madagascar	1978	32.22944414	8.251580e+06	146	
289	Mauritius	1996	9.393333333	1.257054e+07	1.79769313486232e+308	1.7976931
207	Liberia	1993	1.79769313486232e+308	2.596430e+06	190.600001	1.7976931
203	Liberia	1989	1.79769313486232e+308	2.391540e+06	165.2	1.7976931
160	Kenya	1972	35.19454123	1.232986e+07	98	
213	Lesotho	1973	42.52275683	1.131220e+06	128.2	
138	Gambia, The	1976	37.72430669	5.666900e+05	169.4	
102	Ethiopia	1992	64.4192191	5.479000e+07	119	
161	Kenya	1973	35.46295124	1.277806e+07	94.8	
104	Ethiopia	1994	56.9741607	5.489000e+07	114.200002	
237	Madagascar	1971	24.30771479	6.901230e+06	176.5999985	
100	Ethiopia	1990	49.26866451	5.118000e+07	124.2	
164	Kenya	1976	37.90313747	1.425500e+07	85.2	
153	Gambia, The	1991	28.27984753	9.646900e+05	134.1199951	
61	Cameroon	1977	33.64508393	7.920570e+06	102	
33	Botswana	1975	31.60036166	7.551000e+05	80.8	
260	Madagascar	1994	38.99940568	1.324875e+07	90.36000061	
157	Gambia, The	1995	1.79769313486232e+308	1.113000e+06	125.9599991	
9	Burkina Faso	1977	34.31152713	6.486870e+06	127	
171	Kenya	1983	34.21715536	1.859766e+07	65.4	
40	Botswana	1982	12.55695077	9.669400e+05	64	
193	Liberia	1979	34.27732326	1.819140e+06	161.4000041	
255	Madagascar	1989	32.93639906	1.130174e+07	105.2400009	
71	Cameroon	1987	23.9858232	1.053526e+07	74	
181	Kenya	1993	31.22035451	2.534740e+07	60	
180	Kenya	1992	28.50164438	2.467985e+07	61	
43	Botswana	1985	6.496773488	1.075000e+06	59.8	
14	Burkina Faso	1982	32.11751268	7.308230e+06	117	
75	Cameroon	1991	24.25394355	1.182539e+07	63.6	
110	Ghana	1974	51.13838759	9.621420e+06	105	
299	Malawi	1979	39.64158617	5.947940e+06	171.4000041	

	countryc	year	agrgdp	popn	infmort
16	Burkina Faso	1984	32.90579486	7.681280e+06	113.8
279	Mauritius	1986	15.25835866	1.024540e+06	24.8
275	Mauritius	1982	15.26946108	9.938500e+05	28
273	Mauritius	1980	12.3697388	9.660000e+05	32
281	Mauritius	1988	13.05016417	1.040330e+06	22.8
50	Botswana	1992	5.088348271	1.353060e+06	55
87	Ethiopia	1977	56.27994714	3.475976e+07	149
54	Cameroon	1970	31.36392206	6.506000e+06	125.8000031
91	Ethiopia	1981	56.27994714	3.877237e+07	157
88	Ethiopia	1978	56.27994714	3.475976e+07	151

42 rows × 50 columns

Selection

Select columns by their names

In [42]:
```
df1[['countryc', 'year']]
```

Out[42]:

	countryc	year
223	Lesotho	1983
150	Gambia, The	1988
226	Lesotho	1986
296	Malawi	1976
52	Botswana	1994
...	...	...
20	Burkina Faso	1988
46	Botswana	1988
158	Kenya	1970
230	Lesotho	1990
179	Kenya	1991

75 rows × 2 columns

Select columns by index

In [44]:
```
df1[df1.columns[1:8]].head()
```

Out[44]:

	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp
223	1983	23.91304348	1483270.0	98	106.8	21	87.4762725830078
150	1988	31.22936246	841250.0	140.7799988	65	15.25	113.245697021484
226	1986	21.14252061	1603960.0	92	109	23.4	66.7168197631836
296	1976	39.20110669	5409980.0	179.8	56	4	14.338809967041
52	1994	5.199306759	1420270.0	55.39999898	117	53	58.7042083740234

Slicing

In [45]:
```
#all rows till specified index
df.iloc[:3]
```

Out[45]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp	
0	Burkina Faso	1970	35.44188862	5633000.0	141.3999939	13	1	13.3182802200317	1
1	Burkina Faso	1970	35.44188862	5633000.0	141.3999939	13	1	13.3182802200317	1
2	Burkina Faso	1971	36.16739069	5740700.0	139.1999969	13.6	1.2	16.7043991088867	0.6

3 rows × 50 columns

In [46]:
```
#all rows including specified row
df.loc[:3]
```

Out[46]:

	countryc	year	agrgdp	popn	infmort	schprim	schsec	grtdsbp	
0	Burkina Faso	1970	35.44188862	5633000.0	141.3999939	13	1	13.3182802200317	1
1	Burkina Faso	1970	35.44188862	5633000.0	141.3999939	13	1	13.3182802200317	1
2	Burkina Faso	1971	36.16739069	5740700.0	139.1999969	13.6	1.2	16.7043991088867	0.6
3	Burkina Faso	1972	37.51058767	5848380.0	137	14.2	1.4	20.9176502227783	2

4 rows × 50 columns

Challenge

Select only those values in df4 that relate to the concessionary loans to various sectors in Africa.

In [47]:
```
df4[df4.columns[23:29]].head()
```

Out[47]:

	cnlnenp	cnlninp	cnlntacp	cnlnedup	cl
0	0	0.320718288421631	0.00690389983355999	0	
1	0	0.320718288421631	0.00690389983355999	0	
2	0	0.317928194999695	0.00292750005610287	0	
3	0	0.185248598456383	0.567323684692383	0	
13	0.703820884227753	2.31013488769531	0.713891804218292	0.320209890604019	0.201189801

Grouping

Group by multiple columns

In [49]:
```
df1.groupby(['countryc', 'year']).groups
```

Out[49]:
```
{('Botswana', 1976): [34], ('Botswana', 1987): [45], ('Botswana', 1988): [4
6], ('Botswana', 1994): [52], ('Burkina Faso', 1973): [5], ('Burkina Faso', 1
975): [7], ('Burkina Faso', 1976): [8], ('Burkina Faso', 1980): [12], ('Burki
na Faso', 1983): [15], ('Burkina Faso', 1988): [20], ('Burkina Faso', 1990):
[22], ('Burkina Faso', 1994): [26], ('Cameroon', 1971): [55], ('Cameroon', 19
75): [59], ('Cameroon', 1979): [63], ('Cameroon', 1980): [64], ('Cameroon', 1
982): [66], ('Cameroon', 1989): [73], ('Cameroon', 1990): [74], ('Ethiopia',
1971): [81], ('Ethiopia', 1979): [89], ('Ethiopia', 1980): [90], ('Ethiopia',
1982): [92], ('Ethiopia', 1991): [101], ('Gambia, The', 1971): [133], ('Gambi
a, The', 1974): [136], ('Gambia, The', 1975): [137], ('Gambia, The', 1982):
[144], ('Gambia, The', 1988): [150], ('Gambia, The', 1990): [152], ('Ghana',
1970): [106], ('Ghana', 1972): [108], ('Ghana', 1975): [111], ('Ghana', 198
2): [118], ('Ghana', 1986): [122], ('Ghana', 1993): [129], ('Kenya', 1970):
[158], ('Kenya', 1978): [166], ('Kenya', 1985): [173], ('Kenya', 1987): [17
5], ('Kenya', 1988): [176], ('Kenya', 1991): [179], ('Kenya', 1994): [182],
('Lesotho', 1972): [212], ('Lesotho', 1974): [214], ('Lesotho', 1975): [215],
('Lesotho', 1976): [216], ('Lesotho', 1981): [221], ('Lesotho', 1983): [223],
('Lesotho', 1984): [224], ('Lesotho', 1985): [225], ('Lesotho', 1986): [226],
('Lesotho', 1987): [227], ('Lesotho', 1990): [230], ('Lesotho', 1994): [234],
('Liberia', 1970): [184], ('Liberia', 1976): [190], ('Liberia', 1987): [201],
('Liberia', 1991): [205], ('Madagascar', 1972): [238], ('Madagascar', 1973):
[239], ('Madagascar', 1975): [241], ('Madagascar', 1980): [246], ('Madagasca
r', 1984): [250], ('Madagascar', 1987): [253], ('Madagascar', 1988): [254],
('Madagascar', 1995): [261], ('Malawi', 1974): [294], ('Malawi', 1975): [29
5], ('Malawi', 1976): [296], ('Mauritius', 1970): [263], ('Mauritius', 1978):
[271], ('Mauritius', 1989): [282], ('Mauritius', 1992): [285], ('Mauritius',
1993): [286]}
```

Calculate the aggregate of a group

In [50]:
```
df1.groupby(['countryc', 'year']).agg(np.mean)
```

Out[50]:

		popn
countryc	**year**	
Botswana	**1976**	782650.0
	1987	1154280.0
	1988	1195140.0
	1994	1420270.0
Burkina Faso	**1973**	5958700.0
...	**...**	...
Mauritius	**1970**	829000.0
	1978	930800.0
	1989	1048560.0
	1992	1081000.0
	1993	1097000.0

75 rows × 1 columns

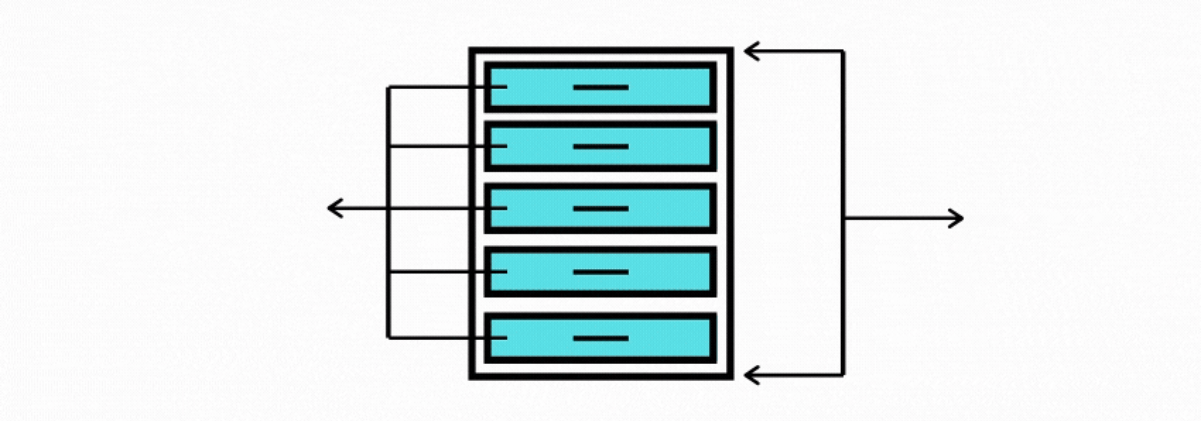
🏆 Challenge

Group the values in df1 by their 'GDP per Capita in 1995'.

In [52]:
```
df1.groupby(['gdnpp']).groups
```

Out[52]:
```
{'1.79769313486232e+308': [89, 205, 81, 90], '1019.97100830078': [190], '102
8.31005859375': [34], '1034.93701171875': [74], '1103.14099121094': [73], '11
50.58203125': [66], '1268.93200683594': [64], '1342.97399902344': [63], '157
9.29895019531': [45], '175.921997070312': [26], '178.331405639648': [225], '1
85.867706298828': [226], '187.163803100586': [101], '193.556701660156': [92],
'2027.98706054688': [271], '2057.32299804688': [46], '2109.35009765625': [28
2], '217.871795654297': [261], '223.952697753906': [212], '231.298294067383':
[224], '231.496795654297': [227], '244.079895019531': [182], '247.67579650878
9': [254], '260.397705078125': [216], '2606.72290039062': [286], '2715.393066
40625': [285], '272.242614746094': [15], '273.797698974609': [253], '274.7084
9609375': [223], '279.595092773438': [295], '283.199096679688': [296], '283.6
29302978516': [294], '284.081390380859': [215], '2864.78198242188': [52], '29
1.868499755859': [5], '299.073699951172': [214], '310.381988525391': [8], '31
5.062408447266': [230], '324.816589355469': [179], '326.678314208984': [7],
'333.131103515625': [150], '335.458892822266': [20], '340.298400878906': [15
2], '342.223693847656': [22], '364.690002441406': [133], '369.052612304688':
[173], '374.082489013672': [221], '381.750610351562': [250], '386.43890380859
4': [234], '387.052307128906': [129], '414.885406494141': [176], '416.9397888
18359': [175], '421.497497558594': [144], '423.212310791016': [12], '446.7815
85693359': [158], '472.230499267578': [137], '526.7744140625': [118], '545.22
4914550781': [238], '560.515014648438': [136], '572.989624023438': [122], '59
8.976013183594': [201], '618.572570800781': [55], '622.341369628906': [239],
'635.165405273438': [166], '656.596923828125': [111], '669.153015136719': [10
8], '685.859313964844': [241], '735.466918945312': [246], '783.319274902344':
[263], '784.416015625': [106], '947.454711914062': [59], '965.92626953125':
[184]}
```

Binning Data



In [53]:
```
#each individual box is a bin (something like dividing up our data)
df.shape
#301 rows, 50 cols
```

Out[53]:
```
(301, 50)
```

In [56]:
```
#binning our data (7 bins)
pd.qcut(df['popn'], q=7).value_counts()
```

Out[56]:
```
(16616600.0, 56404000.0]       43
(2652272.857, 7886652.857]     43
(463999.999, 953097.143]       43
(10664697.143, 16616600.0]     42
(7886652.857, 10664697.143]    42
(1377285.714, 2652272.857]     42
(953097.143, 1377285.714]      42
Name: popn, dtype: int64
```

🏆 Challenge

Bin the data in df2 according to the expenditure on various sectors.

In [57]:
```
df2['year'].shape
```

Out[57]:
```
(56,)
```

In [58]:
```
pd.qcut(df2['year'], q=7).value_counts()
```

Out[58]:
```
(1970.999, 1974.0]      11
(1988.286, 1990.0]       9
(1984.429, 1988.286]     8
(1978.0, 1981.0]         8
(1990.0, 1995.0]         7
(1981.0, 1984.429]       7
(1974.0, 1978.0]         6
Name: year, dtype: int64
```