# Experiment 5 : Perform the following operation on dataframes using Pandas

# a)Basic operations on dataframes using pandas

In [1]:
```
import pandas as pd
df = pd.DataFrame({"Name":['Harika','Jasmine','Kavya','Likitha','Maya','Navya','Sari
                   "rollno":[5401,5402,5403,5404,5405,5406,5407],
                   "age":[18,18,19,20,19,20,21],
                   "marks":[84,89,90,77,87,99,89]},
                   index=['A','B','C','D','E','G', 'H'])
df
```

Out[1]:

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **A** | Harika | 5401 | 18 | 84 |
| **B** | Jasmine | 5402 | 18 | 89 |
| **C** | Kavya | 5403 | 19 | 90 |
| **D** | Likitha | 5404 | 20 | 77 |
| **E** | Maya | 5405 | 19 | 87 |
| **G** | Navya | 5406 | 20 | 99 |
| **H** | Sarika | 5407 | 21 | 89 |

In [2]:
```
##getting the starting data
df.head()
```

Out[2]:

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **A** | Harika | 5401 | 18 | 84 |
| **B** | Jasmine | 5402 | 18 | 89 |
| **C** | Kavya | 5403 | 19 | 90 |
| **D** | Likitha | 5404 | 20 | 77 |
| **E** | Maya | 5405 | 19 | 87 |

In [3]:
```
## getting the last data
df.tail()
```

Out[3]:

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **C** | Kavya | 5403 | 19 | 90 |
| **D** | Likitha | 5404 | 20 | 77 |
| **E** | Maya | 5405 | 19 | 87 |
| **G** | Navya | 5406 | 20 | 99 |

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **H** | Sarika | 5407 | 21 | 89 |

In [4]:
```
## accessing the elemetns
df.loc['C']
```

Out[4]:
```
Name      Kavya
rollno     5403
age          19
marks        90
Name: C, dtype: object
```

In [5]:
```
df.loc['C':'G']
```

Out[5]:

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **C** | Kavya | 5403 | 19 | 90 |
| **D** | Likitha | 5404 | 20 | 77 |
| **E** | Maya | 5405 | 19 | 87 |
| **G** | Navya | 5406 | 20 | 99 |

In [7]:
```
df.loc[['D','H']]
```

Out[7]:

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **D** | Likitha | 5404 | 20 | 77 |
| **H** | Sarika | 5407 | 21 | 89 |

In [8]:
```
##accessing elements by numerical index
df.iloc[1]
```

Out[8]:
```
Name      Jasmine
rollno       5402
age            18
marks          89
Name: B, dtype: object
```

In [10]:
```
df.iloc[[1,3]]
```

Out[10]:

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **B** | Jasmine | 5402 | 18 | 89 |
| **D** | Likitha | 5404 | 20 | 77 |

In [11]:
```
df.iloc[1:3]
```

Out[11]:

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **B** | Jasmine | 5402 | 18 | 89 |

| | Name | rollno | age | marks |
|---|---|---|---|---|
| **C** | Kavya | 5403 | 19 | 90 |

In [14]:
```
df.iloc[[2,3],[2,3]]
```

Out[14]:

| | age | marks |
|---|---|---|
| **C** | 19 | 90 |
| **D** | 20 | 77 |

In [28]:
```
df.iloc[[2],[1,3]]
```

Out[28]:

| | rollno | marks |
|---|---|---|
| **C** | 5403 | 90 |

In [16]:
```
##adding columns

df = pd.DataFrame(df,columns=['Name','rollno','age','marks','branch'])
df
```

Out[16]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **A** | Harika | 5401 | 18 | 84 | NaN |
| **B** | Jasmine | 5402 | 18 | 89 | NaN |
| **C** | Kavya | 5403 | 19 | 90 | NaN |
| **D** | Likitha | 5404 | 20 | 77 | NaN |
| **E** | Maya | 5405 | 19 | 87 | NaN |
| **G** | Navya | 5406 | 20 | 99 | NaN |
| **H** | Sarika | 5407 | 21 | 89 | NaN |

In [17]:
```
df['branch'] = ['AI','ML','CIVIL','MECH','IT','ECE','CSE']
```

In [18]:
```
df
```

Out[18]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **A** | Harika | 5401 | 18 | 84 | AI |
| **B** | Jasmine | 5402 | 18 | 89 | ML |
| **C** | Kavya | 5403 | 19 | 90 | CIVIL |
| **D** | Likitha | 5404 | 20 | 77 | MECH |
| **E** | Maya | 5405 | 19 | 87 | IT |
| **G** | Navya | 5406 | 20 | 99 | ECE |
| **H** | Sarika | 5407 | 21 | 89 | CSE |

In [19]:
```
#adding rows
df.loc[len(df.index)] = ['parvathi',5408,19,85,'AI']
df
```

Out[19]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **A** | Harika | 5401 | 18 | 84 | AI |
| **B** | Jasmine | 5402 | 18 | 89 | ML |
| **C** | Kavya | 5403 | 19 | 90 | CIVIL |
| **D** | Likitha | 5404 | 20 | 77 | MECH |
| **E** | Maya | 5405 | 19 | 87 | IT |
| **G** | Navya | 5406 | 20 | 99 | ECE |
| **H** | Sarika | 5407 | 21 | 89 | CSE |
| **7** | parvathi | 5408 | 19 | 85 | AI |

In [20]:
```
df.rename(index = {7:'I'},inplace =True)
df
```

Out[20]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **A** | Harika | 5401 | 18 | 84 | AI |
| **B** | Jasmine | 5402 | 18 | 89 | ML |
| **C** | Kavya | 5403 | 19 | 90 | CIVIL |
| **D** | Likitha | 5404 | 20 | 77 | MECH |
| **E** | Maya | 5405 | 19 | 87 | IT |
| **G** | Navya | 5406 | 20 | 99 | ECE |
| **H** | Sarika | 5407 | 21 | 89 | CSE |
| **I** | parvathi | 5408 | 19 | 85 | AI |

In [21]:
```
df1=df.reset_index(drop = True)
```

In [22]:
```
df1
```

Out[22]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **0** | Harika | 5401 | 18 | 84 | AI |
| **1** | Jasmine | 5402 | 18 | 89 | ML |
| **2** | Kavya | 5403 | 19 | 90 | CIVIL |
| **3** | Likitha | 5404 | 20 | 77 | MECH |
| **4** | Maya | 5405 | 19 | 87 | IT |
| **5** | Navya | 5406 | 20 | 99 | ECE |
| **6** | Sarika | 5407 | 21 | 89 | CSE |

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **7** | parvathi | 5408 | 19 | 85 | AI |

In [24]:
```
df1.sort_values('Name')
```

Out[24]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **0** | Harika | 5401 | 18 | 84 | AI |
| **1** | Jasmine | 5402 | 18 | 89 | ML |
| **2** | Kavya | 5403 | 19 | 90 | CIVIL |
| **3** | Likitha | 5404 | 20 | 77 | MECH |
| **4** | Maya | 5405 | 19 | 87 | IT |
| **5** | Navya | 5406 | 20 | 99 | ECE |
| **6** | Sarika | 5407 | 21 | 89 | CSE |
| **7** | parvathi | 5408 | 19 | 85 | AI |

In [26]:
```
df1.sort_values('age')
```

Out[26]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **0** | Harika | 5401 | 18 | 84 | AI |
| **1** | Jasmine | 5402 | 18 | 89 | ML |
| **2** | Kavya | 5403 | 19 | 90 | CIVIL |
| **4** | Maya | 5405 | 19 | 87 | IT |
| **7** | parvathi | 5408 | 19 | 85 | AI |
| **3** | Likitha | 5404 | 20 | 77 | MECH |
| **5** | Navya | 5406 | 20 | 99 | ECE |
| **6** | Sarika | 5407 | 21 | 89 | CSE |

In [34]:
```
df1[df1['marks'].between(70,85)]
```

Out[34]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **0** | Harika | 5401 | 18 | 84 | AI |
| **3** | Likitha | 5404 | 20 | 77 | MECH |
| **7** | parvathi | 5408 | 19 | 85 | AI |

In [36]:
```
# select data based on conditions
df1[df1['branch'] == 'AI']
```

Out[36]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **0** | Harika | 5401 | 18 | 84 | AI |

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **7** | parvathi | 5408 | 19 | 85 | AI |

In [39]:
```
df1[(df1['marks'] >= 85) & (df1['age'] > 19)]
```

Out[39]:

| | Name | rollno | age | marks | branch |
|---|---|---|---|---|---|
| **5** | Navya | 5406 | 20 | 99 | ECE |
| **6** | Sarika | 5407 | 21 | 89 | CSE |

# b) Hierarchical Indexing

In [40]:
```
import pandas as pd
df = pd.read_csv("homelessness.csv")
df.head()
```

Out[40]:

| | Unnamed: 0 | region | state | individuals | family_members | state_pop |
|---|---|---|---|---|---|---|
| **0** | 0 | East South Central | Alabama | 2570 | 864 | 4887681 |
| **1** | 1 | Pacific | Alaska | 1434 | 582 | 735139 |
| **2** | 2 | Mountain | Arizona | 7259 | 2606 | 7158024 |
| **3** | 3 | West South Central | Arkansas | 2280 | 432 | 3009733 |
| **4** | 4 | Pacific | California | 109008 | 20964 | 39461588 |

In [41]:
```
col = df.columns
print(col)
```

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

In [61]:
```
df2 = df.set_index(['region','state','individuals'])
```

In [52]:
```
df2.sort_index()
```

Out[52]:

| | | | Unnamed: 0 | family_members | state_pop |
|---|---|---|---|---|---|
| **region** | **state** | **individuals** | | | |
| **East North Central** | **Illinois** | **6752** | 13 | 3891 | 12723071 |
| | **Indiana** | **3776** | 14 | 1482 | 6695497 |
| | **Michigan** | **5209** | 22 | 3142 | 9984072 |
| | **Ohio** | **6929** | 35 | 3320 | 11676341 |
| | **Wisconsin** | **2740** | 49 | 2167 | 5807406 |
| **East South Central** | **Alabama** | **2570** | 0 | 864 | 4887681 |
| | **Kentucky** | **2735** | 17 | 953 | 4461153 |

| region | state | individuals | Unnamed: 0 | family_members | state_pop |
|---|---|---|---|---|---|
| | **Mississippi** | **1024** | 24 | 328 | 2981020 |
| | **Tennessee** | **6139** | 42 | 1744 | 6771631 |
| **Mid-Atlantic** | **New Jersey** | **6048** | 30 | 3350 | 8886025 |
| | **New York** | **39827** | 32 | 52070 | 19530351 |
| | **Pennsylvania** | **8163** | 38 | 5349 | 12800922 |
| **Mountain** | **Arizona** | **7259** | 2 | 2606 | 7158024 |
| | **Colorado** | **7607** | 5 | 3250 | 5691287 |
| | **Idaho** | **1297** | 12 | 715 | 1750536 |
| | **Montana** | **983** | 26 | 422 | 1060665 |
| | **Nevada** | **7058** | 28 | 486 | 3027341 |
| | **New Mexico** | **1949** | 31 | 602 | 2092741 |
| | **Utah** | **1904** | 44 | 972 | 3153550 |
| | **Wyoming** | **434** | 50 | 205 | 577601 |
| **New England** | **Connecticut** | **2280** | 6 | 1696 | 3571520 |
| | **Maine** | **1450** | 19 | 1066 | 1339057 |
| | **Massachusetts** | **6811** | 21 | 13257 | 6882635 |
| | **New Hampshire** | **835** | 29 | 615 | 1353465 |
| | **Rhode Island** | **747** | 39 | 354 | 1058287 |
| | **Vermont** | **780** | 45 | 511 | 624358 |
| **Pacific** | **Alaska** | **1434** | 1 | 582 | 735139 |
| | **California** | **109008** | 4 | 20964 | 39461588 |
| | **Hawaii** | **4131** | 11 | 2399 | 1420593 |
| | **Oregon** | **11139** | 37 | 3337 | 4181886 |
| | **Washington** | **16424** | 47 | 5880 | 7523869 |
| **South Atlantic** | **Delaware** | **708** | 7 | 374 | 965479 |
| | **District of Columbia** | **3770** | 8 | 3134 | 701547 |
| | **Florida** | **21443** | 9 | 9587 | 21244317 |
| | **Georgia** | **6943** | 10 | 2556 | 10511131 |
| | **Maryland** | **4914** | 20 | 2230 | 6035802 |
| | **North Carolina** | **6451** | 33 | 2817 | 10381615 |
| | **South Carolina** | **3082** | 40 | 851 | 5084156 |
| | **Virginia** | **3928** | 46 | 2047 | 8501286 |
| | **West Virginia** | **1021** | 48 | 222 | 1804291 |
| **West North Central** | **Iowa** | **1711** | 15 | 1038 | 3148618 |
| | **Kansas** | **1443** | 16 | 773 | 2911359 |

| | | | Unnamed: 0 | family_members | state_pop |
|---|---|---|---|---|---|
| region | state | individuals | | | |
| | **Minnesota** | **3993** | 23 | 3250 | 5606249 |
| | **Missouri** | **3776** | 25 | 2107 | 6121623 |
| | **Nebraska** | **1745** | 27 | 676 | 1925614 |
| | **North Dakota** | **467** | 34 | 75 | 758080 |
| | **South Dakota** | **836** | 41 | 323 | 878698 |
| **West South Central** | **Arkansas** | **2280** | 3 | 432 | 3009733 |
| | **Louisiana** | **2540** | 18 | 519 | 4659690 |
| | **Oklahoma** | **2823** | 36 | 1048 | 3940235 |
| | **Texas** | **19199** | 43 | 6111 | 28628666 |

In [54]:
```
# selecting data in hierarchical index
df3 = df2.loc[['Pacific','Mountain']]
print(df3.head(10))
```

```
                              Unnamed: 0  family_members  state_pop
region   state      individuals
Pacific  Alaska     1434                1            582     735139
         California 109008              4          20964   39461588
         Hawaii     4131               11           2399    1420593
         Oregon     11139              37           3337    4181886
         Washington 16424              47           5880    7523869
Mountain Arizona    7259                2           2606    7158024
         Colorado   7607                5           3250    5691287
         Idaho      1297               12            715    1750536
         Montana    983                26            422    1060665
         Nevada     7058               28            486    3027341
```

In [56]:
```
# To access inner level indexes we need to access it by tuples
df3_inner = df2.loc[[("Pacific","Alaska",1434),
                     ("Pacific","Hawaii",4131),
                     ("Mountain","Idaho",1297)]]
print(df3_inner)
```

```
                         Unnamed: 0  family_members  state_pop
region   state  individuals
Pacific  Alaska 1434                1            582     735139
         Hawaii 4131               11           2399    1420593
Mountain Idaho  1297               12            715    1750536
```

In [63]:
```
## reset the index
df2.reset_index(["state","individuals"])
df2.head()
```

Out[63]:

| | | | Unnamed: 0 | family_members | state_pop |
|---|---|---|---|---|---|
| region | state | individuals | | | |
| **East South Central** | **Alabama** | **2570** | 0 | 864 | 4887681 |
| **Pacific** | **Alaska** | **1434** | 1 | 582 | 735139 |
| **Mountain** | **Arizona** | **7259** | 2 | 2606 | 7158024 |

| | | Unnamed: 0 | family_members | state_pop |
|---|---|---|---|---|
| region | state | individuals | | |
| West South Central | Arkansas | 2280 | 3 | 432 | 3009733 |
| Pacific | California | 109008 | 4 | 20964 | 39461588 |

# c) Combining and Merging Datasets

In [72]:
```
## merge()
# join()
# concat()
# append()
import pandas as pd
```

In [73]:
```
df1 = pd.DataFrame({'ID':[1,2,3,5,9],
                    'Col_1':[1,2,3,4,5],
                    'Col_2':[6,7,8,9,10],
                    'Col_3':[11,12,13,14,15],
                    'Col_4':['apple','orange','banana','strawberry','raspberry']})

df2 = pd.DataFrame({'ID':[1,1,3,5],
                    'Col_A':[8,9,10,11],
                    'Col_B':[12,13,15,17],
                    'Col_4':['apple','orange','banana','kiwi']})
```

In [74]:
```
df1
```

Out[74]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 6 | 11 | apple |
| 1 | 2 | 2 | 7 | 12 | orange |
| 2 | 3 | 3 | 8 | 13 | banana |
| 3 | 5 | 4 | 9 | 14 | strawberry |
| 4 | 9 | 5 | 10 | 15 | raspberry |

In [75]:
```
df2
```

Out[75]:

| | ID | Col_A | Col_B | Col_4 |
|---|---|---|---|---|
| 0 | 1 | 8 | 12 | apple |
| 1 | 1 | 9 | 13 | orange |
| 2 | 3 | 10 | 15 | banana |
| 3 | 5 | 11 | 17 | kiwi |

In [76]:
```
inner = pd.merge(df1,df2) ##by default it is inner join
inner
```

Out[76]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4 | Col_A | Col_B |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 6 | 11 | apple | 8 | 12 |
| **1** | 3 | 3 | 8 | 13 | banana | 10 | 15 |

In [77]:
```
#it is always good to specify which columns to merge on
pd.merge(df1,df2,on = 'ID')
```

Out[77]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4_x | Col_A | Col_B | Col_4_y |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 6 | 11 | apple | 8 | 12 | apple |
| **1** | 1 | 1 | 6 | 11 | apple | 9 | 13 | orange |
| **2** | 3 | 3 | 8 | 13 | banana | 10 | 15 | banana |
| **3** | 5 | 4 | 9 | 14 | strawberry | 11 | 17 | kiwi |

In [78]:
```
#it is always good to specify which columns to merge on
pd.merge(df1,df2,on = 'ID')
```

Out[78]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4_x | Col_A | Col_B | Col_4_y |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 6 | 11 | apple | 8 | 12 | apple |
| **1** | 1 | 1 | 6 | 11 | apple | 9 | 13 | orange |
| **2** | 3 | 3 | 8 | 13 | banana | 10 | 15 | banana |
| **3** | 5 | 4 | 9 | 14 | strawberry | 11 | 17 | kiwi |

In [79]:
```
#outer join
pd.merge(df1,df2,on = 'Col_4',how = 'outer',suffixes=['_l','_r'])
```

Out[79]:

| | ID_l | Col_1 | Col_2 | Col_3 | Col_4 | ID_r | Col_A | Col_B |
|---|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 1.0 | 6.0 | 11.0 | apple | 1.0 | 8.0 | 12.0 |
| **1** | 2.0 | 2.0 | 7.0 | 12.0 | orange | 1.0 | 9.0 | 13.0 |
| **2** | 3.0 | 3.0 | 8.0 | 13.0 | banana | 3.0 | 10.0 | 15.0 |
| **3** | 5.0 | 4.0 | 9.0 | 14.0 | strawberry | NaN | NaN | NaN |
| **4** | 9.0 | 5.0 | 10.0 | 15.0 | raspberry | NaN | NaN | NaN |
| **5** | NaN | NaN | NaN | NaN | kiwi | 5.0 | 11.0 | 17.0 |

In [80]:
```
#left join
pd.merge(df1,df2,on = 'Col_4',how = 'left',suffixes=['_l','_r'])
```

Out[80]:

| | ID_l | Col_1 | Col_2 | Col_3 | Col_4 | ID_r | Col_A | Col_B |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 6 | 11 | apple | 1.0 | 8.0 | 12.0 |
| **1** | 2 | 2 | 7 | 12 | orange | 1.0 | 9.0 | 13.0 |
| **2** | 3 | 3 | 8 | 13 | banana | 3.0 | 10.0 | 15.0 |

| | ID_l | Col_1 | Col_2 | Col_3 | Col_4 | ID_r | Col_A | Col_B |
|---|---|---|---|---|---|---|---|---|
| **3** | 5 | 4 | 9 | 14 | strawberry | NaN | NaN | NaN |
| **4** | 9 | 5 | 10 | 15 | raspberry | NaN | NaN | NaN |

In [81]:
```
#right join
pd.merge(df1,df2,on = 'Col_4',how = 'right',suffixes=['_l','_r'])
```

Out[81]:

| | ID_l | Col_1 | Col_2 | Col_3 | Col_4 | ID_r | Col_A | Col_B |
|---|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 1.0 | 6.0 | 11.0 | apple | 1 | 8 | 12 |
| **1** | 2.0 | 2.0 | 7.0 | 12.0 | orange | 1 | 9 | 13 |
| **2** | 3.0 | 3.0 | 8.0 | 13.0 | banana | 3 | 10 | 15 |
| **3** | NaN | NaN | NaN | NaN | kiwi | 5 | 11 | 17 |

In [82]:
```
#default join is left
df1.join(df2,on='ID',lsuffix = '_l',rsuffix = '_r')
```

Out[82]:

| | ID_l | Col_1 | Col_2 | Col_3 | Col_4_l | ID_r | Col_A | Col_B | Col_4_r |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 6 | 11 | apple | 1.0 | 9.0 | 13.0 | orange |
| **1** | 2 | 2 | 7 | 12 | orange | 3.0 | 10.0 | 15.0 | banana |
| **2** | 3 | 3 | 8 | 13 | banana | 5.0 | 11.0 | 17.0 | kiwi |
| **3** | 5 | 4 | 9 | 14 | strawberry | NaN | NaN | NaN | NaN |
| **4** | 9 | 5 | 10 | 15 | raspberry | NaN | NaN | NaN | NaN |

In [83]:
```
df1.join(df2,on='ID',lsuffix = '_l',rsuffix = '_r',how = 'inner')
```

Out[83]:

| | ID | ID_l | Col_1 | Col_2 | Col_3 | Col_4_l | ID_r | Col_A | Col_B | Col_4_r |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | 6 | 11 | apple | 1 | 9 | 13 | orange |
| **1** | 2 | 2 | 2 | 7 | 12 | orange | 3 | 10 | 15 | banana |
| **2** | 3 | 3 | 3 | 8 | 13 | banana | 5 | 11 | 17 | kiwi |

In [84]:
```
#default axis in 0(this stacks dataframes)
pd.concat([df1,df2])
```

Out[84]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4 | Col_A | Col_B |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 1.0 | 6.0 | 11.0 | apple | NaN | NaN |
| **1** | 2 | 2.0 | 7.0 | 12.0 | orange | NaN | NaN |
| **2** | 3 | 3.0 | 8.0 | 13.0 | banana | NaN | NaN |
| **3** | 5 | 4.0 | 9.0 | 14.0 | strawberry | NaN | NaN |
| **4** | 9 | 5.0 | 10.0 | 15.0 | raspberry | NaN | NaN |
| **0** | 1 | NaN | NaN | NaN | apple | 8.0 | 12.0 |

| | ID | Col_1 | Col_2 | Col_3 | Col_4 | Col_A | Col_B |
|---|---|---|---|---|---|---|---|
| **1** | 1 | NaN | NaN | NaN | orange | 9.0 | 13.0 |
| **2** | 3 | NaN | NaN | NaN | banana | 10.0 | 15.0 |
| **3** | 5 | NaN | NaN | NaN | kiwi | 11.0 | 17.0 |

In [85]:
```
#reset index
pd.concat([df1,df2],ignore_index =True)
```

Out[85]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4 | Col_A | Col_B |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 1.0 | 6.0 | 11.0 | apple | NaN | NaN |
| **1** | 2 | 2.0 | 7.0 | 12.0 | orange | NaN | NaN |
| **2** | 3 | 3.0 | 8.0 | 13.0 | banana | NaN | NaN |
| **3** | 5 | 4.0 | 9.0 | 14.0 | strawberry | NaN | NaN |
| **4** | 9 | 5.0 | 10.0 | 15.0 | raspberry | NaN | NaN |
| **5** | 1 | NaN | NaN | NaN | apple | 8.0 | 12.0 |
| **6** | 1 | NaN | NaN | NaN | orange | 9.0 | 13.0 |
| **7** | 3 | NaN | NaN | NaN | banana | 10.0 | 15.0 |
| **8** | 5 | NaN | NaN | NaN | kiwi | 11.0 | 17.0 |

In [86]:
```
#can be done side-by-side by specifying axis=1
pd.concat([df1,df2],axis =1)
```

Out[86]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4 | ID | Col_A | Col_B | Col_4 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 6 | 11 | apple | 1.0 | 8.0 | 12.0 | apple |
| **1** | 2 | 2 | 7 | 12 | orange | 1.0 | 9.0 | 13.0 | orange |
| **2** | 3 | 3 | 8 | 13 | banana | 3.0 | 10.0 | 15.0 | banana |
| **3** | 5 | 4 | 9 | 14 | strawberry | 5.0 | 11.0 | 17.0 | kiwi |
| **4** | 9 | 5 | 10 | 15 | raspberry | NaN | NaN | NaN | NaN |

In [87]:
```
pd.concat([df1,df2],axis = 0,join = 'inner')
```

Out[87]:

| | ID | Col_4 |
|---|---|---|
| **0** | 1 | apple |
| **1** | 2 | orange |
| **2** | 3 | banana |
| **3** | 5 | strawberry |
| **4** | 9 | raspberry |
| **0** | 1 | apple |
| **1** | 1 | orange |

| | ID | Col_4 |
|---|---|---|
| **2** | 3 | banana |
| **3** | 5 | kiwi |

In [88]:
```
df1.append(df2)
```

Out[88]:

| | ID | Col_1 | Col_2 | Col_3 | Col_4 | Col_A | Col_B |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 1.0 | 6.0 | 11.0 | apple | NaN | NaN |
| **1** | 2 | 2.0 | 7.0 | 12.0 | orange | NaN | NaN |
| **2** | 3 | 3.0 | 8.0 | 13.0 | banana | NaN | NaN |
| **3** | 5 | 4.0 | 9.0 | 14.0 | strawberry | NaN | NaN |
| **4** | 9 | 5.0 | 10.0 | 15.0 | raspberry | NaN | NaN |
| **0** | 1 | NaN | NaN | NaN | apple | 8.0 | 12.0 |
| **1** | 1 | NaN | NaN | NaN | orange | 9.0 | 13.0 |
| **2** | 3 | NaN | NaN | NaN | banana | 10.0 | 15.0 |
| **3** | 5 | NaN | NaN | NaN | kiwi | 11.0 | 17.0 |

In [89]:
```
df1.append(df2,sort=True)
```

Out[89]:

| | Col_1 | Col_2 | Col_3 | Col_4 | Col_A | Col_B | ID |
|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 6.0 | 11.0 | apple | NaN | NaN | 1 |
| **1** | 2.0 | 7.0 | 12.0 | orange | NaN | NaN | 2 |
| **2** | 3.0 | 8.0 | 13.0 | banana | NaN | NaN | 3 |
| **3** | 4.0 | 9.0 | 14.0 | strawberry | NaN | NaN | 5 |
| **4** | 5.0 | 10.0 | 15.0 | raspberry | NaN | NaN | 9 |
| **0** | NaN | NaN | NaN | apple | 8.0 | 12.0 | 1 |
| **1** | NaN | NaN | NaN | orange | 9.0 | 13.0 | 1 |
| **2** | NaN | NaN | NaN | banana | 10.0 | 15.0 | 3 |
| **3** | NaN | NaN | NaN | kiwi | 11.0 | 17.0 | 5 |

# d) Merging on Index

In [65]:
```
employee = [ (11, 'jack', 34, 'Sydney', 5) ,
             (12, 'Riti', 31, 'Delhi' , 7) ,
             (13, 'Aadi', 16, 'New York', 11) ,
             (14, 'Mohit', 32,'Delhi' , 15) ,
             (15, 'Veena', 33, 'Delhi' , 4) ,
             (16, 'Shaunak', 35, 'Mumbai', 5 ),
              (17, 'Shaun', 35, 'Colombo', 11)
              ]
employee = pd.DataFrame(employee,columns=  ['ID','Name','Age','City','Experience'])
employee = employee.set_index('ID')
```

In [66]:
```
print(employee)
```

```
       Name  Age      City  Experience
ID
11     jack   34    Sydney           5
12     Riti   31     Delhi           7
13     Aadi   16  New York          11
14    Mohit   32     Delhi          15
15    Veena   33     Delhi           4
16  Shaunak   35    Mumbai           5
17    Shaun   35   Colombo          11
```

In [68]:
```
salaries = [(11, 'Junior', 70000, 1000) ,
            (12, 'Senior', 72200, 1100) ,
            (13, 'Expert', 84999, 1000) ,
            (14, 'Expert', 90000, 2000) ,
            (15, 'Junior', 61000, 1500) ,
            (16, 'Junior', 71000, 1000),
            (21, 'Senior',81000, 2000)
             ]
salaries = pd.DataFrame(salaries,columns = ['ID','Experience','salary','bonus'],inde
salaries = salaries.set_index('ID')
```

In [69]:
```
print(salaries)
```

```
   Experience  salary  bonus
ID
11     Junior   70000   1000
12     Senior   72200   1100
13     Expert   84999   1000
14     Expert   90000   2000
15     Junior   61000   1500
16     Junior   71000   1000
21     Senior   81000   2000
```

In [70]:
```
# As both the dataframe contains similar IDs on the index. So, to merge the datafram
mergeDF = employee.merge(salaries,left_index = True,right_index = True)
print(mergeDF)
```

```
       Name  Age      City  Experience_x Experience_y  salary  bonus
ID
11     jack   34    Sydney             5       Junior   70000   1000
12     Riti   31     Delhi             7       Senior   72200   1100
13     Aadi   16  New York            11       Expert   84999   1000
14    Mohit   32     Delhi            15       Expert   90000   2000
15    Veena   33     Delhi             4       Junior   61000   1500
16  Shaunak   35    Mumbai             5       Junior   71000   1000
```

In [71]:
```
# Merge two Dataframes on index of one dataframe and some column of other dataframe
salaries['EmpId'] = salaries.index
salaries.reset_index(inplace = True)
del salaries['ID']
```

In [72]:
```
salaries
```

Out[72]:

| | Experience | salary | bonus | EmpId |
|---|---|---|---|---|
| **0** | Junior | 70000 | 1000 | 11 |

| | Experience | salary | bonus | EmpId |
|---|---|---|---|---|
| **1** | Senior | 72200 | 1100 | 12 |
| **2** | Expert | 84999 | 1000 | 13 |
| **3** | Expert | 90000 | 2000 | 14 |
| **4** | Junior | 61000 | 1500 | 15 |
| **5** | Junior | 71000 | 1000 | 16 |
| **6** | Senior | 81000 | 2000 | 21 |

In [75]:
```
mergeDF2 = employee.merge(salaries,left_index = True,right_on='EmpId')
mergeDF2 = mergeDF2.set_index('EmpId')
print(mergeDF2)
```

```
          Name  Age      City  Experience_x Experience_y  salary  bonus
EmpId
11        jack   34    Sydney             5       Junior   70000   1000
12        Riti   31     Delhi             7       Senior   72200   1100
13        Aadi   16  New York            11       Expert   84999   1000
14       Mohit   32     Delhi            15       Expert   90000   2000
15       Veena   33     Delhi             4       Junior   61000   1500
16     Shaunak   35    Mumbai             5       Junior   71000   1000
```

# e) Concatenate, Combining with overlap

In [90]:
```
## concat() is used for combining Data Frames across rows or columns
df1 = pd.DataFrame({'Name':['Alex','Amy','Allen','Alice','Ayoung'],
                    'subject_id':['s1','s2','s4','s6','s5'],
                    'marks_scored':[98,90,87,69,78]},index = [1,2,3,4,5])
print(df1)
```

```
     Name subject_id  marks_scored
1    Alex         s1            98
2     Amy         s2            90
3   Allen         s4            87
4   Alice         s6            69
5  Ayoung         s5            78
```

In [91]:
```
df2 = pd.DataFrame({'Name':['Billy','Brain','Bran','Bryce','Betty'],
                    'subject_id':['s2','s4','s3','s6','s5'],
                    'marks_scored':[89,80,79,97,88]},index = [4,5,6,7,8])
print(df2)
```

```
    Name subject_id  marks_scored
4  Billy         s2            89
5  Brain         s4            80
6   Bran         s3            79
7  Bryce         s6            97
8  Betty         s5            88
```

In [80]:
```
pd.concat([df1,df2])
```

Out[80]:

| | Name | subject_id | marks_scored |
|---|---|---|---|
| **1** | Alex | s1 | 98 |
| **2** | Amy | s2 | 90 |

| | Name | subject_id | marks_scored |
|---|---|---|---|
| **3** | Allen | s4 | 87 |
| **4** | Alice | s6 | 69 |
| **5** | Ayoung | s5 | 78 |
| **4** | Billy | s2 | 89 |
| **5** | Brain | s4 | 80 |
| **6** | Bran | s3 | 79 |
| **7** | Bryce | s6 | 97 |
| **8** | Betty | s5 | 88 |

In [81]:
```
# to ignore indexes from original dataframes
pd.concat([df1,df2],ignore_index=True)
```

Out[81]:

| | Name | subject_id | marks_scored |
|---|---|---|---|
| **0** | Alex | s1 | 98 |
| **1** | Amy | s2 | 90 |
| **2** | Allen | s4 | 87 |
| **3** | Alice | s6 | 69 |
| **4** | Ayoung | s5 | 78 |
| **5** | Billy | s2 | 89 |
| **6** | Brain | s4 | 80 |
| **7** | Bran | s3 | 79 |
| **8** | Bryce | s6 | 97 |
| **9** | Betty | s5 | 88 |

In [82]:
```
# horizontal operation
pd.concat([df1,df2],axis=1)
```

Out[82]:

| | Name | subject_id | marks_scored | Name | subject_id | marks_scored |
|---|---|---|---|---|---|---|
| **1** | Alex | s1 | 98.0 | NaN | NaN | NaN |
| **2** | Amy | s2 | 90.0 | NaN | NaN | NaN |
| **3** | Allen | s4 | 87.0 | NaN | NaN | NaN |
| **4** | Alice | s6 | 69.0 | Billy | s2 | 89.0 |
| **5** | Ayoung | s5 | 78.0 | Brain | s4 | 80.0 |
| **6** | NaN | NaN | NaN | Bran | s3 | 79.0 |
| **7** | NaN | NaN | NaN | Bryce | s6 | 97.0 |
| **8** | NaN | NaN | NaN | Betty | s5 | 88.0 |

In [92]:
```
#concating unequal shape dataframe
```

```
df1_1 = pd.DataFrame({'A':[1,2,3], 'B':[1,2,3]})
df2_2 = pd.DataFrame({'A':[4,5]})
print(df1)
print('* * * * * * ')
print(df2)
print('* * * * * * ')
df = pd.concat([df1,df2],ignore_index=True)
print(df)
```

```
     Name subject_id  marks_scored
1    Alex         s1            98
2     Amy         s2            90
3   Allen         s4            87
4   Alice         s6            69
5  Ayoung         s5            78
* * * * * *
    Name subject_id  marks_scored
4  Billy         s2            89
5  Brain         s4            80
6   Bran         s3            79
7  Bryce         s6            97
8  Betty         s5            88
* * * * * *
     Name subject_id  marks_scored
0    Alex         s1            98
1     Amy         s2            90
2   Allen         s4            87
3   Alice         s6            69
4  Ayoung         s5            78
5   Billy         s2            89
6   Brain         s4            80
7    Bran         s3            79
8   Bryce         s6            97
9   Betty         s5            88
```

In [89]:

```
#using append() function
df1.append(df2)
```

Out[89]:

| | Name | subject_id | marks_scored |
|---|---|---|---|
| **1** | Alex | s1 | 98 |
| **2** | Amy | s2 | 90 |
| **3** | Allen | s4 | 87 |
| **4** | Alice | s6 | 69 |
| **5** | Ayoung | s5 | 78 |
| **4** | Billy | s2 | 89 |
| **5** | Brain | s4 | 80 |
| **6** | Bran | s3 | 79 |
| **7** | Bryce | s6 | 97 |
| **8** | Betty | s5 | 88 |

In [93]:

```
# append different shape dataframes
df1_1.append(df2_2)
```

Out[93]:

| | A | B |
|---|---|---|

| | A | B |
|---|---|---|
| **0** | 1 | 1.0 |
| **1** | 2 | 2.0 |
| **2** | 3 | 3.0 |
| **0** | 4 | NaN |
| **1** | 5 | NaN |

# f)Reshaping

In [94]:
```
df = pd.read_csv("https://media.geeksforgeeks.org/wp-content/uploads/nba.csv")
print(df.head())
```

```
              Name            Team  Number Position   Age Height  Weight  \
0  Avery Bradley  Boston Celtics     0.0       PG  25.0    6-2   180.0
1    Jae Crowder  Boston Celtics    99.0       SF  25.0    6-6   235.0
2   John Holland  Boston Celtics    30.0       SG  27.0    6-5   205.0
3    R.J. Hunter  Boston Celtics    28.0       SG  22.0    6-5   185.0
4  Jonas Jerebko  Boston Celtics     8.0       PF  29.0   6-10   231.0

              College     Salary
0               Texas  7730337.0
1           Marquette  6796117.0
2  Boston University        NaN
3       Georgia State  1148640.0
4                 NaN  5000000.0
```

In [101…
```
df_stack = df.stack()
df_stack.head(20)
```

Out[101…
```
0  Name          Avery Bradley
   Team         Boston Celtics
   Number                  0.0
   Position                 PG
   Age                    25.0
   Height                  6-2
   Weight                180.0
   College               Texas
   Salary            7730337.0
1  Name            Jae Crowder
   Team         Boston Celtics
   Number                 99.0
   Position                 SF
   Age                    25.0
   Height                  6-6
   Weight                235.0
   College           Marquette
   Salary            6796117.0
2  Name           John Holland
   Team         Boston Celtics
dtype: object
```

In [103…
```
df_unstack = df_stack.unstack()
df_unstack.head(20)
```

Out[103…

| | Name | Team | Number | Position | Age | Height | Weight | College | Salary |
|---|---|---|---|---|---|---|---|---|---|

| | Name | Team | Number | Position | Age | Height | Weight | College | Salary |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Avery Bradley | Boston Celtics | 0.0 | PG | 25.0 | 6-2 | 180.0 | Texas | 7730337.0 |
| **1** | Jae Crowder | Boston Celtics | 99.0 | SF | 25.0 | 6-6 | 235.0 | Marquette | 6796117.0 |
| **2** | John Holland | Boston Celtics | 30.0 | SG | 27.0 | 6-5 | 205.0 | Boston University | NaN |
| **3** | R.J. Hunter | Boston Celtics | 28.0 | SG | 22.0 | 6-5 | 185.0 | Georgia State | 1148640.0 |
| **4** | Jonas Jerebko | Boston Celtics | 8.0 | PF | 29.0 | 6-10 | 231.0 | NaN | 5000000.0 |
| **5** | Amir Johnson | Boston Celtics | 90.0 | PF | 29.0 | 6-9 | 240.0 | NaN | 12000000.0 |
| **6** | Jordan Mickey | Boston Celtics | 55.0 | PF | 21.0 | 6-8 | 235.0 | LSU | 1170960.0 |
| **7** | Kelly Olynyk | Boston Celtics | 41.0 | C | 25.0 | 7-0 | 238.0 | Gonzaga | 2165160.0 |
| **8** | Terry Rozier | Boston Celtics | 12.0 | PG | 22.0 | 6-2 | 190.0 | Louisville | 1824360.0 |
| **9** | Marcus Smart | Boston Celtics | 36.0 | PG | 22.0 | 6-4 | 220.0 | Oklahoma State | 3431040.0 |
| **10** | Jared Sullinger | Boston Celtics | 7.0 | C | 24.0 | 6-9 | 260.0 | Ohio State | 2569260.0 |
| **11** | Isaiah Thomas | Boston Celtics | 4.0 | PG | 27.0 | 5-9 | 185.0 | Washington | 6912869.0 |
| **12** | Evan Turner | Boston Celtics | 11.0 | SG | 27.0 | 6-7 | 220.0 | Ohio State | 3425510.0 |
| **13** | James Young | Boston Celtics | 13.0 | SG | 20.0 | 6-6 | 215.0 | Kentucky | 1749840.0 |
| **14** | Tyler Zeller | Boston Celtics | 44.0 | C | 26.0 | 7-0 | 253.0 | North Carolina | 2616975.0 |
| **15** | Bojan Bogdanovic | Brooklyn Nets | 44.0 | SG | 27.0 | 6-8 | 216.0 | NaN | 3425510.0 |
| **16** | Markel Brown | Brooklyn Nets | 22.0 | SG | 24.0 | 6-3 | 190.0 | Oklahoma State | 845059.0 |
| **17** | Wayne Ellington | Brooklyn Nets | 21.0 | SG | 28.0 | 6-4 | 200.0 | North Carolina | 1500000.0 |
| **18** | Rondae Hollis-Jefferson | Brooklyn Nets | 24.0 | SG | 21.0 | 6-7 | 220.0 | Arizona | 1335480.0 |
| **19** | Jarrett Jack | Brooklyn Nets | 2.0 | PG | 32.0 | 6-3 | 200.0 | Georgia Tech | 6300000.0 |

In [104…

```
# melt() reshape dataframe from wide format to long format
df_melt = df.melt(id_vars=['Name','Team'])
df_melt.head(10)
```

Out[104…

| | Name | Team | variable | value |
|---|---|---|---|---|

| | Name | Team | variable | value |
|---|---|---|---|---|
| **0** | Avery Bradley | Boston Celtics | Number | 0.0 |
| **1** | Jae Crowder | Boston Celtics | Number | 99.0 |
| **2** | John Holland | Boston Celtics | Number | 30.0 |
| **3** | R.J. Hunter | Boston Celtics | Number | 28.0 |
| **4** | Jonas Jerebko | Boston Celtics | Number | 8.0 |
| **5** | Amir Johnson | Boston Celtics | Number | 90.0 |
| **6** | Jordan Mickey | Boston Celtics | Number | 55.0 |
| **7** | Kelly Olynyk | Boston Celtics | Number | 41.0 |
| **8** | Terry Rozier | Boston Celtics | Number | 12.0 |
| **9** | Marcus Smart | Boston Celtics | Number | 36.0 |

# g) Pivoting

The pivot() function is used to reshaped a given DataFrame organized by given index / column values. This function does not support data aggregation, multiple values will result in a MultiIndex in the columns.

In [1]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
df = pd.DataFrame({'fff':['one','one','one','two','two','two'],
                   'bbb':['P','Q','R','P','Q','R'],
                   'baa':[2,3,4,5,6,7],
                   'zzz':['h','i','j','k','l','m']})
print(df)
```

```
   fff bbb  baa zzz
0  one   P    2   h
1  one   Q    3   i
2  one   R    4   j
3  two   P    5   k
4  two   Q    6   l
5  two   R    7   m
```

In [4]:
```
df.pivot(index='fff',columns='bbb',values='baa')
```

Out[4]:

| bbb | P | Q | R |
|---|---|---|---|
| **fff** | | | |
| **one** | 2 | 3 | 4 |
| **two** | 5 | 6 | 7 |

In [6]:
```
df.pivot(index='fff',columns='bbb')['baa']
```

Out[6]:

| bbb | P | Q | R |
|---|---|---|---|
| **fff** | | | |
| **one** | 2 | 3 | 4 |

| bbb | P | Q | R |
|---|---|---|---|
| fff | | | |
| two | 5 | 6 | 7 |

In [7]:
```
df.pivot(index='fff',columns='bbb',values=['baa','zzz'])
```

Out[7]:

| | baa | | | zzz | | |
|---|---|---|---|---|---|---|
| bbb | P | Q | R | P | Q | R |
| fff | | | | | | |
| one | 2 | 3 | 4 | h | i | j |
| two | 5 | 6 | 7 | k | l | m |

# h)Vectorized String Operations

In [8]:
```
import numpy as np
```

In [9]:
```
x = np.array([2,3,5,7,11,13])
x * 2
```

Out[9]:
```
array([ 4,  6, 10, 14, 22, 26])
```

In [11]:
```
data = ['peter','Paul','MARY','guiDo']
[s.capitalize() for s in data]
```

Out[11]:
```
['Peter', 'Paul', 'Mary', 'Guido']
```

In [14]:
```
names = pd.Series(data)
names
```

Out[14]:
```
0    peter
1     Paul
2     MARY
3    guiDo
dtype: object
```

In [15]:
```
names.str.capitalize()
```

Out[15]:
```
0    Peter
1     Paul
2     Mary
3    Guido
dtype: object
```

In [16]:
```
names.str.contains("peter")
```

Out[16]:
```
0     True
1    False
2    False
```

```
3    False
dtype: bool
```

In [17]:
```
names.str.lower()
```

Out[17]:
```
0    peter
1     paul
2     mary
3    guido
dtype: object
```

In [19]:
```
names.str.islower()
```

Out[19]:
```
0     True
1    False
2    False
3    False
dtype: bool
```

In [20]:
```
names.str.isalpha()
```

Out[20]:
```
0    True
1    True
2    True
3    True
dtype: bool
```

In [21]:
```
names.str.len()
```

Out[21]:
```
0    5
1    4
2    4
3    5
dtype: int64
```

In [22]:
```
names.str.startswith("p")
```

Out[22]:
```
0     True
1    False
2    False
3    False
dtype: bool
```

In [25]:
```
ser = pd.Series(['A B','C D','E F','G H'])
print(ser)
```

```
0    A B
1    C D
2    E F
3    G H
dtype: object
```

In [26]:
```
ser.str.split(" ")
```

Out[26]:
```
0    [A, B]
1    [C, D]
2    [E, F]
3    [G, H]
dtype: object
```