```python
import pandas as pd

df = pd.read_csv('C:\\Users\\maanz\\Downloads\\bike_sharing.csv')
df
```

```
                  datetime  season  holiday  workingday  weather
temp  \
0      2011-01-01 00:00:00       1        0           0        1
9.84
1      2011-01-01 01:00:00       1        0           0        1
9.02
2      2011-01-01 02:00:00       1        0           0        1
9.02
3      2011-01-01 03:00:00       1        0           0        1
9.84
4      2011-01-01 04:00:00       1        0           0        1
9.84
...                    ...     ...      ...         ...      ...    ..
.
10881  2012-12-19 19:00:00       4        0           1        1
15.58
10882  2012-12-19 20:00:00       4        0           1        1
14.76
10883  2012-12-19 21:00:00       4        0           1        1
13.94
10884  2012-12-19 22:00:00       4        0           1        1
13.94
10885  2012-12-19 23:00:00       4        0           1        1
13.12

        atemp  humidity  windspeed  casual  registered  count
0      14.395        81     0.0000       3          13     16
1      13.635        80     0.0000       8          32     40
2      13.635        80     0.0000       5          27     32
3      14.395        75     0.0000       3          10     13
4      14.395        75     0.0000       0           1      1
...       ...       ...        ...     ...         ...    ...
10881  19.695        50    26.0027       7         329    336
10882  17.425        57    15.0013      10         231    241
10883  15.910        61    15.0013       4         164    168
10884  17.425        61     6.0032      12         117    129
10885  16.665        66     8.9981       4          84     88

[10886 rows x 12 columns]
```

```python
df.shape
```

```
(10886, 12)
```

```python
df.head(2)
```

```
              datetime  season  holiday  workingday  weather  temp  
atemp  \
0  2011-01-01 00:00:00       1        0           0        1  9.84  
14.395   
1  2011-01-01 01:00:00       1        0           0        1  9.02  
13.635   

   humidity  windspeed  casual  registered  count  
0        81        0.0       3          13     16  
1        80        0.0       8          32     40  
```

```
df.tail(2)
```

```
                  datetime  season  holiday  workingday  weather  
temp  \
10884  2012-12-19 22:00:00       4        0           1        1  
13.94   
10885  2012-12-19 23:00:00       4        0           1        1  
13.12   

        atemp  humidity  windspeed  casual  registered  count  
10884  17.425        61     6.0032      12         117    129  
10885  16.665        66     8.9981       4          84     88  
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   datetime    10886 non-null  object 
 1   season      10886 non-null  int64  
 2   holiday     10886 non-null  int64  
 3   workingday  10886 non-null  int64  
 4   weather     10886 non-null  int64  
 5   temp        10886 non-null  float64
 6   atemp       10886 non-null  float64
 7   humidity    10886 non-null  int64  
 8   windspeed   10886 non-null  float64
 9   casual      10886 non-null  int64  
 10  registered  10886 non-null  int64  
 11  count       10886 non-null  int64  
dtypes: float64(3), int64(8), object(1)
memory usage: 1020.7+ KB
```

```
df.dtypes
```

```
datetime       object
season          int64
holiday         int64
```

```
workingday      int64
weather         int64
temp          float64
atemp         float64
humidity        int64
windspeed     float64
casual          int64
registered      int64
count           int64
dtype: object
```

```python
df.head(2)
```

```
              datetime  season  holiday  workingday  weather  temp
atemp  \
0  2011-01-01 00:00:00       1        0           0        1  9.84
14.395
1  2011-01-01 01:00:00       1        0           0        1  9.02
13.635

   humidity  windspeed  casual  registered  count
0        81        0.0       3          13     16
1        80        0.0       8          32     40
```

```python
import seaborn as sns
df2 = sns.load_dataset('tips')
df2.head()
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

```python
# check the shape
df2.shape
```

```
(244, 7)
```

```python
# find the max total_bill
# find the max tip given by customer
# find the min total bill
# find the min tip given by the customer
# find the maximum size
# find the average total_bill
# find the average tip
# find the count of male and female
# how many people are in size of more than 4
```

```
df2['total_bill'].max()
```

```
3.07
```

```
df2['total_bill'].mean()
```

```
19.785942622950824
```

```
df2['sex'].value_counts()
```

```
Male      157
Female     87
Name: sex, dtype: int64
```

```
# how many people are in size of more than 4
df2[df2['size']  > 4]
```

```
     total_bill   tip     sex smoker   day    time  size
125       29.80  4.20  Female     No  Thur   Lunch     6
141       34.30  6.70    Male     No  Thur   Lunch     6
142       41.19  5.00    Male     No  Thur   Lunch     5
143       27.05  5.00  Female     No  Thur   Lunch     6
155       29.85  5.14  Female     No   Sun  Dinner     5
156       48.17  5.00    Male     No   Sun  Dinner     6
185       20.69  5.00    Male     No   Sun  Dinner     5
187       30.46  2.00    Male    Yes   Sun  Dinner     5
216       28.15  3.00    Male    Yes   Sat  Dinner     5
```

```
# sort the data based on size in descending order
df2.sort_values(by = 'size', ascending = False)
```

```
     total_bill   tip     sex smoker   day    time  size
143       27.05  5.00  Female     No  Thur   Lunch     6
156       48.17  5.00    Male     No   Sun  Dinner     6
125       29.80  4.20  Female     No  Thur   Lunch     6
141       34.30  6.70    Male     No  Thur   Lunch     6
185       20.69  5.00    Male     No   Sun  Dinner     5
..          ...   ...     ...    ...   ...     ...   ...
105       15.36  1.64    Male    Yes   Sat  Dinner     2
67         3.07  1.00  Female    Yes   Sat  Dinner     1
222        8.58  1.92    Male    Yes   Fri   Lunch     1
111        7.25  1.00  Female     No   Sat  Dinner     1
82        10.07  1.83  Female     No  Thur   Lunch     1

[244 rows x 7 columns]
```

```
# groupby, crosstab, pivot_table, map, replace, isna/isnull, fillna, 
dropna
# duplicates, drop_duplicates, apply, concat, merge, join
```

```
df2.head()
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

```
male_df = df2[df2['sex'] == 'Male']
male_df
```

```
     total_bill   tip   sex smoker  day    time  size
1         10.34  1.66  Male     No  Sun  Dinner     3
2         21.01  3.50  Male     No  Sun  Dinner     3
3         23.68  3.31  Male     No  Sun  Dinner     2
5         25.29  4.71  Male     No  Sun  Dinner     4
6          8.77  2.00  Male     No  Sun  Dinner     2
..          ...   ...   ...    ...  ...     ...   ...
236       12.60  1.00  Male    Yes  Sat  Dinner     2
237       32.83  1.17  Male    Yes  Sat  Dinner     2
239       29.03  5.92  Male     No  Sat  Dinner     3
241       22.67  2.00  Male    Yes  Sat  Dinner     2
242       17.82  1.75  Male     No  Sat  Dinner     2

[157 rows x 7 columns]
```

```
female_df = df2[df2['sex'] == 'Female']
female_df
```

```
     total_bill   tip     sex smoker   day    time  size
0         16.99  1.01  Female     No   Sun  Dinner     2
4         24.59  3.61  Female     No   Sun  Dinner     4
11        35.26  5.00  Female     No   Sun  Dinner     4
14        14.83  3.02  Female     No   Sun  Dinner     2
16        10.33  1.67  Female     No   Sun  Dinner     3
..          ...   ...     ...    ...   ...     ...   ...
226       10.09  2.00  Female    Yes   Fri   Lunch     2
229       22.12  2.88  Female    Yes   Sat  Dinner     2
238       35.83  4.67  Female     No   Sat  Dinner     3
240       27.18  2.00  Female    Yes   Sat  Dinner     2
243       18.78  3.00  Female     No  Thur  Dinner     2

[87 rows x 7 columns]
```

```
male_df['total_bill'].mean()
```

```
20.744076433121034
```

```
female_df['total_bill'].mean()
```

```
18.056896551724137
```

```python
df2.groupby(by = 'sex')['total_bill'].mean()
```

```
sex
Male      20.744076
Female    18.056897
Name: total_bill, dtype: float64
```

```python
# find the average total bill for smokers and non smokers using groupby
df2.groupby(by = 'smoker')['total_bill'].mean()
```

```
smoker
Yes    20.756344
No     19.188278
Name: total_bill, dtype: float64
```

```python
# find the average bill of male somkers and non smokers, female smokers and non smokers

df2.groupby( by =  ['sex', 'smoker'] )['total_bill'].mean()
```

```
sex     smoker
Male    Yes       22.284500
        No        19.791237
Female  Yes       17.977879
        No        18.105185
Name: total_bill, dtype: float64
```

```python
# find the average tip for somkers and non smokers at different time
df2.groupby( by = ['smoker', 'time'])['tip'].max()
```

```
smoker  time
Yes     Lunch      5.0
        Dinner    10.0
No      Lunch      6.7
        Dinner     9.0
Name: tip, dtype: float64
```

```python
# pivot_table
pd.pivot_table(data = df2,
               index = 'sex',
               columns= 'smoker',
               values = 'total_bill',
               aggfunc = 'mean')
```

```
smoker         Yes         No
sex
Male    22.284500  19.791237
Female  17.977879  18.105185
```

```
# using pivot table find out the average bill for male and female on
each day
pd.pivot_table(data = df2,
              index = 'day',
              columns = 'sex',
              values = 'total_bill',
              aggfunc = 'mean')
```

```
sex        Male     Female
day
Thur  18.714667  16.715312
Fri   19.857000  14.145556
Sat   20.802542  19.680357
Sun   21.887241  19.872222
```

```
# using pivot table find out the average bill for male and female on
each day
pd.pivot_table(data = df2,
              index = 'day',
              columns = 'sex',
              values = ['total_bill', 'tip'],
              aggfunc = 'mean')
```

```
           tip           total_bill
sex       Male    Female       Male     Female
day
Thur  2.980333  2.575625  18.714667  16.715312
Fri   2.693000  2.781111  19.857000  14.145556
Sat   3.083898  2.801786  20.802542  19.680357
Sun   3.220345  3.367222  21.887241  19.872222
```

```
# find the minimum total bill and tip using pivot table for male and
# female who are smokers and non smokers
pd.pivot_table(data = df2,
              index = 'sex',
              columns = 'smoker',
              values = ['total_bill', 'tip'],
              aggfunc = 'min')
```

```
        tip       total_bill
smoker  Yes    No        Yes    No
sex
Male    1.0  1.25       7.25  7.51
Female  1.0  1.00       3.07  7.25
```

```
pd.pivot_table(data = df2,
              index = ['sex' , 'time'],
              columns = 'smoker',
              values = 'total_bill',
              aggfunc = 'mean')
```

```
smoker               Yes         No
sex    time
Male   Lunch   17.374615  18.486500
       Dinner  23.642553  20.130130
Female Lunch   17.431000  15.902400
       Dinner  18.215652  20.004138
```

```
pd.pivot_table(data = df2,
              index = ['sex'],
              columns = ['smoker', 'day'],
              values = 'total_bill',
              aggfunc = 'mean')
```

```
smoker        Yes                                        No          \
day          Thur        Fri        Sat        Sun    Thur     Fri
sex
Male    19.171000  20.452500  21.837778  26.141333  18.4865  17.475
Female  19.218571  12.654286  20.266667  16.540000  16.0144  19.365

smoker
day           Sat        Sun
sex
Male    19.929063  20.403256
Female  19.003846  20.824286
```

```
pd.pivot_table(data = df2,
              index = 'sex',
              columns = 'smoker',
              values = 'total_bill',
              aggfunc = ['mean','sum', 'min', 'max'])
```

```
             mean                 sum              min          max

smoker        Yes         No      Yes       No    Yes    No    Yes
No
sex

Male    22.284500  19.791237  1337.07  1919.75  7.25  7.51  50.81
48.33
Female  17.977879  18.105185   593.27   977.68  3.07  7.25  44.30
35.83
```

```
df2.shape
```

```
(244, 7)
```

```
pd.crosstab(index = df2['time'],
            columns = df2['smoker'])
```

```
smoker  Yes   No
time
Lunch    23   45
Dinner   70  106
```

```
# find the count of smokers and non smokers on each day
```

```
d1 = {'PID' : [101,102,103],
     'Pname' : ['Laptop', 'Ipad', 'Keyboard'],
     'Price' : [50000, 15000, 1200]}
```

```
product_df = pd.DataFrame(d1)
product_df
```

```
   PID     Pname  Price
0  101    Laptop  50000
1  102      Ipad  15000
2  103  Keyboard   1200
```

```
d2 = {'OID' : ['O1112', 'O1113', 'O1114', 'O1115', 'O1116'],
     'PID' : [102,101,102,101,102],
     'Qty' : [2,3,5,10,15],
     'CID' : ['C113', 'C114', 'C111','C111','C114']}
order_df = pd.DataFrame(d2)
order_df
```

```
     OID  PID  Qty   CID
0  O1112  102    2  C113
1  O1113  101    3  C114
2  O1114  102    5  C111
3  O1115  101   10  C111
4  O1116  102   15  C114
```

```
df3 = pd.merge(product_df, order_df, on = 'PID' )
df3
```

```
   PID   Pname  Price    OID  Qty   CID
0  101  Laptop  50000  O1113    3  C114
1  101  Laptop  50000  O1115   10  C111
2  102    Ipad  15000  O1112    2  C113
3  102    Ipad  15000  O1114    5  C111
4  102    Ipad  15000  O1116   15  C114
```

```
df3 = pd.merge(left = product_df,
               right = order_df,
               on = 'PID' )
df3
```

```
   PID   Pname  Price    OID  Qty   CID
0  101  Laptop  50000  O1113    3  C114
1  101  Laptop  50000  O1115   10  C111
2  102    Ipad  15000  O1112    2  C113
3  102    Ipad  15000  O1114    5  C111
4  102    Ipad  15000  O1116   15  C114
```

```
df3 = pd.merge(left = product_df,
               right = order_df,
               on = 'PID',
               how = 'left' )
df3
```

```
   PID     Pname  Price    OID   Qty   CID
0  101    Laptop  50000  O1113   3.0  C114
1  101    Laptop  50000  O1115  10.0  C111
2  102      Ipad  15000  O1112   2.0  C113
3  102      Ipad  15000  O1114   5.0  C111
4  102      Ipad  15000  O1116  15.0  C114
5  103  Keyboard   1200    NaN   NaN   NaN
```

```
d2 = {'OID' : ['O1112', 'O1113', 'O1114', 'O1115', 'O1116', 'O1117'],
     'PID' : [102,101,102,101,102, 105],
     'Qty' : [2,3,5,10,15, 6],
     'CID' : ['C113', 'C114', 'C111','C111','C114', 'C113']}
order_df = pd.DataFrame(d2)
order_df
```

```
     OID  PID  Qty   CID
0  O1112  102    2  C113
1  O1113  101    3  C114
2  O1114  102    5  C111
3  O1115  101   10  C111
4  O1116  102   15  C114
5  O1117  105    6  C113
```

```
pd.merge(left = product_df,
        right = order_df,
        on = 'PID',
        how = 'right')
```

```
   PID   Pname    Price    OID  Qty   CID
0  102    Ipad  15000.0  O1112    2  C113
1  101  Laptop  50000.0  O1113    3  C114
2  102    Ipad  15000.0  O1114    5  C111
3  101  Laptop  50000.0  O1115   10  C111
```

```
4  102    Ipad  15000.0  O1116   15  C114
5  105     NaN      NaN  O1117    6  C113
```

```
pd.merge(left = product_df,
        right = order_df,
        on = 'PID',
        how = 'outer')
```

```
   PID     Pname    Price    OID   Qty   CID
0  101    Laptop  50000.0  O1113   3.0  C114
1  101    Laptop  50000.0  O1115  10.0  C111
2  102      Ipad  15000.0  O1112   2.0  C113
3  102      Ipad  15000.0  O1114   5.0  C111
4  102      Ipad  15000.0  O1116  15.0  C114
5  103  Keyboard   1200.0    NaN   NaN   NaN
6  105       NaN      NaN  O1117   6.0  C113
```

```
product_df.rename(columns = {'PID' : 'product_id'}, inplace=True)
```

```
product_df
```

```
   product_id     Pname  Price
0         101    Laptop  50000
1         102      Ipad  15000
2         103  Keyboard   1200
```

```
order_df
```

```
     OID  PID  Qty   CID
0  O1112  102    2  C113
1  O1113  101    3  C114
2  O1114  102    5  C111
3  O1115  101   10  C111
4  O1116  102   15  C114
5  O1117  105    6  C113
```

```
pd.merge(left = product_df,
        right = order_df,
        left_on = 'product_id',
        right_on = 'PID',
        how = 'inner')
```

```
   product_id   Pname  Price    OID  PID  Qty   CID
0         101  Laptop  50000  O1113  101    3  C114
1         101  Laptop  50000  O1115  101   10  C111
2         102    Ipad  15000  O1112  102    2  C113
3         102    Ipad  15000  O1114  102    5  C111
4         102    Ipad  15000  O1116  102   15  C114
```

```
# concat
d3 = {'Emp ID' : ['E1113', 'E1115', 'E1116'],
     'Dsignation' : ['Analyst', 'Assosciate', 'Manager']}
emp_df = pd.DataFrame(d3)
emp_df
```

```
  Emp ID  Dsignation
0  E1113     Analyst
1  E1115  Assosciate
2  E1116     Manager
```

```
d3 = {'Emp ID' : ['E1118', 'E11120', 'E11176'],
     'Dsignation' : ['Analyst', 'Analyst', 'Assistant Manager']}
emp_df1 = pd.DataFrame(d3)
emp_df1
```

```
   Emp ID         Dsignation
0   E1118            Analyst
1  E11120            Analyst
2  E11176  Assistant Manager
```

```
pd.concat([emp_df, emp_df1] )
```

```
   Emp ID         Dsignation
0   E1113            Analyst
1   E1115         Assosciate
2   E1116            Manager
0   E1118            Analyst
1  E11120            Analyst
2  E11176  Assistant Manager
```

```
master_emp_df = pd.concat([emp_df, emp_df1] , ignore_index = True)
master_emp_df
```

```
   Emp ID         Dsignation
0   E1113            Analyst
1   E1115         Assosciate
2   E1116            Manager
3   E1118            Analyst
4  E11120            Analyst
5  E11176  Assistant Manager
```

```
emp_df
```

```
  Emp ID  Dsignation
0  E1113     Analyst
1  E1115  Assosciate
2  E1116     Manager
```

```
d4 = {'Salary' : [400000, 750000, 1300000],
     'Dept' : ['Analytics', 'IT', 'Analytics']}
```

```
emp_df2 = pd.DataFrame(d4)
emp_df2
```

```
    Salary       Dept
0   400000  Analytics
1   750000         IT
2  1300000  Analytics
```

```
pd.concat([emp_df, emp_df2], axis = 1)
```

```
  Emp ID  Dsignation   Salary       Dept
0  E1113     Analyst   400000  Analytics
1  E1115  Assosciate   750000         IT
2  E1116     Manager  1300000  Analytics
```

```
# replace/map/apply

df2['smoker'].replace(to_replace=['No', 'Yes'],
                      value = [0,1], inplace= True)

df2
```

```
     total_bill   tip     sex  smoker   day    time  size
0         16.99  1.01  Female       0   Sun  Dinner     2
1         10.34  1.66    Male       0   Sun  Dinner     3
2         21.01  3.50    Male       0   Sun  Dinner     3
3         23.68  3.31    Male       0   Sun  Dinner     2
4         24.59  3.61  Female       0   Sun  Dinner     4
..          ...   ...     ...     ...   ...     ...   ...
239       29.03  5.92    Male       0   Sat  Dinner     3
240       27.18  2.00  Female       1   Sat  Dinner     2
241       22.67  2.00    Male       1   Sat  Dinner     2
242       17.82  1.75    Male       0   Sat  Dinner     2
243       18.78  3.00  Female       0  Thur  Dinner     2

[244 rows x 7 columns]
```

```
# map
df2['sex'].map( lambda x : 0 if x=='Female' else 1 )
```

```
0      0
1      1
2      1
3      1
4      0
      ..
239    1
240    0
241    1
242    1
```

```
243    0
Name: sex, Length: 244, dtype: category
Categories (2, int64): [1, 0]
```

```python
df2['sex'] = df2['sex'].map( lambda x : 0 if x=='Female' else 1 )
df2
```

```
     total_bill   tip sex  smoker   day    time  size
0         16.99  1.01   0       0   Sun  Dinner     2
1         10.34  1.66   1       0   Sun  Dinner     3
2         21.01  3.50   1       0   Sun  Dinner     3
3         23.68  3.31   1       0   Sun  Dinner     2
4         24.59  3.61   0       0   Sun  Dinner     4
..          ...   ...  ..     ...   ...     ...   ...
239       29.03  5.92   1       0   Sat  Dinner     3
240       27.18  2.00   0       1   Sat  Dinner     2
241       22.67  2.00   1       1   Sat  Dinner     2
242       17.82  1.75   1       0   Sat  Dinner     2
243       18.78  3.00   0       0  Thur  Dinner     2

[244 rows x 7 columns]
```

```python
# apply
df2['smoker'].apply(lambda x :  'No' if x == 0 else 'Yes')
```

```
0       No
1       No
2       No
3       No
4       No
      ...
239     No
240    Yes
241    Yes
242     No
243     No
Name: smoker, Length: 244, dtype: object
```

```python
df2[  ['total_bill' , 'tip'] ]
```

```
     total_bill   tip
0         16.99  1.01
1         10.34  1.66
2         21.01  3.50
3         23.68  3.31
4         24.59  3.61
..          ...   ...
239       29.03  5.92
240       27.18  2.00
241       22.67  2.00
242       17.82  1.75
```

```
243       18.78  3.00

[244 rows x 2 columns]
```

```
df2[  ['total_bill' , 'tip'] ].apply(lambda row : row['tip'] +
row['total_bill'], axis = 1 )
```

```
0      18.00
1      12.00
2      24.51
3      26.99
4      28.20
       ...
239    34.95
240    29.18
241    24.67
242    19.57
243    21.78
Length: 244, dtype: float64
```

```
# replace/map/apply

# isnull(), isna(), fillna, dropna

df_1 = sns.load_dataset('titanic')
df_1.head()
```

```
   survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0         0       3    male  22.0      1      0   7.2500        S
Third
1         1       1  female  38.0      1      0  71.2833        C
First
2         1       3  female  26.0      0      0   7.9250        S
Third
3         1       1  female  35.0      1      0  53.1000        S
First
4         0       3    male  35.0      0      0   8.0500        S
Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

```
df_1.shape
```

```
(891, 15)
```

```
df_1.isna()
```

```
     survived  pclass    sex    age  sibsp  parch   fare  embarked
class  \
0       False   False  False  False  False  False  False     False
False
1       False   False  False  False  False  False  False     False
False
2       False   False  False  False  False  False  False     False
False
3       False   False  False  False  False  False  False     False
False
4       False   False  False  False  False  False  False     False
False
..        ...     ...    ...    ...    ...    ...    ...       ...
...
886     False   False  False  False  False  False  False     False
False
887     False   False  False  False  False  False  False     False
False
888     False   False  False   True  False  False  False     False
False
889     False   False  False  False  False  False  False     False
False
890     False   False  False  False  False  False  False     False
False

       who  adult_male   deck  embark_town  alive  alone
0    False       False   True        False  False  False
1    False       False  False        False  False  False
2    False       False   True        False  False  False
3    False       False  False        False  False  False
4    False       False   True        False  False  False
..     ...         ...    ...          ...    ...    ...
886  False       False   True        False  False  False
887  False       False  False        False  False  False
888  False       False   True        False  False  False
889  False       False  False        False  False  False
890  False       False   True        False  False  False

[891 rows x 15 columns]
```

```
df_1.isna().sum()
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
```

```
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
# fill, drop
df_1['age'].fillna('unknown')
```

```
0         22.0
1         38.0
2         26.0
3         35.0
4         35.0
        ...
886       27.0
887       19.0
888    unknown
889       26.0
890       32.0
Name: age, Length: 891, dtype: object
```

```
df_1['age'].fillna('uknown', inplace = True )
```

```
df_1
```

```
     survived  pclass     sex     age  sibsp  parch     fare embarked
class  \
0           0       3    male    22.0      1      0   7.2500        S
Third
1           1       1  female    38.0      1      0  71.2833        C
First
2           1       3  female    26.0      0      0   7.9250        S
Third
3           1       1  female    35.0      1      0  53.1000        S
First
4           0       3    male    35.0      0      0   8.0500        S
Third
..        ...     ...     ...     ...    ...    ...      ...      ...
...
886         0       2    male    27.0      0      0  13.0000        S
Second
887         1       1  female    19.0      0      0  30.0000        S
First
```

```
888         0        3  female  uknown      1      2  23.4500        S
Third
889         1        1    male    26.0      0      0  30.0000        C
First
890         0        3    male    32.0      0      0   7.7500        Q
Third

       who  adult_male deck  embark_town alive  alone
0      man        True  NaN  Southampton    no  False
1    woman       False    C    Cherbourg   yes  False
2    woman       False  NaN  Southampton   yes   True
3    woman       False    C  Southampton   yes  False
4      man        True  NaN  Southampton    no   True
..     ...         ...  ...          ...   ...    ...
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]
```

```
df_1.dropna()
```

```
     survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
1           1        1  female  38.0      1      0  71.2833        C
First
3           1        1  female  35.0      1      0  53.1000        S
First
6           0        1    male  54.0      0      0  51.8625        S
First
10          1        3  female   4.0      1      1  16.7000        S
Third
11          1        1  female  58.0      0      0  26.5500        S
First
..        ...     ...     ...   ...    ...    ...      ...      ...
...
871         1        1  female  47.0      1      1  52.5542        S
First
872         0        1    male  33.0      0      0   5.0000        S
First
879         1        1  female  56.0      0      1  83.1583        C
First
887         1        1  female  19.0      0      0  30.0000        S
First
889         1        1    male  26.0      0      0  30.0000        C
First
```

```
       who  adult_male deck  embark_town alive  alone
1    woman       False    C    Cherbourg   yes  False
3    woman       False    C  Southampton   yes  False
6      man        True    E  Southampton    no   True
10   child       False    G  Southampton   yes  False
11   woman       False    C  Southampton   yes   True
..     ...         ...  ...          ...   ...    ...
871  woman       False    D  Southampton   yes  False
872    man        True    B  Southampton    no   True
879  woman       False    C    Cherbourg   yes  False
887  woman       False    B  Southampton   yes   True
889    man        True    C    Cherbourg   yes   True

[201 rows x 15 columns]
```

```
df_1.dropna( axis = 1)
```

```
     survived  pclass     sex     age  sibsp  parch     fare   class
who  \
0           0       3    male    22.0      1      0   7.2500   Third
man
1           1       1  female    38.0      1      0  71.2833   First
woman
2           1       3  female    26.0      0      0   7.9250   Third
woman
3           1       1  female    35.0      1      0  53.1000   First
woman
4           0       3    male    35.0      0      0   8.0500   Third
man
..        ...     ...     ...     ...    ...    ...      ...     ...
...
886         0       2    male    27.0      0      0  13.0000  Second
man
887         1       1  female    19.0      0      0  30.0000   First
woman
888         0       3  female  uknown      1      2  23.4500   Third
woman
889         1       1    male    26.0      0      0  30.0000   First
man
890         0       3    male    32.0      0      0   7.7500   Third
man

     adult_male alive  alone
0          True    no  False
1         False   yes  False
2         False   yes   True
3         False   yes  False
4          True    no   True
..          ...   ...    ...
```

```
886         True    no   True
887        False   yes   True
888        False    no  False
889         True   yes   True
890         True    no   True

[891 rows x 12 columns]
```

```
# handling duplicates
d1 = {'PID' : [101,102,103, 101],
     'Pname' : ['Laptop', 'Ipad', 'Keyboard', 'Laptop'],
     'Price' : [50000, 15000, 1200, 50000]}

product_df = pd.DataFrame(d1)
product_df
```

```
   PID     Pname  Price
0  101    Laptop  50000
1  102      Ipad  15000
2  103  Keyboard   1200
3  101    Laptop  50000
```

```
product_df.duplicated()
```

```
0    False
1    False
2    False
3     True
dtype: bool
```

```
product_df[product_df.duplicated()]
```

```
   PID   Pname  Price
3  101  Laptop  50000
```

```
product_df.drop_duplicates()
```

```
   PID     Pname  Price
0  101    Laptop  50000
1  102      Ipad  15000
2  103  Keyboard   1200
```

```
product_df.drop_duplicates(inplace=True)

product_df
```

```
   PID     Pname  Price
0  101    Laptop  50000
1  102      Ipad  15000
2  103  Keyboard   1200
```

```
df_1 = sns.load_dataset('titanic')
df_1.head()

   survived  pclass     sex   age  sibsp  parch     fare embarked
class  \
0         0       3    male  22.0      1      0   7.2500        S
Third
1         1       1  female  38.0      1      0  71.2833        C
First
2         1       3  female  26.0      0      0   7.9250        S
Third
3         1       1  female  35.0      1      0  53.1000        S
First
4         0       3    male  35.0      0      0   8.0500        S
Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True

# find the average age of male and female passengers using group by
# find the average fare of male and female passengers in each class
using pivot table
# do a cross tab b/w class and alive column

pd.pivot_table()

pd.crosstab(index = df_1['class'], columns = df_1['alive'] )

alive    no  yes
class
First    80  136
Second   97   87
Third   372  119

# merge , concat, replace(to_replace, value, inplace), map(lambda x),
apply,
# isna, fillna, dropna, duplicated, drop_duplicates
```