```
# create a table with columns and rows
# column name : students,technology,marks
# add the column teacher
# and delete the technology later
# and try to to practice on all the example methods
# 15 to 20 records
dataset = {
    "students": [
        "Navya", "Vyshnavi", "Keerthi", "Amitha", "Vidya",
        "Raksha", "Abinaya", "Rohit", "Joshua", "Sameer",
        "Akhil", "Manish", "Divya", "Shreya", "Ankit",
        "Priya", "Suresh", "Megha", "Rahul", "Sailesh"
    ],
    "technology": [
        "Python", "Java", "Python", "C++", "Java",
        "Python", "Java", "C++", "Python", "Java",
        "C++", "Python", "Java", "C++", "Python",
        "Java", "Python", "C++", "Java", "Python"
    ],
    "marks": [
        79, 68, 49, 64, 69,
        98, 84, 61, 44, 44,
        59, 93, 67, 96, 85,
        46, 41, 82, 74, 74
    ]
}
data_frm = pd.DataFrame(dataset)
```

```
print(data_frm)
print(data_frm.head(3))
print(data_frm.tail(3))
print(data_frm.shape)
print(data_frm.dtypes)
print(data_frm.columns)
print(data_frm.describe())
print(data_frm['students'])
print(data_frm["marks"]>75)
data_frm.drop('technology', axis=1, inplace=True)
data_frm["teacher"] = [
    "Mr. Rao", "Ms. Priya", "Mr. Khan", "Ms. Lee", "Mrs. Das",
    "Mr. Joshi", "Ms. Smith", "Mr. Wang", "Mrs. Patel", "Ms. Kim",
    "Mr. Roy", "Ms. Gupta", "Mr. Chen", "Mrs. Brown", "Ms. Singh",
    "Mr. Kumar", "Ms. White", "Mrs. Thomas", "Mr. Clark", "Ms. Lewis"
]
print(data_frm)
```

Output :

```
students technology  marks
0      Navya     Python     79
1    Vyshnavi       Java     68
2     Keerthi     Python     49
3      Amitha        C++     64
4       Vidya       Java     69
5      Raksha     Python     98
6     Abinaya       Java     84
```

```
7      Rohit       C++     61
8     Joshua     Python     44
9     Sameer       Java     44
10     Akhil       C++     59
11    Manish     Python     93
12     Divya       Java     67
13    Shreya       C++     96
14     Ankit     Python     85
15     Priya       Java     46
16    Suresh     Python     41
17     Megha       C++     82
18     Rahul       Java     74
19   Sailesh     Python     74
   students technology  marks
0     Navya     Python     79
1  Vyshnavi       Java     68
2   Keerthi     Python     49
   students technology  marks
17    Megha       C++     82
18    Rahul       Java     74
19  Sailesh     Python     74
(20, 3)
students      object
technology    object
marks          int64
dtype: object
Index(['students', 'technology', 'marks'], dtype='object')
```

	marks
count	20.000000
mean	68.850000
std	17.921686
min	41.000000
25%	56.500000
50%	68.500000
75%	82.500000
max	98.000000

0	Navya
1	Vyshnavi
2	Keerthi
3	Amitha
4	Vidya
5	Raksha
6	Abinaya
7	Rohit
8	Joshua
9	Sameer
10	Akhil
11	Manish
12	Divya
13	Shreya
14	Ankit
15	Priya
16	Suresh
17	Megha

18 Rahul

19 Sailesh

Name: students, dtype: object

0 True

1 False

2 False

3 False

4 False

5 True

6 True

7 False

8 False

9 False

10 False

11 True

12 False

13 True

14 True

15 False

16 False

17 True

18 False

19 False

Name: marks, dtype: bool

	students	marks	teacher
0	Navya	79	Mr. Rao
1	Vyshnavi	68	Ms. Priya

2	Keerthi	49	Mr. Khan
3	Amitha	64	Ms. Lee
4	Vidya	69	Mrs. Das
5	Raksha	98	Mr. Joshi
6	Abinaya	84	Ms. Smith
7	Rohit	61	Mr. Wang
8	Joshua	44	Mrs. Patel
9	Sameer	44	Ms. Kim
10	Akhil	59	Mr. Roy
11	Manish	93	Ms. Gupta
12	Divya	67	Mr. Chen
13	Shreya	96	Mrs. Brown
14	Ankit	85	Ms. Singh
15	Priya	46	Mr. Kumar
16	Suresh	41	Ms. White
17	Megha	82	Mrs. Thomas
18	Rahul	74	Mr. Clark
19	Sailesh	74	Ms. Lewis

Pandas Assessment

```
import pandas as pd
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Frank', 'Grace', 'Helen'],
    'Age': [25, 30, 35, 40, 28, 32, 45, 38],
    'Department': ['HR', 'Engineering', 'Finance', 'HR', 'Engineering', 'Finance', 'HR', 'Finance'],
    'Salary': [50000, 60000, 70000, 52000, None, 58000, None, 62000]
}
df = pd.DataFrame(data)
df.to_csv('data_csv_using_pandas.csv', index=False)
print("DataFrame with 8 records:")
print(df)

#1. Load data.csv and display the first 5 rows
df = pd.read_csv('data_csv_using_pandas.csv')
print("First 5 rows:")
print(df.head())

#2. Check the shape of the DataFrame (rows, columns)
print("\nShape of DataFrame (rows, columns):")
print(df.shape)

#3. Get summary statistics for numeric columns
print("\nSummary statistics (numeric columns):")
print(df.describe())
```

```
# 4. Extract the 'Age' column as a Series
print("\n'Age' column:")
print(df['Age'])

#5. Show rows where 'Salary' > 50000
print("\nRows where Salary > 50000:")
print(df[df['Salary'] > 50000])

#6. Rows where 'Department' == 'HR' and 'Age' > 30
print("\nRows where Department == 'HR' and Age > 30:")
print(df[(df['Department'] == 'HR') & (df['Age'] > 30)])

#7. Find which columns have NaN values and how many
print("\nMissing values per column:")
print(df.isnull().sum())

#8. Fill NaN values in 'Salary' with 0
df_filled = df.copy()
df_filled['Salary'] = df_filled['Salary'].fillna(0)
print("\n'Salary' column after filling NaNs with 0:")
print(df_filled['Salary'])

#9. Drop duplicate rows and reset the index
df_no_duplicates = df.drop_duplicates().reset_index(drop=True)
print("\nDataFrame after dropping duplicates and resetting index:")
print(df_no_duplicates)
```

```
#10. Sort rows by "Age" in descending order
df_sorted = df.sort_values(by='Age', ascending=False)
print("\nDataFrame sorted by Age (descending):")
print(df_sorted)

#11. Group by "Department" and find average "Salary"
print("\nAverage Salary by Department:")
print(df.groupby('Department')['Salary'].mean())

#12. Count unique departments in the "Department" column
print("\nNumber of unique departments:")
print(df['Department'].nunique())
```

Output :

```
DataFrame with 8 records:
      Name  Age   Department   Salary
0    Alice   25           HR  50000.0
1      Bob   30  Engineering  60000.0
2  Charlie   35      Finance  70000.0
3    David   40           HR  52000.0
4      Eva   28  Engineering      NaN
5    Frank   32      Finance  58000.0
6    Grace   45           HR      NaN
7    Helen   38      Finance  62000.0
First 5 rows:
      Name  Age   Department   Salary
0    Alice   25           HR  50000.0
```

```
1      Bob   30  Engineering  60000.0
2  Charlie   35      Finance  70000.0
3    David   40           HR  52000.0
4      Eva   28  Engineering      NaN

Shape of DataFrame (rows, columns):
(8, 4)

Summary statistics (numeric columns):
             Age        Salary
count   8.000000      6.000000
mean   34.125000  58666.666667
std     6.664136   7229.568913
min    25.000000  50000.000000
25%    29.500000  53500.000000
50%    33.500000  59000.000000
75%    38.500000  61500.000000
max    45.000000  70000.000000

'Age' column:
0    25
1    30
2    35
3    40
4    28
5    32
6    45
```

```
7    38
Name: Age, dtype: int64

Rows where Salary > 50000:
      Name  Age   Department   Salary
1      Bob   30  Engineering  60000.0
2  Charlie   35      Finance  70000.0
3    David   40           HR  52000.0
5    Frank   32      Finance  58000.0
7    Helen   38      Finance  62000.0

Rows where Department == 'HR' and Age > 30:
    Name  Age Department   Salary
3  David   40         HR  52000.0
6  Grace   45         HR      NaN

Missing values per column:
Name          0
Age           0
Department    0
Salary        2
dtype: int64

'Salary' column after filling NaNs with 0:
0    50000.0
1    60000.0
2    70000.0
```

3   52000.0

4   0.0

5   58000.0

6   0.0

7   62000.0

Name: Salary, dtype: float64

DataFrame after dropping duplicates and resetting index:

      Name Age   Department   Salary

0   Alice   25   HR 50000.0

1   Bob   30 Engineering 60000.0

2 Charlie   35   Finance 70000.0

3   David   40   HR 52000.0

4   Eva   28 Engineering   NaN

5   Frank   32   Finance 58000.0

6   Grace   45   HR   NaN

7   Helen   38   Finance 62000.0

DataFrame sorted by Age (descending):

      Name Age   Department   Salary

6   Grace   45   HR   NaN

3   David   40   HR 52000.0

7   Helen   38   Finance 62000.0

2 Charlie   35   Finance 70000.0

5   Frank   32   Finance 58000.0

1   Bob   30 Engineering 60000.0

4   Eva   28 Engineering   NaN

0 Alice 25 HR 50000.0

Average Salary by Department:

Department

Engineering 60000.000000

Finance 63333.333333

HR 51000.000000

Name: Salary, dtype: float64

Number of unique departments:

3