# Data processing_2 (1)

August 19, 2022

[42]:
```
from sklearn.datasets import load_diabetes
import pandas as pd
import numpy as np
import seaborn as sns
```

[2]:
```
data = load_diabetes()
```

[3]:
```
print(data.DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

  :Number of Instances: 442

  :Number of Attributes: First 10 columns are numeric predictive values

  :Target: Column 11 is a quantitative measure of disease progression one year
after baseline

  :Attribute Information:
      - age     age in years
      - sex
      - bmi     body mass index
      - bp      average blood pressure
      - s1      tc, total serum cholesterol
      - s2      ldl, low-density lipoproteins
      - s3      hdl, high-density lipoproteins
      - s4      tch, total cholesterol / HDL
```

```
      - s5      ltg, possibly log of serum triglycerides level
      - s6      glu, blood sugar level

    Note: Each of these 10 feature variables have been mean centered and scaled by
    the standard deviation times `n_samples` (i.e. the sum of squares of each column
    totals 1).

    Source URL:
    https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

    For more information see:
    Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least
    Angle Regression," Annals of Statistics (with discussion), 407-499.
    (https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)
```

[4]:
```
type(data)
```

[4]:
```
sklearn.utils.Bunch
```

[5]:
```
data.feature_names
```

[5]:
```
['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']
```

[6]:
```
my_dataframe = pd.DataFrame(data.data, columns = data.feature_names)
```

[7]:
```
my_dataframe.dtypes
```

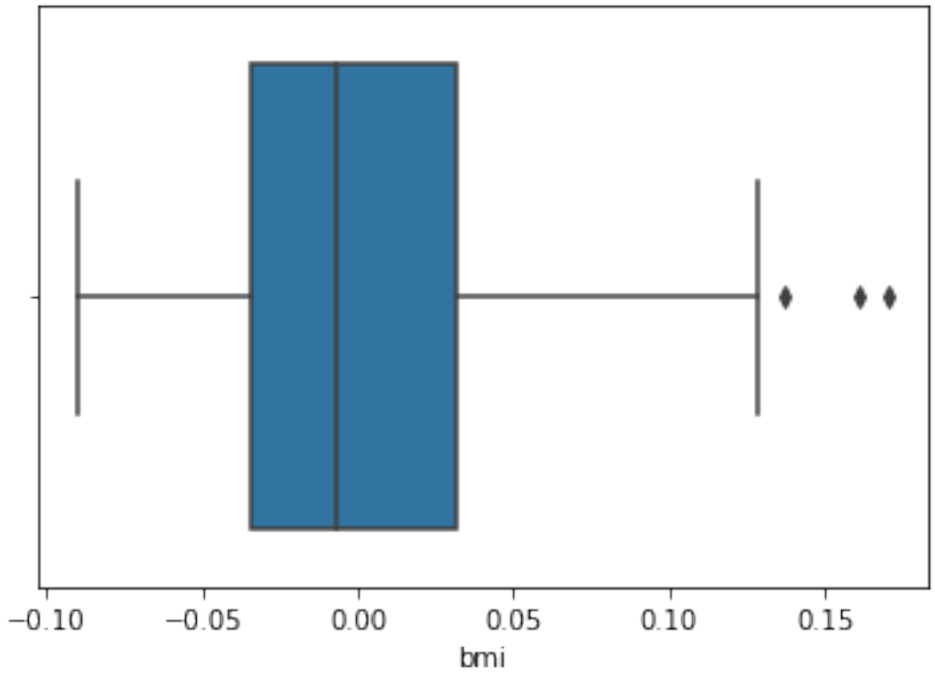
[7]:
```
age    float64
sex    float64
bmi    float64
bp     float64
s1     float64
s2     float64
s3     float64
s4     float64
s5     float64
s6     float64
dtype: object
```

### 0.0.1 Handling Outliers

[8]:
```
sns.boxplot(my_dataframe['bmi']);
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
```

```
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```




Three outliers observed beyond 0.12

[9]:
```
filter = my_dataframe['bmi'].values < 0.12
```

[10]:
```
my_dataframe_filtered = my_dataframe[filter]
```

[11]:
```
my_dataframe[my_dataframe['bmi'].values > 0.12]
```

[11]:

| | age | sex | bmi | bp | s1 | s2 | s3 | \ |
|---|---|---|---|---|---|---|---|---|
| 32 | 0.034443 | 0.050680 | 0.125287 | 0.028758 | -0.053855 | -0.012900 | -0.102307 | |
| 145 | -0.041840 | -0.044642 | 0.128521 | 0.063187 | -0.033216 | -0.032629 | 0.011824 | |
| 256 | -0.049105 | -0.044642 | 0.160855 | -0.046985 | -0.029088 | -0.019790 | -0.047082 | |
| 262 | -0.016412 | 0.050680 | 0.127443 | 0.097616 | 0.016318 | 0.017475 | -0.021311 | |
| 366 | -0.045472 | 0.050680 | 0.137143 | -0.015999 | 0.041086 | 0.031880 | -0.043401 | |
| 367 | -0.009147 | 0.050680 | 0.170555 | 0.014987 | 0.030078 | 0.033759 | -0.021311 | |
| 405 | 0.048974 | 0.050680 | 0.123131 | 0.083844 | -0.104765 | -0.100895 | -0.069172 | |

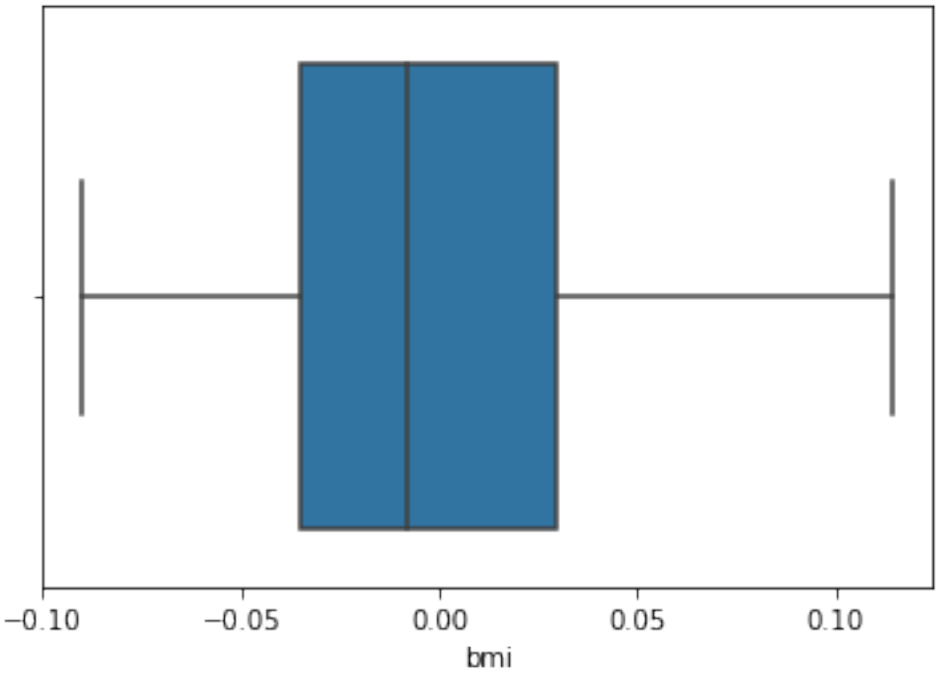
| | s4 | s5 | s6 |
|---|---|---|---|
| 32 | 0.108111 | 0.000271 | 0.027917 |
| 145 | -0.039493 | -0.015998 | -0.050783 |
| 256 | 0.034309 | 0.028017 | 0.011349 |
| 262 | 0.034309 | 0.034864 | 0.003064 |

```
366  0.071210  0.071022  0.048628
367  0.034309  0.033657  0.032059
405 -0.002592  0.036646 -0.030072
```

[12]:
```
sns.boxplot(my_dataframe_filtered['bmi'])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

[12]: <AxesSubplot:xlabel='bmi'>




[19]:
```
data = pd.DataFrame({'Name':
↪['Jitendra','Michael','Manas','Gayatri','Jitendra','Jitendra','Michael','Manas',
                          'Sushil'],
                     'Subject':['Python','Data Science','Data␣
↪Science','Python','Data Science','Python','Python','Python','Data Science'],
                     'Marks':[9,7,8,9,6,5,9,8,4]})
```

[20]:
```
data
```

[20]:

| | Name | Subject | Marks |
|---|---|---|---|
| 0 | Jitendra | Python | 9 |
| 1 | Michael | Data Science | 7 |
| 2 | Manas | Data Science | 8 |
| 3 | Gayatri | Python | 9 |
| 4 | Jitendra | Data Science | 6 |
| 5 | Jitendra | Python | 5 |
| 6 | Michael | Python | 9 |
| 7 | Manas | Python | 8 |
| 8 | Sushil | Data Science | 4 |

[21]:
```
data.groupby('Name').groups
```

[21]:
```
{'Gayatri': [3], 'Jitendra': [0, 4, 5], 'Manas': [2, 7], 'Michael': [1, 6],
'Sushil': [8]}
```

[22]:
```
data.groupby(['Name','Subject'])
```

[22]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f6f9e0d1390>
```

[23]:
```
data.groupby(['Name','Subject']).groups
```

[23]:
```
{('Gayatri', 'Python'): [3], ('Jitendra', 'Data Science'): [4], ('Jitendra',
'Python'): [0, 5], ('Manas', 'Data Science'): [2], ('Manas', 'Python'): [7],
('Michael', 'Data Science'): [1], ('Michael', 'Python'): [6], ('Sushil', 'Data
Science'): [8]}
```

[24]:
```
#Count of occurence of each name
data.groupby(['Name']).count()
```

[24]:

| Name | Subject | Marks |
|---|---|---|
| Gayatri | 1 | 1 |
| Jitendra | 3 | 3 |
| Manas | 2 | 2 |
| Michael | 2 | 2 |
| Sushil | 1 | 1 |

[25]:
```
data.groupby(['Name','Subject']).count()
```

[25]:

| Name | Subject | Marks |
|---|---|---|
| Gayatri | Python | 1 |
| Jitendra | Data Science | 1 |
| | Python | 2 |
| Manas | Data Science | 1 |
| | Python | 1 |

```
Michael  Data Science      1
         Python            1
Sushil   Data Science      1
```

[26]:
```
data
```

[26]:
```
       Name       Subject  Marks
0  Jitendra        Python      9
1   Michael  Data Science      7
2     Manas  Data Science      8
3   Gayatri        Python      9
4  Jitendra  Data Science      6
5  Jitendra        Python      5
6   Michael        Python      9
7     Manas        Python      8
8    Sushil  Data Science      4
```

[27]:
```
#Sum of each name
data.groupby('Name').sum()
```

[27]:
```
          Marks
Name
Gayatri       9
Jitendra     20
Manas        16
Michael      16
Sushil        4
```

[28]:
```
data.groupby(['Name','Subject']).sum()
```

[28]:
```
                       Marks
Name     Subject
Gayatri  Python            9
Jitendra Data Science      6
         Python           14
Manas    Data Science      8
         Python            8
Michael  Data Science      7
         Python            9
Sushil   Data Science      4
```

[29]:
```
data.groupby(['Subject','Name']).sum()
```

[29]:
```
                       Marks
Subject      Name
Data Science Jitendra      6
             Manas         8
```

```
              Michael       7
              Sushil        4
Python        Gayatri       9
              Jitendra     14
              Manas         8
              Michael       9
```

[30]:
```
#Mean of each name
data.groupby('Name').mean()
```

[30]:
```
              Marks
Name
Gayatri    9.000000
Jitendra   6.666667
Manas      8.000000
Michael    8.000000
Sushil     4.000000
```

[31]:
```
data
```

[31]:
```
       Name       Subject  Marks
0  Jitendra        Python      9
1   Michael  Data Science      7
2     Manas  Data Science      8
3   Gayatri        Python      9
4  Jitendra  Data Science      6
5  Jitendra        Python      5
6   Michael        Python      9
7     Manas        Python      8
8    Sushil  Data Science      4
```

[32]:
```
data_new = pd.DataFrame({'Name':['Ashutosh','Sunil','Ashutosh'],
                         'Subject':['Python','Python','Python'],
                         'Marks':[9,8,8]})
data_new
```

[32]:
```
       Name Subject  Marks
0  Ashutosh  Python      9
1     Sunil  Python      8
2  Ashutosh  Python      8
```

[33]:
```
#Concatenate one below the other
# pd.concat([data,data_new])
pd.concat([data,data_new],ignore_index=True)
```

[33]:
```
       Name       Subject  Marks
0  Jitendra        Python      9
```

```
1    Michael  Data Science      7
2      Manas  Data Science      8
3    Gayatri        Python      9
4   Jitendra  Data Science      6
5   Jitendra        Python      5
6    Michael        Python      9
7      Manas        Python      8
8     Sushil  Data Science      4
9   Ashutosh        Python      9
10     Sunil        Python      8
11  Ashutosh        Python      8
```

[34]:
```
#Concatenate along x axis (horizontally)
pd.concat([data,data_new], axis = 1)
```

[34]:
```
        Name       Subject  Marks      Name Subject  Marks
0   Jitendra        Python      9  Ashutosh  Python    9.0
1    Michael  Data Science      7     Sunil  Python    8.0
2      Manas  Data Science      8  Ashutosh  Python    8.0
3    Gayatri        Python      9       NaN     NaN    NaN
4   Jitendra  Data Science      6       NaN     NaN    NaN
5   Jitendra        Python      5       NaN     NaN    NaN
6    Michael        Python      9       NaN     NaN    NaN
7      Manas        Python      8       NaN     NaN    NaN
8     Sushil  Data Science      4       NaN     NaN    NaN
```

[35]:
```
data_new_2 = pd.DataFrame({'Names':['Ashutosh','Sunil','Ashutosh'],
                           'Subject':['Python','Python','Python'],
                           'Grade':['A','A','B']})
data_new_2
```

[35]:
```
      Names Subject Grade
0  Ashutosh  Python     A
1     Sunil  Python     A
2  Ashutosh  Python     B
```

[36]:
```
pd.concat([data,data_new_2])
```

[36]:
```
        Name       Subject  Marks     Names Grade
0   Jitendra        Python    9.0       NaN   NaN
1    Michael  Data Science    7.0       NaN   NaN
2      Manas  Data Science    8.0       NaN   NaN
3    Gayatri        Python    9.0       NaN   NaN
4   Jitendra  Data Science    6.0       NaN   NaN
5   Jitendra        Python    5.0       NaN   NaN
6    Michael        Python    9.0       NaN   NaN
7      Manas        Python    8.0       NaN   NaN
```

```
8    Sushil  Data Science    4.0       NaN   NaN
0       NaN        Python    NaN  Ashutosh     A
1       NaN        Python    NaN     Sunil     A
2       NaN        Python    NaN  Ashutosh     B
```

### 0.0.2 Pivot Tables

[37]:
```
data
```

[37]:
```
       Name       Subject  Marks
0  Jitendra        Python      9
1   Michael  Data Science      7
2     Manas  Data Science      8
3   Gayatri        Python      9
4  Jitendra  Data Science      6
5  Jitendra        Python      5
6   Michael        Python      9
7     Manas        Python      8
8    Sushil  Data Science      4
```

[44]:
```
pd.pivot_table(data, index = ['Name'], columns = ['Subject'], values = 'Marks',␣
↪fill_value=0, aggfunc=np.min)
```

[44]:
```
Subject   Data Science  Python
Name
Gayatri              0       9
Jitendra             6       5
Manas                8       8
Michael              7       9
Sushil               4       0
```

Scenario 1 - not preserving any table

Whatever is common keep that ( Inner JOIN )

[45]:
```
data = pd.DataFrame({'Name':
↪['Sudhir','Hariprasad','Mahesh','Dhanashree','Sudhir','Sudhir','Hariprasad','Mahesh',
                          'Sushil'],
                     'Subject':['Python','Data Science','Data␣
↪Science','Python','Data Science','Python','Python','Python','Data Science'],
                     'Marks':[9,7,8,9,6,5,9,8,4]})
data
```

[45]:
```
         Name       Subject  Marks
0      Sudhir        Python      9
1  Hariprasad  Data Science      7
```

```
2      Mahesh  Data Science      8
3  Dhanashree        Python      9
4      Sudhir  Data Science      6
5      Sudhir        Python      5
6  Hariprasad        Python      9
7      Mahesh        Python      8
8      Sushil  Data Science      4
```

[50]:
```
data_new_3 = pd.DataFrame({'Name':['Sudhir','Hariprasad','Mahesh','Dhanashree',
                                   'Sushil'],
                           'Groups':[1,2,3,1,4]})
data_new_3
```

[50]:
```
          Name  Groups
0      Sudhir       1
1  Hariprasad       2
2      Mahesh       3
3  Dhanashree       1
4      Sushil       4
```

[51]:
```
pd.merge(data,data_new_3,on='Name')
```

[51]:
```
          Name       Subject  Marks  Groups
0      Sudhir        Python      9       1
1      Sudhir  Data Science      6       1
2      Sudhir        Python      5       1
3  Hariprasad  Data Science      7       2
4  Hariprasad        Python      9       2
5      Mahesh  Data Science      8       3
6      Mahesh        Python      8       3
7  Dhanashree        Python      9       1
8      Sushil  Data Science      4       4
```

Scenario 2 - preserve the content of left table always Left JOIN

[52]:
```
#Removed Sudhir deliberately
data_new_4 = pd.DataFrame({'Name':['Hariprasad','Mahesh','Dhanashree',
                                   'Sushil'],
                           'Groups':[2,3,1,4]})
data_new_4
```

[52]:
```
          Name  Groups
0  Hariprasad       2
1      Mahesh       3
2  Dhanashree       1
3      Sushil       4
```

[54]:
```
#No more occurences of Sudhir
pd.merge(data,data_new_4,on='Name')
```

[54]:

| | Name | Subject | Marks | Groups |
|---|---|---|---|---|
| 0 | Hariprasad | Data Science | 7 | 2 |
| 1 | Hariprasad | Python | 9 | 2 |
| 2 | Mahesh | Data Science | 8 | 3 |
| 3 | Mahesh | Python | 8 | 3 |
| 4 | Dhanashree | Python | 9 | 1 |
| 5 | Sushil | Data Science | 4 | 4 |

[55]:
```
data
```

[55]:

| | Name | Subject | Marks |
|---|---|---|---|
| 0 | Sudhir | Python | 9 |
| 1 | Hariprasad | Data Science | 7 |
| 2 | Mahesh | Data Science | 8 |
| 3 | Dhanashree | Python | 9 |
| 4 | Sudhir | Data Science | 6 |
| 5 | Sudhir | Python | 5 |
| 6 | Hariprasad | Python | 9 |
| 7 | Mahesh | Python | 8 |
| 8 | Sushil | Data Science | 4 |

[56]:
```
#if i want to preserve one of my tables(left table)

pd.merge(data,data_new_4,on='Name',how='left')
```

[56]:

| | Name | Subject | Marks | Groups |
|---|---|---|---|---|
| 0 | Sudhir | Python | 9 | NaN |
| 1 | Hariprasad | Data Science | 7 | 2.0 |
| 2 | Mahesh | Data Science | 8 | 3.0 |
| 3 | Dhanashree | Python | 9 | 1.0 |
| 4 | Sudhir | Data Science | 6 | NaN |
| 5 | Sudhir | Python | 5 | NaN |
| 6 | Hariprasad | Python | 9 | 2.0 |
| 7 | Mahesh | Python | 8 | 3.0 |
| 8 | Sushil | Data Science | 4 | 4.0 |

Scenario 3

Preserve content of right table

Right JOIN

[57]:
```
# data = pd.DataFrame({'Name':
 ↪['Jitendra','Michael','Manas','Gayatri','Jitendra','Jitendra','Michael','Manas'],
```

```
#                       'Subject':['Python','Data Science','Data␣
 ↪Science','Python','Data Science','Python','Python','Python'],
#                       'Marks':[9,7,8,9,6,5,9,8]})
data
```

[57]:

| | Name | Subject | Marks |
|---|---|---|---|
| 0 | Sudhir | Python | 9 |
| 1 | Hariprasad | Data Science | 7 |
| 2 | Mahesh | Data Science | 8 |
| 3 | Dhanashree | Python | 9 |
| 4 | Sudhir | Data Science | 6 |
| 5 | Sudhir | Python | 5 |
| 6 | Hariprasad | Python | 9 |
| 7 | Mahesh | Python | 8 |
| 8 | Sushil | Data Science | 4 |

[61]:
```
#Added Rahil
data_new_4 = pd.DataFrame({'Name':['Hariprasad','Mahesh','Dhanashree',
                                   'Sushil', 'Rahil'],
                           'Groups':[2,3,1,4,5]})
data_new_4
```

[61]:

| | Name | Groups |
|---|---|---|
| 0 | Hariprasad | 2 |
| 1 | Mahesh | 3 |
| 2 | Dhanashree | 1 |
| 3 | Sushil | 4 |
| 4 | Rahil | 5 |

[63]:
```
pd.merge(data,data_new_4,on='Name',how='right')
```

[63]:

| | Name | Subject | Marks | Groups |
|---|---|---|---|---|
| 0 | Hariprasad | Data Science | 7.0 | 2 |
| 1 | Hariprasad | Python | 9.0 | 2 |
| 2 | Mahesh | Data Science | 8.0 | 3 |
| 3 | Mahesh | Python | 8.0 | 3 |
| 4 | Dhanashree | Python | 9.0 | 1 |
| 5 | Sushil | Data Science | 4.0 | 4 |
| 6 | Rahil | NaN | NaN | 5 |

Scenario 4 - Preserve content of both tables

Outer JOIN

[64]:
```
data = pd.DataFrame({'Name':
 ↪['Sudhir','Hariprasad','Mahesh','Dhanashree','Sudhir','Sudhir','Hariprasad','Mahesh',
                                   'Sushil', 'Vishal'],
```

```
                    'Subject':['Python','Data Science','Data␣
↪Science','Python','Data Science','Python','Python','Python','Data␣
↪Science','AI'],
                    'Marks':[9,7,8,9,6,5,9,8,4,9]})
data
```

[64]:

| | Name | Subject | Marks |
|---|---|---|---|
| 0 | Sudhir | Python | 9 |
| 1 | Hariprasad | Data Science | 7 |
| 2 | Mahesh | Data Science | 8 |
| 3 | Dhanashree | Python | 9 |
| 4 | Sudhir | Data Science | 6 |
| 5 | Sudhir | Python | 5 |
| 6 | Hariprasad | Python | 9 |
| 7 | Mahesh | Python | 8 |
| 8 | Sushil | Data Science | 4 |
| 9 | Vishal | AI | 9 |

[65]:
```
#Added Rahil
data_new_4 = pd.DataFrame({'Name':['Hariprasad','Mahesh','Dhanashree',
                           'Sushil', 'Rahil'],
                          'Groups':[2,3,1,4,5]})
data_new_4
```

[65]:

| | Name | Groups |
|---|---|---|
| 0 | Hariprasad | 2 |
| 1 | Mahesh | 3 |
| 2 | Dhanashree | 1 |
| 3 | Sushil | 4 |
| 4 | Rahil | 5 |

[66]:
```
pd.merge(data,data_new_4,on='Name',how='outer')
```

[66]:

| | Name | Subject | Marks | Groups |
|---|---|---|---|---|
| 0 | Sudhir | Python | 9.0 | NaN |
| 1 | Sudhir | Data Science | 6.0 | NaN |
| 2 | Sudhir | Python | 5.0 | NaN |
| 3 | Hariprasad | Data Science | 7.0 | 2.0 |
| 4 | Hariprasad | Python | 9.0 | 2.0 |
| 5 | Mahesh | Data Science | 8.0 | 3.0 |
| 6 | Mahesh | Python | 8.0 | 3.0 |
| 7 | Dhanashree | Python | 9.0 | 1.0 |
| 8 | Sushil | Data Science | 4.0 | 4.0 |
| 9 | Vishal | AI | 9.0 | NaN |
| 10 | Rahil | NaN | NaN | 5.0 |

[68]:
```
pd.concat([data,data_new_4], axis = 1)
```

[68]:

| | Name | Subject | Marks | Name | Groups |
|---|---|---|---|---|---|
| 0 | Sudhir | Python | 9 | Hariprasad | 2.0 |
| 1 | Hariprasad | Data Science | 7 | Mahesh | 3.0 |
| 2 | Mahesh | Data Science | 8 | Dhanashree | 1.0 |
| 3 | Dhanashree | Python | 9 | Sushil | 4.0 |
| 4 | Sudhir | Data Science | 6 | Rahil | 5.0 |
| 5 | Sudhir | Python | 5 | NaN | NaN |
| 6 | Hariprasad | Python | 9 | NaN | NaN |
| 7 | Mahesh | Python | 8 | NaN | NaN |
| 8 | Sushil | Data Science | 4 | NaN | NaN |
| 9 | Vishal | AI | 9 | NaN | NaN |

[ ]: