```
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:30:28) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.

>>> import pandas as pd
>>> df=pd.read_csv('C:/Users/ppp/Desktop/data.csv')
>>> df.dtypes
Age              int64
BMI            float64
Glucose          int64
Insulin        float64
HOMA           float64
Leptin         float64
Adiponectin    float64
Resistin       float64
MCP.1          float64
Classification   int64
dtype: object
>>> df.describe()
          Age         BMI     Glucose     Insulin  ...  Adiponectin    Resistin        MCP.1
Classification
count  116.000000  116.000000  116.000000  116.000000  ...   116.000000  116.000000
116.000000      116.000000
mean    57.301724   27.582111   97.793103   10.012086  ...    10.180874   14.725966
534.647000        1.551724
std     16.112766    5.020136   22.525162   10.067768  ...     6.843341   12.390646  345.912663
0.499475
min     24.000000   18.370000   60.000000    2.432000  ...     1.656020    3.210000   45.843000
1.000000
25%     45.000000   22.973205   85.750000    4.359250  ...     5.474283    6.881763
269.978250        1.000000
50%     56.000000   27.662416   92.000000    5.924500  ...     8.352692   10.827740
471.322500        2.000000
75%     71.000000   31.241442  102.000000   11.189250  ...    11.815970   17.755207
700.085000        2.000000
max     89.000000   38.578759  201.000000   58.460000  ...    38.040000   82.100000
1698.440000        2.000000

[8 rows x 10 columns]

>>> df.groupby(by=['Age']).size()
Age
24   1
25   1
28   1
```

29	2
32	1
34	3
35	2
36	2
38	2
40	2
41	1
42	2
43	3
44	4
45	7
46	3
47	1
48	4
49	5
50	1
51	4
52	1
53	1
54	3
55	1
57	1
58	1
59	2
60	2
61	1
62	2
64	2
65	3
66	5
67	1
68	3
69	5
71	3
72	3
73	2
74	1
75	4
76	4
77	2
78	1
81	1
82	2

```
83   1
85   2
86   3
89   1
dtype: int64
>>> df.isna().sum()
Age             0
BMI             0
Glucose         0
Insulin         0
HOMA            0
Leptin          0
Adiponectin     0
Resistin        0
MCP.1           0
Classification  0
dtype: int64
>>> df.isnull()
      Age    BMI  ... MCP.1  Classification
0    False  False ... False      False
1    False  False ... False      False
2    False  False ... False      False
3    False  False ... False      False
4    False  False ... False      False
..    ...   ...  ...  ...        ...
111  False  False ... False      False
112  False  False ... False      False
113  False  False ... False      False
114  False  False ... False      False
115  False  False ... False      False

[116 rows x 10 columns]
>>> type("Age")
<class 'str'>
>>> type("BMI")
<class 'str'>

>>> df.Age.astype(float)
0    48.0
1    83.0
2    82.0
3    68.0
4    86.0
     ...
```

```
111    45.0
112    62.0
113    65.0
114    72.0
115    86.0
Name: Age, Length: 116, dtype: float64
>>> df.Age=df.Age.astype(float)
>>> df.describe()
           Age        BMI     Glucose     Insulin       HOMA      Leptin  Adiponectin    Resistin
MCP.1  Classification
count  116.000000  116.000000  116.000000  116.000000  116.000000  116.000000
116.000000  116.000000  116.000000      116.000000
mean    57.301724   27.582111   97.793103   10.012086    2.694988   26.615080    10.180874
14.725966  534.647000        1.551724
std     16.112766    5.020136   22.525162   10.067768    3.642043   19.183294     6.843341
12.390646  345.912663        0.499475
min     24.000000   18.370000   60.000000    2.432000    0.467409    4.311000     1.656020
3.210000   45.843000        1.000000
25%     45.000000   22.973205   85.750000    4.359250    0.917966   12.313675     5.474283
6.881763  269.978250        1.000000
50%     56.000000   27.662416   92.000000    5.924500    1.380939   20.271000     8.352692
10.827740  471.322500        2.000000
75%     71.000000   31.241442  102.000000   11.189250    2.857787   37.378300    11.815970
17.755207  700.085000        2.000000
max     89.000000   38.578759  201.000000   58.460000   25.050342   90.280000    38.040000
82.100000 1698.440000        2.000000

>>> df.dtypes
Age            float64
BMI            float64
Glucose          int64
Insulin        float64
HOMA           float64
Leptin         float64
Adiponectin      float64
Resistin       float64
MCP.1          float64
Classification   int64
dtype: object

>>> df.sort_values('Age')
    Age     BMI  Glucose  Insulin  ...  Adiponectin  Resistin    MCP.1  Classification
13   24  18.670000       88    6.107  ...    36.060000   6.85000  632.220               1
12   25  22.860000       82    4.090  ...    23.670000   5.14000  313.730               1
```

```
32   28  35.855815    87   8.576  ...   4.794200  21.44366   358.624           1
22   29  32.270788    84   5.810  ...   6.209635  24.60330   904.981           1
11   29  23.010000    82   5.663  ...  26.720000   4.58000   174.800           1
..   ...      ...     ...  ...       ...       ...       ...

87   85  27.688778   196  51.814  ...   7.901685  55.21530  1078.359            2
4    86  21.111111    92   3.549  ...   4.819240  10.57635   773.920           1
78   86  26.666667   201  41.611  ...   5.357135  24.37010  1698.440            2
115  86  27.180000   138  19.910  ...  14.110000   4.35000    90.090            2
6    89  22.700000    77   4.690  ...   5.589865  12.93610  1256.083           1

[116 rows x 10 columns]
>>> df.sort_values('Age',ascending=False)
     Age      BMI  Glucose  Insulin  ...  Adiponectin  Resistin    MCP.1  Classification
6    89  22.700000    77   4.690  ...   5.589865  12.93610  1256.083           1
115  86  27.180000   138  19.910  ...  14.110000   4.35000    90.090            2
78   86  26.666667   201  41.611  ...   5.357135  24.37010  1698.440            2
4    86  21.111111    92   3.549  ...   4.819240  10.57635   773.920           1
49   85  26.600000    96   4.462  ...   7.931700   9.61350   232.006           1
..   ...      ...     ...  ...       ...       ...       ...

22   29  32.270788    84   5.810  ...   6.209635  24.60330   904.981           1
11   29  23.010000    82   5.663  ...  26.720000   4.58000   174.800           1
32   28  35.855815    87   8.576  ...   4.794200  21.44366   358.624           1
12   25  22.860000    82   4.090  ...  23.670000   5.14000   313.730           1
13   24  18.670000    88   6.107  ...  36.060000   6.85000   632.220           1

[116 rows x 10 columns]
>>> df.rename(columns={'Age':'Year Old'})
     Year Old      BMI  Glucose  Insulin  ...  Adiponectin  Resistin    MCP.1  Classification
0          48  23.500000    70   2.707  ...   9.702400   7.99585  417.114           1
1          83  20.690495    92   3.115  ...   5.429285   4.06405  468.786           1
2          82  23.124670    91   4.498  ...  22.432040   9.27715  554.697           1
3          68  21.367521    77   3.226  ...   7.169560  12.76600  928.220           1
4          86  21.111111    92   3.549  ...   4.819240  10.57635  773.920           1
..        ...      ...     ...  ...       ...       ...       ...

111        45  26.850000    92   3.330  ...  12.100000  10.96000  268.230            2
112        62  26.840000   100   4.530  ...  21.420000   7.32000  330.160            2
113        65  32.050000    97   5.730  ...  22.540000  10.33000  314.050            2
114        72  25.590000    82   2.820  ...  33.750000   3.27000  392.460            2
115        86  27.180000   138  19.910  ...  14.110000   4.35000   90.090            2

[116 rows x 10 columns]
>>> df.sort_index()
     Age      BMI  Glucose  Insulin  ...  Adiponectin  Resistin    MCP.1  Classification
0    48  23.500000    70   2.707  ...   9.702400   7.99585  417.114           1
```

```
1    83 20.690495    92   3.115  ...   5.429285  4.06405 468.786          1
2    82 23.124670    91   4.498  ...  22.432040  9.27715 554.697          1
3    68 21.367521    77   3.226  ...   7.169560 12.76600 928.220          1
4    86 21.111111    92   3.549  ...   4.819240 10.57635 773.920          1
..   ...        ...  ...     ...  ...        ...      ...      ...
111  45 26.850000    92   3.330  ...  12.100000 10.96000 268.230          2
112  62 26.840000   100   4.530  ...  21.420000  7.32000 330.160          2
113  65 32.050000    97   5.730  ...  22.540000 10.33000 314.050          2
114  72 25.590000    82   2.820  ...  33.750000  3.27000 392.460          2
115  86 27.180000   138  19.910  ...  14.110000  4.35000  90.090          2

[116 rows x 10 columns]
>>> df.reset_index()
     index Age    BMI Glucose ... Adiponectin Resistin   MCP.1 Classification
0      0  48 23.500000    70  ...   9.702400  7.99585 417.114             1
1      1  83 20.690495    92  ...   5.429285  4.06405 468.786             1
2      2  82 23.124670    91  ...  22.432040  9.27715 554.697             1
3      3  68 21.367521    77  ...   7.169560 12.76600 928.220             1
4      4  86 21.111111    92  ...   4.819240 10.57635 773.920             1
..   ... ...        ...   ...  ...        ...      ...     ...
111  111  45 26.850000    92  ...  12.100000 10.96000 268.230             2
112  112  62 26.840000   100 ...  21.420000  7.32000 330.160             2
113  113  65 32.050000    97  ...  22.540000 10.33000 314.050             2
114  114  72 25.590000    82  ...  33.750000  3.27000 392.460             2
115  115  86 27.180000   138 ...  14.110000  4.35000  90.090             2

[116 rows x 11 columns]
>>> df.drop(columns=['Age'])
         BMI Glucose Insulin    HOMA ... Adiponectin Resistin   MCP.1 Classification
0   23.500000    70   2.707 0.467409 ...   9.702400  7.99585 417.114             1
1   20.690495    92   3.115 0.706897 ...   5.429285  4.06405 468.786             1
2   23.124670    91   4.498 1.009651 ...  22.432040  9.27715 554.697             1
3   21.367521    77   3.226 0.612725 ...   7.169560 12.76600 928.220             1
4   21.111111    92   3.549 0.805386 ...   4.819240 10.57635 773.920             1
..        ...   ...     ...      ... ...        ...      ...     ...
111 26.850000    92   3.330 0.755688 ...  12.100000 10.96000 268.230             2
112 26.840000   100   4.530 1.117400 ...  21.420000  7.32000 330.160             2
113 32.050000    97   5.730 1.370998 ...  22.540000 10.33000 314.050             2
114 25.590000    82   2.820 0.570392 ...  33.750000  3.27000 392.460             2
115 27.180000   138  19.910 6.777364 ...  14.110000  4.35000  90.090             2

[116 rows x 9 columns]
>>> pd.melt(df)
        variable  value
```

```
0          Age    48.0
1          Age    83.0
2          Age    82.0
3          Age    68.0
4          Age    86.0
...        ...    ...
1155  Classification    2.0
1156  Classification    2.0
1157  Classification    2.0
1158  Classification    2.0
1159  Classification    2.0

[1160 rows x 2 columns]
>>> df.drop_duplicates()
     Age       BMI  Glucose  Insulin  ...  Adiponectin  Resistin    MCP.1  Classification
0     48  23.500000       70    2.707  ...     9.702400   7.99585  417.114               1
1     83  20.690495       92    3.115  ...     5.429285   4.06405  468.786               1
2     82  23.124670       91    4.498  ...    22.432040   9.27715  554.697               1
3     68  21.367521       77    3.226  ...     7.169560  12.76600  928.220               1
4     86  21.111111       92    3.549  ...     4.819240  10.57635  773.920               1
..   ...        ...      ...      ...  ...          ...       ...      ...             ...
111   45  26.850000       92    3.330  ...    12.100000  10.96000  268.230               2
112   62  26.840000      100    4.530  ...    21.420000   7.32000  330.160               2
113   65  32.050000       97    5.730  ...    22.540000  10.33000  314.050               2
114   72  25.590000       82    2.820  ...    33.750000   3.27000  392.460               2
115   86  27.180000      138   19.910  ...    14.110000   4.35000   90.090               2

[116 rows x 10 columns]
>>> df.head(5)
   Age       BMI  Glucose  Insulin  ...  Adiponectin  Resistin    MCP.1  Classification
0   48  23.500000       70    2.707  ...     9.702400   7.99585  417.114               1
1   83  20.690495       92    3.115  ...     5.429285   4.06405  468.786               1
2   82  23.124670       91    4.498  ...    22.432040   9.27715  554.697               1
3   68  21.367521       77    3.226  ...     7.169560  12.76600  928.220               1
4   86  21.111111       92    3.549  ...     4.819240  10.57635  773.920               1

[5 rows x 10 columns]
>>> df.tail(5)
     Age    BMI  Glucose  Insulin  ...  Adiponectin  Resistin   MCP.1  Classification
111   45  26.85       92     3.33  ...        12.10     10.96  268.23               2
112   62  26.84      100     4.53  ...        21.42      7.32  330.16               2
113   65  32.05       97     5.73  ...        22.54     10.33  314.05               2
114   72  25.59       82     2.82  ...        33.75      3.27  392.46               2
115   86  27.18      138    19.91  ...        14.11      4.35   90.09               2
```

[5 rows x 10 columns]

```
Age             6647.000000
BMI             3199.524856
Glucose        11344.000000
Insulin         1161.402000
HOMA             312.618557
Leptin          3087.349300
Adiponectin     1180.981380
Resistin        1708.212040
MCP.1          62019.052000
Classification   180.000000
dtype: float64
```

```
>>> df.count()
Age             116
BMI             116
Glucose         116
Insulin         116
HOMA            116
Leptin          116
Adiponectin     116
Resistin        116
MCP.1           116
Classification  116
dtype: int64
>>> df.median()
Age              56.000000
BMI              27.662416
Glucose          92.000000
Insulin           5.924500
HOMA              1.380939
Leptin           20.271000
Adiponectin       8.352692
Resistin         10.827740
MCP.1           471.322500
Classification    2.000000
dtype: float64
>>> df.quantile()
Age              56.000000
BMI              27.662416
Glucose          92.000000
Insulin           5.924500
```

```
HOMA              1.380939
Leptin          20.271000
Adiponectin      8.352692
Resistin        10.827740
MCP.1          471.322500
Classification   2.000000
Name: 0.5, dtype: float64

>>> df.min()
Age             24.000000
BMI             18.370000
Glucose         60.000000
Insulin          2.432000
HOMA             0.467409
Leptin           4.311000
Adiponectin      1.656020
Resistin         3.210000
MCP.1           45.843000
Classification   1.000000
dtype: float64
>>> df.max()
Age             89.000000
BMI             38.578759
Glucose        201.000000
Insulin         58.460000
HOMA            25.050342
Leptin          90.280000
Adiponectin     38.040000
Resistin        82.100000
MCP.1         1698.440000
Classification   2.000000
dtype: float64
>>> df.mean()
Age             57.301724
BMI             27.582111
Glucose         97.793103
Insulin         10.012086
HOMA             2.694988
Leptin          26.615080
Adiponectin     10.180874
Resistin        14.725966
MCP.1          534.647000
Classification   1.551724
dtype: float64
```

```
>>> df.std()
Age             16.112766
BMI              5.020136
Glucose         22.525162
Insulin         10.067768
HOMA             3.642043
Leptin          19.183294
Adiponectin      6.843341
Resistin        12.390646
MCP.1          345.912663
Classification   0.499475
dtype: float64

>>> pd.get_dummies(df['Age'])
     24 25 28 29 32 34 35 36 38 40 41 42 43 44 45 46 47 48 49 50 ... 64 65 66 67
68 69 71 72 73 74 75 76 77 78 81 82 83 85 86 89
0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 ...  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0
1     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  1  0  0  0
2     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  1  0  0  0  0
3     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0
4     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0
..   .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ... .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. ..
111   0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0 ...  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0
112   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0
113   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  1  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0
114   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  0  0  0  0  0  0  1
0  0  0  0  0  0  0  0  0  0  0  0
115   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ...  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0

[116 rows x 51 columns]
>>> pd.get_dummies(df['Glucose'])
     60 70 74 75 76 77 78 79  ... 131 134 138 139 152 196 199 201
0     0  1  0  0  0  0  0  0 ...   0   0   0   0   0   0   0   0
1     0  0  0  0  0  0  0  0 ...   0   0   0   0   0   0   0   0
```

```
2     0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0
3     0  0  0  0  0  1  0  0  ...  0  0  0  0  0  0  0  0
4     0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0
..   ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
111   0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0
112   0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0
113   0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0
114   0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0
115   0  0  0  0  0  0  0  0  ...  0  0  1  0  0  0  0  0

[116 rows x 50 columns]

>>> df.iloc[2:50]
   Age     BMI  Glucose  Insulin    HOMA    Leptin  Adiponectin  Resistin    MCP.1
Classification
2   82  23.124670     91   4.498  1.009651  17.9393   22.432040   9.27715   554.697
1
3   68  21.367521     77   3.226  0.612725   9.8827    7.169560  12.76600   928.220
1
4   86  21.111111     92   3.549  0.805386   6.6994    4.819240  10.57635   773.920
1
5   49  22.854458     92   3.226  0.732087   6.8317   13.679750  10.31760   530.410
1
6   89  22.700000     77   4.690  0.890787   6.9640    5.589865  12.93610  1256.083
1
7   76  23.800000    118   6.470  1.883201   4.3110   13.251320   5.10420   280.694
1
8   73  22.000000     97   3.350  0.801543   4.4700   10.358725   6.28445   136.855
1
9   75  23.000000     83   4.952  1.013839  17.1270   11.578990   7.09130   318.302
1
10  34  21.470000     78   3.469  0.667436  14.5700   13.110000   6.92000   354.600
1
11  29  23.010000     82   5.663  1.145436  35.5900   26.720000   4.58000   174.800
1
12  25  22.860000     82   4.090  0.827271  20.4500   23.670000   5.14000   313.730
1
13  24  18.670000     88   6.107  1.330000   8.8800   36.060000   6.85000   632.220
1
14  38  23.340000     75   5.782  1.069670  15.2600   17.950000   9.35000   165.020
1
15  44  20.760000     86   7.553  1.600000  14.0900   20.320000   7.64000    63.610
1
16  47  22.030000     84   2.869  0.590000  26.6500   38.040000   3.32000   191.720
```

1

17	61	32.038959	85	18.077	3.790144	30.7729	7.780255	13.68392	444.395

1

| 18 | 64 | 34.529723 | 95 | 4.427 | 1.037394 | 21.2117 | 5.462620 | 6.70188 | 252.449 |

1

| 19 | 32 | 36.512637 | 87 | 14.026 | 3.009980 | 49.3727 | 5.100000 | 17.10223 | 588.460 |

1

| 20 | 36 | 28.576676 | 86 | 4.345 | 0.921719 | 15.1248 | 8.600000 | 9.15390 | 534.224 |

1

| 21 | 34 | 31.975015 | 87 | 4.530 | 0.972138 | 28.7502 | 7.642760 | 5.62592 | 572.783 |

1

| 22 | 29 | 32.270788 | 84 | 5.810 | 1.203832 | 45.6196 | 6.209635 | 24.60330 | 904.981 |

1

| 23 | 35 | 30.276817 | 84 | 4.376 | 0.906707 | 39.2134 | 9.048185 | 16.43706 | 733.797 |

1

| 24 | 54 | 30.483158 | 90 | 5.537 | 1.229214 | 12.3310 | 9.731380 | 10.19299 | 1227.910 |

1

| 25 | 45 | 37.035608 | 83 | 6.760 | 1.383997 | 39.9802 | 4.617125 | 8.70448 | 586.173 |

1

| 26 | 50 | 38.578759 | 106 | 6.703 | 1.752611 | 46.6401 | 4.667645 | 11.78388 | 887.160 |

1

| 27 | 66 | 31.446541 | 90 | 9.245 | 2.052390 | 45.9624 | 10.355260 | 23.38190 | 1102.110 |

1

| 28 | 35 | 35.250761 | 90 | 6.817 | 1.513374 | 50.6094 | 6.966895 | 22.03703 | 667.928 |

1

| 29 | 36 | 34.174890 | 80 | 6.590 | 1.300427 | 10.2809 | 5.065915 | 15.72187 | 581.313 |

1

| 30 | 66 | 36.212279 | 101 | 15.533 | 3.869788 | 74.7069 | 7.539550 | 22.32024 | 864.968 |

1

| 31 | 53 | 36.790166 | 101 | 10.175 | 2.534932 | 27.1841 | 20.030000 | 10.26309 | 695.754 |

1

| 32 | 28 | 35.855815 | 87 | 8.576 | 1.840410 | 68.5102 | 4.794200 | 21.44366 | 358.624 |

1

| 33 | 43 | 34.422174 | 89 | 23.194 | 5.091856 | 31.2128 | 8.300955 | 6.71026 | 960.246 |

1

| 34 | 51 | 27.688778 | 77 | 3.855 | 0.732193 | 20.0920 | 3.192090 | 10.37518 | 473.859 |

1

| 35 | 67 | 29.606767 | 79 | 5.819 | 1.133929 | 21.9033 | 2.194280 | 4.20750 | 585.307 |

1

| 36 | 66 | 31.238590 | 82 | 4.181 | 0.845677 | 16.2247 | 4.267105 | 3.29175 | 634.602 |

1

| 37 | 69 | 35.092702 | 101 | 5.646 | 1.406607 | 83.4821 | 6.796985 | 82.10000 | 263.499 |

1

| 38 | 60 | 26.349292 | 103 | 5.138 | 1.305395 | 24.2998 | 2.194280 | 20.25350 | 378.996 |

```
1
39  77  35.587929      76   3.881  0.727558  21.7863    8.125550  17.26150  618.272
1
40  76  29.218408      83   5.376  1.100646  28.5620    7.369960  8.04375  698.789
1
41  76  27.200000      94  14.070  3.262364  35.8910    9.346630  8.41560  377.227
1
42  75  27.300000      85   5.197  1.089638  10.3900    9.000805  7.57670  335.393
1
43  69  32.500000      93   5.430  1.245642  15.1450   11.787960  11.78796  270.142
1
44  71  30.300000     102   8.340  2.098344  56.5020    8.130000  4.29890  200.976
1
45  66  27.700000      90   6.042  1.341324  24.8460    7.652055  6.70520  225.880
1
46  75  25.700000      94   8.079  1.873251  65.9260    3.741220  4.49685  206.802
1
47  78  25.300000      60   3.508  0.519184   6.6330   10.567295  4.66380  209.749
1
48  69  29.400000      89  10.704  2.349885  45.2720    8.286300  4.53000  215.769
1
49  85  26.600000      96   4.462  1.056602   7.8500    7.931700  9.61350  232.006
1
>>>
>>> print(pd.get_dummies(df['Glucose']))
    60  70  74  75  76  77  78  79  80  82  83  84  85  86  \
0    0   1   0   0   0   0   0   0   0   0   0   0   0   0
1    0   0   0   0   0   0   0   0   0   0   0   0   0   0
2    0   0   0   0   0   0   0   0   0   0   0   0   0   0
3    0   0   0   0   0   1   0   0   0   0   0   0   0   0
4    0   0   0   0   0   0   0   0   0   0   0   0   0   0
..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..
111  0   0   0   0   0   0   0   0   0   0   0   0   0   0
112  0   0   0   0   0   0   0   0   0   0   0   0   0   0
113  0   0   0   0   0   0   0   0   0   0   0   0   0   0
114  0   0   0   0   0   0   0   0   0   1   0   0   0   0
115  0   0   0   0   0   0   0   0   0   0   0   0   0   0

    87  88  89  90  91  92  93  94  95  96  97  98  99  100  \
0    0   0   0   0   0   0   0   0   0   0   0   0   0   0
1    0   0   0   0   0   1   0   0   0   0   0   0   0   0
2    0   0   0   0   1   0   0   0   0   0   0   0   0   0
3    0   0   0   0   0   0   0   0   0   0   0   0   0   0
4    0   0   0   0   0   1   0   0   0   0   0   0   0   0
```

```
..   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
111   0    0    0    0    0    1    0    0    0    0    0    0    0    0
112   0    0    0    0    0    0    0    0    0    0    0    0    0    1
113   0    0    0    0    0    0    0    0    0    0    1    0    0    0
114   0    0    0    0    0    0    0    0    0    0    0    0    0    0
115   0    0    0    0    0    0    0    0    0    0    0    0    0    0

     101  102  103  104  105  106  108  110  112  114  116  118  119  128  \
0     0    0    0    0    0    0    0    0    0    0    0    0    0    0
1     0    0    0    0    0    0    0    0    0    0    0    0    0    0
2     0    0    0    0    0    0    0    0    0    0    0    0    0    0
3     0    0    0    0    0    0    0    0    0    0    0    0    0    0
4     0    0    0    0    0    0    0    0    0    0    0    0    0    0
..   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
111   0    0    0    0    0    0    0    0    0    0    0    0    0    0
112   0    0    0    0    0    0    0    0    0    0    0    0    0    0
113   0    0    0    0    0    0    0    0    0    0    0    0    0    0
114   0    0    0    0    0    0    0    0    0    0    0    0    0    0
115   0    0    0    0    0    0    0    0    0    0    0    0    0    0

     131  134  138  139  152  196  199  201
0     0    0    0    0    0    0    0    0
1     0    0    0    0    0    0    0    0
2     0    0    0    0    0    0    0    0
3     0    0    0    0    0    0    0    0
4     0    0    0    0    0    0    0    0
..   ...  ...  ...  ...  ...  ...  ...  ...
111   0    0    0    0    0    0    0    0
112   0    0    0    0    0    0    0    0
113   0    0    0    0    0    0    0    0
114   0    0    0    0    0    0    0    0
115   0    0    1    0    0    0    0    0

[116 rows x 50 columns]
>>>
```