import pandas as pd

df=pd.read\_csv("data.csv")

df.dtypes

Age int64

BMI float64

Glucose int64

Insulin float64

HOMA float64

Leptin float64

Adiponectin float64

Resistin float64

MCP.1 float64 Classification object dtype: object

df.describe()

**Age**

**BMI**

**Glucose**

**Insulin**

**HOMA**

**Leptin**

**Adiponectin**

**Resistin**

**MCP.1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 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 |
| **std** | 16.112766 | 5.020136 | 22.525162 | 10.067768 | 3.642043 | 19.183294 | 6.843341 | 12.390646 | 345.912663 |
| **min** | 24.000000 | 18.370000 | 60.000000 | 2.432000 | 0.467409 | 4.311000 | 1.656020 | 3.210000 | 45.843000 |
| **25%** | 45.000000 | 22.973205 | 85.750000 | 4.359250 | 0.917966 | 12.313675 | 5.474283 | 6.881763 | 269.978250 |
| **50%** | 56.000000 | 27.662416 | 92.000000 | 5.924500 | 1.380939 | 20.271000 | 8.352692 | 10.827740 | 471.322500 |
| **75%** | 71.000000 | 31.241442 | 102.000000 | 11.189250 | 2.857787 | 37.378300 | 11.815970 | 17.755207 | 700.085000 |
| **max** | 89.000000 | 38.578759 | 201.000000 | 58.460000 | 25.050342 | 90.280000 | 38.040000 | 82.100000 | 1698.440000 |
| ** |  |  |  |  |  |  |  |  | ** |

df.groupby(by=['Age']).size()

Age

1. 1
2. 1
3. 1
4. 2

32 1

1. 3
2. 2
3. 2

38 2

1. 2
2. 1
3. 2
4. 3
5. 4
6. 7
7. 3
8. 1
9. 4
10. 5
11. 1
12. 4
13. 1
14. 1
15. 3
16. 1
17. 1
18. 1
19. 2
20. 2
21. 1
22. 2
23. 2
24. 3
25. 5
26. 1
27. 3
28. 5
29. 3
30. 3
31. 2
32. 1
33. 4
34. 4
35. 2
36. 1
37. 1
38. 2
39. 1
40. 2
41. 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 Glucose Insulin HOMA Leptin Adiponectin Resistin MCP.1 Classification**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | False | False | False | False | False | False | False | False | False | False |
| **1** | False | False | False | False | False | False | False | False | False | False |
| **2** | False | False | False | False | False | False | False | False | False | False |
| **3** | False | False | False | False | False | False | False | False | False | False |
| **4** | False | False | False | False | False | False | False | False | False | False |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **111** | False | False | False | False | False | False | False | False | False | False |
| **112** | False | False | False | False | False | False | False | False | False | False |
| **113** | False | False | False | False | False | False | False | False | False | False |
| **114** | False | False | False | False | False | False | False | False | False | False |
| 1. False False 2. rows × 10 columns | | | False | False | False | False | False | False | False | False |
| ** | | |  |  |  |  |  |  |  | ** |

type("Age") str

type("BMI") str

df.Age.astype(float)

0 48.0

1. 83.0
2. 82.0
3. 68.0
4. 86.0 ... 111 45.0
5. 62.0
6. 65.0
7. 72.0
8. 86.0

Name: Age, Length: 116, dtype: float64

df.Age=df.Age.astype(float)

df.describe()

df.dtypes

Age float64

BMI float64

Glucose int64

Insulin float64

HOMA float64

Leptin float64

Adiponectin float64

Resistin float64

MCP.1 float64 Classification object dtype: object

df.sort\_values('Age')

df.sort\_values('Age',ascending=False)

df.rename(columns={'Age':'Year Old'})

df.sort\_index()

df.reset\_index()

df.drop(columns=['Age'])

pd.melt(df)

df.drop\_duplicates()

df.head(5)

df.tail(5)

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

for col in df.columns: print(f"Unique values in {col}: {df[col].unique()}")

# Convert non-numeric values to NaN for the 'Classification' column df['Classification'] = pd.to\_numeric(df['Classification'], errors='coerce')

# Calculate the median for numeric columns only df.median()

Unique values in Age: [48. 83. 82. 68. 86. 49. 89. 76. 73. 75. 34. 29. 25. 24. 38. 44. 47. 61.

**

64. 32. 36. 35. 54. 45. 50. 66. 53. 28. 43. 51. 67. 69. 60. 77. 71. 78.

85. 42. 62. 59. 46. 72. 55. 41. 81. 65. 58. 40. 52. 74. 57.] Unique values in BMI: [23.5 20.69049454 23.12467037 21.36752137 21.11111111 22.85445769

22.7 23.8 22. 23. 21.47 23.01

22.86 18.67 23.34 20.76 22.03 32.03895937

34.5297228 36.51263743 28.57667585 31.97501487 32.27078777 30.27681661 **

30.48315806 37.03560819 38.57875854 31.44654088 35.2507611 34.17489

36.21227888 36.7901662 35.85581466 34.42217362 27.68877813 29.60676726

31.2385898 35.09270153 26.34929208 35.58792924 29.2184076 27.2

27.3 32.5 30.3 27.7 25.7 25.3

29.4 26.6 27.1 25.9 21.30394858 20.82999519

20.9566075 24.24242424 21.35991456 21.08281329 19.13265306 22.65625

22.4996371 21.51385851 22.89281998 22.83287935 23.14049587 24.21875

22.22222222 20.83 19.56 20.26 24.74 18.37

23.62 22.21 26.5625 31.25 26.66666667 26.6727633

28.67262608 31.64036818 32.46191136 25.51020408 29.296875 29.666548

28.125 29.15451895 30.83653053 31.21748179 30.8012487 31.23140988

29.77777778 27.88761707 27.63605442 27.91551882 28.44444444 28.65013774

30.91557669 29.13631634 34.83814777 37.109375 29.38475666 33.18

35.56 30.48 36.05 26.85 26.84 32.05

25.59 27.18 ] Unique values in Glucose: [ 70 92 91 77 118 97 83 78 82 88 75 86 84 85 95 87 90 106

80 101 89 79 103 76 94 93 102 60 96 110 74 112 98 116 114 105

201 100 99 196 199 139 128 134 131 104 108 152 119 138]

Unique values in Insulin: [ 2.707 3.115 4.498 3.226 3.549 4.69 6.47 3.35 4.952 3.469

5.663 4.09 6.107 5.782 7.553 2.869 18.077 4.427 14.026 4.345 4.53 5.81 4.376 5.537 6.76 6.703 9.245 6.817 6.59 15.533 10.175 8.576 23.194 3.855 5.819 4.181 5.646 5.138 3.881 5.376 14.07 5.197 5.43 8.34 6.042 8.079 3.508 10.704 4.462 26.211

4.58 13.852 4.56 12.305 21.699 2.999 6.2 4.364 3.482 5.261 6.683 2.64 2.74 6.862 4.902 3.73 5.7 3.42 15.89 3.44

58.46 6.03 4.42 36.94 10.555 16.635 4.328 41.611 22.033 3.188 9.669 28.677 10.395 4.172 14.649 2.54 51.814 12.162 16.582 41.894

30.212 24.887 30.13 8.396 9.208 2.432 18.2 8.808 3.012 6.524

10.491 10.949 12.548 5.636 4.713 5.75 8.15 7.01 11.91 3.33

5.73 2.82 19.91 ] Unique values in HOMA: [ 0.46740867 0.70689733 1.00965107 0.61272493 0.8053864 0.73208693

0.89078733 1.88320133 0.80154333 1.01383947 0.6674356 1.14543613

0.82727067 1.33 1.06967 1.6 0.59 3.79014433 1.03739367 3.0099796 0.92171933 0.972138 1.203832 0.9067072

1.229214 1.38399733 1.75261107 2.05239 1.513374 1.30042667

3.86978807 2.53493167 1.8404096 5.09185613 0.732193 1.13392913

0.84567693 1.4066068 1.30539453 0.72755813 1.1006464 3.262364 1.08963767 1.245642 2.098344 1.341324 1.8732508 0.519184

2.3498848 1.0566016 7.111918 0.96027333 3.4851632 0.832352

2.85311933 4.9242264 0.6879706 1.55992 1.0011016 0.79018187 1.23282767 1.84629013 0.507936 0.69614267 1.65877413 1.4026256

0.79125733 1.37788 0.742368 4.468268 0.78065067 15.28534133

1.56177 1.14478 7.83620533 2.62960233 3.775036 1.09960053 20.6307338 5.27176247 0.60550747 2.38502 7.0029234 2.871792

1.00851147 3.071407 0.56388 25.05034187 5.9699204 5.68541507 13.22733227 4.45899333 6.4834952 8.22598307 9.73600733 1.44970933 2.2485936 0.61789013 4.66890667 2.3464512 0.6538048 1.43223547

2.5101466 2.24162527 2.94041467 1.86288587 1.046286 1.30486667

2.63353667 2.62828267 3.495982 0.755688 1.1174 1.370998

0.570392 6.777364 ]

Unique values in Leptin: [ 8.8071 8.8438 17.9393 9.8827 6.6994 6.8317 6.964 4.311 4.47

df.quantile

<bound method DataFrame.quantile of Age BMI Glucose Insulin HOMA Leptin Adiponectin \

1. 48.0 23.500000 70 2.707 0.467409 8.8071 9.702400
2. 83.0 20.690495 92 3.115 0.706897 8.8438 5.429285 **
3. 82.0 23.124670 91 4.498 1.009651 17.9393 22.432040
4. 68.0 21.367521 77 3.226 0.612725 9.8827 7.169560
5. 86.0 21.111111 92 3.549 0.805386 6.6994 4.819240 .. ... ... ... ... ... ... ...
6. 45.0 26.850000 92 3.330 0.755688 54.6800 12.100000
7. 62.0 26.840000 100 4.530 1.117400 12.4500 21.420000
8. 65.0 32.050000 97 5.730 1.370998 61.4800 22.540000
9. 72.0 25.590000 82 2.820 0.570392 24.9600 33.750000
10. 86.0 27.180000 138 19.910 6.777364 90.2800 14.110000

Resistin MCP.1 Classification 0 7.99585 417.114 1.0

1. 4.06405 468.786 1.0
2. 9.27715 554.697 1.0
3. 12.76600 928.220 1.0
4. 10.57635 773.920 1.0 .. ... ... ...
5. 10.96000 268.230 2.0
6. 7.32000 330.160 2.0
7. 10.33000 314.050 2.0
8. 3.27000 392.460 2.0
9. 4.35000 90.090 2.0

[116 rows x 10 columns]>

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

for col in df.columns: print(f"Unique values in {col}: {df[col].unique()}") df.mean()

Unique values in Age: [48. 83. 82. 68. 86. 49. 89. 76. 73. 75. 34. 29. 25. 24. 38. 44. 47. 61.

**

**

64. 32. 36. 35. 54. 45. 50. 66. 53. 28. 43. 51. 67. 69. 60. 77. 71. 78.

85. 42. 62. 59. 46. 72. 55. 41. 81. 65. 58. 40. 52. 74. 57.] Unique values in BMI: [23.5 20.69049454 23.12467037 21.36752137 21.11111111 22.85445769

22.7 23.8 22. 23. 21.47 23.01

22.86 18.67 23.34 20.76 22.03 32.03895937

34.5297228 36.51263743 28.57667585 31.97501487 32.27078777 30.27681661 30.48315806 37.03560819 38.57875854 31.44654088 35.2507611 34.17489

36.21227888 36.7901662 35.85581466 34.42217362 27.68877813 29.60676726

31.2385898 35.09270153 26.34929208 35.58792924 29.2184076 27.2

27.3 32.5 30.3 27.7 25.7 25.3

29.4 26.6 27.1 25.9 21.30394858 20.82999519

20.9566075 24.24242424 21.35991456 21.08281329 19.13265306 22.65625

22.4996371 21.51385851 22.89281998 22.83287935 23.14049587 24.21875

22.22222222 20.83 19.56 20.26 24.74 18.37

23.62 22.21 26.5625 31.25 26.66666667 26.6727633

28.67262608 31.64036818 32.46191136 25.51020408 29.296875 29.666548

28.125 29.15451895 30.83653053 31.21748179 30.8012487 31.23140988

29.77777778 27.88761707 27.63605442 27.91551882 28.44444444 28.65013774

30.91557669 29.13631634 34.83814777 37.109375 29.38475666 33.18

35.56 30.48 36.05 26.85 26.84 32.05

25.59 27.18 ] Unique values in Glucose: [ 70 92 91 77 118 97 83 78 82 88 75 86 84 85 95 87 90 106

80 101 89 79 103 76 94 93 102 60 96 110 74 112 98 116 114 105

201 100 99 196 199 139 128 134 131 104 108 152 119 138]

Unique values in Insulin: [ 2.707 3.115 4.498 3.226 3.549 4.69 6.47 3.35 4.952 3.469

5.663 4.09 6.107 5.782 7.553 2.869 18.077 4.427 14.026 4.345 4.53 5.81 4.376 5.537 6.76 6.703 9.245 6.817 6.59 15.533 10.175 8.576 23.194 3.855 5.819 4.181 5.646 5.138 3.881 5.376 14.07 5.197 5.43 8.34 6.042 8.079 3.508 10.704 4.462 26.211

4.58 13.852 4.56 12.305 21.699 2.999 6.2 4.364 3.482 5.261 6.683 2.64 2.74 6.862 4.902 3.73 5.7 3.42 15.89 3.44

58.46 6.03 4.42 36.94 10.555 16.635 4.328 41.611 22.033 3.188 9.669 28.677 10.395 4.172 14.649 2.54 51.814 12.162 16.582 41.894

30.212 24.887 30.13 8.396 9.208 2.432 18.2 8.808 3.012 6.524

10.491 10.949 12.548 5.636 4.713 5.75 8.15 7.01 11.91 3.33

5.73 2.82 19.91 ] Unique values in HOMA: [ 0.46740867 0.70689733 1.00965107 0.61272493 0.8053864 0.73208693

0.89078733 1.88320133 0.80154333 1.01383947 0.6674356 1.14543613

0.82727067 1.33 1.06967 1.6 0.59 3.79014433 1.03739367 3.0099796 0.92171933 0.972138 1.203832 0.9067072

1.229214 1.38399733 1.75261107 2.05239 1.513374 1.30042667

3.86978807 2.53493167 1.8404096 5.09185613 0.732193 1.13392913

0.84567693 1.4066068 1.30539453 0.72755813 1.1006464 3.262364 1.08963767 1.245642 2.098344 1.341324 1.8732508 0.519184

2.3498848 1.0566016 7.111918 0.96027333 3.4851632 0.832352

2.85311933 4.9242264 0.6879706 1.55992 1.0011016 0.79018187 1.23282767 1.84629013 0.507936 0.69614267 1.65877413 1.4026256

0.79125733 1.37788 0.742368 4.468268 0.78065067 15.28534133

1.56177 1.14478 7.83620533 2.62960233 3.775036 1.09960053 20.6307338 5.27176247 0.60550747 2.38502 7.0029234 2.871792

1.00851147 3.071407 0.56388 25.05034187 5.9699204 5.68541507 13.22733227 4.45899333 6.4834952 8.22598307 9.73600733 1.44970933 2.2485936 0.61789013 4.66890667 2.3464512 0.6538048 1.43223547

2.5101466 2.24162527 2.94041467 1.86288587 1.046286 1.30486667

2.63353667 2.62828267 3.495982 0.755688 1.1174 1.370998

0.570392 6.777364 ] Unique values in Leptin: [ 8.8071 8.8438 17.9393 9.8827 6.6994 6.8317 6.964 4.311 4.47 # Convert all numeric columns df = df.apply(pd.to\_numeric, errors='coerce') 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 0.498406 dtype: float64

pd.get\_dummies(df['Age'])

pd.get\_dummies(df['Glucose'])

df.iloc[2:50]

print(pd.get\_dummies(df['Glucose']))

60 70 74 75 76 77 78 79 80 82 \

1. False True False False False False False False False False
2. False False False False False False False False False False
3. False False False False False False False False False False
4. False False False False False True False False False False
5. False False False False False False False False False False .. ... ... ... ... ... ... ... ... ... ...
6. False False False False False False False False False False
7. False False False False False False False False False False
8. False False False False False False False False False False
9. False False False False False False False False False True
10. False False False False False False False False False False

... 119 128 131 134 138 139 152 196 199 201

1. ... False False False False False False False False False False
2. ... False False False False False False False False False False
3. ... False False False False False False False False False False
4. ... False False False False False False False False False False
5. ... False False False False False False False False False False .. ... ... ... ... ... ... ... ... ... ... ...
6. ... False False False False False False False False False False
7. ... False False False False False False False False False False
8. ... False False False False False False False False False False
9. ... False False False False False False False False False False
10. ... False False False False True False False False False False

[116 rows x 50 columns]