Data Science and Machine Learning Internship
(HDLC Technologies)

Name: *B. Deva SriRama Sai Ganesh*
Email: bandarusaiganesh2003@gmail.com

WEEK - 3 Assignment

Question:

Read the given CSV file (insurance.csv) and write a code to perform the following operations on the data:

1. Remove the rows which is having NULL Values.
2. Fill the Empty cells in the “charges” column with
   a) Mean
   b) Median
   c) Mode
3. Replace the values which is more than 60 in the “age” column with the value of 50.
4. Find out the duplicates in the dataset and remove.
5. Find out the correlation between the data.
6. Draw a scatter plot between “age” and “bmi”
7. Draw a histogram plot on “age” column.

Solutions:

insurance.csv

```
In [28]: import pandas as obj
         Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")
         print(Insurance_data.to_string())
```

	age	sex	bmi	children	smoker	region	charges	insuranceclaim
0	19	0	27.900	0	1	3	16884.92400	1
1	18	1	33.770	1	0	2	1725.55230	1
2	28	1	33.000	3	0	2	4449.46200	0
3	33	1	22.705	0	0	1	21984.47061	0
4	32	1	28.880	0	0	1	3866.85520	1
5	31	0	25.740	0	0	2	3756.62160	0
6	46	0	33.440	1	0	2	NaN	1
7	37	0	27.740	3	0	1	7281.50560	0
8	37	0	27.740	3	0	1	7281.50560	0
9	37	1	29.830	2	0	0	6406.41070	0
10	600	0	25.840	0	0	1	28923.13692	0

11	25	1	26.220	0	0	0	2721.32080	1
12	62	0	26.290	0	1	2	27808.72510	1
13	23	1	34.400	0	0	3	1826.84300	1
14	56	0	39.820	0	0	2	11090.71780	1
15	27	1	42.130	0	1	2	39611.75770	1
16	19	1	24.600	1	0	3	1837.23700	0
17	520	0	30.780	1	0	0	10797.33620	1
18	23	1	23.845	0	0	0	NaN	0
19	56	1	40.300	0	0	3	10602.38500	1
20	30	1	35.300	0	1	3	36837.46700	1
21	60	0	36.005	0	0	0	13228.84695	1
22	30	0	32.400	1	0	3	4149.73600	1
23	18	1	34.100	0	0	2	1137.01100	1
24	34	0	31.920	1	1	0	37701.87680	1
25	37	1	28.025	2	0	1	6203.90175	0
26	59	0	27.720	3	0	2	14001.13380	1
27	63	0	23.085	0	0	0	14451.83515	0
28	55	0	32.775	2	0	1	12268.63225	0
29	23	1	17.385	1	0	1	2775.19215	1
30	31	1	36.300	2	1	3	38711.00000	1
31	22	1	35.600	0	1	3	35585.57600	1
32	18	0	26.315	0	0	0	2198.18985	1
33	19	0	28.600	5	0	3	4687.79700	0
34	63	1	28.310	0	0	1	13770.09790	1
35	28	1	36.400	1	1	3	51194.55914	1
36	19	1	20.425	0	0	1	1625.43375	0
37	62	0	32.965	3	0	1	15612.19335	0
38	26	1	20.800	0	0	3	2302.30000	0
39	35	1	36.670	1	1	0	39774.27630	1
40	60	1	39.900	0	1	3	48173.36100	1
41	24	0	26.600	0	0	0	3046.06200	1
42	31	0	36.630	2	0	2	4949.75870	0
43	41	1	21.780	1	0	2	6272.47720	0
44	37	0	30.800	2	0	2	6313.75900	0
45	38	1	37.050	1	0	0	6079.67150	0
46	55	1	37.300	0	0	3	20630.28351	1
47	18	0	38.665	2	0	0	3393.35635	0
48	28	0	34.770	0	0	1	3556.92230	1
49	60	0	24.530	0	0	2	12629.89670	0
50	36	1	35.200	1	1	2	38709.17600	1
51	18	0	35.625	0	0	0	2211.13075	1
52	21	0	33.630	2	0	1	3579.82870	0
53	48	1	28.000	1	1	3	23568.27200	1
54	36	1	34.430	0	1	2	37742.57570	1
55	40	0	28.690	3	0	1	8059.67910	0
56	58	1	36.955	2	1	1	47496.49445	1
57	58	0	31.825	2	0	0	13607.36875	0
58	18	1	31.680	2	1	2	34303.16720	1
59	53	0	22.880	1	1	2	23244.79020	1
60	34	0	37.335	2	0	1	5989.52365	0

61	43	1	27.360	3	0	0	8606.21740	1
62	25	1	33.660	4	0	2	4504.66240	0
63	64	1	24.700	1	0	1	30166.61817	0
64	28	0	25.935	1	0	1	4133.64165	0
65	20	0	22.420	0	1	1	14711.74380	1
66	19	0	28.900	0	0	3	1743.21400	1
67	61	0	39.100	2	0	3	14235.07200	1
68	40	1	26.315	1	0	1	6389.37785	0
69	40	0	36.190	0	0	2	5920.10410	1
70	28	1	23.980	3	1	2	17663.14420	0
71	27	0	24.750	0	1	2	16577.77950	1
72	31	1	28.500	5	0	0	6799.45800	0
73	53	0	28.100	3	0	3	11741.72600	1
74	58	1	32.010	1	0	2	11946.62590	1
75	44	1	27.400	2	0	3	7726.85400	1
76	57	1	34.010	0	0	1	11356.66090	1
77	29	0	29.590	1	0	2	3947.41310	0
78	21	1	35.530	0	0	2	1532.46970	1
79	22	0	39.805	0	0	0	2755.02095	1
80	41	0	32.965	0	0	1	6571.02435	1
81	31	1	26.885	1	0	0	4441.21315	0
82	45	0	38.285	0	0	0	7935.29115	1
83	22	1	37.620	1	1	2	37165.16380	1
84	48	0	41.230	4	0	1	11033.66170	0
85	37	0	34.800	2	1	3	39836.51900	1
86	45	1	22.895	2	1	1	21098.55405	1
87	57	0	31.160	0	1	1	43578.93940	1
88	56	0	27.200	0	0	3	11073.17600	1
89	46	0	27.740	0	0	1	8026.66660	1
90	55	0	26.980	0	0	1	11082.57720	1
91	21	0	39.490	0	0	2	2026.97410	1
92	53	0	24.795	1	0	1	10942.13205	0
93	59	1	29.830	3	1	0	30184.93670	1
94	35	1	34.770	2	0	1	5729.00530	0
95	64	0	31.300	2	1	3	47291.05500	1
96	28	0	37.620	1	0	2	3766.88380	0
97	54	0	30.800	3	0	3	12105.32000	0
98	55	1	38.280	0	0	2	10226.28420	1
99	56	1	19.950	0	1	0	22412.64850	1
100	38	1	19.300	0	1	3	15820.69900	1

1 Ans:

In [12]:
```
import pandas as obj
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

print("Removing null values: \n")

Insurance_data.dropna(axis = 0, inplace = True)
print(Insurance_data.to_string())
```

Output:

Removing null values:

	age	sex	bmi	children	smoker	region	charges	insuranceclaim
0	19	0	27.900	0	1	3	16884.92400	1
1	18	1	33.770	1	0	2	1725.55230	1
2	28	1	33.000	3	0	2	4449.46200	0
3	33	1	22.705	0	0	1	21984.47061	0
4	32	1	28.880	0	0	1	3866.85520	1
5	31	0	25.740	0	0	2	3756.62160	0
7	37	0	27.740	3	0	1	7281.50560	0
8	37	0	27.740	3	0	1	7281.50560	0
9	37	1	29.830	2	0	0	6406.41070	0
10	600	0	25.840	0	0	1	28923.13692	0
11	25	1	26.220	0	0	0	2721.32080	1
12	62	0	26.290	0	1	2	27808.72510	1
13	23	1	34.400	0	0	3	1826.84300	1
14	56	0	39.820	0	0	2	11090.71780	1
15	27	1	42.130	0	1	2	39611.75770	1
16	19	1	24.600	1	0	3	1837.23700	0
17	520	0	30.780	1	0	0	10797.33620	1
19	56	1	40.300	0	0	3	10602.38500	1
20	30	1	35.300	0	1	3	36837.46700	1
21	60	0	36.005	0	0	0	13228.84695	1
22	30	0	32.400	1	0	3	4149.73600	1
23	18	1	34.100	0	0	2	1137.01100	1
24	34	0	31.920	1	1	0	37701.87680	1
25	37	1	28.025	2	0	1	6203.90175	0
26	59	0	27.720	3	0	2	14001.13380	1
27	63	0	23.085	0	0	0	14451.83515	0
28	55	0	32.775	2	0	1	12268.63225	0
29	23	1	17.385	1	0	1	2775.19215	1
30	31	1	36.300	2	1	3	38711.00000	1
31	22	1	35.600	0	1	3	35585.57600	1
32	18	0	26.315	0	0	0	2198.18985	1
33	19	0	28.600	5	0	3	4687.79700	0
34	63	1	28.310	0	0	1	13770.09790	1
35	28	1	36.400	1	1	3	51194.55914	1
36	19	1	20.425	0	0	1	1625.43375	0
37	62	0	32.965	3	0	1	15612.19335	0
38	26	1	20.800	0	0	3	2302.30000	0
39	35	1	36.670	1	1	0	39774.27630	1
40	60	1	39.900	0	1	3	48173.36100	1
41	24	0	26.600	0	0	0	3046.06200	1
42	31	0	36.630	2	0	2	4949.75870	0
43	41	1	21.780	1	0	2	6272.47720	0
44	37	0	30.800	2	0	2	6313.75900	0
45	38	1	37.050	1	0	0	6079.67150	0
46	55	1	37.300	0	0	3	20630.28351	1
47	18	0	38.665	2	0	0	3393.35635	0
48	28	0	34.770	0	0	1	3556.92230	1
49	60	0	24.530	0	0	2	12629.89670	0
50	36	1	35.200	1	1	2	38709.17600	1

51	18	0	35.625	0	0	0	2211.13075	1
52	21	0	33.630	2	0	1	3579.82870	0
53	48	1	28.000	1	1	3	23568.27200	1
54	36	1	34.430	0	1	2	37742.57570	1
55	40	0	28.690	3	0	1	8059.67910	0
56	58	1	36.955	2	1	1	47496.49445	1
57	58	0	31.825	2	0	0	13607.36875	0
58	18	1	31.680	2	1	2	34303.16720	1
59	53	0	22.880	1	1	2	23244.79020	1
60	34	0	37.335	2	0	1	5989.52365	0
61	43	1	27.360	3	0	0	8606.21740	1
62	25	1	33.660	4	0	2	4504.66240	0
63	64	1	24.700	1	0	1	30166.61817	0
64	28	0	25.935	1	0	1	4133.64165	0
65	20	0	22.420	0	1	1	14711.74380	1
66	19	0	28.900	0	0	3	1743.21400	1
67	61	0	39.100	2	0	3	14235.07200	1
68	40	1	26.315	1	0	1	6389.37785	0
69	40	0	36.190	0	0	2	5920.10410	1
70	28	1	23.980	3	1	2	17663.14420	0
71	27	0	24.750	0	1	2	16577.77950	1
72	31	1	28.500	5	0	0	6799.45800	0
73	53	0	28.100	3	0	3	11741.72600	1
74	58	1	32.010	1	0	2	11946.62590	1
75	44	1	27.400	2	0	3	7726.85400	1
76	57	1	34.010	0	0	1	11356.66090	1
77	29	0	29.590	1	0	2	3947.41310	0
78	21	1	35.530	0	0	2	1532.46970	1
79	22	0	39.805	0	0	0	2755.02095	1
80	41	0	32.965	0	0	1	6571.02435	1
81	31	1	26.885	1	0	0	4441.21315	0
82	45	0	38.285	0	0	0	7935.29115	1
83	22	1	37.620	1	1	2	37165.16380	1
84	48	0	41.230	4	0	1	11033.66170	0
85	37	0	34.800	2	1	3	39836.51900	1
86	45	1	22.895	2	1	1	21098.55405	1
87	57	0	31.160	0	1	1	43578.93940	1
88	56	0	27.200	0	0	3	11073.17600	1
89	46	0	27.740	0	0	1	8026.66660	1
90	55	0	26.980	0	0	1	11082.57720	1
91	21	0	39.490	0	0	2	2026.97410	1
92	53	0	24.795	1	0	1	10942.13205	0
93	59	1	29.830	3	1	0	30184.93670	1
94	35	1	34.770	2	0	1	5729.00530	0
95	64	0	31.300	2	1	3	47291.05500	1
96	28	0	37.620	1	0	2	3766.88380	0
97	54	0	30.800	3	0	3	12105.32000	0
98	55	1	38.280	0	0	2	10226.28420	1
99	56	1	19.950	0	1	0	22412.64850	1
100	38	1	19.300	0	1	3	15820.69900	1

2a) Ans:

In [13]:
```
import pandas as obj
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

print("Filling the Empty cells in the 'charges' column with mean: ")

x = Insurance_data["charges"].mean()
print("\nMean = ", x, "\n")

Insurance_data["charges"].fillna(x, inplace = True)
print(Insurance_data.to_string())
```

```
Filling the Empty cells in the 'charges' column with mean:

Mean =  14701.3929969697

     age  sex     bmi  children  smoker  region      charges  insuranceclaim
0     19    0  27.900         0       1       3  16884.924000               1
1     18    1  33.770         1       0       2   1725.552300               1
2     28    1  33.000         3       0       2   4449.462000               0
3     33    1  22.705         0       0       1  21984.470610               0
4     32    1  28.880         0       0       1   3866.855200               1
5     31    0  25.740         0       0       2   3756.621600               0
6     46    0  33.440         1       0       2  14701.392997               1
7     37    0  27.740         3       0       1   7281.505600               0
8     37    0  27.740         3       0       1   7281.505600               0
9     37    1  29.830         2       0       0   6406.410700               0
10   600    0  25.840         0       0       1  28923.136920               0
11    25    1  26.220         0       0       0   2721.320800               1
12    62    0  26.290         0       1       2  27808.725100               1
13    23    1  34.400         0       0       3   1826.843000               1
14    56    0  39.820         0       0       2  11090.717800               1
15    27    1  42.130         0       1       2  39611.757700               1
16    19    1  24.600         1       0       3   1837.237000               0
17   520    0  30.780         1       0       0  10797.336200               1
18    23    1  23.845         0       0       0  14701.392997               0
19    56    1  40.300         0       0       3  10602.385000               1
20    30    1  35.300         0       1       3  36837.467000               1
21    60    0  36.005         0       0       0  13228.846950               1
22    30    0  32.400         1       0       3   4149.736000               1
23    18    1  34.100         0       0       2   1137.011000               1
24    34    0  31.920         1       1       0  37701.876800               1
25    37    1  28.025         2       0       1   6203.901750               0
26    59    0  27.720         3       0       2  14001.133800               1
27    63    0  23.085         0       0       0  14451.835150               0
28    55    0  32.775         2       0       1  12268.632250               0
29    23    1  17.385         1       0       1   2775.192150               1
30    31    1  36.300         2       1       3  38711.000000               1
31    22    1  35.600         0       1       3  35585.576000               1
32    18    0  26.315         0       0       0   2198.189850               1
33    19    0  28.600         5       0       3   4687.797000               0
34    63    1  28.310         0       0       1  13770.097900               1
35    28    1  36.400         1       1       3  51194.559140               1
36    19    1  20.425         0       0       1   1625.433750               0
37    62    0  32.965         3       0       1  15612.193350               0
38    26    1  20.800         0       0       3   2302.300000               0
39    35    1  36.670         1       1       0  39774.276300               1
40    60    1  39.900         0       1       3  48173.361000               1
41    24    0  26.600         0       0       0   3046.062000               1
42    31    0  36.630         2       0       2   4949.758700               0
43    41    1  21.780         1       0       2   6272.477200               0
44    37    0  30.800         2       0       2   6313.759000               0
45    38    1  37.050         1       0       0   6079.671500               0
```

46	55	1	37.300	0	0	3	20630.283510	1
47	18	0	38.665	2	0	0	3393.356350	0
48	28	0	34.770	0	0	1	3556.922300	1
49	60	0	24.530	0	0	2	12629.896700	0
50	36	1	35.200	1	1	2	38709.176000	1
51	18	0	35.625	0	0	0	2211.130750	1
52	21	0	33.630	2	0	1	3579.828700	0
53	48	1	28.000	1	1	3	23568.272000	1
54	36	1	34.430	0	1	2	37742.575700	1
55	40	0	28.690	3	0	1	8059.679100	0
56	58	1	36.955	2	1	1	47496.494450	1
57	58	0	31.825	2	0	0	13607.368750	0
58	18	1	31.680	2	1	2	34303.167200	1
59	53	0	22.880	1	1	2	23244.790200	1
60	34	0	37.335	2	0	1	5989.523650	0
61	43	1	27.360	3	0	0	8606.217400	1
62	25	1	33.660	4	0	2	4504.662400	0
63	64	1	24.700	1	0	1	30166.618170	0
64	28	0	25.935	1	0	1	4133.641650	0
65	20	0	22.420	0	1	1	14711.743800	1
66	19	0	28.900	0	0	3	1743.214000	1
67	61	0	39.100	2	0	3	14235.072000	1
68	40	1	26.315	1	0	1	6389.377850	0
69	40	0	36.190	0	0	2	5920.104100	1
70	28	1	23.980	3	1	2	17663.144200	0
71	27	0	24.750	0	1	2	16577.779500	1
72	31	1	28.500	5	0	0	6799.458000	0
73	53	0	28.100	3	0	3	11741.726000	1
74	58	1	32.010	1	0	2	11946.625900	1
75	44	1	27.400	2	0	3	7726.854000	1
76	57	1	34.010	0	0	1	11356.660900	1
77	29	0	29.590	1	0	2	3947.413100	0
78	21	1	35.530	0	0	2	1532.469700	1
79	22	0	39.805	0	0	0	2755.020950	1
80	41	0	32.965	0	0	1	6571.024350	1
81	31	1	26.885	1	0	0	4441.213150	0
82	45	0	38.285	0	0	0	7935.291150	1
83	22	1	37.620	1	1	2	37165.163800	1
84	48	0	41.230	4	0	1	11033.661700	0
85	37	0	34.800	2	1	3	39836.519000	1
86	45	1	22.895	2	1	1	21098.554050	1
87	57	0	31.160	0	1	1	43578.939400	1
88	56	0	27.200	0	0	3	11073.176000	1
89	46	0	27.740	0	0	1	8026.666600	1
90	55	0	26.980	0	0	1	11082.577200	1
91	21	0	39.490	0	0	2	2026.974100	1
92	53	0	24.795	1	0	1	10942.132050	0
93	59	1	29.830	3	1	0	30184.936700	1
94	35	1	34.770	2	0	1	5729.005300	0
95	64	0	31.300	2	1	3	47291.055000	1
96	28	0	37.620	1	0	2	3766.883800	0
97	54	0	30.800	3	0	3	12105.320000	0
98	55	1	38.280	0	0	2	10226.284200	1
99	56	1	19.950	0	1	0	22412.648500	1
100	38	1	19.300	0	1	3	15820.699000	1

2b) Ans:

In [9]:
```
import pandas as obj
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

print("Filling the Empty cells in the 'charges' column with median: ")

x = Insurance_data["charges"].median()
print("\nMedian = ", x, "\n")

Insurance_data["charges"].fillna(x, inplace = True)
print(Insurance_data.to_string())
```

```
Filling the Empty cells in the 'charges' column with median:

Median =  10797.3362

     age  sex     bmi  children  smoker  region      charges  insuranceclaim
0     19    0  27.900         0       1       3  16884.92400               1
1     18    1  33.770         1       0       2   1725.55230               1
2     28    1  33.000         3       0       2   4449.46200               0
3     33    1  22.705         0       0       1  21984.47061               0
4     32    1  28.880         0       0       1   3866.85520               1
5     31    0  25.740         0       0       2   3756.62160               0
6     46    0  33.440         1       0       2  10797.33620               1
7     37    0  27.740         3       0       1   7281.50560               0
8     37    0  27.740         3       0       1   7281.50560               0
9     37    1  29.830         2       0       0   6406.41070               0
10   600    0  25.840         0       0       1  28923.13692               0
11    25    1  26.220         0       0       0   2721.32080               1
12    62    0  26.290         0       1       2  27808.72510               1
13    23    1  34.400         0       0       3   1826.84300               1
14    56    0  39.820         0       0       2  11090.71780               1
15    27    1  42.130         0       1       2  39611.75770               1
16    19    1  24.600         1       0       3   1837.23700               0
17   520    0  30.780         1       0       0  10797.33620               1
18    23    1  23.845         0       0       0  10797.33620               0
19    56    1  40.300         0       0       3  10602.38500               1
20    30    1  35.300         0       1       3  36837.46700               1
21    60    0  36.005         0       0       0  13228.84695               1
22    30    0  32.400         1       0       3   4149.73600               1
23    18    1  34.100         0       0       2   1137.01100               1
24    34    0  31.920         1       1       0  37701.87680               1
25    37    1  28.025         2       0       1   6203.90175               0
26    59    0  27.720         3       0       2  14001.13380               1
27    63    0  23.085         0       0       0  14451.83515               0
28    55    0  32.775         2       0       1  12268.63225               0
29    23    1  17.385         1       0       1   2775.19215               1
30    31    1  36.300         2       1       3  38711.00000               1
31    22    1  35.600         0       1       3  35585.57600               1
32    18    0  26.315         0       0       0   2198.18985               1
33    19    0  28.600         5       0       3   4687.79700               0
34    63    1  28.310         0       0       1  13770.09790               1
35    28    1  36.400         1       1       3  51194.55914               1
36    19    1  20.425         0       0       1   1625.43375               0
37    62    0  32.965         3       0       1  15612.19335               0
38    26    1  20.800         0       0       3   2302.30000               0
39    35    1  36.670         1       1       0  39774.27630               1
40    60    1  39.900         0       1       3  48173.36100               1
41    24    0  26.600         0       0       0   3046.06200               1
42    31    0  36.630         2       0       2   4949.75870               0
43    41    1  21.780         1       0       2   6272.47720               0
44    37    0  30.800         2       0       2   6313.75900               0
45    38    1  37.050         1       0       0   6079.67150               0
```

46	55	1	37.300	0	0	3	20630.28351	1
47	18	0	38.665	2	0	0	3393.35635	0
48	28	0	34.770	0	0	1	3556.92230	1
49	60	0	24.530	0	0	2	12629.89670	0
50	36	1	35.200	1	1	2	38709.17600	1
51	18	0	35.625	0	0	0	2211.13075	1
52	21	0	33.630	2	0	1	3579.82870	0
53	48	1	28.000	1	1	3	23568.27200	1
54	36	1	34.430	0	1	2	37742.57570	1
55	40	0	28.690	3	0	1	8059.67910	0
56	58	1	36.955	2	1	1	47496.49445	1
57	58	0	31.825	2	0	0	13607.36875	0
58	18	1	31.680	2	1	2	34303.16720	1
59	53	0	22.880	1	1	2	23244.79020	1
60	34	0	37.335	2	0	1	5989.52365	0
61	43	1	27.360	3	0	0	8606.21740	1
62	25	1	33.660	4	0	2	4504.66240	0
63	64	1	24.700	1	0	1	30166.61817	0
64	28	0	25.935	1	0	1	4133.64165	0
65	20	0	22.420	0	1	1	14711.74380	1
66	19	0	28.900	0	0	3	1743.21400	1
67	61	0	39.100	2	0	3	14235.07200	1
68	40	1	26.315	1	0	1	6389.37785	0
69	40	0	36.190	0	0	2	5920.10410	1
70	28	1	23.980	3	1	2	17663.14420	0
71	27	0	24.750	0	1	2	16577.77950	1
72	31	1	28.500	5	0	0	6799.45800	0
73	53	0	28.100	3	0	3	11741.72600	1
74	58	1	32.010	1	0	2	11946.62590	1
75	44	1	27.400	2	0	3	7726.85400	1
76	57	1	34.010	0	0	1	11356.66090	1
77	29	0	29.590	1	0	2	3947.41310	0
78	21	1	35.530	0	0	2	1532.46970	1
79	22	0	39.805	0	0	0	2755.02095	1
80	41	0	32.965	0	0	1	6571.02435	1
81	31	1	26.885	1	0	0	4441.21315	0
82	45	0	38.285	0	0	0	7935.29115	1
83	22	1	37.620	1	1	2	37165.16380	1
84	48	0	41.230	4	0	1	11033.66170	0
85	37	0	34.800	2	1	3	39836.51900	1
86	45	1	22.895	2	1	1	21098.55405	1
87	57	0	31.160	0	1	1	43578.93940	1
88	56	0	27.200	0	0	3	11073.17600	1
89	46	0	27.740	0	0	1	8026.66660	1
90	55	0	26.980	0	0	1	11082.57720	1
91	21	0	39.490	0	0	2	2026.97410	1
92	53	0	24.795	1	0	1	10942.13205	0
93	59	1	29.830	3	1	0	30184.93670	1
94	35	1	34.770	2	0	1	5729.00530	0
95	64	0	31.300	2	1	3	47291.05500	1
96	28	0	37.620	1	0	2	3766.88380	0
97	54	0	30.800	3	0	3	12105.32000	0
98	55	1	38.280	0	0	2	10226.28420	1
99	56	1	19.950	0	1	0	22412.64850	1
100	38	1	19.300	0	1	3	15820.69900	1

2c) Ans:

In [30]:
```
import pandas as obj
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

print("Filling the Empty cells in the 'charges' column with mode: ")

x = Insurance_data["charges"].mode()[0]
print("\n Mode = ", x, "\n")

Insurance_data["charges"].fillna(x, inplace = True)
print(Insurance_data.to_string())
```

```
Filling the Empty cells in the 'charges' column with mode:

 Mode =  7281.5056

     age  sex     bmi  children  smoker  region      charges  insuranceclaim
0     19    0  27.900         0       1       3  16884.92400               1
1     18    1  33.770         1       0       2   1725.55230               1
2     28    1  33.000         3       0       2   4449.46200               0
3     33    1  22.705         0       0       1  21984.47061               0
4     32    1  28.880         0       0       1   3866.85520               1
5     31    0  25.740         0       0       2   3756.62160               0
6     46    0  33.440         1       0       2   7281.50560               1
7     37    0  27.740         3       0       1   7281.50560               0
8     37    0  27.740         3       0       1   7281.50560               0
9     37    1  29.830         2       0       0   6406.41070               0
10   600    0  25.840         0       0       1  28923.13692               0
11    25    1  26.220         0       0       0   2721.32080               1
12    62    0  26.290         0       1       2  27808.72510               1
13    23    1  34.400         0       0       3   1826.84300               1
14    56    0  39.820         0       0       2  11090.71780               1
15    27    1  42.130         0       1       2  39611.75770               1
16    19    1  24.600         1       0       3   1837.23700               0
17   520    0  30.780         1       0       0  10797.33620               1
18    23    1  23.845         0       0       0   7281.50560               0
19    56    1  40.300         0       0       3  10602.38500               1
20    30    1  35.300         0       1       3  36837.46700               1
21    60    0  36.005         0       0       0  13228.84695               1
22    30    0  32.400         1       0       3   4149.73600               1
23    18    1  34.100         0       0       2   1137.01100               1
24    34    0  31.920         1       1       0  37701.87680               1
25    37    1  28.025         2       0       1   6203.90175               0
26    59    0  27.720         3       0       2  14001.13380               1
27    63    0  23.085         0       0       0  14451.83515               0
28    55    0  32.775         2       0       1  12268.63225               0
29    23    1  17.385         1       0       1   2775.19215               1
30    31    1  36.300         2       1       3  38711.00000               1
31    22    1  35.600         0       1       3  35585.57600               1
32    18    0  26.315         0       0       0   2198.18985               1
33    19    0  28.600         5       0       3   4687.79700               0
34    63    1  28.310         0       0       1  13770.09790               1
35    28    1  36.400         1       1       3  51194.55914               1
36    19    1  20.425         0       0       1   1625.43375               0
37    62    0  32.965         3       0       1  15612.19335               0
38    26    1  20.800         0       0       3   2302.30000               0
39    35    1  36.670         1       1       0  39774.27630               1
40    60    1  39.900         0       1       3  48173.36100               1
41    24    0  26.600         0       0       0   3046.06200               1
42    31    0  36.630         2       0       2   4949.75870               0
43    41    1  21.780         1       0       2   6272.47720               0
44    37    0  30.800         2       0       2   6313.75900               0
45    38    1  37.050         1       0       0   6079.67150               0
```

46	55	1	37.300	0	0	3	20630.28351	1
47	18	0	38.665	2	0	0	3393.35635	0
48	28	0	34.770	0	0	1	3556.92230	1
49	60	0	24.530	0	0	2	12629.89670	0
50	36	1	35.200	1	1	2	38709.17600	1
51	18	0	35.625	0	0	0	2211.13075	1
52	21	0	33.630	2	0	1	3579.82870	0
53	48	1	28.000	1	1	3	23568.27200	1
54	36	1	34.430	0	1	2	37742.57570	1
55	40	0	28.690	3	0	1	8059.67910	0
56	58	1	36.955	2	1	1	47496.49445	1
57	58	0	31.825	2	0	0	13607.36875	0
58	18	1	31.680	2	1	2	34303.16720	1
59	53	0	22.880	1	1	2	23244.79020	1
60	34	0	37.335	2	0	1	5989.52365	0
61	43	1	27.360	3	0	0	8606.21740	1
62	25	1	33.660	4	0	2	4504.66240	0
63	64	1	24.700	1	0	1	30166.61817	0
64	28	0	25.935	1	0	1	4133.64165	0
65	20	0	22.420	0	1	1	14711.74380	1
66	19	0	28.900	0	0	3	1743.21400	1
67	61	0	39.100	2	0	3	14235.07200	1
68	40	1	26.315	1	0	1	6389.37785	0
69	40	0	36.190	0	0	2	5920.10410	1
70	28	1	23.980	3	1	2	17663.14420	0
71	27	0	24.750	0	1	2	16577.77950	1
72	31	1	28.500	5	0	0	6799.45800	0
73	53	0	28.100	3	0	3	11741.72600	1
74	58	1	32.010	1	0	2	11946.62590	1
75	44	1	27.400	2	0	3	7726.85400	1
76	57	1	34.010	0	0	1	11356.66090	1
77	29	0	29.590	1	0	2	3947.41310	0
78	21	1	35.530	0	0	2	1532.46970	1
79	22	0	39.805	0	0	0	2755.02095	1
80	41	0	32.965	0	0	1	6571.02435	1
81	31	1	26.885	1	0	0	4441.21315	0
82	45	0	38.285	0	0	0	7935.29115	1
83	22	1	37.620	1	1	2	37165.16380	1
84	48	0	41.230	4	0	1	11033.66170	0
85	37	0	34.800	2	1	3	39836.51900	1
86	45	1	22.895	2	1	1	21098.55405	1
87	57	0	31.160	0	1	1	43578.93940	1
88	56	0	27.200	0	0	3	11073.17600	1
89	46	0	27.740	0	0	1	8026.66660	1
90	55	0	26.980	0	0	1	11082.57720	1
91	21	0	39.490	0	0	2	2026.97410	1
92	53	0	24.795	1	0	1	10942.13205	0
93	59	1	29.830	3	1	0	30184.93670	1
94	35	1	34.770	2	0	1	5729.00530	0
95	64	0	31.300	2	1	3	47291.05500	1
96	28	0	37.620	1	0	2	3766.88380	0
97	54	0	30.800	3	0	3	12105.32000	0
98	55	1	38.280	0	0	2	10226.28420	1
99	56	1	19.950	0	1	0	22412.64850	1
100	38	1	19.300	0	1	3	15820.69900	1

3 Ans:

In [31]:
```
import pandas as obj
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

print("\n Replacing the values which are more than 60 in the 'age' column with the value of 50: \n")

for x in Insurance_data.index:
    if Insurance_data.loc[x, "age"] > 60:
        Insurance_data.loc[x, "age"] = 50

else:
    print(Insurance_data.to_string())
```

```
 Replacing the values which are more than 60 in the 'age' column with the value of 50:

      age  sex     bmi  children  smoker  region      charges  insuranceclaim
0      19    0  27.900         0       1       3  16884.92400               1
1      18    1  33.770         1       0       2   1725.55230               1
2      28    1  33.000         3       0       2   4449.46200               0
3      33    1  22.705         0       0       1  21984.47061               0
4      32    1  28.880         0       0       1   3866.85520               1
5      31    0  25.740         0       0       2   3756.62160               0
6      46    0  33.440         1       0       2          NaN               1
7      37    0  27.740         3       0       1   7281.50560               0
8      37    0  27.740         3       0       1   7281.50560               0
9      37    1  29.830         2       0       0   6406.41070               0
10     50    0  25.840         0       0       1  28923.13692               0
11     25    1  26.220         0       0       0   2721.32080               1
12     50    0  26.290         0       1       2  27808.72510               1
13     23    1  34.400         0       0       3   1826.84300               1
14     56    0  39.820         0       0       2  11090.71780               1
15     27    1  42.130         0       1       2  39611.75770               1
16     19    1  24.600         1       0       3   1837.23700               0
17     50    0  30.780         1       0       0  10797.33620               1
18     23    1  23.845         0       0       0          NaN               0
19     56    1  40.300         0       0       3  10602.38500               1
20     30    1  35.300         0       1       3  36837.46700               1
21     60    0  36.005         0       0       0  13228.84695               1
22     30    0  32.400         1       0       3   4149.73600               1
23     18    1  34.100         0       0       2   1137.01100               1
24     34    0  31.920         1       1       0  37701.87680               1
25     37    1  28.025         2       0       1   6203.90175               0
26     59    0  27.720         3       0       2  14001.13380               1
27     50    0  23.085         0       0       0  14451.83515               0
28     55    0  32.775         2       0       1  12268.63225               0
29     23    1  17.385         1       0       1   2775.19215               1
30     31    1  36.300         2       1       3  38711.00000               1
31     22    1  35.600         0       1       3  35585.57600               1
32     18    0  26.315         0       0       0   2198.18985               1
33     19    0  28.600         5       0       3   4687.79700               0
34     50    1  28.310         0       0       1  13770.09790               1
35     28    1  36.400         1       1       3  51194.55914               1
36     19    1  20.425         0       0       1   1625.43375               0
37     50    0  32.965         3       0       1  15612.19335               0
38     26    1  20.800         0       0       3   2302.30000               0
39     35    1  36.670         1       1       0  39774.27630               1
40     60    1  39.900         0       1       3  48173.36100               1
41     24    0  26.600         0       0       0   3046.06200               1
42     31    0  36.630         2       0       2   4949.75870               0
43     41    1  21.780         1       0       2   6272.47720               0
44     37    0  30.800         2       0       2   6313.75900               0
```

45	38	1	37.050	1	0	0	6079.67150	0
46	55	1	37.300	0	0	3	20630.28351	1
47	18	0	38.665	2	0	0	3393.35635	0
48	28	0	34.770	0	0	1	3556.92230	1
49	60	0	24.530	0	0	2	12629.89670	0
50	36	1	35.200	1	1	2	38709.17600	1
51	18	0	35.625	0	0	0	2211.13075	1
52	21	0	33.630	2	0	1	3579.82870	0
53	48	1	28.000	1	1	3	23568.27200	1
54	36	1	34.430	0	1	2	37742.57570	1
55	40	0	28.690	3	0	1	8059.67910	0
56	58	1	36.955	2	1	1	47496.49445	1
57	58	0	31.825	2	0	0	13607.36875	0
58	18	1	31.680	2	1	2	34303.16720	1
59	53	0	22.880	1	1	2	23244.79020	1
60	34	0	37.335	2	0	1	5989.52365	0
61	43	1	27.360	3	0	0	8606.21740	1
62	25	1	33.660	4	0	2	4504.66240	0
63	50	1	24.700	1	0	1	30166.61817	0
64	28	0	25.935	1	0	1	4133.64165	0
65	20	0	22.420	0	1	1	14711.74380	1
66	19	0	28.900	0	0	3	1743.21400	1
67	50	0	39.100	2	0	3	14235.07200	1
68	40	1	26.315	1	0	1	6389.37785	0
69	40	0	36.190	0	0	2	5920.10410	1
70	28	1	23.980	3	1	2	17663.14420	0
71	27	0	24.750	0	1	2	16577.77950	1
72	31	1	28.500	5	0	0	6799.45800	0
73	53	0	28.100	3	0	3	11741.72600	1
74	58	1	32.010	1	0	2	11946.62590	1
75	44	1	27.400	2	0	3	7726.85400	1
76	57	1	34.010	0	0	1	11356.66090	1
77	29	0	29.590	1	0	2	3947.41310	0
78	21	1	35.530	0	0	2	1532.46970	1
79	22	0	39.805	0	0	0	2755.02095	1
80	41	0	32.965	0	0	1	6571.02435	1
81	31	1	26.885	1	0	0	4441.21315	0
82	45	0	38.285	0	0	0	7935.29115	1
83	22	1	37.620	1	1	2	37165.16380	1
84	48	0	41.230	4	0	1	11033.66170	0
85	37	0	34.800	2	1	3	39836.51900	1
86	45	1	22.895	2	1	1	21098.55405	1
87	57	0	31.160	0	1	1	43578.93940	1
88	56	0	27.200	0	0	3	11073.17600	1
89	46	0	27.740	0	0	1	8026.66660	1
90	55	0	26.980	0	0	1	11082.57720	1
91	21	0	39.490	0	0	2	2026.97410	1
92	53	0	24.795	1	0	1	10942.13205	0
93	59	1	29.830	3	1	0	30184.93670	1
94	35	1	34.770	2	0	1	5729.00530	0
95	50	0	31.300	2	1	3	47291.05500	1
96	28	0	37.620	1	0	2	3766.88380	0
97	54	0	30.800	3	0	3	12105.32000	0
98	55	1	38.280	0	0	2	10226.28420	1
99	56	1	19.950	0	1	0	22412.64850	1
100	38	1	19.300	0	1	3	15820.69900	1

4 Ans:

In [21]:
```
import pandas as obj
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

print("Displaying the duplicates in the dataset: ")
print((Insurance_data.duplicated()).to_string())

print("\n\n Removing the duplicates in the dataset: \n")
Insurance_data.drop_duplicates(inplace = True)
print(Insurance_data.to_string())
```

```
Displaying the duplicates in the dataset:
0      False
1      False
2      False
3      False
4      False
5      False
6      False
7      False
8       True
9      False
10     False
11     False
12     False
13     False
14     False
15     False
16     False
17     False
18     False
19     False
20     False
21     False
22     False
23     False
24     False
25     False
26     False
27     False
28     False
29     False
30     False
31     False
32     False
33     False
34     False
35     False
36     False
37     False
38     False
39     False
40     False
41     False
42     False
43     False
44     False
45     False
46     False
47     False
48     False
49     False
50     False
51     False
52     False
53     False
54     False
55     False
56     False
57     False
58     False
59     False
60     False
61     False
62     False
63     False
64     False
65     False
66     False
67     False
68     False
69     False
70     False
71     False
72     False
73     False
74     False
75     False
76     False
77     False
78     False
79     False
80     False
81     False
82     False
83     False
84     False
85     False
86     False
87     False
88     False
89     False
90     False
91     False
92     False
93     False
94     False
95     False
96     False
97     False
98     False
99     False
100    False
```

Removing the duplicates in the dataset:

	age	sex	bmi	children	smoker	region	charges	insuranceclaim
0	19	0	27.900	0	1	3	16884.92400	1
1	18	1	33.770	1	0	2	1725.55230	1
2	28	1	33.000	3	0	2	4449.46200	0
3	33	1	22.705	0	0	1	21984.47061	0
4	32	1	28.880	0	0	1	3866.85520	1
5	31	0	25.740	0	0	2	3756.62160	0
6	46	0	33.440	1	0	2	NaN	1
7	37	0	27.740	3	0	1	7281.50560	0
9	37	1	29.830	2	0	0	6406.41070	0
10	600	0	25.840	0	0	1	28923.13692	0
11	25	1	26.220	0	0	0	2721.32080	1
12	62	0	26.290	0	1	2	27808.72510	1
13	23	1	34.400	0	0	3	1826.84300	1
14	56	0	39.820	0	0	2	11090.71780	1
15	27	1	42.130	0	1	2	39611.75770	1
16	19	1	24.600	1	0	3	1837.23700	0
17	520	0	30.780	1	0	0	10797.33620	1
18	23	1	23.845	0	0	0	NaN	0
19	56	1	40.300	0	0	3	10602.38500	1
20	30	1	35.300	0	1	3	36837.46700	1
21	60	0	36.005	0	0	0	13228.84695	1
22	30	0	32.400	1	0	3	4149.73600	1
23	18	1	34.100	0	0	2	1137.01100	1
24	34	0	31.920	1	1	0	37701.87680	1
25	37	1	28.025	2	0	1	6203.90175	0
26	59	0	27.720	3	0	2	14001.13380	1
27	63	0	23.085	0	0	0	14451.83515	0
28	55	0	32.775	2	0	1	12268.63225	0
29	23	1	17.385	1	0	1	2775.19215	1
30	31	1	36.300	2	1	3	38711.00000	1
31	22	1	35.600	0	1	3	35585.57600	1
32	18	0	26.315	0	0	0	2198.18985	1
33	19	0	28.600	5	0	3	4687.79700	0
34	63	1	28.310	0	0	1	13770.09790	1
35	28	1	36.400	1	1	3	51194.55914	1
36	19	1	20.425	0	0	1	1625.43375	0
37	62	0	32.965	3	0	1	15612.19335	0
38	26	1	20.800	0	0	3	2302.30000	0
39	35	1	36.670	1	1	0	39774.27630	1
40	60	1	39.900	0	1	3	48173.36100	1
41	24	0	26.600	0	0	0	3046.06200	1
42	31	0	36.630	2	0	2	4949.75870	0
43	41	1	21.780	1	0	2	6272.47720	0
44	37	0	30.800	2	0	2	6313.75900	0
45	38	1	37.050	1	0	0	6079.67150	0
46	55	1	37.300	0	0	3	20630.28351	1
47	18	0	38.665	2	0	0	3393.35635	0
48	28	0	34.770	0	0	1	3556.92230	1

49	60	0	24.530	0	0	2	12629.89670	0
50	36	1	35.200	1	1	2	38709.17600	1
51	18	0	35.625	0	0	0	2211.13075	1
52	21	0	33.630	2	0	1	3579.82870	0
53	48	1	28.000	1	1	3	23568.27200	1
54	36	1	34.430	0	1	2	37742.57570	1
55	40	0	28.690	3	0	1	8059.67910	0
56	58	1	36.955	2	1	1	47496.49445	1
57	58	0	31.825	2	0	0	13607.36875	0
58	18	1	31.680	2	1	2	34303.16720	1
59	53	0	22.880	1	1	2	23244.79020	1
60	34	0	37.335	2	0	1	5989.52365	0
61	43	1	27.360	3	0	0	8606.21740	1
62	25	1	33.660	4	0	2	4504.66240	0
63	64	1	24.700	1	0	1	30166.61817	0
64	28	0	25.935	1	0	1	4133.64165	0
65	20	0	22.420	0	1	1	14711.74380	1
66	19	0	28.900	0	0	3	1743.21400	1
67	61	0	39.100	2	0	3	14235.07200	1
68	40	1	26.315	1	0	1	6389.37785	0
69	40	0	36.190	0	0	2	5920.10410	1
70	28	1	23.980	3	1	2	17663.14420	0
71	27	0	24.750	0	1	2	16577.77950	1
72	31	1	28.500	5	0	0	6799.45800	0
73	53	0	28.100	3	0	3	11741.72600	1
74	58	1	32.010	1	0	2	11946.62590	1
75	44	1	27.400	2	0	3	7726.85400	1
76	57	1	34.010	0	0	1	11356.66090	1
77	29	0	29.590	1	0	2	3947.41310	0
78	21	1	35.530	0	0	2	1532.46970	1
79	22	0	39.805	0	0	0	2755.02095	1
80	41	0	32.965	0	0	1	6571.02435	1
81	31	1	26.885	1	0	0	4441.21315	0
82	45	0	38.285	0	0	0	7935.29115	1
83	22	1	37.620	1	1	2	37165.16380	1
84	48	0	41.230	4	0	1	11033.66170	0
85	37	0	34.800	2	1	3	39836.51900	1
86	45	1	22.895	2	1	1	21098.55405	1
87	57	0	31.160	0	1	1	43578.93940	1
88	56	0	27.200	0	0	3	11073.17600	1
89	46	0	27.740	0	0	1	8026.66660	1
90	55	0	26.980	0	0	1	11082.57720	1
91	21	0	39.490	0	0	2	2026.97410	1
92	53	0	24.795	1	0	1	10942.13205	0
93	59	1	29.830	3	1	0	30184.93670	1
94	35	1	34.770	2	0	1	5729.00530	0
95	64	0	31.300	2	1	3	47291.05500	1
96	28	0	37.620	1	0	2	3766.88380	0
97	54	0	30.800	3	0	3	12105.32000	0
98	55	1	38.280	0	0	2	10226.28420	1
99	56	1	19.950	0	1	0	22412.64850	1
100	38	1	19.300	0	1	3	15820.69900	1

5 Ans:

In [19]:
```
import pandas as obj
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

print("Finding out the correlation between the data: ")
Insurance_data.corr()
```

```
Finding out the correlation between the data:
```

Out[19]:

	age	sex	bmi	children	smoker	region	charges	insuranceclaim
age	1.000000	-0.158964	-0.064789	-0.060234	-0.080577	-0.132087	0.119514	-0.025922
sex	-0.158964	1.000000	-0.054925	-0.037696	0.219385	0.080228	0.205041	0.078158
bmi	-0.064789	-0.054925	1.000000	0.036698	0.013822	0.121468	0.189484	0.230812
children	-0.060234	-0.037696	0.036698	1.000000	-0.079262	-0.012957	-0.043886	-0.433757
smoker	-0.080577	0.219385	0.013822	-0.079262	1.000000	0.228907	0.823336	0.433123
region	-0.132087	0.080228	0.121468	-0.012957	0.228907	1.000000	0.201128	0.201042
charges	0.119514	0.205041	0.189484	-0.043886	0.823336	0.201128	1.000000	0.351759
insuranceclaim	-0.025922	0.078158	0.230812	-0.433757	0.433123	0.201042	0.351759	1.000000

6 Ans:

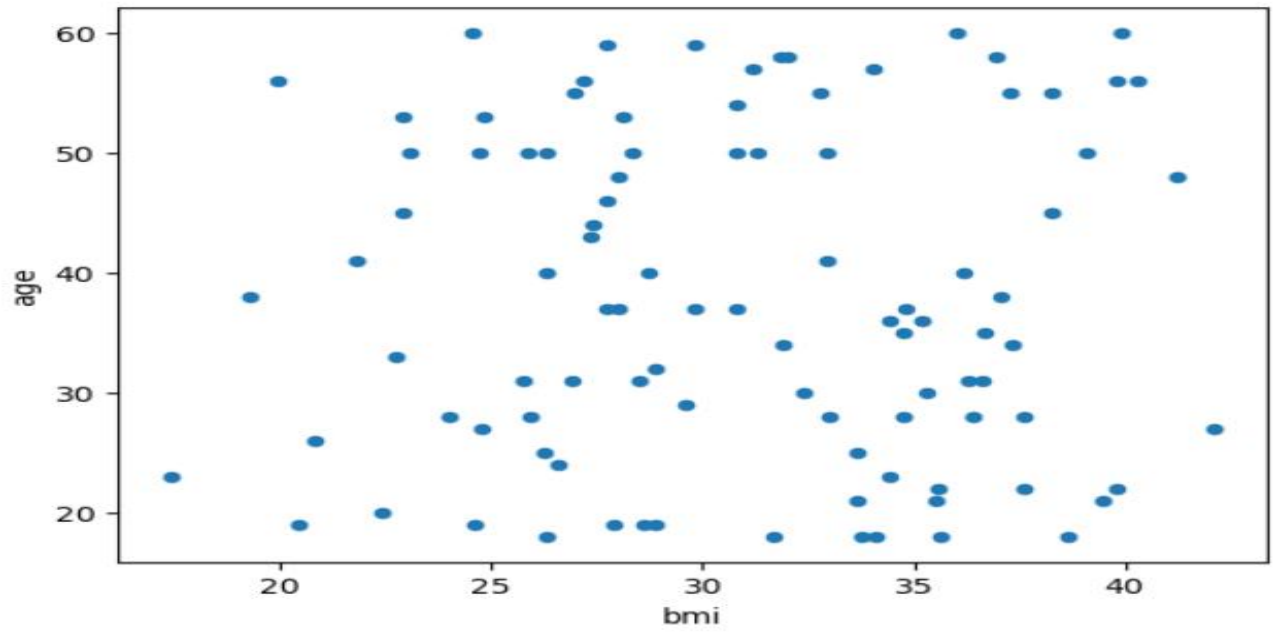
In [32]:
```
import pandas as obj
import matplotlib.pyplot as plt
Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")

Insurance_data.plot(kind = "scatter", x = 'bmi', y = 'age')

print('scatter plot between "age" and "bmi": ')
plt.show()
```

#scatter plot on corrected Insurance_data:

```
scatter plot between "age" and "bmi":
```



7 Ans:

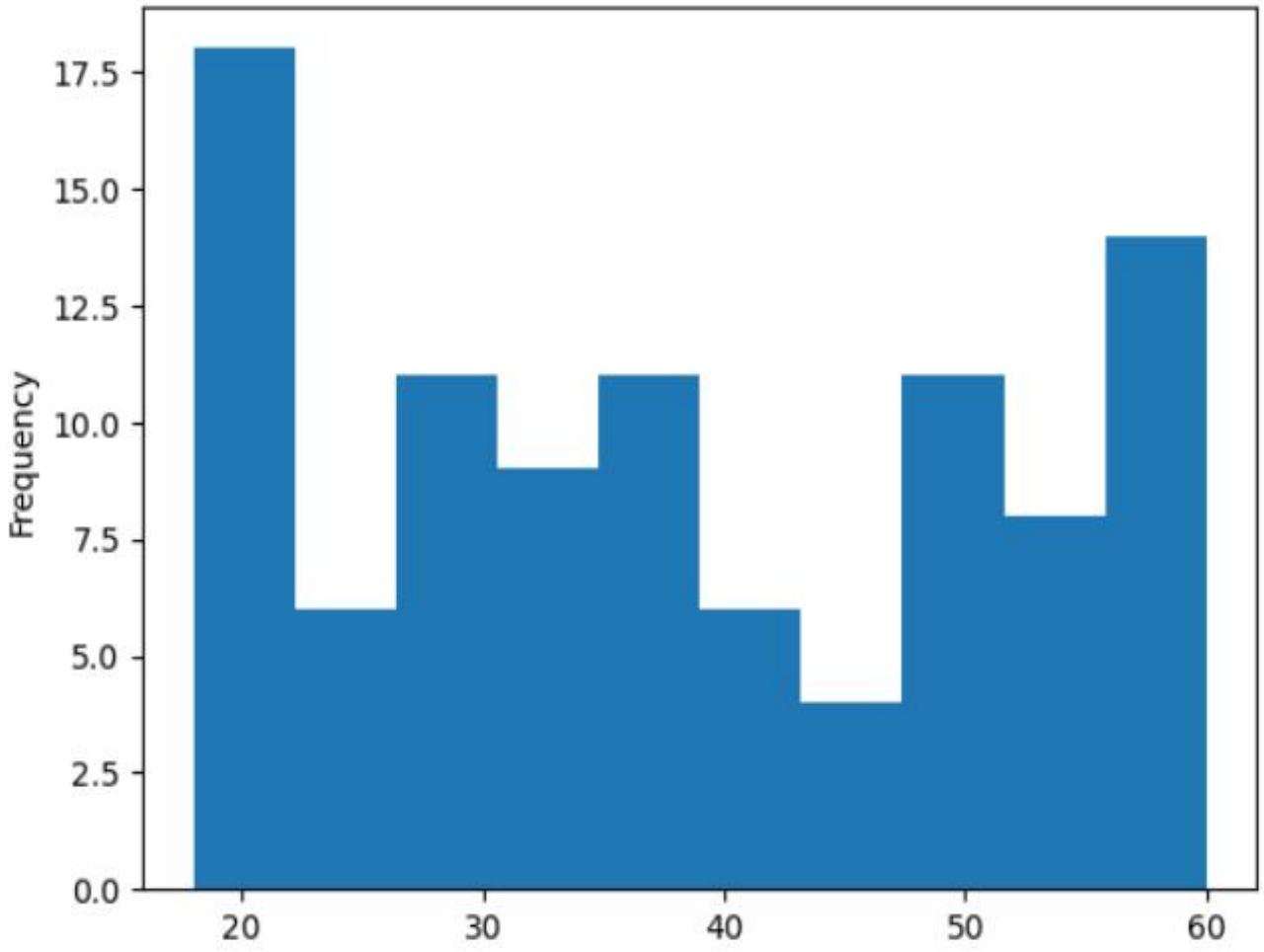
```
In [25]: import pandas as obj
         import matplotlib.pyplot as plt

         Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")
         Insurance_data['age'].plot(kind = 'hist')

         print("histogram plot on 'age' column: ")
         plt.show()
```

#Histogram on corrected Insurance_data:

```
histogram plot on 'age' column:
```



Python code:

```
import pandas as obj
import matplotlib.pyplot as plt

Insurance_data = obj.read_csv("C:\\Users\\banda\\Downloads\\insurance.csv")
print(Insurance_data.to_string())

print("Removing null values: \n")
Insurance_data.dropna(axis = 0, inplace = True)
print(Insurance_data.to_string())

print("Filling the Empty cells in the 'charges' column with mean: ")
x = Insurance_data["charges"].mean()
print("\nMean = ", x, "\n")
Insurance_data["charges"].fillna(x, inplace = True)
print(Insurance_data.to_string())

print("Filling the Empty cells in the 'charges' column with median: ")
y = Insurance_data["charges"].median()
print("\nMedian = ", y, "\n")
Insurance_data["charges"].fillna(y, inplace = True)
print(Insurance_data.to_string())

print("Filling the Empty cells in the 'charges' column with mode: ")
z = Insurance_data["charges"].mode()[0]
print("\n Mode = ", z, "\n")
Insurance_data["charges"].fillna(z, inplace = True)
print(Insurance_data.to_string())

print("\n Replacing the values which are more than 60 in the 'age' column with
the value of 50: \n")

for x in Insurance_data.index:
   if Insurance_data.loc[x, "age"] > 60:
      Insurance_data.loc[x, "age"] = 50

else:
   print(Insurance_data.to_string())
```

```
print("Displaying the duplicates in the dataset: ")
print((Insurance_data.duplicated()).to_string())
print("\n\n Removing the duplicates in the dataset: \n")
Insurance_data.drop_duplicates(inplace = True)
print(Insurance_data.to_string())

print("Finding out the correlation between the data: ")
print(Insurance_data.corr())

Insurance_data.plot(kind = "scatter", x = 'bmi', y = 'age')
print('scatter plot between "age" and "bmi": ')
plt.show()

Insurance_data['age'].plot(kind = 'hist')
print("histogram plot on 'age' column: ")
plt.show()
```