

```
import pandas as pd
```

```
df = pd.read_csv('/content/employee_data.csv')
print("Original Dataset:")
print(df)
```



Original Dataset:

	Employee_ID	Age	Salary	Job_Satisfaction	Work_Hours_Per_Week
0	1001	50.0	108953.0	9.0	36
1	1002	36.0	82995.0	8.0	59
2	1003	29.0	70757.0	2.0	30
3	1004	42.0	39692.0	1.0	30
4	1005	40.0	75758.0	7.0	54
5	1006	44.0	102409.0	7.0	56
6	1007	32.0	NaN	NaN	59
7	1008	32.0	95697.0	5.0	54
8	1009	45.0	67065.0	3.0	49
9	1010	57.0	NaN	8.0	42
10	1011	45.0	41534.0	6.0	38
11	1012	NaN	70397.0	NaN	32
12	1013	43.0	31016.0	NaN	36
13	1014	23.0	119789.0	1.0	35
14	1015	45.0	85591.0	3.0	37
15	1016	51.0	119812.0	5.0	56
16	1017	59.0	53247.0	3.0	38
17	1018	23.0	54300.0	1.0	59
18	1019	42.0	104065.0	5.0	34
19	1020	54.0	112798.0	10.0	30
20	1021	33.0	39268.0	7.0	48
21	1022	43.0	116807.0	7.0	39
22	1023	46.0	42185.0	NaN	41
23	1024	48.0	93704.0	9.0	53
24	1025	49.0	NaN	10.0	44
25	1026	37.0	69099.0	NaN	56
26	1027	36.0	NaN	3.0	51
27	1028	24.0	68044.0	7.0	53
28	1029	58.0	81214.0	1.0	38
29	1030	28.0	91228.0	4.0	49
30	1031	NaN	78984.0	4.0	46
31	1032	30.0	70774.0	NaN	59
32	1033	39.0	32568.0	7.0	46
33	1034	25.0	92592.0	7.0	55
34	1035	NaN	97563.0	NaN	49
35	1036	35.0	32695.0	4.0	41
36	1037	30.0	78190.0	7.0	59
37	1038	47.0	35258.0	NaN	36
38	1039	23.0	NaN	3.0	31
39	1040	NaN	69504.0	6.0	32
40	1041	49.0	63159.0	2.0	46
41	1042	28.0	43986.0	10.0	34
42	1043	29.0	91858.0	9.0	46
43	1044	56.0	42666.0	5.0	53
44	1045	35.0	68660.0	6.0	46

45	1046	38.0	33561.0	4.0	56
46	1047	57.0	56854.0	NaN	46
47	1048	25.0	94505.0	NaN	31
48	1049	23.0	82251.0	7.0	31
49	1050	NaN	52662.0	9.0	57

```
df.isna()
```



	Employee_ID	Age	Salary	Job_Satisfaction	Work_Hours_Per_Week
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	True	True	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	True	False	False
10	False	False	False	False	False
11	False	True	False	True	False
12	False	False	False	True	False
13	False	False	False	False	False
14	False	False	False	False	False
15	False	False	False	False	False
16	False	False	False	False	False
17	False	False	False	False	False
18	False	False	False	False	False
19	False	False	False	False	False
20	False	False	False	False	False
21	False	False	False	False	False
22	False	False	False	True	False
23	False	False	False	False	False
24	False	False	True	False	False
25	False	False	False	True	False
26	False	False	True	False	False
27	False	False	False	False	False
28	False	False	False	False	False
29	False	False	False	False	False



30	False	True	False	False	False
31	False	False	False	True	False
32	False	False	False	False	False
33	False	False	False	False	False
34	False	True	False	True	False
35	False	False	False	False	False
36	False	False	False	False	False
37	False	False	False	True	False
38	False	False	True	False	False
39	False	True	False	False	False
40	False	False	False	False	False
41	False	False	False	False	False
42	False	False	False	False	False
43	False	False	False	False	False
44	False	False	False	False	False
45	False	False	False	False	False
46	False	False	False	True	False
47	False	False	False	True	False
48	False	False	False	False	False
49	False	True	False	False	False

```
df.isnull().sum()
```

**0****Employee\_ID**

0

**Age**

5

**Salary**

5

**Job\_Satisfaction**

10

**Work\_Hours\_Per\_Week**

0

**dtype:** int64

```
df['Salary'].mean()
```



72349.2

```
df['Job_Satisfaction'].mean()
```



5.55

```
df['Salary'].mode()
```

**Salary**

<b>0</b>	31016.0
<b>1</b>	32568.0
<b>2</b>	32695.0
<b>3</b>	33561.0
<b>4</b>	35258.0
<b>5</b>	39268.0
<b>6</b>	39692.0
<b>7</b>	41534.0
<b>8</b>	42185.0
<b>9</b>	42666.0
<b>10</b>	43986.0
<b>11</b>	52662.0
<b>12</b>	53247.0
<b>13</b>	54300.0
<b>14</b>	56854.0
<b>15</b>	63159.0
<b>16</b>	67065.0
<b>17</b>	68044.0
<b>18</b>	68660.0
<b>19</b>	69099.0
<b>20</b>	69504.0
<b>21</b>	70397.0
<b>22</b>	70757.0
<b>23</b>	70774.0
<b>24</b>	75758.0
<b>25</b>	78190.0
<b>26</b>	78984.0
<b>27</b>	81214.0
<b>28</b>	82251.0
<b>29</b>	82995.0

```
30 85591.0
31 91228.0
32 91858.0
33 92592.0
34 93704.0
35 94505.0
36 95697.0
37 97563.0
38 102409.0
39 104065.0
40 108953.0
41 112798.0
42 116807.0
43 119789.0
44 119812.0
```

**dtype:** float64

```
df['Job_Satisfaction'].mode()
```

```

↵
  Job_Satisfaction
0                7.0

```

**dtype:** float64

```
df['Salary'].median()
```

```
↵ 70757.0
```

```
df['Job_Satisfaction'].median()
```

```
↵ 6.0
```

```
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
```

```

↵ <ipython-input-18-8332c1905e34>:1: FutureWarning: A value is trying to be set on a copy
  The behavior will change in pandas 3.0. This inplace method will never work because the
  For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col

```

```
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
```

```
df['Salary'] = df['Salary'].fillna(df['Salary'].mean())
```

```
df['Job_Satisfaction'] =df['Job_Satisfaction'].fillna(df['Job_Satisfaction'].mean())
```

```
df
```





	Employee_ID	Age	Salary	Job_Satisfaction	Work_Hours_Per_Week
0	1001	50.0	108953.0	9.00	36
1	1002	36.0	82995.0	8.00	59
2	1003	29.0	70757.0	2.00	30
3	1004	42.0	39692.0	1.00	30
4	1005	40.0	75758.0	7.00	54
5	1006	44.0	102409.0	7.00	56
6	1007	32.0	72349.2	5.55	59
7	1008	32.0	95697.0	5.00	54
8	1009	45.0	67065.0	3.00	49
9	1010	57.0	72349.2	8.00	42
10	1011	45.0	41534.0	6.00	38
11	1012	NaN	70397.0	5.55	32
12	1013	43.0	31016.0	5.55	36
13	1014	23.0	119789.0	1.00	35
14	1015	45.0	85591.0	3.00	37
15	1016	51.0	119812.0	5.00	56
16	1017	59.0	53247.0	3.00	38
17	1018	23.0	54300.0	1.00	59
18	1019	42.0	104065.0	5.00	34
19	1020	54.0	112798.0	10.00	30
20	1021	33.0	39268.0	7.00	48
21	1022	43.0	116807.0	7.00	39
22	1023	46.0	42185.0	5.55	41
23	1024	48.0	93704.0	9.00	53
24	1025	49.0	72349.2	10.00	44
25	1026	37.0	69099.0	5.55	56
26	1027	36.0	72349.2	3.00	51
27	1028	24.0	68044.0	7.00	53
28	1029	58.0	81214.0	1.00	38
29	1030	28.0	91228.0	4.00	49



<b>30</b>	1031	NaN	78984.0	4.00	46
<b>31</b>	1032	30.0	70774.0	5.55	59
<b>32</b>	1033	39.0	32568.0	7.00	46
<b>33</b>	1034	25.0	92592.0	7.00	55
<b>34</b>	1035	NaN	97563.0	5.55	49
<b>35</b>	1036	35.0	32695.0	4.00	41
<b>36</b>	1037	30.0	78190.0	7.00	59
<b>37</b>	1038	47.0	35258.0	5.55	36
<b>38</b>	1039	23.0	72349.2	3.00	31
<b>39</b>	1040	NaN	69504.0	6.00	32
<b>40</b>	1041	49.0	63159.0	2.00	46
<b>41</b>	1042	28.0	43986.0	10.00	34
<b>42</b>	1043	29.0	91858.0	9.00	46
<b>43</b>	1044	56.0	42666.0	5.00	53
<b>44</b>	1045	35.0	68660.0	6.00	46
<b>45</b>	1046	38.0	33561.0	4.00	56
<b>46</b>	1047	57.0	56854.0	5.55	46
<b>47</b>	1048	25.0	94505.0	5.55	31
<b>48</b>	1049	23.0	82251.0	7.00	31
<b>49</b>	1050	NaN	52662.0	9.00	57

```
print("Dataset after handling missing values:")
print(df)
```



Dataset after handling missing values:

	Employee_ID	Age	Salary	Job_Satisfaction	Work_Hours_Per_Week
0	1001	50.0	108953.0	9.00	36
1	1002	36.0	82995.0	8.00	59
2	1003	29.0	70757.0	2.00	30
3	1004	42.0	39692.0	1.00	30
4	1005	40.0	75758.0	7.00	54
5	1006	44.0	102409.0	7.00	56
6	1007	32.0	72349.2	5.55	59
7	1008	32.0	95697.0	5.00	54
8	1009	45.0	67065.0	3.00	49
9	1010	57.0	72349.2	8.00	42
10	1011	45.0	41534.0	6.00	38
11	1012	NaN	70397.0	5.55	32
12	1013	43.0	31016.0	5.55	36
13	1014	23.0	119789.0	1.00	35
14	1015	45.0	85591.0	3.00	37
15	1016	51.0	119812.0	5.00	56
16	1017	59.0	53247.0	3.00	38
17	1018	23.0	54300.0	1.00	59
18	1019	42.0	104065.0	5.00	34
19	1020	54.0	112798.0	10.00	30
20	1021	33.0	39268.0	7.00	48
21	1022	43.0	116807.0	7.00	39
22	1023	46.0	42185.0	5.55	41
23	1024	48.0	93704.0	9.00	53
24	1025	49.0	72349.2	10.00	44
25	1026	37.0	69099.0	5.55	56
26	1027	36.0	72349.2	3.00	51
27	1028	24.0	68044.0	7.00	53
28	1029	58.0	81214.0	1.00	38
29	1030	28.0	91228.0	4.00	49
30	1031	NaN	78984.0	4.00	46
31	1032	30.0	70774.0	5.55	59
32	1033	39.0	32568.0	7.00	46
33	1034	25.0	92592.0	7.00	55
34	1035	NaN	97563.0	5.55	49
35	1036	35.0	32695.0	4.00	41
36	1037	30.0	78190.0	7.00	59
37	1038	47.0	35258.0	5.55	36
38	1039	23.0	72349.2	3.00	31
39	1040	NaN	69504.0	6.00	32
40	1041	49.0	63159.0	2.00	46
41	1042	28.0	43986.0	10.00	34
42	1043	29.0	91858.0	9.00	46
43	1044	56.0	42666.0	5.00	53
44	1045	35.0	68660.0	6.00	46
45	1046	38.0	33561.0	4.00	56
46	1047	57.0	56854.0	5.55	46
47	1048	25.0	94505.0	5.55	31
48	1049	23.0	82251.0	7.00	31