

```
In [1]: 1 import numpy as np
        2 import pandas as pd
        3 import seaborn as sns
        4 import matplotlib.pyplot as plt
```

```
In [2]: 1 pd.options.display.max_columns = None
        2 pd.options.display.max_rows = None
```

```
In [3]: 1 df = pd.read_csv('Time-Wasters on Social Media.csv')
```

```
In [4]: 1 df.head()
```

Out[4]:

	UserID	Age	Gender	Location	Income	Debt	Owns Property	Profession	Demographics	Pla
0	1	56	Male	Pakistan	82812	True	True	Engineer	Rural	Inst
1	2	46	Female	Mexico	27999	False	True	Artist	Urban	Inst
2	3	32	Female	United States	42436	False	True	Engineer	Rural	Fac
3	4	60	Male	Barzil	62963	True	False	Waiting staff	Rural	Yo
4	5	25	Male	Pakistan	22096	False	True	Manager	Urban	

```
In [5]: 1 df.shape
```

Out[5]: (1000, 31)

In [6]:

1 df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 31 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   UserID                                1000 non-null   int64
1   Age                                   1000 non-null   int64
2   Gender                               1000 non-null   object
3   Location                             1000 non-null   object
4   Income                               1000 non-null   int64
5   Debt                                  1000 non-null   bool
6   Owns Property                        1000 non-null   bool
7   Profession                           1000 non-null   object
8   Demographics                         1000 non-null   object
9   Platform                             1000 non-null   object
10  Total Time Spent                      1000 non-null   int64
11  Number of Sessions                    1000 non-null   int64
12  Video ID                              1000 non-null   int64
13  Video Category                        1000 non-null   object
14  Video Length                          1000 non-null   int64
15  Engagement                            1000 non-null   int64
16  Importance Score                      1000 non-null   int64
17  Time Spent On Video                   1000 non-null   int64
18  Number of Videos Watched             1000 non-null   int64
19  Scroll Rate                           1000 non-null   int64
20  Frequency                             1000 non-null   object
21  ProductivityLoss                      1000 non-null   int64
22  Satisfaction                          1000 non-null   int64
23  Watch Reason                          1000 non-null   object
24  DeviceType                            1000 non-null   object
25  OS                                     1000 non-null   object
26  Watch Time                            1000 non-null   object
27  Self Control                          1000 non-null   int64
28  Addiction Level                       1000 non-null   int64
29  CurrentActivity                       1000 non-null   object
30  ConnectionType                        1000 non-null   object
dtypes: bool(2), int64(16), object(13)
memory usage: 228.6+ KB

```

```
In [7]: 1 df.describe()
```

Out[7]:

	UserID	Age	Income	Total Time Spent	Number of Sessions	Video ID	
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000
mean	500.500000	40.986000	59524.213000	151.406000	10.013000	4891.738000	15
std	288.819436	13.497852	23736.212925	83.952637	5.380314	2853.144258	8
min	1.000000	18.000000	20138.000000	10.000000	1.000000	11.000000	1
25%	250.750000	29.000000	38675.250000	78.000000	6.000000	2542.000000	8
50%	500.500000	42.000000	58805.000000	152.000000	10.000000	4720.500000	15
75%	750.250000	52.000000	79792.250000	223.000000	15.000000	7346.000000	22
max	1000.000000	64.000000	99676.000000	298.000000	19.000000	9997.000000	29

```
In [8]: 1 temp_df = df.select_dtypes(exclude=['object', 'bool'])
```

```
In [9]: 1 temp_df.shape
```

Out[9]: (1000, 16)

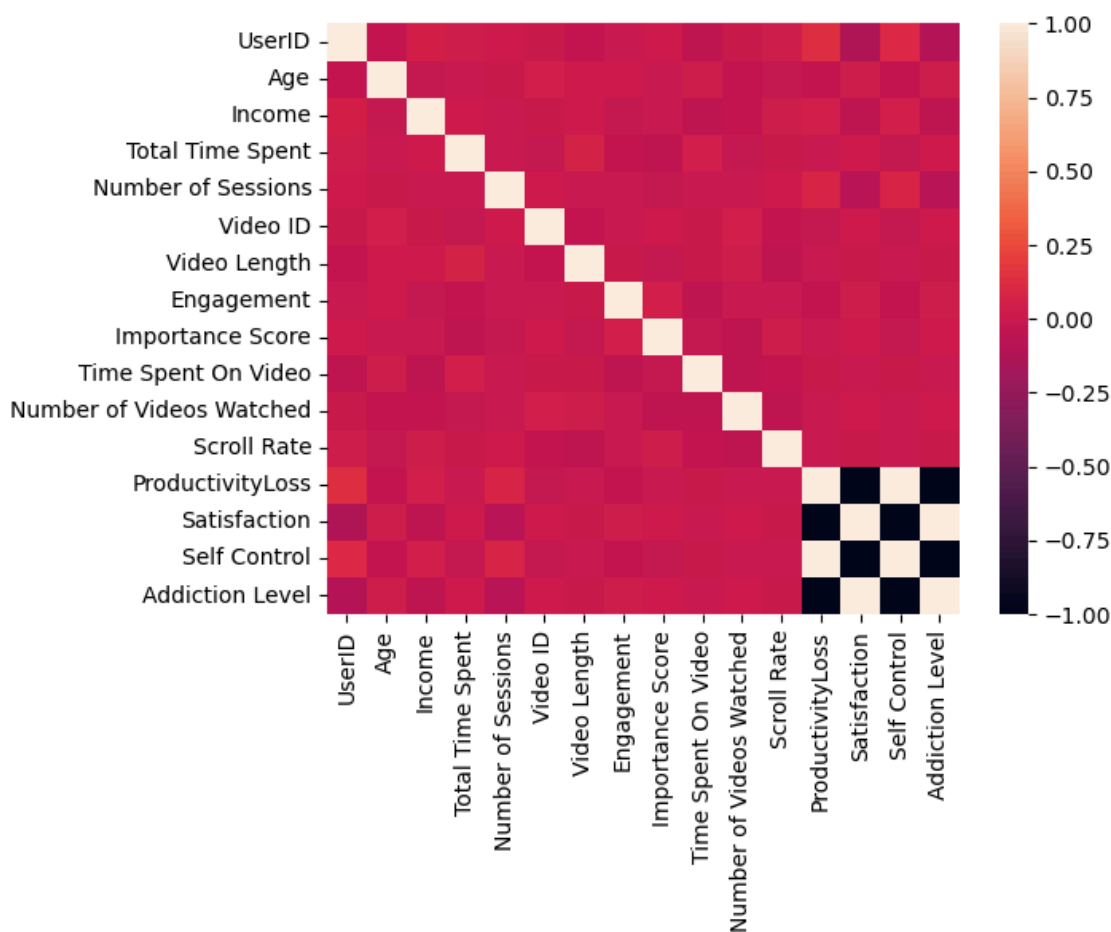
In [10]: 1 temp_df.corr()

Out[10]:

	UserID	Age	Income	Total Time Spent	Number of Sessions	Video ID	Video Length
UserID	1.000000	-0.024857	0.056221	0.035689	0.022750	0.002670	-0.037910
Age	-0.024857	1.000000	-0.017923	-0.004166	0.006563	0.047762	0.009557
Income	0.056221	-0.017923	1.000000	0.013326	-0.009979	0.003320	0.017432
Total Time Spent	0.035689	-0.004166	0.013326	1.000000	-0.013876	-0.018057	0.068607
Number of Sessions	0.022750	0.006563	-0.009979	-0.013876	1.000000	0.010646	-0.002076
Video ID	0.002670	0.047762	0.003320	-0.018057	0.010646	1.000000	-0.032363
Video Length	-0.037910	0.009557	0.017432	0.068607	-0.002076	-0.032363	1.000000
Engagement	-0.013374	0.010417	-0.017183	-0.033743	-0.007600	-0.003288	0.001286
Importance Score	0.009265	-0.014994	-0.000762	-0.043414	-0.016832	0.012940	-0.017086
Time Spent On Video	-0.055781	0.034535	-0.041666	0.053319	-0.014127	0.007697	0.004256
Number of Videos Watched	0.002627	-0.033776	-0.036211	-0.020969	-0.012761	0.043525	0.037753
Scroll Rate	0.025134	-0.018768	0.029066	0.004211	0.013610	-0.026161	-0.044744
ProductivityLoss	0.117922	-0.031068	0.040670	-0.014210	0.081461	-0.018680	-0.004364
Satisfaction	-0.117922	0.031068	-0.040670	0.014210	-0.081461	0.018680	0.004364
Self Control	0.105280	-0.033493	0.039181	-0.016086	0.080961	-0.020655	-0.004914
Addiction Level	-0.105280	0.033493	-0.039181	0.016086	-0.080961	0.020655	0.004914

```
In [11]: 1 sns.heatmap(temp_df.corr())
```

Out[11]: <Axes: >

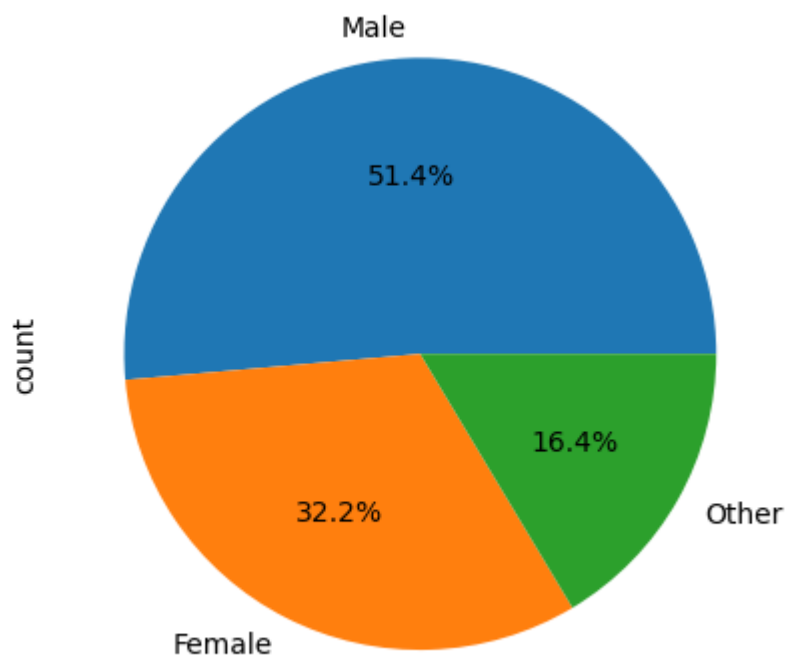


```
In [12]: 1 df['Gender'].value_counts()
```

Out[12]: Gender
Male 514
Female 322
Other 164
Name: count, dtype: int64

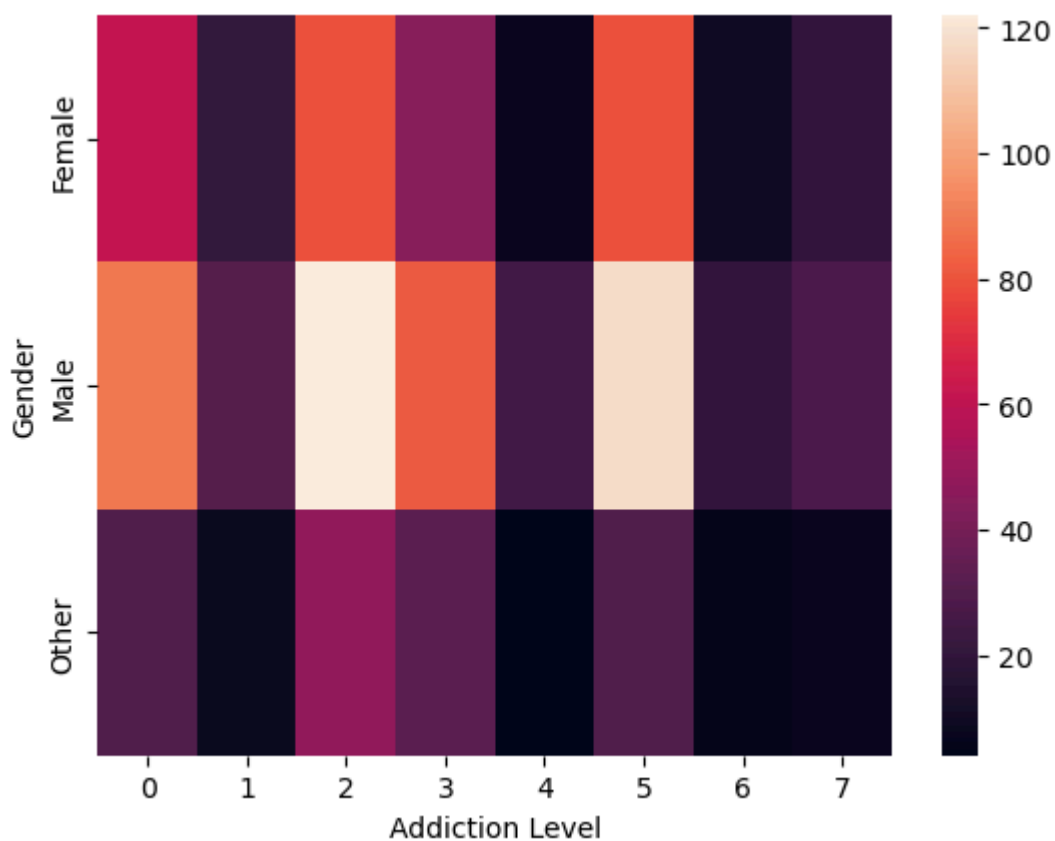
```
In [13]: 1 df['Gender'].value_counts().plot(kind='pie', autopct='%0.1f%%')
```

```
Out[13]: <Axes: ylabel='count'>
```



```
In [14]: 1 sns.heatmap(pd.crosstab(df['Gender'], df['Addiction Level']))
```

```
Out[14]: <Axes: xlabel='Addiction Level', ylabel='Gender'>
```



In [15]: 1 df.groupby('Addiction Level')['Gender'].value_counts()

Out[15]:

Addiction Level	Gender	
0	Male	89
	Female	61
	Other	30
1	Male	31
	Female	21
	Other	8
2	Male	122
	Female	79
	Other	47
3	Male	81
	Female	45
	Other	33
4	Male	25
	Female	7
	Other	4
5	Male	118
	Female	80
	Other	30
6	Male	20
	Female	9
	Other	5
7	Male	28
	Female	20
	Other	7

Name: count, dtype: int64

In [16]: 1 (pd.crosstab(df['Gender'],df['Addiction Level'],normalize='columns')*100)

Out[16]:

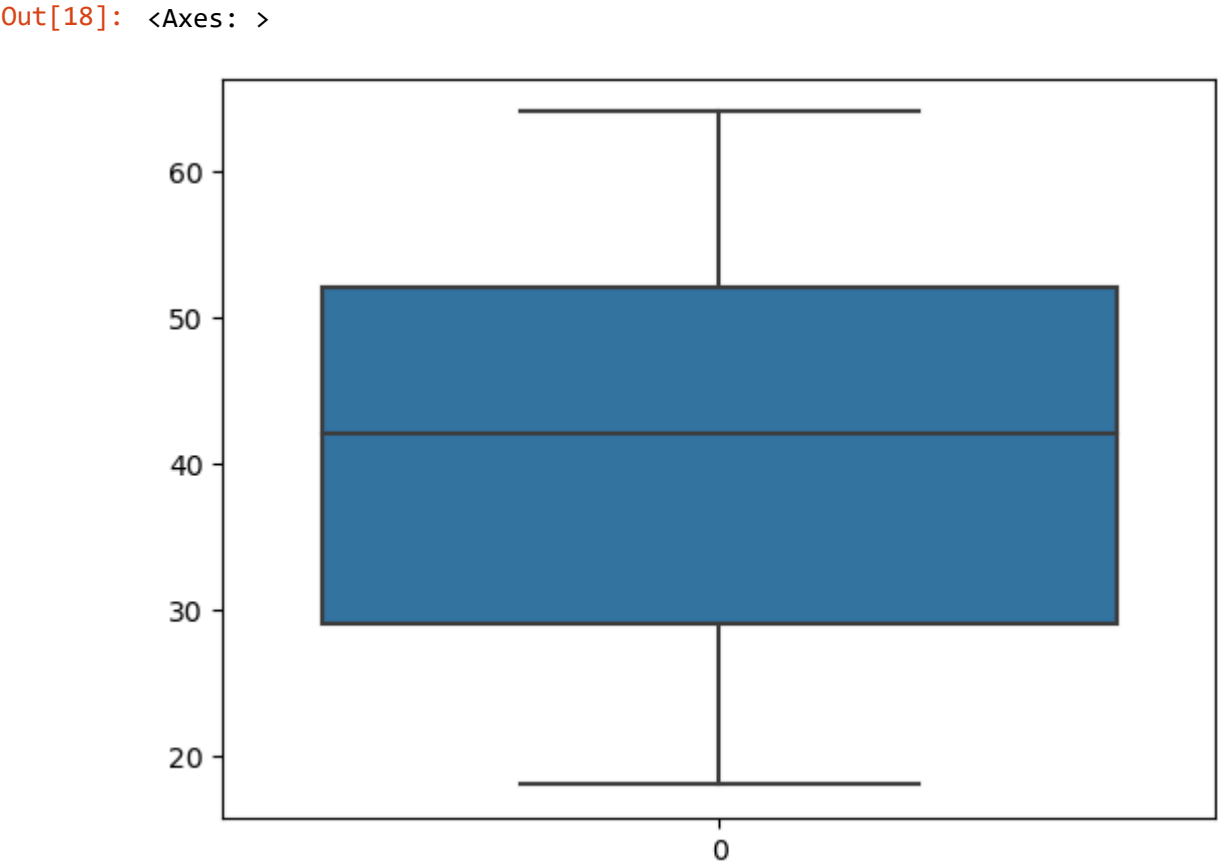
Addiction Level	0	1	2	3	4	5	6	7
Gender								
Female	33.888889	35.000000	31.854839	28.301887	19.444444	35.087719	26.470588	36.111111
Male	49.444444	51.666667	49.193548	50.943396	69.444444	51.754386	58.823529	50.000000
Other	16.666667	13.333333	18.951613	20.754717	11.111111	13.157895	14.705882	12.222222

```
In [17]: 1 df.head()
```

Out[17]:

	UserID	Age	Gender	Location	Income	Debt	Owns Property	Profession	Demographics	Pla
0	1	56	Male	Pakistan	82812	True	True	Engineer	Rural	Inst
1	2	46	Female	Mexico	27999	False	True	Artist	Urban	Inst
2	3	32	Female	United States	42436	False	True	Engineer	Rural	Fac
3	4	60	Male	Barzil	62963	True	False	Waiting staff	Rural	Yo
4	5	25	Male	Pakistan	22096	False	True	Manager	Urban	

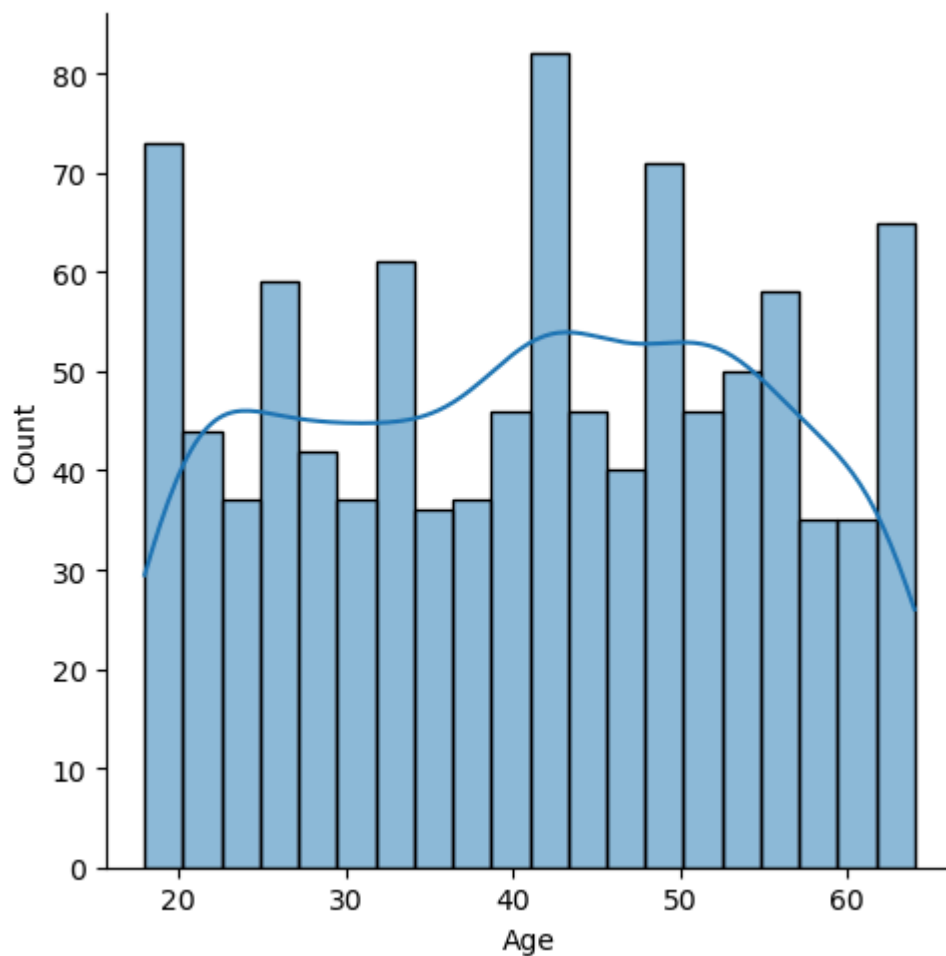
```
In [18]: 1 sns.boxplot(df['Age'])
```




```
In [19]: 1 sns.displot(kind='hist',data=df,x='Age',kde=True,bins=20)
```

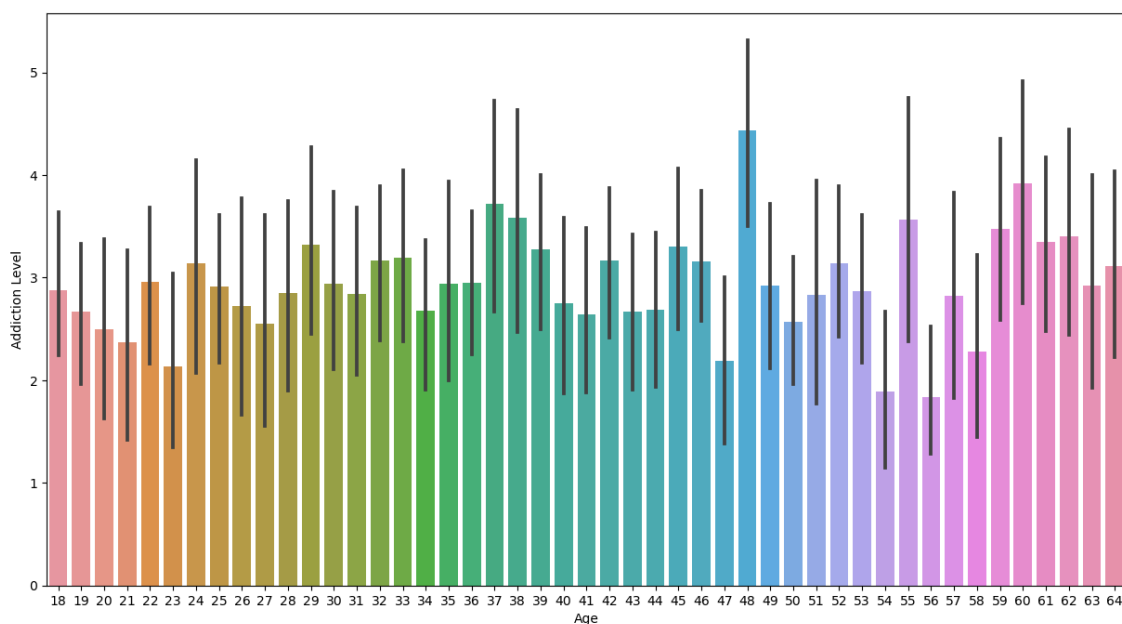
C:\Users\TIRTH PATEL\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)

Out[19]: <seaborn.axisgrid.FacetGrid at 0x282a80e9d10>



```
In [20]: 1 plt.figure(figsize=(15,8))
        2 sns.barplot(data=df,y='Addiction Level',x='Age')
```

```
Out[20]: <Axes: xlabel='Age', ylabel='Addiction Level'>
```

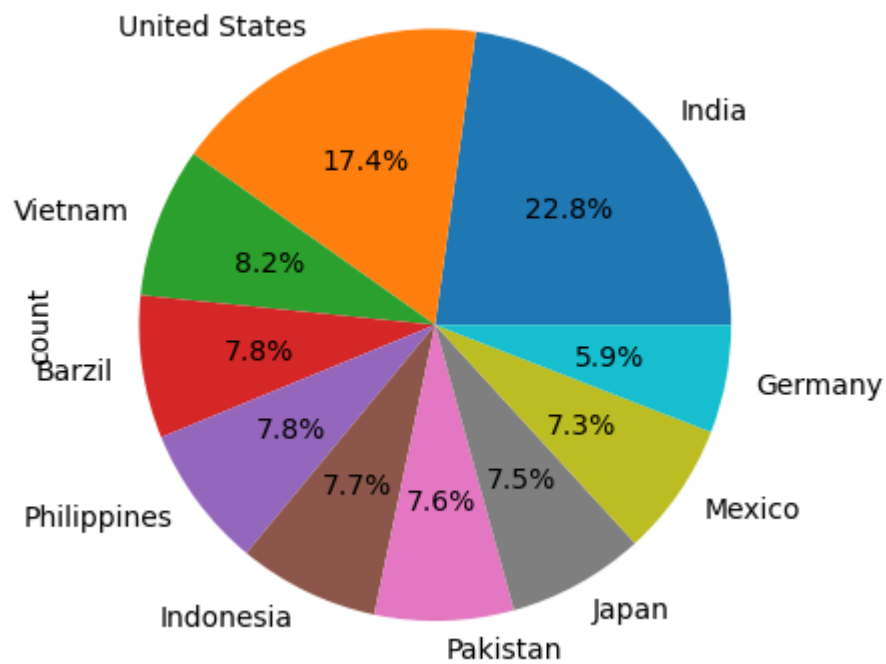


```
In [21]: 1 df['Location'].value_counts()
```

```
Out[21]: Location
India                228
United States        174
Vietnam              82
Barzil               78
Philippines          78
Indonesia            77
Pakistan             76
Japan                75
Mexico              73
Germany             59
Name: count, dtype: int64
```

```
In [22]: 1 df['Location'].value_counts().plot(kind='pie', autopct='%0.1f%%')
```

```
Out[22]: <Axes: ylabel='count'>
```



```
In [23]: 1 df.groupby('Addiction Level')['Location'].value_counts()
```

```

Out[23]:
Addiction Level  Location
0               India      34
               United States 29
               Vietnam     22
               Pakistan     19
               Indonesia    17
               Mexico       15
               Japan        14
               Philippines   12
               Germany      12
               Barzil        6
1               India      17
               United States 12
               Japan         9
               Vietnam       4
               Mexico        4
               Barzil        3
               Indonesia     3
               Pakistan      3
               Philippines    3
               Germany        2
2               India      60
               United States 43
               Japan        24
               Philippines   24
               Indonesia     22
               Pakistan      18
               Mexico        17
               Barzil        16
               Vietnam       16
               Germany        8
3               India      37
               United States 27
               Barzil        17
               Philippines    15
               Vietnam       14
               Germany        13
               Pakistan      11
               Mexico        10
               Indonesia     8
               Japan         7
4               India      10
               United States 8
               Pakistan      5
               Vietnam       3
               Mexico        3
               Barzil        2
               Germany        2
               Indonesia     2
               Philippines    1
5               India      50
               United States 34
               Barzil        29
               Pakistan      19
               Indonesia     18
               Vietnam       17
               Mexico        16
               Japan        16
               Germany        15
               Philippines    14
6               India      9

```

```

United States    4
Germany          4
Indonesia        4
Mexico           3
Philippines      3
Barzil           3
Vietnam          2
Japan            2
7 United States   17
India            11
Philippines      6
Mexico           5
Vietnam          4
Japan            3
Indonesia        3
Germany          3
Barzil           2
Pakistan         1

```

Name: count, dtype: int64

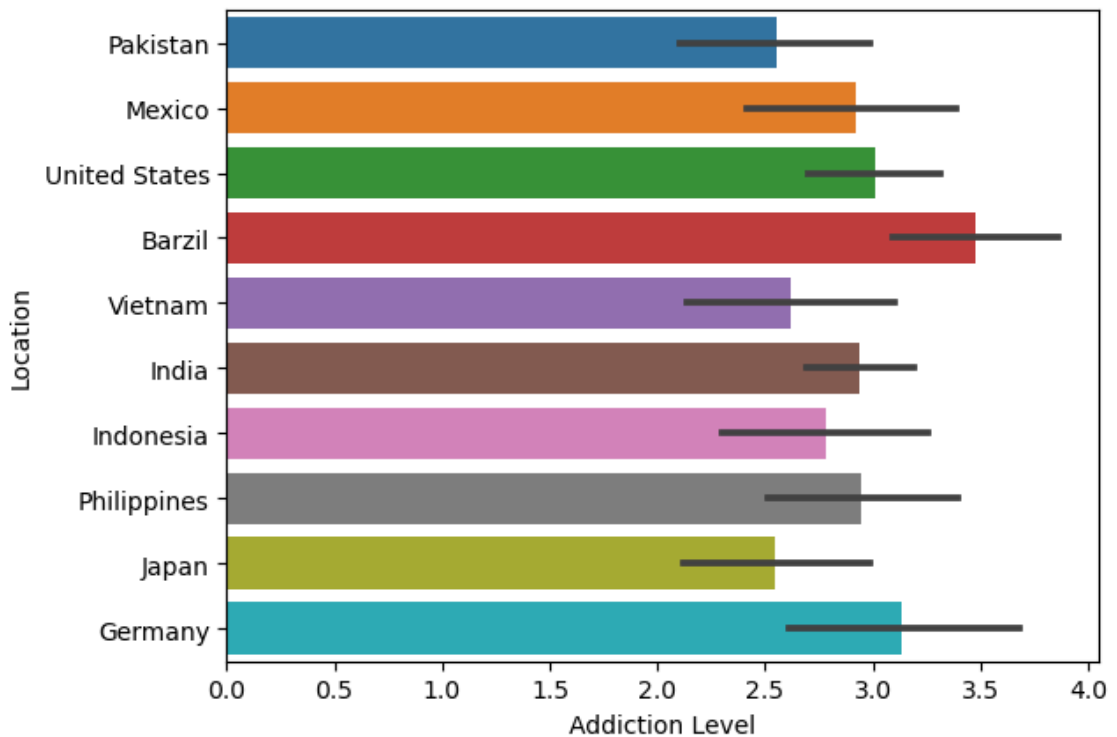
In [24]: 1 (pd.crosstab(df['Location'],df['Addiction Level'],normalize='columns'))

Out[24]:

Addiction Level	0	1	2	3	4	5	6
Location							
Barzil	3.333333	5.000000	6.451613	10.691824	5.555556	12.719298	8.823529
Germany	6.666667	3.333333	3.225806	8.176101	5.555556	6.578947	11.764706
India	18.888889	28.333333	24.193548	23.270440	27.777778	21.929825	26.470588
Indonesia	9.444444	5.000000	8.870968	5.031447	5.555556	7.894737	11.764706
Japan	7.777778	15.000000	9.677419	4.402516	0.000000	7.017544	5.882353
Mexico	8.333333	6.666667	6.854839	6.289308	8.333333	7.017544	8.823529
Pakistan	10.555556	5.000000	7.258065	6.918239	13.888889	8.333333	0.000000
Philippines	6.666667	5.000000	9.677419	9.433962	2.777778	6.140351	8.823529
United States	16.111111	20.000000	17.338710	16.981132	22.222222	14.912281	11.764706
Vietnam	12.222222	6.666667	6.451613	8.805031	8.333333	7.456140	5.882353

```
In [25]: 1 sns.barplot(data=df,y='Location',x='Addiction Level')
```

```
Out[25]: <Axes: xlabel='Addiction Level', ylabel='Location'>
```



```
In [26]: 1 df['Debt'].value_counts()
```

```
Out[26]: Debt
True      599
False     401
Name: count, dtype: int64
```

```
In [27]: 1 df.groupby('Addiction Level')['Debt'].value_counts()
```

```
Out[27]: Addiction Level  Debt
0                        True    116
                        False     64
1                        True     42
                        False     18
2                        True    144
                        False    104
3                        True     96
                        False     63
4                        False    18
                        True     18
5                        True    130
                        False     98
6                        True     23
                        False     11
7                        True     30
                        False     25
Name: count, dtype: int64
```

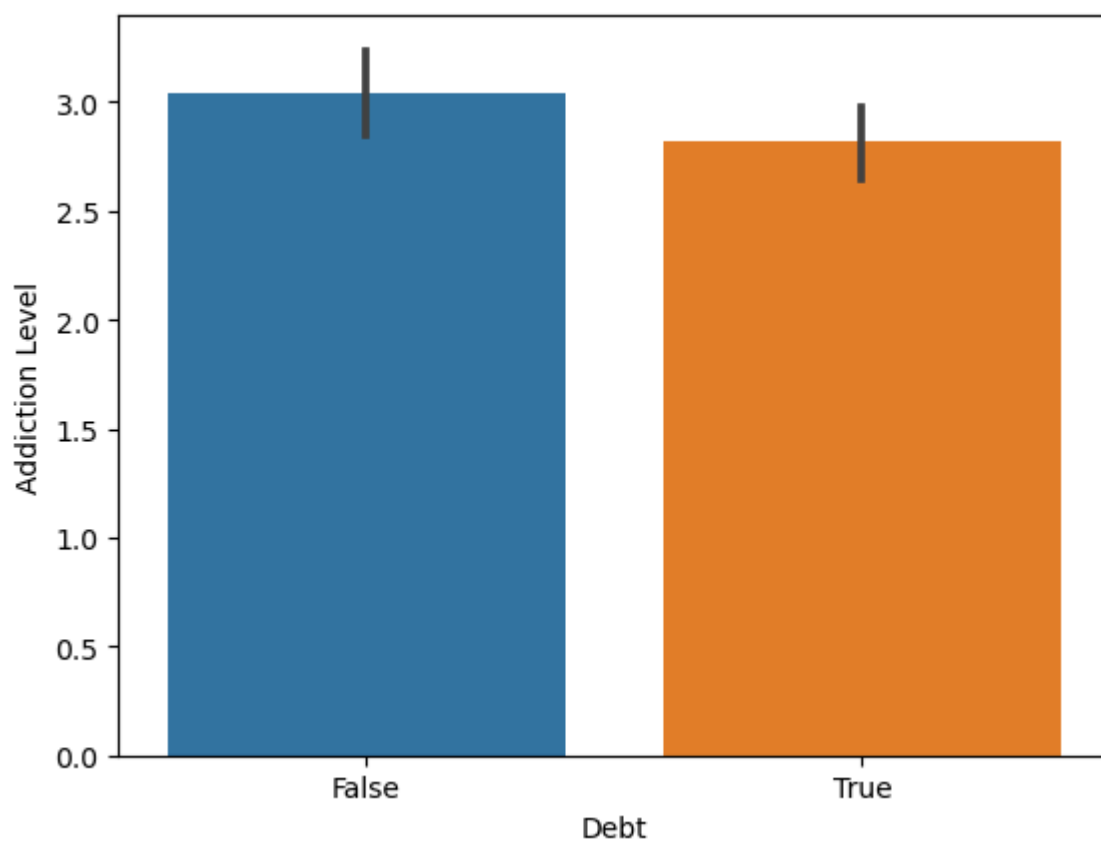
```
In [28]: 1 (pd.crosstab(df['Debt'],df['Addiction Level'],normalize='columns')*100
```

```
Out[28]:
```

	Addiction Level	0	1	2	3	4	5	6	7
Debt									
False		35.555556	30.0	41.935484	39.622642	50.0	42.982456	32.352941	45.454545
True		64.444444	70.0	58.064516	60.377358	50.0	57.017544	67.647059	54.545455

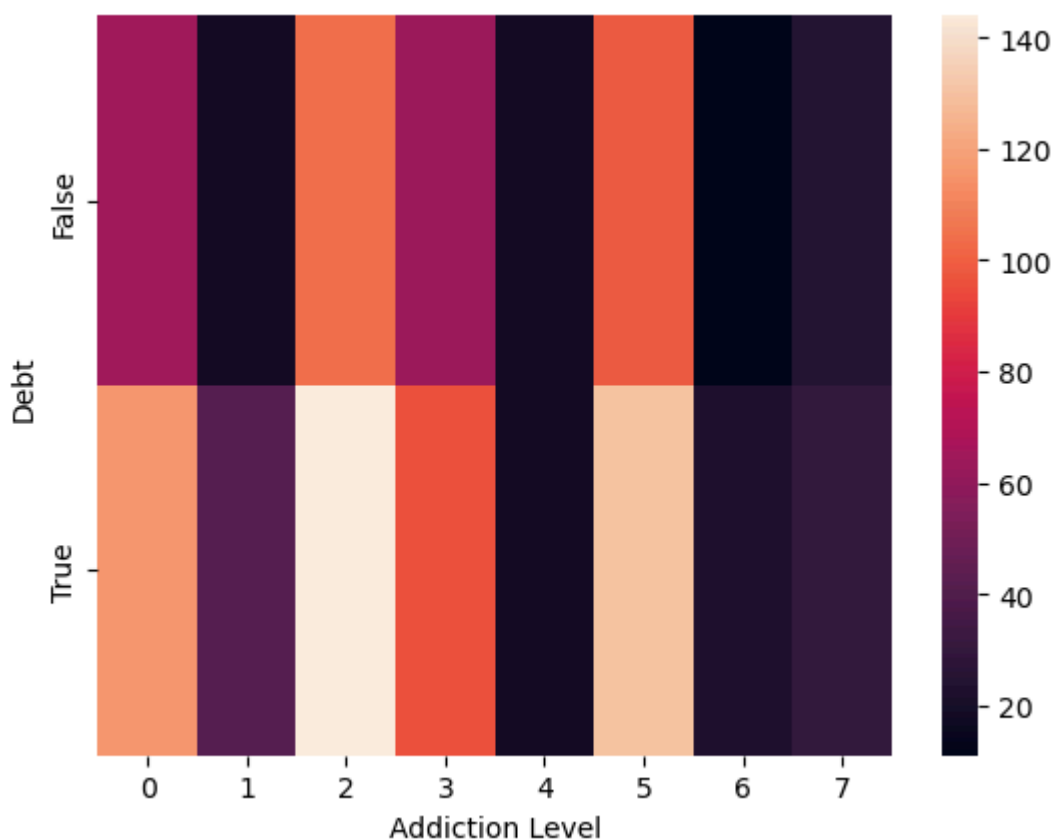
```
In [29]: 1 sns.barplot(data=df,x='Debt',y='Addiction Level')
```

```
Out[29]: <Axes: xlabel='Debt', ylabel='Addiction Level'>
```




```
In [30]: 1 sns.heatmap(pd.crosstab(df['Debt'],df['Addiction Level']))
```

```
Out[30]: <Axes: xlabel='Addiction Level', ylabel='Debt'>
```



```
In [31]: 1 df.head()
```

```
Out[31]:
```

	UserID	Age	Gender	Location	Income	Debt	Owns Property	Profession	Demographics	Pla
0	1	56	Male	Pakistan	82812	True	True	Engineer	Rural	Inst
1	2	46	Female	Mexico	27999	False	True	Artist	Urban	Inst
2	3	32	Female	United States	42436	False	True	Engineer	Rural	Fac
3	4	60	Male	Barzil	62963	True	False	Waiting staff	Rural	Yo
4	5	25	Male	Pakistan	22096	False	True	Manager	Urban	

```
In [32]: 1 df['Owns Property'].value_counts()
```

```
Out[32]: Owns Property
True      542
False     458
Name: count, dtype: int64
```

```
In [33]: 1 df.groupby('Addiction Level')['Owns Property'].value_counts()
```

```
Out[33]:
```

Addiction Level	Owns Property	
0	True	115
	False	65
1	False	31
	True	29
2	False	128
	True	120
3	True	96
	False	63
4	True	19
	False	17
5	False	116
	True	112
6	True	20
	False	14
7	True	31
	False	24

Name: count, dtype: int64

```
In [34]: 1 df['Profession'].value_counts()
```

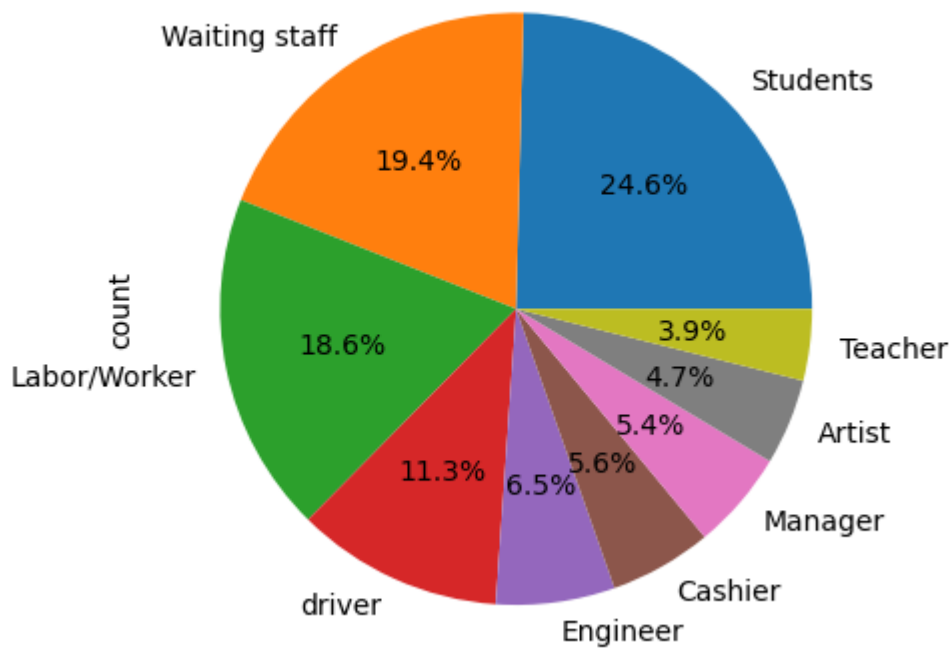
```
Out[34]:
```

Profession	
Students	246
Waiting staff	194
Labor/Worker	186
driver	113
Engineer	65
Cashier	56
Manager	54
Artist	47
Teacher	39

Name: count, dtype: int64

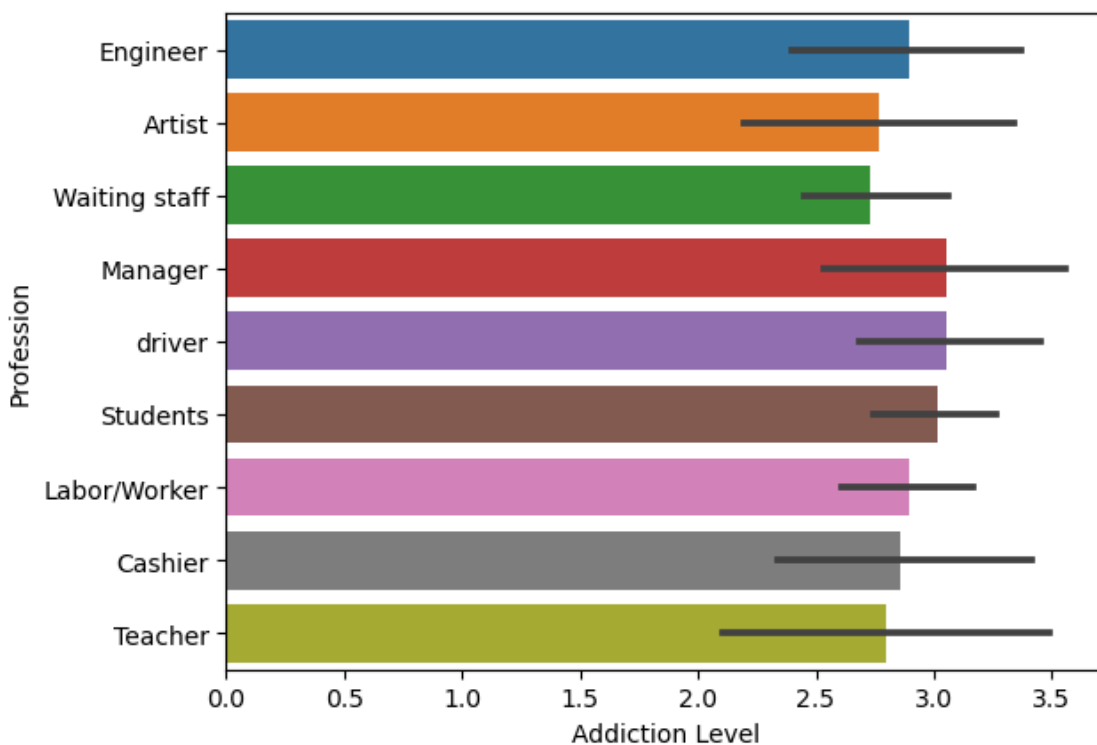
```
In [35]: 1 df['Profession'].value_counts().plot(kind='pie', autopct='%0.1f%%')
```

```
Out[35]: <Axes: ylabel='count'>
```



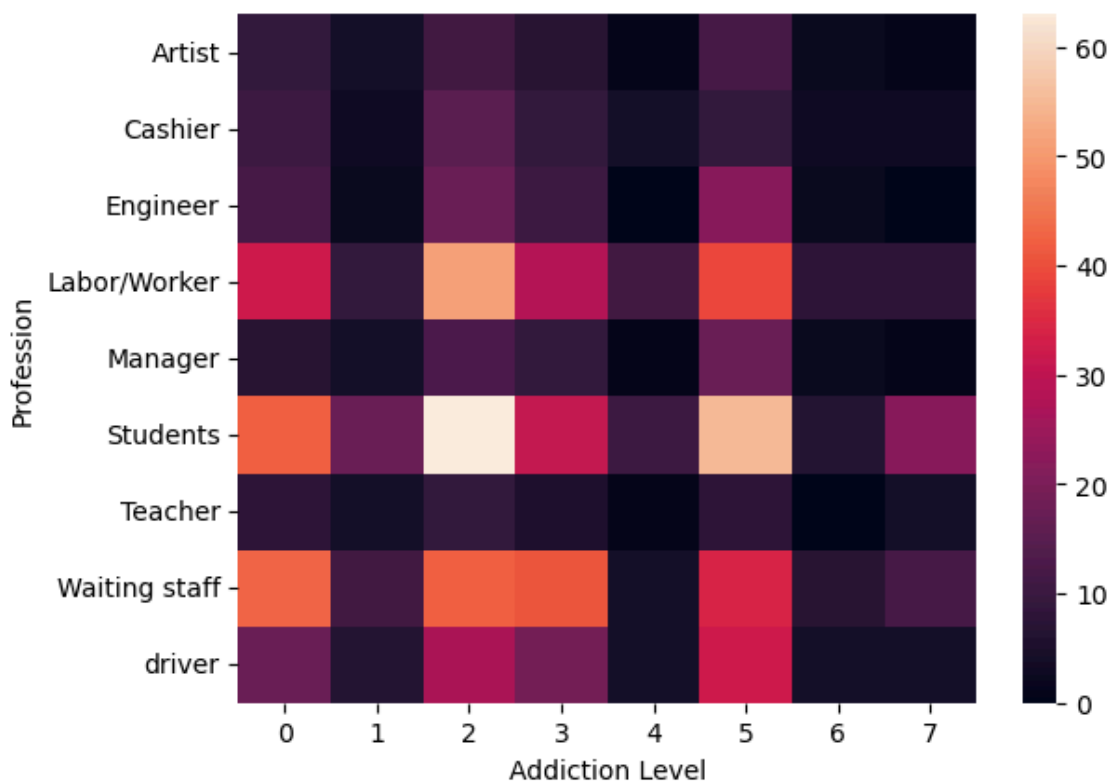
```
In [36]: 1 sns.barplot(data=df, y='Profession', x='Addiction Level')
```

```
Out[36]: <Axes: xlabel='Addiction Level', ylabel='Profession'>
```



```
In [37]: 1 sns.heatmap(pd.crosstab(df['Profession'],df['Addiction Level']))
```

Out[37]: <Axes: xlabel='Addiction Level', ylabel='Profession'>



```
In [38]: 1 df.head()
```

Out[38]:

	UserID	Age	Gender	Location	Income	Debt	Owns Property	Profession	Demographics	Pla
0	1	56	Male	Pakistan	82812	True	True	Engineer	Rural	Inst
1	2	46	Female	Mexico	27999	False	True	Artist	Urban	Inst
2	3	32	Female	United States	42436	False	True	Engineer	Rural	Fac
3	4	60	Male	Barzil	62963	True	False	Waiting staff	Rural	Yo
4	5	25	Male	Pakistan	22096	False	True	Manager	Urban	.

```
In [44]: 1 x = temp_df.drop(columns=['Satisfaction'])
```

```
In [130]: 1 df['Addiction Level'] = x['Addiction Level'].replace({0: 'low', 1: 'lo
2                                     2: 'moderate', 3:
3                                     5: 'high', 6: 'hi
4
```

```
In [65]: 1 x['Addiction Level'] = x['Addiction Level'].replace({ 'low':0, 'low':  
2 'moderate':2, '  
3 'high':5, 'high'
```

```
In [45]: 1 x.corr()['Addiction Level'].sort_values(ascending=False)
```

```
Out[45]: Addiction Level      1.000000  
Age      0.033493  
Engagement      0.027620  
Video ID      0.020655  
Importance Score      0.018474  
Total Time Spent      0.016086  
Number of Videos Watched      0.013286  
Scroll Rate      0.006758  
Video Length      0.004914  
Time Spent On Video      -0.000447  
Income      -0.039181  
Number of Sessions      -0.080961  
UserID      -0.105280  
ProductivityLoss      -0.994939  
Self Control      -1.000000  
Name: Addiction Level, dtype: float64
```

```
In [57]: 1 df['Addiction Level'].value_counts()
```

```
Out[57]: Addiction Level  
moderate      443  
high      317  
low      240  
Name: count, dtype: int64
```

```
In [58]: 1 df
```

```
Out[58]:
```

	UserID	Age	Gender	Location	Income	Debt	Owns Property	Profession	Demograp
0	1	56	Male	Pakistan	82812	True	True	Engineer	F
1	2	46	Female	Mexico	27999	False	True	Artist	U
2	3	32	Female	United States	42436	False	True	Engineer	F
3	4	60	Male	Barzil	62963	True	False	Waiting staff	F
4	5	25	Male	Pakistan	22096	False	True	Manager	U
5	6	38	Male	Vietnam	45279	False	False	driver	U

```
In [59]: 1 Temp_df = df.copy()
```

```
In [107]: 1 x['Owns Property'] = Temp_df['Owns Property'].replace({'False': 0, 'True': 1})
```

```
In [63]: 1 x['Gender'] = Temp_df['Gender']
```

```
In [108]: 1 x.corr()['Addiction Level'].sort_values(ascending=False)
```

```
Out[108]: Addiction Level      1.000000
Frequency      0.183081
Gender         0.027201
Engagement     0.021253
Age            0.020976
Total Time Spent 0.018875
Importance Score 0.018299
Video Length    0.016800
Scroll Rate     0.009297
Time Spent On Video 0.008888
Video ID       0.008590
Number of Videos Watched -0.001642
Platform       -0.004536
Demographics   -0.004821
Income         -0.049396
Debt           -0.058059
Owns Property  -0.061467
Number of Sessions -0.072394
UserID         -0.175041
ProductivityLoss -0.943362
Self Control   -0.946244
Name: Addiction Level, dtype: float64
```

```
In [68]: 1 df
```

```
Out[68]:
```

	UserID	Age	Gender	Location	Income	Debt	Owns Property	Profession	Demographics
0	1	56	Male	Pakistan	82812	True	True	Engineer	F
1	2	46	Female	Mexico	27999	False	True	Artist	U
2	3	32	Female	United States	42436	False	True	Engineer	F
3	4	60	Male	Barzil	62963	True	False	Waiting staff	F
4	5	25	Male	Pakistan	22096	False	True	Manager	U
5	6	38	Male	Vietnam	45279	False	False	driver	U

remove : Demographics,Platform,CurrentActivity,ConnectionType,Video ID,UserID,Importance Score,Scroll Rate

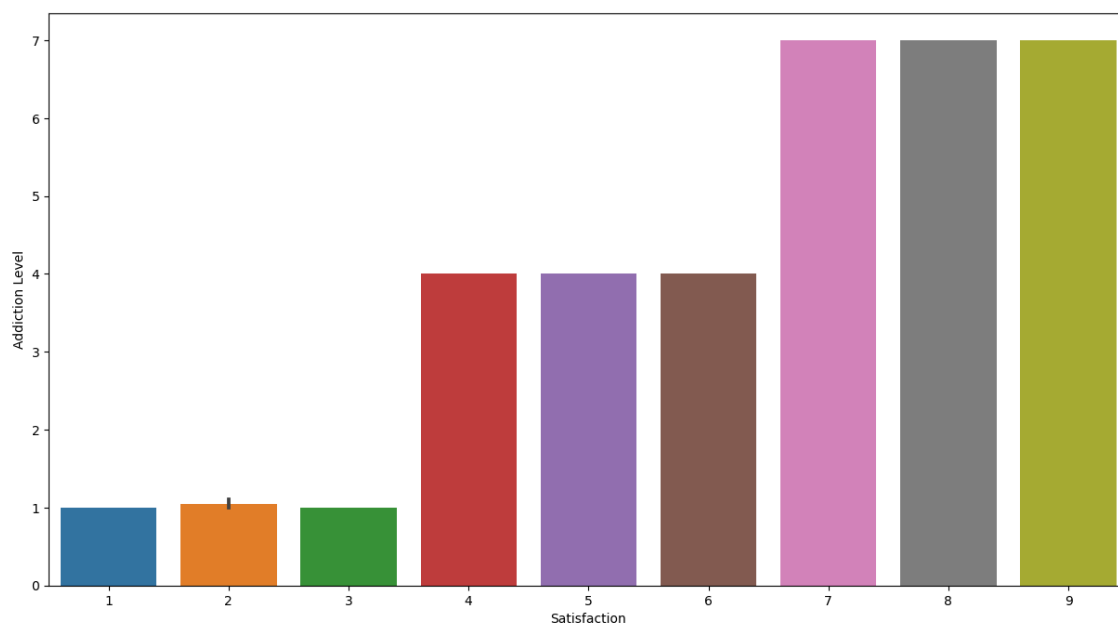
```
In [111]: 1 df['OS'].value_counts()
```

```
Out[111]: OS
Android    500
iOS        261
Windows    123
MacOS      116
Name: count, dtype: int64
```

```
In [73]: 1 Temp_df['Addiction Level'] = Temp_df['Addiction Level'].replace({ 'low
2                                         'moderate':2, '
3                                         'high':5, 'high
```

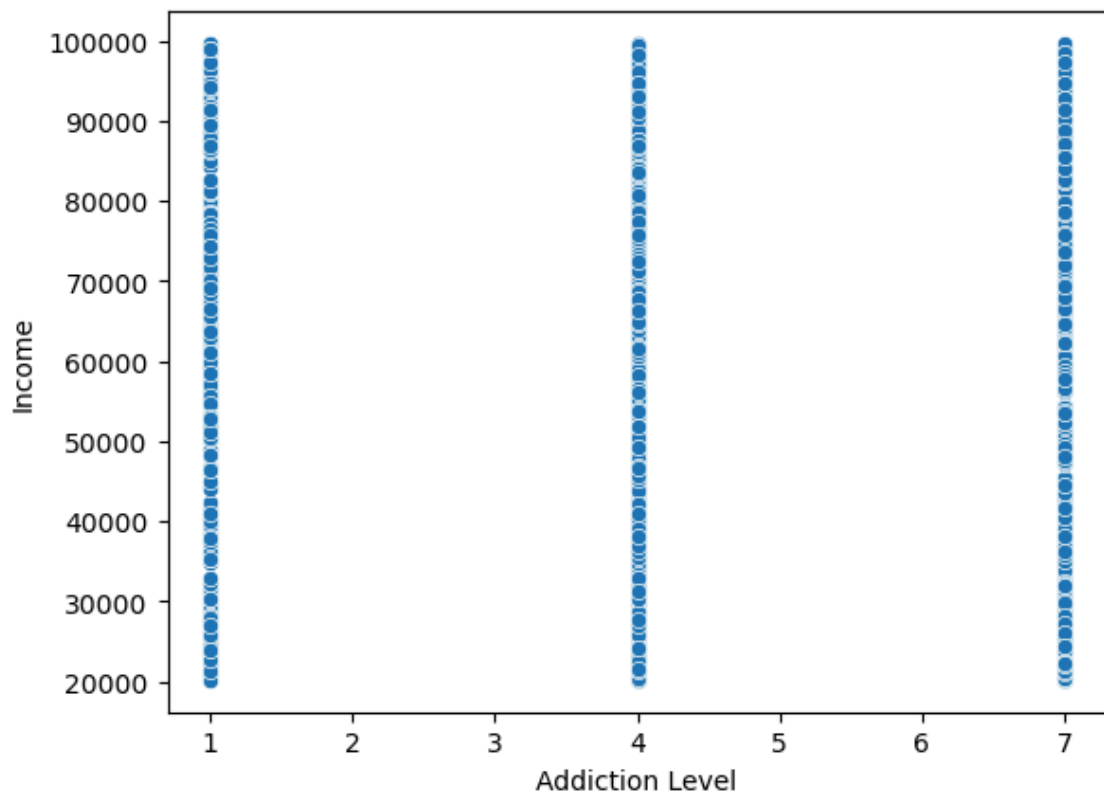
```
In [124]: 1 plt.figure(figsize=(15,8))
2 sns.barplot(data=Temp_df,x='Satisfaction',y='Addiction Level')
```

```
Out[124]: <Axes: xlabel='Satisfaction', ylabel='Addiction Level'>
```



```
In [117]: 1 sns.scatterplot(data=Temp_df, y = 'Income', x = 'Addiction Level')
```

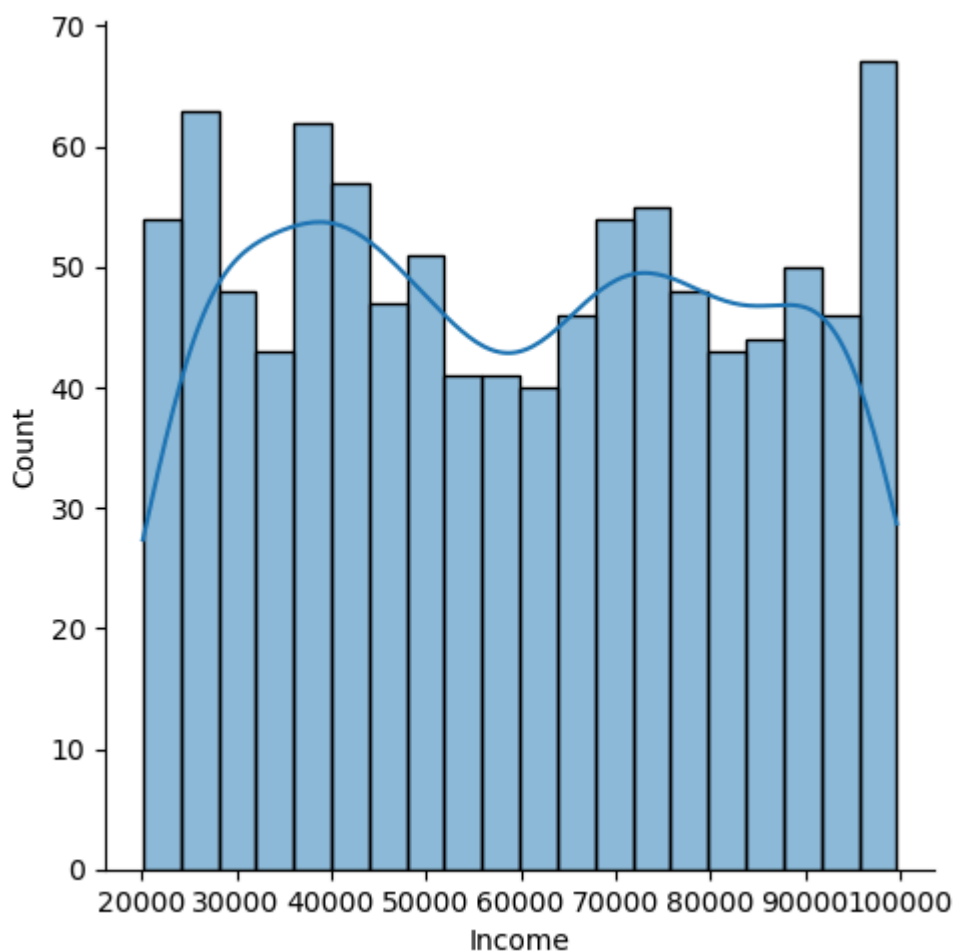
```
Out[117]: <Axes: xlabel='Addiction Level', ylabel='Income'>
```




```
In [118]: 1 sns.displot(kind='hist',data=df,x='Income',kde=True,bins=20)
```

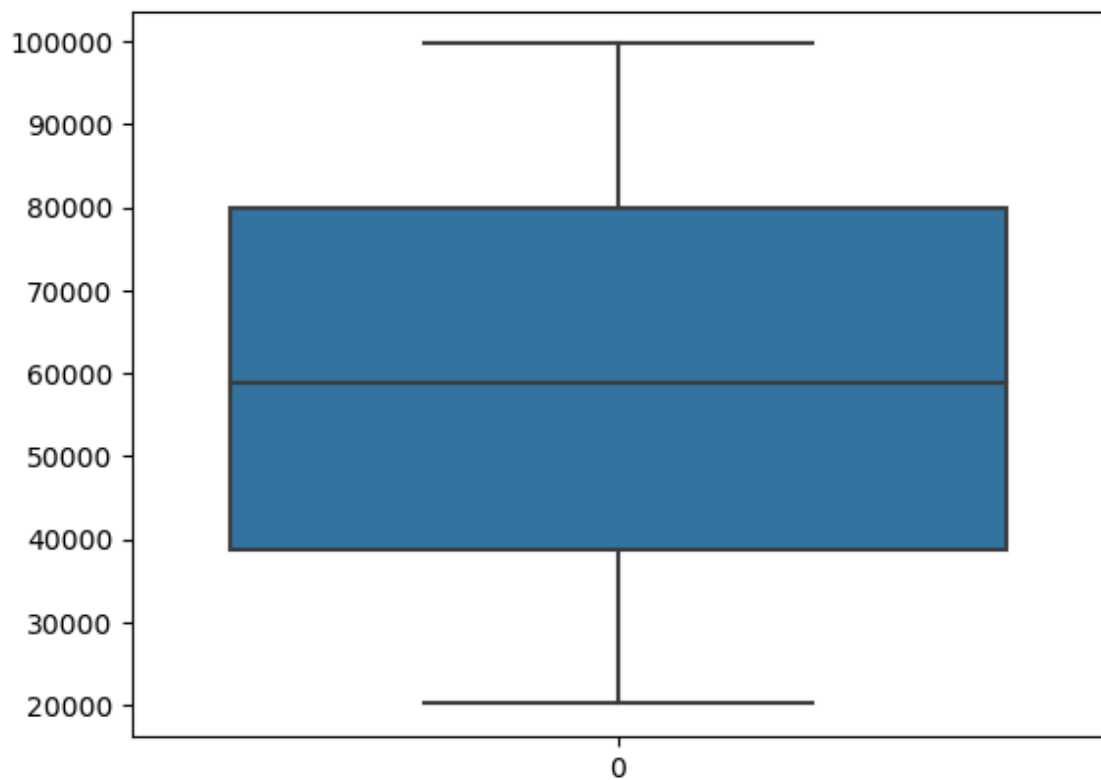
C:\Users\TIRTH PATEL\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)

```
Out[118]: <seaborn.axisgrid.FacetGrid at 0x282b7126750>
```



```
In [119]: 1 sns.boxplot(df['Income'])
```

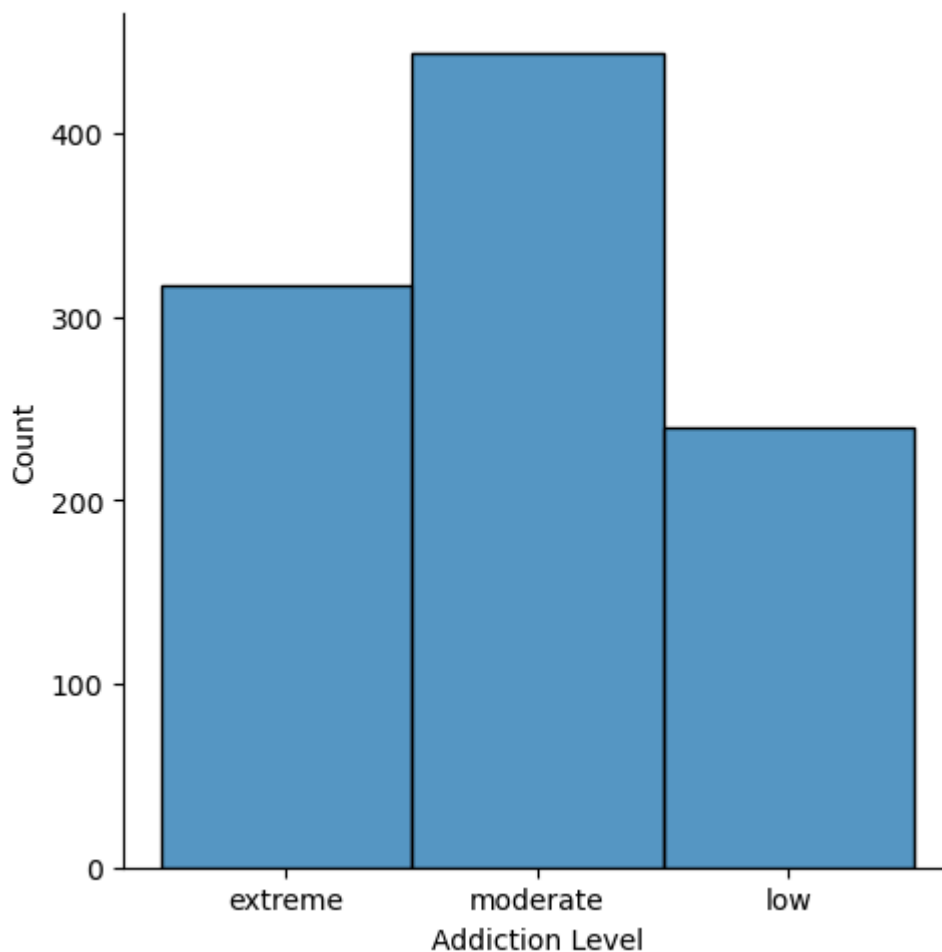
```
Out[119]: <Axes: >
```



```
In [132]: 1 sns.displot(kind='hist',data=df,x='Addiction Level')
```

C:\Users\TIRTH PATEL\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)

```
Out[132]: <seaborn.axisgrid.FacetGrid at 0x282b9764ed0>
```



```
In [133]: 1 df['Addiction Level'].value_counts()
```

```
Out[133]: Addiction Level
moderate    443
extreme     317
low         240
Name: count, dtype: int64
```

```
In [ ]: 1
```

```
In [ ]: 1
```

```
In [ ]: 1
```

```
In [ ]: 1
```

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1