

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/ae
```

data

	Product	Age	Gender	Education	MaritalStatus
0	KP281	18	Male	14	Single
1	KP281	19	Male	15	Single
2	KP281	19	Female	14	Partnered
3	KP281	19	Male	12	Single
4	KP281	20	Male	13	Partnered
...
175	KP781	40	Male	21	Single
176	KP781	42	Male	18	Single
177	KP781	45	Male	16	Single
178	KP781	47	Male	18	Partnered
179	KP781	48	Male	18	Partnered

180 rows × 9 columns

```
data.shape
```

(180, 9)

```
data.describe()
```

	Age	Education	Usage	Fitness
count	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111
std	6.943498	1.617055	1.084797	0.958869
min	18.000000	12.000000	2.000000	1.000000
25%	24.000000	14.000000	3.000000	3.000000
50%	26.000000	16.000000	3.000000	3.000000
75%	33.000000	16.000000	4.000000	4.000000
max	50.000000	21.000000	7.000000	5.000000

```
data.columns
```



Palak Vyas
7:50 PM Today



-There are 180 records in data.

```
Index(['Product', 'Age', 'Gender', 'Education',
      'MaritalStatus', 'Usage',
      'Fitness', 'Income', 'Miles'],
      dtype='object')
```

```
data.keys()
```

```
Index(['Product', 'Age', 'Gender', 'Education',
      'MaritalStatus', 'Usage',
      'Fitness', 'Income', 'Miles'],
      dtype='object')
```

```
data['Income'].unique()
```

```
array([ 29562,  31836,  30699,  32973,  35247,
        37521,  36384,  38658,
         40932,  34110,  39795,  42069,  44343,
        45480,  46617,  48891,
         53439,  43206,  52302,  51165,  50028,
        54576,  68220,  55713,
         60261,  67083,  56850,  59124,  61398,
        57987,  64809,  47754,
         65220,  62535,  48658,  54781,  48556,
        58516,  53536,  61006,
         57271,  52291,  49801,  62251,  64741,
        70966,  75946,  74701,
         69721,  83416,  88396,  90886,  92131,
        77191,  52290,  85906,
         103336,  99601,  89641,  95866, 104581,
        95508])
```

```
data['Income'].value_counts()
```

```
45480    14
52302     9
46617     8
54576     8
53439     8
..
65220     1
55713     1
68220     1
30699     1
95508     1
Name: Income, Length: 62, dtype: int64
```

```
data.index.values
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,
        9, 10, 11, 12,
        13, 14, 15, 16, 17, 18, 19, 20, 21,
       22, 23, 24, 25,
        26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38,
        39, 40, 41, 42, 43, 44, 45, 46, 47,
       48, 49, 50, 51,
        52, 53, 54, 55, 56, 57, 58, 59, 60,
       61, 62, 63, 64,
        65, 66, 67, 68, 69, 70, 71, 72, 73,
       74, 75, 76, 77,
        78, 79, 80, 81, 82, 83, 84, 85, 86,
       87, 88, 89, 90])
```

```

91, 92, 93, 94, 95, 96, 97, 98, 99,
100, 101, 102, 103,
104, 105, 106, 107, 108, 109, 110, 111, 112,
113, 114, 115, 116,
117, 118, 119, 120, 121, 122, 123, 124, 125,
126, 127, 128, 129,
130, 131, 132, 133, 134, 135, 136, 137, 138,
139, 140, 141, 142,
143, 144, 145, 146, 147, 148, 149, 150, 151,
152, 153, 154, 155,
156, 157, 158, 159, 160, 161, 162, 163, 164,
165, 166, 167, 168,
169, 170, 171, 172, 173, 174, 175, 176, 177,
178, 179])

```

```

usage1=data['Usage'].dropna()
sns.distplot(usage1)
plt.show()

```

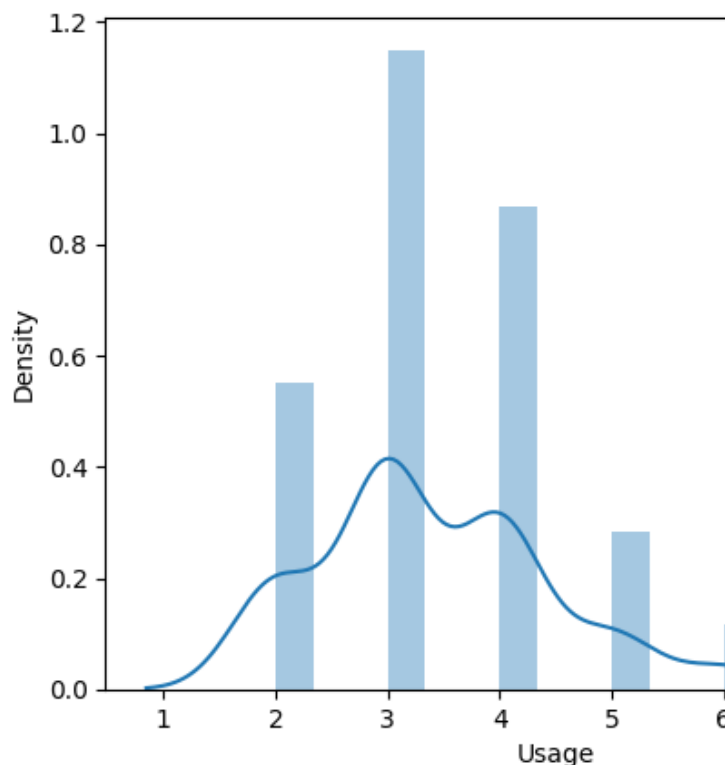
<ipython-input-70-f50d3091734f>:2: UserWarning:

`distplot` is a deprecated function and will be removed

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function that returns a histogram)

For a guide to updating your code to use the new functions, see <https://gist.github.com/mwaskom/de44147ed2974457ad63>.

```
sns.distplot(usage1)
```



Palak Vyas
6:11 PM Today



-Peaople with age group 20-30 are at max.

```

age1=data['Age'].dropna()
sns.distplot(age1)
plt.show()

```

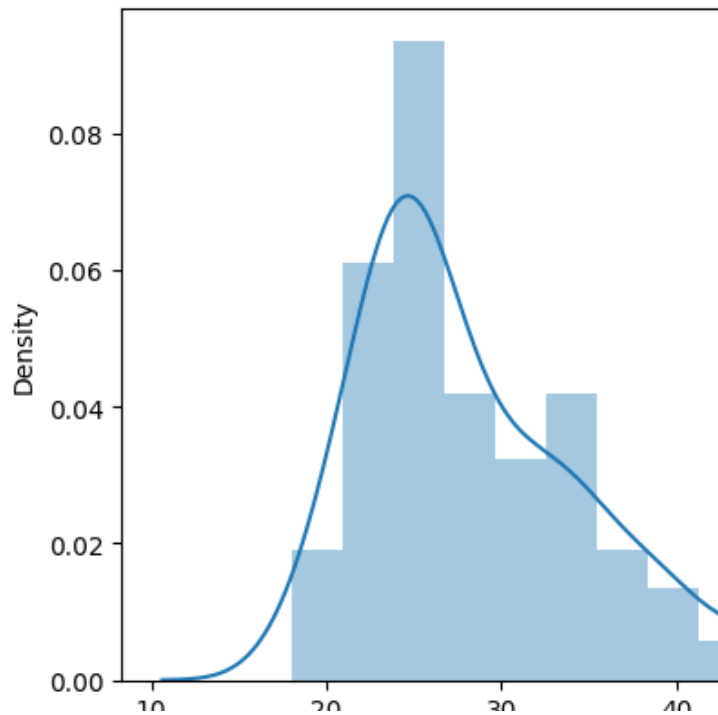
```
<ipython-input-71-21029d567d9e>:2: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in a future version.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function that returns the plot).

For a guide to updating your code to use the new functions, see <https://gist.github.com/mwaskom/de44147ed2974457ad63>.

```
sns.distplot(age1)
```



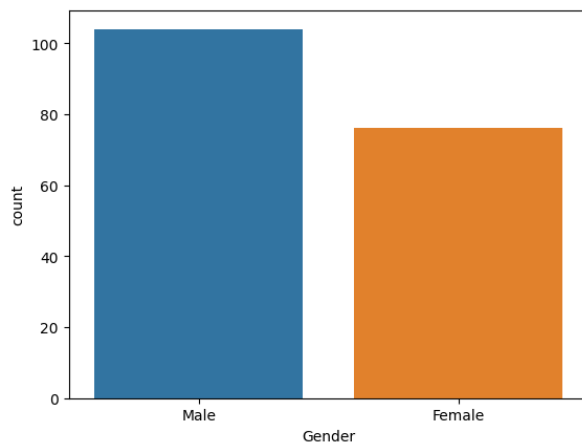
Palak Vyas

6:12 PM Today



-Male customers are more as compared with females.

```
sns.countplot(x="Gender", data=data)
plt.show()
```



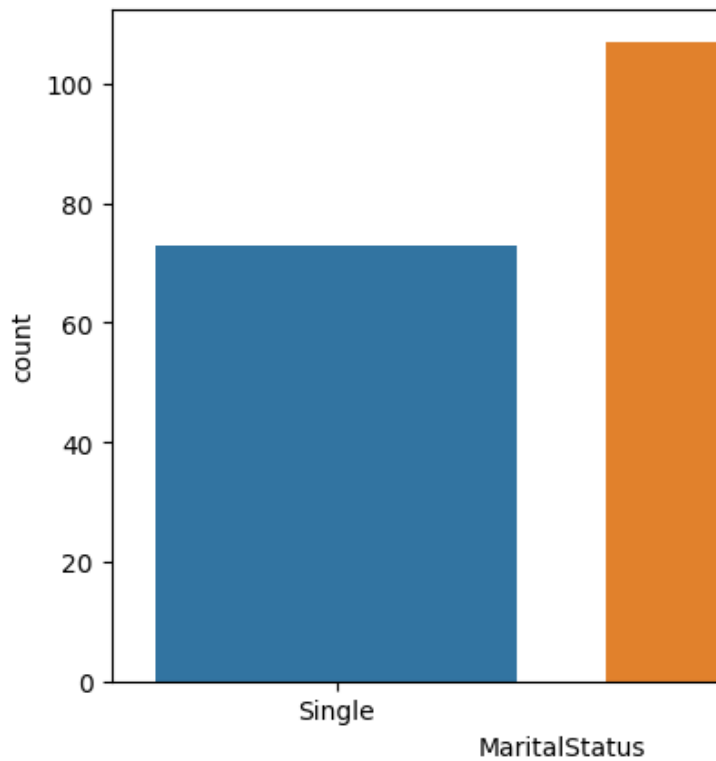
Palak Vyas

7:23 PM Today



-Partnered people are more tend to buy treadmills as compared to singles.

```
sns.countplot(x="MaritalStatus", data=data)  
plt.show()
```

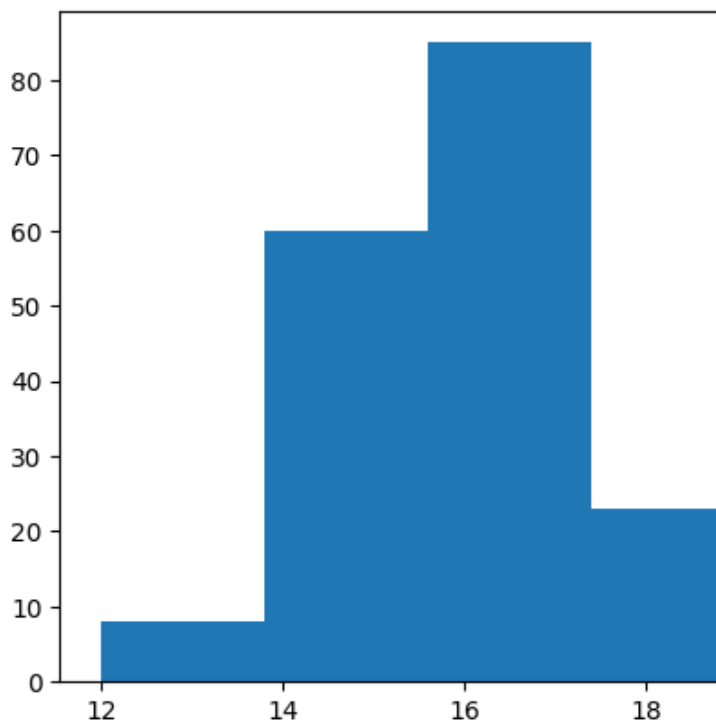


Palak Vyas
7:24 PM Today



-People with education level 16 are most tend to made purchase.

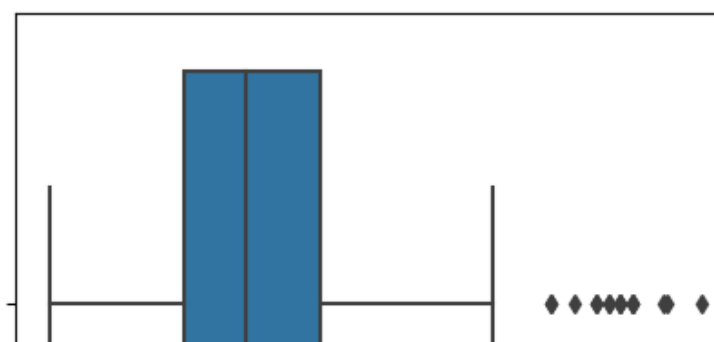
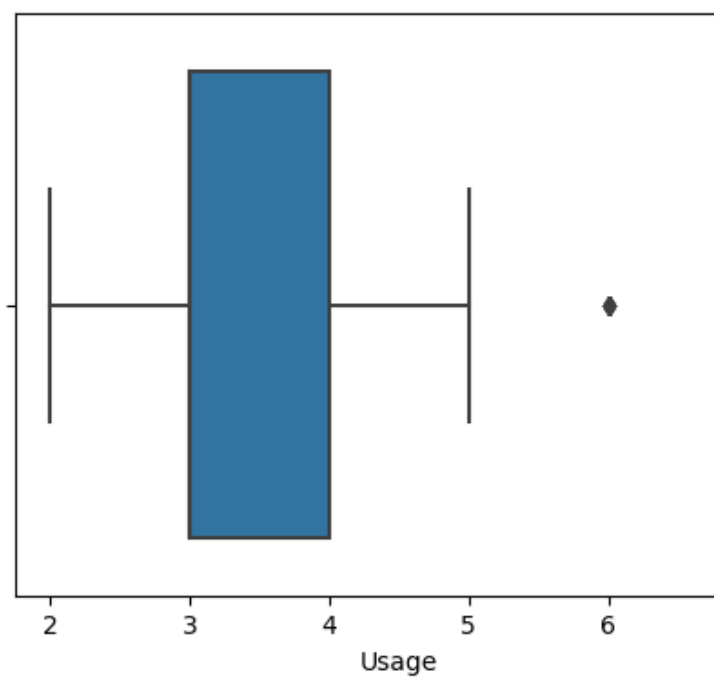
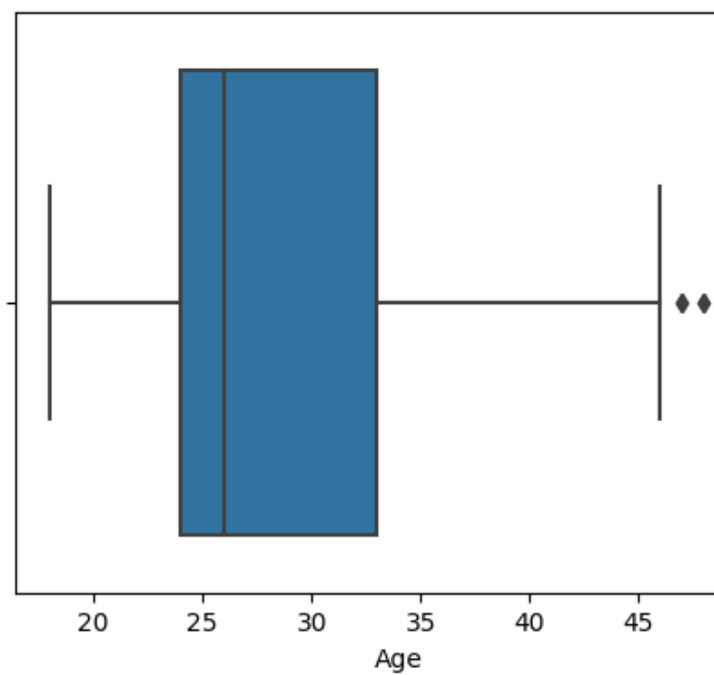
```
plt.hist(data['Education'],bins=5)  
plt.show()
```



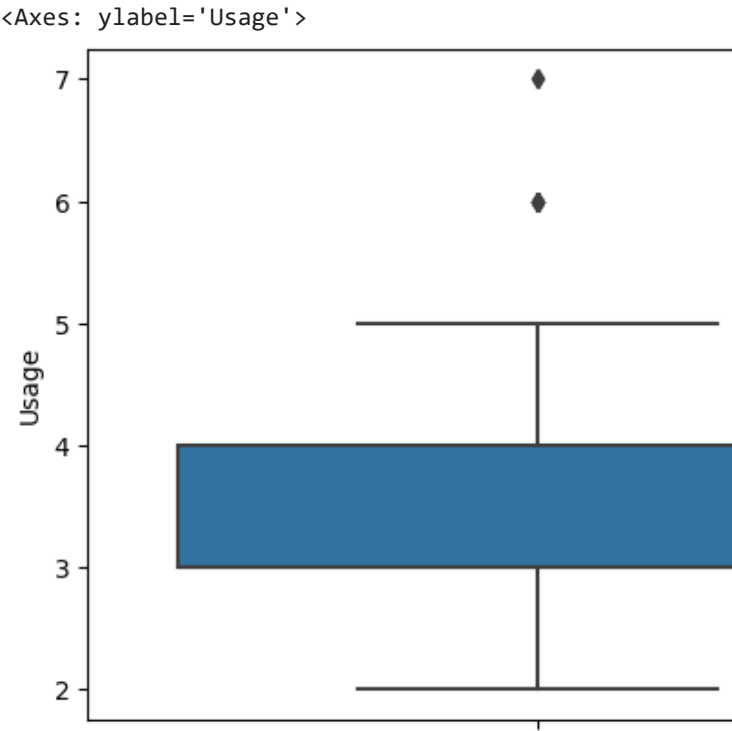
```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))  
fig.subplots_adjust(top=1.2)
```

```
sns.boxplot(data=data, x="Age", orient='h', ax=axis[0,0])  
sns.boxplot(data=data, x="Education", orient='h', ax=axis[0,1])  
sns.boxplot(data=data, x="Usage", orient='h', ax=axis[1,0])  
sns.boxplot(data=data, x="Fitness", orient='h', ax=axis[1,1])
```

```
sns.boxplot(data=data, x="Income", orient='h', ax=axis[2,0])  
sns.boxplot(data=data, x="Miles", orient='h', ax=axis[2,1])  
plt.show()
```



```
sns.boxplot(y=data["Usage"])
```



Palak Vyas
7:32 PM Today



Usage ,fitness are more correlated to miles.

```
data.corr()
```

<ipython-input-77-c44ded798807>:1: FutureWarning: The data.corr()

	Age	Education	Usage	Fitness	Miles
Age	1.000000	0.280496	0.015064	0.061105	0.036618
Education	0.280496	1.000000	0.395155	0.410581	0.307284
Usage	0.015064	0.395155	1.000000	0.668606	0.759130
Fitness	0.061105	0.410581	0.668606	1.000000	0.785702
Income	0.513414	0.625827	0.519537	0.535005	0.519537
Miles	0.036618	0.307284	0.759130	0.785702	1.000000

```
sns.heatmap(data.corr(),annot=True,cmap='Blues')
```

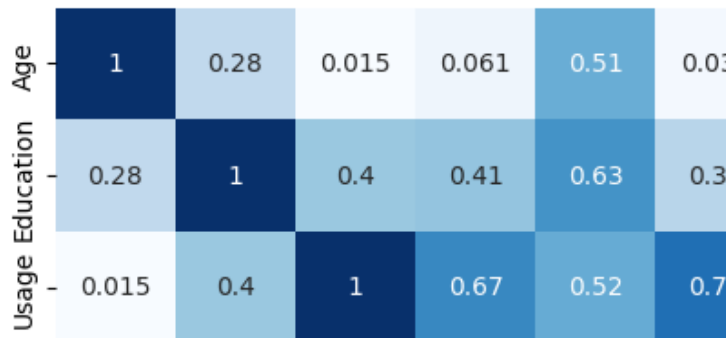


Palak Vyas
7:36 PM Today



Maximun uses of product KP281 are there in data and least uses of KP481 are found.

```
<ipython-input-78-215bc2d2a1fc>:1: FutureWarning: The
sns.heatmap(data.corr(),annot=True,cmap='Blues')
<Axes: >
```



```
df1 = data[['Product']].melt()
df1.groupby(['variable', 'value'])['value'].count()
```

		value	
variable	value		
Product	KP281	80	
	KP481	60	
	KP781	40	



Palak Vyas

7:43 PM Today



-People with education 16 are more tend to buy products whereas higher education people dont buy KP281 and least education level dont buy KP781.



Palak Vyas

7:41 PM Today



Partnered tend to buy more KP281 whereas KP781 is least bought by singles.

```
d=pd.crosstab(data['Product'],data['Gender'])
print(d)
```

Gender	Female	Male
Product		
KP281	40	40
KP481	29	31
KP781	7	33



Palak Vyas

7:46 PM Today



-People with high fitness level opt for KP781.
-People with mid fitness level prefer KP281 and KP481.

```
d=pd.crosstab(data['Product'],data['Education'])
print(d)
```

Education	12	13	14	15	16	18	20	21
Product								
KP281	2	3	30	4	39	2	0	0
KP481	1	2	23	1	31	2	0	0
KP781	0	0	2	0	15	19	1	3



Palak Vyas

7:49 PM Today



-Data shows people with high usage dont opt for KP281 and KP481.

```
d=pd.crosstab(data['Product'],data['MaritalStatus'])
print(d)
```

MaritalStatus	Partnered	Single
Product		
KP281	48	32
KP481	36	24
KP781	23	17


```
d=pd.crosstab(data['Product'],data['Fitness'])  
print(d)
```

Fitness	1	2	3	4	5
Product					
KP281	1	14	54	9	2
KP481	1	12	39	8	0
KP781	0	0	4	7	29

```
d=pd.crosstab(data['Product'],data['Usage'])  
print(d)
```

Usage	2	3	4	5	6	7
Product						
KP281	19	37	22	2	0	0
KP481	14	31	12	3	0	0
KP781	0	1	18	12	7	2

Double-click (or enter) to edit

✓ 0s completed at 7:53 PM

