SCALER

Aerofit - Business Case Descriptive Statistics and Probability

Submitted by :

Harsha Srinivas, Tanna
harshasrinivas.tanna@gmail.com
Scaler DSML - Morning TTS Feb 2023
Submitted on August 22, 2023

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm

df = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?163999
```

```
# the dataset contains 180 rows and 9 columns
df
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Product        180 non-null    object
 1   Age            180 non-null    int64 
 2   Gender         180 non-null    object
 3   Education      180 non-null    int64 
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    int64 
 6   Fitness        180 non-null    int64 
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

```
df.columns
```

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

```
df.shape
```

```
(180, 9)
```

```
df.dtypes
```

```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

```
# No duplicate rows found in the dataset
df[df.duplicated()]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles

```
# No NaN values are present in the dataset
df.isnull().sum()
```

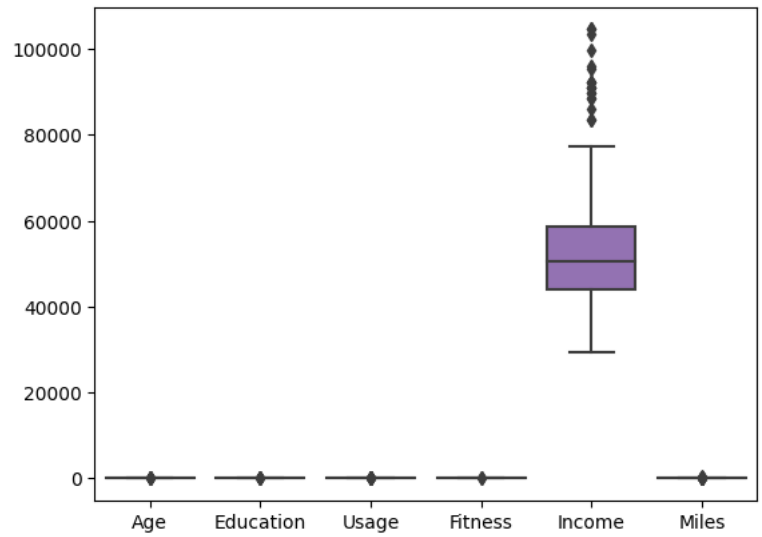
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

```
# statistical summary
df.describe()
```

	Age	Education	Usage	Fitness	Income	Miles
count	180.000000	180.000000	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111	53719.577778	103.194444
std	6.943498	1.617055	1.084797	0.958869	16506.684226	51.863605
min	18.000000	12.000000	2.000000	1.000000	29562.000000	21.000000
25%	24.000000	14.000000	3.000000	3.000000	44058.750000	66.000000
50%	26.000000	16.000000	3.000000	3.000000	50596.500000	94.000000
75%	33.000000	16.000000	4.000000	4.000000	58668.000000	114.750000
max	50.000000	21.000000	7.000000	5.000000	104581.000000	360.000000

```
# Box Plots for various columns
sns.boxplot(data=df)
```

```
<Axes: >
```




```
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

```
df['Product'].value_counts()
```

```
KP281    80
KP481    60
```

```
KP781    40
Name: Product, dtype: int64
```

```
df['Age'].value_counts()
```

```
25    25
23    18
24    12
26    12
28     9
35     8
33     8
30     7
38     7
21     7
22     7
27     7
31     6
34     6
29     6
20     5
40     5
32     4
19     4
48     2
37     2
45     2
47     2
46     1
50     1
18     1
44     1
43     1
41     1
39     1
36     1
42     1
Name: Age, dtype: int64
```

```
df['Education'].value_counts()
```

```
16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: Education, dtype: int64
```

```
df['Usage'].value_counts()
```

```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

```
df['Fitness'].value_counts()
```

```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

```
df['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
```

```
df['Miles'].value_counts()
```

```
85     27
95     12
66     10
75     10
47      9
106     9
94      8
113     8
53      7
100     7
180     6
200     6
56      6
64      6
127     5
160     5
42      4
150     4
38      3
74      3
170     3
120     3
103     3
132     2
141     2
280     1
260     1
300     1
240     1
112     1
212     1
80      1
140     1
21      1
169     1
188     1
360     1
Name: Miles, dtype: int64
```

```
df['Gender'].value_counts()
```

```
Male      104
Female     76
Name: Gender, dtype: int64
```

```
df['MaritalStatus'].value_counts()
```

```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

```
type_1_count = df[df['Product'] == 'KP281'].count()
type_2_count = df[df['Product'] == 'KP481'].count()
type_3_count = df[df['Product'] == 'KP781'].count()
print("count of treadmill of type 1:",type_1_count.values[0],"\ncount of treadmill of type 2:",type_2_count.values[0],"\ncount
print("\n % of treadmill of type 1:",np.round(type_1_count.values[0]/1.80,2),"\n % of treadmill of type 2:",np.round(type_2_co
```

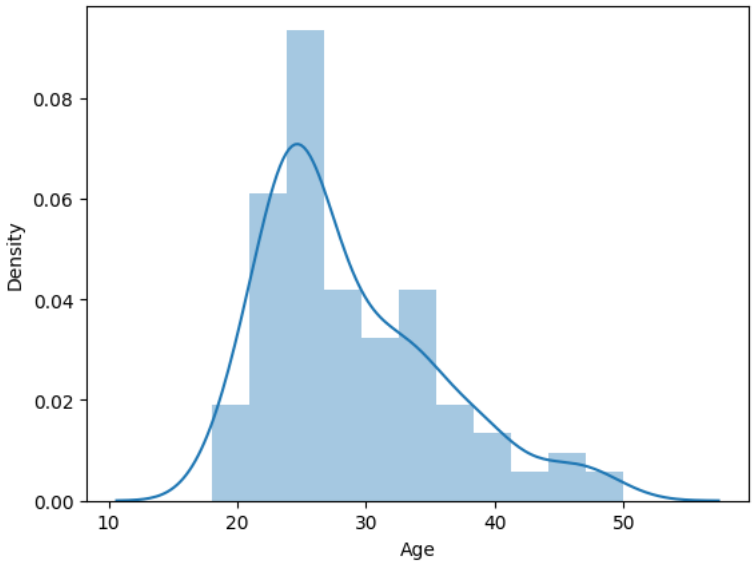
```
count of treadmill of type 1: 80
count of treadmill of type 2: 60
count of treadmill of type 3: 40

 % of treadmill of type 1: 44.44
 % of treadmill of type 2: 33.33
 % of treadmill of type 3: 22.22
```

```
df.info()
```

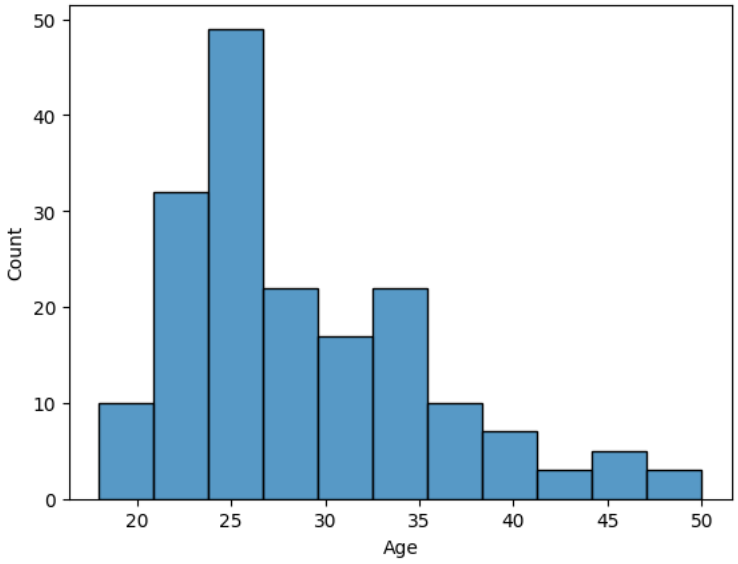
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Product        180 non-null    object
 1   Age            180 non-null    int64 
 2   Gender         180 non-null    object
 3   Education      180 non-null    int64 
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    int64 
 6   Fitness        180 non-null    int64 
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

```
sns.distplot(df.Age)
plt.show()
```

```
<ipython-input-47-0e50cf71bdb3>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Age)
```



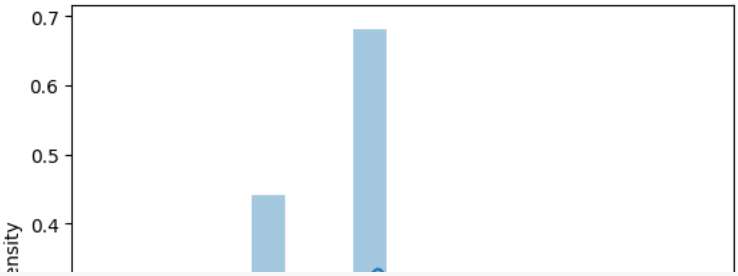

```
sns.histplot(df.Age)
plt.show()
# Most of the buyers have 25 years
# People with 40-50 age are the least buyers of the products
```



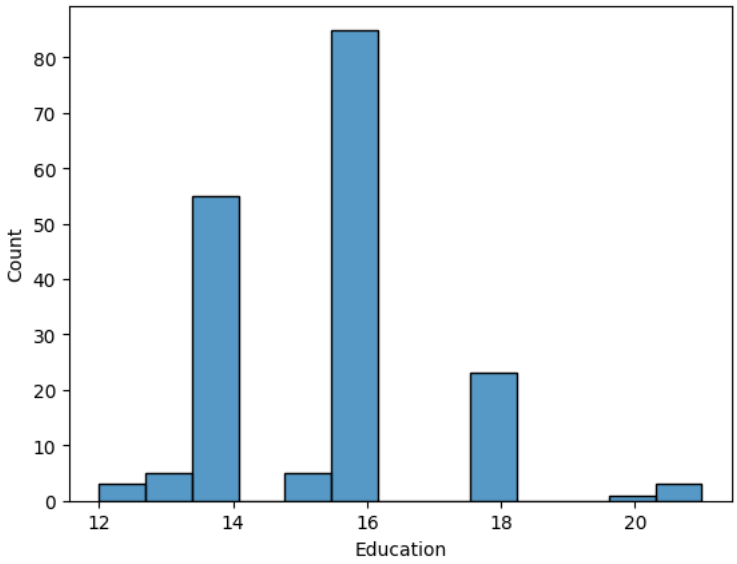

```
sns.distplot(df.Education)
plt.show()
```

```
<ipython-input-43-2456c85bb67c>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(df.Education)
```



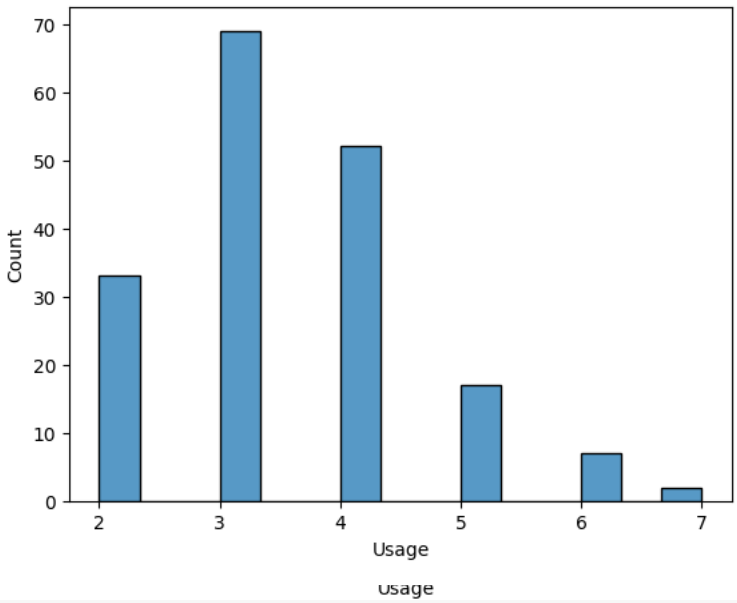

```
sns.histplot(df.Education)
plt.show()
# Most of the buyers have 16 years of education
```




```
sns.distplot(df.Usage)
plt.show()
```

```
<ipython-input-50-d51d2c5337c8>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
```

```
sns.histplot(df.Usage)
plt.show()
# most of the buyers tend to use the treadmill 3 times a week
```

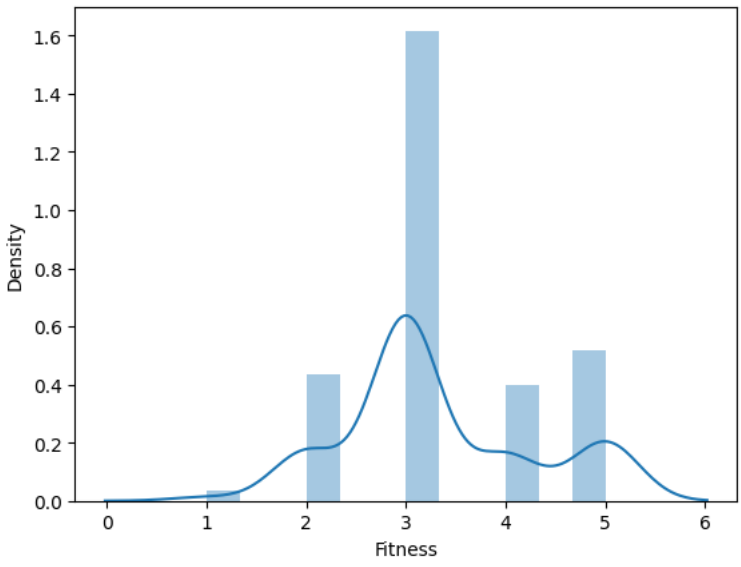


usage

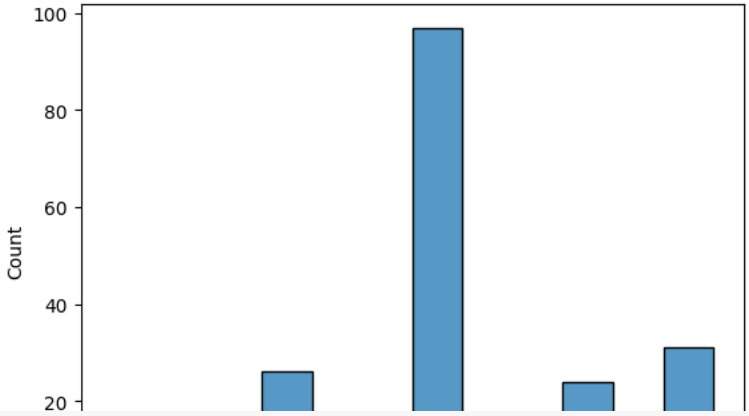
```
sns.distplot(df.Fitness)
plt.show()
```

```
<ipython-input-53-d101120d52c8>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(df.Fitness)
```

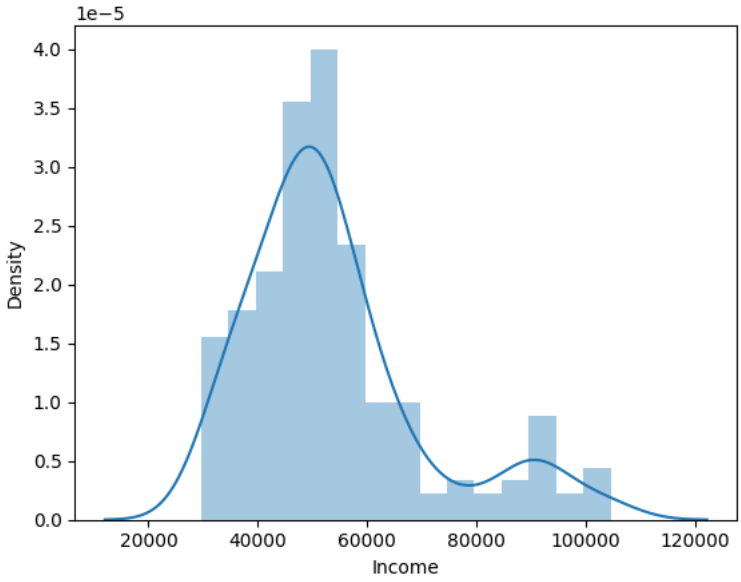



```
sns.histplot(df.Fitness)
plt.show()
# most of the buyers rated themselves '3' on a scale of 1 to 5 where 5 being in excellent shape and 1 being in poor shape
# from this we can say most of the buyers are being modest here
```




```
sns.distplot(df.Income)
plt.show()
```

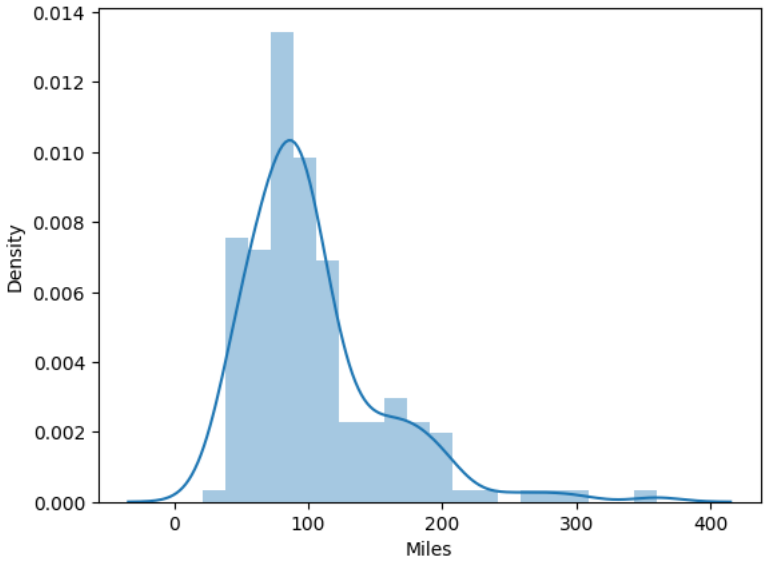
```
<ipython-input-55-52c9657eb147>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Income)
```



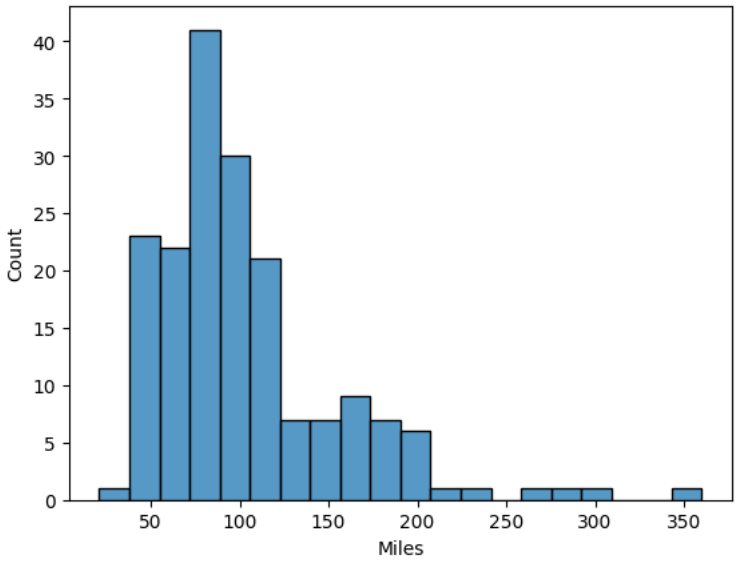

```
sns.histplot(df.Income)
plt.show()
# Most of the buyers have an income of around $ 50k
```

```
sns.distplot(df.Miles)
plt.show()
```

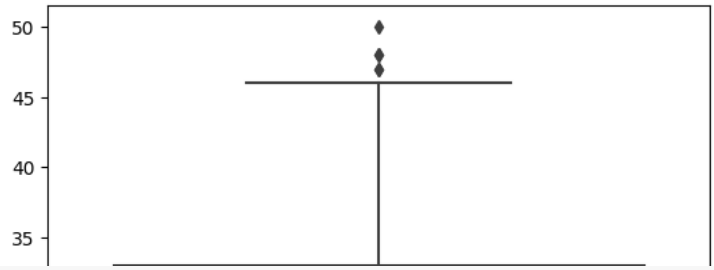
```
<ipython-input-57-3138d4372ef0>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function wit
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Miles)
```



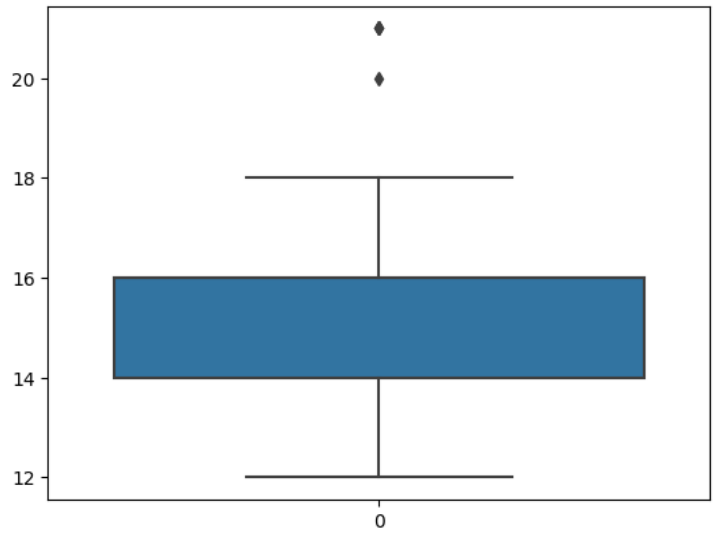

```
sns.histplot(df.Miles)
plt.show()
# most of the buyers expect to walk around 80 miles a week
```



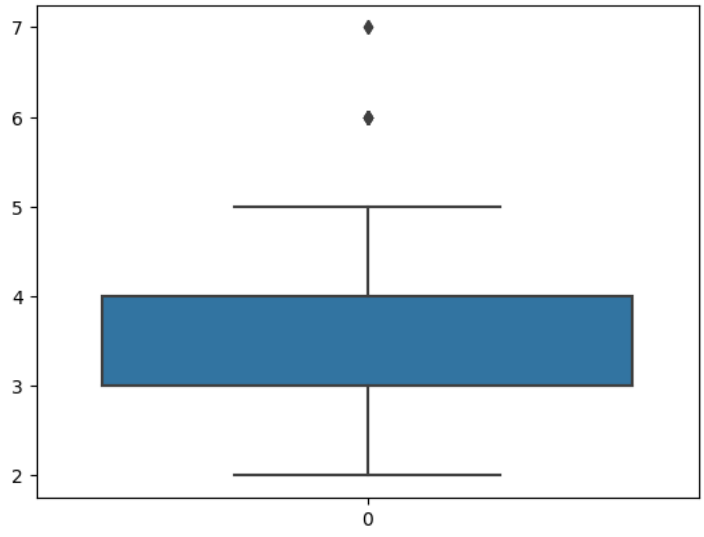

```
sns.boxplot(df.Age)
plt.show()
# The mean age of buyers is around 25 with buyers beyond 46 can be called outliers
```



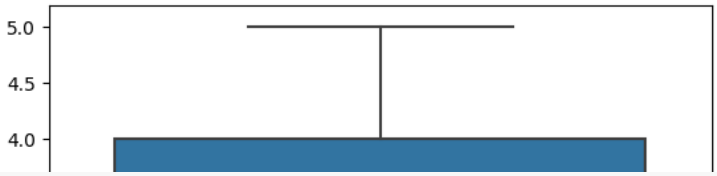

```
sns.boxplot(df.Education)
plt.show()
```



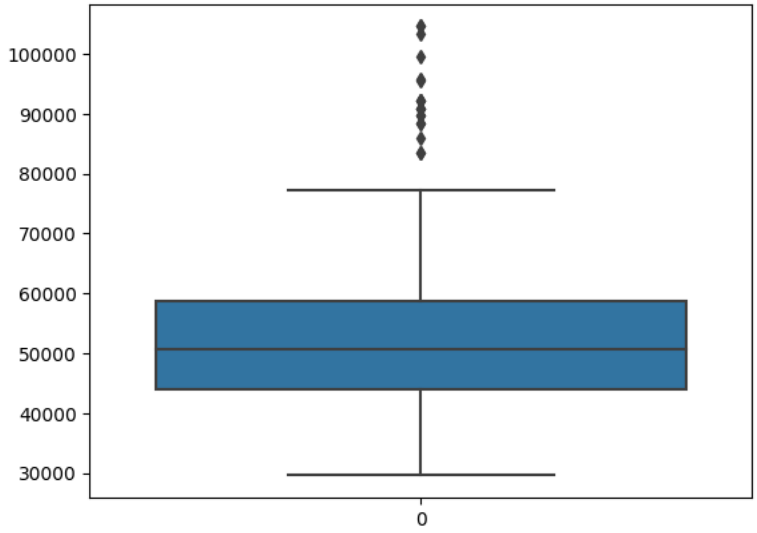

```
sns.boxplot(df.Usage)
plt.show()
```



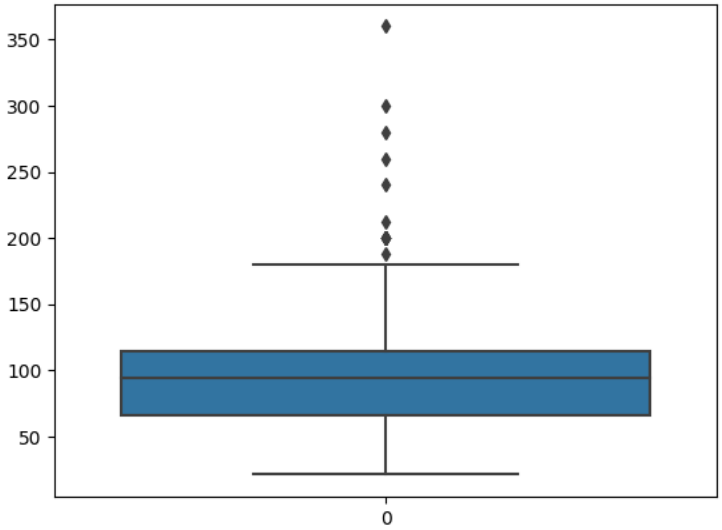

```
sns.boxplot(df.Fitness)
plt.show()
```




```
sns.boxplot(df.Income)
plt.show()
# mean income of the buyers is $ 50k
```



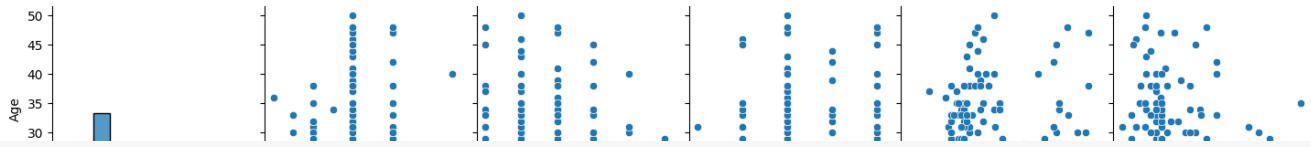

```
sns.boxplot(df.Miles)
plt.show()
# most buyers tend to walk around 100 miles every week
```



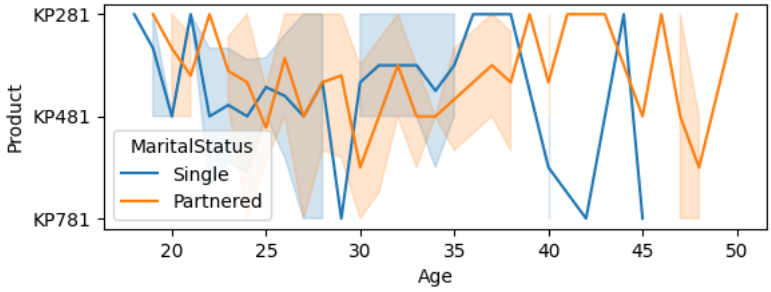

```
sns.heatmap(df.corr(),annot = True)
plt.show()
# we can see from the heatmap
# Age is more positively correlated to Income and least to Usage
# Usage is more positively correlated to Miles and least to Age
# Fitness is more positively correlated to Miles and least to Age
# MIles is least correlated to Age
```

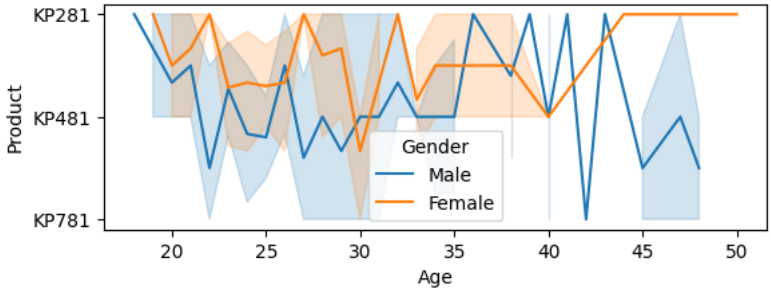
```
<ipython-input-71-7a3f2fbe6c0a>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a
  sns.heatmap(df.corr(),annot = True)
```


```
sns.pairplot(df)
plt.show()
# all possible pair plots for the Aerofit dataset
```

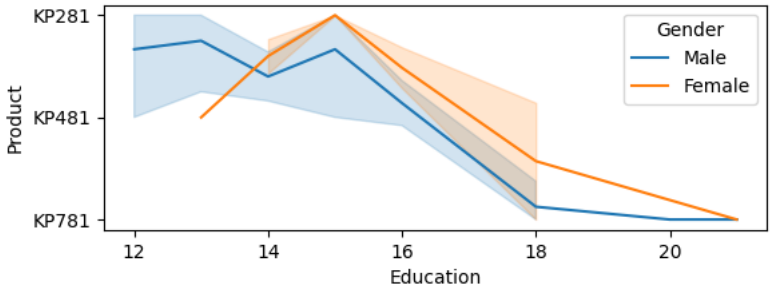


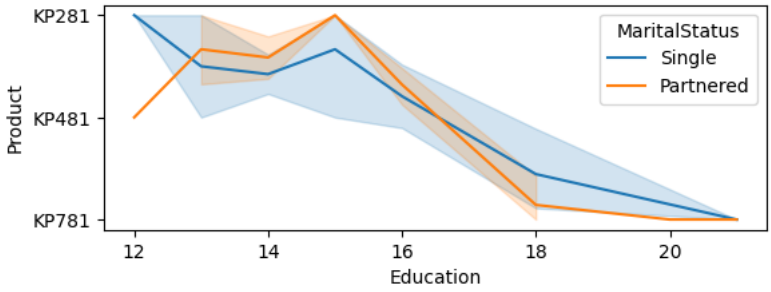

```
plt.subplot(2,1,1)
sns.lineplot(x = df.Age, y = df.Product, data = df, hue=df.MaritalStatus)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(x = df.Age, y = df.Product, data = df, hue=df.Gender)
plt.show()
# For the age group 45+ men prefer KP 481 while women prefer KP 281
```



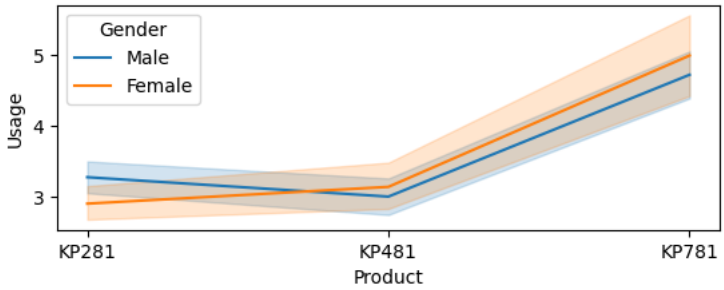


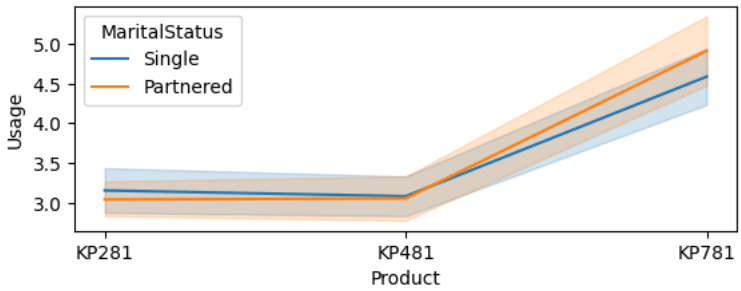

```
plt.subplot(2,1,1)
sns.lineplot(x = df.Education, y = df.Product, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(x = df.Education, y = df.Product, data = df, hue=df.MaritalStatus)
plt.show()
# more educated people prefer KP781 and relatively less educated people tend to buy KP281
```



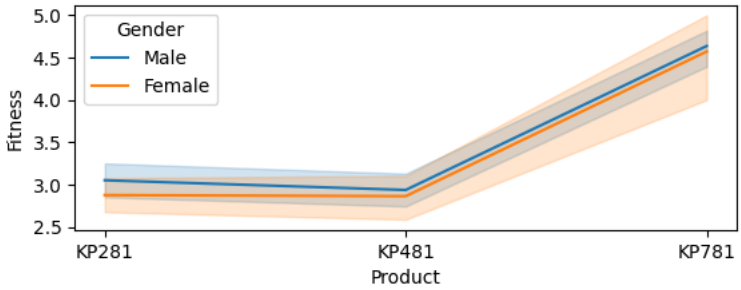


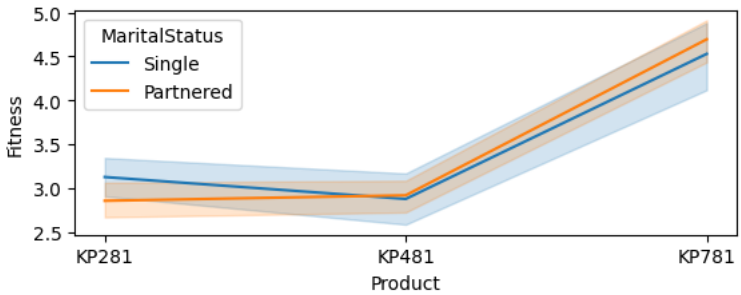

```
plt.subplot(2,1,1)
sns.lineplot(y = df.Usage, x = df.Product, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Usage, x = df.Product, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers who have most usage tend to buy KP 781
```



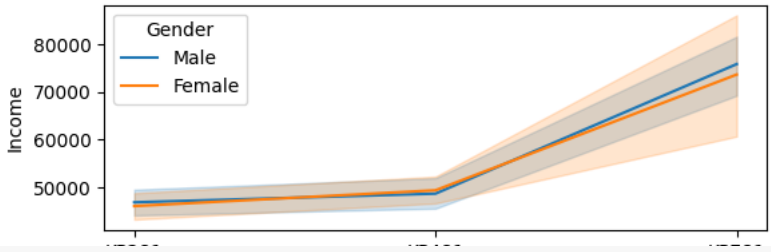



```
plt.subplot(2,1,1)
sns.lineplot(y = df.Fitness, x = df.Product, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Fitness, x = df.Product, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers who are more in good shape tend to but KP 781
```

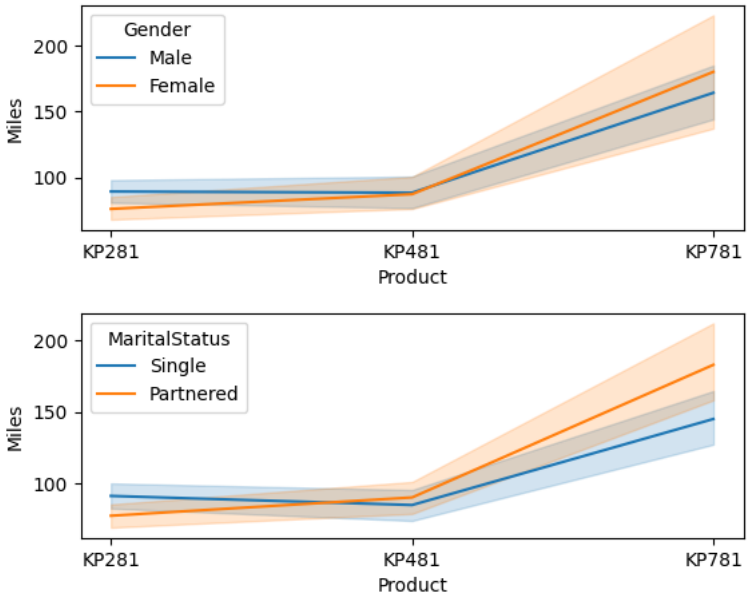




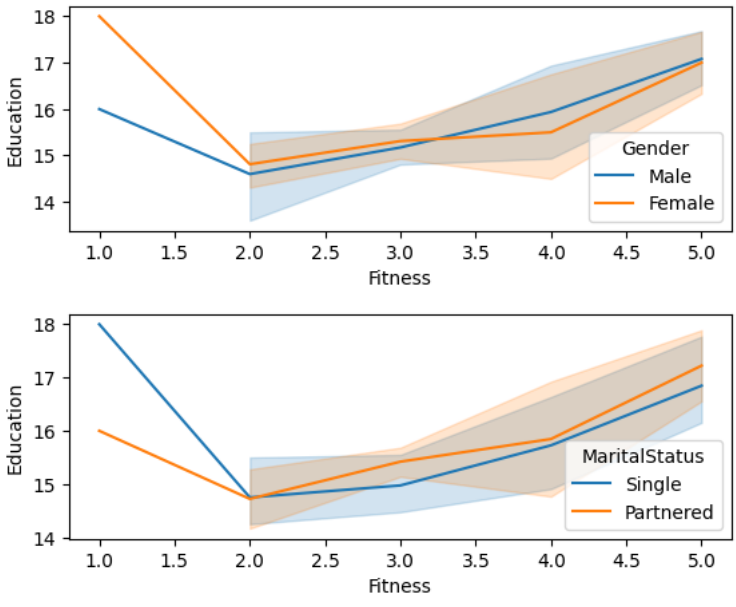

```
plt.subplot(2,1,1)
sns.lineplot(y = df.Income, x = df.Product, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Income, x = df.Product, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers with more Income tend to buy KP 781
```




```
plt.subplot(2,1,1)
sns.lineplot(y = df.Miles, x = df.Product, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Miles, x = df.Product, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers who tend to walk more tend to buy KP 781
```

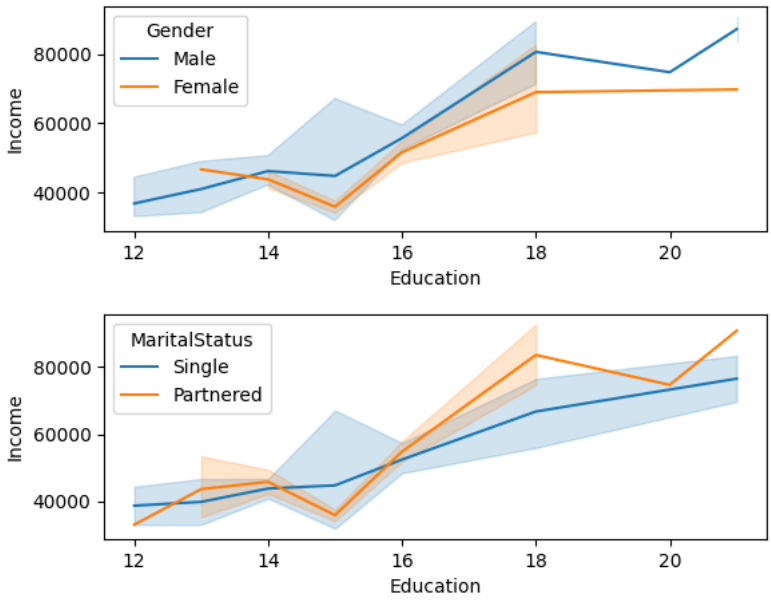



```
plt.subplot(2,1,1)
sns.lineplot(x = df.Fitness, y = df.Education, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(x = df.Fitness, y = df.Education, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers with more education are either less fit or more fit
# Buyers with around 15-16 years of education are moderately fit
```

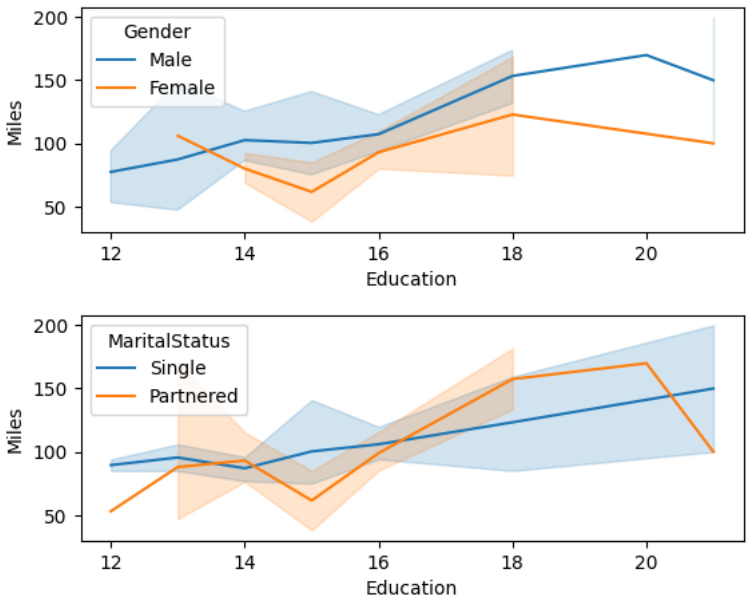



```
plt.subplot(2,1,1)
sns.lineplot(y = df.Income, x = df.Education, data = df, hue=df.Gender)
plt.show()
```

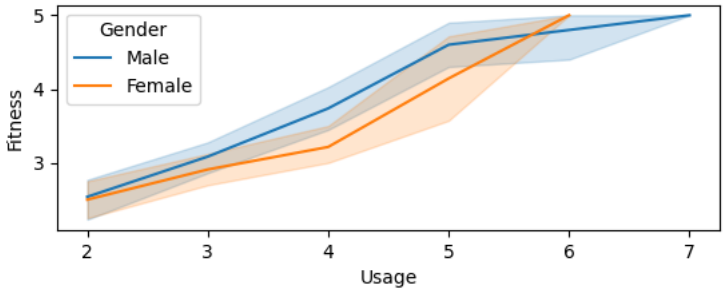
```
plt.subplot(2,1,2)
sns.lineplot(y = df.Income, x = df.Education, data = df, hue=df.MaritalStatus)
plt.show()
```



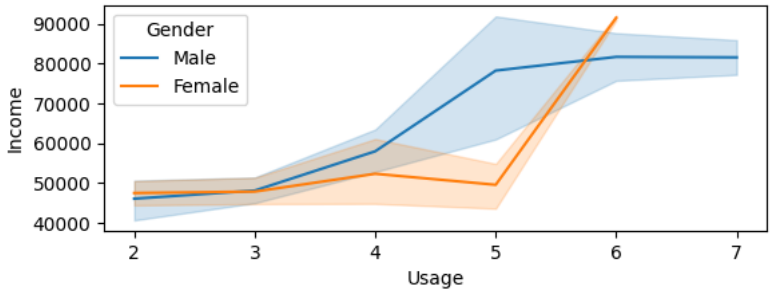

```
plt.subplot(2,1,1)
sns.lineplot(y = df.Miles, x = df.Education, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Miles, x = df.Education, data = df, hue=df.MaritalStatus)
plt.show()
# Single buyers tend to walk for 100-150 miles each week
# Couple buyers with 15+ years education background tend to walk for 75-150 miles...This is a huge spread and the trend is not
# Female buyers with less than 13 years of education tend to walk more than male buyers but the trend is reverse for buyers wi
```

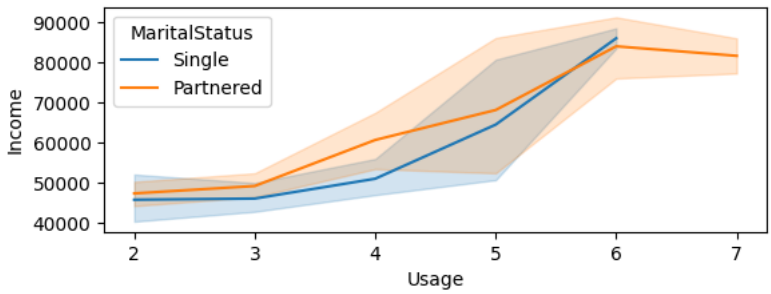



```
plt.subplot(2,1,1)
sns.lineplot(y = df.Fitness, x = df.Usage, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Fitness, x = df.Usage, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers who rated themselves more fit intend to use the treadmill more
```



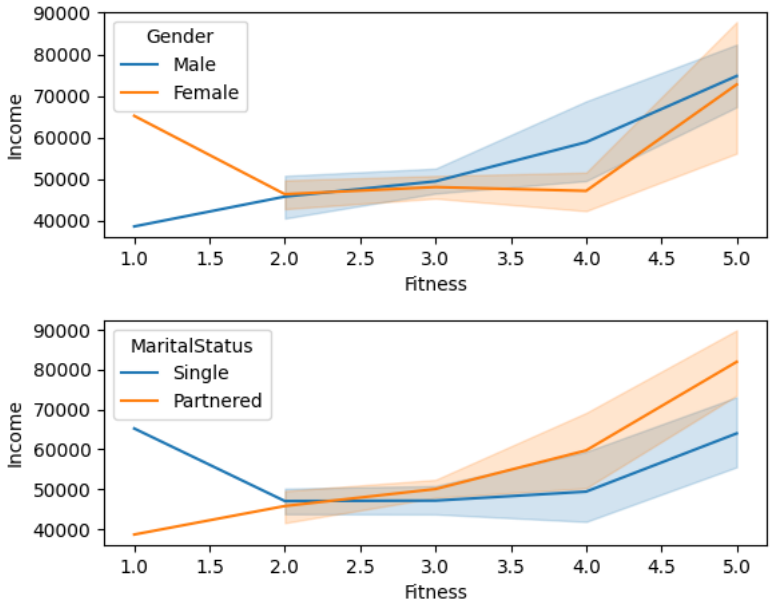

```
plt.subplot(2,1,1)
sns.lineplot(y = df.Income, x = df.Usage, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Income, x = df.Usage, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers with more income tend have more usage
```



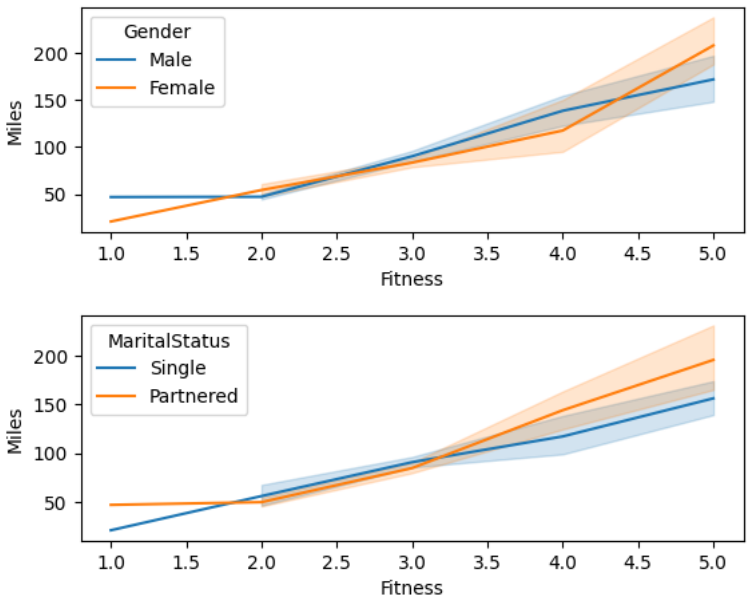



```
plt.subplot(2,1,1)
sns.lineplot(y = df.Miles, x = df.Usage, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Miles, x = df.Usage, data = df, hue=df.MaritalStatus)
plt.show()
# Clearly buyers who tend to walk for more miles tend to use the treadimll more
```

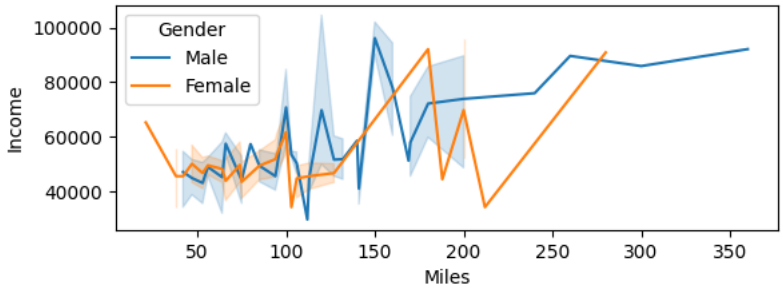
```
plt.subplot(2,1,1)
sns.lineplot(y = df.Income, x = df.Fitness, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Income, x = df.Fitness, data = df, hue=df.MaritalStatus)
plt.show()
# Female buyers with more income are either in good shape or bad shape
# Female buyers with relatively lesser income rated themselves as moderately fit
# Male buyers rated themselves as fit when they have more income and this is a clear trend
# Single Buyers with relatively more income are either less fit or more fit
# Single buyers with relatively less income rated themselves as moderately fit
# Non Single Bueyrs with more income rated themselves as more fit
```

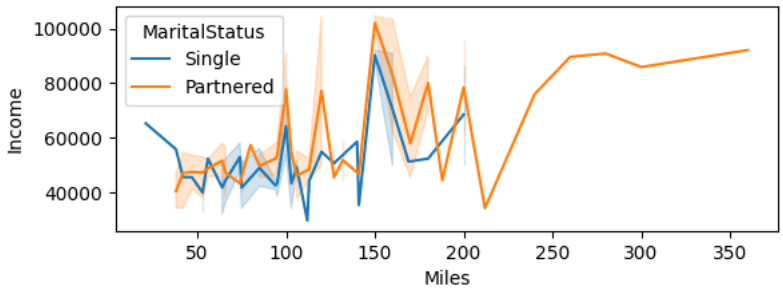



```
plt.subplot(2,1,1)
sns.lineplot(y = df.Miles, x = df.Fitness, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(y = df.Miles, x = df.Fitness, data = df, hue=df.MaritalStatus)
plt.show()
# Buyers who tend to walk more on treadmill rated themselves as more fit
```




```
plt.subplot(2,1,1)
sns.lineplot(x = df.Miles, y = df.Income, data = df, hue=df.Gender)
plt.show()
plt.subplot(2,1,2)
sns.lineplot(x = df.Miles, y = df.Income, data = df, hue=df.MaritalStatus)
plt.show()
```






```
# Business Insights:
1.Buyers age group is 18-50 with mean age being 25
2.More people bought KP281 followed by KP 481 and KP781
3.Most of the buyers have relatively less education and the mean education (in number of years) is 16
4.More people tend to use the treadmills less i.e.,most of the buyers tend to use the treadmill 3 times a week
5.Most people rated themselves as moderately fit which is '3' rating out of '5' from this we can say most of the buyers are be
6.58.8% of the buyers are male
7.59.4 % of the buyers are partenered
8. % of treadmill of type 1: 44.44
   % of treadmill of type 2: 33.33
   % of treadmill of type 3: 22.22
9.People with 40-50 age are the least buyers of the products
10.Most of the buyers have an income of around $ 50k
11.most of the buyers expect to walk around 80-100 miles a week
12.Age is more positively correlated to Income and least to Usage
13.Usage is more positively correlated to Miles and least to Age
14.Fitness is more positively correlated to Miles and least to Age
15.MIles is least correlated to Age
16.For the age group 45+ men prefer KP 481 while women prefer KP 281
17.More educated people prefer KP781 and relatively less educated people tend to buy KP281
18.Buyers who have most usage tend to buy KP 781
19.Buyers who are more in good shape tend to but KP 781
20.Buyers with more Income tend to buy KP 781
21.Buyers who tend to walk more tend to buy KP 781
22.Buyers with more education are either less fit or more fit
23.Buyers with around 15-16 years of education are moderately fit
24.Single buyers tend to walk for 100-150 miles each week
25.Couple buyers with 15+ years education background tend to walk for 75-150 miles...This is a huge spread and the trend is no
26.Female buyers with less than 13 years of education tend to walk more than male buyers but the trend is reverse for buyers w
27.Buyers who rated themselves more fit intend to use the treadmill more
28.Buyers with more income tend have more usage
29.Clearly buyers who tend to walk for more miles tend to use the treadimll more
30.Female buyers with more income are either in good shape or bad shape
31.Female buyers with relatively lesser income rated themselves as moderately fit
32.Male buyers rated themselves as fit when they have more income and this is a clear trend
33.Single Buyers with relatively more income are either less fit or more fit
34.Single buyers with relatively less income rated themselves as moderately fit
35.Non Single Buyers with more income rated themselves as more fit
```

```
# Actionable Insights:
1.Since there are very few buyers with age greater than 40 we
  can offer them discounts to increase the sale of the Aerofit treadmills.
2.People with 12,20 years of education don't buy Aerofit treadmills.
  We can ask our sales people to explainto them more about the
 benefits of being fit and we can also increase the marketing spend to target this age group
3.Since most people rated themselves as moderately fit by choosing a rating of 3 out of 5
  if we can convince this group of people to more premium treadmill
  KP 781 we can maximize our sale
4.For buyers with income less than $50k income we can recommend KP 281 which is
  not a premium treadmill and for buyers with more income we can recommend
  the premium treadmill KP 781
5.As Age is very poorly correlated to Usage and also Age is poorly correlated to Fitness
  we can run marketing campaigns/advertisements/endorsing by celebrities
  that targets young people and by explaining to them the benfits of being fit we can get them to buy Aerofit products more.
```

✓ 0s completed at 00:57