BUSINESS CASE - AEROFIT TREADMILL

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

from scipy.stats import norm, binom, geom
```

In [4]:
```
df = pd.read_csv("BC_1-Aerofit_Treadmill.csv")
df
```

Out[4]:

	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

In [123…
```
df.shape
```

Out[123]:
```
(180, 9)
```

In [125…
```
df.describe()
```

Out[125]:

	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

In [127…
```
df.describe(include = 'object')
```

Out[127]:

	Product	Gender	MaritalStatus
count	180	180	180
unique	3	2	2
top	KP281	Male	Partnered
freq	80	104	107

In [117…
```
df.nunique()
```

Out[117]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

In [118…
```
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
```

In [119…
```
df.groupby("Product").ngroups
```

Out[119]:
```
3
```

#1. Initial Observation from Data Basic Metrics, OBS-1: 1. There are 180 entires(180 rows, 9 colmns) in the data having people buying 3 groups of treadmill Products(Kp281, KP481, KP781). 2. Its a complete data set with no nulls. 3. 3 products bought by Male/Female with Marital status of single/partnered. 4. Age of min 18 to max of 50 has bought th treadmills with mean = 28 and 75th percentile==33(indicating 75 percent of people who bought treadmill are 33 and below) 5. Education (min 15 -21 years of education) with more number of users having 15 years of education with 1 std showing the spread of data to be less that it forms steeper graph compares to others 6. DTYPE : Prodict, Marital status and Gender are object data types which forms categorical data while others are numerical(integer) data

In [130…
```
df["Product"].value_counts()
```

Out[130]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

In [150…
```
df["Product"].unique()
```

Out[150]:
```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

In [151…
```
#Below shows head counts on each AGE
df["Age"].value_counts()
```

Out[151]:
```
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
```

In [154…
```
df["Gender"].value_counts()
```

Out[154]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

In [156…
```
df["MaritalStatus"].value_counts().
```

Out[156]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

#2. Observations: 1. There are 3 unique products ['KP281', 'KP481', 'KP781'] with more number of people (80) using KP281, 60 using KP481 and 40 using KP781. 2. Sales of KP281 is higher since the cost of the product is lower than the others. 3. Also On an average is 3 times a week. 4. We can observe that Paqtnered people has bought more and also when comparing the gender, men has bought the treadmills in a higher ratio than women

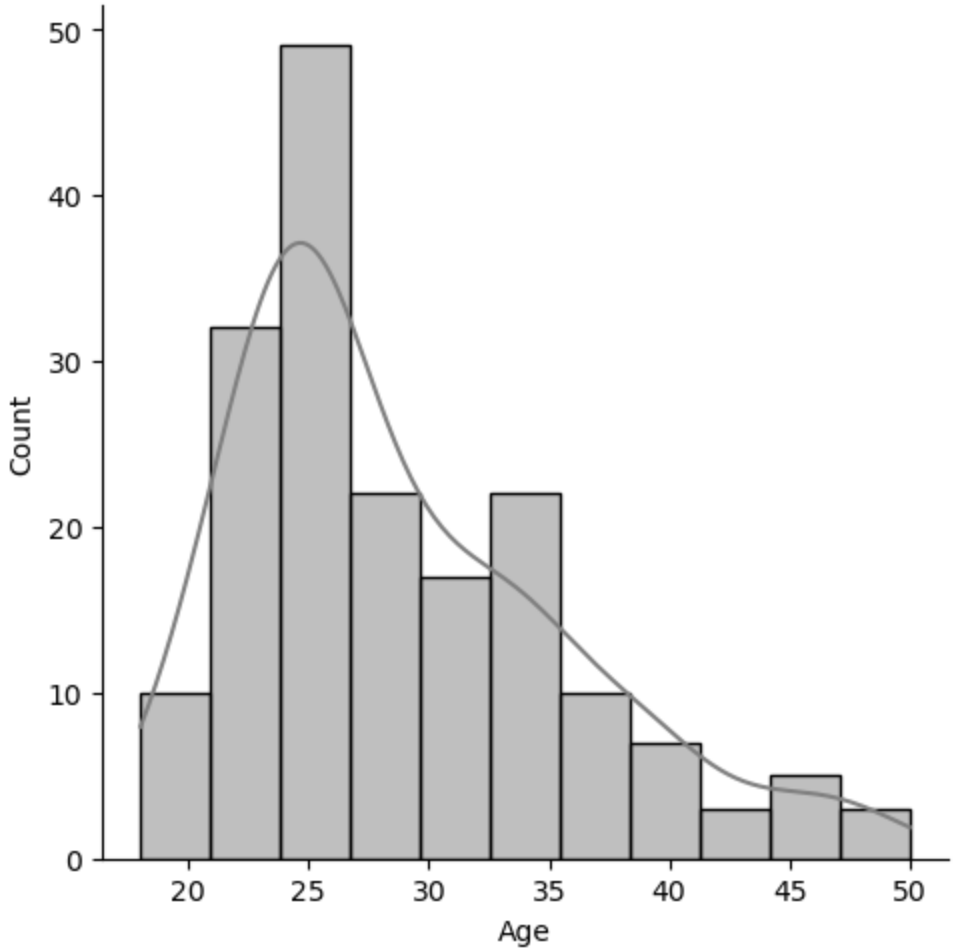
In [166…
```
#3. Data visualisation for more understanding
##UNIVARIATE ANALYSIS -displot (N)

sns.distplot
```

Out[166]:
```
<function seaborn.distributions.distplot(a=None, bins=None, hist=True, kde=True, rug=False, fit=None, hist_kws=None, kde_kws=None, rug_kws=None, fit_kws=None, color=None, vertical=False, norm_hist=False, axlabel=None, label=None, ax=None, x=None)>
```

In [179…
```
sns.displot(df["Age"], kde = True, color = "Grey", )
```

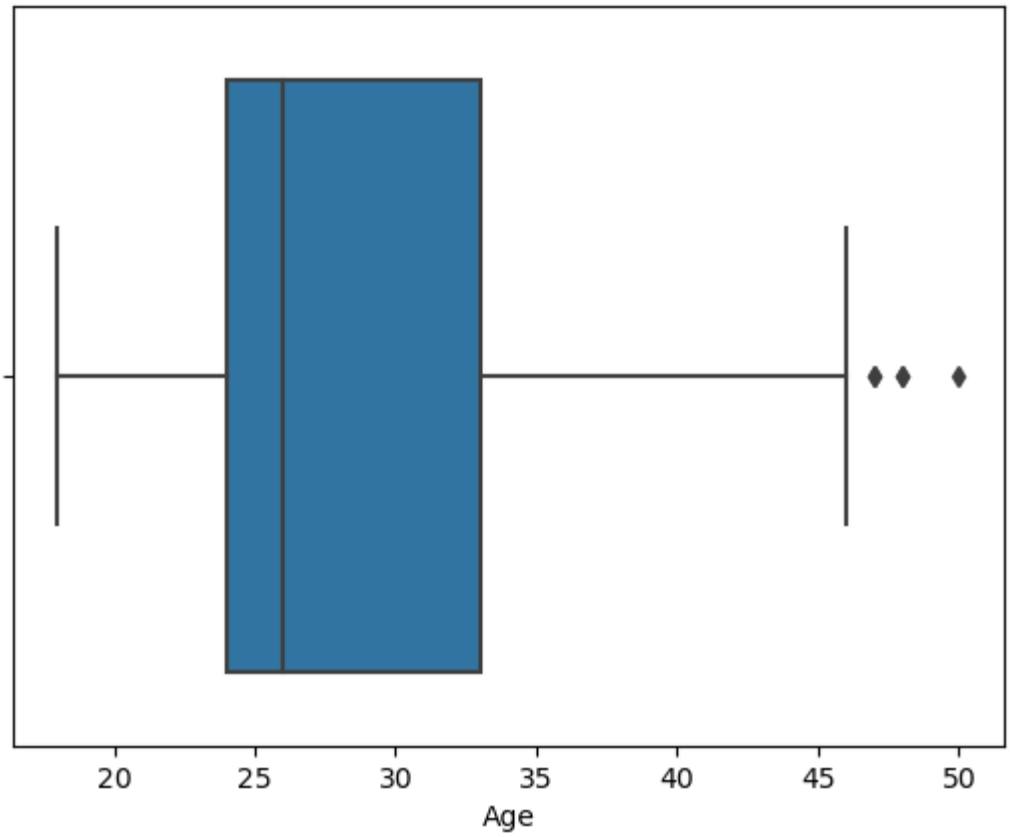
Out[179]:
```
<seaborn.axisgrid.FacetGrid at 0x2c13e9352e0>
```



In [61]:
```
##UNIVARIATE ANALYSIS -displot (Age - Numerical)
sns.boxplot(x = df["Age"])
plt.title("Fig-1:Age")
```

Out[61]:
```
Text(0.5, 1.0, 'Fig-1:Age')
```

Fig-1:Age

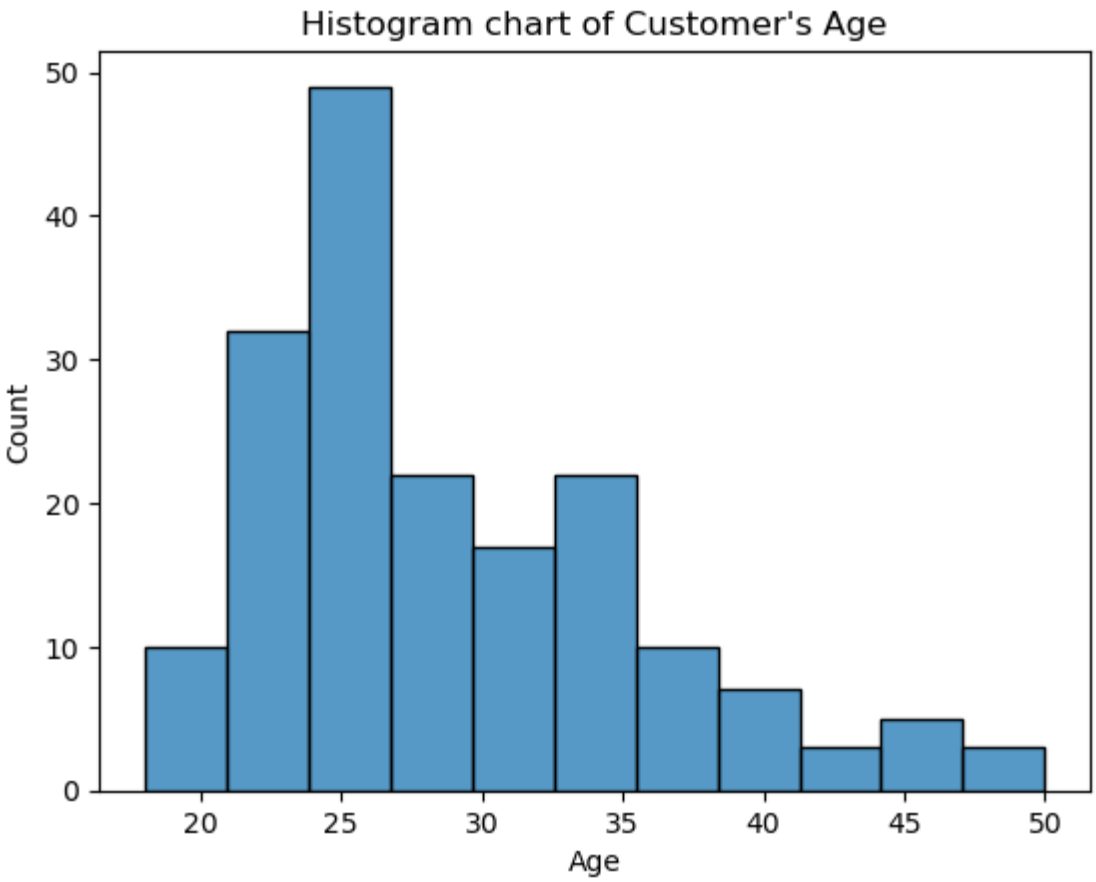


In [180…
```
sns.histplot
```

Out[180]:
```
<function seaborn.distributions.histplot(data=None, *, x=None, y=None, hue=None, weights=None, stat='count', bins='auto', binwidth=None, binrange=None, discrete=None, cumulative=False, common_bins=True, common_norm=True, multiple='layer', element='bars', fill=True, shrink=1, kde=False, kde_kws=None, line_kws=None, thresh=0, pthresh=None, pmax=None, cbar=False, cbar_ax=None, cbar_kws=None, palette=None, hue_order=None, hue_norm=None, color=None, log_scale=None, legend=True, ax=None, **kwargs)>
```

In [66]:
```
#1. ##UNIVARIATE ANALYSIS -histplot (Age - Numerical)
sns.histplot( x = df["Age"])
plt.title("Histogram chart of Customer's Age")
```

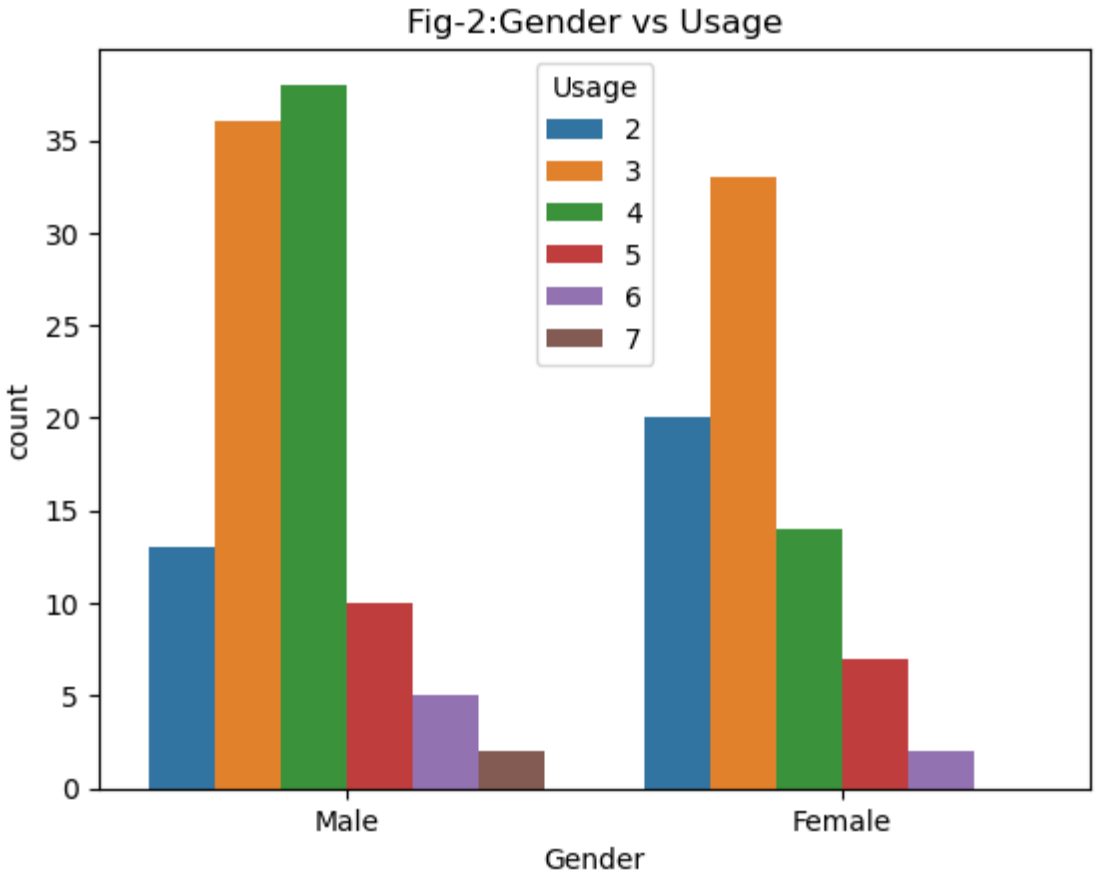
Out[66]:
```
Text(0.5, 1.0, "Histogram chart of Customer's Age")
```



In [62]:
```
##BIVARIATE ANALYSIS -countplot (Gender vs Age - (Categorical-Numerical)

countplot to visualise the men vs women usage of tradmills
sns.countplot( x = df["Gender"], hue = df["Usage"])
plt.title("Fig-2:Gender vs Usage")
```

Out[62]:
```
Text(0.5, 1.0, 'Fig-2:Gender vs Usage')
```



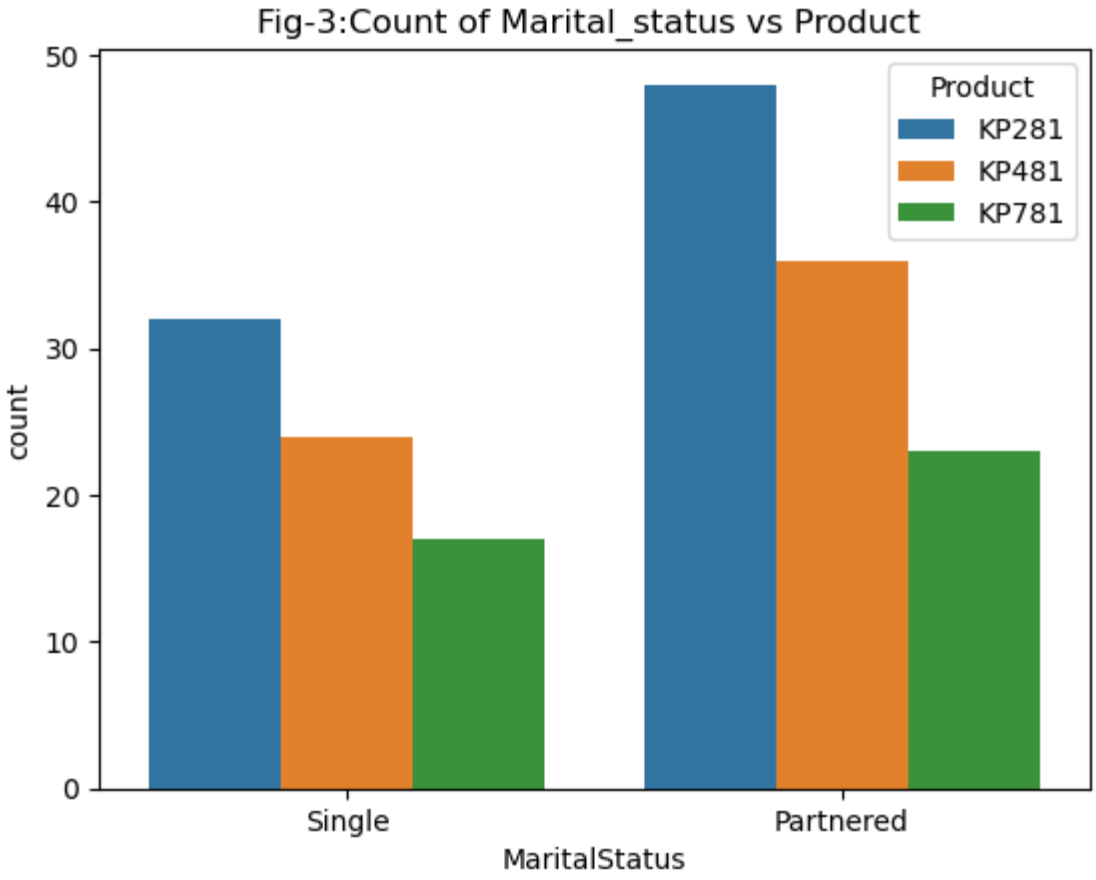
In [181…

```
##BIVARIATE ANALYSIS -countplot (Marital_status vs Product - (Categorical-Categoric

sns.countplot(x = df["MaritalStatus"], hue = df["Product"])
plt.title("Fig-3:Count of Marital_status vs Product ")
```

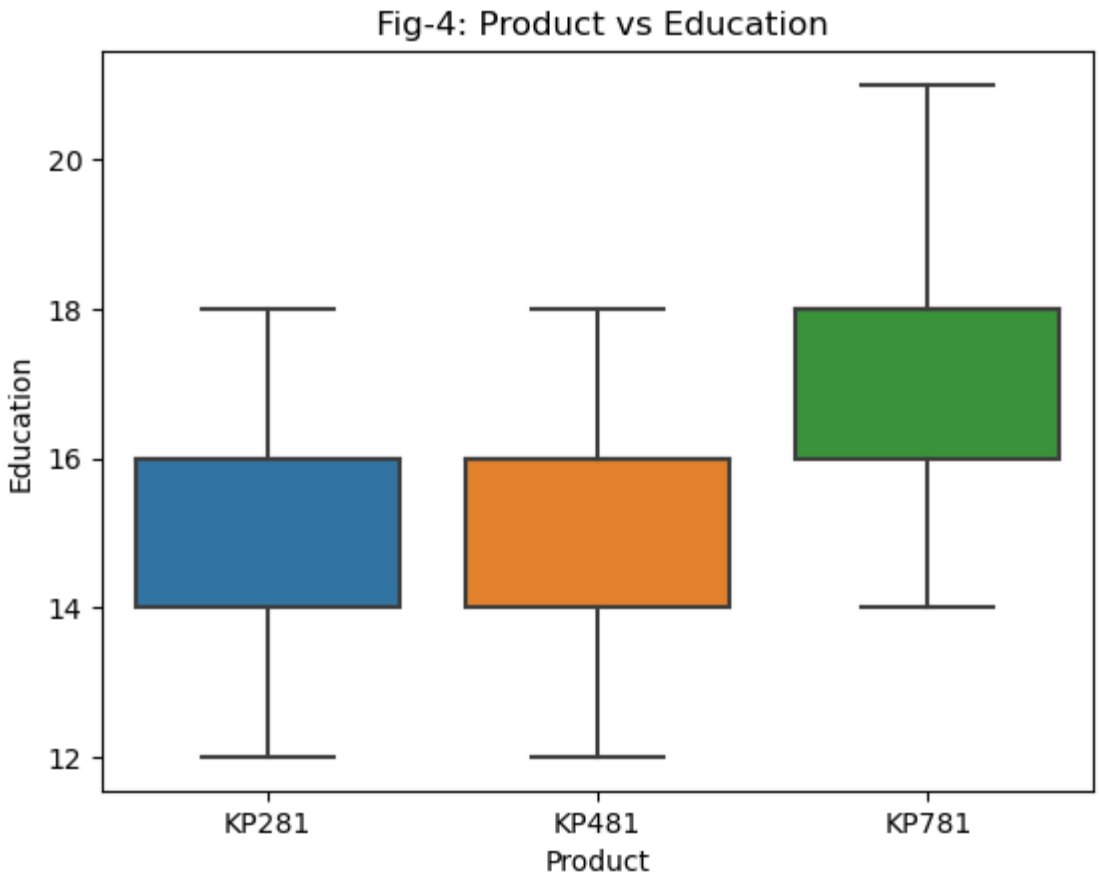
Out[181]:

```
Text(0.5, 1.0, 'Fig-3:Count of Marital_status vs Product ')
```

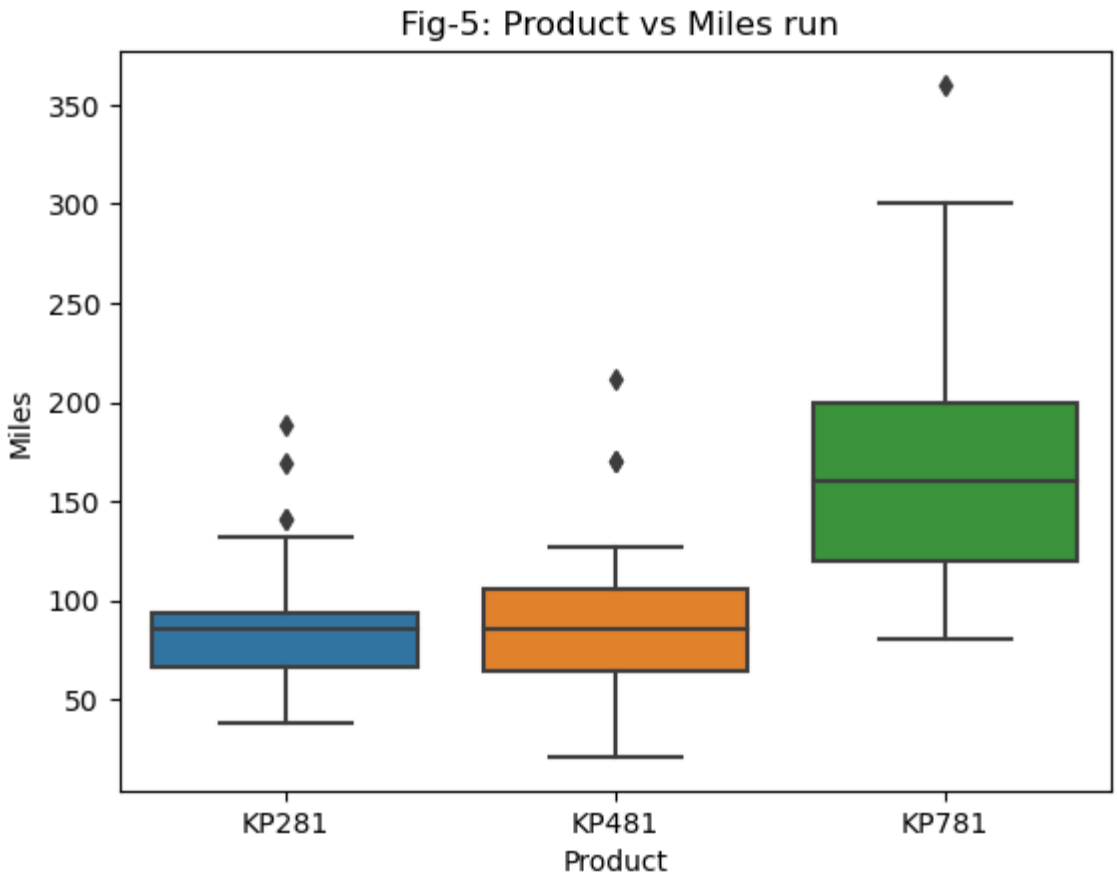


In [46]:
```
sns.boxplot(x = df["Product"], y = df["Education"])
plt.title(" Fig-4: Product vs Education")
plt.show()

sns.boxplot(x = df["Product"], y = df["Miles"])
plt.title(" Fig-5: Product vs Miles run")
```



Out[46]:
```
Text(0.5, 1.0, ' Fig-5: Product vs Miles run')
```



#3. Visual Analysis: Insights From Figure-1 and Figure-2 and figure-3: 1. From boxplot,people of age from 19 - 46 uses Aerofit trademill and >47 are outliers. But looks like most of the people are in range of 23 to 34. The histograms clearly indicates the fact that people 25 age people are in more number(arnd 50) and <10 people of age 50 uses treadmill. 2. The median lies around 26 which indicates people ofage arnd 26 buys the treadmill a lot than than others 3. Usage of treadmill by men is higher compared to women 4. From figure-3, it shows, that marital status has effect on tradmil product sales wherein Patnered people buys more than single which could be more due to health awareness and economical status. 5. From figure-4 and 5, it states that people with higher educartion and who runs a lot tends to buy KP781.

In [45]:
```
##pairplot
sns.pairplot(df)
```

Out[45]:
```
<seaborn.axisgrid.PairGrid at 0x267fe555940>
```

In [226...]
```
df
```

Out[226]:

	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

In [276…
```
#4. Outliers using IQR
data_mean, data_std = np.mean(df["Age"]), np.std(df["Age"])
print("data_mean:", data_mean, "data_std:", data_std)

cut_off = data_std * 1.3
lower, upper = data_mean - cut_off, data_mean + cut_off

print("lower_cutoff:", lower, "upper_cutoff:", upper)

outliers = [x for x in df["Age"] if x < lower or x > upper ]
print("The num of outliers are:",  len(outliers), "Outliers in age ar:", outliers)
```

```
data_mean: 28.788888888888888 data_std: 6.924183777720975
lower_cutoff: 19.787449977851622 upper_cutoff: 37.79032779992615
The num of outliers are: 30 Outliers in age ar: [18, 19, 19, 19, 38, 38, 38, 38, 3
9, 40, 41, 43, 44, 46, 47, 50, 19, 38, 38, 40, 40, 40, 45, 48, 38, 40, 42, 45, 47,
48]
```

In [48]:
```
#Missing values and
df.isnull()
```

Out[48]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
175	False	False	False	False	False	False	False	False	False
176	False	False	False	False	False	False	False	False	False
177	False	False	False	False	False	False	False	False	False
178	False	False	False	False	False	False	False	False	False
179	False	False	False	False	False	False	False	False	False

180 rows × 9 columns

There are 30 outliers and the Inter Quartile Range is between 19.8 to 37.8

In [208…
```
pd.crosstab
```

Out[208]:
```
<function pandas.core.reshape.pivot.crosstab(index, columns, values=None, rownames
=None, colnames=None, aggfunc=None, margins: 'bool' = False, margins_name: 'str' =
'All', dropna: 'bool' = True, normalize=False) -> 'DataFrame'>
```

In [249…
```
percentage_maritalstatus_product = pd.crosstab(df["Product"], df["MaritalStatus"],
percentage_maritalstatus_product
```

Out[249]:

MaritalStatus	Partnered	Single	All
Product			
KP281	26.666667	17.777778	44.444444
KP481	20.000000	13.333333	33.333333
KP781	12.777778	9.444444	22.222222
All	59.444444	40.555556	100.000000

1. 44% of people bought KP281, 33% people bought KP481, 22% bought KP781. Out of which partnered people has upper hand over unmarried persons.

In [233…
```
type(percentage_maritalstatus_product)
```

Out[233]:
```
pandas.core.frame.DataFrame
```

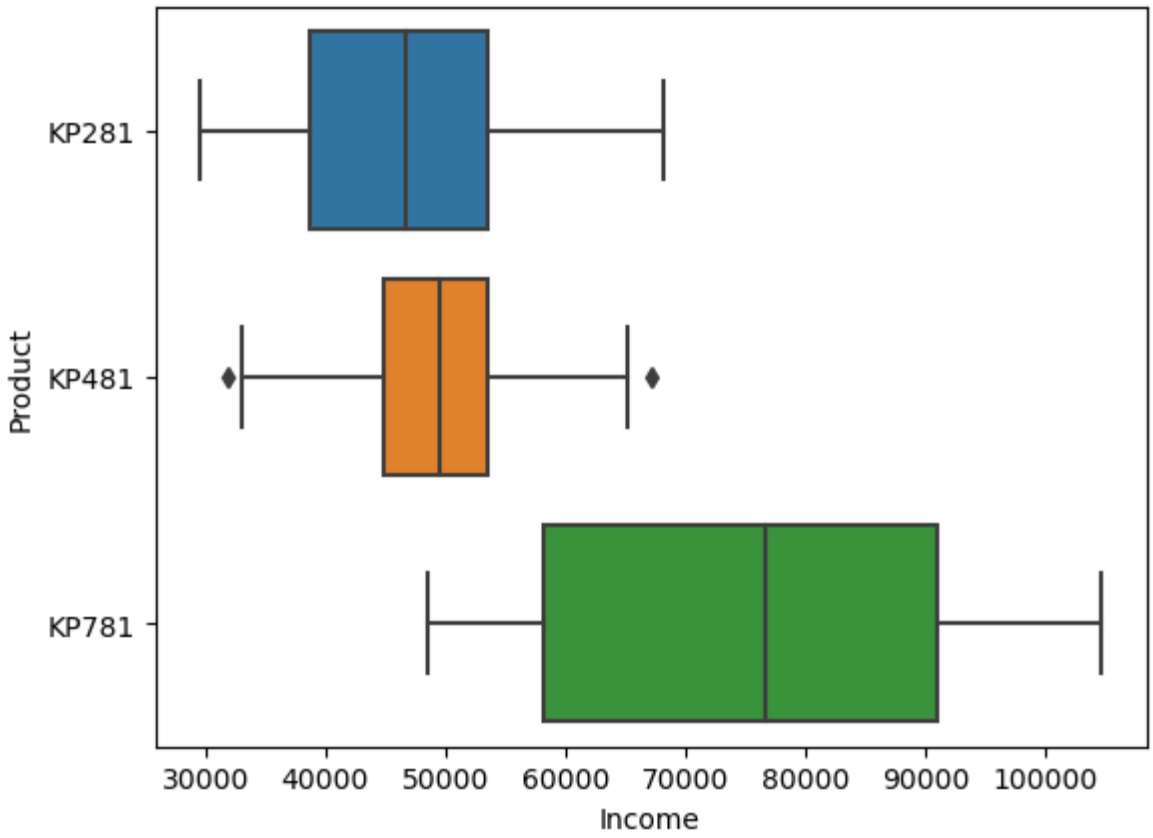
In [256…
```
#people who bought 70%lesser in kp281
df.describe()
```

Out[256]:

	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

In [283…
```
#Relatinship between income and product
sns.boxplot( x = df["Income"], y = df["Product"])
```

Out[283]:
```
<AxesSubplot:xlabel='Income', ylabel='Product'>
```



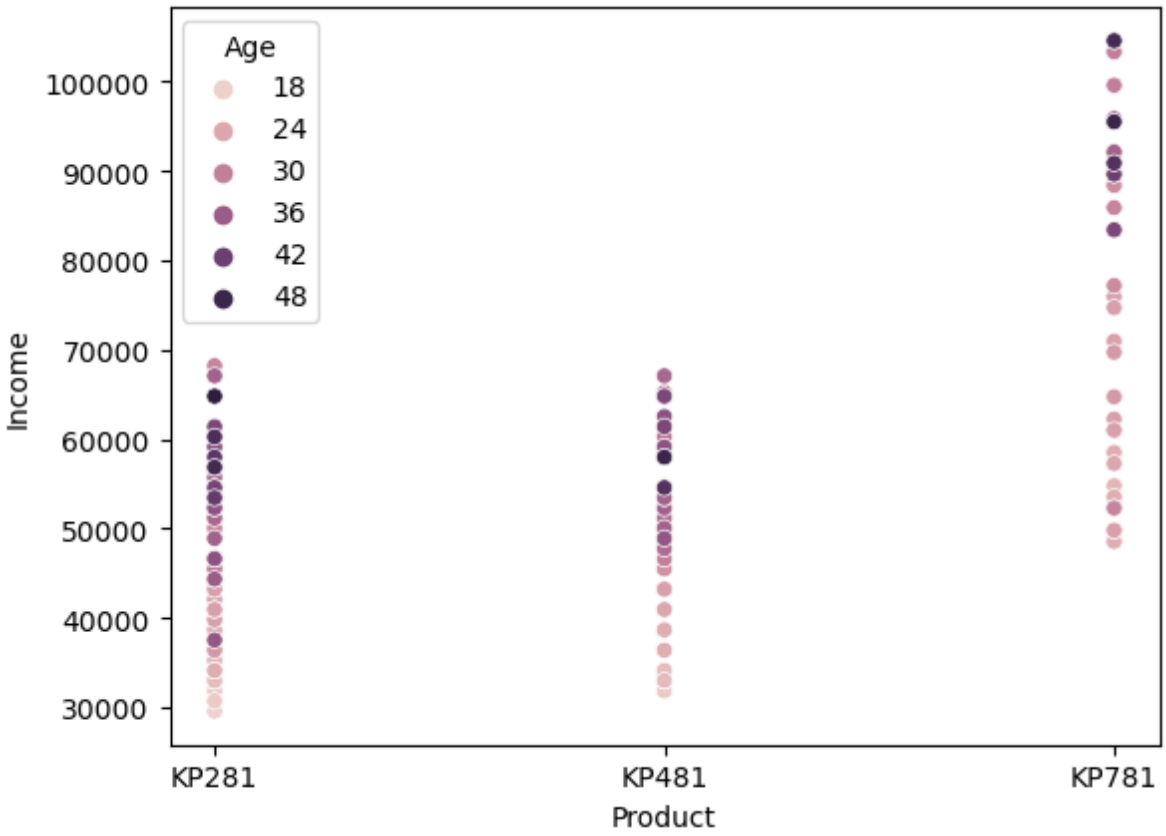
1. Both the variables are well related. since the cost of order of products are KP781 > KP481 >KP281. People having higher income prefers to buy the product KP781. 2. People with income 55K and lower prefer KP481 and KP281 but people with mid range of salaries buys 481. So it shows the income is directly related with product sales.

In [302…
```
sns.scatterplot
```

Out[302]:
```
<function seaborn.relational.scatterplot(*, x=None, y=None, hue=None, style=None,
size=None, data=None, palette=None, hue_order=None, hue_norm=None, sizes=None, siz
e_order=None, size_norm=None, markers=True, style_order=None, x_bins=None, y_bins=
None, units=None, estimator=None, ci=95, n_boot=1000, alpha=None, x_jitter=None, y
_jitter=None, legend='auto', ax=None, **kwargs)>
```

In [303…
```
sns.scatterplot(x = df["Product"], y = df["Income"], hue = df["Age"])
```

Out[303]:
```
<AxesSubplot:xlabel='Product', ylabel='Income'>
```

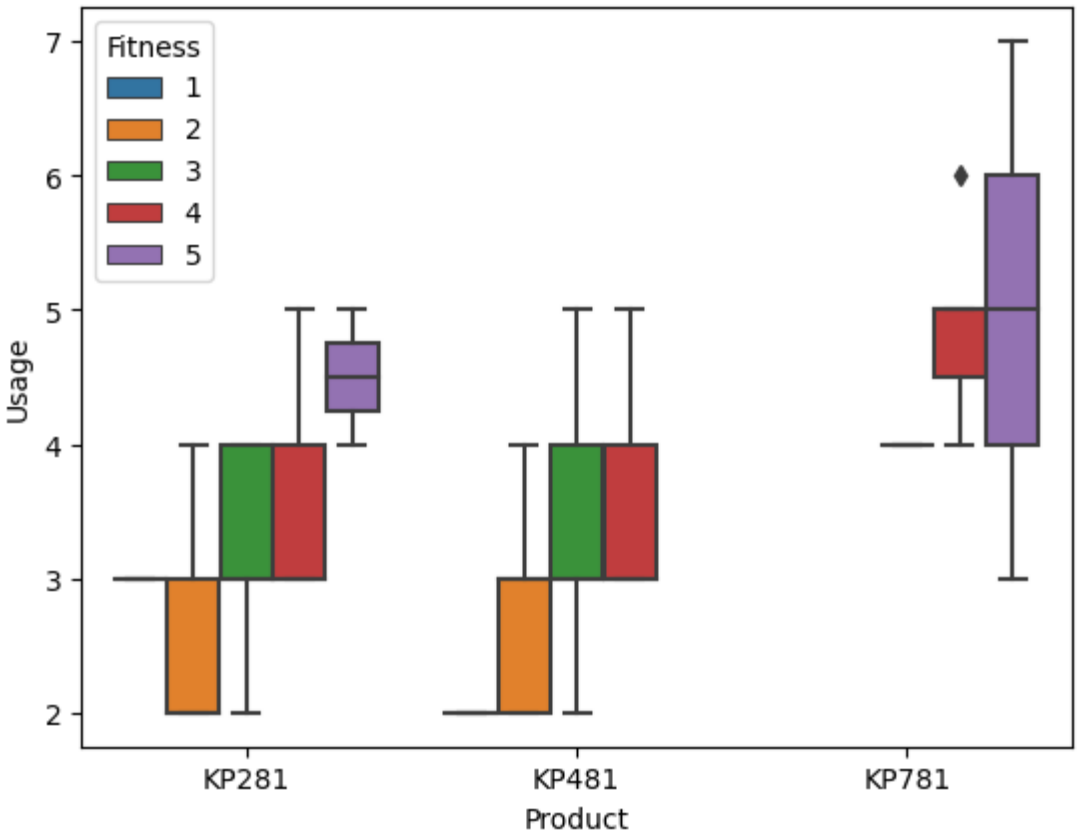


In [318…
```
sns.catplot
```

Out[318]:
```
<function seaborn.categorical.catplot(*, x=None, y=None, hue=None, data=None, row=
None, col=None, col_wrap=None, estimator=<function mean at 0x000002C12C8DB1F0>, ci
=95, n_boot=1000, units=None, seed=None, order=None, hue_order=None, row_order=Non
e, col_order=None, kind='strip', height=5, aspect=1, orient=None, color=None, pale
tte=None, legend=True, legend_out=True, sharex=True, sharey=True, margin_titles=Fa
lse, facet_kws=None, **kwargs)>
```

In [328…
```
#Comparison of usage and Fitness with Product
sns.boxplot(data = df, x = df["Product"], y = df["Usage"], hue = df["Fitness"] )
```

Out[328]:
```
<AxesSubplot:xlabel='Product', ylabel='Usage'>
```



KP781 has advanced features, hence people who bought the KP781 and used 4-5 times a week are in excellent shape of fitness.

In [329…
```
df
```

Out[329]:

	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

In [369…
```
ser = pd.DataFrame( data = [["KP281", 1500], ["KP481",  1750], ["KP781", 2500]], co
ser
```

Out[369]:

	Product	Cost
0	KP281	1500
1	KP481	1750
2	KP781	2500

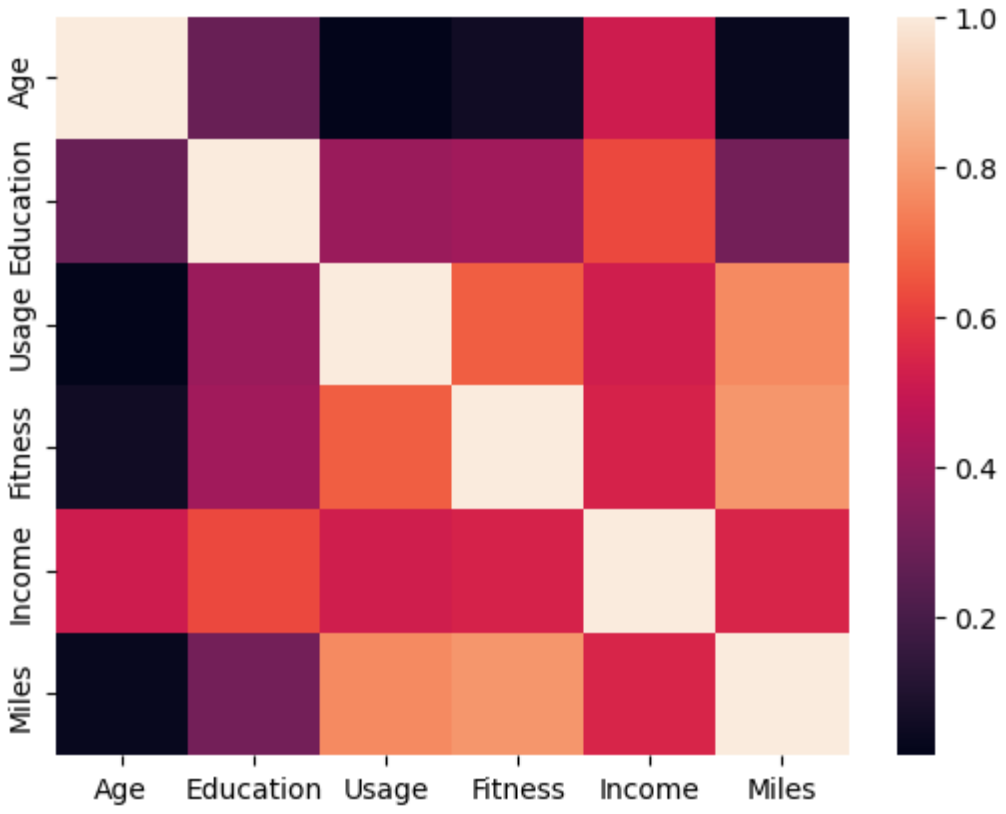
6.Recommendations and Insights: 1. The features and cost are higher in KP781 > KP481 > KP281. People with higher income(especially partnered)has higher chance of buying KP781. So when married people come to the store, we can explain the advance feature and show them statistical results of fitness of KP781 where fitness rate is higher in this product and try to sale. 2. Number of male who bought the tradmills are higher than female which could be more of health conciousness. So If the customer is around the age of 20-28, help them buy the correct tradmill. 3. Usage and fitness is directly proportional, so better the usage better will be the fitness. 4. People who are in healthy in shape, and have good eduscation background tend to buy KP781 with advanced features

In [388…
```
sns.heatmap
```

Out[388]:
```
<function seaborn.matrix.heatmap(data, *, vmin=None, vmax=None, cmap=None, center=
None, robust=False, annot=None, fmt='.2g', annot_kws=None, linewidths=0, linecolor
='white', cbar=True, cbar_kws=None, cbar_ax=None, square=False, xticklabels='aut
o', yticklabels='auto', mask=None, ax=None, **kwargs)>
```

In [391…
```
sns.heatmap(df.corr())
```

Out[391]:
```
<AxesSubplot:>
```



In [ ]: