importing python libraries

In [7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Importing Dataset

In [10]:
```
df = pd.read_csv('D:\\Learning\\scaler data science\\ProbAndStats\\AerofitBusinessCase\\Aerofit.csv')
print(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
```

Analysing the dataset

In [11]:
```
print(df.columns)
print(df.shape)
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
(180, 9)
```

In [49]:
```
print(df.info())
print(df.describe(include="all"))
```

```
<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
None
       Product         Age Gender   Education MaritalStatus       Usage  \
count      180  180.000000    180  180.000000           180  180.000000   
unique       3         NaN      2         NaN             2         NaN   
top      KP281         NaN   Male         NaN     Partnered         NaN   
freq        80         NaN    104         NaN           107         NaN   
mean       NaN   28.788889    NaN   15.572222           NaN    3.455556   
std        NaN    6.943498    NaN    1.617055           NaN    1.084797   
min        NaN   18.000000    NaN   12.000000           NaN    2.000000   
25%        NaN   24.000000    NaN   14.000000           NaN    3.000000   
50%        NaN   26.000000    NaN   16.000000           NaN    3.000000   
75%        NaN   33.000000    NaN   16.000000           NaN    4.000000   
max        NaN   50.000000    NaN   21.000000           NaN    7.000000   

           Fitness         Income       Miles  
count   180.000000     180.000000  180.000000  
unique         NaN            NaN         NaN  
top            NaN            NaN         NaN  
freq           NaN            NaN         NaN  
mean      3.311111   53719.577778  103.194444  
std       0.958869   16506.684226   51.863605  
min       1.000000   29562.000000   21.000000  
25%       3.000000   44058.750000   66.000000  
50%       3.000000   50596.500000   94.000000  
75%       4.000000   58668.000000  114.750000  
max       5.000000  104581.000000  360.000000  
```

In [13]:

```python
print(df.isnull().sum())
print(df.nunique())
```

```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

In [47]:
```
grouped = df.groupby('Product').size()
grouped
```

Out[47]:
```
Product
KP281    80
KP481    60
KP781    40
dtype: int64
```

In [48]:
```
print("probablity for KP281 sold : "+ str(round(80/180,2)))
print("probablity for KP481 sold : "+ str(round(60/180,2)))
print("probablity for KP781 sold : "+ str(round(40/180,2)))
```

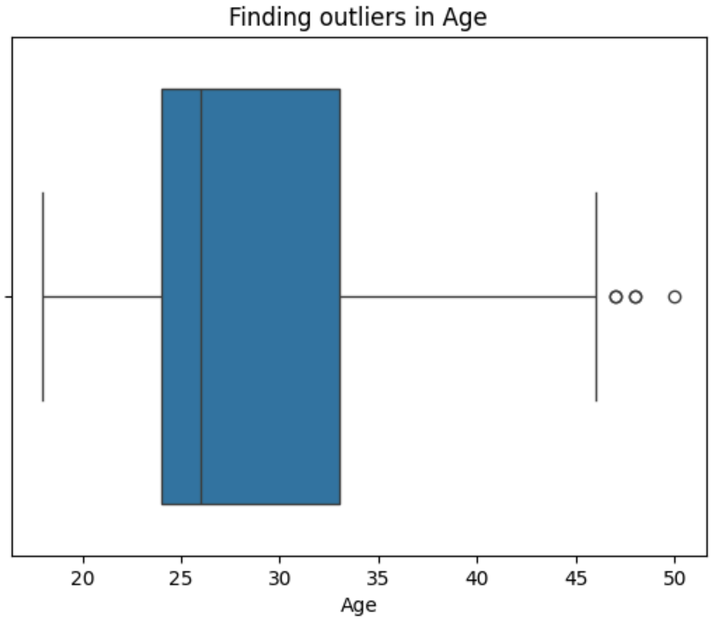
```
probablity for KP281 sold : 0.44
probablity for KP481 sold : 0.33
probablity for KP781 sold : 0.22
```

Inference after analysing dataset:

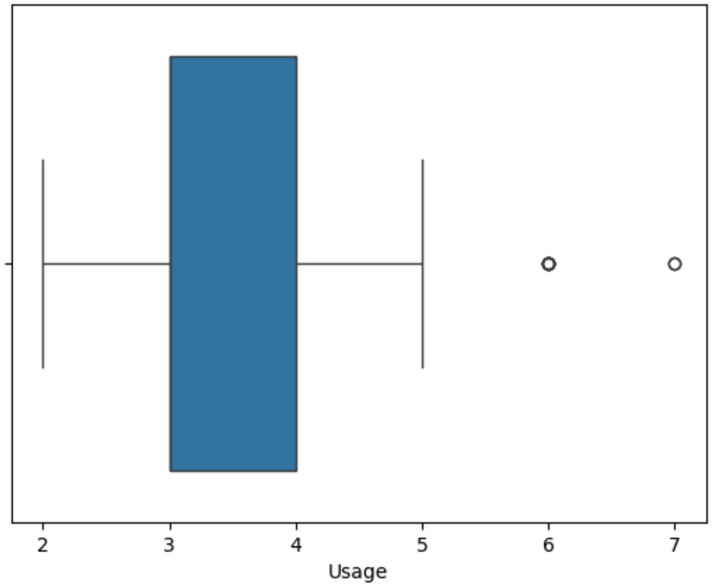
The dataset has total of 180 rows and 9 columns. There are no null values or missing values across any of the columns. There are 3 categorical columns and 6 integer columns. The data is present among people from age group of 18-50. In the Product categorical column we can clearly infer that there are 3 different type of products with KP281 among the most sold. We can also infer that there are more representation of male and partnered people when it comes to buying thr Products. The standard deviations for the "Income" and "Miles" variables are notably high, suggesting the possible presence of outliers in these data points.

##

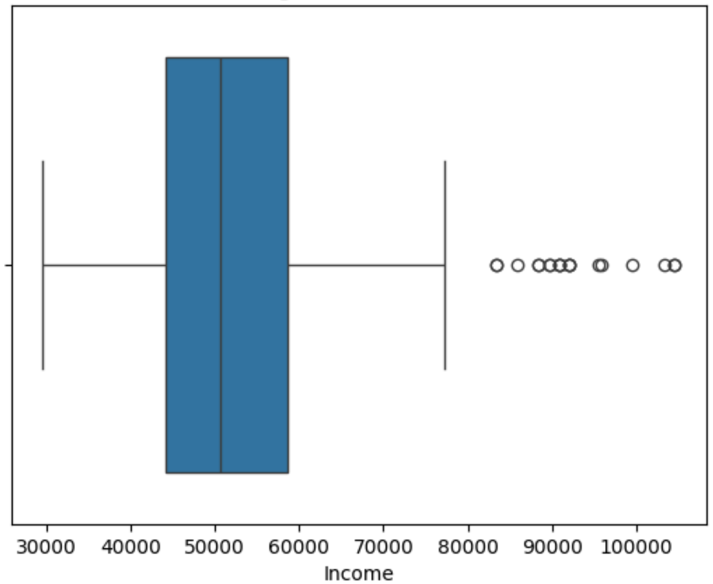
In [18]:
```
columns = [ 'Age', 'Usage', 'Income', 'Miles']
for column in columns:
  sns.boxplot(x=column, data=df)
  plt.title(f'Finding outliers in {column}')
  plt.show()
```

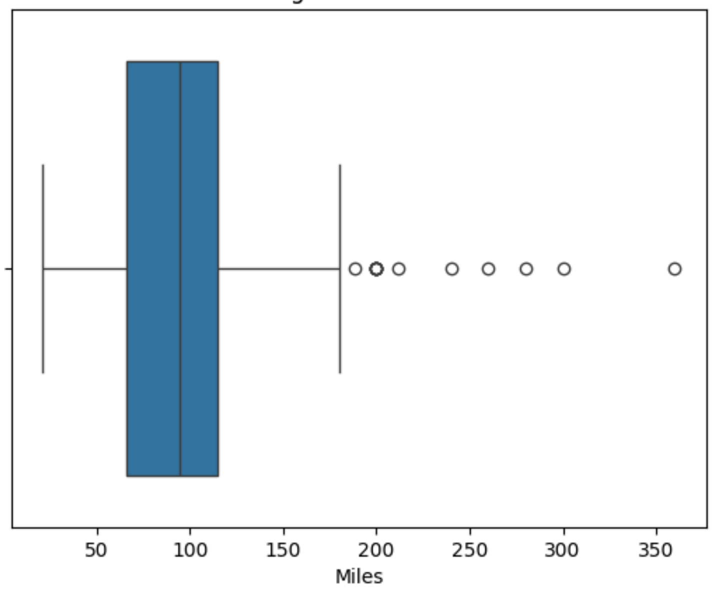


Finding outliers in Usage



Finding outliers in Income





In [56]:
```
income_25 = df.Income.quantile(0.25)
income_75 = df.Income.quantile(0.75)
income_median = df.Income.median()

lower_limit = income_25 - 1.5*(income_75 - income_25)
upper_limit = income_75 + 1.5*(income_75 - income_25)

print(f'Lower limit: {lower_limit}\nUpper limit: {upper_limit}\nMedian: {income_median}')
# print(len(df[df['Income']>upper_limit])
print(f"Outliers: {round((len(df[df['Income']>upper_limit])/len(df))*100,2)}%")
```

```
Lower limit: 22144.875
Upper limit: 80581.875
Median: 50596.5
Outliers: 10.56%
```

In [57]:
```
miles_25 = df.Miles.quantile(0.25)
miles_75 = df.Miles.quantile(0.75)
miles_median = df.Miles.median()

lower_limit = miles_25 - 1.5*(miles_75 - miles_25)
upper_limit = miles_75 + 1.5*(miles_75 - miles_25)

print(f'Lower limit: {lower_limit}\nUpper limit: {upper_limit}\nMedian: {miles_median}')
print(f"Outliers: {round((len(df[df['Miles']>upper_limit])/len(df))*100,2)}%")
```

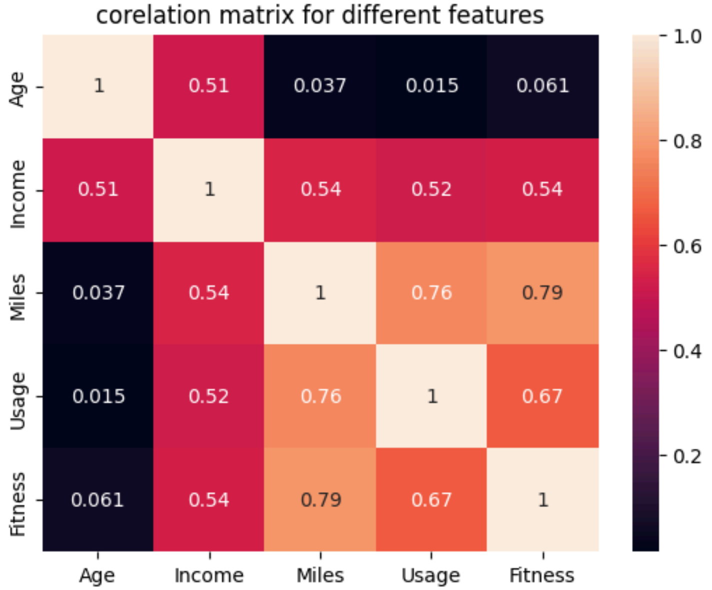
```
Lower limit: -7.125
Upper limit: 187.875
Median: 94.0
Outliers: 7.22%
```

Inference from Outliers:

From the above boxplot it is clear that AGE and Usage have very few outliers. Income and miles have high number of ouliers. from the above calculations we can see that Income has 10.56% outliers and Miles has 7.22% outliers.

##

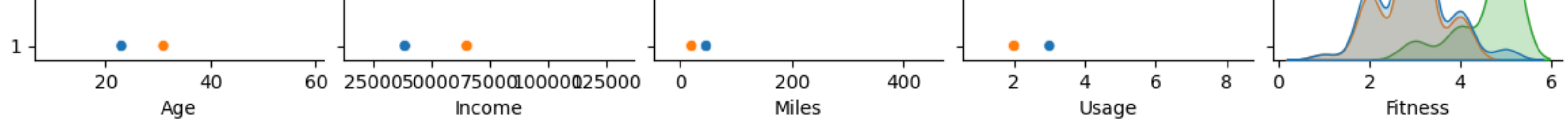
In [24]:
```
df1 = df[['Age','Income','Miles','Usage','Fitness']]
sns.heatmap(df1.corr(),annot = True)
plt.title('corelation matrix for different features')
plt.show()
```



In [39]:
```
df1 = df[['Age','Income','Miles','Usage','Fitness','Product']]
plt.figure(figsize=(11,11))
sns.pairplot(data=df1, hue='Product')
plt.show()
```

<Figure size 1100x1100 with 0 Axes>


Age
Income
Miles
Usage
Fitness
Product
KP281
KP481
KP781



From the above heatmap and pairplots we can infer that:

1. There is a positive correlation between age and income.
2. There is positive correlation between usage, miles and fitness.
3. There is also a positive relationship between income and miles.
4. people who are more fit, have more income, have more usage tends to buy product 'KP781'.
5. people who are less in age, have less income, are not fit are moreinclined towards product 'KP281'.

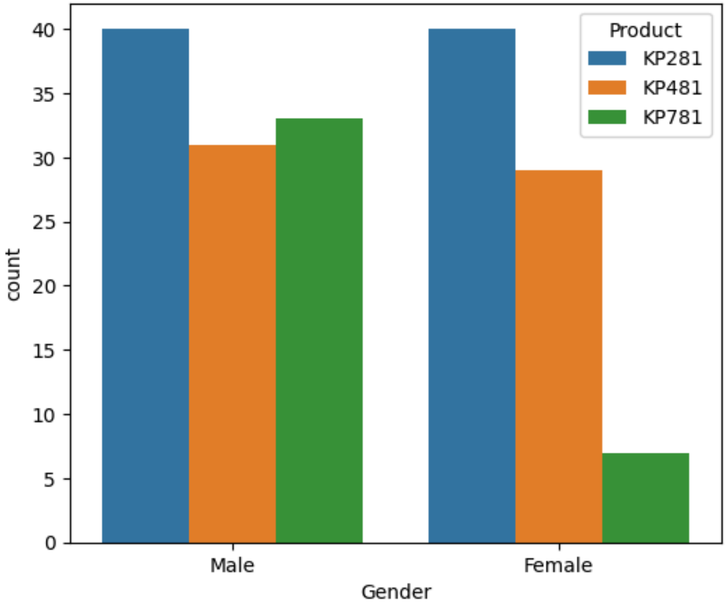
##

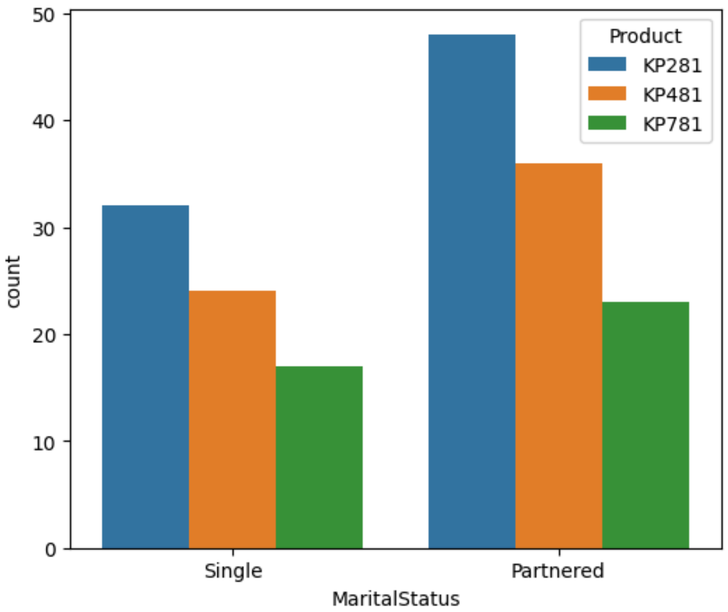
In [ ]:

```
df["Gender"]=df["Gender"].astype("category")
df["MaritalStatus"]=df["MaritalStatus"].astype("category")
df["Product"]=df["Product"].astype("category")
```

In [46]:

```
columns = [ 'Gender', 'MaritalStatus']
for column in columns:
    plt.figure(figsize=(6,5))
    sns.countplot(x=column, hue='Product', data=df)
    plt.xlabel(f'{column}')
    plt.show()
```





In [58]:
```
grouped = df.groupby('Gender').size()
grouped
```

Out[58]:
```
Gender
Female     76
Male      104
dtype: int64
```

In [60]:
```
print("probablity for KP281 sold to male : "+ str(round(40/104,2)))
print("probablity for KP481 sold to male: "+ str(round(31/104,2)))
print("probablity for KP781 sold to male: "+ str(round(33/104,2)))
print("probablity for KP281 sold to female : "+ str(round(40/76,2)))
print("probablity for KP481 sold to female: "+ str(round(29/76,2)))
print("probablity for KP781 sold to female: "+ str(round(7/76,2)))
```

```
probablity for KP281 sold to male : 0.38
probablity for KP481 sold to male: 0.3
probablity for KP781 sold to male: 0.32
probablity for KP281 sold to female : 0.53
probablity for KP481 sold to female: 0.38
probablity for KP781 sold to female: 0.09
```

Getting More insights on Categorical data :

1. Customers in a partnered relationship are more inclined to purchase treadmill models KP281, KP481, and KP781 compared to those who are single
2. Regarding the product KP281 and KP481, both males and females are equally inclined. but for KP781 males are more inclined in buying.

##

Prefered customer for KP281 :

1. Both gender
2. Prefered Partnered
3. Aged between 18-30
4. fitness level - 3

5. usage- 3/week
6. miles - between 50-100 miles
7. income - 30-60k

Prefered customer for KP481 :

1. Both gender
2. Prefered Partnered
3. Aged between 20-30
4. fitness level - 3
5. usage- 3-4/week
6. miles - between 70-100 miles
7. income - 30-60k

Prefered customer for KP781 :

1. prefered male
2. Prefered Partnered
3. Aged between 20-30
4. fitness level - 5
5. usage- 4-5/week
6. miles - above 120 miles
7. income - above 60k

##

Recommendations :

1. KP281 caters to mostly beginner level runner, followed by KP481 to mid level runners and KP781 to regulars and more usage customers.
2. Higher income people takes more interest in KP781 to showcase advanced preferences.
3. Give discount to females for KP781 product in order to increase female purchaces.
4. to customers coming to buy product first time cater them towards KP281 because of the price point and other benefits.
5. to customers who are pro runners should be catered towards KP781 product which showcase advanced benefits.

END