Problem Statement ¶

Analyze data of customers who purchased treadmills from Aerofit and help them decide target audience for various treadmills based on features such as age of customer, Income of customer, Usage of tradmill per week, Expectation of running in miles per week, self rated fitness of customer, education of customer

Also, define probabilities of custome buying specific product based on different features

In [1]:
```
## import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
## load data
df = pd.read_csv("Aerofit.csv")
```

In [3]:
```
## check data
df.head()
```

Out[3]:

	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

In [4]:
```
### check shape of data
df.shape
```

Out[4]: (180, 9)

In [5]:
```
## check basic info
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
```

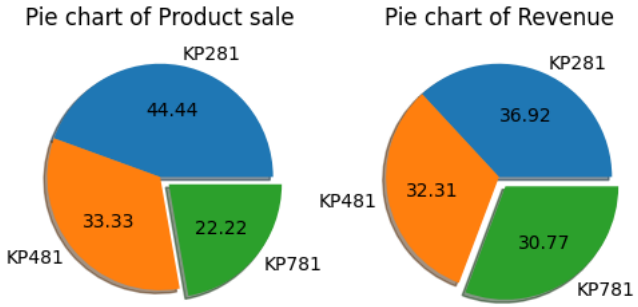
3 categorical and 6 numerical columns are present and none of them have any null value

In [6]:
```
### check number of sell of each product and how much they are contributing to revenue
prod = pd.DataFrame({"count":df["Product"].value_counts(),
                     "price_per_qty":[1500, 1750, 2500],
                     "pct_of_total_sale":round(df["Product"].value_counts()/df.shape[0]*100,2)}
                    )
prod["Revenue"] = prod["count"]*prod["price_per_qty"]
prod["pct_revenue"] = round(prod["Revenue"]/prod["Revenue"].sum()*100,2)
prod
```

Out[6]:

	count	price_per_qty	pct_of_total_sale	Revenue	pct_revenue
KP281	80	1500	44.44	120000	36.92
KP481	60	1750	33.33	105000	32.31
KP781	40	2500	22.22	100000	30.77

In [7]:
```
##### visual representation
plt.figure(figsize=(6,6))
plt.subplot(1,2,1)
plt.pie(prod["count"], labels=prod.index, explode=[0,0,0.1],shadow=True,autopct='%.2f')
plt.title("Pie chart of Product sale")

plt.subplot(1,2,2)
plt.pie(prod["pct_revenue"], labels=prod.index, explode=[0,0,0.1],shadow=True,autopct='%.2f')
plt.title("Pie chart of Revenue")
plt.show()
```



KP281 is contributing to 45% of total sales and generating 37% of revenue

KP481 is contributing to 33% of total sales and generating 32% of revenue

KP781 is contributing to 22% of total sales and generating 31% of revenue

In [8]:
```
### let's Check basic stats of customers
df.describe()
```

Out[8]:

	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

Customers age is between 18 to 50 with 75 percent customers of age less than 33

Customers are earning between 30k to 104k with 75% customers earning less than 60k

75% of customers are expecting to run less than 115 miles per week

In [9]:
```
####customer categorisation based on Gender
Gender = pd.DataFrame({"count":df["Gender"].value_counts(),
                       "pct_of_total_sale":round(df["Gender"].value_counts()/df.shape[0]*100,2)}
                      )
Gender
```

Out[9]:

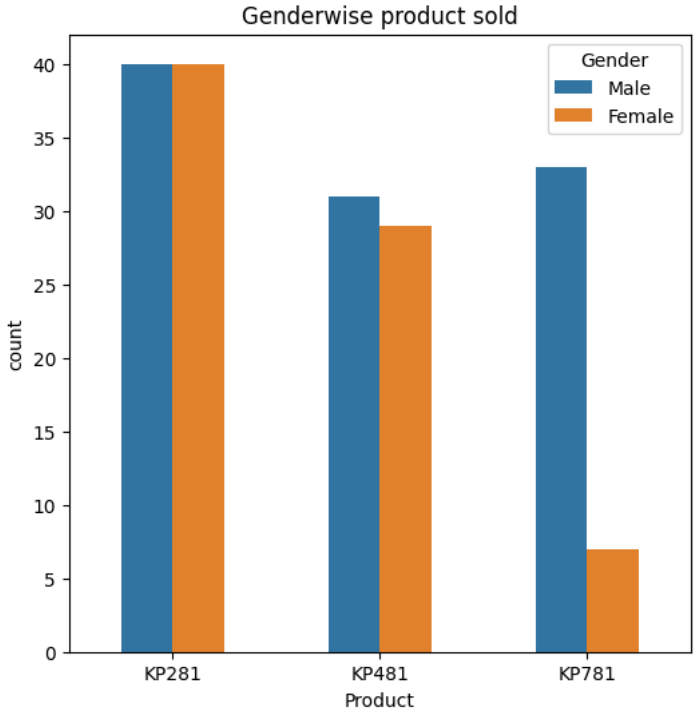
	count	pct_of_total_sale
Male	104	57.78
Female	76	42.22

In [10]:
```
## count of male and female
gender_prod_cont = pd.crosstab(df["Product"], df["Gender"])
gender_prod_cont
```

Out[10]:

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

In [11]:
```
## plot bar graph
plt.figure(figsize=(6,6))
sns.countplot(data=df, x="Product", hue="Gender",width=0.5)
plt.title("Genderwise product sold")
plt.show()
```



In [12]:
```
## lets check probabilities as well
prob_KP281 = round(80/180,2)
prob_KP481 = round(60/180,2)
prob_KP781 = round(40/180,2)
prob_KP281_Male = round(40/180,2)
prob_KP281_Female = round(40/180,2)
prob_KP481_Male = round(31/180,2)
prob_KP481_Female = round(29/180,2)
prob_KP781_Male = round(33/180,2)
prob_KP781_Female = round(7/180,2)
prob_Female = round(gender_prod_cont["Female"].sum()/df.shape[0],2)
prob_Male = round(gender_prod_cont["Male"].sum()/df.shape[0],2)
```

In [13]:
```
print("Probability of customer buying KP281 given Male: ",round(prob_KP281_Male/prob_Male,2))
print("Probability of customer buying KP481 given Male: ",round(prob_KP481_Male/prob_Male,2))
print("Probability of customer buying KP781 given Male: ",round(prob_KP781_Male/prob_Male,2))
print("Probability of customer buying KP281 given Female: ",round(prob_KP281_Female/prob_Female,2))
print("Probability of customer buying KP481 given Female: ",round(prob_KP481_Female/prob_Female,2))
print("Probability of customer buying KP781 given Female: ",round(prob_KP781_Female/prob_Female,2))
```

```
Probability of customer buying KP281 given Male:  0.38
Probability of customer buying KP481 given Male:  0.29
Probability of customer buying KP781 given Male:  0.31
Probability of customer buying KP281 given Female:  0.52
Probability of customer buying KP481 given Female:  0.38
Probability of customer buying KP781 given Female:  0.1
```

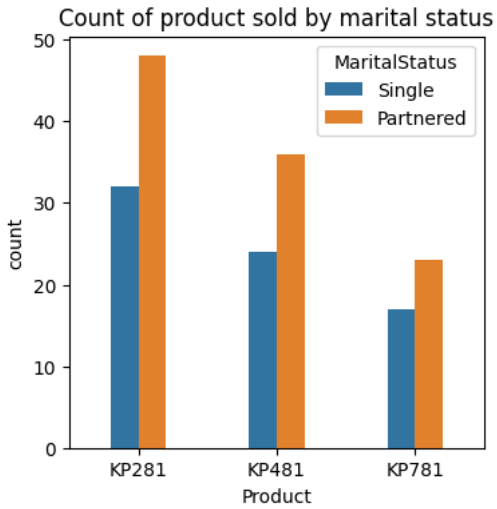
Number of males and females buying KP281 and KP481 are almost same but number of males buying KP781 are 5 times number of females buying KP781 and probability of male buying KP781 is also 3 times probability of female buying KP781

In [14]:
```
####Impact of marital status
Marital_status = pd.DataFrame({"count":df["MaritalStatus"].value_counts(),
                     "pct_of_total_sale":round(df["MaritalStatus"].value_counts()/df.shape[0]*100,2)}
                    )
Marital_status
```

Out[14]:

	count	pct_of_total_sale
Partnered	107	59.44
Single	73	40.56

In [15]:
```
## plot bar graph
plt.figure(figsize=(4,4))
sns.countplot(data=df, x="Product", hue="MaritalStatus",width=0.4)
plt.title("Count of product sold by marital status")
plt.show()
```



In [16]:
```
# contingency table between product and marital status
marital_prod_cont = pd.crosstab(df["Product"], df["MaritalStatus"])
marital_prod_cont
```

Out[16]:

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

In [17]:
```
## calculate probabilities
prob_KP281_Partnered = round(48/180,2)
prob_KP281_Single = round(32/180,2)
prob_KP481_Partnered = round(36/180,2)
prob_KP481_Single = round(24/180,2)
prob_KP781_Partnered = round(23/180,2)
prob_KP781_Single = round(17/180,2)
prob_Partnered = round(marital_prod_cont["Partnered"].sum()/df.shape[0],2)
prob_Single = round(marital_prod_cont["Single"].sum()/df.shape[0],2)
```

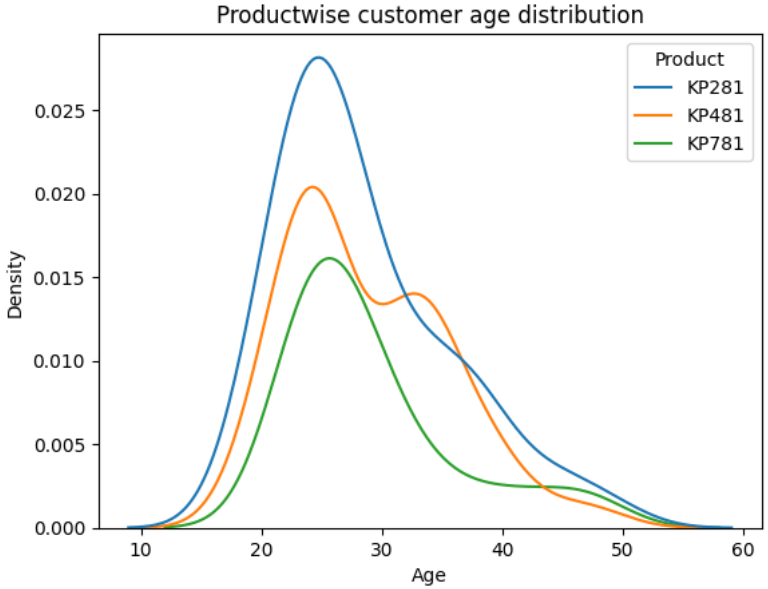
In [18]:
```
print("Probability of customer buying KP281 given Single: ",round(prob_KP281_Single/prob_Single,2))
print("Probability of customer buying KP481 given Single: ",round(prob_KP481_Single/prob_Single,2))
print("Probability of customer buying KP781 given Single: ",round(prob_KP781_Single/prob_Single,2))
print("Probability of customer buying KP281 given partnered: ",round(prob_KP281_Partnered/prob_Partnered,2))
print("Probability of customer buying KP481 given partnered: ",round(prob_KP481_Partnered/prob_Partnered,2))
print("Probability of customer buying KP781 given partnered: ",round(prob_KP781_Partnered/prob_Partnered,2))
```

```
Probability of customer buying KP281 given Single:  0.44
Probability of customer buying KP481 given Single:  0.32
Probability of customer buying KP781 given Single:  0.22
Probability of customer buying KP281 given partnered:  0.46
Probability of customer buying KP481 given partnered:  0.34
Probability of customer buying KP781 given partnered:  0.22
```

Number of partnered customers are more than single customers

44% of single customers bought KP281 and 46% of partnered customers bought KP281

In [19]:
```
### Age based analysis
sns.kdeplot(data = df, x = "Age", hue="Product")
plt.title("Productwise customer age distribution")
plt.show()
```



In [20]:
```
df["Age_cat"] = pd.cut(df['Age'], bins=[18, 30, 45, 60], include_lowest=True, labels=['Young', 'Middle', 'Old'])
## contingency table
Age_prod_cont = pd.crosstab(df["Product"], df["Age_cat"])
Age_prod_cont
```

Out[20]:

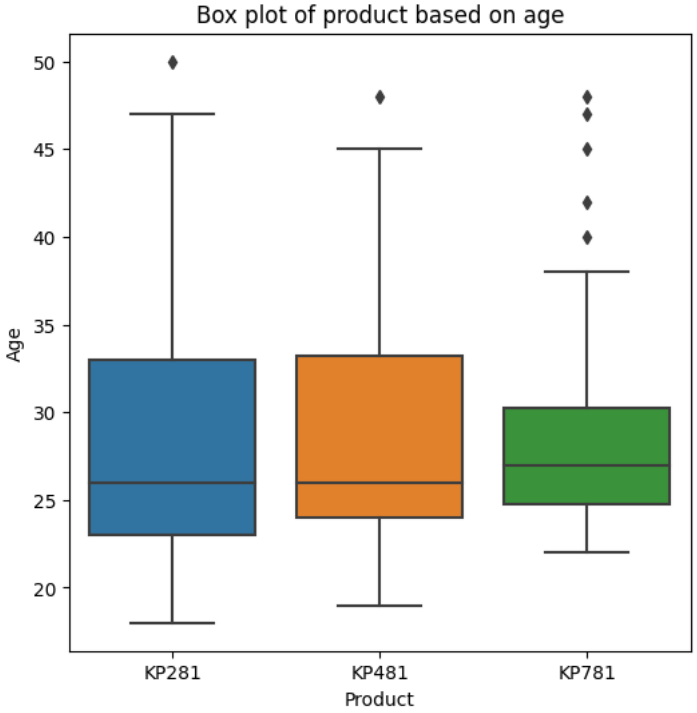
Age_cat	Young	Middle	Old
Product			
KP281	55	22	3
KP481	35	24	1
KP781	30	8	2

In [21]:

```
prob_young = Age_prod_cont["Young"].sum()/df.shape[0]
prob_middle_aged = Age_prod_cont["Middle"].sum()/df.shape[0]
prob_old = Age_prod_cont["Old"].sum()/df.shape[0]

print("Prob of young customers", prob_young)
print("Prob of middle aged customers", prob_middle_aged)
print("Prob of old customers", prob_old)

print("Prob of young customer buying KP281",round((55/180)/prob_young,2))
print("Prob of young customer buying KP481",round((35/180)/prob_young,2))
print("Prob of young customer buying KP781",round((30/180)/prob_young,2))

print("Prob of middle aged customer buying KP281",round((22/180)/prob_middle_aged,2))
print("Prob of middle aged customer buying KP481",round((24/180)/prob_middle_aged,2))
print("Prob of middle aged customer buying KP781",round((8/180)/prob_middle_aged,2))

print("Prob of old customer buying KP281",round((3/180)/prob_old,2))
print("Prob of old customer buying KP481",round((1/180)/prob_old,2))
print("Prob of old customer buying KP781",round((2/180)/prob_old,2))
```

```
Prob of young customers 0.6666666666666666
Prob of middle aged customers 0.3
Prob of old customers 0.03333333333333333
Prob of young customer buying KP281 0.46
Prob of young customer buying KP481 0.29
Prob of young customer buying KP781 0.25
Prob of middle aged customer buying KP281 0.41
Prob of middle aged customer buying KP481 0.44
Prob of middle aged customer buying KP781 0.15
Prob of old customer buying KP281 0.5
Prob of old customer buying KP481 0.17
Prob of old customer buying KP781 0.33
```

In [22]:
```
plt.figure(figsize=(6,6))
sns.boxplot(x='Product', y='Age', data=df)
plt.title("Box plot of product based on age")
plt.show()
```



In [23]:
```
###### from customers who are buying KP781, 75% of customers have age less than 30 year
### there are 5 customers buying KP781 with age more than 38
```

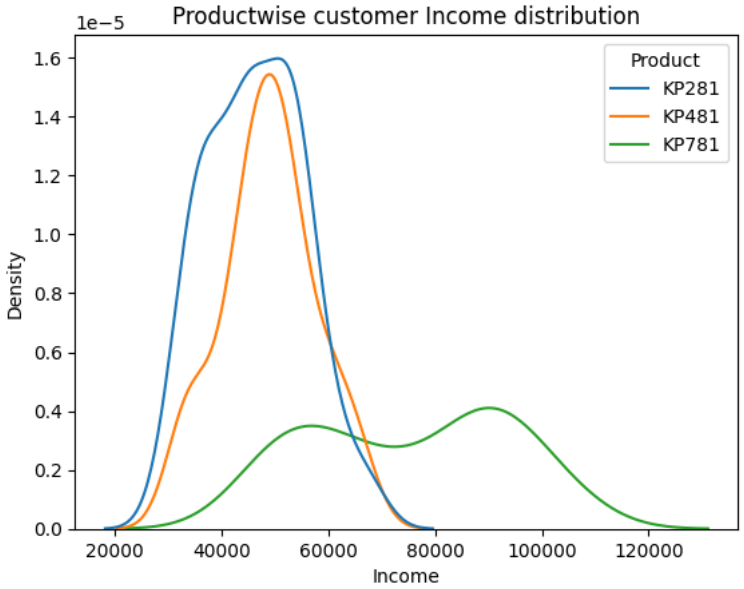
In [24]:
```
### there are 5 customers buying KP781 with age more than 38
#### let's find out this customers
KP_781  = df.loc[df["Product"]=="KP781",:]
age_kp781 = KP_781["Age"].describe()
iqr = age_kp781["75%"] - age_kp781["25%"]
upper = age_kp781["75%"]+1.5*iqr
KP_781.loc[KP_781["Age"]>upper,:]
```

Out[24]:

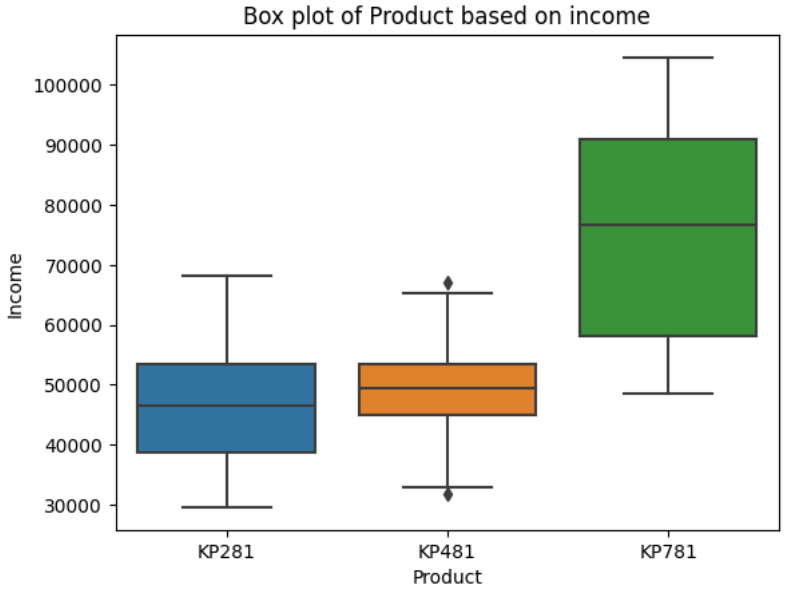
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_cat
175	KP781	40	Male	21	Single	6	5	83416	200	Middle
176	KP781	42	Male	18	Single	5	4	89641	200	Middle
177	KP781	45	Male	16	Single	5	5	90886	160	Middle
178	KP781	47	Male	18	Partnered	4	5	104581	120	Old
179	KP781	48	Male	18	Partnered	4	5	95508	180	Old

Great to see these customers are setting higher fitness goals

In [25]:
```
### income based analysis
sns.kdeplot(data = df, x = "Income", hue="Product")
plt.title("Productwise customer Income distribution")
plt.show()
```



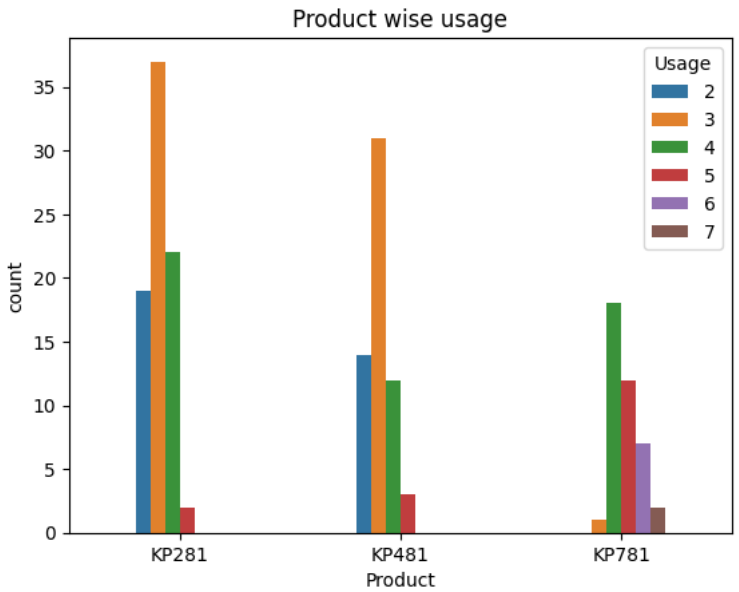
In [26]:
```
sns.boxplot(x='Product', y='Income', data=df)
plt.title("Box plot of Product based on income")
plt.show()
```



Median salary customers buying KP281 and KP481 is almost same.

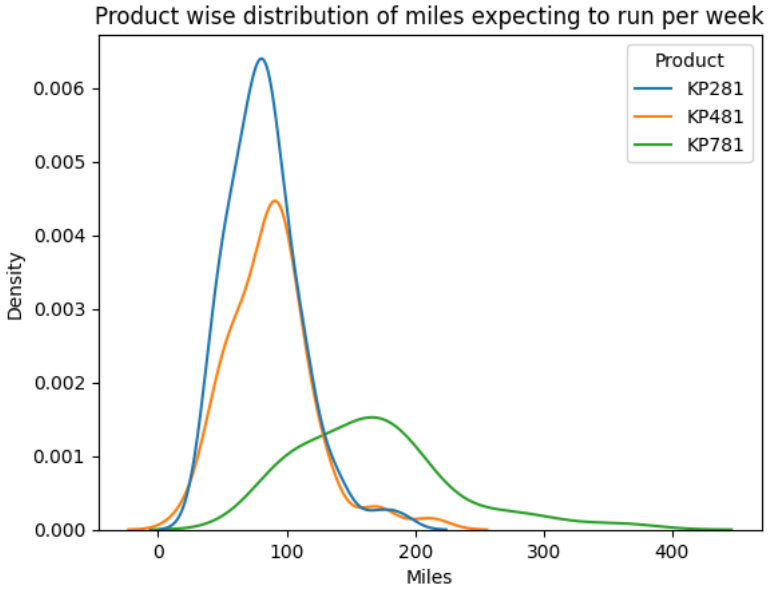
Customers with higher salary have bought KP781

In [27]:
```
sns.countplot(data=df, x="Product", hue="Usage",width=0.4)
plt.title("Product wise usage")
plt.show()
```



KP281 and KP481 are used 3 to 4 days per week while KP781 is used more than 4 days a week

In [28]:
```
sns.kdeplot(data = df, x = "Miles", hue="Product")
plt.title("Product wise distribution of miles expecting to run per week")
plt.show()
```



In [29]:
```
mean_miles = df.groupby(by=["Product"]).agg(
    product_sale=("Product","count"),
    Miles_mean = ("Miles","mean")
)
mean_miles
```

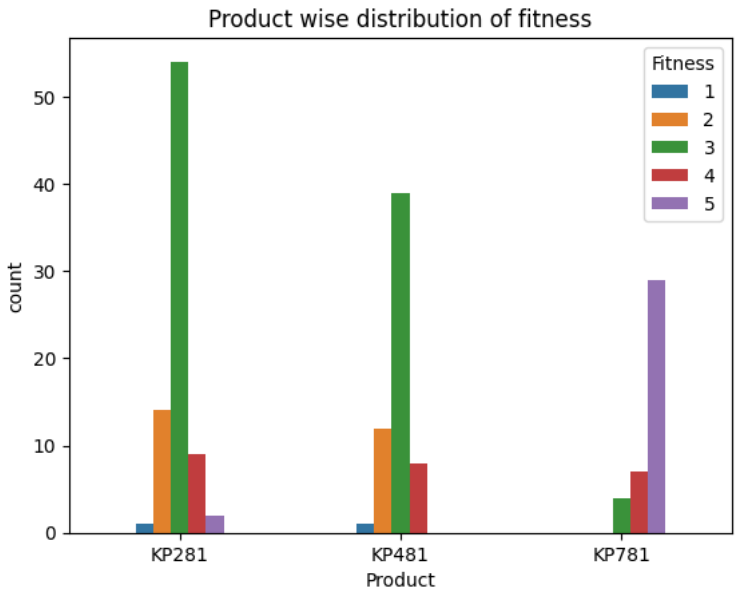
Out[29]:

Product	product_sale	Miles_mean
KP281	80	82.787500
KP481	60	87.933333
KP781	40	166.900000

Customers expecting to run 82 to 84 miles per week on avg bought KP281 and KP481

Customers expecting to run more than 166 miles per week bought KP781

In [30]:
```
sns.countplot(data=df, x="Product", hue="Fitness",width=0.4)
plt.title("Product wise distribution of fitness")
plt.show()
```

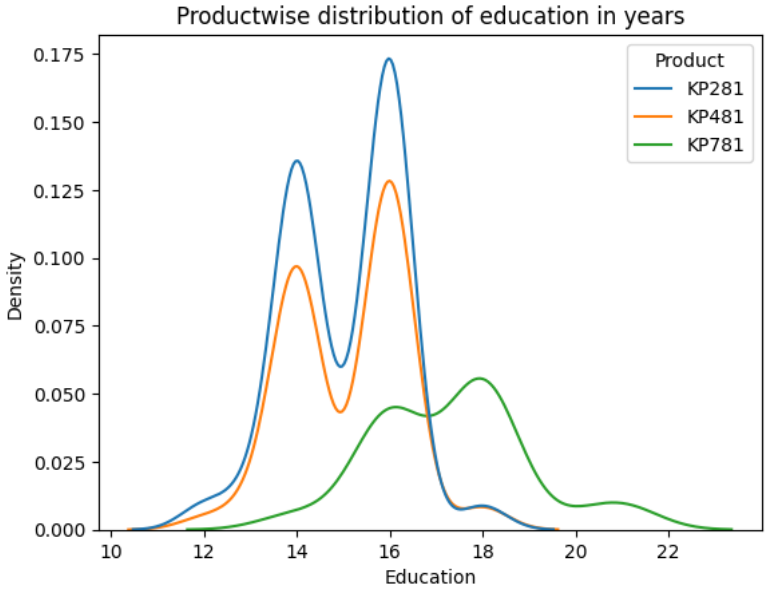


More than 50 customers using KP281 rated themselves as 3

Almost 40 customers using KP481 rated themselves as 3

Almost 30 customers using KP781 rated themselves as 5

In [31]:
```
sns.kdeplot(data = df, x = "Education", hue="Product")
plt.title("Productwise distribution of education in years")
plt.show()
```



In [32]:
```
Education = df.groupby(by=["Product"]).agg(
    product_sale=("Product","count"),
    Median_Education = ("Education","median")
)
Education
```

Out[32]:

Product	product_sale	Median_Education
KP281	80	16.0
KP481	60	16.0
KP781	40	18.0

Customers who have bought KP281 and KP481, have median education of 16 years

Customers who have bought KP781, have median education of 18 years

In [33]:
```
### lets see if there is any correlation bet features
df.corr()
```

```
C:\Users\Sail\AppData\Local\Temp\ipykernel_25832\340520403.py:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default
to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.corr()
```

Out[33]:

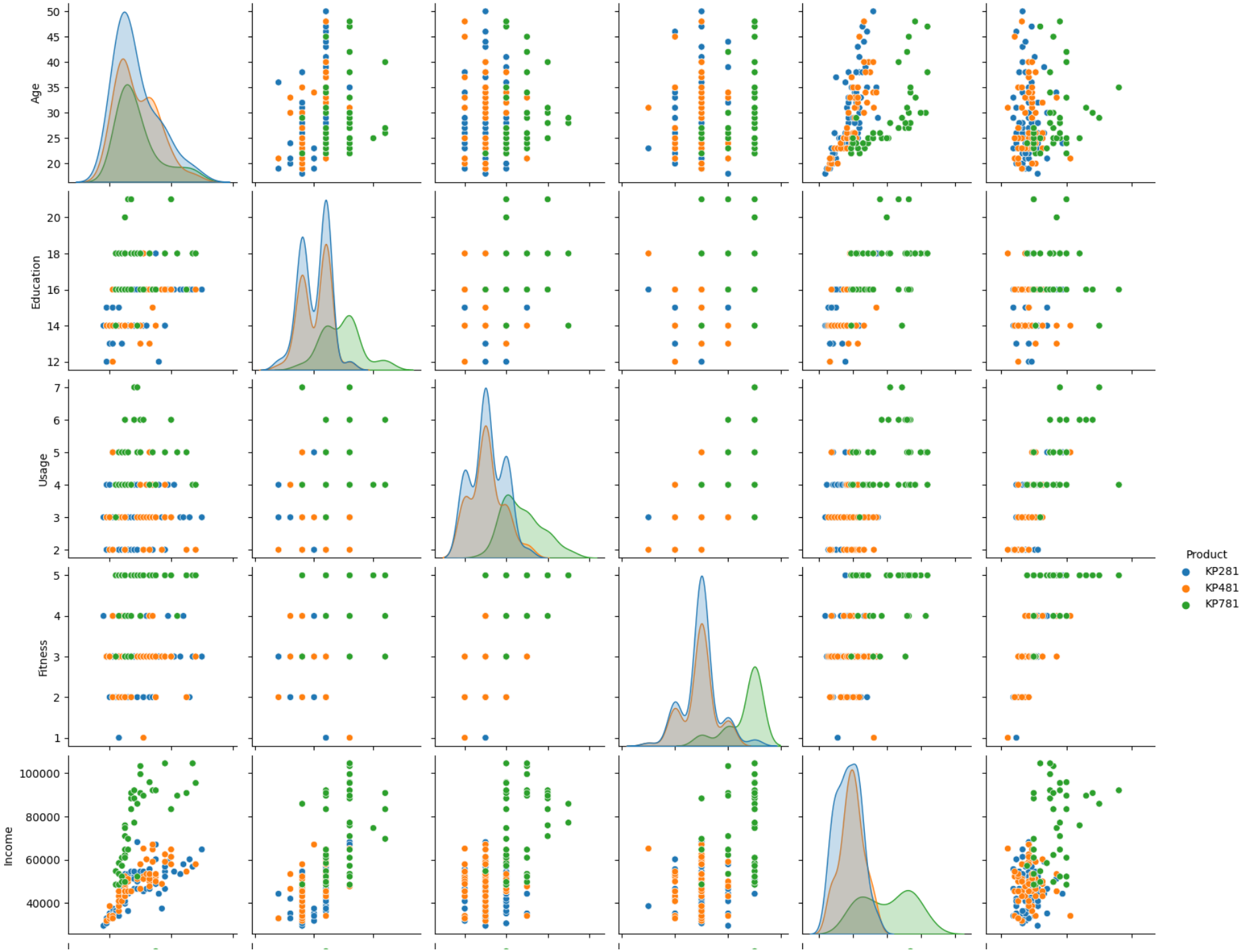
	Age	Education	Usage	Fitness	Income	Miles
Age	1.000000	0.280496	0.015064	0.061105	0.513414	0.036618
Education	0.280496	1.000000	0.395155	0.410581	0.625827	0.307284
Usage	0.015064	0.395155	1.000000	0.668606	0.519537	0.759130
Fitness	0.061105	0.410581	0.668606	1.000000	0.535005	0.785702
Income	0.513414	0.625827	0.519537	0.535005	1.000000	0.543473
Miles	0.036618	0.307284	0.759130	0.785702	0.543473	1.000000

Fitness and miles have strong positive correlation of 0.785

Usage and miles have strong positive correlation of 0.759

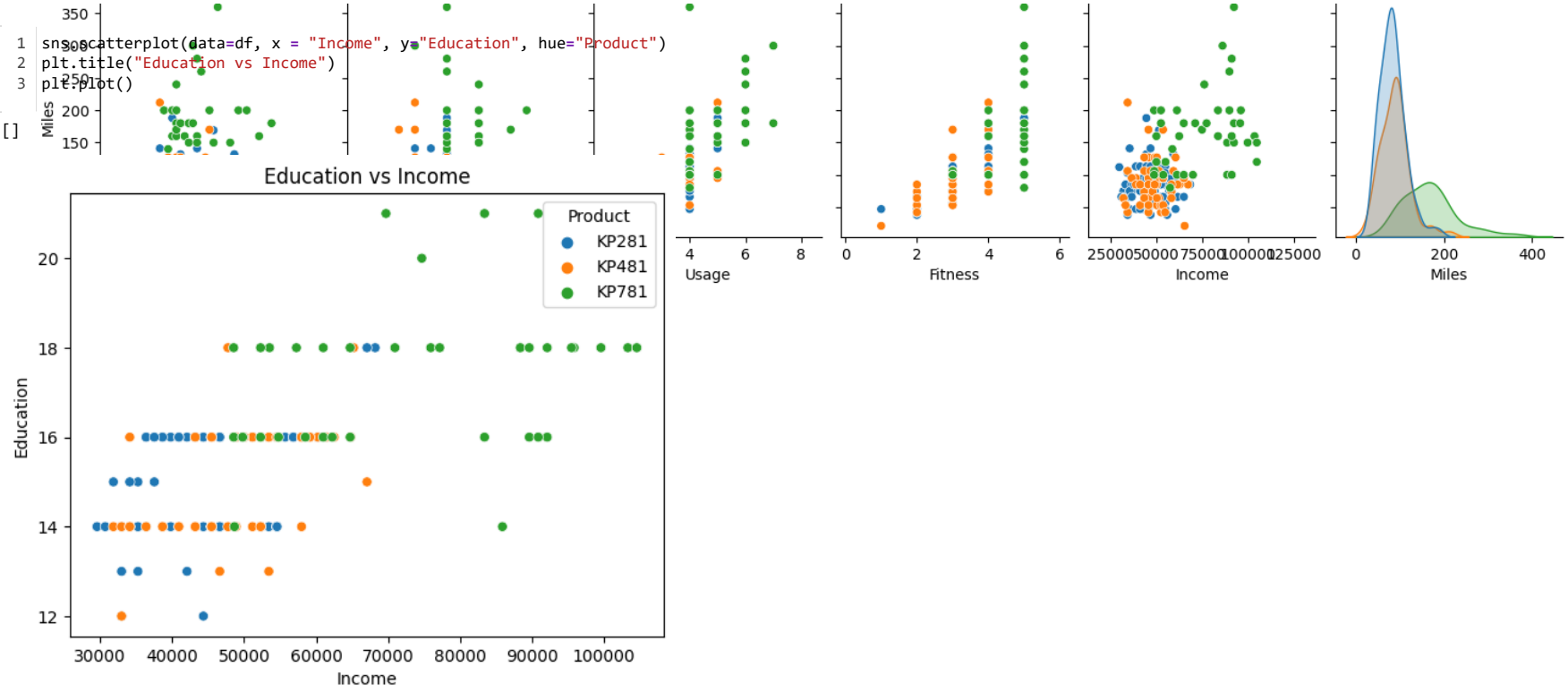
In [34]:
```
#### plot pairplot
sns.pairplot(data=df, hue = "Product")
```

Out[34]:
```
<seaborn.axisgrid.PairGrid at 0x25eb1dd3d60>
```


Age
Education
Usage
Fitness
Income
Product
KP281
KP481
KP781

In [35]:
```
sns.scatterplot(data=df, x = "Income", y="Education", hue="Product")
plt.title("Education vs Income")
plt.plot()
```

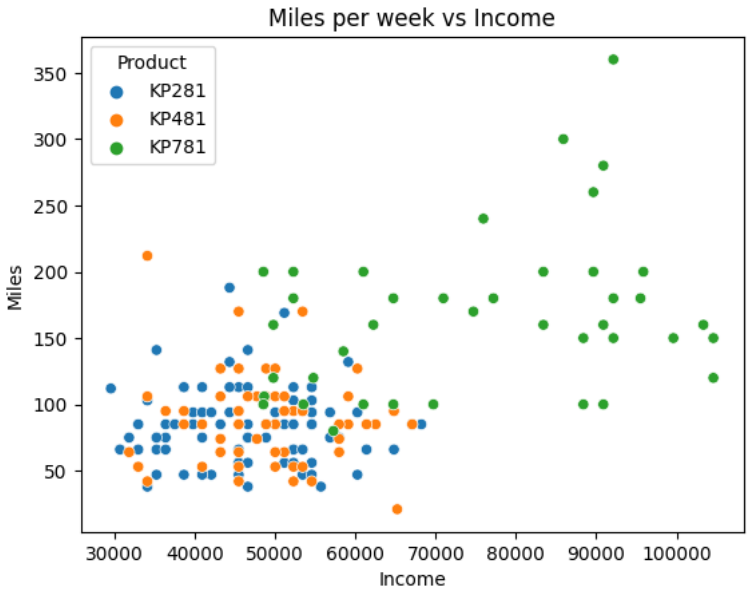
Out[35]: []



People With higher Education and more income bought KP781

In [36]:
```
sns.scatterplot(data=df, x = "Income", y="Miles", hue="Product")
plt.title("Miles per week vs Income")
plt.plot()
```

Out[36]: []



People with salary more than 70k and expecting to run more than 110 miles per week have bought KP781

In [37]:
```
##### customer categorisation on numerical features
df.groupby(by=["Product"]).agg(
    product_sale=("Product","count"),
    Age_median = ("Age","median"),
    Age_min = ("Age","min"),
    Age_max = ("Age","max"),
    Income_mean = ("Income","mean"),
    Miles_mean = ("Miles","mean"),
    Fitness_median = ("Fitness","median"),
    Usage_median = ("Usage","median"),
    Education_median = ("Education", "median")
)
```

Out[37]:

Product	product_sale	Age_median	Age_min	Age_max	Income_mean	Miles_mean	Fitness_median	Usage_median	Education_median
KP281	80	26.0	18	50	46418.025	82.787500	3.0	3.0	16.0
KP481	60	26.0	19	48	48973.650	87.933333	3.0	3.0	16.0
KP781	40	27.0	22	48	75441.575	166.900000	5.0	5.0	18.0

From above table it is hard to differentiate customers who are buying KP281 and KP481.

Customers who have bought KP781 have median age 27 years, avg income 75k, avg expected miles 167 miles per week, with median usage of 5 days per week and rated themselves as 5.

In [38]:
```
## detailed categorisation

df.groupby(by=["Product", "Gender", "MaritalStatus"]).agg(
    product_sale=("Product","count"),
    Age_median = ("Age","median"),
    Age_min = ("Age","min"),
    Age_max = ("Age","max"),
    Income_mean = ("Income","mean"),
    Miles_mean = ("Miles","mean"),
    Fitness_median = ("Fitness","median"),
    Usage_median = ("Usage","median"),
    Education_median = ("Education", "median")
)
```

Out[38]:

Product	Gender	MaritalStatus	product_sale	Age_median	Age_min	Age_max	Income_mean	Miles_mean	Fitness_median	Usage_median	Education_median
KP281	Female	Partnered	27	27.0	19	50	46153.777778	74.925926	3.0	3.0	14.0
		Single	13	26.0	22	44	45742.384615	78.846154	3.0	3.0	16.0
	Male	Partnered	21	30.0	20	47	50028.000000	80.190476	3.0	3.0	16.0
		Single	19	25.0	18	38	43265.842105	99.526316	3.0	3.0	14.0
KP481	Female	Partnered	15	31.0	20	40	49724.800000	94.000000	3.0	3.0	16.0
		Single	14	25.5	23	40	48920.357143	80.214286	3.0	3.0	15.0
	Male	Partnered	21	31.0	21	48	49378.285714	87.238095	3.0	3.0	16.0
		Single	10	25.0	19	34	47071.800000	91.100000	3.0	3.0	14.0
KP781	Female	Partnered	4	29.0	25	33	84972.250000	215.000000	5.0	5.5	18.0
		Single	3	24.0	23	26	58516.000000	133.333333	4.0	5.0	18.0
	Male	Partnered	19	27.0	24	48	81431.368421	176.315789	5.0	4.0	18.0
		Single	14	25.5	22	45	68216.428571	147.571429	5.0	4.5	16.0

Partenered females are buying KP281 twice as single females buying KP281

Partenered males are buying KP481 twice as single males buying KP481

Number of males buying KP781 are 4 times more than females buying KP781

Recommendations

1. Aerofit can target customer between age 18 to 35

2. For KP281 and KP481 target customer's income would be around 50k and for KP781 income would be 80k

3. Male with high education, high salary with expectation of running more than 150 miles per week and usage more than 5 days per week can be potential customer for KP781

4. Aerofit can consider to combine KP281 and KP381 and can produce new variant with features of both around price of 1600-1700k as customers of both products have common features

5. Target customer for KP281 and KP481 are customers with salary around 50k, with average usage 3 to 4 days per week and are expected to run around 90 to 100 miles per week

In [ ]:

```
1
```