INDEX

- Checking the structure & characteristics of the dataset
- Checking for the distribution of values and outliers in various columns
- Handling the Outliers by clipping the values of various columns with values between 5 to 95 percentile
- Analysing the sales of each product
- Finding relation of Age with type of product preferred
- Finding how marital status effects the purchase
- Finding relation between the fitness levels and products preferred
- Finding the how gender affects the purchase of various products
- Finding how Age and Income is affecting the type of product customer purchase
- Finding how Age and Gender is affecting the fitness goals of the customer
- Finding the Marginal probability of each product
- Finding conditional probability
- Creating groups(bins) for further analysis
- Analysing the Age group and purchase pattern
- Analysing the Income group and purchase pattern
- Product Mapping to various income bracket
- Analysing the correlation among different factors
- Customer profiling for KP281
- Customer profiling for KP481
- Customer profiling for KP781

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

```
df=pd.read_csv("aerofit_treadmill.csv")
```

```
df
```

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

180 rows × 10 columns

Checking the structure & characteristics of the dataset

```
#Checking numbr of rows and columns
df.shape
```

```
(180, 9)
```

- our table has 180 rows and 9 columns

```
# fetching the details about the various columns of the dataset
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

```
# Fetching the datatypes of all the columns
df.dtypes
```

	0
Product	object
Age	int64
Gender	object
Education	int64
MaritalStatus	object
Usage	int64
Fitness	int64
Income	int64
Miles	int64

dtype: object

- Our table has columns of data type **object** and **integer** only

```
# Checking the missing values in the dataset
g=pd.isna(df["Income"])
df.loc[g==True].count()
```

	0
Product	0
Age	0
Gender	0
Education	0
MaritalStatus	0
Usage	0
Fitness	0
Income	0
Miles	0

dtype: int64

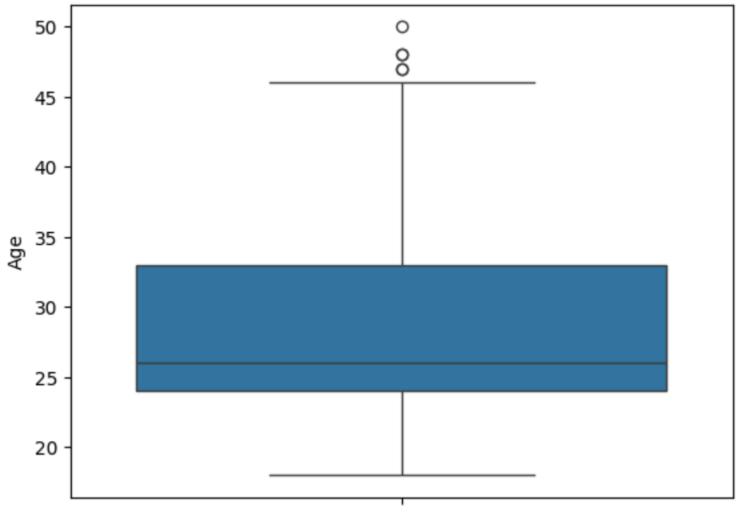
- We do not have missing values in the dataset

Checking for the distribution of values and outliers in various coulumns

Finding the Age bracket of the majority of the buyers

```
sn.boxplot(y=df["Age"]  )
plt.show()
```

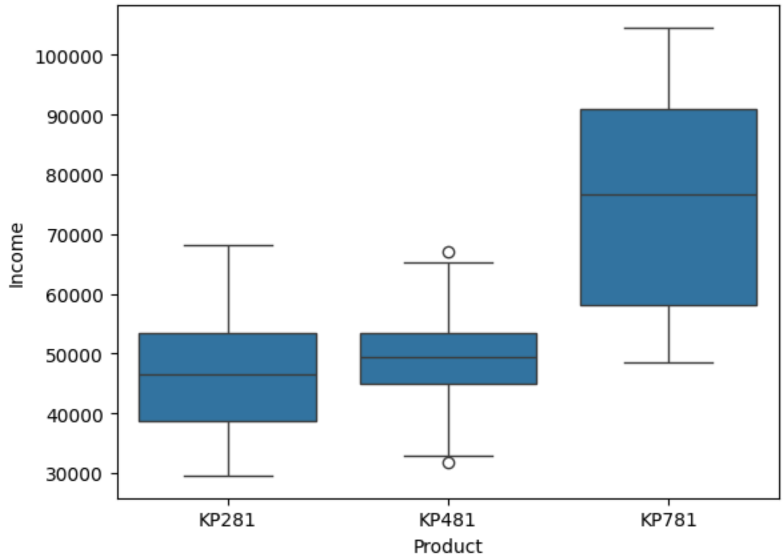
```
plt.show()
```



- **Majority of the buyers** of around the age **25 - 35** years of age.
- So if the customer in this age bracket enquires about the products the he/she will most likely to buy

Finding the product bought by the customers of various income groups

```
sn.boxplot(y=df["Income"] , x=df["Product"])
plt.show()
```

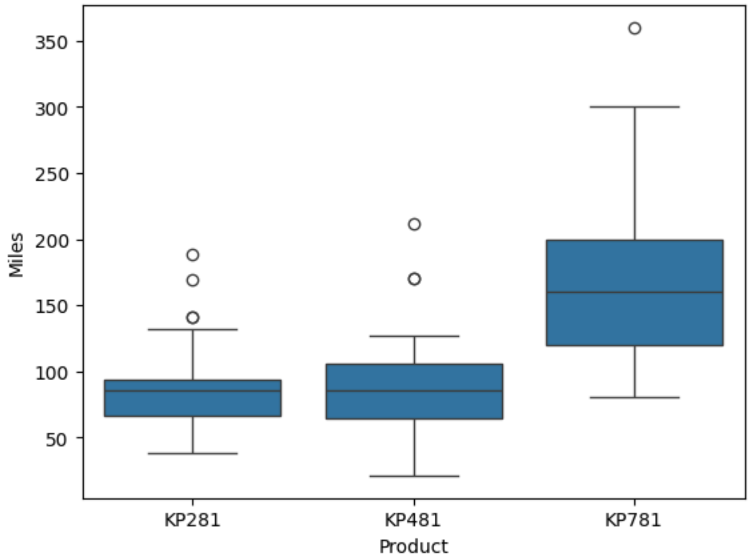


- Product KP781 is preferred by the higher end customers with median salary of around 80000
- Product KP281 and KP481 is most preferred by the customers with salary of around 40000 to 50000

Finding product preferred by the beginners and pros

```
sn.boxplot(y=df["Miles"] , x=df["Product"])
```

```
<Axes: xlabel='Product', ylabel='Miles'>
```



- Product KP781 is preferred by the customer who expects to walk/run more miles each week i.e those who are fitness enthusiast
- Whereas the other two products are preferred by the beginners who are kind of starting

Handling the Outliers by clipping the values of various columns with values between 5 to 95 percentile

```
df
```

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

180 rows × 10 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
incm = np.clip(df["Income"] , a_min=df['Income'].quantile(0.05) , a_max=df['Income'].quantile(0.95)).reset_index()
clp_df = pd.merge(df, incm , on="index" , how="left")

age_df = np.clip(df["Age"] , a_min=df['Age'].quantile(0.05) , a_max=df['Age'].quantile(0.95)).reset_index()
clp_age_df = pd.merge(clp_df, age_df , on="index" , how="left")

miles_df = np.clip(df["Miles"] , a_min=df['Miles'].quantile(0.05) , a_max=df['Miles'].quantile(0.95)).reset_index()
clp_miles_df = pd.merge(clp_age_df, miles_df , on="index" , how="left")

clp_miles_df.rename(columns={'Income_y' : "Clipped_Income" , "Age_y": "Clipped_Age" , "Miles_y" : "Clipped_Miles"}  , inplace =True)
clp_miles_df
```

	index	Product	Age_x	Gender	Education	MaritalStatus	Usage	Fitness	Income_x	Miles_x	Clipped_Income	Clipped_Age	Clippec
0	0	KP281	18	Male	14	Single	3	4	29562	112	34053.15	20.00	
1	1	KP281	19	Male	15	Single	2	3	31836	75	34053.15	20.00	
2	2	KP281	19	Female	14	Partnered	4	3	30699	66	34053.15	20.00	
3	3	KP281	19	Male	12	Single	3	3	32973	85	34053.15	20.00	
4	4	KP281	20	Male	13	Partnered	4	2	35247	47	35247.00	20.00	
...	...	...	...	...	...	...	...	...	...	...	...	...	
175	175	KP781	40	Male	21	Single	6	5	83416	200	83416.00	40.00	
176	176	KP781	42	Male	18	Single	5	4	89641	200	89641.00	42.00	
177	177	KP781	45	Male	16	Single	5	5	90886	160	90886.00	43.05	
178	178	KP781	47	Male	18	Partnered	4	5	104581	120	90948.25	43.05	
179	179	KP781	48	Male	18	Partnered	4	5	95508	180	90948.25	43.05	

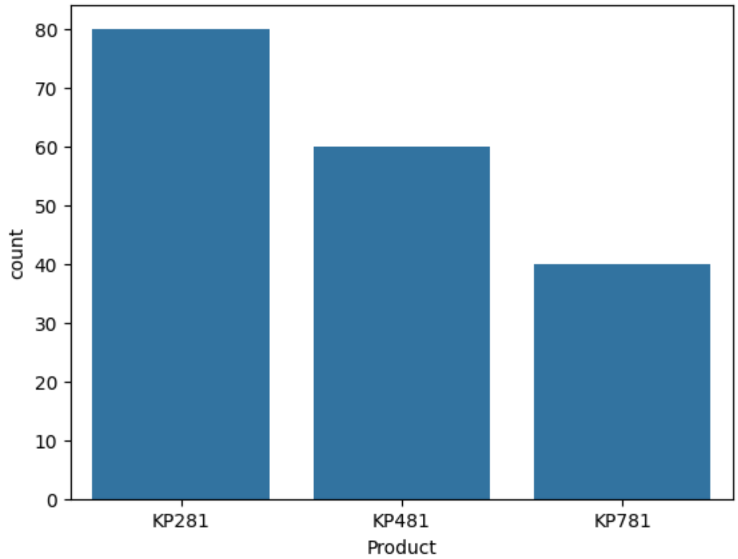
180 rows × 13 columns

Next steps: Generate code with `clp_miles_df` View recommended plots New interactive sheet

Analysing the sales of each product

```
sn.countplot(x=df["Product"] )
```

```
<Axes: xlabel='Product', ylabel='count'>
```

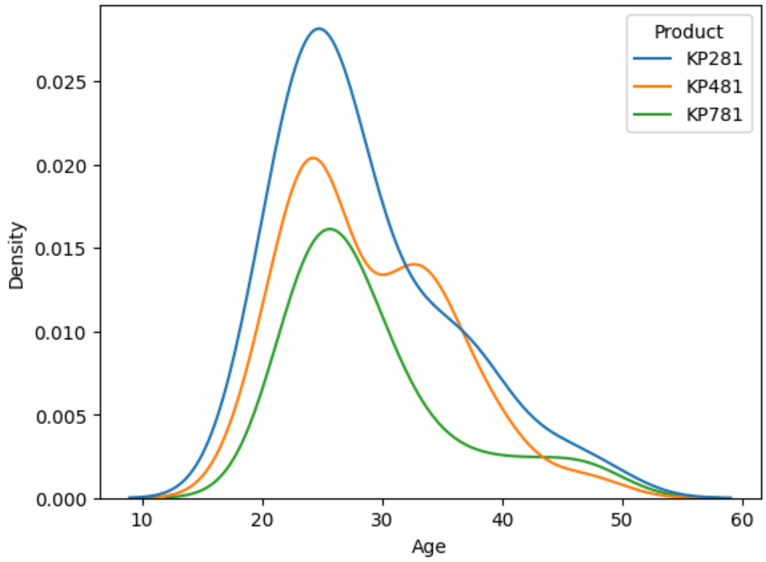


- Here with can see that the **KP281** is sold the most and **KP781 sold the least**

Finding relation of Age with type of product preferred

```
sn.kdeplot( x=df["Age"] , hue=df["Product"])
```

```
<Axes: xlabel='Age', ylabel='Density'>
```



- We can infer from the graph that
  - All the products majorly bought by people from 20-30 years of age
  - KP481 is famous also among the people between 30-40 years old customers

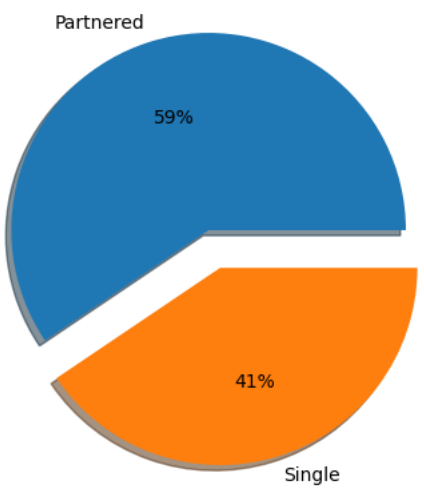
Finding how marital status effects the purchase

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

MaritalStatus	count
Partnered	107
Single	73

dtype: int64

```
a = df["MaritalStatus"].value_counts()
plt.pie(a , labels =["Partnered", "Single"] , explode=[0.2, 0] ,shadow = True , autopct='%1.0f%%')
plt.show()
```



- **Partnered** buyers are significantly **more** than single buyers

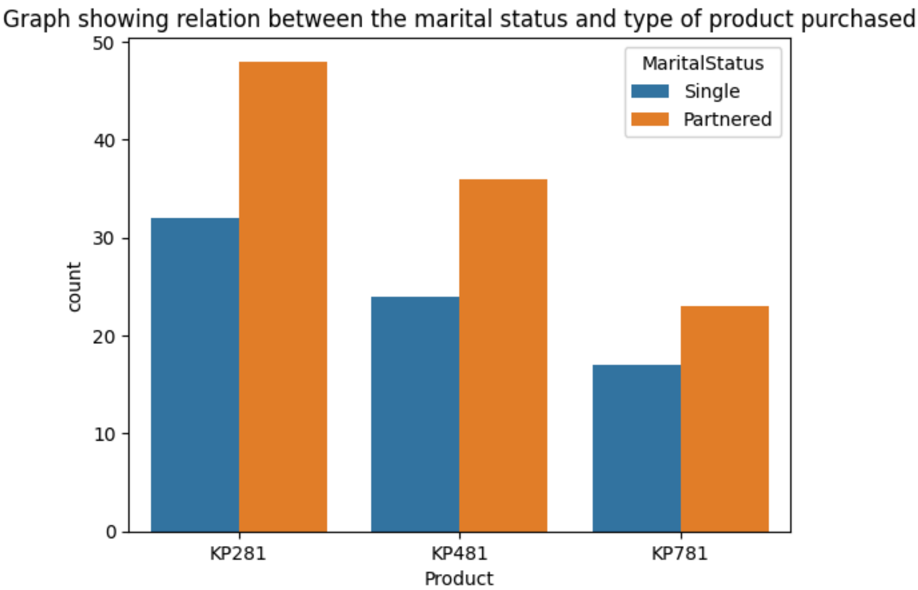
Finding this trend across the products

```
df.groupby(["Product" , "MaritalStatus"])["index"].count()
```

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

dtype: int64

```
sn.countplot(x=df["Product"] , hue =df["MaritalStatus"] )
plt.title("Graph showing relation between the marital status and type of product purchased")
plt.show()
```

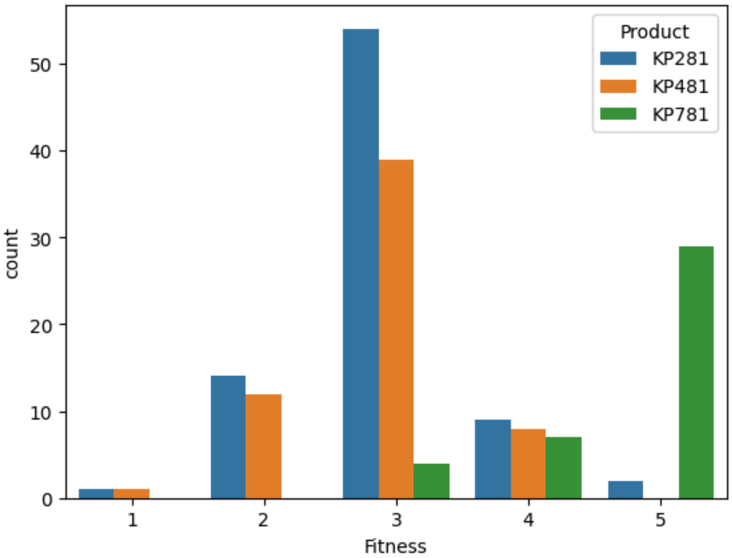


- Across the products **partnered(or married)** buyers are **more** than single buyers

Finding relation between the fitness levels and products preferred

```
sn.countplot(x=df["Fitness"] , hue =df["Product"] )
plt.show("Type of product purchased on the basis of fitness levels")
```

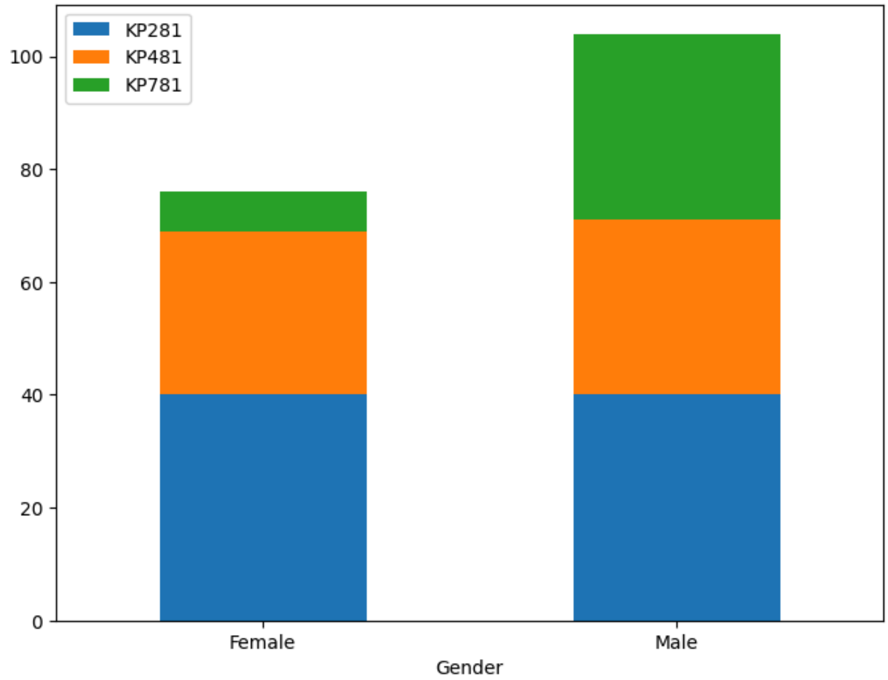
```
<Axes: xlabel='Fitness', ylabel='count'>
```



- People in the **advance level** of fitness prefer to buy **KP781**
- Whereas the people with **average fitness** prefer **KP281 and KP481**
- People with fitness level **above average** i.e. 4 , are interested in buying **any of the three** products

Finding the how gender affects the purchase of various products

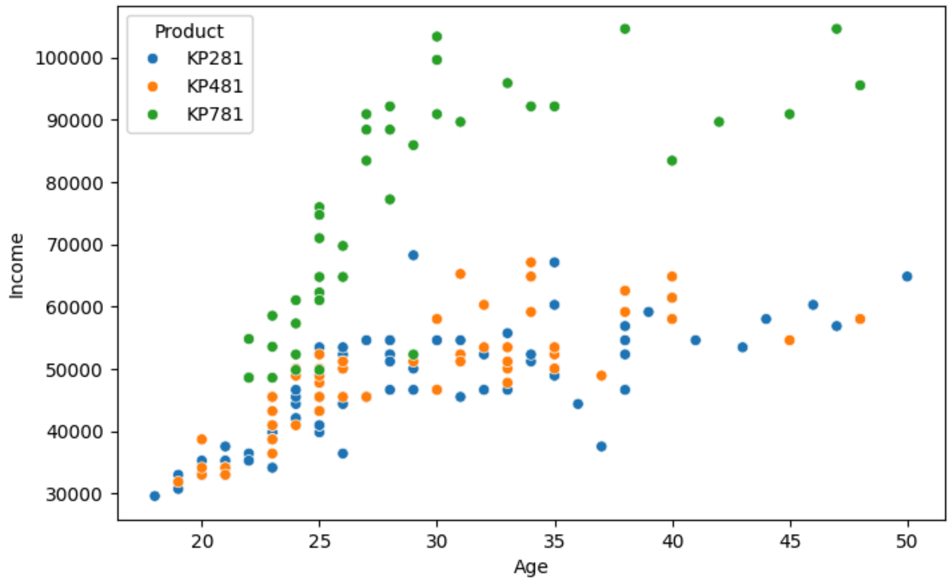
```
df_stacked_plot = pd.crosstab(index=df["Gender"], columns=df["Product"])

df_stacked_plot.plot(kind='bar', stacked=True, figsize=(8, 6))
plt.xticks(rotation=0)
plt.legend(loc='upper left')
plt.show()
```



- **More male buyers** in the overall purchase
- **KP281 and KP481** is equally famous among **both genders**
- However **KP781** is more preferred by the **male buyers**

Finding how Age and Income is affecting the type of product customer purchase

```
plt.figure(figsize=(8,5))
sn.scatterplot(x=df["Age"] , y=df["Income"] , hue=df["Product"])
plt.show()
# different colours show variety of products
# x and y axis represents age and income respectively
```



Inference drawn

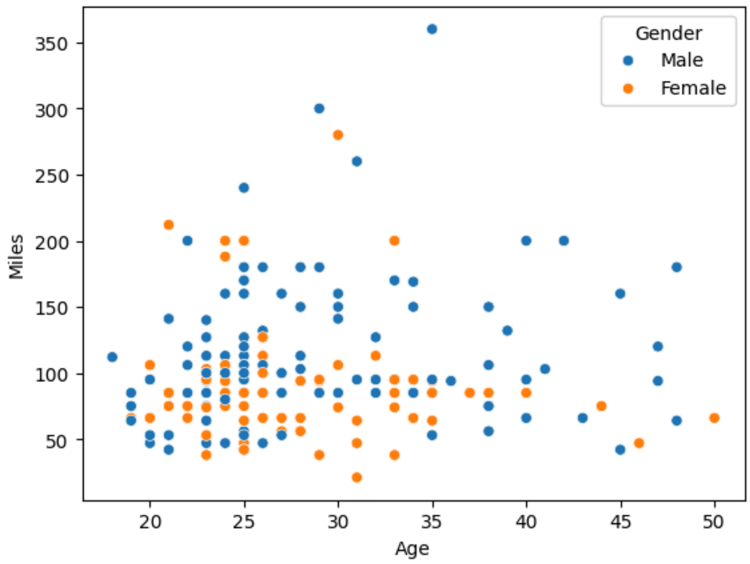
- Most of the buyers of product **KP281 and KP481** aged between **20 to 40** years earning **65000 or less**
- The purchase of the product **KP781** depends on the **income** of the buyer and **not the age** of the buyer

Double-click (or enter) to edit

Finding how Age and Gender is affecting the fitness goals of the customer

```
sn.scatterplot(x=df["Age"] , y=df["Miles"] , hue=df["Gender"] )
```

```
<Axes: xlabel='Age', ylabel='Miles'>
```



- Both **young male and femal**e customer are **passionate** about fitness
- In the **older males** (40 years and above) are **more intrested** in running **than female counterpart**
- Male customers run greater number of miles than female counterpart

Finding the Marginal probability of each product

```
kk = pd.crosstab(index =df["Product"] , columns =df["index"].count() , margins=True)
kk
```

col_0	180	All
Product		
KP281	80	80
KP481	60	60
KP781	40	40
All	180	180

Next steps: Generate code with kk | View recommended plots | New interactive sheet

```
# prob that a customer will buy KP281
a= (kk.loc["KP281","All"] / kk.loc["All","All"])
print(round(a,2))
```

```
0.44
```

```
# prob that a customer will buy KP481
b= (kk.loc["KP481","All"] / kk.loc["All","All"])
print(round(b,2))
```

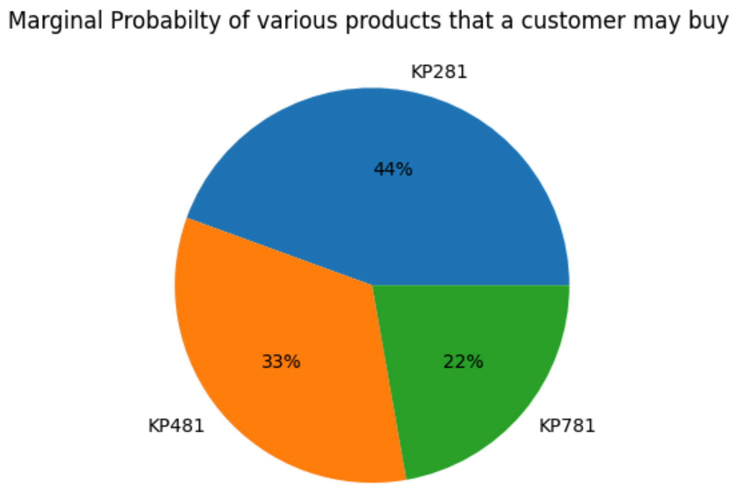
```
0.33
```

```
# prob that a customer will buy KP781
round(40/180, 2)
c= (kk.loc["KP781","All"] / kk.loc["All","All"])
print(round(c,2))
```

```
0.22
```

Graphical representaion

```
d=pd.Series({"a":a  , "b": b , "c":c})
plt.pie(d ,labels =["KP281","KP481","KP781"]  ,shadow = False  , autopct='%1.0f%%' )
plt.title("Marginal Probabilty of various products that a customer may buy " )
plt.show()
```



- Customer will **most likely** to buy **KP281** followed by KP481 and KP781
- An outlet must have sufficient stock of **KP281**

Finding conditional probability

Conditional probability that Female customer will buy a particular product

```
ct = pd.crosstab(index =df["Gender"] , columns =df["Product"] , margins=True)
ct
```

Product	KP281	KP481	KP781	All
Gender				
Female	40	29	7	76
Male	40	31	33	104
All	80	60	40	180

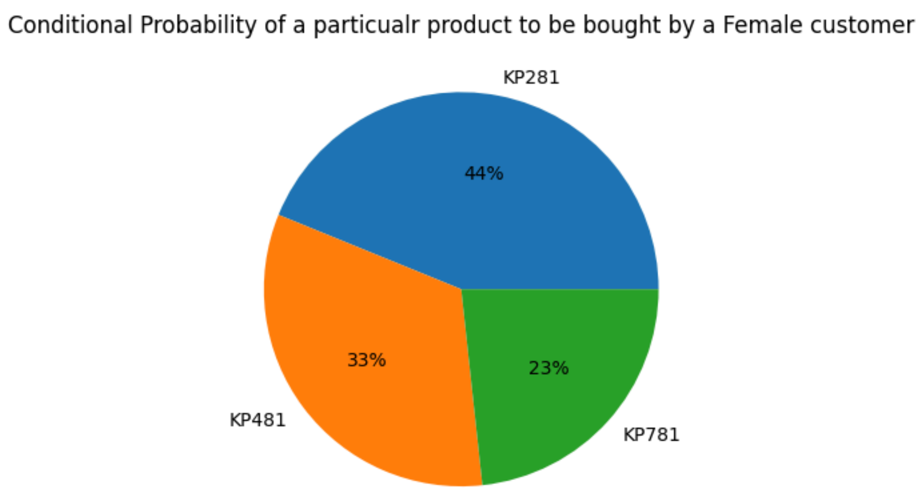
Next steps: Generate code with ct | View recommended plots | New interactive sheet

```
# 1 . Probability that Female customer per will buy KP281
a= ct.loc["Female","KP281"] / ct.loc["Female","All"]
print("1. Probability that Female customer per will buy KP281 = "  ,  round(a,2))
print("")

# 2 . Probability that Female per will buy KP481
b= ct.loc["Female","KP481"] / ct.loc["Female","All"]
print("2. Probability that Female customer per will buy KP481 = "  , round(b,2))
print("")

# 3 . Probability that Female per will buy KP781
c= ct.loc["Female","KP781"] / ct.loc["Female","All"]
print("3. Probability that Female customer per will buy KP781 = "  ,  round(c,2) )

d=pd.Series({"a":a  , "b":b , "c":c})
```

```
1. Probability that Female customer per will buy KP281 =  0.53

2. Probability that Female customer per will buy KP481 =  0.38

3. Probability that Female customer per will buy KP781 =  0.09
```

Graphical representaion

```
plt.pie(d ,labels =["KP281","KP481","KP781"]  ,shadow = False  , autopct='%1.0f%%' )
plt.title("Conditional Probability of a particualr product to be bought by a Female customer")
plt.show()
```



- **Female** customers **most likely** to **buy KP281** hence it should be pitched first to the female customer
- **Female** customers **least likely** to buy **KP781** hence less time should be invested in pitching this product to them

Conditional probability that Male customer will buy a particular product

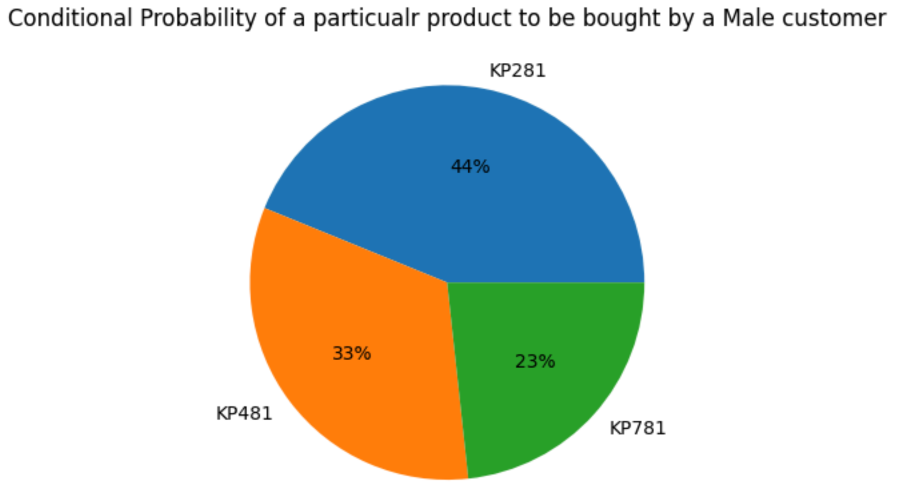
```
# 1 . Probability that Male customer per will buy KP281
a= ct.loc["Male","KP281"] / ct.loc["Male","All"]
print("1. Probability that Male customer customer per will buy KP281 = "  ,  round( a ,2))

print("")

# 2 . Probability that Male customer per will buy KP481
b= ct.loc["Male","KP481"] / ct.loc["Male","All"]
print("1. Probability that Male customer customer per will buy KP481 = "  ,  round( b ,2))
```

```
print("")

# 3 . Probability that Male customer per will buy KP781
c= ct.loc["Male","KP781"] / ct.loc["Male","All"]
print("1. Probability that Male customer customer per will buy KP781 = "  ,  round( c ,2))


d=pd.Series({"a":a  , "b": b , "c":c})
```

```
1. Probability that Male customer customer per will buy KP281 =  0.38

1. Probability that Male customer customer per will buy KP481 =  0.3

1. Probability that Male customer customer per will buy KP781 =  0.32
```

Graphical representation

```
plt.pie(d ,labels =["KP281","KP481","KP781"]  ,shadow = False  , autopct='%1.0f%%' )
plt.title("Conditional Probability of a particualr product to be bought by a Male customer")
plt.show()
```



- **All the three** products appeal to a **male** customer so all the products should be pitched to them
- **KP281** has slightly more chance to get purchased by male with probability of **38%**

Conditional probability that Partnered customer will buy a particular product

```
ct1 = pd.crosstab(index =df["MaritalStatus"] , columns =df["Product"] , margins=True)
ct1
```

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	48	36	23	107
Single	32	24	17	73
All	80	60	40	180

Next steps: Generate code with ct1 View recommended plots New interactive sheet

```
# 1 . Probability that Partnered customer per will buy KP281
a=ct1.loc["Partnered","KP281"] / ct1.loc["Partnered","All"]
print("1. Probability that Partnered customer per will buy KP281 = "  ,  round( a,2))
print("")

# 2 . Probability that Partnered customer per will buy KP481
b=ct1.loc["Partnered","KP481"] / ct1.loc["Partnered","All"]
print("2. Probability that Partnered customer per will buy KP481 = "  , round( b,2)  )
print("")

# 3 . Probability that Partnered customer per will buy KP781
c=ct1.loc["Partnered","KP781"] / ct1.loc["Partnered","All"]
print("3. Probability that Partnered customer per will buy KP781 = "  ,  round( c,2) )
```

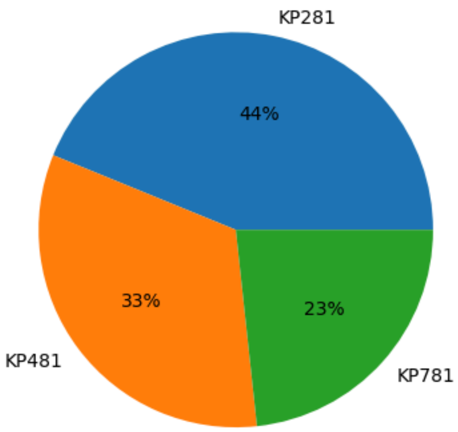
```
d=pd.Series({"a":a  , "b": b , "c":c})
```

```
1. Probability that Partnered customer per will buy KP281 =  0.45

2. Probability that Partnered customer per will buy KP481 =  0.34

3. Probability that Partnered customer per will buy KP781 =  0.21
```

Graphical representation

```
plt.pie(d ,labels =["KP281","KP481","KP781"]  ,shadow = False  , autopct='%1.0f%%' )
plt.title("Conditional Probability of a particualr product to be bought by a Partnered customer")
plt.show()
```

Conditional Probability of a particualr product to be bought by a Partnered customer



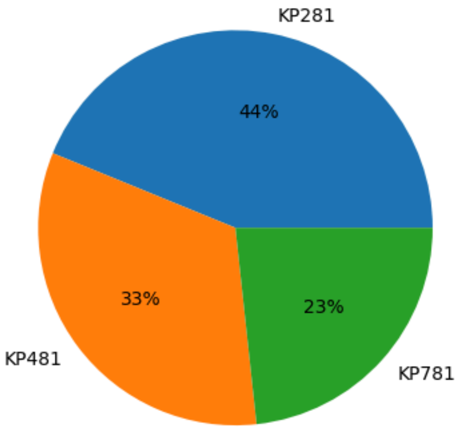

```
# 1 . Probability that Single customer per will buy KP281
a=ct1.loc["Single","KP281"] / ct1.loc["Single","All"]
print("1. Probability that Single customer per will buy KP281 = "  ,  round( a ,2))
print("")

# 2 . Probability that Single per will buy KP481
b=ct1.loc["Single","KP481"] / ct1.loc["Single","All"]

print("2. Probability that Single customer per will buy KP481 = "  , round( b,2)  )
print("")

# 3 . Probability that Single per will buy KP781
c=ct1.loc["Single","KP781"] / ct1.loc["Single","All"]
print("3. Probability that Single customer per will buy KP781 = "  ,  round( c,2) )


d=pd.Series({"a":a  , "b":b , "c":c})
```

```
1. Probability that Single customer per will buy KP281 =  0.44

2. Probability that Single customer per will buy KP481 =  0.33

3. Probability that Single customer per will buy KP781 =  0.23
```

Graphical representation

```
plt.pie(d ,labels =["KP281","KP481","KP781"]  ,shadow = False  , autopct='%1.0f%%' )
plt.title("Conditional Probability of a particualr product to be bought by a Single customer")
plt.show()
```

Conditional Probability of a particualr product to be bought by a Single customer



Creating groups(bins) for further analysis

- Segregating age into age groups
- Segregating income into income groups

```
data = df["Age"]
bins = pd.cut(data, bins=[10, 20, 30, 40 ,50], labels=['Young(10-20)', 'Adult(20-30)', 'Mature(30-40)' , 'Old(40-70)'])
df_new=df.iloc[::]
df_new["Age_bracket"] = bins

data_income = df["Income"]
bins_income = pd.cut(data_income, bins=[0,40000,80000,120000], labels=['Low', 'Medium', 'High'])
df_new["Income_group"] = bins_income
df_new
```

	index	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_bracket	Income_group
0	0	KP281	18	Male	14	Single	3	4	29562	112	Young(10-20)	Low
1	1	KP281	19	Male	15	Single	2	3	31836	75	Young(10-20)	Low
2	2	KP281	19	Female	14	Partnered	4	3	30699	66	Young(10-20)	Low
3	3	KP281	19	Male	12	Single	3	3	32973	85	Young(10-20)	Low
4	4	KP281	20	Male	13	Partnered	4	2	35247	47	Young(10-20)	Low
...	...	...	...	...	...	...	...	...	...	...	...	...
175	175	KP781	40	Male	21	Single	6	5	83416	200	Mature(30-40)	High
176	176	KP781	42	Male	18	Single	5	4	89641	200	Old(40-70)	High
177	177	KP781	45	Male	16	Single	5	5	90886	160	Old(40-70)	High
178	178	KP781	47	Male	18	Partnered	4	5	104581	120	Old(40-70)	High
179	179	KP781	48	Male	18	Partnered	4	5	95508	180	Old(40-70)	High

180 rows × 12 columns

Next steps: Generate code with df_new | View recommended plots | New interactive sheet

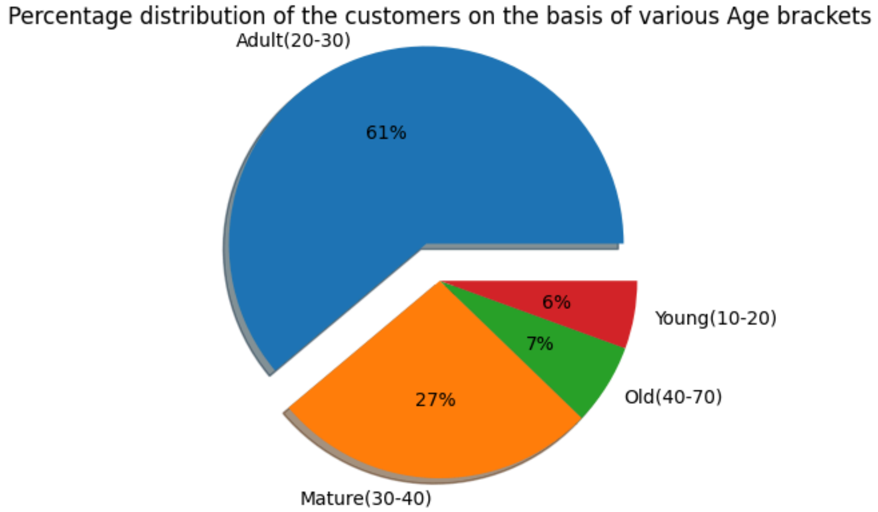
Analysing the Age group and purchase pattern

```
age_prob=df_new["Age_bracket"].value_counts()
age_prob
```

Age_bracket	count
Adult(20-30)	110
Mature(30-40)	48
Old(40-70)	12
Young(10-20)	10

dtype: int64

```
# plt.pie(age_prob ,labels =["Adult","Old","Young"]  ,shadow = True  , autopct='%1.0f%%' )

plt.pie(age_prob ,labels =["Adult(20-30)","Mature(30-40)","Old(40-70)","Young(10-20)"]  ,shadow = True  , autopct='%1.0f%%'  ,explode=((
plt.title("Percentage distribution of the customers on the basis of various Age brackets")
plt.show()
```



- Whopping **61%** of the customers are in the age bracket **of 20 to 30** followed by mature age bracket of 30 -40
- **Young** and **old** people are on the **lesser** side
- The above age groups can be taken into consideration while deciding the most **favourable campaign sites**(like colleges , corporate offices etc.)

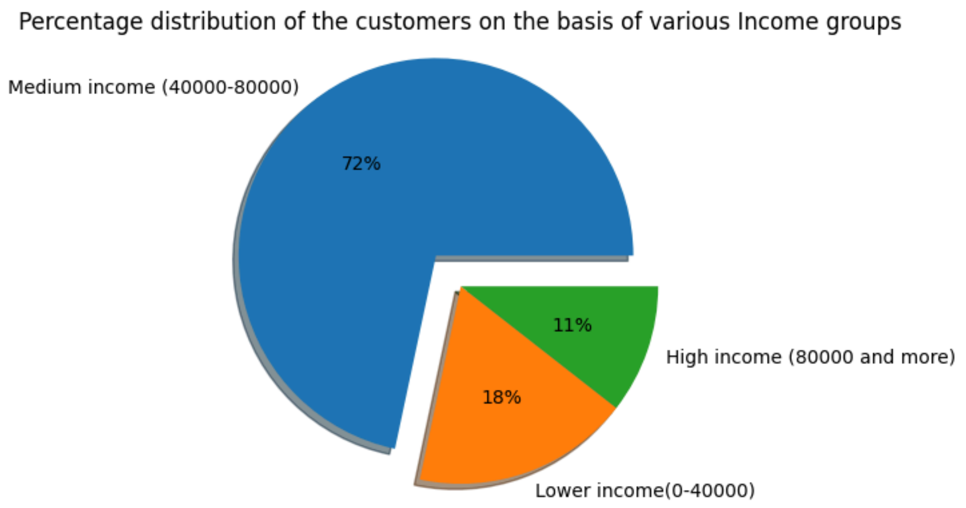
Analysing the Income group and purchase pattern

```
income_prob=df_new["Income_group"].value_counts()
income_prob
```

Income_group	count
Medium	129
Low	32
High	19

dtype: int64

```
plt.pie(income_prob ,labels =['Medium income (40000-80000)', 'Lower income(0-40000)', 'High income (80000 and more)']  ,shadow = True  ,
plt.title("Percentage distribution of the customers on the basis of various Income groups")
plt.show()
```

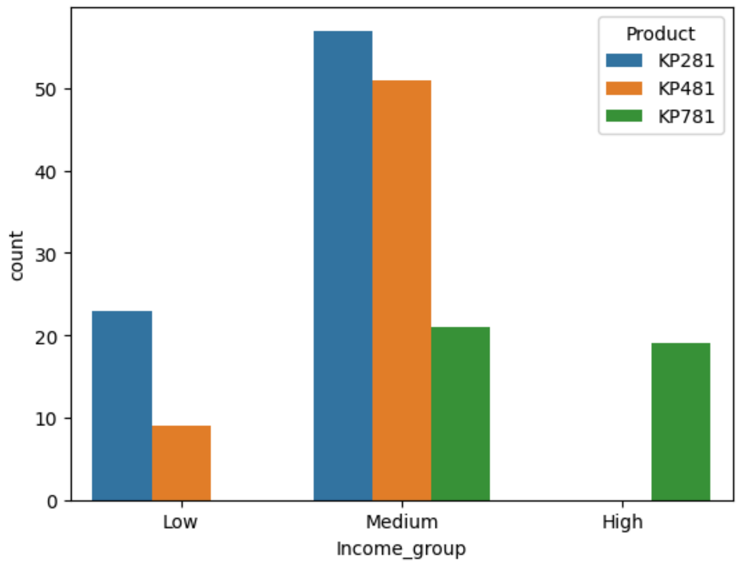


- **72% of the buyers** are in the medium income group earning between **40000 to 80000**
- **Higher income group** buyers are the **least**
- This analysis could be used for determining the **socio-economic criteria** for targeting and campaigning for the future customers

Product Mapping to various income bracket

```
sn.countplot(x=df_new["Income_group"] , hue =df["Product"] )
plt.show()
```

```
<Axes: xlabel='Income_group', ylabel='count'>
```



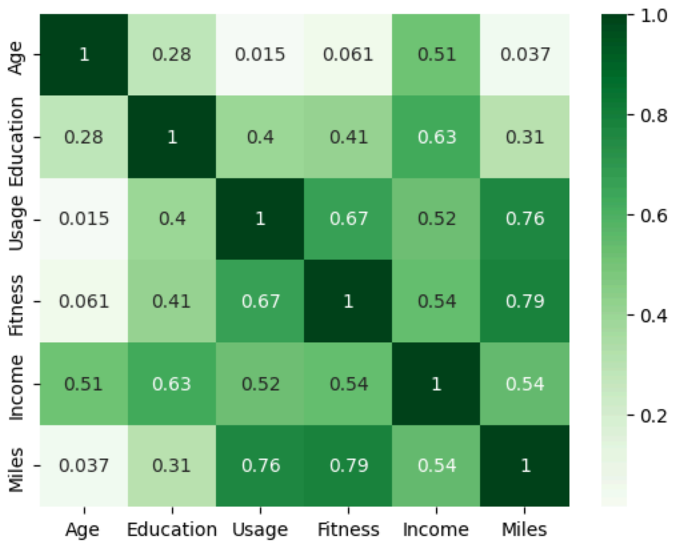
- **Lower income** group people are not interested in buying **KP781**
- **Medium income** people more interested in buying **KP281** and **KP481**
- **Medium income** group people also show interest in **KP781** however it is lesser compared to other two products
- **Higher income** group people only interested in buying **KP781**

Analysing the correlation among different factors

```
num_df = df_new.select_dtypes(include=[float,int])
num_df.drop(columns="index" , inplace=True)
num_df.corr()
```

	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

```
sn.heatmap(num_df.corr(), cmap= "Greens", annot=True)
plt.show()
```



Inferences drawn from the above heatmap

- **Usage-Miles** and **Fitness-Miles** are the most correlated features
- **Miles, Fitness and Usage** are least correlated with the **Age**

Customer profiling for KP281

```
# customer profilings for product - KP281
df_prod1 = df_new.loc[df["Product"]=="KP281"]
df_prod1.head()
```

	index	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_bracket	Income_group
0	0	KP281	18	Male	14	Single	3	4	29562	112	Young(10-20)	Low
1	1	KP281	19	Male	15	Single	2	3	31836	75	Young(10-20)	Low
2	2	KP281	19	Female	14	Partnered	4	3	30699	66	Young(10-20)	Low
3	3	KP281	19	Male	12	Single	3	3	32973	85	Young(10-20)	Low
4	4	KP281	20	Male	13	Partnered	4	2	35247	47	Young(10-20)	Low

Next steps: Generate code with df_prod1 View recommended plots New interactive sheet

```
df_prod1.groupby(df_new["Age_bracket"] , observed=False)["index"].count()
```

Age_bracket	index
Young(10-20)	6
Adult(20-30)	49
Mature(30-40)	19
Old(40-70)	6

dtype: int64

```
df_prod1.groupby(df_new["Income_group"] , observed=False)["index"].count()
```

Income_group	index
Low	23
Medium	57
High	0

dtype: int64

```
df_prod1.groupby(df_new["Gender"] , observed=False)["index"].count()
```

Gender	index
Female	40
Male	40

dtype: int64

Customer profile for the product KP281

- Adult(20-30 years of age) are the most number of buyers of this product
- Young and old people are least interested in buying this product
- Medium income group people (annual income 40000-80000 $) are the major buyers
- High income group people (annual income 80000 $ or more) are not interested in buying this product
- Both male and female show similar interest in buying this product

Customer profiling for KP481

```
# customer profilings for product - KP481
df_prod2 = df_new.loc[df["Product"]=="KP481"]
df_prod2.head()
```

	index	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_bracket	Income_group
80	80	KP481	19	Male	14	Single	3	3	31836	64	Young(10-20)	Low
81	81	KP481	20	Male	14	Single	2	3	32973	53	Young(10-20)	Low
82	82	KP481	20	Female	14	Partnered	3	3	34110	106	Young(10-20)	Low
83	83	KP481	20	Male	14	Single	3	3	38658	95	Young(10-20)	Low
84	84	KP481	21	Female	14	Partnered	5	4	34110	212	Adult(20-30)	Low

```
df_prod2.groupby(df_new["Age_bracket"] , observed=False)["index"].count()
```

Age_bracket	index
Young(10-20)	4
Adult(20-30)	31
Mature(30-40)	23
Old(40-70)	2

dtype: int64

```
df_prod2.groupby(df_new["Income_group"] , observed=False)["index"].count()
```

Income_group	index
Low	9
Medium	51
High	0

dtype: int64

```
df_prod2.groupby(df_new["Gender"] , observed=False)["index"].count()
```

Gender	index
Female	29
Male	31

dtype: int64

Customer profile for the product KP481

- Adult(20-30 years of age) and Mature(30-40 years of age) people are the most number of buyers of this product and shows similar interest in the product
- Young and old people are least interested in buying this product
- Medium income group people (annual income 40000-80000 $) are the major buyers
- High income group people (annual income 80000 $ or more) are not interested in buying this product
- Both male and female show similar interest in buying this product

Customer profiling for KP781

```
# customer profilings for product - KP781
df_prod3 = df_new.loc[df["Product"]=="KP781"]
df_prod3.head(5)
```

	index	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_bracket	Income_group
140	140	KP781	22	Male	14	Single	4	3	48658	106	Adult(20-30)	Medium
141	141	KP781	22	Male	16	Single	3	5	54781	120	Adult(20-30)	Medium
142	142	KP781	22	Male	18	Single	4	5	48556	200	Adult(20-30)	Medium
143	143	KP781	23	Male	16	Single	4	5	58516	140	Adult(20-30)	Medium
144	144	KP781	23	Female	18	Single	5	4	53536	100	Adult(20-30)	Medium

```
df_prod3.groupby(df_new["Age_bracket"] , observed=False)["index"].count()
```

Age_bracket	index
Young(10-20)	0
Adult(20-30)	30
Mature(30-40)	6
Old(40-70)	4

dtype: int64

```
df_prod3.groupby(df_new["Income_group"] , observed=False)["index"].count()
```

Income group	index
Medium	21
High	19

```
df_prod3.groupby(df_new["Gender"] , observed=False)["index"].count()
```

Gender	index
Female	7
Male	33

dtype: int64

Customer profile for the product KP781

- Adult(20-30 years of age) people are the most interested in this product
- Young(10-20 years of age) people are not interested in buying this product
- Low income group people (annual income 0-40000 $) are not interested in this product
- Medium income group (annual income 40000-80000 $) and higher income group people are the only buyers and show similar interest
- Male customers are more inclined for this product than female customers

Detailed Business recommendations

- Majority of the buyers of around the age 25 - 35 years of age. So if the customer in this age bracket enquires about the products the he/she will most likely to buy
- Partnered buyers are significantly more than single buyers so if a partnered person enquires about the product its is more likely that he/she will buy the product
- People in the advance level of fitness prefer to buy KP781. Whereas the people with average fitness prefer KP281 and KP481. People with fitness level above average i.e. 4 , are interested in buying any of the three products
- More male buyers in the overall purchase . KP281 and KP481 is equally famous among both genders . However KP781 is more preferred by the male buyers
- Most of the buyers of product KP281 and KP481 aged between 20 to 40 years and earning 65000 or less . The purchase of the product KP781 depends on the income of the buyer and not the age of the buyer
- Customer will most likely to buy KP281 followed by KP481 and KP781 . KP281 is sold the most so an outlet must have sufficient stock of KP281
- Female customers most likely to buy KP281 hence it should be pitched first to the female customer . Female customers least likely to buy KP781 hence less time should be invested in pitching this product to them.
- All the three products appeal to a male customer so all the products should be pitched to them. KP281 has slightly more chance to get purchased by male with probability of 38%
- Whopping 61% of the customers are in the age bracket of 20 to 30, followed by mature age bracket of 30 -40 . Young and old people are on the lesser side. Mentioned age groups can be taken into consideration while deciding the most favourable campaign sites(like colleges , corporate offices etc.).
- 72% of the buyers are in the medium income group earning between 40000 to 80000 $ annually. Higher income group buyers are the least . This analysis could be used for determining the socio-economic criteria for targeting and campaigning for the future customers
- Lower income group people are not interested in buying KP781.Medium income people more interested in buying KP281 and KP481.Medium income group people also show interest in KP781 however it is lesser compared to other two products. Higher income group people only interested in buying KP781