In [1]:

```
import pandas as pd
```

In [2]:

```
df=pd.read_csv('aerofit.csv')
```

In [3]:

```
df
```

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
...	...	...	...	...	...	...	...	...	...
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 [4]:

```
#basic exploration

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 [6]:

```
df.isnull().sum()
```

Out[6]:

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

In [7]:

```
df.describe()
```

Out[7]:

	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 [ ]:

```
#
```

In [10]:

```
df['Product'].value_counts(normalize=True)*100
```

Out[10]:

```
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: Product, dtype: float64
```

In [12]:

```
df.groupby('Product')["Income"].describe()
```

Out[12]:

	count	mean	std	min	25%	50%	75%	max
Product								
KP281	80.0	46418.025	9075.783190	29562.0	38658.00	46617.0	53439.0	68220.0
KP481	60.0	48973.650	8653.989388	31836.0	44911.50	49459.5	53439.0	67083.0
KP781	40.0	75441.575	18505.836720	48556.0	58204.75	76568.5	90886.0	104581.0

In [ ]:

```
#
```

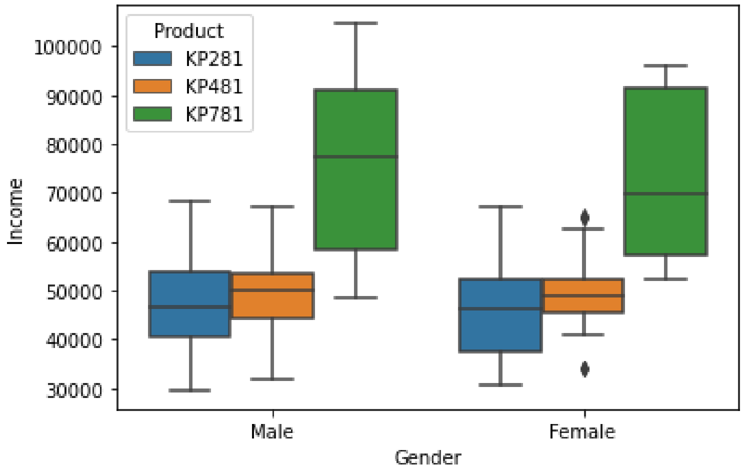
In [14]:

```
import seaborn as suraaj

suraaj.boxplot(x='Gender', y='Income', hue='Product', data=df)
```

Out[14]:

```
<AxesSubplot:xlabel='Gender', ylabel='Income'>
```



In [15]:

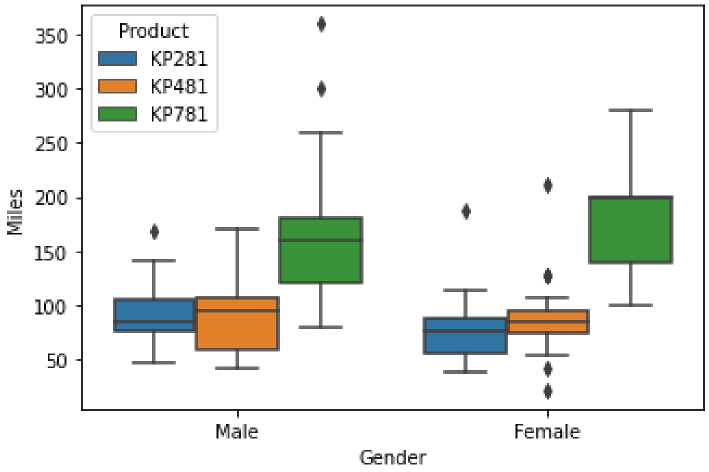
```
#
```

In [16]:

```
suraaj.boxplot(x='Gender', y='Miles', hue='Product', data=df)
```

Out[16]:

```
<AxesSubplot:xlabel='Gender', ylabel='Miles'>
```



In [18]:

```
df.groupby(['Gender','Product'])["Miles"].describe()
```

Out[18]:

		count	mean	std	min	25%	50%	75%	max
Gender	**Product**								
Female	**KP281**	40.0	76.200000	27.988276	38.0	56.0	75.0	87.25	188.0
	KP481	29.0	87.344828	33.456022	21.0	74.0	85.0	95.00	212.0
	KP781	7.0	180.000000	63.245553	100.0	140.0	200.0	200.00	280.0
Male	**KP281**	40.0	89.375000	28.573511	47.0	75.0	85.0	105.25	169.0
	KP481	31.0	88.483871	33.625259	42.0	58.5	95.0	106.00	170.0
	KP781	33.0	164.121212	60.014455	80.0	120.0	160.0	180.00	360.0

In [ ]:

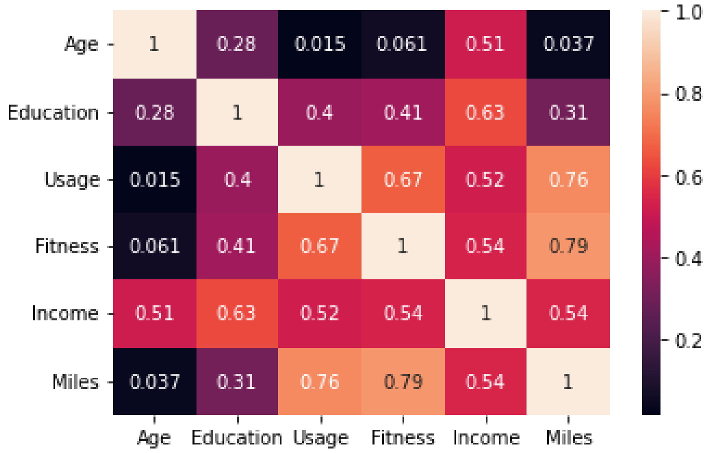
```
#
#Some more insights using different charts
```

In [19]:

```
suraaj.heatmap(df.corr(), annot=True)
```

Out[19]:

```
<AxesSubplot:>
```



In [ ]:

```
#
#
#
#
```

In [ ]:

```
#IQR is not mandatory
```

In [23]:

```
#gender and product

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

Out[23]:

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

In [24]:

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

Out[24]:

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

In [26]:

```
pd.crosstab(index=df['Gender'], columns=df['Product'], margins=True, normalize=True)*100
```

Out[26]:

Product Gender	KP281	KP481	KP781	All
Female	22.222222	16.111111	3.888889	42.222222
Male	22.222222	17.222222	18.333333	57.777778
All	44.444444	33.333333	22.222222	100.000000

In [ ]:

```
#p(female buying 781) - 3.8%
```

In [27]:

```
pd.crosstab(index=df['Gender'], columns=df['Product'], margins=True, normalize='index')*10
```

Out[27]:

Product Gender	KP281	KP481	KP781
Female	52.631579	38.157895	9.210526
Male	38.461538	29.807692	31.730769
All	44.444444	33.333333	22.222222

In [ ]:

```
#conditional

1. P(781|Male)= 31.7%
#
#
```

In [30]:

```
pd.crosstab(index=df['Gender'], columns=df['Product'], margins=True, normalize='columns')*
```

Out[30]:

Product Gender	KP281	KP481	KP781	All
Female	50.0	48.333333	17.5	42.222222
Male	50.0	51.666667	82.5	57.777778

In [ ]:

```
P (Male| 781) - 82.5%
```