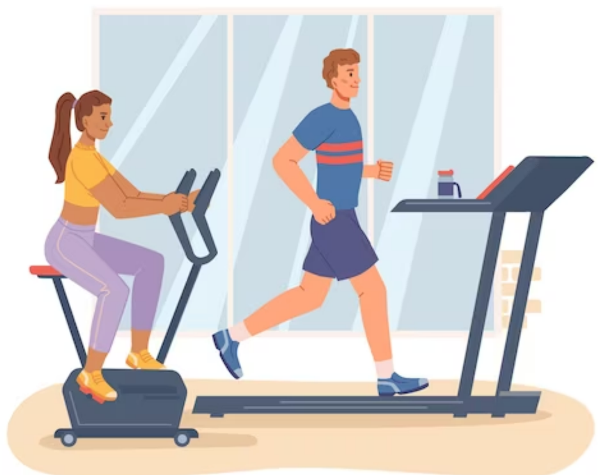
Aerofit Business Case Study



- Importing essential libraries

In [34]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
sns.set_theme(style="whitegrid")
warnings.filterwarnings('ignore')
```

- Importing data and checking the data for first 5 rows

In [35]:
```
df = pd.read_csv('C:/Users/hp/OneDrive/Desktop/Scaler/Aerofit case study/a
df.head()
```

Out[35]:

	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

Problem Statement and Analysing basic metrics

- To identify the characteristics of the target audience for each type of treadmill offered by the company.
- Target correct set of customers for their choice of treadmill
- To provide a better recommendation of the treadmills to the new customers

In [36]:
```
# The data given contains 180 rows and 9 columns
df.shape
```

Out[36]:
```
(180, 9)
```

In [37]:
```
# List of all columns in given data
df.columns
```

Out[37]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usag
e',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

In [38]:
```
# Statistical summary
df.dtypes
```

Out[38]:
```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

In [39]:
```
df.describe()
```

Out[39]:

	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 [40]:

```
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 [41]:

```
df[df.duplicated()]
```

Out[41]:

Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles

In [42]:

```
df.isna().sum()
```

Out[42]:

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

Observations

1. There are no missing values in dataset. So no need to handle missing values.
2. Except Product, Gender, Marital Status column all other columns are having integer data type.
3. There are no duplicates in given data.

Non-Graphical Analysis: Value counts and unique attributes

In [43]:
```
df['Product'].value_counts()
```

Out[43]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

- Maximum product purchased is KP281 followed by KP481 and KP781

In [44]:
```
df['Gender'].value_counts()
```

Out[44]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

- Maximum number of entries are for Males in given data

In [45]:
```
df['MaritalStatus'].unique()
```

Out[45]:
```
array(['Single', 'Partnered'], dtype=object)
```

In [192]:
```
df['MaritalStatus'].value_counts()
```

Out[192]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

- Given data includes Single and Partnered people

In [46]:
```
print('The maximum avg number of miles the customer expects to walk/run ea
print('The minimum avg number of miles the customer expects to walk/run ea
```

```
The maximum avg number of miles the customer expects to walk/run each we
ek:  360
The minimum avg number of miles the customer expects to walk/run each we
ek:  21
```

In [126]:
```
df['Fitness'].value_counts()
```

Out[126]:
```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

- Maxmium people rate themselves as 3 which can be treated as average

In [47]:
```
# We can convert age value to category
df.head()
```

Out[47]:

	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 [197]:
```
print("Mean miles for Males: ",df.loc[df['Gender']=='Male']['Miles'].mean(
print("Mean miles for Females: ",df.loc[df['Gender']=='Female']['Miles'].r
```

```
Mean miles for Males:  112.82692307692308
Mean miles for Females:  90.01315789473684
```

Creating categories for Age column

1. 18 - 30 Years --> Young
2. 31 - 45 Years --< Adult
3. Above 45 -->Old

In [48]:
```
def age_category(x):

    if x >=18 and x <=30:
        return 'Young'
    elif x >=31 and x <=45:
        return 'Adult'
    else:
        return 'Old'

df['Age_Range'] = df['Age'].apply(age_category)
```

In [49]:
```
df['Age_Range'].value_counts()
```

Out[49]:
```
Young    120
Adult     54
Old        6
Name: Age_Range, dtype: int64
```

Creating categories for education.

1. 12 - 15 Years --> Under Graduate
2. 15 - 18 Years --> Post Graduate
3. Above 18 --> HighlyEducated

In [88]:
```
def edu_category(x):

    if x >=12 and x <=15:
        return 'Under_Graduate'
    elif x >15 and x <=18:
        return 'Post_Graduate'
    else:
        return 'Highly_Educated'

df['Education_Level'] = df['Education'].apply(edu_category)
```

In [90]:
```
df['Education_Level'].value_counts()
```

Out[90]:
```
Post_Graduate      108
Under_Graduate      68
Highly_Educated      4
Name: Education_Level, dtype: int64
```

Observations

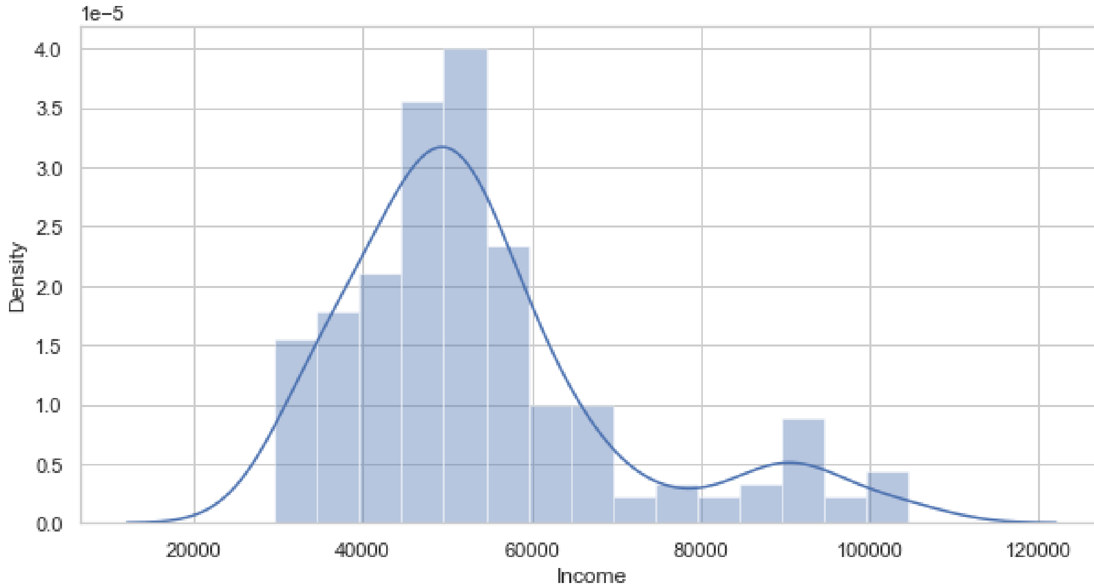
1. Given data includes more young people i.e in range 18 to 30 years
2. High educated count is more in given data.

In [50]:
```
df.groupby(['Product'])['Gender'].value_counts()
```

Out[50]:
```
Product  Gender
KP281    Female    40
         Male      40
KP481    Male      31
         Female    29
KP781    Male      33
         Female     7
Name: Gender, dtype: int64
```

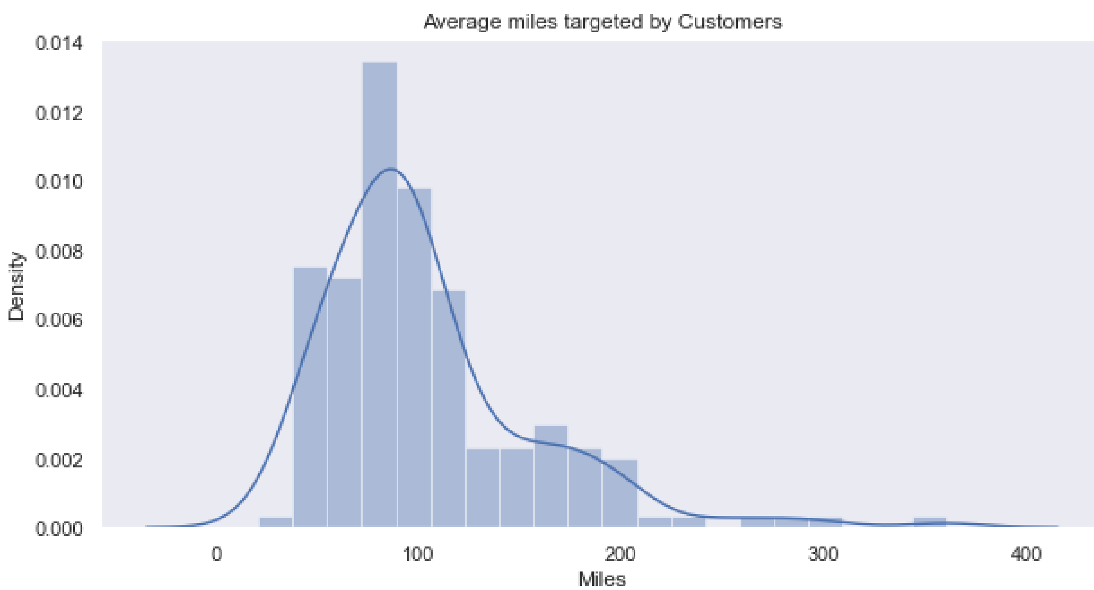
3. Visual Analysis - Univariate

In [52]:
```
fig, ax = plt.subplots(figsize=(10, 5))
r = sns.distplot(df['Income'])
sns.set_style('darkgrid')
```



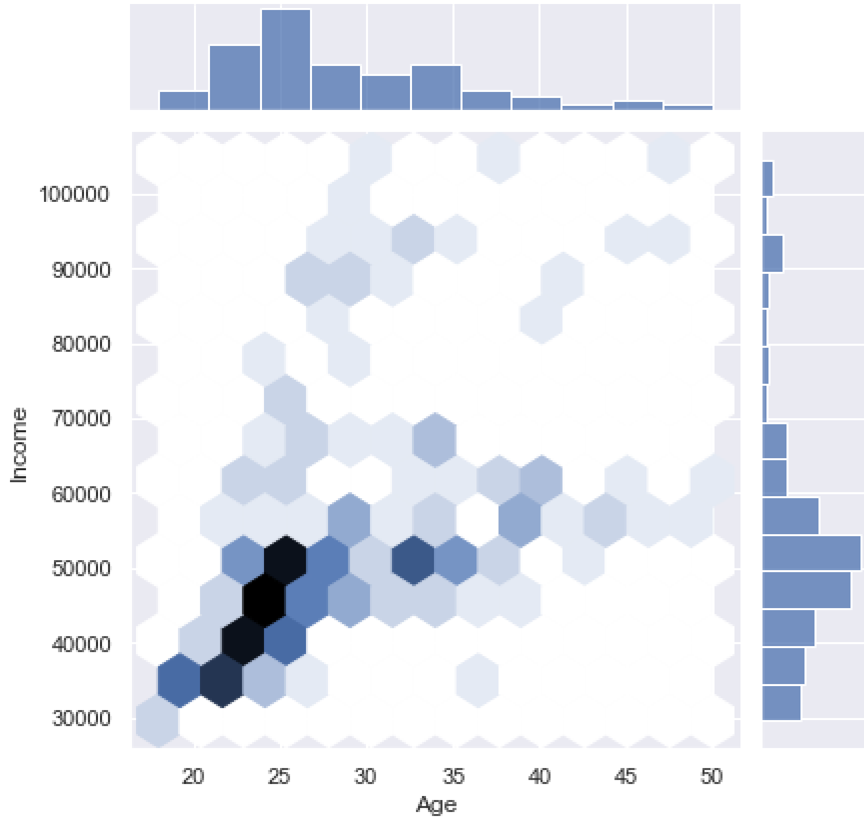
In [ ]:
```
# Distribution of Miles of customers
```

In [198]:
```
fig, ax = plt.subplots(figsize=(10, 5))
r = sns.distplot(df['Miles']).set(title='Average miles targeted by Custome
plt.show()
```



In [54]:
```
sns.jointplot(x ='Age' ,y = 'Income',data = df,kind='hex')
```

Out[54]: <seaborn.axisgrid.JointGrid at 0x1b87dcae220>



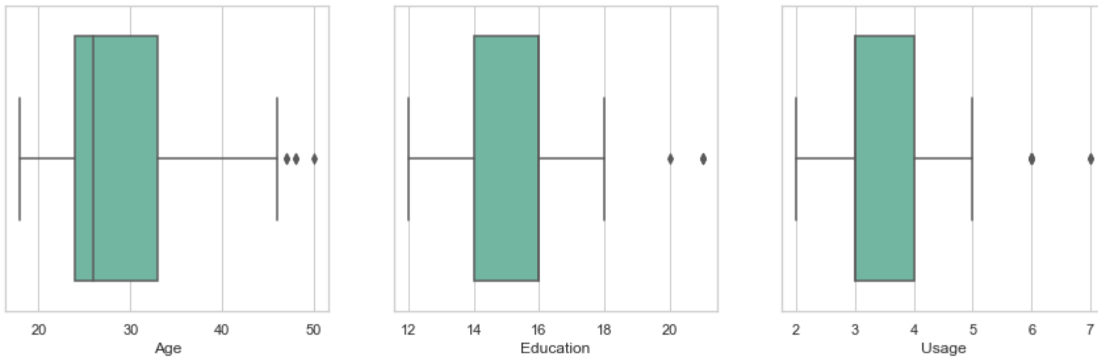
Observation:

- Above graph shows less is the age less is income.
- For age range 20 - 25 income ranges between 30000 to 50000.

Visual Analysis - Bi-variate

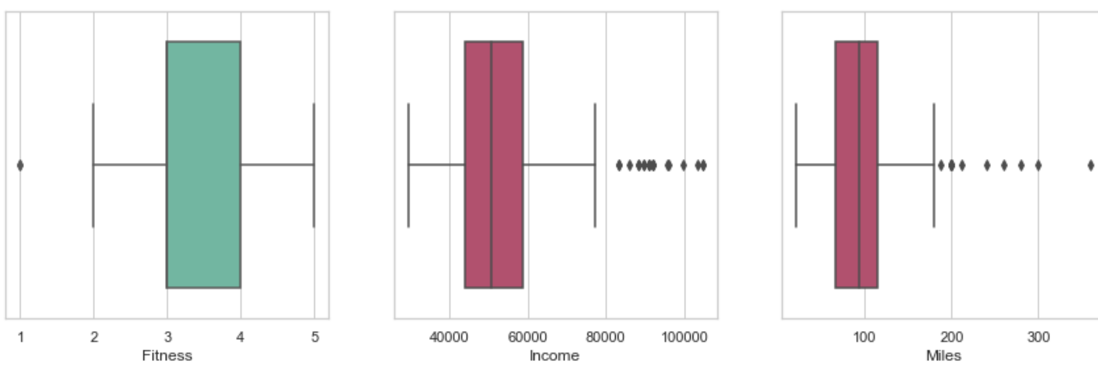
In [123]:

```
fig, axis = plt.subplots(nrows=1, ncols=3, figsize=(15,3))
fig.subplots_adjust(top=1.2)
sns.boxplot(data=df, x="Age", palette='Set2',orient='h', ax=axis[0])
sns.boxplot(data=df, x="Education", palette='Set2',orient='h', ax=axis[1]
sns.boxplot(data=df, x="Usage",palette='Set2', orient='h', ax=axis[2])
plt.show()
```



In [122]:

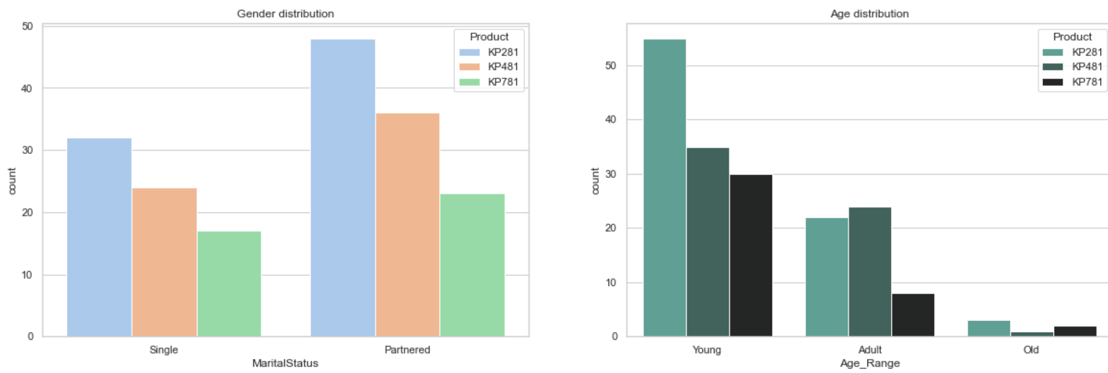
```
fig, axis = plt.subplots(nrows=1, ncols=3, figsize=(15,3))
fig.subplots_adjust(top=1.2)
sns.boxplot(data=df, x="Fitness",palette='Set2',orient='h', ax=axis[0])
sns.boxplot(data=df, x="Income", orient='h',palette='flare', ax=axis[1])
sns.boxplot(data=df, x="Miles", orient='h',palette='flare', ax=axis[2])
plt.show()
```



Observation:

- People tend to use tread mills average 3 to 4 times in a week.
- Income ranges between 40000 to 60000
- Customer expects to walk/run each week average 100 miles.
- There are more outliers in Miles and Income data.
- Very less outliers in Age, Education and usage.

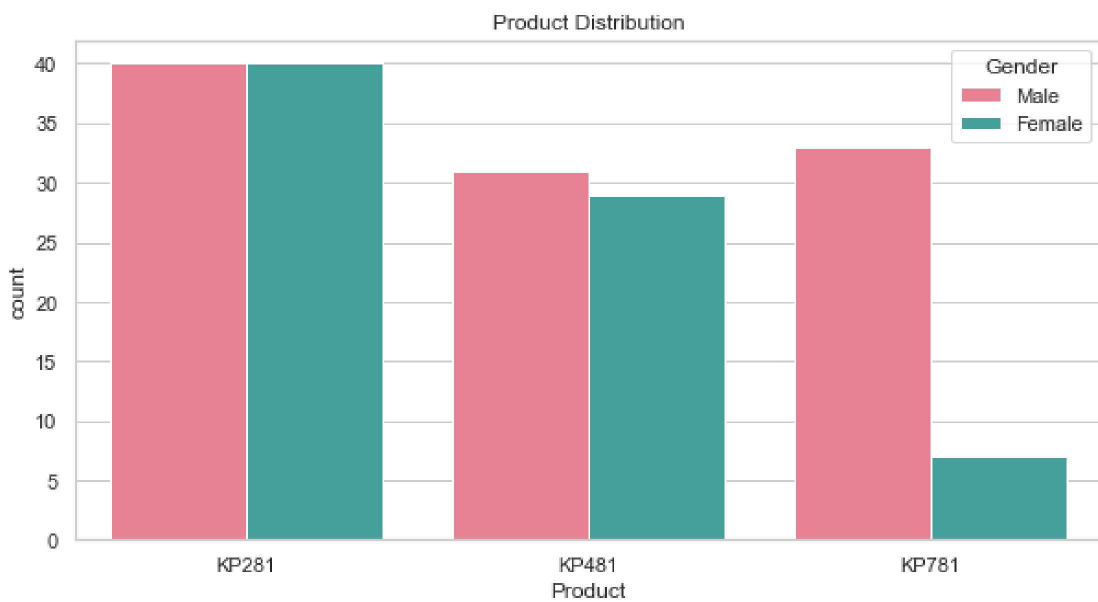
In [79]:
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
sns.countplot(data=df, x='MaritalStatus',hue='Product', palette='pastel',a
sns.countplot(data=df, x='Age_Range',hue='Product',palette='dark:#5A9_r',a
plt.show()
```



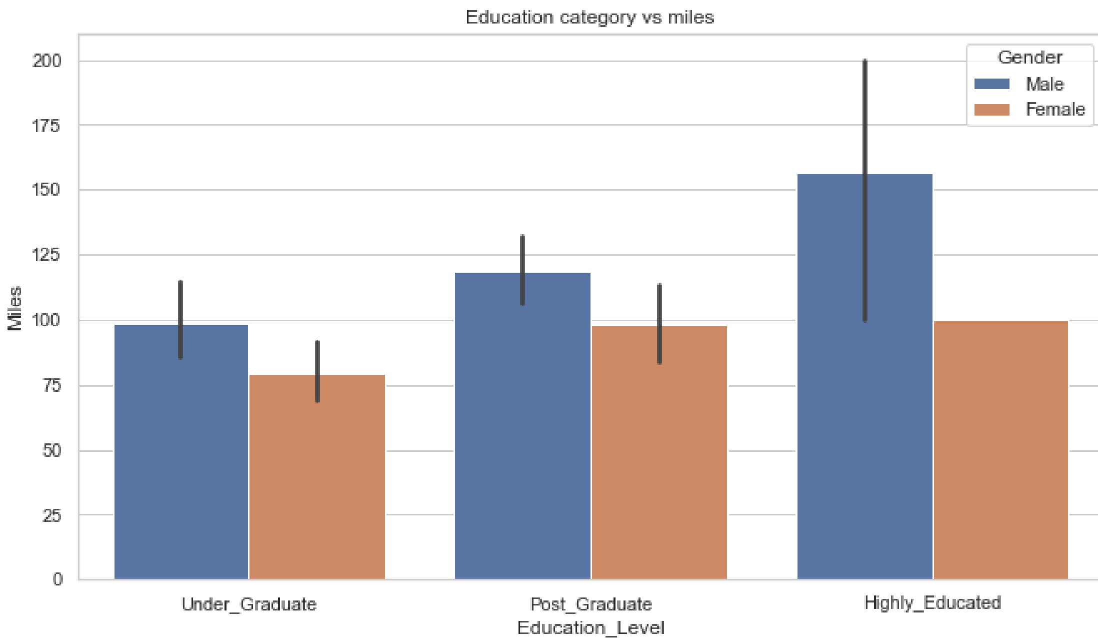
Use of treadmills gender wise

In [74]:
```
fig, ax = plt.subplots(figsize=(10, 5))
sns.countplot(data=df, x='Product', hue='Gender',palette='husl').set(title
```

Out[74]: [Text(0.5, 1.0, 'Product Distribution')]



In [93]:
```
fig, axes = plt.subplots(1, 1,figsize=(11,6))
sns.barplot(data=df,x='Education_Level',y='Miles',hue='Gender')
plt.title('Education category vs miles')
plt.show()
```

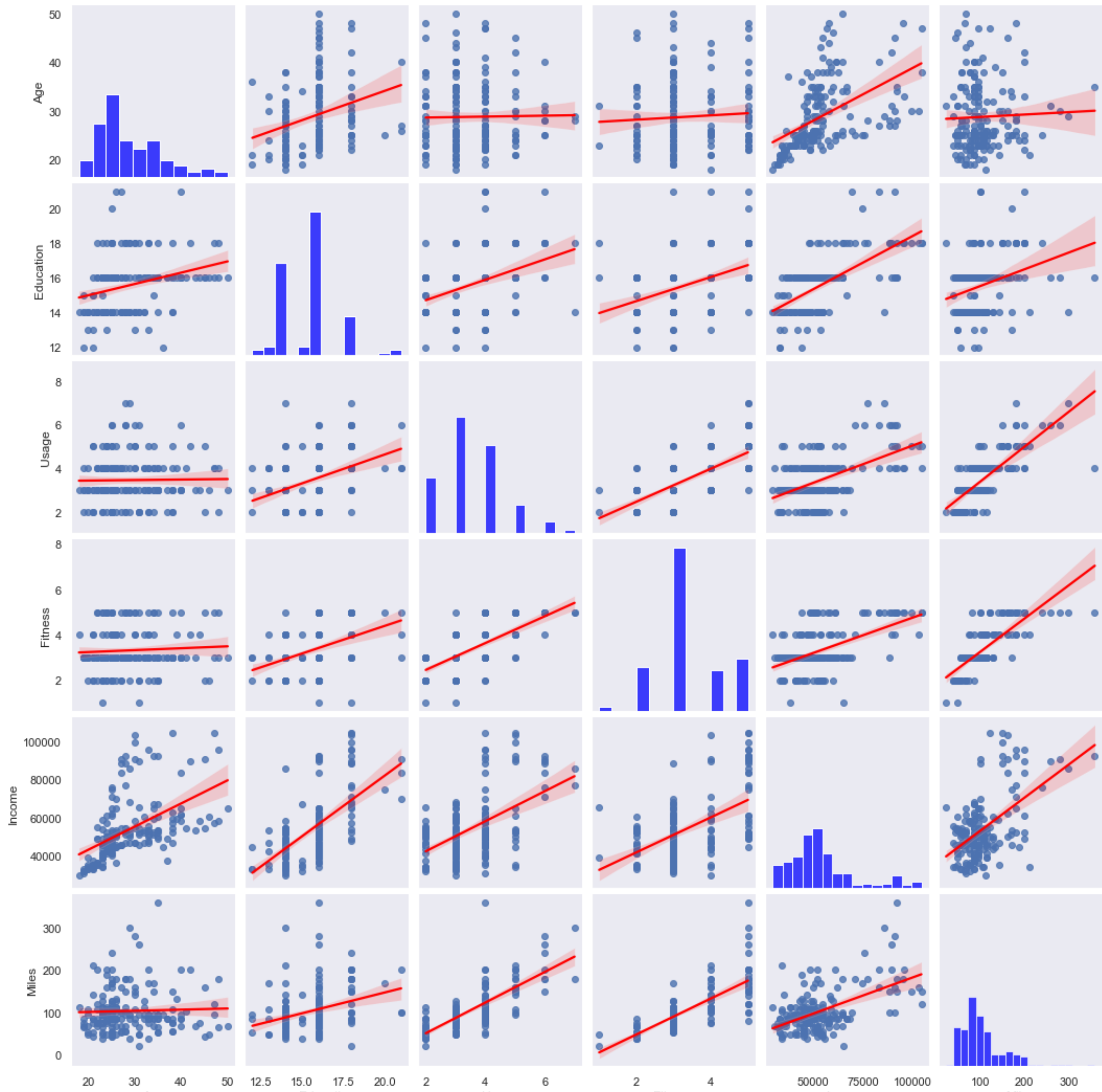


Observation:

- This incidicates highly educated male run more miles than female.
- It also shows as compared to under graduate and post graduate males, highly educated male run more miles.

Visual Analysis - Correlation and Heatmap

In [141]:
```
# pairplot
plt.close()
sns.set_style('dark')
sns.pairplot(df,kind='reg',diag_kind='hist',diag_kws={'color':'blue'},plot
plt.show()
```



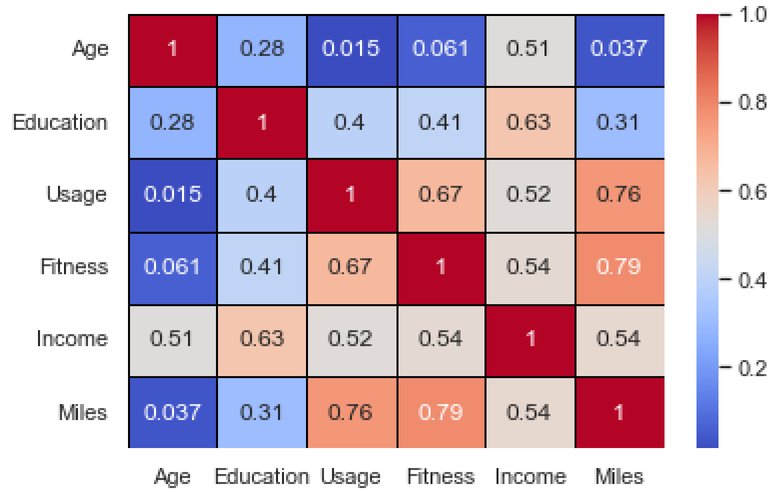
Pairplot Observation:

- As age increases income increases. So age and income are correlated.
- Fitness and Miles : Customers running more are more fit.
- Young age customers tend to run more.
- People within middle income range tend to run more miles.

In [142]:
```
tc = df.corr()
```

In [146]:
```
#sns.heatmap(tc,,cmap='coolwarm')
sns.heatmap(tc,cmap='coolwarm',linecolor='black',linewidths=1,annot=True)#
```

Out[146]: <AxesSubplot:>



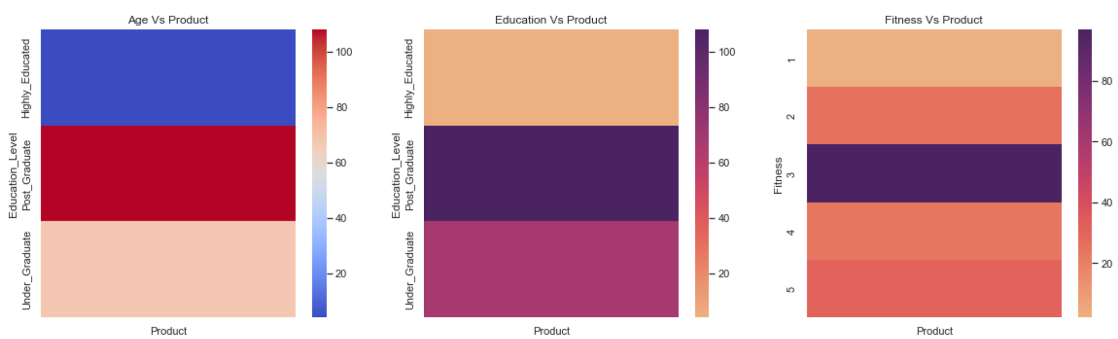
Finding the correlation between Product with Age group, Education and Fitness

In [191]:

```
plt.figure(figsize=(20,12))
df_age = df.groupby('Age_Range')['Product'].count()
df_age = df_age.reset_index('Age_Range')
df_age = df_age.set_index('Age_Range')

df_education = df.groupby('Education_Level')['Product'].count()
df_education = df_education.reset_index('Education_Level')
df_education = df_education.set_index('Education_Level')


df_fitness = df.groupby('Fitness')['Product'].count()
df_fitness = df_fitness.reset_index('Fitness')
df_fitness = df_fitness.set_index('Fitness')

plt.subplot(2, 3, 1)
sns.heatmap(df_education,cmap='coolwarm')
plt.title('Age Vs Product')

plt.subplot(2, 3, 2)
sns.heatmap(df_education,cmap='flare')
plt.title('Education Vs Product')

plt.subplot(2, 3, 3)
sns.heatmap(df_fitness,cmap='flare')
plt.title('Fitness Vs Product')
```

Out[191]:

```
Text(0.5, 1.0, 'Fitness Vs Product')
```



Correlation Observation:

- Young customer, Post gratudate custmoers are buying more tread mills
- Customer rating there fitness level as 3 are buying more treadmills

In [ ]:

Contingency Table

In [202]:

```
Product = df['Product']
Gender = df['Gender']
pd.crosstab(Product,Gender,rownames=['Products'],colnames=['Gender'],marg
```

Out[202]:

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

In [221]:

```
print('Checking the percentage Gender Wise: Male')
print('-'*50)

print('Percentage of Male customers purchases KP281 = ',round(100*40/104,
print('Percentage of Male customers purchases KP481 = ',round(100*31/104,
print('Percentage of Male customers purchases KP781 = ',round(100*33/104,
print()

print('Checking the percentage Gender Wise: Female')
print('-'*50)
print('Percentage of Female customers purchases KP281 = ',round(100*40/76,
print('Percentage of Female customers purchases KP481 = ',round(100*29/76,
print('Percentage of Female customers purchases KP781 = ',round(100*7/76,
print()

print('-'*50)
print('Overall Percentage of customers purchases KP281 = ',round(100*80/18
print('Overall Percentage of customers purchases KP481 = ',round(100*60/18
print('Overall Percentage of customers purchases KP781 = ',round(100*40/18
```

```
Checking the percentage Gender Wise: Male
--------------------------------------------------
Percentage of Male customers purchases KP281 =  38.46
Percentage of Male customers purchases KP481 =  29.81
Percentage of Male customers purchases KP781 =  31.73

Checking the percentage Gender Wise: Female
--------------------------------------------------
Percentage of Female customers purchases KP281 =  52.63
Percentage of Female customers purchases KP481 =  38.16
Percentage of Female customers purchases KP781 =  9.21

--------------------------------------------------
Overall Percentage of customers purchases KP281 =  44.44
Overall Percentage of customers purchases KP481 =  33.33
Overall Percentage of customers purchases KP781 =  22.22
```

In [ ]:

```

```

In [220]:

```
Product = df['Product']
Age = df['Age_Range']
pd.crosstab(Age,Product,rownames=['Age_Range'],colnames=['Products'],marg:
```

Out[220]:

Products	KP281	KP481	KP781	All
Age_Range				
Adult	22	24	8	54
Old	3	1	2	6
Young	55	35	30	120
All	80	60	40	180

In [225]:

```
print('Checking the percentage Age Wise: Young')
print('-'*50)

print('Percentage of Young customers purchases KP281 = ',round(100*55/120,
print('Percentage of Young customers purchases KP481 = ',round(100*35/120,
print('Percentage of Young customers purchases KP781 = ',round(100*30/120,
print()

print('Checking the percentage Age Wise: Adult')
print('-'*50)
print('Percentage of Adult customers purchases KP281 = ',round(100*22/54,2
print('Percentage of Adult customers purchases KP481 = ',round(100*24/54,2
print('Percentage of Adult customers purchases KP781 = ',round(100*8/54,2)
print()

print('Checking the percentage Old Wise: Old')
print('-'*50)
print('Percentage of Old customers purchases KP281 = ',round(100*3/6,2))
print('Percentage of Old customers purchases KP481 = ',round(100*1/6,2))
print('Percentage of Old customers purchases KP781 = ',round(100*2/6,2))
```

```
Checking the percentage Age Wise: Young
--------------------------------------------------
Percentage of Young customers purchases KP281 =  45.83
Percentage of Young customers purchases KP481 =  29.17
Percentage of Young customers purchases KP781 =  25.0

Checking the percentage Age Wise: Adult
--------------------------------------------------
Percentage of Adult customers purchases KP281 =  40.74
Percentage of Adult customers purchases KP481 =  44.44
Percentage of Adult customers purchases KP781 =  14.81

Checking the percentage Old Wise: Old
--------------------------------------------------
Percentage of Old customers purchases KP281 =  50.0
Percentage of Old customers purchases KP481 =  16.67
Percentage of Old customers purchases KP781 =  33.33
```

5. Insights

Range of attributes

1. The given data contains 104 Male customers and 76 Female customers.
2. Out of them 107 are partnered and 73 single.
3. Product category KP281 shows highest sales.
4. Income shows that majority customer income lie between 30k to 70k.

Correlation

5. We can see that income and age are highly correlated.
6. Similarly miles and fitness are highly correlated.

7. People rating there fitness as 3 are buying more treadmills.

Outliers

8. From boxplot we can observe age, education and usage are having very few outliers.
9. While income and miles are having more outliers.
10. Male customers tend to run more with mean 112.82 as compared to 90.01 miles for female.

Recommendations

Gender, fitness level and education level

1. Business should target customers within age range 18 to 35 are they are making.
2. People with fitness Level 3 or less are likely to purchase KP281 and KP481.
3. People with Education levels less than or equal to 16 are likely to purchase KP281 and KP481.
4. Males have high chances of purchasing KP781 as 82% of total sale of KP781 is purchased by Males.

Category KP781

5. As KP781 category is expensive, business can offer schemes so that it would attract more female user.
6. KP781 this category of product shows less sales irrespective of income, age and gender.
7. So it needs modification or cost reduction.
8. People with Education levels greater than or equal to 16 are likely to purchase KP781.

Category KP481

9. Company should offer more feature on KP481 as customer so that it can reach sales level of KP281.
10. Offers can be provided to married people as share large percentage in sales.