

aerofit-treadmill-2

June 13, 2024

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

```
[ ]: df=pd.read_csv('/content/aerofit_treadmill.csv')
df
```

```
[ ]:
   Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  \
0    KP281   18   Male         14         Single      3         4   29562
1    KP281   19   Male         15         Single      2         3   31836
2    KP281   19  Female         14   Partnered      4         3   30699
3    KP281   19   Male         12         Single      3         3   32973
4    KP281   20   Male         13   Partnered      4         2   35247
..  ...  ...
175  KP781   40   Male         21         Single      6         5   83416
176  KP781   42   Male         18         Single      5         4   89641
177  KP781   45   Male         16         Single      5         5   90886
178  KP781   47   Male         18   Partnered      4         5  104581
179  KP781   48   Male         18   Partnered      4         5   95508
```

```

      Miles
0       112
1        75
2        66
3        85
4        47
..      ...
175     200
176     200
177     160
178     120
179     180
```

[180 rows x 9 columns]

```
[ ]: 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

```

```
[ ]: df.keys()
```

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

```
[ ]: df.sample(10)
```

```
[ ]:
   Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  \
99   KP481   25   Male         16     Partnered      2         2   52302
42   KP281   27   Male         16         Single      4         3   54576
140  KP781   22   Male         14         Single      4         3   48658
75   KP281   43   Male         16     Partnered      3         3   53439
100  KP481   25  Female         14     Partnered      5         3   47754
68   KP281   38   Male         16     Partnered      3         3   46617
62   KP281   34  Female         16         Single      2         2   52302
25   KP281   24   Male         13     Partnered      3         2   42069
40   KP281   26   Male         16         Single      3         3   50028
125  KP481   34  Female         16     Partnered      4         3   64809

      Miles
99        42
42        85
140       106
75        66
100       106
68        75
62        66
25        47
40        85

```

125 95

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

```
[ ]:
      count      Age  Education      Usage      Fitness      Income \
count  180.000000  180.000000  180.000000  180.000000  180.000000
mean    28.788889   15.572222   3.455556   3.311111  53719.577778
std      6.943498    1.617055   1.084797   0.958869  16506.684226
min     18.000000   12.000000   2.000000   1.000000  29562.000000
25%     24.000000   14.000000   3.000000   3.000000  44058.750000
50%     26.000000   16.000000   3.000000   3.000000  50596.500000
75%     33.000000   16.000000   4.000000   4.000000  58668.000000
max     50.000000   21.000000   7.000000   5.000000 104581.000000

      Miles
count  180.000000
mean   103.194444
std     51.863605
min     21.000000
25%     66.000000
50%     94.000000
75%    114.750000
max    360.000000
```

```
[ ]: df.shape
```

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

```
[ ]: df.dtypes
```

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

```
[ ]: 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

```
[ ]: df.tail()
```

```
[ ]:
   Product  Age Gender Education MaritalStatus  Usage  Fitness  Income \
175  KP781  40  Male         21         Single      6        5   83416
176  KP781  42  Male         18         Single      5        4   89641
177  KP781  45  Male         16         Single      5        5   90886
178  KP781  47  Male         18        Partnered      4        5  104581
179  KP781  48  Male         18        Partnered      4        5   95508
```

	Miles
175	200
176	200
177	160
178	120
179	180

```
[ ]: df.isnull().sum()
```

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

```
[ ]: df.value_counts()
```

```
[ ]: Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  Miles
KP281    18  Male     14         Single           3      4      29562   112
1
KP481    30  Female   13         Single           4      3      46617   106
1
      31  Female   16         Partnered          2      3      51165    64
1
      18         Single           2      1      65220    21
1
      Male    16         Partnered           3      3      52302    95
1
...
KP281    34  Female   16         Single           2      2      52302    66
```

```

1
      Male    16      Single    4    5      51165    169
1
      35  Female  16      Partnered    3    3      60261    94
1
      18      Single    3    3      67083    85
1
KP781    48  Male    18      Partnered    4    5      95508    180
1
Name: count, Length: 180, dtype: int64

```

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

```
[ ]: 0
```

```
[ ]: df_Gender=(df["Gender"])
df_Gender
```

```

[ ]: 0      Male
1      Male
2     Female
3      Male
4      Male
...
175     Male
176     Male
177     Male
178     Male
179     Male
Name: Gender, Length: 180, dtype: object

```

```
[ ]: df_Gender.value_counts()
```

```

[ ]: Gender
Male      104
Female     76
Name: count, dtype: int64

```

```
[ ]: df_male=(df_Gender[df_Gender=="Male"])
df_male
```

```

[ ]: 0      Male
1      Male
3      Male
4      Male
7      Male
...

```

```
175    Male
176    Male
177    Male
178    Male
179    Male
Name: Gender, Length: 104, dtype: object
```

```
[ ]: df_male.value_counts()
```

```
[ ]: Gender
      Male      104
      Name: count, dtype: int64
```

```
[ ]: df_product=(df["Product"])
      df.Product
```

```
[ ]: 0      KP281
      1      KP281
      2      KP281
      3      KP281
      4      KP281
      ...
      175    KP781
      176    KP781
      177    KP781
      178    KP781
      179    KP781
      Name: Product, Length: 180, dtype: object
```

```
[ ]: df["Product"].value_counts(normalize=True)
```

```
[ ]: Product
      KP281    0.444444
      KP481    0.333333
      KP781    0.222222
      Name: proportion, dtype: float64
```

```
[ ]: df["Age"].unique()
```

```
[ ]: array([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
           35, 36, 37, 38, 39, 40, 41, 43, 44, 46, 47, 50, 45, 48, 42])
```

```
[ ]: df["Gender"].value_counts(normalize=True)
```

```
[ ]: Gender
      Male      0.577778
      Female    0.422222
```

Name: proportion, dtype: float64

```
[ ]: df["Education"].unique()
```

```
[ ]: array([14, 15, 12, 13, 16, 18, 20, 21])
```

```
[ ]: df["MaritalStatus"].value_counts(normalize=True)
```

```
[ ]: MaritalStatus
      Partnered    0.594444
      Single      0.405556
      Name: proportion, dtype: float64
```

```
[ ]: df["Usage"].value_counts(normalize=True)
```

```
[ ]: Usage
      3    0.383333
      4    0.288889
      2    0.183333
      5    0.094444
      6    0.038889
      7    0.011111
      Name: proportion, dtype: float64
```

```
[ ]: df['Fitness'].value_counts(normalize=True)
```

```
[ ]: Fitness
      3    0.538889
      5    0.172222
      2    0.144444
      4    0.133333
      1    0.011111
      Name: proportion, dtype: float64
```

```
[ ]: df["Income"].value_counts(normalize=True)
```

```
[ ]: Income
      45480    0.077778
      52302    0.050000
      46617    0.044444
      54576    0.044444
      53439    0.044444
      ...
      65220    0.005556
      55713    0.005556
      68220    0.005556
      30699    0.005556
```

```
95508    0.005556
Name: proportion, Length: 62, dtype: float64
```

```
[ ]: df["Miles"].value_counts(normalize=True)
```

```
[ ]: Miles
85      0.150000
95      0.066667
66      0.055556
75      0.055556
47      0.050000
106     0.050000
94      0.044444
113     0.044444
53      0.038889
100     0.038889
180     0.033333
200     0.033333
56      0.033333
64      0.033333
127     0.027778
160     0.027778
42      0.022222
150     0.022222
38      0.016667
74      0.016667
170     0.016667
120     0.016667
103     0.016667
132     0.011111
141     0.011111
280     0.005556
260     0.005556
300     0.005556
240     0.005556
112     0.005556
212     0.005556
80      0.005556
140     0.005556
21      0.005556
169     0.005556
188     0.005556
360     0.005556
Name: proportion, dtype: float64
```

```
[ ]: df["Usage"].value_counts(normalize=True)
```



```
[ ]: Usage
3    0.383333
4    0.288889
2    0.183333
5    0.094444
6    0.038889
7    0.011111
Name: proportion, dtype: float64
```

```
[ ]: df.value_counts()
```

```
[ ]: Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  Miles
KP281    18  Male    14         Single           3      4      29562   112
1
KP481    30  Female  13         Single           4      3      46617   106
1
          31  Female  16         Partnered        2      3      51165    64
1
          18         Single           2      1      65220    21
1
          Male    16         Partnered        3      3      52302    95
1
...
KP281    34  Female  16         Single           2      2      52302    66
1
          Male    16         Single           4      5      51165   169
1
          35  Female  16         Partnered        3      3      60261    94
1
          18         Single           3      3      67083    85
1
KP781    48  Male    18         Partnered        4      5      95508   180
1
Name: count, Length: 180, dtype: int64
```

#CHECKING NULL VALUES

```
[ ]: df.isna().sum()
```

```
[ ]: Product      0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
```

```
Miles          0
dtype: int64
```

CHECKING DUPLICATED VALUES

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

```
[ ]: 0
```

```
[ ]: from matplotlib import rcParams
rcParams['figure.figsize'] = 20,7
```

VISUAL ANALYSIS

3.3.CORRELATION

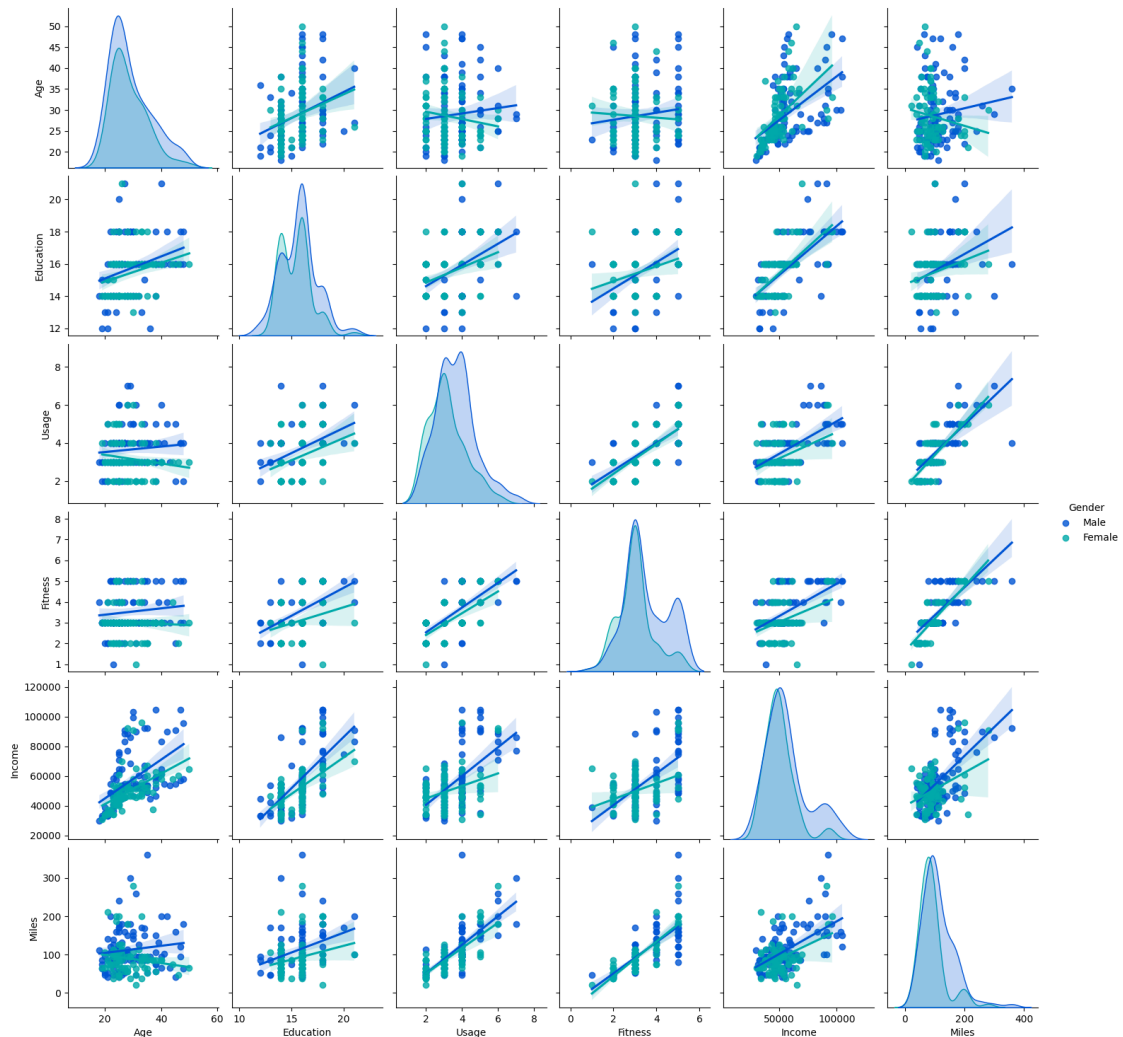
```
[ ]: # Convert relevant columns to numeric type before calculating correlation
numeric_df = df.apply(pd.to_numeric, errors='coerce')

# Calculate and plot the correlation matrix
sns.heatmap(numeric_df.corr(), annot=True, cmap='YlGnBu')
plt.show()
```



PAIR PLOT

```
[ ]: # pairplot
rcParams['figure.figsize'] = 20, 7
sns.pairplot(df, palette='winter', hue='Gender', kind='reg')
plt.show()
```



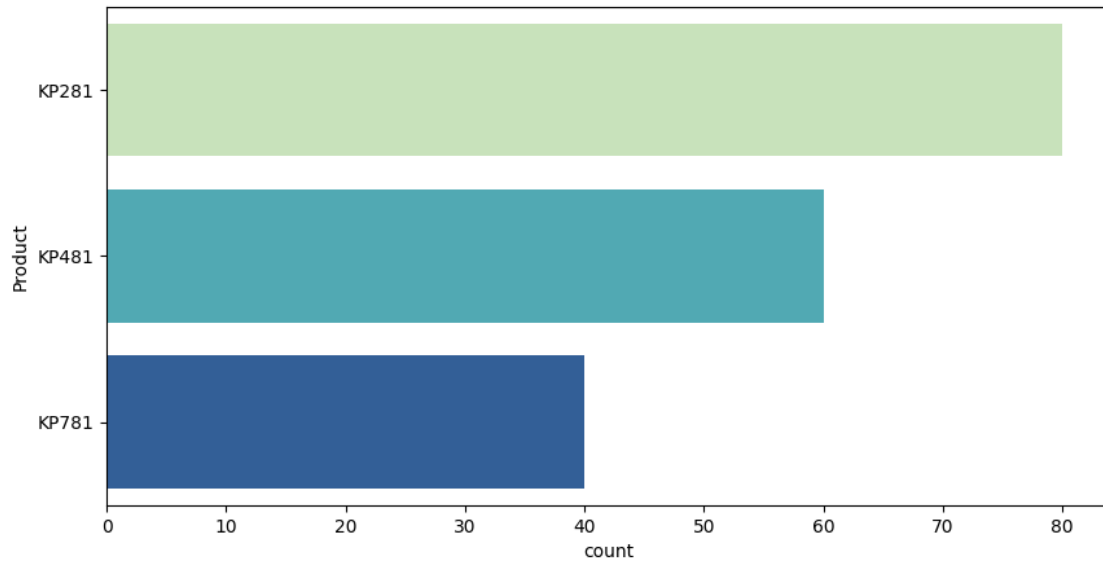
UNIVARIATE ANALYSIS

```
[ ]: #Product
plt.figure(figsize=(10,5))
sns.countplot(data=df, y='Product', palette='YlGnBu')
plt.show()
```

<ipython-input-29-20b140a19820>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, y='Product', palette='YlGnBu')
```

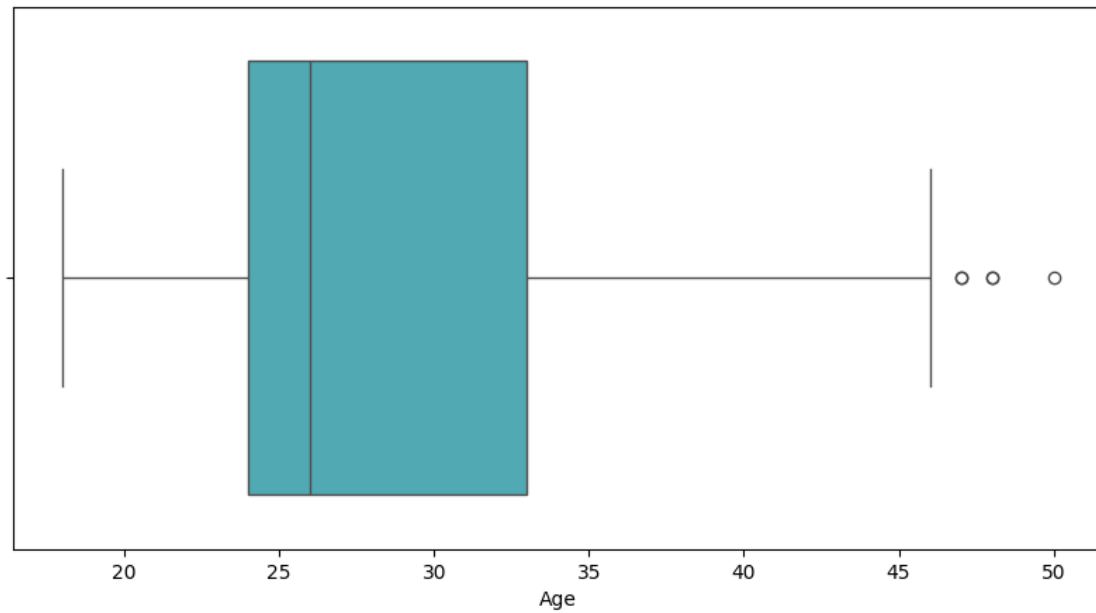


```
[ ]: #product
plt.figure(figsize=(10,5))
sns.boxplot(data=df, x='Age', palette='YlGnBu')
plt.show()
```

<ipython-input-35-05a91822b015>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=df, x='Age', palette='YlGnBu')
```

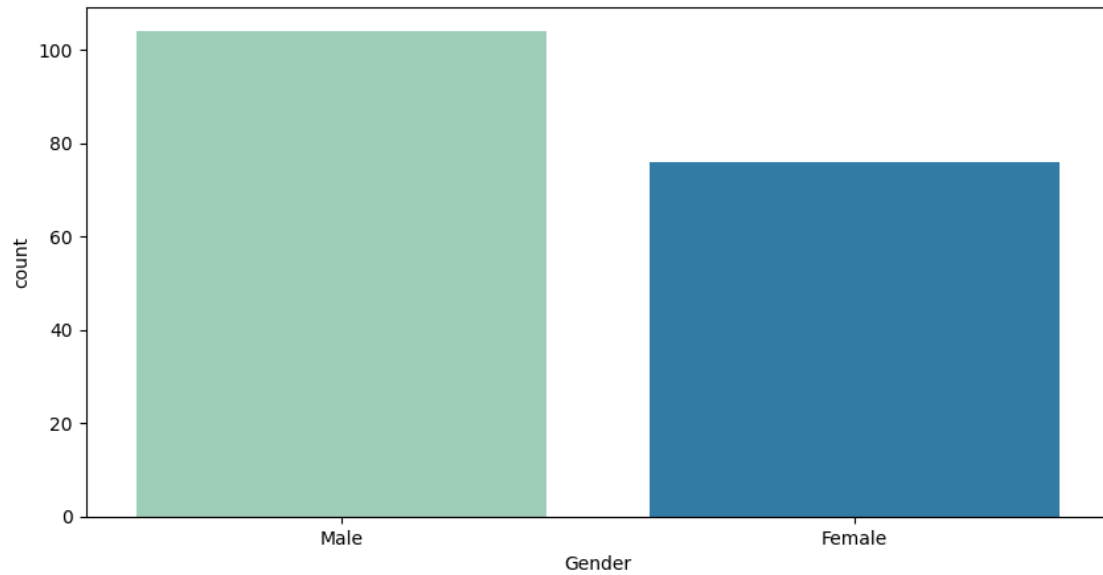


```
[ ]: #Gender
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='Gender', palette='YlGnBu')
plt.show()
```

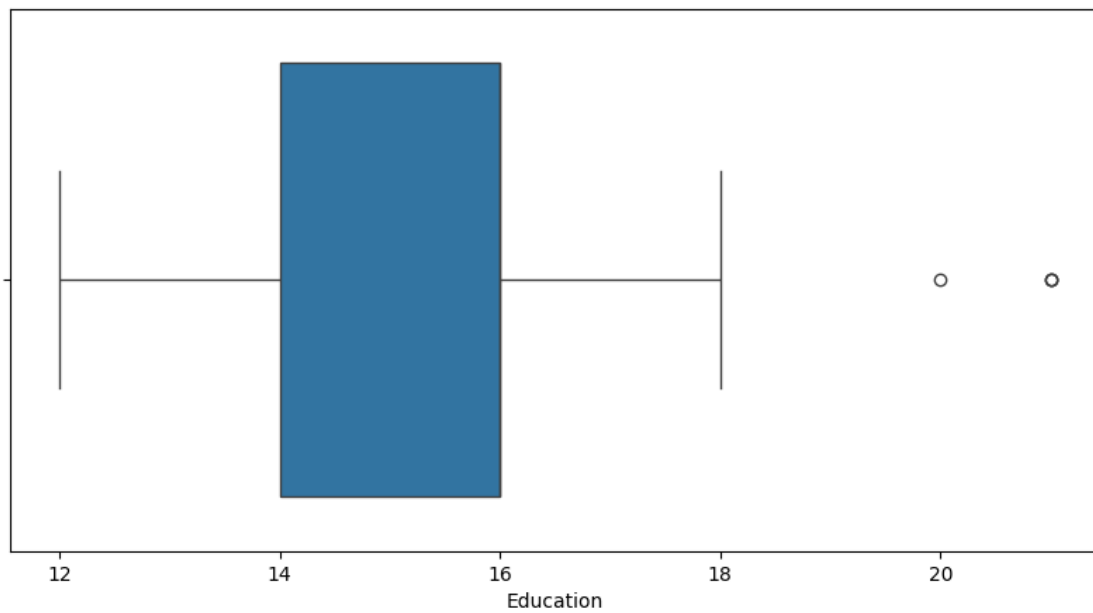
<ipython-input-36-91ad2e08f10c>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, x='Gender', palette='YlGnBu')
```



```
[ ]: #Education
plt.figure(figsize=(10,5))
sns.boxplot(data=df,x="Education")
plt.show()
```



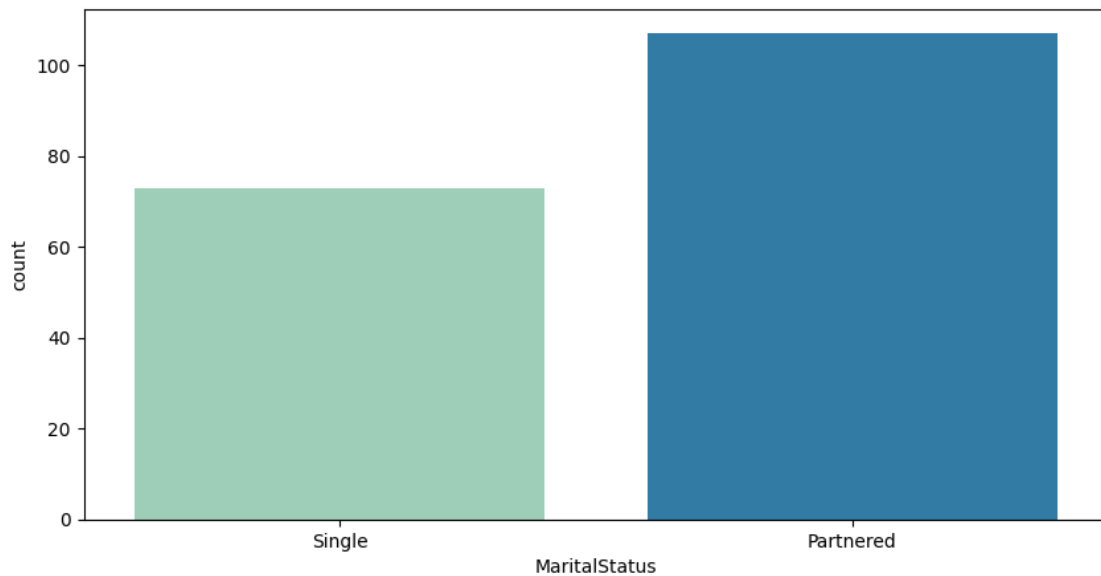
```
[ ]: #Marital status)
plt.figure(figsize=(10,5))
```

```
sns.countplot(data=df, x="MaritalStatus", palette="YlGnBu")
plt.show()
```

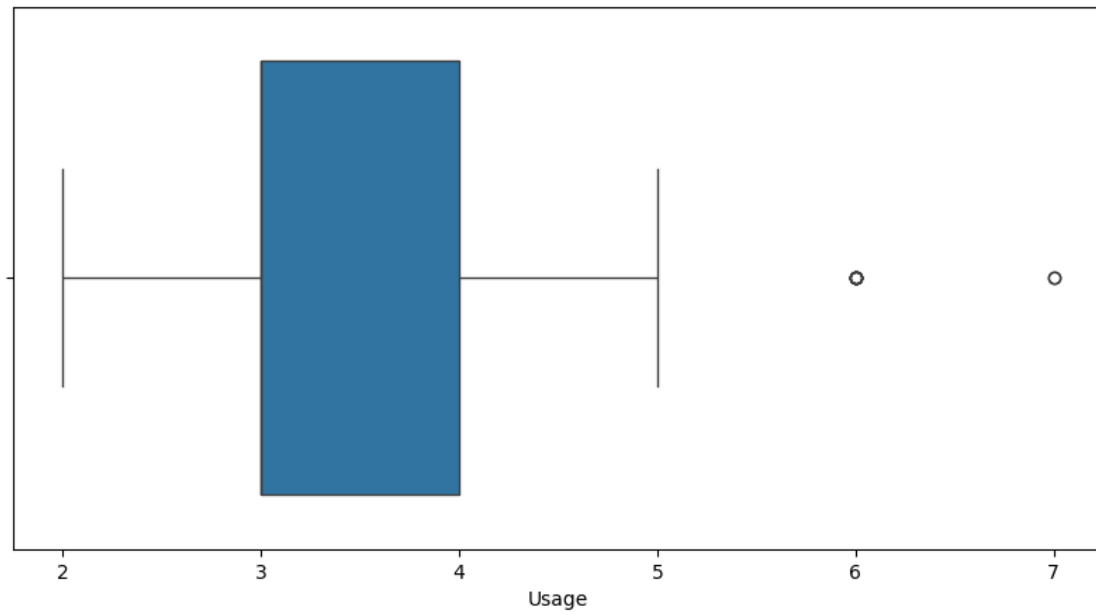
<ipython-input-46-e10f0c142901>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, x="MaritalStatus", palette="YlGnBu")
```

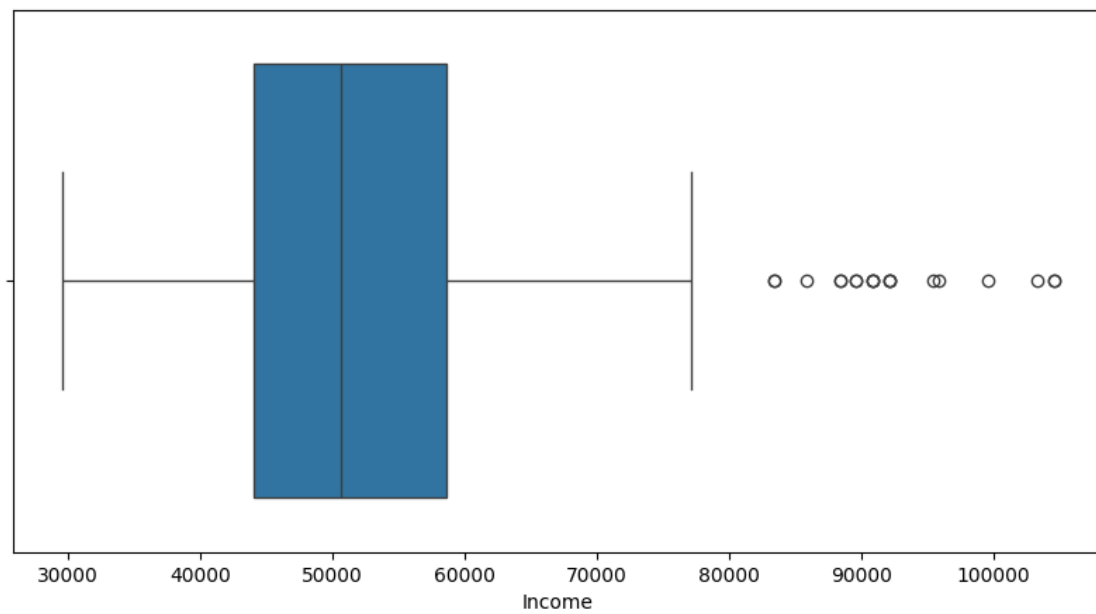


```
[ ]: #Usage Boxplot
plt.figure(figsize=(10,5))
sns.boxplot(data=df,x="Usage")
plt.show()
```

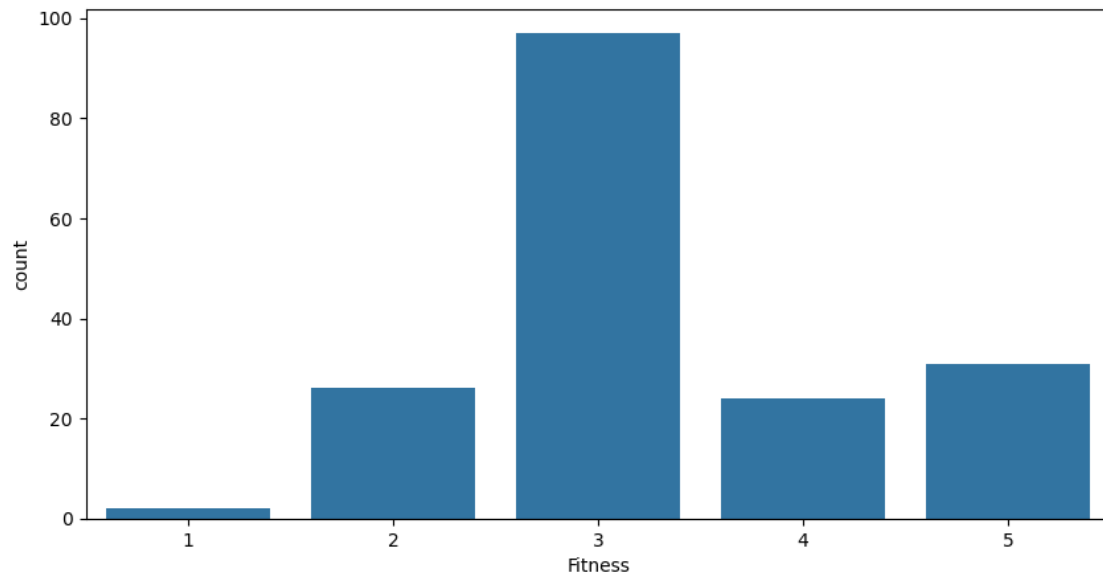


```
[ ]: #Box Plot Income

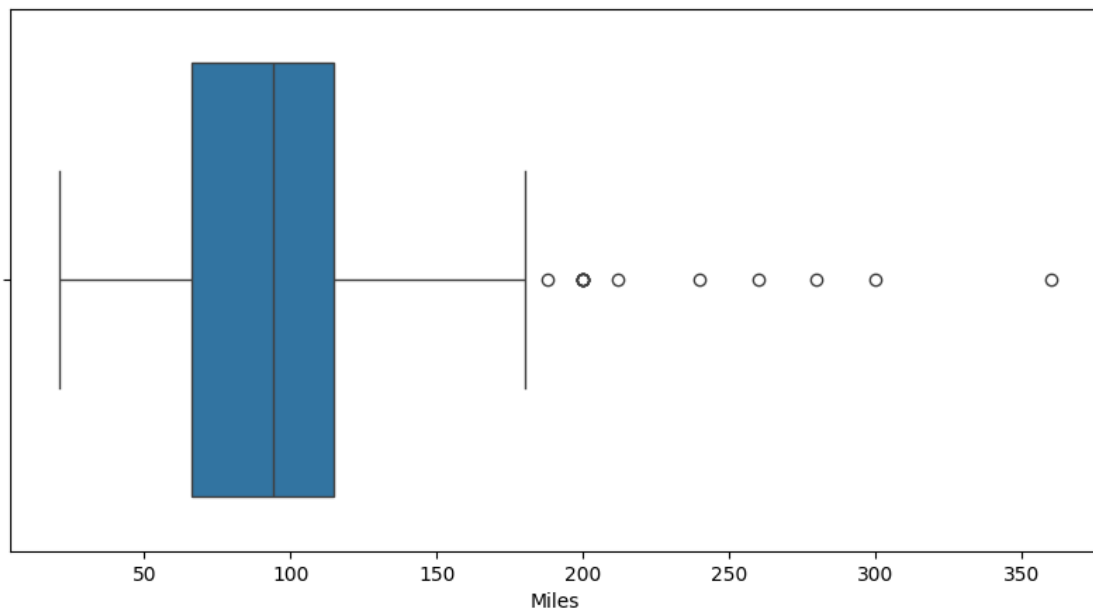
plt.figure(figsize=(10,5))
sns.boxplot(data=df,x="Income")
plt.show()
```




```
[ ]: #Countplot fitness
plt.figure(figsize=(10,5))
sns.countplot(data=df, x="Fitness")
plt.show()
```



```
[ ]: #Miles bpxplot
plt.figure(figsize=(10,5))
sns.boxplot(data=df, x="Miles")
plt.show()
```



```
[ ]: #Average Income of customer buying each model  
df.groupby("Product")["Income"].mean()
```

```
[ ]: Product  
KP281    46418.025  
KP481    48973.650  
KP781    75441.575  
Name: Income, dtype: float64
```

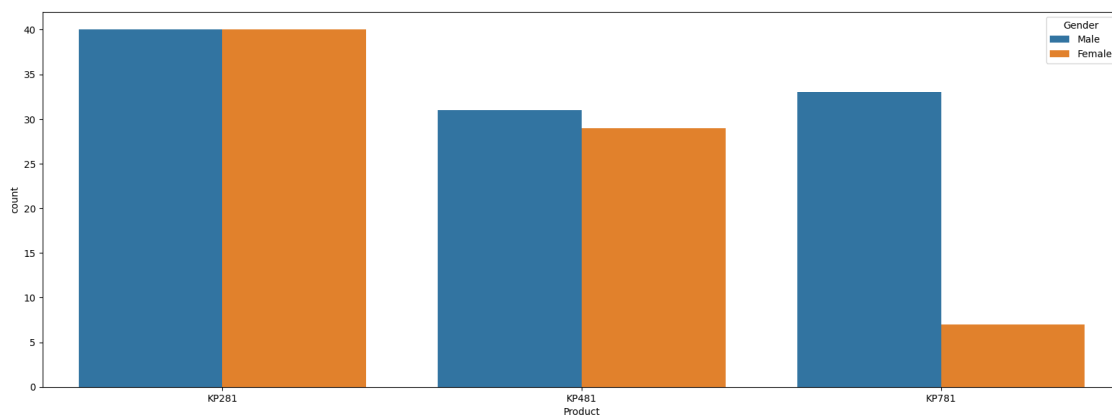
```
[ ]: #Average Usage of customer buying each model  
df.groupby("Product")["Usage"].mean()
```

```
[ ]: Product  
KP281     3.087500  
KP481     3.066667  
KP781     4.775000  
Name: Usage, dtype: float64
```

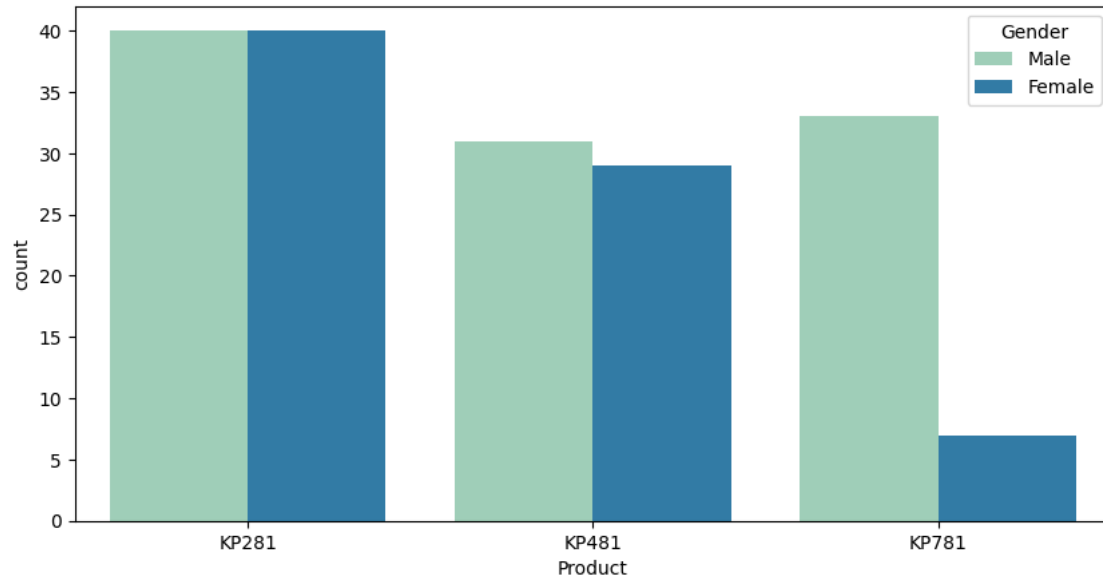
```
[ ]: #Average Fitness of customer buying each model  
df.groupby("Product")["Fitness"].mean()
```

```
[ ]: Product  
KP281     2.9625  
KP481     2.9000  
KP781     4.6250  
Name: Fitness, dtype: float64
```

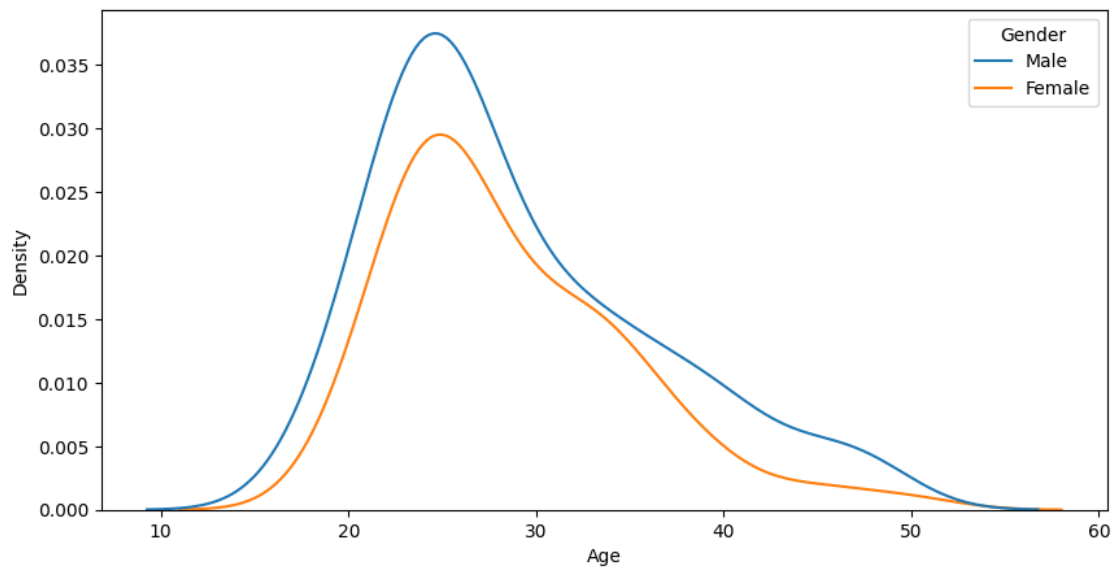
```
[ ]: # Gender & Product  
sns.countplot(data=df, x='Product', hue='Gender')  
plt.show()
```



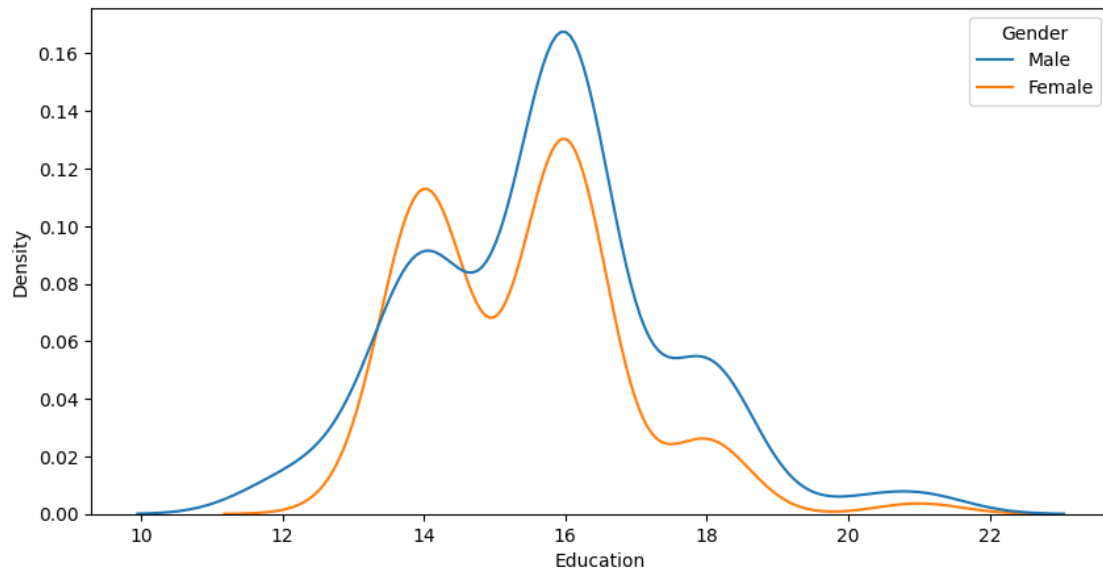
```
[ ]: #Gender & Product
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='Product', hue='Gender', palette='YlGnBu')
plt.show()
```



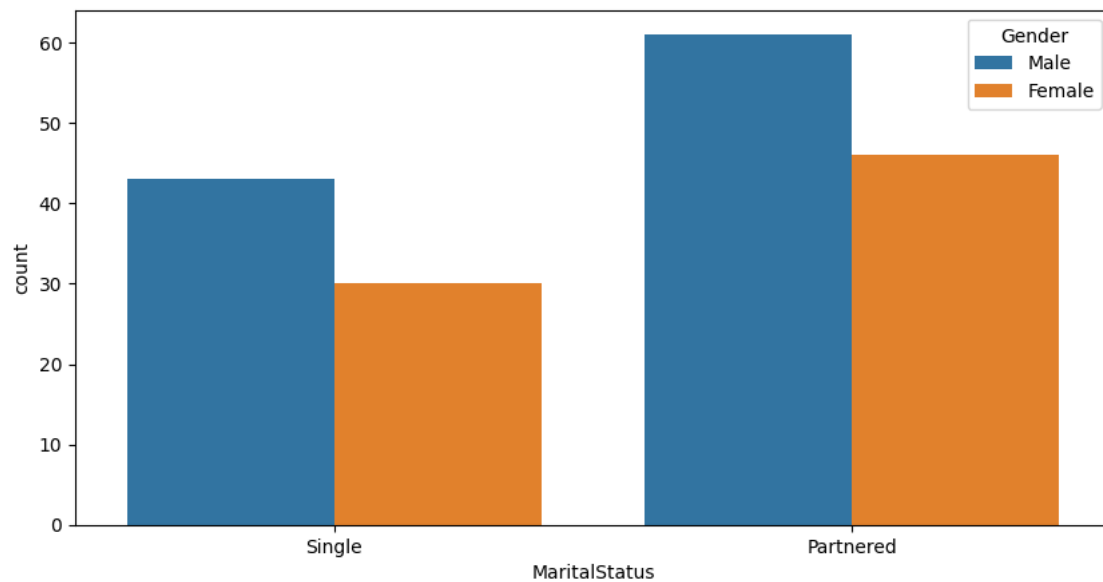
```
[ ]: #Gender & Product
plt.figure(figsize=(10,5))
sns.kdeplot(data=df, x="Age", hue="Gender")
plt.show()
```



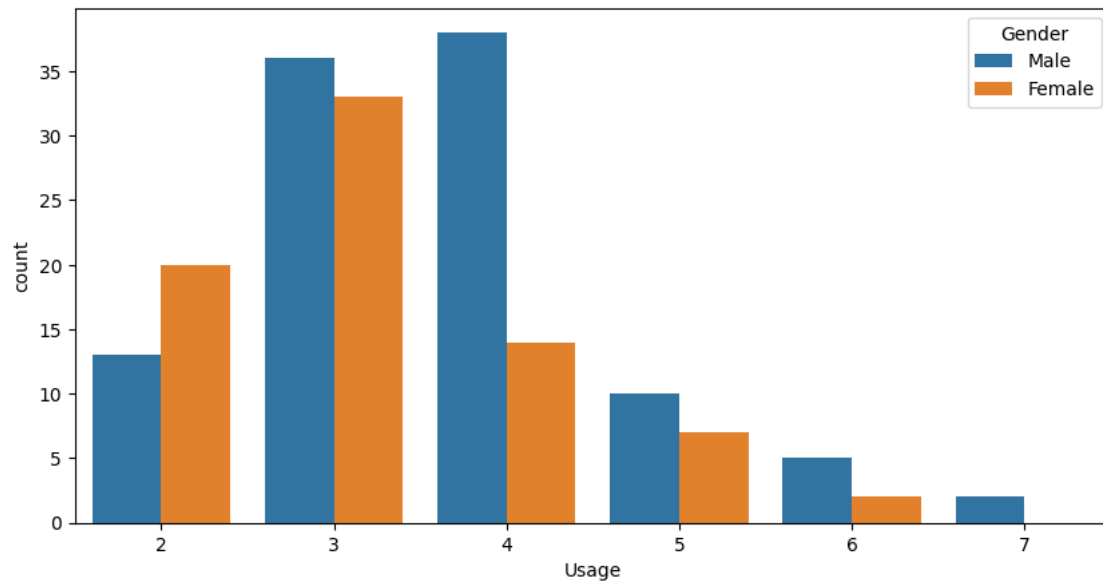
```
[ ]: plt.figure(figsize=(10,5))
sns.kdeplot(data=df, x="Education", hue="Gender")
plt.show()
```



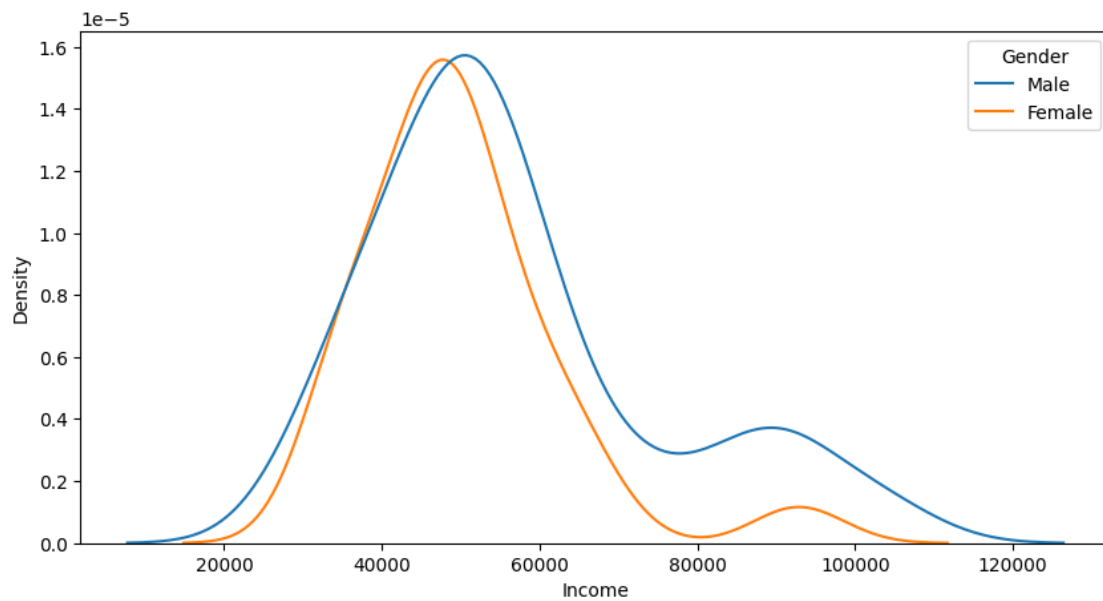
```
[ ]: plt.figure(figsize=(10,5))
sns.countplot(data=df, x='MaritalStatus', hue='Gender')
plt.show()
```



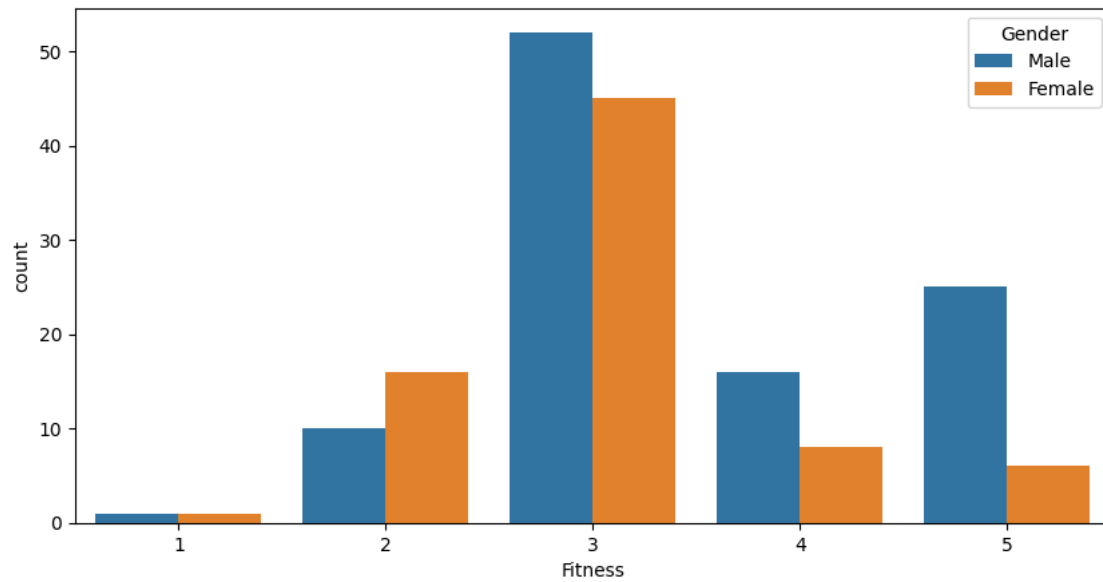
```
[ ]: plt.figure(figsize=(10,5))
sns.countplot(data=df, x='Usage', hue='Gender')
plt.show()
```



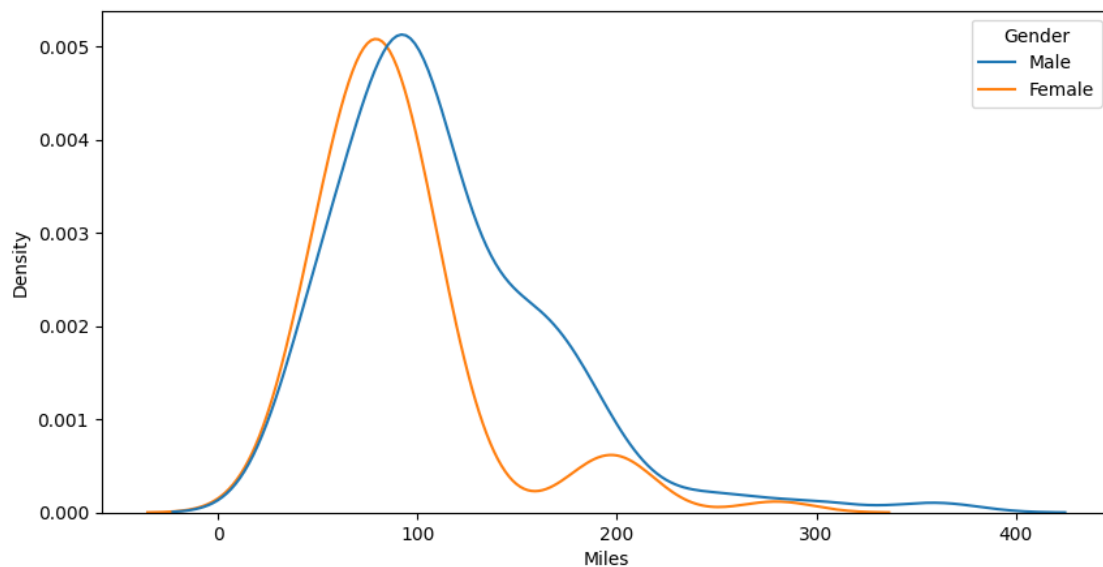
```
[ ]: plt.figure(figsize=(10,5))
sns.kdeplot(data=df, x='Income', hue='Gender')
plt.show()
```



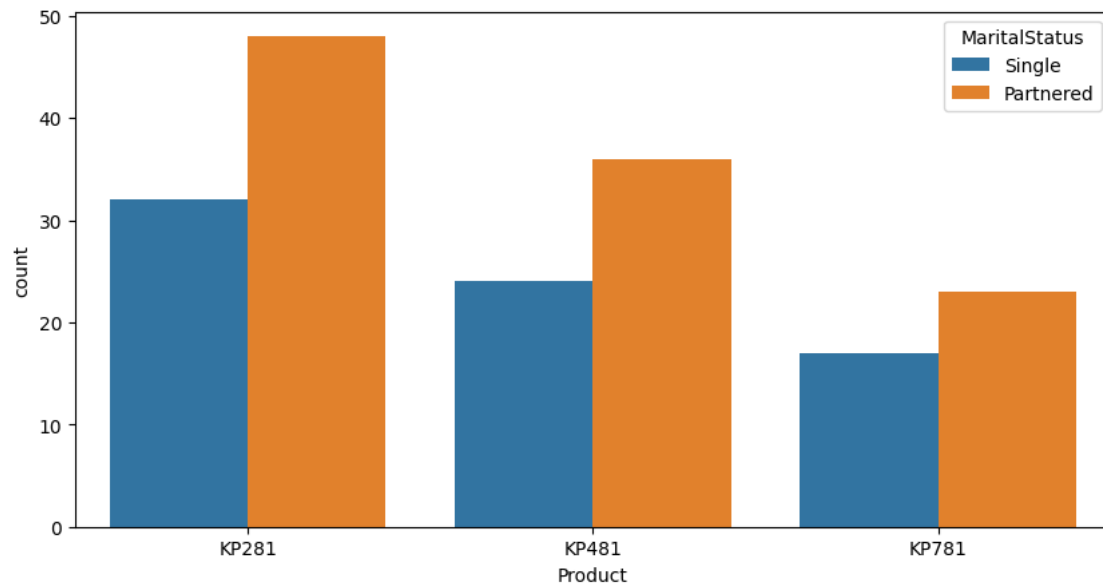
```
[ ]: plt.figure(figsize=(10,5))
sns.countplot(data=df, x='Fitness', hue='Gender')
plt.show()
```



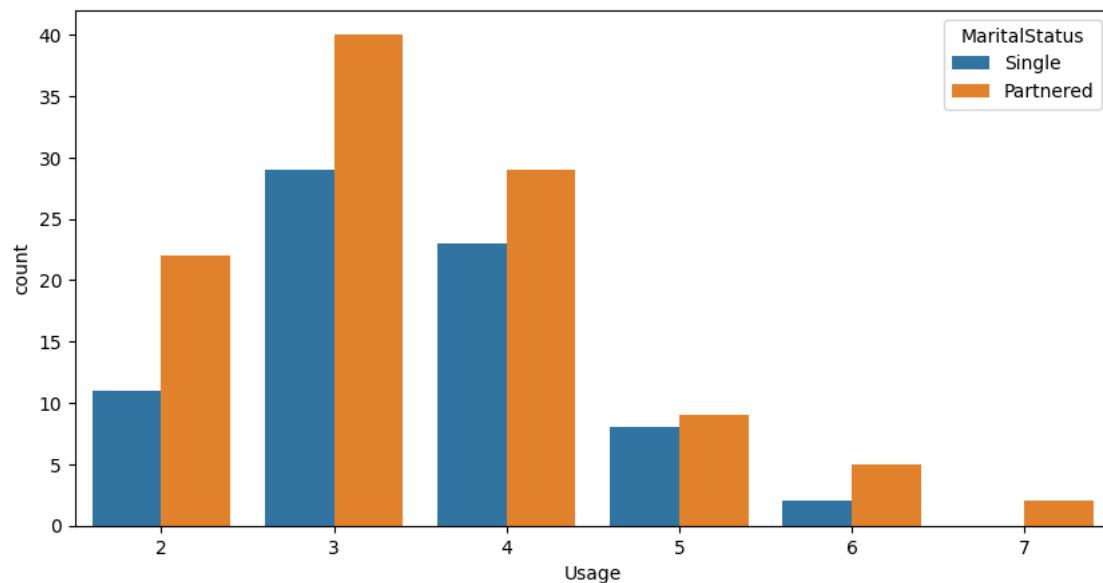
```
[ ]: plt.figure(figsize=(10,5))
sns.kdeplot(data=df, x='Miles', hue='Gender')
plt.show()
```



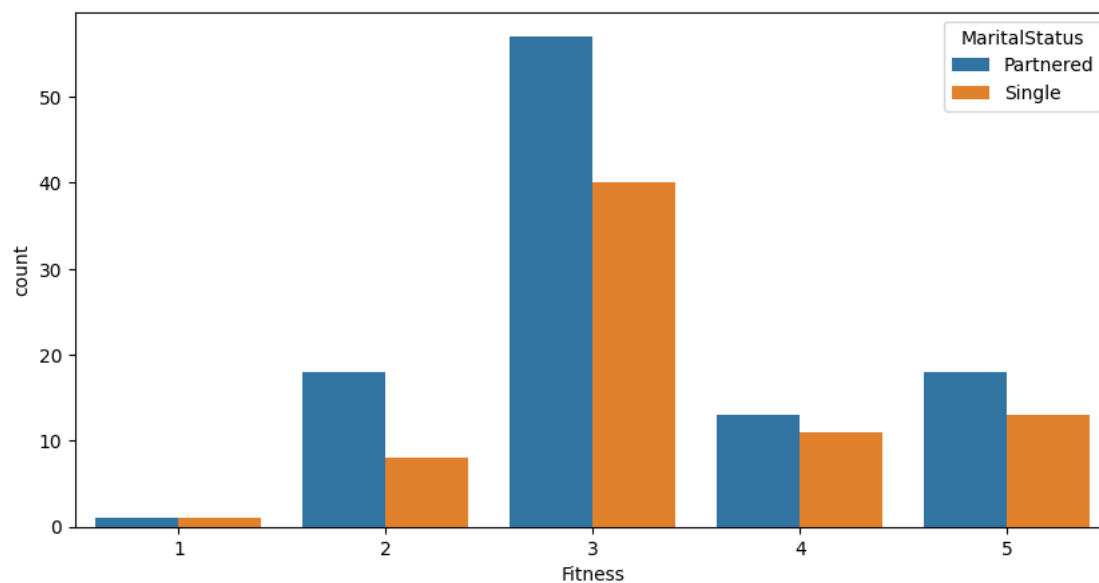
```
[ ]: # Preferred product of Couples and Singles
plt.figure(figsize=(10,5))
sns.countplot(data=df, x="Product", hue='MaritalStatus')
plt.show()
```



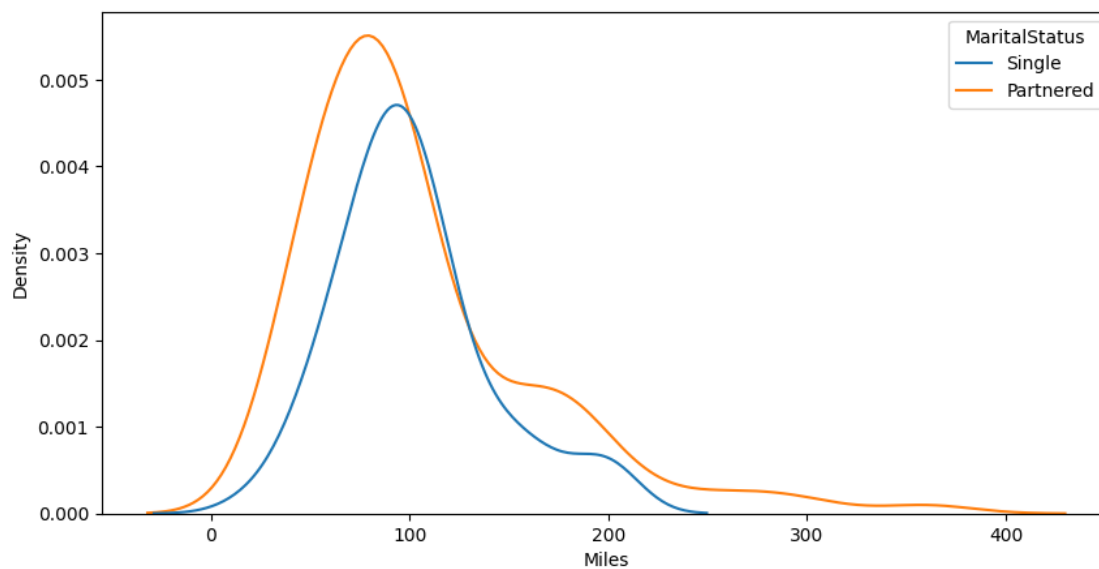
```
[ ]: # Usage of Couples and Singles
plt.figure(figsize=(10,5))
sns.countplot(data=df, x="Usage", hue='MaritalStatus')
plt.show()
```



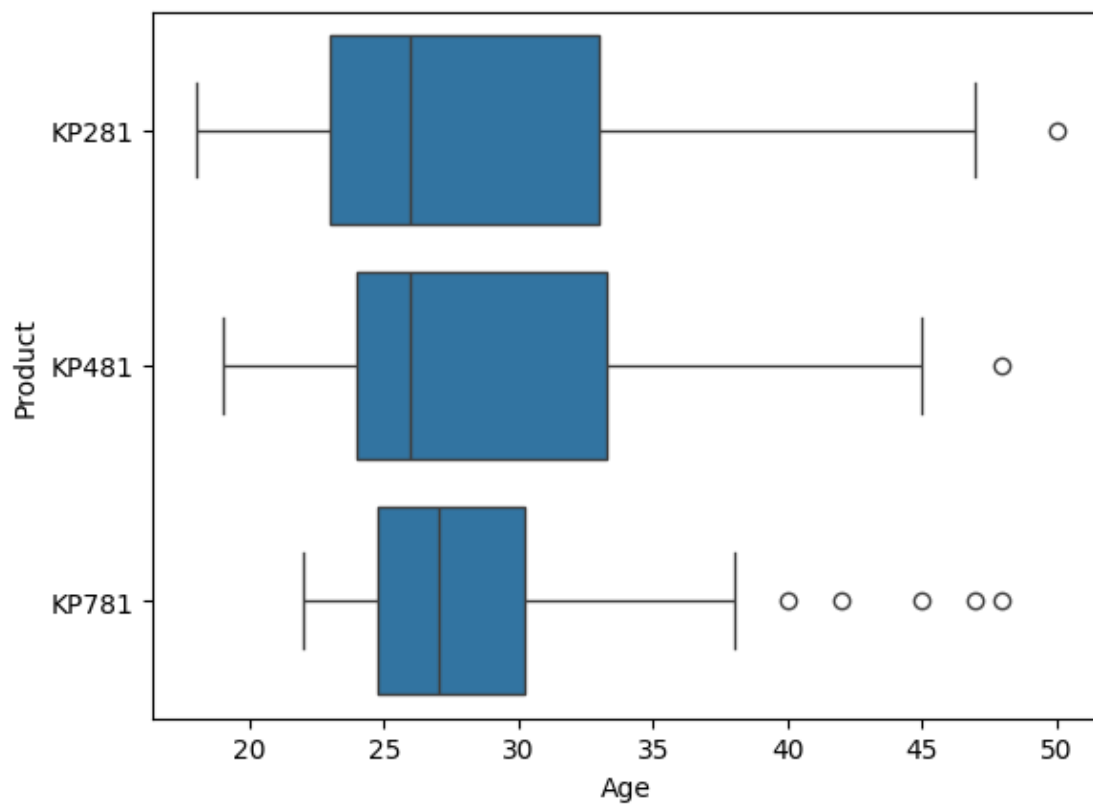
```
[ ]: # Fitness of Couples and Singles
plt.figure(figsize=(10,5))
sns.countplot(data=df, x="Fitness", hue='MaritalStatus')
plt.show()
```



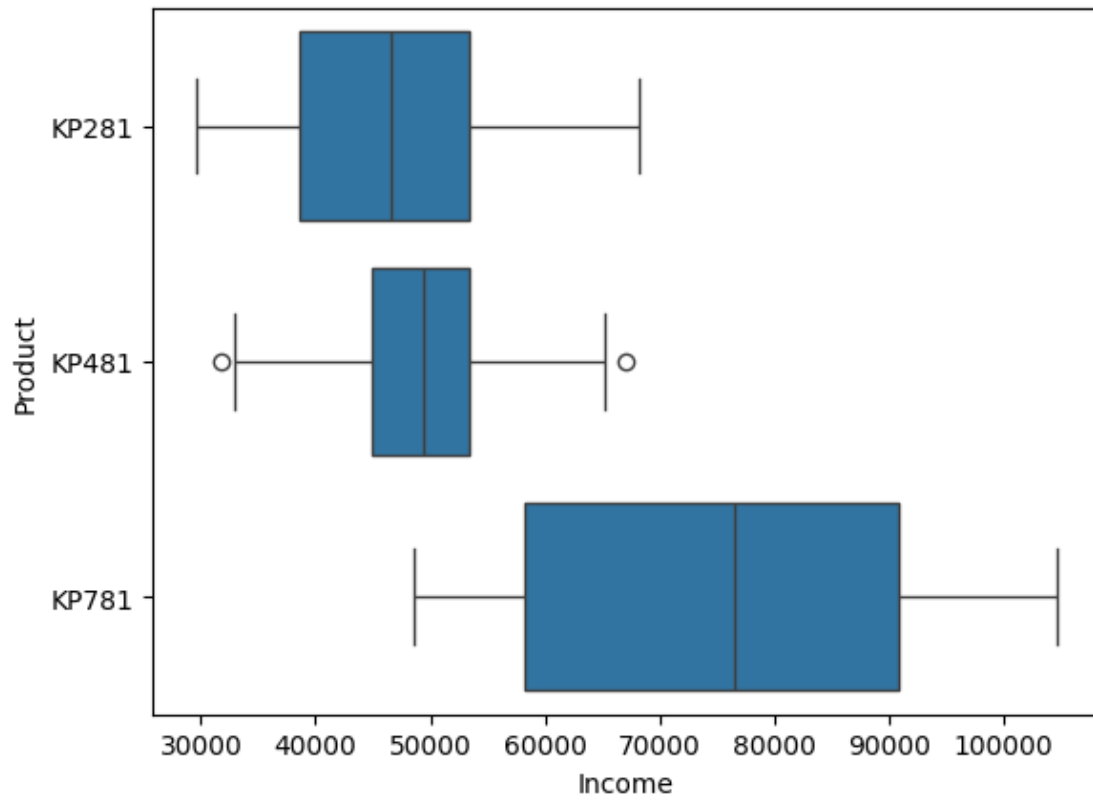
```
[ ]: # Miles (walk/run) of Couples and Singles
plt.figure(figsize=(10,5))
sns.kdeplot(data=df, x="Miles", hue='MaritalStatus')
plt.show()
```



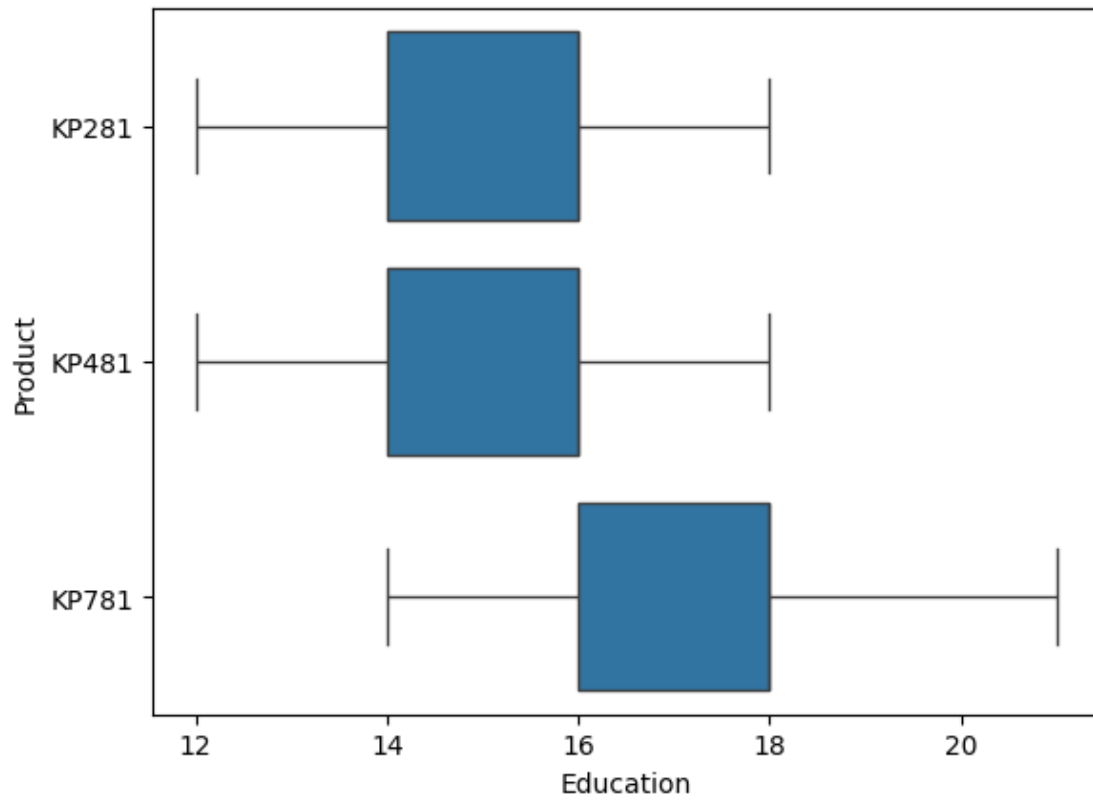

```
[ ]: sns.boxplot(x='Age', y='Product', data=df)
plt.show()
```



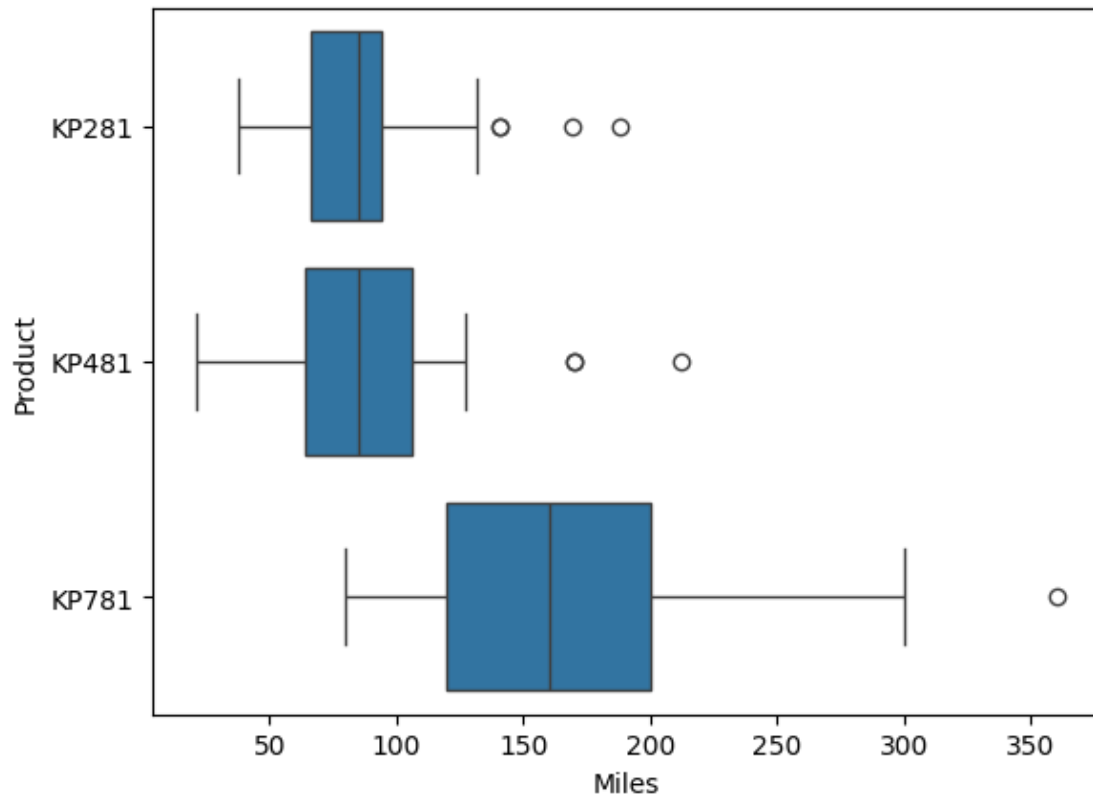
```
[ ]: sns.boxplot(x='Income', y='Product', data=df)
plt.show()
```



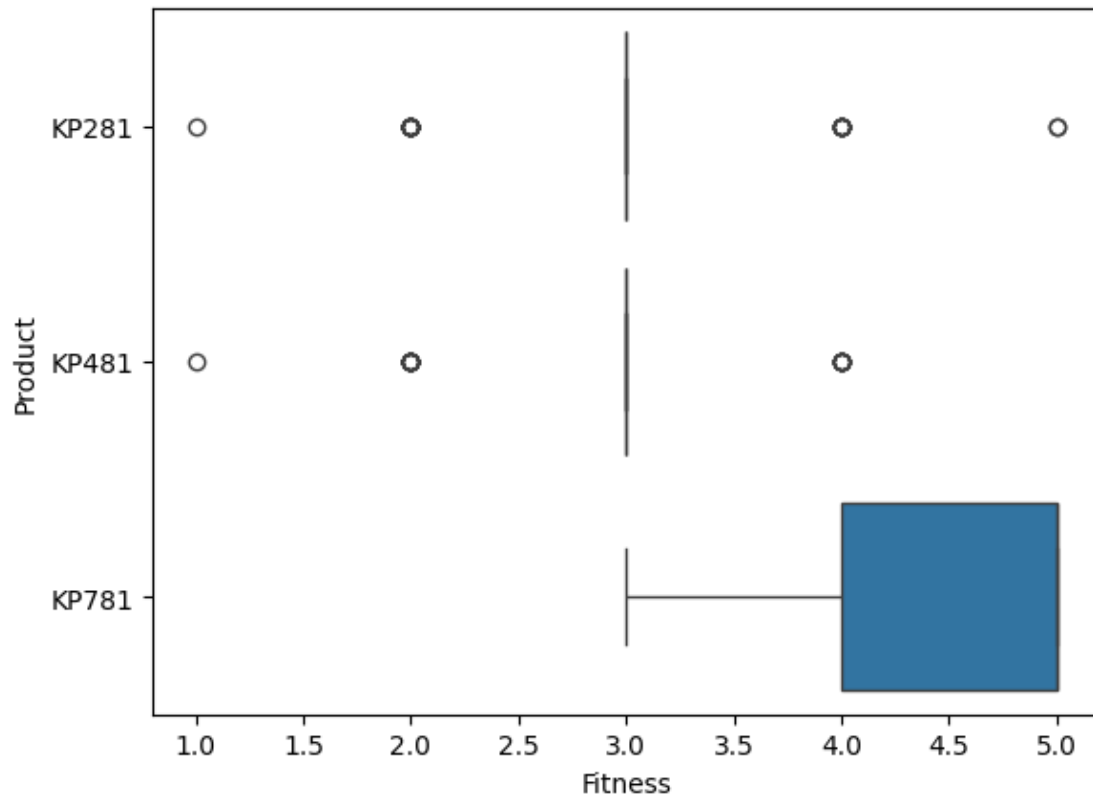
```
[ ]: sns.boxplot(x='Education', y='Product', data=df)
plt.show()
```



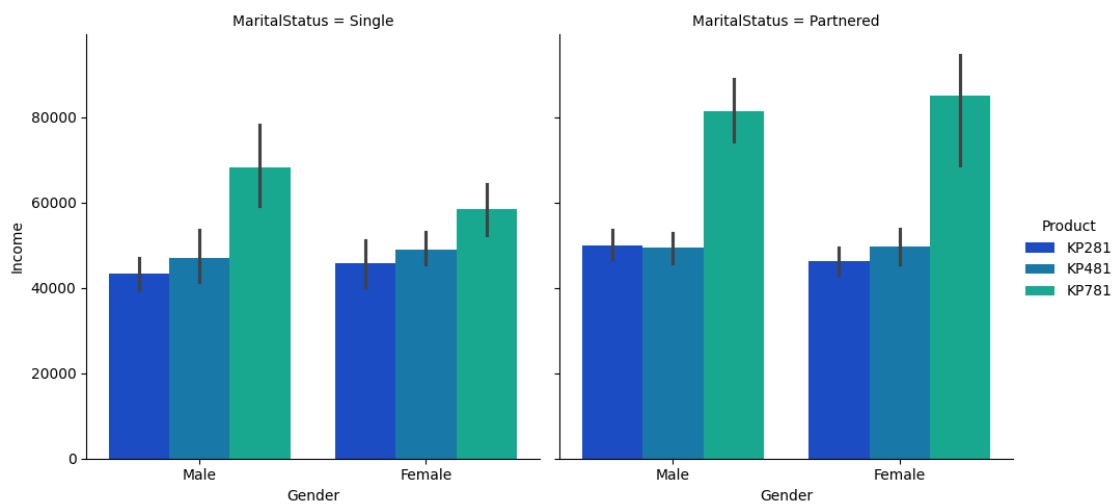
```
[ ]: sns.boxplot(x='Miles', y='Product', data=df)  
plt.show()
```



```
[ ]: sns.boxplot(x='Fitness', y='Product', data=df)
plt.show()
```

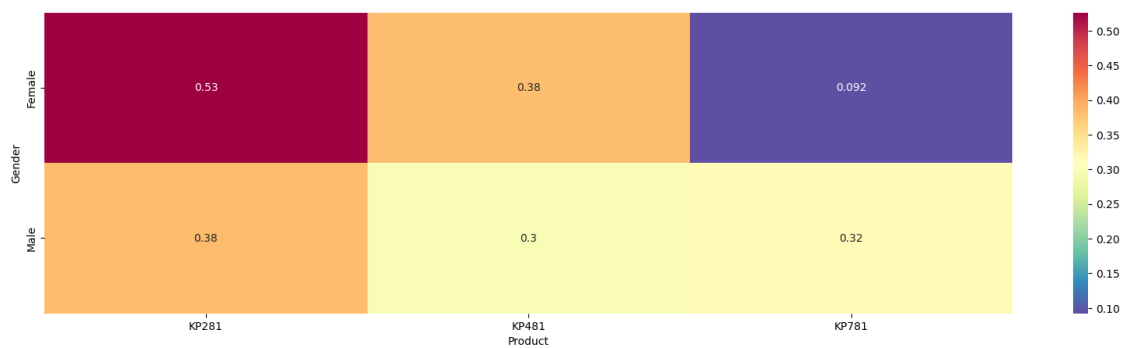


```
[ ]: # Income by gender by product and by marital status
sns.catplot(x='Gender', y='Income', hue='Product', col='MaritalStatus', data=df,
            kind='bar', palette='winter')
plt.show()
```

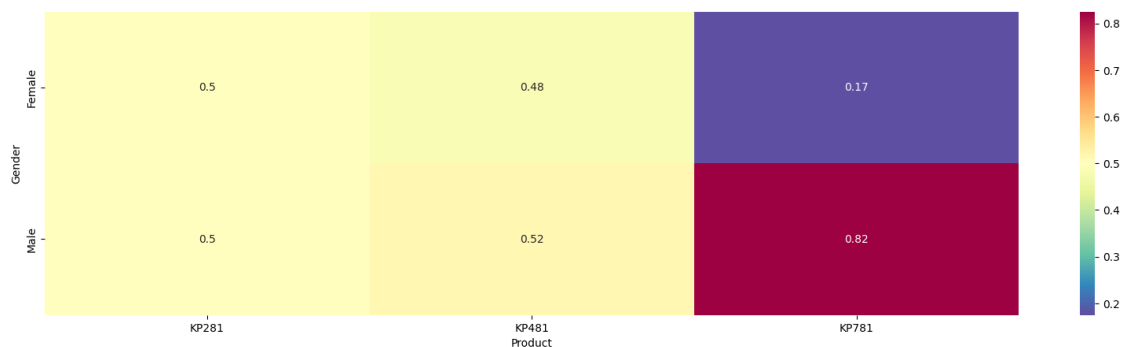


```
[ ]: from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5
```

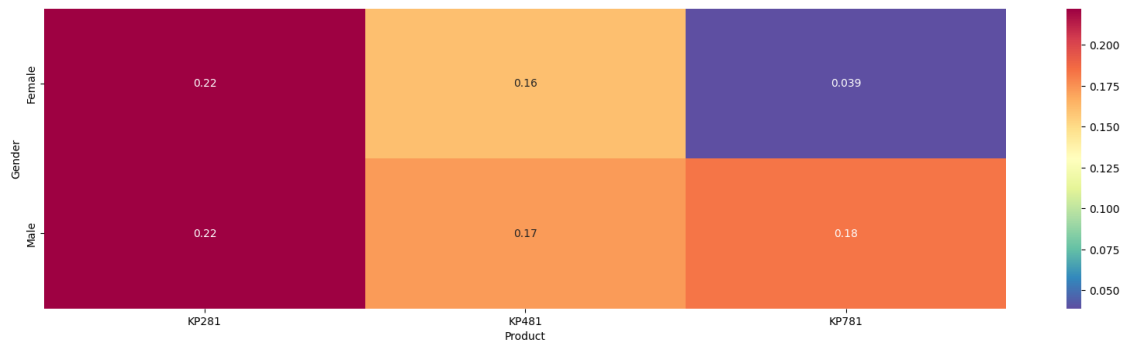
```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize='index'),
            ↪annot=True, cmap='Spectral_r')
plt.show()
```



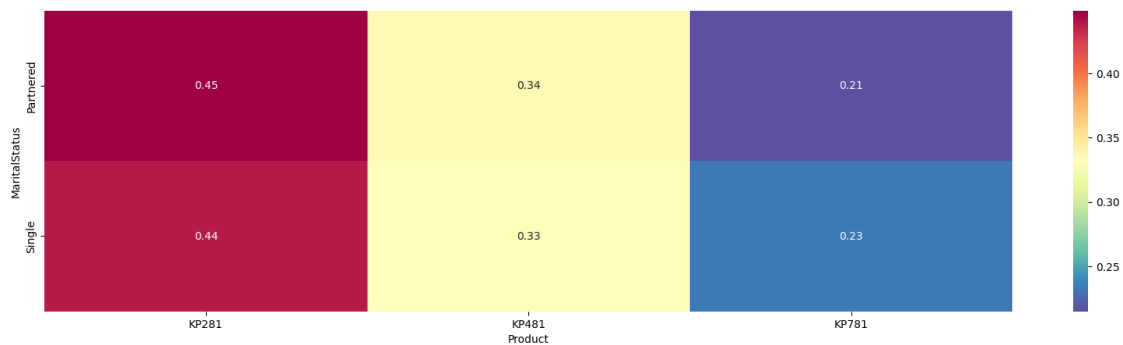
```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize='columns'),
            ↪annot=True, cmap='Spectral_r')
plt.show()
```



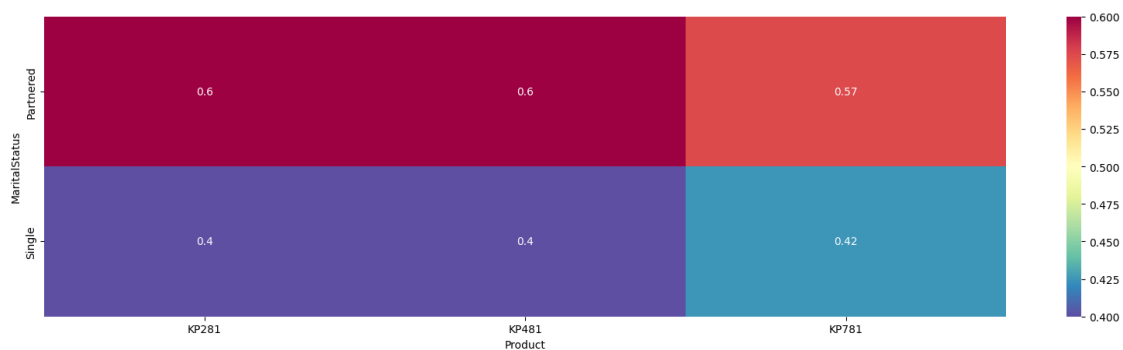
```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize=True),
            ↪annot=True, cmap='Spectral_r')
plt.show()
```



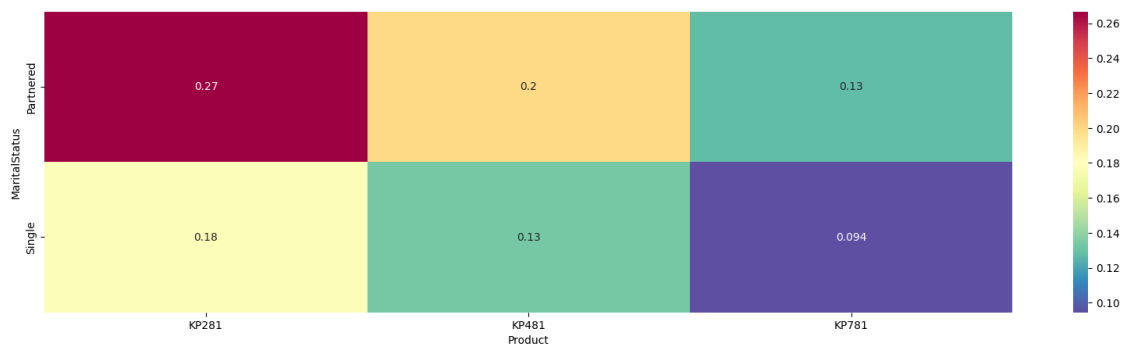
```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'], normalize='index'),
            annot=True, cmap='Spectral_r')
plt.show()
```



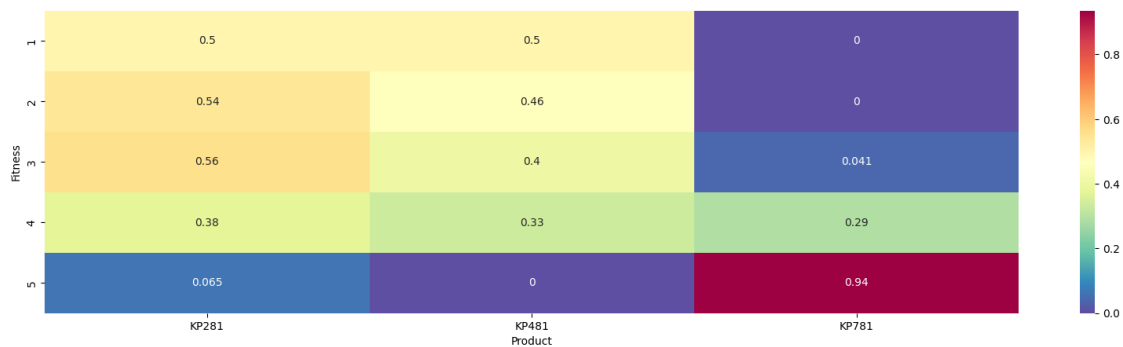
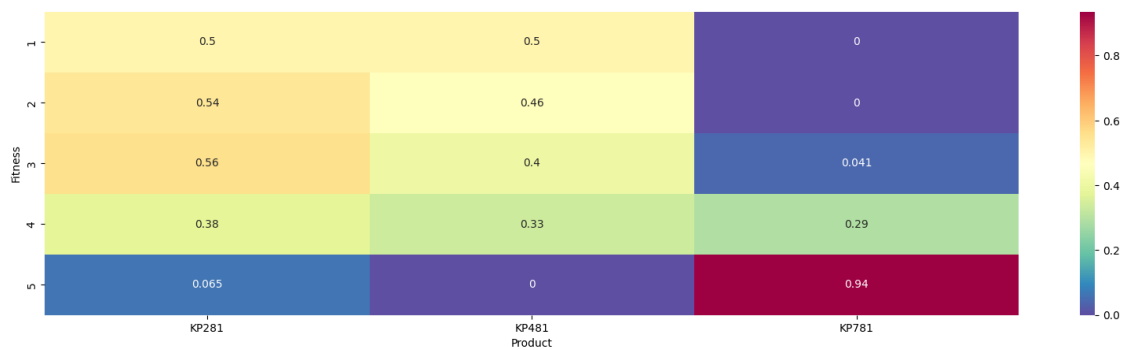
```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'],
            normalize='columns'), annot=True, cmap='Spectral_r')
plt.show()
```



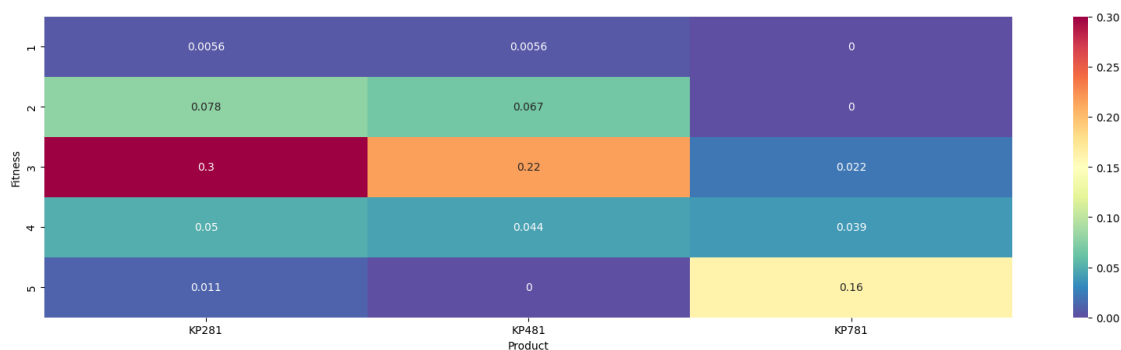
```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'], normalize=True),
            annot=True, cmap='Spectral_r')
plt.show()
```



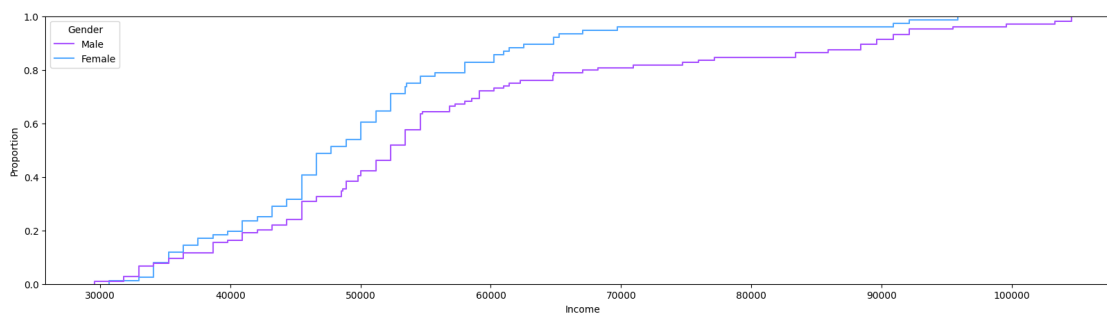
```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['Fitness'], df['Product'], normalize='index'),
            annot=True, cmap='Spectral_r')
plt.show()
```



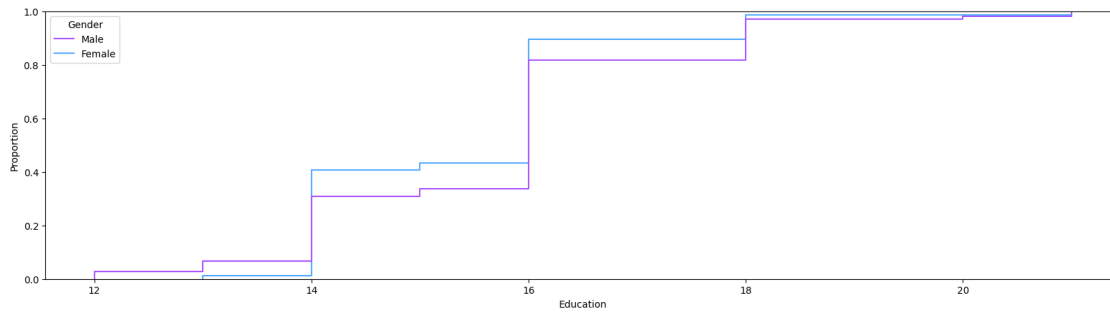

```
[ ]: plt.figure(figsize=(20,5))
sns.heatmap(pd.crosstab(df['Fitness'], df['Product'], normalize=True),
            ↪annot=True, cmap='Spectral_r')
plt.show()
```



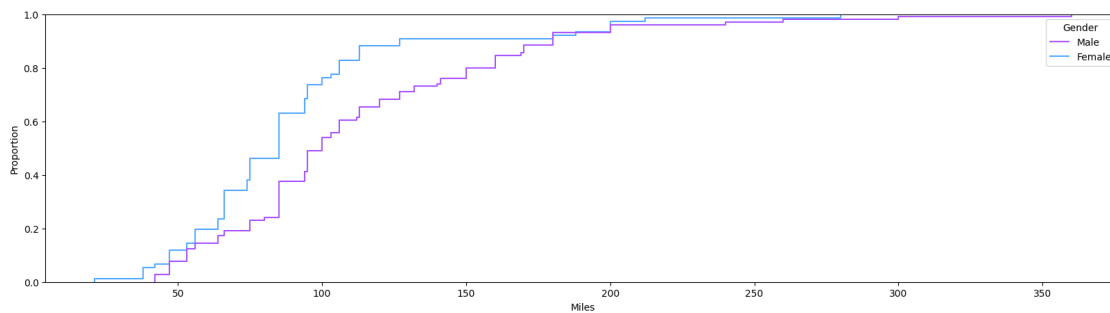
```
[ ]: sns.ecdfplot(data=df, x='Income', complementary=False, palette='cool_r',
                 ↪hue='Gender')
plt.show()
```



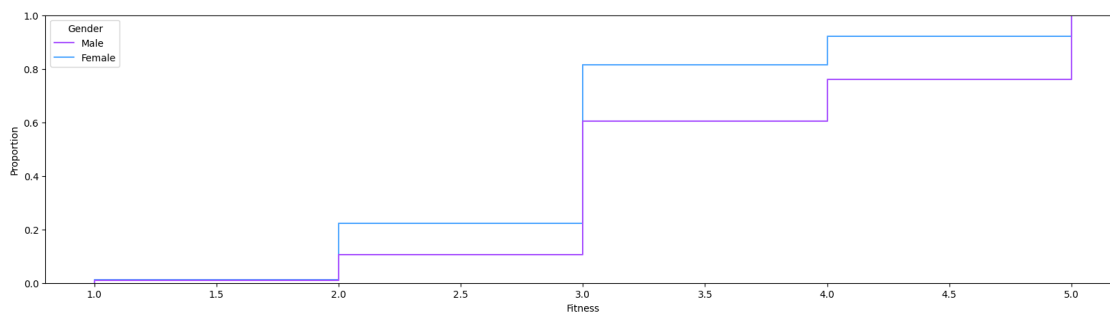
```
[ ]: sns.ecdfplot(data=df, x='Education', complementary=False, palette='cool_r',
                 ↪hue='Gender')
plt.show()
```



```
[ ]: sns.ecdfplot(data=df, x='Miles', complementary=False, palette='cool_r',
    ↪hue='Gender')
plt.show()
```



```
[ ]: sns.ecdfplot(data=df, x='Fitness', complementary=False, palette='cool_r',
    ↪hue='Gender')
plt.show()
```



```
[ ]: sns.ecdfplot(data=df, x='Age', complementary=False, palette='cool_r',
    ↪hue='Gender')
plt.show()
```

