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

In [2]:
```
data=pd.read_csv('aerofit_treadmill.csv')
```

In [3]:
```
data
```

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]:
```
len(data)
```

Out[4]:
```
180
```

In [5]:
```
print('Data type of attributes')
print(data.info())
```

```
Data type of attributes
<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
None
```

In [6]:
```
print('shape of the data',data.shape)
```

```
shape of the data (180, 9)
```

In [7]:
```
print('statistical summary of numerical attributes')
print(data.describe())
```

```
statistical summary of numerical attributes
              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
```

In [8]:
```
Age_counts=data['Age'].value_counts()
print('What are the age distributions purchasing gym equipments ??')
print(" ")
print(Age_counts)
```

```
What are the age distributions purchasing gym equipments ??

25    25
23    18
24    12
26    12
28     9
35     8
33     8
30     7
38     7
21     7
22     7
27     7
31     6
34     6
29     6
20     5
40     5
32     4
19     4
48     2
37     2
45     2
47     2
46     1
50     1
18     1
44     1
43     1
41     1
39     1
36     1
42     1
Name: Age, dtype: int64
```

In [9]:
```
Education_counts=data['Education'].value_counts()
print('Number of people completed their Education in how many years ??')
print(" ")
print(Education_counts)
```

```
Number of people completed their Education in how many years ??

16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: Education, dtype: int64
```

In [10]:
```
Usage_counts=data['Usage'].value_counts()
print('The average number of times the customer plans to use the treadmi
print('')
print(Usage_counts)
```

```
The average number of times the customer plans to use the treadmill eac
h week

3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

In [11]:
```
Fitness_counts=data['Fitness'].value_counts()
print('Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape a
print(' ')
print(Fitness_counts)
```

```
Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 i
s the excellent shape.

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

In [12]:
```
Marital_counts=data['MaritalStatus'].value_counts()
print('Marital Status')
print('')
print(Marital_counts)
```

```
Marital Status

Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

In [13]:
```
Income_counts=data['Income'].value_counts()
print('Number of people and their Annual income')
print('')
print('Number of people at right column and their Annual income on left c
print('')
print(Income_counts)
```

```
Number of people and their Annual income

Number of people at right column and their Annual income on left column

45480    14
52302     9
46617     8
54576     8
53439     8
         ..
65220     1
55713     1
68220     1
30699     1
95508     1
Name: Income, Length: 62, dtype: int64
```

In [14]:
```
Miles_counts=data['Miles'].value_counts()
print('The average number of miles the customer expects to walk/run each
print('')
print(Miles_counts)
```

```
The average number of miles the customer expects to walk/run each week

85     27
95     12
66     10
75     10
47      9
106     9
94      8
113     8
53      7
100     7
180     6
200     6
56      6
64      6
127     5
160     5
42      4
150     4
38      3
74      3
170     3
120     3
103     3
132     2
141     2
280     1
260     1
300     1
240     1
112     1
212     1
80      1
140     1
21      1
169     1
188     1
360     1
Name: Miles, dtype: int64
```

In [15]:
```
data.groupby('Product')['Age'].describe()
```

Out[15]:

Product	count	mean	std	min	25%	50%	75%	max
KP281	80.0	28.55	7.221452	18.0	23.00	26.0	33.00	50.0
KP481	60.0	28.90	6.645248	19.0	24.00	26.0	33.25	48.0
KP781	40.0	29.10	6.971738	22.0	24.75	27.0	30.25	48.0

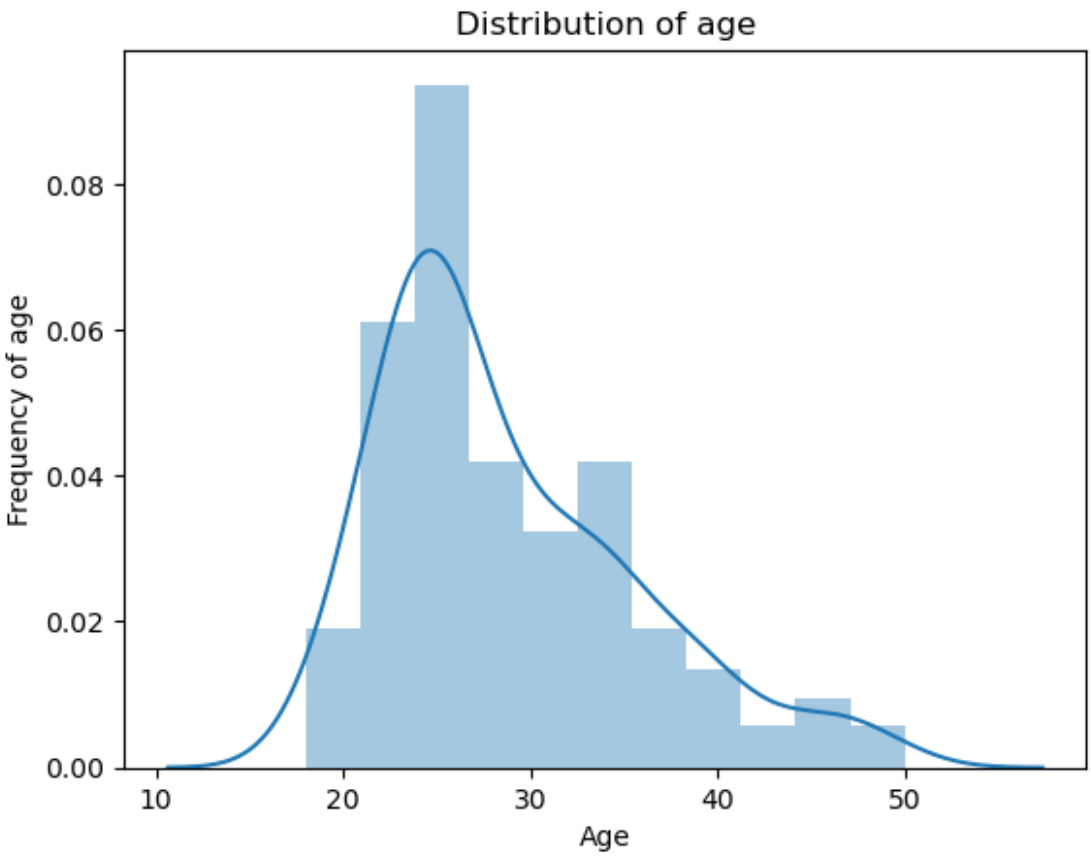
In [16]:
```
data.groupby('Product')['Income'].describe()
```

Out[16]:

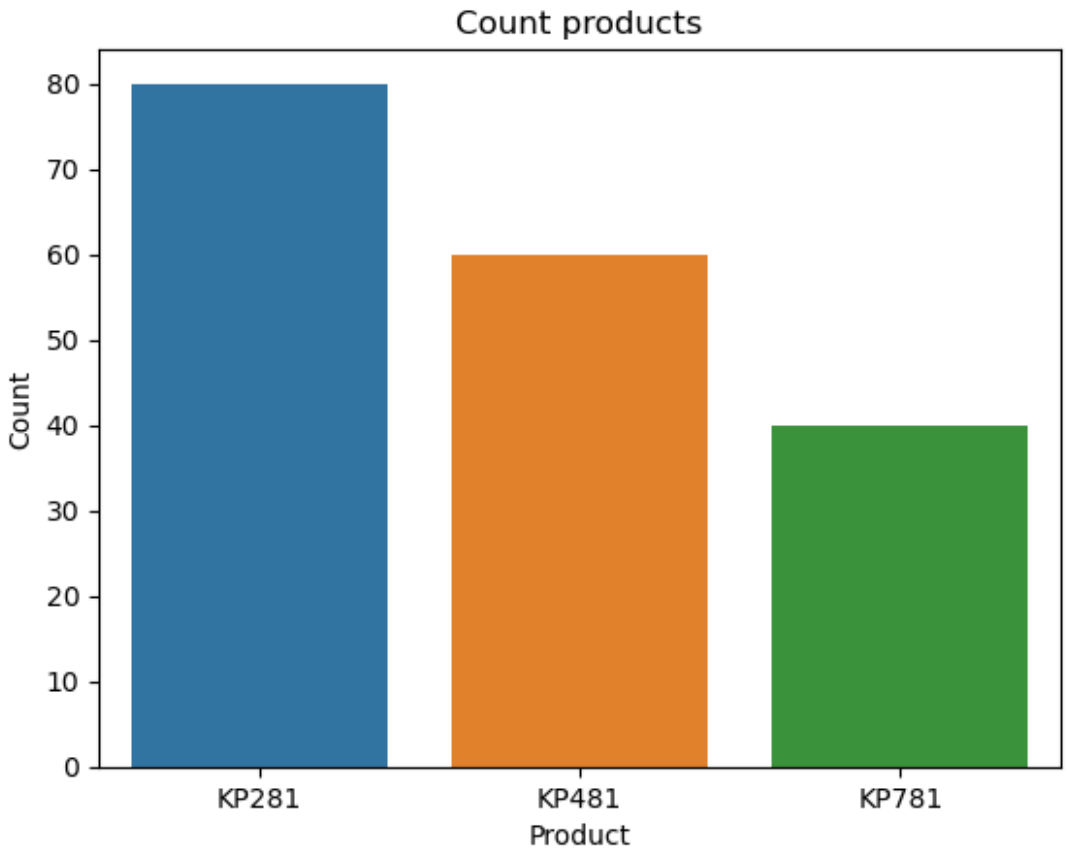
Product	count	mean	std	min	25%	50%	75%	max
KP281	80.0	46418.025	9075.783190	29562.0	38658.00	46617.0	53439.0	68220.0
KP481	60.0	48973.650	8653.989388	31836.0	44911.50	49459.5	53439.0	67083.0
KP781	40.0	75441.575	18505.836720	48556.0	58204.75	76568.5	90886.0	104581.0

In [17]:
```
sns.distplot(data['Age'], kde=True)
plt.title('Distribution of age')
plt.xlabel('Age')
plt.ylabel('Frequency of age')
plt.show()
```

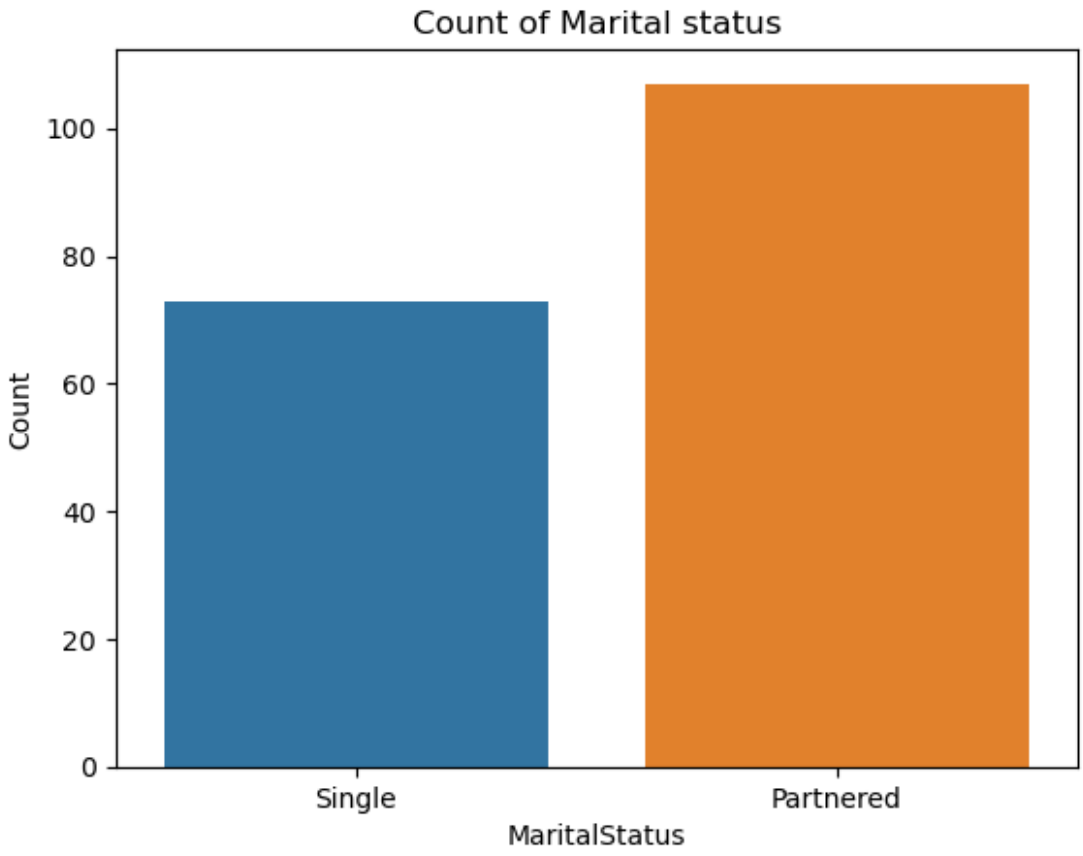
```
/var/folders/fr/xk4r1t0n0kqdxrwjlb8fd4n80000gn/T/ipykernel_7074/7295609
0.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.1
4.0.

Please adapt your code to use either `displot` (a figure-level function
with
similar flexibility) or `histplot` (an axes-level function for histogra
ms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data['Age'], kde=True)
```



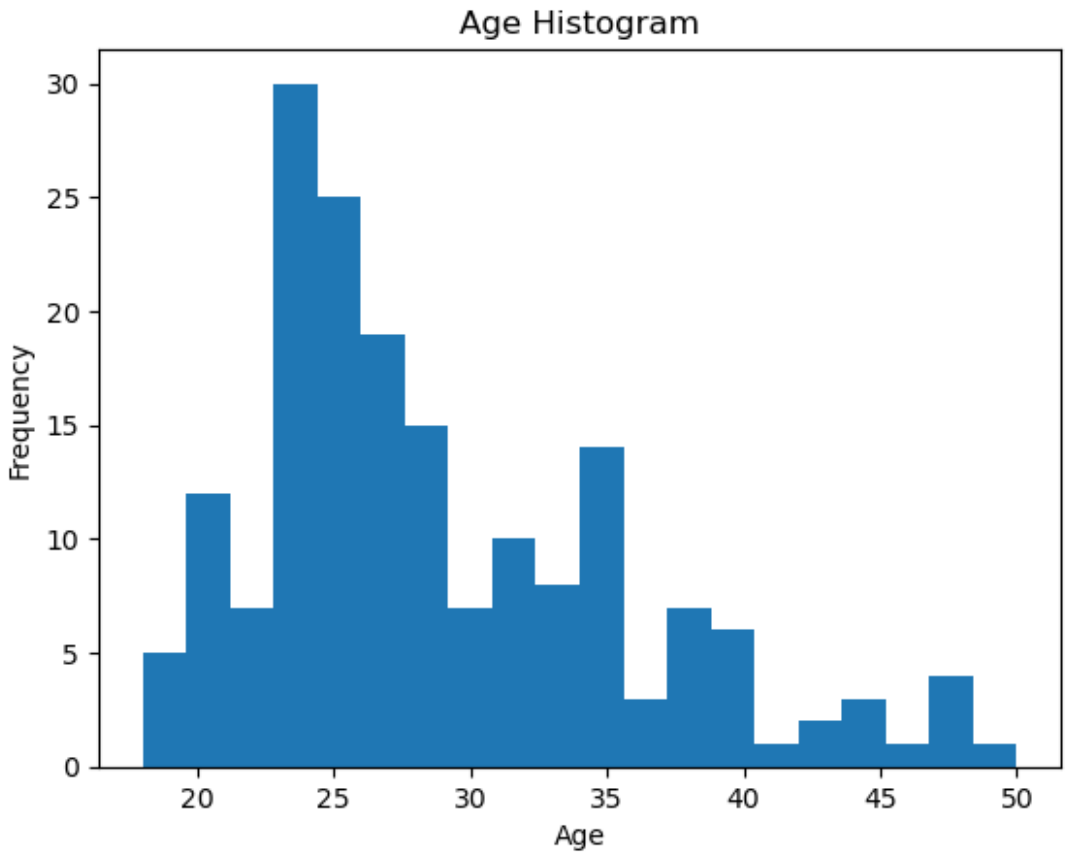
In [18]:
```
sns.countplot(x='Product',data=data)
plt.title('Count products')
plt.xlabel('Product')
plt.ylabel('Count')
plt.show()
```



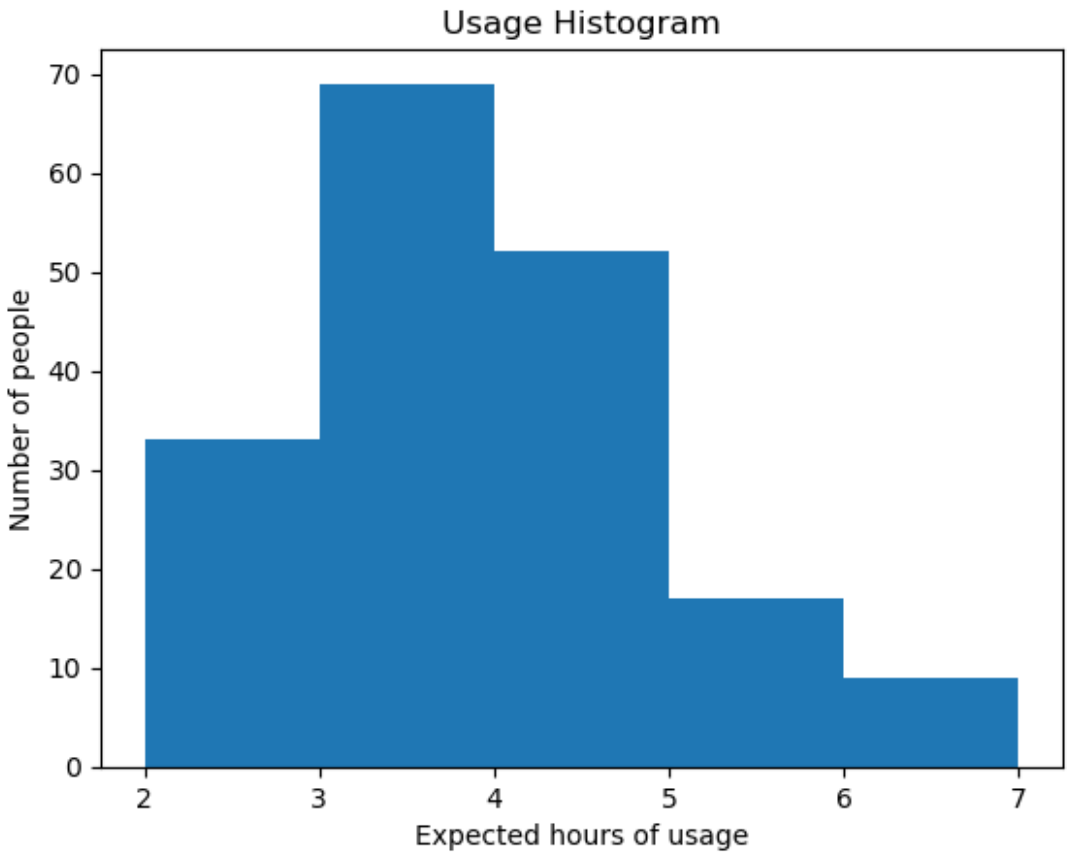
In [19]:
```
sns.countplot(x='MaritalStatus',data=data)
plt.title('Count of Marital status')
plt.xlabel('MaritalStatus')
plt.ylabel('Count')
plt.show()
```



In [20]:
```
plt.hist(data['Age'],bins=20)
plt.title('Age Histogram')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



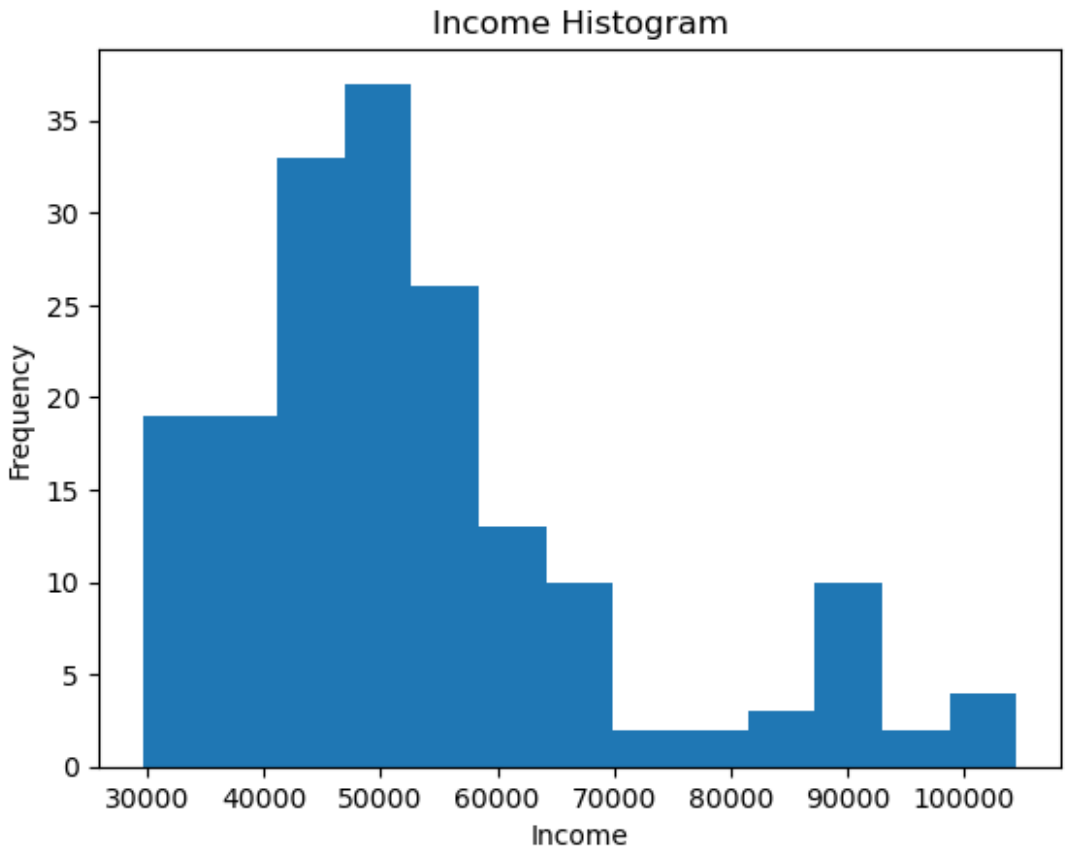
In [21]:
```
plt.hist(data['Usage'],bins=5)
plt.title('Usage Histogram')
plt.xlabel('Expected hours of usage')
plt.ylabel('Number of people')
plt.show()
```



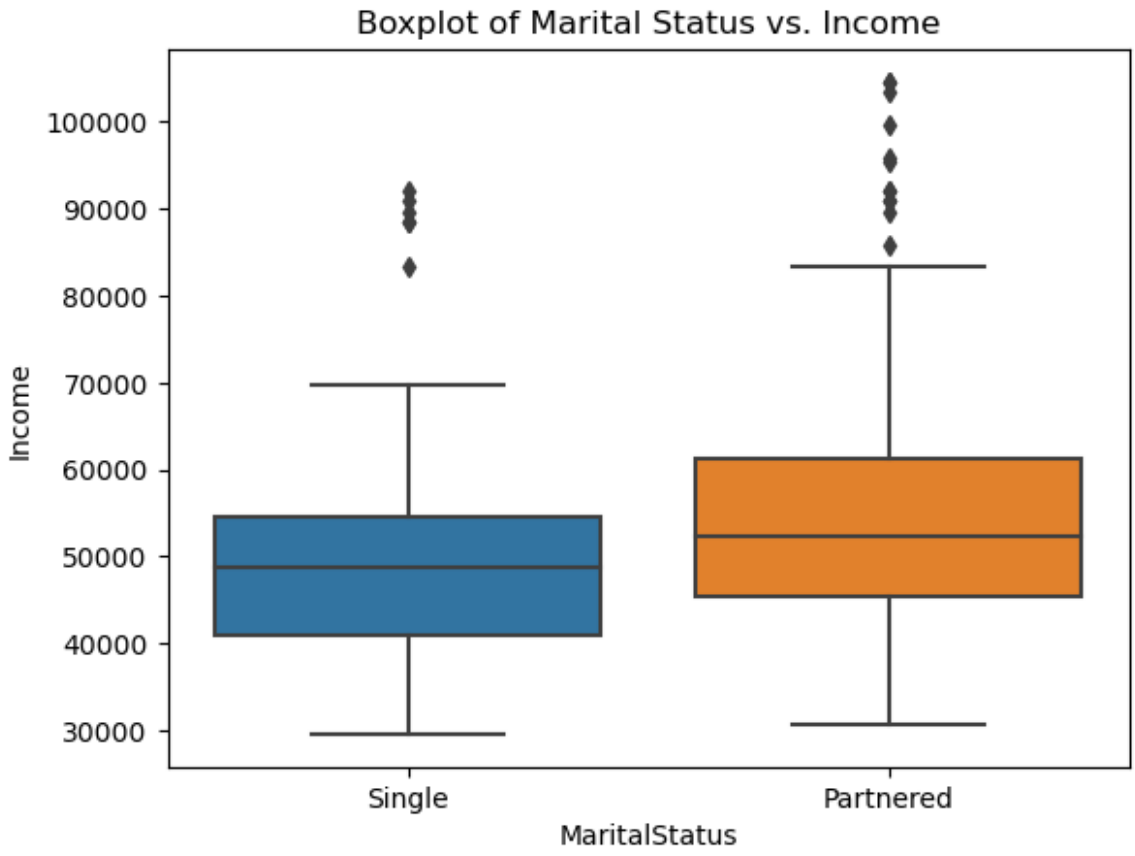
In [22]:
```
num_data_points=len(data['Income'])
binsize=int(np.sqrt(num_data_points))
binsize
```

Out[22]: 13

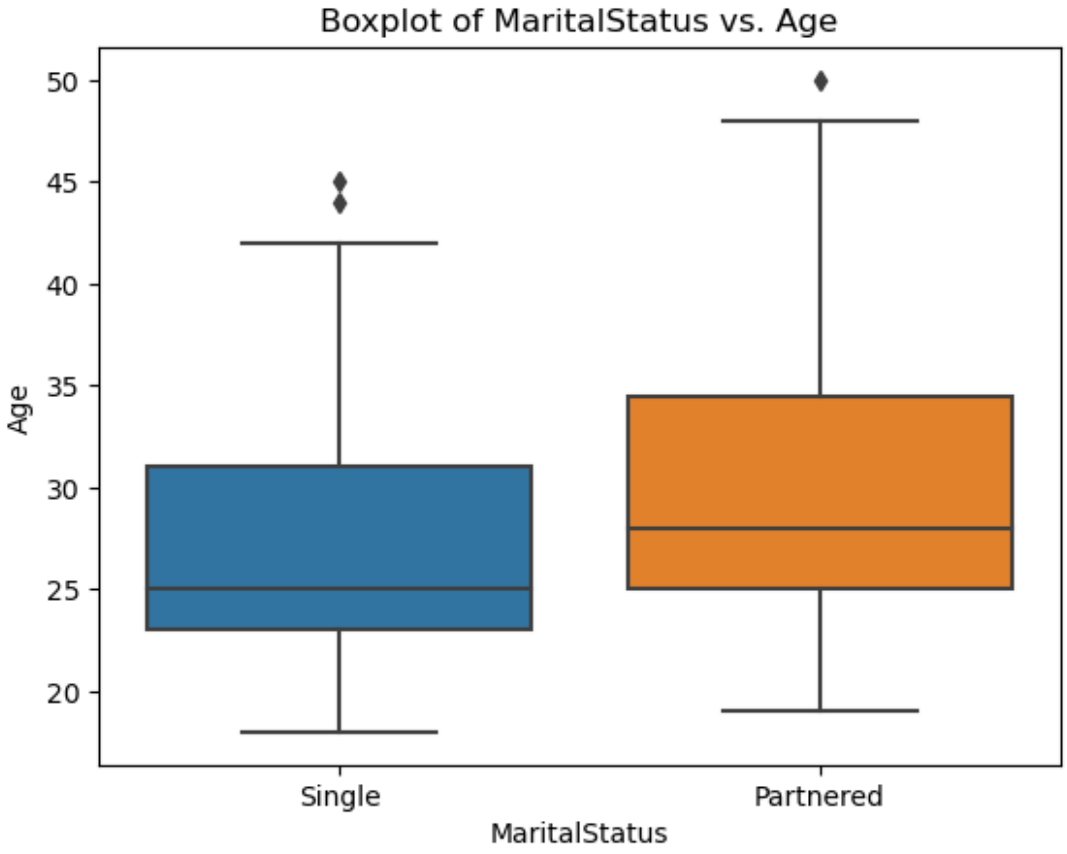
In [23]:
```
plt.hist(data['Income'],bins=13)
plt.title('Income Histogram')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.show()
```



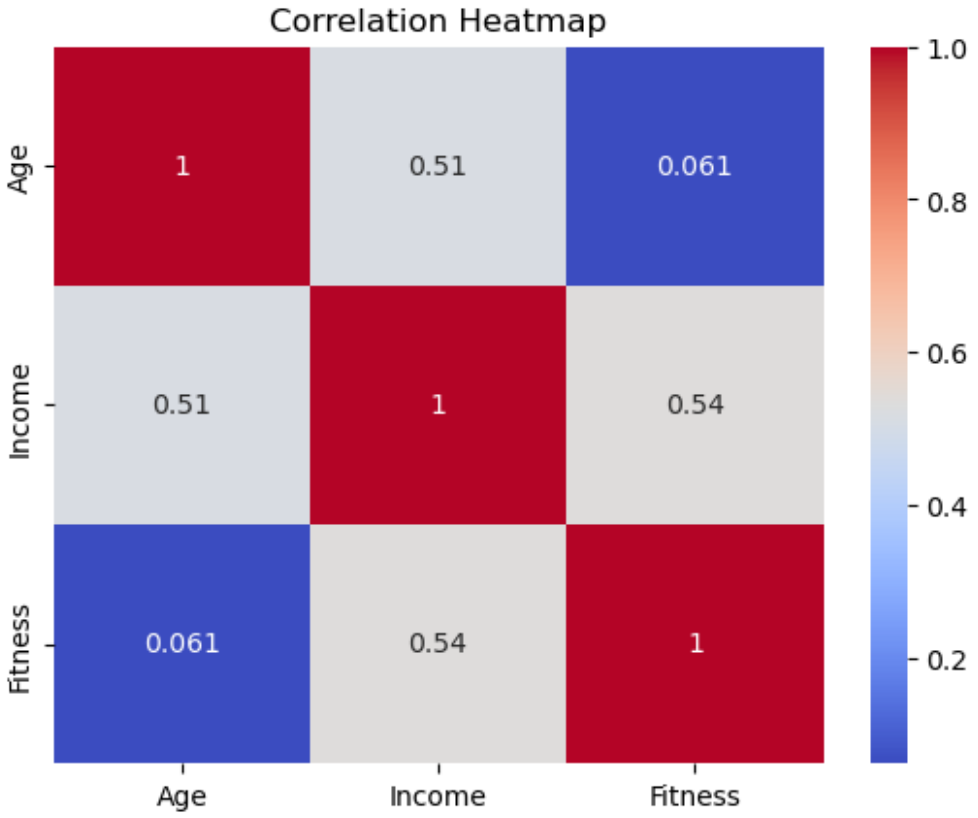
In [24]:
```
sns.boxplot(x='MaritalStatus', y='Income', data=data)
plt.title('Boxplot of Marital Status vs. Income')
plt.xlabel('MaritalStatus')
plt.ylabel('Income')
plt.show()
```



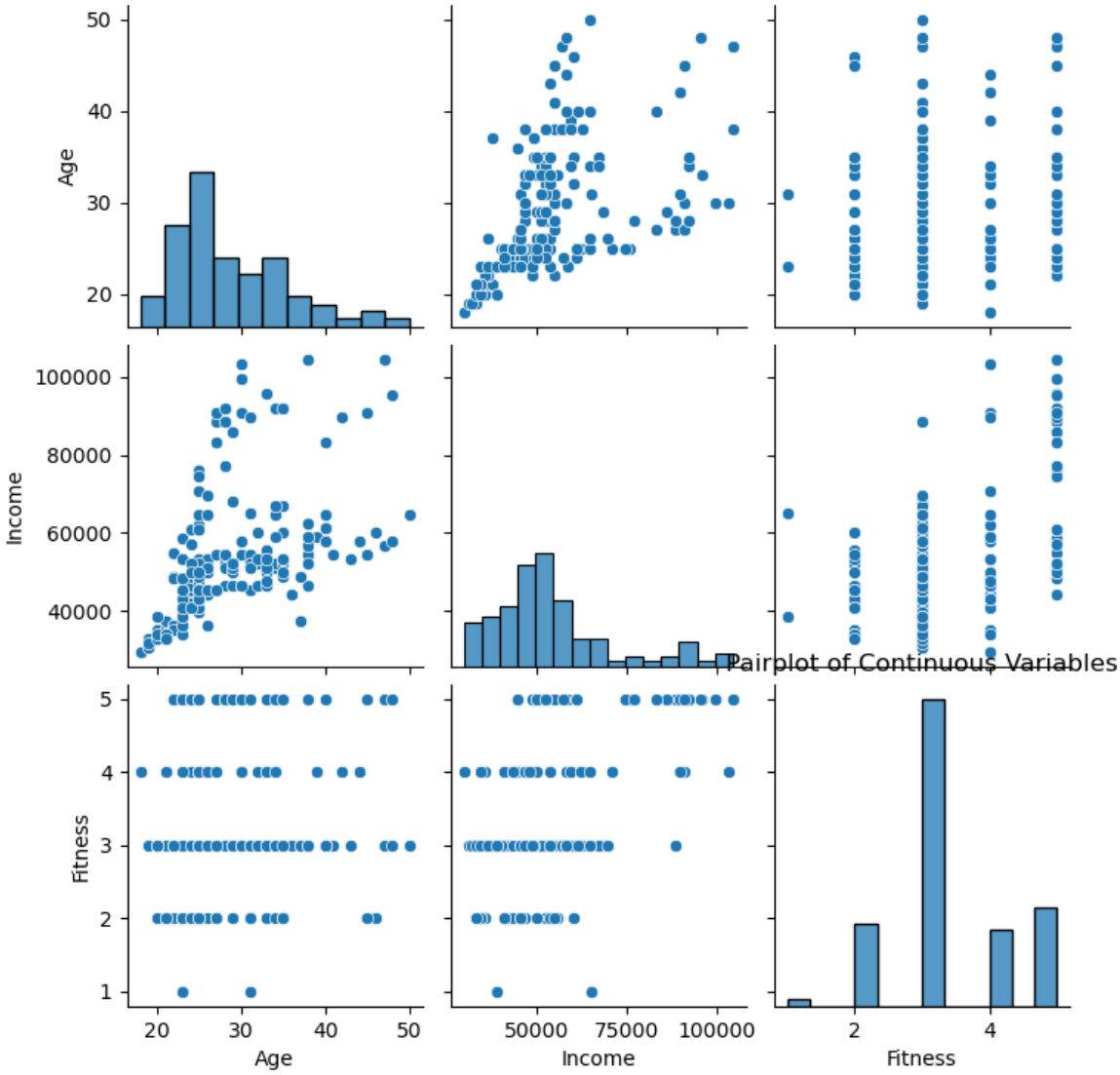
In [25]:
```
sns.boxplot(x='MaritalStatus',y='Age',data=data)
plt.title('Boxplot of MaritalStatus vs. Age')
plt.xlabel('MaritalStatus')
plt.ylabel('Age')
plt.show()
```



In [26]:
```
correlation_matrix = data[['Age', 'Income', 'Fitness']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



In [29]:
```
sns.pairplot(data[['Age', 'Income', 'Fitness']])
plt.title('Pairplot of Continuous Variables')
plt.show()
```



In [27]:
```
gender_by_product=pd.crosstab(data['Product'],data['Gender'])
gender_by_product
```

Out[27]:

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

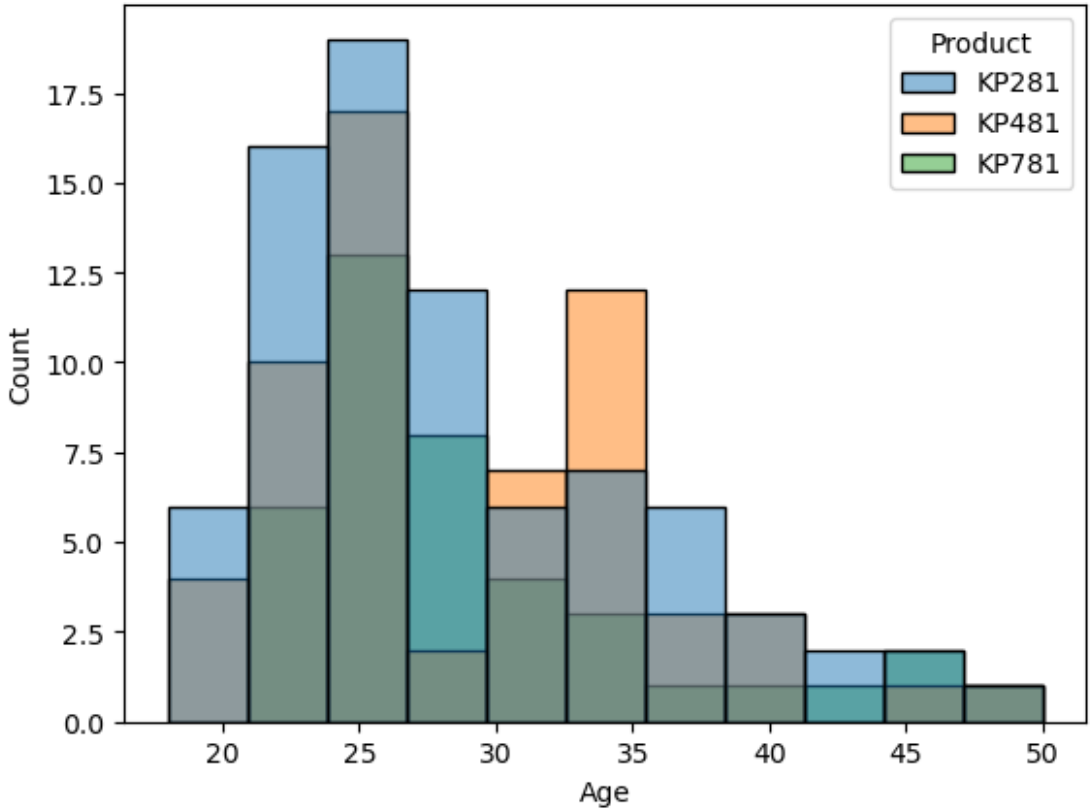
In [28]:
```
gender_by_product=pd.crosstab(data['Product'],data['MaritalStatus'])
gender_by_product
```

Out[28]:

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

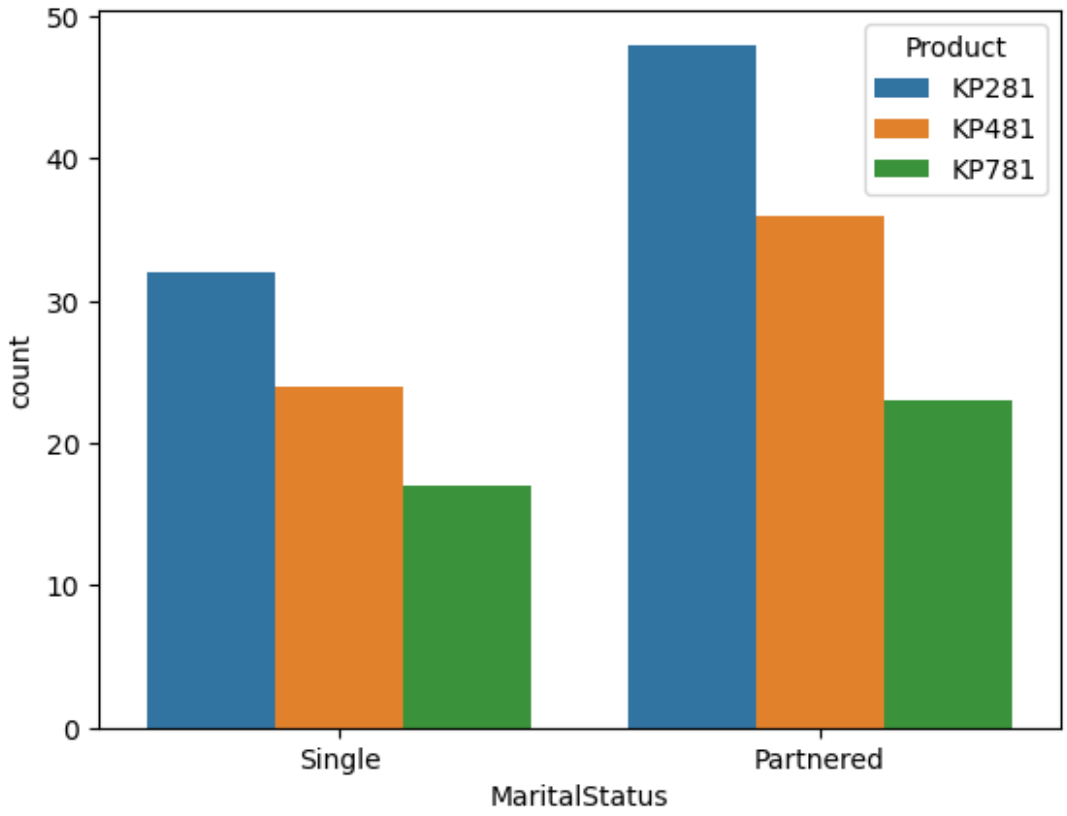
In [29]:
```
sns.histplot(data=data, x='Age', hue='Product', kde=False)
```

Out[29]: <Axes: xlabel='Age', ylabel='Count'>



In [30]:
```
sns.countplot(x='MaritalStatus', hue='Product', data=data)
```

Out[30]:
```
<Axes: xlabel='MaritalStatus', ylabel='count'>
```



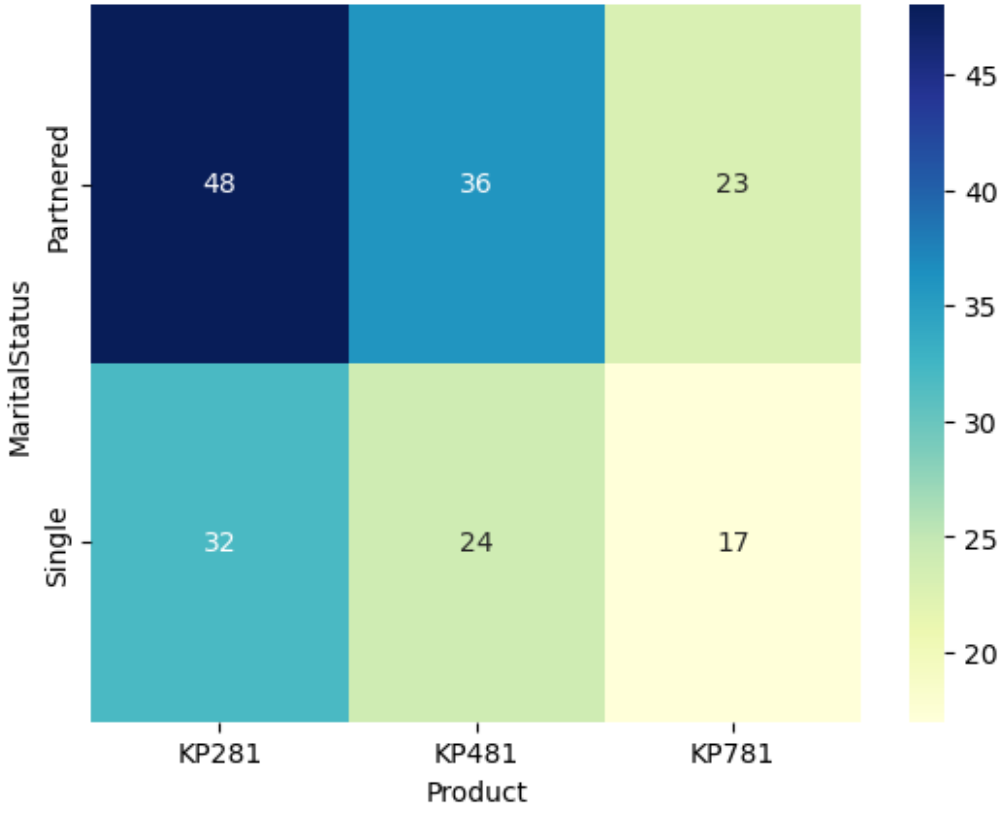
In [31]:
```
marital_product_crosstab = pd.crosstab(data['MaritalStatus'], data['Prod
marital_product_crosstab
```

Out[31]:

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

In [32]:
```
sns.heatmap(marital_product_crosstab, annot=True, cmap='YlGnBu', fmt='d'
```

Out[32]:
```
<Axes: xlabel='Product', ylabel='MaritalStatus'>
```



Conditional Probability

what is the probability of Male Customers buying KP781 ??

In [33]:
```
kp781_data = data[data['Product'] == 'KP781']
kp781_data
```

Out[33]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
140	KP781	22	Male	14	Single	4	3	48658	106
141	KP781	22	Male	16	Single	3	5	54781	120
142	KP781	22	Male	18	Single	4	5	48556	200
143	KP781	23	Male	16	Single	4	5	58516	140
144	KP781	23	Female	18	Single	5	4	53536	100
145	KP781	23	Male	16	Single	4	5	48556	100
146	KP781	24	Male	16	Single	4	5	61006	100
147	KP781	24	Male	18	Partnered	4	5	57271	80
148	KP781	24	Female	16	Single	5	5	52291	200
149	KP781	24	Male	16	Single	5	5	49801	160
150	KP781	25	Male	16	Partnered	4	5	49801	120
151	KP781	25	Male	16	Partnered	4	4	62251	160
152	KP781	25	Female	18	Partnered	5	5	61006	200
153	KP781	25	Male	18	Partnered	4	3	64741	100
154	KP781	25	Male	18	Partnered	6	4	70966	180
155	KP781	25	Male	18	Partnered	6	5	75946	240
156	KP781	25	Male	20	Partnered	4	5	74701	170
157	KP781	26	Female	21	Single	4	3	69721	100
158	KP781	26	Male	16	Partnered	5	4	64741	180
159	KP781	27	Male	16	Partnered	4	5	83416	160
160	KP781	27	Male	18	Single	4	3	88396	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
162	KP781	28	Female	18	Partnered	6	5	92131	180
163	KP781	28	Male	18	Partnered	7	5	77191	180
164	KP781	28	Male	18	Single	6	5	88396	150
165	KP781	29	Male	18	Single	5	5	52290	180
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
168	KP781	30	Male	18	Partnered	5	4	103336	160
169	KP781	30	Male	18	Partnered	5	5	99601	150
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
172	KP781	34	Male	16	Single	5	5	92131	150
173	KP781	35	Male	16	Partnered	4	5	92131	360
174	KP781	38	Male	18	Partnered	5	5	104581	150
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

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

In [34]:
```
total_male_customers = len(data[data['Gender'] == 'Male'])
total_male_customers
```

Out[34]: 104

In [35]:
```
male_customers_kp781 = len(kp781_data[kp781_data['Gender'] == 'Male'])
male_customers_kp781
```

Out[35]: 33

In [36]:
```
probability_male_kp781 = (male_customers_kp781 / total_male_customers)*1(
probability_male_kp781
```

Out[36]: 31.73076923076923

There is 31.7 % probability that Male Customer will purchase KP781

In [37]:
```
total_Female_customers = len(data[data['Gender'] == 'Female'])
total_Female_customers
```

Out[37]: 76

In [38]:
```
kp281_data = data[data['Product'] == 'KP281']
kp281_data
```

Out[38]:

	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
...	...	...	...	...	...	...	...	...	...
75	KP281	43	Male	16	Partnered	3	3	53439	66
76	KP281	44	Female	16	Single	3	4	57987	75
77	KP281	46	Female	16	Partnered	3	2	60261	47
78	KP281	47	Male	16	Partnered	4	3	56850	94
79	KP281	50	Female	16	Partnered	3	3	64809	66

80 rows × 9 columns

In [39]:
```
Female_customers_kp281 = len(kp281_data[kp281_data['Gender'] == 'Female'
Female_customers_kp281
```

Out[39]: 40

In [40]:
```
probability_Female_kp281 = (Female_customers_kp281 / total_Female_custome
probability_Female_kp281
```

Out[40]:
```
52.63157894736842
```

There is 52.6 % probability that Female Customers will purchase KP281

In [46]:
```
total_male_customers=len(data[data['Gender']=="Male"])
total_male_customers
```

Out[46]:
```
104
```

In [42]:
```
kp481_data = data[data['Product'] == 'KP481']
kp481_data
```

Out[42]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
80	KP481	19	Male	14	Single	3	3	31836	64
81	KP481	20	Male	14	Single	2	3	32973	53
82	KP481	20	Female	14	Partnered	3	3	34110	106
83	KP481	20	Male	14	Single	3	3	38658	95
84	KP481	21	Female	14	Partnered	5	4	34110	212
85	KP481	21	Male	16	Partnered	2	2	34110	42
86	KP481	21	Male	12	Partnered	2	2	32973	53
87	KP481	23	Male	14	Partnered	3	3	36384	95
88	KP481	23	Male	14	Partnered	3	3	38658	85
89	KP481	23	Female	16	Single	3	3	45480	95
90	KP481	23	Male	16	Partnered	4	3	45480	127
91	KP481	23	Female	16	Partnered	3	2	43206	74
92	KP481	23	Female	14	Single	3	2	40932	53
93	KP481	23	Male	16	Partnered	3	3	45480	64
94	KP481	24	Female	14	Single	3	2	40932	85
95	KP481	24	Male	14	Single	3	4	48891	106
96	KP481	24	Female	16	Single	3	3	50028	106
97	KP481	25	Female	14	Partnered	2	3	45480	85
98	KP481	25	Female	14	Single	3	4	43206	127
99	KP481	25	Male	16	Partnered	2	2	52302	42
100	KP481	25	Female	14	Partnered	5	3	47754	106
101	KP481	25	Male	14	Single	3	3	45480	95
102	KP481	25	Female	14	Single	2	3	43206	64
103	KP481	25	Male	14	Partnered	4	3	45480	170
104	KP481	25	Male	14	Partnered	3	4	43206	106
105	KP481	25	Male	16	Partnered	2	3	50028	53
106	KP481	25	Female	14	Single	2	2	45480	42
107	KP481	25	Male	14	Single	4	3	48891	127
108	KP481	26	Female	16	Partnered	4	3	45480	85
109	KP481	26	Female	16	Single	4	4	50028	127
110	KP481	26	Male	16	Single	4	3	51165	106
111	KP481	27	Male	14	Single	4	2	45480	53
112	KP481	29	Female	14	Partnered	3	3	51165	95
113	KP481	30	Female	14	Single	3	3	57987	74
114	KP481	30	Female	13	Single	4	3	46617	106
115	KP481	31	Male	16	Partnered	3	3	52302	95
116	KP481	31	Female	16	Partnered	2	3	51165	64
117	KP481	31	Female	18	Single	2	1	65220	21

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
118	KP481	32	Male	16	Single	4	3	60261	127
119	KP481	32	Male	16	Partnered	3	3	53439	95
120	KP481	33	Male	13	Partnered	4	4	53439	170
121	KP481	33	Female	16	Partnered	2	3	50028	85
122	KP481	33	Male	16	Partnered	3	3	51165	95
123	KP481	33	Female	16	Partnered	5	3	53439	95
124	KP481	33	Female	18	Single	3	4	47754	74
125	KP481	34	Female	16	Partnered	4	3	64809	95
126	KP481	34	Male	16	Partnered	3	4	59124	85
127	KP481	34	Male	15	Single	3	3	67083	85
128	KP481	35	Female	14	Partnered	3	2	52302	53
129	KP481	35	Male	16	Partnered	3	2	53439	53
130	KP481	35	Female	16	Single	3	2	50028	64
131	KP481	35	Male	16	Partnered	3	3	53439	95
132	KP481	37	Female	16	Partnered	2	3	48891	85
133	KP481	38	Female	16	Partnered	4	3	62535	85
134	KP481	38	Male	16	Partnered	3	3	59124	106
135	KP481	40	Female	16	Partnered	3	3	61398	85
136	KP481	40	Female	16	Single	3	3	57987	85
137	KP481	40	Male	16	Partnered	3	3	64809	95
138	KP481	45	Male	16	Partnered	2	2	54576	42
139	KP481	48	Male	16	Partnered	2	3	57987	64

In [43]:
```
male_customers_kp481 = len(kp481_data[kp481_data['Gender'] == 'Male'])
male_customers_kp481
```

Out[43]: 31

In [47]:
```
probability_male_kp481=male_customers_kp481/total_male_customers*100
probability_male_kp481
```

Out[47]: 29.807692307692307

There is 29.8 % probability that male Customers will purchase KP481