Cardio Fitness Good

February 27, 2022

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

import warnings
warnings.filterwarnings("ignore")
```

[4]:
```
df = pd.read_csv('CardioGoodFitness.csv')
```

[3]:
```
df.head()
```

[3]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	TM195	18	Male	14	Single	3	4	29562	112
1	TM195	19	Male	15	Single	2	3	31836	75
2	TM195	19	Female	14	Partnered	4	3	30699	66
3	TM195	19	Male	12	Single	3	3	32973	85
4	TM195	20	Male	13	Partnered	4	2	35247	47

[4]:
```
df.isnull().sum()
```

[4]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

[5]:
```
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
```

[6]:
```
df.shape
```

[6]:
```
(180, 9)
```

[7]:
```
df.describe().transpose()
```

[7]:

	count	mean	std	min	25%	50%	\
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	
Usage	180.0	3.455556	1.084797	2.0	3.00	3.0	
Fitness	180.0	3.311111	0.958869	1.0	3.00	3.0	
Income	180.0	53719.577778	16506.684226	29562.0	44058.75	50596.5	
Miles	180.0	103.194444	51.863605	21.0	66.00	94.0	

	75%	max
Age	33.00	50.0
Education	16.00	21.0
Usage	4.00	7.0
Fitness	4.00	5.0
Income	58668.00	104581.0
Miles	114.75	360.0

[8]:
```
df.describe(include='all')
```

[8]:

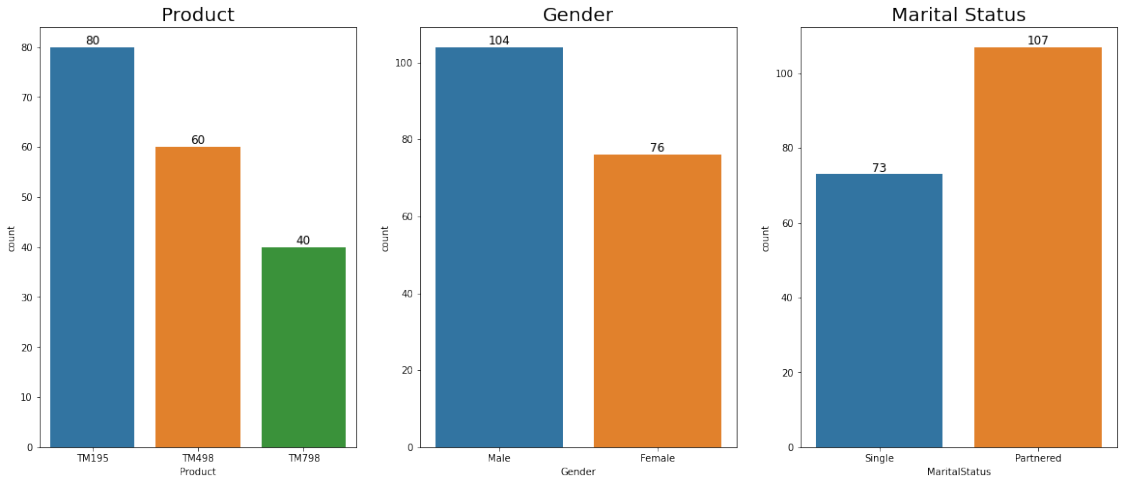
	Product	Age	Gender	Education	MaritalStatus	Usage	\
count	180	180.000000	180	180.000000	180	180.000000	
unique	3	NaN	2	NaN	2	NaN	
top	TM195	NaN	Male	NaN	Partnered	NaN	
freq	80	NaN	104	NaN	107	NaN	
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	

	Fitness	Income	Miles
count	180.000000	180.000000	180.000000
unique	NaN	NaN	NaN
top	NaN	NaN	NaN
freq	NaN	NaN	NaN
mean	3.311111	53719.577778	103.194444
std	0.958869	16506.684226	51.863605
min	1.000000	29562.000000	21.000000
25%	3.000000	44058.750000	66.000000
50%	3.000000	50596.500000	94.000000
75%	4.000000	58668.000000	114.750000
max	5.000000	104581.000000	360.000000

[9]:
```
# Numerical Variables

#    Age variable has 180 values where its range is (18-50), median is at 26␣
 ↪and mean is 28.78, Data is spread with 6 STD
#    Education 180 values with range of 12 to 21, it has mean and median with 0.
 ↪5 units of difference
#    Usage has 180 values with range 2 to 7, it has mean and median of 3.45 and␣
 ↪3.0 respect.
#    Fitness has 180 values with range of 1 to 5 and mean of 3.31 and median 3
#    Income : has 180 non null values iwht mean of 53719.5777778 units and range␣
 ↪is between 29k to 104k
#    Miles: has 180 non null values with mean of 103.19 and range is 21.0 to␣
 ↪360.0

#Categorical Variables

#   Product has 180 non null values,3 unique with top value is of TM195 with␣
 ↪frequency of 80
#   Gender has 180 non null values with 2 unique with top value is Male with␣
 ↪104 freq
#   Marital Status has 180 non null values with 2 unique values| Partnered is␣
 ↪most column has 107 freq
```

[10]:
```
plt.figure(figsize=(20,8))

plt.subplot(1,3,1)
sns.countplot(df['Product'], capsize=0.2)
plt.title("Product", size=20)

ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%d' %int(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom' )
```

```
#gender
plt.subplot(1,3,2)
sns.countplot(df['Gender'], capsize=0.2)
plt.title("Gender", size=20)

ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%d' %int(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom' )

#MaritalStatus
plt.subplot(1,3,3)
sns.countplot(df['MaritalStatus'], capsize=0.2)
plt.title('Marital Status', size=20)

ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%d' %int(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom' )
```

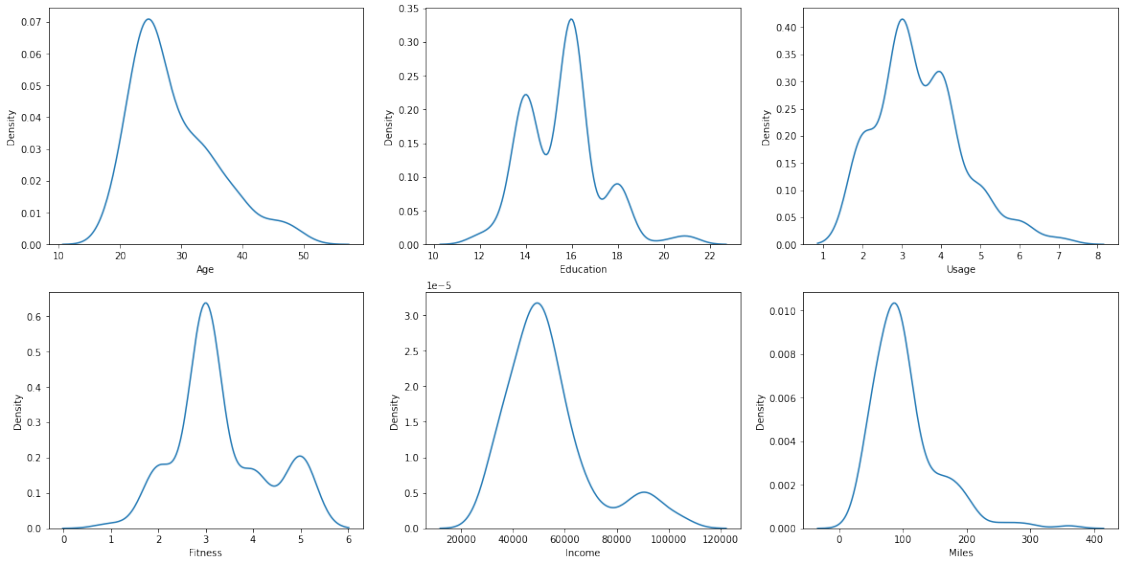


[11]:
```
# For Product we have Three categories:
#                                   - TM195 with 80 values
#                                   - TM498 with 60 values
#                                   - TM798 with 40 values
#    TM195 is dominating, most of the users prefer this product, we investigate␣
 ↪more about the price of this product.

# For Gender, we have two categories here
```

```
#                                     - Male with 104 values
#                                     - Female with 76 values

#                                      Male are dominating the dataset

# MaritalStatus have two categories with Single and Partnered.

#                                     - single are 73 values
#                                     - partnered are 107 values

#                           Partenred are more than sigle for this dataset
```

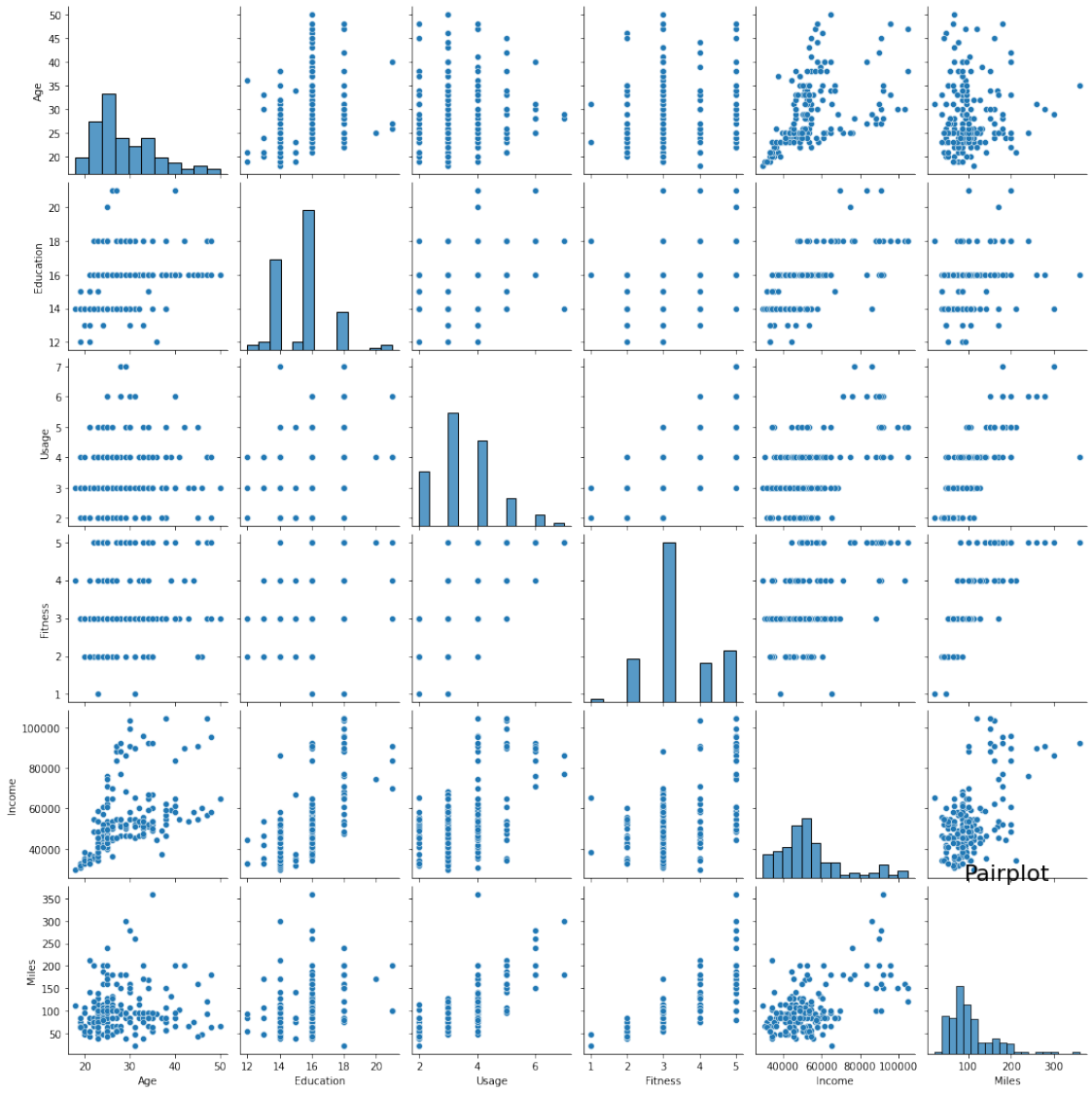
[12]:
```
plt.subplots(2,3, figsize=(20, 10))

plt.subplot(2,3,1)
sns.kdeplot(df['Age'])

plt.subplot(2,3,2)
sns.kdeplot(df['Education'])

plt.subplot(2,3,3)
sns.kdeplot(df['Usage'])

plt.subplot(2,3,4)
sns.kdeplot(df['Fitness'])

plt.subplot(2,3,5)
sns.kdeplot(df['Income'])

plt.subplot(2,3,6)
sns.kdeplot(df['Miles'])
```

[12]:
```
<AxesSubplot:xlabel='Miles', ylabel='Density'>
```



[13]:
```
sns.pairplot(df)
plt.title('Pairplot', size=22)
```

[13]:
```
Text(0.5, 1.0, 'Pairplot')
```

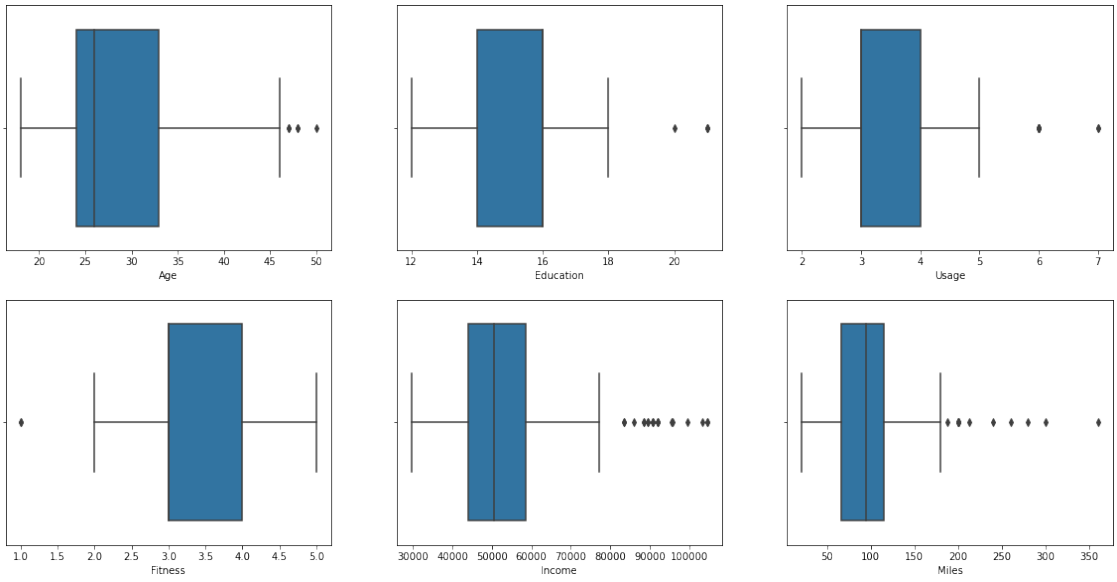


[14]:
```
#  Age and income has kind of linear relationship.
#  Income and miles also share the linear relationship
#  As miles is increasing ussage increase(which is obvious)
#  Fitness increases as mile increases
#  Age and miles also has linear relationship but as it increases the values␣
 ↪has variations in it.
#  Ussage and income : those having less income has high ussage.
```

[15]:
```
plt.subplots(2,3, figsize=(20, 10))

plt.subplot(2,3,1)
sns.boxplot(df['Age'])
```

```
plt.subplot(2,3,2)
sns.boxplot(df['Education'])

plt.subplot(2,3,3)
sns.boxplot(df['Usage'])

plt.subplot(2,3,4)
sns.boxplot(df['Fitness'])

plt.subplot(2,3,5)
sns.boxplot(df['Income'])

plt.subplot(2,3,6)
sns.boxplot(df['Miles'])
```

[15]: <AxesSubplot:xlabel='Miles'>



[16]:
```
# Age: it has 2-3 values which exceed the IQR and our boxplots are considering␣
 ↪them as outlier.
# Education : it has two values while others are ranging 12 to 18
# Fitness has one outlier which doesnot have fitness value(must be me ;p)
# Income : as expected income has lot of outliers in it. income generally most␣
 ↪of people has same in a organization & few has great fancy incomes.
# Miles : has outliers too, means few people really push themself.
```

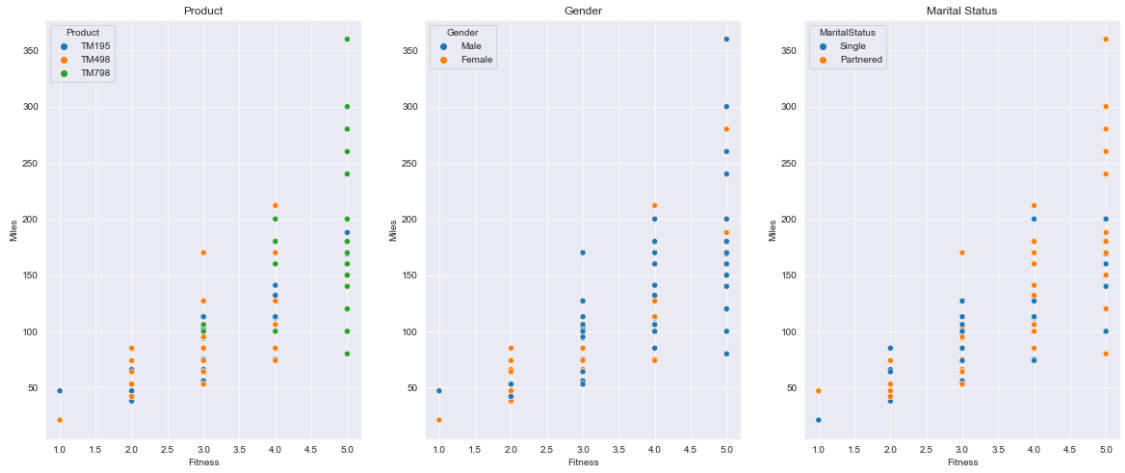
[17]:
```
plt.figure(figsize=(20,8))
corr = df.corr()
```

```
sns.heatmap(corr, annot=True, cmap='Blues', linecolor='white', linewidth=1)
```

[17]: <AxesSubplot:>

[134]:
```
plt.subplots(1 ,3 , figsize=(20, 8))
sns.set_style('darkgrid')

plt.subplot(1,3,1)
sns.scatterplot(data=df, x='Fitness', y='Miles', hue='Product')
plt.title('Product')

plt.subplot(1,3,2)
sns.scatterplot(data=df, x='Fitness', y='Miles', hue='Gender')
plt.title('Gender')

plt.subplot(1,3,3)
sns.scatterplot(data=df, x='Fitness', y='Miles', hue='MaritalStatus')
plt.title('Marital Status')
```

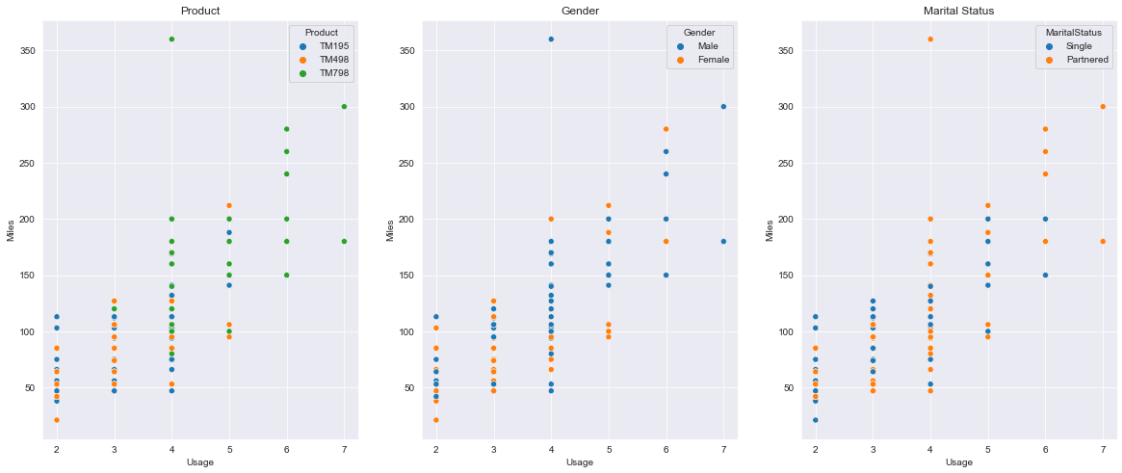
[134]: Text(0.5, 1.0, 'Marital Status')



[19]:
```
# With 0.79 of correlation between fitness and miles, we can clearly see that␣
 ↪the data is linearly increasing,
# So we can say here that once people start getting fit there Run per miles␣
 ↪increases.
# That means High miles value has better fitness but there are exception as we␣
 ↪can see the data has lot of variations at the fitness : 5.
# we can see there is a trend after fitness level 4 we can the people are using␣
 ↪TM798 and which is quiet clear here but Most of the are using TM498
# More Males are Fitness freak, 1,2 are mostly female level but after that Male␣
 ↪values are increasing. but female values are not decreasing, but the␣
 ↪densitiy is decreasing for females since Males are increasing
# Partnered are more fitness freaks
```

[133]:
```
plt.subplots(1 ,3 , figsize=(20, 8))
sns.set_style('darkgrid')

plt.subplot(1,3,1)
sns.scatterplot(data=df, x='Usage', y='Miles', hue='Product')
plt.title('Product')

plt.subplot(1,3,2)
sns.scatterplot(data=df, x='Usage', y='Miles', hue='Gender')
plt.title('Gender')

plt.subplot(1,3,3)
sns.scatterplot(data=df, x='Usage', y='Miles', hue='MaritalStatus')
plt.title('Marital Status')
```

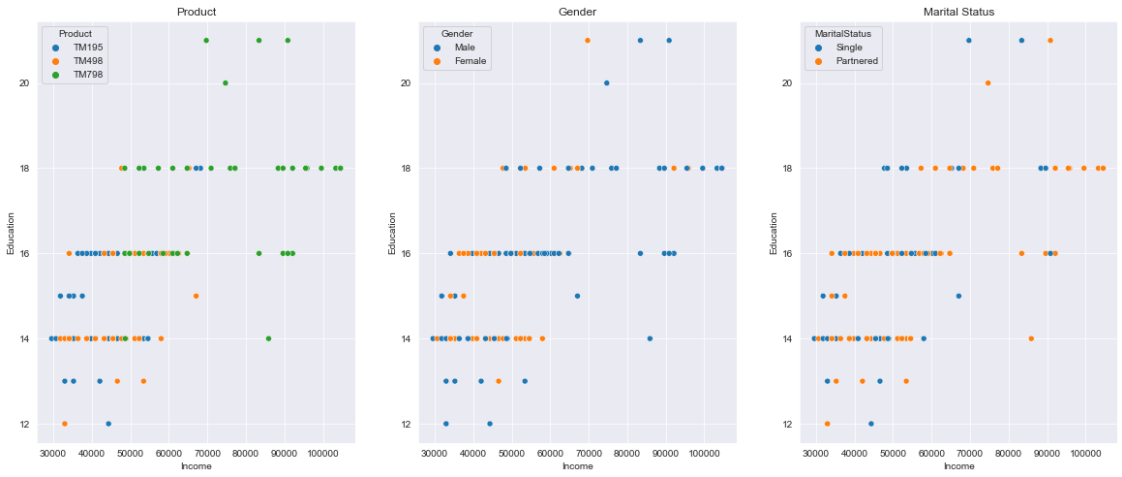
[133]:
```
Text(0.5, 1.0, 'Marital Status')
```



[23]:
```
# Since the corr are +ve all the values are incrementing only
# we can see most of the people usage is TM798 tredmill but with ussage of 2 &3 TM498 is more.
# Gender is not biased both of them has same usage just male are having outlier values of Miles
# Marital status, since in our Partnereds are more we can see more usage and Miles have parterened, ( I think competitions help us to do incredible work)
```

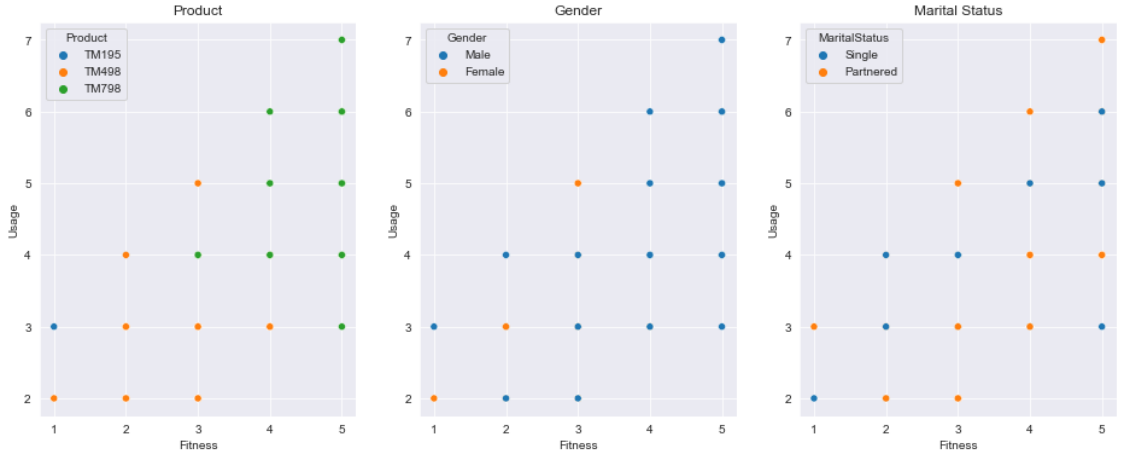
[131]:
```
plt.subplots(1 ,3 , figsize=(20, 8))
sns.set_style('darkgrid')

plt.subplot(1,3,1)
sns.scatterplot(data=df, x='Income', y='Education', hue='Product')
plt.title('Product')

plt.subplot(1,3,2)
sns.scatterplot(data=df, x='Income', y='Education', hue='Gender')
plt.title('Gender')

plt.subplot(1,3,3)
sns.scatterplot(data=df, x='Income', y='Education', hue='MaritalStatus')
plt.title('Marital Status')
```

[131]:
```
Text(0.5, 1.0, 'Marital Status')
```



[54]:
```
plt.subplots(1 ,3 , figsize=(16, 6))
sns.set_style('darkgrid')

plt.subplot(1,3,1)
sns.scatterplot(data=df, x='Fitness', y='Usage', hue='Product')
plt.title('Product')

plt.subplot(1,3,2)
sns.scatterplot(data=df, x='Fitness', y='Usage', hue='Gender')
plt.title('Gender')

plt.subplot(1,3,3)
sns.scatterplot(data=df, x='Fitness', y='Usage', hue='MaritalStatus')
plt.title('Marital Status')
```

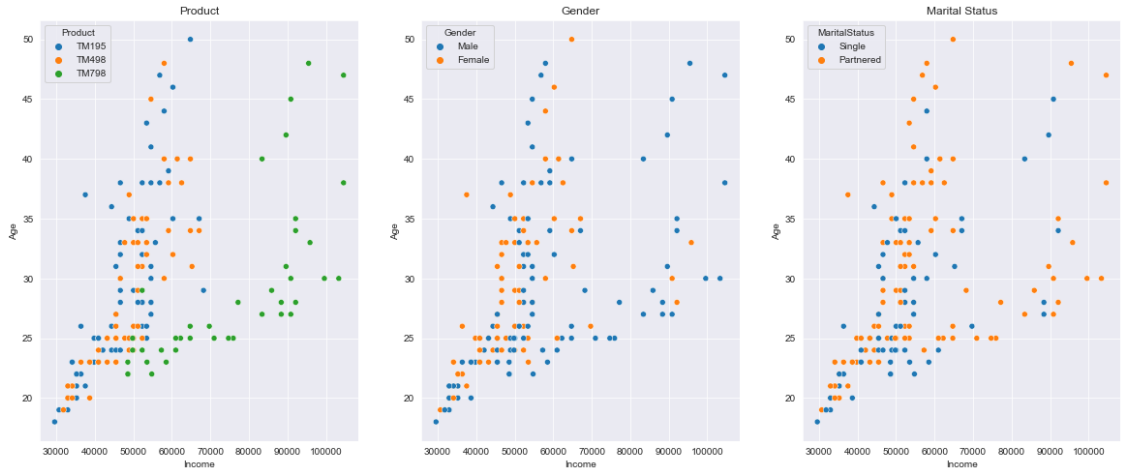
[54]:
```
Text(0.5, 1.0, 'Marital Status')
```



[55]:
```
# Clearly in first image we can see the clusters with Products.
# Yes fitness 3-5 are TM798 and Gender and Maritalstatus has above observations
```

[130]:
```
plt.subplots(1 ,3 , figsize=(20, 8))
sns.set_style('darkgrid')

plt.subplot(1,3,1)
sns.scatterplot(data=df, x='Income', y='Age', hue='Product')
plt.title('Product')

plt.subplot(1,3,2)
sns.scatterplot(data=df, x='Income', y='Age', hue='Gender')
plt.title('Gender')

plt.subplot(1,3,3)
sns.scatterplot(data=df, x='Income', y='Age', hue='MaritalStatus')
plt.title('Marital Status')
```

[130]:
```
Text(0.5, 1.0, 'Marital Status')
```



[57]:
```
# Income increases as Age increase is clearly vissible
# Those who has higher income & lower age are prefering TM798 ( I really think␣
 ↪treadmill 798 is awesome among all)
# Looks like male at lower age earns more and females are more in age of 20-35␣
 ↪with income 40000-60000
# Single and partenered are not different here
```

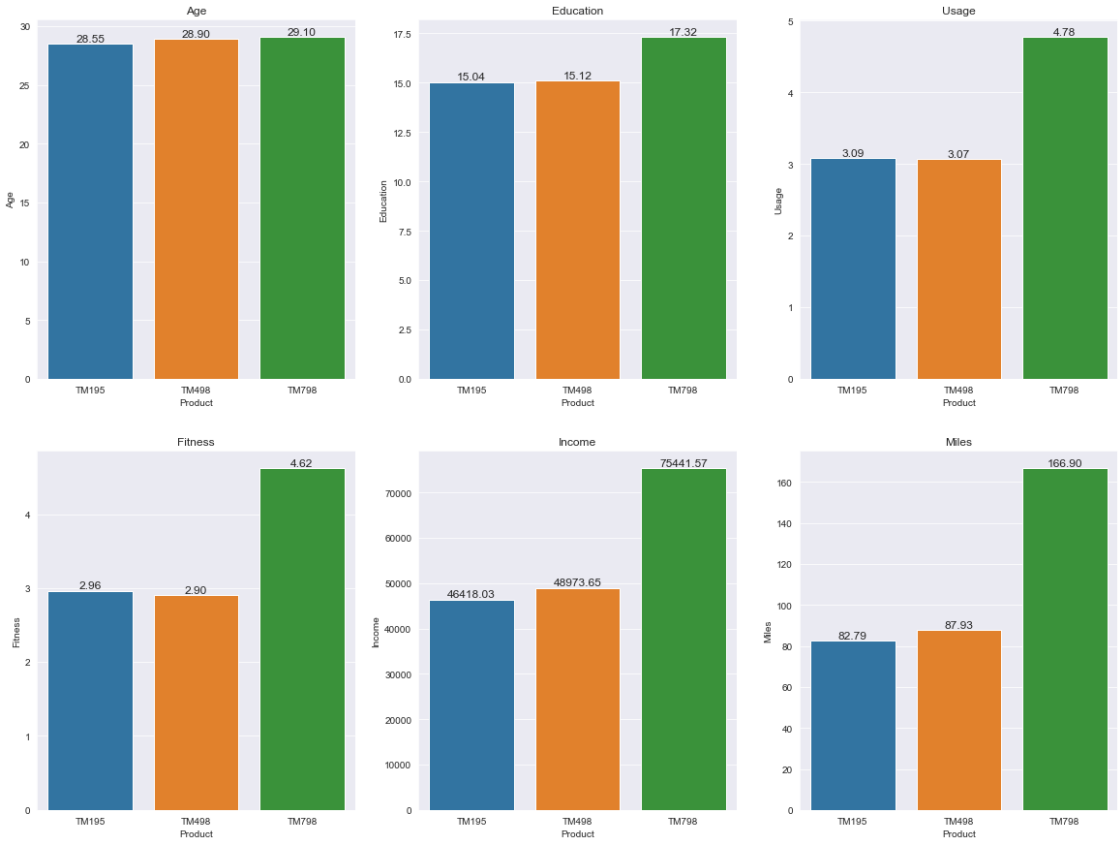
[28]:
```
Miles_group = df.groupby('Product').mean().reset_index()
Miles_group
```

[28]:

	Product	Age	Education	Usage	Fitness	Income	Miles
0	TM195	28.55	15.037500	3.087500	2.9625	46418.025	82.787500
1	TM498	28.90	15.116667	3.066667	2.9000	48973.650	87.933333
2	TM798	29.10	17.325000	4.775000	4.6250	75441.575	166.900000

[105]:
```
plt.subplots(2,3, figsize=(20, 15))

plt.subplot(2,3,1)
sns.barplot(data=Miles_group, x='Product', y='Age')
plt.title('Age')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,2)
sns.barplot(data=Miles_group, x='Product', y='Education')
plt.title('Education')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,3)
sns.barplot(data=Miles_group, x='Product', y='Usage')
plt.title('Usage')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,4)
sns.barplot(data=Miles_group, x='Product', y='Fitness')
plt.title('Fitness')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,5)
sns.barplot(data=Miles_group, x='Product', y='Income')
plt.title('Income')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
```

```
plt.subplot(2,3,6)
sns.barplot(data=Miles_group, x='Product', y='Miles')
plt.title('Miles')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
```



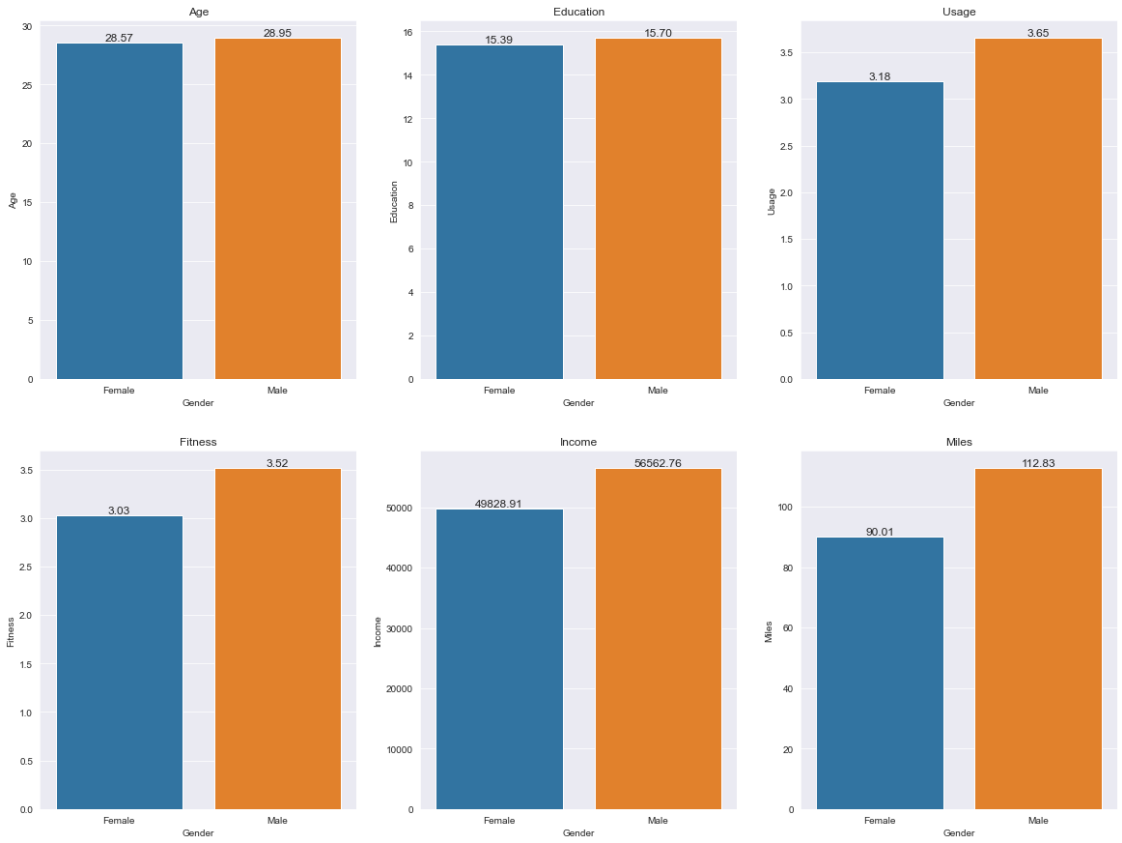
[102]:
```
Gender_group = df.groupby('Gender').mean().reset_index()
Gender_group
```

[102]:

	Gender	Age	Education	Usage	Fitness	Income	Miles
0	Female	28.565789	15.394737	3.184211	3.026316	49828.907895	90.013158
1	Male	28.951923	15.701923	3.653846	3.519231	56562.759615	112.826923

[107]:
```
plt.subplots(2,3, figsize=(20, 15))

plt.subplot(2,3,1)
sns.barplot(data=Gender_group, x='Gender', y='Age')
plt.title('Age')
```

```
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')


plt.subplot(2,3,2)
sns.barplot(data=Gender_group, x='Gender', y='Education')
plt.title('Education')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,3)
sns.barplot(data=Gender_group, x='Gender', y='Usage')
plt.title('Usage')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,4)
sns.barplot(data=Gender_group, x='Gender', y='Fitness')
plt.title('Fitness')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,5)
sns.barplot(data=Gender_group, x='Gender', y='Income')
plt.title('Income')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,6)
sns.barplot(data=Gender_group, x='Gender', y='Miles')
plt.title('Miles')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
```



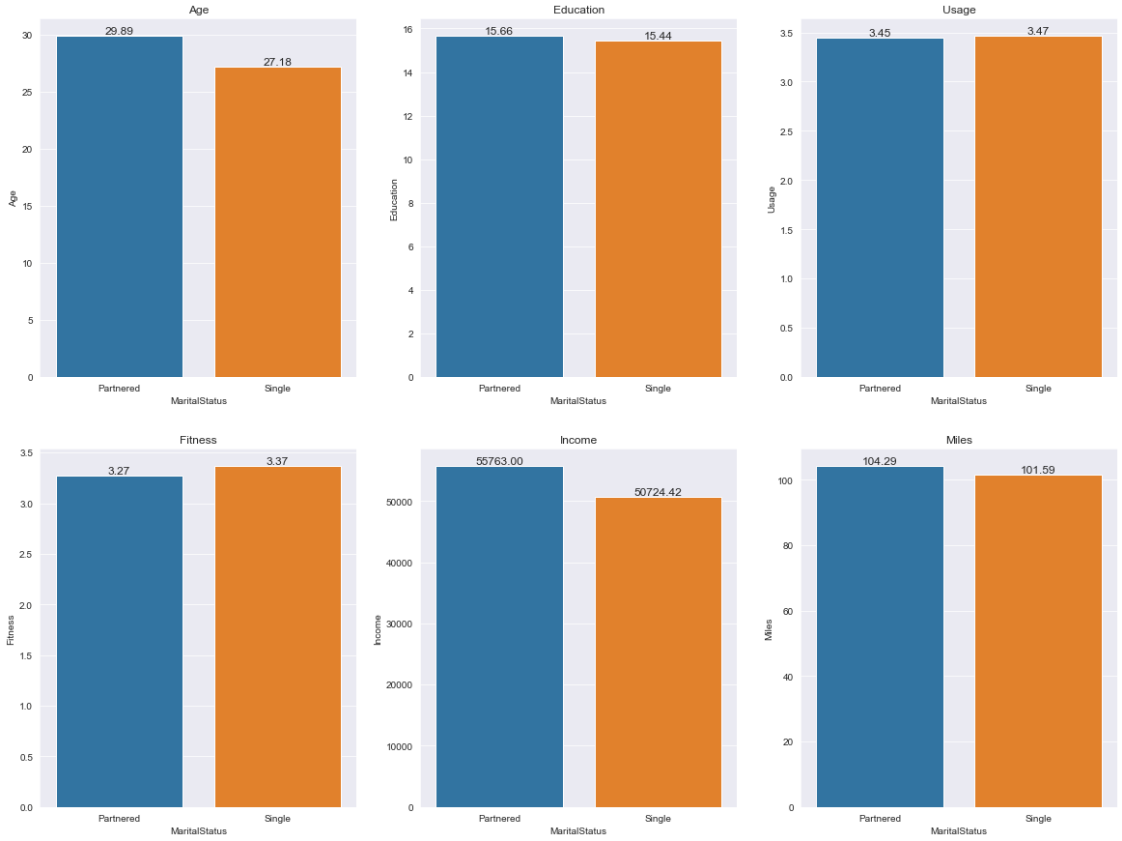
[124]:
```
MaritalStatus_group = df.groupby('MaritalStatus').mean().reset_index()
MaritalStatus_group
```

[124]:

	MaritalStatus	Age	Education	Usage	Fitness	Income	Miles
0	Partnered	29.887850	15.663551	3.448598	3.271028	55763.000000	104.289720
1	Single	27.178082	15.438356	3.465753	3.369863	50724.424658	101.589041

[126]:
```
plt.subplots(2,3, figsize=(20, 15))

plt.subplot(2,3,1)
sns.barplot(data=MaritalStatus_group, x='MaritalStatus', y='Age')
plt.title('Age')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
↪get_height()), fontsize=12, ha='center', va='bottom')
```

```
plt.subplot(2,3,2)
sns.barplot(data=MaritalStatus_group, x='MaritalStatus', y='Education')
plt.title('Education')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,3)
sns.barplot(data=MaritalStatus_group, x='MaritalStatus', y='Usage')
plt.title('Usage')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,4)
sns.barplot(data=MaritalStatus_group, x='MaritalStatus', y='Fitness')
plt.title('Fitness')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,5)
sns.barplot(data=MaritalStatus_group, x='MaritalStatus', y='Income')
plt.title('Income')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')

plt.subplot(2,3,6)
sns.barplot(data=MaritalStatus_group, x='MaritalStatus', y='Miles')
plt.title('Miles')
ax = plt.gca()
for i in ax.patches:
    ax.text(i.get_x() + i.get_width()/2 , i.get_height(), '%.2f' %float(i.
 ↪get_height()), fontsize=12, ha='center', va='bottom')
```


Age
29.89
27.18
Age
Partnered
Single
MaritalStatus
Education
15.66
15.44
Education
Partnered
Single
MaritalStatus
Usage
3.45
3.47
Usage
Partnered
Single
MaritalStatus
Fitness
3.27
3.37
Fitness
Partnered
Single
MaritalStatus
Income
55763.00
50724.42
Income
Partnered
Single
MaritalStatus
Miles
104.29
101.59
Miles
Partnered
Single
MaritalStatus