In [15]:
```python
#Import Python Libraries
import pandas as pd
import numpy as np
```

In [ ]:
```python
#to Ignore Warnings
import warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```python
#Reading Dataset
```

In [192…
```python
gd = pd.read_csv('gym_.csv')
gd.head()
```

Out[192…

	Age	Gender	Weight (kg)	Height (m)	Max_BPM	Avg_BPM	Resting_BPM	Session_Duration (hours)	Calories_Burned	Workout_Type	Fat_Percentage	Water_Intake (liters)	Workout_Frequency (days/week)	Experie
0	56	Male	88.3	1.71	180	157	60	1.69	1313.0	Yoga	12.6	3.5	4	
1	46	Female	74.9	1.53	179	151	66	1.30	883.0	HIIT	33.9	2.1	4	
2	32	Female	68.1	1.66	167	122	54	1.11	677.0	Cardio	33.4	2.3	4	
3	25	Male	53.2	1.70	190	164	56	0.59	532.0	Strength	28.8	2.1	3	
4	38	Male	46.1	1.79	188	158	68	0.64	556.0	Strength	29.2	2.8	3	

In [7]:
```python
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [ ]:

In [13]:
```python
gd.shape
```

Out[13]: (973, 15)

In [19]:
```python
gd.head()
```

Out[19]:

	Age	Gender	Weight (kg)	Height (m)	Max_BPM	Avg_BPM	Resting_BPM	Session_Duration (hours)	Calories_Burned	Workout_Type	Fat_Percentage	Water_Intake (liters)	Workout_Frequency (days/week)	Experie
0	56	Male	88.3	1.71	180	157	60	1.69	1313.0	Yoga	12.6	3.5	4	
1	46	Female	74.9	1.53	179	151	66	1.30	883.0	HIIT	33.9	2.1	4	
2	32	Female	68.1	1.66	167	122	54	1.11	677.0	Cardio	33.4	2.3	4	
3	25	Male	53.2	1.70	190	164	56	0.59	532.0	Strength	28.8	2.1	3	
4	38	Male	46.1	1.79	188	158	68	0.64	556.0	Strength	29.2	2.8	3	

In [21]:
```python
gd.tail()
```

Out[21]:

	Age	Gender	Weight (kg)	Height (m)	Max_BPM	Avg_BPM	Resting_BPM	Session_Duration (hours)	Calories_Burned	Workout_Type	Fat_Percentage	Water_Intake (liters)	Workout_Frequency (days/week)	Expe
968	24	Male	87.1	1.74	187	158	67	1.57	1364.0	Strength	10.0	3.5	4	
969	25	Male	66.6	1.61	184	166	56	1.38	1260.0	Strength	25.0	3.0	2	
970	59	Female	60.4	1.76	194	120	53	1.72	929.0	Cardio	18.8	2.7	5	
971	32	Male	126.4	1.83	198	146	62	1.10	883.0	HIIT	28.2	2.1	3	
972	46	Male	88.7	1.63	166	146	66	0.75	542.0	Strength	28.8	3.5	2	

In [23]:
```python
gd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 973 entries, 0 to 972
Data columns (total 15 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Age                            973 non-null    int64  
 1   Gender                         973 non-null    object 
 2   Weight (kg)                    973 non-null    float64
 3   Height (m)                     973 non-null    float64
 4   Max_BPM                        973 non-null    int64  
 5   Avg_BPM                        973 non-null    int64  
 6   Resting_BPM                    973 non-null    int64  
 7   Session_Duration (hours)       973 non-null    float64
 8   Calories_Burned                973 non-null    float64
 9   Workout_Type                   973 non-null    object 
 10  Fat_Percentage                 973 non-null    float64
 11  Water_Intake (liters)          973 non-null    float64
 12  Workout_Frequency (days/week)  973 non-null    int64  
 13  Experience_Level               973 non-null    int64  
 14  BMI                            973 non-null    float64
dtypes: float64(7), int64(6), object(2)
memory usage: 114.2+ KB
```

In [25]:
```python
gd.nunique()
```

Out[25]:
```
Age                               42
Gender                             2
Weight (kg)                      532
Height (m)                        51
Max_BPM                           40
Avg_BPM                           50
Resting_BPM                       25
Session_Duration (hours)         147
Calories_Burned                  621
Workout_Type                       4
Fat_Percentage                   239
Water_Intake (liters)             23
Workout_Frequency (days/week)      4
Experience_Level                   3
BMI                              771
dtype: int64
```

In [27]:
```python
gd.isnull().sum()
```

Out[27]:
```
Age                              0
Gender                           0
Weight (kg)                      0
Height (m)                       0
Max_BPM                          0
Avg_BPM                          0
Resting_BPM                      0
Session_Duration (hours)         0
Calories_Burned                  0
Workout_Type                     0
Fat_Percentage                   0
Water_Intake (liters)            0
Workout_Frequency (days/week)    0
Experience_Level                 0
BMI                              0
dtype: int64
```

In [29]:
```python
# Check for missing values
print("Missing values:\n", gd.isnull().sum())
```

```
Missing values:
 Age                              0
Gender                           0
Weight (kg)                      0
Height (m)                       0
Max_BPM                          0
Avg_BPM                          0
Resting_BPM                      0
Session_Duration (hours)         0
Calories_Burned                  0
Workout_Type                     0
Fat_Percentage                   0
Water_Intake (liters)            0
Workout_Frequency (days/week)    0
Experience_Level                 0
BMI                              0
dtype: int64
```

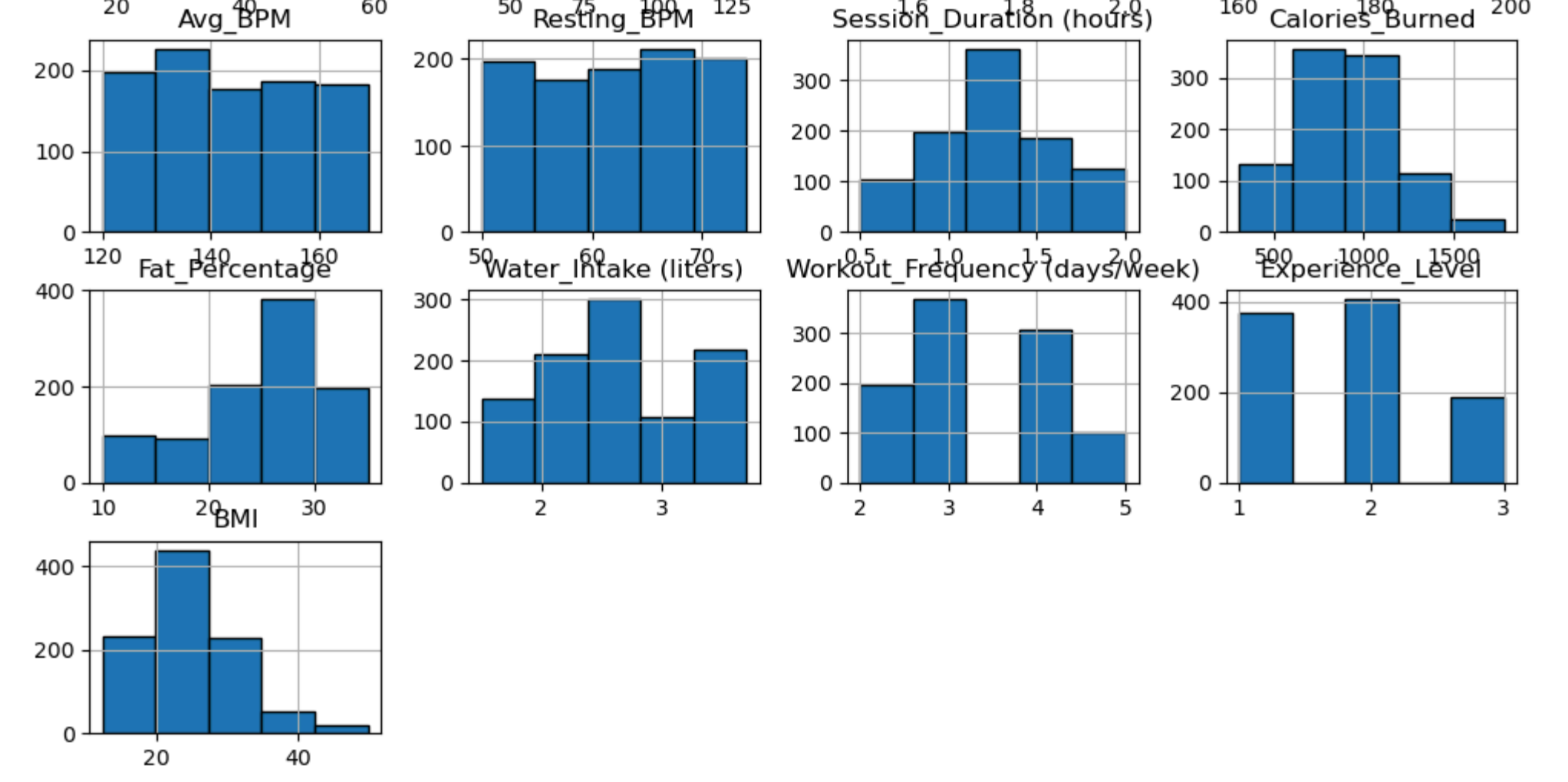
In [33]:
```python
# To seperate the categorical and numerical variables for easy analysis
cat_cols=gd.select_dtypes(include=['object']).columns
num_cols = gd.select_dtypes(include=np.number).columns.tolist()
print("Categorical Variables:")
print(cat_cols)
print("Numerical Variables:")
print(num_cols)
```

```
Categorical Variables:
Index(['Gender', 'Workout_Type'], dtype='object')
Numerical Variables:
['Age', 'Weight (kg)', 'Height (m)', 'Max_BPM', 'Avg_BPM', 'Resting_BPM', 'Session_Duration (hours)', 'Calories_Burned', 'Fat_Percentage', 'Water_I
ntake (liters)', 'Workout_Frequency (days/week)', 'Experience_Level', 'BMI']
```

In [35]:
```python
# EDA Univariate Analysis
```

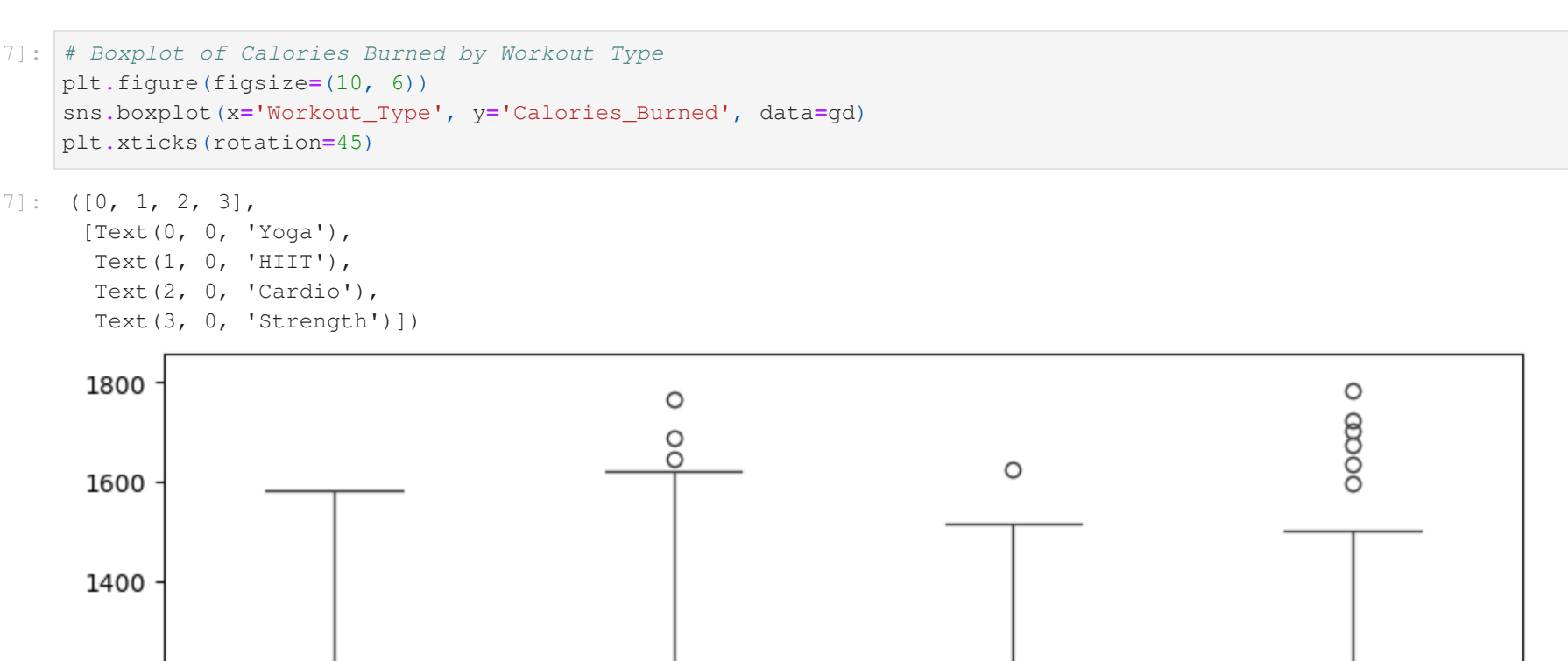
In [39]:
```python
# Libraries to plot the charts
import matplotlib.pyplot as plt
import seaborn as sns
```

In [51]:
```python
# Histograms of numerical features
gd.hist(figsize=(12, 8), bins=5, edgecolor='black')
plt.show()
```



In [47]:
```python
# Boxplot of Calories Burned by Workout Type
plt.figure(figsize=(10, 6))
sns.boxplot(x='Workout_Type', y='Calories_Burned', data=gd)
plt.xticks(rotation=45)
```

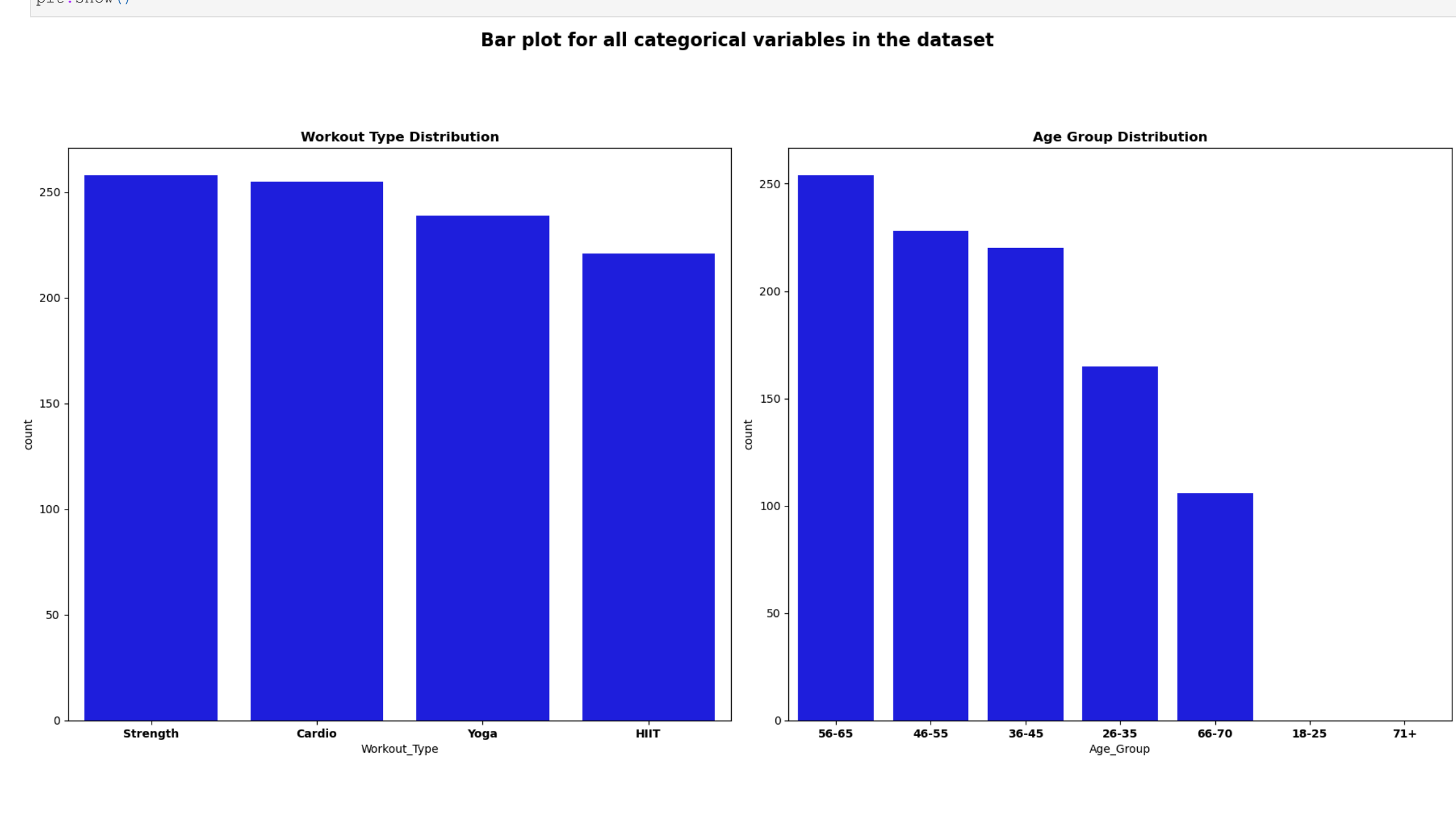
Out[47]:
```
([0, 1, 2, 3],
 [Text(0, 0, 'Yoga'),
  Text(1, 0, 'HIIT'),
  Text(2, 0, 'Cardio'),
  Text(3, 0, 'Strength')])
```



In [ ]:

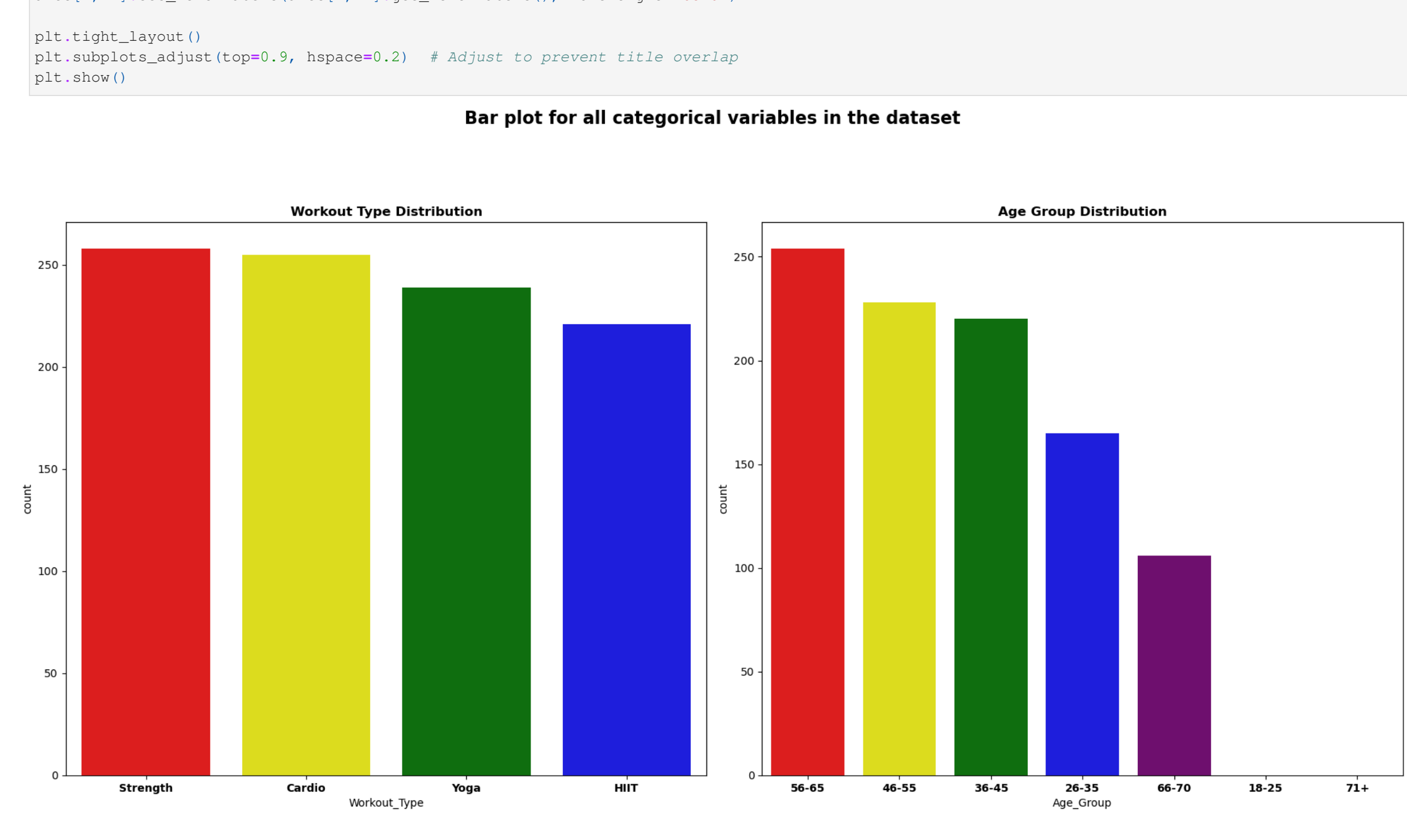
In [129…
```python
# Define age bins
age_bins = [0, 18, 25, 35, 45, 55, 65, 100]
age_labels =['18-25', '26-35', '36-45', '46-55', '56-65', '66-70', '71+']

# Define session duration bins
session_duration_bins = [0, 1, 2, 3, 4, 5, 24]  # Adjust based on your data range
session_duration_labels = ['0-30 Min', '30 Min-1 hour', '1-2 hours', '2-3 hours', '3-4 hours', '4-5 hours']


# Create new grouped columns
gd['Age_Group'] = pd.cut(gd['Age'], bins=age_bins, labels=age_labels, right=False)
gd['Session_Duration_Group'] = pd.cut(gd['Session_Duration (hours)'],
                                      bins=session_duration_bins,
                                      labels=session_duration_labels,
                                      right=False)

# Create subplots
fig, axes = plt.subplots(2, 2, figsize=(18, 18))
fig.suptitle('Bar plot for all categorical variables in the dataset', fontsize=16, fontweight='bold')

# Plot for each categorical variable
sns.countplot(ax=axes[0, 0], x='Workout_Type', data=gd, color= 'blue',
              order=gd['Workout_Type'].value_counts().index)
axes[0, 0].set_title('Workout Type Distribution', fontweight='bold')
axes[0, 0].set_xticklabels(axes[0, 0].get_xticklabels(), fontweight='bold')

sns.countplot(ax=axes[0, 1], x='Age_Group', data=gd, color=  'blue',
              order=gd['Age_Group'].value_counts().index)
axes[0, 1].set_title('Age Group Distribution', fontweight='bold')
axes[0, 1].set_xticklabels(axes[0, 1].get_xticklabels(), fontweight='bold')

sns.countplot(ax=axes[1, 0], x='Session_Duration_Group', data=gd, color=  'blue',
              order=gd['Session_Duration_Group'].value_counts().index)
axes[1, 0].set_title('Session Duration Distribution', fontweight='bold')
axes[1, 0].set_xticklabels(axes[1, 0].get_xticklabels(), fontweight='bold')

sns.countplot(ax=axes[1, 1], x='Workout_Frequency (days/week)', data=gd, color=  'blue',
              order=gd['Workout_Frequency (days/week)'].value_counts().index)
axes[1, 1].set_title('Workout Frequency Distribution', fontweight='bold')
axes[1, 1].set_xticklabels(axes[1, 1].get_xticklabels(), fontweight='bold')

plt.tight_layout()
plt.subplots_adjust(top=0.9, hspace=0.2)  # Adjust to prevent title overlap
plt.show()
```

Bar plot for all categorical variables in the dataset



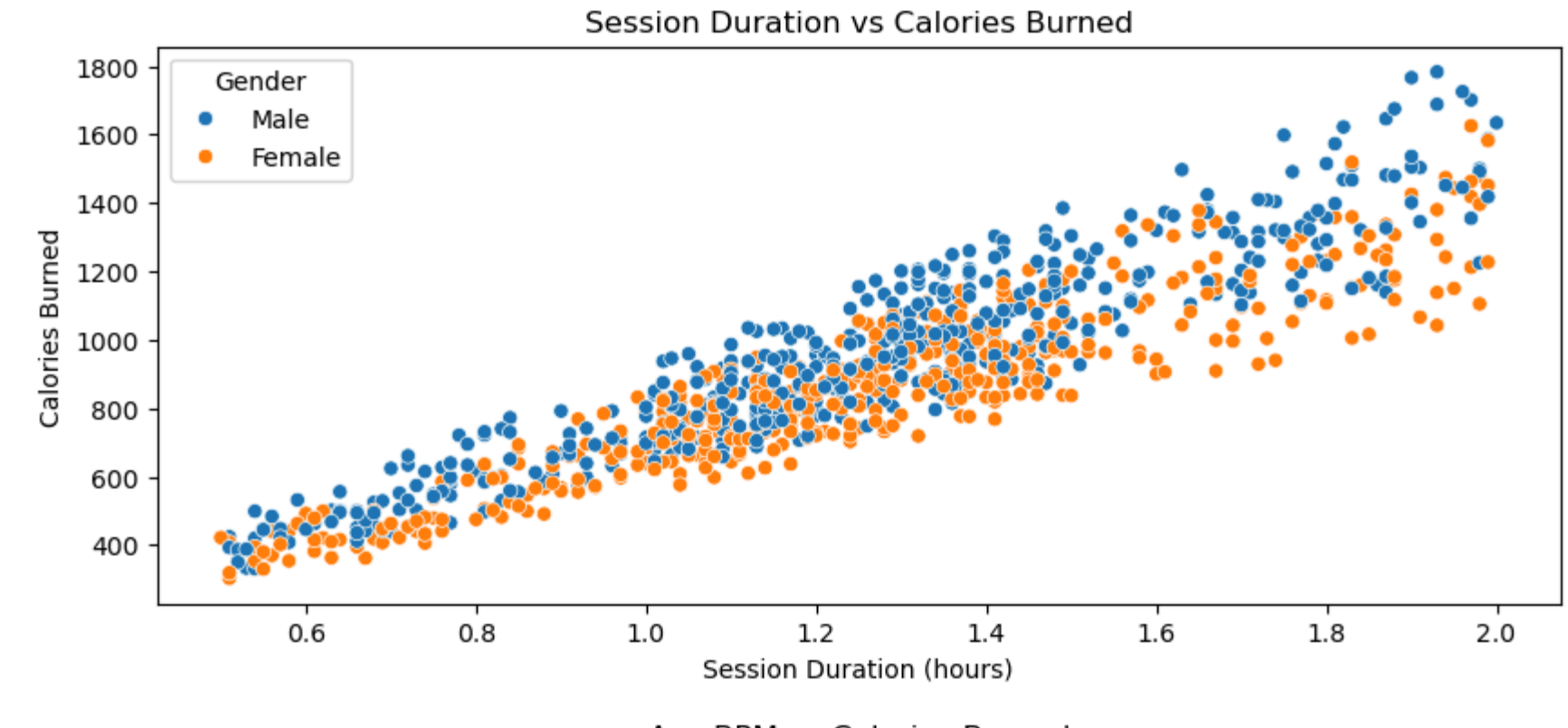
In [133…
```python
# Define age bins
age_bins = [0, 18, 25, 35, 45, 55, 65, 100]
age_labels = ['18-25', '26-35', '36-45', '46-55', '56-65', '66-70', '71+']

# Define session duration bins
session_duration_bins = [0, 1, 2, 3, 4, 5, 24]  # Adjust based on your data range
session_duration_labels = ['0-30 Min', '30 Min-1 hour', '1-2 hours', '2-3 hours', '3-4 hours', '4-5 hours']

# Define custom colors for each category
colors_workout = ['red', 'yellow', 'green', 'blue', 'purple']  # Adjust colors if needed
colors_age = ['red', 'yellow', 'green', 'blue', 'purple', 'pink', 'green']
colors_session = ['red', 'yellow', 'green', 'blue', 'purple', 'pink']
colors_Workout_Frequency = ['red', 'yellow', 'green', 'blue', 'purple', 'pink']

# Create new grouped columns
gd['Age_Group'] = pd.cut(gd['Age'], bins=age_bins, labels=age_labels, right=False)
gd['Session_Duration_Group'] = pd.cut(gd['Session_Duration (hours)'],
                                      bins=session_duration_bins,
                                      labels=session_duration_labels,
                                      right=False)

# Create subplots
fig, axes = plt.subplots(2, 2, figsize=(18, 18))
fig.suptitle('Bar plot for all categorical variables in the dataset', fontsize=16, fontweight='bold')

# Plot for each categorical variable with different colors
sns.countplot(ax=axes[0, 0], x='Workout_Type', data=gd, palette=colors_workout,
              order=gd['Workout_Type'].value_counts().index)
axes[0, 0].set_title('Workout Type Distribution', fontweight='bold')
axes[0, 0].set_xticklabels(axes[0, 0].get_xticklabels(), fontweight='bold')

sns.countplot(ax=axes[0, 1], x='Age_Group', data=gd, palette=colors_age,
              order=gd['Age_Group'].value_counts().index)
axes[0, 1].set_title('Age Group Distribution', fontweight='bold')
axes[0, 1].set_xticklabels(axes[0, 1].get_xticklabels(), fontweight='bold')

sns.countplot(ax=axes[1, 0], x='Session_Duration_Group', data=gd, palette=colors_session,
              order=gd['Session_Duration_Group'].value_counts().index)
axes[1, 0].set_title('Session Duration Distribution', fontweight='bold')
axes[1, 0].set_xticklabels(axes[1, 0].get_xticklabels(), fontweight='bold')

sns.countplot(ax=axes[1, 1], x='Workout_Frequency (days/week)', data=gd, palette=colors_Workout_Frequency,
              order=gd['Workout_Frequency (days/week)'].value_counts().index)
axes[1, 1].set_title('Workout Frequency Distribution', fontweight='bold')
axes[1, 1].set_xticklabels(axes[1, 1].get_xticklabels(), fontweight='bold')

plt.tight_layout()
plt.subplots_adjust(top=0.9, hspace=0.2)  # Adjust to prevent title overlap
plt.show()
```

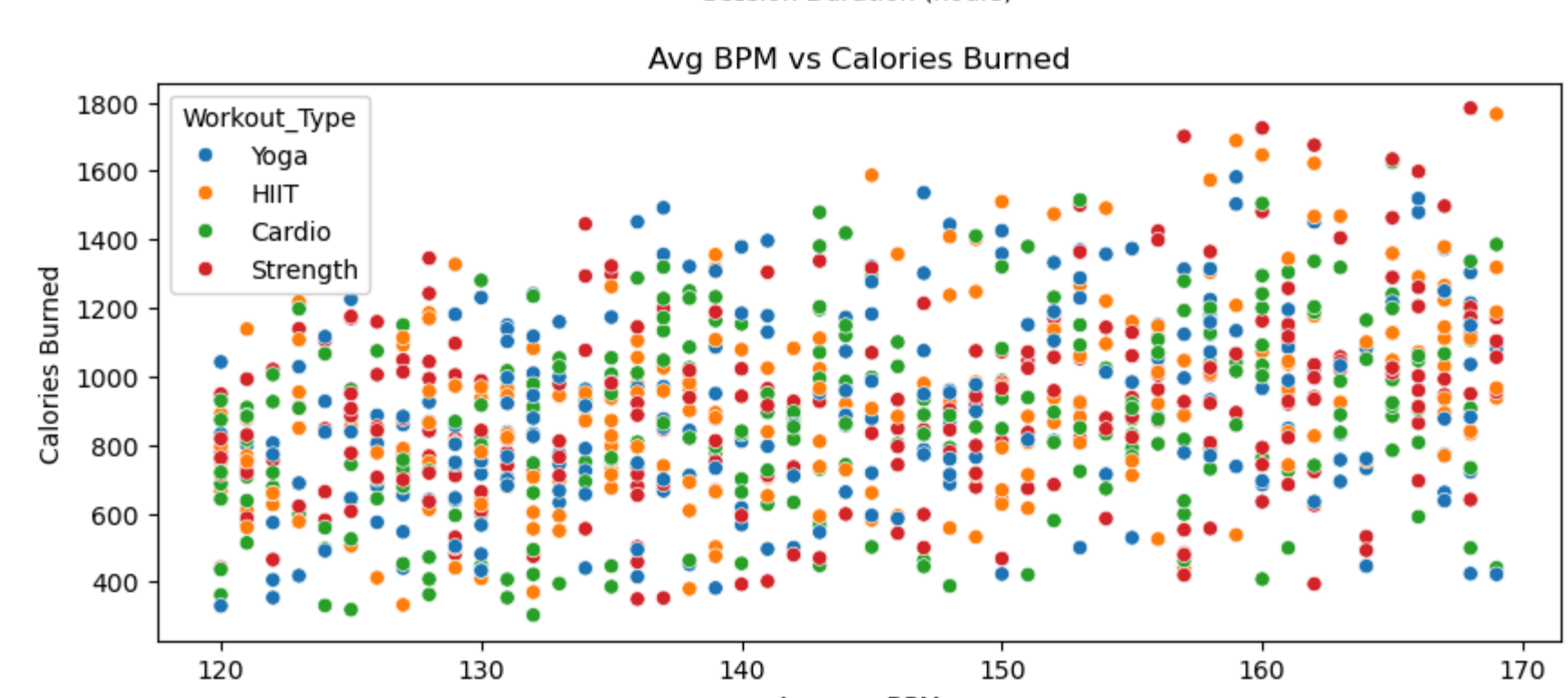
Bar plot for all categorical variables in the dataset



In [151…
```python
# EDA Bivariate Analysis
```

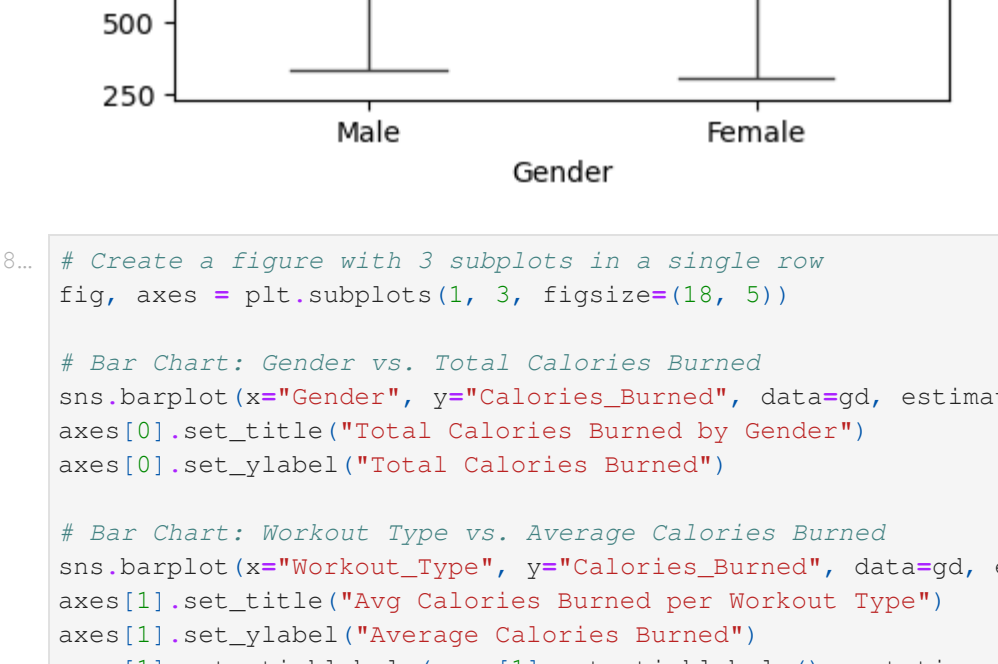
In [180…
```python
# Scatter Plot: Session Duration vs Calories Burned
plt.figure(figsize=(10, 4))
sns.scatterplot(x=gd["Session_Duration (hours)"], y=gd["Calories_Burned"], hue=gd["Gender"])
plt.title("Session Duration vs Calories Burned")
plt.xlabel("Session Duration (hours)")
plt.ylabel("Calories Burned")
plt.show()

# Scatter Plot: Avg BPM vs Calories Burned
plt.figure(figsize=(10, 4))
sns.scatterplot(x=gd["Avg_BPM"], y=gd["Calories_Burned"], hue=gd["Workout_Type"])
plt.title("Avg BPM vs Calories Burned")
plt.xlabel("Average BPM")
plt.ylabel("Calories Burned")
plt.show()
```

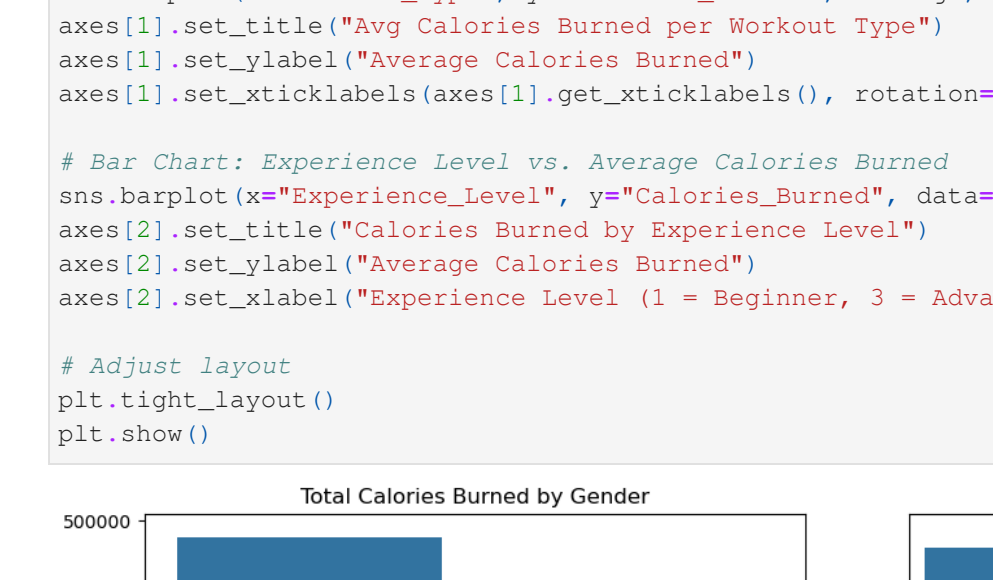




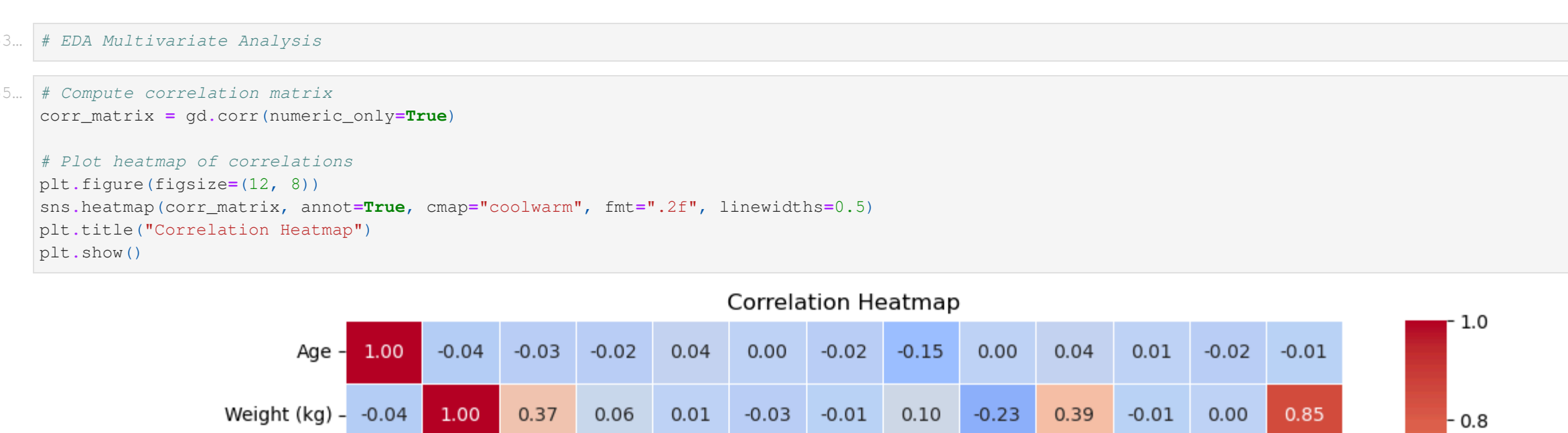
In [ ]:

In [190…
```python
# Box Plot: Workout Type vs Calories Burned
plt.figure(figsize=(5, 3))
sns.boxplot(x=gd["Workout_Type"], y=gd["Calories_Burned"])
plt.xticks(rotation=45)
plt.title("Workout Type vs Calories Burned")
plt.show()

# Box Plot: Gender vs Calories Burned
plt.figure(figsize=(5, 3))
sns.boxplot(x=gd["Gender"], y=gd["Calories_Burned"])
plt.title("Gender vs Calories Burned")
plt.show()
```



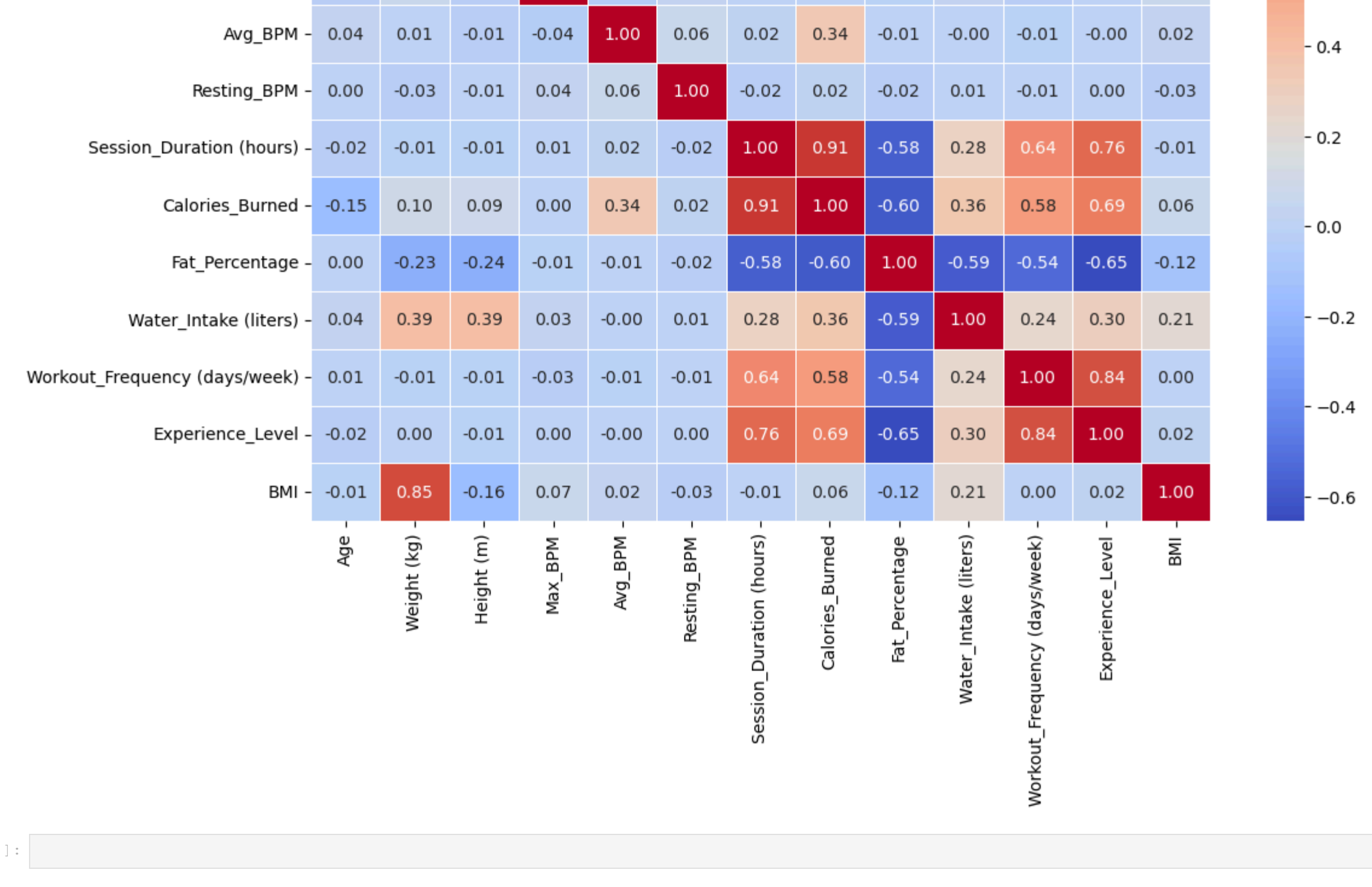


In [188…
```python
# Create a figure with 3 subplots in a single row
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

# Bar Chart: Gender vs. Total Calories Burned
sns.barplot(x="Gender", y="Calories_Burned", data=gd, estimator=sum, ci=None, ax=axes[0])
axes[0].set_title("Total Calories Burned by Gender")
axes[0].set_ylabel("Total Calories Burned")

# Bar Chart: Workout Type vs. Average Calories Burned
sns.barplot(x="Workout_Type", y="Calories_Burned", data=gd, estimator="mean", ci=None, ax=axes[1])
axes[1].set_title("Avg Calories Burned per Workout Type")
axes[1].set_ylabel("Average Calories Burned")
axes[1].set_xticklabels(axes[1].get_xticklabels(), rotation=45)

# Bar Chart: Experience Level vs. Average Calories Burned
sns.barplot(x="Experience_Level", y="Calories_Burned", data=gd, estimator="mean", ci=None, ax=axes[2])
axes[2].set_title("Calories Burned by Experience Level")
axes[2].set_ylabel("Average Calories Burned")
axes[2].set_xlabel("Experience Level (1 = Beginner, 3 = Advanced)")

# Adjust layout
plt.tight_layout()
plt.show()
```



In [153…
```python
# EDA Multivariate Analysis
```

In [155…
```python
# Compute correlation matrix
corr_matrix = gd.corr(numeric_only=True)

# Plot heatmap of correlations
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```



In [ ]: