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

sns.set(style="whitegrid")
```

Interpretation

All the necessary Python libraries will be used in this code to analyze and visualize the data. The computations are done using Pandas, NumPy to handle the data and Matplotlib and Seaborn to visualise. The Seaborn style will be programmed to whitegrid to simplify the graphs and make them easier to understand. These are the popular Exploratory Data Analysis tools.

```
df = pd.read_csv("/content/sports_training_dataset.csv")
df.head()
```

	Athlete_ID	Age	Gender	Sport_Type	Session_ID	Date	Session_Duration	Heart_Rate_Avg	Speed_Avg	Distance_Covered	Enduran
0	1	24	Male	Football	S001	2025-01-01	36	144	5.325258	12226	5
1	2	21	Male	Basketball	S002	2025-01-02	38	146	9.744428	7152	9
2	3	22	Male	Basketball	S003	2025-01-03	53	147	9.828160	14147	8
3	4	24	Female	Basketball	S004	2025-01-04	30	135	9.041987	5585	9
4	5	20	Male	Basketball	S005	2025-01-05	73	134	6.523069	7943	9

Next steps: Generate code with df New interactive sheet

Interpretation

The code takes the dataset of sports training and loads it into a Pandas DataFrame, which is known as df. Next is the head() command that will present the first five rows of the dataset that will help us to become familiar with the format as well as the nature of the variables we have to deal with. We will be able to view the information regarding the athletes, their training regime, and their performance ratings due to the output. The supporting step is a confirmation of a proper loading data.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Athlete_ID         10 non-null     int64
 1   Age                10 non-null     int64
 2   Gender             10 non-null     object
 3   Sport_Type         10 non-null     object
 4   Session_ID         10 non-null     object
 5   Date               10 non-null     object
 6   Session_Duration   10 non-null     int64
 7   Heart_Rate_Avg     10 non-null     int64
 8   Speed_Avg          10 non-null     float64
 9   Distance_Covered   10 non-null     int64
 10  Endurance_Score    10 non-null     float64
 11  Technique_Score    10 non-null     float64
 12  Performance_Level  10 non-null     object
dtypes: float64(3), int64(5), object(5)
memory usage: 1.1+ KB
```

```
df.describe()
```

	Athlete_ID	Age	Session_Duration	Heart_Rate_Avg	Speed_Avg	Distance_Covered	Endurance_Score	Technique_Score
count	10.00000	10.000000	10.000000	10.00000	10.000000	10.000000	10.000000	10.000000
mean	5.50000	21.800000	48.600000	137.80000	7.465927	9335.000000	75.260188	66.048600
std	3.02765	1.813529	16.507237	10.71655	1.734120	3544.659238	17.512748	10.130394
min	1.00000	19.000000	30.000000	122.00000	5.325258	5021.000000	52.035228	52.982026
25%	3.25000	20.250000	37.250000	129.50000	5.838411	7092.750000	58.443645	60.949816
50%	5.50000	22.000000	43.000000	137.50000	7.338324	7893.000000	80.893238	63.641673
75%	7.75000	23.500000	53.000000	145.50000	8.886781	12058.250000	91.180063	70.167655
max	10.00000	24.000000	80.000000	156.00000	9.828160	14805.000000	93.631308	83.149500

Interpretation

The function info provides information on the number of rows, columns, data and data gaps. The describe room provides statistical descriptions in the form of mean, minimum, maximum and standard deviation of numerical variables. The items make it easy to understand overall allocation and size of the data. It is also applied to define any deviant values at its initial analysis stage.

```
df.isnull().sum()
```

	0
Athlete_ID	0
Age	0
Gender	0
Sport_Type	0
Session_ID	0
Date	0
Session_Duration	0
Heart_Rate_Avg	0
Speed_Avg	0
Distance_Covered	0
Endurance_Score	0
Technique_Score	0
Performance_Level	0

dtype: int64

```
df.duplicated().sum()
```

```
np.int64(0)
```

Interpretation

This will be done to identify the existence of any missing or repeated value within a dataset. Loss of values can lead to loss to accuracy of analysis as compared to duplication which can lead to biasing. The output gives information on the necessity to clean up something before it can be subjected to any other analysis. The precision of data would increase depending on the data set.

```
df = df.drop_duplicates()
```

```
df = df.dropna()
```

Interpretation

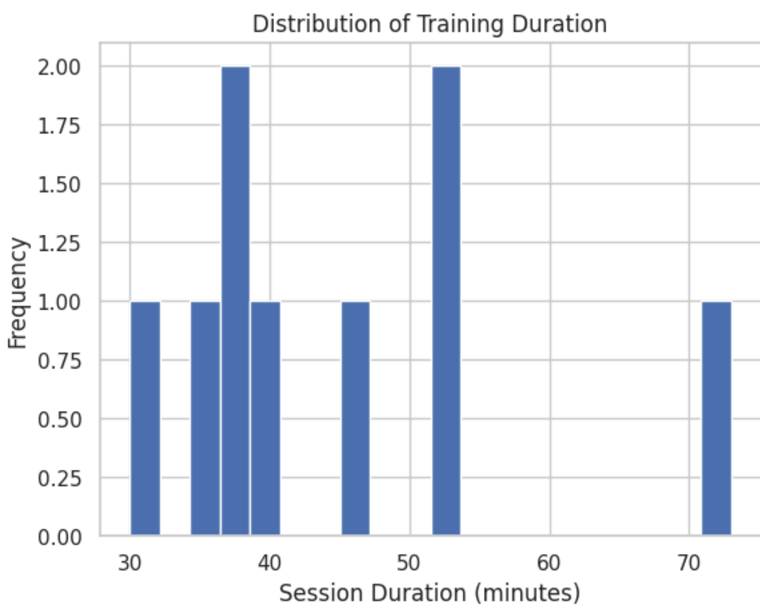
Duplicate data are removed and lost values are removed as a way of achieving data quality. This avoids overlapping of each of the training sessions and inclusion of incomplete contents on analysis, in all the analyses. Regression of these inconsistencies improves visualization and statistical results. It is among the usual procedures in preprocessing EDA.

```
Q1 = df['Session_Duration'].quantile(0.25)
Q3 = df['Session_Duration'].quantile(0.75)
IQR = Q3 - Q1

df = df[
    (df['Session_Duration'] >= Q1 - 1.5 * IQR) &
    (df['Session_Duration'] <= Q3 + 1.5 * IQR)
]
```

Interpretation

This code gets rid of extreme outliers during training time using Interquartile Range (IQR) method. Outliers may be abnormal or wrong training sessions that have the potential to bias reporting. The dataset is able to be filtered to better indicate a normal training behavior. This boosts relationship consistency in the future.

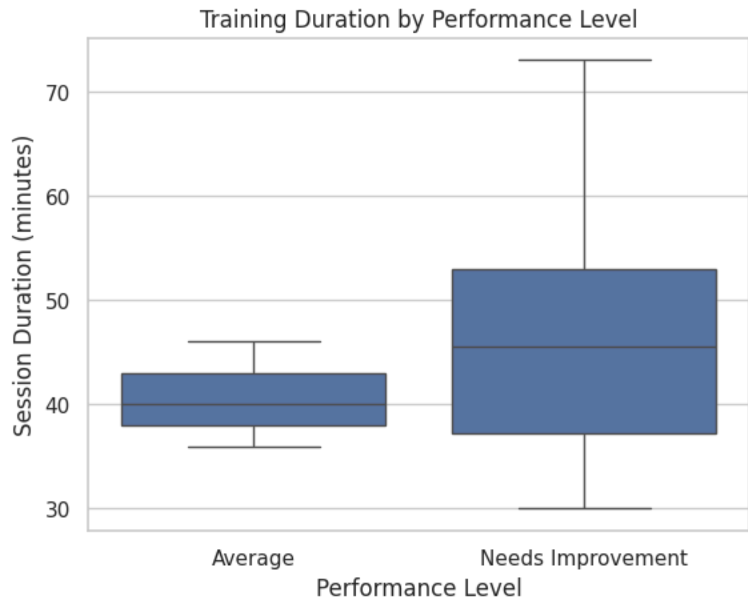
```
plt.hist(df['Session_Duration'], bins=20)
plt.xlabel("Session Duration (minutes)")
plt.ylabel("Frequency")
plt.title("Distribution of Training Duration")
plt.show()
```



Interpretation

This histogram shows the distribution of ages of athletes undergoing the training session. Most of the sessions take medium period, which implies regular training patterns. The sessions of very long and very short sessions are few. This visualization helps one to know how athletes tend to spend some time in training.

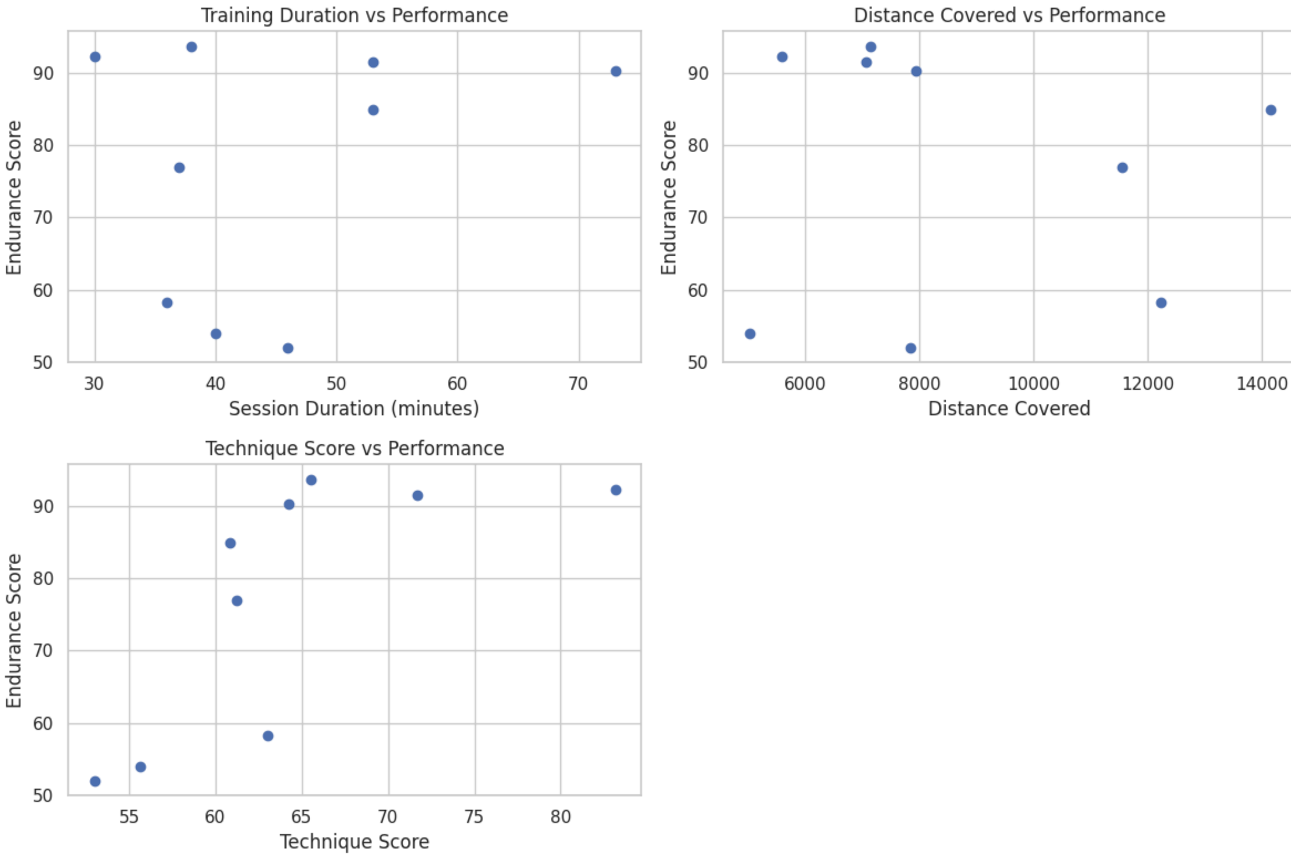
```
sns.boxplot(x='Performance_Level', y='Session_Duration', data=df)
plt.xlabel("Performance Level")
plt.ylabel("Session Duration (minutes)")
plt.title("Training Duration by Performance Level")
plt.show()
```



Interpretation

The boxplot involves a comparison of the training time on different performance levels. The increased performance of the athlete is the one which implies the presence of longer trainings. The groups performing poorly will admitly be likely to be subjected to shorter trainings or have varying training duration. This means that performance outcomes can be caused by the duration of the training.

```
import matplotlib.pyplot as plt

# Create a 2x2 grid of subplots
fig, axes = plt.subplots(2, 2, figsize=(12, 8))

# 1️⃣ Training Duration vs Performance (using Endurance Score as performance)
axes[0, 0].scatter(df['Session_Duration'], df['Endurance_Score'])
axes[0, 0].set_title("Training Duration vs Performance")
axes[0, 0].set_xlabel("Session Duration (minutes)")
axes[0, 0].set_ylabel("Endurance Score")

# 2️⃣ Strength-like metric vs Performance (using Distance Covered)
axes[0, 1].scatter(df['Distance_Covered'], df['Endurance_Score'])
axes[0, 1].set_title("Distance Covered vs Performance")
axes[0, 1].set_xlabel("Distance Covered")
axes[0, 1].set_ylabel("Endurance Score")

# 3️⃣ Skill Score vs Performance (Technique Score)
axes[1, 0].scatter(df['Technique_Score'], df['Endurance_Score'])
axes[1, 0].set_title("Technique Score vs Performance")
axes[1, 0].set_xlabel("Technique Score")
axes[1, 0].set_ylabel("Endurance Score")

# Remove empty subplot (bottom-right)
axes[1, 1].axis('off')

plt.tight_layout()
plt.show()
```



Interpretation

The scatter plots combined have indicated that there is a positive correlation between variables related to training and performance of players. The longer the duration of training, the more is the endurance performance, which results in the fact that the longer the training period, the higher the stamina. Likewise, athletes who travel more during the training sessions are also expected to show higher levels of performance implying the significance of physical work and effort. There is also the positive correlation between the technique score and performance whereby the better the technical skills the better the results. Even though indeed there is a variation in all plots, the general direction is how continuous training on players, physical training along with technical mastery is essential as it augments player