

```
In [10]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [11]: file_path = "Sports_Performance_analysis.csv"
data = pd.read_csv("Sports_Performance_analysis.csv")
```

```
In [12]: data.rename(columns={'index': 'Year'}, inplace=True)
```

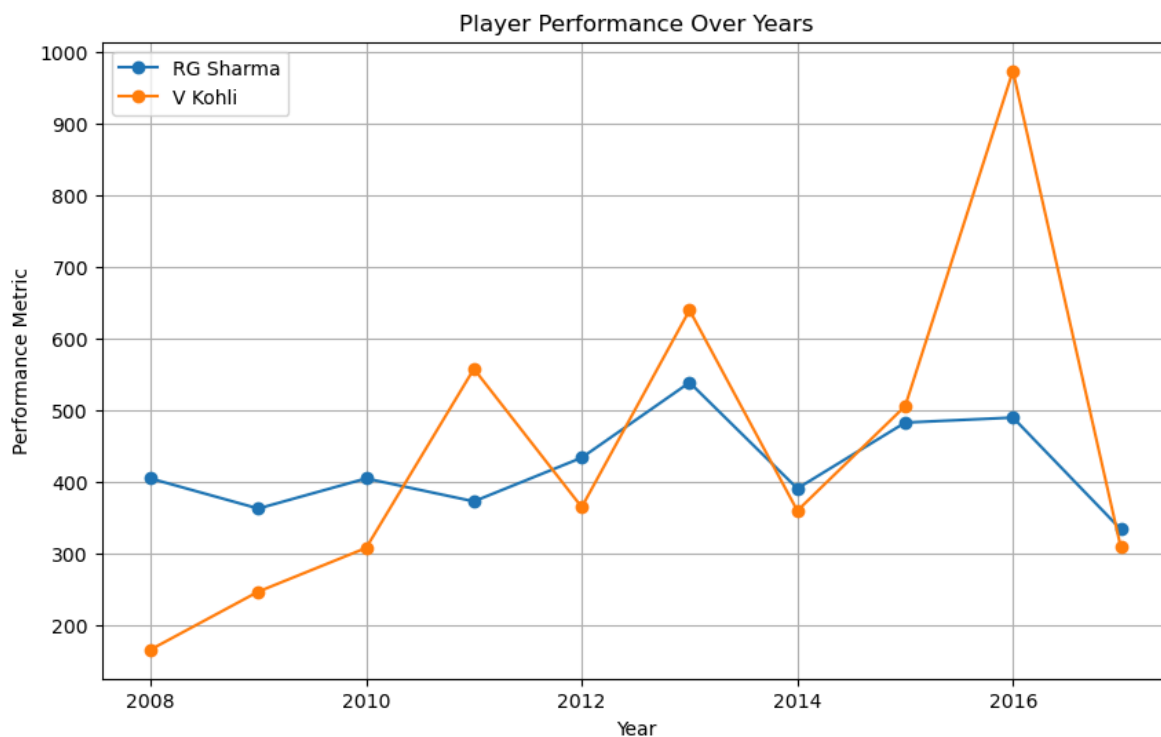
```
In [13]: print(data.head())
```

	Year	RG Sharma	V Kohli
0	2008	404	165
1	2009	362	246
2	2010	404	307
3	2011	372	557
4	2012	433	364

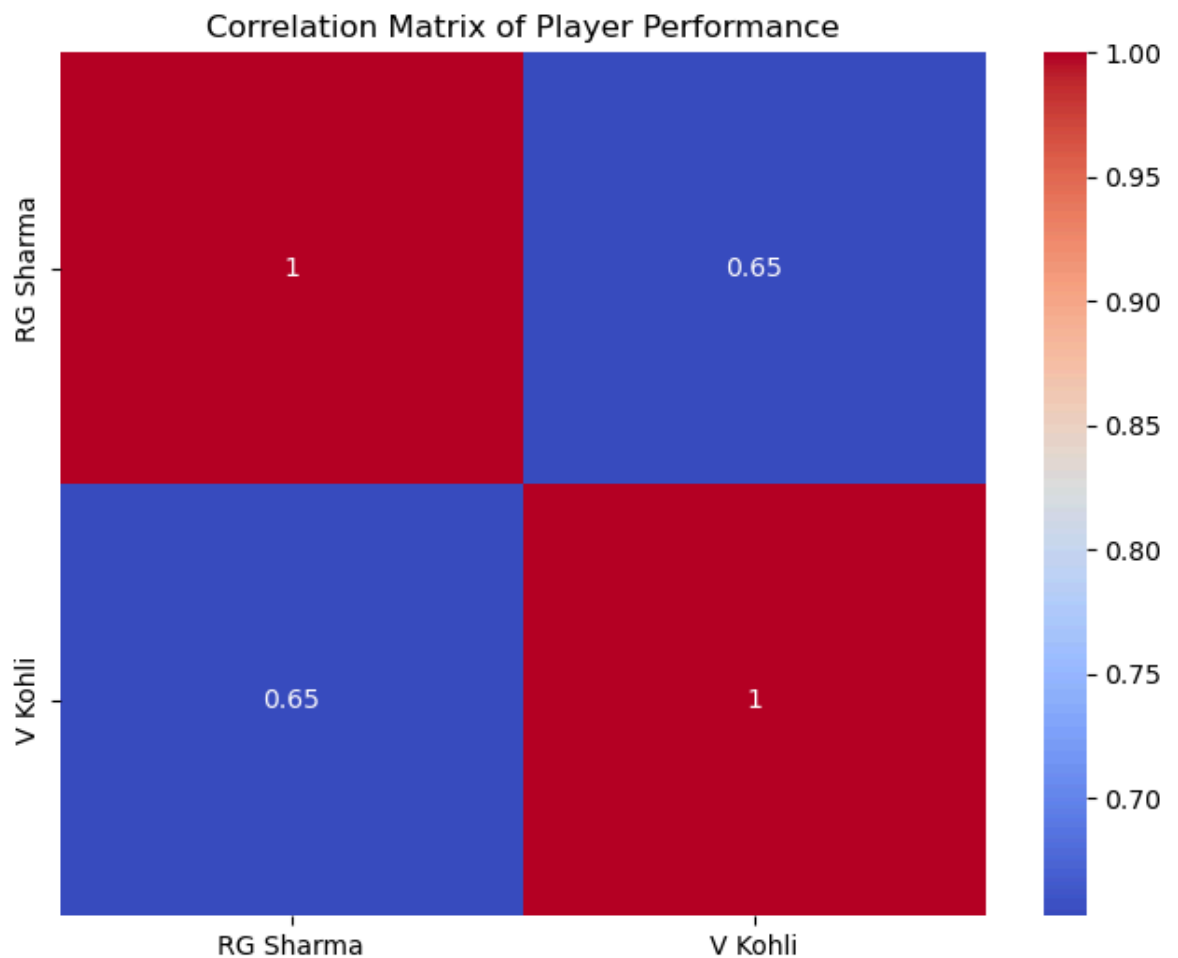
```
In [14]: # Basic Statistics
print(data.describe())
```

	Year	RG Sharma	V Kohli
count	10.00000	10.00000	10.00000
mean	2012.50000	420.70000	442.30000
std	3.02765	64.41368	236.23484
min	2008.00000	333.00000	165.00000
25%	2010.25000	376.50000	307.25000
50%	2012.50000	404.00000	361.50000
75%	2014.75000	469.75000	544.00000
max	2017.00000	538.00000	973.00000

```
In [15]: # Data Visualization
# Plotting player performance metrics
plt.figure(figsize=(10, 6))
for player in data.columns[1:]:
    plt.plot(data['Year'], data[player], marker='o', label=player)
plt.xlabel('Year')
plt.ylabel('Performance Metric')
plt.title('Player Performance Over Years')
plt.legend()
plt.grid()
plt.show()
```



```
In [16]: # Correlation Matrix
plt.figure(figsize=(8, 6))
correlation = data.drop(columns=['Year']).corr()
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Player Performance')
plt.show()
```



```
In [17]: # Predictive Modeling (Example: Linear Regression)
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [18]: # Define features and target variable
X = data[['Year']]
y = data['RG Sharma'] # Predicting RG Sharma's performance
```

```
In [19]: X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                             test_size=0.2, random_state=42)
```

```
In [20]: model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
```

```
In [21]: # Evaluate the model
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
```

```
In [22]: print(f'Mean Squared Error: {mse}')
print(f'R^2 Score: {r2}')
```

Mean Squared Error: 4420.494723543399  
R^2 Score: -0.09628488400853108

```
In [23]: # Calculate total runs scored by RG Sharma
total_runs_rg = data["RG Sharma"].sum()
print("Total runs scored by RG Sharma:", total_runs_rg)
```

Total runs scored by RG Sharma: 4207

```
In [31]: # Compute average runs scored by each player
avg_runs_rg = data["RG Sharma"].mean()
avg_runs_vk = data["V Kohli"].mean()
```

```
In [32]: print("Average runs scored per year:")
print("RG Sharma:", avg_runs_rg)
print("V Kohli:", avg_runs_vk)
```

Average runs scored per year:  
RG Sharma: 420.7  
V Kohli: 442.3