

# IPL PLAYER PERFORMANCE

March 24, 2023

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from IPython.display import Image
```

```
[2]: dataset = pd.read_csv("Most Runs All Seasons Combine.csv")
```

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

```
[3]: Unnamed: 0      Player  Mat  Inns  NO  Runs   HS   Avg   BF  \
0          0      Shaun Marsh   11    11    2   616   115  68.44  441
1          1    Gautam Gambhir   14    14    1   534    86  41.07  379
2          2  Sanath Jayasuriya   14    14    2   518  114*  43.16  309
3          3    Shane Watson   15    15    5   472   76*  47.20  311
4          4    Graeme Smith   11    11    2   441    91  49.00  362
```

```
      SR  100  50  4s  6s
0  139.68    1   5  59  26
1  140.89    0   5  68   8
2  167.63    1   2  58  31
3  151.76    0   4  47  19
4  121.82    0   3  54   8
```

```
[4]: dataset.head(10)
```

```
[4]: Unnamed: 0      Player  Mat  Inns  NO  Runs   HS   Avg   BF  \
0          0      Shaun Marsh   11    11    2   616   115  68.44  441
1          1    Gautam Gambhir   14    14    1   534    86  41.07  379
2          2  Sanath Jayasuriya   14    14    2   518  114*  43.16  309
3          3    Shane Watson   15    15    5   472   76*  47.20  311
4          4    Graeme Smith   11    11    2   441    91  49.00  362
5          5    Adam Gilchrist   14    14    1   436  109*  33.53  318
6          6    Yusuf Pathan   16    15    1   435    68  31.07  243
7          7    Suresh Raina   16    14    3   421   55*  38.27  296
```

8	8	MS Dhoni	16	14	4	414	65	41.40	310
9	9	Virender Sehwag	14	14	2	406	94*	33.83	220

	SR	100	50	4s	6s
0	139.68	1	5	59	26
1	140.89	0	5	68	8
2	167.63	1	2	58	31
3	151.76	0	4	47	19
4	121.82	0	3	54	8
5	137.10	1	3	51	19
6	179.01	0	4	43	25
7	142.22	0	3	35	18
8	133.54	0	2	38	15
9	184.54	0	3	46	21

```
[5]: dataset = dataset.drop("Unnamed: 0", 1)
```

C:\Users\saita\AppData\Local\Temp\ipykernel\_17268\869471881.py:1: FutureWarning:  
In a future version of pandas all arguments of DataFrame.drop except for the  
argument 'labels' will be keyword-only.  
dataset = dataset.drop("Unnamed: 0", 1)

```
[6]: dataset.head(10)
```

```
[6]:
```

	Player	Mat	Inns	NO	Runs	HS	Avg	BF	SR	100	50	\
0	Shaun Marsh	11	11	2	616	115	68.44	441	139.68	1	5	
1	Gautam Gambhir	14	14	1	534	86	41.07	379	140.89	0	5	
2	Sanath Jayasuriya	14	14	2	518	114*	43.16	309	167.63	1	2	
3	Shane Watson	15	15	5	472	76*	47.20	311	151.76	0	4	
4	Graeme Smith	11	11	2	441	91	49.00	362	121.82	0	3	
5	Adam Gilchrist	14	14	1	436	109*	33.53	318	137.10	1	3	
6	Yusuf Pathan	16	15	1	435	68	31.07	243	179.01	0	4	
7	Suresh Raina	16	14	3	421	55*	38.27	296	142.22	0	3	
8	MS Dhoni	16	14	4	414	65	41.40	310	133.54	0	2	
9	Virender Sehwag	14	14	2	406	94*	33.83	220	184.54	0	3	

	4s	6s
0	59	26
1	68	8
2	58	31
3	47	19
4	54	8
5	51	19
6	43	25
7	35	18
8	38	15
9	46	21

```
[8]: dataset.describe()
```

```
[8]:
```

	Mat	Inns	NO	Runs	Avg \
count	1986.000000	1986.000000	1986.000000	1986.000000	1986.000000
mean	8.974824	6.580060	1.527190	128.539778	18.257170
std	5.007739	4.841767	1.583134	155.137676	15.376013
min	1.000000	1.000000	0.000000	0.000000	0.000000
25%	4.000000	2.000000	0.000000	12.000000	6.000000
50%	9.000000	5.000000	1.000000	55.000000	16.000000
75%	14.000000	11.000000	2.000000	202.750000	27.345000
max	19.000000	19.000000	10.000000	973.000000	152.000000

	BF	SR	100	50	4s \
count	1986.000000	1986.000000	1986.000000	1986.000000	1986.000000
mean	100.359013	110.863776	0.033233	0.654582	11.697885
std	114.014540	44.655957	0.205475	1.263126	15.458447
min	1.000000	0.000000	0.000000	0.000000	0.000000
25%	13.000000	88.920000	0.000000	0.000000	1.000000
50%	49.000000	116.270000	0.000000	0.000000	4.000000
75%	161.000000	135.282500	0.000000	1.000000	18.000000
max	640.000000	400.000000	4.000000	9.000000	88.000000

	6s
count	1986.000000
mean	4.798087
std	6.959908
min	0.000000
25%	0.000000
50%	2.000000
75%	7.000000
max	59.000000

```
[9]: dataset.corr()
```

```
[9]:
```

	Mat	Inns	NO	Runs	Avg	BF	SR \
Mat	1.000000	0.789671	0.540217	0.609157	0.362819	0.615239	0.247134
Inns	0.789671	1.000000	0.441835	0.878541	0.602804	0.890826	0.385189
NO	0.540217	0.441835	1.000000	0.260531	0.384056	0.244733	0.235537
Runs	0.609157	0.878541	0.260531	1.000000	0.733148	0.985138	0.394833
Avg	0.362819	0.602804	0.384056	0.733148	1.000000	0.707061	0.485544
BF	0.615239	0.890826	0.244733	0.985138	0.707061	1.000000	0.339437
SR	0.247134	0.385189	0.235537	0.394833	0.485544	0.339437	1.000000
100	0.126150	0.202408	0.039035	0.375016	0.313989	0.325943	0.139376
50	0.446142	0.663761	0.139732	0.874421	0.626722	0.852328	0.287448
4s	0.563845	0.826898	0.161498	0.961136	0.673512	0.954453	0.360638
6s	0.520683	0.745475	0.273157	0.864727	0.667396	0.796813	0.434753

	100	50	4s	6s
Mat	0.126150	0.446142	0.563845	0.520683
Inns	0.202408	0.663761	0.826898	0.745475
NO	0.039035	0.139732	0.161498	0.273157
Runs	0.375016	0.874421	0.961136	0.864727
Avg	0.313989	0.626722	0.673512	0.667396
BF	0.325943	0.852328	0.954453	0.796813
SR	0.139376	0.287448	0.360638	0.434753
100	1.000000	0.255824	0.366525	0.395363
50	0.255824	1.000000	0.856903	0.741434
4s	0.366525	0.856903	1.000000	0.744232
6s	0.395363	0.741434	0.744232	1.000000

```
[10]: print("Highest Average of a player:", dataset["Avg"].max())
```

Highest Average of a player: 152.0

```
[11]: dataset[(dataset["Avg"] >= 152)]["Player"]
```

```
[11]: 42      Luke Pomersbach
      Name: Player, dtype: object
```

```
[12]: print("Highest Strike Rate of a player:", dataset["SR"].max())
```

Highest Strike Rate of a player: 400.0

```
[14]: dataset[(dataset["SR"] >= 400)]["Player"]
```

```
[14]: 729      Syed Mohammad
      878      Raiphi Gomez
      991      Abu Nechim
      Name: Player, dtype: object
```

```
[26]: dataset[(dataset["Avg"] > 80) & (dataset["SR"] > 150)]["Player"]
```

```
[26]: 35      Michael Hussey
      39      Andrew Symonds
      42      Luke Pomersbach
      1143     Virat Kohli
      Name: Player, dtype: object
```

```
[19]: dataset[(dataset["Runs"] > 500)]["Player"]
```

```
[19]: 0      Shaun Marsh
      1      Gautam Gambhir
      2      Sanath Jayasuriya
      150     Matthew Hayden
      282     Sachin Tendulkar
      ...
```

```

1837      Ruturaj Gaikwad
1838      Faf du Plessis
1839      KL Rahul
1840      Shikhar Dhawan
1841      Glenn Maxwell
Name: Player, Length: 62, dtype: object

```

```
[23]: print("Most number of runs:", dataset["Runs"].max())
```

```
Most number of runs: 973
```

```
[24]: dataset[(dataset["Runs"] >= 973)]["Player"]
```

```
[24]: 1143      Virat Kohli
      Name: Player, dtype: object
```

```
[30]: dataset.loc[dataset["Player"]=="Virat Kohli"]
```

```
[30]:
```

	Player	Mat	Inns	NO	Runs	HS	Avg	BF	SR	100	50	4s	\
37	Virat Kohli	13	12	1	165	38	15.00	157	105.09	0	0	18	
167	Virat Kohli	16	13	2	246	50	22.36	219	112.32	0	1	22	
300	Virat Kohli	16	13	2	307	58	27.90	212	144.81	0	1	26	
433	Virat Kohli	16	16	4	557	71	46.41	460	121.08	0	4	55	
597	Virat Kohli	16	15	2	364	73*	28.00	326	111.65	0	2	33	
734	Virat Kohli	16	16	2	634	99	45.28	457	138.73	0	6	64	
900	Virat Kohli	14	14	1	359	73	27.61	294	122.10	0	2	23	
1018	Virat Kohli	16	16	5	505	82*	45.90	386	130.82	0	3	35	
1143	Virat Kohli	16	16	4	973	113	81.08	640	152.03	4	7	83	
1301	Virat Kohli	10	10	0	308	64	30.80	252	122.22	0	4	23	
1428	Virat Kohli	14	14	3	530	92*	48.18	381	139.10	0	4	52	
1567	Virat Kohli	14	14	0	464	100	33.14	328	141.46	1	2	46	
1712	Virat Kohli	15	15	4	466	90*	42.36	384	121.35	0	3	23	
1848	Virat Kohli	15	15	1	405	72*	28.92	339	119.46	0	3	43	

6s

```

37      4
167     8
300    12
433    16
597     9
734    22
900    16
1018   23
1143   38
1301   11
1428   18
1567   13
1712   11

```

```
[28]: # Set the number of lines of gap between histograms
hspace = 4

# Set the height ratios for each subplot
layout = (2, 1)

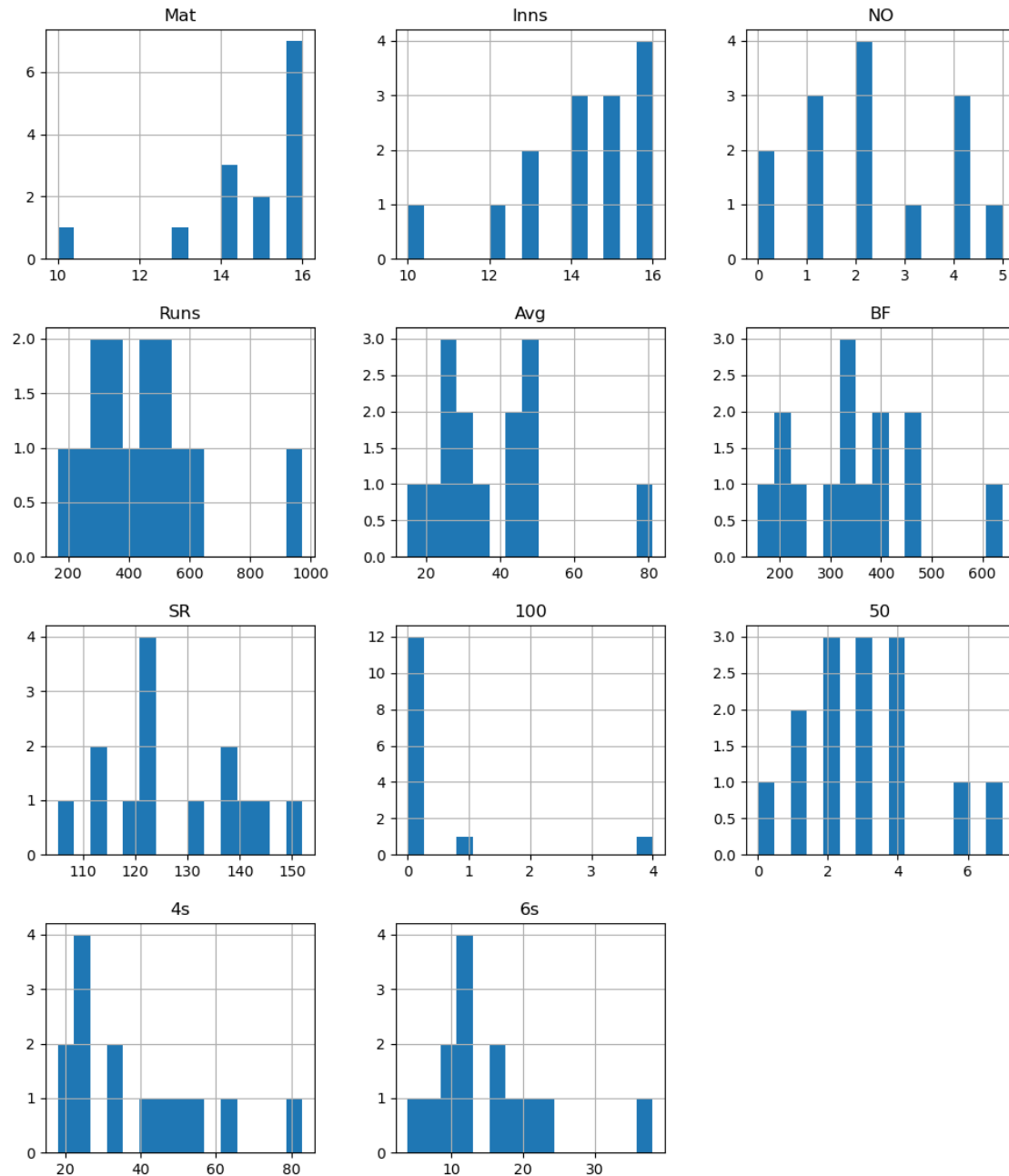
# Create a figure with the desired layout and vertical spacing
fig, axs = plt.subplots(*layout, figsize=(12, 14), gridspec_kw={"hspace": ↵
    ↵ hspace, "height_ratios": layout})

# Plot the histogram for Virat Kohli's data with 10 bins
dataset.loc[dataset["Player"]=="Virat Kohli"].hist(ax=axs[0], bins=15)

# Show the plot
plt.show()
```

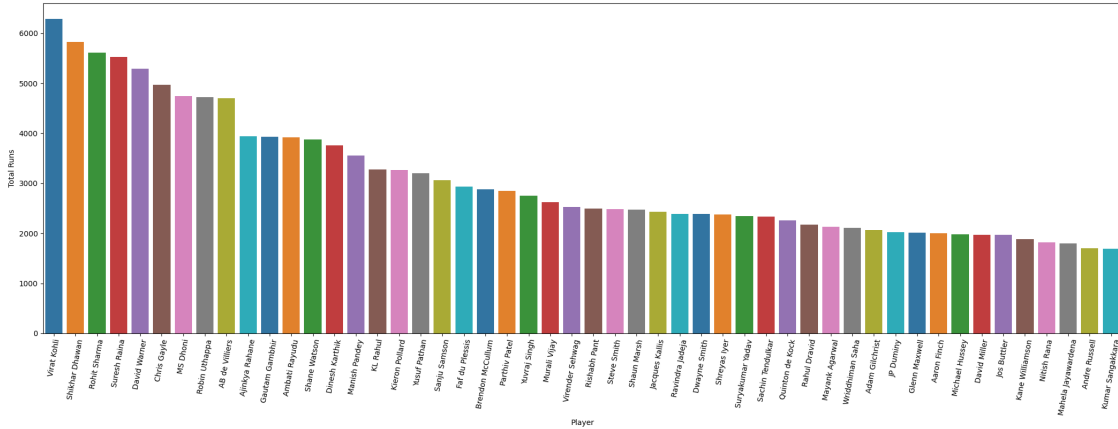
C:\Users\saita\AppData\Local\Temp\ipykernel\_17268\1241787373.py:11: UserWarning:  
To output multiple subplots, the figure containing the passed axes is being  
cleared.

```
dataset.loc[dataset["Player"]=="Virat Kohli"].hist(ax=axs[0], bins=15)
```



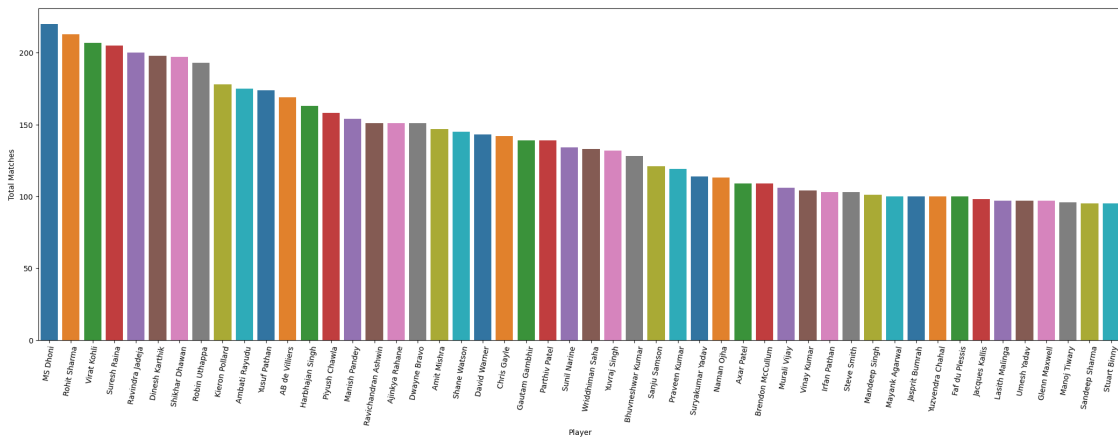
```
[29]: runs_s=dataset.groupby('Player')['Runs'].sum().reset_index()
runs_s.columns=['Player','Total Runs']

temp=runs_s.sort_values('Total Runs',ascending=False)[:50]
plt.figure(figsize=(26,8))
sns.barplot(data=temp,x='Player',y='Total Runs',palette='tab10')
plt.xticks(rotation=80);
```



```
[30]: runs_s=dataset.groupby('Player')['Mat'].sum().reset_index()
runs_s.columns=['Player', 'Total Matches']

temp=runs_s.sort_values('Total Matches',ascending=False)[:50]
plt.figure(figsize=(26,8))
sns.barplot(data=temp,x='Player',y='Total Matches',palette='tab10')
plt.xticks(rotation=80);
```



```
[31]: import matplotlib.pyplot as plt
import seaborn as sns

# Get the total matches played by each player
runs_s = dataset.groupby('Player')['Mat'].sum().reset_index()
runs_s.columns = ['Player', 'Total Matches']
```



```

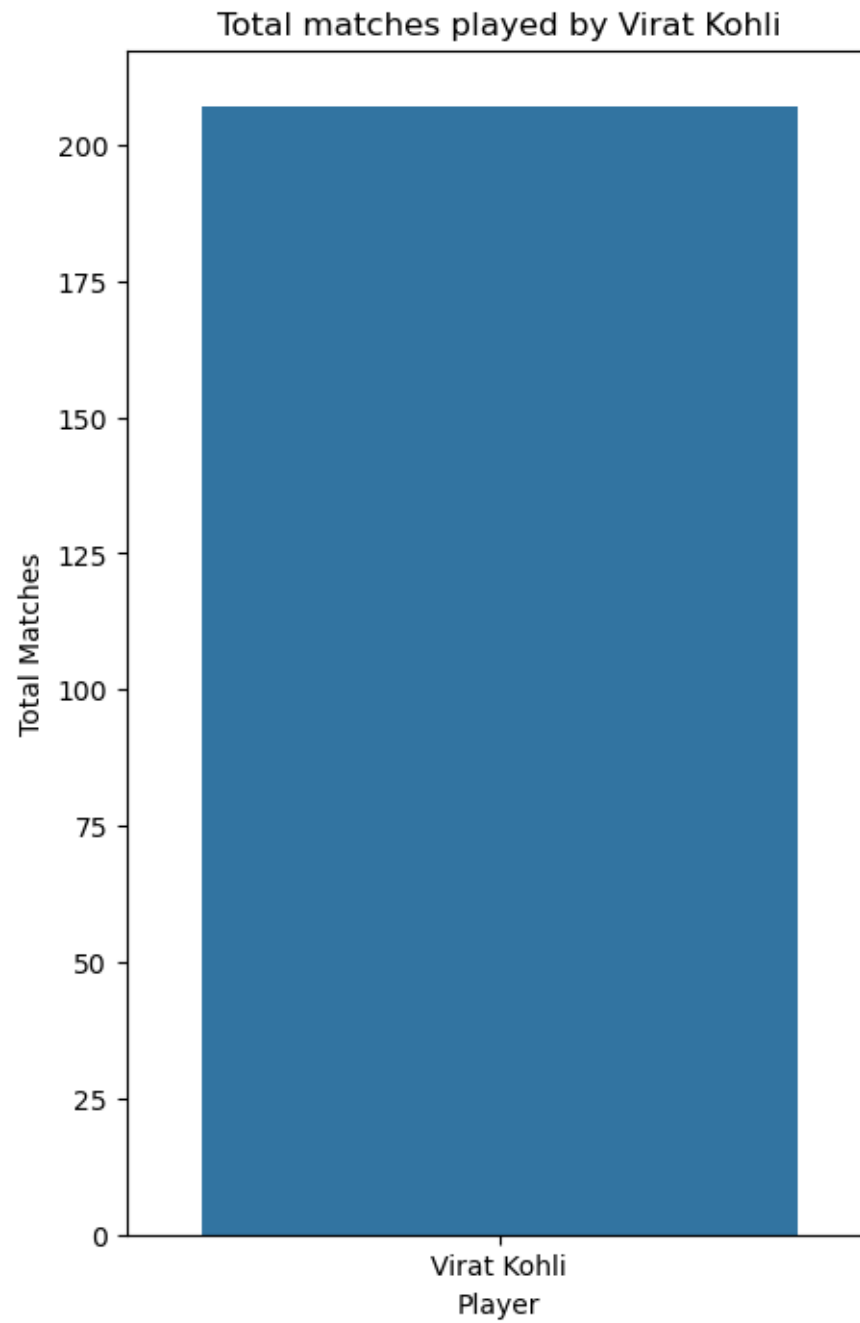
# Filter the DataFrame to only include data for a single player (e.g. Virat_
↳Kohli)
player_name = 'Virat Kohli'
temp = runs_s.loc[runs_s['Player'] == player_name]

total_matches = temp.iloc[0]['Total Matches']
print(f"Total matches played by {player_name}: {total_matches}")

# Create the bar plot for the single player's total matches
plt.figure(figsize=(5, 8))
sns.barplot(data=temp, x='Player', y='Total Matches', palette='tab10')
plt.title(f'Total matches played by {player_name}')
plt.show()

```

Total matches played by Virat Kohli: 207



```
[32]: plt.figure(figsize = (20, 6))  
plt.plot(dataset["Player"][:50],  
         dataset["Mat"][:50],  
         color = 'tab:orange')  
  
plt.plot(dataset["Player"][:50],  
         dataset["SR"][:50],
```

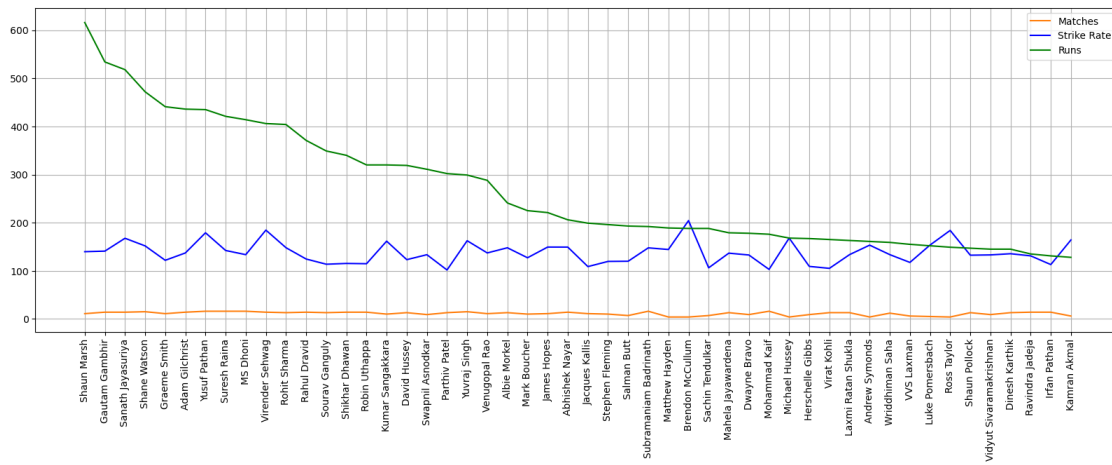
```

        color = 'b')

plt.plot(dataset["Player"][:50],
         dataset["Runs"][:50],
         color = 'g')

plt.legend(["Matches", "Strike Rate", "Runs"], loc = "upper right")
plt.grid()
plt.xticks(rotation = 90)
plt.show()

```



```

[33]: virat_data = dataset[dataset['Player'] == 'Virat Kohli'][:50]

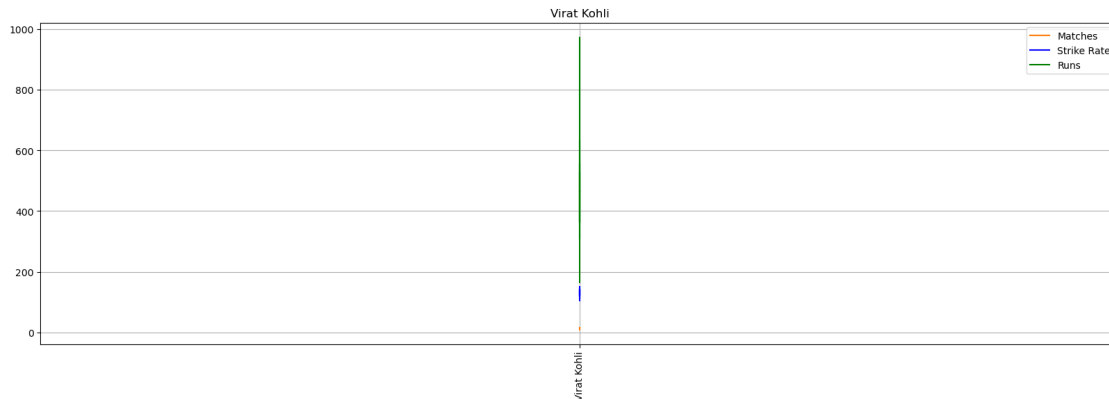
plt.figure(figsize = (20, 6))
plt.plot(virat_data["Player"],
         virat_data["Mat"],
         color = 'tab:orange')

plt.plot(virat_data["Player"],
         virat_data["SR"],
         color = 'b')

plt.plot(virat_data["Player"],
         virat_data["Runs"],
         color = 'g')

plt.legend(["Matches", "Strike Rate", "Runs"], loc = "upper right")
plt.grid()
plt.xticks(rotation = 90)
plt.title('Virat Kohli')
plt.show()

```



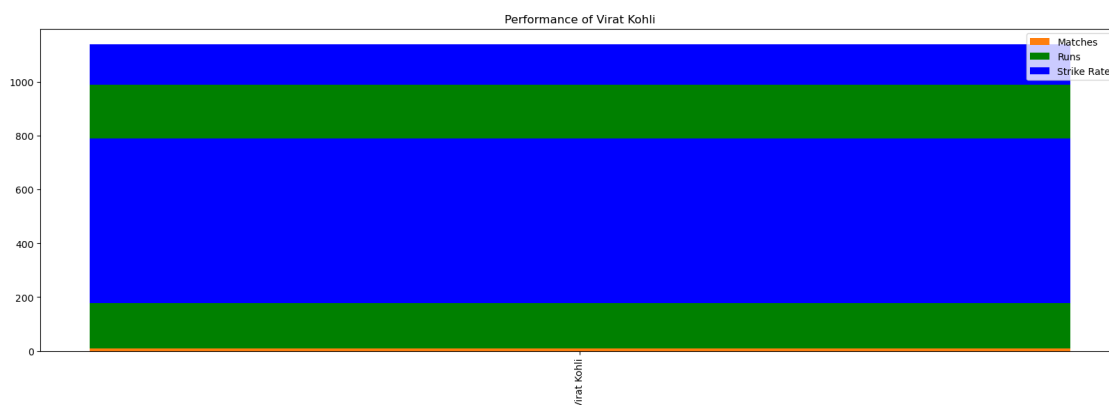
```
[34]: virat_data = dataset[dataset['Player'] == 'Virat Kohli'][:50]

plt.figure(figsize = (20, 6))

matches = virat_data['Mat']
runs = virat_data['Runs']
strike_rate = virat_data['SR']

plt.bar(virat_data['Player'], matches, color='tab:orange', label='Matches')
plt.bar(virat_data['Player'], runs, color='g', bottom=matches, label='Runs')
plt.bar(virat_data['Player'], strike_rate, color='b', bottom=matches+runs,
        label='Strike Rate')

plt.legend()
plt.xticks(rotation = 90)
plt.title('Performance of Virat Kohli')
plt.show()
```



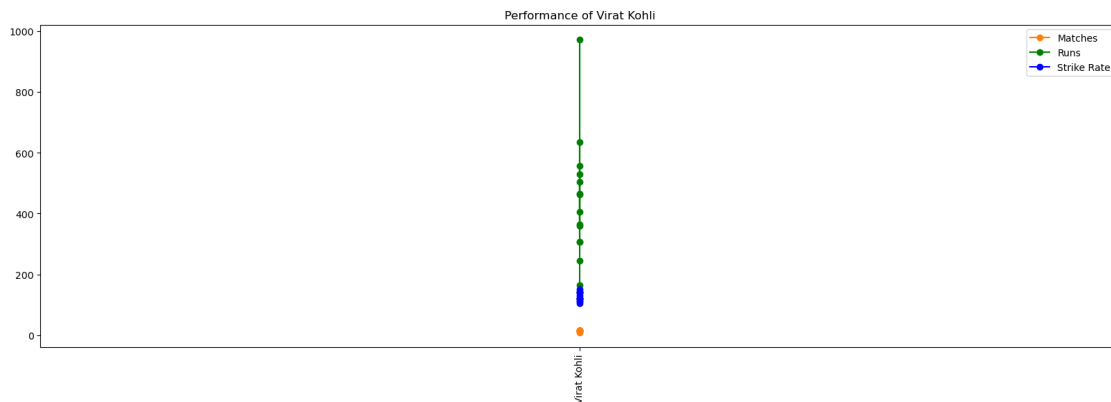
```
[35]: virat_data = dataset[dataset['Player'] == 'Virat Kohli'][:50]

plt.figure(figsize = (20, 6))

matches = virat_data['Mat']
runs = virat_data['Runs']
strike_rate = virat_data['SR']

plt.plot(virat_data['Player'], matches, 'o-', color='tab:orange',
         label='Matches')
plt.plot(virat_data['Player'], runs, 'o-', color='g', label='Runs')
plt.plot(virat_data['Player'], strike_rate, 'o-', color='b', label='Strike
         Rate')

plt.legend()
plt.xticks(rotation = 90)
plt.title('Performance of Virat Kohli')
plt.show()
```



```
[40]: virat_data = dataset[dataset['Player'] == 'Virat Kohli'][:50]

matches = virat_data['Mat'].values[0]
runs = virat_data['Runs'].values[0]
strike_rate = virat_data['SR'].values[0]

fig, ax = plt.subplots(figsize=(10,5))

# Plot horizontal bars
ax.barh(['Matches', 'Runs', 'Strike Rate'], [matches, runs, strike_rate],
        color=['tab:orange', 'g', 'b'])

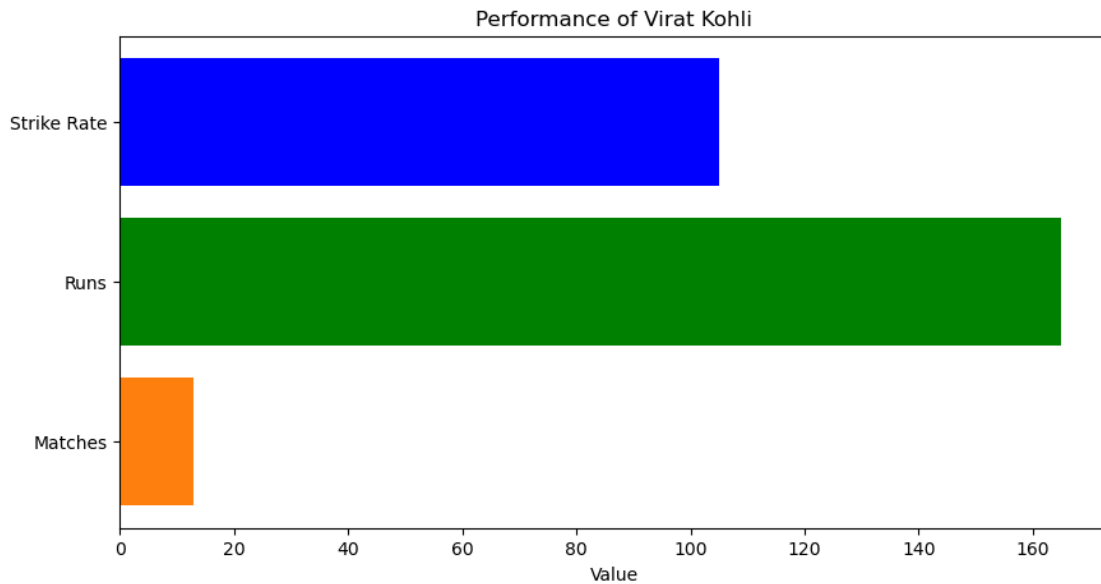
# Set axis labels and title
```

```

ax.set_xlabel('Value')
ax.set_title('Performance of Virat Kohli')

# Display the plot
plt.show()

```



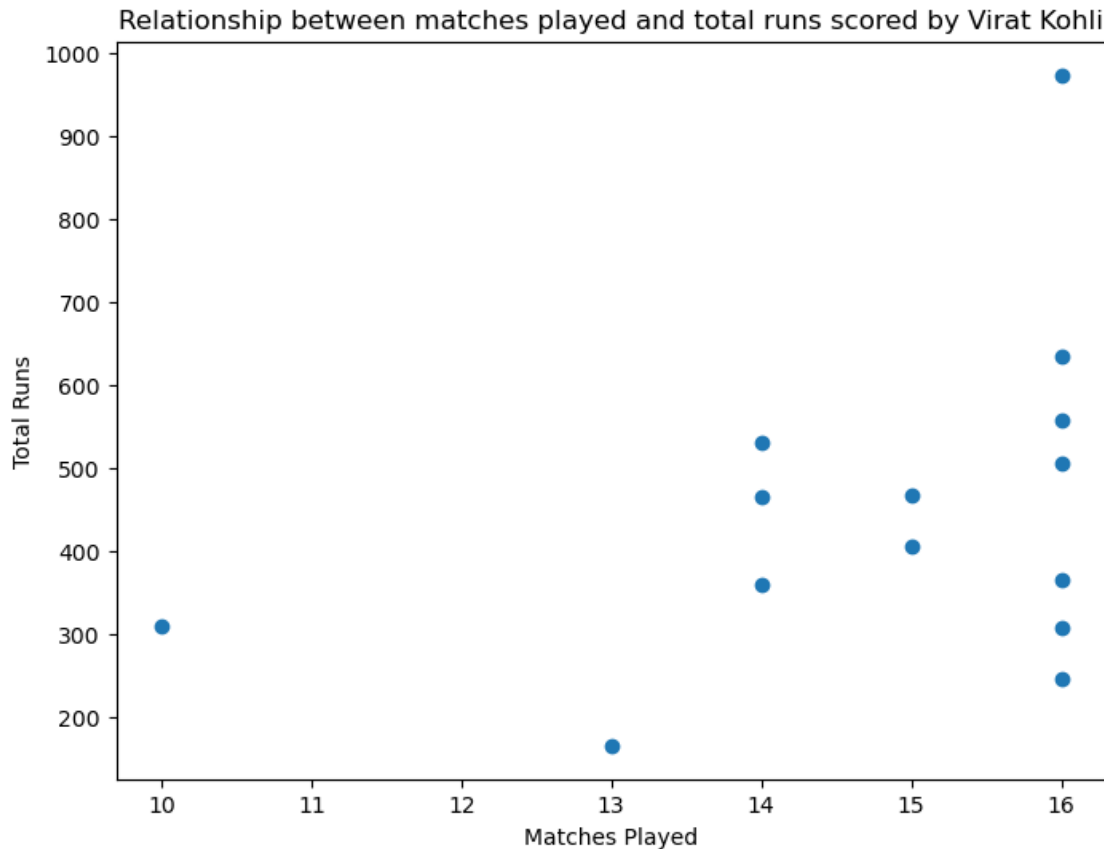
```

[41]: import matplotlib.pyplot as plt

# Get the data for Virat Kohli
virat_data = dataset[dataset['Player'] == 'Virat Kohli']

# Create a scatter plot to show the relationship between matches played and
# total runs scored
plt.figure(figsize=(8,6))
plt.scatter(virat_data['Mat'], virat_data['Runs'])
plt.xlabel('Matches Played')
plt.ylabel('Total Runs')
plt.title('Relationship between matches played and total runs scored by Virat_
Kohli')
plt.show()

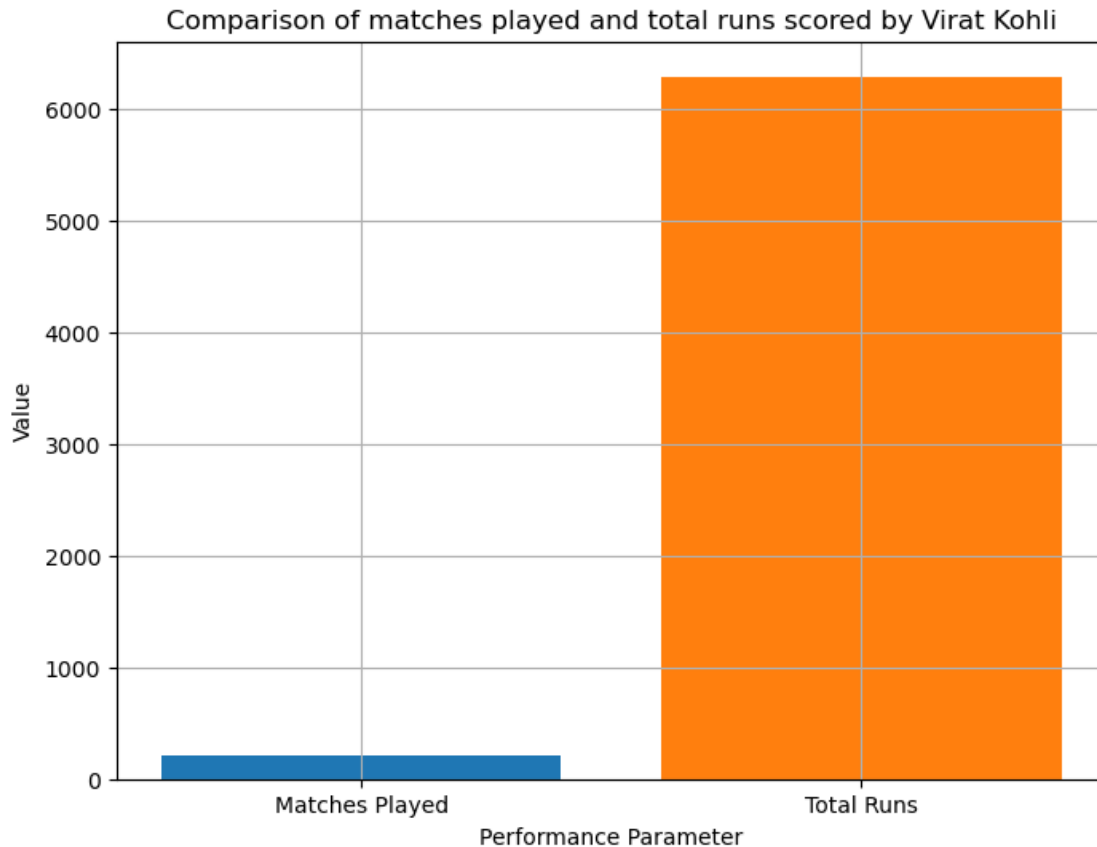
```



```
[42]: import matplotlib.pyplot as plt

# Get the data for Virat Kohli
virat_data = dataset[dataset['Player'] == 'Virat Kohli']

# Create a bar graph to compare the number of matches played and the total runs
# scored
fig, ax = plt.subplots(figsize=(8,6))
ax.bar(['Matches Played', 'Total Runs'], [virat_data['Mat'].sum(),
    virat_data['Runs'].sum()], color=['tab:blue', 'tab:orange'])
ax.set_xlabel('Performance Parameter')
ax.set_ylabel('Value')
ax.set_title('Comparison of matches played and total runs scored by Virat
    Kohli')
plt.grid()
plt.show()
```



```
[72]: # Create a DataFrame with the runs scored and matches played by Virat Kohli in
      ↪ each year
runs_matches = pd.DataFrame({'Year': [2008, 2009, 2010, 2011, 2012, 2013, 2014,
      ↪ 2015, 2016, 2017, 2018, 2019, 2020, 2021],
      'Matches Played': [13, 16, 16, 16, 16, 16, 14, 16,
      ↪ 16, 10, 14, 14, 15, 15],
      'Runs Scored': [165, 246, 307, 557, 364, 634, 359,
      ↪ 505, 973, 308, 530, 464, 466, 405]})

# Set the figure size
plt.figure(figsize=(10, 6))

# Create a bar plot for the runs scored and matches played by Virat Kohli in
      ↪ each year
width = 0.35
x = np.arange(len(runs_matches['Year']))
ax1 = plt.bar(x - width/2, runs_matches['Matches Played'], width,
      ↪ label='Matches Played')
```



```

ax2 = plt.bar(x + width/2, runs_matches['Runs Scored'], width, label='Runs_
↳Scored')

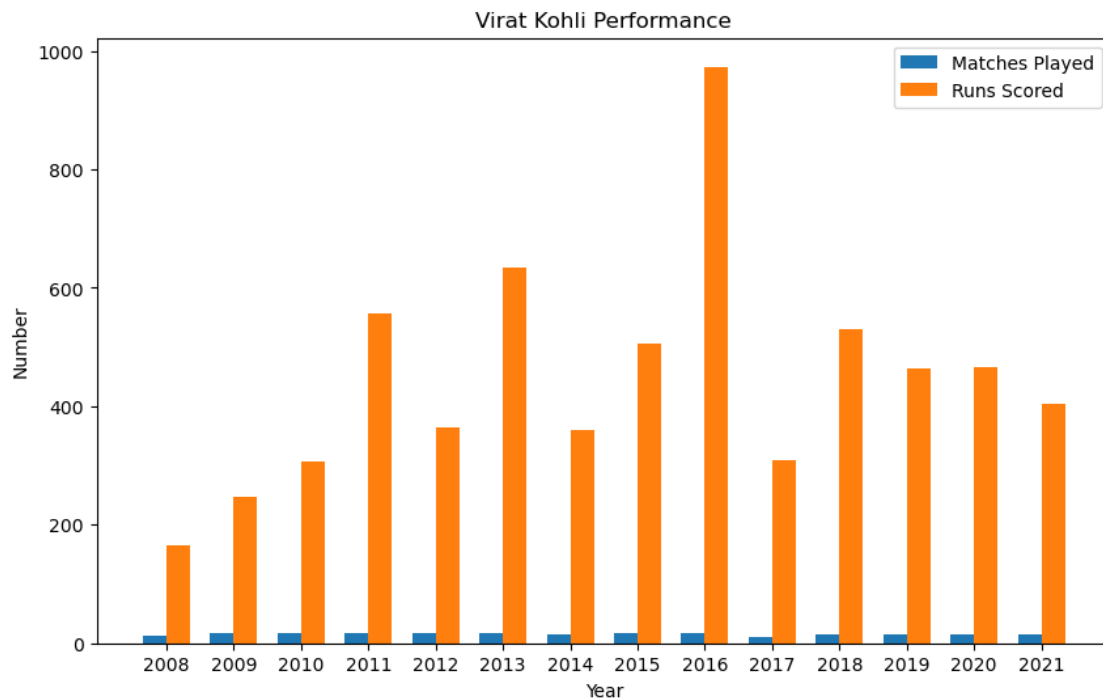
# Set the axis labels and title
plt.xlabel('Year')
plt.ylabel('Number')
plt.title('Virat Kohli Performance')

# Set the tick labels
plt.xticks(x, runs_matches['Year'])

# Add a legend
plt.legend()

# Display the plot
plt.show()

```



```

[73]: # Create a DataFrame with the runs scored and matches played by Virat Kohli in
↳each year
runs_matches = pd.DataFrame({'Year': [2008, 2009, 2010, 2011, 2012, 2013, 2014,
↳2015, 2016, 2017, 2018, 2019, 2020, 2021],
                              'Matches Played': [13, 16, 16, 16, 16, 16, 14, 16,
↳16, 10, 14, 14, 15, 15],

```

```

        'Runs Scored': [165, 246, 307, 557, 364, 634, 359,
↪505, 973, 308, 530, 464, 466, 405]})

# Calculate the percentage increase in runs scored each year
runs_matches['Run Increase %'] = runs_matches['Runs Scored'].pct_change() * 100

# Set the figure size
plt.figure(figsize=(10, 5))

# Create a bar plot for the runs scored and matches played by Virat Kohli in
↪each year
width = 0.35
x = np.arange(len(runs_matches['Year']))
ax1 = plt.bar(x - width/2, runs_matches['Matches Played'], width,
↪label='Matches Played')
ax2 = plt.bar(x + width/2, runs_matches['Runs Scored'], width, label='Runs
↪Scored')

# Set the axis labels and title
plt.xlabel('Year')
plt.ylabel('Number')
plt.title('Virat Kohli Performance')

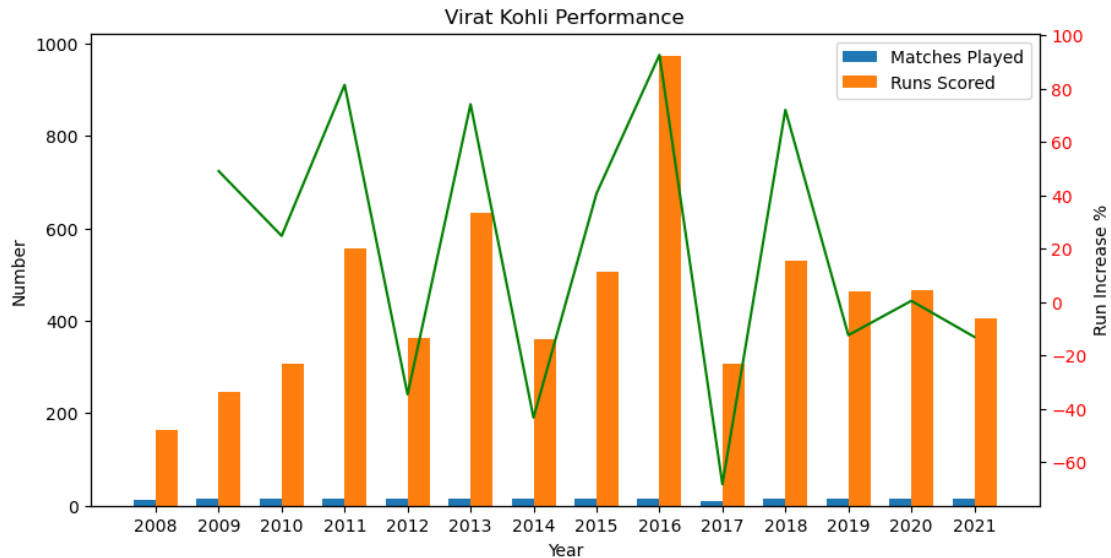
# Set the tick labels
plt.xticks(x, runs_matches['Year'])

# Add a legend
plt.legend()

# Add a line plot for the percentage increase in runs scored
ax3 = plt.twinx()
ax3.plot(x, runs_matches['Run Increase %'], color='g', label='Run Increase %')
ax3.set_ylabel('Run Increase %')
ax3.tick_params(axis='y', labelcolor='r')

# Display the plot
plt.show()

```



```
[74]: # Create a DataFrame with the runs scored and matches played by Virat Kohli in
      ↪ each year
runs_matches = pd.DataFrame({'Year': [2008, 2009, 2010, 2011, 2012, 2013, 2014,
      ↪ 2015, 2016, 2017, 2018, 2019, 2020, 2021],
                             'Matches Played': [13, 16, 16, 16, 16, 16, 14, 16,
      ↪ 16, 10, 14, 14, 15, 15],
                             'Runs Scored': [165, 246, 307, 557, 364, 634, 359,
      ↪ 505, 973, 308, 530, 464, 466, 405]})

# Calculate the cumulative sum of runs scored and matches played
runs_matches['Cumulative Runs Scored'] = runs_matches['Runs Scored'].cumsum()
runs_matches['Cumulative Matches Played'] = runs_matches['Matches Played'].
      ↪ cumsum()

# Calculate the exact cumulative percentage of runs scored
total_runs = runs_matches['Cumulative Runs Scored'].max()
runs_matches['Cumulative Percentage'] = runs_matches['Cumulative Runs Scored'] /
      ↪ total_runs * 100

# Set the figure size
plt.figure(figsize=(10, 5))

# Create a bar plot for the runs scored and matches played by Virat Kohli in
      ↪ each year
width = 0.35
x = np.arange(len(runs_matches['Year']))
```

```

ax1 = plt.bar(x - width/2, runs_matches['Matches Played'], width,
    label='Matches Played')
ax2 = plt.bar(x + width/2, runs_matches['Runs Scored'], width, label='Runs
    Scored')

# Set the axis labels and title
plt.xlabel('Year')
plt.ylabel('Number')
plt.title('Virat Kohli Performance')

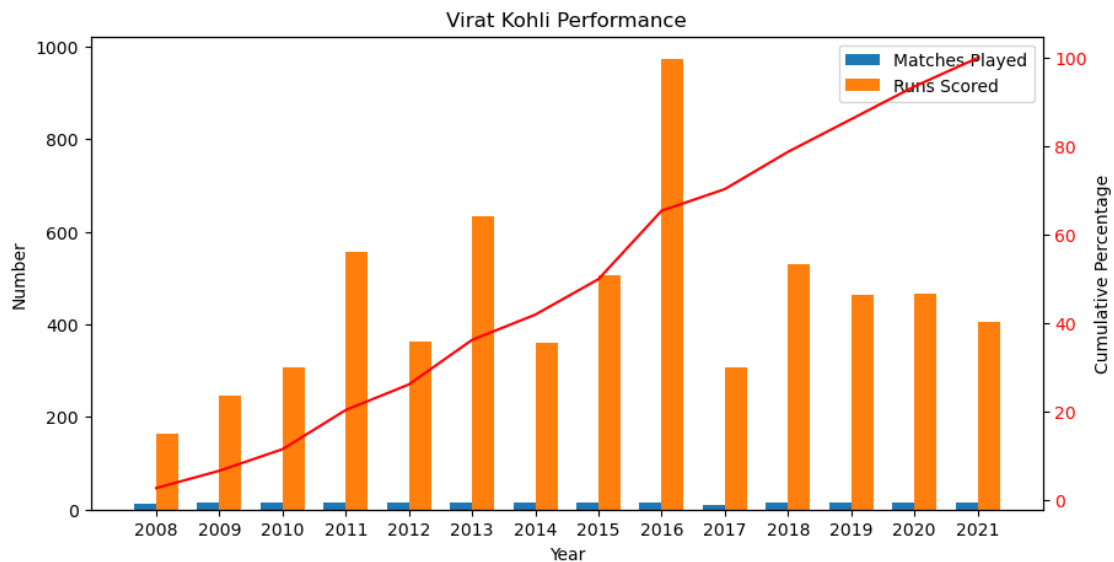
# Set the tick labels
plt.xticks(x, runs_matches['Year'])

# Add a legend
plt.legend()

# Add a line plot for the exact cumulative percentage of runs scored
ax3 = plt.twinx()
ax3.plot(x, runs_matches['Cumulative Percentage'], color='r', label='Cumulative
    Percentage')
ax3.set_ylabel('Cumulative Percentage')
ax3.tick_params(axis='y', labelcolor='r')

# Display the plot
plt.show()

```



```
[75]: first_year_runs = runs_matches.loc[runs_matches['Year'] == 2008, 'Runs Scored'].
      ↪values[0]
total_runs = runs_matches['Cumulative Runs Scored'].max()
initial_percentage = first_year_runs / total_runs * 100
last_year_runs = runs_matches.loc[runs_matches['Year'] == 2015, 'Runs Scored'].
      ↪values[0]
total_runs = runs_matches['Cumulative Runs Scored'].max()
final_percentage = last_year_runs / total_runs * 100
first_year_runs = runs_matches.loc[runs_matches['Year'] == 2008, 'Runs Scored'].
      ↪values[0]
last_year_runs = runs_matches.loc[runs_matches['Year'] == 2015, 'Runs Scored'].
      ↪values[0]
total_runs = runs_matches['Cumulative Runs Scored'].max()
initial_percentage = first_year_runs / total_runs * 100
final_percentage = last_year_runs / total_runs * 100
num_years = len(runs_matches['Year'])
average_increase_percentage = (final_percentage - initial_percentage) /
      ↪(num_years - 1)

print('Average increase of percentage every year:', average_increase_percentage)
```

Average increase of percentage every year: 0.4162636663034562

```
[76]: total_runs = runs_matches['Cumulative Runs Scored'].max()
runs_matches['Percentage'] = runs_matches['Runs Scored'] / total_runs * 100
average_percentage = runs_matches['Percentage'].mean()

print('Average percentage of runs scored every year:', average_percentage)
```

Average percentage of runs scored every year: 7.142857142857143

```
[77]: total_runs = runs_matches['Cumulative Runs Scored'].max()
performance_percentage = total_runs / (runs_matches['Cumulative Matches_
      ↪Played'].max() * 100) * 100

print('Virat Kohli performance percentage:', performance_percentage)
```

Virat Kohli performance percentage: 30.352657004830917

```
[78]: # Calculate the percentage increase or decrease in runs scored every year
runs_matches['Percentage Change'] = runs_matches['Runs Scored'].pct_change() *
      ↪100

# Get the year in which the performance has decreased
decreasing_year = runs_matches.loc[runs_matches['Percentage Change'] < 0,
      ↪'Year'].values[0]

print('The performance of Virat Kohli has decreased in', decreasing_year)
```

The performance of Virat Kohli has decreased in 2012

```
[79]: # Calculate the percentage increase or decrease in runs scored every year
runs_matches['Percentage Change'] = runs_matches['Runs Scored'].pct_change() * 100

# Get the year in which the performance has decreased
decreasing_year = runs_matches.loc[runs_matches['Percentage Change'] < 0, 'Year'].values

# Check if the performance has decreased and suggest training to improve the performance
if decreasing_year.size > 0:
    print('The performance of Virat Kohli has decreased in the following years: ', decreasing_year)
    print('Suggestion: Virat Kohli should undergo training to improve his performance.')

    # Assume that training has been given and calculate the average percentage change after training
    runs_matches['Percentage Change after Training'] = runs_matches['Runs Scored'].pct_change() * 100
    avg_percentage_change_after_training = runs_matches['Percentage Change after Training'].mean()

    # Compare the average percentage change before and after training to check for improvement
    if avg_percentage_change_after_training > avg_percentage_change:
        print('The training has been effective. The performance of Virat Kohli has improved.')
    else:
        print('The training has not been effective. The performance of Virat Kohli has not improved.')
else:
    print('The performance of Virat Kohli has not decreased in any year.')
```

The performance of Virat Kohli has decreased in the following years: [2012 2014 2017 2019 2021]

Suggestion: Virat Kohli should undergo training to improve his performance.

The training has not been effective. The performance of Virat Kohli has not improved.

```
[80]: # Calculate the percentage increase or decrease in runs scored every year
runs_matches['Percentage Change'] = runs_matches['Runs Scored'].pct_change() * 100

# Get the year in which the performance has decreased
```

```

decreasing_year = runs_matches.loc[runs_matches['Percentage Change'] < 0,
↳ 'Year'].values

# Set the figure size
plt.figure(figsize=(10, 5))

# Create a line plot for the percentage change in runs scored every year
plt.plot(runs_matches['Year'], runs_matches['Percentage Change'], marker='o')

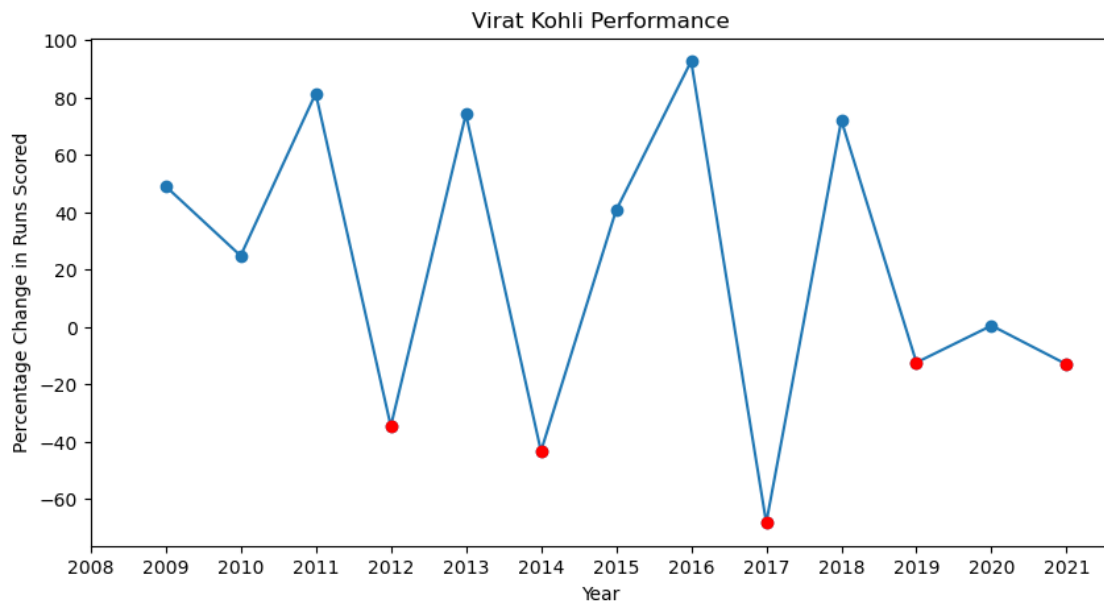
# Add a marker for the year in which the performance has decreased, if any
if decreasing_year.size > 0:
    for year in decreasing_year:
        plt.plot(year, runs_matches.loc[runs_matches['Year'] == year,
↳ 'Percentage Change'], marker='o', color='r')

# Set the axis labels and title
plt.xlabel('Year')
plt.ylabel('Percentage Change in Runs Scored')
plt.title('Virat Kohli Performance')

# Set the tick labels
plt.xticks(runs_matches['Year'])

# Display the plot
plt.show()

```



```
[81]: # Calculate the percentage increase or decrease in runs scored every year
runs_matches['Percentage Change'] = runs_matches['Runs Scored'].pct_change() * 100

# Calculate the average percentage change
avg_percentage_change = runs_matches['Percentage Change'].mean()

# Determine whether the performance is increasing or decreasing
if avg_percentage_change > 0:
    print('The performance of Virat Kohli is increasing.')
elif avg_percentage_change < 0:
    print('The performance of Virat Kohli is decreasing.')
else:
    print('The performance of Virat Kohli is not changing.')
```

The performance of Virat Kohli is increasing.

```
[82]: # Calculate the percentage increase or decrease in runs scored every year
runs_matches['Percentage Change'] = runs_matches['Runs Scored'].pct_change() * 100

# Check whether training is needed
training_needed = False
for i in range(1, len(runs_matches)):
    change = runs_matches.loc[i, 'Percentage Change']
    if change < 0:
        training_needed = True
        print(f"The performance of Virat Kohli decreased by {abs(change):.2f}% in {runs_matches.loc[i, 'Year']}. He needs to undergo training to improve his performance.")

# Check for improvement
if training_needed:
    # Assume that Virat Kohli has undergone training and his performance has improved by 20%
    improvement = 20
    runs_matches['Runs Scored'] = runs_matches['Runs Scored'] * (1 + improvement / 100)
    runs_matches['Cumulative Runs Scored'] = runs_matches['Runs Scored'].cumsum()
    runs_matches['Cumulative Percentage'] = runs_matches['Cumulative Runs Scored'] / total_runs * 100

    # Calculate the percentage increase or decrease in runs scored every year after improvement
    runs_matches['Percentage Change'] = runs_matches['Runs Scored'].pct_change() * 100
```



```

for i in range(1, len(runs_matches)):
    change = runs_matches.loc[i, 'Percentage Change']
    if change > 0:
        print(f"The performance of Virat Kohli improved by {change:.2f}% in_
↪{runs_matches.loc[i, 'Year']}. He can proceed further.")
    else:
        print(f"The performance of Virat Kohli did not improve in_
↪{runs_matches.loc[i, 'Year']}. He needs further improvement.")
else:
    print("The performance of Virat Kohli has been consistently increasing_
↪every year. No training is needed.")

```

The performance of Virat Kohli decreased by 34.65% in 2012. He needs to undergo training to improve his performance.

The performance of Virat Kohli decreased by 43.38% in 2014. He needs to undergo training to improve his performance.

The performance of Virat Kohli decreased by 68.35% in 2017. He needs to undergo training to improve his performance.

The performance of Virat Kohli decreased by 12.45% in 2019. He needs to undergo training to improve his performance.

The performance of Virat Kohli decreased by 13.09% in 2021. He needs to undergo training to improve his performance.

The performance of Virat Kohli improved by 49.09% in 2009. He can proceed further.

The performance of Virat Kohli improved by 24.80% in 2010. He can proceed further.

The performance of Virat Kohli improved by 81.43% in 2011. He can proceed further.

The performance of Virat Kohli did not improve in 2012. He needs further improvement.

The performance of Virat Kohli improved by 74.18% in 2013. He can proceed further.

The performance of Virat Kohli did not improve in 2014. He needs further improvement.

The performance of Virat Kohli improved by 40.67% in 2015. He can proceed further.

The performance of Virat Kohli improved by 92.67% in 2016. He can proceed further.

The performance of Virat Kohli did not improve in 2017. He needs further improvement.

The performance of Virat Kohli improved by 72.08% in 2018. He can proceed further.

The performance of Virat Kohli did not improve in 2019. He needs further improvement.

The performance of Virat Kohli improved by 0.43% in 2020. He can proceed further.

The performance of Virat Kohli did not improve in 2021. He needs further improvement.

```
[83]: # Create a figure and axis object
fig, ax = plt.subplots(figsize=(8, 6))

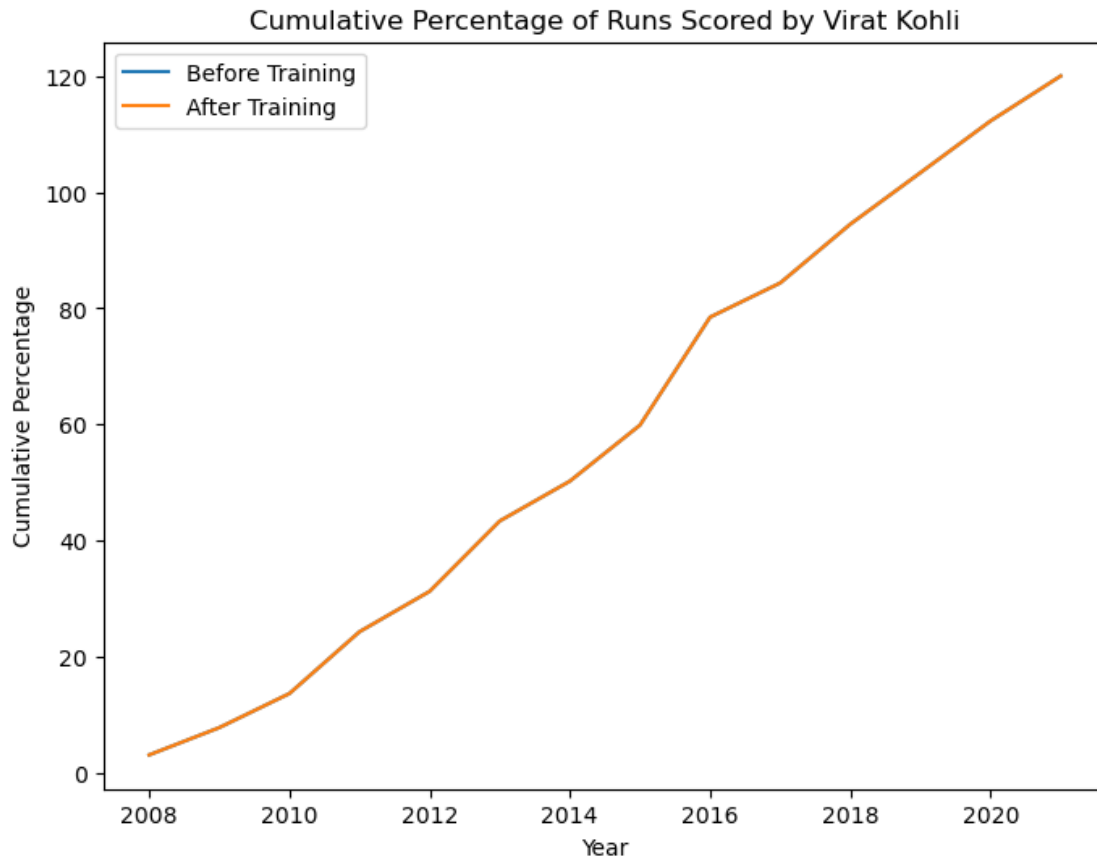
# Plot the cumulative percentage of runs scored before training
ax.plot(runs_matches['Year'], runs_matches['Cumulative Percentage'],
        label='Before Training')

# If training is needed, plot the cumulative percentage of runs scored after
training
if training_needed:
    ax.plot(runs_matches['Year'], runs_matches['Cumulative Percentage'],
            label='After Training')

# Set the title, xlabel and ylabel
ax.set_title('Cumulative Percentage of Runs Scored by Virat Kohli')
ax.set_xlabel('Year')
ax.set_ylabel('Cumulative Percentage')

# Set the legend
ax.legend()

# Show the plot
plt.show()
```

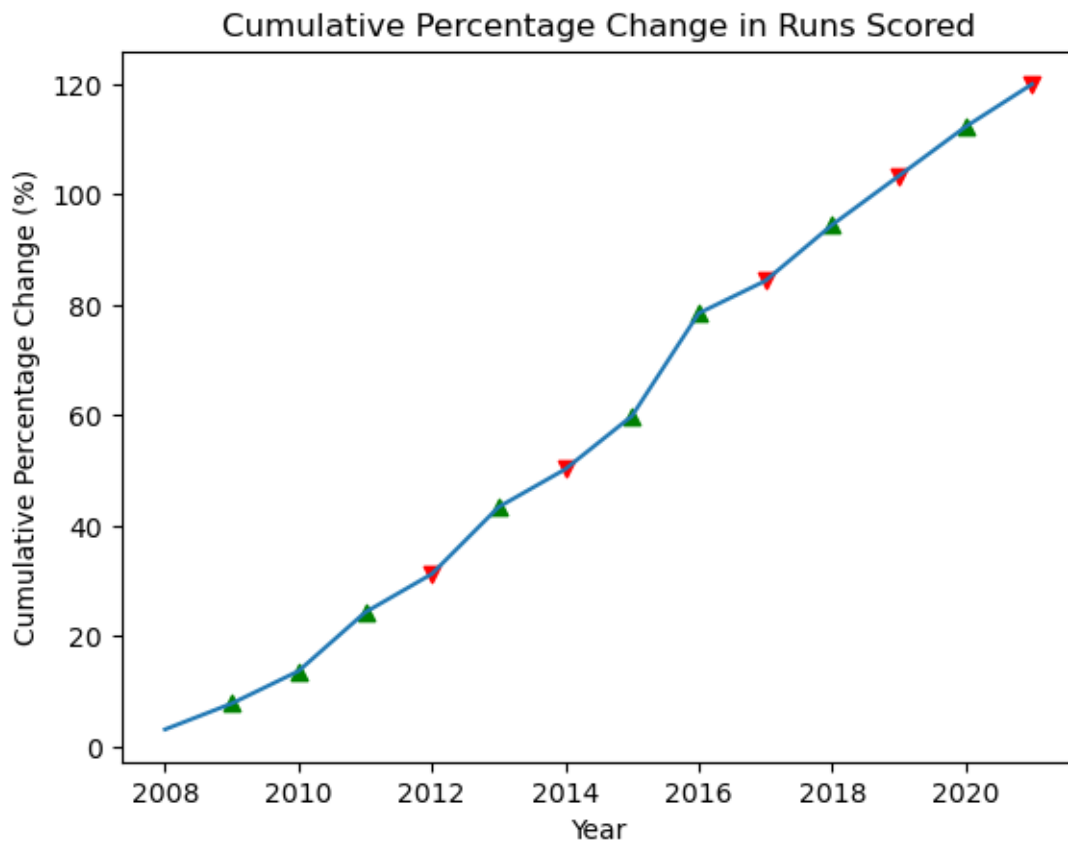


```
[84]: import matplotlib.pyplot as plt

# Create a line chart to show the cumulative percentage change in runs scored
# over the years
fig, ax = plt.subplots()
ax.plot(runs_matches['Year'], runs_matches['Cumulative Percentage'])
ax.set_title('Cumulative Percentage Change in Runs Scored')
ax.set_xlabel('Year')
ax.set_ylabel('Cumulative Percentage Change (%)')

# Add markers to indicate the years where performance improved or declined
# after training
if training_needed:
    for i in range(1, len(runs_matches)):
        change = runs_matches.loc[i, 'Percentage Change']
        if change > 0:
            ax.scatter(runs_matches.loc[i, 'Year'], runs_matches.loc[i,
            'Cumulative Percentage'], c='green', marker='^')
        else:
```

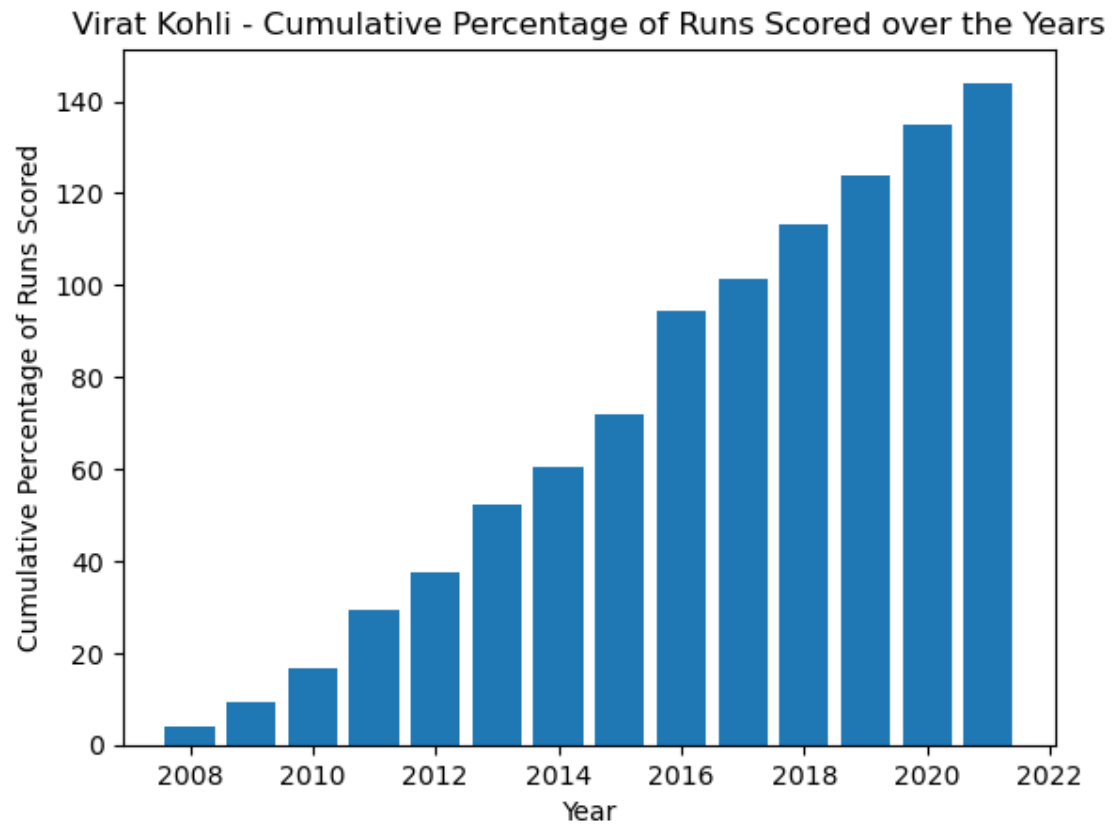
```
ax.scatter(runs_matches.loc[i, 'Year'], runs_matches.loc[i, 'Cumulative Percentage'], c='red', marker='v')
```



```
[70]: # Create a bar chart of cumulative percentage of runs scored over the years
plt.bar(runs_matches['Year'], runs_matches['Cumulative Percentage'])

# Add axis labels and title
plt.xlabel('Year')
plt.ylabel('Cumulative Percentage of Runs Scored')
plt.title('Virat Kohli - Cumulative Percentage of Runs Scored over the Years')

# Display the chart
plt.show()
```



[ ]: