

Cricket venues play a significant role in influencing a batsman's performance and overall gameplay. The characteristics of a cricket venue, such as the pitch conditions, weather, altitude, and even crowd support, can impact how well a batsman performs during a match. Let's delve into some key aspects of how cricket venues affect batsman performance:

Pitch Conditions:

The pitch is one of the most crucial factors that can heavily influence a batsman's performance. Pitches can vary greatly in terms of their hardness, grass cover, and moisture content. Some pitches may be batting-friendly, offering even bounce and little movement, allowing batsmen to play their shots with confidence. On the other hand, pitches with more grass and moisture can assist bowlers, causing uneven bounce and lateral movement, making it challenging for batsmen to time their shots accurately.

Surface and Bounce:

The nature of the surface—whether it's hard or soft—can impact the bounce of the ball. Batsmen accustomed to low bounce may struggle on pitches with extra bounce, and vice versa. Additionally, some venues have characteristics that lead to variable or inconsistent bounce, posing challenges for batsmen to adapt and adjust their shots.

Weather Conditions:

Weather conditions, including temperature, humidity, and wind, can significantly impact a batsman's ability to score runs. Overcast conditions and high humidity levels can assist swing and seam bowlers, while dry and sunny conditions might result in slower pitches and encourage spin bowlers.

Altitude and Air Density:

The altitude of a cricket venue can affect the air density, which, in turn, influences how the ball behaves in flight. At higher altitudes, the air is thinner, causing the ball to travel faster through the air. This can make it challenging for batsmen to judge the pace and movement of the ball accurately.

Crowd Support:

The home crowd can play a substantial role in boosting a batsman's confidence and performance. A supportive crowd can energize a batsman and create a positive atmosphere, enabling them to take more risks and play more confidently. On the contrary, a hostile or noisy crowd, especially when playing away from home, might lead to added pressure and distraction.

Ground Dimensions:

Different cricket venues have varying ground dimensions, such as the size of the playing area and the length of boundaries. Smaller grounds with shorter boundaries can be advantageous for batsmen, as well-timed shots have a higher chance of clearing the ropes. Conversely, larger grounds can make scoring boundaries more challenging.

Pitch History:

The historical performance of a venue can also impact a batsman's mindset. If a venue is known to favour certain types of bowlers or has a history of high-scoring matches, batsmen might adjust their strategies accordingly.















Pitch Wear and Tear:

As a match progresses, the pitch can wear and deteriorate, making it more challenging for batsmen to play their shots fluently. Batsmen need to be adaptable and recognize when the pitch is changing and how it affects shot selection.

In conclusion, cricket venues are more than just physical locations; they possess unique characteristics that influence the tactics, mindset, and adaptability of batsmen. Successful batsmen are those who can quickly assess the conditions, adjust their techniques, and capitalize on the advantages provided by a particular venue, while also overcoming its challenges. As such, an understanding of how cricket venues impact batsman performance is crucial for both players and fans alike.

```
In [ ]: from IPython.display import Image, display
```

```
In [ ]: image_path = "C:\\Users\\Mohamed\\Pictures\\Screenshots\\Screenshot 2023-08-18 122306.png"
display(Image(filename=image_path))
```

	TEAM	M	INN	R	4s	6s	HS	SR
 Shubman Gill		17	17	890	85	33	129	157.80
 Faf du Plessis		14	14	730	60	36	84	153.68
 Devon Conway		16	15	672	77	18	92*	139.70
 Virat Kohli		14	14	639	65	16	101*	139.82
 Yashasvi Jaiswal		14	14	625	82	26	124	163.61
 Suryakumar Yadav		16	16	605	65	28	103*	181.13
 Ruturaj Gaikwad		16	15	590	46	30	92	147.50

Here is a list of the 2023 IPL orange cap leaderboard players.

We employ a more conventional method of measuring that may be used to evaluate player performance and can mimic the course of the game in metrics.

But for a variety of reasons, these antiquated metrics appear to be out of date because actual performers go unrecognized.

For instance, an RCB player can participate in 7-8 matches on batting-friendly fields, where the average score on their home field is 20 runs per innings higher than the overall average score of the competition.

```
In [ ]: #importing IPL2023 Data
import pandas as pd
df=pd.read_csv("C:\\Users\\Mohamed\\Desktop\\data\\ipl_match_details.csv")
```

```
In [ ]: df
```

```
Out[ ]:
```

	Runs1	Wickets1	Overs1	Balls1	Runs2	Wickets2	Overs2	Balls2	Venue
0	178	7	20	NaN	182	5	19	2.0	Ahmedabad
1	191	5	20	NaN	146	7	16	NaN	Mohali
2	193	6	20	NaN	143	9	20	NaN	Lucknow
3	203	5	20	NaN	131	8	20	NaN	Hyderabad (Deccan)
4	171	7	20	NaN	172	2	16	2.0	Bengaluru
...
69	197	5	20	NaN	198	4	19	1.0	Bengaluru
70	172	7	20	NaN	157	10	20	NaN	Chennai
71	182	8	20	NaN	101	10	16	3.0	Chennai
72	233	3	20	NaN	171	10	18	2.0	Ahmedabad
73	214	4	20	NaN	171	5	15	NaN	Ahmedabad

74 rows × 9 columns

```
In [ ]: df["TotalRuns"] = df["Runs1"] + df["Runs2"]

df_filtered = df[(df["Runs1"] != 0) & (df["Runs2"] != 0)]
```

```
venue_runs_permatch = df.groupby("Venue")["TotalRuns"].mean()
```

```
In [ ]: venue_runs_permatch
```

```
Out[ ]: Venue
Ahmedabad      363.333333
Bengaluru      383.142857
Chennai        325.666667
Delhi          335.714286
Dharamsala     393.500000
Guwahati       365.000000
Hyderabad (Deccan) 335.714286
Jaipur         313.600000
Kolkata        365.000000
Lucknow        267.857143
Mohali         371.200000
Wankhede       391.000000
Name: TotalRuns, dtype: float64
```

```
In [ ]: sorted_venue_runs_permatch = venue_runs_permatch.sort_values(ascending=False)
```

```
In [ ]: sorted_venue_runs_permatch
```

```
Out[ ]: Venue
Dharamsala     393.500000
Wankhede       391.000000
Bengaluru      383.142857
Mohali         371.200000
Guwahati       365.000000
Kolkata        365.000000
Ahmedabad      363.333333
Delhi          335.714286
Hyderabad (Deccan) 335.714286
Chennai        325.666667
Jaipur         313.600000
Lucknow        267.857143
Name: TotalRuns, dtype: float64
```

We can accommodate a considerable shift in their disparities because we know the average score for each match in each venue here.

We must standardize the average to its highest level in order to undertake data analysis on removing the average difference.

```
In [ ]: df['venue_coefficient'] = 393.5 / df['TotalRuns']
```

```
In [ ]: venue_coefficient_grouped = df.groupby('Venue')['venue_coefficient'].mean()
```

```
In [ ]: venue_coefficient_grouped
```

```
Out[ ]: Venue
Ahmedabad      1.103380
Bengaluru      1.037893
Chennai        1.236150
Delhi          1.192990
Dharamsala     1.001982
Guwahati       1.082764
Hyderabad (Deccan) 1.186165
Jaipur         1.332493
Kolkata        1.093424
Lucknow        1.634406
Mohali         1.087642
Wankhede       1.015345
Name: venue_coefficient, dtype: float64
```

We have normalized the data for every venue; now we need to forecast the real performance of the players on the venue coefficient.

Here, 393.5 is the average score in Dharmashala, which is the highest in the competition.

```
In [ ]: df_gill=pd.read_csv("C:\\Users\\Mohamed\\Documents\\gill.csv")
df_gill_merged = df_gill.merge(venue_coefficient_grouped, left_on="Venue", right_index=True)
df_gill_merged["ModifiedRuns"] = df_gill_merged["Runs"] * df_gill_merged["venue_coefficient"]
modified_gill_runs = df_gill_merged["ModifiedRuns"].sum()
```

```
In [ ]: modified_gill_runs
```

```
Out[ ]: 988.2057971129713
```

The normalized total runs for Gill in IPL2023 are 988.2057971129713; however, his actual runs are 890.

```
In [ ]: df_faf=pd.read_csv("C:\\Users\\Mohamed\\Documents\\faf.csv")
df_faf_merged = df_faf.merge(venue_coefficient_grouped, left_on="Venue", right_index=True)
```

```
df_faf_merged["ModifiedRuns"] = df_faf_merged["Runs"] * df_faf_merged["venue_coefficient"]
modified_faf_runs = df_faf_merged["ModifiedRuns"].sum()
```

```
In [ ]: modified_faf_runs
```

```
Out[ ]: 821.6086434850725
```

The normalized total runs for Faf in IPL2023 are 821.6086434850725; however, his actual runs are 730.

```
df_conway=pd.read_csv("C:\\Users\\Mohamed\\Documents\\conway.csv")
df_conway_merged = df_conway.merge(venue_coefficient_grouped, left_on="Venue", right_index=True)
df_conway_merged["ModifiedRuns"] = df_conway_merged["Runs"] * df_conway_merged["venue_coefficient"]
modified_conway_runs = df_conway_merged["ModifiedRuns"].sum()
```

```
In [ ]: modified_conway_runs
```

```
Out[ ]: 796.8878139849838
```

The normalized total runs for Conway in IPL2023 are 796.8878139849838; however, his actual runs are 672.

```
df_virat=pd.read_csv("C:\\Users\\Mohamed\\Documents\\virat.csv")
df_virat_merged = df_virat.merge(venue_coefficient_grouped, left_on="Venue", right_index=True)
df_virat_merged["ModifiedRuns"] = df_virat_merged["Runs"] * df_virat_merged["venue_coefficient"]
modified_virat_runs = df_virat_merged["ModifiedRuns"].sum()
```

```
In [ ]: modified_virat_runs
```

```
Out[ ]: 714.4446547524124
```

The normalized total runs for Virat in IPL2023 are 714.4446547524124; however, his actual runs are 639.

```
df_yash=pd.read_csv("C:\\Users\\Mohamed\\Documents\\yash.csv")
df_yash_merged = df_yash.merge(venue_coefficient_grouped, left_on="Venue", right_index=True)
df_yash_merged["ModifiedRuns"] = df_yash_merged["Runs"] * df_yash_merged["venue_coefficient"]
modified_yash_runs = df_yash_merged["ModifiedRuns"].sum()
```

```
In [ ]: modified_yash_runs
```

```
Out[ ]: 712.8562211221304
```

The normalized total runs for Yash in IPL2023 are 712.8562211221304; however, his actual runs are 625.

```
df_sky=pd.read_csv("C:\\Users\\Mohamed\\Documents\\sky.csv")
df_sky_merged = df_sky.merge(venue_coefficient_grouped, left_on="Venue", right_index=True)
df_sky_merged["ModifiedRuns"] = df_sky_merged["Runs"] * df_sky_merged["venue_coefficient"]
modified_sky_runs = df_sky_merged["ModifiedRuns"].sum()
```

```
In [ ]: modified_sky_runs
```

```
Out[ ]: 645.3452004137882
```

The normalized total runs for Suryakumar in IPL2023 are 645.3452004137882; however, his actual runs are 605.

```
df_rutu=pd.read_csv("C:\\Users\\Mohamed\\Documents\\rutu.csv")
df_rutu_merged = df_rutu.merge(venue_coefficient_grouped, left_on="Venue", right_index=True)
df_rutu_merged["ModifiedRuns"] = df_rutu_merged["Runs"] * df_rutu_merged["venue_coefficient"]
modified_rutu_runs = df_rutu_merged["ModifiedRuns"].sum()
```

```
In [ ]: modified_rutu_runs
```

```
Out[ ]: 700.3578314954958
```

The normalized total runs for Ruturaj in IPL2023 are 700.3578314954958; however, his actual runs are 590.

```
data = {
    'NAME': ['GILL', 'FAF', 'CONWAY', 'VIRAT', 'YASH', 'SKY', 'RUTURAJ'],
    'ACTUAL RUNS': [890, 730, 672, 639, 625, 605, 590],
    'MODIFIED RUNS': [988.2057971129713, 821.6086434850725, 796.8878139849838, 714.4446547524124, 712.8562211221304]
}

df = pd.DataFrame(data)
```

```
In [ ]: df
```

Out[]:

	NAME	ACTUAL RUNS	MODIFIED RUNS
0	GILL	890	988.205797
1	FAF	730	821.608643
2	CONWAY	672	796.887814
3	VIRAT	639	714.444655
4	YASH	625	712.856221
5	SKY	605	645.345200
6	RUTURAJ	590	700.357831

In []:

```
df.columns
sorted_df = df.sort_values(by='MODIFIED RUNS', ascending=False)
```

In []:

```
sorted_df
```

Out[]:

	NAME	ACTUAL RUNS	MODIFIED RUNS
0	GILL	890	988.205797
1	FAF	730	821.608643
2	CONWAY	672	796.887814
3	VIRAT	639	714.444655
4	YASH	625	712.856221
6	RUTURAJ	590	700.357831
5	SKY	605	645.345200

Although there may not be a significant difference in the positions of the players, normalized numbers give a more true picture of their performances. For example, although Ruturaj is clearly ahead of Suryakumar in the list above by 55 runs, according to tournament metrics, he is actually 15 runs behind him.

In []:

```
df['RUNS DIFFERENCE'] = df['MODIFIED RUNS'] - df['ACTUAL RUNS']
```

In []:

```
df
```

Out[]:

	NAME	ACTUAL RUNS	MODIFIED RUNS	RUNS DIFFERENCE
0	GILL	890	988.205797	98.205797
1	FAF	730	821.608643	91.608643
2	CONWAY	672	796.887814	124.887814
3	VIRAT	639	714.444655	75.444655
4	YASH	625	712.856221	87.856221
5	SKY	605	645.345200	40.345200
6	RUTURAJ	590	700.357831	110.357831

Conway and Ruturaj have a greater run differential since they have played more matches on the typically slow Chepauk pitch.