

# Student's Activity Teacher's Copy: Runs by IPL teams in the tournament till 2020

In this activity, we will understand which team has been most productive on batting front

## Understanding the Dataset

Dataset consists of 2 columns or features namely `batting_team` and `total_runs`.

1. `batting_team` represents the team which was batting.
2. `total_runs` accounts for the runs attributed to the team for a particular ball.

## Understanding the Approach

1. Understanding the most productive team on batting front: Here, we will take into account the total runs scored by each team. The columns we will use are:

- A. `batting_team`
- B. `total_runs`

We will group the data on `batting_team` and plot a bar graph to interpret the total runs for each team.

## Importing Packages

Packages are imported in following manner.

```
import package_name
```

In the next cell, we have imported the following packages.

1. `pandas`. It is the most common library used by data scientists for data manipulation and cleaning
2. `numpy`. It adds support for arrays, along with a collection of mathematical functions to operate on these arrays.
3. `matplotlib`. It is a plotting library for python. `.pyplot` is a sub-package or set of functions available in `matplotlib` which we'll be using

`pd`, `np`, `plt` are all aliases for their corresponding packages. Alias are second name assigned to values or variables.

`%matplotlib inline` is a "magic function" renders plots

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

## Loading the Dataset

In the cell below, we have created a new pandas DataFrame by the name `df` and imported the mentioned file.

We have used `.head()` function to see the first 5 values of the dataset we created.

`.head()` can show up any number of values based on the parameter given.

If we want to see more, we can pass value in the function like `df.head(10)` will show first 10 values of the dataset

```
In [2]: df = pd.read_csv("https://raw.githubusercontent.com/jainharshit27/datasets/main/runs")
df.head()
```

```
Out[2]:
```

	batting_team	total_runs
0	Kolkata Knight Riders	1
1	Kolkata Knight Riders	1
2	Kolkata Knight Riders	0
3	Kolkata Knight Riders	1
4	Kolkata Knight Riders	1

## Grouping of data

we have grouped data using `.grouby()` function using various values of `batting_team` feature/column. The `groupby()` function is then followed by `.sum()` to summarize values for other numerical columns in the dataframe. The resulting dataframe is then assigned to dataframe `df_teams_total_runs`.

```
In [3]: df_team_total_runs = df.groupby("batting_team").sum()
```

## Reducing the dataset to our need

In the cell below, we have created a new pandas DataFrame by the name `df_team_GT10K` and assigned it a filtered version of dataframe `df_team_total_runs` such that only those observations are accepted which have `total_runs` value more than `10000`. This can be done like:

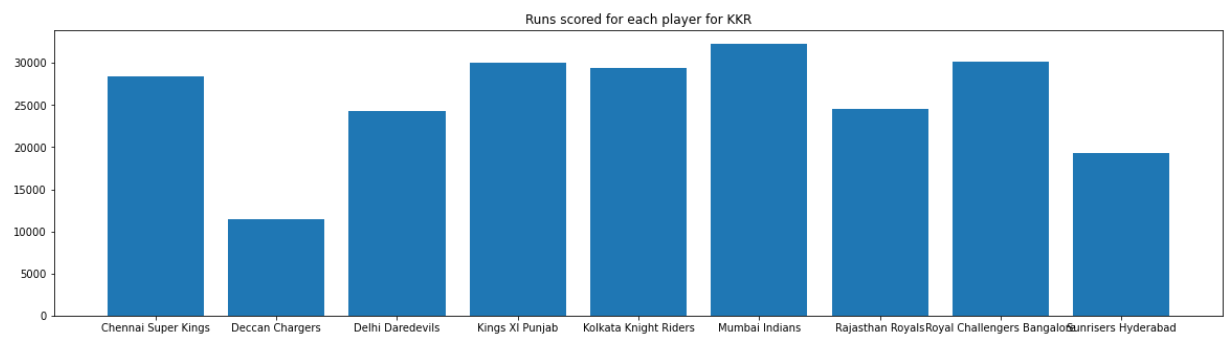
```
df_team_total_runs[df_team_total_runs["total_runs"]>10000]
```

```
In [4]: df_team_GT10K = df_team_total_runs[df_team_total_runs["total_runs"]>10000]
```

## Plotting of information

1. `plt.figure()` is used to increase the size of the figure. The argument `figsize` take a tuple value i.e., value in a `()` such that the first value is width and second is height like `(width, height)` as shown in cell below `(20,5)`.
2. `plt.title()` provides the graph or chart with a title.
3. `plt.bar()` function is used to plot bar chart. We have plotted bar chart for `df_team_GT10K` dataframe's index value which are, infact, each team as categories or x-axis of the chart and the runs scored by them as y-axis.
4. `plt.show()` function combines all the elements of charts and shows them in harmony.

```
In [5]: plt.figure(figsize=(20,5))
plt.title("Runs scored for each player for KKR")
plt.bar(df_team_GT10K.index, df_team_GT10K["total_runs"])
plt.show()
```



In [ ]: