# Configuration Manual

MSc Research Project
MSc Data Analytics

Bhaskar Reddy Yenuga
Student ID: x21150567

School of Computing
National College of Ireland

Supervisor: Cristina Hava Muntean

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** ……Bhaskar Reddy Yenuga……

**Student ID:** ……x21150567……

**Programme:** ……MSc Data Analytics…… **Year:** …2022-23……..

**Module:** ……MSc Research Project……

**Lecturer:** ……Cristina Hava Muntean……

**Submission Due Date:** ……21-12-2022……

**Project Title:** **.....** Selection of Best Players in ODI Cricket using Ranking Based Indexing Method**.....**

**Word Count:** ……1255…… **Page Count:** ……22……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** ……Bhaskar Reddy Yenuga
……

**Date:** ……21-12-2022……

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

**Office Use Only**

| | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration Manual

Bhaskar Reddy Yenuga
x21150567

# 1 Requirements

This configuration manual helps in understanding and viewing the file by following the simple step-by-step procedure. In this document, each and every step will be explained that need to know for the person to open and understand this.

# 2 Specification of the system

## 2.1 Hardware

Below are the specifications of the system used to execute and work on both documentation and coding.

| | |
|---|---|
| OS Name | Microsoft Windows 11 Home Single Language |
| Version | 10.0.22621 Build 22621 |
| Other OS Description | Not Available |
| OS Manufacturer | Microsoft Corporation |
| System Name | DESKTOP-HUSETNI |
| System Manufacturer | HP |
| System Model | HP Spectre x360 Convertible 13-aw2xxx |
| System Type | x64-based PC |
| System SKU | 2D9H6PA#ACJ |
| Processor | 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz, 2701 Mhz, 4 Core(s), 8 L... |
| BIOS Version/Date | AMI F.13, 01-04-2021 |
| SMBIOS Version | 3.3 |
| Embedded Controller Version | 31.32 |
| BIOS Mode | UEFI |
| BaseBoard Manufacturer | HP |
| BaseBoard Product | 8709 |
| BaseBoard Version | 31.32 |
| Platform Role | Mobile |
| Secure Boot State | On |
| PCR7 Configuration | Elevation Required to View |
| Windows Directory | C:\WINDOWS |
| System Directory | C:\WINDOWS\system32 |
| Boot Device | \Device\HarddiskVolume1 |
| Locale | United States |
| Hardware Abstraction Layer | Version = "10.0.22621.819" |
| User Name | DESKTOP-HUSETNI\solip |

***Figure 1: System Hardware***

## 2.2 Softwares

For the next steps, the software that are utilized are as follows.

- Microsoft office (Word, Excel, Power point presentation)
- Anaconda Navigator (Python 6.4.8)
- Microsoft Bing (Browser used for web scrapping)

# 3 The working environment and the steps involved

## 3.1 Data Selection:

The required data is extracted from www.espncricinfo.com. And it has an option for selecting the aspects according to the requirement in a new feature called 'statsguru'.

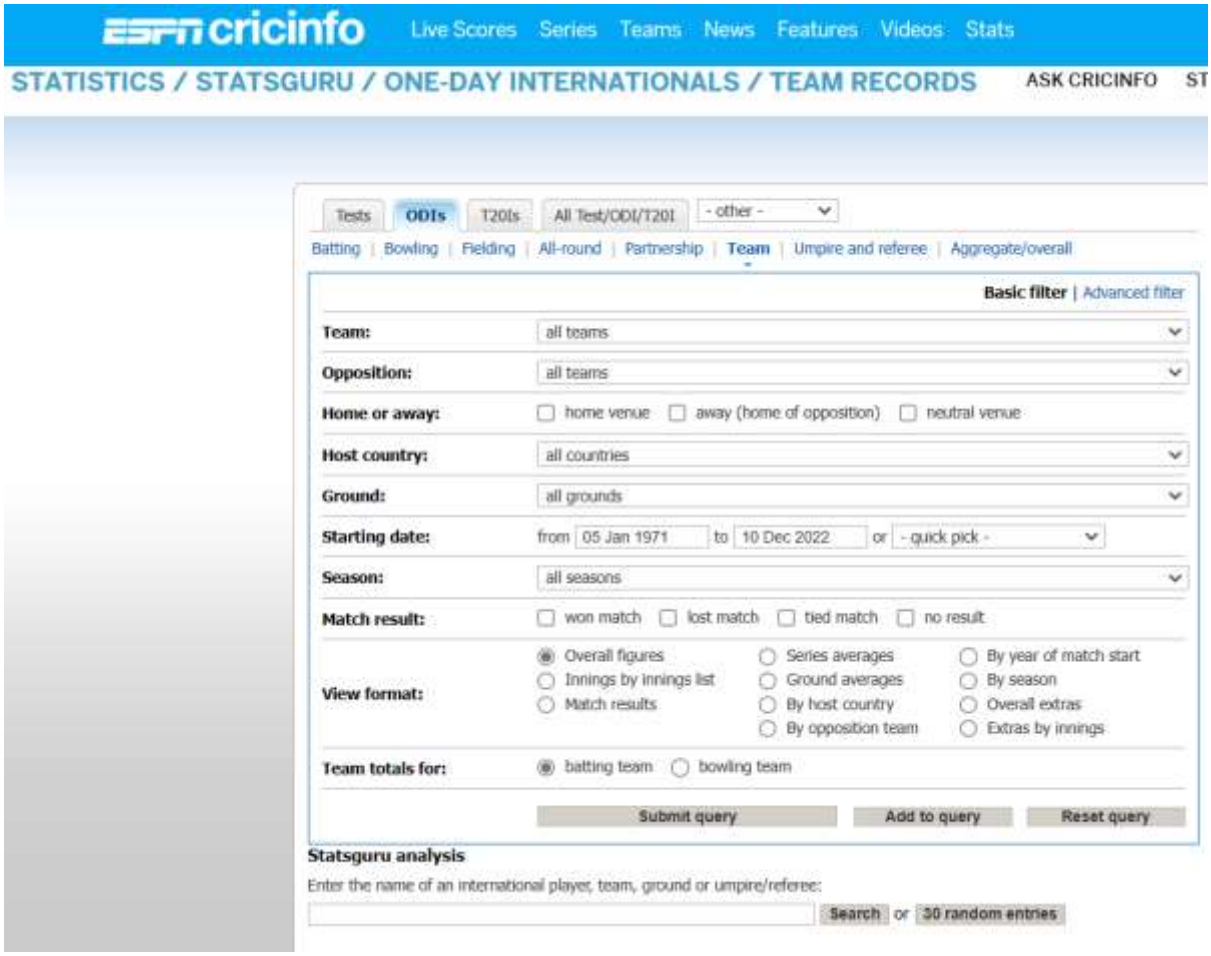


***Figure 2: Data Selection***

## 3.2 Implemenetation

In this section for implementing the code, it is mandatory that there must be some libraries that need to be installed and imported into the kernel notebook. And they have been listed below.

```
In [2]: #importing the all the libraries that are used
        import math
        import numpy as np
        import random
        import pandas as pd
        import re
        import os
        import matplotlib.pyplot as plt
        from bs4 import BeautifulSoup as bs
        import requests
        from statistics import mean
        import seaborn as sb
```

***Code 1: Importing Libraries***

## 3.3 Importing the data

In this scenario, the data is completely not extracted in one go, it has to be according to the requirement i.e., based on batting, bowling, allrounder, fielding, partnership, 1st innings ratio, etc. Player Section during Nov in past four years through web scrapping. Below is the URL for batting stats for the last 4 years. Below is the code for, extracting the HTML page and beautifying it by adding delimiters to make the page into the required dataset.

**How to web scrap the data into pandas data frame**

```
url='https://stats.espncricinfo.com/ci/engine/stats/index.html?class=2;result=1;result=2;result=3;spanmax1=04+Dec+2022;spanmin1=04+Dec+2018;spanval1=span;team=6;template=results;type=batting'
from bs4 import BeautifulSoup as bs
import requests
page=requests.get(url)
soup = bs(page.content,'html.parser')
name1=soup.find_all('tr',class_= 'data1')
player_name=[]
for i in range(0,len(name1)):
  player_name.append(name1[i].get_text())
```

```
player_name[:] = [name1.lstrip('\n') for name1 in player_name]
player_name[:] = [name1.rstrip('\n') for name1 in player_name]
player_name[:] = [name1.replace('\n',',') for name1 in player_name]
var1 = player_name[0].split(",")
df = pd.DataFrame()
df['Index'] = player_name
new = df['Index'].str.split(",", n = 15, expand = True)
Playerquery_bat = pd.DataFrame()
Playerquery_bat['Player']=new[0]
Playerquery_bat['Span']=new[1]
Playerquery_bat['Mat']=new[2]
Playerquery_bat['Inns']=new[3]
Playerquery_bat['NO']=new[4]
Playerquery_bat['Runs']=new[5]
Playerquery_bat['HS']=new[6]
Playerquery_bat['Ave']=new[7]
Playerquery_bat['BF']=new[8]
Playerquery_bat['SR']=new[9]
Playerquery_bat['100']=new[10]
Playerquery_bat['50']=new[11]
Playerquery_bat['0']=new[12]
Playerquery_bat['4s']=new[13]
Playerquery_bat['6s']=new[14]
```

As similar it has been collected total 7 characteristics about cricket player with team performance against the different countries.

1. Bowling Section
2. Partnership Section
3. Fielding Section
4. All-rounder Section
5. India Win/lose ratio from 1st innings batting
6. India Win/lose ratio from 1st innings bowling

and additionally, Wicket Keeper section.

## 3.4 Reading the data

```
In [16]: Playerquery_bat
```

Out[16]:

| | Player | Span | Mat | Inns | NO | Runs | HS | Ave | BF | SR | 100 | 50 | 0 | 4s | 6s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | V Kohli | 2019-2022 | 46 | 46 | 2 | 2121 | 123 | 48.20 | 2295 | 92.41 | 5 | 16 | 3 | 198 | 14 |
| 1 | RG Sharma | 2019-2022 | 40 | 40 | 2 | 1949 | 159 | 51.28 | 2138 | 91.15 | 8 | 8 | 2 | 205 | 49 |
| 2 | S Dhawan | 2019-2022 | 47 | 46 | 4 | 1816 | 143 | 43.23 | 2129 | 85.29 | 2 | 14 | 2 | 224 | 15 |
| 3 | KL Rahul | 2019-2022 | 33 | 32 | 3 | 1421 | 112 | 49.00 | 1572 | 90.39 | 4 | 9 | 1 | 105 | 38 |
| 4 | SS Iyer | 2019-2022 | 28 | 27 | 3 | 1193 | 113* | 49.70 | 1216 | 98.10 | 2 | 11 | 0 | 116 | 24 |
| 5 | RR Pant | 2019-2022 | 24 | 23 | 1 | 814 | 125* | 37.00 | 764 | 106.54 | 1 | 5 | 3 | 83 | 25 |
| 6 | HH Pandya | 2019-2022 | 24 | 21 | 3 | 716 | 92* | 39.77 | 614 | 116.61 | 0 | 4 | 2 | 63 | 23 |
| | Shubman | 2019- | | | | | | | | | | | | | |

***Code 2: Reading Batsmen dataset***

Now, downloading the dataset as a csv file into the computer by the name csv.

```
In [75]: Playerquery_bat.to_csv('bat.csv', index=None, header=True)
```

***Code 3: Downloading the data***

## 3.5 Data pre-processing

```
In [79]: Playerquery_bat.isnull().sum()

Out[79]: Player    0
         Span      0
         Mat       0
         Inns      0
         NO        0
         Runs      0
         HS        0
         Ave       0
         BF        0
         SR        0
         100       0
         50        0
         0         0
         4s        0
         6s        0
         dtype: int64
```

*Batting dataset have so many null values but it shows zero null value.*

***Code 4: Checking for null values***

In [81]:
```
Playerquery_bat_copy=Playerquery_bat_copy.replace("-", '0')
```

In [83]:
```
Playerquery_bat_copy["Player"]=Playerquery_bat_copy["Player"].astype("str")
Playerquery_bat_copy["Player"]=Playerquery_bat_copy["Player"].astype("str")
Playerquery_bat_copy["Span"]=Playerquery_bat_copy["Span"].astype("str")
Playerquery_bat_copy["Mat"]=Playerquery_bat_copy["Mat"].astype("int64")
Playerquery_bat_copy["Inns"]=Playerquery_bat_copy["Inns"].astype("int64")
Playerquery_bat_copy["NO"]=Playerquery_bat_copy["NO"].astype("int64")
Playerquery_bat_copy["Runs"]=Playerquery_bat_copy["Runs"].astype("int64")
Playerquery_bat_copy["HS"]=Playerquery_bat_copy["HS"].astype("str")
Playerquery_bat_copy["Ave"]=Playerquery_bat_copy["Ave"].astype("float")
Playerquery_bat_copy["BF"]=Playerquery_bat_copy["BF"].astype("int64")
Playerquery_bat_copy["SR"]=Playerquery_bat_copy["SR"].astype("float")
Playerquery_bat_copy["100"]=Playerquery_bat_copy["100"].astype("int64")
Playerquery_bat_copy["50"]=Playerquery_bat_copy["50"].astype("int64")
Playerquery_bat_copy["0"]=Playerquery_bat_copy["0"].astype("int64")
Playerquery_bat_copy["4s"]=Playerquery_bat_copy["4s"].astype("int64")
Playerquery_bat_copy["6s"]=Playerquery_bat_copy["6s"].astype("int64")
```

In [84]:
```
Playerquery_bat_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Player  50 non-null     object 
 1   Span    50 non-null     object 
 2   Mat     50 non-null     int64  
 3   Inns    50 non-null     int64  
 4   NO      50 non-null     int64  
 5   Runs    50 non-null     int64  
 6   HS      50 non-null     object 
 7   Ave     50 non-null     float64
 8   BF      50 non-null     int64  
 9   SR      50 non-null     float64
 10  100     50 non-null     int64  
 11  50      50 non-null     int64  
 12  0       50 non-null     int64  
 13  4s      50 non-null     int64  
 14  6s      50 non-null     int64  
dtypes: float64(2), int64(10), object(3)
memory usage: 6.0+ KB
```

## 3.6 Feature engineering

→ Taking Batsmen for instance.

In [85]:
```
Playerquery_bat_copy.describe()
```

Out[85]:

| | Mat | Inns | NO | Runs | Ave | BF | SR | 100 | 50 | 0 | 4s | 6s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 50.000000 | 50.000000 | 50.000000 | 50.000000 | 50.000000 | 50.000000 | 50.000000 | 50.000000 | 50.000000 | 50.000000 | 50.00000 | 50.000000 |
| mean | 13.620000 | 10.160000 | 2.140000 | 312.100000 | 25.664400 | 338.220000 | 79.388000 | 0.460000 | 1.920000 | 0.560000 | 29.86000 | 6.140000 |
| std | 13.117818 | 11.450622 | 2.373407 | 519.580414 | 20.476493 | 567.744168 | 37.717837 | 1.459801 | 3.696882 | 0.860944 | 53.10406 | 10.210059 |
| min | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.00000 | 0.000000 |
| 25% | 2.250000 | 1.250000 | 0.000000 | 13.000000 | 6.625000 | 19.750000 | 60.252500 | 0.000000 | 0.000000 | 0.000000 | 1.00000 | 0.000000 |
| 50% | 8.500000 | 6.000000 | 2.000000 | 99.500000 | 24.500000 | 119.000000 | 85.630000 | 0.000000 | 0.000000 | 0.000000 | 8.50000 | 1.500000 |
| 75% | 24.000000 | 13.750000 | 3.000000 | 314.250000 | 42.222500 | 324.750000 | 99.782500 | 0.000000 | 2.000000 | 1.000000 | 25.75000 | 6.750000 |
| max | 47.000000 | 46.000000 | 11.000000 | 2121.000000 | 66.000000 | 2295.000000 | 200.000000 | 8.000000 | 16.000000 | 3.000000 | 224.00000 | 49.000000 |

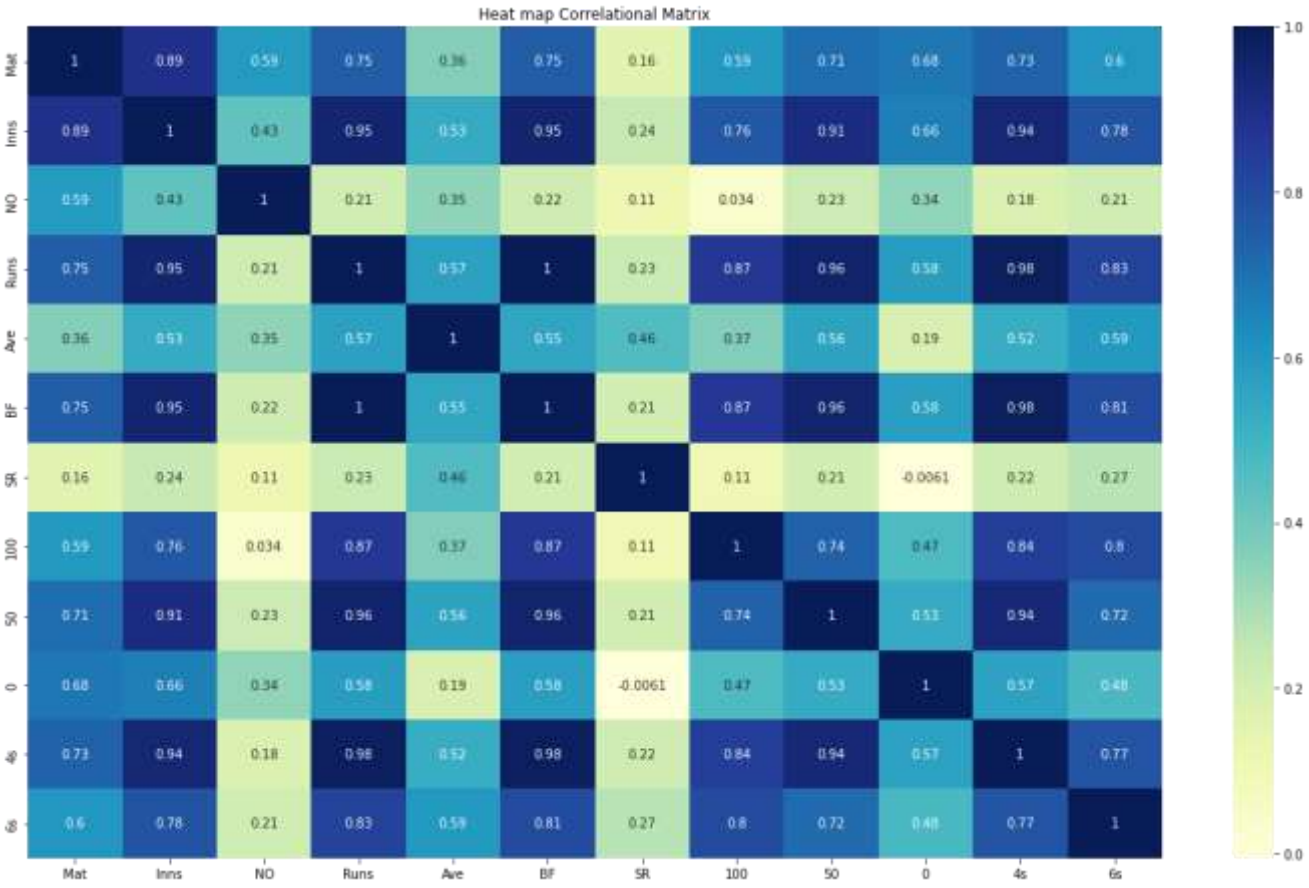


***Figure 3: Correlation matrix***

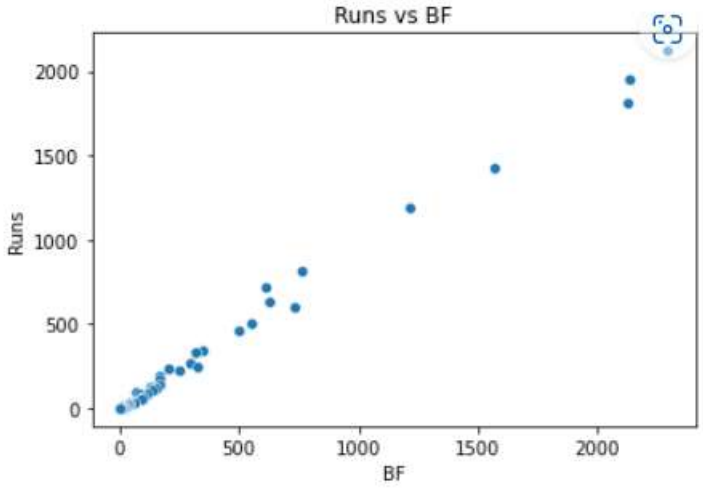


***Figure 4: Over proportion between Balls Faced and Runs: Scatter Plot***

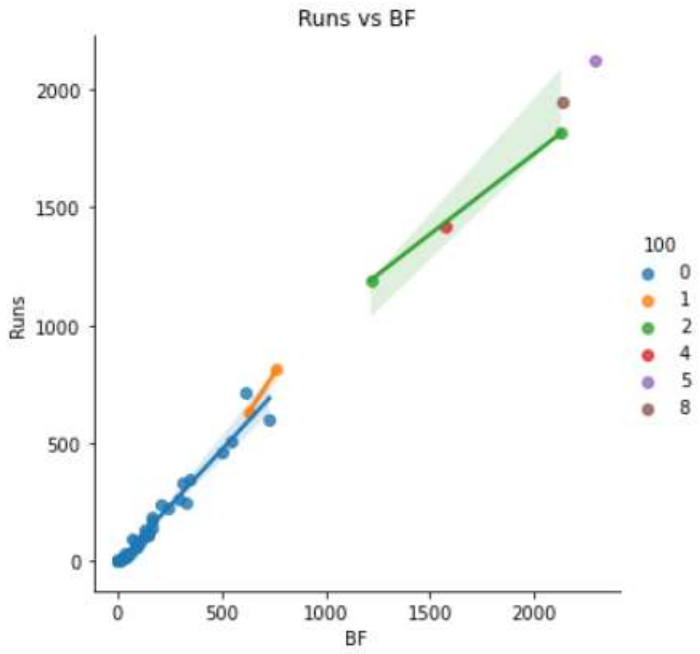


*Figure 5: Most 100s from the proportional graph between Runs and BF*

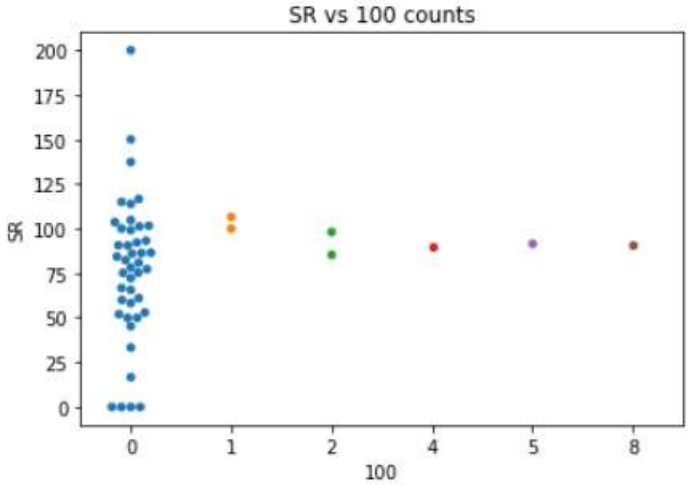


**Figure 6: Good Strike Rate as well as most 100s in the ODI**

- **Who have equal 8 50s in the ODI?**

```
In [91]: Playerquery_bat_copy.loc[Playerquery_bat_copy['50'] ==8]
```

Out[91]:

| | Player | Span | Mat | Inns | NO | Runs | HS | Ave | BF | SR | 100 | 50 | 0 | 4s | 6s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | RG Sharma | 2019-2022 | 40 | 40 | 2 | 1949 | 159 | 51.28 | 2138 | 91.15 | 8 | 8 | 2 | 205 | 49 |

*Code 5: Batsmen scoring 8 50s*

- **Who have more than 100 Strike Rate as well as more than 2 50s in the ODI?**

```
In [92]: Playerquery_bat_copy.loc[(Playerquery_bat_copy['SR'] > 100) & (Playerquery_bat_copy['50'] > 2 )]
```

Out[92]:

| | Player | Span | Mat | Inns | NO | Runs | HS | Ave | BF | SR | 100 | 50 | 0 | 4s | 6s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | RR Pant | 2019-2022 | 24 | 23 | 1 | 814 | 125* | 37.00 | 764 | 106.54 | 1 | 5 | 3 | 83 | 25 |
| 6 | HH Pandya | 2019-2022 | 24 | 21 | 3 | 716 | 92* | 39.77 | 614 | 116.61 | 0 | 4 | 2 | 63 | 23 |

*Code 6: Batsmen with strike rate greater than 100 and two 50s*

- **Select those Players who have played more than 10 matches and having topmost SR?**

```
In [93]: data_select = Playerquery_bat_copy.nlargest(10,['SR'])
         data_select.loc[(data_select['Mat'] > 10 )]
```

Out[93]:

| | Player | Span | Mat | Inns | NO | Runs | HS | Ave | BF | SR | 100 | 50 | 0 | 4s | 6s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | HH Pandya | 2019-2022 | 24 | 21 | 3 | 716 | 92* | 39.77 | 614 | 116.61 | 0 | 4 | 2 | 63 | 23 |
| 15 | SN Thakur | 2019-2022 | 24 | 15 | 5 | 238 | 50* | 23.80 | 207 | 114.97 | 0 | 1 | 0 | 26 | 6 |
| 5 | RR Pant | 2019-2022 | 24 | 23 | 1 | 814 | 125* | 37.00 | 764 | 106.54 | 1 | 5 | 3 | 83 | 25 |
| 12 | SV Samson | 2021-2022 | 11 | 10 | 5 | 330 | 86* | 66.00 | 315 | 104.76 | 0 | 2 | 0 | 25 | 15 |

*Code 7: Batsmen with more strike rate and played more than 10 matches*

## 4. Ranking

- **Scatter plot among the features**

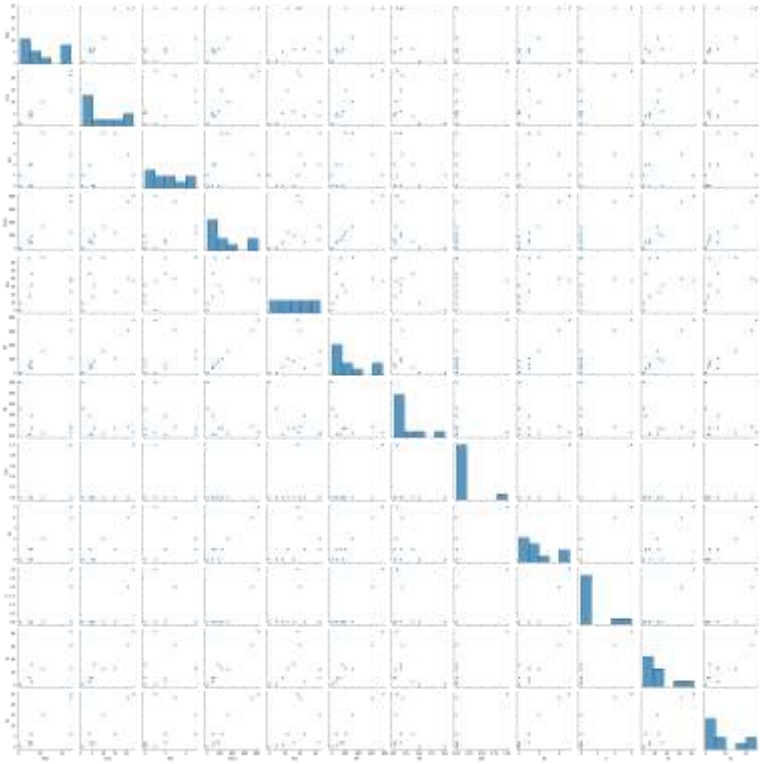

*Figure 7: Scatter plot*

- **Selective features with basis of priority table weight**

```
Playerquery_bat_copy['Rank_SR']=0
Playerquery_bat_copy['Rank_Inns']=0
Playerquery_bat_copy['Rank_4s']=0
Playerquery_bat_copy['Rank_50']=0
Playerquery_bat_copy['Rank_6s']=0
Playerquery_bat_copy['Rank_100']=0
Playerquery_bat_copy['Rank_Ave']=0
Playerquery_bat_copy['Rank_0']=0
Playerquery_bat_copy['Total_Weight']=0
Playerquery_bat_copy['Rank']=0
```

- **Rank Analysis**

```
Playerquery_bat_copy['Rank_SR']=Playerquery_bat_copy['SR'] -
 Playerquery_bat_copy['SR'].mean()
Playerquery_bat_copy['Rank_Inns']=Playerquery_bat_copy['Inns'] -
 Playerquery_bat_copy['Inns'].mean()
Playerquery_bat_copy['Rank_4s']=Playerquery_bat_copy['4s'] -
 Playerquery_bat_copy['4s'].mean()
Playerquery_bat_copy['Rank_50']=Playerquery_bat_copy['50'] -
 Playerquery_bat_copy['50'].mean()
Playerquery_bat_copy['Rank_6s']=Playerquery_bat_copy['6s'] -
 Playerquery_bat_copy['6s'].mean()
Playerquery_bat_copy['Rank_100']=Playerquery_bat_copy['100'] -
 Playerquery_bat_copy['100'].mean()
Playerquery_bat_copy['Rank_Ave']=Playerquery_bat_copy['Ave'] -
 Playerquery_bat_copy['Ave'].mean()
Playerquery_bat_copy['Rank_0']=Playerquery_bat_copy['0'] -
 Playerquery_bat_copy['0'].mean()
```

- **Total Weight Estimation**

```
Playerquery_bat_copy['Total_Weight'] = Playerquery_bat_copy['Rank_SR'] * 0.90 + Playerq
uery_bat_copy['Rank_Inns'] * 0.85 + Playerquery_bat_copy['Rank_4s'] * 0.80 + Playerquery
```

_bat_copy['Rank_50'] * 0.75 + Playerquery_bat_copy['Rank_6s'] * 0.70 + Playerquery_bat_copy['Rank_100'] * 0.65 + Playerquery_bat_copy['Rank_Ave'] * 0.60 - Playerquery_bat_copy['Rank_0'] * 0.80

Playerquery_bat_copy

- **Select up to First 10 Ranks**

data_select = Playerquery_bat_copy.nlargest(10,['Total_Weight'])

data_select


***Code 8: Batsmen who made most 100s in the ODI***


***Code 9: Batsmen who made most 4s in the ODI***


***Code 10: Batsmen who made most 6s in the ODI***


***Code 11: Batsmen who made most 50s in the ODI***

```
In [106]: Playerquery_bat_copy.loc[Playerquery_bat_copy['Runs'] ==Playerquery_bat_copy['Runs'].max()]
```

Out[106]:

| layer | Span | Mat | Inns | NO | Runs | HS | Ave | BF | SR | ... | Rank_SR | Rank_Inns | Rank_4s | Rank_50 | Rank_6s | Rank_100 | Rank_Ave | Rank_0 | Total_Wei |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| V Kohli | 2019-2022 | 46 | 45 | 2 | 2121 | 123 | 48.2 | 2295 | 92.41 | ... | 13.022 | 35.84 | 168.14 | 14.08 | 7.66 | 4.54 | 22.5356 | 2.44 | 207.27 |

rs = 25 columns

***Code 12: Batsmen who made most Runs in the ODI***

Similarly, all other characteristics of cricket players have been analyzed. Then separate ranking has been received from all criteria. Now, it is required to estimate the final ranking for the selection of 11 squad team from 15 players.

**Formation by Team Selectors:**

Tournament: World Cup

Bench Size = 15

Batting player with Wicket Keeper and allrounder = 9

Bowler = 6

Condition: All matches are away

- **First Selection for best 15 players**

Out of 15, first it goes to 9 batsmen.

```
In [271]: select_batting_player
```

Out[271]:

| | Player | Batting_Rank | Total_Weight_Batting |
|---|---|---|---|
| 0 | RG Sharma | 1.0 | 229.74216 |
| 1 | S Dhawan | 2.0 | 216.73816 |
| 2 | V Kohli | 3.0 | 207.27816 |
| 3 | SS Iyer | 4.0 | 135.24916 |
| 4 | KL Rahul | 5.0 | 132.14016 |
| 5 | RR Pant | 6.0 | 98.57516 |
| 6 | HH Pandya | 7.0 | 89.60016 |
| 7 | Shubman Gill | 8.0 | 75.44716 |
| 8 | Ravi Bishnoi | 9.0 | 56.68916 |

***Code 13: Selecting the 9 batsmen***

- **Checking the Partnership from the batting selection**

```
In [273]: data_select_bat_partnership = rank_for_partnership.nlargest(15,['Total_Weight'])
          data_select_bat_partnership[['Player1','Player2','Rank']]
```

Out[273]:

| | Player1 | Player2 | Rank |
|---|---|---|---|
| 0 | KL Rahul | RG Sharma | 1.0 |
| 1 | S Dhawan | RG Sharma | 2.0 |
| 2 | S Dhawan | Shubman Gill | 3.0 |
| 3 | KL Rahul | RG Sharma | 4.0 |
| 4 | KL Rahul | RG Sharma | 5.0 |
| 5 | Ishan Kishan | SS Iyer | 6.0 |
| 6 | S Dhawan | RG Sharma | 7.0 |
| 7 | RA Jadeja | HH Pandya | 8.0 |
| 8 | MS Dhoni | KM Jadhav | 9.0 |
| 9 | Ishan Kishan | Shubman Gill | 10.0 |
| 10 | MS Dhoni | RG Sharma | 11.0 |
| 11 | V Kohli | RG Sharma | 12.0 |
| 12 | V Kohli | RG Sharma | 13.0 |
| 13 | KL Rahul | RG Sharma | 14.0 |
| 14 | SS Iyer | KL Rahul | 15.0 |

```
In [283]: select_batting_player
```

Out[283]:

| | Player | Batting_Rank | Total_Weight_Batting | Highest_Rank_partnership |
|---|---|---|---|---|
| 0 | RG Sharma | 1.0 | 229.74216 | 1 |
| 1 | S Dhawan | 2.0 | 216.73816 | 2 |
| 2 | V Kohli | 3.0 | 207.27816 | 12 |
| 3 | SS Iyer | 4.0 | 135.24916 | 6 |
| 4 | KL Rahul | 5.0 | 132.14016 | 1 |
| 5 | RR Pant | 6.0 | 98.57516 | 16 |
| 6 | HH Pandya | 7.0 | 89.60016 | 8 |
| 7 | Shubman Gill | 8.0 | 75.44716 | 3 |
| 8 | Ravi Bishnoi | 9.0 | 56.68916 | 171 |

***Code 14: Selecting the batsmen including partnership weightage***

- **Removing player from the bench**

Reason to remove is because it has very low partnership rank.

```
In [284]: remove_player = select_batting_player.nlargest(1,['Highest_Rank_partnership'])

In [285]: remove_player
```

Out[285]:

| | Player | Batting_Rank | Total_Weight_Batting | Highest_Rank_partnership |
|---|---|---|---|---|
| 8 | Ravi Bishnoi | 9.0 | 56.68916 | 171 |

***Code 15: Removing the players***

- **Add player from batting Records**

```
In [290]: addplayer=pd.DataFrame()
          addplayer=rank_for_bat.loc[((rank_for_bat['Player']=="SV Samson"))]
          addplayer
```

Out[290]:

| Player | Span | Mat | Inns | NO | Runs | HS | Ave | BF | SR | ... | Rank_SR | Rank_Inns | Rank_4s | Rank_50 | Rank_6s | Rank_100 | Rank_Ave | Rank_0 | Total_W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SV Samson | 2021-2022 | 11 | 10 | 5 | 330 | 86* | 66.0 | 315 | 104.76 | ... | 25.372 | -0.16 | -4.86 | 0.08 | 8.86 | -0.46 | 40.3356 | -0.56 | 49.4 |

ws × 25 columns

```
In [293]: select_batting_player
```

Out[293]:

| | Player | Batting_Rank | Total_Weight_Batting | Highest_Rank_partnership |
|---|---|---|---|---|
| 0 | RG Sharma | 1.0 | 229.74216 | 1.0 |
| 1 | S Dhawan | 2.0 | 216.73816 | 2.0 |
| 2 | V Kohli | 3.0 | 207.27816 | 12.0 |
| 3 | SS Iyer | 4.0 | 135.24916 | 6.0 |
| 4 | KL Rahul | 5.0 | 132.14016 | 1.0 |
| 5 | RR Pant | 6.0 | 98.57516 | 16.0 |
| 6 | HH Pandya | 7.0 | 89.60016 | 8.0 |
| 7 | Shubman Gill | 8.0 | 75.44716 | 3.0 |
| 8 | SV Samson | 10.0 | 49.42316 | 43.0 |

So Again the batting partnership of SV Samson is not acceptable

```
In [294]: remove_player = select_batting_player.nlargest(1,['Highest_Rank_partnership'])
          select_batting_player.drop(8,axis=0,inplace=True)
```

Looking for Player whose partnership is acceptable and marginable.

```
In [296]: data_select = rank_for_partnership.nlargest(10,['Runs'])
          data_select=data_select.loc[(data_select['Runs'] > 120 )]
          data_select
```

Out[296]:

| | Player1 | Player2 | Wkt | Runs | Overs | RR | In | Out | Inns | Opposition | Ground | Start Date | Rank_Runs | Rank_Overs | Rank_RR | Rank_Inns | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | KL Rahul | RG Sharma | 1 | 227 | 37.0 | 6.13 | - | 1/227 | 1 | v West Indies | Visakhapatnam | 18 Dec 2019 | 190.264574 | 30.862108 | 0.397758 | -0.461883 | 392 |
| 1 | S Dhawan | RG Sharma | 1 | 193 | 31.0 | 6.22 | - | 1/193 | 1 | v Australia | Mohali | 10 Mar 2019 | 156.264574 | 24.862108 | 0.467758 | -0.461883 | 322 |
| 2 | S Dhawan | Shubman Gill | 1 | 192 | 30.5 | 6.22 | - | 0/192 | 2 | v Zimbabwe | Harare | 18 Aug 2022 | 155.264574 | 24.362108 | 0.467758 | 0.538117 | 320 |
| 3 | KL Rahul | RG Sharma | 1 | 189 | 30.1 | 6.26 | - | 1/189 | 2 | v Sri Lanka | Leeds | 6 Jul 2019 | 152.264574 | 23.962108 | 0.527758 | 0.538117 | 314 |
| 4 | KL Rahul | RG Sharma | 1 | 180 | 29.2 | 6.13 | - | 1/180 | 1 | v Bangladesh | Birmingham | 2 Jul 2019 | 143.264574 | 23.062108 | 0.397758 | -0.461883 | 296 |
| 5 | Ishan Kishan | SS Iyer | 3 | 161 | 25.4 | 6.27 | 2/48 | 3/209 | 2 | v South Africa | Ranchi | 9 Oct 2022 | 124.264574 | 19.262108 | 0.537758 | 0.538117 | 256 |
| 6 | S Dhawan | RG Sharma | 1 | 154 | 25.2 | 6.07 | - | 1/154 | 1 | v New Zealand | Mount Maunganui | 26 Jan 2019 | 117.264574 | 19.062108 | 0.337758 | -0.461883 | 241 |
| 7 | RA Jadeja | HH Pandya | 6 | 150 | 18.0 | 8.33 | 5/152 | 5/302 | 1 | v Australia | Canberra | 2 Dec 2020 | 113.264574 | 11.962108 | 2.597758 | -0.461883 | 230 |
| 8 | MS Dhoni | KM Jadhav | 5 | 141 | 24.5 | 5.67 | 4/99 | 4/240 | 2 | v Australia | Hyderabad (Deccan) | 2 Mar 2019 | 104.264574 | 18.362108 | -0.062242 | 0.538117 | 216 |
| 9 | Ishan Kishan | Shubman Gill | 3 | 140 | 21.1 | 6.61 | 2/64 | 3/224 | 1 | v Zimbabwe | Harare | 22 Aug 2022 | 103.264574 | 14.962108 | 0.877758 | -0.461883 | 211 |

***Code 16: Top 10 player-partnership ranking***

```
In [297]: player=()
          player=data_select['Player1'].unique()

In [298]: player1=()
          player1=data_select['Player2'].unique()

In [299]: player=np.append(player,player1,axis=0)

In [300]: player

Out[300]: array(['KL Rahul', 'S Dhawan', 'Ishan Kishan', 'RA Jadeja', 'MS Dhoni',
                 'RG Sharma', 'Shubman Gill', 'SS Iyer', 'HH Pandya', 'KM Jadhav'],
                dtype=object)
```

```
In [301]: data_select=rank_for_partnership.loc[(rank_for_partnership['Player1'] =="RA Jadeja" )|(rank_for_partnership['Player2'] =="RA
          data_select[['Player1','Player2','Rank']].iloc[0]

Out[301]: Player1    RA Jadeja
          Player2    HH Pandya
          Rank             8.0
          Name: 7, dtype: object

In [302]: data_select=rank_for_partnership.loc[(rank_for_partnership['Player1'] =="MS Dhoni" )|(rank_for_partnership['Player2'] =="MS D
          data_select[['Player1','Player2','Rank']].iloc[0]

Out[302]: Player1     MS Dhoni
          Player2    KM Jadhav
          Rank             9.0
          Name: 8, dtype: object

In [303]: data_select=rank_for_partnership.loc[(rank_for_partnership['Player1'] =="KM Jadhav" )|(rank_for_partnership['Player2'] =="KM
          data_select[['Player1','Player2','Rank']].iloc[0]

Out[303]: Player1     MS Dhoni
          Player2    KM Jadhav
          Rank             9.0
          Name: 8, dtype: object
```

***Code 17: Checking the best player partnership amongst RA Jadeja, MS Dhoni, and KM Jadhav***

Out[304]:

| | Player | Batting_Rank | Total_Weight_Batting | Highest_Rank_partnership |
|---|---|---|---|---|
| 0 | RG Sharma | 1.0 | 229.74216 | 1.0 |
| 1 | S Dhawan | 2.0 | 216.73816 | 2.0 |
| 2 | V Kohli | 3.0 | 207.27816 | 12.0 |
| 3 | SS Iyer | 4.0 | 135.24916 | 6.0 |
| 4 | KL Rahul | 5.0 | 132.14016 | 1.0 |
| 5 | RR Pant | 6.0 | 98.57516 | 16.0 |
| 6 | HH Pandya | 7.0 | 89.60016 | 8.0 |
| 7 | Shubman Gill | 8.0 | 75.44716 | 3.0 |
| 8 | RA Jadeja | 15.0 | 30.01316 | 8.0 |

*Code 18: Adding RA Jadeja into the squad*

➔ **Now pick up from the Bowling Side**

In [305]:
```
data_select_bowl_personal_6 = rank_for_bowler.nlargest(6,['Total_Weight'])
select_bowling_player=data_select_bowl_personal_6[['Player','Rank','Total_Weight']]
```

In [309]:
```
select_bowling_player
```

Out[309]:

| | Player | Bowling_Rank | Total_Weight_Bowling | Highest_Rank_partnership |
|---|---|---|---|---|
| 0 | Kuldeep Yadav | 1.0 | 248.13944 | 0 |
| 1 | YS Chahal | 2.0 | 211.08394 | 0 |
| 2 | JJ Bumrah | 3.0 | 184.17044 | 0 |
| 3 | Mohammed Shami | 4.0 | 176.83894 | 0 |
| 4 | B Kumar | 5.0 | 136.73444 | 0 |
| 5 | RA Jadeja | 6.0 | 135.45744 | 0 |

Since, RA Jadeja is already inserted into the batting Squad so here this player is removed

*Code 19: Selecting the 6 bowlers from the data*

In [312]:
```
data_select_bowl_personal_15 = rank_for_bowler.nlargest(15,['Total_Weight'])
data_select_bowl_personal_15[['Player','Rank']]
```

Out[312]:

| | Player | Rank |
|---|---|---|
| 0 | Kuldeep Yadav | 1.0 |
| 1 | YS Chahal | 2.0 |
| 2 | JJ Bumrah | 3.0 |
| 3 | Mohammed Shami | 4.0 |
| 4 | B Kumar | 5.0 |
| 5 | RA Jadeja | 6.0 |
| 6 | SN Thakur | 7.0 |
| 7 | HH Pandya | 8.0 |
| 8 | Mohammed Siraj | 9.0 |
| 9 | M Prasidh Krishna | 10.0 |
| 10 | NA Saini | 11.0 |
| 11 | DL Chahar | 12.0 |
| 12 | KM Jadhav | 13.0 |
| 13 | AR Patel | 14.0 |
| 14 | Washington Sundar | 15.0 |

*Code 20: Top 15 bowlers based on ranks*

As SN Thakur is next ranking player so this player is selected

Out[292]:

| | Player | Bowling_Rank | Total_Weight_Bowling | Highest_Rank_partnership |
|---|---|---|---|---|
| 0 | Kuldeep Yadav | 1.0 | 248.13944 | 200.0 |
| 1 | YS Chahal | 2.0 | 211.08394 | 206.0 |
| 2 | JJ Bumrah | 3.0 | 184.17044 | 106.0 |
| 3 | Mohammed Shami | 4.0 | 176.83894 | 157.0 |
| 4 | B Kumar | 5.0 | 136.73444 | 54.0 |
| 5 | SN Thakur | 7.0 | 119.51944 | 52.0 |

***Code 21: The final bowlers***

In [294]:
```
select_player
```

Out[294]:

| | Player | Batting_Rank | Total_Weight_Batting | Highest_Rank_partnership | Bowling_Rank | Total_Weight_Bowling |
|---|---|---|---|---|---|---|
| 0 | RG Sharma | 1.0 | 229.74216 | 1.0 | NaN | NaN |
| 1 | S Dhawan | 2.0 | 216.73816 | 2.0 | NaN | NaN |
| 2 | V Kohli | 3.0 | 207.27816 | 12.0 | NaN | NaN |
| 3 | SS Iyer | 4.0 | 135.24916 | 6.0 | NaN | NaN |
| 4 | KL Rahul | 5.0 | 132.14016 | 1.0 | NaN | NaN |
| 5 | RR Pant | 6.0 | 98.57516 | 16.0 | NaN | NaN |
| 6 | HH Pandya | 7.0 | 89.60016 | 8.0 | NaN | NaN |
| 7 | Shubman Gill | 8.0 | 75.44716 | 3.0 | NaN | NaN |
| 8 | RA Jadeja | 15.0 | 30.01316 | 8.0 | NaN | NaN |
| 9 | Kuldeep Yadav | NaN | NaN | 200.0 | 1.0 | 248.13944 |
| 10 | YS Chahal | NaN | NaN | 206.0 | 2.0 | 211.08394 |
| 11 | JJ Bumrah | NaN | NaN | 106.0 | 3.0 | 184.17044 |
| 12 | Mohammed Shami | NaN | NaN | 157.0 | 4.0 | 176.83894 |
| 13 | B Kumar | NaN | NaN | 54.0 | 5.0 | 136.73444 |
| 14 | SN Thakur | NaN | NaN | 52.0 | 7.0 | 119.51944 |

***Code 22: The total bench***

For missing value replacement, the data is inserted through a query from the player's characteristics dataset.

Out[298]:

| | Player | Batting_Rank | Total_Weight_Batting | Highest_Rank_partnership | Bowling_Rank | Total_Weight_Bowling |
|---|---|---|---|---|---|---|
| 0 | RG Sharma | 1.0 | 229.74216 | 1.0 | 37.0 | -36.43906 |
| 1 | S Dhawan | 2.0 | 216.73816 | 2.0 | 37.0 | -36.43906 |
| 2 | V Kohli | 3.0 | 207.27816 | 12.0 | 37.0 | -36.43906 |
| 3 | SS Iyer | 4.0 | 135.24916 | 6.0 | 47.0 | -37.70906 |
| 4 | KL Rahul | 5.0 | 132.14016 | 1.0 | 37.0 | -36.43906 |
| 5 | RR Pant | 6.0 | 98.57516 | 16.0 | 37.0 | -36.43906 |
| 6 | HH Pandya | 7.0 | 89.60016 | 8.0 | 8.0 | 82.61744 |
| 7 | Shubman Gill | 8.0 | 75.44716 | 3.0 | 37.0 | -36.43906 |
| 8 | RA Jadeja | 15.0 | 30.01316 | 8.0 | 6.0 | 135.45744 |
| 9 | Kuldeep Yadav | 34.0 | -47.60384 | 200.0 | 1.0 | 248.13944 |
| 10 | YS Chahal | 37.0 | -54.53484 | 206.0 | 2.0 | 211.08394 |
| 11 | JJ Bumrah | 35.0 | -49.45184 | 106.0 | 3.0 | 184.17044 |
| 12 | Mohammed Shami | 33.0 | -33.55984 | 157.0 | 4.0 | 176.83894 |
| 13 | B Kumar | 32.0 | -32.70284 | 54.0 | 5.0 | 136.73444 |
| 14 | SN Thakur | 14.0 | 31.29216 | 52.0 | 7.0 | 119.51944 |

***Code 23: The final bench with bowling proper bowling weightage***

Now check this dataset according to fielding, allrounder and Wicket Keeper features then finalise the ODI squad for the international matches.

Out[307]:

| | Player | Rank | Batting_Rank | Highest_Rank_partnership | Rank_for_allrounder | Rank_for_fielding | Rank_for_WicketKeeper |
|---|---|---|---|---|---|---|---|
| 2 | V Kohli | 1.0 | 3.0 | 12.0 | 19 | 1 | 0 |
| 1 | S Dhawan | 2.0 | 2.0 | 2.0 | 25 | 2 | 0 |
| 0 | RG Sharma | 3.0 | 1.0 | 1.0 | 21 | 5 | 0 |
| 9 | Kuldeep Yadav | 4.0 | 34.0 | 200.0 | 13 | 19 | 0 |
| 8 | RA Jadeja | 5.0 | 15.0 | 8.0 | 4 | 7 | 0 |
| 6 | HH Pandya | 6.0 | 7.0 | 8.0 | 5 | 14 | 0 |
| 10 | YS Chahal | 7.0 | 37.0 | 206.0 | 7 | 10 | 0 |
| 12 | Mohammed Shami | 8.0 | 33.0 | 157.0 | 10 | 9 | 0 |
| 14 | SN Thakur | 9.0 | 14.0 | 52.0 | 9 | 25 | 0 |
| 4 | KL Rahul | 10.0 | 5.0 | 1.0 | 22 | 4 | 2 |
| 11 | JJ Bumrah | 11.0 | 35.0 | 106.0 | 17 | 34 | 0 |
| 3 | SS Iyer | 12.0 | 4.0 | 6.0 | 23 | 8 | 0 |
| 13 | B Kumar | 13.0 | 32.0 | 54.0 | 16 | 18 | 0 |
| 5 | RR Pant | 14.0 | 6.0 | 16.0 | 27 | 3 | 1 |
| 7 | Shubman Gill | 15.0 | 8.0 | 3.0 | 20 | 17 | 0 |

***Figure 8: The final squad***

# References