# VIRGINIA COMMONWEALTH UNIVERSITY

STATISTICAL ANALYSIS & MODELING

A1b: ANALYSIS OF IPL DATA USING PYTHON AND R

AZIM ZIYAN A

V01102412

Date of Submission: 18/06/2024

CONTENTS

|  |  |
| --- | --- |
| Content: | Page no: |
| INTRODUCTION | 3 |
| OBJECTIVE | 3 |
| BUSINESS SIGNIFICANC | 3 |
| RESULTS AND INTERPRETATIONS | 4-9 |

Analyzing Consumption in the State of Andhra Pradesh Using R

# INTRODUCTION

With the aim of organizing the IPLdata round-by-round and by batter, ball, runs, and wickets per player every match, this study focuses on the analysis of the IPL dataset. and to list the top three scorers in each IPL round, along with the top three wicket-takers.In the three IPL tournaments that were lost, it was also necessary to determine the best possible allocation of wickets and runs for the top three bowlers and batsmen.and to use R and Python to analyze the data to determine the correlation between a player's performance and his pay. In an effort to facilitate study, we have produced a dataset that includes the wages for the IPL as well as ball-by-ball updates until 2024.

# OBJECTIVES

1. Extract the files in R/Python
2. Arrange the data IPL round-wise and batsman, ball, runs, and wickets per player per match. Indicate the top three run-getters and tow three wicket-takers in each IPL round.
3. Fit the most appropriate distribution for runs scored and wickets taken by the top three batsmen and bowlers in the lost three IPL tournaments.
4. Find the relationship between a player’s performance and the salary he gets in your data.

# BUSINESS SIGNIFICANCE

. Understanding the link between player performance and salary has important commercial ramifications for IPL clubs. Through the use of performance-based incentives, teams can design player contracts that maximize effort during games and ensure that investments positively correlate with on-field performance. It also helps teams allocate budgets more effectively by identifying key players who deliver the most value. By supporting data-driven choices in hiring fresh talent based on statistical proof of success, it also helps with talent scouting. Franchises are able to make better selections about squad makeup and strategic planning by precisely predicting individual performances by fitting suitable distributions for wickets and runs.

# RESULTS AND INTERPRETATION

## Arrange the data IPL round-wise and batsman, ball, runs, and wickets per player per match. Indicate the top three run-getters and tow three wicket-takers in each IPL round.

*.*

Code and Result:

player\_runs = grouped\_data.groupby(['Season', 'Striker'])['runs\_scored'].sum().reset\_index()

player\_wickets = grouped\_data.groupby(['Season', 'Bowler'])['wicket\_confirmation'].sum().reset\_index()

player\_runs[player\_runs['Season']=='2023'].sort\_values(by='runs\_scored',ascending=False)

top\_run\_getters = player\_runs.groupby('Season').apply(lambda x: x.nlargest(3, 'runs\_scored')).reset\_index(drop=True)

bottom\_wicket\_takers = player\_wickets.groupby('Season').apply(lambda x: x.nlargest(3, 'wicket\_confirmation')).reset\_index(drop=True)

print("Top Three Run Getters:")

print(top\_run\_getters)

print("Top Three Wicket Takers:")

print(bottom\_wicket\_takers)

Top Three Run Getters:

Season Striker runs\_scored

0 2007/08 SE Marsh 616

1 2007/08 G Gambhir 534

2 2007/08 ST Jayasuriya 514

3 2009 ML Hayden 572

4 2009 AC Gilchrist 495

5 2009 AB de Villiers 465

6 2009/10 SR Tendulkar 618

7 2009/10 JH Kallis 572

8 2009/10 SK Raina 528

9 2011 CH Gayle 608

10 2011 V Kohli 557

11 2011 SR Tendulkar 553

12 2012 CH Gayle 733

13 2012 G Gambhir 590

14 2012 S Dhawan 569

15 2013 MEK Hussey 733

16 2013 CH Gayle 720

17 2013 V Kohli 639

18 2014 RV Uthappa 660

19 2014 DR Smith 566

20 2014 GJ Maxwell 552

21 2015 DA Warner 562

22 2015 AM Rahane 540

23 2015 LMP Simmons 540

24 2016 V Kohli 973

25 2016 DA Warner 848

26 2016 AB de Villiers 687

27 2017 DA Warner 641

28 2017 G Gambhir 498

29 2017 S Dhawan 479

30 2018 KS Williamson 735

31 2018 RR Pant 684

32 2018 KL Rahul 659

33 2019 DA Warner 692

34 2019 KL Rahul 593

35 2019 Q de Kock 529

36 2020/21 KL Rahul 676

37 2020/21 S Dhawan 618

38 2020/21 DA Warner 548

39 2021 RD Gaikwad 635

40 2021 F du Plessis 633

41 2021 KL Rahul 626

42 2022 JC Buttler 863

43 2022 KL Rahul 616

44 2022 Q de Kock 508

45 2023 Shubman Gill 890

46 2023 F du Plessis 730

47 2023 DP Conway 672

48 2024 RD Gaikwad 509

49 2024 V Kohli 500

50 2024 B Sai Sudharsan 418

Top Three Wicket Takers:

Season Bowler wicket\_confirmation

0 2007/08 Sohail Tanvir 24

1 2007/08 IK Pathan 20

2 2007/08 JA Morkel 20

3 2009 RP Singh 26

4 2009 A Kumble 22

5 2009 A Nehra 22

6 2009/10 PP Ojha 22

7 2009/10 A Mishra 20

8 2009/10 Harbhajan Singh 20

9 2011 SL Malinga 30

10 2011 MM Patel 22

11 2011 S Aravind 22

12 2012 M Morkel 30

13 2012 SP Narine 29

14 2012 SL Malinga 25

15 2013 DJ Bravo 34

16 2013 JP Faulkner 33

17 2013 R Vinay Kumar 27

18 2014 MM Sharma 26

19 2014 SP Narine 22

20 2014 B Kumar 21

21 2015 DJ Bravo 28

22 2015 SL Malinga 26

23 2015 A Nehra 25

24 2016 B Kumar 24

25 2016 SR Watson 23

26 2016 YS Chahal 22

27 2017 B Kumar 28

28 2017 JD Unadkat 27

29 2017 JJ Bumrah 23

30 2018 AJ Tye 28

31 2018 S Kaul 24

32 2018 Rashid Khan 23

33 2019 K Rabada 29

34 2019 Imran Tahir 26

35 2019 JJ Bumrah 23

36 2020/21 K Rabada 32

37 2020/21 JJ Bumrah 30

38 2020/21 TA Boult 26

39 2021 HV Patel 35

40 2021 Avesh Khan 27

41 2021 JJ Bumrah 22

42 2022 YS Chahal 29

43 2022 PWH de Silva 27

44 2022 K Rabada 23

45 2023 MM Sharma 31

46 2023 Mohammed Shami 28

47 2023 Rashid Khan 28

48 2024 HV Patel 19

49 2024 Mukesh Kumar 15

50 2024 Arshdeep Singh 14

Interpretation: The data reveals the top three IPL run-scorers and wicket-takers for each season. For 2023, Shubman Gill scored the most runs (890), while Mohit Sharma was the leading wicket-taker (31). Over the years, players like Virat Kohli and David Warner frequently appear among top run-scorers, and bowlers like Jasprit Bumrah and Bhuvneshwar Kumar among top wicket-takers.

*.*

## Fit the most appropriate distribution for runs scored and wickets taken by the top three batsmen and bowlers in the last three IPL tournaments.

ipl\_year\_id = pd.DataFrame(columns=["id", "year"])

ipl\_year\_id["id"] = ipl\_bbb["Match id"]

ipl\_year\_id["year"] = pd.to\_datetime(ipl\_bbb["Date"], dayfirst=True).dt.yea

#create a copy of ipl\_bbbc dataframe

ipl\_bbbc= ipl\_bbb.copy()

ipl\_bbbc['year'] = pd.to\_datetime(ipl\_bbb["Date"], dayfirst=True).dt.year

ipl\_bbbc[["Match id", "year", "runs\_scored","wicket\_confirmation","Bowler",'Striker']].head()

import scipy.stats as st

def get\_best\_distribution(data):

dist\_names = ['alpha','beta','betaprime','burr12','crystalball',

'dgamma','dweibull','erlang','exponnorm','f','fatiguelife',

'gamma','gengamma','gumbel\_l','johnsonsb','kappa4',

'lognorm','nct','norm','norminvgauss','powernorm','rice',

'recipinvgauss','t','trapz','truncnorm']

dist\_results = []

params = {}

for dist\_name in dist\_names:

dist = getattr(st, dist\_name)

param = dist.fit(data)

params[dist\_name] = param

# Applying the Kolmogorov-Smirnov test

D, p = st.kstest(data, dist\_name, args=param)

print("p value for "+dist\_name+" = "+str(p))

dist\_results.append((dist\_name, p))

# select the best fitted distribution

best\_dist, best\_p = (max(dist\_results, key=lambda item: item[1]))

# store the name of the best fit and its p value

print("\nBest fitting distribution: "+str(best\_dist))

print("Best p value: "+ str(best\_p))

print("Parameters for the best fit: "+ str(params[best\_dist]))

return best\_dist, best\_p, params[best\_dist]

total\_run\_each\_year = ipl\_bbbc.groupby(["year", "Striker"])["runs\_scored"].sum().reset\_index(

total\_run\_each\_year.sort\_values(["year", "runs\_scored"], ascending=False, inplace=True)

print(total\_run\_each\_year)

list\_top\_batsman\_last\_three\_year = {}

for i in total\_run\_each\_year["year"].unique()[:3]:

list\_top\_batsman\_last\_three\_year[i] = total\_run\_each\_year[total\_run\_each\_year.year == i][:3]["Striker"].unique().tolist()

list\_top\_batsman\_last\_three\_year

import warnings

warnings.filterwarnings('ignore')

runs = ipl\_bbbc.groupby(['Striker','Match id'])[['runs\_scored']].sum().reset\_index()

for key in list\_top\_batsman\_last\_three\_year:

for Striker in list\_top\_batsman\_last\_three\_year[key]:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("year:", key, " Batsman:", Striker)

get\_best\_distribution(runs[runs["Striker"] == Striker]["runs\_scored"])

print("\n\n")

total\_wicket\_each\_year = ipl\_bbbc.groupby(["year", "Bowler"])["wicket\_confirmation"].sum().reset\_index()

total\_wicket\_each\_year.sort\_values(["year", "wicket\_confirmation"], ascending=False, inplace=True)

print(total\_wicket\_each\_year)

list\_top\_bowler\_last\_three\_year = {}

for i in total\_wicket\_each\_year["year"].unique()[:3]:

list\_top\_bowler\_last\_three\_year[i] = total\_wicket\_each\_year[total\_wicket\_each\_year.year == i][:3]["Bowler"].unique().tolist()

list\_top\_bowler\_last\_three\_year

*#TOP RUN GETTERS*

2024: RD Gaikwad, V Kohli, B Sai Sudharsan

2023: Shubman Gill, F du Plessis, DP Conway

2022: JC Buttler, KL Rahul, Q de Kock

*#TOP WICKET TAKERS*

2024: HV Patel, Mukesh Kumar, Arshdeep Singh

2023: MM Sharma, Mohammed Shami, Rashid Khan

2022: YS Chahal, PWH de Silva, K Rabada

Interpretation: The best-fitting distributions for runs scored by top batsmen and wickets taken by top bowlers in the last three IPL seasons were identified. For 2024, top batsmen RD Gaikwad, V Kohli, and B Sai Sudharsan, and top bowlers HV Patel, Mukesh Kumar, and Arshdeep Singh. For 2023, Shubman Gill, F du Plessis, DP Conway, and MM Sharma, Mohammed Shami, Rashid Khan. For 2022, JC Buttler, KL Rahul, Q de Kock, and YS Chahal, PWH de Silva, K Rabada.

## Find the relationship between a player’s performance and the salary he gets in your data

Code and Result:

# Filter the wickets taken by Avesh Khan

Avesh\_khan\_wickets = wickets[wickets["Bowler"] == "Avesh Khan"]["wicket\_confirmation"]

# Fit the distribution to Avesh khan's wickets taken

get\_best\_distribution(Avesh\_Khan\_wickets)

**Relationship between the performance of a player and the salary he gets**

R2024 =total\_run\_each\_year[total\_run\_each\_year['year']==2024]

pip install fuzzywuzzy

from fuzzywuzzy import process

# Convert to DataFrame

df\_salary = ipl\_salary.copy()

df\_runs = R2024.copy()

# Function to match names

def match\_names(name, names\_list):

match, score = process.extractOne(name, names\_list)

return match if score >= 80 else None # Use a threshold score of 80

# Create a new column in df\_salary with matched names from df\_runs

df\_salary['Matched\_Player'] = df\_salary['Player'].apply(lambda x: match\_names(x, df\_runs['Striker'].tolist()))

# Merge the DataFrames on the matched names

df\_merged = pd.merge(df\_salary, df\_runs, left\_on='Matched\_Player', right\_on='Striker')

df\_merged.info()

# Calculate the correlation

correlation = df\_merged['Rs'].corr(df\_merged['runs\_scored'])

print("Correlation between Salary and Runs:", correlation)

Interpretation: The correlation between a player's salary and their runs scored in the 2024 IPL season is 0.27645, indicating a weak positive relationship. This suggests that higher-paid players tend to score more runs, but the correlation is not strong, implying that salary is not a significant predictor of performance in terms of runs scored.