VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A1b: Preliminary preparation and analysis of data- Descriptive statistics

FERAH SHAN SHANAVAS RABIYA

V01101398

Date of Submission: 16-06-2024

CONTENTS

Introduction

Based on statistics from the Indian Premier League (IPL) spanning 18 seasons from 2007 to 2024, this report provides an analytical study. This analysis's main goal is to get significant insights from the IPL datasets, which contain a variety of player performance measures like runs scored, wickets taken, and player salaries.

A number of critical tasks are included in the assignment, such as data extraction, data preprocessing, and carrying out in-depth statistical analyses. These assignments are meant to give a thorough grasp of how players performed during several IPL rounds, pinpoint the best players, and investigate the connection between salary and player performance.

Objectives

- Data Extraction and Organization: Extract data from the files that have been provided, organize it in an IPL round-by-round format, and summarize important variables such as the number of runs, wickets, and player statistics for each match.
- Identifying the Top Performers: In each round of the Indian Premier League, it is important to identify the top three players in terms of both run-scoring and wicket-taking. This will allow you to emphasize the individuals who have excelled throughout the season.
- Statistical Distribution Fitting: Fit the most appropriate statistical distributions for the number of runs scored and wickets taken by the top three batsmen and bowlers throughout the course of the last three Indian Premier League seasons.
- Performance-salary relationship: In order to identify how a player's earnings are affected by their performance on the field, it is necessary to conduct an analysis of the link that exists between a player's performance measurements and their salaries.
- Comparison of incomes: In order to compare the incomes of the top 10 batsmen and the top 10 wicket-taking bowlers over the course of the past three years, it is important to evaluate whether or not there are any notable disparities in their earnings.

Business Significance

The purpose of this study is to provide significant insights into player performances, the economic aspects of the Indian Premier League (IPL), and how statistical analysis may be used to sports data in order to obtain conclusions that can be put into action. In order to guarantee reliable and exhaustive conclusions, the analysis makes use of a wide range of statistical and data visualization methodological approaches.

Overall, this research not only highlights the top performers in the Indian Premier League (IPL), but it also dives into the financial consequences of player performances. As a result, it provides a comprehensive assessment of the relationship between athletic accomplishments and economic benefits in one of the most popular cricket leagues in the world.

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.**

```
grouped_data = ipl_bbb.groupby(['Season', 'Innings No', 'Striker', 'Bowler'
]).agg({'runs_scored': sum, 'wicket_confirmation': sum}).reset_index()

player_runs = grouped_data.groupby(['Season', 'Striker'])['runs_scored'].su
m().reset_index()
player_wickets = grouped_data.groupby(['Season', 'Bowler'])['wicket_confirm
ation'].sum().reset_index()

player_runs[player_runs['Season']=='2023'].sort_values(by='runs_scored',as
cending=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.n
largest(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 provides insights into trends over the years, identifying players who consistently performed well and contributed significantly to their teams' success. In the case of top three run getters, players like Virat Kohli, David Warner and Chris Gayle appear multiple times across different seasons, indicating their consistency and dominance in run-scoring over the years. The list highlights the impact of key players on their teams' performances across seasons, showcasing their role in shaping outcomes through their batting prowess. Likewise, in the case of top three wicket takers, Bowlers like Jasprit Bumrah, Kagiso Rabada and Dwayne Bravo feature prominently, demonstrating their skill in taking wickets consistently across different seasons. The list includes spinners like Imran Tahir and Harbhajan Singh, as well as fast bowlers such as Bhuvneshwar Kumar and Mitchell Starc, showcasing a mix of bowling styles and specialties. The wicket-taking ability of these bowlers indicates their crucial role in restricting opponents and influencing match outcomes through their bowling performances.

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

```
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_r
un_each_year.year == i][:3]["Striker"].unique().tolist()
list_top_batsman_last_three_year
```

```
{2024: ['RD Gaikwad', 'V Kohli', 'B Sai Sudharsan'],
 2023: ['Shubman Gill', 'F du Plessis', 'DP Conway'],
 2022: ['JC Buttler', 'KL Rahul', 'Q de Kock']}
```

```
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_sc
ored"])
        print("\n\n")
```

```
************************
year: 2024  Batsman: RD Gaikwad
p value for alpha = 2.599259711013304e-20
```

```
p value for beta = 0.02041902689492492
p value for betaprime = 0.019503763598668566
p value for burr12 = 0.46882020698395865
p value for crystalball = 0.2495364698727055
p value for dgamma = 0.15707438431209653
p value for dweibull = 0.20046582403736823
p value for erlang = 1.893799588395604e-06
p value for exponnorm = 0.4644304230917985
p value for f = 1.3560920695663998e-07
p value for fatiguelife = 1.304427037367869e-14
p value for gamma = 0.005830868576003678
p value for gengamma = 0.015331622187827243
p value for gumbel_l = 0.05546236480086464
p value for johnsonsb = 4.646964117947127e-13
p value for kappa4 = 0.006363220770325362
p value for lognorm = 1.1719355665219537e-16
p value for nct = 0.5881570496217812
p value for norm = 0.24953651809309751
p value for norminvgauss = 0.5538573365184996
p value for powernorm = 0.1788753268739086
p value for rice = 0.1828753218433654
p value for recipinvgauss = 0.06459275668874154
p value for t = 0.2494021485911212
p value for trapz = 7.476391685388162e-13
p value for truncnorm = 0.24173236832621992

Best fitting distribution: nct
Best p value: 0.5881570496217812
Parameters for the best fit: (5.718048022849898, 9.399490726283615, -54.252
77343780452, 8.497060689079994)



************************
year: 2024  Batsman: V Kohli
p value for alpha = 0.15371704349416937
p value for beta = 0.7807091136830002
p value for betaprime = 0.15634788776461095
p value for burr12 = 0.2201385645469427
p value for crystalball = 0.0013439120565839657
p value for dgamma = 0.00010919434981556638
p value for dweibull = 0.00012533056352014233
p value for erlang = 1.7690285330312436e-06
p value for exponnorm = 0.19376408619173924
p value for f = 2.67581083049327e-28
p value for fatiguelife = 0.11580928039819094
p value for gamma = 0.00878530144799014
p value for gengamma = 0.12789719547406364
p value for gumbel_l = 9.544555237684654e-09
p value for johnsonsb = 0.6600676697983927
p value for kappa4 = 7.270307243307106e-18
p value for lognorm = 6.635544190553261e-64
p value for nct = 0.1460773085917223
```

```
p value for norm = 0.0013439146566564463
p value for norminvgauss = 0.16537494306738054
p value for powernorm = 0.001959224898154651
p value for rice = 0.0019496833019799402
p value for recipinvgauss = 0.08835236633247623
p value for t = 0.001870132740059356
p value for trapz = 3.7326843413039495e-73
p value for truncnorm = 0.08872852288813304

Best fitting distribution: beta
Best p value: 0.7807091136830002
Parameters for the best fit: (0.816277299300862, 2.3391761669196907, -3.025
1144495756596e-31, 130.79371484721577)



***********************
year: 2024  Batsman: B Sai Sudharsan
p value for alpha = 0.9519530946513592
p value for beta = 0.2800374272685796
p value for betaprime = 0.7272275700648236
p value for burr12 = 0.03413730383965219
p value for crystalball = 0.835174953613428
p value for dgamma = 0.9003132708081405
p value for dweibull = 0.8965770306228721
p value for erlang = 0.2710277691398305
p value for exponnorm = 0.8246418777999891
p value for f = 0.9743698554720728
p value for fatiguelife = 0.8259440652110397
p value for gamma = 0.004088711345359375
p value for gengamma = 0.029688848326628436
p value for gumbel_l = 0.391243924609637
p value for johnsonsb = 0.6775536294207896
p value for kappa4 = 0.04273156928199129
p value for lognorm = 0.9006026891568572
p value for nct = 0.9627359408368513
p value for norm = 0.8351750214399875
p value for norminvgauss = 0.8696382419018381
p value for powernorm = 0.837790705015941
p value for rice = 0.8419161308192361
p value for recipinvgauss = 0.7846020832234206
p value for t = 0.8945403499225024
p value for trapz = 4.962305050994183e-07
p value for truncnorm = 0.8112138570439418

Best fitting distribution: f
Best p value: 0.9743698554720728
Parameters for the best fit: (7.230079711691059, 94.80999484543659, -0.4687
0159044880233, 39.84202109781083)



***********************
```

```
year: 2023  Batsman: Shubman Gill
p value for alpha = 0.19370998562525277
p value for beta = 0.35556757767764935
p value for betaprime = 0.3320890781747331
p value for burr12 = 0.17538338566759115
p value for crystalball = 0.04047310237062518
p value for dgamma = 0.004654508243065125
p value for dweibull = 0.011388953681876424
p value for erlang = 0.10415431199992453
p value for exponnorm = 0.4076479842986115
p value for f = 1.211921514554867e-19
p value for fatiguelife = 0.2203915030909802
p value for gamma = 0.01932605267751175
p value for gengamma = 0.15830394669705838
p value for gumbel_l = 0.00016365306017313027
p value for johnsonsb = 0.6214006077216168
p value for kappa4 = 8.537718673686839e-12
p value for lognorm = 3.0444374367609376e-26
p value for nct = 0.10819705795130274
p value for norm = 0.0404730725346123
p value for norminvgauss = 0.2256809493002525
p value for powernorm = 0.008933578018930133
p value for rice = 0.009231529839363262
p value for recipinvgauss = 0.25695076184687626
p value for t = 0.06288757117420063
p value for trapz = 7.559368072972744e-39
p value for truncnorm = 0.03322263046428764

Best fitting distribution: johnsonsb
Best p value: 0.6214006077216168
Parameters for the best fit: (1.127462972555547, 0.7082040622620326, -1.078
5135120261573, 140.5794643798755)



************************
year: 2023  Batsman: F du Plessis
p value for alpha = 2.6514415564811303e-46
p value for beta = 0.5913252599657466
p value for betaprime = 0.21607006903997794
p value for burr12 = 1.4054517820032704e-09
p value for crystalball = 0.17738239944644252
p value for dgamma = 0.0192505709952403
p value for dweibull = 0.11610399857369136
p value for erlang = 1.5300500072467267e-05
p value for exponnorm = 0.029960734734523542
p value for f = 2.3763783336197345e-18
p value for fatiguelife = 0.4484315774329326
p value for gamma = 2.658122267546294e-07
p value for gengamma = 0.02408727588734938
p value for gumbel_l = 0.0014475463566171465
p value for johnsonsb = 0.18738807412325909
p value for kappa4 = 7.855215717595119e-07
```

```
p value for lognorm = 7.76777670084355e-36
p value for nct = 0.3074928968583557
p value for norm = 0.17738241885083328
p value for norminvgauss = 0.5294908193576565
p value for powernorm = 0.10747661134694209
p value for rice = 0.10596246415943456
p value for recipinvgauss = 0.25232880325823404
p value for t = 0.17742481659951348
p value for trapz = 2.2917131806009114e-31
p value for truncnorm = 0.4976264771179164

Best fitting distribution: beta
Best p value: 0.5913252599657466
Parameters for the best fit: (0.964930449377772, 2.3654747855916978, -2.497
9006319546827e-31, 110.45316400426368)



************************
year: 2023  Batsman: DP Conway
p value for alpha = 0.24224437379078456
p value for beta = 0.9335739280635688
p value for betaprime = 0.5939028036769798
p value for burr12 = 0.031686490382365484
p value for crystalball = 0.5919833978299178
p value for dgamma = 0.659050680685497
p value for dweibull = 0.47709033274534696
p value for erlang = 0.5856582107400496
p value for exponnorm = 0.5919442519144027
p value for f = 0.03191068848461143
p value for fatiguelife = 2.4470875845519328e-05
p value for gamma = 0.5772798774478447
p value for gengamma = 0.010638224653254702
p value for gumbel_l = 0.6434008985606366
p value for johnsonsb = 0.0010884744390042833
p value for kappa4 = 0.39160448071756937
p value for lognorm = 3.1507840694396127e-06
p value for nct = 0.5925999092825844
p value for norm = 0.5919834368439854
p value for norminvgauss = 0.5925748844419921
p value for powernorm = 0.45248629955798125
p value for rice = 0.45768623194758373
p value for recipinvgauss = 0.031005955700378007
p value for t = 0.5919821236916709
p value for trapz = 0.002896838839657856
p value for truncnorm = 0.2820881279467663

Best fitting distribution: beta
Best p value: 0.9335739280635688
Parameters for the best fit: (0.6250316512826838, 0.6786342050356671, -3.47
41633120498916, 95.47416331204991)
```

```
************************
year: 2022  Batsman: JC Buttler
p value for alpha = 3.235109657468491e-34
p value for beta = 0.33455794816369444
p value for betaprime = 0.0040250475185371615
p value for burr12 = 0.7069656630104211
p value for crystalball = 0.004608459861307201
p value for dgamma = 0.00604199317470544
p value for dweibull = 0.002843068054754871 5
p value for erlang = 0.0018449508774974754
p value for exponnorm = 0.7137955109895673
p value for f = 3.9553917967759444e-17
p value for fatiguelife = 0.38179178822012705
p value for gamma = 0.0007081454329517234
p value for gengamma = 0.30583328083419026
p value for gumbel_l = 0.00010416429669054019
p value for johnsonsb = 0.5217216451704005
p value for kappa4 = 1.0421737381705364e-12
p value for lognorm = 5.0571684202935185e-28
p value for nct = 0.45209196275779084
p value for norm = 0.004608461486487414
p value for norminvgauss = 0.4852525149516915
p value for powernorm = 0.004689395332742374
p value for rice = 0.004972139278291876
p value for recipinvgauss = 0.2745923469661913
p value for t = 0.007226707680555
p value for trapz = 8.531784262849386e-37
p value for truncnorm = 0.038943153796554775

Best fitting distribution: exponnorm
Best p value: 0.7137955109895673
Parameters for the best fit: (3054.885295608514, -0.031805252610631926, 0.0
1119090499814962)



************************
year: 2022  Batsman: KL Rahul
p value for alpha = 3.439822697019343e-50
p value for beta = 0.3005191042009908
p value for betaprime = 0.3083252430394988
p value for burr12 = 0.46187713102710526
p value for crystalball = 0.02169172684247167
p value for dgamma = 0.06770258558041709
p value for dweibull = 0.10186919378179626
p value for erlang = 0.5713953642722212
p value for exponnorm = 0.21607213755074883
p value for f = 3.271576641222778e-23
p value for fatiguelife = 0.4121975839714658
p value for gamma = 0.5713982751559553
p value for gengamma = 0.16010152392031385
p value for gumbel_l = 0.001680677455102142
```

```
p value for johnsonsb = 0.9402453631468569
p value for kappa4 = 1.3895397566735892e-07
p value for lognorm = 9.796218603186654e-32
p value for nct = 0.20349727522799965
p value for norm = 0.02169172706709699
p value for norminvgauss = 0.38170378589734333
p value for powernorm = 0.026645565499311186
p value for rice = 0.027062729391134077
p value for recipinvgauss = 0.4426895366659932
p value for t = 0.02169408819105212
p value for trapz = 1.8532732379092856e-35
p value for truncnorm = 0.6753901355264902

Best fitting distribution: johnsonsb
Best p value: 0.9402453631468569
Parameters for the best fit: (0.9331207997896902, 0.7776389044559282, -2.34
5202857963142, 143.0833194837059)



************************
year: 2022  Batsman: Q de Kock
p value for alpha = 0.22421213312317712
p value for beta = 0.2878667203270271
p value for betaprime = 0.057402804910011485
p value for burr12 = 0.4931279667432148
p value for crystalball = 0.05846912701914453
p value for dgamma = 0.0014560083713105465
p value for dweibull = 0.010478670398011536
p value for erlang = 0.08677035591445126
p value for exponnorm = 0.43726373790797446
p value for f = 4.2346585152678845e-12
p value for fatiguelife = 0.12498847851930361
p value for gamma = 0.027350558506526124
p value for gengamma = 0.0926892512677634
p value for gumbel_l = 9.485045980257123e-06
p value for johnsonsb = 0.3450941869097196
p value for kappa4 = 3.832745782875419e-18
p value for lognorm = 2.3658846096591403e-28
p value for nct = 0.2843302460638113
p value for norm = 0.058469111112182226
p value for norminvgauss = 0.2268711891858597
p value for powernorm = 0.033823716873628396
p value for rice = 0.03349090516310227
p value for recipinvgauss = 0.1073883725317526
p value for t = 0.041656498991066715
p value for trapz = 3.947363741930107e-50
p value for truncnorm = 0.08860764609495919

Best fitting distribution: burr12
Best p value: 0.4931279667432148
Parameters for the best fit: (590926023.7998527, 0.05483081555360233, -9698
03927.022117, 969803927.160071)
```

Interpretation:

Being included in these compilations generally indicates acknowledgment for exceptional contributions to their teams or remarkable accomplishments in competitions held during those years. These lists are commonly utilized to honor players who have had a notable influence on match outcomes and team standings through their batting performance. Each batsman has been analyzed for their batting performance using different statistical distributions. the data provided offers a statistical perspective on the batting performance of prominent cricket players across different years, using various distributions to capture the nuances of their performance data.

```
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_wicke
t_each_year.year == i][:3]["Bowler"].unique().tolist()
list_top_bowler_last_three_year
```

```
{2024: ['HV Patel', 'Mukesh Kumar', 'Arshdeep Singh'],
 2023: ['MM Sharma', 'Mohammed Shami', 'Rashid Khan'],
 2022: ['YS Chahal', 'PWH de Silva', 'K Rabada']}
```

```
import warnings
warnings.filterwarnings('ignore')
wickets = ipl_bbbc.groupby(['Bowler','Match id'])[['wicket_confirmation']].
sum().reset_index()

for key in list_top_bowler_last_three_year:
    for bowler in list_top_bowler_last_three_year[key]:
        print("************************")
        print("year:", key, " Bowler:", bowler)
        get_best_distribution(wickets[wickets["Bowler"] == bowler]["wicket_
confirmation"])
        print("\n\n")
```

```
************************
year: 2024  Bowler: HV Patel
p value for alpha = 0.0002993252328930706
p value for beta = 2.777571908776589e-19
p value for betaprime = 1.7052883875145053e-30
p value for burr12 = 5.427998338605459e-15
p value for crystalball = 1.1109118198587684e-05
p value for dgamma = 4.375428528574276e-05
p value for dweibull = 1.8553295107771936e-05
p value for erlang = 5.473635282991912e-24
p value for exponnorm = 0.0002813279943461815
p value for f = 1.9012983291282487e-09
p value for fatiguelife = 1.9734428958773156e-05
p value for gamma = 1.470787431589663e-16
p value for gengamma = 1.4345058849022962e-16
p value for gumbel_l = 4.541523588271283e-05
p value for johnsonsb = 2.827201329331457e-51
p value for kappa4 = 9.177530010006471e-23
```

```
p value for lognorm = 5.2162358572043325e-22
p value for nct = 0.0001960277304576293
p value for norm = 1.1109124960635979e-05
p value for norminvgauss = 3.811196478020768e-05
p value for powernorm = 3.2186417463058256e-05
p value for rice = 3.354567282896991e-05
p value for recipinvgauss = 5.05058721389515e-12
p value for t = 9.451105792399515e-05
p value for trapz = 1.0447243016629734e-51
p value for truncnorm = 0.0002182292327632623

Best fitting distribution: alpha
Best p value: 0.0002993252328930706
Parameters for the best fit: (5.200800514990576, -4.106246473111661, 27.580
368990504883)



************************
year: 2024  Bowler: Mukesh Kumar
p value for alpha = 0.6028771589628603
p value for beta = 0.01195401496533166
p value for betaprime = 0.001059893235946907
p value for burr12 = 0.13577547952316893
p value for crystalball = 0.2874602836058904
p value for dgamma = 0.31965148068347327
p value for dweibull = 0.34346643238289587
p value for erlang = 1.0115032724485677e-06
p value for exponnorm = 0.5154597105302978
p value for f = 0.11745949856748239
p value for fatiguelife = 0.30877430134651196
p value for gamma = 0.009841759821405782
p value for gengamma = 0.07933719921899518
p value for gumbel_l = 0.25997636144422587
p value for johnsonsb = 0.0878807795320421
p value for kappa4 = 0.058739565059041765
p value for lognorm = 0.00048729251059054235
p value for nct = 0.5480580718802858
p value for norm = 0.2874600799525868
p value for norminvgauss = 0.3895684674359622
p value for powernorm = 0.39511432172869
p value for rice = 0.3950169895189477
p value for recipinvgauss = 0.025198651172109288
p value for t = 0.2874574742538948
p value for trapz = 9.722628535925783e-06
p value for truncnorm = 0.2598105493516787

Best fitting distribution: alpha
Best p value: 0.6028771589628603
Parameters for the best fit: (6.113363581345144, -5.245777123804531, 39.577
45263632695)
```

```
************************
year: 2024  Bowler: Arshdeep Singh
p value for alpha = 0.002547644307209551
p value for beta = 3.7725133611153275e-15
p value for betaprime = 5.062381659741898e-22
p value for burr12 = 4.603956720503075e-14
p value for crystalball = 0.0002501762149918564
p value for dgamma = 0.00028566200697101806
p value for dweibull = 0.0016211491850549598
p value for erlang = 2.269289539862191e-12
p value for exponnorm = 0.0019097947631203649
p value for f = 0.000227258408802241
p value for fatiguelife = 2.169103029961132e-15
p value for gamma = 6.618486511618167e-29
p value for gengamma = 5.948936850168967e-23
p value for gumbel_l = 0.00026864389982599567
p value for johnsonsb = 5.472387372640376e-24
p value for kappa4 = 8.181970339328129e-12
p value for lognorm = 1.9909678840157557e-12
p value for nct = 0.001425707010244702
p value for norm = 0.00025017539197677184
p value for norminvgauss = 0.0001290021448063343
p value for powernorm = 0.00047137775975730436
p value for rice = 0.00047472774494963083
p value for recipinvgauss = 1.9623061606588953e-10
p value for t = 0.004473243416688644
p value for trapz = 1.1911079182772876e-29
p value for truncnorm = 0.00034221379785853717

Best fitting distribution: t
Best p value: 0.004473243416688644
Parameters for the best fit: (4.822497644715119, 1.1162819391895469, 0.9153
269129308039)
```

```
************************
year: 2023  Bowler: MM Sharma
p value for alpha = 5.261792307574885e-09
p value for beta = 3.369903415982389e-18
p value for betaprime = 3.4236065288569164e-34
p value for burr12 = 7.707563359968149e-27
p value for crystalball = 5.614290141391915e-05
p value for dgamma = 1.0498635614441156e-05
p value for dweibull = 2.4126502201215078e-05
p value for erlang = 2.203151538560566e-17
p value for exponnorm = 7.116980583029457e-10
p value for f = 6.394862208673673e-10
p value for fatiguelife = 1.3371709463319658e-24
p value for gamma = 2.599880000032353e-21
p value for gengamma = 9.811276806787944e-14
p value for gumbel_l = 3.5245319536008275e-05
```

```
p value for johnsonsb = 2.4461951672713995e-40
p value for kappa4 = 1.804941215806713e-17
p value for lognorm = 1.7804559351656542e-19
p value for nct = 6.513780696080299e-05
p value for norm = 5.614083233477072e-05
p value for norminvgauss = 2.385888242491267e-11
p value for powernorm = 3.7448415090755237e-05
p value for rice = 3.8846082842387146e-05
p value for recipinvgauss = 1.932872667384276e-17
p value for t = 0.00012008020713636171
p value for trapz = 9.04818074400941e-47
p value for truncnorm = 6.39486602704708e-10

Best fitting distribution: t
Best p value: 0.00012008020713636171
Parameters for the best fit: (29.05846643939152, 1.2878076424619436, 1.1974
04368883093)



************************
year: 2023  Bowler: Mohammed Shami
p value for alpha = 0.0005609846480252995
p value for beta = 8.949702621553806e-16
p value for betaprime = 1.0457228098472159e-27
p value for burr12 = 3.809437306589196e-09
p value for crystalball = 8.97379813361614e-06
p value for dgamma = 1.3065638273544516e-11
p value for dweibull = 1.0406851960138218e-05
p value for erlang = 8.670599832745995e-28
p value for exponnorm = 0.00047630665162716083
p value for f = 2.404756281608377e-07
p value for fatiguelife = 7.5219130194197114e-06
p value for gamma = 5.248327144461885e-42
p value for gengamma = 4.371554773381843e-42
p value for gumbel_l = 2.275582226089825e-06
p value for johnsonsb = 8.40193769288202e-62
p value for kappa4 = 5.440679375551408e-12
p value for lognorm = 8.538407160860825e-23
p value for nct = 0.0003740512893746841
p value for norm = 8.973880770320002e-06
p value for norminvgauss = 3.3178705246034226e-05
p value for powernorm = 0.00011849751955444802
p value for rice = 0.00011833002960228116
p value for recipinvgauss = 1.957916752902072e-07
p value for t = 8.972846375529713e-06
p value for trapz = 1.8983891174798298e-38
p value for truncnorm = 2.539236515610462e-06

Best fitting distribution: alpha
Best p value: 0.0005609846480252995
Parameters for the best fit: (6.734843933630203, -5.500744811228249, 44.826
257131250145)
```

```
***********************
year: 2023  Bowler: Rashid Khan
p value for alpha = 1.4259399000489275e-06
p value for beta = 8.8954046965209e-27
p value for betaprime = 3.407105814148136e-65
p value for burr12 = 2.5587675833251047e-18
p value for crystalball = 2.99049361738744e-09
p value for dgamma = 6.928485900596178e-10
p value for dweibull = 6.928168431614811e-10
p value for erlang = 1.052461604472364e-41
p value for exponnorm = 7.720335528170629e-07
p value for f = 4.940207066298226e-10
p value for fatiguelife = 1.4667845015790087e-07
p value for gamma = 3.120866167200452e-31
p value for gengamma = 3.3780076161228415e-35
p value for gumbel_l = 7.911140658362043e-09
p value for johnsonsb = 6.659510229977693e-18
p value for kappa4 = 6.390225516379688e-22
p value for lognorm = 6.677625232671758e-27
p value for nct = 8.389699838025371e-07
p value for norm = 2.9905103094429466e-09
p value for norminvgauss = 1.9883690059384983e-07
p value for powernorm = 5.69320390726131e-08
p value for rice = 6.008338811339319e-08
p value for recipinvgauss = 1.0204427503324627e-07
p value for t = 4.1495986291836466e-08
p value for trapz = 4.291139733358819e-55
p value for truncnorm = 3.0854549274395264e-07

Best fitting distribution: alpha
Best p value: 1.4259399000489275e-06
Parameters for the best fit: (5.783058438949956, -4.20986029264825, 30.8789
91656277478)



***********************
year: 2022  Bowler: YS Chahal
p value for alpha = 1.1180274965710719e-05
p value for beta = 1.0295677049868252e-44
p value for betaprime = 6.005755537239427e-40
p value for burr12 = 1.7979353447013811e-12
p value for crystalball = 5.1232708024114544e-08
p value for dgamma = 4.012289620255995e-08
p value for dweibull = 1.3446088982977968e-07
p value for erlang = 2.6044501249608127e-33
p value for exponnorm = 9.70188325365383e-06
p value for f = 4.3760412135414686e-11
p value for fatiguelife = 1.0610357499785987e-07
p value for gamma = 3.2021687139045712e-55
```

```
p value for gengamma = 4.0264602677437785e-26
p value for gumbel_l = 8.01003405037582e-08
p value for johnsonsb = 9.127045203599366e-44
p value for kappa4 = 5.8742872003226356e-27
p value for lognorm = 1.2869567438882943e-32
p value for nct = 5.296213377700368e-06
p value for norm = 5.1235707238843755e-08
p value for norminvgauss = 3.3808295582037935e-07
p value for powernorm = 1.021178511514112e-06
p value for rice = 1.0373024397997343e-06
p value for recipinvgauss = 1.53711078374615e-21
p value for t = 1.1782910213333637e-07
p value for trapz = 1.8568421933146807e-70
p value for truncnorm = 1.609035128404315e-07

Best fitting distribution: alpha
Best p value: 1.1180274965710719e-05
Parameters for the best fit: (6.054854001673274, -4.898293043793716, 36.817
47298117385)



************************
year: 2022  Bowler: PWH de Silva
p value for alpha = 0.20501605213397434
p value for beta = 6.089293734595811e-08
p value for betaprime = 3.597368592551267e-07
p value for burr12 = 2.7078633279028545e-05
p value for crystalball = 0.12578198773774552
p value for dgamma = 0.04130328255260218
p value for dweibull = 0.08384976427162982
p value for erlang = 0.0002485071992361352
p value for exponnorm = 0.3076424973571079
p value for f = 0.0065835107143813465
p value for fatiguelife = 0.0879596136953581
p value for gamma = 8.727963496024317e-05
p value for gengamma = 0.00519063892676308
p value for gumbel_l = 0.014493692496563626
p value for johnsonsb = 2.0634443260981352e-05
p value for kappa4 = 1.8620061578617215e-06
p value for lognorm = 5.934676005942877e-06
p value for nct = 0.18287627001224627
p value for norm = 0.12578246429025397
p value for norminvgauss = 0.10918449199764368
p value for powernorm = 0.1963520712744381
p value for rice = 0.1985929094578025
p value for recipinvgauss = 4.423190500679613e-05
p value for t = 0.1973319936827771
p value for trapz = 1.9360347216700493e-15
p value for truncnorm = 0.10632743012364088

Best fitting distribution: exponnorm
Best p value: 0.3076424973571079
```

```
Parameters for the best fit: (1.5651879172672551, 0.40254290759385924, 0.62
74498232929551)


************************
year: 2022  Bowler: K Rabada
p value for alpha = 0.017666063432803525
p value for beta = 4.443616547466671e-12
p value for betaprime = 4.702163459968348e-17
p value for burr12 = 1.0217952890763225e-11
p value for crystalball = 0.003016635703159909
p value for dgamma = 0.004039539567683215
p value for dweibull = 0.004897361468685357
p value for erlang = 6.666902843060855e-10
p value for exponnorm = 0.012447792991605588
p value for f = 6.634692021556237e-06
p value for fatiguelife = 0.011517197590084738
p value for gamma = 1.032396146883282e-12
p value for gengamma = 2.6816733980980167e-12
p value for gumbel_l = 0.00045795960689101544
p value for johnsonsb = 3.123503411674573e-12
p value for kappa4 = 2.016542974865221e-05
p value for lognorm = 2.015341179637063e-18
p value for nct = 0.01550593593647065
p value for norm = 0.003016639761756701
p value for norminvgauss = 0.011593590051028446
p value for powernorm = 0.012612430707673927
p value for rice = 0.012664345659931242
p value for recipinvgauss = 0.011156908993035786
p value for t = 0.0030166123509550724
p value for trapz = 2.238131859007279e-22
p value for truncnorm = 0.007005335434665971

Best fitting distribution: alpha
Best p value: 0.017666063432803525
Parameters for the best fit: (8.172744476082507, -7.746415964015842, 75.180
55369544504)
```

Interpretation:

The provided analysis uses statistical tests to identify the best-fitting distribution for the wicket data of top bowlers across different years in cricket. For each bowler in each year, a variety of probability distributions were tested (like alpha, beta, gamma, etc.). The "best fitting distribution" refers to the distribution that statistically best describes the pattern of wickets taken by a bowler in a particular year. This determination is based on the p-value associated with each distribution. The analysis provides a statistical framework to understand and predict the wicket-taking performance of top bowlers in cricket, leveraging the principles of probability distributions and their parameters.

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

```
import warnings
warnings.filterwarnings('ignore')

runs = ipl_bbbc.groupby(['Striker','Match id'])[['runs_scored']].sum(
).reset_index()
chosen_Striker = "RG Sharma"
print("")
print("Best fit distribution for wickets taken by:", chosen_Striker)
get_best_distribution(runs[runs["Striker"] == chosen_Striker]["runs_s
cored"])
print("\n\n")
```

```
Best fit distribution for wickets taken by: RG Sharma
p value for alpha = 4.266049939383962 6e-58
p value for beta = 0.11415840466328053
p value for betaprime = 0.08497957251270649
p value for burr12 = 7.308468380168264e-13
p value for crystalball = 0.00025185125549376415
p value for dgamma = 9.01396419525256e-06
p value for dweibull = 8.630302322439815e-07
p value for erlang = 0.0020042152764490513
p value for exponnorm = 0.25792765995841016
p value for f = 9.489851892820956e-44
p value for fatiguelife = 0.03266837625674257
p value for gamma = 1.725622048913048e-07
p value for gengamma = 0.028459845061022948
p value for gumbel_l = 6.206497964432987e-11
p value for johnsonsb = 0.358351649664293
p value for kappa4 = 7.412469126839709e-30
p value for lognorm = 4.6940510185140297e-66
p value for nct = 0.0748787798870133
p value for norm = 0.00025185154681253 54
p value for norminvgauss = 0.05593102898731506
p value for powernorm = 7.580125838068554e-05
p value for rice = 7.019444872806493e-05
p value for recipinvgauss = 1.1153131551816901e-05
p value for t = 0.00017966399509757345
p value for trapz = 7.953825110064789e-93
p value for truncnorm = 0.0007355631284158111

Best fitting distribution: johnsonsb
Best p value: 0.358351649664293
Parameters for the best fit: (1.0188059199310264, 0.6031242643208398, -
0.516788739160724, 112.44619735061926)
```

Interpretation:

Based on the analysis, the best-fit distribution for the wickets taken by RG Sharma is the Johnson SB distribution. Higher wickets taken might correlate with higher salary if performance directly influences pay. Johnson SB distribution parameters can help understand the distribution of wickets taken, potentially indicating how frequently certain performance levels occur. By understanding the distribution of wickets taken and its relationship with salary, you can provide insights into how performance metrics translate into compensation, which is crucial in sports analytics and player evaluation.

– Last three-year performance with latest salary 2024

```
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
df_salary['Matched_Player'] = df_salary['Player'].apply(lambda x: match_nam
es(x, df_runs['Striker'].tolist()))
df_merged = pd.merge(df_salary, df_runs, left_on='Matched_Player', right_on
='Striker')
```

Interpretation:

The code segment essentially facilitates matching players' names between two datasets (df_salary and df_runs) using fuzzy string matching. Fuzzy matching allows for flexibility in matching names that are not exact matches but are similar enough to be considered the same entity. This approach helps in ensuring data integrity and completeness when performing subsequent analyses or generating reports that require combined information from multiple datasets.

- **Significant Difference Between the Salaries of the Top 10 Batsmen and Top Wicket-Taking Bowlers Over the Last Three Years.**

```
correlation = df_merged['Rs'].corr(df_merged['runs_scored'])
print("Correlation between Salary and Runs:", correlation)
```

```
Correlation between Salary and Runs: 0.30612483765821674
```

Interpretation:

The correlation coefficient between salary (Rs) and runs scored (runs_scored) in df_merged is approximately 0.3061. This suggests a positive but moderate linear relationship between salary and runs scored. This means that while there is a discernible pattern where higher runs scored tend to be associated with higher salaries, the relationship is not extremely strong. For teams, analysts, or stakeholders in cricket, understanding this correlation can help in making informed decisions related to player contracts, negotiations, and team strategies. Players who consistently score higher runs might be viewed as more valuable assets, potentially justifying higher salary offers.

Recommendation

RG Sharma's performance indicators, including total runs and wickets, are compared to his income statistics. The analysis suggests that his salary should be adjusted to reflect his performance on the field. The ratio of performance to value should be assessed to determine if his current salary accurately reflects his total performance. Comparing his performance to other high-performing individuals in similar positions can provide valuable information on the competitiveness of his salary. Long-term performance trends should be monitored to identify trends in his performance consistency and its correlation with salary fluctuations. The analysis can be used in contract negotiations to discuss future contracts or wage modifications. Performance-based incentives should be considered to enhance motivation and ensure compensation accurately represents individual and team achievements.

R Codes

```
install.packages("pacman")
require("pacman")
library("pacman")
library(datasets)
install.packages("readr")
library(readr)
install.packages("dplyr")
library(dplyr)
install.packages("readxl")
library(readxl)

setwd('E:\\VCU\\Summer 2024\\Statistical Analysis & Modeling')
ball_by_ball_data = read.csv('IPL_ball_by_ball_updated till 2024.csv')
salary_data = read_excel('IPL SALARIES 2024.xlsx')

install.packages("tidyverse")
library(tidyverse)

player_stats <- ball_by_ball_data %>%
  group_by(Match.id, Season, Striker, Bowler) %>%
  summarise(
    runs = sum(runs_scored, na.rm = TRUE),
    wickets = sum(ifelse(!is.na(wicket_confirmation) & wicket_confirmation == 1, 1, 0), na.rm = TRUE)
  ) %>%
  ungroup()

top_wicket_takers <- player_stats %>%
  group_by(Season, Bowler) %>%
  summarise(total_wickets = sum(wickets, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(Season, desc(total_wickets)) %>%
  group_by(Season) %>%
  slice_head(n = 3)
print("Top Wicket Takers by Round:")
print(top_wicket_takers)

top_run_getters <- player_stats %>%
  group_by(Season, Striker) %>%
  summarise(total_runs = sum(runs, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(Season, desc(total_runs)) %>%
  group_by(Season) %>%
  slice_head(n = 3)
print("Top Run Getters by Round:")
print(top_run_getters)

install.packages("fitdistrplus")
library(fitdistrplus)

  for (season in unique(top_run_getters$Season)) {
    cat("\nSeason:", season, "\n")
    cat("Top 3 Batsmen:\n")
    season_data <- top_run_getters %>% filter(Season == season)
    print(season_data)
    print("Top Wicket Takers by Season:")
    for (season in unique(top_wicket_takers$Season)) {
      cat("\nSeason:", season, "\n")
      cat("Top 3 Wicket Takers:\n")

      install.packages("ggplot2")
      library(ggplot2)
      last_three_seasons <- c(2022, 2023, 2024)
      filtered_data <- ball_by_ball_data %>%
        filter(Season %in% last_three_seasons)
      player_stats <- filtered_data %>%
        group_by(Match.id, Season, Striker, Bowler) %>%
        summarise(
          runs = sum(runs_scored, na.rm = TRUE),
          wickets = sum(ifelse(!is.na(wicket_confirmation) & wicket_confirmation == 1, 1, 0), na.rm = TRUE)
        ) %>%
        ungroup()
      top_run_getters <- player_stats %>%
        group_by(Season, Striker) %>%
        summarise(total_runs = sum(runs, na.rm = TRUE)) %>%
        ungroup() %>%
        arrange(Season, desc(total_runs)) %>%
        group_by(Season) %>%
        slice_head(n = 3)
      top_wicket_takers <- player_stats %>%
        group_by(Season, Bowler) %>%
        summarise(total_wickets = sum(wickets, na.rm = TRUE)) %>%
        ungroup() %>%
        arrange(Season, desc(total_wickets)) %>%
        group_by(Season) %>%
        slice_head(n = 3)
      print("Top Run Getters by Season:")
      for (season in unique(top_run_getters$Season)) {
        cat("\nSeason:", season, "\n")
        cat("Top 3 Batsmen:\n")
        season_data <- top_run_getters %>% filter(Season == season)
        print(season_data)
      }
      print("Top Wicket Takers by Season:")
      for (season in unique(top_wicket_takers$Season)) {
        cat("\nSeason:", season, "\n")
        cat("Top 3 Wicket Takers:\n")
        season_data <- top_wicket_takers %>% filter(Season == season)
        print(season_data)
```

```
}
runs_data <- top_run_getters$total_runs
wickets_data <- top_wicket_takers$total_wickets
runs_data_exp <- runs_data[runs_data > 0]
fit_runs_norm <- fitdist(runs_data, "norm")
fit_runs_exp <- tryCatch(fitdist(runs_data_exp, "exp"), error = function(e) NULL)
fit_runs_pois <- fitdist(runs_data, "pois")
runs_fits <- list(norm = fit_runs_norm, pois = fit_runs_pois)
if (!is.null(fit_runs_exp)) {
  runs_fits$exp <- fit_runs_exp
}
goftest_runs <- gofstat(runs_fits)
print(goftest_runs)
wickets_data_exp <- wickets_data[wickets_data > 0]
fit_wickets_norm <- fitdist(wickets_data, "norm")
fit_wickets_exp <- tryCatch(fitdist(wickets_data_exp, "exp"), error = function(e) NULL)
fit_wickets_pois <- fitdist(wickets_data, "pois")
wickets_fits <- list(norm = fit_wickets_norm, pois = fit_wickets_pois)
if (!is.null(fit_wickets_exp)) {
  wickets_fits$exp <- fit_wickets_exp
}
goftest_wickets <- gofstat(wickets_fits)
print(goftest_wickets)
par(mfrow = c(2, 2))
plot(fit_runs_norm)
if (!is.null(fit_runs_exp)) {
  plot(fit_runs_exp)
}
plot(fit_runs_pois)
performance_metrics <- filtered_data %>%
  group_by(Striker) %>%
  summarise(
    total_runs = sum(runs_scored, na.rm = TRUE),
    matches_played = n_distinct(Match.id),
    average_runs = mean(runs_scored, na.rm = TRUE)
  ) %>%
  ungroup() %>%
  rename(Player = Striker)
bowler_metrics <- filtered_data %>%
  group_by(Bowler) %>%
  summarise(
    total_wickets = sum(ifelse(!is.na(wicket_confirmation) & wicket_confirmation == 1, 1, 0), na.rm = TRUE),
    matches_played = n_distinct(Match.id),
    average_wickets = mean(ifelse(!is.na(wicket_confirmation) & wicket_confirmation == 1, 1, 0), na.rm = TRUE)
  ) %>%
  ungroup() %>%
  rename(Player = Bowler)
combined_metrics <- full_join(performance_metrics, bowler_metrics, by = "Player")
performance_salary_data <- left_join(salary_data, combined_metrics, by = "Player")
performance_salary_data[is.na(performance_salary_data)] <- 0
rgsharma_data <- performance_salary_data %>% filter(Player == "RG Sharma")
print("RG Sharma's Performance and Salary Data:")
print(rgsharma_data)
ggplot(rgsharma_data, aes(x = Salary)) +
  geom_bar(aes(y = total_runs), stat = "identity", fill = "blue", alpha = 0.7) +
  geom_bar(aes(y = total_wickets), stat = "identity", fill = "red", alpha = 0.7) +
  labs(title = "RG Sharma's Performance Metrics and Salary",
       x = "Salary",
       y = "Performance Metrics",
       fill = "Metric") +
  theme_minimal() +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Total Wickets")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

References

1. www.github.com

2. www.geeksforgeeks.com