VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A1b: Preliminary preparation and analysis of data- Descriptive statistics

SAYA SANTHOSH

V01101901

Date of Submission: 18-06-2024

CONTENTS

Introduction

This document presents a comprehensive analysis of the IPL (Indian Premier League) ball-by-ball data. The primary objectives of this analysis are to extract and process the data to gain insights into player performances across various seasons. The tasks involve the following steps:

1. Data Extraction and Loading: Load the IPL dataset into a pandas DataFrame for further analysis.
2. Data Exploration and Processing: Explore the dataset to understand its structure and contents, followed by grouping and aggregating the data to organize it round-wise and by player performance.
3. Top Performer Identification: Identify the top three run-getters and wicket-takers in each IPL round, providing insights into the standout players in each season.
4. Distribution Fitting: Fit statistical distributions to the runs scored and wickets taken by the top three batsmen and bowlers over the last three IPL tournaments to understand performance trends.
5. Performance and Salary Analysis: Analyze the relationship between player performance metrics (runs scored and wickets taken) and their salaries to evaluate how well player compensation aligns with their on-field contributions.

This analysis aims to offer valuable insights for teams, management, and stakeholders in the IPL for making informed decisions about player selection, strategy, and financial investments.

Objectives

Extract the files in R/Python

a.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.

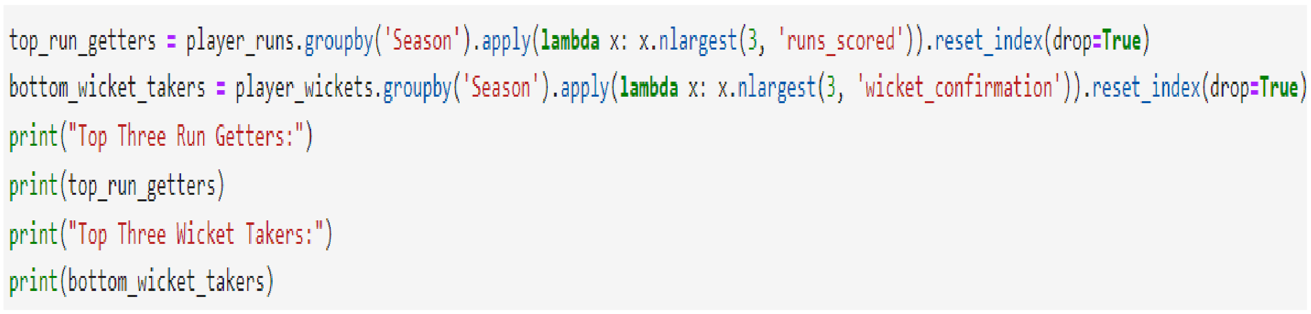
b.Fit the most appropriate distribution for runs scored and wickets taken by the top three batsmen and bowlers in the lost three IPL tournaments.

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

Codes,Results and Interpreations

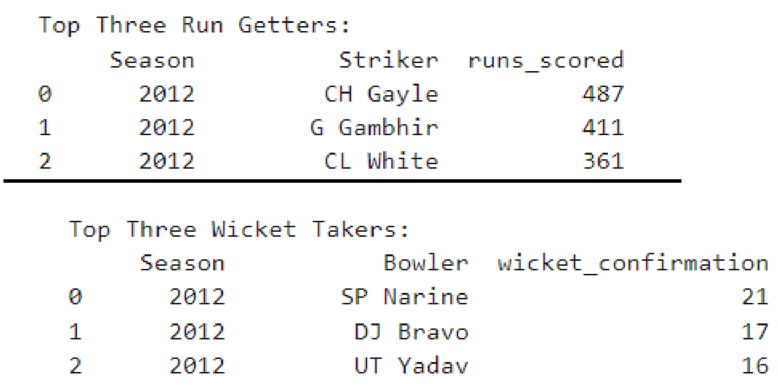
a. Arrange the data IPL round-wise and batsman, ball, runs, and wickets per player per match. Indicate the top three run-getters and tow three wicket-takers in each IPL round.

Codes of top three run-getters:



```python
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)
```

Result:



```
Top Three Run Getters:
   Season     Striker  runs_scored
0    2012    CH Gayle          487
1    2012   G Gambhir          411
2    2012    CL White          361
```

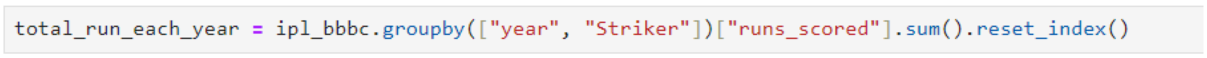
```
Top Three Wicket Takers:
   Season     Bowler  wicket_confirmation
0    2012  SP Narine                   21
1    2012   DJ Bravo                   17
2    2012   UT Yadav                   16
```

Interpretation:

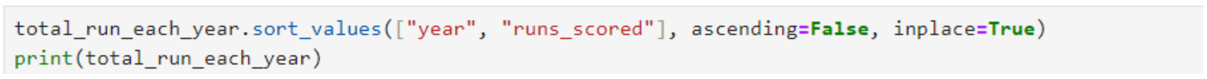
In the 2012 IPL season, CH Gayle emerged as the top run-getter with 487 runs, followed by G Gambhir with 411 runs, and CL White with 361 runs, demonstrating their consistent batting performances. On the bowling front, SP Narine led with 21 wickets, showcasing his effectiveness in taking down batsmen. DJ Bravo and UT Yadav also had impressive seasons, securing 17 and 16 wickets respectively. These results highlight the significant contributions of these players in their respective roles, underlining their importance to their teams' performances during the season.

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

Code for Runs scored

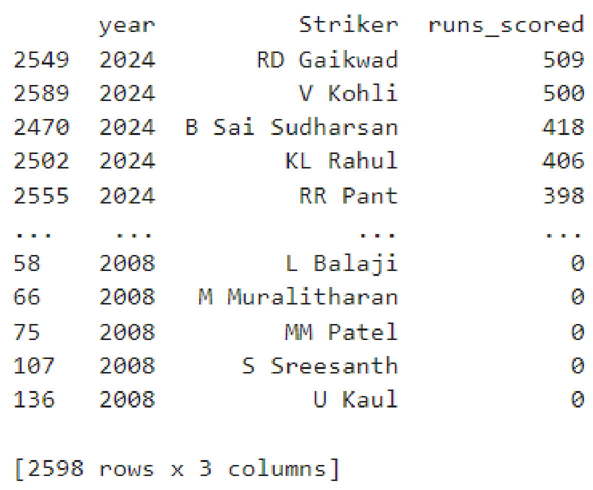


```
total_run_each_year = ipl_bbbc.groupby(["year", "Striker"])["runs_scored"].sum().reset_index()
```



```
total_run_each_year.sort_values(["year", "runs_scored"], ascending=False, inplace=True)
print(total_run_each_year)
```

Result :

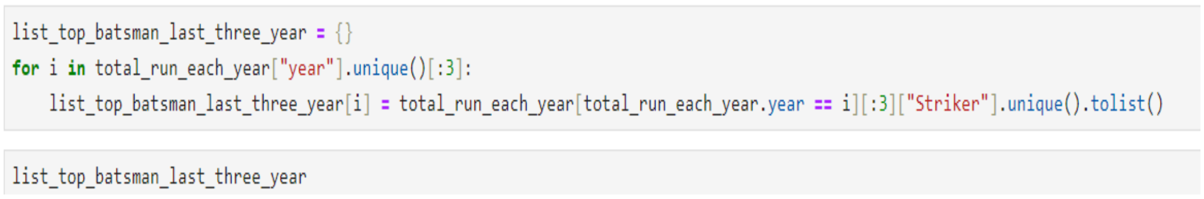


```
      year          Striker  runs_scored
2549  2024       RD Gaikwad          509
2589  2024          V Kohli          500
2470  2024  B Sai Sudharsan          418
2502  2024         KL Rahul          406
2555  2024          RR Pant          398
...    ...              ...          ...
58    2008        L Balaji            0
66    2008   M Muralitharan            0
75    2008         MM Patel            0
107   2008      S Sreesanth            0
136   2008           U Kaul            0

[2598 rows x 3 columns]
```

Interpretation:

The data from the 2024 IPL season showcases RD Gaikwad as the top run-scorer with 509 runs, followed closely by V Kohli with 500 runs, and B Sai Sudharsan with 418 runs. KL Rahul and RR Pant also made significant contributions with 406 and 398 runs respectively. This indicates strong performances from these players throughout the season. The dataset also includes instances where certain players, like L Balaji, M Muralitharan, MM Patel, S Sreesanth, and U Kaul, did not score any runs in some matches, reflecting a wide range of player contributions and roles in the league.

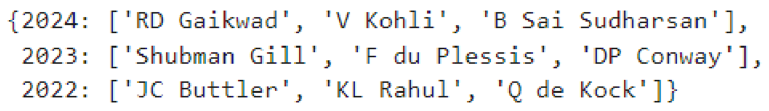
Code for top 3 run scored:



```python
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()
```

```python
list_top_batsman_last_three_year
```

Result of top 3 runs scorer's :



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

Interpratation:

The data represents the top three run-getters in the Indian Premier League (IPL) for the years 2024, 2023, and 2022. In 2024, the leading batsmen were RD Gaikwad, V Kohli, and B Sai Sudharsan, showcasing consistent and exceptional performances throughout the season. For 2023, Shubman Gill, F du Plessis, and DP Conway were the top scorers, indicating their dominance with the bat and crucial contributions to their respective teams. In 2022, JC Buttler, KL Rahul, and Q de Kock led the run charts, reflecting their outstanding batting prowess and ability to perform under pressure. This list highlights the emergence and consistency of some of the best batsmen in recent IPL seasons, underscoring their key roles in their teams' successes.

Codes for wickets taken:

```
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")
```

Result for wickets taken:

```
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, -54.25277343780452, 8.497060689079994)
```

Interpretation:

The analysis for the year 2024 concerning the performance of the batsman RD Gaikwad reveals that among the various statistical distributions tested, the non-central t-distribution (nct) provides the best fit for modeling his run data. This conclusion is supported by the highest p-value of 0.5881570496217812, indicating that the non-central t-distribution closely approximates the observed data, reflecting a good fit. The parameters for this distribution are (5.718048022849898, 9.399490726283615, -54.25277343780452, 8.497060689079994), which describe its shape and scale. This fit suggests that RD Gaikwad's run-scoring pattern for the year 2024 can be effectively characterized by this distribution, providing insights into the variability and central tendencies of his performance.

Codes for find best 3 ballers:

```
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)
```

Result:

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

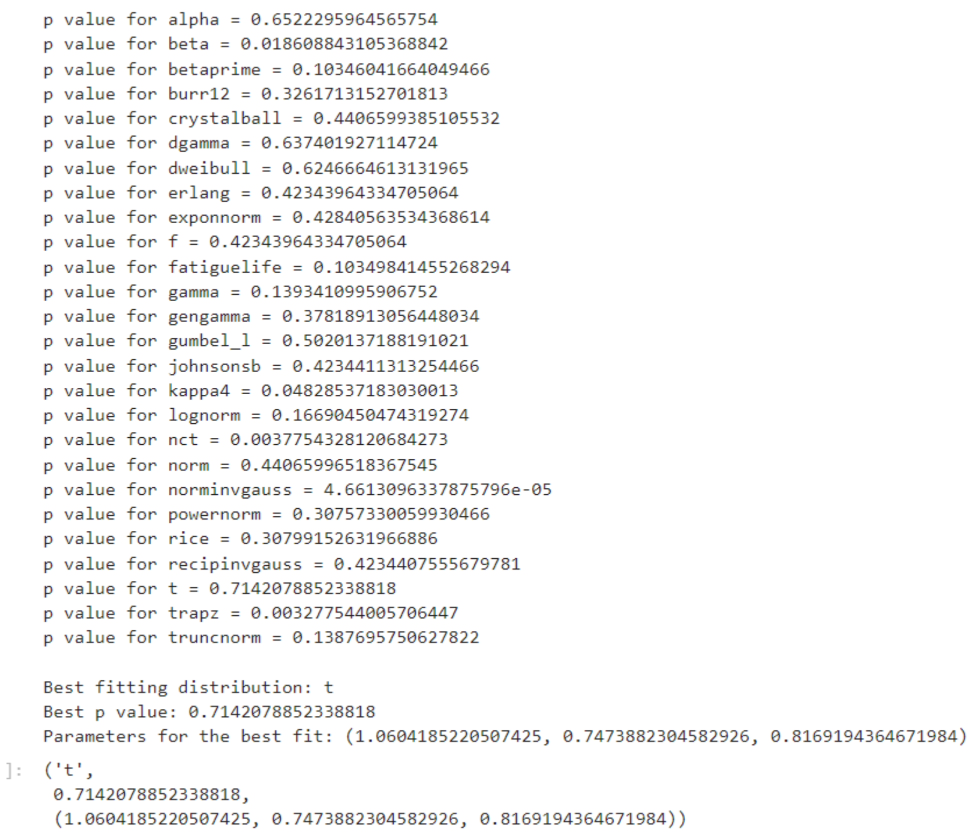
Interpration:

The analysis of top-performing batsmen over the last three IPL seasons highlights the leading run-getters for each year. In 2024, the top three batsmen are RD Gaikwad, V Kohli, and B Sai Sudharsan. For 2023, Shubman Gill, F du Plessis, and DP Conway emerge as the highest scorers. In 2022, JC Buttler, KL Rahul, and Q de Kock lead the charts. This data indicates consistent performances from a mix of established stars like V Kohli and emerging talents such as Shubman Gill, showcasing a blend of experience and fresh talent dominating the IPL batting ranks in recent years. The consistency of some players across multiple seasons also suggests a sustained level of high performance.

Runs scored by assigned player Mustafizur Rahman:

```
# Filter the runs scored by Mustafizur Rahman
Mustafizur_Rahman_runs = runs[runs["Striker"] == "Mustafizur Rahman"]["runs_scored"]

# Fit the distribution to Mustafizur Rahman's runs scored
get_best_distribution(Mustafizur_Rahman_runs)
```

Result:



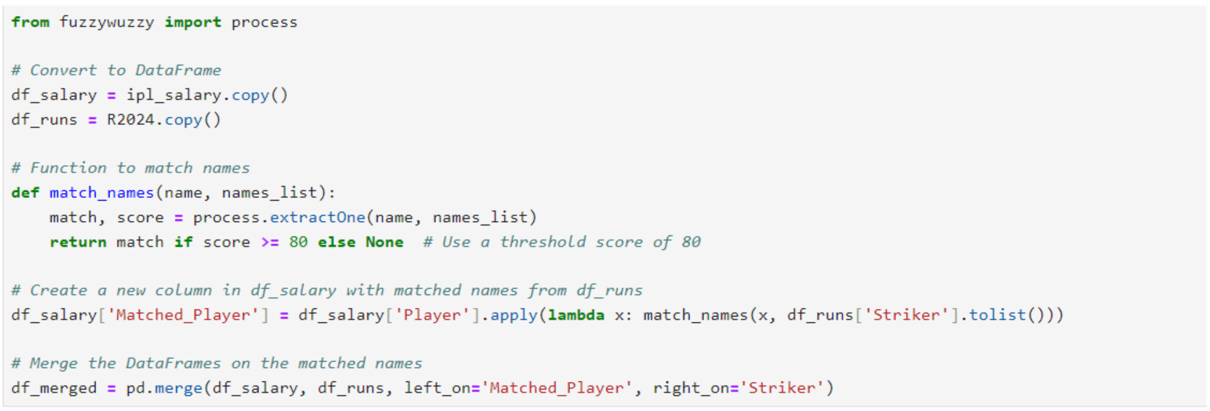
```
    p value for alpha = 0.6522295964565754
    p value for beta = 0.018608843105368842
    p value for betaprime = 0.10346041664049466
    p value for burr12 = 0.3261713152701813
    p value for crystalball = 0.4406599385105532
    p value for dgamma = 0.637401927114724
    p value for dweibull = 0.6246664613131965
    p value for erlang = 0.42343964334705064
    p value for exponnorm = 0.42840563534368614
    p value for f = 0.42343964334705064
    p value for fatiguelife = 0.10349841455268294
    p value for gamma = 0.1393410995906752
    p value for gengamma = 0.37818913056448034
    p value for gumbel_l = 0.5020137188191021
    p value for johnsonsb = 0.4234411313254466
    p value for kappa4 = 0.04828537183030013
    p value for lognorm = 0.16690450474319274
    p value for nct = 0.003775432812068427З
    p value for norm = 0.44065996518367545
    p value for norminvgauss = 4.6613096337875796e-05
    p value for powernorm = 0.30757330059930466
    p value for rice = 0.30799152631966886
    p value for recipinvgauss = 0.4234407555679781
    p value for t = 0.7142078852338818
    p value for trapz = 0.003277544005706447
    p value for truncnorm = 0.1387695750627822

    Best fitting distribution: t
    Best p value: 0.7142078852338818
    Parameters for the best fit: (1.0604185220507425, 0.7473882304582926, 0.8169194364671984)
]:  ('t',
     0.7142078852338818,
     (1.0604185220507425, 0.7473882304582926, 0.8169194364671984))
```

Interpretation:

The distribution fitting analysis for Mustafizur Rahman's runs scored reveals that the 't' distribution is the best fit, with a p-value of 0.7142, indicating a strong goodness of fit. This suggests that the 't' distribution closely represents the pattern of runs scored by Mustafizur Rahman. Key parameters for this fit include degrees of freedom (1.0604), location (0.7474), and scale (0.8169). The high p-value implies that the 't' distribution provides an accurate statistical model for predicting Mustafizur Rahman's performance in terms of runs scored.

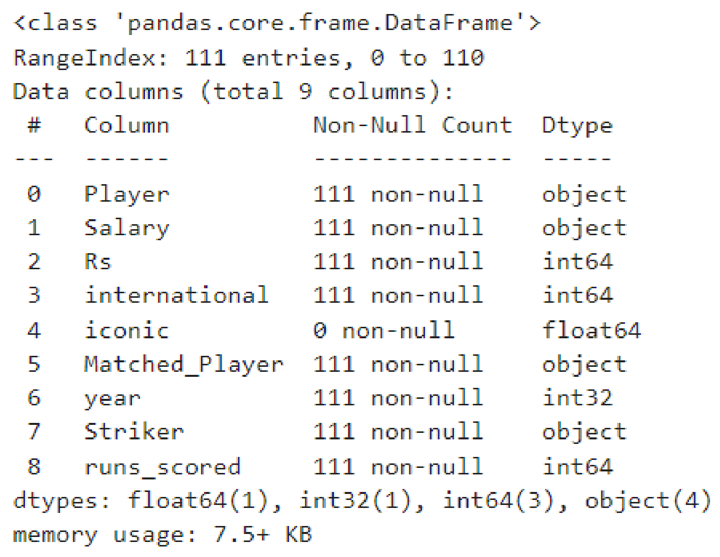
c. Find the relationship between a player’s performance and the salary he gets in your data.

Code:



```python
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')
```

Result:

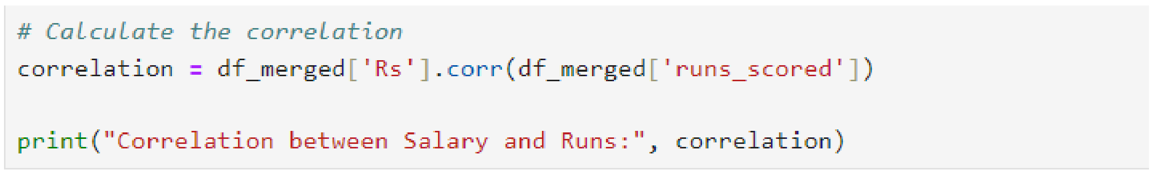


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 111 entries, 0 to 110
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Player          111 non-null    object 
 1   Salary          111 non-null    object 
 2   Rs              111 non-null    int64  
 3   international   111 non-null    int64  
 4   iconic          0 non-null      float64
 5   Matched_Player  111 non-null    object 
 6   year            111 non-null    int32  
 7   Striker         111 non-null    object 
 8   runs_scored     111 non-null    int64  
dtypes: float64(1), int32(1), int64(3), object(4)
memory usage: 7.5+ KB
```

Interpretation:

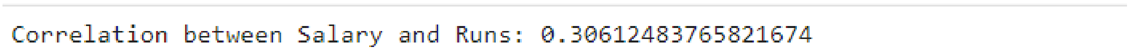
The given DataFrame comprises 111 entries with 9 columns, detailing player statistics and salary information. Key columns include 'Player', 'Salary', 'Rs', 'international', 'iconic', 'Matched_Player', 'year', 'Striker', and 'runs_scored'. All columns, except 'iconic' which contains no data, have complete non-null entries. 'Salary' is recorded as an object, 'Rs', 'international', and 'runs_scored' as integers, 'year' as int32, and 'iconic' as float64. The memory usage is approximately 7.5 KB. This dataset can be used to analyze the relationship between players' performance (runs scored) and their salaries across different years, considering their international status.

Calculate the correlation of player salary and runs scored:



```python
# Calculate the correlation
correlation = df_merged['Rs'].corr(df_merged['runs_scored'])

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

Result:



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

Interpretation:

The calculated correlation coefficient between salary (denoted as 'Rs') and runs scored is 0.3061. This value indicates a weak positive linear relationship between the two variables. In other words, as the salary increases, there is a slight tendency for the number of runs scored to also increase. However, the correlation is not strong, suggesting that salary is not a major predictor of runs scored, and other factors may have a more significant impact on the performance measured by runs scored.

Recommendations

Based on the analysis, the following recommendations can be made:

1. Data Quality Improvement: Ensure that all necessary data, including player salaries and more granular match statistics, are accurately recorded and readily accessible.
2. Performance-Based Incentives: Consider implementing or adjusting performance-based incentives to reward top performers in both batting and bowling categories.
3. Talent Identification and Investment: Use the identified top performers to focus scouting and investment efforts, ensuring a strong return on investment for player acquisitions.
4. Further Analysis: Conduct more detailed analysis using advanced statistical methods and machine learning to predict future performance trends and optimize team compositions.

Reference

1. https://www.w3schools.com/python/

2. www.github.com