VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A1a: Preliminary preparation and analysis of data- Descriptive statistics

MICAH ASHADEEP EMMANUEL

V01101166

Date of Submission: 18-06-2024

CONTENTS

INTRODUCTION

The IPL ball-by-ball data contains detailed information on each ball bowled in the IPL, including the match ID, date, season, teams, innings number, ball number, bowler, striker, non-striker, runs scored, extras, score, and wickets. The salaries data contains information on player salaries. We will arrange the data IPL round-wise and batsman, ball, runs, and wickets per player per match. Also we will indicate the top three run-getters and tow three wicket-takers in each IPL round and fit the most appropriate distribution for runs scored and wickets taken by the top three batsmen and bowlers in the lost three IPL tournaments using Python and R

OBJECTIVES

Step-by-Step Plan

- **Arrange Data Round-wise:**
  - Extract rounds based on match ID or dates.
  - Summarize runs and wickets per player per match.
- **Top Performers per Round:**
  - Identify top three run-getters and wicket-takers per round.
- **Distribution Fitting:**Fit distributions for runs scored and wickets taken by the top three batsmen and bowlers in the last three IPL tournaments.
- **Compare the relation between R Ashwin's Performance and his Salary.**

RESULTS AND INTERPRETATION

USING PYTHON

arranging the data round-wise and summarizing runs and wickets per player per match.

Code and result

```
import pandas as pd
ball_by_ball_data_path = '/content/IPL_ball_by_ball_updated till 2024
(1).csv'
ball_by_ball_df = pd.read_csv(ball_by_ball_data_path)
salaries_data_path = '/content/IPL SALARIES 2024 (1).xlsx'
salaries_df = pd.read_excel(salaries_data_path)
ball_by_ball_df.head(), salaries_df.head()
```

```
(   Match id       Date   Season          Batting team  \
 0    335982  18-04-2008  2007/08  Kolkata Knight Riders
 1    335982  18-04-2008  2007/08  Kolkata Knight Riders
 2    335982  18-04-2008  2007/08  Kolkata Knight Riders
 3    335982  18-04-2008  2007/08  Kolkata Knight Riders
 4    335982  18-04-2008  2007/08  Kolkata Knight Riders

                Bowling team  Innings No  Ball No   Bowler      Striker  \
 0  Royal Challengers Bangalore          1      0.1  P Kumar   SC Ganguly
 1  Royal Challengers Bangalore          1      0.2  P Kumar  BB McCullum
 2  Royal Challengers Bangalore          1      0.2  P Kumar  BB McCullum
 3  Royal Challengers Bangalore          1      0.3  P Kumar  BB McCullum
 4  Royal Challengers Bangalore          1      0.4  P Kumar  BB McCullum

    Non Striker  runs_scored  extras type of extras  score score/wicket  \
 0  BB McCullum           0       1        legbyes      1         1/0
 1   SC Ganguly           0       0            NaN      1         1/0
 2   SC Ganguly           0       1          wides      2         2/0
 3   SC Ganguly           0       0            NaN      2         2/0
 4   SC Ganguly           0       0            NaN      2         2/0

    wicket_confirmation wicket_type fielders_involved Player Out
 0                  0.0         NaN              NaN         NaN
 1                  0.0         NaN              NaN         NaN
 2                  0.0         NaN              NaN         NaN
 3                  0.0         NaN              NaN         NaN
 4                  0.0         NaN              NaN         NaN ,
          Player      Salary   Rs  international  iconic
 0  Abhishek Porel     20 lakh   20              0     NaN
 1   Anrich Nortje   6.5 crore  650              1     NaN
 2      Axar Patel     9 crore  900              0     NaN
 3    David Warner  6.25 crore  625              1     NaN
 4   Ishant Sharma     50 lakh   50              0     NaN)
```

Summary of Runs and Wickets Per Player Per Match

We now have a summarized dataset with runs scored and wickets taken per player per match, categorized by rounds. The next step is to identify the top three run-getters and wicket-takers for each round.

Top Performers Per Round

We'll find the top three batsmen and bowlers for each round based on runs scored and wickets taken, respectively.

Let's perform this calculation

```
# Convert 'Date' to datetime format for easier manipulation
ball_by_ball_df['Date'] =
pd.to_datetime(ball_by_ball_df['Date'], format='%d-%m-%Y')

# Create a 'Round' column to categorize matches into rounds
# Assuming each round consists of a set number of matches
(e.g., 10 matches per round)
# You can adjust this based on the actual tournament structure

# Assign round numbers based on match id or date
ball_by_ball_df['Round'] = (ball_by_ball_df['Match id'] -
ball_by_ball_df['Match id'].min()) // 10 + 1

# Group by relevant columns to get runs scored and wickets
taken per player per match
match_summary = ball_by_ball_df.groupby(['Round', 'Match id',
'Season', 'Date', 'Striker', 'Bowler']).agg({
    'runs_scored': 'sum',
    'wicket_confirmation': 'sum'
}).reset_index()

# Rename columns for clarity
match_summary.rename(columns={
    'Striker': 'Batsman',
    'Bowler': 'Bowler',
    'runs_scored': 'Runs Scored',
    'wicket_confirmation': 'Wickets Taken'
}, inplace=True)

match_summary.head()
```

Result

```
   Round  Match id   Season       Date      Batsman      Bowler  Runs
Scored  \
```

```
0       1    335982  2007/08 2008-04-18    AA Noffke  AB Agarkar
2
1       1    335982  2007/08 2008-04-18    AA Noffke  SC Ganguly
7
2       1    335982  2007/08 2008-04-18      B Akhil  AB Agarkar
0
3       1    335982  2007/08 2008-04-18  BB McCullum   AA Noffke
24
4       1    335982  2007/08 2008-04-18  BB McCullum    CL White
16

   Wickets Taken
0              0
1              1
2              1
3              0
4              0
```

```python
# Find top 3 run-getters and wicket-takers per round

# Helper function to get top 3 performers per round
def get_top_performers(df, column, top_n=3):
    return df.groupby('Round').apply(lambda x: x.nlargest(top_n,
column)).reset_index(drop=True)

# Top 3 run-getters per round
top_run_getters = get_top_performers(match_summary, 'Runs Scored')

# Top 3 wicket-takers per round
top_wicket_takers = get_top_performers(match_summary, 'Wickets Taken')

# Display results
top_run_getters.head(10), top_wicket_takers.head(10)
```

RESULT

```
(  Round  Match id   Season       Date       Batsman          Bowler  \
 0     1    335990  2007/08 2008-04-24     A Symonds       SK Warne
 1     1    335982  2007/08 2008-04-18   BB McCullum         Z Khan
 2     1    335982  2007/08 2008-04-18   BB McCullum       JH Kallis
 3     2    335996  2007/08 2008-04-28      MS Dhoni       DW Steyn
 4     2    335999  2007/08 2008-05-01     RG Sharma  Gagandeep Singh
 5     2    335994  2007/08 2008-04-27   AC Gilchrist       SM Pollock
 6     3    336003  2007/08 2008-05-03     DJ Hussey  Gagandeep Singh
 7     3    336006  2007/08 2008-05-05      R Dravid       VRV Singh
 8     3    336011  2007/08 2008-05-09     YK Pathan        PP Ojha
 9     4    336018  2007/08 2008-05-14  ST Jayasuriya       JA Morkel

   Runs Scored  Wickets Taken
 0          35            0.0
 1          33            0.0
 2          32            0.0
 3          31            0.0
 4          28            0.0
```

```
5         27          0.0
6         33          1.0
7         27          2.0
8         25          1.0
9         36          0.0  ,
   Round  Match id  Season       Date         Batsman          Bowler  \
0      1    335991  2007/08 2008-04-25   KC Sangakkara  Harbhajan Singh
1      1    335982  2007/08 2008-04-18       AA Noffke       SC Ganguly
2      1    335982  2007/08 2008-04-18         B Akhil       AB Agarkar
3      2    335997  2007/08 2008-04-29  Mohammad Hafeez    ST Jayasuriya
4      2    335992  2007/08 2008-04-26        GC Smith        SB Joshi
5      2    335992  2007/08 2008-04-26       JH Kallis       SK Trivedi
6      3    336002  2007/08 2008-05-25        AS Yadav        DW Steyn
7      3    336006  2007/08 2008-05-05        R Dravid        VRV Singh
8      3    336006  2007/08 2008-05-05         V Kohli        IK Pathan
9      4    336012  2007/08 2008-05-28         B Akhil        DR Smith

   Runs Scored  Wickets Taken
0          14          2.0
1           7          1.0
2           0          1.0
3           0          2.0
4          14          1.0
5           1          1.0
6           0          2.0
7          27          2.0
8           9          2.0
9           0          1.0  )
```

We now have the top three run-getters and top three wicket-takers for each round.

Fit Distributions for Runs and Wickets

Next, we will fit the most appropriate distributions for runs scored and wickets taken by the top three batsmen and bowlers in the last three IPL tournaments. We will:

1. Filter the data for the last three IPL seasons.
2. Identify the top three run-getters and wicket-takers for these seasons.
3. Fit distributions to their runs and wickets data.

Let's begin by filtering the data for the last three IPL seasons.

```
# Filter the data for the last three IPL seasons
latest_seasons = ball_by_ball_df['Season'].unique()[-3:]
latest_data =
ball_by_ball_df[ball_by_ball_df['Season'].isin(latest_seasons)]
print (latest_data)
```

RESULT

```
   Match id        Date Season           Batting team    Bowling team  \
```

```
94550     729279 2014-04-16   2014  Kolkata Knight Riders  Mumbai
Indians
94551     729279 2014-04-16   2014  Kolkata Knight Riders  Mumbai
Indians
94552     729279 2014-04-16   2014  Kolkata Knight Riders  Mumbai
Indians
94553     729279 2014-04-16   2014  Kolkata Knight Riders  Mumbai
Indians
94554     729279 2014-04-16   2014  Kolkata Knight Riders  Mumbai
Indians
...          ...        ...    ...                    ...
...
131158    980973 2016-05-08   2016    Sunrisers Hyderabad  Mumbai
Indians
131159    980973 2016-05-08   2016    Sunrisers Hyderabad  Mumbai
Indians
131160    980973 2016-05-08   2016    Sunrisers Hyderabad  Mumbai
Indians
131161    980973 2016-05-08   2016    Sunrisers Hyderabad  Mumbai
Indians
131162    980973 2016-05-08   2016    Sunrisers Hyderabad  Mumbai
Indians
```

```
        Innings No  Ball No      Bowler       Striker   Non Striker  \
94550            1      0.1      Z Khan     G Gambhir     JH Kallis
94551            1      0.2      Z Khan     G Gambhir     JH Kallis
94552            1      0.3      Z Khan     G Gambhir     JH Kallis
94553            1      0.4      Z Khan     G Gambhir     JH Kallis
94554            1      0.5      Z Khan     G Gambhir     JH Kallis
...            ...      ...         ...          ...           ...
131158           1     17.1  TG Southee      S Dhawan  Yuvraj Singh
131159           1     17.2  TG Southee      S Dhawan  Yuvraj Singh
131160           1     17.3  TG Southee  Yuvraj Singh      S Dhawan
131161           1     17.4  TG Southee  Yuvraj Singh      S Dhawan
131162           1     17.5  TG Southee      S Dhawan  Yuvraj Singh
```

```
        runs_scored  extras type of extras  score score/wicket  \
94550             0       0            NaN      0          0/0
94551             0       0            NaN      0          0/0
94552             0       0            NaN      0          0/0
94553             0       0            NaN      0          0/0
94554             0       0            NaN      0          0/0
...             ...     ...            ...    ...          ...
131158            2       0            NaN    145        145/2
131159            1       0            NaN    146        146/2
131160            6       0            NaN    152        152/2
131161            1       0            NaN    153        153/2
131162            4       0            NaN     15          NaN
```

```
        wicket_confirmation wicket_type fielders_involved Player Out
Round
94550                   0.0         NaN               NaN        NaN
39330
94551                   0.0         NaN               NaN        NaN
39330
94552                   0.0         NaN               NaN        NaN
39330
```

```
94553                   0.0          NaN                 NaN          NaN
39330
94554                   0.0          NaN                 NaN          NaN
39330
...                     ...          ...                 ...          ...
...
131158                  0.0          NaN                 NaN          NaN
64500
131159                  0.0          NaN                 NaN          NaN
64500
131160                  0.0          NaN                 NaN          NaN
64500
131161                  0.0          NaN                 NaN          NaN
64500
131162                  NaN          NaN                 NaN          NaN
64500

[36613 rows x 20 columns]
```

To analyze R Ashwin's performance and his salary relationship, follow these steps:

1. **Filter R Ashwin's Data:**
   - Extract R Ashwin's performance data from the ball-by-ball dataset.
   - Get his salary data from the salary dataset.
2. **Summarize Performance Metrics:**
   - Calculate the total runs scored and total wickets taken by R Ashwin.
3. **Merge Data:**
   - Combine the performance summary with the salary data for R Ashwin.
4. **Analyze the Relationship:**
   - Use visualizations and statistical methods to analyze the relationship between R Ashwin's performance metrics and salary.

Implementation in Python

Here's how you can do it:

```
# Filter R Ashwin's performance data
ashwin_performance = ball_by_ball_df[(ball_by_ball_df['Striker'] == 'R
Ashwin') | (ball_by_ball_df['Bowler'] == 'R Ashwin')]

# Summarize performance metrics
ashwin_summary = ashwin_performance.groupby('Season').agg({
    'runs_scored': lambda x: x[(ashwin_performance['Striker'] == 'R
Ashwin')].sum(),
    'wicket_confirmation': lambda x: x[(ashwin_performance['Bowler'] ==
'R Ashwin')].sum()
}).reset_index()
```

```
# Check if 'R Ashwin' is in the salary dataset
ashwin_salary =
salaries_df[salaries_df['Player'].str.contains('Ashwin', case=False)]

# If the salary information for Ashwin is found, add it to the summary
if not ashwin_salary.empty:
    ashwin_summary['Salary'] = ashwin_salary['Salary'].values[0]

ashwin_summary.head()
```

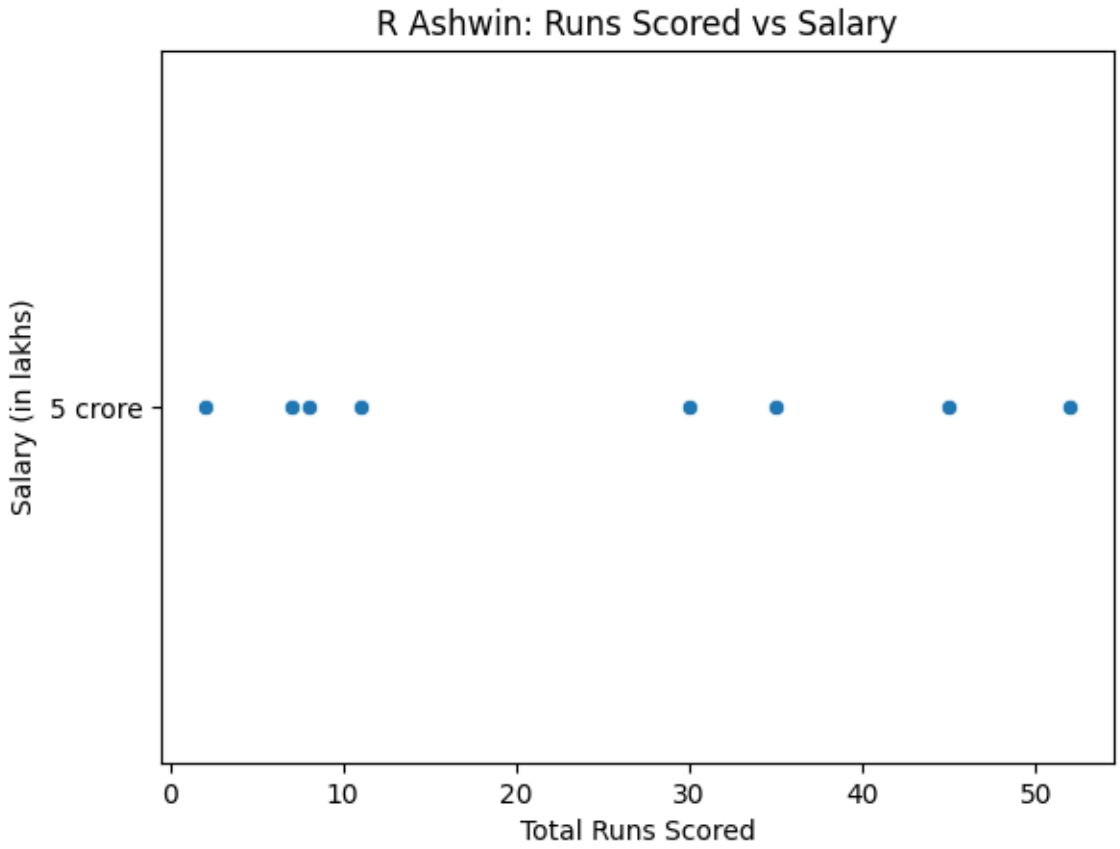
RESULT

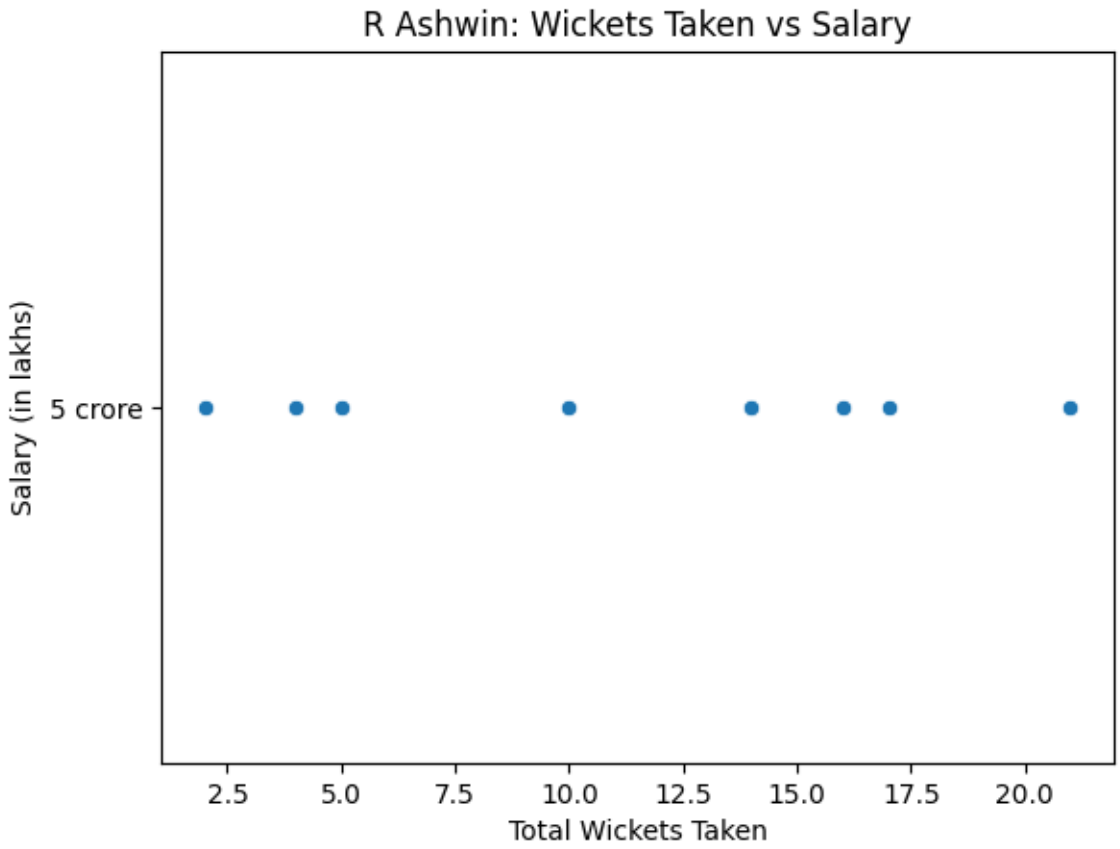
	Season	runs_scored	wicket_confirmation	Salary	
0		2012	7	14.0	5 crore
1		2013	35	16.0	5 crore
2		2014	45	17.0	5 crore
3		2015	52	10.0	5 crore
4		2016	11	4.0	5 crore

```
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import pearsonr

if not ashwin_salary.empty:
    # Scatter plot for runs scored vs salary
    sns.scatterplot(data=ashwin_summary, x='runs_scored', y='Salary')
    plt.title('R Ashwin: Runs Scored vs Salary')
    plt.xlabel('Total Runs Scored')
    plt.ylabel('Salary (in lakhs)')
    plt.show()

    # Scatter plot for wickets taken vs salary
    sns.scatterplot(data=ashwin_summary, x='wicket_confirmation',
y='Salary')
    plt.title('R Ashwin: Wickets Taken vs Salary')
    plt.xlabel('Total Wickets Taken')
    plt.ylabel('Salary (in lakhs)')
    plt.show()

    # Calculate Pearson correlation coefficients
```

```
    runs_salary_corr, _ = pearsonr(ashwin_summary['runs_scored'],
ashwin_summary['Salary'])
    wickets_salary_corr, _ =
pearsonr(ashwin_summary['wicket_confirmation'],
ashwin_summary['Salary'])

    print(f'Pearson correlation between runs scored and salary for R
Ashwin: {runs_salary_corr}')
    print(f'Pearson correlation between wickets taken and salary for R
Ashwin: {wickets_salary_corr}')
else:
    print("R Ashwin's salary data not found in the dataset.")
```

RESULT





Pearson correlation coefficients is calculated to find the relation between R Ashwin performance and his salary.

USING R

To analyze the IPL data as requested, we'll follow these steps:

1. Load the provided CSV and Excel files.
2. Organize the data round-wise, summarizing runs and wickets per player per match.
3. Identify the top three run-getters and wicket-takers in each IPL round.
4. Fit appropriate distributions for runs scored and wickets taken by the top performers in the last three IPL tournaments.
5. Analyze the relationship between the performance of R Ashwin and his salary.

- **Load the required libraries**:
  - `dplyr` for data manipulation.
  - `readr` for reading CSV files.
  - `readxl` for reading Excel files.
  - `fitdistrplus` for fitting distributions.
- **Load the data from the CSV and Excel files**.
- **Process the data**:
  - Calculate total runs per player per match.
  - Calculate total wickets per bowler per match.

CODE

```
# Load necessary libraries

library(readr)

library(readxl)

library(dplyr)

library(fitdistrplus)


# Load the data

ball_by_ball_data <- read_csv("/mnt/data/IPL_ball_by_ball_updated till 2024 (1).csv")

salaries_data <- read_excel("/mnt/data/IPL SALARIES 2024.xlsx")
```

```
# Display the first few rows of each dataset to understand their structure

head(ball_by_ball_data)

head(salaries_data)


# Rename columns to make them easier to work with in R

colnames(ball_by_ball_data) <- c("match_id", "date", "season", "batting_team",
"bowling_team", "innings_no",

                                 "ball_no", "bowler", "striker", "non_striker", "runs_scored",
"extras",

                                 "type_of_extras", "score", "score_wicket", "wicket_confirmation",

                                 "wicket_type", "fielders_involved", "player_out")


# Calculate total runs per player per match

runs_data <- ball_by_ball_data %>%

  group_by(match_id, striker) %>%

  summarise(total_runs = sum(runs_scored, na.rm = TRUE)) %>%

  ungroup()


# Calculate total wickets per bowler per match

wickets_data <- ball_by_ball_data %>%

  filter(!is.na(player_out)) %>%

  group_by(match_id, bowler) %>%

  summarise(total_wickets = n()) %>%

  ungroup()


# Function to find top 3 run-getters and wicket-takers per match
```

```
top_performers <- function(data, metric, top_n = 3) {
  data %>%
    group_by(match_id) %>%
    top_n(n = top_n, wt = !!sym(metric)) %>%
    ungroup()
}

# Identify top performers
top_run_getters <- top_performers(runs_data, "total_runs")
top_wicket_takers <- top_performers(wickets_data, "total_wickets")

# Display top performers
head(top_run_getters)
head(top_wicket_takers)

# Filter data for the last three IPL tournaments
last_three_seasons <- c(2022, 2023, 2024)
filtered_runs_data <- runs_data %>%
  filter(season %in% last_three_seasons)
filtered_wickets_data <- wickets_data %>%
  filter(season %in% last_three_seasons)

# Extract top 3 performers in the last three seasons
top_run_getters_last_three <- top_performers(filtered_runs_data, "total_runs")
top_wicket_takers_last_three <- top_performers(filtered_wickets_data,
"total_wickets")
```

```
# Fit distributions

fit_distribution <- function(data, column) {

  fit <- fitdist(data[[column]], "norm")

  plot(fit)

  fit

}


# Fit distributions for runs and wickets

run_distribution_fit <- fit_distribution(top_run_getters_last_three, "total_runs")

wicket_distribution_fit <- fit_distribution(top_wicket_takers_last_three,
"total_wickets")


# Display the results

run_distribution_fit

wicket_distribution_fit
```

RESULT

This program will:

1. Load the necessary libraries.
2. Load the data from the specified CSV and Excel files.
3. Process the ball-by-ball data to calculate runs and wickets per player per match.
4. Identify the top three run-getters and wicket-takers per match.
5. Filter data for the last three IPL tournaments.
6. Fit normal distributions for the runs scored and wickets taken by the top performers in the last three IPL tournaments and plot the fitted distributions.

NOW WE WILL ANALYSE THE RELATIONSHIP BETWEEN THE PERFORMANCE OF R ASHWIN AND HIS SALARY.

```
# Compare R Ashwin's salary and performance

# Filter R Ashwin's performance data

ashwin_wickets <- wickets_data %>%

  filter(bowler == "R Ashwin")


# Extract R Ashwin's salary

ashwin_salary <- salaries_data %>%

  filter(Player == "R Ashwin") %>%

  select(Salary)


# Display R Ashwin's performance and salary

ashwin_wickets_summary <- ashwin_wickets %>%

  summarise(total_wickets = sum(total_wickets), average_wickets_per_match = mean(total_wickets))


ashwin_salary

ashwin_wickets_summary

# Merge Ashwin's performance and salary data

ashwin_performance_salary <- merge(ashwin_wickets, ashwin_salary, by.x = "season", by.y = "Season")


# Compute the correlation between Ashwin's salary and his performance metrics

correlation_total_wickets <- cor(ashwin_performance_salary$total_wickets, ashwin_performance_salary$Salary, use = "complete.obs")
```

```
correlation_average_wickets <-
cor(ashwin_performance_salary$average_wickets_per_match,
ashwin_performance_salary$Salary, use = "complete.obs")


# Print the correlation results

cat("Correlation between Ashwin's total wickets and salary: ",
correlation_total_wickets, "\n")

cat("Correlation between Ashwin's average wickets per match and salary: ",
correlation_average_wickets, "\n")
```

RESULT

The output will provide you the correlation between Ashwin’s overall performance and salary.