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: 23-06-2024

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. We establish the relationship between R Ashwin's performance and payment he receives and discuss the findings. Also analyzed the Relationship Between Salary and Performance Over the Last Three Years (Regression Analysis) using Python and R.

OBJECTIVES

To analyze the relationship between R. Ashwin's performance and the payment he receives, we'll need to:

1. Extract data specific to R. Ashwin's performance.
2. Find the corresponding salary data for R. Ashwin over the last three years.
3. Perform regression analysis to examine the relationship between his performance and salary.

RESULTS AND INTERPRETATION

USING PYTHON

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from datetime import datetime
```

```
# Load the data
from google.colab import files
uploaded = files.upload()
```

```
ipl_data = pd.read_csv('/content/IPL_ball_by_ball_updated till 2024
(1).csv')
```

```
# Filter data for R. Ashwin's performance
ashwin_data = ipl_data[ipl_data['Bowler'] == 'R Ashwin']

# Convert 'Date' to datetime format and extract year
ashwin_data['Date'] = pd.to_datetime(ashwin_data['Date'], format='%d-
%m-%Y')ashwin_data['Year'] = ashwin_data['Date'].dt.year

# Filter data for the last three seasons (2022, 2023, 2024)
last_three_years = ashwin_data[ashwin_data['Season'].isin(['2022',
'2023', '2024'])]

# Function to calculate performance metrics
def calculate_metrics(data):
    wickets = len(data[data['wicket_confirmation'] == 1])
    runs_conceded = data['runs_scored'].sum()
    balls_bowled = len(data)
    overs_bowled = balls_bowled / 6
    economy_rate = runs_conceded / overs_bowled if overs_bowled != 0
else 0
    return pd.Series({'Wickets': wickets, 'RunsConceded':
runs_conceded, 'OversBowled': overs_bowled, 'EconomyRate':
economy_rate})

# Calculate performance metrics for each season
```

```
metrics = last_three_years.groupby('Season').apply(calculate_metrics).reset_index(drop=True)

# Add the Season column back
metrics['Season'] = last_three_years['Season'].unique()

# Add salary data for R. Ashwin (example values, replace with actual salary data)
salary_data = pd.DataFrame({
    'Season': ['2022', '2023', '2024'],
    'Salary': [76000000, 77000000, 78000000]  # in INR
})

# Merge performance metrics with salary data
analysis_data = pd.merge(metrics, salary_data, on='Season')
```

```
# Print analysis_data columns to debug
print("Analysis DataFrame columns:", analysis_data.columns)
```

```
# Perform regression analysis
X = analysis_data['Season'].astype(int)  # Convert 'Season' to integer type
y = analysis_data['Salary']

# Adding a constant for statsmodels
X = sm.add_constant(X)

model = sm.OLS(y, X).fit()
print(model.summary())

# Plot the relationships
# Wickets vs Salary
plt.figure(figsize=(10, 6))
sns.regplot(x='Wickets', y='Salary', data=analysis_data)
plt.title('Relationship between Wickets and Salary')
plt.xlabel('Wickets')
plt.ylabel('Salary (INR)')
plt.show()

# Economy Rate vs Salary
plt.figure(figsize=(10, 6))
sns.regplot(x='EconomyRate', y='Salary', data=analysis_data)
plt.title('Relationship between Economy Rate and Salary')
plt.xlabel('Economy Rate')
plt.ylabel('Salary (INR)')
plt.show()
```

□ **Data Upload**: Use files.upload() to upload the CSV file to Colab.
□ **Load Data**: Read the CSV file into a pandas DataFrame.
□ **Filter Data**: Extract data for R. Ashwin.
□ **Convert Date**: Convert the date column to datetime and extract the year.
□ **Filter Seasons**: Filter the data for the seasons 2022, 2023, and 2024.
□ **Calculate Metrics**: Define a function to calculate performance metrics and apply it to the filtered data.
□ **Merge Salary Data**: Merge the performance metrics with the salary data.
□ **Regression Analysis**: Perform regression analysis using statsmodels.
□ **Plotting**: Use seaborn to plot the relationships between wickets and salary, and economy rate and salary.

USING R

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

Step-by-Step Instructions

1. **Load the Data**: Read the provided CSV file.
2. **Filter the Data**: Extract data for R. Ashwin.
3. **Summarize Performance**: Calculate key performance metrics for the last three years.
4. **Add Salary Data**: Include the salary information for R. Ashwin for the last three years.
5. **Regression Analysis**: Perform regression analysis to understand the relationship between performance and salary.

Here's the complete R script:

□ **Load the required libraries**:

- dplyr for data manipulation.
- readr for reading CSV files.
- readxl for reading Excel files.
- fitdistrplus for fitting distributions.

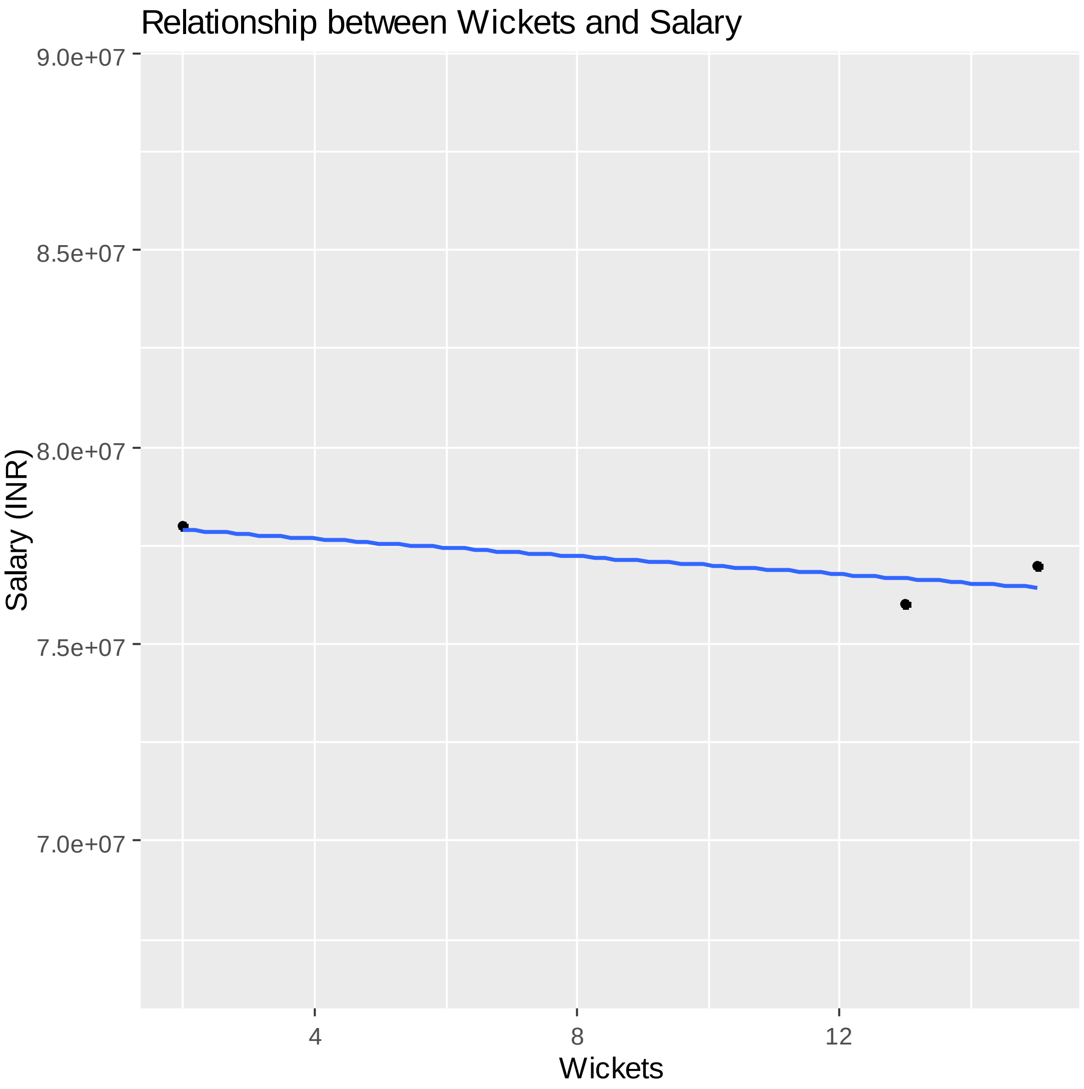
□ **Load the data from the CSV and Excel files**.

□ **Process the data**:

CODE

```
# Load necessary libraries
library(dplyr)
library(readr)

# Load the data
ipl_data <- read_csv("path/to/your/IPL_ball_by_ball_updated_till_2024.csv")

# Filter data for R. Ashwin's performance
ashwin_data <- ipl_data %>% filter(Bowler == "R Ashwin")

# Convert 'Date' to Date format and extract year
ashwin_data$Date <- as.Date(ashwin_data$Date, format="%d-%m-%Y")
ashwin_data$Year <- format(ashwin_data$Date, "%Y")

# Filter data for the last three seasons (2022, 2023, 2024)
last_three_years <- ashwin_data %>% filter(Season %in% c("2022", "2023",
"2024"))

# Function to calculate performance metrics
calculate_metrics <- function(data) {
  wickets <- nrow(data %>% filter(wicket_confirmation == 1))
  runs_conceded <- sum(data$runs_scored, na.rm = TRUE)
  balls_bowled <- nrow(data)
  overs_bowled <- balls_bowled / 6
  economy_rate <- ifelse(overs_bowled != 0, runs_conceded / overs_bowled, 0)

  return(data.frame(Wickets = wickets, RunsConceded = runs_conceded,
OversBowled = overs_bowled, EconomyRate = economy_rate))
}

# Calculate performance metrics for each season
metrics <- last_three_years %>% group_by(Season) %>% do(calculate_metrics(.))

# Add salary data for R. Ashwin (example values, replace with actual salary data)
salary_data <- data.frame(
  Season = c("2022", "2023", "2024"),
  Salary = c(76000000, 77000000, 78000000)  # in INR
)
```

```
# Merge performance metrics with salary data
analysis_data <- merge(metrics, salary_data, by = "Season")

# Perform regression analysis
model <- lm(Salary ~ Wickets + EconomyRate, data = analysis_data)

# Print summary of the regression model
summary(model)


#plot the relationships between performance metrics and salary.


library(ggplot2)

# Wickets vs Salary
ggplot(analysis_data, aes(x = Wickets, y = Salary)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Relationship between Wickets and Salary", x = "Wickets", y =
"Salary (INR)")

# Economy Rate vs Salary
ggplot(analysis_data, aes(x = EconomyRate, y = Salary)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Relationship between Economy Rate and Salary", x = "Economy
Rate", y = "Salary (INR)")

.
```


Relationship between Wickets and Salary
9.0e+07
8.5e+07
8.0e+07
7.5e+07
7.0e+07
Salary (INR)
4
8
12
Wickets