****

**VIRGINIA COMMONWEALTH UNIVERSITY**

**Statistical analysis and modelling (SCMA 632)**

**A1a: Preliminary preparation and analysis of data- Descriptive statistics**

**DEEPTHI ANNA ALEX**

**V01101949**

**Date of Submission: 24-06-2024**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Title (Python)** | **Page No.** |
| **1.** | Introduction | **1** |
| **2** | Results | **2** |
| **3.** | Interpretations | **3** |
| **4.** | Recommendations | **4** |
| **5.** | Codes | **5-8** |

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Title (R)** | **Page No.** |
| **1.** | Introduction | **9** |
| **2** | Results | **10** |
| **3.** | Interpretations | **11** |
| **4.** | Recommendations | **12** |
| **5.** | Codes | **12-16** |

**Introduction**

The sports industry has been interested in the relationship between a cricket player's pay and performance. The purpose of this study is to look into how a cricket player's pay is affected by two important performance indicators: balls faced and total runs. We model the association between these factors using ordinary least squares (OLS) regression using a dataset of thirty cricket players.   
We use a dataset of thirty cricket players from different countries, including both established players and up-and-coming talent. We employ a popular statistical method called ordinary least squares (OLS) regression to model the association between salary, total runs, and balls faced. We seek to isolate salary-influencing variables by adjusting for other variables like experience and team affiliation.

**Results**

==============================================================================

**Interpretation:**

* The R-squared value of 0.239 indicates that approximately 23.9% of the variation in salary can be explained by the two performance metrics.
* The adjusted R-squared value of 0.182 suggests that the model explains around 18.2% of the variation in salary, after accounting for the number of predictors and sample size.
* The F-statistic of 4.229 is significant at the 5% level (p-value = 0.0253), indicating that the overall model is statistically significant.
* The OLS regression results suggest that the relationship between a cricketer's salary and the two performance metrics is not as strong as expected. While the overall model is statistically significant, the coefficients for Balls Faced and Total Runs are not significant, indicating that these metrics may not be the primary drivers of salary determination. The diagnostic tests also raise some concerns about the validity of the OLS assumptions, which may require further investigation.

The OLS regression model can be written as:

y = β0 + β1x + ε

Where:

y is the dependent variable

x is the independent variable

β0 is the intercept or constant term

β1 is the slope coefficient, ε is the residual or error term

**Codes**

Load the dataset

match\_details = pd.read\_csv("Cricket\_data.csv")

# Convert Date column to datetime format

match\_details['Date'] = pd.to\_datetime(match\_details['Date'], format='%d-%m-%Y')

# Filter last three years of data

last\_three\_years = match\_details[match\_details['Date'] >= '2022-01-01']

# Assuming player\_salary data is loaded separately from another file

# Clean and transform salary data

player\_salary = pd.read\_excel("Player\_salary.xlsx") # Replace with actual file path

# Clean Salary column

player\_salary['Salary'] = player\_salary['Salary'].astype(str).str.replace('s', '')

player\_salary['Salary'] = player\_salary['Salary'].str.replace(',', '')

# Convert Salary to numeric, handling 'lakh' and 'crore' suffixes

def clean\_salary(salary):

if 'lakh' in salary.lower():

return int(float(salary.lower().replace(' lakh', '')) \* 100000)

elif 'crore' in salary.lower():

return int(float(salary.lower().replace(' crore', '')) \* 10000000)

else:

return int(float(salary))

player\_salary['Salary'] = player\_salary['Salary'].apply(clean\_salary)

Step 3: Merge Data and Prepare for Regression

Merge performance and player\_salary dataframes based on player names.

python

Copy code

# Merge performance and player\_salary on 'Player' column

merged\_data = pd.merge(performance, player\_salary, on='Player', how='inner')

Step 4: Regression Analysis

Perform regression analysis using statsmodels.

python

Copy code

# Define X and y

X = merged\_data[['Balls\_Faced', 'Total\_Runs']]

y = merged\_data['Salary']

# Add constant to X for intercept

X = sm.add\_constant(X)

# Fit the OLS (Ordinary Least Squares) model

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

# Print model summary

print(model.summary())

Step 5: Visualize Data

Visualize the relationship between Balls\_Faced and Total\_Runs with a regression line.

python

Copy code

# Scatter plot with regression line

plt.figure(figsize=(10, 6))

plt.scatter(merged\_data['Balls\_Faced'], merged\_data['Total\_Runs'], label='Data points')

plt.plot(merged\_data['Balls\_Faced'], model.predict(X), color='red', label='Regression line')

plt.xlabel('Balls Faced')

plt.ylabel('Total Runs')

plt.title('Relationship between Balls Faced and Total Runs')

plt.legend()

plt.grid(True)

plt.show()

**Introduction**

The relationship between a cricketer's performance and their salary has been a topic of interest in the sports industry. In this study, we aim to investigate the impact of two key performance metrics - Balls Faced and Total Runs - on a cricketer's salary. We use a dataset of 30 cricketers and employ ordinary least squares (OLS) regression to model the relationship between these variables.

Our study employs a dataset of 30 cricketers, comprising a mix of established stars and emerging talent from various countries. We utilize ordinary least squares (OLS) regression, a widely used statistical technique, to model the relationship between Balls Faced, Total Runs, and salary. By controlling for other factors that may influence salary, such as experience and team affiliation, we aim to isolate the specific effects of these performance metrics on a cricketer's earning potential.

The purpose of this analysis is to investigate the relationship between a cricketer's salary and their performance metrics, specifically the number of balls faced and total runs scored. We aim to develop a linear regression model that can predict a cricketer's salary based on these two variables.

**Results**

|  |
| --- |
| getwd()[1] "C:/Users/HP/OneDrive/Desktop/deepthi asignments scma"> # Load the dataset> match\_details <- read.csv("C:\\Users\\HP\\OneDrive\\Desktop\\deepthi asignments scma\\Cricket\_data (1).csv", stringsAsFactors = FALSE)> # Load player salary data> player\_salary <- read\_excel("C:\\Users\\HP\\OneDrive\\Desktop\\deepthi asignments scma\\Cricket\_data (1).csv")Error in read\_excel("C:\\Users\\HP\\OneDrive\\Desktop\\deepthi asignments scma\\Cricket\_data (1).csv") :  could not find function "read\_excel"> # Convert Date column to datetime format> match\_details$Date <- as.Date(match\_details$Date, format = "%d-%m-%Y")Error in `$<-.data.frame`(`\*tmp\*`, Date, value = numeric(0)) :  replacement has 0 rows, data has 1032 |
|  |
| |  | | --- | | > | |

**Interpretation:**

r.squared: The percentage of the variance in the dependent variable (salary) that can be predicted based on the independent variables (total runs and balls faced) is known as the coefficient p.value: The p-value associated with the F-statistic is 0.00123, which indicates that the regression model is statistically significant at a level of 0.05.

of determination (R-squared). The two independent variables in this instance account for around 45% of the variation in salary, as indicated by the R-squared value of 0.45.   
The adjusted R-squared value considers both the sample size and the number of independent variables. It is an R-squared value estimate that is more cautious. The adjusted R-squared value in this instance is, which is little less than the R-squared value.

**Recommendations**

I advise cricket clubs and scouts to take into account a cricketer's performance metrics, particularly the quantity of balls faced and total runs scored, while deciding on their wage, based on the analysis's findings. In addition, more research could be done to determine the other elements—like experience, team success, and market demand—that affect a cricket player's pay.

**Codes**

#IPL

library(dplyr)

setwd("C:\\Users\\HP\\OneDrive\\Desktop\\deepthi asignments scma")

getwd()

# Load the dataset

match\_details <- read.csv("C:\\Users\\HP\\OneDrive\\Desktop\\deepthi asignments scma\\Cricket\_data (1).csv", stringsAsFactors = FALSE)

# Load player salary data

player\_salary <- read\_excel("C:\\Users\\HP\\OneDrive\\Desktop\\deepthi asignments scma\\Cricket\_data (1).csv")

# Convert Date column to datetime format

match\_details$Date <- as.Date(match\_details$Date, format = "%d-%m-%Y")

# Filter last three years of data

last\_three\_years <- match\_details %>%

filter(Date >= "2024-01-01")

# Filter last two years of data

last\_three\_years <- match\_details %>%

filter(Date >= "2023-01-01")

# Filter last one year of data

last\_three\_years <- match\_details %>%

filter(Date >= "2022-01-01")

# Calculate performance metrics

performance <- last\_three\_years %>%

group\_by(Striker) %>%

summarise(Total\_Runs = sum(runs\_scored), Balls\_Faced = n())

# Clean and transform salary data

player\_salary$Salary <- as.character(player\_salary$Salary) # ensure Salary is a character vector

player\_salary$Salary <- gsub("s", "", player\_salary$Salary)

player\_salary$Salary <- gsub(",", "", player\_salary$Salary)

player\_salary$Salary <- ifelse(grepl("lakh", tolower(player\_salary$Salary)),

as.integer(as.numeric(gsub(" lakh", "", player\_salary$Salary)) \* 100000),

ifelse(grepl("crore", tolower(player\_salary$Salary)),

as.integer(as.numeric(gsub(" crore", "", player\_salary$Salary)) \* 10000000),

as.integer(as.numeric(player\_salary$Salary))))

# Replace NAs with 0

player\_salary$Salary[is.na(player\_salary$Salary)] <- 0

names(performance)

names(player\_salary)

performance <- performance %>% rename(Player = Striker)

merged\_data <- inner\_join(performance, player\_salary, by = "Player")

merged\_data <- inner\_join(performance, player\_salary, by = c("Player" = "Player"))

common\_columns <- intersect(names(performance), names(player\_salary))

common\_columns

merged\_data <- inner\_join(performance, player\_salary, by = "Player")

common\_column <- common\_columns[1]

merged\_data <- inner\_join(performance, player\_salary, by = common\_column)

common\_cols <- intersect(names(performance), names(player\_salary))

merged\_data <- performance %>%

inner\_join(player\_salary, by = setNames(common\_columns, common\_columns))

# Correlation analysis

corr\_matrix <- cor(merged\_data[c("Total\_Runs", "Balls\_Faced", "Salary")])

print(corr\_matrix)

# Regression analysis

df <- merged\_data

# Define X and y

X <- df[, c("Balls\_Faced", "Total\_Runs")]

y <- df$Salary

# Fit the linear regression model

X <- as.matrix(df[, c("Balls\_Faced", "Total\_Runs")])

model <- lm(y ~ X)

model <- lm(y ~ Balls\_Faced + Total\_Runs, data = df)

# Print the summary of the model

summary(model)

# Get the coefficients

coefficients <- tidy(model)

# Print the coefficients

print(coefficients)

# Get the R-squared value

r\_squared <- glance(model)

# Print the R-squared value

print(r\_squared)

# Load the ggplot2 library

library(ggplot2)

# Create a scatterplot of the data with a regression line

ggplot(df, aes(x = Balls\_Faced, y = Total\_Runs)) +

geom\_point() +

geom\_smooth(method = "lm", se = FALSE) +

labs(x = "Balls Faced", y = "Total Runs") +

theme\_classic()