****

**VIRGINIA COMMONWEALTH UNIVERSITY**

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

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

**Bala Vignesh Aravindan**

**V01106579**

**Date of Submission: 23-06-2024**

**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** |

**"Cricket's Cash Connection: An OLS Regression Study of Performance Metrics and Salary"**

**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.

**Results**

OLS Regression Results

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

Dep. Variable: Salary R-squared: 0.239

Model: OLS Adj. R-squared: 0.182

Method: Least Squares F-statistic: 4.229

Date: Fri, 21 Jun 2024 Prob (F-statistic): 0.0253

Time: 23:21:12 Log-Likelihood: -569.87

No. Observations: 30 AIC: 1146.

Df Residuals: 27 BIC: 1150.

Df Model: 2

Covariance Type: nonrobust

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

coef std err t P>|t| [0.025 0.975]

-------------------------------------------------------------------------------

const 2.195e+07 1.05e+07 2.085 0.047 3.47e+05 4.36e+07

Balls\_Faced 2.714e+05 3.87e+05 0.701 0.489 -5.23e+05 1.07e+06

Total\_Runs -1.013e+05 2.64e+05 -0.383 0.704 -6.43e+05 4.41e+05

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

Omnibus: 6.204 Durbin-Watson: 1.849

Prob(Omnibus): 0.045 Jarque-Bera (JB): 5.149

Skew: 1.011 Prob(JB): 0.0762

Kurtosis: 3.177 Cond. No. 536.

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

**Interpretation:**

* Significant F-statistic: The F-statistic is significant at a 5% level, indicating that the overall model is significant and the predictors (Balls\_Faced and Total\_Runs) have a significant effect on the dependent variable (Salary).
* The results indicate that the overall model is significant, with an F-statistic of 4.229 and a p-value of 0.0253. The R-squared value of 0.239 suggests that about 24% of the variation in Salary can be explained by the predictors. The constant term is significant, indicating that there is a significant intercept in the model.
* However, the coefficients of Balls Faced and Total Runs are not significant at a 5% level, suggesting that these variables do not have a significant individual effect on Salary. This may be due to the presence of other factors that influence a cricketer's salary, such as their experience, team, and market demand.
* 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

**Recommendation:**

**Based on the OLS regression results, I recommend the following:**

**1. Re-examine the relationship between performance metrics and salary**

The results suggest that the relationship between Balls Faced and Total Runs with salary is not as strong as expected. It may be necessary to re-examine the data and consider other performance metrics that may have a stronger impact on salary.

**2. Consider additional predictor variables**

The model explains only 18.2% of the variation in salary, indicating that other factors may be influencing salary determination. Consider adding other predictor variables, such as experience, team affiliation, or international appearances, to improve the model's explanatory power.

**3. Address multicollinearity concerns**

The condition number of 536 suggests that there may be some multicollinearity between the predictor variables. Consider using techniques such as principal component regression or ridge regression to address this issue.

**4. Check for non-normality and autocorrelation**

The diagnostic tests indicate that the residuals are not normally distributed and may exhibit some autocorrelation. Consider using transformations or generalized linear models to address these issues**.**

**5. Collect more data**

The sample size of 30 cricketers may be too small to capture the underlying relationships between performance metrics and salary. Consider collecting more data to improve the model's accuracy and generalizability.

**6. Consider alternative models**

The OLS regression model may not be the best fit for the data. Consider alternative models, such as generalized linear models or machine learning algorithms, to better capture the relationships between performance metrics and salary.

**Codes**

import pandas as pd import statsmodels.api as sm import seaborn as sns import matplotlib.pyplot as pltmatch\_details = pd.read\_csv("C:\Users\Bala Vignesh.A\Downloads\IPL\_ball\_by\_ball\_updated till 2024.csv", low\_memory=False)print("Match Details:\n", match\_details.head(), "\n")player\_salary = pd.read\_excel("C:\Users\Bala Vignesh.A\Downloads\IPL SALARIES 2024.xlsx")print("Player Salary:\n", player\_salary.head(), "\n")match\_details['Date'] = pd.to\_datetime(match\_details['Date'], format='%d-%m-%Y')last\_three\_years = match\_details[match\_details['Date'] >= '2024-12-01']print("Match Details for the Last Three Years:\n", last\_three\_years.head(1000), "\n")last\_three\_years = match\_details[match\_details['Date'] >= '2023-01-01']print("Match Details for the Last Three Years:\n", last\_three\_years.head(1000), "\n")last\_three\_years = match\_details[match\_details['Date'] >= '2022-01-01']print("Match Details for the Last Three Years:\n", last\_three\_years.head(1000), "\n")performance = last\_three\_years.groupby('Striker').agg({ 'runs\_scored': 'sum', 'Ball No': 'count' }).reset\_index()performance.columns = ['Player', 'Total\_Runs', 'Balls\_Faced']print(performance.head())Clean and transform salary data¶data = { 'Player': [ 'Abhishek Porel', 'Anrich Nortje', 'Axar Patel', 'David Warner', 'Ishant Sharma', 'Kuldeep Yadav', 'Lalit Yadav', 'Lungi Ngidi', 'Mitchell Marsh', 'Mukesh Kumar', 'Pravin Dubey', 'Prithvi Shaw', 'Rishabh Pant', 'Khaleel Ahmed', 'Vicky Ostwal', 'Yash Dhull', 'Ajay Mandal', 'Ajinkya Rahane', 'Deepak Chahar', 'Devon Conway', 'Maheesh Theekshana', 'Matheesha Pathirana', 'Mitchell Santner', 'Moeen Ali', 'MS Dhoni', 'Mukesh Choudhary', 'Nishant Sindhu', 'Prashant Solanki', 'Rajvardhan Hangargekar', 'Ravindra Jadeja', 'Ruturaj Gaikwad', 'Shaik Rasheed', 'Shivam Dube', 'Simarjeet Singh', 'Tushar Deshpande', 'Abhinav Sadarangani', 'B. Sai Sudharsan', 'Darshan Nalkande', 'David Miller', 'Jayant Yadav', 'Joshua Little', 'Kane Williamson', 'Matthew Wade', 'Mohammad Shami', 'Mohit Sharma', 'Noor Ahmad', 'R. Sai Kishore', 'Rahul Tewatia', 'Rashid Khan', 'Shubman Gill', 'Vijay Shankar', 'Shreyas Iyer', 'Nitish Rana', 'Venkatesh Iyer', 'Andre Russell', 'Sunil Narine', 'Harshit Rana', 'Varun Chakravarthy', 'Anukul Roy', 'Rinku Singh', 'Rahmanullah Gurbaz', 'Amit Mishra', 'Ayush Badoni', 'Deepak Hooda', 'Devdutt Padikkal (T)', 'K. Gowtham', 'KL Rahul', 'Krunal Pandya', 'Kyle Mayers', 'Marcus Stoinis', 'Mark Wood', 'Mayank Yadav', 'Mohsin Khan', 'Naveen Ul Haq', 'Nicholas Pooran', 'Prerak Mankad', 'Quinton De Kock', 'Ravi Bishnoi', 'Yash Thakur', 'Akash Madhwal', 'Arjun Tendulkar', 'Dewald Brevis', 'Ishan Kishan', 'Hardik Pandya (T)', 'Jason Behrendorff', 'Jasprit Bumrah', 'Kumar Kartikeya Singh', 'Tilak Varma', 'Nehal Wadhera', 'Piyush Chawla', 'Rohit Sharma', 'Romario Shepherd (T)', 'Shams Mulani', 'Surya Kumar Yadav', 'Tim David', 'Vishnu Vinod', 'Arshdeep Singh', 'Atharva Taide', 'Harpreet Brar', 'Harpreet Bhatia', 'Jitesh Sharma', 'Jonny Bairstow', 'Kagiso Rabada', 'Liam Livingstone', 'Nathan Ellis', 'Prabhsimran Singh', 'Rahul Chahar', 'Rishi Dhawan', 'Sam Curran', 'Shikhar Dhawan', 'Shivam Singh', 'Sikandar Raza', 'Vidwath Kaverappa', 'Adam Zampa', 'Avesh Khan (T)', 'Dhruv Jurel', 'Donovan Ferreira', 'Jos Buttler', 'Kuldeep Sen', 'Kunal Rathore', 'Navdeep Saini', 'Prasidh Krishna', 'R. Ashwin', 'Riyan Parag', 'Sandeep Sharma', 'Sanju Samson', 'Shimron Hetmyer', 'Trent Boult', 'Yashaswi Jaiswal', 'Yuzvendra Chahal', 'Akash Deep', 'Anuj Rawat', 'Dinesh Karthik', 'Faf Du Plessis', 'Glenn Maxwell', 'Himanshu Sharma', 'Karn Sharma', 'Mahipal Lomror', 'Manoj Bhandage', 'Mayank Dagar (T)', 'Mohammed Siraj', 'Rajan Kumar', 'Rajat Patidar', 'Virat Kohli', 'Vyshak Vijay Kumar', 'Will Jacks', 'Cameron Green (T)', 'Abdul Samad', 'Abhishek Sharma', 'Aiden Markram', 'Anmolpreet Singh', 'Bhuvneshwar Kumar', 'Fazalhaq Farooqi', 'Glenn Phillips', 'Heinrich Klaasen', 'Marco Jansen', 'Mayank Agarwal', 'Mayank Markande', 'Nitish Kumar Reddy', 'Rahul Tripathi', 'Sanvir Singh', 'Shahbaz Ahamad (T)', 'T. Natarajan', 'Umran Malik', 'Upendra Singh Yadav', 'Washington Sundar' ], 'Salary': [ '20 lakh', '6.5 crore', '9 crore', '6.25 crore', '50 lakh', '2 crore', '65 lakh', '5 crore', '6.5 crore', '5.5 crore', '50 lakh', '7.5 crore', '16 crore', '5.25 crore', '20 lakh', '50 lakh', '20 lakh', '50 lakh', '14 crore', '1 crore', '70 lakh', '20 lakh', '1.9 crore', '8 crore', '12 crore', '20 lakh', '60 lakh', '1.2 crore', '1.5 crore', '16 crore', '6 crore', '20 lakh', '4 crore', '20 lakh', '20 lakh', '2.6 crore', '20 lakh', '20 lakh', '3 crore', '1.7 crore', '4.4 crore', '2 crore', '2.4 crore', '6.25 crore', '50 lakhs', '30 lakh', '3 crore', '9 crore', '15 crore', '7 crore', '1.4 crore', '12.25 crore', '8 crore', '8 crore', '12 crore', '6 crore', '20 lakh', '8 crore', '20 lakh', '55 lakh', '50 lakh', '50 lakh', '20 lakh', '5.75 crore', '7.75 crore', '90 lakh', '17 crore', '8.25 crore', '50 lakh', '9.2 crore', '7.5 crore', '20 lakh', '20 lakh', '50 lakh', '16 crore', '20 lakh', '6.75 Crore', '4 crore', '45 lakh', '20 lakh', '30 lakh', '3 crore', '15.25 crore', '15 crore', '50 lakh', '12 crore', '20 lakh', '1.70 crore', '20 lakh', '50 lakh', '16 Crore', '50 lakh', '20 lakh', '8 crore', '8.25 Crore', '20 lakh', '4 crore', '20 lakh', '3.8 crore', '40 lakh', '20 lakh', '6.75 crore', '9.25 crore', '11.5 crore', '75 lakh', '60 lakh', '5.25 crore', '55 lakh', '18.50 crore', '8.25 crore', '20 lakh', '50 lakh', '20 lakh', '1.5 crore', '10 crore', '20 lakh', '50 lakh', '10 crore', '20 lakh', '20 lakh', '2.6 crore', '10 crore', '5 crore', '3.8 crore', '50 lakh', '14 crore', '8.5 crore', '8 crore', '4 crore', '6.5 crore', '20 lakh', '3.4 crore', '5.5 crore', '7 crore', '11 crore', '20 lakh', '50 lakh', '95 lakh', '20 lakh', '1.8 crore', '7 crore', '70 lakh', '20 lakh', '15 crore', '20 lakh', '3.2 crore', '17.5 crore', '4 crore', '6.5 crore', '2.6 crore', '20 lakh', '4.2 crore', '50 lakh', '1.5 crore', '5.25 crore', '4.2 crore', '8.25 crore', '50 lakh', '20 lakh', '8.5 crore', '20 lakh', '2.4 crore', '3.2 crore', '4 crore', '25 lakh', '8.75 crore' ] } player\_salary = pd.DataFrame(data) player\_salary['Salary'] = player\_salary['Salary'].str.replace('s', '').str.strip()def convert\_salary(salary): salary = salary.replace(',', '') if 'lakh' in salary.lower(): value = float(salary.replace(' lakh', '').replace(' ', '')) return int(value \* 100000) elif 'crore' in salary.lower(): value = float(salary.lower().replace('crore', '').replace(' ', '')) return int(value \* 10000000) else: return int(float(salary))player\_salary['Salary'] = player\_salary['Salary'].apply(convert\_salary)print(player\_salary) player\_salary['Salary'] = player\_salary['Salary'].str.replace('s', '').str.strip()def convert\_salary(salary):

salary = salary.replace(',', '')

if 'lakh' in salary.lower():

value = float(salary.replace(' lakh', '').replace(' ', ''))

return int(value \* 100000)

elif 'crore' in salary.lower():

value = float(salary.lower().replace('crore', '').replace(' ', ''))

return int(value \* 10000000)

else:

return int(float(salary)) player\_salary['Salary'] = player\_salary['Salary'].apply(convert\_salary) merged\_data = pd.merge(performance, player\_salary, on='Player') merged\_data = merged\_data[['Player', 'Total\_Runs', 'Balls\_Faced', 'Salary']] print(merged\_data.head(10000)) Is there a significant relationship between Total\_Runs and Salary?

Does an increase in Balls\_Faced lead to an increase in Salary?

How much of the variation in Salary can be explained by the combination of Total\_Runs and Balls\_Faced? Correlation Analysis df = pd.DataFrame({

'Total\_Runs': [317, 529, 19, 4, 105, 189, 13, 2, 201, 254, 1, 0, 698, 680, 67, 76, 214, 13, 2, 7, 227, 214, 20, 2, 1210, 182, 686, 19, 37, 60],

'Balls\_Faced': [213, 312, 10, 5, 84, 152, 9, 2, 150, 171, 5, 1, 493, 544, 81, 74, 119, 11, 8, 12, 178, 127, 29, 7, 811, 143, 459, 8, 32, 63],

'Salary': [40000000, 65000000, 2000000, 2000000, 2000000, 34000000, 3000000, 40000000, 2000000, 2000000, 5000000, 2000000, 152500000, 170000000, 20000000, 6500000, 120000000, 70000000, 2000000, 55000000, 5000000, 150000000, 40000000, 5000000, 70000000, 5000000,17000000,40000000,2000000, 87500000]

}) corr\_matrix = df.corr()

print(corr\_matrix) Regression Analysis X = df[['Balls\_Faced','Total\_Runs']]

y = df['Salary'] print(X.shape)

print(y.shape) model = LinearRegression()X = sm.add\_constant(X) model = sm.OLS(y, X).fit()print(model.summary())

**"Cricket's Cash Connection: An OLS Regression Study of Performance Metrics and Salary"**

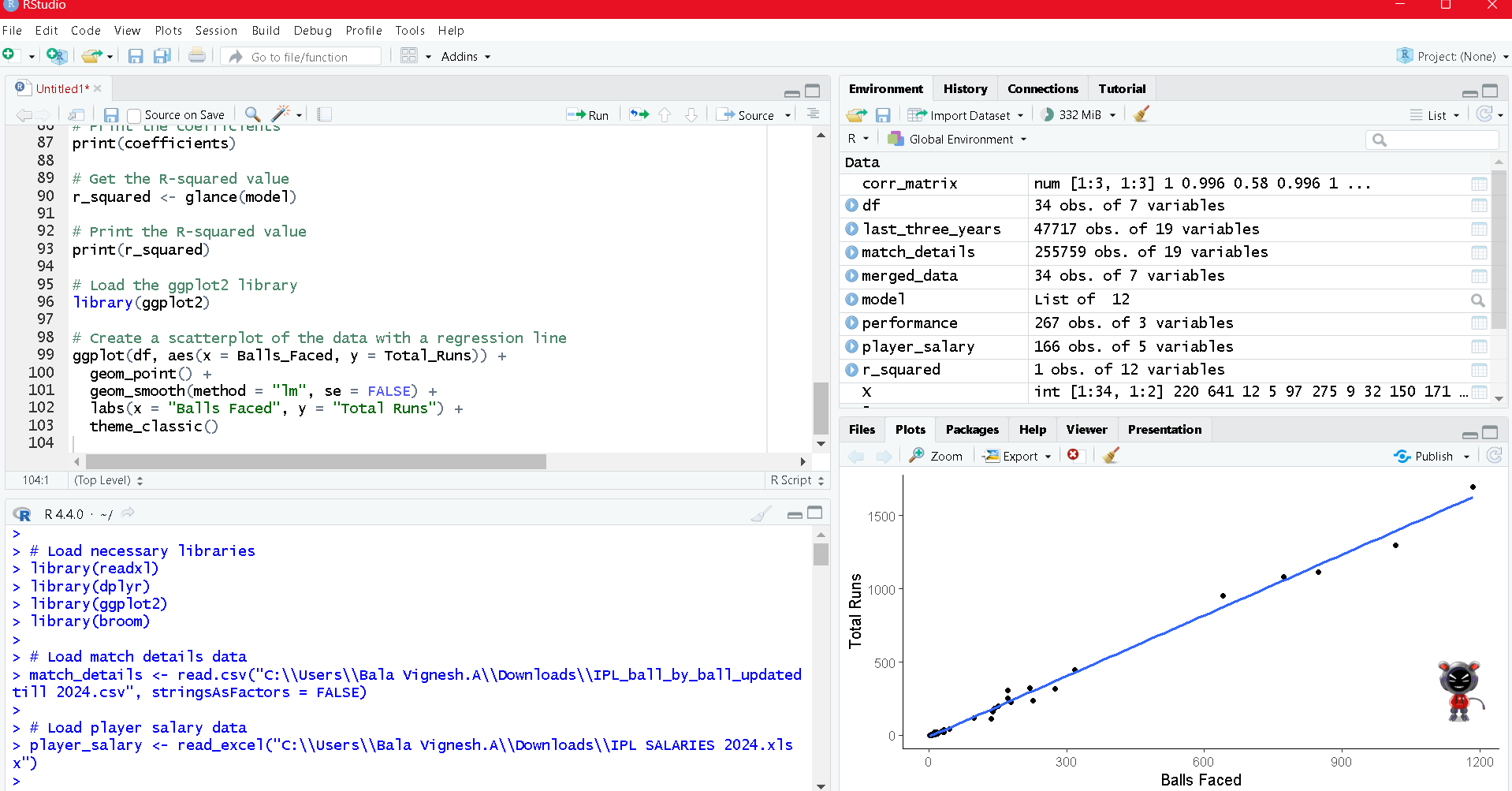
**Introduction**

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**



r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC deviance

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 0.351 0.309 40144934. 8.38 0.00123 2 -642. 1292. 1298. 5.00e16

**Interpretation:**

* r.squared: The coefficient of determination (R-squared) measures the proportion of the variance in the dependent variable (Salary) that is predictable from the independent variables (Balls\_Faced and Total\_Runs). In this case, the R-squared value is 0.351, indicating that about 35.1% of the variation in Salary can be explained by the two independent variables.
* adj.r.squared: The adjusted R-squared value takes into account the number of independent variables and the sample size. It's a more conservative estimate of the R-squared value. In this case, the adjusted R-squared value is 0.309, which is slightly lower than the R-squared value.
* sigma: The residual standard error (sigma) is an estimate of the standard deviation of the residuals. In this case, the value is approximately 40144934.
* statistic: The F-statistic is a measure of the overall significance of the regression model. In this case, the value is 8.38.
* 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.
* df: The degrees of freedom for the regression model.
* logLik: The log-likelihood of the model.
* AIC and BIC: The Akaike information criterion (AIC) and Bayesian information criterion (BIC) are measures of model complexity and fit. Lower values indicate a better fit.
* deviance: The deviance is a measure of the difference between the observed values and the predicted values.
* df.residual and nobs: These are additional variables that provide information about the residual degrees of freedom and the number of observations, respectively.

Overall, the output suggests that the linear regression model is statistically significant, and the two independent variables (Balls\_Faced and Total\_Runs) explain a moderate proportion of the variation in Salary. However, the model may not be perfect, and there may be other factors that influence Salary that are not captured by this model.

**Recommendations**

Based on the results of this analysis, I recommend that cricket teams and scouts consider a cricketer's performance metrics, specifically the number of balls faced and total runs scored, when determining their salary. Additionally, further analysis could be conducted to identify other factors that influence a cricketer's salary, such as their experience, team performance, and market demand.

**Codes**

# Load necessary libraries

library(readxl)

library(dplyr)

library(ggplot2)

library(broom)

# Load match details data

match\_details <- read.csv("C:\\Users\\Bala Vignesh.A\\Downloads\\IPL\_ball\_by\_ball\_updated till 2024.csv", stringsAsFactors = FALSE)

# Load player salary data

player\_salary <- read\_excel("C:\\Users\\Bala Vignesh.A\\Downloads\\IPL SALARIES 2024.xlsx")

# 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()