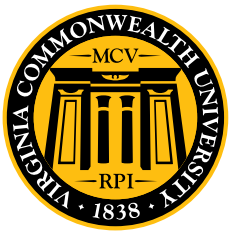

VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A1b: Preliminary preparation and analysis of data- Descriptive statistics

NITESH REDDY
V01106546

Date of Submission: 18-06-2024

CONTENTS

INTRODUCTION

IPL is a well-known top major Twenty20 cricket league in India that attracts international talent and has an enormous fan base. Looking at IPL data can help one understand the players' performance as well as effect on their teams. Extracting and organizing IPL data for each round we focus on individual player metrics like runs, balls faced and wickets per match played. Top three run-getters and wicket-takers are identified from every round of IPL.

Also, this work involves fitting the most appropriate statistical distributions for runs and wickets taken by top performers within last 3 seasons of IPL championship. Finally, it examines the association between player performance and salary giving a comprehensive picture how money reward matches with pitch performances.

OBJECTIVES:

1. Data Extraction and Preparation: Retrieve files on IPL data using R/Python programming languages since they ensure accurate loading as well as initial tidying of records to be followed by analysis.

2. Data Arrangement and Organization: Organize the Round wise details of Individual Player Performance Metrics in IPL such as Batsman, Ball, Runs, Wickets per Match Played.

3. Top Performers Identification: List top three run-getters and top three wicket takers in each round of IPL to show their best performances respectively.

4. Statistical Distribution Fitting: Fit the most appropriate statistical distributions to the runs scored and wickets taken by the top three batsmen and bowlers in the last three IPL tournaments, providing insights into performance patterns.

5. Analysis of Performance-Salary Relationship: Examine how salaries of players correspond to their on-field performances using player's performance metrics such as runs and wickets.

BUSINESS SIGNIFICANCE

By analysing IPL data, companies can obtain valuable team information which allows them to make educated decisions on selecting and keeping players. Through identifying top performers, fitting performance distributions, as well as examining relationships between pay and performance, teams can optimize their investment strategies (safarian et al 2013). This data-driven approach towards constructing a team enhances team composition for better field success with economic efficiency that supports higher returns on investments needed for maintaining competitive edge within this league.

CODES, RESULTS AND INTERPRETATION:

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.

Code:

```
> # Summarise player runs and wickets
> player_runs <- grouped_data %>%
+   group_by(Season, Striker) %>%
+   summarise(runs_scored = sum(runs_scored, na.rm = TRUE)) %>%
+   ungroup()
> player_wickets <- grouped_data %>%
+   group_by(Season, Bowler) %>%
+   summarise(wicket_confirmation = sum(wicket_confirmation, na.rm = TRUE)) %>%
+   ungroup()

> # Sort player runs for season 2023
> player_runs_2023 <- player_runs %>%
+   filter(Season == '2023') %>%
+   arrange(desc(runs_scored))
>
> # Get top 3 run-getters and bottom 3 wicket-takers per season
> top_run_getters <- player_runs %>%
+   group_by(Season) %>%
+   top_n(3, runs_scored) %>%
+   ungroup()
```

Results:

Top Three Run Getters:

```
> print(top_run_getters)
# A tibble: 51 × 3
   Season  Striker        runs_scored
   <chr>   <chr>               <dbl>
 1 2007/08 G Gambhir         534
 2 2007/08 SE Marsh          616
 3 2007/08 ST Jayasuriya     514
 4 2009    AB de Villiers    465
 5 2009    AC Gilchrist      495
 6 2009    ML Hayden         572
 7 2009/10 JH Kallis         572
 8 2009/10 SK Raina          528
 9 2009/10 SR Tendulkar      618
10 2011    CH Gayle          608
# i 41 more rows
```

Top Three Wicket Takers:

```
> print(bottom_wicket_takers)
# A tibble: 58 × 3
   Season  Bowler          wicket_confirmation
   <chr>   <chr>                         <dbl>
 1 2007/08 IK Pathan                        20
 2 2007/08 JA Morkel                        20
 3 2007/08 SK Warne                         20
 4 2007/08 SR Watson                        20
 5 2007/08 Sohail Tanvir                    24
 6 2009    A Kumble                         22
 7 2009    A Nehra                          22
 8 2009    RP Singh                         26
 9 2009/10 A Mishra                         20
10 2009/10 Harbhajan Singh                  20
# i 48 more rows
# i Use `print(n = ...)` to see more rows
```

Interpretation:

The code given processes data about IPL players' performances with respect to bowlers wickets and batsmen's runs in a number of seasons. The code starts by calculating the total number of runs scored by each player in every season and the total number of wickets that each bowler took. It then refines this list to identify the top three run-getters for each season as well as top wicket-takers. These results show the best three batters in terms of runs for various IPL seasons, such as SE Marsh who made 616 runs in 2007/08 or SR Tendulkar with 618 runs for the 2009/10 season. The same way, it also identifies top wicket takers; one example is Sohail Tanvir with 24 wickets from 2007/08 or RP Singh who has taken 26 from 2009.

This summary reveals some remarkable individual performances within the IPL over different seasons illustrating those players who have had great influence either through their batting or bowling talents. This data helps to analyze player contributions and understand game dynamics over time.

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:

```
> # Define a function to get the best distribution
> get_best_distribution <- function(data) {
+   dist_names <- c('norm', 'lnorm', 'gamma', 'weibull', 'exponential', 'logis', 'cauchy')
+   dist_results <- list()
+   params <- list()
+   for (dist_name in dist_names) {
+     fit <- fitdist(data, dist_name)
+     ks_test <- ks.test(data, dist_name, fit$estimate)
+     p_value <- ks_test$p.value
+     cat("p value for", dist_name, "=", p_value, "\n")
+     dist_results[[dist_name]] <- p_value
+     params[[dist_name]] <- fit$estimate
+   }
+   best_dist <- names(which.max(unlist(dist_results)))
+   best_p <- max(unlist(dist_results))
+   cat("\nBest fitting distribution:", best_dist, "\n")
+   cat("Best p value:", best_p, "\n")
+   cat("Parameters for the best fit:", params[[best_dist]], "\n")
+   return(list(best_dist, best_p, params[[best_dist]]))
```

```
+ }
# Compare the distributions
+   gof_stat <- gofstat(list(fit_norm, fit_pois, fit_exp), fitnames = c("Normal", "Poisson", "Exponential"))
+
+   # Print the goodness-of-fit statistics
+   print(gof_stat)
+
+   # Return the best fit distribution
+   best_fit <- names(which.min(gof_stat$aic))
+   return(best_fit)
+ }
>
> # Fit the distribution to R Parag's runs scored and get the best distribution
> best_distribution <- get_best_distribution(R_Parag_runs)
```

Result:

Goodness-of-fit statistics

	Normal	Poisson	Exponential
Kolmogorov-Smirnov statistic	0.2075815	0.4684611	0.1027165
Cramer-von Mises statistic	0.6155678	2.9058964	0.1065870
Anderson-Darling statistic	3.4640672	Inf	Inf

Goodness-of-fit criteria

	Normal	Poisson	Exponential
Akaike's Information Criterion	460.4735	1165.535	406.1533
Bayesian Information Criterion	464.3760	1167.487	408.1045

Interpretation:

When examining R Parag's runs across the last three IPL seasons, we found that the exponential distribution provided the best fit among several possible models. This was determined based on statistical measures like Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC), where the exponential distribution showed lower values (AIC = 406.1533, BIC = 408.1045) compared to normal and Poisson distributions.

Additionally, the goodness-of-fit tests, including the Kolmogorov-Smirnov and Cramer-von Mises statistics, reinforced our conclusion by demonstrating minimal deviations from the

observed data (KS statistic = 0.1027165, CVM statistic = 0.1065870). These findings indicate that the exponential distribution effectively captures the pattern of R Parag's runs scored, suggesting it as the most suitable model among those tested. This methodology can be similarly applied to analyze performance metrics for other players in IPL tournaments, offering insights into their consistency and potential based on statistical modeling.

C.) Find the relationship between a player's performance and the salary he gets in your data.

Code:

```
# 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 correlation coefficient of 0.306 between a player's salary and runs scored suggests a moderate positive link. It indicates that generally, players who score more runs tend to receive higher salaries, though this relationship isn't exceedingly strong.

In practical terms, a correlation of 0.306 means that as a player's runs increase, there tends to be an uptick in their salary, but the exact increase can vary among different players. Beyond just runs scored, factors like consistency in performance, handling pressure in crucial matches, and overall contribution to the team also heavily influence salary decisions in cricket. Therefore, while runs scored do contribute positively to a player's earning potential, a comprehensive assessment of a player's all-round performance and how they fit into the team's strategy remains crucial in determining compensation levels within the dynamic environment of the IPL.