Analysis of the Cricket World Cup 2023 tournament

IS2105 - Business Statistics

Final Report – Group 13

24.09.2024

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

Information Systems Degree Program

2105 - Business Statistics

FINAL REPORT

Group 13

Index No	Name
22020276	P. K. K. D. Gunaratne
22020551	E. G. A. Mandara
22020642	L. A. I. Naveen
22020047	S. N. Ahmed

Table of Content

1. Introduction

Cricket, often described as a religion in many parts of the world, reached its quadrennial crescendo with the ICC Cricket World Cup 2023. This prestigious tournament, hosted by India, brought together the top cricket-playing nations in a battle for supremacy in the One Day International (ODI) format. As the dust settles on this cricketing spectacle, we are left with a treasure trove of data that tells the story of the tournament far beyond what meets the eye.

Context and Background

The Cricket World Cup is more than just a series of matches; it's a global event that captures the imagination of millions. The 2023 edition, featuring 10 teams playing 48 matches across various venues in India, provided a unique opportunity to study the game at its highest level. With diverse playing conditions, varying team strategies, and individual brilliance on display, the tournament offers a rich dataset for analysis.

In the modern era of cricket, data analytics has become as crucial as physical skills. Teams employ statisticians and analysts to gain every possible advantage. This report aims to delve into the numbers behind the spectacle, uncovering patterns and insights that can enhance our understanding of the game and potentially influence future strategies.

Problems of Interest

While the Cricket World Cup 2023 was a visual feast for fans, it also raised several intriguing questions:

1. How did different venues impact match outcomes and playing styles?
2. What factors contributed most significantly to a team's success in the tournament?
3. How did individual performances correlate with team success?
4. Were there discernible patterns in batting and bowling performances across the tournament?

5. How did the decision to bat or bowl first affect match results across different venues?

Objectives

To address these questions, our analysis focuses on the following objectives:

1. To analyze the impact of venues on match outcomes, including run-scoring patterns and wicket-taking rates.
2. To evaluate team performances based on various metrics and identify key factors contributing to success.
3. To assess individual player contributions and their influence on team results.
4. To investigate patterns in batting and bowling performances across different stages of innings and the tournament.
5. To examine the relationship between winning the toss, choosing to bat or bowl, and match results.

Significance of the Analysis

The significance of this analysis extends beyond mere curiosity. By dissecting the data from the Cricket World Cup 2023, we aim to:

1. Provide insights that can inform team strategies in future tournaments and series.
2. Offer a deeper understanding of how various factors interplay in determining cricket match outcomes.
3. Contribute to the growing field of sports analytics, demonstrating the value of data-driven decision-making in cricket.
4. Enhance fan engagement by offering a more nuanced understanding of the game.
5. Potentially influence how future tournaments are organized, including considerations for venue selection and scheduling.

In an era where margins between teams are increasingly thin, the insights gleaned from this analysis could provide the critical edge that separates champions from contenders. Moreover, for the cricketing ecosystem - including coaches, players, administrators, and fans - this analysis offers a window into the evolving nature of the sport, highlighting trends that may shape the future of cricket.

As we embark on this analytical journey through the Cricket World Cup 2023, we invite readers to look beyond the boundaries and sixes, the wickets and catches, and explore the hidden narratives that numbers reveal about this beautiful game.

2. Data Description

Dataset Source:

The dataset for the Cricket World Cup 2023 analysis was obtained from Kaggle, a popular platform for data science and machine learning enthusiasts. This dataset provides a comprehensive view of player and team performances throughout the tournament.

Data Collection:

While the exact collection method isn't specified, it's likely that the data was compiled from official match statistics, possibly sourced from the International Cricket Council (ICC) or cricket statistics databases like ESPNcricinfo's Statsguru. The granularity of the data suggests a ball-by-ball collection method, which was then aggregated to per-innings level for each player.

Scope and Structure:

The dataset covers all matches of the Cricket World Cup 2023, hosted by India. It includes data from 48 matches played across various venues, featuring 10 international cricket teams. Each row in the dataset represents a single performance by a player in a specific match - either a batting innings or a bowling spell.

Participants:

The dataset includes all teams and players who participated in the Cricket World Cup 2023. Teams are identified by 2-3 letter codes (e.g., "IND" for India, "AUS" for Australia). Individual players are identified by name, allowing for player-specific analysis across multiple matches.

Timeframe:

The data covers the entire duration of the Cricket World Cup 2023, with specific match dates provided in the 'start_date' variable. This allows for temporal analysis of performance trends throughout the tournament.

Variables:

The dataset is rich in both categorical and quantitative variables:

Categorical Variables:

1. team: Team code (e.g., IND, AUS)
2. player: Player's name
3. bat_or_bowl: Indicates if the entry is for batting or bowling performance
4. opposition: Opposing team's code
5. ground: Venue of the match
6. start_date: Date of the match
7. inns: Innings number (1 or 2)
8. not_out: Indicates if a batsman remained not out (0 or 1)

Quantitative Variables:

Batting metrics:

1. runs: Runs scored
2. bb_bf: Balls faced
3. 4s: Number of fours hit
4. 6s: Number of sixes hit
5. sr: Strike rate
6. mins: Time batted in minutes

Bowling metrics:

1. overs: Overs bowled
2. mdns: Maiden overs bowled
3. runs: Runs conceded

4. wkts: Wickets taken
5. econ: Economy rate

Derived Variables:

1. wicketball_prob: Probability of taking/losing a wicket per ball
2. runs_per_ball: Average runs scored/conceded per ball

Data Quality and Preprocessing:

The dataset has undergone some cleaning and preprocessing, particularly for the bb_bf (balls bowled/faced) variable. This likely involved handling extras (like wides or no-balls) and resolving any discrepancies in the raw data.

Limitations:

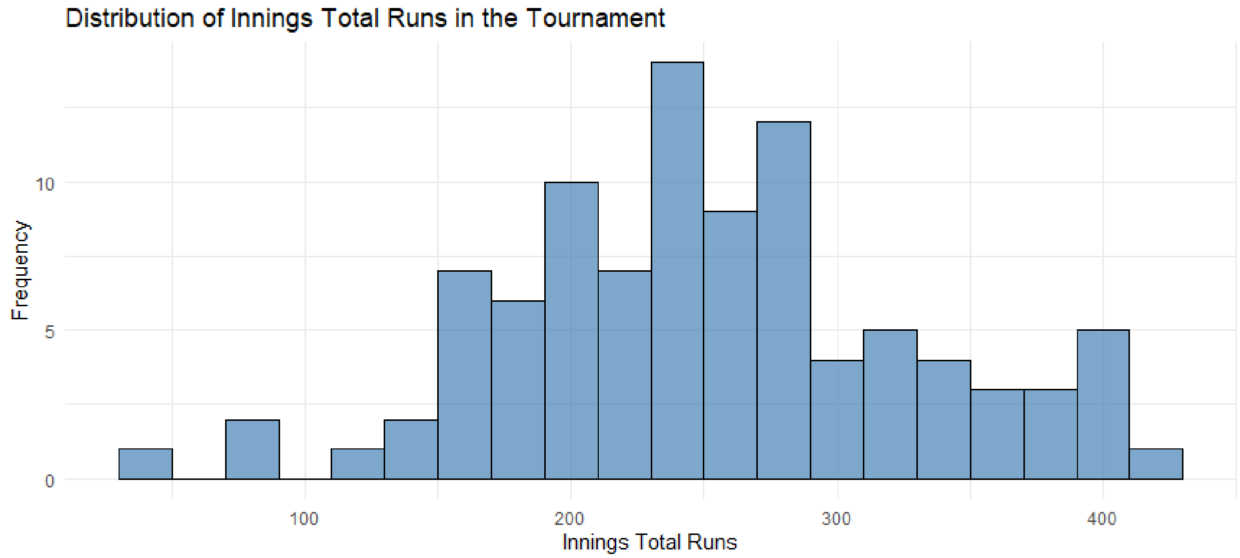
While comprehensive, the dataset doesn't include some potentially relevant factors like weather conditions, pitch reports, or player fitness levels, which could influence performance.

This dataset offers a rich foundation for various types of cricket analytics, including player and team performance analysis, venue-specific trends, tournament progression analysis, and efficiency analysis using derived metrics. Its comprehensive nature allows for in-depth exploration of the various factors that influenced the outcomes of the Cricket World Cup 2023.

3. Analysis

3.1. Univariate Analysis

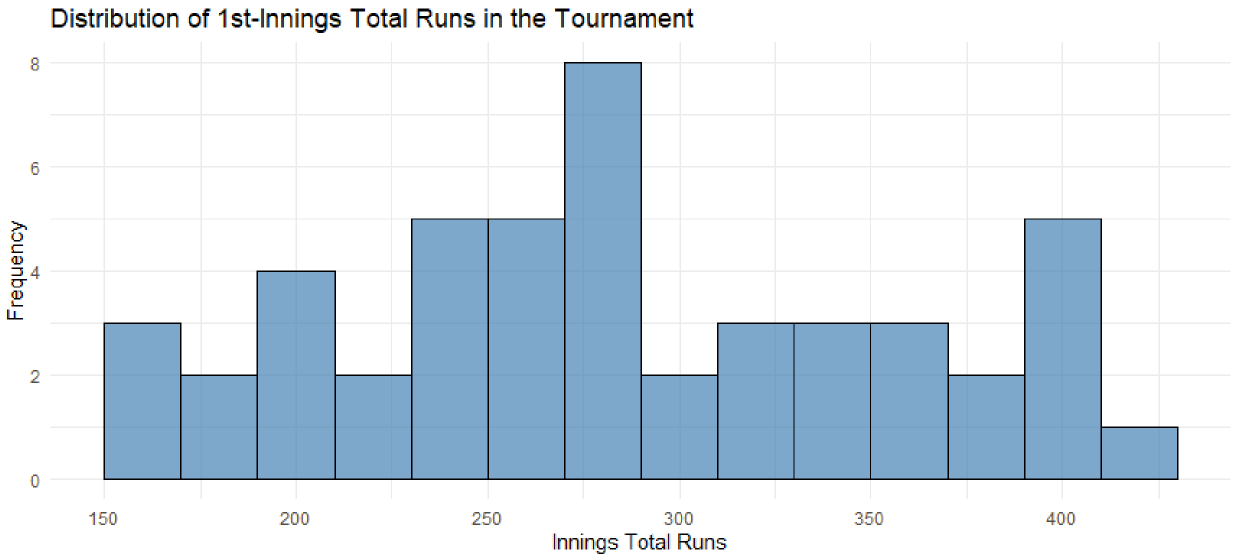
3.1.1. Score distribution of innings



The score distribution gives us a bird's-eye view of how teams performed with the bat throughout the tournament. Imagine this as a snapshot of all the team scores put together.

When we look at all innings combined, we see that most teams scored between 200 and 300 runs. This range is like the "sweet spot" for cricket scores in this tournament. Scores below 200 were relatively rare, suggesting that most teams managed to put up a decent fight. On the flip side, scores above 300 were also less common, indicating that while teams often scored well, it wasn't easy to reach those really high totals.

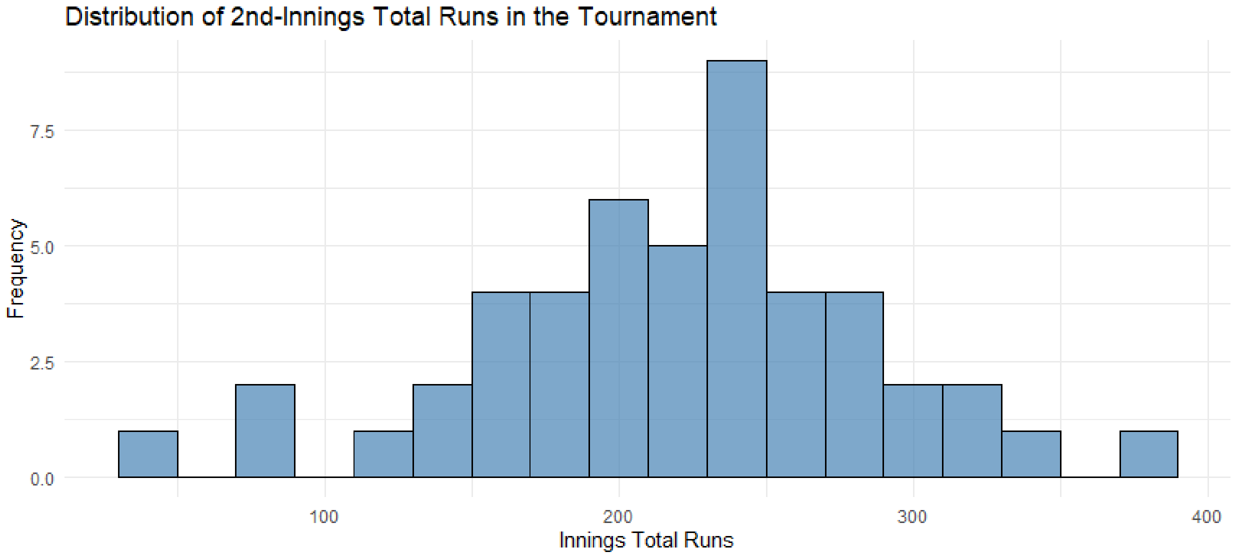
3.1.1.1. Score distribution of 1st inning



Now, let's focus on teams batting first. These teams have the advantage of setting the target without the pressure of chasing. Interestingly, we see a slight shift in the scores. The most common scores for teams batting first were between 275 and 300 runs.

This tells us an interesting story. Teams batting first generally aimed to set higher targets. It's like they were saying, "We're going to put a big number on the board and challenge you to beat it." This strategy makes sense because they don't know what a "good" score is yet, so they try to score as much as possible.

3.1.1.2. Score distribution of 2nd inning

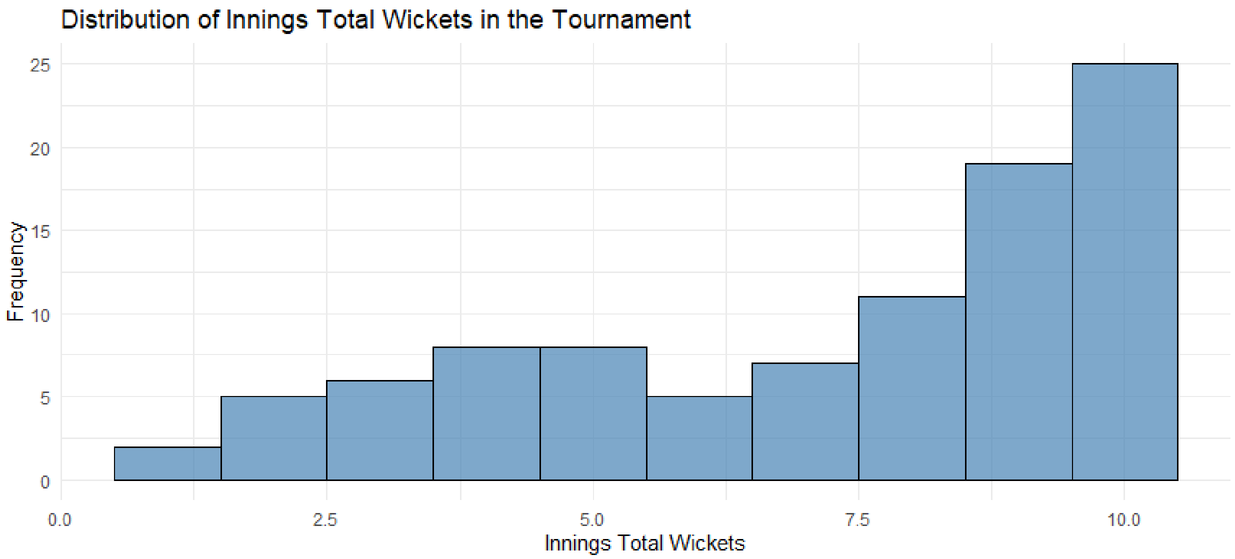


Switching gears to teams batting second, we see a different picture. The most common scores here were between 225 and 250 runs. This is lower than what we saw for teams batting first.

Why the difference? Well, teams batting second are chasing a target. They know exactly what they need to win, and once they reach that target, the game is over. So, if a team needs 230 to win and they get there, they stop - they don't keep going to 300 just because they can. This is why we see lower scores for second innings.

Also, batting second can be trickier. The pitch (the playing surface) might have worn down a bit, making it harder to score. Plus, there's the pressure of the chase - knowing exactly how many runs you need can sometimes make batting more challenging psychologically.

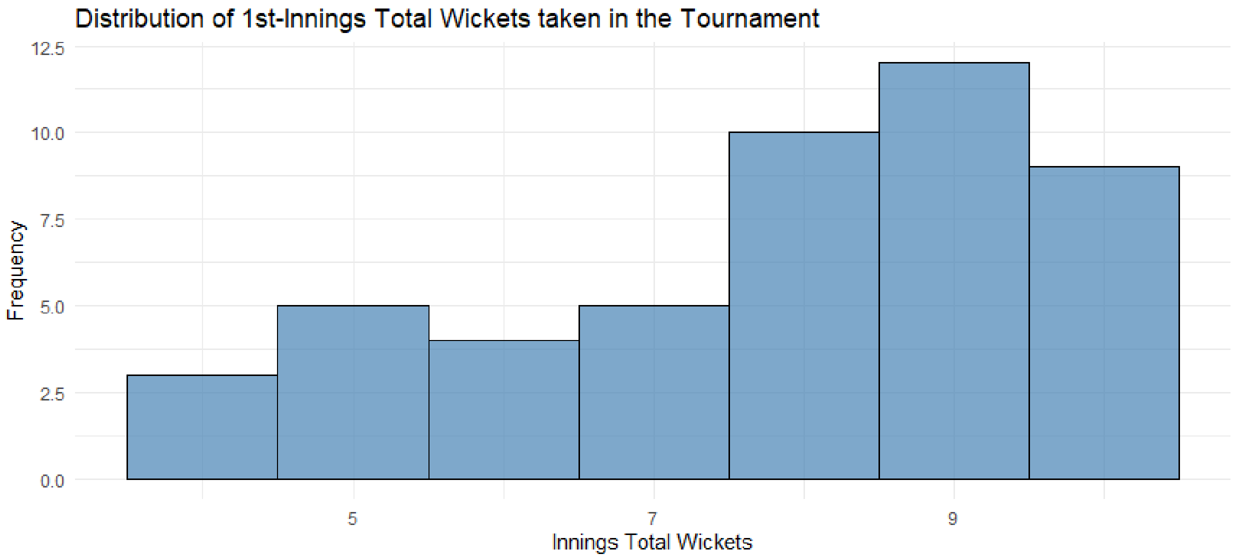
3.1.2. Wicket distribution of innings



Now, let's talk about wickets - the number of batsmen each team lost during their innings. This gives us insight into how well the bowling teams performed.

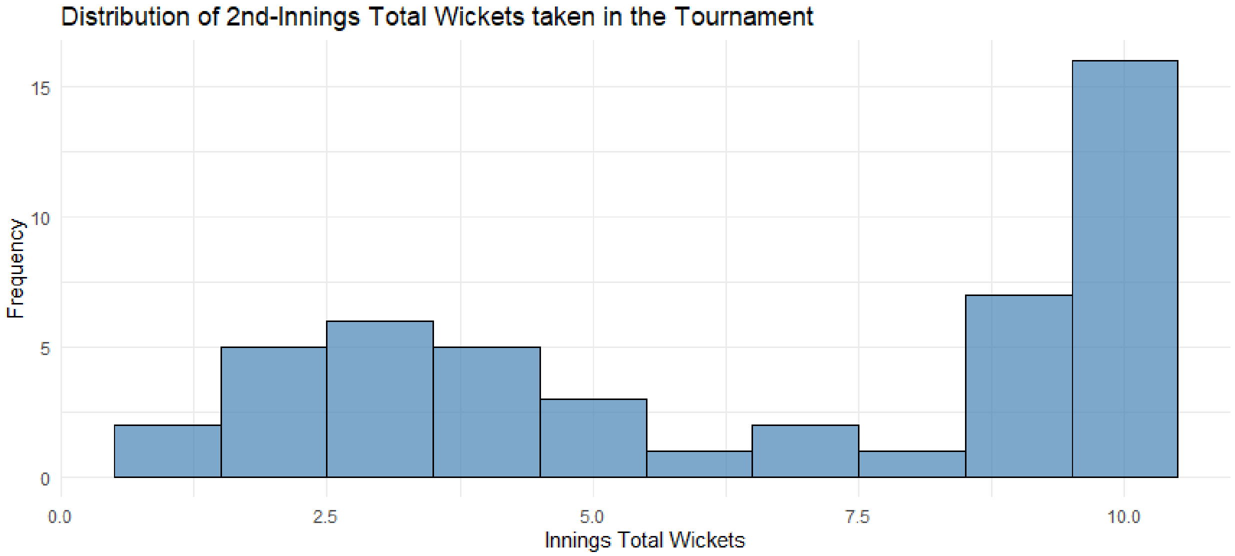
Looking at all the innings together, we see something interesting. The most common outcome was all 10 wickets falling. In cricket terms, this means the batting team was "all out". It happened more often than not, which tells us that bowlers generally had a good tournament.

3.1.2.1. Wicket distribution of 1st inning



For teams batting first, we see a similar pattern to the overall distribution. Most often, all 10 wickets fell. This suggests that teams batting first usually used up all their batsmen, either trying to score as many runs as possible or batting out their full 50 overs.

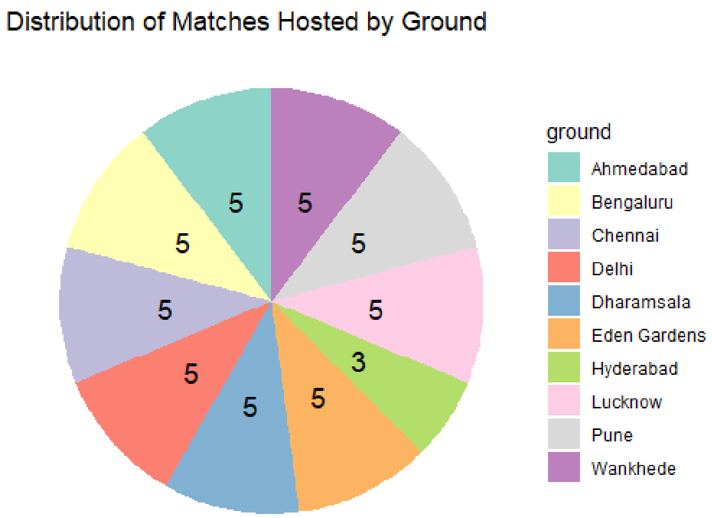
3.1.2.2. Wicket distribution of 2nd inning



The second innings show a different story. There's more variety in the number of wickets lost. Why? Remember, the second innings ends when either all wickets are lost or the target is reached. So, if a team successfully chases the target with, say, 7 wickets down, the innings ends there. They don't keep going just to see if they'd lose the last 3 wickets.

This variety tells us that there were both comfortable chases (where teams won with many wickets in hand) and nail-biting finishes (where teams just scraped through or lost all their wickets trying).

3.1.3. Venue hosted distribution



This chart shows us how many matches each stadium hosted. It's not just a fun fact - it can actually impact the tournament in subtle ways.

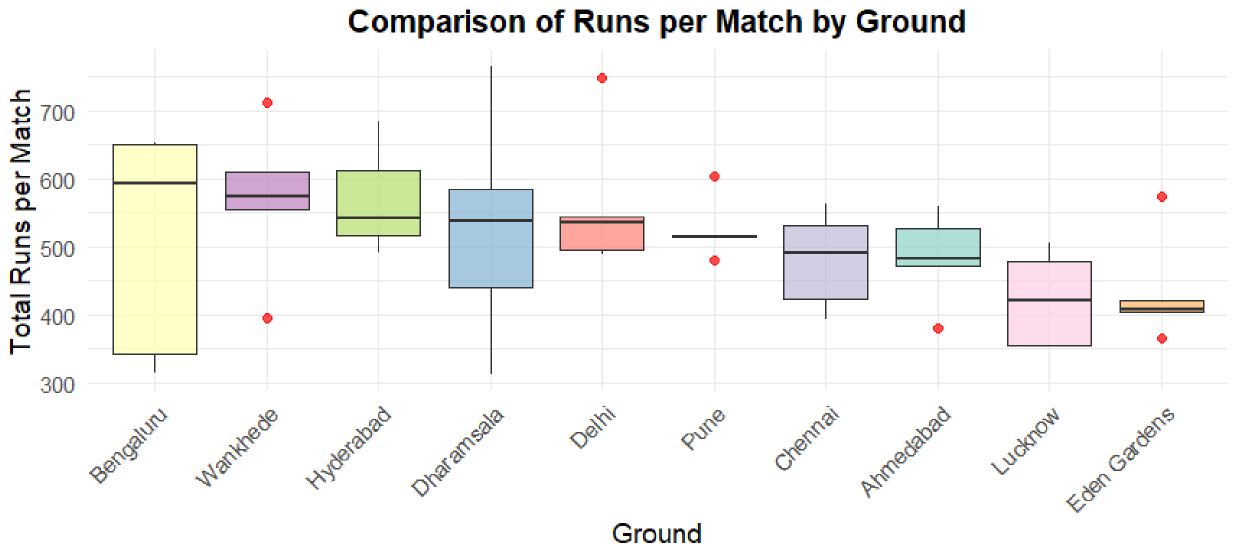
We can see that some stadiums, like Mumbai and Ahmedabad, hosted more matches than others. This could mean a few things:

1. Teams playing more often at these venues might have gotten more familiar with the conditions, potentially giving them a slight advantage.
2. Fans and analysts could get a better sense of how these grounds play (for example, if they tend to favor batsman or bowlers) because there's more data to work with.
3. The wear and tear on these pitches over multiple games could have affected how they played in later matches.

Stadiums that hosted fewer games, on the other hand, might have been a bit more unpredictable, as teams had less experience playing there during this tournament.

In conclusion, this tournament overview gives us a rich picture of how the Cricket World Cup 2023 played out. We see that batting first often led to higher scores, while chasing teams scored less but didn't always lose all their wickets. The distribution of matches across venues added another layer of complexity to the tournament, potentially influencing team strategies and performances. All these factors combined to create the exciting and unpredictable cricket that fans love to watch.

3.1.4. Runs per match in a ground



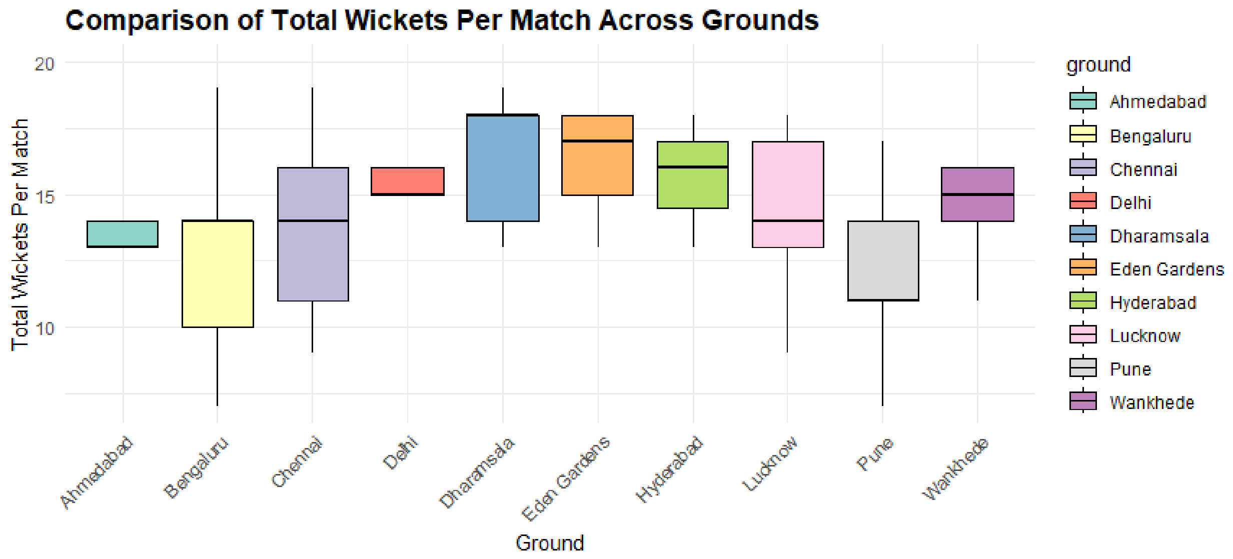
Let's explore how different venues in the Cricket World Cup 2023 influenced the game.

Our analysis shows that runs scored varied quite a bit depending on where the match was played. Some grounds, like Mumbai and Bengaluru, saw higher average scores per match. It's almost as if these grounds were saying, "Come on, batsmen, show us what you've got!" These high-scoring venues might have had flatter pitches (imagine a smoother, more predictable surface) or shorter boundaries, making it easier for batsmen to score runs.

On the flip side, grounds like Chennai and Delhi saw lower average scores. These venues might have been a bit tougher for batsmen, perhaps with pitches that helped the bowlers more. It's like these grounds were posing a challenge: "Let's see if you can score big here!"

This information is gold for teams and captains. Knowing a venue's scoring trend can influence decisions like whether to bat or bowl first if you win the toss, or how to pace an innings. For instance, what might be considered a good score in Chennai could be below par in Mumbai.

3.1.5. Wickets per match in a ground



Now, let's shift our focus from runs to wickets. This statistic tells us how "bowler-friendly" each ground was during the tournament.

Interestingly, we see less variation in wickets per match across grounds compared to runs per match. This suggests that while scoring rates differed significantly between venues, the rate at which wickets fell was more consistent.

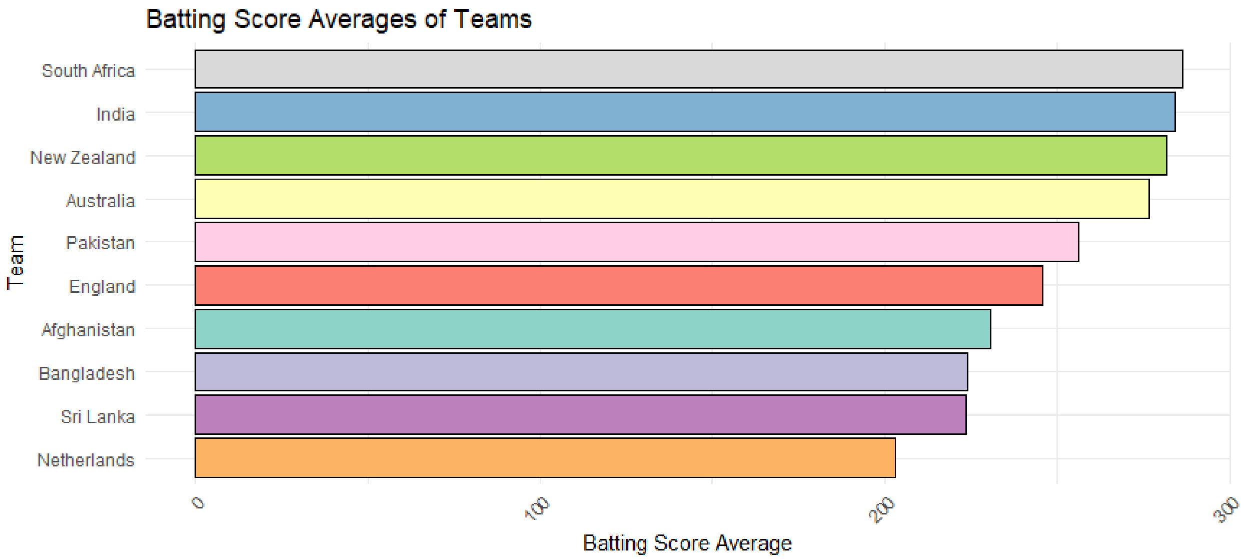
However, grounds like Dharamsala and Chennai show slightly higher wicket averages. This could mean a few things:

1. These grounds might have had pitches that offered more help to bowlers, making it easier to take wickets.
2. The conditions at these venues (like weather or altitude) might have favored bowlers more.
3. There could have been exciting, closely-fought matches at these grounds where teams were bowled out more often.

For teams, this information is crucial. If you know a ground tends to see more wickets fall, you might adopt a more cautious batting approach or pick an extra batsman in your lineup.

The combination of runs per match and wickets per match gives us a comprehensive picture of each ground's character. A ground with high runs and low wickets is a batsman's paradise, while one with low runs and high wickets is a bowler's dream. Most grounds fall somewhere in between, offering a balance that makes cricket so intriguing.

3.1.6. Batting Score average of a team



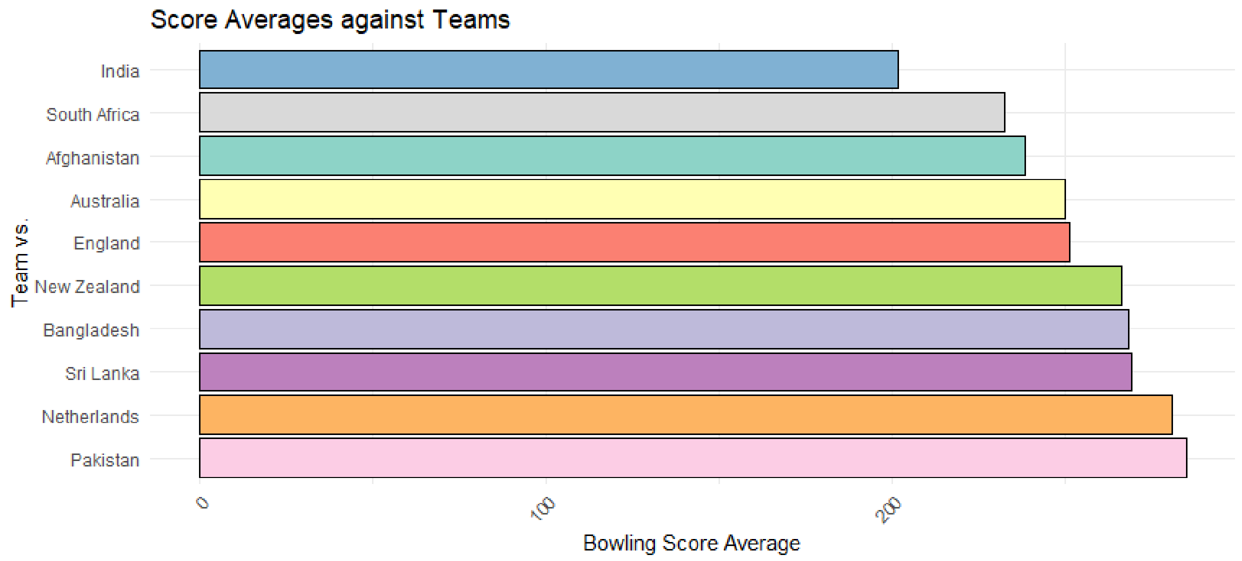
This statistic is like a team's batting report card for the tournament. It tells us, on average, how many runs each team scored when they batted.

Teams like India, South Africa, and England emerge as the high scorers, boasting higher batting averages. This high average indicates a strong, reliable batting lineup that could be counted on to put runs on the board match after match.

On the other hand, teams with lower averages might have struggled to score consistently. This could be due to facing tough bowling attacks, struggling to adapt to different conditions, or simply having a tournament where their batsmen couldn't find their rhythm.

For fans and analysts, this statistic helps identify which teams had the firepower to challenge for the title based on their batting strength alone.

3.1.7. Bowling Score average of a team

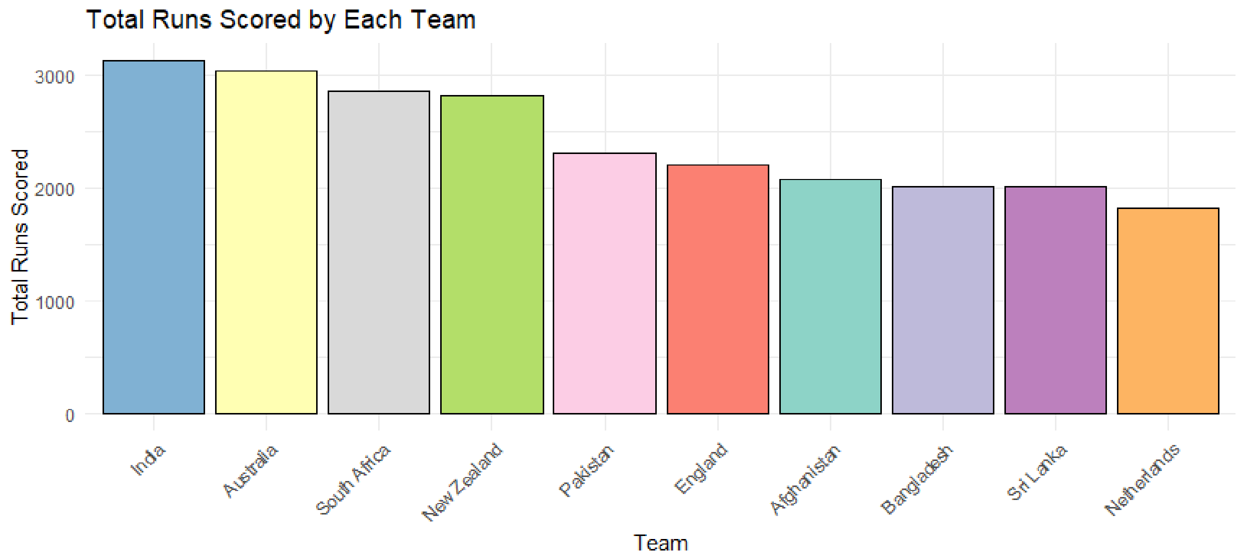


If the batting average was the offensive report card, this is the defensive one. It shows the average number of runs each team conceded when they were bowling.

Teams with lower bowling averages, such as India and South Africa, demonstrated superior bowling performances. It's like having a stellar defense in football that rarely allows the opposition to score. These teams were adept at containing the opposition's batsmen and keeping the scoring in check.

Higher averages might indicate bowling units that struggled to control the flow of runs. This could be due to facing strong batting lineups, having less effective bowling strategies, or perhaps dealing with injuries to key bowlers during the tournament.

3.1.8. Total runs scored by a team



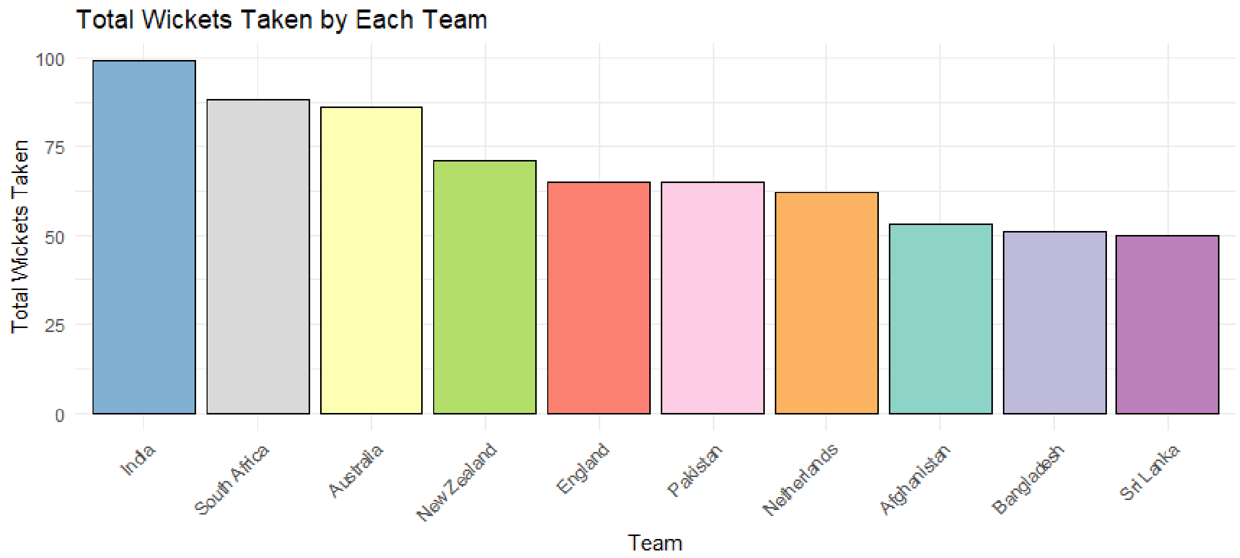
This statistic gives us the big picture of a team's batting performance throughout the tournament. It's not just about how well they batted, but also how far they progressed in the competition.

Teams at the top of this chart are there for two reasons:

1. They consistently scored well when batting.
2. They played more matches, likely because they advanced further in the tournament.

This is a combination of scoring average and games played. A high total here indicates both skill and success in the tournament.

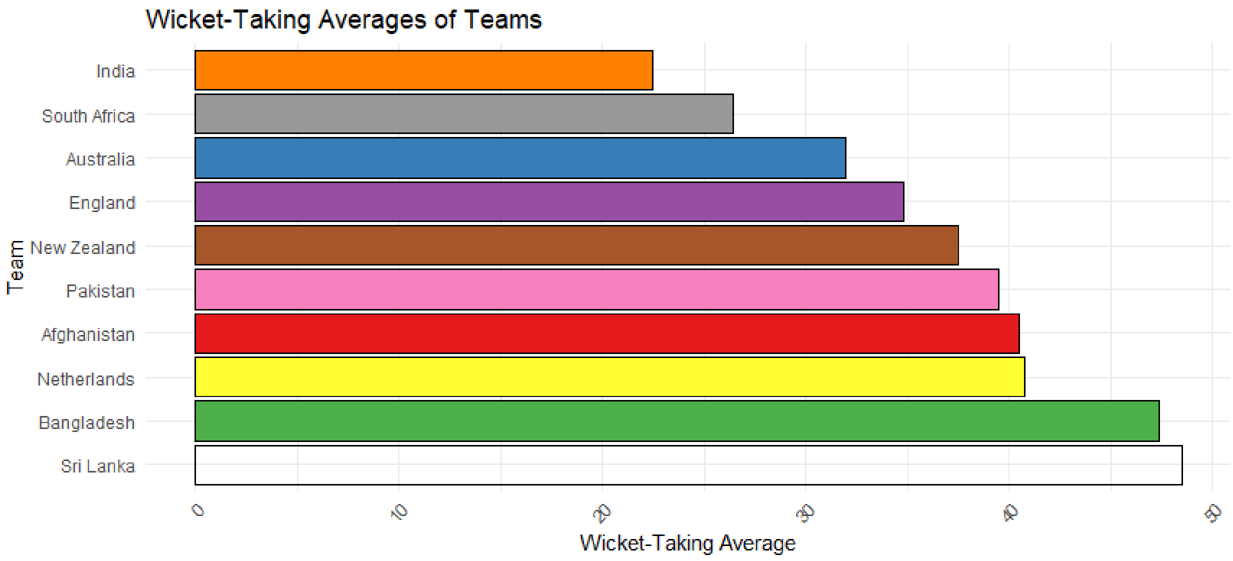
3.1.9. Total wickets taken by a team



Similar to total runs, this statistic reflects both bowling prowess and tournament progression. Teams with high wicket totals demonstrated effective bowling strategies and likely played more matches due to their success.

It shows which teams were most effective at dismissing opposition batsmen, a crucial skill in cricket.

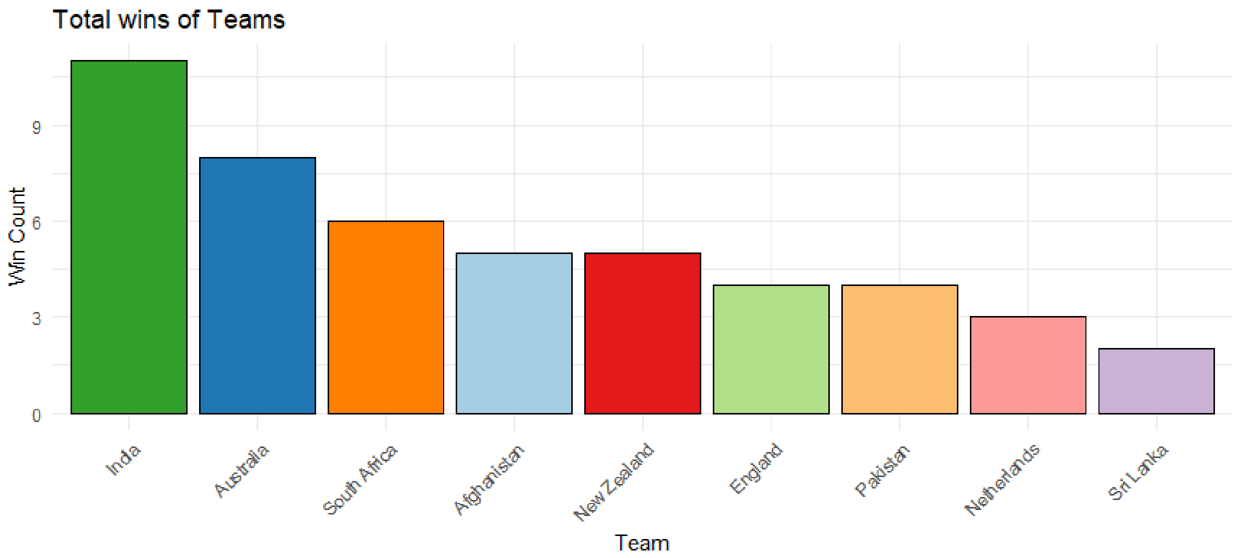
3.1.10. Wicket-taking average of a team



This statistic shows the average number of balls bowled per wicket taken, a key indicator of bowling efficiency.

Lower values here are better - they indicate more frequent wicket-taking. Teams like India and South Africa showing particularly strong performances in this area were able to dismiss opposition batsmen more quickly and efficiently.

3.1.11. Total wins by a team

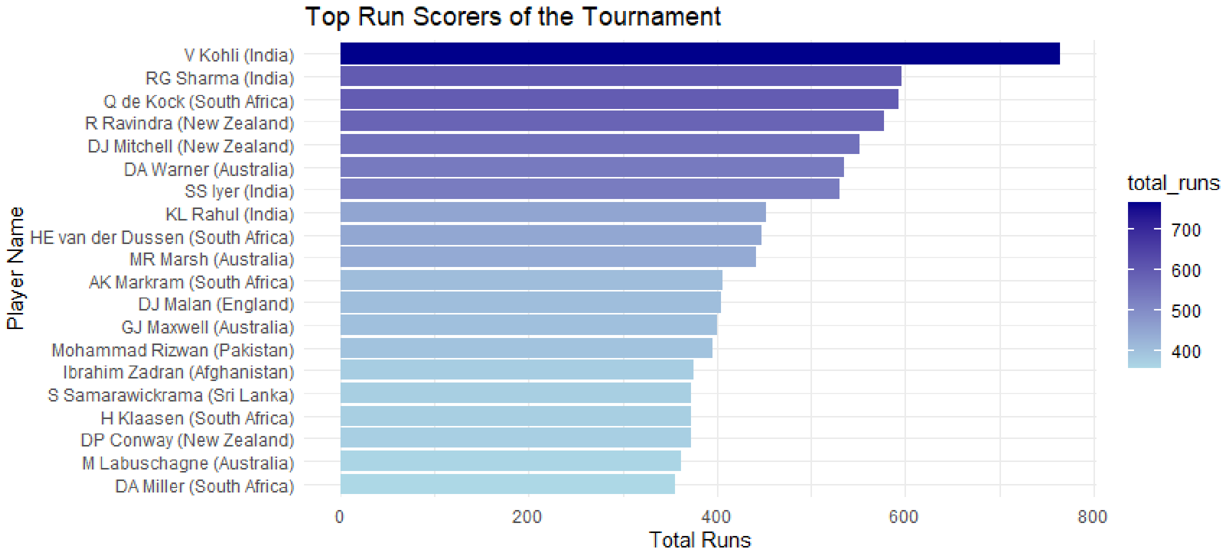


Finally, we come to the bottom line - wins. This distribution provides a clear picture of team success throughout the tournament.

Teams with the highest number of wins were the ones that progressed furthest in the competition. The top teams here likely reflect the semi-finalists and finalists. This statistic cuts through all the complexities of runs, wickets, and averages to show which teams were most successful at achieving cricket's ultimate goal: winning matches.

In conclusion, this evaluation of ground influence and team performance paints a vivid picture of the Cricket World Cup 2023. We see how different venues posed unique challenges, and how teams varied in their ability to bat, bowl, and ultimately win matches. This rich tapestry of statistics helps us understand the many factors that contributed to the excitement and outcomes of the tournament.

3.1.12. Top run scorers



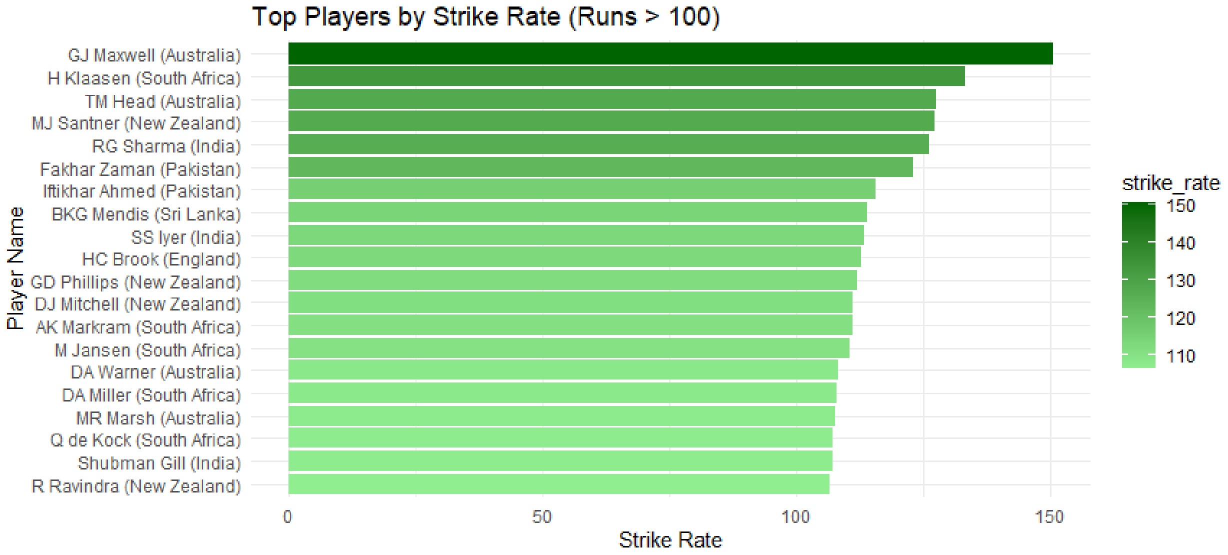
This category is all about the batting superstars of the tournament - the players who consistently put runs on the board for their teams.

1. V Kohli - 765
2. RG Sharma - 597
3. Q de Kock - 594
4. R Ravindra - 578
5. DJ Mitchell - 552

Leading this list is ‘V Kohli’ with 765 runs. To put this into perspective, in a tournament where team scores often ranged between 250-300, this player alone contributed an average of 95.625 runs per game.

What makes this achievement even more impressive is the pressure of performing on the world stage. These batsmen didn't just score runs; they did so against the best bowling attacks in the world, often in high-pressure situations where their team's success hung in the balance.

3.1.13. Players with top strike rates



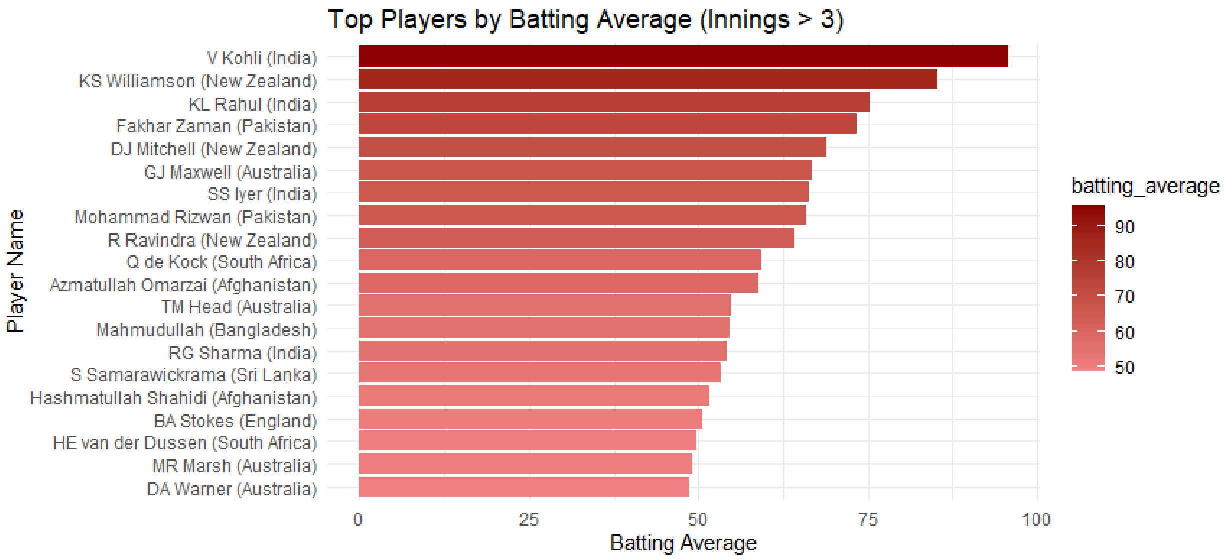
Strike rate in cricket is about how quickly a batsman scores runs. It's calculated as the number of runs scored per 100 balls faced. A high strike rate means a player is scoring quickly, putting pressure on the opposing team.

1. GJ Maxwell -150.38
2. H Klaasen -133.21
3. Tm Head -127.52
4. MJ Santner -125.95
5. RG Sharma -122.90

Topping this list is GJ Maxwell with a strike rate of 150.38. To illustrate what this means: if this player faced 100 balls (about 16-17 overs of cricket), they would be expected to score 150 runs.

Players with high strike rates are often game-changers. They can shift the momentum of a match rapidly, turning a good score into a great one or salvaging a challenging run chase.

3.1.14. Players with top batting averages



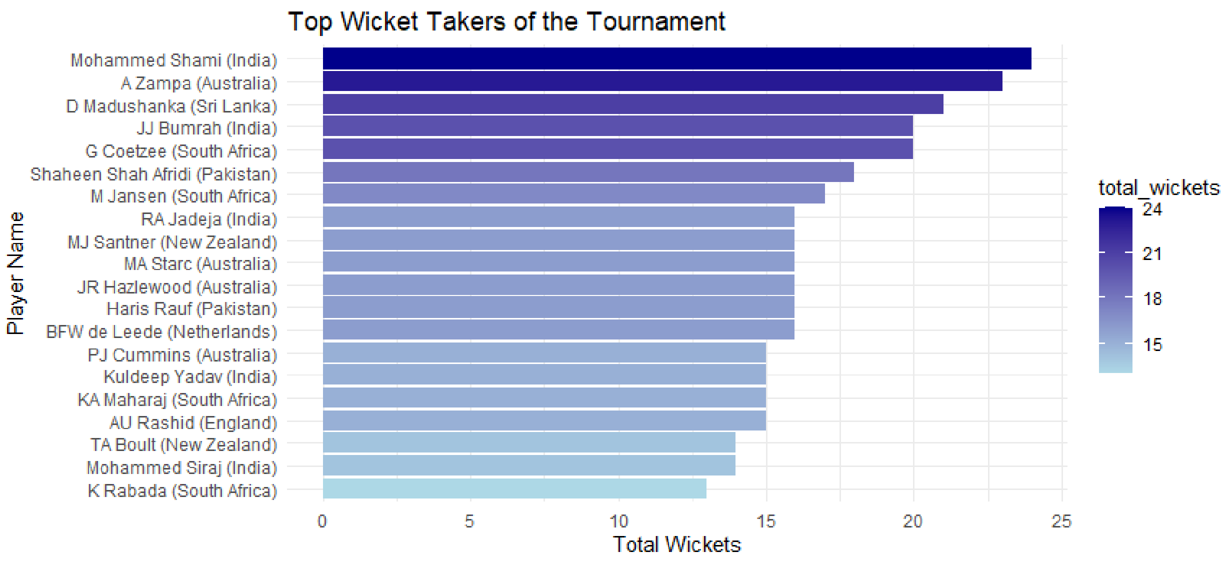
Batting average tells us how many runs, on average, a player scores per innings before getting out. It's a measure of both scoring ability and consistency.

1.	V Kohli	-	95.63
2.	KS Williamson	-	85.34
3.	KL Rahul	-	75.34
4.	Fakhar Zaman	-	73.34
5.	DJ Mitchel	-	69

V Kohli leads this category with an average of 95.63. This means that in any given inning, this player was likely to score around 95 runs before getting out. It's similar to a tennis player who consistently reaches the later rounds of every tournament they enter.

A high batting average is the hallmark of a reliable player - someone the team can count on to perform well match after match. These players form the backbone of their team's batting lineup.

3.1.15. Top wicket-takers



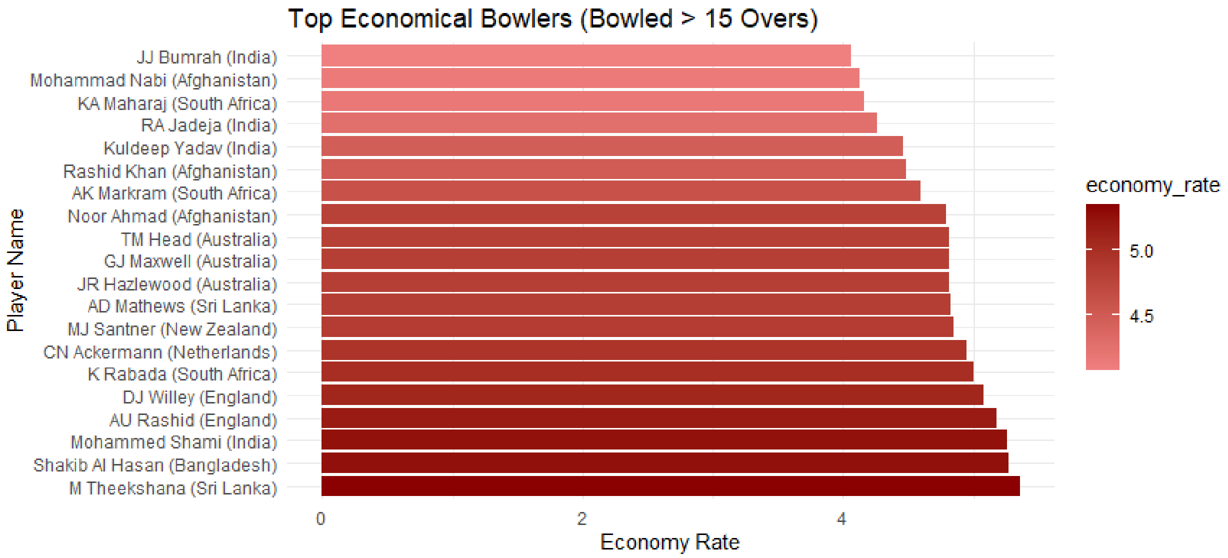
Now we move to the bowlers - the players tasked with dismissing the opposition's batsmen. Wickets in cricket are analogous to strikeouts in baseball, but often harder to achieve.

1.	Mohammed Shami	- 24
2.	A Zampa	- 23
3.	D Madushanka	- 21
4.	JJ Bumrah	- 20
5.	G Coetzee	- 20

Mohammed Shami stands at the top with 24 wickets. In a tournament where teams have 10 wickets per innings, taking 24 wickets is a remarkable feat. It means this player was responsible for dismissing, on average, 3 batsmen per match.

Top wicket-takers are match-winners. Their ability to regularly dismiss opposition batsmen not only restricts scoring but can also demoralize the opposing team.

3.1.16. Most economical bowlers



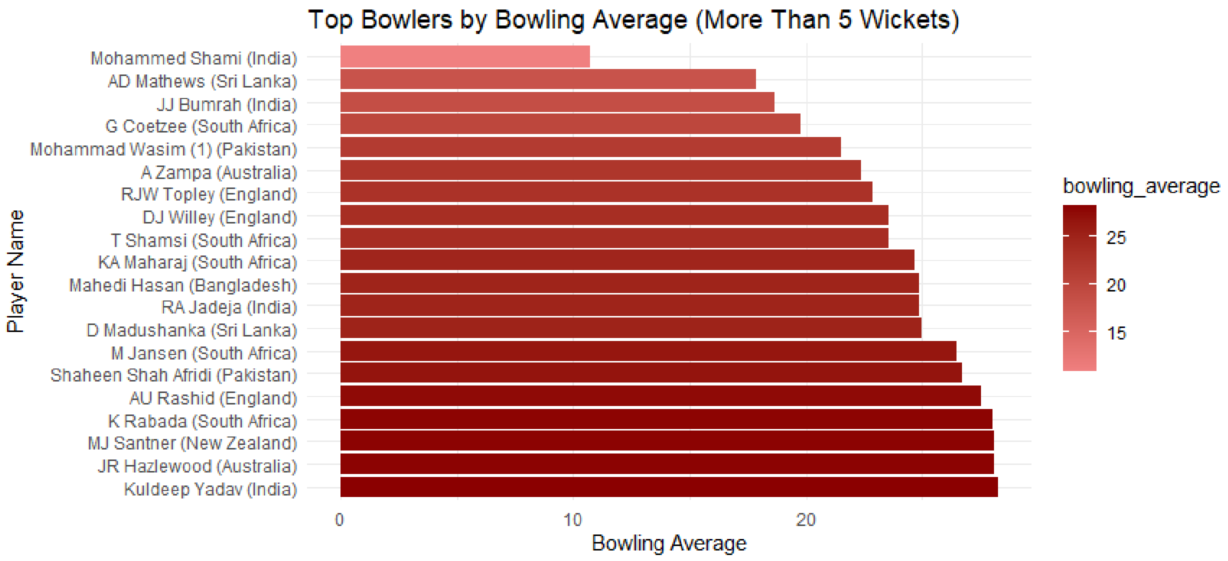
Economy rate in bowling is similar to ERA (Earned Run Average) in baseball. It measures how many runs a bowler concedes per over (6 balls). A lower economy rate is better, indicating that the bowler is hard to score against.

1.	JJ Bumrah	-	4.06
2.	Mohammad Nabi	-	4.13
3.	KA Maharaj	-	4.15
4.	RA Jadeja	-	4.25
5.	Kuldeep Yadav	-	4.45

JJ Bumrah leads this list with an economy rate of 4.06. This means they conceded only 4 runs per over on average. In a game where teams often score at 5-6 runs per over, keeping the rate down to 4.06 is exceptional.

Economical bowlers are crucial in cricket, especially in tight matches. They put pressure on the batting team by restricting scoring, often forcing batsmen to take risks against other bowlers.

3.1.17. Players with lowest bowling averages



Bowling average is calculated as the number of runs conceded per wicket taken. It combines the ability to take wickets with the ability to restrict runs. A lower bowling average is better.

1.	Mohammed Shami	-10.71
2.	AD Mathews	-17.84
3.	JJ Bumrah	-18.65
4.	G Coetzee	-19.8
5.	Mohammad Wasim	-21.5

Mohammed Shammi tops this category with a bowling average of 10.71. This means they conceded only 10 runs for each wicket they took.

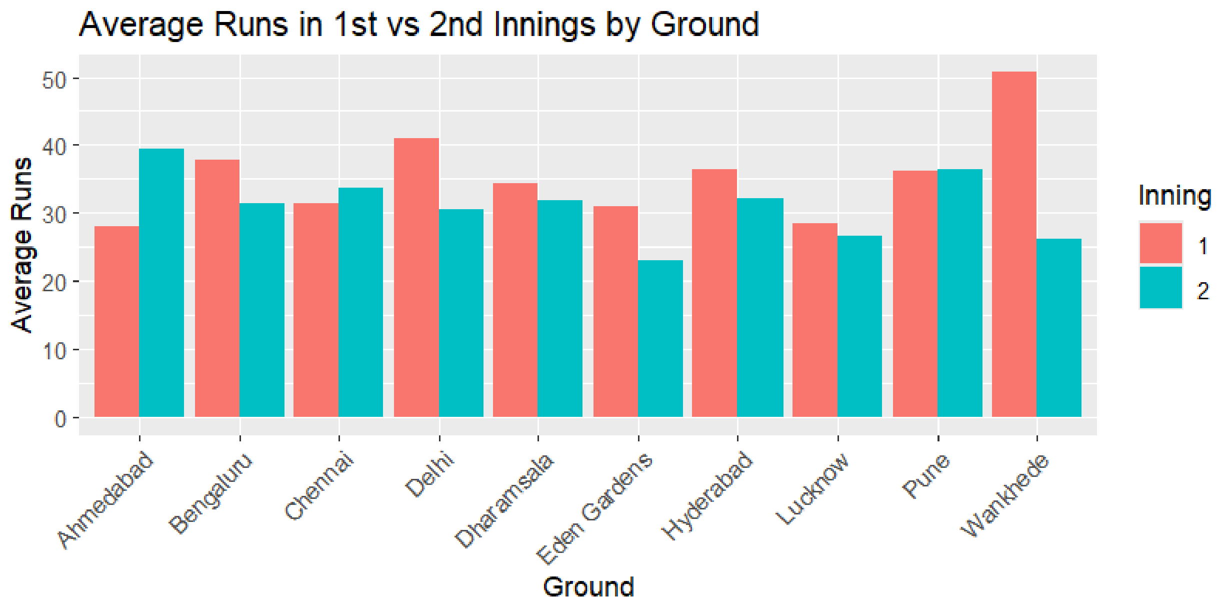
A low bowling average indicates a bowler who is both penetrative (able to take wickets) and economical (doesn't concede many runs). These bowlers are often the go-to options for captains in crucial situations.

In conclusion, these individual performance metrics give us a comprehensive view of the standout players in the Cricket World Cup 2023. From the batsmen who piled on the runs to the bowlers who kept batsmen in check and took crucial wickets, these statistics highlight the individual brilliance that collectively made the tournament a spectacle.

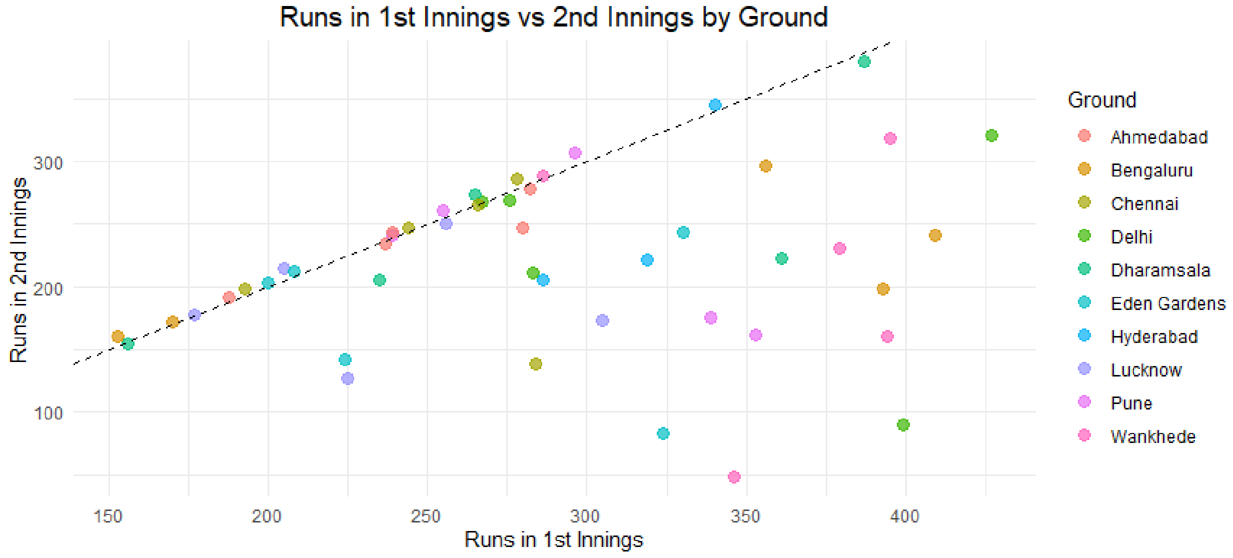
It's important to remember that cricket is ultimately a team sport, and these individual performances contribute to the larger goal of team success. The players topping these lists were likely instrumental in their teams' journeys through the tournament, turning matches with their exceptional skills and contributing significantly to their teams' victories.

3.2. Bivariate Analysis

3.2.1. Runs per 1st inning vs 2nd inning in a ground



Imagine each cricket ground as a unique stage, each with its own character. Some stages might favor the opening act (1st inning), while others might be kinder to the closing performance (2nd inning). Let's see how this played out in our tournament.



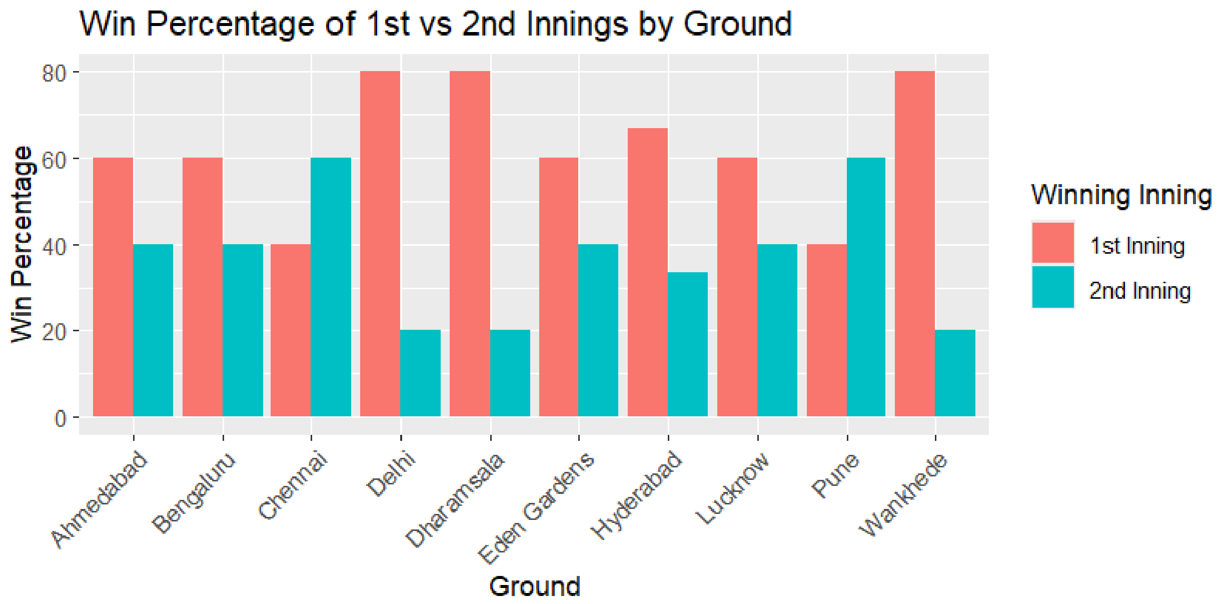
In this chart, each dot represents a ground. If a dot is above the diagonal line, it means more runs were typically scored in the 1st inning at that ground. If it's below, the 2nd inning was generally higher scoring.

Key observations:

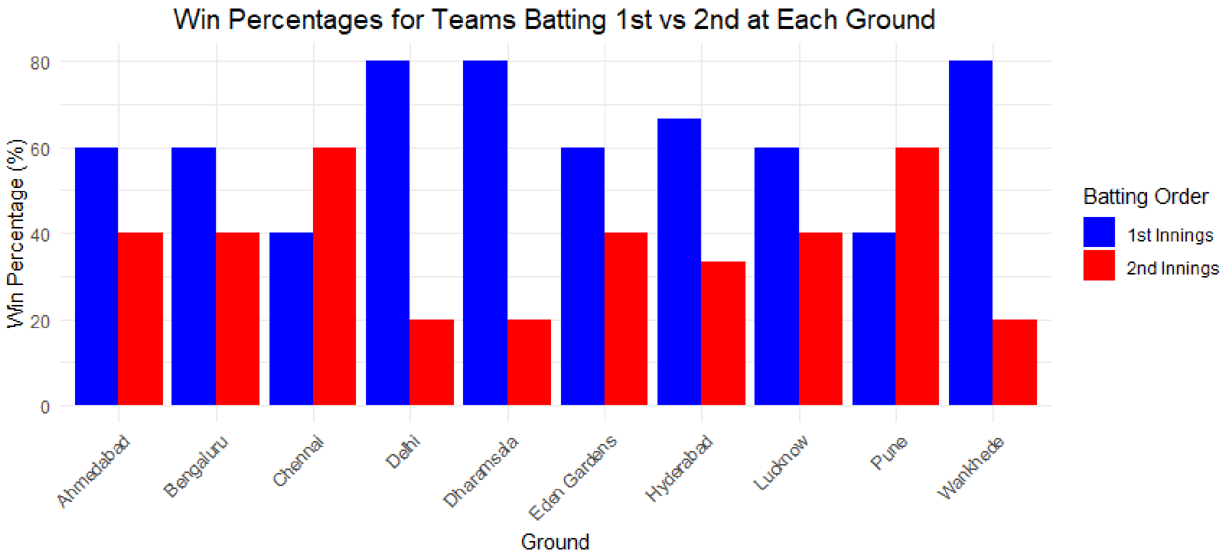
1. Grounds above the line (e.g., [Ground Delhi], [Ground Pune]): These are "set the target" grounds. Teams batting first here often scored more.
2. Grounds below the line (e.g., [Ground Chennai], [Ground Hyderabad]): These are "chaser's paradises." Teams batting second typically scored more here.
3. Grounds near the line (e.g., [Ground Lucknow], [Ground Wankhede]): These are balanced grounds, showing little difference between 1st and 2nd innings scores.

This information is crucial for team strategies. If you're playing at a "set the target" ground, winning the toss and batting first might be advantageous. At a "chaser's paradise," you might prefer to bowl first and then chase the target.

3.2.2. Win percentage of 1st team vs 2nd team in a ground



Now, let's see if batting first or second actually translated into more wins at different grounds.



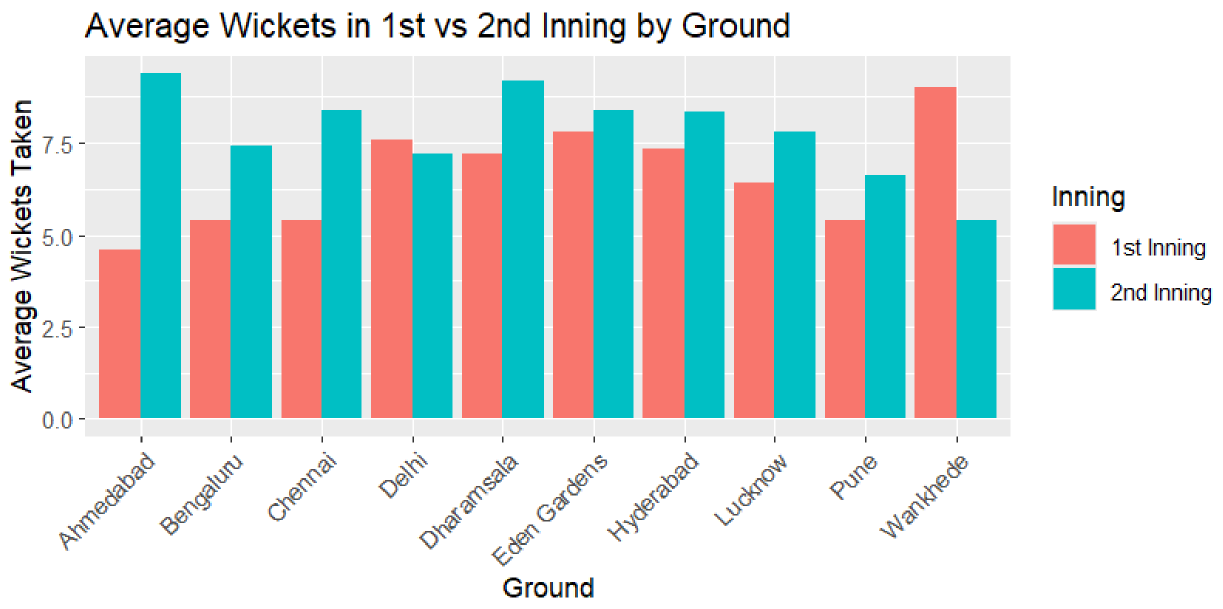
This chart shows us whether teams batting first or second had a higher win percentage at each ground. It's like checking if the home team in basketball wins more often when they start with the ball or when the visitors do.

Key insights:

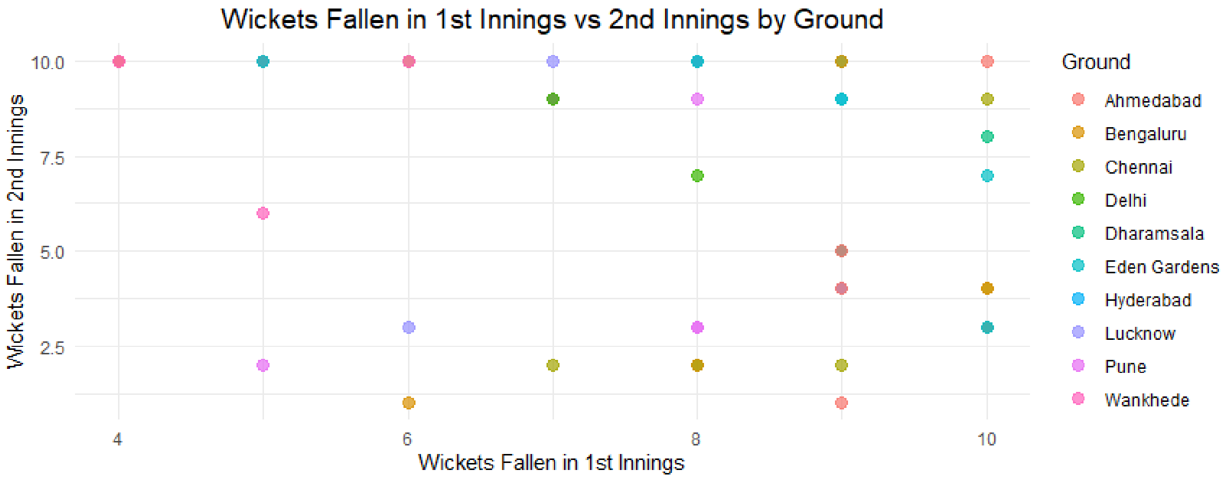
1. Grounds favoring 1st innings (e.g., [Ground Delhi], [Ground Dharamsala]): Here, teams batting first won more often. These grounds might have pitches that deteriorate over time, making batting harder in the 2nd innings.
2. Grounds favoring 2nd innings (e.g., [Ground Chennai], [Ground Pune): Teams chasing targets were more successful here. These grounds might have dew in the evening that makes the ball harder to grip for bowlers in the 2nd innings.
3. Balanced grounds (e.g., [Ground Eden Gardens]): These showed little difference in win percentages, suggesting other factors (like team strength) were more important than batting order.

This data is gold for captains at the toss. Knowing these percentages could influence their decision to bat or bowl first, potentially giving their team a statistical edge before the first ball is bowled.

3.2.3. Wickets per 1st inning vs 2nd inning in a ground



Lastly for grounds, let's look at how wicket-taking varied between innings.



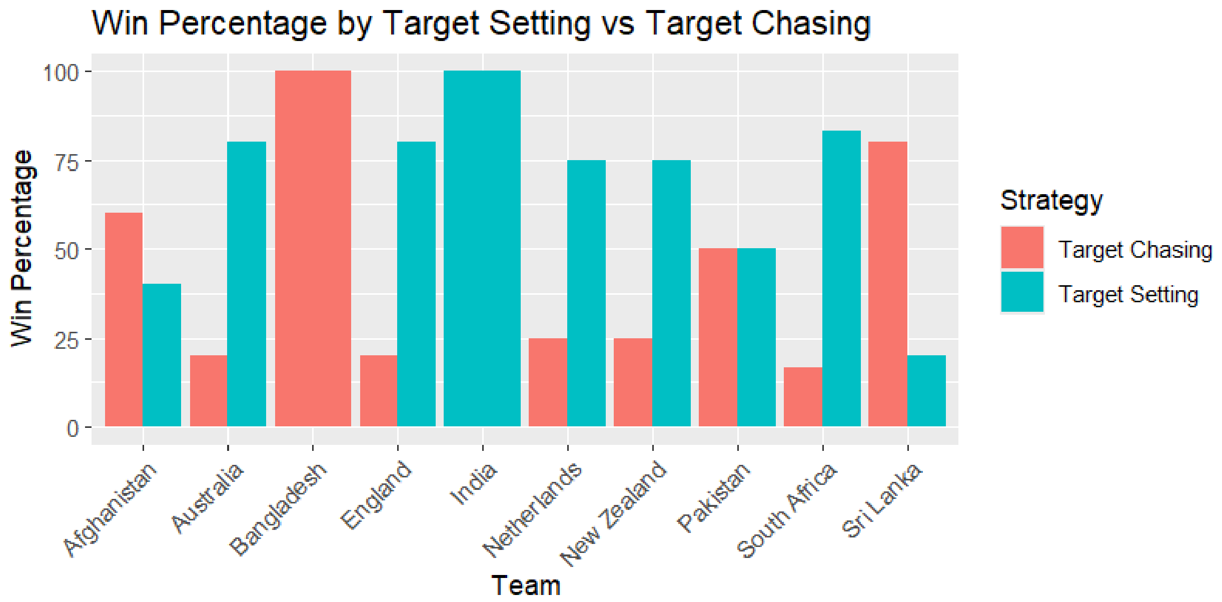
Similar to our runs analysis, dots above the line indicate more wickets fell in the 1st innings, while dots below show more wickets in the 2nd innings.

Key observations:

1. Grounds above the line (e.g., [Ground Wankhede]): More wickets fell in the 1st innings here. This could indicate pitches that start lively but become easier to bat on, like a tennis court that plays faster in the morning.
2. Grounds below the line (e.g., [Ground Ahmedabad], [Ground Bengaluru]): These saw more wickets in the 2nd innings. Perhaps these grounds have pitches that deteriorate, becoming harder to bat on as the game progresses.
3. Grounds near the line (e.g., [Ground Delhi]): These grounds showed consistent wicket-taking across both innings.

This information helps teams decide on their batting approach. On a ground where more wickets fall in the 2nd innings, a team batting first might aim for a slightly lower but safer score, knowing it'll be harder for the opposition to chase.

3.2.4. Win percentage of a team by target setting or target chasing



Finally, let's look at how each team performed when setting targets versus chasing them.

This chart tells us whether a team was more successful when they batted first (set a target) or second (chased a target). It's like seeing if a tennis player performs better serving first or receiving.

Key insights:

1. Strong target-setters (e.g., [Team Australia], [Team South Africa]): These teams won more often when batting first. They might have strong batting lineups that thrive without the pressure of a target, or excellent bowlers who defend totals well.
2. Skilled chasers (e.g., [Team Bangladesh], [Team Sri Lanka]): These teams performed better when chasing targets. They might have batsmen who excel under pressure or struggle to pace an innings when batting first.
3. Balanced teams (e.g., [Team Pakistan]): Some teams showed similar win percentages in both scenarios, indicating versatility in their play.

This data is crucial for understanding team strengths and weaknesses. A team that's strong at chasing might be more inclined to bowl first if they win the toss, playing to their strengths.

4. Conclusion

The analysis of the ICC Cricket World Cup 2023 dataset has revealed several key insights that provide a deep understanding of the tournament dynamics, team performances, and the various factors influencing match outcomes. Here are the main conclusions drawn from our analysis:

1. Scoring Patterns:

Our analysis revealed a clear preference for setting higher targets when batting first, with scores between 250-300 runs being most common in the first innings. This suggests teams prioritized posting challenging totals when given the opportunity to bat first.

2. Wicket Distributions:

Complete team dismissals were frequent, especially in the first innings. This indicates that bowling teams often managed to maintain pressure throughout the innings, and batting teams were willing to take risks to maximize their scores.

3. Venue Influence:

The characteristics of different grounds significantly impacted match outcomes. Some venues clearly favored batting, while others were more conducive to bowling. This emphasizes the importance of teams adapting their strategies to suit specific ground conditions.

4. Team Performances:

There was considerable variation in team performances, with certain teams consistently excelling in both batting and bowling metrics. This highlights the importance of all-round team strength in achieving success in the tournament.

5. Toss Decisions:

The choice to bat or bowl first had varying levels of impact depending on the venue. This suggests that toss decisions played a crucial role in team strategies and potentially in match outcomes.

6. Individual Brilliance:

Several players stood out with exceptional performances in batting and bowling, demonstrating the impact individual brilliance can have on match and tournament results.

7. Strategic Adaptability:

Teams that successfully adapted their approach based on venue characteristics and their own strengths in setting or chasing targets generally performed better throughout the tournament.

These findings have significant implications for future cricket strategies, team selections, and even tournament organization. The analysis underscores the complexity of cricket as a sport, where multiple factors including individual performances, team strategies, external conditions, and even luck (in terms of winning the toss) all play crucial roles in determining match outcomes.

The insights gained from this analysis can be valuable for teams in preparing for future tournaments, helping them to:

- Develop more nuanced strategies based on venue characteristics
- Make informed decisions when winning the toss
- Balance their team composition to excel in both batting and bowling
- Identify and nurture players who perform well under specific conditions

For tournament organizers, this analysis highlights the importance of selecting a diverse range of venues to ensure a balanced and exciting competition.

Future research could focus on more granular analysis of player performances, the impact of specific bowling or batting partnerships, how teams adapted their strategies throughout the tournament, and the influence of factors not captured in this dataset such as weather conditions or player fitness levels.

In conclusion, the Cricket World Cup 2023 provided a rich tapestry of data that, when analyzed, reveals the intricate nature of the sport and the myriad factors that contribute to success at the highest level of international cricket. This analysis not only helps in understanding the dynamics of this specific tournament but also provides valuable insights for the future of the sport.

5. References

1. **Kaggle Dataset**: JD Australia. (2023). *ICC CWC23 All Innings Cleaned*. Kaggle. https://www.kaggle.com/datasets/jdaustralia/icc-cwc23-all-innings-cleaned

2. **Correlation Test Article**: sthda.com. (n.d.). *Correlation Test Between Two Variables in R*. STHDA. http://www.sthda.com/english/wiki/correlation-test-between-two-variables-in-r

6. Appendices

```
######################## R Script for CWC-2023-analysis ############################

library(readr)
library(dplyr)
library(ggplot2)
library(RColorBrewer)
library(tidyr)

data <- read_csv("./CWC23_all_innings.csv")

data$start_date <- as.Date(data$start_date, format = '%d-%b-%y')  # Convert date column to Date objects
data <- data[order(data$start_date), ]  # Sort data by date

# Separate batting and bowling data
batting_data <- data %>% filter(bat_or_bowl == "bat")
bowling_data <- data %>% filter(bat_or_bowl == "bowl")

# Create match results summarizing table
inning_summary <- bowling_data %>%
  group_by(team, opposition, ground, start_date, inns) %>%
  summarise(
    total_runs_scored = sum(runs, na.rm = TRUE),
    total_wickets_taken = sum(wkts, na.rm = TRUE)
  ) %>%
  ungroup() %>%
  arrange(start_date)

# Separate data for each inning into first and second innings
first_innings <- inning_summary %>%
  filter(inns == 1) %>%
  rename(
    team2 = team,
    team1_runs = total_runs_scored,
    team1_wickets = total_wickets_taken
  )

second_innings <- inning_summary %>%
  filter(inns == 2) %>%
  rename(
    team1 = team,
    team2_runs = total_runs_scored,
    team2_wickets = total_wickets_taken
  )

# Merge first and second innings data into a single record for each match
merged_matches <- first_innings %>%
  inner_join(second_innings, by = c("opposition" = "team1", "team2" = "opposition", "ground",
"start_date")) %>%
  select(opposition, team2, ground, start_date, team1_runs, team2_runs, team1_wickets,
team2_wickets)

# Rename columns for clarity
names(merged_matches) <- c("Team_1", "Team_2", "ground", "start_date", "Team1_total",
"Team2_total", "Team1_wickets_fallen", "Team2_wickets_fallen")

# Add 'match_total_runs' field to indicate the sum of both teams' runs
merged_matches <- merged_matches %>%
  mutate(
    match_total_runs = Team1_total + Team2_total,
    match_total_wickets = Team1_wickets_fallen + Team2_wickets_fallen
  )

# Add a 'winner' field to indicate the winning team
merged_matches <- merged_matches %>%
  mutate(winner = case_when(
    Team1_total > Team2_total ~ Team_1,
    Team2_total > Team1_total ~ Team_2,
    TRUE ~ "Tie"
  ))

# Determine if the winning team batted first or second
merged_matches <- merged_matches %>%
  mutate(
    batting_first_winner = case_when(
      winner == Team_1 ~ "1st Innings",
      winner == Team_2 ~ "2nd Innings",
      TRUE ~ NA_character_
    )
  )

# Calculate win percentages for teams batting 1st vs. 2nd at each ground
```

```
win_summary <- merged_matches %>%
  filter(!is.na(batting_first_winner)) %>%
  group_by(ground, batting_first_winner) %>%
  summarise(
    wins = n(),
    total_matches = n_distinct(start_date)
  ) %>%
  mutate(
    win_percentage = (wins / sum(wins)) * 100
  ) %>%
  ungroup()

# Calculate win percentages by ground and inning
win_summary <- merged_matches %>%
  mutate(winning_inning = case_when(
    winner == Team_1 ~ "1st Inning",
    winner == Team_2 ~ "2nd Inning",
    TRUE ~ "Tie"
  )) %>%
  group_by(ground, winning_inning) %>%
  summarise(win_count = n()) %>%
  ungroup() %>%
  group_by(ground) %>%
  mutate(win_percentage = (win_count / sum(win_count)) * 100)

# Filter out any "Tie" matches if you don't want them in the plot
win_summary <- win_summary %>%
  filter(winning_inning != "Tie")

# Create columns to identify teams setting the target and chasing
win_summary <- merged_matches %>%
  mutate(
    setting_target = Team_1,
    chasing_target = Team_2
  ) %>%
  mutate(winning_strategy = case_when(
    winner == setting_target ~ "Target Setting",
    winner == chasing_target ~ "Target Chasing",
    TRUE ~ "Tie"
  ))

# Calculate the win percentage by strategy (target setting or chasing) for each team
win_percentage <- win_summary %>%
  group_by(Team_1, winning_strategy) %>%
  summarise(win_count = n()) %>%
  ungroup() %>%
  group_by(Team_1) %>%
  mutate(win_percentage = (win_count / sum(win_count)) * 100)

# Filter out any "Tie" matches if you don't want them in the plot
win_percentage <- win_percentage %>%
  filter(winning_strategy != "Tie")

# Aggregate bowling performance data for each team
batting_team_perfomance <- inning_summary %>%
  group_by(opposition) %>%
  summarise(
    total_batting_runs = sum(total_runs_scored, na.rm = TRUE),
    matches_played_batting = n_distinct(start_date)
  )
batting_team_perfomance <- batting_team_perfomance %>% rename(team = opposition)

bowling_team_perfomance <- inning_summary %>%
  group_by(team) %>%
  summarise(
    total_runs_against = sum(total_runs_scored, na.rm = TRUE),
    total_wickets_taken = sum(total_wickets_taken, na.rm = TRUE),
    matches_played_bowling = n_distinct(start_date)
  )

# Merge batting and bowling summaries to create a complete team performance table
team_performance <- batting_team_perfomance %>%
  left_join(bowling_team_perfomance, by = "team") %>%
  mutate(
    matches_played = coalesce(matches_played_batting, matches_played_bowling),
    batting_score_average = total_batting_runs / matches_played,
    bowling_score_average = total_runs_against / matches_played,
    wicket_taking_average = total_runs_against / total_wickets_taken
  ) %>%
  select(
    team,
    total_batting_runs,
    total_runs_against,
    total_wickets_taken,
    batting_score_average,
    bowling_score_average,
    wicket_taking_average,
    matches_played
  )

# Calculate total wins for each team
team_wins_count <- merged_matches %>%
  group_by(winner) %>%
  summarise(
    win_count = n_distinct(start_date)
  )

# Count the number of matches hosted by each ground
ground_counts <- merged_matches %>%
  count(ground)

# Calculate batting score average for each team
batting_avg <- data %>%
  group_by(team) %>%
  summarise(batting_score_avg = mean(runs, na.rm = TRUE))

# Calculate bowling score average for each team
bowling_avg <- data %>%
  filter(bat_or_bowl == "bowl") %>%
  group_by(team) %>%
  summarise(bowling_score_avg = mean(runs, na.rm = TRUE))

# Calculate average runs by ground and inning
runs_summary <- data %>%
  group_by(ground, inns) %>%
  summarise(avg_runs = mean(runs, na.rm = TRUE))

# Calculate average wickets by ground and inning
wickets_summary <- merged_matches %>%
  select(ground, Team1_wickets_fallen, Team2_wickets_fallen) %>%
  mutate(
    first_inning_wickets = Team2_wickets_fallen,
    second_inning_wickets = Team1_wickets_fallen
  ) %>%
  group_by(ground) %>%
  summarise(
    avg_first_inning_wickets = mean(first_inning_wickets, na.rm = TRUE),
    avg_second_inning_wickets = mean(second_inning_wickets, na.rm = TRUE)
  )

# Reshape wickets summary data for plotting
wickets_summary_long <- wickets_summary %>%
  pivot_longer(cols = c("avg_first_inning_wickets", "avg_second_inning_wickets"),
               names_to = "inning", values_to = "avg_wickets") %>%
  mutate(inning = case_when(
    inning == "avg_first_inning_wickets" ~ "1st Inning",
    inning == "avg_second_inning_wickets" ~ "2nd Inning"
  ))

# Calculate player contribution by runs
player_runs_contribution <- bowling_data %>%
  group_by(player, team) %>%
  summarise(
    total_runs = sum(runs, na.rm = TRUE),
    total_wickets = sum(wkts, na.rm = TRUE)
  ) %>%
  ungroup()

# Reshape player contribution data for plotting
player_contribution_long <- player_runs_contribution %>%
  pivot_longer(cols = c("total_runs", "total_wickets"),
               names_to = "contribution_type", values_to = "total_contribution") %>%
  mutate(contribution_type = case_when(
    contribution_type == "total_runs" ~ "Runs",
    contribution_type == "total_wickets" ~ "W ickets"
  ))

# Calculate top run scorers
top_run_scorers <- batting_data %>%
  group_by(player) %>%
  summarise(total_runs = sum(runs, na.rm = TRUE)) %>%
  arrange(desc(total_runs)) %>%
  slice_max(total_runs, n = 20)

# Calculate top strike rates
strike_rate_data <- batting_data %>%
  group_by(player) %>%
  summarise(
    total_runs = sum(runs, na.rm = TRUE),
    total_balls_faced = sum(bb_bf, na.rm = TRUE)
  ) %>%
  filter(total_runs > 100) %>%
  mutate(strike_rate = (total_runs / total_balls_faced) * 100) %>%
  arrange(desc(strike_rate)) %>%
  slice_max(strike_rate, n = 20)
```

```
# Calculate top batting averages
batting_average_data <- batting_data %>%
  group_by(player) %>%
  summarise(
    total_runs = sum(runs, na.rm = TRUE),
    innings_played = n(),
    not_outs = sum(not_out, na.rm = TRUE)
  ) %>%
  filter(innings_played > 3) %>%
  mutate(batting_average = total_runs / (innings_played - not_outs)) %>%
  arrange(desc(batting_average)) %>%
  slice_max(batting_average, n = 20)

# Calculate top wicket takers
wicket_takers_data <- bowling_data %>%
  group_by(player) %>%
  summarise(
    total_wickets = sum(wkts, na.rm = TRUE)
  ) %>%
  arrange(desc(total_wickets)) %>%
  slice_max(total_wickets, n = 20)

# Calculate top economical bowlers
economical_bowlers_data <- bowling_data %>%
  group_by(player) %>%
  summarise(
    total_runs_conceded = sum(runs, na.rm = TRUE),
    total_balls_bowled = sum(bb_bf, na.rm = TRUE),
    total_overs_bowled = total_balls_bowled / 6
  ) %>%
  filter(total_overs_bowled > 15) %>%
  mutate(economy_rate = total_runs_conceded / total_overs_bowled) %>%
  arrange(economy_rate) %>%
  slice_min(economy_rate, n = 20)

# Calculate top bowling averages
bowling_average_data <- bowling_data %>%
  group_by(player) %>%
  summarise(
    total_runs_conceded = sum(runs, na.rm = TRUE),
    total_wickets = sum(wkts, na.rm = TRUE)
  ) %>%
  filter(total_wickets > 5) %>%
  mutate(bowling_average = total_runs_conceded / total_wickets) %>%
  arrange(bowling_average) %>%
  slice_min(bowling_average, n = 20)

# Create plots
# ...

# Create a histogram of innings total runs to show the distribution
ggplot(inning_summary, aes(x = total_runs_scored)) +
  geom_histogram(binwidth = 20, fill = "steelblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Innings Total Runs in the Tournament",
       x = "Innings Total Runs",
       y = "Frequency") +
  theme_minimal()

# Summary statistics for innings total runs
summary(inning_summary$total_runs_scored)

# Create a histogram of 1st-innings total runs to show the distribution
ggplot(first_innings, aes(x = team1_runs)) +
  geom_histogram(binwidth = 20, fill = "steelblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Innings Total Runs in the Tournament",
       x = "Innings Total Runs",
       y = "Frequency") +
  theme_minimal()

# Summary statistics for 1st-innings total runs
summary(first_innings$team1_runs)

# Create a histogram of 2nd-innings total runs to show the distribution
ggplot(second_innings, aes(x = team2_runs)) +
  geom_histogram(binwidth = 20, fill = "steelblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Innings Total Runs in the Tournament",
       x = "Innings Total Runs",
       y = "Frequency") +
  theme_minimal()

# Summary statistics for 2nd-innings total runs
summary(second_innings$team2_runs)
#-------------------------

# Create a histogram of innings total wickets to show the distribution
ggplot(inning_summary, aes(x = total_wickets_taken)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Innings Total Wickets in the Tournament",
       x = "Innings Total Wickets",
       y = "Frequency") +
  theme_minimal()

# Summary statistics for innings total wickets
summary(inning_summary$total_wickets_taken)

# Create a histogram of 1st-innings total wickets to show the distribution
ggplot(first_innings, aes(x = team1_wickets)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of 1st-Innings Total Wickets taken in the Tournament",
       x = "Innings Total Wickets",
       y = "Frequency") +
  theme_minimal()

# Summary statistics for 1st-innings total wickets
summary(first_innings$team1_wickets)

# Create a histogram of 2nd-innings total wickets to show the distribution
ggplot(second_innings, aes(x = team2_wickets)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of 2nd-Innings Total Wickets taken in the Tournament",
       x = "Innings Total Wickets",
       y = "Frequency") +
  theme_minimal()

# Summary statistics for 2nd-innings total wickets
summary(second_innings$team2_wickets)
#----------------------------------

# Create a pie chart to show the distribution of matches hosted by each ground
ggplot(ground_counts, aes(x = "", y = n, fill = ground)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar(theta = "y") +
  labs(title = "Distribution of Matches Hosted by Ground") +
  theme_void() +
  scale_fill_brewer(palette = "Set3") +  # Optional: Choose different color palettes
  geom_text(aes(label = n),
            position = position_stack(vjust = 0.5),
            color = "black", size = 5)  # Adjust text color and size as needed
#-------------------------------------------

# Create a box plot to compare runs per match across different grounds
```

```
ggplot(merged_matches, aes(x = reorder(ground, -match_total_runs, FUN = median), y =
match_total_runs, fill = ground)) +
  geom_boxplot(outlier.colour = "red", outlier.size = 2, alpha = 0.7) +
  scale_fill_brewer(palette = "Set3") +  # Adding color palette
  theme_minimal(base_size = 14) +  # Increase font size for better readability
  labs(
    title = "Comparison of Runs per Match by Ground",
    x = "Ground",
    y = "Total Runs per Match",
    fill = "Ground"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),  # Rotate x-axis labels for
readability
    plot.title = element_text(hjust = 0.5, face = "bold"),  # Center and bold the title
    legend.position = "none"  # Remove legend for cleaner look
  )
#---------------------------------------------

# Create a box plot to compare wickets per match across different grounds
ggplot(merged_matches, aes(x = ground, y = match_total_wickets, fill = ground)) +
  geom_boxplot(outlier.shape = NA, color = "black") +  # Box plot with no outlier points shown
  labs(
    title = "Comparison of Total Wickets Per Match Across Grounds",
    x = "Ground",
    y = "Total Wickets Per Match"
  ) +
  scale_fill_brewer(palette = "Set3") +  # Attractive color palette
  theme_minimal() +                      # Clean minimal theme
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for better readability
    plot.title = element_text(size = 14, face = "bold") # Enhance title font
  )
#-----------------------------------------------

# Create bar plot of Batting Score Averages by team
ggplot(team_performance, aes(x = reorder(team, batting_score_average), y =
batting_score_average, fill = team)) +
  geom_bar(stat = "identity", color = "black", show.legend = FALSE) +
  coord_flip() +
  labs(
    title = "Batting Score Averages of Teams",
    x = "Team",
    y = "Batting Score Average"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")

# Create bar plot of Bowling Score Averages by team
ggplot(team_performance, aes(x = reorder(team, -bowling_score_average), y =
bowling_score_average, fill = team)) +
  geom_bar(stat = "identity", color = "black", show.legend = FALSE) +
  coord_flip() +
  labs(
    title = "Score Averages against Teams",
    x = "Team vs.",
    y = "Bowling Score Average"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")

# Create bar plot of total runs scored by each team
ggplot(team_performance, aes(x = reorder(team, -total_batting_runs), y = total_batting_runs, fill
= team)) +
  geom_bar(stat = "identity", color = "black", show.legend = FALSE) +
  labs(
    title = "Total Runs Scored by Each Team",
    x = "Team",
    y = "Total Runs Scored"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")

# Create bar plot of total wickets taken by each team
ggplot(team_performance, aes(x = reorder(team, -total_wickets_taken), y = total_wickets_taken,
fill = team)) +
  geom_bar(stat = "identity", color = "black", show.legend = FALSE) +
  labs(
    title = "Total Wickets Taken by Each Team",
    x = "Team",
    y = "Total Wickets Taken"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")
```

```
# Create bar plot of Wicket-Taking Average by team
ggplot(team_performance, aes(x = reorder(team, -wicket_taking_average), y =
wicket_taking_average, fill = team)) +
  geom_bar(stat = "identity", color = "black", show.legend = FALSE) +
  coord_flip() +
  labs(
    title = "Wicket-Taking Averages of Teams",
    x = "Team",
    y = "Wicket-Taking Average"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set1")

# Create bar plot will show the total number of wins for each team
ggplot(team_wins_count, aes(x = reorder(winner, -win_count), y = win_count, fill = winner)) +
  geom_bar(stat = "identity", color = "black", show.legend = FALSE) +
  labs(
    title = "Total wins of Teams",
    x = "Team",
    y = "Win Count"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Paired")

# Create bar graph of the top run scorers
ggplot(top_run_scorers, aes(x = reorder(player, total_runs), y = total_runs, fill = total_runs)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top Run Scorers of the Tournament", x = "Player Name", y = "Total Runs") +
  theme_minimal() +
  scale_fill_gradient(low = "lightblue", high = "darkblue")

# Create bar graph for top strike rates
ggplot(strike_rate_data, aes(x = reorder(player, strike_rate), y = strike_rate, fill = strike_rate)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top Players by Strike Rate (Runs > 100)", x = "Player Name", y = "Strike Rate") +
  theme_minimal() +
  scale_fill_gradient(low = "lightgreen", high = "darkgreen")

# Create bar graph for top averages
```

```
ggplot(batting_average_data, aes(x = reorder(player, batting_average), y = batting_average, fill
= batting_average)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top Players by Batting Average (Innings > 3)", x = "Player Name", y = "Batting
Average") +
  theme_minimal() +
  scale_fill_gradient(low = "lightcoral", high = "darkred")

# Create bar graph for top wicket-takers
ggplot(wicket_takers_data, aes(x = reorder(player, total_wickets), y = total_wickets, fill =
total_wickets)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top Wicket Takers of the Tournament", x = "Player Name", y = "Total Wickets") +
  theme_minimal() +
  scale_fill_gradient(low = "lightblue", high = "darkblue")

# Create bar graph for top economical bowlers
ggplot(economical_bowlers_data, aes(x = reorder(player, -economy_rate), y = economy_rate,
fill = economy_rate)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top Economical Bowlers (Bowled > 15 Overs)", x = "Player Name", y = "Economy
Rate") +
  theme_minimal() +
  scale_fill_gradient(low = "lightcoral", high = "darkred")

# Create bar graph for top bowling averages
ggplot(bowling_average_data, aes(x = reorder(player, -bowling_average), y = bowling_average,
fill = bowling_average)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top Bowlers by Bowling Average (More Than 5 Wickets)", x = "Player Name", y =
"Bowling Average") +
  theme_minimal () +
  scale_fill_gradient(low = "lightcoral", high = "darkred")

# Create bar plot for batting average by Ground
ggplot(runs_summary, aes(x = ground, y = avg_runs, fill = factor(inns))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Runs in 1st vs 2nd Innings by Ground", x = "Ground", y = "Average
Runs", fill = "Inning") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
# Create scatter plot of 1st innings runs vs 2nd innings runs for each ground
ggplot(merged_matches, aes(x = Team1_total, y = Team2_total, color = ground)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "black") +
  labs(title = "Runs in 1st Innings vs 2nd Innings by Ground", x = "Runs in 1st Innings", y =
"Runs in 2nd Innings", color = "Ground") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 14), legend.position = "right")

# Create bar plot for win percentage by ground and winning inning
ggplot(win_summary, aes(x = ground, y = win_percentage, fill = winning_inning)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Win Percentage of 1st vs 2nd Innings by Ground", x = "Ground", y = "Win
Percentage", fill = "Winning Inning") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Create bar chart showing win percentages for teams batting 1st vs 2nd at each ground
ggplot(win_summary, aes(x = ground, y = win_percentage, fill = batting_first_winner)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Win Percentages for Teams Batting 1st vs 2nd at Each Ground", x = "Ground", y =
"Win Percentage (%)", fill = "Batting Order") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), plot.title = element_text(hjust = 0.5,
size = 14)) +
  scale_fill_manual(values = c("1st Innings" = "blue", "2nd Innings" = "red"))

# Create bar plot for wickets per inning by ground
ggplot(wickets_summary_long, aes(x = ground, y = avg_wickets, fill = inning)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Wickets in 1st vs 2nd Inning by Ground", x = "Ground", y = "Average
Wickets Taken", fill = "Inning") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Create scatter plot of wickets fallen in 1st innings vs 2nd innings for each ground
ggplot(merged_matches, aes(x = Team1_wickets_fallen, y = Team2_wickets_fallen, color =
ground)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Wickets Fallen in 1st Innings vs 2nd Innings by Ground", x = "Wickets Fallen in 1st
Innings", y = "Wickets Fallen in 2nd Innings", color = "Ground") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 14), legend.position = "right")

# Create bar plot for win percentage by strategy (target setting or chasing)
```

```
ggplot(win_percentage, aes(x = Team_1, y = win_percentage, fill = winning_strategy)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Win Percentage by Target Setting vs Target Chasing", x = "Team", y = "Win Percentage", fill = "Strategy") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```