What Variables Impact Wicket Production in the IPL*

Wicket can come in bunches in the IPL

Siddharth Gowda

December 13, 2024

The paper analyzes what variables are the best predictors of wickets in the Indian Permier League (IPL). The current over, the bowling style, batting hand, and the number of wickets in the previous over impact the probability of a wicket occuring. Moreover, the number of wickets in the previous over is the greatest predictor of wickets occuring. These findings can be used by IPL teams when attempting large comebacks or when trying to prevent comebacks.

Table of contents

*Code and data are available at: https://github.com/siddharthgowda/ipl_wicket_analysis.

1 Introduction

The game of cricket is simple. One player tries to bowl or delivery a ball to another player: the batsman. The batsman then tries to hit the ball onto the field in way that generates the most runs. Runs in cricket are scored by the two batsmen running between the wickets, the ball rolling out of the venue or if the ball is hit of the venue in the air. However, batters can get out or dismissed, and this called a wicket. In cricket, once a wicket occurs, a batsman cannot return to bat. As a result, one or two wickets can completely change a match.

The goal of this paper is to figure out which variables are the best at predicting a future wicket in T20 Cricket. In T20 cricket, there are only 20 overs for each team to score. For more information on T20 cricket and cricket in generally read this article (USA TODAY 2024). The estimated for the research paper is the probability a ball will result in a wicket.

The analysis was done using IPL (Board of Control for Cricket in India (BCCI) 2024) men's data using the `cricketdata` (Hyndman et al. 2023) R (R Core Team 2023) CRAN package. The `cricketdata` (Hyndman et al. 2023) package gets its data from ESPN Cricinfo (ESPN 2024) and Cricsheet (Cricsheet 2024). All code and data analysis was created using R (R Core Team 2023). The `tidyverse` (Wickham et al. 2019) package was used for data cleaning and feature creation, `knitr` (Xie 2023) was used for formatting tables, `here` (Müller 2020) was used for reading in data, `modelsummary` (Arel-Bundock 2022) was used to generate summaries for the models created, and `marginaleffects` (Arel-Bundock, Greifer, and Heiss Forthcoming) was used to generate predictions based on the models.

The paper explore the IPL (Board of Control for Cricket in India (BCCI) 2024) play-by-play from the 2021, 2022, 2023, and 2024 seasons. The paper first used descriptive statistical analysis techniques to figure out which variables most associated with wicket prediction. This incorporated summary tables, scatter plots with trend lines, histograms, and more. From this analysis, the current over, the number of wickets in the previous over, the bowling style, and the batting style were determined to be the most important factors in terms of wicket production.

From those variables, three logistic models were created ranging from lower complexity to high complexity. Lower complexity models had less input variables while high complexity models more input variables. The response variable was wicket occurrence, a Boolean indicating whether or not a wicket occurred. The training and testing split for these models was 80% to 20%. The model that used the over number and the number of wickets in the previous over as the input variables did the best job of predicting wickets. Ultimately, the number of wickets in the previous over was the best predictor of a wicket, suggesting that wickets in the IPL (Board of Control for Cricket in India (BCCI) 2024) can come in bunches.

2 Data

Table 1: All Variables in the Cleaned Data Set

Match ID	Over	Striker	Bowler	Runs Off Bat	Wicket	Run Rate	Batting Style	Bowling Style	Previous Over Wickets
1254058	1	RG Sharma	Mohammed Siraj	0	FALSE	6	Right hand Bat	Right arm Fast	0

Table 1 show a sample of a row in the cleaned data set. While there technically are more variables in the data set, these are the most important variables in the data set. Information about all variables in the cleaned data set are visible in Section .2, including variables not in Table 1.

2.1 Measurement

All cricket data is from the R (R Core Team 2023) package `cricketdata` (Hyndman et al. 2023). The data that was used is from the CRAN version of the package and not the Github (GitHub 2024) developer tools version because the CRAN is more stable. The `cricketdata` (Hyndman et al. 2023) package takes data from ESPN Cricinfo (ESPN 2024) and Cricsheet (Cricsheet 2024). While not stated by `cricketdata` (Hyndman et al. 2023), based on the code, the team behind the package are scrapping the HTML of ESPN Cricinfo (ESPN 2024) and downloading CSV data from Cricsheet (Cricsheet 2024), As a result, anyone using the package must respect the rate limits set by these sites. Therefore, for this analysis the raw data was first saved as CSV before performing any other data cleaning and analysis.

More information about the raw data sets, measurements, and data sources is in Section .4.

2.2 Data Cleaning

In terms of data cleaning, only data from the IPL (Board of Control for Cricket in India (BCCI) 2024) seasons from 2021 to 2024 (the most recent tournament) is used. This was done to make sure all data analysis about IPL cricket is up to date, as certain strategies that used to be effective are no longer effective in IPL (Board of Control for Cricket in India (BCCI) 2024) cricket. Also, two data sets were used, one was player metadata and another was for play by play data (more about this in Section .4). The IPL men's data set had play-by-play data for each ball bowled in each game in the IPL season. The two data sets were merged so that striker (batsman) metadata and bowler metadata were included in each row. This including

things like batting style and bowling style. Moreover, certain rows with missing data were removed and the current run rate and number of wickets in the previous over variables were created based variables originally included in the raw data set. Also, some data rows had clearly incorrect values, like a innings value of 6, when there are only 2 innings in T20 IPL cricket. These rows were removed from analysis.

2.3 Predictor Variables

All other variables besides wicket listed in the begging of Section 2 are predictor variables.

Table 2: Fast Bowlers Are More Common Then Spinners

Bowling Style	Number of Bowlers
Left arm Fast	4
Left arm Fast medium	13
Left arm Medium	6
Left arm Medium fast	9
Left arm Wrist spin	4
Legbreak	17
Legbreak Googly	11
Right arm Fast	25
Right arm Fast medium	28
Right arm Medium	29
Right arm Medium fast	22
Right arm Offbreak	27
Slow Left arm Orthodox	23

Based on Table 2, there are more pace bowlers than spin bowlers. Spin bowlers include wrist spin, leg break, and leg break googly, and off break, and pace bowlers are fast, medium fast, and medium. Also, there are more right arm bowlers than left arm bowlers.

Table 3: There are more right hand batsman.

Batting Style	Number of Batters
Left hand Bat	83
Right hand Bat	207

From Table 3, significantly more batsman are right handed than left handed.

2.4 Relationship Between Wickets and other Variables

2.4.1 Wickets and Run Rate

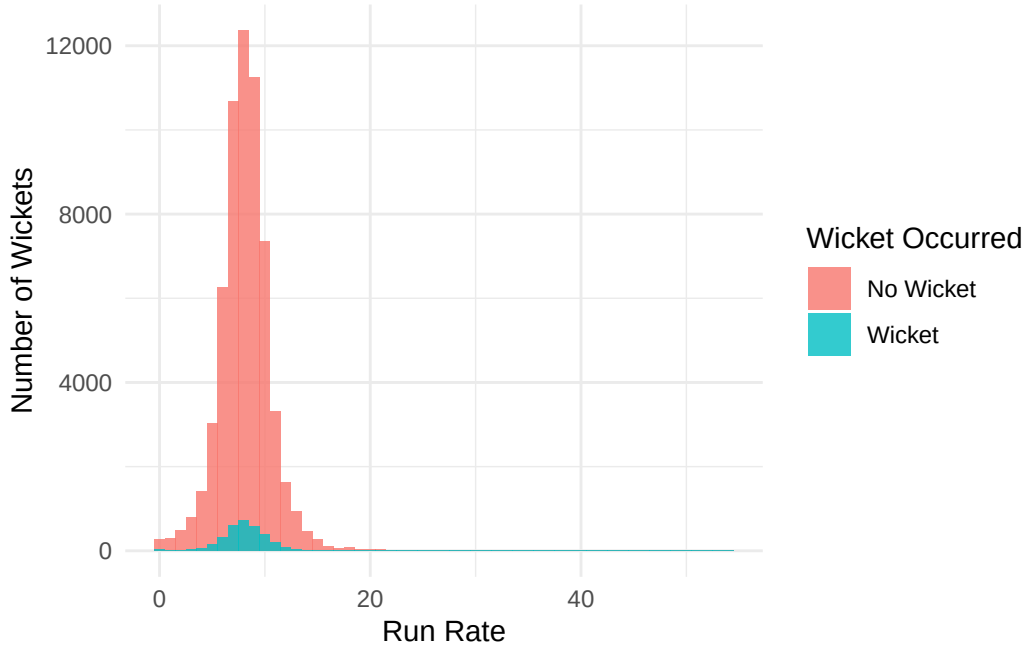

Figure 1: Current Run Rate is Not Associated With Wicket Production

From Figure 1, both the distribution of current run rates of teams when a wicket occurs and when a wicket does not occur are approximately normally distributed around a 8.5 run rate. The variability for both distributions also appears to be the same. However, the distribution of run rate for balls that are not wickets occur has more outliers than for wickets occurrences.

2.4.2 Wickets and Over

From Figure 2, there is a pretty strong linear positive relationship between the over number and the number of wickets that occur. It is important to note that in the last over (over 20) wicket occur way more than other overs. Also over 5 has more wickets than the linear trend.

2.4.3 Wickets Based on Batsman Bowler Matchup

Based on Figure 3, for left hand batters, left arm fast bowlers have the highest chance of generating a wicket. For right hand batters, medium pace or medium fast pace bowlers are the most likely to get wickets.

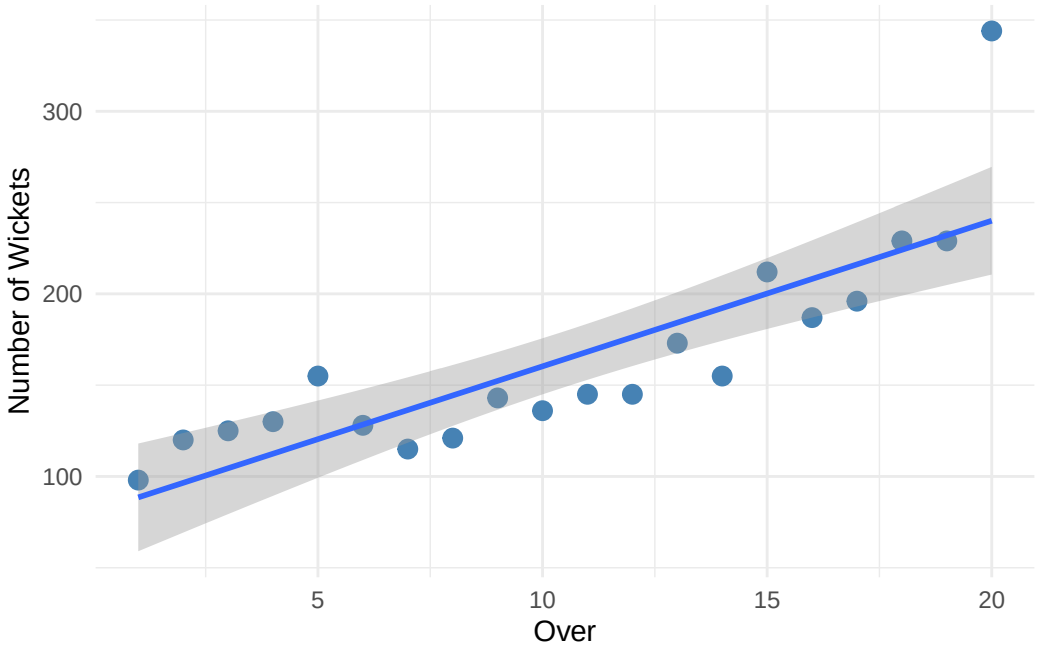

Figure 2: Wickets are more likely later in the innings

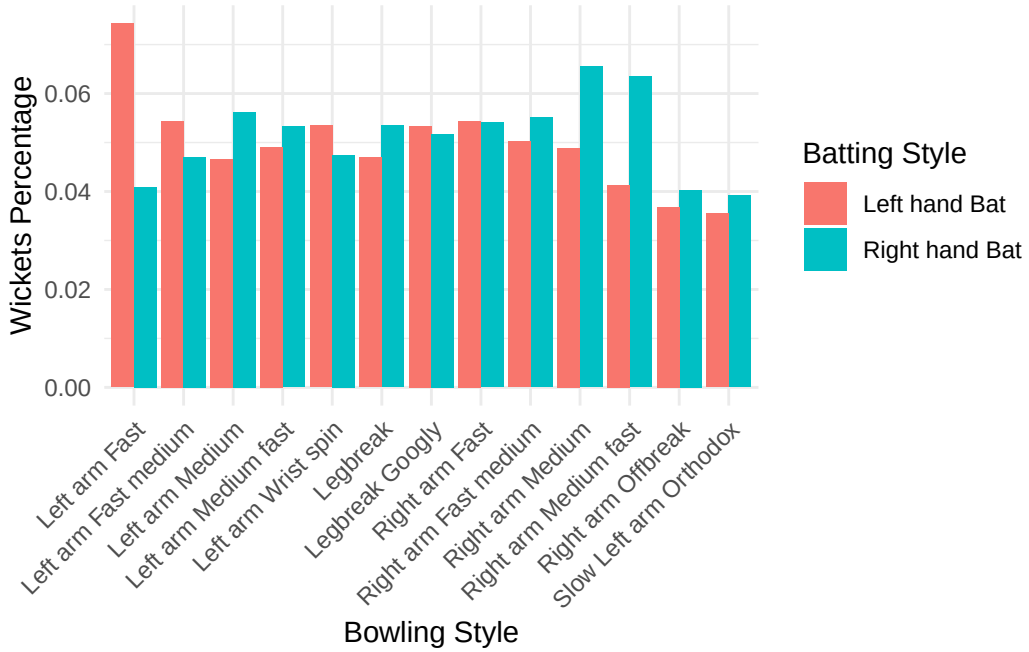

Figure 3: Fast Bowlers Generate More Wickets

2.4.4 Wickets and Number of Wickets in the Previous Over

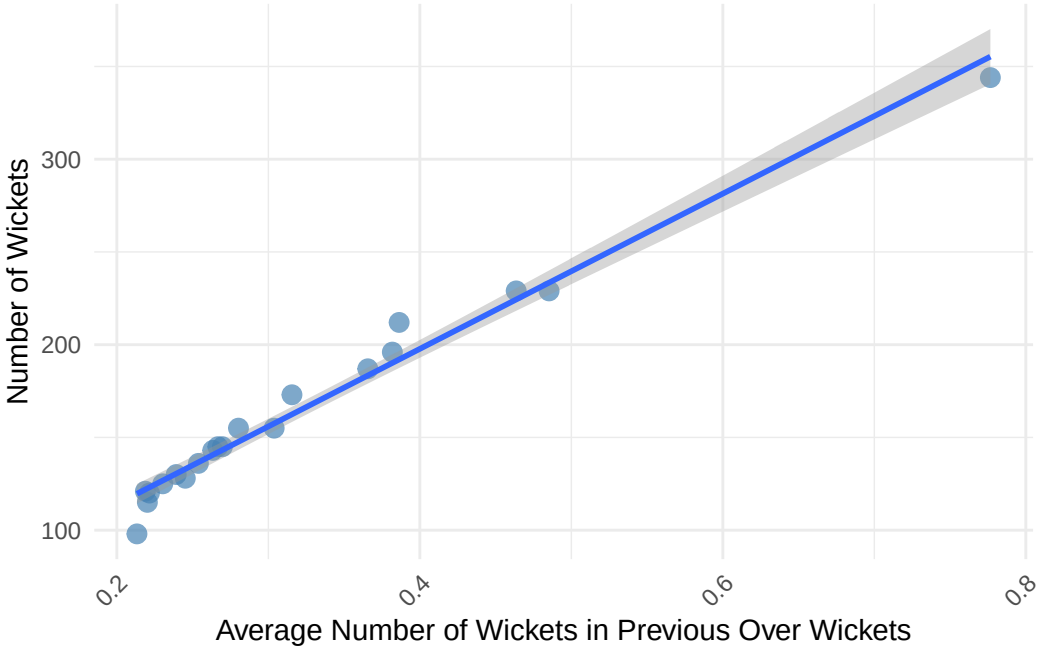

Figure 4: Positive Association Between Wickets in the Last and Next Over

In Figure 4, there is a strong positive correlation between the average number of wickets in the previous over and the number of wickets in the current over. It is important to note that there is a gap in data for 0.5 average wickets in previous over to 0.8. Thus, it is difficult to say if from 0.5 to 0.8 the trend is still positive and linear.

3 Model

The goal of the model is accurately predict when a wicket will occur in an IPL (Board of Control for Cricket in India (BCCI) 2024) game based on relevant variables from the data set. Furthermore, the idea is to create a model that would achieve the most accuracy with the least amount of variables. To achieve this, three different generalized binomial family linear models (a form of logistic regression) were used. The first model only had one input variable, the second one had two input variables, and third had four input variables. All models had the wicket Boolean variable as the response variable.

3.1 Model Set-up

All models are set up using the `glm` base R (R Core Team 2023) function. It uses the binomial family. All models trained on a training data set, which contains 80% of the original data. The other 20% is the testing set.

3.1.1 Single Variable Generalized Linear Model

The single variable model uses a GLM model with only the over number as the input variable.

$$\Pr(\text{Wicket} = 1) = \text{logit}^{-1}\left(\beta_0 + \beta_1 \cdot \text{over}\right)$$

Beta Explanation:

β_0: Represents the general intercept, which represents the log-odds of a wicket occurring when the over variable does not matter

β_1: Represents the coefficient for the current over number

Based on Table 4, modeling with only the over variable accounts 0.22 RSME variability. Also for an increase in one over, the log odds of a wicket occurring increase by 0.058213 with a p value that is almost zero (less than $2 * 10^{-16}$). In terms of deviance, while not shown in the table, residual deviance of the model is roughly 300 degrees of freedom less than the null. The null deviance is 20849 on 51586 degrees of freedom while the residual deviance is 20586 on 51585 degrees of freedom.

3.1.2 Model Set-Up for Two Variable Variable Generalized Linear Model

The two variable model uses the over number and number of wickets in the previous over as the input variables.

$$\Pr(\text{Wicket} = 1) = \text{logit}^{-1}\left(\beta_0 + \beta_1 \cdot \text{over} + \beta_2 \cdot \text{previousOverWickets}\right)$$

Table 4: Logistic Regression Model Ouput for Wicket Prediction Using Only the Over

	Wicket Prediction (Single Variable Input)
(Intercept)	−3.569
	(0.048)
over	0.058
	(0.004)
Num.Obs.	51 587
AIC	20 589.7
BIC	20 607.4
Log.Lik.	−10 292.858
RMSE	0.22

β_0: Represents the general intercept, which is the change in log-odds that occurs when all other variables cancel each other out

β_1: Represents the coefficient for the current over number

β_2: Represents the coefficient for the number of wickets in the previous over

Based on Table 5, modeling with only the over variable accounts 0.22 RSME variability. Also for an increase in the number of wickets in the previous over, the log odds of a wicket occurring increase by 1.852 with a p value that is almost zero (less than $2 * 10^{-16}$). The null deviance is 20849 on 51586 degrees of freedom while the residual deviance is 16584 on 51584 degrees of freedom. The logs odds increase for the over variable is 0.005 with a p value of 0.168.

3.1.3 Model Set-Up for Four Variable Generalized Linear Model

Adding the two extra variables, batting style and bowling style, to the model did not change the effectiveness in predictability at all. More rigorous analysis available in Section .1.

3.2 Model justification

Based on Figure 2, there was a strong positive linear relationship between the over and the number of wickets taken, suggesting that the over can be a predictor of a wicket occurring. From Figure 3 certain match-up produced a higher chance of a wicket occurring, such as left hand fast bowling on a left hand batter, which is why both variables were added to the four variable model in Section .1. Finally from Figure 4, the average number of wickets that occurred in the previous over had a strong positive linear relationship with the number of

Table 5: Logistic Regression Model Ouput for Wicket Prediction Using Over and Previous Wicket

	Wicket Prediction (Two Variable Input)
(Intercept)	−4.139
	(0.052)
over	0.005
	(0.004)
prev_over_wickets	1.852
	(0.031)
Num.Obs.	51 587
AIC	16 590.1
BIC	16 616.7
Log.Lik.	−8292.074
RMSE	0.21

wickets that occurred in balls in the next over, implying that previous over wickets impact the wicket probability of a ball in the next over.

4 Results

4.1 One Variable Model

Based on Figure 5, there's a strong linear positive relationship between the over number and the single variable model wicket prediction probability. Also, in the 20th over, the model will almost always predict a wicket.

Table 6: One Variable Model Underfits the Test Data

Was Actually A Wicket?	Correct	Incorrect
No	12253	0
Yes	0	644

From Table 6, the model has is 95.000% accurate, with a 100% prediction accuracy when a wicket does not occur. However, the model has 0% change of correctly predicting a wicket when one actually occur. For this model, it is assumed that when the estimated probability of a wicket occurring is greater than or equal to 50%, the model is predicting a wicket.

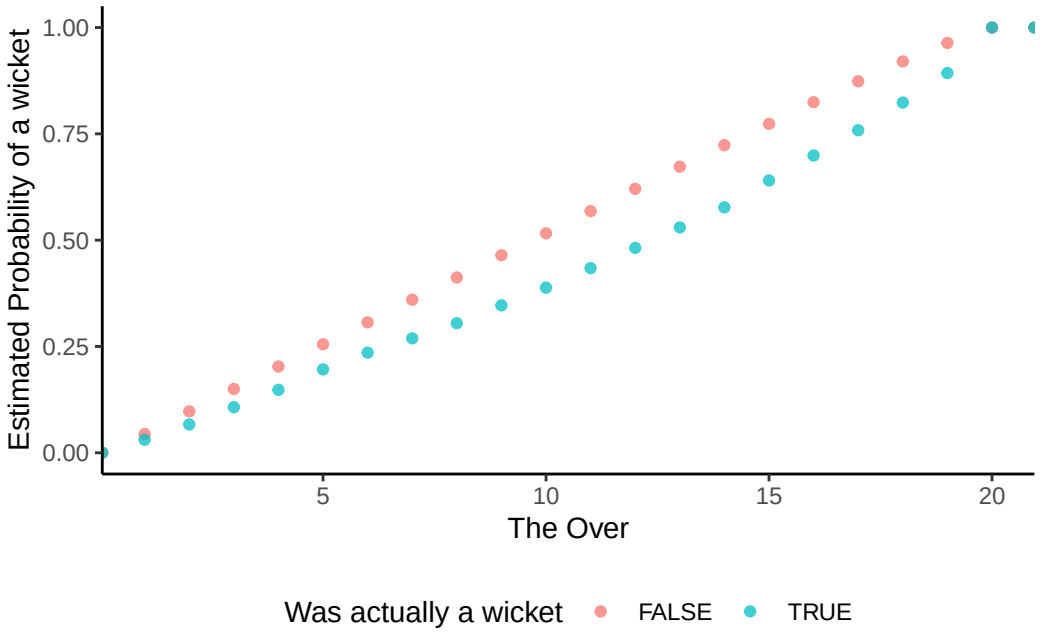

Figure 5: Linear Corrleation Between Wicket Probability and the Over

4.2 Two Variable Model

Based on Figure 6, the number of wickets in the previous over had a positive log-like relationship with the estimated probability of a wicket occurring in the two variable model. Also, when the previous over had two or more wickets, the model predicts 90% or more chance of a wicket occurring. It is important to note that this model also factors in variation due to the current over of the game when generating predictions probability estimates.

Table 7: Two Variable Model Struggles to Predict Wickets in Test Data

Was Actually A Wicket?	Correct	Incorrect
No	12229	24
Yes	17	627

Via Table 7, the two variable model has a 94.952% accuracy on the test data. Wicket occurrences are predicted with a 2.640% accuracy and a wicket not occurring is predicted with a 99.804% accuracy. For this model, it is assumed that when the estimated probability of a wicket occurring is greater than or equal to 50%, the model is predicting a wicket.

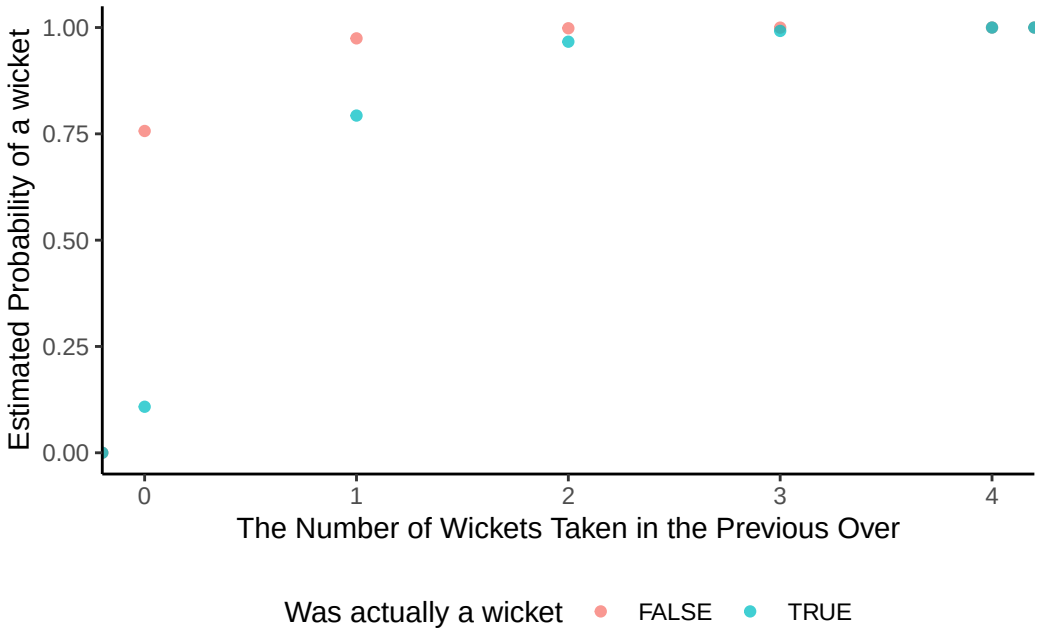

Figure 6: Logarithmic Relationship Between Wicket Probability and the previous overs wickets

4.3 Four Variable Model

The four variable model is not discussed in the results section since most of the variables did not have a large impact on the model. More information on that model in Section .1.

5 Discussion

5.1 Variables that Lead to Wickets

From the analysis, there were three variables that seemed to impact the probability of a wicket occurring: the over the ball was bowled in, the number of wickets in the previous over, and the bowling style and the batting style match up. From Figure 5 and Figure 2, there is a strong evidence to suggest that wickets are more likely to happen later in overs. This is likely true for two reasons. One is that in T20 cricket, batters are willing to take more risks in later over because there are only a few balls left and they want to maximize the amount of runs they can score. The other interpret is that in later over, top batsman are usually already out, so lower quality batsman are playing and they are more likely to get out. As a result, wicket occurrences are more likely. While this finding is not anything most cricket players or teams do not know, it is still important to see it backed up through data since oftentimes convention wisdom is not actually true in data. Also, from Figure 2, the fifth over seems to have an unusually high amount of wickets compared to the trend line. However, this makes sense since the fifth over is the end of the power play. In the power play, fielders have to be positioned closer to the pitch, making it easier to score runs. Therefore, batsman might be willing to take more risks in the 5th over so they can maximize the runs scored in the power play, leading to more wickets.

Based on Figure 6 and Figure 4, if more wickets were taken in the previous over, the more likely it is to get a wicket in the next over. This also can be interpreted in two ways. One is that when more wickets are taken in the previous over, worse batsman are playing in the next over are worse batsman are more likely to get out than better batsman. The other interpretation is that when a lot of wickets occur in the previous over, new batsman come in and new batsman are not in rhythm and are still trying to figure learn the bowlers' strategies. Thus, they are more likely to get out, resulting in a wicket. From the second interpretation, if a team is behind by a lot in a match and suddenly gets a wicket, in the next over, the captain could go to their top wicket producing bowler in order to get more wickets to put them back in the game.

Via Figure 3, certain bowling and batting match-ups result in more wickets. The general trend is that fast bowlers are more likely to take wickets compared to spinners (wrist spin, leg break, leg break googly, off break). The chart showed that left hand fast bowlers had the highest wicket taking percentage against left handed batters. and that right hand medium and right hand medium fast bowlers had the highest wicket taking percentage against right handed batters. However, there are limitations to these conclusions. First, there were only four left-hand fast bowlers in this data set (Table 2), and there are a lot more right handed batsman than left handed batsman (Table 3). However, assuming the trends in the graph are true, a captain could potentially use the trend to choose bowlers that give them the best chance of a wicket, depending on the batsman match up.

From Figure 1, there does not appear to be a relationship between the current run rate of a team and if a wicket occurs. This is important because it shows that teams should not merely assume that if the opposition has an extremely high run rate, they are taking too many risks and will eventually get out.

5.2 Model Discussion

From Section 3, three models were created, all predicting the probability a wicket would occur. The first model used the over as the only input; the second model used the over and the number of wickets in the previous over as the two input variables; and the third model used the the over, the number of wickets in the previous over, the bowling style, and the batting style (left hand or right hand) as the four input variables. It is clear that the second model is the most effective at predicting when a wicket will occur. While the first model does technically predict more accurately than the second model (95.000% compared to 94.952%), the first model never correctly predicting when a wicket would occur in the test data (Table 6). In terms of the third model, none of the variables' coefficients were statistically significant except for the number of wickets in the previous over. Therefore, that model cannot be the best. However, while the second model is the best predictor, it also rarely correctly predicted when a wicket would occur (2.640% wicket prediction accuracy).

Based on the second model, it clear that the number of wickets in the previous over is by far the most important variable when predicting wicket occurrences. From Figure 6, a jump from zero to two wicket in the previous over almost guaranteed a wicket prediction from the model. This is important because it demonstrates than in the IPL wickets can potentially come in bunches. Therefore, if a team is leading by a significant margin in a match and one of their batters gets out, the captain might be wise to choose a more stable batsman to replace him rather than one that might score more runs, but is more likely to get out.

5.3 Next Steps, Weaknesses, and Limitations

The biggest limitation of this study comes from the fact that only IPL (Board of Control for Cricket in India (BCCI) 2024) data was used. The IPL (Board of Control for Cricket in India (BCCI) 2024) is a T20 Cricket League that has the best players in the world, so it will be difficult to use these findings in other leagues and especially in different cricket formats like test and one-day-international. The IPL (Board of Control for Cricket in India (BCCI) 2024) pitches and stadiums generally favor offense and scoring a lot of runs compared to bowling. This also makes the conclusions from this paper less applicable in other cricket leagues.

The biggest weakness of this paper comes from the lack of fielding data. In the future, the paper would incorporate play-by-play data that shows the position of the fielders and who recovers a ball and some distance and speed data, like distance between the ball and nearest fielder and speed or rpms (revolutions per minute) of a bowling delivery. Also in the future,

the paper would incorporate data from other T20 cricket leagues to verify if these findings can be applicable to all forms of T20 cricket.

Appendix

.1 Four Variable Model

The model is set up using the `glm` base (R Core Team 2023) CRAN package. It uses binomial family with the specified formula.

$$\Pr(\text{Wicket} = 1) = \text{logit}^{-1}\left(\beta_0 + \beta_1 \cdot \text{over} + \beta_2 \cdot \text{previousOverWickets} + \beta_3 \cdot \text{BowlingStyle} + \beta_4 \cdot \text{BattingStyle}\right)$$

β_0: Represents the general intercept, which is the value that occurs when all other variables cancel each other out.

β_1: Represents the coefficient for the current over number

β_2: Represents the coefficient for the number of wickets in the previous over

β_3: Represents the coefficient for the bowling style. Each bowling style has its own coefficient.

β_4: Represents the coefficient for the batting style (left handed or right handed). Each batting style has its own coefficient.

Based on Table 8, none of the batting styles or bowling style coefficient are statistically significant all with p values greater than 10%. This implies that most likely these variables are not great predictors of wicket creation. The number of wickets in the previous over still has a statistically significant impact and has an increase of 0.031446 log odds per an increase of one wicket in the previous over. The residual deviance is 16574 on 51571 degrees of freedom compared to the null of 20849 on 51586 degrees of freedom.

Table 8: Batting Style vs Bowling Style Do Not Heavily Impact Wicket Probability

	(1)
(Intercept)	−4.316
	(0.192)
over	0.005
	(0.004)
prev_over_wickets	1.853
	(0.031)
batting_styleRight hand Bat	0.044
	(0.045)
bowling_styleLeft arm Fast medium	0.099
	(0.207)
bowling_styleLeft arm Medium	0.255
	(0.212)
bowling_styleLeft arm Medium fast	0.205
	(0.206)
bowling_styleLeft arm Wrist spin	0.158
	(0.235)
bowling_styleLegbreak	0.257
	(0.212)
bowling_styleLegbreak Googly	0.207
	(0.196)
bowling_styleRight arm Fast	0.190
	(0.192)
bowling_styleRight arm Fast medium	0.211
	(0.197)
bowling_styleRight arm Medium	0.109
	(0.193)
bowling_styleRight arm Medium fast	0.165
	(0.204)
bowling_styleRight arm Offbreak	0.104
	(0.200)
bowling_styleSlow Left arm Orthodox	0.011
	(0.202)
Num.Obs.	51 587
AIC	16 605.8
BIC	16 747.4
Log.Lik.	−8286.914
RMSE	0.21

.2 Cleaned Data Dictionary

Below are all of the variables in the cleaned data set with an explanation of each variable, examples, and the data type.

Match id (Integer)
Example: 335982
A randomly generated ID uniquely identifies each cricket match in the data set.

Year (Integer)
Example: 2022
The year the match was played in.

Venue (string)
Example: M Chinnaswamy Stadium
The location the match is played in.

Innings (Integer)
Example: 1
The innings the current ball is played in. Should only be either 1 or two.

Over (Integer)
Example: 7
The over the current ball is played in. This should be from 1 to 20 in T20 Cricket.

Ball (Integer)
Example: 3
The current ball number in the over. This typically should be from 1 to 6, but could be more if no balls or wides are given.

Batting Team (String)
Example: Kings XI Punjab
The team that is currently batting.

Bowling Team (String)
Example: Mumbai Indians
The team that is currently bowling.

Striker (String)
Example: AJ Finch
The player that is currently on the wicket the bowler is bowling to.
Bowler (String)
Example: SMSM Senanayake
The player that is bowling.

Runs off of the bat (Integer)
Example: 3
The number of the runs the team scored from this ball directly off of the striker's bat.

Wicket Lost Yet (Boolean)
Example: True
Did a wicket occur in this ball.
Wickets Lost (Integer)
Example: 7
The number of wickets that have fallen. From 0 to 10.

Target (Integer)
Example: 207
The par course for the given venue's pitch on that match's day.

Run Rate (Float)
Example: 6.50
The current run of the batting team in the match. This is the number of runs the team scores per over.

Batting Style (String)
Example: Left hand Bat
The batting style of a player. This is either left hand or right hand.

Bowling Style (String)
Example: Slow Left arm Orthodox
The bowling style of a player. This only exists if a player is a bowler.

Batting Player Role (String)
Example: Allrounder
The role of a batting player on the team.

Bowling Player Role (String)
Example: Bowler
The role of a bowling player on the team.

Previous Over Wickets (Number)
Example: 2
The number of wickets the bowling team had in the previous over.

.3 Idealized Methodology

This section will go over a comprehensive plan that could be used to translate real time cricket plays to actual data sets and databases that can be accessed by individuals interested in performing statistical analysis. Since this data is not a survey and is not really sampling based, there will be no mention of terms like stratified or cluster sampling or selection and

response bias. This example will only be about IPL (Board of Control for Cricket in India (BCCI) 2024) cricket since this paper is only about the IPL (Board of Control for Cricket in India (BCCI) 2024). However, most of this methodology can be used in other cricket leagues and other formats.

.3.1 Translating Plays to Data

The process of converting live cricket plays into structured data begins with real-time observation and documentation. During an Indian Premier League (Board of Control for Cricket in India (BCCI) 2024) match, a dedicated data entry team should closely monitor every moment of the game, capturing critical information about each play as it unfolds. A primary data collector focuses on immediate, live-capturable elements such as the bowler's name, the batsman on strike, the non-striking batsman, the type of bowling delivery, number of runs scored, and any fielding actions. This real-time data collection requires trained personnel with a deep understanding of cricket's nuanced rules and scoring mechanisms.

Simultaneously, a secondary team prepares for post-match data verification and enrichment. After the live match, this team reviews video recordings, carefully cross-referencing the initial data entries to correct any potential errors or omissions. This post-match review allows for more precise data collection, particularly for complex plays that might have been challenging to capture in real-time. The verification process involves multiple checks, including reviewing ball-tracking technologies, replay footage, and official match scorecards to ensure absolute accuracy.

The data entry process is designed with multiple layers of validation. Trained data entry specialists should use specialized software that provides immediate validation checks, flagging potential inconsistencies or unusual entries for immediate review. This approach minimizes human error and ensures the highest possible data integrity. The entire process is structured to capture not just the basic outcomes of plays, but more distance and vector based data like fielding, ball trajectory, bowling rpms and speed, and more.

.3.2 Making Data Easy to Use

To make this data easy to use, the team should create a clean structured or SQL database. The proposed database design incorporates multiple interconnected tables that provide a comprehensive view of IPL cricket data. A player table should serve as a central table, with each player assigned a unique primary key that can be referenced across other tables. This player's primary key ID becomes a link in tables such as play, match, and season tables, enabling complex queries and data relationships.

The database design prioritizes data reliability and efficiency. By using multiple tables, the system eliminates redundancy and minimizes storage requirements. For instance, a match table would contain high-level match information, while a play table would capture granular,

play-by-play details. A season table allows for broader and simple analysis, tracking changes and trends, winners, top scorers and more, across different IPL seasons. This structured approach not only improves data storage efficiency but also enhances the usability of the data set for researchers, analysts, and cricket fans.

.3.3 Making Data Accessible

Data accessibility is fundamental to the value of this cricket data set. The proposed system implements a controlled access mechanism through a carefully designed API that allows broad yet secure data retrieval. An API key system ensures that access is regulated, preventing potential system overloads while maintaining open accessibility. Rate limiting mechanisms are implemented to protect the system from potential denial-of-service attacks, ensuring consistent data availability for all users.

The public API is deliberately designed as a read-only interface, preventing unauthorized modifications to the core data set. This approach maintains data integrity while still providing transparent access to researchers, analysts, and fans. By making the data publicly accessible through a controlled API, the system inherently supports data validation. Multiple users can cross-reference and verify the data, creating a collaborative environment of data integrity and trust.

.4 Data Sheet for The Raw Data

.4.1 Questions

.4.1.1 Who put the data set together?

The raw data set was acquired from the `cricketdata` (Hyndman et al. 2023) R package. This data set had many contributors which can be seen through this link, but the main developer is Rob Hyndman. The data set scraps the html from ESPN Cricinfo (ESPN 2024) and downloads CSV data from Cricsheet (Cricsheet 2024). These are two reputable organizations when it comes to cricket game data. Cricsheet (Cricsheet 2024) is more so responsible for play by play cricket data, like giving the data for what happens for every ball in a game, whereas ESPN Cricinfo (ESPN 2024) is more responsible for general player career information and statistics.

.4.1.2 Who paid for the data set to be created?

The `cricketdata` (Hyndman et al. 2023) R package is completely open source and free under the GPL-3 license. The source code can be seen through this link. In terms of ESPN Cricinfo (ESPN 2024), this is a site and organization that makes money talking about cricket and airing games. So, it is in their best interest to give basic statistics for free on their website to entice

people to use their services to watch and interact with content related to cricket. Cricsheet (Cricsheet 2024) is operated by Stephen Rushe. He has written code to extract play-by-play cricket match data, which most likely means he is web scraping. There is not much other information about the code that extracts the data and what sites he uses.

.4.1.3 How complete is the data set?

The play by play data is very complete. For each ball in a game, it includes the bowler, batter, the batter on the opposite wicket, the venue, teams, the runs given, and more. The important data that seems to be missing is data related to the position of the fielders and the distance and final location of the ball once and if it hits the bat. In terms of player info, it is also very complete. It includes information about their batting style, blowing style if applicable, and their role on the team. The only issue here is that a player's role on the team can change depending on the team they are on. For instance, a player might be an opening batsman when playing in test cricket but could be a middle over batsman when playing in T20. This data set does not account for this discrepancy.

.4.1.4 Which variables are present, and, equally, not present, for particular observations?

For play-by-play data the bowler, batsman, and teams involved in the play are present and so is general game information. Furthermore, the final outcome of the play, such as the number of runs, or if a wicket occurred are also given. Information about ball speed, distance, location, and fielding is not given. For player information, information about their country of citizenship, country of birth, batting style, bowling style, and batting order, and player role on a team are given. Potential missing observations with the player information are listed in Section .4.1.3.

.4.2 Data Dictionary For Raw Data

For this report, there are only two data sets used, the IPL (Indian Premier League) play-by-play data and player metadata. The variables for these data sets will be listed here with examples.

.4.2.1 Play-by-play IPL Data

Match id (Integer)
Example: 335982
A randomly generated ID uniquely identifies each cricket match in the data set.

Season (String)
Example: 2007/8 or 2024

Represents the IPL tournament year of the match. For instance “2007/8” represents the 2007 to 2008 IPL season and 2024 represents the tournament that was played only in 2024.

Season (Date)
Example: 2024-04-15
Represents the day the match was played. It is in a year-month-day format.

Venue (String)
Example: M Chinnaswamy Stadium
The location the match is played in.

Innings (Integer)
Example: 1
The innings the current ball is played in. Should only be either 1 or two.

Over (Integer)
Example: 7
The over the current ball is played in. This should be from 1 to 20 in T20 Cricket.

Ball (Integer)
Example: 3
The current ball number in the over. This typically should be from 1 to 6, but could be more if no balls or wides are given.

Batting Team (String)
Example: Kings XI Punjab
The team that is currently batting.

Bowling Team (String)
Example: Mumbai Indians
The team that is currently bowling.

Striker (String)
Example: AJ Finch
The player that is currently on the wicket the bowler is bowling to.

Non-Striker (String)
Example: A Ashish Reddy
The player that is batting but is on the opposition wicket.

Bowler (String)
Example: SMSM Senanayake
The player that is bowling.

Runs off of the bat (Integer)
Example: 3
The number of the runs the team scored from this ball directly off of the strikers bat.

Extra Runs (Integer)
Example: 1
The number of runs that the team scored that was not because of the striker's bat. This could be because of wides.

Extra Ball (Boolean)
Example: False
Was this ball an extra ball? For instance, this ball could be extra due to a wide.

Balls Remaining (Integer)
Example: 79
The number of balls remaining in the game. This is from 0 to 120.

Runs Scored (Integer)
Example: 101
The number of runs already scored by the batting team.

Wicket (Boolean)
Example: True
Did a wicket occur in this ball.
Wickets Lost (Integer)
Example: 7
The number of wickets that have fallen. From 0 to 10.

First Innings Total (Integer)
Example: 165
The total number of runs earned in the first innings.

Second Innings Total (Integer)
Example: 168
The total number of runs earned in the second innings.

Target (Integer)
Example: 207
The par course for the given venue's pitch on that match's day.

Wides (Integer)
Example: 4
The number of wides bowled in this over.

No Balls (Integer)
Example: 2
The number of no balls bowled in this over.

Byes (Integer)
Example: 4
Runs scored by byes in this ball. This ranges from 0 to 4.

Leg Byes (Integer)
Example: 2
Runs scored by leg byes in this ball. This ranges from 0-4.

Wicket Type (String)
Example: Bowled
If a wicket occurred, this column will describe how it happened. It includes bowled, stumped, and caught.

Player Dismissed (String)
Example: A Ashish Reddy
If a wicket occurred, this column will have the name of the player that was dismissed.

.4.2.2 Player Metadata

Cricinfo Player id (Integer)
Example: 1175501
A randomly generated ID uniquely identifies each cricket player from Cricinfo data.

Cricsheet Player id (Alphanumeric)
Example: 21d38d47
A randomly generated ID uniquely identifies each cricket player from Cricinfo data.

Unique Name (String)
Example: Alkandari Abdulrahman
The unique name of the player that is stored in the (Hyndman et al. 2023) package. All other tables will only use the unique name of a player and not the full name of the player.

Full Name (String)
Example: AGHM Alkandari Abdulrahman
The full name of a player.

Country (String)
Example: Zimbabwe
The country of citizenship for this player.

Date of Birth (Date)
Example: 1971-09-21
The day the player was born in a year-month-day format.

Birthplace (String)
Example: Nangrahar, Afghanistan
The city and sometimes country a player was born in.

Batting Style (String)
Example: Left hand Bat
The batting style of a player. This is either left hand or right hand.

Bowling Style (String)
Example: Slow Left arm Orthodox
The bowling style of a player. This only exists if a player is a bowler.

Player Role (String)
Example: Allrounder
The role of a player on the team.

References

Arel-Bundock, Vincent. 2022. "modelsummary: Data and Model Summaries in r." *Journal of Statistical Software* 103 (1): 1–23. https://doi.org/10.18637/jss.v103.i01.

Arel-Bundock, Vincent, Noah Greifer, and Andrew Heiss. Forthcoming. "How to Interpret Statistical Models Using marginaleffects in R and Python." *Journal of Statistical Software*, Forthcoming.

Board of Control for Cricket in India (BCCI). 2024. "Indian Premier League." IPL. https://www.iplt20.com.

Cricsheet. 2024. "Cricsheet." https://cricsheet.org/downloads/.

ESPN. 2024. "ESPN Cricinfo." https://stats.espncricinfo.com/ci/engine/stats/index.html.

GitHub, Inc. 2024. "GitHub." *GitHub Repository*. GitHub. https://github.com/.

Hyndman, Rob, Charles Gray, Sayani Gupta, Timothy Hyndman, Hassan Rafique, and Jacquie Tran. 2023. *cricketdata: International Cricket Data*. https://CRAN.R-project.org/package=cricketdata.

Müller, Kirill. 2020. *here: A Simpler Way to Find Your Files*. https://CRAN.R-project.org/package=here.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

USA TODAY. 2024. "USA Today." USA TODAY. https://www.usatoday.com/story/sports/2024/06/01/what-is-out-over-wicket-cricket/73617682007/.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Xie, Yihui. 2023. *knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.