# Data-Analytics_Report (Sample Only)

## Introduction

In cricket, matches are occasionally interrupted due to rain or bad lighting, which leads to reduced overs of play. This affects the chances of winning for both teams. In order to tackle this problem, various methods were introduced, such as Average Run Rate (ARR) or Most Productive Overs (MPO), which often results in leaving one side at a disadvantage. The DL method was introduced in 1997, which calculated the expected number of runs in the lost overs to estimate a new target. This method was significantly better than its predecessors, yet it still had limitations. In this project, we look into a new approach to tackle cricket interruptions in a more 'fair' manner called the Isoprobability method. It was developed by Michael Carter and Graeme Guthrie. As the name suggests, it works on the principle of conserving the probability of winning for each team before and after the interruption. Using data from matches between 1999 and 2011, we test this method and analyze its effectiveness.

## Methods

In order to implement this method, we define a cumulative distribution function $F(r; n, w)$ that models the number of runs $r$ a team is expected to score, given $n$ overs remaining and $w$ wickets in hand. This function will be the foundation for calculating the probability of winning based on various game states, both before and after an interruption.

### Defining Probability of Winning

If Team 2 has a target $t$ and has already scored $s$ runs, then the probability that Team 2 will win the match with the remaining nnn overs and www wickets is given by:

$$1 - F(t - s; n, w)$$

Now, suppose an interruption occurs at this point, reducing the remaining overs to $n'$ and revising the target to $t'$. The isoprobability condition requires that the probability of winning remains consistent before and after the interruption. Therefore, we set:

$$1 - F(t' - s; n', w) = 1 - F(t - s; n, w)$$

This equation ensures that the probability of winning for Team 2 remains unchanged, thereby maintaining fairness.

## Modeling the Run-Scoring Process

To accurately estimate $F(r; n, w)$ we model the run-scoring process with three possible outcomes for each ball, denoted by $b$, the number of deliveries remaining, and $b$, wickets in hand. We assume that the following events can occur on each ball:

1. No Ball/Wide: With probability $p_x$, the score increases by 1 run, but bbb and www remain the same.
2. **Wicket**: With probability $p(b, w)$, a wicket falls, reducing $w$ by 1 and $b$ by 1.
3. **Run Scored**: With probability $q(i; b, w)$, $r$ increases by $i$, while $b$ decreases by 1 and $w$ remains the same.

Using this approach, we build a probability model for each possible score trajectory, leading to a comprehensive cumulative distribution function for run outcomes over the course of the remaining deliveries.

## Boundary Conditions

To finalize the model, we set boundary conditions that represent winning or losing situations:

- $F(r, 0, w) = 1$ if $r \geq 0$ : The team loses if there are no balls remaining.
- $F(r, b, 0) = 1$ if $r \geq 0$ : The team loses if there are no wickets left.
- $F(r, n, w) = 0$ if $r < 0$ : The team wins if they have scored the required runs.

# Estimating Probability parameters

## Wide or No Ball

The probability of wide or no ball is estimated using the following

$$P_x = \frac{\text{Nos extra deliveries}}{\text{Nos extra deliveries} + \text{Nos legitimate deliveries}}$$

## Wickets

The probability of loosing a wicket $p(b, w)$ while having $b$ deliveries and $w$ in hand is estimated using a probit model. Let the unobserved variable be $y^*_{b,w}$ defined as

$$y^*_{b,w} = \alpha_0 + \alpha_1 b + \alpha_2 w + \theta_{b,w}$$

where $\alpha_0, \alpha_1, \alpha_2$ are constants and $\theta_{b,w} \sim N(0, 1)$. According to the property of probit model, a wicket falls if $y^*_{b,w} < 0$ which occurs with a probability of $p(b, w) = \Phi(-\alpha_0 - \alpha_1 b - \alpha_2 w)$ where $\Phi$ is cumulative distribution function of a normal distribution. Using our knowledge of the game we can say:

1. $\alpha_1 > 0$ as it is more likely for a wicket to fall as the game progresses.
2. $\alpha_2 > 0$ as lower batting order more easily to fall. In order to estimate these parameters, we use a random variable $y_b$ which is 1 if wicket falls and 0 if it does. Assuming outcomes are independent

of deliveries, we get the likelihood function as follows:

$$\text{Likelihood} = \prod_{n=1} \Phi(-\alpha_0 - \alpha_1 b - \alpha_2 w)_b^y (1 - \Phi(-\alpha_0 - \alpha_1 b - \alpha_2 w))^{1-y_b}$$

We choose value of $\alpha_0, \alpha_1, \alpha_2$ which maximizes the value of log-likelihood for an inning.

## Runs

The probability of scoring runs $i$ for $i \epsilon \{1, 2, 3, 4, 5, 6\}$ is given by $q(i; b, w)$ for $b$ deliveries and $w$ wickets in hand. Here we will use an ordered probit model with an unobserved variable $i^*_{b,w}$ as

$$i^*_{b,w} = \beta_0 + \beta_1 b + \beta_2 w + \epsilon_{b,w}$$

where the $\beta$'s are constants and $\epsilon_{b,w} \sim N(0,1)$. The thresholds for runs are denoted by $\mu_0 \leq \mu_1 \leq \mu_2 \leq \mu_3 \leq \mu_4 \leq \mu_5$. The runs scored by batting team are i for $i = \{1, 2, 3, 4, 5\}$ if $\mu_{i-1} < i^*_{b,w} < \mu_i$, zero runs if $i^*_{b,w} < \mu_0$ and six runs if $i^*_{b,w} > \mu_5$.

$$q(i; b, w) = \begin{cases} \Phi(\mu_0 - \beta_0 - \beta_1 b - \beta_2 w), & \text{if } i = 0, \\ 1 - \Phi(\mu_5 - \beta_0 - \beta_1 b - \beta_2 w), & \text{if } i = 6, \\ \Phi(\mu_r - \beta_0 - \beta_1 b - \beta_2 w) & \\ \quad -\Phi(\mu_{r-1} - \beta_0 - \beta_1 b - \beta_2 w), & \text{otherwise.} \end{cases}$$

Using our knowledge of the game we can say:

1. $\beta_1 < 0$ as scoring accelerates as game moves ahead
2. $\beta_2 > 0$ scoring decreases in lower batting order In order to calculate the probability, we do the same as previous estimation and calculate the parameters which results in the maximum value of Likelihood.

## Function F

Using these parameters, we construct our cumulative distribution function $F(r; b, w)$