waRne - Putting cricket analytics in a spin

Contents

Abstract

The importance of reproducibility, and the related issue of open access to data, has received a lot of recent attention.^[why this not work] Momentum on these issues is gathering in the sports analytics community. While cricket is the second largest commercial sport in the world, unlike other popular international sports, there has been no mechanism for the public to access comprehensive statistics on players and teams. Expert commentary currently relies heavily on data that isn't made readily accessible and this produces an unnecessary barrier for the development of an inclusive sports analytics community.

Introduction

Data access is a key enabler for any analytics community. Most major sports have easy access to match statistics, for example nbastatR for NBA, Lahman for baseball, deuce for tennis and nflfastR for NFL. Through access to sports data and reproducible findings and metrics fans clubs and researchers are better able to understand the game, predict match outcomes and rate players. For example the way teams tackled 4th down decisions in the NFL has changed since (Romer 2006) seminial work. As teams have changed their 4th down decision making this allows follow up research such as (Yam and Lopez 2019) which looked at more granular data to see if this happens in practice.

By making data more accessible and more advanced metrics more accessible fans data journalism in sports has grown in recent years. For example in (Horowitz, Yurko, and Ventura 2017) has enabled EPA to enter popular discussion among fans.

Cricket is the second largest sport in the world. However, unfortunetaly there is no easy accessible way to access ball by ball data nor aggregated statistics of teams and players. Data while available on sites like espncricinfo are not in an easy to use form. For example, each match is listed on different webpages so hours upon hours of time would be required to copy and paste a single season, not to mention the added difficulty of linking players between games and competitions in different countries. Hence, there are significant logistical barriers for prospective fans and analysts studying the game, which stagnats understanding of cricket.

This paper describes the waRne package, the first to provide free and easy access to data for cricket for fans. Web scraping tools are avaiable for fans to easily scrape the play by play commentary data on espncricinfo. For the first time fans can evaluate their favourite teams and players and do so in a reproducible and accessible manner. We hope that this package can be used for fans to better understand the game, for teachers to use for interesting examples in class and for analysts in clubland who might not have access to ball by ball data.

Why cricket needs reproducibility

Data accessibility enables fans, analyts and researchers to better understand the game. Through being able to reproduce common popular metrics, visualisations and article findings.

Through being able to reproduce, fans are able to make accessible findings for others and importantly they are able to extend and grow concepts. Unfortuntely what we see through leading cricket analytics providers is a track record of confusing output for fans. This can lead to lower engagement and dismissial of cricket analytics.

For example in this series of tweets we see the narrative being pushed that Steve Smith is a good player vs pace bowling unfortunetly just a few months prior the same company and journalist published an article which had Steve Smith doing much worse against pace (balls above 140km/h). Unlike a similar sport baseball, fans have no easily accisible way of seeing if Steve Smith vs pace is a strength as alluded to in the original tweet, or a weakness like the same persons published online article. In comparsion, fans are able to get a breakdown of Mike Trout vs fastballs from using baseballr which provides access through statcast data from baseballsavant.

Unfortunetly this is just one of many examples whereby a relatively simple statistic is provided by media and fans have no mechanism to fact check. Fact checking is an important avenue for fans to not only engage and understand statistics, but having this mechanism also stops analytics people/companies from putting out misleading conclusions and findings.

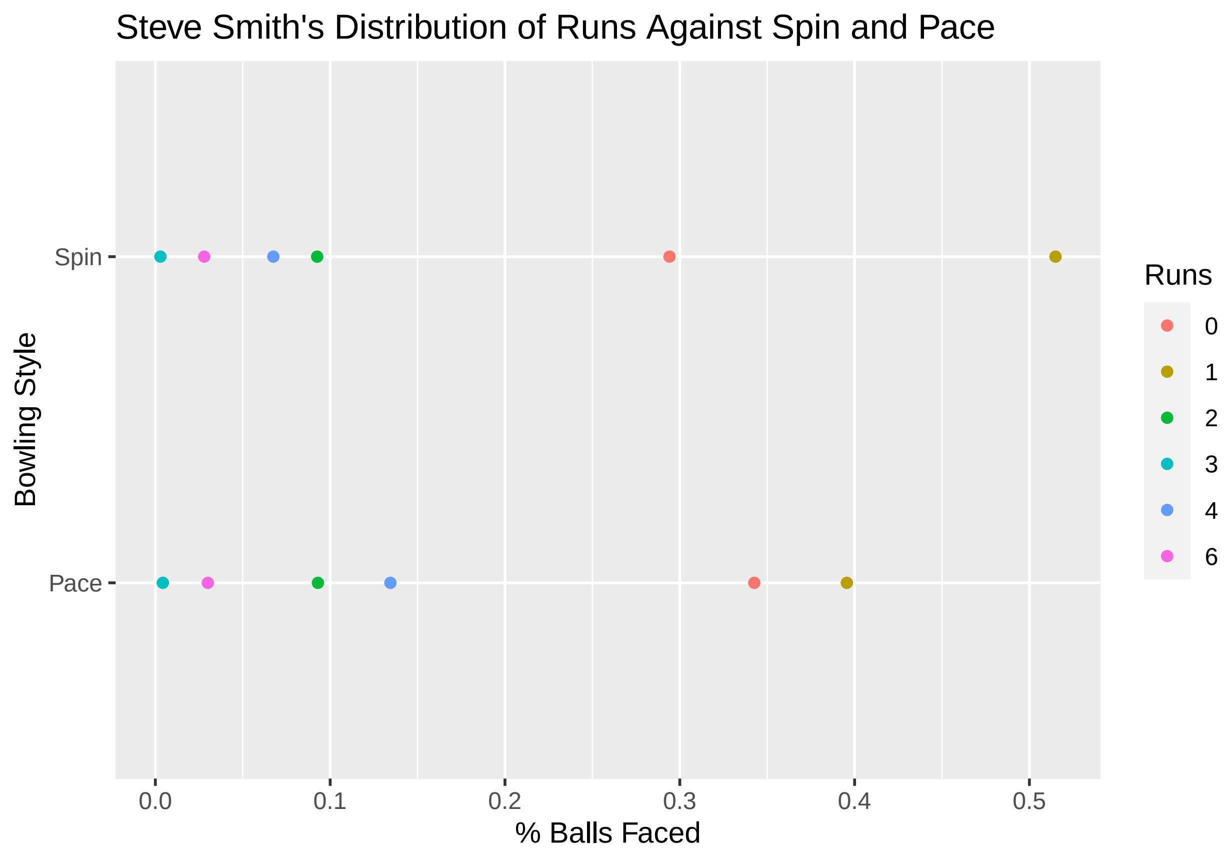
However though using waRne we are able to not only compare Steve Smiths performance vs bowling type but we are able to easily compare Steve Smiths performance vs bowling types with other players.

Steve Smith vs Bowling Type Chart

Fans of cricket and Australian cricket especially might be interested to see if this is a statistical qwerk of small samples or if Steve Smith generally does perform as an elite cricketer vs pace bowling. Teachers of statistics classes might use this as an interesting example to introduce tidyverse principals (Wickham et al. 2019).

```r
library(tidyverse)
cricjam_data_big %>%
    dplyr::filter(batsman_id == 267192,
                                !is.na(bowling_style)) %>%
    group_by(bowling_style, earned_runs) %>%
    summarise(total = n()) %>%
    group_by(bowling_style) %>%
    mutate(totals = sum(total),
                 percent = total/totals,
                 earned_runs = as.factor(earned_runs)) %>%
    ggplot(aes(y = bowling_style, x = percent, color = earned_runs))+
    geom_point() +
    labs(color = "Runs",
             x = "% Balls Faced",
             y = "Bowling Style",
             title = "Steve Smith's Distribution of Runs Against Spin and Pace")
```

```
## `summarise()` regrouping output by 'bowling_style' (override with `.groups` argument)
```



```
# insert above plots here
```

Using waRne to teach undergraduate statistics

We can also use waRne and other R packages as an easy way to engage students in learning statistical concepts.

For example using the above we can not only look at the answers graphically, but we can run a statistical test. A common classroom example for learning the binomial distribution is to ask if a coin is based given a proportion of heads and tails given a sample size (McElreath 2020). Instead of asking about coins, fans of sport and cricket might be interested to ask the question; after winning the coin toss, should a team decide to bat first or bat second (Kvam and Sokol 2004).

While a coin toss is a seemingly trivial example, like in most professional sports the winner of the coin toss gets to decide what to do. In cricket, the winner of the coin toss gets to decide if they want to bat first and thus set the total that the opposition needs to pass by a single run to win, or if they want to bowl first and thus chase down the total set. Deciding what to do after winning a coin toss has proved to be a popular media piece in recent years\footnote{https://www.espncricinfo.com/story/_/id/21489056/stuart-wark-cricket-move-away-coin-toss }\footnote{https://www.espncricinfo.com/video/clip/_/id/23230682%5D}

\footnote{https://www.espncricinfo.com/story/_/id/21489056/stuart-wark-cricket-move-away-coin-toss}

^[https://www.espncricinfo.com/story/_/id/20429499/lehmann-backs-scrapping-toss] ^[https://www.espncricinfo.com/story/_
much-does-losing-tosses-impact-visiting-teams] ^[https://www.forbes.com/sites/tristanlavalette/2018/08/20/are-cricket-matches-being-decided-by-the-luck-of-a-coin-toss/#735456837eff]

^[https://www.statsinsider.com.au/bbl/how-important-is-winning-the-toss-in-the-big-bash-league]

^[https://www.espncricinfo.com/story/_/id/18568387/tim-wigmore-how-batting-second-become-more-fruitful-more-popular]

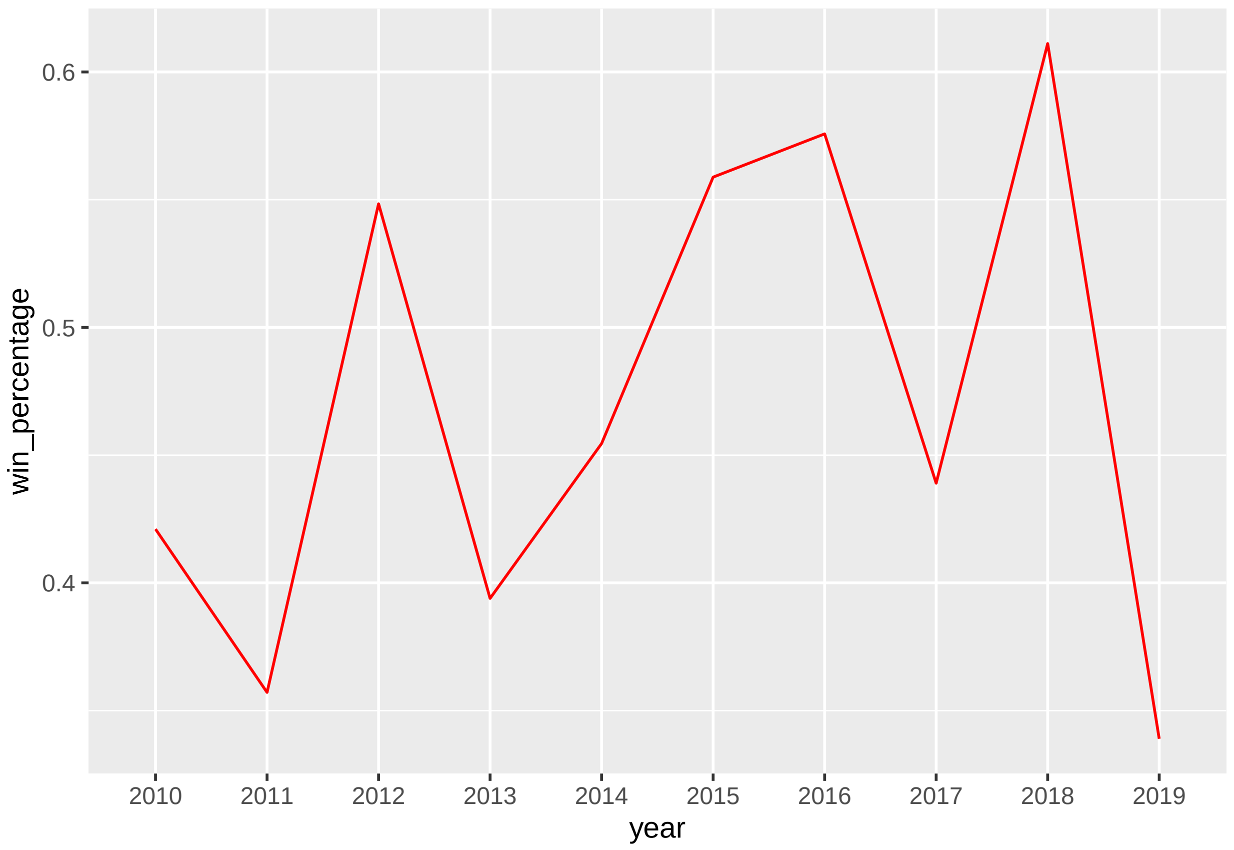
Fans and analysts of the game might want to make this decision based on a simple statistic, for example they might make it based on answer these questions instead.

- Do teams that bat second have a higher winning percentage than those who bat first?
- Is this consistent across leagues and levels of cricket?

```r
cricjam_data_big %>%
    dplyr::filter(tournament_name %in% c("Twenty20 Big Bash", "Big Bash League"),
                                inning == 2) %>%
    mutate(year = if_else(nchar(year)==4, year, as.numeric(str_sub(as.character(year), 1, 4)))) %>%
    group_by(match_id) %>%
    slice(1) %>%
    dplyr::select(batting_team_result, year) %>%
    mutate(binary_result = if_else(batting_team_result=="win", 1, 0)) %>%
    group_by(batting_team_result, year) %>%
    summarise(win_count = n()) %>%
    ungroup() %>%
    pivot_wider(names_from = batting_team_result, values_from = win_count) %>%
    rename(superover = 'super-over') %>%
    replace_na(list(superover=0)) %>%
    mutate(total_games = lose+win+superover,
             win_percentage = win/total_games,
             year = as.factor(year)) %>%
    ggplot(aes(x = year, y = win_percentage, group = 1)) +
    geom_line(color = 'red')
```

```
## Adding missing grouping variables: `match_id`
```

```
## `summarise()` regrouping output by 'batting_team_result' (override with `.groups` argument)
```



Instead of just looking at this graphically, fans of crickets and teachers of undergraduate statistics might use the dataset as a ''biased coin" example. So instead of asking, given a proportion of heads over $n$ tosses instructors could ask is a team better off batting first or second.

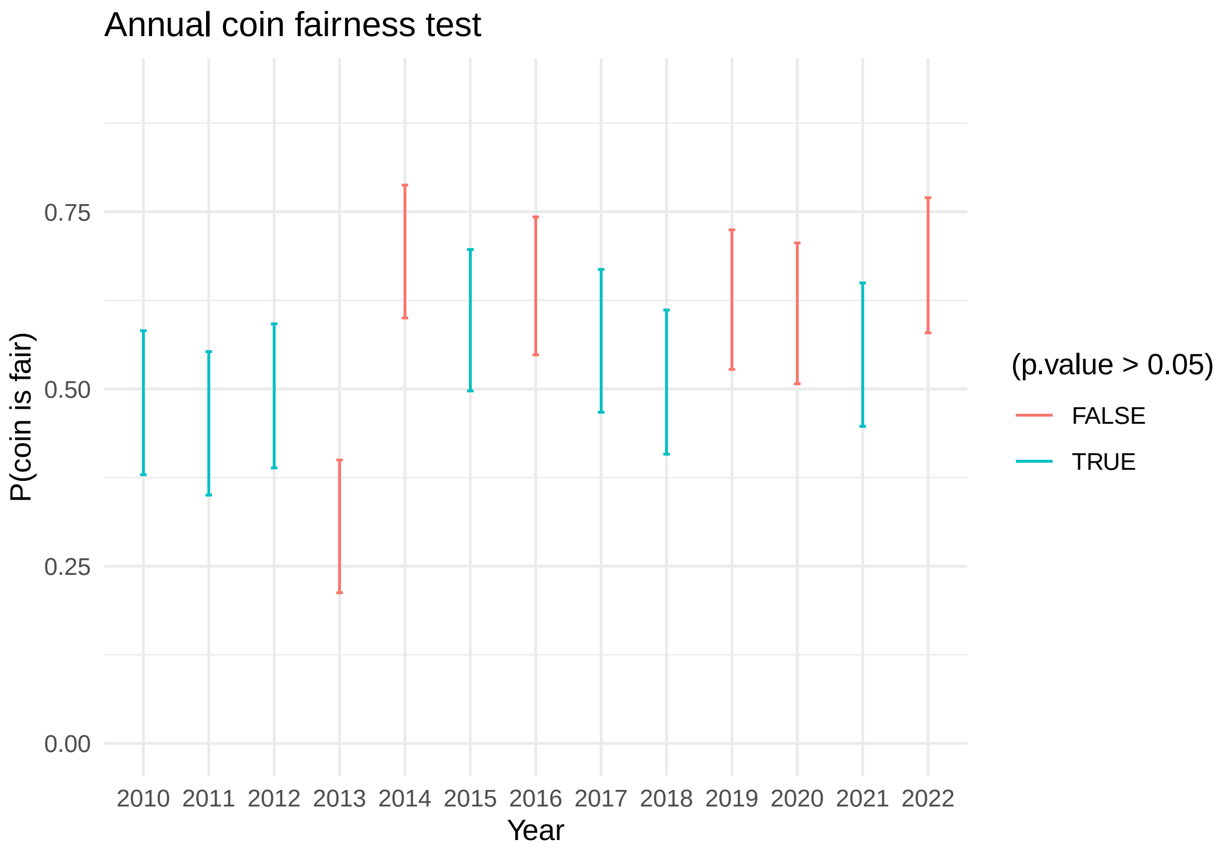
```
n<-900
p<-0.5
qbinom(c(0.025,0.975),size=n,prob=p)
```

```
## [1] 421 479
```

```
df<-data.frame(
             Year = c(2010L,2011L,2012L,2013L,2014L,
                      2015L,2016L,2017L,2018L,2019L,2020L,2021L,2022L),
  number.of.heads = c(52L,55L,51L,70L,30L,40L,35L,
                      43L,49L,37L,39L,45L,32L)
)

library(broom)
df %>%
  # create variables to pass to binom.test
  mutate(n = 100,
         x = n - number.of.heads,
         p = 0.5,
         alternative = "two.sided",
         conf.level = 0.95) %>%
  # Run a binom.test for each row/year
  mutate(result = pmap(list(x, n,p), binom.test)) %>%
  # Extract results from test into dataframe using broom::tidy
```

```r
  mutate(result = map(result, tidy)) %>%
  mutate(result = map(result, ~select(.x, p.value, conf.low, conf.high))) %>%
  unnest(cols = c(result)) %>%
  # plot the annual confidence intervals
  ggplot(aes(factor(Year), p.value)) +
    geom_errorbar(aes(ymin=conf.low, ymax=conf.high, color = (p.value > 0.05)), width=.1) +
  theme_minimal() +
  labs(y = "P(coin is fair)", x = "Year") +
  ggtitle("Annual coin fairness test")
```



James Stein Estimator - Why not a cricket example

https://bookdown.org/content/922/james-stein.html

Easier engagement of fans

To the surprise of many, there has not been a real push to engage fans in the analytics of the game of cricket. Unlike other sports which has seen a boom through the use of accessible data like Ice Hockey, NFL, Baseball and Basketball for example.

Without an easy accessible medium how can crickets version of an analytics community grow.

something something look towards how EPA has changed the way fans are engaged in NFL analytics - so maybe the change in run rate can be like EPA?

https://twitter.com/cricvizanalyst/status/1311260989267087361

Cool cricket examples

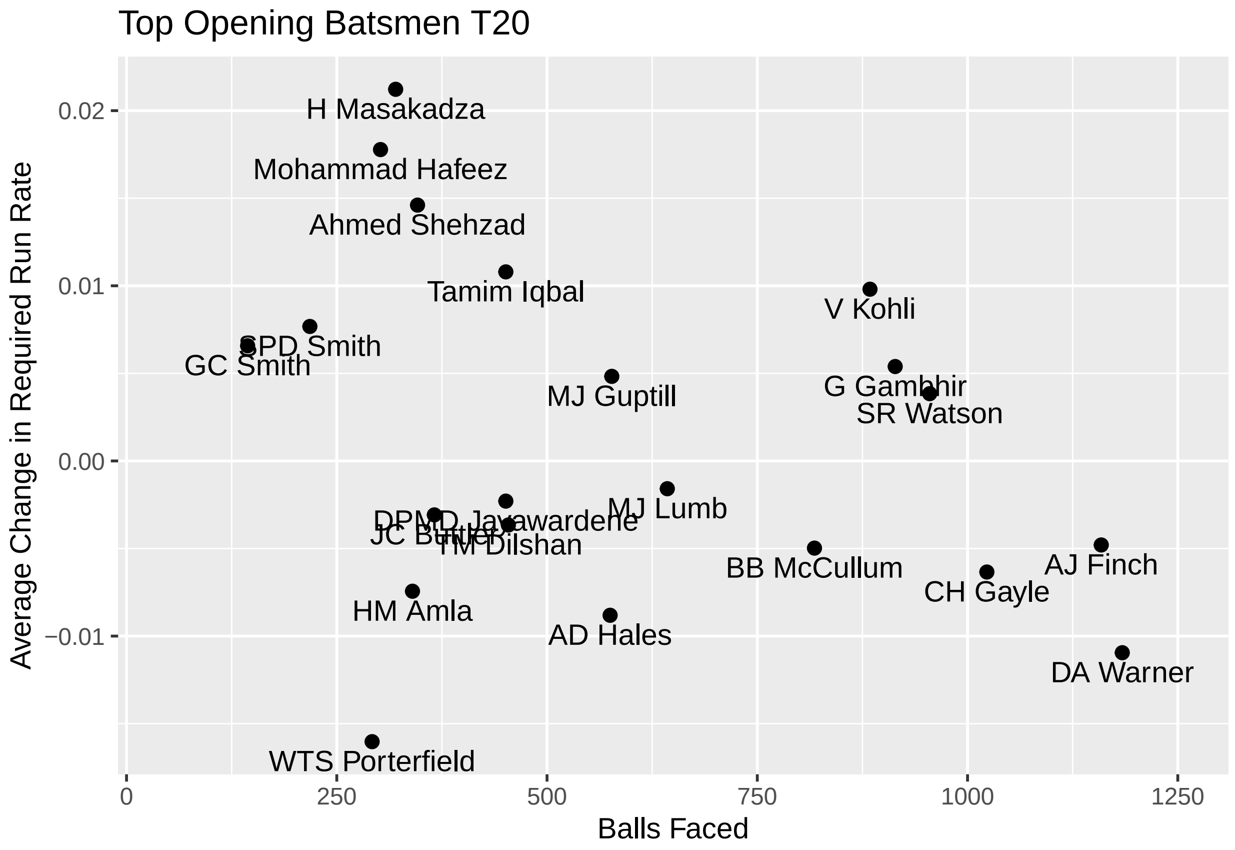
- Dhoni spin/pace/yorkr
    - shows off joins to player information and extracting information from play by play
- espncricinfo articles but instead of ipl do same tables for big bash
- recreate stats from espncricinfo like this page

Dhoni Dilemma

With this new dataset, we allow fans to put on their analyst thats and to be able to ask themselves, what exactly makes a good closer and what exactly is the Dhoni Dilemma. To understand the Dhoni Deilemma, we need to see some interesting things to do with the change in RRR.

Cricket is an obvious sort of sport, to win in the end you have to get one run more than your opposition. A commonly used statistic during the broadcast is the required run rate, for example if you need 120 to win off 20 overs your RRR is $6$ runs an over. What follows is then obvious is the chasing team wins if they are able to lower the RRR that is score quicker.

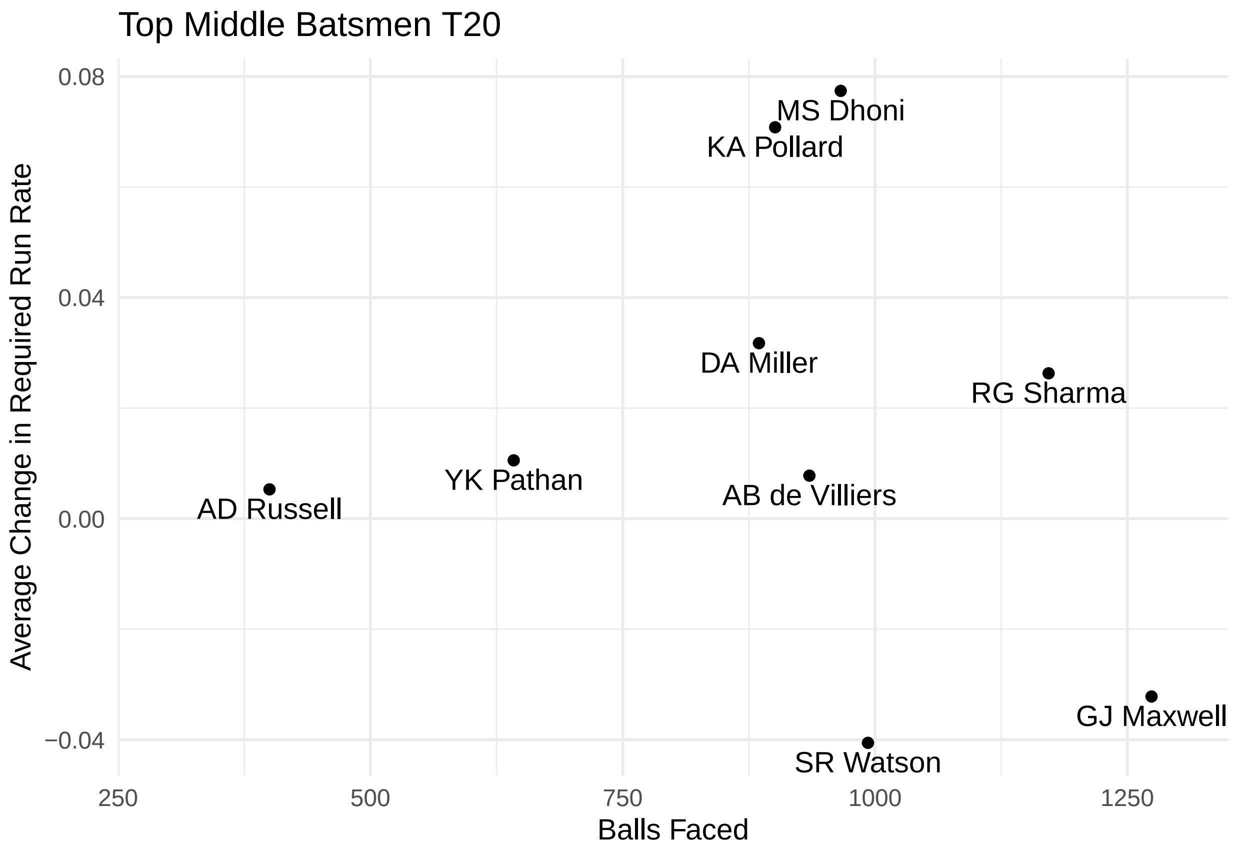
Espncricinfo did article on teams that win games get a higher proportion of their runs in the powerplay and first 10 overs. But proportion of team runs in T20 games doesn't really make sense as a batting metric as we are not just concerned with how many runs they get, but how quickly they get them. So we introduce a new way to measure batsman change in RRR.



```
## List of 93
##  $ line                    :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
##  $ legend.spacing.y          : NULL
##  $ legend.key                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size           : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.border              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

This graph makes sense as we can see some of the best openers in the world lower the required run rate. So what is the Dhoni Dilemma?

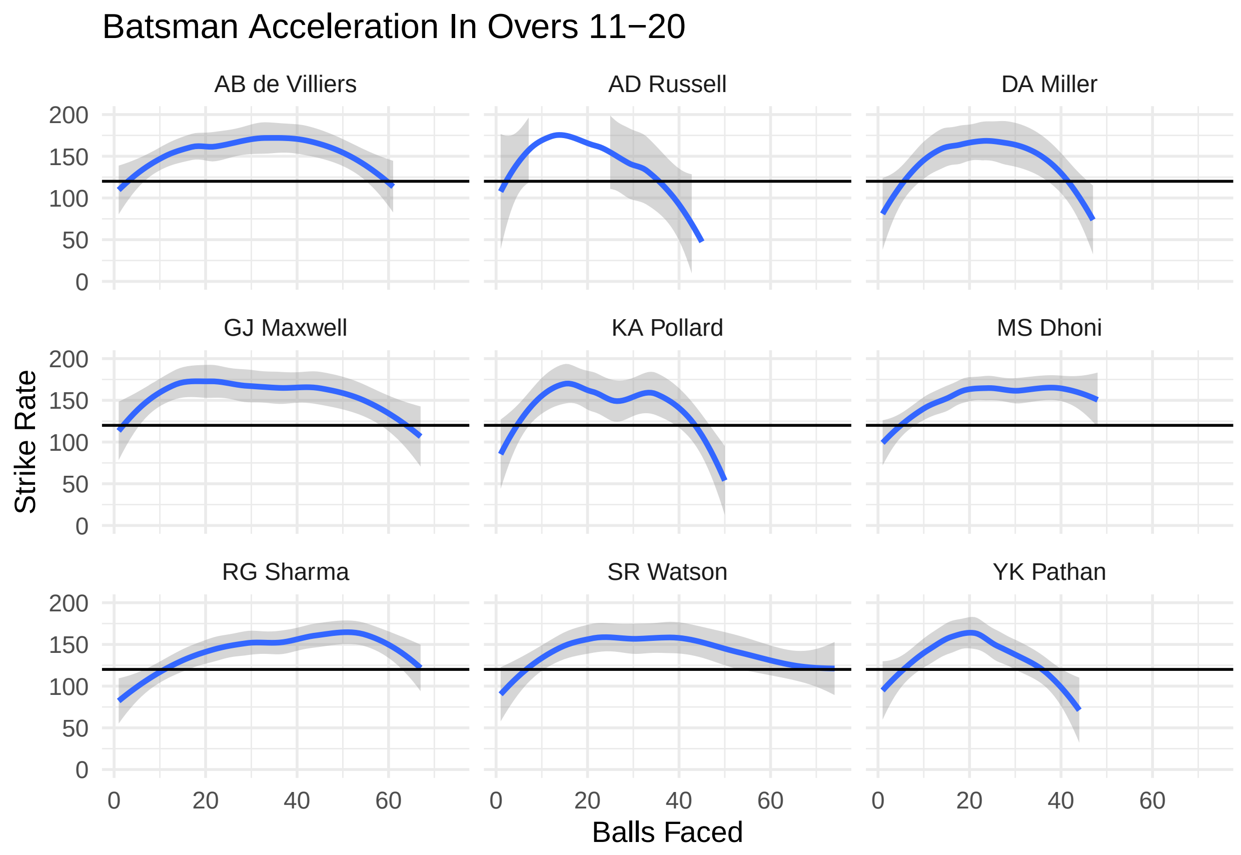
Dhoni is considered the greatest closing batsman in the world. He has pulled of many spectacular saves, and has T20 and T20I strike rates after 14 years in the game. But how does his required run rate compare to other middle order and closing batsman



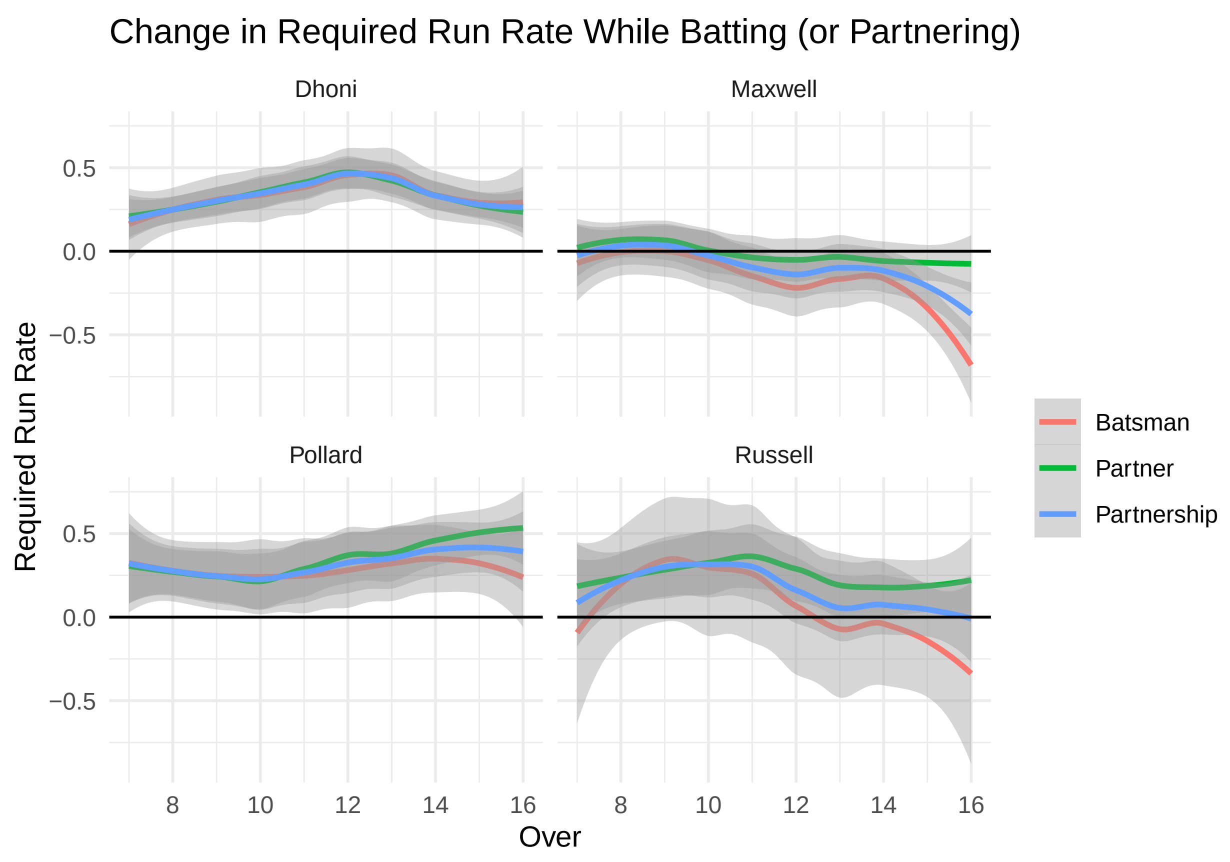
Dhoni is renowed for being one of the worlds best closers but we can see it seems as though in the middle he is letting the game get out of hand. How can one of the worlds best closers let the run rate go up so consistently in the middle overs?

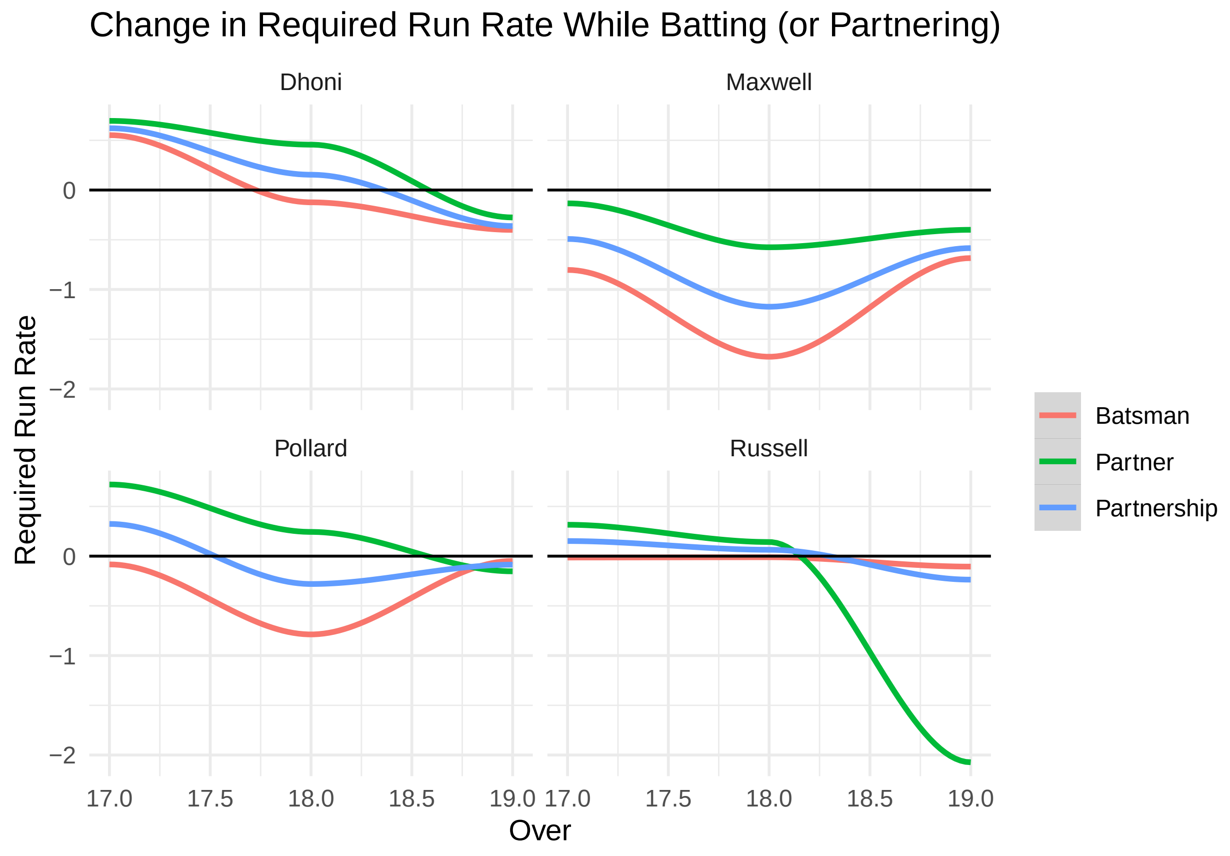
One possible explanation comes from looking at the acceleration of Dhoni compared to these other middle order batsmen. Dhoni comes in usually around overs 11/12, and from the plot below we can see that he is a slow starter, it takes several overs for him to get his strike rate up, which consistently stays high throughout the game even when other batsmen's begins to drop off.

```
## # A tibble: 18 x 3
##    bucketed_over total `mean(total)`
##            <dbl> <int>         <dbl>
##  1             3     1          11.5
##  2             4     1          11.5
##  3             5     4          11.5
##  4             6     6          11.5
##  5             7     6          11.5
##  6             8     4          11.5
##   7             9    11           11.5
##   8            10    12           11.5
##   9            11    21           11.5
##  10            12    19           11.5
##  11            13    28           11.5
##  12            14    18           11.5
##  13            15    29           11.5
##  14            16    11           11.5
##  15            17    19           11.5
##  16            18     5           11.5
##  17            19     7           11.5
##  18            20     5           11.5
```



Another is the effort of Dhoni's partners in moving the game along while Dhoni is up


Change in Required Run Rate While Batting (or Partnering)
Dhoni
Maxwell
Pollard
Russell
Required Run Rate
0.5
0.0
−0.5
8
10
12
14
16
Over
Batsman
Partner
Partnership


Change in Required Run Rate While Batting (or Partnering)
Dhoni
Maxwell
Pollard
Russell
Required Run Rate
0
−1
−2
17.0
17.5
18.0
18.5
19.0
Over
Batsman
Partner
Partnership

What we can tell is that in second inning chases, the required run rate for Dhoni is lower than our other top middle order batsmen, contrasting our inital findings. One would assume that this was due to Dhoni's partners, but it appears he still performs at an elite level

Horowitz, Maksim, Ron Yurko, and Samuel L Ventura. 2017. "NflscrapR: Compiling the Nfl Play-by-Play Api for Easy Use in R." *URL Https://Github. Com/Maksimhorowitz/nflscrapR, R Package Version* 1 (0).

Kvam, Paul H, and Joel Sokol. 2004. "Teaching Statistics with Sports Examples." *INFORMS Transactions on Education* 5 (1): 75–87.

McElreath, Richard. 2020. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC press.

Romer, David. 2006. "Do Firms Maximize? Evidence from Professional Football." *Journal of Political Economy* 114 (2): 340–65.

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.

Yam, Derrick R, and Michael J Lopez. 2019. "What Was Lost? A Causal Estimate of Fourth down Behavior in the National Football League." *Journal of Sports Analytics* 5 (3): 153–67.