*Who looks outside, dreams; who looks inside, awakes.*  
*Show me a sane man and I will cure him for you.*

Carl Jung

*We’re made of star stuff. We are a way for the cosmos to know itself.*  
*If you want to make an apple pie from scratch, you must first create the universe.*

Carl Sagan

**Introduction**

The *biggest nag* in the collective psyche of cricketing fraternity these days, is whether Virat Kohli has surpassed Sachin Tendulkar. This question has been troubling cricket lovers the world over and particularly in India, for quite a while. This nagging question has only grown stronger with Kohli’s 41st ODI century and with Michael Vaughan bestowing the *GOAT* title to Virat Kohli for ODI cricket. Hence, I decided to do my bit in addressing this, by doing analysis of Kohli’s and Tendulkar’s performance in ODI cricket. I also wanted to address the the best among the cricketing idols of India in Test cricket, namely Sunil Gavaskar, Sachin Tendulkar and Virat Kohli. Hence this post has 2 parts

1. Analysis of Tendulkar, Gavaskar and Kohli in Test cricket
2. Analysis of Tendulkar and Kohli in ODIs

In this post, I analyze the performances of these titans in Test and ODI cricket using my R package [**cricketr**](https://cran.r-project.org/web/packages/cricketr/index.html). While some may feel that comparisons are not possible as these batsmen are from different eras. To some extent this is true. I would give some leeway to Gavaskar as he had to bat in a pre-helmet era. But with Tendulkar and Kohli a fair and objective comparison is possible. There were pre-eminient bowlers in the times of Tendulkar as there are now.

From the analysis below, it can be seen that Tendulkar to of everybody else in Test cricket. However it must be noted that Tendulkar’s performance deteriorated towards the end of his career. Such was not the case with Gavaskar. Kohli has some catching up to do and he still has a lot of Test cricket in him.

In ODI Kohli can be seen to pulling ahead of Tendulkar in several aspects.

My R package **cricketr** can be installed directly from CRAN and you can use it analyze cricketers.

This package uses the statistics info available in [ESPN Cricinfo Statsguru](http://stats.espncricinfo.com/ci/engine/stats/index.html). The current version of this package supports all formats of the game including Test, ODI and Twenty20 versions.

You should be able to install the package from GitHub and use the many functions available in the package. Please mindful of the [ESPN Cricinfo Terms of Use](http://www.espncricinfo.com/ci/content/site/company/terms_use.html)

Take a look at my short video tutorial on my R package cricketr on Youtube – [R package cricketr – A short tutorial](https://www.youtube.com/edit?o=U&video_id=q9uMPFVsXsI)

Do check out my interactive Shiny app implementation using the cricketr package – [Sixer – R package cricketr’s new Shiny avatar](https://tvganesh.shinyapps.io/Sixer)

**Note 1**: If you would like to do a similar analysis for a different set of batsman and bowlers, you can clone/download my skeleton [cricketr template](https://github.com/tvganesh/cricketr-template" \t "_blank)from Github (which is the R Markdown file I have used for the analysis below).

**Note 2**: I sprinkle the charts with my observations. Feel free to look at them more closely and come to your conclusions.

**Important note**: Do check out the python avatar of cricketr, ‘cricpy’ in my post [Introducing cricpy:A python package to analyze performances of cricketers](https://gigadom.in/2018/10/28/introducing-cricpya-python-package-to-analyze-performances-of-cricketrs/)

* This post is also available at Rpubs [Clash of the Titans](http://rpubs.com/tvganesh/476500)
* You can download this in PDF format at [Clash of the Titans](https://drive.google.com/open?id=1kT2f0QzK8UpAmCjMQ9KYqUv6LPTPOx7k)
* You can download this R Markdown file from Github at [Clash of titans](https://github.com/tvganesh/clash-of-titans)

**1 Load the cricketr package**

if (!require("cricketr")){

install.packages("cricketr",lib = "c:/test")

}

library(cricketr)

**A Test cricket  – Analysis of Gavaskar, Tendulkar and Kohli**

**2. Get player data**

tendulkar <- getPlayerData(35320,dir=".",file="tendulkar.csv",type="batting")

kohli <- getPlayerData(253802,dir=".",file="kohli.csv",type="batting")

gavaskar <- getPlayerData(28794,dir=".",file="gavaskar.csv",type="batting")

**3a. Basic analyses for Tendulkar**

par(mfrow=c(1,3))

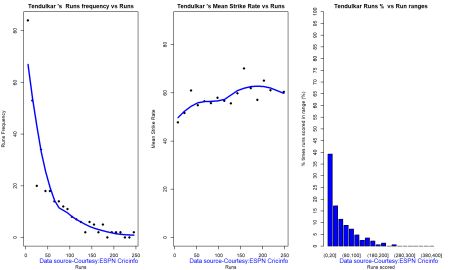
par(mar=c(4,4,2,2))

batsmanRunsFreqPerf("./tendulkar.csv","Tendulkar")

batsmanMeanStrikeRate("./tendulkar.csv","Tendulkar")

batsmanRunsRanges("./tendulkar.csv","Tendulkar")

dev.off()



**3b Basic analyses for Kohli**

par(mfrow=c(1,3))

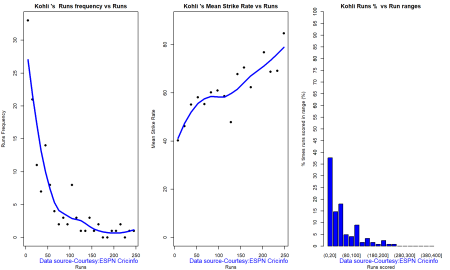
par(mar=c(4,4,2,2))

batsmanRunsFreqPerf("./kohli.csv","Kohli")

batsmanMeanStrikeRate("./kohli.csv","Kohli")

batsmanRunsRanges("./kohli.csv","Kohli")

dev.off()



**3c Basic analyses for Gavaskar**

par(mfrow=c(1,3))

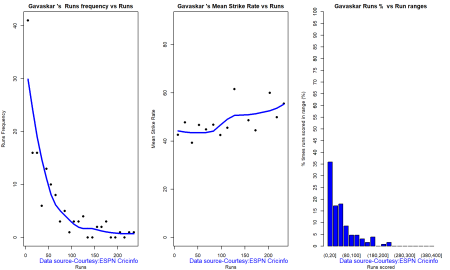
par(mar=c(4,4,2,2))

batsmanRunsFreqPerf("./gavaskar.csv","Gavaskar")

batsmanMeanStrikeRate("./gavaskar.csv","Gavaskar")

batsmanRunsRanges("./gavaskar.csv","Gavaskar")

dev.off()



**4a.More analyses for Tendulkar**

It can be seen that Tendulkar and Gavaskar has been bowled more often than Kohli. Also Kohli does not have as many sixes in Test cricket as Tendulkar and Gavaskar

par(mfrow=c(1,3))

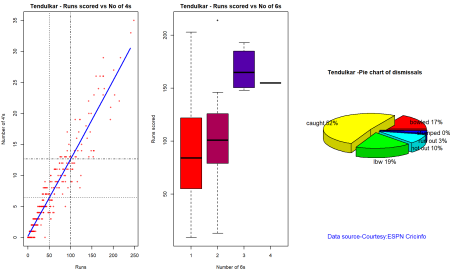
par(mar=c(4,4,2,2))

batsman4s("./tendulkar.csv","Tendulkar")

batsman6s("./tendulkar.csv","Tendulkar")

batsmanDismissals("./tendulkar.csv","Tendulkar")

dev.off()



**4b. More analyses for Kohli**

par(mfrow=c(1,3))

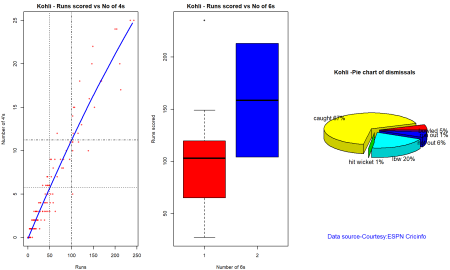
par(mar=c(4,4,2,2))

batsman4s("./kohli.csv","Kohli")

batsman6s("./kohli.csv","Kohli")

batsmanDismissals("./kohli.csv","Kohli")

dev.off()



**4c More analyses for Gavaskar**

par(mfrow=c(1,3))

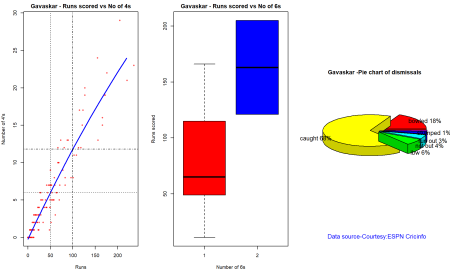
par(mar=c(4,4,2,2))

batsman4s("./gavaskar.csv","Gavaskar")

batsman6s("./gavaskar.csv","Gavaskar")

batsmanDismissals("./gavaskar.csv","Gavaskar")

dev.off()



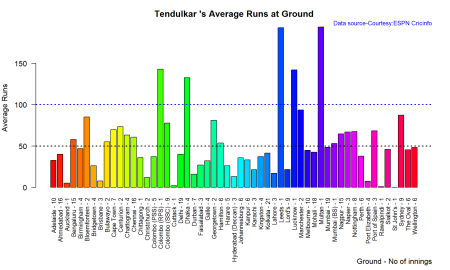
**5 Performance of batsmen on different grounds**

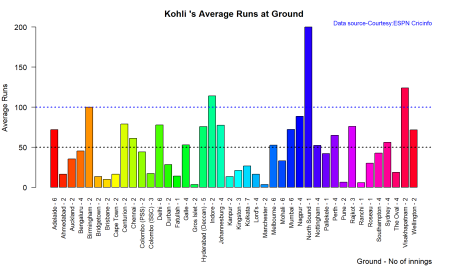
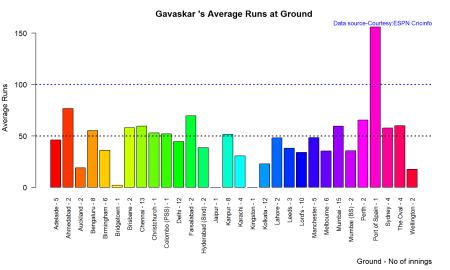
par(mar=c(4,4,2,2))

batsmanAvgRunsGround("./tendulkar.csv","Tendulkar")

batsmanAvgRunsGround("./kohli.csv","Kohli")

batsmanAvgRunsGround("./gavaskar.csv","Gavaskar")



a  


#dev.off()

**6. Performance if batsmen against different Opposition**

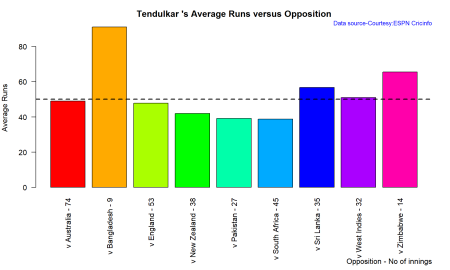
1. Tendulkar averages 50 against the following countries – Australia, Bangladesh, England, Sri Lanka, West Indies and Zimbabwe
2. Kohli average almost 50 against all the nations he has played – Australia, Bangladesh, England, New Zealand, Sri Lanka and West Indies
3. Gavaskar averages 50 against Australia, Pakistan, West Indies, Sri Lanka

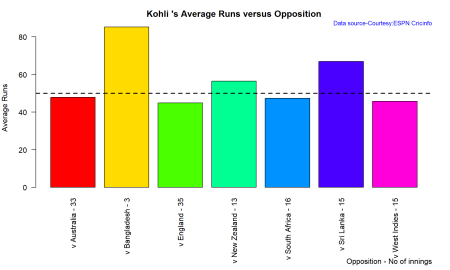
par(mar=c(4,4,2,2))

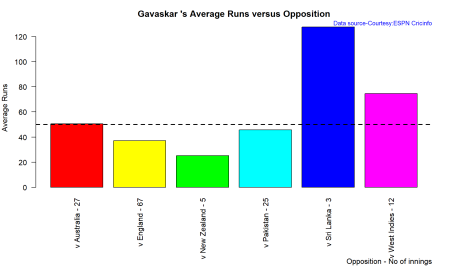
batsmanAvgRunsOpposition("./tendulkar.csv","Tendulkar")

batsmanAvgRunsOpposition("./kohli.csv","Kohli")

batsmanAvgRunsOpposition("./gavaskar.csv","Gavaskar")







**7. Get player data special**

This is required for the next 2 function calls

tendulkarsp <- getPlayerDataSp(35320,tdir=".",tfile="tendulkarsp.csv",ttype="batting")

kohlisp <- getPlayerDataSp(253802,tdir=".",tfile="kohlisp.csv",ttype="batting")

gavaskarsp <- getPlayerDataSp(28794,tdir=".",tfile="gavaskarsp.csv",ttype="batting")

#dev.off()

**8 Get contribution of batsmen in matches won and lost**

Kohli contribution has had an equal contribution in won and lost matches. Tendulkar’s runs seem to have not helped in winning as much as only 50% of matches he has played have been won

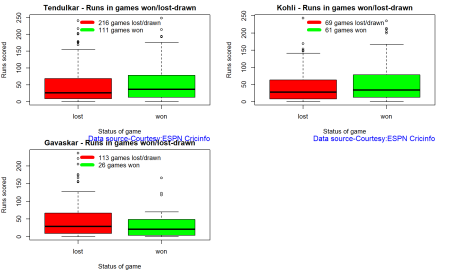
par(mfrow=c(2,2))

par(mar=c(4,4,2,2))

batsmanContributionWonLost("tendulkarsp.csv","Tendulkar")

batsmanContributionWonLost("./kohlisp.csv","Kohli")

batsmanContributionWonLost("./gavaskarsp.csv","Gavaskar")

a

**9 Performance of batsmen at home and overseas**

The boxplots show that Kohli performs better overseas than at home. The 3rd quartile is higher, though the median seems to lower overseas. For Tendulkar the performance is similar both ways. Gavaskar’s median runs scored overseas is higher.

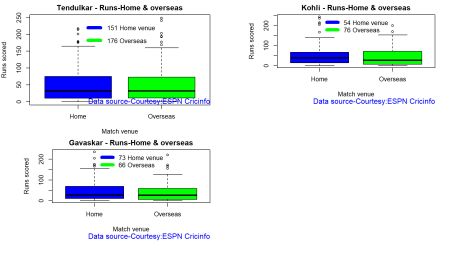
par(mfrow=c(2,2))

par(mar=c(4,4,2,2))

batsmanPerfHomeAway("tendulkarsp.csv","Tendulkar")

batsmanPerfHomeAway("./kohlisp.csv","Kohli")

batsmanPerfHomeAway("./gavaskarsp.csv","Gavaskar")



**10. Moving average of runs**

Gavaskar’s moving average was very good at the time of his retirement. Kohli seems to be going very strong. Tendulkar’s performance shows signs of deterioration around the time of his retirement.

par(mfrow=c(2,2))

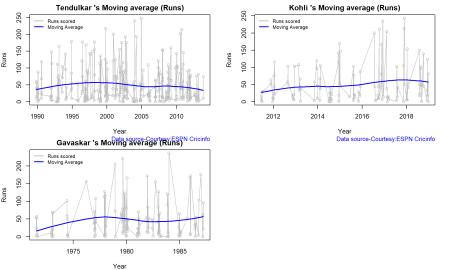
par(mar=c(4,4,2,2))

batsmanMovingAverage("./tendulkar.csv","Tendulkar")

batsmanMovingAverage("./kohli.csv","Kohli")

batsmanMovingAverage("./gavaskar.csv","Gavaskar")

#dev.off()



**11 Boxplot and histogram of runs**

Kohli has a marginally higher average (50.69) than Tendulkar (48.65) while Gavaskar 46. The median runs are same for Tendulkar and Kohli at 32

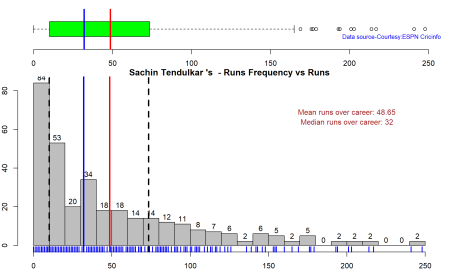
par(mfrow=c(2,2))

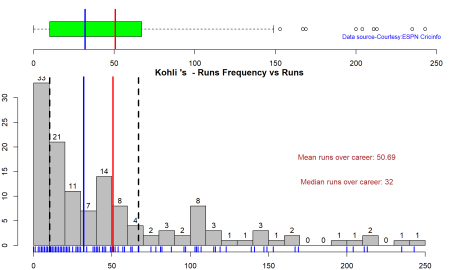
par(mar=c(4,4,2,2))

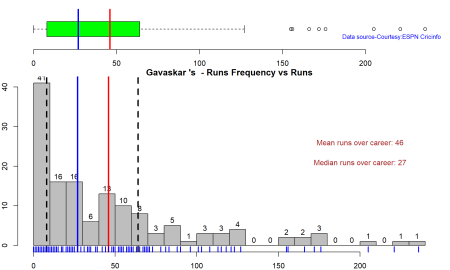
batsmanPerfBoxHist("./tendulkar.csv","Sachin Tendulkar")

batsmanPerfBoxHist("./kohli.csv","Kohli")

batsmanPerfBoxHist("./gavaskar.csv","Gavaskar")







**12 Cumulative average Runs for batsmen**

Looking at the cumulative average runs we can see a gradual drop in the cumulative average for Tendulkar while Kohli and Gavaskar’s performance seems to be getting better

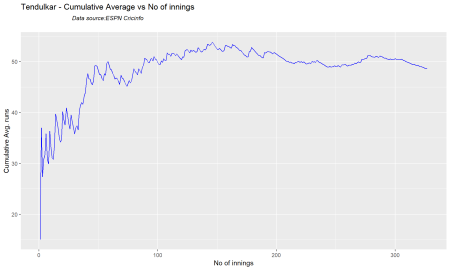
par(mfrow=c(2,2))

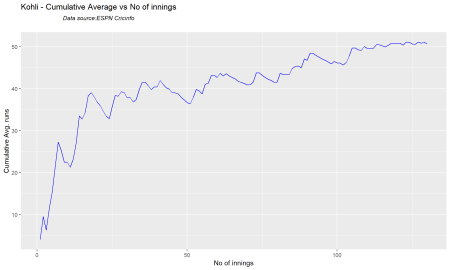
par(mar=c(4,4,2,2))

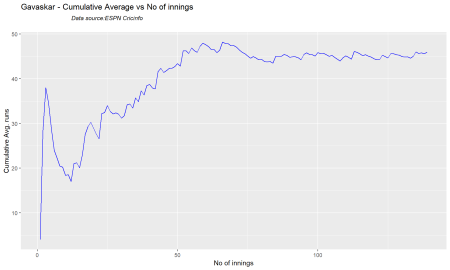
batsmanCumulativeAverageRuns("./tendulkar.csv","Tendulkar")

batsmanCumulativeAverageRuns("./kohli.csv","Kohli")

batsmanCumulativeAverageRuns("./gavaskar.csv","Gavaskar")







**13. Cumulative average strike rate of batsmen**

Tendulkar’s strike rate is better than Kohli and Gavaskar

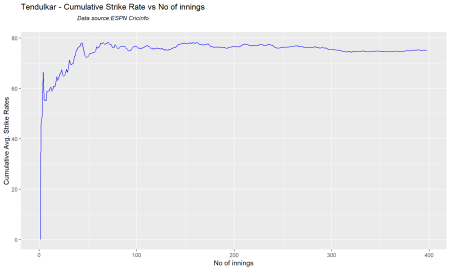
par(mfrow=c(2,2))

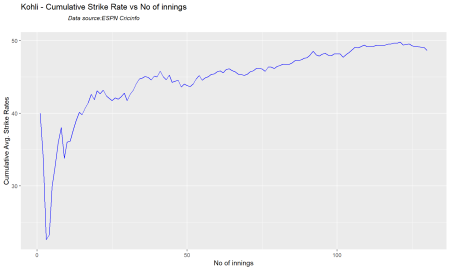
par(mar=c(4,4,2,2))

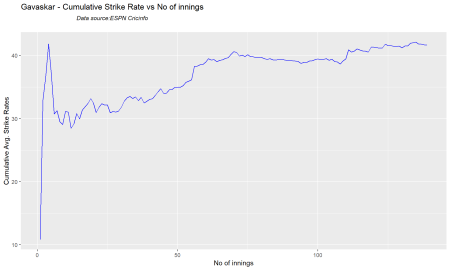
batsmanCumulativeStrikeRate("./tendulkar.csv","Tendulkar")

batsmanCumulativeStrikeRate("./kohli.csv","Kohli")

batsmanCumulativeStrikeRate("./gavaskar.csv","Gavaskar")







**14 Performance forecast of batsmen**

The forecasted performance for Kohli and Gavaskar is higher than that of Tendulkar

par(mfrow=c(2,2))

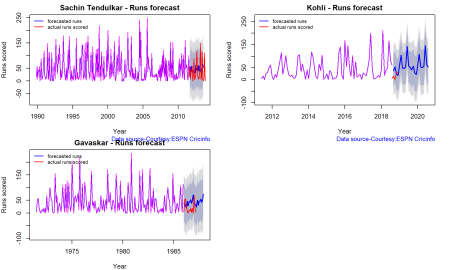
par(mar=c(4,4,2,2))

batsmanPerfForecast("./tendulkar.csv","Sachin Tendulkar")

batsmanPerfForecast("./kohli.csv","Kohli")

batsmanPerfForecast("./gavaskar.csv","Gavaskar")

#dev.off()



**15. Relative strike rate of batsmen**

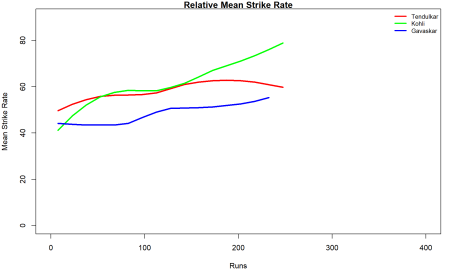
par(mar=c(4,4,2,2))

frames <- list("./tendulkar.csv","./kohli.csv","gavaskar.csv")

names <- list("Tendulkar","Kohli","Gavaskar")

relativeBatsmanSR(frames,names)

#dev.off()



**16. Relative Runs frequency of batsmen**

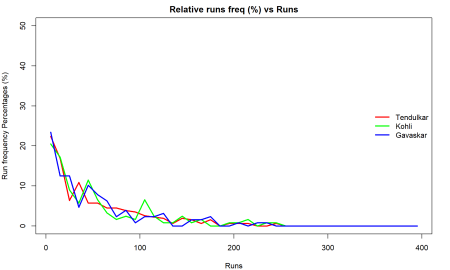
par(mar=c(4,4,2,2))

frames <- list("./tendulkar.csv","./kohli.csv","gavaskar.csv")

names <- list("Tendulkar","Kohli","Gavaskar")

relativeRunsFreqPerf(frames,names)

#dev.off()



**17. Relative cumulative average runs of batsmen**

Tendulkar leads the way here, but it can be seem Kohli catching up.

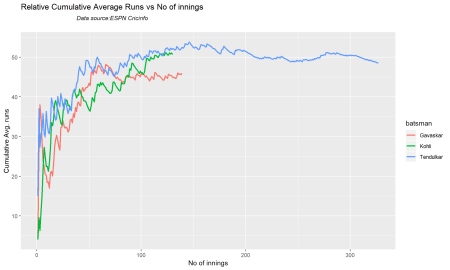
par(mar=c(4,4,2,2))

frames <- list("./tendulkar.csv","./kohli.csv","gavaskar.csv")

names <- list("Tendulkar","Kohli","Gavaskar")

relativeBatsmanCumulativeAvgRuns(frames,names)

#dev.off()



**18. Relative cumulative average strike rate**

Tendulkar has better strike rate than the other two.

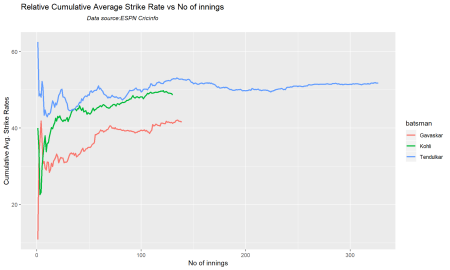
par(mar=c(4,4,2,2))

frames <- list("./tendulkar.csv","./kohli.csv","gavaskar.csv")

names <- list("Tendulkar","Kohli","Gavaskar")

relativeBatsmanCumulativeStrikeRate(frames,names)

#dev.off()



**19. Check batsman in form**

As in the moving average and performance forecast and cumulative average runs, Kohli and Gavaskar are in-form while Tendulkar was out-of-form towards the end.

checkBatsmanInForm("./tendulkar.csv","Sachin Tendulkar")

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Form status of Sachin Tendulkar \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\n\n Population size: 294 Mean of population: 50.48 \n Sample size: 33 Mean of sample: 32.42 SD of

sample: 29.8 \n\n Null hypothesis H0 : Sachin Tendulkar 's sample average is within 95% confidence interval

of population average\n Alternative hypothesis Ha : Sachin Tendulkar 's sample average is below

the 95% confidence interval of population average\n\n

Sachin Tendulkar 's Form Status: Out-of-Form because the p value: 0.000713 is less than alpha= 0.05 \n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

checkBatsmanInForm("./kohli.csv","Kohli")

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Form status of Kohli \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n Population size: 117

Mean of population: 50.35 \n Sample size: 13 Mean of sample: 53.77 SD of sample: 46.15 \n\n Null

hypothesis H0 : Kohli 's sample average is within 95% confidence interval of population average\n

Alternative hypothesis Ha : Kohli 's sample average is below the 95% confidence interval of population

average\n\n Kohli 's Form Status: In-Form because the p value: 0.603244 is greater than alpha= 0.05 \n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

checkBatsmanInForm("./gavaskar.csv","Gavaskar")

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Form status of Gavaskar \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n

Population size: 125 Mean of population: 44.67 \n Sample size: 14 Mean of sample: 57.86 SD of sample:

58.55 \n\n Null hypothesis H0 : Gavaskar 's sample average is within 95% confidence interval of population

average\n Alternative hypothesis Ha : Gavaskar 's sample average is below the 95% confidence interval of

population average\n\n Gavaskar 's Form Status: In-Form because the p value: 0.793276 is greater

than alpha= 0.05 \n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

#dev.off()

**20. Performance 3D**

A 3D regression plane is fitted between the the Balls faced, Minutes at crease and Runs scored

par(mfrow=c(2,2))

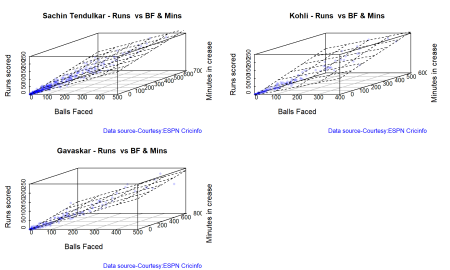
par(mar=c(4,4,2,2))

battingPerf3d("./tendulkar.csv","Sachin Tendulkar")

battingPerf3d("./kohli.csv","Kohli")

battingPerf3d("./gavaskar.csv","Gavaskar")

#dev.off()



**20. Runs likelihood**

This functions computes the K-Means and determines the runs the batsmen are likely to score.

par(mar=c(4,4,2,2))

batsmanRunsLikelihood("./tendulkar.csv","Tendulkar")

## Summary of Tendulkar 's runs scoring likelihood

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##

## There is a 16.51 % likelihood that Tendulkar will make 139 Runs in 251 balls over 353 Minutes

## There is a 25.08 % likelihood that Tendulkar will make 66 Runs in 122 balls over 167 Minutes

## There is a 58.41 % likelihood that Tendulkar will make 16 Runs in 31 balls over 44 Minutes

batsmanRunsLikelihood("./kohli.csv","Kohli")

## Summary of Kohli 's runs scoring likelihood

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##

## There is a 20 % likelihood that Kohli will make 143 Runs in 232 balls over 330 Minutes

## There is a 33.85 % likelihood that Kohli will make 51 Runs in 92 balls over 127 Minutes

## There is a 46.15 % likelihood that Kohli will make 11 Runs in 24 balls over 31 Minutes

batsmanRunsLikelihood("./gavaskar.csv","Gavaskar")

## Summary of Gavaskar 's runs scoring likelihood

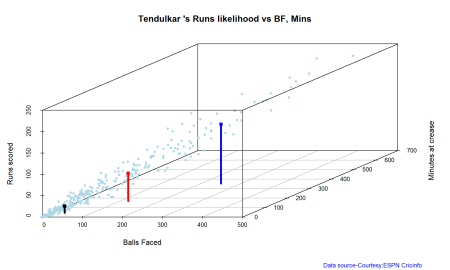
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

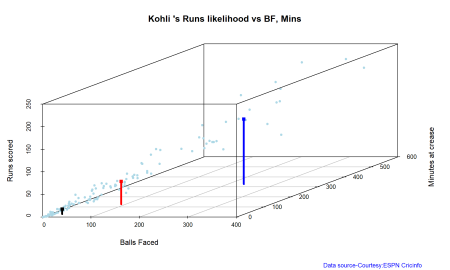
##

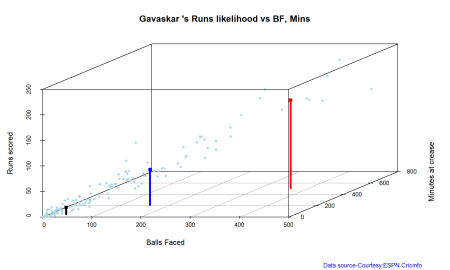
## There is a 33.81 % likelihood that Gavaskar will make 69 Runs in 159 balls over 214 Minutes

## There is a 8.63 % likelihood that Gavaskar will make 172 Runs in 364 balls over 506 Minutes

## There is a 57.55 % likelihood that Gavaskar will make 13 Runs in 35 balls over 48 Minutes







**21. Predict runs for a random combination of Balls faced and runs scored**

BF <- seq( 10, 400,length=15)

Mins <- seq(30,600,length=15)

newDF <- data.frame(BF,Mins)

tendulkar <- batsmanRunsPredict("./tendulkar.csv","Tendulkar",newdataframe=newDF)

kohli <- batsmanRunsPredict("./kohli.csv","Kohli",newdataframe=newDF)

gavaskar <- batsmanRunsPredict("./gavaskar.csv","Gavaskar",newdataframe=newDF)

batsmen <-cbind(round(tendulkar$Runs),round(kohli$Runs),round(gavaskar$Runs))

colnames(batsmen) <- c("Tendulkar","Kohli","Gavaskar")

newDF <- data.frame(round(newDF$BF),round(newDF$Mins))

colnames(newDF) <- c("BallsFaced","MinsAtCrease")

predictedRuns <- cbind(newDF,batsmen)

predictedRuns

## BallsFaced MinsAtCrease Tendulkar Kohli Gavaskar

## 1 10 30 7 6 4

## 2 38 71 23 24 17

## 3 66 111 39 42 30

## 4 94 152 54 60 43

## 5 121 193 70 78 56

## 6 149 234 86 96 69

## 7 177 274 102 114 82

## 8 205 315 118 132 95

## 9 233 356 134 150 108

## 10 261 396 150 168 121

## 11 289 437 165 186 134

## 12 316 478 181 204 147

## 13 344 519 197 222 160

## 14 372 559 213 240 173

## 15 400 600 229 258 186

#dev.off()

**Key findings**

1. Kohli has a marginally higher average than Tendulkar
2. Tendulkar has the best strike rate of all the 3.
3. The cumulative average runs and the performance forecast for Kohli and Gavaskar show an improving trend, while Tendulkar’s numbers deteriorate towards the end of his career
4. Kohli is fast catching up Tendulkar on cumulative average runs vs innings in career.

**B ODI Cricket – Analysis of Tendulkar and Kohli**

The functions below get the ODI data for Tendulkar and Kohli as CSV files so that the analyses can be done

**22 Get player data for ODIs**

tendulkarOD <- getPlayerDataOD(35320,dir=".",file="tendulkarOD.csv",type="batting")

kohliOD <- getPlayerDataOD(253802,dir=".",file="kohliOD.csv",type="batting")

#dev.off()

**23a Basic performance of Tendulkar in ODI**

par(mfrow=c(3,2))

par(mar=c(4,4,2,2))

batsmanRunsFreqPerf("./tendulkarOD.csv","Tendulkar")

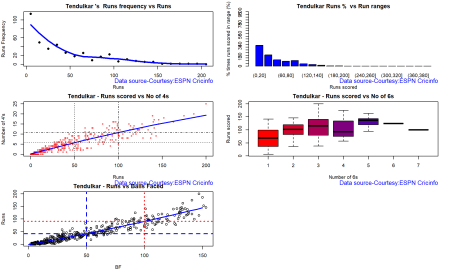
batsmanRunsRanges("./tendulkarOD.csv","Tendulkar")

batsman4s("./tendulkarOD.csv","Tendulkar")

batsman6s("./tendulkarOD.csv","Tendulkar")

batsmanScoringRateODTT("./tendulkarOD.csv","Tendulkar")

#dev.off()



**23b. Basic performance of Kohli in ODI**

par(mfrow=c(3,2))

par(mar=c(4,4,2,2))

batsmanRunsFreqPerf("./kohliOD.csv","Kohli")

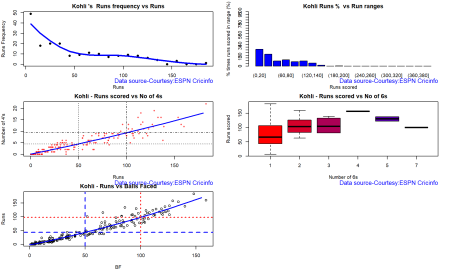
batsmanRunsRanges("./kohliOD.csv","Kohli")

batsman4s("./kohliOD.csv","Kohli")

batsman6s("./kohliOD.csv","Kohli")

batsmanScoringRateODTT("./kohliOD.csv","Kohli")

#dev.off()



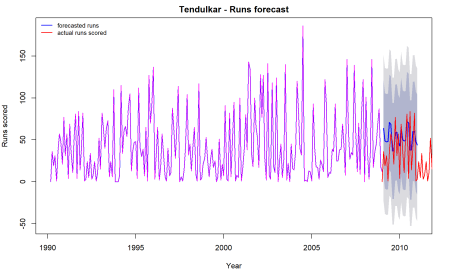
**24. Performance forecast in ODIs**

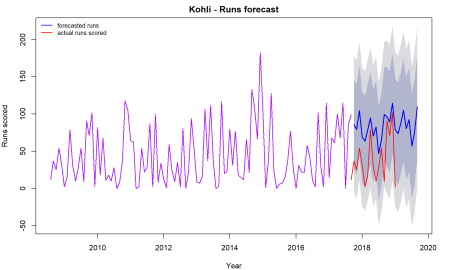
Kohli’s forecasted runs are much higher than Tendulkar’s in ODIs

par(mar=c(4,4,2,2))

batsmanPerfForecast("./tendulkarOD.csv","Tendulkar")

batsmanPerfForecast("./kohliOD.csv","Kohli")





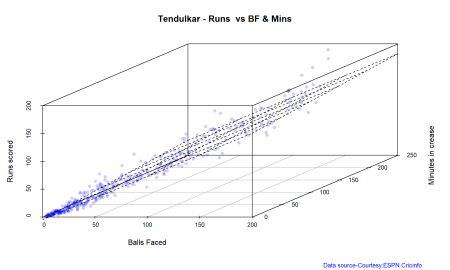
**25. Batting performance**

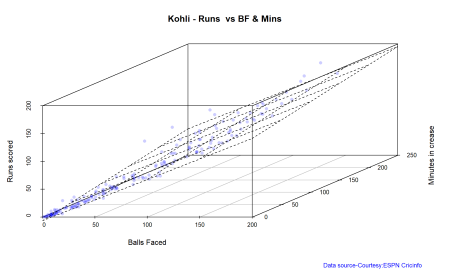
A 3D regression plane is fitted between Balls faced, Minutes at crease and Runs scored.

par(mar=c(4,4,2,2))

battingPerf3d("./tendulkarOD.csv","Tendulkar")

battingPerf3d("./kohliOD.csv","Kohli")





**26. Predicting runs scored for the ODI batsmen**

Kohli will score runs than Tendulkar for the same minutes at crease and balls faced.

BF <- seq( 10, 200,length=10)

Mins <- seq(30,220,length=10)

newDF <- data.frame(BF,Mins)

tendulkarDF <- batsmanRunsPredict("./tendulkarOD.csv","Tendulkar",newdataframe=newDF)

kohliDF <- batsmanRunsPredict("./kohliOD.csv","Kohli",newdataframe=newDF)

batsmen <-cbind(round(tendulkarDF$Runs),round(kohliDF$Runs))

colnames(batsmen) <- c("Tendulkar","Kohli")

newDF <- data.frame(round(newDF$BF),round(newDF$Mins))

colnames(newDF) <- c("BallsFaced","MinsAtCrease")

predictedRuns <- cbind(newDF,batsmen)

predictedRuns

## BallsFaced MinsAtCrease Tendulkar Kohli

## 1 10 30 7 8

## 2 31 51 26 28

## 3 52 72 45 48

## 4 73 93 64 68

## 5 94 114 83 88

## 6 116 136 102 108

## 7 137 157 121 128

## 8 158 178 140 149

## 9 179 199 159 169

## 10 200 220 178 189

**27. Runs likelihood for the ODI batsmen**

Tendulkar has clusters around 13, 53 and 111 runs while Kohli has clusters around 13, 63,116. So it more likely that Kohli will tend to score higher

par(mar=c(4,4,2,2))

batsmanRunsLikelihood("./tendulkarOD.csv","Tendulkar")

## Summary of Tendulkar 's runs scoring likelihood

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##

## There is a 18.09 % likelihood that Tendulkar will make 111 Runs in 118 balls over 172 Minutes

## There is a 28.39 % likelihood that Tendulkar will make 53 Runs in 63 balls over 95 Minutes

## There is a 53.52 % likelihood that Tendulkar will make 13 Runs in 18 balls over 27 Minutes

batsmanRunsLikelihood("./kohliOD.csv","Kohli")

## Summary of Kohli 's runs scoring likelihood

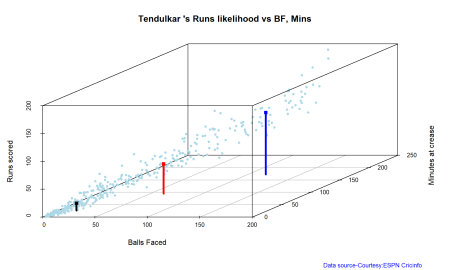
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

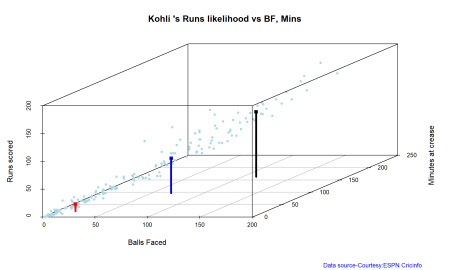
##

## There is a 31.41 % likelihood that Kohli will make 63 Runs in 69 balls over 97 Minutes

## There is a 49.74 % likelihood that Kohli will make 13 Runs in 18 balls over 24 Minutes

## There is a 18.85 % likelihood that Kohli will make 116 Runs in 113 balls over 163 Minutes



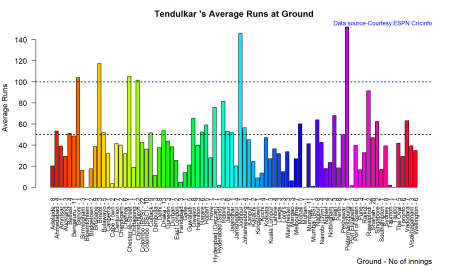


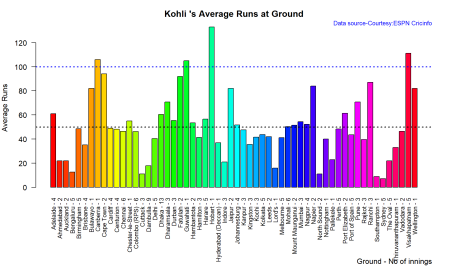
**28. Runs in different venues for the ODI batsmen**

par(mar=c(4,4,2,2))

batsmanAvgRunsGround("./tendulkarOD.csv","Tendulkar")

batsmanAvgRunsGround("./kohliOD.csv","Kohli")





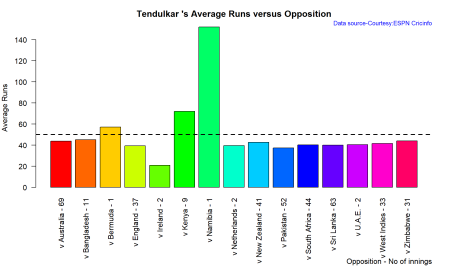
**28. Runs against different opposition for the ODI batsmen**

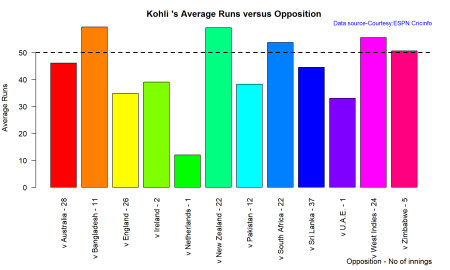
Tendulkar’s has 50+ average against Bermuda, Kenya and Namibia. While Kohli has a 50+ average against New Zealand, West Indies, South Africa, Zimbabwe and Bangladesh

par(mar=c(4,4,2,2))

batsmanAvgRunsOpposition("./tendulkarOD.csv","Tendulkar")

batsmanAvgRunsOpposition("./kohliOD.csv","Kohli")





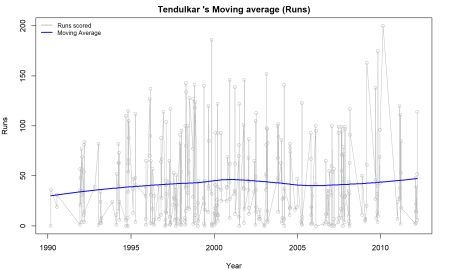
**29. Moving average of runs for the ODI batsmen**

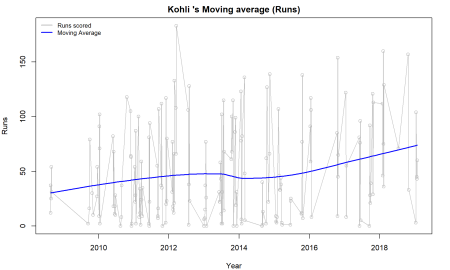
Tendulkar’s moving average shows an improvement (50+) towards the end of his career, but Kohli shows a marked increase 60+ currently

par(mar=c(4,4,2,2))

batsmanMovingAverage("./tendulkarOD.csv","Tendulkar")

batsmanMovingAverage("./kohliOD.csv","Kohli")





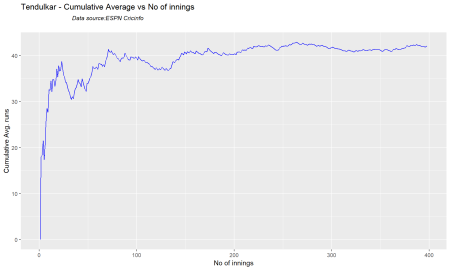
**30. Cumulative average runs of ODI batsmen**

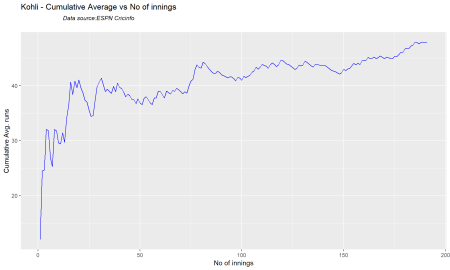
Tendulkar plateaus at 40+ while Kohli’s cumulative average runs goes up and up!!!

par(mar=c(4,4,2,2))

batsmanCumulativeAverageRuns("./tendulkarOD.csv","Tendulkar")

batsmanCumulativeAverageRuns("./kohliOD.csv","Kohli")



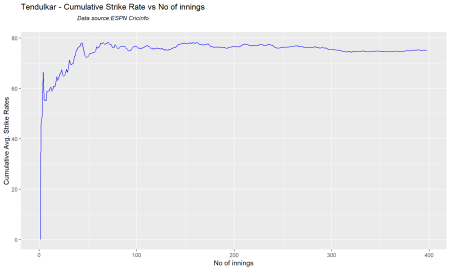


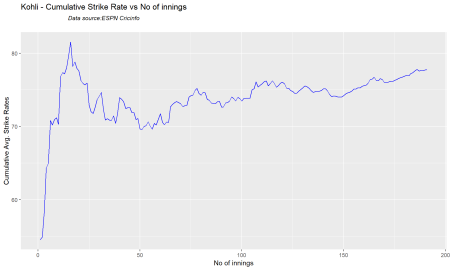
**31 Cumulative strike rate of ODI batsmen**

par(mar=c(4,4,2,2))

batsmanCumulativeStrikeRate("./tendulkarOD.csv","Tendulkar")

batsmanCumulativeStrikeRate("./kohliOD.csv","Kohli")





**32. Relative batsmen strike rate**

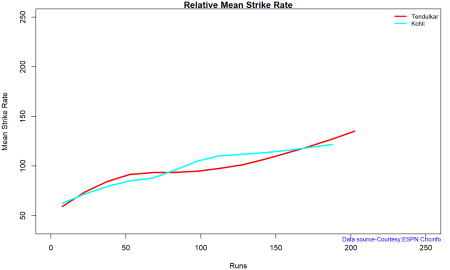
par(mar=c(4,4,2,2))

frames <- list("./tendulkarOD.csv","./kohliOD.csv")

names <- list("Tendulkar","Kohli")

relativeBatsmanSRODTT(frames,names)

#dev.off()



**33. Relative Run Frequency percentages**

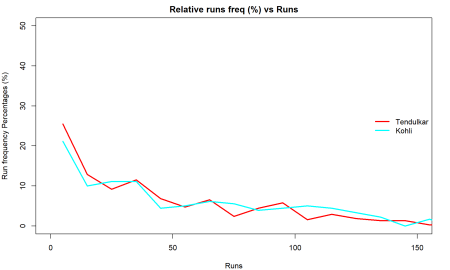
par(mar=c(4,4,2,2))

frames <- list("./tendulkarOD.csv","./kohliOD.csv")

names <- list("Tendulkar","Kohli")

relativeRunsFreqPerfODTT(frames,names)

#dev.off()



**34. Relative cumulative average runs of ODI batsmen**

Kohli breaks away from Tendulkar in cumulative average runs after 100 innings

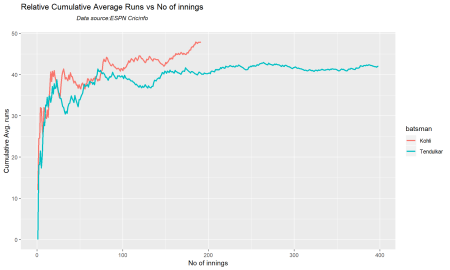
par(mar=c(4,4,2,2))

frames <- list("./tendulkarOD.csv","./kohliOD.csv")

names <- list("Tendulkar","Kohli")

relativeBatsmanCumulativeAvgRuns(frames,names)

#dev.off()



**35. Relative cumulative strike rate of ODI batsmen**

This seems to be tussle with Kohli having an edge till about 40 innings and then from 40+ to 180 innings Tendulkar leads. Kohli just seems to be edging forward.

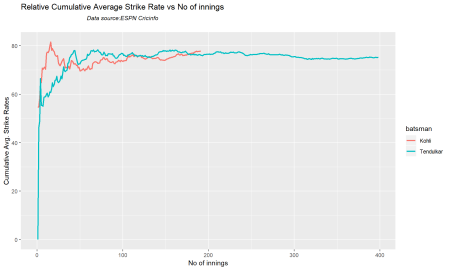
par(mar=c(4,4,2,2))

frames <- list("./tendulkarOD.csv","./kohliOD.csv")

names <- list("Tendulkar","Kohli")

relativeBatsmanCumulativeStrikeRate(frames,names)

#dev.off()



**36. Batsmen 4s and 6s**

par(mar=c(4,4,2,2))

frames <- list("./tendulkarOD.csv","./kohliOD.csv")

names <- list("Tendulkar","Kohli")

batsman4s6s(frames,names)

## Tendulkar Kohli

## Runs(1s,2s,3s) 66.29 69.67

## 4s 29.65 25.90

## 6s 4.06 4.43

#dev.off()

**37. Check ODI batsmen form**

par(mar=c(4,4,2,2))

checkBatsmanInForm("./tendulkar.csv","Tendulkar")

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Form status of Tendulkar \*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n Population size: 294 Mean of population: 50.48 \n

Sample size: 33 Mean of sample: 32.42 SD of sample: 29.8 \n\n

Null hypothesis H0 : Tendulkar 's sample average is within 95% confidence

interval of population average\n Alternative hypothesis

Ha : Tendulkar 's sample average is below the 95% confidence interval

of population average\n\n Tendulkar 's Form Status: Out-of-Form because the p value: 0.000713 is less than alpha= 0.05 \n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

checkBatsmanInForm("./kohli.csv","Kohli")

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Form status of Kohli \*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n Population size: 117 Mean of population: 50.35 \n

Sample size: 13 Mean of sample: 53.77 SD of sample: 46.15 \n\n

Null hypothesis H0 : Kohli 's sample average is within 95% confidence

interval of population average\n Alternative hypothesis

Ha : Kohli 's sample average is below the 95% confidence interval

of population average\n\n Kohli 's Form Status: In-Form because

the p value: 0.603244 is greater than alpha= 0.05 \n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

#dev.off()

**Key Findings**

1. Kohli has a better performance against oppositions like West Indies, South Africa and New Zealand
2. Kohli breaks away from Tendulkar in cumulative average runs
3. Tendulkar has been leading the strike rate rate but Kohli in recent times seems to be breaking loose.