**Introduction**

Within the last two weeks, I was introduced to Benford’s Law by 2 of my friends. Initially, I looked it up and Google and was quite intrigued by the law. Subsequently another friends asked me to check the ‘Digits’ episode, from the “Connected” series on Netflix by Latif Nasser, which I strongly recommend you watch.

Benford’s Law also called the Newcomb–Benford law, the law of anomalous numbers, or the First Digit Law states that, when dealing with quantities obtained from Nature, the frequency of appearance of each digit in the first significant place is logarithmic. For example, in sets that obey the law, the number 1 appears as the leading significant digit about 30.1% of the time, the number 2 about 17.6%, number 3 about 12.5% all the way to the number 9 at 4.6%. This interesting logarithmic pattern is observed in most natural datasets from population densities, river lengths, heights of skyscrapers, tax returns etc. What is really curious about this law, is that when we measure the lengths of rivers, the law holds perfectly regardless of the units used to measure. So the length of the rivers would obey the law whether we measure in meters, feet, miles etc. There is something almost mystical about this law.

The law has also been used widely to detect financial fraud, manipulations in tax statements, bots in twitter, fake accounts in social networks, image manipulation etc. In this age of deep fakes, the ability to detect fake images will assume paramount importance. While deviations from Benford Law do not always signify fraud, to large extent they point to an aberration. Prof Nigrini, of Cape Town used this law to identify financial discrepancies in Enron’s financial statement resulting in the infamous scandal. Also the 2009 Iranian election was found to be fradulent as the first digit percentages did not conform to those specified by Benford’s Law.

While it cannot be said with absolute certainty, marked deviations from Benford’s law could possibly indicate that there has been manipulation of natural processes. Possibly Benford’s law could be used to detect large scale match-fixing in cricket tournaments. However, we cannot look at this in isolation and the other statistical and forensic methods may be required to determine if there is fraud.

A set of numbers is said to satisfy Benford’s law if the leading digit d (d ∈ {1, …, 9}) occurs with probability

P(d)=log_{10}(1+1/d)

This law also works for number in other bases, in base b >=2

P(d)=log_{b}(1+1/d)

Interestingly, this law also applies to sports on the number of point scored in basketball etc. I was curious to see if this applied to cricket. I wanted to check if Benford’s Law worked on the runs scored, or deliveries faced by batsmen at team level or at a tournament level (IPL, Intl. T20 or ODI).

Thankfully, R has a package [benford.analysis](https://cran.r-project.org/web/packages/benford.analysis/index.html" \t "_blank) to check for data behaviour in accordance to Benford’s Law, and I have used this package in my post

This post is also available in RPubs as Benford’s Law meets IPL, Intl. T20 and ODI

library(data.table)

library(reshape2)

library(dplyr)

library(benford.analysis)

library(yorkr)

In this post, I have randomly check data with Benford’s law. The fully converted dataset is available in yorkrData2020 which I have included above. You can try on any dataset including ODI (men,women),Intl T20(men,women),IPL,BBL,PSL,NTB and WBB.

**1. Check the runs distribution by Royal Challengers Bangalore**

We can see the behaviour is as expected with Benford’s law, with minor deviations

load("/Users/tvganesh/backup/software/cricket-package/yorkr-cricsheet/yorkrData2020/ipl/iplBattingBowlingDetails/Royal Challengers Bangalore-BattingDetails.RData")

rcbRunsTrends = benford(battingDetails$runs, number.of.digits = 1, discrete = T, sign = "positive")

rcbRunsTrends

##

## Benford object:

##

## Data: battingDetails$runs

## Number of observations used = 1205

## Number of obs. for second order = 99

## First digits analysed = 1

##

## Mantissa:

##

## Statistic Value

## Mean 0.458

## Var 0.091

## Ex.Kurtosis -1.213

## Skewness -0.025

##

##

## The 5 largest deviations:

##

## digits absolute.diff

## 1 1 14.26

## 2 7 13.88

## 3 9 8.14

## 4 6 5.33

## 5 4 4.78

##

## Stats:

##

## Pearson's Chi-squared test

##

## data: battingDetails$runs

## X-squared = 5.2091, df = 8, p-value = 0.735

##

##

## Mantissa Arc Test

##

## data: battingDetails$runs

## L2 = 0.0022852, df = 2, p-value = 0.06369

##

## Mean Absolute Deviation (MAD): 0.004941381

## MAD Conformity - Nigrini (2012): Close conformity

## Distortion Factor: -18.8725

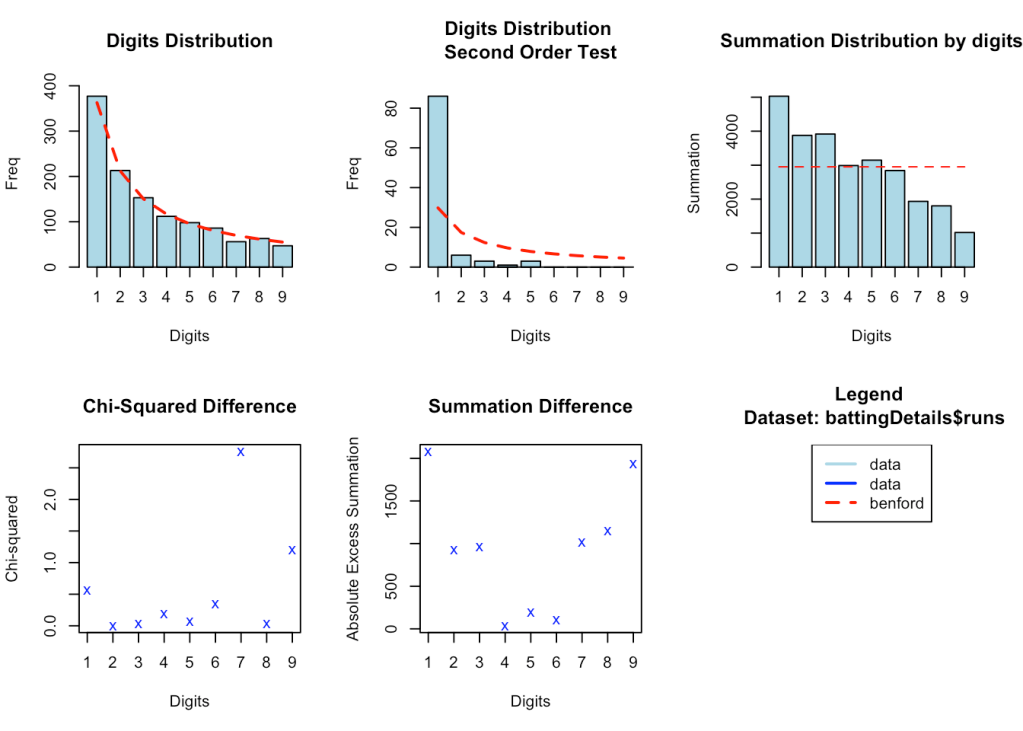
##

## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!

**1a. Plot trends**

**Note**: The Digits Distribution plot, is the plot of interest. The second order Digits Distribution is a relatively new test and is based on sorting the data and plotting the differences. The test can be applied to any data set and nonconformity usually signals an unusual issue related to data integrity. Anyway, Benford’s Law applies only to the first Digits Distribution plot. For a deeper analysis, the other plots besides other statistical tests may be required. There are other approaches to determine anamolies. I would assume, an easy way is to use Benford’s law and progressively dig deeper.

plot(rcbRunsTrends)

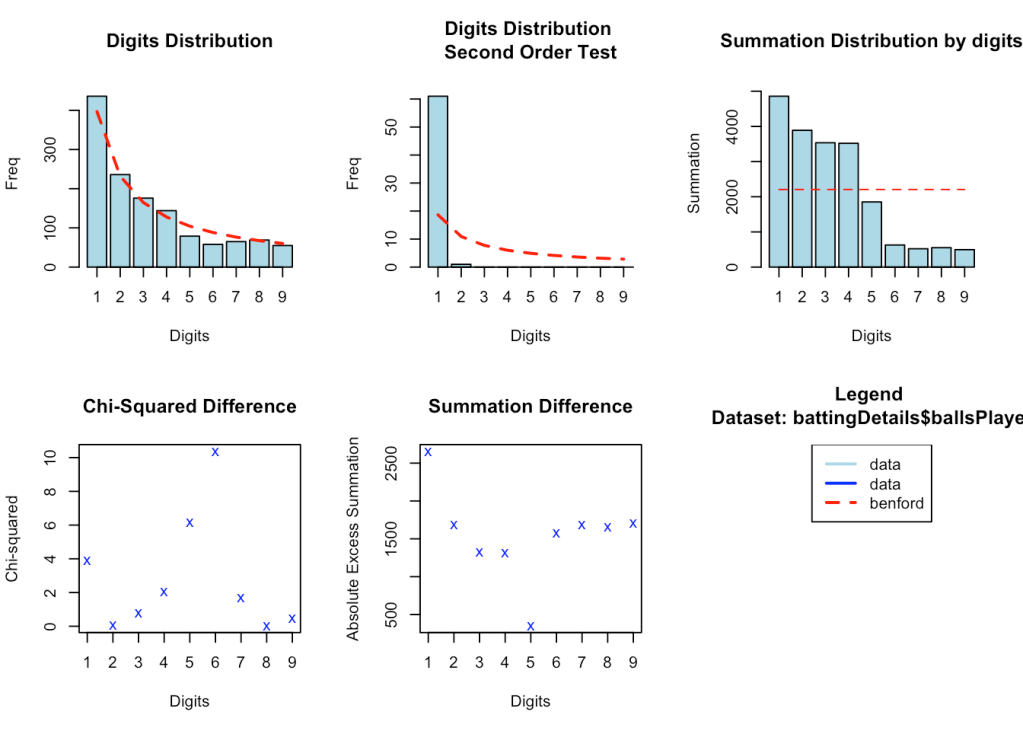


**2. Check the ‘balls played’ distribution by Royal Challengers Bangalore**

load("/Users/tvganesh/backup/software/cricket-package/yorkr-cricsheet/yorkrData2020/ipl/iplBattingBowlingDetails/Royal Challengers Bangalore-BattingDetails.RData")

rcbBallsPlayedTrends = benford(battingDetails$ballsPlayed, number.of.digits = 1, discrete = T, sign = "positive")

plot(rcbBallsPlayedTrends)



**3. Check the runs distribution by Chennai Super Kings**

The trend seems to deviate from the expected behavior to some extent in the number of digits for 5 & 7.

load("/Users/tvganesh/backup/software/cricket-package/yorkr-cricsheet/yorkrData2020/ipl/iplBattingBowlingDetails/Chennai Super Kings-BattingDetails.RData")

cskRunsTrends = benford(battingDetails$runs, number.of.digits = 1, discrete = T, sign = "positive")

cskRunsTrends

##

## Benford object:

##

## Data: battingDetails$runs

## Number of observations used = 1054

## Number of obs. for second order = 94

## First digits analysed = 1

##

## Mantissa:

##

## Statistic Value

## Mean 0.466

## Var 0.081

## Ex.Kurtosis -1.100

## Skewness -0.054

##

##

## The 5 largest deviations:

##

## digits absolute.diff

## 1 5 27.54

## 2 2 18.40

## 3 1 17.29

## 4 9 14.23

## 5 7 14.12

##

## Stats:

##

## Pearson's Chi-squared test

##

## data: battingDetails$runs

## X-squared = 22.862, df = 8, p-value = 0.003545

##

##

## Mantissa Arc Test

##

## data: battingDetails$runs

## L2 = 0.002376, df = 2, p-value = 0.08173

##

## Mean Absolute Deviation (MAD): 0.01309597

## MAD Conformity - Nigrini (2012): Marginally acceptable conformity

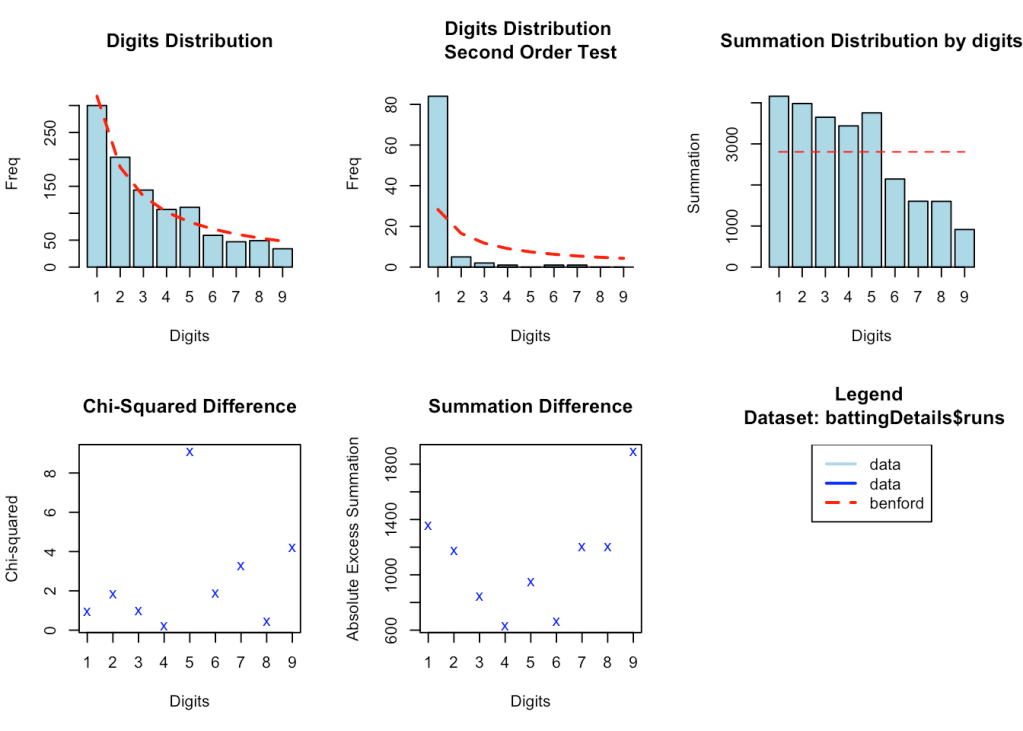
## Distortion Factor: -17.90664

##

## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!

**3a. Plot the trends**

plot(cskRunsTrends)



##3b. Check details of suspicious behavious Interestingly the package benford.analysis has functions to get details of data which result in the deviation. The getSuspects() function returns data which are ‘suspicious’. We probably need to look at aberrations with a pinch of salt. A look at the statistical distribution and other investigation would need to be carried out to determine the cause.

suspects <- getSuspects(cskRunsTrends, battingDetails)

suspects

## batsman ballsPlayed fours sixes runs strikeRate bowler

## 1: JA Morkel 18 1 2 29 161.11 nobody

## 2: MS Dhoni 31 2 0 23 74.19 Shahid Afridi

## 3: MS Dhoni 22 1 1 22 100.00 Shoaib Ahmed

## 4: SK Raina 19 1 2 25 131.58 nobody

## 5: MS Dhoni 37 6 1 58 156.76 nobody

## ---

## 311: MS Dhoni 12 3 1 25 208.33 nobody

## 312: SK Raina 43 5 2 54 125.58 nobody

## 313: Harbhajan Singh 8 0 0 2 25.00 nobody

## 314: SK Raina 13 4 0 22 169.23 nobody

## 315: AT Rayudu 21 2 0 25 119.05 nobody

## wicketFielder wicketKind wicketPlayerOut date

## 1: nobody notOut notOut 2008-05-06

## 2: Shahid Afridi caught MS Dhoni 2008-05-06

## 3: Shoaib Ahmed caught MS Dhoni 2009-04-27

## 4: nobody caught and bowled SK Raina 2009-04-27

## 5: nobody notOut notOut 2009-05-04

## ---

## 311: nobody notOut notOut 2018-04-22

## 312: nobody notOut notOut 2018-04-22

## 313: nobody notOut notOut 2018-05-22

## 314: nobody bowled SK Raina 2018-05-22

## 315: nobody notOut notOut 2019-04-17

## venue opposition

## 1: MA Chidambaram Stadium, Chepauk Deccan Chargers

## 2: MA Chidambaram Stadium, Chepauk Deccan Chargers

## 3: Kingsmead Deccan Chargers

## 4: Kingsmead Deccan Chargers

## 5: Buffalo Park Deccan Chargers

## ---

## 311: Rajiv Gandhi International Stadium, Uppal Sunrisers Hyderabad

## 312: Rajiv Gandhi International Stadium, Uppal Sunrisers Hyderabad

## 313: Wankhede Stadium Sunrisers Hyderabad

## 314: Wankhede Stadium Sunrisers Hyderabad

## 315: Rajiv Gandhi International Stadium, Uppal Sunrisers Hyderabad

## winner result

## 1: Deccan Chargers NA

## 2: Deccan Chargers NA

## 3: Deccan Chargers NA

## 4: Deccan Chargers NA

## 5: Chennai Super Kings NA

## ---

## 311: Chennai Super Kings NA

## 312: Chennai Super Kings NA

## 313: Chennai Super Kings NA

## 314: Chennai Super Kings NA

## 315: Sunrisers Hyderabad NA

**4. Check runs distribution in all of Indian Premier League (IPL)**

battingDF <- NULL

teams <-c("Chennai Super Kings","Deccan Chargers","Delhi Daredevils",

"Kings XI Punjab", 'Kochi Tuskers Kerala',"Kolkata Knight Riders",

"Mumbai Indians", "Pune Warriors","Rajasthan Royals",

"Royal Challengers Bangalore","Sunrisers Hyderabad","Gujarat Lions",

"Rising Pune Supergiants")

setwd("/Users/tvganesh/backup/software/cricket-package/yorkr-cricsheet/yorkrData2020/ipl/iplBattingBowlingDetails")

for(team in teams){

battingDetails <- NULL

val <- paste(team,"-BattingDetails.RData",sep="")

print(val)

tryCatch(load(val),

error = function(e) {

print("No data1")

setNext=TRUE

}

)

details <- battingDetails

battingDF <- rbind(battingDF,details)

}

## [1] "Chennai Super Kings-BattingDetails.RData"

## [1] "Deccan Chargers-BattingDetails.RData"

## [1] "Delhi Daredevils-BattingDetails.RData"

## [1] "Kings XI Punjab-BattingDetails.RData"

## [1] "Kochi Tuskers Kerala-BattingDetails.RData"

## [1] "Kolkata Knight Riders-BattingDetails.RData"

## [1] "Mumbai Indians-BattingDetails.RData"

## [1] "Pune Warriors-BattingDetails.RData"

## [1] "Rajasthan Royals-BattingDetails.RData"

## [1] "Royal Challengers Bangalore-BattingDetails.RData"

## [1] "Sunrisers Hyderabad-BattingDetails.RData"

## [1] "Gujarat Lions-BattingDetails.RData"

## [1] "Rising Pune Supergiants-BattingDetails.RData"

trends = benford(battingDF$runs, number.of.digits = 1, discrete = T, sign = "positive")

trends

##

## Benford object:

##

## Data: battingDF$runs

## Number of observations used = 10129

## Number of obs. for second order = 123

## First digits analysed = 1

##

## Mantissa:

##

## Statistic Value

## Mean 0.4521

## Var 0.0856

## Ex.Kurtosis -1.1570

## Skewness -0.0033

##

##

## The 5 largest deviations:

##

## digits absolute.diff

## 1 2 159.37

## 2 9 121.48

## 3 7 93.40

## 4 8 83.12

## 5 1 61.87

##

## Stats:

##

## Pearson's Chi-squared test

##

## data: battingDF$runs

## X-squared = 78.166, df = 8, p-value = 1.143e-13

##

##

## Mantissa Arc Test

##

## data: battingDF$runs

## L2 = 5.8237e-05, df = 2, p-value = 0.5544

##

## Mean Absolute Deviation (MAD): 0.006627966

## MAD Conformity - Nigrini (2012): Acceptable conformity

## Distortion Factor: -20.90333

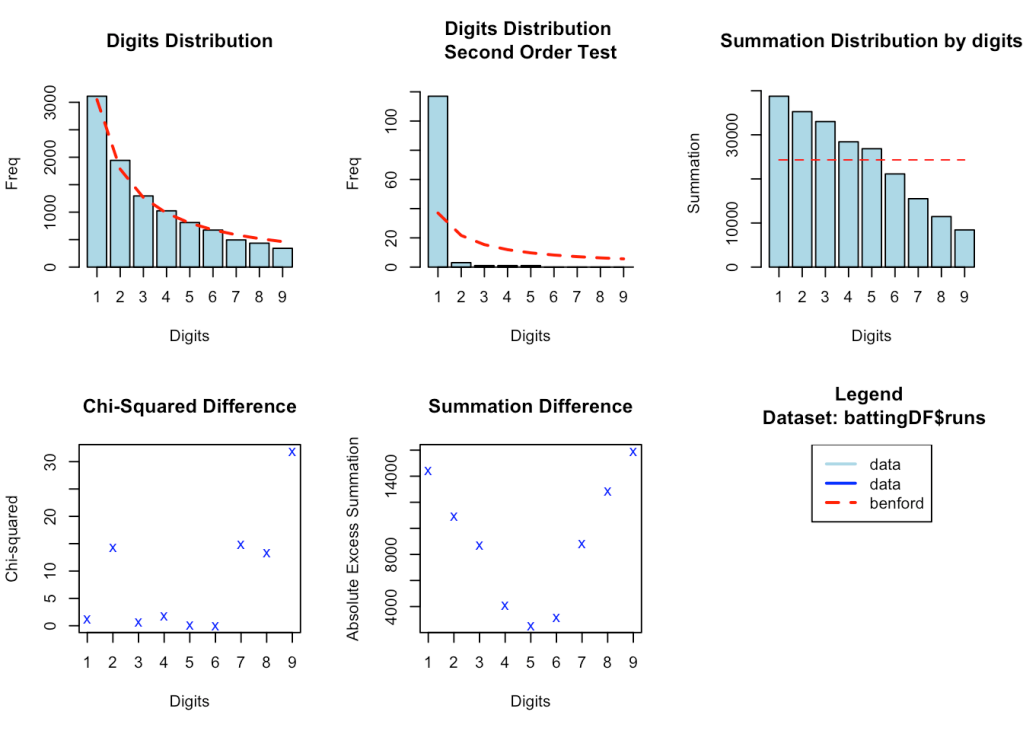
##

## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!

**4b. Plot trends all of IPL**

We can see that the trend follows quite closely to Benford’s curve for all of IPL

plot(trends)



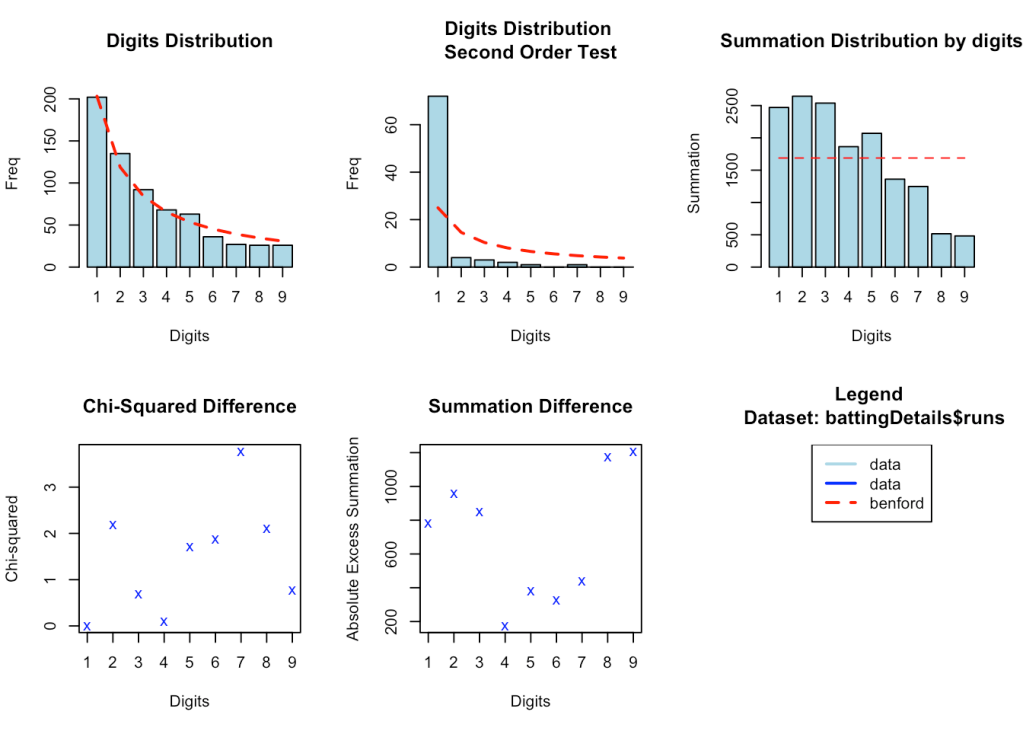
**5. Check Benford’s law in India matches**

setwd("/Users/tvganesh/backup/software/cricket-package/yorkr-cricsheet/yorkrData2020/t20/t20BattingBowlingDetails")

load("India-BattingDetails.RData")

indiaTrends = benford(battingDetails$runs, number.of.digits = 1, discrete = T, sign = "positive")

plot(indiaTrends)



**6. Check Benford’s law in all of Intl. T20**

setwd("/Users/tvganesh/backup/software/cricket-package/yorkr-cricsheet/yorkrData2020/t20/t20BattingBowlingDetails")

teams <-c("Australia","India","Pakistan","West Indies", 'Sri Lanka',

"England", "Bangladesh","Netherlands","Scotland", "Afghanistan",

"Zimbabwe","Ireland","New Zealand","South Africa","Canada",

"Bermuda","Kenya","Hong Kong","Nepal","Oman","Papua New Guinea",

"United Arab Emirates","Namibia","Cayman Islands","Singapore",

"United States of America","Bhutan","Maldives","Botswana","Nigeria",

"Denmark","Germany","Jersey","Norway","Qatar","Malaysia","Vanuatu",

"Thailand")

for(team in teams){

battingDetails <- NULL

val <- paste(team,"-BattingDetails.RData",sep="")

print(val)

tryCatch(load(val),

error = function(e) {

print("No data1")

setNext=TRUE

}

)

details <- battingDetails

battingDF <- rbind(battingDF,details)

}

intlT20Trends = benford(battingDF$runs, number.of.digits = 1, discrete = T, sign = "positive")

intlT20Trends

##

## Benford object:

##

## Data: battingDF$runs

## Number of observations used = 21833

## Number of obs. for second order = 131

## First digits analysed = 1

##

## Mantissa:

##

## Statistic Value

## Mean 0.447

## Var 0.085

## Ex.Kurtosis -1.158

## Skewness 0.018

##

##

## The 5 largest deviations:

##

## digits absolute.diff

## 1 2 361.40

## 2 9 276.02

## 3 1 264.61

## 4 7 210.14

## 5 8 198.81

##

## Stats:

##

## Pearson's Chi-squared test

##

## data: battingDF$runs

## X-squared = 202.29, df = 8, p-value < 2.2e-16

##

##

## Mantissa Arc Test

##

## data: battingDF$runs

## L2 = 5.3983e-06, df = 2, p-value = 0.8888

##

## Mean Absolute Deviation (MAD): 0.007821098

## MAD Conformity - Nigrini (2012): Acceptable conformity

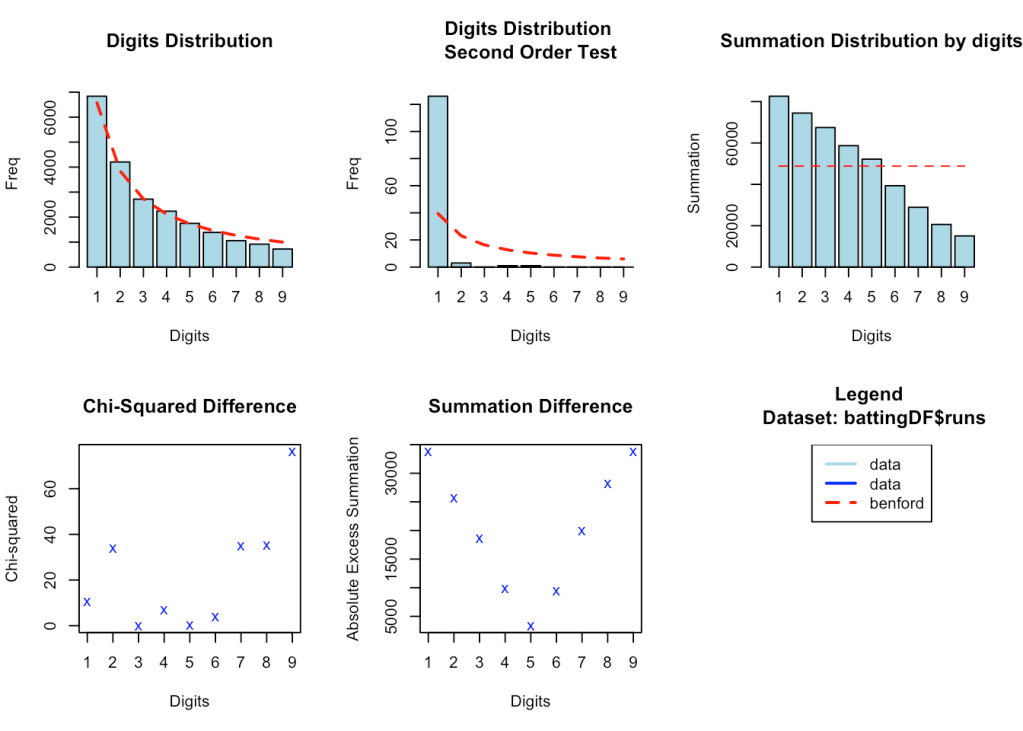
## Distortion Factor: -24.11086

##

## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!

**5a. Plot trends**

plot(intlT20Trends)



**6. Check Benford’s law in ODI**

This plot also nicely follows the Benford’s predicted curve

setwd("/Users/tvganesh/backup/software/cricket-package/yorkr-cricsheet/yorkrData2020/odi/odiBattingBowlingDetails")

teams <-c("Australia","India","Pakistan","West Indies", 'Sri Lanka',

"England", "Bangladesh","Netherlands","Scotland", "Afghanistan",

"Zimbabwe","Ireland","New Zealand","South Africa","Canada",

"Bermuda","Kenya","Hong Kong","Nepal","Oman","Papua New Guinea",

"United Arab Emirates","Namibia","Cayman Islands","Singapore",

"United States of America","Bhutan","Maldives","Botswana","Nigeria",

"Denmark","Germany","Jersey","Norway","Qatar","Malaysia","Vanuatu",

"Thailand")

battingDF<-NULL

for(team in teams){

battingDetails <- NULL

val <- paste(team,"-BattingDetails.RData",sep="")

print(val)

tryCatch(load(val),

error = function(e) {

print("No data1")

setNext=TRUE

}

)

details <- battingDetails

battingDF <- rbind(battingDF,details)

}

odiTrends = benford(battingDF$runs, number.of.digits = 1, discrete = T, sign = "positive")

odiTrends

##

## Benford object:

##

## Data: battingDF$runs

## Number of observations used = 23766

## Number of obs. for second order = 179

## First digits analysed = 1

##

## Mantissa:

##

## Statistic Value

## Mean 0.468

## Var 0.089

## Ex.Kurtosis -1.204

## Skewness -0.069

##

##

## The 5 largest deviations:

##

## digits absolute.diff

## 1 5 240.18

## 2 4 190.84

## 3 9 177.47

## 4 8 157.69

## 5 1 66.28

##

## Stats:

##

## Pearson's Chi-squared test

##

## data: battingDF$runs

## X-squared = 100.07, df = 8, p-value < 2.2e-16

##

##

## Mantissa Arc Test

##

## data: battingDF$runs

## L2 = 0.002845, df = 2, p-value < 2.2e-16

##

## Mean Absolute Deviation (MAD): 0.004609365

## MAD Conformity - Nigrini (2012): Close conformity

## Distortion Factor: -14.92332

##

## Remember: Real data will never conform perfectly to Benford's Law. You should not focus on p-values!

**Plot trends**

plot(odiTrends)

