**The intuition**

Let us begin with a brief explanation about Benford’s law and why should it work as a fraud detector method. Given a set of numbers, the first thing we need to do is to extract the first digit of each number. For example, for (121,245,12,55) the first digits will be (1,2,1,5). Perhaps our intuition would say that for a large set of numbers, each first digit, from 1 to 9, would appear in equal proportion, that is P(x = digit) = 1/9for each digit between 1 and 9. However, Benford’s law shows us that this is not true. In fact, smaller digits will have larger probabilits. If you want to see a very didactic explanation of why this happen just watch this video <https://www.youtube.com/watch?v=XXjlR2OK1kM&t=460s> . We could not give a better explanation.

The probability of each first digit can be obtained through the following formula:

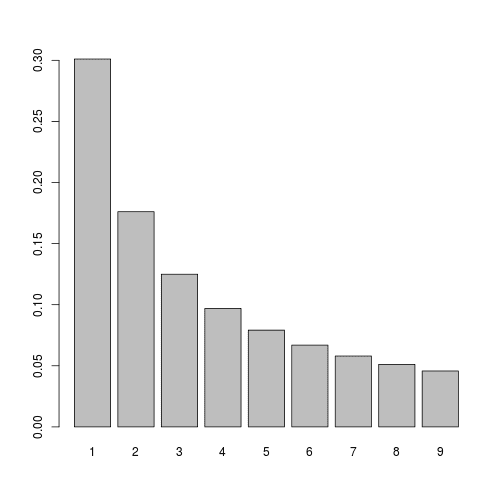
\displaystyle \log_{10}\left(1+\frac{1}{d}\right)

which will result in the following probabilities:

benford1 = log10(1+1/1:9)

names(benford1) = 1:9

barplot(benford1)



benford1

## 1 2 3 4 5 6

## 0.30103000 0.17609126 0.12493874 0.09691001 0.07918125 0.06694679

## 7 8 9

## 0.05799195 0.05115252 0.04575749

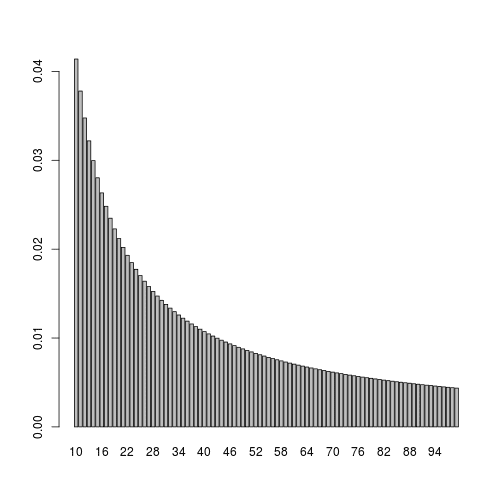
**Generalization to more than one digit**

The Benford’s law can be generalized to more digits. We will show the results for the first two digits and for the second digit alone. The formula for the first two digits is the same as the formula for the first digit with dbetween 10 and 99.

benford12 = log10(1+1/10:99)

names(benford12) = 10:99

barplot(benford12)



Finally, probabilities for the second digit alone can be obtained from the plot above by adding the probabilities that have the same second digits, from 0 to 9.

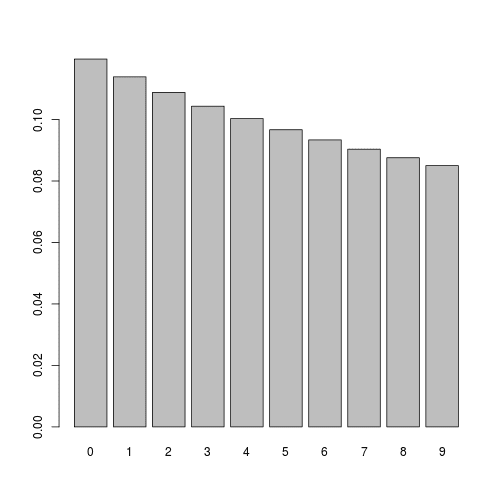
library(tidyverse)

aux = 10:99 - floor(10:99/10)\*10

benford2 = (data.frame(v1 = benford12,v2 = aux) %>% group\_by(v2) %>% summarise(v1 = sum(v1)))$v1

names(benford2) = 0:9

barplot(benford2)



**When can we use the Benford’s law?**

Naturally, the Benford’s law requires data in a large range that goes through several orders of magnitude. For example, the law will not work for heights and weights of human beings. However, it is more likely to work for average heights of all species in the world or for the population of all countries in the world.

**Application: Brazilian Presidential Elections from 2002 to 2018**

The Brazilian electoral system has raised a lot of inflamed discussion in the previous years, especially when we were closer to the 2018 elections. Most of the critiques come from the fact that the Brazilian elections use an electronic ballot box that does not print the vote. Some people say these electronic devices are safe and some people say they make the elections blurry and impossible to audit. The latest discussion were mostly focused on the following facts:

* Brazilian congress passes a bill that forces the electronic devices to print the votes. The voter would then verify the information in the paper and put it in a regular ballot box. If the electoral results came to be suspicious we could count the paper votes to see if the results match.
* President Dilma used her veto power on this bill, forcing the elections to stay as they were, without the printed votes.
* The Congress had a new voting that bypassed the president’s veto and reestablished the printed votes.
* The Supreme Court declared the new bill to be unconstitutional and that was the end of the discussion. We had the elections only with the electronic votes.

Of course this raised a lot of critiques because there was a lot a pressure coming from the government to stop printed votes. Moreover, the same party was on the government for the fourth four years term and most of the judges is the Supreme Court were put there by them. We also had some public trials where hackers would try to break the electronic ballot and so they did without any problems.

On the other hand, no one knew exactly how a potential fraud would be performed without raising suspicion, the discussion was very speculative and did not solve anything. A alternative media company called Brasil Paralelo started a research on the Benford’s law and committed to use the law in the 2018 elections. Previous tests showed that the 2010 elections were fine and raised some doubts about the 2014 elections. They showed their results in a video and said that they found close to 80% inconsistency in the data in the 2018 first round. They were going to publish a document with more details on the results but we were not able to find it. Therefore, we will base our opinion solely on the video. The data is published by the government and they used it in a aggregation level of electoral zones, which is in fact the most adequate aggregation. An electoral zone is like a school, where we have several rooms with ballots where people go to vote. The size of electoral zones has a lot of variation going from a few hundreds or less to more than 50000, thus the use of the Benford’s law seems adequate. However, this study did  not answer all our questions (that is our personal opinion) because of the following reasons:

* We have our doubts on how efficiently the Benford’s law can detect frauds. In the empirical example we will fraud the data to show that we need a significant change to be able to see the fraud. Moreover, it is very hard that any real data will fit perfectly to the law. This has been widely shown by a chi2 test that rejects the data following the law, even if the figures show a very good fit. Therefore, we can only be suspicious if the data deviates a lot from the Benford’s law and, in our opinion, we should also be concerned about changing patterns in the data. For example, the same tests should be performed to many elections to see what changes between them. The data from many elections is easy to download and organize (as we will show in the application) and we could not find comparisons between elections in the Brasil Paralelo application.
* It is possible to fail to detect the fraud if we apply the law to all the data at the same time. Therefore, one should try to use sub-samples such as regions and states to give more robustness to the results. Brasil Paralelo did that, but they went to far in our opinion. Many of their sub-samples are not big enough. The fact is that we need large samples to have a reasonable convergence to the Benford’s law distribution. What we understood from the video is that they tested more than 100 sub-samples, only for the first digit equals 1. They based their analysis a lot on State level subsamples and many States may have a bad result just because the sample is very small. This comments take me to the next problem:
* They did more than 100 tests of the Benford’s law and reported them in the video for the first digit equals 1. They used a confidence band of 3% and rejected all tests that had a proportion of 1s as the first digit outside the interval. The high inconsistency they reported in the end means that many of the sub-samples had proportions outside the intervals. The tricky thing here is that the intervals are the same for all sub-samples, i.e. they did not control the intervals for the sample size, which is something very obvious. If they did, and compared to all elections since 2002, they would probably find that based on their criteria all elections would be inconsistent, which only means that this test is not very good.

We would like to apologize to Brasil Paralelo if our comments were unfair. They were based on the video alone and we may have missed something. Their analysis was very good and they were very brave to do it. They only deserve our respect.

Disclosure: We are not going to try to fake neutrality. We are all for transparency and the chain of events that killed the printed vote bill is very weird. However, we will try to make our analysis as unbiased as possible. The most important thing here is that all the results can be reproduced by anyone with a little knowledge of R. We are not audit experts and we do can make mistakes.

**2002 elections**

library(tidyverse)

The first step was to download de data and to create a function that transformed the votes into digits proportion to compare with the Benford’s law. We put the download code, the function and some metadata on Brazilian regions in the end of the post.

We will start the analysis by assuming that the 2002 elections were fine, otherwise we have no base to compare the other elections. Moreover, the analysis will be on the second round of the elections.

## Prepare the data for the Benford's law ##

data2002 = prepare\_data(elections2002, round = 2, ncandidates = 2)

Now let us see how the data fits to the Benford’s law. The bars represent the Benford’s law and the lines are the proportion of digits for each candidate. The adjustment of the data to the Benford’s law is very good in the three figures below.

dt = data2002

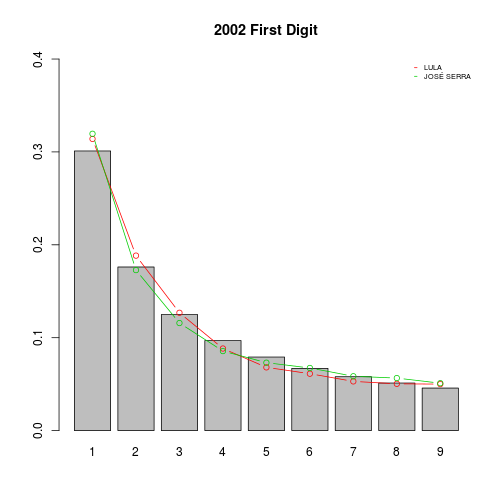
bar = barplot(benford1, ylim = c(0,0.4), main = "2002 First Digit")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



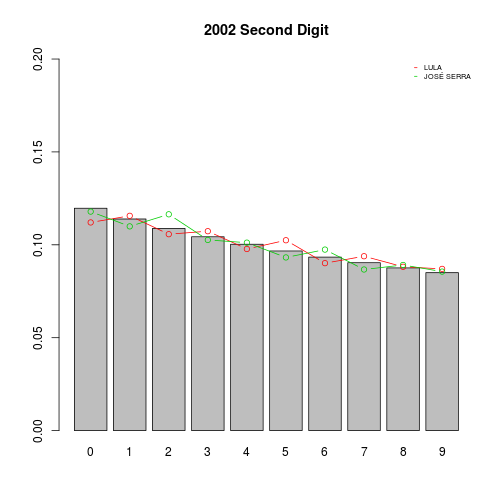
bar = barplot(benford2, ylim = c(0,0.2), main = "2002 Second Digit")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$second$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



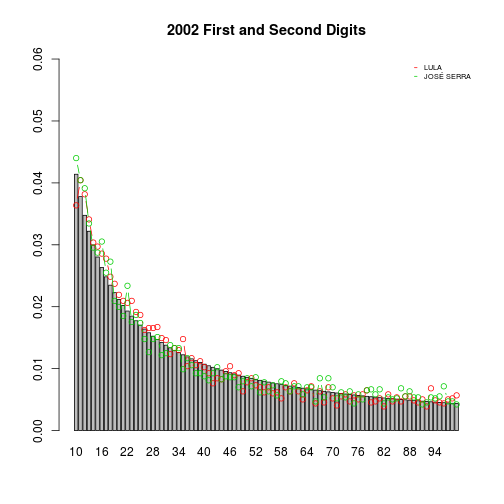
bar = barplot(benford12, ylim = c(0,0.06), main = "2002 First and Second Digits")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$fands$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



We could not find anything interesting in the second digit analysis alone. Therefore, we will only show the results for the first digit and the first and second digits together from here on. If you want to see the results for the second digit you can do it yourself by modifying the second digit code above to other data. Additionally, in the end of the post we show an analysis where we induce some fraud to the data to see how the Benford’s law behaves. The code is very simple and you can use it to test several types of fraud.

**Looking at the Regions**

Brazil has five regions of States, in this part of the post we will look at the two biggest regions (Southeastern and Northeastern regions). The reason is that the fraud may be small, local and hard to detect if we look at the entire country. These two regions have enough data to make us comfortable with the sample size. First, let’s prepare the data:

data2002SE = prepare\_data(filter(elections2002, SIGLA\_UF %in%SE),2,2)

data2002NE = prepare\_data(filter(elections2002, SIGLA\_UF %in%NE),2,2)

The plots are generated with the code below. As you can see, if we move to regions we find some bigger deviations, especially in the SE region. It is very hard to say if this is a pattern from the data or a fraud. The best we could think of is to look at other years and see if the pattern stays the same.

dtlist = list("2002SE"=data2002SE, "2002NE" = data2002NE)

lim = 0.4

for(j in 1:2){

dt = dtlist[[j]]

bar = barplot(benford1, ylim = c(0,lim), main = names(dtlist)[j])

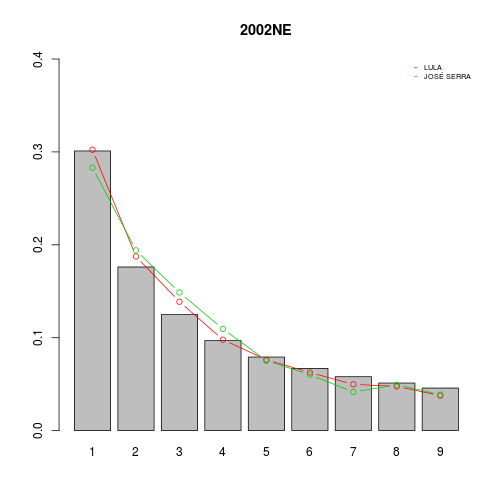
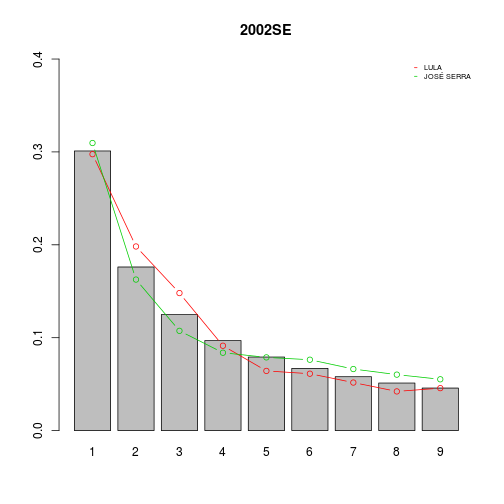
for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

}



**Comparing all elections from 2002 to 2014**

**Brazil**

Preparing the data:

data2002 = prepare\_data(elections2002,2,2)

data2006 = prepare\_data(elections2006,2,2)

data2010 = prepare\_data(elections2010,2,2)

data2014 = prepare\_data(elections2014,2,2)

The code below plots the Benford’s law and the data for all elections between 2002 and 2014 for the entire country. Although we may see small variations across the years. we don’t think we can say anything about fraud by looking only at these plots. Assuming 2002 was ok, all the figures below can say is that 2006, 2010 and 2014 were also ok.

dtlist = list("2002"=data2002,"2006"=data2006,"2010"=data2010,"2014"=data2014)

par(mfrow=c(2,2))

lim = 0.4

for(j in 1:4){

dt = dtlist[[j]]

bar = barplot(benford1, ylim = c(0,lim), main = names(dtlist)[j])

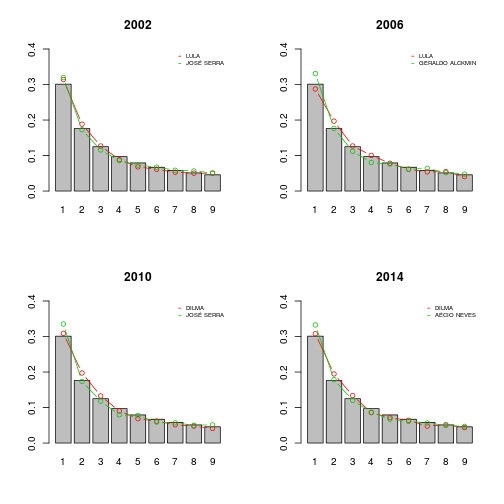
for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

}



lim = 0.06

for(j in 1:4){

dt = dtlist[[j]]

bar = barplot(benford12, ylim = c(0,lim), main = names(dtlist)[j])

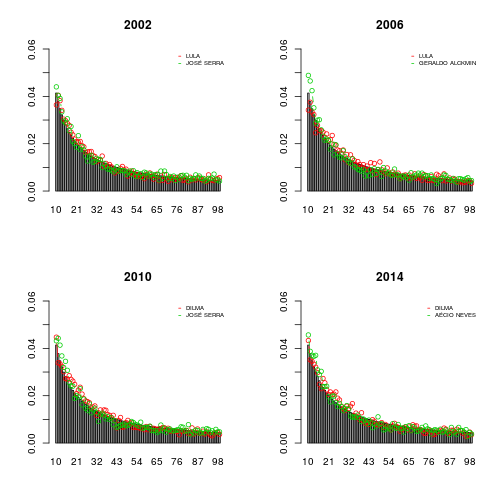
for(i in 1:length(dt)){

lines(bar,dt[[i]]$fands$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

}



**Southeastern Region**

Preparing the data:

data2002SE = prepare\_data(filter(elections2002, SIGLA\_UF %in%SE),2,2)

data2006SE = prepare\_data(filter(elections2006, SIGLA\_UF %in%SE),2,2)

data2010SE = prepare\_data(filter(elections2010, SIGLA\_UF %in%SE),2,2)

data2014SE = prepare\_data(filter(elections2014, SIGLA\_UF %in%SE),2,2)

The code below plots the Benford’s law and the data for all elections between 2002 and 2014 for the Southeastern region. The conclusions here as the same as the conclusions for the entire country. If 2002 was fine then the other years were also fine. Note that the plots show the winner in red and the looser in green. Since Aecio won in the Southeast the colors are alternated in the 2014 plot. Nevertheless, Dilma’s patters remains the same as in 2010 and the same as Lula’s pattern in 2002 and 2006. Dilma was Lula’s candidate after his two terms and they are from the same party. The opposition candidate was always from the same party too.

dtlist = list("2002SE"=data2002SE,"2006SE"=data2006SE,"2010SE"=data2010SE,"2014SE"=data2014SE)

par(mfrow=c(2,2))

lim = 0.4

for(j in 1:4){

dt = dtlist[[j]]

bar = barplot(benford1, ylim = c(0,lim), main = names(dtlist)[j])

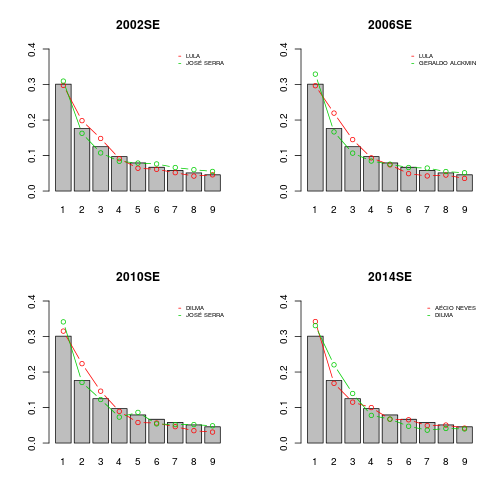
for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

}



lim = 0.06

for(j in 1:4){

dt = dtlist[[j]]

bar = barplot(benford12, ylim = c(0,lim), main = names(dtlist)[j])

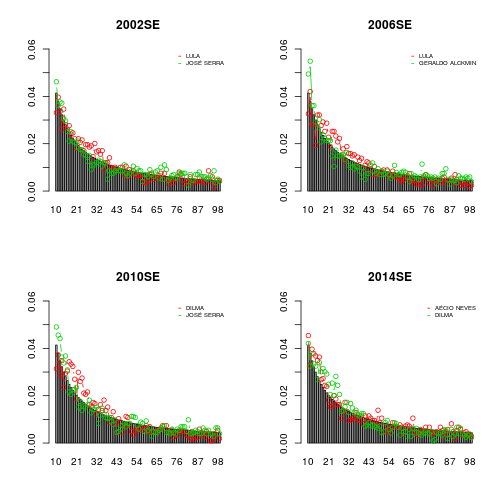
for(i in 1:length(dt)){

lines(bar,dt[[i]]$fands$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

}



**Northeastern Region**

Preparing the data:

data2002NE = prepare\_data(filter(elections2002, SIGLA\_UF %in%NE),2,2)

data2006NE = prepare\_data(filter(elections2006, SIGLA\_UF %in%NE),2,2)

data2010NE = prepare\_data(filter(elections2010, SIGLA\_UF %in%NE),2,2)

data2014NE = prepare\_data(filter(elections2014, SIGLA\_UF %in%NE),2,2)

The code below plots the Benford’s law and the data for all elections between 2002 and 2014 for the Northeastern region. Here we can start to see some difference in the plots that may raise some doubts. In 2002, everything was fine and the data for both candidates fit the Benford’s law very well. However, since 2006 we have a clear deviation from the law for the candidate that won the elections. The proportion of ones as the first digit decreases a lot and the proportion of larger values are mostly bigger than what the Benford’s law would predict. Similar results are found if we look at the first two digits together. This could be due to a change in voting patters or fraud. However, if we choose the fraud hypothesis we would have to assume that all elections were fraudulent between 2006 and 2014. This could be the case, because in 2006 Lula was reelected, meaning that he was already in the government and may have had some influence in the electoral process. Still, we don’t think the Benford’s law is useful to detect frauds alone. It can only be useful to support other evidence and perhaps give a stronger confirmation. Results like this could be used as a starting point to further investigation on the elections. However, auditing electronic ballot boxes is not easy.

dtlist = list("2002NE"=data2002NE,"2006NE"=data2006NE,"2010NE"=data2010NE,"2014NE"=data2014NE)

par(mfrow=c(2,2))

lim = 0.4

for(j in 1:4){

dt = dtlist[[j]]

bar = barplot(benford1, ylim = c(0,lim), main = names(dtlist)[j])

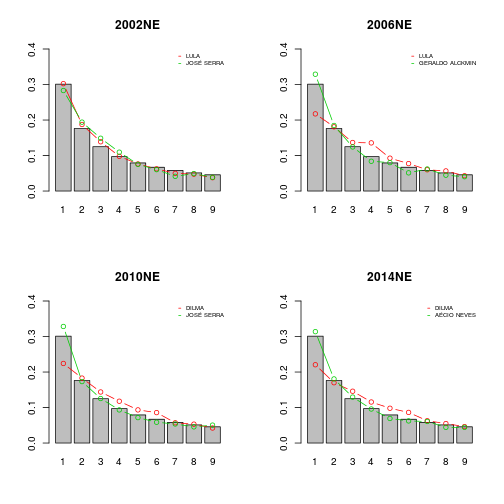
for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

}



lim = 0.06

for(j in 1:4){

dt = dtlist[[j]]

bar = barplot(benford12, ylim = c(0,lim), main = names(dtlist)[j])

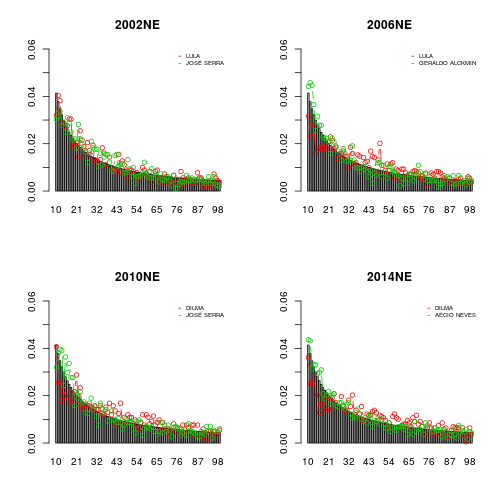
for(i in 1:length(dt)){

lines(bar,dt[[i]]$fands$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

}



**2018 elections**

The 2018 elections were very different. First, the use of social media was very important and second, the opposition candidate in the second round was from a whole new party. The worker’s party still managed to put a candidate (Fernando Haddad) in the second round. Moreover, Dilma was impeached in 2016 and the vice president Temer assumed. Recall that the 2018 elections must be downloaded directly from the government page. In the end of the post we give the details on how to download the data and put it in the same format as the 2002-2014 data for the same code to work.

First, let us look at the entire country. The results are generated by the code below. The figure is just below the code. Once again, every thing looks fine if we look at the entire country. The data fits very well to the Benford’s law.

# prepare the data #

data2018 = prepare\_data(elections2018,2,2)

dt = data2018

par(mfrow = c(2,2))

bar = barplot(benford1, ylim = c(0,0.4), main = "2018 First Digit")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

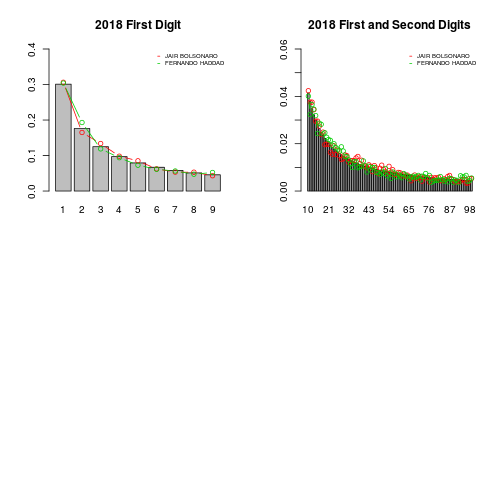
bar = barplot(benford12, ylim = c(0,0.06), main = "2018 First and Second Digits")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$fands$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



**Southeast Region**

Now let’s move to the Southeastern region. Here we have something very unusual. The curve for both candidates deviate significantly from the Benford’s law if we compare this region with previous elections. The curve for Fernando Haddad overestimates the smaller digits and underestimates the bigger digits and the curve for Jair Bolsonaro shows the opposite situation. Although the fit for the SE region was not perfect in previous elections, it followed the same pattern that was broken in 2018.

# prepare the data #

data2018SE = prepare\_data(filter(elections2018,SIGLA\_UF%in%SE),2,2)

dt = data2018SE

par(mfrow = c(2,2))

bar = barplot(benford1, ylim = c(0,0.4), main = "2018SE First Digit")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

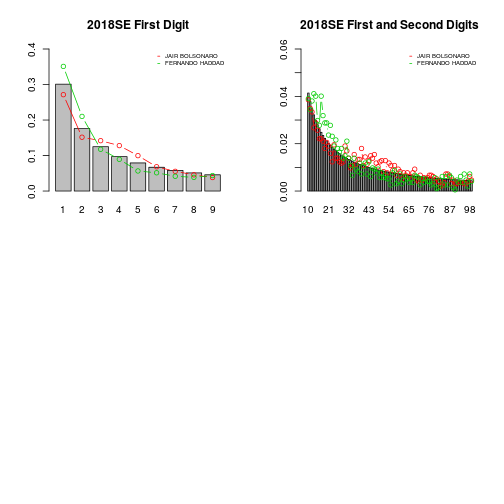
bar = barplot(benford12, ylim = c(0,0.06), main = "2018SE First and Second Digits")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$fands$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



**Northeast Region**

Next, the NE region. There is nothing new here. The worker’s party candidate has the same pattern from the 2006-2014 period and the other candidate looks fine. Therefore, to say that the 2018 elections were fraudulent in the NE region we must also say that all elections from 2006 to 2014 were also fraudulent.

# prepare the data #

data2018NE = prepare\_data(filter(elections2018,SIGLA\_UF%in%NE),2,2)

dt = data2018NE

par(mfrow = c(2,2))

bar = barplot(benford1, ylim = c(0,0.4), main = "2018NE First Digit")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

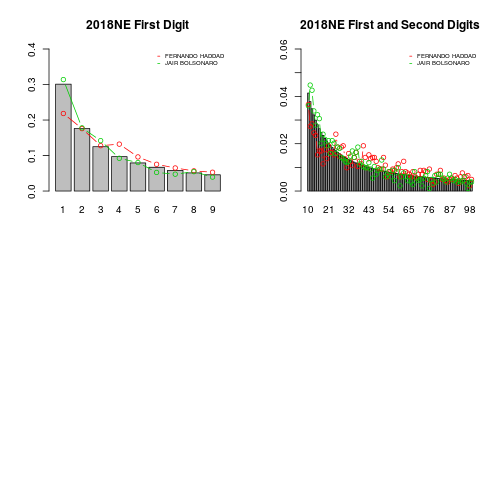
bar = barplot(benford12, ylim = c(0,0.06), main = "2018NE First and Second Digits")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$fands$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



**A brief look at the first round**

Finally, let’s have a look at the first round of the 2018 elections in the NE and SE regions to see if there is something new. The first plot is for the SE region. It also shows the votes for the third place candidate. The pattern is the same as the second round. However, since Fernando Haddad and Ciro Gomes had similar results, if the elections were fraudulent it was probably to influence the results of Jair Bolsonaro. It is impossible to know if the fraud was to increase or decrease the number of votes. However, most critiques on the electronic system came from him and his supporters and he is also the author of the printed vote bill.

data2018SEfr = prepare\_data(filter(elections2018,SIGLA\_UF%in%SE),1,3)

dt = data2018SEfr

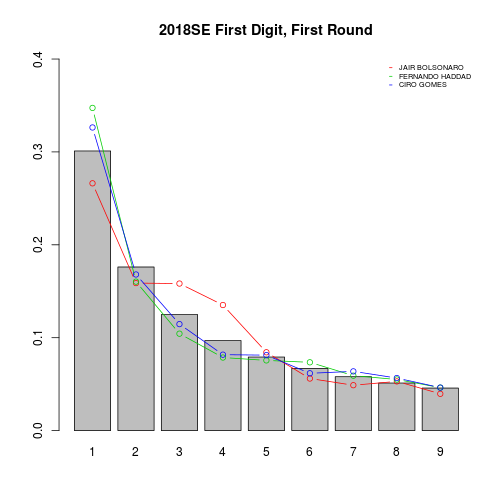
bar = barplot(benford1, ylim = c(0,0.4), main = "2018SE First Digit, First Round")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



Finally, the NE region. We don’t know if we can get any new information from that. The worker’s party candidate has the same weird pattern from previous elections and from the second round of the 2018 elections. However, in the second round the first digit equals one was closer to 0.2 and now it is closer to the Benford’s value of 0.3. Bolsonaro had similar results from the second round with a little more variation in smaller digits. We will not risk to say anything about the other candidate because we have nothing to compare it with.

data2018NEfr = prepare\_data(filter(elections2018,SIGLA\_UF%in%NE),1,3)

dt = data2018NEfr

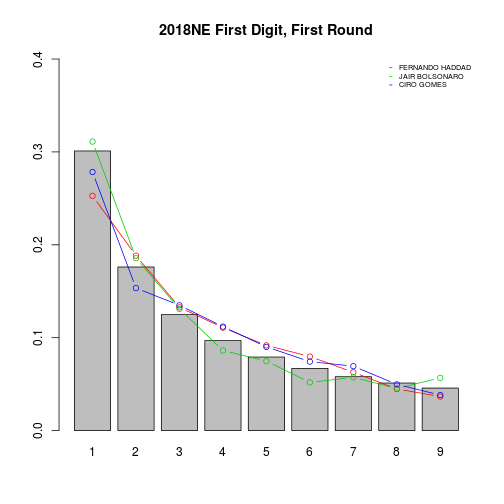
bar = barplot(benford1, ylim = c(0,0.4), main = "2018NE First Digit, First Round")

for(i in 1:length(dt)){

lines(bar,dt[[i]]$first$prop,type ="b", col = i+1)

}

legend("topright", legend = names(dt), col = c(2:(length(dt)+1)), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



**Conclusion**

We think that the main conclusion here is that the Benford’s law alone does not prove anything. It still need the narrative and other evidence. In my opinion, the lack of transparency in the electronic system is a bigger problem than what the data is saying through the Benford’s law. If the elections were more transparent any suspicious results one could find could be further investigated. However, We don’t think that is the case.

**Downloading the data**

The data from 2002 to 2014 can be downloaded with the electionsBR package. The 2018 data must be downloaded manually.

######## Download Data from TSE ##########

library(electionsBR)

elections2010 = electionsBR::vote\_mun\_zone\_fed(2010)

elections2014 = electionsBR::vote\_mun\_zone\_fed(2014)

elections2002 = electionsBR::vote\_mun\_zone\_fed(2002)

elections2006 = electionsBR::vote\_mun\_zone\_fed(2006)

After you download all the data, save the tables in a .csv or a .rda to avoid downloading it again. The 2018 data can be downloaded from:

<http://agencia.tse.jus.br/estatistica/sead/odsele/votacao_candidato_munzona/votacao_candidato_munzona_2018.zip>

The presidential data must be arranged to be in the same format as the other years:

library(tidyverse)

elections2018 = read.csv("votacao\_candidato\_munzona\_2018\_BR.csv",sep =";",header = TRUE,fileEncoding = "latin1")

## select one the variables we used ##

elections2018 = elections2018 %>% select(DS\_CARGO,SG\_UF,QT\_VOTOS\_NOMINAIS,NM\_MUNICIPIO,NR\_ZONA,NM\_URNA\_CANDIDATO,NR\_TURNO) %>%

filter(QT\_VOTOS\_NOMINAIS>0)

## unify variable names

colnames(elections2018) = c("DESCRICAO\_CARGO","SIGLA\_UF","TOTAL\_VOTOS","NOME\_MUNICIPIO","NUMERO\_ZONA","NOME\_URNA\_CANDIDATO","NUM\_TURNO")

elections2018$DESCRICAO\_CARGO = toupper(elections2018$DESCRICAO\_CARGO)

**Function to arrange data and other metadata**

The code below shows the function we used to arrange de data. It filters for president, the desired round and the number of candidates. If you chose 3 candidates it will select the three most voted. Note that if you chose round = 2 the number of candidate must be 2 or 1. The function also extracts the first and second digits for each zone and calculates the proportion of each digit. It returns a list where each element is named after a candidate and each candidate has three data.frames: for the first digit, second digit and first and second digits together.

### States in each Brazilian region

SE = c("SP","MG","ES","RJ")

NE = c("BA","PE","CE","MA","AL","RN","PA","PI","SE")

S = c("PR", "RS", "SC")

CO = c("GO","DF","MT","MS")

N = c("AM", "RR", "AP", "PA","TO","RO","AC")

regions = list(SE=SE,NE=NE,S=S,CO=CO,N=N)

prepare\_data = function(data, round=1, ncandidates=2)

{

## select only second round president and create one table for each candidate ##

president = data %>%filter(NUM\_TURNO == round, DESCRICAO\_CARGO == "PRESIDENTE")

candidates = president %>% group\_by(NOME\_URNA\_CANDIDATO)%>%

summarise(votes = sum(TOTAL\_VOTOS)) %>% arrange(desc(votes))

candidates\_data = list()

for(i in 1:ncandidates){

candidates\_data[[i]] = president %>% filter(NOME\_URNA\_CANDIDATO == candidates$NOME\_URNA\_CANDIDATO[i])

}

names(candidates\_data) = candidates$NOME\_URNA\_CANDIDATO[1:ncandidates]

for(i in 1:ncandidates){

aux = candidates\_data[[i]]

rem = ifelse(aux$TOTAL\_VOTOS>=10,1,NA)

candidates\_data[[i]]$fd = as.numeric(str\_sub(aux$TOTAL\_VOTOS,1,1))

candidates\_data[[i]]$sd = as.numeric(str\_sub(aux$TOTAL\_VOTOS,2,2))\*rem

candidates\_data[[i]]$f2d = as.numeric(str\_sub(aux$TOTAL\_VOTOS,1,2))\*rem

}

output = list()

for(i in 1:length(candidates\_data)){

aux = candidates\_data[[i]]

d1 = aux%>% group\_by(fd) %>% count()

d2 = (aux%>% group\_by(sd) %>% count())[1:10,]

d12 = (aux%>% group\_by(f2d) %>% count())[1:90,]

n1 = nrow(aux)

n2 = n12 = nrow(filter(aux,![is.na](http://is.na)(sd)))

df1 = data.frame(1:9,d1[,2]/n1,n1)

df2 = data.frame(0:9,d2[,2]/n2,n2)

df12 = data.frame(10:99,d12[,2]/n12,n12)

colnames(df1) = colnames(df2) = colnames(df12) = c("digit", "prop", "n")

dflist = list(first = df1, second = df2, fands = df12)

output[[i]] = dflist

}

names(output) = names(candidates\_data)

return(output)

}

**Code to induce fraud**

Here we will induce fraud in the 2002 data to see if the Benford’s law can capture it by adding 1000 votes in all electoral zones. We know this is not a realistic fraud because some electoral zones are very small, but this test is just to see how the fraudulent data fits to the Benford’s law. Feel free to try different types of fraud.

## create fraudulent data ##

elections2002fake = elections2002 %>%

mutate(TOTAL\_VOTOS = TOTAL\_VOTOS+1000)

## Prepare the real data and the fraudulent data for the Benford's law ##

data2002 = prepare\_data(elections2002, round = 2, ncandidates = 2)

data2002fake = prepare\_data(elections2002fake, round = 2, ncandidates = 2)

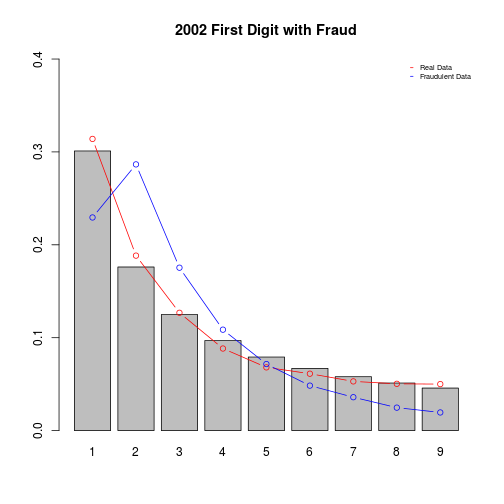
Now let us have a look at what happens in the fraudulent data compared to the real data, just for the first candidate (Lula) to save some space. Feel free to look at the second candidate by changing the value [[1]] to [[2]]. The three figures below show that the fraudulent data deviates a lot from the Benford’s law. However, if you try very small changes in the data the deviation would be harder to identify.

bar = barplot(benford1, ylim = c(0,0.4), main = "2002 First Digit with Fraud")

lines(bar,data2002[[1]]$first$prop,type ="b", col = 2)

lines(bar,data2002fake[[1]]$first$prop,type ="b", col = 4)

legend("topright", legend = c("Real Data","Fraudulent Data"), col = c(2,4), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

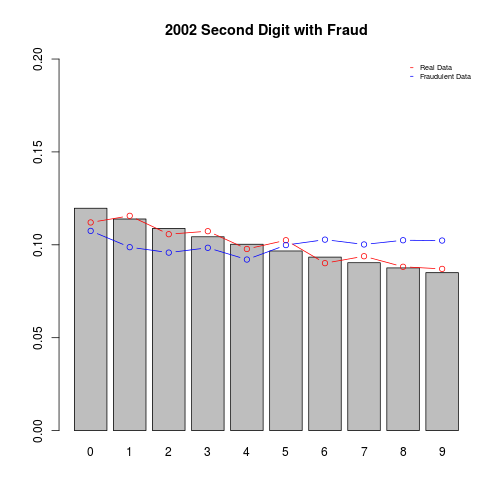


bar = barplot(benford2, ylim = c(0,0.2), main = "2002 Second Digit with Fraud")

lines(bar,data2002[[1]]$second$prop,type ="b", col = 2)

lines(bar,data2002fake[[1]]$second$prop,type ="b", col = 4)

legend("topright", legend = c("Real Data","Fraudulent Data"), col = c(2,4), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")

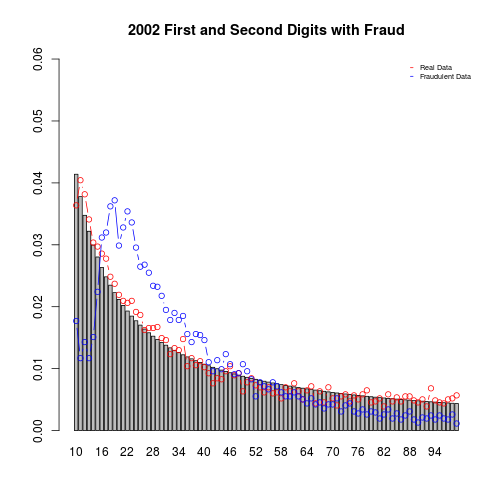


bar = barplot(benford12, ylim = c(0,0.06), main = "2002 First and Second Digits with Fraud")

lines(bar,data2002[[1]]$fands$prop,type ="b", col = 2)

lines(bar,data2002fake[[1]]$fands$prop,type ="b", col = 4)

legend("topright", legend = c("Real Data","Fraudulent Data"), col = c(2,4), cex = 0.6, seg.len = 0.4 ,lty = 1,bty = "n")



Now we are going to show a justification for not using the chi2 test through all the post. The test statistic is defined as follows:

\displaystyle n\sum_{i=1}^9 \frac{(\hat{p}_i - p_i)^2}{p_i} \sim \chi_8^2

where \hat{p}_iis the estimated proportion of digit ifrom de data and p_iis the proportion of digit iif given by the Benford’s law. The distribution is a chi2 with 8 degrees of freedom. we calculated the p-values for this test below to show that it rejects the real data and the fraudulent data.

chireal = data2002[[1]]$first$n[1]\*sum(((data2002[[1]]$first$prop - benford1)^2)/(benford1))

chifraud = data2002fake[[1]]$first$n[1]\*sum(((data2002fake[[1]]$first$prop - benford1)^2)/(benford1))

1-pchisq(chireal,8)

## [1] 1.215743e-10

1-pchisq(chifraud,8)

## [1] 0