Problem Set 2: Election Fraud in Russia

Alin Ierima

Figure 1: Protesters in the Aftermath of the 2011 State Duma Election.

The poster says, 'We don't believe Churov! We believe Gauss!' Churov is the head of the State Electoral Commissions and Gauss refers to a 18th century German mathematician, Carl Friedrich Gauss, whom the Gaussian (Normal) distribution was named after. Source: http://darussophile.com/2011/12/measuring-churovs-beard/. [This source includes many additional analyses.]

In this exercise, we use the rules of probability to detect election fraud by examining voting patterns in the 2011 Russian State Duma election. (The State Duma is the federal legislature of Russia.) This exercise is based on:

Arturas Rozenas (2016). *Inferring Election Fraud from Distributions of Vote-Proportions.* Working Paper.

The ruling political party, United Russia, won this election, but faced many accusations of election fraud, which the Russian government denied. Some protesters highlighted irregular patterns of voting as evidence of election fraud, as shown in the Figure. In particular, protesters pointed out the relatively high frequency of common fractions such as 1/4, 1/3, and 1/2 in the official vote shares.

We use official election results, contained in the object called `russia2011`, which comes in a data frame format and as a part of the `fraud.RData` file. Use the `russia2011` data to investigate whether there is any evidence for election fraud. Hint: the `fraud.RData` file can be loaded using the `load("data/filename.RData")` function. After the `load()` function you can use the `ls()` function to see which datasets were loaded in the R environment.

In addition to `russia2011`, the `fraud.RData` file contains election results from the 2003 Russian Duma election `russia2003`, the 2012 Russian presidential election `russia2012`, and the 2011 Canadian election `canada2011` as separate objects saved in data frame format. Each of these data sets has the same variables, described in the table below.

Name	Description
`N`	Total number of voters in a precinct
`turnout`	Total number of turnout in a precinct
`votes`	Total number of votes for winner (United Russia) in a precinct

Question 1 (20 points)

To analyze the 2011 Russian election results (`russia2011`)

- create a new variable called `rvote` which specified the vote share for United Russia - the proportion of votes for United Russia from all those who turned out in a precinct. *(5 points)*
- Identify the 10 most frequently occurring fractions for the vote share. Shortly discuss your findings in the text. Hint: you can use the `table()` function here and sort the table output to go from highest number of cases to lowest (decreasing order). Show the 10 most frequent proportions, otherwise the table output might be too long. *(5 points)*
- Create a histogram that sets the number of bins to the **number** of *unique* fractions, with one bar created for each uniquely observed fraction, to differentiate between similar fractions like 1/2 and 51/100. This can be done by using the `breaks` argument in the `hist` function. *(5 points)*
- Discuss in the text what does this histogram look like in terms of fractions with low numerators and and low denominators such as 1/2, 1/3, 1/4, 2/3? Note that 1/3 is equal to roughly 0.333, 3/7 to roughly 0.429, 4/7 to roughly 0.571, 6/9 to roughly 0.75 (5 points)

Answer 1

```r
load("data/fraud.RData")
ls() ## check what objects qxist in your workspace
```

```
## [1] "canada2011" "russia2003" "russia2011" "russia2012"
```

For your information: first few observations of United Russia's vote share as a proportion of the voters who turned out (variable `rvote`)

```r
russia2011$rvote = russia2011$votes / russia2011$turnout
rvote_table = table(russia2011$rvote)
head(russia2011$rvote)
```

```
## [1] 0.5797101 0.5745721 0.6919431 0.4135802 0.7099237 0.5970149
```

Number of times different vote shares for United Russia occurred across precincts (10 most frequent occurring fractions

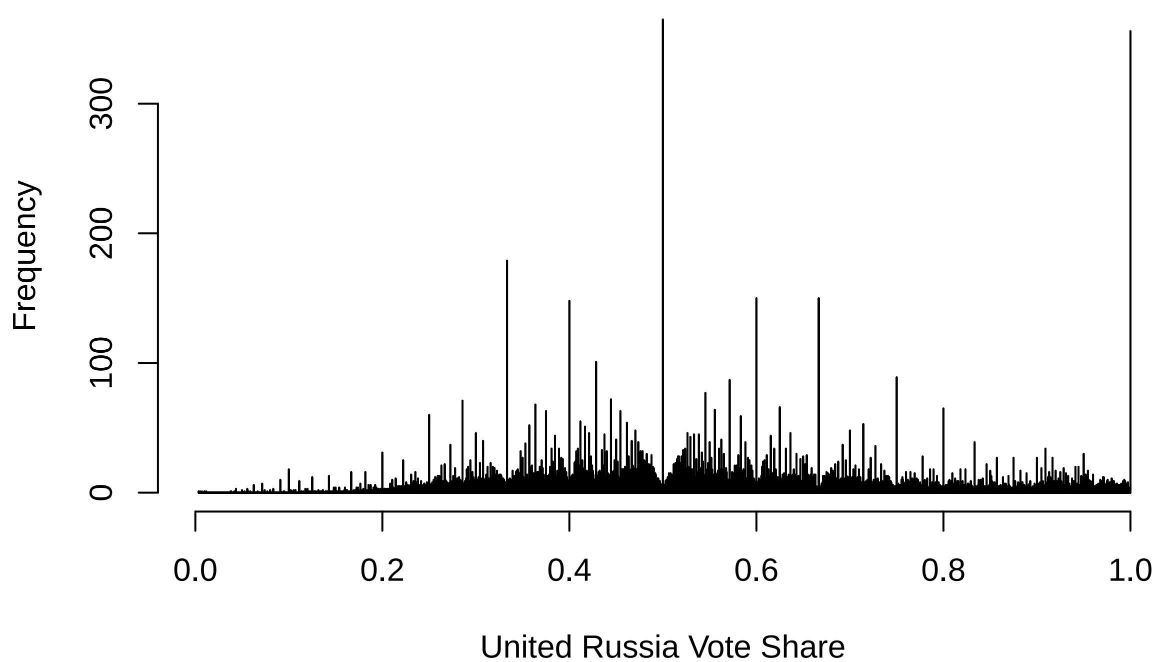
```r
head(sort(rvote_table, decreasing = TRUE), 10)
```

```
##
##               0.5                 1 0.333333333333333               0.6
##               365               356               179               150
## 0.666666666666667               0.4 0.428571428571429              0.75
##               150               148               101                89
## 0.571428571428571 0.545454545454545
##                87                77
```

Insert your answer here: The histogram shows that United Russia receiving 50% or 100% of the votes is by far the most popular outcome, followed by other other fractions with low numerators and denominators which should be really unlikely, probabilistically. This indicates there might have been fraud, however we cannot say for sure at this stage.

Histogram

```r
hist(russia2011$rvote, breaks = length(unique(russia2011$rvote)), xlim = c(0,1), main = NULL, xlab = "Un
```



Question 2 (40 points)

Assume the following probability model:

- **Assumption 1:** Turnout for a precinct is binomially distributed, with size equal to the number of voters in the precinct and success probability equal to its observed turnout rate.

- **Assumption 2:** United Russia's vote counts (how many people in the precinct vote for United Russia) in a precinct is binomially distributed with size (or independent trials) equal to the *number* of voters whom we *simulated to turn out* in the previous step and success probability equal to the precinct's *observed* vote share for United Russia. Let us use a concrete example to illustrate this probability model:

Given the above 2 assumptions (Assumption 1 and Assumption 2):

1. Conduct a Monte Carlo simulation under the above 2 assumptions. *(20 points)*
2. 1000 simulated elections should be sufficient. (Be aware that this may be computationally intensive code. Write your code for a small number of simulations (5 or 10) for all exercises in this problem set to test before running all 1000 simulations! Do the 1000 simulations as the very last change to your script.)
3. What are the 10 most frequent United Russia's vote share values? Discuss this in the text. *(10 points)*
4. Create a histogram of the simulated United Russia's vote shares, similar to the one in the previous question. Comment on the results. *(10 points)*

Answer 2

First few observation in the variable `rturn`

```
russia2011$rturn = russia2011$turnout / russia2011$N
head(russia2011$rturn, 6)
```

```
## [1] 0.7709497 0.6794020 0.6028571 0.5785714 0.8136646 0.6232558
```

First few observation in the variable `rvote`

```
head(russia2011$rvote)
```

```
## [1] 0.5797101 0.5745721 0.6919431 0.4135802 0.7099237 0.5970149
```

dimensions of the `final_vote_share` object:

```
russia2011_4 = russia2011[1:4,]

final_vote_share = NULL

set.seed(123456)

for (i in 1:1000) {
  simulated_turnout = rbinom( n = nrow(russia2011), size = russia2011$N, prob = russia2011$rturn)
```

```
  simulated_votes = rbinom( n = nrow(russia2011), size = simulated_turnout, prob = russia2011$rvote)

 simulated_vote_share = simulated_votes / simulated_turnout

 final_vote_share = cbind(final_vote_share, simulated_vote_share)

}
```

If you followed the instructions above you will end up with 1000 repetitions. One loop round will produce 94995 simulations, which you will combine with the next round of calculations. In this way your object `final_vote_share` where you combined the 1000 iterations of 94995 simulations (1 for each precinct), each row is a given precinct, each column is a given repetition. In a sense you will simulate the `russia2011` data with the 94995 precincts 1000 times. You should thus have 94995 rows (or precinct) and 1000 columns (or simulations). (The loop will take about 2 minutes to run in the console)

Number of rows and columns in the `final_vote_share` object

```
dim(final_vote_share)
```

```
## [1] 94995  1000
```

First 5 precincts and first 10 simulations for each precinct from the `final_vote_share` object.

```
head_final_vote_share = head(as.data.frame(final_vote_share[, 1:10]),5)
colnames(head_final_vote_share) = c("Simulation 1","Simulation 2","Simulation 3","Simulation 4","Simula
head_final_vote_share
```

```
##   Simulation 1 Simulation 2 Simulation 3 Simulation 4 Simulation 5 Simulation 6
## 1    0.5338346    0.5851852    0.5797101    0.5422535    0.5616438    0.5724638
## 2    0.5635910    0.5334988    0.5417722    0.5666667    0.5621891    0.5829384
## 3    0.6782178    0.7125891    0.6873508    0.7100737    0.7061503    0.7026379
## 4    0.4271845    0.4071856    0.4112426    0.3656958    0.3943218    0.4086687
## 5    0.7076923    0.6370370    0.6985294    0.6875000    0.6803279    0.7120000
##   Simulation 7 Simulation 8 Simulation 9 Simulation 10
## 1    0.5845070    0.6319444    0.4861111     0.6074074
## 2    0.5223881    0.5447942    0.5662651     0.5731132
## 3    0.6839623    0.7132701    0.6804598     0.6489104
## 4    0.4126984    0.4342508    0.4805195     0.3836478
## 5    0.6865672    0.7352941    0.7218045     0.7318841
```

10 most frequent proportions, decreasing from highest frequency (Note that the exact number of proportions will vary from simulation to simulation. You might have slightly different results. This step might take about 2 minutes to show in the console)

```
proportions_frequency = table(final_vote_share)
head(sort(proportions_frequency, decreasing = TRUE), 10)
```

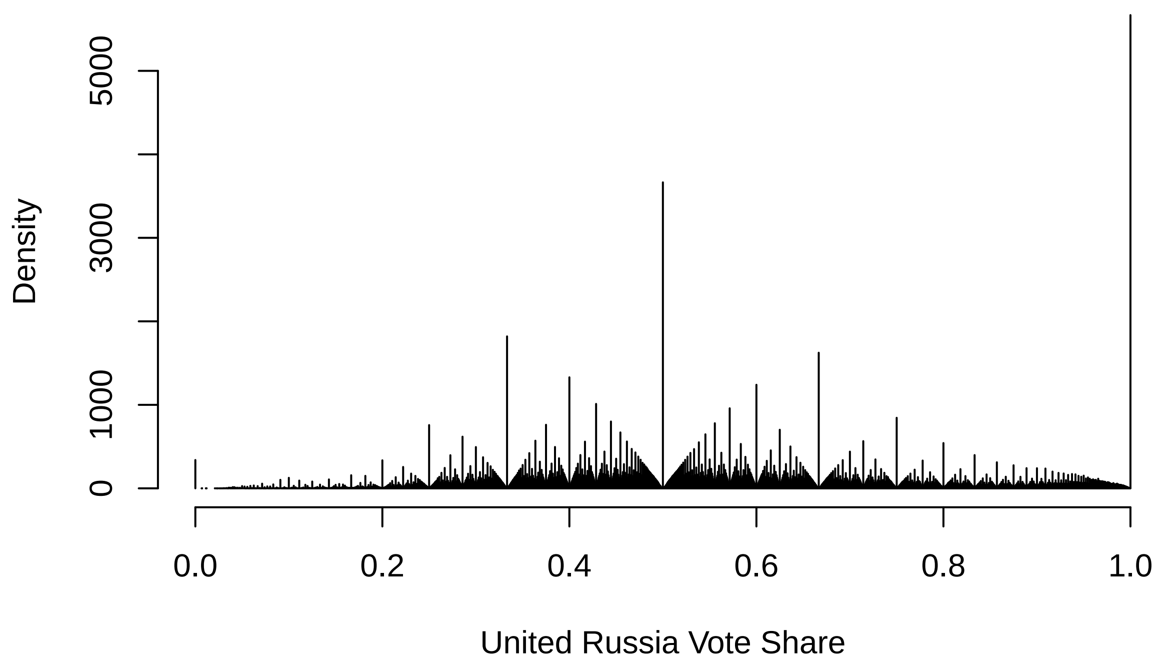
```
## final_vote_share
##                1               0.5 0.333333333333333 0.666666666666667
##           538412            348170            172895            154284
##              0.4               0.6 0.428571428571429 0.571428571428571
##           126340            117843             96065             91073
##             0.75 0.444444444444444
##            80289             76070
```

Remember that:

- 1/2 is equal to 0.5
- 1/3 is equal to 0.3333333
- 2/3 is equal to 0.6666667
- 2/5 is equal to 0.4
- 3/7 is equal to 0.4285714
- 3/5 is equal to 0.6
- 4/7 is equal to 0.5714286
- 6/9 is equal to 0.6666667
- 11/25 is equal to 0.44

Histogram of the simulated United Russia vote shares (this can take about 30 seconds to plot)

```
hist(final_vote_share, breaks = length(unique(final_vote_share)), main = NULL, xlab = "United Russia Vo
```



Insert your answer here: We can see that the simulated elections are quite similar to the observed data, as fractions such as 1/2 or 1/3 have high relative frequencies (a difference being that in the observed data the most popular result is 1/2 followed by 1, and in simulated data it is the other way around). This similarity indicates that the simulation aligns with the observed pattern or, in other words, the election results are not unexpected. While this suggests that the outcome was statistically likely, it does not necessarily mean there was no fraud, it is just inconclusive at this stage.

Question 3 (20 points)

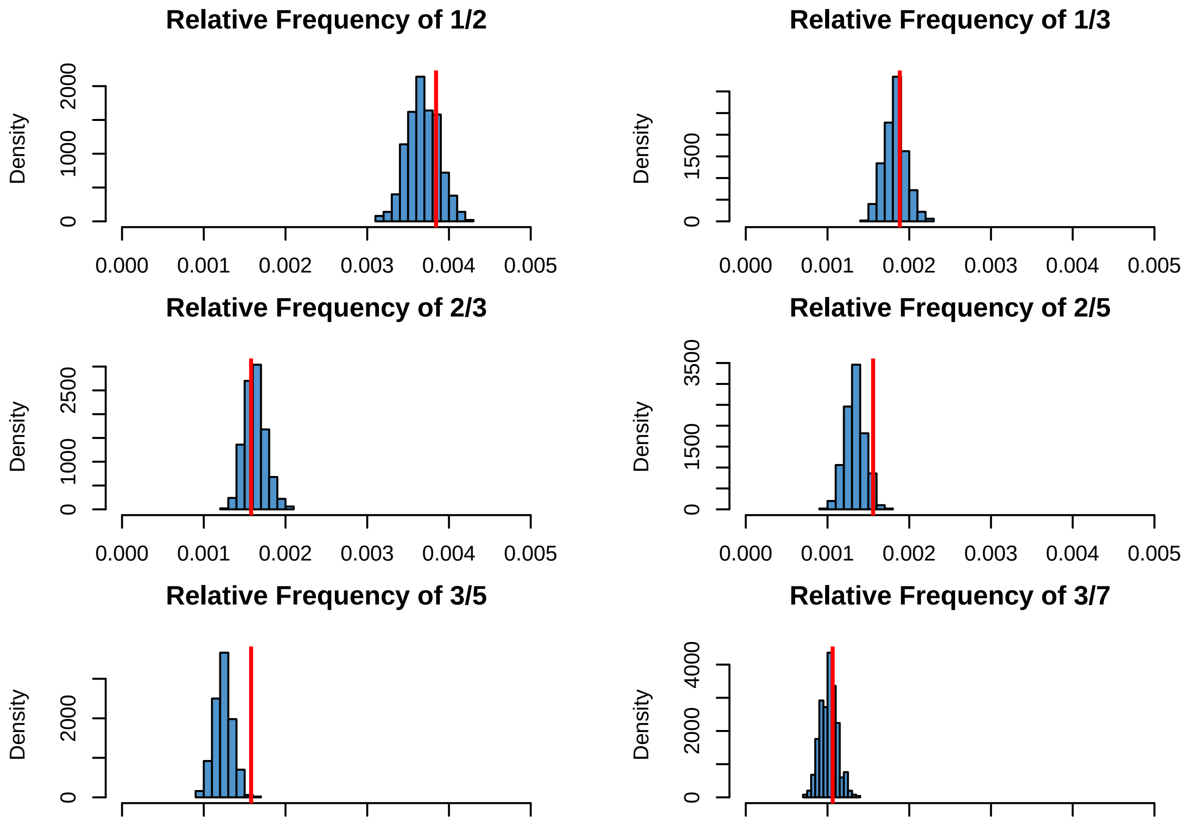
To judge the Monte Carlo simulation results against the actual results of the 2011 Russian election, we will compare the *observed* frequency (in percentages) of a given fraction with its simulated counterpart.

The `vote_share_final` object has 94995 rows (precincts) and 1000 columns (simulations). In a sense, each column contains the results from one simulation of all precincts. Consequently we have 1000 simulated versions of the real voting results (94995 precincts, with simulated results 1000 times). **For each** of the 1000 simulated versions of the real data (*for each of the 1000 columns*), we can calculate the frequency of a given fraction (say the frequency of UR vote equal to exactly 1/3) and plot the 1000 frequencies. The frequency of a given fraction should not be in absolute numbers, but in percentages (number of times the fraction occurs/total number of cases) In such a way we can get 1000 frequencies of a given fraction and then plot the distribution of these frequencies. Finally, we can also indicate the observed frequency of this voting fraction (e.f. 1/3) from the `russia2011` dataset.

- Compare these frequency distributions with the fractions' frequency from the *actual 2011_election*. Indicate the observed fraction frequency with a red vertical line. (*5 points*)
- Briefly interpret the your findings in the text. (*5 points*)

Answer 3

```
#pray for my computer
final_vote_500 = final_vote_share[, 1:500]

most_frequent_fractions = c(1/2, 1/3, 2/3, 2/5, 3/5, 3/7)
names(most_frequent_fractions) = c("1/2", "1/3", "2/3", "2/5", "3/5", "3/7")

histograms = list()

reallife_frequencies = sapply(most_frequent_fractions, function(frac) {
  sum(russia2011$rvote == frac) / nrow(russia2011)
})

par(mfrow = c(3, 2), mar = c(1, 5, 4, 2) + 0.1)

for (i in 1:length(most_frequent_fractions)) {
  fraction = most_frequent_fractions[i]
  fraction_name = names(most_frequent_fractions)[i]
  fraction_frequency = colMeans(final_vote_500 == fraction, na.rm = TRUE)
  histogram = hist(fraction_frequency,
                   main = paste("Relative Frequency of", fraction_name),
                   xlab = "Proportion (%)",
                   ylab = "Density",
                   col = "steelblue3",
                   xlim = c(0.000, 0.005),
                   freq = F)
  histograms[[fraction_name]] = histogram
    abline(v = reallife_frequencies[i], col = "red", lwd = 2)
}
```



Insert your answer here: Comparing the frequency distributions with the observed fraction frequency we delve much deeper into the differences between the simulated and observed data. While the relative frequency is quite typical in some fractions (1/2, 1/3, 2/3, 3/7), there is also an unusual outcome in the case of frequencies for fractions 2/5 and 3/5, for which the observed frequency was very statitstically unlikely, albeit still possible. However, it lends some support to the hypothesis that there have been some voting irregularities in the election.

Question 4 (20 points + 20 Bonus)

Now, let us compare the relative frequency of observed fractions with the simulated ones beyond the six fractions examined in the previous question.

1. create intervals between 0 and 1 (the last interval should be 1) in steps of 0.01. You will use these intervals to create bin sizes of 0.01. One bin will range from 0 to 0.01 (0 including, 0.01 excluding), the next been from 0.01 to 0.02 (0.01 including, 0.02 excluding), the next from 0.02 to 0.03 etc. *(2 points)*
2. In this step focus on the *observed data* - `russia2011` and compute the number of observations (UR vote shares) that fall into each bin of 0.01 in the `russia2011` data. This means you are asked to compute in how many precincts do we find a vote share between 0 and 0.01 (including 0, but excluding 0.01) for UR, the same for vote shares between 0.01 and 0.02 (including 0.01 but excluding 0.02) etc. Note that the right number of each interval (bin), should be excluded. *(10 points)*
3. Using the number of of observations that fall within each bin, calculate the *proportion* of observations that fall within each bin of 0.01. In essence, here we want to know what percentage of precincts (from a total of 94995) have UR vote share which are between 0 and 0.01, between 0.01 and 0.02 and so on. *(3 points)*

4. In this step you are asked to repeat step 2 and 3 for the simulated data - hence 1000 times. *(10 points)* Recall that your simulated data has 94995 rows (or precincts), and 1000 columns (or simulations). This in practice means that you have 1000 simulated versions of the `russia2011` data, or 1000 simulated elections. To repeat step 2 and 3 (calculate the number and then proportions of observations with UR vote shares falling within given bin of size 0.01) 1000 times you can utilize a loop. Again, this would be rather inefficient and time costly. Instead, you can create a function which does step 2 and 3 and then apply this function to each *column* (each simulation) in your simulated data. Hint: the `apply()` function allows you to apply a function by rows or columns. We will guide you through this process.

5. After you repeated step 2 and step 3 for the 1000 simulated elections (or for every of the 1000 columns in the simulated data), examine whether or not the observed proportion (from `russia2011` data) falls within the 2.5 and 97.5 percentiles of the corresponding simulated proportions. To do so follow the below steps:

- calculate the 2.5 and 97.5 percentiles of the frequencies of UR vote shares falling within each interval of 0.01. For example for the interval [0, 0.01] you will have 1000 numbers, which indicate how many UR vote shares fall within this interval. You will have 1000 frequencies for this interval, because you have 1000 simulations. What you are asked is to calculate the 2.5 and the 97.5 percentile from these 1000 frequencies for each interval. Hint: you can use the function `percentile()`. The idea is to show the distribution of these frequencies for each interval in a meaningful way - the middle 95% of all simulated frequencies and then check whether the observed frequency of United Russia vote shares between 0 and 0.01 falls inside the middle 95% or outside of it. *(5 points)*
- Plot the result with vote share bins on the horizontal axis and the percentage of times these vote shares occurred on the vertical axis. Use blue dots for the observed vote shares and their frequencies. Use lines for the 2.5th and 95th percentiles from the simulated data. This plot attempts to reproduce the one held by protesters in the figure. *(5 points)*

- As a next step count the number of times an observed precinct vote share (the blue dots) falls outside its simulated 95% interval. Hint: when the observed frequencies for each 0.01 interval are smaller than the 2.5th and bigger than the 95th percentiles from the simulated data. Interpret the results. *(5 points)*

Answer 4

desired intervals

```
(intervals = seq(0, 1, 0.01))
```

```
##   [1] 0.00 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12 0.13 0.14
##  [16] 0.15 0.16 0.17 0.18 0.19 0.20 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29
##  [31] 0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.40 0.41 0.42 0.43 0.44
##  [46] 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59
##  [61] 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74
##  [76] 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89
##  [91] 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.00
```

```
interval_hist = hist(russia2011$rvote,
     breaks = intervals,
     right = F,
     plot = F)
```

Create the function

```r
compute_bin_proportions =function(data) {
  hist_data = hist(data, breaks = intervals, plot = FALSE)
  bin_counts = hist_data$counts
  bin_proportions = bin_counts / sum(bin_counts)
  return(bin_proportions)
}
```

Observed frequency of UR vote shares for each 0.01 interval: shares for the first few intervals out of a total of 100 intervals Hint: this calculations are done on the observed data, therefore you should have the same numbers.

```r
observed_bin_proportions = compute_bin_proportions(russia2011$rvote)

head(data.frame(
  interval = intervals[-length(intervals)],
  proportion = observed_bin_proportions)
  ,4)
```

```
##   interval   proportion
## 1     0.00 4.210748e-05
## 2     0.01 1.052687e-05
## 3     0.02 0.000000e+00
## 4     0.03 2.105374e-05
```

Frequencies of simulated UR vote shares for each interval of 0.01. Show the frequencies for the first 7 intervals and 4 simulations

```r
simulated_bin_prop = apply(final_vote_500, 2, compute_bin_proportions) #this is the total

knitr::kable(head(data.frame(
  interval = intervals[-length(intervals)],
  S1 = simulated_bin_prop[, 1],
  S2 = simulated_bin_prop[, 2],
  S3 = simulated_bin_prop[, 3],
  S4 = simulated_bin_prop[, 4])
, 7))
```

interval	S1	S2	S3	S4
0.00	0.0003684	0.0003579	0.0003579	0.0003790
0.01	0.0000105	0.0000105	0.0000000	0.0000105
0.02	0.0000000	0.0000000	0.0000105	0.0000105
0.03	0.0001053	0.0000632	0.0000842	0.0000737
0.04	0.0000632	0.0000421	0.0001158	0.0000421
0.05	0.0000316	0.0000947	0.0001158	0.0001053
0.06	0.0000421	0.0001158	0.0000526	0.0000842

For your information: the dimensions of the results

```r
dim(simulated_bin_prop) # I am doing 500 simulations instead of 1000
```

```
## [1] 100 500
```

Compute 95% intervals for each bin proportion (hence by row)

```r
percentiles = apply(simulated_bin_prop, 1, function(x) quantile(x, c(0.025, 0.975)))
```

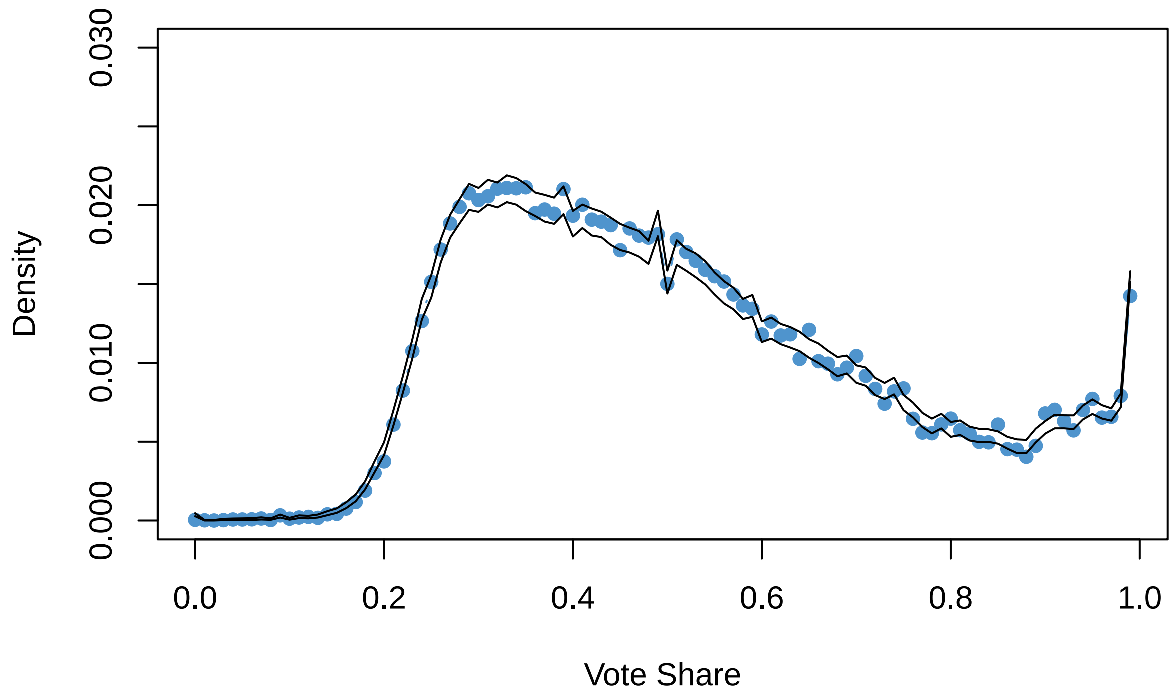
For your information: 2.5% and 97.5% for the first 7 bins

```r
knitr::kable(as.data.frame(percentiles)[,1:7 ])
```

	V1	V2	V3	V4	V5	V6	V7
2.5%	0.0002737	0.00e+00	0.00e+00	0.0000211	0.0000316	0.0000366	0.0000316
97.5%	0.0004632	3.16e-05	3.16e-05	0.0001053	0.0001263	0.0001368	0.0001474

Plot

```r
plot(intervals[-length(intervals)], observed_bin_proportions, type = "b", col = "steelblue3", pch = 16,
   xlab = "Vote Share", ylab = "Density",
   main = NULL,
   ylim = c(0, 0.030))

lines(intervals[-length(intervals)], percentiles[1,], col = "black", lty = 1)
lines(intervals[-length(intervals)], percentiles[2,], col = "black", lty = 1)
```



Number of times observed proportions are outside the simulated 97.5% interval

```
sum(observed_bin_proportions > percentiles[2,] | observed_bin_proportions < percentiles[1,])
```

```
## [1] 31
```

Insert your answer here: The fact that the observed votes fall outside the 95% interval 31 times suggests that there are significant discrepancies between the observed votes and what would be expected if the assumption of clean elections were correct. This lends even stronger support for the hypothesis of electoral fraud in Russia, however it should also be potentially compared with some clean elections in order to have the bigger picture.

Question 5 (Bonus question for 20 points)

- To put the results of the previous question in perspective, apply the procedure developed in the previous question to the 2011 Canadian elections and the 2003 Russian election, where no major voting irregularities were reported. *(10 points)*
- In addition, apply this procedure to the 2012 Russian presidential election, where election fraud allegations were reported. *(5 points)*
- No plot needs to be produced.
- Briefly comment on the results you obtain. *(5 points)*

Answer 5

Since we repeat the same analysis using different data sets, we create a function that implements the analysis.

```
simulate_election = function(data, num_simulations) {
  final_vote_share = NULL

  for (i in 1:num_simulations) {
    simulated_turnout = rbinom(n = nrow(data), size = data$N, prob = data$rturn)
    simulated_votes = rbinom(n = nrow(data), size = simulated_turnout, prob = data$rvote)
    simulated_vote_share = simulated_votes / simulated_turnout
    final_vote_share = cbind(final_vote_share, simulated_vote_share)
  }

  return(final_vote_share)
}
```

Number of times the observed frequency exceeds the [2.5% and 97.5%] range of the simulated frequency in the:

Canadian 2011 elections: number of times observed proportions are outside the simulated 95% interval

```
set.seed(123456)
canada2011$rvote = canada2011$votes / canada2011$N
canada2011$rturn = canada2011$turnout / canada2011$N

final_vote_share_canada2011 = simulate_election(canada2011, num_simulations = 250)


simulated_bin_prop_canada2011 = apply(final_vote_share_canada2011, 2, compute_bin_proportions)

percentiles_canada =  apply(simulated_bin_prop_canada2011, 1, function(x) quantile(x, c(0.025, 0.975)))

observed_bin_proportions_canada2011 = compute_bin_proportions(canada2011$rvote)


sum(observed_bin_proportions_canada2011 > percentiles_canada[2,] | observed_bin_proportions_canada2011
```

```
## [1] 21
```

Russian 2003 elections: number of times observed proportions are outside the simulated 95% interval

```
set.seed(123456)
russia2003$rvote = russia2003$votes / russia2003$N
russia2003$rturn = russia2003$turnout / russia2003$N

final_vote_share_russia2003 = simulate_election(russia2003, num_simulations = 250)


simulated_bin_prop_russia2003 = apply(final_vote_share_russia2003, 2, compute_bin_proportions)
```

```r
percentiles_russia2003 =  apply(simulated_bin_prop_russia2003, 1, function(x) quantile(x, c(0.025, 0.975

observed_bin_proportions_russia2003 = compute_bin_proportions(russia2003$rvote)


sum(observed_bin_proportions_russia2003 > percentiles_russia2003[2,] | observed_bin_proportions_russia20
```

```
## [1] 19
```

Russian 2012 elections: number of times observed proportions are outside the simulated 95% interval

```r
set.seed(123456)
russia2012$rvote = russia2012$votes / russia2012$N
russia2012$rturn = russia2012$turnout / russia2012$N

final_vote_share_russia2012 = simulate_election(russia2012, num_simulations = 250)


simulated_bin_prop_russia2012 = apply(final_vote_share_russia2012, 2, compute_bin_proportions)

percentiles_russia2012 =  apply(simulated_bin_prop_russia2012, 1, function(x) quantile(x, c(0.025, 0.975

observed_bin_proportions_russia2012 = compute_bin_proportions(russia2012$rvote)


sum(observed_bin_proportions_russia2012 > percentiles_russia2012[2,] | observed_bin_proportions_russia20
```

```
## [1] 26
```

Russian 2011 elections: number of times observed proportions are outside the simulated 95% interval

```r
set.seed(123456)
russia2011$rvote = russia2011$votes / russia2011$N
russia2011$rturn = russia2011$turnout / russia2011$N

final_vote_share_russia2011 = simulate_election(russia2011, num_simulations = 250)


simulated_bin_prop_russia2011 = apply(final_vote_share_russia2011, 2, compute_bin_proportions)

percentiles_russia2011 =  apply(simulated_bin_prop_russia2011, 1, function(x) quantile(x, c(0.025, 0.975

observed_bin_proportions_russia2011 = compute_bin_proportions(russia2011$rvote)


sum(observed_bin_proportions_russia2011 > percentiles_russia2011[2,] | observed_bin_proportions_russia20
```

```
## [1] 21
```

Insert your answer here: In this answer, the simulations were reduced to 250 for ease of computer generation. The 2003 Russian elections and the Canadian 2011 elections seem to have been the ones where the observed values fall the most within the 95%, however they still report quite sizeable discrepancies 19, respectively 21 times. On the other hand, the 2012 Russian elections' result is very noticeably different from the other two, as their results fell out of the 95% 26 times. Judging by the fact those elections were criticised for electoral fraud while the others do not, this represents quite solid proof to me that the elections were indeed fraudulent, whereas the 2003 Russian elections were not, at least to a significant degree.

Evaluation

- 4 questions for a total of 100 points
- 1 bonus question for a total of 40 points (split between question 4 and question 5)