Election Fraud?

AUTHOR
Pari Pandya

Load Libraries

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ────────────────────────
tidyverse 2.0.0 ──
✔ dplyr     1.1.3     ✔ readr     2.1.4
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.4.4     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.0
✔ purrr     1.0.2
── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>)
to force all conflicts to become errors
```

```
library(stat20data)
library(patchwork)
data(iran)
New_York_Data_set <- read_csv("https://raw.githubusercontent.co
```

```
Rows: 806 Columns: 6
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
chr (4): county, office, party, candidate
dbl (1): votes
lgl (1): district

ℹ Use `spec()` to retrieve the full column specification for
this data.
ℹ Specify the column types or set `show_col_types = FALSE` to
quiet this message.
```

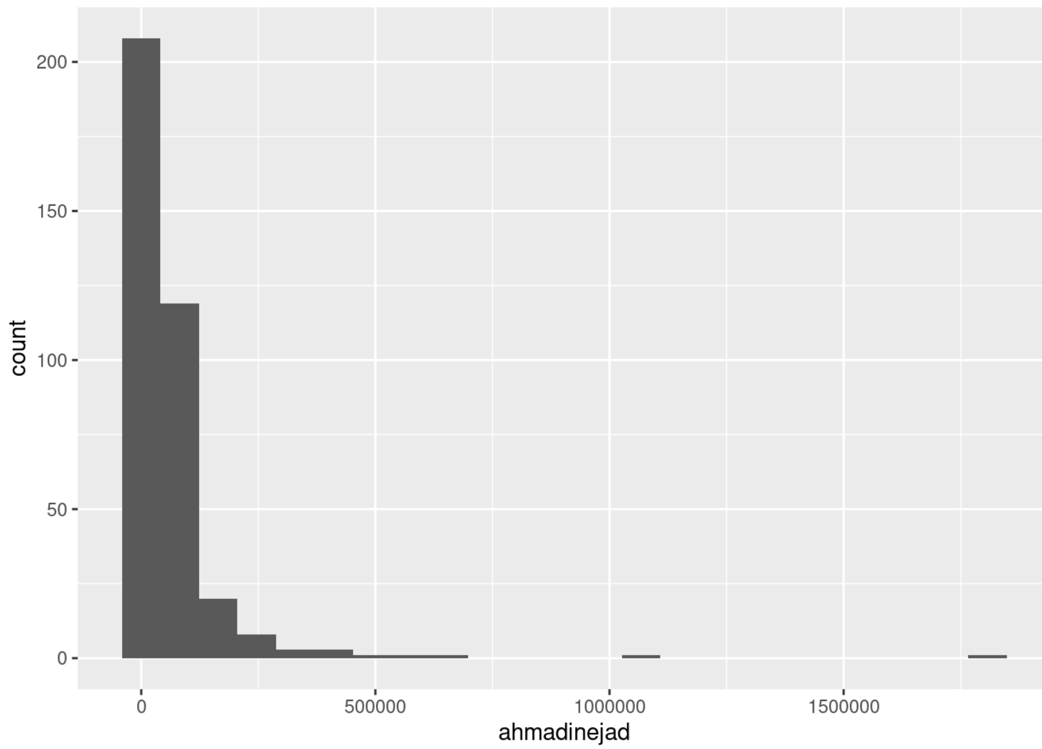
Question 1:

The unit of observation in the Iran data frame is cities.

Question 2:

Plot:

```
ggplot(data = iran, mapping = aes(x = ahmadinejad)) +
  geom_histogram(bins = 23)
```



Numerical Summaries:

```
iran %>%
  summarise("Mean" = mean(ahmadinejad),
            "Center" = sd(ahmadinejad))
```

```
# A tibble: 1 × 2
    Mean  Center
   <dbl>   <dbl>
```

```
1 66981. 130010.
```

Interpretation:

This plot clearly shows a rightward skew. The lack of a normal distribution may likely be an indicator of biased or tampered data.

Question 3:

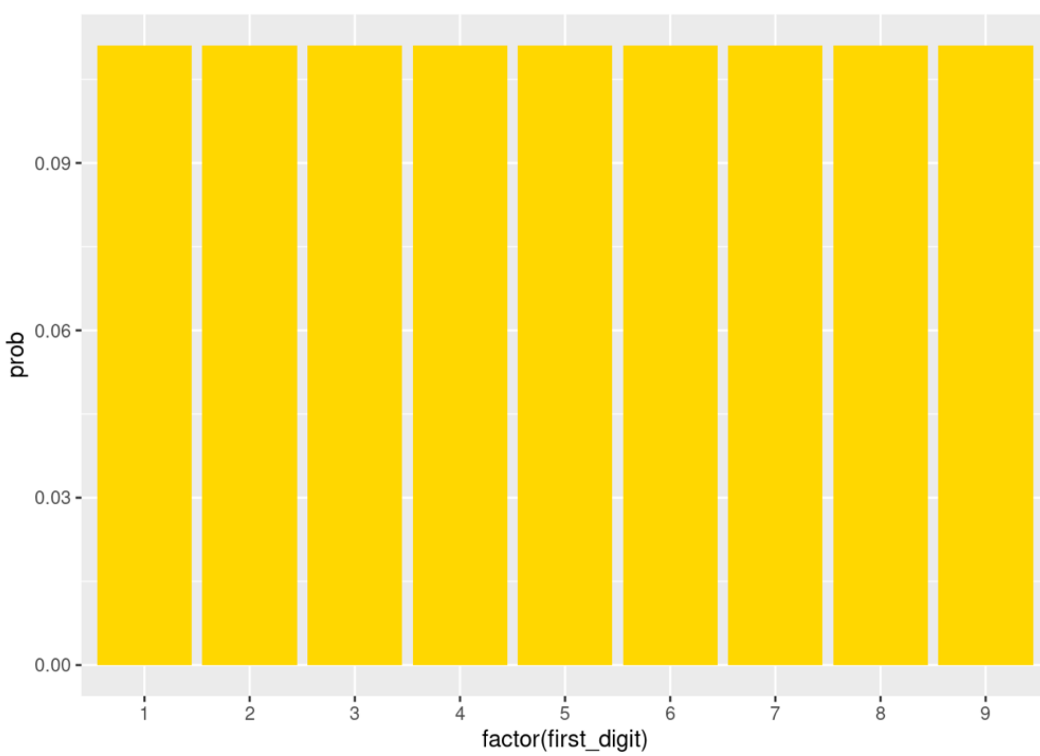
Mutating and Saving the Prob Column:

```
fd_unif <- data.frame(first_digit = seq(1, 9))

fd_unif <- mutate(fd_unif, prob = 1/9)
```

Plot:

```
fd_unif %>%
  ggplot(aes(x = factor(first_digit),
             y = prob)) +
  geom_col(fill = "gold")
```



Question 4:

```
fd_unif <- mutate(fd_unif, "expected_val" = (prob) * (first_dig

fd_unif %>%
  summarize(expected = sum(expected_val))
```

```
  expected
1        5
```

Question 5:

```
fd_unif <- mutate(fd_unif, "x_squ" = (first_digit)*(first_digit

fd_unif %>%
  summarize(variance = sum((x_squ)*(prob)) - 25)
```

```
  variance
1 6.666667
```

Question 6:

```
fd_benford <- data.frame(first_digit = seq(1, 9))

fd_benford <- mutate(fd_benford, prob = log10(1 + (1/first_digi

fd_benford %>%
  summarize(sum_prob = sum(prob) == 1)
```

```
  sum_prob
1     TRUE
```

Question 7:

Expected Value (Benford)

```
fd_benford <- mutate(fd_benford, "expected_val" = (prob) * (firs

expected <- summarize(fd_benford, expected = sum(expected_val))

print(expected)
```

```
  expected
1 3.440237
```

Variance (Benford)

```
fd_benford <- mutate(fd_benford, "x_squ" = (first_digit)*(first_

fd_benford %>%
  summarize(variance = sum((x_squ)*(prob)) - (3.440237)^2)
```

```
  variance
1 6.056512
```

Question 8:

```
fd_benford %>%
  slice_sample(n = 366,
               replace = TRUE,
               weight_by = prob)
```

```
    first_digit       prob expected_val x_squ
1             2 0.17609126    0.3521825     4
2             4 0.09691001    0.3876401    16
3             8 0.05115252    0.4092202    64
4             1 0.30103000    0.3010300     1
5             7 0.05799195    0.4059436    49
6             8 0.05115252    0.4092202    64
7             9 0.04575749    0.4118174    81
8             8 0.05115252    0.4092202    64
9             1 0.30103000    0.3010300     1
10            3 0.12493874    0.3748162     9
11            5 0.07918125    0.3959062    25
12            2 0.17609126    0.3521825     4
13            2 0.17609126    0.3521825     4
14            1 0.30103000    0.3010300     1
```

```
15            1 0.30103000    0.3010300     1
16            8 0.05115252    0.4092202    64
17            3 0.12493874    0.3748162     9
18            2 0.17609126    0.3521825     4
19            3 0.12493874    0.3748162     9
20            4 0.09691001    0.3876401    16
21            1 0.30103000    0.3010300     1
22            5 0.07918125    0.3959062    25
23            8 0.05115252    0.4092202    64
24            8 0.05115252    0.4092202    64
25            1 0.30103000    0.3010300     1
26            5 0.07918125    0.3959062    25
27            2 0.17609126    0.3521825     4
28            5 0.07918125    0.3959062    25
29            5 0.07918125    0.3959062    25
30            1 0.30103000    0.3010300     1
31            4 0.09691001    0.3876401    16
32            1 0.30103000    0.3010300     1
33            4 0.09691001    0.3876401    16
34            2 0.17609126    0.3521825     4
35            7 0.05799195    0.4059436    49
36            8 0.05115252    0.4092202    64
37            4 0.09691001    0.3876401    16
38            2 0.17609126    0.3521825     4
39            3 0.12493874    0.3748162     9
40            1 0.30103000    0.3010300     1
41            2 0.17609126    0.3521825     4
42            6 0.06694679    0.4016807    36
43            1 0.30103000    0.3010300     1
44            2 0.17609126    0.3521825     4
45            2 0.17609126    0.3521825     4
46            4 0.09691001    0.3876401    16
47            4 0.09691001    0.3876401    16
48            1 0.30103000    0.3010300     1
49            4 0.09691001    0.3876401    16
50            5 0.07918125    0.3959062    25
51            1 0.30103000    0.3010300     1
52            1 0.30103000    0.3010300     1
53            3 0.12493874    0.3748162     9
54            4 0.09691001    0.3876401    16
55            1 0.30103000    0.3010300     1
56            3 0.12493874    0.3748162     9
57            2 0.17609126    0.3521825     4
58            1 0.30103000    0.3010300     1
59            8 0.05115252    0.4092202    64
```

60	9	0.04575749	0.4118174	81
61	1	0.30103000	0.3010300	1
62	1	0.30103000	0.3010300	1
63	8	0.05115252	0.4092202	64
64	2	0.17609126	0.3521825	4
65	4	0.09691001	0.3876401	16
66	1	0.30103000	0.3010300	1
67	3	0.12493874	0.3748162	9
68	1	0.30103000	0.3010300	1
69	1	0.30103000	0.3010300	1
70	2	0.17609126	0.3521825	4
71	1	0.30103000	0.3010300	1
72	1	0.30103000	0.3010300	1
73	3	0.12493874	0.3748162	9
74	2	0.17609126	0.3521825	4
75	1	0.30103000	0.3010300	1
76	1	0.30103000	0.3010300	1
77	6	0.06694679	0.4016807	36
78	9	0.04575749	0.4118174	81
79	2	0.17609126	0.3521825	4
80	9	0.04575749	0.4118174	81
81	8	0.05115252	0.4092202	64
82	1	0.30103000	0.3010300	1
83	1	0.30103000	0.3010300	1
84	5	0.07918125	0.3959062	25
85	1	0.30103000	0.3010300	1
86	2	0.17609126	0.3521825	4
87	8	0.05115252	0.4092202	64
88	1	0.30103000	0.3010300	1
89	6	0.06694679	0.4016807	36
90	1	0.30103000	0.3010300	1
91	3	0.12493874	0.3748162	9
92	4	0.09691001	0.3876401	16
93	1	0.30103000	0.3010300	1
94	1	0.30103000	0.3010300	1
95	1	0.30103000	0.3010300	1
96	6	0.06694679	0.4016807	36
97	1	0.30103000	0.3010300	1
98	3	0.12493874	0.3748162	9
99	6	0.06694679	0.4016807	36
100	4	0.09691001	0.3876401	16
101	5	0.07918125	0.3959062	25
102	6	0.06694679	0.4016807	36
103	4	0.09691001	0.3876401	16
104	4	0.09691001	0.3876401	16

105	5	0.07918125	0.3959062	25
106	1	0.30103000	0.3010300	1
107	2	0.17609126	0.3521825	4
108	6	0.06694679	0.4016807	36
109	7	0.05799195	0.4059436	49
110	6	0.06694679	0.4016807	36
111	2	0.17609126	0.3521825	4
112	5	0.07918125	0.3959062	25
113	4	0.09691001	0.3876401	16
114	1	0.30103000	0.3010300	1
115	1	0.30103000	0.3010300	1
116	8	0.05115252	0.4092202	64
117	2	0.17609126	0.3521825	4
118	4	0.09691001	0.3876401	16
119	1	0.30103000	0.3010300	1
120	2	0.17609126	0.3521825	4
121	5	0.07918125	0.3959062	25
122	1	0.30103000	0.3010300	1
123	3	0.12493874	0.3748162	9
124	9	0.04575749	0.4118174	81
125	3	0.12493874	0.3748162	9
126	4	0.09691001	0.3876401	16
127	3	0.12493874	0.3748162	9
128	1	0.30103000	0.3010300	1
129	1	0.30103000	0.3010300	1
130	1	0.30103000	0.3010300	1
131	4	0.09691001	0.3876401	16
132	7	0.05799195	0.4059436	49
133	5	0.07918125	0.3959062	25
134	3	0.12493874	0.3748162	9
135	5	0.07918125	0.3959062	25
136	8	0.05115252	0.4092202	64
137	7	0.05799195	0.4059436	49
138	2	0.17609126	0.3521825	4
139	2	0.17609126	0.3521825	4
140	5	0.07918125	0.3959062	25
141	2	0.17609126	0.3521825	4
142	3	0.12493874	0.3748162	9
143	1	0.30103000	0.3010300	1
144	1	0.30103000	0.3010300	1
145	8	0.05115252	0.4092202	64
146	1	0.30103000	0.3010300	1
147	1	0.30103000	0.3010300	1
148	1	0.30103000	0.3010300	1
149	4	0.09691001	0.3876401	16

150	6	0.06694679	0.4016807	36
151	5	0.07918125	0.3959062	25
152	1	0.30103000	0.3010300	1
153	2	0.17609126	0.3521825	4
154	2	0.17609126	0.3521825	4
155	1	0.30103000	0.3010300	1
156	6	0.06694679	0.4016807	36
157	3	0.12493874	0.3748162	9
158	2	0.17609126	0.3521825	4
159	3	0.12493874	0.3748162	9
160	2	0.17609126	0.3521825	4
161	1	0.30103000	0.3010300	1
162	3	0.12493874	0.3748162	9
163	1	0.30103000	0.3010300	1
164	8	0.05115252	0.4092202	64
165	9	0.04575749	0.4118174	81
166	7	0.05799195	0.4059436	49
167	1	0.30103000	0.3010300	1
168	2	0.17609126	0.3521825	4
169	2	0.17609126	0.3521825	4
170	6	0.06694679	0.4016807	36
171	6	0.06694679	0.4016807	36
172	5	0.07918125	0.3959062	25
173	4	0.09691001	0.3876401	16
174	1	0.30103000	0.3010300	1
175	1	0.30103000	0.3010300	1
176	4	0.09691001	0.3876401	16
177	6	0.06694679	0.4016807	36
178	8	0.05115252	0.4092202	64
179	1	0.30103000	0.3010300	1
180	1	0.30103000	0.3010300	1
181	6	0.06694679	0.4016807	36
182	7	0.05799195	0.4059436	49
183	7	0.05799195	0.4059436	49
184	4	0.09691001	0.3876401	16
185	2	0.17609126	0.3521825	4
186	6	0.06694679	0.4016807	36
187	3	0.12493874	0.3748162	9
188	1	0.30103000	0.3010300	1
189	4	0.09691001	0.3876401	16
190	1	0.30103000	0.3010300	1
191	2	0.17609126	0.3521825	4
192	3	0.12493874	0.3748162	9
193	1	0.30103000	0.3010300	1
194	5	0.07918125	0.3959062	25

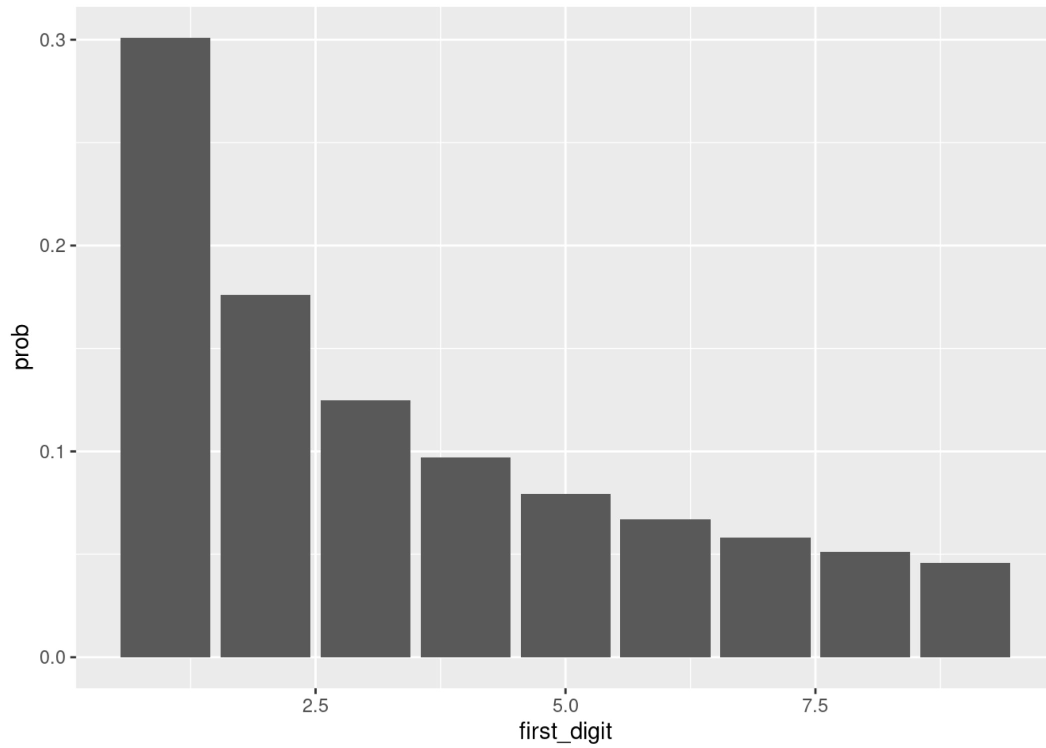
195	4	0.09691001	0.3876401	16
196	3	0.12493874	0.3748162	9
197	2	0.17609126	0.3521825	4
198	7	0.05799195	0.4059436	49
199	5	0.07918125	0.3959062	25
200	2	0.17609126	0.3521825	4
201	6	0.06694679	0.4016807	36
202	1	0.30103000	0.3010300	1
203	2	0.17609126	0.3521825	4
204	2	0.17609126	0.3521825	4
205	3	0.12493874	0.3748162	9
206	4	0.09691001	0.3876401	16
207	8	0.05115252	0.4092202	64
208	1	0.30103000	0.3010300	1
209	6	0.06694679	0.4016807	36
210	1	0.30103000	0.3010300	1
211	2	0.17609126	0.3521825	4
212	6	0.06694679	0.4016807	36
213	1	0.30103000	0.3010300	1
214	2	0.17609126	0.3521825	4
215	2	0.17609126	0.3521825	4
216	2	0.17609126	0.3521825	4
217	1	0.30103000	0.3010300	1
218	1	0.30103000	0.3010300	1
219	2	0.17609126	0.3521825	4
220	3	0.12493874	0.3748162	9
221	3	0.12493874	0.3748162	9
222	4	0.09691001	0.3876401	16
223	7	0.05799195	0.4059436	49
224	1	0.30103000	0.3010300	1
225	1	0.30103000	0.3010300	1
226	3	0.12493874	0.3748162	9
227	1	0.30103000	0.3010300	1
228	4	0.09691001	0.3876401	16
229	2	0.17609126	0.3521825	4
230	1	0.30103000	0.3010300	1
231	4	0.09691001	0.3876401	16
232	3	0.12493874	0.3748162	9
233	1	0.30103000	0.3010300	1
234	1	0.30103000	0.3010300	1
235	6	0.06694679	0.4016807	36
236	7	0.05799195	0.4059436	49
237	5	0.07918125	0.3959062	25
238	7	0.05799195	0.4059436	49
239	6	0.06694679	0.4016807	36

240	1	0.30103000	0.3010300	1
241	7	0.05799195	0.4059436	49
242	1	0.30103000	0.3010300	1
243	4	0.09691001	0.3876401	16
244	1	0.30103000	0.3010300	1
245	5	0.07918125	0.3959062	25
246	1	0.30103000	0.3010300	1
247	1	0.30103000	0.3010300	1
248	2	0.17609126	0.3521825	4
249	3	0.12493874	0.3748162	9
250	2	0.17609126	0.3521825	4
251	1	0.30103000	0.3010300	1
252	9	0.04575749	0.4118174	81
253	1	0.30103000	0.3010300	1
254	3	0.12493874	0.3748162	9
255	7	0.05799195	0.4059436	49
256	2	0.17609126	0.3521825	4
257	1	0.30103000	0.3010300	1
258	2	0.17609126	0.3521825	4
259	3	0.12493874	0.3748162	9
260	1	0.30103000	0.3010300	1
261	6	0.06694679	0.4016807	36
262	1	0.30103000	0.3010300	1
263	1	0.30103000	0.3010300	1
264	2	0.17609126	0.3521825	4
265	9	0.04575749	0.4118174	81
266	6	0.06694679	0.4016807	36
267	7	0.05799195	0.4059436	49
268	1	0.30103000	0.3010300	1
269	2	0.17609126	0.3521825	4
270	3	0.12493874	0.3748162	9
271	3	0.12493874	0.3748162	9
272	1	0.30103000	0.3010300	1
273	9	0.04575749	0.4118174	81
274	7	0.05799195	0.4059436	49
275	2	0.17609126	0.3521825	4
276	1	0.30103000	0.3010300	1
277	5	0.07918125	0.3959062	25
278	6	0.06694679	0.4016807	36
279	1	0.30103000	0.3010300	1
280	1	0.30103000	0.3010300	1
281	4	0.09691001	0.3876401	16
282	3	0.12493874	0.3748162	9
283	8	0.05115252	0.4092202	64
284	3	0.12493874	0.3748162	9

285	4	0.09691001	0.3876401	16
286	4	0.09691001	0.3876401	16
287	2	0.17609126	0.3521825	4
288	1	0.30103000	0.3010300	1
289	4	0.09691001	0.3876401	16
290	2	0.17609126	0.3521825	4
291	1	0.30103000	0.3010300	1
292	9	0.04575749	0.4118174	81
293	4	0.09691001	0.3876401	16
294	2	0.17609126	0.3521825	4
295	1	0.30103000	0.3010300	1
296	1	0.30103000	0.3010300	1
297	6	0.06694679	0.4016807	36
298	3	0.12493874	0.3748162	9
299	8	0.05115252	0.4092202	64
300	9	0.04575749	0.4118174	81
301	4	0.09691001	0.3876401	16
302	1	0.30103000	0.3010300	1
303	6	0.06694679	0.4016807	36
304	8	0.05115252	0.4092202	64
305	5	0.07918125	0.3959062	25
306	8	0.05115252	0.4092202	64
307	6	0.06694679	0.4016807	36
308	3	0.12493874	0.3748162	9
309	8	0.05115252	0.4092202	64
310	3	0.12493874	0.3748162	9
311	1	0.30103000	0.3010300	1
312	8	0.05115252	0.4092202	64
313	2	0.17609126	0.3521825	4
314	2	0.17609126	0.3521825	4
315	8	0.05115252	0.4092202	64
316	1	0.30103000	0.3010300	1
317	1	0.30103000	0.3010300	1
318	1	0.30103000	0.3010300	1
319	1	0.30103000	0.3010300	1
320	4	0.09691001	0.3876401	16
321	9	0.04575749	0.4118174	81
322	1	0.30103000	0.3010300	1
323	4	0.09691001	0.3876401	16
324	3	0.12493874	0.3748162	9
325	3	0.12493874	0.3748162	9
326	2	0.17609126	0.3521825	4
327	9	0.04575749	0.4118174	81
328	1	0.30103000	0.3010300	1
329	5	0.07918125	0.3959062	25

```
330             4 0.09691001    0.3876401    16
331             5 0.07918125    0.3959062    25
332             1 0.30103000    0.3010300     1
333             1 0.30103000    0.3010300     1
334             6 0.06694679    0.4016807    36
335             9 0.04575749    0.4118174    81
336             1 0.30103000    0.3010300     1
337             3 0.12493874    0.3748162     9
338             3 0.12493874    0.3748162     9
339             4 0.09691001    0.3876401    16
340             5 0.07918125    0.3959062    25
341             8 0.05115252    0.4092202    64
342             1 0.30103000    0.3010300     1
343             1 0.30103000    0.3010300     1
344             7 0.05799195    0.4059436    49
345             1 0.30103000    0.3010300     1
346             2 0.17609126    0.3521825     4
347             4 0.09691001    0.3876401    16
348             1 0.30103000    0.3010300     1
349             2 0.17609126    0.3521825     4
350             1 0.30103000    0.3010300     1
351             1 0.30103000    0.3010300     1
352             3 0.12493874    0.3748162     9
353             7 0.05799195    0.4059436    49
354             1 0.30103000    0.3010300     1
355             3 0.12493874    0.3748162     9
356             1 0.30103000    0.3010300     1
357             2 0.17609126    0.3521825     4
358             9 0.04575749    0.4118174    81
359             6 0.06694679    0.4016807    36
360             1 0.30103000    0.3010300     1
361             1 0.30103000    0.3010300     1
362             8 0.05115252    0.4092202    64
363             5 0.07918125    0.3959062    25
364             1 0.30103000    0.3010300     1
365             2 0.17609126    0.3521825     4
366             1 0.30103000    0.3010300     1
```

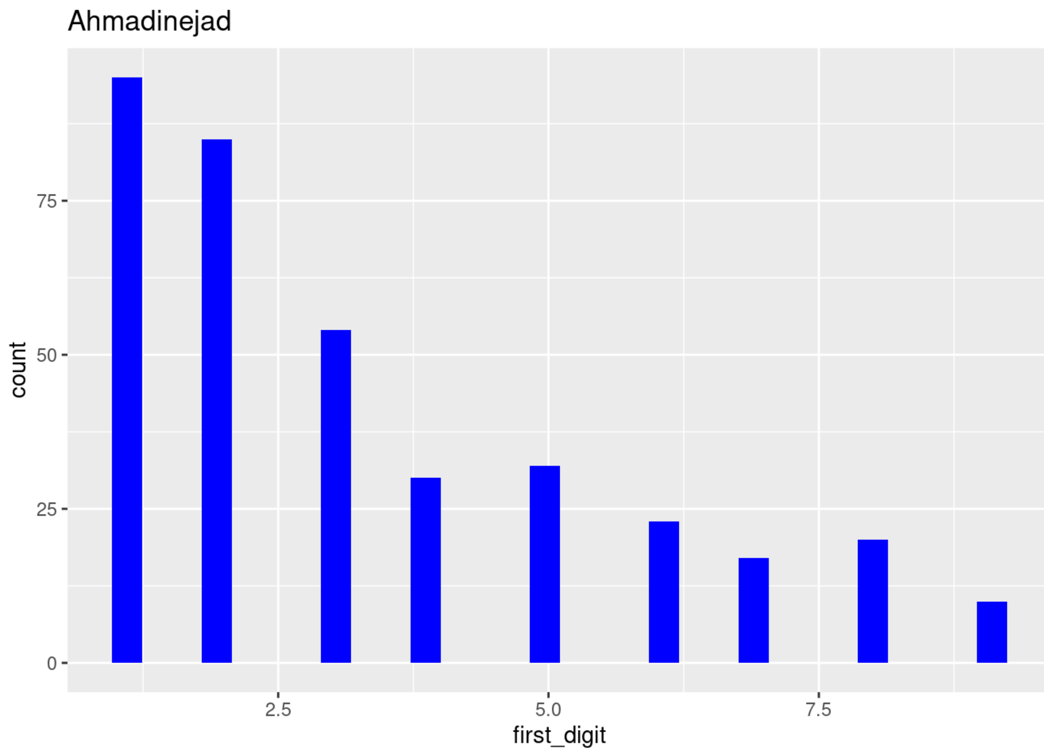
```
fd_benford %>%
  ggplot(aes (x = first_digit, y = prob)) +
  geom_col()
```



Question 9:

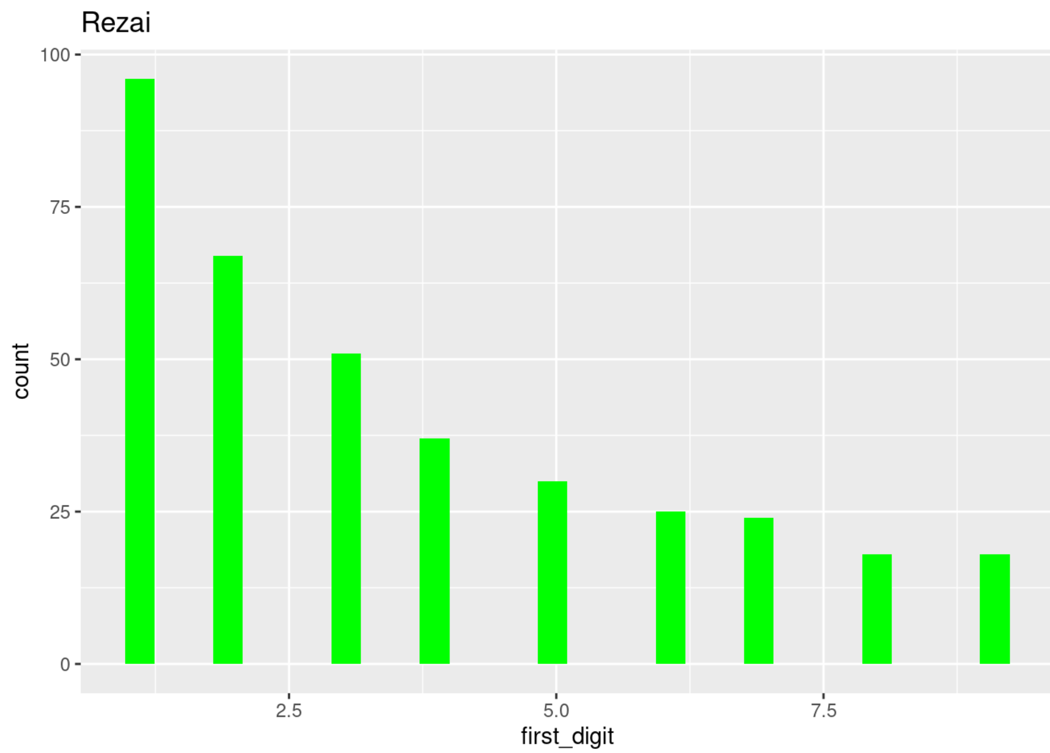
```
iran %>%
  mutate(first_digit = get_first(ahmadinejad)) %>%
  select(ahmadinejad, first_digit) %>%
  ggplot(aes(x = first_digit)) +
  geom_histogram(fill = "blue") +
  ggtitle('Ahmadinejad')
```

```
`stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```



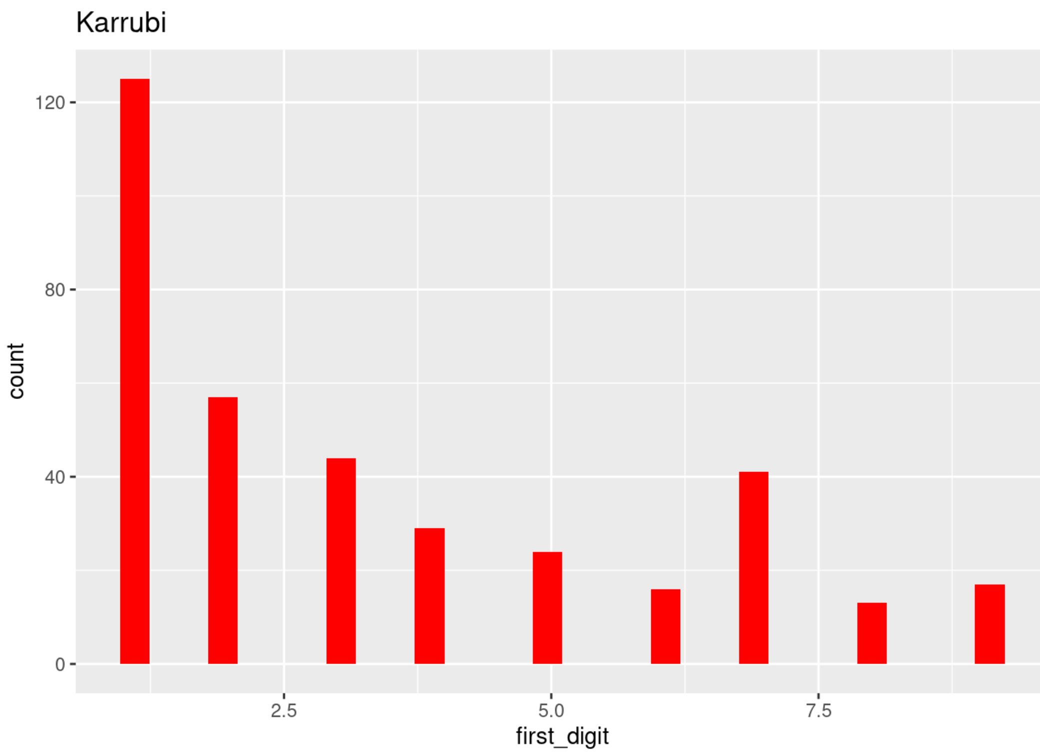

```
iran %>%
  mutate(first_digit = get_first(rezai)) %>%
  select(rezai, first_digit) %>%
  ggplot(aes(x = first_digit)) +
  geom_histogram(fill = "green") +
  ggtitle('Rezai')
```

```
`stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```



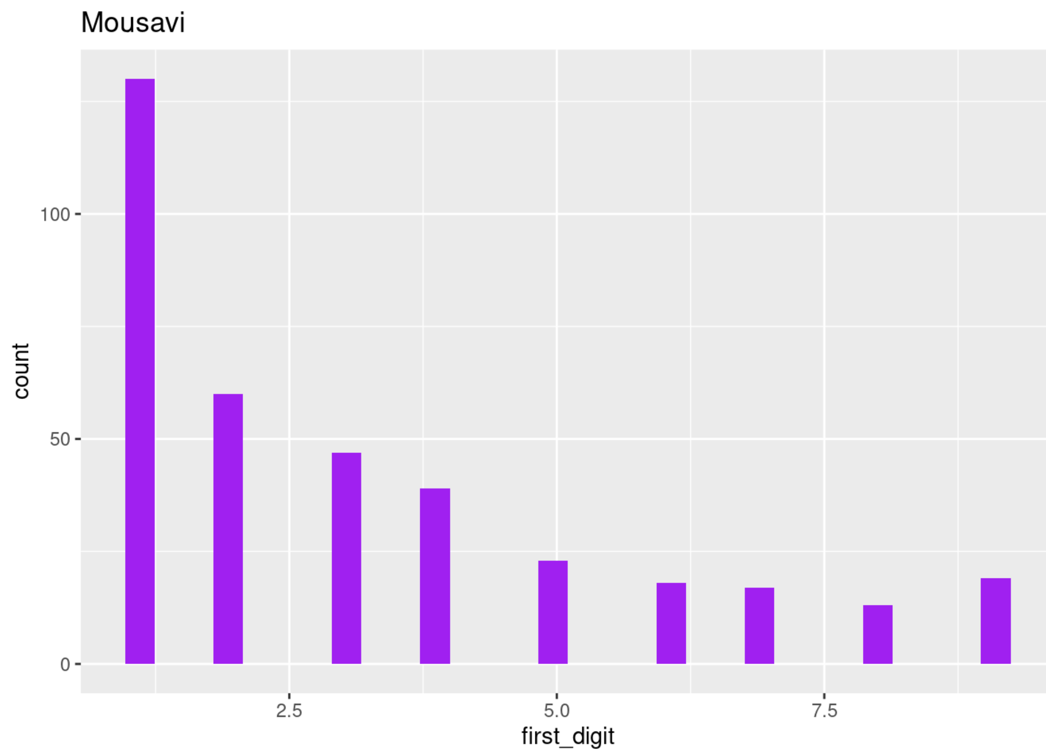

```
iran %>%
  mutate(first_digit = get_first(karrubi)) %>%
  select(karrubi, first_digit) %>%
  ggplot(aes(x = first_digit)) +
  geom_histogram(fill = "red") +
  ggtitle('Karrubi')
```

```
`stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```




```
iran %>%
  mutate(first_digit = get_first(mousavi)) %>%
  select(mousavi, first_digit) %>%
  ggplot(aes(x = first_digit)) +
  geom_histogram(fill = "purple") +
  ggtitle('Mousavi')
```

```
`stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```



plot1 + plot2 + plot3 + plot4

Question 10:

While it initially seems like Ahmadinejad has the most differences from the Benford's law, it can be seen that the largest difference actually occurs with the Karrubi plot because of the steep drop between the first two bars.

U.S. Elections

Question 11:

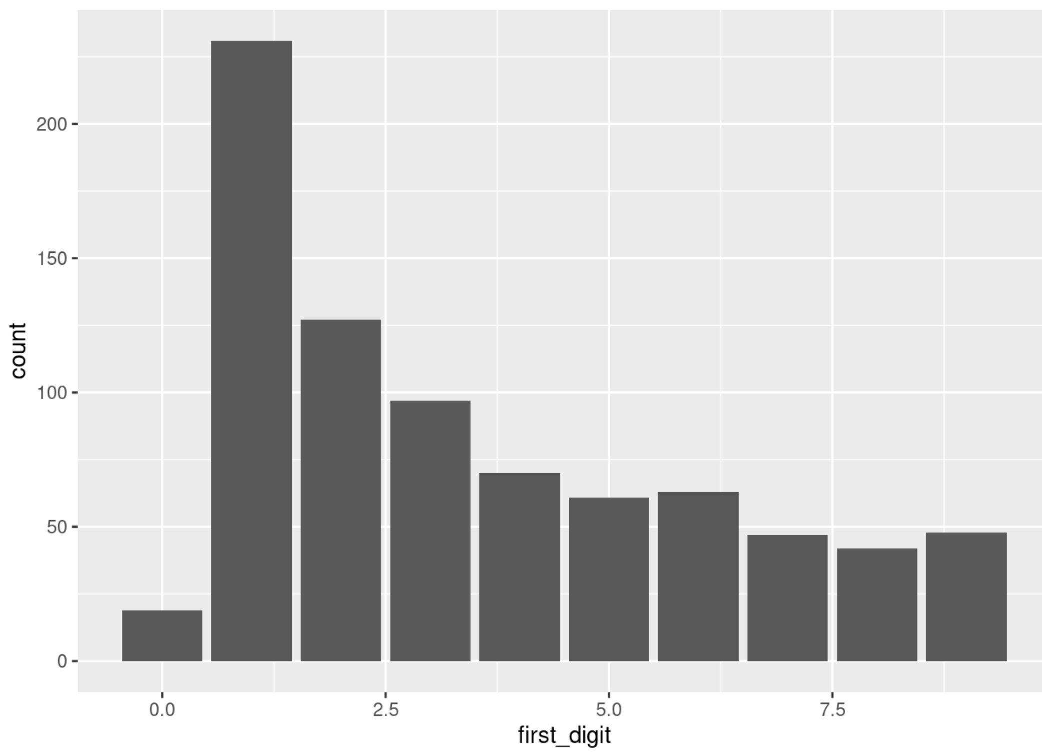
The state I chose to study was New York. The unit of observation in New York's data frame appears to be counties as that is what each row entry is differentiated by. The dimensions of this data frame are 807 x 6.

Question 12:

```
New_York_Data_set %>%
  mutate(first_digit = get_first(votes)) %>%
  select(votes, first_digit) %>%
  ggplot(aes(x = first_digit, fill = votes)) +
    geom_bar()
```

```
Warning: Removed 1 rows containing non-finite values
(`stat_count()`).

Warning: The following aesthetics were dropped during
statistical transformation: fill
ℹ This can happen when ggplot fails to infer the correct
grouping structure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a
numerical
  variable into a factor?
```



Question 13:

This data seems to fit the Benford model better than the Iran data set. This is extremely important because had there been significant variability, then there the U.S. elections could have been tampered with.