Andy Zhu - STAT196K HW2

Questions

1

(3 pts)

1. **Why is it better to take a simple random sample, instead of just the first k rows?**

> Taking a simple random sample decreases the likelihood of producing a bias sample of a population. If you only take the first k rows, you are excluding a portion of your population from your sample.

2. **Suppose we halt reservoir sampling at element m, with m < n, where n is the size of the entire stream. Can this be a sample of the entire data? Explain.**

> It would not be a sample of the **entire** data because we have not iterated through the entire data stream. However, if the data stream is not finite, would it iterate through the data stream infinitely and never produce a sample?

3. **I read on the internet that `shuf -n 100 data.txt` uses reservoir sampling. The following commands each produce 100 lines from `data.txt`. For each command, will it produce a simple random sample of the lines of the file `data.txt`? Why or why not?**

```
seq 500 > data.txt
```

The following command does not take a simple random sample because it only takes the values 1-100 from a population of 1-500.

```
head -n 100 data.txt | shuf         # 1
41
59
37
60
5
17
26
```

```
86
88
48
78
54
22
76
25
85
57
87
74
58
77
98
52
94
80
14
61
71
3
8
46
56
67
50
55
15
39
96
49
64
44
62
9
21
18
45
11
20
27
29
70
```

```
10
35
63
1
13
19
40
99
32
92
34
16
36
90
68
31
97
28
43
42
6
30
75
83
79
82
24
38
2
12
65
7
53
93
4
23
89
33
69
95
84
73
66
91
```

```
51
81
47
72
100
```

The following command does appear to take a simple random sample. The pipeline takes a simple random sample of 100 values from a population of 500 then displays the first 100 values.

```
shuf -n 100 data.txt | head -n 100  # 2
120
456
355
311
371
401
227
111
393
229
14
299
470
80
370
308
4
188
424
291
151
13
217
34
24
135
201
351
481
186
484
441
```

```
329
117
103
410
89
99
187
10
233
206
452
294
458
336
476
374
485
434
324
241
419
453
55
62
115
301
433
368
367
412
237
158
77
420
236
207
307
240
160
152
226
350
28
49
```

```
45
16
480
130
245
295
352
9
449
312
78
72
404
303
204
278
81
250
19
168
208
273
31
376
```

Like the previous command, the following command takes a simple random sample of 200 values among a population of 500 and displays the first 100 values of that sample.

```
shuf -n 200 data.txt | head -n 100  # 3
15
75
26
290
89
92
292
16
331
152
317
193
456
```

```
185
420
471
153
151
83
264
463
272
114
318
381
82
484
163
171
477
242
77
155
209
217
349
179
55
20
487
423
31
373
339
131
27
106
186
60
233
176
228
125
438
260
449
369
```

```
84
127
97
80
405
310
210
68
41
346
126
350
167
470
121
383
38
286
444
440
330
293
3
43
393
17
111
364
208
159
36
498
397
298
359
72
311
219
124
267
64
328
391
```

The following command takes a simple random sample of 100 values out of a population of 500, displays the first 100 values of the sample, and then sorts it.

```
shuf -n 100 data.txt | head -n 100  | sort  # 4
```

```
102
103
104
116
126
128
132
133
138
139
141
145
163
165
17
171
181
188
197
208
214
216
217
220
223
226
230
234
242
243
245
246
248
256
26
261
263
267
```

```
274
279
280
282
287
288
296
300
312
313
32
323
344
351
355
358
36
363
366
368
370
378
379
385
39
393
395
397
404
415
416
418
423
429
430
432
436
441
449
450
452
457
468
47
```

```
478
48
484
485
494
495
496
498
499
63
68
78
8
84
86
87
92
95
```

2

(10 pts)

Implement reservoir sampling by writing a program in Julia called `shuf.jl` that works like a simple version of `shuf`. It should accept one positional argument with the number of elements to sample, and default to 100.

Verify that it works for the following cases:

1. `seq 10 | julia shuf.jl` shuffles the integers from 1 to 10.
2. `seq 100 | julia shuf.jl 20` samples 20 random integers without replacement from 1 to 100.
3. `seq 1000 | julia shuf.jl` samples 100 random integers without replacement from 1 to 1000.
4. `seq 1e7 | julia shuf.jl` samples 100 random integers without replacement from 1 to 1000.

3 - Testing

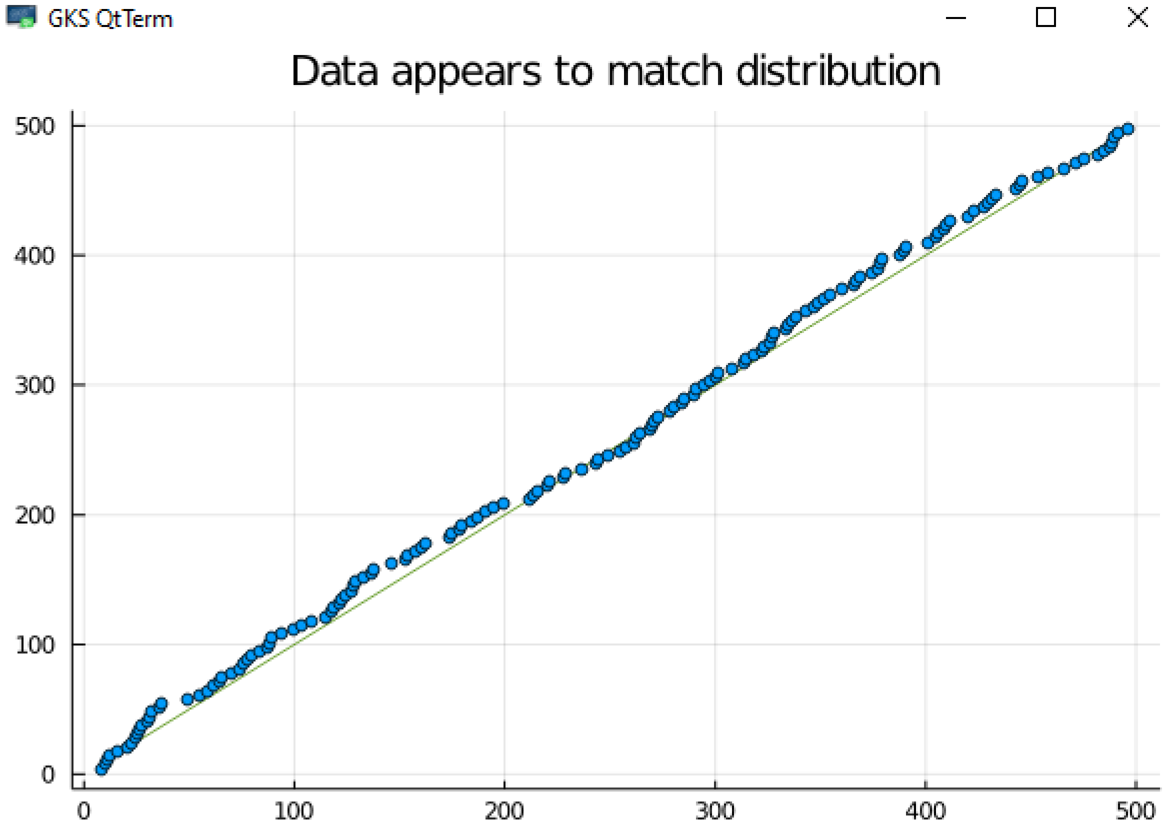
(7 pts)

Note: I will explain this step further in subsequent classes.

Use the Chi Square test or Kolmogorov Smirnov test together with seq to check if your implementation of reservoir sampling differs from the uniform distribution on the integers 1 to n. Describe how you designed the test, state the null hypothesis, show your calculations, and explain your conclusion.

```
seq 500 | julia shuf.jl 150 > output.txt
julia> sample = []
julia> open("output.txt") do f
           while !eof(f)
             x = parse(Int64, readline(f))
             push!(sample, x)
           end
         end
julia> maximum(sample)
500
julia> U500 = Uniform(0,500)
julia> qqplot(sample, U500)
julia> title!("Data appears to match distribution")
```

> Based on the qqplot, the reservoir sample appears to match the same type of distribution as a uniform distribution. Thus, the implementation reservior sample does take a simple random sample where every member of our population (n=500) had the same probabilty of being chosen for our sample (m=150).


GKS QtTerm
Data appears to match distribution
0
100
200
300
400
500
0
100
200
300
400
500