ST3009: Statistical Methods for Computer Science

MidTerm Assignment - Senán d'Art - 17329580

Q1

(a)

Please see Appendix A, Fig 1.

(b)

This was the result of summing all of the access times for X_0 and dividing by the number of accesses. Code used for this problem can be found in Appendix B, Section 2.
$Prob(X_0 = 1) = 0.2540$

(c)

The following results were obtained using the MatLab code in Appendix B, Section 3.

Chebyshev:

$$\mu - \frac{\sigma}{\sqrt{0.05N}} \leq Y \leq \mu + \frac{\sigma}{\sqrt{0.05N}}$$

$$0.1924 \leq X \leq 0.3156$$

CLT:

$$\frac{\frac{(X_1+X_2+...+X_n)}{n} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

$$\sigma = \sqrt{\mu * (1-\mu)}$$

$$0.2270 \leq X \leq 0.2810$$

Bootstrapping:
Sampling the original dataset.

$$0.2244 \leq X \leq 0.2790$$

CLT Pro - Only requires mean and variance to fully describe this distribution
CLT Cons - Only provides an approximation (not an actual bound)

Chebyshev Pro - Provides an actual bound (not an approximation)
Chebyshev Con - Results in loose bounds

Bootstrapping Pro - Can be calculated using only the original sample
Bootstrapping Con - Hard to be sure how accurate the results are

Q2

Using code in Appendix B, Section 4.
$Prob(X_0 = 1) = 0.2540$
$Prob(X_1 = 1) = 0.3460$
$Prob(X_2 = 1) = 0.2980$
$Prob(X_3 = 1) = 0.3800$
$Prob(X_4 = 1) = 0.2250$
$Prob(X_5 = 1) = 0.2820$
$Prob(X_6 = 1) = 0.3170$

Q3

Multiplying each probability by likelihood of access:
$0.0608\ 0.0067\ 0.0693\ 0.0636\ 0.0253\ 0.0550\ 0.0107$

Sum of all previous values:
0.2915

Code in Appendix B, Section 5.

Q4

Probability that the user number is 0 given that the request time is over 10ms.

$P(User_0) = 0.23926015099014$
$P(Z_n > 10) = 0.2915$
$P(Z_n > 10|User_0) = 0.2540$

$$P(User_0|Z_n > 10) = \frac{P(Z_n > 10|User_0)P(User_0)}{P(Z_n > 10)}$$

$$P(User_0|Z_n > 10) = \frac{(0.2540)(0.23926015099014)}{(0.2915)}$$

$$P(User_0|Z_n > 10) = 0.2084805432$$

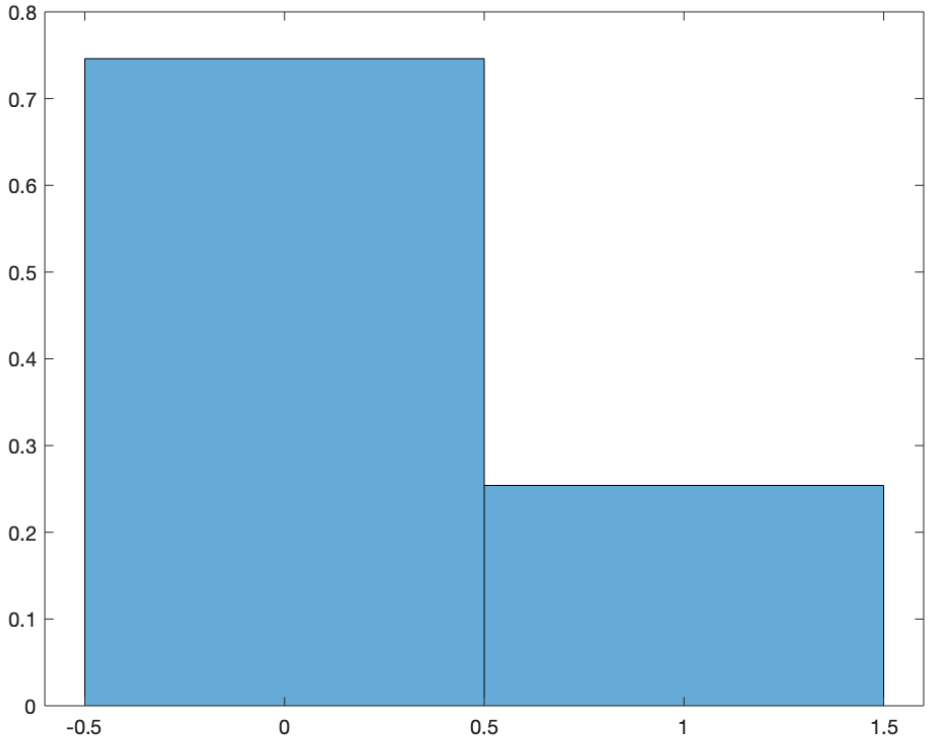
Q5

The simulation resulted in: 0.2914
This is very close to the result that was calculated.
There are some issues with using computer simulations when looking at statistics, the primary one being that computers are only pseudorandom and not actually random.
However, this type of simulation can also be extremely useful due to how easy it is to run, in the case of the example calling the API - it is easy to write a simulation to compare the real world data to expected data.

Code used can be found in Appendix B, Section 6.

Appendix A

Fig 1, graph generated by code in Appendix B, Section 1:



Appendix B

Section 1:

```
A = readmatrix('dataset_vals', 'Delimiter', ' ');

for i = 1:numel(A)
    A(i) = A(i) > 10;
end
histogram(A(:,1),'Normalization','probability');
```

Section 2:

```
A = readmatrix('dataset_vals', 'Delimiter', ' ');
A = A(:,1);
[total, x] = size(A);
lag = 0;

for i = 1:numel(A)
    x = cast(A(i), 'uint8');
    lag = lag + (x > 10);
end

mean = lag / total
```

Section 3:

```
A = readmatrix('dataset_vals', 'Delimiter', ' ');
A = A(:,1);
[total, x] = size(A);
lag = 0;

for i = 1:numel(A)
    x = cast(A(i), 'uint8');
    lag = lag + (x > 10);
end

mean = lag / total

mu = mean

sigma = sqrt(mu*(1-mu))

N = 1000

lower_chebychev = mu - (sigma / sqrt(0.05 * N))
upper_chebychev = mu + (sigma / sqrt(0.05 * N))

lower_clt = (-1.96) * (sigma / sqrt(N)) + mu
upper_clt = (1.96) * (sigma / sqrt(N)) + mu
```

```
A = readmatrix('dataset_vals', 'Delimiter', ' ');
A = A(:,1);
datasetSize = 1000;
for i = 1:numel(A)
    A(i) = A(i) > 10;
end

ci = bootci(datasetSize, {@mean, A})
```

Section 4:

```
A = readmatrix('dataset_vals', 'Delimiter', ' ');
[rows, columns] = size(A);
N = 1000;


for i = 1:columns
    X = 0;
    for j = 1:rows
        tmp = cast(A(j, i), 'uint8');
        X = X + (tmp > 10);
    end
    disp(i)
    disp(X/N)
end
```

Section 5:

```
baseProb = [0.2540 0.3460 0.2980 0.3800 0.2250 0.2820 0.3170];

A = readmatrix('dataset_probs', 'Delimiter', ' ');

A .* baseProb

sum(A .* baseProb)
```

Section 6:

```
L = [0.23926015099014 0.019489570349559 0.2325200588192 0.16747634972563
     0.11241271100911 0.19502896312529 0.033812195981071];
P = [0.2540 0.3460 0.2980 0.3800 0.2250 0.2820 0.3170];

iterations = 100000;

cumulative = 0;

for i = 1: 100000
    cumulative = cumulative + getRandomUser(L, P);
end
```

```
cumulative / iterations


function F = getRandomUser(P,X)
    p = cumsum(P);
    [~, a] = histc(rand,p);
    F = X(a + 1);
end
```