H_0 X

H_a V $P(\ ) < 0.05$,

EE 503 HW 6 Chih-Cheng Hsieh

1. $H_0: \mu = 5mm$

$H_a: \mu \neq 5mm$

$P(H_a) = P\left(\frac{\bar{X}_n - \mu}{\frac{\sigma}{\sqrt{n}}} \leq X\right) \times 2 = P\left(\frac{5.027-5}{\frac{0.1}{\sqrt{100}}} \leq X\right) \times 2 = P(2.7 \leq X) \times 2$

$= P(Z \geq 2.7) \times 2 = 0.0035 \times 2 = 0.007$

Usually if prob. < 0.05, the value is statistical significance, we reject H_0, which means we refute the engineer's conjecture.

#

2. $f(x, a) = \begin{cases} \frac{1}{a}, & 0 \leq x \leq a \\ 0, & \text{otherwise} \end{cases}$

$l(x_0, \hat{a}) = \prod_j f(x_j, a) = \left(\frac{1}{a}\right)^n$

$L(x_n, a) = \ln\left(\left(\frac{1}{a}\right)^n\right) = -n \ln(a)$

$\frac{d}{da} L(x_n, a) = \frac{d}{da}(-n \ln a) = \frac{-n}{a} = 0 \Rightarrow \hat{a} \to \infty \Rightarrow$

$\hat{a} = \max(X_1, \ldots, X_n)$

$E(\hat{a}) = E(\max(X_1, \ldots, X_n))$

$Var(\hat{a}) = Var(\max(X_1, \ldots, X_n))$

When n is huge, we have more X_i, which means we're more likely that we detect a.

$$P(X \mid Y) \cdot P(Y) = P(X\ Y)$$

$$\frac{P(XY)}{P(Y)}$$

3. Let X = input

Y = output

$$H(X|Y) = -\sum_x \sum_y P_{XY}(x,y) \log P_{X|Y}(X|y)$$

$$\Rightarrow H(X|Y=010) = -P(X=000, Y=010)\log(P(X=000 \mid Y=010))$$

$$- P(X=111, Y=010)\log(P(X=111 \mid Y=010))$$

$$= -p^2(1-p)\log\frac{p^2(1-p)}{p^2(1-p)+p(1-p)^2} - p(1-p)^2\log\frac{p(1-p)^2}{p^2(1-p)+p(1-p)^2}$$

#

4. (a)

$$H(m) = -\sum p \log p = \sum p \log \frac{1}{p}$$

$$= 2\cdot\frac{1}{4}\log 4 + 3\cdot\frac{1}{8}\log 8 + 1\cdot\frac{1}{16}\log 16 + 2\cdot\frac{1}{32}\log 32$$

$$= \frac{1}{2}\cdot 2 + \frac{3}{8}\cdot 3 + \frac{1}{16}\cdot 4 + \frac{1}{16}\cdot 5$$

$$= 1 + \underset{1.125}{\frac{9}{8}} + \underset{0.5625}{\frac{9}{16}} = 2.6875$$

(b) Let length of codeword be $\log_2 \frac{1}{p}$ of each message

5.

(a)(b)(c)

```python
import random
mean = 3
rate = 1 / mean

samples = [random.expovariate(rate) for _ in range(n)]
a = sum(samples) / n
a
```

```
2.9908993294875703
```

```python
import scipy.stats as stats
import math

C = 0.95
alpha = 1 - C
z_star = stats.norm.ppf(1 - alpha / 2)
sigma = 1

lower_limit = mean - z_star * sigma / math.sqrt(n)
upper_limit = mean + z_star * sigma / math.sqrt(n)
print("Confidence Interval:", (lower_limit, upper_limit))
if(a > lower_limit and a < upper_limit): print('This interval contains mean of the exp dist.')
else: print("This interval doesn't contains mean of the exp dist.")
```

```
Confidence Interval: (2.9722819235130062, 3.0277180764869938)
This interval contains mean of the exp dist.
```

(d)

```python
import random
import scipy.stats as stats
import math

mean = 3
rate = 1 / mean
C = 0.95
alpha = 1 - C
z_star = stats.norm.ppf(1 - alpha / 2)
sigma = 1
count = 0

for x in range(10):
    samples = [random.expovariate(rate) for _ in range(n)]
    a = sum(samples) / n

    lower_limit = mean - z_star * sigma / math.sqrt(n)
    upper_limit = mean + z_star * sigma / math.sqrt(n)
    if(a > lower_limit and a < upper_limit): count = count + 1

print(count, "true mean contains this interval.")
```

```
6 true mean contains this interval.
```