cheg304 hw6 q3

AUTHOR
author

PUBLISHED
March 31, 2025

(a)

starting with our kinda definition of t

$$t = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$$

then if we rearrange slightly

$$\bar{x} - \mu = t \frac{\sigma}{\sqrt{n}}$$

now the problem we need to solve is

$$1 = t(\nu = n - 1) \frac{9}{\sqrt{n}}$$

which could be done numerically or graphically or with mathematica or excel solver or like 30 other ways

```
from scipy.optimize import fsolve
from scipy.stats import t
import numpy as np

alpha = 0.05
sigma = 9

def func(n):
    return 1 - t.ppf(1-alpha/2, n-1) * sigma / np.sqrt(n)

n = np.round(fsolve(func, 10)[0])
print(f'n: {n:.0f}')
```

```
n: 314
```

(b)

we absoltely must assume the underlying distribution is normal to continue our analysis. from there, we can rearrange our C thingy that follows the χ^2 distribution

$$C = \frac{(n-1)S^2}{\sigma^2}$$

which becomes

$$\sigma^2 = \frac{(n-1)S^2}{C}$$

and sadly the χ^2 distribution is asymmetric so our confidence inteval will be as follows

$$\left[\frac{(n-1)S^2}{\chi^2(\alpha/2)}, \frac{(n-1)S^2}{\chi^2(1-\alpha/2)} \right]$$

where $\nu = n - 1 = 313$ (also, for scipy its backwards since it integrates from 0-x)

```
from scipy.stats import chi2

alpha = 0.05
inputs = np.array([1-alpha/2, alpha/2])
chi = chi2.ppf(inputs, n-1)

ci = (n-1) * (sigma**2) / (chi)
ci = tuple(np.round(ci, 1))
print(f'interval for variance: {ci}')
```

```
interval for variance: (69.7, 95.4)
```

```
# filler
```