Stats384 - Homework 1

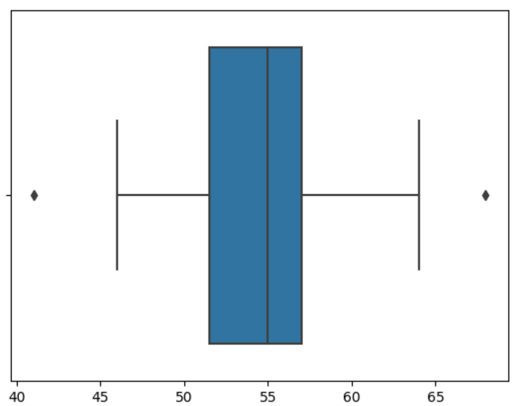
Pratyush Jha

September 27, 2023

1 7.16

1.1 Construct a boxplot of the data and comment on interesting features.

Data: [62 50 53 57 41 53 55 61 59 64 50 53 64 62 50 68 54 55 57 50 55 50 56 55 46 55 53 54 52 47 47 55 57 48 63 57 57 55 53 59 53 52 50 55 60 50 56 58]



Box and Whisker Plot

- Data has a relatively symmetric distribution with a small difference between q1 and q3 although this difference alone does not seem to be an accurate depiction of the data

1.2 Calculate and interpret a 95% CI for true average breakdown voltage µ. Does it appear that µ has been precisely estimated? Explain.

$$ConfidenceInterval = \bar{x} \pm \left(\frac{t \cdot s}{\sqrt{n}}\right)$$

Where $\bar{x} \approx 54.708333333333336$, t = .95, s$\approx$ 5.175899009404604, n = 48,
$\sqrt{48} \approx 6.92820323028 = \sqrt{n}$
95% CI = (53.244, 56.173)

1.3 Suppose the investigator believes that virtually all values of breakdown voltage are between 40 and 70. What sample size would be appropriate for the 95CI to have a width of 2 kV (so that µ is estimated to within 1 kV with 95% confidence)?

$$n = \left(\frac{Z \cdot \sigma}{E}\right)^2$$

where z = 1.96 for the given 95% confidence interval, $\sigma \approx 5.175899009404604$,
and E is given to be 1
n$\approx$102.91619722222221$\approx$103
Thus, a sample size of 103 would be needed to answer the question

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import math

#array of given data
nums = [
    62, 50, 53, 57, 41, 53, 55, 61, 59, 64,
    50, 53, 64, 62, 50, 68, 54, 55, 57, 50,
    55, 50, 56, 55, 46, 55, 53, 54, 52, 47,
    47, 55, 57, 48, 63, 57, 57, 55, 53, 59,
    53, 52, 50, 55, 60, 50, 56, 58
]

#make box and whisker plot
sns.boxplot(x = nums, orient = 'h')
plt.show()

confidenceLevel = 0.95
marginOfError = 1
zScore = 1.96
sdev=np.std(nums)
sampleSize = ((zScore*sdev)/marginOfError) ** 2

```

```
#calculate and print out desired values
mean = sum(nums)/len(nums)
print("Mean: ", mean)
print("Standard Deviation: ", sdev)
print("Number of datapoints:", len(nums))
print("Confidence Interval Lower: ", (mean-(.95*sdev))/(math
    .sqrt(48)))
print("Confidence Interval Upper: ", (mean+(.95*sdev))/(math
    .sqrt(48)))
print("Necessary sample size for 95\% CI with 2 kV width",
    sampleSize)
```

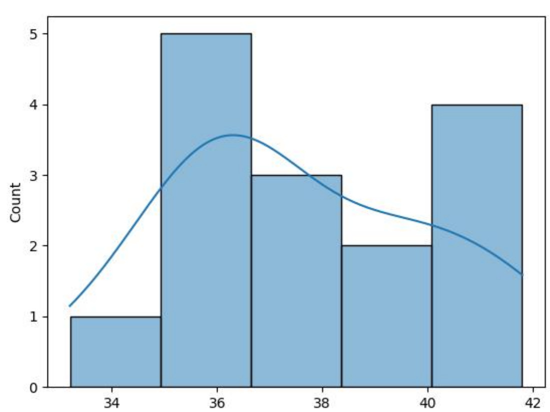
Listing 1: Computations for 7.16 in Python

```
#Compilation, Output of Listing 1 Code
Prat:Stats384 pratyush$ python3 7_16_0927.py
Mean:  54.708333333333336
Standard Deviation:  5.175899009404604
Number of datapoints: 48
Confidence Interval Lower:  53.998610495318545
Confidence Interval Upper:  55.41805617134813
Necessary sample size for 95% CI: with 2 kV width w/ 95% CI 102.91619722222221
Prat:Stats384 pratyush$
```

2 7.46

2.1 Is it plausible that this sample was selected from a normal population distribution?

Data: [33.2 41.8 37.3 40.2 36.7 39.1 36.2 41.8 36.0 35.2 36.7 38.9 35.8 35.2 40.1]



Univariate Histogram of the Data

- It is plausible that this sample was selected from a normal population distribution. The data visualized as a histogram vaguely resembles a normal distribution curve, although the test statistic following the Shapiro-Wilk test is very close to 1, at 0.9442427754402161

2.2 Calculate an upper confidence bound with confidence level 95% for the population standard deviation of maximum pressure.

$$\sqrt{\frac{(n-1)\cdot s^2}{\chi^2_{\alpha/2,n-1}}},$$

where n = 15, s $\approx$3.26, Confidence level is known to be .95, so $\alpha = .05$.
We'll use Python's SciPy library to compute χ^2 and the result.

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import chi2


data = np.array([33.2, 41.8, 37.3, 40.2, 36.7, 39.1, 36.2,
                 41.8, 36.0, 35.2, 36.7, 38.9, 35.8, 35.2, 40.1])
mean = np.mean(data)
sampleSize = len(data)
s = (data-mean)**2
root_s=np.sum(s)
sdev = np.std(data, ddof = 1)
confidenceLevel = 0.95
alpha = 1 - confidenceLevel
degreesOfFreedom = sampleSize-1
chiSquareCritical = chi2.ppf((1-alpha)/2, df=degreesOfFreedom)
ucl = np.sqrt(((degreesOfFreedom * sdev**2) / chiSquareCritical))


sns.histplot(data=data,kde=True)
plt.show()

res = stats.shapiro(data)
res.statistic
print(res)
print("Upper Confidence Bound: ", ucl)
print("n: ", sampleSize)
print("sdev: ", sdev)
```

```
print("Chi Square Crit: ", chiSquareCritical)
```

Listing 2: Computations for 7.46 in Python

```
#Compilation, Output of Listing 2 Code
Prat:Stats384 pratyush$ python3 7_46_0927.py
ShapiroResult(statistic=0.9442427754402161, pvalue=0.43871933221817017)
Upper Confidence Bound:  2.6665981083563
n:  15
sdev:  2.5715105807028404
Chi Square Crit:  13.01935693022946
Prat:Stats384 pratyush$
```

As such, the Upper Confidence Bound at 95% Confidence is calculated at roughly 2.6665981083563 standard deviations.

3 7.48

3.1 Calculate a confidence interval with confidence level 98% for the true average compressive strength under these circumstances

$$\text{Confidence Interval} = \bar{x} \pm \left(\frac{t \cdot s}{\sqrt{n}}\right),$$

Where $\bar{x} = 64.41, t = 2.224, s = 10.32, n = 18, n = 48.$

The CI is computed to be (59.00018526375335, 69.81981473624664).

3.2 Calculate a 98% lower prediction bound for the compressive strength of a single future specimen tested under the given circumstances.

Using the previously constructed CI, it can be said that the 98% lower prediction bound is ≈ 59

4 7.54

4.1 It is important that face masks used by firefighters be able to withstand high temperatures because firefighters commonly work in temperatures of 200–500°F. In a test of one type of mask, 11 of 55 masks had lenses pop out at 250°. Construct a 90% upper confidence bound for the true proportion of masks of this type whose lenses would pop out at 250°.

The formula for the upper confidence bound for a proportion is
$\hat{p} + Z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, where\ \hat{p} = 11/55, z_{\alpha/2} = 1.645, and\ n = 55.$
This computes to $\approx$ .3002. Contextualized, this means it is with 90% confidence we can say that the true proportion of popped masks is $\leq$ 30.02%.

5 7.56

5.1 Estimate the true average crack initiation depth with a 99% CI and interpret the resulting interval.

Data: [4.7, 5.1, 5.2, 5.3, 5.6, 5.8, 6.3, 6.7, 7.2, 7.4, 7.7, 8.5, 8.9, 9.3, 10.1, 11.2]
Using SciPy's statistics library, it is computed that the 99% confidence interval for the data is (5.791, 8.584).
Contextualized, it is with 99% confidence that the true average of the data lies between 5.791 and 8.584.

```python
import numpy as np
import scipy.stats as stats

data = [4.7, 5.1, 5.2, 5.3, 5.6, 5.8, 6.3, 6.7, 7.2, 7.4,
    7.7, 8.5, 8.9, 9.3, 10.1, 11.2]
mean = np.mean(data)
std_dev = np.std(data)

ci = stats.t.interval(0.99, len(data) - 1, loc=mean,
                      scale=std_dev / (len(data)**0.5))

ci_lower, ci_upper = ci

print(f"The 99% confidence interval is ({ci_lower:.3f}, {
    ci_upper:.3f})")
```

Listing 3: Computation for 7.56 Confidence Interval

```
#Compilation, output for 7.56 computation
```

```
Prat:Stats384 pratyush$ python3 7_56_0927.py
The 99% confidence interval is (5.791, 8.584)
Prat:Stats384 pratyush$
```

6 8.3.35a

6.1 Is there compelling evidence for concluding that true average repair time exceeds 200 min? Carry out a test of hypotheses using a significance level of .05.

Data: [120, 480, 149, 270, 547, 340, 43, 228, 202, 240, 218]
In addition, we are given $\bar{x} = 249.7, s = 145.1, and\ \alpha = 0.05\ and\ n = 11$.
So, $\mathrm{H}_0 : \mu \leq 200\ and\ H_a : \mu > 200$
$\mathrm{t} = (\bar{x} - \mu)/(s/\sqrt{n})$,
where t is computed to be ≈ 2.331 Using SciPy's statistics library, the critical value for $\alpha = .05\ and\ df = 10\ is \approx 1.812$.
$t = 2.331 > 1.812$, so H_0 can be rejected. Contextualized, this means that there is compelling evidence to conclude that the true average repair time is over 200 minutes when $\alpha = 0.05$.

```
import scipy.stats as st

alpha = .05
df = 10
critVal = st.t.ppf(1-alpha, df)

print(critVal)
```

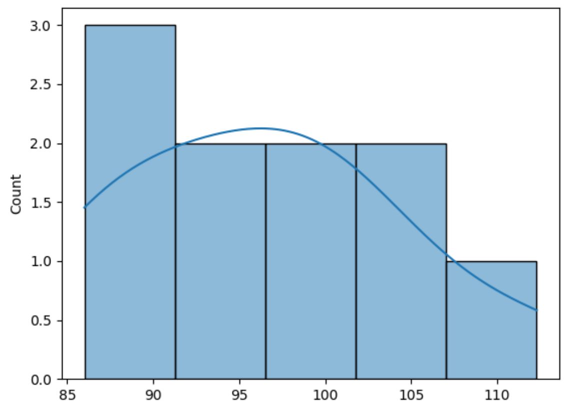
Listing 4: Critical Value 8.3.35a

```
 #Compilation, output for critical value needed for 8.3.35a
Prat:Stats384 pratyush$ python3 8_335_0927.py
1.8124611228107335
Prat:Stats384 pratyush$
```

7 8.3.37

7.1 Is it plausible that the compressive strength for this type of concrete is normally distributed?

Data: [112.3, 97.0, 92.7, 86.0, 102.0, 99.2, 95.8, 103.5, 89.0, 86.7]



Univariate Histogram of the Data

- It is plausible that this sample was selected from a normal population distribution. The data visualized as a histogram largely resembles a normal distribution curve, and the test statistic following the Shapiro-Wilk test is very close to 1, at 0.9578684568405151

7.2 Suppose the concrete will be used for a particular application unless there is strong evidence that true average strength is less than 100 MPa. Should the concrete be used? Carry out a test of appropriate hypotheses.

H_0: $\mu \geq 100\ MPa$

H_a: $\mu < 100\ MPa$

It is calculated that $\bar{x} = 96.42, n = 10, and\ s \approx 7.83477$.

α will be set to 0.05 to adhere to common standards.

t = $(\bar{x} - \mu)/(s/\sqrt{n})$, where t is computed to be $\approx$ -1.445 using the calculated values.

Using SciPy's statistics library, the critical value is computed to be ≈ 1.833.

Since $-1.445 \leq 1.833$, we fail to reject the null hypothesis.

Contexualized, this means that there is insufficient evidence to conclude that the average strength is less than 100 MPa.

```
    import seaborn as sns
import numpy as np
from scipy import stats
import math
import matplotlib.pyplot as plt


```

```
data = np.array([112.3, 97.0, 92.7, 86.0, 102.0, 99.2, 95.8,
     103.5, 89.0, 86.7])

sns.histplot(data=data, kde=True)
plt.show()

mean = np.mean(data)
sampSize = len(data)
sdev = np.std(data)
t = (mean-100)/(sdev/(math.sqrt(sampSize)))
alpha = .05
df = sampSize-1
critVal = stats.t.ppf(1-alpha, df)

res=stats.shapiro(data)
res.statistic
print(res)

print("Mean", mean)
print("n: ", sampSize)
print("S: ", sdev)
print("t: ", t)
print("Critical T Value: ", critVal)
```

Listing 5: Computations for 8.3.37

```
#Compilation, output for previously listed code
Prat:Stats384 pratyush$ python3 8_337_0927.py
ShapiroResult(statistic=0.9578684568405151, pvalue=0.7613213062286377)
Mean 96.42
n:  10
S:  7.834768662825979
t:  -1.4449634074223394
Critical T Value:  1.8331129326536335
Prat:Stats384 pratyush$
```