Ali Serdar Aydogdu
1/25/2021
ECE 3710
Homework M01

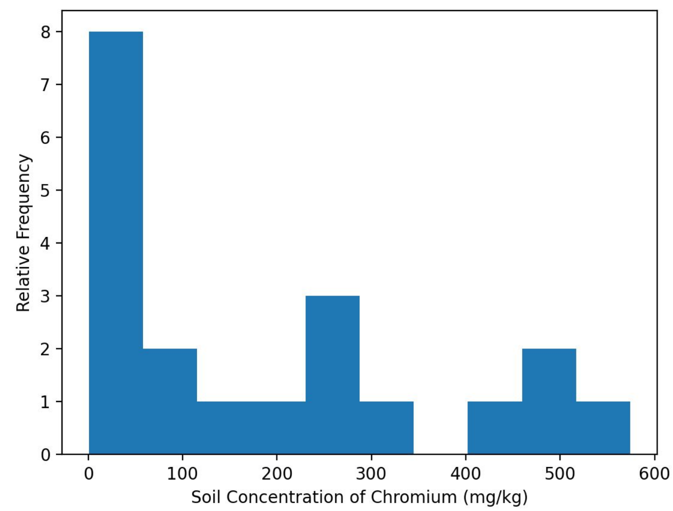
The python code is attached below. You can also access the code via GitHub here.

1- A: 1.56
B: 1.305
C: 2
D: 0

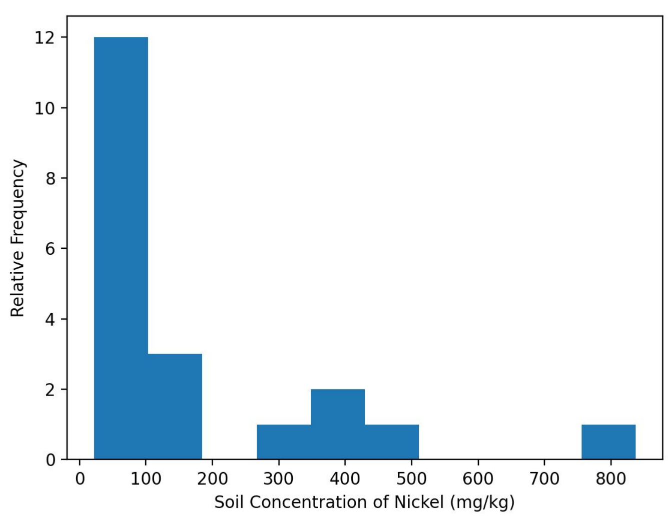
2- A: 22.74375, 20.7, 21.0125, 20.80625
B: 23.5, 20.4, 21.0, 20.7
C: 23.25, 20.7, 21.04, 20.69
D: (21.5, 25), (19.975, 22), (20.775, 21.5), (20.15, 21,2)
E: 2.8724, 1.3535, 0.4193, 0.7451
F: Method A has the largest standard deviation, because it is estimated by eye. It's the least accurate and varying measurement out of all the methods used.
G: If the other things are equal, smaller standard deviation is better, since we want a minimal variance in the measurements.

3- The mean, median and standard deviation is multiplied by 10

4- A:

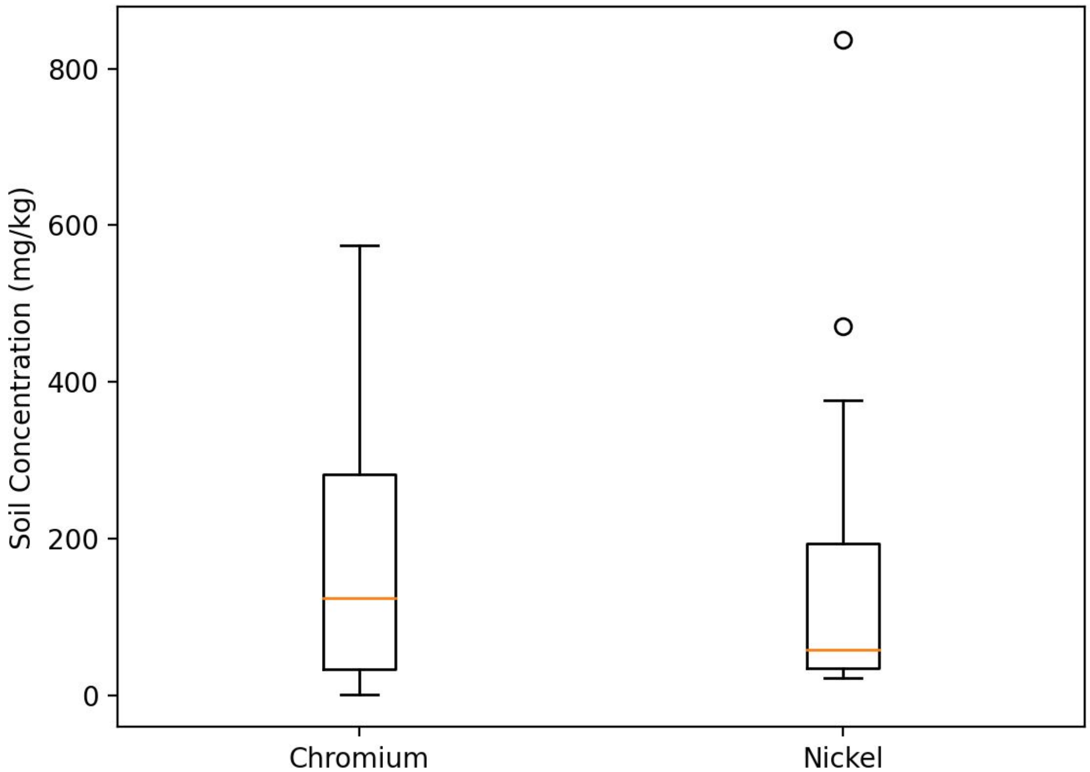


Relative Frequency
Soil Concentration of Chromium (mg/kg)



Relative Frequency
Soil Concentration of Nickel (mg/kg)

B:



C: Both Chromium and Nickel are skewed to the right. Nickel Median is very close to the first quartile. Nickel has several outliers.

5- 1 - (.05 + .1 + .15 + .25 + .2 + .1) = 0.15

```
from statistics import mean, stdev, median, quantiles
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt

# Section 1.2 Exercise 10 a-d
children = [0, 1, 2, 3, 4, 5]
num_of_women = [27, 22, 30, 12, 7, 2]

num_of_children = []

for number in children:
```

```
   num_of_children.extend([number] * num_of_women[number])

"""
num_of_children =
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 5, 5]
"""

# a. Sample mean
sample_mean = mean(num_of_children)
print("mean:", sample_mean)  # 1.56

# b. Sample standard deviation
sample_standard_deviation = stdev(num_of_children)
print("standard deviation:", sample_standard_deviation)  # 1.3051568271416059

# c. Sample median
sample_median = median(num_of_children)
print("median:", sample_median)  # 2

# d. First quartile
all_quartiles = quantiles(num_of_children, n=4)
print("quartiles:", all_quartiles)  # [0.0, 2.0, 2.0]
first_quartile = all_quartiles[0]
print("first quartile:", first_quartile)  # 0

# Section 1.2 Exercise 12 a-g
# Estimated by eye
method_a = [
   18.0,
   18.0,
   18.0,
   20.0,
   22.0,
   22.0,
   22.5,
   23.0,
   24.0,
   24.0,
```

```
    25.0,
    25.0,
    25.0,
    25.0,
    26.0,
    26.4,
]
print("Length of method_a:", len(method_a))

# Measured with a ruler
method_b = [
    18.8,
    18.9,
    18.9,
    19.6,
    20.1,
    20.4,
    20.4,
    20.4,
    20.4,
    20.5,
    21.2,
    22.0,
    22.0,
    22.0,
    22.0,
    23.6,
]

print("Length of method_b:", len(method_b))

# Measured with a ruler and string
method_c = [
    20.2,
    20.5,
    20.5,
    20.7,
    20.8,
    20.9,
    21.0,
    21.0,
    21.0,
```

```
    21.0,
    21.0,
    21.5,
    21.5,
    21.5,
    21.5,
    21.6,
]

print("Length of method_c:", len(method_c))

# Measured by rolling the ball along a ruler
method_d = [
    20.0,
    20.0,
    20.0,
    20.0,
    20.2,
    20.5,
    20.5,
    20.7,
    20.7,
    20.7,
    21.0,
    21.1,
    21.5,
    21.6,
    22.1,
    22.3,
]

print("Length of method_d:", len(method_d))

# a. Mean for each method
mean_method_a = mean(method_a)
print("Mean for Method A:", mean_method_a)  # 22.74375
mean_method_b = mean(method_b)
print("Mean for Method B:", mean_method_b)  # 20.7
mean_method_c = mean(method_c)
print("Mean for Method C:", mean_method_c)  # 21.0125
mean_method_d = mean(method_d)
print("Mean for Method D:", mean_method_d)  # 20.80625
```

```python
# b. Median for each method
median_method_a = median(method_a)
print("Median for Method A:", median_method_a)  # 23.5
median_method_b = median(method_b)
print("Median for Method B:", median_method_b)  # 20.4
median_method_c = median(method_c)
print("Median for Method C:", median_method_c)  # 21.0
median_method_d = median(method_d)
print("Median for Method D:", median_method_d)  # 20.7

# c. 20% trimmed mean for each method
trimmed_mean_percentage = 0.2
trimmed_mean_method_a = stats.trim_mean(method_a, trimmed_mean_percentage)
print("20% Trimmed Mean for Method A:", trimmed_mean_method_a)  # 23.25
trimmed_mean_method_b = stats.trim_mean(method_b, trimmed_mean_percentage)
print("20% Trimmed Mean for Method B:", trimmed_mean_method_b)  # 20.7
trimmed_mean_method_c = stats.trim_mean(method_c, trimmed_mean_percentage)
print("20% Trimmed Mean for Method C:", trimmed_mean_method_c)  # 21.04
trimmed_mean_method_d = stats.trim_mean(method_d, trimmed_mean_percentage)
print("20% Trimmed Mean for Method D:", trimmed_mean_method_d)  # 20.69

# d. First and Third Quartiles for each method.
quartiles_method_a = np.percentile(method_a, [25, 75])
print("First and Third Quartiles for Method A:", quartiles_method_a)  # 21.5, 25
quartiles_method_b = np.percentile(method_b, [25, 75])
print("First and Third Quartiles for Method B:", quartiles_method_b)  # 19.975, 22
quartiles_method_c = np.percentile(method_c, [25, 75])
print("First and Third Quartiles for Method C:", quartiles_method_c)  # 20.775, 21.5
quartiles_method_d = np.percentile(method_d, [25, 75])
print("First and Third Quartiles for Method D:", quartiles_method_d)  # 20.15, 21.2

# e. Standard Deviation for each method
standard_deviation_method_a = stdev(method_a)
print("Standard Deviation for Method A:", standard_deviation_method_a)  # 2.8724
standard_deviation_method_b = stdev(method_b)
print("Standard Deviation for Method B:", standard_deviation_method_b)  # 1.3535
standard_deviation_method_c = stdev(method_c)
print("Standard Deviation for Method C:", standard_deviation_method_c)  # 0.4193
standard_deviation_method_d = stdev(method_d)
print("Standard Deviation for Method D:", standard_deviation_method_d)  # 0.7451
```

```
# f. Method A has the largest standard deviation, because it is estimated by eye. It's
# the least accurate and varying measurement out of all the methods used.

# g. If the other things are equal, smaller standard deviation is better, since we
want
# a minimal variance in the measurements.

# If the tennis ball data for one of the methods from the previous exercise were
# converted from centimeters to millimeters, how would this affect the sample mean?
# The median? The standard deviation?
method_a_2 = [
   180,
   180,
   180,
   200,
   220,
   220,
   225,
   230,
   240,
   240,
   250,
   250,
   250,
   250,
   260,
   264,
]
print(mean_method_a, mean(method_a_2))  # 22.74375 227.4375
print(median_method_a, median(method_a_2))  # 23.5 235.0
print(standard_deviation_method_a, stdev(method_a_2))  # 2.8724 28.7239
# The mean, median and standard deviation is multiplied by 10

# Section 1.3 Exercise 4 a-c
chromium = (
   34,
   1,
   511,
   2,
   574,
   496,
   322,
```

```
    424,
    269,
    140,
    244,
    252,
    76,
    108,
    24,
    38,
    18,
    34,
    30,
    191,
)

nickel = (
    23,
    22,
    55,
    39,
    283,
    34,
    159,
    37,
    61,
    34,
    163,
    140,
    32,
    23,
    54,
    837,
    64,
    354,
    376,
    471,
)

# a. Histograms
plt.hist(chromium, bins = 10)
plt.ylabel('Relative Frequency')
plt.xlabel('Soil Concentration of Chromium (mg/kg)');
```

```
plt.show()

plt.hist(nickel, bins = 10)
plt.ylabel('Relative Frequency')
plt.xlabel('Soil Concentration of Nickel (mg/kg)');
plt.show()

# b. Boxplots
data = [chromium, nickel]

plt.boxplot(data)
plt.ylabel('Soil Concentration (mg/kg)')
plt.xticks([1, 2], ['Chromium', 'Nickel'])
plt.show()

# c. Both Chromium and Nickel are skewed to the right. Nickel Median is very close to
# the first quartile. Nickel has several outliers.

# Section 1.3 Exercise 8
x = 1 - (.05 + .1 + .15 + .25 + .2 + .1)
print (x) # 0.15
```