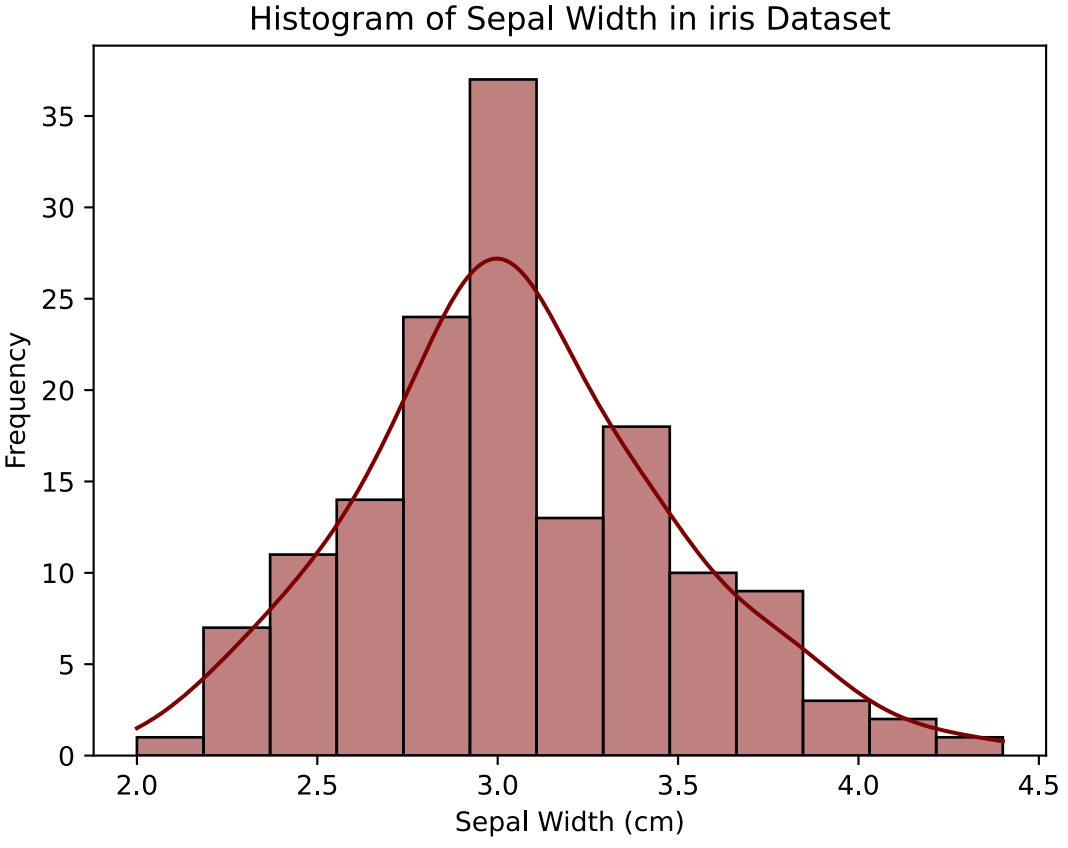
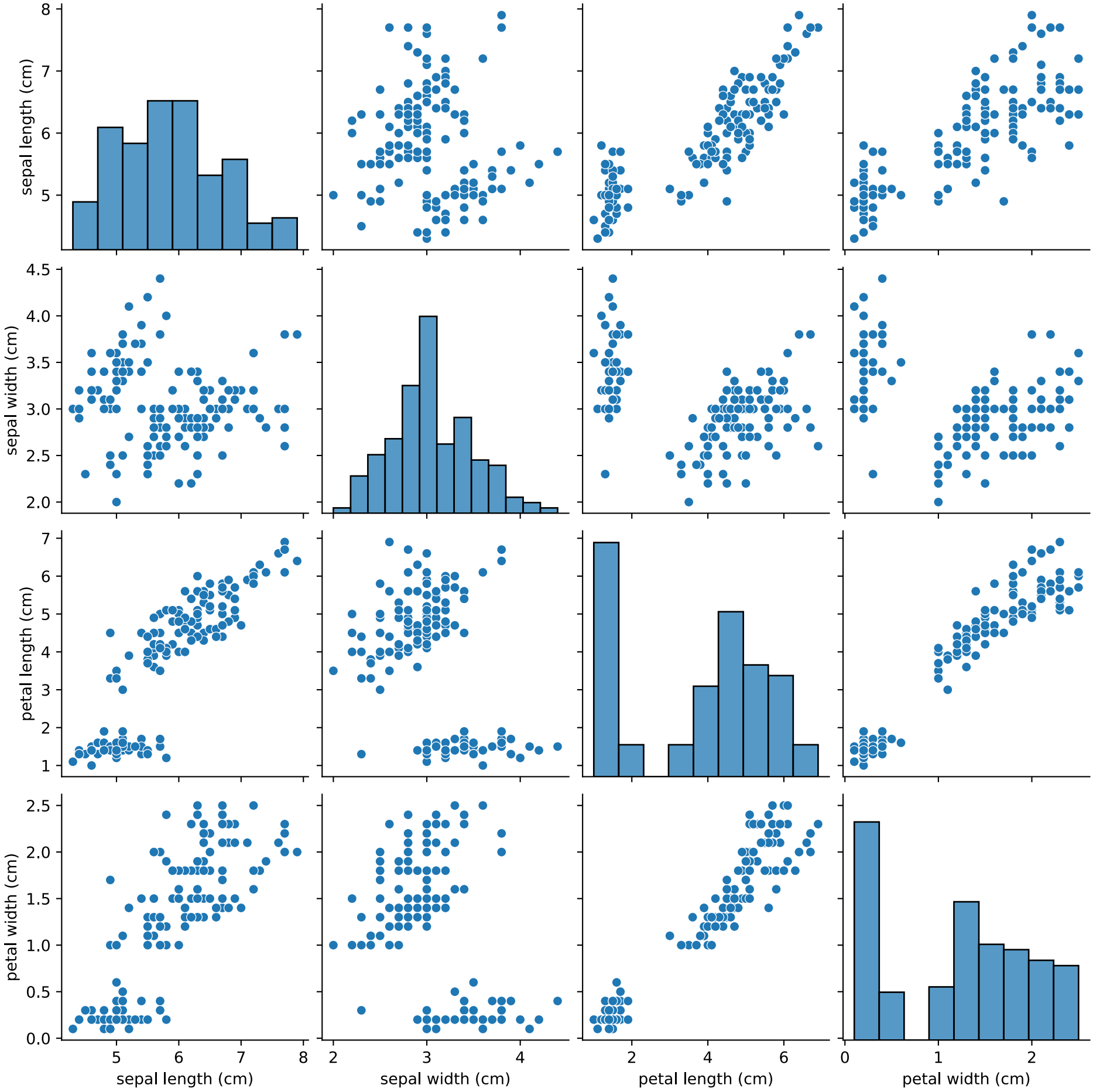
Week 3 Assignment.py

```python
# Week 3 Assignment
# Due Date: 5/25/2025
# Author: Drake Shaub

# import required packages
from sklearn.datasets import load_iris
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# import iris dataset as pandas DataFrame
iris = load_iris(as_frame = True)
df_iris = iris.frame

# create PlantGrowth dataset as pandas DataFrame
data = { "weight": [4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14, 4.81, 4.17,
                    4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69, 6.31, 5.12, 5.54, 5.50, 5.37, 5.29,
                    4.92, 6.15, 5.80, 5.26], "group": ["ctrl"] * 10 + ["trt1"] * 10 + ["trt2"] * 10}
PlantGrowth = pd.DataFrame(data)


# Question 1a
# Make a histogram of the variable Sepal.Width
sns.histplot(df_iris['sepal width (cm)'], kde=True, color='maroon')
plt.title('Histogram of Sepal Width in iris Dataset')
plt.xlabel('Sepal Width (cm)')
plt.ylabel('Frequency')

# Save the figure as .pdf
plt.savefig('/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/Summer 2025/GRAD 505 - Foundations in Data Science/Week 3/Sepal Width Histogram.pdf')

# Show the histogram
plt.show()


# Question 1b
# Based on the histogram from 1a, which would you expect to be higher, the mean or
```

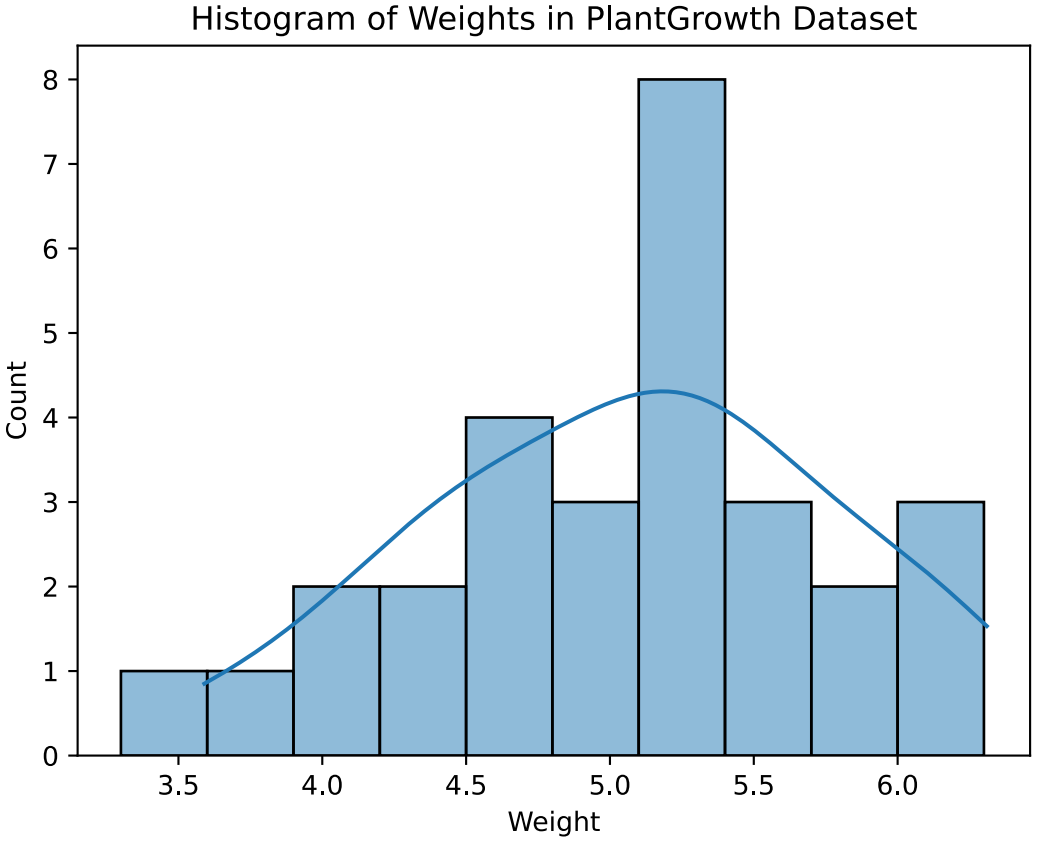

Histogram of Sepal Width in iris Dataset
Frequency
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
4.5
Sepal Width (cm)

```
the median? why?

print("See commented code for Question 1b")
# I would expect the mean to be slightly higher because it appears that the dataset is slightly right-skewed.
# Because it's right-skewed, I would expect the mean to be higher than the median because the overall sum would be greater
# due to the higher values on the right hand side of the dataset (towards the maximum).


# Question 1c
# Confirm #1b by finding the median and mean of Sepal.Width values.
mean = np.mean(df_iris['sepal width (cm)'])
median = np.median(df_iris['sepal width (cm)'])

# making the mean and median print out look cleaner
print("Question 1c:")
print(f"Mean: {mean}")
print(f"Median: {median}")


# Question 1d
# Only 27% of flowers have Sepal.Width higher than ____ cm. Fill in the blank.

# If 27% of flowers have sepal width higher than this number, this number would represent the (100-27) percentile, i.e. 73rd percentile.
# Use the np.percentile() function to calculate the 73rd percentile value, which represents the number at which 73% of the values
# fall below, but that 27% of the values fall above.
percentile = np.percentile(df_iris['sepal width (cm)'], 73)

# make percentile print out look cleaner
print("Question 1d:")
print(f"27% of the flowers have a sepal width greater than {percentile} cm")


#Question 1e
# Make scatterplots of each pair of the numerical variables in iris (there should be 6 plots)

# can use scatterplot matrix (pairplot) to perform this in one go.
iris_num_vars = ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

```

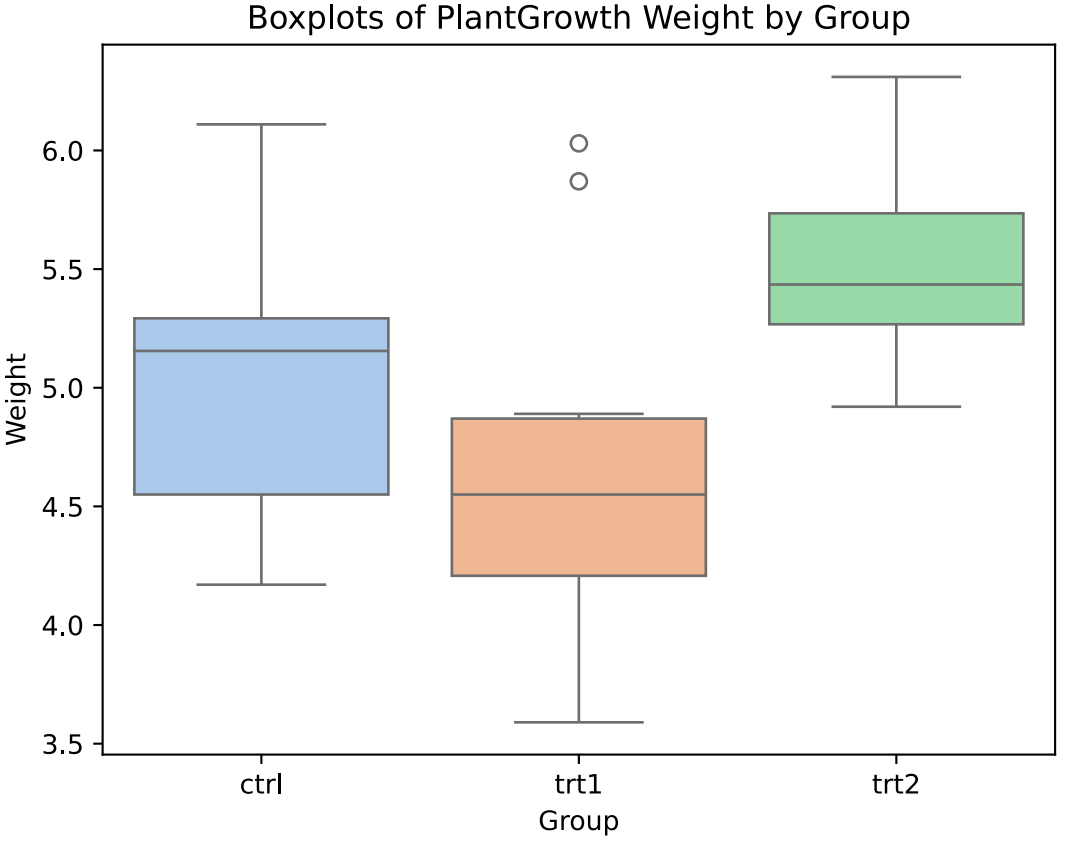

sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

```
sns.pairplot(df_iris[iris_num_vars], markers='o')

# save the figure
plt.savefig('/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/Summer 2025/GRAD 505 - Foundations in Data Science/Week 3/Pairs of Numerical Variables Scatterplots.pdf')

# show the scatterplots
plt.show()

# Additional answers for 1e
# Can also make all 6 individual scatterplots (so you don't get redundant scatterplots). That code is shown below.

# Sepal Length vs Sepal Width
# sns.scatterplot(data=df_iris, x='sepal length (cm)', y='sepal width (cm)', color='blue')
# plt.xlabel('Sepal Length (cm)')
# plt.ylabel('Sepal Width (cm)')
# plt.title('Sepal Length vs Sepal Width')
# plt.show()

# Sepal Length vs Petal Length
# sns.scatterplot(data=df_iris, x='sepal length (cm)', y='petal length (cm)', color='red')
# plt.xlabel('Sepal Length (cm)')
# plt.ylabel('Petal Length (cm)')
# plt.title('Sepal Length vs Petal Length')
# plt.show()

# Sepal Length vs Petal Width
# sns.scatterplot(data=df_iris, x='sepal length (cm)', y='petal width (cm)', color='cyan')
# plt.xlabel('Sepal Length (cm)')
# plt.ylabel('Petal Width (cm)')
# plt.title('Sepal Length vs Petal Width')
# plt.show()

# Sepal Width vs Petal Length
# sns.scatterplot(data=df_iris, x='sepal width (cm)', y='petal length (cm)', color='pink')
# plt.xlabel('Sepal Width (cm)')
# plt.ylabel('Petal Length (cm)')
# plt.title('Sepal Width vs Petal Length')
# plt.show()
```

```
114
115 # Sepal Width vs Petal Width
116 # sns.scatterplot(data=df_iris, x='sepal width (cm)', y='petal width (cm)',
    color='maroon')
117 # plt.xlabel('Sepal Width (cm)')
118 # plt.ylabel('Petal Width (cm)')
119 # plt.title('Sepal Width vs Petal Width')
120 # plt.show()
121
122 # Petal Length vs PEtal Width
123 # sns.scatterplot(data=df_iris, x='petal length (cm)', y='petal width (cm)',
    color='green')
124 # plt.xlabel('Petal Length (cm)')
125 # plt.ylabel('Petal Width (cm)')
126 # plt.title('Petal Length vs Petal Width')
127 # plt.show()
128
129
130 # Question 1f
131 # Based on #1e, which two variables appear to have the strongest relationship? And
    which two appear to have the weakest relationship?
132
133 # Petal width and petal length appear to have the strongest relationship. Sepal
    legnth and sepal width appear to have the weakest
134 # relationship.
135
136
137 #Question 2a
138 # Make a histogram of the variable weight with breakpoints (bin edges) at every
    0.3 units, starting at 3.3
139
140 min_edge = 3.3 # defined per question statement
141 max_edge = np.max(PlantGrowth['weight']) # maximum value from PlantGrowth weight
    column
142 breakpoints = 0.3 # defined per question statement
143
144 # develop an array with values starting at 3.3, going to the max value, with
    interval of 0.3
145 bin_array = np.arange(min_edge, max_edge, breakpoints)
146
147 # pass this bin_array into seaborn histplot function
148 sns.histplot(PlantGrowth['weight'], bins=bin_array, kde=True)
149
150 # give the graph a title and label axes
151 plt.title('Histogram of Weights in PlantGrowth Dataset')
```

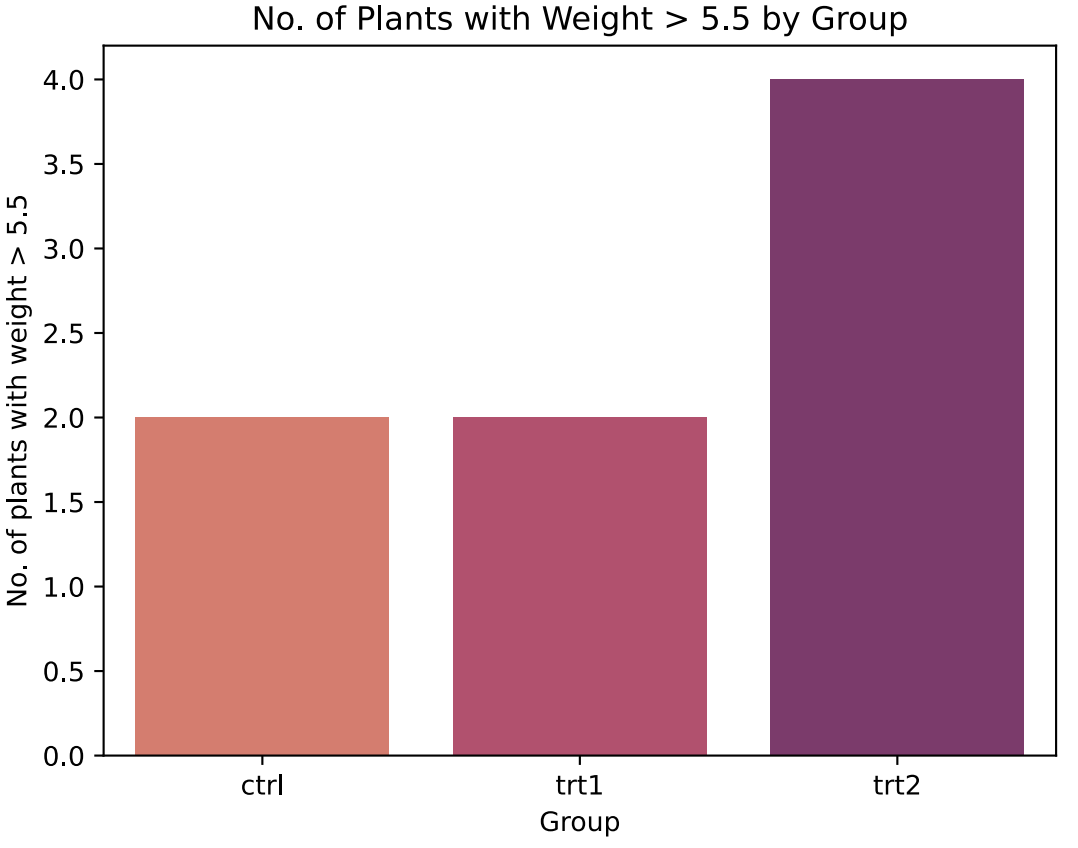

Histogram of Weights in PlantGrowth Dataset
Count
Weight
0
1
2
3
4
5
6
7
8
3.5
4.0
4.5
5.0
5.5
6.0

```
plt.xlabel('Weight')
plt.ylabel('Count')

# save figure
plt.savefig('/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/Summer 2025/GRAD 505 - Foundations in Data Science/Week 3/PlantGrowth Weight Histogram with Breakpoints at Every 0.3 Units.pdf')

# show the plot
plt.show()


# Question 2b
# Make boxplots of weight separated by group in a single graph

sns.boxplot(x='group', y='weight', data=PlantGrowth, hue='group', legend=False, palette='pastel')

# give the graph a title and label axes
plt.title('Boxplots of PlantGrowth Weight by Group')
plt.xlabel('Group')
plt.ylabel('Weight')

# save figure
plt.savefig('/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/Summer 2025/GRAD 505 - Foundations in Data Science/Week 3/Boxplots of Weight Separated by Group.pdf')

# show the plot
plt.show()

# Question 2c
# Based on the boxplots in #2b, approximately what percentage of the "trt1" weights are below the minimum "trt2" weight?

print("See commented code for Question 2c")
# Minimum "trt2" weight = ~ 4.9
# ~ 75% of the trt1 weights are below the minimum weight for trt2. The 75th percentile (Q3) is less than
# the minimum value for trt2. Therefore, at least 75% of the values fall below the minimum value for trt2.


# Question 2d
# Find the exact percentage of the "trt1" weights that are below the minimum "trt2" weight.
```


Boxplots of PlantGrowth Weight by Group
Weight
6.0
5.5
5.0
4.5
4.0
3.5
ctrl
trt1
trt2
Group

```
189
190 # filter datasets to create new dataframes grouped by column
191 ctrl_df = PlantGrowth[PlantGrowth['group'] == 'ctrl']
192 trt2_df = PlantGrowth[PlantGrowth['group'] == 'trt2']
193
194 # find minimum value from trt2_df dataframe and assign to variable
195 min_trt2_df = np.min(trt2_df['weight'])
196
197 # filter PlantGrowth to only include trt1 group and those values less than the
    minium value of trt2
198 trt1_df = PlantGrowth[(PlantGrowth['group'] == 'trt1') & (PlantGrowth['weight'] <
    min_trt2_df)]
199
200 # determine percentile by dividing the filtered count by the total count of trt1
    group in PlantGrowth
201 percentage = (trt1_df.count()[0] / PlantGrowth[PlantGrowth['group'] ==
    'trt1'].count()[0]) * 100
202
203 # print out answer
204 print("Question 2d:")
205 print(f"{percentage}% of trt1 weights are below the minimum value of trt2
    weights.")
206
207
208 # Question 2e
209 # Only including plants with a weight abvove 5.5, make a barplot of the variable
    group.
210 # Make the barplot colorful using some color palette.
211
212 # filter PlantGrowth dataset to only include those with weight > 5.5
213 barplot_df = PlantGrowth[PlantGrowth['weight'] > 5.5]
214
215 # use .value_counts() function of dataframes to pull value counts for each label
216 frequency_table = barplot_df['group'].value_counts()
217
218 # create labels (groups) and their values
219 labels = sorted(frequency_table.index)
220 values = sorted(frequency_table.values)
221
222 # create bar plot
223 sns.barplot(x=labels, y=values, hue=labels, legend=False, palette='flare')
224
225 # create title and label axes
226 plt.title('No. of Plants with Weight > 5.5 by Group')
227 plt.xlabel('Group')
```

```
plt.ylabel('No. of plants with weight > 5.5')

# save figure
plt.savefig('/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/Summer 2025/GRAD 505 - Foundations in Data Science/Week 3/Barplot of Plants with Weight Above 5.5.pdf')

# show the plot
plt.show()
```


No. of Plants with Weight > 5.5 by Group
No. of plants with weight > 5.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
ctrl
trt1
trt2
Group

extension-output-formulahendry.code-runner-#1-Code

```
[Running] python3 -u "/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/GitHub/Assignment---2/Week 3 Assignment.py"
See commented code for Question 1b
Question 1c:
Mean: 3.0573333333333337
Median: 3.0
Question 1d:
27% of the flowers have a sepal width greater than 3.3 cm
See commented code for Question 2c
/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/GitHub/Assignment---2/Week 3 Assignment.py:201: FutureWarning:
Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  percentage = (trt1_df.count()[0] / PlantGrowth[PlantGrowth['group'] == 'trt1'].count()[0]) * 100
/Users/drakeshaub/Documents/Future/Education/Purdue University 2025-2027/GitHub/Assignment---2/Week 3 Assignment.py:201: FutureWarning:
Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  percentage = (trt1_df.count()[0] / PlantGrowth[PlantGrowth['group'] == 'trt1'].count()[0]) * 100
Question 2d:
80.0% of trt1 weights are below the minimum value of trt2 weights.

[Done] exited with code=0 in 6.886 seconds


```