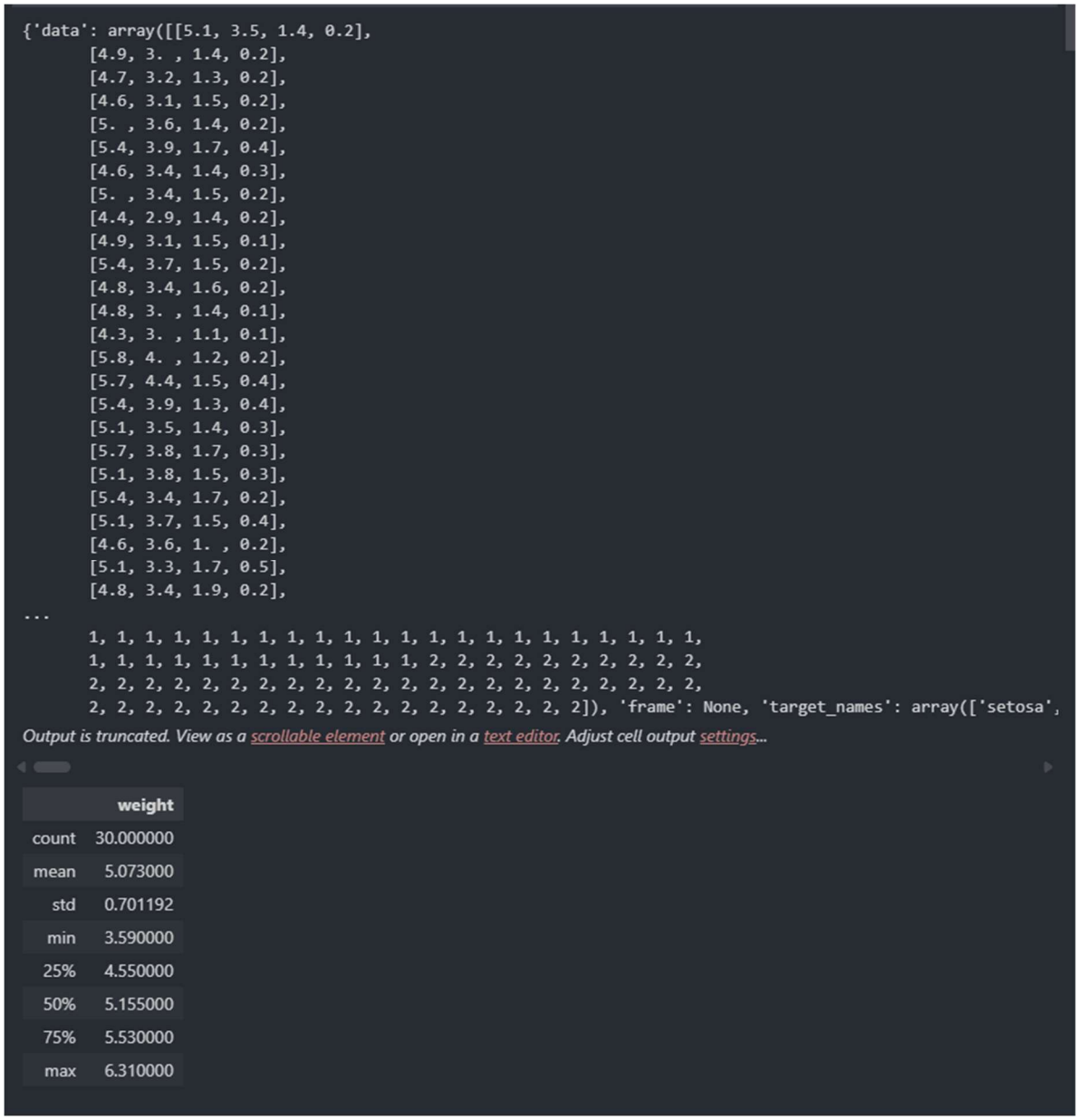
```
from sklearn import datasets

iris = datasets.load_iris()

import pandas as pd

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)

print(iris)

PlantGrowth.describe()
```



```
{'data': array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
       [5.4, 3.4, 1.7, 0.2],
       [5.1, 3.7, 1.5, 0.4],
       [4.6, 3.6, 1. , 0.2],
       [5.1, 3.3, 1.7, 0.5],
       [4.8, 3.4, 1.9, 0.2],
...
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 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, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]), 'frame': None, 'target_names': array(['setosa',
```

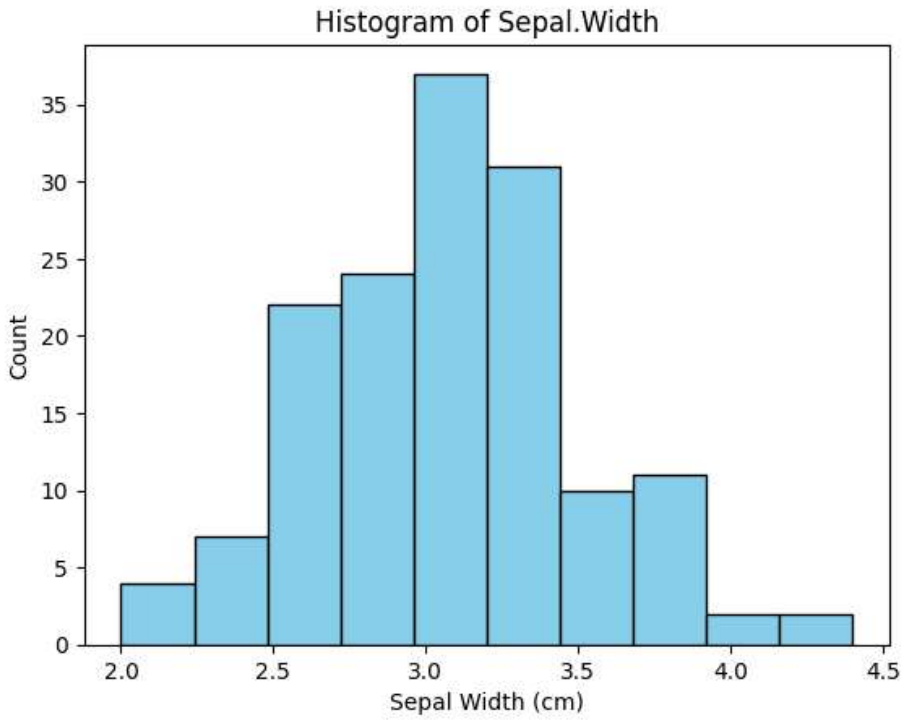
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

	weight
count	30.000000
mean	5.073000
std	0.701192
min	3.590000
25%	4.550000
50%	5.155000
75%	5.530000
max	6.310000

1. Using the iris dataset…

a. Make a histogram of the variable Sepal.Width

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

from sklearn import datasets
iris = datasets.load_iris()
dfIris = pd.DataFrame(iris.data, columns=iris.feature_names)

plt.hist(dfIris['sepal width (cm)'], color='skyblue',edgecolor='black')
plt.title('Histogram of Sepal.Width')
plt.xlabel('Sepal Width (cm)')
plt.ylabel('Count')
plt.show()
```



b. Based on the histogram from #1a, which would you expect to be higher, the mean or the median? Why?

I believe that the mean will be higher than the median. The histogram has a distribution curve that resembles a right skew (outlier is on the right side, making the graph look more flat as it approaches the right side) from what I can tell, and usually with right skews, the median is less than the mean.

c. Confirm your answer to #1b by actually finding these values.

import numpy *as* np

from sklearn *import* datasets

iris = datasets.load_iris()

dfIris = pd.DataFrame(iris.data, *columns*=iris.feature_names)

sw = dfIris['sepal width (cm)']

avg_value = np.mean(sw)

med_value = np.median(sw)

print("The Mean/Average Value is: ", avg_value)

print("The Median Value is: ", med_value)

```
import numpy as np

from sklearn import datasets
iris = datasets.load_iris()
dfIris = pd.DataFrame(iris.data, columns=iris.feature_names)

sw = dfIris['sepal width (cm)']
avg_value = np.mean(sw)
med_value = np.median(sw)

print("The Mean/Average Value is: ", avg_value)
print("The Median Value is: ", med_value)
```

```
The Mean/Average Value is:  3.0573333333333337
The Median Value is:  3.0
```

d. Only 27% of the flowers have a Sepal.Width higher than _____ cm.

import numpy *as* np

import pandas *as* pd

from sklearn *import* datasets

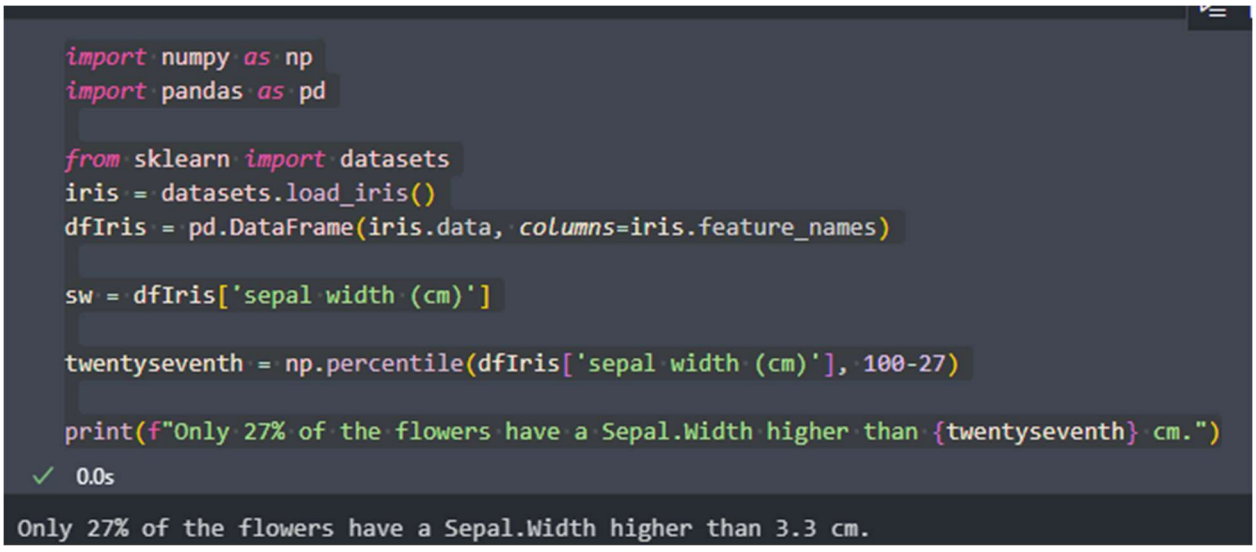
iris = datasets.load_iris()

dfIris = pd.DataFrame(iris.data, *columns*=iris.feature_names)

sw = dfIris['sepal width (cm)']

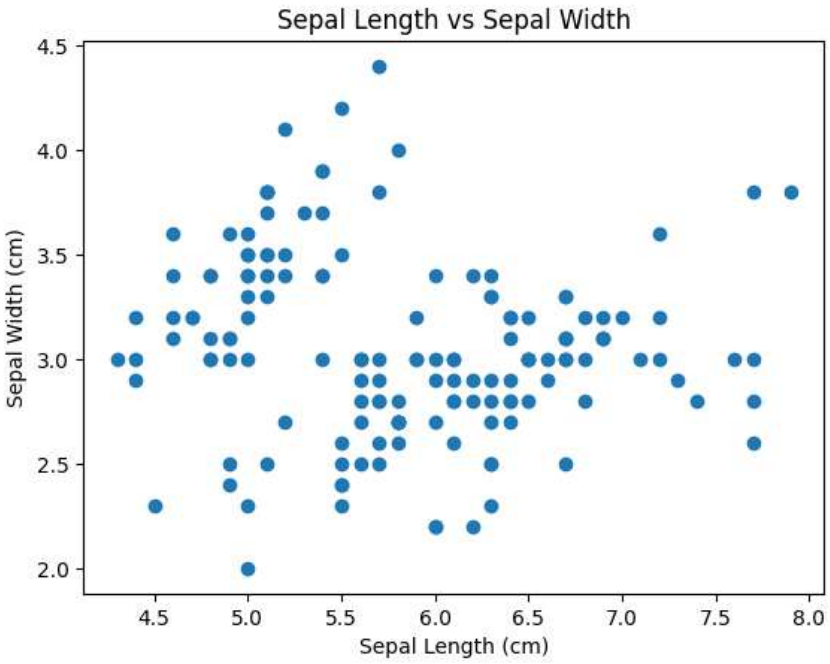
twentyseventh = np.percentile(dfIris['sepal width (cm)'], 100-27)

print(f"Only 27% of the flowers have a Sepal.Width higher than {twentyseventh} cm.")



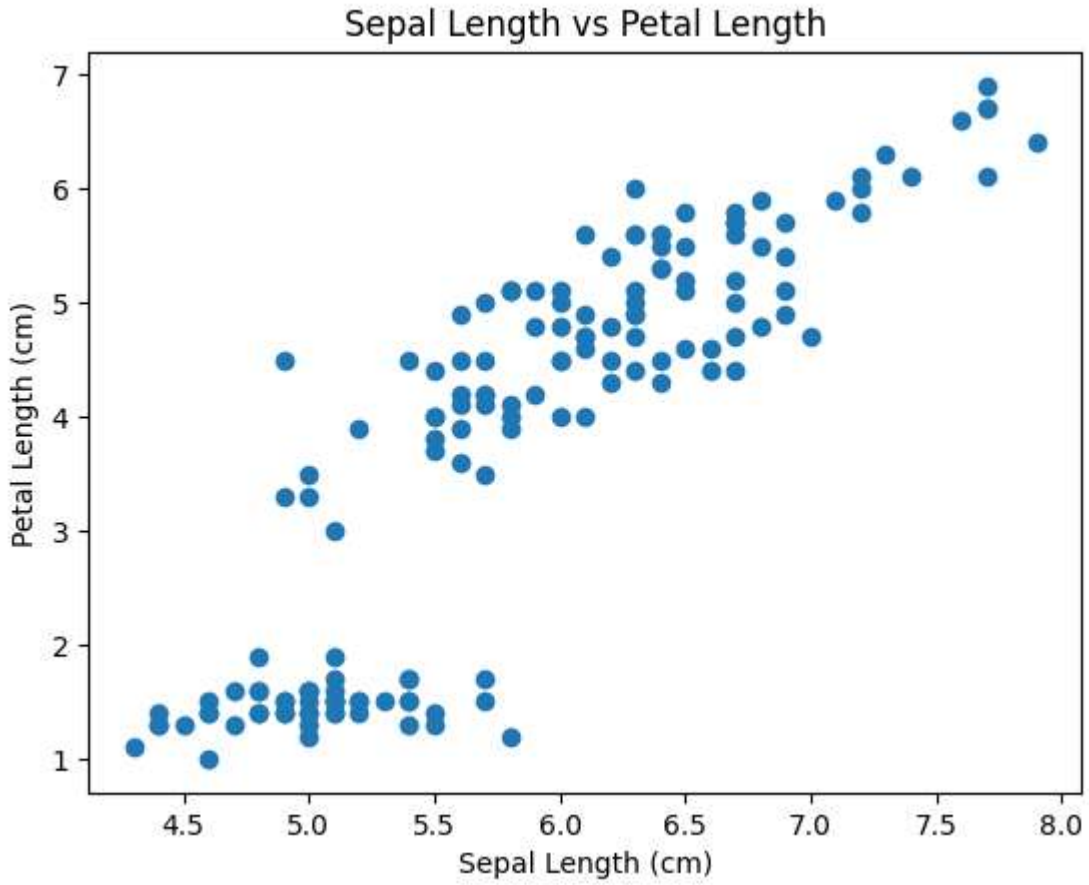
e. Make scatterplots of each pair fo the numerical variables in iris (There should be 6 pairs/plots).

```
import matplotlib.pyplot as plt

import pandas as pd

from sklearn import datasets


# Load iris

iris = datasets.load_iris()

dfIris = pd.DataFrame(iris.data, columns=iris.feature_names)


# 1. Sepal Length vs Sepal Width

plt.scatter(dfIris['sepal length (cm)'], dfIris['sepal width (cm)'])

plt.xlabel("Sepal Length (cm)")

plt.ylabel("Sepal Width (cm)")

plt.title("Sepal Length vs Sepal Width")

plt.show()
```



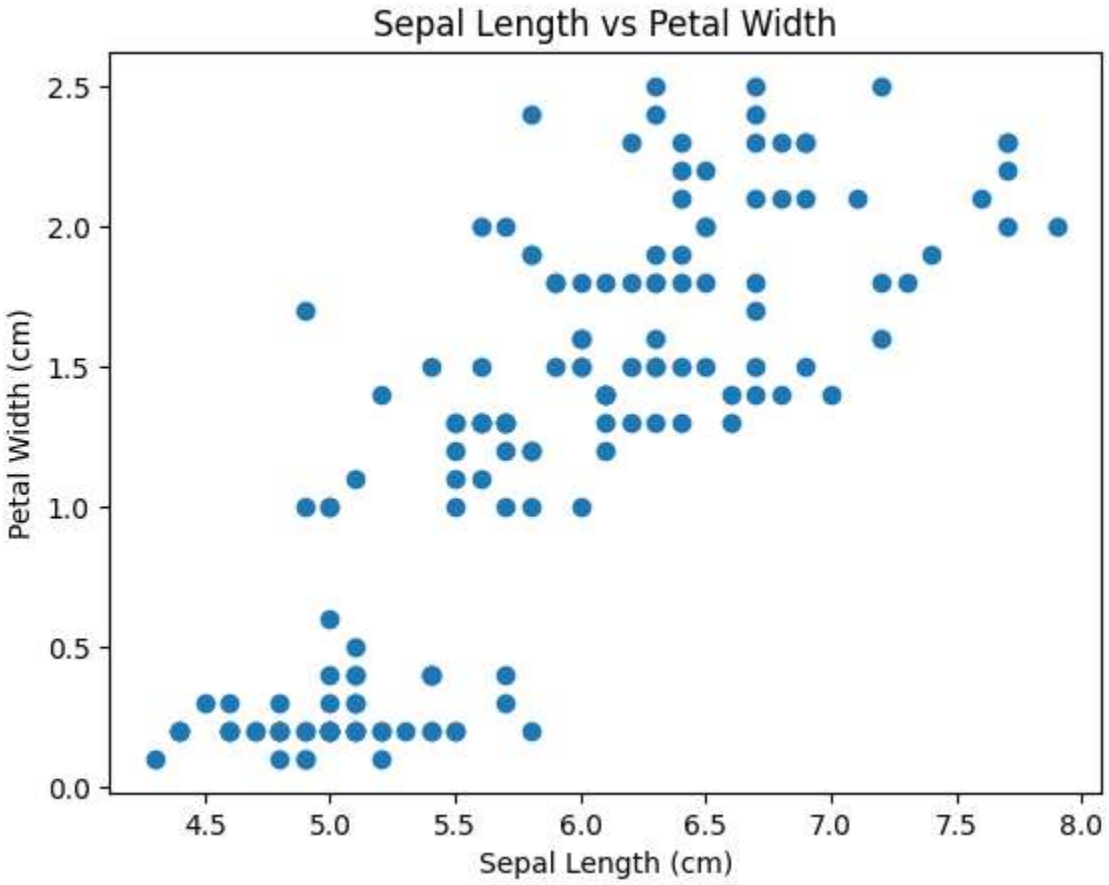
 # 2. Sepal Length vs Petal Length

```
plt.scatter(dfIris['sepal length (cm)'], dfIris['petal length (cm)'])
plt.xlabel("Sepal Length (cm)")
plt.ylabel("Petal Length (cm)")
plt.title("Sepal Length vs Petal Length")
plt.show()
```



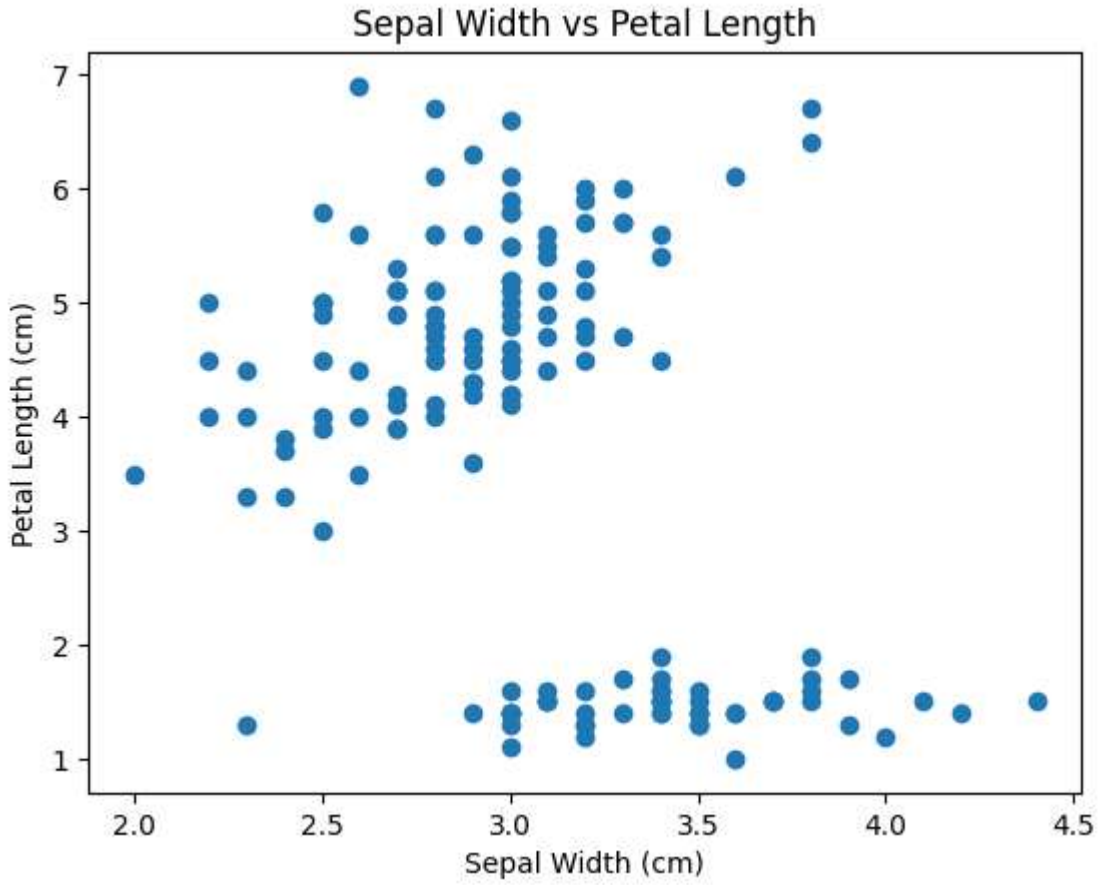
 # 3. Sepal Length vs Petal Width

```
plt.scatter(dfIris['sepal length (cm)'], dfIris['petal width (cm)'])
plt.xlabel("Sepal Length (cm)")
plt.ylabel("Petal Width (cm)")
plt.title("Sepal Length vs Petal Width")
plt.show()
```



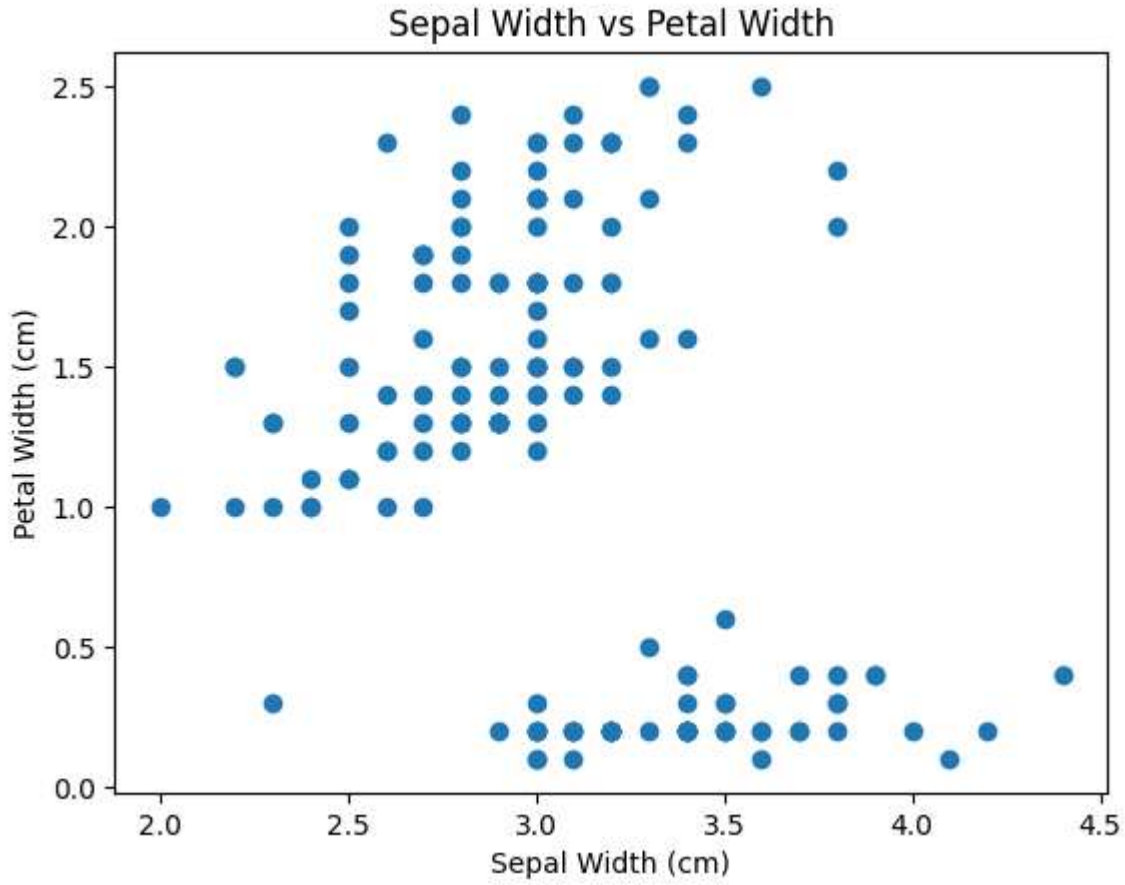
 # 4. Sepal Width vs Petal Length

```
plt.scatter(dfIris['sepal width (cm)'], dfIris['petal length (cm)'])
plt.xlabel("Sepal Width (cm)")
plt.ylabel("Petal Length (cm)")
plt.title("Sepal Width vs Petal Length")
plt.show()
```



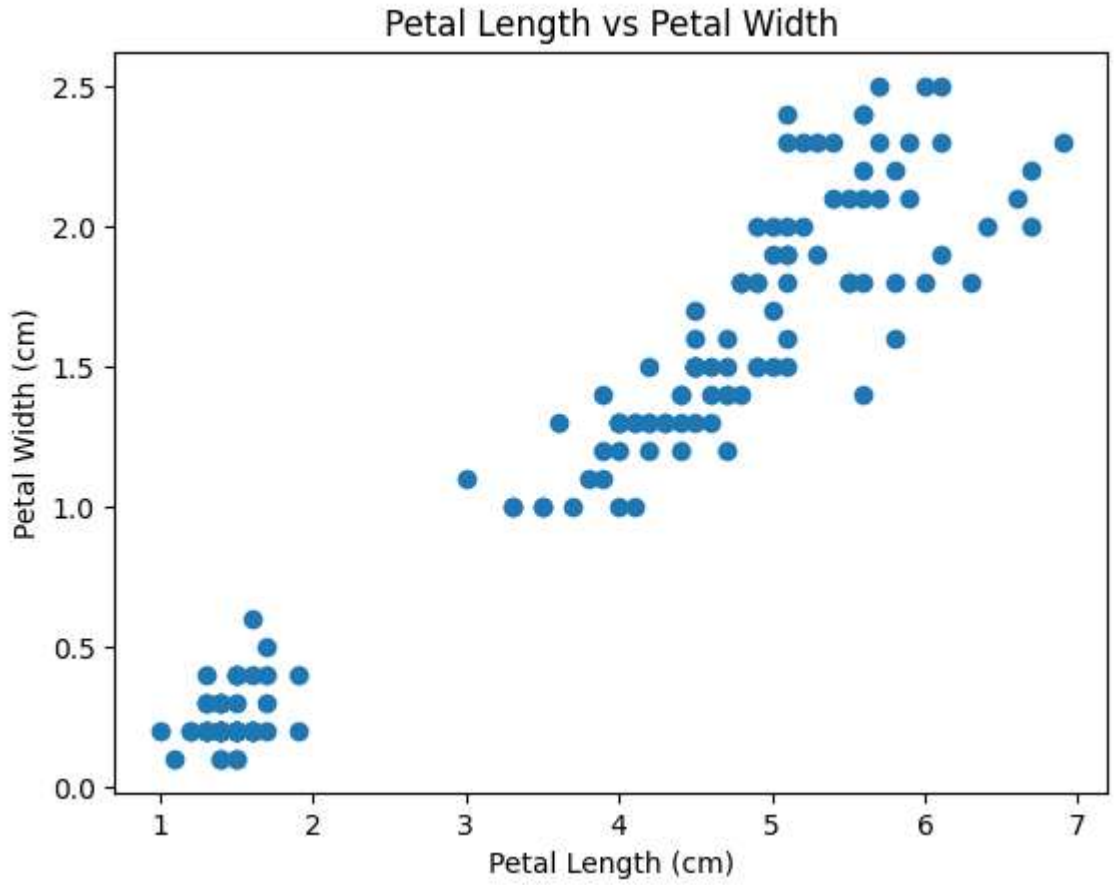
 # 5. Sepal Width vs Petal Width

```
plt.scatter(dfIris['sepal width (cm)'], dfIris['petal width (cm)'])
plt.xlabel("Sepal Width (cm)")
plt.ylabel("Petal Width (cm)")
plt.title("Sepal Width vs Petal Width")
plt.show()
```



 # 6. Petal Length vs Petal Width

```
plt.scatter(dfIris['petal length (cm)'], dfIris['petal width (cm)'])
plt.xlabel("Petal Length (cm)")
plt.ylabel("Petal Width (cm)")
plt.title("Petal Length vs Petal Width")
plt.show()
```



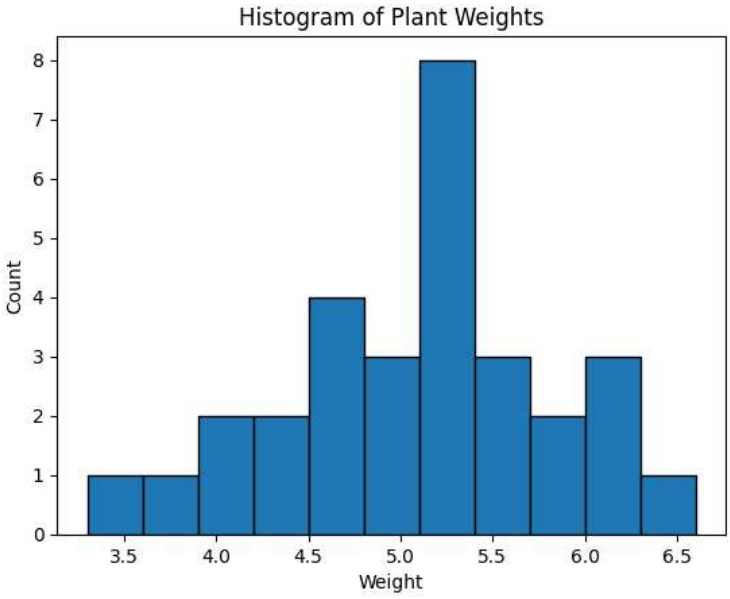
f. Based on #1e, which two variables appear to have the strongest relationship? And which two appear to have the weakest relationship?

I believe the variables that plot the strongest relationship was Petal Length vs Sepal Length. The trend line is very positive and all points seem to stay relatively close to the line of slope. The second highest contender would be Petal Width vs Petal Length, as the line there is also very distinct, however the gap between the two clusters is making me think the correlation is less strong that the one between Petal Length vs Sepal Length. The variables that plot the weakest relationship to me was Sepal Length and Sepal Width. This scatterplot had the most randomized plots it seemed and I was unable to determine a correlation line.

2. Using the PlantGrowth dataset…

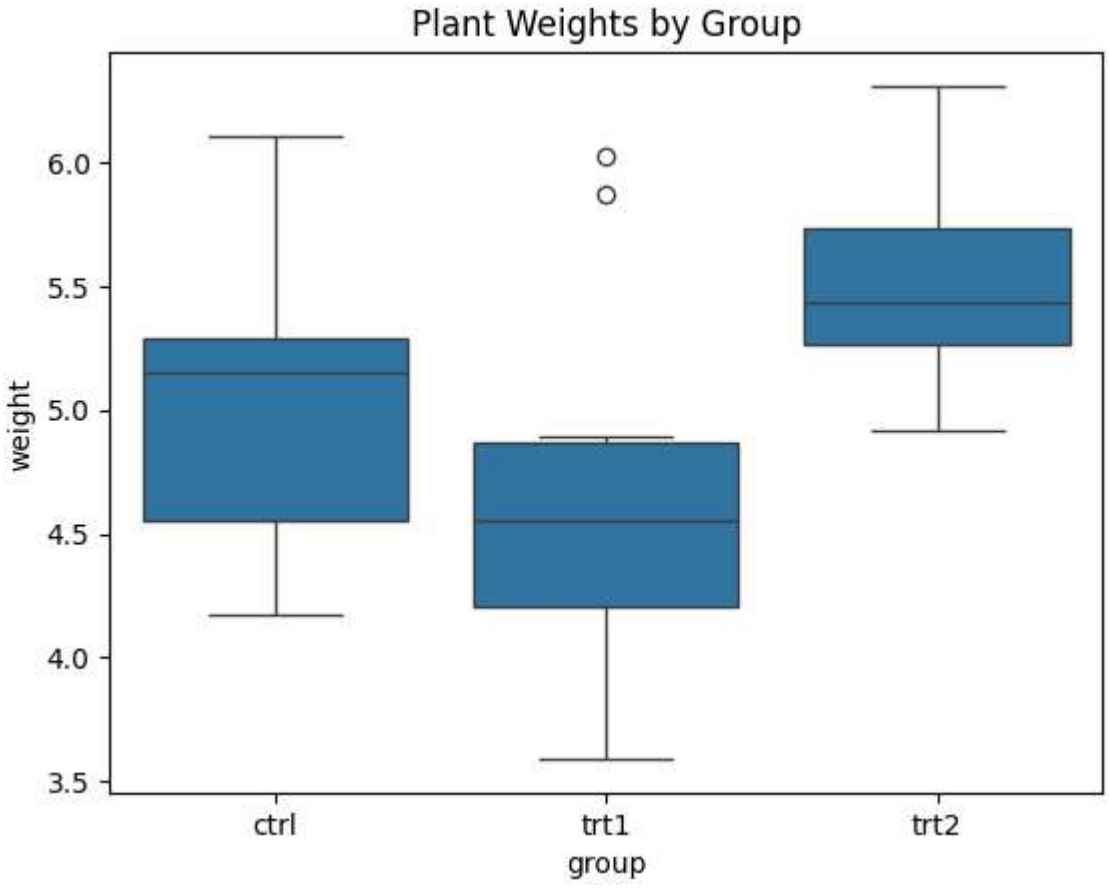
a.Make a histogram of the variable weight with breakpoints (bin edges) at every 0.3 units, starting at 3.3.

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

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)


bins = np.arange(3.3, PlantGrowth["weight"].max() + 0.3, 0.3)


plt.hist(PlantGrowth["weight"], bins=bins, edgecolor="black")

plt.title("Histogram of Plant Weights")

plt.xlabel("Weight")

plt.ylabel("Count")

plt.show()
```



b. Make boxplots of weight separated by group in a single graph.

```
import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

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)


sns.boxplot(x='group', y='weight', data=PlantGrowth)

plt.title("Plant Weights by Group")

plt.show()
```



c. Based on the boxpots in #2b, approximately what percntage of the "tft1" weights are below the minimum "trt2" weight?

From what I can tell with the boxplots, it looks like the entirety, 100% of trt1 is below the minimum trt2 weight.

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

import pandas *as* pd

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)

trt1 = PlantGrowth[PlantGrowth["group"]=="trt1"]["weight"]

trt2 = PlantGrowth[PlantGrowth["group"]=="trt2"]["weight"]

min_trt2 = trt2.min()

exact_pct = (trt1 < min_trt2).sum() / len(trt1) * 100

print("Minimum trt2 weight:", min_trt2)

print("Exact % of trt1 weights below this:", round(exact_pct, 2), "%")

```
import pandas as pd
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.
PlantGrowth = pd.DataFrame(data)

trt1 = PlantGrowth[PlantGrowth["group"]=="trt1"]["weight"]
trt2 = PlantGrowth[PlantGrowth["group"]=="trt2"]["weight"]

min_trt2 = trt2.min()
exact_pct = (trt1 < min_trt2).sum() / len(trt1) * 100

print("Minimum trt2 weight:", min_trt2)
print("Exact % of trt1 weights below this:", round(exact_pct, 2), "%")
```

✓ 0.0s Python

```
Minimum trt2 weight: 4.92
Exact % of trt1 weights below this: 80.0 %
```

e. Only including plants with a weight above 5.5, make a barplot of the variable group. Make the barplot colorful using some color palette

```
import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

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)


filtering = PlantGrowth[PlantGrowth['weight']>5.5]


sns.barplot(x='group', data=filtering, palette='pink')

plt.title('Variable Group over 5.5')

plt.show()
```