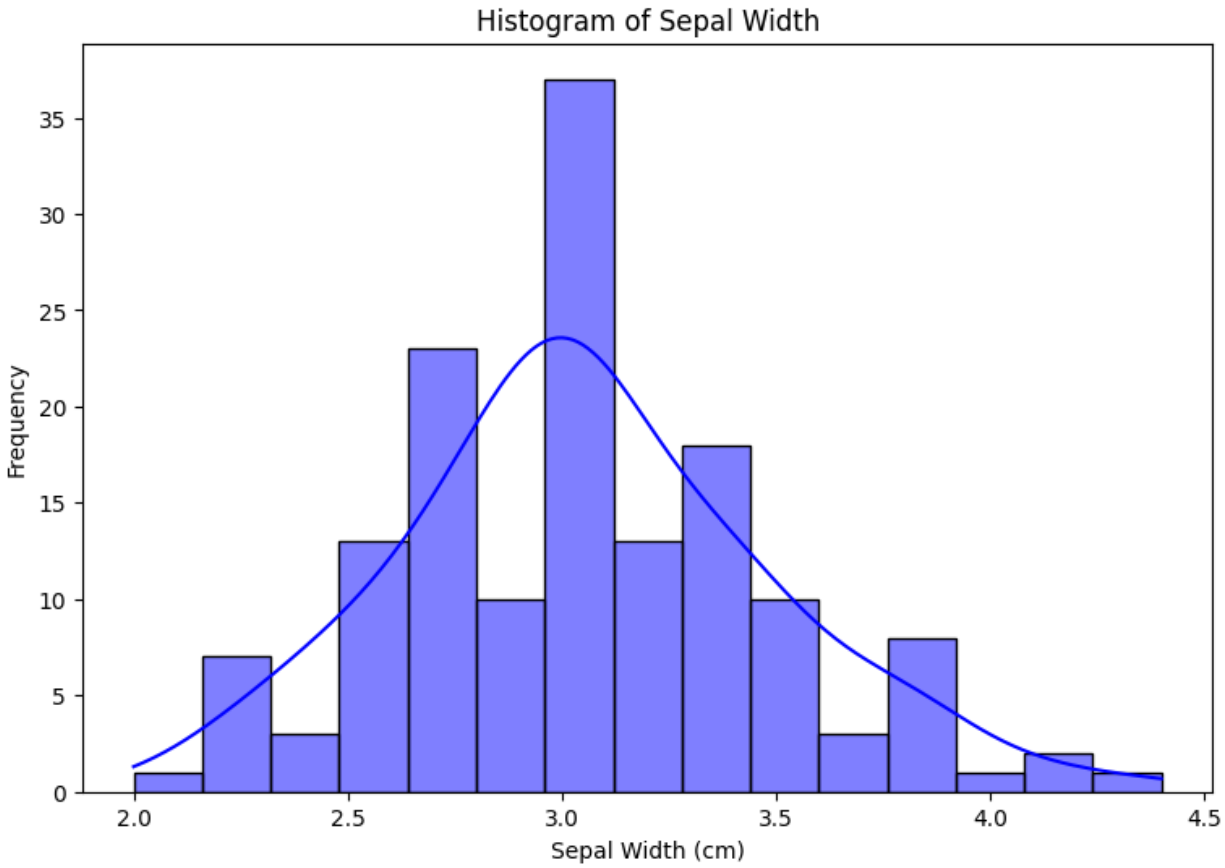
Week 3 Assignment

. Using the iris dataset…

a. **Make a histogram of the variable Sepal.Width.**

```
import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import datasets

import pandas as pd

iris = datasets.load_iris()

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

iris_df.columns = ["Sepal.Length", "Sepal.Width", "Petal.Length",
"Petal.Width"]

plt.figure(figsize=(9, 6))

sns.histplot(iris_df["Sepal.Width"], bins=15, kde=True, color="blue")

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

plt.ylabel("Frequency")

plt.title("Histogram of Sepal Width")

plt.show()
```



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

The histogram of Sepal Width looks slightly right-skewed, as there are some larger values pulling the data to the right. In this case, the mean is higher than the median because the mean is affected by larger values, while the median is just the middle number. So, I expect the mean to be greater than the median.

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

```
iris = datasets.load_iris()

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

iris_df.columns = ["Sepal.Length", "Sepal.Width", "Petal.Length",
"Petal.Width"]

mean_sepal_width = iris_df["Sepal.Width"].mean()

median_sepal_width = iris_df["Sepal.Width"].median()

print(f"Mean Sepal Width: {mean_sepal_width:.2f} cm")

print(f"Median Sepal Width: {median_sepal_width:.2f} cm")
```

Output:

Mean Sepal Width: 3.06 cm

Median Sepal Width: 3.00 cm

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

Only 27% of the flowers have a Sepal Width higher than 3.30 cm. Below is the code.

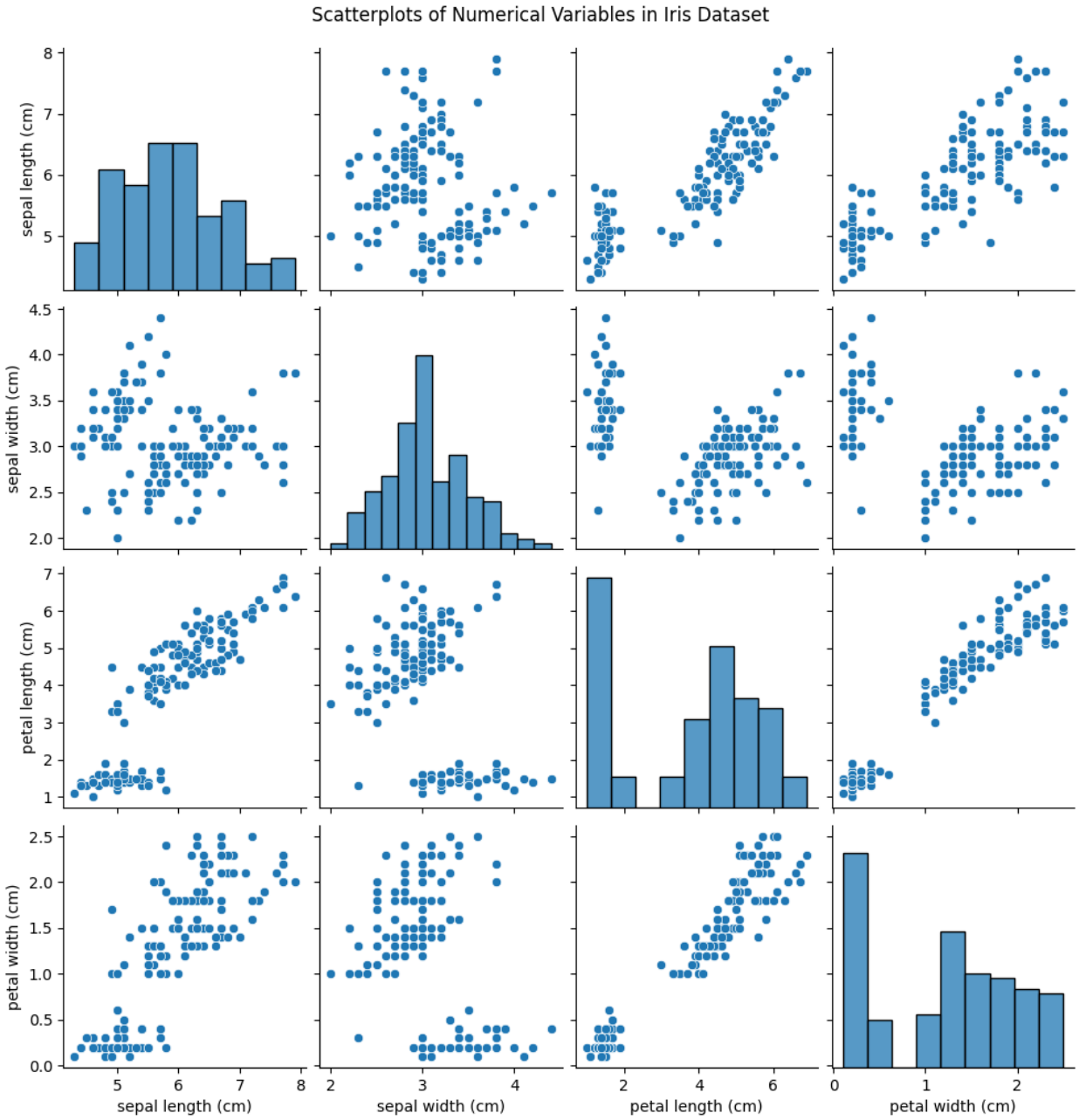
```
import numpy as np

percentile_73 = np.percentile(iris_df["Sepal.Width"], 73)

print(f"Only 27% of the flowers have a Sepal.Width higher than
{percentile_73:.2f} cm.")
```

Output:

Only 27% of the flowers have a Sepal.Width higher than 3.30 cm.

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

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import datasets
import pandas as pd
iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
sns.pairplot(iris_df, kind="scatter")
plt.suptitle("Scatterplots of Numerical Variables in Iris Dataset", y=1.02)
plt.show()
```


Scatterplots of Numerical Variables in Iris Dataset
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

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

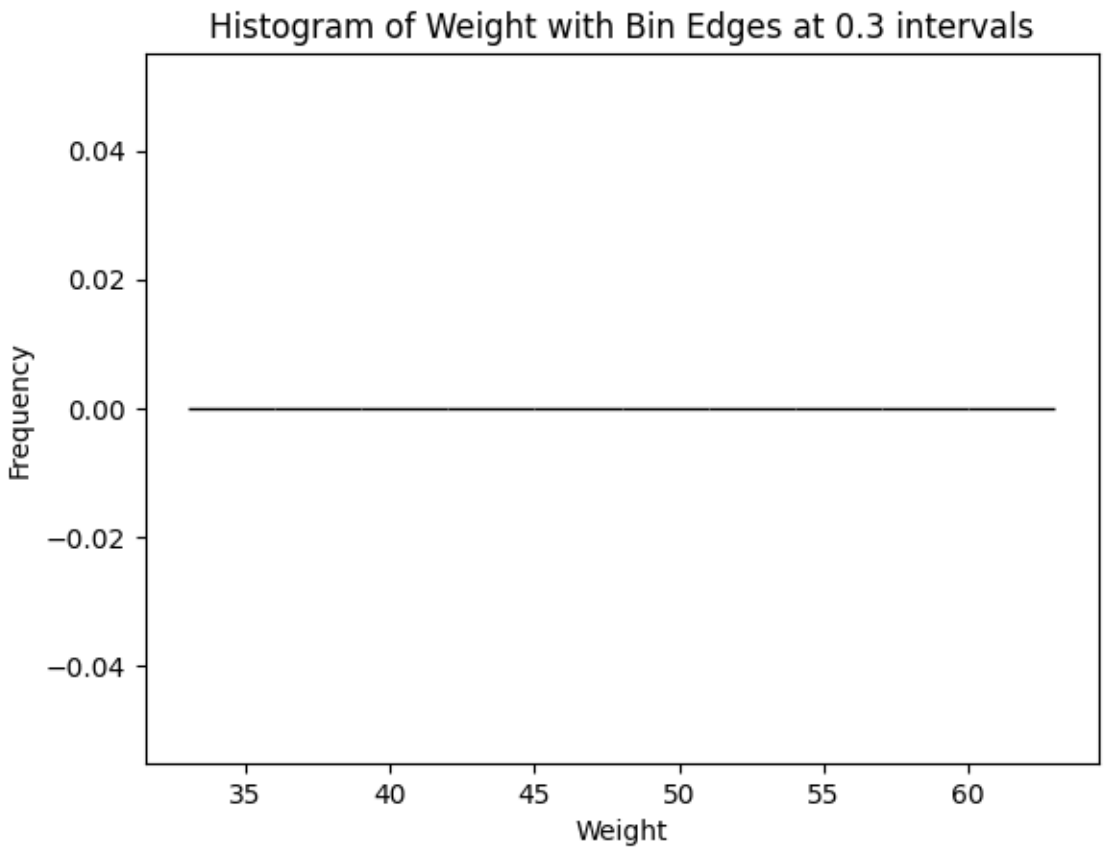
Strongest relationship: Petal Length and Petal Width. These two variables have a clear pattern, where as one increases, the other tends to increase as well.

Weakest relationship: Sepal Length and Sepal Width. These two don't seem to follow any clear pattern and look more scattered.

2. Using the PlantGrowth dataset...

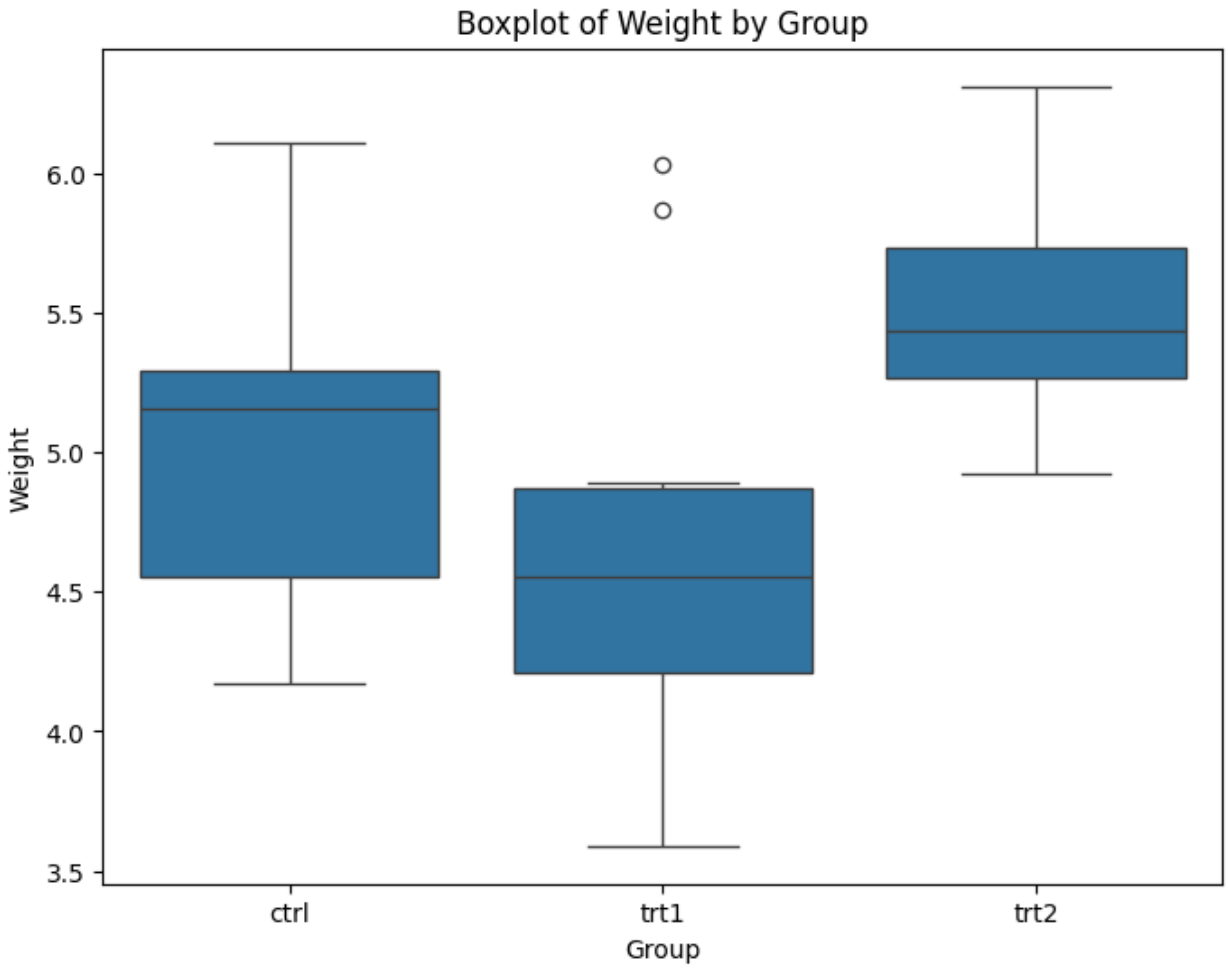
a. Make a histogram of the variable weight with breakpoints (bin edges) at every 0.3 units, starting at 3.3.

```
import 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)

bin_edges = list(range(33, 65, 3))  # 3.3 to 6.3 with step of 0.3 (multiplied
by 10 for convenience)

plt.hist(PlantGrowth['weight'], bins=bin_edges, edgecolor='black')
plt.xlabel('Weight')
plt.ylabel('Frequency')
plt.title('Histogram of Weight with Bin Edges at 0.3 intervals')
plt.show()
```


Histogram of Weight with Bin Edges at 0.3 intervals
Frequency
0.04
0.02
0.00
−0.02
−0.04
35
40
45
50
55
60
Weight

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

```
import matplotlib.pyplot as plt
import seaborn as sns
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)

# Create the boxplot
plt.figure(figsize=(8,6))
sns.boxplot(x='group', y='weight', data=PlantGrowth)
plt.title('Boxplot of Weight by Group')
plt.xlabel('Group')
plt.ylabel('Weight')
plt.show()
```



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

Based on the boxplot of the "trt1" group, it seems that a few of the "trt1" weights are below 3.6. Specifically, it appears that about 2 out of 10 of the "trt1" weights fall below this minimum threshold. So, approximately 20% of the "trt1" weights are below the minimum "trt2" weight based on the boxplots.

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

```
min_trt2_weight = PlantGrowth[PlantGrowth['group'] == 'trt2']['weight'].min()

weights_below_min_trt2 = (PlantGrowth[PlantGrowth['group'] == 'trt1']['weight'] < min_trt2_weight).sum()

total_trt1_weights = len(PlantGrowth[PlantGrowth['group'] == 'trt1'])

percentage_below_min_trt2 = (weights_below_min_trt2 / total_trt1_weights) * 100

print(f"Percentage of 'trt1' weights below the minimum 'trt2' weight: {percentage_below_min_trt2:.2f}%")
```

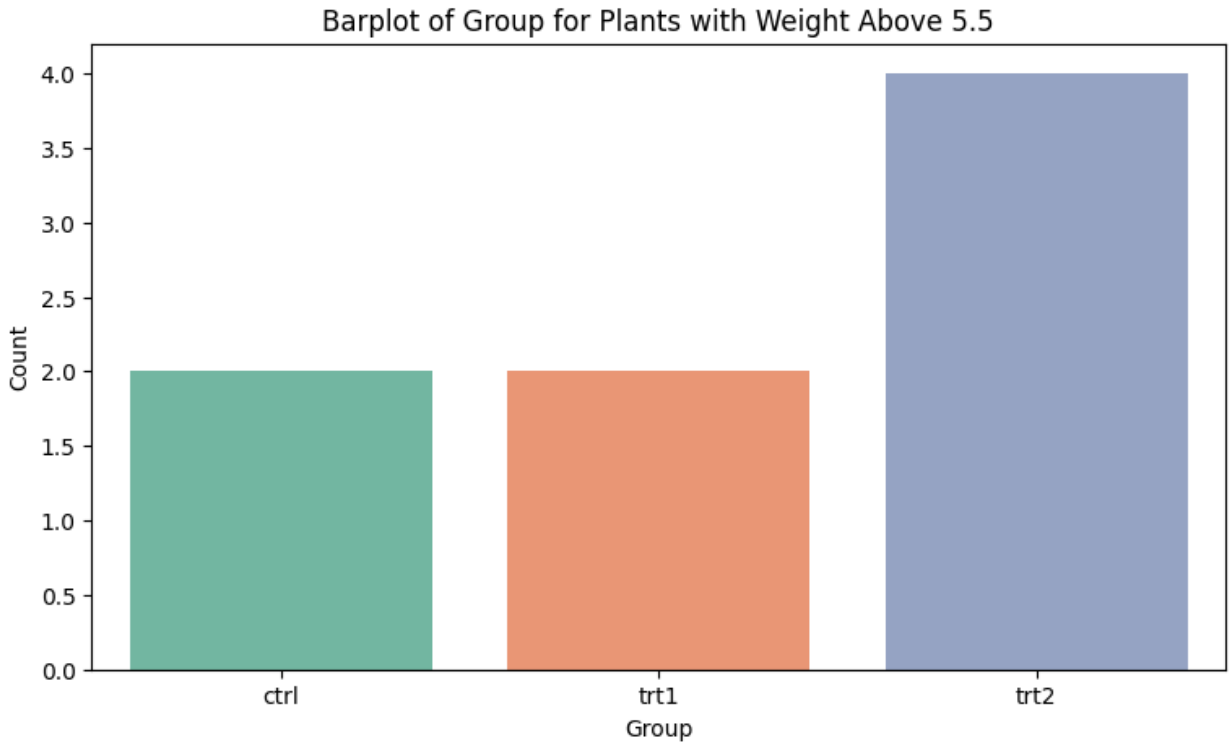
Output:

```
Percentage of 'trt1' weights below the minimum 'trt2' weight: 80.00%
```

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 matplotlib.pyplot as plt

import seaborn as sns

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)

# Filter the data to include only plants with weight > 5.5
filtered_data = PlantGrowth[PlantGrowth['weight'] > 5.5]

# Create a barplot of the 'group' variable for the filtered data
plt.figure(figsize=(9,5))
sns.countplot(x='group', data=filtered_data, palette='Set2')
plt.title('Barplot of Group for Plants with Weight Above 5.5')
plt.xlabel('Group')
plt.ylabel('Count')
plt.show()
```


Barplot of Group for Plants with Weight Above 5.5
Count
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
ctrl
trt1
trt2
Group