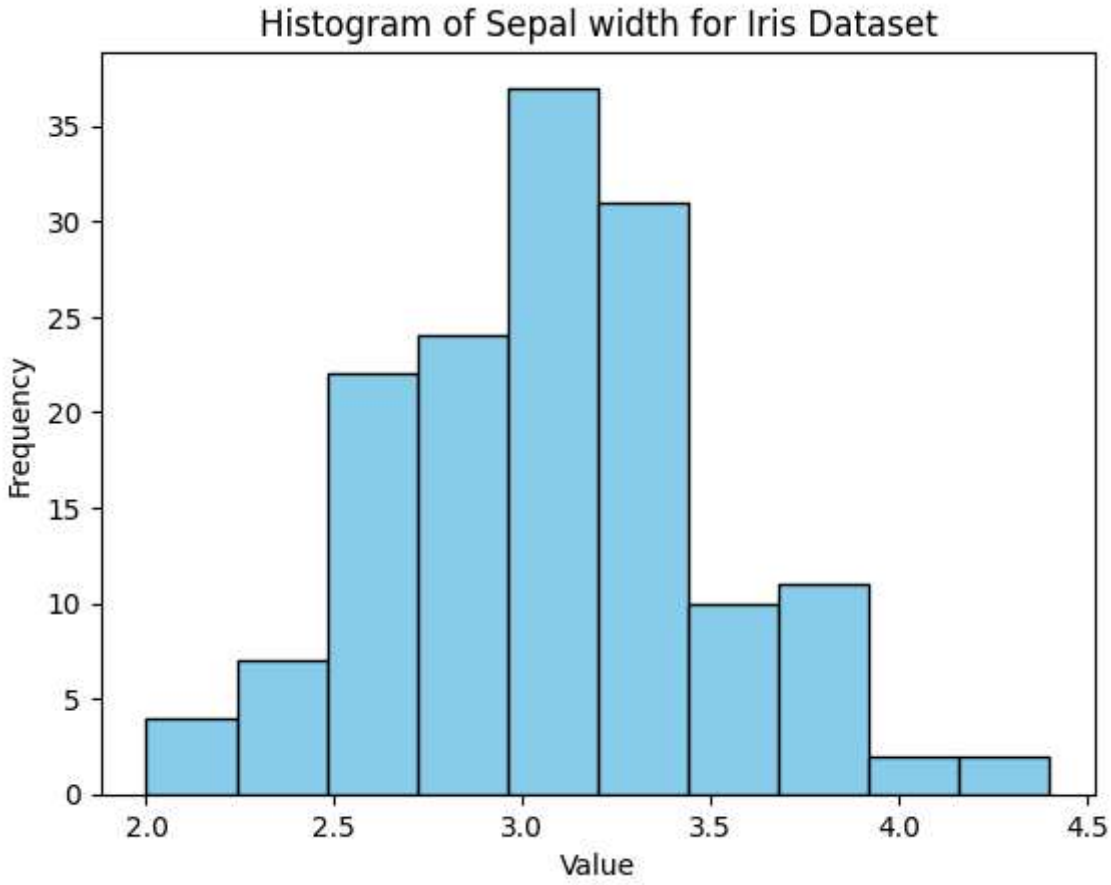
In [2]:
```
#imports
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import datasets

#iris data
iris = datasets.load_iris()

#plantgrowth data
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
PlantGrowth = pd.DataFrame(data)

#put iris in a dataframe
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
```

In [ ]:
```
"""
IRIS DATASET
Question 1a:
    Make a histogram of the variable Sepal.Width
"""
#get the data
plt.hist(iris_df['sepal width (cm)'], color='skyblue', edgecolor='black')

#Titles and labels
plt.title('Histogram of Sepal width for Iris Dataset')
plt.xlabel('Value')
plt.ylabel('Frequency')

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



In [4]:
```
"""
IRIS DATASET
Question 1b: Based on the histogram from #1a,
which would you expect to be higher, the mean or the median? Why?

I would expect the mean to be higher than the median. The reason is because the data appears to be right
skewed slightly. When datasets have a right skew mean > median
"""
```

Out[4]:
```
'\nIRIS DATASET\nQuestion 1b: Based on the histogram from #1a, \nwhich would you expect to be higher, the mean or th
e median? Why?\n\nI would expect the mean to be higher than the median. The reason is because the data appears to be
right\nskewed slightly. When datasets have a right skew mean > median\n'
```

In [5]:
```
"""
IRIS DATASET
Question 1c: Confirm your answer to #1b by actually finding these values.
"""
mean = np.mean(iris_df['sepal width (cm)'])
median = np.median(iris_df['sepal width (cm)'])

print(f'Mean: {mean}')
print(f'Median: {median}')
```

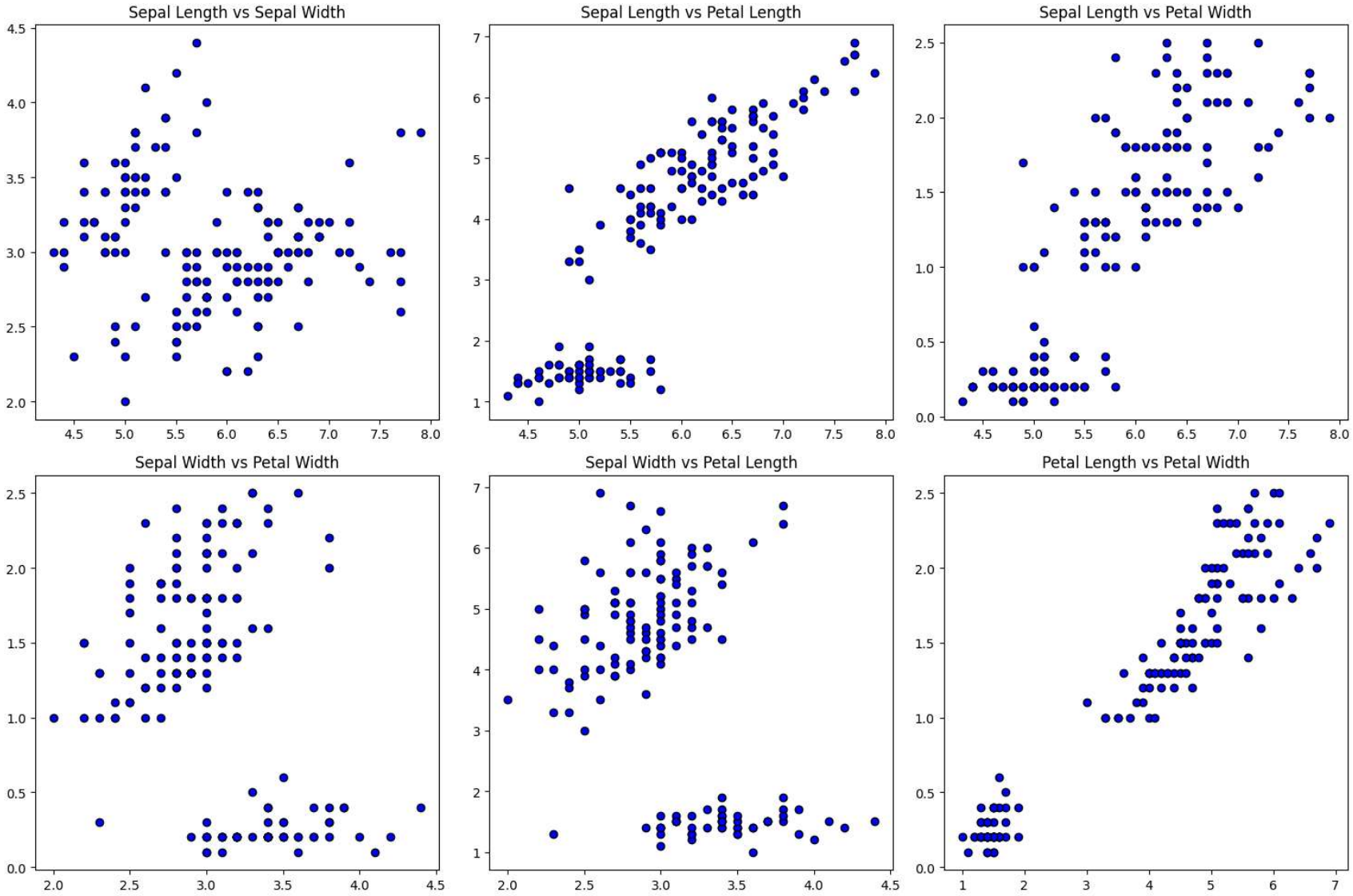
```
Mean: 3.0573333333333337
Median: 3.0
```

In [6]:
```
"""
IRIS DATASET
Question 1d: Only 27% of the flowers have a Sepal.Width higher than ________ cm.
"""
#Percintile is that p% of values are LESS than or equal to the value so since we want values HIGHER
#we need the 73 percentile here.

perc = np.percentile(iris_df['sepal width (cm)'], 73)

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

```
Only 27% of the flowers have a Sepal.Width higher than 3.3 cm.
```

In [ ]:
```
"""
IRIS DATASET
Question 1e: Make scatterplots of each pair of the numerical variables in iris
(There should be 6 pairs/plots).
"""
# Create the scatter plot (matplotlib)
# Create subplots (2 rows, 3 columns)
fig, axs = plt.subplots(2, 3, figsize=(15, 10))

# Scatter plot 1
axs[0, 0].scatter(iris_df['sepal length (cm)'], iris_df['sepal width (cm)'], color='blue', edgecolor='black')
axs[0, 0].set_title('Sepal Length vs Sepal Width')

# Scatter plot 2
axs[0, 1].scatter(iris_df['sepal length (cm)'], iris_df['petal length (cm)'], color='blue', edgecolor='black')
axs[0, 1].set_title('Sepal Length vs Petal Length')
```

```
# Scatter plot 3
axs[0, 2].scatter(iris_df['sepal length (cm)'], iris_df['petal width (cm)'], color='blue', edgecolor='black')
axs[0, 2].set_title('Sepal Length vs Petal Width')

# Scatter plot 4
axs[1, 0].scatter(iris_df['sepal width (cm)'], iris_df['petal width (cm)'], color='blue', edgecolor='black')
axs[1, 0].set_title('Sepal Width vs Petal Width')

# Scatter plot 5
axs[1, 1].scatter(iris_df['sepal width (cm)'], iris_df['petal length (cm)'], color='blue', edgecolor='black')
axs[1, 1].set_title('Sepal Width vs Petal Length')

# Scatter plot 6
axs[1, 2].scatter(iris_df['petal length (cm)'], iris_df['petal width (cm)'], color='blue', edgecolor='black')
axs[1, 2].set_title('Petal Length vs Petal Width')

# Adjust layout for better spacing
plt.tight_layout()

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

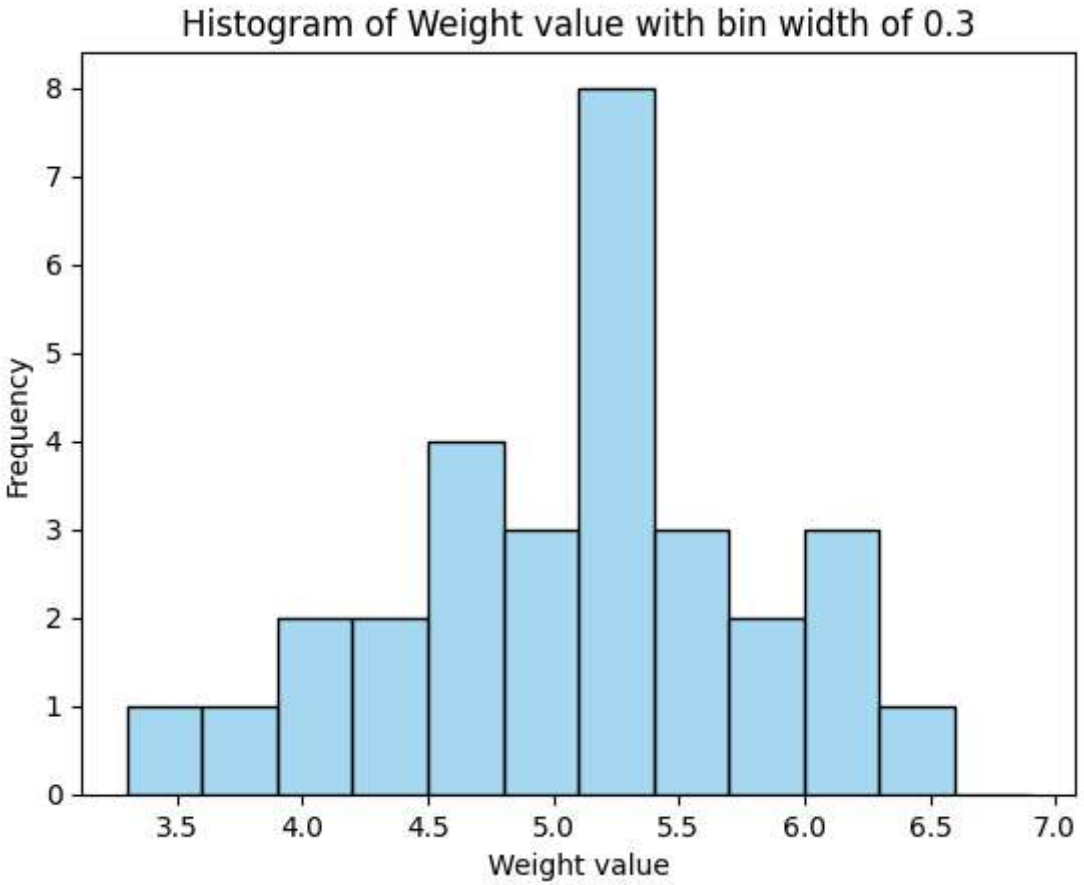


In [8]:
```
"""
IRIS DATASET
Question 1f: Based on #1e, which two variables appear to have the strongest relationship?
And which two appear to have the weakest relationship?

The strongest relationship seems to be between Petal length and Petal width. The weakest relationship
is between Sepal length and Sepal width.
```

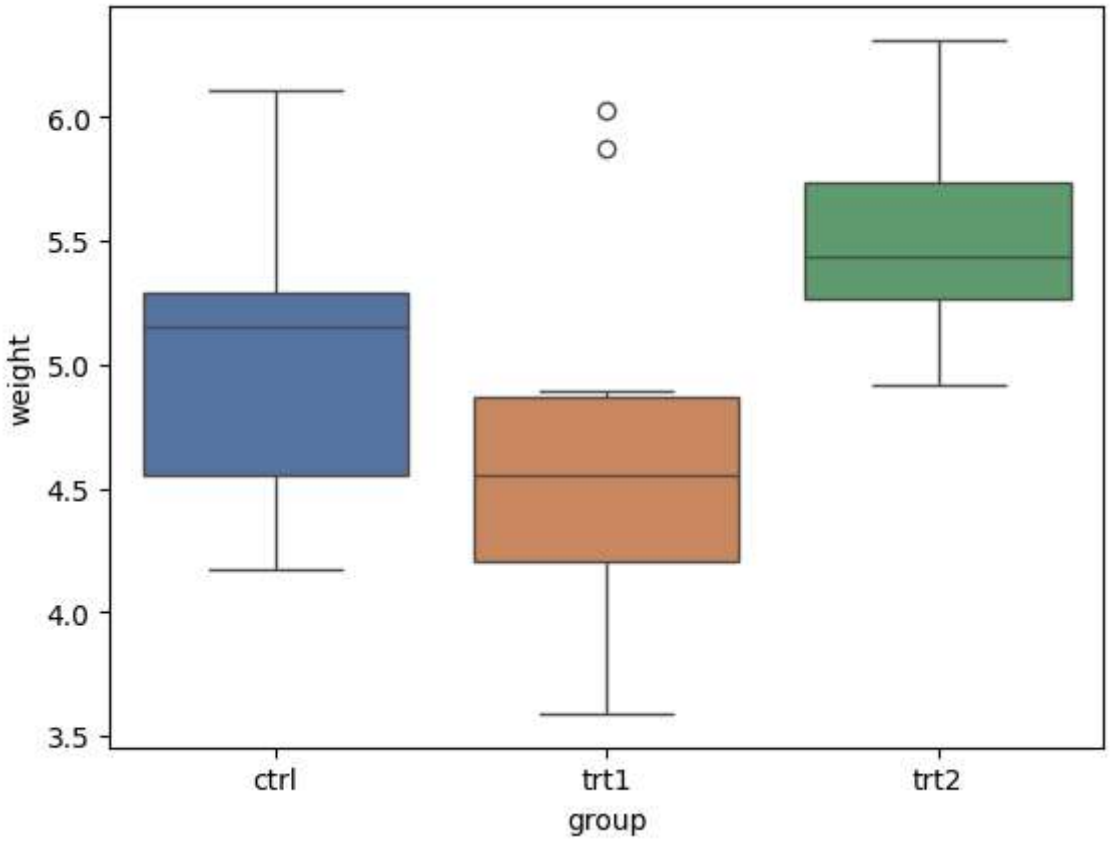
```
"""
```

Out[8]:
```
'\nIRIS DATA SET\nQuestion 1f: Based on #1e, which two variables appear to have the strongest relationship? \nAnd wh
ich two appear to have the weakest relationship?\n\nThe strongest relationship seems to be between Petal length and
Petal width. The weakest relationship\nis between Sepal length and Sepal width.\n\n'
```

In [ ]:
```
"""
PLANT GROWTH DATASET
Question 2a: Make a histogram of the variable weight with breakpoints (bin edges) at every 0.3 units,
starting at 3.3.
"""
# Create histogram with bin width of 0.3 starting at 3.3
sns.histplot(data['weight'], bins = np.arange(3.3, 7, .3), color='skyblue', edgecolor='black')

# Add labels and title
plt.xlabel("Weight value")
plt.ylabel("Frequency")
plt.title("Histogram of Weight value with bin width of 0.3")

plt.show()
```



In [14]:
```
"""
PLANT GROWTH DATASET
Question 2b: Make boxplots of weight separated by group in a single graph
"""
#build the box plot of weight
sns.boxplot(x='group', y='weight',  hue='group', data=data, palette='deep', legend=False)

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



In [ ]:
```
"""
PLANT GROWTH DATASET
Question 2c: Based on the boxplots in #2b, approximately what percentage of the "trt1" weights are below the minimum

Approximately 80% of the "trt1" weights are below the minimum "trt2" weight

"""
```

In [24]:
```
"""
PLANT GROWTH DATASET
Question 2d: Find the exact percentage of the "trt1" weights that are below the minimum "trt2" weight
"""
#find min of trt2 and find percent of values that are less than that value

trt2_min = PlantGrowth.loc[PlantGrowth["group"] == "trt2", "weight"].min()
```

```
trt1_values = PlantGrowth.loc[PlantGrowth["group"] == "trt1", "weight"]


#how many values are greater than trt2_min
counter = 0
for i in trt1_values:
    if i < trt2_min:
        counter += 1

#should be 10:
trt1_amount = len(trt1_values)

#Get percent of values:
percent = (counter/trt1_amount) * 100


print(f"Exact percentage of the trt1 weights that are below the minimum trt2 weight: {percent}%")
```

```
Exact percentage of the trt1 weights that are below the minimum trt2 weight: 80.0%
```

In [39]:

```
"""
PLANT GROWTH DATASET
Question 2e: Only including plants with a weight above 5.5, make a barplot of the variable group. Make the barplot co
"""
# Filter the data to include only weights >= 5.5
filtered_data = PlantGrowth[PlantGrowth["weight"] > 5.5]["group"].value_counts()




# Get the frequency of the filtered weights
labels_int = filtered_data.index.tolist()
labels = list(map(str, labels_int)) # convert those integers to strings
values = filtered_data.values

#Palette
colors = sns.color_palette("Set2", len(labels))

#make the plot
plt.bar(labels, values, color = colors)
```

```
plt.title("barplot of the variable group where weight is above 5.5")
plt.xlabel("Weight")
plt.ylabel("Values")
plt.show()
```