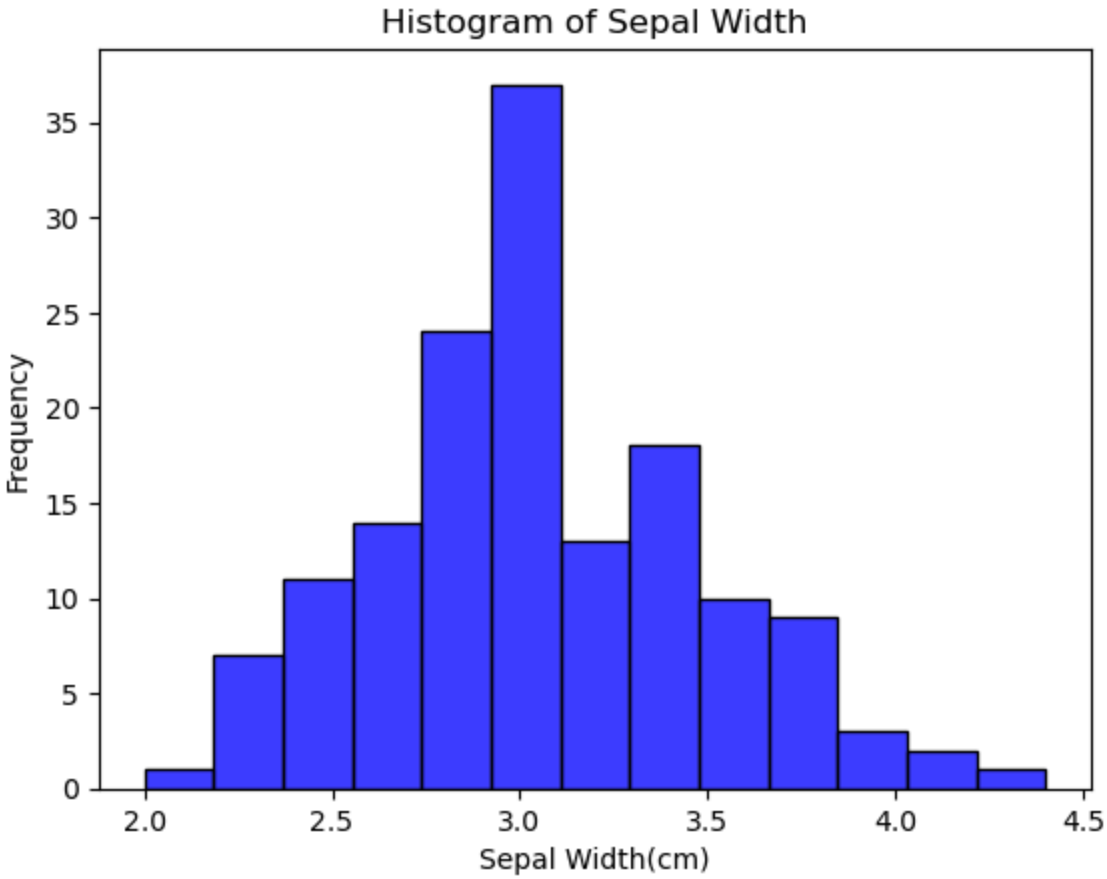
1a. Make a histogram of the variable Sepal.Width.

In [35]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
iris = load_iris(as_frame=True)
df = iris.frame
print(df['sepal width (cm)'])
sns.histplot(df['sepal width (cm)'],color='blue',edgecolor='black')
plt.title('Histogram of Sepal Width')
plt.xlabel('Sepal Width(cm)')
plt.ylabel('Frequency')
plt.show()
```

```
0      3.5
1      3.0
2      3.2
3      3.1
4      3.6
      ...
145    3.0
146    2.5
147    3.0
148    3.4
149    3.0
Name: sepal width (cm), Length: 150, dtype: float64
```



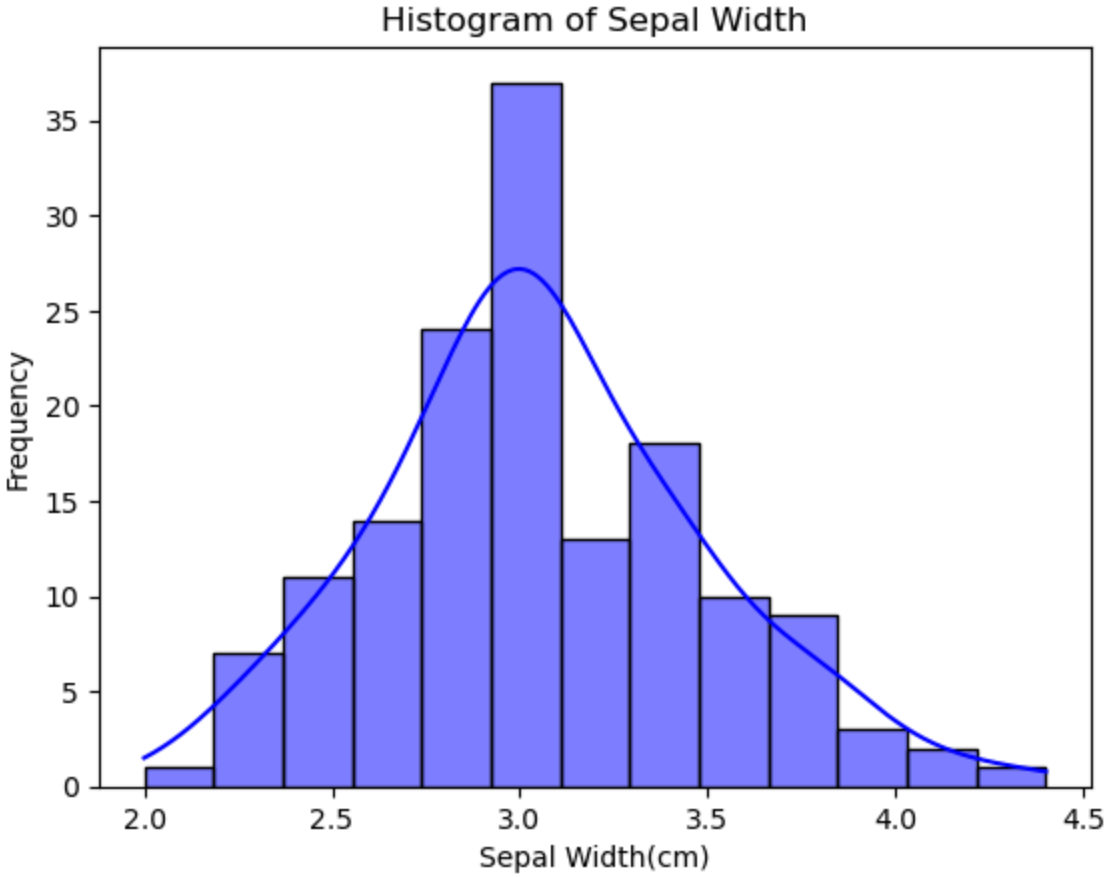
1b. Based on the histogram from #1a, which would you expect to be higher, the mean or the median? Why?

The sepal width values mean is expected to be slightly higher than the median. The histogram shows slightly-right skewed or positively skewed because most values are clusters around 3.0 cm and few flowers have larger sepal width around 4 cm that create a right tail.

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

In [44]:
```
import numpy as np
from sklearn.datasets import load_iris
iris = load_iris(as_frame=True)
df = iris.frame
print(df['sepal width (cm)'])
mean=np.mean(df['sepal width (cm)'])
median=np.median(df['sepal width (cm)'])
print(f"Mean:{mean}")
print(f"Median:{median}")
sns.histplot(df['sepal width (cm)'],color='blue',kde=True)
plt.title('Histogram of Sepal Width')
plt.xlabel('Sepal Width(cm)')
plt.ylabel('Frequency')
plt.show()
```

```
0      3.5
1      3.0
2      3.2
3      3.1
4      3.6
      ...
145    3.0
146    2.5
147    3.0
148    3.4
149    3.0
Name: sepal width (cm), Length: 150, dtype: float64
Mean:3.0573333333333337
Median:3.0
```



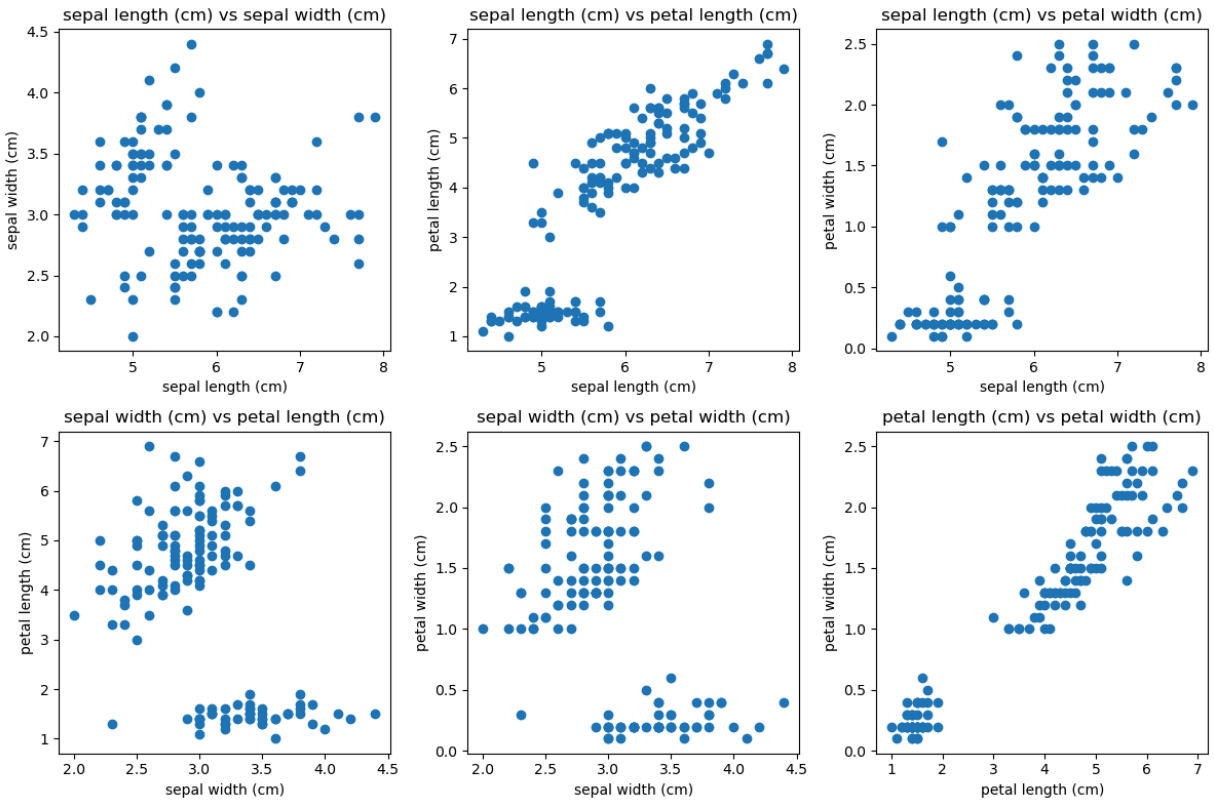
1d. Only 27% of the flowers have a Sepal.Width higher than _____ cm.

In [8]:
```
from sklearn.datasets import load_iris
iris = load_iris(as_frame=True)
df = iris.frame
sw=df['sepal width (cm)']
cutoff=sw.quantile(0.73)
pct_above=(sw>cutoff).mean()*100
print(f"cutoff(cm={cutoff:.3f})")
print(f"only 27% of the flowers have a sepal width higher than {cutoff:.3f}"
```

```
cutoff(cm=3.300)
only 27% of the flowers have a sepal width higher than 3.300
```

1e. Make scatterplots of each pair of the numerical variables in iris (There should be 6 pairs/plots).

In [ ]:
```
pairs = [
    ('sepal length (cm)', 'sepal width (cm)'),
    ('sepal length (cm)', 'petal length (cm)'),
    ('sepal length (cm)', 'petal width (cm)'),
    ('sepal width (cm)', 'petal length (cm)'),
    ('sepal width (cm)', 'petal width (cm)'),
    ('petal length (cm)', 'petal width (cm)')
]
plt.figure(figsize=(12, 8))
for i, (x, y) in enumerate(pairs, 1):
    plt.subplot(2, 3, i)
    plt.scatter(df[x], df[y])
    plt.xlabel(x)
    plt.ylabel(y)
    plt.title(f"{x} vs {y}")
plt.tight_layout()
plt.show()
```



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

The scatterplots in #1e indicate that petal length and petal width share the closest relationship, with a clear, almost linear pattern, reflecting a strong positive correlation. In contrast, sepal width and petal length are more dispersed, showing a weaker connection.

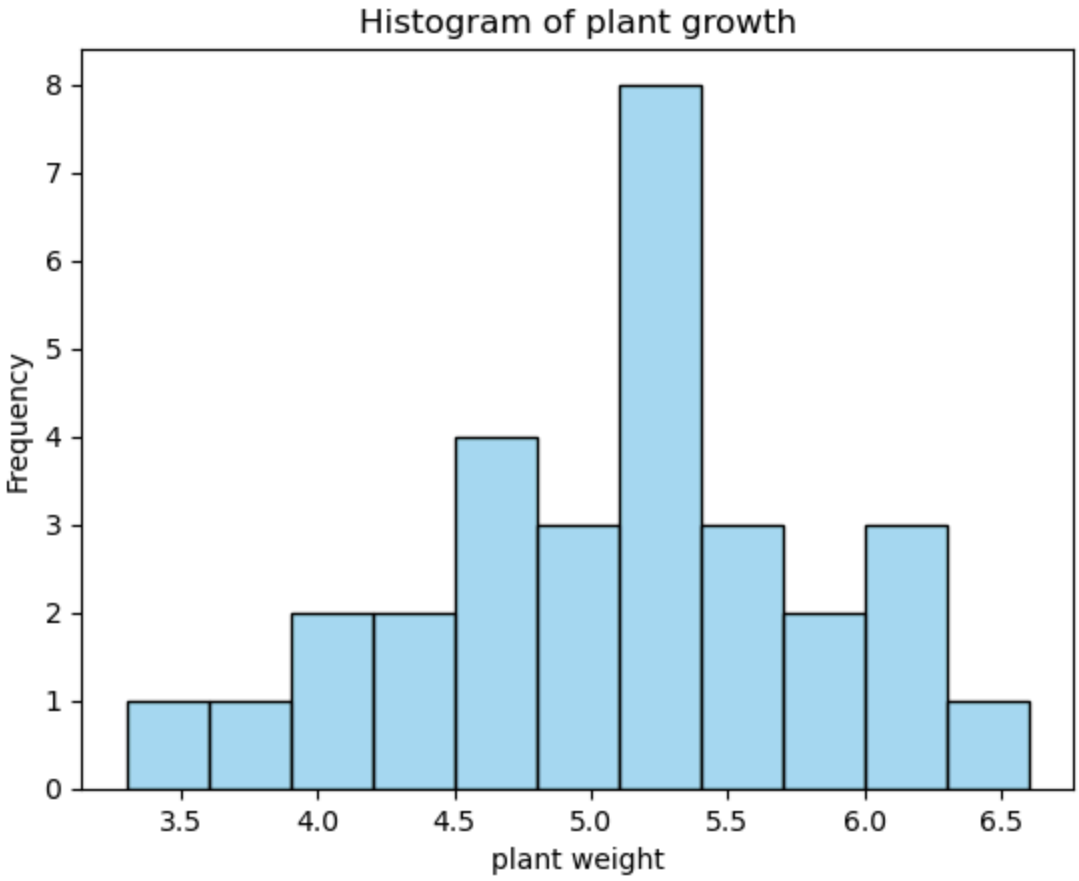
These impressions are supported by correlation coefficients. The correlation between petal length and petal width is 0.963, the strongest among all pairs. Conversely, sepal width and petal length have the weakest association with a correlation of −0.428.

In [ ]:
```
pairs = [
    ('sepal length (cm)', 'sepal width (cm)'),
    ('sepal length (cm)', 'petal length (cm)'),
    ('sepal length (cm)', 'petal width (cm)'),
    ('sepal width (cm)', 'petal length (cm)'),
    ('sepal width (cm)', 'petal width (cm)'),
    ('petal length (cm)', 'petal width (cm)')
]
for x, y in pairs:
    corr = df[x].corr(df[y])
    print(f"{x} vs {y}: {corr:.3f}")
```

```
sepal length (cm) vs sepal width (cm): -0.118
sepal length (cm) vs petal length (cm): 0.872
sepal length (cm) vs petal width (cm): 0.818
sepal width (cm) vs petal length (cm): -0.428
sepal width (cm) vs petal width (cm): -0.366
petal length (cm) vs petal width (cm): 0.963
```

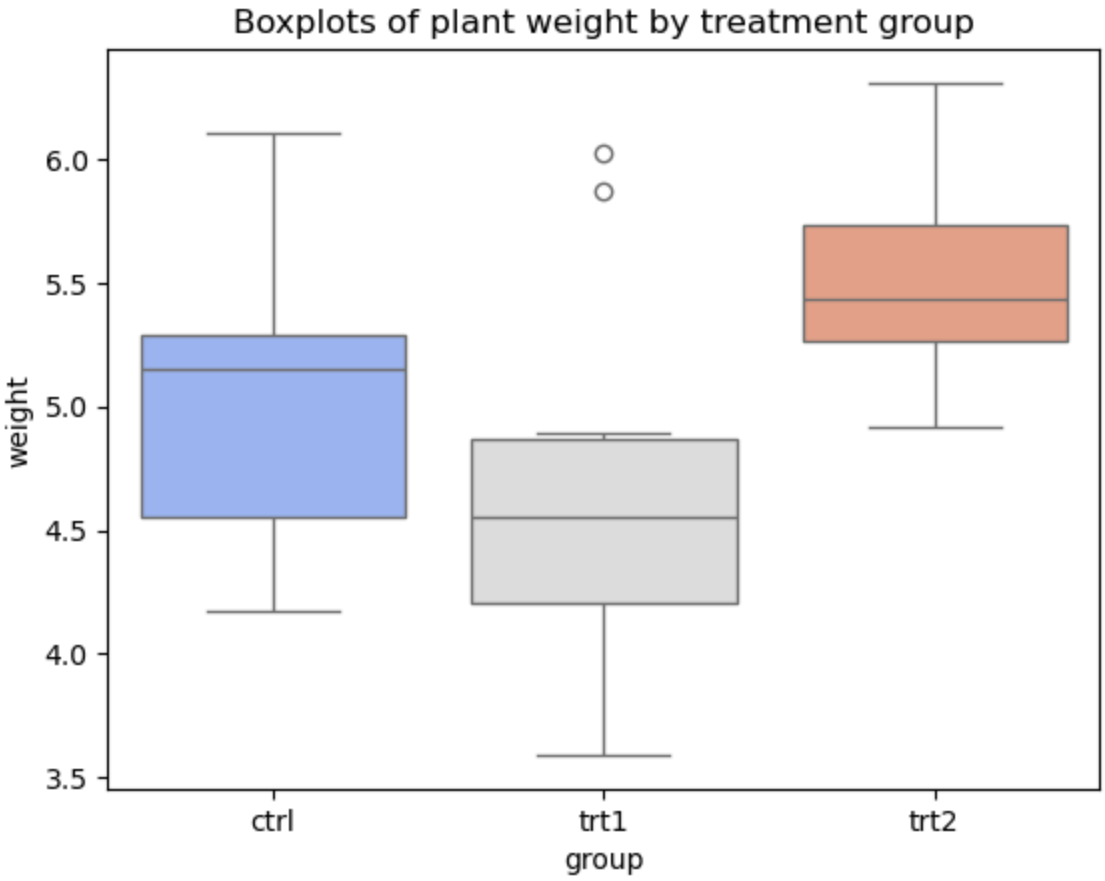
2a. Make a histogram of the variable weight with breakpoints (bin edges) at every 0.3 units, starting at 3.3.

In [75]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = { "weight": [4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.
        "group": ["ctrl"] * 10 + ["trt1"] * 10 + ["trt2"] * 10}
PlantGrowth = pd.DataFrame(data)
bins=np.arange(3.3,max(data['weight'])+0.3,0.3)
sns.histplot(data['weight'],bins=bins,color='skyblue')
plt.title('Histogram of plant growth')
plt.xlabel('plant weight')
plt.ylabel('Frequency')
plt.show()
```



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

In [83]:
```
data = { "weight": [4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.
       "group": ["ctrl"] * 10 + ["trt1"] * 10 + ["trt2"] * 10}
PlantGrowth = pd.DataFrame(data)
sns.boxplot(x='group',y='weight',hue='group',data=data,palette='coolwarm')
plt.title('Boxplots of plant weight by treatment group')
plt.show()
```



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

From the boxplot, the minimum weights of trt2 is near 5.0 and most of the trt1 weights lies below that value. Approximately 75-80% of the trt1 distribution is below the minimum weights of trt2.

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

In [ ]:
```
min_trt2 = PlantGrowth[PlantGrowth["group"] == "trt2"]["weight"].min()

# Get all trt1 weights
trt1_weights = PlantGrowth[PlantGrowth["group"] == "trt1"]["weight"]

# Calculate exact percentage
percentage = (trt1_weights < min_trt2).sum() / len(trt1_weights) * 100

print("Minimum trt2 weight:", min_trt2)
print("Percentage of trt1 below min(trt2):", percentage, "%")
```

```
Minimum trt2 weight: 4.92
Percentage of trt1 below min(trt2): 80.0 %
```

2e. Only including plants with a weight above 5.5, make a barplot of the variable group. Make the barplot colorful using some color palette

In [ ]:
```
data = { "weight": [4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.
        "group": ["ctrl"] * 10 + ["trt1"] * 10 + ["trt2"] * 10}
PlantGrowth = pd.DataFrame(data)
heavy_plants = PlantGrowth [PlantGrowth ['weight'] > 5.5]
sns.countplot(x='group',hue='group',legend=False,data=heavy_plants, palette=
plt.title('Number of heavy Plants by Treatment Group')
plt.xlabel('Treatment')
plt.ylabel('Number of Plants')
plt.show()
```