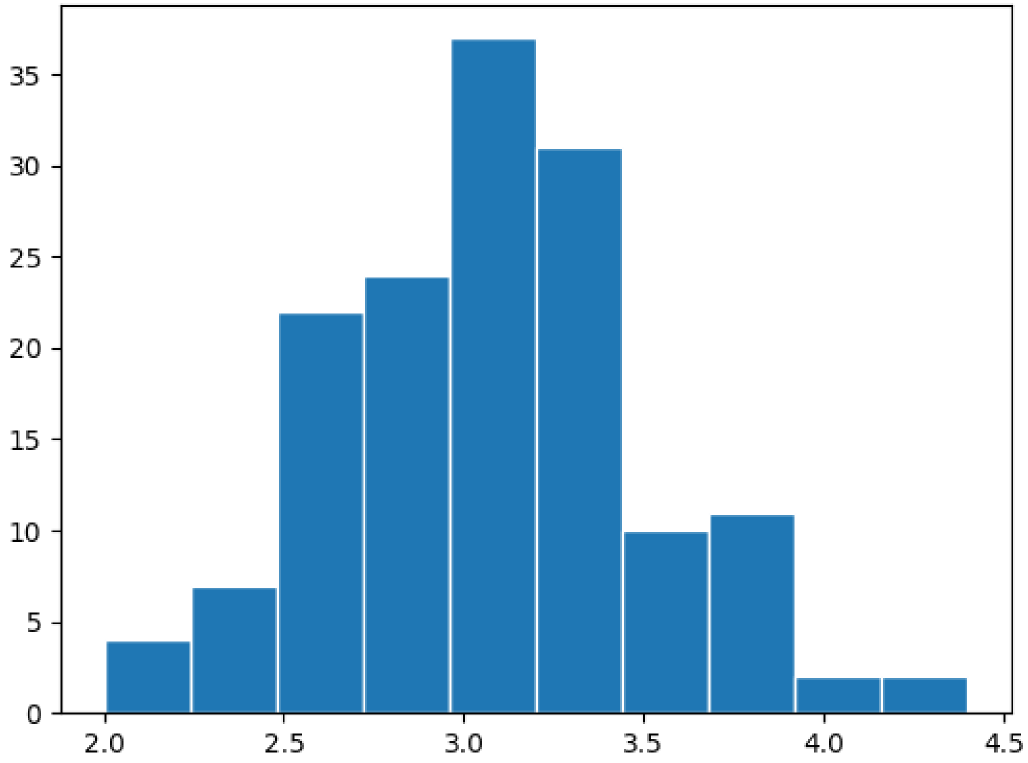
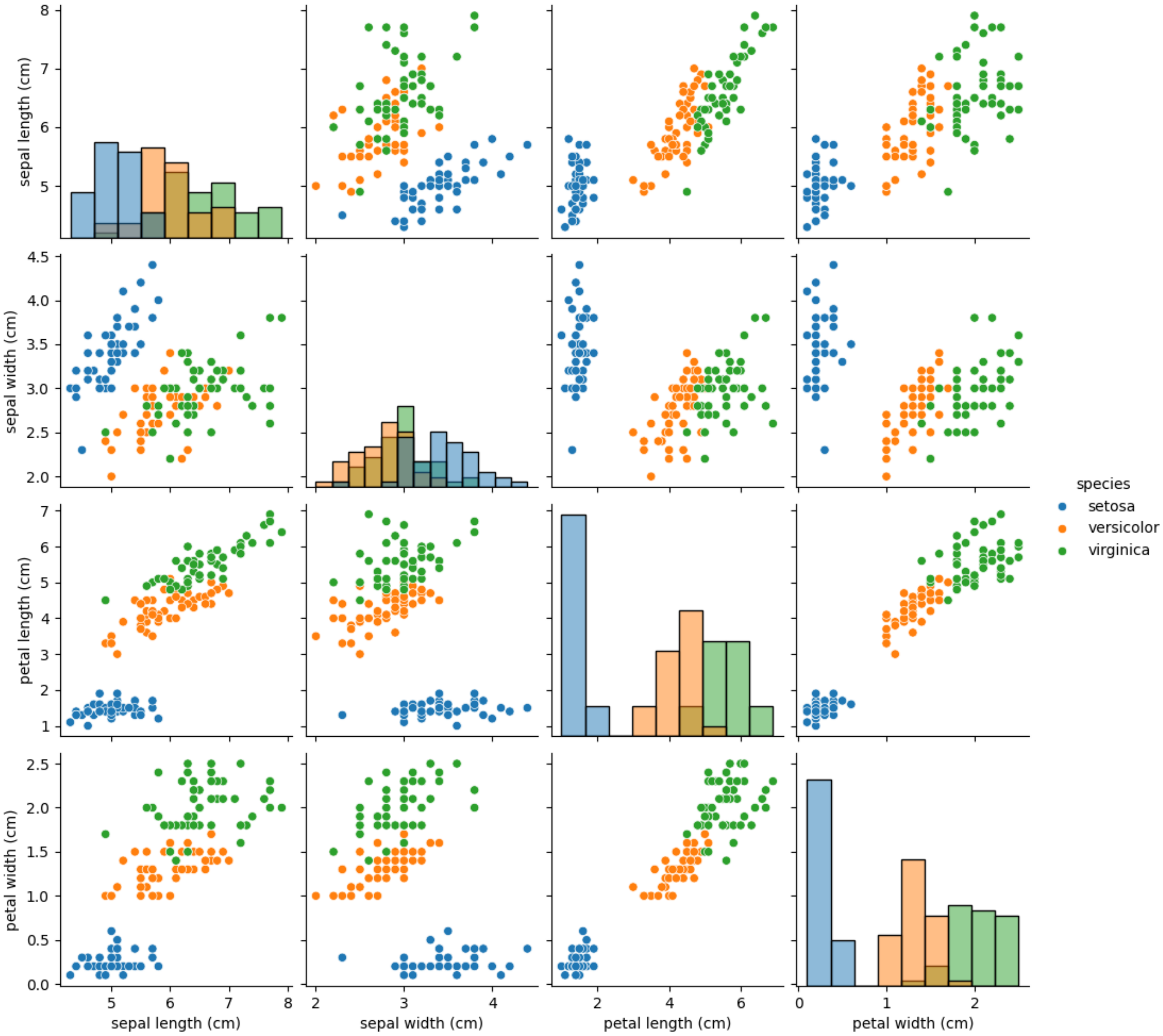
```
from sklearn import datasets
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


iris = datasets.load_iris()
weight_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(weight_data)

X = iris.data
y = iris.target

df = pd.DataFrame(data=X, columns=iris.feature_names)

df['species'] = iris.target_names[iris.target]

# Make a histogram of the variable Sepal.Width.

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

plt.hist(sw, edgecolor='white')
plt.show()
```



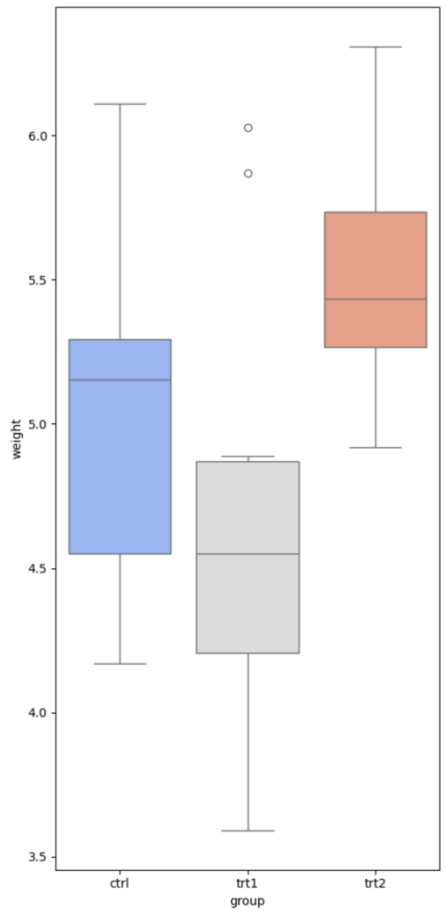
```
#Based on the histogram from #1a, which would you expect to be higher,
#the mean or the median? Why?

#answer: I would say they are really close, with median being just slightly larger

# print(sw.mean())
# print(sw.median())

#answer:
# 3.057337
# 3.0

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

twenty_seven = sw.quantile(0.73)
```

```
print(f"Only 27% of the flowers have a sepal width higher than {twenty_seven} cm")

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

sns.pairplot(df, hue='species', diag_kind='hist')

plt.show()
```



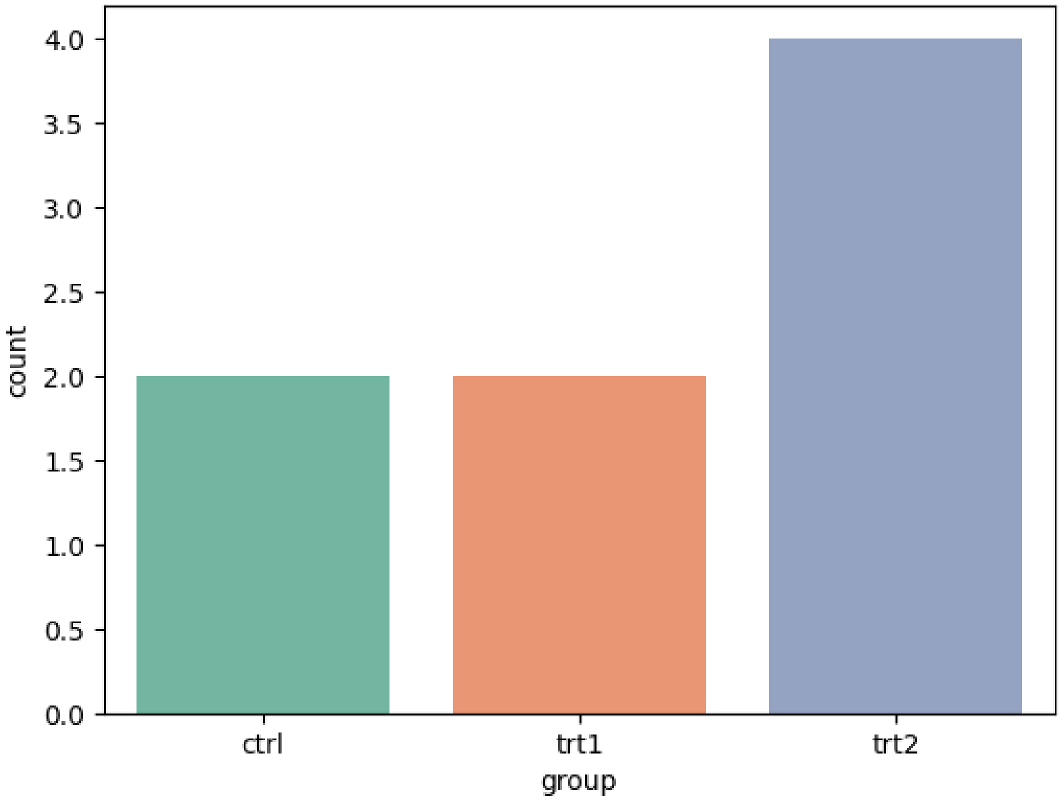
```
# del df['species']

# print(df)
```

```
print(df.corr())

# confirm the correlation

# Based on #1e, which two variables appear to have the strongest relationship?
# petal width and petal length
# And which two appear to have the weakest relationship?
# sepal length and sepal width

#part 2

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

binwidth = 0.3

wt = PlantGrowth["weight"]

plt.hist(wt, bins = np.arange(3.3, wt.max() + 0.3, 0.3), edgecolor='white' )

# plt.show()

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

sns.boxplot(x='group', y='weight', data=PlantGrowth, palette='coolwarm')
plt.show()
```



```
# Based on the boxplots in #2b, approximately what percentage of the "trt1" weights are below the minimum "trt2" weight?
# answer: all of trt1 but two are below min of trt2, so 80% since each group has 10 elements

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

trt2_min = (PlantGrowth["weight"][PlantGrowth['group'] == 'trt2']).min()

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

trt1_larger_than_min = (trt1[trt1 > trt2_min]).count()
```

```
# print((trt1_larger_than_min * 100) / trt1.count())

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

heavier_than = PlantGrowth[PlantGrowth["weight"] > 5.5]

sns.countplot(data=heavier_than, x='group', palette="Set2")
plt.show()
```


count
ctrl
trt1
trt2
group