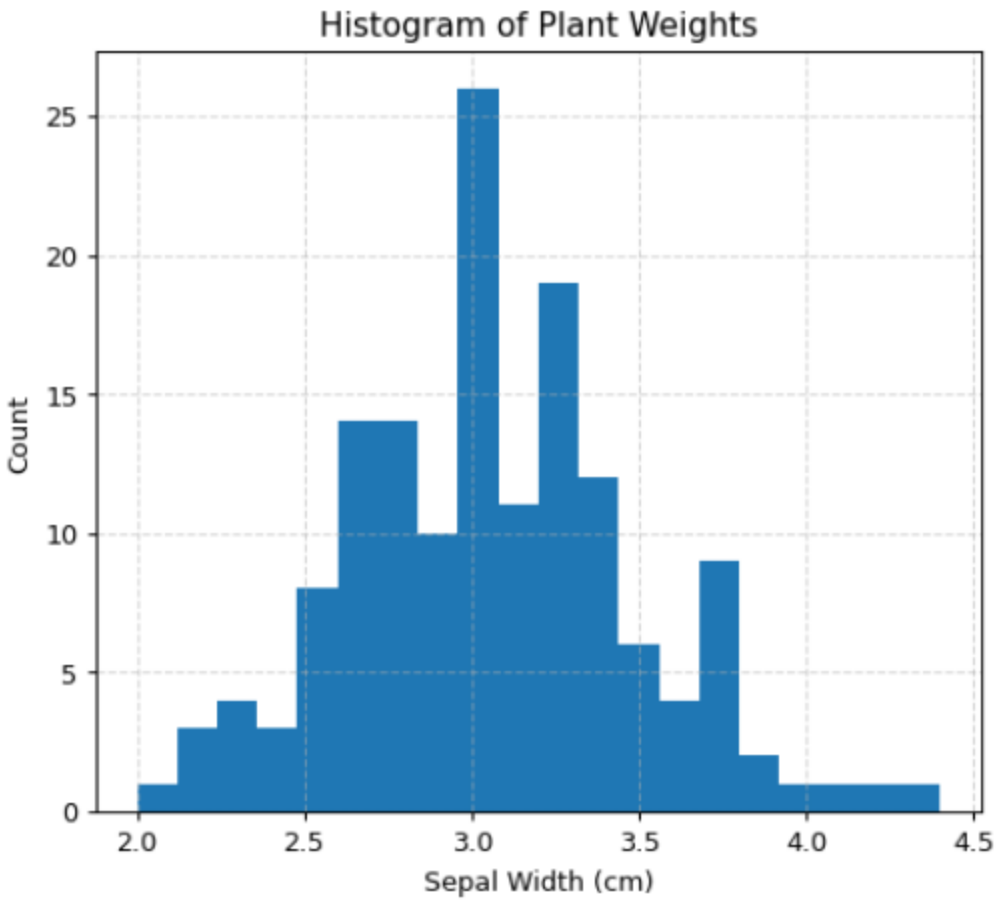
```
from sklearn import datasets

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)

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np



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

irisNames = pd.DataFrame(iris.target, columns=['Names'])

irisTotal = pd.concat([irispd, irisNames], ignore_index=False, axis=1)

irisTotal['Names'] = irisTotal['Names'].apply(lambda x: iris.target_names[x])


# Problem 1.A

plt.close()

plt.hist(irisTotal['sepal width (cm)'], bins=20)

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

plt.ylabel("Count")

plt.show()
```



Problem 1.B

#I expect the mean to be almost the same a the medium. hard to tell.

Problem 1.C

irisTotal.describe()

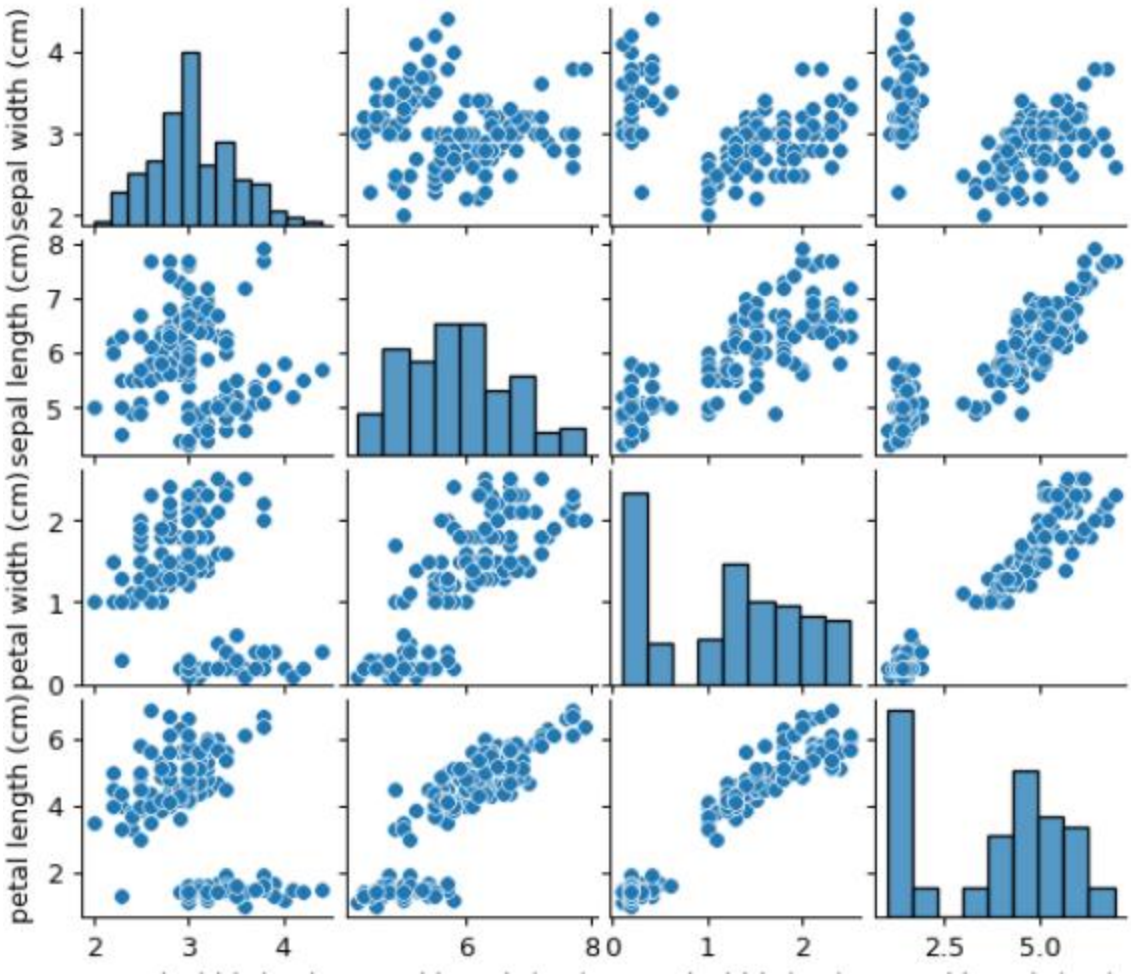
Mean is 5.843. Medium is 5.8

Problem 1.D

sepalAmount = round(150*.27)

swOrder = irisTotal.sort_values(by='sepal width (cm)', ascending=False)

```
sw27 = swOrder.head(sepalAmount)

print('Only 27% of the flowers have a Sepal.Width higher than {x}', sw27['sepal width (cm)'].iloc[-1] - 0.01)

#3.29


# Problem 1.E


columns_to_plot = ['sepal width (cm)', 'sepal length (cm)', 'petal width (cm)','petal length (cm)']

plt.close()

sns.pairplot(irisTotal[columns_to_plot], markers='o')

plt.show()
```



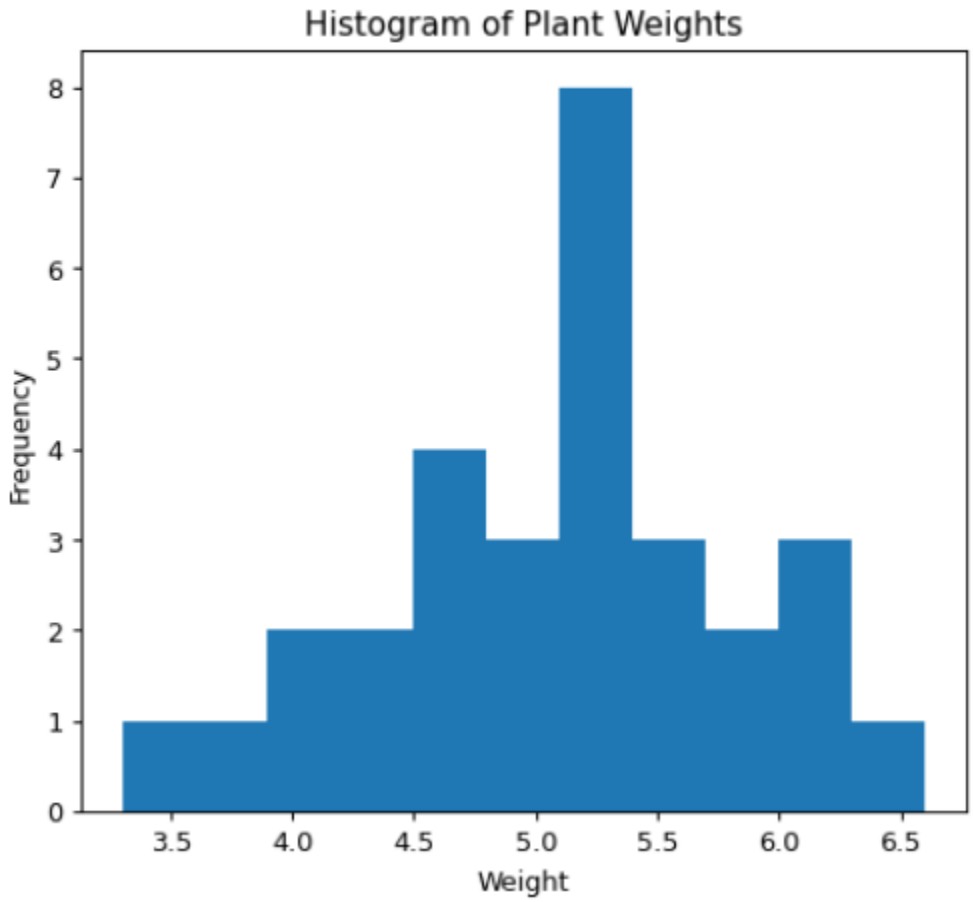
Problem 1.F

#Petal Length and Petal Width seems to have the strongest relationship. Sepal Length and Sepal Width have the weakest relationship.

#Problem 2.A

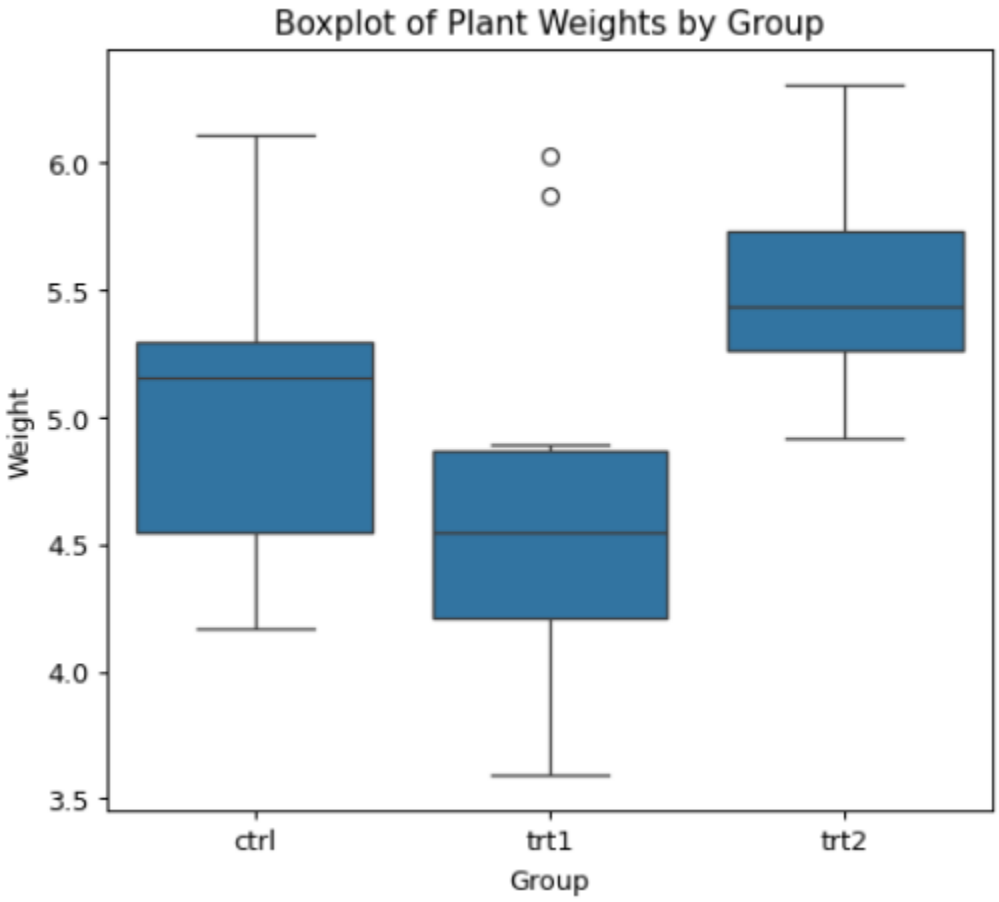
```
plt.close()
bin_edges = [round(x, 2) for x in list(
   pd.Series([3.3 + 0.3 * i for i in range(int((PlantGrowth['weight'].max() - 3.3) / 0.3) + 2)])
)]
```

```
plt.figure(figsize=(8, 5))
plt.hist(PlantGrowth['weight'], bins=bin_edges)
plt.title('Histogram of Plant Weights')
plt.xlabel('Weight')
plt.ylabel('Frequency')
plt.show()
```




```
#Problem 2.B
plt.close()
plt.figure(figsize=(8, 5))
```

```
sns.boxplot(x='group', y='weight', data=PlantGrowth)
plt.title('Boxplot of Plant Weights by Group')
plt.xlabel('Group')
plt.ylabel('Weight')
plt.show()
```



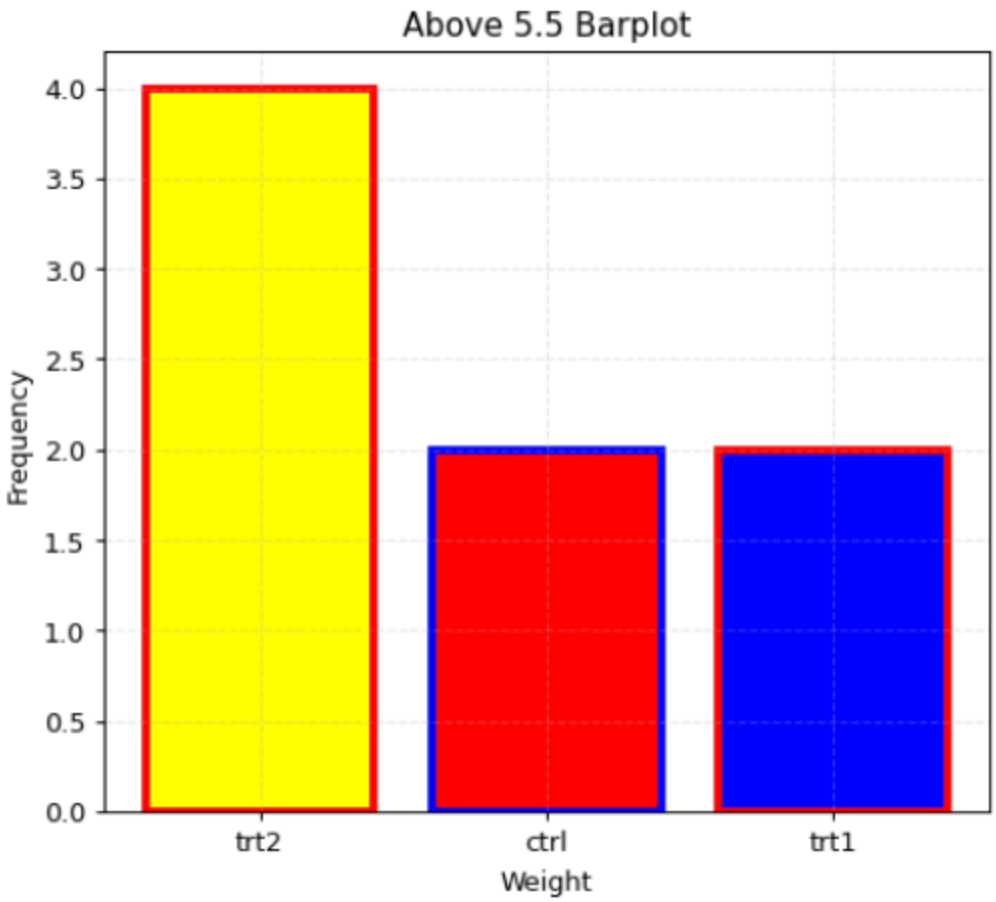
#Problem 2.C

#80%. Only outliers are above the min of trt2 from trt1

#Problem 2.D

```
groupCounts = PlantGrowth['group'].value_counts()
```

```
print(groupCounts) #10 for trt1
trt1 = PlantGrowth.loc[PlantGrowth['group'] == 'trt1']
trt2 = PlantGrowth.loc[PlantGrowth['group'] == 'trt2']
trt2Min = trt2[PlantGrowth['group'] == 'trt2']['weight'].min()
countBelow = (trt1['weight']<trt2Min).sum()
percent = countBelow/10 * 100
print(f'{percent:.2f}% of trt1 weights are below trt2 min weight')
#80%

#Problem 2.E
above55 = PlantGrowth.loc[PlantGrowth['weight']>5.5]
frequency_table = above55['group'].value_counts()
labels_int = frequency_table.index.tolist()
labels = list(map(str, labels_int))
values = frequency_table.values
colors = ['yellow', 'red', 'blue']
edgecolor = ['red', 'blue', 'red']
plt.close()
plt.figure(figsize=(8, 5))
bars = plt.bar(labels, values, color=colors, edgecolor=edgecolor, linewidth=3)
plt.title("Above 5.5 Barplot")
plt.xlabel('Weight')
plt.ylabel('Frequency')
plt.grid(True, linestyle='--', alpha=0.3)
plt.show()
```


Above 5.5 Barplot
Frequency
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
trt2
ctrl
trt1
Weight