```
from sklearn import datasets
import pandas as pd
import numpy as np

iris = datasets.load_iris()

X = iris.data
y = iris.target

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

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

rows, cols = df.shape

print(rows, cols)
```

answer: 150 5 (if we include species as well)

```
# Create a new data frame named sub1 which includes only the first 9 rows and the last row of iris.
# Display sub1

first_nine_rows = df.iloc[:9]

last_row = df.iloc[[-1]]

sub1 = pd.concat([first_nine_rows, last_row])

print(sub1)
```

answer:

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)    species
0                5.1               3.5                1.4               0.2     setosa
1                4.9               3.0                1.4               0.2     setosa
2                4.7               3.2                1.3               0.2     setosa
3                4.6               3.1                1.5               0.2     setosa
4                5.0               3.6                1.4               0.2     setosa
5                5.4               3.9                1.7               0.4     setosa
6                4.6               3.4                1.4               0.3     setosa
7                5.0               3.4                1.5               0.2     setosa
8                4.4               2.9                1.4               0.2     setosa
```

```
149          5.9         3.0          5.1          1.8  virginica
```

```
# Create a vector named Versicolor_Is_The_Best, which has the value 100 whenever Species/target is
# equal to "versicolor" and has the value 0 otherwise.
# Display Versicolor_Is_The_Best.

Versicolor_Is_The_Best = []

for sp in df['species']:
    if sp == 'versicolor':
        Versicolor_Is_The_Best.append(100)
    else:
        Versicolor_Is_The_Best.append(0)

Versicolor_Is_The_Best = pd.Series(Versicolor_Is_The_Best)

print(Versicolor_Is_The_Best)



#Save the column named Sepal.Width as its own vector named sw.
# Use functions to find the mean, median, maximum, and minimum of sw.

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

min = min(sw)
max = max(sw)
sw_mean = np.mean(sw)
sw_median = np.median(sw)

print(f"Min: {min}, max: {max}, mean: {sw_mean}, median: {sw_median}")



print("The sum when is:", sum)
print("The number of loops is:", counter)
```

```
# Create a new data frame named sub2 which includes only rows in iris where the Sepal.
# Width is less than 2.4 and includes only the columns Sepal.Length,
# Sepal.Width, and Species/target. Display sub2.
```

```
sub2 = df.loc[df['sepal width (cm)'] < 2.4,
        ['sepal length (cm)', 'sepal width (cm)', 'species']]
print(sub2)
```

answer:

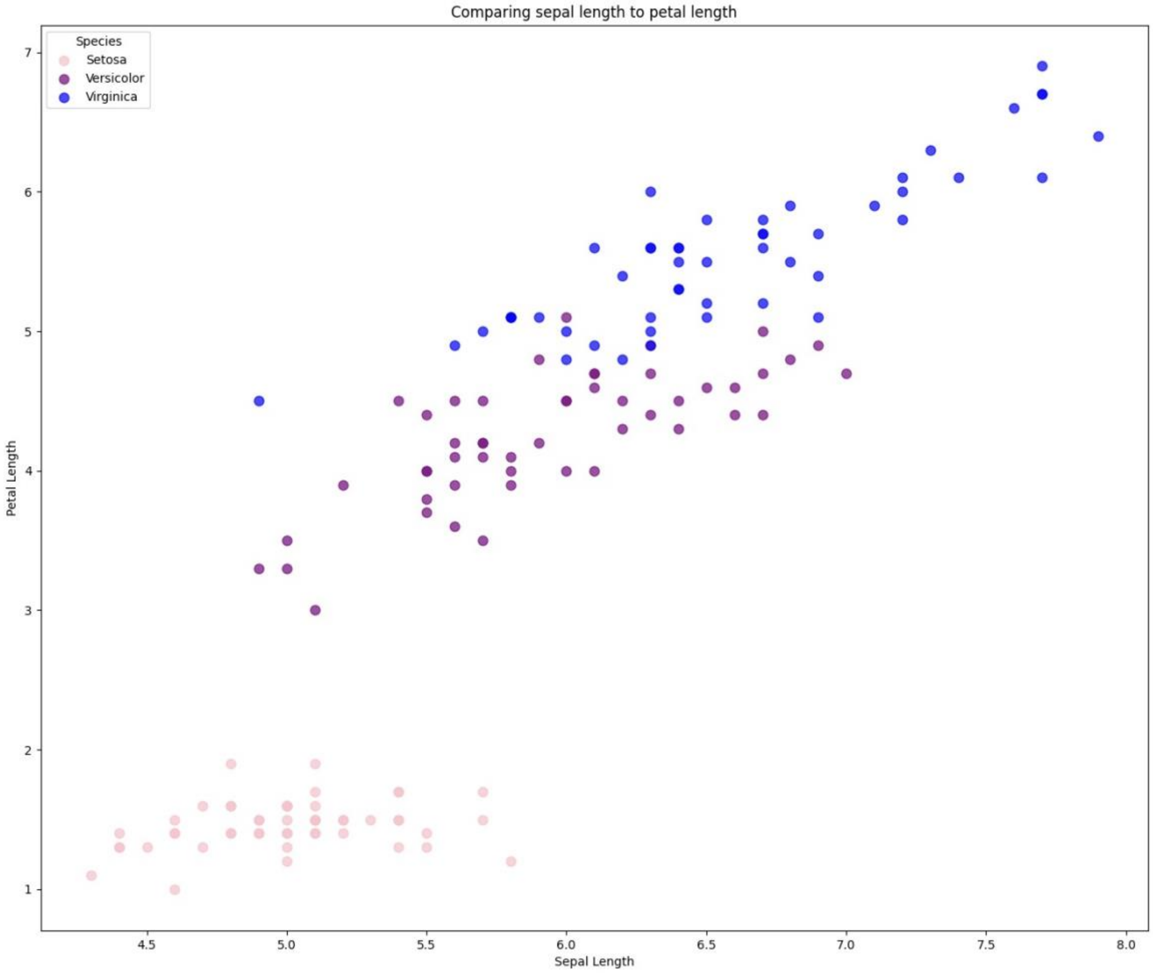
```
     sepal length (cm)  sepal width (cm)     species
41                 4.5               2.3      setosa
53                 5.5               2.3  versicolor
60                 5.0               2.0  versicolor
62                 6.0               2.2  versicolor
68                 6.2               2.2  versicolor
87                 6.3               2.3  versicolor
93                 5.0               2.3  versicolor
119                6.0               2.2   virginica
```

```
#Use a loop to add up the values in sw one at a time until the sum first exceeds 100.
# What is this sum, and how many times did the loop have to execute to reach it?
# For example, if we were adding up the values in sw until the sum first exceeded 10 instead,
# the answer would be 3.5+3.0+3.2+3.1 = 12.8 after 4 loops.
# Also, be aware this could be done more efficiently in R without using loops, but I want you to get the practice.

sum = 0.0
counter = 0


for value in sw:
    sum += value
    counter += 1
    if sum > 100:
        break
```

```
# Create a new function called cmtoin() that converts centimeters to inches (1 inch = 2.54 cm).
# The values in sw are currently recorded in centimeters.
# Apply your function to sw and save the result as a new vector named sw_in. Display the first 7 values of sw_in.

def cmtoin():
    sw_in = []
    for value in sw:
        sw_in.append(value / 2.54)
    print(sw_in[:7])

cmtoin()
```

```
[1.3779527559055118, 1.1811023622047243, 1.2598425196850394, 1.220472440944882, 1.4173228346456692,
1.5354330708661417, 1.3385826771653542]
```

```
# Create a plot that compares Sepal.Length to Petal.Length.
# Add some informative/interesting axis labels, colors, etc.
# Have some fun with it.

plt.scatter(df[df['species']=='setosa']['sepal length (cm)'],
        df[df['species']=='setosa']['petal length (cm)'],
        color='pink', label='Setosa', s=60, alpha=0.7)

plt.scatter(df[df['species']=='versicolor']['sepal length (cm)'],
        df[df['species']=='versicolor']['petal length (cm)'],
        color='purple', label='Versicolor', s=60, alpha=0.7)

plt.scatter(df[df['species']=='virginica']['sepal length (cm)'],
        df[df['species']=='virginica']['petal length (cm)'],
        color='blue', label='Virginica', s=60, alpha=0.7)

sep_len = df['sepal length (cm)']
pet_len = df['petal length (cm)']

plt.xlabel("Sepal Length")
plt.ylabel("Petal Length")
plt.title("Comparing sepal length to petal length")
plt.legend(title="Species")

plt.show()
```


Comparing sepal length to petal length
Species
Setosa
Versicolor
Virginica
Petal Length
1
2
3
4
5
6
7
Sepal Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0