Data Set-up/Exploration

```
# import necessary packages
import warnings
warnings.filterwarnings('ignore')

from plotnine import *

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split #TTS function
from sklearn import svm #svm model
from sklearn import metrics #scikit-learn metrics module

# load iris data set
iris =
pd.read_csv("https://raw.githubusercontent.com/lannen/iris-1/main/
iris-1.csv")

# look at features
iris.head()
```

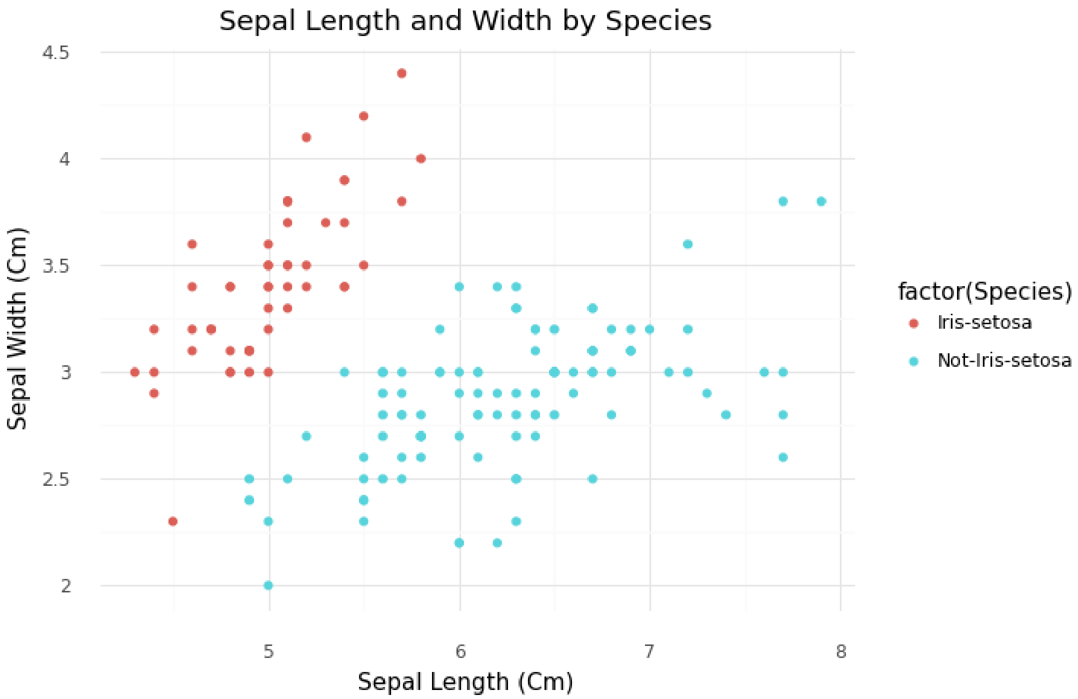
```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
0   1            5.1           3.5            1.4           0.2  Iris-
setosa
1   2            4.9           3.0            1.4           0.2  Iris-
setosa
2   3            4.7           3.2            1.3           0.2  Iris-
setosa
3   4            4.6           3.1            1.5           0.2  Iris-
setosa
4   5            5.0           3.6            1.4           0.2  Iris-
setosa
```

```
# look at size
iris.shape
```

```
(150, 6)
```

Sepal & Petal (length/width) Visualizations

```
# comparing sepal length and width by species
(ggplot(iris, aes(x = "SepalLengthCm", y = "SepalWidthCm", color =
"factor(Species)")) +
 geom_point() + theme_minimal() + xlab("Sepal Length (Cm)") +
ylab("Sepal Width (Cm)") +
 ggtitle("Sepal Length and Width by Species"))
```

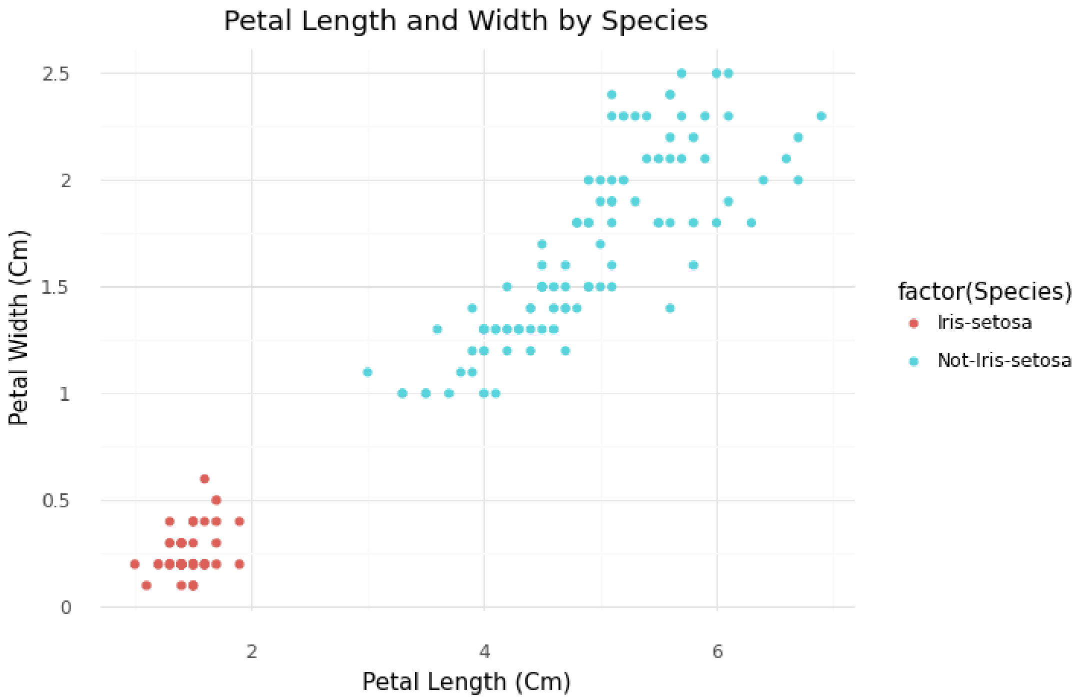


```
<ggplot: (8741076909357)>
```

Graph description

- Iris-Setosa: larger Sepal width (~ 3-4.5cm), shorter Sepal length (~ 4-6cm)
- Not-Iris-Setosa: smaller Sepal width (~ 2-3.5cm), longer Sepal length (~ 5-8cm)

```
# comparing petal length and width by species
(ggplot(iris, aes(x = "PetalLengthCm", y = "PetalWidthCm", color =
"factor(Species)")) +
 geom_point() + theme_minimal() + xlab("Petal Length (Cm)") +
ylab("Petal Width (Cm)") +
 ggtitle("Petal Length and Width by Species"))
```



```
<ggplot: (8741076811881)>
```

Graph description

- Iris-Setosa: smaller Petal width (~ 0-0.5cm), shorter Petal length (~ 0.5-2cm)
- Not-Iris-Setosa: larger Petal width (~ 1-2.5cm), longer Petal length (~ 3-7cm)

Data Transformation

```
# make into a dataframe
iris_df = pd.DataFrame(iris)

# change species feature to a dummy variable for analysis
# target - Iris-setosa OR Not-Iris-setosa
iris_df.Species = iris_df.Species.eq('Iris-setosa').mul(1)

# check if species is a dummy variable
iris_df.head()
```

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  
Species
0   1            5.1           3.5            1.4           0.2  
1
1   2            4.9           3.0            1.4           0.2  
1
2   3            4.7           3.2            1.3           0.2  
1
3   4            4.6           3.1            1.5           0.2  
1
```

```
4   5             5.0           3.6            1.4           0.2
1
```

Train, Test, Split & Support Vector Machines

```
# split dataset into 70% training and 30% test
X_train, X_test, y_train, y_test = train_test_split(iris_df,
iris.Species, test_size=0.3,random_state=109)

# create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel

# train/fit the model using the training set
clf.fit(X_train, y_train)

# predict the response for test set
y_pred = clf.predict(X_test)
```

Accuracy, Precision, Recall

```
# model accuracy: how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

# model precision: what percentage of positive tuples are labeled as
such?
print("Precision:",metrics.precision_score(y_test, y_pred))

# model recall: what percentage of positive tuples are labeled as
such?
print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
```