Data Analysis with python

December 20, 2024

```
[5]: import pandas as pd
```

```
[12]: # using the iris dataset
      from sklearn.datasets import load_iris
```

```
[52]: iris = load_iris()
      df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
      df['species'] = iris.target # this is the target variable
```

```
[16]: df.head()
```

```
[16]:    sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
      0                5.1               3.5                1.4               0.2
      1                4.9               3.0                1.4               0.2
      2                4.7               3.2                1.3               0.2
      3                4.6               3.1                1.5               0.2
      4                5.0               3.6                1.4               0.2

         species
      0        0
      1        0
      2        0
      3        0
      4        0
```

```
[18]: df.info() # summary of the DataFrame
      df.isnull().sum() # counts number of missing values in each column
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   species            150 non-null    int64
```

```
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

```
[18]: sepal length (cm)    0
      sepal width (cm)     0
      petal length (cm)    0
      petal width (cm)     0
      species              0
      dtype: int64
```

```
[20]: # drop any rows with missing values
      df.dropna(inplace=True)
```

```
[22]: # basic statistics - mean, median , standard deviation etc
      df.describe()
```

```
[22]:        sepal length (cm)  sepal width (cm)  petal length (cm)  \
      count         150.000000        150.000000         150.000000
      mean            5.843333          3.057333           3.758000
      std             0.828066          0.435866           1.765298
      min             4.300000          2.000000           1.000000
      25%             5.100000          2.800000           1.600000
      50%             5.800000          3.000000           4.350000
      75%             6.400000          3.300000           5.100000
      max             7.900000          4.400000           6.900000

             petal width (cm)     species
      count        150.000000  150.000000
      mean           1.199333    1.000000
      std            0.762238    0.819232
      min            0.100000    0.000000
      25%            0.300000    0.000000
      50%            1.300000    1.000000
      75%            1.800000    2.000000
      max            2.500000    2.000000
```

```
[24]: # group by species and compute mean
      group_means = df.groupby('species').mean()
      group_means
```

```
[24]:          sepal length (cm)  sepal width (cm)  petal length (cm)  \
      species
      0                    5.006             3.428              1.462
      1                    5.936             2.770              4.260
      2                    6.588             2.974              5.552

               petal width (cm)
```

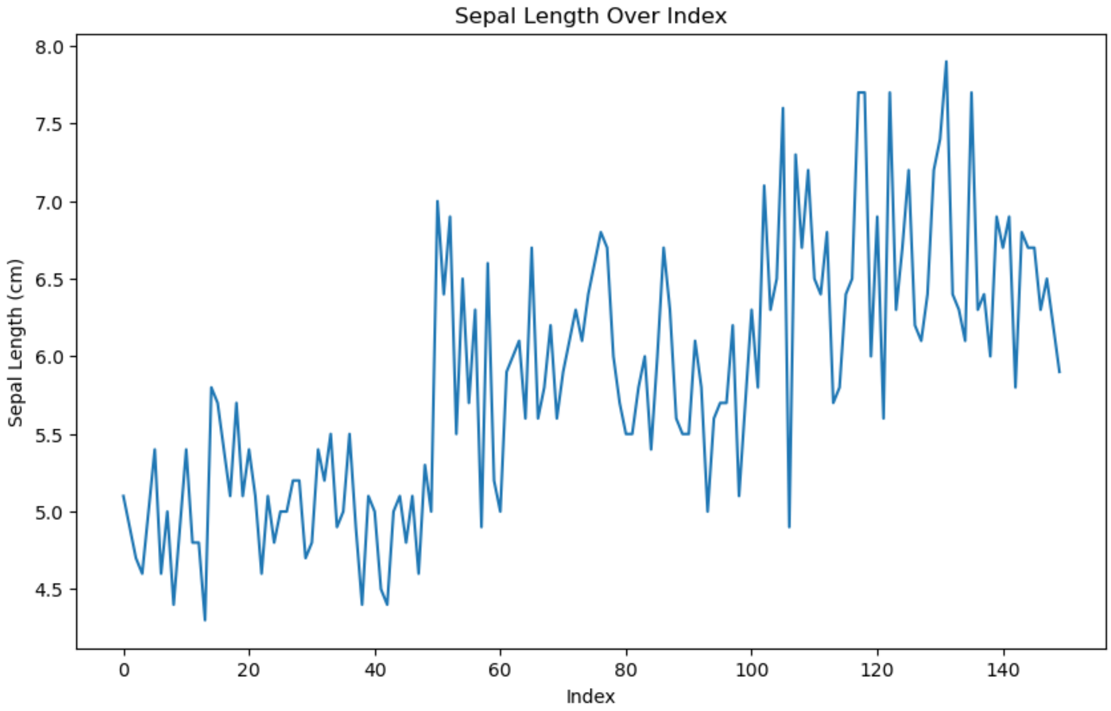
```
species
0                   0.246
1                   1.326
2                   2.026
```

[42]:

```
# Data Visualization
import matplotlib.pyplot as plt # for creating plots
import seaborn as sns  # for more advanced and attractive data visualizations
import numpy as np  # for numerical operations
```

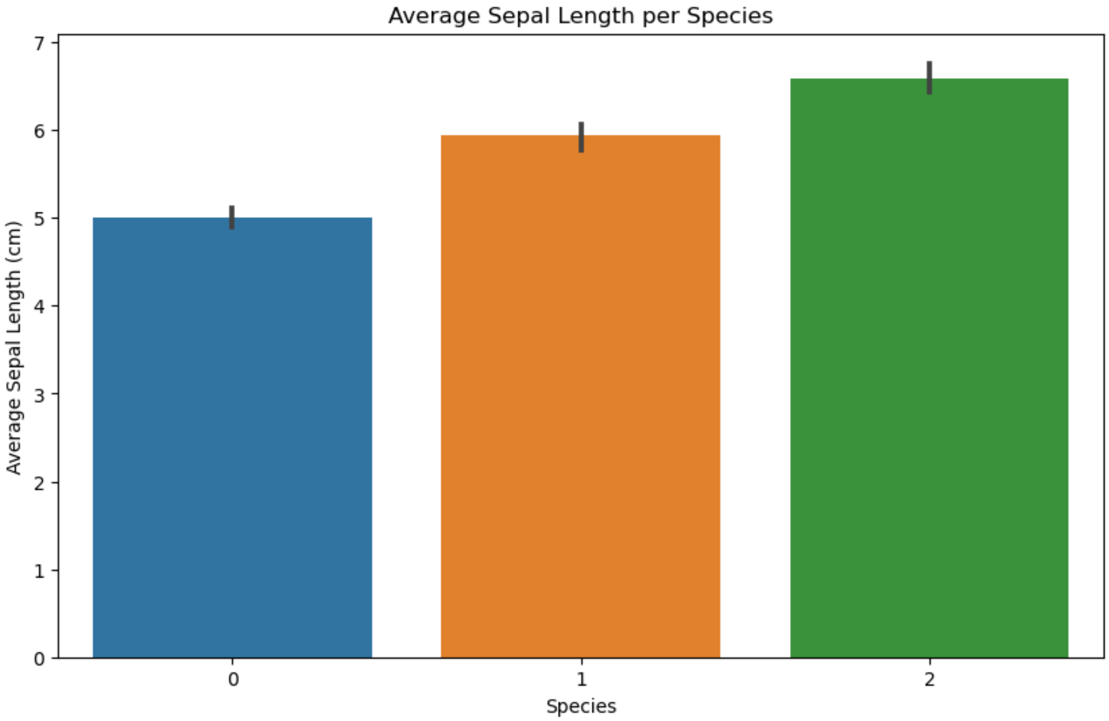
[44]:

```
# Line chart
plt.figure(figsize=(10, 6))
plt.plot(df.index, df['sepal length (cm)'])
# Plotting 'sepal length()' against index
plt.title('Sepal Length Over Index')
plt.xlabel('Index')
plt.ylabel('Sepal Length (cm)')
plt.show()
```



[46]:

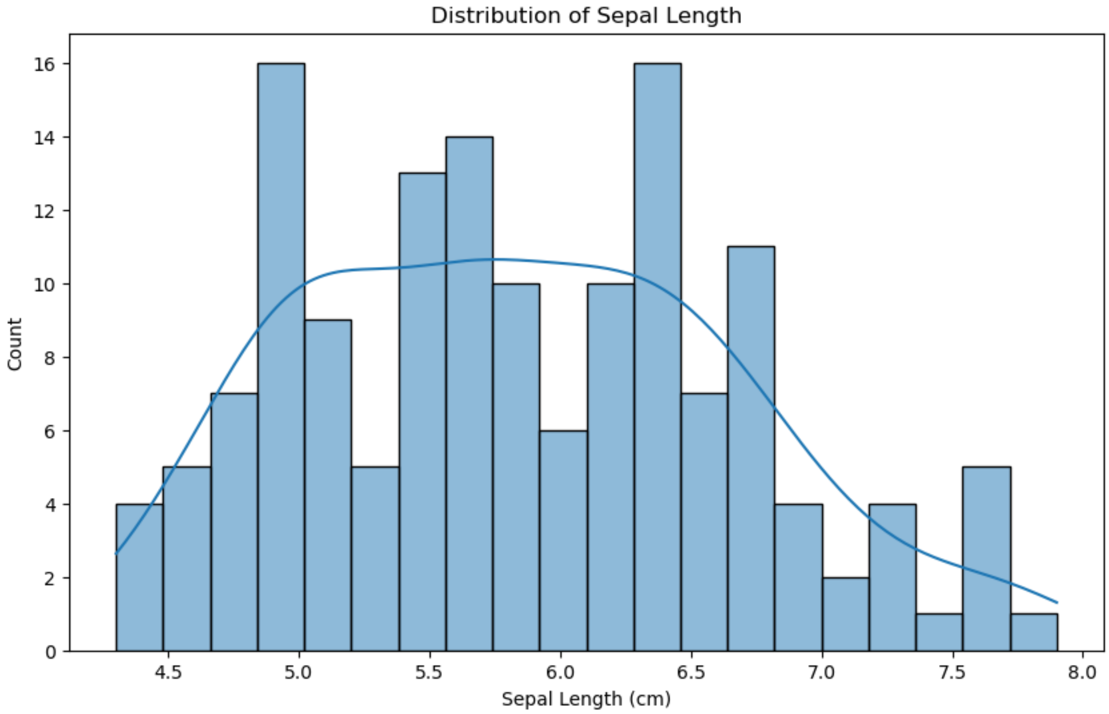
```
# Bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x=df['species'], y=df['sepal length (cm)'], estimator=np.mean)
plt.title('Average Sepal Length per Species')
plt.xlabel('Species')
plt.ylabel('Average Sepal Length (cm)')
plt.show()
```



[48]:

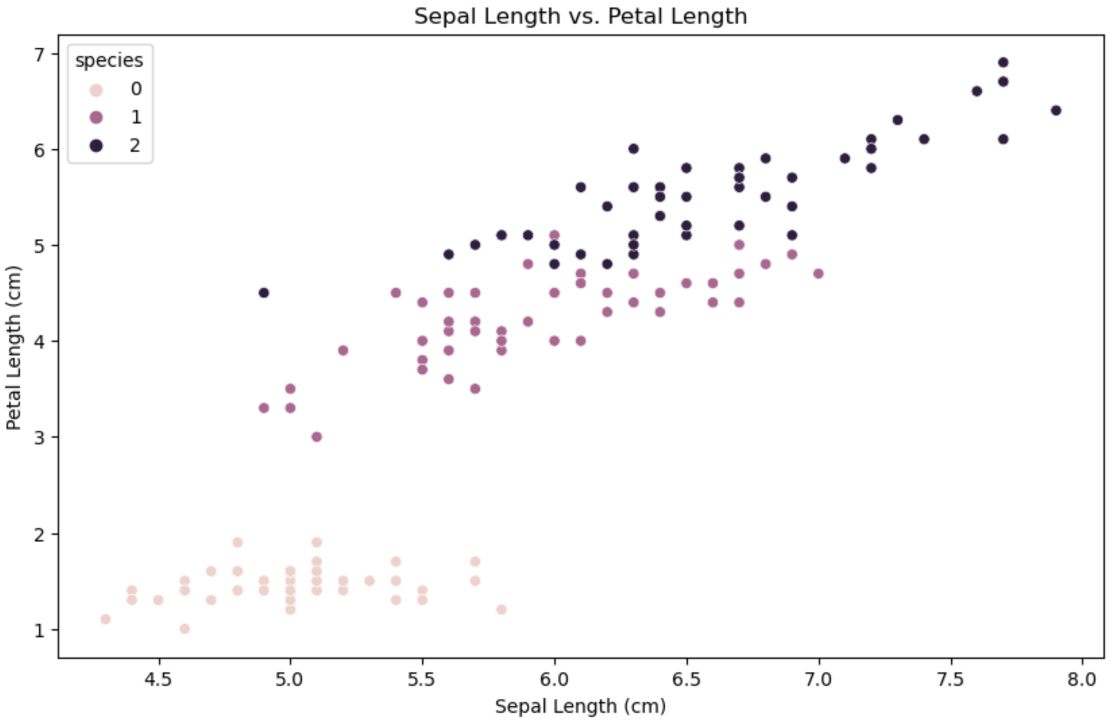
```
# Histogram
plt.figure(figsize=(10, 6))
sns.histplot(df['sepal length (cm)'], bins=20, kde=True)
# Histogram with a kernel density estimate (kde) overlay
plt.title('Distribution of Sepal Length')
plt.xlabel('Sepal Length (cm)')
plt.show()
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is
deprecated and will be removed in a future version. Convert inf values to NaN
before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



[50]:

```
# Scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['sepal length (cm)'], y=df['petal length (cm)'],␣
 ↪hue=df['species'])
# Scatter Plot with different colors for each species
plt.title('Sepal Length vs. Petal Length')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Petal Length (cm)')
plt.show()
```


Sepal Length vs. Petal Length
species
0
1
2
Petal Length (cm)
1
2
3
4
5
6
7
Sepal Length (cm)
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

```
[ ]:
```