palmer_penguins

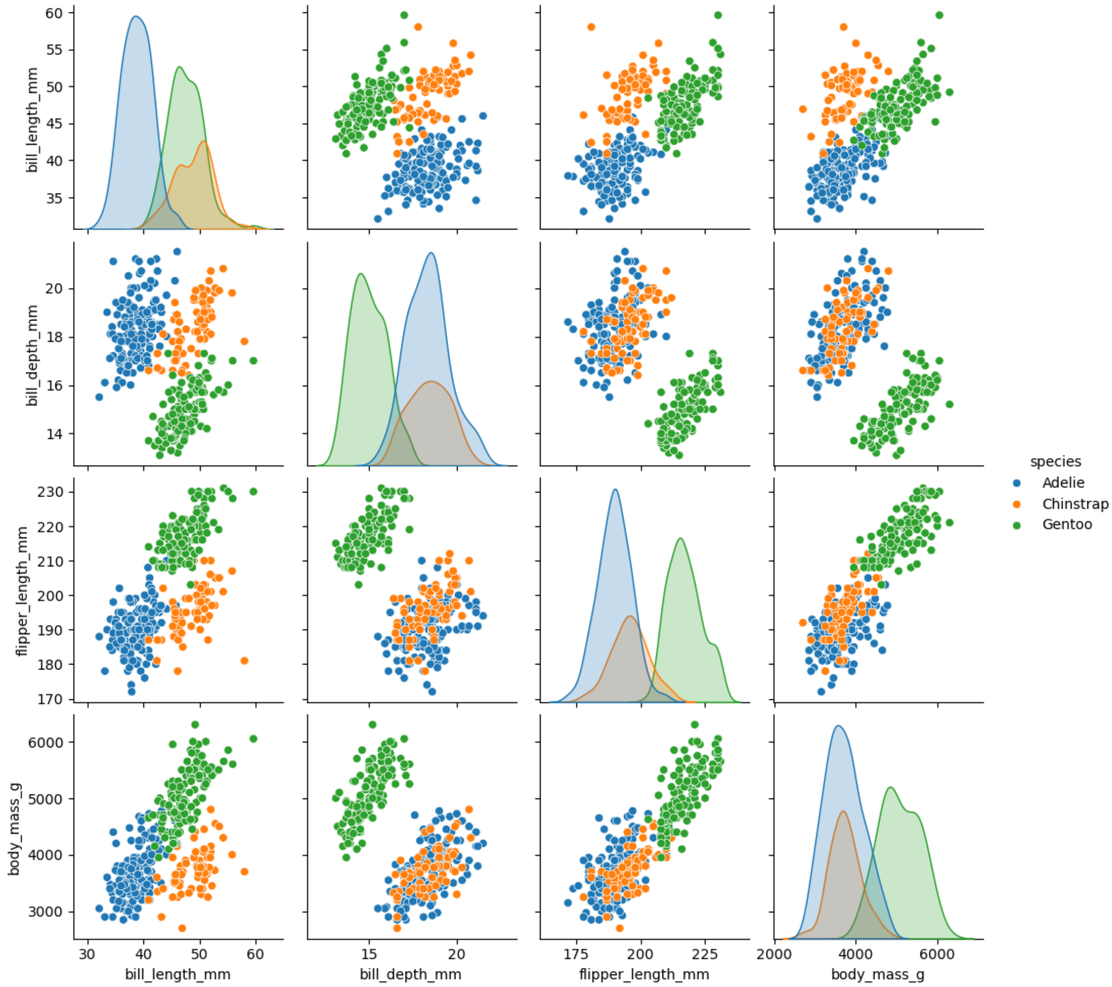
April 4, 2025

[1]:
```
#Import all packages to be used
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn as sk
```

[2]:
```
penguinData = sns.load_dataset('penguins')
```

[3]:
```
#Pairplot to get an overview of how each species compares
sns.pairplot(penguinData, hue='species')
```

[3]:
```
<seaborn.axisgrid.PairGrid at 0x25776eafa10>
```



0.1 The Problem

We have plenty of data about each species, but we are missing values for some instances, as seen here.

[4]:
```
penguinData[penguinData.isna().any(axis=1)]
```

[4]:
```
    species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
3    Adelie  Torgersen             NaN            NaN                NaN
8    Adelie  Torgersen            34.1           18.1              193.0
9    Adelie  Torgersen            42.0           20.2              190.0
10   Adelie  Torgersen            37.8           17.1              186.0
11   Adelie  Torgersen            37.8           17.3              180.0
47   Adelie      Dream            37.5           18.9              179.0
246  Gentoo     Biscoe            44.5           14.3              216.0
286  Gentoo     Biscoe            46.2           14.4              214.0
324  Gentoo     Biscoe            47.3           13.8              216.0
```

```
336  Gentoo     Biscoe            44.5           15.7              217.0
339  Gentoo     Biscoe             NaN            NaN                NaN

     body_mass_g  sex
3            NaN  NaN
8         3475.0  NaN
9         4250.0  NaN
10        3300.0  NaN
11        3700.0  NaN
47        2975.0  NaN
246       4100.0  NaN
286       4650.0  NaN
324       4725.0  NaN
336       4875.0  NaN
339          NaN  NaN
```

The observations indexed 3 and 339 are missing all values besides the species and island, so we will ignore them for the purposes of our analysis.

[5]:
```
penguinData = penguinData.drop(axis=1, index=[3,339])
```

[6]:
```
penguinData[penguinData.isna().any(axis=1)]
```

[6]:
```
    species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
8    Adelie  Torgersen            34.1           18.1              193.0
9    Adelie  Torgersen            42.0           20.2              190.0
10   Adelie  Torgersen            37.8           17.1              186.0
11   Adelie  Torgersen            37.8           17.3              180.0
47   Adelie      Dream            37.5           18.9              179.0
246  Gentoo     Biscoe            44.5           14.3              216.0
286  Gentoo     Biscoe            46.2           14.4              214.0
324  Gentoo     Biscoe            47.3           13.8              216.0
336  Gentoo     Biscoe            44.5           15.7              217.0

     body_mass_g  sex
8         3475.0  NaN
9         4250.0  NaN
10        3300.0  NaN
11        3700.0  NaN
47        2975.0  NaN
246       4100.0  NaN
286       4650.0  NaN
324       4725.0  NaN
336       4875.0  NaN
```

We can see here that the only other missing values are sex. We can use some classification algorithims to try and predict the sex of these penguins. We will explore a few methods.

0.2 KNeighbors

[7]:
```
# Initialize the Model
from sklearn.neighbors import KNeighborsClassifier
knm = KNeighborsClassifier(n_neighbors = 5)
```

[8]:
```
X = penguinData.dropna()[['bill_length_mm', 'bill_depth_mm',␣
↪'flipper_length_mm', 'body_mass_g']].values
y = penguinData.dropna()['sex'].values

knm = knm.fit(X, np.ravel(y))
```

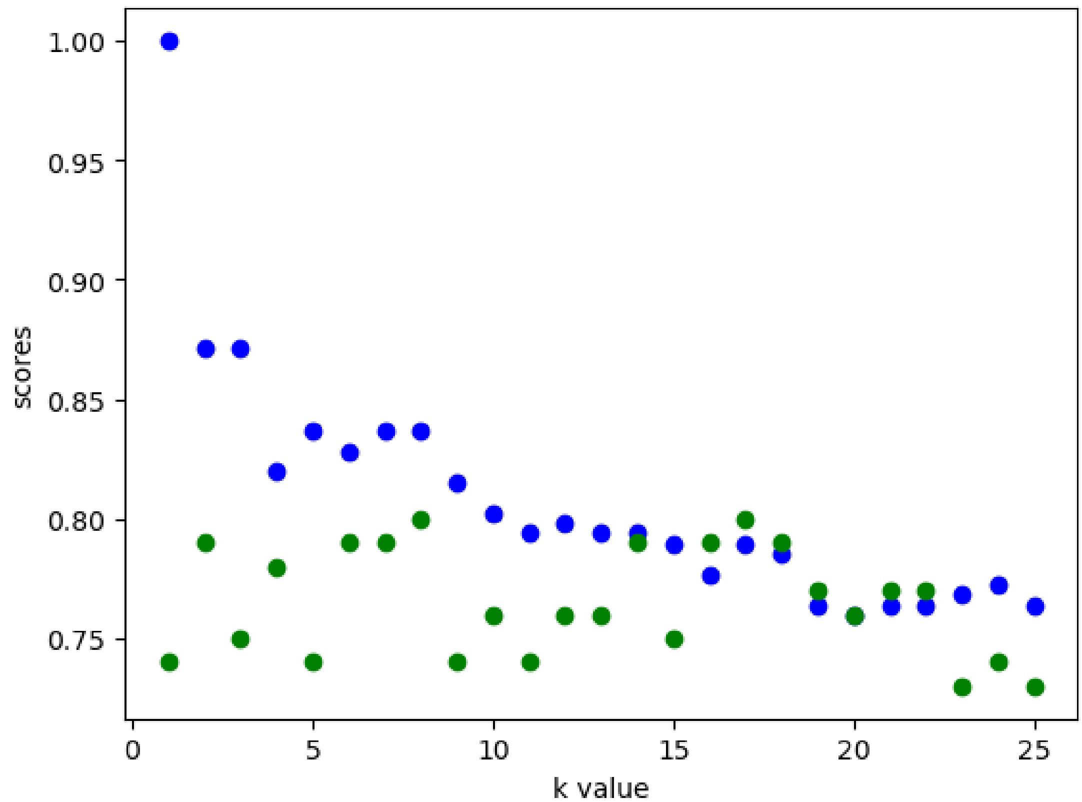
[9]:
```
knm.predict([penguinData.iloc[8, 2:6]])[0]
penguinData.iloc[8,6]
```

[9]: nan

[10]:
```
from sklearn.model_selection import train_test_split
```

[11]:
```
K = []
training = []
test = []
scores = {}

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=0)

for k in range(1,26):
    knm = KNeighborsClassifier(n_neighbors = k)
    knm.fit(X_train, y_train)

    training_score = knm.score(X_train, y_train)
    test_score = knm.score(X_test, y_test)

    K.append(k)
    training.append(training_score)
    test.append(test_score)
    scores[k] = [training_score, test_score]
```

[12]:
```
plt.scatter(K, training, color ='b')
plt.scatter(K, test, color ='g')
plt.xlabel('k value')
plt.ylabel('scores')
plt.show()
```



[13]:
```
knm = KNeighborsClassifier(n_neighbors = 2)
knm.fit(X, y)

predictSex = penguinData[penguinData.isna().any(axis=1)]

for i in range(predictSex.shape[0]):
    print(
        knm.predict(
            [predictSex.iloc[i, 2:6]]
        )
    )
```

```
['Female']
['Male']
['Female']
['Female']
['Female']
['Male']
['Female']
['Female']
['Female']
```

So using a very simple model we were able to provide preedictions for the sex of the penguins who had a missing entry.