EDA on the Iris dataset

Data Scientist : Simon Pierre Charbel

In [1]:
```
# Import libraries and packages
import pandas as pd
from sklearn.datasets import load_iris

import seaborn as sns
import matplotlib.pyplot as plt
```

1. Loading and pre-processing the data

In [2]:
```
# Load the Iris dataset
iris = load_iris()
print(f"Type of Scikit-Learn object returned : {type(iris)}\n")

# Create a pandas DataFrame
iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

# Add the target column to the DataFrame
iris_df['target'] = iris.target
# Add the target names as a new column
iris_df['species'] = iris.target_names[iris.target]

# Print shape and the first few rows of the dataframe
print(f"DataFrame Shape: {iris_df.shape[0]} rows and {iris_df.shape[1]}
iris_df.head()
```

```
Type of Scikit-Learn object returned : <class 'sklearn.utils._bunch.Bunc
h'>

DataFrame Shape: 150 rows and 6 columns (shape).
```

Out[2]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	species
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa

In [3]:
```
# Checking for missing values
iris_df.isnull().sum()
```

Out[3]:
```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
target               0
species              0
dtype: int64
```

In [4]:
```
# Check for duplicate rows
duplicates = iris_df.duplicated(keep=False)
# Display all rows that are duplicates
duplicate_rows = iris_df[duplicates]
#print( f" Duplicate rows : \n {duplicate_rows}")
print( f" Duplicate rows count : {duplicate_rows.shape[0]}")

# Drop the Duplicates
iris_df.drop_duplicates(inplace=True)
```

```
 Duplicate rows count : 2
```

In [5]:
```
 # Print the shape of the DataFrame
print(f"New DataFrame Shape: {iris_df.shape[0]} rows and {iris_df.shape

# Print the columne types
print(f"\nColumns types: \n")
print(iris_df.info())

# Print a statistical summary of the dataset
print(f"\n\n A statistical summary of the dataset:")
iris_df.describe()
```

```
New DataFrame Shape: 149 rows and 6 columns (shape).


Columns types:

<class 'pandas.core.frame.DataFrame'>
Int64Index: 149 entries, 0 to 149
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  149 non-null    float64
 1   sepal width (cm)   149 non-null    float64
 2   petal length (cm)  149 non-null    float64
 3   petal width (cm)   149 non-null    float64
 4   target             149 non-null    int32  
 5   species            149 non-null    object 
dtypes: float64(4), int32(1), object(1)
memory usage: 7.6+ KB
None


 A statistical summary of the dataset:
```

Out[5]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	149.000000	149.000000	149.000000	149.000000	149.000000
mean	5.843624	3.059732	3.748993	1.194631	0.993289
std	0.830851	0.436342	1.767791	0.762622	0.817847
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.300000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

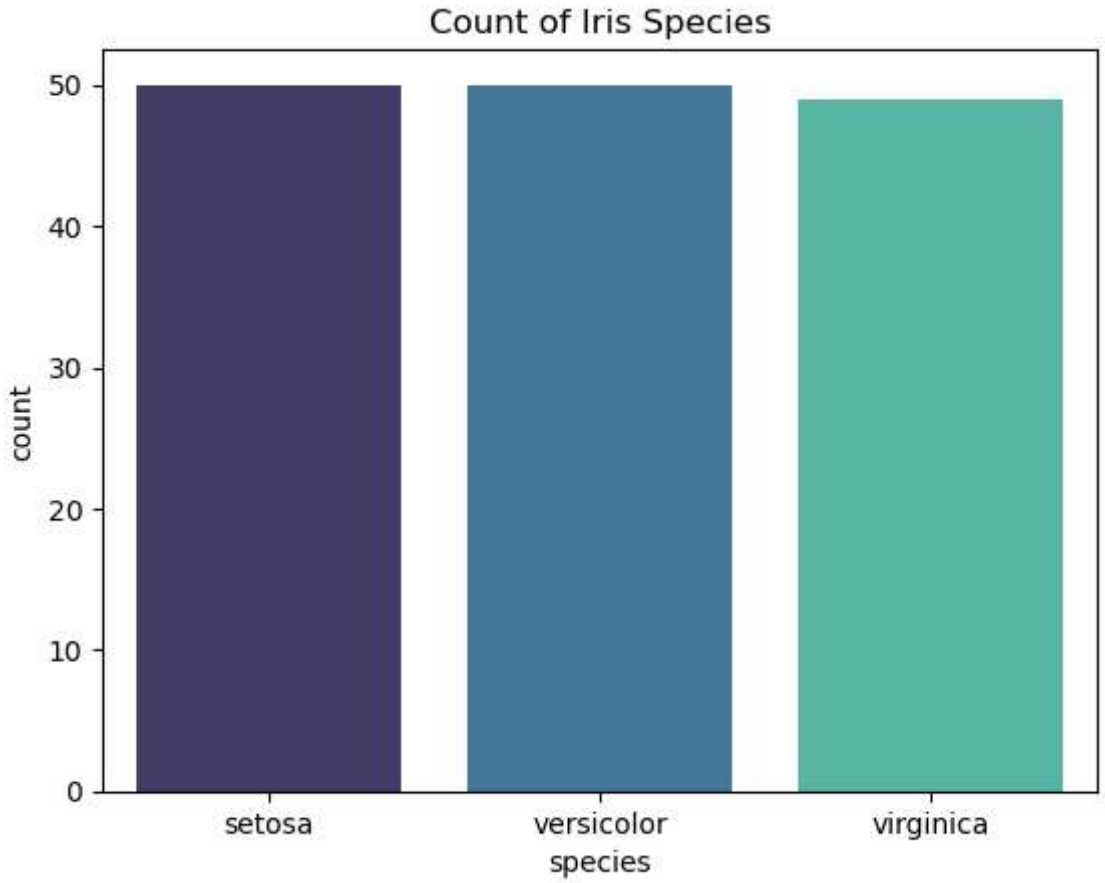
In [6]:
```
# Check how many species we have and their counts
iris_df.value_counts("species")
```

Out[6]:
```
species
setosa        50
versicolor    50
virginica     49
dtype: int64
```

2. Data Visualization

In [7]:
```
# Visualizing the target column (Countplot)
sns.countplot(data=iris_df, x='species', palette='mako')
plt.title("Count of Iris Species", fontsize=12)
plt.figure(figsize=(4, 4))
```

Out[7]:
```
<Figure size 400x400 with 0 Axes>
```

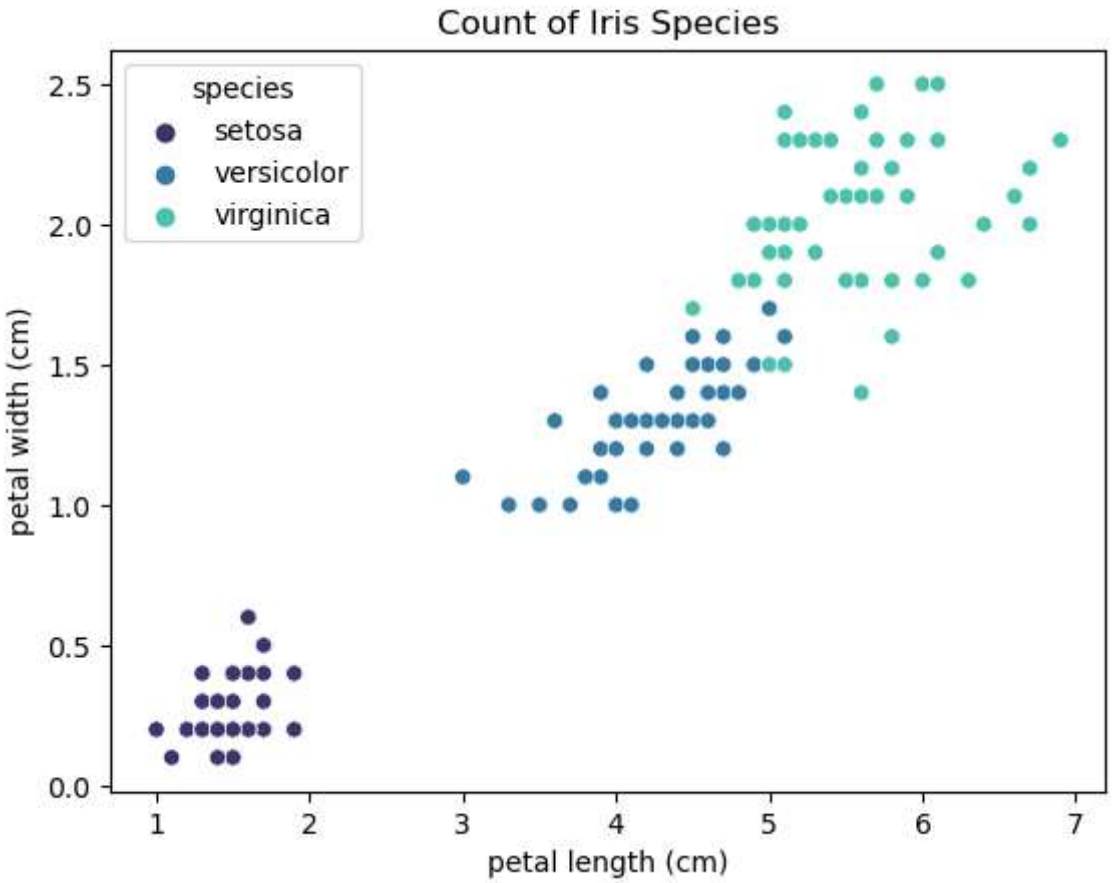



```
<Figure size 400x400 with 0 Axes>
```

- Explore the correlations between sepal length and sepal width, as well as between petal length and petal width.

In [8]:
```
# Petal
sns.scatterplot(data=iris_df, x='petal length (cm)',y='petal width (cm)
                hue='species', palette='mako')
plt.title("Count of Iris Species", fontsize=12)
plt.figure(figsize=(4, 4))
```

Out[8]: <Figure size 400x400 with 0 Axes>

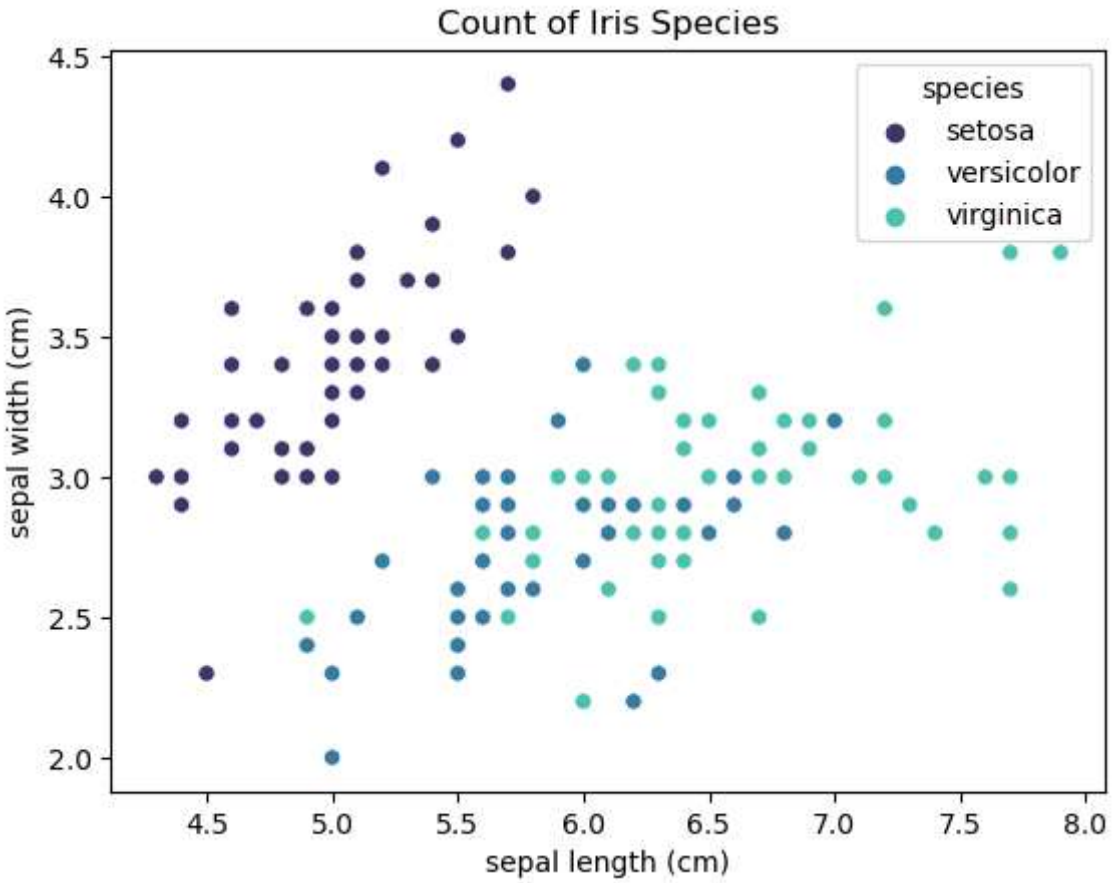


<Figure size 400x400 with 0 Axes>

- The Setosa species exhibits comparatively smaller petal lengths and widths.
- The Versicolor species falls in between the other two species regarding petal length and width.
- The Virginica species boasts the largest petal lengths and widths among them.

In [9]:
```
# Septal
sns.scatterplot(data=iris_df, x='sepal length (cm)',y='sepal width (cm)
                hue='species', palette='mako')
plt.title("Count of Iris Species", fontsize=12)
plt.figure(figsize=(4, 4))
```

Out[9]: <Figure size 400x400 with 0 Axes>



<Figure size 400x400 with 0 Axes>

- The Setosa species showcases smaller sepal lengths but wider sepal widths.
- The Versicolor species occupies an intermediate position between the other two species concerning sepal length and width.
- The Virginica species boasts longer sepal lengths but narrower sepal widths.

In [158]:
```
file_path = r'C:\Users\simon\AppData\Roaming\JetBrains\PyCharmCE2022.2\

# Save the DataFrame to a CSV file with UTF-8 encoding
iris_df.to_csv(file_path, encoding='utf-8', index=False)
```