

```
In [1]: greetings ="Assalam-o-Alaikum!"
print(greetings)
```

Assalam-o-Alaikum!

## Import Libraries

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

## Import DataSet

```
In [3]: df = pd.read_csv("IRIS.csv")
df.head(5)
```

```
Out[3]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  -
 0   sepal_length    150 non-null   float64
 1   sepal_width     150 non-null   float64
 2   petal_length    150 non-null   float64
 3   petal_width     150 non-null   float64
 4   species         150 non-null   object  
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

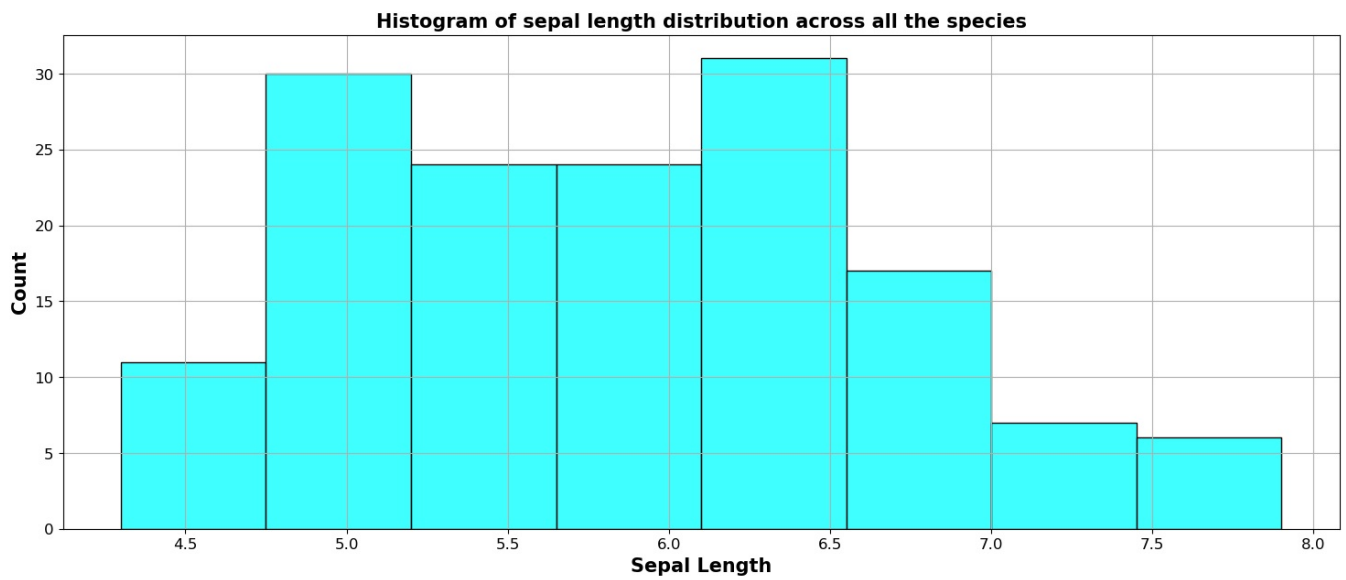
```
In [5]: df.describe()
```

```
Out[5]:
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

1. Create a histogram to visualize the distribution of sepal lengths across all species.

```
In [6]: plt.figure(figsize=(18, 7))
sns.histplot(x ="sepal length", data = df, color = "cyan", bins = 8)
plt.title("Histogram of sepal length distribution across all the species", weight ="bold", size = 15 )
plt.xticks(size =12)
plt.yticks(size =12)
plt.xlabel("Sepal Length", weight ="bold", size =15)
plt.ylabel("Count", weight ="bold", size =15)
plt.grid()
plt.show()
```



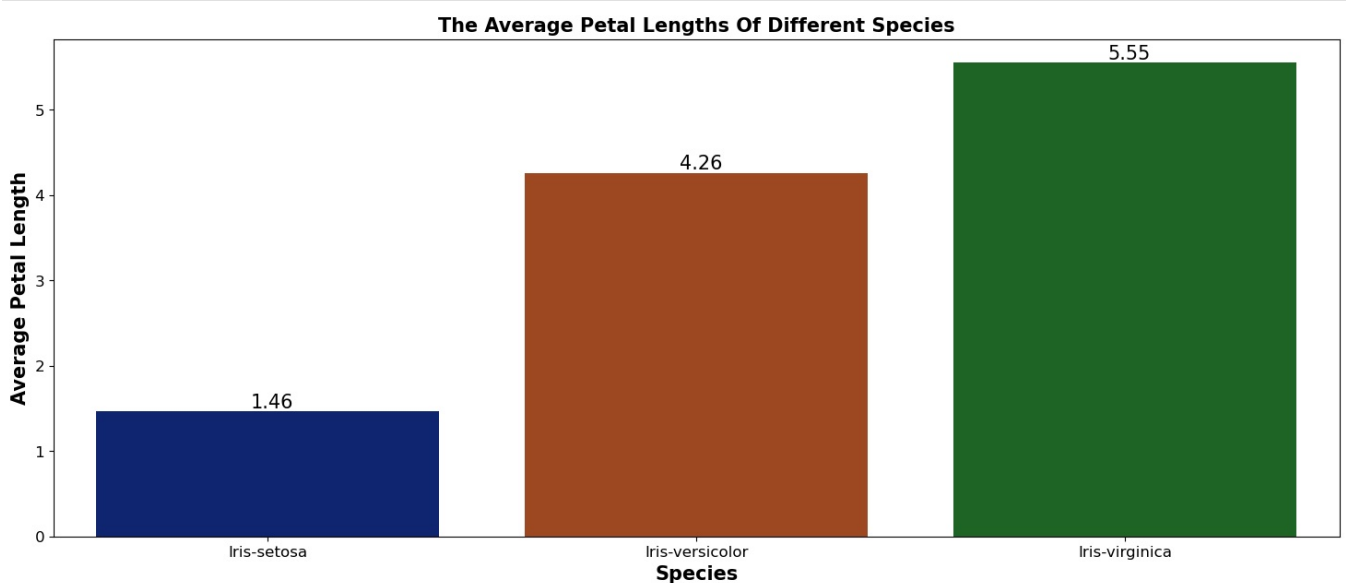
2. Generate a bar chart comparing the average petal lengths of different species.

```
In [7]: Average_petal_length = df.groupby("species")["petal_length"].agg("mean").to_frame().reset_index()
Average_petal_length.columns = ["Species", "Average Petal Length"]
Average_petal_length
```

```
Out[7]:
```

	Species	Average Petal Length
0	Iris-setosa	1.464
1	Iris-versicolor	4.260
2	Iris-virginica	5.552

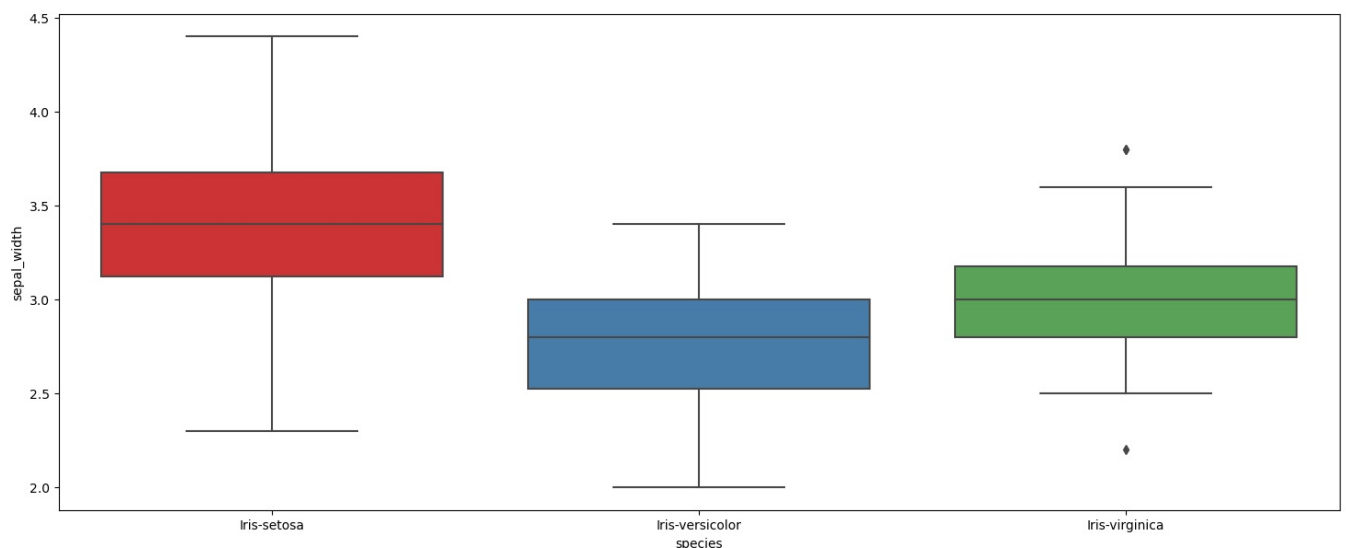
```
In [8]: plt.figure(figsize =(18, 7))
graph = sns.barplot(x = "Species", y = "Average Petal Length", data = Average_petal_length, palette = "dark")
for p in graph.patches:
    graph.annotate('{:.02f}'.format(p.get_height()),
                    (p.get_x()+0.41, p.get_height()),
                    ha='center', va='bottom',color= 'black', size = 15)
plt.title("The Average Petal Lengths Of Different Species", weight ="bold", size =15)
plt.xticks(size =12)
plt.yticks(size =12)
plt.xlabel("Species", weight ="bold", size = 15)
plt.ylabel("Average Petal Length", weight ="bold", size =15)
plt.show()
```



3. Create a box plot to compare the sepal widths of each species.

```
In [9]: plt.figure(figsize =(18, 7))
sns.boxplot(x ="species", y = "sepal_width", data = df, palette ="Set1")
```

```
Out[9]: <Axes: xlabel='species', ylabel='sepal_width'>
```



In [10]: `df.head(1)`

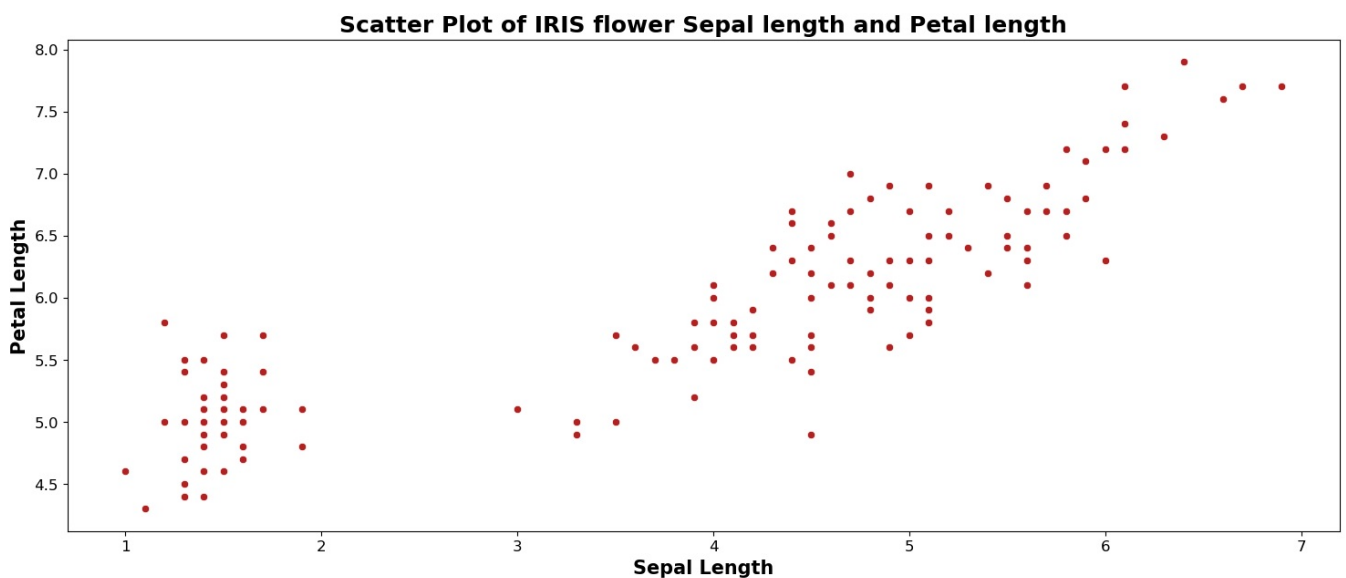
Out[10]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa

4. Visualize the relationship between sepal length and petal length using a scatter plot.

In [19]:

```
plt.figure(figsize=(18, 7))
sns.scatterplot(y="sepal_length", x="petal_length", data=df, color="firebrick")
plt.title("Scatter Plot of IRIS flower Sepal Length and Petal length", weight="bold", size=18)
plt.xticks(size=12)
plt.yticks(size=12)
plt.xlabel("Sepal Length", weight="bold", size=15)
plt.ylabel("Petal Length", weight="bold", size=15)
plt.show()
```



5. Generate a pie chart to display the distribution of species in the dataset.

In [25]:

```
Distribution = df["species"].value_counts().to_frame().reset_index()
Distribution.columns = ["Species", "Counts"]
Distribution
```

Out[25]:

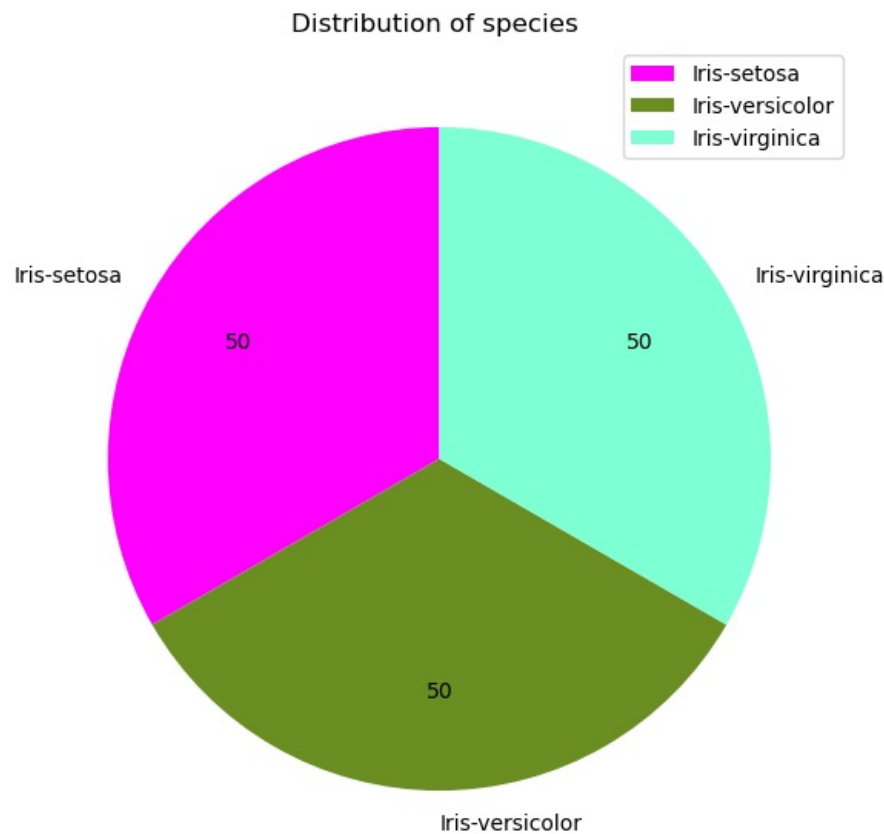
	Species	Counts
0	Iris-setosa	50
1	Iris-versicolor	50
2	Iris-virginica	50

In [29]:

```
plt.figure(figsize=(18, 7))
pie = plt.pie(Distribution['Counts'], labels=Distribution['Species'], startangle=90, colors = ["magenta", "oliv
```

```
# Adding numbers as labels
for i, (count, label) in enumerate(zip(Distribution['Counts'], Distribution['Species'])):
    angle = (pie[0][i].theta2 + pie[0][i].theta1) / 2
    x = pie[0][i].r * 0.7 * np.cos(np.deg2rad(angle))
    y = pie[0][i].r * 0.7 * np.sin(np.deg2rad(angle))
    plt.text(x, y, str(count), ha='center', va='center', color = 'black')

# Adding a title
plt.title('Distribution of species ')
plt.legend()
# Display the chart
plt.show()
```



6. Create a violin plot to compare the sepal lengths of Iris-setosa and Iris-versicolor species.

```
In [36]: Comparing = df[(df["species"] == "Iris-setosa") | (df["species"] == "Iris-versicolor")]
Comparing
```

```
Out[36]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
95	5.7	3.0	4.2	1.2	Iris-versicolor
96	5.7	2.9	4.2	1.3	Iris-versicolor
97	6.2	2.9	4.3	1.3	Iris-versicolor
98	5.1	2.5	3.0	1.1	Iris-versicolor
99	5.7	2.8	4.1	1.3	Iris-versicolor

100 rows × 5 columns

```
In [48]: plt.figure(figsize = (18,7))
sns.violinplot(x = "species", y = "sepal_length", data = Comparing, palette = ["cyan", "violet"], hue = "species")
plt.legend()
plt.title("Violin plot to compare the sepal lengths of Iris-setosa and Iris-versicolor species ", weight = "bold")
plt.xticks(size = 12)
```

```
plt.yticks(size = 12)
plt.xlabel("Species", weight = "bold", size = 15)
plt.ylabel("Sepal Length", weight = "bold", size = 15)
plt.show()
```



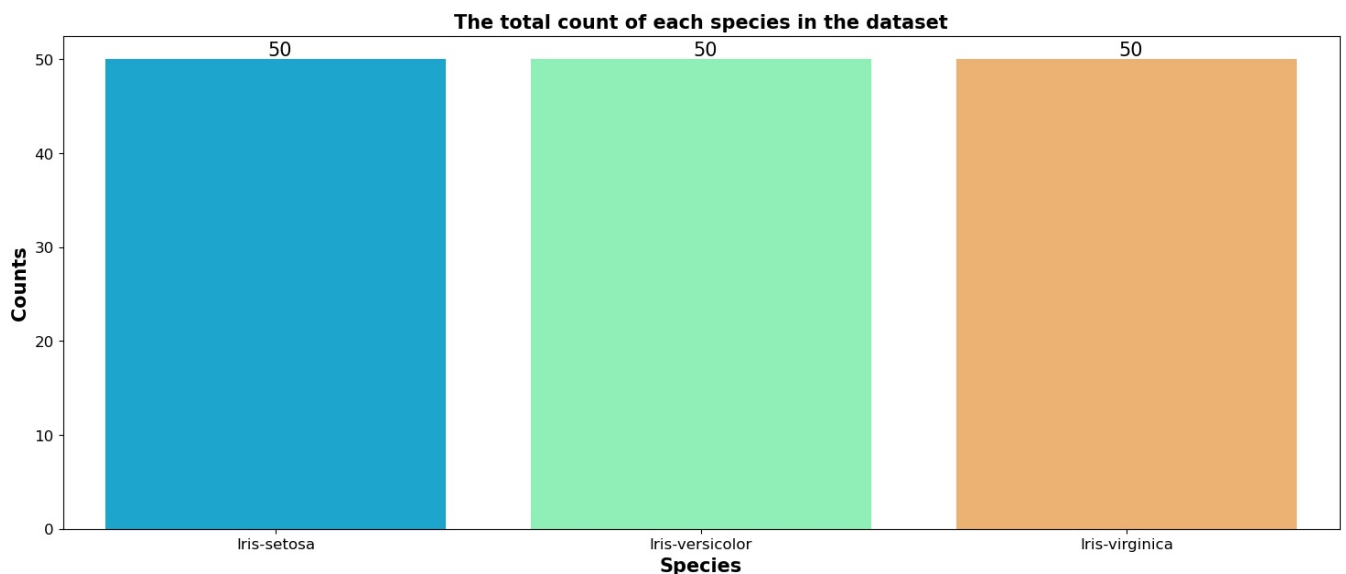
7. Generate a bar chart showing the total count of each species in the dataset.

```
In [50]: total_count = df["species"].value_counts().to_frame().reset_index()
total_count.columns = ["Species", "Counts"]
total_count
```

```
Out[50]:
```

	Species	Counts
0	Iris-setosa	50
1	Iris-versicolor	50
2	Iris-virginica	50

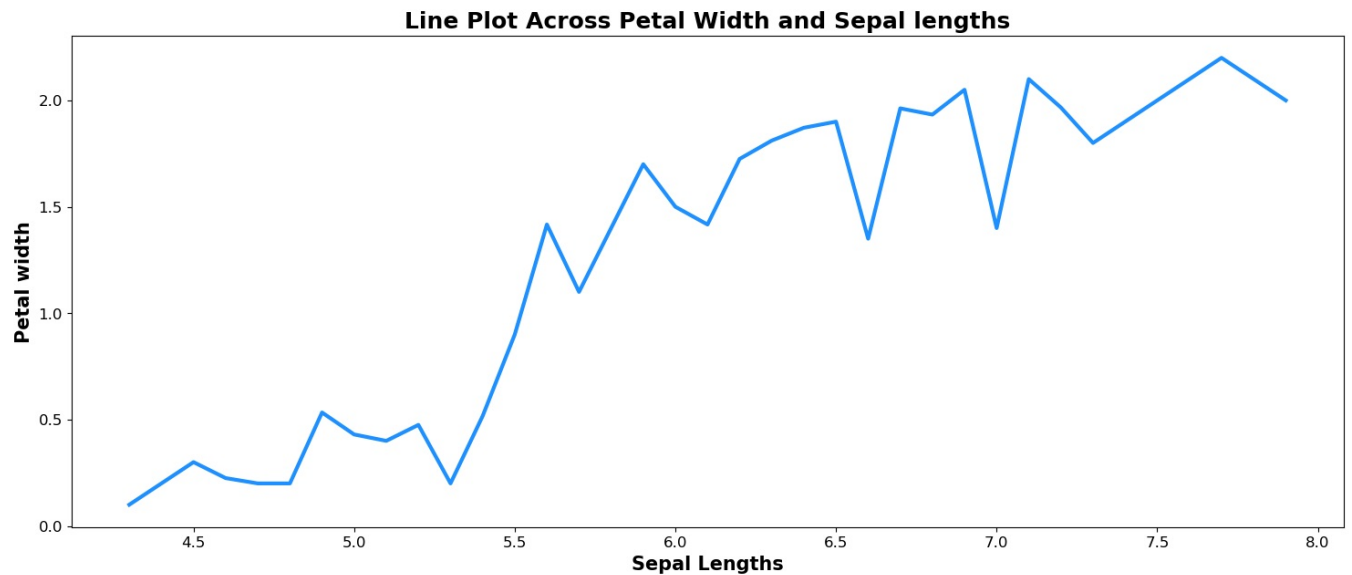
```
In [58]: plt.figure(figsize = (18, 7))
graph = sns.barplot(x = "Species", y = "Counts", data = total_count, palette = "rainbow")
for p in graph.patches:
    graph.annotate('{:.0f}'.format(p.get_height()),
                  (p.get_x()+0.41, p.get_height()),
                  ha='center', va='bottom', color= 'black', size = 15)
plt.title("The total count of each species in the dataset", weight = "bold", size = 15)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Species", weight = "bold", size = 15)
plt.ylabel("Counts", weight = "bold", size = 15)
plt.show()
```



8. Create a line plot to visualize the changes in petal widths across different sepal lengths.

```
In [63]: plt.figure(figsize = (18, 7))
sns.lineplot(x = "sepal_length", y = "petal_width", data = df, color = "dodgerblue", errorbar = None, linewidth = 2)
plt.title("Line Plot Across Petal Width and Sepal lengths", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
```

```
plt.xlabel("Sepal Lengths", weight = "bold", size = 15)
plt.ylabel("Petal width", weight = "bold", size = 15)
plt.show()
```



9. Generate a stacked bar chart to display the proportion of each species based on petal lengths.

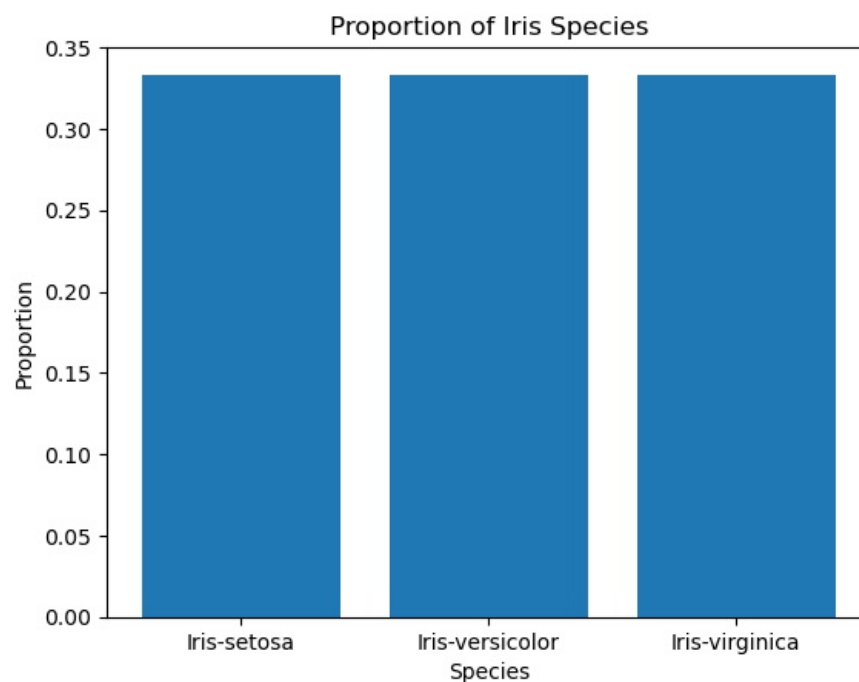
```
In [67]: # Group the data by species and count the occurrences
species_counts = df["species"].value_counts()

# Calculate the proportions
total_count = species_counts.sum()
proportions = species_counts / total_count

# Plotting the stacked bar chart
plt.bar(species_counts.index, proportions)

# Add labels and title
plt.xlabel('Species')
plt.ylabel('Proportion')
plt.title('Proportion of Iris Species')

# Display the plot
plt.show()
```



```
In [69]: # Group the data by species and calculate the mean petal length
mean_petal_length = df.groupby('species')['petal_length'].mean()
```

```

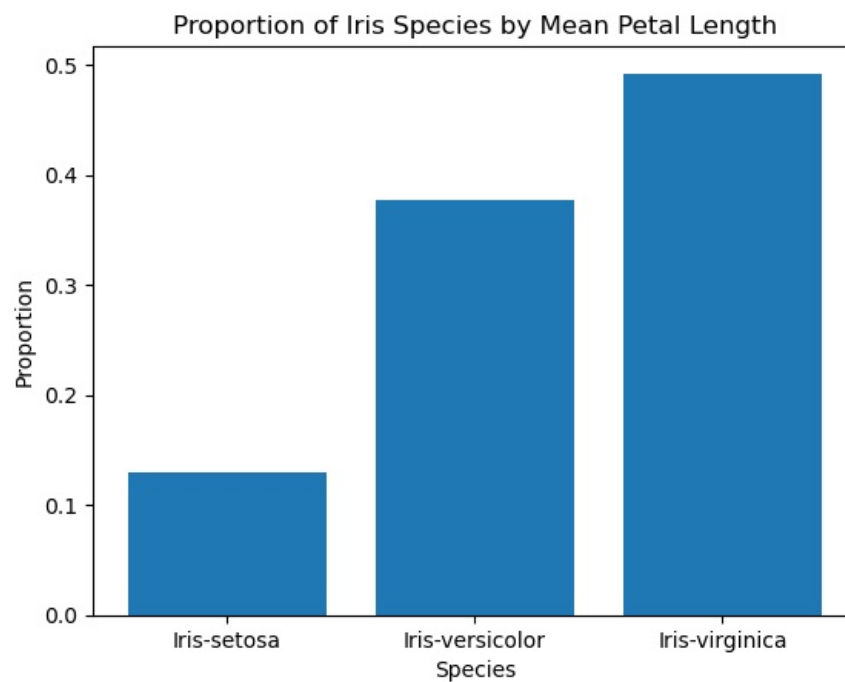
# Calculate the proportions
total_length = mean_petal_length.sum()
proportions = mean_petal_length / total_length

# Plotting the stacked bar chart
plt.bar(mean_petal_length.index, proportions)

# Add labels and title
plt.xlabel('Species')
plt.ylabel('Proportion')
plt.title('Proportion of Iris Species by Mean Petal Length')

# Display the plot
plt.show()

```



In [70]: mean\_petal\_length

```

Out[70]: species
Iris-setosa      1.464
Iris-versicolor  4.260
Iris-virginica   5.552
Name: petal_length, dtype: float64

```

In [71]: mean\_petal\_length.sum()

```

Out[71]: 11.276

```

In [72]: mean\_petal\_length / total\_length

```

Out[72]: species
Iris-setosa      0.129833
Iris-versicolor  0.377794
Iris-virginica   0.492373
Name: petal_length, dtype: float64

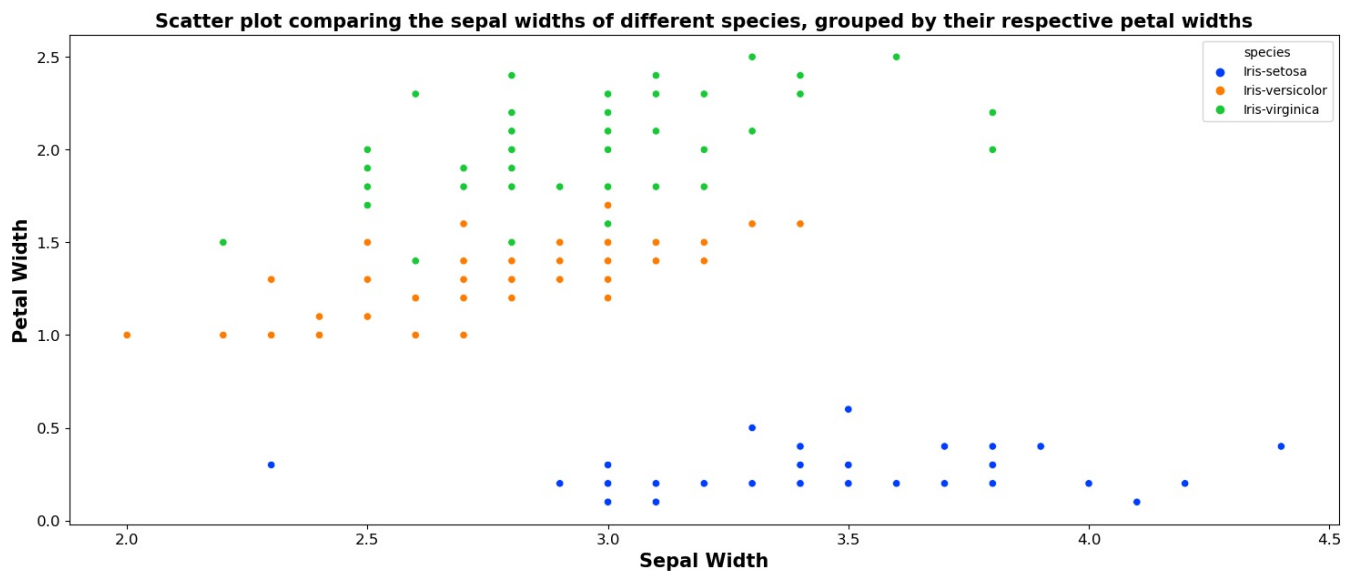
```

**10. Create a Scatter plot comparing the sepal widths of different species, grouped by their respective petal widths.**

```

In [83]: plt.figure(figsize = (18, 7))
sns.scatterplot(x = "sepal_width", y = "petal_width", data = df, palette = "bright", hue = "species")
plt.title("Scatter plot comparing the sepal widths of different species, grouped by their respective petal width")
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Sepal Width", weight = "bold", size = 15)
plt.ylabel("Petal Width", weight = "bold", size = 15)
plt.show()

```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js