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

In [3]:
```
import pandas as pd


url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv"


df = pd.read_csv(url)
df.head()
```

Out[3]:

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

In [4]:
```
df.columns
```

Out[4]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [5]:
```
df.sum().isnull()
```

Out[5]:
```
sepal_length    False
sepal_width     False
petal_length    False
petal_width     False
species         False
dtype: bool
```

In [6]:
```
print(df.head())
```

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

In [7]:
```
print(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
None
```

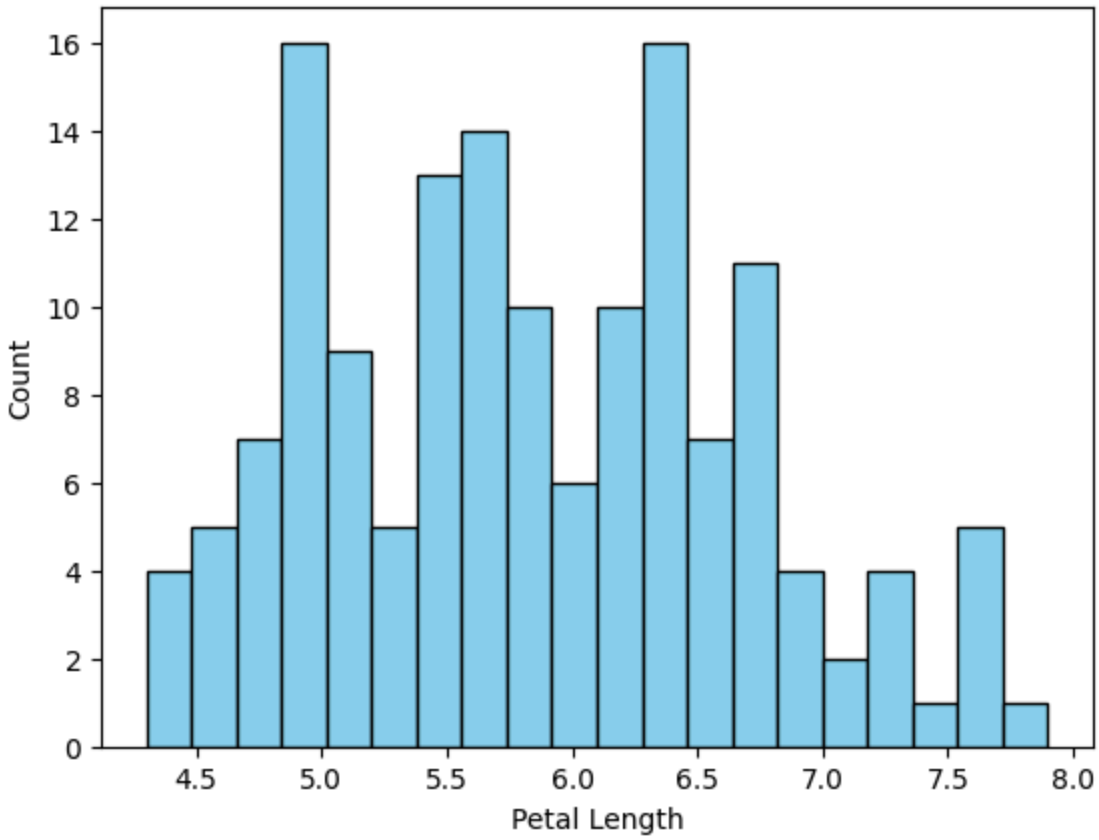
In [8]:
```
print(df.describe())
```

```
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
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
```

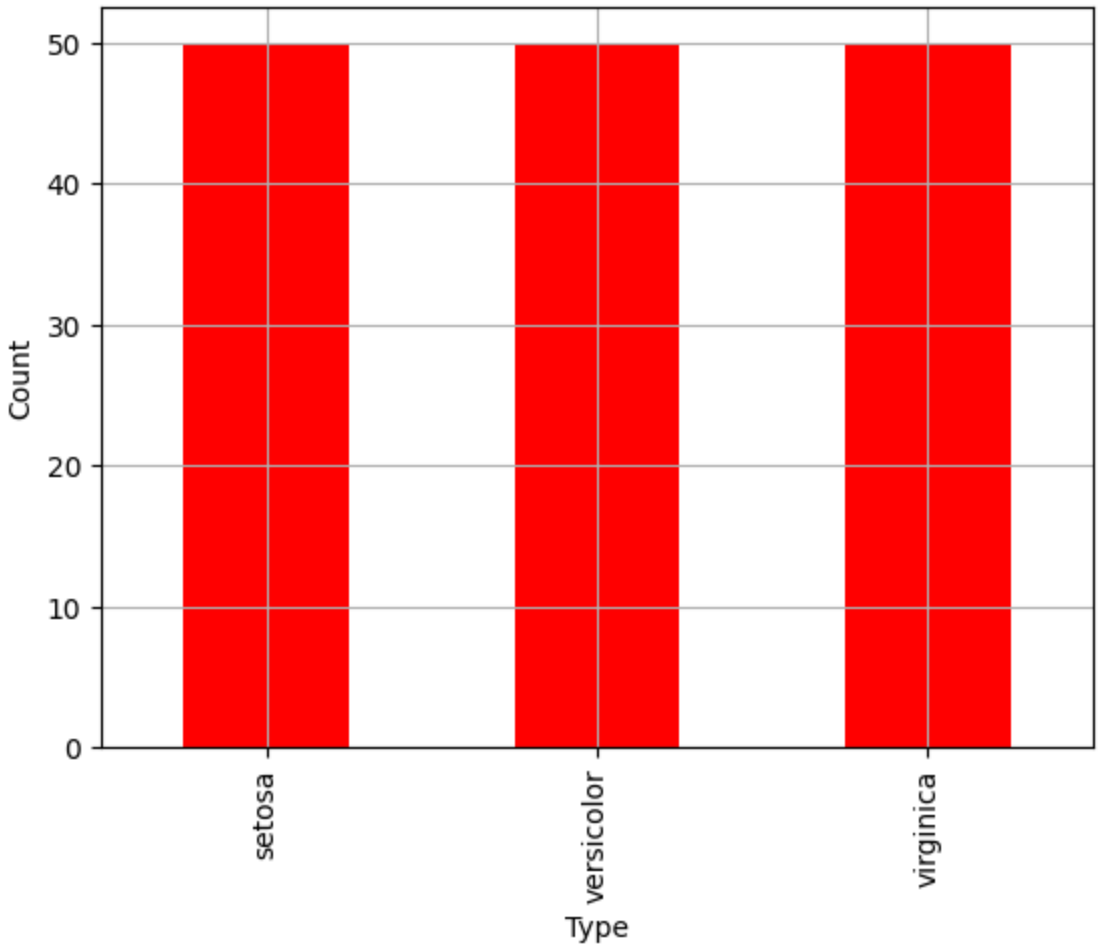
In [9]:
```
# count species counts
print(df['species'].value_counts())
```

```
species
setosa        50
versicolor    50
virginica     50
Name: count, dtype: int64
```

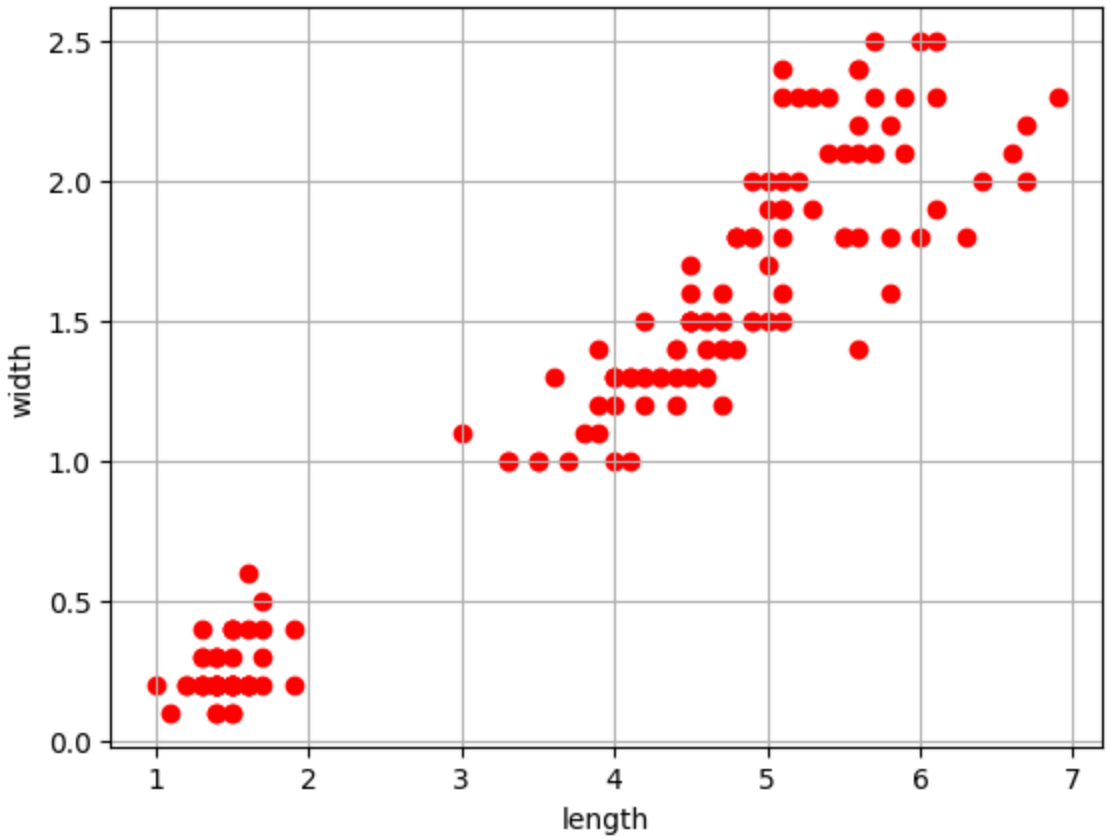
In [10]:
```
plt.hist(df['sepal_length'],bins=20,color='skyblue',edgecolor='black')
plt.xlabel('Petal Length')
plt.ylabel('Count')
plt.show()
```



In [11]:
```
df['species'].value_counts().plot(kind='bar',color='red')

plt.xlabel('Type')
plt.ylabel('Count')
plt.grid(True)
plt.show()
```



In [12]:
```
plt.scatter(df['petal_length'], df['petal_width'], c='red')
plt.xlabel('length')
plt.ylabel('width')
plt.grid(True)
plt.show()
```



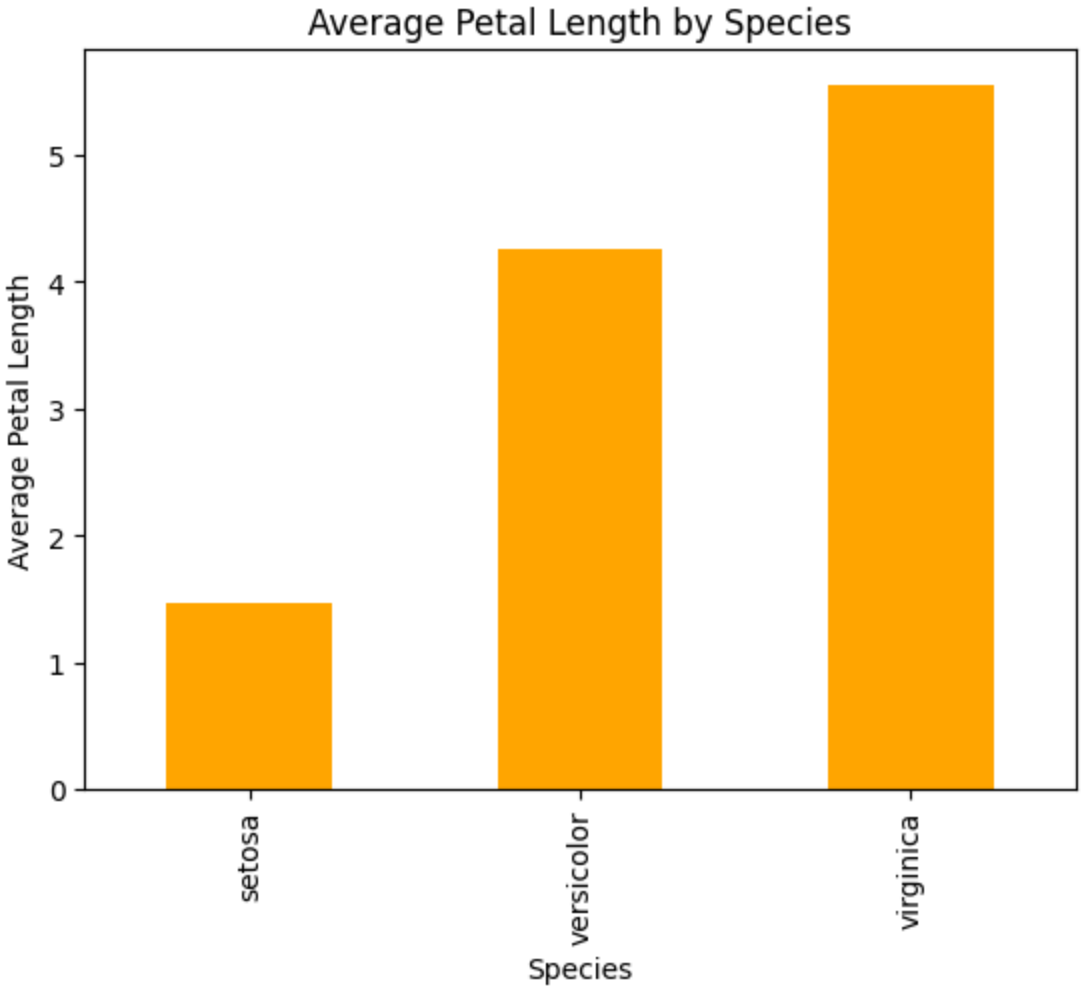
In [13]:
```
# Group by species and calculate mean of numeric columns
species_group=df.groupby('species').mean()
print(species_group)
```

```
            sepal_length  sepal_width  petal_length  petal_width
species
setosa             5.006        3.428         1.462        0.246
versicolor         5.936        2.770         4.260        1.326
virginica          6.588        2.974         5.552        2.026
```

In [14]:
```
species_petal=df.groupby('species')['petal_length'].agg(['min','max'])
print(species_petal)
```

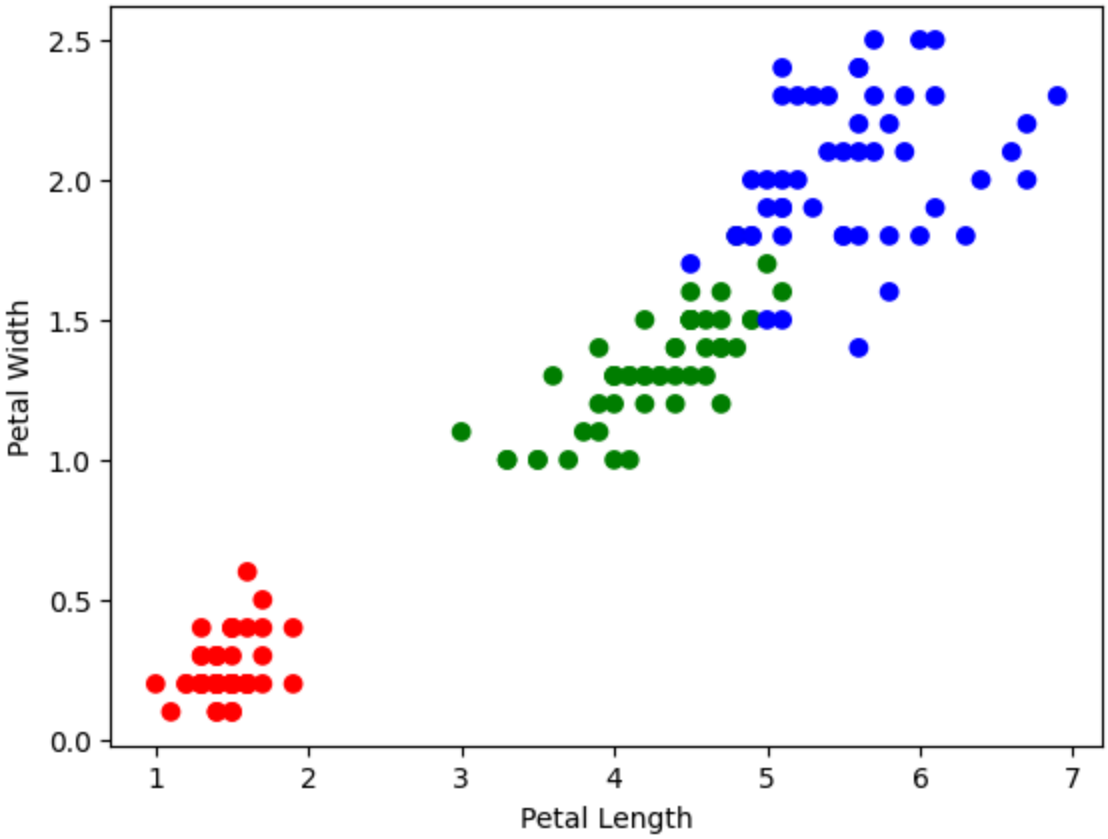
```
            min  max
species
setosa      1.0  1.9
versicolor  3.0  5.1
virginica   4.5  6.9
```

In [15]:
```
species_group['petal_length'].plot(kind='bar', color='orange')
plt.title('Average Petal Length by Species')
plt.xlabel('Species')
plt.ylabel('Average Petal Length')
plt.show()
```



In [16]:
```
# Filter flowers with petal length greater than 4.5
long_petals=df[df['petal_length']>4.5]
print("Flowers with petal length > 4.5:")
print(long_petals)

colors = {'setosa':'red', 'versicolor':'green', 'virginica':'blue'}
plt.scatter(df['petal_length'], df['petal_width'],
            c=df['species'].map(colors))

plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.show()
```

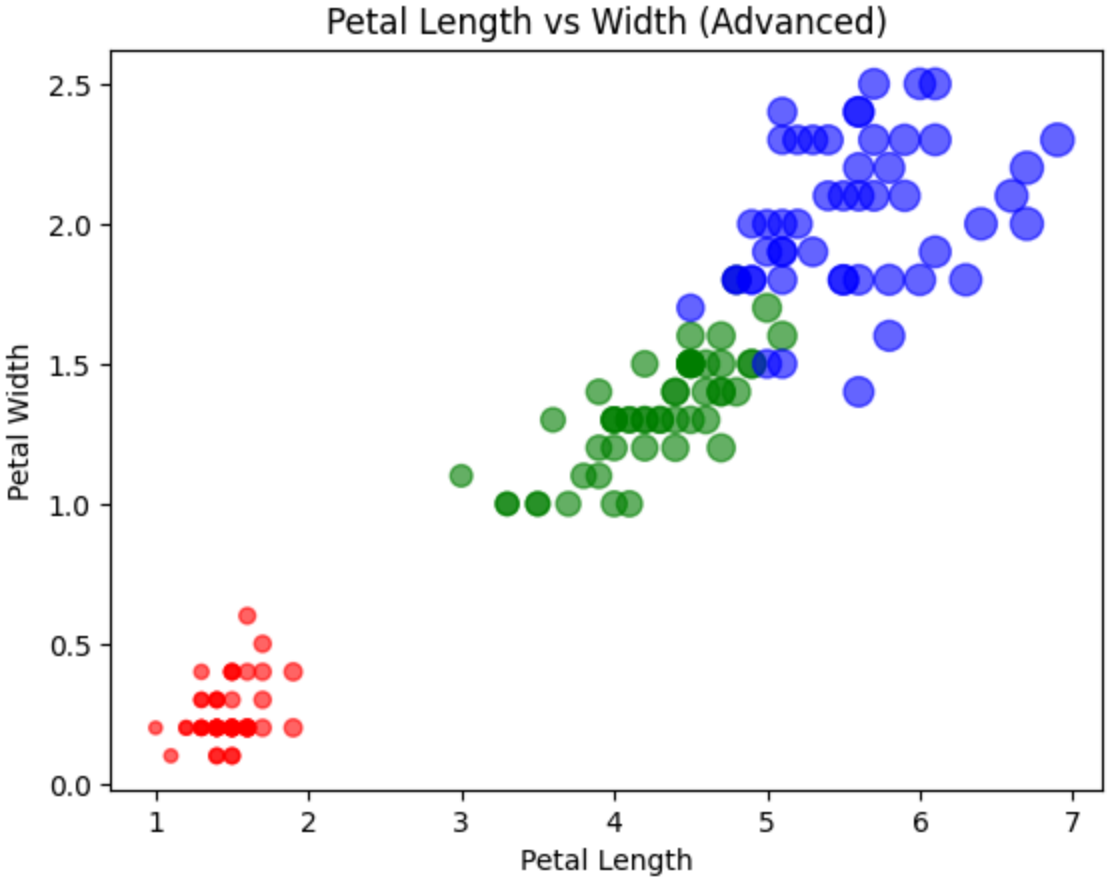
```
Flowers with petal length > 4.5:
     sepal_length  sepal_width  petal_length  petal_width     species
50            7.0          3.2           4.7          1.4  versicolor
52            6.9          3.1           4.9          1.5  versicolor
54            6.5          2.8           4.6          1.5  versicolor
56            6.3          3.3           4.7          1.6  versicolor
58            6.6          2.9           4.6          1.3  versicolor
..            ...          ...           ...          ...         ...
145           6.7          3.0           5.2          2.3   virginica
146           6.3          2.5           5.0          1.9   virginica
147           6.5          3.0           5.2          2.0   virginica
148           6.2          3.4           5.4          2.3   virginica
149           5.9          3.0           5.1          1.8   virginica

[63 rows x 5 columns]
```



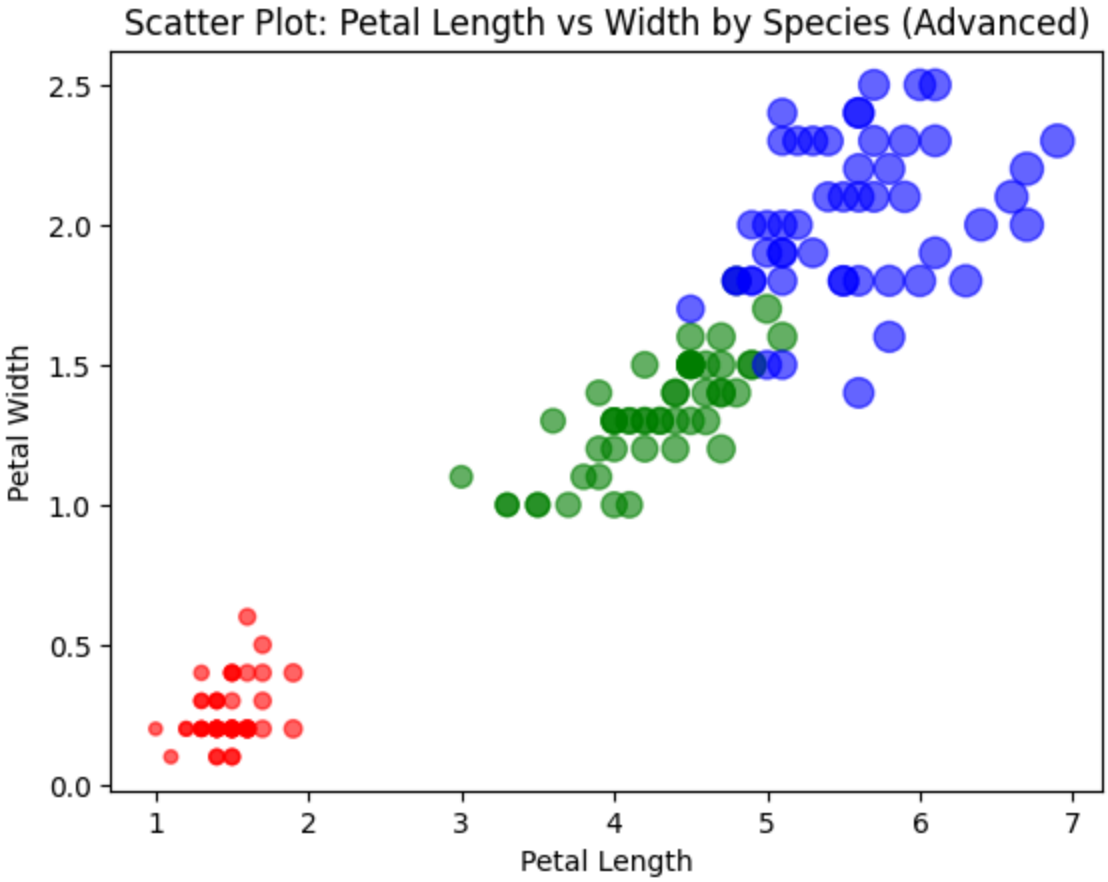
In [17]:
```python
# Filter flowers above species mean petal length
species_mean_petals = df.groupby('species')['petal_length'].mean()
above_avg = df[df.apply(lambda row: row['petal_length'] > species_mean_petals[row['
print(above_avg.head())

# Count by species
print(above_avg['species'].value_counts())

# Scatter plot with color & size by species
colors = {'setosa':'red', 'versicolor':'green', 'virginica':'blue'}
sizes = df['petal_length']*20
plt.scatter(df['petal_length'], df['petal_width'], c=df['species'].map(colors), s=s
plt.title('Petal Length vs Width (Advanced)')
plt.xlabel('Petal Length')
```

```
plt.ylabel('Petal Width')
plt.show()
```

```
    sepal_length  sepal_width  petal_length  petal_width species
3            4.6          3.1           1.5          0.2  setosa
5            5.4          3.9           1.7          0.4  setosa
7            5.0          3.4           1.5          0.2  setosa
9            4.9          3.1           1.5          0.1  setosa
10           5.4          3.7           1.5          0.2  setosa
species
versicolor    27
setosa        26
virginica     25
Name: count, dtype: int64
```



In [18]:
```
colors = {'setosa':'red', 'versicolor':'green', 'virginica':'blue'}
sizes = df['petal_length']*20  # size proportional to petal length
plt.scatter(df['petal_length'], df['petal_width'],
            c=df['species'].map(colors), s=sizes, alpha=0.6)
plt.title('Scatter Plot: Petal Length vs Width by Species (Advanced)')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.show()
```



This report analyzes the Iris dataset. The goal is to explore flower features, compare species, and extract useful patterns to better understand the data.

Conclusion

Species differ clearly in petal length and width

Most flowers with longer petals are Virginica

Scatter plots help visualize patterns in the data