Age-Income Dataset

In [1]:
```
import pandas as pd
```

In [2]:
```
age_df = pd.read_csv('Age-Income.csv')
```

In [4]:
```
age_df.head(3)
```

Out[4]:

	Age	Income
0	Young	25000
1	Middle Age	54000
2	Old	60000

In [27]:
```
def calculate_mean(column):
    length = len(column)
    total_sum = 0
    for value in column:
        total_sum += value
    mean = total_sum / length
    print("Mean Value: ", mean)

def calculate_median(column):
    column.sort_values()
    length = len(column)
    if length % 2 == 0:
        first = column[length//2]
        second = column[length//2 - 1]
        median = (first + second)/2
    else:
        median = column[length//2]
    print("Median :",median)

def minimum_value(column):
    min_val = column[0]
    for value in column:
        if value < min_val:
            min_val = value
    print("Minimum value :", min_val)

def maximum_value(column):
    max_val = column[0]  # Initialize max_val with the first element of the column
    for value in column:
        if value > max_val:
            max_val = value
    print("Maximum value :", max_val)

def standard_deviation(column):
    length = len(column)
    total_sum = 0
    summation = 0
    for value in column:
        total_sum += value
    mean = total_sum / length
    for i in column:
        summation+= (i-mean)**2
    std_deviation = ((summation)/length)**0.5
    print("Standard Deviation: ", std_deviation)
```

In [63]:
```
calculate_mean(age_df['Income'])
calculate_median(age_df['Income'])
minimum_value(age_df['Income'])
maximum_value(age_df['Income'])
standard_deviation(age_df['Income'])
```

```
Mean Value:  50966.0
Median : 91500.0
Minimum value : 15000
Maximum value : False
Standard Deviation:  20884.6509187968
```

In [66]:
```
age_df.groupby('Age')['Income'].mean()
```

Out[66]:
```
Age
Middle Age    52453.333333
Old           53942.105263
Young         46037.500000
Name: Income, dtype: float64
```

In [67]:
```
age_df.groupby('Age')['Income'].median()
```

Out[67]:
```
Age
Middle Age    53200.0
Old           45300.0
Young         41500.0
Name: Income, dtype: float64
```

In [69]:
```
age_df.groupby('Age')['Income'].std()
```

Out[69]:
```
Age
Middle Age    20497.800114
Old           20868.165968
Young         22356.859499
Name: Income, dtype: float64
```

Iris Dataset

In [77]:
```
iris_df = pd.read_csv("iris.csv")
iris_df.drop('Id', axis = 1, inplace = True)
```

In [78]:
```
categories = [i for i in iris_df['Species'].unique()]
categories
```

Out[78]:
```
['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
```

In [79]:
```
features = [feat for feat in iris_df.columns if iris_df[feat].dtype != 'O']
features
```

Out[79]:
```
['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']
```

In [80]:
```
species_group = iris_df.groupby('Species')
```

In [81]:
```
species_group.mean()
```

Out[81]:

Species	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Iris-setosa	5.006	3.418	1.464	0.244
Iris-versicolor	5.936	2.770	4.260	1.326
Iris-virginica	6.588	2.974	5.552	2.026

In [82]:
```
species_group.std()
```

Out[82]:

Species	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Iris-setosa	0.352490	0.381024	0.173511	0.107210
Iris-versicolor	0.516171	0.313798	0.469911	0.197753
Iris-virginica	0.635880	0.322497	0.551895	0.274650

In [83]:
```
species_group.var()
```

Out[83]:

Species	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Iris-setosa	0.124249	0.145180	0.030106	0.011494
Iris-versicolor	0.266433	0.098469	0.220816	0.039106
Iris-virginica	0.404343	0.104004	0.304588	0.075433

In [87]:
```
print('Iris-setosa')
setosa = iris_df['Species'] == 'Iris-setosa'
print(iris_df[setosa].describe())
print('\nIris-versicolor')
versicolor = iris_df['Species'] == 'Iris-versicolor'
print(iris_df[versicolor].describe())
print('\nIris-virginica')
virginica = iris_df['Species'] == 'Iris-virginica'
print(iris_df[virginica].describe())
```

```
Iris-setosa
       SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count       50.00000     50.000000      50.000000      50.00000
mean         5.00600      3.418000       1.464000       0.24400
std          0.35249      0.381024       0.173511       0.10721
min          4.30000      2.300000       1.000000       0.10000
25%          4.80000      3.125000       1.400000       0.20000
50%          5.00000      3.400000       1.500000       0.20000
75%          5.20000      3.675000       1.575000       0.30000
max          5.80000      4.400000       1.900000       0.60000

Iris-versicolor
       SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count      50.000000     50.000000      50.000000     50.000000
mean        5.936000      2.770000       4.260000      1.326000
std         0.516171      0.313798       0.469911      0.197753
min         4.900000      2.000000       3.000000      1.000000
25%         5.600000      2.525000       4.000000      1.200000
50%         5.900000      2.800000       4.350000      1.300000
75%         6.300000      3.000000       4.600000      1.500000
max         7.000000      3.400000       5.100000      1.800000
```

```
Iris-virginica
       SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count       50.00000     50.000000      50.000000      50.00000
mean         6.58800      2.974000       5.552000       2.02600
std          0.63588      0.322497       0.551895       0.27465
min          4.90000      2.200000       4.500000       1.40000
25%          6.22500      2.800000       5.100000       1.80000
50%          6.50000      3.000000       5.550000       2.00000
75%          6.90000      3.175000       5.875000       2.30000
max          7.90000      3.800000       6.900000       2.50000
```

In [90]:

```
def covariance(x, y):
    mean_x = sum(x)/len(x)
    mean_y = sum(y)/len(y)
    sub_x = [i - mean_x for i in x]
    sub_y = [i - mean_y for i in y]
    numerator = sum([sub_x[i]*sub_y[i] for i in range(len(sub_x))])
    denominator = len(x)-1
    cov = numerator/denominator
    return cov
```

In [94]:

```
for i in [0,1,2,3]:
    for j in [0,1,2,3]:
        if (i < j and i != j):
            val = covariance(iris_df[features[i]],iris_df[features[j]])
            print('Covariance for {} and {} : {}'.format(features[i],features[j], val))
```

```
Covariance for SepalLengthCm and SepalWidthCm : -0.03926845637583892
Covariance for SepalLengthCm and PetalLengthCm : 1.2736823266219242
Covariance for SepalLengthCm and PetalWidthCm : 0.5169038031319912
Covariance for SepalWidthCm and PetalLengthCm : -0.32171275167785246
Covariance for SepalWidthCm and PetalWidthCm : -0.11798120805369122
Covariance for PetalLengthCm and PetalWidthCm : 1.2963874720357946
```

In [95]:

```
def correlation(x, y):
    mean_x = sum(x)/float(len(x))
    mean_y = sum(y)/float(len(y))
    sub_x = [i-mean_x for i in x]
    sub_y = [i-mean_y for i in y]
    numerator = sum([sub_x[i]*sub_y[i] for i in range(len(sub_x))])
    std_deviation_x = sum([sub_x[i]**2.0 for i in range(len(sub_x))])
    std_deviation_y = sum([sub_y[i]**2.0 for i in range(len(sub_y))])
    denominator = (std_deviation_x*std_deviation_y)**0.5
    cor = numerator/denominator
    return cor
```

In [96]:

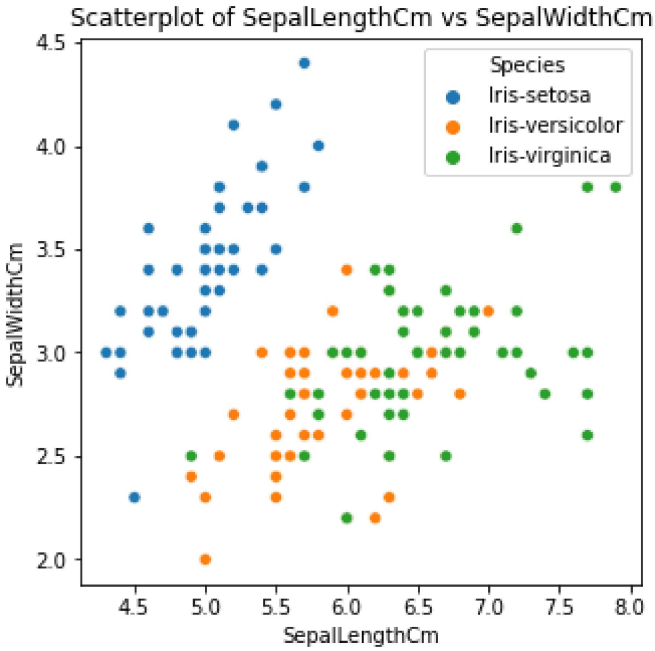
```
for i in [0,1,2,3]:
    for j in [0,1,2,3]:
        if (i < j and i != j):
            val = correlation(iris_df[features[i]],iris_df[features[j]])
            print('Correlation coefficient for {} and {} : {}'.format(features[i],features[j],
```

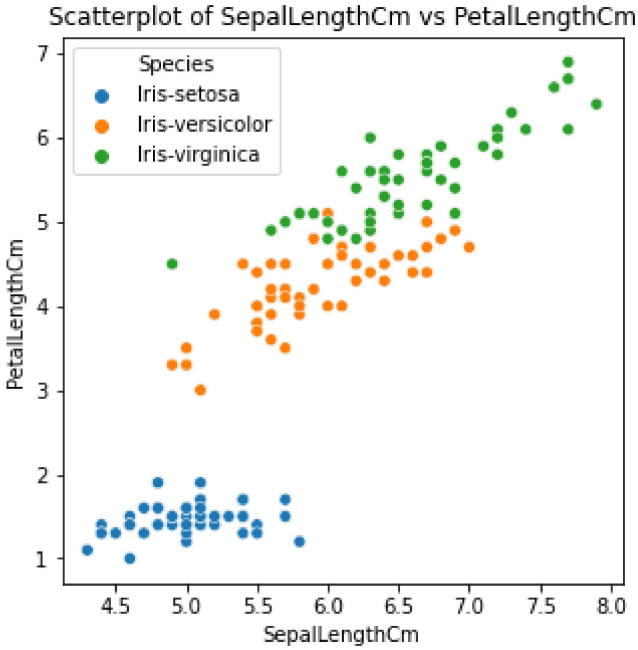
```
Correlation coefficient for SepalLengthCm and SepalWidthCm : -0.10936924995064935
Correlation coefficient for SepalLengthCm and PetalLengthCm : 0.8717541573048719
Correlation coefficient for SepalLengthCm and PetalWidthCm : 0.8179536333691635
Correlation coefficient for SepalWidthCm and PetalLengthCm : -0.42051609640115484
Correlation coefficient for SepalWidthCm and PetalWidthCm : -0.3565440896138055
Correlation coefficient for PetalLengthCm and PetalWidthCm : 0.9627570970509667
```

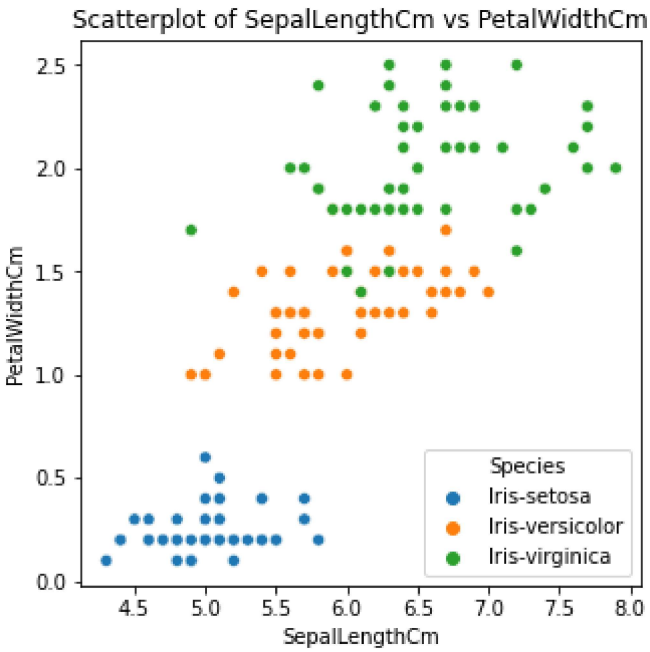
In [107…

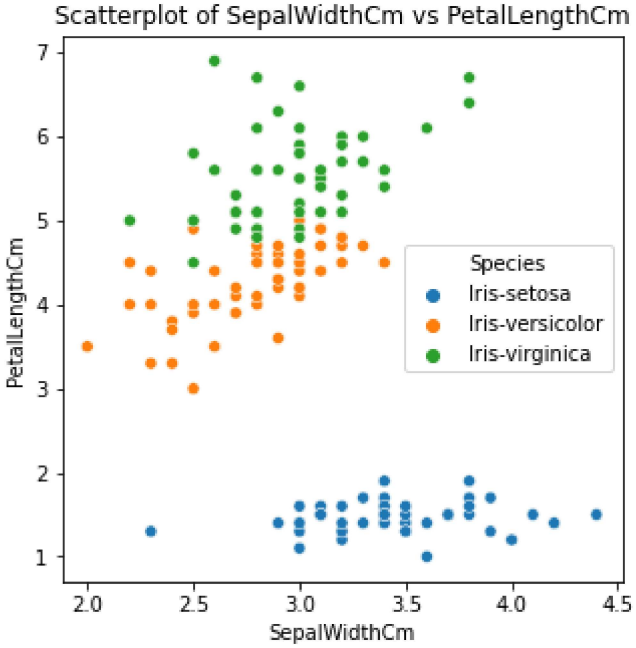
```
import matplotlib.pyplot as plt
import seaborn as sns

features = [feat for feat in iris_df.columns if iris_df[feat].dtype != 'O']
```

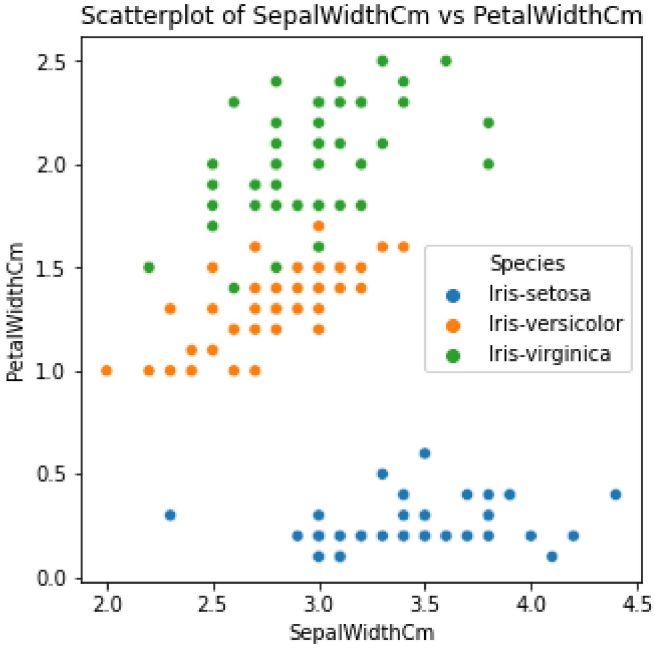
```
for i in range(len(features)):
    for j in range(len(features)):
        if (i < j and i != j):
            fig = plt.figure()
            fig.set_figheight(5)
            fig.set_figwidth(5)
            ax = sns.scatterplot(x=features[i], y=features[j], data=iris_df, hue='Species')
            plt.xlabel(features[i])
            plt.ylabel(features[j])
            plt.title('Scatterplot of {} vs {}'.format(features[i], features[j]))
            plt.show()
```

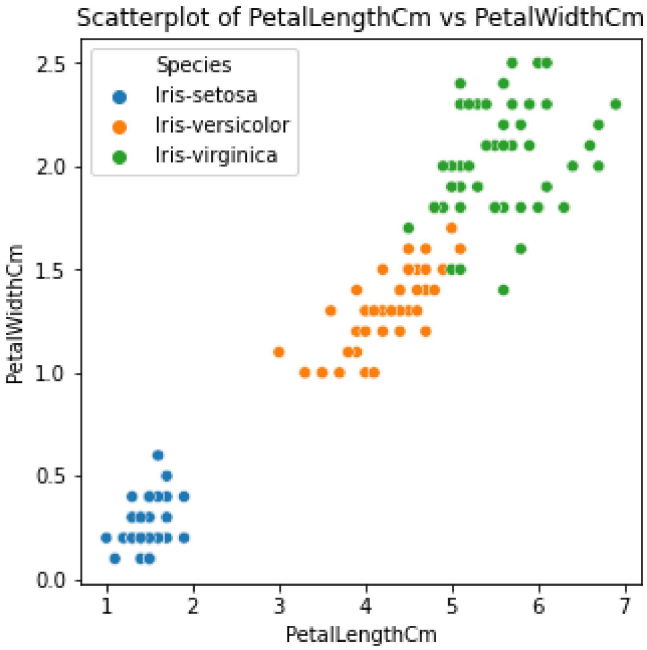





Scatterplot of SepalLengthCm vs PetalWidthCm
PetalWidthCm
SepalLengthCm
Species
Iris-setosa
Iris-versicolor
Iris-virginica


Scatterplot of SepalWidthCm vs PetalLengthCm
PetalLengthCm
SepalWidthCm
Species
Iris-setosa
Iris-versicolor
Iris-virginica


Scatterplot of SepalWidthCm vs PetalWidthCm
PetalWidthCm
SepalWidthCm
Species
Iris-setosa
Iris-versicolor
Iris-virginica



In [108…

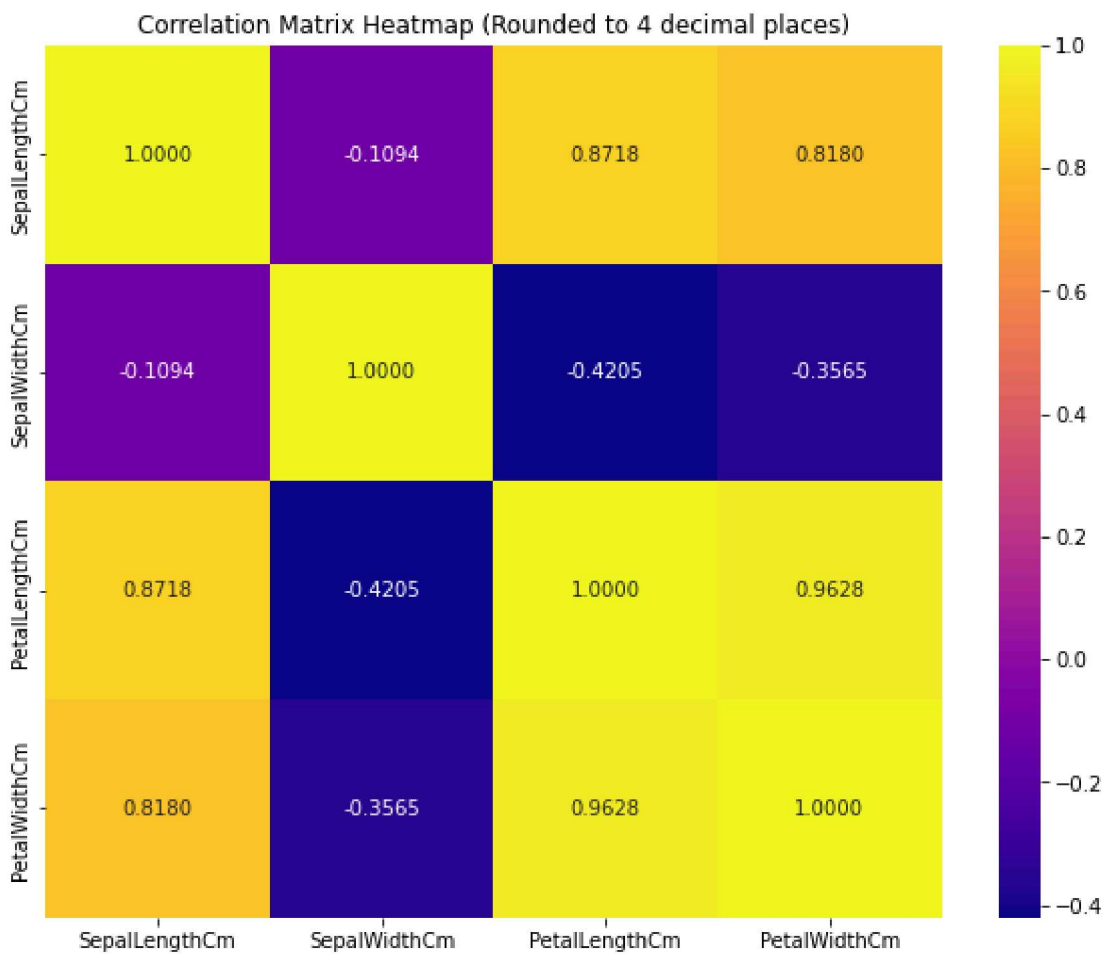
```
cormatrix = iris_df.corr()
round(cormatrix,4)
```

Out[108…

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.0000	-0.1094	0.8718	0.8180
SepalWidthCm	-0.1094	1.0000	-0.4205	-0.3565
PetalLengthCm	0.8718	-0.4205	1.0000	0.9628
PetalWidthCm	0.8180	-0.3565	0.9628	1.0000

In [109…

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

cormatrix = iris_df.corr()
cormatrix_rounded = round(cormatrix, 4)

plt.figure(figsize=(10, 8))
sns.heatmap(cormatrix_rounded, annot=True, cmap='plasma', fmt=".4f", annot_kws={"size": 10})
plt.title('Correlation Matrix Heatmap (Rounded to 4 decimal places)')
plt.show()
```


Correlation Matrix Heatmap (Rounded to 4 decimal places)
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
1.0000
-0.1094
0.8718
0.8180
-0.1094
1.0000
-0.4205
-0.3565
0.8718
-0.4205
1.0000
0.9628
0.8180
-0.3565
0.9628
1.0000
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

In [ ]: