A3_31378

January 25, 2022

Name: Sourav Kotkar

Roll No: 31378

0.1 Descriptive Statistics - Measures of Central Tendency and variability

Perform the following operations on any open-source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.

2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset.

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

0.2 1. Mall Customers Dataset

[2]:
```
data = pd.read_csv('Mall_Customers_data.csv')
```

[3]:
```
data
```

[3]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
..	…	…	…	…	…
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74

```
198         199    Male   32                  137                      18
199         200    Male   30                  137                      83

[200 rows x 5 columns]
```

[4]:
```
data.isnull().sum() #Returns number of null values
```

[4]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

[5]:
```
data.dtypes #Returns datatypes of all columns
```

[5]:
```
CustomerID                 int64
Gender                    object
Age                        int64
Annual Income (k$)         int64
Spending Score (1-100)     int64
dtype: object
```

[6]:
```
data.describe() #Statistical information
```

[6]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

[7]:
```
data.groupby('Gender').mean() #Mean values grouped by gender
```

[7]:

Gender	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Female	97.562500	38.098214	59.250000	51.526786
Male	104.238636	39.806818	62.227273	48.511364

[8]:
```
data.groupby('Gender').median() #Median values grouped by gender
```

[8]:

Gender	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Female	94.5	35.0	60.0	50.0
Male	106.5	37.0	62.5	50.0

[9]:
```
data.groupby('Gender').min() #Minimum values grouped by gender
```

[9]:

Gender	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Female	3	18	16	5
Male	1	18	15	1

[10]:
```
data.groupby('Gender').max() #Maximum values grouped by gender
```

[10]:

Gender	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Female	197	68	126	99
Male	200	70	137	97

[11]:
```
data.groupby('Gender').std() #Standard deviation grouped by gender
```

[11]:

Gender	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Female	58.276412	12.644095	26.011952	24.11495
Male	57.483830	15.514812	26.638373	27.89677

[12]:
```
data.groupby('Gender').agg(lambda x:x.value_counts().index[0]) #Mode values␣
↪grouped by gender
```

[12]:

Gender	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Female	3	31	78	42
Male	1	19	54	46

[13]:
```
data.groupby('Gender')['Age'].describe() #Statistical information of age␣
↪grouped by gender
```

[13]:

Gender	count	mean	std	min	25%	50%	75%	max
Female	112.0	38.098214	12.644095	18.0	29.00	35.0	47.5	68.0
Male	88.0	39.806818	15.514812	18.0	27.75	37.0	50.5	70.0

[14]:
```
data.groupby('Gender')['Annual Income (k$)'].describe() #Statistical␣
↪information of annual income grouped by gender
```

[14]:

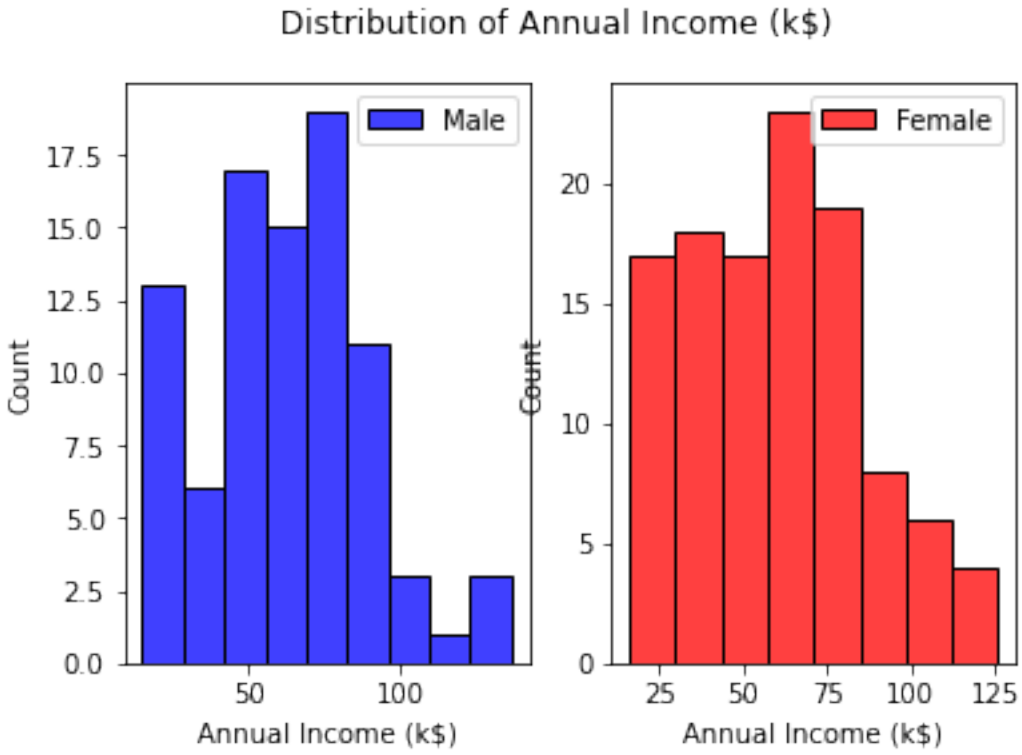
Gender	count	mean	std	min	25%	50%	75%	max
Female	112.0	59.250000	26.011952	16.0	39.75	60.0	77.25	126.0
Male	88.0	62.227273	26.638373	15.0	45.50	62.5	78.00	137.0

[15]:
```
data.groupby('Gender')['Spending Score (1-100)'].describe() #Statistical␣
↪information of spending score grouped by gender
```

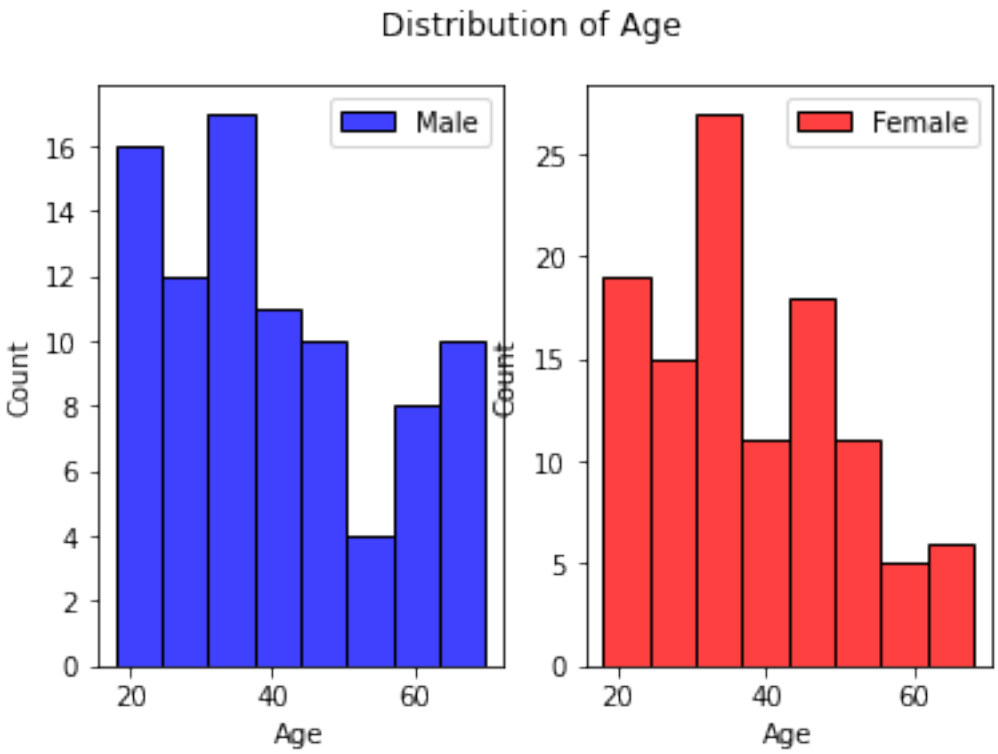
[15]:

Gender	count	mean	std	min	25%	50%	75%	max
Female	112.0	51.526786	24.11495	5.0	35.0	50.0	73.0	99.0
Male	88.0	48.511364	27.89677	1.0	24.5	50.0	70.0	97.0

[16]:
```
#Distribution of annual income

x1 = data[data['Gender'] == 'Male']
x2 = data[data['Gender'] == 'Female']

plt.subplot(1,2,1)
sns.histplot(x1['Annual Income (k$)'], color = 'blue', label='Male')
plt.legend()

plt.subplot(1,2,2)
sns.histplot(x2['Annual Income (k$)'], color = 'red', label='Female')
plt.legend()

plt.suptitle('Distribution of Annual Income (k$)')

plt.show()
```



[17]:
```
#Distribution of age

x1 = data[data['Gender'] == 'Male']
x2 = data[data['Gender'] == 'Female']

plt.subplot(1,2,1)
sns.histplot(x1['Age'], color = 'blue', label='Male')
plt.legend()

plt.subplot(1,2,2)
sns.histplot(x2['Age'], color = 'red', label='Female')
plt.legend()

plt.suptitle('Distribution of Age')

plt.show()
```



[18]:
```
#Distribution of spending score

x1 = data[data['Gender'] == 'Male']
x2 = data[data['Gender'] == 'Female']
```

```
plt.subplot(1,2,1)
sns.histplot(x1['Spending Score (1-100)'], color = 'blue', label='Male')
plt.legend()

plt.subplot(1,2,2)
sns.histplot(x2['Spending Score (1-100)'], color = 'red', label='Female')
plt.legend()

plt.suptitle('Distribution of Spending Score (1-100)')

plt.show()
```

[ ]:

0.3 2. Iris Dataset

[19]:
```
df = pd.read_csv('Iris.csv')
```

[20]:
```
df
```

[20]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	\
0	1	5.1	3.5	1.4	0.2	

```
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   …            …             …              …             …
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
..              …
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]
```

[21]:
```
df.isnull().sum() #Returns number of null values
```

[21]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

[22]:
```
df.dtypes #Returns datatypes of all columns
```

[22]:
```
Id                int64
SepalLengthCm   float64
SepalWidthCm    float64
PetalLengthCm   float64
PetalWidthCm    float64
Species          object
dtype: object
```

[23]:
```
df.describe() #Statistical information
```

[23]:
```
                Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count   150.000000     150.000000    150.000000     150.000000    150.000000
mean     75.500000       5.843333      3.054000       3.758667      1.198667
std      43.445368       0.828066      0.433594       1.764420      0.763161
min       1.000000       4.300000      2.000000       1.000000      0.100000
25%      38.250000       5.100000      2.800000       1.600000      0.300000
50%      75.500000       5.800000      3.000000       4.350000      1.300000
75%     112.750000       6.400000      3.300000       5.100000      1.800000
max     150.000000       7.900000      4.400000       6.900000      2.500000
```

[24]:
```
df.groupby('Species').mean() #Mean values grouped by species
```

[24]:
```
                  Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  \
Species
Iris-setosa       25.5          5.006         3.418          1.464
Iris-versicolor   75.5          5.936         2.770          4.260
Iris-virginica   125.5          6.588         2.974          5.552

                 PetalWidthCm
Species
Iris-setosa             0.244
Iris-versicolor         1.326
Iris-virginica          2.026
```

[25]:
```
df.groupby('Species').median() #Median values grouped by species
```

[25]:
```
                  Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  \
Species
Iris-setosa       25.5            5.0           3.4           1.50
Iris-versicolor   75.5            5.9           2.8           4.35
Iris-virginica   125.5            6.5           3.0           5.55

                 PetalWidthCm
Species
Iris-setosa               0.2
Iris-versicolor           1.3
Iris-virginica            2.0
```

[26]:
```
df.groupby('Species').min() #Minimum values grouped by species
```

[26]:
```
                 Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
Iris-setosa       1            4.3           2.3            1.0           0.1
Iris-versicolor  51            4.9           2.0            3.0           1.0
Iris-virginica  101            4.9           2.2            4.5           1.4
```

[27]:
```
df.groupby('Species').max() #Maximum values grouped by species
```

[27]:
```
                 Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
Iris-setosa      50            5.8           4.4            1.9           0.6
Iris-versicolor  100           7.0           3.4            5.1           1.8
Iris-virginica   150           7.9           3.8            6.9           2.5
```

[28]:
```
df.groupby('Species').std() #Standard deviation grouped by species
```

[28]:
```
                      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  \
Species
Iris-setosa      14.57738       0.352490      0.381024       0.173511
Iris-versicolor  14.57738       0.516171      0.313798       0.469911
Iris-virginica   14.57738       0.635880      0.322497       0.551895

                 PetalWidthCm
Species
Iris-setosa          0.107210
Iris-versicolor      0.197753
Iris-virginica       0.274650
```

[29]:
```
df.groupby('Species').agg(lambda x:x.value_counts().index[0]) #Mode values␣
↪grouped by species
```

[29]:
```
                 Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
Iris-setosa       1            5.1           3.4            1.5           0.2
Iris-versicolor  51            5.5           3.0            4.5           1.3
Iris-virginica  101            6.3           3.0            5.1           1.8
```

[30]:
```
df.groupby('Species')['SepalLengthCm'].describe() #Statistical information of␣
↪sepal length grouped by species
```

[30]:
```
                 count   mean       std  min    25%  50%  75%  max
Species
Iris-setosa       50.0  5.006  0.352490  4.3  4.800  5.0  5.2  5.8
Iris-versicolor   50.0  5.936  0.516171  4.9  5.600  5.9  6.3  7.0
Iris-virginica    50.0  6.588  0.635880  4.9  6.225  6.5  6.9  7.9
```

[31]:
```
df.groupby('Species')['SepalWidthCm'].describe() #Statistical information of␣
↪sepal width grouped by species
```

[31]:
```
                 count   mean       std  min    25%  50%    75%  max
Species
Iris-setosa       50.0  3.418  0.381024  2.3  3.125  3.4  3.675  4.4
Iris-versicolor   50.0  2.770  0.313798  2.0  2.525  2.8  3.000  3.4
Iris-virginica    50.0  2.974  0.322497  2.2  2.800  3.0  3.175  3.8
```

[32]:
```
df.groupby('Species')['PetalLengthCm'].describe() #Statistical information of␣
↪petal length grouped by species
```

[32]:

Species	count	mean	std	min	25%	50%	75%	max
Iris-setosa	50.0	1.464	0.173511	1.0	1.4	1.50	1.575	1.9
Iris-versicolor	50.0	4.260	0.469911	3.0	4.0	4.35	4.600	5.1
Iris-virginica	50.0	5.552	0.551895	4.5	5.1	5.55	5.875	6.9

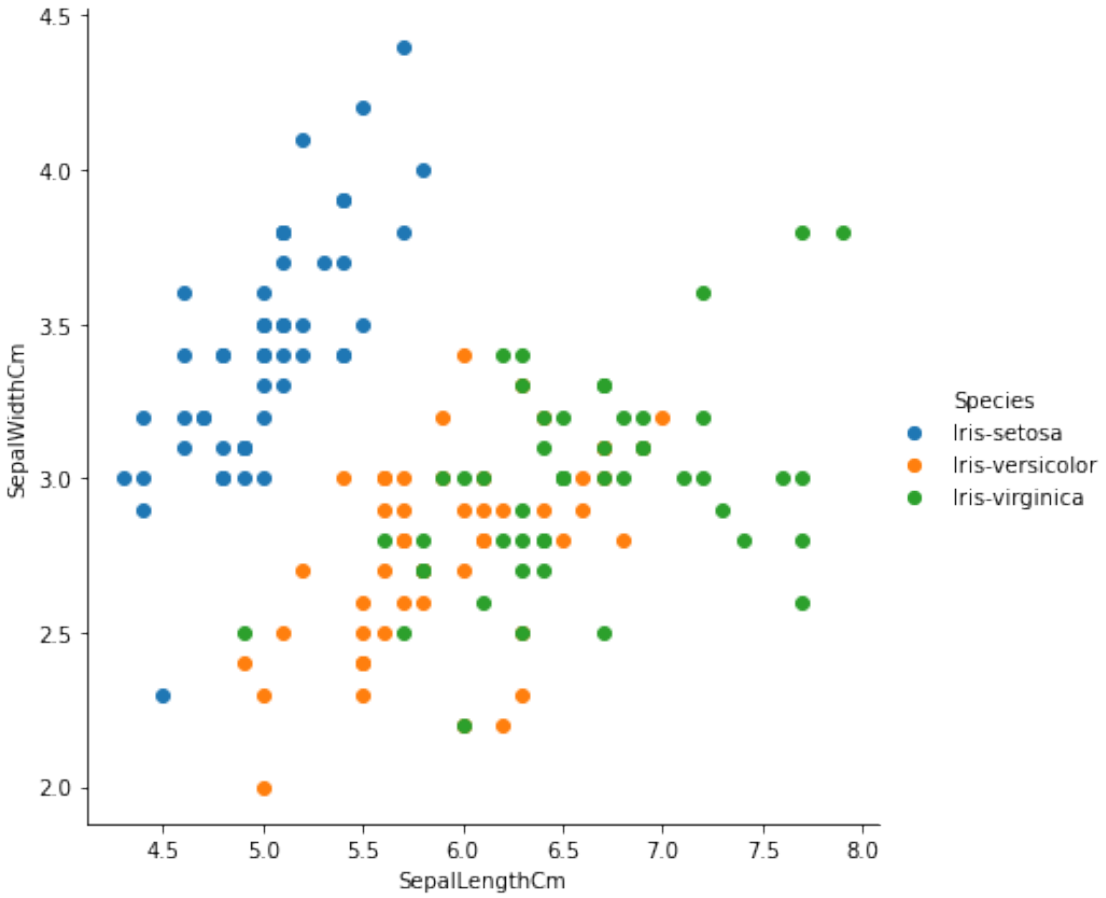
[33]:
```
df.groupby('Species')['PetalWidthCm'].describe() #Statistical information of␣
↪petal width grouped by species
```

[33]:

Species	count	mean	std	min	25%	50%	75%	max
Iris-setosa	50.0	0.244	0.107210	0.1	0.2	0.2	0.3	0.6
Iris-versicolor	50.0	1.326	0.197753	1.0	1.2	1.3	1.5	1.8
Iris-virginica	50.0	2.026	0.274650	1.4	1.8	2.0	2.3	2.5

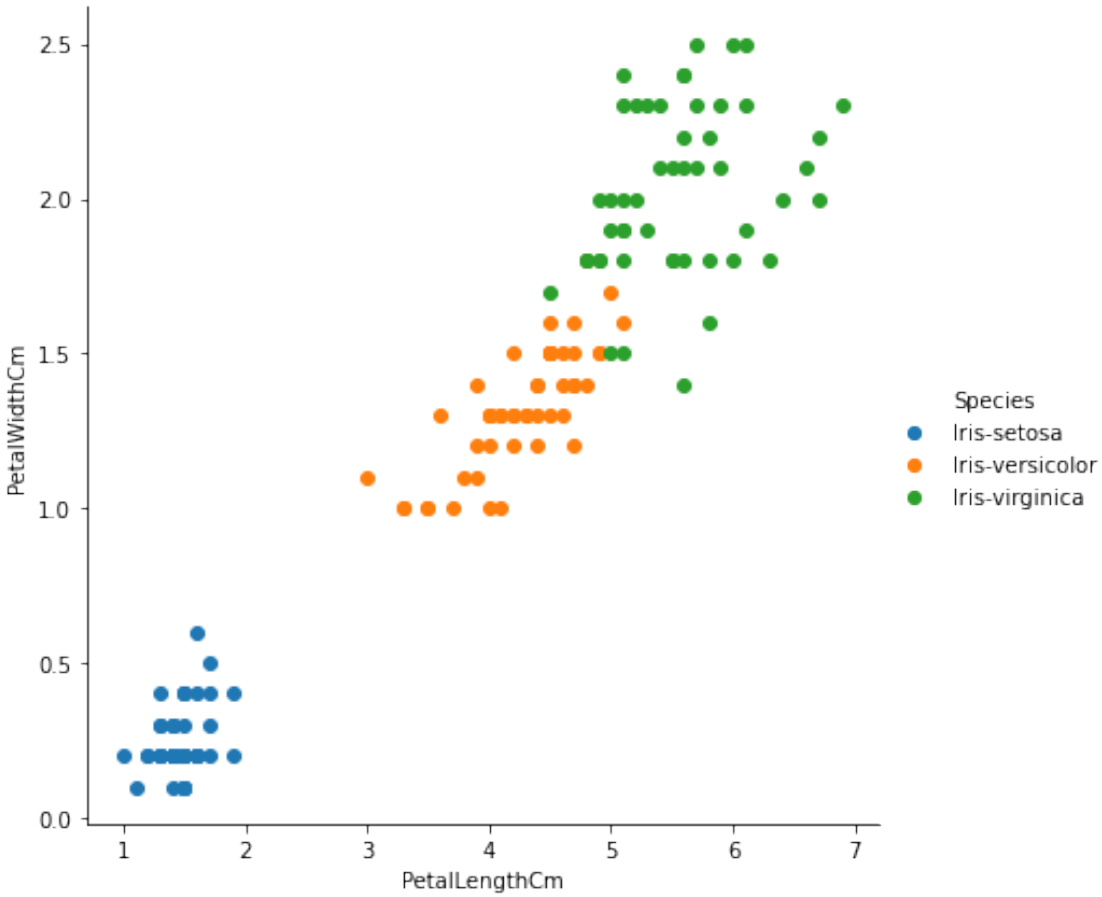
[34]:
```
sns.FacetGrid(df, hue ='Species', height = 6).map(plt.scatter, 'SepalLengthCm',␣
↪'SepalWidthCm').add_legend()
```

[34]: <seaborn.axisgrid.FacetGrid at 0x20c3f4de5e0>



[35]:
```
sns.FacetGrid(df, hue ='Species',  height = 6).map(plt.scatter,␣
↪'PetalLengthCm', 'PetalWidthCm').add_legend()
```

[35]: <seaborn.axisgrid.FacetGrid at 0x20c3f55efa0>



[ ]: