Live Session Unit 13 Assignment

MSDS 6306: Introduction to Data Science (403)

Misael Santana

1. Create a list named my_list in python with the following data points - 45.4 44.2 36.8 35.1 39.0 60.0 47.4 41.1 45.8 35.6

In [91]:
```
my_list=[45.4, 44.2 ,36.8, 35.1, 39.0 ,60.0 ,47.4, 41.1 ,45.8,35.6]
```

In [92]:
```
print(my_list)
```

```
[45.4, 44.2, 36.8, 35.1, 39.0, 60.0, 47.4, 41.1, 45.8, 35.6]
```

a. Print the 5th element in the list

In [93]:
```
print(my_list[4])
```

```
39.0
```

b. Append 55.2 to my_list

In [94]:
```
my_list.append(55.2)
```

In [95]:
```
print(my_list)
```

```
[45.4, 44.2, 36.8, 35.1, 39.0, 60.0, 47.4, 41.1, 45.8, 35.6, 55.2]
```

c. Remove the 6th element in the list

In [96]:
```
my_list.remove(my_list[5])
```

In [97]:
```
print(my_list)
```

```
[45.4, 44.2, 36.8, 35.1, 39.0, 47.4, 41.1, 45.8, 35.6, 55.2]
```

d. Iterate over the list to print data points greater than 45

In [98]:
```
print([x for x in my_list if x > 45])
```

```
[45.4, 47.4, 45.8, 55.2]
```

2. Introduction to numpy –

a. Import the numpy library using the following command – import numpy

In [99]:
```
import numpy as np
```

b. Declare numpy array with the same data points as in my_list using numpy.array()

In [100]:
```
my_listnp=np.array(my_list)
```

In [101]:
```
print(my_list)
```

```
[45.4, 44.2, 36.8, 35.1, 39.0, 47.4, 41.1, 45.8, 35.6, 55.2]
```

c. Compute the mean and standard deviation using numpy.mean() and numpy.std() of the above array

In [102]:
```
my_listnp.mean()
```

Out[102]: 42.560000000000002

In [103]:
```
my_listnp.std()
```

Out[103]: 5.9709630713981143

d. Use logical referencing to get only those values that are less than 45

In [104]:
```
my_listnp[np.where(my_listnp<45)]
```

Out[104]: array([44.2, 36.8, 35.1, 39. , 41.1, 35.6])

e. Compute the max and min of the array using numpy.max() and numpy.min()

In [105]:
```
np.max(my_listnp)
```

Out[105]: 55.200000000000003

In [106]:
```
np.min(my_listnp)
```

Out[106]: 35.100000000000001

3. Introduction to pandas –

a. Import the pandas library – import pandas

In [107]:
```
import pandas as pd
```

b. Read the IRIS dataset into iris using pandas.read_csv(). Data file –

In [108]:
```
iris = pd.read_csv('c:\sasdata\iris.csv')
```

c. Using iris.head(), display the head of the dataset

In [109]:
```
iris.head()
```

Out[109]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

d. Use DataFrame.drop() to drop the id column

In [110]:
```
iris=iris.drop('Id',1)
iris.head()
```

Out[110]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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

e. Subset dataframe to create a new data frame that includes only the measurements for the setosa species

In [55]:
```
iris2=iris[iris['Species']=='Iris-setosa']
```

In [56]:
```
iris2.head(10)
```

Out[56]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	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
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

f. Use DataFrame.describe() to get the summary statistics

In [57]:
```
iris.describe()
```

Out[57]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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

g. Use DataFrame.groupby() to create grouped data frames by Species and compute summary statistics using DataFrame.describe()

In [58]:
```
groupby_species=iris.groupby(by='Species')
groupby_species.describe()
```

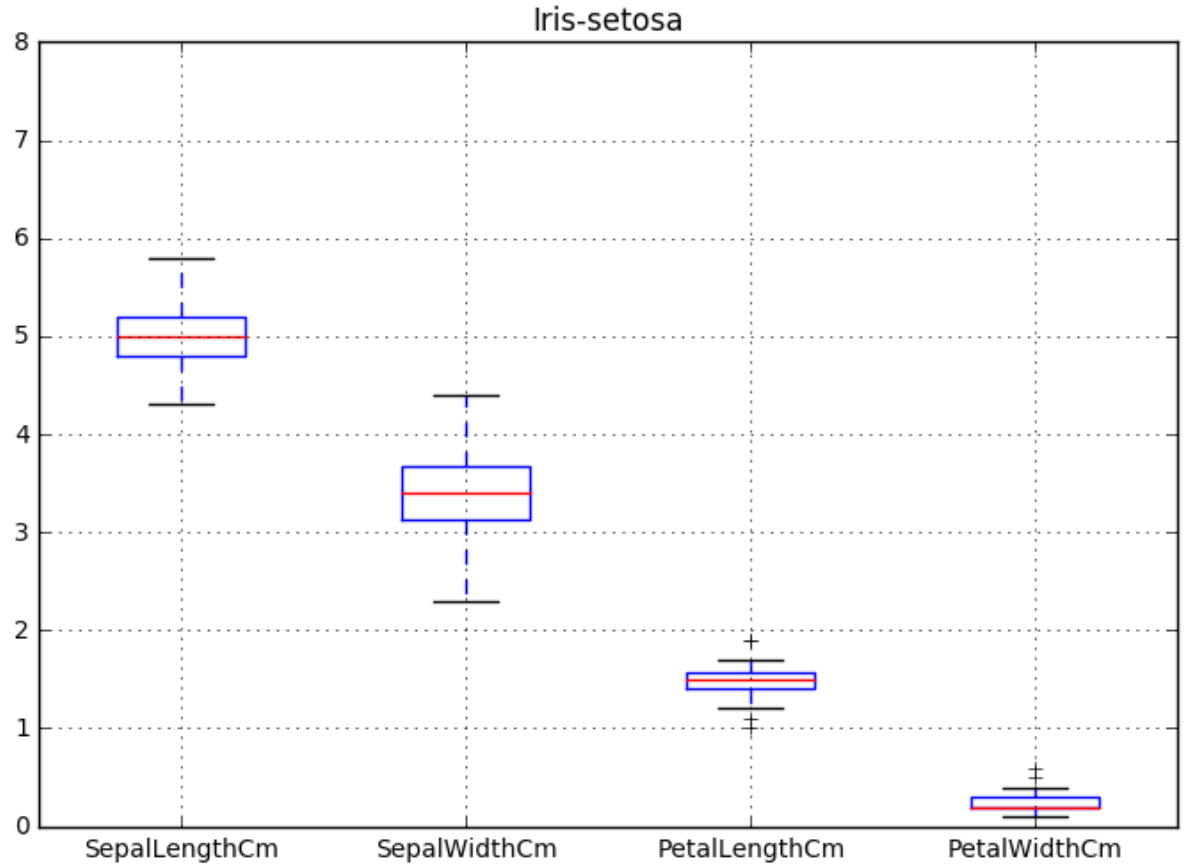
Out[58]:

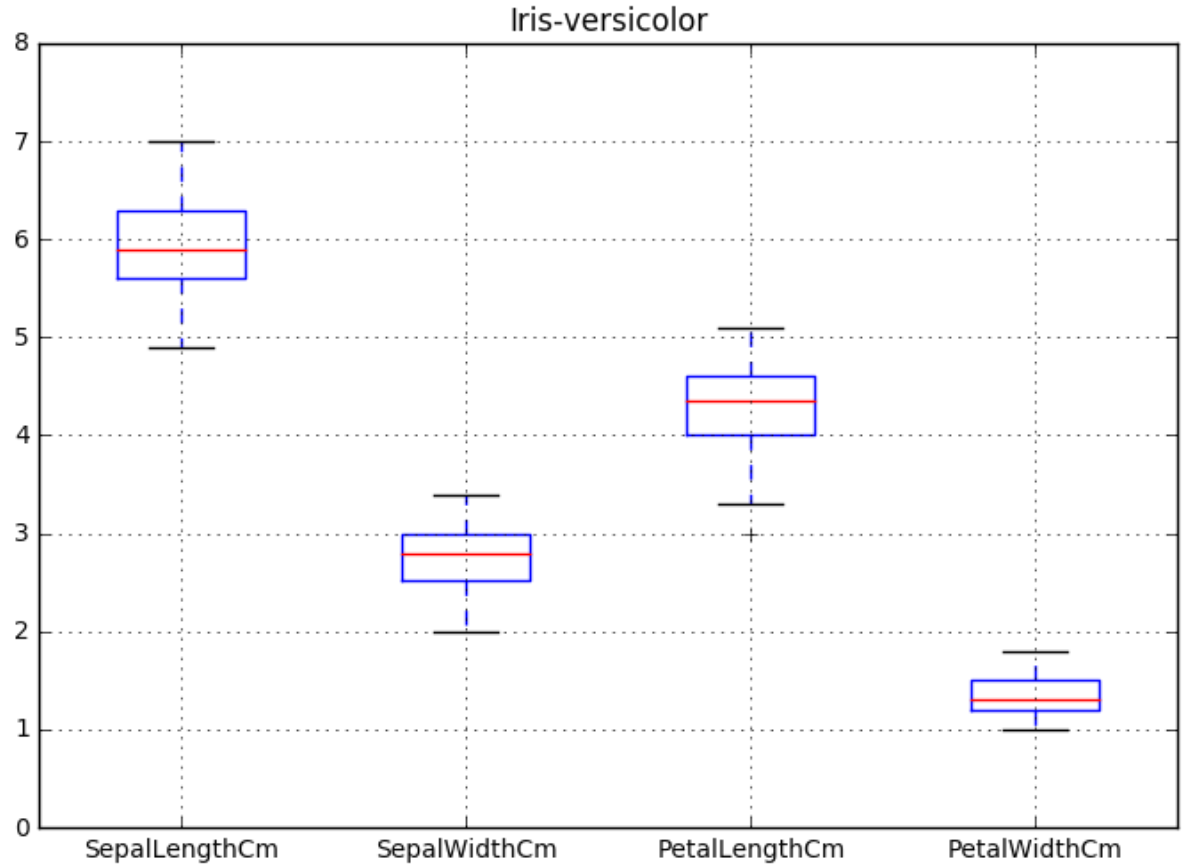
		PetalLengthCm	PetalWidthCm	SepalLengthCm	SepalWidthCm
Species					
Iris-setosa	**count**	50.000000	50.000000	50.000000	50.000000
	mean	1.464000	0.244000	5.006000	3.418000
	std	0.173511	0.107210	0.352490	0.381024
	min	1.000000	0.100000	4.300000	2.300000
	25%	1.400000	0.200000	4.800000	3.125000
	50%	1.500000	0.200000	5.000000	3.400000
	75%	1.575000	0.300000	5.200000	3.675000
	max	1.900000	0.600000	5.800000	4.400000
Iris-versicolor	**count**	50.000000	50.000000	50.000000	50.000000
	mean	4.260000	1.326000	5.936000	2.770000
	std	0.469911	0.197753	0.516171	0.313798
	min	3.000000	1.000000	4.900000	2.000000
	25%	4.000000	1.200000	5.600000	2.525000
	50%	4.350000	1.300000	5.900000	2.800000
	75%	4.600000	1.500000	6.300000	3.000000
	max	5.100000	1.800000	7.000000	3.400000
Iris-virginica	**count**	50.000000	50.000000	50.000000	50.000000
	mean	5.552000	2.026000	6.588000	2.974000
	std	0.551895	0.274650	0.635880	0.322497
	min	4.500000	1.400000	4.900000	2.200000
	25%	5.100000	1.800000	6.225000	2.800000
	50%	5.550000	2.000000	6.500000	3.000000
	75%	5.875000	2.300000	6.900000	3.175000
	max	6.900000	2.500000	7.900000	3.800000

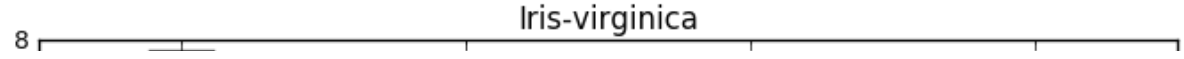
h. Use DataFrame.boxplot() to plot boxplots by Species

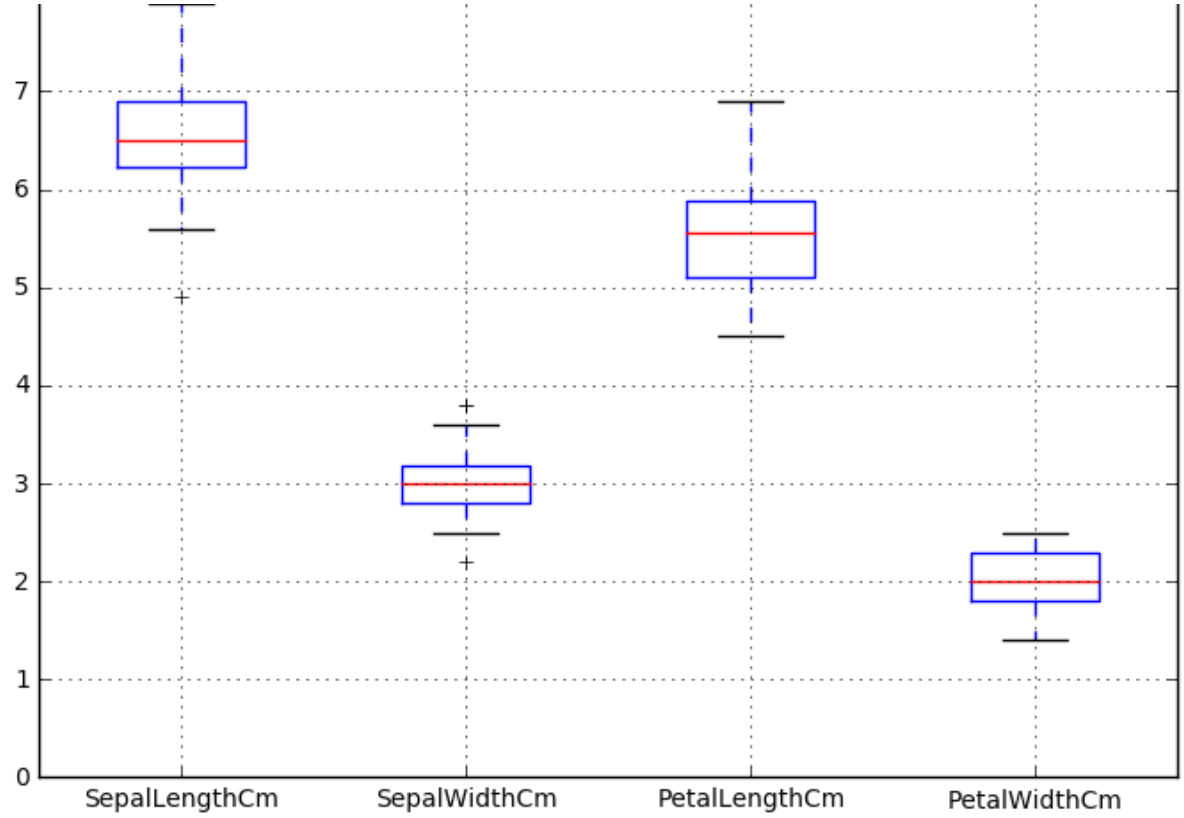
In [79]:
```
import matplotlib.pyplot as plt
fig=plt.figure()
groupby_species.boxplot(return_type='dict', layout = (3,1), figsize=(8,20))
plt.show()
```

```
<matplotlib.figure.Figure at 0x22cc1c37198>
```


Iris-setosa
8
7
6
5
4
3
2
1
0
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm


Iris-versicolor
8
7
6
5
4
3
2
1
0
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm


Iris-virginica
8



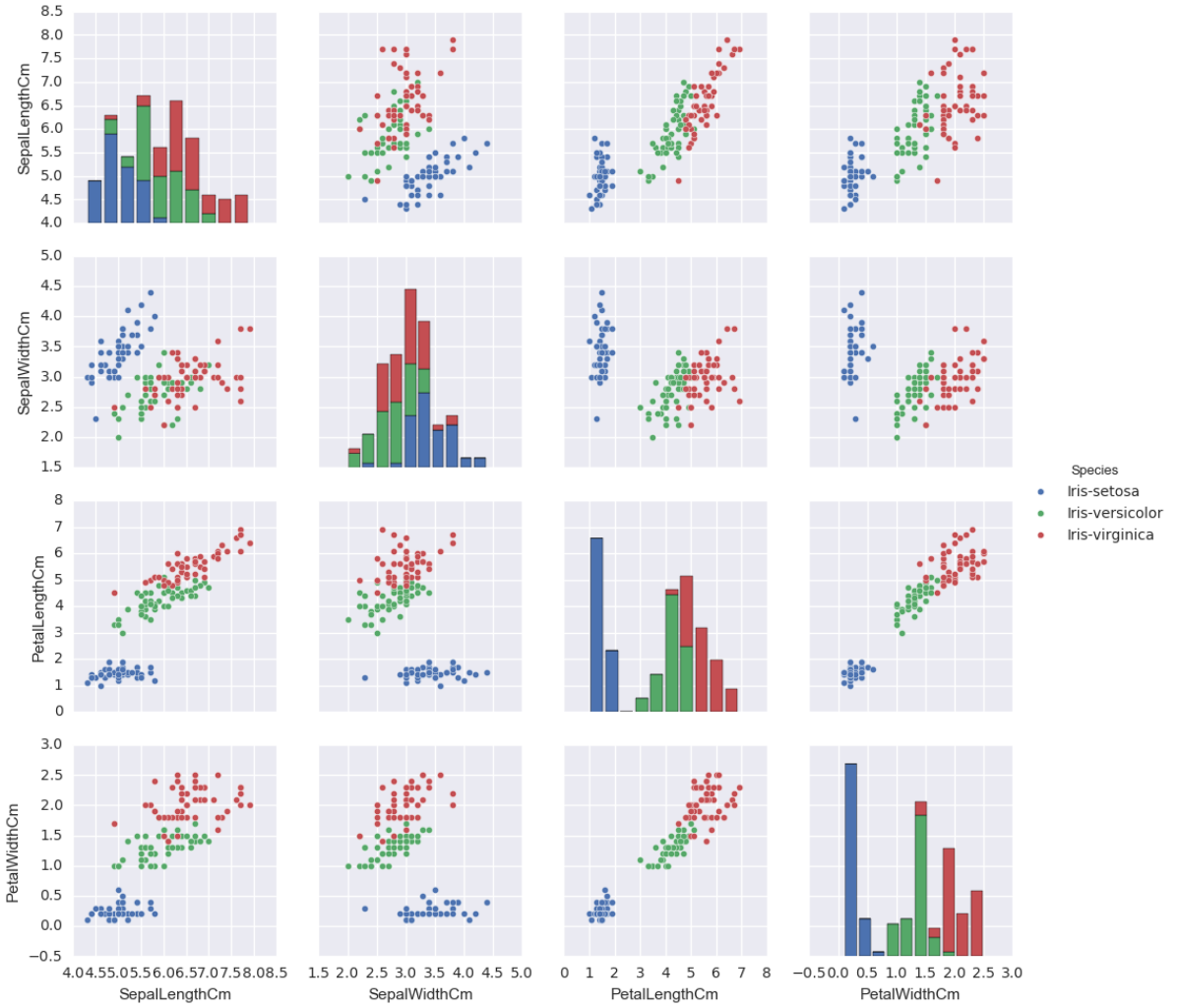
i. Plot a scatter matrix plot using the seaborn library. Use the following to load and plot

i. `Import seaborn`

ii. `Seaborn.pairplot(dataframe,by='column_name')`

In [89]:
```
import seaborn as sns
import matplotlib.pyplot as plt1
fig=plt1.figure()
sns.pairplot(iris,hue='Species')
plt1.show()
```

```
<matplotlib.figure.Figure at 0x22cc423c400>
```



In [ ]: