Assignment_2_Iris_dataset

September 26, 2021

[2]:

```
%pylab inline
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
```

```
Populating the interactive namespace from numpy and matplotlib
```

[3]:

```
column_names = ["sepal_length(cm)", "sepal_width(cm)", "petal_length(cm)",␣
 ↪"petal_width(cm)", "class"]
df = pd.read_csv("http://archive.ics.uci.edu/ml/machine-learning-databases/iris/
 ↪iris.data", names=column_names, header=None)
df.head()
```

[3]:

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

         class
0  Iris-setosa
1  Iris-setosa
2  Iris-setosa
3  Iris-setosa
4  Iris-setosa
```

[4]:

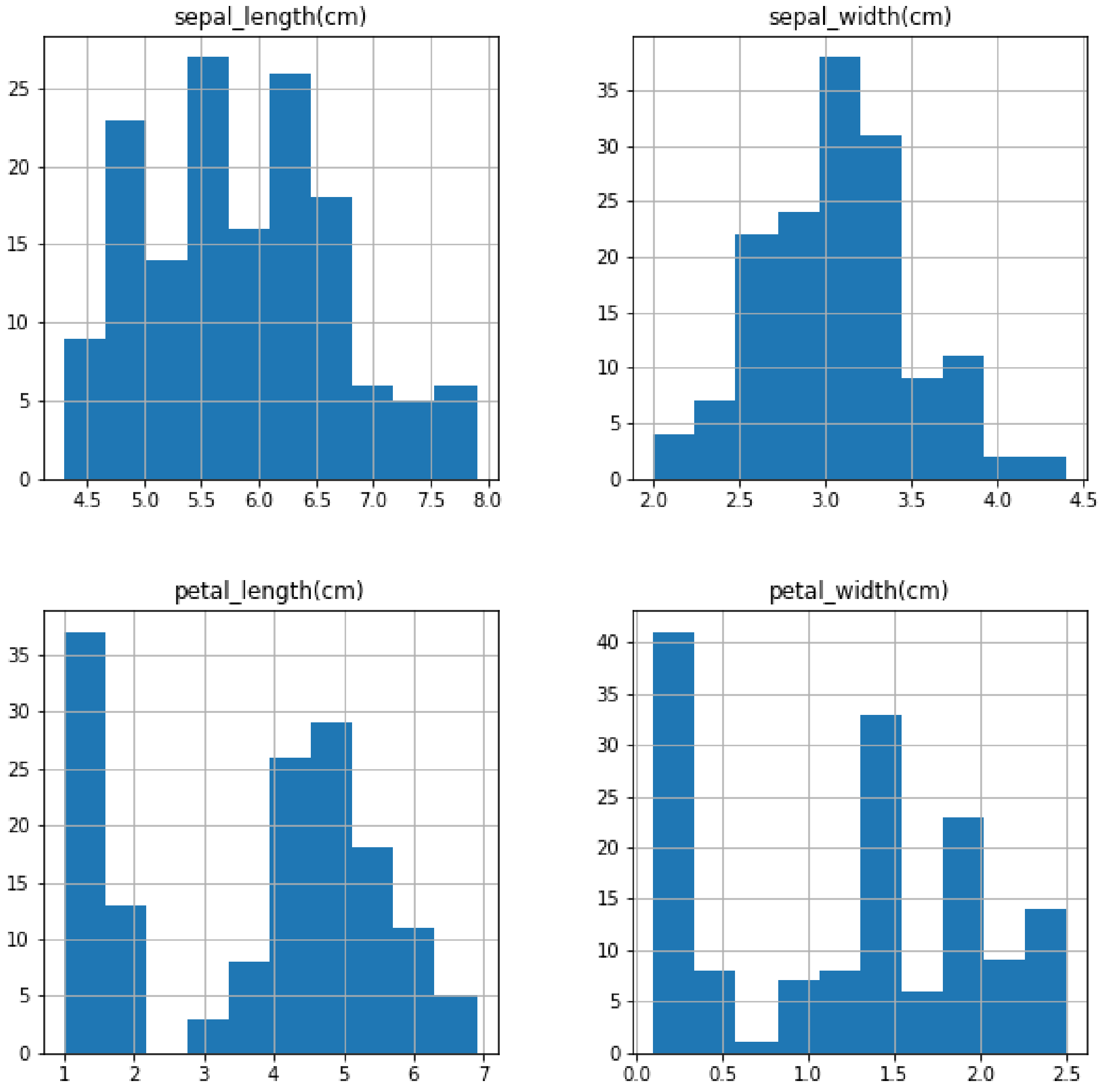
```
print("range   ", df['sepal_length(cm)'].max() - df['sepal_length(cm)'].min())
print("variance", df['sepal_length(cm)'].var())
print(df['sepal_length(cm)'].describe())
print("===========================================")
print("range   ", df['sepal_width(cm)'].max() - df['sepal_width(cm)'].min())
print("variance", df['sepal_width(cm)'].var())
print(df['sepal_width(cm)'].describe())
print("===========================================")
print("range   ", df['petal_length(cm)'].max() - df['petal_length(cm)'].min())
print("variance", df['petal_length(cm)'].var())
print(df['petal_length(cm)'].describe())
print("==========================================")
print("range   ", df['petal_width(cm)'].max() - df['petal_width(cm)'].min())
print("variance", df['petal_width(cm)'].var())
print(df['petal_width(cm)'].describe())
```

```
range    3.6000000000000005
variance 0.6856935123042507
count    150.000000
mean       5.843333
std        0.828066
min        4.300000
25%        5.100000
50%        5.800000
75%        6.400000
max        7.900000
Name: sepal_length(cm), dtype: float64
==========================================
range    2.4000000000000004
variance 0.1880040268456376
count    150.000000
mean       3.054000
std        0.433594
min        2.000000
25%        2.800000
50%        3.000000
75%        3.300000
max        4.400000
Name: sepal_width(cm), dtype: float64
==========================================
range    5.9
variance 3.113179418344519
count    150.000000
mean       3.758667
std        1.764420
min        1.000000
25%        1.600000
50%        4.350000
75%        5.100000
max        6.900000
Name: petal_length(cm), dtype: float64
==========================================
range    2.4
variance 0.582414317673378
count    150.000000
mean       1.198667
std        0.763161
```

```
min         0.100000
25%         0.300000
50%         1.300000
75%         1.800000
max         2.500000
Name: petal_width(cm), dtype: float64
```

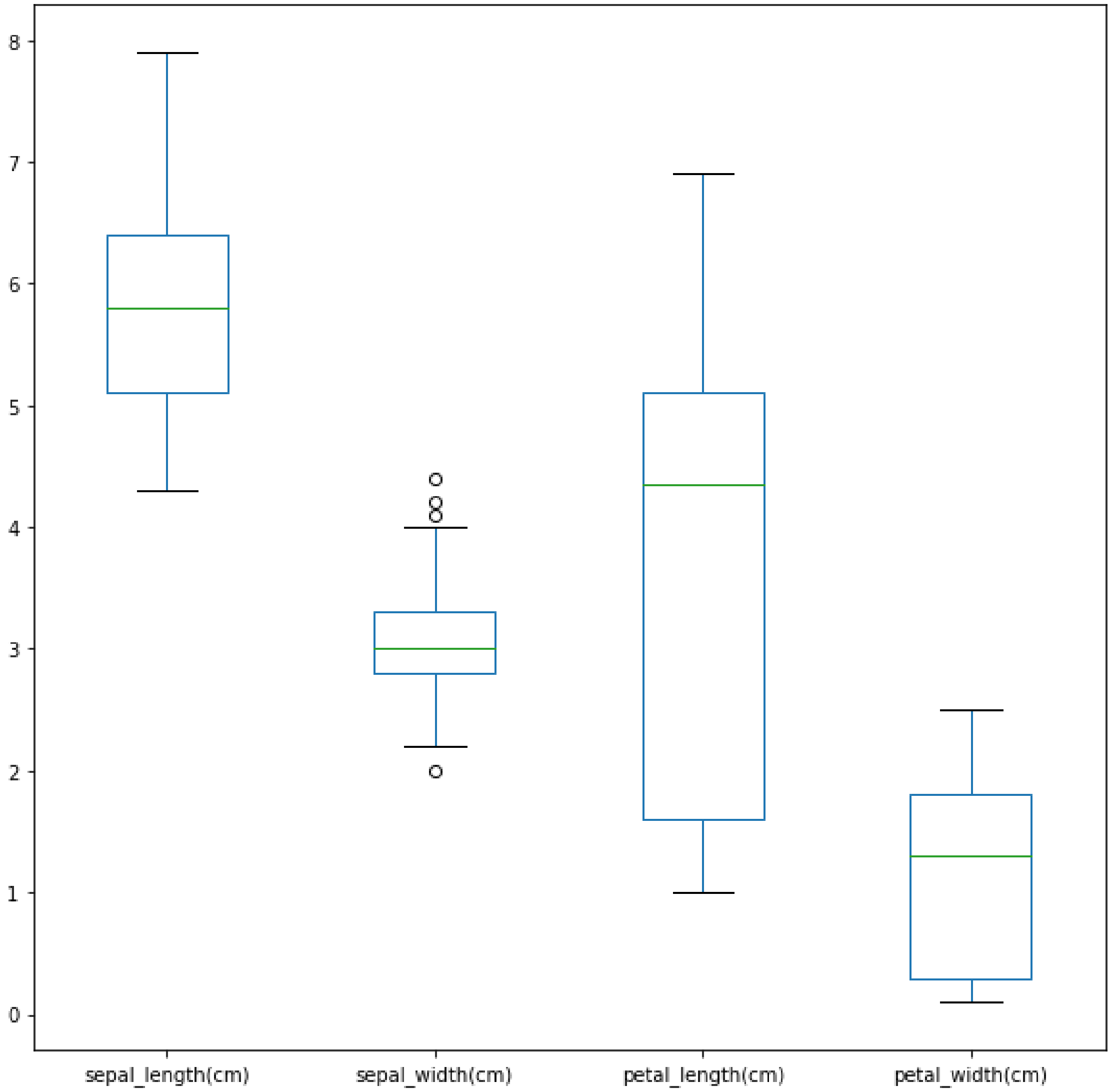
[9]:

```
# Histogram
fig = plt.figure(figsize = (10,10))
ax = fig.gca()
histogram = df.hist(ax = ax)
```



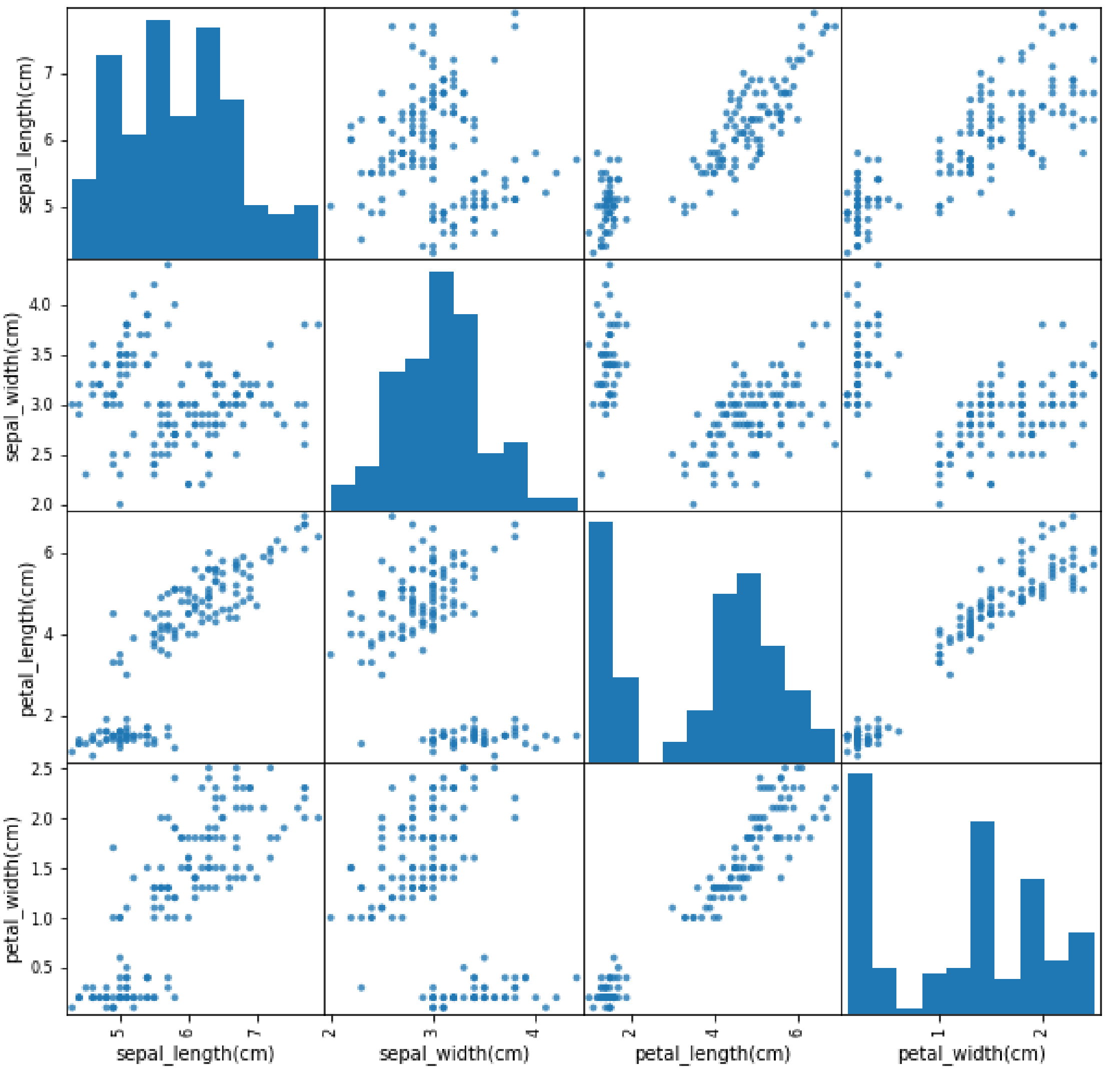
[39]:

```
# Box plots
fig = plt.figure(figsize = (10,10))
ax = fig.gca()
box_plot = df.boxplot(ax = ax, grid=False, return_type='axes')
```



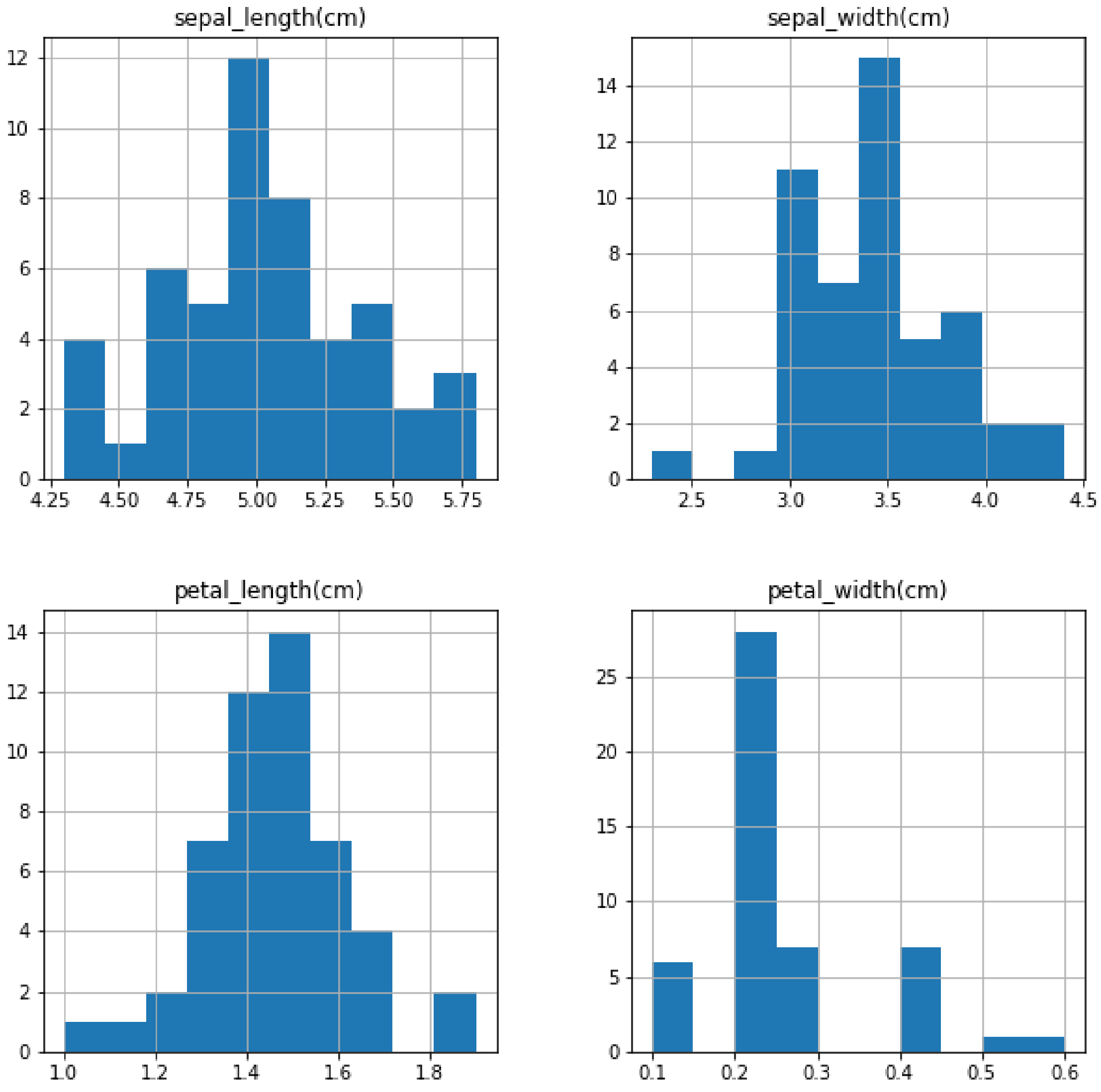
[7]:

```
# Pairwise Plots
pairwise_plots = pd.plotting.scatter_matrix(df, figsize=(10,10), marker = '.',␣
 ↪s = 60, alpha = 0.8)
```



[12]:
```
x = df[df['class'] == 'Iris-setosa']
print("SETOSA CLASS WISE VISUALIZATION")
setosa = x.hist(figsize=(10,10))
```

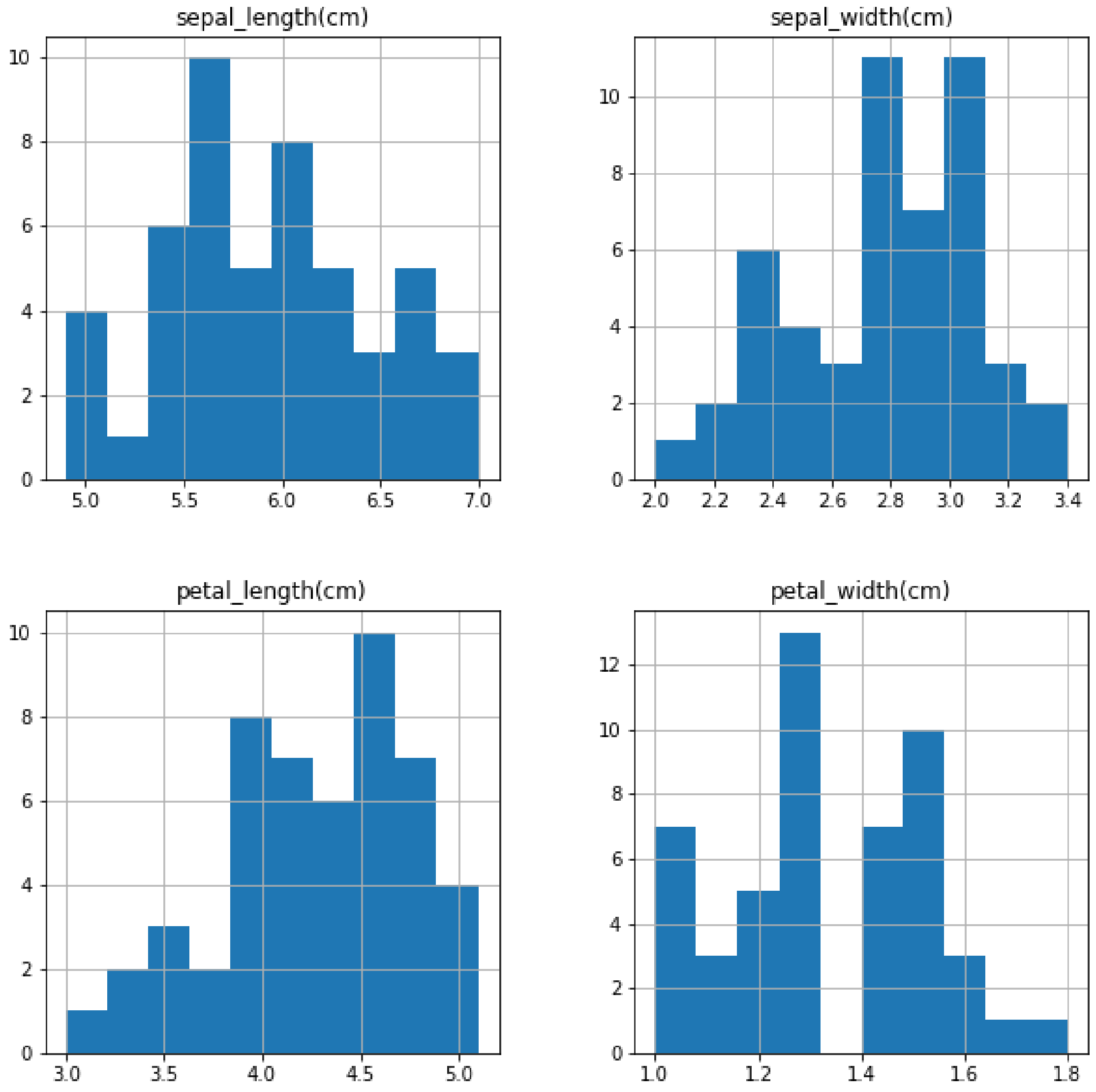
```
SETOSA CLASS WISE VISUALIZATION
```



[68]:

```
x = df[df['class'] == 'Iris-versicolor']
print("VERSICOLOR CLASS WISE VISUALIZATION")
setosa = x.hist(figsize=(10,10))
```

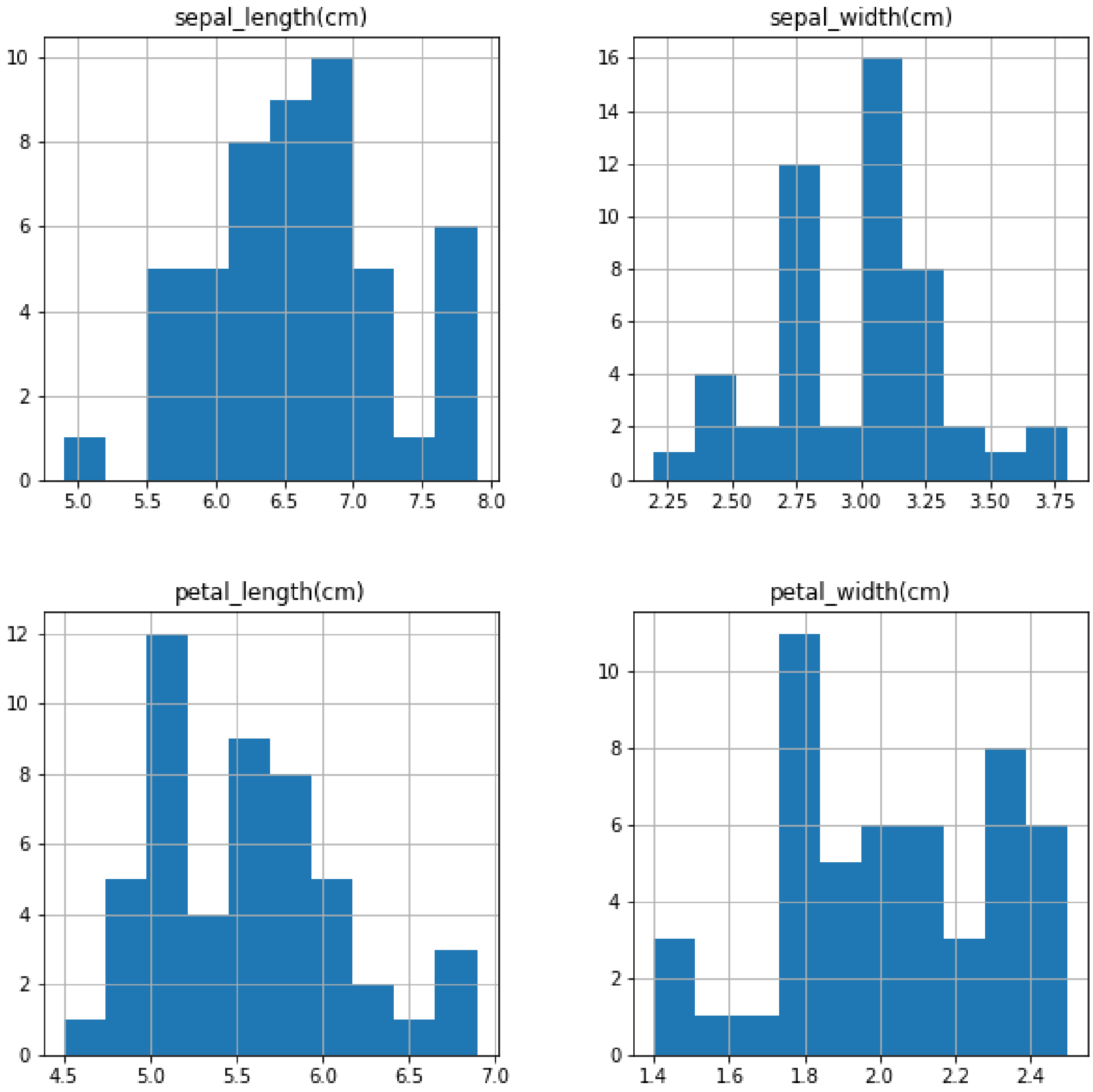
```
VERSICOLOR CLASS WISE VISUALIZATION
```



[69]:

```
x = df[df['class'] == 'Iris-virginica']
print("VIRGINICA CLASS WISE VISUALIZATION")
setosa = x.hist(figsize=(10,10))
```

```
VIRGINICA CLASS WISE VISUALIZATION
```



[15]:

```
#df.groupby(['class']).hist(figsize=(30,10))
```

1 1.3 - CONCEPTUAL QUESTIONS

1.0.1 1.

There are four features (Sepal Length (cm), Sepal Width (cm), Petal Length (cm), Petal Width (cm)). They are all numeric.

1.0.2 2.

From the initial glance we can see that there are considerable amount of flowers in every bin of sepal length and width. This means the data from this perspective is less grained. But when you look at petal length and width there are gaps which suggest that flowers with that petal length and width are rare to find.

In the histograms for whole data, the petal length histogram is divided into two parts. The first part ends at pental_length = 2.1 cms and the second part starts at petal_length ~ 2.8 cms. So, any value in this range can segment the histogram into two seperate parts.

1.0.3 3.

The pair of (sepal_length, petal_width) has the biggest parity in their medians. And, petal_length is the feature that explains the greatest amount of data.

1.0.4 4.

From the pairwise plot, the three pairs that have a strong correlation are (petal_length, petal_width), (petal_length, sepal_length), and (petal_width, sepal_length).

1.0.5 5.

IRIS SETOSA For Iris-Setosa class of flowers, the petal_width is the most differentiating factor, with all of them falling in the low ranges of the values when compared to other classes. This might be the reason for the low median score for petal_width which is represented in the box plot of the whole data.

IRIS VERSICOLOR For Iris-Versicolor class of flowers, while sepal_length and petal_width occupy the spots around the median, sepal_width and petal_length are mostly > median of the whole dataset.

IRIS VIRGINICA For Iris-Virginica class of flowers, the sepal_length is mostly >= median of the whole dataset, while the rest of the features are mostly populated around the median.

[3]:

```
import nbconvert
```