Python Code for Chapter 3 of Introduction to Data Mining

1 Basics Statistics

Import Iris dataset and neccessary packages.

```
import pandas as pd
from pandas import Series,DataFrame
import numpy as np
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
import seaborn as sns
```

Load Iris dataset and store it into a dateframe data structure.

```
#load the iris data
data=load_iris()
data.target.shape=(len(data.data),1)
#concatenate the target column and the feature columns
new_data=np.concatenate((data.data,data.target),axis=1)
iris=pd.DataFrame(new_data,columns=
['sepal_length','sepal_width','petal_length','petal_width','target'])
```

Check its basic statistics.

```
print(iris.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
sepal_length    150 non-null float64
sepal_width     150 non-null float64
petal_length    150 non-null float64
petal_width     150 non-null float64
target          150 non-null float64
dtypes: float64(5)
memory usage: 5.9 KB
None
```

Note that there are no missing (NA) values.

Find Mean, Standard Deviation, minimum and maximum values of each column (exclusive of the last categorical column).

```
#Mean,SD,median,min and max to each column.
iris_mean=iris.ix[:,0:4].dropna().mean(axis=0)
```

```
sepal_length    5.843333
sepal_width     3.054000
petal_length    3.758667
petal_width     1.198667
dtype: float64
```

```
iris_sd=iris.ix[:,0:4].dropna().std(axis=0)
```

```
sepal_length    0.828066
sepal_width     0.433594
petal_length    1.764420
petal_width     0.763161
dtype: float64
```

```
iris_median=iris.ix[:,0:4].dropna().median(axis=0)
```

```
sepal_length    5.80
sepal_width     3.00
petal_length    4.35
petal_width     1.30
dtype: float64
```

```
iris_min=iris.ix[:,0:4].dropna().max(axis=0)
```

```
sepal_length    7.9
sepal_width     4.4
petal_length    6.9
petal_width     2.5
dtype: float64
```

```
iris_max=iris.ix[:,0:4].dropna().min(axis=0)
```

```
sepal_length    4.3
sepal_width     2.0
petal_length    1.0
petal_width     0.1
dtype: float64
```

Correlation Matrix of Iris dataset.

```
iris_corr=iris.corr()
print(iris_corr)
```

	sepal_length	sepal_width	petal_length	petal_width	target
sepal_length	1.000000	-0.109369	0.871754	0.817954	0.782561
sepal_width	-0.109369	1.000000	-0.420516	-0.356544	-0.419446
petal_length	0.871754	-0.420516	1.000000	0.962757	0.949043
petal_width	0.817954	-0.356544	0.962757	1.000000	0.956464
target	0.782561	-0.419446	0.949043	0.956464	1.000000

Covariance Matrix of Iris dataset.

```
iris_quantile=iris.quantile([0.0,0.25,0.5,0.75,1.0])
print(iris_quantile)
```

	sepal_length	sepal_width	petal_length	petal_width	target
sepal_length	0.685694	-0.039268	1.273682	0.516904	0.530872
sepal_width	-0.039268	0.188004	-0.321713	-0.117981	-0.148993
petal_length	1.273682	-0.321713	3.113179	1.296387	1.371812
petal_width	0.516904	-0.117981	1.296387	0.582414	0.597987
target	0.530872	-0.148993	1.371812	0.597987	0.671141

Percentiles of Iris dataset.

```
iris_quantile=iris.quantile([0.0,0.25,0.5,0.75,1.0])
print(iris_quantile)
```

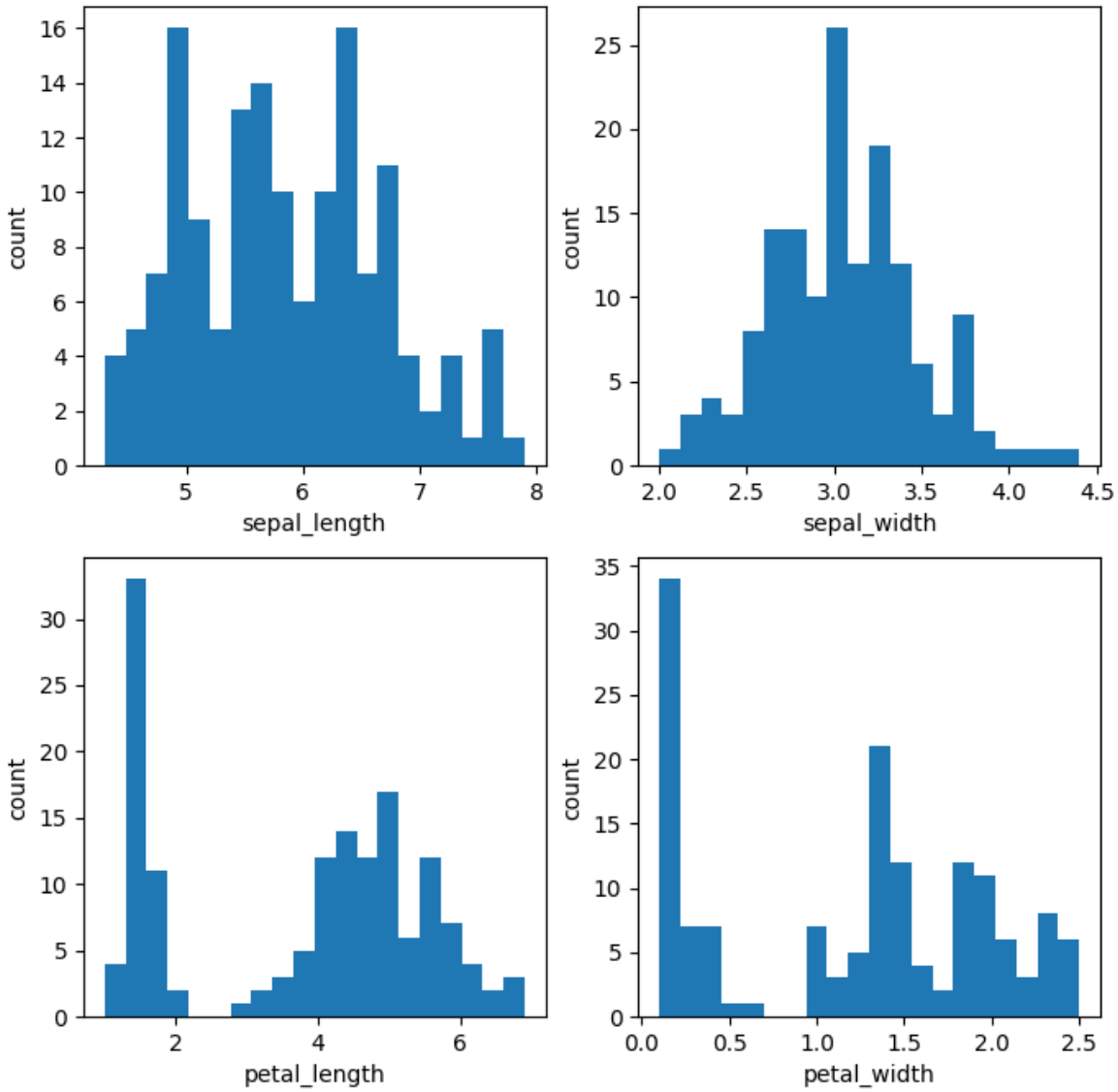
	sepal_length	sepal_width	petal_length	petal_width	target
0.00	4.3	2.0	1.00	0.1	0.0
0.25	5.1	2.8	1.60	0.3	0.0
0.50	5.8	3.0	4.35	1.3	1.0
0.75	6.4	3.3	5.10	1.8	2.0
1.00	7.9	4.4	6.90	2.5	2.0

2 Visualizations

The histogram of the first three features of Iris.

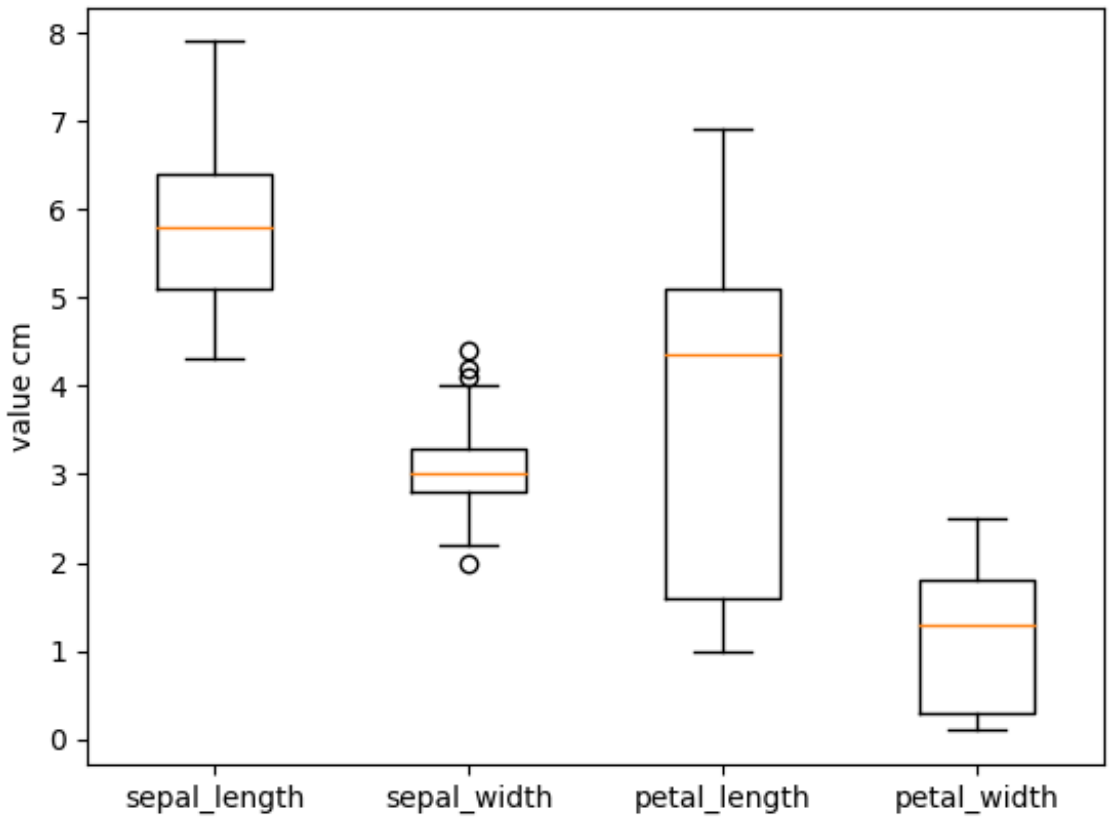
```
fig=plt.figure(figsize=(8,8))

ax1=fig.add_subplot(2,2,1)
```

```
sepal_length=iris.sepal_length
plt.hist(sepal_length,bins=20)
plt.xlabel('sepal_length')
plt.ylabel('count')

sepal_width=iris.sepal_width
ax2=fig.add_subplot(2,2,2)
plt.hist(sepal_width,bins=20)
plt.xlabel('sepal_width')
plt.ylabel('count')

petal_length=iris.petal_length
ax3=fig.add_subplot(2,2,3)
plt.hist(petal_length,bins=20)
plt.xlabel('petal_length')
plt.ylabel('count')

petal_width=iris.petal_width
ax4=fig.add_subplot(2,2,4)
plt.hist(petal_width,bins=20)
plt.xlabel('petal_width')
plt.ylabel('count')
plt.savefig('histogram')
plt.show()
```



The box plot of Iris.

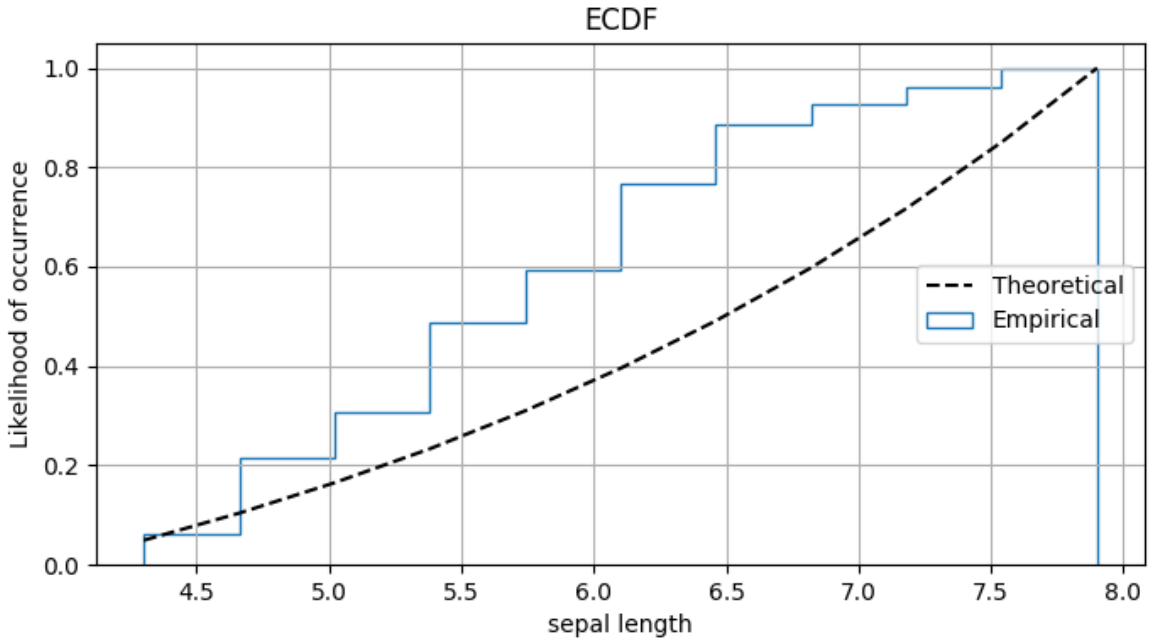
```
fig=plt.figure()
ax=fig.add_subplot(111)
data=[sepal_length,sepal_width,petal_length,petal_width]
ax.boxplot(data)
ax.set_xticklabels(['sepal_length', 'sepal_width', 'petal_length',
'petal_width'])
ax.set_ylabel('value cm')
plt.show()
```



The Empirical Cumulative Distribution Function, ECDF.

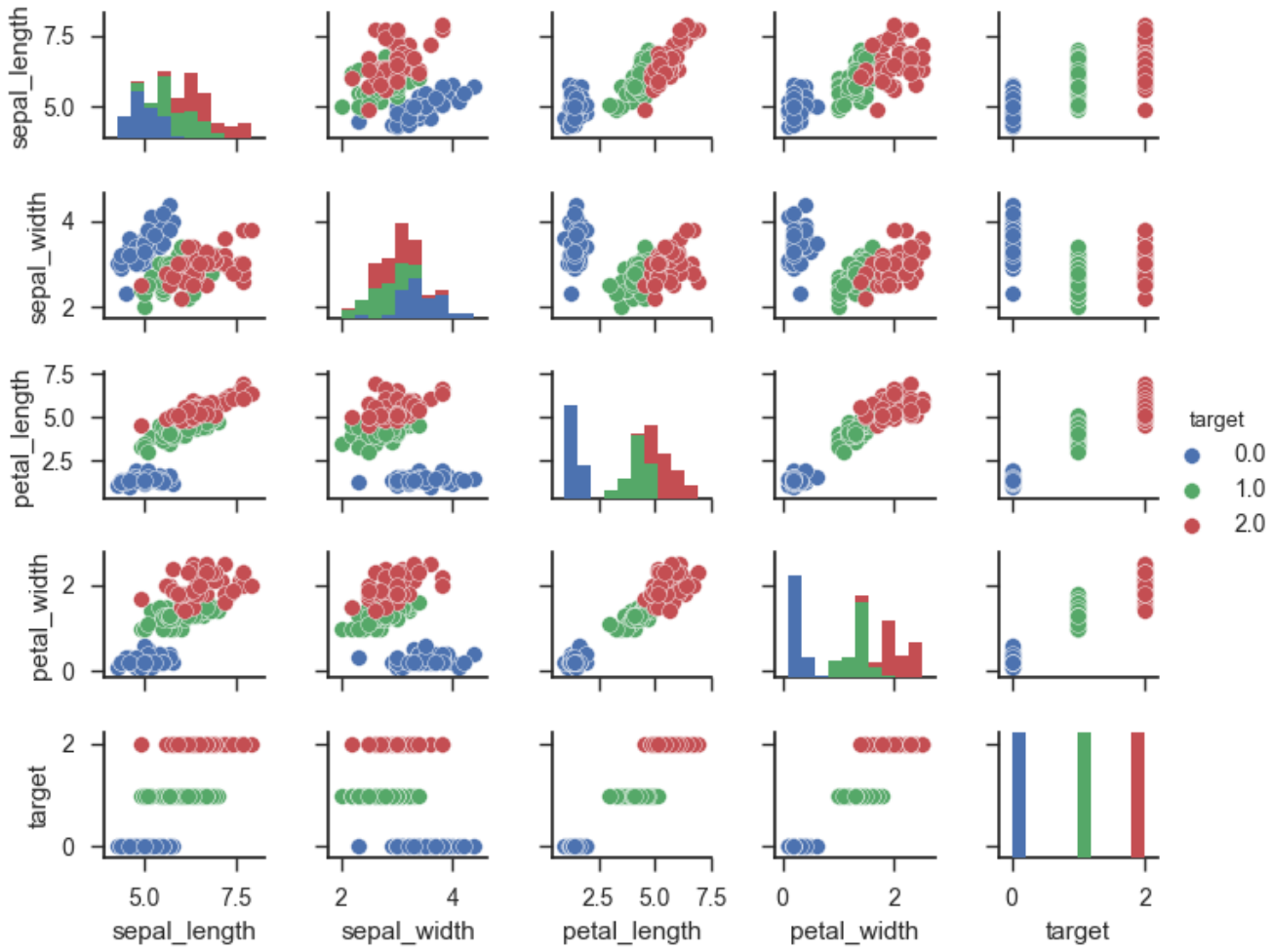
```
from matplotlib import mlab
fig,ax=plt.subplots(figsize=(8,4))
mu=200
sigma=25
n_bins=20
#plot the cumulative histogram
n,bins,patches=ax.hist(iris.sepal_length,normed=1,histtype='step',cumul
ative=True,label='Empirical')
#Add a line showing the expected distribution
y=mlab.normpdf(bins,mu,sigma).cumsum()
y/=y[-1]
ax.plot(bins,y,'k--',linewidth=1.5,label='Theoretical')

ax.grid(True)
ax.legend(loc='right')
ax.set_title('ECDF')
ax.set_xlabel('sepal length')
```

```
17.  ax.set_ylabel('Likelihood of occurrence')
18.  plt.show()
```



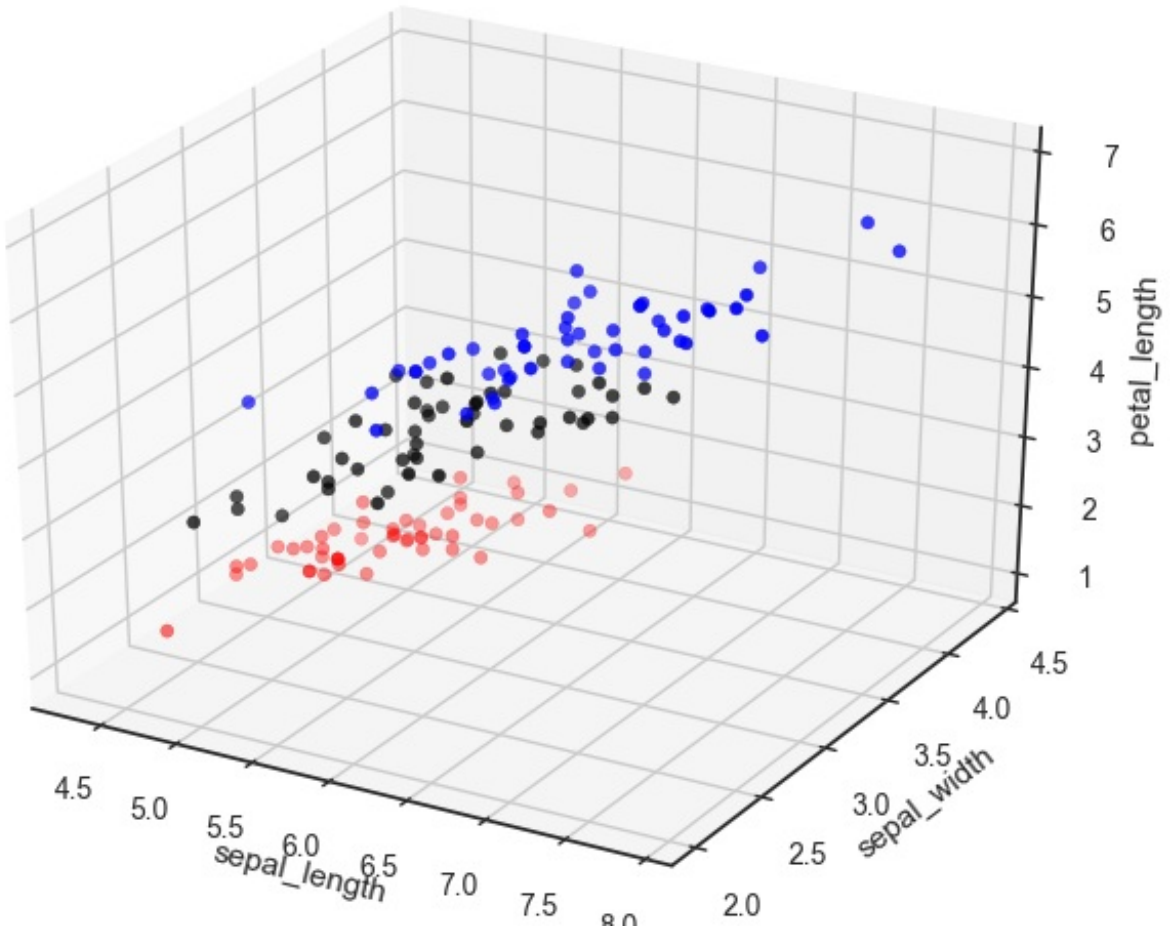
The scatter plot of Iris datset.

```
1.  sns.set(style="ticks")
2.  fig, ax = plt.subplots()
3.  sns.pairplot(iris,hue="target",size=1.2,aspect=1.2)
4.  plt.savefig("Scatter Matrix.png")
5.  plt.show()
```



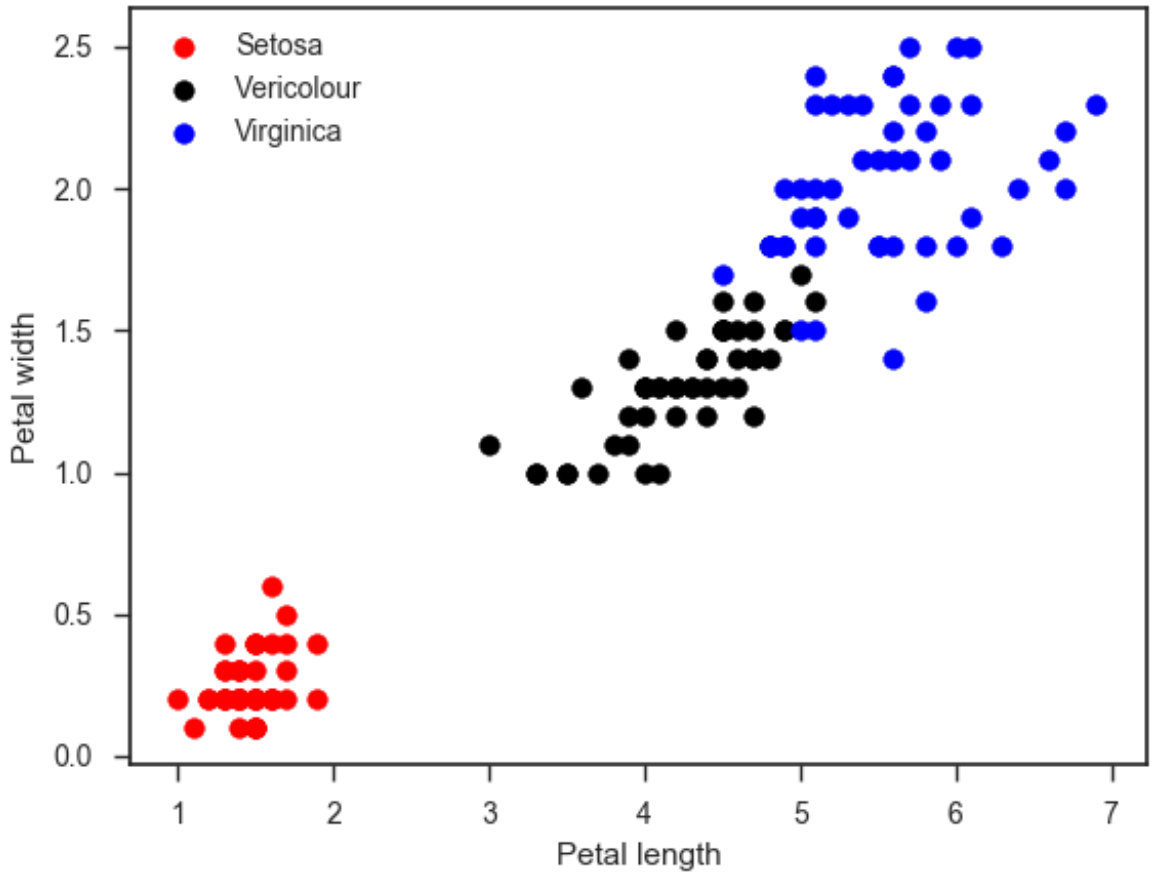
Plot 3D scatter plot for the first three features.

```
from mpl_toolkits.mplot3d import Axes3D
fig=plt.figure()
ax=Axes3D(fig)
colors=['red','k','blue']
x_vals=iris.sepal_length; y_vals=iris.sepal_width;
z_vals=iris.petal_length
ax.scatter(x_vals,y_vals,z_vals,c=iris.target.apply(lambda x: colors[in
t(x)]))
ax.set_xlabel('sepal_length')
ax.set_ylabel('sepal_width')
ax.set_zlabel('petal_length')
plt.show()
```



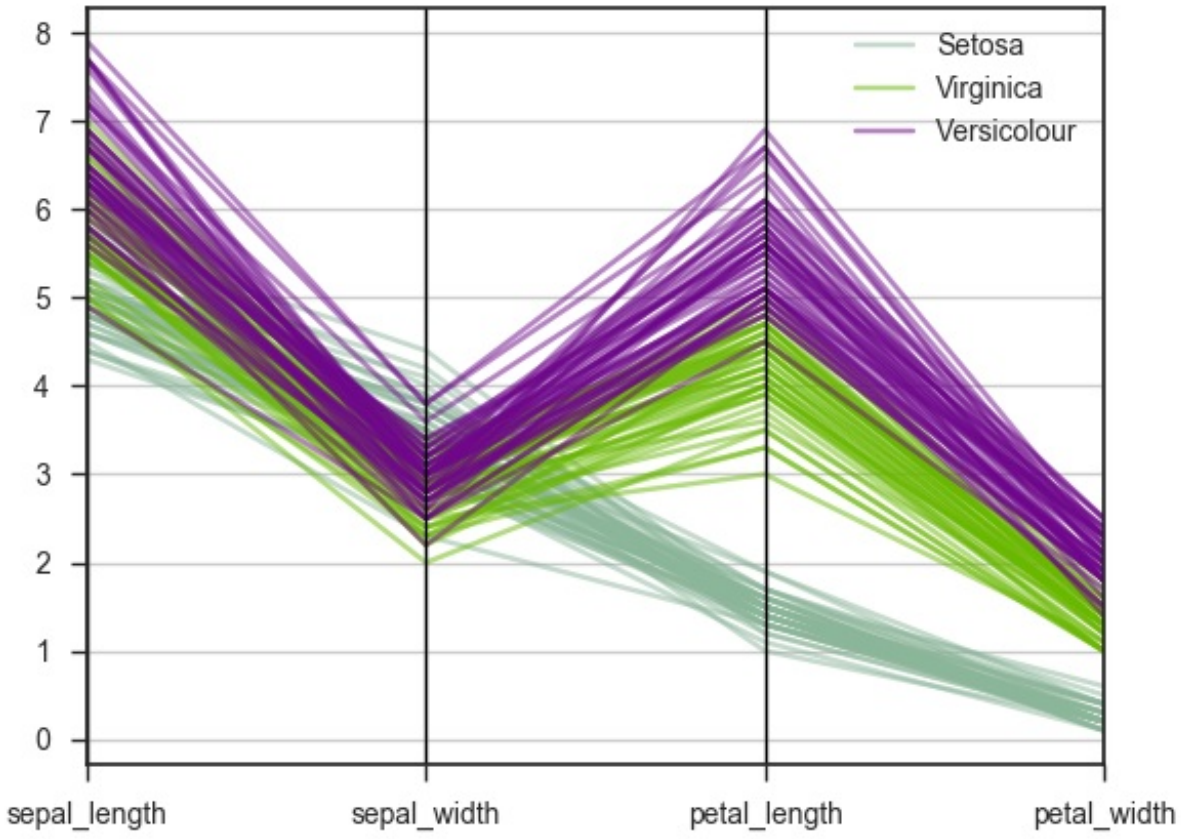
Scatter plot of petal_length and petal_width.

```
fig=plt.figure()
ax_scatter=fig.add_subplot(111)
data_0=iris.ix[iris.target==0,['petal_length','petal_width']]
data_1=iris.ix[iris.target==1,['petal_length','petal_width']]
data_2=iris.ix[iris.target==2,['petal_length','petal_width']]

ax_scatter.scatter(data_0.petal_length,data_0.petal_width,c='red',label
='Setosa')
ax_scatter.scatter(data_1.petal_length,data_1.petal_width,c='k',label='
Vericolour')
ax_scatter.scatter(data_2.petal_length,data_2.petal_width,c='blue',labe
l='Virginica')
ax_scatter.set_xlabel('Petal length')
ax_scatter.set_ylabel('Petal width')
ax_scatter.legend(loc='best')
plt.show()
```



Parallel coordinates.

```
from pandas.plotting import parallel_coordinates
#Replace the values 0,1 and 2 in column 'target' by their
corresponding flower's names
mapping={0:'Setosa',1:'Virginica',2:'Versicolour'}
iris_new=iris.copy()
iris_new.target=iris_new.target.apply(lambda x: mapping[int(x)])

fig=plt.figure()
parallel_coordinates(iris_new,'target',alpha=0.5)
plt.show()
```


8
7
6
5
4
3
2
1
0
Setosa
Virginica
Versicolour
sepal_length
sepal_width
petal_length
petal_width