Data_Analysis

October 5, 2022

1 Exploring the Iris data set

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

df=sns.load_dataset('iris')
df.head()
```

[1]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

[2]:
```
df.mean()
```

[2]:
```
sepal_length    5.843333
sepal_width     3.057333
petal_length    3.758000
petal_width     1.199333
dtype: float64
```

[3]:
```
df.describe()
```

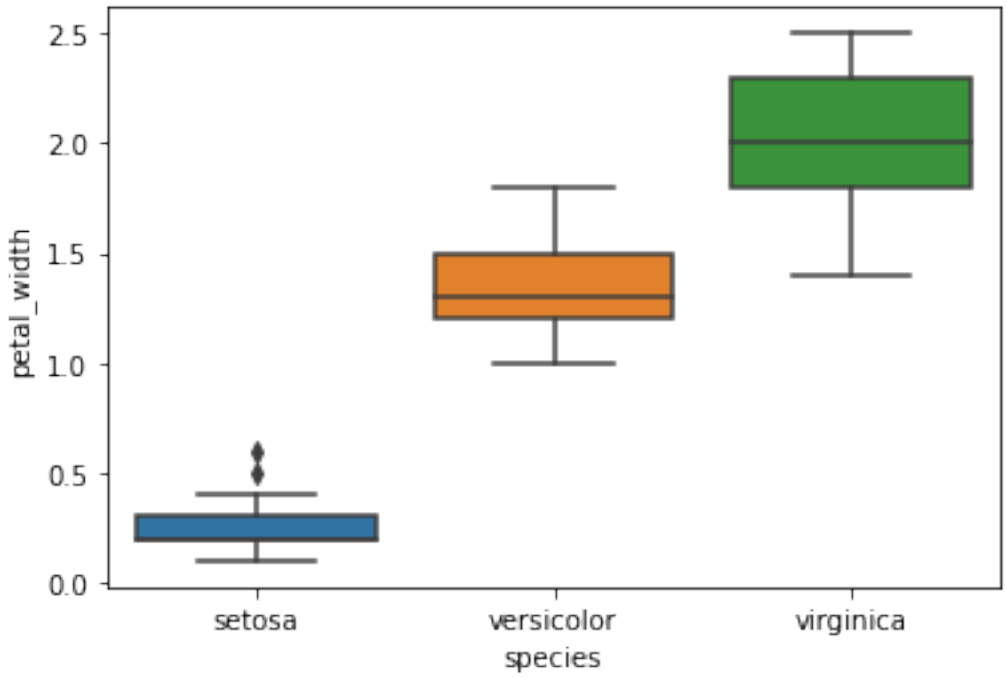
[3]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
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

2 Some exploratory plots!

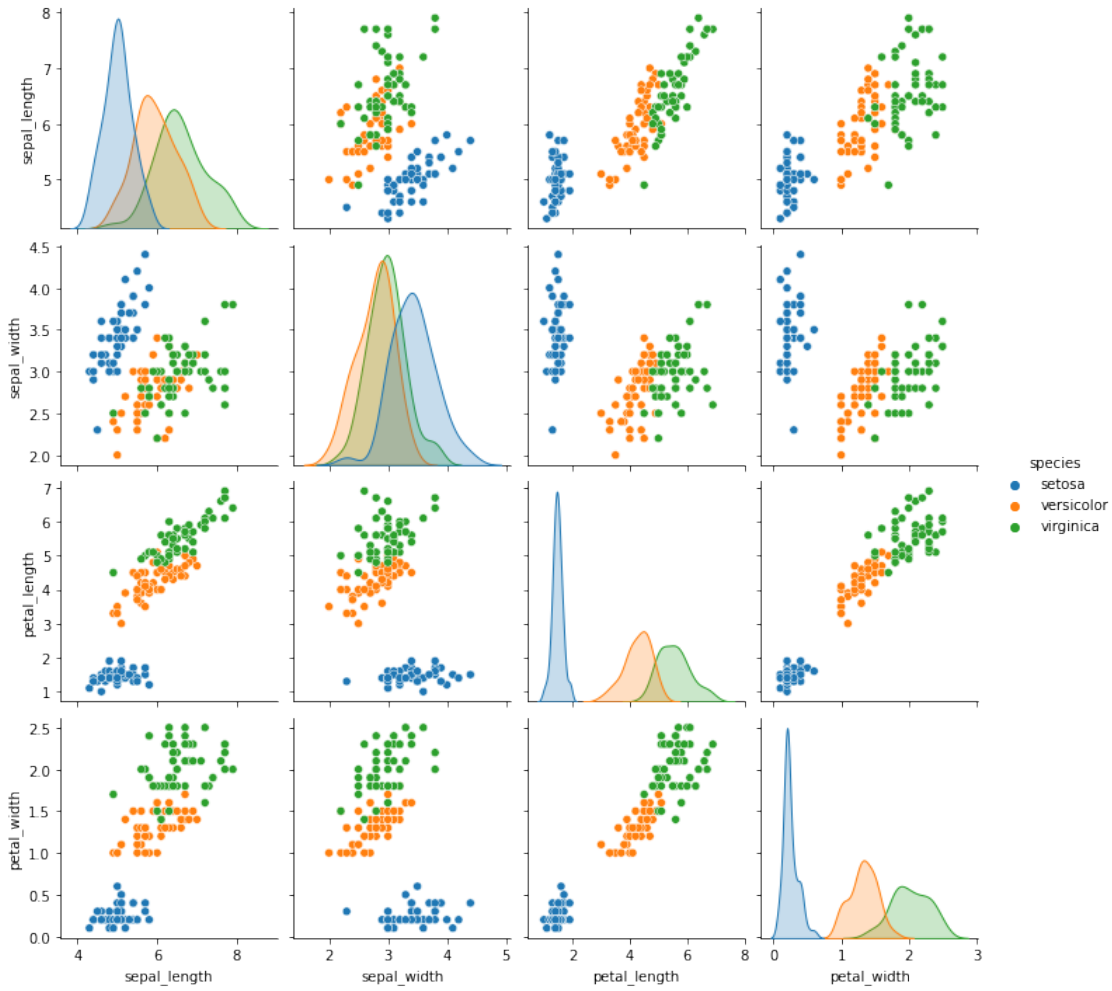
[4]:
```
sns.boxplot(data=df, x='species', y='petal_width')
```

[4]:
```
<AxesSubplot:xlabel='species', ylabel='petal_width'>
```



[5]:
```
sns.pairplot(data=df, hue='species')
```

[5]:
```
<seaborn.axisgrid.PairGrid at 0x7f90403987c0>
```



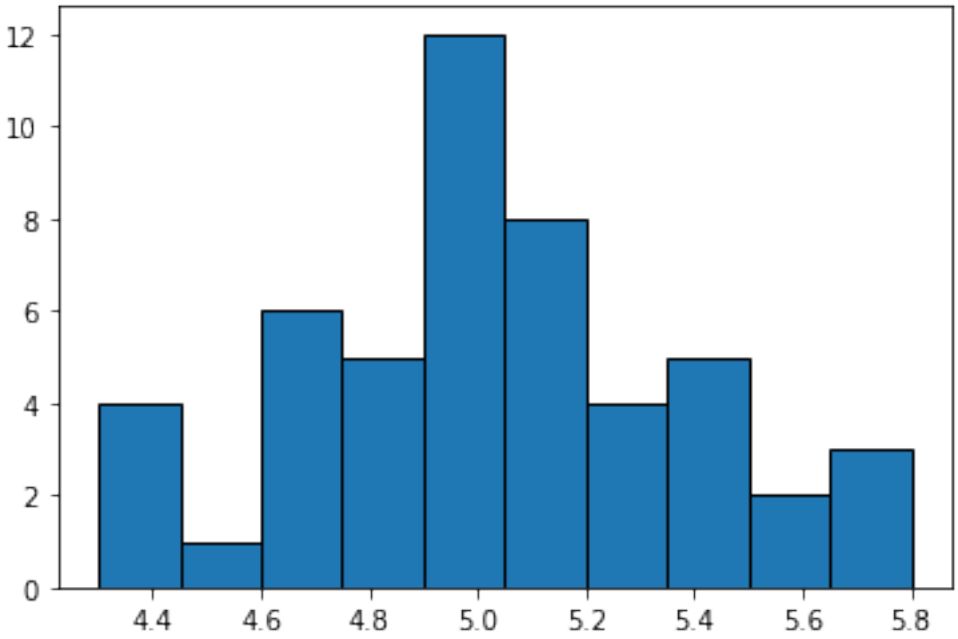
3 Statistical testing

[6]:
```
from scipy import stats
setosa = df['species'] == 'setosa'
setosa_sepal_length = df[setosa]['sepal_length']

stats.shapiro(setosa_sepal_length)
```

[6]:
```
ShapiroResult(statistic=0.9776989221572876, pvalue=0.4595281183719635)
```

[7]:
```
plt.hist(setosa_sepal_length, edgecolor='k')
```

[7]:
```
(array([ 4.,  1.,  6.,  5., 12.,  8.,  4.,  5.,  2.,  3.]),
 array([4.3 , 4.45, 4.6 , 4.75, 4.9 , 5.05, 5.2 , 5.35, 5.5 , 5.65, 5.8 ]),
 <BarContainer object of 10 artists>)
```



[8]:
```
from scipy import stats
setosa = df['species'] == 'setosa'
virginica = df['species'] == 'virginica'
setosa_sepal_length = df[setosa]['sepal_length']
virginica_sepal_length = df[virginica]['sepal_length']
stats.ttest_ind(setosa_sepal_length, virginica_sepal_length, equal_var=False)
```

[8]:
```
Ttest_indResult(statistic=-15.386195820079404, pvalue=3.9668672709859296e-25)
```

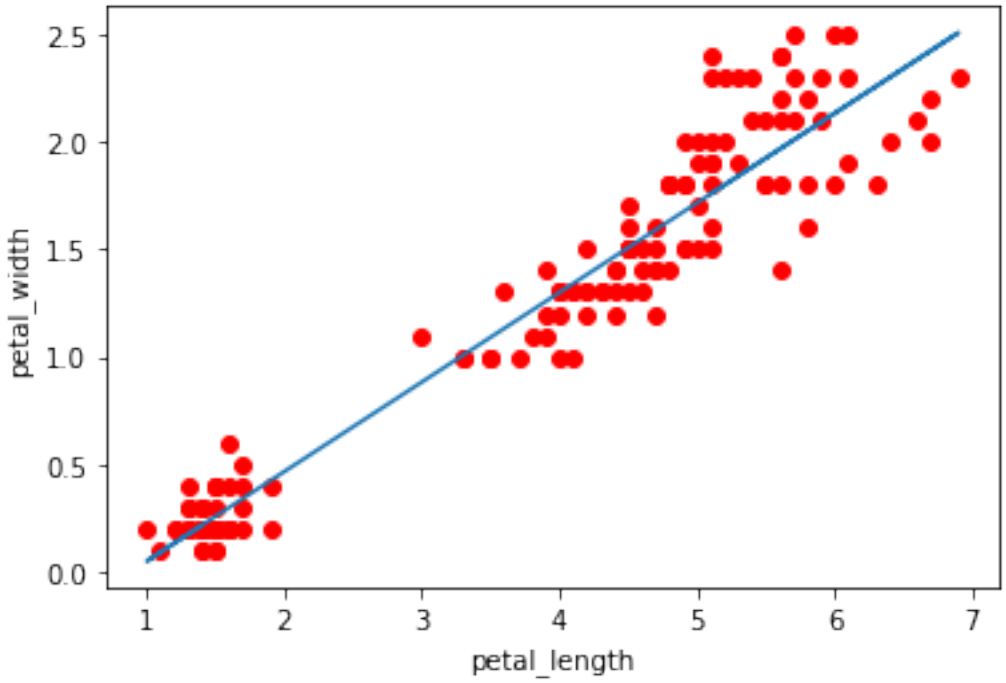
4 Modelling and fitting to a curve

[9]:
```
from scipy.optimize import curve_fit

def linear(x,m,b):
  return m*x+b

xdata = df['petal_length']
ydata = df['petal_width']

pars, cov = curve_fit(f=linear, xdata=xdata, ydata=ydata)
```

[10]:
```
Y_pred = [ ]

for x in xdata:
  Y_pred.append(linear(x, pars[0], pars[1]))
```

```
plt.scatter(xdata, ydata, c='r')
plt.plot(xdata, Y_pred)
plt.xlabel('petal_length')
plt.ylabel('petal_width')
```

[10]: Text(0, 0.5, 'petal_width')



[11]:
```
residuals = ydata- linear(xdata, pars[0], pars[1])
ss_res = np.sum(residuals**2)
ss_tot = np.sum((ydata-np.mean(ydata))**2)
r_squared = 1 - (ss_res / ss_tot)
print(r_squared)
```

```
0.9271098389904927
```