

inclass_plotting

October 20, 2022

```
[2]: import matplotlib.pyplot as plt
import pandas as pd
```

```
[5]: # read in data
iris = pd.read_csv('iris.data', header=None)
iris.columns =
    ↳ ['sepal_width', 'sepal_length', 'petal_width', 'petal_length', 'species']
print(iris)
```

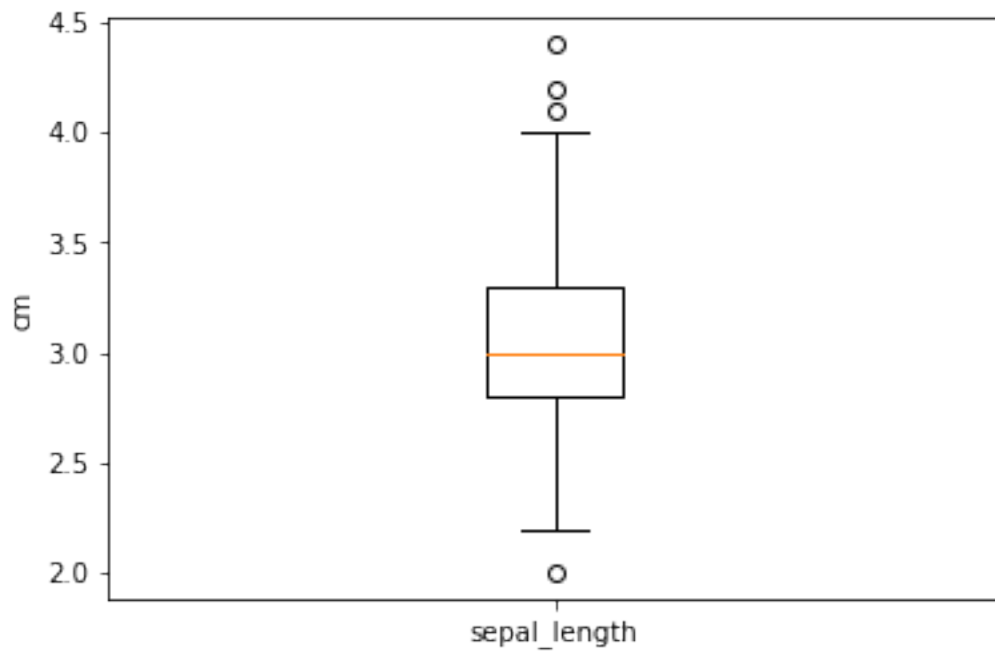
	sepal_width	sepal_length	petal_width	petal_length	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
..
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

[150 rows x 5 columns]

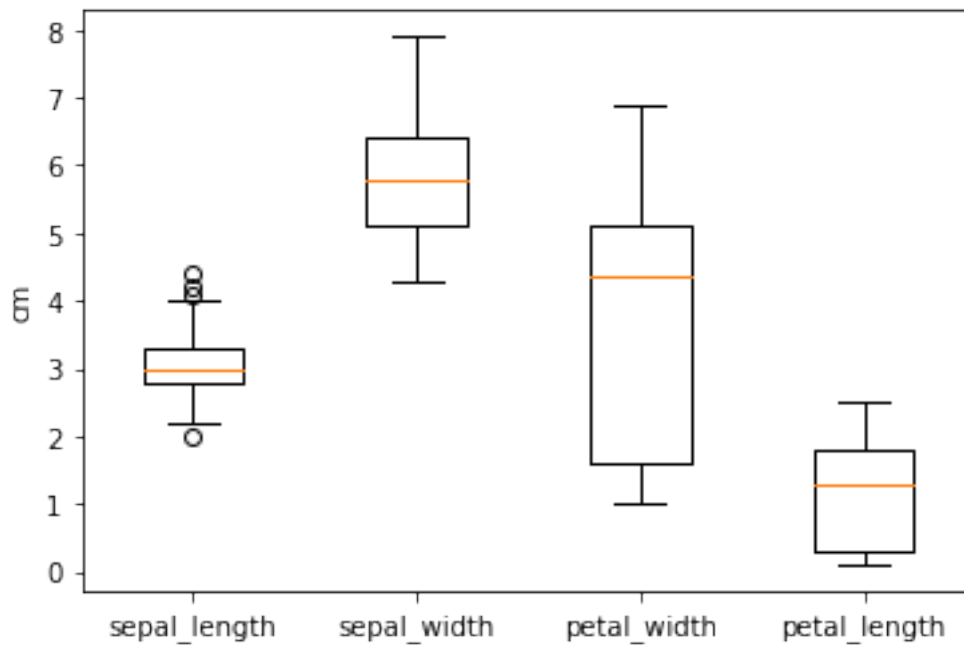
```
[11]: # print(iris['petal_width'])
# print(iris[['petal_width', 'species']])

# iris[iris['species'] == 'Iris-virginica']
```

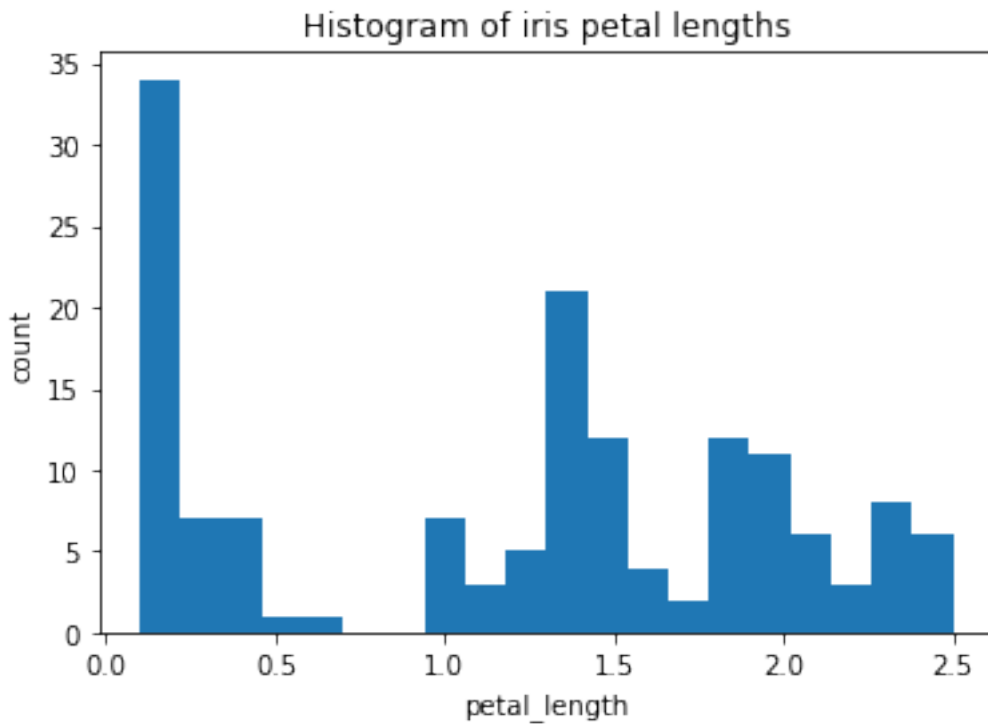
```
[15]: plt.boxplot(iris['sepal_length'], labels=['sepal_length'])
plt.ylabel('cm')
plt.show()
```



```
[16]: measurement_names = ['sepal_length', 'sepal_width', 'petal_width', 'petal_length']
plt.boxplot(iris[measurement_names], labels=measurement_names)
plt.ylabel('cm')
plt.show()
```

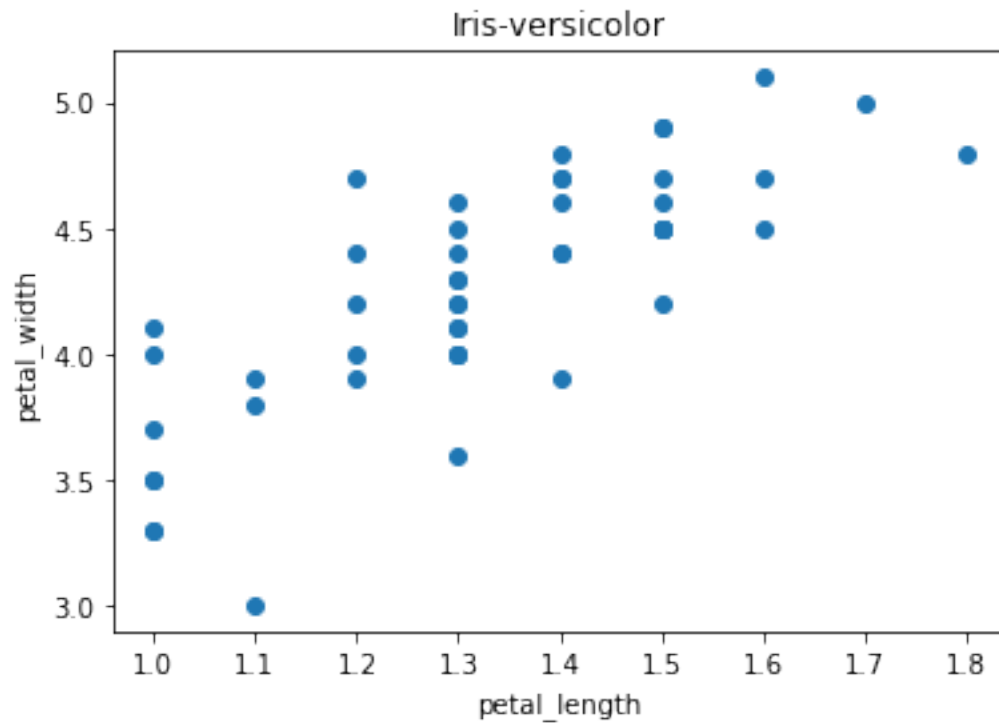


```
[22]: plt.hist(iris['petal_length'],bins=20)
plt.ylabel('count')
plt.xlabel('petal_length')
plt.title('Histogram of iris petal lengths')
plt.show()
```

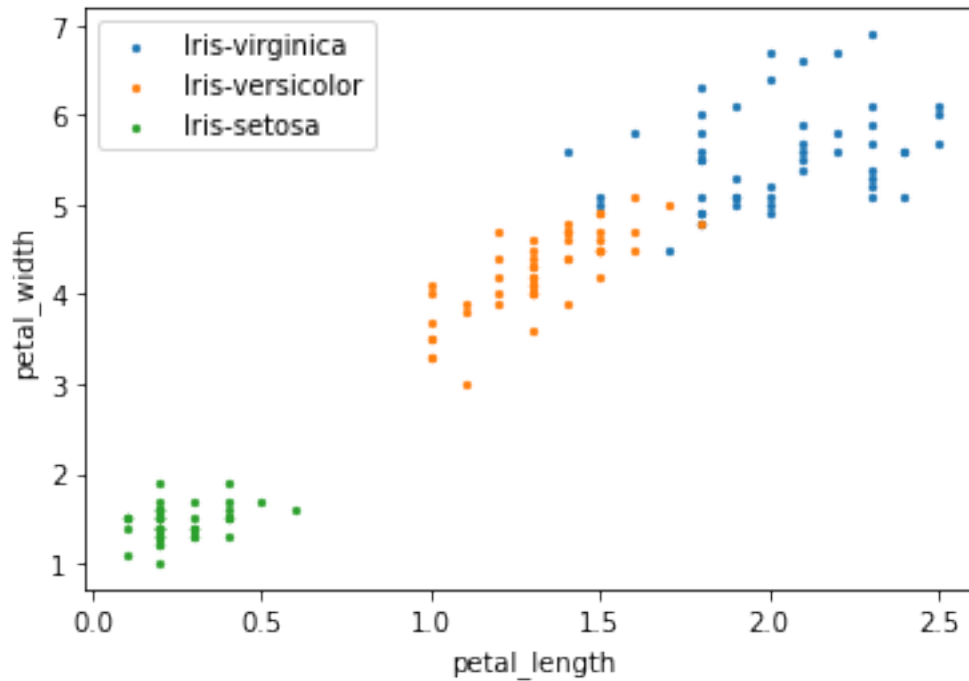


```
[31]: # create a dataframe with just the rows related to Iris-versicolor
iris['species'].unique()
versicolor = iris[iris['species'] == 'Iris-versicolor']

plt.scatter(versicolor['petal_length'],versicolor['petal_width'])
plt.ylabel('petal_width')
plt.xlabel('petal_length')
plt.title('Iris-versicolor')
plt.show()
```



```
[43]: for species_name in set(iris['species']):
        iris_subset = iris[iris['species'] == species_name]
        plt.scatter(iris_subset['petal_length'], iris_subset['petal_width'],
                    label=species_name, s=5)
plt.legend()
plt.xlabel('petal_length')
plt.ylabel('petal_width')
plt.show()
```

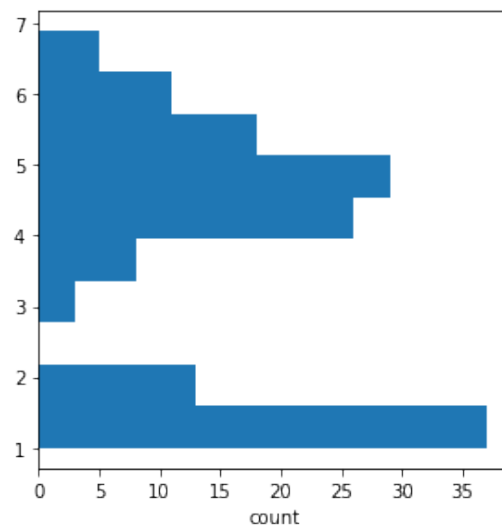
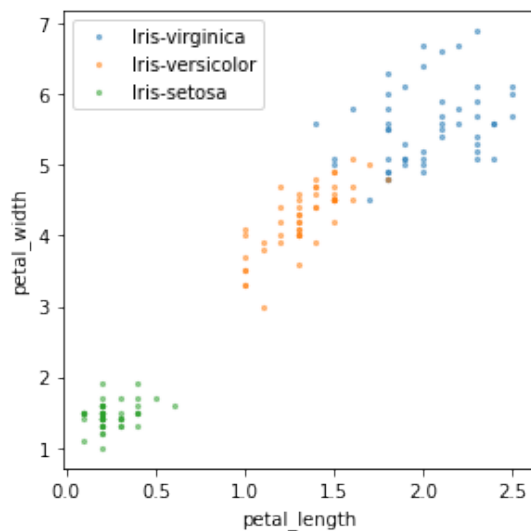
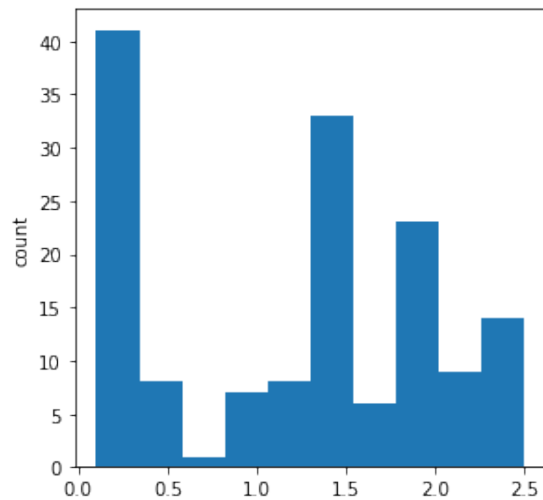


```
[54]: fig, axes = plt.subplots(2,2)
fig.set_size_inches(10,10)
fig.delaxes(axes[0,1])

print(axes.shape)
for species_name in set(iris['species']):
    iris_subset = iris[iris['species'] == species_name]
    axes[1,0].scatter(iris_subset['petal_length'], iris_subset['petal_width'],
        label=species_name, s=5, alpha=.5)
axes[1,0].legend()
axes[1,0].set_xlabel('petal_length')
axes[1,0].set_ylabel('petal_width')

axes[0,0].hist(iris['petal_length'])
axes[0,0].set_ylabel('count')
axes[1,1].hist(iris['petal_width'], orientation='horizontal')
axes[1,1].set_xlabel('count')
plt.show()
```

(2, 2)



```
[64]: fig, axes = plt.subplots(2,2)
fig.set_size_inches(10,10)

# top left scatter plot
axes[0,0].scatter(versicolor['petal_length'],versicolor['petal_width'])
axes[0,0].set_ylabel('petal_width')
axes[0,0].set_xlabel('petal_length')
axes[0,0].set_title('Iris-versicolor')

# top right boxplot
axes[0,1].boxplot(iris[measurement_names],labels=measurement_names)
axes[0,1].set_ylabel('cm')

# bottom left
```

```

axes[1,0].hist(iris['sepal_length'],bins=20)
axes[1,0].set_ylabel('count')
axes[1,0].set_xlabel('sepal_length')

# bottom right
for species_name in set(iris['species']):
    iris_subset = iris[iris['species'] == species_name]
    axes[1,1].scatter(iris_subset['petal_length'], iris_subset['petal_width'],
        label=species_name, s=5, alpha=.5)
axes[1,1].legend()
axes[1,1].set_xlabel('petal_length')
axes[1,1].set_ylabel('petal_width')

# BONUS CODE!!!
# Remove top and right borders from each plot
# for each row
for i in range(2):
    # for each column
    for j in range(2):
        # choose to hide or show certain borders or "spines"
        axes[i,j].spines['top'].set_visible(False)
        axes[i,j].spines['right'].set_visible(False)
        axes[i,j].spines['bottom'].set_visible(True)
        axes[i,j].spines['left'].set_visible(True)

plt.savefig('my_beautiful_plot.png')
plt.show()

```

