Day7Matplotlib

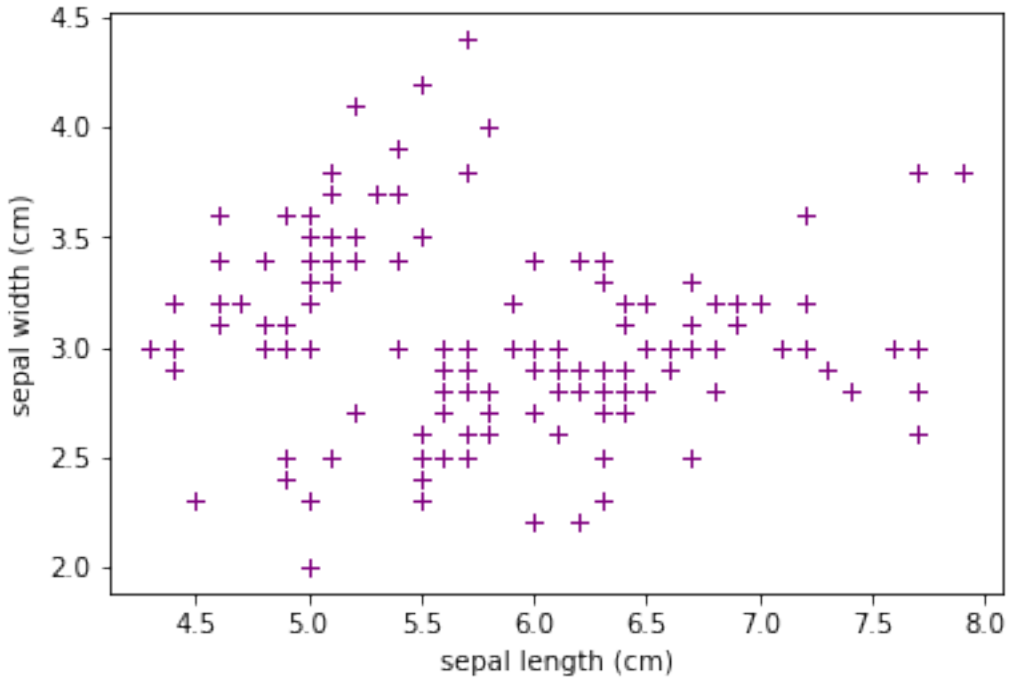
February 25, 2021

Scatter plot using plt.plot

[25]:
```
%matplotlib inline
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
iris = load_iris()
features = iris.data.T
%timeit plt.plot(features[0],features[1], '+', color='purple')
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
```

```
1.06 ms ± 143 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

[25]:
```
Text(0, 0.5, 'sepal width (cm)')
```



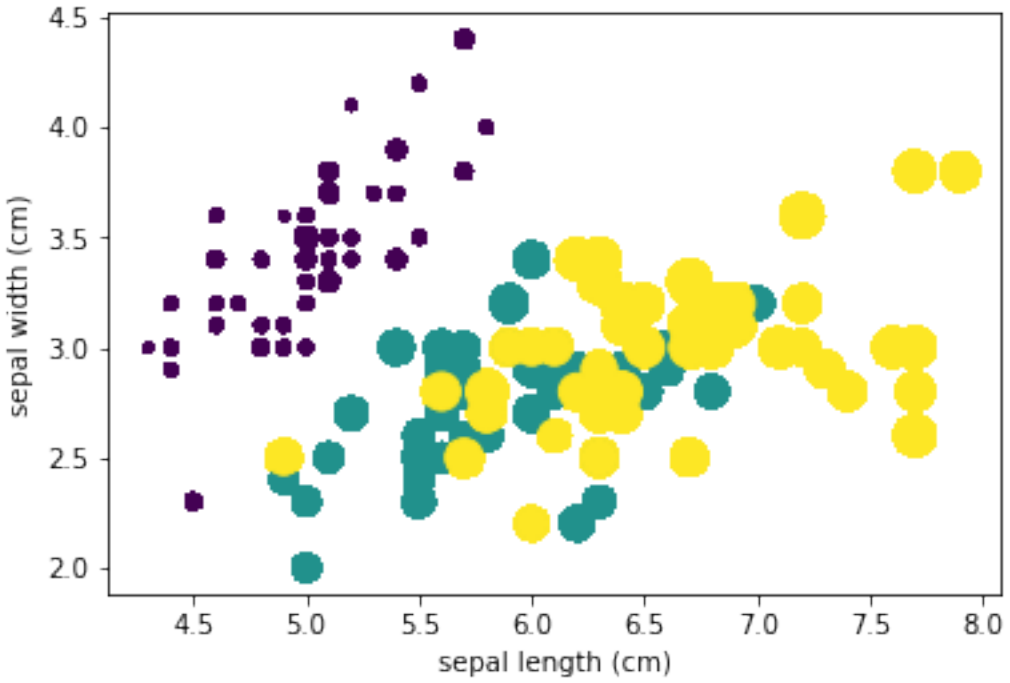
Scatter plot (bubble plot) using plt.scatter Notice the time increase. This plot is not recommended

for large datasets.

[27]:
```
%timeit plt.scatter(features[0],features[1], c = iris.target, \
                    s = 100*features[3], cmap='viridis')
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
```

```
5.36 ms ± 646 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

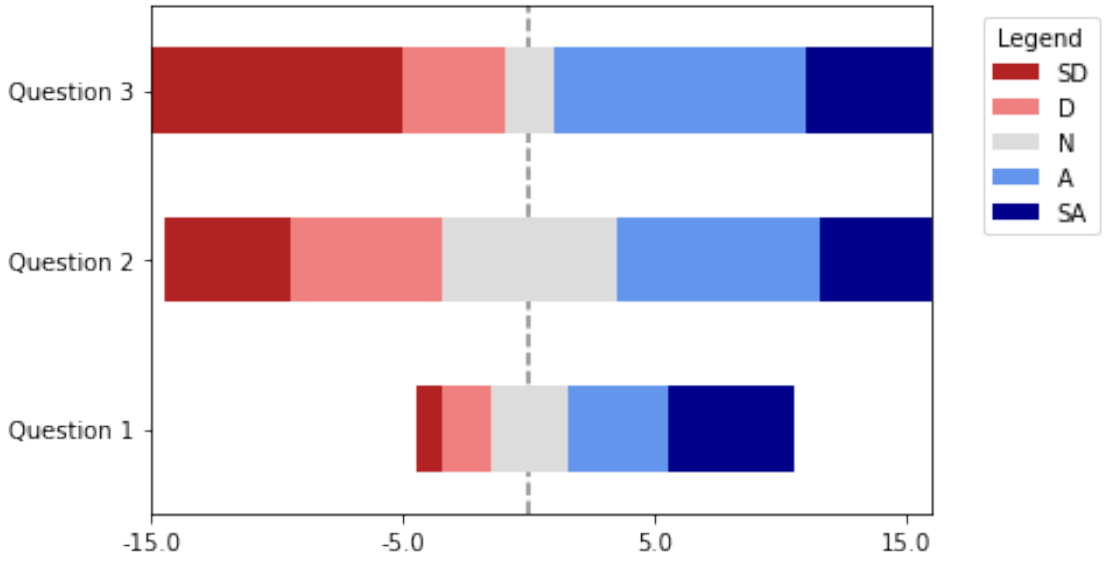
[27]:
```
Text(0, 0.5, 'sepal width (cm)')
```



Stacked bar chart to act as divergent chart. You can do the same thing much simpler using plotly, but you have to install plotly and import it.

[31]:
```
likert_colors = ['white', 'firebrick','lightcoral','gainsboro', \
                 'cornflowerblue', 'darkblue']
dummy = pd.DataFrame([[1,2,3,4, 5], [5,6,7,8, 5], [10, 4, 2, 10, 5]],
                     columns=["SD", "D", "N", "A", "SA"],
                     index=["Question 1", "Question 2", "Question 3"])
middles = dummy[["SD", "D"]].sum(axis=1)+dummy["N"]*.5
longest = middles.max()
complete_longest = dummy.sum(axis=1).max()
dummy.insert(0, '', (middles - longest).abs())

dummy.plot.barh(stacked=True, color=likert_colors, edgecolor='none', \
```

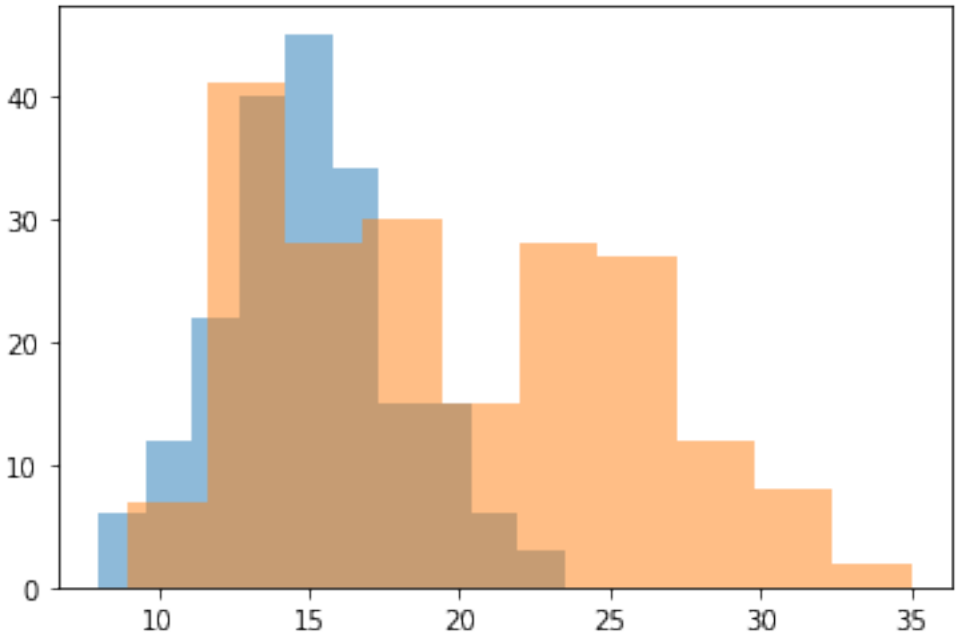
```
                legend=False)
z = plt.axvline(longest, linestyle='--', color='black', alpha=.5)
z.set_zorder(-1)

plt.xlim(0, complete_longest)
xvalues = range(0,complete_longest,10)
xlabels = [str(x-longest) for x in xvalues]
plt.xticks(xvalues, xlabels)
plt.legend(title='Legend', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



Layered histogram. The alpha=0.5 makes it transparent. You can lower the number more if you like.

[16]:
```
cars = pd.read_csv('cars1.csv')
```

[24]:
```
plt.hist(cars['acceleration'],alpha=0.5)
plt.hist(cars['mpg'],alpha=0.5);
```


40
30
20
10
0
10
15
20
25
30
35

[ ]: