

<http://www.pieriandata.com>

Pandas Data Visualization Exercise

This is just a quick exercise for you to review the various plots we showed earlier. Use **df3** to replicate the following plots.

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
df3 = pd.read_csv('df3.csv')
%matplotlib inline
```

```
In [3]: df3.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype  
---  -
 0    a      500 non-null     float64
 1    b      500 non-null     float64
 2    c      500 non-null     float64
 3    d      500 non-null     float64
dtypes: float64(4)
memory usage: 15.8 KB
```

```
In [4]: df3.head()
```

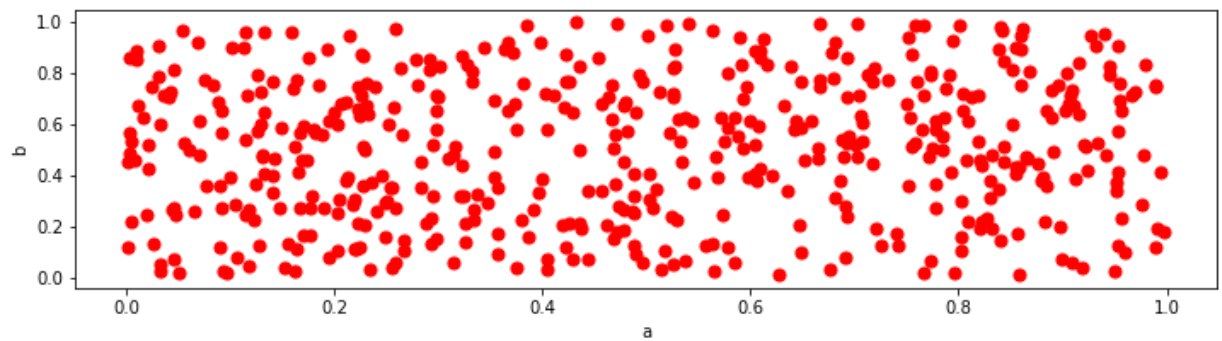
```
Out[4]:
```

	a	b	c	d
0	0.336272	0.325011	0.001020	0.401402
1	0.980265	0.831835	0.772288	0.076485
2	0.480387	0.686839	0.000575	0.746758
3	0.502106	0.305142	0.768608	0.654685
4	0.856602	0.171448	0.157971	0.321231

**** Recreate this scatter plot of b vs a. Note the color and size of the points. Also note the figure size. See if you can figure out how to stretch it in a similar fashion. Remember back to your matplotlib lecture... ****

```
In [5]: df3.plot.scatter(x='a',y='b',c='red',s=50,figsize=(12,3))
```

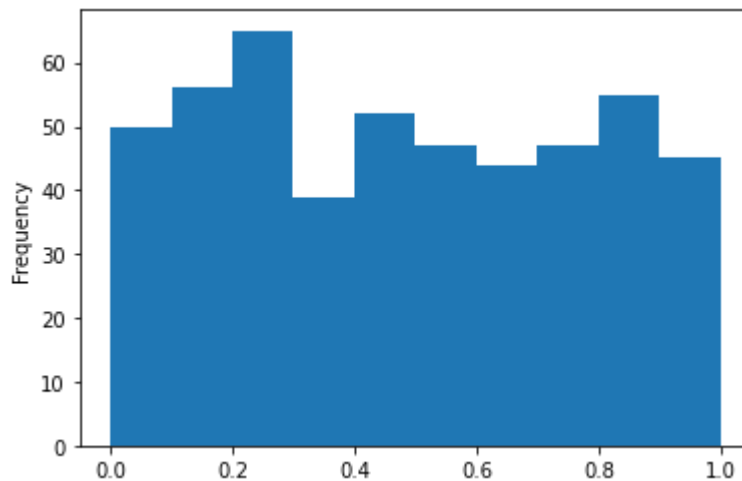
```
Out[5]: <AxesSubplot:xlabel='a', ylabel='b'>
```



**** Create a histogram of the 'a' column.****

```
In [6]: df3['a'].plot.hist()
```

```
Out[6]: <AxesSubplot:ylabel='Frequency'>
```

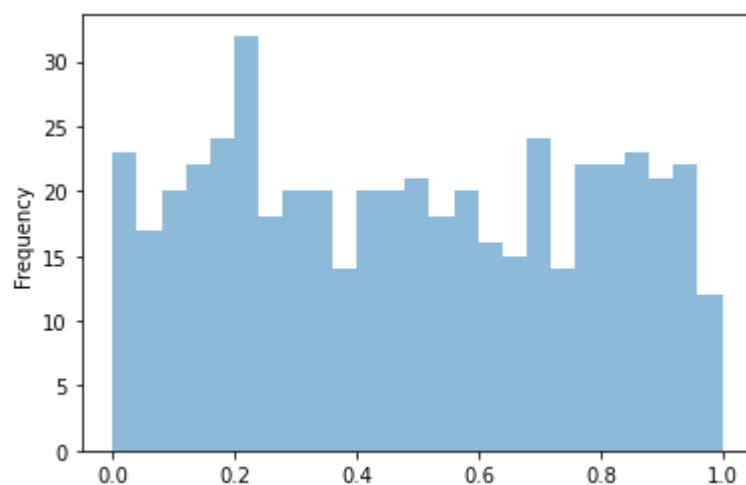


**** These plots are okay, but they don't look very polished. Use style sheets to set the style to 'ggplot' and redo the histogram from above. Also figure out how to add more bins to it.****

```
In [ ]: plt.style.use('ggplot')
```

```
In [7]: df3['a'].plot.hist(alpha=0.5,bins=25)
```

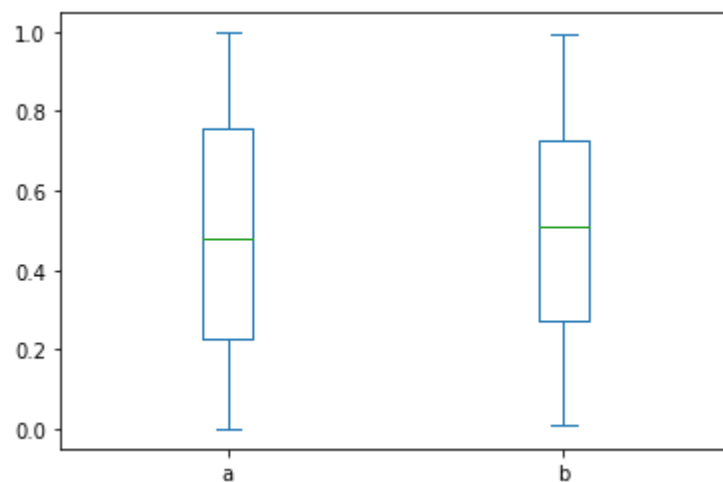
```
Out[7]: <AxesSubplot:ylabel='Frequency'>
```



**** Create a boxplot comparing the a and b columns.****

```
In [8]: df3[['a', 'b']].plot.box()
```

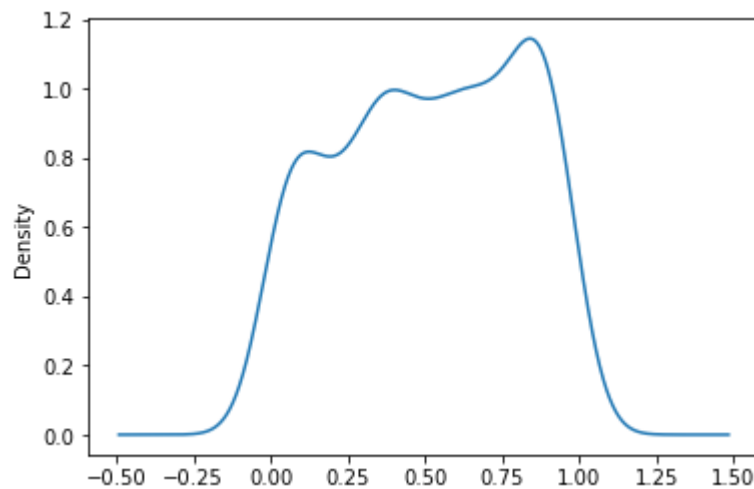
```
Out[8]: <AxesSubplot:>
```



**** Create a kde plot of the 'd' column ****

```
In [9]: df3['d'].plot.kde()
```

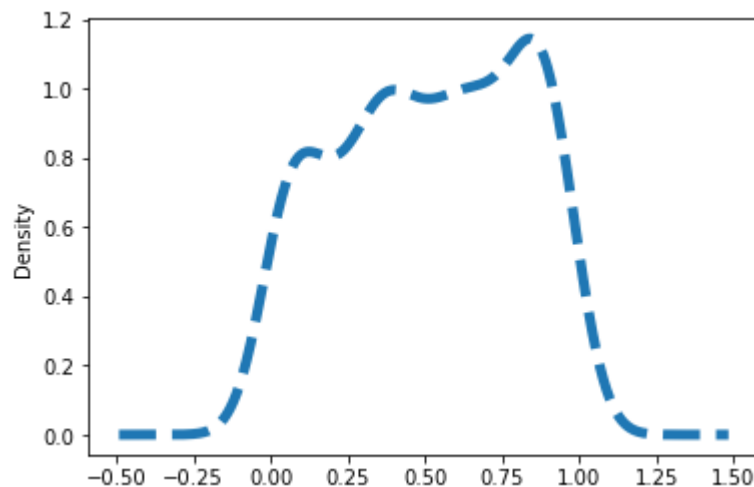
```
Out[9]: <AxesSubplot:ylabel='Density'>
```



**** Figure out how to increase the linewidth and make the linestyle dashed. (Note: You would usually not dash a kde plot line)****

```
In [10]: df3['d'].plot.density(lw=5,ls='--')
```

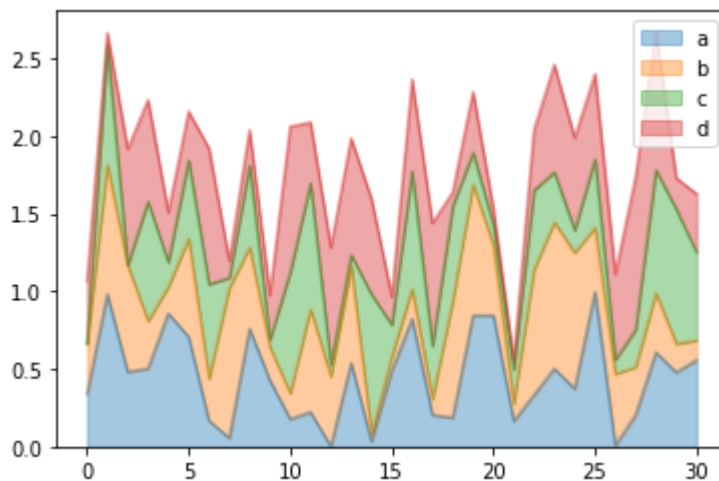
```
Out[10]: <AxesSubplot:ylabel='Density'>
```



**** Create an area plot of all the columns for just the rows up to 30. (hint: use .ix).****

```
In [19]: df3.loc[0:30].plot.area(alpha=0.4)
```

```
Out[19]: <AxesSubplot:>
```

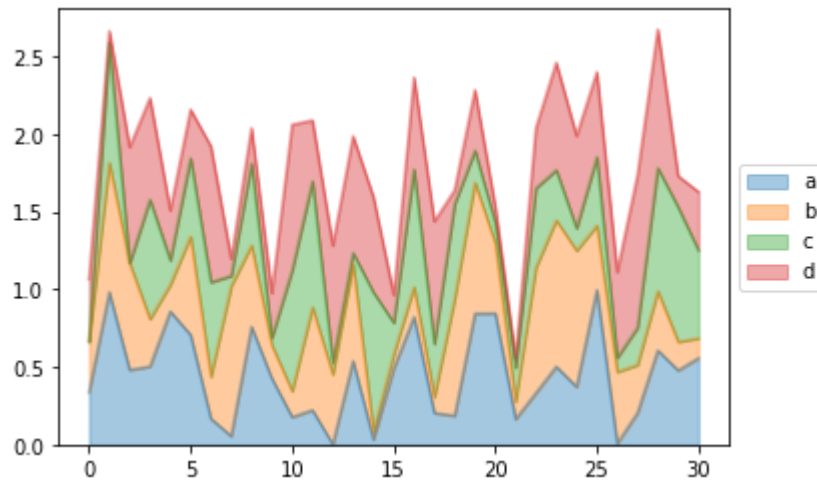


Bonus Challenge!

Note, you may find this really hard, reference the solutions if you can't figure it out! ** Notice how the legend in our previous figure overlapped some of actual diagram. Can you figure out how to display the legend outside of the plot as shown below?**

** Try searching Google for a good stackoverflow link on this topic. If you can't find it on your own - [use this one for a hint. \(http://stackoverflow.com/questions/23556153/how-to-put-legend-outside-the-plot-with-pandas\)](http://stackoverflow.com/questions/23556153/how-to-put-legend-outside-the-plot-with-pandas)**

```
In [18]: f = plt.figure()
df3.loc[0:30].plot.area(alpha=0.4,ax=f.gca())
plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))
plt.show()
```



Great Job!