

Pandas Data Visualization Übung - Aufgabe

Das ist eine kleine Übung zur Anwendung der eingebauten Visualisierungsmöglichkeiten in *Pandas*. Dazu nutzen wir das **df3** CSV. Bitte stellt die folgenden Diagramme durch Nutzung von *Pandas* dar. Die Imports seien gegeben:

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
df3 = pd.read_csv('df3')
%matplotlib inline
```

```
In [2]: df3.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
a      500 non-null float64
b      500 non-null float64
c      500 non-null float64
d      500 non-null float64
dtypes: float64(4)
memory usage: 15.7 KB
```

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

```
Out[3]:
```

| | 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 |

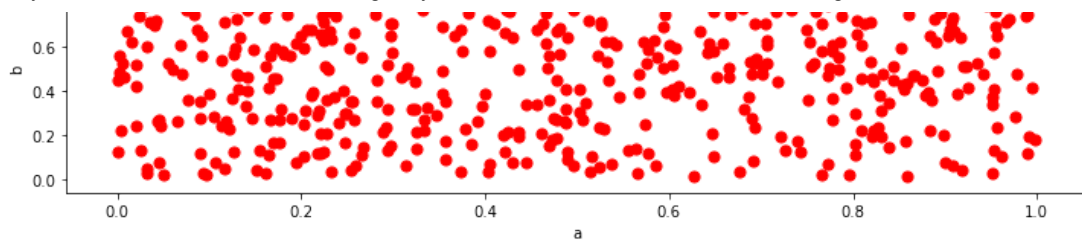
Erstelle dieses Scatter Plot von b vs. a. Achte auf die Farbe und Größe der Punkte. Versuche außerdem es in die vorgegebene Form zu bringen. Dazu könnte eine Auffrischung der Matplotlib-Befehle helfen.

```
In [ ]:
```

```
In [4]:
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x10e8d29e8>
```



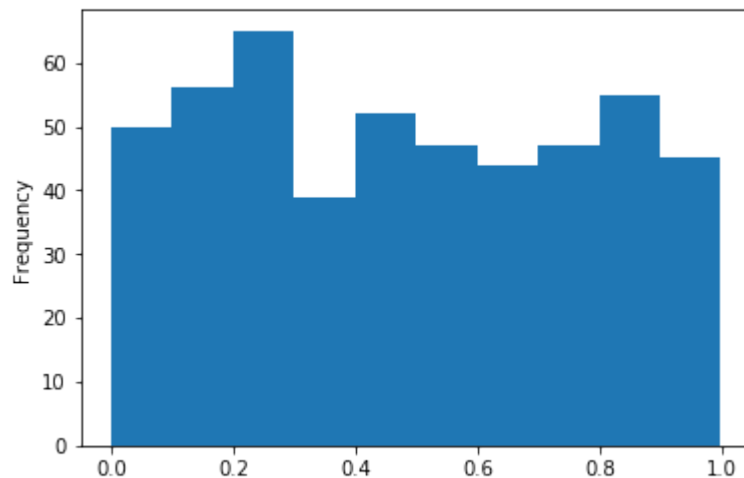


Erstelle ein Histogramm der "a" Spalte.

In []:

In [10]: `df3['a'].plot.hist()`

Out[10]: `<matplotlib.axes._subplots.AxesSubplot at 0x111dfc8d0>`



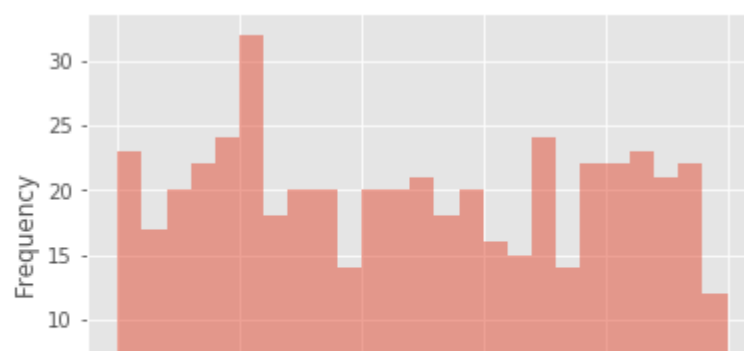
Diese Plots sind soweit okay. Sie sehen aber nicht einheitlich aus. Nutze das `ggplot` Stylesheet, um das Erscheinungsbild der folgenden Diagramme zu beeinflussen. Erstelle dann das Histogramm von eben erneut und füge zusätzlich mehr Teilbalken hinzu.

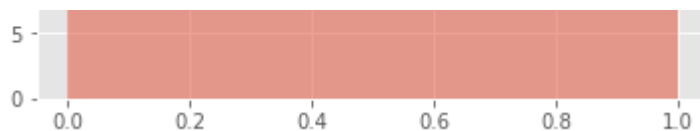
In [11]: `# Style hier`

In []: `# Plot hier`

In [7]:

Out[7]: `<matplotlib.axes._subplots.AxesSubplot at 0x111e13f60>`



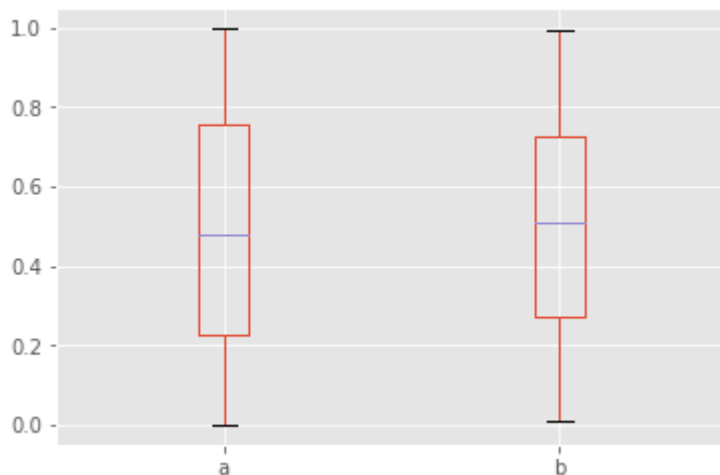


Erstelle ein Box Plot, dass die Spalten a und b vergleicht.

```
In [ ]: # Plot hier
```

```
In [8]:
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x111fbffd0>
```

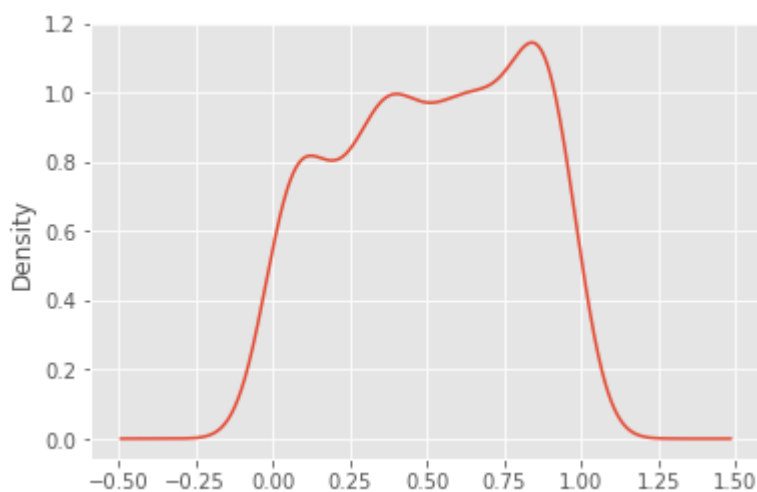


Erstelle ein KDE Plot der Spalte "d".

```
In [ ]: # Plot hier
```

```
In [9]:
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x111fd8748>
```



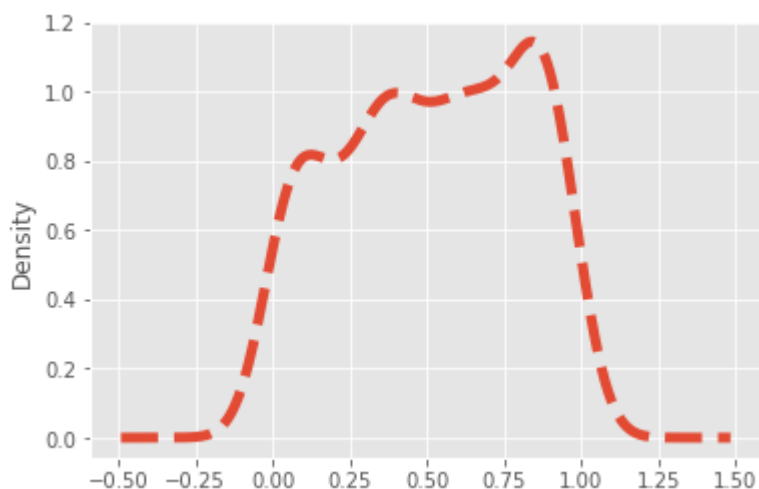
Finde heraus, wie wir die Linienbreite erhöhen und die Linie stricheln können.

Hinweis: Normalerweise würden wir ein KDE Diagramm nicht stricheln.

```
In [ ]: # Plot hier
```

```
In [10]:
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x113d8e9b0>
```



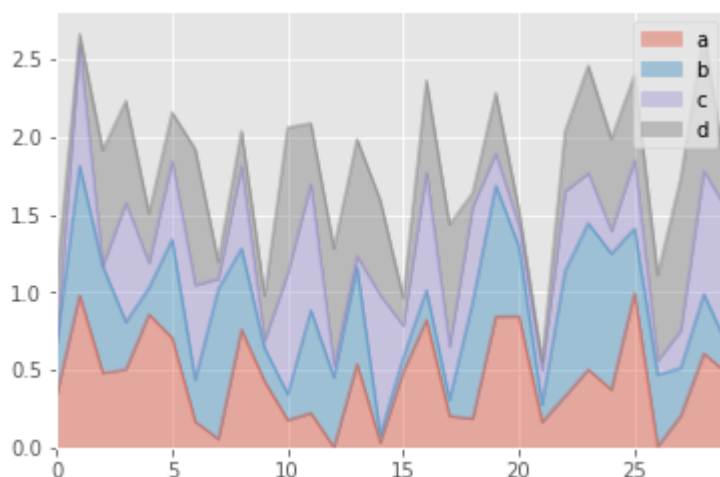
Erstelle ein Flächendiagramm (en. area plot) für alle Spalten. Nutze dabei aber nur die ersten 30 Zeilen.

Hinweis: Für die Zeilen kannst du .iloc nutzen.

```
In [ ]: # Plot hier
```

```
In [13]:
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1141a1550>
```



Bonus Aufgabe!

Das könnte wirklich schwer sein, weshalb es okay ist, sich auf die Lösung zu beziehen, wenn man nicht weiter kommt.

Sicher fällt euch auf, dass die Legende in der vorherigen Lösung das eigentliche Diagramm überdeckt. Findest du einen Weg, um die Legende

neben dem eigentlichen Diagramm anzuzeigen?

Hinweis: Google Suche nach einem guten Stackoverflow-Link zu diesem Thema hilft.

In []: `# Plot hier`

In [12]:

