histogram

August 26, 2016

0.0.1 Histogram plotter

---

1. Case Study: Plot of Gaussian data

```
In [16]: import random as random
         import matplotlib.pyplot as plt
         import math as math
         import numpy as np
```

```
In [17]: data_points = 100
```

```
In [43]: #x = [random.uniform(0.0,10.0) for k in range(data_points)]
         x = np.random.normal(5, 1, 100000)
```

```
In [44]: x[0:10]
```

```
Out[44]: array([ 4.97463244,  4.53220668,  3.73256279,  5.34760518,  6.70448723,
                 6.7285636 ,  4.46832196,  6.02182185,  4.95829785,  5.61225992])
```

```
In [45]: math.pi
```

```
Out[45]: 3.141592653589793
```

```
In [46]: area = 2*math.pi*np.array(x)**2
```

```
In [47]: area[0:10]
```

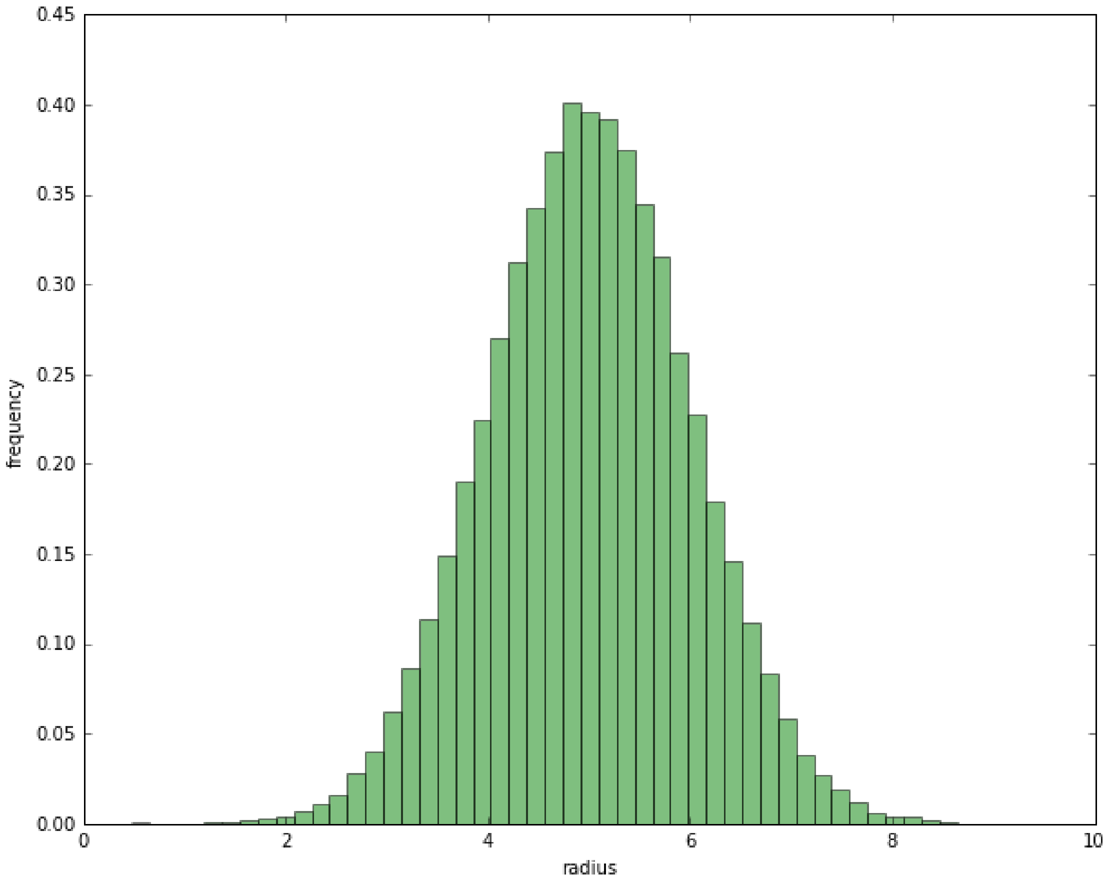
```
Out[47]: array([ 155.48978528,  129.06226485,   87.5374947 ,  179.67950379,
                 282.43011558,  284.4622178 ,  125.44945694,  227.84299152,
                 154.47033618,  197.90438694])
```

Plot of radious histogram

```
In [48]: %matplotlib inline
```

```
In [61]: num_bins =50
         plt.figure(1)
         plt.figure(figsize=(10, 8))
         plt.xlabel("radius")
         plt.ylabel("frequency")
         plt.hist(x,num_bins, normed= 1.0, facecolor="green",alpha = 0.5)
         plt.show()
```

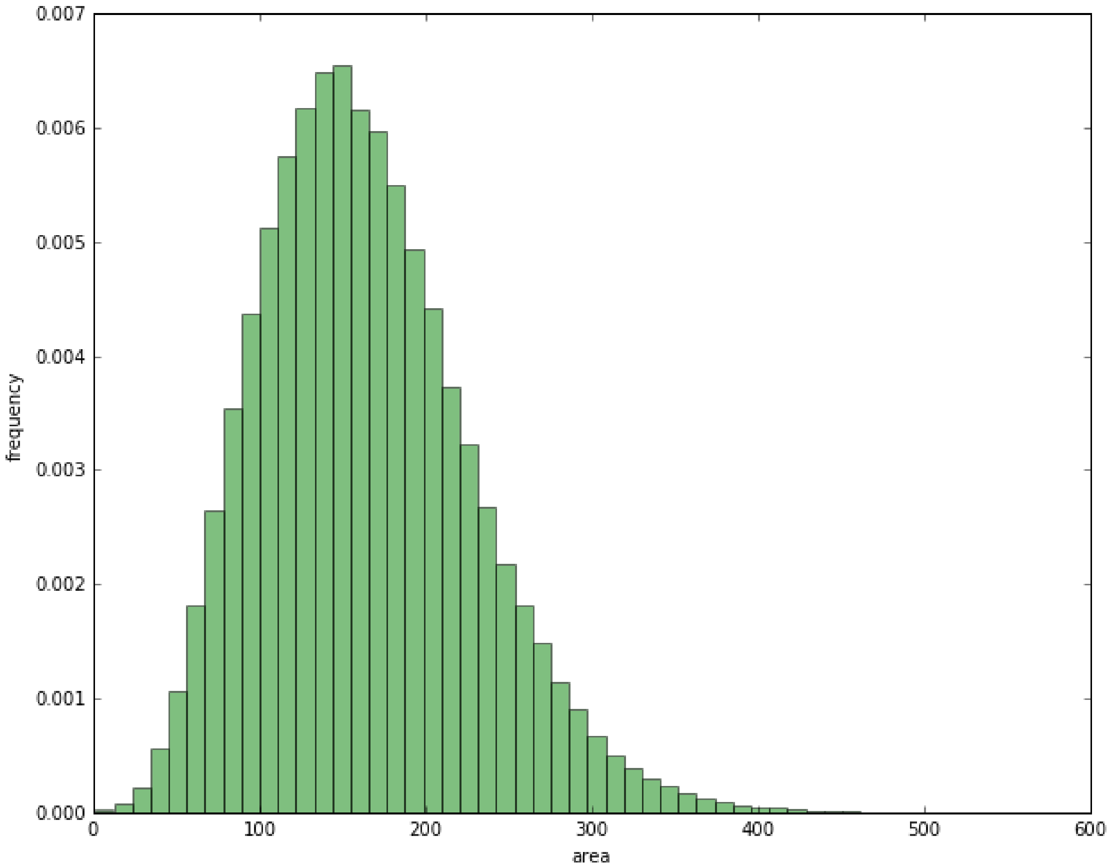
```
<matplotlib.figure.Figure at 0x119f5d150>
```



Area Histogram

```
In [64]: num_bins =50
         plt.figure(1)
         plt.figure(figsize=(10, 8))
         plt.xlabel("area")
         plt.ylabel("frequency")
         plt.hist(area,num_bins, normed= 1.0, facecolor="green",alpha = 0.5)
         plt.show()
```

```
<matplotlib.figure.Figure at 0x119f5df10>
```



```
In [51]: 2*math.pi*5**2

Out[51]: 157.07963267948966

In [52]: max(area)

Out[52]: 549.14205240606145
```

0.0.2 Plot from data file

```
In [76]: import pandas as pd

In [77]: df = pd.read_csv('histogram.csv')

In [78]: df.shape

Out[78]: (47, 3)

In [79]: df

Out[79]:     Currents (pA)  Radius (um)  Effective area(um^2)
         0          179.20     0.492660              1.525015
         1           65.66     0.180514              0.204739
```

```
2           152.90     0.420355                 1.110230
3           125.50     0.345027                 0.747972
4            33.96     0.093363                 0.054769
5            25.70     0.070655                 0.031366
6           117.30     0.322483                 0.653422
7           150.90     0.414857                 1.081375
8           144.40     0.396987                 0.990222
9            51.58     0.141805                 0.126346
10           70.37     0.193462                 0.235165
11           90.56     0.248969                 0.389467
12           27.25     0.074916                 0.035264
13           27.25     0.074916                 0.035264
14           41.36     0.113708                 0.081238
15          100.25     0.275609                 0.477273
16          103.45     0.284407                 0.508229
17          100.70     0.276846                 0.481567
18          102.00     0.280420                 0.494081
19          114.20     0.313961                 0.619342
20          121.17     0.333123                 0.697250
21          111.92     0.307692                 0.594858
22          117.33     0.322566                 0.653757
23          124.93     0.343460                 0.741193
24          146.33     0.402293                 1.016868
25          158.82     0.436631                 1.197866
26           21.76     0.059823                 0.022486
27          134.62     0.370100                 0.860631
28           45.84     0.126024                 0.099790
29           52.09     0.143207                 0.128857
30           94.68     0.260296                 0.425711
31           83.33     0.229092                 0.329762
32           41.19     0.113240                 0.080572
33          102.90     0.282894                 0.502839
34           59.06     0.162369                 0.165648
35           93.04     0.255787                 0.411090
36          183.68     0.504976                 1.602219
37           20.79     0.057156                 0.020526
38           18.68     0.051355                 0.016571
39          136.30     0.374718                 0.882246
40           60.24     0.165613                 0.172333
41           78.89     0.216886                 0.295557
42           90.00     0.247430                 0.384665
43          100.10     0.275197                 0.475846
44           80.00     0.219937                 0.303933
45           80.00     0.219937                 0.303933
46          167.50     0.460494                 1.332379
```

```
In [80]: df.columns = ['I','R','A']
```

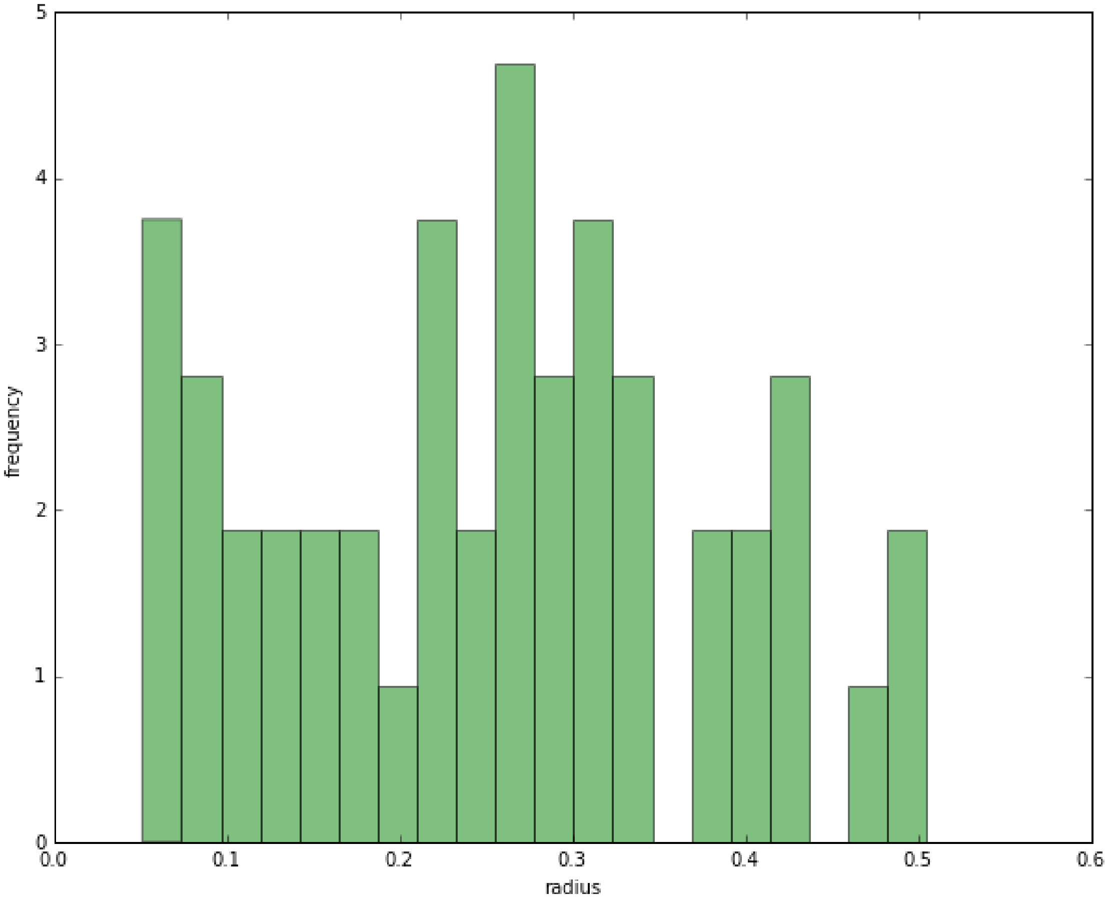
```
In [81]: df.head()
```

```
Out[81]:          I         R         A
         0  179.20  0.492660  1.525015
         1   65.66  0.180514  0.204739
         2  152.90  0.420355  1.110230
         3  125.50  0.345027  0.747972
         4   33.96  0.093363  0.054769
```

Radius histogram

```
In [82]: num_bins =20
         plt.figure(1)
         plt.figure(figsize=(10, 8))
         plt.xlabel("radius")
         plt.ylabel("frequency")
         plt.hist(df.R,num_bins, normed= 1.0, facecolor="green",alpha = 0.5)
         plt.show()
```

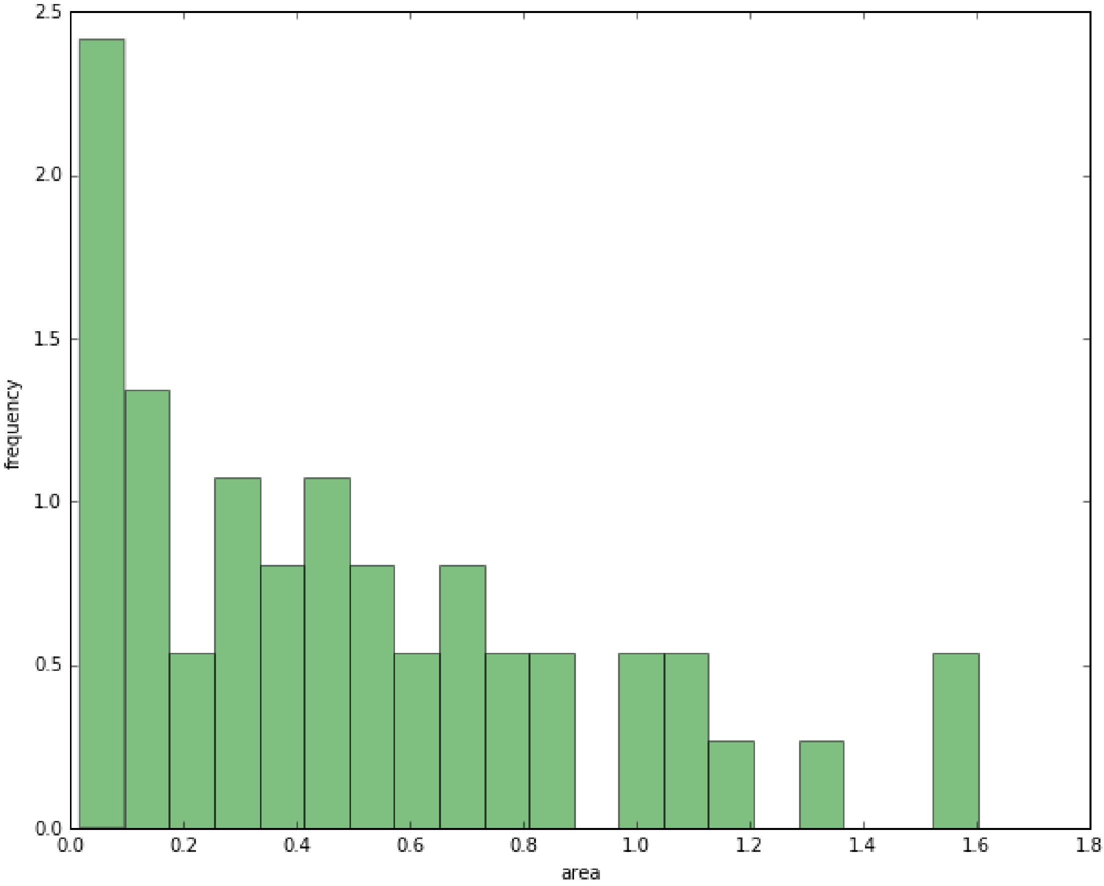
```
<matplotlib.figure.Figure at 0x11b027d10>
```



Area Histogram

```
In [83]: num_bins =20
         plt.figure(1)
         plt.figure(figsize=(10, 8))
         plt.xlabel("area")
         plt.ylabel("frequency")
         plt.hist(df.A,num_bins, normed= 1.0, facecolor="green",alpha = 0.5)
         plt.show()
```

```
<matplotlib.figure.Figure at 0x11b0044d0>
```



0.1 Conclusion Data is not enough to draw histogram!

```
In [ ]:
```