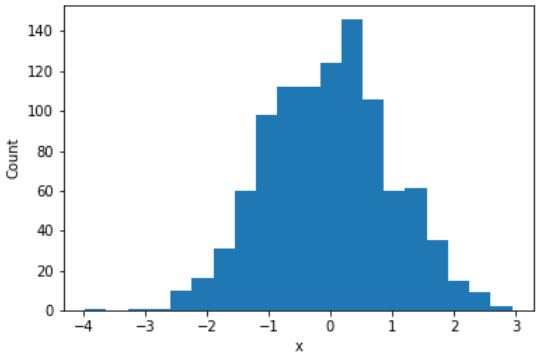
Figure 1: A histogram with defined number of bins



1 Plotting histograms with matplotlib and NumPy

Matplotlib has an easy method for plotting data. NumPy has an easy method for obtaining histogram data.

1.1 Plotting histograms with Matplotlib

Plotting a histogram with a defined number of bins:

```
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline

x=np.random.randn(1000) # samples from a normal distribution

plt.hist(x,bins=20)
plt.title ('Defined number of bins')
plt.xlabel ('x')
plt.ylabel ('Count')

plt.show()
```

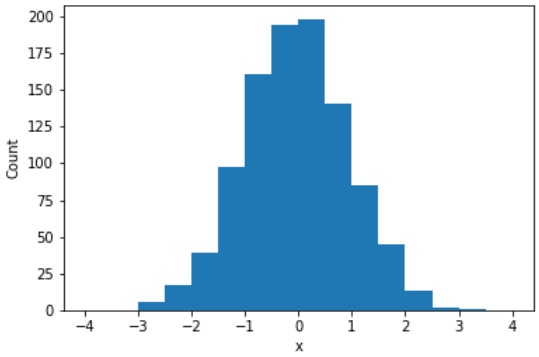
Plotting a histogram with a defined range of bins:

```
x=np.random.randn(1000) # samples from a normal distribution

# Use np.arange to create bins from -4 to +4 in steps of 0.5
# A custom list could also be used

plt.hist(x,bins=np.arange(-4,4.5,0.5))
plt.title ('Defined bin range and width')
plt.xlabel ('x')
plt.ylabel ('Count')

plt.show()
```

1.2 Obtaining histogram data with NumPy

If histogram data is needed in addition to, or instead of, a plot, NumPy may be used. Here a defined number of bins is used:

Figure 2: A histogram with defined bin range and width




```
import numpy as np
count, bins = np.histogram(x, bins=20)
print ('Bins:')
print (bins)
print ('\nCount:')
print (count)

OUT:

Bins:
[-2.88652288 -2.57566476 -2.26480664 -1.95394852 -1.6430904  -1.33223228
 -1.02137416 -0.71051604 -0.39965792 -0.0887998   0.22205832  0.53291644
  0.84377456  1.15463268  1.4654908   1.77634892  2.08720704  2.39806516
  2.70892328  3.0197814   3.33063952]

Count:
[  5   9  12  22  32  69  80 113 125 116 108  94  77  59  40  22  10   4
   1   2]
```

And here a defined bin range is used.

```
import numpy as np
count, bins = np.histogram(x, bins=np.arange(-5,6,1))
print ('Bins:')
print (bins)
print ('\nCount:')
print (count)

OUT:

Bins:
[-5 -4 -3 -2 -1  0  1  2  3  4  5]

Count:
[  0   0  20 134 332 340 154  18   2   0]
```