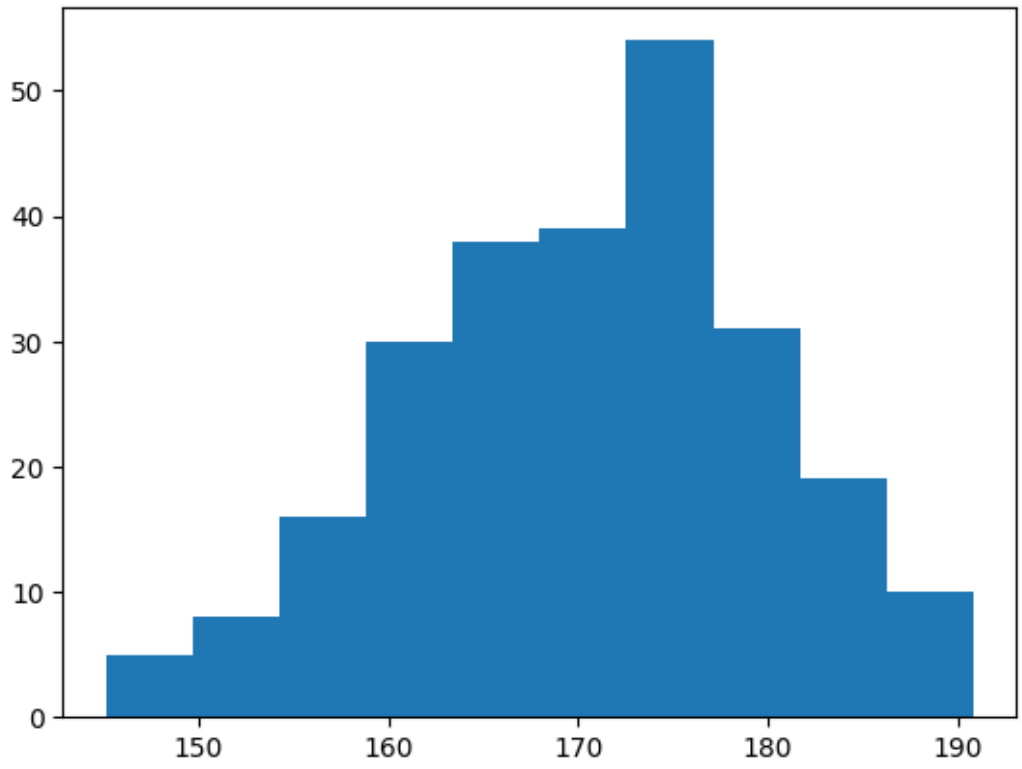
histogram

November 1, 2024

[1]:
```
import matplotlib.pyplot as plt
import numpy as np

x = np.random.normal(170, 10, 250)
plt.hist(x)
plt.show()
```



[2]:
```
'''Syntax: matplotlib.pyplot.hist(x, bins=None, range=None, density=False,
                                  weights=None, cumulative=False, bottom=None,␣
 ↪histtype='bar', align='mid',
                                  orientation='vertical', rwidth=None, log=False,␣
 ↪color=None, label=None,
```

```
                                stacked=False, \*, data=None, \*\*)

Parameters: This method accept the following parameters that are described␣
 ↪below:

x : This parameter are the sequence of data.
bins : This parameter is an optional parameter and
    it contains the integer or sequence or string.

range : This parameter is an optional parameter and
    it the lower and upper range of the bins.

density : This parameter is an optional parameter and
    it contains the boolean values.

weights : This parameter is an optional parameter and
    it is an array of weights, of the same shape as x.

bottom : This parameter is the location of the bottom baseline
    of each bin.
histtype : This parameter is an optional parameter and
    it is used to draw type of histogram.
    {'bar', 'barstacked', 'step', 'stepfilled'}

align : This parameter is an optional parameter and
    it controls how the histogram is plotted.
    {'left', 'mid', 'right'}

rwidth : This parameter is an optional parameter and
    it is a relative width of the bars
    as a fraction of the bin width

log : This parameter is an optional parameter and
    it is used to set histogram axis to a log scale

color : This parameter is an optional parameter and
    it is a color spec or sequence of color specs,
    one per dataset.

label : This parameter is an optional parameter and
    it is a string, or sequence of strings
    to match multiple datasets.

normed : This parameter is an optional parameter and
    it contains the boolean values.
    It uses the density keyword argument instead.
```

```
    Returns:
        n :This returns the values of the histogram bins.
bins :This returns the edges of the bins.
patches :This returns the list of individual patches used to
    create the histogram.'''
```

[2]:
```
"Syntax: matplotlib.pyplot.hist(x, bins=None, range=None, density=False,\n
weights=None, cumulative=False, bottom=None, histtype='bar', align='mid', \n
orientation='vertical', rwidth=None, log=False, color=None, label=None,\n
stacked=False, \\*, data=None, \\*\\*)\n\nParameters: This method accept the
following parameters that are described below:\n\nx : This parameter are the
sequence of data.\nbins : This parameter is an optional parameter and \n    it
contains the integer or sequence or string.\n    \nrange : This parameter is an
optional parameter and\n    it the lower and upper range of the bins.\n
\ndensity : This parameter is an optional parameter and\n    it contains the
boolean values.\n    \nweights : This parameter is an optional parameter and \n
it is an array of weights, of the same shape as x.\n    \nbottom : This
parameter is the location of the bottom baseline\n    of each bin.\nhisttype :
This parameter is an optional parameter and\n    it is used to draw type of
histogram.\n    {‘bar’, ‘barstacked’, ‘step’, ‘stepfilled’}\n    \nalign : This
parameter is an optional parameter and\n    it controls how the histogram is
plotted.\n    {‘left’, ‘mid’, ‘right’}\n    \nrwidth : This parameter is an
optional parameter and \n    it is a relative width of the bars \n    as a
fraction of the bin width\n    \nlog : This parameter is an optional parameter
and \n    it is used to set histogram axis to a log scale\n    \ncolor : This
parameter is an optional parameter and \n    it is a color spec or sequence of
color specs, \n    one per dataset.\n    \nlabel : This parameter is an optional
parameter and \n    it is a string, or sequence of strings \n    to match
multiple datasets.\n    \nnormed : This parameter is an optional parameter and
\n    it contains the boolean values.\n    It uses the density keyword argument
instead.\n    \n    Returns:\n        n :This returns the values of the
histogram bins.\nbins :This returns the edges of the bins.\npatches :This
returns the list of individual patches used to \n    create the histogram."
```

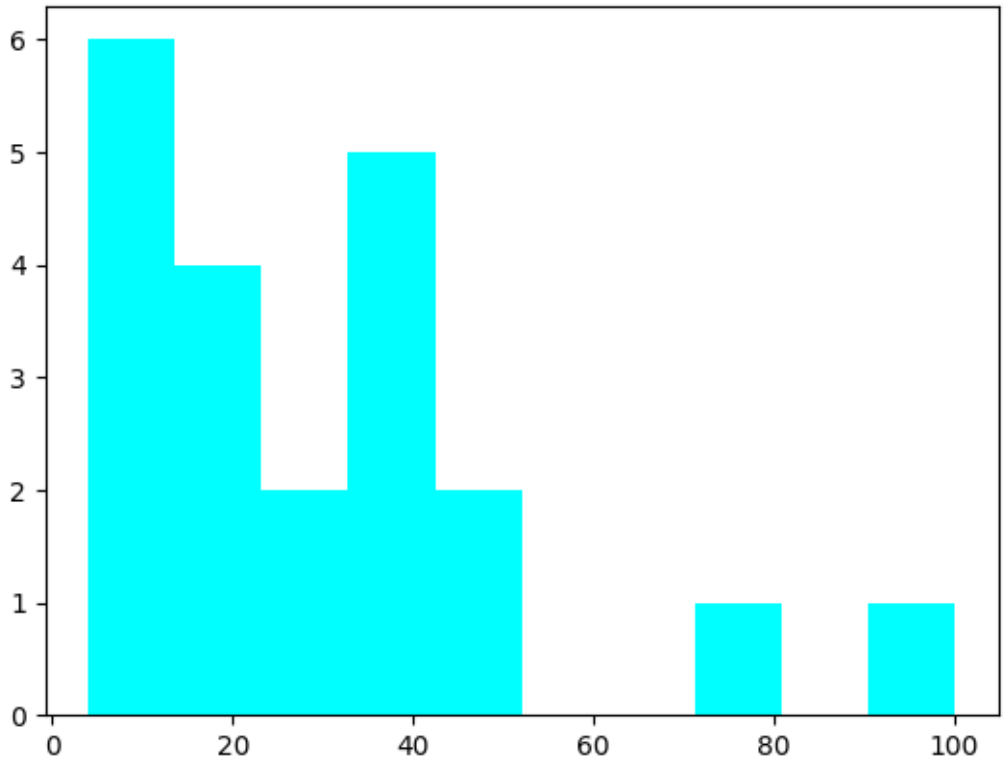
[3]:
```
x
```

[3]:
```
array([180.1891559 , 181.82130153, 151.0668477 , 164.74761425,
       156.2688702 , 163.6915134 , 160.66162479, 173.42438473,
       169.38124832, 171.94088107, 158.14620939, 151.00103365,
       174.88031851, 175.09007443, 173.59376329, 189.6826205 ,
       173.3506868 , 168.63890476, 169.11992538, 170.72441942,
       183.4699989 , 176.66731067, 177.77064196, 163.47498432,
       173.32572649, 177.93999991, 172.14267539, 161.93396696,
       179.69351339, 170.91869829, 157.46563784, 178.21022054,
       187.45887605, 164.98677176, 162.07027002, 149.47145897,
       161.13846585, 176.66130575, 151.14384026, 166.15445415,
       170.35016316, 155.43680541, 171.86872469, 161.09826464,
```

```
162.14347399, 168.42889575, 184.01099535, 165.95270288,
158.22502074, 178.77659096, 175.58163288, 165.5931718 ,
182.83269122, 168.753855  , 170.13104198, 174.49564919,
181.17215491, 155.99992172, 156.45939994, 178.00824447,
161.80007414, 158.13358705, 177.17690621, 162.25869367,
182.07035293, 154.01765828, 153.11680382, 166.41730348,
181.63473271, 179.03854895, 158.06988626, 173.39452932,
159.07434101, 174.21462882, 152.01916398, 176.23598915,
165.54675575, 161.44268824, 164.37923421, 171.9356047 ,
184.17090877, 176.30459248, 182.62864513, 169.37111736,
176.90419913, 161.02017308, 182.78360832, 172.98717014,
171.77450139, 146.65165918, 177.90293963, 169.43326009,
175.84744201, 188.88152738, 167.20759594, 167.10356235,
172.24937591, 172.71098658, 168.15534057, 174.10411343,
172.60458931, 174.85063902, 186.43178479, 160.70577265,
173.70747588, 163.08938047, 176.37712032, 161.65851484,
179.49879928, 172.3821393 , 164.20456516, 160.78842538,
183.95665191, 172.77981707, 163.64172854, 176.1890116 ,
162.3232011 , 163.78928812, 174.22252627, 167.50461407,
166.6792524 , 171.97320553, 177.00088154, 182.71599223,
170.25969077, 155.50786061, 187.31405951, 170.17182328,
172.59460215, 160.73071251, 149.62250015, 167.15264382,
164.45721781, 171.14116328, 183.00837686, 174.85516674,
190.83714678, 167.44552831, 185.82337777, 172.83572763,
172.94753023, 182.71833908, 179.59932041, 174.21053205,
172.63577566, 187.59324866, 183.79744561, 175.95671517,
172.00800751, 156.2733159 , 178.92254929, 165.34046579,
187.2098573 , 176.59351614, 154.97658562, 181.14221905,
182.52125413, 163.62244694, 172.61099678, 182.18928159,
162.33101453, 163.10074031, 152.53478013, 155.61285449,
169.36077576, 176.04330126, 181.20616   , 172.2412444 ,
170.33324403, 164.68528635, 168.96827276, 160.52692431,
162.63623663, 169.4550788 , 176.25737146, 179.00431882,
175.88978438, 161.89308904, 167.54408039, 175.94683532,
167.61426588, 162.1327988 , 183.51838304, 177.555114  ,
167.54018547, 169.92044994, 171.46557142, 169.09062221,
177.91512621, 164.59787423, 178.32761306, 171.88130167,
166.84220133, 159.56706329, 159.94401771, 170.60770969,
178.02760858, 154.13383453, 180.27145769, 168.07217441,
177.21222836, 173.57034618, 174.54837206, 178.09082136,
163.93240385, 175.34037568, 155.69816066, 174.55757414,
159.76557563, 159.73965996, 176.45165302, 165.62634865,
172.37069318, 164.59499487, 172.33941233, 172.70385329,
146.99029766, 162.52164597, 145.13208403, 187.33156942,
180.35543922, 178.11564027, 166.44322925, 166.31082958,
179.34398995, 175.10394809, 160.57386495, 182.13447835,
173.45220398, 167.09081408, 165.19759455, 155.78112776,
```

```
       157.49635031, 172.89709373, 182.9511535 , 178.74028621,
       168.08279516, 167.66981233, 174.9560128 , 177.20456812,
       170.3215117 , 162.38248385, 165.81939794, 176.63727045,
       181.042489  , 188.11357173, 174.84603481, 175.54192573,
       166.32833033, 173.74250277])
```

[4]:
```
import numpy as np
import matplotlib.pyplot as plt
x = [21,22,23,4,5,6,77,8,9,10,31,32,33,34,35,36,37,18,49,50,100]
n_bins = 10     #no of bins
#patches is the specifics of histogram diagram measurements
bin_heights, bins, patches = plt.hist(x,bins=10, facecolor='cyan')
plt.show()
```



[5]:
```
bin_heights
```

[5]:
```
array([6., 4., 2., 5., 2., 0., 0., 1., 0., 1.])
```

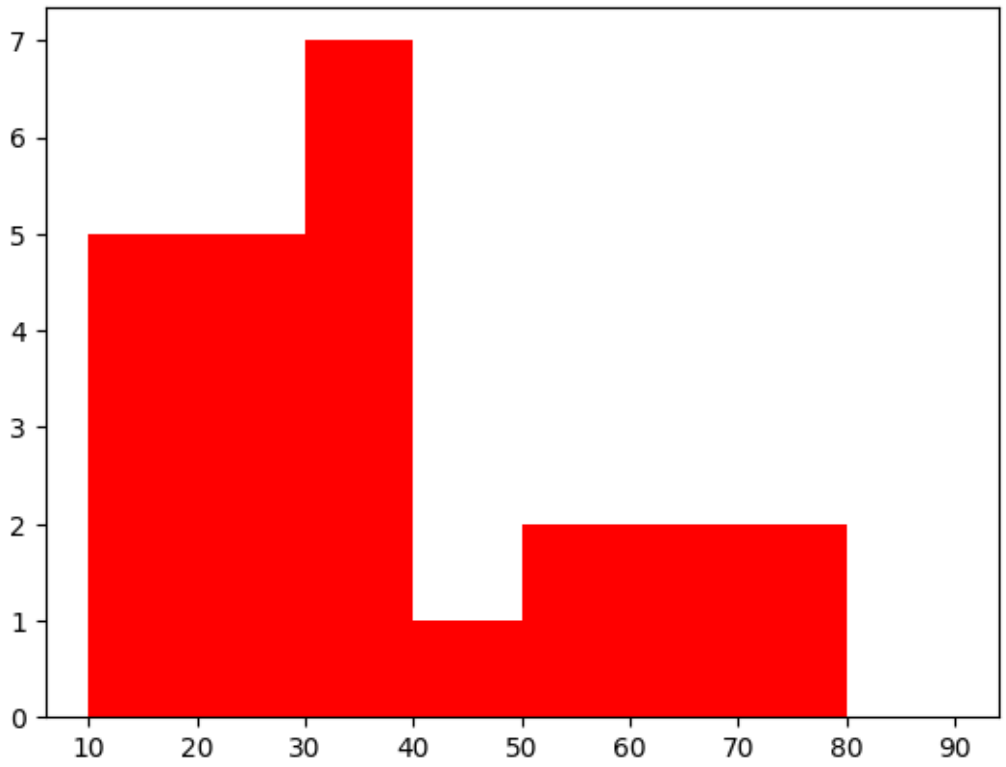
[6]:
```
bins
```

[6]:
```
array([  4. ,  13.6,  23.2,  32.8,  42.4,  52. ,  61.6,  71.2,  80.8,
        90.4, 100. ])
```

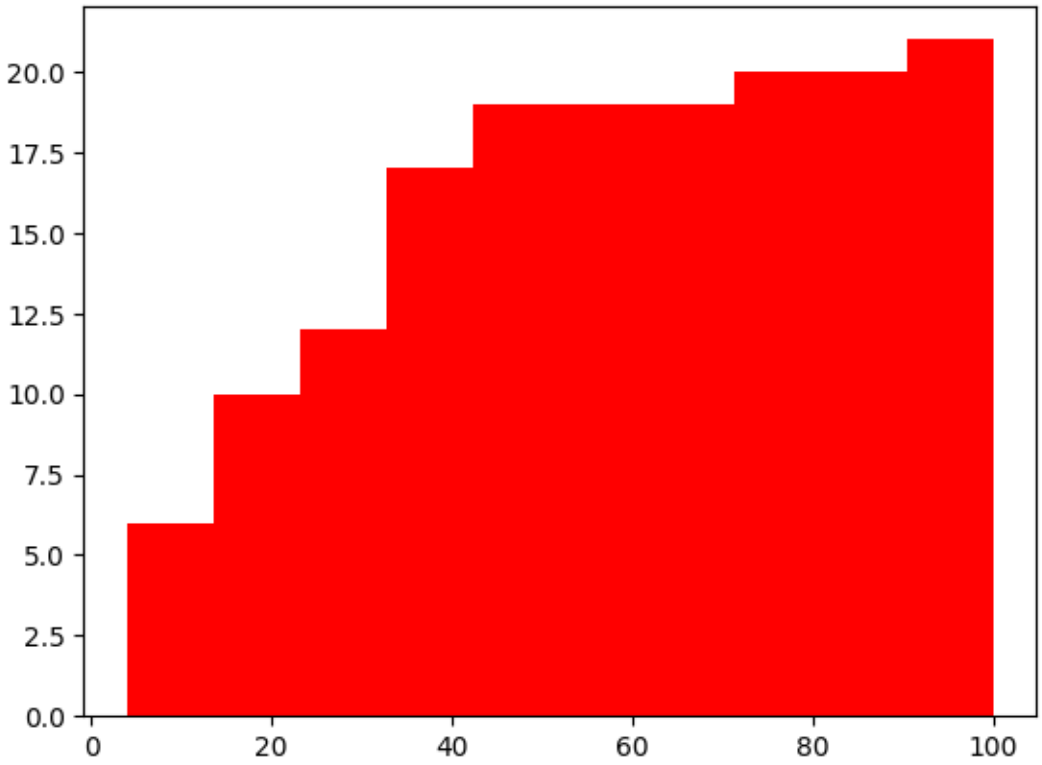
[7]:
```
print(patches[0])
```

```
Rectangle(xy=(4, 0), width=9.6, height=6, angle=0)
```

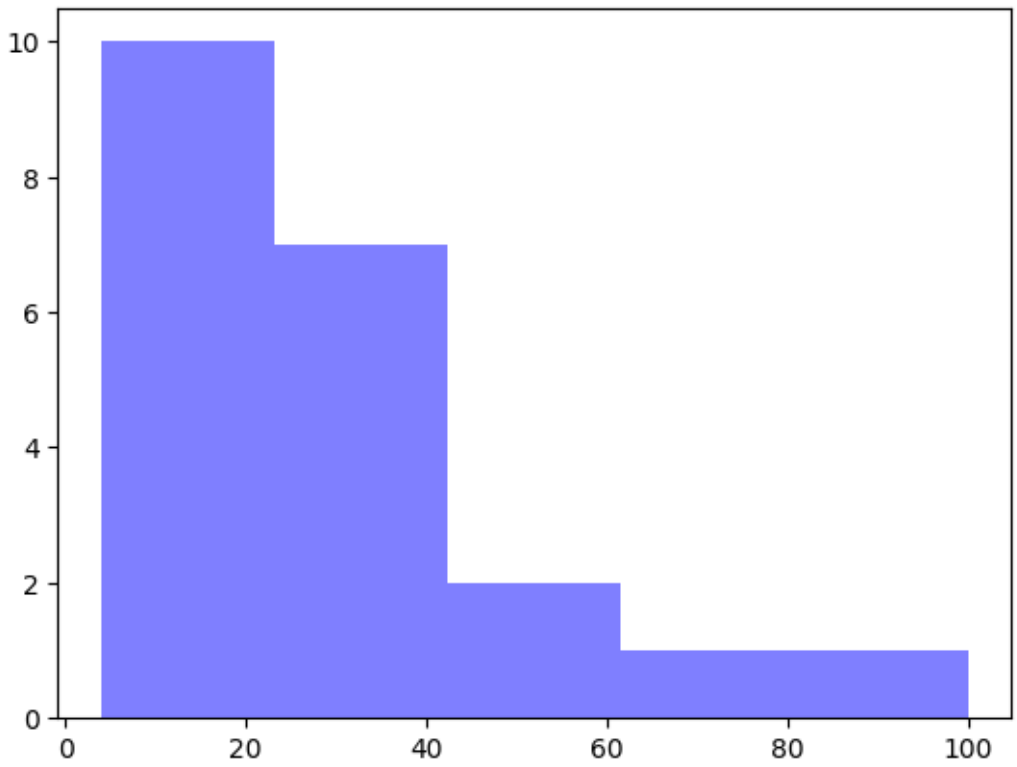
[8]:
```
#to create unequally sized bins

n_bins = [10, 30, 40, 50,80,90]      #bin values
bin_heights, bins, patches = plt.hist(x, n_bins, facecolor='red')
plt.show()
```



[9]:
```
#cumulative histogram -> sets the bin height as, plotted_bin(n) = actual_bin(n)␣
 ↪+ plotted_bin(n-1) for all bins
plt.hist(x, cumulative=True, facecolor='red')
plt.show()
```

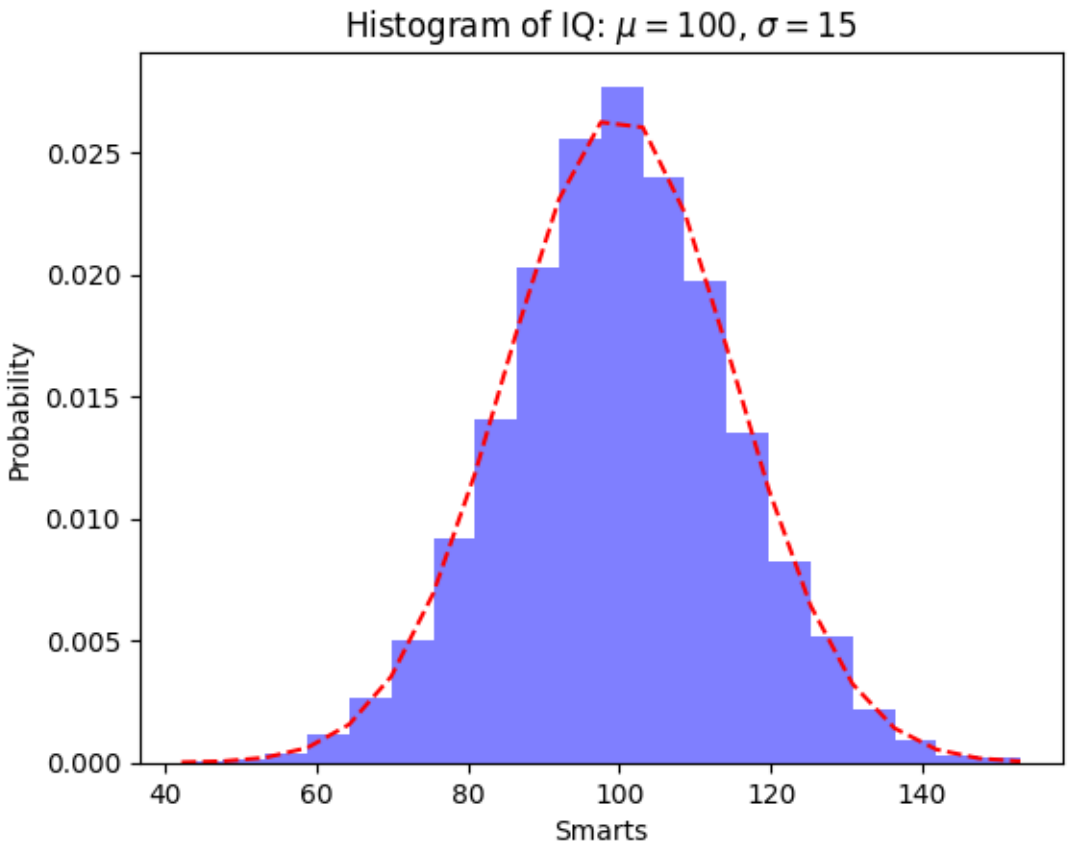


[10]:
```
import numpy as np

import matplotlib.pyplot as plt

x = [21,22,23,4,5,6,77,8,9,10,31,32,33,34,35,36,37,18,49,50,100]
num_bins = 5
#n, bins, patches = plt.hist(x, num_bins, facecolor='blue', alpha=0.5)
plt.hist(x, num_bins, facecolor='blue', alpha=0.5)
plt.show()
```



[11]:
```
#!/usr/bin/env python

import numpy as np
#import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
from scipy.stats import norm
# example data
mu = 100 # mean of distribution
sigma = 15 # standard deviation of distribution
x = mu + sigma * np.random.randn(10000)

num_bins = 20
# the histogram of the data
n, bins, patches = plt.hist(x, num_bins, density=1, facecolor='blue', alpha=0.5)

# add a 'best fit' line
y = norm.pdf(bins, mu, sigma)
plt.plot(bins, y, 'r--')
plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title(r'Histogram of IQ: $\mu=100$, $\sigma=15$')
```

```
# Tweak spacing to prevent clipping of ylabel
plt.subplots_adjust(left=0.15)
plt.show()
```



[12]:
```
n
```

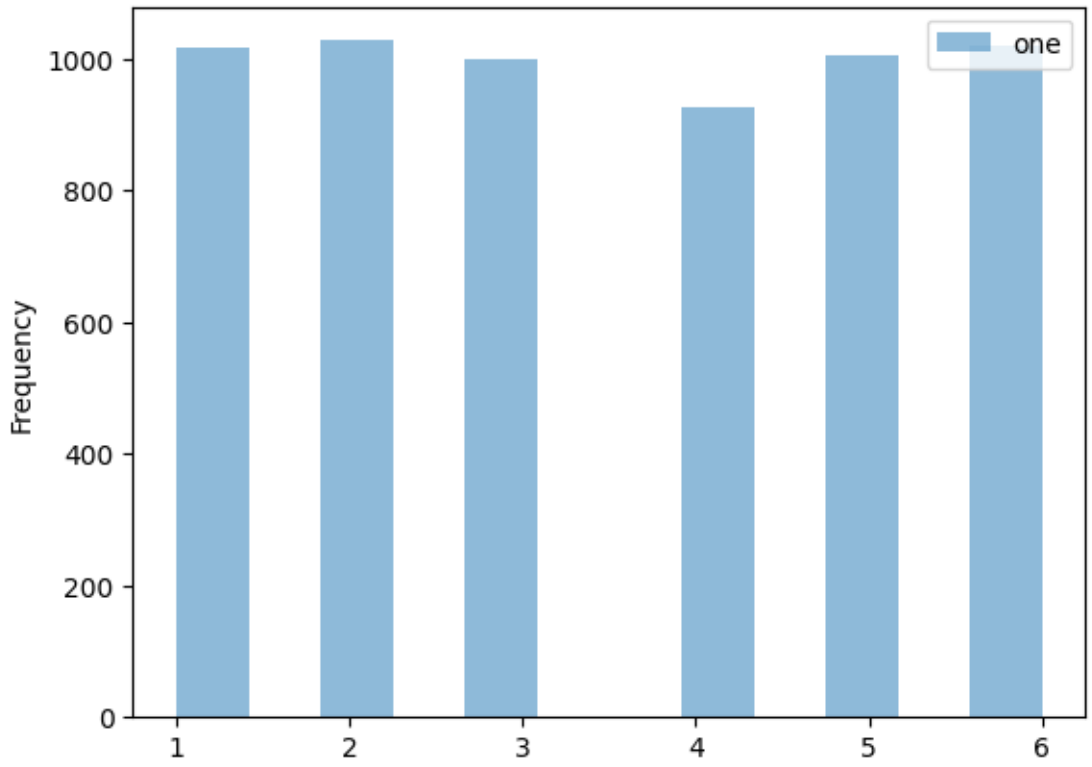
[12]:
```
array([1.80419699e-05, 1.44335759e-04, 3.60839398e-04, 1.13664410e-03,
       2.63412761e-03, 4.97958370e-03, 9.20140466e-03, 1.40907785e-02,
       2.02611322e-02, 2.55293874e-02, 2.77124658e-02, 2.39958200e-02,
       1.97018311e-02, 1.35495194e-02, 8.28126419e-03, 5.19608733e-03,
       2.20112033e-03, 9.56224405e-04, 2.88671519e-04, 1.80419699e-04])
```

[13]:
```
import pandas as pd
import numpy as np
df = pd.DataFrame(
    np.random.randint(1, 7, 6000),
    columns = ['one'])
```

[14]:
```
df
```

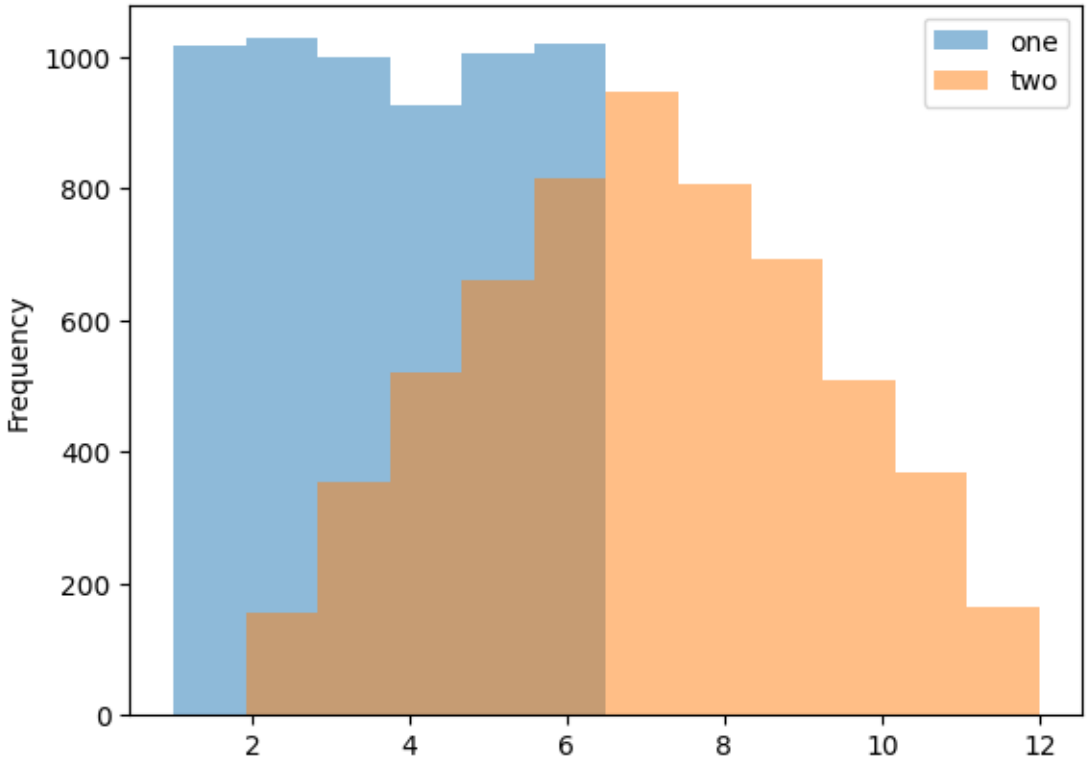
[14]:
```
      one
0       1
1       6
2       2
3       1
4       6
…     …
5995    6
5996    2
5997    1
5998    6
5999    2

[6000 rows x 1 columns]
```

[15]:
```
%matplotlib inline
ax = df.plot.hist(bins=12, alpha=0.5)
```



[16]:
```
df['two'] = df['one'] + np.random.randint(1, 7, 6000)
ax = df.plot.hist(bins=12, alpha=0.5)
9
```

[16]: 9



[ ]: