

Python

- printing formats**
- file writing [twoways]**
- plotting using matplotlib**

Programming example

```
import units
```

```
units.info()
```

```
print yards_to_meters(10.0)
```

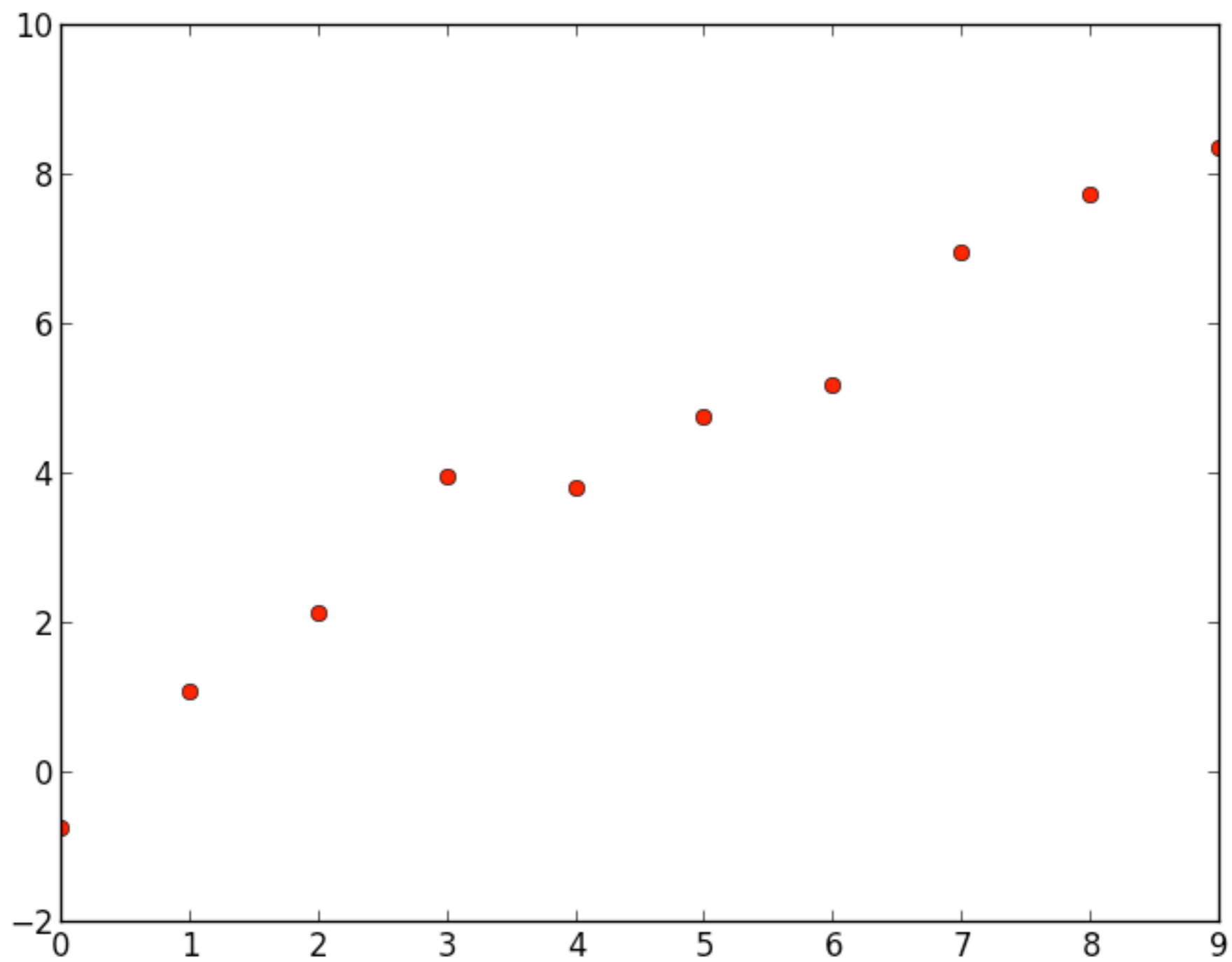
```
from units import yard_to_meter
```

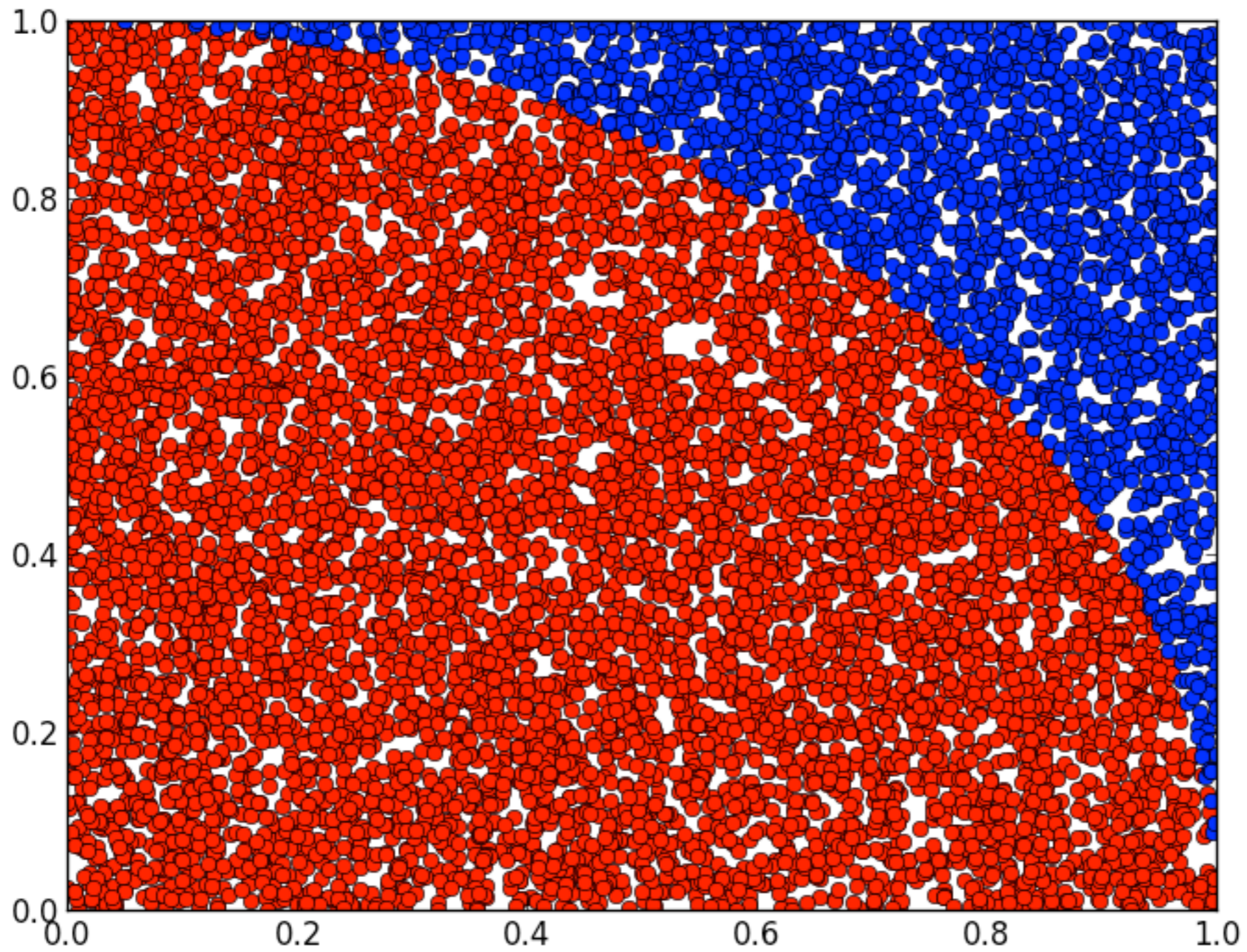
```
yard_to_meter(11)
```

```
from units import *
```

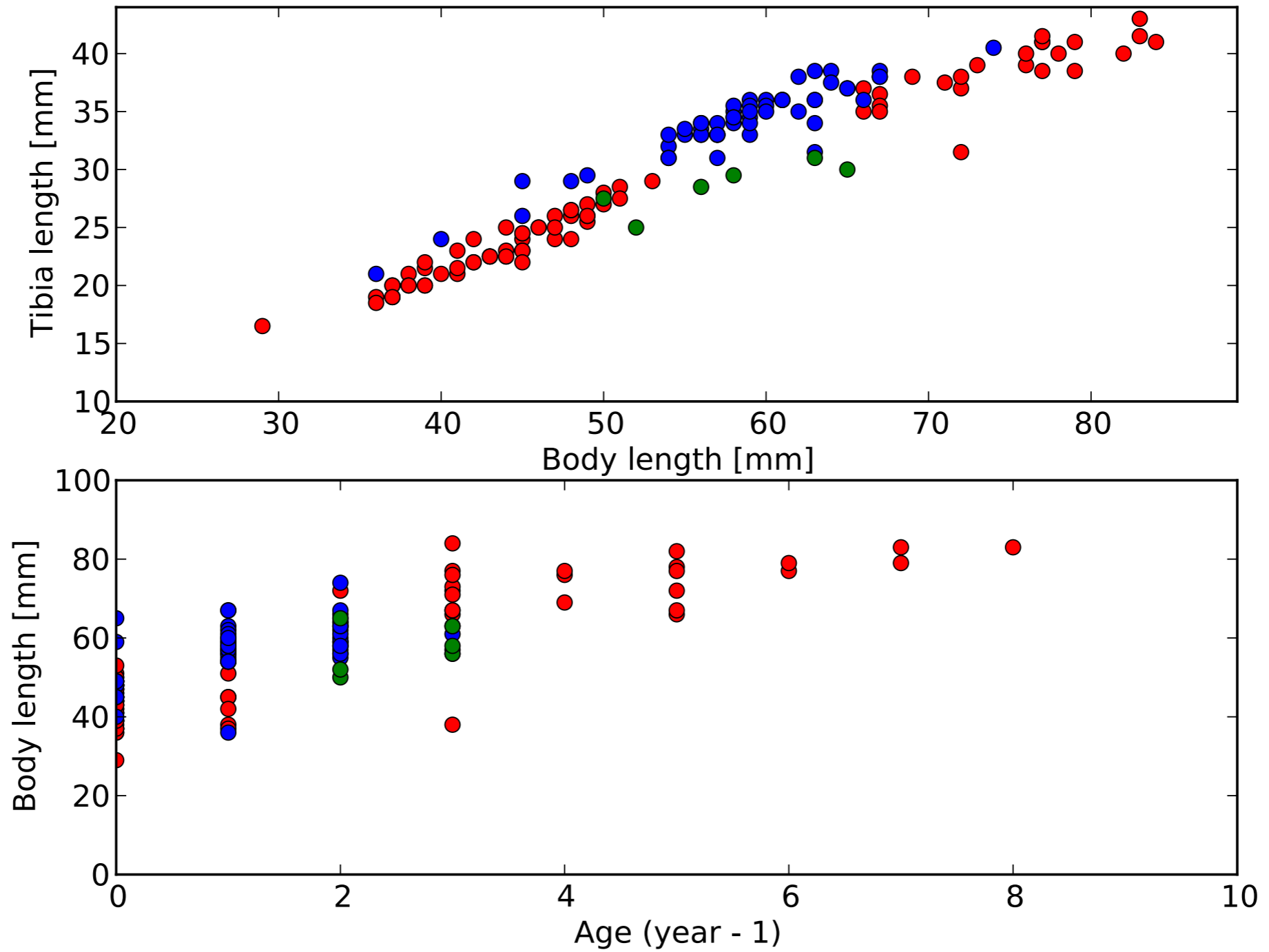
```
yard_to_meter(11)
```

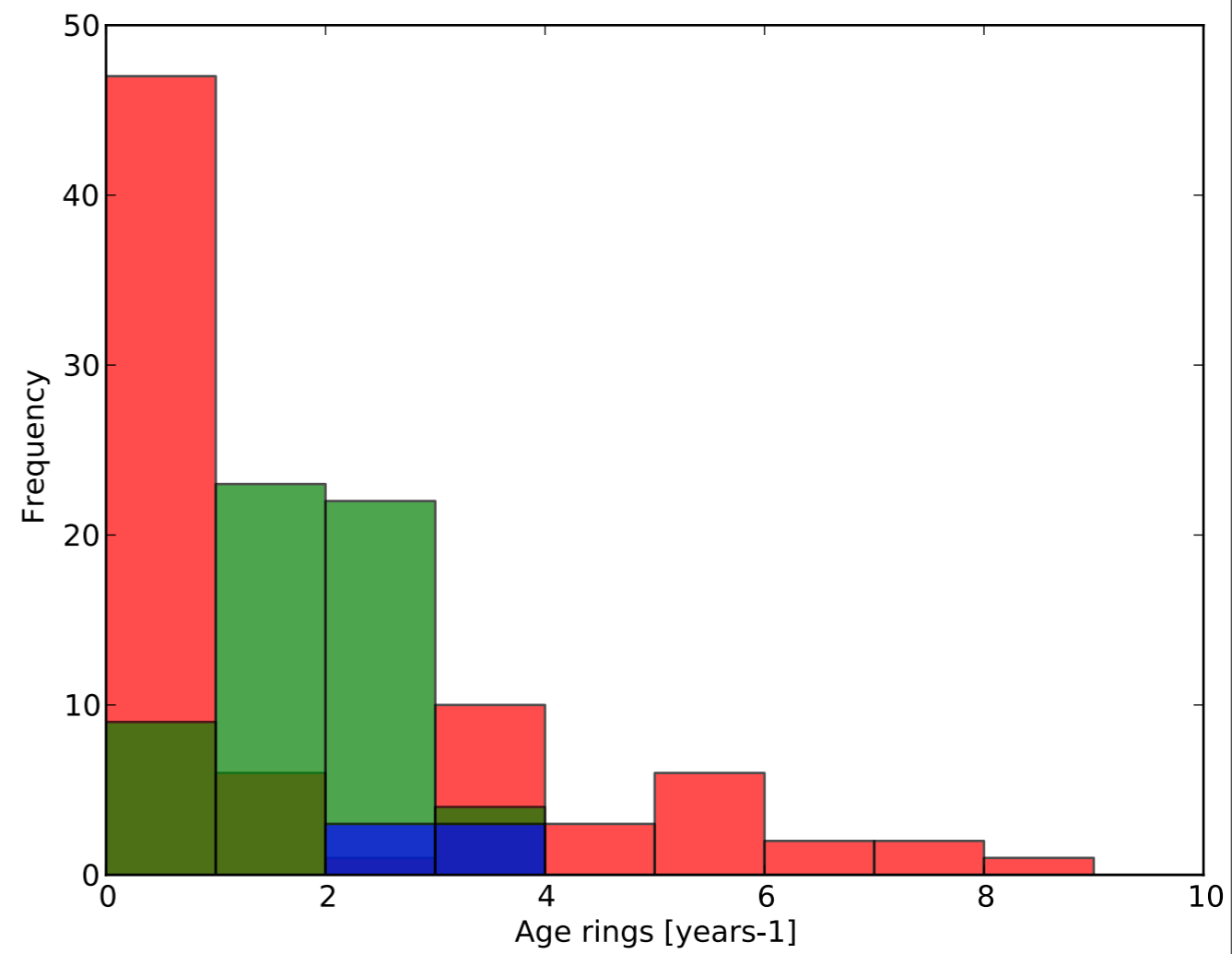
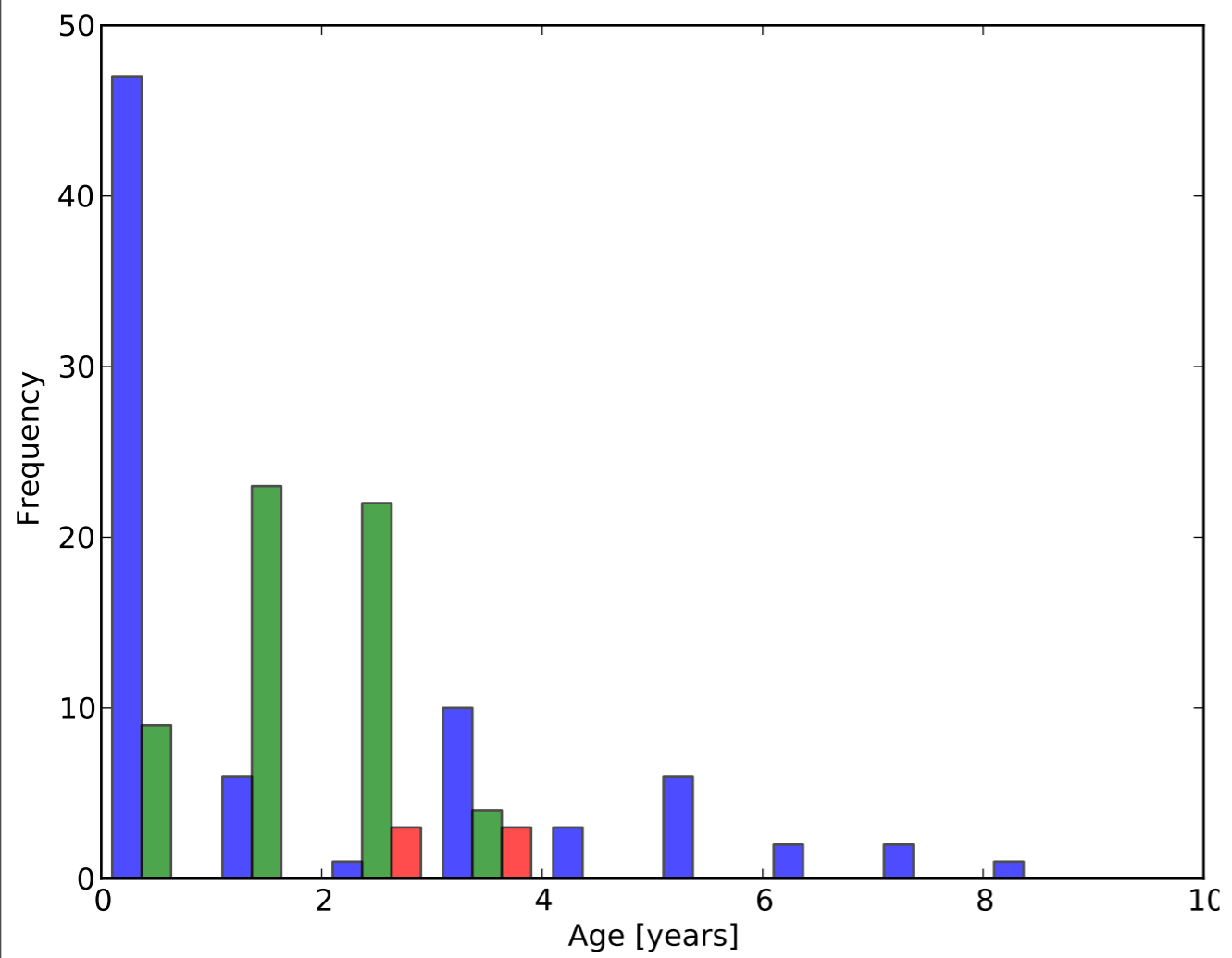
```
meter_to_yard(5.23)
```



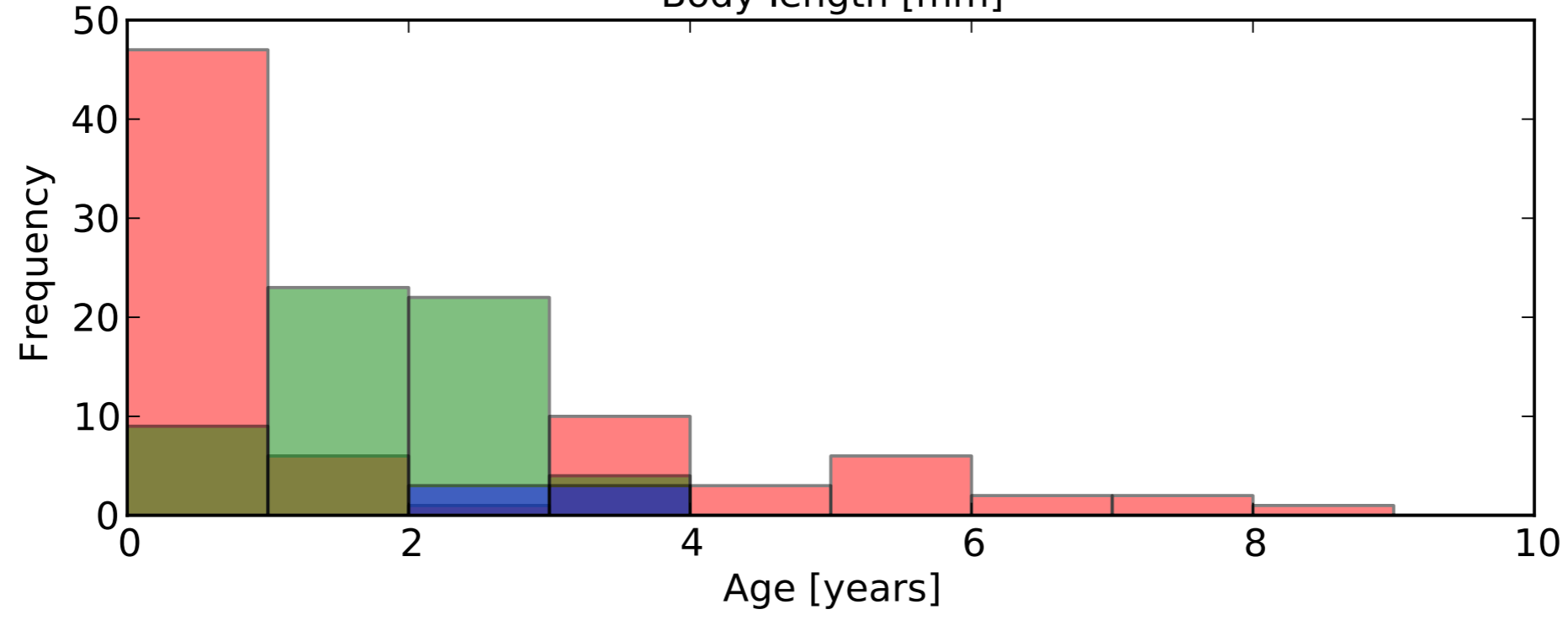
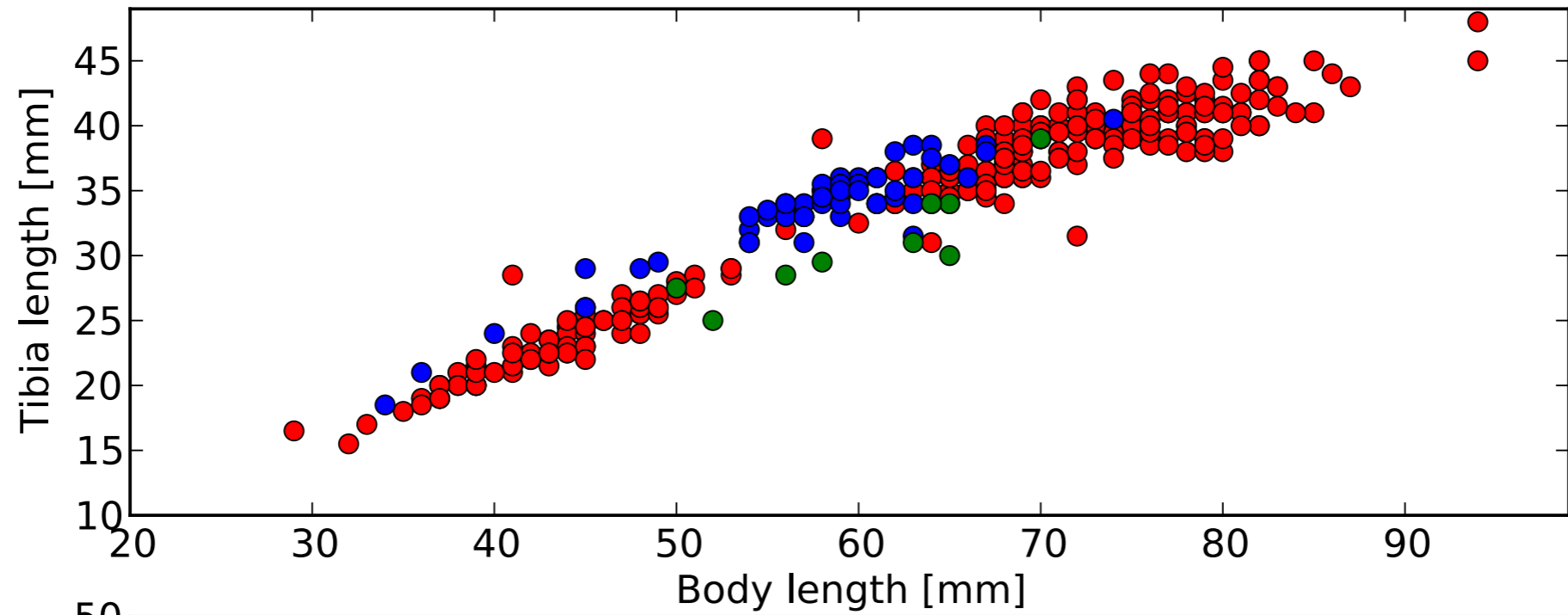


Examples of plots

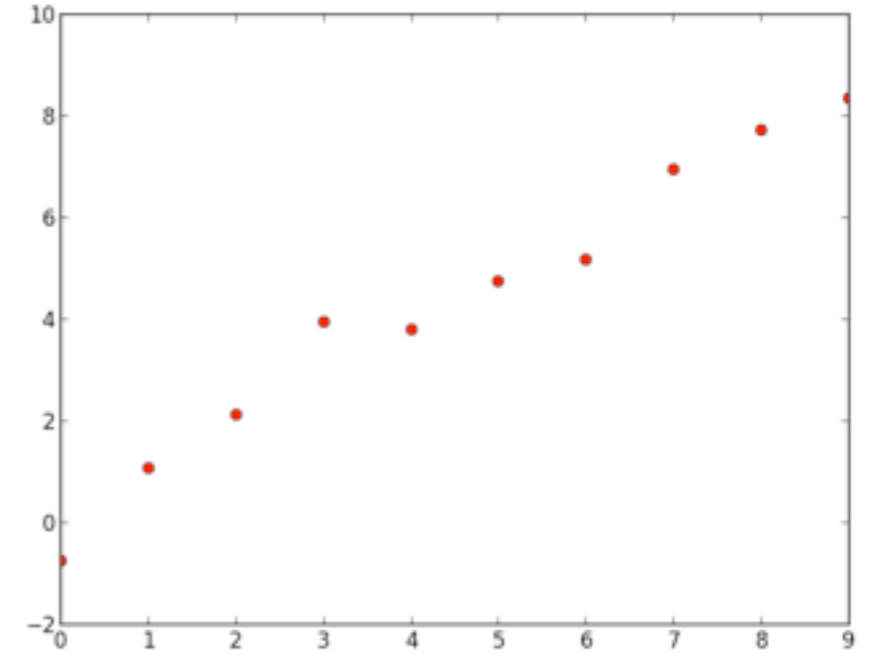




Examples of plots

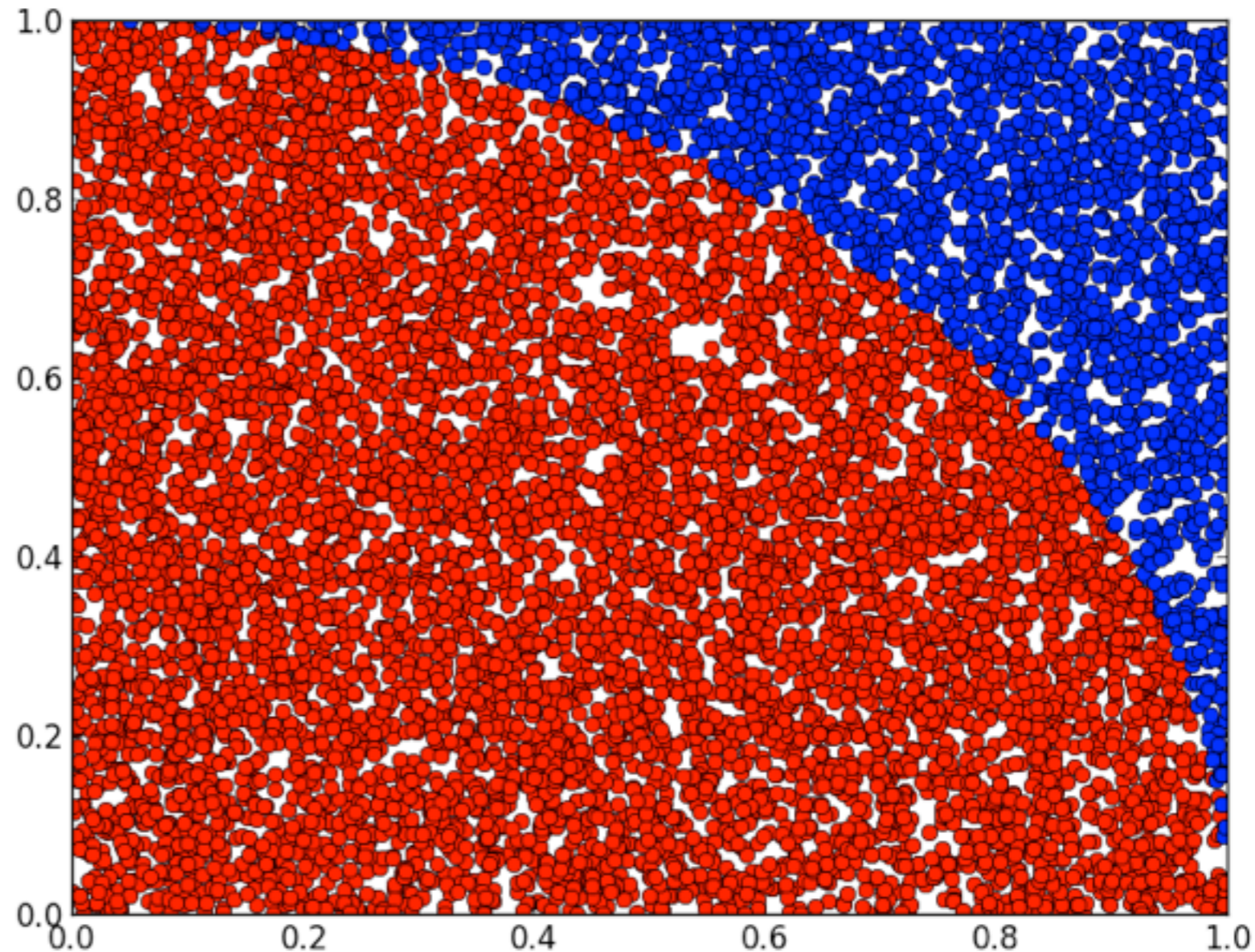


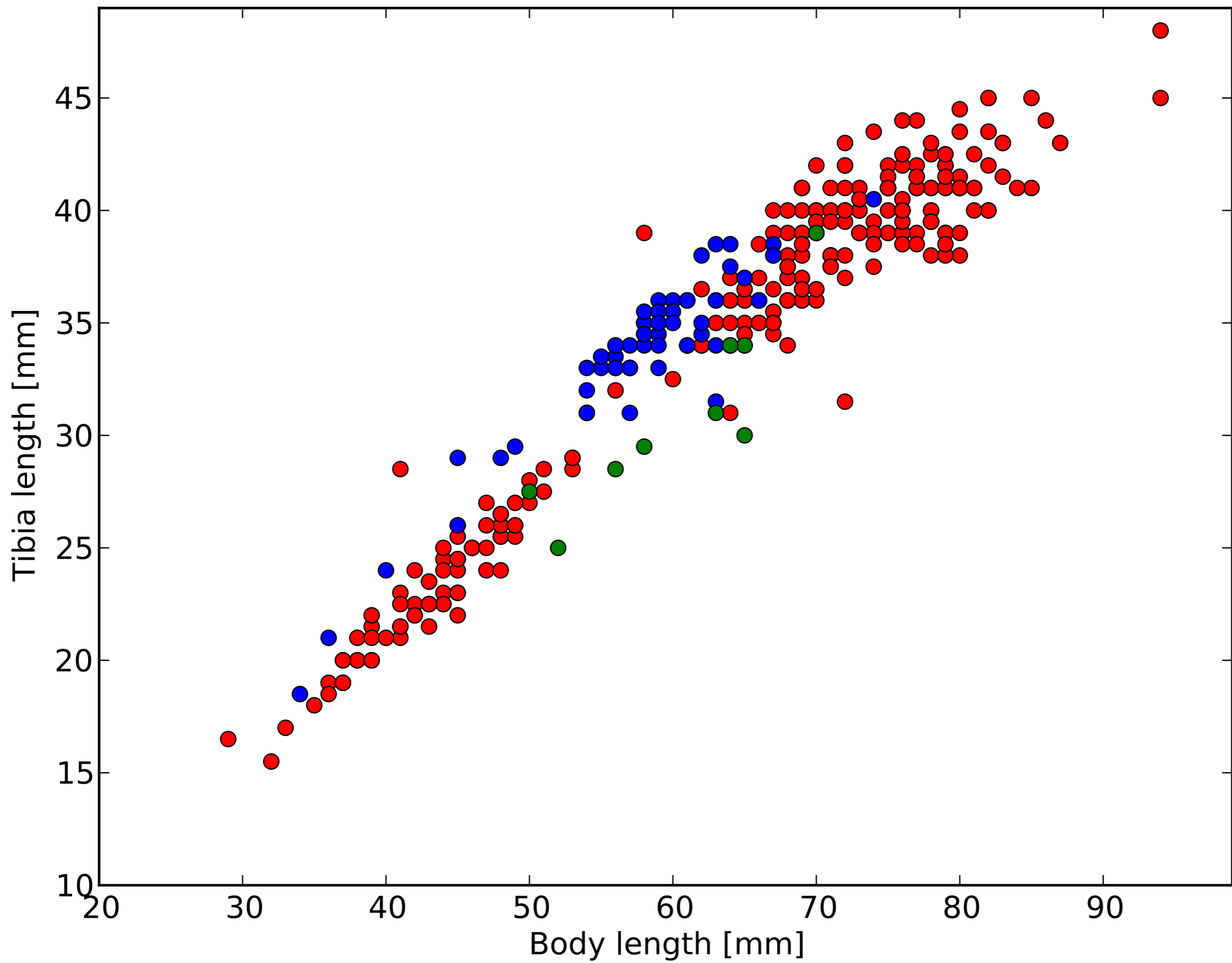
```
#!/usr/bin/env python
import sys
import matplotlib
#matplotlib.use('PDF')
import matplotlib.pyplot as plt
#from matplotlib.backends.backend_pdf import
PdfFile
import random
x = range(10)
y = range(10)
y = [yi + random.uniform(-1.0,1.0) for yi in y]
plt.figure()
plt.plot(x,y, 'ro')
#plt.savefig('plot.pdf', format='pdf')
plt.show()
```



Programming example

```
import random
import math
import sys
import matplotlib
import matplotlib.pyplot as plt
i = 0
n = 10000
r = 1.0
circle = 0.0
square = 0.0
xc=[],yc=[],xr=[],yr=[]
while i < n:
    i += 1
    x = random.uniform(0.,r)
    y = random.uniform(0.,r)
    d = math.sqrt(x*x + y*y)
    if d < r:
        xc.append(x)
        yc.append(y)
        circle += 1
    else:
        xr.append(x)
        yr.append(y)
        square += 1
print "pi=", circle/square * 4.0
plt.figure()
plt.plot(xc,yc,'ro')
plt.plot(xr,yr,'bo')
plt.show()
```



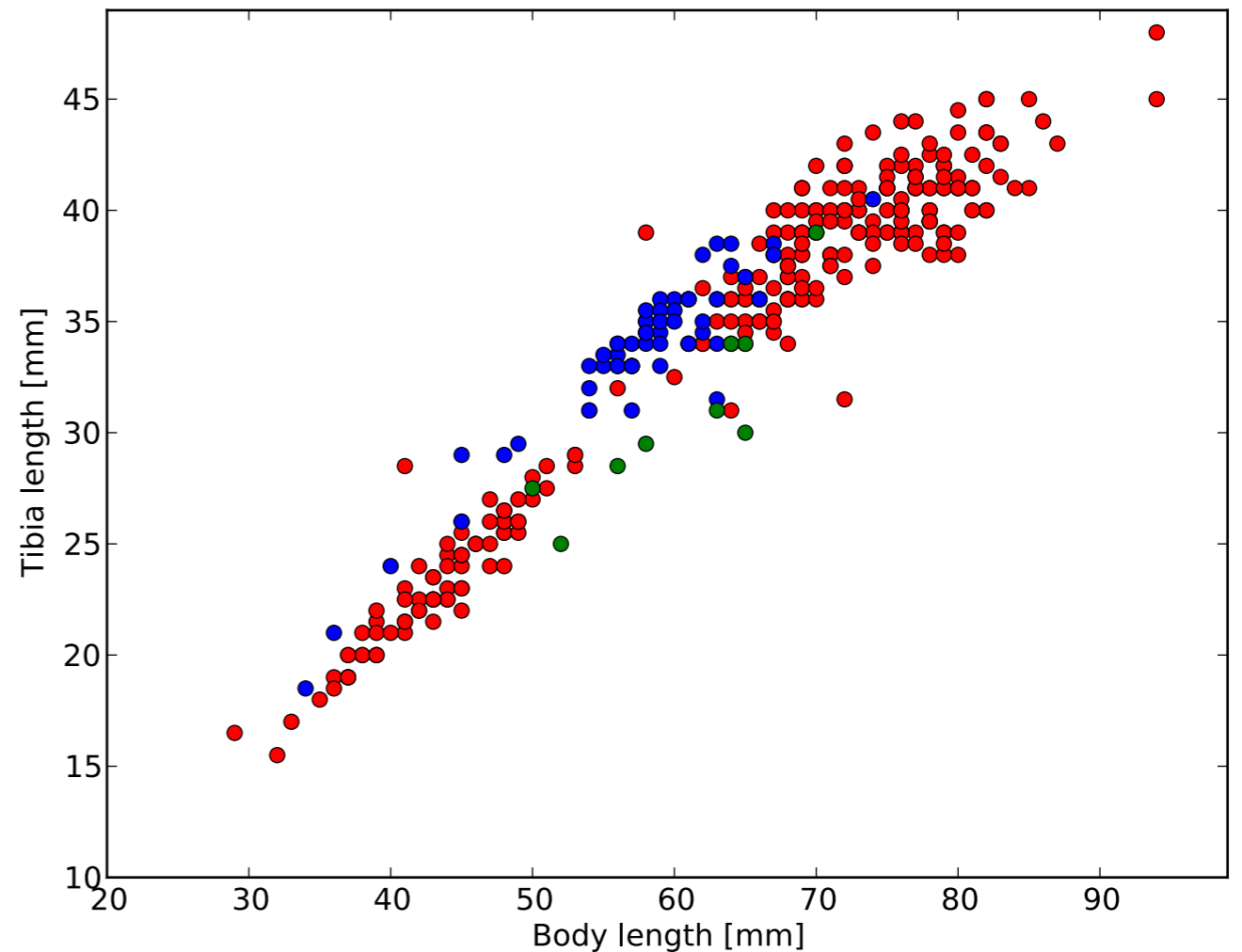


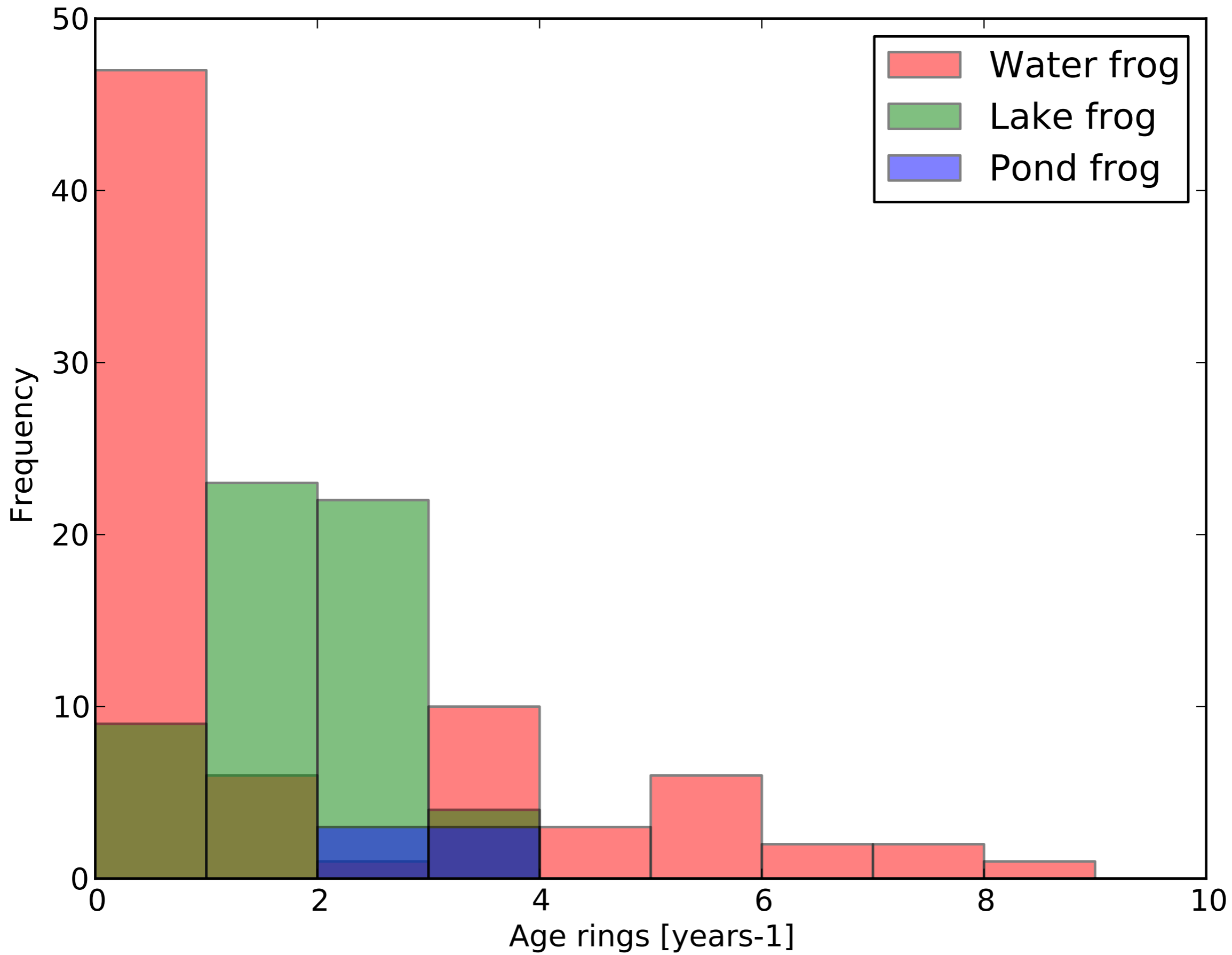
Programming example

```
#!/usr/bin/env python
import sys
import matplotlib
matplotlib.use('PDF')
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfFile
import frogs

def column(matrix,i):
    return [float(row[i]) for row in matrix]

data = frogs.read_data('swissfrogs.txt')
newdata = frogs.extract(data,[0,3,4,5])
sl = frogs.unique(newdata,0)
finaldata = frogs.partition_species(sl, newdata)
#print finaldata
xe = column(finaldata[-1],1)
ye = column(finaldata[-1],2)
xl = column(finaldata[1],1)
yl = column(finaldata[1],2)
xr = column(finaldata[0],1)
yr = column(finaldata[0],2)
xheader="Body length [mm]"
yheader="Tibia length [mm]"
xmin = 20
ymin = 10
xmax = max(xe)+5
ymax = max(ye)+1
plt.figure()
plt.plot(xe,ye,'ro')
plt.plot(xr,yr,'bo')
plt.plot(xl,yl,'go')
plt.axis([xmin,xmax,ymin,ymax])
plt.xlabel(xheader)
```





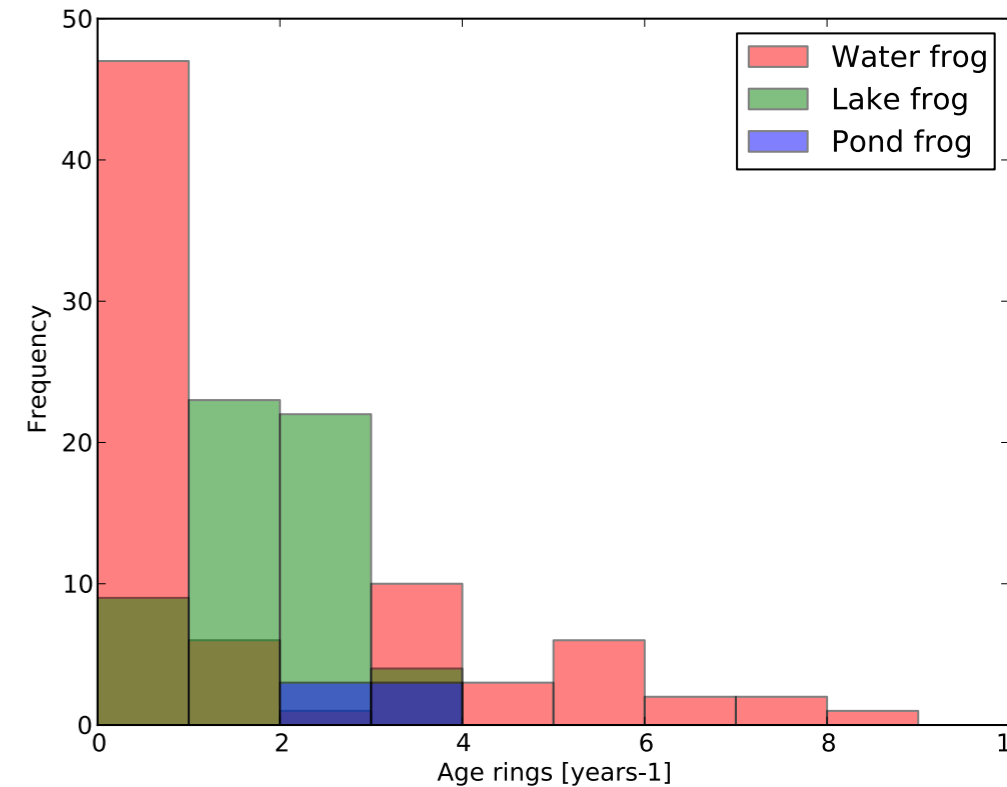
```

#!/usr/bin/env python
import sys
import matplotlib
matplotlib.use('PDF')
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfFile
import frogs

def column(matrix,i):
    return [float(row[i]) for row in matrix]

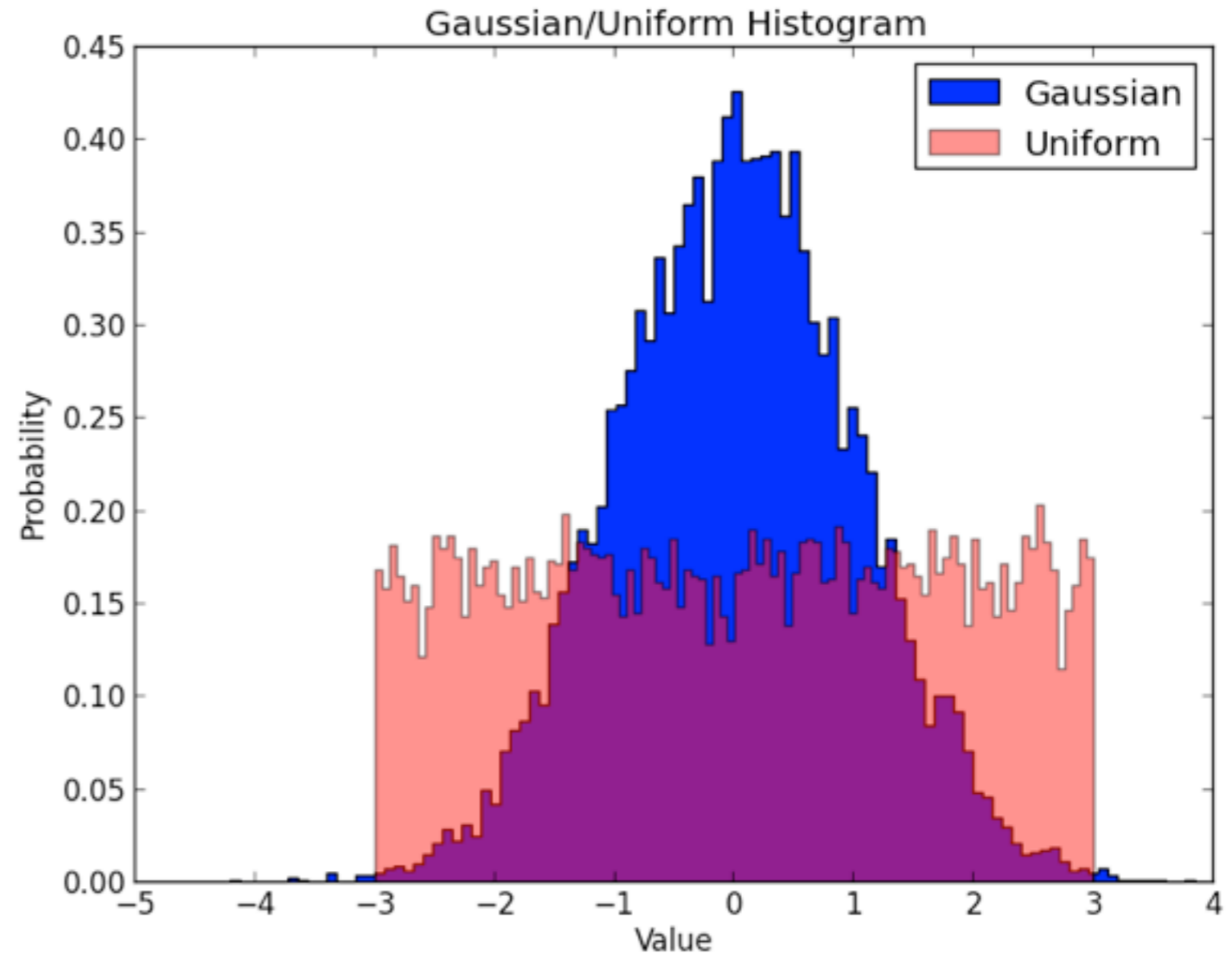
data = frogs.read_data('swissfrogs.txt')
newdata = frogs.extract(data,[0,12])
specieslist = frogs.unique(newdata,0)
finaldata = frogs.partition_species(specieslist, newdata)
xe = column(finaldata[-1],1)
xl = column(finaldata[1],1)
xr = column(finaldata[0],1)
xheader="Age rings [years-1]"
yheader="Frequency"
plt.figure()
#plt.hist([xe,xr,xl],range=[0, 10], alpha=0.7,align='mid')
plt.hist(xe,range=[0, 10], facecolor='red',
alpha=0.5,align='mid',label='Water frog')
plt.hist(xr,range=[0, 10], facecolor='green', alpha=0.5,
align='mid',label='Lake frog')
plt.hist(xl,range=[0, 10], facecolor='blue', alpha=0.5, align='mid',
label='Pond frog')
#plt.axis([0,xmax,ymin,ymax])
plt.xlabel(xheader)
plt.ylabel(yheader)
plt.legend()
plt.savefig('hist.pdf', format='pdf')

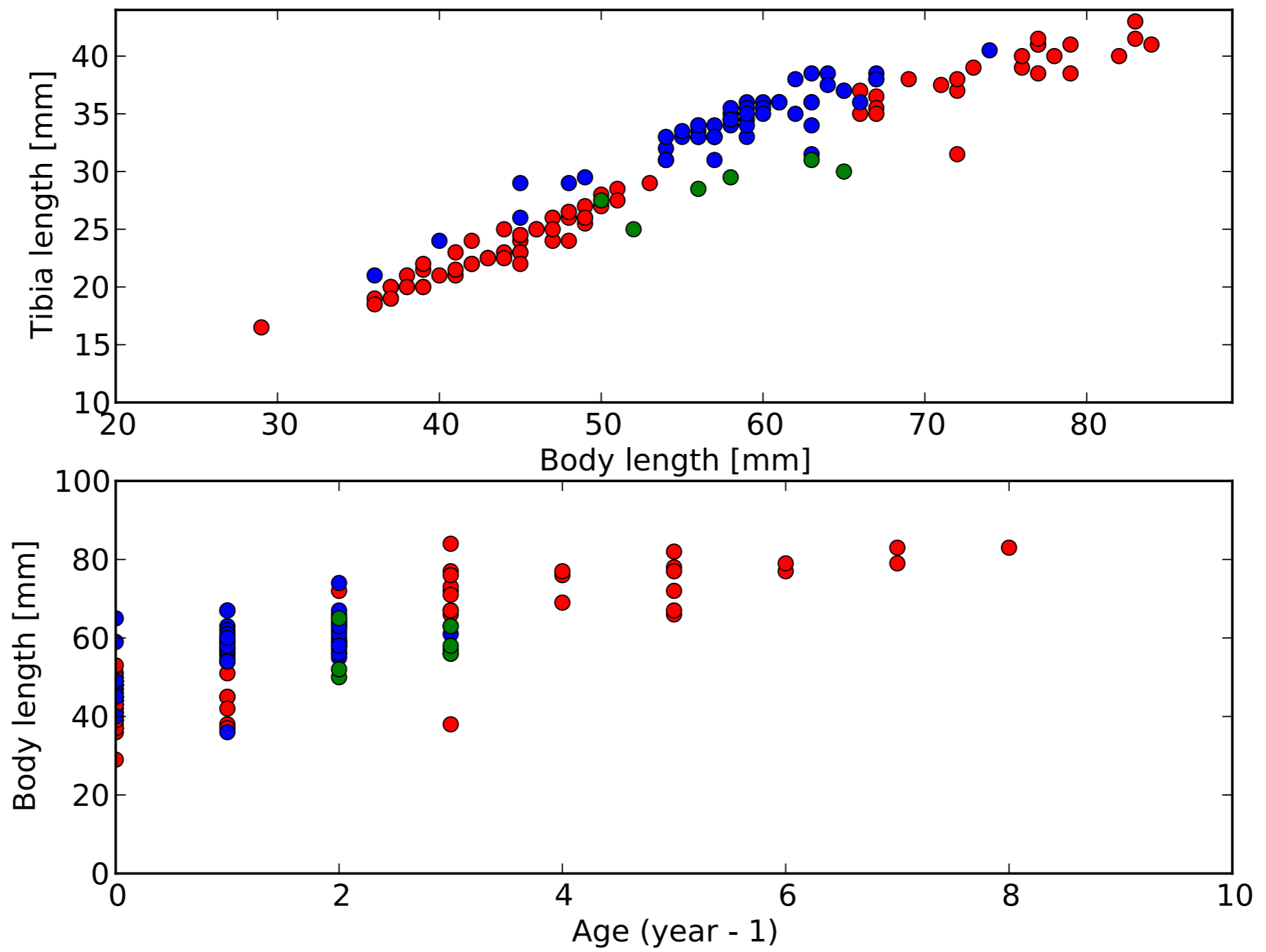
```



Examples of plots

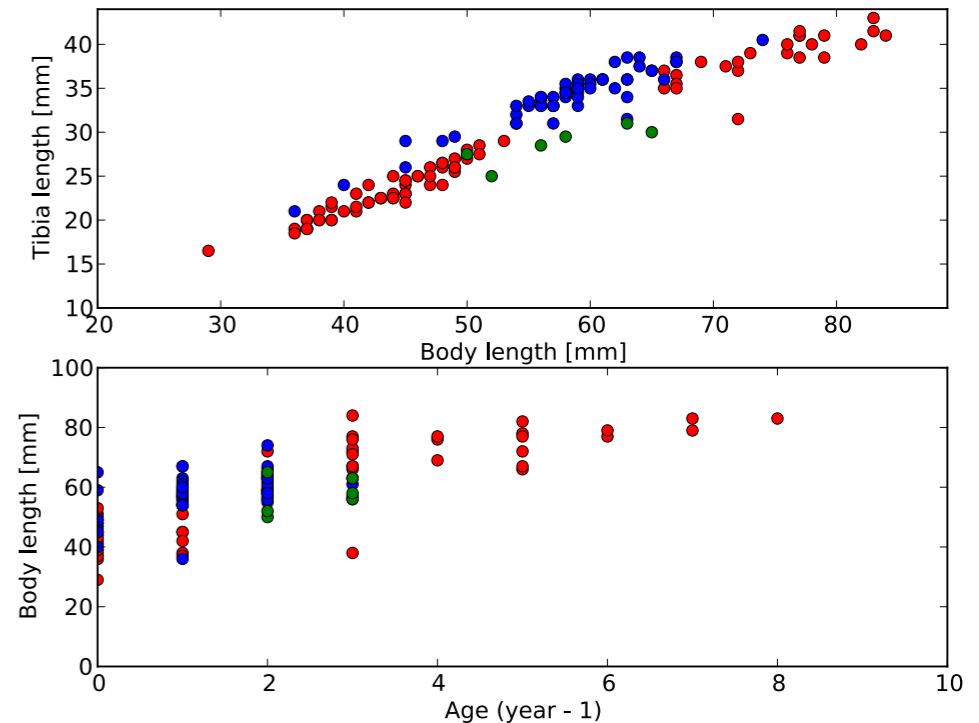
```
import matplotlib.pyplot as plt
from numpy.random import normal, uniform
gaussian_numbers = normal(size=1000)
uniform_numbers = uniform(low=-3, high=3, size=1000)
plt.hist(gaussian_numbers, bins=20, histtype='stepfilled', normed=True,
color='b', label='Gaussian')
plt.hist(uniform_numbers, bins=20, histtype='stepfilled', normed=True,
color='r', alpha=0.5, label='Uniform')
plt.title("Gaussian/Uniform Histogram")
plt.xlabel("Value")
plt.ylabel("Probability")
plt.legend()
plt.show()
```





Programming example

```
#!/usr/bin/env python
import sys
import matplotlib
matplotlib.use('PDF')
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from matplotlib.backends.backend_pdf import PdfFile
import frogs
def column(matrix,i):
    return [float(row[i]) for row in matrix]
data = frogs.read_data('swissfrogs.txt')
newdata = frogs.extract(data,[0,3,4,5,12])
specieslist = frogs.unique(newdata,0)
finaldata = frogs.partition_species(specieslist, newdata,
#print finaldata
xe = column(finaldata[-1],1)
ye = column(finaldata[-1],2)
xl = column(finaldata[1],1)
yl = column(finaldata[1],2)
xr = column(finaldata[0],1)
yr = column(finaldata[0],2)
xheader="Body length [mm]"
yheader="Tibia length [mm]"
xmin = 20
ymin = 10
xmax = max(xe)+5
ymax = max(ye)+1
# plotting instructions
```



Programming example

```
#plotting instructions
plt.figure()
gs = gridspec.GridSpec(2,1)
# set up subplots
ax1 = plt.subplot(gs[0])
ax2 = plt.subplot(gs[1])
ax1.plot(xe,ye,'ro')
ax1.plot(xr,yr,'bo')
ax1.plot(xl,y1,'go')
ax1.axis([xmin,xmax,ymin,ymax])
ax1.set_xlabel(xheader)
ax1.set_ylabel(yheader)
# second plot
xe2 = column(finaldata[-1],4)
xl2 = column(finaldata[1],4)
xr2 = column(finaldata[0],4)
print xe2
ax2.plot(xe2,xe,'ro')
ax2.plot(xr2,xr,'bo')
ax2.plot(xl2,xl,'go')
ax2.axis([0,10,0,100])
ax2.set_xlabel(xheader)
ax2.set_ylabel(yheader)
ax2.set_xlabel("Age (year - 1)")
ax2.set_ylabel(xheader)
plt.savefig('combined2.pdf', format='pdf')
```

