In [9]:

```
#Paige Brady and Austin Jones
```

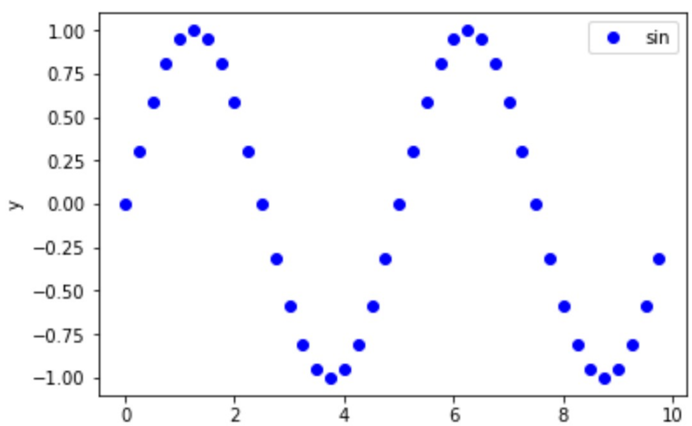
```
Populating the interactive namespace from numpy and matplotlib
```

In [15]:

```
UPPER = 10
STEP = 0.25
x = np.arange(0, UPPER, STEP)
y = sin(2*pi*x/ 5.0)
print("dumping first five entries:")
print("x[:5]:", x[:5], "...")
print("y[:5]:", np.around(y[:5],2), "...")
plt.plot(x,y,"bo", label="sin")
plt.ylabel("x")
plt.ylabel("y")
```

```
dumping first five entries:
x[:5]: [0.   0.25 0.5  0.75 1.  ] ...
y[:5]: [0.   0.31 0.59 0.81 0.95] ...
```

Out[15]: <matplotlib.legend.Legend at 0x1ef0eae3710>



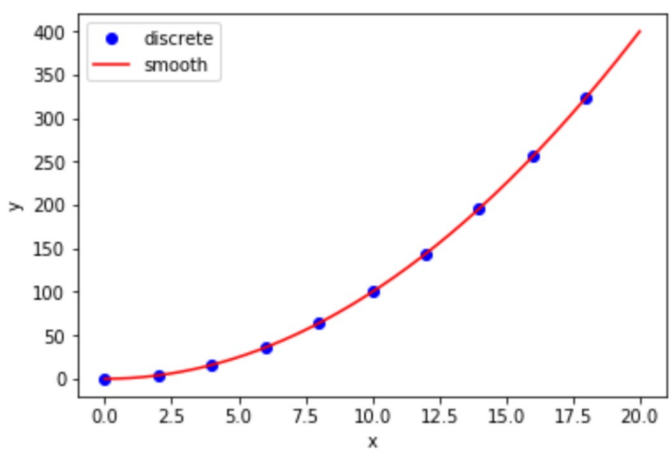
In [33]:

```
UPPER = 20
STEP = 2
x = np.arange(0, UPPER, STEP)
y = x**2
print("dumping first five entries:")
print("x[:5]:", x[:5], "...")
print("y[:5]:", np.around(y[:5],2), "...")
plt.plot(x,y,"bo", label="discrete")
plt.ylabel("x")
plt.ylabel("y")
plt.legend()


UPPER = 20
STEP = .001
xfine = np.arange(0, UPPER, STEP)
yfine = xfine**2
print("dumping first five entries:")
print("x[:5]:", xfine[:5], "...")
print("y[:5]:", np.around(yfine[:5],2), "...")
plt.plot(xfine,yfine,"r-", label="smooth")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
```

```
dumping first five entries:
x[:5]: [0 2 4 6 8] ...
y[:5]: [ 0  4 16 36 64] ...
dumping first five entries:
x[:5]: [0.    0.001 0.002 0.003 0.004] ...
y[:5]: [0. 0. 0. 0. 0.] ...
```

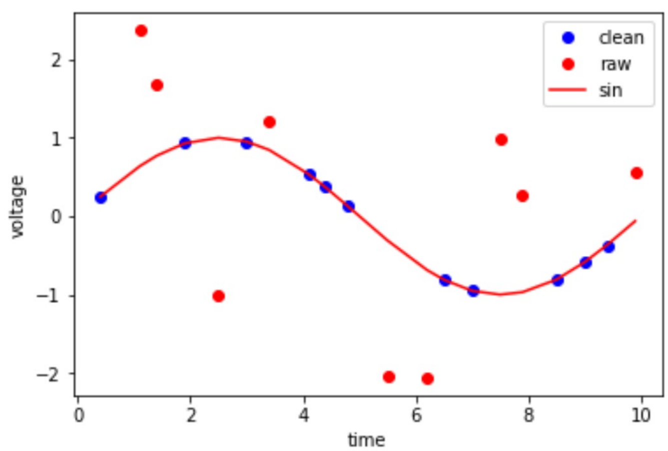
Out[33]: <matplotlib.legend.Legend at 0x1ef104200b8>



In [53]:

```
t = np.array([0.4, 1.1, 1.4, 1.9, 2.5, 3.0, 3.4, 4.1, 4.4, 4.8,
    5.5, 6.2, 6.5, 7.0, 7.5, 7.9, 8.5, 9.0, 9.4, 9.9])
v = np.array([ 0.25, 2.37, 1.69, 0.93, -1.0, 0.95, 1.22,
    0.54, 0.37, 0.13, -2.04, -2.06, -0.81, -0.95, 0.98, 0.27, -0.81, -0.59,
    -0.37, 0.56])
n = np.array([2.8, 7.3, 9.7, 1.3, 6.2, 4.8, 6.9, 4.0, 1.9, 4.0,
9.5, 8.7, 2.3, 5.3, 9.7, 8.3, 0.1, 5.1, 4.4, 9.9])
keep = (n <= 6.0)
show = (n > 6.0)
q = sin(2*pi*t/10)
plt.plot(t[keep],v[keep],"bo", label="clean")
plt.xlabel("time")
plt.ylabel("voltage")
cut = (v > 1)
plt.plot(t[show],v[show],"ro", label="raw")
plt.xlabel("time")
plt.ylabel("voltage")
plt.legend()
plt.plot(t,q,"r-", label="sin")
plt.xlabel("time")
plt.ylabel("voltage")
plt.legend()
print("cut: ", cut)
print("v subject to cut: ", v[cut])
print("t subject to cut: ", t[cut])
print("n subject to cut: ", n[cut])
```

```
cut:  [False  True  True False False False  True False False False False False
 False False False False False False False False]
v subject to cut:  [2.37 1.69 1.22]
t subject to cut:  [1.1 1.4 3.4]
n subject to cut:  [7.3 9.7 6.9]
```

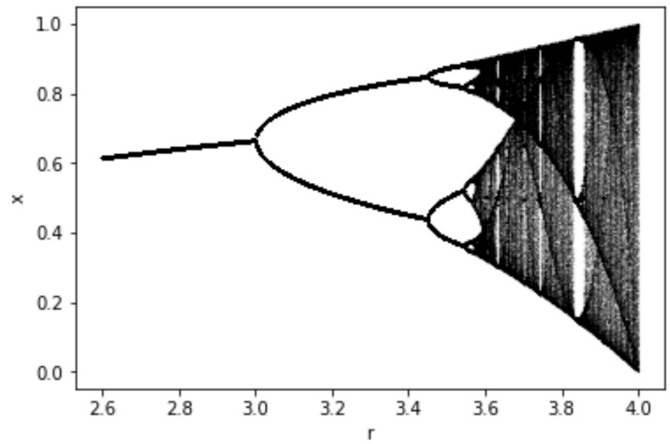


In [65]:

```
r = np.arange(2.6, 4.0, 0.002)
R_SIZE = r.size
x = np.full(R_SIZE, 0.01)
ITER = 10000
PLOT = 1000
for i in range(ITER):
    x = r * x * (1.0 - x)
for n in range(PLOT):
    x = r * x * (1.0 - x)
    plt.scatter(r, x, s = 0.001, color = "black")
plt.xlabel("r")
plt.ylabel("x")

```

Out[65]: Text(0, 0.5, 'x')



In [ ]: