

In [1]:

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

In [66]:

```
np.random.seed(98)  # a not-so-random starting seed

xy = np.random.rand(2, 1e3) * 2 - 1.0
r = np.sqrt(xy[0, :]**2 + xy[1, :]**2)
```

```
/usr/local/lib/python3.4/site-packages/ipykernel/__main__.py:3: DeprecationWarning: using a non-integer number instead of an integer will result in an error in the future
  app.launch_new_instance()
```

In [67]:

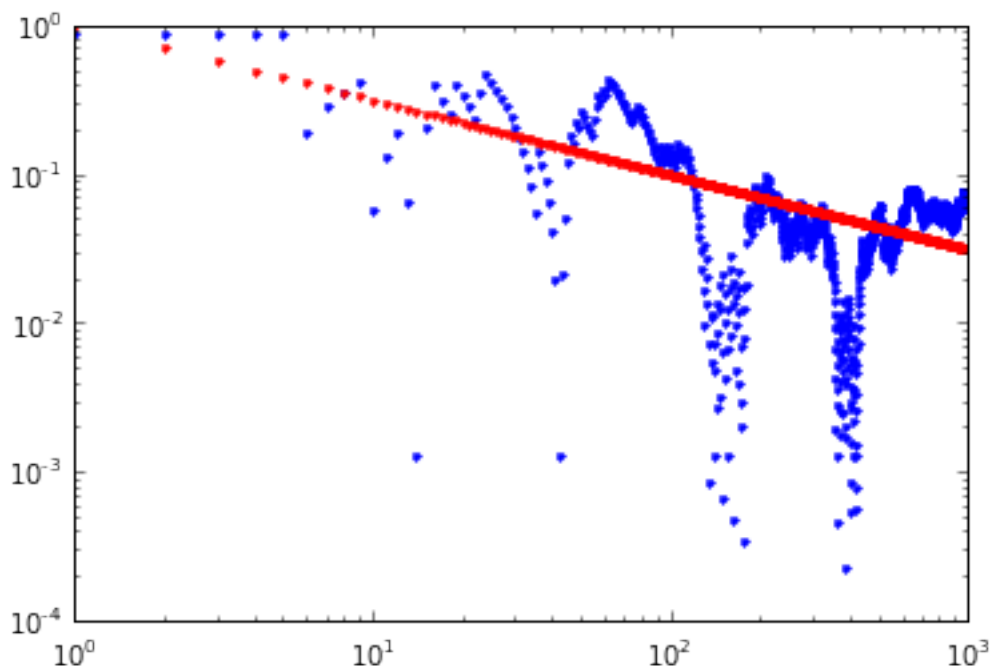
```
incircle = r<=1.0
countincircle = 4 * incircle.cumsum() / np.arange(1, len(r)+1)
```

In [68]:

```
error = np.abs(countincircle - np.pi)
plt.loglog(np.arange(1, len(r)+1), error, '.')
plt.loglog(np.arange(1, len(r)+1), np.arange(1, len(r)+1)**(-0.5), 'r.')
#plt.semilogy(np.arange(1, len(r)+1), countincircle - np.pi * np.ones(r.shape))
```

Out[68]:

[<matplotlib.lines.Line2D at 0x15cd13550>]

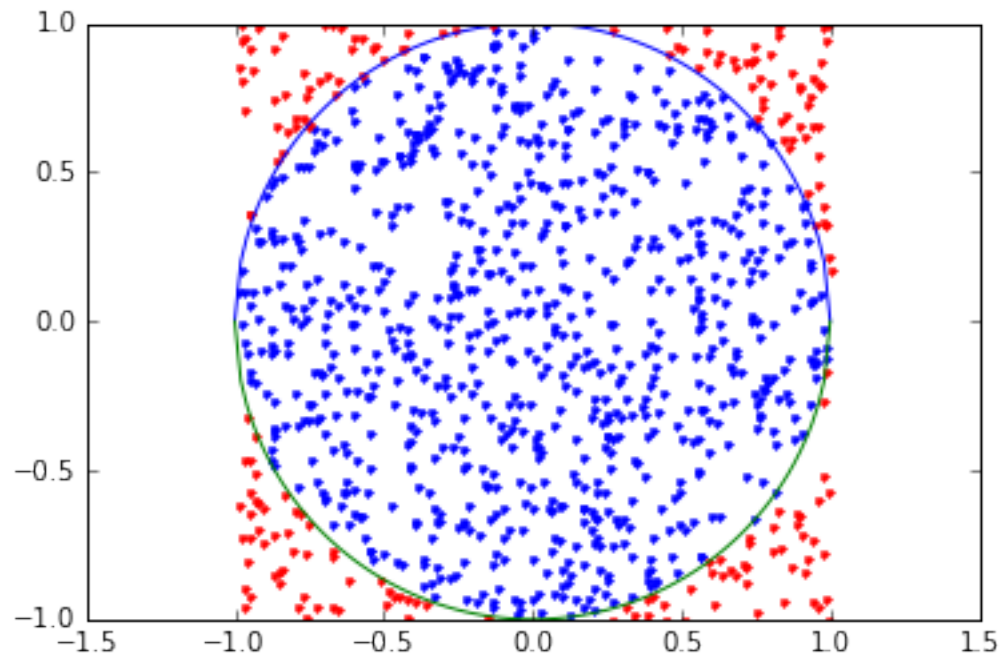


In [72]:

```
plt.plot(xy[0,np.where(incircle)[0]], xy[1,np.where(incircle)[0]], 'b.')
plt.plot(xy[0,np.where(incircle==False)[0]], xy[1,np.where(incircle==False)[0]],
'r.')
xx = np.linspace(-1,1,100)
plt.plot(xx, np.sqrt(1-xx**2), xx, -np.sqrt(1-xx**2))
plt.axis('equal')
```

Out[72]:

(-1.0, 1.0, -1.0, 1.0)



In [ ]:

In [ ]: