Astro518_HW_01

September 2, 2014

In[1]:
```
import numpy as np
import matplotlib.pyplot as plt
n = np.linspace(0,10,11)
print n
```

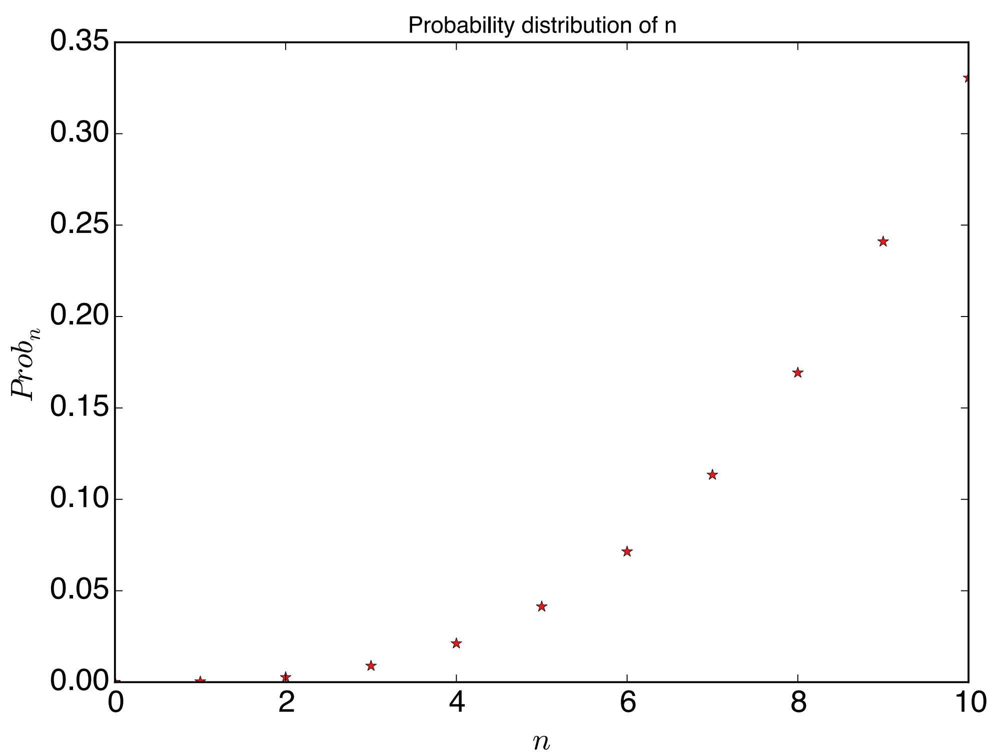
```
[  0.   1.   2.   3.   4.   5.   6.   7.   8.   9.  10.]
```

In[2]:
```
Prob_n = (n/10)**3/np.sum((n/10)**3)
print Prob_n, np.sum(Prob_n)
```

```
[ 0.          0.00033058  0.00264463  0.00892562  0.02115702  0.04132231
  0.07140496  0.11338843  0.1692562   0.24099174  0.33057851] 1.0
```

In[3]:
```
plt.plot(n, Prob_n, '*')
plt.title('Probability distribution of n')
plt.xlabel('$n$')
plt.ylabel('$Prob_n$')
```

Out[3]: <matplotlib.text.Text at 0x10ac28d50>



In[4]:
```
P = sum((n/10)**3*Prob_n) # the final probability
print P
```

```
0.654018181818
```

0.1 Estimation Sample Size Needed

0.2 Generating Fake Data

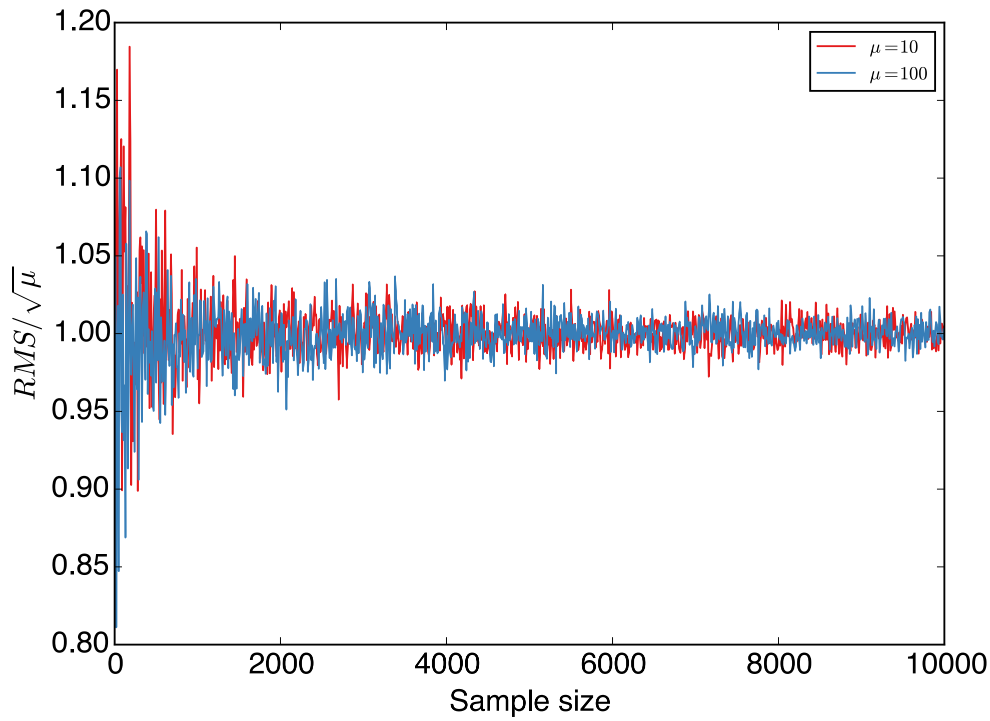
In [5]:
```
# mu = 10
dist1 = np.random.poisson(lam = 10.0, size = 1000)
average1 = np.mean(dist1)
rms1 = np.std(dist1, ddof = 1)
print "for mu = 10"
print "average = ", average1
print "RMS = ", rms1

# mu = 100
dist2 = np.random.poisson(lam = 100.0, size = 1000)
average2 = np.mean(dist2)
rms2 = np.std(dist2, ddof = 1)
print "for mu = 100"
print "average = ", average2
print "RMS = ", rms2
```

```
for mu = 10
average =  10.098
RMS =  3.21196399617
for mu = 100
average =  100.056
RMS =  9.89862853363
```

0.2.1 RMS/$\sqrt{\mu}$ changes with sample size

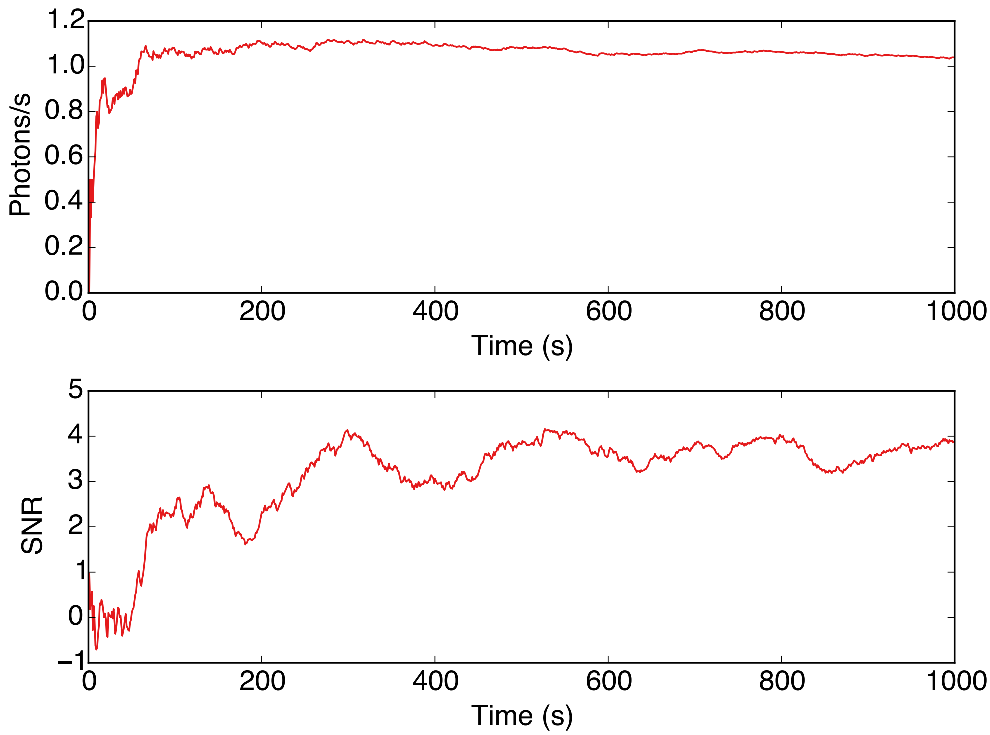
In [9]:
```
rms = lambda lam, size: np.std(np.random.poisson(lam,size), ddof = 1)
sample_size = np.linspace(10, 10000, 1000)
rms_10 = map(lambda size: rms(10, size), sample_size)
rms_100 = map(lambda size: rms(100,size), sample_size)
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(sample_size, rms_10/np.sqrt(10), label = '$\mu = 10$')
ax.plot(sample_size, rms_100/np.sqrt(100), label = '$\mu = 100$')
ax.set_xlabel("Sample size")
ax.set_ylabel("$RMS/\sqrt{\mu}$")
ax.legend()
plt.show()
```



0.2.2 Simulating the detection of photons

In[7]:
```
#photon from the source, 0-1000s
source = np.random.poisson(1.0, size = 1000)
#photon from sky
sky = np.random.poisson(30, size = 1000)
sky2 = np.random.poisson(30, size = 1000) # flux from the sky when there is no source
#time series
t = np.linspace(1,1000,1000)
```

In[8]:
```
#plots
fig = plt.figure()
# source photons/s
ax1 = fig.add_subplot(211)
ax1.plot(t, np.cumsum(source + sky - sky)/t)
ax1.set_xlabel('Time (s)')
ax1.set_ylabel('Photons/s')
# SNR as a function of time
ax2 = fig.add_subplot(212)
ax2.plot(t, np.cumsum(source + sky - sky2)/np.sqrt(np.cumsum(sky + sky2 + source)))
ax2.set_xlabel('Time (s)')
ax2.set_ylabel('SNR')
fig.tight_layout()
plt.show()
```



Photons/s
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
Time (s)
SNR
5
4
3
2
1
0
−1
0
200
400
600
800
1000
Time (s)

In [ ]: