00_poisson

June 19, 2023

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

0.0.1 Poisson Statistics

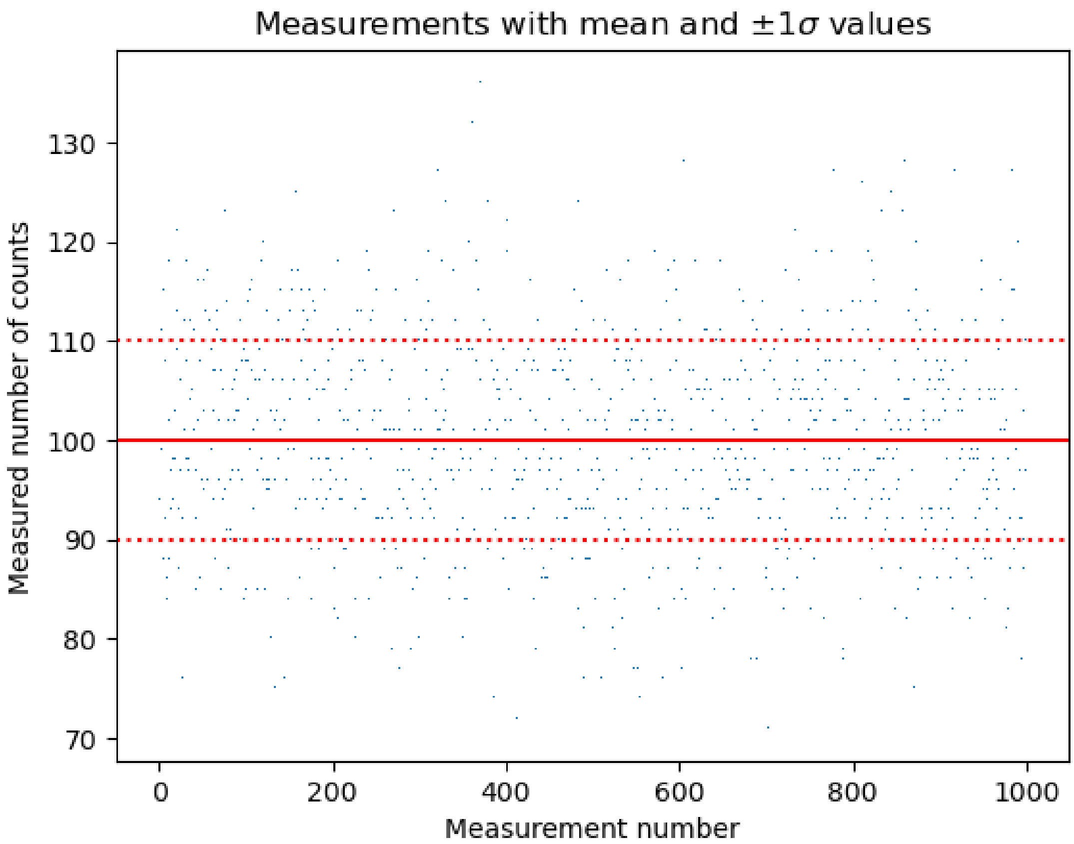
Consider a signal rate of λ counts in the measurement time, with (for $\lambda >> 1$).

The standard deviation of a single measurement is $\sigma = \sqrt{\lambda}$ so the signal to noise ratio is $\lambda/\sqrt{\lambda} = \sqrt{\lambda}$.

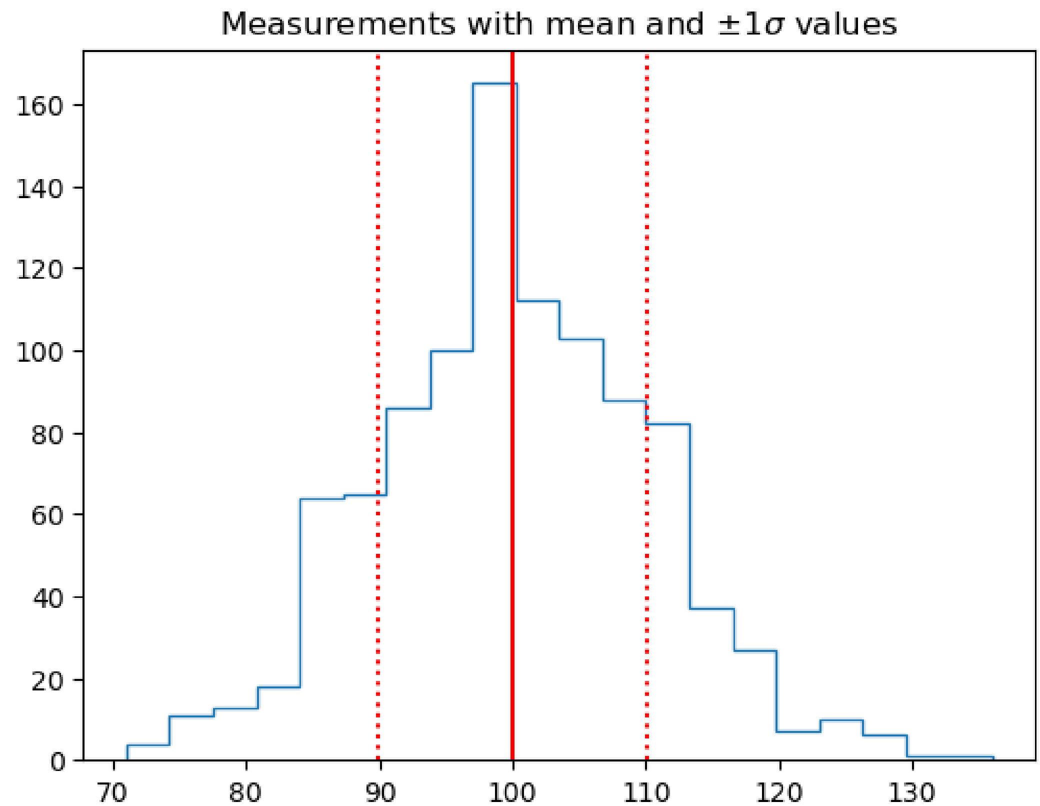
[2]:
```
np.random.seed(1234599)
```

[3]:
```
# Repeat the experiment nMeasurements time, with mean number of expected counts
mean = 100
nMeasurements = 1000
measurements = np.random.poisson(lam = mean, size=nMeasurements)
```

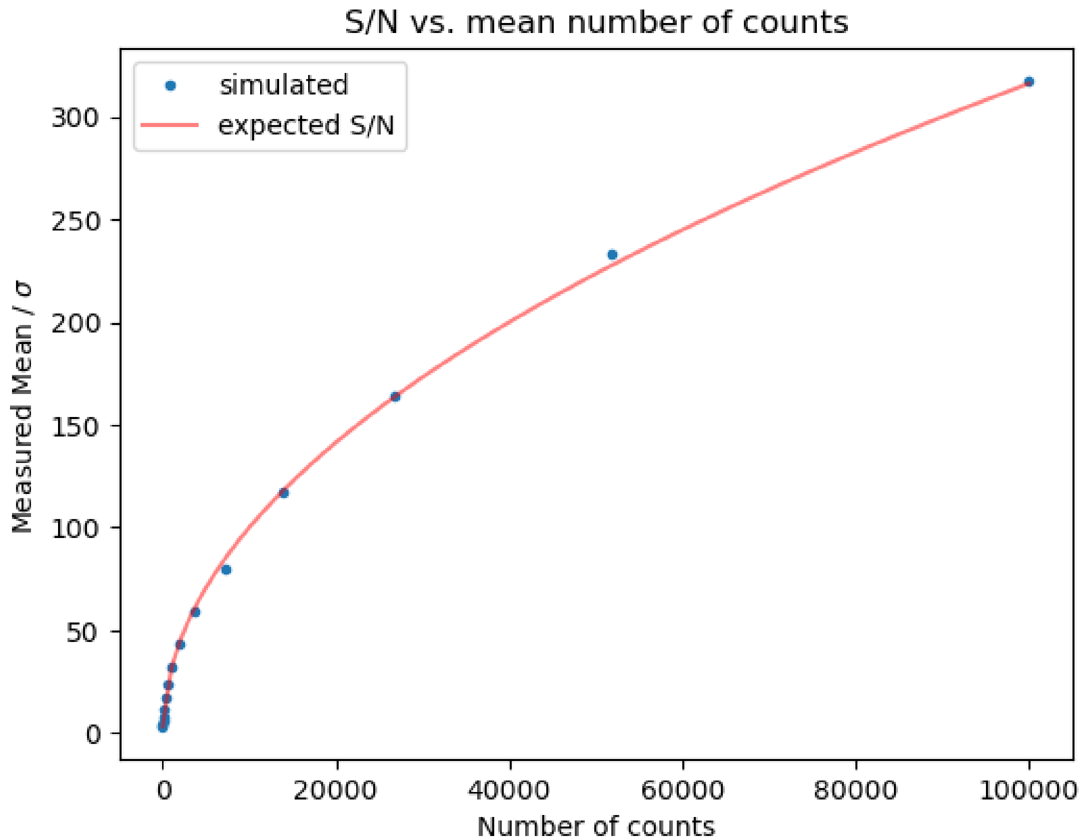
[4]:
```
plt.plot(measurements, ',')
std = measurements.std()
plt.axhline(mean, c='r')
plt.axhline(mean+std, c='r', linestyle=":")
plt.axhline(mean-std, c='r', linestyle=":")
plt.xlabel("Measurement number")
plt.ylabel("Measured number of counts")
_ = plt.title("Measurements with mean and $\pm 1 \sigma$ values")
```



[5]:
```
plt.hist(measurements, histtype='step', bins=20)
plt.axvline(mean, c='r')
plt.axvline(mean+std, c='r', linestyle=":")
plt.axvline(mean-std, c='r', linestyle=":")
_ = plt.title("Measurements with mean and $\pm 1 \sigma$ values")
```



[6]:
```
rates = np.logspace(1,5,num=15)
stds = np.zeros(len(rates))
means = np.zeros(len(rates))
for i,rate in enumerate(rates):
    measurements = np.random.poisson(lam = rate, size=nMeasurements)
    means[i] =  measurements.mean()
    stds[i] = measurements.std()
plt.plot(rates,means/stds,'.', label="simulated")
xFits = np.linspace(rates.min(),rates.max(),100)
yFits = np.sqrt(xFits)
plt.plot(xFits,yFits, 'r-',alpha=0.5, label="expected S/N")
plt.xlabel("Number of counts")
plt.ylabel("Measured Mean / $\sigma$")
plt.title("S/N vs. mean number of counts")
_ = plt.legend()
```


S/N vs. mean number of counts
simulated
expected S/N
Measured Mean / σ
Number of counts
0
50
100
150
200
250
300
0
20000
40000
60000
80000
100000

[ ]: