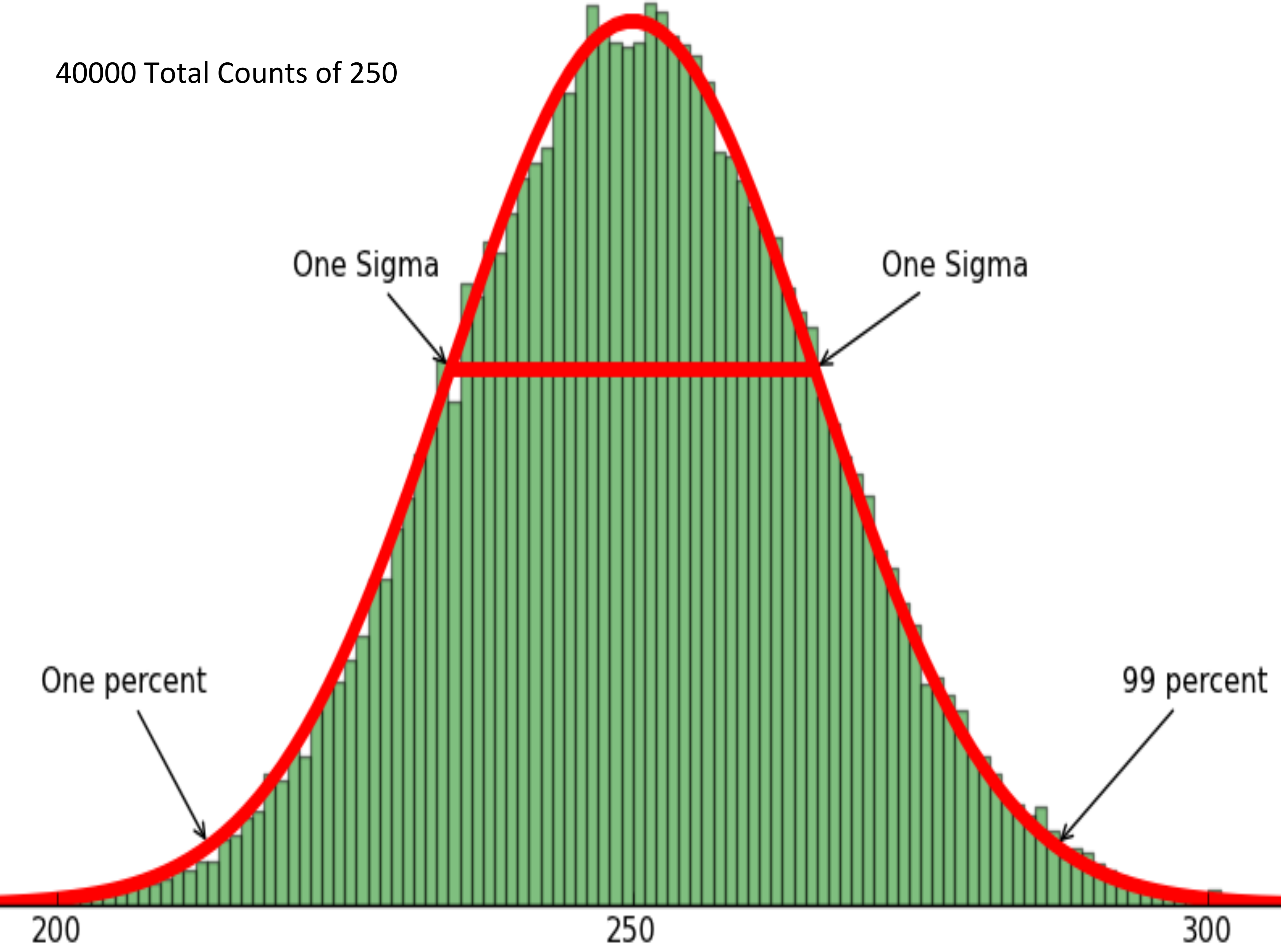
Plotting the 40,000 trials

- figure("Count 25 Events 40000 times")
- Numcounts25, binedges25, patches = hist(Counters25, bins = 50, range = (0,50), color = "green", alpha = 0.5)
- centers25 = 0.5*(binedges25[1:] + binedges25[:-1])
- y25 = 40000 * Normal(centers25, 25, sqrt(25))
- xbar25 = np.zeros(2)
- ybar25 = np.zeros(2)
- xbar25[0] = 25 - sqrt(25)
- xbar25[1] = 25 + sqrt(25)
- ybar25 = 40000*Normal(xbar25, 25, sqrt(25))
- plot(xbar25, ybar25, color= "red", alpha = 1.0, lw =5)
- plot(centers25, y25, alpha = 1.0, color = "red", lw =5)

Adding Annotations

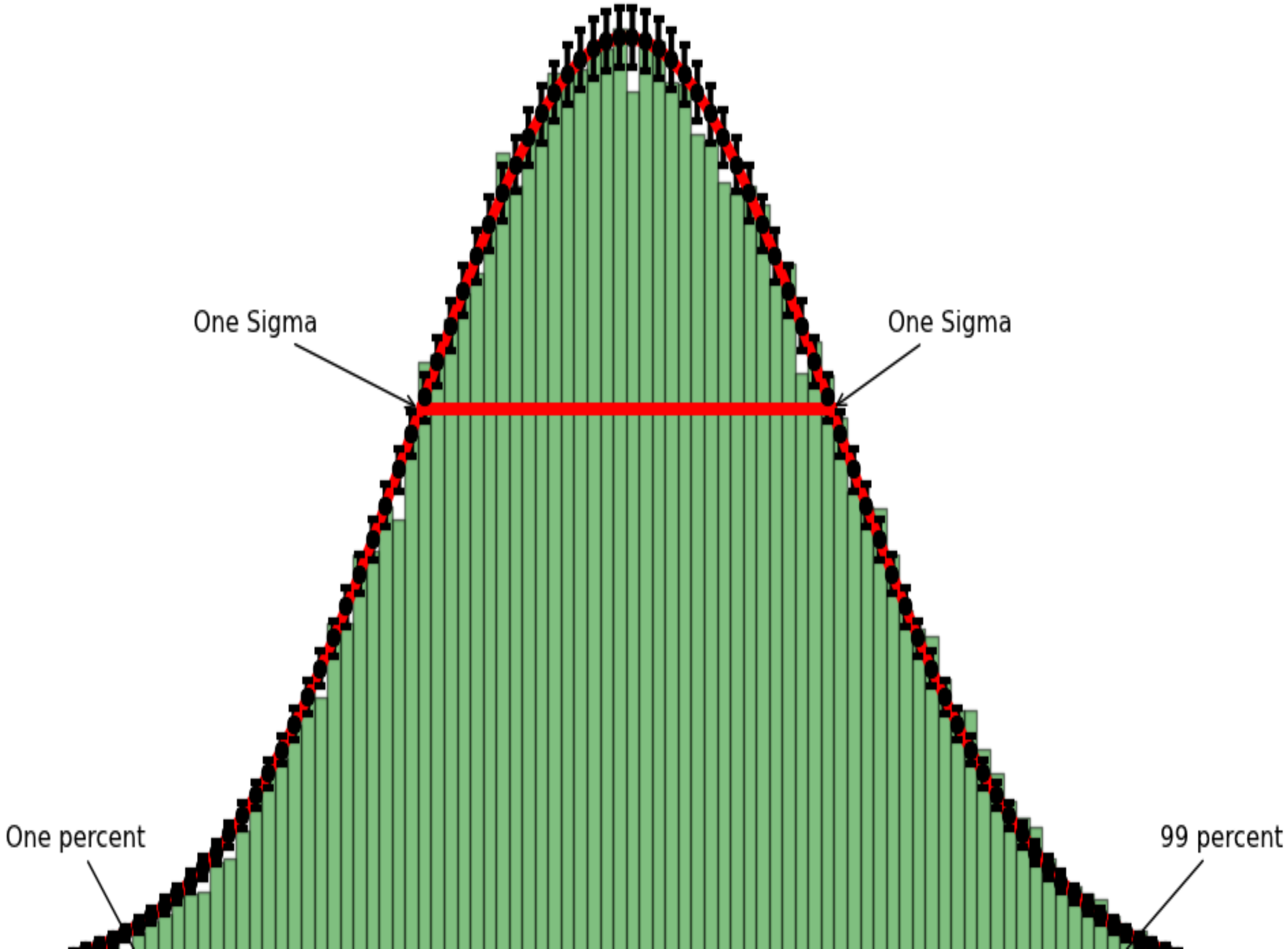
- from scipy.special import ndtri
- prob1percent25 = 25 + sqrt(25) * ndtri(0.01)
- prob99percent25 = 25 + sqrt(25) * ndtri(0.99)
- y1percent25 = 40000*Normal(prob1percent25, 25, sqrt(25))
- y99percent25 = 40000*Normal(prob99percent25, 25, sqrt(25))
- annotate('One percent', xycoords="data", textcoords='offset points', arrowprops=dict(facecolor='black', arrowstyle="->"), xytext =(-75,50), xy = (prob1percent25, y1percent25))
- annotate('99 percent', xycoords="data", textcoords='offset points', arrowprops=dict(facecolor='black', arrowstyle="->"), xytext =(30,50), xy = (prob99percent25, y99percent25))
- annotate('One Sigma', xycoords="data", textcoords='offset points', xy = (20,ybar25[0]), xytext = (-70,30), arrowprops=dict(facecolor='black', arrowstyle="->"))
- annotate('One Sigma', xycoords="data", textcoords='offset points', xy = (30,ybar25[1]), xytext = (30,30), arrowprops=dict(facecolor='black', arrowstyle="->"))

40000 Total Counts of 250

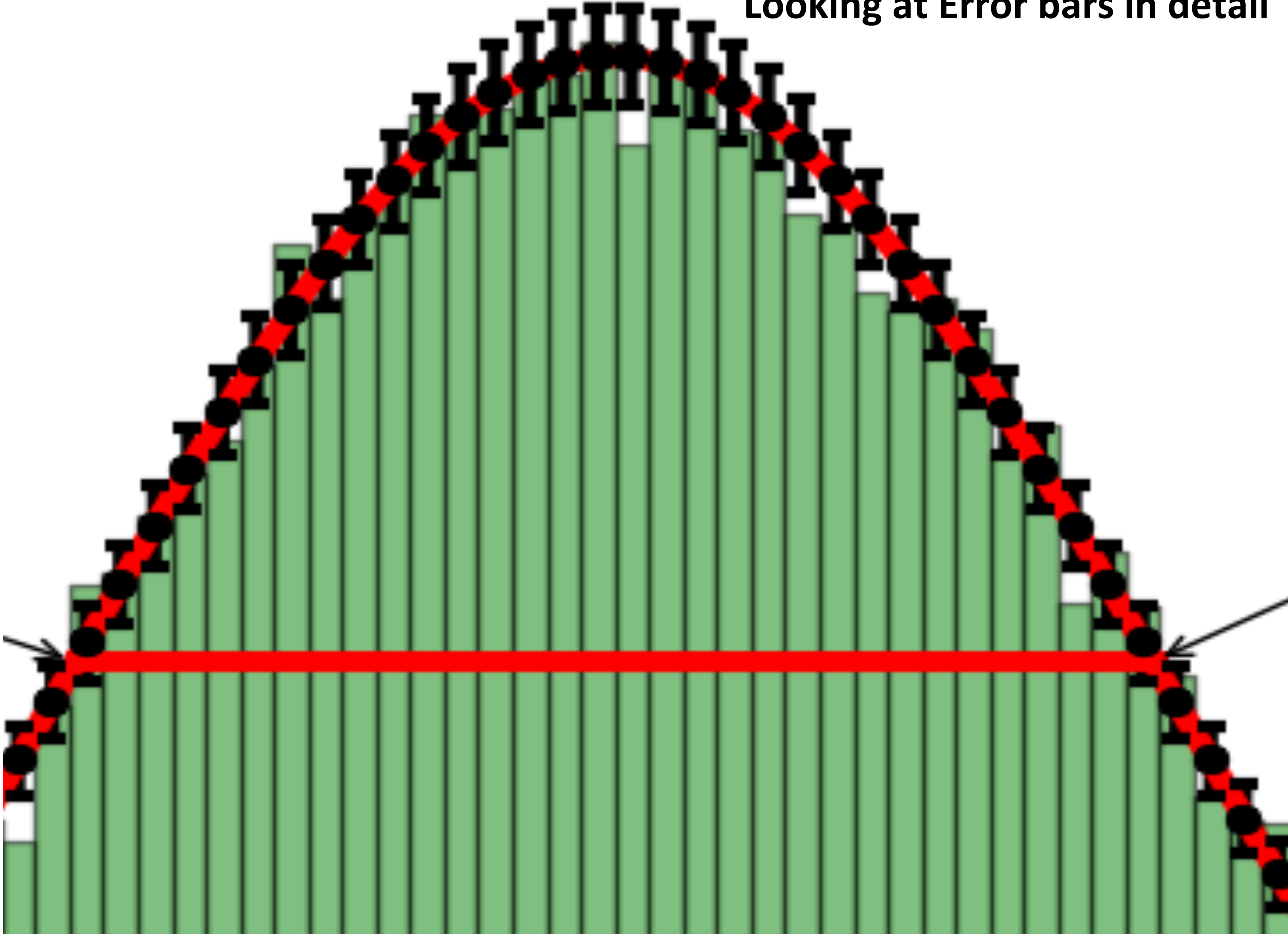


A larger set of Events

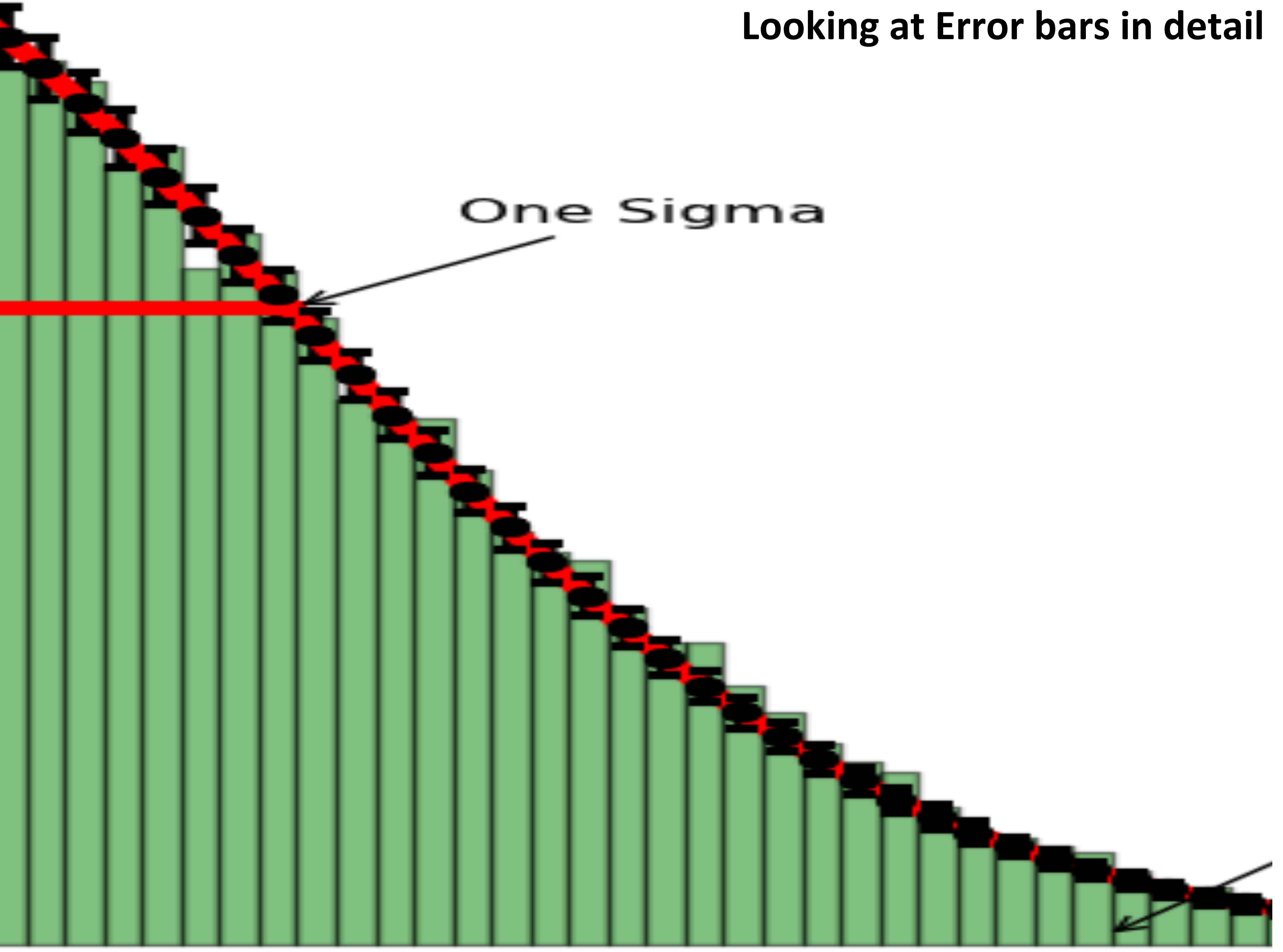
- Note if you had a total of ten million events but still 40,000 categories, then the red curve is nearer the green curve and 250 is expected number of events with an error of √250 ≈ 16
- The one sigma line runs from 250- √250 to 250 + √250
- The plots get rough if reduce number of events from 40000 to 400 or 4000
- Use larger bin size (1 becomes 5) if number of events small


One Sigma
One Sigma
One percent
99 percent

Looking at Error bars in detail



Looking at Error bars in detail



Making Error Bars
√ Law for y axis

- Numcounts25, binedges25, patches = hist(Counters25, bins = 50, range = (0,50), color = "green", alpha = 0.5)
- centers25 = 0.5*(binedges25[1:] + binedges25[:-1])
- y25 = 40000 * Normal(centers25, 25, sqrt(25))
- errors25 = sqrt(y25)
- errorbar(centers25, y25, yerr = errors25, linestyle='None', linewidth = 3.0, markeredgewidth = 3.0, marker ='o', color = 'black', markersize= 5.0)
- Note the √ law governs the width of distribution (x-axis) and counts in each bin (y axis)

Making One Sigma Line
√ Law for x axis

- xbar25 = np.zeros(2)
- ybar25 = np.zeros(2)
- xbar25[0] = 25 - sqrt(25)
- xbar25[1] = 25 + sqrt(25)
- ybar25 = 40000*Normal(xbar25, 25, sqrt(25))
- plot(xbar25, ybar25, color= "red", alpha = 1.0, lw =5)