Problem Set 5 problems

Question 1: Clebsch Gordon Tables (15 Points)

Learning objectives

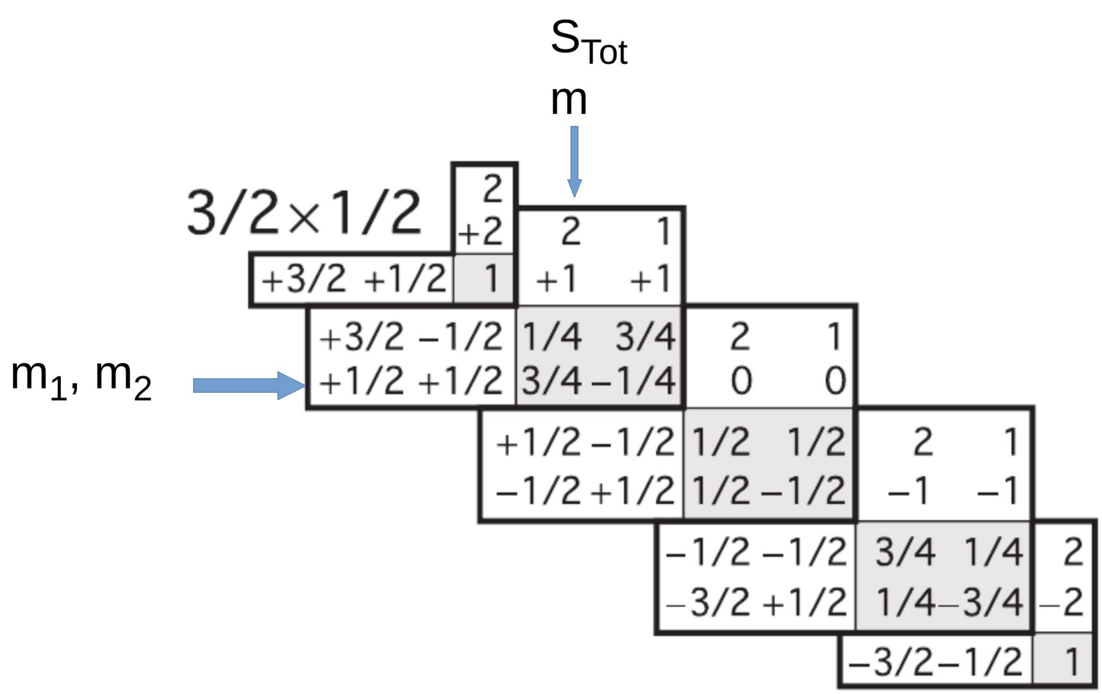
In this question you will:

- Review how to read Clebsch Gordon tables and use these tables to translate between the coupled and uncoupled bases in angular momentum or isospin

While it is informative to calculate Clebsch-Gordon coefficients once to see how it is done, it is tediuous to have to do this for every case. Tables of Clebsch Gordon coefficients have been calculated. A good compilation can be found at

[https://pdg.lbl.gov/2020/reviews/rpp2020-rev-clebsch-gordan-coefs.pdf (https://pdg.lbl.gov/2020/reviews/rpp2020-rev-clebsch-gordan-coefs.pdf)](https://pdg.lbl.gov/2020/reviews/rpp2020-rev-clebsch-gordan-coefs.pdf).

The compilation contains a number of blocks, one for each combination of angular momenta to be added together. For example, adding the spin for a spin-3/2 particle to the spin of a spin-1/2 particle (labeled $\frac{3}{2} \times \frac{1}{2}$ in the tables) gives the following block:



This block contains five sub-tables. The two left-most columns in each sub-table give the two values m_1 and m_2. The two top-most rows in each sub-table give the two values s and m . Note that the value of m is the same for all columns in a given subtable and also note that $m_1 + m_2$ always equals this value of m for each row. This is a reflection of the fact that the Clebsch-Gordan coefficients are all zero unless $m_1 + m_2 = m$. The other entries give numbers which are related to the values of the coefficients. You need to put a square-root over each entry but if there is minus sign, then that sign goes outside the square root. You can use this table to either decompose a state of given m_1 and m_2 into a sum of states of different s (by reading a row) or to decompose a state of given s amd m into a sum of states of different m_1 amd m_2 (by reading a column). Our notation is to write the coefficent as $C^{Jm_j}_{m1m_2}$. In other words, the subscript give the m_z components in the uncoupled basis and the numerator gives the total J and m_j in the coupled basis

1a.

Use the tables to find the coefficients:

- $C^{10}_{\frac{1}{2}-\frac{1}{2}}$ when we start with two spin $\frac{1}{2}$ particles
- C^{20}_{00} when we start with a particle with orbital angular momentum 2 and spin 1
- C^{32}_{11} when we start with a particle with orbital angular momentum 2 and spin 1

- $C^{10}_{\frac{1}{2}-\frac{1}{2}} = \sqrt{1} = 1$
- $C^{20}_{00} = \sqrt{0} = 0$
- $C^{32}_{11} = \sqrt{\frac{2}{3}}$

1b.

For the state $\left|\ell = \frac{3}{2}, s = 1, j = \frac{3}{2}, m_j = \frac{1}{2}\right\rangle$, what are the possible combinations of m_ℓ and m_s that can appear in the decomposition? Use the table of Clebsch-Gordan coefficients to expand this state in terms of the $|\ell s m_\ell m_s\rangle$ basis states and show that this expansion is properly normalized.

Using the table, we can construct:

$$\left|\ell = \frac{3}{2}, s = 1, j = \frac{3}{2}, m_j = \frac{1}{2}\right\rangle = \sqrt{\frac{2}{5}}\left|\ell = \frac{3}{2}, s = 1, m_\ell = \frac{3}{2}, m_s = -1\right\rangle + \sqrt{\frac{1}{15}}\left|\frac{3}{2}, 1\right.$$

Checking for normalization:

$$\left(\sqrt{\frac{2}{5}}\right)^2 + \left(\sqrt{\frac{1}{15}}\right)^2 + \left(-\sqrt{\frac{8}{15}}\right)^2 = \frac{2}{5} + \frac{1}{15} + \frac{8}{15} = \frac{15}{15} = 1$$

The expansion is normalized.

Question 2: Isospin: πN scattering (20 Points)

Learning objectives

In this question you will:

- Review how conservation of isospin in the strong interactions can be used to related scattering rates for different processes related by isospin
- Apply this concept to pion nucleon scattering

Consider the scattering of a pion with a nucleon. Since the pion has $I = 1$ and the nucleon has $I = \frac{1}{2}$, there are two possible values of total isospin for this scattering: $I = \frac{3}{2}$ and $I = \frac{1}{2}$. Isospin conservation in the strong interactions means that the matrix element $H_{if} = \langle f| H_{int} |i\rangle$ (where i and f are the initial and final states) depends only on I and not on I_z.

There are six possible *elastic scattering* processes:

$$\pi^+ p \to \pi^+ p$$
$$\pi^0 p \to \pi^0 p$$
$$\pi^- p \to \pi^- p$$
$$\pi^+ n \to \pi^+ n$$
$$\pi^0 n \to \pi^0 n$$
$$\pi^- n \to \pi^- n$$

And four possible *charge exchange* processes:

$$\pi^+ n \to \pi^- p$$
$$\pi^0 p \to \pi^+ n$$
$$\pi^- p \to \pi^0 n$$
$$\pi^0 n \to \pi^- p$$

Since all 10 of these processes have the same phase space terms, the relative rates for the processes can be calculated in terms of only two independent terms: $M_{\frac{1}{2}} = \left\langle I = \frac{1}{2} \middle| H_{int} \middle| I = \frac{1}{2} \right\rangle$ and $M_{\frac{3}{2}} = \left\langle I = \frac{3}{2} \middle| H_{int} \middle| I = \frac{3}{2} \right\rangle$

2a.

For the state $\pi^+ p$ we can describe the isospin of the state in the uncoupled and coupled bases:

$$\left| m_1 = +1, m_2 = \frac{1}{2} \right\rangle = \left| I = \frac{3}{2}, m = \frac{3}{2} \right\rangle$$

For the other five possible πN charge combinations (corresponding to the left hand side of the elastic scattering interactions listed above), write the description of the isospin state using the

Expressing each indepdenent basis in coupled basis (I's):

$$\pi^+ n : |m=1, m=-1/2\rangle = \sqrt{\frac{1}{3}}\,|I=3/2, m=1/2\rangle + \sqrt{\frac{2}{3}}\,|I=1/2, m=1/2\rangle$$
$$\pi^0 p : |0, 1/2\rangle = \sqrt{\frac{2}{3}}\,|3/2, 1/2\rangle - \sqrt{\frac{1}{3}}\,|1/2, 1/2\rangle$$
$$\pi^0 n : |0, -1/2\rangle = \sqrt{\frac{2}{3}}\,|3/2, -1/2\rangle + \sqrt{\frac{1}{3}}\,|1/2, -1/2\rangle$$
$$\pi^- p : |-1, 1/2\rangle = \sqrt{\frac{1}{3}}\,|3/2, -1/2\rangle - \sqrt{\frac{2}{3}}\,|1/2, -1/2\rangle$$
$$\pi^- n : |-1, -1/2\rangle = |3/2, -3/2\rangle$$

2b.

Using the results from 2a. and the fact that the scatttering cross sections are proportional to $|H_{if}|^2$ calculate the ratios of the cross sections for the following processes:

$$(a)\ \frac{\sigma(\pi^- p \to \pi^- p)}{\sigma(\pi^+ p \to \pi^+ p)}$$
$$(b)\ \frac{\sigma(\pi^- p \to \pi^0 n)}{\sigma(\pi^- p \to \pi^- p)}$$
$$(c)\ \frac{\sigma(\pi^0 n \to \pi^- p)}{\sigma(\pi^0 n \to \pi^0 n)}$$

Express your answers in terms of $M_{\frac{1}{2}}$ and $M_{\frac{3}{2}}$

Based on results from 2a, the above definitions of $M_{\frac{1}{2}}$, $M_{\frac{3}{2}}$, and that $\langle I=1/2|H|I=3/2\rangle = 0$ and $\langle 3/2|H|1/2\rangle = 0$,

i)

$$|\left\langle \pi^- p|H_{int}|\pi^- p\right\rangle|^2 = \left|\left(\sqrt{\frac{1}{3}}\right)^2 M_{\frac{3}{2}} + \left(-\sqrt{\frac{2}{3}}\right)^2 M_{\frac{1}{2}}\right|^2 = |\frac{1}{3}M_{\frac{3}{2}} + \frac{2}{3}M_{\frac{1}{2}}|^2$$

$$|\left\langle \pi^+ p|H_{int}|\pi^+ p\right\rangle|^2 = |M_{\frac{3}{2}}|^2$$

$$\therefore \frac{\sigma(\pi^- p \to \pi^- p)}{\sigma(\pi^+ p \to \pi^+ p)} = \left|\frac{\frac{1}{3}M_{\frac{3}{2}} + \frac{2}{3}M_{\frac{1}{2}}}{M_{\frac{3}{2}}}\right|^2 = \left|\frac{1}{3} + \frac{2}{3}\frac{M_{\frac{1}{2}}}{M_{\frac{3}{2}}}\right|^2$$

ii) Convention is that final state is the bra:

$$|\left\langle \pi^0 n|H_{int}|\pi^- p\right\rangle|^2 = \left|\frac{\sqrt{2}}{3}M_{\frac{3}{2}} - \frac{\sqrt{2}}{3}M_{\frac{1}{2}}\right|^2$$

$$|\left\langle \pi^- p | H_{int} | \pi^- p \right\rangle|^2 = |\frac{1}{3} M_{\frac{3}{2}} + \frac{2}{3} M_{\frac{1}{2}}|^2$$

$$\therefore \frac{\sigma(\pi^- p \to \pi^0 n)}{\sigma(\pi^- p \to \pi^- p)} = \left| \frac{\frac{\sqrt{2}}{3} M_{\frac{3}{2}} - \frac{\sqrt{2}}{3} M_{\frac{1}{2}}}{\frac{1}{3} M_{\frac{3}{2}} + \frac{2}{3} M_{\frac{1}{2}}} \right|^2$$

iii)

$$|\left\langle \pi^- p | H_{int} | \pi^0 n \right\rangle|^2 = |\left\langle \pi^0 n | H_{int} | \pi^- p \right\rangle|^2 = \left| \frac{\sqrt{2}}{3} M_{\frac{3}{2}} - \frac{\sqrt{2}}{3} M_{\frac{1}{2}} \right|^2$$

$$|\left\langle \pi^0 n | H_{int} | \pi^0 n \right\rangle|^2 = \left| \frac{2}{3} M_{\frac{3}{2}} + \frac{1}{3} M_{\frac{1}{2}} \right|^2$$

$$\therefore \frac{\sigma(\pi^0 n \to \pi^- p)}{\sigma(\pi^0 n \to \pi^0 n)} = \left| \frac{\frac{\sqrt{2}}{3} M_{\frac{3}{2}} - \frac{\sqrt{2}}{3} M_{\frac{1}{2}}}{\frac{2}{3} M_{\frac{3}{2}} + \frac{1}{3} M_{\frac{1}{2}}} \right|^2$$

Question 3: Isospin: The Δ resonance (Thomson 9.3) (15 points)

Learning objectives

In this question you will:

- Review the fact that isospin conservation in the strong interaction can also be used to relate decay rates
- Apply this concept to the case od Δ decays

By considering the isospin states show that the rates for the following strong interaction decays occur in the ratios:

$$\begin{aligned} &\Gamma(\Delta^- \to \pi^- n) : \quad \Gamma(\Delta^0 \to \pi^- p) : \quad \Gamma(\Delta^0 \to \pi^0 n) : \quad \Gamma(\Delta^+ \to \pi^+ n) : \\ &\Gamma(\Delta^+ \to \pi^0 p) : \quad \Gamma(\Delta^{++} \to \pi^+ p) \quad = \quad 3:1:2:1:2:3 \end{aligned}$$

Note: Because the Δ can be created in $\pi^+ p$ scattering, we know that it has isospin $I = \frac{3}{2}$

Because we know Δ has $I = 3/2$, we can just compute the norm square inner products of the initial and final states directly. Since we're comparing their ratios, the constant of proportionality is irrelevant. Thus:

$$\Gamma(\Delta^- \to \pi^- n) \propto |\langle I = 3/2, m = -3/2 | I = 3/2, m = -3/2 \rangle|^2 = 1$$

$$\Gamma(\Delta^0 \to \pi^- p) \propto \left|\left\langle I = 3/2, m = -1/2 | \sqrt{\frac{1}{3}} | I = 3/2, m = -1/2 \right\rangle\right|^2 = 1/3$$

$$\Gamma(\Delta^0 \to \pi^0 n) \propto \left|\left\langle I = 3/2, m = -1/2 | \sqrt{\frac{2}{3}} | I = 3/2, m = -1/2 \right\rangle\right|^2 = 2/3$$

$$\Gamma(\Delta^+ \to \pi^+ n) \propto \left|\left\langle I = 3/2, m = 1/2 | \sqrt{\frac{1}{3}} | I = 3/2, m = 1/2 \right\rangle\right|^2 = 1/3$$

$$\Gamma(\Delta^+ \to \pi^0 p) \propto \left|\left\langle I = 3/2, m = 1/2 | \sqrt{\frac{2}{3}} | I = 3/2, m = 1/2 \right\rangle\right|^2 = 2/3$$

$$\Gamma(\Delta^{++} \to \pi^+ p) \propto |\langle I = 3/2, m = 3/2 | I = 3/2, m = 3/2 \rangle|^2 = 1$$

Thus, comparing their ratios in that exact order, we get

$$3 : 1 : 2 : 1 : 2 : 3$$

Question 4 SU(3) λ matrices (20 points)

Learning objectives

In this question you will:

- Review the definition of the λ matrices (the generators of SU(3) symmetry)
- Show that the SU(3) group contains three SU(2) subgroups and construct the raising and lowering operators for these subgroups

We learned in Quantum Mechanics that the smallest representation of angular momentum has dimension 2 (spin-$\frac{1}{2}$) and that the generators of infinitesmal rotations in this fundamental representation can be written as the three 2×2 Pauli matrices σ_x, σ_y and σ_z that satisfy the SU(2) commutation relations

$$\left[\frac{1}{2}\sigma_i, \frac{1}{2}\sigma_j\right] = i\epsilon_{ijk}\frac{1}{2}\sigma_k$$

Similarly, the fundamental representation of SU3) can be written as a set of eight 3×3 matrices called the λ-martices:

$$\lambda_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \lambda_2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \end{pmatrix} \lambda_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

4a.

Examining the first three λ shows that they are just the Pauli matrices with zeros added in the third row and column. Thus, these three matrices form a subgroup of SU(2) within the SU(3) group and obey the commutation relations we know from angular momentum. This is one of three SU(2) subgroups within the SU(2) group. We call them:

$$\begin{array}{llll} \text{Isospin} & \lambda_1 & \lambda_2 & \lambda_3 \\ \text{U}-\text{spin}: & \lambda_4 & \lambda_5 & \frac{1}{2}\left(\sqrt{3}\lambda_8 + \lambda_3\right) \\ \text{V}-\text{spin}: & \lambda_6 & \lambda_7 & \frac{1}{2}\left(\sqrt{3}\lambda_8 - \lambda_3\right) \end{array}$$

Using explicity matrix multiplication. Show that the generator of U-spin and V-spin satisfy the commutation relations that define the SU(2) algebra.

Done on paper.

4b.

The fundamental representation of SU(3) can be written as the three column matrices which in we call in the quark model:

$$u \equiv \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad d \equiv \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad s \equiv \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Show that the raising (lowering) operators

$$I_\pm = \tfrac{1}{2}(\lambda_1 \pm i\lambda_2) \quad U_\pm = \tfrac{1}{2}(\lambda_4 \pm i\lambda_5) \quad V_\pm = \tfrac{1}{2}(\lambda_6 \pm i\lambda_7)$$

turn u into d (and visa versa), u into s (and visa versa) and d into s (and visa veras)

Done on paper.

Question 5: Fitting a Signal in the Presence of Background (30 points)

Learning objectives

In this question you will:

- Gain experience in performing χ^2 fits to histogrammed data; explore how the statistical significance of a signal depends on the number of signal events and the signal-to-background ratio

5a.

Physicists often fit for signals in the presence of background. In such cases, the significance of the fitted signal depends not only on the number of signal events but also on the amount of background and our ability to statistically separate the two. In this problem, we will explore fitting signal and background for a very simple case: The signal is a Gaussian peak centered at $x = 10$ with a width $\sigma = 1$ and the background is uniformly distributed between $x = 0$ and $x = 20$. The code below generates fake data, allowing you to change both the number of events in the signal and the ratio of signal-to-background. To make sure that our definition of background does not depend on the fit range, the code below defines the signal-to-background ratio as the ratio number of signal events to the number of background events in a $\pm 2\sigma$ window around the signal peak. Here is the code you will use to generate the fake data:

In [131]:
```
# Write import math
import numpy as np
import random
import matplotlib.pyplot as plt
from scipy.optimize import minimize

def makeData( SignalToBackground, nSigEvents, seed=12345):
    """Generates a dataset consisting of nEvents some of which are signal
    at x=5) and the rest of which is background
    Definition of SignalToBackground:  The ratio of the number of signal
    of background events in that same x-range  (this definition is just a

    Parameters
    ==========
     SignalToBackground : float
     Ratio of the number of signal events within 2 sigma of the peak to t
    of background events in that same x-range  (this definition is just a

    nSigEvents : int
      total number of signal events generated (NOT the number in +- 2 sig

    seed : int
      seed for the random number generator

    Returns
    =======
    data : array
      the measurements of x
    """
    fracOutsideTwoSigma = 4.55e-2
    # 1-fracOutsideTwoSigma is the fraction of the Signal Events within +
    # To get the total number of background events, find the number in +-
    # which is 4 units of x and then since the background is flat from 0
    # multiply by 20/4=5

    nBackground = 5*nSigEvents*(1-fracOutsideTwoSigma)/SignalToBackground
    nEvents = nSigEvents+nBackground
    fSig = nSigEvents/nEvents
```

```
    # Make an array to hold the data (ie the x measurements)
    data = []

    # set the random seed.  This will allow us to reproduce the results i
    np.random.seed(seed)

    # Retrieve nEvents random numbers that will be used to pick which eve
    # and which are background
    n = int(nEvents)
    tests = np.random.uniform(0,1,n)
    bck = np.random.uniform(0,20,n)
    sig = np.random.normal(10,1,n)

    count = 0
    for test in tests:
        if(test<fSig):
            data.append(sig[count])
        else:
            data.append(bck[count])
        count+=1

    # Loop over the events and pick either signal or background and draw
    # pdf in each case

    # return the data to the user
    return data, nSigEvents/np.sqrt(nEvents)
```

Below is a simple test to show you how to use this code. Remember if you run this code multiple times you should change the seed each time (for example, you could increment the seed each time you call the function)

In [268]:
```
mydata = []
#  make 10 signal events with a signalToBackground of 1 using random seed
mydata, fSig =makeData(1,20,123)
print("number of total data events: ",len(mydata))
```

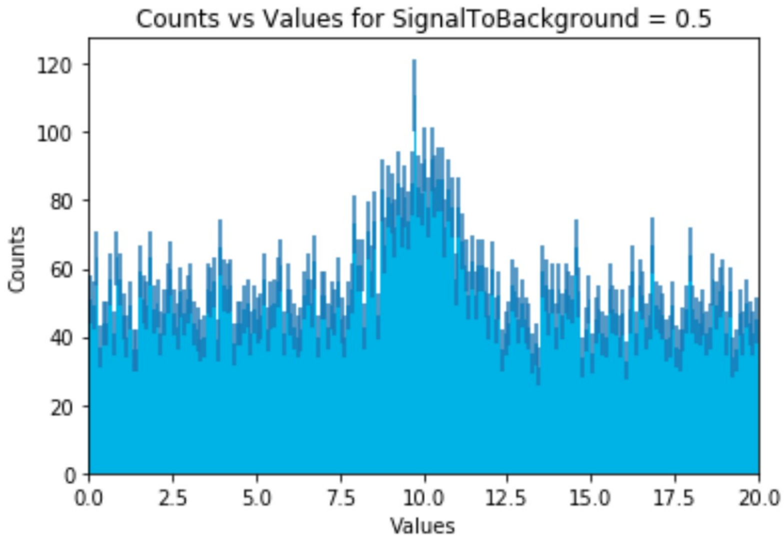
```
number of total data events:  115
data:  [11.047401505881462, 11.57102936217666, 10.430661187946646, 0.0
5376129148641384, 19.7669083856564, 18.106831513232198, 4.152717223926
49, 5.849788255848496, 10.40020306144967, 18.038227453213413, 19.67261
7698234465, 5.150841283081662, 11.287180858495633, 10.579689779144333,
7.887401079055496, 14.621460716891141, 3.221380288584297, 12.013971356
671798, 17.317289166065294, 19.67043218407111, 1.5873158075603144, 8.5
66945494018984, 4.0908571909285545, 9.012729810374696, 10.955271452577
08, 1.8665342073964153, 5.937215509613589, 18.55168480304295, 11.38007
4628603907, 9.148239950472238, 8.762646787312146, 14.837243036840746,
0.9715806568853758, 14.17394790885492, 16.784866956101673, 3.318757684
1390776, 15.619958759999147, 5.730732334582038, 6.129395066591146, 13.
30522930699366, 2.227843432154315, 9.800887617242747, 17.7571358535244
53, 13.926225364708127, 8.806557533308181, 8.764287687544494, 15.30192
```

Generate a sample of 1000 signal events with signalToBackground=0.5 and make a histogram of your results. Make sure that the number of bins in your histogram is small enough that you have at least 10 entries per bin (so that it is reasonable to do a binned fit to the histogram). To make life a bit easier, here is function you can use to make your histograms. You are of course free to write your own function and not use this code.

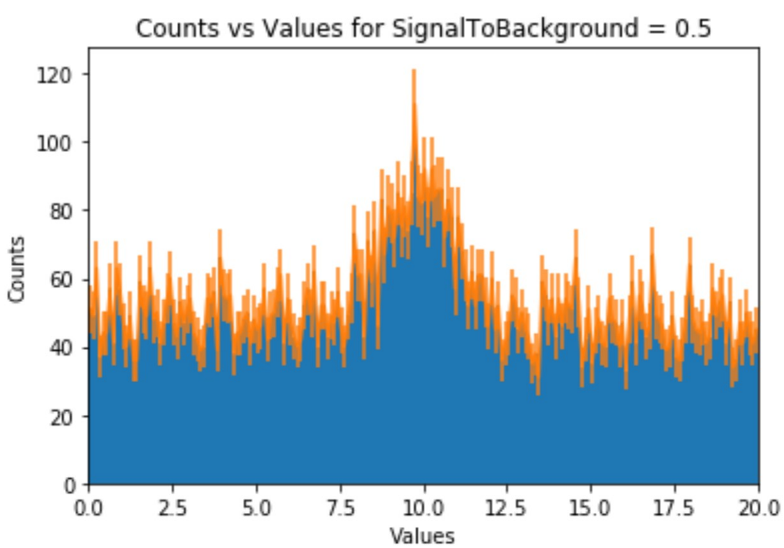
In [386]:
```python
#Import the pyplot module of matplotlib as "plt"
import matplotlib.pyplot as plt


#Makes a histogram filled with the random numbers we generate
def plot_histogram(samples,xtitle,ytitle, title, nbins, limits):

    #Plot the histogram of the sampled data with nbins and a nice color
    n, bins, patches =plt.hist(samples, bins=nbins, range=limits, color=(
                                                                    #  use
    bincenters = 0.5*(bins[1:]+bins[:-1])
    errs = np.sqrt(n)

    plt.errorbar(bincenters, n, yerr=errs, fmt='none')
    #Add some axis labels and a descriptive title
    plt.xlabel(xtitle)
    plt.ylabel(ytitle)
    plt.title(title)

    #Get rid of the extra white space on the left/right edges (you can de
    xmin, xmax, ymin, ymax = plt.axis()
    plt.axis([limits[0],limits[1],ymin,ymax])

    #Not necessarily needed in a Jupyter notebook, but it doesn't hurt
    plt.show()
```

In [387]:
```python
# Using given function

sig2back = 0.5
nSigEvents = 1000
nbins = 200
data, Sig = makeData(sig2back, nSigEvents, seed = 124)
lims = [np.min(data), np.max(data)]
```

```
title = "Counts vs Values for SignalToBackground = " + str(sig2back)
n, bins, patches = plot_histogram(data,'Values','Counts',title = title, n
```



In [388]:
```
# My own method: gets the "bincenters" value, which is more helpful for f
# not as streamlined labelling system though
data, fSig = makeData(sig2back, nSigEvents, seed = 124)
lims = [np.min(data), np.max(data)]

counts, binEdges = np.histogram(data, nbins)
bincenters = 0.5*(binEdges[1:] + binEdges[:-1])
err = np.sqrt(counts)

plt.figure()
plt.title("Counts vs Values for SignalToBackground = " + str(sig2back))
plt.xlabel("Values")
plt.ylabel("Counts")
plt.xlim(lims)
plt.hist(data, bins)
plt.errorbar(bincenters, counts, yerr = err)
```



In [ ]:

5b.

Pretend this is real data. Your goal is to find the best estimate of how many events are in a Gaussian peak with unknown mean and sigma and what the uncertainty on this estimate is. In your fit, you can make the assumption that the background is flat (a zeroth order polynomial) but that you don't have a prediction for the background rate. Use your favorite minimization package to fit the data. Deterimine the best estimate of the number of events in signal and the uncertainty on that estimate. (remember, that you must let the position and width of the Gaussian and the size of the background vary in your fit).

Hint: For examples of how to perform a least squared fit of a function to data see:

- [https://github.com/berkeley-physics/Python-Tutorials/blob/master/3%20-%20Specific%20topics/Fitting.ipynb (https://github.com/berkeley-physics/Python-Tutorials/blob/master/3%20-%20Specific%20topics/Fitting.ipynb)](https://github.com/berkeley-physics/Python-Tutorials/blob/master/3%20-%20Specific%20topics/Fitting.ipynb)
- [https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html (https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html)](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html)

In [389]:

In [390]:

```
# With the assumption that the background is flat, we can have the model

def gaussian(x, norm, mu, sigma, Background):
    other_factor = (1/(sigma*np.sqrt(2*np.pi)))
```

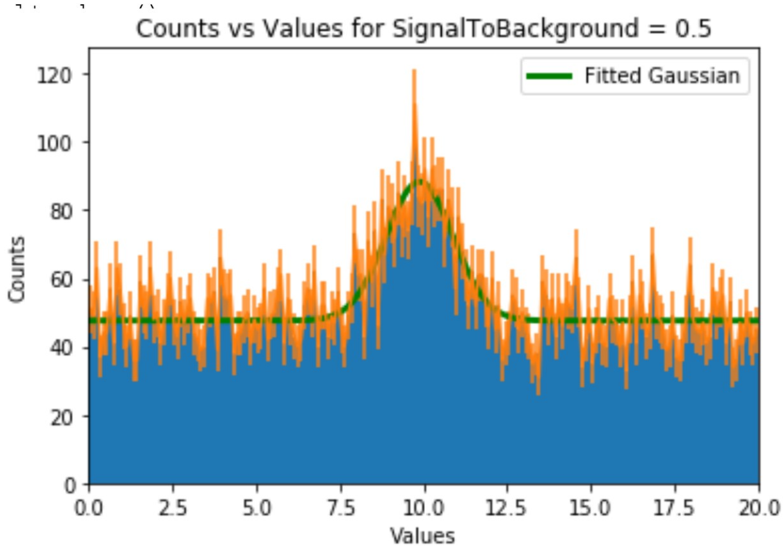
In [391]:

Out[391]: 200

In [392]:

```
# binned fit
fittedparam, m = curve_fit(gaussian, bincenters, counts, p0 = [100, 10, 1
paramserror = np.sqrt(np.diag(m))

norm = np.abs(fittedparam[0])
mu = fittedparam[1]
sigma = np.abs(fittedparam[2])
background = fittedparam[3]
norm_err = paramserror[0]
mu_err = paramserror[1]
sigma_err = paramserror[2]
background_err = paramserror[3]
# tmu = fittedparam[0]
# tsig = fittedparam[1]
# tback = fittedparam[2]



x = np.linspace(np.min(data), np.max(data), 1000)

plt.figure()
```

```
plt.title("Counts vs Values for SignalToBackground = " + str(sig2back))
plt.xlabel("Values")
plt.ylabel("Counts")
plt.xlim(lims)
plt.hist(data, bins)
plt.errorbar(bincenters, counts, yerr = err)
plt.plot(x, gaussian(x, norm, mu, sigma, background), linewidth = 3, colo
plt.legend()
```



In [393]:
```
print("Parameters and their Errors")
print('Norm factor:', norm, "+/-", norm_err)
print('Mean:', mu, "+/-", mu_err)
print("Sigma:", sigma, "+/-", sigma_err)
```

```
Parameters and their Errors
Norm factor: 102.30553017956363 +/- 6.697221768419848 5
Mean: 9.884511444630999 +/- 0.06487598421862505
Sigma: 1.0020958198911847 +/- 0.068690613531736
Background:  47.609477851877344 +/- 0.6487957477155936
```

In [401]:
```
# Best estimate for number of eventsin signal
# total events - background*bins should readily filter out the noise
Estimated_signal_events = np.sum(counts) - background*len(bincenters)
#error propagation
error = np.sqrt(mu_err**2 + sigma_err**2 + background_err**2 + norm_err**
```

```
Estimated events in signal: 1023.1044296245309 +/- 6.729237893800919
```

5c.

What is the χ^2 per degree of freedom for your fit? What does this number tell you about how well your fit describes the data?

Reduced χ^2 is defined as

$$\chi^2_{red} = \frac{1}{d} \sum_i \frac{(Obs_i - Exp_i)^2}{\sigma_i^2}$$

In [376]:
```
fittedvals = gaussian(bincenters, norm, mu, sigma, background)
err = np.sqrt(counts)
chi2 = 0
for i in range(len(fittedvals)):
    chi2 += ((counts[i] - fittedvals[i])**2)/(err[i]**2)

df = len(fittedvals) - len(fittedparam)
reduced_chi2 = chi2/df
```

Out[376]: 1.2161647847037593

A χ^2_{red} of ≈ 1 indicates a good fit of the model to the data. Thus, because the above χ^2_{red} is close to 1, it means the model is a good fit.

5d.

If a signal of size N_S^{meas} has a fitted uncertainty σ_S, the signficance of the measurement is defined to be

$$S^{meas} \equiv \frac{N_S^{meas}}{\sigma_S}$$

When the size of the data sample is large enough that Gaussian uncertainties are appropriate, a rule of thumb can be used to give a crude estimate of the expected significance $S^{expected}$ of the measurement. This predicted signficance depends on the number of signal events (N_S) and is the number of background events populating the region **beneath the signal** (N_B):

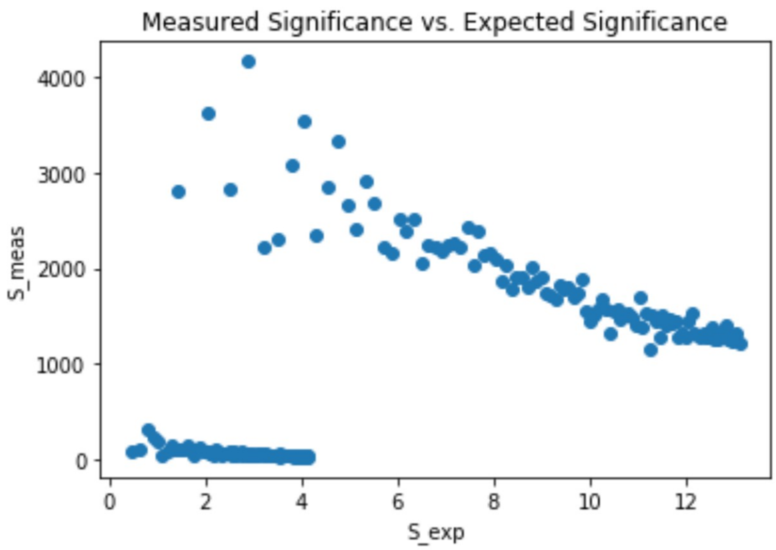
$$S^{expected} \approx \frac{N_S}{N_S + N_B}$$

Repeat the above exercise changing both the number of events in your signal and the signalToBackground ratio. Plot the values of the measured significance S^{meas} obtained from your fits as a function of $\frac{N_S}{\sqrt{N_S+N_B}}$. How do your results compare to the simple rule of thumb quoted above?

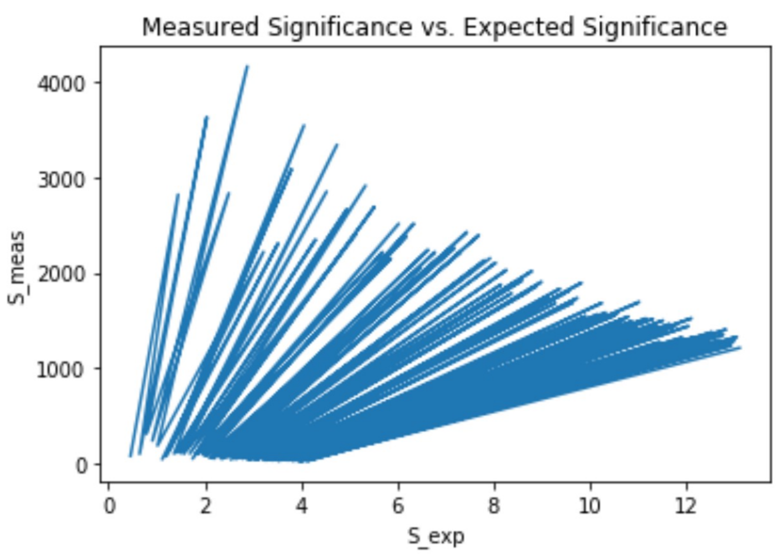
In [408]:
```
seed = 12345
S_exps = []
S_meass = []
for i in range(1,100):
    sig2back = i/100
    seed += 1
    for j in range(2,4):
        nSigEvents = 10**j
        temp_data, S_exp = makeData(sig2back, nSigEvents, seed)
        S_exps.append(S_exp)
        nbins = int(len(temp_data)/5)
        counts, binEdges = np.histogram(temp_data, nbins)
        bincenters = 0.5*(binEdges[1:] + binEdges[:-1])
        fittedparam, m = curve_fit(gaussian, bincenters, counts, p0 = [nS
```

```
        perror = np.sqrt(np.diag(m))
        total_err = np.sqrt(perror[0]**2 + perror[1]**2 + perror[2]**2 +
        mu = fittedparam[1]
        sigma = np.abs(fittedparam[2])
        background = fittedparam[3]
        N_meas = np.sum(counts) - background*len(bincenters)
        S_meas = N_meas/total_err
        S_meass.append(S_meas)
```

In [410]:
```
plt.figure()
plt.xlabel('S_exp')
plt.ylabel("S_meas")
plt.title("Measured Significance vs. Expected Significance")
plt.scatter(S_exps, S_meass)
```



In [411]:
```
plt.figure()
plt.xlabel('S_exp')
plt.ylabel("S_meas")
plt.title("Measured Significance vs. Expected Significance")
plt.plot(S_exps, S_meass)
```



In [ ]: