Problem Set #4 problems

Question 1: Likelihood Fits, Statistical Methods (40 points)

Learning objectives

In this question you will:

- Construct a likelihood function for a probability distribution with one free parameter
- Determine the best fit value of the parameter and estimate its uncertainty both by graphing the likelihood function and using a standard minimization package

Consider the problem of determining the lifetime of a species of particle that we can stop in our detector by observing its decays. Assume each time a particle stops, we set the stopping time to be $t = 0$ and that we only observe decays that occur up to a time T_{max} after the particle stops. The distribution of measured times therefore follows the form:

$$\begin{aligned} R(t) &= R_0 e^{-\Gamma t} && 0 \leq t \leq T_{max} \\ &= 0 && \text{otherwise} \end{aligned}$$

For this problem, we'll take as the true decay parameter $\Gamma = 2\ \text{sec}^{-1}$ and maximum time that we wait for a decay to be $T_{max} = 3$ sec. We can imagine doing the experiment over many times (each experiment takes three seconds) to accumulate a lot of data.

1a.

First, let's generate some fake data. Generate 1000 decay times that follow the formula above. (Hint: use numpy.random.exponential and reject events with decay times larger than T_{max}. Verify that your event generation looks reasonable by making a histogram of the decay times.

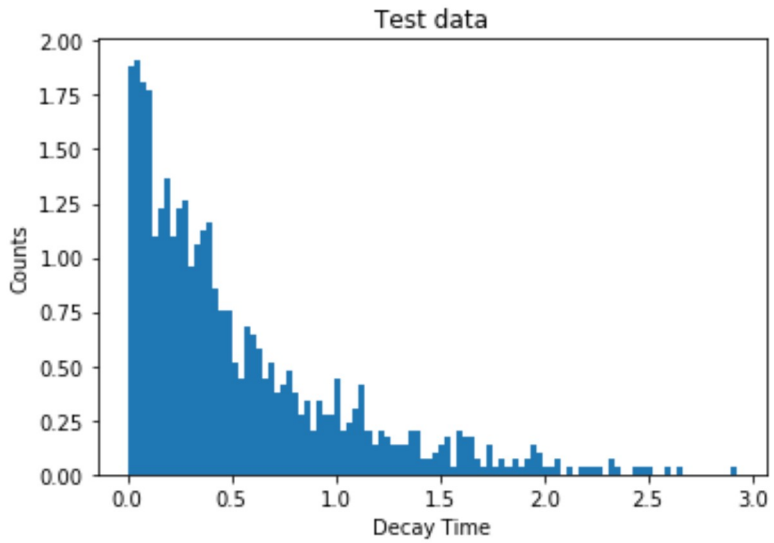
In [113]:
```
import math
import numpy as np
import random
import matplotlib.pyplot as plt
from scipy.optimize import minimize
```

In [114]:
```
def makeData( Gamma, Tmax, nDecays ):
    """Generates a dataset of decay times.  The distribution of events fo
    decay parameter Gamma, but where the decays are cut-off after time Tm

    Parameters
    ==========
```

```
    Gamma : float
      decay parameter of the exponential distribution

    Tmax : float
      maximum decay time that can be generated in the dataset

    nDecays : int
      number of decay times to generate

    Returns
    =======
    decayTimes : array
      nDecays number of decay times generated according to the exponentia
      parameter gamma and maximum decay time Tmax
    """

    # Make an array to hold the decay times
#     decayTimes = np.zeros(nDecays)
    first_array = np.random.exponential(scale = (1/Gamma), size = nDecays
    decayTimes = first_array[np.where(first_array <= Tmax)]
```

In [115]:
```
gamma = 2 # s^-1
Tmax = 3 #s
nDecays = 1000
```

In [116]:
```
n, bins, patches = plt.hist(test_data, 100, density = True)
plt.title("Test data")
plt.xlabel("Decay Time")
plt.ylabel("Counts")
```



1b.

Calculate the negative log-likelihood function, $-\ln \mathcal{L}$, for your data. Express your likelihood in terms of the free parameters, Γ.

Hint: your pdf for the expected distribution of decay times $f(t)$ is an exponental that only extends to t_{max} so:

$$\int_0^{t_{max}} f(t)dt = \int_0^{t_{max}} R_0 e^{-\Gamma t} dt = 1$$

Because of the above property, we can treat the dataset as a PDF, so we can take the negative log of it directly:

Also, doing the integral directly to solve for the normalization factor:

$$R_0 = \frac{-\Gamma}{\exp(-\Gamma t_{max}) - 1}$$

In [117]:
```
R_0 = -gamma/(np.exp(-gamma*Tmax) - 1)
```

Out[117]:
```
2.0049698233136892
```

In [118]:
```
def exp(norm, gamma, times):
```

In [119]:
```
function = exp(R_0, gamma, decaytimes)
negloglikelihood = -1*np.sum(np.log(function))
```

Out[119]:
```
285.4354611285834
```

1c.

We will study the simulated data, pretending that we dont know what value of Γ was used to generate it. We want to find the best estimate of Γ from the data.

We saw in class that for high statistics $-2\ln\mathcal{L}$ is distributed like a χ^2 distribution and the uncertainty on the estimate of a parameter of the function can be obtained by finding how much the you can change the parameter to increase $-2\ln\mathcal{L}$ by 1. Write code to calculate the negative log-likelihood:

$$-2\ln\mathcal{L} = -2\sum_i \ln f(\Gamma, t_i)$$

where the t_i are the time values you generated. Using this code, find the value of $-2\ln\mathcal{L}$ for $\Gamma - \Gamma_{true}$.

In [120]:
```
def minusloglikelihoodFn(Gamma, maxT, decayTimes):
    """calculates the -ln(Likelihood) for the decayTimes for specified va

    Parameters
    ==========
    Gamma : float
      hypothesis for lifetime

    Tmax : float
      maximum time for observation
```

```
    nDecays : array
      a dataset of decay times generated according to the distribution de

    Returns
    =======
    minusLogLikelihood : float
      -ln(Likelihood) given the hypothesized Gamma and maxT for the input
    """
    # decaytimes = result from 'makeData'
    #decayTimes = makeData(Gamma, maxT, nDecays)
    R_0 = -Gamma/(np.exp(-Gamma*maxT) - 1)
    function = exp(R_0, Gamma, decayTimes)

    minusLogLikelihood = -1*np.sum(np.log(function))
    # print(mean_time)

    # Factor of 2 Apply it outside?
```

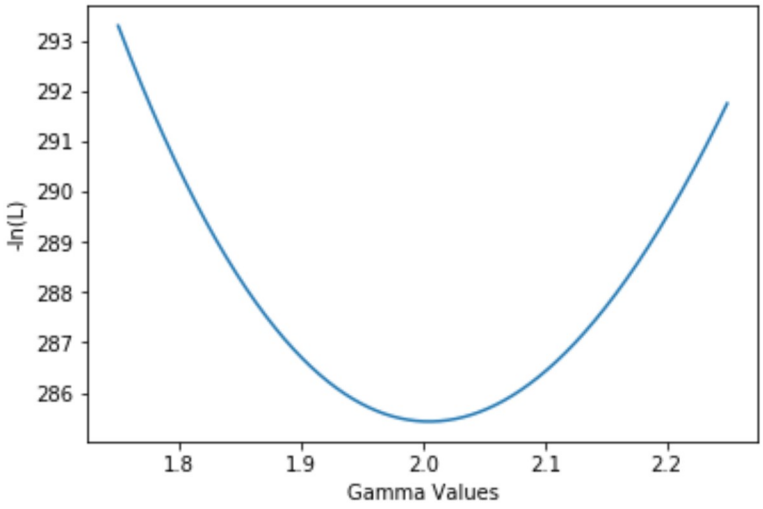
In [121]:
```
tLikelihood = 2*minusloglikelihoodFn(Gamma = gamma, maxT = Tmax,decayTime
```

Out[121]: 570.8709222571669

1d.

There are lots of algorithms for finding the minimum of a non-linear function such as our negative log-likelihood, but we won't bother to use any of these algorithms for yet. Instead, we will explore the minimum by inspecting the behavour of the function. Plot the value of $-\ln \mathcal{L}$ you obtain from your simulated data as you vary Γ in the region of the true answer ($\Gamma = 2$). How close is the Γ that gives minimum negative log-likelihood to the true value of Γ? Estimate the uncertainty on your estimate of Γ by finding the values corresponding to an increase of $\mathcal{L}$ of 1.0

In [122]:
```
ngrid=1000 # How many points to scan
G = np.arange(1.75,2.25,0.5/ngrid)  #The gamma values to scan
LLG = np.zeros(ngrid)               #The likelihood for these values

for i in range(len(LLG)):
    LLG[i] = minusloglikelihoodFn(Gamma = G[i], maxT = Tmax, decayTimes =


plt.figure()
plt.xlabel('Gamma Values')
plt.ylabel('-ln(L)')
plt.plot(G,LLG)
```



1e.

While this graphical method of finding the minimum works well for simple cases, we often will use a minimization package instead. Using the data you have already created, pick your favorite minimization package and see if it finds the right minimum and whether the uncertainty it returns agrees with your estimate above.

Hint: The uncertainty on the fitted minumum can be estimated by taking the square root of the inverse Hessian of the likelihood function $\mathcal{L}(x|\alpha$ where x are the observed values of the measurements and $\alpha = (\alpha_0, \alpha_1, \dots)$ are the parameters to be fit. The Hessian is defined as

$$H(\alpha) = \frac{\partial^2}{\partial\alpha_i \partial\alpha_j}$$

One example of a minimization package that returns the inverse Hessian is *scipy.optimize.minimize* (https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html) using the 'BFGS' method.

In [123]:
```
from scipy.optimize import minimize
```

In [124]:
```
```

In [125]:

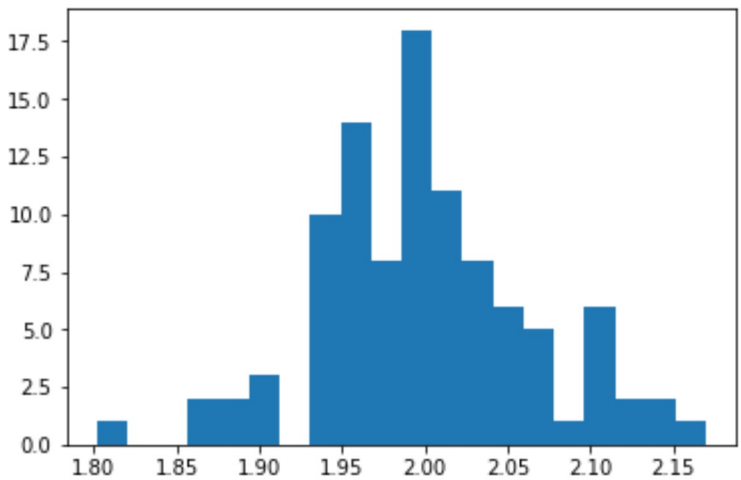
In [126]:
```
# Final result:
```

```
Minimized Gamma value =  2.005070263062478 +/- 0.004971522530540824
```

1f.

To verify your estimate of the uncertainty on the measured value of Γ, generate 100 samples each with 1000 events. Histogram the estimated Γ for these samples and find the rms of the "measurements." How does this rms compare to your answers above?

In [127]:
```
vals = []
errs = []
for i in range(100):
    decaytimes = makeData(gamma, Tmax, 1000)
    MLL = minusloglikelihoodFn(gamma, Tmax, decaytimes)
    min_gamma = scipy.optimize.minimize(minusloglikelihoodFn, 2, args = (
    val = min_gamma.x[0]
    err = min_gamma.hess_inv[0][0]
    vals.append(val)
    errs.append(err)

plt.figure()
n, bins, patches = plt.hist(vals, 20)
plt.show()

# interpreting RMS to be equivalent as STDEV, then
stdev = np.std(vals)
```



Out[127]:
```
0.0649312305322798
```

Standard deviation is larger than the resulting error from scipy.optimize.minimize, but it should generally decrease as the number of samples increase.

Question 2: Noether's Theorem (20 points)

Learning objectives

In this question you will:

- Review the meaning of Noether's theorem
- Apply Noether's theorm to a non-relativistic quantum mechanical example

In non-relativistic quantum mechanics, we learned that if an operator commutes with the Hamiltonian, the expectation value of that operator is a conserved quantity. The same concept holds in relativistic quantum mechanics but is expressed using Lagrangian language. The Lagrangian can be expressed as

$$L = T - V$$

where T is the kinetic energy of the system and V is the potential energy of the system. The action, S, of a trajectory is defined as the integral of the Lagrangian with respect to time during the trajectory, i.e.

$$S = \int L \, dt$$

.

Noether's theorem tells us that for every symmetry of the action there is a conserved quantum number.

Consider the following example for the case of non-relativisitic quantum mechanics. A particle with spin-$\frac{1}{2}$ and magnetic moment $\vec{\mu} = g\frac{q}{2m}\vec{S}$ where q is the charge of the particle and $\vec{S}$ is the spin is placed in a constant magnetic field in the z-direction $\vec{B} = B_0\hat{z}$. The Lagrangian is therefore

$$L = \frac{\left(\vec{p}\right)^2}{2m} + \vec{\mu} \cdot B$$

Which of the following are conserved quantities:

- (a) p_x, the x-component of the momentum
- (b) p_z, the z-component of the momentum
- (c) S_x, the x-component of the spin
- (d) S_z, the z-component of the spin
- (e) $\vec{S}^2$ the magnitude of the spin-squared operator

According to Noether's theorem, there's a 1:1 relation between symmetries and conservation laws. In QM, this means that

$$[\hat{H}, \hat{O}] = 0 \rightarrow < \hat{O} > \; conserved$$

a) $[\overrightarrow{p_x}, \vec{p}^2] = 0, [\overrightarrow{p_x}, \hat{S_z}] = 0$

$\therefore p_x$ conserved

b) $[\overrightarrow{p_z}, \vec{p}^2] = 0, [\overrightarrow{p_z}, \hat{S}_z] = 0$

$\therefore p_z$ conserved

c) $[\overrightarrow{S_x}, \vec{p}^2] = 0, [\overrightarrow{S_x}, \hat{S}_z] \neq 0$

$\therefore S_x$ NOT conserved

d) $[\overrightarrow{S_z}, \vec{p}^2] = 0, [\overrightarrow{S_z}, \hat{S}_z] = 0$

$\therefore S_x$ NOT conserved

e) $[\overrightarrow{S^2}, \vec{p}^2] = 0, [\overrightarrow{S^2}, \hat{S}_z] = 0$

$\therefore S^2$

Note that $\hat{S}$ and $\vec{p}$ occupy different eigenspaces, so they'll always commute. The reason we check for $\vec{p}^2$ and $\hat{S}_z$ (comes from $\vec{S} \cdot \hat{z}$) is because $\hat{H}$ is composed of both of these operators

Question 3: Parity Properties of Various Operators (10 points)

Learning objectives

In this question you will:

- Review the meaning of the terms vector, axial vector, scalar and pseudoscalar and determine the parity property of several operators

We saw in class that operators could be classified according to their properties under the parity operator. Vector operators change sign under parity

$$\mathbf{P}\ \vec{r}\ \mathbf{P}^\dagger \rightarrow -\vec{r}$$

while axial vectors do not:

$$\mathbf{P}\ \vec{L}\ \mathbf{P}^\dagger \rightarrow +\vec{L}$$

similarly, scalar operators do not change sign under parity

$$\mathbf{P}\ \left(\vec{r} \cdot \vec{p}\right)\ \mathbf{P}^\dagger \rightarrow +\vec{r} \cdot \vec{p}$$

while pseudoscalar operators do change sign:

$$\mathbf{P}\ \left(\vec{p} \cdot \vec{L}\right)\ \mathbf{P}^\dagger \rightarrow -\vec{p} \cdot \vec{L}$$

For each of the operators below, state whether they are scalar, pseudoscalar. vector or pseudovector:

- (a) $\vec{p}_1 \cdot \left(\vec{p}_2 \times \vec{p}_3\right)$ where 1. 2 and 3 are three different particles
- (b) $\vec{p}_1 \times \left(\vec{p}_2 \times \vec{p}_3\right)$ where 1. 2 and 3 are three different particles
- (c) The magnetic moment $\vec{\mu}$ of a particle

- (d) The magnetic field $\vec{B}$;; (Hint: look at the Biot-Savart law)
- (e) $\left(\vec{p}_1 \times \vec{p}_2\right) \cdot \left(\vec{S}_1 + \vec{S}_2\right)$ where 1 and 2 are two different particles

Some rules:

- vector × vector = pseudovector
- pseudovec × pseudovec = pseudovec
- vector × pseudovec = vector
- pseudovec × vector = vector

Based on the above, then:
a) $\overrightarrow{p_1} \cdot (\overrightarrow{p_2} \times \overrightarrow{p_3}) =$ vector · pseudovec = pseudoscalar
b) $\overrightarrow{p_1} \times (\overrightarrow{p_2} \times \overrightarrow{p_3}) =$ vector × pseudovec = vector
c) Magnetic moment is invariant under a coordinate axes inversion, and therefore $\vec{\mu} =$ pseudovector
d) A different argument than using Biot-Savart: we know that torque is a psuedovector as well. We also know

$$\tau = \vec{\mu} \times \vec{B}$$

Thus, this above relation + 2nd bullet point from previous segment means that $\vec{B}$ must also be a pseudovector
e) $\vec{S} = k\vec{\mu}$, so $\vec{S}$ must also be a pseudovector, so
pseudovec · pseudovec = scalar, ∴ scalar

Question 4: Parity in Particle Decays (10 points)

Learning objectives

In this question you will:

- Explore how conservation laws can be used to exclude specific decay channels

Show that a scalar meson (a meson with spin 0 and parity +1) cannot decay to three pseudoscalar mesons (mesons with spin 0 and parity -1) in a parity-conserving process such as the strong interaction.

Parity must be conserved in a strong interaction. Total parity is calculated by multiplying the parities of each of the constituents at each stage. Thus,

$$U_P m_s U_P^{-1} = +m_s \rightarrow \therefore \pi_{m_s} = +1$$
$$\pi_{final} = \pi_{f_1}\pi_{f_2}\pi_{f_3} = (-1)(-1)(-1) = -1$$

∴ This decay cannot be the result of a strong interaction because the starting parity is not conserved.

Question 5: Λ^0 polarization in the strong interactions (20 points)

Learning objectives

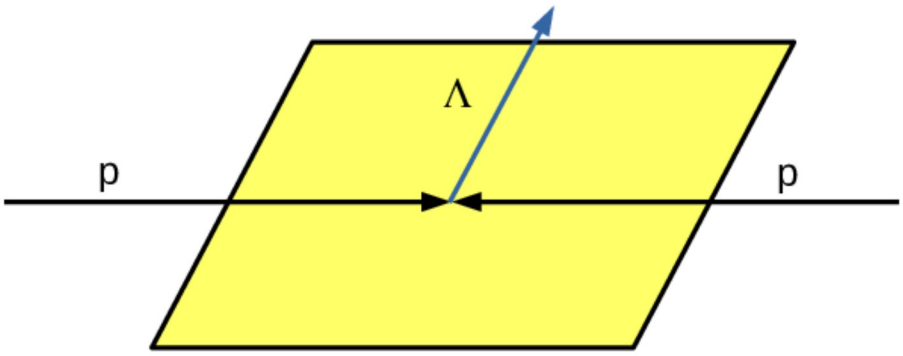
In this question you will:

- Apply the concept of parity conservation in the strong interactions to determine what functional forms are allowed for a specific observable (the polarization)

Consider the production of Λ particles via a strong interaction:

$$p + p \rightarrow \Lambda^0 + X$$

where X represents one of more additional final state particles that are not necessarily observed. Assume that the Λ^0 and X each have the same parity as the proton. Also assume that the initial protons are unpolarized. In the center-of-mass frame, the momentum vectors of the two protons and the Λ^0 all lie in a plane, called the production plane:



It has been observed that Λ^0 (which are spin 1/2 particles) are polarized. This means that the expectation value $\left\langle \vec{S}_\Lambda \cdot \hat{n} \right\rangle \neq 0$ for certain unit vectors $\hat{n}$. Here $\vec{S}_\Lambda$ is the spin vector of the Λ^0.

In this problem, you will determine what possible directions $\hat{n}$ can have.

(a) Show that if $\hat{n}$ lies in the production plane, then $\left\langle \vec{S}_\Lambda \cdot \hat{n} \right\rangle = 0$ if parity is conserved

If $\hat{n}$ is in the same plane as the momenta of the protons, it can be said that $\hat{n}$ shares the same plane as

$$a\overrightarrow{p_1} + b\overrightarrow{p_2}$$

which is a linear combination of vectors. This result means that $\hat{n}$, in this case, must be a vector (as opposed to a pseudovector). We know also, from question 4, that $\hat{S}$ is a pseudovector, which has even parity. In addition, we also know that pseudovector $\cdot$ vector $=$ pseudoscalar, a quantity that does change sign under parity inversion. So if we were to find the expectation value of a function of odd parity over finite range (like integrating over 1D wavefunctions in a box), we would find that integral to come out to zero. Therefore, due to parity considerations we have to conclude that

$$\left\langle \overrightarrow{S_\Lambda} \cdot \hat{n} \right\rangle = 0$$

(b) Show that a non-zero value of $\left\langle \vec{S}_\Lambda \cdot \hat{n} \right\rangle$ is possible if $\hat{n}$ is perpendicular to the production plane

Hints: 1) In the center of mass frame of two unpolarized colliding particles of the same species, the scattering rate must be be a symmetric function of $\cos\theta^*$ (the polar angle with respect to the pp direction) since there is no distinction between the forward and backward directions.
2) Work in the center-of-mass frame since this is where the production plane is defined. In each case, express $\hat{n}$ as a function of the 3-momenta of the specified particles.

In this case,

$$\hat{n} = \overrightarrow{p_1} \times \overrightarrow{p_2}$$

This makes $\hat{n}$ a pseudovector. From question 4, we know that $\hat{S}$ is also a pseudovector. Also from Q4, we know that pseudovector $\cdot$ pseudovector $=$ scalar. Because pseudovectors each have even parity, the resulting scalar is also of even parity (i.e. doesn't flip signs upon inversion). Therefore, $< \vec{S} \cdot \hat{n} >$ can be a non-zero value (still can be, but doesn't have to be).

In [ ]: