Tanmay Garg

CS20BTECH11063

Data Science Analysis Programming Exam

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy import optimize
import astroML
from astroML.stats import sigmaG
import pandas as pd
import seaborn as sns
np.random.seed(0)
import emcee
import corner
from sklearn.neighbors import KernelDensity
from IPython.display import display, Math
import dynesty
# import nestle
from astroML.correlation import bootstrap_two_point_angular
```

Q1

In [ ]:
```
hubble_cons1 = 67
hubble_cons_error1 = 2.3

hubble_cons2 = 71
hubble_cons_error2 = 1.3

def z_score(hubble_cons1, hubble_cons_error1, hubble_cons2, hubble_cons_error2):
    return abs((hubble_cons1 - hubble_cons2)/np.sqrt(hubble_cons_error1**2 + hubble_cons_error2**2))

print("Z Score of hubble constant: ", z_score(hubble_cons1, hubble_cons_error1, hubble_cons2, hubble_cons_error2))

# # calculate the Z Score of the two hubble constants in terms of sigma
# z_score1 = (hubble_cons1 - hubble_cons2)/np.sqrt(hubble_cons_error1**2 + hubble_cons_error2**2)
# z_score2 = (hubble_cons2 - hubble_cons1)/np.sqrt(hubble_cons_error1**2 + hubble_cons_error2**2)
```

```
# print("Z Score of hubble constant 1: ", z_score1)
# print("Z Score of hubble constant 2: ", z_score2)

# # calculate the p value of the two hubble constants
# p_value1 = stats.norm.sf(abs(z_score1))
# p_value2 = stats.norm.sf(abs(z_score2))
# print("P value of hubble constant 1: ", p_value1)
# print("P value of hubble constant 2: ", p_value2)

# print(stats.norm.isf(p_value1))
# print(stats.norm.isf(p_value2))
```

```
Z Score of hubble constant:  1.5140223282089291
```

Q2

What values of constant delta chi square contours should be used for a model with 2 free parameters, 10 data points if we want 85% confidence?

In [ ]:
```
stats.chi2(2).ppf(0.85)
print("The constant delta chi squared value is: ", stats.chi2(2).ppf(0.85))
```

```
The constant delta chi squared value is:  3.794239969771762
```

Q3

In [ ]:
```
reduced_chi2_A = 1.3
reduced_chi2_B = 0.7

num_free_param_A = 6
num_free_param_B = 8

num_data_point_fit = 10

# Calculate BIC between the two models

def BIC_func1(num_free_param, num_data_point_fit, reduced_chi2):
    return num_free_param*np.log(num_data_point_fit) + reduced_chi2

def BIC_func2(num_free_param, num_data_point_fit, reduced_chi2):
    return num_free_param*np.log(num_data_point_fit) - 2 * np.log(reduced_chi2 * num_free_param)

# BIC_A = BIC_func1(num_free_param_A, num_data_point_fit, reduced_chi2_A)
# BIC_B = BIC_func1(num_free_param_B, num_data_point_fit, reduced_chi2_B)
```

```python
# print("BIC of model A: ", BIC_A)
# print("BIC of model B: ", BIC_B)
# print("BIC of model A - B: ", BIC_A - BIC_B)


BIC_A = BIC_func2(num_free_param_A, num_data_point_fit, reduced_chi2_A)
BIC_B = BIC_func2(num_free_param_B, num_data_point_fit, reduced_chi2_B)
print("BIC of model A: ", BIC_A)
print("BIC of model B: ", BIC_B)
print("BIC of model A - B: ", BIC_A - BIC_B)
```

```
BIC of model A:  9.707263090573182
BIC of model B:  14.97514754847016
BIC of model A - B:  -5.267884457896978
```

Q4

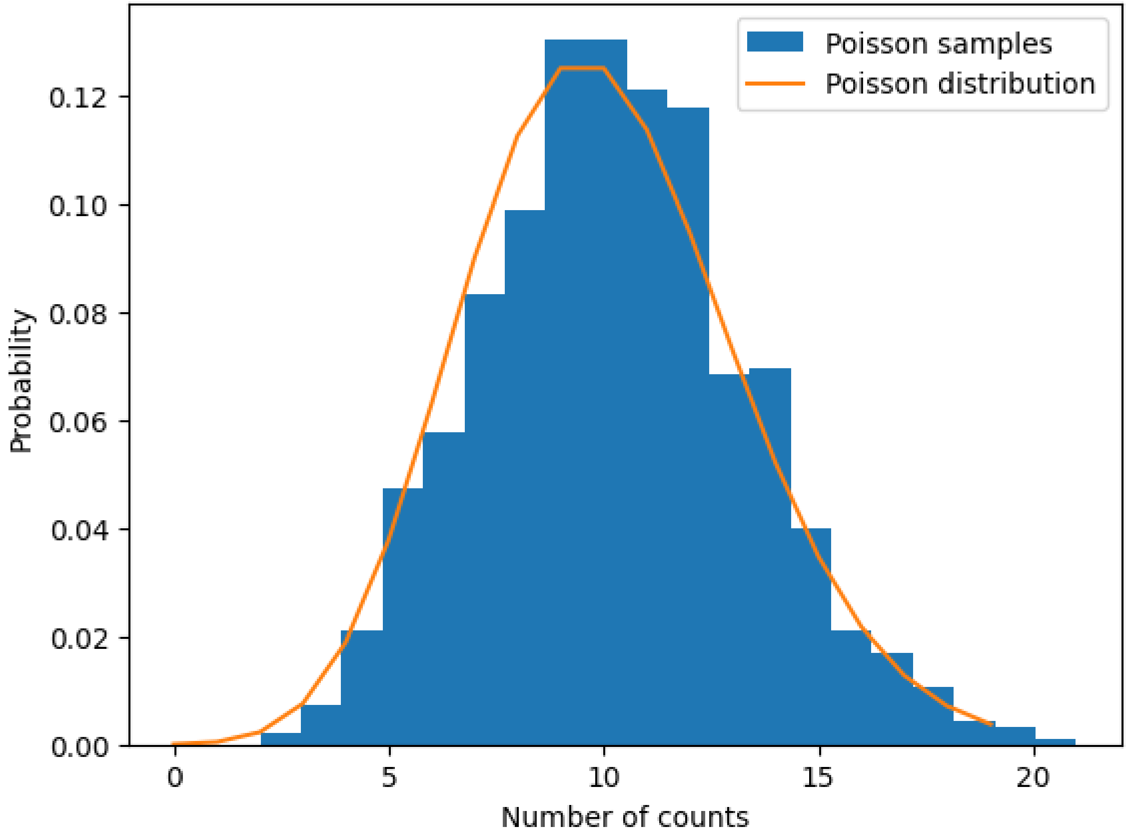
In [ ]:

```python
# Show the code for generating 1000 Poisson distributed random numbers with mean 10

mean = 10
num_samples = 1000
poisson_samples = np.random.poisson(mean, num_samples)
print("Poisson samples: ", poisson_samples)
print("Mean of poisson samples: ", np.mean(poisson_samples))

# plot the histogram of the poisson samples and compare it to the poisson distribution
plt.figure()
plt.hist(poisson_samples, bins=20, density=True, label="Poisson samples")
plt.plot(np.arange(0, 20), stats.poisson.pmf(np.arange(0, 20), mean), label="Poisson distribution")
plt.xlabel("Number of counts")
plt.ylabel("Probability")
plt.legend()
plt.show()
```

```
Poisson samples:  [ 6  9 10  8  9 13 12  9  7 11  8  9  9 15 13 11 13 10  7 12 12  9 14 12
 10 13 12  9 10 11 10 11 10  8  4 15 11  9 11 12  7  7 18 14  7 10  6  8
  7  7  9  8 11  6 13 11 13 12 12 13 11  4 10  9  4  9  8 14  8  7  9  9
 12 12  7  9  3 15 11 10 13  8  6 14 12  7 13 12  6 10 11  7  2 17  8 11
 12 10  8  7 10 15  7 16  8 10 11 12 18 10  7  5 11 10  9  5  9 12 12  6
 12 11  3  8  6  8 17  9  5  8 14  6 11  4  7  8 14 15 10 13 15  8 15 13
 10 15  7 12  9 13 12 10 10 12  8 20  8 12  8 10 12  9  8  9 12  5 10 20
 11 18  8  9 12  7 12 11 10 16 11 11  6  8 10 14 12 14 13  8  4  8  8 15
 11 10  9  9 15  9 13 10 11  9 10  9  7  9 12 14 13  9  7  6  6 16 14  7
 12  9 14 12 10  6 10 13 13  9 12 10 17 11 14 10 11  8 13 13 13  7  9 11
  7  4  6 15 11  6 11  9 14 10 12  8  7 10  9 11 13  5 10  7 11 10  6  8
 14 13 16 12 11  8 12 14 12  9  5 10 11  9  8  8  9 12 10  7  8 12 14 18
  7 10  5  6  9  8  8 10 13  8 11 10 11  6  9 10  9 10  9  7  8 13  9 12
  9  8  5 14  7  8  8 12 12 10  8  7  7  5 10  8 15  9  7  4 11 16  3 12
 10  7 10 11 11 10 14 11  7  9  7 14  3  9 10 14 11  9 13  7  9 12 12 11
 10 14 11  9 19 16 13 13 12 13 10 12 12 10 12 17  6 10 11 12  9 16  9  9
 17  8  4  9  5  9  9  8  5  6 11 11 11 14 10  5 12  7 15 13 16  8 12  8
 18  9  6 11 12  5  7 15 16 12  9 13 14 14  6  6  9 12  4  4 19 12 13 13
  4 14 16  9 12  5 11  8 12  5 12 14  4 14 15 11 14  4 10 10  9  9  8  3
 14 10 17 12  9  6 15 14  9 12  8 12  7 11  5 12 14  6 13 12 13  4  9 10
  9 10 12  6 10  9  5 10 11 17  8  9  9 16 16 15 11 18  5 12  7 10  7  9
  8  6 17  6  6 12  7 15 11  9  6 14  9  7 12  8  4  7 11  8 11  9 10 12
  7 10 10 15  8 12 15  7 12 11  7 11 11 13  8 10  7  8 18 10  9 10  9  7
  6 10 13 15 15  7 13  5  7  3 13  9 10 10 13  2 10  8 16  5  9 10 11 11
  6 14 10 12  6  6 14 12  5  8 15 10 10 12 11 11  9 12 10 12  8  8  9  9
  8  8 11  9 10 15  8  5 16 11 12 13 10  6  7 11  6  6  7 10 13  8 12 12
  8 14 10 11 13 10 12 10 17 11  5 14  8 14 10 10 15  7  7 11 10  9 12 11
  3 11  7  5  5 11 10 11  9 10  9 11 10  9  5  7 10  6  9  9 13  8 11  9
 10 11  5 12 12 14 10  7 13  7  9 12 12 11 11  8 10 13  8 18 14 10 11 15
 13 12  5  9 10  4  8 15  9 11 11 14 14 12 16  6 14 11 13 10  6 12  6 11
 14  5 12 11  6  9  8  9  9 10 11  9 13 17  5 12 14  6  8 10 14 11  9 10
 10 13  5  9  8 12  5  8 13 13 11 11 11  6  4  9 13  8 10 15  7 14  9 12
  5 17  8 10  4 10 10 21 12  7 11  5 11  8  8  8 13  8  8  5 11  4  7 20
  8  8  7 15  7 14 14  4 17 14 17  7 13 12 15 13 13 17  5 19 11  6  6 11
 12 15  9  6 10 14  9 10 10  6  9 12 12 10 14 15  9 14  8 11 16  9 10 10
  9 16 14  9  8  6 10 11  9 14 14  8 10  6 13 12  8  9  5  8 11 12 12  5
  7  9 12 18  9  9  9 12 11  8 11 11 17 14  6  7  5  5  9  7 12  7 11 13
  8 12 10 10  6 12 13  5  6  9 12 15 11 16 12 10 12  9  9 12 11 17  7 14
 15 19  7 11 10 12 10 14 10 11  8  8 10 16  9 14 11  9 11 10 12  8  7  5
 11  9  6  7  7 11 13 11 18  7 13 10  9 13 11 13  6 14  7  6  9  5 11  9
  5 11 11  7 13 14 12 14 10 11  9 11  9 12  7 11 12 10 15 12 16  8  8 14
 15 15  9 10  5 15 14  7 13  9 10 14 11 14  7 10]
Mean of poisson samples:  10.198
```


Poisson samples
Poisson distribution
Probability
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
Number of counts