es21btech11025-quiz3

April 26, 2024

EP4130: Data Science Analysis

Final - Quiz 3

Ranveer Sahu- ES21BTECH11025

[23]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
```

Q1 The p-value of an experiment looking for a new particle is 2.3 × 10−6 .What is the Z-score?

[24]:
```
from scipy.stats import norm
#assumption is particle 2-tailed
p_value = 2.3e-6
#taking inverse to find z_score
z_score = norm.ppf(1 - p_value/2)  # z_score from p_value

print("Z-score:", z_score)
```

```
Z-score: 4.725101895193788
```

Q2

What values of constant 2 contours should be used for a model with 4 free parameters, 10 data points, if we want to get 80% confidence intervals? [2 pts]

Answer:

$ ˆ2 $ is distributed with #free params degree of freedoms. Thus, we require $\Delta\chi^2$ such that

$\Pr(\chi^2 < \Delta\chi^2, \#\text{free params}) = p\%$

[25]:
```
dof = 4
confidence_percentage = 0.80
print("The constant countour value is {:.3f}".format(sp.stats.chi2(dof).
 ↪ppf(confidence_percentage)))
```

```
The constant countour value is 5.989
```

Q3

Two models (A and B) have reduced 2 of 1.4 and 0.7 for 6 and 8 number of free parameters, respectively. Calculate BIC between Model A and model B. Assume both models have been contain 10 data points.

[26]:
```
# BIC = -2 ln(L) + M ln(N)

M1 = 6
M2 = 8

likelihood1 = 1.4 * (10 - 6)
likelihood2 = 0.7 * (10 - 8)

def BIC(likelihood , N  , M):
    return -2 * np.log(likelihood) + M * np.log(N)

BIC1 = BIC(likelihood1 , 10 , 6)
BIC2 = BIC(likelihood2 , 10 , 8)

print("The difference in BIC of model A and model B is {}".format(np.abs(BIC1 -␣
  ↪BIC2)))
```

```
The difference in BIC of model A and model B is 7.377758908227875
```

Q4 Write down the subroutine for Nested sampling using Dynesty for a model with two free parameters 1 and 2 with uniform priors for both of them given by 1 U[−10, 5] and 2 U[0, 20] respectively.

[27]:
```
pip install dynesty
```

```
Requirement already satisfied: dynesty in /usr/local/lib/python3.10/dist-
packages (2.1.3)
```

[28]:
```
#installing necessary
import dynesty
```

[29]:
```
def log_prior(theta):

    theta1, theta2 = theta
    if -10 <= theta1 <= 5 and 0 <= theta2 <= 20:
        return 0.0  # log(1) = 0 because parameters are within prior bounds
    else:
        return -np.inf  # log(0) = -inf because parameters are outside prior␣
  ↪bounds
```