**Problem Set 4**

**Research method Problem Set 4 due Fri 23th Nov, 23:59**

**LIU BOWEN (1004028)**

For Problem Set 4, I use the following packages:

|  |
| --- |
| import matplotlib.pyplot as plt  import numpy as np  import scipy as sp  import scipy.stats as ss  import math |

**Problem 1**

Answer:

Follow the procedure of MLE, first define the PDF and then define the likelihood function. For this case, the beta value is the maximum likelihood value in the trials. The code is below:

|  |
| --- |
| def time (x, b):  return 1.0/b \* math.exp (-x / b)  def likelihood (trials, arrival):  problist =[]  for line in trials :  prob = 1.  for l in arrival :  prob = prob \* time(l, line)  problist.append(prob)  return problist, trials[problist.index(max(problist))]  arrive = [3.2, 2.1, 5.3, 4.2, 1.2, 2.8, 6.4, 1.5, 1.9, 3.0]  expeTrials = np.linspace(0.0001, 10, 10000)  likely, b = likelihood(expeTrials, arrive)  print 'The max likelihood estimate beta is:', b  plt.figure()  plt.plot (expeTrials, likely)  plt.xlabel ('beta lambda')  plt.ylabel ('Likelihood') |

I set the trial range np.linspace(0.0001, 10, 10000) and meanwhile, obtain the max value in trials[]. The result is shown in Figure 1 and the plot figure is shown in Figure 2.



Figure 1

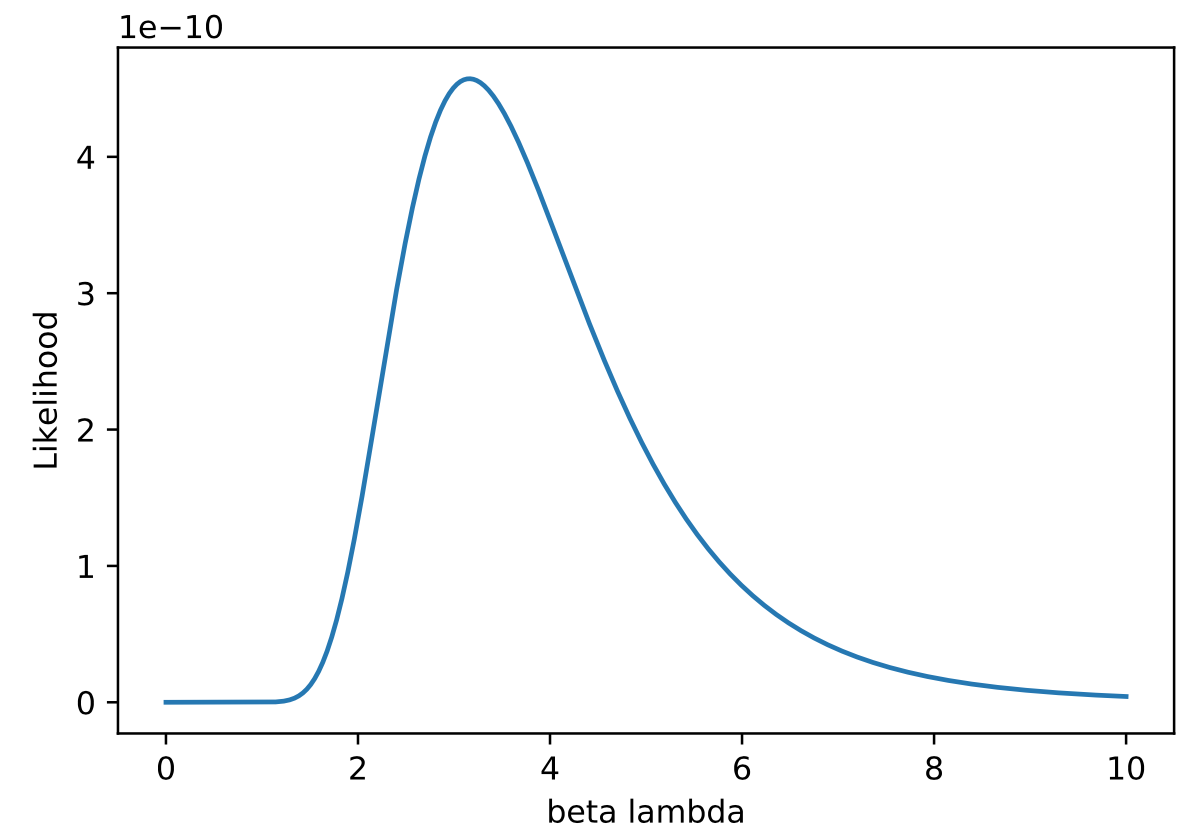


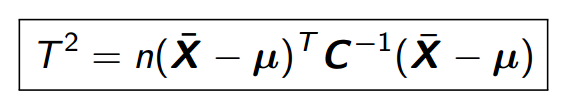
Figure 2

**Problem 2**

Answer:

1)

I use the equation 1 to calculate the T2 where X is a vector of sample means, C-1 is a vector of the corresponding variance-covariance matrix, and µ is a vector of the corresponding population means. The code is shown below:



Equation 1

|  |
| --- |
| def Ex2():  meanMat = np.mat([60, 24, 161])  whole = np.mat([[65, 28, 175], [80, 26, 159], [70, 27, 180], [62, 29, 167], [74, 24, 170]]).T  C = np.cov(whole)  sample1 = [65, 80, 70, 62,74]  m1 = np.mean(sample1)  sample2 = [28, 26,27, 29, 24]  m2 = np.mean(sample2)  sample3 = [175, 159, 180, 167, 170]  m3 = np.mean(sample3)  meanSample = np.mat([m1, m2, m3])  T\_square = 5.0 \* (meanSample - meanMat) \* np.matrix(C).I \* (np.mat([m1, m2, m3]).T-np.mat([60, 24, 161]).T)  F = (5.0 - 3.0) / (3.0\*(5-1)) \* T\_square  Fc = ss.f.ppf(0.99, 3, 5-3)  print T\_square, F, Fc  Ex2() |

The result is:

T2 = 173.22918028,

F = 28.87153005,

Fc = 99.16620137447148

2)

At the α= 0.01 significance level, the critical F-value F < Fc therefore there is no sample significantly different from the population means.

**Problem 3**

1)

The error(within) for DoF is 16 and for SS is 159.89. MSB = SSB/k-1 = SSB/3 = 12.05 and MSW = SSW/n-k= SSW/16 = 9.9931. F = MSB/MSW = 1.2058

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source | DoF | SS | MS | F | P |
| Factor (between) | 3 | 36.15 | 12.05 | 1.2058 | 0.66 |
| Error (within) | 16 | 159.89 | 9.9931 |  |  |
| Total | 19 | 196.04 |  |  |  |

The code is below:

|  |
| --- |
| H1 =[98,97,99,96]  H2 =[91,90,93,92]  H3 =[96,95,97,95]  H4 =[95,96,99,98]  def SSW (\*arg):  n= len(arg[0])  levels = len(arg)  ssw = 0  for i in range(levels):  var =np.var(arg[i], ddof=0)  ssw = ssw + var  df =(n\*levels)-levels  return (n\* ssw), df  def SSB (\*arg):  grandmean =0  levels = len(arg)  for i in range(len(arg)):  grandmean = grandmean +np.mean(arg[i])  grandmean = grandmean /len(arg)  ssb =0  for i in range (len(arg)):  temp =np.mean(arg[i])  ssb = ssb +(temp-grandmean) \*\*2  ssb = len(arg[0])\*ssb  df= levels-1  return ssb,df  ssw, sswdf = SSW(H1 , H2 , H3, H4)  ssb, ssbdf = SSB(H1 , H2 , H3, H4)  sst1 = np.var(H1 + H2 + H3 + H4)\* len(H1+ H2 +H3 + H4 )  sst2 = ssw + ssb  msw = ssw / sswdf  msb = ssb / ssbdf  F= msb / msw  Fc=ss.f.ppf(0.95,3,12)  print " SSW =", ssw  print " MSW =", msw  print " SSB =", ssb  print " MSB =", msb  print "F=", F  print "Critical F=", Fc  if F > Fc:  print "F>Fc , Reject H0"  else :  print "F<Fc , Accept H0" |

The answer is shown in Figure 3:

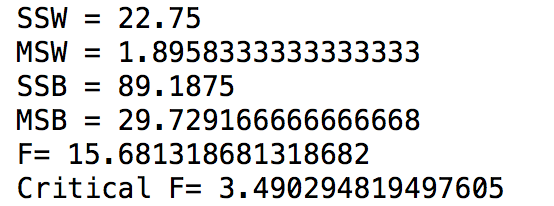


Figure 3

2) SSB = 89.1875

3) MSB = 29.729166666666668

4) SSE = 22.75

5) MSE = 1.8958333333333333

6)F = 15.681318681318682 > Fc = 3.490294819497605, so we reject H0 there is a significant difference between the looms at the 0.05 significance level.