Parameter Estimation: MLE and MAP

In [1]:
```
# import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy
from scipy import stats
```

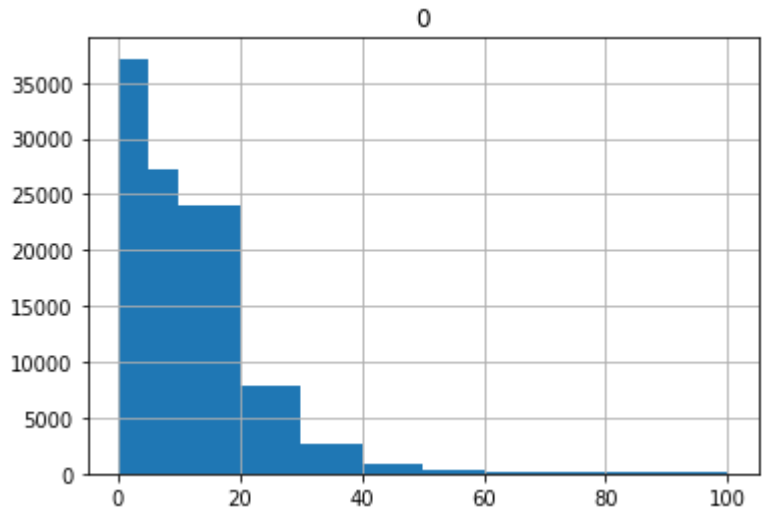
Read the dataset into `elems` array

In [2]:
```
# load data from 'parameter_estimation_dataset.txt' file into a variable named
"elems"
elems = pd.read_csv('parameter_estimation_dataset.txt', header=None)
```

Visualize data to see if it follows Geometric distribution.
Plot the histogram of `elems` for the `bins` =[0, 5, 10, 20, 30, 40, 50, 60, 100]

In [3]:
```
# Plot histogram of elems for the given bins parameter values
bins = [0, 5, 10, 20, 30, 40, 50, 60, 100]
elems.hist(bins=bins)
```

Out[3]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000020235AD70F0
>]],
      dtype=object)
```



1. (8 pts) Maximum Likelihood Estimation

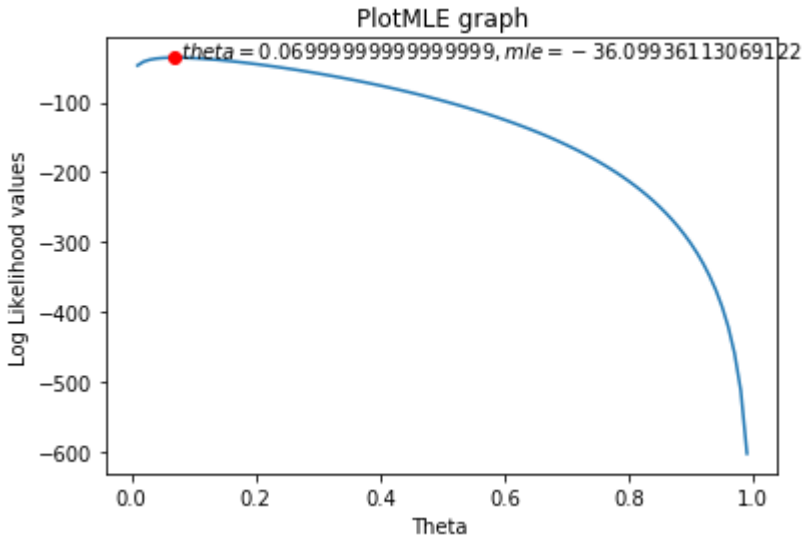
i. (4 pts) We will compute an approximation of the MLE, by just computing the maximum of the log-likelihood function over a given finite set of candidate parameters. Write a function `plotMLE(X, theta)` that takes as input a set of samples, and a set of candidate parameters θ, and produces a plot with the log-likelihood function $\ell(\theta)$ on the Y-axis, candidate parameters θ on the X-axis, and also mark that candidate parameter $\hat{\theta}$ from the given set of candidate parameters with the maximum log-likelihood (as the approximate MLE).

In [4]:
```
def plotMLE(X,theta):

    num_of_seq = len(X)  # number of elements in elems
    sum_of_seq = X.iloc[:,0].sum()  # sum of the elems

    log_likelihood = [np.log(1-t)*sum_of_seq + num_of_seq*np.log(t) for t in t
heta] # compute log-likelihood wrt each theta. It will be an array

    mle = np.amax(log_likelihood) # find the max of log_likelihood. This will
 be our mle estimate
    mle_index = log_likelihood.index(mle) # find the index for which we have m
le estimate

    X = theta[mle_index] # select the best theta based on mle_index and store
 it in variable X

    # plotting
    plt.plot(theta, log_likelihood,X,mle,'ro',label='mle')
    plt.text(X, mle, r'$\ theta={},mle={}$'.format(X, mle))
    plt.xlabel('Theta ')
    plt.ylabel('Log Likelihood values')
    plt.title('PlotMLE graph')
    plt.show()
```

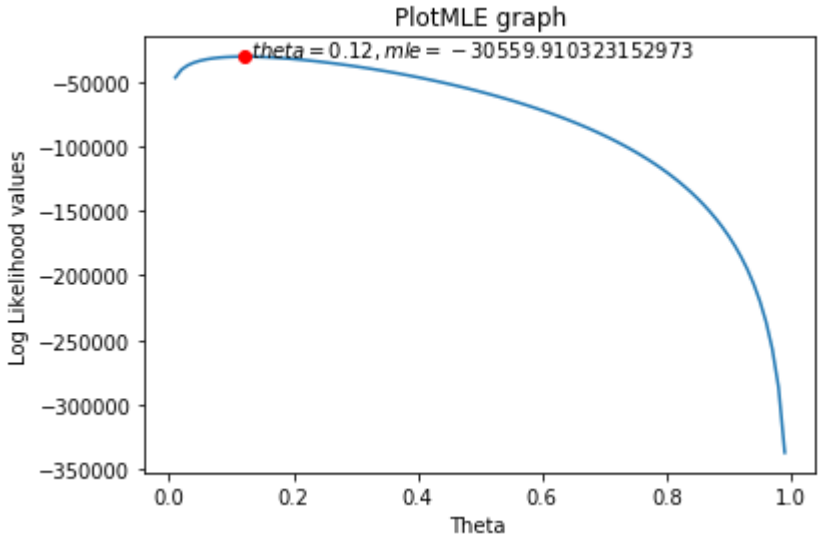
ii. (4 pts) Consider the following sequence of 100000 samples (stored in `parameter_estimation_dataset.txt`) obtained from Casino Coruscant on number of trials to first win on a slot machine.
Use your program to produce three plots: (a) with the first ten samples, (b) with the first ten-thousand, and (c) with all hundred-thousand. For each of the three plots, for the set of candidate parameters use $0.01, 0.02, \ldots, 0.99$. What do you observe from the resulting plots? Does the estimate change across the three plots? If yes, what is its trend?

I observe that the MLE estimate's theta for 10 data points is .06999, .12 for 10,000 data points, and .09999 for all 100,000 data points. The trend is that theta increases from 10 to 10,000 points, but then decreases slightly when going to 100,000.

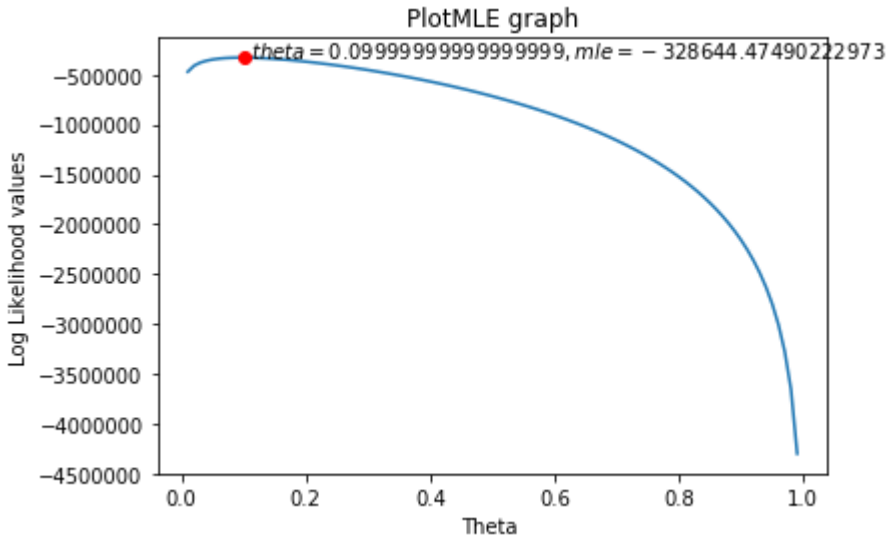
In [5]:
```
# call plotMLE for first ten elems
plotMLE(elems[:10], theta=np.arange(0.01,1,0.01))
```



In [6]:
```
# call plotMLE for first ten-thousand elems for theta = arange(0.01,1,0.01)
plotMLE(elems[:10000], theta=np.arange(0.01,1,0.01))
```



In [7]:
```
# call plotMLE for all elems for theta = arange(0.01,1,0.01)
plotMLE(elems, theta=np.arange(0.01,1,0.01))
```



2. (12 pts) Maximum a Posteriori Estimation

i. (6 pts) Write a function `plotMAP(X,theta,alpha,beta)` that that takes as input a set of samples, and a set of candidate parameters θ, a value for alpha, and a value for beta, and produces a plot with the log-posterior function $\ell(\theta)$ on the Y-axis, candidate parameters θ on the X-axis, and also mark that candidate parameter $\hat{\theta}$ from the given set of candidate parameters which has the maximum posterior density (as the approximate MAP). [Note : Use Beta distribution for prior.

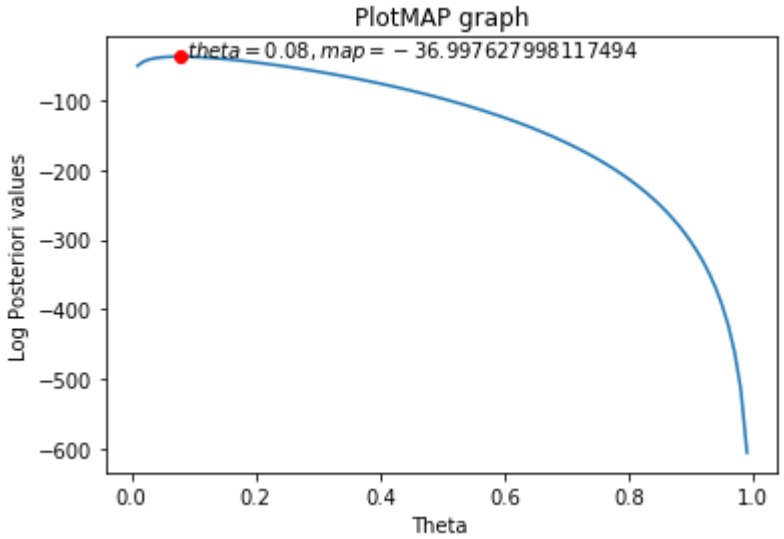
$$Beta(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

, where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and $\alpha > 0, \beta > 0$.]

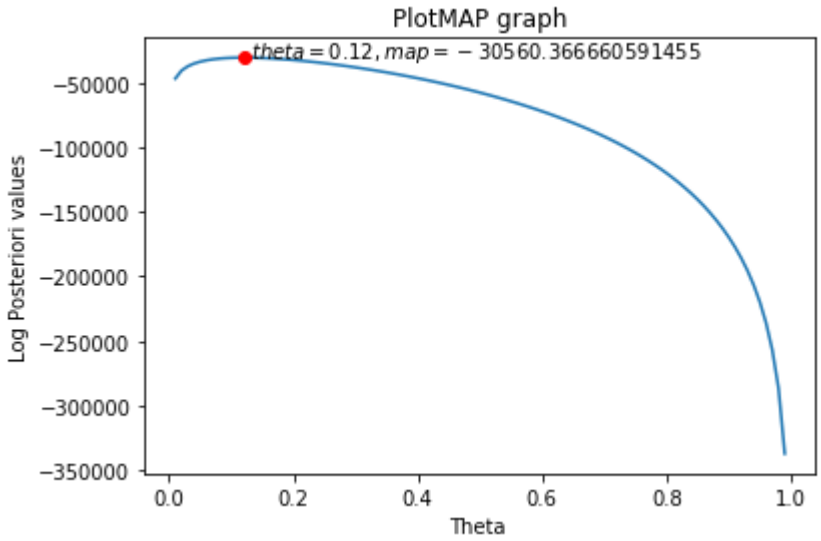
In [8]:
```
def plotMAP(X,theta,alpha,beta):

    num_of_seq = len(X)  # number of elements in elems
    sum_of_seq = X.iloc[:,0].sum()  # sum of the elems

    prior = [scipy.stats.beta.pdf(t, alpha, beta) for t in theta]  # Hint: You
can use scipy.stats.beta for computing the prior
    log_posteriori = [np.log(1-theta[v])*sum_of_seq + num_of_seq*np.log(theta[
v])+np.log(prior[v]) for v in range(len(prior)) ]

    MAP = np.amax(log_posteriori) # find the max of log_posteriori. This will
 be our map estimate
    map_index = log_posteriori.index(MAP) # find the index for which we have m
ap estimate

    X = theta[map_index] # select the best theta based on map_index and store
 it in variable X

    # plotting
    plt.plot(theta, log_posteriori,X,MAP,'ro',label='map')
    plt.text(X, MAP, r'$\ theta={},map={}$'.format(X, MAP))
    plt.xlabel('Theta ')
    plt.ylabel('Log Posteriori values')
    plt.title('PlotMAP graph')
    plt.show()
```

ii. (4 pts) Redo the three plots you made in the previous part, but with the log-posterior function instead, and mark the MAP estimators. Set $\alpha = 2$, $\beta = 2$.

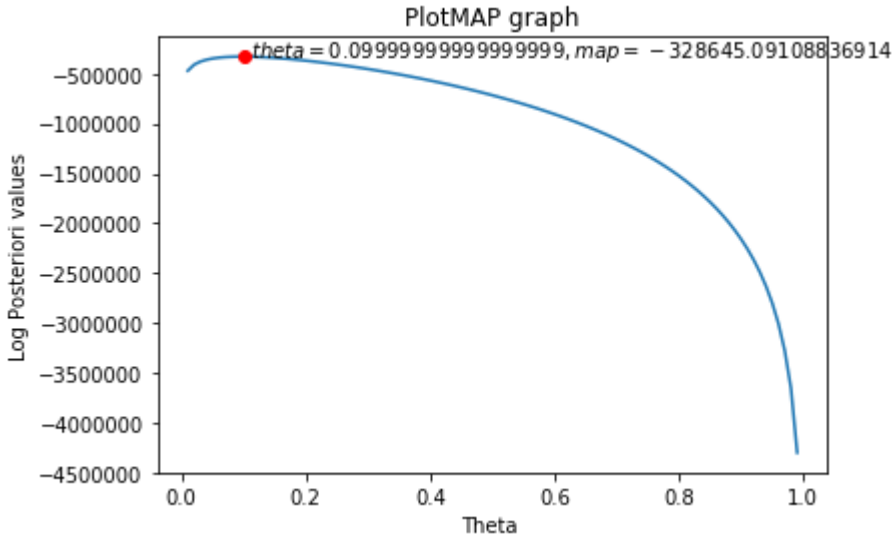
In [9]:
```
# call plotMAP for first 10 elems for theta=(0.01,1,0.01), and approriate prio
r parameters
plotMAP(elems[:10], theta=np.arange(0.01,1,0.01), alpha=2, beta=2)
```



In [10]:
```
# call plotMAP for first ten thousand elems for theta=(0.01,1,0.01), and appro
riate prior parameters
plotMAP(elems[:10000], theta=np.arange(0.01,1,0.01), alpha=2, beta=2)
```



In [11]:
```
# call plotMAP for all elems for theta=(0.01,1,0.01), and approriate prior par
ameters
plotMAP(elems, theta=np.arange(0.01,1,0.01), alpha=2, beta=2)
```



iii. (2 pts) Do you see any significant differences between the MLE and MAP estimates? Why or why not? Explain in 1-2 sentences.

Yes, I see a difference in the MLE and MAP estimates for the small selection of elements (elems[:10]), and this makes sense because the impact of the prior distribution will have more impact for less data - as the number of observations increases, the effect of prior beliefs on the estimation decreases, and so when we are using all 100,000 data points the MLE and MAP estimates are very similar or the same.