3.6_Likelihood_Weighting

October 23, 2017

In [173]:
```
import random
import collections
import matplotlib
import matplotlib.pylab as plt

def bin_to_dec(B):
    dec = 0
    for i in range(0,10):
        dec += B[i] * (2**i)
    return dec


def calc_alpha_powers(alpha, b_samples):
    alpha_powers = []
    for B in b_samples:
        alpha_powers += [alpha**abs(128 - bin_to_dec(B))]
    return alpha_powers

def do_likelihood_weighting(size,q,z):
    alpha = 0.1
    c = (1-alpha)/(1+alpha)
    b_samples=[]
    for i in range(0,size):
        b=[]
        for j in range(0,10):
            prob_random = random.uniform(0.0,1.0)
            if(prob_random>0.5):
                b += [1]
            else:
                b += [0]
        b_samples.append(b)
    alpha_powers = calc_alpha_powers(alpha, b_samples)
    denominator = sum(alpha_powers) * c
    numerator = 0
    for i in range(0,size):
        if b_samples[i][q]:
            numerator += alpha_powers[i]
```

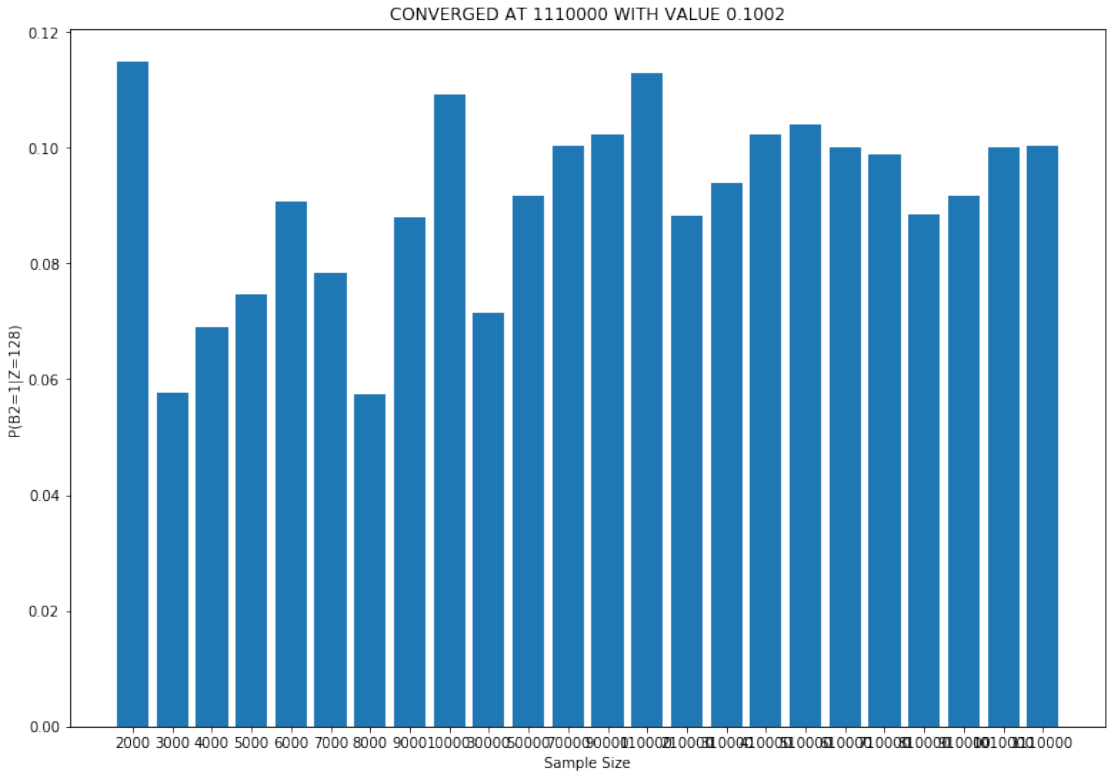
```
            numerator = numerator * c
            return numerator/denominator

        def get_step(sample_size):
            if sample_size<10000:
                return 1000
            if sample_size<100000:
                return 20000
            return 100000

        def converge(i):
            start_sample = 1000
            prev = -1.0
            curr = round(do_likelihood_weighting(start_sample,i,128),4)
            sample_size=[]
            query_prob=[]
            while abs(curr-prev)>0.0001:
                start_sample += get_step(start_sample)
                prev = curr
                curr = round(do_likelihood_weighting(start_sample,i,128),4)
                sample_size = sample_size + [start_sample]
                query_prob = query_prob + [curr]
            print "CONVERGED AT SAMPLE SIZE ", start_sample," WITH VALUE ",curr
            converge_dict = dict(zip(sample_size, query_prob))
            ordered_c = collections.OrderedDict(sorted(converge_dict.items()))
            plt.figure(figsize=(13, 9))
            plt.xticks(range(len(ordered_c)), ordered_c.keys())
            plt.bar(range(len(ordered_c)), ordered_c.values(), align='center')
            plt.xlabel("Sample Size")
            plt.ylabel("P(B{}=1|Z=128)".format(i+1))
            plt.title("CONVERGED AT {} WITH VALUE {}".format(start_sample,curr))
            plt.show()
```

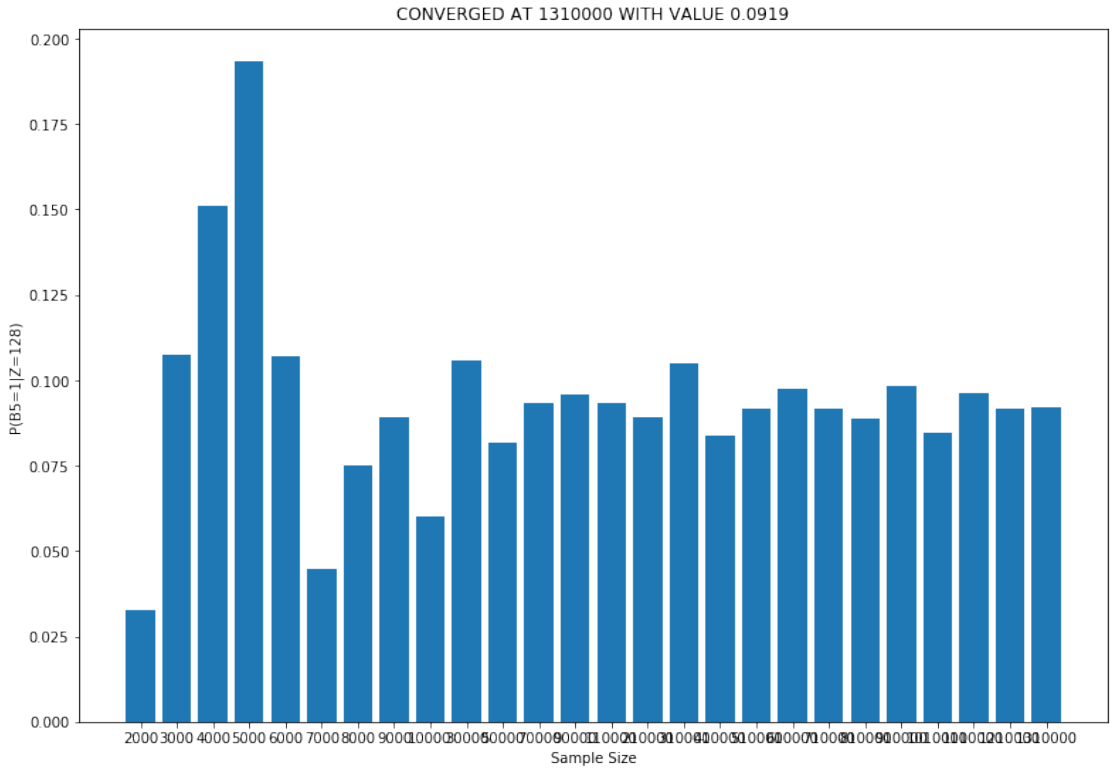
```
In [158]: converge(1)
```

```
CONVERGED AT SAMPLE SIZE  1110000  WITH VALUE  0.1002
```



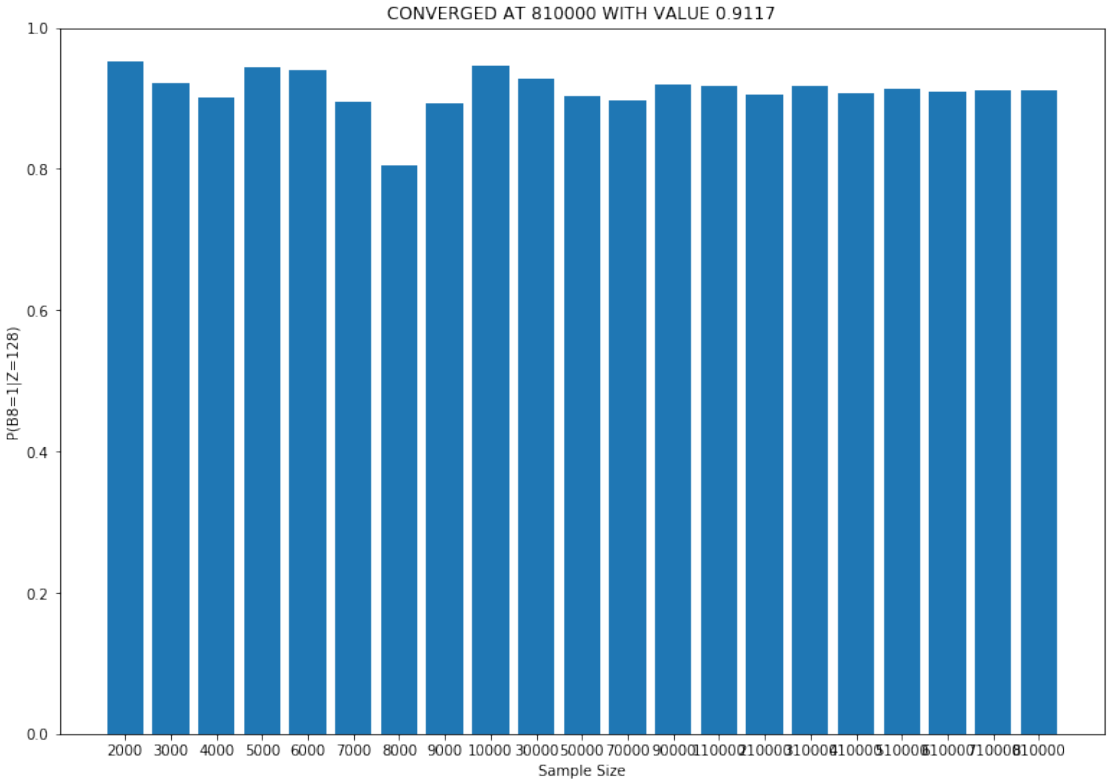
In [159]: converge(4)

```
CONVERGED AT SAMPLE SIZE  1310000  WITH VALUE  0.0919
```



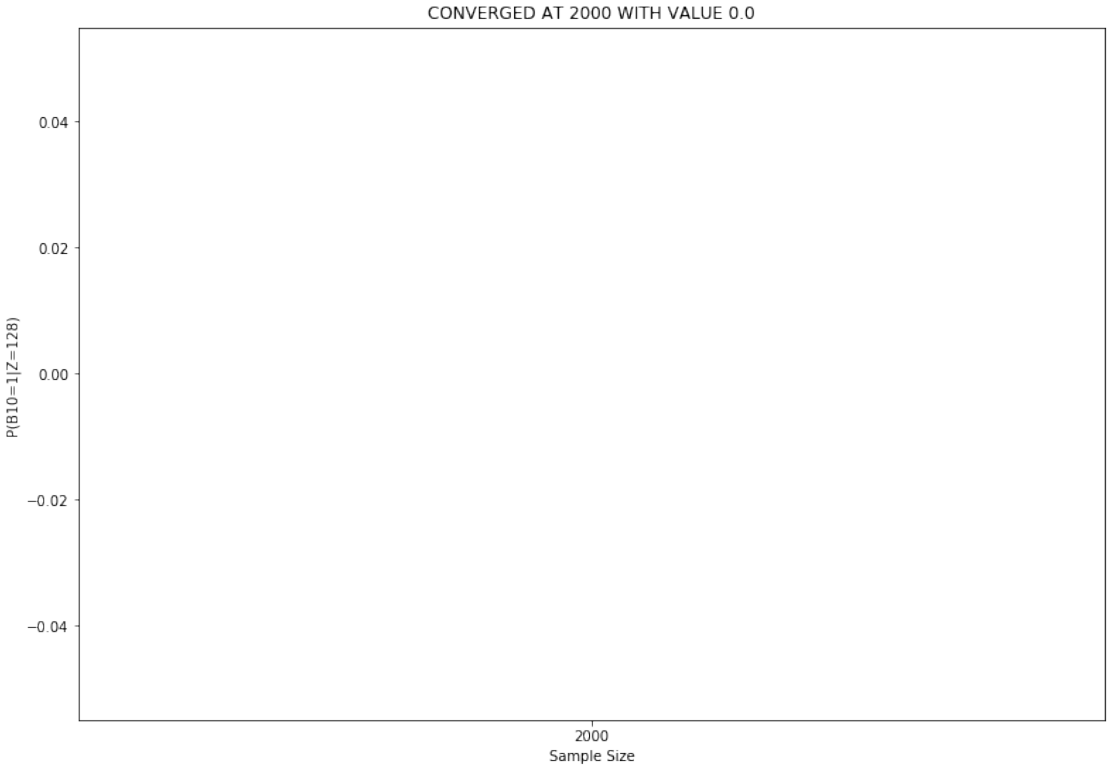
In [161]: converge(7)

```
CONVERGED AT SAMPLE SIZE  810000  WITH VALUE  0.9117
```



In [176]: converge(9)

```
CONVERGED AT SAMPLE SIZE  2000  WITH VALUE  0.0
```


CONVERGED AT 2000 WITH VALUE 0.0
0.04
0.02
0.00
−0.02
−0.04
P(B10=1|Z=128)
2000
Sample Size