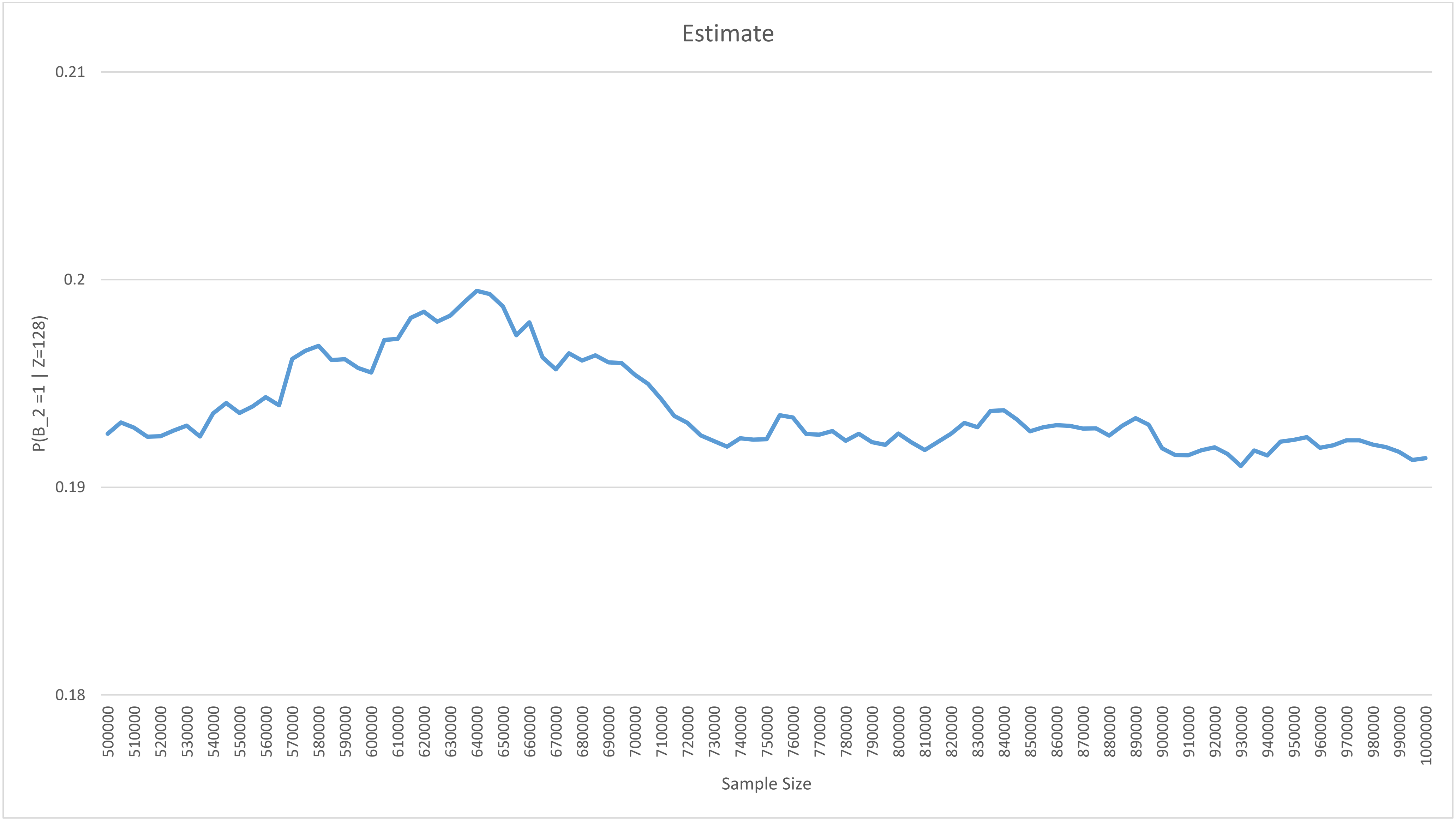
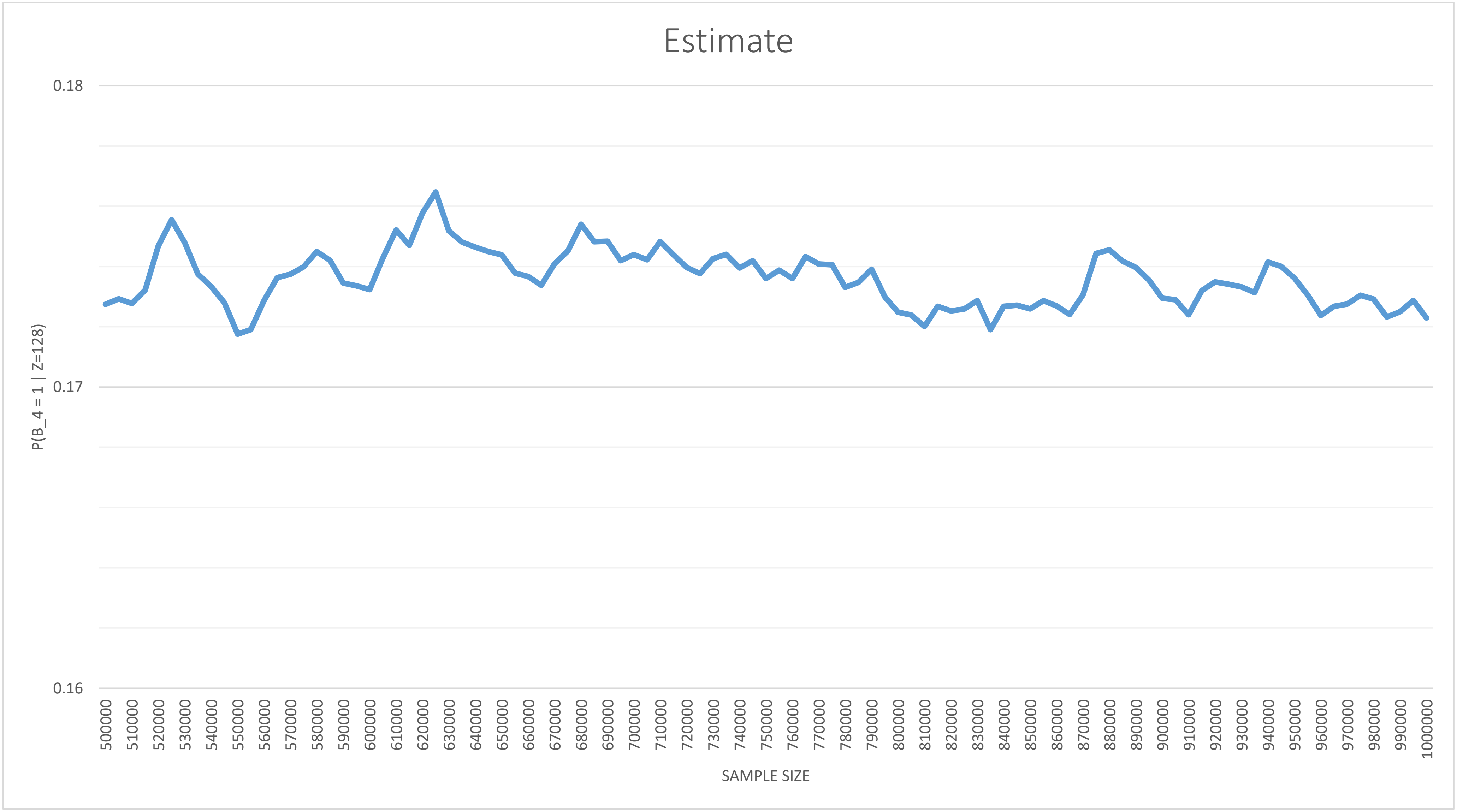
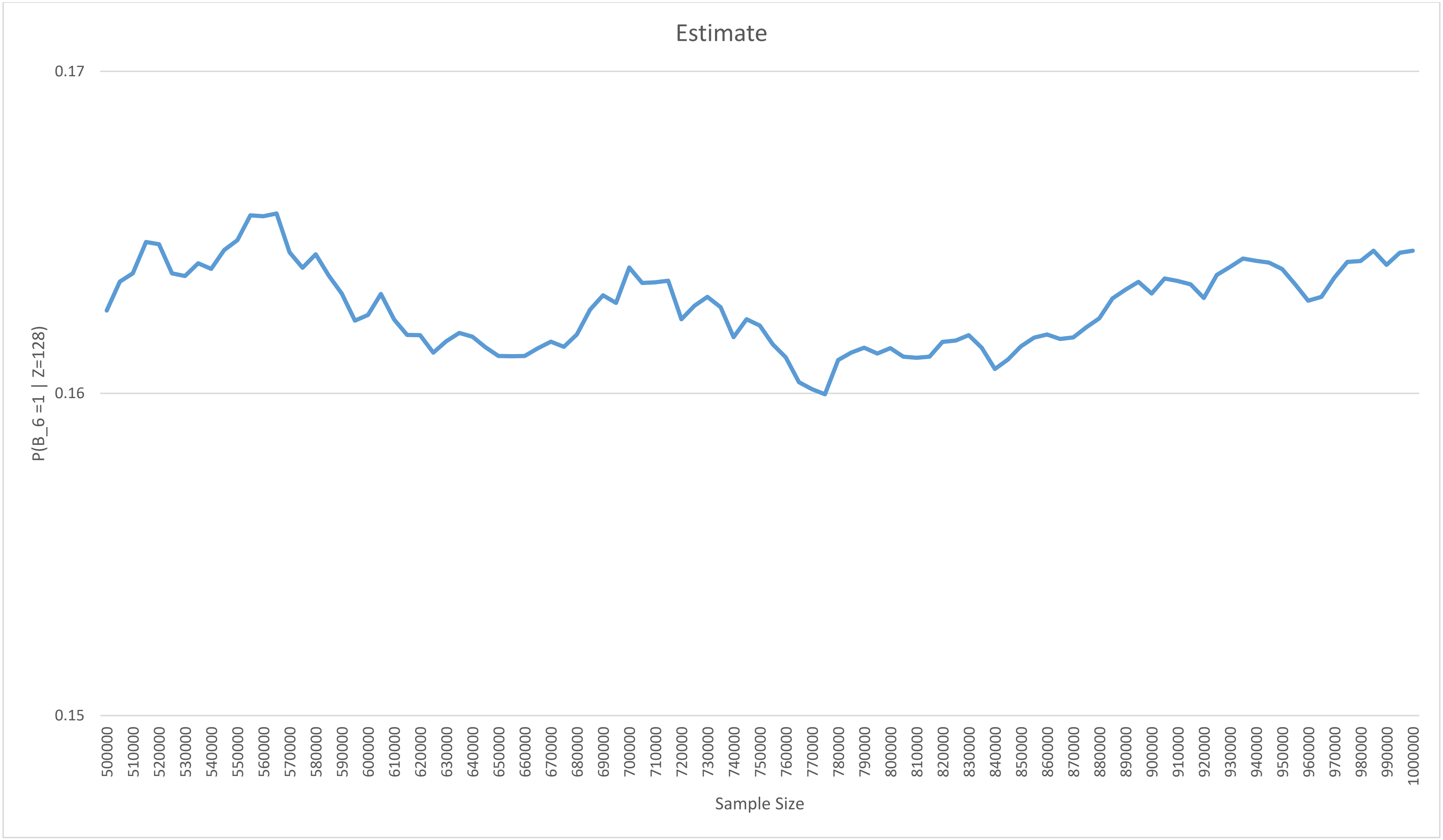
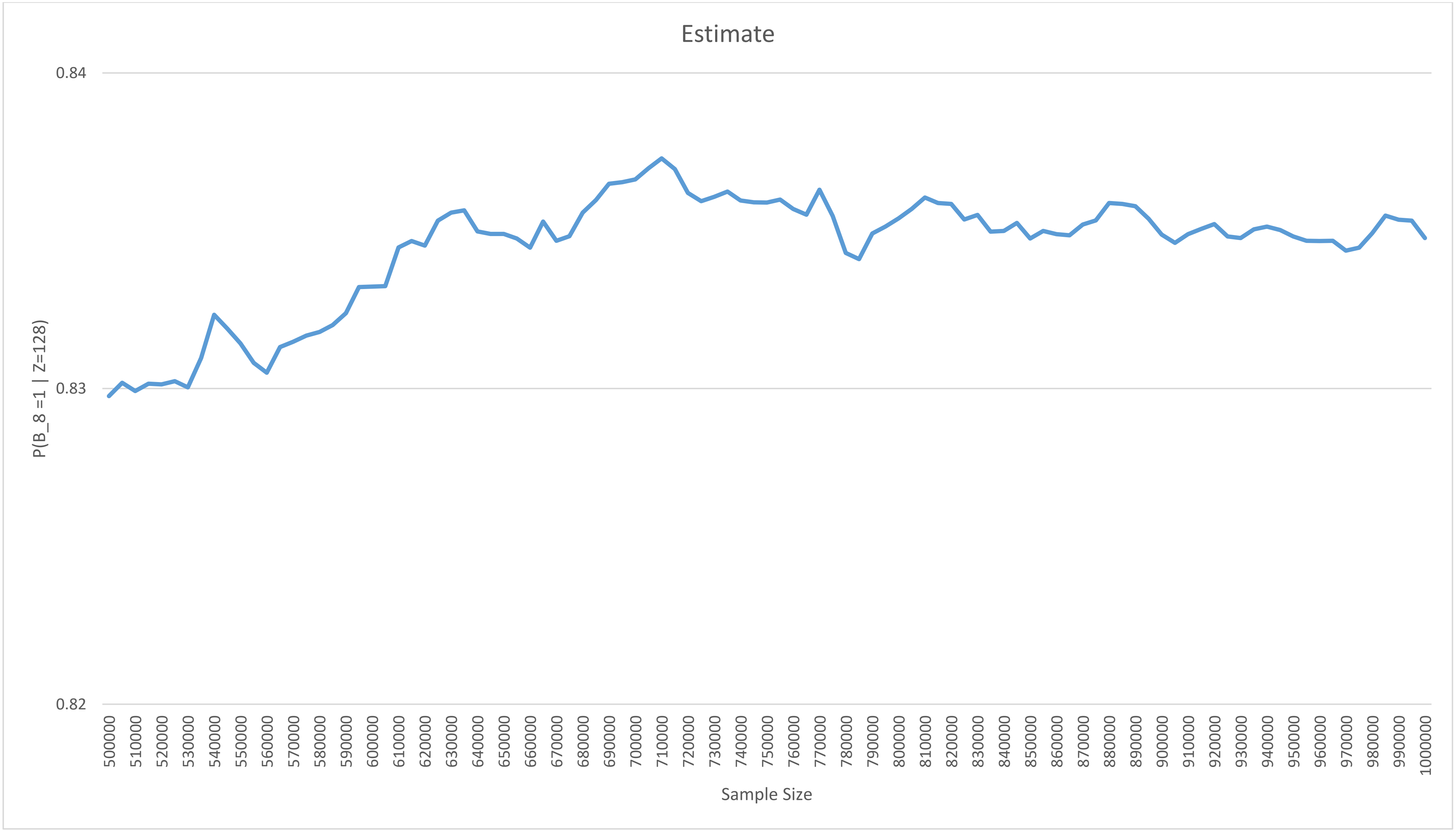
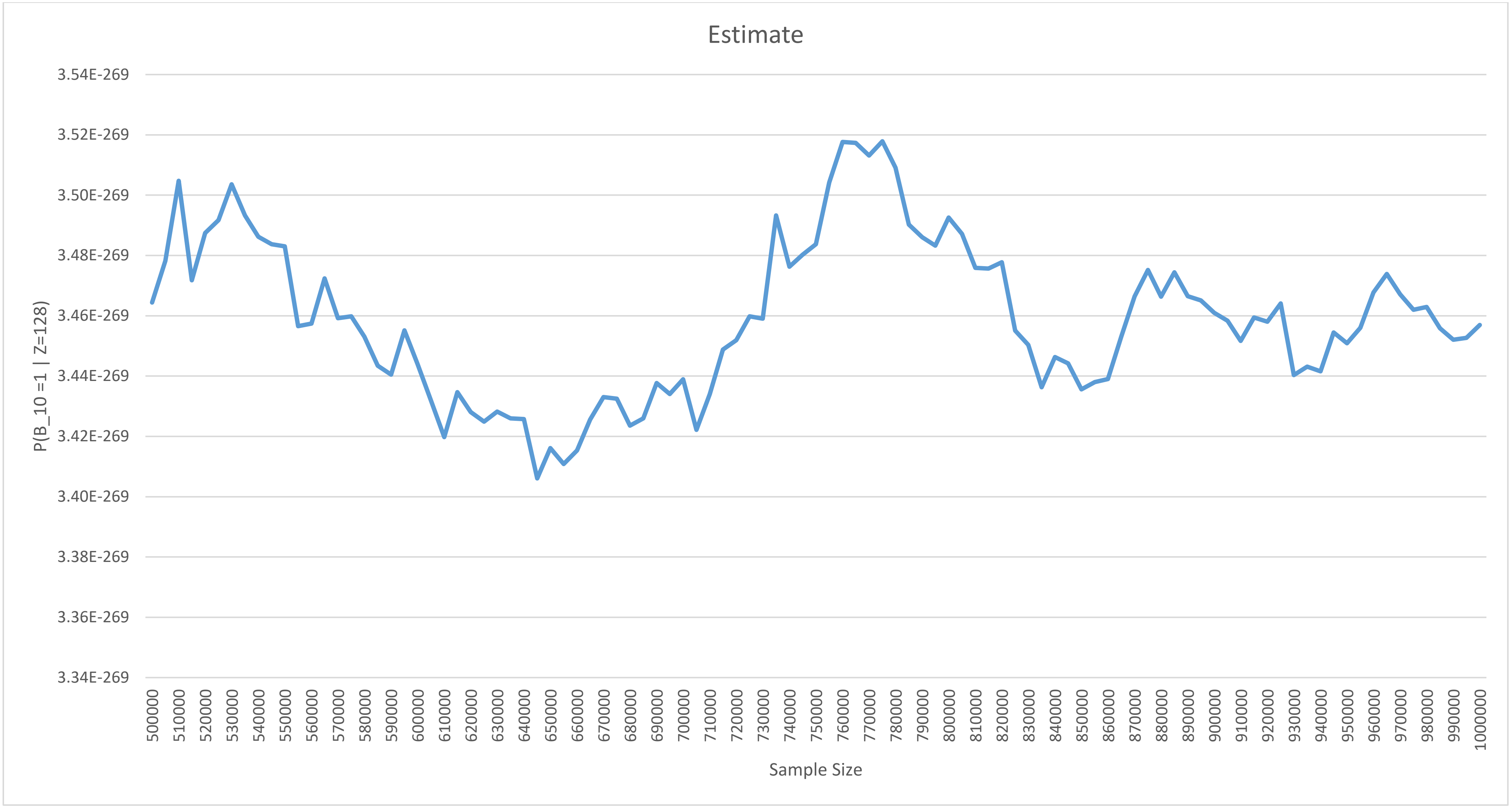
```python
"""
@author: gopal
"""

import numpy as np

def fB(B):
    n = len(B)
    b=0;
    for i in range(n):
        if(B[i] == 1):
            b = b + np.power(2,i)
    return b

#Z = constant, B = Matrix, alpha = noise
def PZgivenB(Z, B, alpha):
    ret = [(np.power(alpha, np.abs(Z-fB(B[i])))) * (1-alpha)/(1+alpha) for i in range(len(B))];
    return ret

#calculate the estimate for B i_val given sample size and B i's.
def calc(B, PZ_B, i_val):
    n = len(PZ_B)
    numerator = sum(PZ_B[i] for i in range(n) if B[i][i_val]==1)
    denominator = sum(PZ_B[i] for i in range(n))
    return numerator/denominator

def plotDict(d):
    lists = sorted(d.items()) # sorted by key, return a list of tuples
    x, y = zip(*lists) # unpack a list of pairs into two tuples
    for i in range(len(x)):
        print (x[i], '\t', y[i])

#generate big random observation data and iteratively add them to sample data.
def run(rows, Z_val, i_val, bits_val, alpha, begin_rows, increment, epsilon):
    sample_B = np.random.randint(2, size=(rows, bits_val))
    PZ_B = np.array(PZgivenB(Z_val, sample_B, alpha))
    curRows = begin_rows
    PB_Z = dict()
    PB_Z[begin_rows- increment] = 0
    PB_Z[begin_rows- 2*increment] = 0
    delta=1000
    while (curRows <= rows and delta > epsilon):
        subSample_B = sample_B[:curRows, :]
        subPZ_B = PZ_B[:curRows]
        PB_Z[curRows] = calc(subSample_B, subPZ_B, i_val)
        delta = np.abs(PB_Z[curRows] - PB_Z[curRows-increment]) + np.abs(PB_Z[curRows] - PB_Z[curRows-2*increment])
        curRows = curRows + increment
    plotDict(PB_Z)

#all indexes are zero reference. So from the Assignment question, if i_val is 2, put 1 here.
run(rows=1000000,Z_val=128, i_val=9,bits_val=10,alpha=0.2, begin_rows=500000, increment = 5000, epsilon = 0.001)
```


Estimate
0.21
0.2
0.19
0.18
P(B_2 =1 | Z=128)
500000
510000
520000
530000
540000
550000
560000
570000
580000
590000
600000
610000
620000
630000
640000
650000
660000
670000
680000
690000
700000
710000
720000
730000
740000
750000
760000
770000
780000
790000
800000
810000
820000
830000
840000
850000
860000
870000
880000
890000
900000
910000
920000
930000
940000
950000
960000
970000
980000
990000
1000000
Sample Size


Estimate
0.18
0.17
0.16
P(B_4 = 1 | Z=128)
SAMPLE SIZE
500000
510000
520000
530000
540000
550000
560000
570000
580000
590000
600000
610000
620000
630000
640000
650000
660000
670000
680000
690000
700000
710000
720000
730000
740000
750000
760000
770000
780000
790000
800000
810000
820000
830000
840000
850000
860000
870000
880000
890000
900000
910000
920000
930000
940000
950000
960000
970000
980000
990000
1000000


Estimate
0.17
0.16
0.15
P(B_6 =1 | Z=128)
500000
510000
520000
530000
540000
550000
560000
570000
580000
590000
600000
610000
620000
630000
640000
650000
660000
670000
680000
690000
700000
710000
720000
730000
740000
750000
760000
770000
780000
790000
800000
810000
820000
830000
840000
850000
860000
870000
880000
890000
900000
910000
920000
930000
940000
950000
960000
970000
980000
990000
1000000
Sample Size


Estimate
0.84
0.83
0.82
P(B_8 =1 | Z=128)
500000
510000
520000
530000
540000
550000
560000
570000
580000
590000
600000
610000
620000
630000
640000
650000
660000
670000
680000
690000
700000
710000
720000
730000
740000
750000
760000
770000
780000
790000
800000
810000
820000
830000
840000
850000
860000
870000
880000
890000
900000
910000
920000
930000
940000
950000
960000
970000
980000
990000
1000000
Sample Size


Estimate
3.54E-269
3.52E-269
3.50E-269
3.48E-269
3.46E-269
3.44E-269
3.42E-269
3.40E-269
3.38E-269
3.36E-269
3.34E-269
P(B_10 =1 | Z=128)
500000
510000
520000
530000
540000
550000
560000
570000
580000
590000
600000
610000
620000
630000
640000
650000
660000
670000
680000
690000
700000
710000
720000
730000
740000
750000
760000
770000
780000
790000
800000
810000
820000
830000
840000
850000
860000
870000
880000
890000
900000
910000
920000
930000
940000
950000
960000
970000
980000
990000
1000000
Sample Size