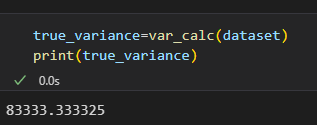
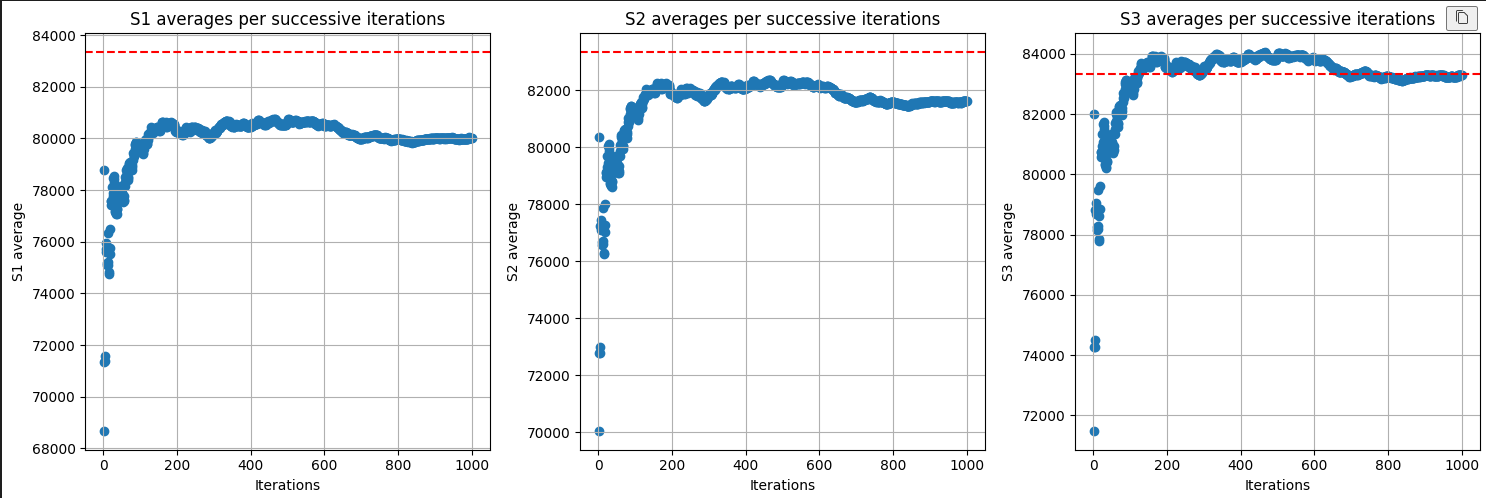
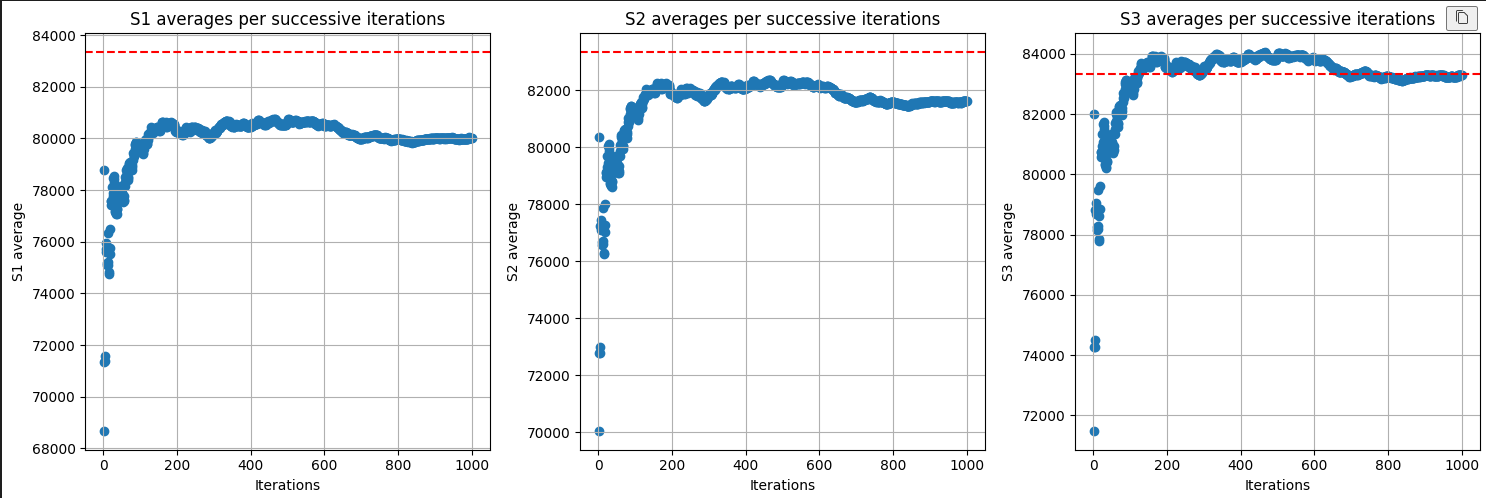
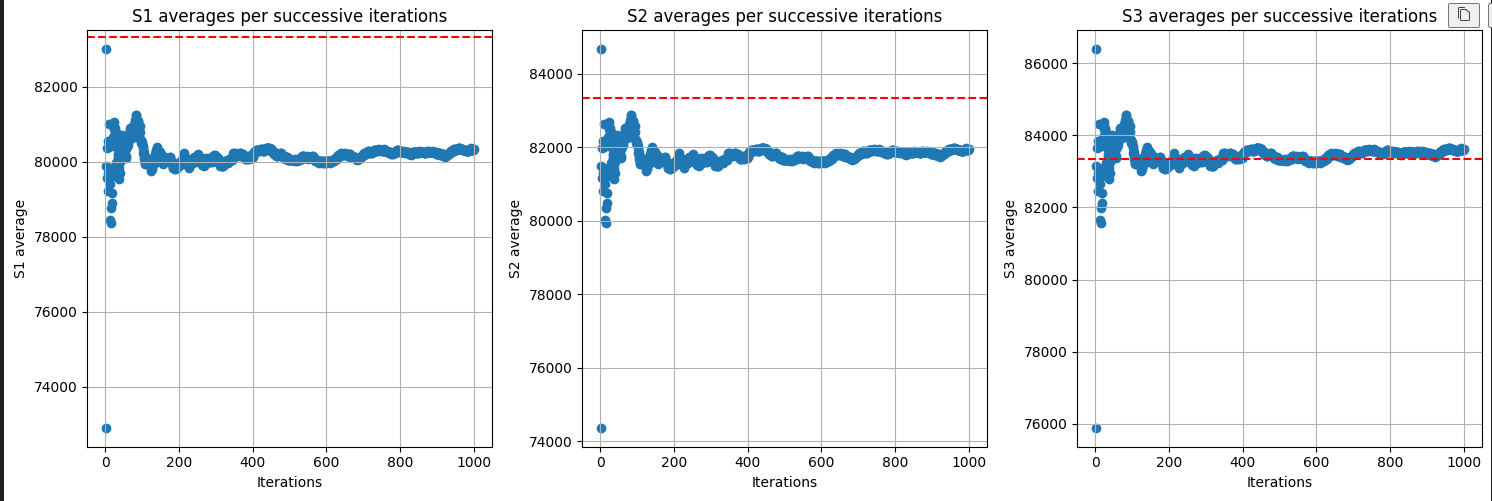
**Assignment-1 DSC**

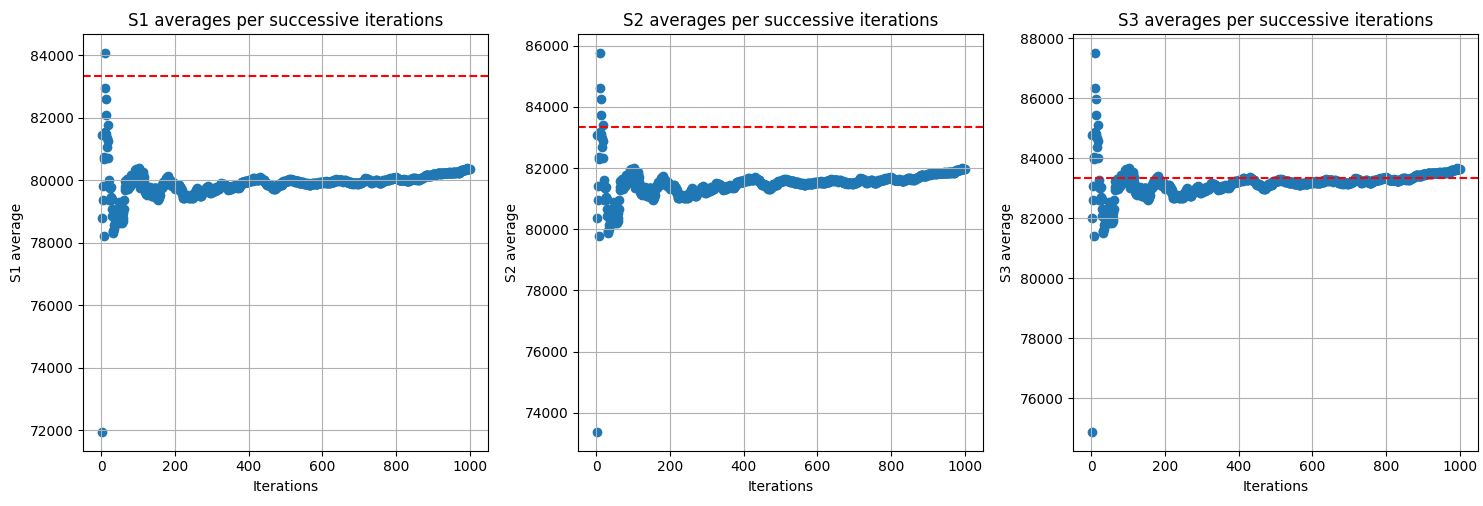
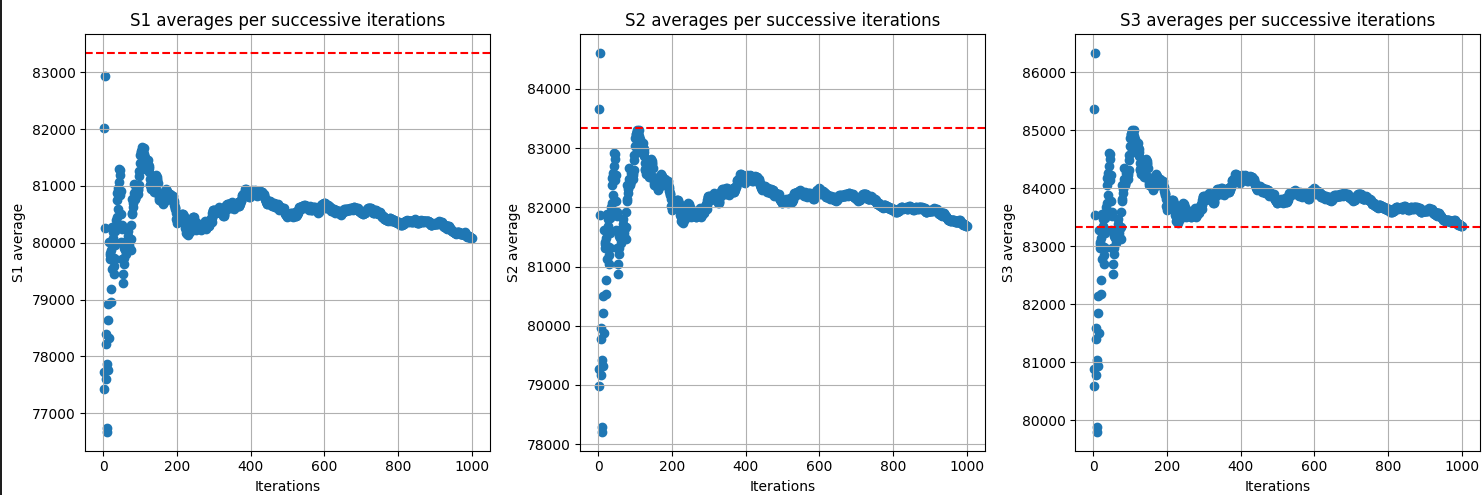
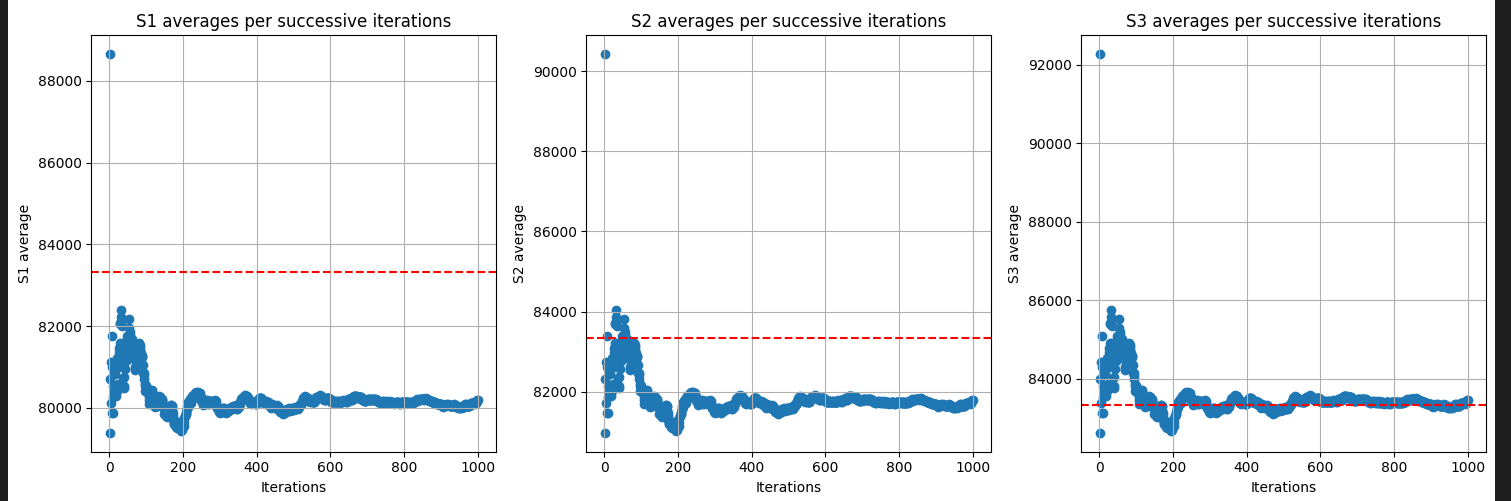
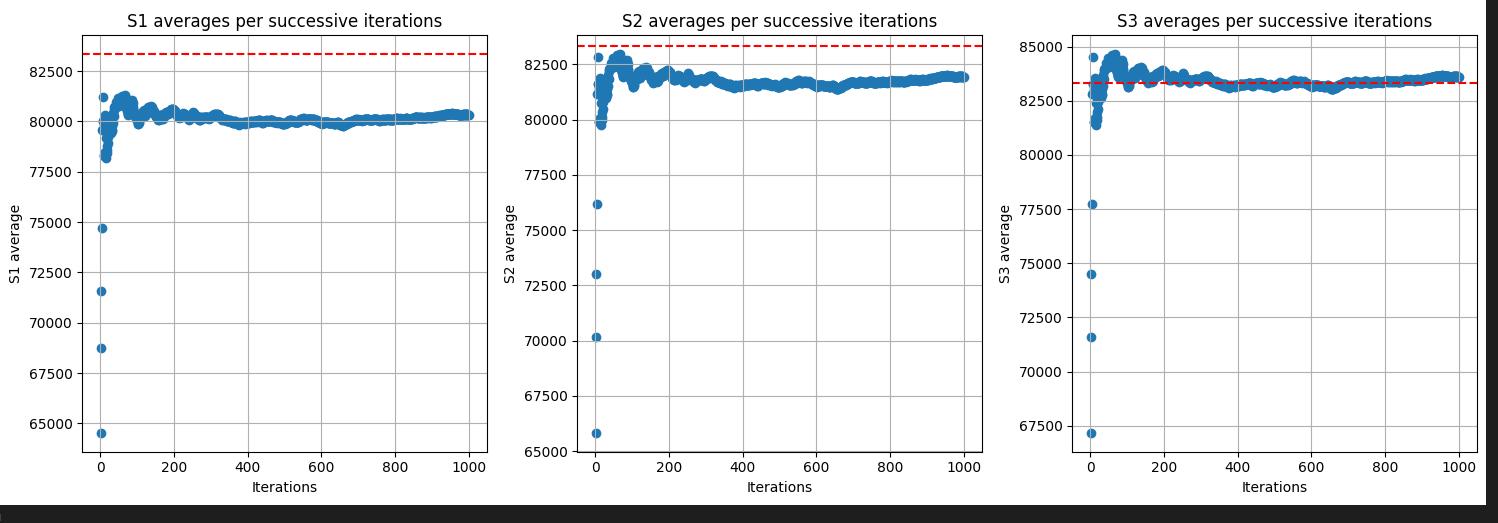
**Q2.a)**

****We got true variance as follows :

**Q2.d)**

****We obtained the following scatter graphs for s1, s2, s3.

**Q2.e)**

****It is quite clearly visible from the graphs that s3 reaches the true variance much more quickly than s2 which in turn is faster than s1. We also notice that s3 reaches much more near to true variance than s2 which also reaches near to true variance much more than s1.

s3 reaches near to true variance much faster because it is the unbiased variance unlike s2 and s1. It is unbiased because it accounts for degree of freedom being n-1 instead of n for the sample drawn from the dataset.

**Q3.a)**

Probability of seeing number √k on face = 1/k (unbiased die)

Let X = Geometric random variable equal to the number of times we need to roll the die to see √k

PMF(X) = (1-p)^(x-1) \* p where x = no of iterations required to see √k

and p = 1/k

Expected number of times we need to roll for √k : E[X] = 1/p

= 1/(1/k) = k

We ned to roll die k times to expect to see √k.

**Q3.b)**

Probability to roll any face = 1/k (unbiased die)

Probability to roll unique ith face = [k-(i-1)]/k

Let variable Xi represent geometric variable for number of times we need to roll die for seeing ith unique number on die

PMF(Xi) = (1-p)^(xi-1) \* p where xi = no of iterations required to see √k

And p = [k-(i-1)]/k

Expected number of times we need to roll for ith unique number : E[X] = 1/p

= 1/([k-(i-1)]/k)

= k/(k+1-i)

Total Expected number of times we need to roll die to see all numbers = E[X1]+ E[X2]….+E[Xk]

=

= k \* log(k)

**Q3.c)**

Let variable Xi represent geometric variable for number of times we need to roll die for seeing ith unique face on die

For X1, p = ¼ + ¼ + ½ =1

Hence, E[X1] = 1 (E[X]=1/p)

i.e. It will take only 1 roll to get our first unique face.

For X2, cases:

* 1st unique number was 1 or 3 :

p for X2 case1 = 1- ¼ = ¾

Hence, E[X2 case1] = 4/3 (E[X]=1/p)

For X3, cases again:

* + 2nd unique number was 1 or 3 :

p for X3 case1 = 1- ¼ - ¼ = ½

Hence, E[X3 case1] = 2 (E[X]=1/p)

* + 2nd unique number was 2 :

p for X3 case2 = 1- ¼ - ½ = ¼

Hence, E[X3 case2] = 4 (E[X]=1/p)

* 1st unique number was 2 :

p for X2 case2 = 1- ½ = ½

Hence, E[X2 case2] = 2 (E[X]=1/p)

2nd unique number was 1 or 3 :

p for X3 case3 = 1- ½ - ¼ = ¼

Hence, E[X3 case3] = 4 (E[X]=1/p)

E[X2] = P(X2 case1)\*E[X2 case1] + P(X2 case2)\*E[X2 case2]

= P(1st draw = 1 or 3)\*E[X2 case1] + P(1st draw = 2)\*E[X2 case2]

= ½ \* 4/3 + ½ \* 2

= 5/3

E[X3] = P(X2 case1)\*(P(X3 case1)\*E[X3 case1] + P(X3 case2)\*E[X3 case2]) + P(X2 case2)\*P(X3 case3)\*E[X3 case3]

= ½ \* ( ¼ /( ¼ + ½ )\*2 + ½ /( ¼ + ½ )\*4 ) + ½ \* 1 \* 4

= ½ \* 10/3 + 2

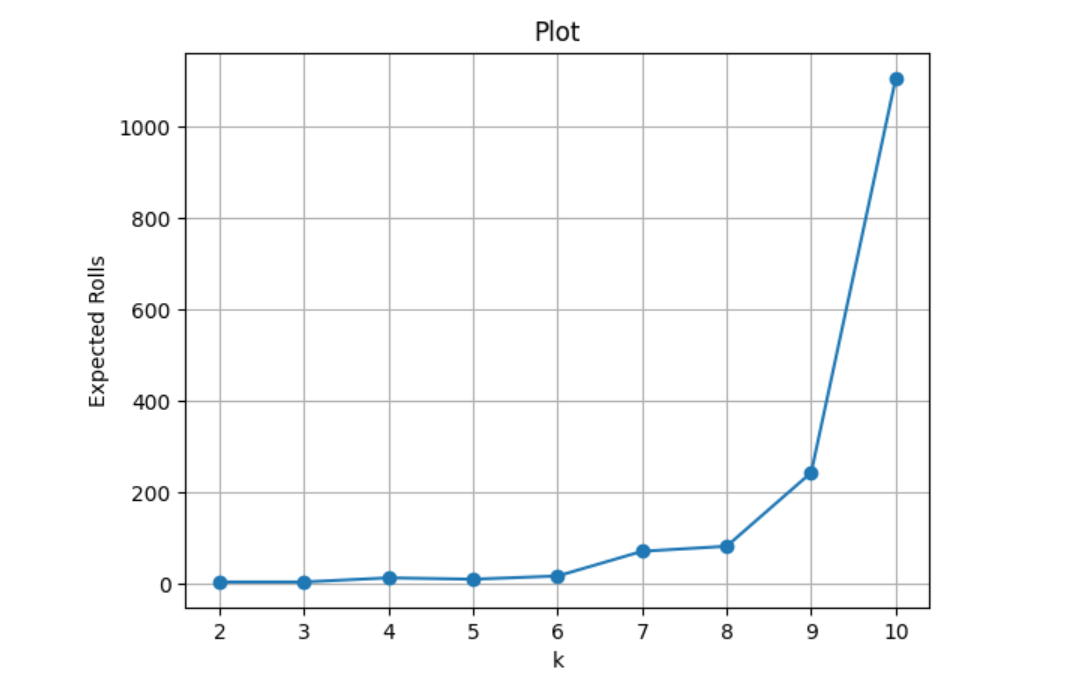
= 11/3

Total expected rolls required = E[X1]+E[X2]+ E[X3]

= 1+5/3+11/3

= 19/3

**Q3.d)**

We got the following graph for number of rolls required for increasing k :

This plot matches with the total expected rolls values found in (c) part as the number of rolls required are nearly 5-6.