

# 1 Question 1

First Notice,

$$\begin{aligned} S_n^2 &= \left( \frac{1}{n} \sum_{k=1}^n x_k^2 \right) - \bar{x}_n^2 \\ n S_n^2 &= \sum_{k=1}^n x_k^2 - n \bar{x}_n^2 \\ \sum_{k=1}^n x_k^2 &= n S_n^2 + n \bar{x}_n^2 \end{aligned}$$

So we can then see,

$$\begin{aligned} S_{n+1}^2 &= \left( \frac{1}{n+1} \sum_{k=1}^{n+1} x_k^2 \right) - \bar{x}_{n+1}^2 \\ &= \frac{1}{n+1} \left[ \sum_{k=1}^n x_k^2 + x_{n+1}^2 \right] - \bar{x}_{n+1}^2 \\ &= \frac{1}{n+1} \left[ n S_n^2 + n \bar{x}_n^2 + x_{n+1}^2 \right] - \bar{x}_{n+1}^2 \\ &= \frac{n S_n^2 + n \bar{x}_n^2 + x_{n+1}^2}{n+1} - \bar{x}_{n+1}^2 \\ &= \frac{n (S_n^2 + \bar{x}_n^2) + x_{n+1}^2}{n+1} - \bar{x}_{n+1}^2 \end{aligned}$$

Hence an equation for  $S_{n+1}^2$  using only  $(n, S_n^2, \bar{x}_n, \bar{x}_{n+1}, x_{n+1})$

## 2 Question 2

```
lcMean :: [Float] -> Float
lcMean [] = undefined
lcMean [x] = x
lcMean (x:xs) = (x + n * average) / (n + 1)
  where
    average = lcMean xs
    n       = fromIntegral (length xs)

lcVariance :: [Float] -> Float
lcVariance [] = undefined
lcVariance [x] = 0
lcVariance (x:xs) = (((n * s') + (n * (avg ** 2)) + (x ** 2)) / (n + 1)) - (avg' ** 2)
  where
    avg = lcMean xs
    avg' = lcMean (x:xs)
    s' = lcVariance xs
    n = fromIntegral (length xs)
```

## 3 Question 3

```
trMean :: [Float] -> Float
trMean [] = undefined
trMean xs = htrMean 0 0 xs

htrMean :: Float -> Float -> [Float] -> Float
htrMean _ avg [] = avg
htrMean n avg (x:xs) = htrMean n' avg' xs
  where
    n' = n + 1
    avg' = (x + (avg * n)) / (n + 1)

trVariance :: Float -> Float -> Float -> [Float] -> Float
trVariance _ _ s [] = s
trVariance n avg s (x:xs) = trVariance n' avg' s' xs
  where
    n' = n + 1
    avg' = htrMean n avg [x]
    s' = ((n * (s + (avg ** 2)) + (x ** 2)) / (n + 1)) - (avg' ** 2)
```

## 4 Question 4

```
variance :: [Float] -> Float
variance [] = undefined
variance xs = trVariance 0 0 0 xs
```

## 5 Question 5

### Iteration invariant:

where  $ys$  is everything processed already and  $n = \text{length } ys$

$$\text{trVariance } n \ \bar{x} \ S_n^2 \ (x : xs) = (n + 1) \ \overline{x : ys} \ S_{n+1}^2 \ xs$$

## 6 Question 6

### Iteration invariant proof:

$$\begin{aligned} \text{trVariance } n \ \bar{x} \ S_n^2 \ \emptyset &= s \\ &= \text{variance } ys \end{aligned}$$

Meaning that we have the variance of the whole list that has now been processed.

$$\begin{aligned} \text{trVariance } n \ \bar{x} \ S_n^2 \ (x : xs) &= \text{trVariance } (n + 1) \ (\overline{x : ys}) \left( \frac{n \times (S_n^2 + \bar{x}^2) + x^2}{n + 1} - \overline{x : ys^2} \right) xs \\ &= \text{trVariance } (n + 1) \ \overline{x : ys} \left( \frac{\frac{n \times \sum_{k=1}^n ys_k^2}{n} + (x^2)}{n + 1} - \overline{x : ys^2} \right) xs \\ &= \text{trVariance } (n + 1) \ \overline{x : ys} \left( \frac{\sum_{k=1}^n ys_k^2 + x^2}{n + 1} - \overline{x : ys^2} \right) xs \\ &= \text{trVariance } (n + 1) \ \overline{x : ys} \left( \frac{\sum_{k=1}^{n+1} (x : ys)_k^2}{n + 1} - \overline{x : ys^2} \right) xs \\ &= \text{trVariance } (n + 1) \ \overline{x : ys} \left( \left( \frac{1}{n + 1} \times \sum_{k=1}^{n+1} (x : ys)_k^2 \right) - \overline{x : ys^2} \right) xs \\ &= \text{trVariance } (n + 1) \ \overline{x : ys} \ S_{n+1}^2 \ xs \end{aligned}$$

Meaning that we have the variance of  $(ys:x)$  and will continue to process with the rest of the list that hasn't been processed yet.

$$\begin{aligned} \text{trVariance } 0 \ 0 \ 0 \ (x : xs) &= \text{trVariance } (0 + 1) \ \bar{x} \left( \frac{0 \times (0 + 0^2) + x^2}{1} - \bar{x}^2 \right) xs \\ &= \text{trVariance } 1 \ \bar{x} \left( \frac{x^2}{1} - \bar{x}^2 \right) xs \\ &= \text{trVariance } 1 \ x \ 0 \ xs \\ &= \text{trVariance } 1 \ \bar{x} \ (\text{variance } x) \ xs \end{aligned}$$

Meaning that the first call will always return a variance of 0 which is correct for the variance of 1 value.

And thus we have our formula for variance to satisfy our iteration invariant for a base case, a recursive case, and our final (edge) case

## 7 Question 7

### Bound Value:

Our bound value would be the length of our input list xs

## 8 Question 8

### Decreasing and positivity of bound value:

Each time we take an element out of xs and process it so each iteration shortens the length of the list. The length will always be positive because our function can never be called on an empty list, and once we run out of elements in the list we return and finish. Thus we have a value which is always positive and decreases every iteration down to zero.

## 9 Question 9

### Quick Checks:

```
-- Var(X) > 0
property1 :: IO()
property1 = quickCheck $ \xs -> null xs || variance xs >= 0

-- Var(X + c) = Var(X), where c is some constant.
property2 :: IO()
property2 = quickCheck $ \a xs -> null xs || diff < 10**(-2)
  where
    diff      = abs (variance xsplusa - variance xs)
    xsplusa   = map (+ a) xs

-- Var(aX) = a^2 * Var(X), where a is some constant.
property3 :: IO()
property3 = quickCheck $ \a xs -> null xs || diff < 10**(-2)
  where
    diff          = abs (atimesxsvar - asquaredtimesvar)
    atimesxsvar    = variance (map (a *) xs)
    asquaredtimesvar = (a ** 2) * variance xs

-- Var(aX + b) = a^2 * Var(X), the result of the previous two properties combined.
property4 :: IO()
property4 = quickCheck $ \a b xs -> null xs || diff < 10**(-2)
  where
    diff          = abs (atimesxsplusbvar - asquaredtimesvar)
    atimesxsplusbvar = variance map (\x -> a * x + b) xs
    asquaredtimesvar = (a ** 2) * variance xs
```

These are based on the properties of population variance.

The first property of variance is that variance will always be a positive number.

The second property of variance is that if you add some constant to every value in the list, the variance does not change.

The third property of variance is that if you multiply some constant with every value in the list, the variance is the same as if you didn't do that multiplication and then multiplied the result by the constant squared.

The fourth and final property is that the second and third property can be combined.

## 10 Question 10

**Base Case,  $n = \text{Zero}$**   
asldkfjh