

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

```
1 lcMean :: [Float] -> Float
2 lcMean [] = undefined
3 lcMean [x] = x
4 lcMean (x:xs) = (x + n * average) / (n + 1)
5   where
6     average = lcMean xs
7     n       = fromIntegral (length xs)
8
9 lcVariance :: [Float] -> Float
10 lcVariance [] = undefined
11 lcVariance [x] = 0
12 lcVariance (x:xs) = (((n * s') + (n * (avg ** 2)) + (x ** 2)) / (n + 1)) - (avg' ** 2)
13   where
14     avg = lcMean xs
15     avg' = lcMean (x:xs)
16     s' = lcVariance xs
17     n = fromIntegral (length xs)
```

3 Question 3

```
1 trMean :: [Float] -> Float
2 trMean [] = undefined
3 trMean xs = htrMean 0 0 xs
4
5 htrMean :: Float -> Float -> [Float] -> Float
6 htrMean _ avg [] = avg
7 htrMean n avg (x:xs) = htrMean n' avg' xs
8   where
9     n' = n + 1
10    avg' = (x + (avg * n)) / (n + 1)
11
12 trVariance :: Float -> Float -> Float -> [Float] -> Float
13 trVariance _ _ s [] = s
14 trVariance n avg s (x:xs) = trVariance n' avg' s' xs
15   where
16     n' = n + 1
17     avg' = htrMean n avg [x]
18     s' = ((n * (s + (avg ** 2)) + (x ** 2)) / (n + 1)) - (avg' ** 2)
```

4 Question 4

```
1 variance :: [Float] -> Float
2 variance [] = undefined
3 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.

—

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

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:

```
1  -- Var(X) > 0
2  property1 :: IO()
3  property1 = quickCheck $ \xs -> null xs || variance xs >= 0
4
5  -- Var(X + c) = Var(X), where c is some constant.
6  property2 :: IO()
7  property2 = quickCheck $ \a xs -> null xs || diff < 10**(-2)
8      where
9          diff      = abs (variance xsplusa - variance xs)
10         xsplusa   = map (+ a) xs
11
12  -- Var(aX) = a^2 * Var(X), where a is some constant.
13  property3 :: IO()
14  property3 = quickCheck $ \a xs -> null xs || diff < 10**(-2)
15      where
16          diff      = abs (atimesxsvar - asquaredtimesvar)
17          atimesxsvar = variance (map (a *) xs)
18          asquaredtimesvar = (a ** 2) * variance xs
19
20  -- Var(aX + b) = a^2 * Var(X), the result of the previous two properties combined.
21  property4 :: IO()
22  property4 = quickCheck $ \a b xs -> null xs || diff < 10**(-2)
23      where
24          diff      = abs (atimesxsplusbvar - asquaredtimesvar)
25          atimesxsplusbvar = variance map (\x -> a * x + b) xs
26          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 final property is that the second and third property can be combined.

10 Question 10

Given that, $\text{Plus } m \text{ Zero} = m$ Using PMI, prove that,
 $\text{Plus Zero } n = n$

10.1 Base Case, $n = \text{Zero}$

$$\text{Plus Zero } n \implies \text{Plus Zero Zero}$$

$$\text{Plus Zero Zero} = \text{Zero}$$

This is true as $0 + 0 = 0$ under the naturals.

10.2 Inductive Hypothesis, $n = k$

Assume that $\text{Plus Zero } k = k$

10.3 Inductive Step

for our $k + 1$ case, we are simply using $\text{Succ } k$, to get

$$\text{Plus Zero } (\text{Succ } k) = \text{Succ } (\text{Plus Zero } k)$$

By definition, and since we have assumed in our inductive hypothesis that $\text{Plus Zero } k = k$ we have

$$\text{Plus Zero } (\text{Succ } k) = \text{Succ } k$$

Or in other words,

$$0 + (k + 1) = k + 1$$

which is true under the natural numbers as well. Thus because of our base case being correct, and our inductive step being correct if we assume that our equation works for some k th step. By the Principle of Mathematical Induction, we know that $\text{Plus Zero } n = n, \forall n \in \mathbb{N}$

11 Question 11

Using PMI Prove the commutativity of Plus, that is,

Using PMI prove that, $\text{Plus } m \ n = \text{Plus } n \ m$

11.1 Base Case, $n = \text{Zero}$

LHS = $\text{Plus } m \ \text{Zero} = m$, By construction

RHS = $\text{Plus } \text{Zero } m = m$, From our conclusions in Q10

LHS = RHS

11.2 Inductive Hypothesis, $n = k$

Assume that,

$$\text{Plus } m \ k = \text{Plus } k \ m$$

11.3 Inductive Step

For our $k + 1$ case, we are using $\text{Succ } k$ to represent $k + 1$, if we then let

$$z = \text{Plus } m \ k = \text{Plus } k \ m \text{ (From our inductive hypothesis.)} \quad (1)$$

This gives us

LHS = $\text{Plus } m \ (\text{Succ } k)$

= $\text{Succ } (\text{Plus } m \ k)$, By definition

= $\text{Succ } z$, From (1)

RHS = $\text{Plus } (\text{Succ } k) \ m$

= $\text{Plus } k \ (\text{Succ } m)$, Assuming Plus is associative

= $\text{Succ } (\text{Plus } k \ m)$

= $\text{Succ } z$, From (1)

LHS = RHS, By our Inductive hypothesis

Since our LHS is equal to our RHS under our assumption in our inductive hypothesis. Our base case has been proven true. By the Principle of Mathematical Induction, we have proven that Plus is commutative when we iterate over n . We can see that the converse is also true with a symmetric proof.