**Computational Statistics**

**Exercise 3.3**: Errors in computations

1. Truncation errors occur when series expansion is used for approximation but only a finite number of terms is summed up, yet cancellation errors occur when an operation on two numbers increases relative error substantially more than it increases absolute error, for example in subtracting two nearly equal numbers
2. Relative precision of multiplication depends on exponents, their signs, and the first few digits in the significand. Because none of them are significantly affected by cancellation, we can say cancellation is not a problem in multiplication.

**Exercise 3.6**: Using recursion tree, we know , with the first term representing the leaves and the second representing other nodes. Therefore, assuming , we have

**Exercise 3.17**: Backward error analysis would be a good method to determine if is an adequate approximation to . By plugging problem into to get a computed answer , we could find what perturbation of the original problem would yield the computed answer exactly. As the solution is unknown, each step in the backward analysis involves numbers that could actually have participated in the computations that were performed.

**Exercise 3.21**:

Advantages:

1. NaN could serve as a place holder and prevent algorithm from breaking down due to lack of data input
2. Algorithm will be well-informed about where a data point is missing
3. NaN has a uniform representation under IEEE 756 as a floating-point number, which means it could be recorded and read across different platforms

Disadvantages:

1. Requires adoption from the data structure being used if all existing data types are numerical
2. Algorithms need to have additional function to identify NaN as compared to other type of data
3. Variables with richer contents could be used to indicate both “missing data” and “reason behind (nonresponse, discontinuation, etc.)”

**SAS Programming**

1. BenAndJerrys.dat
   1. **DATA** BandJ;

INFILE '\\Mac\Home\Desktop\BenAndJerrys.dat' DSD DLM=',';

INPUT name :$100. size cal fatcal fat satfat transfat

cholesterol sodium carbohyd fiber sugars protein

yintro yretir description :$100. notes :$100.;

IF yretir = **.** AND notes ^= 'Scoop Shop Exclusive';

**RUN**;

**PROC** **PRINT** DATA = BandJ;

* 1. **DATA** BandJb;

SET BandJ;

caltb = cal\***15**/size;

caltb = cal\***15**/size;

IF caltb ^= **.**;

**RUN**;

**PROC** **PRINT** DATA = BandJb;

* 1. **PROC** **PRINT** DATA = BandJb;

SUM caltb;

**RUN**;

* 1. **DATA** BandJc;

SET BandJ;

RETAIN calmax;

calmax = MAX(calmax, cal);

**RUN**;

**PROC** **PRINT** DATA = BandJc;

1. CurrentAge = INT (YRDIF(DoB, TODAY(), 'AGE'));

AgeNV = INT (YRDIF(DoB, DoNV, 'AGE'));

AgeAfterNV90 = INT (YRDIF(DoB, DoNV, 'AGE') + **90.0**/**365.0**);

1. Which function can be used to replace text?
   1. TRIM
   2. INDEX
   3. TRANWRD
   4. PROPCASE
2. Which of the following is a valid function for finding the average of X1, X2, and X3?
   1. AVERAGE(X1, X2, X3)
   2. AVG(X1, X2, X3)
   3. MEAN(X1, X2, X3)
   4. MU(X1, X2, X3)
3. Which set of IF-THEN/ELSE statements will run without errors?
   1. IF 0 <= Age <= 50 THEN Group = ’A’;

ELSE 50 < Age <= 70 THEN Group = ’B’;

ELSE Age > 70 THEN Group = ’C’;

* 1. IF0<=Age<=50THENGroup=’A’;

ELSE IF 50 < Age <=70 THEN Group = ’B’;

ELSE Age > 70 THEN Group = ’C’;

* 1. IF0<=Age<=50THENGroup=’A’;

ELSE IF 50 < Age <=70 THEN Group = ’B’;

ELSE IF Age > 70 THEN Group = ’C’;

* 1. All of the above will work.