Statistical Debugging

Question 1. Post-deployment software monitoring is an increasingly popular method that complements in-house (pre-deployment) software testing in order to improve software quality. State one advantage of post-deployment monitoring over in-house testing, and state one advantage of of in-house testing over post-deployment monitoring.

Answer: Post-deployment monitoring provides better coverage of real world use cases than in house testing, and allows bugs to be prioritized by the frequency users experience them. In-house testing catches bugs before the user experiences them, and is necessary to create software that is usable at launch. There are other reasonable answers.

Question 2. When prioritizing which predicates to examine in a multi-bug scenario, what is the problem with ranking predicates by their Increase score? Describe a ranking strategy that can be used to help mitigate this problem.

Answer: The problem is that the top of the list is often dominated by predicates with high increase scores but very few failing runs; they are sub-bug predictors, or special cases of more general bugs. The problem can be mitigated by ranking the predicates by the harmonic mean of the Increase(P) and F(P) score.

Question 3. Consider the following code fragment:

```
w = 2           /* The code fragment is run 500 times. */
if (x > 0)
   {y = foo(x)} /* 300 runs reach this line, and 60 fail on this line. */
else
   {y = bar(x)} /* 200 runs reach this line, and 120 fail on this line. */
z = 10           /* All 320 runs reaching this line succeed. */
```

Let P be the branch condition predicate (x > 0).

a. What is Increase(P)?

Answer: Failure(P) = 60 / 300 = 0.2, Context(P) = 180 / 500 = 0.36, Increase(P) = -0.16

b. What is Increase(¬P)?

Answer: Failure(¬P) = 120 / 200 = 0.6, Context(¬P) = 180 / 500 = 0.36, Increase(¬P) = 0.24