

## Where the Bugs Were

### a. logistic-regression function

The convergence test,

```
if max(abs(b - bLast)/(abs(bLast) + 0.01*tol)) < tol) break
```

was in the wrong place – before the coefficients were computed in the next iteration. Since both `b` and `bLast` start out with 0's, no iterations are executed, causing the program to fail when it tries to compute the covariance matrix of the coefficients – far from the site of the error.

Moving the convergence test to immediately after the computation of `b` fixes the problem.

### b. row-echelon-form function

The problem here is that (in two places) the function compares floating-point (“real”) numbers to 0; these tests can easily fail because floating-point computations are not exact.

I added a `tol` (tolerance) argument to the function, with a default value of `1.e-6`. Changing the test for a zero pivot to

```
if (abs(pivot) < tol) next
```

and the check for zero rows to

```
for (i in 1:n) if (max(abs(A[i,])) < tol) X[c(i,n),] <- X[c(n,i),]
```

does the trick.

### c. running-median function

The `for` loop in this function should start at 2 rather than at 1:

```
for (i in 2:length) X[1:(n-i+1), i] <- x[-(1:(i-1))]
```