```
> sim(250,2500)
[1] 0.6188
[1] 0.9096
```

That's quite a difference! Eicker-White worked well, whereas assuming homoscedasticity fared quite poorly. (Similar results were obtained for $n = 100$.)

4.5.3 Example: Bike Sharing Data

In our bike-sharing data (Section 1.1), there are two kinds of riders, *registered* and *casual*. We may be interested in factors determining the mix, i.e.,

$$\frac{\text{registered}}{\text{registered + casual}} \tag{4.47}$$

Since the mix proportion is between 0 and 1, we might try the logistic model, introduced in (1.36) in the context of classification. Note, though, that the example here does not involve a classification problem. so we should not reflexively use **glm()** as before. Indeed, that function not only differs from our current situation in that here Y takes on values in [0,1] rather than in {0,1}, but also **glm()** assumes

$$Var(Y \mid X =) = \mu(t)(1 - \mu(t)) \tag{4.48}$$

(as implied by Y being in {0,1}), which we have no basis for assuming here. Thus use of **glm()**, at least in the form we have seen so far, would be inappropriate. Here are the results:

```
> shar <- read.csv("day.csv",header=T)
> shar$temp2 <- shar$temp^2
> shar$summer <- as.integer(shar$season == 3)
> shar$propreg <- shar$reg / (shar$reg+shar$cnt)
> names(shar)[15] <- "reg"
> library(minpack.lm)
> logit <- function(t1,t2,t3,t4,b0,b1,b2,b3,b4)
   1 / (1 + exp(-b0 - b1*t1 -b2*t2 -b3*t3 -b4*t4))
> z <- nlsLM(propreg ~
logit(temp,temp2,workingday,summer,b0,b1,b2,b3,b4),
   data=shar,start=list(b0=1,b1=1,b2=1,b3=1,b4=1))
```

```
> summary(z)
...
Parameters:
     Estimate Std. Error t value Pr(>|t|)
b0 -0.083417   0.020814  -4.008 6.76e-05 ***
b1 -0.876605   0.093773  -9.348  < 2e-16 ***
b2  0.563759   0.100890   5.588 3.25e-08 ***
b3  0.227011   0.006106  37.180  < 2e-16 ***
b4  0.012641   0.009892   1.278    0.202
...
```

As expected, on working days, the proportion of registered riders is higher, as we are dealing with the commute crowd on those days. On the other hand, the proportion doesn't seem to be much different during the summer, even though the vacationers would presumably add to the casual-rider count.

But are those standard errors trustworthy? Let's look at the Eicker-White versions:

```
> sqrt(diag(nlsvcovhc(z)))
      fakex1      fakex2      fakex3      fakex4
0.021936045 0.090544374 0.092647403 0.007766202
      fakex5
0.007798938
```

Again, we see some substantial differences.

4.5.4 The "Elephant in the Room": Convergence Issues

So far we have sidestepped the fact that any iterative method runs the risk of nonconvergence. Or it might converge to some point at which there is only a local minimum, not the global one — worse than nonconvergence, in the sense that the user might be unaware of the situation.

For this reason, it is best to try multiple, diverse sets of starting values. In addition, there are refinements of the Gauss-Newton method that have better convergence behavior, such as the Levenberg-Marquardt method.

Gauss-Newton sometimes has a tendency to "overshoot," producing too large an increment in b from one iteration to the next. Levenberg-Marquardt generates smaller increments. Interestingly it is a forerunner of *ridge re-*