A Literate Programming Example

February 25, 2022

The Log-Likelihood

The log-likelihood for the leukemia data is

$$\ell(\beta_0, \beta_1, \beta_2, \delta) = -n \log \delta + \sum (z_i - e^{z_i})$$

with $\delta = 1/\gamma$ and $z_i = (\log t_i - \beta_0 - \beta_1 x_i - \beta_2 u_i)/\delta$.

The log likelihood is computed by the expression

```
⟨compute log-likelihood⟩=
 -n * log(delta) + sum(z - exp(z))
```

after computing `n` and `z` as

```
⟨compute n and z⟩=
 n <- length(t)
 z <- (log(t) - beta[1] - beta[2] * x - beta[3] * u) / delta
```

A function to compute the log-likelihood is therefore

```
⟨log-likelihood⟩=
 ll <- function(beta, delta, t, x, u) {
     ⟨compute n and z⟩
     ⟨compute log-likelihood⟩
 }
```

The Gradient of the Log-Likelihood

The values of `exp(z)` and `exp(z) - 1` are needed several times in computing the gradient, so it is useful to compute these once and save them in variables:

```
⟨compute exp_z and exp_z_m_1⟩=
 exp_z <- exp(z)
 exp_z_m_1 <- exp_z - 1
```

The gradient is then computed by the expression

```
⟨compute gradient⟩=
 c(sum(exp_z_m_1) / delta,
   sum(exp_z_m_1 * x) / delta,
   sum(exp_z_m_1 * u) / delta,
   (sum(exp_z * z - z) - n) / delta)
```

A function to compute the gradient of the log-likelihood is then

```
⟨gradient⟩=
 llg <- function(beta, delta, t, x, u) {
     ⟨compute n and z⟩
     ⟨compute exp_z and exp_z_m_1⟩
     ⟨compute gradient⟩
 }
```