Since these were simulated data, we can compare the extrapolations to the true relationship.
In order to achieve the same precision as the Ln scale regression with n=20 points, we would need to do nonlinear least squares \((0.027641 / 0.027641)^2 = 18 \text{ times as many points, 360 points.}\)

- You make your job much more efficient by using the right tool for the job.
```r
x = 1:22
for (sim in 1:4000) {
  Ln_y = 1 + 0.2*x + 0.2*rnorm(22)
y = exp(Ln_y)
  Ln_y = 1 + 0.2*x + 0.2*rnorm(22)
  Ln_fit = lm(Ln_y ~ x)
  xy = data.frame(cbind(x,y))
  nl_fit = nls(y ~ b0*exp(b1*x),xy,list(b0=2.7,b1=0.02))

  if( sim == 1) { Ln_fit_coef = coef(Ln_fit)
    nl_fit_coef = coef(nl_fit)}
  else
    { Ln_fit_coef = rbind(Ln_fit_coef, coef(Ln_fit))
      nl_fit_coef = rbind(nl_fit_coef,coef(nl_fit))  }
}
  Ln_fit_coef[,1] = exp(Ln_fit_coef[,1])
  mean_Ln_fit = apply(Ln_fit_coef,2,mean)
  se_Ln_fit = sqrt(apply(Ln_fit_coef,2,var))

  mean_nl_fit = apply(nl_fit_coef,2,mean)
  se_nl_fit = sqrt(apply(nl_fit_coef,2,var))

  rbind(mean_Ln_fit, se_Ln_fit)
     (Intercept)         x
   mean_Ln_fit  2.7258574 0.2000911
   se_Ln_fit         0.2329652 0.0065311

> rbind(mean_nl_fit, se_nl_fit)
   b0         b1
mean_nl_fit  2.905644 0.20248772
se_nl_fit        1.179793 0.02764136

# Find relatively how many more data are required
# for nonlinear least squares to achieve same precision
(se_nl_fit/se_Ln_fit)^2
   b0         b1
25.64660 17.91208
```