Non-linear regression

ST552 Lecture 28

Charlotte Wickham

Mar 9, 2015

Announcements

- What should we do on Friday?
- Office hours in finals week: Wed 1-2pm, Thu 2-3pm

Non-linear regression

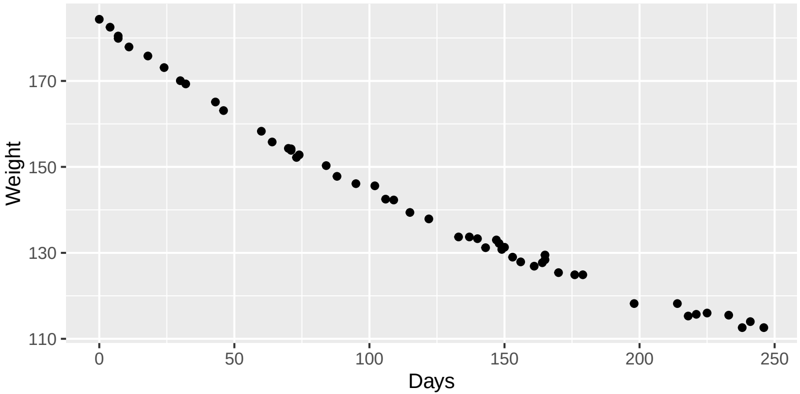
$$y_i = \eta(\mathbf{x_i}, \boldsymbol{\beta}) + \epsilon_i \quad \epsilon_i \sim N(0, \sigma^2)$$

where η is a known function $\mathbf{x}_i$ is a vector of covariates and $\boldsymbol{\beta}$ a vector of p unknown parameters.
Distinctions:

- If $\eta(\mathbf{x}, \boldsymbol{\beta}) = \mathbf{x}^T \boldsymbol{\beta}$ then we are in the usual linear regression setting.
- If $\eta(\mathbf{x}, \boldsymbol{\beta})$ is unknown but we are willing to represent it using basis functions, we are in the smooth regression setting.

Example: MASS `wtloss`

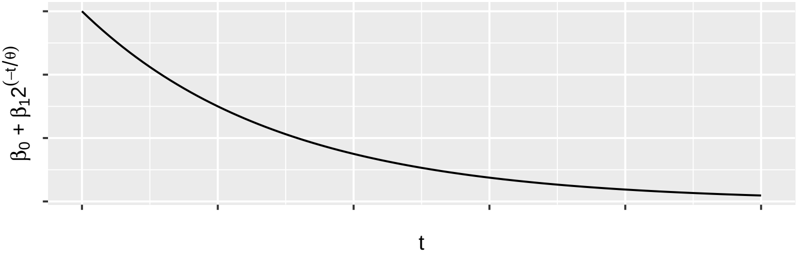
The data frame gives the weight, in kilograms, of an obese patient at 52 time points over an 8 month period of a weight rehabilitation programme.



Example: MASS `wtloss`

$$y_t = \beta_0 + \beta_1 2^{-t/\theta} + \epsilon_t$$

When $t = 0$, $\mathsf{E}(y_i) = ?$,
When $t = \infty$, $\mathsf{E}(y_i) = ?$,
$\mathsf{E}(y_i) = \beta_0 + \frac{1}{2}\beta_1$ when \$t = \$?
β_0, β_1 are linear parameters, θ is a non-linear parameter.



Fitting non-linear models

Under the Normal error assumption, the MLE of $\boldsymbol{\beta}$, minimizes

$$\sum_{i=1}^{n}(y_i - \eta(\mathbf{x_i}, \boldsymbol{\beta}))^2$$

(the sum of squared residuals) a.k.a non-linear least squares.

- There isn't in general a closed form solution, so iterative procedures are used.
- This means you need to provide starting values from the parameters and check that the procedure converged.

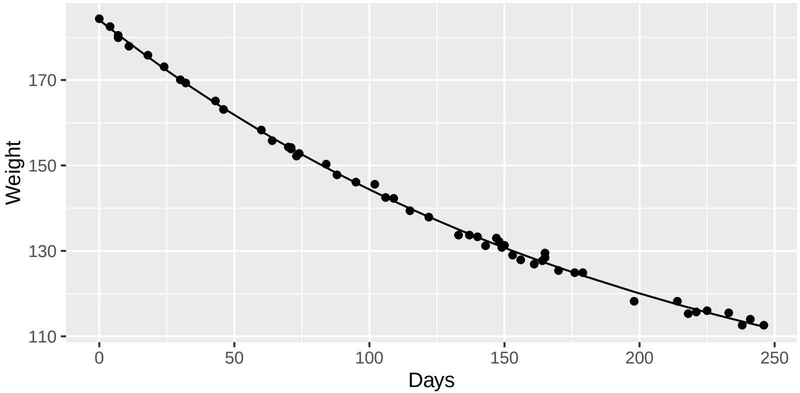
```r
wtloss.st <- c(b0 = -9000, b1 = 6, th = 4)
fit_nls <- nls(Weight ~ b0 + b1*2^(-Days/th),
  data = wtloss, start = wtloss.st)
fit_nls
```

```
## Nonlinear regression model
##   model: Weight ~ b0 + b1 * 2^(-Days/th)
##    data: wtloss
##     b0     b1     th
##  81.37 102.68 141.91
##  residual sum-of-squares: 39.24
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 6.458e-07
```

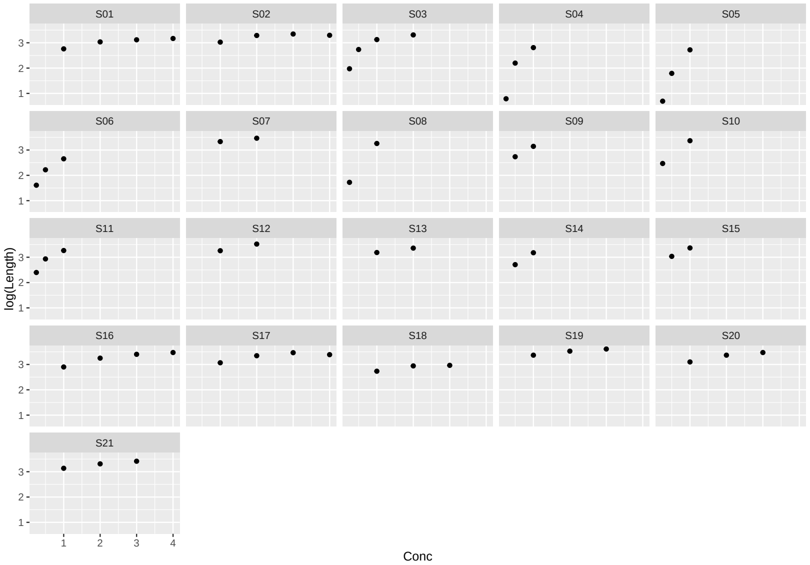
```r
summary(fit_nls)
```

```
##
## Formula: Weight ~ b0 + b1 * 2^(-Days/th)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## b0   81.374      2.269   35.86   <2e-16 ***
## b1  102.684      2.083   49.30   <2e-16 ***
## th  141.910      5.295   26.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8949 on 49 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 6.458e-07
```

```r
qplot(Days, Weight, data = wtloss) +
  geom_line(aes(y = fitted(fit_nls)))
```



```r
head(muscle)
qplot(Conc, log(Length), data = muscle) + facet_wrap(~ Strip)
```



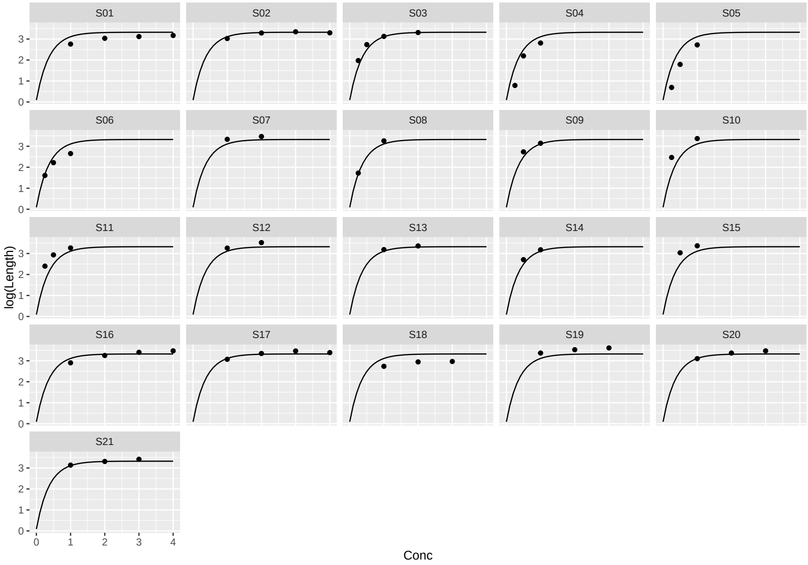
$$\log y_{ij} = \alpha + \beta\rho^{x_{ij}} + \epsilon_{ij}$$

j indexes Strip, i the ith measurement on a strip.

```
fit_all <- nls(log(Length) ~ cbind(1, rho^Conc), muscle,
  start = list(rho = 0.05), algorithm = "plinear")

pred_grid <- with(muscle, expand.grid(Conc = seq(0, 4, 0.1),
  Strip = unique(Strip)))
pred_grid$fit_all <- predict(fit_all, pred_grid)
```

The `plinear` algorithm takes advantage of the linear parameters.

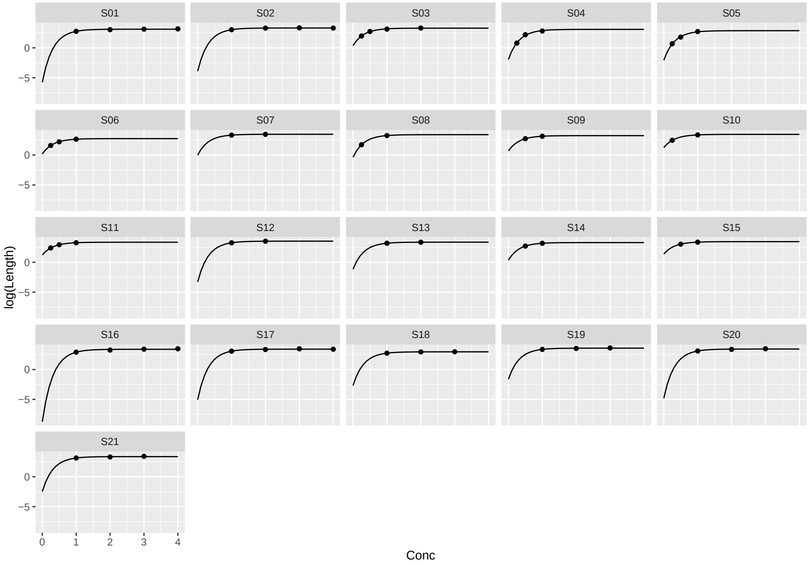


S01
S02
S03
S04
S05
S06
S07
S08
S09
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
log(Length)
Conc

$$\log y_{ij} = \alpha_j + \beta_j \rho^{x_{ij}} + \epsilon_{ij}$$

```
fit_each <- nls(log(Length) ~ alpha[Strip] + beta[Strip] * rho^Conc,
  muscle,
  start = list(rho = coef(fit_all)[1], alpha = rep(coef(fit_all)[2], 21
    beta = rep(coef(fit_all)[3], 21)))

pred_grid$fit_each <- predict(fit_each, pred_grid)
```


S01
S02
S03
S04
S05
S06
S07
S08
S09
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
0
−5
0
1
2
3
4
log(Length)
Conc