Fitting models

Bjarki Þór Elvarsson and Einar Hjörleifsson

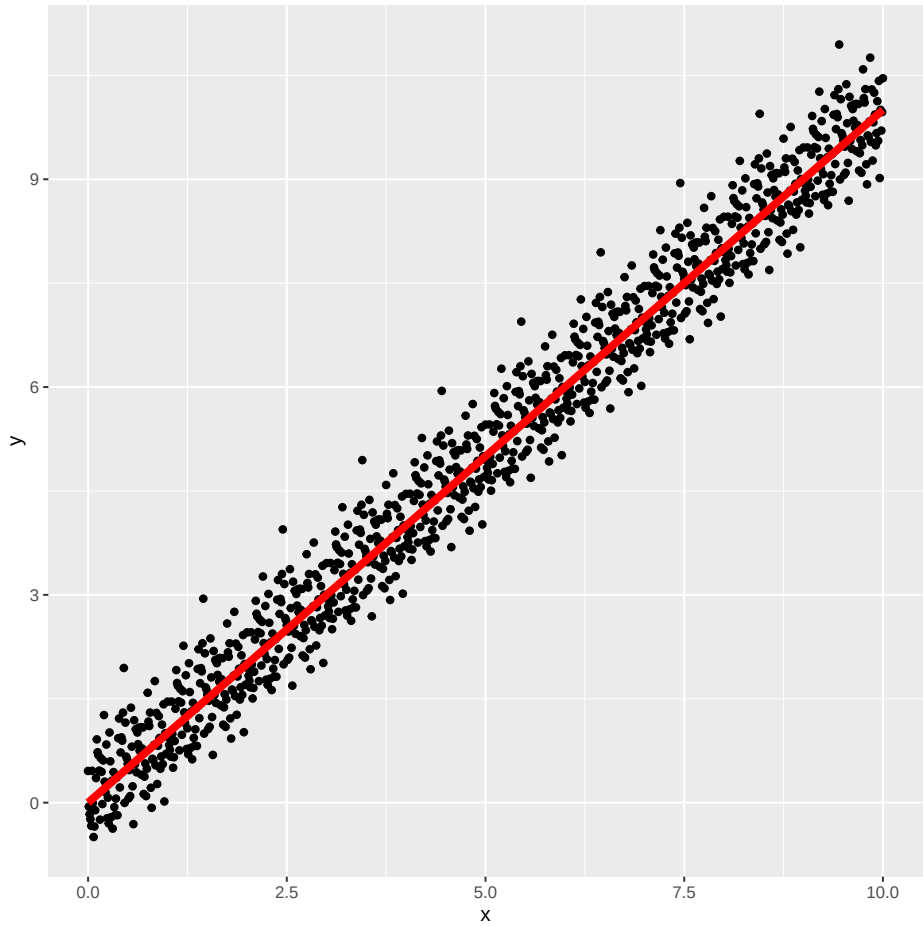
Marine Research Institute

Linear models

- A linear model is a model that can be written as:

$$y = a + b * x_1 + c * x_2 + \dots$$

  where y is the predicted value, $x_1, x_2, \dots$ the input variables and a and b the parameters
- In statistics the goal is often to relate y, i.e. the observations, and the independent variables (x_i-s)
- This can be done using linear regression



Simple model fitting

- In essence regression analysis is simply the calculation of a slope parameter and the intercept.
- Observations, however, are often noisy which is often written as:

$$Y_i = \alpha + \beta X_i + \epsilon_i$$

where ϵ_i denotes the difference between observations and predictions

- The goal with regression is to find minimize this difference to produce more accurate predictions of the observations, i.e. find α and β such that

$$\sum_i (Y_i - \alpha - \beta X_i)^2$$

is as small as possible

Length weight relationship

- Now we want to predict the weights for fish that were not weighed, using linear regression
- Often one sees the length weight relationship charaterised as:

$$W = a * L^b$$

- Now this is a non-linear relationship (if $b \neq 1$) but is easy to linearise using $\log$:

$$\log(W) = \log(a) + b * \log(L)$$

Let's play a bit

```
wlFun <- function(dat,a,b){
  return(list(p=ggplot(dat,aes(log(length),log(ungutted))) +
                geom_point() +
                geom_abline(intercept=log(a),slope=b)+
                ylim(c(0,10)) + xlim(c(0,10)),
              ss=sum((log(dat$ungutted) - log(a) -
                        b*log(dat$length))^2)))
}
 fish %>% filter(!is.na(weight)) %>% wlFun(exp(0),0)
```

Estimate using R

R has linear regression by default, invoked using "lm"

```
fit <- lm(log(weight)~log(length),data=minke)
summary(fit)

##
## Call:
## lm(formula = log(weight) ~ log(length), data = minke)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24333 -0.10753 -0.02323  0.08834  0.31672
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.6445     2.4452  -4.353 0.000342 ***
## log(length)   2.8601     0.3695   7.741 2.72e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1402 on 19 degrees of freedom
##   (169 observations deleted due to missingness)
## Multiple R-squared:  0.7592,^^IAdjusted R-squared:  0.7466
## F-statistic: 59.92 on 1 and 19 DF,  p-value: 2.725e-07
```

Estimate using R

One can add other variables into the regression fairly easily:

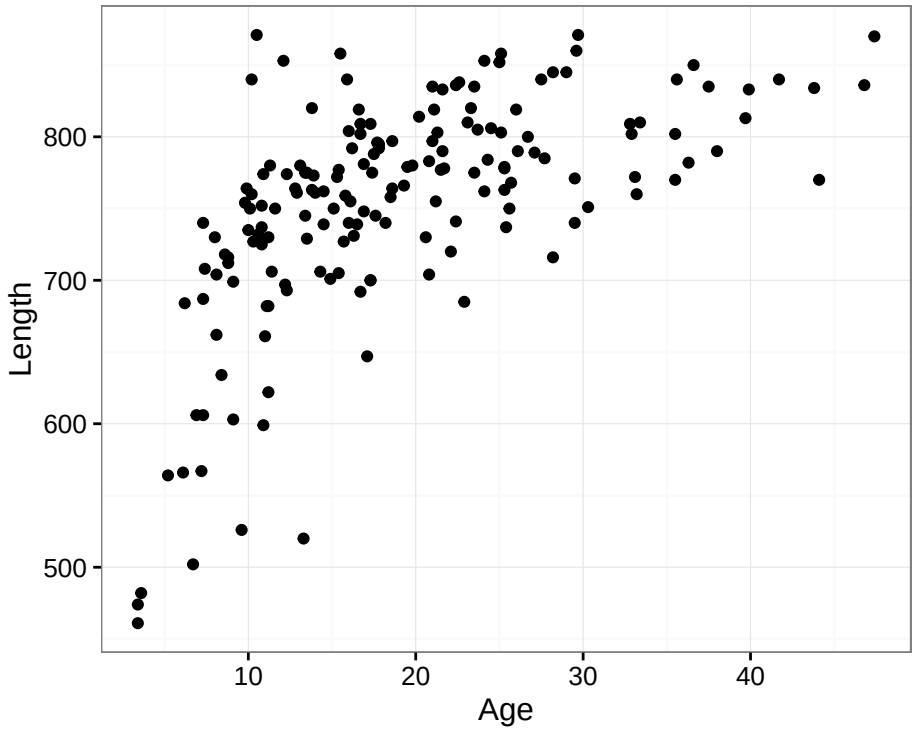
```
fit <- lm(log(weight)~log(length) + sex,data=minke)
summary(fit)

##
## Call:
## lm(formula = log(weight) ~ log(length) + sex, data = minke)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24682 -0.10954 -0.01908  0.08742  0.31123
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.510660   2.730928  -3.849  0.00118 **
## log(length)   2.840432   0.410935   6.912 1.84e-06 ***
## sexMale      -0.008575   0.068755  -0.125  0.90213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.144 on 18 degrees of freedom
##   (169 observations deleted due to missingness)
## Multiple R-squared:  0.7595,^^IAdjusted R-squared:  0.7327
## F-statistic: 28.41 on 2 and 18 DF,  p-value: 2.697e-06
```

Non-linear model fitting

- Now say we want to fit a growth curve to our minke whale data
- Typically this would by a Von Bertalanffy growth curve of the form:

$$l = L_\infty(1 - e^{-k(a-t_0)})$$

- How do we do this in R?



What do we want to do exactly?

- Again we want to find the best fitting curve through the datapoints, although now we want estimate a more arbitrary function
- This means that we want to "draw" a line that minimized on average the distance to all data points, i.e. find x that solves

$$\min_{\mathbf{x}} \left(\sum_i (l_i - \text{VonB}(\mathbf{x}, a(i)))^2 \right)$$

- In the Von B function there are three parameters, L_∞, k and t_0 that can be adjusted so the task here is to find values of these three parameters such that the above sum is minimized

In R:

```
age.data <- filter(minke,!is.na(age))
minke.vonB.par <-
  nls(length~Linf*(1-exp(-K*(age-t0))),
      data=age.data, start=list(Linf=1100, K=0.1, t0=-1))
minke.vonB.par

## Nonlinear regression model
##   model: length ~ Linf * (1 - exp(-K * (age - t0)))
##    data: age.data
##     Linf         K        t0
## 799.9246   0.1866  -1.4930
##  residual sum-of-squares: 458504
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 4.678e-06
```

Formulas in nls

```
nls(length~Linf*(1-exp(-K*(age-t0))),
    data=age.data, start=list(Linf=1100, K=0.1, t0=-1))
```

- Formulas in R typically look for variables in the data, in this case the minke whale dataset.
- If a variable is not in the data, such as variables "Linf", "K" and "t0", they are assumed to be parameters that need to be estimated
- Starting values are given in the input as "start". If not given the function may converge to a wrong minima or not at all.

Confidence intervals

Recall that a 95% confidence interval represents the potential range of the data, i.e. one can not reject the hypothesis that the parameter estimate is within the range. Confidence intervals can be computed using the following command:

```
confint(minke.vonB.par)

## Waiting for profiling to be done...

##              2.5%        97.5%
## Linf 785.9044786 816.53179542
## K      0.1413882   0.23771389
## t0    -3.9617852   0.06320937
```

Plotting the VonB estimate

First define a Von B function:

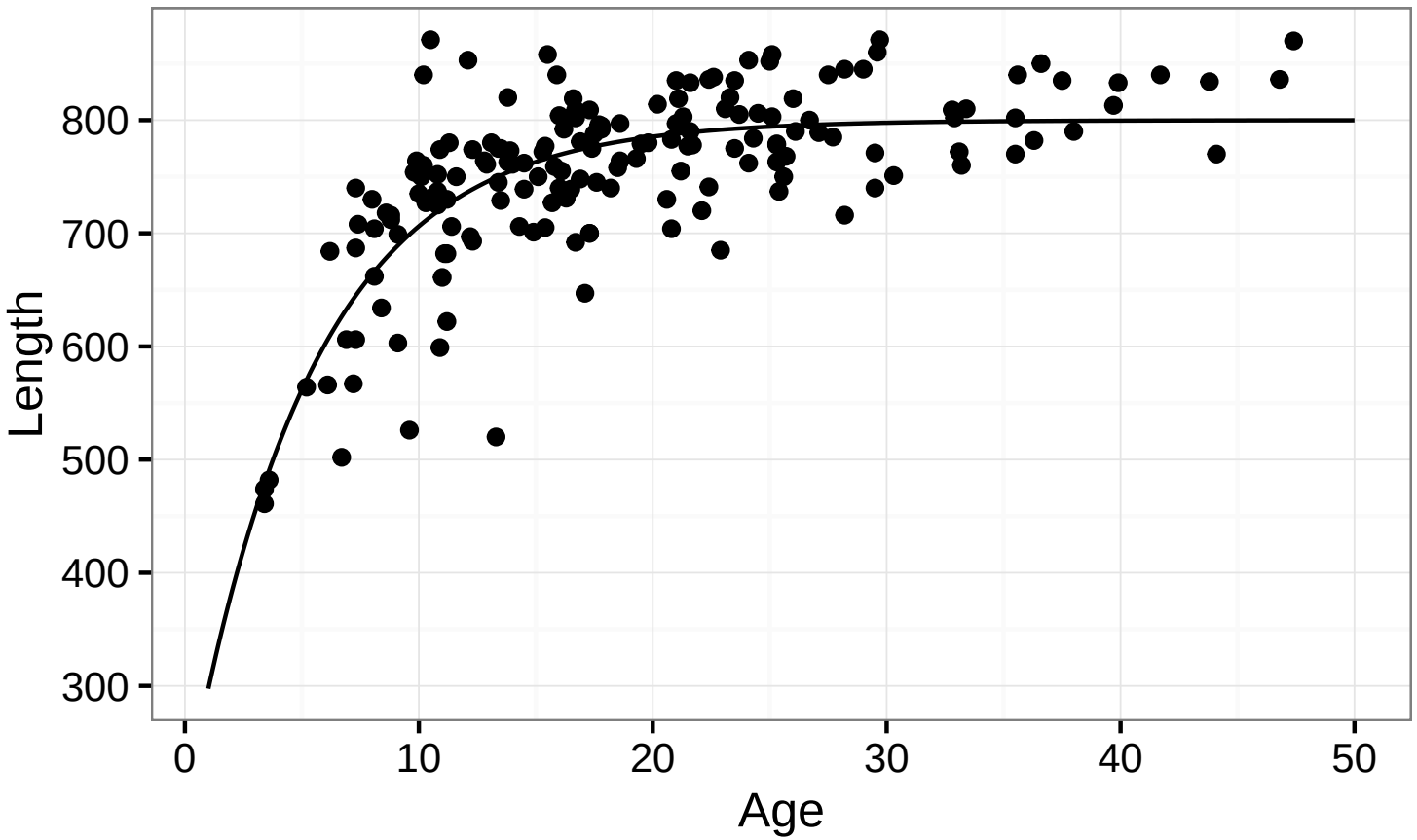
```
vonB <- function(linf,k,a,t0){
  gr <- linf * (1 - exp(-k * (a-t0)))
  return(gr)
}
```

Then calculate the average length for each age

```
x <- coefficients(minke.vonB.par) ## get the coefficients
age <- seq(1,50,by=1/12) ## age by month
pred.length <- vonB(x[1],x[2],age,x[3])
pred.dat <- ## create a data table
  data.frame(age=age,length=pred.length)
```

And plot:

```
ggplot(minke,aes(age,length)) + geom_point() +
  geom_line(data=pred.dat) +
  theme_bw() + ylab('Length') + xlab('Age')
```



In class excercise

Using a sample species from the DATRAS CA table:

- Estimate the length - weight relationship
- Estimate the Von Bertalanffy curve
- Estimate the maturity ogive
- Plot the results