An introduction to Multiple Linear Model (LM) using R

Benoit Liquet[1]

November 29, 2016

Outline

1. Reminder on regression and the linear model:
    - Model.
    - Assumptions.
    - The `lm()` function.

Regression and ANOVA models

Regression:

Relationship between a quantitative response and **quantitative** (continuous) explanatory variables.

ANOVA:

Relationship between quantitative response and **categorical** (discrete) explanatory variables.

The most basic regression model involves a linear relationship between the response and a single explanatory variable.

Data analysis

Aquatic example: Problematic

Rivers in North Carolina contain small concentrations of mercury which can accumulate in fish over their lifetimes. Because mercury cannot be excreted from the body, it builds up in the tissues. The concentration of mercury in fish tissue can be obtained at considerable expense by catching fish and sending samples to a lab for analysis. Directly measuring the mercury concentration in the water is impossible since it is almost always below detectable limits.

Aquatic example: Study

A study was recently conducted in the Wacamaw and Lumber Rivers to investigate mercury levels in tissues of large mouth bass. At several stations along each river, a group of fish were caught, weighed, and measured. In addition a filet from each fish caught was sent to the lab so that the tissue concentration of mercury could be determined for each fish. The recorded information for each fish is: `river`, `station`, `length in cm`, `weight in grams`, `mercury concentration in parts per million`.

Data analysis

Some questions:

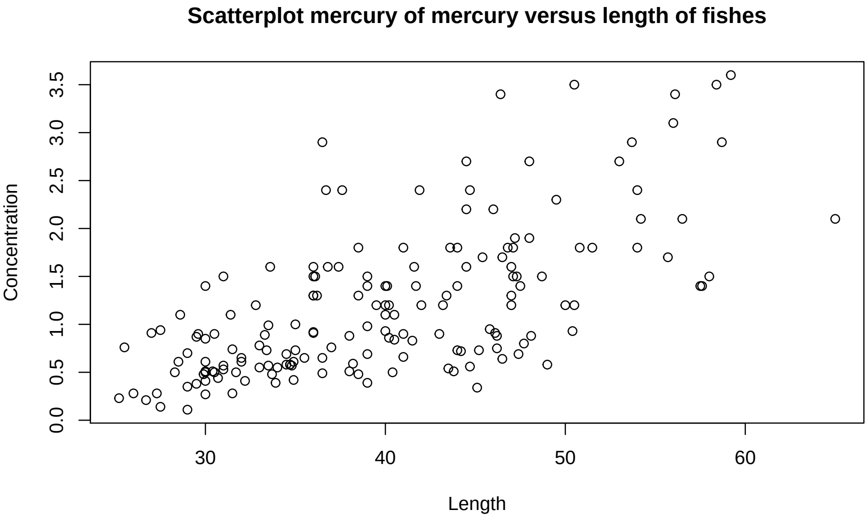
- Is there a relationship between mercury concentration and size (weight and/or length) of a fish?
- Is this relationship the same for the two rivers?

The data

```
fishHG <- read.table("/Users/liquetwe/Dropbox/ANGLET/TEACHING/M2/GLM/DATA/fishHG.txt"
    header = TRUE)
attach(fishHG)
```

```
head(fishHG)

   river station length_cm weight_g HG_conc_ppm
1 lumber      l1      47.0     1616        1.60
2 lumber      l1      48.7     1862        1.50
3 lumber      l1      55.7     2855        1.70
4 lumber      l1      45.2     1199        0.73
5 lumber      l1      44.7     1320        0.56
6 lumber      l1      43.8     1225        0.51

dim(fishHG)

[1] 171   5
```

Scatterplot



Simple Linear Regression Model

Simple Linear Regression Model

The response data $Y_1, \ldots, Y_n$ depend on explanatory variables $x_1, \ldots, x_n$ via the linear relationship

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i\ , \quad i = 1, \ldots, n\ , \tag{1}$$

where $\varepsilon_1, \ldots, \varepsilon_n \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$.

We view the responses as random variables which would lie exactly on the **regression line** $y = \beta_0 + \beta_1 x$, were it not for some "disturbance" or "error" term (represented by the $\{\varepsilon_i\}$).

Aquatic example data

Let x_i be the i-th length fish in cm (stored in `length_cm`) and y_i the corresponding mercury concentration in ppm (stored in `HG_conc_ppm`).

For the pairs $(x_1, Y_1), \ldots, (x_n, Y_n)$, we assume model (1).

Note that the model has three unknown parameters: β_0, β_1, and σ^2.

What can we say about the model parameters on the basis of the observed data $(x_1, y_1), \ldots, (x_n, y_n)$?

Estimating the parameters

Obviously we do not know the true regression line $y = \beta_0 + \beta_1 x$, but we can try to fit a line $y = \widehat{\beta}_0 + \widehat{\beta}_1 x$ that best "fits" the data.

$\widehat{\beta}_0$ and $\widehat{\beta}_1$ are estimates for the unknown intercept β_0 and slope β_1.

For each x_i, let $\widehat{y_i} = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$.

The difference $e_i = y_i - \widehat{y_i}$ is called a **residual error**, or simply **residual**.

There are various measures for "best fit", but a very convenient one is minimise the sum of the squared residual errors, $\mathrm{SSE} = \sum_{i=1}^{n} e_i^2$. This gives the following *least-squares* criterion:

$$\text{minimise SSE} . \tag{2}$$

Least squares estimates

Least squares estimates

The values for $\widehat{\beta}_1$ and $\widehat{\beta}_0$ that minimise the least-squares criterion are:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{3}$$

$$\widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x} \ . \tag{4}$$

Proof

We seek to minimise the function

$$g(a, b) = \text{SSE} = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

with respect to a and b.

Proof

We seek to minimise the function

$$g(a,b) = \text{SSE} = \sum_{i=1}^{n}(y_i - a - bx_i)^2$$

with respect to a and b.

To find the optimal a and b, we take the derivative of SSE with respect to a, b and set it equal to 0. This leads to two linear equations:

$$\frac{\partial \sum_{i=1}^{n}(y_i - a - bx_i)^2}{\partial a} = -2\sum_{i=1}^{n}(y_i - a - bx_i) = 0$$

and

$$\frac{\partial \sum_{i=1}^{n}(y_i - a - bx_i)^2}{\partial b} = -2\sum_{i=1}^{n}x_i(y_i - a - bx_i) = 0\ .$$

Properties

Both $\widehat{\beta}_0$ and $\widehat{\beta}_1$ have a normal distribution. Their expected values are

$$\mathbb{E}(\widehat{\beta}_0) = \beta_0 \quad \text{and} \quad \mathbb{E}(\widehat{\beta}_1) = \beta_1 \ , \tag{5}$$

so both are *unbiased* estimators. Their variances are

$$\mathrm{Var}(\widehat{\beta}_0) = \sigma^2 \left(\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^n (x_i - \overline{x})^2} \right) \tag{6}$$

and

$$\mathrm{Var}(\widehat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \overline{x})^2} \ . \tag{7}$$

Hypothesis testing

It is of interest to test whether there is no association between the response and the explanatory variable.

Hence, we consider $H_0 : \beta_1 = 0$ (slope = 0) and see if there is evidence against it.

There

Hypothesis testing

It is of interest to test whether there is no association between the response and the explanatory variable.

Hence, we consider $H_0 : \beta_1 = 0$ (slope = 0) and see if there is evidence against it.

There are two approaches that we could use to construct a good test statistic.

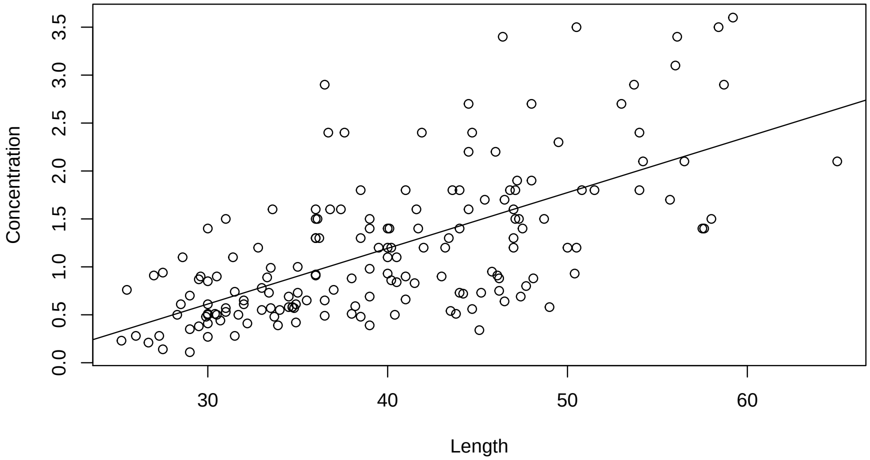
- t-test approach
- ANOVA approach

Linear regression with length as predictor

```
Hg.river.length.lm = lm(HG_conc_ppm ~ length_cm, data = fishHG)
summary(Hg.river.length.lm)
```

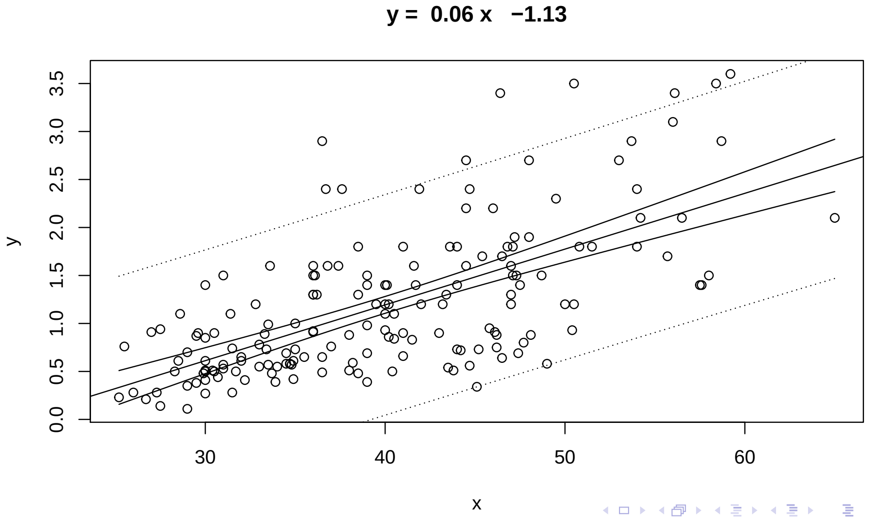
```
Call:
lm(formula = HG_conc_ppm ~ length_cm, data = fishHG)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1499 -0.3436 -0.1022  0.3123  1.9100

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.131645   0.213615  -5.298 3.62e-07 ***
length_cm    0.058127   0.005228  11.119  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5805 on 169 degrees of freedom
Multiple R-squared:  0.4225,Adjusted R-squared:  0.4191
F-statistic: 123.6 on 1 and 169 DF,  p-value: < 2.2e-16
```

Linear regression



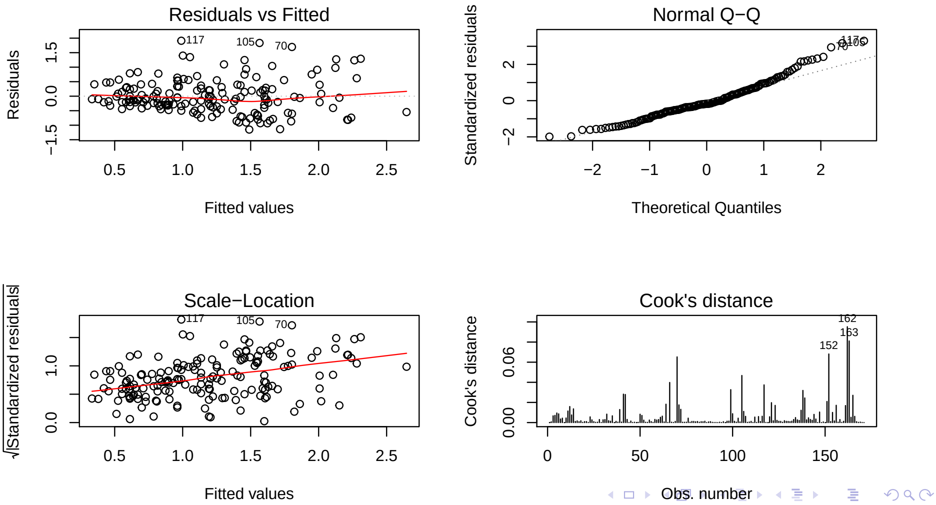
Linear model

```r
require(UsingR)
model.s <- simple.lm(length_cm, HG_conc_ppm, show.ci = TRUE)
```



Check asssumptions

```
par(mfrow = c(2, 2))
plot(Hg.river.length.lm, which = 1:4)
```



Summary for simple linear regression

The table below presents the main functions to use for simple linear regression between the response variable `y` and the explanatory variable `x`.

Table: Main R functions for simple linear regression.

R instruction	Description
`plot(y~x)`	scatter plot
`lm(y~x)`	estimation of the linear model
`summary(lm(y~x))`	description of results of the model
`abline(lm(y~x))`	draw the estimated line
`confint(lm(y~x))`	confidence interval for regression parameters
`predict()`	function for predictions
`plot(lm(y~x))`	graphical analysis on residuals

Multiple Linear Regression Model

A linear regression model that contains more than one explanatory variable is called a *multiple linear regression model*.

multiple linear regression

In a **multiple linear regression model** the response data $Y_1, \ldots, Y_n$ depend on d-dimensional explanatory variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$, with $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^\top$, via the linear relationship

$$Y_i = \beta_0 + \beta_1 \, x_{i1} + \cdots + \beta_d \, x_{id} + \varepsilon_i \,, \quad i = 1, \ldots, n \,, \tag{8}$$

where $\varepsilon_1, \ldots, \varepsilon_n \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$.

Linear Model

Much of modeling in applied statistics is done via the versatile class of linear models.

Linear Model

Much of modeling in applied statistics is done via the versatile class of linear models.

Both the ANOVA and linear regression models are special cases of linear models.

Linear Model

Much of modeling in applied statistics is done via the versatile class of linear models.

Both the ANOVA and linear regression models are special cases of linear models.

Let $\mathbf{Y}$ be the column vector of response data $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$.

Linear model

In a **linear model** the response data vector $\mathbf{Y}$ depends on a matrix $\mathcal{X}$ of explanatory variables (called the **design matrix**) via the linear relationship

$$\mathbf{Y} = \mathcal{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\beta}$ is a vector of parameters and $\boldsymbol{\varepsilon}$ a vector of independent error terms, each $\mathrm{N}(0, \sigma^2)$ distributed.

Example: simple linear regression

For the simple linear regression model

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

we have

$$\boldsymbol{\mathcal{X}} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}.$$

Example: 1-factor ANOVA model

Consider a 1-factor ANOVA model with 3 levels and 2 replications per levels. Denoting the responses by

$$\underbrace{Y_1, Y_2}_{level 1}, \underbrace{Y_3, Y_4}_{level 2}, \underbrace{Y_5, Y_6}_{level 3},$$

and the expectations within the levels by μ_1, μ_2, and μ_3, we can write the vector $\mathbf{Y}$ as

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_1 \\ \mu_2 \\ \mu_2 \\ \mu_3 \\ \mu_3 \end{pmatrix} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix}}_{\boldsymbol{\varepsilon}} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathcal{X}} \underbrace{\begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}}_{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}\,.$$

Indicator variables

If we denote for each response Y the level by x, then we can write

$$Y = \mu_1 \,\mathrm{I}(x = 1) + \mu_2 \,\mathrm{I}(x = 2) + \mu_3 \,\mathrm{I}(x = 3) + \varepsilon, \tag{9}$$

where $\mathrm{I}(x = k)$ is an **indicator** or *dummy* variable that is 1 if $x = k$ and 0 otherwise, $k = 1, 2, 3$. As an alternative to (9) we could use the “factor effects” representation

$$Y = \mu + \alpha_1 \mathrm{I}(x = 1) + \alpha_2 \,\mathrm{I}(x = 2) + \alpha_3 \,\mathrm{I}(x = 3) + \varepsilon, \tag{10}$$

where $\alpha_1 + \alpha_2 + \alpha_3 = 0$. Or we could use the representation

$$Y = \mu + \alpha_2 \,\mathrm{I}(x = 2) + \alpha_3 \,\mathrm{I}(x = 3) + \varepsilon, \tag{11}$$

where $\alpha_1 = 0$. In this case μ should be interpreted as the expected response in level 1.

In R, all data from a general linear model is assumed to be of the form

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, \ldots, n \,, \tag{12}$$

where x_{ij} is the j-th explanatory variable for individual i and the errors ε_i are independent random variables such that $\mathbb{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$.

In matrix form, $\mathbf{Y} = \mathcal{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathcal{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix} \text{ and } \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

R formula

Thus, the first column can always be interpreted as an “intercept” parameter. The corresponding `R` formula for this model would be

$$y \sim x1 + x2 + \cdots + xp \; .$$

It is important to note that `R` automatically treats quantitative and qualitative explanatory variables differently. For any linear model you can retrieve the design matrix via the function `model.matrix()`.

Example

Suppose the data are given in the following table

y	x1	x2
10	7.4	1
9	1.2	1
4	3.1	2
2	4.8	2
4	2.8	3
9	6.5	3

where y is the response, x_1 is quantitative (continuous) and x_2 is qualitative (factor).

```
> my.dat <- data.frame(y = c(10,9,4,2,4,9),
+     x1=c(7.4,1.2,3.1,4.8,2.8,6.5),x2=as.factor(c(1,1,2,2,3,3)))
> mod <- lm(y~x1+x2,data = my.dat)
```

Print the design matrix:

```
> print(model.matrix(mod))
```

```
  (Intercept)  x1 x22 x23
1           1 7.4   0   0
2           1 1.2   0   0
3           1 3.1   1   0
4           1 4.8   1   0
5           1 2.8   0   1
6           1 6.5   0   1
```

Warning

By default, R sets the incremental effect α_i of the first-named level (in alphabetical order) to zero.

The mathematical model is thus:

$$Y = \mu + \beta_1 x_1 + \alpha_2 \mathrm{I}(x_2 = 2) + \alpha_3 \mathrm{I}(x_2 = 3) + \varepsilon. \quad (13)$$

Estimating $\boldsymbol{\beta}$

Suppose we have a vector data $\mathbf{y}$ from a linear model

$$\mathbf{Y} = \mathcal{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\mathcal{X}$ is a known design matrix.

To estimate the parameter vector $\boldsymbol{\beta}$ we can again use a least-squares: Find $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_0, \ldots, \widehat{\beta}_p)^\top$ such that

$$\sum_{i=1}^{n} (y_i - \{\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \widehat{\beta}_2 x_{i2} + \cdots + \widehat{\beta}_p x_{ip}\})^2 \quad \text{is minimal.}$$

It can be shown that this gives the least squares estimate

$$\widehat{\boldsymbol{\beta}} = (\mathcal{X}^\top \mathcal{X})^{-1} \mathcal{X}^\top \mathbf{y},$$

where $(\mathcal{X}^\top \mathcal{X})^{-1}$ is the inverse of the matrix $\mathcal{X}^\top \mathcal{X}$.

Data analysis

Some questions:

- Is there a relationship between mercury concentration and size (weight and/or length) of a fish?
- Is this relationship the same for the two rivers?

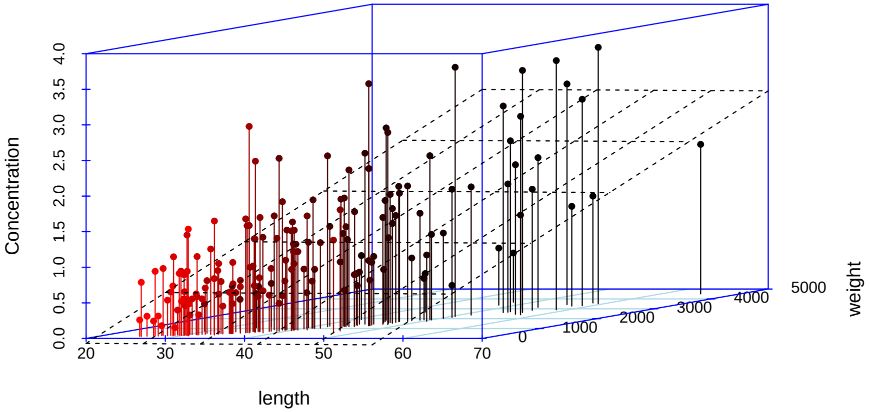
Linear model with length and weight as predictors

```
Hg.lm2 <- lm(HG_conc_ppm ~ length_cm + weight_g, data = fishHG)
summary(Hg.lm2)
```

```
Call:
lm(formula = HG_conc_ppm ~ length_cm + weight_g, data = fishHG)

Residuals:
     Min       1Q   Median       3Q      Max
-1.14980 -0.35556 -0.08829  0.31022  1.87271

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.4961930  0.3658015  -4.090 6.67e-05 ***
length_cm    0.0713530  0.0119785   5.957 1.47e-08 ***
weight_g    -0.0001429  0.0001165  -1.227    0.222
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5797 on 168 degrees of freedom
Multiple R-squared:  0.4276,Adjusted R-squared:  0.4208
F-statistic: 62.76 on 2 and 168 DF,  p-value: < 2.2e-16
```

Visualisation

```
Warning:  package 'scatterplot3d' was built under R version 3.2.5
```



Visualisation

```r
library(Rcmdr)
attach(mtcars)
scatter3d(wt, disp, mpg)
```

Prediction

Prediction with the predictors, used for fitting the model

```
pred <- predict(Hg.lm2)
# Prediction with new predictors
x1 <- seq(25, 70, l = 100)
x2 <- seq(400, 1400, l = 100)
new.data <- data.frame(length_cm = x1, weight_g = x2)
pr2 <- predict.lm(Hg.lm2, new.data)
head(data.frame(new.data, pr2))

  length_cm weight_g       pr2
1  25.00000 400.0000 0.2304546
2  25.45455 410.1010 0.2614439
3  25.90909 420.2020 0.2924332
4  26.36364 430.3030 0.3234225
5  26.81818 440.4040 0.3544118
6  27.27273 450.5051 0.3854011
```

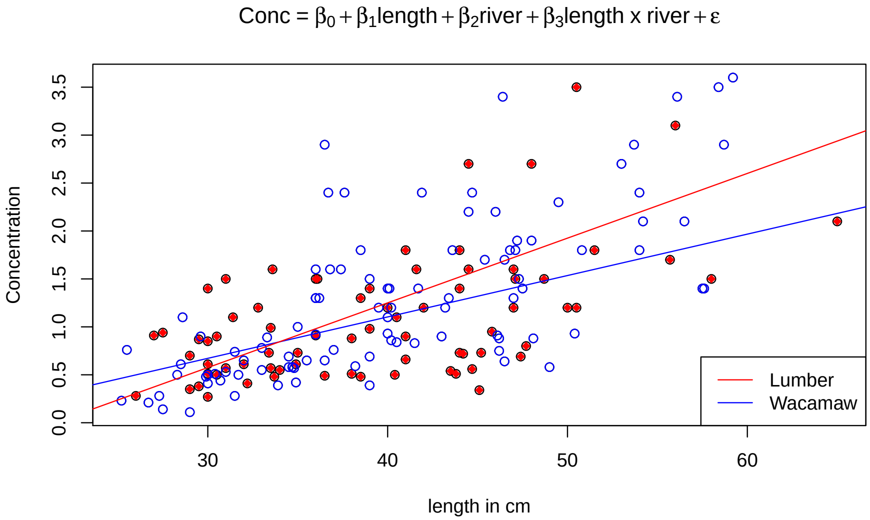
Is this relationship the same for the two rivers?

```
Hg.lm3 <- lm(HG_conc_ppm ~ length_cm + river + length_cm * river,
    data = fishHG)
summary(Hg.lm3)



Call:
lm(formula = HG_conc_ppm ~ length_cm + river + length_cm * river,
    data = fishHG)

Residuals:
     Min       1Q   Median       3Q      Max
-1.27784 -0.35402 -0.08314  0.30650  1.94304

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            -0.623875   0.325576  -1.916   0.0570 .
length_cm               0.043185   0.008085   5.341 2.99e-07 ***
riverwacamaw           -0.826291   0.426529  -1.937   0.0544 .
length_cm:riverwacamaw  0.024326   0.010483   2.321   0.0215 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5705 on 167 degrees of freedom
Multiple R-squared:  0.4488,Adjusted R-squared:  0.4389
F-statistic: 45.33 on 3 and 167 DF,  p-value: < 2.2e-16
```

Visualisation

```r
co <- coef(Hg.lm3)
a0 <- co[1]
a1 <- co[1] + co[3]
b0 <- co[2]  # Effect of length on lumber.
b1 <- co[2] + co[4]  # Effect of length on wacamaw.
plot(HG_conc_ppm ~ length_cm, xlab = "length in cm", ylab = "Concentration",
    main = expression(Conc ~ "=" ~ beta[0] + beta[1] * length +
        beta[2] * river + beta[3] * length ~ "x" ~ river + epsilon))
points(length_cm[river == "lumber"], HG_conc_ppm[river == "lumber"],
    col = "red", pch = 18)
points(length_cm[river == "wacamaw"], HG_conc_ppm[river == "wacamaw"],
    col = "blue")
abline(a = a0, b0, col = "blue")
abline(a = a1, b1, col = "red")
legend("bottomright", c("Lumber", "Wacamaw"), col = c("red",
    "blue"), lty = 1)
```

Visualisation



Summary for linear model

Table: Main R functions for linear model.

R instruction	Description
`pairs()`	graphical inspection
`lm(y~x1+x2+...+x3)`	estimation of the multiple linear model
`summary(lm())`	description of the results of the model
`confint(lm())`	confidence interval for regression parameters
`predict()`	function for predictions
`plot(lm())`	graphical analysis of residuals
`x1:x2`	interaction between x_1 and x_2