Case Study 3: Basic model fitting: Leaf burning

Richard Wilkinson

The data

An experiment to determine the effect on the burn time of leaves on the concentration of Nitrogen, Chlorine and Potassium. To build a purely scientific model from first principles based on chemistry and physics alone, would be extremely challenging. So instead, we propose various statistical models and fit them to the observations.

```
filepath <- "https://www.maths.nottingham.ac.uk/personal/pmzrdw/LeafData.txt"
# Download the data from the internet
download.file(filepath, destfile = "LeafData.txt", method = "curl")
LeafData <- read.table(file='LeafData.txt', header=TRUE)
LeafData[1:10,]
```

```
##    Nitrogen Chlorine Potassium log_burn_time
## 1      3.05     1.45      5.67          0.34
## 2      4.22     1.35      4.86          0.11
## 3      3.34     0.26      4.19          0.38
## 4      3.77     0.23      4.42          0.68
## 5      3.52     1.10      3.17          0.18
## 6      3.54     0.76      2.76          0.00
## 7      3.74     1.59      3.81          0.08
## 8      3.78     0.39      3.23          0.11
## 9      2.92     0.39      5.44          1.53
## 10     3.10     0.64      6.16          0.77
```

```
str(LeafData) # look at the data structure
```

```
## 'data.frame':    30 obs. of  4 variables:
##  $ Nitrogen     : num  3.05 4.22 3.34 3.77 3.52 3.54 3.74 3.78 2.92 3.1 ...
##  $ Chlorine     : num  1.45 1.35 0.26 0.23 1.1 0.76 1.59 0.39 0.39 0.64 ...
##  $ Potassium    : num  5.67 4.86 4.19 4.42 3.17 2.76 3.81 3.23 5.44 6.16 ...
##  $ log_burn_time: num  0.34 0.11 0.38 0.68 0.18 0 0.08 0.11 1.53 0.77 ...
```

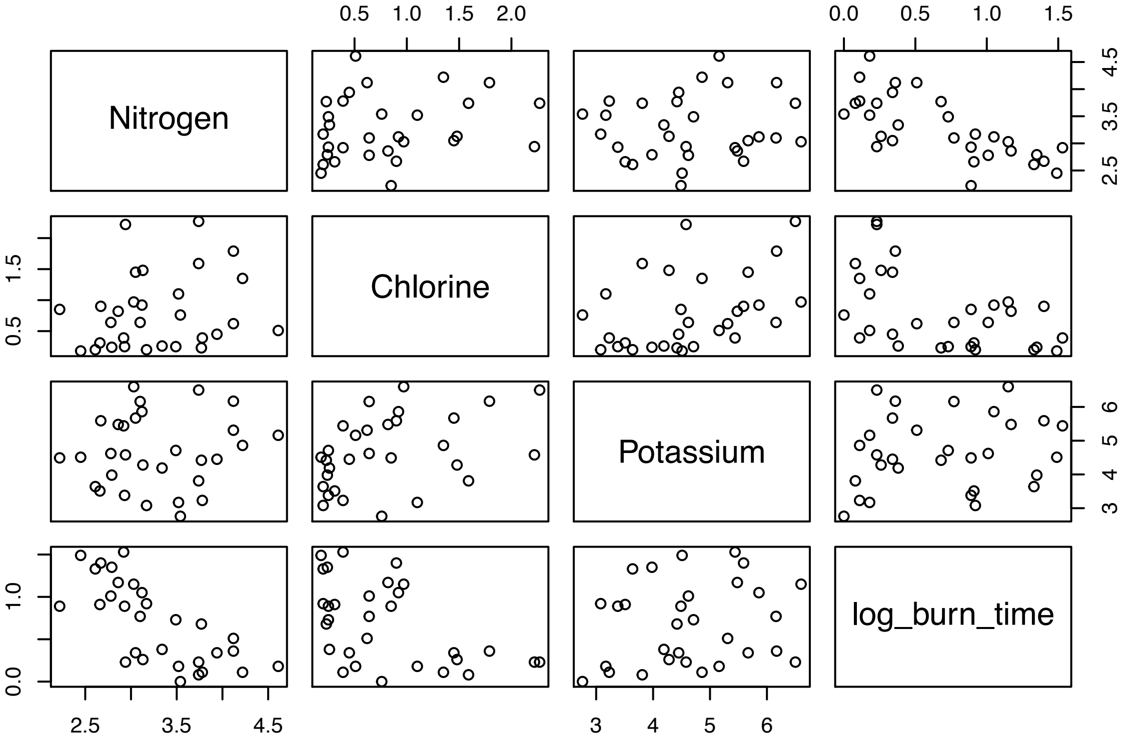
The variables are

- Nitrogen %
- Chlorine %

- Potassium %
- log of leaf burn time

Lets start (as always), by visualising the data.

```
plot(LeafData)
```



If you choose not to use R markdown, then you will have to manually save all of your figures using a command such as

```
dev.print(pdf, file="FuelScatterPlots.pdf", width=12, height=12) # save the file
```

What questions might we want to ask?

- How does the percentage of nitrogen, chlorine and potassium affect the leaf burn time?
- Which percentage is most important?

Possible models

For the ith observation let y_i be the log of leaf burn time, $x_{i,1}$ be the Nitrogen %, $x_{i,2}$ be the Chlorine %, and $x_{i,3}$ be the Potassium %. Let $\beta = (a, b, c)^T$.

Which of the following are linear models?

- $y_i = bx_{i,1} + \epsilon_i$,
- $y_i = a + bx_{i,1} + \epsilon_i$,
- $y_i = b^3 x_{i,1} + cx_{i,2} + \epsilon_i$,
- $y_i = a + b(x_{i,1} - x_{i,2})^2 + \epsilon_i$,
- $y_i = a + 2bx_{i,1} + c\log(x_{i,2}) + \epsilon_i$.
- $y_i = ax_{i,1} + bx_{i,1}^2 + \epsilon_i$.

What is the design matrix Z and parameter vector β in the cases that are linear models?

Simple linear regression

Let y be the log leaf burn time and x be the Nitrogen %. We can write our simple linear regression model as

$$y_i = a + bx_i + \epsilon_i, \qquad i = 1, \ldots, 30.$$

What are β, $g(x)^T$ and Z here?

To find the sum of squares estimates for this model using R, we can do the following:

```r
fit1 <- lm(log_burn_time~Nitrogen, data = LeafData)
fit1
```

```
##
## Call:
## lm(formula = log_burn_time ~ Nitrogen, data = LeafData)
##
## Coefficients:
```

```
## (Intercept)     Nitrogen
##      2.6257      -0.5916
```

```
coef(fit1)  # explicitly gives the fitted coefficients
```

```
## (Intercept)    Nitrogen
##   2.6257040  -0.5916137
```

```
deviance(fit1)
```

```
## [1] 3.243512
```

Let D_1 denote the deviance of this first order linear fit.

Model Choice

We can fit a quadratic model.

$$y_i = a + bx_i + cx_i^2 + \epsilon_i \qquad i = 1, \ldots, n.$$

The vector of parameters is $\beta = (a, b, c)^T$,

$z_i^T = [1 \ x_i \ x_i^2]$ and the design matrix is

$$Z = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}$$

NB: This is a linear model since it is linear in the *parameters*.

```
fit2 <- lm(log_burn_time~Nitrogen+I(Nitrogen^2), data =LeafData)
fit2
```

```
##
## Call:
## lm(formula = log_burn_time ~ Nitrogen + I(Nitrogen^2), data = LeafData)
##
## Coefficients:
##   (Intercept)       Nitrogen  I(Nitrogen^2)
##        4.6985        -1.8525         0.1861
```

```
deviance(fit2)
```

```
## [1] 3.103197
```

Let D_2 denote the deviance of this quadratic linear model.

When $D_1 - D_2$ is large then the quadratic model is much better than the simple linear regression, i.e. the straight line model is significantly improved by adding a quadratic term.

Note the use of

```
I(Nitrogen^2)
```

in the model formula. This is needed to seperate the two terms Nitrogen and Nitrogen^2.

If we had used

```
lm(formula = log_burn_time ~ Nitrogen + Nitrogen^2, data=LeafData)
```

then we would be fitting the linear model

$$y_i = a + b(x_i + x_i^2) + \epsilon_i \qquad i = 1, \ldots, n.$$

instead (try it!).

We can also fit the null model

$$y_i = a + \epsilon_i \qquad i = 1, \ldots, n.$$

```
fit0 <- lm(log_burn_time~1, data=LeafData)
fit0
```

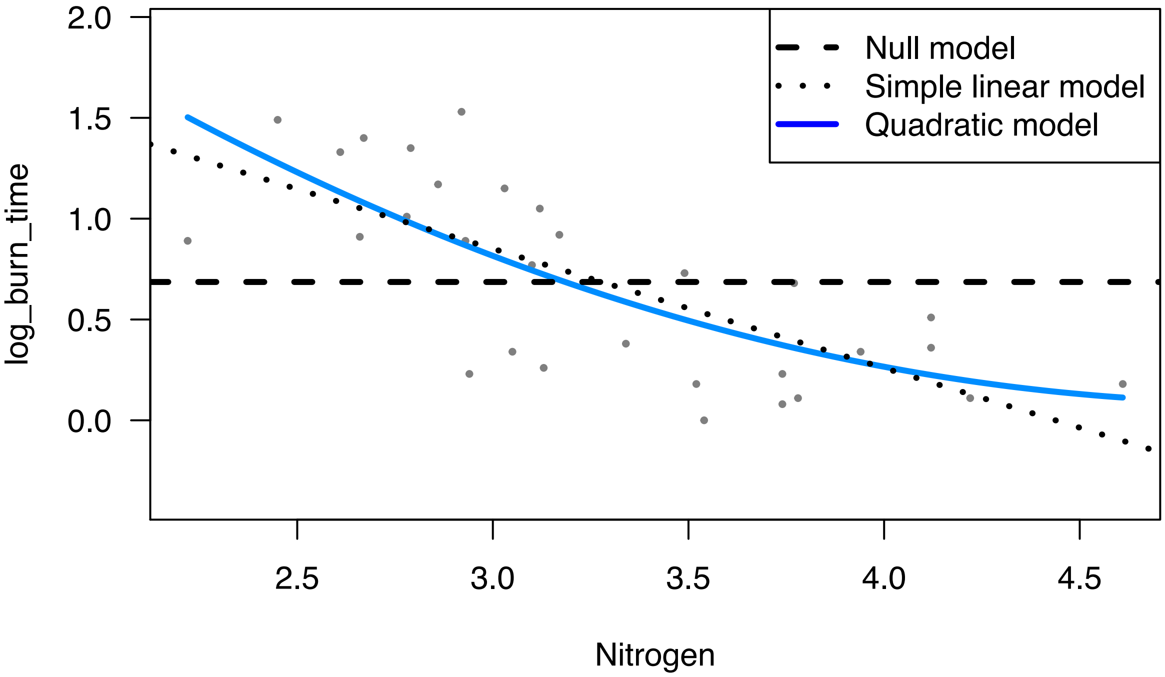
```
##
## Call:
## lm(formula = log_burn_time ~ 1, data = LeafData)
##
## Coefficients:
## (Intercept)
##       0.686
```

```
deviance(fit0)
```

```
## [1] 6.68952
```

We can plot these three models

```
library(visreg)
out = visreg(fit2, band=FALSE)
abline(fit1, lty=3, lwd=3)
abline(fit0, lty=2, lwd=3)
legend(x="topright",   lty = c(2, 3, 1), c("Null model", "Simple linear model", "Quadratic model"),  me
```



The plot and the deviances suggest that the linear model is probably sufficient. This is a subjective judgement based on eyeballing the data and fits - perhaps the most important aspect of model fitting. A more theoretical approach to model selection will be covered in Chapter 7.

Simple linear model revisited, this time with matrices

Consider the simple linear model considered above. The design matrix is

$$Z = \begin{bmatrix} 1 & x_{1\,1} \\ 1 & x_{2\,1} \\ \vdots & \vdots \\ 1 & x_{30\ 1} \end{bmatrix}.$$

We know that:

- The unbiased least squares estimator is

$$\hat{\beta} = (Z^T Z)^{-1} Z^T y = (2.63, -0.59)^T$$

- An unbiased estimator for σ^2 is

$$s^2 = \frac{1}{n-p}(y - Z\hat{\beta})^T (y - Z\hat{\beta}) = 0.12$$

- An estimate for $\mathrm{Var}(\hat{\beta})$ is given by

$$s^2 (Z^T Z)^{-1} = \begin{bmatrix} 0.130 & -0.04 \\ -0.04 & 0.012 \end{bmatrix}$$

These quantities are all available from R:

```r
coef(fit1)
```

```
## (Intercept)    Nitrogen 
##   2.6257040  -0.5916137
```

```r
fit1.sum <- summary(fit1)
fit1.sum$sigma^2
```

```
## [1] 0.1158397
```

```r
vcov(fit1)
```

```
##             (Intercept)   Nitrogen
## (Intercept)   0.1303385 -0.0385758
## Nitrogen     -0.0385758  0.0117657
```

The most useful R command for summarizing the model fit is

```r
summary(fit1)
```

```
## 
## Call:
## lm(formula = log_burn_time ~ Nitrogen, data = LeafData)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.65636 -0.27698  0.03712  0.27876  0.63181 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   2.6257     0.3610   7.273 6.44e-08 ***
## Nitrogen     -0.5916     0.1085  -5.454 8.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3404 on 28 degrees of freedom
## Multiple R-squared:  0.5151, Adjusted R-squared:  0.4978 
## F-statistic: 29.75 on 1 and 28 DF,  p-value: 8.025e-06
```

R^2 and adjusted R^2

Above we fitted the model

$$y_i = a + bx_i + \epsilon_i$$

and found that $s^2 = 0.12$ and $D = 3.24$.

For the null model

$$y_i = a + \epsilon_i$$

we find that $s^2(\text{null}) = 0.23$ and $D_0 = 6.69$.

Using R:

```
fit1.sum <- summary(fit1)
fit1.sum$r.squared  ## gives the R^2 value
```

```
## [1] 0.5151353
```

```
fit1.sum$adj.r.squared ## gives the adjusted R^2 value
```

```
## [1] 0.4978187
```

Next, define a new input variate w to be 30 independent observations from a $U(0, 1)$ distribution and fit the model

$$y = a + bx + cw + \epsilon.$$

For this model we get that

$$R^2 = 0.516 \qquad \text{and} \qquad R^2_{adj} = 0.480.$$

Notice that R^2 must improve when an input variate is added to the model. In this case, the input variate is unrelated to y and so R^2 only improves by a very small amount. However R^2_{adj} has gone down: it reflects the fact that this new input variate is not providing any useful information and is not worth including in the model.

This shows why the adjusted R-squared is useful.

Confidence intervals using R

```
confint(fit1)    # gives 95% CI for both parameters
```

```
##                  2.5 %     97.5 %
## (Intercept)  1.886179  3.3652287
## Nitrogen    -0.813804 -0.3694234
```

```
confint(fit1, level=0.99)    # gives 99% CI for both parameters
```

```
##                   0.5 %     99.5 %
## (Intercept)  1.6280992  3.6233088
## Nitrogen    -0.8913442 -0.2918831
```

```
confint(fit1, level=0.99, parm="Nitrogen")    ## supplies just the nitrogen CI.
```

```
##                0.5 %     99.5 %
## Nitrogen -0.8913442 -0.2918831
```