An introduction to linear models

April 9, 2015

Introduction

In this lab we will we will learn how to fit linear models in R. We will also cover some model selection techniques and see to check a model is a reasonable approximation of the process generating our data.

An introduction to linear models

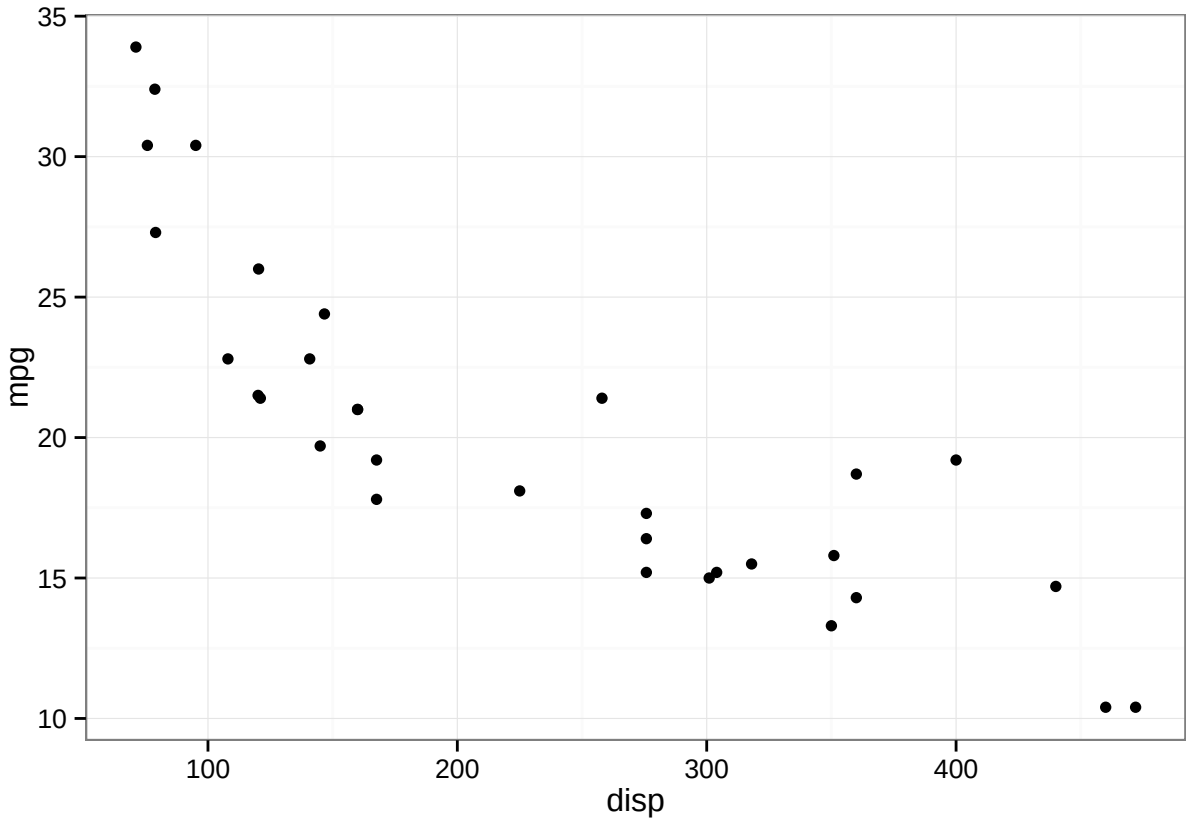
We will be analysing the `mtcars` dataset. For more information about the dataset run `?mtcars`

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

We are interested in modelling the relationship between engine displacement (`disp`) and fuel consumption (`mpg`).

```
library(ggplot2)
p1 <- ggplot(mtcars) + geom_point(aes(x = disp, y = mpg)) + theme_bw()
p1
```



The relationship does not look linear, so we might want to fit a model of the form:

$$y_i = \alpha + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$$

Here the y_i is fuel consumption for the ith car and x_i is the engine displacement. α, β_1 and β_2 are coefficients and ϵ is the error term.

This can be expressed in `R` relatively simply:

```
mpg_model <- lm(mpg ~ disp + I(disp^2), data = mtcars)
```

Why do you think the `I()` is needed?

Let's have a look at the `mpg_model`

```
mpg_model
```

```
##
## Call:
## lm(formula = mpg ~ disp + I(disp^2), data = mtcars)
##
## Coefficients:
## (Intercept)         disp    I(disp^2)
##  35.8286989   -0.1052732    0.0001255
```

```
names(mpg_model)
```

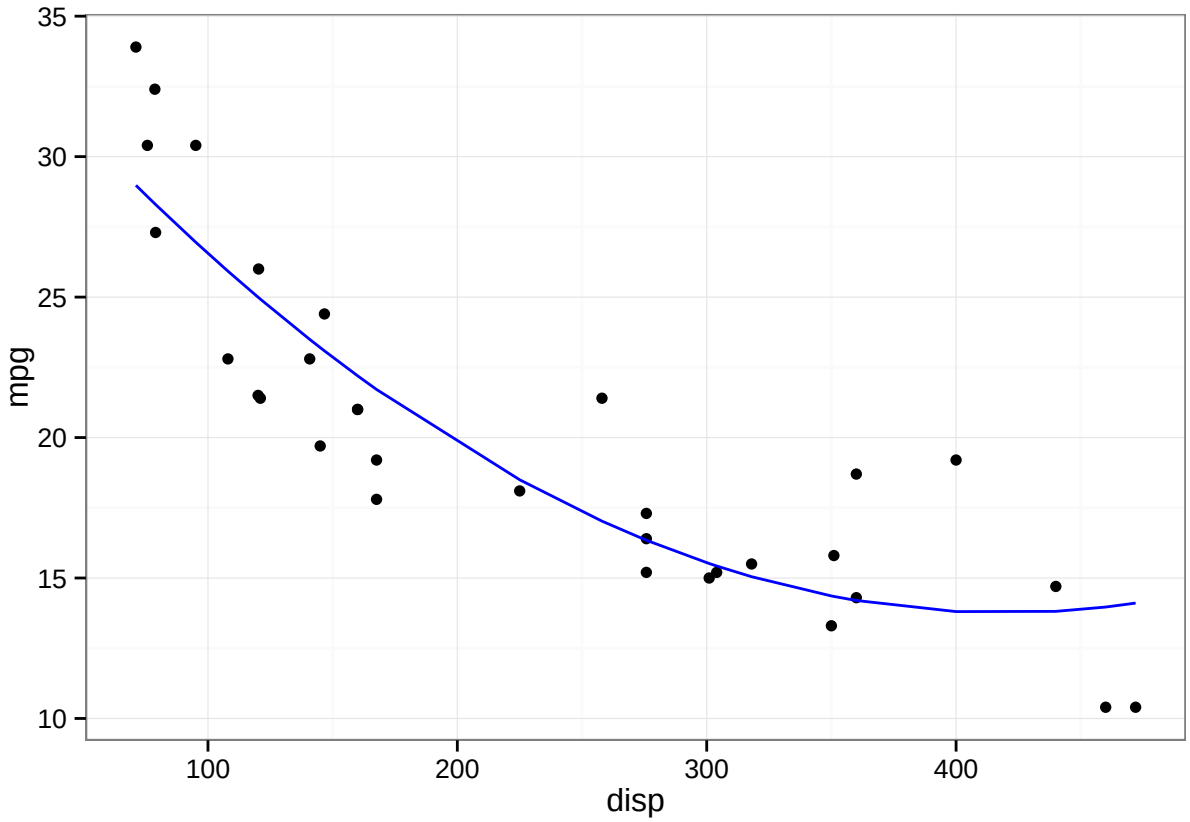
```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
mpg_model$coefficients
```

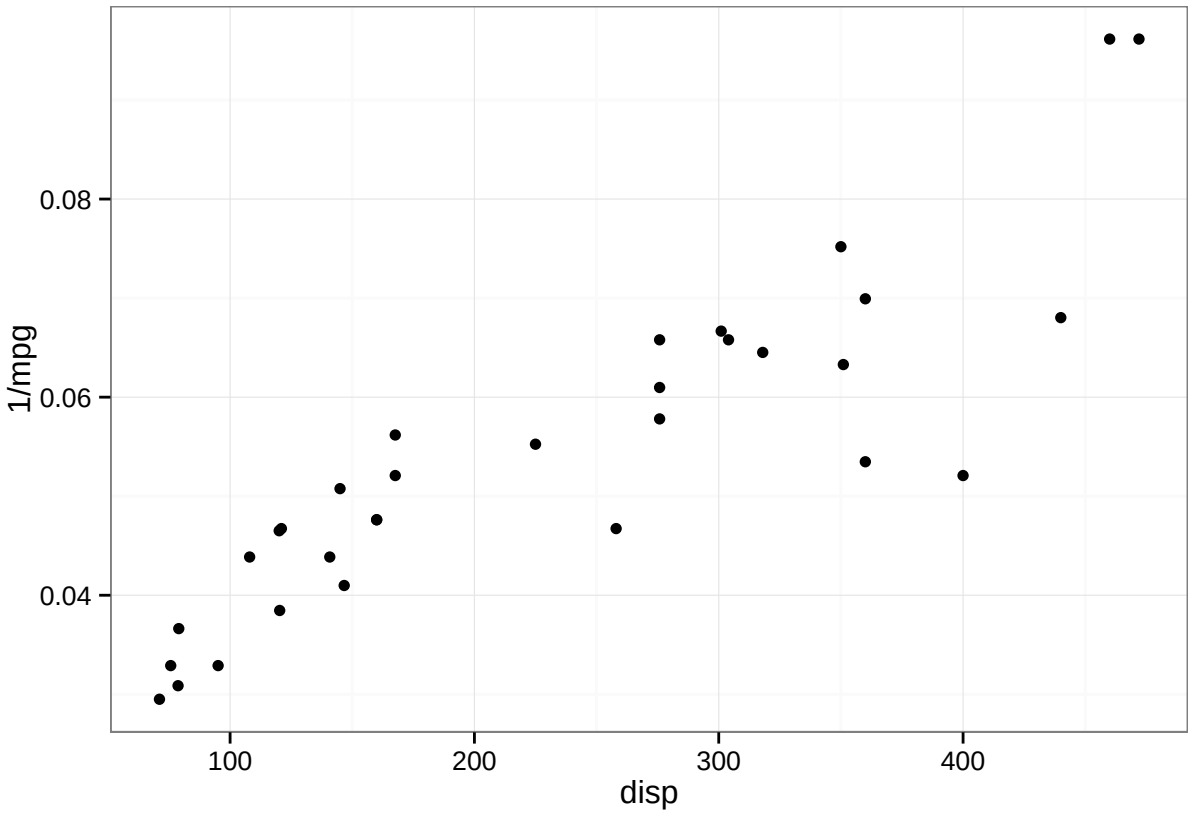
```
##   (Intercept)          disp     I(disp^2)
## 35.8286989277 -0.1052732424  0.0001255373
```

A natural next step is to plot the fitted values and see how well they aline with the truth:

```
mtcars$mpg_pred <- predict(mpg_model, newdata = mtcars)
p1 + geom_line(data = mtcars, aes(x = disp, y = mpg_pred), col = "blue")
```

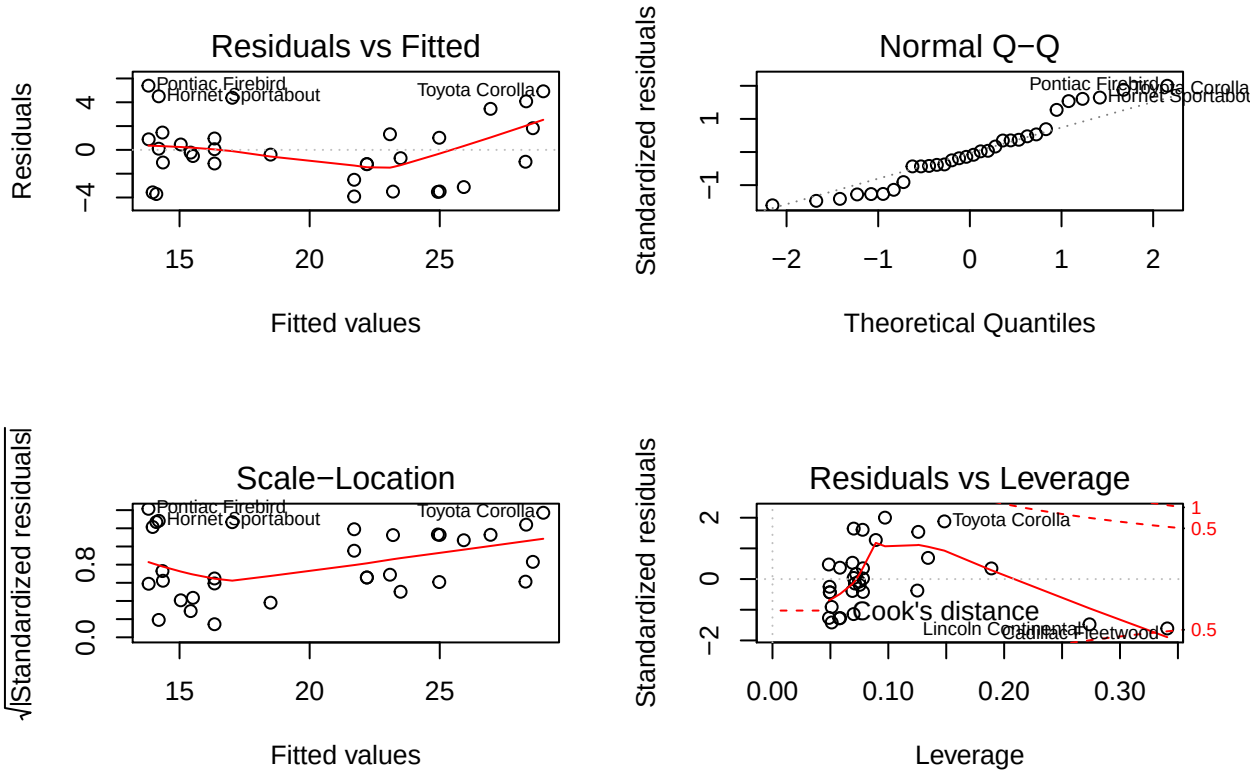


They seeem to be a reasonable match; however in practice fitting complicated models can be avoided by making a suitable transformation. For example if treat 1/mpg as the outcome, things look very linear.



In the real world it is important to check that your model describes reality or is at least a reasonable simulacrum of reality. R provides built in plots to check some aspects of the suitability of your model

```
par(mfrow = c(2,2))
plot(mpg_model)
```



Optional exercises

1. Read about diamonds dataset (`head(diamonds); ?diamonds`).
2. We are interested in using a modelling the relationship where the response is the price of the diamond and the covariates are the cut, color, clarity and carat weight of the dimaond.
3. Construct exploratory plots to see which terms might be needed in your linear model.
4. Fit a sensible linear model.
5. Look at the fitted coefficients and try to interpret them.
6. Check whether some of the assumptions of the linear model are met using the function `?plot.lm`.