Applied Machine Learning and Predictive Modelling 1 - Exercises

Nils Rechberger

2026-02-21

Series 1: Linear Models

In class we fitted a model to the "cats" dataset. You may remember that the interpretation of the intercept was somehow problematic. Let's get the data, visualise it and refit the model again.

```
d.cats <- read.csv(
    file = "/home/nils/dev/mscids-notes/fs26/mpm1/data/Cats.csv",
    header = TRUE,
    stringsAsFactors = TRUE
    )

str(d.cats)
```

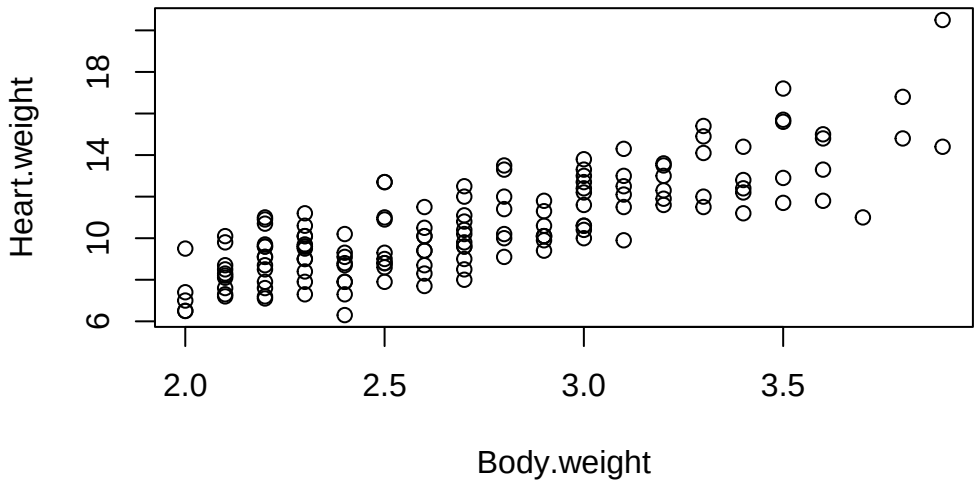
```
'data.frame':   144 obs. of  3 variables:
 $ Sex         : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
 $ Body.weight : num  2 2 2 2.1 2.1 2.1 2.1 2.1 2.1 2.1 ...
 $ Heart.weight: num  7 7.4 9.5 7.2 7.3 7.6 8.1 8.2 8.3 8.5 ...
```

```
head(d.cats)
```

```
  Sex Body.weight Heart.weight
1   F         2.0          7.0
2   F         2.0          7.4
3   F         2.0          9.5
4   F         2.1          7.2
5   F         2.1          7.3
6   F         2.1          7.6
```

Let's display the effect of Body.weight.

```
plot(Heart.weight ~ Body.weight, data = d.cats)
```



The first model we fitted was:

```
lm.cats <- lm(Heart.weight ~ Body.weight, data = d.cats)
```

The estimated coefficients of this model are:

```
coef(lm.cats)
```

```
(Intercept) Body.weight
 -0.3566624   4.0340627
```

As mentioned in class, the correct interpretation of the intercept is "a cat with zero body.weight, is expected to have a heart weight of -0.36". It is obviously nonsensical for two reasons:

1. There is no cat of zero body weight
2. A negative prediction for the response variable heart.weight is impossible in reality.

Questions

Question 1

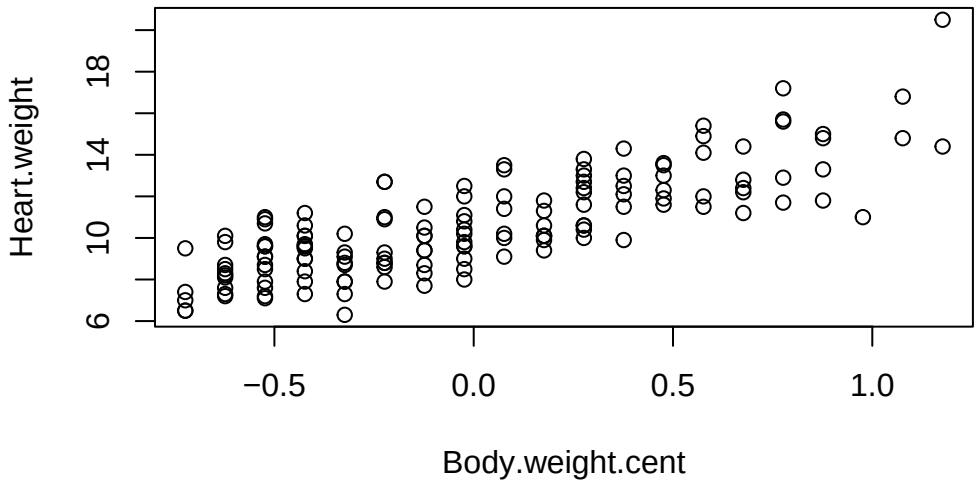
How would you proceed to simplify the interpretation of the intercept in this model? Hint: try to manipulate the predictor body.weight (e.g. by centering it).

Answer

By centering the variable `Body.weight`, we can get a interpretable intercept.

$$\text{Body.weight.cent} = \text{Body.weight} - \bar{x}$$

```
d.cats$Body.weight.cent <- d.cats$Body.weight - mean(d.cats$Body.weight)
plot(Heart.weight ~ Body.weight.cent, data = d.cats)
```



Question 2

Let's turn our attention to the model that contains sex too, but no interaction. Reparametrise this model such that "M" is the reference. Hint: use the relevel() function.

Answer

First we check the current level of `Sex`.

```
levels(d.cats$Sex)
```

```
[1] "F" "M"
```

We can see that `F` ist the reference level. To change that:

```
d.cats$Sex <- relevel(x = d.cats$Sex, ref = "M")
levels(d.cats$Sex)
```

```
[1] "M" "F"
```

Now we need to refit our model:

```
lm.cats.relevelled <- lm(Heart.weight ~ Body.weight + Sex, data = d.cats)
coef(lm.cats.relevelled)
```

```
(Intercept) Body.weight        SexF
-0.49704946  4.07576892  0.08209684
```

Question 3

When the predictor sex was added to the model, the estimated coefficient for 'body.weight' slightly changed. Refit both models, show the estimated coefficients and write a sentence that correctly describes their "biological interpretation" of the Body.weight predictor in each model.

Answer

```
cat("Ceoff for without sex:", coef(lm.cats), "\n")
```

```
Ceoff for without sex: -0.3566624 4.034063
```

```
cat("Ceoff for with sex:", coef(lm.cats.relevelled))
```

```
Ceoff for with sex: -0.4970495 4.075769 0.08209684
```

- First model: By increasing by one unit body weight, we expect an increase of 4.03 in the response variable.
- Second model: By increasing by one unit body weight, while keeping all the other predictors fixed, we expect an increase of 4.08 in the response variable.

Question 4

This time we assume that Body.weight was not provided as a continuous variable, but rather as a categorical one. We do this by creating four classes with similar size. With this purpose in mind, we use the `quantil()` and `cut()` functions.

```
quantiles.Body.weight <- quantile(d.cats$Body.weight)
quantiles.Body.weight
```

```
   0%   25%   50%   75%  100%
2.000 2.300 2.700 3.025 3.900
```

```
d.cats$Body.weight.Class <- cut(
    x = d.cats$Body.weight,
    breaks = quantiles.Body.weight,
    include.lowest = TRUE
    )
```

Let's check how many observations are present in each class.

```
table(d.cats$Body.weight.Class)
```

```
   [2,2.3]  (2.3,2.7] (2.7,3.02] (3.02,3.9]
        42         40         26         36
```

Fit a model with Sex and Body.weight.Class and compute a p-value for both predictors.

Answer

```r
lm.cats.bodyClass <- lm(
    Heart.weight ~ Sex + Body.weight.Class,
    data = d.cats)

drop1(lm.cats.bodyClass, test = "F")
```

```
Single term deletions

Model:
Heart.weight ~ Sex + Body.weight.Class
                  Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>                         354.77 139.84
Sex                1      0.30 355.06 137.96  0.1166 0.7333
Body.weight.Class  3    350.49 705.26 232.78 45.7751 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Body.weight.Class seems to play a relevant role, while Sex does not. This is in full agreement with the model we have seen last week where Body.weight was taken as a continuous predictor.

Question 5

Now run some contrasts to see whether all pair of levels of the Body.weight.Class predictor differ from each other. Comment on the results.

Answer

```r
require(multcomp)
```

```
Loading required package: multcomp

Loading required package: mvtnorm

Loading required package: survival

Loading required package: TH.data
```

```
Loading required package: MASS


Attaching package: 'TH.data'

The following object is masked from 'package:MASS':

    geyser
```

```r
glht.1 <- glht(lm.cats.bodyClass, linfct = mcp(Body.weight.Class = "Tukey"))
##
summary(glht.1)
```

```
     Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: lm(formula = Heart.weight ~ Sex + Body.weight.Class, data = d.cats)

Linear Hypotheses:
                             Estimate Std. Error t value Pr(>|t|)
(2.3,2.7] - [2,2.3] == 0       0.7155     0.3813   1.876 0.241722
(2.7,3.02] - [2,2.3] == 0      2.4273     0.4382   5.539  < 1e-04 ***
(3.02,3.9] - [2,2.3] == 0      4.6086     0.4401  10.473  < 1e-04 ***
(2.7,3.02] - (2.3,2.7] == 0    1.7118     0.4042   4.235 0.000222 ***
(3.02,3.9] - (2.3,2.7] == 0    3.8932     0.3816  10.202  < 1e-04 ***
(3.02,3.9] - (2.7,3.02] == 0   2.1814     0.4166   5.236  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Question 6

Ask generative AI to provide the interpretation of - the coefficients from a linear model - the p-values from a linear model

and compare it with the definitions you find in the lecture materials. Do you think they are different in any way? Which one is easier for you to understand?

Answer

Prompt:

```
Provide a interpretation of the coefficients and the p-values from a linear model
```

Note: Used Gemeni 3 Fast

Answer (excerpt) :

```
Interpreting a linear model is all about understanding the "story" the data is telling. When
```

The general statement is accurate, although dividing p-values into “significant” and “not significant” is very bad practice.