Series 1: Applied Machine Learning and Predictive Modelling 1

Dr. Luisa Barbanti and Dr. Matteo Tanadini

Fall Semester 2025 (HSLU)

`Solutions`

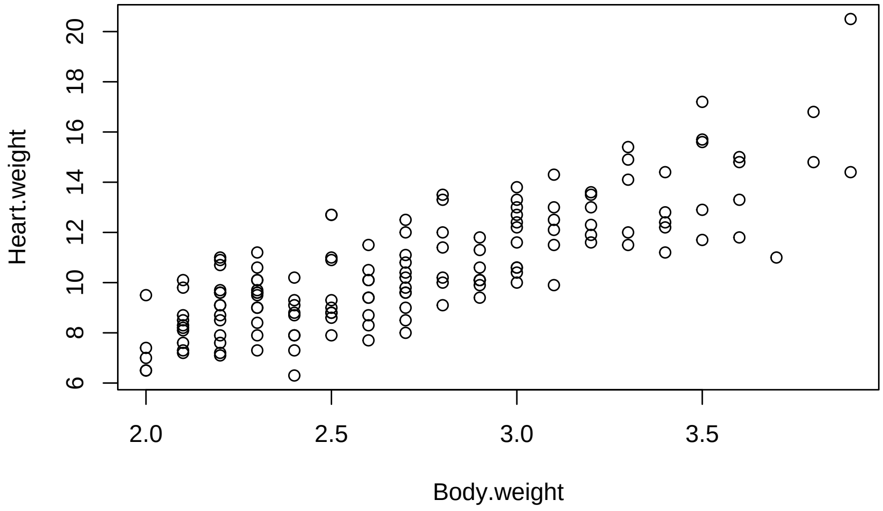
Exercise

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.

```
## (results are hidden)
d.cats <- read.csv("../../../Datasets/Cats.csv",
                   header = TRUE, stringsAsFactors = TRUE)
##
str(d.cats)
head(d.cats)
```

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
      -0.36        4.03
```

As mentioned in the class, the correct intrepretation of the intercept is “a cat with zero bodyweight, 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 and 2) a negative prediction for the response variable “heart weight” is impossible in reality.

Question

How would you proceed to improve the intrepretability of the intercept in this model? Hint: try to manipulate the predictor “body weight” (e.g. by centring it).

Answer

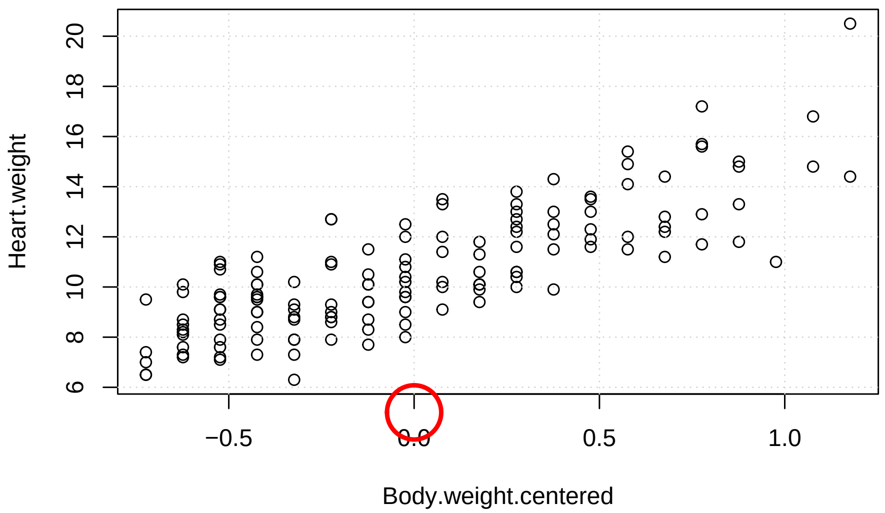
We create a new predictor named `Body.weight.centered` by substracting the mean to the `Body.weight` predictor.

```
mean.Body.weight <- mean(d.cats$Body.weight)
mean.Body.weight
```

```
[1] 2.7
```

```
##
d.cats$Body.weight.centered <- d.cats$Body.weight - mean.Body.weight
```

Let’s plot the data again and refit the linear model.



Let’s see how the coefficient for the intercept is affected.

```
lm.cats.BIS <-lm(Heart.weight ~ Body.weight.centered, data = d.cats)
coef(lm.cats.BIS)["(Intercept)"]
```

```
(Intercept) 
         11 
```

The interpretation of the intercept becomes: “a cat of average body weight (i.e. 2.72) is expected to have an hearth weighing 11”[1]. This makes much more sense.

Let’s double check that the centering of this predictor did not change anything else in the model.

```r
summary(lm.cats.BIS)
```

```


Call:
lm(formula = Heart.weight ~ Body.weight.centered, data = d.cats)

Residuals:
   Min     1Q Median     3Q    Max 
-3.569 -0.963 -0.092  1.043  5.124 

Coefficients:
                     Estimate Std. Error t value
(Intercept)            10.631      0.121    87.8
Body.weight.centered    4.034      0.250    16.1
                     Pr(>|t|)    
(Intercept)            <2e-16 ***
Body.weight.centered   <2e-16 ***
---
Signif. codes:  
0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 1.4 on 142 degrees of freedom
Multiple R-squared:  0.647, Adjusted R-squared:  0.644 
F-statistic:  260 on 1 and 142 DF,  p-value: <2e-16
```

Indeed, the R^2, the estimated coefficient for `Body.weight` and the associated test statistics are the same. So, by recentering the continuous predictor `Body.weight` we obtaine a meaningful interpretation of the intercept, without affecting any other parameter in the model.

Note that there are other ways to obtain alternative parametrisations of this model such that the intercept has a meaningful biological interpretation. One possibility is to substract a given value (not necessarily the average).

Question

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

Answer

Let’s set `"M"` as the reference level.

[1] Unfortunately, it is not known what the unit measures in this data set are. This stresses the importance of well documenting analyses and data.

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

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

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

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

Let's fit both models.

```r
## Female is reference
lm.cats.2 <- lm(Heart.weight ~ Body.weight + Sex, data = d.cats)
coef(lm.cats.2)
```

```
(Intercept) Body.weight        SexM
     -0.415       4.076      -0.082
```

```r
##
## Male is reference
lm.cats.2.BIS <- lm(Heart.weight ~ Body.weight + Sex.relevel, data = d.cats)
coef(lm.cats.2.BIS)
```

```
 (Intercept)  Body.weight Sex.relevelF
      -0.497        4.076        0.082
```

As expected, the gender effect is the same but of different sign (i.e. ±0.082). The `(Intercept)` estimate changed as it now represents the intercept for males. The slope for "body weight" remains unaffected.

Question

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

Let's look at the coefficients of these two models.

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

```
(Intercept) Body.weight
      -0.36        4.03
```

```r
coef(lm.cats.2)
```

```
(Intercept) Body.weight        SexM
     -0.415       4.076      -0.082
```

Let's start with the interpretation of `Body.weight` in `lm.cats` model: "by increasing by one unit body weight, we expect an increase of 4.03 in the response variable."

Let's turn our attention to the model that contains gender as well. The coefficient for `Body.weight` now represents the effect of this variable when you control for cat gender.

Indeed, the correct intrepration for `Body.weight` in the `lm.cats.2` model is: "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."

So the only difference is that we assume that all the other predictors are fixed and not varied, but that only the predictor of interest is increased by one.

Question

This time we assume that `Body.weight` was not provided as a continuous variable, but rather as a categorical variable. Let's create this situation by creating four classes with similar size. To do that, we use the quantiles and the `cut()` function.

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

```
  0%  25%  50%  75% 100% 
 2.0  2.3  2.7  3.0  3.9 
```

```
##
d.cats$Body.weight.Class <- cut(d.cats$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

Let's fit the model:

```
mod.cats.1 <- lm(Heart.weight ~ Sex + Body.weight.Class, data = d.cats)
```

To test these two predictors we can use the `drop1()` function.

```
drop1(mod.cats.1, test = "F")
```

```
Single term deletions

Model:
Heart.weight ~ Sex + Body.weight.Class
```

```
                  Df Sum of Sq RSS AIC F value Pr(>F)
<none>                         355 140
Sex                1         0 355 138    0.12   0.73
Body.weight.Class  3       350 705 233   45.78 <2e-16

<none>
Sex
Body.weight.Class ***
---
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.

Side note: In general, by making discrete a continuous variable we loose information. Here we do that for didactic reasons. Nevertheless, remember that we may want to make a continuous variable discrete to be able to fit an interaction between two variables that were originally continuous (not the case here). To be entirely correct, you can fit a model with an interaction between two continuous variables, the only issue being the interpretation of the coefficients. When the interaction is done between two continuous variables the interpretation of the coefficient is fairly difficult. On the other hand the interpretation of the coefficients of an interaction between categorical-continuous is fairly simple.

Note that `Sex` is a dummy variable (i.e. a categorical variable with two levels), therefore, we could compute a p-value via the `summary()` function.

```
summary(mod.cats.1)
```

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

Residuals:
   Min     1Q Median     3Q    Max
-3.589 -1.194 -0.196  0.939  7.011

Coefficients:
                            Estimate Std. Error
(Intercept)                    8.761      0.266
SexM                           0.119      0.349
Body.weight.Class(2.3,2.7]     0.715      0.381
Body.weight.Class(2.7,3.02]    2.427      0.438
Body.weight.Class(3.02,3.9]    4.609      0.440
                            t value Pr(>|t|)
(Intercept)                   32.95  < 2e-16 ***
SexM                           0.34    0.733
Body.weight.Class(2.3,2.7]     1.88    0.063 .
Body.weight.Class(2.7,3.02]    5.54  1.5e-07 ***
Body.weight.Class(3.02,3.9]   10.47  < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.6 on 139 degrees of freedom
Multiple R-squared:  0.581, Adjusted R-squared:  0.569
F-statistic: 48.3 on 4 and 139 DF,  p-value: <2e-16
```

The p-value for `SexM` here is identical to the one obtain with the `drop1()` function. Obviously, we do not obtain a single p-value for `Body.weight.Class` because it is a factor with more than two levels. To obtain a p-value for the variable `Body.weight.Class` we must run an F-test via the `anova()` or the `drop1()` functions.

Question

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

To do that we load the `{multcomp}` package and use the `glht()` function.

```
require(multcomp)
glht.1 <- glht(mod.cats.1, 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
(2.3,2.7] - [2,2.3] == 0        0.715      0.381
(2.7,3.02] - [2,2.3] == 0       2.427      0.438
(3.02,3.9] - [2,2.3] == 0       4.609      0.440
(2.7,3.02] - (2.3,2.7] == 0     1.712      0.404
(3.02,3.9] - (2.3,2.7] == 0     3.893      0.382
(3.02,3.9] - (2.7,3.02] == 0    2.181      0.417
                             t value Pr(>|t|)
(2.3,2.7] - [2,2.3] == 0        1.88     0.24
(2.7,3.02] - [2,2.3] == 0       5.54   <0.001 ***
(3.02,3.9] - [2,2.3] == 0      10.47   <0.001 ***
(2.7,3.02] - (2.3,2.7] == 0     4.23   <0.001 ***
(3.02,3.9] - (2.3,2.7] == 0    10.20   <0.001 ***
(3.02,3.9] - (2.7,3.02] == 0    5.24   <0.001 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Apparently, all pairwise comparisons, but one are significant. In particular, the two smallest classes do not differ from each other.

Question

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?