Lab 2 - linear regression

Vittorio Zampinetti[*] Mauro Gasparini[†]

2022-11-17

Linear regression lab

This lab focuses on linear models and on the R functions dedicated to fitted linear models inspection.

For the following tasks, we will use a real dataset, `insulate.csv`, which features two predictors and one response variable (fuel consumption). You can read some information inside the `insulate.names` text file.

EDA (20 min)

Start by loading the `insulate.csv` data in the R environment, making sure that the type of the parsed variables is correctly inferred, then explore it using the tools seen in the lectures (e.g. dimension, summary, plots, etc.).

You might answer these questions:

- How many observations does the dataset provide?
- What are the predictors? Are they qualitative or quantitative?
- What is the relationship between the response variable and the two predictors taken individually.
    - Draw some *useful* plot and interpret what you see.

Fitting a linear model (15 min)

Now use the `lm()` function to fit two linear models: one which only uses the temperature as predictor, and one with both predictors.

- Which one is better? Why?
- Plot both regression lines (you can use any plotting tool that you prefer).

Interaction (10 min)

In this setting, we are mostly interested in whether we should include the interaction between the two predictors or not. If the effects of the predictors are additive, we don't need interaction.

- Are the two effects additive? The next task can help in answering this question.
- Draw a plot showing the consumption trend against the temperature, separately for before and after insulation. You may (or may not) use the following code as template, which is enough to fulfill the task after replacing the ellipsis with the proper function arguments:

```r
insulate %>% # dataset
  ggplot(...) + # main ggplot call
  geom_point(...) + # scatter plot
  geom_smooth(...) # trend line
```

[*] Politecnico di Torino, vittorio.zampinetti@polito.it

[†] Politecnico di Torino, mauro.gasparini@polito.it

Hint: remember that with ggplot, in order to differentiate between classes (i.e. *before* and *after*) in the same plot, you can select the class to be interpreted as color encoding for the points/line: `aes(..., color = when)`.

- Fit a linear model which includes interaction between `temp` and `when`.
- How many additional parameters does the model have to estimate?

AIC and model selection (10 min)

Looking at the R-squared index, the model with interaction seems to perform a bit better. However, this doesn't mean that the interaction model is a better choice.

- Are the additive model (`when + temp`) and the interaction model nested one into the other?
- Run an ANOVA test comparing the two models, analyze the result, draw a conclusion.

Another criterion used for model selection is the Akaike Information Criterion, which is defined as

$$\text{AIC} = 2k - 2\ln(\hat{L})$$

where $\hat{L}$ is the maximized likelihood of the data i.e. $p(y|\hat{\beta})$. The AIC takes into consideration the model complexity (number of predictors considered), favoring small models over the bigger ones.

- Find the AIC index using the dedicated function `AIC()`. Which model is better according to the AIC? Why?

Prediction intervals (35 min)

Imagine we have a new unseen (*future*) row of observed measures x_f, with no true response observation (Y_f is unknown).

We are interested in making inference on the mean $x_f\beta$. As you've seen in the theory lectures, this is a particular case of the inference of $C\beta$. Being just one row, the F-type intervals can be translated to T-type intervals:

$$F_\alpha(1, n-p) = t^2_{\frac{\alpha}{2}}(n-p)$$

These are standard confidence intervals, and you can find them with `predict.lm(..., interval = "confidence")`.

- Find a prediction for the following additional rows using the best model you found so far, together with their confidence intervals.

```
when,temp
before,5.8
before,-1.0
after,4.8
after,9.8
```

With new observations, instead of making inference on the coefficients β, we can make inference on the unseen response variable Y_f. In this case we are not doing either estimation or testing, but rather just *prediction*. In *machine learning* the focus of inference is prediction: neural networks, for instance, have a large set of parameters which are learnt automatically from the data, but their values are most of the times non-interpretable, thus not useful per se, while the output of interest is $\hat{Y}_f$, eventually together with its uncertainty.

Inference on Y_f can be derived similarly to how you did with confidence intervals. Formally (but not too much in detail):

$$Y_f \sim \mathcal{N}(x_f'\beta, \sigma^2)$$

independent from $Y_1, ..., Y_n$. Under the homogeneity assumption of the future response with respect to the past,

$$x_f'\hat{\beta} = \hat{Y} \sim \mathcal{N}(x_f'\beta, \sigma^2 x_f'(X'X)^{-1}x_f)\,.$$

This leads to

$$Y_f - \hat{Y}_f \sim \mathcal{N}(0, \sigma(1 + x_f'(X'X)^{-1}x_f))\,,$$

which gives us the predictio intervals:

$$\hat{Y}_f \pm t_{\frac{\alpha}{2}}(n-p)\sqrt{\text{MSR}(1 + x_f'(X'X)^{-1}x_f)}\,.$$

- Using this formula, compute the prediction intervals for *one* of the four new future observations written above (from scratch).

  Hint: We've seen how to compute the MSR (or RMS) in the first linear regression R lecture
- Check your computations using `predict.lm(..., interval = "prediction")`
- Is the interval center the same as for the confidence intervals?
- Is it wider or more narrow than the CI? Can you tell why (intuitively)?