Logistic regression and predictive models

Biomedical Data Science

Marco Colombo

Lecture 3, 2017/2018

Case-control studies

A *case-control study* is a type of epidemiological study design where two groups of individuals are compared:

1. Cases are selected on the bases of their disease outcome, independently of whether they had or had not an exposure to the risk factor(s) of interest.

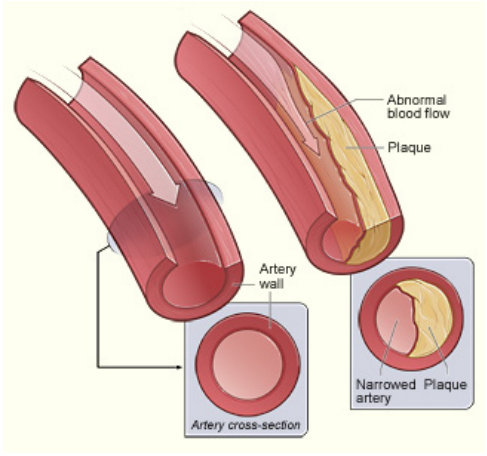
2. Controls are selected on the bases of not having that disease outcome (so to act as a comparison group) again independently of a specific exposure.

3. Data about exposures (covariates) for both cases and controls are collected retrospectively.

The objective is to identify the *risk factors* for the outcome of interest by comparing the proportions in which the risk factors are present in the two groups.

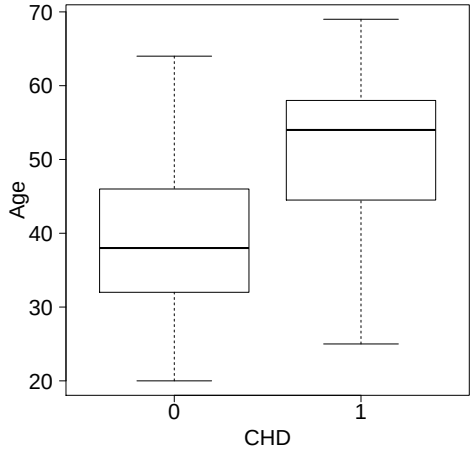
Example: coronary heart disease I

Let's consider a case-control study of coronary heart disease: the outcome variable is coded as 1 for cases (patients who have CHD), and 0 for controls (healthy people).



Example: coronary heart disease II

	AGE	CHD
1	20	0
2	23	0
3	24	0
4	25	0
5	25	1
6	...	
96	63	1
97	64	0
98	64	1
99	65	1
100	69	1



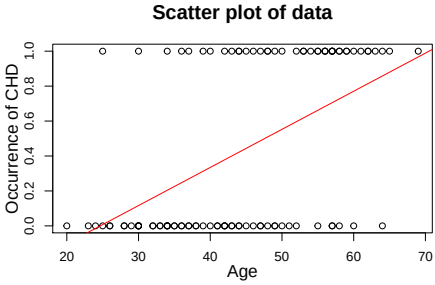
Example: coronary heart disease III

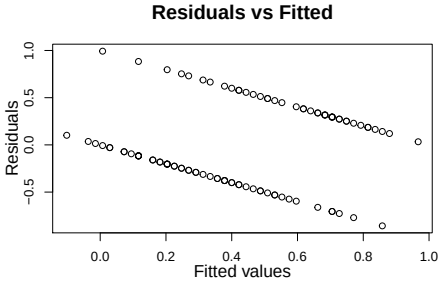
After fitting a linear regression model, can we say that age is associated with occurrence of CHD?

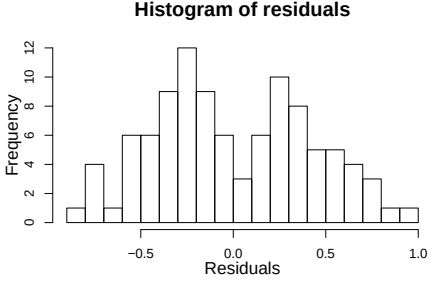
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.538	0.169	-3.187	0.002
AGE	0.022	0.004	5.929	0.000

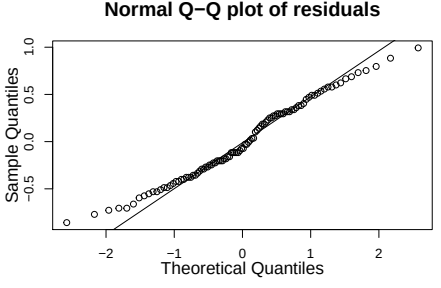
According to the usual interpretation, the regression coefficient for age would indicate that on average, each additional year of age brings us 0.022 units closer to being considered a case.

What does the intercept term tell us? Do the assumptions for a linear regression model hold in this case?


Scatter plot of data
Occurrence of CHD
Age
0.0
0.2
0.4
0.6
0.8
1.0
20
30
40
50
60
70


Residuals vs Fitted
Residuals
Fitted values
−0.5
0.0
0.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of residuals
Frequency
Residuals
0
2
4
6
8
10
12
−0.5
0.0
0.5
1.0


Normal Q−Q plot of residuals
Sample Quantiles
Theoretical Quantiles
−0.5
0.0
0.5
1.0
−2
−1
0
1
2

Generalized linear models

Linear regression can be seen as a special case of the *generalized linear model*:

$$\begin{aligned} g(y) &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon \\ &= X\beta + \varepsilon, \end{aligned}$$

where $g(y)$ (called the *link function*) is a transformation of the outcome that produces a linear relationship with the predictors.

For linear regression, the link function is $g(y) = y$ (identity link).

Other choices of the link function $g(y)$ lead to models that can be used to describe different phenomena or accommodate different types of outcomes (binary, counts, rates, survival times).

Binary outcomes

In the presence of a binary outcome $y_i \in \{0, 1\}$, we are interested in modelling what is the probability p that an individual is classified into the highest category (having an event, being a case).

In this setting, a linear regression model is not appropriate anymore:

- errors are not normally distributed and are not homoscedastic
- predicted values $\hat{y}$ may lie outside of the $(0, 1)$ interval, so they cannot be interpreted as probabilities

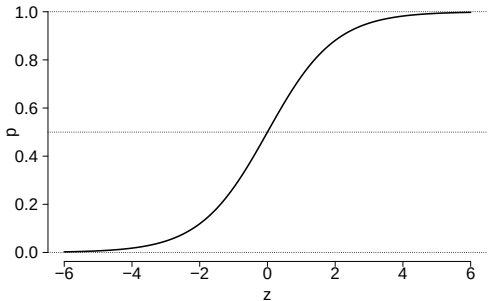
In terms of generalized linear models, we need to find a link function $g(y)$ that allows to restore the validity of those assumptions.

Logit

The *logistic function*

$$p = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

has the property of mapping any real number z to a value $p \in (0, 1)$.



Applying the logistic function to $z = X\beta$ and rearranging, we obtain the *logit* tranformation:

$$\log\left(\frac{p}{1-p}\right) = \mathit{logit}(p) = X\beta$$

Logistic regression

For a binary outcome we use the *logistic regression* model:

$$\mathit{logit}(p) = \log\left(\frac{p}{1-p}\right) = X\beta + \varepsilon$$

which is a generalized linear model with the logit link.

Note that despite containing the term *regression* in its name, logistic regression is used to solve a *classification problem*: from the fitted model we can work out what is the probability that an observation is classified in the higher group.

Log-odds

The term $\log(\frac{p}{1-p})$ represents the *log-odds* of an event having probability p.

p	odds $(\frac{p}{1-p})$	log-odds
0.5	$0.5 \,/\, 0.5 = 1$	0
0.2	$0.2 \,/\, 0.8 = 0.25$	-1.39
0.1	$0.1 \,/\, 0.9 = 0.11$	-2.20
0.9	$0.9 \,/\, 0.1 = 9$	2.20

The sign of the log-odds allows us infer whether an event is more likely (log-odds > 0) or less likely (log-odds < 0) than an equally likely event (log-odds $= 0$).

The magnitude of the log-odds describes how more/less likely an event is.

Interpretation I

What is the meaning of a coefficient in a logistic regression?

Suppose we fitted the following model:

$$logit(p) = \hat{\beta}_0 + \hat{\beta}_1 X_1$$

What happens when we change X_1 by one unit?

$$logit(p') = \hat{\beta}_0 + \hat{\beta}_1 (X_1 + 1)$$

The overall change is

$$logit(p') - logit(p) = \hat{\beta}_1$$

Interpretation II

Considering the properties of logarithms:

$$\hat{\beta}_1 = logit(p') - logit(p) = \log\left(\frac{p'}{1-p'}\right) - \log\left(\frac{p}{1-p}\right) = \log\left(\frac{\frac{p'}{1-p'}}{\frac{p}{1-p}}\right)$$

Therefore each logistic regression coefficient represents a *log-odds ratio*.

A simple exponentiation allows us to retrieve the *odds ratio*:

$$e^{\hat{\beta}_1} = \frac{\frac{p'}{1-p'}}{\frac{p}{1-p}} = \frac{\text{odds}'}{\text{odds}}$$

Odds ratios

An *odds ratio* describes how the odds for an event (such as having a disease) change with a 1 unit increase in that variable (holding all other variables constant).

This is easier to understand for a binary predictor:

	Exposed	Not exposed	Total
Case	a	b	a + b
Control	c	d	c + d
Total	a + c	b + d	n

- Odds of being a case in the exposed group: a/c
- Odds of being a case in the unexposed group: b/d

The odds ratio $\text{OR} = \frac{a/c}{b/d}$ represents the increased ($\text{OR} > 1$) or decreased ($\text{OR} < 1$) odds of being a case associated to having that exposure.

Example: odds ratios

A case-control study includes 1327 women aged 50–81 with hip fractures as well as 3262 randomly selected women in the same age range. Some women might be taking hormone replacement therapy (HRT), others not. Does taking HRT affect the probability of having a hip fracture?

	HRT	No HRT	Total
Case	40	1287	1327
Control	239	3023	3262
Total	279	4310	4589

- Odds of hip fracture for HRT users: $40/239 = 0.167$
- Odds of hip fracture for non-HRT users: $1287/3023 = 0.426$
- Odds ratio: $0.167/0.426 = 0.39$

Therefore HRT has a protective effect with respect to hip fractures: the treatment reduces the odds of a fracture by 61% $(1 - 0.39)$.

Example: coronary heart disease

Let's revisit the CHD example: now we fit a logistic regression model.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-5.310	1.1300	-4.68	2.82e-06
AGE	0.111	0.0241	4.61	4.02e-06

The log-odds ratio for age (0.111) implies a direct relationship between age and probability of disease.

The odds ratio is $e^{0.111} = 1.12$: each additional year of age increases the odds by 12%.

The likelihood function

In linear regression we followed the linear least squares approach to estimate the regression coefficients. Given the nonlinearity of the link function, for logistic regression we need to adopt a more general method: *maximum likelihood estimation*.

- Find the setting of β for which the *likelihood function* is maximized:

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} = \prod_{i \in \text{case}} p_i \prod_{i \in \text{ctrl}} (1-p_i),$$

where $p_i = \frac{1}{1+e^{-X\beta}}$ is the computed probability for individual i, and $y_i \in \{0, 1\}$.

The optimization problem is solved numerically by using an iterative gradient method, as there is no close-form solution.

Log-likelihood

Instead of dealing with the likelihood function, we generally use the *log-likelihood*:

$$\log \mathcal{L}(\beta) = \sum_{i \in \text{case}} \log p_i + \sum_{i \in \text{ctrl}} \log(1 - p_i)$$

Maximizing the likelihood (or the log-likelihood) favours settings of β that produce $p_i \to 1$ if $y_i = 1$, and $p_i \to 0$ if $y_i = 0$.

Operating with the log-likelihood has the advantage of turning multiplications into additions and ratios into subtractions.

The log-likelihood is an essential quantity to evaluate the goodness-of-fit of a model and a tool for model comparison.

Likelihood ratio test

Models are *nested* when one is a special case of the other:

$$\text{e.g.} \quad M_1 = \beta_0 + \beta_1 X_1 \quad \text{is nested in} \quad M_2 = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

When comparing two nested models, the one with more parameters will always fit at least as well as the other, so its likelihood will be at least as large.

The *likelihood ratio test* allows to compare two nested models by taking the ratio of their likelihoods. The test statistic

$$LRT = -2 \log \frac{\mathcal{L}(M_1)}{\mathcal{L}(M_2)} = 2\Big(\log \mathcal{L}(M_2) - \log \mathcal{L}(M_1) \Big)$$

is distributed approximately as a χ^2 with degrees of freedom equal to the difference in number of parameters between the two models.

In this test the null hypothesis is that the smaller model is true, while the alternative hypothesis is that the larger model is true.

Deviance

The *deviance* is a statistic of model fitness which compares the log-likelihood of the fitted model to the log-likelihood of the fully saturated model (one that has n parameters and therefore fits all data points exactly):

$$D = 2\left(\log \mathcal{L}^* - \log \mathcal{L}(\hat{\beta})\right)$$

The *null deviance* is the deviance of a model that only includes the intercept term: such model assigns the same probability (the proportion of cases) to all observations:

$$D_0 = 2\left(\log \mathcal{L}^* - \log \mathcal{L}(\hat{\beta}_0)\right)$$

In logistic regression $\log \mathcal{L}^* = 0$, while in linear regression $\log \mathcal{L}^* = \text{const}$, so you'll often see the deviance defined simply as $D = -2 \log \mathcal{L}(\hat{\beta})$.

AIC and BIC

The *Akaike information criterion* (AIC) is a measure used in model comparison that accounts for model complexity:

$$AIC = 2k - 2\log \mathcal{L}(\hat{\beta}),$$

where k is the number of parameters estimated in the model.

When comparing models, the one with lowest AIC is to be preferred.

Note that the AIC can be used also for non-nested models, as it does not make or require any assumption about the distribution: indeed, it's not a test statistic, so it cannot be used to produce a p-value.

The *Bayesian information criterion* (BIC) applies a stronger penalty for model complexity:

$$BIC = k\log(n) - 2\log \mathcal{L}(\hat{\beta})$$

From probabilities to classification

While a probability is a good measure to establish the risk for a patient, it is often required to decide whether an intervention (say) should occur or not: this requires converting the probability into a binary value.

- Given a threshold θ, classify a patient as a case if $p > \theta$, and as a control otherwise.

Such thresholding has the effect of cancelling the difference between, say, 0.51 and 0.99, for $\theta = 0.5$, which instead is captured when using the log-likelihood.

Two main considerations come into play to decide the value of θ:

- Proportion of cases in the training data
- Cost assigned to misclassification errors

Classification errors

Similarly to what we saw with hypothesis testing, also in a classification setting there two types of errors we can make. These can be visualised in the *confusion matrix*.

	Predicted case	Predicted control	Total
Case	TP	FN	n_{case}
Control	FP	TN	n_{ctrl}
Total	$\hat{n}_{\text{case}}$	$\hat{n}_{\text{ctrl}}$	n

FP corresponds to making a type I error, while FN corresponds to making a type II error.

There is a cost in misclassification:

- A false positive may lead a subject to undergo an unnecessary treatment
- A false negative may not receive an intervention when one would have been beneficial

Sensitivity and specificity

Several statistical measures can be computed from the confusion matrix.

- *Accuracy* measures the proportion of correct predictions:

$$(TP + TN)/n$$

- *Sensitivity* (true positive rate) measures the ability of a model to identify cases correctly:

$$TP/n_{\text{case}}$$

- *Specificity* (true negative rate) measures the ability of a model to identify controls correctly:

$$TN/n_{\text{ctrl}}$$

A model that classifies all patients as cases has 100% sensitivity, but it would not be a good model as it would have 0% specificity.

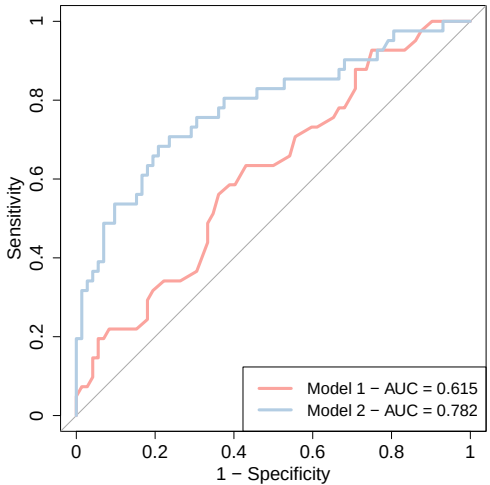
ROC curve

There is usually a trade-off between sensitivity and specificity: this is controlled by the classification threshold θ, and can be displayed graphically through the *receiver operating curve*.

A model with ROC curve that is strictly above another one is a dominating classifier.

The *area under the ROC curve* (AUC, AUROC, or C-statistic) provides a measure of the performance of a classification model over all values of θ.

The AUC is independent of the number of cases and controls in the dataset, that is it's independent of the *prevalence*.



AUC and its limitations

The area under the ROC curve can be interpreted as the probability that the model will assign a higher score to a randomly chosen case than to a randomly chosen control: this effectively measures the *discrimination performance* of a model.

The AUC is a widely used measure of performance of a classifier. However, in recent years it has been subject of criticism:

- it ignores the calibration of a model: a model which over- or under-estimates all probabilities may still have a high AUC
- it summarises the performance over extreme regions of the ROC space which would never be considered in practice
- it assigns the same cost to false positives and false negatives
- an increment in AUC obtained by using a more complex model may be over- or under-estimated depending on the AUC of the baseline model

Hosmer-Lemeshow test

Calibration refers to the ability of a model to estimate the true risk for each individual. This can be assessed through the *Hosmer-Lemeshow test*.

- Rank the predicted probabilities and divide the sample into G groups (typically 10)
- Within each subgroup, compare the observed numbers of cases and controls with the expected numbers according to the model

$$H = \sum_{g=1}^{G} \frac{\left(n_{\text{case}}^{(g)} - \sum \hat{p}^{(g)}\right)^2}{\sum \hat{p}^{(g)}} + \frac{\left(n_{\text{ctrl}}^{(g)} - \sum (1 - \hat{p}^{(g)})\right)^2}{\sum (1 - \hat{p}^{(g)})}$$

The statistic follows a χ^2 distribution with $G - 2$ degrees of freedom, and can be used to test the null hypothesis that the model is well calibrated.

The Hosmer-Lemeshow test has been heavily criticized for being very sensitive to the choice of bins and having low power.

Predictive models I

Until now we have looked at how a model can be used to describe relationships in the data.

While this is perfectly valid in an exploratory setting, this does not tell us how well the same model would perform when applied to a different sample of data with similar characteristics.

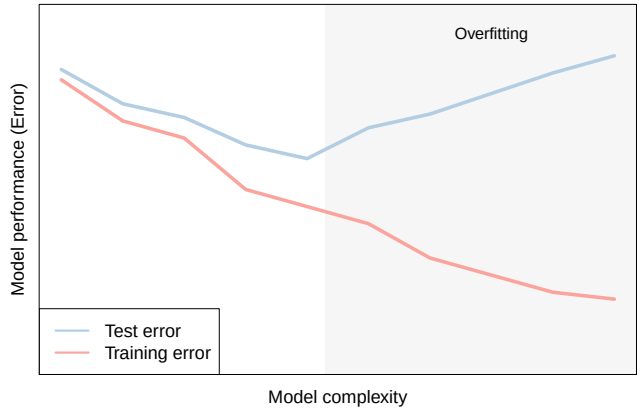
Knowing this is the first step in building a *predictive model*, that is a model that can be reliably used in predicting an outcome for data that was not used in building the model in the first place.

The prediction paradigm is also very valuable for obtaining a true measure of model fit.

Predictive models II

As function of the model complexity, the residuals can be made as small as desired. In the limit, a dataset of size n can be fitted perfectly by using n predictors.

When the model is applied to new data there is a sweet spot where the error is minimized, but the addition of further predictors causes overfitting.



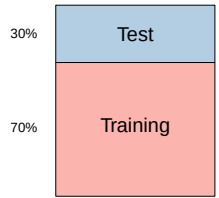
The bias–variance trade-off

The prediction error made on the test set can be decomposed in three parts:

- The *bias* measures the error made by approximating the data with a model: in general, more complex models have lower bias.
- The *variance* measures how the model would change if it was learnt on a different dataset: in general, more complex models have higher variance.
- An *irreducible error*, which measures the noise in the data that cannot be controlled by the model.

Training and test sets

A basic tool for building a predictive model is to use part of the available data for learning the model parameters (*training set*) while keeping the rest of the data untouched (*test set*) to be used only to evaluate predictions.

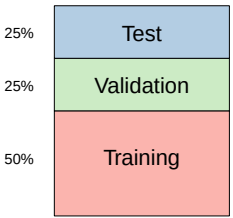


In the training set we are allowed to know the observed outcome and use it to estimate the coefficients of the model (supervised learning). In the test set, we can only use the outcome variable to compare our prediction to what was effectively observed.

Training-validation-test

Using a training/test partition allows us to evaluate the predictive performance of a single model.

In a model comparison setting, or when there are additional parameters to tune (*hyperparameters*), we need an extra partition to avoid overfitting.



Training: estimate the parameters of different models (according to different hyperparameter settings)

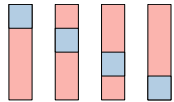
Validation: evaluate the performance of each model and choose the optimal one

Test: evaluate the performance of the optimal model on withdrawn data

Cross-validation

Creating just one partition of the data is not ideal, especially if the size of the dataset is limited: by chance the withdrawn set may be particularly difficult to predict, in which case we would underestimate the predictive performance of the model, or vice-versa.

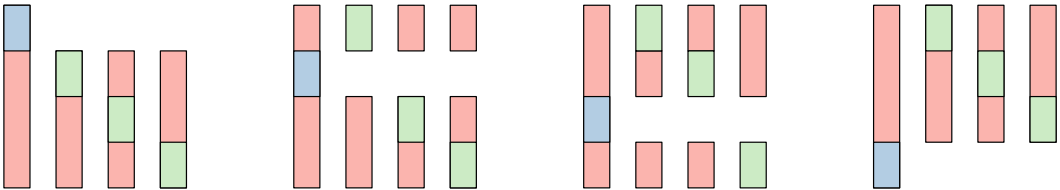
k-fold cross-validation overcomes this problem by dividing the data in k disjoint chunks: at each step one of them is withdrawn to be used as test set, while the rest forms the training set. A model is therefore fitted and tested k times.



Often $k = 10$ is enough, but for small datasets, it is better to have more folds, so that the size of the training set is larger. In the limit, if $k = n$ we have *leave-one-out cross-validation*.

Nested cross-validation

Nested cross-validation is a natural extension of cross-validation that is used for model comparison and/or tuning of hyperparameters: within the training set of each fold, a complete "inner" cross-validation is performed.



Evaluating predictions

Once we have made predictions on the test (or validation) sets, we need to evaluate and summarize the predictive performance.

For linear regression models:

- Mean square prediction error ($\frac{1}{n}\sum(y-\hat{y})^2$), mean absolute error ($\frac{1}{n}\sum|y-\hat{y}|$), test log-likelihood

For logistic regression models:

- Misclassification error, test AUC, test log-likelihood

The performance measure chosen can be summarized across the cross-validation folds by computing the mean (or the sum, for test log-likelihoods).

Further (optional) reading

The classification problem and logistic regression are presented in:

- *Introduction to Statistical Learning*, Chapter 4, especially sections 4.1 and 4.3.

Cross-validation is discussed in:

- *Introduction to Statistical Learning*, Chapter 5.1.

A very easy and colourful paper that explains sensitivity and specificity in more detail is:

- T-W. Loong, *Understanding sensitivity and specificity with the right side of the brain*, BMJ (2003)