Multivariate data analysis with R: key concepts

James G. Scott

UT Summer Statistics Institute

1 Models for grouping variables

Example data sets and scripts: rxntime, georgia.

Notation. We will use the letter y to denote a response variable, and x to denote a predictor. We will usually have more than one predictor variable (x_1, x_2, and so forth), but at least in the course, only one response. The subscript i will index cases or observations, and j will index variables. Thus y_i is the response for the ith case; x_{ij} is the value of the jth predictor for the ith case.

Residuals and fitted values. One important purpose of a statistical model is to partition variation into predictable and unpredictable components. In a simple group-wise model, we write each observation as "individual case = group mean + deviation of that case," or

$$y_i = \hat{y}_i + e_i = \text{Group mean} + \text{Residual}\,.$$

More generally, $\hat{y}_i$ is the predicted or fitted value from the model. The residual is often called the "error," but it need not be an error in the sense of observational noise. More often it is just the sum of all the effects we've chosen to leave out of the model. Residuals should have a mean of zero. If not, we could improve the model by moving the group means up or down.

Dummy variables. We usually express group-wise models in terms of *indicator* or *dummy* variables. Take the simple case of a single grouping variable x with two levels: "on" ($x = 1$) and "off" ($x = 0$). We can write this model in "baseline/offset" form:

$$y_i = \beta_0 + \beta_1 \mathbf{1}_{\{x_i = 1\}} + e_i\,.$$

The quantity $\mathbf{1}_{\{x_i=1\}}$ is called a dummy variable; it takes the value 1 when $x_i = 1$, and the value 0 otherwise. We call β_0 and β_1 the *coefficients* of the model. This way of expressing the model implies the following.

$$\begin{aligned}
\text{Group mean for case where } x \text{ is off} &= \beta_0 \\
\text{Group mean for case where } x \text{ is on} &= \beta_0 + \beta_1\,.
\end{aligned}$$

Therefore, we can think of β_0 as the baseline (or *intercept*), and β_1 as the offset.

We estimate the values of β_0 and β_1 using the least-squares criterion: that is, make the sum of squared errors, $\sum_{i=1}^n e_i^2$, as small as possible. It turns out that this is mathematically equivalent to computing the group-wise means separately. In light of this, you might wonder: why bother with the baseline/offset form? One reason is simple: we are often interested not in the means themselves, but in the *differences* between the means (in this case, the offset β_1).

Variance decomposition and R^2. The variance decomposition of a linear statistical model is

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 ,$$

or

$$TV = PV + UV .$$

Variance of the data = variance of the fitted values + variance of the residuals. PV is the variation of the predictable part of y; UV is the variation of the unpredictable part. This additive decomposition doesn't work for sums of absolute values, only sums of squares. This is not just a metaphor. It turns out to be a important consequence of the Pythagorean theorem in a high-dimensional Euclidean space. It's a big reason we use sums of squares to describe variability in statistical models.

We define R^2 as the ratio of predictable variation to total variation: PV/TV = 1-UV/TV. This quantifies the preciseness of the fit, and therefore the information content of the predictor. Some people abbreviate the variance decomposition as TSS = ESS + RSS, but I don't like this. Do the letters mean Total = Explained + Residual? Or Total = Error + Regression?

The individual terms in the variance decomposition are perfectly well defined in nonlinear statistical models. But the three terms will not, in general, add together. In this case people often just quote the mean squared error, or MSE, as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 ,$$

where n is the sample size.

More than two levels. If the predictor x has more than two levels, we must expand it in terms of more than one dummy variable. Suppose that x can take four levels, labeled arbitrarily as 0 through 3. Then our model is

$$y_i = \beta_0 + \beta_1^{(1)} \mathbf{1}_{\{x_i=1\}} + \beta_1^{(2)} \mathbf{1}_{\{x_i=2\}} + \beta_1^{(3)} \mathbf{1}_{\{x_i=3\}} + e_i .$$

More generally, $\beta_j^{(k)}$ is the coefficient associated with the kth level of the jth variable. Notice that there is no dummy variable for the case $x = 0$: this is the baseline case, whose group mean is described by the intercept β_0. In general, for a categorical variable with K levels, we will have $K-1$ dummy variables.

More than one grouping variable. Take the case of two grouping variables x_1 and x_2, each of which can take the value 0 ("off") or 1 ("on"). One approach to modeling the effect of x_1 and x_2 is to slice and dice. That is: take subsets of the data for each of the four combinations of x_1 and x_2, and compute the mean within each subset.

This approach is intuitively reasonable, but combinatorially explosive. For example, with 10 grouping variables, there will be $2^{10} = 1024$ possible subsets, and thus 1024 group-wise means to estimate. If you want to do this, you will need a lot of data—not merely overall, but for each combination separately.

A second strategy is to treat the effect of x_1 and x_2 as if they are separable:

$$y_i = \hat{y}_i + e_i = \text{Baseline} + (\text{Effect if } x_1 \text{ on}) + (\text{Effect if } x_2 \text{ on}) + \text{Residual}\,.$$

This notation gets cumbersome. We can write it more concisely as

$$y_i = \beta_0 + \beta_1 \mathbf{1}_{\{x_1\}} + \beta_2 \mathbf{1}_{\{x_2\}} + e_i\,.$$

Notice, for example, that if $x_2 = 0$, then the $\beta_2 \mathbf{1}_{\{x_2\}}$ term falls away, and we're left with the baseline, plus the effect of x_1 being on, plus the residual. We refer to β_1 and β_2 as the *main effects*.

Interactions. What if the effects of x_1 and x_2 aren't separable? That is, we believe

$$y_i = \text{Baseline} + (\text{Effect if } x_1 \text{ on}) + (\text{Effect if } x_2 \text{ on}) + (\text{Effect if both } x_1 \text{ and } x_2 \text{ on}) + \text{Residual}\,.$$

We can create such a model by multiplying dummy variables together:

$$y_i = \beta_0 + \beta_1 \mathbf{1}_{\{x_1=1\}} + \beta_2 \mathbf{1}_{\{x_2=1\}} + \beta_{12} \mathbf{1}_{\{x_1\}} \mathbf{1}_{\{x_2\}} + e_i\,.$$

We call β_{12} an *interaction term*. This one is a two-way interaction. We may also have multi-way interactions involving arbitrary numbers of predictors. The caveat is: the more multi-way interaction terms we add, the closer we come to the pure slice-and-dice approach.

Quantifying uncertainty. As we have seen, one purpose of a statistical model is to partition observed variation. Another purpose is to quantify uncertainty about any trends we see in the data. This pre-supposes that we don't know the whole story—in other words, that the data are an imperfect reflection of some underlying reality. There are three common "creation myths" that play a central role in statistical analysis.

1. The data set comprises the entire relevant population.

2. The data are a random sample from a wider population. (Archetypal examples: political polls, surveys, animals in a lab experiment.)

3. The data are one realization of a random process. (Archetypal examples: earthquakes, hurricanes, nucleotide sequences in extant organisms, photons from a distant star.)

In the first case, there is no uncertainty, and thus no need for statistical thinking! But in the other two cases, we reason as follows. Our model parameters are estimated from the data. But in a parallel universe, our data would have been different merely by random chance. Therefore our estimated model parameters might have been different, too.

How different? The answer to this question is the classical notion of an estimator's *sampling distribution*: that is, the distribution of model estimates we would get in all those parallel universes invoked by the relevant creation myth.

The standard deviation of an estimator's sampling distribution is referred to as the *standard error*. In quoting the standard error of an estimator's sampling distribution, you are saying: "If I were to take repeated samples from the population and use this estimator for every sample, my estimate is typically off from the truth by about this much." Notice that this is a claim about a procedure, not a particular estimate. The bigger the standard error, the less stable the estimator across different samples, and the less you can trust that estimator for any particular sample. This is the core idea of frequentist statistics: *uncertainty equals instability across different samples.*

Uncertainty quantification is too big a topic to treat in detail here, and I refer you to Kaplan's *Statistical Modeling: A Fresh Approach* for an introductory treatment. But the core idea of the bootstrap is quite simple, and so I'll say a few words on that.

Bootstrapping and the frequentist dream. If you really could take repeated samples from the population, life would be easy. You could simply peer into all of those alternate universes, tap each version of yourself on the shoulder, and ask, "What estimate you get for *your* sample?" By tallying up these estimates and seeing how much they differed from one another, you could discover precisely how much confidence you should place in your own estimates of β_0 and β_1, and report appropriate error bars.[1] (See below.) I refer to this thought experiment as the *frequentist dream.*

In reality, of course, we're stuck with one sample. Thus we're stuck with one of two imperfect approaches for characterizing the sampling distribution: the bootstrap, or the parametric probability model. For many data sets, there is little practical difference between the two approaches, in that they give similar standard errors. Nonetheless, there is a conceptual distinction worth preserving.

In most cases we can't repeatedly take samples of size n from the population. But we can repeatedly take samples of size *n from the sample itself*, and compute our estimator afresh for each notional sample. The idea is that the variability of the estimates across all these notional samples

[1] Let's ignore the obvious fact that, if you had access to all those alternate universes, you'd also have more data. The presence of sample-to-sample variability is the important thing to focus on here.

can be used to approximate the sampling distribution of the corresponding estimator. Each block of n resampled data points is called a bootstrapped sample. Modern software makes a non-issue of the calculational tedium involved.

You might be puzzled by something here. If there are n data points in the original sample, and we resample n data points from this "pseudo-population," won't each bootstrapped sample be precisely equal to the original sample? It turns out that the answer is no—as long as the resampling is done *with replacement* from the original sample. Sampling with replacement means that each bootstrapped sample will have duplicates and omissions from the original sample. These duplicates and omissions induce variation from one bootstrapped sample to the next. Ths variation mimics the variation you'd expect to see across the real repeated samples you're unable to take.

Resampling won't yield the true sampling distribution of an estimator. But it is often good enough for approximating the standard error. The quality of the approximation depends almost entirely on one thing: how closely the original sample resembles the wider population. Alas, this often isn't under your control, and is almost always the limiting factor in the accuracy of the bootstrap. You can't magic your way to sensible error bars by bootstrapping a biased, woefully small, or otherwise poor sample.[2]

Confidence intervals. We use standard errors to construct *confidence intervals*, or error bars. If θ is a parameter, $\hat{\theta}$ is an estimate of that parameter, and $\text{se}(\hat{\theta})$ is the standard error of the estimate, then we can quote a confidence interval of the form

$$\hat{\theta} \pm z_\alpha \cdot \text{se}(\hat{\theta}) \, .$$

The factor z_α is a constant that expresses your tolerance for error, denoted by α and expressed as a number between 0 and 1. A typical confidence level is 0.95, meaning that you'll allow your confidence interval to miss the answer 5% of the time ($\alpha = 0.05$) of the time. It's important to keep in mind that a confidence interval is a claim about the long-run properties of a statistical procedure—in how many parallel universes will the intervals so generated cover the true value? It is not a probabilistic claim about a specific data set.

If you've taken an introductory statistics course, you will likely have picked z_α by laborious calculations involving the t distribution. We'll skip this tedium and use some simple rules of thumb that statisticians have discovered to reasonably accurate: For a 68% confidence interval, choose $z_\alpha = 1$. For a 95% confidence interval, choose $z_\alpha = 2$. For a 99.5% confidence interval, choose $z_\alpha = 3$.

The signal-to-noise ratio, a.k.a. the t statistic. It is also common to quote a t-statistic, which is simply the ratio of an estimate to its standard error: $t_{\hat{\theta}} = \hat{\theta} / \text{se}(\hat{\theta})$. This conveys strictly less information than a confidence interval for the parameter. They are only useful if you wish to test

[2] The approximation also depends on how many bootstrapped samples you take from the original sample. More bootstrapped samples help—up to a point. But taking more bootstrapped samples is never a substitute for having more actual samples in the real data set.

the null hypothesis that θ is equal to some hypothesized value θ_0 (often zero). Under the null hypothesis,

$$t_{\hat{\theta}} = \frac{\hat{\theta} - \theta_0}{\mathrm{se}(\hat{\theta})} \quad \overset{(H_0)}{\longrightarrow} \quad N(0,1),$$

which means that the t-statistic approximately has a standard normal distribution under the null hypothesis. Thus, for example, if your t statistic is much bigger than 3, then either your null hypothesis is wrong or you have seen a miracle.

Two practical guidelines. First, always plot your data. This will often give you a good sense of whether your modeling assumptions are sensible, or whether you're churning through a hapless exercise in "garbage in, garbage out." Second, try never to report a guess without an error bar. The corollary of this second point is: don't be afraid to quote an estimate with weak information! Just make sure the error bars are appropriately wide.

2 Linear regression

Example data sets and scripts: kidney, gala, ut2000, epigen, profs

One predictor. In a simple one-variable regression model, we relate the response y to the predictor x using a linear equation:

$$y_i = \hat{y}_i + e_i = \beta_0 + \beta_1 x_i + e_i \, .$$

As before, we fit the model parameters by least squares. The same variance decomposition (TV = PV = UV) holds here, as does the same definition of R^2.

There are three common goals of regression analysis:

(1) Predicting a future value of y at a given x. For example, we could regress a patient's score on a clinical test for kidney function (y) on his or her age (x). When a new 55-year-old patient walks in the door, we would estimate his score as $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot 55$.

(2) Summarizing the trend. The intercept β_0 is the expected value of y when $x = 0$. The slope β_1 describes the expected change in y for every one-unit change in x.

(3) Statistical adjustment, or taking the "x"-ness out of y. The response variable y is systematically associated with x. If we fit a model

$$y_i = \hat{y}_i + e_i = \beta_0 + \beta_1 x_i + e_i \, ,$$

then the fitted value $\hat{y}_i = \beta_0 + \beta_1 x_i$ captures this systematic component of variation. Thus the residual e_i can be interpreted as the y variable, having "adjusted for" or "partialled out" x.

More than one predictor. In a multiple-regression model, we just add up the linear effects of each predictor individually,

$$y_i = \hat{y}_i + e_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + e_i \, ,$$

once again choosing the coefficients $\hat{\beta}_j$ by the principle of least squares. The interpretation of each coefficient is now slightly more involved than in the one-variable case. Each is an estimated *partial slope*: that is, the change in y associated with a one-unit change in x_j, holding all other variables constant. To see this, imagine two hypothetical cases who are identical in all predictors x_j except the first: case i has $x_1 = x^\star$, and case j has $x_1 = x^\star + 1$. Then

$$\hat{y}_j - \hat{y}_i = \{\beta_0 + \beta_1(x^\star + 1) + \beta_2 x_2 + \cdots + \beta_p x_p\} - \{\beta_0 + \beta_1 x^\star + \beta_2 x_2 + \cdots + \beta_p x_p\} \, .$$

Because x_2 through x_p are held constant, all terms but those involving β_1 cancel. We are left with

$$\hat{y}_j - \hat{y}_i = \beta_1(x^\star + 1 - x^\star) = \beta_1 \, .$$

Continuous and grouping variables together. You will often encounter situations with both continuous and categorical predictors. To handle this we simply incorporate the dummy variables associated with the categorical predictor directly into the multiple-regression equation. These dummy variables systematically shift the intercept up or down, depending on their sign. To see this, consider a case with one continuous predictor x_1, and one grouping variable x_2 that takes three levels, arbitrarily labeled 0–2. The regression equation is then

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2^{(1)} \mathbf{1}_{\{x_{i2}=1\}} + \beta_2^{(2)} \mathbf{1}_{\{x_{i2}=2\}} + e_i \, .$$

As before, there is no dummy variable for the reference category, $x_2 = 0$, as that case is handled by the intercept β_0. We can interpret this as three separate regression equations, with three different intercepts and a common slope:

$$\begin{array}{lrcl} \text{Model when } x_2 = 0: & y_i & = & \beta_0 + \beta_1 x_{i1} + e_i \\ \text{Model when } x_2 = 1: & y_i & = & \{\beta_0 + \beta_2^{(1)}\} + \beta_1 x_{i1} + e_i \\ \text{Model when } x_2 = 2: & y_i & = & \{\beta_0 + \beta_2^{(2)}\} + \beta_1 x_{i1} + e_i \, . \end{array}$$

It is quite common ask questions of the form: "How much larger or smaller are the $x_2 = 2$ cases than the $x_2 = 0$ cases, adjusting for x_1?" The estimate and error bar for $\beta_2^{(2)}$ provide the answer.

We can also have interactions between dummy variables and continuous predictors. The notation for this is straightforward, if a bit cumbersome:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2^{(1)} \mathbf{1}_{\{x_{i2}=1\}} + \beta_2^{(2)} \mathbf{1}_{\{x_{i2}=2\}} + \gamma_1^{(1)} x_{i1} \mathbf{1}_{\{x_{i2}=1\}} + \gamma_1^{(2)} x_{i1} \mathbf{1}_{\{x_{i2}=2\}} + e_i \, .$$

We may again interpret this as three separate regression equations, each with a distinct slope and intercept:

$$\begin{aligned}
\text{Model when } x_2 = 0\text{:} \quad y_i &= \beta_0 + \beta_1 x_{i1} + e_i \\
\text{Model when } x_2 = 1\text{:} \quad y_i &= \{\beta_0 + \beta_2^{(1)}\} + \{\beta_1 + \gamma_1^{(1)}\} x_{i1} + e_i \\
\text{Model when } x_2 = 2\text{:} \quad y_i &= \{\beta_0 + \beta_2^{(2)}\} + \{\beta_1 + \gamma_1^{(2)}\} x_{i1} + e_i \, .
\end{aligned}$$

We may be interested in a question of the form: "How much faster or slower does y grow with x among the cases where $x_2 = 2$ than the cases where $x_2 = 0$?" The estimate and error bar for $\gamma_1^{(2)}$ provide the answer.

Choosing a model. Often you'll face the problem of comparing models with different numbers of predictors, or with different powers of a single predictor. The model with more variables will always fit the data better, but it might not be a better model, because it might end up overfitting noise in the data.

For these kinds of cross-dimensional comparisons, R^2 is useless. In fact, in many ways it is worse than nothing: R^2 will always go up when we add new predictors, even if those predictors have nothing to do with the response. This is why it is crucial not to think of R^2 as measuring "goodness of fit." You'll just end up tying yourself into a mental knot pondering how a "good" model can still be rotten.

So let's put aside R^2. Instead, we need a criterion that balances the twin virtues of fit and simplicity. Put another way, we need a quantitative version of Occam's Razor, the philosophical principle that instructs us to make explanations only as complex as they need to be. When it comes to regression models, fit and simplicity are both easily operationalized. Models that fit more precisely have higher R^2, and lower residual variance. Simpler models have fewer free parameters to estimate.

Here are two commonly used criteria for scoring models according to the tradeoff they offer between fit and simplicity.

1) Adjusted R^2, denoted R^2_A. We recall that the original R^2 was defined as

$$R^2 = 1 - \frac{UV}{TV} \, .$$

Adjusted R^2 is defined in almost the same way, but with a subtle modification to punish models that have larger numbers of parameters:

$$R^2_A = 1 - \frac{UV}{TV} \cdot \left(\frac{n-1}{n-p-1} \right) = 1 - \left\{ (1 - R^2) \cdot \left(\frac{n-1}{n-p-1} \right) \right\} \, .$$

Here n denotes the sample size, and p the number of regressors in the model (not counting the intercept), just as for the F test. Higher values of R^2_A are (ostensibly) better. But unlike

R^2, R_A^2 can sometimes go down when you add another predictor to the model. Adjusted R^2 is part of the standard output in most regression software, but even if you can't find it, it's easy to calculate from regular R^2.

2) AIC, which stands for "Akaike information criterion." Don't pay too much attention to the full name; what's important is not the derivation of AIC, but the manner in which it balances fit and simplicity:

$$\mathrm{AIC} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + 2p\,.$$

The first term, the residual sum of squares, measures lack of fit; models with larger residuals fit the data less precisely. The second term measures complexity; p is the number of free parameters in the model, and is therefore larger for more complex models. From this, we reason that lower values of AIC are (ostensibly) better.

Of course, the best Occam's Razor of all is to test models by having them predict y's for x's they've never seen before—that is, on fresh data that hasn't itself been used to fit the models. This is called out-of-sample predictive validation, and is brutally effective at winnowing down your list of good models. The only problem is that data is sometimes expensive, and you might not be able to collect enough extra data to run a good out-of-sample test. That's when these other Occam's Razors can be very useful. (Out-of-sample predictive validation can also be imperfectly approximated using cross-validation, not covered here.)

3 Logistic regression

Example data sets and scripts: spam, brca, gardasil, cmc, resume

The linear probability model. In many situations, we would like to forecast the outcome of a binary event, given some relevant information:

- Given the pattern of word usage and punctuation in an e-mail, is it likely to be spam?
- Given the temperature and cloud cover on Christmas Eve, is it likely to snow on Christmas?
- Given a person's credit history, is he or she likely to default on a mortgage?

In all of these cases, the y variable is the answer to a yes-or-no question. Nonetheless, we can still use regression for these problems. Let's suppose, for simplicity's sake, that we have only one predictor x, and that we let $y_i = 1$ for a "yes" and $y_i = 0$ for a "no." One naïve way of forecasting y is simply to plunge ahead with the basic, one-variable regression equation:

$$\mathrm{E}(y_i \mid x_i) = \beta_0 + \beta_1 x_i\,.$$

Since y_i can only take the values 0 or 1, the expected value of y_i is simply a weighted average of these two cases:

$$\begin{aligned} \mathrm{E}(y_i \mid x_i) &= 1 \cdot P(y_i = 1 \mid x_i) + 0 \cdot P(y_i = 0 \mid x_i) \\ &= P(y_i = 1 \mid x_i) \end{aligned}$$

Therefore, the regression equation is just a linear model for the conditional probability that $y_i = 1$, given the predictor x_i:

$$P(y_i = 1 \mid x_i) = \beta_0 + \beta_1 x_i \, .$$

This model allows us to plug in some value of x_i and read off the forecasted probability of a "yes" answer to whatever yes-or-no question is being posed. It is often called the linear probability model, since the probability of a "yes" varies linearly with x.

The logistic link function. The linear probability model is perfectly reasonable in many situations. But suffers from a noticeable problem. The left-hand side of the regression equation, $P(y_i = 1 \mid x_i)$, must be between 0 and 1. But the right-hand side, $\beta_0 + \beta_1 x_i$, can be any real number between $-\infty$ and ∞. We'd be better off with some transformation g that takes an unconstrained number from the right-hand side, and maps it to a constrained number on the left-hand side:

$$P(y_i \mid x_i) = g(\beta_0 + \beta_1 x_i) \, .$$

Such a function g is called a *link function*. A model that incorporates such a link function is called a *generalized linear model*; and the part inside the parentheses $(\beta_0 + \beta_1 x_i)$ is called the *linear predictor*, and is often denoted as ψ_i.

We use link functions and generalized linear models in most situations where we are trying to predict a number that is, for whatever reason, constrained. Here, we're dealing with probabilities, which are constrained to be no smaller than 0 and no larger than 1. Therefore, the function g must map real numbers on $(-\infty, \infty)$ to numbers on $(0, 1)$. It must therefore be shaped a bit like a flattened letter "S," approaching zero for large negative values of ψ_i, and approaching 1 for large positive values.

With multiple regressors $(x_{i1}, \ldots, x_{ip})$, we have

$$\Pr(y_i = 1 \mid x_i) = w_i = \frac{\exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})} \, . \qquad (1)$$

Recall that odds are just a different way of expressing probabilities:

$$(\text{Odds that } y_i \text{ is } 1) = O_i = \frac{w_i}{1 - w_i} \, .$$

If you churn through the algebra and re-express the logistic-regression equation (1) in terms of odds, you will see that the log-odds of success—or equivalently the *logit transform* of the success probability—are being modeled as a linear function of the predictors:

$$\text{logit}(w_i) = \log O_i = \log \left(\frac{w_i}{1 - w_i} \right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} \, .$$

A technical aside: this model cannot be fit by least squares. Instead, it is fit via maximum-likelihood, and requires a nonlinear optimization routine. The most commonly used is a variation on the Newton–Raphson algorithm called *iteratively re-weighted least squares.* This can sometimes break! Thus if you are getting very strange answers

Interpreting the coefficients. For the sake of simplicity, imagine a data set with only a single regressor x_i that can take the values 0 or 1 (a dummy variable). Perhaps, for example, x_i denotes whether someone received the new treatment (as opposed to the control) in a clinical trial.

For this hypothetical case, let's consider the ratio of two quantities: the odds of success for person i with $x_i = 1$, versus the odds of success for person j with $x_j = 0$. Denote this ratio by R_{ij}. We can write this as

$$\begin{aligned}
R_{ij} &= \frac{O_i}{O_j} \\
&= \frac{\exp\{\log(O_i)\}}{\exp\{\log(O_j)\}} \\
&= \frac{\exp\{\beta_0 + \beta_1 \cdot 1\}}{\exp\{\beta_0 + \beta_1 \cdot 0\}} \\
&= \exp\{\beta_0 + \beta_1 - \beta_0 - 0\} \\
&= \exp(\beta_1) \, .
\end{aligned}$$

Therefore, we can interpret the quantity e^{β_1} as an *odds ratio*. Since $R_{ij} = O_i / O_j$, we can also write this as:

$$O_i = e^{\beta_1} \cdot O_j \, .$$

In words: if we start with $x = 0$ and move to $x = 1$, our odds of success ($y = 1$) will change by a multiplicative factor of e^{β_1}.

The ordinal logit model. We can modify the logistic regression model to handle ordinal responses. The hallmark of ordinal variables is that they are measured on a scale that can't easily be associated with a numerical magnitude, but that does imply an ordering: employee evaluations, survey responses, bond ratings, and so forth.

There are several varieties of ordinal logit model. Here we consider the *proportional-odds* model, which is most easily understood as a family of related logistic regression models. Label the categories as $1, \ldots, K$, ordered in the obvious way. Consider the probability $c_{ik} = P(y_i \leq k)$: the probability that the outcome for the ith case falls in category k *or any lower category.* (We call it c_{ik} because it is a cumulative probability of events at least as "low" as k.) The proportional-odds logit model assumes that the logit transform of c_{ik} is a linear function of predictors:

$$\operatorname{logit}(c_{ik}) = \log\left(\frac{c_{ik}}{1-c_{ik}}\right) = \eta_k + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\,.$$

Crucially, this relationship is assumed to hold for all categories at once. Because $c_{iK} = 1$ for the highest category K, we have specified $K-1$ separate binary logit models that all share the same predictors x_j and the same coefficients β_j. The only thing that differs among the models are the intercepts η_k; these are commonly referred to as the *cutpoints*. Since the log odds differ only by an additive constant for different categories, the odds differ by a multiplicative factor—thus the term "proportional odds."

To interpret the ordinal-logit model, I find it easiest to re-express individual fitted values in terms of covariate-specific category probabilities $w_{ik} = P(y_i = k)$:

$$w_{ik} = P(y_i \leq k) - P(y_i \leq k-1) = c_{ik} - c_{i,k-1}\,,$$

with the convention that $c_{i0} = 0$. Good software makes it fairly painless to do this.

The multinomial logit model. Another generalization of the binary logit model is the multinomial logit model. This is intended for describing *unordered* categorical responses: PC/Mac/Linux, Ford/Toyota/Chevy, plane/train/automobile, and so forth. Without a natural ordering to the categories, the quantity $P(y_i \leq k)$ ceases to be meaningful, and we must take a different approach.

Suppose there are K possible outcomes ("choices"), again labeled as $1, \ldots, K$ (but without the implied ordering). As before, let $w_{ik} = P(y_i = k)$. For every observation, and for each of the K choices, we imagine that there is a linear predictor ψ_{ik} that measures the preference of subject i for choice k. Intuitively, the higher ψ_{ik}, the more likely that $y_i = k$.

The specific mathematical relationship between the linear predictors and the probabilities w_{ik} is given the multinomial logit transform:

$$\begin{aligned} w_{ik} &= \frac{\exp(\psi_{ik})}{\sum_{l=1}^{K} \exp(\psi_{il})} \\ \psi_{ik} &= \beta_0^{(k)} + \beta_1^{(k)} x_{i1} + \cdots \beta_p^{(k)} x_{ip}\,. \end{aligned}$$

Each category gets its own set of coefficients, but the same set of predictors x_1 through x_p.

There is one minor issue here. With a bit of algebra, you could convince yourself that adding a constant factor to each ψ_{ik} would not change the resulting probabilities w_{ik}, as this factor would cancel from both the numerator and denominator of the above expression. To fix this indeterminacy, we choose one of the categories (usually the first or last) to be the reference category, and set its coefficients equal to zero.

4 Models for count outcomes

Example data sets and scripts: springbok, flutrends

The Poisson model. For modeling event-count data (photons, organisms, heart attacks), a useful place to start is the Poisson distribution. The key feature of counts is that they must be non-negative integers. Like the case of logistic regression, where probabilities had to live between 0 and 1, this restriction creates some challenges that take us beyond ordinary least squares.

The Poisson distribution is parametrized by a rate parameter, often written as λ. Let k denote an integer, and y_i denote the event count for subject i. In a Poisson model, we assume that

$$P(y_i = k) = \frac{\lambda_i^k}{k!} e^{-\lambda_i} \, ,$$

and we wish to model λ_i in terms of covariates. Because the rate parameter of the Poisson cannot be negative, we must employ the same device of a link function to relate λ_i to covariates. By far the most common is the (natural) log link:

$$\log \lambda_i = \beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip} \, ,$$

or equivalently,

$$\lambda_i = \exp\{\beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip}\} \, .$$

As with the case of logistic regression, the model is fit via maximum-likelihood.

Interpreting the coefficients. Because we are fitting a model on the log-rate scale, additive changes to an x variable are associated with multiplicative changes in the y variable. As before, let's consider the ratio of two quantities: the rate of events for person i with $x_1 = x^\star + 1$, versus the rate of events for person j with $x_1 = x^\star$. Let's further imagine that all other covariates are held constant at values x_2 to x_p, respectively. This implies that the only difference between subjects i and j is a one-unit difference in the first predictor, x_1.

We can write their ratio of rates as

$$\begin{aligned} R_{ij} &= \frac{\lambda_i}{\lambda_j} \\ &= \frac{\exp\{\beta_0 + \beta_1 \cdot (x^\star + 1) + \beta_2 x_2 + \cdots \beta_p x_p\}}{\exp\{\beta_0 + \beta_1 \cdot x^\star + \beta_2 x_2 + \cdots \beta_p x_p\}} \\ &= \exp\{\beta_1 (x^\star + 1 - x^\star)\} \\ &= \exp(\beta_1) \, . \end{aligned}$$

Thus person i experiences events events e^{β_1} times as frequently as person j.

Overdispersion. For most data sets outside of particle physics, the Poisson assumption is usually one of convenience. Like the normal distribution, it is familiar and easy to work with. It also has teeth, and may bite if used improperly. One crucial feature of the Poisson is that its mean and variance are equal: that is, if $y_i \sim \text{Pois}(\lambda_i)$, then the expected value of y_i is λ_i, and the standard deviation of y_i is $\sqrt{\lambda_i}$. (Since λ_i depends on covariates, we should really be calling these the *conditional* expected value and standard deviation.)

As a practical matter, this means that if your data satisfy the Poisson assumption, then roughly 95% of observations should fall within $\pm 2\sqrt{\lambda_i}$ of their conditional mean λ_i. This is quite narrow, and many (if not most) data sets exhibit significantly more variability about their mean. If the conditional variance exceeds the conditional mean, the data exhibits *overdispersion with respect to the Poisson*, or just *overdispersion* for short.

Overdispersion can really mess with your standard errors. In other words, if you use (i.e. let your software use) the Poisson assumption to calculate error bars, but your data are overdispersed, then you will end up overstating your confidence in the model coefficients. Sometimes the effect is dramatic, meaning that the blind use of the Poisson assumption is a recipe for trouble.

There are three common strategies for handling overdispersion:

1. Use a quasi-likelihood approach ("family=quasipoisson" in R's glm function);
2. Fit a different count-data model, such as the negative binomial or Poisson-lognormal, that can accommodate overdispersion;
3. Fit a hierarchical model.

5 Survival analysis

Example data sets and scripts: colon, recid

Survival times. Suppose that we decide to run an epidemiological cohort study, which is a kind way of saying that we follow people and wait until something bad happens to them (an "event"). Let T_i be the time elapsed from the start of the study until the event. The random variable T_i is often called a survival time—even if the event in question isn't an actual death—or alternatively, a failure time.

In most studies of this kind, the goal is to understand how a subject's survival time depends on covariates:

- Under which treatment arm of a clinical trial do people survive longer?
- Does this computer screen last longer under manufacturing process A or B?
- Do criminals who read Nietzsche in prison recidivate at higher rates?

There are many ways to proceed. We could directly model $F(t) = P(T_i \leq t)$, the cumulative distribution function of the random variable T_i. Equivalently, we could model the corresponding probability density $f(t)$, or the *survival curve* $S(t) = 1 - F(t) = P(T_i > t)$. This is the most natural extension of regression analysis—specify a probability model, and describe changes in the model's parameters as a function of covariates. Many approaches to survival analysis involve just this; examples include the Weibull, gamma, and log-normal.

Modeling the hazard function. An alternative approach is to model the *hazard function*, denoted $h(t)$:

$$h_i(t) \approx \frac{P\left(t < T_i < t + \Delta t \mid T_i > t\right)}{\Delta t},$$

for some small time interval of width Δt. We actually define the hazard function using calculus, as the limit of this quantity as Δt approaches 0. Intuitively, the hazard function is the instantaneous rate of failure at time t, conditional upon having survived up to time t. It turns out that the density $f(t)$ and the hazard function $h(t)$ can be used to give mathematically equivalent specifications of the distribution of the random survival time T_i.

The Cox proportional-hazards model is a model for the hazard function $h(t)$. It is the most popular tool for survival analysis because it is simple, and because it can easily accommodate *right-censoring*: that is, the presence of subjects in the data set who have not yet experienced a failure by the end of the study period. Virtually all survival analyses involve right-censoring, which is not as easily or transparently handled in models for $f(t)$.

The key assumption of the Cox model is proportionality, or separability. Specifically, it assumes that subject i, having covariates x_{i1} through x_{ip}, has the hazard function

$$h_i(t) = h_0(t) \cdot \exp\left\{\beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip}\right\}.$$

Notice that $h_0(t)$ is a function of the time t. We call this the *baseline hazard function.* Everything else on the right-hand side just boils down to a single scalar $\exp(\psi_i)$ that depends on a subject's covariates. This factor uniformly inflates or deflates the baseline hazard across all values of t. The Cox model is therefore *semiparametric*, in that it allows a flexible nonparametric model for the baseline hazard $h_0(t)$, but requires that the effect of covariates enter through a parametric linear model.

Interpreting the coefficients. Consider the ratio of two hazard functions: the hazard for person i with $x_1 = x^\star + 1$, versus the hazard for person j with $x_1 = x^\star$. As before, we imagine that all other covariates are held constant at values x_2 to x_p, respectively. Thus the only difference between subjects i and j is a one-unit difference in the first predictor.

We can write their ratio of hazard functions as

$$
\begin{aligned}
\frac{h_i(t)}{h_j(t)} &= \frac{h_0(t)\exp\{\beta_0 + \beta_1 \cdot (x^\star + 1) + \beta_2 x_2 + \cdots \beta_p x_p\}}{h_0(t)\exp\{\beta_0 + \beta_1 x^\star + \beta_2 x_2 + \cdots \beta_p x_p\}} \\
&= \exp\{\beta_1(x^\star + 1 - x^\star)\} \\
&= \exp(\beta_1)\,.
\end{aligned}
$$

Thus person i has a hazard function e^{β_1} times higher (or lower) than person j. Crucially, this is assumed to hold across all values of t. This explains why, to summarize the results of a Cox model, people usually exponentiate the coefficients and quote them as *hazard ratios*.