12
Generalized linear models

In this chapter, we combine regression models with other parametric probability models like the binomial and Poisson distributions.

Binary responses

In many situations, we would like to predict 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, pressure, and cloud cover on Christmas Eve, is it likely to snow on Christmas Day?
- Given a person's credit history and income, is he or she likely to default on a mortgage loan?

In all of these cases, the Y variable is the answer to a yes-or-no question. This is a bit different to the kinds of problems we've become used to seeing, where the response is a real number.

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:

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

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:

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

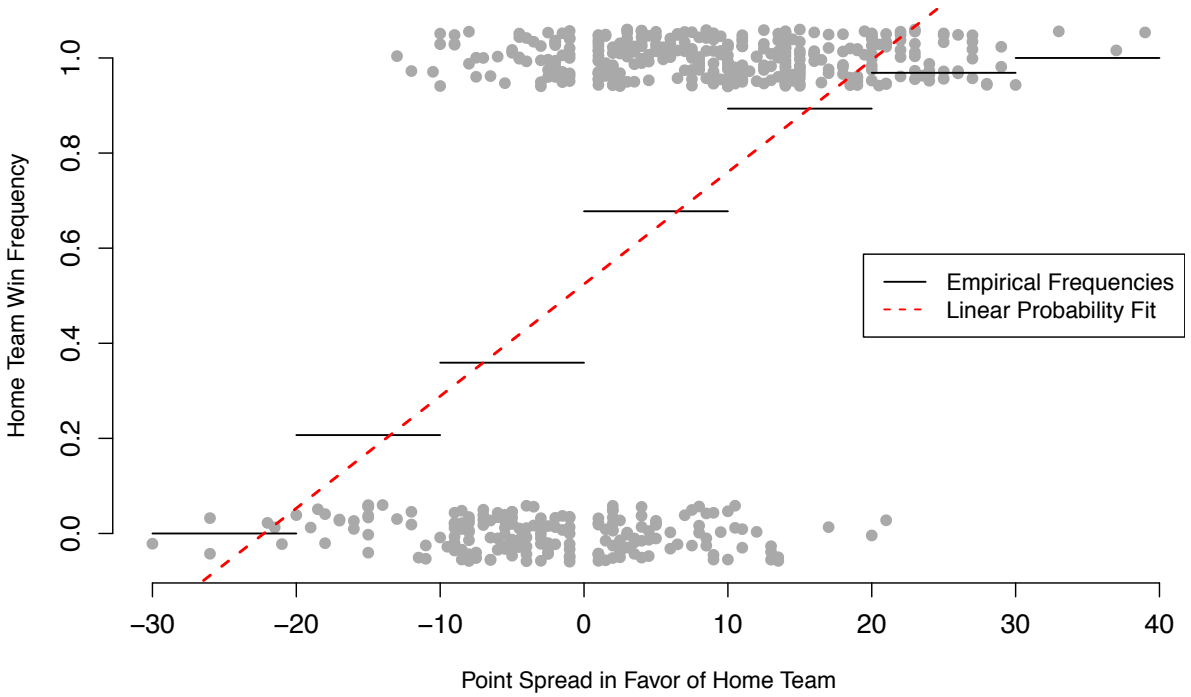

Figure 12.1: Win frequency versus point spread for 553 NCAA basketball games. Actual wins are plotted as 1's and actual losses as zeros. Some artificial vertical jitter has been added to the 1's and 0's to allow the dots to be distinguished from one another.

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.

Let's try fitting it to some example data to understand how this kind of model behaves. In Table 12.1 on page 277, we see an excerpt of a data set on 553 men's college-basketball games. Our y variable is whether the home team won ($y_i = 1$) or lost ($y_i = 0$). Our x variable is the Las Vegas "point spread" in favor of the home team. The spread indicates the betting market's collective opinion about the home team's expected margin of victory—or defeat, if the spread is negative. Large spreads indicate that one team is heavily favored to win. It is therefore natural to use the Vegas spread to predict the probability of a home-team victory in any particular game.

Figure 12.1 shows each of the 553 results in the data set. The

home-team point spread is plotted on the x-axis, while the result of the game is plotted on the y-axis. A home-team win is plotted as a 1, and a loss as a 0. A bit of artificial vertical jitter has been added to the 1's and 0's, just so you can distinguish the individual dots.

The horizontal black lines indicate empirical win frequencies for point spreads in the given range. For example, home teams won about 65% of the time when they were favored by more than 0 points, but less than 10. Similarly, when home teams were 10–20 point underdogs, they won only about 20% of the time.

Finally, the dotted red line is the linear probability fit:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.524435   0.019040   27.54   <2e-16 ***
spread      0.023566   0.001577   14.94   <2e-16 ***
---
Residual standard error: 0.4038 on 551 degrees of freedom
Multiple R-squared: 0.2884
```

This is the result of having regressed the binary y_i's on the point spreads, simply treating the 1's and 0's as if they were real numbers. Under this model, our estimated regression equation is

$$\mathrm{E}(y_i \mid x_i) = P(y_i = 1 \mid x_i) = 0.524 + 0.024 \cdot x_i \,.$$

Plug in an x, and read off the probability of a home-team victory. Here, we would expect the intercept to be 0.5, meaning that the home team should win exactly 50% of the time when the point spread is 0. Of course, because of sampling variability, the estimated intercept $\widehat{\beta}_0$ isn't exactly 0.5. But it's certainly close—about 1 standard error away.

The linear probability model, however, has a serious flaw. Try plugging in $x_i = 21$ and see what happens:

$$P(y_i = 1 \mid x_i = 21) = 0.524 + 0.024 \cdot 21 = 1.028 \,.$$

We get a probability larger than 1, which is clearly nonsensical. We could also get a probability less than zero by plugging in $x_1 = -23$:

$$P(y_i = 1 \mid x_i = -23) = 0.524 - 0.024 \cdot 23 = -.028 \,.$$

The problem is that the straight-line fit does not respect the rule that probabilities must be numbers between 0 and 1. For many values of x_i, it gives results that aren't even mathematically legal.

Game	Win	Spread
1	0	-7
2	1	7
3	1	17
4	0	9
5	1	-2.5
6	0	-9
7	1	10
8	1	18
9	1	-7.5
10	0	-8
⋮		
552	1	-4.5
553	1	-3

Table 12.1: An excerpt from a data set on 553 NCAA basketball games. "Win" is coded 1 if the home team won the game, and 0 otherwise. "Spread" is the Las Vegas point spread in favor of the home team (at tipoff). Negative point spreads indicate where the visiting team was favored.

Link functions and generalized linear models

THE PROBLEM can be summarized as follows. The right-hand side of the regression equation, $\beta_0 + \beta_1 x_i$, can be any real number between $-\infty$ and ∞. But the left-hand side, $P(y_i = 1 \mid x_i)$, must be between 0 and 1. Therefore, we need 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*, or GLM. The part inside the parentheses $(\beta_0 + \beta_1 x_i)$ is called the *linear predictor*.

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 the linear predictor, and approaching 1 for large positive values.

Figure 12.2 contains the most common example of such a link function. This is called the *logistic* link, which gives rise to the *logistic regression model*:

$$P(y_i = 1 \mid x_i) = g(\beta_0 + \beta_1 x_i) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \,.$$

Think of this as just one more transformation, like the logarithm or powers of some predictor x. The only difference is that, in this case, the transformation gets applied to the whole linear predictor at once.

With a little bit of algebra, it is also possible to isolate the linear predictor $\beta_0 + \beta_1 x_i$ on one side of the equation. If we let p_i denote

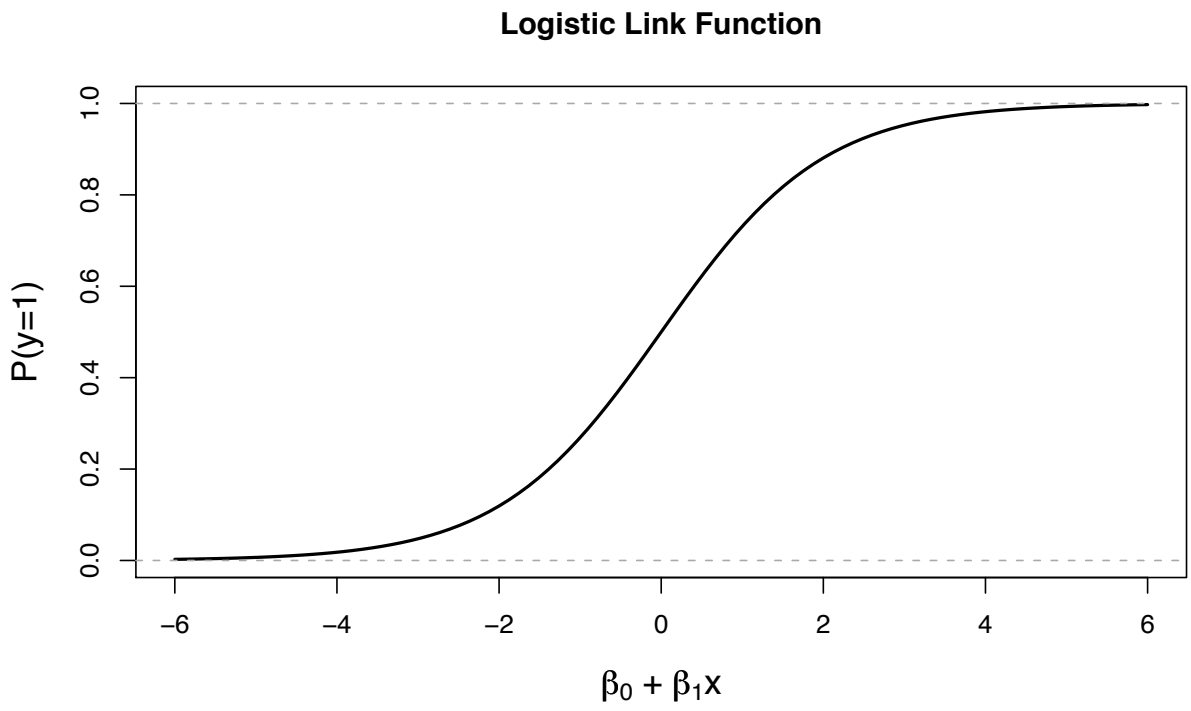

Figure 12.2: The logistic link function.

the probability that $y_i = 1$, given x_i, then

$$
\begin{aligned}
p_i &= \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}} \\
p_i + p_i e^{\beta_0+\beta_1 x_i} &= e^{\beta_0+\beta_1 x_i} \\
p_i &= (1-p_i)e^{\beta_0+\beta_1 x_i} \\
\log\left(\frac{p_i}{1-p_i}\right) &= \beta_0 + \beta_1 x_i
\end{aligned}
$$

Since $p_i = P(y_i = 1 \mid x_i$, we know that $1 - p_i = P(y_i = 0 \mid x_i)$. Therefore, the ratio $p_i/(1-p_i)$ is the odds in favor of the proposition that $y_i = 1$, given the predictor x_i. This means that the $\beta_0 + \beta_1 x_i$ is a linear predictor for the logarithm of the odds in favor of success ($y_i = 1$). This is

Estimating the parameters of the logistic regression model

In previous chapters we learned how to estimate the parameters of a linear regression model using the least-squares criterion. This involved choosing values of the regression parameters to minimize the quantity

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

where $\hat{y}_i$ is the value for y_i predicted by the regression equation.

In logistic regression, the analogue of least-squares is Gauss's principle of maximum likelihood. The idea here is to choose values for β_0 and β_1 that make the observed patterns of 1's and 0's as small a miracle as possible.

To understand how this works, observe the following two facts:

- If $y_i = 1$, then we have observed an event that occurred with probability $P(y_i = 1 \mid x_i)$. Under the logistic-regression model, we can write this probability as

$$P(y_i = 1 \mid x_i) = \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}}$$

- If $y_i = 0$, then we have observed an event that occurred with probability $P(y_i = 0 \mid x_i) = 1 - P(y_i = 1 \mid x_i)$. Under the logistic regression model, we can write this probability as

$$1 - P(y_i = 1 \mid x_i) = 1 - \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}}$$

Since all of the individual 1's and 0's are independent, given the parameters β_0 and β_1, the probability of having observed our entire data set is the product of the probabilities for the individual 1's and 0's. We can write this as:

$$P(y_1, \ldots, y_n) = \prod_{i:y_i=1}\left(\frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}}\right) \cdot \prod_{i:y_i=0}\left(1 - \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}}\right).$$

This expression is our *likelihood*; it is the probability of having observed our data, given some particular configuration of the model parameters.[1] The logic of maximum likelihood is to choose values for β_0 and β_1 such that $P(y_1, \ldots, y_n)$ is as large as possible. We denote these choices by $\widehat{\beta}_0$ and $\widehat{\beta}_1$. These are called the *maximum-likelihood estimates* (MLE's) for the logistic regression model.

[1] Remember that the big $\prod$ signs mean "product," just like $\sum$ means "sum." The first product is for the observations where y_i was a 1, and the second product is for the observations where y_i was a 0.

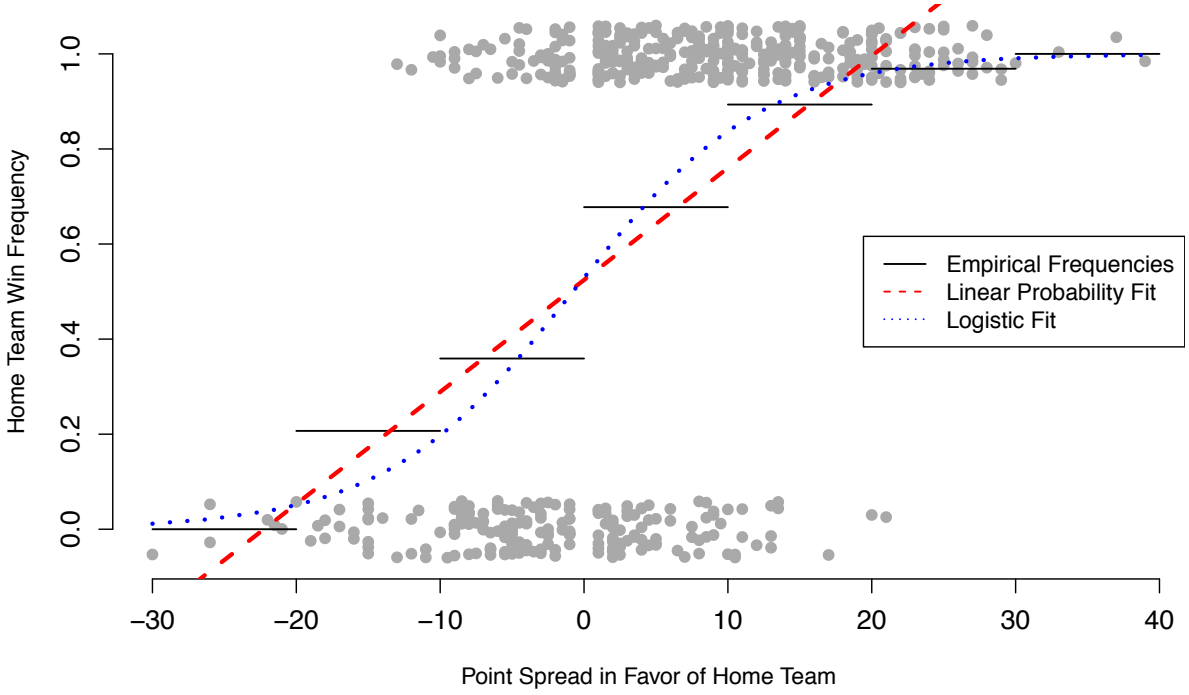

Figure 12.3: Win frequency versus point spread for 553 NCAA basketball games. Actual wins are plotted as 1's and actual losses as zeros. Some artificial vertical jitter has been added to the 1's and 0's to allow the dots to be distinguished from one another.

This likelihood is a difficult expression to maximize by hand (i.e. using calculus and algebra). Luckily, most major statistical software packages have built-in routines for fitting logistic-regression models, absolving you of the need to do any difficult analytical work.

The same is true when we move to multiple regression, when we have p predictors rather than just one. In this case, the logistic-regression model says that

$$P(y_i = 1 \mid x_{i1}, \ldots, x_{i,p} = g(\beta_0 + \beta_1 x_i) = \frac{e^{\psi_{ij}}}{1 + e^{e^{\psi_{ij}}}}, \quad , \psi_{ij} = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$$

where ψ_{ij} is the linear predictor for observation i.

The logistic regression fit for the point-spread data

Let's return briefly to the data on point spreads in NCAA basketball games. The figure above compares the logistic model to the

linear-probability model. The logistic regression fit ($\widehat{\beta}_0 = 0.117$, $\widehat{\beta}_1 = 0.152$) eliminates the undesirable behavior of the linear model, and ensures that all forecasted probabilities are between 0 and 1. Note the clearly non-linear behavior of the dotted blue curve. Instead of fitting a straight line to the empirical success frequencies, we have fit an S-shape.

Interpreting the coefficients

Interpreting the coefficients in a logistic regression requires a bit of algebra. 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}.

For this reason, we usually refer to the exponentiated coefficient e^{β_j} as the odds ratio associated with predictor j.

Extensions to the basic logit model

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:

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

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:[2]

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

[2] Some people, usually computer scientists, will refer to this as the *softmax* function.

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.

Models for count outcomes

The Poisson model. For modeling event-count data (photons, mortgage defaults in a ZIP code, heart attacks in a town), 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.

Alas, these topics are for a more advanced treatment of generalized linear models.