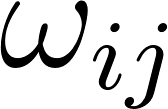
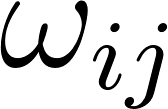
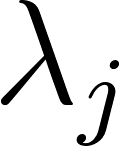
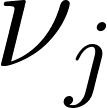
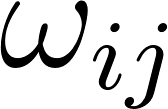
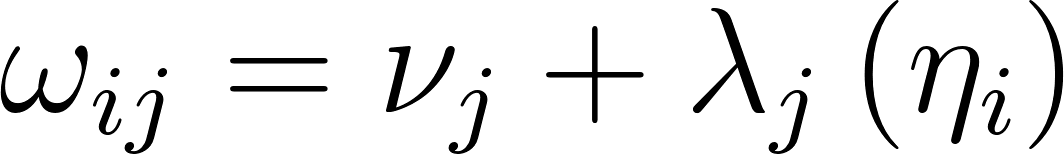
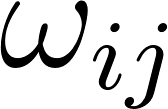
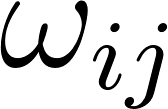
***Nonlinear latent variable models***

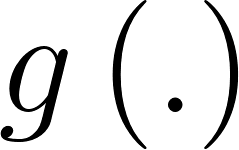
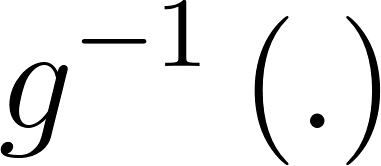
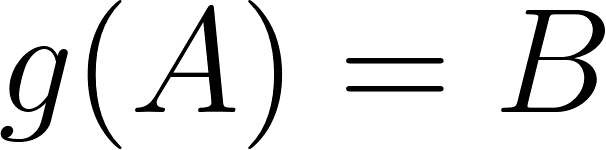
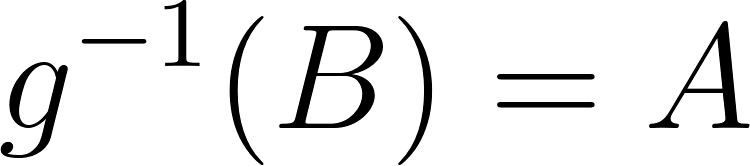
The latent variable model outlined in Equation 1 in our chapter is most developmental scientists’ starting point. But for a binary or ordinal variable, like most of the ones in our example, this assumption is not plausible. For instance, imagine that item [](https://www.codecogs.com/eqnedit.php?latex=y_%7Bij%7D#0) is a binary variable - i.e., a variable that can take a value of 0 or 1. We do not model this using a normal distribution, because unlike a continuous variable it can only take a value of 0 or 1. Thus, instead of modeling each individual’s response [](https://www.codecogs.com/eqnedit.php?latex=y_%7Bij%7D#0), we model the probability that they will provide an answer of 1, which we denote as [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0).

To review what we have so far, each individual provides a response to an item, [](https://www.codecogs.com/eqnedit.php?latex=y_%7Bij%7D#0), which is based on a probability [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0). That probability [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0) is a function of that individual’s value of the latent variable [](https://www.codecogs.com/eqnedit.php?latex=%5Ceta_%7Bij%7D#0). In order to get from [](https://www.codecogs.com/eqnedit.php?latex=%5Ceta_i#0) to [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0) we have to go through a number of steps, which we will break down here. While this is a relatively long path, each successive step is not too great a leap from the previous one.

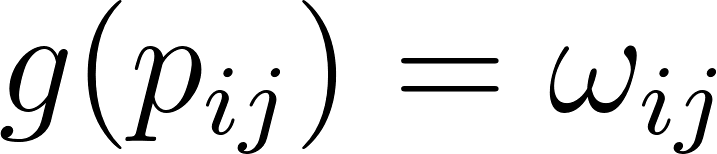
Our first step is to define a new variable: instead of modeling [](https://www.codecogs.com/eqnedit.php?latex=y_%7Bij%7D#0), we instead model an intermediate quantity, which we refer to as [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0). Just like [](https://www.codecogs.com/eqnedit.php?latex=y_%7Bij%7D#0), [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0) is a linear function of [](https://www.codecogs.com/eqnedit.php?latex=%5Ceta_%7Bi%7D#0). To clarify what we mean by “linear function,” notice the form of Equation 1, in which[](https://www.codecogs.com/eqnedit.php?latex=%20y_%7Bij%7D#0) is related to a multiple (given by [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_j#0)) of an individual’s value of the latent variable (given by [](https://www.codecogs.com/eqnedit.php?latex=%5Ceta_%7Bij%7D#0)), plus a constant (given by [](https://www.codecogs.com/eqnedit.php?latex=%5Cnu_j#0)). We can calculate [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0) in the same way as [](https://www.codecogs.com/eqnedit.php?latex=y_%7Bij%7D#0) in Equation 1, minus the residual [](https://www.codecogs.com/eqnedit.php?latex=%5Cepsilon_%7Bij%7D#0):

[](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D%20%3D%20%5Cnu_j%20%2B%20%5Clambda_j%5Cleft(%5Ceta_i%5Cright)#0) (2)

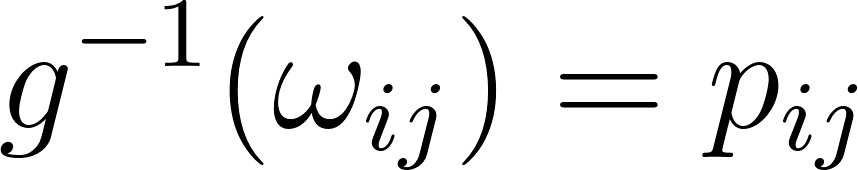
This intermediary, [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0), is referred to as the *linear predictor*. We can now use it to model the probability of endorsement, [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0). We connect [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0) to [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0)using the aptly named *link function*.

This link function essentially translates a linear response into a nonlinear one. The link function is denoted [](https://www.codecogs.com/eqnedit.php?latex=g%5Cleft(.%5Cright)#0) and its inverse is [](https://www.codecogs.com/eqnedit.php?latex=g%5E%7B-1%7D%5Cleft(.%5Cright)#0); in other words, if [](https://www.codecogs.com/eqnedit.php?latex=g(A)%20%3D%20B#0), then [](https://www.codecogs.com/eqnedit.php?latex=g%5E%7B-1%7D(B)%20%3D%20A#0).[[1]](#footnote-1) In the case of the binary variable we are working with, recall that we are interested in modeling the probability of a correct response, [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0), for each person. The link function relates the value of omega to the value of Shape

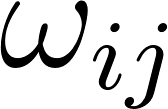
Description automatically generated with medium confidence:

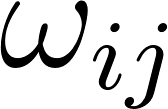
[](https://www.codecogs.com/eqnedit.php?latex=g(p_%7Bij%7D)%20%3D%20%5Comega_%7Bij%7D#0) (3)

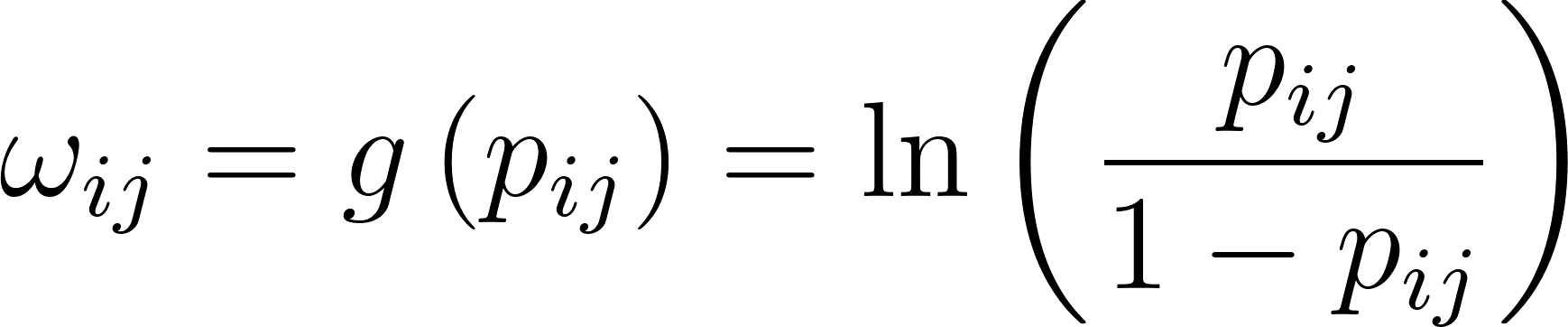
And if we wish to solve for the probability [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0), we can use the inverse link function:

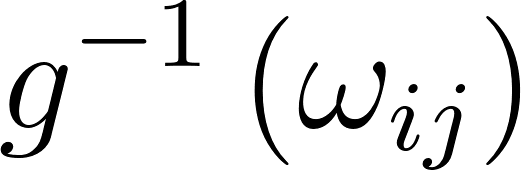
[](https://www.codecogs.com/eqnedit.php?latex=g%5E%7B-1%7D(%5Comega_%7Bij%7D)%20%3D%20p_%7Bij%7D#0) (4)

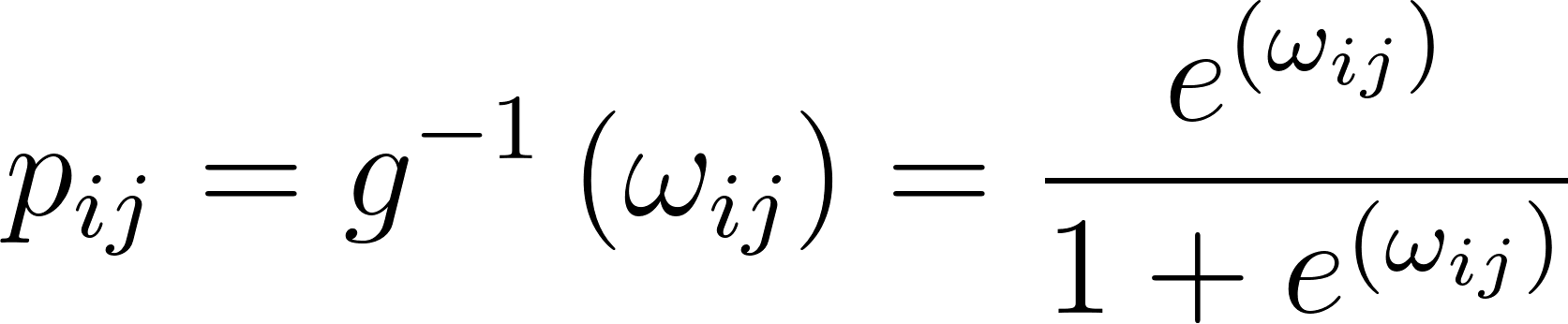
What exactly is our link function? These can be any type of function meeting a few mathematical requirements, and different link functions yield different types of results.

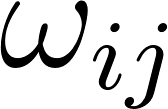
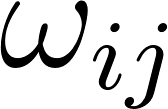
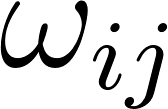
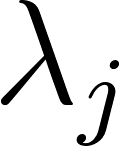
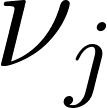
The link function is chosen based on the type of outcome we wish to model. Probabilities must be positive, and they must take a value between zero and one. Thus, our link function needs to take every value of linear predictor [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0), which can be both positive and negative and of any magnitude, and translate it into a positive number between zero and one. An ideal choice for this is what is known as the logit link function.

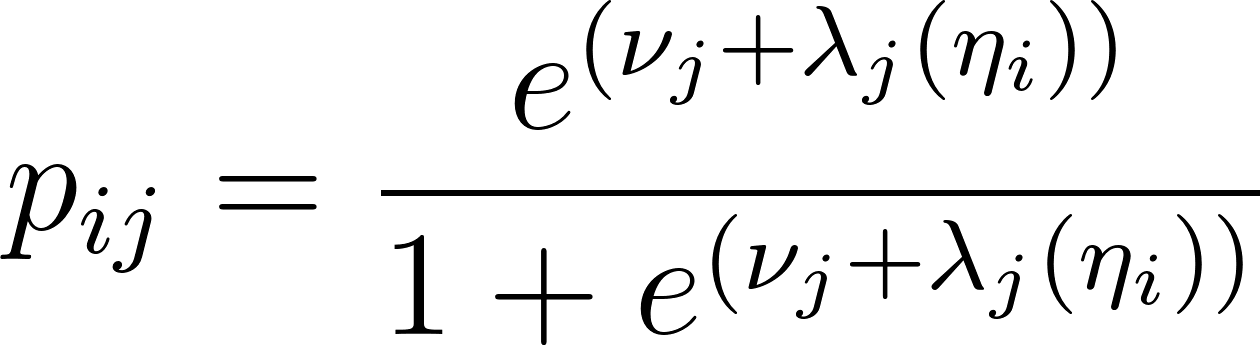
We could theoretically apply the link function, [](https://www.codecogs.com/eqnedit.php?latex=g%5Cleft(p_%7Bij%7D%5Cright)#0), to obtain [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0), as follows:

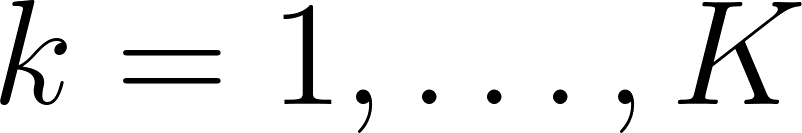
[](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D%20%3D%20g%5Cleft(p_%7Bij%7D%5Cright)%20%3D%20%5Cln%5Cleft(%5Cfrac%7Bp_%7Bij%7D%7D%7B1-p_%7Bij%7D%7D%5Cright)#0) (5)

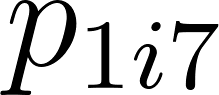
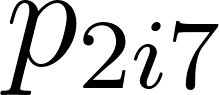
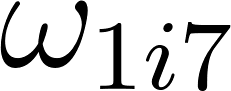
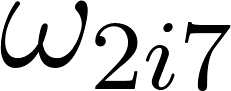
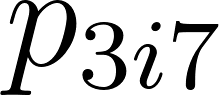
However, we are more interested in applying the inverse function, [](https://www.codecogs.com/eqnedit.php?latex=g%5E%7B-1%7D%5Cleft(%5Comega_%7Bij%7D%5Cright)#0), to obtain [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0). Though this may not be intuitive unless one has worked with logistic regressions before, the inverse of the function given in Equation 5 is:

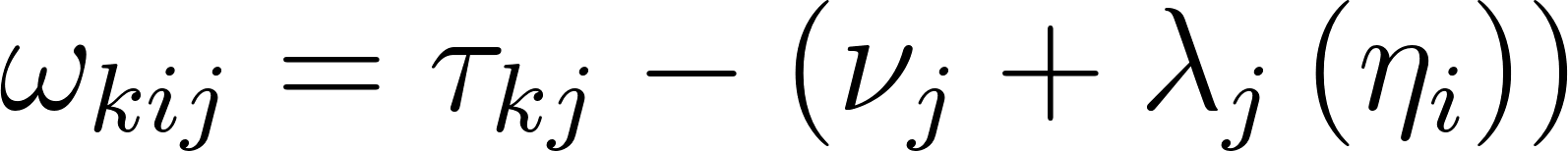
[](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D%20%3D%20g%5E%7B-1%7D%5Cleft(%5Comega_%7Bij%7D%5Cright)%20%3D%20%5Cfrac%7Be%5E%7B%5Cleft(%5Comega_%7Bij%7D%5Cright)%7D%7D%7B1%20%2Be%5E%7B%5Cleft(%5Comega_%7Bij%7D%5Cright)%7D%7D%20#0) (6)

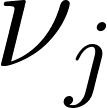
At this point, we have a way to obtain the probability, [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0), from the linear predictor [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0). In combination with our expression for [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0) in Equation 4, this is everything we need. Recall that [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bij%7D#0) is simply a linear function of [](https://www.codecogs.com/eqnedit.php?latex=%5Ceta_i#0), weighted by the loading [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_j#0) and incremented by the intercept [](https://www.codecogs.com/eqnedit.php?latex=%5Cnu_j#0). So, to tie everything together, we can combine Equations 4 and 6 to obtain the following:

[](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D%20%3D%20%5Cfrac%7Be%5E%7B%5Cleft(%5Cnu_%7Bj%7D%20%2B%20%5Clambda_%7Bj%7D%5Cleft(%5Ceta_%7Bi%7D%5Cright)%5Cright)%7D%7D%7B1%20%2Be%5E%7B%5Cleft(%5Cnu_%7Bj%7D%20%2B%20%5Clambda_%7Bj%7D%5Cleft(%5Ceta_%7Bi%7D%5Cright)%5Cright)%7D%7D%20#0) (7)

The above formulation is appropriate for binary items, such as items 1-5 in our example. However, we also have ordinal items in our example; item 6 is a 5-level ordinal variable and item 7 is a 3-level ordinal variable. For these models, we use what is referred to as the *cumulative logit* formulation. When we work with ordinal items, we are also modeling a probability, and we therefore use the same logit function as Equation 6. However, we are modeling a slightly different probability than if we were modeling a binary item. Suppose we have a *K*- level ordinal item. Then we are no longer modeling the probability of endorsing a given item, as in the previous example, but modeling the probability of giving a response in the [](https://www.codecogs.com/eqnedit.php?latex=k%5E%7Bth%7D#0) category or lower ([](https://www.codecogs.com/eqnedit.php?latex=k%20%3D%201%2C%20%5Cldots%2C%20K#0)), [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bkij%7D#0). This corresponds to a linear predictor denoted [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bkij%7D#0).

So suppose that we are working with item 7, which has three levels (*K* = 3): 1, 2, or 3. Then [](https://www.codecogs.com/eqnedit.php?latex=p_%7B1i7%7D#0) is the probability of giving a response of 1 and [](https://www.codecogs.com/eqnedit.php?latex=p_%7B2i7%7D#0) is the probability of giving a response of either 1 or 2. For each of these, we must calculate a linear predictor, [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7B1i7%7D#0) and [](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7B2i7%7D#0), respectively, which we then put into Equation 6 to get the corresponding probabilities. Note that we do not calculate [](https://www.codecogs.com/eqnedit.php?latex=p_%7B3i7%7D#0) - we only calculate *K* - 1 probabilities, because the probability of giving a response of *K* or lower is necessarily 1. In order to obtain these probabilities, we alter the equation for the linear predictor to include a new parameter, known as the threshold parameter, [](https://www.codecogs.com/eqnedit.php?latex=%5Ctau_%7Bkj%7D#0). For each category up to *K -* 1, we model the linear predictor as follows:

[](https://www.codecogs.com/eqnedit.php?latex=%5Comega_%7Bkij%7D%20%3D%20%5Ctau_%7Bkj%7D%20-%20%5Cleft(%5Cnu_j%20%2B%20%5Clambda_j%5Cleft(%5Ceta_i%5Cright)%5Cright)#0) (7)

In general, the intercept term [](https://www.codecogs.com/eqnedit.php?latex=%5Cnu_j#0) is set to zero so that all thresholds can be identified. We can think of the threshold parameter [](https://www.codecogs.com/eqnedit.php?latex=%5Ctau_%7Bkj%7D#0) as the value of the latent variable one must exceed to endorse the [](https://www.codecogs.com/eqnedit.php?latex=k%5E%7Bth%7D#0) category. We have explored this formulation incompletely here – for our purposes it is mainly helpful to know that [](https://www.codecogs.com/eqnedit.php?latex=%5Ctau_%7Bkj%7D#0) is a category-specific threshold which relates to the probability of endorsing that specific category. However, readers who are interested in employing this formulation are referred to other work where the assumptions and logic of this model are explained in more detail. In particular, Bauer and Hussong (2009) explain this formulation fully; it is introduced as Equation 5 in that paper.

Note that there are many cases in which calculations like the ones performed above are not necessary, because we are working with a continuous outcome and we can assume normally distributed residuals. This would be the case for item 8 in our example. In this case, we do not need to generate a linear predictor - we can simply proceed with the formulation of a latent variable model given in the previous section. We can also use different link functions for different items in a given dataset, which is exactly what we are doing here, using a logit link formulation for items 1-5, cumulative logit formulation for items 6-7, and a standard linear model for item 8.

1. [↑](#footnote-ref-1)