The suitable choice of model here refers to using a Binomial likelihood. The number of successful (Nsucc) putts out of a number of attempts (Ntrys) is modelled as being Binomially distributed with a probability of success parameter (p say) that depends on some function of the distance.

Each model you fit specifies a different function relating the distance to the probability of success. Since the Binomial probability parameter p must lie between 0 and 1 we should first transform our distance function to this interval. Looking at this in reverse, we want a function of p that transforms to the $[-\infty, \infty]$. interval. The logit (log-odds) function is a good choice for this, as per the second GLM example in Section 3 of the lab sheet, because logit(p) could be any real number when p is between 0 and 1.

Now what can we use for the function of distance that logit(p) is equal to? In the usual linear model approach we could specify that for each observation:

$$\text{logit}(p_i) = \alpha + \beta(d_i - \bar{d})$$

This is the model used in `putting_linear.model`.

But maybe the logit of probability of success is related to the distance in some other way? As you move away from a target the area around the target grows quadratically rather than linearly. So maybe including the square of the distance is a good idea. But the putt would still need to travel the right distance so let's leave that linear term in. `putting_quadratic.model` therefore uses the model

$$\text{logit}(p_i) = \alpha + \beta_1(d_i - \bar{d})) + \beta_2(d_i - \bar{d})^2$$

Naturally we could add more terms with higher powers of the distance. How do we choose the "best" model? We'll fit a few and then use DIC to compare between them using `putting.R`. Of course you could try other models than just the ones I've specified here. In any case, select the one with the lowest DIC.

It's nice to see what the model looks like plotted against the data. `plot_fits.R` allows you to visualise the various posterior mean curves along with the data as a function of distance.

In order to estimate the mean success rate at 5, 10, and 30 feet (note that the last one is outside the range of the data and is thus extrapolating using the model) we need to include lines in either the model files or after running MCMC in `R`. We could simply use the posterior mean values of α and $\boldsymbol{\beta}$ and fill in to the linear model equation for each model using distance equal to 5, 10, or 30. But since we're Bayesians we prefer to get the posterior predictive distributions for these 3 distances. Doing it in the model files we first calculate the logit(p) for the current samples of α and β and then transform this using the inverse of the logit function which is $\frac{e^x}{1+e^x}$. e.g. at 5 feet in the linear model:

```
logitpstar5 = alpha+beta*(5-mean(dist[]))
pstar5 = exp(logitpstar5)/(1+exp(logitpstar5))
```