Other Generalized Linear Models

Logistic Binomial Model For count data we have discussed Poisson and Negative-Binomial sampling models. *It is also possible to use a Binomial distribution, but know that the support of the response will not be countably infinite.*

A common example of binomial data would be free throw shooting for basketball players or batting data for baseball players.

```r
batting <- read_csv('http://math.montana.edu/ahoegh/teaching/stat491/data/BattingAverage.csv') %>%
  mutate(NotHits = AtBats - Hits)
```

```
## Parsed with column specification:
## cols(
##   Player = col_character(),
##   PriPos = col_character(),
##   Hits = col_double(),
##   AtBats = col_double(),
##   PlayerNumber = col_double(),
##   PriPosNumber = col_double()
## )
```

```r
batting %>% sample_n(5)
```

```
## # A tibble: 5 x 7
##   Player           PriPos        Hits AtBats PlayerNumber PriPosNumber NotHits
##   <chr>            <chr>        <dbl>  <dbl>        <dbl>        <dbl>   <dbl>
## 1 Martin Maldonado Catcher         62    233          539            2     171
## 2 Jeff Suppan      Pitcher          1     10          842            1       9
## 3 Nathan Eovaldi   Pitcher          3     32          252            1      29
## 4 Eric Young       Center Field    55    174          940            8     119
## 5 Reed Johnson     Left Field      78    269          445            7     191
```

The logistic-binomial framework is written as:

$$y_i \sim Binomial(n_i, p_i), \tag{1}$$
$$logit(p_i) = X_i \beta \tag{2}$$

```r
log_binom <- stan_glm(cbind(Hits, NotHits ) ~ PriPos - 1,
              family = binomial(link = "logit"), data = batting, refresh = 0)

print(log_binom, digits = 2)
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      cbind(Hits, NotHits) ~ PriPos - 1
##  observations: 948
##  predictors:   9
## ------
##                     Median MAD_SD
## PriPos1st Base     -1.05   0.02
## PriPos2nd Base     -1.07   0.02
## PriPos3rd Base     -1.02   0.02
## PriPosCatcher      -1.11   0.02
## PriPosCenter Field -1.03   0.02
## PriPosLeft Field   -1.05   0.02
## PriPosPitcher      -1.91   0.04
## PriPosRight Field  -1.03   0.02
## PriPosShortstop    -1.07   0.02
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Overdispersion can also occur with binomial data. Recall that the variance of binomial trials is $np(1-p)$. *Then define the residuals as* $z_i = \frac{y_i - \hat{y}_i}{sd(\hat{y}_i)}$.

Then the z_i *terms should be approximately iid N(0,1). A formal test for* $\sum z_i^2$ *using a* χ^2 *distribution can be used to detect overdispersion.*

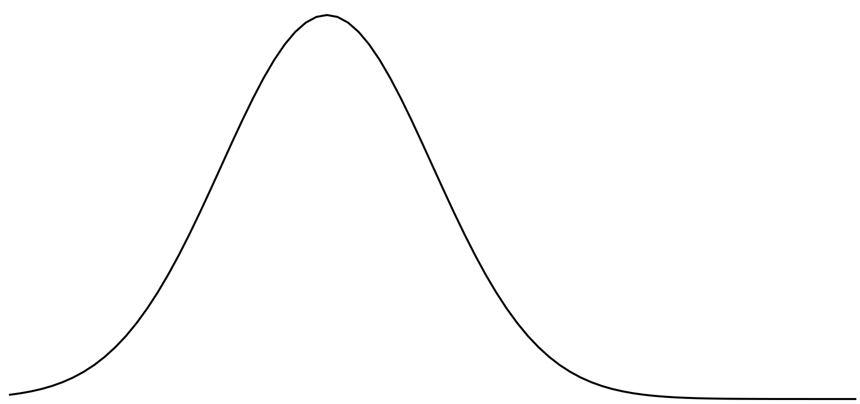
Often hierarchical models will solve some of these issues, otherwise an overdispersion model can be formulated with variance equal to $\omega np(1-p)$. *See* `brm` *or write your own in stan.*

Probit Model *Consider an alternative link function for binary/binomial data.*

$$
\begin{aligned}
y_i \sim & Binomial(n_i, p_i), && (3)\\
\Phi^{-1}(p_i) = & X_i\beta && (4)\\
p_i = & \Phi(X_i\beta), && (5)
\end{aligned}
$$

where $\Phi()$ *is the cumulative distribution function for a standard normal random variable.*

This model is a latent data model, which are very common and useful in statistics. We assume there is an underlying continuous random variable that is mapped to a standard normal distribution.



where

$$y_i = \begin{cases} 1 \text{ if } & z_i > 0 \\ 0 \text{ if } & z_i < 0 \end{cases}$$

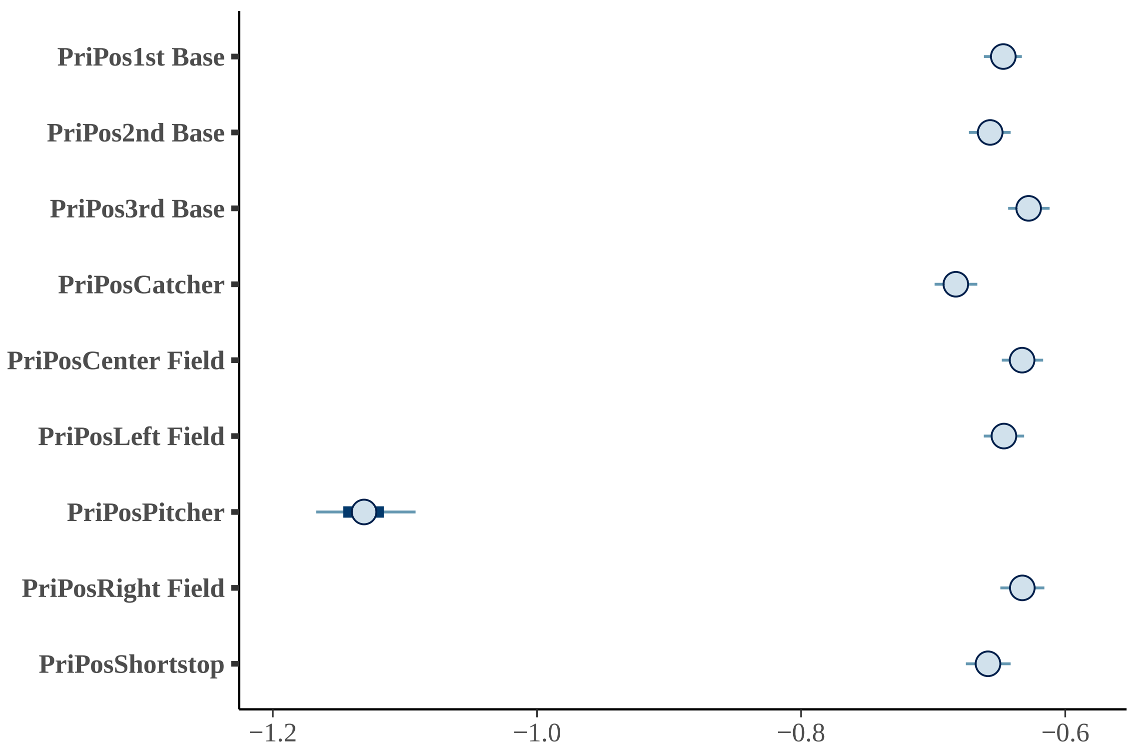
$$
\begin{aligned}
z_i = & X_i\beta + \epsilon\\
\epsilon \sim & N(0,1)
\end{aligned}
$$

Note $\epsilon \sim N(0,1)$ is a necessary constraint for this model.

```r
probit_binom <- stan_glm(cbind(Hits, NotHits ) ~ PriPos - 1,
              family = binomial(link = "probit"), data = batting, refresh = 0)
print(probit_binom)
```

```
## stan_glm
##  family:       binomial [probit]
##  formula:      cbind(Hits, NotHits) ~ PriPos - 1
##  observations: 948
##  predictors:   9
## ------
##                      Median MAD_SD
## PriPos1st Base      -0.6    0.0
## PriPos2nd Base      -0.7    0.0
## PriPos3rd Base      -0.6    0.0
## PriPosCatcher       -0.7    0.0
## PriPosCenter Field -0.6    0.0
## PriPosLeft Field    -0.6    0.0
## PriPosPitcher       -1.1    0.0
## PriPosRight Field   -0.6    0.0
## PriPosShortstop     -0.7    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
plot(probit_binom)
```



So what do the coefficients mean in this model?

```
log_binom$coefficients
```

```
##     PriPos1st Base     PriPos2nd Base     PriPos3rd Base      PriPosCatcher 
##          -1.051951          -1.068610          -1.020217          -1.112554 
## PriPosCenter Field   PriPosLeft Field      PriPosPitcher  PriPosRight Field 
##          -1.028060          -1.050782          -1.906119          -1.027139 
##    PriPosShortstop 
##          -1.070992
```

```
probit_binom$coefficients
```

```
##     PriPos1st Base     PriPos2nd Base     PriPos3rd Base      PriPosCatcher 
##         -0.6469227         -0.6568591         -0.6277234         -0.6828566 
## PriPosCenter Field   PriPosLeft Field      PriPosPitcher  PriPosRight Field 
##         -0.6326863         -0.6464448         -1.1309219         -0.6325274 
##    PriPosShortstop 
##         -0.6584987
```

```
invlogit(log_binom$coefficients) * 1000
```

```
##     PriPos1st Base     PriPos2nd Base     PriPos3rd Base      PriPosCatcher 
##           258.8506           255.6674           264.9851           247.3950 
## PriPosCenter Field   PriPosLeft Field      PriPosPitcher  PriPosRight Field 
##           263.4604           259.0749           129.4175           263.6391 
##    PriPosShortstop 
##           255.2145
```

```
pnorm(probit_binom$coefficients) * 1000
```

```
##     PriPos1st Base     PriPos2nd Base     PriPos3rd Base      PriPosCatcher 
##           258.8410           255.6358           265.0926           247.3487 
## PriPosCenter Field   PriPosLeft Field      PriPosPitcher  PriPosRight Field 
##           263.4692           258.9957           129.0440           263.5211 
##    PriPosShortstop 
##           255.1089
```