Generalized Linear Models, part 1

Frank Edwards

3/27/2020

Generalized linear models

The components of a model

- Likelihood of the data (the small-world data generator): *e.g.* $y \sim Normal(\mu, \sigma)$
- The linear function (converts likelihood parameters to linear combinations of predictors): *e.g.* $\mu = \alpha + \beta x$
- Priors (our initial beliefs about plausible parameter values) *e.g.*:

$$\alpha \sim Normal(0, 1)$$

$$\beta \sim Normal(0, 0.5)$$

$$\sigma \sim Exp(1)$$

Limits of the Normal model

There's no reason we must be constrained to the Normal (Gaussian) likelihood for data:

$$y \sim Normal(\mu, \sigma)$$

While powerful, the Normal distribution forces us into assumptions that may not suit our data.

McElreath proposes *maximum entropy* as a set of principals to guide likelihood selection. Which model meets the constraints of the data while maximizing the number of ways a distribution could have arisen.

Limits of the Normal model

What if our outcome is binary, and we wish to model probability?

- Nothing constrains a model to the [0,1] probability scale

Limits of the Normal model

What if our outcome is an event count?

- The continuous normal likelihood will predict on a real number scale, inappropriate for non-negative integers (births, deaths, books read, unemployment filings, infections, etc)

Limits of the Normal model

What if our outcome is categorical?

- The continuous normal model doesn't easily model categorical outcomes (favorite brand of soda, choice of college major)

Generalizing the linear model

If we have a good reason, we could choose any other statistical distribution for the likelihood. A likelihood can take the following general form

$$\text{outcome} \sim \text{Probability distribution}(\text{parameter})$$

But we generally want to include predictor variables in our model that may help to explain variation in the outcome. That's why we include a *linear predictor* with the following general form

$$\text{parameter} = \text{intercept} + \text{slope} \times \text{predictor}$$

Mapping a linear predictor to any parameter scale

What if we want to map our linear predictions to some other scale like probability [0,1]?

Mapping a linear predictor to any parameter scale

What if we want to map our linear predictions to some other scale like probability [0,1]?

The link function

We can use a link function g to transform our linear predictor from a continuous linear scale to any other scale. The general form is:

$$g(\text{parameter}) = \text{intercept} + \text{slope} \times \text{predictor}$$

The link function

We can use a link function g to transform our linear predictor from a continuous linear scale to any other scale. The general form is:

$$g(\text{parameter}) = \text{intercept} + \text{slope} \times \text{predictor}$$

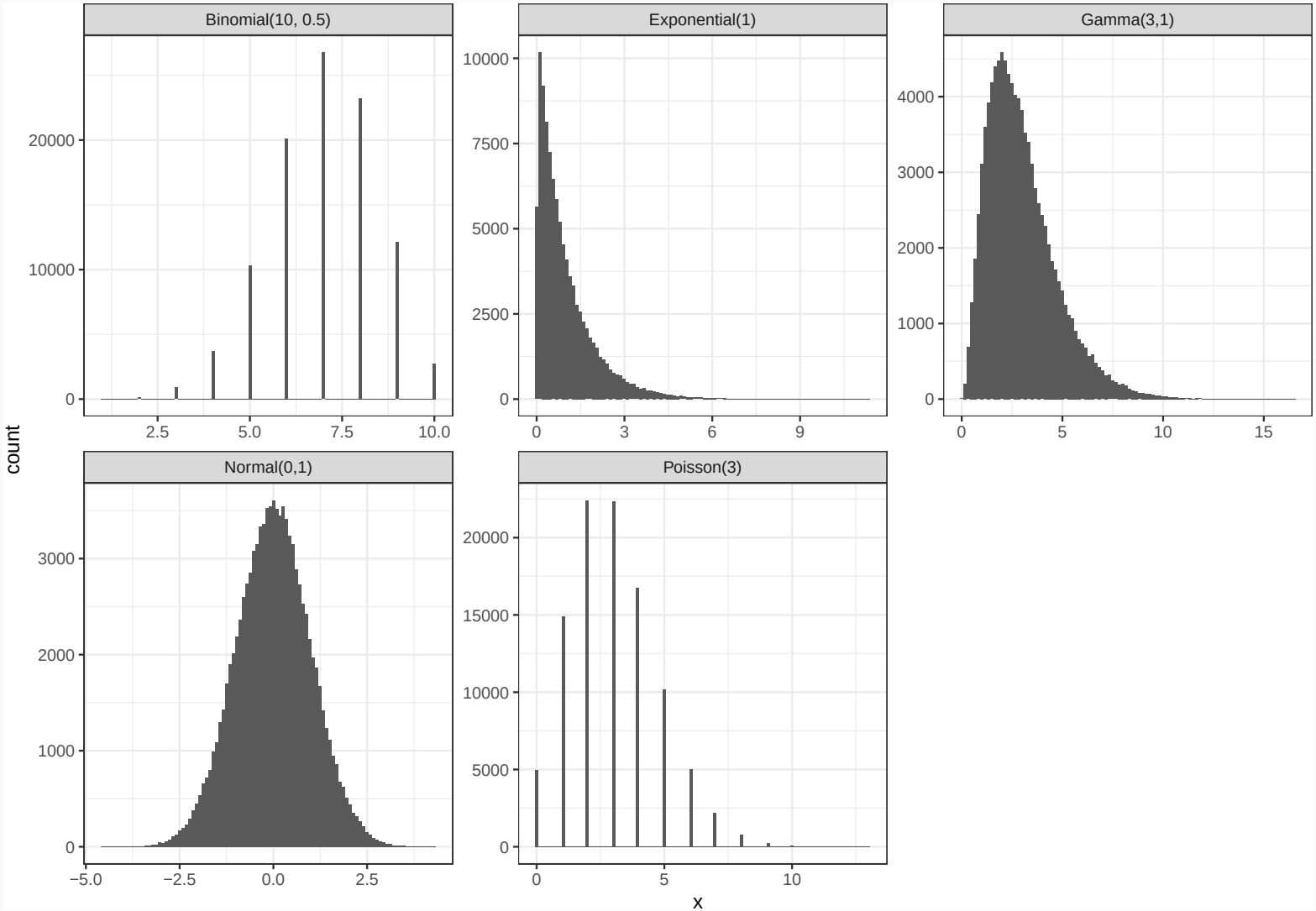
or:

$$\text{parameter} = g^{-1}(\text{intercept} + \text{slope} \times \text{predictor})$$

Diverse likelihood functions

- Continuous outcome: the Normal distribution
- Continuous non-negative outcome: the exponential distribution
- Continuous non-negative outcome with mode above zero: the Gamma distribution
- Binary outcome: the binomial distribution
- Categorical outcome: the multinomial distribution
- Count outcome: the Poisson distribution

Common probability distributions used in GLMs



Logistic regression

Read in grad school admissions data

```
admissions <- read_csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
head(admissions)
```

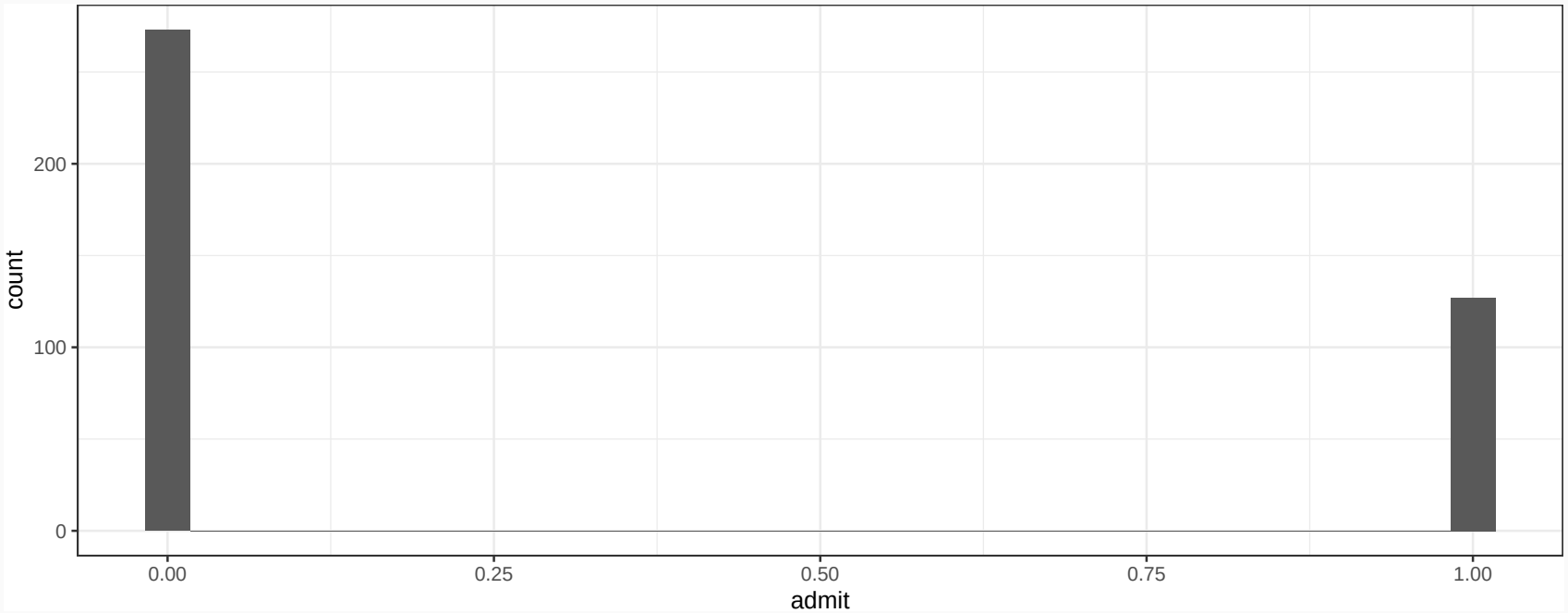
```
## # A tibble: 6 x 4
##   admit   gre   gpa  rank
##   <dbl> <dbl> <dbl> <dbl>
## 1     0   380  3.61     3
## 2     1   660  3.67     3
## 3     1   800  4        1
## 4     1   640  3.19     4
## 5     0   520  2.93     4
## 6     1   760  3        2
```

```
nrow(admissions)
```

```
## [1] 400
```

Evaluate distribution of binary admission variable

```
ggplot(admissions, aes(x = admit)) +
  geom_histogram()
```



How do GRE scores relate to admission?

Let's consider the linear regression model, where GRE.s is a mean-centered SD scaled GRE score, so a 1 is one unit above the mean:

$$admit_i \sim Normal(\mu, \sigma)$$

$$\mu = \alpha + \beta \times \mathrm{GRE.s}$$

$$\alpha \sim N(0.5, 1)$$

$$\beta \sim N(0, 1)$$

$$\sigma \sim Exp(1)$$

Because this is a Normal likelihood with a binary outcome, we call it a *linear probability model*.

Fit the model using MCMC

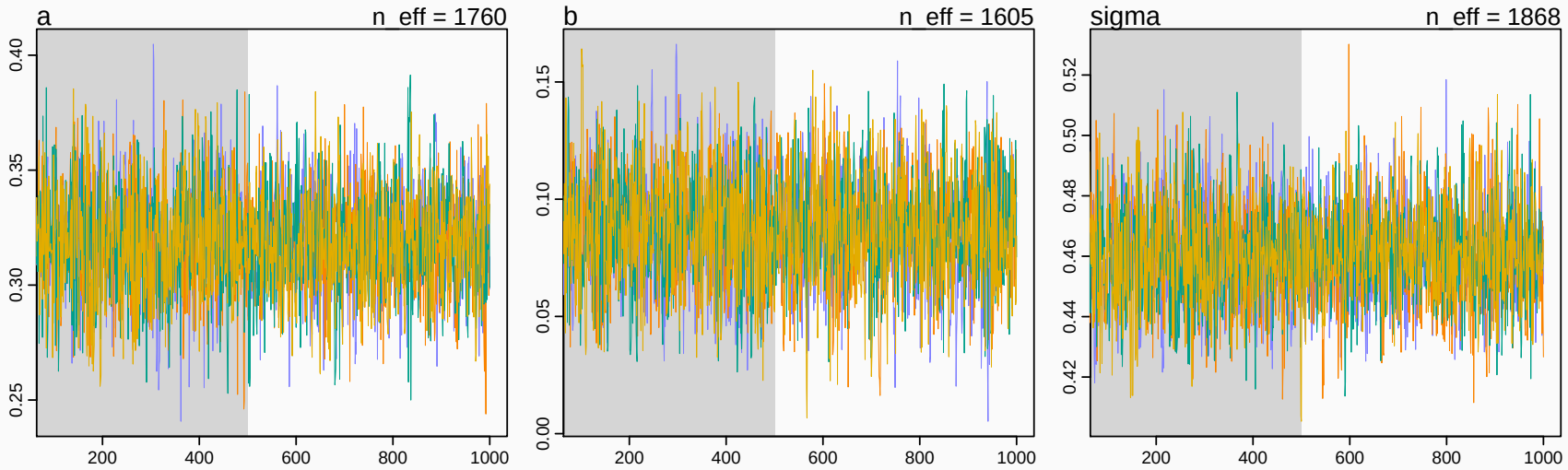
```
### DON'T use PERIODS in VARIABLE names for Stan
### I have done it 5 times already...
d<-admissions %>%
  mutate(gre_s = (gre - mean(gre))/sd(gre)) %>%
  select(admit, gre_s)

d<-as.data.frame(d)

m0<-ulam(alist(
  admit ~ dnorm(mu, sigma),
  mu<- a + b * gre_s,
  a ~ dnorm(0.5, 1),
  b ~ dnorm(0, 1),
  sigma ~ dexp(1)
), data = d, cores = 2, chains = 4)
```

Checking convergence diagnostics

```
precis(m0)
```

```
##              mean         sd       5.5%     94.5%    n_eff    Rhat4
## a     0.31807266 0.02189190 0.28380833 0.3527726 1760.341 1.000326
## b     0.08651567 0.02310262 0.04939676 0.1226978 1604.785 1.001241
## sigma 0.46053356 0.01601925 0.43590867 0.4869228 1867.887 1.000221
```

```
traceplot(m0)
```



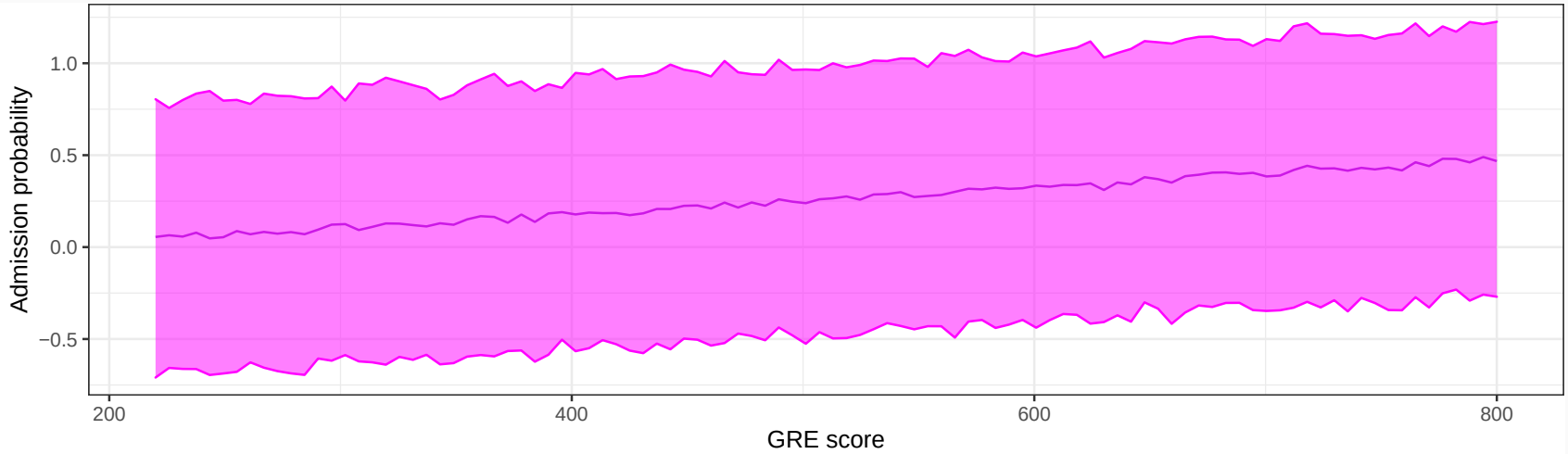
Evaluating the fit

Let's predict admissions probabilities for the full range of GRE scores.

```
## set up counterfactual scenarios
sim_dat<-data.frame(gre_s =
                      seq(min(d$gre_s),
                          max(d$gre_s),
                          length.out = 100))
## draw from posterior predictive distribution
preds<-sim(m0, sim_dat)
## set up for plot
sim_dat<-sim_dat %>%
  mutate(admit_mn = apply(preds, 2, mean),
         admit_lwr = apply(preds, 2, PI)[1,],
         admit_upr = apply(preds, 2, PI)[2,])
```

Visualize the estimated relationship between GRE scores and admissions (posterior prediction)

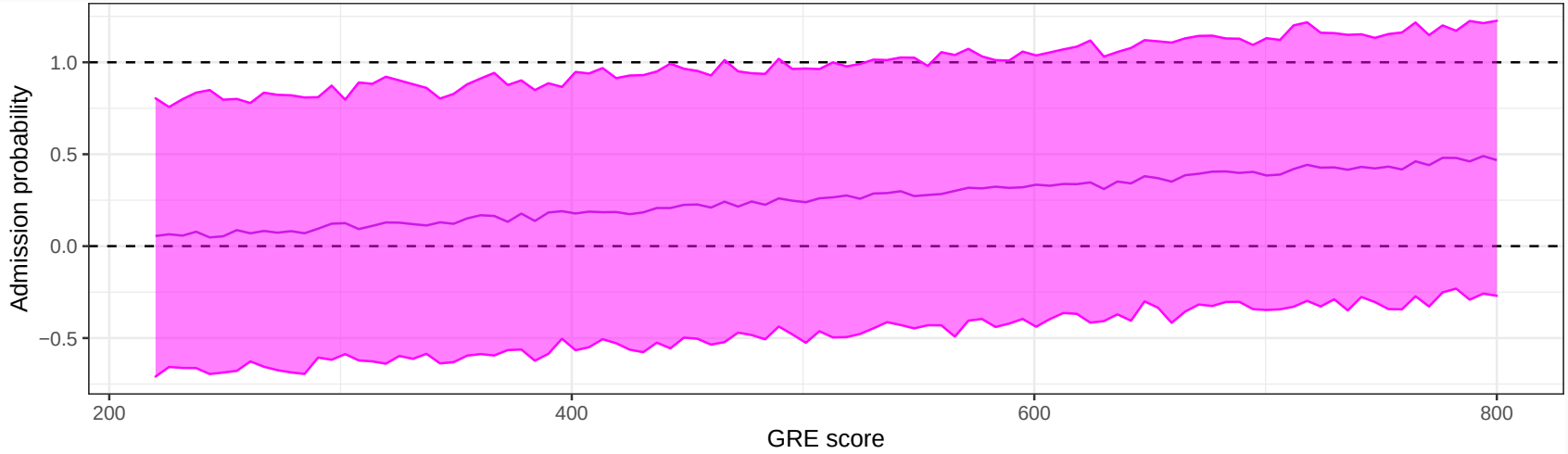
```
ggplot(sim_dat, aes(x = gre_s * sd(admissions$gre) + mean(admissions$gre),
                    y = admit_mn,
                    ymin = admit_lwr,
                    ymax = admit_upr)) +
  geom_line(color = "darkorchid3") +
  geom_ribbon(alpha = 0.5, fill = "magenta1",
              color = "magenta1") +
  labs(y = "Admission probability", x = "GRE score")
```



Thank you to Dorothy for picking wonderful colors. Notice any problems?

One of the limits of the linear probability model

```
ggplot(sim_dat, aes(x = gre_s * sd(admissions$gre) + mean(admissions$gre),
                    y = admit_mn,
                    ymin = admit_lwr,
                    ymax = admit_upr)) +
  geom_hline(yintercept = 0, lty = 2) +
  geom_hline(yintercept = 1, lty = 2) +
  geom_line(color = "darkorchid3") +
  geom_ribbon(alpha = 0.5, fill = "magenta1",
              color = "magenta1") +
  labs(y = "Admission probability", x = "GRE score")
```



The Bernoulli distribution

Binary outcomes can be described easily by the Bernoulli distribution

$$y \sim Bernoulli(p)$$

$$\Pr(y = 1) = p = 1 - \Pr(y = 0)$$

$$E(y) = \bar{y} = p$$

$$Var(y) = p(1 - p)$$

Bernoulli variables are a special case of the Binomial distribution, with $n = 1$

What is E(admit) and var(admit)

```
### E(y) = p = sum(y==1)/n
mean(admissions$admit)
```

```
## [1] 0.3175
```

```
sum(admissions$admit==1)/nrow(admissions)
```

```
## [1] 0.3175
```

```
### var(y) = p(1-p)
var(admissions$admit)
```

```
## [1] 0.2172368
```

```
mean(admissions$admit) * (1 - mean(admissions$admit))
```

```
## [1] 0.2166937
```

Let's treat admission as a Bernoulli / Binomial random variable

Because admission is a binary outcome, ["admitted", "not admitted"], a Bernoulli (equivalent to Binomial with n=1) model is the most appropriate. It is the distribution that satisfies the constraints of the data generating process, while also satisfying maximum entropy.

$$\text{admit}_i \sim \mathit{Binomial}(n = 1, p_i)$$

Let's treat admission as a Bernoulli / Binomial random variable

Because admission is a binary outcome, ["admitted", "not admitted"], a Bernoulli (equivalent to Binomial with n=1) model is the most appropriate. It is the distribution that satisfies the constraints of the data generating process, while also satisfying maximum entropy.

$$\text{admit}_i \sim \textit{Binomial}(n = 1, p_i)$$

We'll use a linear predictor with link function g

$$g(p_i) = \alpha + \beta \times \text{GRE}_i$$

Let's try a different approach

This Binomial distribution has one unobserved parameter, p. The parameter p is the probability of "success". So we need a link function that translates a continuous linear function onto a [0,1] probability space.

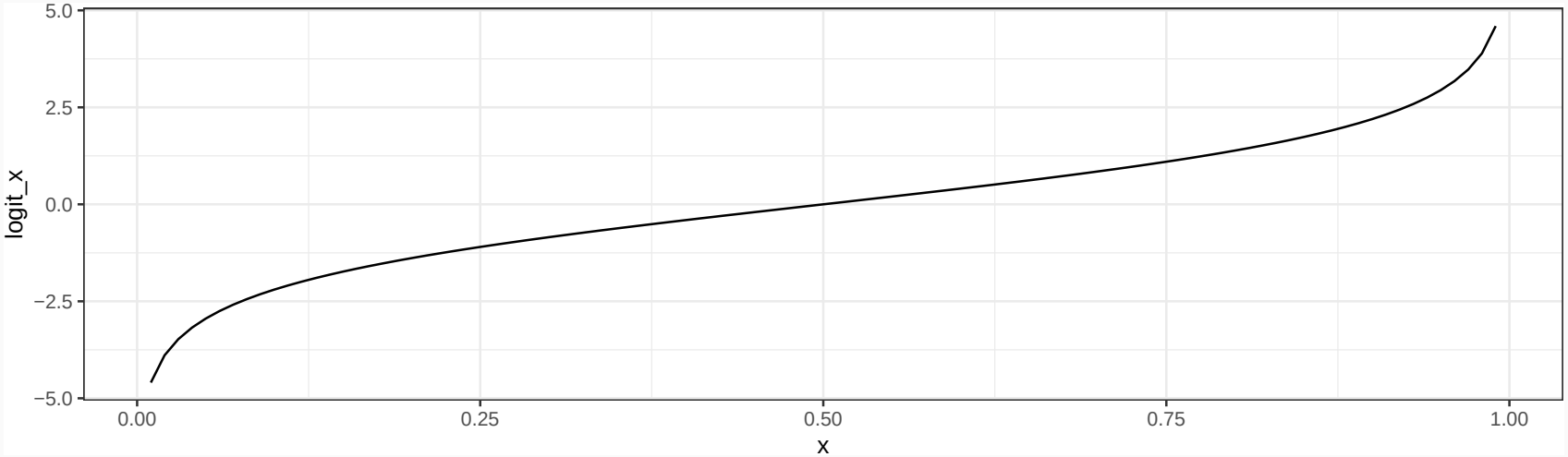
The logit function is constrained to [0,1], and serves this job well. Along with log, the logit is a very common link function for GLMs.

$$logit(p) = log\frac{p}{1-p}$$

The logit function

To translate a $[0, 1]$ probability to a continuous scale, we use the logit function: $logit(p) = log(\frac{p}{1-p})$

```
x<-seq(0.01, 0.99, 0.01)
logit_x = log(x/(1-x))
plot_dat<-data.frame(x = x, logit_x = logit_x)
ggplot(plot_dat, aes(x = x, y = logit_x)) +
  geom_line()
```

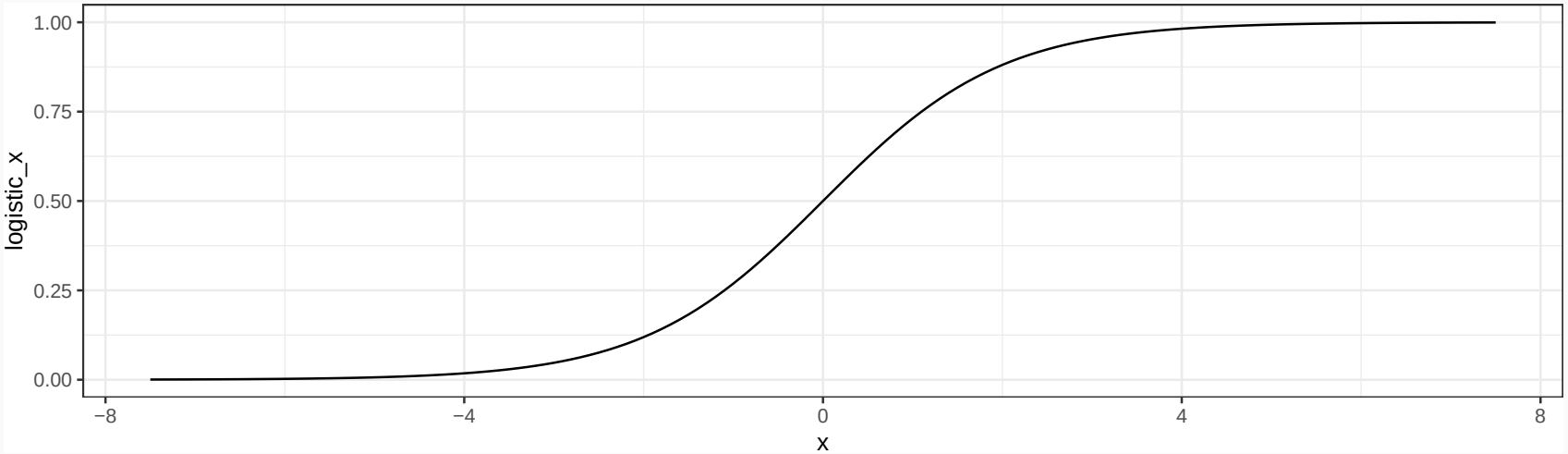


The inverse logit (logistic) function

To translate the linear predictor $[-\infty, \infty]$ to the probability $[0, 1]$ scale, we can use the inverse logit function (also called the logistic function)

$$logit^{-1}(x) = \frac{1}{1 + exp(-x)}$$

```
x<-seq(-7.5, 7.5, 0.1)
logistic_x<-1/(1 + exp(-x))
plot_dat<-data.frame(x = x, logistic_x = logistic_x)
ggplot(plot_dat, aes(x = x, y = logistic_x)) +
  geom_line()
```



A generalized linear model

We use a logit link function to move back and forth between a linear (logit) scale, and a probability (outcome) scale. I'll use weakly informative priors.

$$\mathrm{admit}_i \sim \mathit{Binomial}(1, p_i)$$

$$\mathit{logit}(p_i) = \alpha + \beta \times \mathrm{GRE}$$

$$\alpha \sim N(0, 2)$$

$$\beta \sim N(0, 2)$$

Estimating the model with ulam()

```
m1<-ulam(alist(
  admit ~ dbinom(1, p),
  logit(p) <- a + b * gre_s,
  a ~ dnorm(0, 2),
  b ~ dnorm(0,2)
), data = d, cores = 2, chains = 4)
```

Logistic regression parameter estimates

Remember that our linear predictor is now on the *logit* scale. This makes interpretation hard!

```
precis(m1)
```

```
##          mean        sd       5.5%      94.5%    n_eff     Rhat4
## a -0.7987652 0.1054576 -0.9687024 -0.6335077 1358.360 0.9992972
## b  0.4211905 0.1104706  0.2482838  0.5980747 1553.505 0.9988907
```

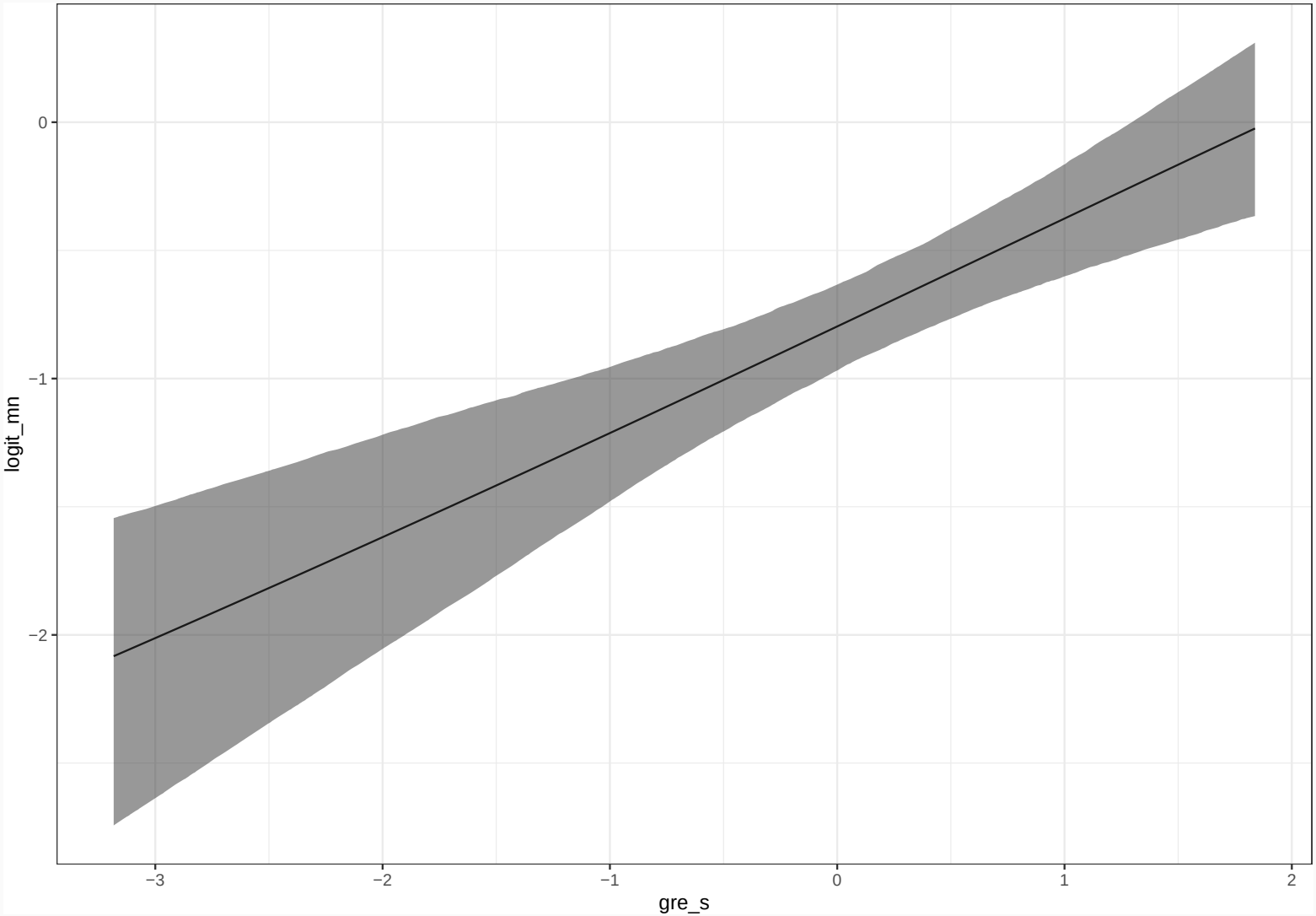
α is now the *log-odds* of admission when GRE scores at the mean. β is now the change in log-odds of admission when GRE scores increase by one SD. We can exponentiate these numbers to obtain the *odds*.

Odds are defined as $\frac{p}{1-p}$. The value of a change in odds on the probability scale depends on the probability of the event!

Visualizing the impact of GRE scores

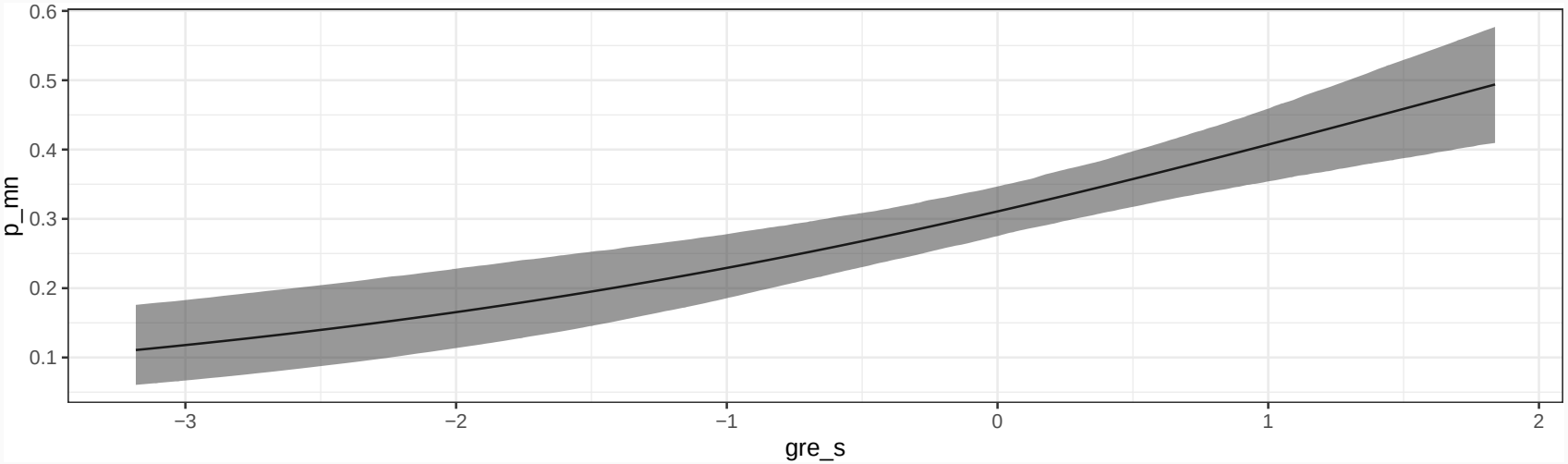
General advice for GLMs: Avoid over-interpreting parameter estimates, they are now non-linear and often multiplicative. Use the inverse-link functions to visualize relationships on the scale of the outcome (e.g. probability).

On the logit scale



On the probability scale

```
### define GRE scores over the range of observed
sim_dat<-data.frame(gre_s = seq(
  min(d$gre_s), max(d$gre_s), length.out = 500))
### link applies the link function for you!
posterior<-link(m1, sim_dat)
p_mn<-apply(posterior, 2, mean)
p_pi<-apply(posterior, 2, PI)

ggplot(sim_dat, aes(x = gre_s,
                    y = p_mn,
                    ymin = p_pi[1,],
                    ymax = p_pi[2,])) +
  geom_line() +
  geom_ribbon(alpha = 0.5)
```



Binomial (count) regression

Extending the binomial model

A logistic regression is a binomial regression with the special context that the parameter n is fixed at 1. It estimates individual level event probabilities.

Let's load different grad school admissions data from rethinking. We want to know whether female applicants are less likely to be admitted than male applicants.

```
data("UCBadmit")
d<-UCBadmit
```

Count data

```
head(d)
```

```
##   dept applicant.gender admit reject applications
## 1    A             male   512    313          825
## 2    A           female    89     19          108
## 3    B             male   353    207          560
## 4    B           female    17      8           25
## 5    C             male   120    205          325
## 6    C           female   202    391          593
```

We don't have individual admission data. These are individual-level data, aggregated to the department level by applicant gender, counts of individual outcomes.

Translating counts into proportions

```
d %>%
  group_by(applicant.gender) %>%
  summarise(admit_prop = sum(admit) / sum(applications))
```

```
## # A tibble: 2 x 2
##   applicant.gender admit_prop
##   <fct>                  <dbl>
## 1 female                 0.304
## 2 male                   0.445
```

Defining a model

The count of admissions in department *i* will be modeled with two likelihood parameters: the number of applicants, *n*, and the probability of admission *p*. We will use a logit link to define a linear function for *p* to include applicant gender as a predictor of admission rates.

$$admit_i \sim Binomial(n_i, p_i)$$

$$logit(p_i) = \alpha_{gender[i]}$$

$$\alpha \sim N(0, 1.5)$$

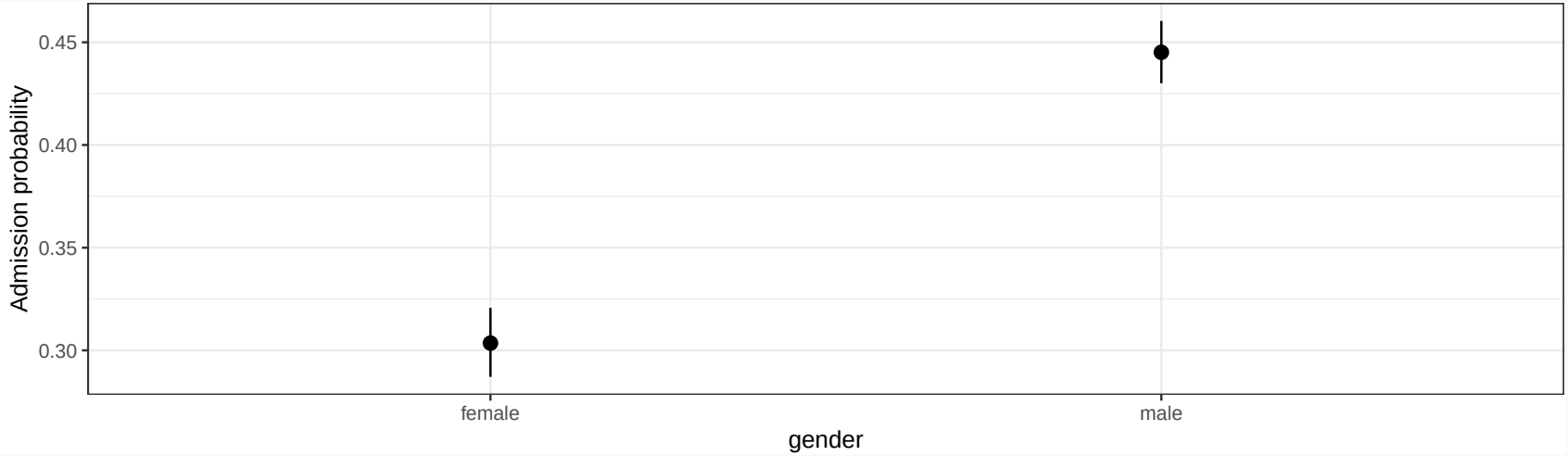
On the logit scale, $N(0, 1.5)$ is a weakly informative prior centered on p = 0.5 with substantial variance.

Estimating the model

```
d_slim<-d %>%
  mutate(gender = (applicant.gender == "female") + 1) %>%
  select(gender, admit, applications)

m2<-ulam(alist(
  admit ~ dbinom(applications, p),
  logit(p) <- a[gender],
  a[gender] ~ dnorm(0, 1.5)
), data = d_slim, cores = 2, chains = 4)
```

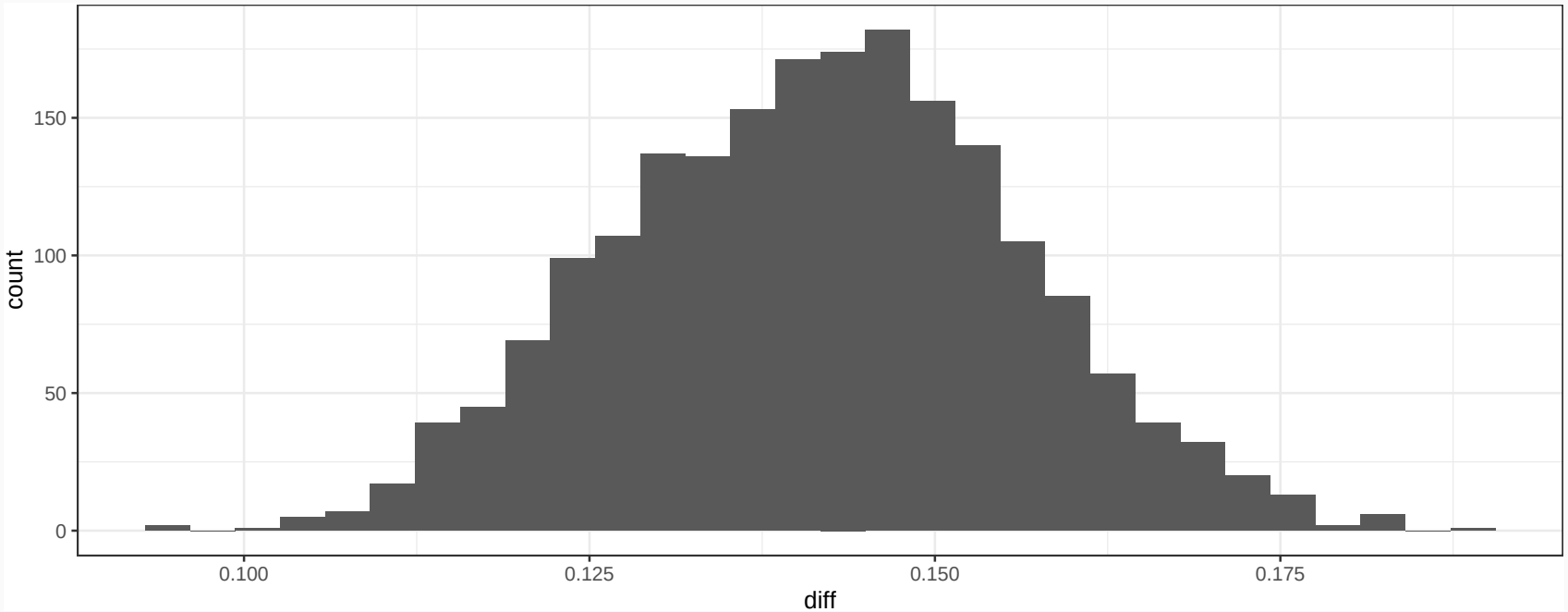
Interpreting the model: probability

```
sim_dat<-data.frame(gender = c(1, 2))
p_post<-link(m2, sim_dat)
sim_dat<-sim_dat %>%
  mutate(gender = ifelse(gender == 2, "female", "male"),
         p_mn = apply(p_post, 2, mean),
         p_upr = apply(p_post, 2, PI)[1,],
         p_lwr = apply(p_post, 2, PI)[2,])
ggplot(sim_dat, aes(x = gender, y = p_mn,
                    ymin = p_upr, ymax = p_lwr)) +
  geom_pointrange() +
  labs(y = "Admission probability")
```



Interpreting the model: difference in probability

```
plot_dat<-data.frame(diff = p_post[,1] - p_post[,2])
ggplot(plot_dat, aes(x = diff)) +
  geom_histogram()
```



Perhaps selecting to apply to more competitive programs plays a role?

$$admit_i \sim Binomial(n_i, p_i)$$

$$logit(p_i) = \alpha_{gender[i]} + \delta_{program[i]}$$

$$\alpha_{gender} \sim N(0, 1.5)$$

$$\delta_{program} \sim N(0, 1.5)$$

Estimating the model

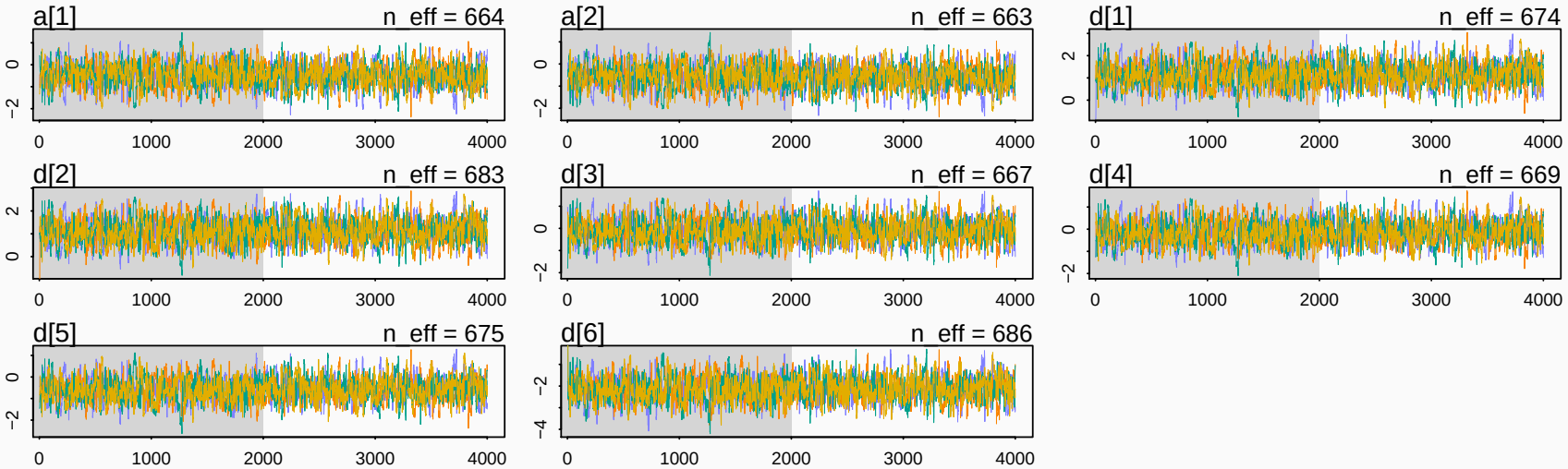
```
d_slim<-d %>%
  mutate(gender = as.numeric(applicant.gender),
         dept = as.numeric(dept)) %>%
  select(gender, admit, applications, dept)

m3<-ulam(alist(
  admit ~ dbinom(applications, p),
  logit(p) <- a[gender] + d[dept],
  a[gender] ~ dnorm(0, 1.5),
  d[dept] ~ dnorm(0, 1.5)
), data = d_slim, cores = 2, chains = 4, iter = 4000)
```

Checking convergence

```
precis(m3, depth =2)
```

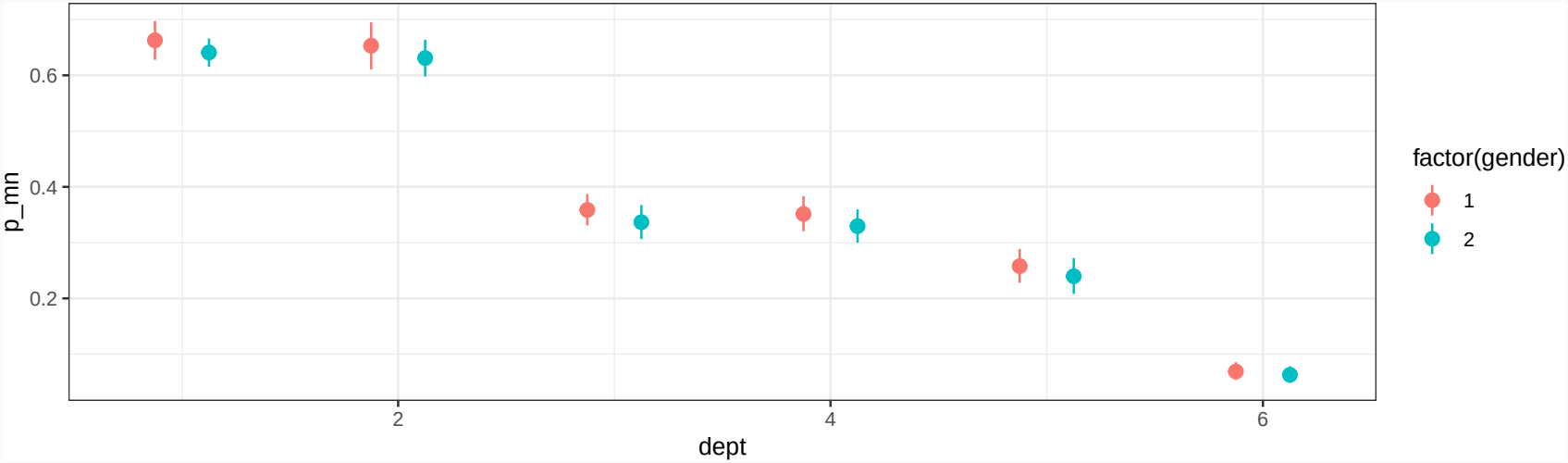
```
##             mean        sd       5.5%      94.5%    n_eff    Rhat4
## a[1] -0.51475867 0.5195253 -1.3753615  0.3139460 663.5331 1.007670
## a[2] -0.61235018 0.5187219 -1.4814136  0.2150939 663.1427 1.007622
## d[1]  1.19219115 0.5220750  0.3682534  2.0586129 674.4118 1.007307
## d[2]  1.14981819 0.5238702  0.3234592  2.0246662 682.8848 1.007190
## d[3] -0.06714671 0.5204576 -0.9091783  0.7987713 666.5278 1.007495
## d[4] -0.09894724 0.5223862 -0.9318525  0.7666352 668.8530 1.007684
## d[5] -0.54494294 0.5244659 -1.3832015  0.3229085 675.2379 1.007907
## d[6] -2.09908746 0.5339167 -2.9414954 -1.2225385 686.3046 1.007629
```

```
traceplot(m3)
```



Visualizing the results

```
sim_dat<-data.frame(dept = rep(1:6, 2),
                    gender = rep(1:2, each = 6))
p_post<-link(m3, sim_dat)
sim_dat<-sim_dat %>%
  mutate(p_mn = apply(p_post, 2, mean),
         p_lwr = apply(p_post, 2, PI)[1,],
         p_upr = apply(p_post, 2, PI)[2,])
ggplot(sim_dat, aes(x = dept, color = factor(gender),
                    y = p_mn, ymin = p_lwr, ymax = p_upr)) +
         geom_pointrange(position = position_dodge(width = 0.5))
```



Wrapping up

- GLMs extend the logic of the Normal model to a wide range of outcomes

Wrapping up

- GLMs extend the logic of the Normal model to a wide range of outcomes
- The Normal likelihood is not the best choice for many outcomes (restrictions of data coupled with maximum entropy principals)

Wrapping up

- GLMs extend the logic of the Normal model to a wide range of outcomes
- The Normal likelihood is not the best choice for many outcomes (restrictions of data coupled with maximum entropy principals)
- GLMs take diverse likelihood functions and link functions to transform linear predictors onto non-linear parameter spaces

Wrapping up

- GLMs extend the logic of the Normal model to a wide range of outcomes
- The Normal likelihood is not the best choice for many outcomes (restrictions of data coupled with maximum entropy principals)
- GLMs take diverse likelihood functions and link functions to transform linear predictors onto non-linear parameter spaces
- Homework: predicting survival on the titanic. See HW7.rmd and look for the data in ./HW/data/titanic.csv