Logistic regression

Rafał Urbaniak, Nikodem Lewandowski
(LoPSE research group, University of Gdansk)

Likelihoods so far

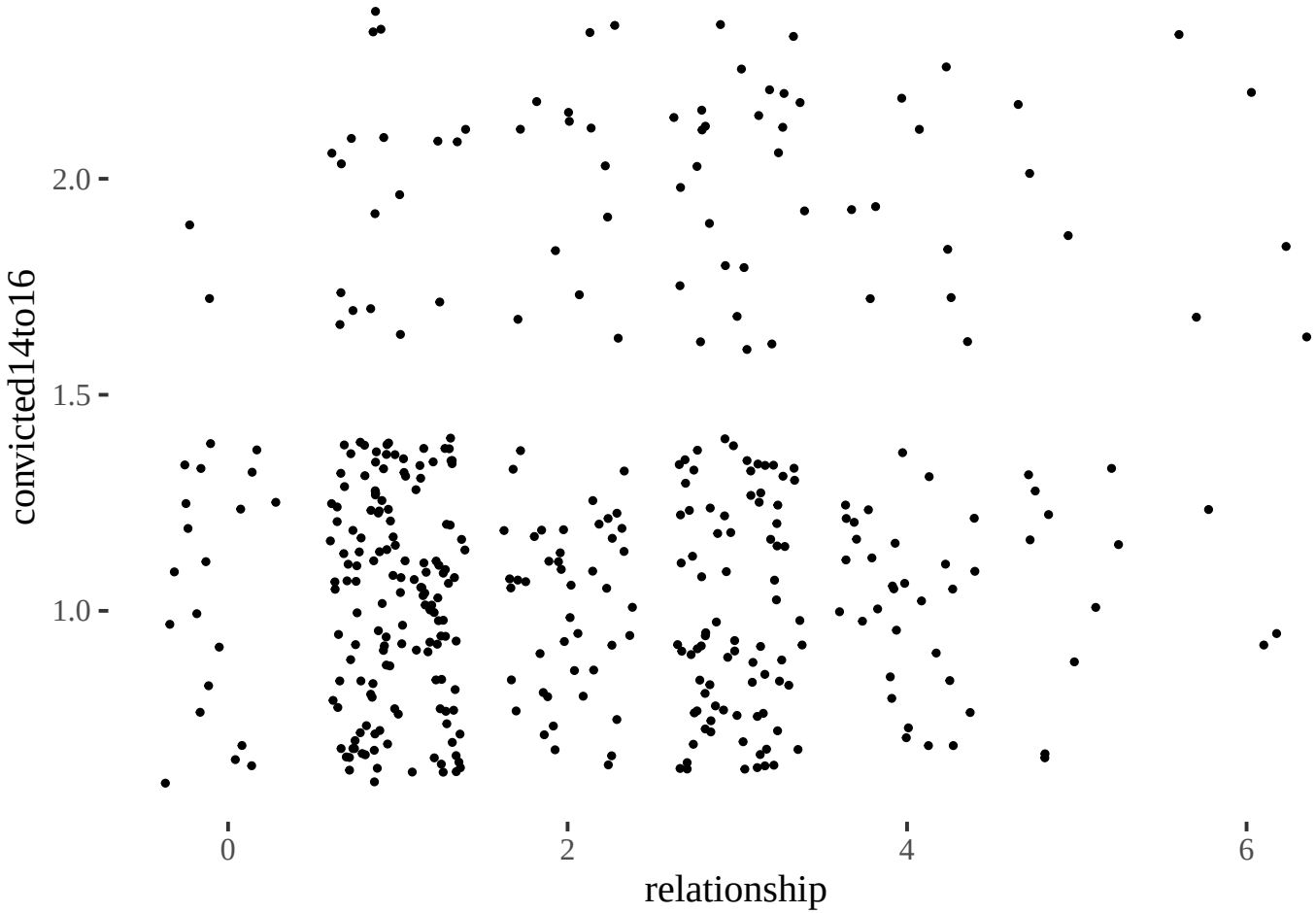
$$y_i \sim \mathit{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

Binary outcomes

```
# 677 no, casual, steady, engaged,
#married, cohabiting 6:  convicted 14-16
data <- as.data.frame(read_xpt("crimeLife.xpt"))
small <- data[,c(6, 677)]

names(small) <- c("convicted14to16", "relationship")

cors <- cor(small, method = "spearman")
ggcorrplot(cors, lab= TRUE, lab_size = 5, tl.srt = 0) + corSize
```

Binary outcomes

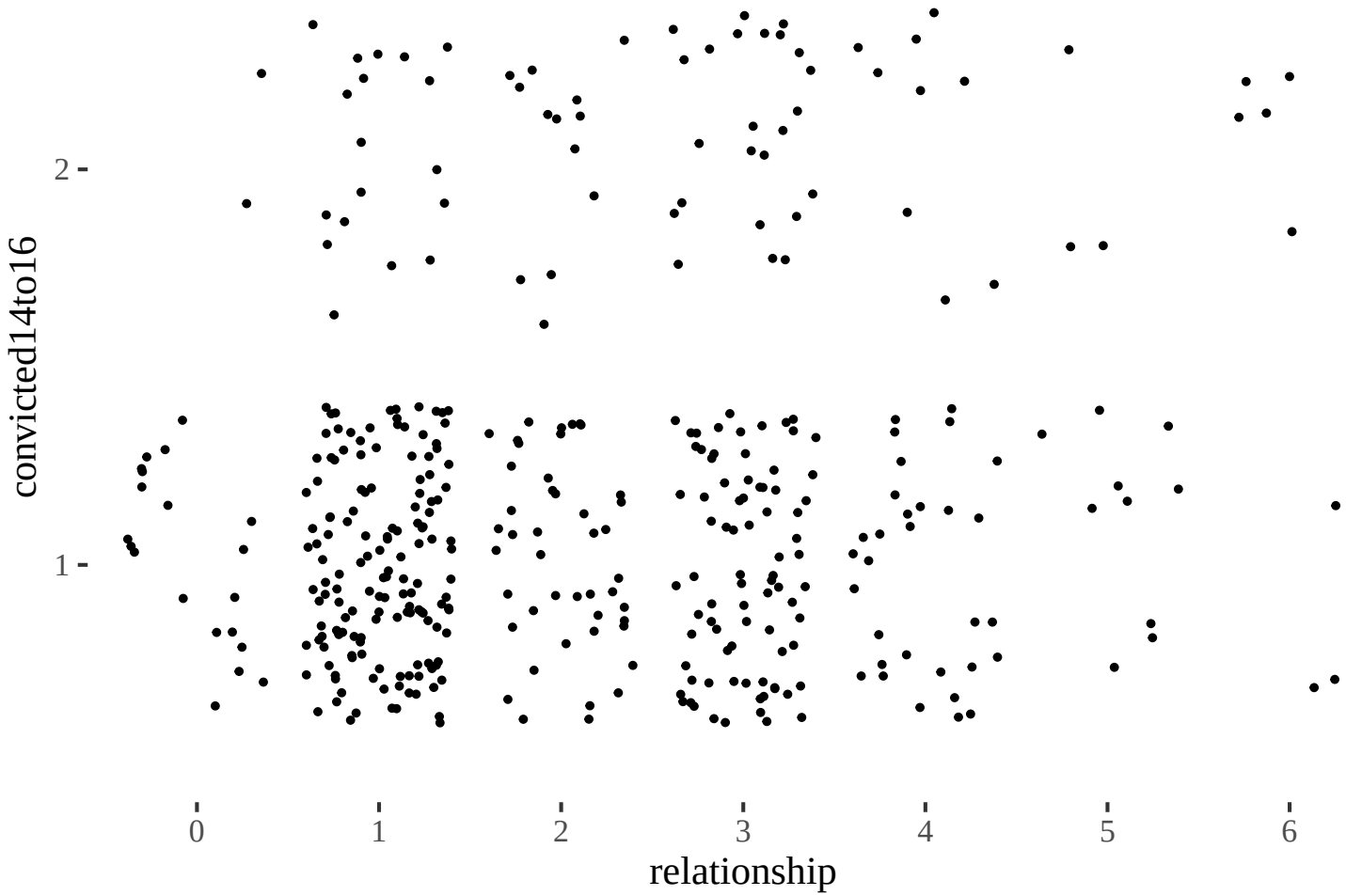
```
ggplot(small, aes(x = relationship, y = convicted14to16))+
  geom_jitter() + th
```




```
small$relationship <- factor(small$relationship)
small$convicted14to16 <- factor(small$convicted14to16, level = c(1,2))
```

Binary outcomes

```
ggplot(small, aes(x = relationship, y = convicted14to16))+
  geom_jitter() + th
```



Binary outcomes

```
levels(small$relationship) <-
  c(NA, "no", "casual", "steady", "engaged",
    "married", "cohabiting")

nrow(small)
```

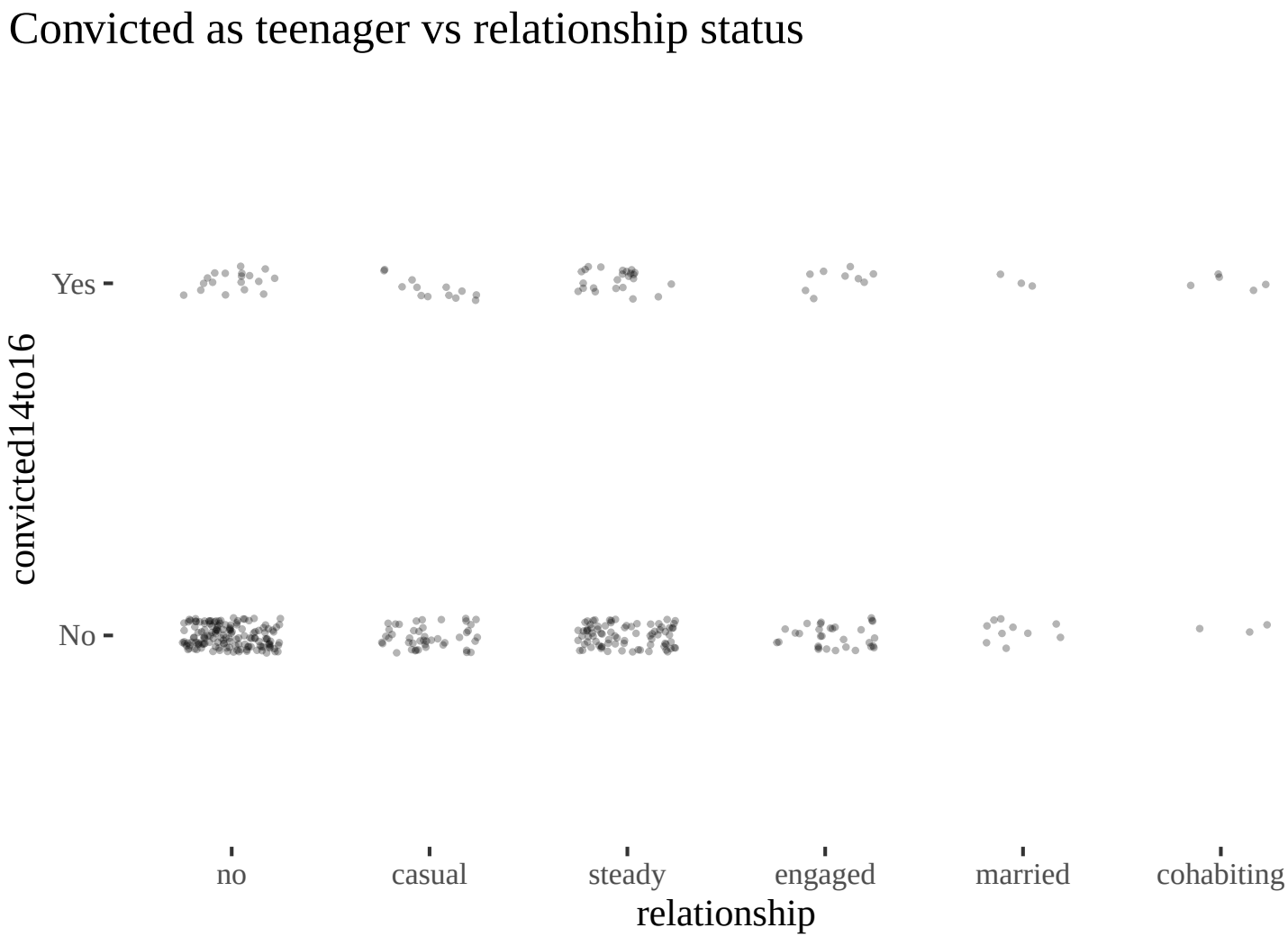
```
## [1] 411
```

```
small <- small[complete.cases(small),]
nrow(small)
```

```
## [1] 389
```

Binary outcomes

```
ggplot(small, aes(x = relationship, y = convicted14to16))+
  geom_jitter(height = .05, width = .25, size = 1.2, alpha = .3)+
  ggtitle("Convicted as teenager vs relationship status")+
  scale_y_discrete(labels = c("No","Yes")) + th
```



Why we need link functions

This makes no sense

An oversimplification?

Throw cohabiting below engaged, treat as numeric. Never do at home!

Prep your data

```
rel <- small$relationship
levels(rel) <- c(1,2,3,5,6,4)

data <- list(
        rel = as.numeric(as.character(rel)),
        conv = as.numeric(small$convicted14to16)-1,
        relFactor = as.numeric(small$relationship)
)
```

What are link functions anyway?

In general

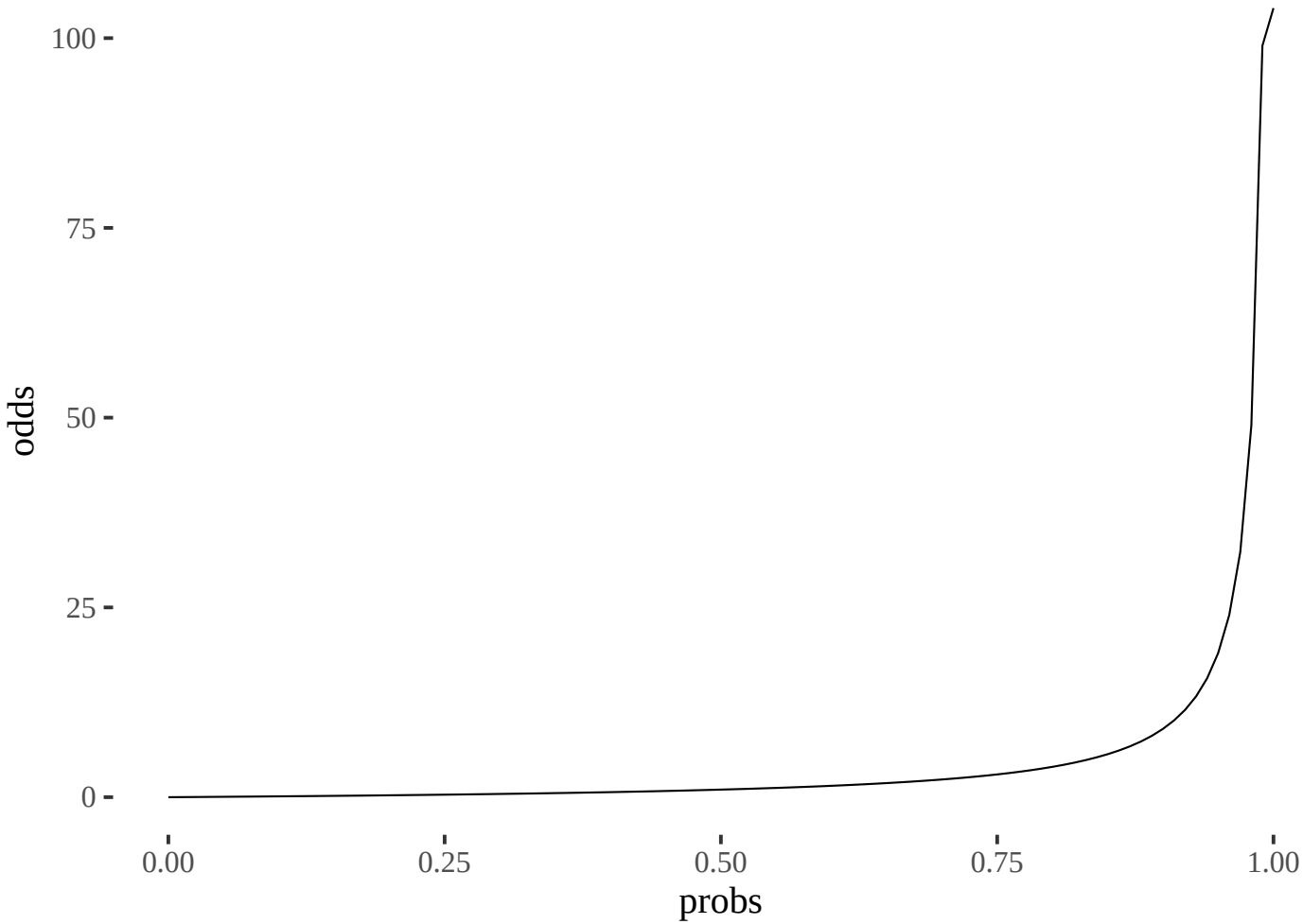
$$y_i \sim \text{Blah}(\theta_i, \phi)$$
$$f(\theta) = \alpha + \beta(x_i - \bar{x})$$

Logit link

$$y_i \sim \text{Binomial}(n, p_i)$$
$$\text{logit}(p_i) = \alpha + \beta(x_i - \bar{x})$$
$$\text{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right)$$
$$\log\left(\frac{p_i}{1 - p_i}\right) = \alpha + \beta(x_i - \bar{x})$$
$$p_i = \frac{\exp(\alpha + \beta x_i)}{1 + \exp(\alpha + \beta x_i)}$$

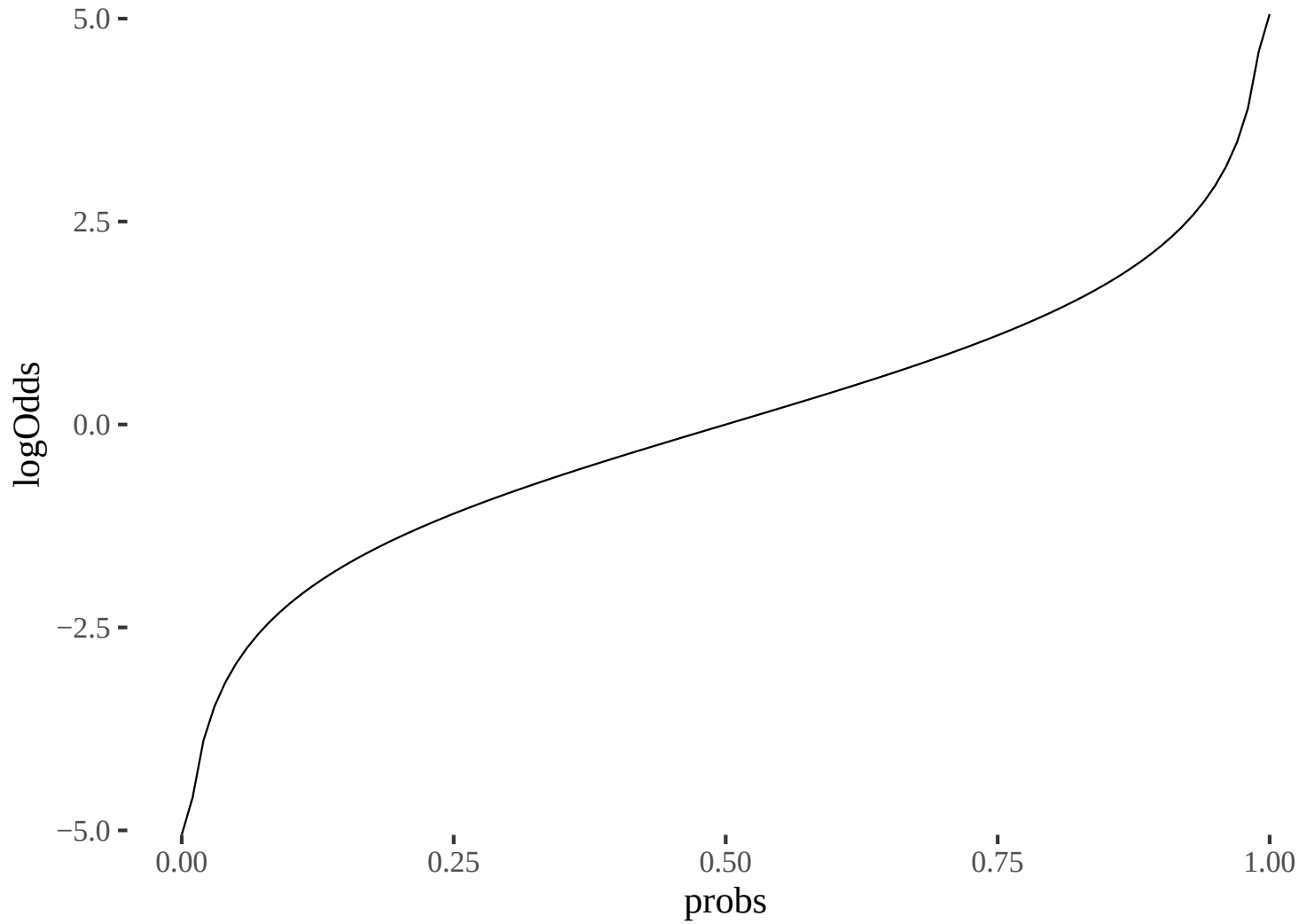
Logit link

```
probs <- seq(0,1,.01)
odds <- probs/ (1-probs)
logOdds <- log(odds)

ggplot()+geom_line(aes(x = probs, y = odds))+th
```



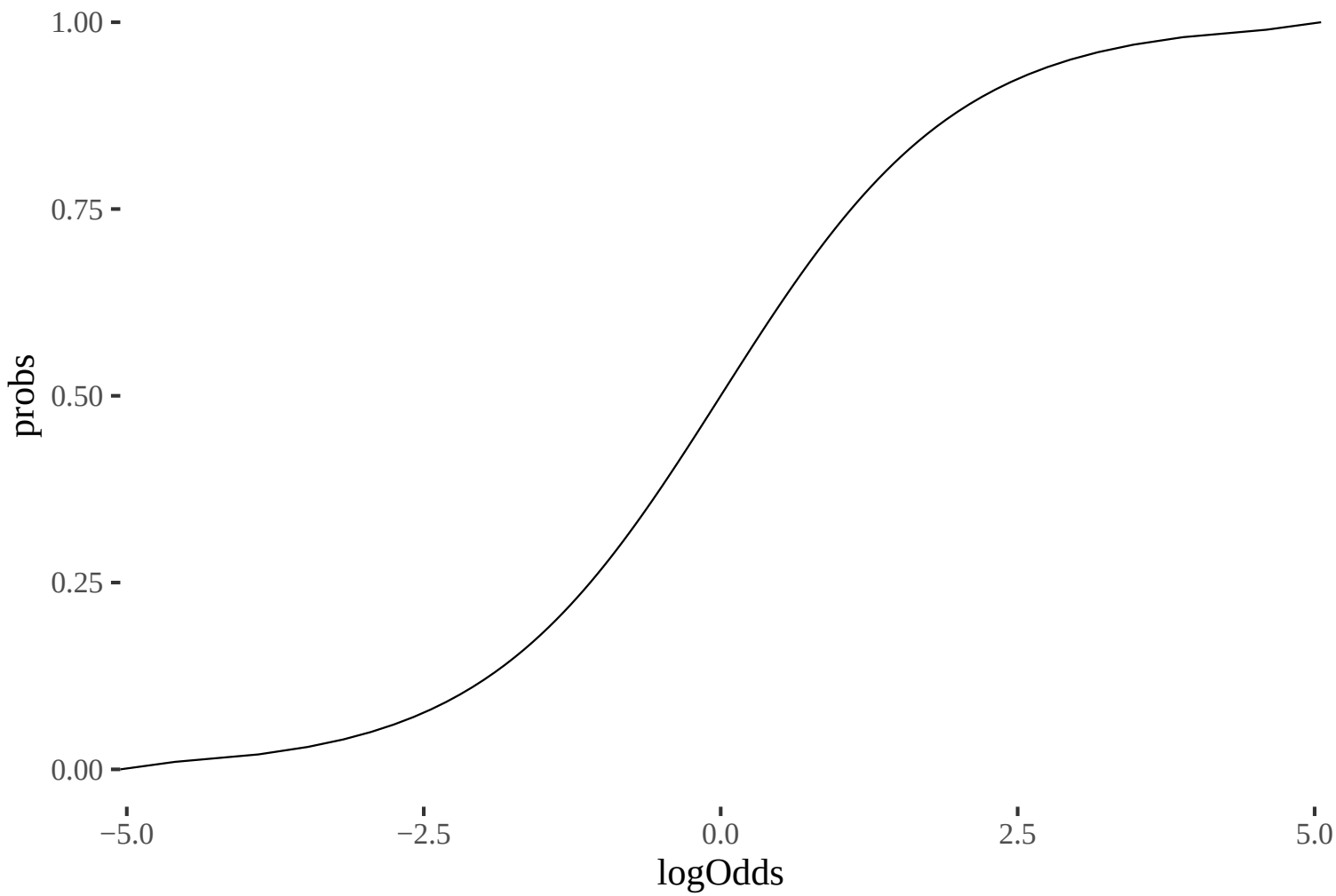
Logit link

```
ggplot()+geom_line(aes(x = probs, y = logOdds))+th
```



Logit link

```
ggplot()+geom_line(aes(y = probs, x = logOdds))+th
```

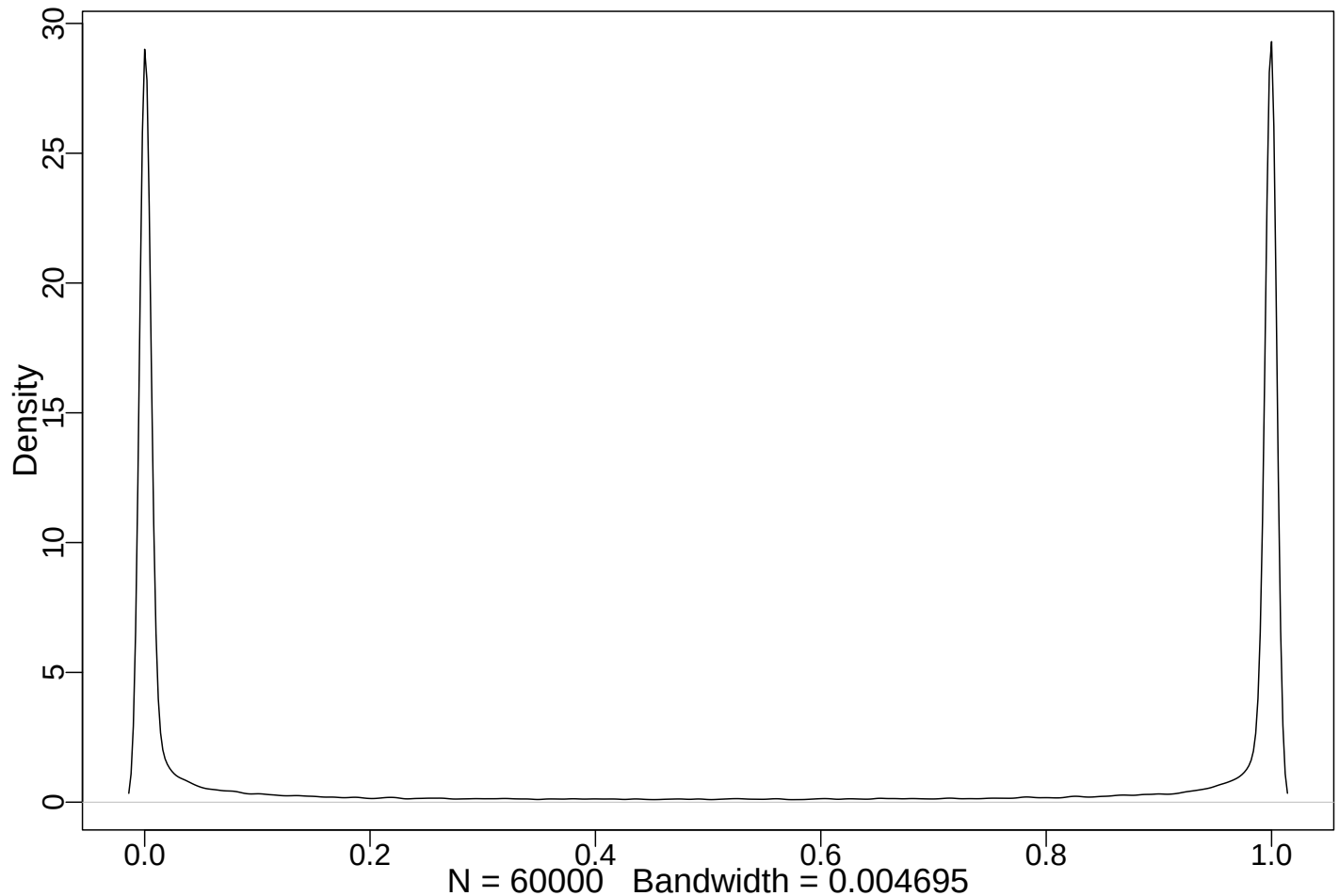


Let's build this!

```
crimeFactorial <- ulam(
  alist(
    conv ~ dbinom( 1 , p ) ,
    logit(p) <- a + b[relFactor] ,
    a ~ dnorm( 0, 10),
    b[relFactor] ~ dnorm( 0 , 10 )
  ) , data=data, log_lik = TRUE )
```

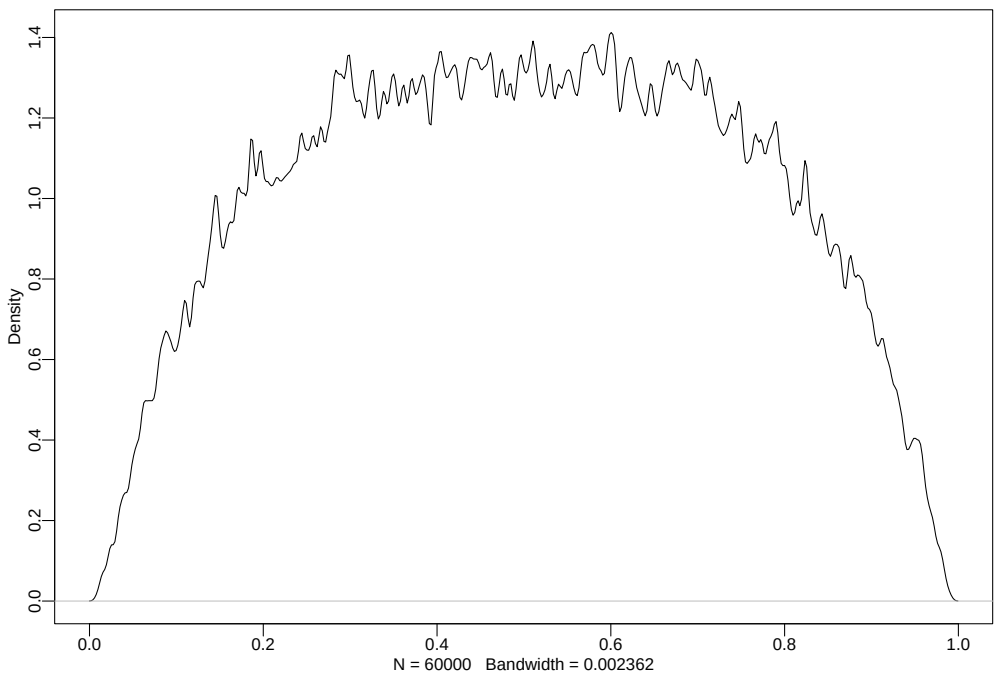
Check your priors!

```
prior <- extract.prior( crimeFactorial , n=1e4 )

p <- sapply( 1:6 , function(k) inv_logit( prior$a + prior$b[,k] ) )

dens( p , adj=0.1, cex.axis=1.3, cex.lab=1.5 )
```



Check your priors!

```
crimeFactorialNarrow <- ulam(
  alist(
    conv ~ dbinom( 1 , p ) ,
    logit(p) <- a +  b[relFactor] ,
    a ~ dnorm( 0, 1.1),
    b[relFactor] ~ dnorm( 0 , .5 )
  ) , data=data, log_lik = TRUE )

priorN <- extract.prior( crimeFactorialNarrow , n=1e4 )

pN <- sapply( 1:6 , function(k) inv_logit( priorN$a + priorN$b[,k] ) )

dens( pN, adj=0.1 )
```



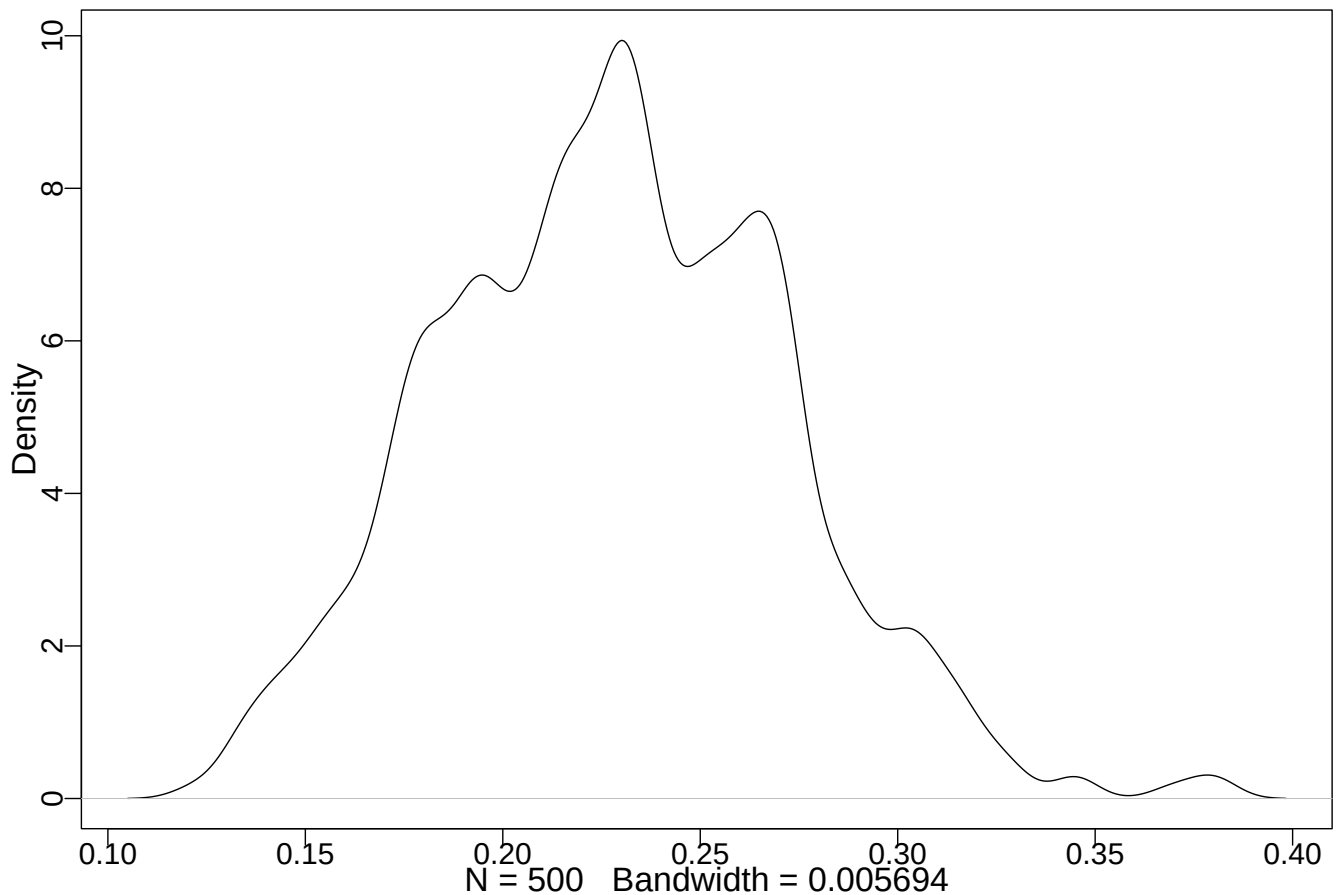
Now the posteriors

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

```
##               mean         sd       5.5%      94.5%    n_eff     Rhat4
## a    -1.234630978 0.2531478 -1.6541653 -0.8405635 257.3871 1.0017452
## b[1] -0.717296575 0.2990603 -1.1587627 -0.2187249 291.7744 0.9990760
## b[2] -0.026760902 0.3376146 -0.5708770  0.4971146 452.8202 0.9986847
## b[3] -0.009521166 0.3074331 -0.4814737  0.4810955 408.4980 1.0032418
## b[4] -0.024761448 0.3360923 -0.5416487  0.4988446 434.0371 0.9984822
## b[5] -0.029644701 0.3819540 -0.6490443  0.5898717 503.0951 0.9998532
## b[6]  0.534888415 0.4480727 -0.2041754  1.2468165 296.6330 0.9986290
```

Now the posteriors

```
post <- extract.samples(crimeFactorialNarrow)

baseline <- inv_logit(post$a)

dens (baseline, cex.axis=1.3, cex.lab=1.5)
```



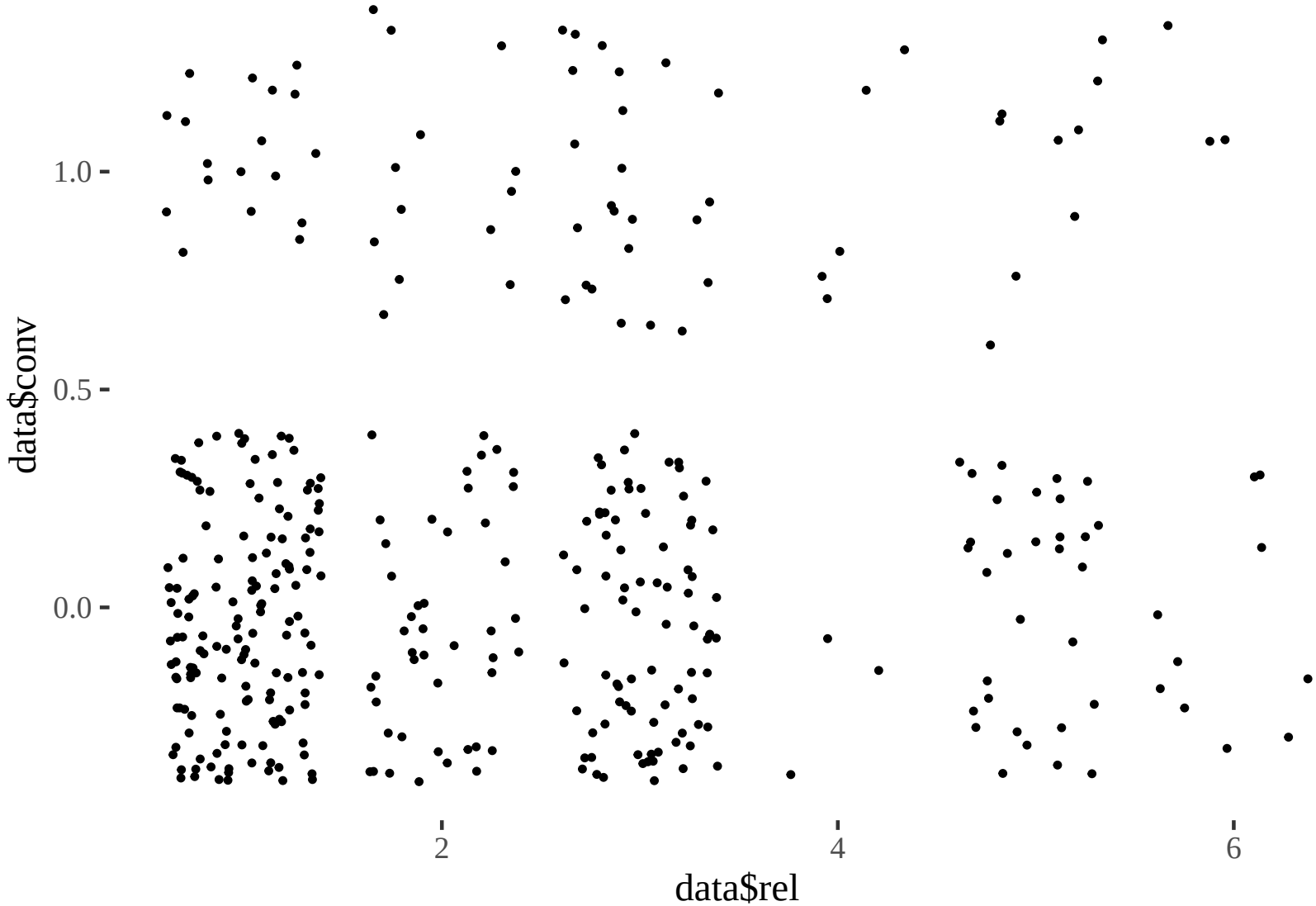
Now the posteriors

```
postDF <- sapply( 1:6 , function(k) inv_logit(
  post$a + post$b[,k]))

postDFLong <- melt(postDF)
names(postDFLong) <- c("id", "girlfriend", "probability")

precDF <- precis( crimeFactorialNarrow , depth=2 )
means <- inv_logit(precDF$mean[1] + precDF$mean[2:7])
means
```

```
## [1] 0.1243433 0.2207344 0.2237141 0.2210785 0.2202387 0.3318693
```

Now the posteriors

Continuous predictors



Continuous predictors

```
crimeContinuous <- ulam(
  alist(
    conv ~ dbinom( 1 , p ) ,
    logit(p) <- a +  b * rel ,
    a ~ dnorm( 0, 1.1),
    b ~ dnorm( 0 , .5 )
  ) , data=data, log_lik = TRUE )
```

```
precis(crimeContinuous)
```

```
##          mean          sd        5.5%      94.5%    n_eff     Rhat4
## a -1.9181769 0.25878472 -2.32716535 -1.5070056 117.3896 1.004339
## b  0.1885123 0.08830667  0.04364142  0.3272134 135.7982 1.000016
```

```
inv_logit(-1.96)
```

```
## [1] 0.123467
```

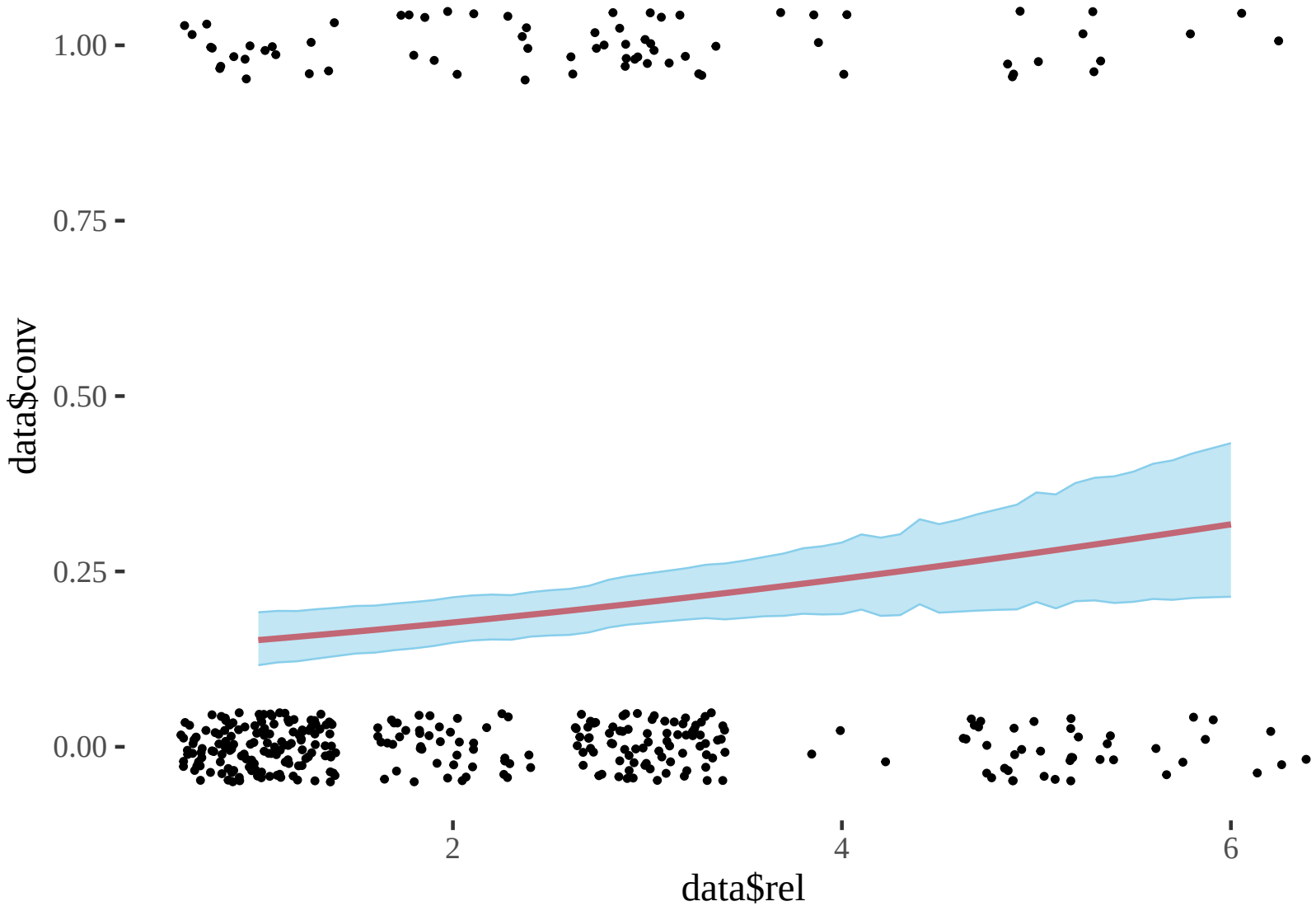
```
exp(precis(crimeContinuous)$mean)
```

```
## [1] 0.1468745 1.2074520
```

Continuous predictors

```
fake <- seq(1,6, by = .1)
estimates <- link(crimeContinuous, data.frame(rel = fake))
meanEstimates <- apply(estimates, 2, mean)
hpdiEstimates <- data.frame(t(apply(
  estimates, 2, HPDI, prob = .89)))
names(hpdiEstimates) <- c("meanLow", "meanHigh")
predsDF <- cbind(meanEstimates, hpdiEstimates)
```

Continuous predictors



Continuous predictors

```
compare(crimeContinuous, crimeFactorialNarrow, crimeFactorial)
```

```
##                           WAIC       SE     dWAIC      dSE    pWAIC    weight
## crimeFactorialNarrow 367.4572 22.80859 0.0000000       NA 4.347992 0.4999365
## crimeFactorial       368.2796 23.88357 0.8223495 4.405066 6.325817 0.3313935
## crimeContinuous      369.6303 22.23951 2.1730738 3.657024 1.958021 0.1686700
```

UC Berkeley admission

```
data(UCBadmit)
d <- UCBadmit
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
## 7     D             male   138    279          417
## 8     D           female   131    244          375
## 9     E             male    53    138          191
## 10    E           female    94    299          393
## 11    F             male    22    351          373
## 12    F           female    24    317          341
```

UC Berkeley admission

```
dat_list <- list(
  admit = d$admit,
  applications = d$applications,
  gid = ifelse( d$applicant.gender=="male" , 1 , 2 )
)
```

UC Berkeley admission

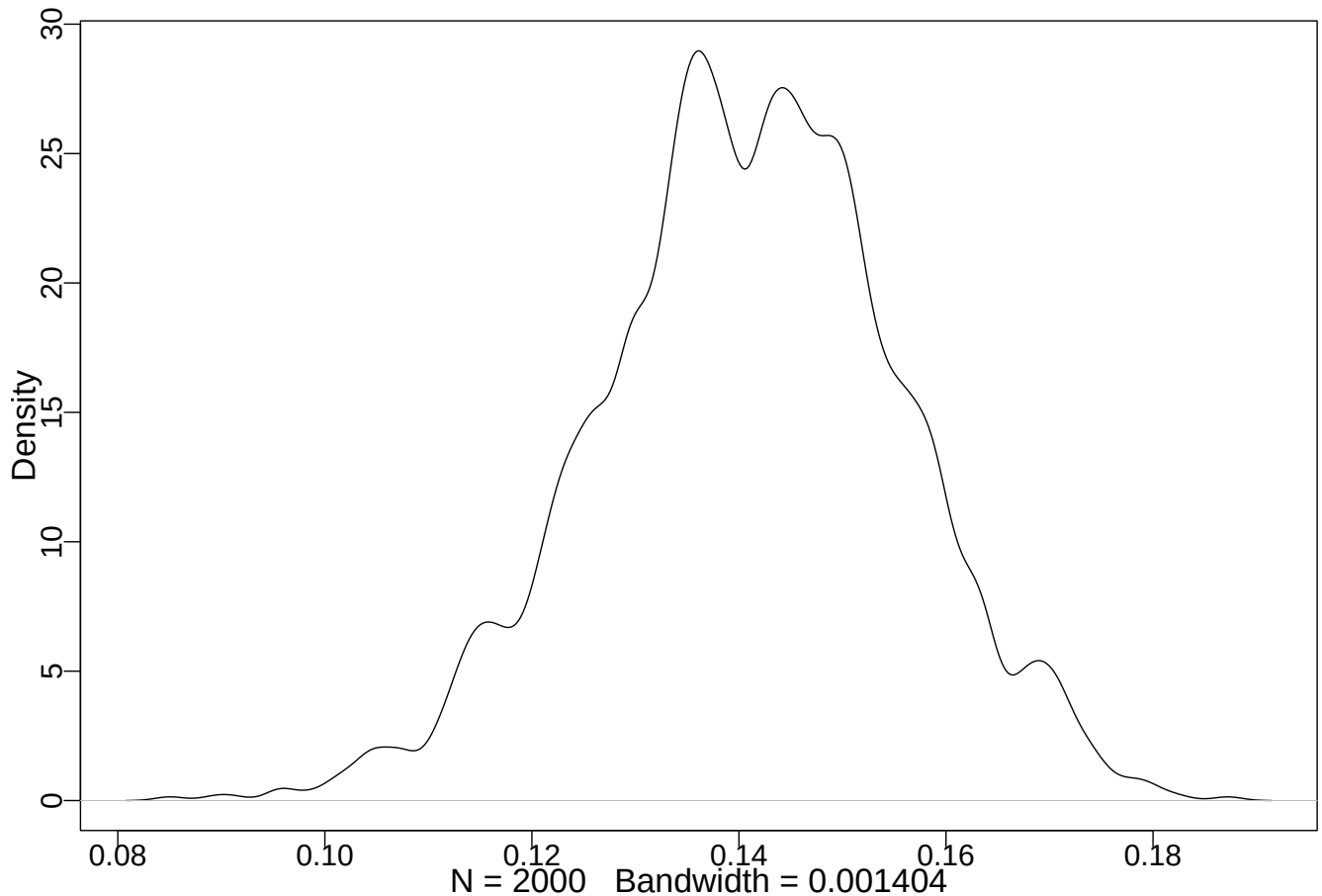
```
ucbModelSimple <- ulam(
  alist(
admit ~ dbinom( applications , p ) ,
logit(p) <- a[gid] ,
a[gid] ~ dnorm( 0 , 1.5 )
) , data=dat_list , chains=4 )
```

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

```
##             mean         sd       5.5%      94.5%    n_eff     Rhat4
## a[1] -0.2213142 0.04128791 -0.2888276 -0.1556763 1447.886 1.0018329
## a[2] -0.8287490 0.04896605 -0.9079320 -0.7516819 1318.184 0.9993704
```

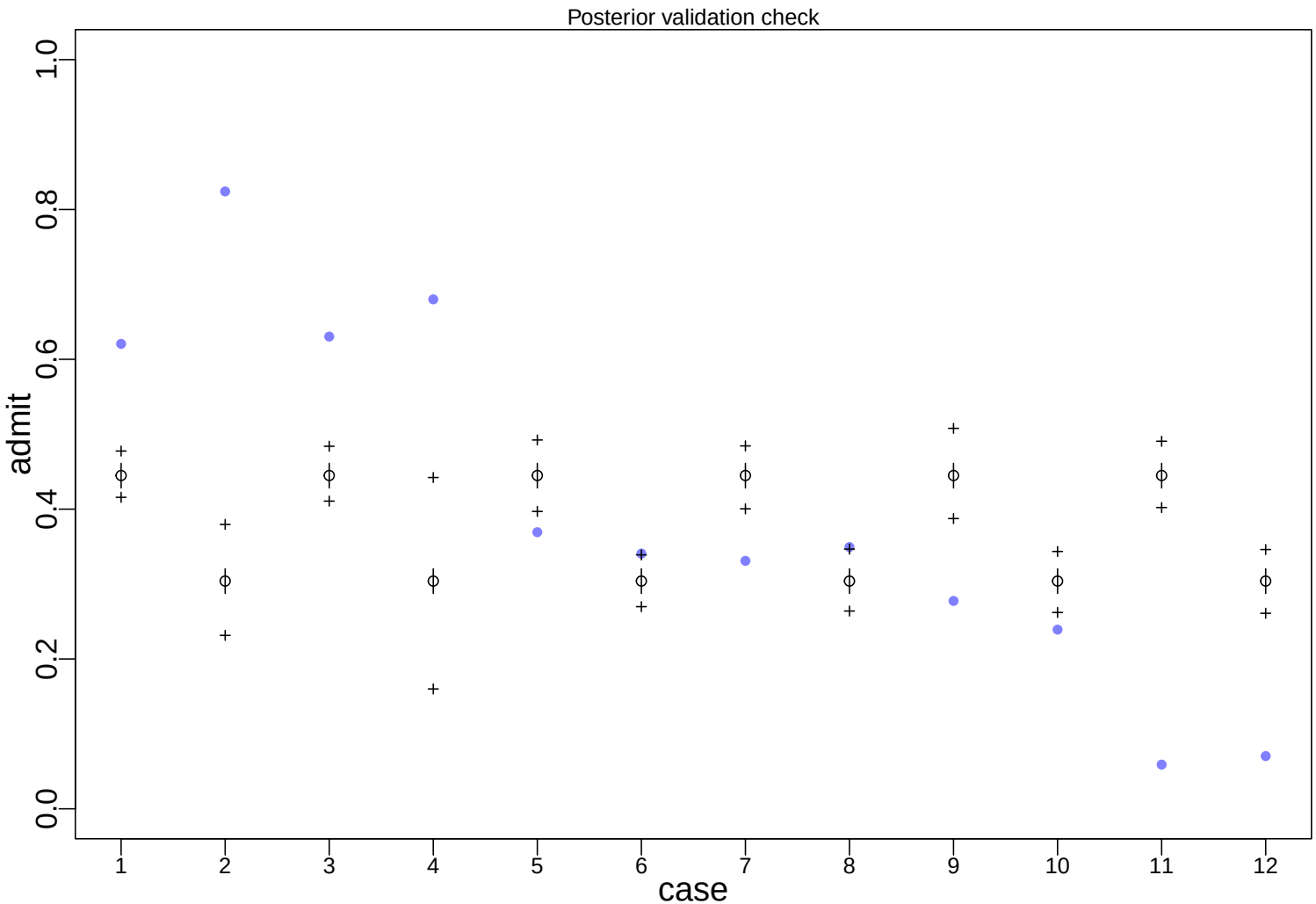
UC Berkeley admissions

```
post <- extract.samples(ucbModelSimple)
diff_p <- inv_logit(post$a[,1]) - inv_logit(post$a[,2])
dens(diff_p, cex.axis=1.3, cex.lab=1.5)
```

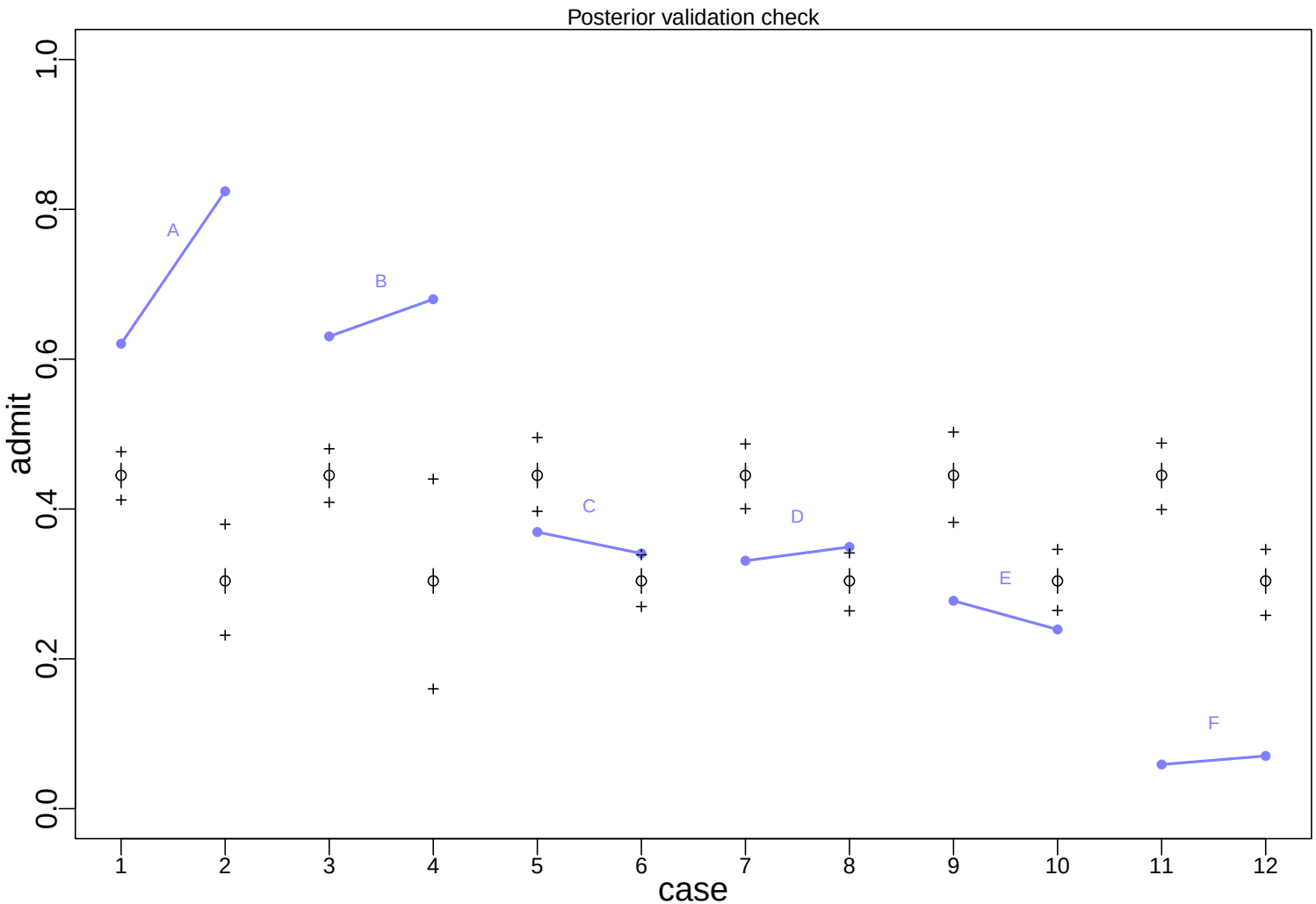


UC Berkeley admissions

```
postcheck( ucbModelSimple, cex.axis=1.3, cex.lab=1.5)
```



UC Berkeley admissions



Within departments

```
dat_list$dept_id <- rep(1:6,each=2)

ucbModelWithin <- ulam(
  alist(
    admit ~ dbinom( applications , p ) ,
    logit(p) <- a[gid] + delta[dept_id] ,
    a[gid] ~ dnorm( 0 , 1.5 ) ,
    delta[dept_id] ~ dnorm( 0 , 1.5 )
  ) , data=dat_list , chains=4 , iter=4000 )
```

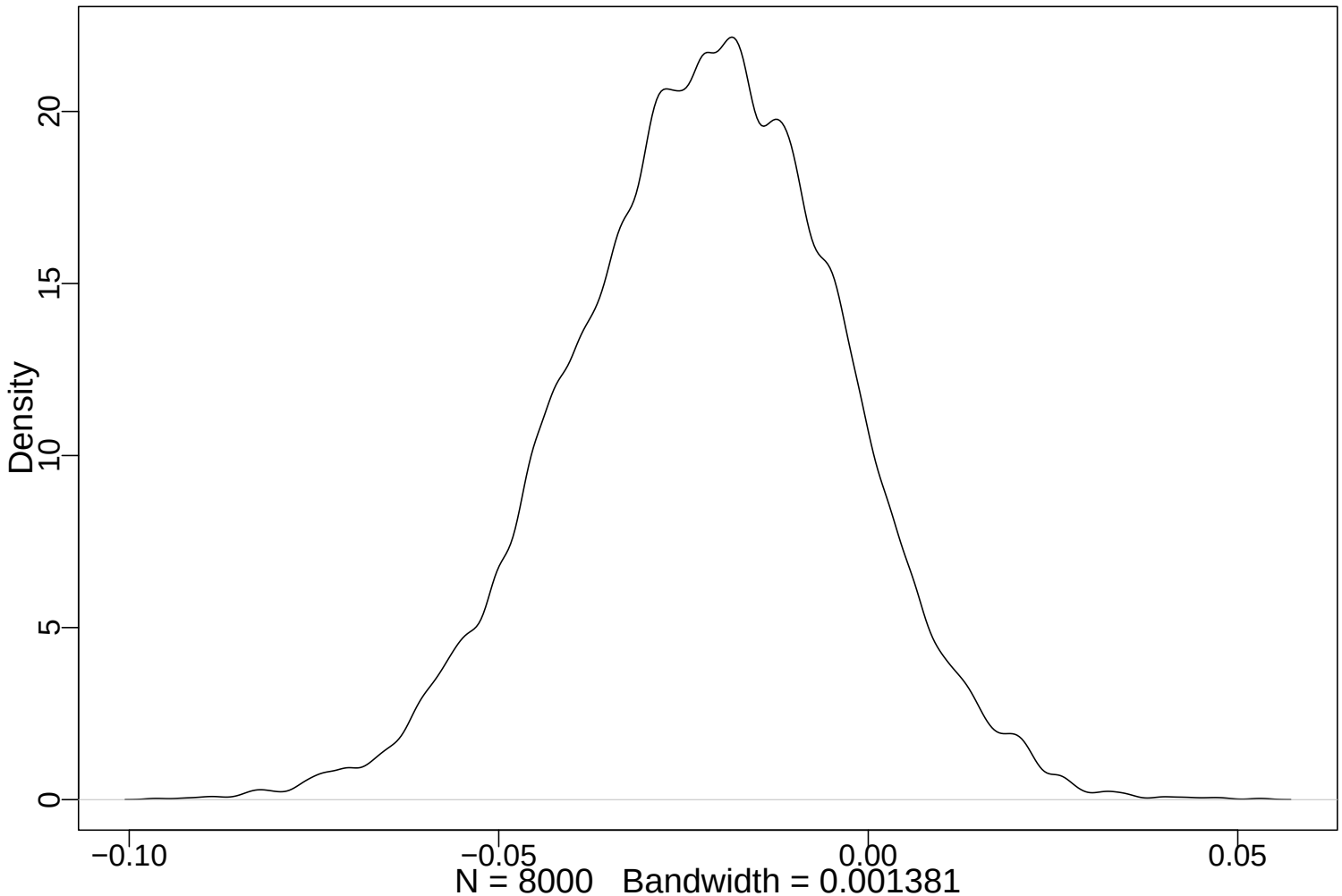
Within departments

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

```
##                 mean         sd       5.5%      94.5%    n_eff     Rhat4
## a[1]      -0.5224407 0.5239205 -1.3403938  0.3187166 585.2709 1.004598
## a[2]      -0.4234901 0.5273469 -1.2518935  0.4375151 585.1799 1.004341
## delta[1]   1.1016874 0.5274853  0.2440103  1.9264046 590.2360 1.004519
## delta[2]   1.0582925 0.5301608  0.2037090  1.8962324 598.7096 1.004363
## delta[3]  -0.1580576 0.5281081 -1.0274138  0.6735273 595.9356 1.004377
## delta[4]  -0.1896556 0.5287345 -1.0496564  0.6469719 591.5600 1.004547
## delta[5]  -0.6336240 0.5312767 -1.5067861  0.2051101 593.2036 1.004229
## delta[6]  -2.1929551 0.5402724 -3.0750511 -1.3558785 631.5289 1.003866
```

Within departments

```
post <- extract.samples(ucbModelWithin)
diff_p <- inv_logit(post$a[,1]) - inv_logit(post$a[,2])
dens(diff_p,  cex.axis=1.3, cex.lab=1.5)
```

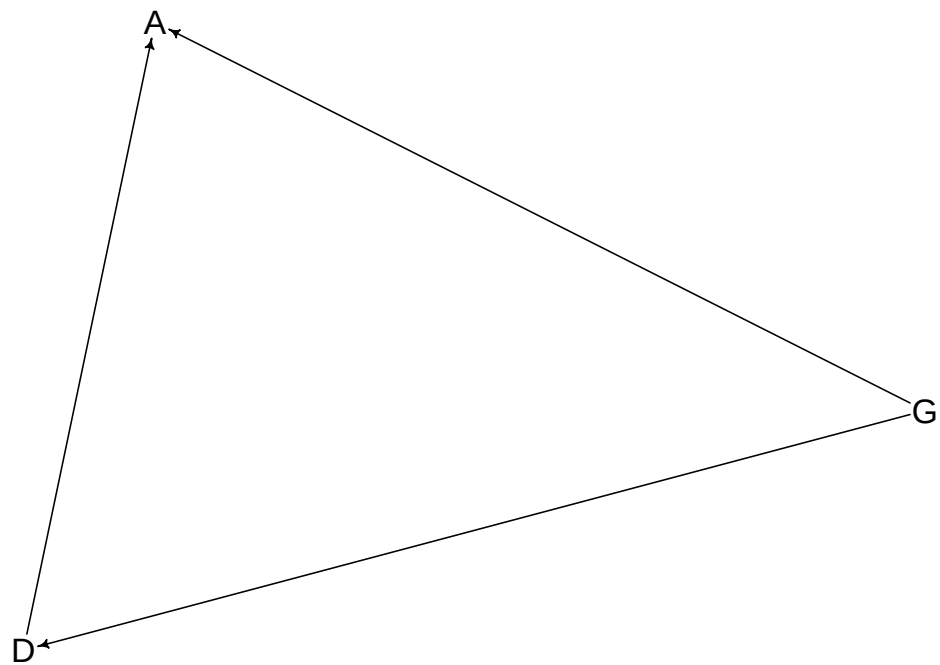


Within departments

```
##                      A    B    C    D    E    F
## male              0.88 0.96 0.35 0.53 0.33 0.52
## female            0.12 0.04 0.65 0.47 0.67 0.48
## multiplicative 0.75 0.74 0.46 0.45 0.35 0.10
```

Within departments

```
ucbDAG <- dagitty(
  "dag{
  G -> D; G -> A; D -> A
  }"
)
drawdag(ucbDAG, goodarrow = TRUE, cex = 2, radius = 3)
```

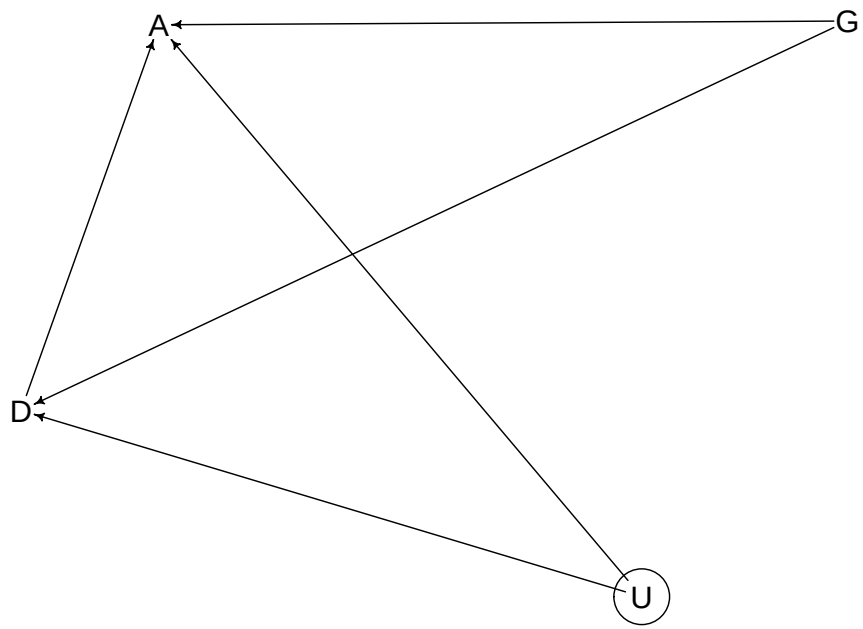



```
adjustmentSets(ucbDAG, exposure = "G",
               outcome = "A", effect = "direct")
```

```
## { D }
```

Within departments

```
ucbDAG2 <- dagitty(
  "dag{
  U [unobserved]
  G -> D; G -> A; D -> A
  A <- U -> D
  }"
)
drawdag(ucbDAG2, goodarrow = TRUE, cex = 2, radius = 8)
```




```
adjustmentSets(ucbDAG2, exposure = "G",
               outcome = "A", effect = "direct")
```

NONE!