Logistic Regression

Nikodem Lewandowski

University of Gdańsk

Likelihoods so far

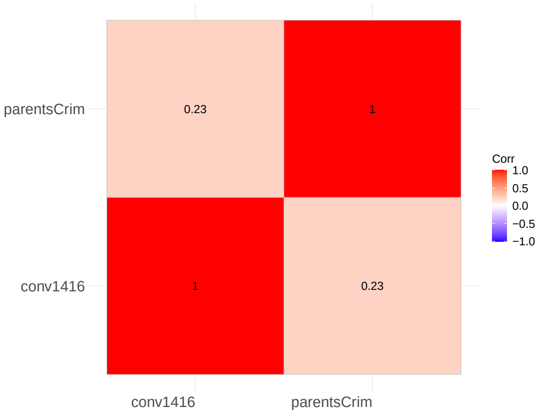
$$y_i \sim Normal(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

- Until now, we have been using the Normal distribution to describe the relationship between two or more variables. However, it is important to note that different distributions are possible.
  - In our first lecture, we utilized the binomial distribution, but solely for one variable.
- Normal distributions have their limitations, as the following example will illustrate. It is, at the very least, inappropriate to use them in modeling a binary outcome.
  - Modeling categorical outcomes or discrete digits can also be cumbersome. (And these challenges are not exhaustive either!)

University of Gdańsk

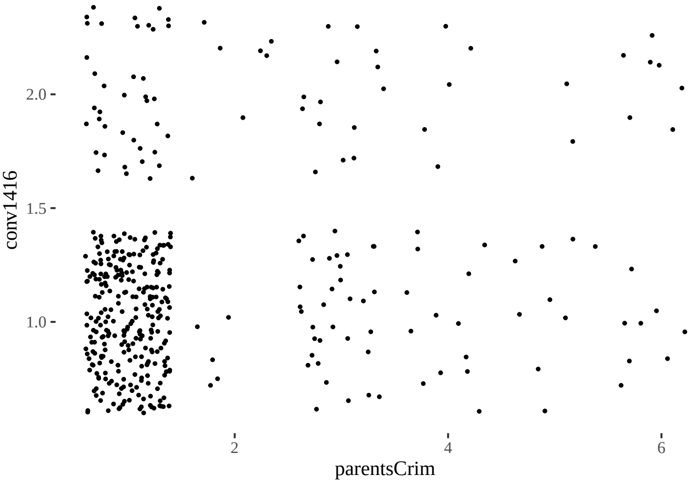
Binary outcomes

```
data <- as.data.frame(read_xpt("crimeLife.xpt"))
small <- data[,c(6, 300)]

names(small) <- c("conv1416", "parentsCrim")

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



Binary outcomes

```
ggplot(small, aes(x = parentsCrim, y = conv1416))+
  geom_jitter() + th
```



Binary outcomes

```r
small$parentsCrim <- as.factor(small$parentsCrim)
levels(small$parentsCrim) <- c('no', "as_juv", "as_adult1", "as_adult2",
                               "as_adult3", "as_adult4")


small$conv1416 <- as.factor(small$conv1416)

nrow(small)
```

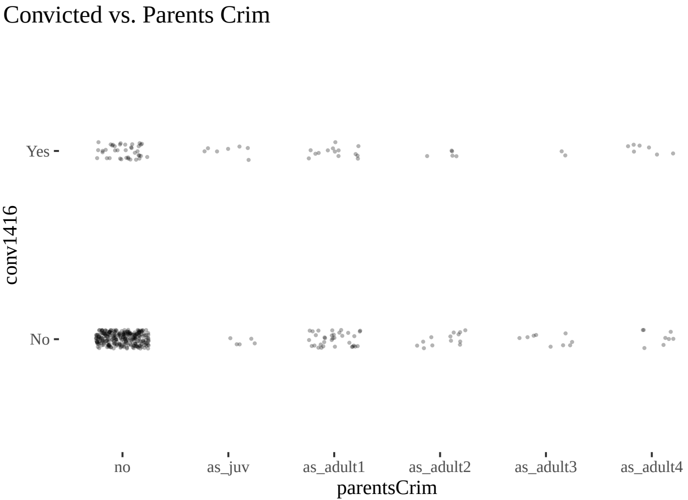
```
[1] 411
```

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

```
[1] 411
```

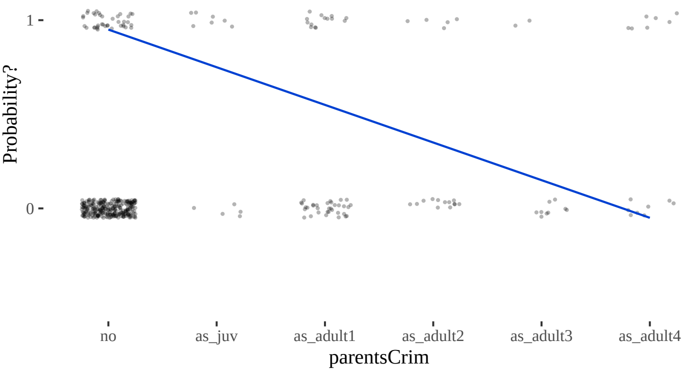
Binary outcomes

```
ggplot(small, aes(x = parentsCrim, y = conv1416))+
  geom_jitter(height = .05, width = .25, size = 1.2, alpha = .3)+
  ggtitle("Convicted vs. Parents Crim")+
  scale_y_discrete(labels = c("No","Yes")) + th
```



Trying to fit a line

This makes no sense



GLM

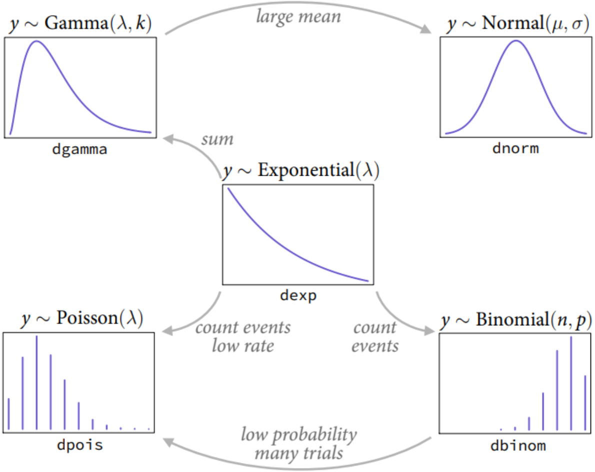
- General Linear Models - as the name suggests we will generalize the linear model, many different functions are possible:

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

- $Blah$ is an empty space for a distribution
- f is a link function
- θ_i and ϕ depends on the distribution of choice

University of Gdańsk

Some options


$y \sim \text{Gamma}(\lambda, k)$
dgamma
large mean
$y \sim \text{Normal}(\mu, \sigma)$
dnorm
sum
$y \sim \text{Exponential}(\lambda)$
dexp
count events low rate
count events
$y \sim \text{Poisson}(\lambda)$
dpois
$y \sim \text{Binomial}(n, p)$
dbinom
low probability many trials

Our choice

$$\begin{aligned} y_i &\sim \mathsf{Binomial}(n, p_i) \\ \mathsf{logit}(p_i) &= \alpha + \beta(x_i - \bar{x}) \end{aligned}$$

And our link function will be the logit function which is defined as the log-odds:

$$\begin{aligned} \mathsf{logit}(p_i) &= \log\left(\frac{p_i}{1-p_i}\right) \\ \log\left(\frac{p_i}{1-p_i}\right) &= \alpha + \beta(x_i - \bar{x}) \\ p_i &= \frac{\exp(\alpha + \beta x_i)}{1 + \exp(\alpha + \beta x_i)} \end{aligned}$$

University of Gdańsk

Logit function in action

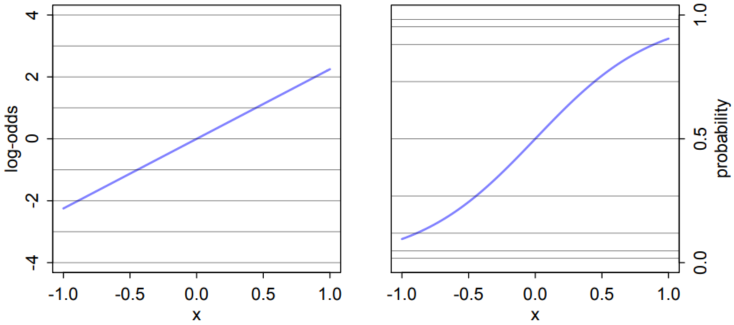


FIGURE 10.7. The logit link transforms a linear model (left) into a probability (right). This transformation compresses the geometry far from zero, such that a unit change on the linear scale (left) means less and less change on the probability scale (right).

Prep your data

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

data <- list(
        conv = as.numeric(small$conv1416 ) - 1,
        parentsCrim = as.numeric(small$parentsCrim)
)
```

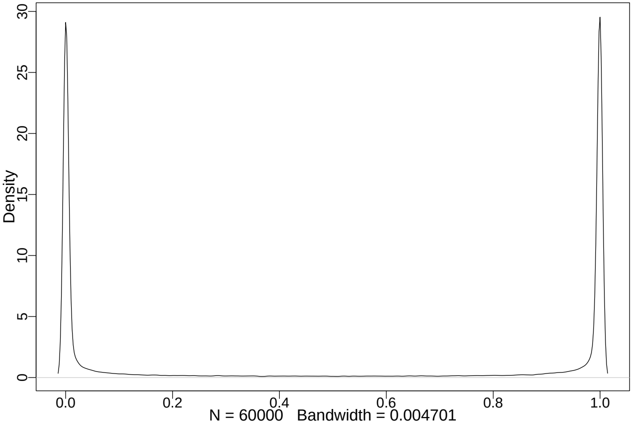
Let's build this!

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

- That's our initial guess about priors, we aim at creating flat priors, but it's not obvious as logit function is in effect

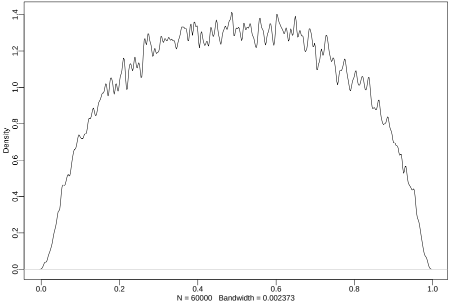
Check your priors!

```
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[parentsCrim] ,
    a ~ dnorm( 0, 1.1),
    b[parentsCrim] ~ 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 )
```



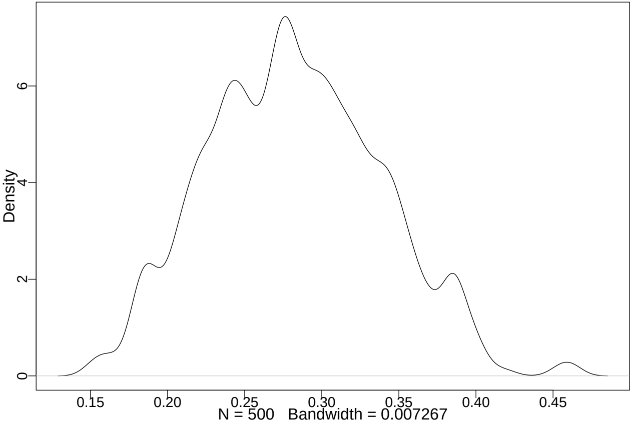
Now the posteriors

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

```
              mean        sd       5.5%      94.5%      rhat ess_bulk
a     -0.948519118 0.2808535 -1.4351043 -0.4953357 1.0011897 302.8537
b[1] -0.891117756 0.2975936 -1.3840576 -0.4141175 1.0021928 296.2278
b[2]  0.527066777 0.4203968 -0.1755131  1.1760870 1.0000182 551.2204
b[3]  0.039552922 0.3488355 -0.5190769  0.6309233 0.9993460 453.6668
b[4] -0.008142837 0.3740394 -0.6165708  0.6050240 0.9984504 373.6815
b[5] -0.181370511 0.4151439 -0.8481785  0.4952756 0.9980626 562.7357
b[6]  0.366456114 0.4092810 -0.2850920  1.0414571 1.0025092 471.0983
```

Now the posteriors

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

baseline <- inv_logit(post$a)

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

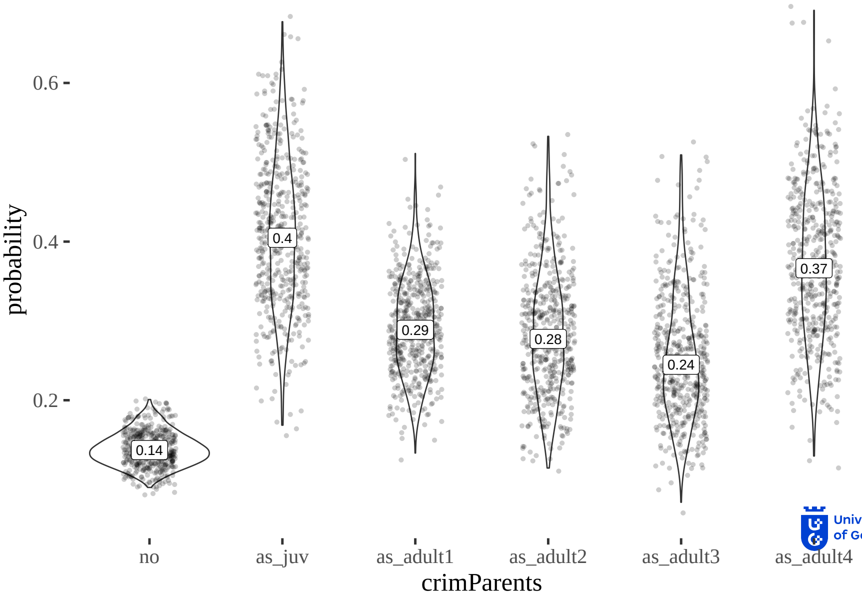


Now the posteriors

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

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

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

```
[1] 0.1372066 0.4048288 0.2887993 0.2773385 0.2447697 0.3664221
```

Probabilities for each category



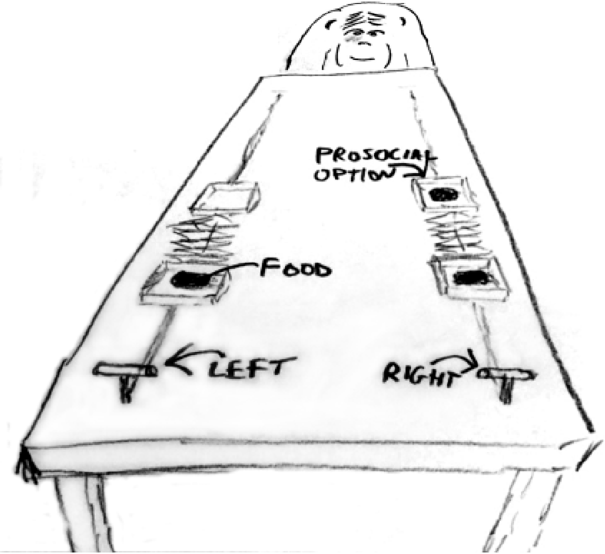
University of Gdańsk

Monkeys

```r
data(chimpanzees)
d <- chimpanzees
head(d)
```

```
  actor recipient condition block trial prosoc_left chose_prosoc pulled_left
1     1        NA         0     1     2           0            1           0
2     1        NA         0     1     4           0            0           1
3     1        NA         0     1     6           1            0           0
4     1        NA         0     1     8           0            1           0
5     1        NA         0     1    10           1            1           1
6     1        NA         0     1    12           1            1           1
```

Monkeys and Pro-Social Behaviour



Prepering the data

```r
d$treatment <- 1 + d$prosoc_left + 2*d$condition

dat_list <- list(
pulled_left = d$pulled_left,
actor = d$actor,
treatment = as.integer(d$treatment) )

str(dat_list)
```

```
List of 3
 $ pulled_left: int [1:504] 0 1 0 0 1 1 0 0 0 0 ...
 $ actor      : int [1:504] 1 1 1 1 1 1 1 1 1 1 ...
 $ treatment  : int [1:504] 1 1 2 1 2 2 2 2 1 1 ...
```

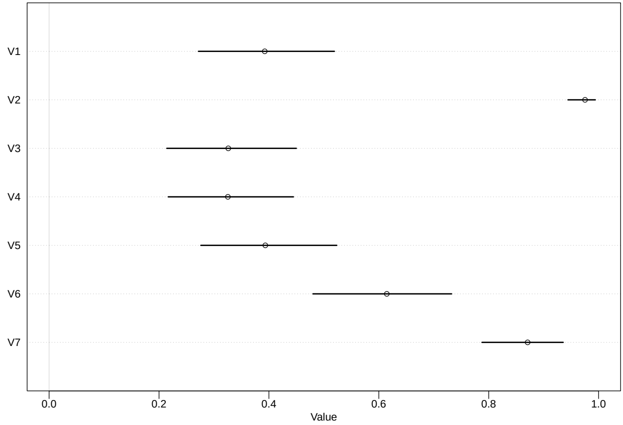
Monkey Model (adjusted priors)

```
chimpMod1 <- ulam(
alist(
pulled_left ~ dbinom( 1 , p ) ,
logit(p) <- a[actor] + b[treatment] ,
a[actor] ~ dnorm( 0 , 1.5 ),
b[treatment] ~ dnorm( 0 , 0.5 )
) , data=dat_list , chains=4 , log_lik=TRUE )

precis( chimpMod1 , depth=2 )
```

```
            mean        sd        5.5%       94.5%     rhat  ess_bulk
a[1] -0.46784275 0.3251865 -0.98861984  0.05837208 1.004885  729.1876
a[2]  3.87246332 0.7347622  2.76504855  5.16002495 1.002288 1059.9340
a[3] -0.76501624 0.3278244 -1.28253110 -0.22718083 1.003876  821.3490
a[4] -0.76008809 0.3344820 -1.30182445 -0.22927992 1.005010  757.6469
a[5] -0.46557635 0.3186123 -0.97497611  0.03418083 1.005617  744.2427
a[6]  0.46054039 0.3346066 -0.06090706  1.02945190 1.001850  790.2930
a[7]  1.94848435 0.4033888  1.34127350  2.62005035 1.008127  851.0381
b[1] -0.02307893 0.2817484 -0.48650471  0.40517833 1.005232  635.9403
b[2]  0.49790563 0.2750531  0.04341291  0.92356510 1.007474  574.2554
b[3] -0.36993908 0.2825284 -0.82629321  0.08221905 1.006678  640.0806
b[4]  0.38548651 0.2877124 -0.06509703  0.84322291 1.009836  586.3724
```

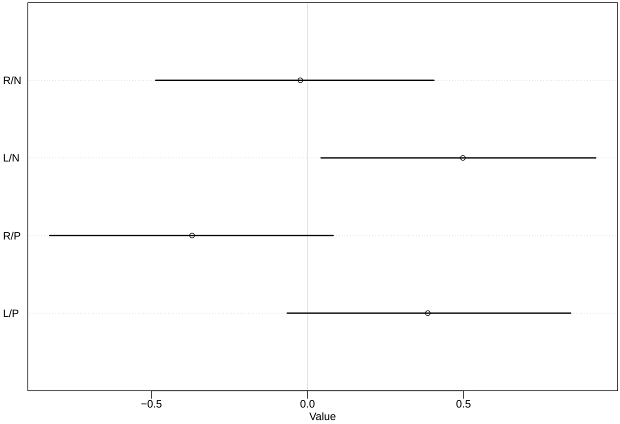
Sampling from the posteriors

```
post <- extract.samples(chimpMod1)
p_left <- inv_logit( post$a )
plot( precis( as.data.frame(p_left) ) , xlim=c(0,1) )
```



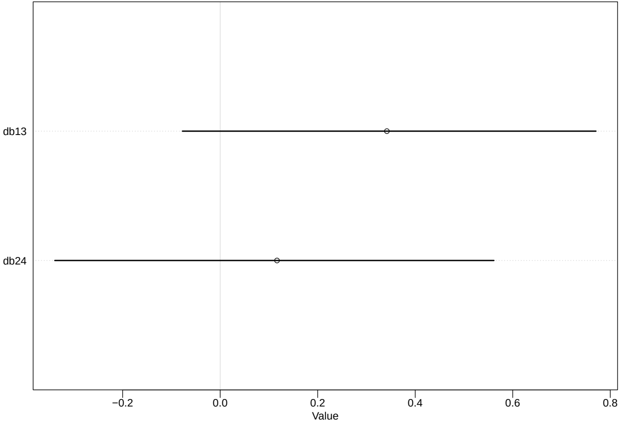
Overall Scenerios frequency

```
labs <- c("R/N","L/N","R/P","L/P")
plot( precis( chimpMod1 , depth=2 , pars="b" ) , labels=labs )
```

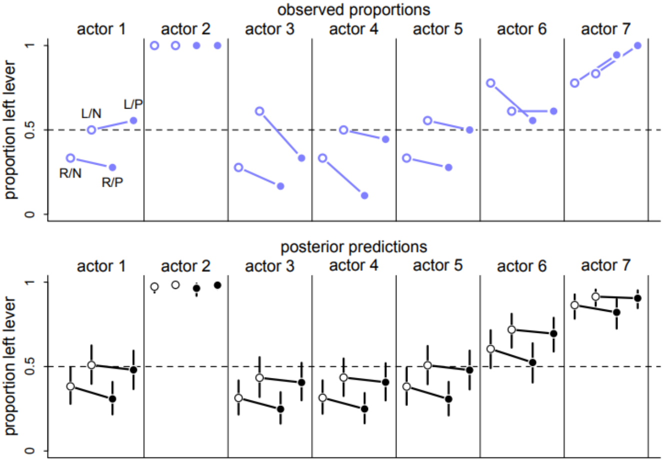


Contrasts

```
diffs <- list(
db13 = post$b[,1] - post$b[,3],
db24 = post$b[,2] - post$b[,4] )
plot( precis(diffs) )
```



Post. pred. check



University of California, Berkeley (UCB)

```
data(UCBadmit)
d <- UCBadmit

d # dataset about admissions and rejections
```

```
   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
```

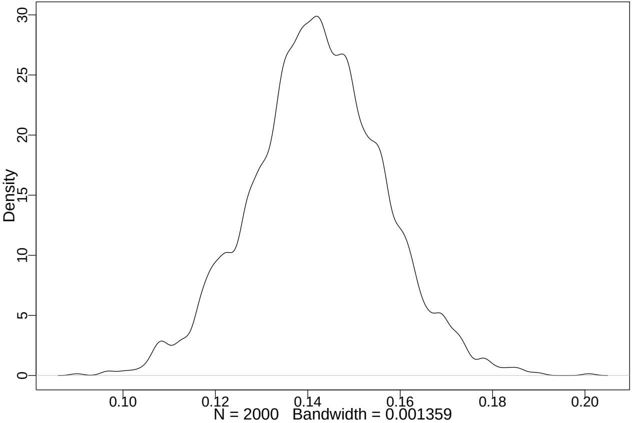
UCB Model

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

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 )
```

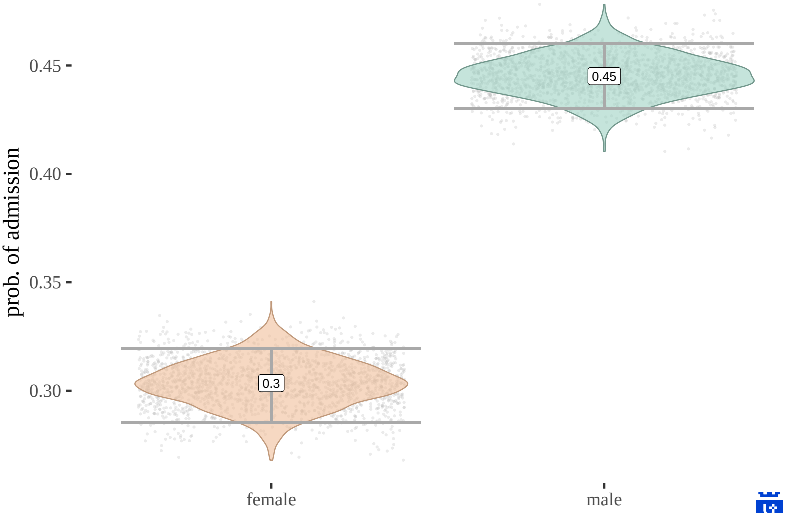
```
           mean         sd       5.5%      94.5%     rhat ess_bulk
a[1] -0.2194527 0.03823689 -0.2823439 -0.1570496 1.001694 1556.256
a[2] -0.8322337 0.05028066 -0.9125176 -0.7529984 1.004124 1148.146
```

Contrast

```
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)
```

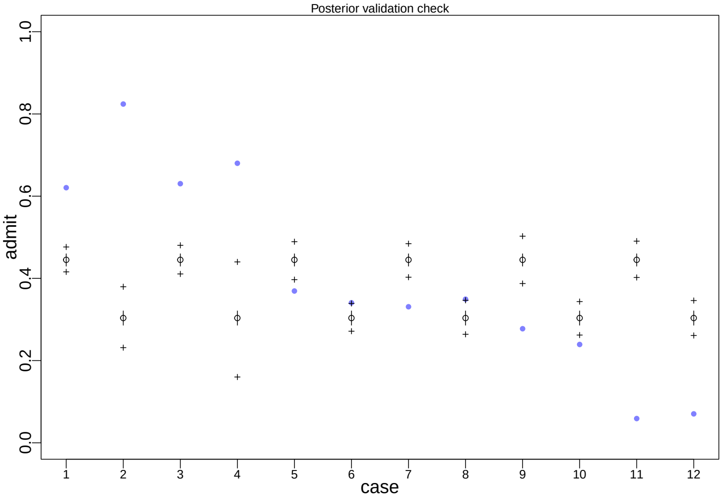


UCB Probability of Admission

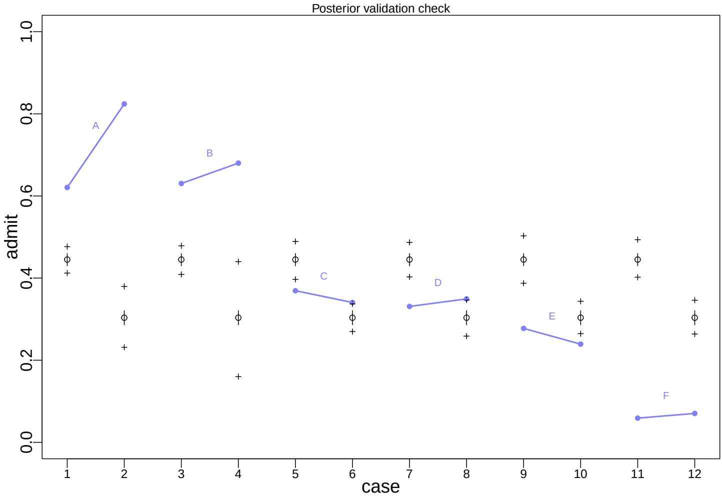


UC Berkeley admissions

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



UC Berkeley admissions



University of Gdańsk

Within departments

```r
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 )
```

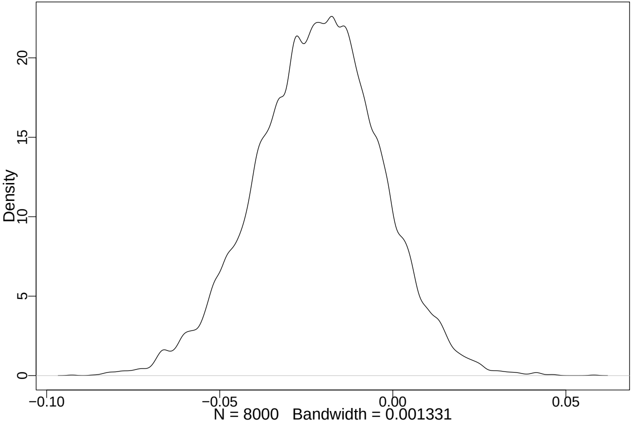
Within departments

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

```
               mean        sd       5.5%      94.5%     rhat ess_bulk
a[1]     -0.5383123 0.5321202 -1.4205526  0.2986248 1.010772 661.4652
a[2]     -0.4418056 0.5329128 -1.3216779  0.3881090 1.009861 671.4347
delta[1]  1.1196447 0.5347978  0.2784980  1.9989427 1.010024 666.6947
delta[2]  1.0744904 0.5362065  0.2326648  1.9528943 1.010223 672.8193
delta[3] -0.1409889 0.5345427 -0.9684240  0.7364742 1.010480 668.0050
delta[4] -0.1731733 0.5354684 -1.0100719  0.7093724 1.010411 667.1571
delta[5] -0.6163853 0.5394283 -1.4643872  0.2655678 1.010432 681.4037
delta[6] -2.1727340 0.5462799 -3.0189259 -1.2775413 1.009070 710.5595
```

Within departments

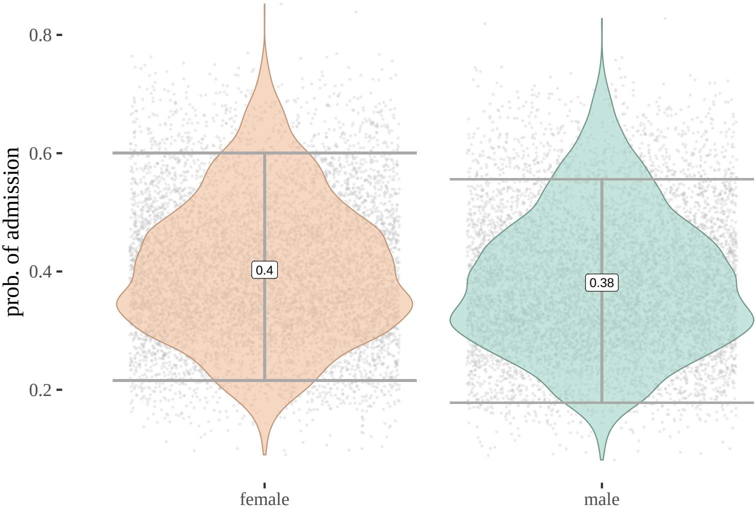
```
post <- extract.samples(ucbModelWithin)
diff_p <- inv_logit(post$a[,1]) - inv_logit(post$a[,2])
dens(diff_p,  cex.axis=1.3, cex.lab=1.5)
```



Within departments

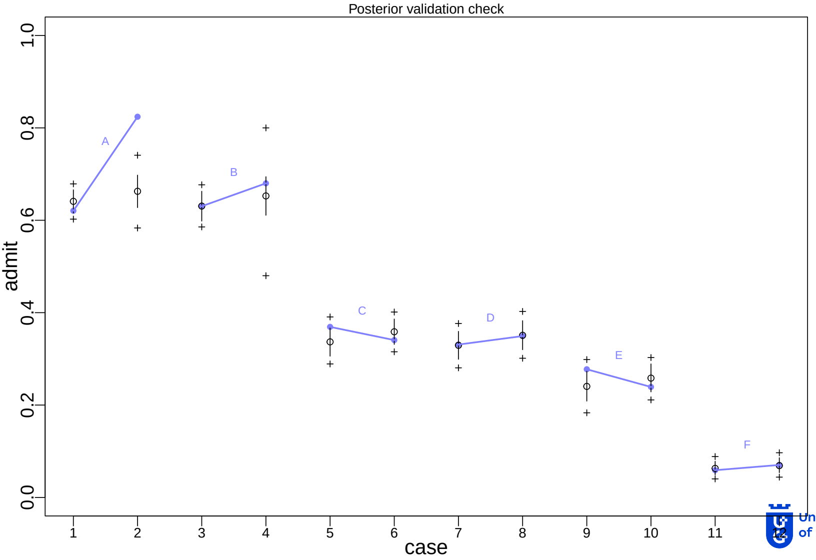
```
                  A    B    C    D    E    F
male           0.88 0.96 0.35 0.53 0.33 0.52
female         0.12 0.04 0.65 0.47 0.67 0.48
multiplicative 0.75 0.75 0.46 0.46 0.35 0.10
```

UCB probability of admission (knowing the departments)



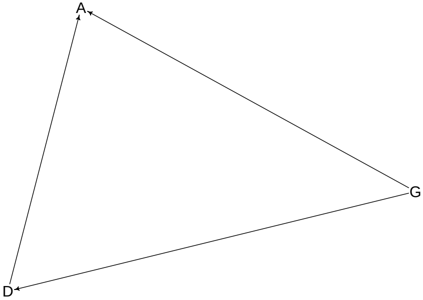
University of Gdańsk

UC Berkeley admissions



Within departments

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

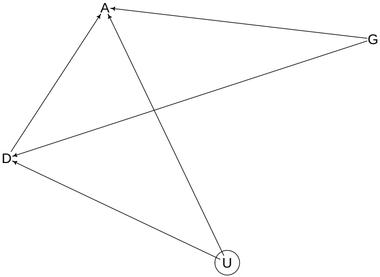


```
adjustmentSets(ucbDAG, exposure = "G",
               outcome = "A", effect = "direct")
```

```
{ D }
```

Within departments

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



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

- NONE!