§14.1 Multilevel Logistic Regression

Response is binary. Useful in analyzing survey data, medical treatments, ...
Varying Intercepts: Use binomial likelihood for response, Gaussian for "random effects" in the intercepts.

$$\begin{aligned} P(y_i = 1) &= \text{invlogit}(\alpha_{j[i]} + \mathbf{x}_i\boldsymbol{\beta}), \ i = 1, \ldots, n \\ \alpha_j &\sim iid\ N(\mu_\alpha, \sigma^2_\alpha) \end{aligned}$$

OR

$$P(y_i = 1) = \text{invlogit}(\mathbf{x}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_{j[i]})$$

$$\mathbf{b}_j \sim iid\ N(0, \sigma^2_\alpha), \ j = 1, \ldots, J$$

State Estimates based on National Poll

To estimate a response from all eligible voters, we might post-stratify. Use mixed model to predict responses for each subcategory, then use weighted average (with weights based on census) to get overall estimate.

1. Break up the data into $2 \cdot 2 \cdot 4 \cdot 4 \cdot 51 = 3264$ strata:
   - gender (2)
   - race (2)
   - age (4)
   - education (4)
   - state (51)

   Estimate effects for each cross classification (if not missing)
2. Combine estimates together $\theta_j = \sum_{i \in j} N_i\theta_i / \sum_{i \in j} N_i$ using census data for each state with N_i people in category $i = 1, \ldots, 3264$

Raw Proportions

```
cat("national mean of raw data:", round(mean(poll9$y), 3),
    "\n")

## national mean of raw data: 0.558

cat("national weighted mean of raw data:", with(poll9, round(sum((weight *
    y))/sum(weight), 3), "\n"))

## national weighted mean of raw data: 0.543

(weighted.raw <- with(poll9, tapply(y * weight, state, sum)/tapply(weight,
    state, sum)))

##     1     3     4     5     6     7     8     9    10
## 0.755 0.617 0.458 0.492 0.600 0.570 0.457 0.000 0.599
##    11    13    14    15    16    17    18    19    20
## 0.588 0.254 0.430 0.599 0.267 0.783 0.692 0.502 0.600
##    21    22    23    24    25    26    27    28    29
## 0.200 0.323 0.511 0.324 0.571 0.545 0.426 0.597 0.658
##    30    31    32    33    34    35    36    37    38
## 0.742 0.537 0.400 0.436 0.628 0.459 0.603 0.635 0.365
##    39    40    41    42    43    44    45    46    47
## 0.536 0.493 0.633 0.431 0.801 0.563 1.000 1.000 0.700
```


lmer fit in equation (14.2)

```
pollFit <- glmer(y ~ black + female + black:female + v.prev
    (1 | age) + (1 | edu) + (1 | age.edu) + (1 | state) +
    (1 | region), data = poll9, family = binomial(link = "l
displayFixed.mer(pollFit)
```

	Estimate	SE	Ratio
(Intercept)	0.18	0.20	0.88
black	-1.65	0.33	-5.07
female	-0.09	0.10	-0.91
v.prev	4.72	1.44	3.29
black:female	-0.17	0.42	-0.42

```
sapply(VarCorr(pollFit), function(x) x[[1]])

##   state age.edu  region     edu     age
## 0.03961 0.01590 0.09408 0.01663 0.00274
```

Interpret lmer fit

1. For white male when previous vote was 50% Rep: 0.18 (.20)
2. Black: -1.65 (0.33) black men are 40% less likely to vote R
3. Female: -0.09 (0.10) little/no evidence of a shift
4. v.prev: 4.7 (1.4) 3.3 SEs or shift of 1.2% when v.prev increases by 1%
5. black:female: -0.17 (.4) little/no evidence of a shift

MCMC fits would better show all components of variation.

MCMC

Markov Chain Monte Carlo sampling

"Monte Carlo" means random sample (roulette wheel?)
"Markov Chain" means there is a stepping process; sample values at step j depend on where values at step $j-1$.
In practice: use conditional distributions to cycle through the various parameters.
In a one level linear model with noninformative priors,
$\boldsymbol{\beta}|X,\ \mathbf{y},\ \sigma \sim$ normal.
$\sigma|\boldsymbol{\beta},\ \mathbf{X},\ \mathbf{y} \sim$ scaled inverse χ^2.
Multilevel models add more steps, and various computational tricks are used: Metropolis-Hastings, Gibbs sampling, ...
Result: a sample of size J for each parameter of interest. Cautions:
Math theory guarantees convergence to the posterior distribution of interest as $J \to \infty$
First several 1000 samples are often discarded as "burn-in".
Use convergence diagnostics to see if we need more cycles.

JAGs code in model file

```
model {  for(i in 1:n){
    y[i] ~ dbin (p[i], 1)
    logit(p[i]) <- Xbeta[i]
    Xbeta[i] <- b.0 + b.female*female[i] +
      b.black*black[i] +
      b.female.black*female[i]*black[i] +
      a.age[age[i]] + a.edu[edu[i]] +
      a.age.edu[age[i],edu[i]] +
      a.state[state[i]]
  }
  for (j in 1:n.age) {a.age[j] ~ dnorm(0, tau.age)}
  for (j in 1:n.edu) {a.edu[j] ~ dnorm(0, tau.edu)}
  for (j in 1:n.age) {for (k in 1:n.edu){
    a.age.edu[j,k] ~ dnorm(0, tau.age.edu)}}
  for (j in 1:n.state) {
    a.state[j] ~ dnorm(a.state.hat[j], tau.state)
    a.state.hat[j] <- a.region[region[j]] + b.v.prev*
   }
   b.v.prev ~ dnorm (0, .0001)
  for (j in 1:n.region) {a.region[j] ~ dnorm(0, tau.region)}
```

```
  b.0 ~ dnorm (0, .0001)
  b.female ~ dnorm (0, .0001)
  b.black ~ dnorm (0, .0001)
  b.female.black ~ dnorm (0, .0001)

  sigma.age <- pow(tau.age, -.5)
  sigma.edu <- pow(tau.edu, -.5)
  sigma.age.edu <- pow(tau.age.edu, -.5)
  sigma.state <- pow(tau.state, -.5)
  sigma.region <- pow(tau.region, -.5)

  tau.age ~ dgamma(1.0e-03, 1.0e-03)
  tau.edu ~ dgamma(1.0e-03, 1.0e-03)
  tau.age.edu ~ dgamma(1.0e-03, 1.0e-03)
  tau.state ~ dgamma(1.0e-03, 1.0e-03)
  tau.region ~ dgamma(1.0e-03, 1.0e-03)
}
```

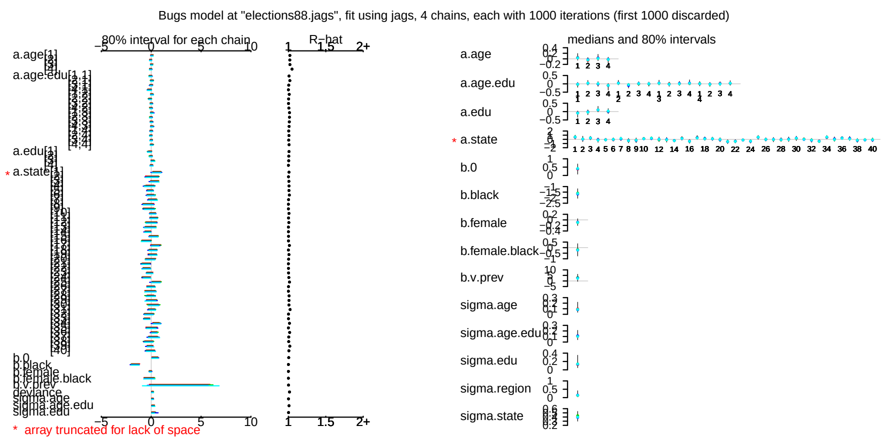
save in file: `elections88.jags`

Call JAGs from R

```
require(R2jags)
modelFile <- "elections88.jags"
data4JAGs <- with( poll9, list( "y" = y,
            "female" = female, "black" = black,
            "age" = age, "edu" = edu,"state" = state,
            "region" = region, "v.prev" = v.prev,
      "n" = nrow(poll9), "n.age" = 4,"n.edu" = 4,
      "n.state" = 51, "n.region" = 5  ))
set.seed(314160)
electionJAGs <- jags(model.file=modelFile,
          data= data4JAGs, parameters.to.save=c("b.0",
          "b.female","b.black","b.female.black","b.v.prev",
          "sigma.age","sigma.edu","sigma.age.edu","sigma.state",
          "sigma.region",  "a.age","a.edu","a.age.edu", "a.state" ),
              ##   inits=list( ),
            n.chains=4, n.iter=1000,  n.burnin= 500,  n.thin=10)
electionJAGs <- autojags(electionJAGs,  n.iter = 1000,
                           n.thin =10, Rhat = 1.05 )
```

Plot JAGs Output

```
plot(electionJAG2)
```



Print JAGs Output

```
Inference for Bugs model at "elections88.jags", fit using jags,
 4 chains, each with 1000 iterations (first 1000 discarded), n.thin = 10
 n.sims = 400 iterations saved
                mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
a.age[1]          0.061   0.122  -0.117  -0.005   0.039   0.110   0.338 1.019   400
a.age[2]         -0.049   0.113  -0.299  -0.102  -0.035   0.011   0.146 1.019   400
a.age[3]          0.037   0.117  -0.153  -0.020   0.023   0.081   0.295 1.020   400
a.age[4]         -0.041   0.119  -0.267  -0.098  -0.021   0.020   0.123 1.049   400
a.age.edu[1,1]   -0.050   0.140  -0.357  -0.103  -0.028   0.024   0.169 1.009   400
a.age.edu[2,1]    0.054   0.130  -0.156  -0.021   0.036   0.118   0.317 1.019   290
a.age.edu[3,1]   -0.003   0.125  -0.297  -0.053  -0.003   0.064   0.224 1.005   400
a.age.edu[4,1]   -0.144   0.180  -0.587  -0.225  -0.095  -0.019   0.091 1.000   400
a.age.edu[1,2]    0.058   0.114  -0.132  -0.014   0.039   0.122   0.334 1.011   400
a.age.edu[2,2]   -0.091   0.123  -0.372  -0.155  -0.065  -0.007   0.105 1.003   400
a.age.edu[3,2]    0.009   0.102  -0.205  -0.052   0.009   0.069   0.219 1.002   400
a.age.edu[4,2]   -0.008   0.104  -0.231  -0.068   0.002   0.054   0.208 1.006   290
a.age.edu[1,3]    0.061   0.124  -0.140  -0.017   0.037   0.119   0.366 1.015   260
 [ reached getOption("max.print") -- omitted 73 rows ]

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 74.5 and DIC = 2686.0
DIC is an estimate of expected predictive error (lower deviance is better).
```


Plot Effects Code

Fig 14.1 uses MCMC samples, shows 50% and 95% posterior intervals.
Fig 14.2 uses plots of several draws from linear predictor:

$$\widehat{v}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_{1i} + \widehat{\beta}_2 x_{2i} + \widehat{\beta}_3 x_{3i} + \widehat{\alpha}_{\text{age},k[i]} + \widehat{\alpha}_{\text{educ},l[i]} + \widehat{\alpha}_{\text{age-educ},k[i],l[i]}$$

$$\widehat{P}(y_i = 1) = \text{logit}^{-1}(\widehat{v}_i + \widehat{\alpha}_{\text{state},j[i]})$$

Typo in legend of 14.3: $\alpha_j^{\text{state}} = \alpha_{m[j]}^{\text{region}} + \beta^{\text{vprev}}\text{v.prev}_j$. β does not change with j.

Obtain State-wide Estimates

via weighted average of subgroups in state

```
L <- nrow(census)
y.pred <- array(NA, c(400, L))
for (el in 1:L) {
    y.pred[, el] <- invlogit(b.0 + a.state[, census$state[el]] + a.age[, census
        a.edu[, census$edu[el]] + a.age.edu[, census$age[el] + (census$edu[el]
        b.female * census$female[el] + b.black * census$black[el] + b.female.bl
        census$black[el])
}
# average over strata within each state
y.pred.state <- array(NA, c(400, 51))
for (s in 1:400) {
    y.pred.state[s, ] <- tapply(y.pred[s, ] * census$N, census$state, sum)/tapp
        census$state, sum)
}
# average over strata within each state
state.pred <- apply(y.pred.state, 2, quantile, c(0.025, 0.25, 0.5, 0.75, 0.975)
```

How well did we do?

Fig 14.5 – shows we did well.
14.6 how well do raw estimates do?

§14.2 Connecticut Outlier

Republicans do better in poor states, Dems in richer ones. But high income folks tend to favor Republicans within any state.
Adjustments in Fig 14.9, 10, 11, 12

§14.3 Item Response and Ideal Point models

Supreme Court voting

A judge has a position on the theoretical liberal – conserv scale: α_j. A case has character as well: β_k. A "yes" vote (1) is coded to mean more conservative than a "no" (0).

$$P(y_i = 1) = \text{invlogit}(\alpha_{j[i]} - \beta_{k[i]})$$

Identifiability problem:
Adding a constant to α's and subtracting it from β's gives same likelihood. Need to constrain one set to add to 0, or set one of the α's or β's to 0 as a reference level.
More general: $\text{invlogit}(\gamma_{k[i]}(\alpha_{j[i]} - \beta_{k[i]}))$ where γ's are *discrimination* of the case. Near 0 means votes are up in the air, far from 0 means votes are more predictable given α and β.
`http://sct.tahk.us/background.html`