MLE - Lab 12

Andy Ballard

April 7, 2017

Today

- Homework 5
- More Hierarchical Models

Homework 5

You'll have two datasets to work with, each of which will ask you to do things with hierarchical models. The first examines findings from Steenbergen and Jones (AJPS 1996) who model support for the EU as a function of individual- and country-level variables.

```r
load(paste0(labPath, "EUsupport.RData")); eu <- EUsupport; rm(EUsupport)
head(eu)
```

```
##   age support inclow inchi lright      olead male country eutrade tenure
## 1  19       8      1     0      1  1.6012008    1 AUSTRIA   0.658      1
## 2  52       8      0     1      9  1.6012008    1 AUSTRIA   0.658      1
## 3  29       6      0     1      8 -0.3987992    0 AUSTRIA   0.658      1
## 4  49       8      0     0      0 -0.3987992    1 AUSTRIA   0.658      1
## 5  19       8      0     0      4 -0.3987992    1 AUSTRIA   0.658      1
## 6  31       6      0     0      5 -0.3987992    0 AUSTRIA   0.658      1
##     gdp   infl     gdpz   tenurez       tradez      inflz cntry
## 1 24017 0.9205 1.012186 -1.253767 -0.009633879 -0.6628329     1
## 2 24017 0.9205 1.012186 -1.253767 -0.009633879 -0.6628329     1
## 3 24017 0.9205 1.012186 -1.253767 -0.009633879 -0.6628329     1
## 4 24017 0.9205 1.012186 -1.253767 -0.009633879 -0.6628329     1
## 5 24017 0.9205 1.012186 -1.253767 -0.009633879 -0.6628329     1
## 6 24017 0.9205 1.012186 -1.253767 -0.009633879 -0.6628329     1
```

The second dataset describes contraception usage in Bangladesh, from the Bangladesh fertility survey in 1989. You'll model the use of contraception with a hierarchical model, where individuals are nested within districts.

```r
load(paste0(labPath, "contraception.RData"))
head(contra)
```

```
##   contrac urban        age children district
## 1       0     1  18.440001        3        1
## 2       0     1  -5.559990        0        1
## 3       0     1   1.440001        2        1
## 4       0     1   8.440001        3        1
## 5       0     1 -13.559900        0        1
## 6       0     1 -11.559900        0        1
```

Hierarchical Models

We'll use the same data as last week, from the American National Election Study (1990-2000). To refresh your memory, here are the variables:

partyid7: Party identification (Left-right, 7pt. Scale; Strong Dem = 1) state: State age: Age in years female: Female dummy black: Black dummy year: Year of survey married: Married dummy educ: Educational attainment (1-4) urban: 1=urban, 2=suburban, 3=rural union: Union member dummy south: Southern state dummy

```
load(paste0(labPath, "partyid.RData")); pid <- partyid; rm(partyid)
pid$union <- 2 - pid$union #recode to [0,1]
#Alternatively, the recode function ('car' package) is very useful, particularly for more complicated v
pid$union <- recode(pid$union, "2=0")

pid$party <- recode(pid$partyid7, "1='Democrat'; 2='Democrat'; 3='Democrat'; 5='Republican'; 6='Republi

pid90 <- pid[pid$year >= 1990,] #So we don't run into those weird southern Dems from the 1970s

head(pid)
```

```
##   year resid age educ urban union partyid7 black female south married
## 1 1992  2292  64    2     2     0        3     0      1     0       0
## 2 1992   679  40    4     1     0        2     0      1     0       1
## 3 1992  2217  40    4     1     1        2     0      1     0       0
## 4 1984  1687  31    4     3     1        7     0      1     0       1
## 5 1980   233  59    2     2     1        1     0      1     0       1
## 6 1986  1673  48    3     1     0        5     0      1     0       1
##       blackm state      party
## 1 0.04166667     1   Democrat
## 2 0.04166667     1   Democrat
## 3 0.04166667     1   Democrat
## 4 0.04166667     1 Republican
## 5 0.04166667     1   Democrat
## 6 0.04166667     1 Republican
```

We even ran a model predicting party ID. Well, we ran a few models. First, we had a pooled model with no inferred hierarchy.

```
m.pooled <- lm(partyid7 ~ urban + union + south + female + age +
                 black, data=pid90)
summary(m.pooled)
```

```
##
## Call:
## lm(formula = partyid7 ~ urban + union + south + female + age +
##     black, data = pid90)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4388 -1.7857 -0.2252  1.7575  5.4447
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.258902   0.206758  20.598  < 2e-16 ***
## urban        0.097525   0.068353   1.427 0.153839
## union       -0.710181   0.138520  -5.127 3.31e-07 ***
## south        0.054706   0.119553   0.458 0.647309
## female      -0.363461   0.101319  -3.587 0.000344 ***
## age         -0.005772   0.002915  -1.980 0.047902 *
## black       -1.784467   0.162688 -10.969  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.003 on 1577 degrees of freedom
## Multiple R-squared:  0.1065, Adjusted R-squared:  0.1031
## F-statistic: 31.32 on 6 and 1577 DF,  p-value: < 2.2e-16
```

Then we ran a model with fixed effects for year.

```
m.unpooled <- lm(partyid7 ~ urban + union + south + female + age + black + factor(year) - 1, data=pid90)
summary(m.unpooled)
```

```
##
## Call:
## lm(formula = partyid7 ~ urban + union + south + female + age +
##     black + factor(year) - 1, data = pid90)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5027 -1.7286 -0.2299  1.7709  5.4792
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## urban             0.099233   0.068388   1.451 0.146971
## union            -0.698838   0.138785  -5.035 5.32e-07 ***
## south             0.059192   0.120251   0.492 0.622618
## female           -0.361539   0.101382  -3.566 0.000373 ***
## age              -0.006106   0.002934  -2.081 0.037583 *
## black            -1.791863   0.162841 -11.004  < 2e-16 ***
## factor(year)1990  4.175439   0.226887  18.403  < 2e-16 ***
## factor(year)1992  4.240439   0.228487  18.559  < 2e-16 ***
## factor(year)1994  4.391373   0.226350  19.401  < 2e-16 ***
## factor(year)1996  4.348103   0.238418  18.237  < 2e-16 ***
## factor(year)1998  4.093849   0.240169  17.046  < 2e-16 ***
## factor(year)2000  4.548419   0.319437  14.239  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.002 on 1572 degrees of freedom
## Multiple R-squared:  0.7792, Adjusted R-squared:  0.7775
## F-statistic: 462.4 on 12 and 1572 DF,  p-value: < 2.2e-16
```

Then we ran a random intercept model for individuals within years.

```
m0 <- lmer(partyid7 ~ 1 + south + (1 | year), data=pid)
summary(m0)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: partyid7 ~ 1 + south + (1 | year)
##    Data: pid
##
## REML criterion at convergence: 17158.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.3488 -0.8533 -0.2378  1.0580  1.7165
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept) 0.004238 0.0651
##  Residual             4.329906 2.0808
## Number of obs: 3985, groups:  year, 15
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  3.77029    0.04218   89.38
## south       -0.28870    0.07506   -3.85
##
## Correlation of Fixed Effects:
##       (Intr)
## south -0.467
```

What do we mean by the difference between random effects and fixed effects? Turns out, there are lots of different definitions. Check out this piece by Andrew Gelman (http://www.stat.columbia.edu/~gelman/research/published/AOS259.pdf) about a bunch of different defintions:

1. Fixed effects are constant across individuals, and random effects vary. For example, in a growth study, a model with random intercepts a_i and fixed slope b corresponds to parallel lines for different individuals i, or the model $y_{it} = a_i + bt$. Kreft and De Leeuw (1998) thus distinguish between fixed and random coefficients.
2. Effects are fixed if they are interesting in themselves or random if there is interest in the underlying population. Searle, Casella, and McCulloch (1992, Section 1.4) explore this distinction in depth.
3. "When a sample exhausts the population, the corresponding variable is fixed; when the sample is a small (i.e., negligible) part of the population the corresponding variable is random." (Green and Tukey, 1960)
4. "If an effect is assumed to be a realized value of a random variable, it is called a random effect." (LaMotte, 1983)
5. Fixed effects are estimated using least squares (or, more generally, maximum likelihood) and random effects are estimated with shrinkage ("linear unbiased prediction" in the terminology of Robinson, 1991). This definition is standard in the multilevel modeling literature (see, for example, Snijders and Bosker, 1999, Section 4.2) and in econometrics.

Activity

Are we using any single one of these definitions? Are they mutually exclusive? Take five minutes and talk about this with your neighbors. How do you think of fixed/random effects, how have they been taught in this course?

Random Slope Models

Our random intercept model above ($m0$) is a random intercept model. This means that we estimate a different intercept for each year, but that each of these is a parallel line. Now, we'll allow there to be a different slope for each year.

```r
m.slope <- lmer(partyid7 ~ 1 + south + (south | year), data=pid)
summary(m.slope)
```

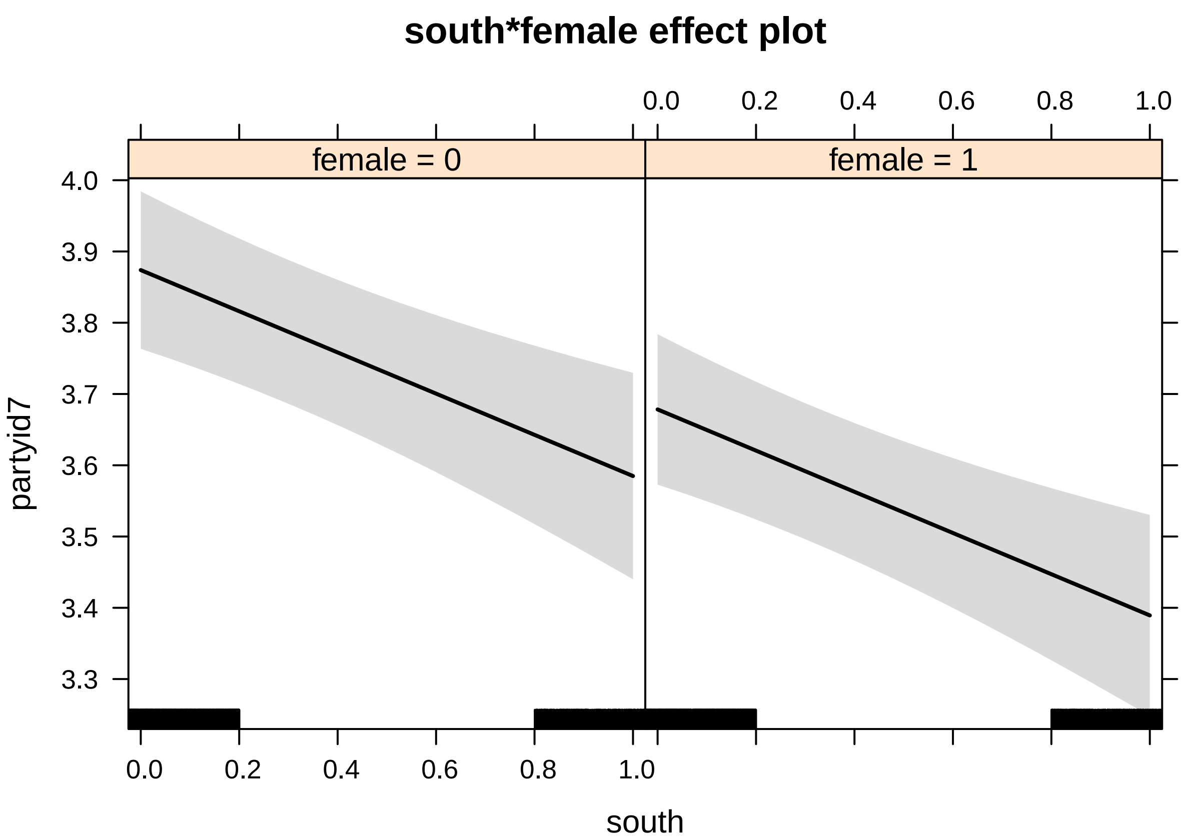
```
## Linear mixed model fit by REML ['lmerMod']
```

```
## Formula: partyid7 ~ 1 + south + (south | year)
##    Data: pid
##
## REML criterion at convergence: 17158.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.3557 -0.8548 -0.2336  1.0519  1.7072
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  year     (Intercept) 0.006705 0.08188
##           south       0.001976 0.04446  -1.00
##  Residual             4.328969 2.08062
## Number of obs: 3985, groups:  year, 15
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  3.76995    0.04424   85.22
## south       -0.28909    0.07598   -3.80
##
## Correlation of Fixed Effects:
##       (Intr)
## south -0.517
```

We ran into a strange phenomenon last time. Because our data span 1972-2000, we captured a transitional period in American party politics and the effect of the `south` variable in the pooled model is very unlike we would see today. So the `m.slope` model above allows for random slopes for the `south` variable for each year. This way, we can see how things change over time.

I found an extremely cool package that I've been playing with, and now I'm going to show it to you. It's the `effects` package, which creates objects for constructing effects plots. It does a lot of what we've been doing for prediction for us.

```r
pid$female <- as.factor(pid$female)
m.slope2 <- lmer(partyid7 ~ 1 + south + female +  (south | year), data=pid)
plot(Effect(c("south", "female"), m.slope2))
```



Whoa, huh? I just found this last night, so I haven't done a ton of exploring. But if the options are fairly flexible you could do all sorts of cool things with this.

As is, it doesn't help us a ton. This is just the fixed effect, but we want to look at the random effects and the random slopes.

Now we'll do another group activity, and then we'll build a plot of the random slopes and intercepts together.

Activity

Pick a year of ANES data. Compute what the model thinks that the party ID value will be for southern and nonsouthern individuals. HINT: Look at the lecture slides that were distributed with this code, a bit more than halfway down.

Here's the example I did, for 1972. First, here is the estimate for party ID based on the model.

$$\hat{y}_j = (\mu_\alpha + \epsilon_j) + (\mu_\beta + \xi_j)x$$

```
#Year 1972
sum(fixef(m.slope) + ranef(m.slope)[[1]][1,]) #southern
```

```
## [1] 3.48605
```

```
sum(fixef(m.slope)[1] + ranef(m.slope)[[1]][1,1]) #nonsouthern
```

```
## [1] 3.781307
```

```r
# All years
yhats <- matrix(data=NA, ncol = 2, nrow=length(unique(pid$year)))
colnames(yhats) <- c("Nonsouthern", "Southern")
rownames(yhats) <- sort(unique(pid$year))
yhats
```

```
##      Nonsouthern Southern
## 1972          NA       NA
## 1974          NA       NA
## 1976          NA       NA
## 1978          NA       NA
## 1980          NA       NA
## 1982          NA       NA
## 1984          NA       NA
## 1986          NA       NA
## 1988          NA       NA
## 1990          NA       NA
## 1992          NA       NA
## 1994          NA       NA
## 1996          NA       NA
## 1998          NA       NA
## 2000          NA       NA
```

```r
for(i in 1:length(unique(pid$year))){
  yhats[i,1] <- sum(fixef(m.slope) + ranef(m.slope)[[1]][i,])
  yhats[i,2] <- sum(fixef(m.slope)[1] + ranef(m.slope)[[1]][i,1])
}
yhats
```

```
##      Nonsouthern Southern
## 1972    3.486050 3.781307
## 1974    3.483422 3.775557
## 1976    3.499808 3.811406
## 1978    3.469851 3.745865
## 1980    3.497043 3.805357
## 1982    3.447955 3.697960
## 1984    3.504037 3.820660
## 1986    3.456410 3.716458
## 1988    3.501135 3.814311
## 1990    3.472119 3.750828
## 1992    3.468939 3.743870
## 1994    3.499719 3.811212
## 1996    3.484786 3.778542
## 1998    3.454901 3.713158
## 2000    3.486737 3.782810
```

Plotting random intercepts and slopes

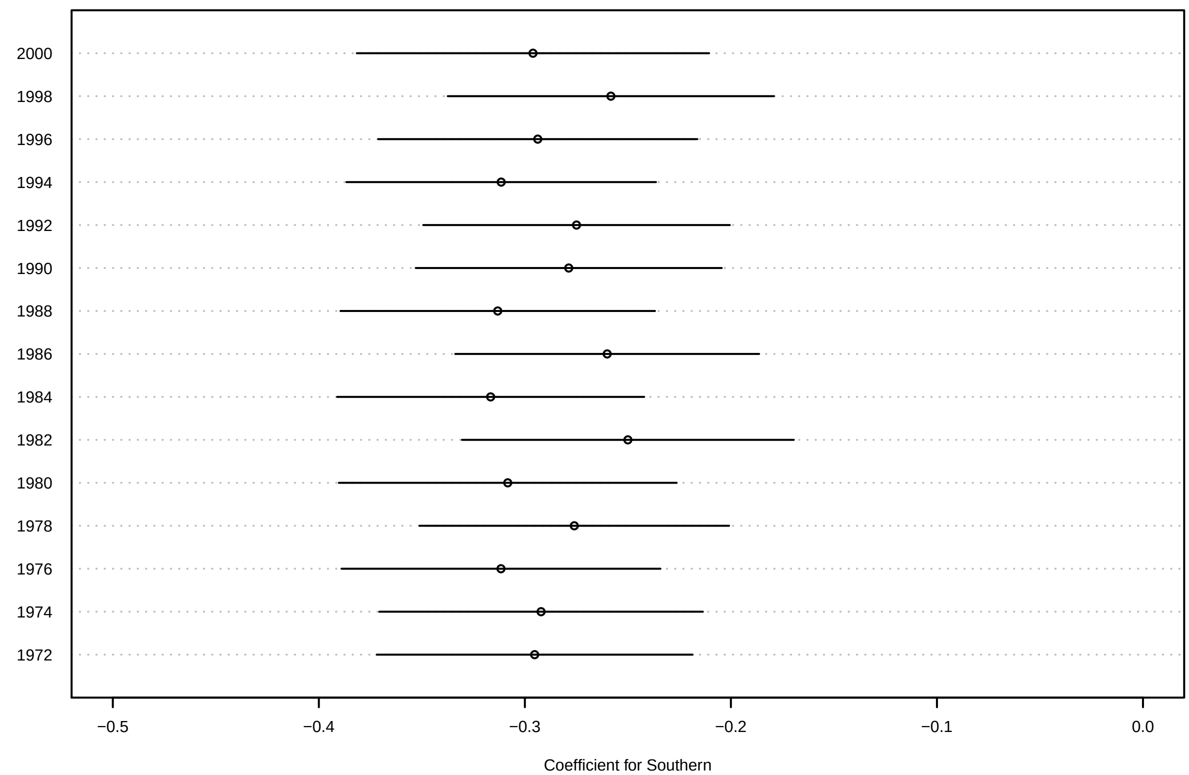
Okay, now that we've figured out how to compute estimated values of party ID based on different levels of our predictor variable `south`, let's construct a plot of the random slopes.

```r
# Extract fixed effects
a <- fixef(m.slope)
south.fe <- a[2]
```

```
# Extract random effects
b <- ranef(m.slope, condVar=TRUE)
south.res <- b[[1]][2]

# Extract the variances of the random effects
qq <- attr(b[[1]], "postVar")
e <- (sqrt(qq))
```

```
## Warning in sqrt(qq): NaNs produced
```

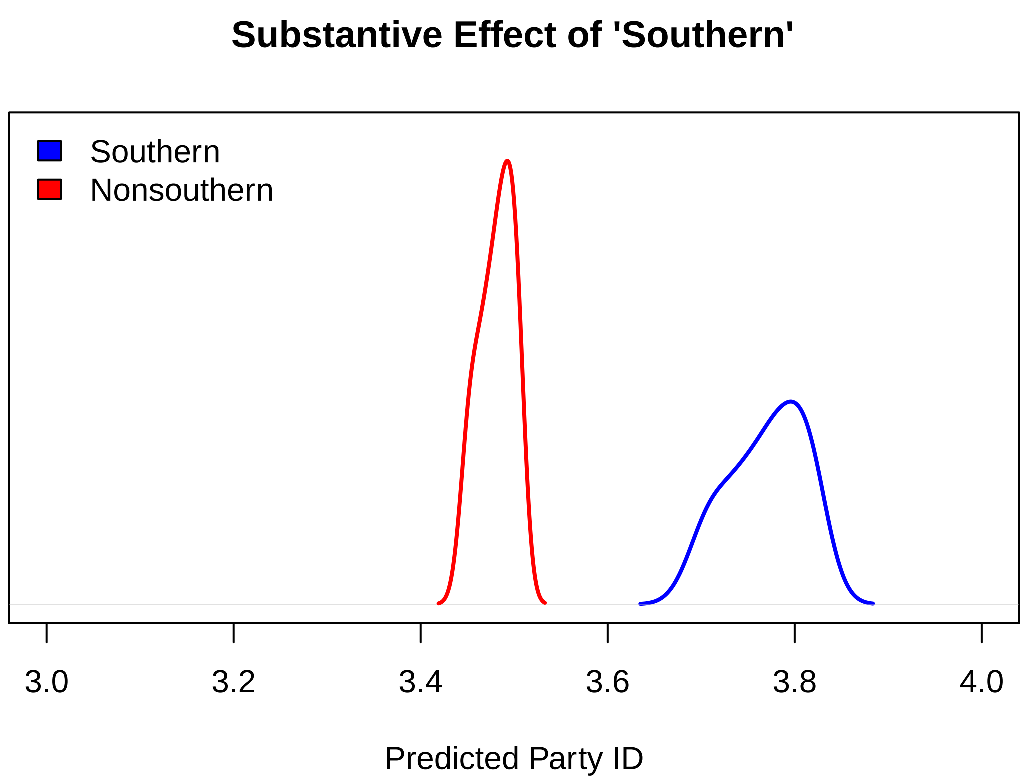
```
e <- e[2,2,] #here we want to access `south`, which is stored in column 2 in b[[1]], that's why I use t

# Calculate CI's

lo <- (south.res+south.fe)-(e*2)
mu <- (south.res+south.fe)
hi <- (south.res+south.fe)+(e*2)

#Plot betas and CIs
dotchart(mu$south, labels = rownames(mu), cex = 0.5,
         xlim = c(-0.5,0), xlab = "Coefficient for Southern")
for (i in 1:nrow(mu)){
  lines(x = c(lo[i,1], hi[i,1]), y = c(i,i))
}
```



Prediction

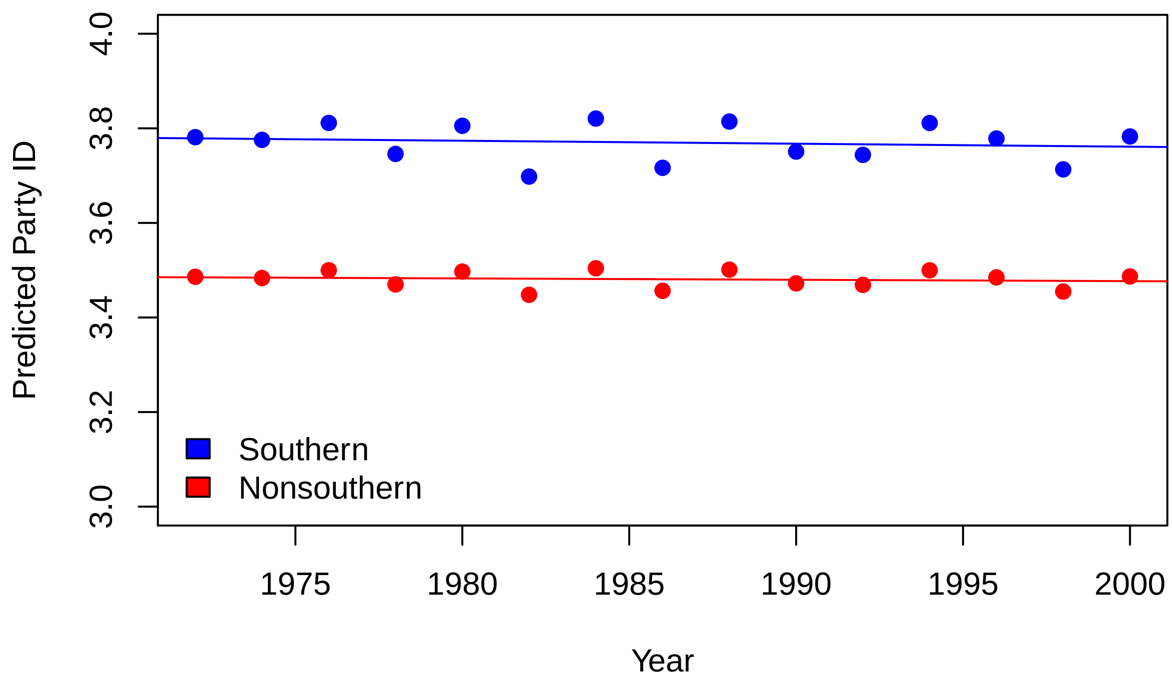
Let's look at some predicted values of party ID, based on our random slope model.

```
scen1 <- data.frame(partyid7=seq(min(pid$partyid7),
                                  max(pid$partyid7),
                                  length.out=length(unique(pid$year))),
                    south=0,
                    year=sort(unique(pid$year)))
pred1 <- predict(m.slope, newdata=scen1)


scen2 <- data.frame(partyid7=seq(min(pid$partyid7),
                                  max(pid$partyid7),
                                  length.out=length(unique(pid$year))),
                    south=1,
                    year=sort(unique(pid$year)))
pred2 <- predict(m.slope, newdata=scen2)

plot(density(pred1), lwd=2, col="blue", main="Substantive Effect of 'Southern'",
     xlab="Predicted Party ID", xlim=c(3,4), ylim=c(0,20), ylab="", yaxt="n")
lines(density(pred2), lwd=2, col="red")
legend("topleft", c("Southern", "Nonsouthern"), fill=c("Blue", "Red"), bty="n")
```




```
plot(sort(unique(pid$year)), pred1, col="blue",pch=19, ylim=c(3,4), ylab="Predicted Party ID", xlab="Yea
points(sort(unique(pid$year)), pred2, col="red", pch=19)
legend("bottomleft", c("Southern", "Nonsouthern"), fill=c("Blue", "Red"), bty="n")
```

```r
abline(lm(pred1~sort(unique(pid$year))), col="blue")
abline(lm(pred2~sort(unique(pid$year))), col="red")
```



Are we really taking uncertainty into account here? Are there easy ways to do this?