Naive Aggregate Model Creation

Theodore Dounias

10/10/2018

Introduction

```
voter_sample <- read_csv("~/Desktop/Reed_Senior_Thesis/Data_and_results/data/voter_sample.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   VOTER_ID = col_integer(),
##   COUNTY = col_character(),
##   ACTIVE = col_character(),
##   PARTY = col_character(),
##   GENDER = col_character(),
##   AGE = col_integer(),
##   ELECTION_TYPE = col_character(),
##   ELECTION_DATE = col_integer(),
##   MAIL_VOTE = col_integer(),
##   voted = col_integer()
## )
```

```
county_demographics <- read_csv("~/Desktop/Reed_Senior_Thesis/Data_and_results/data/colorado_demographi
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   COUNTY = col_character(),
##   URBAN = col_integer(),
##   RURAL = col_integer(),
##   PCT_URBAN = col_double(),
##   TOTAL_POP = col_integer(),
##   WHITE = col_integer(),
##   PCT_WHITE = col_double(),
##   TOTAL_REGISTERED = col_integer(),
##   REP = col_integer(),
##   DEM = col_integer(),
##   OTHER = col_integer(),
##   UAF = col_integer(),
##   PCT_REGISTERED = col_double(),
##   PCT_OF_STATE_POP = col_double(),
##   PCT_OF_STATE_REG = col_double()
## )
```

```
voter_sample <- left_join(voter_sample, county_demographics, by = "COUNTY")

voter_sample$VOTER_ID <- as.factor(voter_sample$VOTER_ID)
voter_sample$COUNTY <- as.factor(voter_sample$COUNTY)
```

```
voter_sample$PARTY <- as.factor(voter_sample$PARTY)
voter_sample$GENDER <- (voter_sample$GENDER == "Male")
voter_sample$ELECTION_TYPE <- as.factor(voter_sample$ELECTION_TYPE)
voter_sample$ELECTION_DATE <- as.factor(voter_sample$ELECTION_DATE)

voter_sample <- mutate(voter_sample,
                       ELECTION = paste(voter_sample$ELECTION_TYPE, voter_sample$ELECTION_DATE, sep = "

voter_sample$ELECTION <- as.factor(voter_sample$ELECTION)
```

For the rest of this write-up, assume the following:

$$y_i \sim \text{Bernoulli}(p\_vote)$$

Where $y_i \in \{0, 1\}$ is the probability that the i-th ballot was completed. Indexes and variables coded according to the following table:

Purpose/Unit	Variable	Index
Ballot	u	i
Individual	z	j
County	x	k
Election	w	l
Number of Vars	n	–
General Index	–	i’

Additionally, I do veer away a bit from the structure we talked about on Monday; I do not linearly add to the model until it reaches a final stage. The reasoning here is that there is no exact linear path to follow; there is an overarching unit of observation–the ballot–and all the rest are dependent between each other. For instance, adding a variable for Party at the ballot level would not significantly change the way I later add percentage of white residents at the county level. Therefore, the way I proceed is the following: I “build” the models step by step and separatelly for each group of variables (grouping by unit of observation). Then I present one example of what a model using two of these initial “building blocks” would look like. Since this is fairly generalizable, I then proceed directly to the full model which inlcudes all different variables.

No Data Guess

If receiving a ballot with no information, I would predict that the probability that an additional ballot was a vote in favor would be equal to turnout, as calculated through all other ballots. Therefore:

$$p\_\hat{v}ote_i = \frac{\#\text{votes cast}}{\#\text{ballots}}$$

Estimation with only one type of data

There are four levels of data I will go through here: County, Election, Person, and Ballot.

County Level

Assume that the ballot I am trying to assess completion for has the name of the county it is from written on it. There are two ways I can think of for predicting p_vote. First, assume that each different county has a different, independent p_vote. Therefore, in model-lingo this would look like:

$$p\_\hat{vote}_i \sim \text{logit}^{-1}(\sum_{k=1}^{64} x_{k,i}\beta_k)$$

Where k counts over the 64 counties of Colorado, and x_k is an indicator variable for each county. If I, quite reasonably, throw away the assumption of independance–these counties are, after all, in the same state and the same country–I could also fit a mixed effects model as such:

$$p\_\hat{vote} \sim \text{logit}^{-1}(a_{k[i]}),$$
$$a_k \sim \text{N}(\gamma_0, \sigma_\alpha^2)$$

Where $\alpha_{k[i]}$ varies by county, constrained by its standard deviation and γ_0, an intercept coefficient. Let's say now that along with the one ballot, I was given a short list of $n^{\text{county vars}}$ other county-level variables, be they discrete, continuous, or indicators. The two models would then look like:

$$p\_\hat{vote}_i \sim \text{logit}^{-1}(\sum_{k=1}^{64} x_k\beta_k + \sum_{i'=1}^{n^{\text{county vars}}} x_{k[i],i'}\beta_{i'+64})$$

Where $x_{k[i],l}$ is the k-th value of the i'-th variable. If, as before, I do not assume independance, the model can be written as:

$$p\_\hat{vote} \sim \text{logit}^{-1}(a_{k[i]}),$$
$$a_k \sim \text{N}(\gamma_0 + \sum_{i'=1}^{n^{\text{county vars}}} x_{k[i],i'}\gamma_{i'}, \sigma_\alpha^2)$$

In the case of my specific data, for the time being I have county-level data for white population and urban population, so $n^{\text{county vars}} = 2$.

```r
#Model 1
m1_c <- glm(data = voter_sample, family = "binomial", voted ~ COUNTY)
summary(m1_c)
```

```
##
## Call:
## glm(formula = voted ~ COUNTY, family = "binomial", data = voter_sample)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.229  -1.047  -0.924   1.294   1.810
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.538367   0.077306  -6.964 3.31e-12 ***
## COUNTYAlamosa       0.250685   0.447684   0.560 0.575506
## COUNTYArapahoe      0.223225   0.099536   2.243 0.024918 *
## COUNTYArchuleta     0.848522   0.404416   2.098 0.035892 *
## COUNTYBaca          2.617809   1.063474   2.462 0.013833 *
## COUNTYBent          0.538367   0.637163   0.845 0.398142
## COUNTYBoulder       0.267655   0.107826   2.482 0.013054 *
## COUNTYBroomfield    0.311594   0.196598   1.585 0.112982
## COUNTYChaffee       0.856821   0.337536   2.538 0.011134 *
## COUNTYCheyenne    -12.027696 229.628483  -0.052 0.958227
## COUNTYClear Creek   0.538367   0.399778   1.347 0.178088
## COUNTYConejos       1.326825   0.544872   2.435 0.014887 *
## COUNTYCostilla      1.049193   0.734377   1.429 0.153095
## COUNTYCrowley      -0.560245   1.157285  -0.484 0.628313
## COUNTYCuster        0.826049   0.767665   1.076 0.281903
## COUNTYDelta         0.503276   0.275995   1.823 0.068228 .
## COUNTYDenver       -0.091926   0.097463  -0.943 0.345587
## COUNTYDolores       0.132902   0.916138   0.145 0.884657
## COUNTYDouglas       0.296918   0.114660   2.590 0.009610 **
## COUNTYEagle         0.244128   0.227815   1.072 0.283898
## COUNTYEl Paso       0.054754   0.097942   0.559 0.576130
## COUNTYElbert        0.814621   0.293110   2.779 0.005449 **
## COUNTYFremont       0.106234   0.224827   0.473 0.636559
## COUNTYGarfield      0.484300   0.245091   1.976 0.048155 *
## COUNTYGilpin       -0.067768   0.513373  -0.132 0.894979
## COUNTYGrand         0.602906   0.367618   1.640 0.100998
## COUNTYGunnison      0.447396   0.311566   1.436 0.151015
## COUNTYHinsdale     13.104430 324.743705   0.040 0.967812
## COUNTYHuerfano      1.837650   0.655911   2.802 0.005084 **
## COUNTYJackson      -0.154780   1.227182  -0.126 0.899632
## COUNTYJefferson     0.340783   0.096980   3.514 0.000441 ***
## COUNTYKiowa       -12.027696 324.743705  -0.037 0.970455
## COUNTYKit Carson    0.132902   0.462936   0.287 0.774047
## COUNTYLa Plata     -0.038520   0.209917  -0.184 0.854405
## COUNTYLake          0.538367   0.582503   0.924 0.355366
## COUNTYLarimer       0.333336   0.108592   3.070 0.002143 **
## COUNTYLas Animas    0.473829   0.367618   1.289 0.197427
## COUNTYLincoln       1.924662   0.794340   2.423 0.015394 *
## COUNTYLogan         0.612475   0.392846   1.559 0.118979
## COUNTYMesa          0.077192   0.136802   0.564 0.572579
## COUNTYMineral      13.104430 324.743705   0.040 0.967812
## COUNTYMoffat        0.276003   0.427667   0.645 0.518688
## COUNTYMontezuma     0.270103   0.271753   0.994 0.320257
## COUNTYMontrose      0.618410   0.243711   2.537 0.011166 *
## COUNTYMorgan        0.371313   0.299820   1.238 0.215548
## COUNTYOtero         0.395267   0.386738   1.022 0.306756
## COUNTYOuray         2.936263   1.047323   2.804 0.005054 **
## COUNTYPark          0.236087   0.329056   0.717 0.473087
## COUNTYPhillips     -0.154780   0.869469  -0.178 0.858710
## COUNTYPitkin       -0.359574   0.365954  -0.983 0.325821
## COUNTYProwers       0.433007   0.465926   0.929 0.352710
## COUNTYPueblo        0.092461   0.146961   0.629 0.529248
## COUNTYRio Blanco    0.761511   0.480600   1.585 0.113080
## COUNTYRio Grande    0.856821   0.471047   1.819 0.068916 .
## COUNTYRoutt         0.093682   0.267614   0.350 0.726292
## COUNTYSaguache      0.356046   0.610445   0.583 0.559721
## COUNTYSan Juan      0.538367   1.002984   0.537 0.591429
## COUNTYSan Miguel   -0.640288   0.576974  -1.110 0.267114
## COUNTYSedgwick     13.104430 324.743705   0.040 0.967812
## COUNTYSummit       -0.883018   0.307646  -2.870 0.004102 **
## COUNTYTeller       -0.130682   0.279259  -0.468 0.639812
## COUNTYWashington    0.315224   0.675260   0.467 0.640630
## COUNTYWeld          0.004727   0.127066   0.037 0.970324
## COUNTYYuma          0.643728   0.465926   1.382 0.167092
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13547  on 9999  degrees of freedom
## Residual deviance: 13388  on 9936  degrees of freedom
## AIC: 13516
##
## Number of Fisher Scoring iterations: 11
```

```
#Model 2
m2_c <- glmer(data = voter_sample, family = binomial,
              voted ~ (1|COUNTY))
display(m2_c)
```

```
## glmer(formula = voted ~ (1 | COUNTY), data = voter_sample, family = binomial)
## coef.est  coef.se
##    -0.26     0.05
##
## Error terms:
##  Groups   Name        Std.Dev.
##  COUNTY   (Intercept) 0.22
##  Residual             1.00
## ---
## number of obs: 10000, groups: COUNTY, 64
## AIC = 13515.3, DIC = 13387.5
## deviance = 13449.4
```

```
#Model 3
m3_c <- glm(data = voter_sample, family = "binomial",
            voted ~ COUNTY + PCT_WHITE + PCT_URBAN)
summary(m3_c)
```

```
##
## Call:
## glm(formula = voted ~ COUNTY + PCT_WHITE + PCT_URBAN, family = "binomial",
##     data = voter_sample)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.229  -1.047  -0.924   1.294   1.810
##
## Coefficients: (2 not defined because of singularities)
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.538367   0.077306  -6.964 3.31e-12 ***
## COUNTYAlamosa       0.250685   0.447684   0.560 0.575506
## COUNTYArapahoe      0.223225   0.099536   2.243 0.024918 *
## COUNTYArchuleta     0.848522   0.404416   2.098 0.035892 *
## COUNTYBaca          2.617809   1.063474   2.462 0.013833 *
## COUNTYBent          0.538367   0.637163   0.845 0.398142
## COUNTYBoulder       0.267655   0.107826   2.482 0.013054 *
## COUNTYBroomfield    0.311594   0.196598   1.585 0.112982
## COUNTYChaffee       0.856821   0.337536   2.538 0.011134 *
## COUNTYCheyenne    -12.027696 229.628483  -0.052 0.958227
## COUNTYClear Creek   0.538367   0.399778   1.347 0.178088
## COUNTYConejos       1.326825   0.544872   2.435 0.014887 *
## COUNTYCostilla      1.049193   0.734377   1.429 0.153095
## COUNTYCrowley      -0.560245   1.157285  -0.484 0.628313
## COUNTYCuster        0.826049   0.767665   1.076 0.281903
## COUNTYDelta         0.503276   0.275995   1.823 0.068228 .
## COUNTYDenver       -0.091926   0.097463  -0.943 0.345587
## COUNTYDolores       0.132902   0.916138   0.145 0.884657
## COUNTYDouglas       0.296918   0.114660   2.590 0.009610 **
## COUNTYEagle         0.244128   0.227815   1.072 0.283898
## COUNTYEl Paso       0.054754   0.097942   0.559 0.576130
## COUNTYElbert        0.814621   0.293110   2.779 0.005449 **
## COUNTYFremont       0.106234   0.224827   0.473 0.636559
## COUNTYGarfield      0.484300   0.245091   1.976 0.048155 *
## COUNTYGilpin       -0.067768   0.513373  -0.132 0.894979
## COUNTYGrand         0.602906   0.367618   1.640 0.100998
## COUNTYGunnison      0.447396   0.311566   1.436 0.151015
## COUNTYHinsdale     13.104430 324.743705   0.040 0.967812
## COUNTYHuerfano      1.837650   0.655911   2.802 0.005084 **
## COUNTYJackson      -0.154780   1.227182  -0.126 0.899632
## COUNTYJefferson     0.340783   0.096980   3.514 0.000441 ***
## COUNTYKiowa       -12.027696 324.743705  -0.037 0.970455
## COUNTYKit Carson    0.132902   0.462936   0.287 0.774047
## COUNTYLa Plata     -0.038520   0.209917  -0.184 0.854405
## COUNTYLake          0.538367   0.582503   0.924 0.355366
## COUNTYLarimer       0.333336   0.108592   3.070 0.002143 **
## COUNTYLas Animas    0.473829   0.367618   1.289 0.197427
## COUNTYLincoln       1.924662   0.794340   2.423 0.015394 *
## COUNTYLogan         0.612475   0.392846   1.559 0.118979
## COUNTYMesa          0.077192   0.136802   0.564 0.572579
## COUNTYMineral      13.104430 324.743705   0.040 0.967812
## COUNTYMoffat        0.276003   0.427667   0.645 0.518688
## COUNTYMontezuma     0.270103   0.271753   0.994 0.320257
## COUNTYMontrose      0.618410   0.243711   2.537 0.011166 *
## COUNTYMorgan        0.371313   0.299820   1.238 0.215548
## COUNTYOtero         0.395267   0.386738   1.022 0.306756
## COUNTYOuray         2.936263   1.047323   2.804 0.005054 **
## COUNTYPark          0.236087   0.329056   0.717 0.473087
## COUNTYPhillips     -0.154780   0.869469  -0.178 0.858710
## COUNTYPitkin       -0.359574   0.365954  -0.983 0.325821
## COUNTYProwers       0.433007   0.465926   0.929 0.352710
## COUNTYPueblo        0.092461   0.146961   0.629 0.529248
## COUNTYRio Blanco    0.761511   0.480600   1.585 0.113080
## COUNTYRio Grande    0.856821   0.471047   1.819 0.068916 .
## COUNTYRoutt         0.093682   0.267614   0.350 0.726292
## COUNTYSaguache      0.356046   0.610445   0.583 0.559721
## COUNTYSan Juan      0.538367   1.002984   0.537 0.591429
## COUNTYSan Miguel   -0.640288   0.576974  -1.110 0.267114
## COUNTYSedgwick     13.104430 324.743705   0.040 0.967812
## COUNTYSummit       -0.883018   0.307646  -2.870 0.004102 **
## COUNTYTeller       -0.130682   0.279259  -0.468 0.639812
## COUNTYWashington    0.315224   0.675260   0.467 0.640630
## COUNTYWeld          0.004727   0.127066   0.037 0.970324
## COUNTYYuma          0.643728   0.465926   1.382 0.167092
## PCT_WHITE                 NA         NA      NA       NA
## PCT_URBAN                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13547  on 9999  degrees of freedom
## Residual deviance: 13388  on 9936  degrees of freedom
## AIC: 13516
##
## Number of Fisher Scoring iterations: 11
```

```
#Model 4
m4_c <- glmer(data = voter_sample, family = binomial,
              voted ~ (1|COUNTY) + PCT_WHITE + PCT_URBAN)

display(m4_c)
```

```
## glmer(formula = voted ~ (1 | COUNTY) + PCT_WHITE + PCT_URBAN,
##     data = voter_sample, family = binomial)
##             coef.est coef.se
## (Intercept)  0.19     0.35
## PCT_WHITE   -0.15     0.39
## PCT_URBAN   -0.52     0.15
##
## Error terms:
##  Groups   Name        Std.Dev.
##  COUNTY   (Intercept) 0.18
##  Residual             1.00
## ---
## number of obs: 10000, groups: COUNTY, 64
## AIC = 13506.3, DIC = 13400.9
## deviance = 13449.6
```

Individual Level

Assuming that I know the voter ID of the individual that cast their ballot, I can treat this piece of information in about the same way that I did for county as described above. This means that the following is mostly an exercise in maintaining notation constant. For these purposes, let n^{ID} be the number of total unique voter IDs–individuals–that I have data on, and j an indice that sums over all individuals. Also let z_j be an indicator variable for each individual. Then:

$$p\_\hat{v}ote_i \sim \text{logit}^{-1}(\sum_{j=1}^{n^{ID}} z_j \beta_j)$$

And the second model, not assuming independence, would be:

$$p\_\hat{v}ote \sim \text{logit}^{-1}(\delta_{j[i]}),$$
$$\delta_j \sim \text{N}(\zeta_0, \sigma_\delta^2)$$

Again, in a similar way to county level data, there are variables at an individual level, thus making it relativelly easy to build further models. Let's say now that along with the one ballot, I was given a short list of $n^{\text{indiv vars}}$ other individual-level variables, be they discrete, continuous, or indicators. The two models would then look like:

$$p\_\hat{v}ote_i \sim \text{logit}^{-1}(\sum_{j=1}^{n^{ID}} z_j \beta_j + \sum_{i'=1}^{n^{\text{indiv vars}}} z_{j[i],i'} \beta_{i'+n^{ID}})$$

Where $z_{j[i],l}$ is the j-th value of the i'-th variable. If, as before, I do not assume independance, the model can be written as:

$$p\_\hat{v}ote \sim \text{logit}^{-1}(\delta_{j[i]}),$$

$$\delta_j \sim \text{N}(\zeta_0 + \sum_{i'=1}^{n^{\text{indiv vars}}} z_{j[i],i'} \delta_{i'}, \sigma_\delta^2)$$

In the case of my specific data, for the time being I have individual-level data for gender, so $n^{\text{indiv vars}} = 1$.

```
#Model 1
#m1_i <- glm(data = voter_sample, family = "binomial", voted ~ VOTER_ID)
#Does not work

#Model 2
m2_i <- glmer(data = voter_sample, family = "binomial",
            voted ~ (1|VOTER_ID))
display(m2_i)
```

```
## glmer(formula = voted ~ (1 | VOTER_ID), data = voter_sample,
##     family = "binomial")
## coef.est  coef.se
##    -0.36     0.02
##
## Error terms:
##  Groups   Name        Std.Dev.
##  VOTER_ID (Intercept) 0.06
##  Residual             1.00
## ---
## number of obs: 10000, groups: VOTER_ID, 9991
## AIC = 13551.3, DIC = 13510.1
## deviance = 13528.7
```

```
#Model 4
m4_i <- glmer(data = voter_sample, family = "binomial",
            voted ~ (1|VOTER_ID) + GENDER)
display(m4_i)
```

```
## glmer(formula = voted ~ (1 | VOTER_ID) + GENDER, data = voter_sample,
##     family = "binomial")
##             coef.est coef.se
## (Intercept) -0.30     0.03
## GENDERTRUE  -0.13     0.04
##
## Error terms:
##  Groups   Name        Std.Dev.
##  VOTER_ID (Intercept) 0.06
##  Residual             1.00
## ---
## number of obs: 10000, groups: VOTER_ID, 9991
## AIC = 13542.8, DIC = 13497.6
## deviance = 13517.2
```

```r
#It's hard to estimate models because
length(unique(voter_sample$VOTER_ID))
```

```
## [1] 9991
```

Election Level

Again as previously, four models come from including election level data. The first two are assuming I only knew what specific election the ballot comes from. Let $w_{i'}$ be an indicator variable for each election and n^{elect} the number of elections. The model assuming independence, with $w_{i'}$ being indicator variables for each election, is:

$$p\_\hat{v}ote_i \sim \text{logit}^{-1}(\sum_{l=1}^{n^{elect}} w_l\beta_l)$$

Again, as previously, it would be safe to assume that each election is not held in a vacuum. Adding mixed effects this model would be:

$$p\_\hat{v}ote \sim \text{logit}^{-1}(\eta_{l[i]}),$$
$$\eta_l \sim \text{N}(\nu_0, \sigma^2_\nu)$$

Again, in a similar way to county and individual level data, I add in variables at an election level. Let's say now that along with the one ballot, I was given a short list of $n^{\text{election vars}}$ other election-level variables, be they discrete, continuous, or indicators. The two models would then look like:

$$p\_\hat{v}ote_i \sim \text{logit}^{-1}(\sum_{l=1}^{n^{elect}} w_l\beta_l + \sum_{i'=1}^{n^{\text{election vars}}} w_{l[i],i'}\beta_{i'+n^{elect}})$$

Where $w_{l[i],i'}$ is the l-th value of the i'-th variable.

Assuming independence:

$$p\_\hat{v}ote \sim \text{logit}^{-1}(\eta_{l[i]}),$$
$$\eta_l \sim \text{N}(\nu_0 + \sum_{i'=1}^{n^{\text{election vars}}} w_{l[i],i'}\nu_{i'}, \sigma^2_\nu)$$

For the time being I have two different variables that describe individual elections: date and type. Note that the above models may not be the best way to describe dates! An alternative could be fitting a glm, with some smoothing spline function for year. As for type, this would include four distinct indicators; one for each election type.

```r
#Model 3
m3_l <- glm(data = voter_sample, family = "binomial",
            voted ~ ELECTION_TYPE + ELECTION_DATE)
summary(m3_l)
```

```
## 
## Call:
## glm(formula = voted ~ ELECTION_TYPE + ELECTION_DATE, family = "binomial", 
##     data = voter_sample)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.6179  -0.9301  -0.6397   1.0255   1.8651  
## 
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error z value Pr(>|z|)    
## (Intercept)           -0.61408    0.06713  -9.148  < 2e-16 ***
## ELECTION_TYPEGeneral   1.80809    0.12658  14.284  < 2e-16 ***
## ELECTION_TYPEMidterm   0.98233    0.09221  10.653  < 2e-16 ***
## ELECTION_TYPEPrimary  -0.66840    0.09845  -6.789 1.13e-11 ***
## ELECTION_DATE2011      0.03345    0.09887   0.338   0.7351    
## ELECTION_DATE2012     -0.26373    0.10349  -2.548   0.0108 *  
## ELECTION_DATE2013      0.06255    0.09583   0.653   0.5139    
## ELECTION_DATE2014     -0.13190    0.07488  -1.762   0.0781 .  
## ELECTION_DATE2015           NA         NA      NA       NA    
## ELECTION_DATE2016     -0.20013    0.10159  -1.970   0.0488 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 13547  on 9999  degrees of freedom
## Residual deviance: 11763  on 9991  degrees of freedom
## AIC: 11781
## 
## Number of Fisher Scoring iterations: 4
```

```
#Model 4
m4_l <- glmer(data = voter_sample, family = "binomial",
            voted ~ (1|ELECTION) + ELECTION_DATE + ELECTION_TYPE)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```
display(m4_l)
```

```
## glmer(formula = voted ~ (1 | ELECTION) + ELECTION_DATE + ELECTION_TYPE, 
##     data = voter_sample, family = "binomial")
##                       coef.est coef.se
## (Intercept)           -1.28     0.07  
## ELECTION_DATE2011      0.70     0.10  
## ELECTION_DATE2012     -0.26     0.10  
## ELECTION_DATE2013      0.73     0.10  
## ELECTION_DATE2014     -0.13     0.07  
## ELECTION_DATE2015      0.67     0.10  
## ELECTION_DATE2016     -0.20     0.10  
## ELECTION_TYPEGeneral   2.48     0.08  
## ELECTION_TYPEMidterm   1.65     0.08  
## 
## Error terms:
##  Groups   Name        Std.Dev.
##  ELECTION (Intercept) 0.00
##  Residual             1.00
## ---
## number of obs: 10000, groups: ELECTION, 11
## AIC = 11782.6, DIC = 11762.6
## deviance = 11762.6
```

Ballot Level

In this section I assume that the ballot has some key features written on it, like the voting method, age, or party registration of the person that filled it out. A mixed effects model here would make no sense, since all the data is at the same unit of observation. Therefore, when adding ballot level variables, the model would look like:

$$p\_\hat{v}ote_i \sim \text{logit}^{-1}(\beta_0 + \sum_{i'=1}^{n^{\text{ballot vars}}} u_{i,i'}\beta_{l'})$$

Where $u_{i,i'}$ is the i-th value of the i'-th variable, and $n^{\text{ballot vars}}$ is the number of ballot level variables. For now, I have data on voting method, age, and party. Voting method is coded as a binary variable with value one if the method was a Mail Vote. Party includes four distinct indicators for REP, DEM, Other, and Unaffiliated. Age is tricky; for now the options would be: straight up inclusion as an integer, inclusion as a cubic polynomial, unclusion as a 2nd degree polynomial, inclusion in some form of spline function.

```r
#The model fit here is:
m_ballot_level <- glm(data = voter_sample, family = "binomial",
                      voted ~ GENDER + PARTY + AGE)
summary(m_ballot_level)
```

```
##
## Call:
## glm(formula = voted ~ GENDER + PARTY + AGE, family = "binomial",
##     data = voter_sample)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9905  -0.9622  -0.6774   1.1139   2.0118
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.788106   0.077207 -23.160  < 2e-16 ***
## GENDERTRUE  -0.115666   0.043408  -2.665  0.00771 **
## PARTYNO_DATA -0.426533   0.080383  -5.306 1.12e-07 ***
## PARTYOther  -0.153208   0.195647  -0.783  0.43358
## PARTYREP     0.230929   0.055638   4.151 3.32e-05 ***
## PARTYUAF    -0.594397   0.056714 -10.481  < 2e-16 ***
## AGE          0.034241   0.001312  26.095  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13535  on 9990  degrees of freedom
## Residual deviance: 12380  on 9984  degrees of freedom
##   (9 observations deleted due to missingness)
## AIC: 12394
##
## Number of Fisher Scoring iterations: 4
```

Estimation with two types of data

After the work of setting up the four models at four different levels of observation, combining them in twos should be failry straightforward. To avoid being nedlessly cumulative, I will pursue this combination for County and Individual level only–instead of the six different possible combinations.

With the assumption that both counties and individuals are independent of one another, I proceed to the first type of model:

$$p\_\hat{v}ote_i \sim \text{logit}^{-1}(\sum_{k=1}^{64} x_k\beta_k + \sum_{i'=1}^{n^{\text{county vars}}} x_{k[i],i'}\beta_{i'+64} + \sum_{j=1}^{n^{ID}} z_j\beta_{j+n^{\text{county vars}}+64} + \sum_{i'=1}^{n^{\text{indiv vars}}} z_{j[i],i'}\beta_{i'+n^{ID}+n^{\text{county vars}}+64})$$

This is large and clunky. It includes variables as described above: indicators for each county and individual, and all individual or county-level variables. For the corresponding mixed-effects model, I assume the tree-like structure we discussed on Monday. The hierarchy has two "levels", with the second level consisting of two different regressions:

$$\begin{aligned} p\_\hat{v}ote &\sim \text{logit}^{-1}(\delta_{j[i]} + a_{k[i]}), \\ a_k &\sim \text{N}(\gamma_0 + \sum_{i'=1}^{n^{\text{county vars}}} x_{k[i],i'}\gamma_{i'}, \sigma_\alpha^2) \\ \delta_j &\sim \text{N}(\zeta_0 + \sum_{i'=1}^{n^{\text{indiv vars}}} z_{j[i],i'}\delta_{i'}, \sigma_\delta^2) \end{aligned}$$

Estimation with the full dataset

I now proceed to include variables from all units of observation into one model. The first model, assuming independence, is:

$$\begin{aligned} p\_\hat{v}ote_i \sim \text{logit}^{-1}(&\sum_{k=1}^{64} x_k\beta_* + \sum_{i'=1}^{n^{\text{county vars}}} x_{k[i],i'}\beta_* + \sum_{j=1}^{n^{ID}} z_j\beta_* + \sum_{i'=1}^{n^{\text{indiv vars}}} z_{j[i],i'}\beta_* + \\ &\sum_{l=1}^{n^{elect}} w_l\beta_* + \sum_{i'=1}^{n^{\text{election vars}}} w_{l[i],i'}\beta_* + \sum_{i'=1}^{n^{\text{ballot vars}}} u_{i,i'}\beta_*) \end{aligned}$$

You will notice that I have ommited the subscript for all beta coefficients. This is because after two or three parameters, this becomes very, very large. I think it's reasonable to assume increasing indexes for different beta coefficients from left to right in this expression.

The mixed effects model will again operate on two "levels" of hierarchy, but the second level will now include three distinct regressions. Caveats for variables like age and date should be noted from previous sections.

$$p\_\hat{v}ote \sim \text{logit}^{-1}(\sum_{i'=1}^{n^{\text{ballot vars}}} u_{i,i'}\beta_{l'} + \delta_{j[i]} + a_{k[i]} + \eta_{l[i]}),$$

$$a_k \sim \text{N}(\gamma_0 + \sum_{i'=1}^{n^{\text{county vars}}} x_{k[i],i'}\gamma_{i'}, \sigma_\alpha^2)$$

$$\delta_j \sim \text{N}(\zeta_0 + \sum_{i'=1}^{n^{\text{indiv vars}}} z_{j[i],i'}\delta_{i'}, \sigma_\delta^2)$$

$$\eta_l \sim \text{N}(\nu_0 + \sum_{i'=1}^{n^{\text{election vars}}} w_{l[i],i'}\nu_{i'}, \sigma_\nu^2)$$

```
#Whole data models
#Model 1


#Model 2
#Issues with model matrix!
m2_full <- glmer(data = voter_sample, family = "binomial",
                 voted ~ (1|COUNTY) + (1|ELECTION) + GENDER + AGE +
                   PARTY + PCT_WHITE + PCT_URBAN + MAIL_VOTE)
display(m2)
```