Lesson 11

Clark

Today we're going to talk about two types of models, one that I don't think gets used a lot, and another that *should* be used a lot, but most people don't know about.

The first question we're going to analyze is whether we can determine an individual's political party (Democrat, Republican, or Independent) based on age, education, and income. (Before anyone jumps down my throat here, this is from 1996, so take all of these findings with a grain of salt. . .)

Our response variable is **political party**. Let's consider our response variable y_i, what are some values that it can take on? How would we represent it in a statistical model?

If we take a sample from the population, the number of each political party that we observe would follow a **multinomial distribution**.

If we only had two political parties, let's see what would happen to the multinomial.

So, essentially we are generalizing the binomial distribution. Now, let's say we want to analyze the probabilities, that is, we want to put structure on the probability that someone from our sample came from each of the parties. Note that if we weren't interested in this we could easily come up with the π values through descriptive analytics.

```
library(faraway)
library(tidyverse)
pol_dat <- read.csv("PoliticalPartyData.csv")

pol_dat %>%
  group_by(PolParty) %>%
  summarize(pi_hat = n()/nrow(pol_dat))
```

```
## # A tibble: 3 x 2
```

```
##   PolParty    pi_hat
##   <chr>        <dbl>
## 1 Democrat     0.337
## 2 Independent  0.412
## 3 Republican   0.251
```

However, since we're statisticians, we think we can likely do better than this. In fact, what we just found is the null model for a multinomial logit model.

This model is sometimes called the baseline-category logit model. Here, we assume that our data follow a multinomial probability distribution. It seems like to do this we could just write out (ignoring the constants)

However, we have an issue, even though it looks like I have three π values, I actually only have two. Why?

Now, really, we're sort of in the same world we were when we developed out the Binomial exponential dispersion family. To account for this, we can do:

So, our natural parameter (which is now a vector) that we will put structure on is:

This whole notion of choosing a baseline category may seem rather arbitrary. But really, it's no less arbitrary than in logistic regression when we choose one of our outcomes to be a `success'' and another to be a`failure''.

So, our entire model is:

Which we can fit in R

```
library(nnet)
mmod <- multinom(PolParty ~ Race + Gender +Age + HighestDegree,
                 data=pol_dat)
```

```
## # weights:  30 (18 variable)
## initial  value 26736.927269
## iter  10 value 25229.451478
## iter  20 value 24802.360685
## final  value 24684.672407
## converged
```

```
summary(mmod)
```

```
## Call:
## multinom(formula = PolParty ~ Race + Gender + Age + HighestDegree,
##     data = pol_dat)
##
## Coefficients:
##             (Intercept) RaceOther RaceWhite GenderMale          Age
## Independent  -0.3025338  1.082644  1.254419  0.3230857 -0.016731468
## Republican   -2.4153567  1.592685  2.635991  0.3374939 -0.001706418
##             HighestDegreeGraduate HighestDegreeHighSchool HighestDegreeJrCol
## Independent            -0.2401794               0.1898370          0.1674902
## Republican             -0.5701902              -0.1521903         -0.1013558
##             HighestDegreeSomeHS
## Independent           0.3579976
## Republican           -0.5981805
##
## Std. Errors:
##             (Intercept)  RaceOther  RaceWhite GenderMale          Age
## Independent  0.06562289 0.06116588 0.04251792 0.03114656 0.0009145936
## Republican   0.09974459 0.10691773 0.08244307 0.03574855 0.0010280506
##             HighestDegreeGraduate HighestDegreeHighSchool HighestDegreeJrCol
## Independent            0.06079600              0.04337008         0.06690654
## Republican             0.06676265              0.04662257         0.07422047
##             HighestDegreeSomeHS
## Independent          0.05630771
## Republican           0.06954739
##
## Residual Deviance: 49369.34
## AIC: 49405.34
```

We can analyze the impact of education through a likelihood ratio test (drop in deviance test) as our model is fit via MLE and therefore our coefficients are asymptotically normal

```
mod2 <- multinom(PolParty ~ Race + Gender +Age,
                 data=pol_dat)
```

```
## # weights:  18 (10 variable)
## initial  value 26736.927269
## iter  10 value 26049.276015
## final  value 24839.955030
## converged
```

```
anova(mmod,mod2,test="Chisq")
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: PolParty
##                                 Model Resid. df Resid. Dev   Test    Df
## 1                 Race + Gender + Age     48664   49679.91
## 2 Race + Gender + Age + HighestDegree     48656   49369.34 1 vs 2     8
##   LR stat. Pr(Chi)
## 1
## 2 310.5652       0
```

Now, perhaps this is a little surprising that there is a diffence of 8 degrees in freedom when we remove education since:

```
unique(pol_dat$HighestDegree)
```

```
## [1] "HighSchool" "JrCol"      "Graduate"   "SomeHS"     "Bachelor"
```

What do you think is going on?

Assessing the model is not, to me, entirely straightforward. If our data were grouped, we could compare our deviance to saturated model, but since it's not, I'm not entire sure.

We can find residuals through:

```
resids <- residuals(mmod,type="response")
```

These can then be plotted against explanatory variables or linear predictors.

The real *fun* though is interpreting the coefficients. Going back to the summary output, we have the following fitted models.

Let's just focus on Age for a second. If we examine the confidence intervals we get (using Wald interval)

```
-.01673-1.96*0.0009145936
```

```
## [1] -0.0185226
```

```
-.01673+1.96*0.0009145936
```

```
## [1] -0.0149374
```

So, comparing independents to democrats, we note that an approximate odds ratio is:

```
exp(-.01673-1.96*0.0009145936)
```

```
## [1] 0.9816479
```

```
exp(-.01673+1.96*0.0009145936)
```

```
## [1] 0.9851736
```

Meaning, for every year older the odds that you are an independent over a democrat decrease by a rate of about 98%.

We can do the same thing comparing republicans to democrats

```
-0.001706418-1.96*0.0010280506
```

```
## [1] -0.003721397
```

```
-0.001706418+1.96*0.0010280506
```

```
## [1] 0.0003085612
```

Here we see no effect as the CI contains 0.

What if we want to compare republicans to independents? Well, then we have to do some work:

So, we want the distribution of $\beta_{r,age} - \beta_{i,age}$. However, we note to find this we have:

This requires us to look at the covariance matrix.

```
#vcov(mmod)
se <- sqrt(0.0010280506^2+0.0009145936^2-2*4.749699e-07)

#Difference in betas minus new se
exp(((-0.001706418)-(-.01673))-1.96*se)
```

```
## [1] 1.013206
```

```
exp(((-0.001706418)-(-.01673))+1.96*se)
```

```
## [1] 1.017071
```

One other type of model I want to talk about that I think should be used more. In fact, Cadets deal with data like this all the time. That is, ordinal data.

If we've ever taken a survey where the possible responses are strongly disagree, disagree, neutral, agree, strongly agree, we have deal with ordinal data.

Here our data differ from what we talked about above in that there is a natural ordering of the data. In this case, we want to express the model in terms of cumulative probabilities.

Cumulative logits can be constructed by thinking about the data like every category above the one we are looking at is a failure and every category below the one we're looking at is a success.

That is:

So, for the cumulative logit model, we place structure on the cumulative logit.

Note that the underlying statistical model is still a multinomial, but we are, sort of, changing our link function. Note that each logit has its own intercept. The intercept values *must* increase, why?

```
mental_health <- read.table('https://users.stat.ufl.edu/~aa/glm/data/Mental.dat', header=TRUE)
```

Here, our statistical question is:

Our model is:

```
library(VGAM)

fit.mod <- vglm(impair~life+ses,data=mental_health,
                family=cumulative(parallel=TRUE))
```

Note there are a couple of things to address here. One is the `parallel=TRUE`. This ensures that you have the same β for each of your cut-off values. The `Hauck-Donner` effect is when your data are perfectly separated. When this occurs, you get a "perfect'' fit, so essentially your β terms are not estimable.

Here we don't have that issue.

We can write out three separate fitted models.

Additionally, for each individual in our study we can get estimated categories

```
fitted(fit.mod)%>%head()
```

```
##           1         2          3          4
## 1 0.6249170 0.2564211 0.07131461 0.04734732
## 2 0.1150217 0.2518290 0.24398492 0.38916438
## 3 0.6962144 0.2146331 0.05428144 0.03487106
## 4 0.3902833 0.3502169 0.14495617 0.11454363
## 5 0.4682286 0.3287367 0.11707622 0.08595848
## 6 0.2850362 0.3548973 0.18808559 0.17198084
```

As we are fitting our model via maximum likelihood we can test nested models via:

```
smaller.mod <- vglm(impair~life,data=mental_health,
                    family=cumulative(parallel=TRUE))
chi_stat <- -2*(logLik(smaller.mod)-logLik(fit.mod))

1-pchisq(chi_stat,1)
```

```
## [1] 0.06405392
```