POLSCI 9592

Lecture 6: Unordered Categorical Data Models

Dave Armstrong

Goals for This Session

1. Develop Multinomial Logit Model
2. Effects and Effect Displays
3. Model Fit and Evaluation
4. IIA Assumption
5. Conditional Logit
6. Separation Problems

Multinomial Logit

- The MNL model is used when the dependent variable is a set of unordered categories.
- We can think of MNL as a generalization of the Ordered Logit model (i.e., the OL model is nested in the MNL model)
- Before, we suggested that there is very little difference between logits and probits - that our inferences will likely be very similar. In the multinomial case, these are two quite different models that require different estimation techniques and make different sets of assumptions.
    - We will focus exclusively on the multinomial logit (MNL) model here.

The MNL Model

The MNL model basically works like a bunch of binary logit models. Assume our dependent variable y has $m = \{1, 2, \ldots, M\}$ different values.

- Let's say we want to consider the probability of being a 2 versus the probability of being a 1. We could do the following:
  - Subset the data to only those observations where $y = \{1, 2\}$.
  - Estimate the binary logit model: $\log\left(\frac{Pr(y=2|X)}{Pr(y=1|X)}\right) = \mathbf{Xb}$.
- The MNL model basically does this in turn for every value of y in $\{2, \ldots, M\}$.
  - Each time, estimating $\log\left(\frac{Pr(y=m|X)}{Pr(y=1|X)}\right) = \mathbf{Xb}_m$

The MNL Model

- The group against which we estimate all of the binary logits (i.e., 1 in the example above) is called the reference group.

- We can calculate probabilities of being in a single group as follows:

$$Pr(y = m|X) = \frac{exp(\mathbf{X}\beta_m)}{\sum_{j=1}^{J} exp(\mathbf{X}\beta_j)}$$

- For identification purposes, we set $\beta_1 = \mathbf{0}$. That is, all of the coefficients for the first group are set to zero. Remember, $exp(0) = 1$.

Likelihood Function

Just like in our other models, the likelihood is simply the probability that the observation takes on its observed y value. We could write this as:

$$L(\beta_1, \beta_2, \ldots, \beta_{J-1}|\mathbf{X}) = \prod_{m=1}^{J} \prod_{y_i=m} \frac{exp(\mathbf{X}\beta_m)}{\sum_{j=1}^{J} exp(\mathbf{X}\beta_j)}$$

The Log-likelihood function, then, is:

$$\log L(\beta_1, \beta_2, \ldots, \beta_{J-1}|\mathbf{X}) = \sum_{m=1}^{J} \sum_{y_i=m} \log\left(\frac{exp(\mathbf{X}\beta_m)}{\sum_{j=1}^{J} exp(\mathbf{X}\beta_j)}\right)$$

France Data

```r
dat <- import("data/france_mnl.dta")
dat <- factorize(dat)
labs <- sapply(1:ncol(dat), function(i)attr(dat[[i]], "label"))
data.frame(var = names(dat), description = labs)
```

```
##          var                                         description
## 1       vote                                       Party of vote
## 2    votenum                         Party number of chosen party
## 3       male                          R gender (0=female, 1=male)
## 4        age                                            Age of R
## 5      urban                         size of city in which R lives
## 6   soclass                               Subjective social class
## 7   hhincome                     Household income (in francs/month)
## 8      union                                      Member of a union
## 9    retnat              Retrospective national economic evaluations
## 10    demsat satisfaction with the functioning of democracy in france
## 11     eusup                      Good choice for France to belong to EU
## 12    lrself self-position on the left-right scale (0=left, 10=right)
## 13      rp10        Respondent identified left-right placement of PCF
## 14      rp20         Respondent identified left-right placement of PS
## 15      rp50      Respondent identified left-right placement of Green
## 16      rp70        Respondent identified left-right placement of UDF
## 17      rp73        Respondent identified left-right placement of UMP
## 18      rp80         Respondent identified left-right placement of FN
```

Estimation: multinom

The model can be estimated with the `multinom` function in the `nnet` package.

```
mod <- multinom(vote ~ lrself + male + retnat + age +
    union , data=dat, trace=F)
noquote(t(mnlSig(mod)))
```

```
##                          PS      Green   UDF     UMP*    FN
## (Intercept)              3.809*  3.253*  -0.996  -1.504  -1.153
## lrself                   0.227*  0.389*  0.925*  1.151*  0.832*
## male                     0.574   0.223   0.727   0.565   0.231
## retnatsame               -0.909  -1.306  -1.827  -1.514  -1.196
## retnatworse              -1.041  -1.712  -2.524* -2.635* -0.840
## age                      -0.020* -0.047* -0.004  -0.010  -0.017
## unionyes, union member   -1.325* -1.004* -1.272* -1.279* -1.485*
```

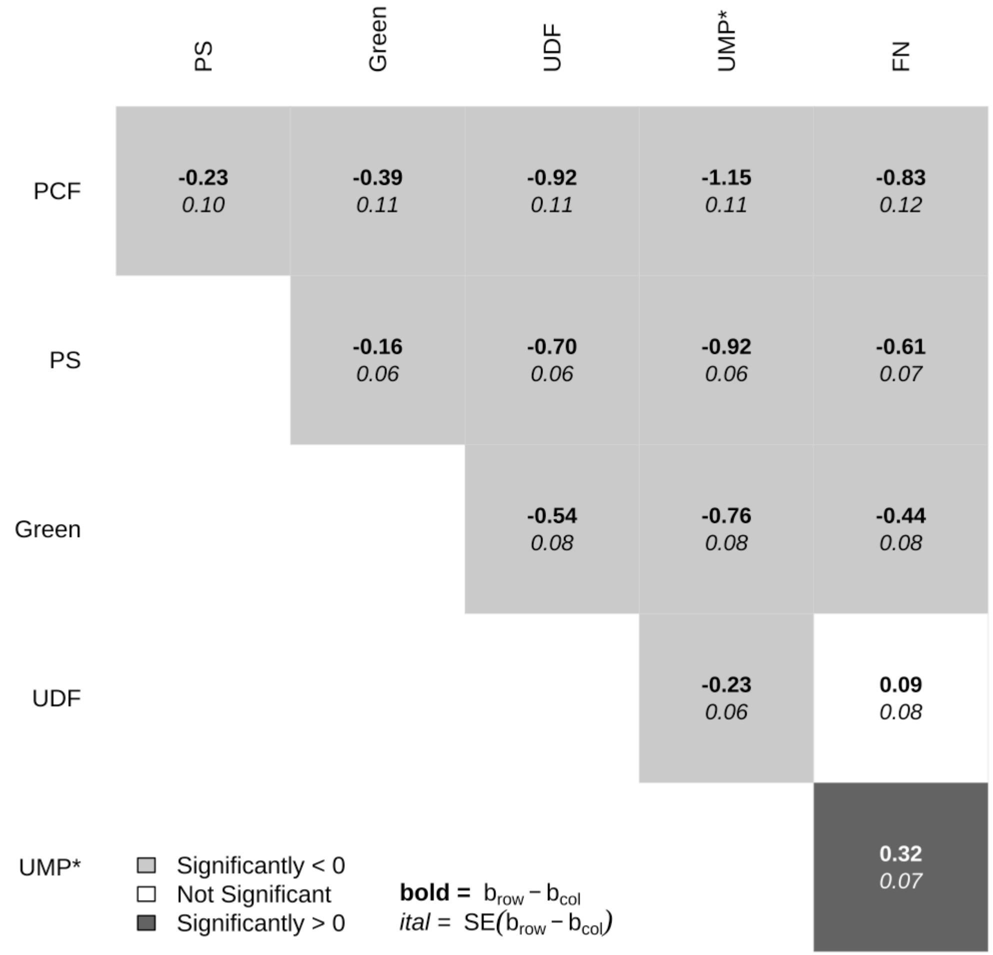
You could also use `broom::tidy(mod)` to get a tidier model output.

Interpretation: Comparing Coefficients

- Coefficients are in log-odds of voting for the party identified in the row relative to the base category (in R, the base category is always the first one).
- Note that the coefficients relating covariates to non-base-category choices must be considered separately.
- For example, the effect of left-right self-placement on the choice between Greens and Socialists is -0.174, meaning that as left-right increases, voters are more likely to vote for the Green party than the Socialists.

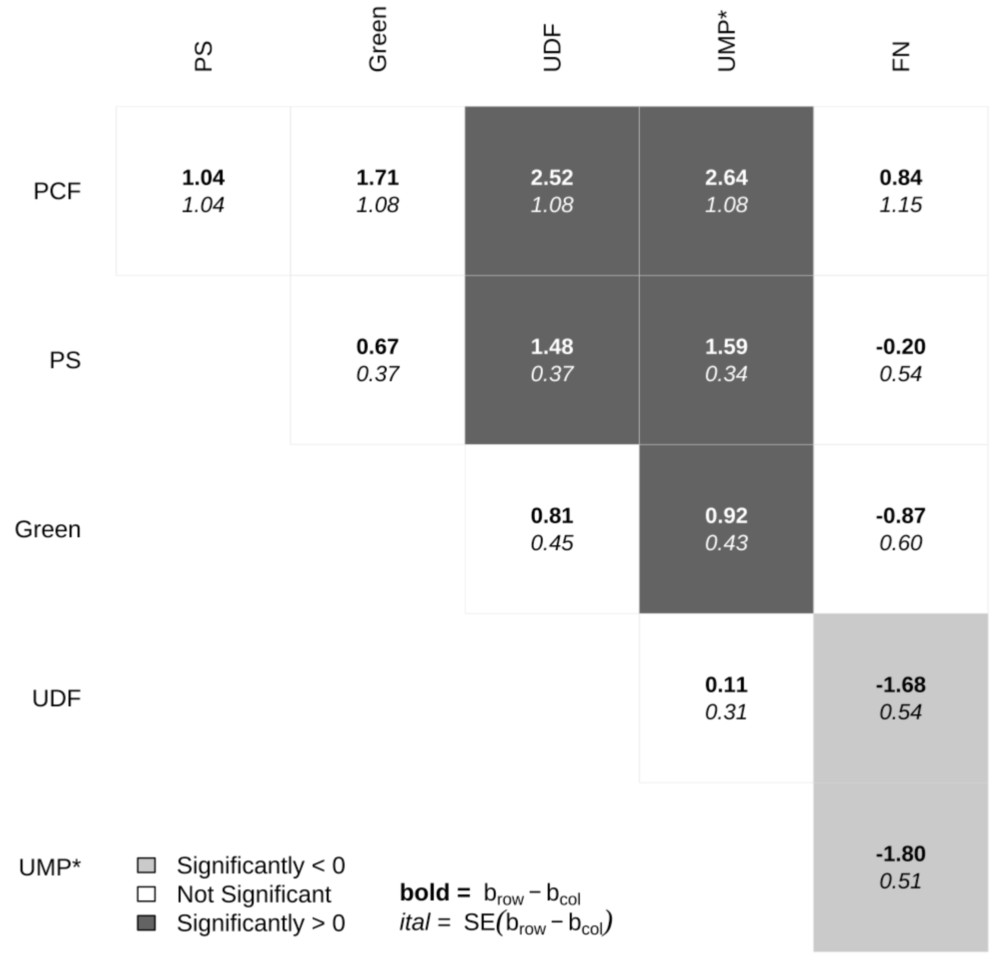
Pairwise Comparisons

```r
plot(factorplot(mod, variable="lrself"))
```



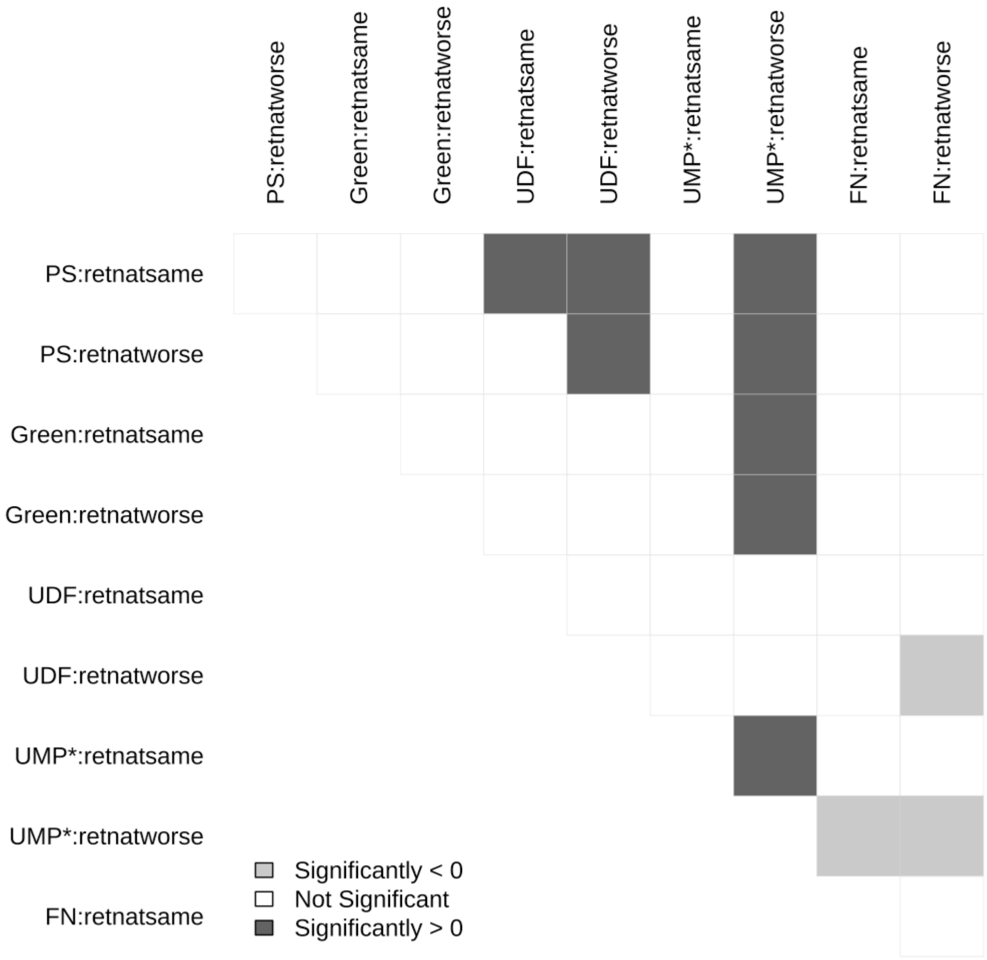
Pairwise Comparisons with Factors

```
plot(factorplot(mod,
  variable="retnatworse"))
```



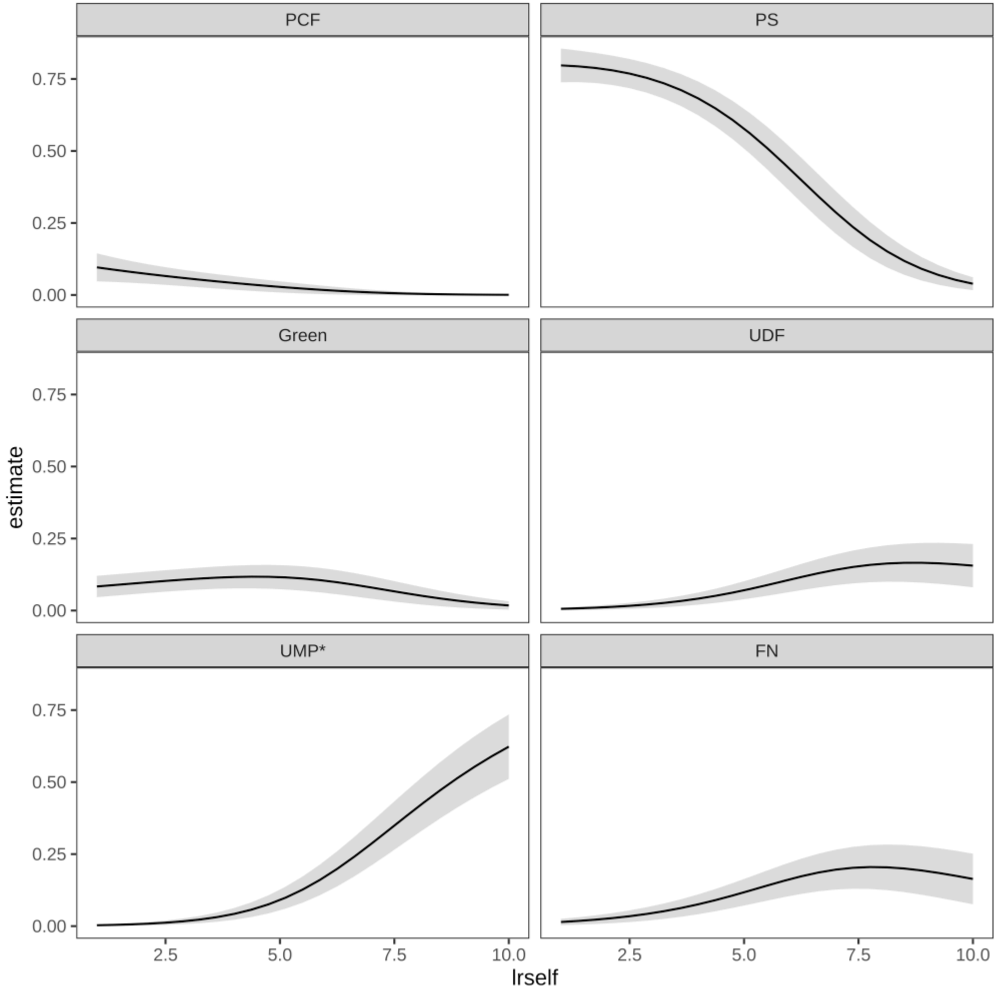
Pairwise Comparisons with Factors

```r
b <- c(t(coef(mod)))
v <- vcov(mod)
rn_inds <- grep("retnat", rownames(v))
b_rn <- b[rn_inds]
v_rn <- v[rn_inds, rn_inds]
names(b_rn) <- rownames(v_rn)
plot(factorplot(b_rn, var=v_rn, resdf=Inf),
     print.est=FALSE, print.se=FALSE,
     print.square.leg=FALSE, abbrev.char = 25)
```



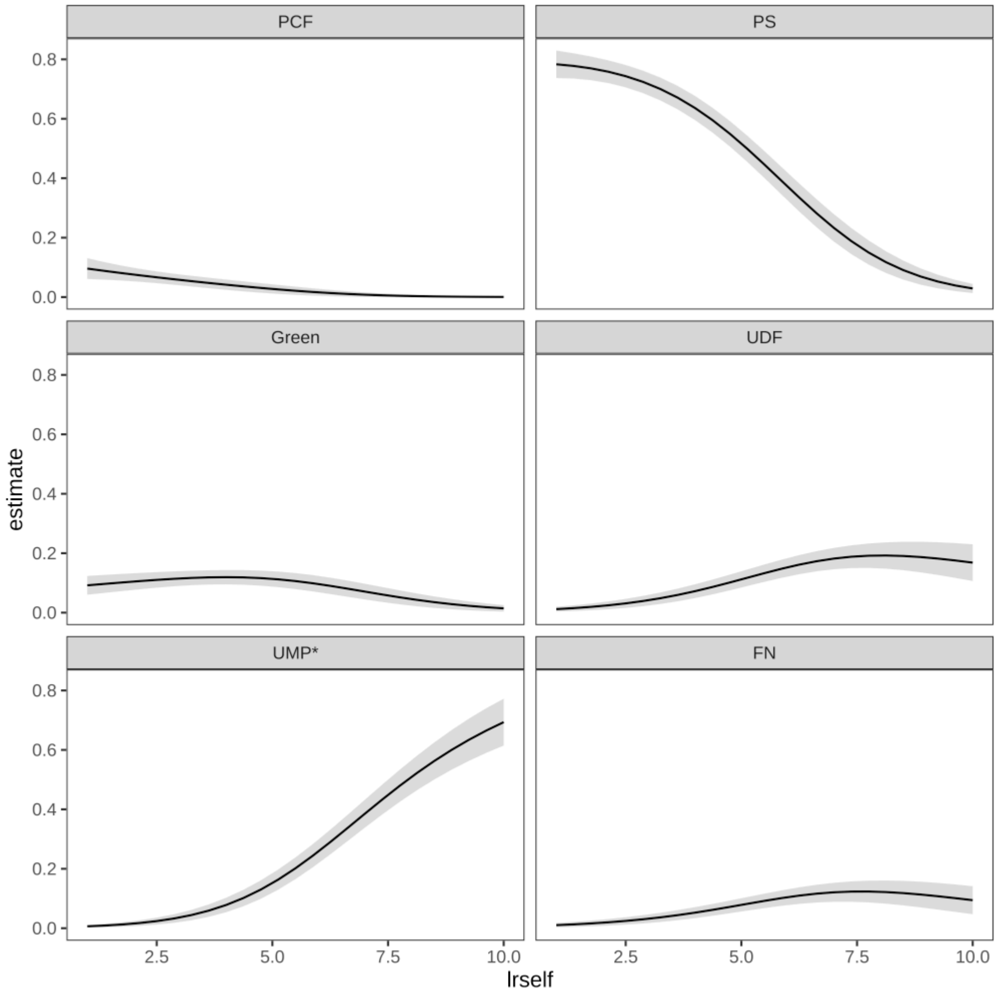
Effects Plots: MER

```r
seq_range <- function(x, n=25, ...){x <- na.omit(x); seq(min(x),
pred_lrs <- predictions(mod,
            newdata="median",
            variables=list(lrself = seq_range(dat$lrself)))

ggplot(pred_lrs,
       aes(x=lrself,
           y=estimate,
           ymin = conf.low,
           ymax=conf.high)) +
  geom_ribbon(alpha=.2, colour="transparent") +
  geom_line() +
  facet_wrap(~group, ncol=2) +
  theme_bw() +
  theme(panel.grid=element_blank())
```



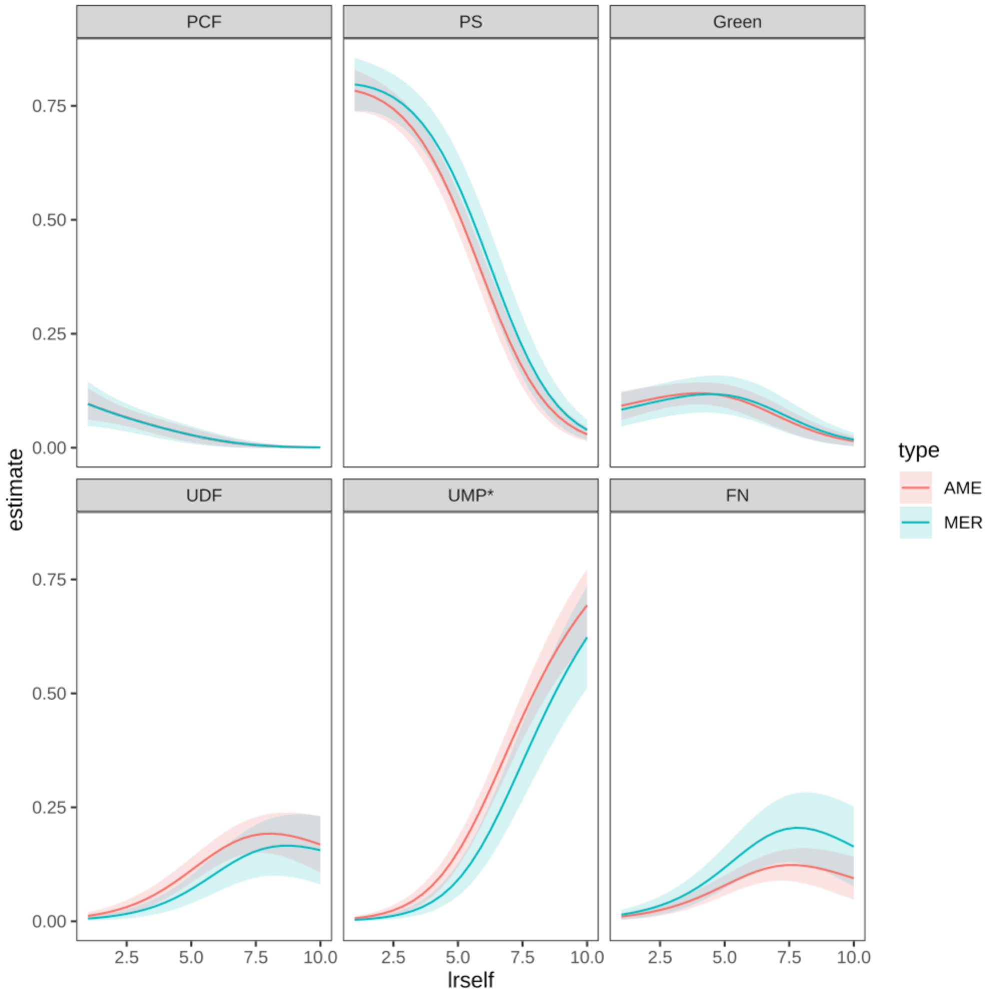
Effect Plot: AME

```
ave_pred_lrs <- avg_predictions(mod,
            variables=list(lrself = seq_range(dat$lrself)))
ggplot(ave_pred_lrs,
       aes(x=lrself,
           y=estimate,
           ymin = conf.low,
           ymax=conf.high)) +
  geom_ribbon(alpha=.2, colour="transparent") +
  geom_line() +
  facet_wrap(~group, ncol=2) +
  theme_bw() +
  theme(panel.grid=element_blank())
```



Comparing Effects

```r
all_lrs <- bind_rows(pred_lrs %>% mutate(type = "MER"),
          ave_pred_lrs %>% mutate(type = "AME"))
ggplot(all_lrs,
       aes(x=lrself,
           y=estimate,
           ymin = conf.low,
           ymax=conf.high,
           colour=type,
           fill=type)) +
  geom_ribbon(alpha=.2, colour="transparent") +
  geom_line() +
  facet_wrap(~group) +
  theme_bw() +
  theme(panel.grid=element_blank())
```



Discrete Changes: MER

```
comps <- comparisons(mod, newdata="median",
            variables = list(lrself = "2sd",
                             male = c(0,1),
                             retnat = "minmax",
                             age = "2sd",
                             union = "reference"))

comps %>%
  mutate(eff = sprintf("%.2f%s",
                       estimate,
                       ifelse(p.value < .05, "*", ""))) %>%
  select(2:4, eff) %>%
  pivot_wider(names_from="group", values_from="eff")
```

```
## # A tibble: 5 × 8
##   term   contrast                     PCF    PS     Green UDF    `UMP*` FN
##   <chr>  <chr>                        <chr>  <chr>  <chr> <chr>  <chr>  <chr>
## 1 age    (x + sd) - (x - sd)          0.02   -0.00  -0.1… 0.04*  0.03   0.01
## 2 lrself (x + sd) - (x - sd)          -0.0…  -0.5…  -0.03 0.14*  0.35*  0.18*
## 3 male   1 - 0                        -0.01  0.04   -0.03 0.02   0.01   -0.03
## 4 retnat worse - better               0.02*  0.23*  -0.02 -0.1…  -0.18* 0.06
## 5 union  yes, union member - no, not a uni… 0.07*  -0.06  0.03  -0.00  -0.01  -0.03
```

Discrete Changes: AME

```
ave_comps <- avg_comparisons(mod,
      variables = list(lrself = "2sd",
       male = c(0,1),
       retnat = "minmax",
       age = "2sd",
       union = "reference"))
ave_comps %>%
  mutate(eff = sprintf("%.2f%s",
                       estimate,
                       ifelse(p.value < .05, "*", ""))) %>%
  select(1:3, eff) %>%
  pivot_wider(names_from="group", values_from="eff")
```

```
## # A tibble: 5 × 8
##   term   contrast                          PCF   PS    Green UDF   `UMP*` FN
##   <chr>  <chr>                             <chr> <chr> <chr> <chr> <chr>  <chr>
## 1 age    mean(x + sd) - mean(x - sd)       0.03* -0.01 -0.0… 0.04* 0.02   -0.00
## 2 lrself mean(x + sd) - mean(x - sd)       -0.0… -0.6… -0.0… 0.17* 0.44*  0.11*
## 3 male   mean(1) - mean(0)                 -0.02 0.04  -0.02 0.02  0.00   -0.02
## 4 retnat mean(worse) - mean(better)        0.03* 0.13* -0.03 -0.0… -0.12* 0.05*
## 5 union  mean(yes, union member) - mean(no… 0.07* -0.07 0.02  -0.00 -0.00  -0.01
```

Model Fit

```
mnlfit(mod)
```

```
## $result
##                                Estimate    p-value
## Fagerland, Hosmer and Bonfi 57.5035680 0.03593795
## Count R2                     0.6354481         NA
## Count R2 (Adj)               0.2760000         NA
## ML R2                        0.5289860         NA
## McFadden R2                  0.2622133         NA
## McFadden R2 (Adj)            0.2334525         NA
## Cragg-Uhler(Nagelkerke) R2   0.5607412         NA
##
## attr(,"class")
## [1] "mnlfit"
```

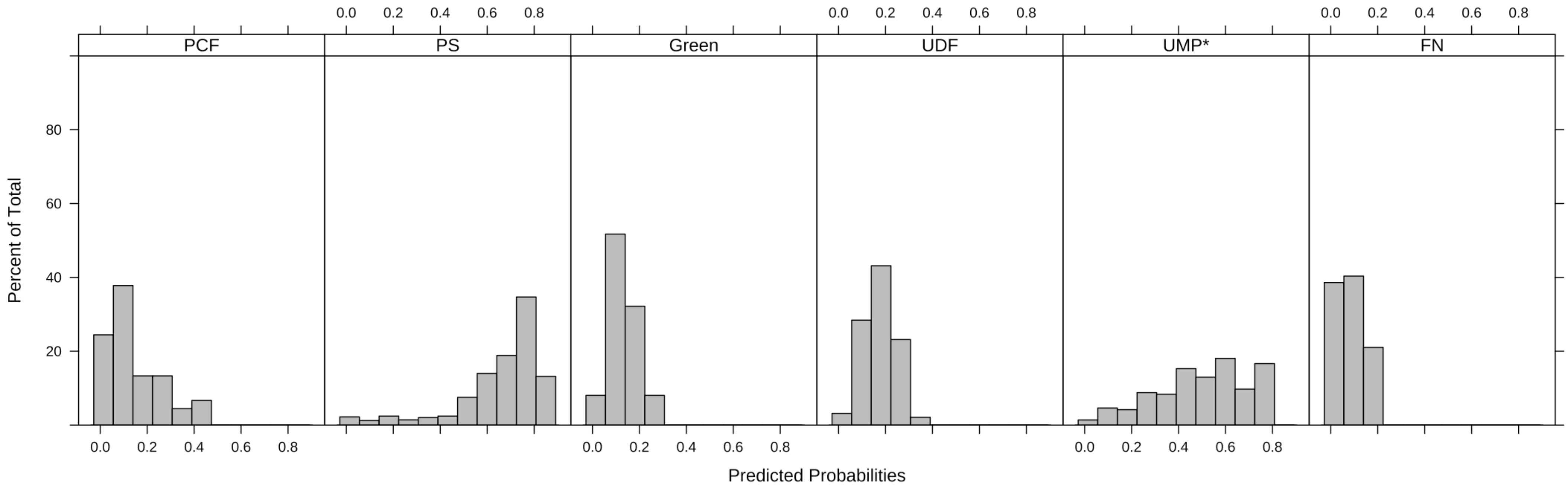
PRE

```
pre(mod)
```

```
## mod1:  vote ~ lrself + male + retnat + age + union
## mod2:  vote ~ 1
##
## Analytical Results
##  PMC =  0.496
##  PCP =  0.635
##  PRE =  0.276
## ePMC =  0.316
## ePCP =  0.478
## ePRE =  0.237
```

Probabilities by Group

```
probgroup(mod)
```



Classification Table

We could also look at the classification table where the rows are the actual votes and the columns are the predicted votes

```
y <- model.response(model.frame(mod))
yhat <- predict(mod)
table(y, yhat)
```

```
##        yhat
## y       PCF  PS Green UDF UMP*  FN
##   PCF     0  44     0   0    1   0
##   PS      0 457     0   0   36   0
##   Green   0  82     0   0    5   0
##   UDF     0  39     0   0   56   0
##   UMP*    0  42     0   0  174   0
##   FN      0  31     0   0   26   0
```

Note that we predict most people will vote for the two biggest parties - socialists and UMP and we predict a few will vote for the national front (FN).

IIA Assumption

$$\text{IIA} = \text{Independence of Irrelevant Alternatives}$$

- The addition or removal of choices from the choice set should not change the odds of choosing one category over another.
- This is forced to be the case in MNL. Multinomial probit does not make this assumption.

Tests for IIA

There are tests for IIA, but Long and Freese argue they shouldn't be used.

- They are not particularly powerful and can often give conflicting results.
- Dow and Endersby suggest that the relaxing the assumption is rarely important and that doing so comes with its own potential set of problems.
- In my own work, I have found if trying to estimated predicted probabilities, both models work about the same.

Conditional Logit

In the MNL models above, all of the variables varied by individual (i.e., they are attributes of the individuals), but not by choice within individual.

- In this case, each there is a coefficient for each choice Occasionally, there are attributes of the choices we would like to incorporate.
- In this case, Conditional Logit can be used.

The Model

Remember, before we assumed that the relationship between y and $\mathbf{X}$ was captured with β_m

$$Pr(y = m|X) = \frac{exp(\mathbf{X}\beta_m)}{\sum_{j=1}^{J} exp(\mathbf{X}\beta_j)}$$

where $\mathbf{X}\beta_m = \beta_{0,m} + \beta_{1,m}X_1 + \beta_{2,m}X_2 + \ldots + \beta_{k,m}X_k$. By introducing choice-specific information, we need to change the linear predictor to: $\mathbf{Z}_m\gamma + \mathbf{X}\beta_m$. Notice here, that there are some variables that have an m subscript and those variables only get one coefficient - that is, there is one coefficient relating variable Z_1 to $\mathbf{y}$ because the information in Z_1 varies within observations.

Data Preparation

```
names(dat)[13:18]
```

```
## [1] "rp10" "rp20" "rp50" "rp70" "rp73" "rp80"
```

```
names(dat)[13:18] <- paste("rp", levels(dat$vote), sep=".")
dat2 <- dat %>%
  dplyr::select(c("vote", "lrself", "urban", "union", "retnat",
           "age", "male", starts_with("rp"))) %>%
  na.omit()
library(fastDummies)
library(dfidx)
dat2 <- dat2 %>%
  dummy_cols("vote", remove_selected_columns=TRUE)
names(dat2) <- gsub("vote_", "vote.", names(dat2))
dat2 <- dat2 %>%
  mutate(obs = row_number()) %>%
  pivot_longer(7:18,
               names_pattern = "(.*)\\.(.*)",
               names_to = c(".value", "alt"))
dat2l <- dfidx(dat2, idx = c("obs", "alt"), choice = "vote")
```

Conditional Logit Model

```r
mlogit2 <- mlogit(vote ~ 0 + I(abs(lrself - rp)) |  urban + union + retnat +
    age + male, data=dat2l, reflevel="PS")
printCoefmat(summary(mlogit2)$CoefTable)
```

```
##                                    Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):FN                   -2.2506029  0.7393348  -3.0441 0.0023338 **
## (Intercept):Green                -0.0489038  0.5260797  -0.0930 0.9259362
## (Intercept):PCF                  -4.2232729  1.1722648  -3.6027 0.0003150 ***
## (Intercept):UDF                  -1.5483687  0.5143635  -3.0103 0.0026102 **
## (Intercept):UMP**                -0.5136178  0.4324413  -1.1877 0.2349450
## I(abs(lrself - rp))              -0.5122424  0.0284676 -17.9939 < 2.2e-16 ***
## urbansmall or medium city:FN     -0.0159390  0.3865481  -0.0412 0.9671092
## urbansmall or medium city:Green  -0.1439434  0.2796757  -0.5147 0.6067769
## urbansmall or medium city:PCF     0.0577400  0.3784491   0.1526 0.8787373
## urbansmall or medium city:UDF    -0.0467703  0.2977841  -0.1571 0.8751968
## urbansmall or medium city:UMP** -0.1612852  0.2502350  -0.6445 0.5192288
## urbanbig city:FN                  0.3215769  0.4453963   0.7220 0.4702934
## urbanbig city:Green              -0.7541499  0.3812699  -1.9780 0.0479293 *
## urbanbig city:PCF                -0.2370372  0.5087779  -0.4659 0.6412905
## urbanbig city:UDF                 0.0590352  0.3461118   0.1706 0.8645643
## urbanbig city:UMP**               0.2174351  0.2829513   0.7685 0.4422174
## unionyes, union member:FN         0.0342450  0.5477225   0.0625 0.9501467
## unionyes, union member:Green      0.2708221  0.3472815   0.7798 0.4354883
## unionyes, union member:PCF        1.2122812  0.3615435   3.3531 0.0007992 ***
## unionyes, union member:UDF       -0.0684448  0.4150341  -0.1649 0.8690119
## unionyes, union member:UMP**     -0.0514542  0.3432224  -0.1499 0.8808316
## retnatsame:FN                     0.1536325  0.6381573   0.2407 0.8097536
## retnatsame:Green                 -0.5581574  0.4260458  -1.3101 0.1901661
## retnatsame:PCF                    0.9195162  1.1049479   0.8322 0.4053070
## retnatsame:UDF                   -0.6974712  0.3764854  -1.8526 0.0639419 .
```

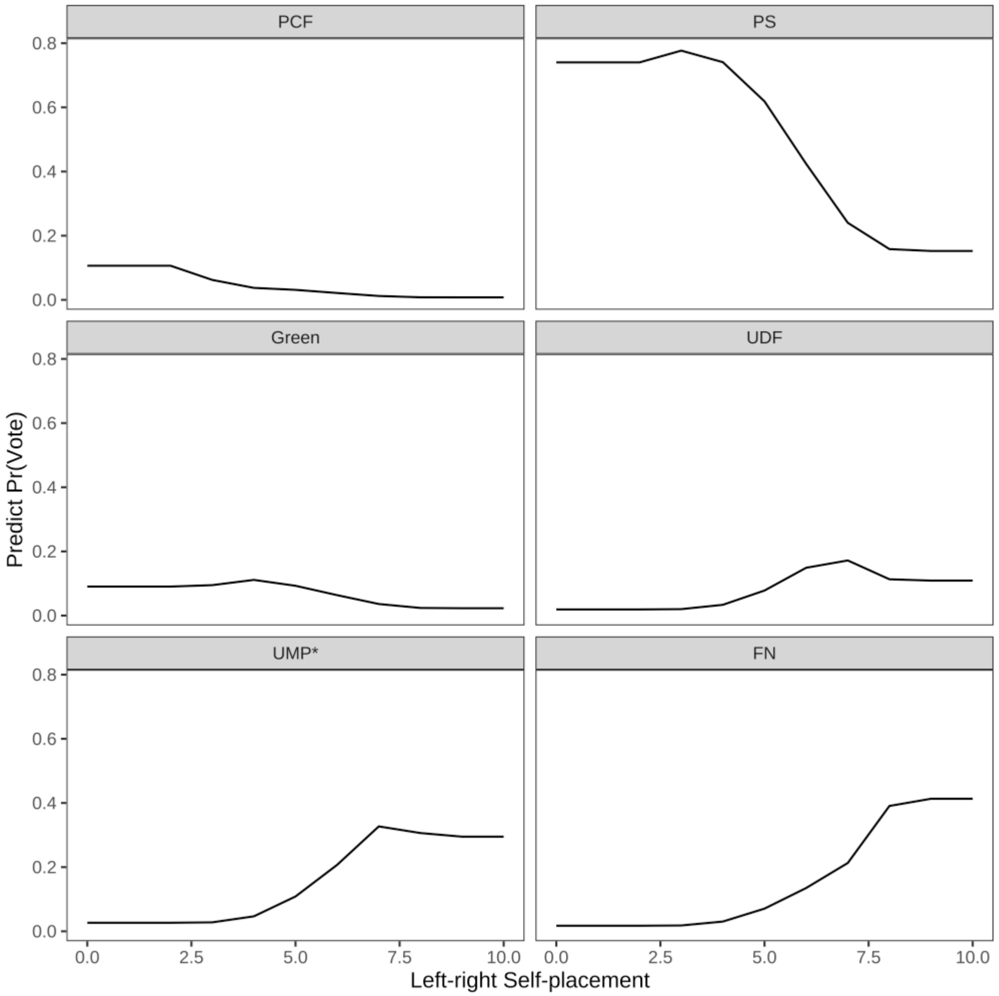
Effect Plots

For conditional logit, there are no canned functions to make effects plots. Here are the steps you have to apply:

1. Make the data used to generate predictions (either one fake dataset holding all variables at some hypothetical value or a bunch of datasets each with the variable of interest held constant.)
2. Generate predictions using each dataset.
3. If confidence intervals are desired, use simulation to calculate sampling variability of the predicted probabilities.
4. Plot predictions

MER approach

```r
lrs <- 0:10
fake <- makeFakeData(mlogit2,
                     dat2l,
                     change=list(lrself=lrs),
                     varying="rp")
probs <- predict(mlogit2, newdata=fake)
plot.dat <- data.frame(
  prob = c(t(probs)),
  party = factor(rep(colnames(probs),
                     length(lrs)),
                 levels=levels(dat$vote)),
  lrself = rep(lrs, each=6))
ggplot(plot.dat, aes(x=lrself, y=prob)) +
  geom_line() +
  facet_wrap(~party, ncol=2) +
  theme_bw() +
  theme(panel.grid=element_blank()) +
  labs(x="Left-right Self-placement",
       y="Predict Pr(Vote)")
```

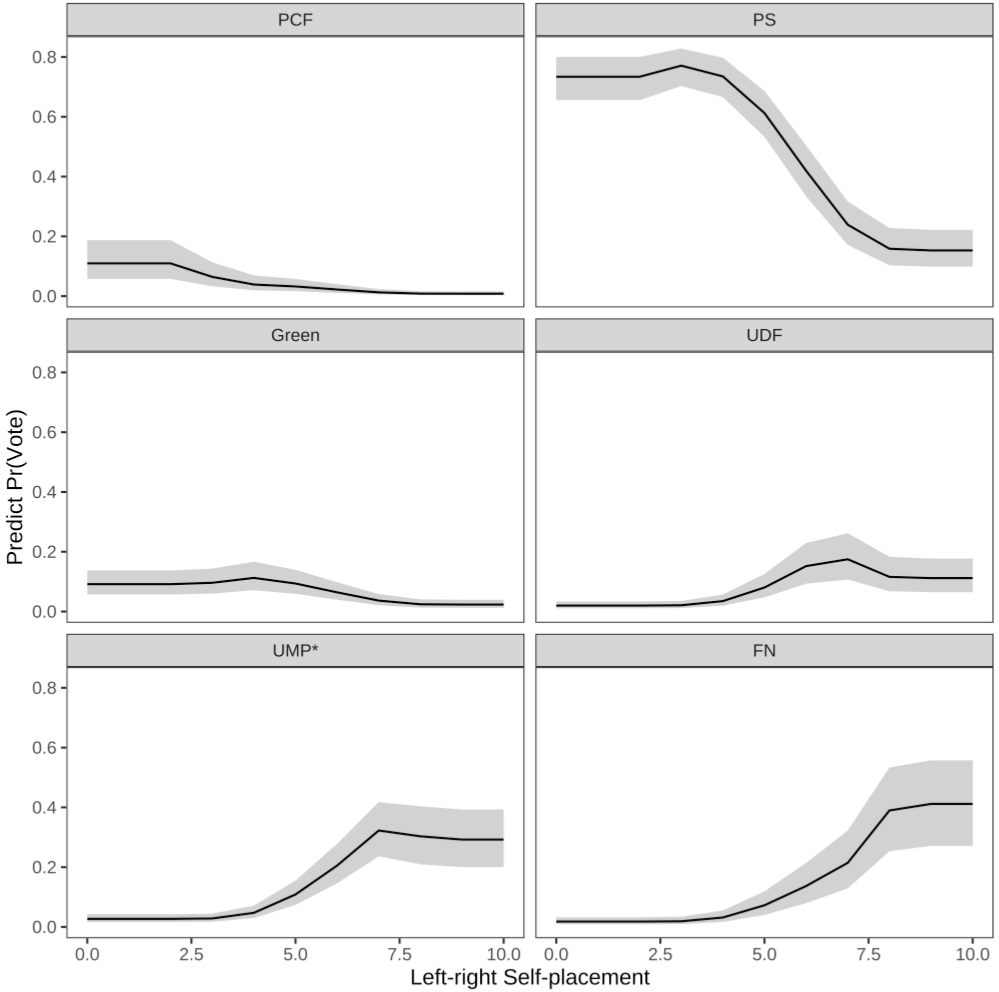


Confidence Intervals

```
B <- MASS::mvrnorm(2500, coef(mlogit2), vcov(mlogit2))
tmp <- mlogit2
probs <- NULL
for(i in 1:2500){
  tmp$coefficients <- B[i,]
  probs <- rbind(probs,
   cbind(data.frame(lrself = unique(fake$lrself),
                    sim=i),
         predict(tmp,
                 newdata=fake)))
}
plot.dat <- probs %>%
  pivot_longer(PS:`UMP*`,
               names_to="party",
               values_to="prob") %>%
  mutate(party = factor(party,
                        levels=levels(dat$vote))) %>%
  group_by(lrself, party) %>%
  summarise(p = mean(prob),
            lwr= quantile(prob, .025),
            upr = quantile(prob, .975))
```

Figure

```
ggplot(plot.dat, aes(x=lrself,
                 y=p,
                 ymin = lwr,
                 ymax=upr)) +
  geom_ribbon(alpha=.25,
              colour="transparent") +
  geom_line() +
  facet_wrap(~party, ncol=2) +
  theme_bw() +
  theme(panel.grid=element_blank()) +
  labs(x="Left-right Self-placement",
       y="Predict Pr(Vote)")
```



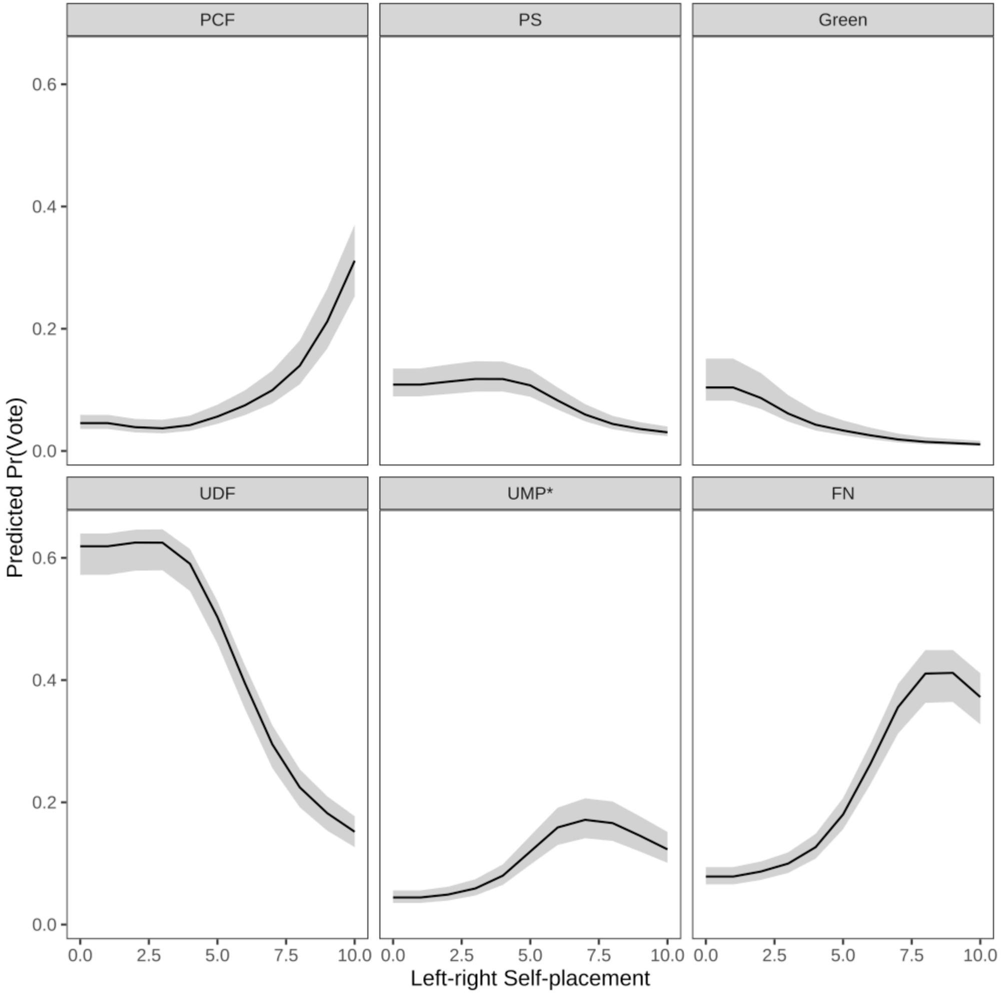
AME Approach

```r
B <- MASS::mvrnorm(2500, coef(mlogit2), vcov(mlogit2))
lrs <- 0:10
out <- data.frame(
  lrself = rep(lrs, each=6),
  party = rep(levels(dat$vote), length(lrs)),
  p = NA,
  lwr = NA,
  upr = NA
)
```

```r
for(j in lrs){
  tmp <- dat2l
  tmp$lrself <- j
  X <- model.matrix(
    update(mlogit2, data=tmp))

  b <- coef(mlogit2)
  XB <- X %*% t(B)
  Xb <- X %*% b
  p <- prop.table(matrix(exp(Xb), ncol=6, byrow=TRUE), 1)
  m <- colMeans(p)

  res <- NULL
  for(i in 1:ncol(XB)){
    exb <- exp(matrix(XB[,i], ncol=6, byrow=TRUE))
    p <- prop.table(exb, 1)
    res <- rbind(res, colMeans(p))
  }

  l <- apply(res, 2, quantile, .025)
  u <- apply(res, 2, quantile, .975)
  out$p[which(out$lrself == j)] <- m
  out$lwr[which(out$lrself == j)] <- l
  out$upr[which(out$lrself == j)] <- u
}
out$party <- factor(out$party, levels=levels(dat$vote))
```

Figure

```r
ggplot(out, aes(x=lrself, y=p,
                ymin = lwr, ymax=upr)) +
  geom_ribbon(alpha=.25, colour="transparent") +
  geom_line() +
  facet_wrap(~party) +
  theme_bw() +
  theme(panel.grid=element_blank()) +
  labs(x="Left-right Self-placement",
       y="Predicted Pr(Vote)")
```



First Differences: MER

```r
fake <- makeFakeData(mlogit2,
                     dat2l,
                     change=list(lrself=c(0,10)),
                     varying = "rp")
B <- MASS::mvrnorm(2500, coef(mlogit2), vcov(mlogit2))

tmp <- mlogit2
probs <- NULL
for(i in 1:2500){
  tmp$coefficients <- B[i,]
  probs <- rbind(probs,
   cbind(data.frame(lrself = unique(fake$lrself),
                    sim=i),
         predict(tmp,
                 newdata=fake)))
}

fd <- probs %>%
  group_by(sim) %>%
  summarise(across(PS:`UMP*`, ~diff(.x))) %>%
  ungroup %>%
  summarise(across(-sim, list(mean = ~mean(.x),
                              lwr = ~quantile(.x, .025),
                              upr = ~quantile(.x, .975),
                              pval = ~mean(.x < 0)))) %>%
  mutate(across(contains("pval"), ~ifelse(.x > .5, 1-.x, .x))) %>
  pivot_longer(everything(),
               names_pattern="(.*)_(.*)",
               names_to=c("party", ".value"))
```

```r
fd %>%
  mutate(across(-party,
                ~round(.x, 2)))
```

```
## # A tibble: 6 × 5
##   party  mean   lwr   upr  pval
##   <chr> <dbl> <dbl> <dbl> <dbl>
## 1 PS    -0.58 -0.66 -0.5      0
## 2 FN     0.39  0.26  0.53     0
## 3 Green -0.07 -0.1  -0.04     0
## 4 PCF   -0.1  -0.16 -0.05     0
## 5 UDF    0.09  0.05  0.14     0
## 6 UMP*   0.27  0.18  0.37     0
```

First Differences: AME

```
fake <- makeFakeData(mlogit2,
                     dat2l,
                     change=list(lrself=c(0,10)),
                     varying="rp")


fake1 <- fake0 <- dat2l
fake0$lrself <- 0
fake1$lrself <- 10

B <- MASS::mvrnorm(100, coef(mlogit2), vcov(mlogit2))
tmp <- mlogit2
res <- NULL
for(i in 1:100){
  tmp$coefficients <- B[i,]
  p_hat0 <- colMeans(predict(tmp, newdata=fake0))
  p_hat1 <- colMeans(predict(tmp, newdata=fake1))
  res <- rbind(res, p_hat1-p_hat0)
}

fd_ame <- as.data.frame(res) %>%
  summarise(across(everything(),
                   list(p = ~mean(.x),
                        lwr = ~quantile(.x, .025),
                        upr = ~quantile(.x, .975),
                        pval = ~mean(.x > 0)))) %>%
  mutate(across(contains("pval"),
                ~ifelse(.x > 0, 1-.x, .x))) %>%
  pivot_longer(everything(),
               names_pattern="(.*)_(.*)",
               names_to=c("party", ".value"))
```

```
fd_ame %>%
  mutate(across(-party,
                ~round(.x, 2)))
```

```
## # A tibble: 6 × 5
##   party      p   lwr   upr  pval
##   <chr> <dbl> <dbl> <dbl> <dbl>
## 1 PS    -0.46 -0.5  -0.43     0
## 2 FN     0.27  0.22  0.32     0
## 3 Green -0.08 -0.1  -0.06     0
## 4 PCF   -0.1  -0.13 -0.07     0
## 5 UDF    0.08  0.06  0.1      0
## 6 UMP*   0.29  0.26  0.32     0
```

Model Fit

PRE

```
yhat <- predict(mlogit2,newdata=dat2l)
yhat <- colnames(yhat)[apply(yhat, 1, which.max)]
yhat <- factor(yhat, levels=levels(dat$vote))
obs_vote <- dat2l %>%  filter(vote == 1) %>% ungroup %>% unnest(idx) %>% select(alt) %>% pull()
tab <- table(obs_vote, yhat)
pcp <- sum(obs_vote == yhat)/sum(tab)
pmc <- max(table(dat$vote))/sum(tab)
(pcp-pmc)/(1-pmc)
```

```
## [1] 0.251073
```

LR Test

```
ll1 <- logLik(mod)
ll2 <- logLik(mlogit2)
x2 <- -2*(ll1-ll2)
pchisq(x2, 9, lower.tail=FALSE)
```

```
## 'log Lik.' 2.629382e-28 (df=35)
```

Separation

Separation can be a problem in unordered categorical models, too. Let's add `demsat` to the model we ran in the beginning of the lecture:

```
mods <- multinom(vote ~ lrself + male + retnat + age +
    union + demsat, data=dat, trace=F)
mnlSig(mods)
```

```
##        (Intercept) lrself   male retnatsame retnatworse     age
## PS          3.572* 0.214* 0.601     -0.977      -1.054  -0.021*
## Green       1.840  0.388* 0.308     -1.340      -1.783  -0.048*
## UDF        -0.842  0.919* 0.705     -1.847      -2.392* -0.005
## UMP*       -1.251  1.151* 0.515     -1.508      -2.438* -0.011
## FN        -14.142* 0.802* 0.341     -1.382      -1.373  -0.018
##        unionyes, union member demsatsomewhat satisfied demsata little satisfied
## PS                    -1.339*                    0.490                    0.405
## Green                 -1.065*                    1.616                    1.552
## UDF                   -1.234*                    0.033                   -0.264
## UMP*                  -1.231*                   -0.118                   -0.534
## FN                    -1.536*                   13.139*                  13.949*
##        demsatnot satisfied at all
## PS                         -0.146
## Green                       1.573
## UDF                        -1.350
## UMP*                       -1.464
## FN                         14.195*
```

Bias Reduced Model

```
mods_br <- brmultinom(formula(mods), data=dat)
```

```
noquote(t(mnlSig(mods_br)))
```

```
##                             PS      Green   UDF     UMP*    FN
## (Intercept)                 3.045*  1.531   -1.253  -1.664  -3.447
## lrself                      0.201*  0.373*  0.889*  1.115*  0.775*
## male                        0.581   0.295   0.684   0.498   0.331
## retnatsame                  -0.636  -1.011  -1.489  -1.157  -1.077
## retnatworse                 -0.672  -1.414  -1.990* -2.034* -1.054
## age                         -0.021* -0.047* -0.005  -0.011  -0.017
## unionyes, union member      -1.310* -1.020* -1.174* -1.194* -1.447*
## demsatsomewhat satisfied    0.585   1.509   0.126   -0.007  2.192
## demsata little satisfied    0.488   1.441   -0.167  -0.424  2.976
## demsatnot satisfied at all -0.065   1.462   -1.161  -1.319  3.211*
```

Review

1. Develop Multinomial Logit Model
2. Effects and Effect Displays
3. Model Fit and Evaluation
4. IIA Assumption
5. Conditional Logit
6. Separation Problems

Replicating Anderson and Stephenson

Since Anderson and Stephenson used CES data, we can replicate their work. We won't do all the models, but we can look at one of them to compare the effects they got with effects in the different ways we have calculated them. The authors write:

> “We expect that, if the issue is positional, we should see clear differentiation between the parties based on the left-right ideological split. If the environment is a valence issue, however, then we expect to find that environmental support has the greatest effect on the party perceived as best able to govern on ”

1. How would you estimate these models?
2. How well do they fit?
3. What are the variable effects?
4. Which idea do you think is right?