Lab 10: Multinomial and Ordered Probit Models

Eric Parajon

Code has been adapted from Simon Hoellerbauer and Isabel Laterzo.

Load necessary packages:

Ordered Probit

Data and resources for this section from http://stats.idre.ucla.edu/r/dae/ordinal-logistic-regression/.

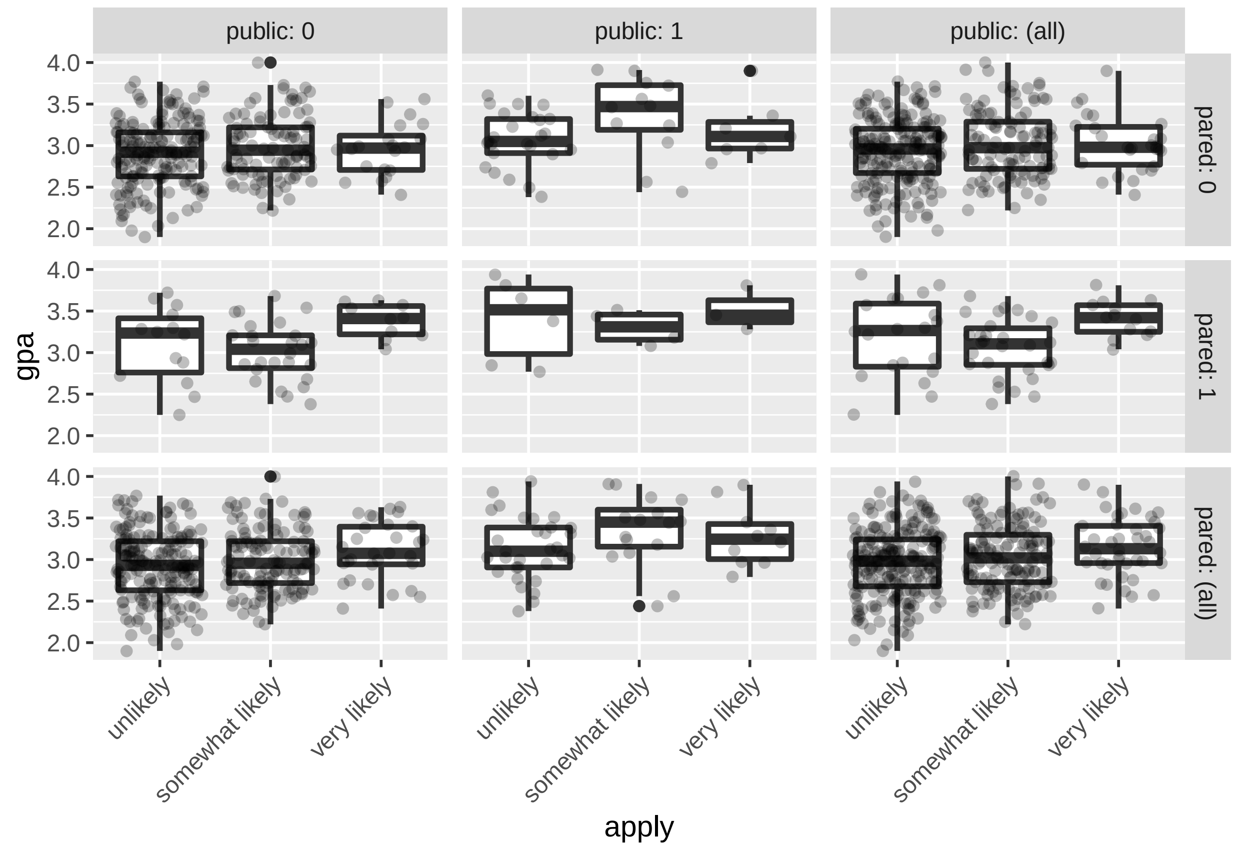
We are going to expand an earlier example that we used for logistic regression: the data about application/acceptance into graduate school.

These data include our DV, 'a three level variable called apply, with levels unlikely, somewhat likely, and very likely, and three predictor variables: pared (dichotomous, does either parent have a graduate degree?) public (dichotomous, 0 or 1 indicating whether or not the students' undergrad institution is public) and gpa (continuous).

```
grad <- read_dta("https://stats.idre.ucla.edu/stat/data/ologit.dta")
#when using read_dta, and we want to turn labelled vectors into factors,
#we have to use to_factor from labelled packaged
grad$apply <- to_factor(grad$apply)
```

Remember that before we even ran any tests for our logit model, we made a plot to understand the distribution of our variables of interest? We should do that again. Let's try a boxplot- or rather, a series of boxplots.

```
ggplot(grad, aes(x = apply, y = gpa)) +
  geom_boxplot(size = 1) +
  geom_jitter(alpha = .25) +
  facet_grid(pared~public, margins = TRUE, labeller=label_both) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```



Now, after looking at our data and checking for any unusual patterns, we can build a simple model.

For ordered outcomes, we need to use the MASS package (rather than base R). Otherwise, our syntax looks much the same.

```
#First, a few checks to make sure our variables are in order:
is.factor(grad$apply)
```

```
## [1] TRUE
```

```
print(levels(grad$apply))
```

```
## [1] "unlikely"        "somewhat likely" "very likely"
```

```
#Now, our model:
m1 <- polr(apply ~ pared + public + gpa,
           data = grad, method = "probit", Hess = TRUE)
summary(m1)
```

```
## Call:
## polr(formula = apply ~ pared + public + gpa, data = grad, Hess = TRUE,
##     method = "probit")
##
## Coefficients:
##          Value Std. Error t value
## pared  0.59811     0.1579 3.78881
## public 0.01016     0.1728 0.05878
## gpa    0.35815     0.1568 2.28479
##
```

```
## Intercepts:
##                              Value  Std. Error t value
## unlikely|somewhat likely    1.2968 0.4675     2.7738
## somewhat likely|very likely 2.5028 0.4766     5.2517
##
## Residual Deviance: 717.4951
## AIC: 727.4951
```

```
#What is the effect of increasing each predictor by one unit?
#How do the odds of moving from a lower to adjacent higher category change?
coefs <- m1$coefficients[1:3]
#odds ratios
exp(coefs)
```

```
##    pared   public      gpa
## 1.818675 1.010211 1.430684
```

We can interpret these exponentiated coefficients in the same way as odds ratios in a logistic regression, but for the odds of each outcome category. Here, we can say that for pared, a one unit increase in parental education, i.e., going from 0 (Low) to 1 (High), the odds of very likely applying versus somewhat likely or unlikely applying combined are 1.81x greater, given that all of the other variables in the model are held constant.

We can (and should) also informally test the proportional odds assumption. What's this?

```
m1_prop_1 <- glm(I(as.numeric(apply) >= 2) ~ pared, family = "binomial",
                 data = grad)


m1_prop_2 <- glm(I(as.numeric(apply) >= 3) ~ pared, family = "binomial",
                 data = grad)

#Pared = 0
check_1 <- m1_prop_1$coef[1] - m1_prop_2$coef[1]
check_1
```

```
## (Intercept)
##    2.062399
```

```
#Pared = 1
check_2 <- (m1_prop_1$coef[1] + m1_prop_1$coef[2]) -
  (m1_prop_2$coef[1] + m1_prop_2$coef[2])
check_2
```

```
## (Intercept)
##    2.112541
```

When pared is equal to no (0) the difference between the predicted value for apply greater than or equal to two and apply greater than or equal to three is roughly 2 (-0.378 - -2.440 = 2.062). For pared equal to yes the difference in predicted values for apply greater than or equal to two and apply greater than or equal to three is also roughly 2 (0.765 - -1.347 = 2.112).

We will also want to generate predicted probabilities and plot them. Let's first do so with a plot of the cumulative probability (so the probabilities of being in each subcategory sum to 1).

We can also check this first:

```
tau1 <- m1$zeta[1] #extract latent thresholds (calculated w polr) from our model!
tau2 <- m1$zeta[2]
```

```r
p1 <- invlogit(tau1) # 1
p2 <- invlogit(tau2) - invlogit(tau1) # 2
p3 <- 1-invlogit(tau2)

p1 + p2 + p3
```

```
## unlikely|somewhat likely
##                        1
```

And now move on to plotting.

```r
# if we wanted to plot the range of gpa for all four combinations of pared and
#public, we could do the following:
#grad_2 <- expand.grid(pared = c(0:1),
#                      public = c(0,1),
#                      gpa = seq(from = 1.9, to = 4, length.out = 1000))
#We would then create an indicator variable for the four combinations, and then
#use facet_wrap in ggplot to create 4 different plots
#however, if we just want one easily interpretable plot, we can do the following:
table(grad$pared); table(grad$public)
```

```
##
##   0   1
## 337  63
```

```
##
##   0   1
## 343  57
```

```r
grad_2 <- expand.grid(pared = 0, #modal value
                      public = 0, #modal value
                      gpa = seq(from = 1.9, to = 4, length.out = 1000))
colnames(grad_2)<- c("pared", "public", "gpa")
probs <- predict(m1, grad_2, type = "probs") #type = is different than for binomial model
head(rowSums(probs)) #just to double check
```

```
## 1 2 3 4 5 6
## 1 1 1 1 1 1
```

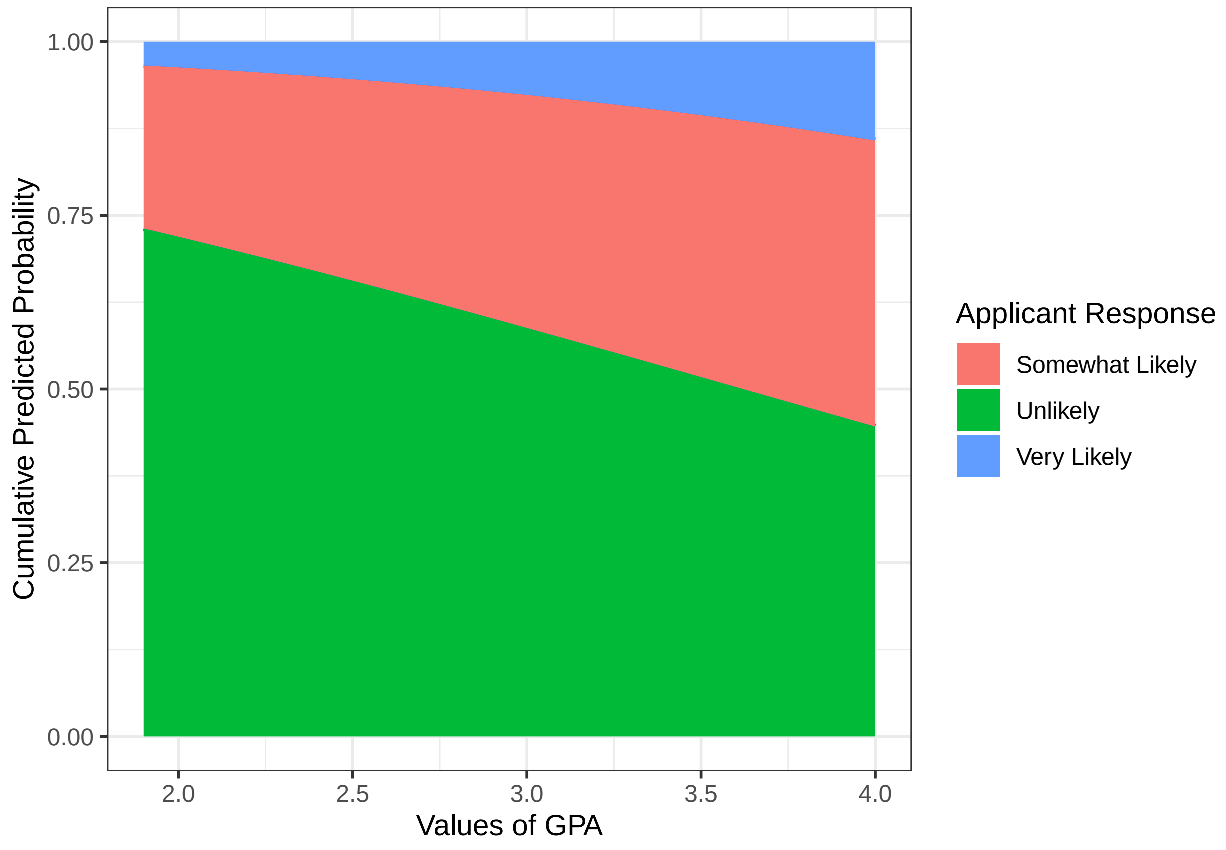
```r
cumul_probs <- t(apply(probs, 1, cumsum)) #get the cumulative sum of the rows
#transpose into columns for plotting and merging with values of gpa
head(cumul_probs)
```

```
##    unlikely somewhat likely very likely
## 1 0.7311624       0.9657982           1
## 2 0.7309140       0.9657411           1
## 3 0.7306654       0.9656839           1
## 4 0.7304167       0.9656266           1
## 5 0.7301679       0.9655692           1
## 6 0.7299190       0.9655118           1
```

```r
plot_data <- data.frame(grad_2$gpa, cumul_probs)
ggplot(data=plot_data, aes(x=grad_2.gpa))+
    geom_smooth(data=plot_data, aes(x=grad_2.gpa, y=unlikely), se=FALSE)+
    geom_smooth(data=plot_data, aes(x=grad_2.gpa, y=somewhat.likely), se=FALSE)+
    geom_smooth(data=plot_data, aes(x=grad_2.gpa, y=very.likely), se=FALSE)+
    geom_ribbon(data=plot_data, aes(ymin=0, ymax=unlikely, fill="Unlikely"))+
    geom_ribbon(data=plot_data, aes(ymin=unlikely, ymax=somewhat.likely,
                                    fill="Somewhat Likely"))+
    geom_ribbon(data=plot_data, aes(ymin=somewhat.likely, ymax=very.likely,
                                    fill="Very Likely"))+
    labs(x="Values of GPA", y="Cumulative Predicted Probability")+
    scale_fill_discrete(name="Applicant Response")+
    theme_bw()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 3)
```

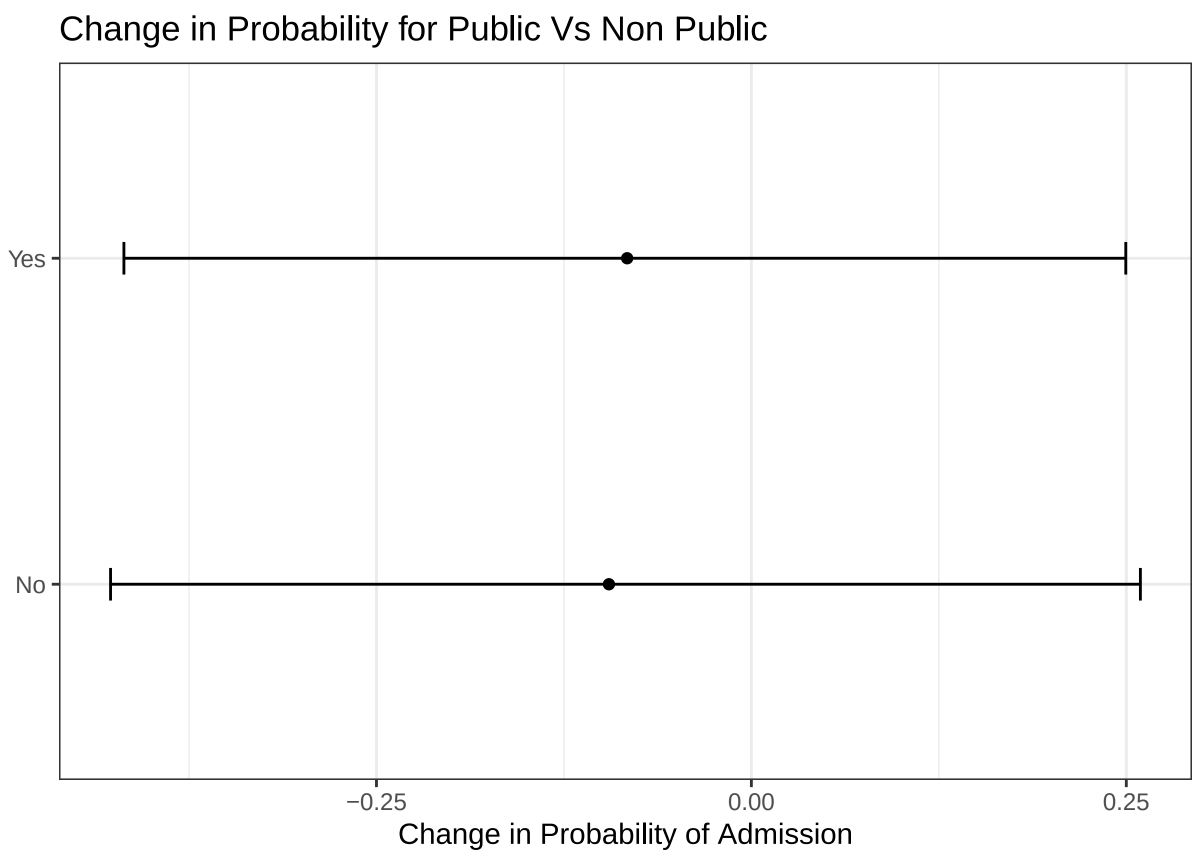


And plotting a non-cumulative probability example.

```
#data frame
grad_3 <- expand.grid(
  pared = 1, #assume parents education =1
  public = c(1,0), #vary values of public
  gpa = mean(grad$gpa)) #average gpa

library(MASS)
set.seed(831213)

coef_samp <- mvrnorm(1e3,coef(m1)
                    ,vcov(m1)[1:3,1:3]) #keep relevant vals of vcov matrix

## Get linear predictors
hyp_X <- model.matrix(~pared+public+gpa
                      ,data = grad_3)[,-1] #no intercept
eta <- hyp_X %*% t(coef_samp)
summary(t(eta))
```

```
##        1                 2
##  Min.   :0.3812   Min.   :0.419
##  1st Qu.:1.3953   1st Qu.:1.391
##  Median :1.7183   Median :1.691
##  Mean   :1.7040   Mean   :1.701
##  3rd Qu.:2.0157   3rd Qu.:2.010
##  Max.   :3.0432   Max.   :3.061
```

```
## Get pred probs samples
taus <- m1$zeta #extract thresholds
pred_prob_diffs <- apply(eta,c(1,2)
                    ,function(x, J, tau){
                      prob_vec <- array(NA,J) #J categories - array() is generalizable
                      prob_vec[1] <- plogis(tau[1]-x) #prob of being in 1st category
                      for(i in 2:(J-1)) #move to next category
                        prob_vec[i] <- plogis(tau[i]-x) - plogis(tau[i-1]-x)
                        #probability of being in middle category
                      prob_vec[J] <- 1 - plogis(tau[J-1]-x)
                        #probability of being in top category (sums to 1)
                      return(prob_vec[J]-prob_vec[1])
                      # Diff. in prob from top to bottom cat.
                    }
                    ,J = 3 #3 possible outcomes
                    ,tau = taus)

## Obtain 90% confidence interval on probabilities

fx_on_prob_ci <- t(apply(pred_prob_diffs,1
                       ,quantile,probs=c(0.05,0.5,0.95)))

#find difference in predicted probabilities and confidence around those differences
public <- c("Yes", "No")
#for easy plotting
fx_on_prob_ci <- as.data.frame(cbind(fx_on_prob_ci, public))
colnames(fx_on_prob_ci)<-c("low", "pe", "high", "public")

fx_on_prob_ci$low <- as.numeric(fx_on_prob_ci$low) #fix issues with factors
fx_on_prob_ci$pe <- as.numeric(fx_on_prob_ci$pe)
fx_on_prob_ci$high <- as.numeric(fx_on_prob_ci$high)


## Produce plot
ggplot(fx_on_prob_ci, aes(x=pe, y=as.factor(public))) +
  labs(x="Change in Probability of Admission", y="") +
  ggtitle("Change in Probability for Public Vs Non Public")+
  geom_point(aes(x=pe)) +
  geom_errorbarh(aes(xmin=low, xmax=high), height=0.1) + #errorbar for confidence
  theme_bw()
```



```
#not much of an effect, as our model told us originally!
```

Multinomial and Conditional Logit

```
#Loading additional necessary packages

pacman::p_load(nnet,#for multinom()
               mlogit,#for conditional mnl
               cowplot)#we will use it to put plots together (an extension to ggplot2)
```

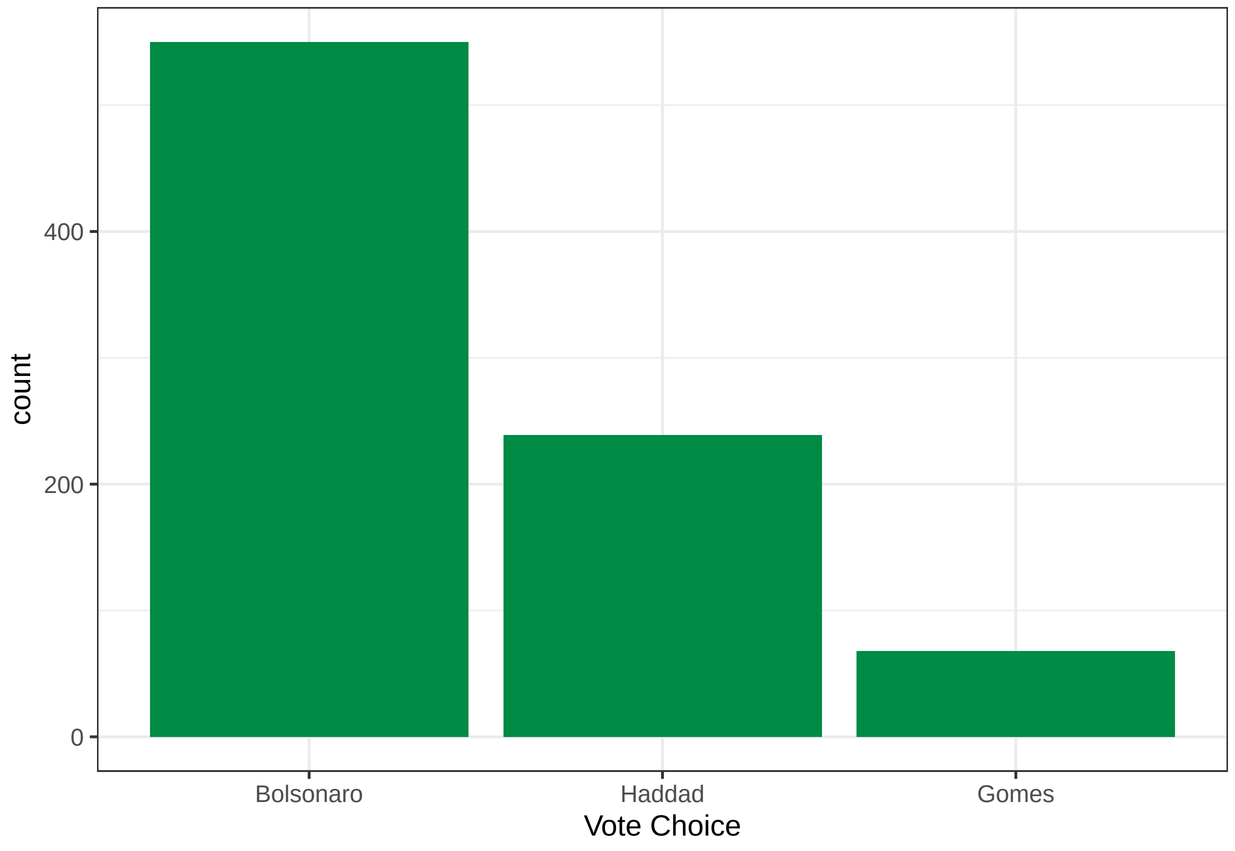
For this example, we'll use data from the 2017/18 round of LAPOP in Brazil. Ths data has a variable called "vote_choice" which is a categorical variable which we'll treat as unordered. In particular, we'll examine who individuals voted for in Brazil in the 2018 national election (note there were MANY candidates, we'll only examine three - Bolsonaro, Haddad, and Gomes).

```
brazil_data <- readRDS("BrazilLAPOP_2018.rds")
#A bit of housekeeping:

#vote choice variable
brazil_data <- subset(brazil_data, vote_choice == "1501" |
                        vote_choice == "1502" |
                        vote_choice == "1503")
brazil_data$vote_choice <- as.factor(brazil_data$vote_choice)
```

```r
recode(brazil_data$vote_choice, "1501" = "Bolsonaro", "1502" = "Haddad",
       "1503" = "Gomes")

levels(brazil_data$vote_choice) <- c("Bolsonaro", "Haddad", "Gomes")

#capital punishment variable
brazil_data$capital_pun <- ifelse(brazil_data$capital_pun == 1, 1,
                                  ifelse(brazil_data$capital_pun == 2, 0, NA))

#sex variable
#male is 1, female is 0
brazil_data$sex <- ifelse(brazil_data$sex == 1, 1,
                                  ifelse(brazil_data$sex == 2, 0, NA))
```

Let's make a descriptive plot of our DV. This is good practice - something to consider for your replication papers!

```r
ggplot(data=brazil_data)+
  geom_bar(aes(x=vote_choice), fill="springgreen4")+
  scale_x_discrete(name="Vote Choice")+
  theme_bw()
```



Let's run our first model.

```r
multi_1 <- multinom(vote_choice ~ sex + age + capital_pun,
                    data=brazil_data, Hess=TRUE)
summary(multi_1)
```

```
#p values
z <- summary(multi_1)$coefficients/summary(multi_1)$standard.errors
# 2-tailed Wald z tests to test significance of coefficients
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p

#variance covariance
vcov(multi_1)
```

We can change the reference category if we wish, and re-run:

```
## # weights:  15 (8 variable)
## initial  value 909.650975
## iter  10 value 678.683284
## final  value 678.338150
## converged
```

```
## Call:
## multinom(formula = vote_choice ~ sex + age + capital_pun, data = brazil_data,
##     Hess = TRUE)
##
## Coefficients:
##           (Intercept)       sex          age capital_pun
## Bolsonaro  0.07747397 0.5880453  0.008399133   0.2222934
## Gomes     -0.62025531 0.3802102 -0.019085614  -0.3093388
##
## Std. Errors:
##           (Intercept)       sex         age capital_pun
## Bolsonaro   0.2639588 0.1597405 0.005448676   0.1602376
## Gomes       0.4609828 0.2863508 0.010459885   0.2879088
##
## Residual Deviance: 1356.676
## AIC: 1372.676
```

We can also, as you may recall, run this as a series of binary logits (after some minor data recoding):

```
#filter data
brazil_data_2<-subset(brazil_data[which(brazil_data$vote_choice=="Bolsonaro" |
                                          brazil_data$vote_choice=="Haddad"),])

brazil_data_3<-subset(brazil_data[which(brazil_data$vote_choice=="Haddad" |
                                          brazil_data$vote_choice=="Gomes"),])

#recode values
brazil_data_2$vote_choice<-ifelse(brazil_data_2$vote_choice=="Haddad", 0,1)
#votes for Bolsonaro
brazil_data_3$vote_choice<-ifelse(brazil_data_3$vote_choice=="Haddad", 0,1)
#votes for Gomes

#check it out
head(brazil_data_2$vote_choice)
head(brazil_data_3$vote_choice)



#models!
```

```r
glm_1<-glm(vote_choice~sex + age + capital_pun,
           data=brazil_data_2,
           family=binomial (link="logit"))


glm_2<-glm(vote_choice~sex + age + capital_pun,
           data=brazil_data_3,
           family=binomial (link="logit"))

coefs<-rbind(coef(glm_1), coef(glm_2), coef(multi_2))
coefs
```

Finally, we can run this as a slightly different model using a multinomial choice model and another package. To do so, we need to take a few extra steps. This allows us to change our data from a single observation per individual that records their vote intent to *multiple observations for an individual that encompass all of their choices.*

This means we can have information about alternatives that would be constant across individuals (ie policy position for each of these candidates) *and* information about individuals that is constant across alternatives (like gender or age of an individual).

In *these particular data* we only have individual-level information. However, we can set up these data in the same way we would a model that allows us alternative choices.

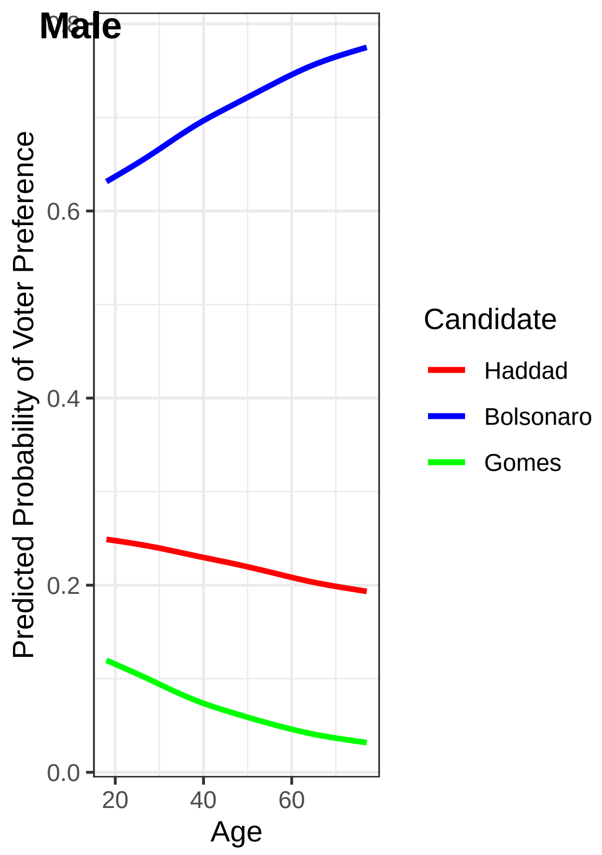
```r
#First, a look at our data
brazil_data[1:10,]
levels(brazil_data$vote_choice)

#Reshape- we can do this with a function in the mlogit package

brazil_data <- as.data.frame(na.omit(brazil_data))

brazil_long <- mlogit.data(brazil_data,
                            shape = "wide",
                            choice = "vote_choice",
                            alt.levels = c("Haddad", "Bolsonaro", "Gomes"))
                           #These match the levels above



#Check our data now
head(brazil_long)
brazil_long[1:10,]

#Replicate our multinomial logit and check with the 2nd model, above:
multi_3 <- mlogit(vote_choice ~ 0|sex+age+capital_pun,
                  data=brazil_long, reflevel="Haddad")
# 0| is used in this case because we have no 'alternative-level variables'

head(model.matrix(multi_3))

check<-list(summary(multi_3), summary(multi_2))
check #look at the intercept for multi_3 and compare to multi_2
```

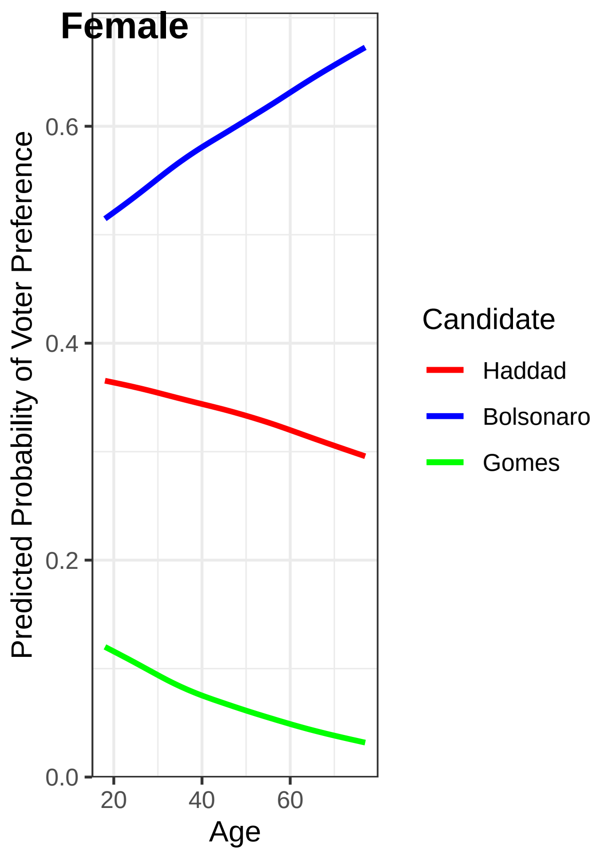
Let's return to our multinomial model (multi_2) and generate some QIs (predicted probabilities). We'll take a random sample of our data and then use our expand.grid() to see the effect of different combinations of our explanatory variables.

```r
set.seed(2876)
brazil_test<-brazil_data[,c("vote_choice", "sex", "age", "capital_pun")]
brazil_probs<-brazil_test[sample(nrow(brazil_test), 50),]
#expand.grid gives us ALL possible combinations of the values in our data
#we randomly sampled 50 rows to make the computation easier.
brazil_d<-expand.grid(brazil_probs)

pred<-predict(multi_2, newdata=brazil_d, type="probs")
#'probs' gives us information about the predicted probability of each category
# in our DV

#What should this look like?
head(rowSums(pred)) #looks good!

plotdat<-as.data.frame(cbind(pred, brazil_d$sex,
                             brazil_d$age, brazil_d$capital_pun))
```

And we can plot, with a bit more detail:

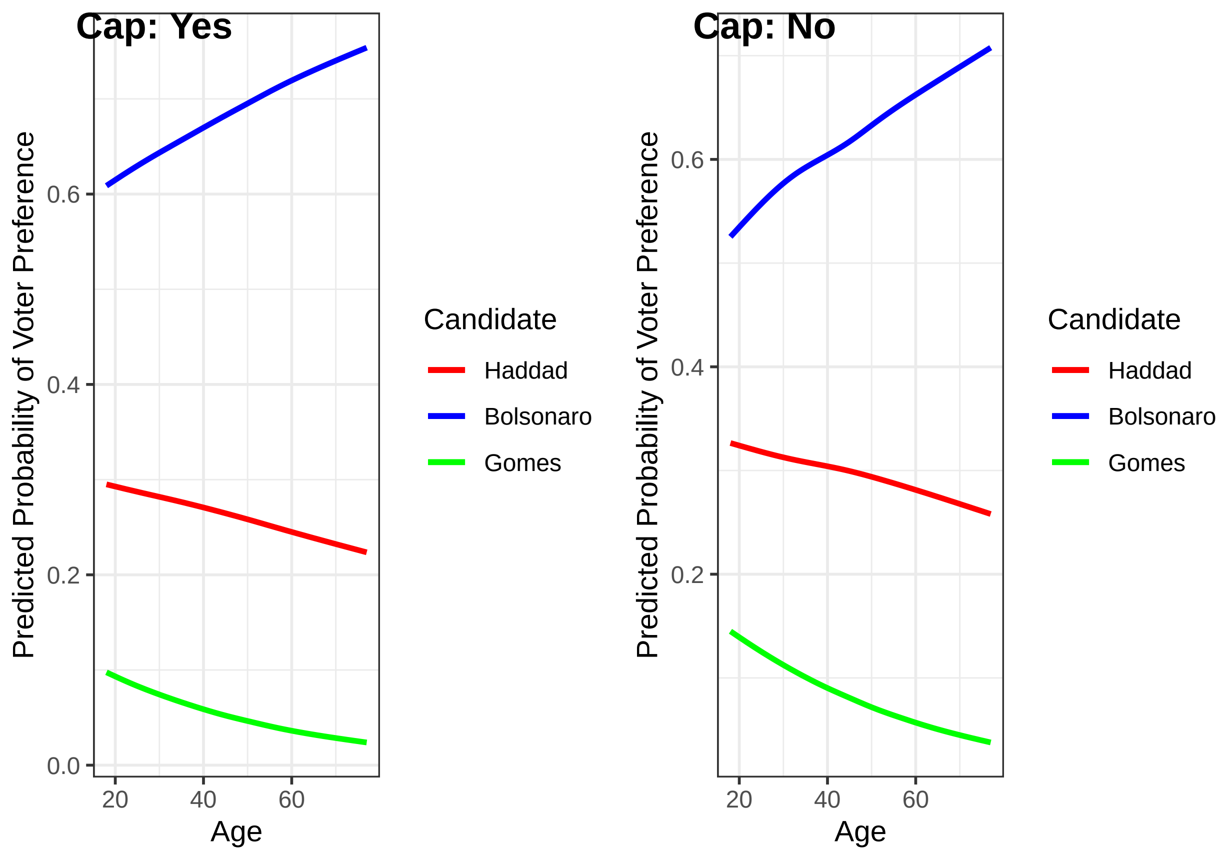
```r
#Let's separate out our predicted probabilities for each candidate by gender:
set.seed(12)
m_pred <-plotdat[which(plotdat$V4==1),]
m_pred <- m_pred[sample(nrow(m_pred), 1000), ]
f_pred <-plotdat[which(plotdat$V4==0),]
f_pred <- f_pred[sample(nrow(f_pred), 1000), ]


mplot<-ggplot(data=m_pred)+
  geom_smooth(data=m_pred, aes(x=V5, y=Haddad, colour="Haddad"), se=FALSE)+
  geom_smooth(data=m_pred, aes(x=V5, y=Bolsonaro, colour="Bolsonaro"), se=FALSE)+
  geom_smooth(data=m_pred, aes(x=V5, y=Gomes, colour="Gomes"), se=FALSE)+
  labs(x="Age", y="Predicted Probability of Voter Preference")+
  scale_colour_manual(name="Candidate",
    values=c(Haddad="red", Bolsonaro="blue", Gomes="green"))+
  theme_bw()


fplot<-ggplot(data=f_pred)+
  geom_smooth(data=f_pred, aes(x=V5, y=Haddad, colour="Haddad"), se=FALSE)+
  geom_smooth(data=f_pred, aes(x=V5, y=Bolsonaro, colour="Bolsonaro"), se=FALSE)+
  geom_smooth(data=f_pred, aes(x=V5, y=Gomes, colour="Gomes"), se=FALSE)+
  labs(x="Age", y="Predicted Probability of Voter Preference")+
  scale_colour_manual(name="Candidate",
    values=c(Haddad="red", Bolsonaro="blue", Gomes="green"))+
  theme_bw()


both <- plot_grid(mplot, fplot, labels = c('Male', 'Female'))
both
```





```
#Now let's separate out by support for capital punishment

cap_pred <-plotdat[which(plotdat$V6==1),]
cap_pred <- cap_pred[sample(nrow(cap_pred), 1000), ]
nocap_pred <-plotdat[which(plotdat$V6==0),]
nocap_pred <- nocap_pred[sample(nrow(nocap_pred), 1000), ]


cap_plot<-ggplot(data=cap_pred)+
  geom_smooth(data=cap_pred, aes(x=V5, y=Haddad, colour="Haddad"), se=FALSE)+
  geom_smooth(data=cap_pred, aes(x=V5, y=Bolsonaro, colour="Bolsonaro"), se=FALSE)+
  geom_smooth(data=cap_pred, aes(x=V5, y=Gomes, colour="Gomes"), se=FALSE)+
  labs(x="Age", y="Predicted Probability of Voter Preference")+
  scale_colour_manual(name="Candidate",
    values=c(Haddad="red", Bolsonaro="blue", Gomes="green"))+
  theme_bw()


nocap_plot<-ggplot(data=nocap_pred)+
  geom_smooth(data=nocap_pred, aes(x=V5, y=Haddad, colour="Haddad"), se=FALSE)+
  geom_smooth(data=nocap_pred, aes(x=V5, y=Bolsonaro, colour="Bolsonaro"), se=FALSE)+
  geom_smooth(data=nocap_pred, aes(x=V5, y=Gomes, colour="Gomes"), se=FALSE)+
  labs(x="Age", y="Predicted Probability of Voter Preference")+
  scale_colour_manual(name="Candidate",
    values=c(Haddad="red", Bolsonaro="blue", Gomes="green"))+
  theme_bw()
```

```r
both2 <- plot_grid(cap_plot, nocap_plot, labels = c('Cap: Yes', 'Cap: No'))
both2
```



Lab Assignment

Ordered Outcome Practice

Use the data below. Estimate an ordered logit with 'warm' as the DV, using male, white, age, education, and prestige as independent variables.

Check the proportional odds assumption for at least one of your variables - does it hold?

Generate a cumulative predicted probability plot.

Data and description from Christopher Adolph: Example data from 1977, 1989 GSS: Attitudes towards working mothers: "A working mother can establish just as warm and secure of a relationship with her child as a mother who does not work." SD, D, A, SA Covariates: male respondent; white respondent; age of respondent; years of education of respondent; prestige of respondent's occupation (% considering prestigious)

```r
hw <- read.csv(url("http://faculty.washington.edu/cadolph/mle/ordwarm2.csv"))


#Cleaning

hw<-hw %>%
   mutate(warm = case_when(.$warm == 1 ~ "Strongly disagree",
                           .$warm==2 ~ "Disagree",
                           .$warm==3 ~ "Agree",
                           .$warm==4 ~ "Strongly agree")) %>%
    mutate(warm = factor(warm, levels=c("Strongly disagree","Disagree","Agree","Strongly agree")))

#Create the model
warm_model <- polr(warm~male+white+age+ed+prst,
           data = hw, method = "probit", Hess = TRUE)
summary(warm_model)

#What is the effect of increasing each predictor by one unit?
#How do the odds of moving from a lower to adjacent higher category change?
coefs <- warm_model$coefficients[1:5]
#odds ratios
exp(coefs)


#Check the proportional odds assumption for male
m1_prop_1 <- glm(I(as.numeric(warm) >= 2) ~ male, family = "binomial",
                 data = hw)


m1_prop_2 <- glm(I(as.numeric(warm) >= 3) ~ male, family = "binomial",
                 data = hw)

#male=0
check_1 <- m1_prop_1$coef[1] - m1_prop_2$coef[1]
check_1

#male=1
check_2 <- (m1_prop_1$coef[1] + m1_prop_1$coef[2]) -
  (m1_prop_2$coef[1] + m1_prop_2$coef[2])
check_2

#Generate a cumulative predicted probability plot.
table(hw$male); table(hw$white); mean(hw$ed)
hw_2 <- expand.grid(male = 1, #modal value
                      white = 1, #modal value
                      ed=mean(hw$ed),
                      age=mean(hw$age),
                      prst = seq(from = min(hw$prst), to = max(hw$prst), length.out = 2293))

probs <- predict(warm_model, hw_2, type = "probs") #type = is different than for binomial model
head(rowSums(probs)) #just to double check
cumul_probs <- t(apply(probs, 1, cumsum)) #get the cumulative sum of the rows
#transpose into columns for plotting and merging with values of gpa
head(cumul_probs)
plot_data <- data.frame(hw$prst, cumul_probs)
ggplot(data=plot_data, aes(x=hw.prst))+
    geom_smooth(data=plot_data, aes(x=hw.prst, y=Strongly.disagree), se=FALSE)+
    geom_smooth(data=plot_data, aes(x=hw.prst, y=Disagree), se=FALSE)+
    geom_smooth(data=plot_data, aes(x=hw.prst, y=Agree), se=FALSE)+
      geom_smooth(data=plot_data, aes(x=hw.prst, y=Strongly.agree), se=FALSE)+
    geom_ribbon(data=plot_data, aes(ymin=0, ymax=Strongly.disagree, fill="Strongly.disagree"))+
    geom_ribbon(data=plot_data, aes(ymin=Strongly.disagree, ymax=Disagree,
                                fill="Disagree"))+
    geom_ribbon(data=plot_data, aes(ymin=Disagree, ymax=Agree,
                                fill="Agree"))+
         geom_ribbon(data=plot_data, aes(ymin=Agree, ymax=Strongly.agree,
                                fill="Strongly.agree"))+
    labs(x="Values of prestige of occupation", y="Cumulative Predicted Probability")+
    scale_fill_discrete(name="Hard Work")+
    theme_bw()
```