Homework 2 and 3 Suggested Solution

Haohan Chen

February 28, 2018

Assignment 2

Ordered outcomes

1

If you analyze the data using a linear model estimated using OLS. Here are the two major consequences.

- **You make a strong assumption on the intervals between levels**. Using a linear model, you assume your dependent variable is an interval variable, which means the intervals between values are equally spaced. For example, if your dependent variable is a typical five-level Likert-scale item and you code it as 1, 2, 3, 4, 5, you assume the difference between "neutral" and "disagree" is the same as the difference between "strongly agree" and "agree". That said, please note that using a linear model does *NOT force* you to assume the difference between levels equal. For example, You can recode a Likert item into 1 (strongly disagree), 10 (disagree), 100 (neutral), 1000 (agree), 10000 (strongly agree). In this way, you assume the difference between "strongly agree" and "agree" is way larger than the difference between "strongly disagree" and "disagree". You are free to make any assumption about intervals between levels. But you have to make **some** assumption if you use a linear model. Such assumptions may be hard to defend. In contrast, many GLM for ordered outcomes does not force you to make assumption about the intervals between levels for your dependent variables.
- **It's hard to make sense of a predicted outcome.** Your predicted outcome is usually different from the numeric values associated with different levels in your coding scheme. It makes it hard to interpret the predicted outcome. For instance, if you code a Linkert-scale outcome into 1 - 5 (i.e. "strongly disagree" = 1, . . . , "strongly agree" = 5), what does a predicted outcome 3.5 with a 95% Confidence Interval $(3.1, 3.7)$ mean? You can create your rule to map them to predicted outcomes (levels). But this means you add another set of assumption, which may be hard to defend.

Additional reference: UCLA idre "What is the difference between categorical, ordinal and interval variables?"

2

I use a Proportional-Odds Cumulative Logit Model. Let Y be the outcome and $\mathbf{x}$ be a vector of independent variables.

$$logit(Pr(Y \leq j|\mathbf{x})) = \alpha_j + \boldsymbol{\beta}'\mathbf{x} \quad j = 1, 2, ...J-1$$

With 4 total ordered categories, I first obtain the predicted probability $Pr(Y \leq 3|\mathbf{x})$ and $Pr(Y \leq 2|\mathbf{x})$.

$$\begin{aligned} logit(Pr(Y \leq 3|\mathbf{x})) = \alpha_3 + \boldsymbol{\beta}'\mathbf{x} \quad &\Rightarrow \quad Pr(Y \leq 3|x) = \frac{1}{1 + e^{-\alpha_3 - \boldsymbol{\beta}'\mathbf{x}}} \\ logit(Pr(Y \leq 2|\mathbf{x})) = \alpha_2 + \boldsymbol{\beta}'\mathbf{x} \quad &\Rightarrow \quad Pr(Y \leq 2|\mathbf{x}) = \frac{1}{1 + e^{-\alpha_2 - \boldsymbol{\beta}'\mathbf{x}}} \end{aligned}$$

Then the expression to predict $Pr(Y = 3|\mathbf{x})$ can be written as follows

$$Pr(Y=3) = Pr(Y \leq 3|\mathbf{x}) - Pr(Y \leq 2|\mathbf{x}) = \frac{1}{1+e^{-\alpha_3-\boldsymbol{\beta}'\mathbf{x}}} - \frac{1}{1+e^{-\alpha_2-\boldsymbol{\beta}'\mathbf{x}}}$$

Additional reference: PennState "The Proportional-Odds Cumulative Logit Model"

Grading Rubics: You get the credit if you write the euqation of a ordered logit model. But please be aware that getting the predicted probability is not straightforward.

Social Insurance

1.

I fit an Order Probit Model with the Parallel Assumption. I use the `VGAM` package in R (feel free to use other packages).

$$Y_i^* \sim P(y_i^* \mid \boldsymbol{X}_i \boldsymbol{\beta}) \text{ where } \boldsymbol{X}_i = (\text{female}_i, \text{age}_i, \text{eduyrs}_i, \text{conserv}_i)$$

$$\text{with an observation mechanism } y_{ji} = \begin{cases} 1 & \tau_{j-1,i} \leq y_i^* < \tau_{j,i} \\ 0 & \text{otherwise} \end{cases}$$

```
# Load libraries
# ---------------
library(ggplot2)
# Load the data
# -----------------
d <- read.table("socialinsurance.dat", header = T)
# Change the format of DV: spending into Ordered
# ----------------------------------------------
d$spend <- ordered(d$spend)
# Fit an ordered Probit model
# ----------------------------
m <- MASS::polr(spend ~ female + age + eduyrs + conserv, method = "probit",
                data = d, Hess = T)
```

2.

I use the `predict` function to obtained predicted outcomes.

```
# Check model by comparing predicted and observed outcome
# -------------------------------------------------------
pred_y <- predict(m, newdata = d, type="class")
```

Then I examine the overall proportion of correct prediction ($\hat{y} = y$).

```
# Total proportion of correct prediction
sum(ordered(pred_y) == d$spend) / nrow(d)
```

```
## [1] 0.472
```

The model correctly predicts **47.2%** of the observations.

I further examine the proportion of correct prediction conditional on the observed levels. I build a confusion matrix as below.

```r
mat_confusion <- table(`Actual Spending` = d$spend, `Predicted Spending` = pred_y)
mat_confusion
```

```
##                Predicted Spending
## Actual Spending   0   1   2
##               0  85 223   5
##               1  66 385   4
##               2  11 219   2
```

I calculate the marginal proportions by row.

```r
round(prop.table(mat_confusion, margin = 1), 3)
```

```
##                Predicted Spending
## Actual Spending     0     1     2
##               0 0.272 0.712 0.016
##               1 0.145 0.846 0.009
##               2 0.047 0.944 0.009
```

The above matrices show that the model tends to predict the spending level to be 1. Why is this? An obvious reason is that the outcome classes are imbalanced in the dataset.

```r
prop.table(table(Spending = d$spend))
```

```
## Spending
##     0     1     2
## 0.313 0.455 0.232
```

The above table shows that nearly half of the observations have outcome = 1. This is not a desirable behavior, especially if the researcher cares about prediction.

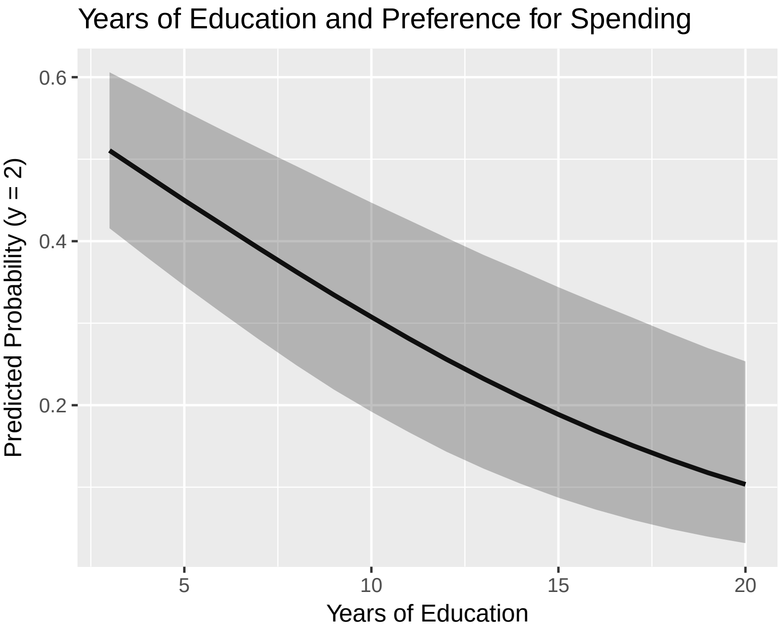
3.

```r
# Get the estimated coefficients and the variance-covariance matrix
beta_mu <- c(-m$coefficients, m$zeta)
beta_V <- solve(m$Hessian) # vcov = (H)^(-1)
# Simulate a sample of coefficients
N_sim <- 10000
beta_sim <- MASS::mvrnorm(N_sim, mu = beta_mu, Sigma = beta_V)

# Get a sequence of eduyrs levels
eduyrs_seq <- seq(min(d$eduyrs), max(d$eduyrs))
# Construct x for prediction
`x_new0|1` = cbind(female = mean(d$female), age = mean(d$age),
              eduyrs = eduyrs_seq,
              conserv = mean(d$conserv), `0|1` = 1, `1|2` = 0)
`z0|1` <- `x_new0|1` %*% t(beta_sim)

`x_new1|2` = cbind(female = mean(d$female), age = mean(d$age),
              eduyrs = eduyrs_seq,
              conserv = mean(d$conserv), `0|1` = 0, `1|2` = 1)
`z1|2` <- `x_new1|2` %*% t(beta_sim)

# Below I obtain a sample of predicted probabilities
# This part is critical. Make sure you understand it.
prob <- list()
prob[["y=0"]] <- t(apply(`z0|1`, 1, function(x) pnorm(x)))
prob[["y=2"]] <- t(apply(`z1|2`, 1, function(x) 1 - pnorm(x)))
```

```
prob[["y=1"]] <- 1 - prob[["y=0"]] - prob[["y=2"]]

# Below I obtain the 95% predictive intervals
prob_ci <- list()
prob_ci[["y=0"]] <- t(apply(prob[["y=0"]], 1, function(x) quantile(x, c(.025, .5, .975))))
prob_ci[["y=1"]] <- t(apply(prob[["y=1"]], 1, function(x) quantile(x, c(.025, .5, .975))))
prob_ci[["y=2"]] <- t(apply(prob[["y=2"]], 1, function(x) quantile(x, c(.025, .5, .975))))

# Merge the CI for plotting
prob_ci_d <- as.data.frame(do.call(rbind, prob_ci))
prob_ci_d$eduyrs <- rep(eduyrs_seq, times = 3)
prob_ci_d$spend <- rep(factor(c(0, 1, 2)), each = length(eduyrs_seq))
```

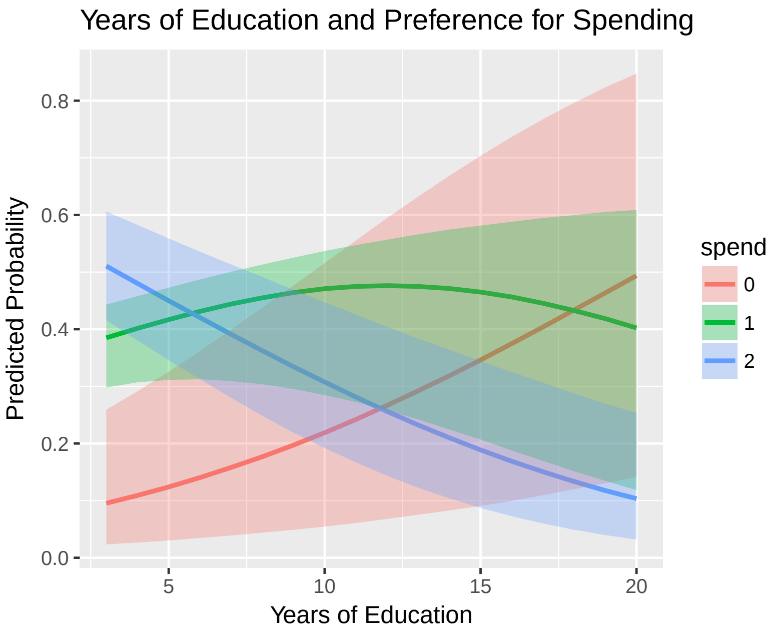
Required: Below is a plot of the predicted probability preferring spending level 2 $Pr(y_i = 2)$ as a function of years of education.

```
ggplot(subset(prob_ci_d, spend==2), aes(x=eduyrs, y=`50%`)) + geom_line(lwd=1) +
  geom_ribbon(aes(ymin=`2.5%`, ymax=`97.5%`), alpha=.3) +
  xlab("Years of Education") + ylab("Predicted Probability (y = 2)") +
  ggtitle("Years of Education and Preference for Spending")
```



Not required, just FYI: Below I plot the predicted probability of a respondent choosing all three spending levels as a function of years of education.

```
ggplot(prob_ci_d, aes(x=eduyrs, y=`50%`, group=spend)) + geom_line(aes(color=spend),lwd=1) +
  geom_ribbon(aes(ymin=`2.5%`, ymax=`97.5%`, fill = spend), alpha=.3) +
  xlab("Years of Education") + ylab("Predicted Probability") +
  ggtitle("Years of Education and Preference for Spending")
```



4.

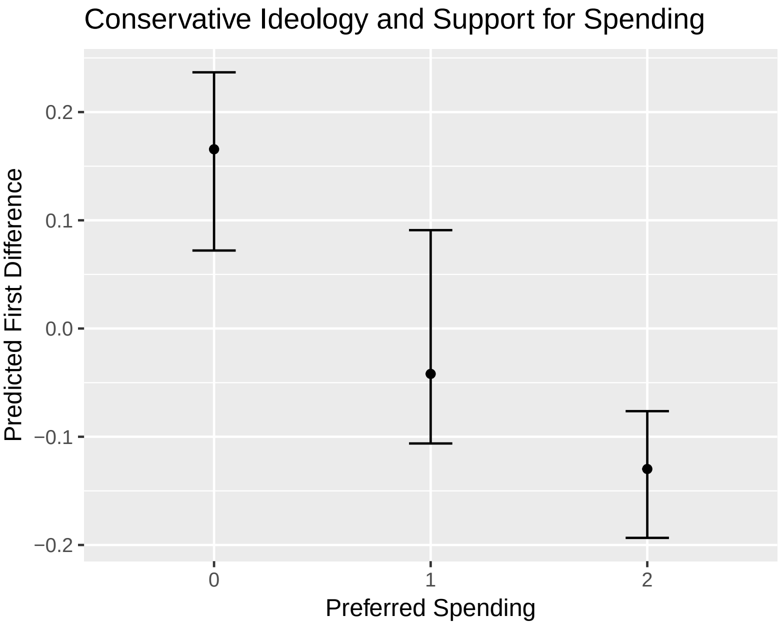
Just make some minor change to the chunk of code I used for Question 3. Now the varying variable is `conserv`.

```
# Get a sequence of conserv levels
conserv_seq <- c(0, 1)
# Construct x for prediction
`x_new0|1` = cbind(female = mean(d$female), age = mean(d$age),
              eduyrs = mean(d$eduyrs),
              conserv = conserv_seq, `0|1` = 1, `1|2` = 0)
`z0|1` <- `x_new0|1` %*% t(beta_sim)

`x_new1|2` = cbind(female = mean(d$female), age = mean(d$age),
              eduyrs = mean(d$eduyrs),
              conserv = conserv_seq, `0|1` = 0, `1|2` = 1)
`z1|2` <- `x_new1|2` %*% t(beta_sim)

# Below I obtain a sample of predicted probabilities
# This part is critical. Make sure you understand it.
prob <- list()
prob[["y=0"]] <- t(apply(`z0|1`, 1, function(x) pnorm(x)))
prob[["y=2"]] <- t(apply(`z1|2`, 1, function(x) 1 - pnorm(x)))
prob[["y=1"]] <- 1 - prob[["y=0"]] - prob[["y=2"]]

# First difference
first_diff <- matrix(ncol=3, nrow=N_sim)
first_diff[,1] <- prob[["y=0"]][2,] - prob[["y=0"]][1,]
first_diff[,2] <- prob[["y=1"]][2,] - prob[["y=1"]][1,]
first_diff[,3] <- prob[["y=2"]][2,] - prob[["y=2"]][1,]
first_diff_ci <- as.data.frame(
  t(apply(first_diff, 2, function(x) quantile(x, c(.025, .5, .975)))))
first_diff_ci$spend = ordered(c(0, 1, 2))
first_diff_ci
```

```
##          2.5%         50%       97.5% spend
## 1  0.07209242  0.16564305  0.23673071     0
## 2 -0.10619426 -0.04190169  0.09091022     1
## 3 -0.19342088 -0.12978429 -0.07635295     2
```

The first difference of being ideologically conservative (holding all else at the mean) is shown in the following figure.

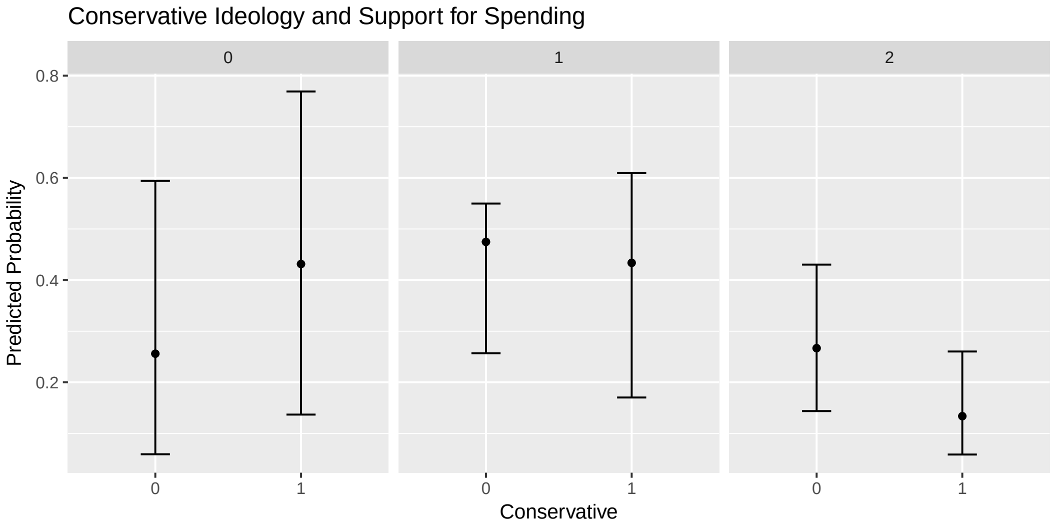
```
ggplot(first_diff_ci, aes(x=spend, y=`50%`)) + geom_point() +
  geom_errorbar(aes(ymin=`2.5%`, ymax=`97.5%`), width=.2) +
  xlab("Preferred Spending") + ylab("Predicted First Difference") +
  ggtitle("Conservative Ideology and Support for Spending")
```



Interpretation: The graph shows that the ideologically conservative has a significantly higher probability (i.e. 95% confidence interval does not cross 0) of supporting neither spending compared to the non-conservative. Also the ideologically conservative has a significantly lower probability of supporting spending on both. There is no significant difference between the conservative and the non-conservative in their probability supporting one item.

Not required, just FYI: We can also plot the predicted probabilities of preferring the 3 spending levels as a function of ideological conservatism. Interpretation of this graph would also be be helpful. This graph provide information about the variation of the predicted outcome as a function of the independent variable of interest, as well as the predicted outcomes themselves.

```
# Below I obtain the 95% predictive intervals
prob_ci <- list()
prob_ci[["y=0"]] <- t(apply(prob[["y=0"]], 1, function(x) quantile(x, c(.025, .5, .975))))
prob_ci[["y=1"]] <- t(apply(prob[["y=1"]], 1, function(x) quantile(x, c(.025, .5, .975))))
prob_ci[["y=2"]] <- t(apply(prob[["y=2"]], 1, function(x) quantile(x, c(.025, .5, .975))))

# Merge the CI for plotting
prob_ci_d <- as.data.frame(do.call(rbind, prob_ci))
prob_ci_d$conserv <- as.factor(rep(conserv_seq, times = 3))
prob_ci_d$spend <- rep(factor(c(0, 1, 2)), each = length(conserv_seq))

ggplot(prob_ci_d, aes(x=conserv, y=`50%`)) + geom_point() +
  geom_errorbar(aes(ymin=`2.5%`, ymax=`97.5%`), width=.2) +
  facet_grid(.~spend) +
  xlab("Conservative") + ylab("Predicted Probability") +
  ggtitle("Conservative Ideology and Support for Spending")
```


Conservative Ideology and Support for Spending
0
1
2
Predicted Probability
0.8
0.6
0.4
0.2
0
1
0
1
0
1
Conservative

Assignment 3

MNL and CL models

Individual-level independent variables may have different effect on the odds (or probability) of realizing one alternative versus another. For example, in a study of voting behavior in a multi-party system, the individual-level independent variable `age` is expected to be positively associated with the odds of voting for a left-wing party against a moderate party ($\beta_{age,\text{left}|\text{moderate}} > 0$). But it is expected to be negatively associated with the odds of voting for a right-wing party against a moderate party ($\beta_{age,\text{right}|\text{moderate}} < 0$). Hence we need alternative-specific coefficients for individuel-level independent variables. [I thank Hongshen Zhu for contributing this example.]

Vote choice in a multi-party system

```
rm(list=ls())
# Load the data
load("BESdata.Rdata")
# Load libraries
library(mlogit)
library(VGAM)
library(dplyr)
library(stargazer)
```

1

I fit a multinomial logit model using the `VGAM` package. I set `Brown` as the refereence level. Below I summarize the fitted model and interpret the coefficients of interest.

```
m_union <- vglm(vote ~ union + age + gender, data = BESdata,
                family = multinomial(refLevel = "Brown"))
summary(m_union)
```

```
##
## Call:
## vglm(formula = vote ~ union + age + gender, family = multinomial(refLevel = "Brown"),
##     data = BESdata)
##
##
## Pearson residuals:
##                        Min      1Q  Median      3Q   Max
## log(mu[,2]/mu[,1]) -1.295 -1.1222 -0.4550  0.9635 1.316
## log(mu[,3]/mu[,1]) -0.851 -0.7488 -0.2107 -0.1683 2.630
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  0.226341   0.270814   0.836   0.4033
## (Intercept):2 -0.349499   0.352008  -0.993   0.3208
## union:1       -0.415396   0.186180  -2.231   0.0257 *
## union:2       -0.016143   0.234405  -0.069   0.9451
## age:1          0.004055   0.004591   0.883   0.3770
## age:2         -0.007094   0.006158  -1.152   0.2493
## gender:1       0.004786   0.156380   0.031   0.9756
## gender:2       0.028228   0.207471   0.136   0.8918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Number of linear predictors:  2
##
## Names of linear predictors: log(mu[,2]/mu[,1]), log(mu[,3]/mu[,1])
##
## Residual deviance: 1668.075 on 1640 degrees of freedom
##
## Log-likelihood: -834.0376 on 1640 degrees of freedom
##
## Number of iterations: 4
##
## Reference group is level  1  of the response
```

Interpretation: Controlling `age` and `gender`,

- A respondent with union membership has $e^{-0.415396} = 0.66$ times the odds voting for Cameron against voting for Brown;
- A respondent with union membership has $e^{-0.016143} = 0.984$ times the odds voting for Clegg against voting for Brown.

2.

```r
# ------------------------------
# transform a dataset for mlogit
# ------------------------------
# Create an ID variable
BESdata$id <- 1:nrow(BESdata)
# Create an identifiers for the long-form data
identifiers <- data.frame(id = rep(1:nrow(BESdata), times=1, each=3),
                          candidate = rep(levels(BESdata$vote),
                                          times=nrow(BESdata), each=1))
d_t <- merge(identifiers, BESdata, by="id")
# create the alternative characteristics
d_t$approval <- apply(d_t, 1, function(x) x[paste0("app", x["candidate"])])
d_t$approval <- as.numeric(d_t$approval)
d_t$choice <- as.numeric(d_t$candidate == d_t$vote)
# Transform the model into mlogit input
d_final <- mlogit.data(d_t, shape = "long", choice = "choice",
                   alt.var = "candidate", id = "id")
```

Models: I fit two models. Model `m0` includes only the approval rating as the predictor for vote choice. Model `m1` includes approval rate and individual-level covariates *union membership*, *gender age* and *income* as predictors. Below I summarize the two models.

```r
m0 <- mlogit(choice ~ approval, data = d_final)
m1 <- mlogit(choice ~ approval | union + gender + age + income, data = d_final)
```

```r
stargazer(m0, m1, header = F)
```

Conclusion: Approval rating is sigifnicantly positively associated with vote choice. The conclusion does NOT change when I account for individual-level characteristics.

3.

I calculate the variation of predicted probability of voting for each of the three candidates as a function of approval ratings using simulation methods.

Table 1:

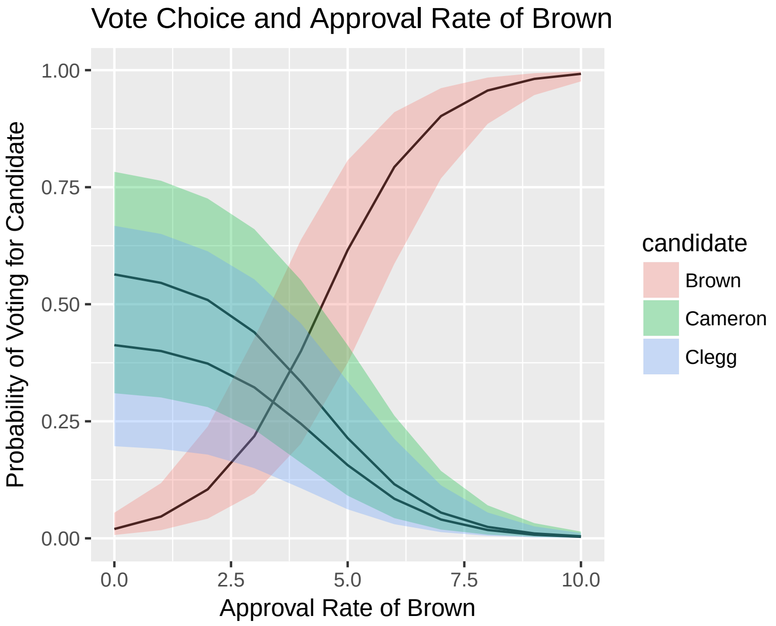
	Dependent variable:	
	choice	
	(1)	(2)
Cameron:(intercept)	0.037	-1.421^{***}
	(0.128)	(0.551)
Clegg:(intercept)	-0.499^{***}	-1.154^{**}
	(0.137)	(0.580)
approval	0.869^{***}	0.873^{***}
	(0.050)	(0.051)
Cameron:union		-0.031
		(0.296)
Clegg:union		0.037
		(0.308)
Cameron:gender		-0.043
		(0.261)
Clegg:gender		-0.009
		(0.276)
Cameron:age		0.021^{**}
		(0.008)
Clegg:age		0.011
		(0.009)
Cameron:income		0.067^{*}
		(0.035)
Clegg:income		0.018
		(0.037)
Observations	824	824
R^2	0.508	0.513
Log Likelihood	-413.209	-408.669
LR Test	852.395^{***} (df = 3)	861.476^{***} (df = 11)
Note:		*p<0.1; **p<0.05; ***p<0.01

```r
# Simulation of coefficients
# --------------------------
n_sim <- 10000
beta_m <- m1$coefficients
beta_vcov <- solve(-m1$hessian)
beta_sim <- MASS::mvrnorm(n_sim, mu = beta_m, Sigma = beta_vcov)
```

```r
# Below is a function to plot predictive probability
# ----------------------------------------------------
plot_pred_prob <- function(candidate = "Brown"){
  require(dplyr); require(ggplot2)
  # Get means of indepdnent variables (what to hold constant)
  m_union <- mean(BESdata$union)
  m_gender <- mean(BESdata$gender)
  m_age <- mean(BESdata$age)
  m_income <- mean(BESdata$income)
  m_app_brown <- rep(mean(BESdata$appBrown), 11)
  m_app_cameron <- rep(mean(BESdata$appCameron), 11)
  m_app_clegg <- rep(mean(BESdata$appClegg), 11)
  X <- list()
  X[["Brown"]] <- cbind(`Cameron:(intercept)` = 0, `Clegg:(intercept)` = 0,
                   `approval` = m_app_brown,
                   `Cameron:union` = m_union, `Clegg:union` = m_union,
                   `Cameron:gender` = m_gender, `Clegg:gender` = m_gender,
                   `Cameron:age` = m_age, `Clegg:age` = m_age,
                   `Cameron:income` = m_income, `Clegg:income` = m_income)
  X[["Cameron"]] <- cbind(`Cameron:(intercept)` = 1, `Clegg:(intercept)` = 0,
                     `approval` = m_app_cameron,
                     `Cameron:union` = m_union, `Clegg:union` = m_union,
                     `Cameron:gender` = m_gender, `Clegg:gender` = m_gender,
                     `Cameron:age` = m_age, `Clegg:age` = m_age,
                     `Cameron:income` = m_income, `Clegg:income` = m_income)
  X[["Clegg"]] <- cbind(`Cameron:(intercept)` = 0, `Clegg:(intercept)` = 1,
                   `approval` = m_app_clegg,
                   `Cameron:union` = m_union, `Clegg:union` = m_union,
                   `Cameron:gender` = m_gender, `Clegg:gender` = m_gender,
                   `Cameron:age` = m_age, `Clegg:age` = m_age,
                   `Cameron:income` = m_income, `Clegg:income` = m_income)
  # Substitute the approval rate of the candidate of interest by a sequence 0-10
  # This is the only thing that varies across the three plots
  X[[candidate]][, "approval"] <- 0:10
  # Calculate Odds
  odds <- lapply(X, function(x) exp(x %*% t(beta_sim)))
  odds_d <- do.call(rbind.data.frame, odds)
  odds_d$app <- rep(0:10, times = 3)
  odds_d$candidate <- rep(c("Brown", "Cameron", "Clegg"), each=11)
  odds_d2 <- reshape2::melt(odds_d, id.vars = c("app", "candidate"))
  # Get 95% predictive interval
  pred <- odds_d2 %>% group_by(variable, app) %>%
    mutate(prob = value / sum(value)) %>% ungroup %>%
    group_by(app, candidate) %>%
    summarise(`prob_lo` = quantile(prob, 0.025),
              `prob_mu` = quantile(prob, 0.5),
              `prob_hi` = quantile(prob, 0.975)) %>% as.data.frame
  # Plot
  ggplot(pred, aes(x = app, y = prob_mu, group = candidate)) +
    geom_line() + geom_ribbon(aes(ymin=prob_lo, ymax=prob_hi, fill=candidate), alpha=.3) +
    xlab(paste0("Approval Rate of ", candidate)) + ylab("Probability of Voting for Candidate") +
    ggtitle(paste0("Vote Choice and Approval Rate of ", candidate))
}
```

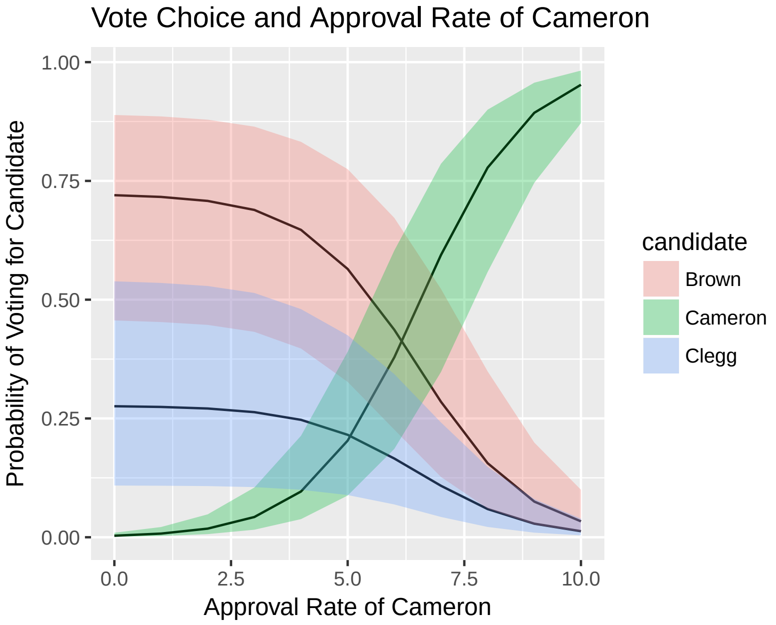
How Approval Rate of *Brown* Affects Vote Choice

```
# Predicted Prob. Voting for three candidates cond. on Brown's approval rate
plot_pred_prob("Brown")
```



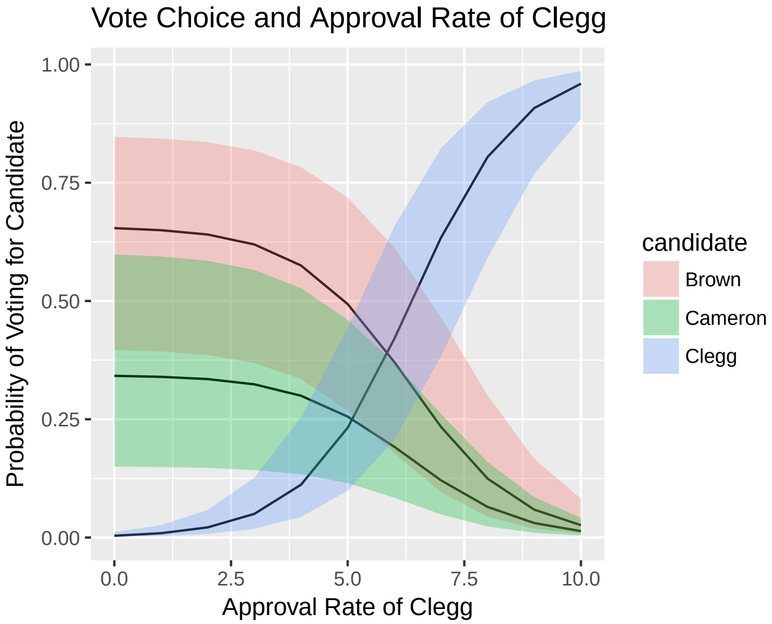
How Approval Rate of *Cameron* Affects Vote Choice

```
# Predicted Prob. Voting for three candidates cond. on Cameron's approval rate
plot_pred_prob("Cameron")
```



How Approval Rate of *Clegg* Affects Vote Choice

```
# Predicted Prob. Voting for three candidates cond. on Clegg's approval rate
plot_pred_prob("Clegg")
```



Grading Rubics: You don't need all three graphs to get full credit. What's important is showing you understand what to hold constant and what varies.