

# Lab 5: Generalized Linear Regression

*Hao Wang*

*March 09, 2017*

## 1. Load essential packages

```
library(VGAM)
library(car)
library(MASS)
library(effects)
library(nnet) #neural net package, which contains multinomial function
library(ggplot2)
library(Zelig)
library(ZeligChoice)
```

## 2. Binary Dependent Variable

When our dependent variable is binary, OLS is not applicable due to the violation of the residual variance. Generally we have two different models: probit and logit. They share the same idea: coerce the predicted y values into a continuous distribution ranging from 0 to 1.

### 2.1 Logistic Regression

The basic idea of logistic regression is to make a logistic transformation of our linear function.

#### 2.1.a Logistic Function

The logistic function is

$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

and

$$y = XB = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 \dots$$

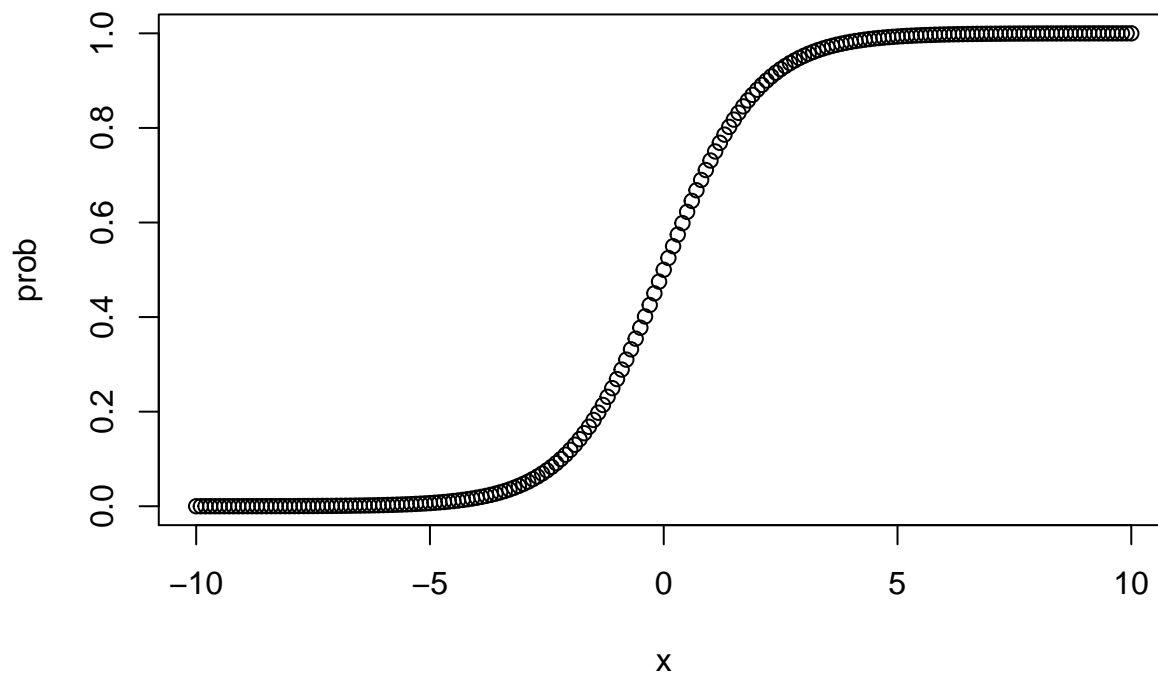
Apply logistic function to the linear combination y, we got the prediction (known as the ‘predicted probability’ of the logistic regression)

$$Y = \sigma(y) = F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 \dots)}}$$

With the logistic transformation, the outcome has a range of (0,1), we can simulate its outcome.

```
x <- seq(-10, 10, by =0.1)
prob <- 1/(1+ exp(-x))
plot(x, prob, type = "b", main = "Logistic Function")
```

## Logistic Function



### 2.1.b Logistic Regression R Codes

Use gender as an example, we can interpret the regression result with odds ratio. Odds ratio: the ratio of the probability the event occurs to the probability it does not occur

$$odds = \frac{p}{1 - p}$$

In this example the odds ratio percentage change is -21%, which means on average the odds of being a volunteer will diminish by 21% if the subject is male.

We can also get the predicted probability by the predict function.

```
#Load Cowles data from the effects package
```

```
#Explanation
```

```
help(Cowles)
```

```
## Help on topic 'Cowles' was found in the following packages:
```

```
##
```

```
##   Package           Library
```

```
##   effects           /Library/Frameworks/R.framework/Versions/3.3/Resources/library
```

```
##   car               /Library/Frameworks/R.framework/Versions/3.3/Resources/library
```

```
##
```

```
##
```

```
## Using the first match ...
```

```
mydata <- Cowles
```

```
summary(mydata)
```

```
##   neuroticism   extraversion      sex    volunteer
```

```
##   Min.    : 0.00   Min.    : 2.00  female:780  no :824
```

```
##   1st Qu.: 8.00   1st Qu.:10.00  male  :641  yes:597
```

```

## Median :11.00   Median :13.00
## Mean   :11.47   Mean   :12.37
## 3rd Qu.:15.00   3rd Qu.:15.00
## Max.   :24.00   Max.   :23.00

logit <- glm(formula = volunteer ~ neuroticism + sex + extraversion,
             data = mydata, family = binomial(link = "logit"))
summary(logit)

##
## Call:
## glm(formula = volunteer ~ neuroticism + sex + extraversion, family = binomial(link = "logit"),
##      data = mydata)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3977  -1.0454  -0.9084   1.2601   1.6849
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.116496   0.249057  -4.483 7.36e-06 ***
## neuroticism    0.006362   0.011357   0.560  0.5754
## sexmale       -0.235161   0.111185  -2.115  0.0344 *
## extraversion   0.066325   0.014260   4.651 3.30e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1933.5  on 1420  degrees of freedom
## Residual deviance: 1906.1  on 1417  degrees of freedom
## AIC: 1914.1
##
## Number of Fisher Scoring iterations: 4

#get odds ratio
100*(exp(logit$coefficients[-1])-1)

##      neuroticism      sexmale extraversion
##      0.638215      -20.955677      6.857438

#prediction of a certain point
new <- data.frame(sex = "male", extraversion = 10, neuroticism = 10)
predict(logit, newdata = new, type = 'response')

##           1
## 0.348694

#we can also calculate the AIC by hand
#the formula is deviance + 2*(p+1), while p is the number of variables
aic <- logit$deviance+ 2*(4)
aic

## [1] 1914.061

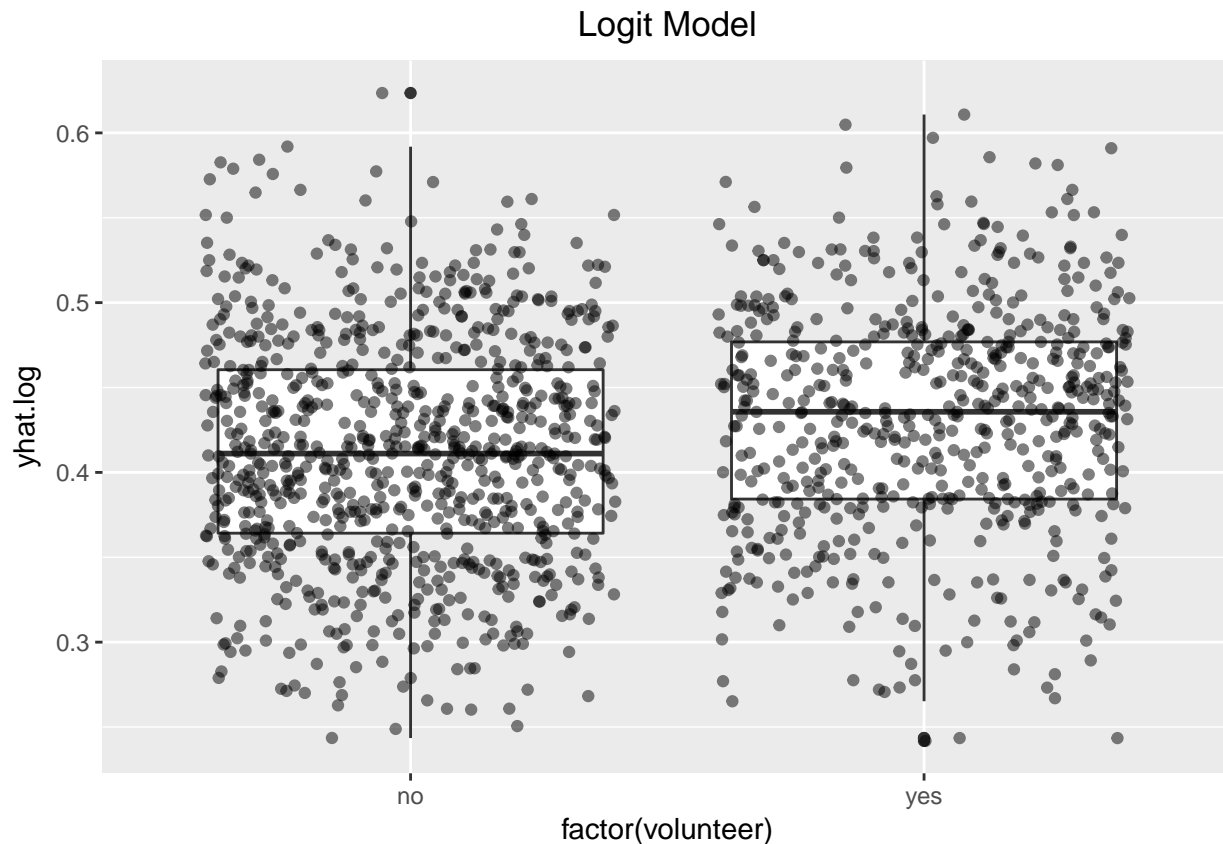
```

- Lab Practice 1: based on the logistic function, can you get the predicted outcome with the coefficients given by the summary table?

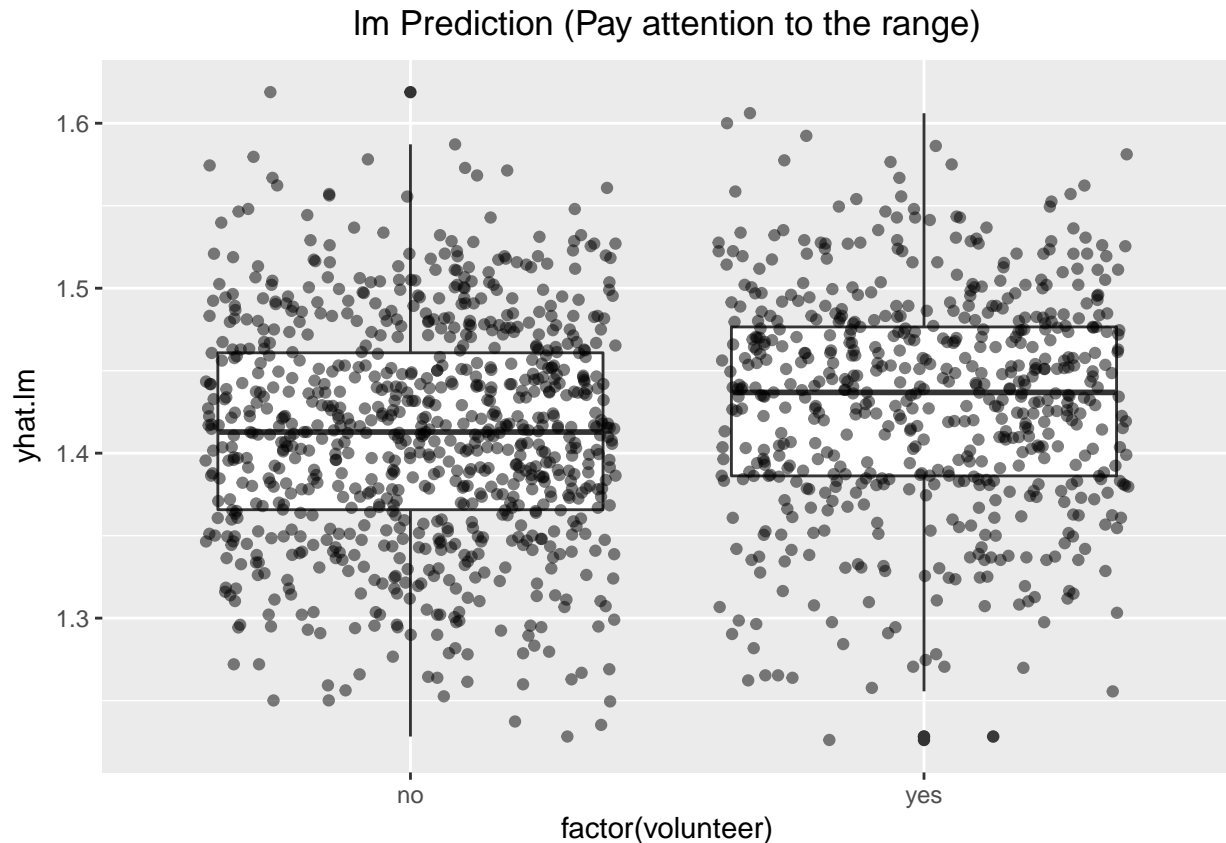
```
#we can compute the prediction by hand
y <- logit$coefficients[1]+ logit$coefficients[2]*10 + logit$coefficients[3]*1 + logit$coefficients[4]*
#Then apply it to the logistic function
#logistic funtion is???
```

### 2.1.c Compare the results between logit and regular lm

```
yhat.log <- predict(logit, data=mydata, type = 'response')
lm <- lm(as.numeric(volunteer) ~ neuroticism + sex + extraversion, data = mydata)
yhat.lm <- predict(lm, data=mydata)
mydata$yhat.log <- yhat.log
mydata$yhat.lm <- yhat.lm
ggplot(data = mydata, aes(factor(volunteer), yhat.log)) +
  geom_boxplot() +
  ggtitle("Logit Model") +
  geom_jitter(alpha = .5) +
  theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot(data = mydata, aes(factor(volunteer), yhat.lm)) +
  geom_boxplot() +
  geom_jitter(alpha = .5) +
  ggtitle("lm Prediction (Pay attention to the range)") +
  theme(plot.title = element_text(hjust = 0.5))
```



## 2.2 Probit Model

Probit model share the similar idea with logit model, but in the probit case we are doing a gaussian transformation (the CDF funtion of a standard normal distribution).

### 2.2.a Probit Function

It's simply the CDF funtion of a standard normal distribution

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-t^2/2} dt$$

And y is the linear combination

$$y = XB = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 \dots$$

```
probit <- glm(formula = volunteer ~ neuroticism + sex + extraversion,
              data = mydata, family = binomial(link = "probit"))
summary(probit)
```

```
##
## Call:
## glm(formula = volunteer ~ neuroticism + sex + extraversion, family = binomial(link = "probit"),
##      data = mydata)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.3947 -1.0468 -0.9091 1.2611 1.6901
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.692584 0.153670 -4.507 6.58e-06 ***
## neuroticism 0.004104 0.007044 0.583 0.5601
## sexmale     -0.145677 0.068957 -2.113 0.0346 *
## extraversion 0.040934 0.008790 4.657 3.21e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1933.5 on 1420 degrees of freedom
## Residual deviance: 1906.1 on 1417 degrees of freedom
## AIC: 1914.1
##
## Number of Fisher Scoring iterations: 4
#prediction of a certain point
new <- data.frame(sex = "male", extraversion = 10, neuroticism =10)
predict(probit, newdata = new, type = 'response')

##          1
## 0.3490514
```

- Lab Practice 2: pick another observation (you need to set values for sex, extraversion and neuroticism), compare the results of logit model and probit model.

### 3. Ordered Model

When our DV is an ordered/ranked outcome, such as ‘bad’, ‘ok’, ‘good’; or ‘unlikely’, ‘somewhat likely’, ‘very likely’, we need to use ordered model.

```
mydata <- WVS
summary(mydata)
```

```
##      poverty      religion      degree      country      age
## Too Little :2708    no : 786    no :4238    Australia:1874    Min.   :18.00
## About Right:1862    yes:4595    yes:1143    Norway   :1127    1st Qu.:31.00
## Too Much   : 811                      Sweden   :1003    Median  :43.00
##                                     USA      :1377    Mean    :45.04
##                                     3rd Qu.:58.00
##                                     Max.    :92.00
##
##      gender
## female:2725
## male  :2656
##
##
##
```

```
help(WVS)
#convert our DV to an ordered factor
mydata$poverty <- ordered(as.factor(mydata$poverty))
```

```
#change reference level to USA
mydata$country <- relevel(mydata$country, ref="USA")
#overview
table(mydata$poverty, mydata$country)
```

```
##
##           USA Australia Norway Sweden
## Too Little 555         952   597   604
## About Right 372         622   494   374
## Too Much   450         300    36    25
```

### 3.1 Ordered Logit

```
# the function y~. means regressing y on the rest variables
ologit <- polr(poverty ~., method = 'logistic', data = mydata)
summary(ologit)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = poverty ~ ., data = mydata, method = "logistic")
##
## Coefficients:
##              Value Std. Error t value
## religionyes      0.17973   0.077346   2.324
## degreeyes        0.14092   0.066193   2.129
## countryAustralia -0.61778   0.070665  -8.742
## countryNorway    -0.94012   0.078468 -11.981
## countrySweden    -1.22107   0.083957 -14.544
## age               0.01114   0.001561   7.139
## gendermale        0.17637   0.052972   3.329
##
## Intercepts:
##              Value      Std. Error t value
## Too Little|About Right  0.1120    0.1134    0.9872
## About Right|Too Much   1.9147    0.1170   16.3719
##
## Residual Deviance: 10402.59
## AIC: 10420.59
```

#### 3.1.b Ologit Interpretation

(Also see PA book p.114) Standard interpretation of the ordered logit coefficient is that for a one unit increase in the predictor, the response variable level is expected to change by its respective regression coefficient in the ordered log-odds scale while the other variables in the model are held constant. For instance, the religious ‘YES’ will increase the response variable level by an ordered log-odds scale of 0.179.

We can also interpret it in terms of odds ratio. One unit increase of age, the odds that they will report ‘too little work’ relative to any higher categories will increase by 1.12.

```
100*(exp((ologit$coefficients)) -1)
```

```
## religionyes      degreeyes countryAustralia  countryNorway
```

```
##          19.689369          15.132879          -46.085741          -60.941921
##    countrySweden          age          gendermale
##          -70.508524          1.120323          19.287748
```

Prediction can be done in a similar way with logit model.

```
new <- data.frame(country = "Norway", religion = "no", age =50, degree="no", gender="female")
predict(ologit, newdata = new, type = "prob")
```

```
## Too Little About Right    Too Much
## 0.62129816 0.28739457 0.09130727
```

## 4. Introduction of Zelig Project.

Website: [click here](http://www.zeligproject.org/) Zelig allows a unified grammar in estimating generalized linear models, besides it provides an easy way of simulating probability distributions.

### 4.1 Logit Model

```
mydata <- Cowles
zlogit <- zelig(volunteer ~neuroticism + sex + extraversion, model ="logit", data=mydata)
```

```
## How to cite this model in Zelig:
##   R Core Team. 2007.
##   logit: Logistic Regression for Dichotomous Dependent Variables
##   in Christine Choirat, Christopher Gandrud, James Honaker, Kosuke Imai, Gary King, and Olivia Lau,
##   "Zelig: Everyone's Statistical Software," http://zeligproject.org/
```

```
summary(zlogit)
```

```
## Model:
##
## Call:
## z5$zelig(formula = volunteer ~ neuroticism + sex + extraversion,
##   data = mydata)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3977  -1.0454  -0.9084   1.2601   1.6849
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.116496   0.249057  -4.483 7.36e-06
## neuroticism    0.006362   0.011357   0.560  0.5754
## sexmale       -0.235161   0.111185  -2.115  0.0344
## extraversion   0.066325   0.014260   4.651 3.30e-06
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1933.5  on 1420  degrees of freedom
## Residual deviance: 1906.1  on 1417  degrees of freedom
## AIC: 1914.1
##
```

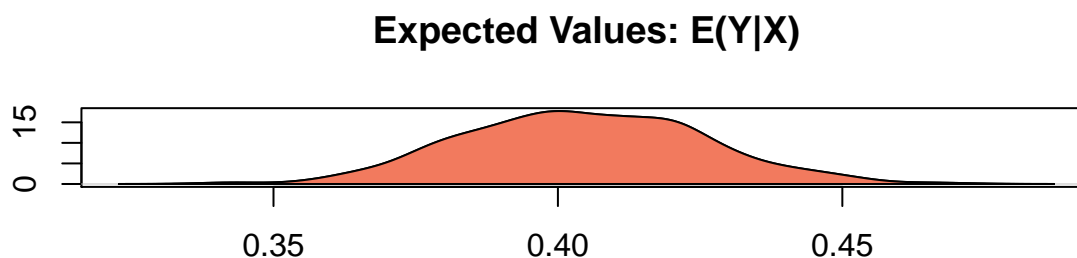
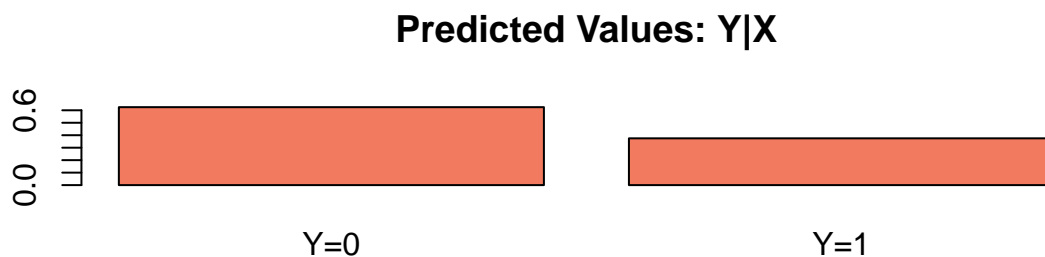


```
## Number of Fisher Scoring iterations: 4
##
## Next step: Use 'setx' method
```

We can simulate the results by `sim()` and `setx()`. (Stata package `clarify` uses the same command, if you're familiar with Stata)

```
x.zlogit <- setx(zlogit, sex="female", neuroticism = 10, extraversion =10)
zlogit.out <- sim(zlogit, x=x.zlogit)
summary(zlogit.out)
```

```
##
## sim x :
## -----
## ev
##          mean          sd          50%          2.5%          97.5%
## [1,] 0.4044295 0.02099261 0.4040466 0.3649755 0.4462039
## pv
##          0          1
## [1,] 0.625 0.375
plot(zlogit.out)
```



Use the similar logit we can calculate the differences at lower level  $X$ s and higher level  $X$ s.

```
#when not specified, other variables are held at mean level
x.high <- setx(zlogit, neuroticism = quantile(mydata$neuroticism, prob = 0.75), sex="male")
x.low <- setx(zlogit, neuroticism = quantile(mydata$neuroticism, prob = 0.25), sex="male")
s.out2 <- sim(zlogit, x = x.high, x1 = x.low)
summary(s.out2)
```

```
##
## sim x :
## -----
## ev
```

```
##           mean          sd        50%        2.5%        97.5%
## [1,] 0.3937042 0.02287278 0.3927433 0.3504786 0.4373471
## pv
##           0          1
## [1,] 0.634 0.366
##
## sim x1 :
## -----
## ev
##           mean          sd        50%        2.5%        97.5%
## [1,] 0.3822408 0.02009027 0.382393 0.3419874 0.4200692
## pv
##           0          1
## [1,] 0.628 0.372
## fd
##           mean          sd        50%        2.5%        97.5%
## [1,] -0.01146343 0.01941646 -0.01128483 -0.04955465 0.02588242
#plot(s.out2)
```

## 4.2 Ordered Logit Model

```
#Load Data and make sure our DV is in ordered
mydata <- WVS
mydata$poverty <- factor(mydata$poverty, ordered = TRUE, levels = c("Too Little", "About Right", "Too Much"))
summary(mydata$poverty)

## Too Little About Right Too Much
##           2708           1862           811
#change reference level to USA
mydata$country <- relevel(mydata$country, ref="USA")
```

Apply the Zelig Function

```
z.ologit <- zelig(poverty ~ country + religion + age + degree, data=mydata, model="ologit")

## How to cite this model in Zelig:
## William N. Venables, and Brian D. Ripley. 2011.
## ologit: Ordinal Logit Regression for Ordered Categorical Dependent Variables
## in Christine Choirat, Christopher Gandrud, James Honaker, Kosuke Imai, Gary King, and Olivia Lau,
## "Zelig: Everyone's Statistical Software," http://zeligproject.org/
summary(z.ologit)

## Model:
## Call:
## z5$zelig(formula = poverty ~ country + religion + age + degree,
## data = mydata)
##
## Coefficients:
## Value Std. Error t value
## countryAustralia -0.61939 0.070641 -8.768
## countryNorway -0.93825 0.078449 -11.960
## countrySweden -1.21183 0.083853 -14.452
```

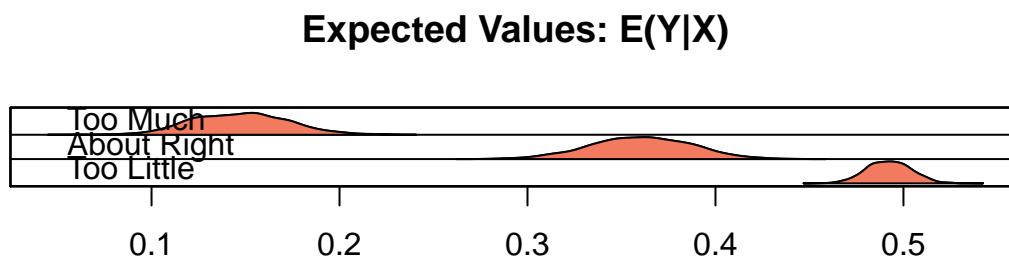
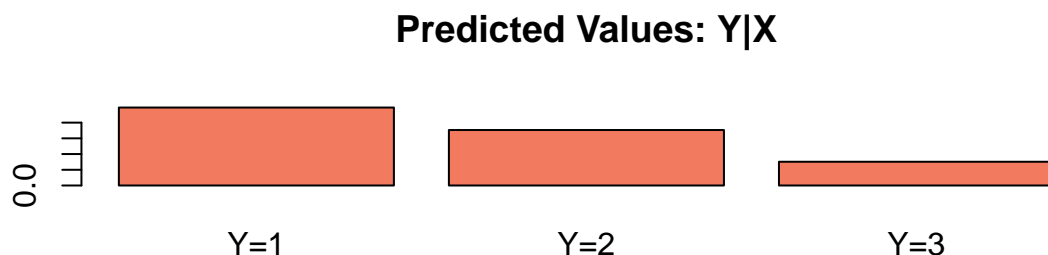
```
## religionyes      0.15614  0.076982  2.028
## age              0.01128  0.001559  7.233
## degreeeyes      0.13634  0.066156  2.061
##
## Intercepts:
##               Value      Std. Error t value
## Too Little|About Right  0.0109   0.1092   0.1002
## About Right|Too Much    1.8108   0.1125  16.0919
##
## Residual Deviance: 10413.69
## AIC: 10429.69
## Next step: Use 'setx' method
```

Set the explanatory variables to their observed values and simulate fitted values given x.out and view the results:

```
x.out <- setx(z.ologit)
s.out <- sim(z.ologit, x = x.out)
summary(s.out)
```

```
##
## sim x :
## -----
## ev
##               mean          sd      50%      2.5%      97.5%
## Too Little  0.4922523 0.01168408 0.4922182 0.4698153 0.5147063
## About Right 0.3604329 0.02638104 0.3605037 0.3093277 0.4099042
## Too Much    0.1473148 0.02512098 0.1474332 0.1021685 0.1981609
## pv
##      mean          sd 50% 2.5% 97.5%
## [1,] 1.655 0.7269825  2   1   3
```

```
plot(s.out)
```



## 4.3 Multinomial Logit Model

Multinomial logit model should be applied when our DV contains multiple nominal categories

We load the British Election Panel Study, in which the DV is the vote choice (Conservative, Labour and Liberal Democrat)

```
#Load British Election Panel Study
```

```
mydata <- BEPS  
summary(mydata$vote)
```

```
##      Conservative      Labour Liberal Democrat  
##           462           720           343
```

Apply zelig function of mlogit.

In the default model, the reference category is the last level and will be omitted. You can change the reference level in the way I did for countries in the previous sections. Essentially in multinomial model our software is estimating k-1 models, while k is the number of the categories of the DV. Therefore, since the parameter estimates are relative to the referent group, the standard interpretation of the multinomial logit is that for a unit change in the predictor variable, the logit of outcome m relative to the referent group is expected to change by its respective parameter estimate (which is in log-odds units) given the variables in the model are held constant.

```
z.out <- zelig(vote ~ age + economic.cond.national + economic.cond.household + gender + political.know
```

```
## How to cite this model in Zelig:
```

```
##   Thomas W. Yee. 2007.
```

```
##   mlogit: Multinomial Logistic Regression for Dependent Variables with Unordered Categorical Values  
##   in Christine Choirat, Christopher Gandrud, James Honaker, Kosuke Imai, Gary King, and Olivia Lau,  
##   "Zelig: Everyone's Statistical Software," http://zeligproject.org/
```

```
summary(z.out)
```

```
## Model:
```

```
##
```

```
## Call:
```

```
## z5$zelig(formula = vote ~ age + economic.cond.national + economic.cond.household +  
##         gender + political.knowledge, data = mydata)
```

```
##
```

```
##
```

```
## Pearson residuals:
```

```
##           Min      1Q  Median      3Q      Max  
## log(mu[,1]/mu[,3]) -1.911 -0.5491 -0.2180 0.8546 4.512  
## log(mu[,2]/mu[,3]) -2.958 -0.7857 -0.1919 0.8202 3.551
```

```
##
```

```
## Coefficients:
```

```
##           Estimate Std. Error z value Pr(>|z|)  
## (Intercept):1      0.741085   0.423821   1.749 0.080363  
## (Intercept):2     -1.078783   0.409783  -2.633 0.008474  
## age:1              0.017473   0.004734   3.691 0.000223  
## age:2             -0.002270   0.004356  -0.521 0.602293  
## economic.cond.national:1 -0.451722  0.090235 -5.006 5.56e-07  
## economic.cond.national:2  0.469289  0.085932  5.461 4.73e-08  
## economic.cond.household:1 -0.048323  0.083399 -0.579 0.562311  
## economic.cond.household:2  0.229512  0.077379  2.966 0.003016  
## gendermale:1       -0.121606   0.148046  -0.821 0.411412  
## gendermale:2        0.092787   0.136867   0.678 0.497813
```

```
## political.knowledge:1      0.092451   0.071525   1.293 0.196157
## political.knowledge:2     -0.272234   0.063992  -4.254 2.10e-05
##
## Number of linear predictors: 2
##
## Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])
##
## Residual deviance: 2940.786 on 3038 degrees of freedom
##
## Log-likelihood: -1470.393 on 3038 degrees of freedom
##
## Number of iterations: 4
##
## Reference group is level 3 of the response
## Next step: Use 'setx' method
```

We can also simulate the results in a similar way

```
x.weak <- setx(z.out, economic.cond.national = 1, economic.cond.household = 1, age =45,
               gender ="male", political.knowledge = 2)
x.strong <- setx(z.out, economic.cond.national = 4, economic.cond.household = 4, age =45,
                 gender ="male", political.knowledge = 2)
s.out.mlogit <- sim(z.out, x = x.strong, x1 = x.weak)
summary(s.out.mlogit)
```

```
##
## sim x :
## -----
## ev
##          mean          sd          50%          2.5%          97.5%
## Pr(Y=Conservative)    0.1368174 0.01496164 0.1362560 0.1095806 0.1670540
## Pr(Y=Labour)          0.6563546 0.02409758 0.6568802 0.6093518 0.7037439
## Pr(Y=Liberal Democrat) 0.2068280 0.01972305 0.2068086 0.1710345 0.2471762
## pv
##          1          2          3
## [1,] 0.13 0.651 0.219
##
## sim x1 :
## -----
## ev
##          mean          sd          50%          2.5%
## Pr(Y=Conservative)    0.67736104 0.04503782 0.68021448 0.5859017
## Pr(Y=Labour)          0.09200907 0.01807117 0.09090274 0.0609623
## Pr(Y=Liberal Democrat) 0.23062990 0.04032508 0.22644385 0.1597917
##          97.5%
## Pr(Y=Conservative)    0.7620905
## Pr(Y=Labour)          0.1321072
## Pr(Y=Liberal Democrat) 0.3134549
## pv
##          1          2          3
## [1,] 0.7 0.079 0.221
## fd
##          mean          sd          50%          2.5%
## Pr(Y=Conservative)    0.54054365 0.05050474 0.54637253 0.4370306
## Pr(Y=Labour)          -0.56434550 0.03288730 -0.56593387 -0.6232259
```

```
## Pr(Y=Liberal Democrat) 0.02380185 0.04743686 0.02080939 -0.0599146
##                               97.5%
## Pr(Y=Conservative)    0.6321641
## Pr(Y=Labour)          -0.4981722
## Pr(Y=Liberal Democrat) 0.1215493
```

```
plot(s.out)
```

