Draper HW #5-6

Matthew Draper

May 13, 2019

```
## Fit a probit mode describing the probability of voting for impeachment. Include AFL-CIO score and Clinton
's share of the 1996 vote in the Members' districts, and the interaction of these two in the model.

imp<-read.csv("impeach.csv")
imp$votesum[is.na(imp$votesum)] <- 0
imp$yea<-as.numeric(imp$votesum>0)

imp.glm<-glm(yea ~ clint96 + aflcio97,
family = binomial (link = "probit"), data = imp)
summary(imp.glm)
```

```
##
## Call:
## glm(formula = yea ~ clint96 + aflcio97, family = binomial(link = "probit"),
##     data = imp)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.27515  -0.06654   0.00273   0.00884   2.52051
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.440808   1.553579   3.502 0.000462 ***
## clint96     -0.029082   0.030825  -0.943 0.345453
## aflcio97    -0.066295   0.008016  -8.271  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 597.76  on 431  degrees of freedom
## Residual deviance:  75.39  on 429  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 81.39
##
## Number of Fisher Scoring iterations: 9
```

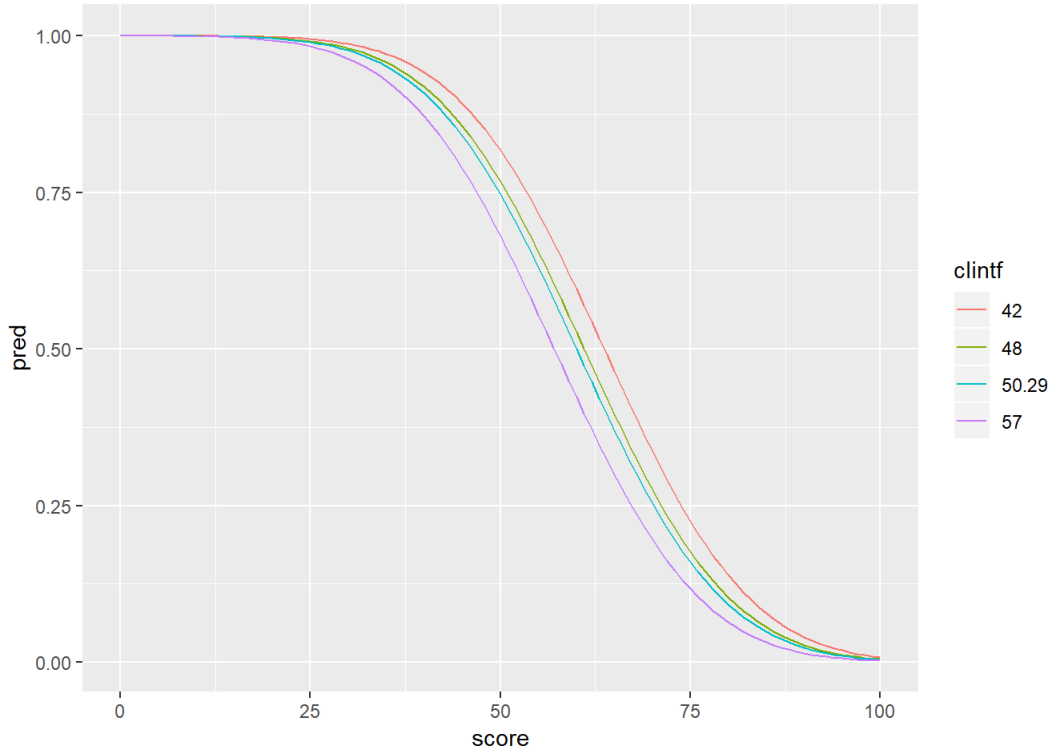
```
imp.glm2<-glm(yea ~ clint96 + aflcio97 + clint96:aflcio97,
family = binomial (link = "probit"), data = imp)
summary(imp.glm2)
```

```
##
## Call:
## glm(formula = yea ~ clint96 + aflcio97 + clint96:aflcio97, family = binomial(link = "probit"),
##     data = imp)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.24873  -0.07821   0.00019   0.00542   2.56818
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       8.4528676  5.3878969   1.569   0.1167
## clint96          -0.0890624  0.1066665  -0.835   0.4037
## aflcio97         -0.1057590  0.0641169  -1.649   0.0991 .
## clint96:aflcio97  0.0007816  0.0012472   0.627   0.5309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 597.758  on 431  degrees of freedom
## Residual deviance:  74.987  on 428  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 82.987
##
## Number of Fisher Scoring iterations: 9
```

```
## Using simulation methods, present a plot describing the in predicted probability of voting for impeachmen
t implied by a change of the AFL-CIO score from 0 to 100 as Clinton's vote share moves across its interquart
ile range.
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

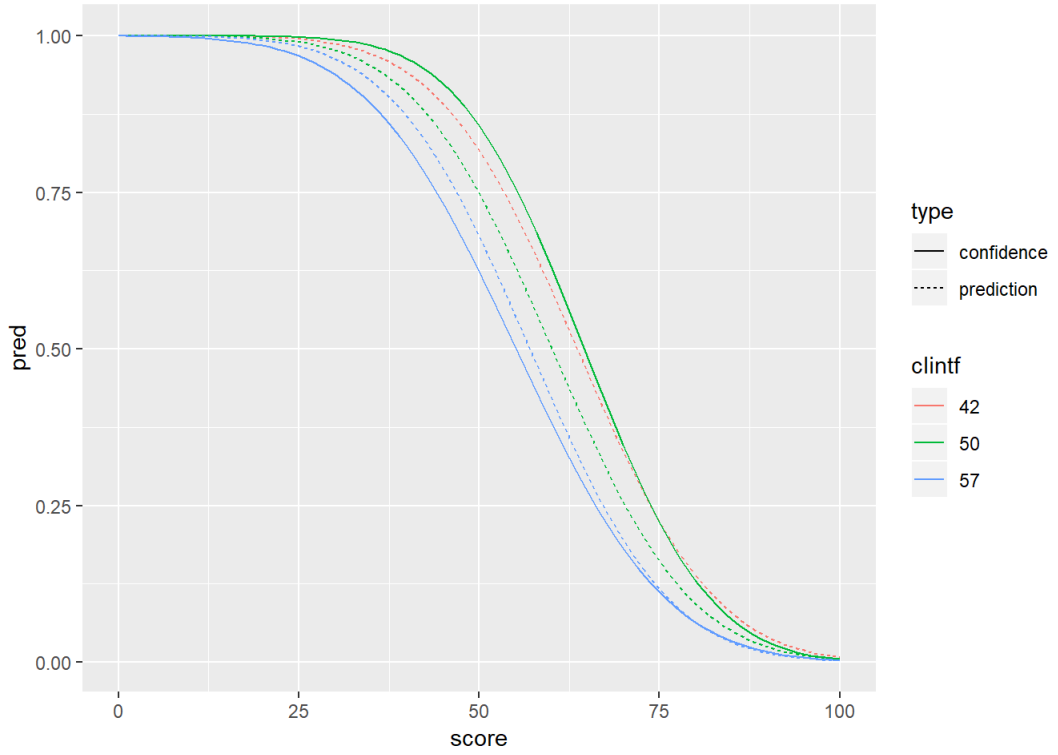
```
clint <-c(42,48,50.29,57)
predict<-matrix(1:101*length(clint),101,length(clint))
for (i in 1:4)
{score <- 0:100
predict[,i] <- predict(imp.glm,type='response',newdata=data.frame('clint96'=rep(clint[i],101),'aflcio97'=sco
re))}

data1<-data.frame(pred=c(predict[,1],predict[,2],predict[,3],predict[,4]),clintf=as.factor(c(rep(clint[1],10
1),rep(clint[2],101),rep(clint[3],101),rep(clint[4],101))),score=rep(0:100,4))

plot1<-ggplot(data1,aes(x=score,y=pred,colour=clintf))+geom_line(aes(group=clintf))
plot1
```




```
## Provide a visual display of the uncertainty around these changes.
library(ggplot2)
clint <-c(42,50,57)
predict<-matrix(1:101*length(clint),101,length(clint))
for (i in 1:3)
{score <- 0:100
predict[,i] <- predict(imp.glm,type='response',newdata=data.frame('clint96'=rep(clint[i],101),'aflcio97'=sco
re))}

data1<-data.frame(pred=c(predict[,1],predict[,2],predict[,3]),clintf=as.factor(c(rep(clint[1],101),rep(clint
[2],101),rep(clint[3],101)),score=rep(0:100,3)))
```

```
## Error in as.factor(c(rep(clint[1], 101), rep(clint[2], 101), rep(clint[3], : unused argument (score = rep
(0:100, 3))
```

```
datalow<-cbind(rep(1,101),rep(clint[1],101),0:100,clint[1]*(0:100))
datamiddle<-cbind(rep(1,101),rep(clint[2],101),0:100,clint[2]*(0:100))
datahigh<-cbind(rep(1,101),rep(clint[3],101),0:100,clint[3]*(0:100))

Betas<- coefficients(imp.glm2)
p.hat.low<-p.hat.hi<-p.hat.middle<-NULL
for(i in 1:nrow(datalow)){
    p.hat.low<-cbind(p.hat.low,pnorm(Betas%*%datalow[i,]))
    p.hat.middle<-cbind(p.hat.middle,pnorm(Betas%*%datalow[i,]))
    p.hat.hi<-cbind(p.hat.hi,pnorm(Betas%*%datahigh[i,]))
}

lo.mean<-apply(p.hat.low,2,mean)
hi.mean<-apply(p.hat.hi,2,mean)
lo.ci<-apply(p.hat.low,2,quantile,c(0.025,.975))
mid.ci<-apply(p.hat.middle,2,quantile,c(0.025,.975))
hi.ci<-apply(p.hat.hi,2,quantile,c(0.025,.975))

data1<-data.frame(pred=c(predict[,1],predict[,2],predict[,3],lo.ci[1,],mid.ci[1,],hi.ci[1,],lo.ci[2,],mid.ci
[2,],hi.ci[2,]),clintf=as.factor(rep(c(rep(clint[1],101),rep(clint[2],101),rep(clint[3],101)),3)),score=rep(
rep(0:100,3),3),type=c(rep('prediction',303),rep('confidence',606)))
data1$grp<-paste(data1$clintf,data1$type)

plot1<-ggplot(data1,aes(x=score,y=pred,colour=clintf))+geom_line(aes(group=grp,linetype=type))
plot1
```




```
## Decide which model performs better: one with the multiplicative interaction term included or one without.
Using some of the model selection heuristics covered in class, provide a short justification for your decisi
on.
```

We wish to select the model that minimizes information loss. This can be proxied by AIC, and the quantity exp((AICmin ??? AICi)/2) gives us the probability that the ith model minimizes information loss. The model without the interaction term (imp.glm) has an AIC of 81.39, and the model with the interaction term (imp.glm2) has an AIC of 82.99. We prefer the model with the lower AIC (and fewer parameters), though in this case they are very close.

```
## Run the following three lines of R code and explain your results.

income<-ordered(c("Mid","High","Low"))
## This line is creating a nominal variable with three bins, mid, high and low. Since there is an order to t
hese variables, we can fit an ordered logit. If we just had arbitrary category labels then we would have to
use a multinomial logit because the categories would lack a natural ordering.
income
```

```
## [1] Mid  High Low
## Levels: High < Low < Mid
```

```
## However, this code seems to be putting mid, high and low in alphabetical order. That is, High < Low < Mid
. This is giving the nominal variable an order, turning it into an ordinal variable (though not in the way we
would expect).
as.numeric(income)
```

```
## [1] 3 1 2
```

```
## This line is converting the ordinal variable income into a numeric variable with three values. We have as
signed a number to each category, though the order is not what we want.
```

```
## Using Drury's data, use an ordered logit regression to evaluate the success of economic sanctions (result
) as a function of the (log) GNP ratio (gnprat), amount of trade (trade), target GNP cost (tarcst), sender c
ost (cost), and whether there is a cooperative relationship (coop).

library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.5.3
```

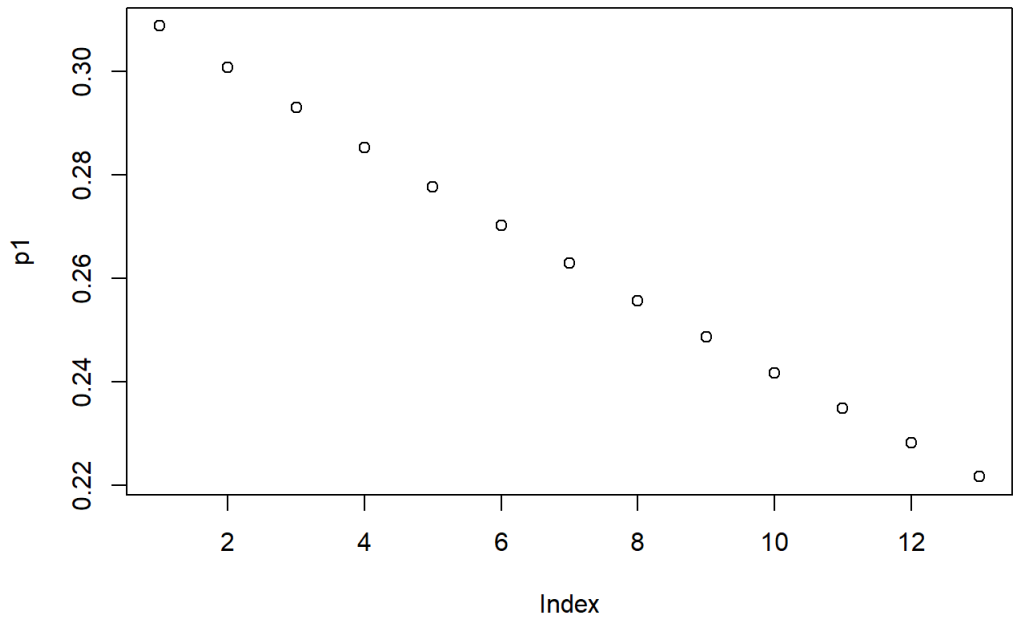
```
drury<-read.csv('drury_jpr_data.csv')
drury$loggnprat<-log(drury$gnprat)
dglm<-polr(as.ordered(result) ~ loggnprat + trade + tarcst + cost + coop, data = drury, method="logistic")
```

```
## Construct a counterfactual scenario for a particular variable, justify it, and analyze it in a one page w
rite up with text, graphic, and tabular information integrated. You can find the polr() function in the MASS
libray. The rms library also has ordered logit functionality.

beta <- coef(dglm)
tau <- dglm$zeta
attach(drury)
```

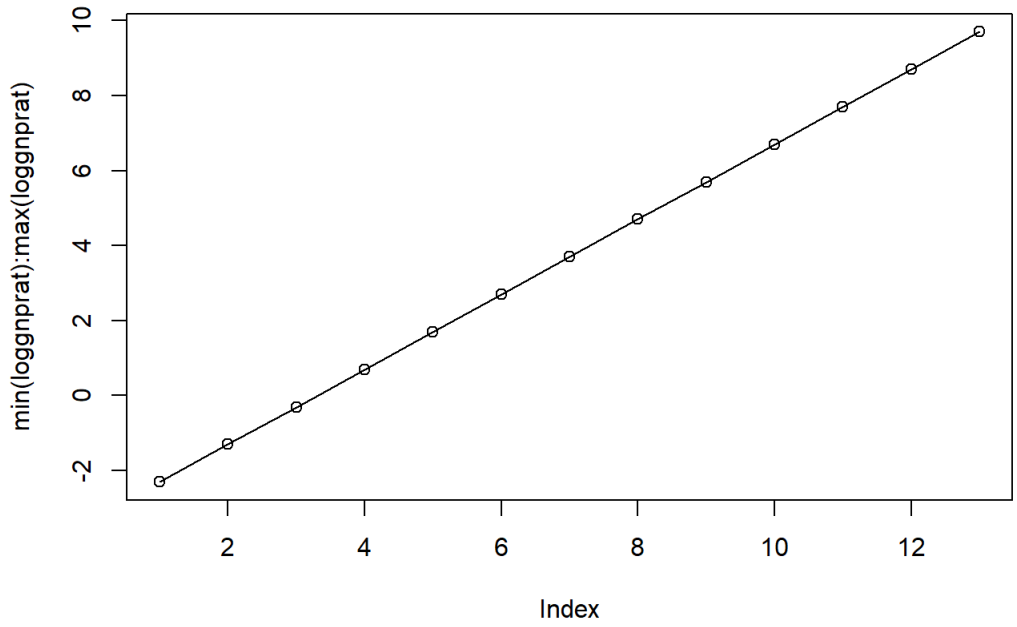
```
## The following object is masked from package:MASS:
##
##     coop
```

```
## We will construct a counterfactual scenario where all variables are held at their median values except fo
r logged GNP ratio. We wish to isolate the effect of marginal increases in logged GNP ratio.

X <- cbind(min(loggnprat):max(loggnprat), median(trade), median(tarcst), median(cost), median(coop))

p1 <- plogis(tau[1] - X %*% beta)
p2 <- plogis(tau[2] - X %*% beta) - plogis(tau[1] - X %*% beta)
p3 <- plogis(tau[3] - X %*% beta) - plogis(tau[2] - X %*% beta)
p4 <- 1.0 - plogis(tau[6] - X %*% beta)
```

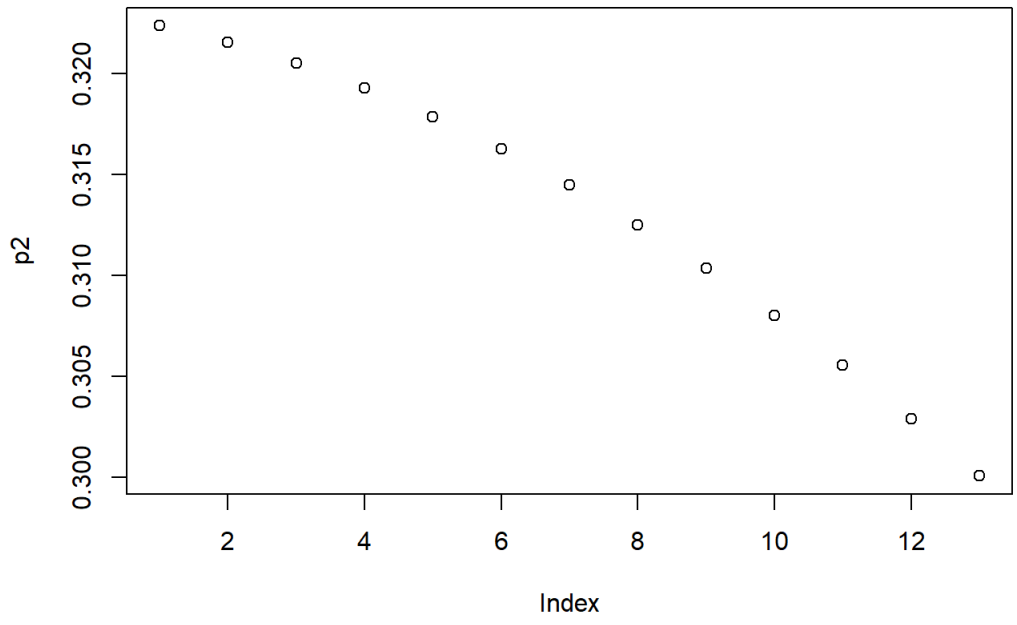
```
plot(p1)
```



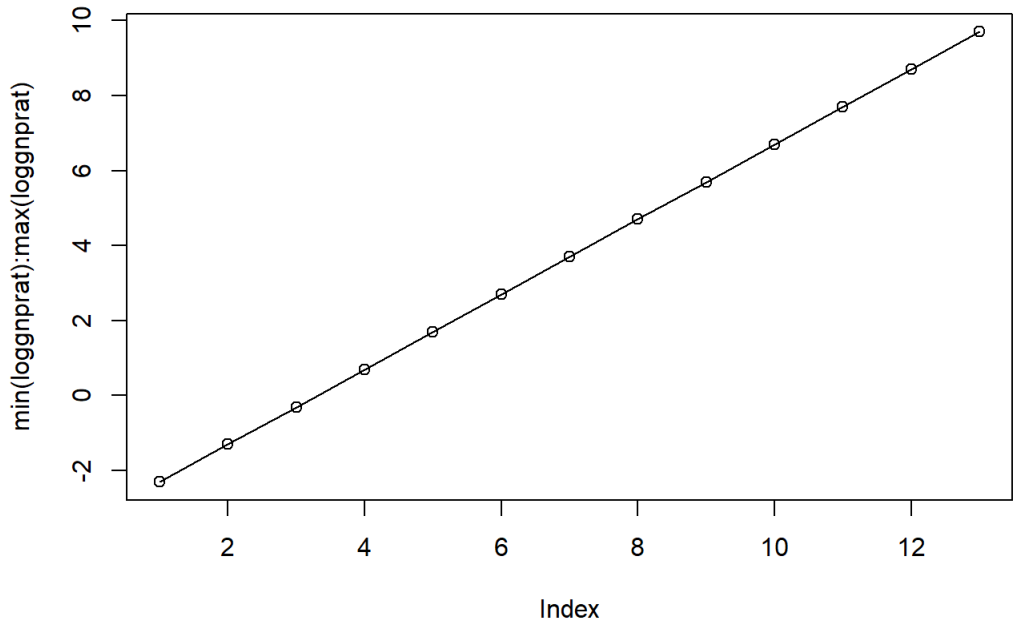

```
plot(min(loggnprat):max(loggnprat))
lines(min(loggnprat):max(loggnprat))
```



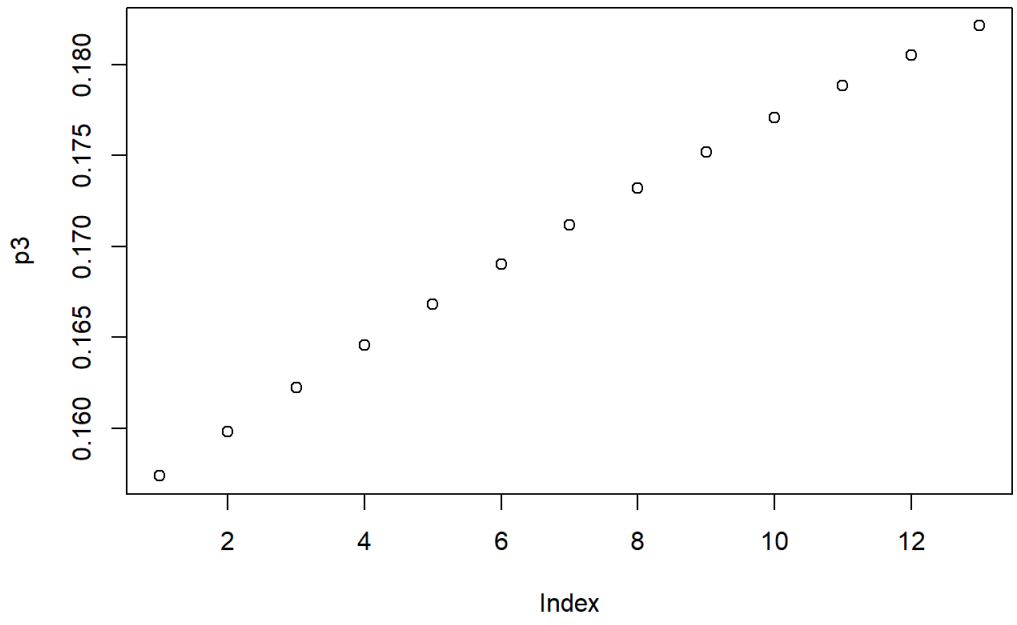

```
plot(p2)
```



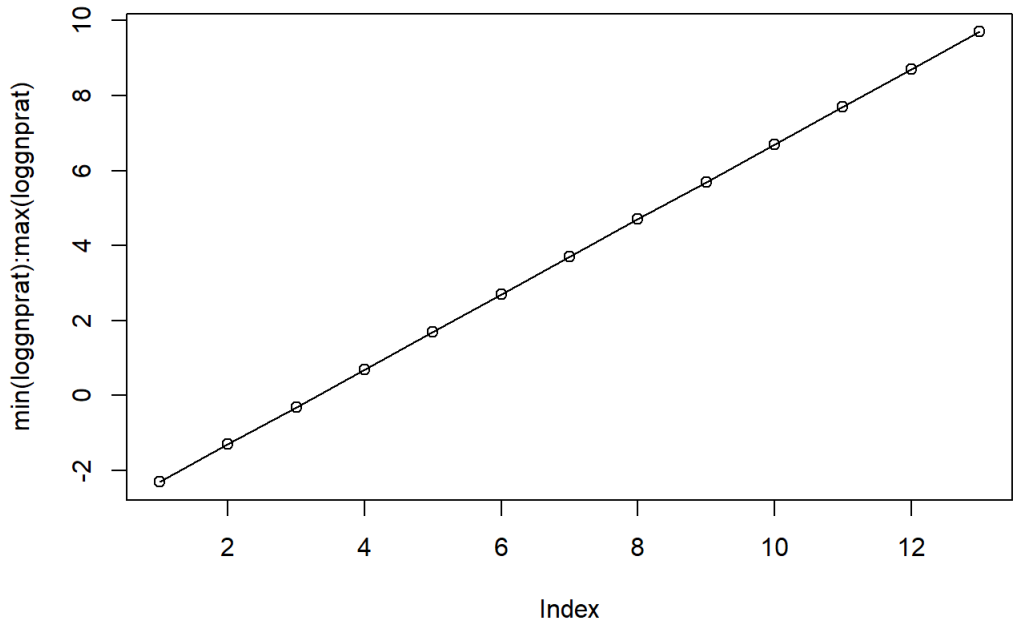

```
plot(min(loggnprat):max(loggnprat))
lines(min(loggnprat):max(loggnprat))
```




```
plot(p3)
```




```
plot(min(loggnprat):max(loggnprat))
lines(min(loggnprat):max(loggnprat))
```




```
## Fit the same linear specification but fitting the model as a multinomial logit. Evaluate whether the para
llel regressions assumption holds for this model.

library(mlogit)
```

```
## Warning: package 'mlogit' was built under R version 3.5.3
```

```
## Loading required package: Formula
```

```
## Warning: package 'Formula' was built under R version 3.5.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.5.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: lmtest
```

```
## Warning: package 'lmtest' was built under R version 3.5.2
```

```
library(nnet)
```

```
## Warning: package 'nnet' was built under R version 3.5.3
```

```
multmodel<-multinom((result) ~ loggnprat + trade + tarcst + cost + coop, data=drury)
```

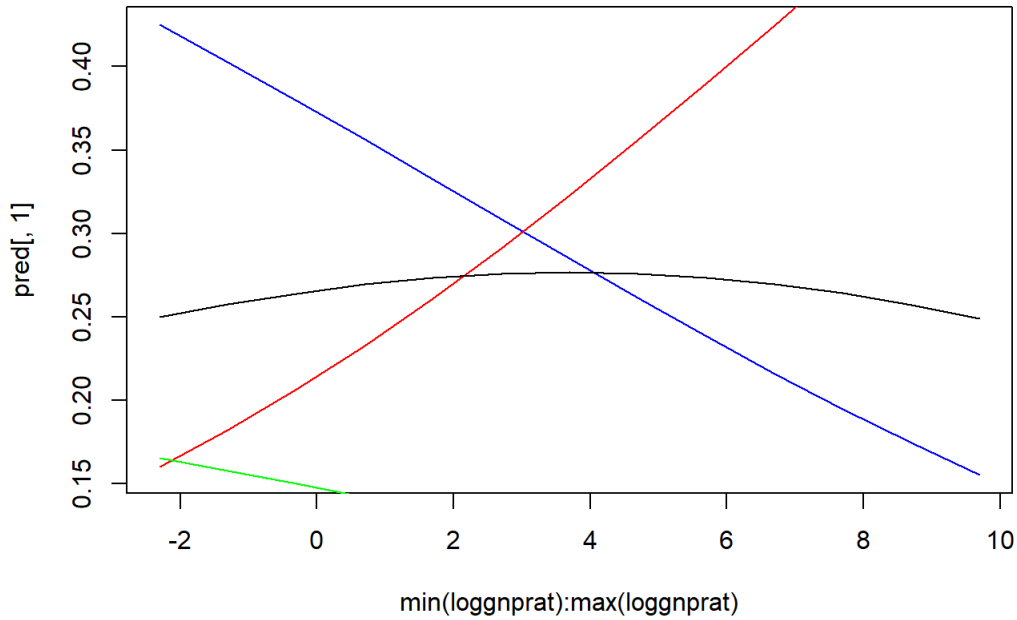
```
## # weights:  28 (18 variable)
## initial  value 160.810146
## iter  10 value 149.716660
## iter  20 value 144.305997
## final  value 144.300206
## converged
```

```
summary(multmodel)
```

```
## Call:
## multinom(formula = (result) ~ loggnprat + trade + tarcst + cost +
##     coop, data = drury)
##
## Coefficients:
##   (Intercept)   loggnprat      trade      tarcst       cost       coop
## 2   0.4489268 0.183869302 0.00303299 0.001370793 -0.1704769 -0.4081447
## 3   1.9246732 0.007374492 0.01438233 0.002967986 -1.1660786 -0.5541778
## 4   0.6681935 0.083692739 0.01561906 0.002766077 -0.2687851 -0.5442789
##
## Std. Errors:
##   (Intercept) loggnprat       trade      tarcst      cost      coop
## 2    1.081758 0.1357278 0.008112545 0.001707285 0.4227107 0.2786250
## 3    1.274881 0.1640398 0.008974747 0.001574317 0.5345912 0.3468039
## 4    1.051694 0.1397161 0.007695706 0.001549867 0.4114106 0.2796418
##
## Residual Deviance: 288.6004
## AIC: 324.6004
```

```
## The parallel regressions assumption is not necessary for a multinomial logit, because we are allowing the slope to change for each value that the target variable can take. We are not making the parallel regressions assumption here. However, from the results of this model, we can see that the parallel regressions assumption does not hold for our data. Indeed, we can see how (for example) the slope of the variable loggnprat changes between results 2 and 3, and the same is true for tarcst and cost.
```

```
## Using the scenario you devised in the first part of this question provide an interpretation of the MNL version of the model that you just fit. The multinom() function in the nnet library fits multinomial logit. You might also consider the mlogit library.

X <- data.frame(loggnprat=min(loggnprat):max(loggnprat), trade=median(trade), tarcst=median(tarcst),cost= median(cost), coop=median(coop))

pred <- predict(multmodel,X,type='probs')

plot(min(loggnprat):max(loggnprat), pred[,1], type='l', col='blue')
lines(min(loggnprat):max(loggnprat), pred[,2], col='red')
lines(min(loggnprat):max(loggnprat), pred[,3], col='green')
lines(min(loggnprat):max(loggnprat), pred[,4])
```




```
## If the parallel regressions assumption held, the lines above would be parallel.

summary(multmodel)
```

```
## Call:
## multinom(formula = (result) ~ loggnprat + trade + tarcst + cost +
##     coop, data = drury)
##
## Coefficients:
##   (Intercept)   loggnprat      trade      tarcst       cost       coop
## 2   0.4489268 0.183869302 0.00303299 0.001370793 -0.1704769 -0.4081447
## 3   1.9246732 0.007374492 0.01438233 0.002967986 -1.1660786 -0.5541778
## 4   0.6681935 0.083692739 0.01561906 0.002766077 -0.2687851 -0.5442789
##
## Std. Errors:
##   (Intercept) loggnprat       trade      tarcst      cost      coop
## 2    1.081758 0.1357278 0.008112545 0.001707285 0.4227107 0.2786250
## 3    1.274881 0.1640398 0.008974747 0.001574317 0.5345912 0.3468039
## 4    1.051694 0.1397161 0.007695706 0.001549867 0.4114106 0.2796418
##
## Residual Deviance: 288.6004
## AIC: 324.6004
```

p3 <- plogis(X %*% *coefficients(multmodel)[2,]) p4 <- plogis(X %*%* coefficients(multmodel)[3,]) p1 <- 1 - p2 - p3 - p4

plot(min(loggnprat):max(loggnprat), pred[,1], type='l', col='blue') lines(min(loggnprat):max(loggnprat), pred[,2], col='red') lines(min(loggnprat):max(loggnprat), pred[,3], col='green') lines(min(loggnprat):max(loggnprat), pred[,4]) ```