Assignment Seven

Ryan Honea

10/16/2017

Housing Conditions in Copenhagen

First, we load the `copen` data from the Princeton course website and then use `head` to see the first few rows of the data.

```
copen <- read.table("http://data.princeton.edu/wws509/datasets/copen.dat")
head(copen)
```

```
##   housing influence contact satisfaction  n
## 1   tower       low     low          low 21
## 2   tower       low     low       medium 21
## 3   tower       low     low         high 28
## 4   tower       low    high          low 14
## 5   tower       low    high       medium 19
## 6   tower       low    high         high 37
```

We can see several ordinal columns of data after the `head` command.

```
copen$g <- rep(1:24, rep(3,24))
```

> **Question (a):** What is the reason for creating the column called "'g"' in the "'copen"' dataframe?
>
> It classifies each observation by its covariate pattern, which in the case of this model is covariate patterns 1 to 24 with 3 repetitions each time. So, $g = 1$ is the 1st covariate pattern and $g = 5$ is the 5th coviarate pattern and so forth.

We now calculate the log-likelihood of the saturated multinomial logit model where each of the covariate patterns has its own distriution.

> **Question (b):** Why do they bother building the multinomial logistic model when the response is clearly ordinal?
>
> As stated, the multinomial model is saturated because of every covariate pattern has its own distriution. It is useful for comparitive purposes when we develop other models.

```
msat <- multinom(satisfaction ~ as.factor(g), weights = n, data=copen)
```

```
## # weights:  75 (48 variable)
## initial  value 1846.767257
## iter  10 value 1723.705246
## iter  20 value 1716.225889
## iter  30 value 1715.715730
## final  value 1715.710848
## converged
```

```
logLik(msat)
```

```
## 'log Lik.' -1715.711 (df=48)
```

Question (c): Build the multinomial model as they have, and run summary(msat) to view the output.

See below.

```
summary(msat)
```

```
## Call:
## multinom(formula = satisfaction ~ as.factor(g), data = copen,
##     weights = n)
##
## Coefficients:
##        (Intercept) as.factor(g)2 as.factor(g)3 as.factor(g)4 as.factor(g)5
## low     -0.2886478    -0.6832365     0.2313863    -0.5669864    -0.9923510
## medium  -0.2885274    -0.3779573    -0.2039615    -0.2647869    -0.8965552
##        as.factor(g)6 as.factor(g)7 as.factor(g)8 as.factor(g)9
## low        -1.748078     1.5662992     0.8842921     0.3611493
## medium     -1.237036     0.5908042     0.3561069     0.1552409
##        as.factor(g)10 as.factor(g)11 as.factor(g)12 as.factor(g)13
## low        -0.2944131     -0.4420131     -1.1306771      0.5509163
## medium     -0.3589863     -0.8097365     -0.6197527      0.1835797
##        as.factor(g)14 as.factor(g)15 as.factor(g)16 as.factor(g)17
## low         0.2885343     -0.1170797     -0.5868062    -0.11656204
## medium      0.4283366     -0.1170912      0.2014774     0.03743493
##        as.factor(g)18 as.factor(g)19 as.factor(g)20 as.factor(g)21
## low        -0.8098888      1.2330628      1.7668204      0.4325519
## medium     -0.4532797      0.1348777      0.8590568      0.2892427
##        as.factor(g)22 as.factor(g)23 as.factor(g)24
## low         1.1578662     -0.1630204     -0.6666918
## medium      0.7684034     -0.4997137     -0.4842675
##
## Std. Errors:
##        (Intercept) as.factor(g)2 as.factor(g)3 as.factor(g)4 as.factor(g)5
## low      0.2886764     0.4263692     0.3748657     0.4088496     0.4594891
## medium   0.2886665     0.4037162     0.3956744     0.3896221     0.4494186
##        as.factor(g)6 as.factor(g)7 as.factor(g)8 as.factor(g)9
## low        0.6783282     0.3981843     0.3455609     0.3627566
## medium     0.5716039     0.4308476     0.3582246     0.3700002
##        as.factor(g)10 as.factor(g)11 as.factor(g)12 as.factor(g)13
## low         0.3402876      0.3745831      0.4076057      0.5101746
## medium      0.3423111      0.3967269      0.3734380      0.5425955
##        as.factor(g)14 as.factor(g)15 as.factor(g)16 as.factor(g)17
## low         0.4281797      0.5400699      0.4743390      0.6009228
## medium      0.4204792      0.5400519      0.4128562      0.5807758
##        as.factor(g)18 as.factor(g)19 as.factor(g)20 as.factor(g)21
## low         0.5232673      0.5308137      0.4216675      0.4763875
## medium      0.4805637      0.6267298      0.4513702      0.4870314
##        as.factor(g)22 as.factor(g)23 as.factor(g)24
## low         0.4387818      0.5631062      0.6002167
## medium      0.4559374      0.6117482      0.5717386
##
## Residual Deviance: 3431.422
## AIC: 3527.422
```

The Proportional Odds Model

The first step in fitting the additive ordered logit model is to indicate a reference level. This is done in the bhhhelow code alongside ordering outcomes from low to high.

```
copen$satisfaction <- ordered(copen$satisfaction,c("low","medium","high"))

copen$housing   <- relevel(copen$housing,  ref="tower")

copen$influence <- factor(copen$influence,c("low","medium","high"))

copen$contact   <- relevel(copen$contact,  ref="low")
```

Question (d): As a preprocessing step for building the proportional odds model, they set the ordering on the response with the "ordered()" function, and set reference levels for the explanatory factors with the "relevel()" function. Are both of these steps absolutely necessary, or just preferred? What could go wrong if the response is not ordered properly when doing proportional odds? How does using "relevel()" make comparing models easier?

For the proportional odds model, having proper ordering is required as compared to the nominal where it is not. The culumative odds for the jth model are

$$\frac{P(z \leq C_j)}{P(z > C_j)} = \frac{\pi_1 + \pi_2 ... \pi_j}{\pi_{j+1} + ... + \pi_J}$$

where C_j are cutpoints in the model and so proper ordering is necessary to create proper cutpoints. In the case of the `relevel()` commands, which are being used on nominal factors, it aids in comparison by ensuring that all additives models' βs refer to the same associated factors.

Question (e): After ordering the response and setting reference levels for the explanatory factors, build the proportional odds model as they have and run summary(madd) to view the output. Compare the outputs of the two models (multinomial logistic vs. proportional odds). Specifically, compare the log-likelihood values, deviances, and model coefficients (you can get these beta's with > coef(msat) and > coef(madd)). Which model do you prefer and why?

See below for model and log-likelihoods. As the log-likelihood of the model that uses an ordinal factor is not significantly larger (in magnitude), and it is far easier to interpret and understand than the multinomial model, I am far more likely to use the ordinal model.

```
madd <- polr(satisfaction ~ housing + influence + contact, weights = n, data = copen)
logLik(madd)
```

```
## 'log Lik.' -1739.575 (df=8)
```

```
logLik(msat)
```

```
## 'log Lik.' -1715.711 (df=48)
```

```
summary(madd)
```

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

```
## Call:
## polr(formula = satisfaction ~ housing + influence + contact,
##     data = copen, weights = n)
##
## Coefficients:
##                      Value Std. Error t value
## housingapartments -0.5724    0.11924  -4.800
## housingatrium     -0.3662    0.15517  -2.360
## housingterraced   -1.0910    0.15149  -7.202
## influencemedium    0.5664    0.10465   5.412
## influencehigh      1.2888    0.12716  10.136
## contacthigh        0.3603    0.09554   3.771
##
## Intercepts:
##             Value   Std. Error t value
## low|medium  -0.4961  0.1248    -3.9739
## medium|high  0.6907  0.1255     5.5049
##
## Residual Deviance: 3479.149
## AIC: 3495.149
```

Question (f): One of these models results in substantially more output (more coefficients). Which one is it and why? Based on this, which model (multinomial logistic or proportional odds) is easier to understand?

Definitely the ordinal model. It is an easily interpeted equation having only six coefficients and 2 intercepts compared to the many coefficients of the multinomial model.

Question (g): Note the deviance calculate `> 2*(logLik(msat) - logLik(madd))`. Now it seems that the multinomial model is considered to be saturated, and they are using it to compute residual deviance for the proportional odds model. Is the multinomial model a saturated model and is this appropriate?

Yes, it is appropriate. The multinomial model treats every covariate pattern as its own group, and so it is indeed a saturated model. The deviance is calculated by two times the difference between the alternate model and the saturated model.

Question (h): Use the output resulting from `> 2*(logLik(msat) - logLik(madd))` to obtain a p-value. Take care to use correct degrees of freedom. What does this p-value tell you?

Based on this p-value below (0.813), we fail to reject that this model doesn't fit the data. As we fail to reject and the simpler model works, it would make sense to continue using the ordinal model.

```
D <- 2*(logLik(msat) - logLik(madd))
cat(pchisq(D, 40))
```

```
## 0.812559
```

Models with Interactions

Now, we create models with two-factor interactions, but to find which is the best, we check all possible two factor interactions and use the `update` command to update the model in temporary memory.

```
deviance(madd) - deviance(update(madd, . ~ . + housing:influence))
```

```
## [1] 22.50935
```

```
deviance(madd) - deviance(update(madd, . ~ . + housing:contact))
```

```
## [1] 8.666155
```

```
deviance(madd) - deviance(update(madd, . ~ . + influence:contact))
```

```
## [1] 0.2089536
```

Obviously the largest change is using the housing and influence interaction.

Question (i): Next investigate whether any of the two-way interactions of explanatory variables are significant. Run the R code there, comparing the deviance of madd and those of the three models with two-way explanatory interactions. The first line of text beneath this code reads, "The interaction between housing and influence reduces the deviance by 25.22 at the expense of only six d.f., so it is worth a second look." Is this a typo? How did they calculate this 25.22 reduction in deviance?

Yes, this is a typo. The 25.22 reduction in deviance is the difference between the model with housing:influence and the saturated model. The correct change in deviance between the ordinal model and the ordinal model with housing:influence is 22.51.

Question (j): Fit the new proportional odds model that now includes the housing:influence interaction:

```
> mint <- update(madd, . ~ . + housing:influence)
> summary(mint)
```

Is the model an improvement over the proportional odds model without any interaction terms? Explain.

With a p-value of approximately 0 for the difference in deviances at 6 degrees of freedom, the difference is definitely significant. So, it is quite the improvement.

```
mint <- update(madd, . ~ . + housing:influence)
summary(mint)
```

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

```
## Call:
## polr(formula = satisfaction ~ housing + influence + contact +
##     housing:influence, data = copen, weights = n)
##
## Coefficients:
##                                    Value Std. Error t value
## housingapartments                -1.1885    0.19724 -6.0256
## housingatrium                    -0.6067    0.24457 -2.4808
## housingterraced                  -1.6062    0.24100 -6.6650
## influencemedium                  -0.1390    0.21255 -0.6541
## influencehigh                     0.8689    0.27434  3.1671
## contacthigh                       0.3721    0.09599  3.8764
## housingapartments:influencemedium  1.0809    0.26585  4.0657
## housingatrium:influencemedium      0.6511    0.34500  1.8873
## housingterraced:influencemedium    0.8210    0.33067  2.4829
## housingapartments:influencehigh    0.7198    0.32873  2.1896
## housingatrium:influencehigh       -0.1555    0.41048 -0.3789
## housingterraced:influencehigh      0.8446    0.43027  1.9630
##
## Intercepts:
##             Value   Std. Error t value
## low|medium  -0.8882  0.1672    -5.3135
## medium|high  0.3126  0.1657     1.8871
##
## Residual Deviance: 3456.64
## AIC: 3484.64
```

```
D_1 <- deviance(mint) - deviance(madd)
D_2 <- deviance(mint) - deviance(msat)
cat(pchisq(D_1, 6),"\n")
```

```
## 0
```

Question (k): Calculate the percentage difference in odds of reporting high satisfaction (relative to medium or low satisfaction) for respondents who have high contact with neighbors versus those with low neighbor contact.

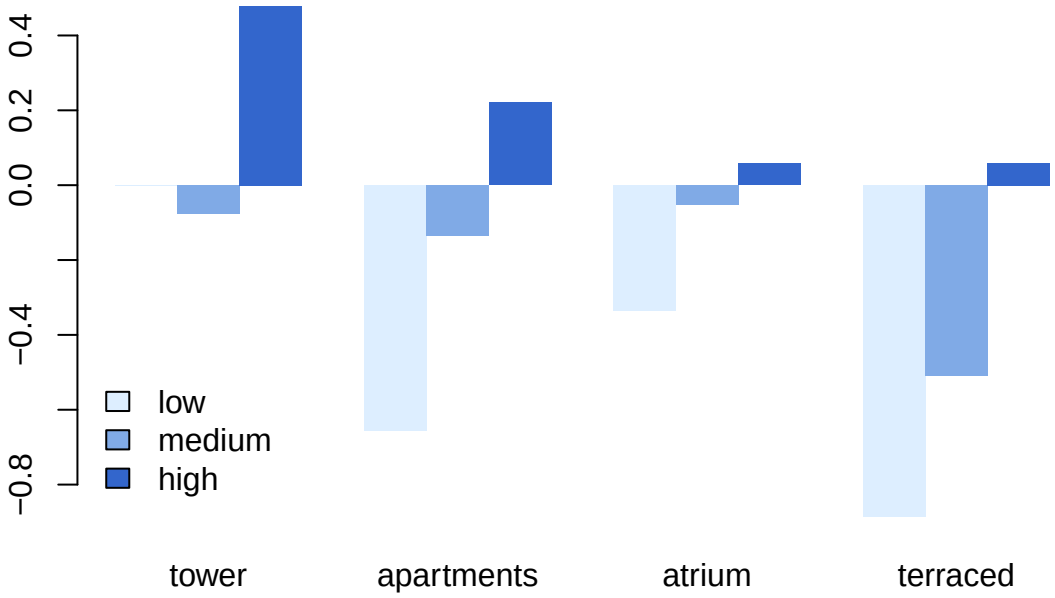
This is done in the below code. We only need to find the percentage difference from the normal because the only two factor levels are low and high. So, the low is included in the intercept. The difference in odds, therefore, is 45.1 percent.

```
b <- coef(mint)
b["contacthigh"]/(pi/sqrt(3))
```

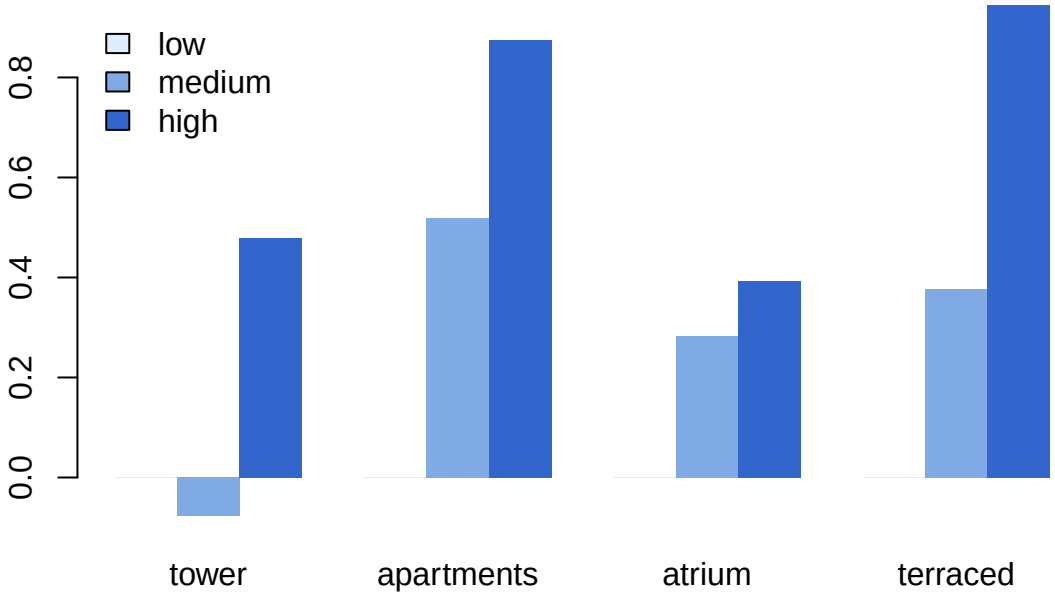
```
## contacthigh
##   0.2051395
```

Now, to see the joint effects of influence and housing, we create the below model and use the graph to visualize various probability differences.

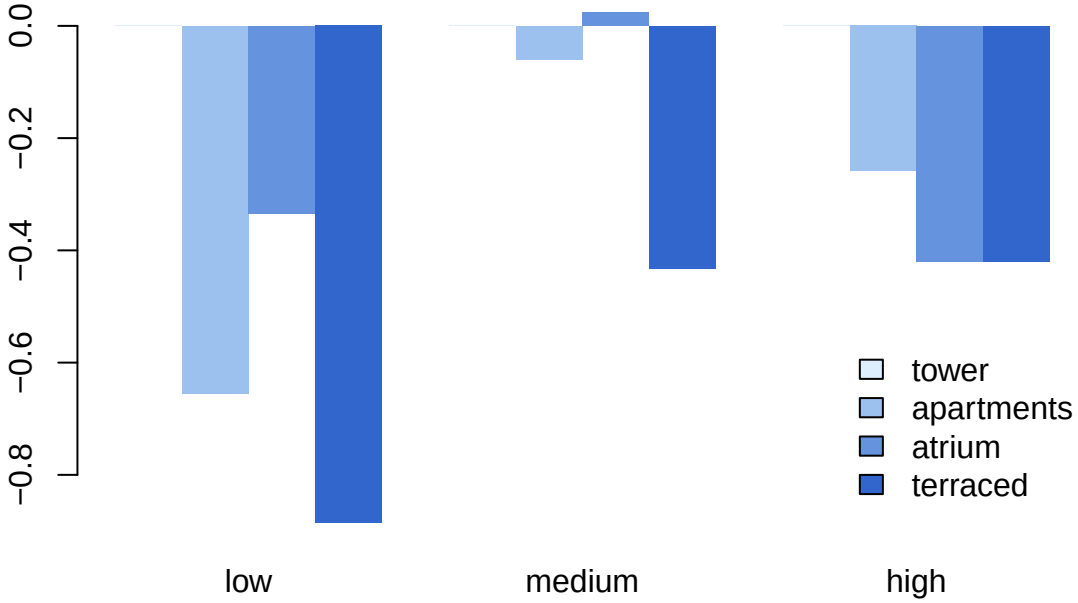
```
mint2 <- polr(satisfaction ~ contact + housing:influence, weights = n, data = copen)
HI <- matrix(c(coef(mint2)[-1],0),4,3)
HI <- (HI - HI[1,1])/(pi/sqrt(3))
rownames(HI) <- levels(copen$housing)
colnames(HI) <- levels(copen$influence)
trio <- c("#ddeeff","#80aae6", "#3366cc")
barplot(t(HI), beside=TRUE, col=trio, border=NA)
legend("bottomleft", fill=trio, legend=levels(copen$influence), bty="n")
```




```
quartet <- c("#ddeeff", "#9dc1ee","#6593dd", "#3366cc")
barplot(apply(HI,1,function(x)x-x[1]), beside=TRUE, col=trio, border=NA)
legend("toplef", fill=trio, legend=levels(copen$influence), bty="n")
```




```
barplot(apply(HI,2,function(x)x-x[1]), beside=TRUE, col=quartet, border=NA)
legend("bottomright", fill=quartet, legend=levels(copen$housing), bty="n")
```



Question (l): In the next part, Dr. Rodriguez investigates housing type, influence, and the interaction between the two by standardizing the coefficients (or effects) of the explanatory variales in a new model (one that just uses contact and the housing:influence interaction). Make all the graphs. Write two or three sentences that summarize what these graphs indicate about housing satisfaction.

The first graph describes effects on the probability differences on satisfaction for various effects. The second is similar to the first but uses the low influence as a reference point. The final graph is essentially the reverse of the first.

Question (m): Using the "mint" model (the one with the three main factors and one interaction), use the predict() or fitted() command to obtain the probability estimates for each of the 24 groups. Then report the estimated probabilities of low, medium, and high satisfaction for terraced tenants with medium influence and both contact levels. If you lived in a terraced living situation with medium influence, would you rather have low or high contact with your neighbors?

Based on the output below, one would want high contact with their neighbors because the estimated probability of having high satisfaction is approximately 2

```
copen$probs <- predict(mint, type="probs")
terraceHigh<- subset(copen, housing == "terraced" & influence == "medium"
                     & satisfaction == "high")
print(terraceHigh[,c("contact","probs")])
```

```
##    contact probs.low probs.medium probs.high
## 63     low 0.5090174    0.2660024  0.2249802
## 66    high 0.4167794    0.2868819  0.2963387
```