Ordered logistic regression R

```
plot(s, which=1:3, pch=1:3,
    xlab=’logit’, main=’ ’,
    xlim=range(s[,3:4]))
```

Ordered logistic regression R

- Once we are done assessing whether the assumptions of our model hold, we can obtain predicted probabilities, which are usually easier to understand than either the coefficients or the odds ratios.
- For example, we can vary gpa for each level of pared and public and calculate the probability of being in each category of apply.
- We do this by creating a new dataset of all the values to use for prediction.

Ordered logistic regression R

```
newdat <- data.frame(
  pared = rep(0:1, 200),
  public = rep(0:1, each = 200),
  gpa = rep(seq(from = 1.9, to = 4,
      length.out = 100), 4))

newdat <- cbind(newdat, predict(m,
   newdat, type = "probs"))
```

Ordered logistic regression R

```
# Show first few rows
head(newdat)
  pared public   gpa unlikely somewhat likely very likely
1     0      0 1.900   0.7376          0.2205     0.04192
2     1      0 1.921   0.4932          0.3946     0.11221
3     0      0 1.942   0.7325          0.2245     0.04299
4     1      0 1.964   0.4867          0.3985     0.11484
5     0      0 1.985   0.7274          0.2285     0.04407
6     1      0 2.006   0.4802          0.4023     0.11753
```

Ordered logistic regression R

- Now we can reshape the data long with the reshape2 package and plot all of the predicted probabilities for the different conditions.
- We plot the predicted probilities, connected with a line, coloured by level of the outcome, apply, and facetted by level of pared and public.
- We also use a custom label function, to add clearer labels showing what each column and row of the plot represent.

Ordered logistic regression R

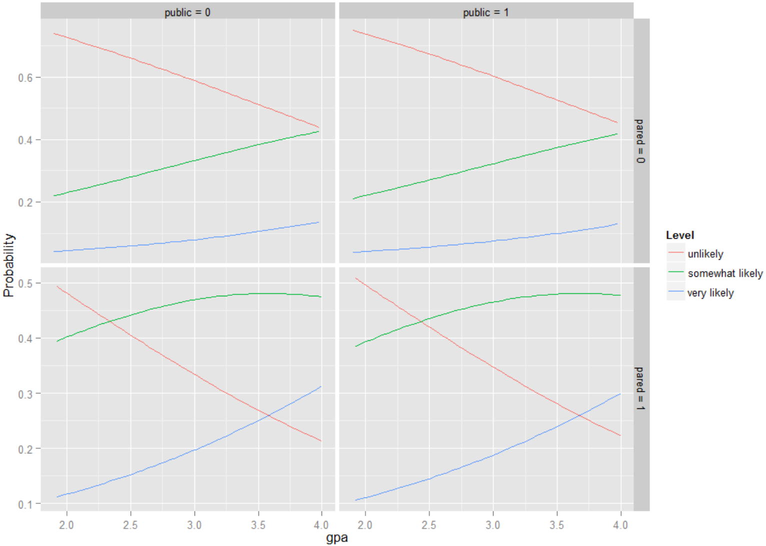
```
library(reshape2)

lnewdat <- melt(newdat,
  id.vars = c("pared", "public", "gpa"),
  variable.name = "Level",
  value.name="Probability")
```

Ordered logistic regression R

```
%## view first few rows
%head(lnewdat)
%##   pared public   gpa    Level Probability
%## 1     0      0 1.900 unlikely      0.7376
%## 2     1      0 1.921 unlikely      0.4932
%## 3     0      0 1.942 unlikely      0.7325
%## 4     1      0 1.964 unlikely      0.4867
%## 5     0      0 1.985 unlikely      0.7274
%## 6     1      0 2.006 unlikely      0.4802
```

Ordered logistic regression R

```
ggplot(lnewdat, aes(x = gpa, y = Probability,
  colour = Level)) +
  geom_line() +
  facet_grid(pared ~ public, scales="free",
    labeller=function(x, y) sprintf("%s = %d", x, y))
```


public = 0
public = 1
pared = 0
pared = 1
Probability
0.6
0.4
0.2
0.5
0.4
0.3
0.2
0.1
2.0
2.5
3.0
3.5
4.0
gpa
Level
unlikely
somewhat likely
very likely

Ordered logistic regression R

Things to consider **Perfect prediction:** Perfect prediction means that one value of a predictor variable is associated with only one value of the response variable.

Ordered logistic regression R

Sample size: Both ordered logistic and ordered probit, using maximum likelihood estimates, require sufficient sample size.
Empty cells or small cells: You should check for empty or small cells by doing a crosstab between categorical predictors and the outcome variable. If a cell has very few cases, the model may become unstable or it might not run at all.

Ordered logistic regression R

Pseudo-R-squared: There is no exact analog of the R-squared found in OLS. There are many versions of pseudo-R-squares. Please see Long and Freese 2005 for more details and explanations of various pseudo-R-squares.

Diagnostics: Doing diagnostics for non-linear models is difficult, and ordered logit/probit models are even more difficult than binary models.