Ordinal Models

Week 7
POLS 8830: Advanced Quantitative Methods

Ryan Carlin
Georgia State University
rcarlin@gsu.edu

Development of Ordinal Models

- Types of variables:
  1. Interval
  2. Ratio
  3. Ordinal
  4. Nominal
- Originally OLS was used to estimate models with ordinal dependent variables
- Assumes a latent variable underlying the observed categories of the dependent variable

Example of Ordinal Variable

- Evaluation of the economy:
  1. Much Worse — 45%
  2. Somewhat Worse — 30%
  3. No Change — 12 %
  4. Somewhat Better — 8%
  5. Much Better — 5%
- Consider a latent (unobserved) variable (y^*) the measures 'real' evaluations of the economy on a scale for $-\infty$ to ∞
- Working on the assumption that our observed values for $y = 1, 2, 3, 4, 5$ reflect the unobserved y^* we can estimate with OLS

Example of Ordinal Variable

- To do this we map the observe values of y onto the continuous (theoretical) scale of y^* based on cutpoints between the categories

$$y_i = \begin{cases} 1 & \text{Much Worse} & \text{if} & -\infty \leq & y^* < & \tau_1 \\ 2 & \text{Somewhat Worse} & \text{if} & \tau_1 \leq & y^* < & \tau_2 \\ 3 & \text{No Change} & \text{if} & \tau_2 \leq & y^* < & \tau_3 \\ 4 & \text{Somewhat Better} & \text{if} & \tau_3 \leq & y^* < & \tau_4 \\ 5 & \text{Much Better} & \text{if} & \tau_4 \leq & y^* < & \infty \end{cases}$$

Using OLS to Estimate Latent y^*

- Using OLS to estimate models with ordinal dependent variables is NOT problematic based on the above logic if two assumptions are met:
  1. The distance between categories is equal
  2. The variance across categories is constant
- Both of these are pretty strong assumptions
- Ordered logit/probit models allow us to estimate the location of the cutpoints (τ) while relaxing the strong assumption the distance between categories must be equal
- However, in practice the ordered probit/logit models converge to OLS when the dependent variable contains more than 5 categories. (We should **NOT** assume this. More on how to test for it later.)

Using OLS to Estimate Latent y^*

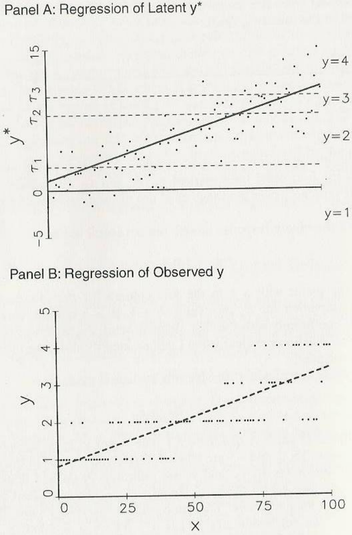


Figure 5.1. Regression of a Latent Variable y^* Compared to the Regression of the Corresponding Observed Variable y

Ordered Logit/Probit

- Instead of assuming that y^* is a continuous variable, we can:
- Assume that the observed variable y contains observations drawn from a probability distribution
- This probability distribution extends across each value of y

Ordered Logit/Probit

$$\Pr(y_i = 1 \mid \mathbf{x}_i) = \Pr(\tau_0 \leq y_i^* < \tau_1 \mid \mathbf{x}_i)$$

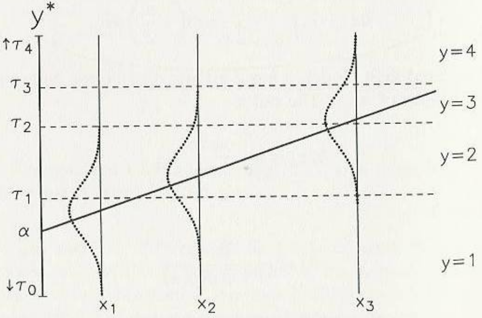


Figure 5.2. Distribution of y^* Given x for the Ordered Regression Model

Ordered Logit/Probit

- Begin by estimating the probability that $y = 1$

$$\Pr(y_1 = 1|\mathbf{X}_i) = \Pr(\tau_0 \leq y_i^* < \tau_1|\mathbf{X}_i) \tag{1}$$

- Since the underlying latent variable y^* is unobservable, we substitute the predicted regression line $\mathbf{X}\beta + \epsilon$

$$\Pr(y_1 = 1|\mathbf{X}_i) = \Pr(\tau_0 \leq \mathbf{X}\beta + \epsilon < \tau_1|\mathbf{X}_i) \tag{2}$$

- Subtracting $\mathbf{X}\beta$ within the inequality helps identify specific cutpoints

$$\Pr(y_1 = 1|\mathbf{X}_i) = \Pr(\tau_0 - \mathbf{X}\beta \leq +\epsilon < \tau_1 - \mathbf{X}\beta|\mathbf{X}_i) \tag{3}$$

Ordered Logit/Probit

- Equation 3 specifies that the probability of a random variable (y) taking a specific value (e.g. $y = 1$) can be calculated by determining whether y falls between two specific values
- Doing so, requires calculating the difference between the cdf evaluated at both values

$$\Pr(y_i = 1|\mathbf{X}_i) = \Pr(\epsilon < \tau_1 - \mathbf{X}\beta|\mathbf{X}_i) - \Pr(\epsilon \leq \tau_0 - \mathbf{X}\beta|\mathbf{X}_i) \quad (4)$$

$$\Pr(y_i = 1|\mathbf{X}_i) = F(\tau_1 - \mathbf{X}\beta) - F(\tau_0 - \mathbf{X}\beta) \quad (5)$$

- Where F in Equation 5 is the cdf of the normal distribution (Φ) for an ordered probit, or the cdf of the logistic distribution (Λ) for an ordered logit

Ordered Logit/Probit

- The previous steps can be generalized to any m category dependent variable y

$$\Pr(y_i = 1|\mathbf{X}_i) = F(\tau_m - \mathbf{X}\beta) - F(\tau_{m-1} - \mathbf{X}\beta) \tag{6}$$

- When calculating $\Pr(y_i = 1)$, the second term on the right-hand side drops out, as:

$$F(\tau_0 - \mathbf{X}\beta) = F(-\infty - \mathbf{X}\beta) = 0 \tag{7}$$

- Additionally, when calculating the probability of the last category ($y = m$), the first term on the right-hand side equals 1, as:

$$F(\tau_m - \mathbf{X}\beta) = F(\infty - \mathbf{X}\beta) = 1 \tag{8}$$

Ordered Logit/Probit

- Problem with proceeding under this estimation
  - The four previous equations remain unidentified
  - We have more parameters than equations (5 to 4)
- In order to estimate, we must arbitrarily set one parameter equal to 0
  - Our options are either $\tau_1 = 0$ or $\alpha = 0$
  - R prefers to estimate all cutpoints so we generally use $\alpha = 0$

Ordered Logit/Probit

- Once we set one parameter equal to 0 we can maximize the likelihood function
- Must remember that maximization needs to occur for a value that falls between two categories

$$
\begin{aligned}
L(\beta,\tau|y,\mathbf{X}) &= \prod_{m=1}^{M}\prod_{y_i=m}\Pr(y_i=m|\mathbf{X},\beta,\tau) \\
&= \prod_{m=1}^{M}\prod_{y_i=m}[F(\tau_m-\mathbf{X}\beta)-F(t_{m-1}-\mathbf{X}\beta)]
\end{aligned} \tag{9}
$$

Ordered Logit/Probit

- Taking the natural log gives us

$$\ln L(\beta, \tau | y, \mathbf{X}) = \sum_{m=1}^{M} \sum_{y_i = m} \ln[F(\tau_m - \mathbf{X}\beta) - F(t_{m-1} - \mathbf{X}\beta)] \quad (10)$$

- In essence, this is the same as taking the natural log for the equation of each y value and then summing across all equations

Parallel Lines Assumption

- An core assumption in ordered logit/probit models that unfortunately is often ignored.
- Essentially, the parallel lines assumption (sometimes called proportional odds assumption) states that the relationship between each independent variable and the dependent variable should not change across categories
- Can use the Brant test to check if the assumption is violated
  - R package of **brant**, with R command of `brant(ordered_model_object)`
  - R package of **car**, with R command of `poTest(ordered_model_object)`
- Can then use `clm` framework (cumulative link function) to only estimate those explanatory variables that do not violate the PLA

Estimating Ordered Logit/Probit in R

- Instead of the `glm` framework, we use an expansion of glm with the package **MASS**, and the command `polr()` (Proportional Odds Logistic Regression)
  - Despite the name, `polr` can handle probit, and other ordinal models by specifying `method`
  - Default is logistic

Estimating Ordered Logit/Probit in R

- Basic syntax:
  - `polr(formula, data, method, Hess, ...)`
  - `method = c("logistic", "probit", "loglog", "cloglog", "cauchit")`
  - Hess default is False; need to set to True to call `summary` and/or `vcov`
  - `weights` and `subset` options are also available, and function as their use in `glm`

A Note on Objects of Class Factor

- Categorical variables are often assigned as ‘factor’ class objects
- This allows for a text 'label' at each level of the ordinal variable
- lm, glm, and polr treat these in a dissimilar manner than numeric class variables
- Functionally in estimation, create a series of binary indicators for each level and provide a coefficient for each
    - The lowest level of the factor will be omitted as the baseline (to avoid perfect multicollinearity)
    - Can change this with `relevel(..., ref = ...)`
- This may or may not be useful for you, but you need to remain conscious of this and check object classes when importing/managing data

A Note on Objects of Class Factor

- For example (outcome is the importance of religion (1 to 4))

Factors		Numeric	
Abortion: Yes	2.139***	Abortion	2.174***
	(0.119)		(0.118)
Gender: Male	−0.637***	Gender	−0.643***
	(0.080)		(0.080)
Urban: Urban	−0.147	Urban	−0.167*
	(0.093)		(0.092)
Education: Community college	−0.061	Education	0.084***
	(0.117)		(0.024)
Education: Graduate Degree	−0.311**		
	(0.146)		
Education: High School Graduate	0.127		
	(0.120)		
Education: Some High School	0.610***		
	(0.146)		
Education: Some post-secondary	0.206		
	(0.142)		
Observations	2,231		2,231
Note:			*p<0.1; **p<0.05; ***p<0.01

Estimating Ordered Logit/Probit in R

- Object class requirements:
    - `polr` demands the outcome variable be of class ‘factor’
    - You can bypass this by specifying as.factor(DV) if you so desire, e.g. if your DV is numeric
    - IVs can be numeric or factor, but factor class IVs will be calculated as indicators, increasing the *k* and thus reducing *df* as in the next example

Example Output

```
> polr_object<-polr(poverty ~ religion + degree + country + age + gender, data=wvs, Hess = TRUE)
> summary(polr_object)
Call:
polr(formula = poverty ~ religion + degree + country + age +
    gender, data = wvs, Hess = TRUE)

Coefficients:
                 Value Std. Error t value
religionyes    0.17973   0.077346   2.324
degreeyes      0.14092   0.066193   2.129
countryNorway -0.32235   0.073766  -4.370
countrySweden -0.60330   0.079494  -7.589
countryUSA     0.61777   0.070665   8.742
age            0.01114   0.001561   7.139
gendermale     0.17637   0.052972   3.329

Intercepts:
                        Value   Std. Error t value
Too Little|About Right  0.7298  0.1041      7.0128
About Right|Too Much    2.5325  0.1103     22.9496

Residual Deviance: 10402.59
AIC: 10420.59
```

Estimating Ordered Logit/Probit in R

- Next step should always to be to check the Parallel Lines Assumption
  - `brant(polr_object)`
  - `poTest(polr_object)`
- Significant results ($p < 0.05$) indicate that a coefficient violates the PLA

Example Output

```
> brant(wvs_fit2)
--------------------------------------------
Test for        X2      df      probability
--------------------------------------------
Omnibus         90.38   5       0
religion        0.16    1       0.69
degree          15.12   1       0
country         72.36   1       0
age             3.49    1       0.06
gender          3.26    1       0.07
--------------------------------------------

H0: Parallel Regression Assumption holds
> poTest(wvs_fit2)

Tests for Proportional Odds
polr(formula = as.factor(poverty) ~ religion + degree + country +
    age + gender, data = wvs2, Hess = TRUE)

         b[polr]  b[>1]   b[>2] Chisquare df Pr(>Chisq)
Overall                             90.37  5     <2e-16 ***
religion -0.0740 -0.0478 -0.0895     0.16  1     0.6918
degree    0.0217  0.1078 -0.2664    15.12  1     0.0001 ***
country   0.1492  0.0782  0.3429    72.36  1     <2e-16 ***
age       0.0126  0.0116  0.0156     3.49  1     0.0619 .
gender    0.1471  0.1771  0.0428     3.26  1     0.0709 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Estimating Ordered Logit/Probit in R

- As we can see, the covariates of degree and country violate the PLA
  - Original Model: `polr(as.factor(poverty) ~ religion + degree + country + age + gender, data=wvs2, Hess = TRUE)`
  - CLM Model: `clm(as.factor(poverty) ~ religion + age + gender, nominal = ~ degree + country, data=wvs2)`
- `nominal` = allows us to relegate those covariates that violate the PLA to a nominal treatment, and are placed within the threshold report
  - `nominal = ~ IV1 + IV2 ...`

Example Output

```
> summary(clm_fit)
formula: as.factor(poverty) ~ religion + age + gender
nominal: ~degree + country
data:    wvs2

 link  threshold nobs logLik   AIC      niter max.grad cond.H
 logit flexible  5381 -5263.81 10545.63 5(0)  2.47e-09 3.0e+05

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
religion -0.067959   0.074625  -0.911  0.36246
age       0.012436   0.001542   8.065  7.3e-16 ***
gender    0.149591   0.052446   2.852  0.00434 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Threshold coefficients:
                 Estimate Std. Error z value
1|2.(Intercept)   0.98906    0.19865   4.979
2|3.(Intercept)   2.94450    0.22411  13.139
1|2.degree       -0.11505    0.06836  -1.683
2|3.degree        0.28071    0.09828   2.856
1|2.country      -0.07612    0.02409  -3.160
2|3.country      -0.34741    0.03255 -10.675
```

Interpretation of Coefficients

- Interpretation of ordered logit/probit coefficients is different than regular logit/probit models
  - In the regular model the interpretation is based on the change in probability of y going to 1
  - In the ordered models, interpretations are based on specific values of the dependent variable ($y = 1$, $y = 2$, etc.)
- Therefore, calculation of marginal effects or predicted probabilities must be estimated separately for each value of y
  - When calculating predicted probabilities with `margins`, this is to some degree automated

Example Output Using Margins

- When compared to logit models, the `margins` command for ordered models will produce a much larger matrix due to the increased number of outcome categories
  - This generally means that there will be a delay in R as each level is calculated
- Even in a simple case, with 5 outcome categories and a single binary explanatory variable, the `margins` results can be somewhat confusing to interpret
- The variables in the following examples are:
  - A three category measure of labor participation (not working, part time, full time)
  - An four category measure on the importance of religion(not important, not very important, somewhat important, very important)

Example Output Using Margins

at(hincome)	at(children)	at(region)	hincome	children	region
1	1	1	-0.011262	-0.44436	0.0081752
2	1	1	-0.011501	-0.45379	0.0083486
3	1	1	-0.011730	-0.46285	0.0085152
4	1	1	-0.011949	-0.47149	0.0086741
5	1	1	-0.012156	-0.47967	0.0088247
6	1	1	-0.012351	-0.48736	0.0089662
7	1	1	-0.012533	-0.49452	0.0090979
8	1	1	-0.012700	-0.50110	0.0092190
9	1	1	-0.012851	-0.50709	0.0093291
10	1	1	-0.012987	-0.51243	0.0094274
11	1	1	-0.013105	-0.51711	0.0095135
12	1	1	-0.013206	-0.52110	0.0095869
13	1	1	-0.013289	-0.52437	0.0096471
14	1	1	-0.013354	-0.52691	0.0096939
15	1	1	-0.013399	-0.52871	0.0097269
16	1	1	-0.013426	-0.52975	0.0097460
17	1	1	-0.013433	-0.53002	0.0097510
18	1	1	-0.013420	-0.52953	0.0097420
19	1	1	-0.013388	-0.52828	0.0097190
20	1	1	-0.013338	-0.52627	0.0096821
21	1	1	-0.013268	-0.52352	0.0096315
22	1	1	-0.013180	-0.52004	0.0095675
23	1	1	-0.013074	-0.51586	0.0094905
24	1	1	-0.012950	-0.51099	0.0094009
25	1	1	-0.012810	-0.50546	0.0092992
26	1	1	-0.012654	-0.49931	0.0091859
27	1	1	-0.012483	-0.49255	0.0090617
28	1	1	-0.012298	-0.48524	0.0089272
29	1	1	-0.012099	-0.47741	0.0087831
30	1	1	-0.011888	-0.46909	0.0086301
31	1	1	-0.011666	-0.46033	0.0084689
32	1	1	-0.011434	-0.45117	0.0083003
33	1	1	-0.011193	-0.44164	0.0081251
34	1	1	-0.010943	-0.43180	0.0079440
35	1	1	-0.010687	-0.42168	0.0077579
36	1	1	-0.010424	-0.41133	0.0075674
37	1	1	-0.010157	-0.40078	0.0073733
38	1	1	-0.009886	-0.39007	0.0071763
39	1	1	-0.009611	-0.37925	0.0069772
40	1	1	-0.009335	-0.36835	0.0067766
41	1	1	-0.009058	-0.35740	0.0065752
42	1	1	-0.008780	-0.34644	0.0063736
43	1	1	-0.008503	-0.33550	0.0061723
44	1	1	-0.008227	-0.32461	0.0059719
45	1	1	-0.007953	-0.31379	0.0057730

Example Output Using Margins

at(abortion)	at(gender)	at(education)	at(urban)	abortion	gender	education	urban
1	1	1	1	-0.42523	0.12576	-0.016459	0.032712
2	1	1	1	-0.08295	0.02453	-0.003211	0.006381
1	2	1	1	-0.52547	0.15540	-0.020339	0.040423
2	2	1	1	-0.14704	0.04349	-0.005692	0.011311
1	1	2	1	-0.40831	0.12075	-0.015804	0.031410
2	1	2	1	-0.07675	0.02270	-0.002971	0.005904
1	2	2	1	-0.51661	0.15278	-0.019996	0.039741
2	2	2	1	-0.13678	0.04045	-0.005294	0.010522
1	1	3	1	-0.39102	0.11564	-0.015135	0.030079
2	1	3	1	-0.07097	0.02099	-0.002747	0.005460
1	2	3	1	-0.50618	0.14970	-0.019593	0.038939
2	2	3	1	-0.12712	0.03760	-0.004921	0.009779
1	1	4	1	-0.37350	0.11046	-0.014457	0.028732
2	1	4	1	-0.06560	0.01940	-0.002539	0.005047
1	2	4	1	-0.49434	0.14619	-0.019134	0.038028
2	2	4	1	-0.11805	0.03491	-0.004569	0.009081
1	1	5	1	-0.35591	0.10525	-0.013776	0.027379
2	1	5	1	-0.06061	0.01793	-0.002346	0.004663
1	2	5	1	-0.48122	0.14231	-0.018626	0.037018
2	2	5	1	-0.10954	0.03239	-0.004240	0.008427
1	1	6	1	-0.33836	0.10006	-0.013097	0.026028
2	1	6	1	-0.05598	0.01656	-0.002167	0.004306
1	2	6	1	-0.46698	0.13810	-0.018075	0.035923
2	2	6	1	-0.10157	0.03004	-0.003931	0.007813
1	1	1	2	-0.45716	0.13520	-0.017695	0.035168
2	1	1	2	-0.09665	0.02858	-0.003741	0.007435
1	2	1	2	-0.53803	0.15911	-0.020825	0.041389
2	2	1	2	-0.16928	0.05006	-0.006552	0.013022
1	1	2	2	-0.44143	0.13055	-0.017086	0.033958
2	1	2	2	-0.08952	0.02647	-0.003465	0.006886
1	2	2	2	-0.53259	0.15750	-0.020615	0.040970
2	2	2	2	-0.15778	0.04666	-0.006107	0.012138
1	1	3	2	-0.42502	0.12569	-0.016451	0.032695
2	1	3	2	-0.08287	0.02451	-0.003208	0.006375
1	2	3	2	-0.52537	0.15537	-0.020335	0.040415
2	2	3	2	-0.14691	0.04345	-0.005686	0.011301
1	1	4	2	-0.40809	0.12069	-0.015796	0.031393
2	1	4	2	-0.07667	0.02267	-0.002968	0.005898
1	2	4	2	-0.51648	0.15274	-0.019991	0.039731
2	2	4	2	-0.13666	0.04041	-0.005290	0.010513
1	1	5	2	-0.39080	0.11557	-0.015126	0.030063
2	1	5	2	-0.07090	0.02097	-0.002744	0.005454
1	2	5	2	-0.50604	0.14965	-0.019587	0.038928
2	2	5	2	-0.12701	0.03756	-0.004916	0.009770
1	1	6	2	-0.37328	0.11039	-0.014449	0.028715
2	1	6	2	-0.06554	0.01938	-0.002537	0.005042
1	2	6	2	-0.49418	0.14615	-0.019128	0.038015
2	2	6	2	-0.11794	0.03488	-0.004565	0.009073

Predicted Probabilities

- Unlike the margins command in STATA which allows both predicted probabilities and marginal effects, the `margins` command in R has limited functionality
- The solution is a bit more intensive
- First, need to create a new data frame with the covariate values desired
- Next, use the `predict` function with the `type = "probs"` to calculate predicted probabilities
- Alternatively, one can use `type = "class"` to predict the category (factor value) at different covariate values

Predicted Probabilities

- This will look something like:
  - polr_object< −polr(as.factor(DV) X1 + X2 + X3 ..., data=df, ...)
  - PPdf < − data.frame(X1=rep(mean(df$X1),2), X2=rep(mean(df$X2),2), X3=c(1,2))
    - The right side of the arrow is creating a data.frame with three columns and two rows, where X1 and X2 are set to their mean, and X3 is set to 1 and 2. You have to specify the correct number of rows even where you are holding a value constant – hence the use of the rep function.
  - PPdf[, c("pred.prob")] < − predict(polr_object, newdata=PPdf, type="probs")
  - PPdf

Example Predicted Probabilities Outputs

- A three category measure of labor participation and a five-category explanatory variable yield fifteen predicted probabilities

```
> pred_data_1 <- data.frame(region=c(1, 2, 3, 4, 5))
> pred_data_1[, c("pred.prob")] <- predict(wlf_fit3,
> pred_data_1 # Read as the predicted probability of
  region pred.prob.1 pred.prob.2 pred.prob.3
1      1   0.2084221   0.5984605   0.1931175
2      2   0.2260488   0.5964814   0.1774698
3      3   0.2447054   0.5924604   0.1628342
4      4   0.2643758   0.5864376   0.1491866
5      5   0.2850307   0.5784729   0.1364964
```

Predicted Probabilities

- The complexity of this scales exponentially
- For example, a four category outcome with three binary IVs and one six category IV yields 192 predicted probabilities

0.26677983	0.15653303	0.36537812	0.21130901
0.03973677	0.03731493	0.22094511	0.70200319
0.40899500	0.17366739	0.29386556	0.12347205
0.30073676	0.16383322	0.35064871	0.18478132
0.07296376	0.06406438	0.30968637	0.55328549
0.44994543	0.17273543	0.27083653	0.10648261
0.04663262	0.04318481	0.24429911	0.66588346
0.08511467	0.07291454	0.33029190	0.51167889
0.25064652	0.15226864	0.37141200	0.22567283
0.03664746	0.03462924	0.20942169	0.71930161
0.38882205	0.17324801	0.30505893	0.13287101
0.28334348	0.16037039	0.35849199	0.19779413
0.06747275	0.05990556	0.29864389	0.57397780
0.42921956	0.17349721	0.28252562	0.11475762
0.04303122	0.04014064	0.23249236	0.68433578
0.07878693	0.06836570	0.32017676	0.53267061
0.23517593	0.14767190	0.37643713	0.24071504
0.03378987	0.03211375	0.19812284	0.73597354
0.36902282	0.17223881	0.31586954	0.14286882
0.26657267	0.15648161	0.36546010	0.21148561
0.06236717	0.05594771	0.28727088	0.59441424
0.40873897	0.17366580	0.29400848	0.12358675
0.03969637	0.03728004	0.22079884	0.70222475
0.07289214	0.06401078	0.30954978	0.55354730

0.22037939	0.14279265	0.38040011	0.25642785
0.03114790	0.02976112	0.18709431	0.75199666
0.34965500	0.17065330	0.32620595	0.15348576
0.25044761	0.15221273	0.37148166	0.22585800
0.05762405	0.05219113	0.27565611	0.61452871
0.38857034	0.17323895	0.30519760	0.13299311
0.03661008	0.03459653	0.20927794	0.71951545
0.06740613	0.05985448	0.29850257	0.57423681
0.20626275	0.13768116	0.38325828	0.27279781
0.02870637	0.02756377	0.17637509	0.76735477
0.33077080	0.16851246	0.33597671	0.16474002
0.23498544	0.14761212	0.37649373	0.24090871
0.05322118	0.04863438	0.26388501	0.63425942
0.36877619	0.17222240	0.31600281	0.14299860
0.03375531	0.03208313	0.19798222	0.73617934
0.06230525	0.05589916	0.28712598	0.59466960
0.19282670	0.13238721	0.38498057	0.28980551
0.02645099	0.02551403	0.16599754	0.78203745
0.31241656	0.16584420	0.34509193	0.17664730
0.22019744	0.14272963	0.38044304	0.25662989
0.04913718	0.04527451	0.25203875	0.65354957
0.34941414	0.17062975	0.32633266	0.15362345
0.03111595	0.02973251	0.18695737	0.75219418
0.05756655	0.05214512	0.27550871	0.61477962

Predicted Probabilities

- The complexity of this scales exponentially
- For example, a four category outcome with three binary IVs and one six category IV yields 192 predicted probabilities

0.26677983	0.15653303	0.36537812	0.21130901
0.03973677	0.03731493	0.22094511	0.70200319
0.40899500	0.17366739	0.29386556	0.12347205
0.30073676	0.16383322	0.35064871	0.18478132
0.07296376	0.06406438	0.30968637	0.55328549
0.44994543	0.17273543	0.27083653	0.10648261
0.04663262	0.04318481	0.24429911	0.66588346
0.08511467	0.07291454	0.33029190	0.51167889
0.25064652	0.15226864	0.37141200	0.22567283
0.03664746	0.03462924	0.20942169	0.71930161
0.38882205	0.17324801	0.30505893	0.13287101
0.28334348	0.16037039	0.35849199	0.19779413
0.06747275	0.05990556	0.29864389	0.57397780
0.42921956	0.17349721	0.28252562	0.11475762
0.04303122	0.04014064	0.23249236	0.68433578
0.07878693	0.06836570	0.32017676	0.53267061
0.23517593	0.14767190	0.37643713	0.24071504
0.03378987	0.03211375	0.19812284	0.73597354
0.36902282	0.17223881	0.31586954	0.14286882
0.26657267	0.15648161	0.36546010	0.21148561
0.06236717	0.05594771	0.28727088	0.59441424
0.40873897	0.17366580	0.29400848	0.12358675
0.03969637	0.03728004	0.22079884	0.70222475
0.07289214	0.06401078	0.30954978	0.55354730

0.22037939	0.14279265	0.38040011	0.25642785
0.03114790	0.02976112	0.18709431	0.75199666
0.34965500	0.17065330	0.32620595	0.15348576
0.25044761	0.15221273	0.37148166	0.22585800
0.05762405	0.05219113	0.27565611	0.61452871
0.38857034	0.17323895	0.30519760	0.13299311
0.03661008	0.03459653	0.20927794	0.71951545
0.06740613	0.05985448	0.29850257	0.57423681
0.20626275	0.13768116	0.38325828	0.27279781
0.02870637	0.02756377	0.17637509	0.76735477
0.33077080	0.16851246	0.33597671	0.16474002
0.23498544	0.14761212	0.37649373	0.24090871
0.05322118	0.04863438	0.26388501	0.63425942
0.36877619	0.17222240	0.31600281	0.14299860
0.03375531	0.03208313	0.19798222	0.73617934
0.06230525	0.05589916	0.28712598	0.59466960
0.19282670	0.13238721	0.38498057	0.28980551
0.02645099	0.02551403	0.16599754	0.78203745
0.31241656	0.16584420	0.34509193	0.17664730
0.22019744	0.14272963	0.38044304	0.25662989
0.04913718	0.04527451	0.25203875	0.65354957
0.34941414	0.17062975	0.32633266	0.15362345
0.03111595	0.02973251	0.18695737	0.75219418
0.05756655	0.05214512	0.27550871	0.61477962

- This is rarely useful nor interpretable

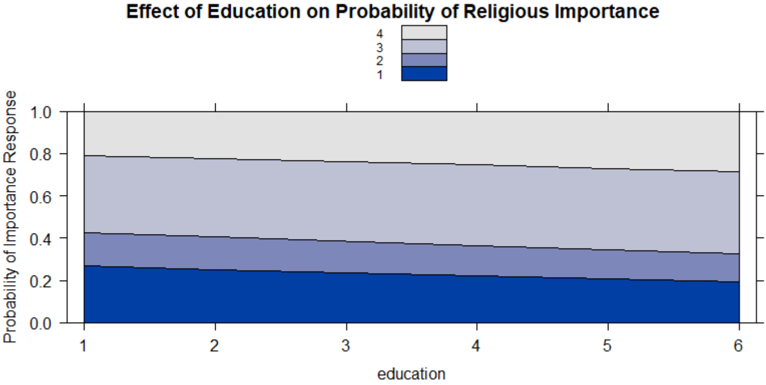
Example Predicted Probabilities Outputs

- A better approach with so many combinations is to only vary the covariate you are interested in
  - Reminder: set continuous variables to their mean and binary or categorical variables to their median
  - The values you set the covariate to need to make substantive sense – no such thing as 1.4 urban where urban residency is defined as a yes/no

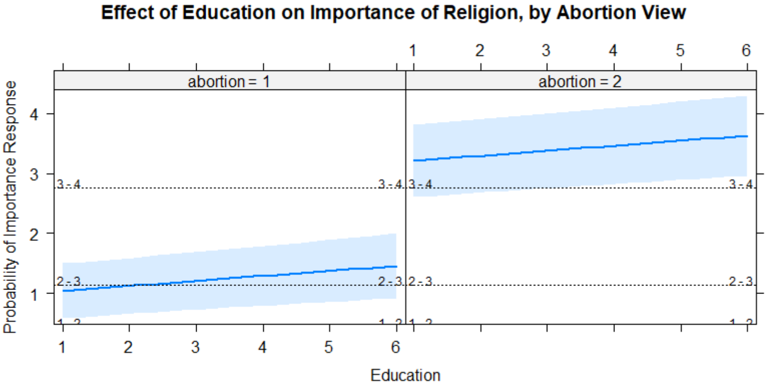
abortion	gender	urban	education	pred.prob.1	pred.prob.2	pred.prob.3	pred.prob.4
1	1	2	1	0.3007368	0.1638332	0.3506487	0.1847813
1	1	2	2	0.2833435	0.1603704	0.3584920	0.1977941
1	1	2	3	0.2665727	0.1564816	0.3654601	0.2114856
1	1	2	4	0.2504476	0.1522127	0.3714817	0.2258580
1	1	2	5	0.2349854	0.1476121	0.3764937	0.2409087
1	1	2	6	0.2201974	0.1427296	0.3804430	0.2566299

Example Output Using 'Effects' Package

- As with most analyses, graphical illustrations are often preferable to tables, especially where not easily interpretable



Example Output Using 'Effects' Package



Example Output Using GGPlot