# Example 2b: Predicting Categorical (Ordinal and Nominal) Outcomes via STATA GOLOGIT2 and MLOGIT; R GLM and VGLM; and SAS GLIMMIX and LOGISTIC

***(complete syntax data, and output available for STATA, R, and SAS electronically)***

The (fake) data for this example came from: https://stats.idre.ucla.edu/sas/dae/ordinal-logistic-regression/. In this example we will predict a student's **categorical decision** of how likely it is that they will apply to grad school (0=not, 1=eh, or 2=very) using undergraduate GPA (centered at 3.0), whether at least one of their parents has a graduate degree (0=no, 1=yes), and whether they attended a private university (0=no, 1=yes). We will examine three types of models that each use a multinomial conditional response distribution: (1) a standard "proportional odds ordinal regression" (i.e., using a "cumulative logit" link and assuming equal predictor slopes across submodels), (2) a modified ordinal regression for "non-proportional" or "partial-proportional" odds (still with a cumulative logit link, but allowing at least some different predictor slopes across submodels), and (3) a "nominal" or "multinomial" regression (i.e., using a "baseline category" or "generalized logit" link to predict each outcome category in relation to a reference category).

For the polychoric and polyserial correlations, I am using a user-created STATA command POLYCHORIC and POLYCOR in R. For the predictive models, the standard STATA package for ordinal regression, OLOGIT, provides thresholds instead of intercepts and it does not have any means to test or specify non-proportional odds models. To solve these problems, we will be using the user-created STATA program GOLOGIT2. In R, we will be using GLM and VGLM (the latter is from the VGAM package). I chose VGLM over other R functions (such as CLM from ORDINAL and POLR from MASS) because it can fit non-proportional odds, allows intercepts instead of thresholds, and works with GLHT for linear combinations of the model fixed effects. Unfortunately, because the VGLM function uses expected information instead of observed information (as used in STATA and SAS), the standard errors for the parameter estimates (and thus any Wald test results) will differ between STATA/SAS and R. Likelihood ratio test results are the same, however. Btw, in SAS GLIMMIX, I set denominator DF to "none" so that the SAS Wald test results will match those of STATA.

For syntax for importing and preparing the data (and which packages were used), please see PSQF 6270 Example 2a.

## STATA and R Syntax and Output for Descriptive Statistics:

```
pwcorr apply3 grad priv gpa3, sig  // STATA: Pearson correlations

             |   apply3     grad     priv     gpa3
-------------+------------------------------------
      apply3 |   1.0000 
             |
             |
        grad |   0.2190   1.0000 
             |   0.0000
             |
        priv |  -0.0497  -0.0790   1.0000 
             |   0.3213   0.1148
             |
        gpa3 |   0.1526   0.1856  -0.2275   1.0000 
             |   0.0022   0.0002   0.0000
             |
```

```
cor(x=Example2a) # R Pearson correlations

             apply3         grad         priv        gpa3
apply3  1.000000000  0.219036320 -0.049713226  0.15257848
grad    0.219036320  1.000000000 -0.078974399  0.18559072
priv   -0.049713226 -0.078974399  1.000000000 -0.22747377
gpa3    0.152578477  0.185590719 -0.227473769  1.00000000
```

Next, let's examine **polychoric** correlations (between ordinal variables with ≤ 10 categories) or **polyserial** correlations (between an ordinal variable and a continuous variable with > 10 categories), computed here without *p*-values:

```
polychoric apply3 grad priv gpa3, pw  // STATA: Polychoric or Polyserial (>10 options) correlations

            apply3        grad        priv        gpa3
apply3           1
  grad    .3599378           1
  priv  -.07800662  -.16969222           1
  gpa3   .17918182   .27952343  -.35043179           1
```

```
# Recognize categorical variables as factor variables
Example2b$apply3 = as.factor(Example2b$apply3)
Example2b$grad   = as.factor(Example2a$grad)
Example2b$priv   = as.factor(Example2a$priv)
print("hetcor determines correlation type based on variable type")
hetcor(data=Example2b, ML=TRUE, std.err=TRUE, use="pairwise.complete.obs")

Correlations/Type of Correlation:
          apply3       grad       priv       gpa3
apply3         1 Polychoric Polychoric Polyserial
grad     0.35927          1 Polychoric Polyserial
priv    -0.07792   -0.16975          1 Polyserial
gpa3     0.17895    0.27905   -0.35099          1
```

Most of the relations among variables are stronger when indexed by these correlations that use a bivariate normal distribution to describe what the correlation would be for their "underlying" unobserved continuous distributions:

- **Tetrachoric** = binary with binary (as a special case of "polychoric" here)
- **Polychoric** = ordinal with ordinal
- **Biserial** = binary with continuous (as a special case of "polyserial" here)
- **Polyserial** = ordinal with continuous

```
tabulate apply3  // STATA frequencies and proportions

     apply3:|      Freq.     Percent        Cum.
------------+-----------------------------------
      Not 0 |        220       55.00       55.00
       Eh 1 |        140       35.00       90.00
     Very 2 |         40       10.00      100.00
------------+-----------------------------------
      Total |        400      100.00
```

So now we know that **55% of the respondents have apply3=0, 35% have apply3=1, and 10% have apply3=2**. This information will come in handy in making sure we understand which value our categorical regression models are predicting!

Btw, I did not add value labels to this outcome to keep the code transferable to other outcomes.

```
# R frequencies and proportions
prop.table(table(x=Example2$apply3))

   0    1    2
0.55 0.35 0.10
```

---

## Clarifying the outcomes to be predicted in each binary CUMULATIVE submodel ($y_i = 0, 1, \text{or } 2$):

$$\log\left(\frac{\text{prob}(Apply3_i\text{=1or2})}{\text{prob}(Apply3_i\text{=0})}\right) = \text{logit}(Apply3_i > 0), \quad \log\left(\frac{\text{prob}(Apply3_i\text{=2})}{\text{prob}(Apply3_i\text{=0or1})}\right) = \text{logit}(Apply3_i > 1)$$

## Empty Ordinal Model predicting the cumulative logit of 3-category apply using INTERCEPTS:

$$\log\left(\frac{\text{prob}(Apply3_i\text{=1or2})}{\text{prob}(Apply3_i\text{=0})}\right) = \beta_{00} \rightarrow \text{prob}(Apply3_i > 0) = \frac{\exp(\beta_{00})}{1+\exp(\beta_{00})} = \frac{\exp(-0.2007)}{[1+\exp(-0.2007)]} \;/ = .450$$

$$\log\left(\frac{\text{prob}(Apply3_i\text{=2})}{\text{prob}(Apply3_i\text{=0or1})}\right) = \beta_{01} \rightarrow \text{prob}(Apply3_i > 1) = \frac{\exp(\beta_{01})}{1+\exp(\beta_{01})} = \frac{\exp(-2.1972)}{[1+\exp(-2.1972)]} \;/ = .100$$

## STATA Syntax and Partial Output for Empty Ordinal Model using GOLOGIT2—*which values are being predicted?*

```
display "STATA Empty Model Predicting Ordinal Apply3"
display "GOLOGIT2 Gives Intercepts (Logit of Higher Category), not Thresholds"
gologit2 apply3, nolog

Generalized Ordered Logit Estimates                   Number of obs =       400
                                                      LR chi2(0)    =     -0.00
                                                      Prob > chi2   =         .
Log likelihood = -370.60264                           Pseudo R2     =   -0.0000
------------------------------------------------------------------------------
      apply3 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0      _cons |  -.2006707   .1005038    -2.00   0.046    -.3976545   -.0036869 → intercept for y>0
1      _cons |  -2.197225   .1666667   -13.18   0.000    -2.523885   -1.870564 → intercept for y>1
------------------------------------------------------------------------------
display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 741.20528

margins          // All 3 probabilities
------------------------------------------------------------------------------
             |            Delta-method
    _predict |     Margin   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
           1 |        .55   .0247747    22.11   0.000     .5012465    .5987535
           2 |        .35   .0238485    14.68   0.000     .3032578    .3967422
           3 |         .1       .015     6.67   0.000     .0706005    .1293995
------------------------------------------------------------------------------
```

**Proportions from data:**
0 1 2
0.55 0.35 0.10

**Converted logits:**
$\text{prob}(y_i > 0) = .45$
$\text{prob}(y_i > 1) = .10$

**Margins** computes predicted probability of each response (not just for the probability for each submodel)

## For comparison, using STATA OLOGIT instead (which is more common, but it gives thresholds):

```
display "STATA Empty Model Predicting Ordinal Apply3 Using OLOGIT Instead"
display "OLOGIT Gives Thresholds (Logit of Lower Category), not Intercepts"
ologit apply3, nolog

Ordered logistic regression                           Number of obs =       400
Log likelihood = -370.60264                           Pseudo R2     =   -0.0000
------------------------------------------------------------------------------
      apply3 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       /cut1 |   .2006707   .1005038                      .0036869    .3976545 → threshold for y<1
       /cut2 |   2.197225   .1666667                      1.870564    2.523885 → threshold for y<2
------------------------------------------------------------------------------
```

**Proportions from data:**
0 1 2
0.55 0.35 0.10

**Converted logits:**
$\text{prob}(y_i < 1) = .55$
$\text{prob}(y_i < 2) = .90$

## R Syntax and Partial Output for Empty Ordinal Model—*which values are being predicted?*

```
print("R Empty Model Predicting Ordinal Apply3")
Model3Empty = vglm(data=Example2, family=cumulative(link="logitlink",reverse=TRUE,parallel=TRUE),
                   formula=apply3~1); summary(Model3Empty)

Coefficients:
              Estimate Std. Error  z value Pr(>|z|)
(Intercept):1 -0.20067    0.10050  -1.9966  0.04586 → logit of y>0
(Intercept):2 -2.19722    0.16667 -13.1833  < 2e-16 → logit of y>1


Names of linear predictors: logitlink(P[Y>=2]), logitlink(P[Y>=3])

Residual deviance: 741.20528 on 798 degrees of freedom → model -2LL
Log-likelihood: -370.60264 on 798 degrees of freedom   → model LL instead (like STATA)

print("Convert logits to probability to check interpretation")
1/(1+exp(-1*coefficients(Model3Empty)))
(Intercept):1 (Intercept):2
         0.45          0.10
```

**Reverse=TRUE** provides **intercepts** (for y>0 and y>1) instead of thresholds

**Um, NO.** These CANNOT be the "names" of the linear predictors…

**Proportions from data:**
0 1 2
0.55 0.35 0.10

**Proportional Odds Ordinal Model with 3 Predictors—*to what extent do undergraduate GPA, parent education, and undergraduate type uniquely predict a "higher" decision to apply to graduate school?***

$$\log\left(\frac{\text{prob}(Apply3_i = 1\text{or}2)}{\text{prob}(Apply3_i = 0)}\right) = \beta_{00} + \beta_1(GPA_i - 3) + \beta_2(Grad_i) + \beta_3(Private_i)$$

$$\log\left(\frac{\text{prob}(Apply3_i = 2)}{\text{prob}(Apply3_i = 0\text{or}1)}\right) = \beta_{01} + \beta_1(GPA_i - 3) + \beta_2(Grad_i) + \beta_3(Private_i)$$

## STATA Syntax and Partial Output:

```
display "STATA Proportional Odds Model Predicting Ordinal Apply3"
gologit2 apply3 c.gpa3 c.grad c.priv, pl nolog

Generalized Ordered Logit Estimates                Number of obs =     400
                                                   LR chi2(3)    =   24.18 → LRT for MODEL
                                                   Prob > chi2   = 0.0000
Log likelihood = -358.51244                        Pseudo R2     = 0.0326
------------------------------------------------------------------------------
      apply3 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0            |
        gpa3 |   .6157458   .2606311     2.36   0.018     .1049183    1.126573  beta1
        grad |   1.047664   .2657891     3.94   0.000     .5267266    1.568601  beta2
        priv |   .0586828   .2978589     0.20   0.844    -.5251098    .6424754  beta3
       _cons |  -.4147686   .2829697    -1.47   0.143     -.969379    .1398418  beta00
-------------+----------------------------------------------------------------
1            |
        gpa3 |   .6157458   .2606311     2.36   0.018     .1049183    1.126573  beta1
        grad |   1.047664   .2657891     3.94   0.000     .5267266    1.568601  beta2
        priv |   .0586828   .2978589     0.20   0.844    -.5251098    .6424754  beta3
       _cons |  -2.510213   .3191656    -7.86   0.000    -3.135766    -1.88466  beta01
------------------------------------------------------------------------------
display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 717.02487

display "Get Odds Ratios Instead of Logit Fixed Effects"
gologit2 apply3 c.gpa3 c.grad c.priv, pl or nolog
------------------------------------------------------------------------------
      apply3 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0            |
        gpa3 |   1.851037   .4824377     2.36   0.018      1.11062    3.085067  exp(beta1)
        grad |   2.850983   .7577602     3.94   0.000      1.69338    4.799927  exp(beta2)
        priv |   1.060439   .3158611     0.20   0.844     .5914904    1.901181  exp(beta3)
       _cons |   .6604931   .1868995    -1.47   0.143     .3793185    1.150092  exp(beta00)
-------------+----------------------------------------------------------------
1            |
        gpa3 |   1.851037   .4824377     2.36   0.018      1.11062    3.085067  exp(beta1)
        grad |   2.850983   .7577602     3.94   0.000      1.69338    4.799927  exp(beta2)
        priv |   1.060439   .3158611     0.20   0.844     .5914904    1.901181  exp(beta3)
       _cons |   .0812509   .0259325    -7.86   0.000     .0434665    .1518807  exp(beta01)
------------------------------------------------------------------------------
```

## R Syntax and Partial Output

```
print("R Proportional Odds Model Predicting Ordinal Apply3")
Model3PO = vglm(data=Example2, family=cumulative(link="logitlink",reverse=TRUE,parallel=TRUE),
                formula=apply3~1+gpa3+grad+priv); summary(Model3PO)

Coefficients:
               Estimate Std. Error z value  Pr(>|z|)
(Intercept):1 -0.414757   0.273224 -1.5180   0.12901 beta00
(Intercept):2 -2.510201   0.310320 -8.0891 6.013e-16 beta01
gpa3           0.615754   0.262578  2.3450   0.01903 beta1
grad           1.047655   0.268448  3.9026 9.515e-05 beta2
priv           0.058672   0.288610  0.2033   0.83891 beta3
```

**Interpret the intercept 1:**

**Interpret the intercept 2:**

**Interpret the slope for gpa3:**

**Interpret the slope for grad:**

**Interpret the slope for private:**

```
Residual deviance: 717.02487 on 795 degrees of freedom → model -2LL
Log-likelihood: -358.51244 on 795 degrees of freedom   → model   LL

Exponentiated coefficients:
     gpa3      grad      priv
1.8510513 2.8509581 1.0604268 → exp(beta)

print("Likelihood Ratio Test of Predictors")
print("Analogous to F-test for model R2 in general LM")
anova(Model3Empty, Model3PO, type=1) # Nested "fewer" model goes first

Analysis of Deviance Table
Model 1: apply3 ~ 1
Model 2: apply3 ~ 1 + gpa3 + grad + priv
  Resid. Df Resid. Dev Df Deviance    Pr(>Chi)
1       798    741.205
2       795    717.025  3  24.1804 0.000022905

print("Get odds ratios with 95% CIs")
exp(cbind(OR = coefficients(Model3PO), confint.default(Model3PO)))

[                       OR       2.5 %     97.5 %
(Intercept):1 0.660500671 0.386638232 1.12834454  exp(beta00)
(Intercept):2 0.081251906 0.044227087 0.14927215  exp(beta01)
gpa3          1.851051312 1.106397837 3.09688870  exp(beta1)
grad          2.850958157 1.684562648 4.82496892  exp(beta2)
priv          1.060426845 0.602303375 1.86700779  exp(beta3)
```

**These ordinal models rely on an assumption of <u>proportional odds</u>: that all predictor slopes are equal across sub-models. Next is an alternative, <u>a non-proportional odds model</u>, which allows us to test the difference between each predictor slope across submodels:**

---

**Non-Proportional Odds Model with 3 Predictors—*to what extent do undergraduate GPA, parent education, and undergraduate school type uniquely predict a "higher" decision to apply to graduate school (differently across submodels)?***

$$\log\left(\frac{\text{prob}(Apply3_i = 1\text{or}2)}{\text{prob}(Apply3_i = 0)}\right) = \beta_{00} + \beta_{10}(GPA_i - 3) + \beta_{20}(Grad_i) + \beta_{30}(Private_i)$$

$$\log\left(\frac{\text{prob}(Apply3_i = 2)}{\text{prob}(Apply3_i = 0\text{or}1)}\right) = \beta_{01} + \beta_{11}(GPA_i - 3) + \beta_{21}(Grad_i) + \beta_{31}(Private_i)$$

## <u>STATA</u> Syntax and Partial Output:

```
display "STATA Non-Proportional Odds Model Predicting Ordinal Apply3"
display "Directly provides each slope and differences in slopes across submodels"
gologit2 apply3 c.gpa3 c.grad c.priv, gamma nolog
```

```
Generalized Ordered Logit Estimates                      Number of obs =     400
                                                         LR chi2(6)    =   28.19 → LRT for MODEL
                                                         Prob > chi2   = 0.0001
Log likelihood = -356.50556                              Pseudo R2     = 0.0380
------------------------------------------------------------------------------
       apply |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
0            |
        gpa3 |   .5920653   .2690337     2.20   0.028     .0647689    1.119362  beta10
        grad |   1.083129   .2959475     3.66   0.000     .5030823    1.663175  beta20
        priv |   .2307488   .3062506     0.75   0.451    -.3694912    .8309889  beta30
       _cons |  -.5684777   .2888819    -1.97   0.049    -1.134676   -.0022796  beta00
-------------+----------------------------------------------------------------
1            |
        gpa3 |   .7190314   .4536953     1.58   0.113    -.1701951    1.608258  beta11
        grad |   .9946781   .3740984     2.66   0.008     .2614588    1.727897  beta21
        priv |  -.5366997   .4293132    -1.25   0.211    -1.378138    .3047388  beta31
       _cons |  -2.027556    .405012    -5.01   0.000    -2.821365   -1.233747  beta01
------------------------------------------------------------------------------

Alternative parameterization: Gammas are deviations from proportionality → Slope differences directly!
------------------------------------------------------------------------------
       apply |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
beta         |
        gpa3 |   .5920653   .2690337     2.20   0.028     .0647689    1.119362  beta10
        grad |   1.083129   .2959475     3.66   0.000     .5030823    1.663175  beta20
        priv |   .2307488   .3062506     0.75   0.451    -.3694912    .8309889  beta30
-------------+----------------------------------------------------------------
Gamma_2      |
        gpa3 |   .1269661   .4383381     0.29   0.772    -.7321607     .986093  beta11 – beta10
        grad |  -.0884506   .3871321    -0.23   0.819    -.8472157    .6703144  beta21 – beta20
        priv |  -.7674485   .4056115    -1.89   0.058    -1.562432    .0275354  beta31 – beta30
-------------+----------------------------------------------------------------
Alpha        |
     _cons_1 |  -.5684777   .2888819    -1.97   0.049    -1.134676   -.0022796  beta00
     _cons_2 |  -2.027556    .405012    -5.01   0.000    -2.821365   -1.233747  beta01
------------------------------------------------------------------------------
display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 713.01111

estimates store NPO        // Save for LRT
lrtest NPO PO              // LRT for overall proportional odds ("fewer" model goes LAST)

Likelihood-ratio test                                 LR chi2(3)  =      4.01
(Assumption: PO nested in NPO)                        Prob > chi2 =    0.2600
```

## R Syntax and Partial Output:

```
print("R Non-Proportional Odds Model Predicting Ordinal Apply3")
Model3NPO = vglm(data=Example2, family=cumulative(link="logitlink",reverse=TRUE,parallel=FALSE),
                 formula=apply3~1+gpa3+grad+priv); summary(Model3NPO)
```

parallel=FALSE → nonproportional odds

```
Coefficients:
              Estimate Std. Error z value  Pr(>|z|)
(Intercept):1 -0.56848    0.28717 -1.9796 0.0477492  beta00
(Intercept):2 -2.02757    0.39878 -5.0845 3.686e-07  beta01

gpa3:1         0.59207    0.27247  2.1729 0.0297843  beta10
gpa3:2         0.71902    0.45280  1.5879 0.1123017  beta11

grad:1         1.08312    0.29826  3.6314 0.0002819  beta20
grad:2         0.99470    0.37695  2.6388 0.0083192  beta21

priv:1         0.23075    0.30485  0.7569 0.4491039  beta30
priv:2        -0.53669    0.42006 -1.2776 0.2013748  beta31

Residual deviance: 713.01111 on 792 degrees of freedom → Model -2LL
Log-likelihood: -356.50556 on 792 degrees of freedom   → Model   LL
```

```
Exponentiated coefficients:
   gpa3:1    gpa3:2    grad:1    grad:2    priv:1    priv:2
1.8077234 2.0524197 2.9538950 2.7039030 1.2595402 0.5846818   exp(beta)

print("Likelihood Ratio Test for Overall Proportional Odds")
anova(Model3PO, Model3NPO, type=1) # Nested "fewer" model goes first

Analysis of Deviance Table
Model 1: apply3 ~ 1 + gpa3 + grad + priv
Model 2: apply3 ~ 1 + gpa3 + grad + priv
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       795    717.025
2       792    713.011  3  4.01376  0.25998

print("Univ Wald tests of submodel slope differences")
glhtModel3NPO = glht(model=Model3NPO, linfct=rbind(
  "gpa3 slope diff" = c(0,0,-1,1, 0,0, 0,0),  # in order of fixed effects
  "grad slope diff" = c(0,0, 0,0,-1,1, 0,0),
  "priv slope diff" = c(0,0, 0,0, 0,0,-1,1)))
glhtSummaryOR(glhtModel3NPO) # print summary adding -2LL and odds ratios

                    Estimate    Std.Err     z.value p.value Odds.Ratio
gpa3 slope diff  0.126951177 0.44027051  0.28834812  0.7731 1.13536158  beta11 – beta10
grad slope diff -0.088428374 0.39015315 -0.22665042  0.8207 0.91536867  beta21 – beta20
priv slope diff -0.767434293 0.39542539 -1.94078151  0.0523 0.46420255  beta31 – beta30
```

---

Both SAS PROC LOGISTIC and STATA GOLOGIT2 can automate the selection of which slopes should differ—see the online files for what happens when we let them do it while requesting that all predictors remain in the model even if nonsignificant. But I did not try to figure this out in R…

**Here is the final model they came up with—now only the <u>slope for private differs</u> across submodels:**

$$\log\left(\frac{\text{prob}(Apply3_i = 1\text{or}2)}{\text{prob}(Apply3_i = 0)}\right) = \beta_{00} + \beta_1(GPA_i - 3) + \beta_2(Grad_i) + \boldsymbol{\beta_{30}}(Private_i)$$

$$\log\left(\frac{\text{prob}(Apply3_i = 2)}{\text{prob}(Apply3_i = 0\text{or}1)}\right) = \beta_{01} + \beta_1(GPA_i - 3) + \beta_2(Grad_i) + \boldsymbol{\beta_{31}}(Private_i)$$

***Here is how to specify this same model in which YOU select which slopes are held equal:***

**<u>STATA</u> Syntax and Partial Output (npl = non-proportional odds only for private slope):**

```
display "STATA Partial Proportional Odds Model Predicting Ordinal Apply3"
gologit2 apply3 c.gpa3 c.grad c.priv, npl(c.priv) gamma nolog

Generalized Ordered Logit Estimates                   Number of obs =     400
                                                      LR chi2(4)    =   28.06 → LRT for MODEL
                                                      Prob > chi2   = 0.0000
Log likelihood = -356.57077                           Pseudo R2     = 0.0379
------------------------------------------------------------------------------
      apply3 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0            |
        gpa3 |   .6105983   .2607849     2.34   0.019     .0994694    1.121727  beta1
        grad |   1.057633   .2665412     3.97   0.000     .5352216    1.580044  beta2
        priv |   .2350038   .3052548     0.77   0.441    -.3632847    .8332922  beta30
       _cons |  -.5690629   .2876884    -1.98   0.048    -1.132922   -.005204  beta00
-------------+----------------------------------------------------------------
1            |
        gpa3 |   .6105983   .2607849     2.34   0.019     .0994694    1.121727  beta1
        grad |   1.057633   .2665412     3.97   0.000     .5352216    1.580044  beta2
        priv |  -.5732671   .4106292    -1.40   0.163    -1.378086    .2315513  beta31
       _cons |  -2.005542     .37073    -5.41   0.000     -2.73216   -1.278925  beta01
------------------------------------------------------------------------------
```

```
Alternative parameterization: Gammas are deviations from proportionality → Slope differences directly!
------------------------------------------------------------------------------
      apply3 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
beta         |
        gpa3 |   .6105983   .2607849     2.34   0.019     .0994694    1.121727  beta1
        grad |   1.057633   .2665412     3.97   0.000     .5352216    1.580044  beta2
        priv |   .2350038   .3052548     0.77   0.441    -.3632847    .8332922  beta30
-------------+----------------------------------------------------------------
Gamma_2      |
        priv |  -.8082709   .3780655    -2.14   0.033    -1.549266   -.0672762  beta31 - beta30
-------------+----------------------------------------------------------------
Alpha        |
     _cons_1 |  -.5690629   .2876884    -1.98   0.048    -1.132922    -.005204  beta00
     _cons_2 |  -2.005542     .37073    -5.41   0.000     -2.73216   -1.278925  beta01
------------------------------------------------------------------------------

display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 713.14154

display "Get Odds Ratios Instead of Logit Fixed Effects"
gologit2 apply3 c.gpa3 c.grad c.priv, npl(c.priv) gamma or nolog
------------------------------------------------------------------------------
      apply3 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0            |
        gpa3 |   1.841533    .480244     2.34   0.019     1.104585    3.070153  exp(beta1)
        grad |   2.879546   .7675177     3.97   0.000     1.707827    4.855169  exp(beta2)
        priv |   1.264914   .3861209     0.77   0.441     .6953885    2.300881  exp(beta30)
       _cons |   .5660557   .1628476    -1.98   0.048     .3220908    .9948095  exp(beta00)
-------------+----------------------------------------------------------------
1            |
        gpa3 |   1.841533    .480244     2.34   0.019     1.104585    3.070153  exp(beta1)
        grad |   2.879546   .7675177     3.97   0.000     1.707827    4.855169  exp(beta2)
        priv |   .5636808   .2314638    -1.40   0.163     .2520606    1.260554  exp(beta31)
       _cons |   .1345873   .0498956    -5.41   0.000     .0650786    .2783364  exp(beta01)
------------------------------------------------------------------------------
```

## R Syntax and Partial Output (FALSE~priv → non-proportional odds only for private slope):

```
print("R Partial Proportional Odds Model Predicting Ordinal Apply3")
Model3CPO = vglm(data=Example2, family=cumulative(link="logitlink",reverse=TRUE,parallel=FALSE~priv),
                 formula=apply3~1+gpa3+grad+priv); summary(Model3CPO)

Coefficients:
              Estimate Std. Error z value  Pr(>|z|)
(Intercept):1 -0.56906    0.28652 -1.9861   0.04702  beta00
(Intercept):2 -2.00553    0.37084 -5.4081 6.370e-08  beta01
gpa3           0.61061    0.26289  2.3227   0.02019  beta1
grad           1.05763    0.26920  3.9288 8.536e-05  beta2
priv:1         0.23501    0.30433  0.7722   0.43998  beta30
priv:2        -0.57328    0.40935 -1.4004   0.16138  beta31

Residual deviance: 713.14154 on 794 degrees of freedom → model -2LL
Log-likelihood: -356.57077 on 794 degrees of freedom   → model  LL

Exponentiated coefficients:
      gpa3       grad     priv:1     priv:2
1.84155529 2.87952956 1.26491688 0.56367392 → exp(beta)

print("Univ Wald test of submodel slope difference")
glhtModel3CPO = glht(model=Model3CPO, linfct=rbind(
  "priv Slope PO" = c(0,0,0,0,-1,1)))
glhtSummaryOR(glhtModel3CPO) # print summary adding -2LL and odds ratios

                Estimate   Std.Err    z.value p.value Odds.Ratio
priv Slope PO -0.80828577 0.3792699 -2.1311625  0.0331 0.44562131  beta31 - beta30
```

## STATA Code for Getting Predicted Outcomes for Fake People via Usual MARGINS:

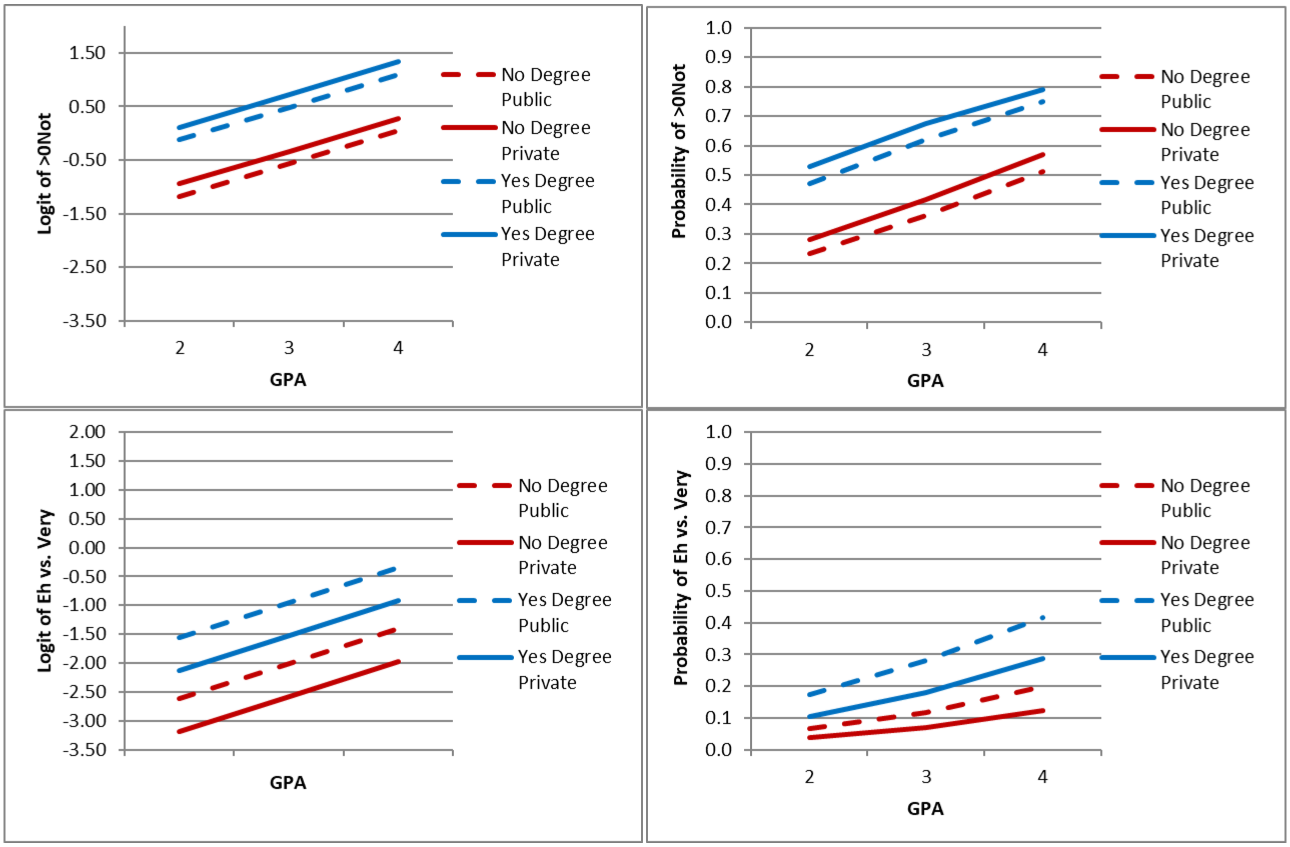
```
margins, at(c.gpa3=(-1(1)1) c.grad=(0(1)1) c.priv=(0(1)1)) predict(xb) // Yhat>0 in logits
margins, at(c.gpa3=(-1(1)1) c.grad=(0(1)1) c.priv=(0(1)1))      // Each Yhat in probability
```

## R Code for Getting Predicted Outcomes for Fake People via PREDICT (so that I can get predicted probabilities for all outcome categories):

```
# Create fake people for use in generating predicted outcomes
FakeGpa3 = c(-1,0,1,-1,0,1,-1,0,1,-1,0,1)
FakeGrad = c( 0,0,0, 0,0,0, 1,1,1, 1,1,1)
FakePriv = c( 0,0,0, 1,1,1, 0,0,0, 1,1,1)
# Create dataset using just-created columns and constants for other model variables
FP = data.frame(gpa3=FakeGpa3, grad=FakeGrad, priv=FakePriv)

print("Get Yhat for specific values of predictors in fake people")
print("Y column = predicted yhat, Yprob = predicted probability")
PredCPO = data.frame(FP, Y=predict(object=Model3CPO, newdata=FP, type="link"),
                     Yprob=predict(object=Model3CPO, newdata=FP, type="response"))
print("Rename columns into something meaningful")
names(PredCPO)[names(PredCPO)=='Y.logitlink.P.Y..2..']='YlogitGT0'
names(PredCPO)[names(PredCPO)=='Y.logitlink.P.Y..3..']='YlogitGT1'; PredCPO
```

See the excel file online for the plots below!

```
   gpa3 grad priv    YlogitGT0   YlogitGT1    Yprob.0    Yprob.1     Yprob.2
1    -1    0    0 -1.179675381 -2.61614217 0.76488943 0.16700383 0.068106736
2     0    0    0 -0.569064907 -2.00553169 0.63854738 0.24282927 0.118623352
3     1    0    0  0.041545567 -1.39492122 0.48961510 0.31176162 0.198623274

4    -1    0    1 -0.944668969 -3.18942152 0.72004180 0.24039244 0.039565756
5     0    0    1 -0.334058495 -2.57881105 0.58274654 0.34673884 0.070514618
6     1    0    1  0.276551980 -1.96820057 0.43129931 0.44611840 0.122582294

7    -1    1    0 -0.122048440 -1.55851523 0.53047429 0.29566590 0.173859805
8     0    1    0  0.488562034 -0.94790475 0.38023237 0.34046124 0.279306388
9     1    1    0  1.099172508 -0.33729428 0.24989497 0.33363815 0.416466878

10   -1    1    1  0.112957972 -2.13179458 0.47179050 0.42216476 0.106044746
11    0    1    1  0.723568447 -1.52118411 0.32660767 0.49410511 0.179287220
12    1    1    1  1.334178921 -0.91057363 0.20846896 0.50464857 0.286882468
```




For public versus private school, there is a positive slope in the first submodel (for y>0) as indicated by higher solid lines, but there is a negative slope in the second submodel (for y>1) as indicated by lower solid lines.

---

**Let's examine one last set of models—treating our 3-category outcome as "nominal" or "multinomial"** instead (i.e., unordered categories in which one category is the reference against which to compare each other category). For comparison with the prior ordinal models, we will choose Apply3=1 ("eh" in the middle) to be the reference outcome category. Although the empty ordinal and nominal models are equivalent, the conditional (predictor) models are not.

**Clarifying the outcomes to be predicted in each *conditional* binary submodel ($y_i = 0, 1, \text{or } 2$):**

$$\log\left(\frac{\text{prob}(Apply3=0)}{\text{prob}(Apply3_i=1)}\right) = \text{logit}(Apply3_i = 0 \text{ instead of } 1) \rightarrow \text{Only for responses of 0 or 1 (ignoring 2)}$$

$$\log\left(\frac{\text{prob}(Apply3=2)}{\text{prob}(Apply3_i=1)}\right) = \text{logit}(Apply3_i = 2 \text{ instead of } 1) \rightarrow \text{Only for responses of 2 or 1 (ignoring 0)}$$

**Empty Model Predicting Apply3 as Nominal—*which values are being predicted?***

$$\log\left(\frac{\text{prob}(Apply3=0)}{\text{prob}(Apply3_i=1)}\right) = \beta_{00} \rightarrow \text{prob}(Apply_i = 0 \text{ instead of } 1) = \frac{\exp(\beta_{00})}{1+\exp(\beta_{00})}$$

$$\log\left(\frac{\text{prob}(Apply3=2)}{\text{prob}(Apply3_i=1)}\right) = \beta_{02} \rightarrow \text{prob}(Apply_i = 2 \text{ instead of } 1) = \frac{\exp(\beta_{02})}{1+\exp(\beta_{02})}$$

**STATA Syntax and Partial Output:**

```
display "STATA Empty Model Predicting Nominal Apply3 -- ref is SECOND category of y=1"
mlogit apply3, baseoutcome(1) nolog
```

```
Multinomial logistic regression                         Number of obs =     400
                                                        LR chi2(0)    =    0.00
                                                        Prob > chi2   =       .
Log likelihood = -370.60264 * -2 = -2LL                 Pseudo R2     = 0.0000
------------------------------------------------------------------------------
      apply3 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0      _cons |   .4519851   .1081125     4.18   0.000     .2400885    .6638817 → logit of 0 vs 1
-------------+---------------------------------------------------------------- → prob = .6111
1            |  (base outcome)
-------------+----------------------------------------------------------------
2      _cons |  -1.252763   .1792843    -6.99   0.000    -1.604154   -.9013722 → logit of 2 vs 1
------------------------------------------------------------------------------ → prob = .2222

display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 741.20528 → Same as empty ordinal model!

margins   // All 3 probabilities → Put back together again, same as empty ordinal model!
------------------------------------------------------------------------------
 Marginal    |            Delta-method
 Probability |     Margin   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          1  |        .55   .0248747    22.11   0.000     .5012465    .5987535
          2  |        .35   .0238485    14.68   0.000     .3032578    .3967422
          3  |         .1       .015     6.67   0.000     .0706005    .1293995
------------------------------------------------------------------------------
```

***Given that y* = 0 or *y* = 1 :**

$$\text{prob}(Apply_i = 0) = \frac{\exp(0.4520)}{[1+\exp(0.4520)]} = .6111$$

***Given that y* = 2 or *y* = 1 :**

$$\text{prob}(Apply_i = 2) = \frac{\exp(-1.2528)}{[1+\exp(-1.2528)]} = .2222$$

```
     apply3:|      Freq.     Percent        Cum.
------------+-----------------------------------
      Not 0 |        220       55.00       55.00
       Eh 1 |        140       35.00       90.00
     Very 2 |         40       10.00      100.00
------------+-----------------------------------

Prob that y=0 or 1: .90, so y=0 is .55/.90 = .6111
Prob that y=2 or 1: .45, so y=2 is .10/.45 = .2222
```

## R Syntax and Partial Output:

```
print("R Empty Model Predicting Nominal Apply3 -- ref is SECOND category of y=1")
Model3NomEmpty = vglm(data=Example2, family=multinomial(refLevel=2),reverse=TRUE,
                      formula=apply3~1); summary(Model3NomEmpty)

Coefficients:
              Estimate Std. Error z value  Pr(>|z|)
(Intercept):1  0.45199    0.10811  4.1807 2.906e-05 → logit of 0 vs 1
(Intercept):2 -1.25276    0.17928 -6.9876 2.797e-12 → logit of 2 vs 1

Names of linear predictors: log(mu[,1]/mu[,2]), log(mu[,3]/mu[,2])
```

"Name" is correct only IF you re-order the 0,1,2 as 1,2,3… (ugh)

```
Residual deviance: 741.20528 on 798 degrees of freedom → model -2LL → Same as empty ordinal model!
Log-likelihood: -370.60264 on 798 degrees of freedom   → model LL instead (like STATA)

Reference group is level  2  of the response → so y=1 is reference (in refLevel=2)

print("Convert logits to probability to check interpretation")
1/(1+exp(-1*coefficients(Model3NomEmpty)))

(Intercept):1 (Intercept):2
   0.61111111    0.22222222
```

**Nominal Model with 3 Predictors—*to what extent do undergraduate GPA, parent education, and undergraduate school type uniquely predict each kind decision to apply to graduate school (differently across submodels)?***

$$\log\left(\frac{\text{prob}(Apply3=0)}{\text{prob}(Apply3_i=1)}\right)=\beta_{00}+\beta_{10}(GPA_i-3)+\beta_{20}(Grad_i)+\beta_{30}(Private_i)$$

$$\log\left(\frac{\text{prob}(Apply3=2)}{\text{prob}(Apply3_i=1)}\right)=\beta_{02}+\beta_{12}(GPA_i-3)+\beta_{22}(Grad_i)+\beta_{32}(Private_i)$$

## <u>STATA</u> Syntax and Partial Output

```
display "STATA 3-Predictor Model Predicting Nominal Apply3 -- ref is SECOND category of y=1"
mlogit apply3 c.gpa3 c.grad c.priv, baseoutcome(1) nolog

Multinomial logistic regression                         Number of obs =     400
                                                        LR chi2(6)    =   27.21 → LRT for MODEL
                                                        Prob > chi2   = 0.0001
Log likelihood = -356.99698                             Pseudo R2     = 0.0367
------------------------------------------------------------------------------
      apply3 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0       gpa3 |  -.4487507   .2902058    -1.55   0.122    -1.017544    .1200421  beta10
        grad |  -.9516468   .3170624    -3.00   0.003    -1.573078   -.3302159  beta20
        priv |  -.4188184   .3432943    -1.22   0.222    -1.091663    .2540261  beta30
       _cons |   .9515263   .3258247     2.92   0.003     .3129217    1.590131  beta00
-------------+----------------------------------------------------------------
1            |  (base outcome)
-------------+----------------------------------------------------------------
2       gpa3 |   .4752888   .4871448     0.98   0.329    -.4794974    1.430075  beta12
        grad |   .4225062   .4082719     1.03   0.301     -.377692    1.222704  beta22
        priv |  -.7788807   .4705994    -1.66   0.098    -1.701239    .1434771  beta32
       _cons |  -.7640601    .451101    -1.69   0.090    -1.648202    .1200817  beta02
------------------------------------------------------------------------------

display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 713.99396
```

Because the first submodel is predicting a **lower** outcome, we would expect **negative** slopes, whereas the second submodel is predicting a **higher** outcome, for which we would expect **positive** slopes. So to test the difference in the absolute value of the slope for the same predictor across submodels, we need to reverse the sign in one submodel. Thus, in the statements below, it looks like the slopes are being added instead of subtracted (as they actually are).

```
// Univ Wald tests of submodel slope diffs after reversing sign of [0]
lincom [0]c.gpa3*1 + [2]c.gpa3*1  // gpa3 slope diff
------------------------------------------------------------------------------
      apply3 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |    .026538   .6466994     0.04   0.967    -1.240969    1.294046  beta12 - beta10*-1
------------------------------------------------------------------------------

lincom [0]c.grad*1 + [2]c.grad*1  // grad slope diff
------------------------------------------------------------------------------
      apply3 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.5291406    .596828    -0.89   0.375    -1.698902    .6406208  beta22 - beta20*-1
------------------------------------------------------------------------------

lincom [0]c.priv*1 + [2]c.priv*1  // priv slope diff
------------------------------------------------------------------------------
      apply3 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -1.197699   .6942388    -1.73   0.084    -2.558382    .1629839  beta32 - beta30*-1
------------------------------------------------------------------------------
```

There is disagreement in what to call the EXP(logit slope) terms: SAS says they are still "**odds ratios**" whereas STATA insists they are "**relative risk**" (**rrr** below) ratios. The values provided by each are the same, though, so….?

```
display "Get Odds (Relative Risk) Ratios Instead of Logit Fixed Effects"
mlogit apply3 c.gpa3 c.grad c.priv, baseoutcome(1) rrr

------------------------------------------------------------------------------
      apply3 |        RRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
0            |
        gpa3 |   .6384252   .1852747    -1.55   0.122     .3614818    1.127544   exp(beta10)
        grad |   .3861047   .1224193    -3.00   0.003     .2074059    .7187686   exp(beta20)
        priv |   .6578236   .2258271    -1.22   0.222     .3356578    1.289205   exp(beta30)
       _cons |   2.589659   .8437749     2.92   0.003     1.367414    4.904391   exp(beta00)
-------------+----------------------------------------------------------------
1            |  (base outcome)
-------------+----------------------------------------------------------------
2            |
        gpa3 |   1.608479   .7835619     0.98   0.329     .6190945    4.179012   exp(beta12)
        grad |   1.525781   .6229334     1.03   0.301     .6854416    3.396361   exp(beta22)
        priv |   .4589194   .2159672    -1.66   0.098     .1824574     1.15428   exp(beta32)
       _cons |   .4657715     .21011    -1.69   0.090     .1923955    1.127589   exp(beta02)
------------------------------------------------------------------------------

// Odds ratios for submodel slope diffs after reversing sign of [0]
lincom [0]c.gpa3*1 + [2]c.gpa3*1, or  // gpa3 slope diff
------------------------------------------------------------------------------
      apply3 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         (1) |   1.026893   .6640913     0.04   0.967     .2891038    3.647513
------------------------------------------------------------------------------

lincom [0]c.grad*1 + [2]c.grad*1, or  // grad slope diff
------------------------------------------------------------------------------
      apply3 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         (1) |   .5891111    .351598    -0.89   0.375     .1828842    1.897659
------------------------------------------------------------------------------

lincom [0]c.priv*1 + [2]c.priv*1, or  // priv slope diff
------------------------------------------------------------------------------
      apply3 | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         (1) |    .301888   .2095824    -1.73   0.084     .0774299    1.177018
------------------------------------------------------------------------------
```

## R Syntax and Partial Output:

```
print("R Main-Effects Nominal Model -- ref is SECOND category of y=1")
Model3NomMain = vglm(data=Example2, family=multinomial(refLevel=2),reverse=TRUE,
                     formula=apply3~1+gpa3+grad+priv); summary(Model3NomMain)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1  0.95153    0.32582  2.9204 0.003496  beta00
(Intercept):2 -0.76406    0.45110 -1.6938 0.090308  beta02
gpa3:1        -0.44875    0.29021 -1.5463 0.122028  beta10
gpa3:2         0.47529    0.48714  0.9757 0.329229  beta12
grad:1        -0.95165    0.31706 -3.0014 0.002687  beta20
grad:2         0.42251    0.40827  1.0349 0.300731  beta22
priv:1        -0.41882    0.34329 -1.2200 0.222466  beta30
priv:2        -0.77888    0.47060 -1.6551 0.097907  beta32

Residual deviance: 713.99396 on 792 degrees of freedom  → model -2LL
Log-likelihood: -356.99698 on 792 degrees of freedom    → model   LL
Reference group is level  2  of the response → so y=1 is reference (in refLevel=2)
```

```
print("Likelihood Ratio Test of Predictors")
print("Analogous to F-test for model R2 in general LM")
anova(Model3Empty, Model3NomMain, type=1) # Nested "fewer" model goes first

Analysis of Deviance Table
Model 1: apply3 ~ 1
Model 2: apply3 ~ 1 + gpa3 + grad + priv

  Resid. Df Resid. Dev Df Deviance   Pr(>Chi)
1       798    741.205
2       792    713.994  6  27.2113 0.00013218

print("Univ Wald tests of submodel slope differences after reversing sign of 0-model slopes")
glhtModel3NomMain = glht(model= Model3NomMain, linfct=rbind(
  "gpa3 slope diff" = c(0,0, 1,1, 0,0, 0,0),  # in order of fixed effects
  "grad slope diff" = c(0,0, 0,0, 1,1, 0,0),
  "priv slope diff" = c(0,0, 0,0, 0,0, 1,1)))
glhtSummaryOR(glhtModel3NomMain) # print summary adding -2LL and odds ratios

                   Estimate    Std.Err      z.value p.value Odds.Ratio
gpa3 slope diff  0.02653804 0.64669727  0.041036265  0.9673 1.02689331  beta12 – beta10*-1
grad slope diff -0.52914057 0.59682741 -0.886588927  0.3753 0.58911105  beta22 – beta20*-1
priv slope diff -1.19769917 0.69423837 -1.725198744  0.0845 0.30188801  beta32 – beta30*-1

print("Odds ratios for model slopes")
exp(coefficients(Model3NomMain))

(Intercept):1 (Intercept):2  gpa3:1    gpa3:2       grad:1     grad:2      priv:1     priv:2
2.58965924  0.46577148  0.63842521  1.60847863  0.38610466  1.52578072  0.65782362  0.45891938
```

## STATA Code for Getting Predicted Outcomes for Fake People via Usual MARGINS:

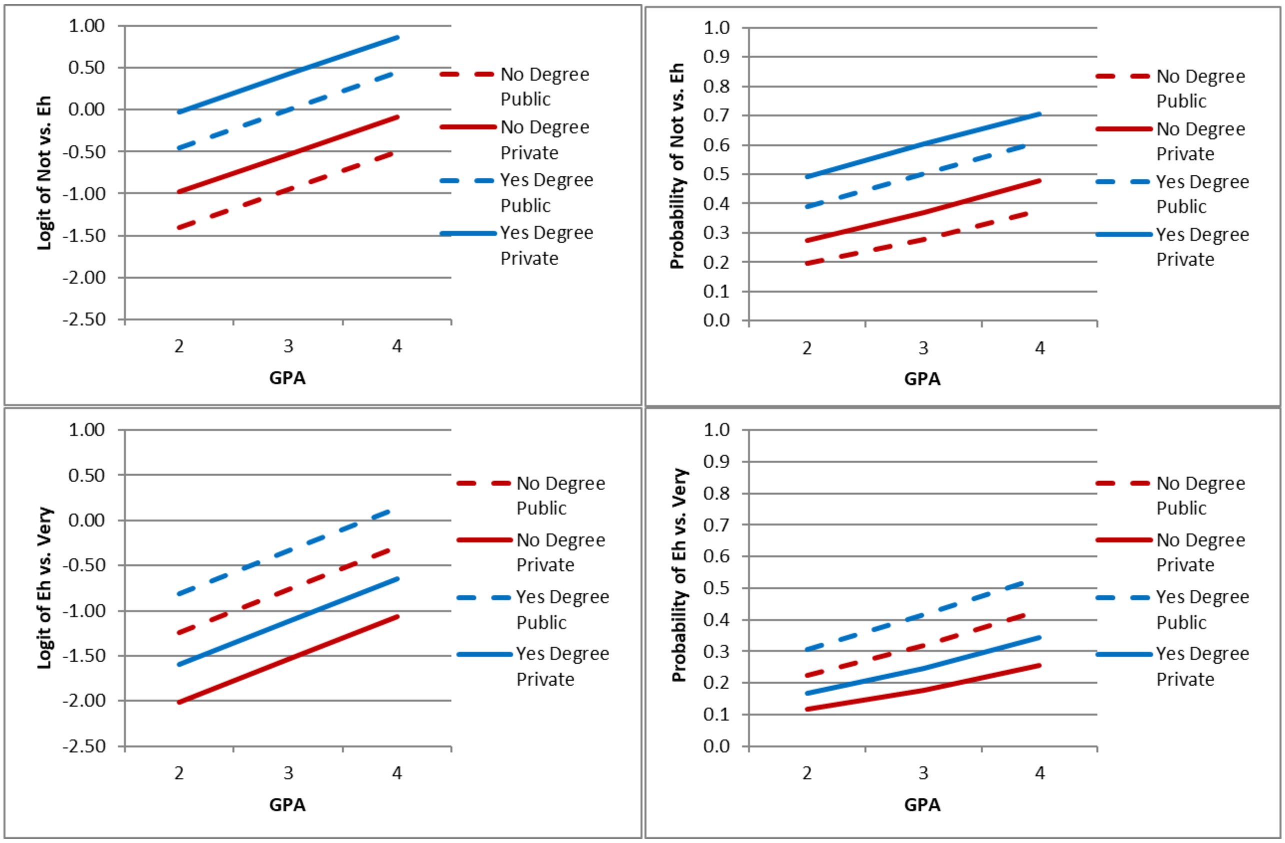
```
margins, at(c.gpa3=(-1(1)1) c.grad=(0(1)1) c.priv=(0(1)1)) predict(xb) // Yhat logits for 1 vs 0
margins, at(c.gpa3=(-1(1)1) c.grad=(0(1)1) c.priv=(0(1)1))             // All 3 probabilities
```

## R Code for Getting Predicted Outcomes for Fake People via PREDICT (so that I can get predicted probabilities for all outcome categories):

```
print("Get Yhat for specific values of predictors in fake people")
print("Y column = predicted yhat, Yprob = predicted probability")
PredNom = data.frame(FP, Y=predict(object=Model3NomMain, newdata=FP, type="link"),
                     Yprob=predict(object=Model3NomMain, newdata=FP, type="response"))

print("Rename columns into something meaningful")
names(PredNom)[names(PredNom)=='Y.log.mu..1..mu..2..']='Ylogit1vs0'
names(PredNom)[names(PredNom)=='Y.log.mu..3..mu..2..']='Ylogit1vs2'; PredNom

   gpa3 grad priv     Ylogit1vs0  Ylogit1vs2    Yprob.0    Yprob.1     Yprob.2
1    -1    0    0  1.40027704027 -1.23934893 0.75877334 0.18705937 0.054167285
2     0    0    0  0.95152629782 -0.76406014 0.63856577 0.24658293 0.114851298
3     1    0    0  0.50277555536 -0.28877136 0.48591035 0.29390265 0.220187007

4    -1    0    1  0.98145859580 -2.01822965 0.70196785 0.26307233 0.034959819
5     0    0    1  0.53270785335 -1.54294087 0.58394560 0.34278382 0.073270576
6     1    0    1  0.08395711089 -1.06765208 0.44730754 0.41128618 0.141406282

7    -1    1    0  0.44863024244 -0.81684270 0.52066846 0.33244793 0.146883615
8     0    1    0 -0.00012050002 -0.34155392 0.36888509 0.36892954 0.262185368
9     1    1    0 -0.44887124247  0.13373486 0.22950297 0.35952626 0.410970767

10   -1    1    1  0.02981179797 -1.59572343 0.46137496 0.44782354 0.090801504
11    0    1    1 -0.41893894449 -1.12043464 0.33154403 0.50406215 0.164393825
12    1    1    1 -0.86768968694 -0.64514586 0.21595225 0.51426928 0.269778473
```

See the excel file online for the plots!

**Note that in the plots, I reversed the signs of the (0 instead of 1) submodel so both submodels would be predicting the higher category! This will be much more intuitive for your readers.**




**Sample results section (should also report what software and version you used):**

We examined the extent to which a three-category decision for how likely a student was to apply to graduate school (55% 0=No, 35% 1=Eh, 10% 2=Very) could be predicted by a student's undergraduate GPA (M = 3.00, SD = 0.40, range = 1.90 to 4.00), whether at least one of their parents has a graduate degree (15.75% 0=No, 84.25% 1=Yes), and whether they attended a private university (14.25% 0=No, 85.75% 1=Yes). Specifically, we estimated two alternative sets of generalized linear models with conditional multinomial distributions using maximum likelihood. The GPA predictor was centered such that 0 indicated a GPA = 3. Effect sizes are provided using odds ratios (OR), in which OR values between 0 and 1 indicate negative effects, 1 indicates no effect, and values above 1 indicate positive effects. Nested model comparisons were conducted using likelihood ratio tests (i.e., the difference in −2LL between nested models with degrees of freedom equal to the number of new parameters).

First, we treated the three-category outcome as ordinal using a cumulative logit link function—this parameterization requires two submodels that predict the logit of $y_i > 0$ and $y_i > 1$. By default, separate intercepts are estimated for each submodel, but all model slopes are constrained equal across submodels (i.e., proportional odds). This first ordinal model examined the main effects of the three predictors, which together resulted in a significant prediction of the logit of the probability of each level of decision to apply to graduate school, $-2\Delta LL(3) = 23.61$, $p < .0001$. GPA had a significantly positive effect, such that for every unit greater GPA, the logit of the higher response was greater by 0.616 (SE = 0.261; OR = 1.851). Likewise, the logit of the higher response was significantly greater for students for whom at least one parent had a graduate degree by 1.048 (SE = 0.266, OR = 2.851). However, the logit of the higher response was nonsignificantly greater for students who attended a private university by 0.059 (SE = 0.298, OR = 1.060). We then tested the proportional odds assumption by specifying an alternative model in which separate slopes were estimated for the two submodels. Only the slope for parent graduate degree differed across models—although neither slope was significant, the slope was significantly more negative in predicting $y_i > 1$ than $y_i > 0$.

Second, we treated the outcome as nominal using a generalized logit link function—this approach requires choosing a reference category (1=Eh). The submodels then predict the logit of choosing each other possible response (i.e., $y_i = 0$ given $y_i = 0$ or 1; $y_i = 2$ given $y_i = 2$ or 1). All parameters are estimated separately across submodels, and only one slope was significant. First, the logit of choosing 0=No instead of 1=Eh was significantly smaller for students for whom at least one parent had a graduate degree by 0.952 (SE = 0.317, OR = 0.386). In addition, none of the slopes differed significantly across submodels.