# 432 Class 21 Slides

github.com/THOMASELOVE/432-2018

2018-04-03

# Setup

```r
library(skimr)
library(rms)
library(MASS)
library(nnet)
library(tidyverse)
```

# Today's Materials

**Regression Models for Ordered Multi-Categorical Outcomes**

- Proportional Odds Logistic Regression Models
- Using `polr`
- Using `lrm`
- Understanding and Interpreting the Model
- Testing the Proportional Odds Assumption
- Picturing the Model Fit

# Applying to Graduate School

# These are simulated data

This is a simulated data set of 530 students.

A study looks at factors that influence the decision of whether to apply to graduate school.

College juniors are asked if they are unlikely, somewhat likely, or very likely to apply to graduate school. Hence, our outcome variable has three categories. Data on parental educational status, whether the undergraduate institution is public or private, and current GPA is also collected. The researchers have reason to believe that the “distances” between these three points are not equal. For example, the “distance” between “unlikely” and “somewhat likely” may be shorter than the distance between “somewhat likely” and “very likely”.

```
gradschool <- read.csv("gradschool_new.csv") %>% tbl_df
```

# The `gradschool` data and my Source

The **gradschool** example is adapted from this UCLA site.

- There, they look at 400 students.
- I simulated a new data set containing 530 students.

| Variable | Description |
|---|---|
| `student` | subject identifying code (A001 - A530) |
| `apply` | 3-level ordered outcome: "unlikely", "somewhat likely" and "very likely" to apply |
| `pared` | 1 = at least one parent has a graduate degree, else 0 |
| `public` | 1 = undergraduate institution is public, else 0 |
| `gpa` | student's undergraduate grade point average (max 4.00) |

# Cleanup

```r
gradschool <- gradschool %>%
    mutate(apply = fct_relevel(apply, "unlikely",
                     "somewhat likely", "very likely"),
           apply = factor(apply, ordered = TRUE))

is.ordered(gradschool$apply)
```

```
[1] TRUE
```

# gradschool %>% select(-student) %>% skim

```
> gradschool %>% select(-student) %>% skim
Skim summary statistics
 n obs: 530
 n variables: 4

Variable type: factor
 variable missing complete   n n_unique                        top_counts ordered
    apply       0      530 530        3 unl: 303, som: 172, ver: 55, NA: 0    TRUE

Variable type: integer
 variable missing complete   n mean   sd p0 p25 median p75 p100     hist
    pared       0      530 530 0.19 0.4   0   0      0   0    1
   public       0      530 530 0.25 0.43  0   0      0   0    1

Variable type: numeric
 variable missing complete   n mean   sd  p0  p25 median  p75 p100     hist
      gpa       0      530 530 3.01 0.52 1.9 2.61   3.08 3.44    4
```

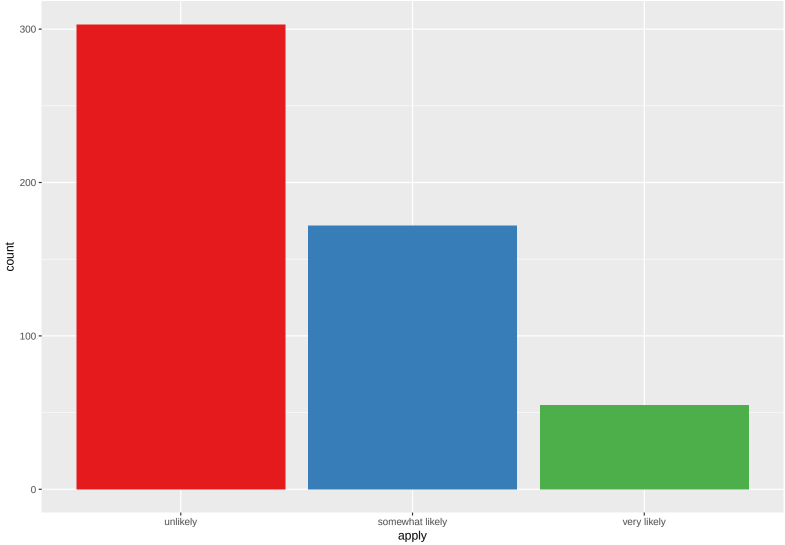
# Displaying Categorical Data

# Data (besides gpa) as Cross-Tabulation

```r
ftable(xtabs(~ public + apply + pared, data = gradschool))
```

```
                       pared   0   1
public apply                        
0      unlikely              206  17
       somewhat likely       111  32
       very likely            22  12
1      unlikely               62  18
       somewhat likely        15  14
       very likely            11  10
```

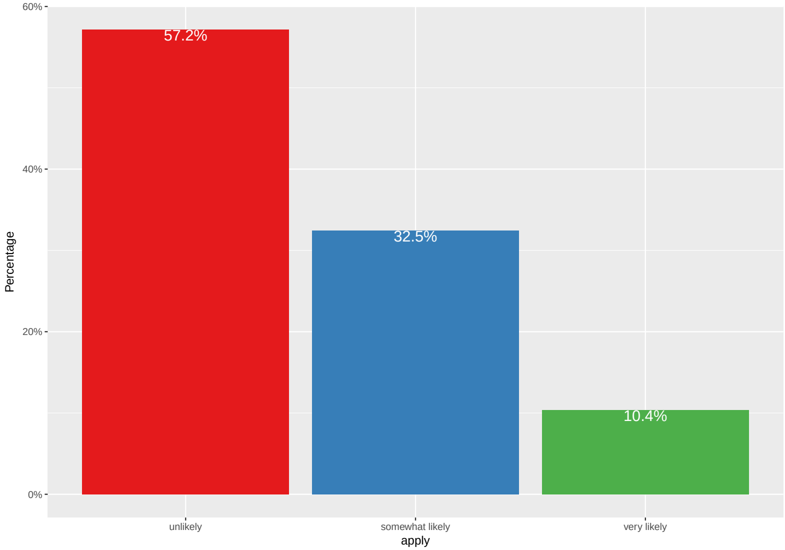
# Bar Chart of `apply` classifications



# Bar Chart of apply classifications (code)

```
ggplot(gradschool, aes(x = apply, fill = apply)) +
    geom_bar() +
    scale_fill_brewer(palette = "Set1") +
    guides(fill = FALSE)
```

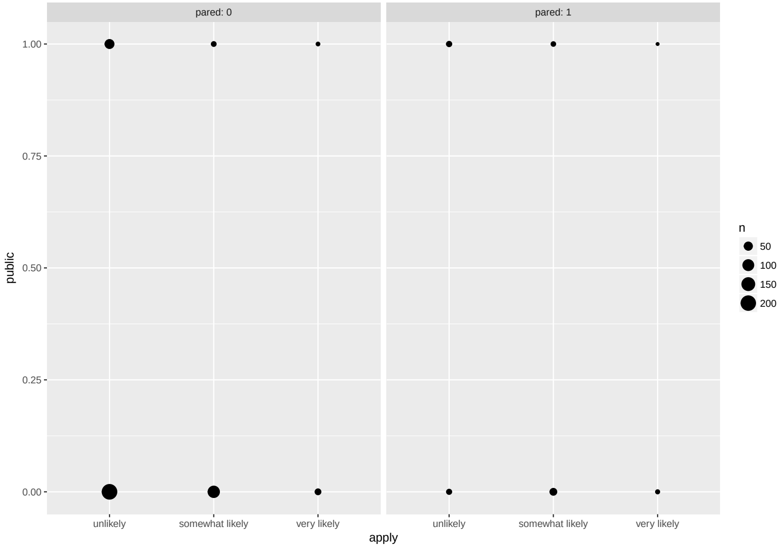
# Maybe you'd prefer to show the percentages?



# Maybe you'd prefer to show the percentages? (code)

```
ggplot(gradschool, aes(x = apply, fill = apply)) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    geom_text(aes(y = (..count..)/sum(..count..),
                  label = scales::percent((..count..) /
                                      sum(..count..))),
              stat = "count", vjust = 1,
              color = "white", size = 5) +
    scale_y_continuous(labels = scales::percent) +
    scale_fill_brewer(palette = "Set1") +
    guides(fill = FALSE) +
    labs(y = "Percentage")
```

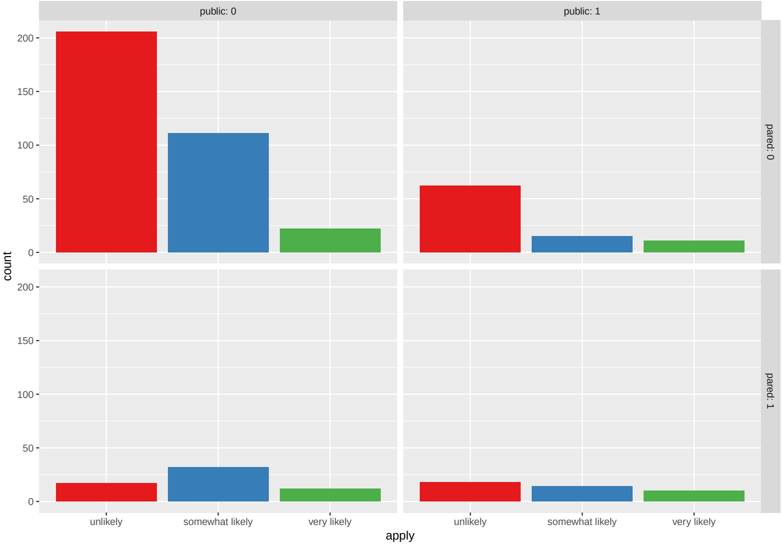
# Facetted Counts Chart for a Three-Way Cross-Tabulation



# Facetted Counts Chart for a 3-Way Cross-Tabulation (code)

```r
ggplot(gradschool, aes(x = apply, y = public)) +
    geom_count() +
    facet_wrap(~ pared, labeller = "label_both")
```

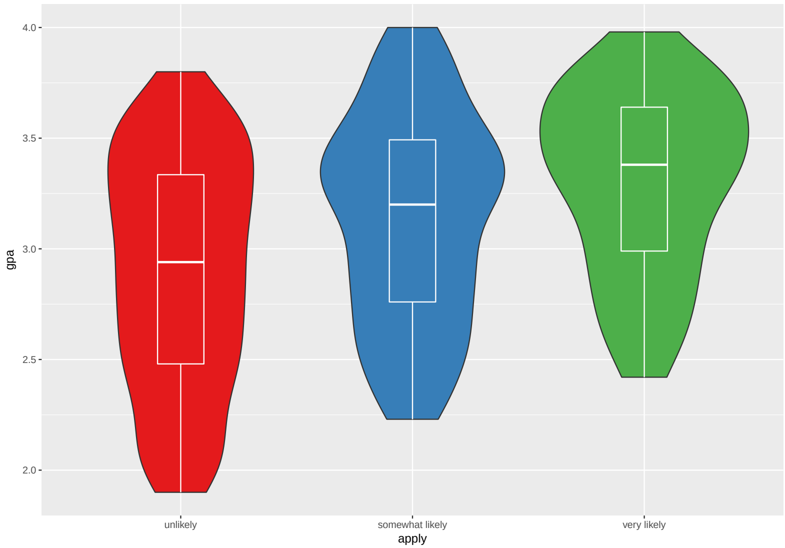
# Breakdown of `apply` percentages by `public`, `pared`



# Breakdown of `apply` percentages by `public`, `pared` (code)

```
ggplot(gradschool, aes(x = apply, fill = apply)) +
    geom_bar() +
    scale_fill_brewer(palette = "Set1") +
    guides(fill = FALSE) +
    facet_grid(pared ~ public, labeller = "label_both")
```

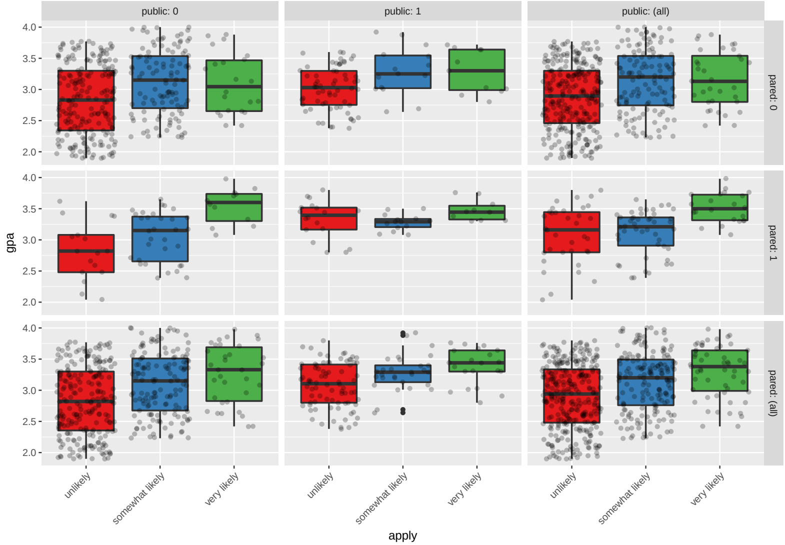
# Breakdown of `gpa` by `apply`



# Breakdown of gpa by apply (code)

```
ggplot(gradschool, aes(x = apply, y = gpa, fill = apply)) +
    geom_violin(trim = TRUE) +
    geom_boxplot(col = "white", width = 0.2) +
    scale_fill_brewer(palette = "Set1") +
    guides(fill = FALSE)
```

# Breakdown of `gpa` by all 3 other variables

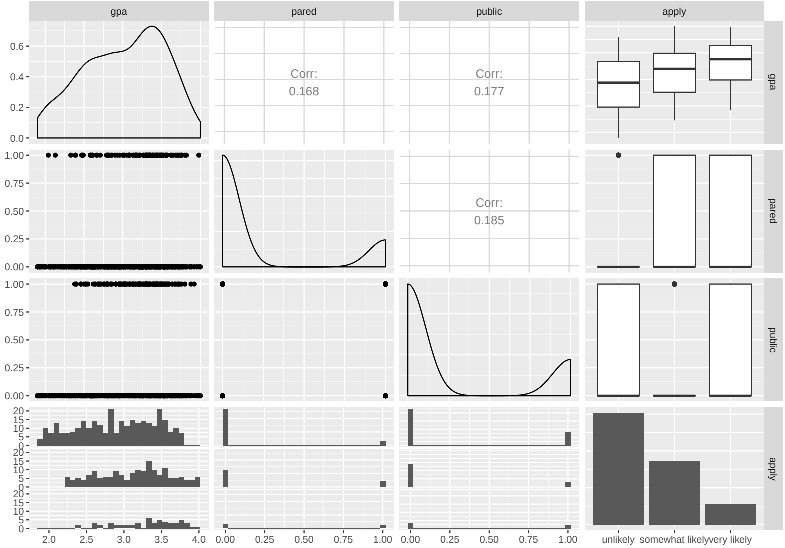


# Breakdown of gpa by all 3 other variables (code)

```
ggplot(gradschool, aes(x = apply, y = gpa)) +
    geom_boxplot(aes(fill = apply), size = .75) +
    geom_jitter(alpha = .25) +
    facet_grid(pared ~ public, margins = TRUE,
               labeller = "label_both") +
    scale_fill_brewer(palette = "Set1") +
    guides(fill = FALSE) +
    theme(axis.text.x =
            element_text(angle = 45, hjust = 1, vjust = 1))
```

# Proportional Odds Logit Model via `polr`

# Scatterplot Matrix (run with `message = F`)



# Scatterplot Matrix (code, run with `message = F`)

```
GGally::ggpairs(gradschool %>%
                    select(gpa, pared, public, apply))
```

# Fitting the Model

We use the polr function from the MASS package:

```
m <- polr(apply ~ pared + public + gpa,
          data = gradschool, Hess=TRUE)
```

The polr name comes from proportional odds logistic regression, highlighting a key assumption of this model.

polr uses the standard formula interface in R for specifying a regression model with outcome followed by predictors. We also specify Hess=TRUE to have the model return the observed information matrix from optimization (called the Hessian) which is used to get standard errors.

# Obtaining Predicted Probabilities from `m`

To start we'll obtain predicted probabilities, which are usually the best way to understand the model.

For example, we can vary gpa for each level of `pared` and `public` and calculate the model's estimated probability of being in each category of `apply`.

First, create a new dataset of values to use for prediction.

```
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))
```

# Obtaining Predicted Probabilities from `m`

Now, make predictions using model `m`

```
newdat1 <- cbind(newdat, predict(m, newdat, type = "probs"))
head(newdat1, 5)
```

```
  pared public      gpa  unlikely somewhat likely
1     0      0 1.900000 0.8460125       0.1315031
2     1      0 1.921212 0.6287747       0.3017965
3     0      0 1.942424 0.8395968       0.1368294
4     1      0 1.963636 0.6174011       0.3099749
5     0      0 1.984848 0.8329664       0.1423188
  very likely
1  0.02248434
2  0.06942884
3  0.02357380
4  0.07262398
5  0.02471472
```

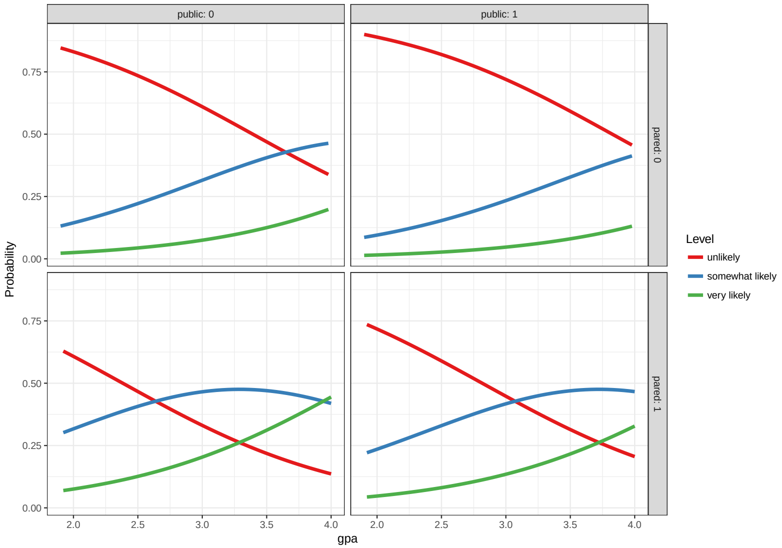
# Reshape data

Now, we reshape the data with gather

```
newdat.long <- gather(newdat1, "Level", "Probability", 4:6)
newdat.long <- newdat.long %>%
    mutate(Level = fct_relevel(Level, "unlikely",
                                       "somewhat likely"))
head(newdat.long)
```

```
  pared public      gpa    Level Probability
1     0      0 1.900000 unlikely   0.8460125
2     1      0 1.921212 unlikely   0.6287747
3     0      0 1.942424 unlikely   0.8395968
4     1      0 1.963636 unlikely   0.6174011
5     0      0 1.984848 unlikely   0.8329664
6     1      0 2.006061 unlikely   0.6058974
```

# Plot the prediction results. . .



# Plot the prediction results... (code)

```
ggplot(newdat.long, aes(x = gpa, y = Probability,
                        color = Level)) +
    geom_line(size = 1.5) +
    scale_color_brewer(palette = "Set1") +
    theme_bw() +
    facet_grid(pared ~ public, labeller="label_both")
```

# Cross-Tabulation of Predicted/Observed Classifications

Predictions in the rows, Observed in the columns

```r
addmargins(table(predict(m), gradschool$apply))
```

```
                  unlikely somewhat likely very likely Sum
  unlikely             264             112          29 405
  somewhat likely       39              60          25 124
  very likely            0               0           1   1
  Sum                  303             172          55 530
```

We only predict one subject to be in the "very likely" group by modal prediction.

# Describing the Proportional Odds Logistic Model

Our outcome, `apply`, has three levels. Our model has two logit equations:

- one estimating the log odds that `apply` will be less than or equal to 1 (`apply` = unlikely)
- one estimating the log odds that `apply` $\leq 2$ (`apply` = unlikely or somewhat likely)

That's all we need to estimate the three categories, since Pr(`apply` $\leq 3) = 1$, because very likely is the maximum category for `apply`.

- The parameters to be fit include two intercepts:
  - $\zeta_1$ will be the `unlikely|somewhat likely` parameter
  - $\zeta_2$ will be the `somewhat likely|very likely` parameter
- We'll have a total of five free parameters when we add in the slopes ($\beta$) for `pared`, `public` and `gpa`.

The two logistic equations that will be fit differ only by their intercepts.

# summary(m)

```
Call:
polr(formula = apply ~ pared + public + gpa, data = gradschool
    Hess = TRUE)

Coefficients:
         Value Std. Error t value
pared   1.1525     0.2184   5.276
public -0.4949     0.2195  -2.254
gpa     1.1416     0.1850   6.171

Intercepts:
                            Value   Std. Error t value
unlikely|somewhat likely     3.8727  0.5721     6.7692
somewhat likely|very likely  5.9413  0.6063     9.7993

Residual Deviance: 900.9629
AIC: 910.9629
```

# Understanding the Model

$$logit[Pr(apply \leq 1)] = \zeta_1 - \beta_1 pared - \beta_2 public - \beta_3 gpa$$

$$logit[Pr(apply \leq 2)] = \zeta_2 - \beta_1 pared - \beta_2 public - \beta_3 gpa$$

So we have:

$$logit[Pr(apply \leq unlikely)] = 3.87 - 1.15 pared - (-0.49) public - 1.14 gpa$$

and

$$logit[Pr(apply \leq somewhat)] = 5.94 - 1.15 pared - (-0.49) public - 1.14 gpa$$

# `confint(m)`

Confidence intervals for the slope coefficients on the log odds scale can be estimated in the usual way.

```
Waiting for profiling to be done...
```

```
             2.5 %      97.5 %
pared   0.7257019  1.58305735
public -0.9320573 -0.07029727
gpa     0.7837559  1.50974002
```

These CIs describe results in units of ordered log odds.

- For example, for a one unit increase in `gpa`, we expect a 1.14 increase in the expected value of `apply` (95% CI 0.78, 1.51) in the log odds scale, holding `pared` and `public` constant.
- This would be more straightforward if we exponentiated.

# Exponentiating the Coefficients

```r
exp(coef(m))
```

```
    pared    public       gpa 
3.1660446 0.6096623 3.1318247 
```

```r
exp(confint(m))
```

```
Waiting for profiling to be done...
```

```
           2.5 %    97.5 %
pared  2.0661808 4.8698218
public 0.3937428 0.9321167
gpa    2.1896811 4.5255541
```

# Interpreting the Coefficients

| Variable | Estimate | 95% CI |
|---|---|---|
| `gpa` | 3.13 | (2.19, 4.53) |
| `public` | 0.61 | (0.39, 0.93) |
| `pared` | 3.17 | (2.07, 4.87) |

- When a student's gpa increases by 1 unit, the odds of moving from "unlikely" applying to "somewhat likely" or "very likely" applying are multiplied by 3.13 (95% CI 2.19, 4.52).
- For `public`, the odds of moving from a lower to higher status are multiplied by 0.61 (95% CI 0.39, 0.93) as we move from private to public.
- How about `pared`?

# Comparison to a Null Model

```r
m0 <- polr(apply ~ 1, data = gradschool)

anova(m, m0)
```

```
Likelihood ratio tests of ordinal regression models

Response: apply
                 Model Resid. df Resid. Dev   Test    Df
1                    1       528   975.1828
2 pared + public + gpa       525   900.9629 1 vs 2     3
  LR stat.      Pr(Chi)
1
2 74.21989 5.551115e-16
```

# AIC and BIC are available, too

We could also compare model `m1` to the null model `m0` with AIC or BIC.

```
AIC(m, m0)
```

```
   df      AIC
m   5 910.9629
m0  2 979.1828
```

```
BIC(m, m0)
```

```
   df      BIC
m   5 932.3273
m0  2 987.7286
```

# Testing the Proportional Odds Assumption

One way to test the proportional odds assumption is to compare the fit of the proportional odds logistic regression to a model that does not make that assumption. A natural candidate is a **multinomial logit** model, which is typically used to model unordered multi-categorical outcomes, and fits a slope to each level of the `apply` outcome in this case, as opposed to the proportional odds logit, which fits only one slope across all levels.

Since the proportional odds logistic regression model is nested in the multinomial logit, we can perform a likelihood ratio test. To do this, we first fit the multinomial logit model, with the `multinom` function from the `nnet` package.

# Fitting the multinomial model

```
m1_multi <- multinom(apply ~ pared + public + gpa,
                     data = gradschool)
```

```
# weights:  15 (8 variable)
initial  value 582.264513
iter  10 value 446.199617
final  value 445.443366
converged
```

# The multinomial model

```
m1_multi
```

```
Call:
multinom(formula = apply ~ pared + public + gpa, data = gradsc

Coefficients:
                (Intercept)   pared      public       gpa
somewhat likely   -3.527249 1.072451 -0.97765580 0.9857488
very likely       -7.311227 1.400955 -0.02934361 1.6937996

Residual Deviance: 890.8867
AIC: 906.8867
```

# Comparing the Models

The multinomial logit fits two intercepts and six slopes, for a total of 8 estimated parameters.

The proportional odds logit, as we've seen, fits two intercepts and three slopes, for a total of 5. The difference is 3, and we use that number in the sequence below to build our test of the proportional odds assumption.

# Testing the Proportional Odds Assumption

```
LL_1 <- logLik(m)
LL_1m <- logLik(m1_multi)
(G <- -2 * (LL_1[1] - LL_1m[1]))
```

```
[1] 10.07618
```

```
pchisq(G, 3, lower.tail = FALSE)
```

```
[1] 0.01792959
```

The $p$ value is 0.018, so it indicates that the proportional odds model fits less well than the more complex multinomial logit.

# What to do in light of this test...

- A non-significant $p$ value here isn't always the best way to assess the proportional odds assumption, but it does provide some evidence of model adequacy.
- Given the significant result here, we have concerns about the proportional odds assumption.
  - One alternative would be to fit the multinomial model instead.
  - Another would be to fit a check of residuals (see Frank Harrell's RMS text.)
  - Another would be to fit a different model for ordinal regression. Several are available (check out `orm` in the `rms` package, for instance.)

# Fitting the Proportional Odds Logistic Regression with `lrm`

# Using lrm to work through this model

```
d <- datadist(gradschool)
options(datadist = "d")
mod <- lrm(apply ~ pared + public + gpa,
           data = gradschool, x = T, y = T)
```

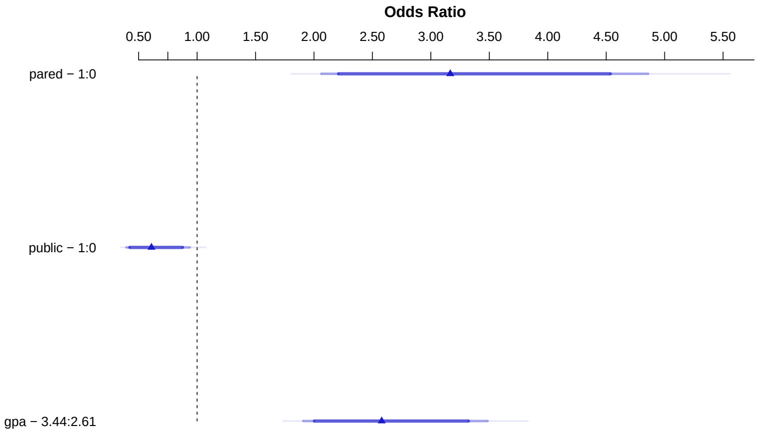
# mod output

```
> mod
Logistic Regression Model

 lrm(formula = apply ~ pared + public + gpa, data = gradschool,
     x = T, y = T)

                         Model Likelihood     Discrimination     Rank Discrim.
                            Ratio Test           Indexes           Indexes
 Obs              530    LR chi2      74.22   R2        0.155   C       0.684
  unlikely        303    d.f.             3   g         0.895   Dxy     0.369
  somewhat likely172     Pr(> chi2) <0.0001   gr        2.448   gamma   0.369
  very likely      55                         gp        0.200   tau-a   0.206
 max |deriv| 5e-09                            Brier     0.216

                    Coef    S.E.   Wald Z Pr(>|Z|)
 y>=somewhat likely -3.8728 0.5721 -6.77  <0.0001
 y>=very likely     -5.9413 0.6063 -9.80  <0.0001
 pared               1.1525 0.2184  5.28  <0.0001
 public             -0.4949 0.2195 -2.25  0.0242
 gpa                 1.1416 0.1850  6.17  <0.0001
```

## summary(mod)

```
             Effects              Response : apply

 Factor      Low  High Diff. Effect   S.E.    Lower 0.95
 pared       0.00 1.00 1.00   1.15250 0.21843  0.72436
  Odds Ratio 0.00 1.00 1.00   3.16600      NA  2.06340
 public      0.00 1.00 1.00  -0.49486 0.21951 -0.92509
  Odds Ratio 0.00 1.00 1.00   0.60966      NA  0.39650
 gpa         2.61 3.44 0.83   0.94756 0.15354  0.64662
  Odds Ratio 2.61 3.44 0.83   2.57940      NA  1.90910
 Upper 0.95
  1.580600
  4.857900
 -0.064629
  0.937410
  1.248500
  3.485100
```

# plot(summary(mod))



# Coefficients in our equation

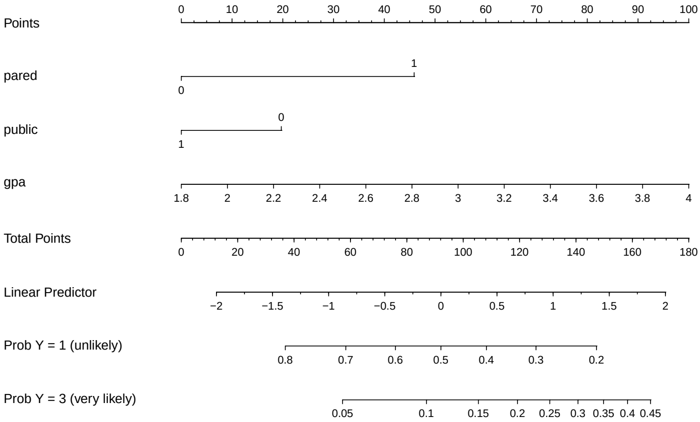
```
mod$coef
```

```
y>=somewhat likely     y>=very likely              pared
         -3.872786          -5.941317           1.152479
            public                gpa
         -0.494859           1.141633
```

# Nomogram of `mod` (code)

```
fun.1 <- function(x) 1 - plogis(x)
fun.3 <- function(x)
    plogis(x - mod$coef[1] + mod$coef[2])

plot(nomogram(mod,
    fun=list('Prob Y = 1 (unlikely)' = fun.1,
             'Prob Y = 3 (very likely)' = fun.3)))
```

# Nomogram of `mod` (result)



# set.seed(432); validate(mod)

```
          index.orig training    test optimism
Dxy           0.3687   0.3751  0.3631   0.0120
R2            0.1553   0.1633  0.1505   0.0128
Intercept     0.0000   0.0000 -0.0071   0.0071
Slope         1.0000   1.0000  0.9813   0.0187
Emax          0.0000   0.0000  0.0054   0.0054
D             0.1382   0.1466  0.1335   0.0131
U            -0.0038  -0.0038 -0.4635   0.4597
Q             0.1419   0.1504  0.5970  -0.4466
B             0.2155   0.2139  0.2173  -0.0034
g             0.8954   0.9185  0.8793   0.0392
gp            0.2004   0.2033  0.1971   0.0062
          index.corrected  n
Dxy                0.3567 40
R2                 0.1426 40
Intercept         -0.0071 40
Slope              0.9813 40
```

# Next Time

- Multinomial Models for nominal multi-categorical responses