432 Class 15 Slides

thomaselove.github.io/432

2022-03-03

Setup

```r
library(here); library(magrittr); library(janitor)
library(conflicted); library(skimr)
library(rms)
library(MASS)
library(nnet)
library(tidyverse)

theme_set(theme_bw())

conflict_prefer("select", "dplyr")
conflict_prefer("filter", "dplyr")
```

Today's Materials

Regression Models for Ordered Multi-Categorical Outcomes

- Applying to Graduate School: An Example
- Proportional Odds Logistic Regression Models
- Using `polr`
- Using `lrm`
- Understanding and Interpreting the Model
- Testing the Proportional Odds Assumption
- Picturing the Model Fit

Not Discussed in Detail: slides 54-end

- Asbestos: A Second POLR example

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(here("data" , "gradschool_new.csv")) %>%
  type.convert(as.is = FALSE)
```

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)

Ensuring that our outcome is an ordered factor

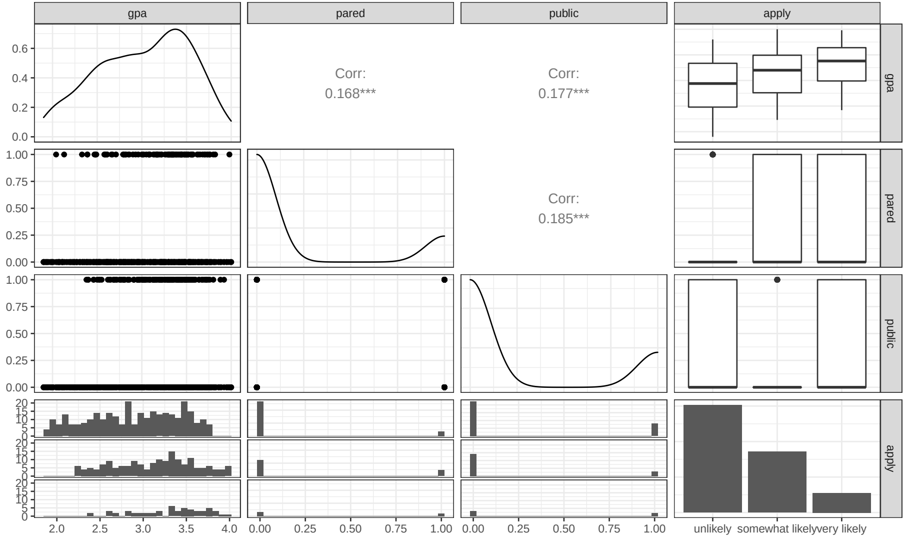
```
gradschool <- gradschool %>%
    mutate(apply = fct_relevel(apply, "unlikely",
                          "somewhat likely", "very likely"),
           apply = factor(apply, ordered = TRUE))

is.ordered(gradschool$apply)
```

```
[1] TRUE
```

Skim of the `gradschool` data

```
> gradschool %>% select(-student) %>% skim
-- Data Summary ------------------------
                           Values
Name                       Piped data
Number of rows             530
Number of columns          4
_______________________
Column type frequency:
  factor                   1
  numeric                  3
________________________
Group variables            None

-- Variable type: factor ------------------------------------------------------------------------------------------------------
# A tibble: 1 x 6
  skim_variable n_missing complete_rate ordered n_unique top_counts
* <chr>             <int>         <dbl> <lgl>      <int> <chr>
1 apply                 0             1 TRUE           3 unl: 303, som: 172, ver: 55

-- Variable type: numeric -----------------------------------------------------------------------------------------------------
# A tibble: 3 x 11
  skim_variable n_missing complete_rate  mean    sd    p0   p25   p50   p75  p100 hist
* <chr>             <int>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 pared                 0             1 0.194 0.396   0     0     0     0       1 ▇▁▁▁▂
2 public                0             1 0.245 0.431   0     0     0     0       1 ▇▁▁▁▂
3 gpa                   0             1 3.01  0.516   1.9   2.61  3.08  3.44    4 ▂▃▅▇▂
```

Scatterplot Matrix (run with `message = F`)



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

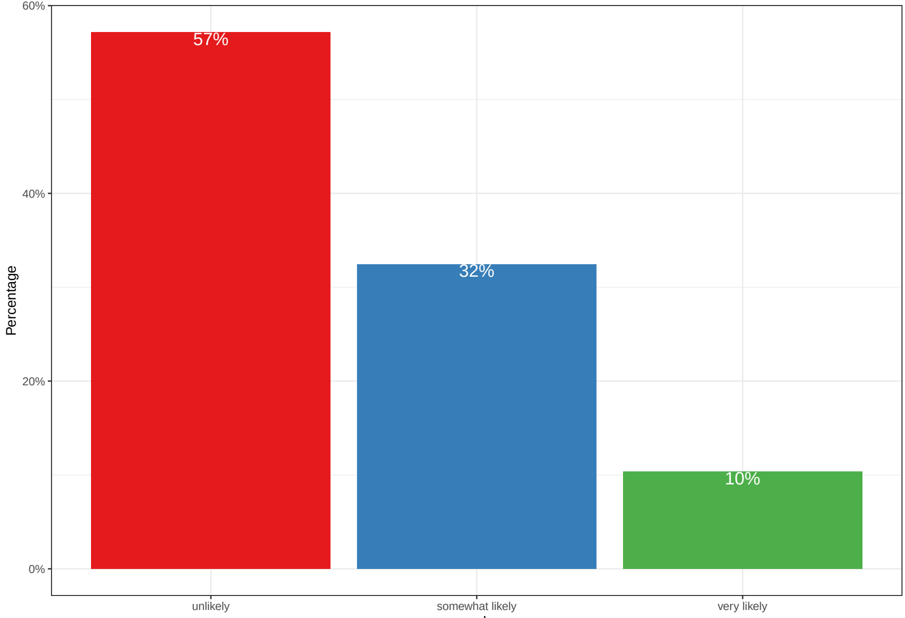
```
GGally::ggpairs(gradschool %>%
                    select(gpa, pared, public, apply))
```

Data (besides gpa) as Cross-Tabulation

```
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
```

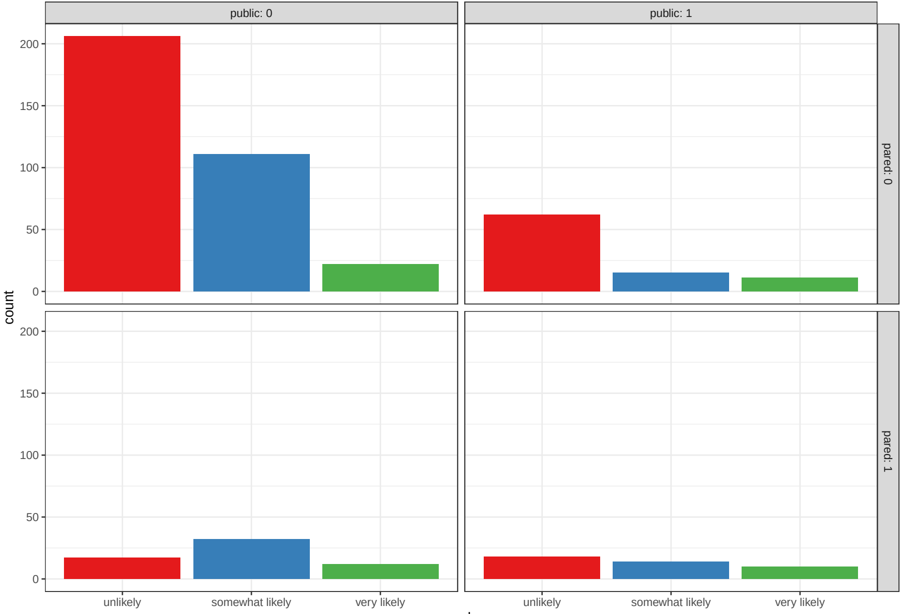
Bar Chart of `apply` classifications with %s



Showing the percentages in each bar (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 = "none") +
    labs(y = "Percentage")
```

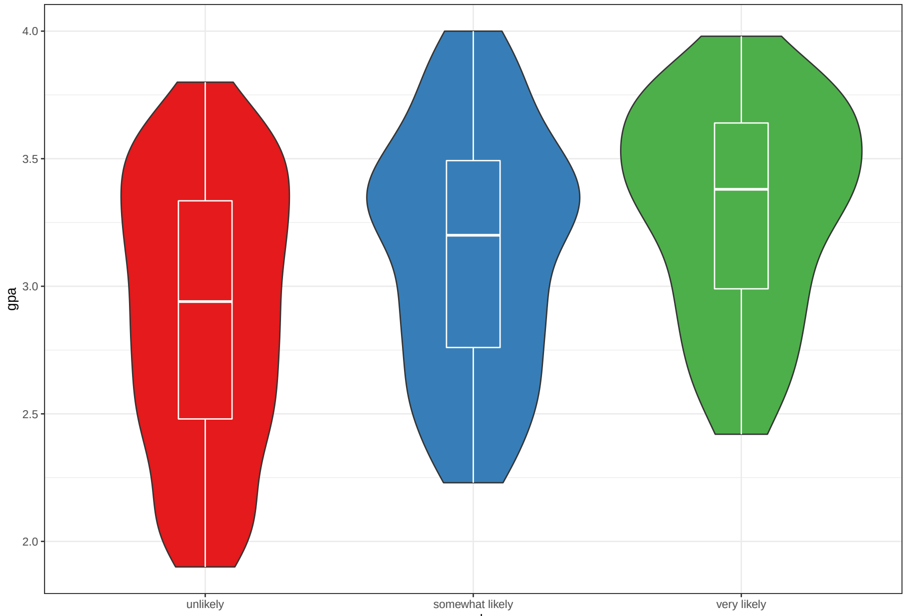
Breakdown of `apply` percentages by `public`, `pared`



Breakdown of apply percentages by public, pared (code)

```
ggplot(gradschool, aes(x = apply, fill = apply)) +
    geom_bar() +
    scale_fill_brewer(palette = "Set1") +
    guides(fill = "none") +
    facet_grid(pared ~ public, labeller = "label_both")
```

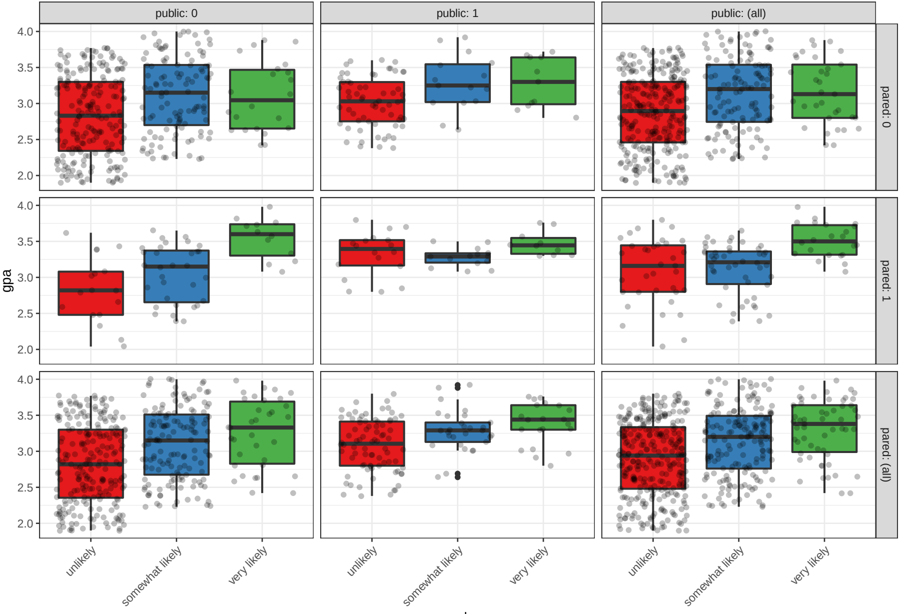
Breakdown of `gpa` by `apply`



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 = "none")
```

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 = "none") +
    theme(axis.text.x =
            element_text(angle = 45, hjust = 1, vjust = 1))
```

Proportional Odds Logit Model via `polr`

Fitting the POLR model with MASS::polr

We use the polr function from the MASS package:

```
mod_p1 <- 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 `mod_p1`

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 tibble of values to use for prediction.

```
newdat <- tibble(
  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 `mod_p1`

Now, make predictions using model `mod_p1`

```
newdat_p1 <- cbind(newdat,
                   predict(mod_p1, newdat, type = "probs"))
head(newdat_p1, 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
```

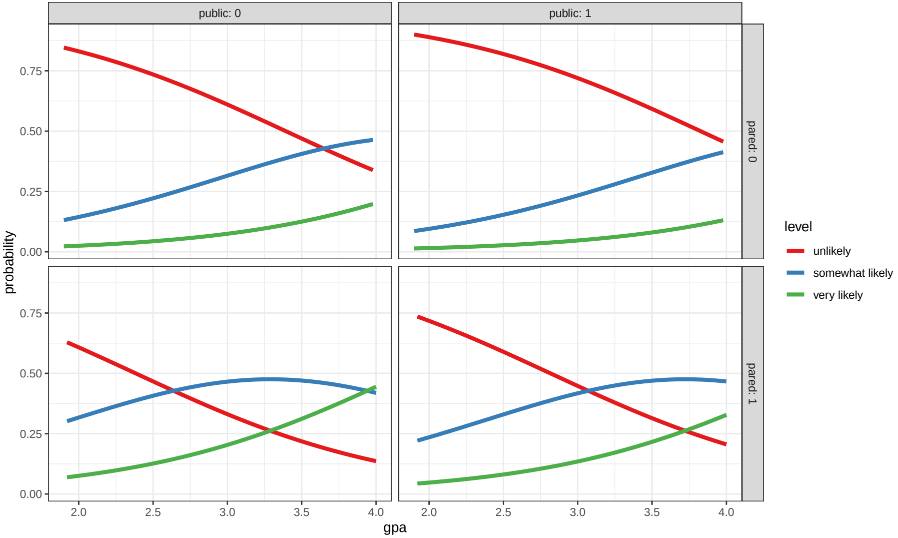
Reshape data

Now, we reshape the data with `pivot_longer`

```
newdat_long <-
  pivot_longer(newdat_p1,
               cols = c("unlikely":"very likely"),
               names_to = "level",
               values_to = "probability") %>%
  mutate(level = fct_relevel(level, "unlikely",
                             "somewhat likely"))

head(newdat_long, 3)
```

```
# A tibble: 3 x 5
  pared public   gpa level           probability
  <int>  <int> <dbl> <fct>                 <dbl>
1     0      0   1.9 unlikely             0.846
2     0      0   1.9 somewhat likely      0.132
3     0      0   1.9 very likely          0.0225
```

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

```
addmargins(table(predict(mod_p1), 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` ≤ 2 (`apply` = unlikely or somewhat likely)

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

- The parameters to be fit include two intercepts:
  - ζ_1 will be the `unlikely|somewhat likely` parameter
  - ζ_2 will be the `somewhat likely|very likely` parameter
- We'll have a total of five free parameters when we add in the slopes (β) for `pared`, `public` and `gpa`.

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

summary(mod_p1)

```
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(mod_p1)

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

```
exp(coef(mod_p1))
```

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

```
exp(confint(mod_p1))
```

```
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), all else held constant.
- For `public`, the odds of moving from a lower to higher `apply` status are multiplied by 0.61 (95% CI 0.39, 0.93) as we move from private to public, all else held constant.
- How about `pared`?

Comparison to a Null Model

```
mod_p0 <- polr(apply ~ 1, data = gradschool)

anova(mod_p1, mod_p0)
```

```
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 `mod_p1` to the null model `mod_p0` with AIC or BIC.

```
AIC(mod_p1, mod_p0)
```

```
       df      AIC
mod_p1  5 910.9629
mod_p0  2 979.1828
```

```
BIC(mod_p1, mod_p0)
```

```
       df      BIC
mod_p1  5 932.3273
mod_p0  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

```r
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

```r
LL_1 <- logLik(mod_p1)
LL_1m <- logLik(m1_multi)
(G <- -2 * (LL_1[1] - LL_1m[1]))
```

```
[1] 10.07618
```

```r
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.

Comparing AIC and BIC

```r
AIC(mod_p1)
```

```
[1] 910.9629
```

```r
AIC(m1_multi)
```

```
[1] 906.8867
```

```r
BIC(mod_p1)
```

```
[1] 932.3273
```

```r
BIC(m1_multi)
```

```
[1] 941.0697
```

What to do in light of these results...

- 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.
- The stronger BIC (and only slightly worse AIC) for our POLR model relative to the multinomial gives us some conflicting advice.
  - 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.)

Using `lrm` for Proportional Odds Logistic Regression

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)
```

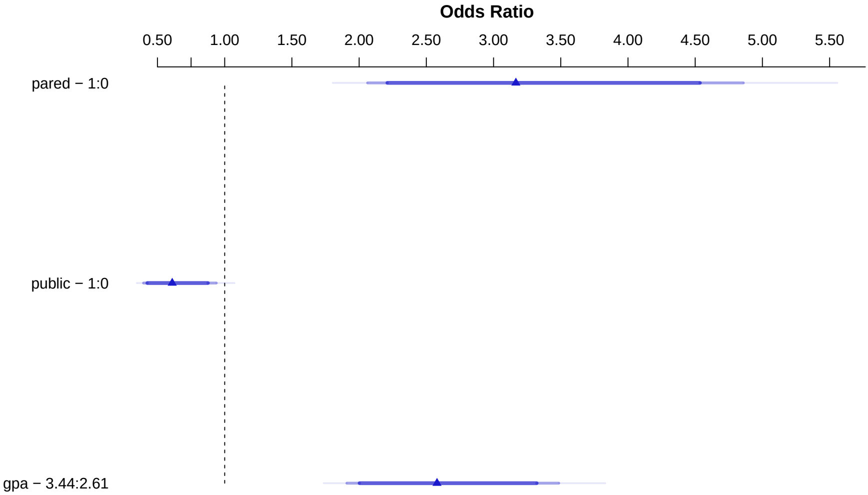
mod output

```
> 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

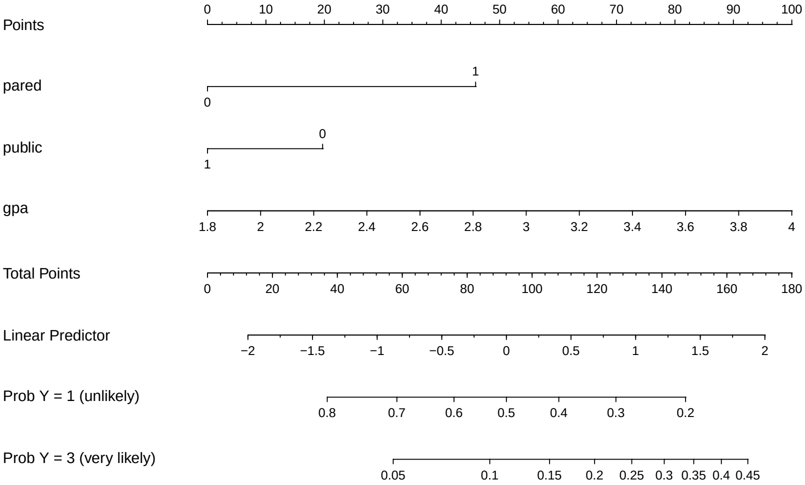
```
mod$coef
```

```
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   0.3663  0.3646   0.0017
R2            0.1553   0.1528  0.1511   0.0018
Intercept     0.0000   0.0000  0.0231  -0.0231
Slope         1.0000   1.0000  1.0170  -0.0170
Emax          0.0000   0.0000  0.0078   0.0078
D             0.1382   0.1359  0.1340   0.0019
U            -0.0038  -0.0038 -0.4637   0.4599
Q             0.1419   0.1397  0.5978  -0.4581
B             0.2155   0.2136  0.2171  -0.0035
g             0.8954   0.8833  0.8814   0.0019
gp            0.2004   0.1958  0.1975  -0.0016
          index.corrected  n
Dxy                0.3670 40
R2                 0.1536 40
Intercept          0.0231 40
Slope              1.0170 40
```

Some Sources for Ordinal Logistic Regression

- A good source of information on fitting these models is https://stats.idre.ucla.edu/r/dae/ordinal-logistic-regression/
  - Another good source, that I leaned on heavily here, using a simple example, is https://onlinecourses.science.psu.edu/stat504/node/177.
  - Also helpful is https://onlinecourses.science.psu.edu/stat504/node/178 which shows a more complex example nicely.

What's in the rest of this slide deck?

The remaining slides present a second, detailed example, for fitting ordinal regression models using some data on asbestos, as well as comparing the fit of `lrm` models to multinomial alternatives. This material may be especially helpful in doing Lab 4.

What's coming after Spring Break

- Fitting Models for Nominal Multi-Categorical Outcomes

A Second Example (Asbestos)

Setup for our Asbestos Example

```
library(knitr); library(janitor); library(magrittr)
library(caret); library(nnet); library(MASS)
library(broom); library(rms)
library(conflicted)
library(tidyverse)

theme_set(theme_bw())

conflict_prefer("select", "dplyr")
conflict_prefer("summarize", "dplyr")
```

```
asbestos <- read_csv("data/asbestos.csv") %>%
    type.convert(as.is = FALSE)
```

Asbestos Exposure in the U.S. Navy

These data describe 83 Navy workers, engaged in jobs involving potential asbestos exposure.

- The workers were either removing asbestos tile or asbestos insulation, and we might reasonably expect that those exposures would be different (with more exposure associated with insulation removal).
- The workers either worked with general ventilation (like a fan or naturally occurring wind) or negative pressure (where a pump with a High Efficiency Particulate Air filter is used to draw air (and fibers) from the work area.)
- The duration of a sampling period (in minutes) was recorded, and their asbestos exposure was measured and classified in three categories:
    - low exposure ($<$ 0.05 fibers per cubic centimeter),
    - action level (between 0.05 and 0.1) and
    - above the legal limit (more than 0.1 fibers per cc).

Source Simonoff JS (2003) *Analyzing Categorical Data*. New York: Springer, Chapter 10.

Our Outcome and Modeling Task

We'll predict the ordinal Exposure variable, in an ordinal logistic regression model with a proportional odds assumption, using the three predictors

- Task (Insulation or Tile),
- Ventilation (General or Negative pressure, which I'll abbreviate as NP) and
- Duration (in minutes).

Exposure is determined by taking air samples in a circle of diameter 2.5 feet around the worker's mouth and nose.

Summarizing the Asbestos Data

We'll make sure the Exposure factor is ordinal...

```
asbestos <- asbestos %>%
    mutate(Exposure = factor(Exposure, ordered = TRUE))
```

```
summary(asbestos[,2:5])
```

```
         Task       Ventilation      Duration
 Insulation:46   General:34   Min.   : 30.0
 Tile      :37   NP     :49   1st Qu.: 85.0
                              Median :138.0
                              Mean   :147.1
                              3rd Qu.:212.5
                              Max.   :300.0
         Exposure
 1_Low      :45
 2_Action   : 6
 3_AboveLimit:32
```

Fitting `polr` models with the `MASS::polr` function

The Proportional-Odds Cumulative Logit Model

We'll use the `polr` function in the `MASS` library to fit our ordinal logistic regression.

- Clearly, Exposure group (3) Above legal limit, is worst, followed by group (2) Action level, and then group (1) Low exposure.
- We'll have two indicator variables (one for Task and one for Ventilation) and then one continuous variable (for Duration).
- The model will have two logit equations: one comparing group (1) to group (2) and one comparing group (2) to group (3), and three slopes, for a total of five free parameters.

Equations to be Fit

The equations to be fit are:

$$log(\frac{Pr(Exposure \leq 1)}{Pr(Exposure > 1)}) = \beta_{0[1]} + \beta_1 Task + \beta_2 Ventilation + \beta_3 Duration$$

and

$$log(\frac{Pr(Exposure \leq 2)}{Pr(Exposure > 2)}) = \beta_{0[2]} + \beta_1 Task + \beta_2 Ventilation + \beta_3 Duration$$

where the intercept term is the only piece that varies across the two equations.

- A positive coefficient β means that increasing the value of that predictor tends to *lower* the Exposure category, and thus the asbestos exposure.

Fitting the Model with the `polr` function in `MASS`

```
model.A <- polr(Exposure ~ Task + Ventilation + Duration,
                data=asbestos, Hess = TRUE)
```

Model Summary

```
summary(model.A)
```

```
Call:
polr(formula = Exposure ~ Task + Ventilation + Duration, data
    Hess = TRUE)

Coefficients:
                   Value Std. Error t value
TaskTile       -2.251333   0.644793 -3.4916
VentilationNP -2.156979   0.567541 -3.8006
Duration       -0.000708   0.003799 -0.1864

Intercepts:
                        Value   Std. Error t value
1_Low|2_Action          -2.0575  0.6611    -3.1123
2_Action|3_AboveLimit -1.5111  0.6344    -2.3820
```

Explaining the Model Summary

The first part of the output provides coefficient estimates for the three predictors.

```
Coefficients:
                  Value Std. Error t value
TaskTile      -2.251333   0.644793 -3.4916
VentilationNP -2.156979   0.567541 -3.8006
Duration      -0.000708   0.003799 -0.1864
```

- The estimated slope for Task = Tile is -2.25. This means that Task = Tile provides less exposure than does the other Task (Insulation) so long as the other predictors are held constant.
- Typically, we would express this in terms of an odds ratio.

Odds Ratios and CI for Model A

```
exp(coef(model.A))
```

```
    TaskTile VentilationNP      Duration
   0.1052589     0.1156740     0.9992922
```

```
exp(confint(model.A))
```

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

```
                   2.5 %    97.5 %
TaskTile      0.02718379 0.3538549
VentilationNP 0.03641039 0.3427734
Duration      0.99187230 1.0069533
```

tidy for polr models...

```
tidy(model.A, conf.int = TRUE)
```

term	estimate	std.error	statistic
TaskTile	-2.251	0.645	-3.492
VentilationNP	-2.157	0.568	-3.801
Duration	-0.001	0.004	-0.186
1_Low|2_Action	-2.057	0.661	-3.112
2_Action|3_AboveLimit	-1.511	0.634	-2.382

term	conf.low	conf.high	coef.type
TaskTile	-3.605	-1.039	coefficient
VentilationNP	-3.313	-1.071	coefficient
Duration	-0.008	0.007	coefficient
1_Low|2_Action	NA	NA	scale
2_Action|3_AboveLimit	NA	NA	scale

`tidy` for `polr` models, exponentiated..

```
tidy(model.A, exponentiate = TRUE, conf.int = TRUE)
```

term	estimate	std.error	statistic
TaskTile	0.105	0.645	-3.492
VentilationNP	0.116	0.568	-3.801
Duration	0.999	0.004	-0.186
1_Low|2_Action	0.128	0.661	-3.112
2_Action|3_AboveLimit	0.221	0.634	-2.382

term	conf.low	conf.high	coef.type
TaskTile	0.027	0.354	coefficient
VentilationNP	0.036	0.343	coefficient
Duration	0.992	1.007	coefficient
1_Low|2_Action	NA	NA	scale
2_Action|3_AboveLimit	NA	NA	scale

Assessing the Ventilation Coefficient

```
Coefficients:
                  Value Std. Error t value
TaskTile      -2.251333   0.644793 -3.4916
VentilationNP -2.156979   0.567541 -3.8006
Duration      -0.000708   0.003799 -0.1864
```

Similarly, the estimated slope for Ventilation = Negative pressure (-2.16) means that Negative pressure provides less exposure than does General Ventilation. We see a relatively modest effect (near zero) associated with Duration.

Summary of Model A: Estimated Intercepts

```
Intercepts:
                            Value   Std. Error t value
1_Low|2_Action             -2.0575  0.6611    -3.1123
2_Action|3_AboveLimit -1.5111  0.6344    -2.3820
```

The first parameter (-2.06) is the estimated log odds of falling into category (1) low exposure versus all other categories, when all of the predictor variables (Task, Ventilation and Duration) are zero. So the first estimated logit equation is:

$$log(\frac{Pr(Exposure \leq 1)}{Pr(Exposure > 1)}) =$$

$$-2.06 - 2.25[Task = Tile] - 2.16[Vent = NP] - 0.0007 Duration$$

Summary of Model A: Estimated Intercepts

```
Intercepts:
                          Value   Std. Error t value
1_Low|2_Action            -2.0575  0.6611    -3.1123
2_Action|3_AboveLimit -1.5111  0.6344    -2.3820
```

The second parameter (-1.51) is the estimated log odds of category (1) or (2) vs. (3). The estimated logit equation is:

$$log(\frac{Pr(Exposure \leq 2)}{Pr(Exposure > 2)}) =$$

$$-1.51 - 2.25[Task = Tile] - 2.16[Vent = NP] - 0.0007 Duration$$

Comparing Model A to an "Intercept only" Model

```r
model.1 <- polr(Exposure ~ 1, data=asbestos)
anova(model.1, model.A)
```

```
Likelihood ratio tests of ordinal regression models

Response: Exposure
                          Model Resid. df Resid. Dev   Test
1                             1        81  147.61971
2 Task + Ventilation + Duration        78   99.87952 1 vs 2
     Df LR stat.     Pr(Chi)
1
2     3 47.74019 2.41857e-10
```

What about AIC and BIC?

Comparing Model A to an “Intercept only” Model

```
AIC(model.1, model.A)
```

```
        df      AIC
model.1  2 151.6197
model.A  5 109.8795
```

```
BIC(model.1, model.A)
```

```
        df      BIC
model.1  2 156.4574
model.A  5 121.9737
```

Comparing Model A to Model without Duration

```r
model.TV <- polr(Exposure ~ Task + Ventilation, data=asbestos)
anova(model.A, model.TV)
```

```
Likelihood ratio tests of ordinal regression models

Response: Exposure
                                Model Resid. df Resid. Dev   Test
1             Task + Ventilation        79   99.91421
2 Task + Ventilation + Duration         78   99.87952 1 vs 2
     Df    LR stat.    Pr(Chi)
1
2     1 0.03469471 0.8522368
```

Comparing Model A to Model without Duration

```
AIC(model.A, model.TV)
```

```
         df      AIC
model.A   5 109.8795
model.TV  4 107.9142
```

```
BIC(model.A, model.TV)
```

```
         df      BIC
model.A   5 121.9737
model.TV  4 117.5896
```

Is a Task*Ventilation Interaction helpful?

```
model.TxV <- polr(Exposure ~ Task * Ventilation, data=asbestos
anova(model.TV, model.TxV)
```

```
Likelihood ratio tests of ordinal regression models

Response: Exposure
                Model Resid. df Resid. Dev   Test    Df
1 Task + Ventilation        79   99.91421
2 Task * Ventilation        78   99.64326 1 vs 2     1
   LR stat.   Pr(Chi)
1
2 0.2709469 0.6026973
```

Is a Task*Ventilation Interaction helpful?

```
AIC(model.TV, model.TxV)
```

```
          df      AIC
model.TV   4 107.9142
model.TxV  5 109.6433
```

```
BIC(model.TV, model.TxV)
```

```
          df      BIC
model.TV   4 117.5896
model.TxV  5 121.7375
```

asbestos Likelihood Ratio Tests

Model	Elements	DF	Deviance	Test	p
1	Intercept	81	147.62	–	–
2	D	80	142.29	vs 1	0.021
3	T	80	115.36	vs 1	< 0.0001
4	V	80	115.45	vs 1	< 0.0001
5	T+V	79	99.91	vs 4	< 0.0001
6	T*V	78	99.64	vs 5	0.60
7	T+V+D	78	99.88	vs 5	0.85

- T = Task
- V = Ventilation
- D = Duration

In-Sample Predictions with our T+V model

```
model.TV <- polr(Exposure ~ Task + Ventilation,
                 data=asbestos)

asbestos <- asbestos %>%
  mutate(TV_preds = predict(model.TV))

asbestos %>% tabyl(TV_preds, Exposure) %>%
  adorn_title() %>% kable()
```

	Exposure		
TV_preds	1_Low	2_Action	3_AboveLimit
1_Low	42	3	10
2_Action	0	0	0
3_AboveLimit	3	3	22

Accuracy of These Classifications?

```
asbestos %>% tabyl(TV_preds, Exposure) %>%
  adorn_title() %>% kable()
```

	Exposure		
TV_preds	1_Low	2_Action	3_AboveLimit
1_Low	42	3	10
2_Action	0	0	0
3_AboveLimit	3	3	22

- Predicting Low exposure led to 42 right and 13 wrong.
- We never predicted Action Level
- Predicting Above Legal Limit led to 22 right and 6 wrong.

Total: 64 right, 19 wrong. Accuracy $= 64/83 = 77.1\%$

5-fold cross-validation for polr model?

We'll use some tools from the caret package for this work, rather than tidymodels because I want to use the polr engine.

```
set.seed(2021)
train.control <- trainControl(method = "cv", number = 5)
modTV_cv <- train(Exposure ~ Task + Ventilation,
                       data = asbestos, method = "polr",
                  trControl = train.control)
```

Results of 5-fold cross-validation `modTV_cv`

```
Ordered Logistic or Probit Regression

83 samples
 2 predictor
 3 classes: '1_Low', '2_Action', '3_AboveLimit'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 67, 66, 67, 65, 67
Resampling results across tuning parameters:

  method    Accuracy   Kappa
  cauchit   0.7716503  0.5464191
  cloglog   0.7605392  0.5277378
  logistic  0.7716503  0.5464191
  loglog    0.7716503  0.5464191
  probit    0.7716503  0.5464191
```

Which `kappa` is that?

Fleiss' `kappa`, or κ describes the extent to which the observed agreement between the predicted classifications and the actual classifications exceeds what would be expected if the predictions were made at random.

- Larger values of κ indicate better model performance ($\kappa = 0$ indicates very poor agreement between model and reality, κ near 1 indicates almost perfect agreement.)

```
Resampling results across tuning parameters:

  method     Accuracy   Kappa
  cauchit    0.7716503  0.5464191
  cloglog    0.7605392  0.5277378
  logistic   0.7716503  0.5464191
  loglog     0.7716503  0.5464191
  probit     0.7716503  0.5464191
```

Is the proportional odds assumption reasonable?

Alternative: fit a multinomial model?

```
mult_TV <- multinom(Exposure ~ Task + Ventilation,
                    data = asbestos, trace = FALSE)
```

View the Multinomial Model?

```
mult_TV
```

```
Call:
multinom(formula = Exposure ~ Task + Ventilation, data = asbes
    trace = FALSE)

Coefficients:
             (Intercept)  TaskTile VentilationNP
2_Action      0.05268936 -1.160153     -2.316099
3_AboveLimit  2.07821627 -2.699743     -2.496044

Residual Deviance: 98.08263
AIC: 110.0826
```

In-Sample Predictions with the multinomial T+V model

```
asbestos <- asbestos %>%
  mutate(TVmult_preds = predict(mult_TV))

asbestos %>% tabyl(TVmult_preds, Exposure) %>%
  adorn_title() %>% kable()
```

	Exposure		
TVmult_preds	1_Low	2_Action	3_AboveLimit
1_Low	42	3	10
2_Action	0	0	0
3_AboveLimit	3	3	22

Compare Models with Likelihood Ratio Test?

```
(LL_multTV <- logLik(mult_TV)) # multinomial model: 6 df
```

```
'log Lik.' -49.04131 (df=6)
```

```
(LL_polrTV <- logLik(model.TV)) # polr model: 4 df
```

```
'log Lik.' -49.9571 (df=4)
```

```
(G = -2 * (LL_polrTV[1] - LL_multTV[1]))
```

```
[1] 1.831584
```

```
pchisq(G, 2, lower.tail = FALSE)
```

```
[1] 0.4001996
```

$p = 0.4$ testing the difference in goodness of fit between the proportional odds model and the more complex multinomial logistic regression model. AIC and BIC?

AIC and BIC for multinomial vs. polr models

```
AIC(mult_TV, model.TV)
```

```
         df      AIC
mult_TV   6 110.0826
model.TV  4 107.9142
```

```
BIC(mult_TV, model.TV)
```

```
         df      BIC
mult_TV   6 124.5957
model.TV  4 117.5896
```

- `mult_TV` is the multinomial model
- `model.TV` is the polr model

Using `rms` to fit ordinal logistic regression models

Proportional Odds Ordinal Logistic Regression with `lrm`

```
d <- datadist(asbestos)
options(datadist = "d")

model_TV_LRM <- lrm(Exposure ~ Task + Ventilation,
                    data = asbestos, x = TRUE, y = TRUE)

# note that Exposure must be an ordered factor
```

POLR results via lrm (slide 1)

```
model_TV_LRM
```

```
Logistic Regression Model

 lrm(formula = Exposure ~ Task + Ventilation,
      data = asbestos, x = TRUE, y = TRUE)

                                     Model Likelihood
                                        Ratio Test
 Obs                          83    LR chi2      47.71
  (1) Low exposure            45    d.f.             2
  (2) Action level             6    Pr(> chi2) <0.0001
  (3) Above legal limit       32
 max |deriv|               3e-10
```

POLR results via `lrm` (slide 2)

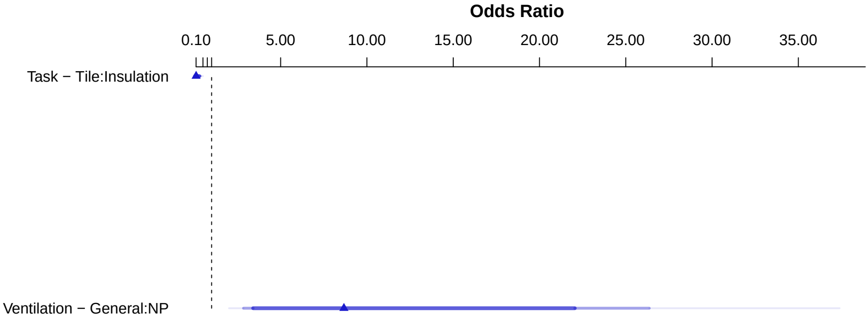
```
lrm(formula = Exposure ~ Task + Ventilation + Duration,
     data = asbestos, x = TRUE, y = TRUE)

 Discrimination    Rank Discrim.
    Indexes          Indexes
 R2       0.526    C       0.854
 g        2.064    Dxy     0.708
 gr       7.877    gamma   0.839
 gp       0.371    tau-a   0.396
 Brier    0.127
```

POLR results via `lrm` (slide 3)

```
lrm(formula = Exposure ~ Task + Ventilation + Duration,
     data = asbestos, x = TRUE, y = TRUE)

                                Coef    S.E.   Wald Z Pr(>|Z|)
y>=(2) Action level              1.9713 0.4695  4.20  <0.0001
y>=(3) Above legal limit         1.4256 0.4348  3.28   0.0010
Task=Tile                       -2.2868 0.6173 -3.70   0.0002
Ventilation=Negative pressure -2.1596 0.5675 -3.81   0.0001
```

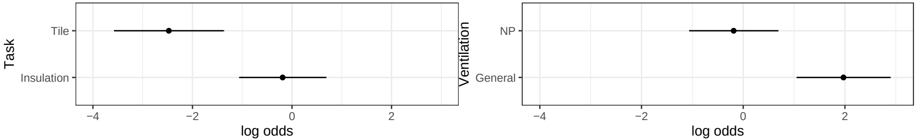
Plot effects of the coefficients (with `lrm`)

```
plot(summary(model_TV_LRM))
```

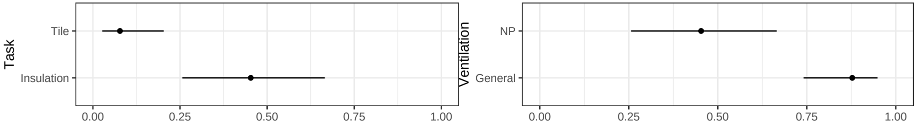


POLR results with lrm, plotted

```
ggplot(Predict(model_TV_LRM))
```




```
ggplot(Predict(model_TV_LRM, fun = plogis))
```



Ordinal Logistic Regression for T+V with orm

```
d <- datadist(asbestos)
options(datadist = "d")

model_TV_ORM <- orm(Exposure ~ Task + Ventilation,
                    data = asbestos, x = TRUE, y = TRUE)

# note that Exposure must be an ordered factor
```

Results for model_TV_ORM fit with orm

(I'll neaten these up on the next two slides.)

```
model_TV_ORM
```

```
Logistic (Proportional Odds) Ordinal Regression Model

 orm(formula = Exposure ~ Task + Ventilation, data = asbestos,
     x = TRUE, y = TRUE)

                        Model Likelihood               Discrim
                              Ratio Test
 Obs              83    LR chi2      47.71    R2
  1_Low           45    d.f.             2    g
  2_Action         6    Pr(> chi2) <0.0001    gr
  3_AboveLimit    32    Score chi2   42.42    |Pr(Y>=median)-0.5
 Distinct Y        3    Pr(> chi2) <0.0001
 Median Y          1
 max |deriv| 6e-05
```

orm fit for T+V model (slide 1 of 2)

```
model_TV_ORM
```

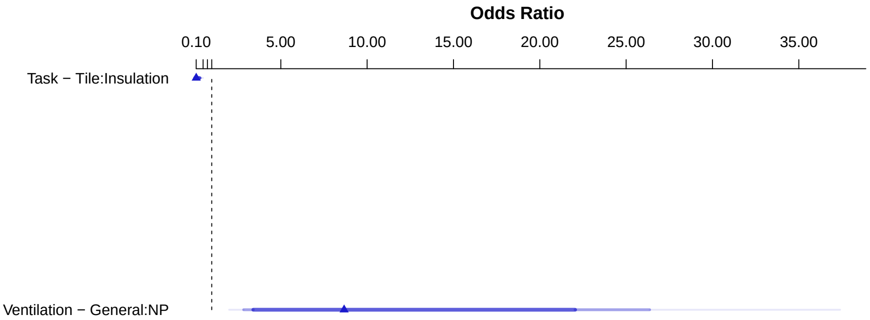
```
Logistic (Proportional Odds) Ordinal Regression Model

orm(formula = Exposure ~ Task + Ventilation,
      data = asbestos, x = TRUE, y = TRUE)

                              Model Likelihood
                                 Ratio Test
 Obs                       83  LR chi2      47.71
  (1) Low exposure         45  d.f.             2
  (2) Action level          6  Pr(> chi2) <0.0001
  (3) Above legal limit 32  Score chi2   42.42
 Distinct Y      3             Pr(> chi2) <0.0001
 Median Y        1
 max |deriv| 6e-05
```

orm fit for T+V model (slide 2 of 2)

```
Logistic (Proportional Odds) Ordinal Regression Model

Discrimination Indexes
R2   0.526    rho   0.697
g    2.064    gr    7.877   |Pr(Y>=median)-0.5| 0.301

                                 Coef    S.E.   Wald Z Pr(>|Z|)
 y>=(2) Action level             1.9713 0.4695  4.20  <0.0001
 y>=(3) Above legal limit        1.4256 0.4348  3.28   0.0010
 Task=Tile                      -2.2868 0.6173 -3.70   0.0002
 Ventilation=Negative pressure -2.1596 0.5675 -3.81   0.0001
```

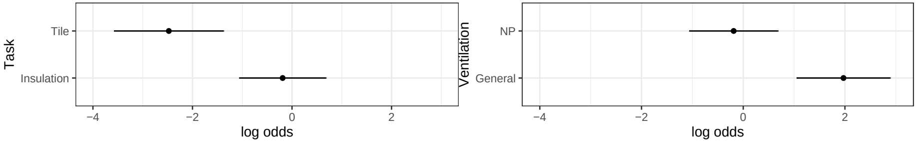
Plot effects of coefficients from `orm`

```
plot(summary(model_TV_ORM))
```

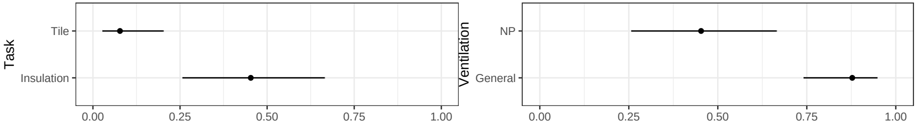


POLR model fit with orm, plotted

```
ggplot(Predict(model_TV_ORM))
```




```
ggplot(Predict(model_TV_ORM, fun = plogis))
```



rms::validate results from lrm

```
set.seed(432)
validate(model_TV_LRM)
```

```
          index                                    index   
           orig training    test optimism corrected  n
Dxy      0.7077   0.7175  0.7082   0.0093    0.6984 40
R2       0.5260   0.5426  0.5183   0.0243    0.5017 40
Intercept 0.0000  0.0000 -0.0279   0.0279   -0.0279 40
Slope    1.0000   1.0000  0.9464   0.0536    0.9464 40
```

rms::validate results from orm

```
set.seed(4322021)
validate(model_TV_ORM)
```

```
      index.orig training   test optimism index.corrected
rho       0.6970   0.7052 0.6975   0.0078          0.6893
R2        0.5260   0.5396 0.5171   0.0225          0.5035
Slope     1.0000   1.0000 0.9700   0.0300          0.9700
g         2.0639   2.1573 2.0194   0.1380          1.9259
pdm       0.3010   0.3217 0.3058   0.0160          0.2850
       n
rho   40
R2    40
Slope 40
g     40
pdm   40
```

rho = Spearman's rank correlation between linear predictor and outcome
R2 = Nagelkerke R-square

Predictions (greater than or equal to)

```
head(predict(model_TV_LRM, type = "fitted"),3)
```

```
  y>=2_Action y>=3_AboveLimit
1  0.07762357       0.0464946
2  0.45306969       0.3243171
3  0.45306969       0.3243171
```

Predictions (individual)

```
head(predict(model_TV_LRM, type = "fitted.ind"),3)
```

```
  Exposure=1_Low Exposure=2_Action Exposure=3_AboveLimit
1      0.9223764        0.03112897             0.0464946
2      0.5469303        0.12875255             0.3243171
3      0.5469303        0.12875255             0.3243171
```

Nomogram?

First, we'll create the functions to estimate the probabilities of falling into groups 1, 2, and 3.

```
model_TV_LRM$coef
```

```
    y>=2_Action y>=3_AboveLimit       Task=Tile
       1.971284        1.425557       -2.286807
 Ventilation=NP
      -2.159559
```

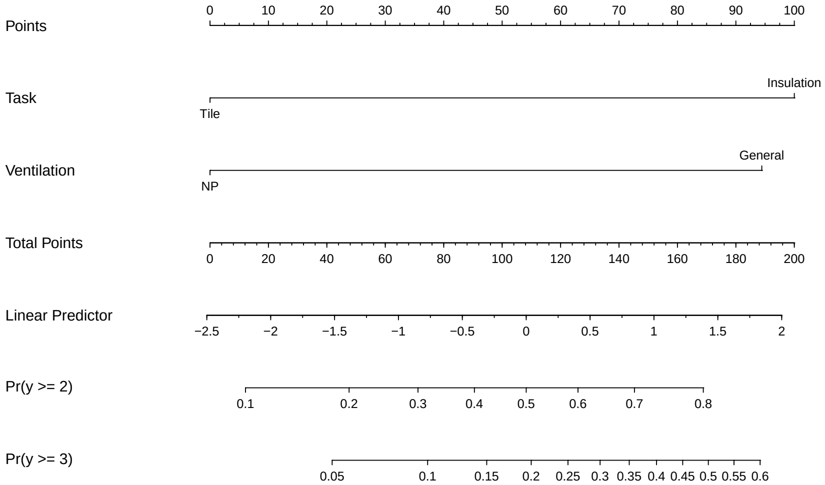
So `plogis` by default uses the first intercept shown, and to get the machine to instead use the second one, we need:

```
fun3 <- function(x) plogis(x - model_TV_LRM$coef[2])
```

Plot the Nomogram

```
plot(nomogram(model_TV_LRM,
         fun = list( 'Pr(y >= 2)' = plogis,
                     'Pr(y >= 3)' = fun3)))
```

Shown on next slide.


Points
0 10 20 30 40 50 60 70 80 90 100
Task
Insulation
Tile
Ventilation
General
NP
Total Points
0 20 40 60 80 100 120 140 160 180 200
Linear Predictor
−2.5 −2 −1.5 −1 −0.5 0 0.5 1 1.5 2
Pr(y >= 2)
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
Pr(y >= 3)
0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55 0.6