

Multinomial Response - Categorical Data

Michael Winton

June 16, 2018

Multicategory Responses

Multinomial Probability Distribution

The categorical response variable Y now has multiple levels $j = 1, \dots, J$, where each category has $\pi_j = P(Y = j)$ and N_j is the number of trials responding with category j . These may or may not be ordinal responses. The PMF is known as the “multinomial probability distribution”. (This is for a *single sample* of n observations.)

$$P(N_1 = n_1, \dots, N_J = n_J) = \frac{n!}{\prod_{j=1}^J n_j!} \prod_{j=1}^J \pi_j^{n_j}$$

The MLE for each π_j is $\hat{\pi}_j = n_j/n$ (ie. proportion in each category).

```
pi_j <- c(0.25, 0.35, 0.2, 0.1, 0.1) # assume these are the true probabilities of each class
n_j <- rmultinom(n = 1, size = 1000, prob = pi_j) # just create 1 sample with 1000 trials
(data.frame(n_j, pihat_j = n_j/1000, pi_j)) # shows n_j randomly generated from each class

##   n_j pihat_j pi_j
## 1 241   0.241 0.25
## 2 355   0.355 0.35
## 3 187   0.187 0.20
## 4 114   0.114 0.10
## 5 103   0.103 0.10
```

Contingency Tables for mixture of two multinomial distributions

If you have *one* multinomial distribution that is characterized by levels of two random variables X, Y , the data can be represented in a contingency table with a row for each X and a column for each Y . The PMF for N_{11}, \dots, N_{IJ} is modified to:

$$P(N_{11} = n_{11}, \dots, N_{IJ} = n_{IJ}) = \frac{n!}{\prod_{i=1}^I \prod_{j=1}^J n_{ij}!} \prod_{i=1}^I \prod_{j=1}^J \pi_{ij}^{n_{ij}}$$

This isn't really a new equation, just new terminology to account for the multiple rows and columns in the contingency table.

```
# set up the true probability table
pi_ij <- c(0.2, 0.3, 0.2, 0.1, 0.1, 0.1) # assume these are the true probabilities of each class
(pi_table <- array(data = pi_ij, dim = c(2, 3), dimnames = list(X = 1:2,
  Y = 1:3)))
```

```
##      Y
## X      1      2      3
## 1 0.2 0.2 0.1
## 2 0.3 0.1 0.1

# simulate a sample with 1000 obsefvations
n_ij <- rmultinom(n = 1, size = 1000, prob = pi_ij)
(c_table <- array(data = n_ij, dim = c(2, 3), dimnames = list(X = 1:2,
  Y = 1:3)))
```

```
##      Y
## X      1      2      3
## 1 189 192 103
## 2 314  96 106
```

```
(pi_hat_table <- c_table/sum(c_table))
```

```
##      Y
## X      1      2      3
## 1 0.189 0.192 0.103
## 2 0.314 0.096 0.106
```

In some cases, instead of having one multinomial distribution, we may intentionally sample from I different groups (corresponding to the rows in the contingency table). This is called the “product multinomial model”. In this case, we have to replace π_{ij} values with $\pi_{j|i}$, so that probabilities in each row add up to 1.

Independence

If X does not have an effect on probabilities of outcomes of Y , they are *independent*. In this case, the probability of a joint outcome of $X = i, Y = j$ is the product of the marginal probabilities $P(X), P(Y)$. In our notation, that is $\pi_{ij} = \pi_{i+} \cdot \pi_{+j}$.

We often want to test for independence; ie. we want to know if $\pi_{j|1} = \dots = \pi_{j|I} = \pi_{+j}$ for all J .

Testing for independence:

$H_0 : \pi_{ij} = \pi_{i+} \cdot \pi_{+j}$ for all i and j . (H_a : at least some aren't equal.)

We use a Pearson χ^2 test. The test statistic is:

$$X^2 = \frac{(\text{observed count} - \text{estimated expected count})^2}{\text{estimated expected count}} = \sum_{i=1}^I \sum_{j=1}^J \frac{(n_{ij} - n\hat{\pi}_{i+}\hat{\pi}_{+j})^2}{n\hat{\pi}_{i+}\hat{\pi}_{+j}} = \sum_{i=1}^I \sum_{j=1}^J \frac{(n_{ij} - n_{i+}n_{+j}/n)^2}{n_{i+}n_{+j}/n}$$

Alternately, we can do a LRT with $\Lambda = \frac{\text{Max likelihood under } H_0}{\text{Max likelihood under } H_0 \text{ or } H_a}$. Our test statistic in this case is:

$$-2\log(\Lambda) = 2 \sum_{i=1}^I \sum_{j=1}^J n_{ij} \log\left(\frac{n_{ij}}{n_{i+}n_{+j}/n}\right)$$

With either test statistic, if H_0 is true, X^2 has an approximate $\chi^2_{(I-1)(J-1)}$ distribution. Reject H_0 if X^2 or $-2\log(\Lambda)$ is great than χ^2 . NOTE: test results for X^2 and $-2\log(\Lambda)$ may vary greatly for small sample sizes; generally we want an expected value in each cell to be > 1 or > 5 .

```
# example of reading data in to a contingency table
```

```
diet <- read.csv("Fiber.csv")
```

```
head(diet)
```

```
##   fiber  bloat count
## 1  bran   high     0
## 2   gum   high     5
## 3 both   high     2
## 4  none   high     0
## 5  bran medium     1
## 6   gum medium     3
```

```
diet$fiber <- factor(diet$fiber, levels = c("none", "bran", "gum", "both"))
diet$bloat <- factor(diet$bloat, levels = c("none", "low", "medium", "high"))
(diet_table <- xtabs(count ~ fiber + bloat, data = diet))
```

```
##           bloat
## fiber  none low medium high
##  none     6  4      2     0
##  bran     7  4      1     0
##   gum     2  2      3     5
##  both     2  5      3     2
```

Now, here are 3 ways to test for independence:

```
(indep_test <- chisq.test(diet_table, correct = FALSE)) # don't let it apply extra correction.
```

```
## Warning in chisq.test(diet_table, correct = FALSE): Chi-squared
```

```
## approximation may be incorrect
```

```
##
```

```
## Pearson's Chi-squared test
```

```
##
```

```
## data:  diet_table
```

```
## X-squared = 16.943, df = 9, p-value = 0.04962
```

```
summary(diet_table)
```

```
## Call: xtabs(formula = count ~ fiber + bloat, data = diet)
```

```
## Number of cases in table: 48
```

```
## Number of factors: 2
```

```
## Test for independence of all factors:
```

```
##  Chisq = 16.943, df = 9, p-value = 0.04962
```

```
## Chi-squared approximation may be incorrect
```

```
library(vcd) # calculates X^2 test only
```

```
## Loading required package: grid
```

```

assocstats(diet_table)  # calculates both X^2 and LRT tests

##                X^2 df P(> X^2)
## Likelihood Ratio 18.880  9 0.026230
## Pearson          16.943  9 0.049621
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.511
## Cramer's V        : 0.343

qchisq(p = 0.95, df = 9)  # this is the critical value that the test stats are being compared to

## [1] 16.91898

(p_x2 <- 1 - pchisq(16.943, df = 9))  # we can also validate the p-values reported above

## [1] 0.04961611

(p_lrt <- 1 - pchisq(18.88, df = 9))  # we can also validate the p-values reported above

## [1] 0.02623278

```

Note that $df = 9 = (I - 1)(J - 1) = (4 - 1)(4 - 1)$. All 3 methods give same result for X^2 test. Small p-values mean we reject the null hypothesis of independence.

We can also get the expected counts to see whether they meet the > 1 or > 5 thresholds for χ^2 being a good approximation. The warning in some of the above methods is because some of these are < 5 .

```

indep_test$expected

##      bloat
## fiber  none  low medium high
##   none 4.25 3.75  2.25 1.75
##   bran 4.25 3.75  2.25 1.75
##   gum  4.25 3.75  2.25 1.75
##   both 4.25 3.75  2.25 1.75

```

Nominal Response Regression Models

We can define *odds* as a comparison of *any pair of response categories*; a popular regression model for multinomial responses is by forming the odds of the remaining $J - 1$ categories against a base category. This multinomial regression model looks like:

$$\log(\pi_j/\pi_1) = \beta_{j0} + \beta_{j1}x_1 + \dots + \beta_{jp}x_p \forall j = 2, \dots, J$$

Note that there is a separate set of β parameters *for each response category*, so each response's log-odds can relate to the explanatory variables in a different way. It's easy to compare other categories:

$$\log(\pi_2/\pi_3) = \log(\pi_2/\pi_1) - \log(\pi_3/\pi_1) = \log(\pi_2) - \log(\pi_3) = (\beta_{20} - \beta_{30}) + (\beta_{21} - \beta_{31})x_1 + \dots + (\beta_{2p} - \beta_{3p})x_p$$

Calculating Probabilities

We can also calculate probabilities by maximum likelihood:

$$\pi_j = \pi_1 \exp(\beta_{j0} + \beta_{j1}x_1 + \dots + \beta_{jp}x_p) \forall j = 2, \dots, J$$

But we need to find π_1 first. Since $\pi_1 + \pi_2 + \dots + \pi_J = 1$, we can get an expression for π_1 :

$$\pi_1 = \frac{1}{1 + \sum_{j=2}^J \exp(\beta_{j0} + \beta_{j1}x_1 + \dots + \beta_{jp}x_p)}$$

Combining, we get:

$$\pi_j = \frac{\exp(\beta_{j0} + \beta_{j1}x_1 + \dots + \beta_{jp}x_p)}{1 + \sum_{j=2}^J \exp(\beta_{j0} + \beta_{j1}x_1 + \dots + \beta_{jp}x_p)} \forall j = 2, \dots, J$$

For a sample of size m observations, the likelihood function is the product of m multinomial distributions, with π_j as described in these equations. Iterative numerical procedures are used to find these MLEs using `nnet::multinom(...)`:

```
wheat <- read.csv("wheat.csv")
head(wheat)

##   class density hardness    size weight moisture   type
## 1   hrw  1.349253 60.32952 2.30274 24.6480 12.01538 Healthy
## 2   hrw  1.287440 56.08972 2.72573 33.2985 12.17396 Healthy
## 3   hrw  1.233985 43.98743 2.51246 31.7580 11.87949 Healthy
## 4   hrw  1.336534 53.81704 2.27164 32.7060 12.11407 Healthy
## 5   hrw  1.259040 44.39327 2.35478 26.0700 12.06487 Healthy
## 6   hrw  1.300258 48.12066 2.49132 33.2985 12.18577 Healthy

levels(wheat$type) # observe that 'Healthy' is the base case

## [1] "Healthy" "Scab"    "Sprout"

library(nnet)
wheat_fit <- multinom(type ~ class + density + hardness + size + weight +
  moisture, data = wheat)

## # weights:  24 (14 variable)
## initial value 302.118379
## iter  10 value 234.991271
## iter  20 value 192.127549
## final value 192.112352
## converged
```

```
summary(wheat_fit)
```

```
## Call:
## multinom(formula = type ~ class + density + hardness + size +
##          weight + moisture, data = wheat)
##
## Coefficients:
##          (Intercept)  classsrw  density  hardness    size    weight
## Scab          30.54650 -0.6481277 -21.59715 -0.01590741 1.0691139 -0.2896482
## Sprout         19.16857 -0.2247384 -15.11667 -0.02102047 0.8756135 -0.0473169
##          moisture
## Scab          0.10956505
## Sprout       -0.04299695
##
## Std. Errors:
##          (Intercept)  classsrw  density  hardness    size    weight
## Scab          4.289865 0.6630948 3.116174 0.010274587 0.7722862 0.06170252
## Sprout         3.767214 0.5009199 2.764306 0.008105748 0.5409317 0.03697493
##          moisture
## Scab          0.1548407
## Sprout        0.1127188
##
## Residual Deviance: 384.2247
## AIC: 412.2247
```

We interpret these results as:

$$\log(\hat{\pi}_{scab}/\hat{\pi}_{healthy}) = 30.55 - 0.65I(class = SRW) - 21.60density - 0.016hardness + \dots$$

$$\log(\hat{\pi}_{sprout}/\hat{\pi}_{healthy}) = 19.17 - 0.22I(class = SRW) - 15.12density - 0.021hardness + \dots$$

NOTE: `mcprofile(...)` and `confint(...)` cannot be used for profile likelihood ratio *intervals* for multinomial models. Wald intervals are possible to calculate because `multinom(...)` gives you standard errors.

LRTs for significance of coefficients are straightforward using `Anova(...)`. Typically we want to explore whether a particular explanatory variable has an effect on *all* response categories (not just one). That hypothesis test is $H_0 : \beta_{jr} = 0 \forall j = 2, \dots, J$.

```
library(car)
Anova(wheat_fit)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: type
##          LR Chisq Df Pr(>Chisq)
## class          0.964  2    0.6175
```

```
## density      90.555  2  < 2.2e-16 ***
## hardness     7.074  2      0.0291 *
## size         3.211  2      0.2008
## weight      28.230  2  7.411e-07 ***
## moisture     1.193  2      0.5506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here are two equivalent ways to see the estimated probabilities for each class for each observation:

```
head(wheat_fit$fitted.values)
```

```
##      Healthy      Scab      Sprout
## 1 0.8552110 0.046396827 0.09839221
## 2 0.7492553 0.021572158 0.22917255
## 3 0.5172800 0.068979903 0.41374011
## 4 0.8982064 0.006740716 0.09505287
## 5 0.5103245 0.176260796 0.31341473
## 6 0.7924907 0.015304122 0.19220522
```

```
head(predict(wheat_fit, newdata = wheat, type = "probs"))
```

```
##      Healthy      Scab      Sprout
## 1 0.8552110 0.046396827 0.09839221
## 2 0.7492553 0.021572158 0.22917255
## 3 0.5172800 0.068979903 0.41374011
## 4 0.8982064 0.006740716 0.09505287
## 5 0.5103245 0.176260796 0.31341473
## 6 0.7924907 0.015304122 0.19220522
```

```
head(predict(wheat_fit, newdata = wheat, type = "class"))
```

```
## [1] Healthy Healthy Healthy Healthy Healthy Healthy
## Levels: Healthy Scab Sprout
```

Odds Ratios for Multinomial Models

Odds of a category j response vs. a category 1 response change by $\exp(c\beta_{jr})$ for every $c - unit$ change in x_r , holding other variables constant.

Odds of a category j response vs. a category j' response change by $\exp[c(\beta_{jr} - \beta_{j'r})]$ for every $c - unit$ change in x_r , holding other variables constant.

```
sd_wheat <- apply(wheat[, -c(1, 7)], MARGIN = 2, FUN = sd) # find st dev for continuous vars
(c_value <- c(1, sd_wheat))
```

```
##      density  hardness      size      weight  moisture
## 1.0000000 0.1313021 27.3561563 0.4906125 7.9154398 2.0332132
```

```
beta_hat_scab <- coef(wheat_fit)[1, 2:7]
beta_hat_sprout <- coef(wheat_fit)[2, 2:7]
```

```

# We calculate the OR (according to a c = 1 * sd change) for all
# variables, but can only change one at a time
or_scab <- exp(c_value * beta_hat_scab)
round(or_scab, 2) # OR for a c-unit increase (scab vs healthy)

##          density hardness      size  weight moisture
##    0.52      0.06      0.65    1.69      0.10      1.25
round(1/or_scab, 2) # OR for a c-unit decrease (scab vs healthy)

##          density hardness      size  weight moisture
##    1.91     17.04      1.55    0.59      9.90      0.80
or_sprout <- exp(c_value * beta_hat_sprout)
round(or_sprout, 2) # OR for a c-unit increase (sprout vs healthy)

##          density hardness      size  weight moisture
##    0.80      0.14      0.56    1.54      0.69      0.92
round(1/or_sprout, 2) # OR for a c-unit decrease (sprout vs healthy)

##          density hardness      size  weight moisture
##    1.25      7.28      1.78    0.65      1.45      1.09

```

Confidence Intervals for Parameters

The `confint(...)` function for multinomial regression does *not* use likelihood ratios; it uses Wald. We calculate CI for Odds Ratios similarly to binomial case, by finding the CI for the linear predictor first, and then exponentiating.

Note how confidence intervals are stored: [coefficients, lower:upper limits, class sequence from fit output].

```

(ci_betas <- confint(wheat_fit, level = 0.95))

## , , Scab
##
##          2.5 %      97.5 %
## (Intercept) 22.13851497 38.954475222
## classsrw    -1.94776958  0.651514098
## density     -27.70474380 -15.489565975
## hardness     -0.03604523  0.004230411
## size         -0.44453927  2.582767006
## weight       -0.41058295 -0.168713512
## moisture     -0.19391723  0.413047326
##
## , , Sprout
##
##          2.5 %      97.5 %
## (Intercept) 11.78496433 26.552173165

```



```

## classssrw      -1.20652328  0.757046542
## density        -20.53461137 -9.698731394
## hardness       -0.03690744 -0.005133494
## size           -0.18459306  1.935820104
## weight         -0.11978643  0.025152642
## moisture       -0.26392179  0.177927888

or_scab_ci <- exp(c_value * ci_betas[2:7, 1:2, 1]) # make sure to get the set for scab
round(or_scab_ci, 2) # OR CI for a c-unit increase

##          2.5 % 97.5 %
## classssrw 0.14  1.92
## density   0.03  0.13
## hardness  0.37  1.12
## size      0.80  3.55
## weight    0.04  0.26
## moisture  0.67  2.32

round(1/or_scab_ci, 2) # OR CI for a c-unit decrease

##          2.5 % 97.5 %
## classssrw 7.01  0.52
## density   38.00  7.64
## hardness  2.68  0.89
## size      1.24  0.28
## weight    25.79  3.80
## moisture  1.48  0.43

or_sprout_ci <- exp(c_value * ci_betas[2:7, 1:2, 2]) # make sure to get the set for sprout
round(or_sprout_ci, 2) # OR CI for a c-unit increase

##          2.5 % 97.5 %
## classssrw 0.30  2.13
## density   0.07  0.28
## hardness  0.36  0.87
## size      0.91  2.59
## weight    0.39  1.22
## moisture  0.58  1.44

round(1/or_sprout_ci, 2) # OR CI for a c-unit decrease

##          2.5 % 97.5 %
## classssrw 3.34  0.47
## density   14.82  3.57
## hardness  2.74  1.15
## size      1.09  0.39
## weight    2.58  0.82
## moisture  1.71  0.70

```

Applying Multinomial Regression model to Contingency Table to Test Independence (Alternative Approach)

The multinomial regression model provides a convenient framework for performing a LRT to test for independence (alternative to method described earlier). We create an *indicator variable* for each level of X (ie. $x_2 = I(x_{j=2})$). If we have *dependence*, the model looks like:

$$H_a : \log(\pi_j/\pi_1) = \beta_{j0} + \beta_{j2}x_2 + \dots + \beta_{jp}x_p \forall j = 2, \dots, J$$

Note that for consistency, we start with $j = 2$ and β_{j2} so that subscripts match class levels for non-base cases. When we have *independence*, the model becomes:

$$H_0 : \log(\pi_j/\pi_1) = \beta_{j0} \forall j = 2, \dots, J$$

This means that, while each category of Y can have a difference π_j , these values do *not* change as a function of X . These two models can be compared in a hypothesis test. This is the same as writing $H_0 : \beta_{j2} = \dots = \beta_{jI} = 0 \forall j = 2, \dots, J$ and H_a : not all of these β parameters are 0, for some j .

In the following example, weights are the counts from each cell of the contingency table, even though we aren't using the crosstab-formatted matrix.

```
library(nnet)
head(diet)

##   fiber  bloat count
## 1  bran   high     0
## 2   gum   high     5
## 3 both   high     2
## 4  none   high     0
## 5  bran medium     1
## 6   gum medium     3

diet_fit_nominal <- multinom(formula = bloat ~ fiber, weights = count,
                             data = diet) # not in crosstab c-table format

## # weights:  20 (12 variable)
## initial value 66.542129
## iter  10 value 54.519963
## iter  20 value 54.197000
## final value 54.195737
## converged

summary(diet_fit_nominal)

## Call:
## multinom(formula = bloat ~ fiber, data = diet, weights = count)
##
## Coefficients:
##           (Intercept)  fiberbran  fibergum  fiberboth
## low          -0.4057626 -0.1538545  0.4055575  1.322135
## medium       -1.0980713 -0.8481379  1.5032639  1.503764
```

```
## high    -12.4401085 -4.1103893 13.3561038 12.440403
##
## Std. Errors:
##      (Intercept)    fiberbran    fibergum    fiberboth
## low      0.6455526    0.8997698    1.190217    1.056797
## medium   0.8163281    1.3451836    1.224593    1.224649
## high    205.2385583 1497.8087307 205.240263 205.240994
##
## Residual Deviance: 108.3915
## AIC: 132.3915
```

To perform LRT for independence, we simply use `Anova(...)`:

```
Anova(diet_fit_nominal)

## # weights:  8 (3 variable)
## initial  value 66.542129
## final    value 63.635876
## converged

## Analysis of Deviance Table (Type II tests)
##
## Response: bloat
##      LR Chisq Df Pr(>Chisq)
## fiber    18.88  9    0.02623 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the p-value (0.02623) is the same as that reported earlier in the $-2\log(\Lambda)$ test with `vcd:assocstats(...)`.

Proportional Odds Model for Ordinal Response data

Basically, when we have ordinal data, we just apply the logit transformation to the *cumulative* probability distribution (CDF). The *cumulative* probability for Y is $P(Y \leq j) = \pi_1 + \pi_2 + \dots + \pi_j$. Then we apply the logit transformation:

$$\text{logit}[P(Y \leq j)] = \log \left[\frac{P(Y \leq j)}{1 - P(Y \leq j)} \right] = \log \left[\frac{\pi_1 + \dots + \pi_j}{\pi_{j+1} + \dots + \pi_J} \right]$$

Then the “Proportional Odds Model” applies a big simplification by assuming the β parameters *except for the intercept* are the same for all explanatory variables (for all $j = 1 \dots J - 1$). This means that *effects of explanatory variables are the same, no matter which cumulative probabilities were used to form the log odds!* If we were to plot these logit curves, they would have the identical shape, just be shifted on the x axis according to intercept terms.

$$\text{logit}[P(Y \leq j)] = \beta_{j0} + \beta_1 x_1 + \dots + \beta_p x_p \quad \forall j = 1, \dots, J - 1$$

Also, this can be rearranged to:

$$P(Y \leq j) = \frac{\exp(\beta_{j0} + \beta_1 x_1 + \dots + \beta_p x_p)}{1 + \exp(\beta_{j0} + \beta_1 x_1 + \dots + \beta_p x_p)} \forall j = 1, \dots, J-1$$

It's important to remember that because of the nature of cumulative probabilities, $\beta_{j0} > \dots > \beta_{10}$.

If we wanted a model where the β parameters were allowed to vary for level of J , that would be the “cumulative probability model” (not covered in class). This model also gets more complicated when there are more than 2 categorical explanatory variables (because we need a m-dimensional contingency table).

We can also look at how we express π_j in terms of these cumulative probabilities:

$$\pi_j = P(Y = j) = P(Y \leq j) - P(Y \leq j-1) = \frac{\exp(\beta_{j0} + \beta_1 x_1 + \dots)}{1 + \exp(\beta_{j0} + \beta_1 x_1 + \dots)} - \frac{\exp(\beta_{j-1,0} + \beta_1 x_1 + \dots)}{1 + \exp(\beta_{j-1,0} + \beta_1 x_1 + \dots)} \forall j = 2, \dots, J-1$$

Estimation and Inference

Parameters are estimated using MLE, with the `MASS::polr(...)` function. If we have one explanatory variable, the hypotheses of interest are $H_0 : \beta_1 = 0$ and $H_a : \beta_1 \neq 0$.

If we reject H_0 , then the ordering of log-odds comparing $P(Y \leq j)$ and $P(Y > j)$ holds. Log odds progressively grow larger (or smaller, depending on sign of β_1).

If we fail to reject H_0 , then the log-odds comparing $P(Y \leq j)$ and $P(Y > j)$ do not depend on this explanatory variable. For the case of two explanatory variables, this is equivalent to *independence*.

WARNING: levels of the Y variable must be properly ordered before using `polr(...)`. Also, all parameters generated by `polr(...)` need to have signs reversed, except for β_{j0} .

```
levels(wheat$type)

## [1] "Healthy" "Scab"      "Sprout"

wheat$type <- factor(wheat$type, levels = c("Scab", "Sprout", "Healthy"))
levels(wheat$type)

## [1] "Scab"      "Sprout"    "Healthy"

library(MASS)
wheat_fit_polr <- polr(formula = type ~ class + density + hardness + size +
  weight + moisture, data = wheat, method = "logistic")
summary(wheat_fit_polr)

##
## Re-fitting to get Hessian

## Call:
## polr(formula = type ~ class + density + hardness + size + weight +
##       moisture, data = wheat, method = "logistic")
##
```

```
## Coefficients:
##               Value Std. Error t value
## classsrw    0.17370   0.391764  0.4434
## density    13.50534   1.713009  7.8840
## hardness    0.01039   0.005932  1.7522
## size       -0.29253   0.413095 -0.7081
## weight      0.12721   0.029996  4.2411
## moisture   -0.03902   0.088396 -0.4414
##
## Intercepts:
##               Value Std. Error t value
## Scab|Sprout    17.5724  2.2460    7.8237
## Sprout|Healthy 20.0444  2.3395    8.5677
##
## Residual Deviance: 422.4178
## AIC: 438.4178
```

We would interpret this as:

$$\text{logit}[P(Y \leq j)] = \beta_{j0} - 0.17I(SRW) - 13.51\text{density} - 0.01\text{hardness} + 0.29\text{size}...$$

with $\beta_{10} = 17.57$ and $\beta_{20} = 20.04$. The `t` value column is the Wald statistic (ie. compare to 1.96). We can also do LRTs with `Anova(...)`:

```
Anova(wheat_fit_polr)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: type
##      LR Chisq Df Pr(>Chisq)
## class      0.197  1  0.65749
## density    98.437  1 < 2.2e-16 ***
## hardness    3.084  1  0.07908 .
## size        0.499  1  0.47982
## weight     18.965  1 1.332e-05 ***
## moisture    0.195  1  0.65872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can also predict probabilities and classes. ** However, `predict(...)` does not calculate the standard errors that would be needed in order to calculate Wald confidence intervals for π_j **.

```
wheat_classes_pi_hat <- predict(wheat_fit_polr, type = "probs")
wheat_classes <- predict(wheat_fit_polr, type = "class")
head(data.frame(wheat_classes_pi_hat, wheat_classes))
```

```
##      Scab    Sprout   Healthy wheat_classes
## 1 0.03661601 0.2738502 0.6895338      Healthy
## 2 0.03351672 0.2576769 0.7088064      Healthy
## 3 0.08379891 0.4362428 0.4799583      Healthy
```

```
## 4 0.01694278 0.1526100 0.8304472 Healthy
## 5 0.11408176 0.4899557 0.3959626 Sprout
## 6 0.02874814 0.2308637 0.7403882 Healthy
```

Odds Ratio

Our odds ratio interpretation as $\exp(\beta_{j0} + \beta_1 x_1 + \dots \beta_p x_p)$ is natural, except that we are using cumulative probabilities. Recall:

$$\text{logit}[P(Y \leq j)] = \log \left[\frac{P(Y \leq j)}{1 - P(Y \leq j)} \right] = \beta_{j0} + \beta_1 x_1 + \dots \beta_p x_p$$

As usual, for a c-unit increase in x_1 :

$$OR = \frac{\text{Odds}_{x_1+c}(Y \leq j)}{\text{Odds}_{x_1}(Y \leq j)} = \exp(c\beta_1)$$

The interpretation is that the odds of $Y \leq j$ vs. $Y > j$ change by $\exp(c\beta_1)$ for every c-unit increase in x_1 , holding all other variables constant. Due to the nature of the proportional odds model, this is the same result *no matter which category is used for j*.

```
head(wheat)
```

```
##   class density hardness    size weight moisture   type
## 1   hrw 1.349253 60.32952 2.30274 24.6480 12.01538 Healthy
## 2   hrw 1.287440 56.08972 2.72573 33.2985 12.17396 Healthy
## 3   hrw 1.233985 43.98743 2.51246 31.7580 11.87949 Healthy
## 4   hrw 1.336534 53.81704 2.27164 32.7060 12.11407 Healthy
## 5   hrw 1.259040 44.39327 2.35478 26.0700 12.06487 Healthy
## 6   hrw 1.300258 48.12066 2.49132 33.2985 12.18577 Healthy
```

```
levels(wheat$type)
```

```
## [1] "Scab"    "Sprout"  "Healthy"
```

```
c_value # we calculated the standard deviations for each parameter to use as our c-unit
```

```
##           density hardness      size      weight  moisture
## 1.0000000  0.1313021 27.3561563  0.4906125  7.9154398  2.0332132
```

```
or_increase <- exp(c_value * (-coef(wheat_fit_polr))) # remember we have to take negative of
or_decrease <- 1/or_increase
```

Interpretations: in all cases, “holding all other variables constant”

- the estimated odds of a scab vs. sprout or healthy are 0.84 times as large for soft rather than hard wheat.
- the estimated odds of a scab vs. sprout or healthy are 0.36 times as large for a c=1sd increase in weight.
- the estimated odds of a scab vs. sprout or healthy are 5.89 times as large for a c=1sd decrease in density.

- the estimated odds of a scab vs. sprout or healthy are 2.74 times as large for a c=1sd decrease in weight.

Also, due to definition of proportional odds model, each of the following statements could also apply to "... a scab or sprout vs. healthy" (ie. comparing different class thresholds).

It is possible to do Wald and LR Intervals for the Odds Ratio.

```
(ci_betas_polr <- confint(wheat_fit_polr, level = 0.95))

## Waiting for profiling to be done...

##
## Re-fitting to get Hessian

##           2.5 %      97.5 %
## classsrw -0.595305729  0.9435846
## density  10.315429541 17.0363926
## hardness -0.001207582  0.0221078
## size     -1.103021561  0.5245184
## weight    0.069318186  0.1872189
## moisture -0.213254701  0.1339876

(or_increase_ci <- exp(c_value * (-ci_betas_polr))) # note these are in reversed upper/lower

##           2.5 %      97.5 %
## classsrw  1.8135853  0.3892301
## density   0.2580925  0.1067876
## hardness  1.0335865  0.5461922
## size      1.7179919  0.7731090
## weight    0.5777096  0.2272011
## moisture  1.5427897  0.7615303

(or_decrease_ci <- 1/exp(c_value * (-ci_betas_polr))) # note these are in reversed upper/lower

##           2.5 %      97.5 %
## classsrw  0.5513940  2.569174
## density   3.8745799  9.364385
## hardness  0.9675049  1.830858
## size      0.5820749  1.293479
## weight    1.7309734  4.401388
## moisture  0.6481765  1.313145
```

Interpretation: Holding all other variables constant, with 95% confidence, the odds of a scab instead of a sprout or healthy response increase by between 3.87 and 9.36 times for every c=1sd decrease in density.