**Section 3.3 – Nominal response models**

Suppose there are J categories for the response variable with corresponding probabilities π1, π2, …, πJ. Using the first category as a “baseline”, we can form “baseline category logits” as log(πj/π1) for j = 2, …, J, which are simply log odds.

When J = 2, we have log(π2/π1) = log(π2/(1-π2)), which is equivalent to log(π/(1-π)) in logistic regression with π = π2.

When there is only one explanatory variable x, we can form the multinomial regression model of

log(πj/π1) = βj0 + βj1x for j = 2, …, J

One can easily compare other categories so that category 1 is not always used. For example, suppose you would like to compare category 2 to 3. Then

log(π2/π1) – log(π3/π1) = log(π2) – log(π3) = log(π2/π3)

and

β20 + β21x – β30 – β31x = β20 – β30 + x(β21 – β31)

For more than one explanatory variable, the model becomes:

log(πj/π1) = βj0 + βj1x1 + … + βjpxp for j = 2, …, J

What is πj only? Consider the case of one explanatory variable x again:

We can re-write the model as . Noting that , we have



Thus,



Also, we can now find that



for j = 2, …, J.

Regression parameters are estimated using maximum likelihood estimation. For a sample of size m, the likelihood function is simply the product of m multinomial distributions with the πj parameters as just found. Iterative numerical procedures are used then to find the parameter estimates.

The multinom() function from the nnet package (within the default installation of R) and the vglm() function from the VGAM package (need to download to install) are the two main ways to find the MLEs. Below are a few notes about these functions:

* multinom()
  + This function is widely used because it is in the default installation of R.
  + The estimation algorithm used by it can take many iterations to converge for some data sets. This especially can occur when the explanatory variables are on different scales or are highly correlated.
  + Increasing the number of iterations (maxit argument) and decreasing the convergence criterion (reltol argument) helps to achieve convergence and match estimates from other model fitting functions.
  + Profile LR inference procedures are not available with related functions.
  + Wald inference procedures are available through the package’s method function for confint() when needed for simple regression parameters. The emmeans package can be used for more complicated situations.
* vglm()
  + This function is in a user-contributed package. While the package has been around for some time, it is much less used for these models.
  + The package's help warns “This package is undergoing continual development and improvement, therefore users should treat everything as subject to change.” This warning should not be taken lightly, because I have needed to change my own code when using this package due to its updates.
  + Profile LR intervals can be calculated for simple regression parameters using the package’s method function for confint(). However, this function cannot perform calculations for more complicated situations, like a linear combination of regression parameters. Also, the mcprofile package does not have the capability to find these intervals.
  + The emmeans package cannot be used with it.

Neither function and corresponding package is as good as I would like. In my notes, I will focus on multinom() and provide additional content with vglm() as needed.

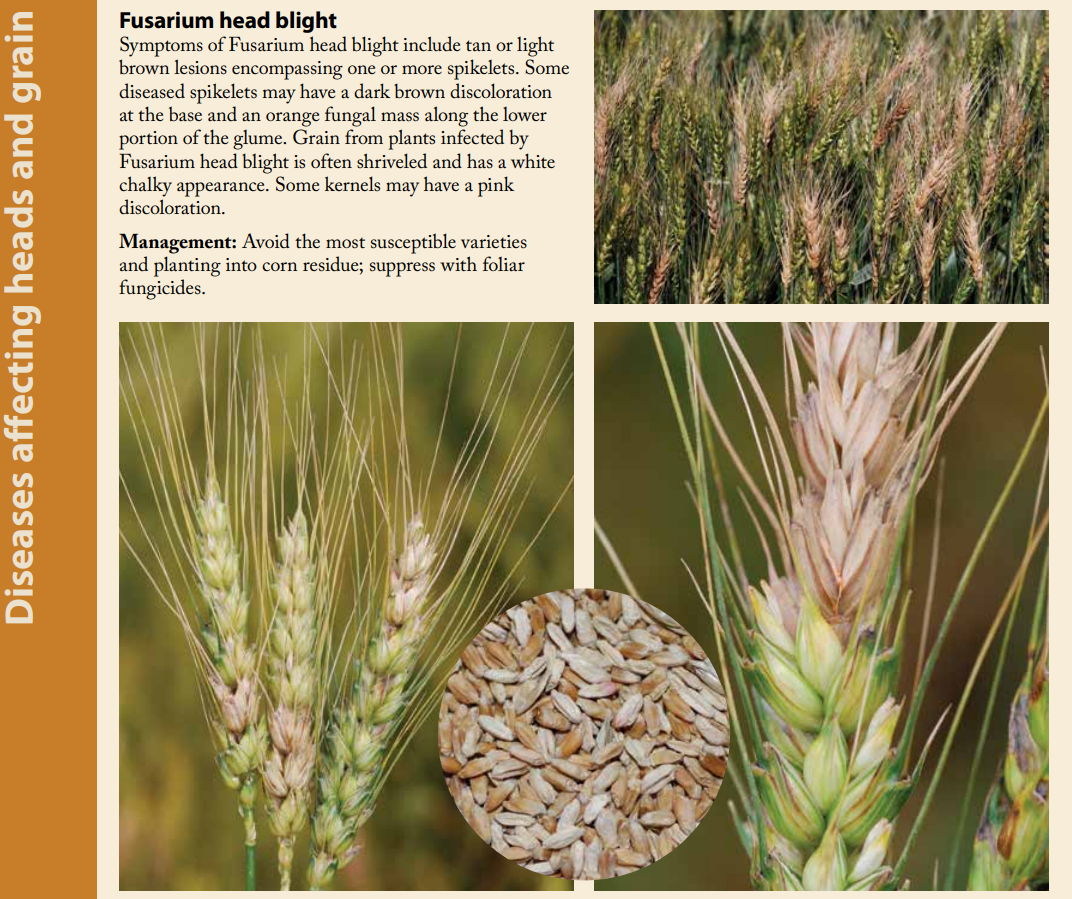
The covariance matrix for the regression parameter estimates follows from using standard likelihood procedures as outlined in Appendix B.

Example: Wheat kernels (Wheat.R, Wheat.csv)

Wheat producers want to identify kernels that are in poor condition after being harvested. To facilitate this identification process, categorization systems have been developed to partition kernels into different categories. For this example, we will look at the categories of “Healthy”, “Sprout”, or “Scab”. In summary,

* Healthy is the preferred condition because these kernels have not been damaged
* Sprout is less preferred than healthy because they have reduced weight and poorer flour quality
* Scab is less preferred than healthy because they come from plants that have been infected by a disease and have undesirable qualities in their appearance

Example of scab kernels from “Wheat Disease Identification”:



Ideally, it would be preferred to make these categorizations for each kernel through using an automated process. To test a new system out, 275 wheat kernels were classified by human examination (assumed to be perfect). The automated system uses information about the class of the wheat kernel (soft red winter or hard red winter) and measurements for density, hardness, size, weight, and moisture for the kernel. Below is part of the data and plots of the data:

> wheat <- read.csv(file = "C:\\data\\wheat.csv", stringsAsFactors = TRUE)

Added after video: The stringsAsFactors = TRUE argument should be included in read.csv() because the class and type variables are categorical.

> head(wheat, n = 3)

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

> tail(wheat, n = 3)

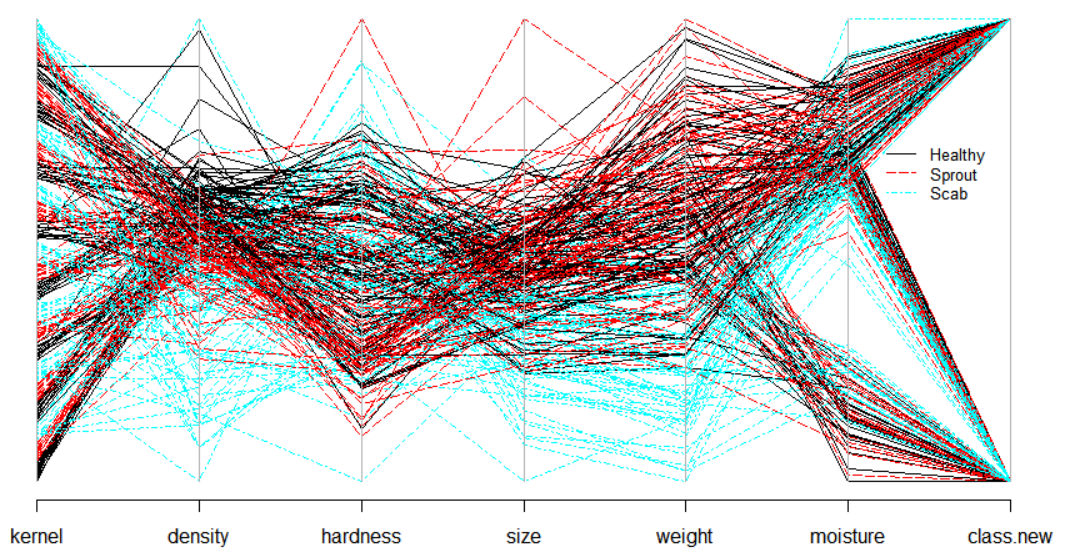
class density hardness size weight moisture type

273 srw 0.8491887 34.06615 1.40665 12.0870 11.92744 Scab

274 srw 1.1770230 60.97838 1.05690 9.4800 12.24046 Scab

275 srw 1.0305543 -9.57063 2.05691 23.8185 12.64962 Scab

Below is a parallel coordinate plot (code is in program).



Comments about the parallel coordinate plot:

* Scab kernels generally have smaller density, size, and weight values
* Healthy kernels generally have higher densities
* There is much overlap for healthy and sprout kernels
* The moisture content appears to be dependent on hard or soft red winter wheat class

I would like to estimate the following model:

log(πj/π1) = βj0 + βj1x1 + … + βj6x6 for j = 2, 3

What is j = 1, 2, and 3? R uses the same method as we saw in previous notes to order the levels of categorical variables.

> levels(wheat$type)

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

Thus, j = 1 is healthy, j = 2 is scab, and j = 3 is sprout.

Below is how we can estimate a multinomial regression model using the explanatory variables in a linear form:

> library(package = nnet)

> mod.fit <- multinom(formula = 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(mod.fit)

Call: multinom(formula = type ~ class + density + hardness + size + weight + moisture, data = wheat)

Coefficients:

(Intercept) classsrw density hardness

Scab 30.54650 -0.6481277 -21.59715 -0.01590741

Sprout 19.16857 -0.2247384 -15.11667 -0.02102047

size weight moisture

Scab 1.0691139 -0.2896482 0.10956505

Sprout 0.8756135 -0.0473169 -0.04299695

Std. Errors:

(Intercept) classsrw density hardness size

Scab 4.289865 0.6630948 3.116174 0.010274587 0.7722862

Sprout 3.767214 0.5009199 2.764306 0.008105748 0.5409317

weight moisture

Scab 0.06170252 0.1548407

Sprout 0.03697493 0.1127188

Residual Deviance: 384.2247

AIC: 412.2247

The estimated models is



and



Notice how R forms an indicator variable for the class of the wheat.

Now that we have the estimated model, many of the basic types of analyses done in Chapter 2 can be performed here. The R code used is very similar as well. Below is a summary.

1. Perform a Wald test for a βjr for some r = 1, …, p.

> sum.fit <- summary(mod.fit)

> test.stat <- sum.fit$coefficients / sum.fit$standard.errors

> p.value <- 2\*(1 - pnorm(q = abs(test.stat), mean = 0, sd = 1))

> round(test.stat,2)

(Intercept) classsrw density hardness size weight

Scab 7.12 -0.98 -6.93 -1.55 1.38 -4.69

Sprout 5.09 -0.45 -5.47 -2.59 1.62 -1.28

moisture

Scab 0.71

Sprout -0.38

> round(p.value,2)

(Intercept) classsrw density hardness size weight

Scab 0 0.33 0 0.12 0.17 0.0

Sprout 0 0.65 0 0.01 0.11 0.2

moisture

Scab 0.48

Sprout 0.70

Why may this test not be of interest?

1. How can we perform a LRT for an explanatory variable?

> library(package = car)

> Anova(mod.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

> qchisq(p = 0.95, df = 2)

[1] 5.991465

The anova() function can be used as well for one variable at a time.

1. How can we estimate the probability of healthy for the first observation?



The easiest way is with the method function for predict().

> predict(object = mod.fit, newdata = wheat[1,], type = "probs")

Healthy Scab Sprout

0.85521096 0.04639683 0.09839221

A new data frame can be created for newdata that allows one to estimate for explanatory variable values not in the data set.

1. How could we determine a classification for the kernels?

Classify a kernel based on the largest probability. Below is the predicted classification using predict() and the type = "class" argument value (default).

> predict(object = mod.fit, newdata = wheat[1,], type = "class")

[1] Healthy

Levels: Healthy Scab Sprout

Methods like cross-validation can be used to obtain a good overall measure of accuracy for these predictions. These methods are not discussed in our course but would be focused on in multivariate and statistical learning courses.

1. What does head(mod.fit$fitted.values) do?

1. How could you estimate the covariance matrix and print it in R?
2. How could you include some type of transformation of an explanatory variable(s) in the model?

Confidence intervals for πj are more complicated to calculate than what we saw for logistic regression. Both nnet and VGAM do not provide ways to perform these calculations. The reason is because one-at-time intervals, meaning one for πHealthy, one for πScab, and one for πSprout may lead to potential probability values that sum to a value greater than one. For example, one-at-a-time 95% Wald intervals for the first observation are

0.7376 < πHealthy < 0.9728

-0.0067 < πScab < 0.0995

0.0143 < πSprout < 0.1825

Of course, πHealthy + πScab + πSprout = 1 needs to occur. If we added the upper limits from the intervals together, we have a total greater than 1! For this reason, a simultaneous confidence region is needed.

These confidence regions are much more difficult to calculate and there are not R functions available for them. Still, I think the one-at-a-time Wald intervals are better than no intervals. Below is how these intervals can be calculated using emmeans with the first observation.

> library(package = emmeans)

> calc.prob <- emmeans(object = mod.fit, specs = ~ type, mode = "prob", at = wheat[1,1:6])

> summary(object = calc.prob, df = Inf, level = 0.95)

type prob SE df asymp.LCL asymp.UCL

Healthy 0.8552 0.0600 Inf 0.7376 0.9728

Scab 0.0464 0.0271 Inf -0.0067 0.0995

Sprout 0.0984 0.0429 Inf 0.0143 0.1825

Degrees-of-freedom method: user-specified

Confidence level used: 0.95

Notes:

* Rather than using specs to indicate the variables of interest, the argument now requires one to indicate the response variable.
* The mode argument is used to indicate that probabilities are wanted.
* The at argument still needs to have an object of a list format. Because a data frame is a special case of a list and there is only one observation here, my code works.
* I specified df = Inf because a t-distribution would be used otherwise for the interval. Wald intervals use the standard normal distribution (remember a t-distribution with an infinite degrees of freedom is equivalent to a standard normal distribution).

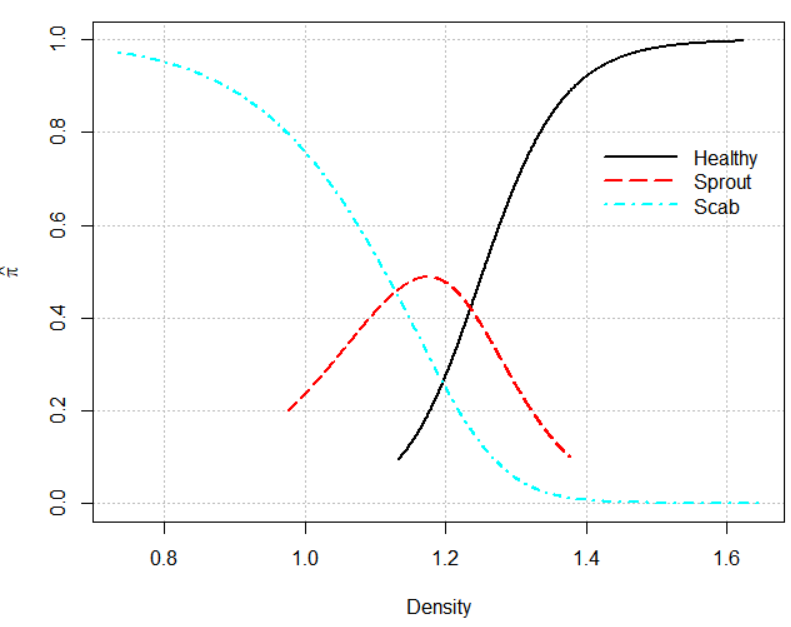
When there is only one explanatory variable in the model, we can easily examine the estimated probabilities through a plot. My program shows how to estimate this model. The model using only density is



and



Through using multiple calls to the curve() function (see program), I constructed the plot below.



Notes:

* The lines are drawn from the smallest to the largest observed density value for a wheat kernel condition.
* We see that the estimated scab probability is the largest for the smaller density kernels. The estimated healthy probability is the largest for the high-density kernels. For density levels in the middle, sprout has the largest estimated probability. The parallel coordinates plot displays similar findings where the density levels tend to follow the scab < sprout < healthy ordering.

We could also construct a plot like this for a model with more than one explanatory variable, but we would need to focus on only one of the variables while conditioning on values for the other variables.