

MULTINOMIAL MODEL

Andrew Zieffler

April 20, 2012

CHAPTER INGREDIENTS

R Packages: *vgam*, *ggplot2*, *sm*

After completion of these notes, you should understand ...

1 Wine Classification Data

The data in the file *Wine-Classification.csv* are from a common dataset used for data mining. This dataset only contains a subset of the original variables. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The original analysis determined the quantities of 13 constituents found in each of the three types of wines. We will use this data to examine whether the intensity of a wine (darkness) is predictive of wine type. We also have the color hue of each wine and its alcohol concentration.

Wine hue and color intensity are two common measures of the appearance of wine color. Both are measured using UV-Visible spectrophotometry. Wine hue is a measurement reported as a ratio of the absorbance in the violet to the absorbance in the green. In general positive values represent a redder value, negative values represent more green. In our sample, all of the wines are red. Wine intensity is a measure of how dark the wine is. Lower values of intensity indicate a more translucent, less intensely colored wine, and higher values indicate a darker more intensely colored wine. The alcohol variable provides the percentage of alcohol by volume. Higher values indicate a higher concentration of alcohol.

```
> wine <- read.csv(file = "/Users/andrewz/Documents/EPsy-8252/  
  Data/Wine-Classification.csv")  
> head(wine)
```

	Type	Color	Hue	Alcohol
1	1	5.64	1.04	14.2
2	1	4.38	1.05	13.2
3	1	5.68	1.03	13.2
4	1	7.80	0.86	14.4
5	1	4.32	1.04	13.2
6	1	6.75	1.05	14.2

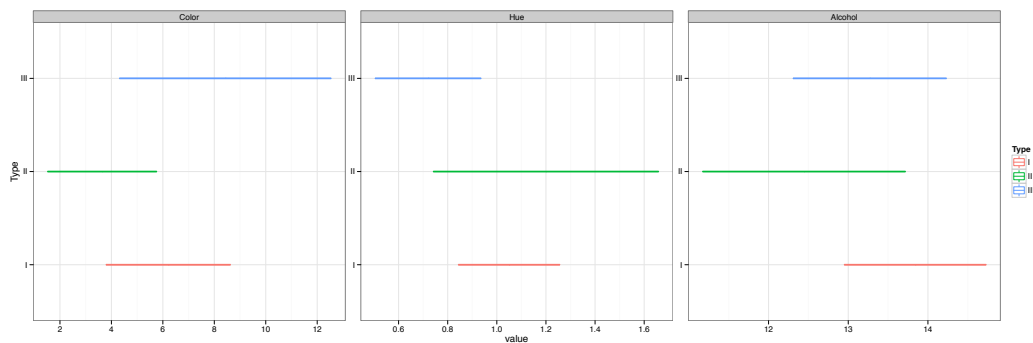
```
> ## Create integer type variable for plotting
> wine$Type <- factor(wine$Type)
> levels(wine$Type) <- c("I", "II", "III")
```

1.1 Plotting the Response Variable

```
> ## Reshape the data so it is useable for ggplot2
> library(reshape2)
> wine2 <- melt(wine, id.vars = "Type")
> head(wine2)
> ## Plot of response by each predictor
> library(ggplot2)
> ggplot(data = wine2, aes(x = value, y = Type, color = Type))
+
+   geom_boxplot() +
+   facet_wrap(~variable, scales = "free") +
+   theme_bw()
```

Figure 1

Relationship between the wine classification and color intensity, hue and alcohol concentration.



2 Multinomial Distribution

Consider a random variable Y_i that may take one of several discrete values, which we index $1, 2, \dots, K$. In the example, the response variable is wine type and it takes the values 'I', 'II', and 'III' which we index 1, 2 and 3. Let,

$$\mathcal{P}(Y_i = k) = p_k$$

denote the probability that the i^{th} response falls in the k^{th} category. In the example p_1 is the probability that the i^{th} wine is type 'I'. Assuming that the response categories are mutually exclusive and exhaustive, we have

$$\sum_{k=1}^K p_k = 1$$

for each i (i.e., the probabilities add up to one). In our example,

$$\begin{aligned} \sum_{k=1}^3 p_k &= p_1 + p_2 + p_3 \\ &= 0.33 + 0.40 + 0.27 \\ &= 1 \end{aligned}$$

For individual level data (as we have in our data set) we can create $K - 1$ indicator (or dummy) variables, Y_k , that takes the value 1 if the i^{th} response falls in the k^{th} category and 0 otherwise. In our example, we would have two indicator variables Y_1 and Y_2 .

Table 1
Wine Data Shown with Two Type Indicator Variables

Type	Y_1	Y_2	Color	Hue	Alcohol
I	1	0	5.64	1.04	14.23
I	1	0	4.38	1.05	13.20
I	1	0	5.68	1.03	13.16
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
II	0	1	1.95	1.05	12.37
II	0	1	3.27	1.25	12.33
II	0	1	5.75	0.98	12.64
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
III	0	0	4.1	0.76	12.86
III	0	0	5.4	0.74	12.88
III	0	0	5.7	0.66	12.81
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

A multinomial distribution is the probability distribution of the outcomes from a multinomial experiment. In our example, the possible outcomes and associated probabilities are

The table completely defines the probabilities associated with every possible outcome from this multinomial experiment. It is the multinomial distribution for this experiment.

A multinomial probability refers to the probability of obtaining a specified frequency in a multinomial experiment. For example, suppose we toss a single die four times. We might ask:

Table 2
Multinomial Distribution for Wine Type

Outcome	Type I	Type II	Type III
Probability	0.33	0.40	0.27

What is the probability that we roll a 1, a 3, and a two 5's? The multinomial formula defines the probability of any outcome from a multinomial experiment. Suppose a multinomial experiment consists of n trials, and each trial can result in any of k possible outcomes: Y_1, Y_2, \dots, Y_k . Suppose, further, that each possible outcome can occur with probabilities p_1, p_2, \dots, p_k . Then, the probability (\mathcal{P}) that Y_1 occurs n_1 times, Y_2 occurs n_2 times, \dots , and Y_k occurs n_k times is

$$\mathcal{P}_K(y_1, y_2, \dots, y_k) = \binom{N}{n_k} p_1^{n_1} p_2^{n_2} p_3^{n_3} \dots p_K^{n_K}$$

where N is the total sample size, n_k is the sample size for the k^{th} category of the response variable Y , and p_k is the probability associated with the k^{th} category of the response variable. Furthermore, let $\binom{N}{n_k}$ represent the *multinomial coefficient* computed as

$$\frac{N!}{(n_1!)(n_2!)(n_3!) \dots (n_K!)}$$

where $!$ is the factorial operator.

For our example,

$$\begin{aligned} \mathcal{P}_3(59, 71, 48) &= \frac{178!}{(59!)(71!)(48!)} 0.33^{59} + 0.40^{71} + 0.27^{48} \\ &= 0.00471 \end{aligned}$$

3 Multinomial Logit Model

We now consider models for the probabilities p_k . In particular, we would like to consider models where these probabilities depend on a vector \mathbf{x}_i of covariates associated with the i^{th} individual. In terms of our example, we would like to model how the probabilities of being type I, type II or type III depend on the color intensity, hue, and alcohol concentration of a wine.

3.1 Multinomial Logits

Perhaps the simplest approach to multinomial data is to nominate one of the response categories as a baseline or reference cell, calculate log-odds for all other categories relative to the baseline, and then let the log-odds be a linear function of the predictors.

Typically the last category, K is chosen as a baseline and the odds that an observation falls in category k as opposed to the baseline is calculated as $\frac{p_k}{p_K}$. In our example we could look at the odds of being type I rather than type III and the odds of being type II rather than type III.


3.2 Modeling the Logits

In the multinomial logit model we assume that the log-odds of each response follow a linear model

$$\log\left(\frac{p_k}{p_K}\right) = \beta_0 + \beta_1(X_{i1}) + \beta_2(X_{i2}) + \dots + \beta_j(X_{ij})$$

This model is analogous to a logistic regression model, except that the probability distribution of the response is multinomial instead of binomial and we have $K - 1$ equations instead of one. The $K - 1$ multinomial logit equations contrast each of categories $1, 2, \dots, K - 1$ with category K , whereas the single logistic regression equation is a contrast between successes and failures. If $K = 2$ the multinomial logit model reduces to the usual logistic regression model.

Note that we need only $K - 1$ equations to describe a variable with K response categories and that it really makes no difference which category we pick as the reference cell, because we can always convert from one formulation to another. In our example with $K = 3$ categories we contrast categories 1 versus 3 and 2 versus 3. The missing contrast between categories 1 and 2 can easily be obtained in terms of the other two, since

$$\log\left(\frac{p_1}{p_2}\right) = \log\left(\frac{p_1}{p_3}\right) - \log\left(\frac{p_2}{p_3}\right).$$


3.3 Maximum Likelihood Estimation

Estimation of the parameters of this model by maximum likelihood proceeds by maximization of the multinomial likelihood with the probabilities p_k viewed as functions of the parameters $\beta_0, \beta_1, \dots, \beta_j$. This usually requires numerical procedures (e.g., Fisher scoring or Newton–Raphson methods often work rather well). Most statistical packages include a multinomial logit procedure.

4 Fitting the Multinomial Model Using R

To fit a multinomial model in R, we use the `vglm()` function from the **VGAM** library. We initially only look at color intensity as a predictor of type.

```
> library(VGAM)
> vglm.1 <- vglm(Type ~ Color, data = wine, family =
  multinomial)
> summary(vglm.1)
```

```

Call:
vglm(formula = Type ~ Color, family = multinomial, data = wine)

Pearson Residuals:
           Min      1Q  Median      3Q      Max
log(mu[,1]/mu[,3])  -3 -0.5 -0.140  0.8      2
log(mu[,2]/mu[,3])  -3 -0.1 -0.002  0.2      8

Coefficients:
              Estimate Std. Error z value
(Intercept):1      3.8         0.9      4
(Intercept):2     12.5         1.6      8
Color:1           -0.6         0.1     -4
Color:2           -2.6         0.4     -7

Number of linear predictors: 2

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

Dispersion Parameter for multinomial family: 1

Residual deviance: 211 on 352 degrees of freedom

Log-likelihood: -106 on 352 degrees of freedom

Number of iterations: 7

```

In terms of our example, fitting the multinomial logit model leads to a deviance of 210.9 on 352 df . Based on the output, the coefficients suggest that higher values of color intensity decrease the likelihood that the wine will be classified as type I as opposed to type III. We see the same thing for wine classified as type II, but the effects are even more pronounced.

We can write the two prediction equations as,

$$\log \left(\frac{\mathcal{P}(Y = \text{Type I})}{\mathcal{P}(Y = \text{Type III})} \right) = 3.8 - 0.6(\text{Color})$$

and

$$\log \left(\frac{\mathcal{P}(Y = \text{Type II})}{\mathcal{P}(Y = \text{Type III})} \right) = 12.5 - 2.6(\text{Color})$$

Of most interest are the coefficients associated with the effect of color intensity. Remember, each effect is in reference to the baseline group (type III) and in the logit scale. For example, in comparing type III to type I, a one unit change in color intensity is associated with an average change of -0.6 logits. In comparing type III to type II, a one unit change in color intensity is associated with an average change of -2.6 logits. Figure 2 (a) shows the linear relationship expressed between color intensity and the log-odds (logits) of wine type versus the reference category of type III.

We can re-express the relationship in the two equations as odds as,

$$\frac{\mathcal{P}(Y = \text{Type I})}{\mathcal{P}(Y = \text{Type III})} = e^{3.8-0.6(\text{Color})}$$

and

$$\frac{\mathcal{P}(Y = \text{Type II})}{\mathcal{P}(Y = \text{Type III})} = e^{12.5-2.6(\text{Color})}$$

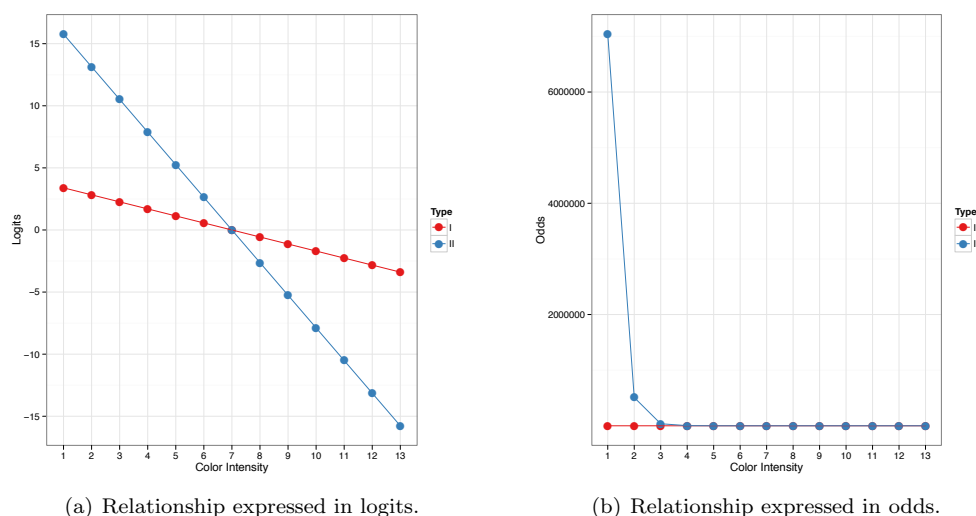
```
> exp(coef(vglm.1))
```

(Intercept):1	(Intercept):2	Color:1	Color:2
44.4549	259337.9463	0.5681	0.0722

Expressed as odds, a one unit change in color intensity is associated with an average change in odds of 0.57 to 1 (type I vs. type III) and an average change in odds of 0.07 to 1 (type II vs. type III). Figure 2 (b) shows the nonlinear relationship expressed between color intensity and the odds of wine type versus the reference category of type III.

Figure 2

Relationship between the color intensity and wine type.



4.1 Expressing the Relationship as Probabilities

To best aid interpretation of the coefficients, we convert the relationship to the probability scale. When there are more than two groups, computing probabilities is a little more complicated than it was in logistic regression. For $k = 1, 2, \dots, K - 1$

$$\mathcal{P}(Y = k) = \frac{e^{\beta_0 + \beta_1(X_{i1})}}{1 + \sum_{k=1}^{K-1} e^{\beta_0 + \beta_1(X_{i1})}}$$

For the reference category,

$$\mathcal{P}(Y = K) = \frac{1}{1 + \sum_{k=2}^K e^{\beta_0 + \beta_1(X_{i1})}}$$

In other words, you compute each of the $K - 1$ log odds and exponentiate them. Once you have done that the calculation of the probabilities is straightforward. For example, at the color intensity value of two, the log-odds are

```
> ## Compute logits at color = 2
> exp(3.79 - 0.57 * 2)
```

```
[1] 14.2
```

```
> exp(12.47 - 2.63 * 2)
```

```
[1] 1353
```

The probabilities are then,

$$\begin{aligned}\mathcal{P}(\text{Type I}) &= \frac{14}{1 + 14 + 1353} = 0.010482 \\ \mathcal{P}(\text{Type II}) &= \frac{1353}{1 + 14 + 1353} = 0.98879 \\ \mathcal{P}(\text{Type III}) &= \frac{1}{1 + 14 + 1353} = 0.00073075\end{aligned}$$

We can use R directly to compute this using the `predict()` function and the argument `type="response"`.

```
> ## Create new data frame
> new <- data.frame(Color = 2)
> ## Use model to produce probabilities
> predict(vglm.1, type="response", newdata = new)
```

```
      I      II      III
1 0.0105 0.989 0.000731
```

It is easier to represent the probabilities in a plot that shows the three wine types. To do so, we first create a data frame that contains a sequence of color values. Then we use the `predict()` function to produce the fitted values. The probabilities are produced side-by-side, which we stack on top of each other for use in `ggplot()`. We then bind the color values to this new stacked data frame to coordinate each color value with a predicted probability.


```

> fit <- data.frame(expand.grid(
  Color = 1:13
))
> plotdata <- data.frame(predict(vglm.1, newdata = fit, type =
  "response"))
> head(plotdata)

```

	I	II	III
1	0.00135	0.9986	0.0000533
2	0.01048	0.9888	0.0007307
3	0.07623	0.9144	0.0093556
4	0.36479	0.5564	0.0788094
5	0.63522	0.1232	0.2415830
6	0.59026	0.0146	0.3951838

```

> plotdata <- stack(plotdata, select =c(I, II, III))
> plotdata <- cbind(fit, plotdata)
> head(plotdata)

```

	Color	values	ind
1	1	0.00135	I
2	2	0.01048	I
3	3	0.07623	I
4	4	0.36479	I
5	5	0.63522	I
6	6	0.59026	I

Then we plot those probabilities using `ggplot()`

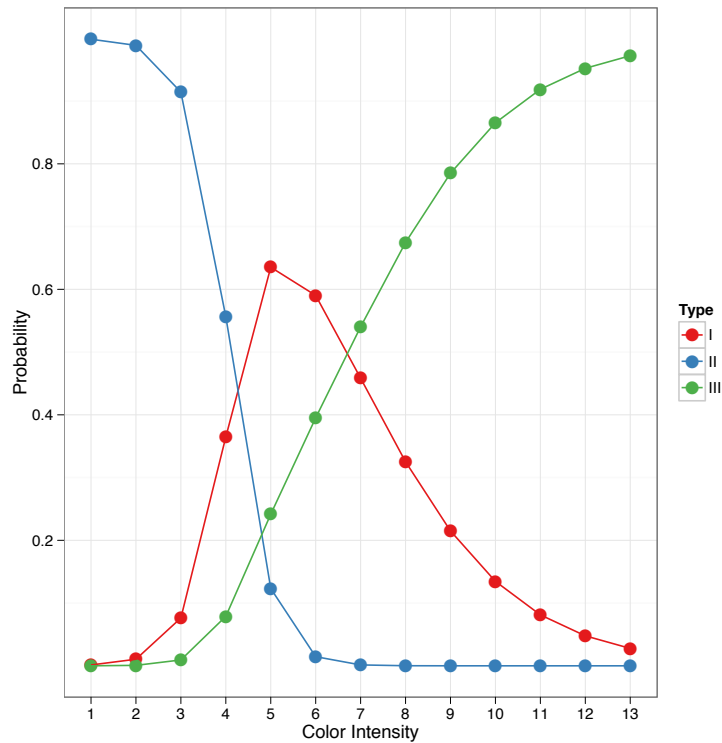
```

> ggplot(data = plotdata, aes(x = Color, y = values, color =
  ind)) +
  geom_point(size = 4) +
  geom_line() +
  scale_x_discrete() +
  scale_color_brewer(palette = "Set1", name = "Type") +
  xlab("Color Intensity") +
  ylab("Probability") +
  theme_bw()

```

Figure 3

Relationship between the probability of wine classification and color intensity.



5 Fit a Model with Multiple Predictors

Now we will fit a model using color intensity, hue and alcohol concentration to predict variation in wine type.

```
> vglm.2 <- vglm(Type ~ Color + Hue + Alcohol, data = wine,
  family = multinomial)
> summary(vglm.2)
```

```
Call:
vglm(formula = Type ~ Color + Hue + Alcohol, family = multinomial,
  data = wine)
```

Pearson Residuals:

	Min	1Q	Median	3Q	Max
log(mu[,1]/mu[,3])	-3	-0.07	-0.0170	0.08	2
log(mu[,2]/mu[,3])	-1	-0.04	-0.0006	0.04	2

Coefficients:

	Estimate	Std. Error	z value
(Intercept):1	-40.2	11.6	-3
(Intercept):2	22.7	12.6	2
Color:1	-0.5	0.4	-1
Color:2	-2.1	0.5	-4

```

Hue:1          19.1          4.7          4
Hue:2          19.4          4.6          4
Alcohol:1       2.0          0.9          2
Alcohol:2      -2.3          1.0         -2

Number of linear predictors: 2

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

Dispersion Parameter for multinomial family: 1

Residual deviance: 80.8 on 348 degrees of freedom

Log-likelihood: -40.4 on 348 degrees of freedom

Number of iterations: 8

```

In terms of our example, fitting the multinomial logit model leads to a deviance of 80.8 on 348 *df*. Based on the output, the coefficients suggest that higher values of color intensity decrease the likelihood that the wine will be classified as type I as opposed to type III, while higher values of hue (darker wines) and alcohol concentration increase the likelihood of classification of type I as opposed to type III.

We see the same effect of hue and color intensity for wine classified as type II, but the effects of color intensity are even more pronounced. However, higher alcohol concentration decreases the likelihood of wine classified as type II as opposed to type III.

Again, to understand these relationships, we plot the predicted probabilities

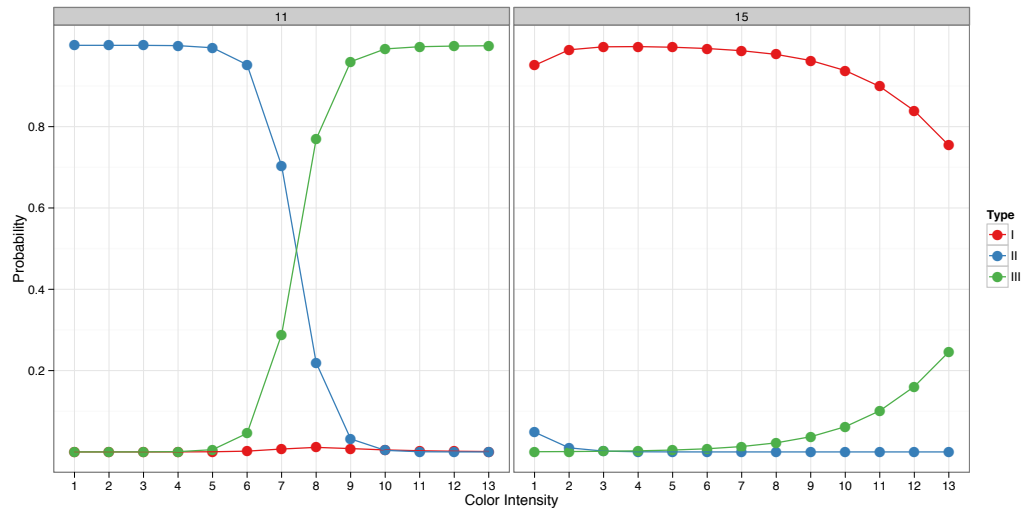
```

> ## Create data frame of values to use in the model
> fit2 <- data.frame(expand.grid(
  Color = 1:13,
  Hue = mean(wine$Hue),
  Alcohol = c(11, 15)
))
> ## Fit the model to the new data
> plotdata <- data.frame(predict(vglm.2, newdata = fit2, type
  = "response"))
> plotdata <- stack(plotdata, select = c(I, II, III))
> plotdata <- cbind(fit2, plotdata)
> plotdata$Alcohol <- factor(plotdata$Alcohol)
> ## Plot probabilities vs. color values
> ggplot(data = plotdata, aes(x = Color, y = values, color =
  ind)) +
  geom_point(size = 4) +
  geom_line() +
  scale_x_discrete() +
  scale_color_brewer(palette = "Set1", name = "Type") +
  xlab("Color Intensity") +
  ylab("Probability") +
  facet_wrap(~Alcohol) +
  theme_bw()

```

Figure 4

Relationship between the probability of wine classification and color intensity at the mean hue value faceted by alcohol concentration.



6 Using Data From a Table to Fit a Multinomial Model

Consider the following table which tabulates data on people's the frequencies for 3165 currently married women classified by age, grouped in five-year intervals, and current use of contraception, classified as sterilization, other methods, and no method. (These data come from the final report of the *Demographic and Health Survey* conducted in El Salvador in 1985.)

Table 3

Current Use of Contraception by Age of Married Women. El Salvador, 1985

Age	Contraceptive Method			All
	Ster.	Other	None	
15–19	3	61	232	296
20–24	80	137	400	617
25–29	216	131	301	648
30–34	268	76	203	547
35–39	197	50	188	435
40–44	150	24	164	338
45–49	91	10	183	284
All	1005	489	1671	3165

We can use these data to answer the question of whether age is predictive of type of contraception used for married women in El Salvador.

We will use the `expand.grid()` function chained in the `data.frame()` function to enter these data. When entering the variables, start with the 'innermost' variable in the table and move out. For example, start with `Contraception` and then `Age`. For now, we are

only entering the variables and their values or categories—not the frequencies. For the **Age** predictor, rather than entering the entire name for the category, the midpoint will be used (e.g., ‘17’ will represent the ‘15–19’ category).

```
> tab <- data.frame(expand.grid(
  Contraception = c("Sterilization", "Other", "None"),
  Age = c(17, 22, 27, 32, 37, 42, 47)
))
> head(tab)
```

	Contraception	Age
1	Sterilization	17
2	Other	17
3	None	17
4	Sterilization	22
5	Other	22
6	None	22

We can now create a frequency variable (**Freq**) that contains all of the cell frequencies. Notice from the **head()** output that by starting with the ‘innermost’ variable in the table and moving out, the order of the data frame rows corresponds to the first row of the table, then the second row, etc. We need to enter the frequencies so that they are in the same order.

```
> tab$Freq <- c(
  3, 61, 232,
  80, 137, 400,
  216, 131, 301,
  268, 76, 203,
  197, 50, 188,
  150, 24, 164,
  91, 10, 183
)
> head(tab)
```

	Contraception	Age	Freq
1	Sterilization	17	3
2	Other	17	61
3	None	17	232
4	Sterilization	22	80
5	Other	22	137
6	None	22	400

6.1 Exploratory Analysis

We can start the analysis by creating a contingency table of the data.

```
> my.tab <- xtabs(Freq ~ Age + Contraception, data = tab)
> my.tab
```

	Contraception		
Age	Sterilization	Other	None
17	3	61	232

22	80	137	400
27	216	131	301
32	268	76	203
37	197	50	188
42	150	24	164
47	91	10	183

Although we had this originally (it is, after all, what we entered the data from) we will use the table to compute the empirical logits. The empirical logits are the log odds based on the sample data. Using the contraception method of **None** as the reference we create the logits for both the sterilization group (**logit1**) and the other group (**logit2**).

```
> logit1 <- log(my.tab[ , 1] / my.tab[ , 3])
> logit2 <- log(my.tab[ , 2] / my.tab[ , 3])
```

We can then create a data frame that includes the age predictor and the two sets of empirical logits.

```
> emp <- data.frame(
  Age = c(17, 22, 27, 32, 37, 42, 47),
  Logit1 = logit1,
  Logit2 = logit2
)
> emp
```

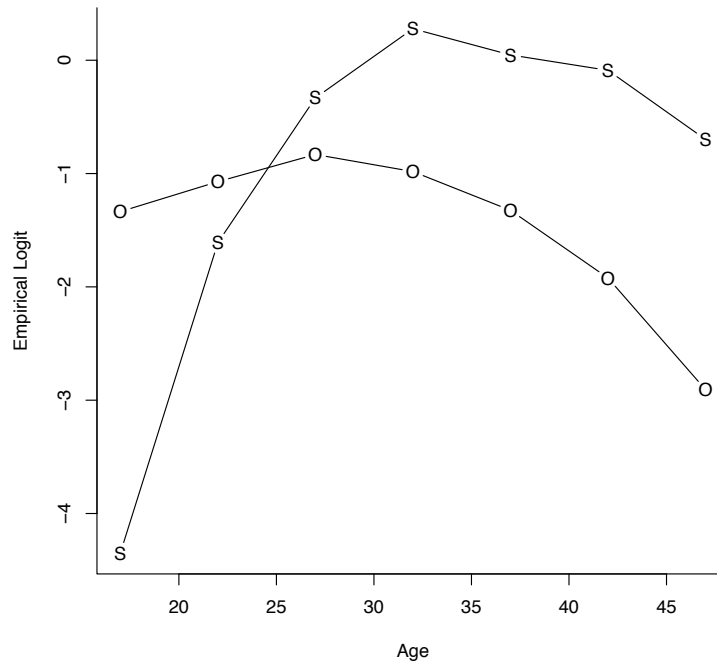
	Age	Logit1	Logit2
17	17	-4.3481	-1.336
22	22	-1.6094	-1.071
27	27	-0.3318	-0.832
32	32	0.2778	-0.982
37	37	0.0468	-1.324
42	42	-0.0892	-1.922
47	47	-0.6986	-2.907

Finally, we plot the empirical logits versus the age predictor separately for the two groups. We change the plotting characters to 'S' and 'O' respectively and also add connecting lines so that we can see whether the relationship is linear.

```
> ## Plot the first set of logits
> plot(Logit1 ~ Age, data = emp, pch = "S", type = "b")
> ## Add the second set of logits
> points(Logit2 ~ Age, data = emp, pch = "O", type = "b")
```

Figure 5

Relationship between age and the empirical logits for the sterilization vs. none and other vs. none groups.



The relationship between age and the empirical logits does not appear linear. We may want to consider a quadratic model rather than a linear model.

6.2 Fitting the Multinomial Model to the Data

To fit the model we again use the `vglm()` function. Since we are using grouped data however, we add the `weights=` argument to weight each row of the data frame by the corresponding cell frequency. We will treat the `Age` predictor as an interval level predictor rather than as categorical. We will fit both the linear and quadratic models, as well as an intercept only reference model.

```
> model.int <- vglm(Contraception ~ 1, data = tab, weights =
  Freq, family = multinomial)
> model.linear <- vglm(Contraception ~ Age, data = tab,
  weights = Freq, family = multinomial)
> model.quad <- vglm(Contraception ~ Age + I(Age ^ 2), data =
  tab, weights = Freq, family = multinomial)
```

We then compute the likelihood ratio test to compare the two models using the `lrtest()` function.

```
> lrtest(model.quad, model.linear, model.int)
```

```

Likelihood ratio test

Model 1: Contraception ~ Age + I(Age^2)
Model 2: Contraception ~ Age
Model 3: Contraception ~ 1
#Df LogLik Df Chisq Pr(>Chisq)
1 36 -2883
2 38 -3020 2 274 <0.0000000000000002 ***
3 40 -3133 2 227 <0.0000000000000002 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The results of this test, $\chi^2(2) = 273.5$, $p < 0.001$, suggests that the additional quadratic term is warranted in the model.

6.3 Pseudo R^2

Another method of comparing models is through the comparison of the pseudo R^2 values from the fitted models. There are several methods to calculate a pseudo R^2 value (see, http://www.ats.ucla.edu/stat/mult_pkg/faq/general/pseudo_rsquareds.htm). These measures are called pseudo R^2 s because the computation is comparable to that of R^2 . They, however, are not interpreted the same way. The simplest method, based on McFadden's method, is

$$R_D^2 = \frac{\text{deviance}(\text{null}) - \text{deviance}(\text{full})}{\text{deviance}(\text{null})}$$

where $\text{deviance}(\text{null})$ is the deviance obtained from a reduced (null) model and $\text{deviance}(\text{full})$ is the deviance obtained from a full model. The deviance of the intercept and quadratic models can be obtained from the `summary()` output, or directly with the `deviance()` function. Computing McFadden's pseudo R^2 for the comparison of the intercept-only model to the linear model, and the intercept-only model to the quadratic model results in,

```
> (deviance(model.int) - deviance(model.linear))/deviance(
  model.int)
```

```
[1] 0.0362
```

```
> (deviance(model.int) - deviance(model.quad))/deviance(
  model.int)
```

```
[1] 0.0799
```

The proportion reduction the deviances between the null model (intercept only model) and quadratic model reduced the deviance by 7% as opposed to using the linear model which reduces the deviance by only 3%.

Typically these measures are not reported and are just introduced here as a courtesy since many software packages provide them. Homer and Lemeshow (2000) write, "...low R^2

values in logistic regression are the norm and this presents a problem when reporting their values to an audience accustomed to seeing linear regression values. ... Thus [arguing by reference to running examples in the text] we do not recommend routine publishing of R^2 values with results from fitted logistic models. However, they may be helpful in the model building state as a statistic to evaluate competing models" (p. 164).

Pseudo R^2 s cannot be interpreted independently or compared across datasets. They are only valid and useful in evaluating multiple models predicting the same outcome on the same dataset. In other words, a pseudo R^2 statistic without context has little meaning. **A pseudo R^2 only has meaning when compared to another pseudo R^2 of the same type, on the same data, predicting the same outcome.** In this situation, the higher pseudo R^2 indicates which model better predicts the outcome.

6.4 Model Coefficients

The estimates for the parameters are:

```
> coef(model.quad)
```

(Intercept):1	(Intercept):2	Age:1	Age:2	I(Age^2):1
-12.26573	-4.41895	0.69999	0.25931	-0.00973
I(Age^2):2				
-0.00476				

6.5 Final Plot

We will again plot the probabilities associated with contraception choice conditioned on age.

```
> ## Create data frame to predict from
> a <- data.frame(
  Age = c(17, 22, 27, 32, 37, 42, 47)
)
> ## Compute probabilities from model
> plotdata <- data.frame(predict(model.quad, type = "response",
  , newdata = a))
> ## Get data frame ready for ggplot
> plotdata <- stack(plotdata, select = c(Sterilization, Other,
  None))
> plotdata <- cbind(a, plotdata)
> ## Plot probabilities
> ggplot(data = plotdata, aes(x = Age, y = values, col = ind))
+
  geom_point(size = 4) +
  geom_line() +
  scale_color_brewer(palette = "Set1", name = "Type") +
  xlab("Age") +
  ylab("Probability") +
  theme_bw()
```

Figure 6
Probability associated with three different contraception choices conditioned on age for married women in El Salvador.

