

# Multinomial Regression

2018-04-14

## Preparation

The data in the file *italian-wine.csv* are based on those from a common dataset used for data mining. This dataset contain measurements obtained from a chemical analysis on 175 Italian red wines. The grapes used to make these wines were all grown in the same region in Italy (Piedmont) but derived from three different cultivars (Barbera, Dolcetto, and Nebbiolo). We will use this data to examine whether we can classify red wine cultivars based on the hue (color), intensity (darkness), and alcohol concentration of a wine.

Wine hue and color intensity are two common measures of the appearance of wine. 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. 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.

```
# Load libraries
library(AICcmodavg)
library(broom)
library(corr)
library(dplyr)
library(ggplot2)
library(nnet) # Used to fit multinomial models
library(readr)
library(sm)
library(texreg)
library(tidyr)

# Read in italian wine data
wine = read_csv("~/Dropbox/epsy-8252/data/italian-wines.csv")
head(wine)
```

cultivar	intensity	hue	alcohol
Dolcetto	5.64	1.04	14.2
Dolcetto	4.38	1.05	13.2
Dolcetto	5.68	1.03	13.2
Dolcetto	7.80	0.86	14.4
Dolcetto	4.32	1.04	13.2
Dolcetto	6.75	1.05	14.2

## Data Exploration

We will begin the analysis by examining the counts and proportions each cultivar type.

```
wine %>%
  group_by(cultivar) %>%
```

```

summarize(
  N = n(),
  p = n() / nrow(wine)
)

```

cultivar	N	p
Barbera	46	0.263
Dolcetto	58	0.331
Nebbiolo	71	0.406

We will also examine scatterplots of between each of the predictors. Since the outcome is categorical, plotting the cultivar type (on  $Y$ ) against each predictor (on  $X$ ) will lead to some of the same problems we had when we created scatterplots in the logistic regression notes. Instead, we will plot each predictor against every other predictor and can color the observations according to wine cultivar. This will help us evaluate whether the characteristics are predictive of cultivar type.

Below, I use the `ggpairs()` function from the **GGally** package to plot the three scatterplots simultaneously.

```

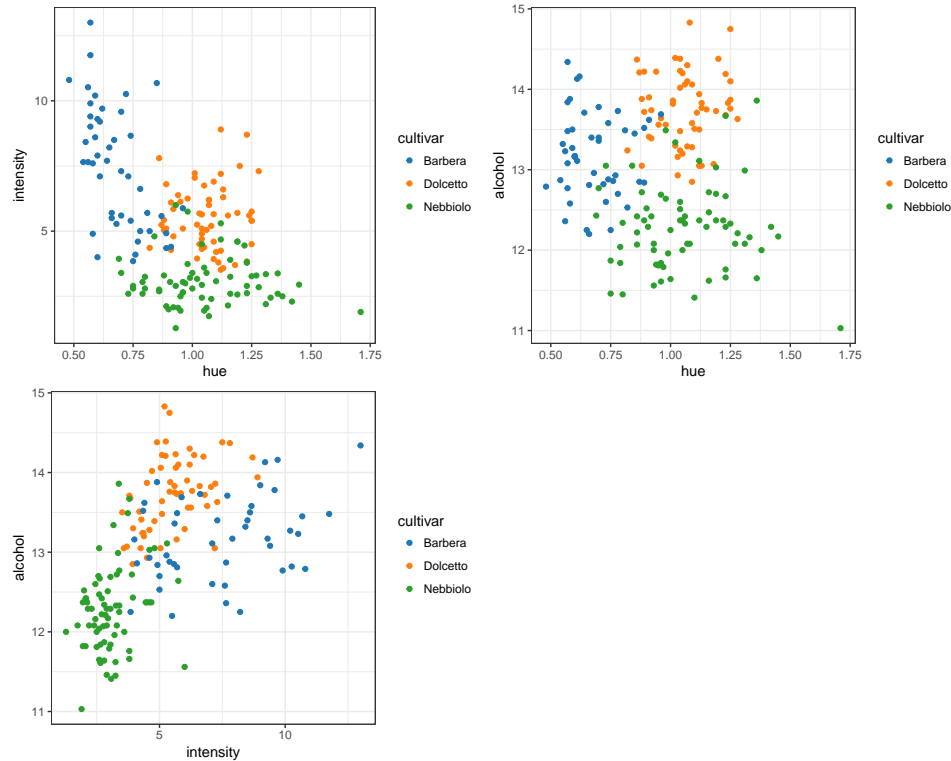
p1 = ggplot(data = wine, aes(x = hue, y = intensity, color = cultivar)) +
  geom_point() +
  theme_bw() +
  ggsci::scale_color_d3()

p2 = ggplot(data = wine, aes(x = hue, y = alcohol, color = cultivar)) +
  geom_point() +
  theme_bw() +
  ggsci::scale_color_d3()

p3 = ggplot(data = wine, aes(x = intensity, y = alcohol, color = cultivar)) +
  geom_point() +
  theme_bw() +
  ggsci::scale_color_d3()

gridExtra::grid.arrange(p1, p2, p3, ncol = 2)

```



Based on these plots we can see:

- Wines based on the Barbera cultivar generally have lower values of hue than wines based on the other two cultivars.
- Wines based on the Barbera cultivar generally have higher values of color intensity than wines based on the other two cultivars.
- Wines based on the Dolcetto cultivar generally have higher values of alcohol concentration than wines based on the Nebbiolo cultivar.

We also note that the predictors are related to one another. We can confirm this by examining the correlation matrix of the predictors.

```
wine %>%
  select(hue, intensity, alcohol) %>%
  correlate() %>%
  shave()
```

rowname	hue	intensity	alcohol
hue	NA	NA	NA
intensity	-0.520	NA	NA
alcohol	-0.088	0.557	NA

## Multinomial Experiments

Consider a random variable  $Y_i$  that may take one of several categorical (discrete) values, which we index as  $1, 2, \dots, K$ . We call this a *multinomial experiment*. In our wine data example, the outcome variable, cultivar, can be considered a multinomial experiment: wine cultivar take one of several categorical values (“Barbera”, “Dolcetto”, and “Nebbiolo”) which we index as 1, 2 and 3.

We can denote the probability that the  $i$ th response falls in the  $k$ th category as,

$$\Pr(Y_i = k) = p_k,$$

and, assuming that the response categories are mutually exclusive and exhaustive,

$$\sum_{i=1}^K p_i = 1.$$

In the wine example:

- $p_1$  is the probability that the  $i$ th wine is the Barbera cultivar,
- $p_2$  is the probability that the  $i$ th wine is the Dolcetto cultivar, and
- $p_3$  is the probability that the  $i$ th wine is the Nebbiolo cultivar.

And, using the probabilities (proportions) we computed earlier,

$$\sum_{i=1}^3 p_i = p_1 + p_2 + p_3 = .263 + .331 + .406 = 1$$

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 that  $y_1$  occurs  $n_1$  times,  $y_2$  occurs  $n_2$  times,  $\dots$ , and  $y_k$  occurs  $n_k$  times is,

$$\Pr(y_1, y_2, \dots, y_n) = \binom{N}{n_k} \times p_1^{n_1} \times p_2^{n_2} \times \dots \times p_k^{n_k},$$

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

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

where  $!$  is the factorial operator.

In our example, say we wanted to compute the probability of seeing: 3 Barbera cultivars, 4 Dolcetto cultivars, and 3 Nebbiolo cultivars in 10 sampled wines. The probability is computed as,

$$\frac{10!}{(3!)(4!)(3!)} \times .263^3 \times .331^4 \times .406^3 = .061$$

We can compute this probability using R

```
factorial(10) / (factorial(3) * factorial(4) * factorial(3)) * .263^3 * .331^4 * .406^3
```

```
## [1] 0.0614
```

## Modeling a Multinomial Outcome

When we had a categorical predictor with multiple categories, we needed to create  $K - 1$  dummy variables (where  $K$  is the number of categories) to include in the model. When the outcome has multiple categories we need to do the same thing, we just use them as the outcome rather than as a predictor. For example in the wine data, we will create the dummy variables `barbera` and `dolcetto` (the reference cultivar is `Nebbilo`).

```
wine %>%
  mutate(
    barbera = if_else(cultivar == "Barbera", 1, 0),
    dolcetto = if_else(cultivar == "Dolcetto", 1, 0)
  ) %>%
  head(.)
```

cultivar	intensity	hue	alcohol	barbera	dolcetto
Dolcetto	5.64	1.04	14.2	0	1
Dolcetto	4.38	1.05	13.2	0	1
Dolcetto	5.68	1.03	13.2	0	1
Dolcetto	7.80	0.86	14.4	0	1
Dolcetto	4.32	1.04	13.2	0	1
Dolcetto	6.75	1.05	14.2	0	1

The idea of multinomial regression is that we fit multiple logistic regression models to the data. In this case we would fit one logistic regression model to predict variation in the `barbera` variable, and another to predict variation in the `dolcetto` variable. This is carried out automatically, and simultaneously, when we actually fit the model using statistical software.

The multinomial logit model computes the log-odds for all other categories relative to the reference category, and then expresses the log-odds as a linear function of any predictors. The general form for the multinomial model is

$$\ln \left( \frac{p_k}{p_0} \right) = \beta_0 + \beta_1(X_{1i}) + \beta_2(X_{2i}) + \dots + \beta_j(X_{ji}) + \epsilon_i$$

where  $p_k$  is the probability of the  $k$ th category and  $p_0$  is the probability of the reference category. The only assumption for the multinomial logit model is the assumption of independence among the outcome categories. This assumption states that an observation's membership in one category is not related to any other observation's membership in another category. In software, the parameters from these models are usually estimated using Maximum Likelihood.

## Fitting the Multinomial Model using R

To fit a multinomial model in R, we will use the `multinom()` function from the **nnet** library. To illustrate this, we will fit a multinomial logit model that includes color intensity as a predictor of cultivar type. The first thing we need to do is set the reference group. To do this we first coerce cultivar type into a factor and then re-level it. Here we set the reference group to Nebbiolo.

```
# Set Nebbiolo as reference group
wine$cultivar = relevel(factor(wine$cultivar), ref = "Nebbiolo")

# Check the ordering of levels; Nebbiolo should now be first
levels(wine$cultivar)
```

```
## [1] "Nebbiolo" "Barbera" "Dolcetto"
```

```
model.1 = multinom(cultivar ~ 1 + intensity, data = wine)
```

```
## # weights: 9 (4 variable)
## initial value 192.257151
## iter 10 value 103.382540
## final value 103.366707
## converged
```

```
summary(model.1)
```

```
## Call:
## multinom(formula = cultivar ~ 1 + intensity, data = wine)
##
## Coefficients:
##          (Intercept) intensity
## Barbera          -12.41      2.60
## Dolcetto          -8.61      2.04
##
## Std. Errors:
##          (Intercept) intensity
## Barbera           1.62    0.362
## Dolcetto           1.40    0.338
##
## Residual Deviance: 207
## AIC: 215
```

The output splits up the estimated coefficients and their estimated standard errors. In the coefficient part of the output, note that there are two intercept values and two intensity values. These are associated with the Barbera and Dolcetto cultivars. Writing the two fitted equations,

$$\ln \left[ \frac{\Pr(\text{Barbera})}{\Pr(\text{Nebbiolo})} \right] = -12.41 + 2.60(\text{Color Intensity}_i)$$

and

$$\ln \left[ \frac{\Pr(\text{Dolcetto})}{\Pr(\text{Nebbiolo})} \right] = -8.61 + 2.04(\text{Color Intensity}_i)$$

We interpret the coefficients exactly the same as we interpret logistic regression coefficients, which means we can interpret the coefficients in the scale of logits, odds, or probabilities. Transforming the coefficients to the odds scale,

```
exp(coef(model.1))
```

	(Intercept)	intensity
Barbera	0	13.53
Dolcetto	0	7.72

we can interpret the values from the first fitted equation as,

- For wines that have a color intensity value of 0, the probability of being a Barbera cultivar is 0.000004 times that of being a Nebbiolo cultivar (most likely a Nebbiolo cultivar rather than a Barbera cultivar).
- Each one-unit difference in color intensity increases the odds of a wine cultivar being a Barbera cultivar by 13.5 times (relative to being a Nebbiolo)

### 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. The probability of  $Y_i$  being a member of one of the non-reference groups is

$$\Pr(Y_i) = \frac{e^{\beta_0 + \beta_1(X_i)}}{1 + \sum_1^{k-1} e^{\beta_0 + \beta_1(X_i)}}$$

The probability of  $Y_i$  being a member of one of the reference groups is

$$\Pr(Y_i) = \frac{1}{1 + \sum_1^{k-1} e^{\beta_0 + \beta_1(X_i)}}$$

In other words, first, you need to compute the log-odds for each of the non-reference categories and exponentiate them. In our example, these are computed as:

```
# Odds of Barbera
exp(-12.41 + 2.60 * 2)
```

```
## [1] 0.000739
```

```
# Odds of Dolcetto
exp(-8.61 + 2.04 * 2)
```

```
## [1] 0.0108
```

Then we compute the probabilities as:

$$\Pr(\text{Barbera}) = \frac{0.000739}{1 + 0.000739 + 0.0108} = .000731$$

$$\Pr(\text{Dolcetto}) = \frac{0.0108}{1 + 0.000739 + 0.0108} = .0107$$

$$\Pr(\text{Nebbiolo}) = \frac{1}{1 + 0.000739 + 0.0108} = .989$$

This suggests that, given our model, a wine with a color intensity value of 2 is most likely a Nebbiolo wine, and is least likely to be a Barbera wine. We can also compute these using the `predict()` function. To obtain the probabilities, we set the argument `type="probs"`.

```
## Create new data frame
my_wine = data.frame(
  intensity = 2
)

## Use model to predict probabilities
predict(model.1 , newdata = my_wine, type = "probs")

## Nebbiolo Barbera Dolcetto
## 0.98847 0.00074 0.01079
```

### Plotting the Probabilities for Different Values of Intensity

It is easier to represent the probabilities in a plot that shows the three wine cultivars. To do so, we first create a data frame that contains a sequence of color intensity values. Then we use the `predict()` function to produce the fitted values.

```
# Set up data frame
my_wine = expand.grid(
  intensity = seq(from = 1, to = 13, by = 1)
)

# Predict probabilities
prob = predict(model.1 , newdata = my_wine, type = "probs")
prob
```

Nebbiolo	Barbera	Dolcetto
0.999	0.000	0.001
0.988	0.001	0.011
0.914	0.009	0.077
0.559	0.077	0.364
0.127	0.235	0.638
0.015	0.386	0.599
0.002	0.530	0.469
0.000	0.664	0.336
0.000	0.776	0.224
0.000	0.858	0.142
0.000	0.914	0.086



Nebbiolo	Barbera	Dolcetto
0.000	0.949	0.051
0.000	0.970	0.030

The probabilities from the `predict()` function are produced in three separate columns. To plot them using `ggplot()`, we need to put them in a single column and set up another variable that indicates the wine cultivar. To do this we can use the `gather()` function, but we first need to turn the `prob` object into a data frame. We also bind the probabilities to the data we predicted from prior to tidying it.

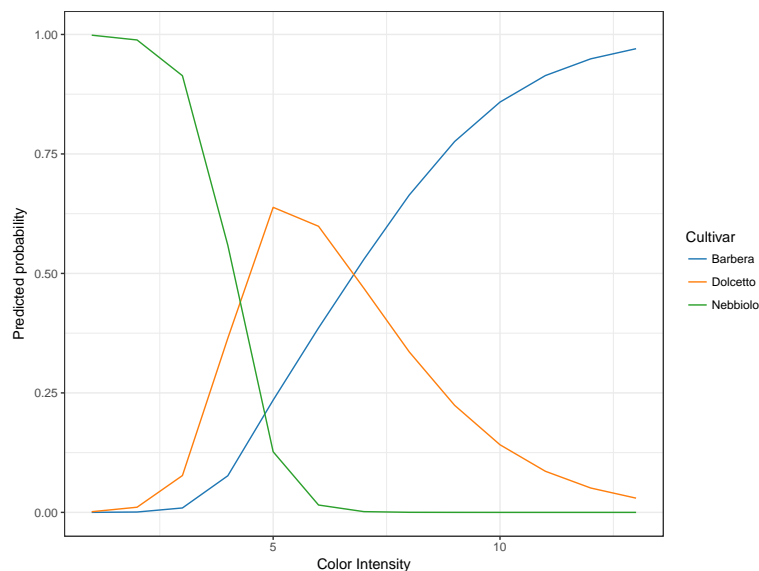
```
plot_data = data.frame(prob) %>%
  cbind(., my_wine) %>%
  gather(key = "cultivar", value = "probability", Nebbiolo:Dolcetto)

head(plot_data)
```

intensity	cultivar	probability
1	Nebbiolo	0.999
2	Nebbiolo	0.988
3	Nebbiolo	0.914
4	Nebbiolo	0.559
5	Nebbiolo	0.127
6	Nebbiolo	0.015

Now we can plot the results.

```
ggplot(data = plot_data, aes(x = intensity , y = probability , color = cultivar)) +
  geom_line() +
  ggsci::scale_color_d3(name = "Cultivar") +
  xlab("Color Intensity") +
  ylab("Predicted probability") +
  theme_bw()
```



We can interpret this by looking at the curve with the highest probability for a given color intensity value. For example, wines with a color intensity value less than approximately 4.7 are most likely Nebbiolo cultivars. Those with color intensity values between 4.7 and 7 are most probably Dolcetto cultivars. And those with color intensity values over 7 are most probably Nebbiolos.

## Fit a Model with Multiple Predictors

Now we will fit a model using color intensity, hue and alcohol concentration to predict variation in wine cultivar. We will also test whether the additional parameters add to the explanation in variation by using a likelihood ratio test to compare Model 1 and Model 2.

```
# Fit model with all three predictors
model.2 = multinom(cultivar ~ intensity + hue + alcohol, data = wine)
```

```
## # weights: 15 (8 variable)
## initial value 192.257151
## iter 10 value 48.540633
## iter 20 value 43.193581
## iter 30 value 41.195123
## iter 40 value 40.441899
## iter 50 value 40.355854
## iter 60 value 40.337716
## iter 70 value 40.321593
## iter 80 value 40.318633
## iter 90 value 40.317170
## iter 100 value 40.316525
## final value 40.316525
## stopped after 100 iterations
```

```
# LRT to compare Model 1 and Model 2
anova(model.1, model.2)
```

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
1 + intensity	346	206.7		NA	NA	NA
intensity + hue + alcohol	342	80.6	1 vs 2	4	126	0

Fitting the multinomial logit model with the three predictors leads to a deviance of 80.6 on 342 df. (This is lower than the deviance in Model 1 which was 207, but more complex by 4 parameters; two hue parameters and two alcohol concentration parameters.) The  $p$ -value of the LRT suggests that we should adopt the more complex model (Model 2).

Next we will examine the results from fitting Model 2.

```
# Obtain results from fitting Model 2
summary(model.2)
```

```
## Call:
## multinom(formula = cultivar ~ intensity + hue + alcohol, data = wine)
##
## Coefficients:
```

```
##           (Intercept) intensity    hue alcohol
## Barbera      -22.8      2.12 -19.234    2.31
## Dolcetto     -62.5      1.60  -0.316    4.28
##
## Std. Errors:
##           (Intercept) intensity    hue alcohol
## Barbera       12.6      0.537  4.62    0.996
## Dolcetto      13.2      0.515  2.91    0.989
##
## Residual Deviance: 80.6
## AIC: 96.6
```

Writing the fitted equations,

$$\ln \left[ \frac{\Pr(\text{Barbera})}{\Pr(\text{Nebbiolo})} \right] = -22.8 + 2.12(\text{Color Intensity}_i) - 19.23(\text{Hue}_i) + 2.31(\text{Alcohol Concentration}_i)$$

and

$$\ln \left[ \frac{\Pr(\text{Dolcetto})}{\Pr(\text{Nebbiolo})} \right] = -62.5 + 1.60(\text{Color Intensity}_i) - 0.32(\text{Hue}_i) + 4.28(\text{Alcohol Concentration}_i)$$

Based on the output, the coefficients suggest that:

- Higher values of color intensity increases the likelihood that the wine will be classified as Barbera as opposed to Nebbiolo, controlling for differences in hue and alcohol concentration.
- Higher values of hue (darker wines) decreases the likelihood that the wine will be classified as Barbera as opposed to Nebbiolo, controlling for differences in color intensity and alcohol concentration.
- Higher values of alcohol concentration increases the likelihood that the wine will be classified as Barbera as opposed to Nebbiolo, controlling for differences in color intensity and hue.

We see the effects in the same direction for the likelihood of Dolcetto relative to Nebbiolo, but the effects of color intensity and hue are less pronounced, while the effect of alcohol concentration is more pronounced.

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

## Plotting the Predicted Probabilities

```
## Create data frame of values to use in the model
my_wine = expand.grid(
  intensity = 1:13,
  hue = c(0.79, 1.12),
  alcohol = c(11, 15)
)

## Fit the model to the new data, add intensity and alcohol values, and tidy it
plot_data = data.frame(
  predict(model.2, newdata = my_wine, type = "probs")
) %>%
cbind(., my_wine) %>%
gather(key = "cultivar", value = "probability", Nebbiolo:Dolcetto)

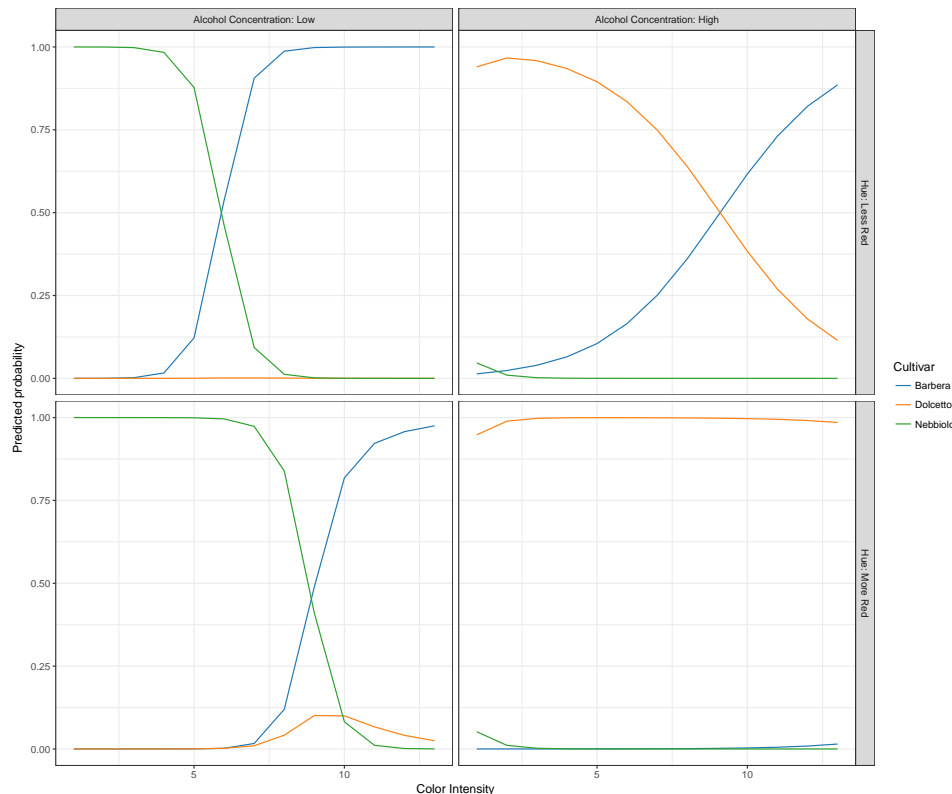
head(plot_data)
```

intensity	hue	alcohol	cultivar	probability
1	0.79	11	Nebbiolo	1.000
2	0.79	11	Nebbiolo	1.000
3	0.79	11	Nebbiolo	0.998
4	0.79	11	Nebbiolo	0.984
5	0.79	11	Nebbiolo	0.878
6	0.79	11	Nebbiolo	0.462

```
# Turn alcohol into a factor for better plotting
plot_data$alcohol = factor(plot_data$alcohol,
  levels = c(11, 15),
  labels = c("Alcohol Concentration: Low",
    "Alcohol Concentration: High")
)

plot_data$hue = factor(plot_data$hue,
  levels = c(0.79, 1.12),
  labels = c("Hue: Less Red",
    "Hue: More Red")
)

ggplot(data = plot_data, aes(x = intensity, y = probability, color = cultivar)) +
  geom_line() +
  ggsci::scale_color_d3(name = "Cultivar") +
  xlab("Color Intensity") +
  ylab("Predicted probability") +
  theme_bw() +
  facet_grid(hue ~ alcohol)
```



Looking at the plot,

- If the wine has a high alcohol concentration, it is most probably made from a Dolcetto cultivar. Unless that wine also has a low hue value and high color intensity, in which case it is likely made from a Barbera cultivar.
- If the wine has a low alcohol concentration, it is most probably made from a Nebbiolo cultivar if that wine has a low color intensity value or made from a Barbera cultivar if it has a high color intensity value.

## Table of Regression Results

We can use the `texreg()`, `htmlreg()`, or `screenreg()` functions to output the coefficients, standard errors, and coefficient-level statistical significance for the two models. The `pos='H'` argument will force the table to be directly where you put the syntax rather than trying to float the table. (You need to load the latex package `float` in the YAML part of the document to use this; see the RMD document.)

```
texreg(list(model.1, model.2), pos = 'H',
        include.nobs = FALSE, include.loglik = FALSE,
        caption = "Results from Fitting a Taxonomy of Multinomial Logistic Regression Models to Predict Cultivar Type",
        )
```

	Barbera	Dolcetto	Barbera	Dolcetto
(Intercept)	-12.41*** (1.62)	-8.61*** (1.40)	-22.80 (12.62)	-62.50*** (13.15)
intensity	2.60*** (0.36)	2.04*** (0.34)	2.12*** (0.54)	1.60** (0.52)
hue			-19.23*** (4.62)	-0.32 (2.91)
alcohol			2.31* (1.00)	4.28*** (0.99)
AIC	214.73	214.73	96.63	96.63
BIC	227.39	227.39	121.95	121.95
Deviance	206.73	206.73	80.63	80.63

\*\*\*  $p < 0.001$ , \*\*  $p < 0.01$ , \*  $p < 0.05$

Table 10: Results from Fitting a Taxonomy of Multinomial Logistic Regression Models to Predict Cultivar Type in 175 Wines.