Model Assessment

Logistic regression – Stat 230

In this class, you will fit and assess binary logistic regression models both for their utility in describing the observed associations and making predictions. In addition, you will check the assumptions required for valid inference.

Data overview

The Boundary Waters Blowdown was a derecho (a severe windstorm) that occurred in July 1999 that caused massive damage to the Boundary Waters Canoe Area Wilderness in northeastern Minnesota, among other areas. In total, the storm caused over $100 million dollars in damage. Researchers studied the effects of this storm using an extensive ground survey of the area, determining the status (alive or dead) of more than 3600 trees.

The data may be loaded using the command

```
blowdown <- read.csv("http://aloy.rbind.io/data/blowdown.csv")
```

The variables in the data set are:

- `y`: status, 1 = dead, 0 = survived
- `d`: tree diameter, in cm
- `s`: proportion of basal area killed (a measure of local severity of the storm)
- `spp`: tree species (there are 9)

Plotting an initial model

Task 1. Fit a logistic regression model using diameter to describe the status of each tree. Report the fitted equation for the linear predictor.

Task 2. Interpret the diameter coefficient in context.

Task 3. To create plots of a fitted logistic regression model, I recommend constructing an effect plot. (Remember those from linear regression?). To create an effect plot for this model, load the {effects} package. Then run the following command:

```
Effect("d", mod1) |>
  plot(type = "response", xlab = "diameter (cm)", ylab = "Prob. of death")
```

Binned residual plots

Now that you have an initial fitted model, we need to see whether it adequately describes the observed data.

To construct a binned residual plot for a fitted binary logistic regression model, we'll use the `binnedplot()` command from the {arm} package. Building this plot requires two steps: first we will create a data frame with the necessary data, then we will run a plotting function. To create the data frame, we'll use the `augment()` command from the {broom} package, so be sure to load broom in your setup code chunk!

```
# Create a data frame with response residuals and x-axis variables
mod1_aug <- augment(mod1, blowdown) |>
  mutate(.resp.resid = resid(mod1, type = "response"))

# Make the plot
arm::binnedplot(mod1_aug$d, mod1_aug$.resp.resid, xlab = "diameter", col.int = NULL)
```

I don't recommend loading the {arm} package since it can cause issues with other packages, so we use `arm::binnedplot()` to call the function without loading the package.

Task 4. Construct the binned residual plot and comment on what it indicates about the adequacy of the logistic regression model.

Task 5. In the last task, you should have found indications of a model violation. To remedy this issue, let's apply a natural log transformation to the diameter. Fit a transformed logistic regression model (just like in linear regression) and construct a new binned residual plot. Does the transformation appear to have remedied the issue?

Task 6. You likely found that your binned residual plot still isn't fully "satisfactory" yet. Currently, we have a very simple model, so let's consider a more complex model where we add in severity as a predictor. After fitting this model, create a new augmented data frame and create binned residual plots for the fitted values and each predictor variable. Describe any potential issues you see.

Task 7. Now, add species as a predictor. After fitting this model, create a new augmented data frame and create binned residual plots for the fitted values and each predictor variable. In addition, calculate the mean of the response residuals for each species. (Look at your EDA notes to recall how we calculated means by groups. *Hint:* You can use the `mean()` function from {mosaic}.) Describe any potential issues you see.

Measures of association

Once you have verified that your model is valid, you might wish to determine how well the model describes the observed data. In linear regression we used R^2 as such measure. In logistic regression, Kuiper and Sklar recommend the use of a few association metrics to determine the degree of association between the model's predictions and the observed data. A better model should produce predictions that are more highly associated with the observed data.

In order to obtain the three measures of association described in the textbook, we need to use a different function to fit our logistic regression model. We'll use the `lrm()` function in the {rms}

package. For example, we can refit the regression model from the previous task using the below code:

```
lrm(y ~ log(d) + s + spp, data = blowdown)
```

Task 8. Run the above `lrm()` code and report the three measures of association described in the textbook and the video.

Task 9. Next, let's consider an expanded model with interactions between the three predictors that are included in the model thus far. Run the below code chunk to fit this model

```
model4_glm <- glm(y ~ (log(d) + s + spp)^3, data = blowdown)
```

Task 10. Use `anova()` to run a drop-in deviance test comparing the model with and without the interaction. (Make sure you are using the two models fit using the `glm()` command.) What do you find?

Task 11. The data set has 3666 rows, so we have enough observations to find statistically discernible "improvements" to the model that are minimal boosts to the association. Remember that we prefer simpler models, so we would prefer the no interaction if the predictions and the observed data had roughly the same level of association as the more complex model. Refit the interaction model using `lrm()` and report the three measures of association. Do the interactions appear to substantially improve the association between the predictions and the observed data?

Visualizing the fitted model

Once you settle on a final model, it can be useful to plot the fitted model to communicate your findings. The below code creates effects plots for the interaction model using three difference values for severity. What do you learn from this plot?

```
# Specify interest in the main effects and levels for s to display
eff <- Effect(c("d", "s", "spp"), mod = model4_glm, xlevels = list(s=c(.2, .35, .55 )))

# Create the plot
plot(eff, grid=TRUE, x.var = "d", multiline = TRUE,
     xlab="Diameter", main="", lines=c(1, 3, 2),
     ylab="Prob. of blow down", type = "response", rug=FALSE, cex = .45)
```

Adapt the above code to use the no interaction model. How does the plot change?