Luke Watson

958

**Generalized linear modeling**

*Part One*

To uncover how the department makes admission decisions, I investigated how the different quantitative admission criteria were weighted by the reviewers. I sought to create a model that could give insight into how each of the predictors factored into an applicant’s probability of being accepted into their program of choice. To start, I created a standard multiple regression model using each of the components of the GRE (verbal, quantitative, and analytical writing) along with GPA to predict their acceptance, which was coded as 1 (accept) or 0 (reject). The overall model was found to be significant (*F*(4, 250) = 4.92, *p* < .001). The weights, standard errors, and significance are summarized in Table 1, but no weights reached significance although they indicated a positive relationship to acceptance. Aside from GPA though, the size of each of the GRE predictor weights were very small.

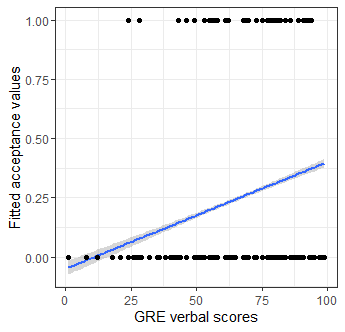
**Table 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | B | SE | t | p |
| Intercept | -.472 | .262 | -1.80 | .073 |
| GREQ | .003 | .001 | 1.84 | .066 |
| GREV | .003 | .002 | 1.68 | .094 |
| GREAW | .001 | .001 | .59 | .554 |
| GPA | .106 | .076 | 1.39 | .166 |

*Note.* Estimates and errors represent values from 0 to 1 where 0 is rejection and 1 is acceptance. Positive values indicate that higher scores on those terms result in a higher rate of acceptance. No terms reached significance threshold of p < .05.

However, looking under the hood of this model, I noticed many discrepancies. Using the check\_model() function from the performance package in R, I noticed huge deviations from the assumptions of the standard regression model I ran including violations to linearity, homogeneity of error variance, and normality of errors. The only issue that did not seem to crop up was multicollinearity, which is often a culprit for reduced estimate sizes, but not in this case. Likely this issue and others are the result of this outcome variable being treated as numerical rather than binomial (accept/reject). This is a deeply flawed model that struggles to find anything of substance to get at my original question. The model fit compared to the original data across scores of GRE verbal are compared in Figure 1 below.

**Figure 1**

****

*Figure 1* – Representation of issues of modelling binomial outcomes with linear regression. The model fit (line in blue) does not fit the data well and even predicts negative acceptances for 5th percentile GRE verbal scores and below! Error ribbon represents +/- SE.

I compared the previous model to a null model including no predictors. Using AIC, I compared the fit of the models to each other and the null model (AIC = 293) fit worse than the previous model (AIC = 282).

Finally, I conducted a posterior predictive check on this original model and found that the observed data deviated drastically from the normally distributed data the model predicted. The distribution of observed data had two peaks, one larger around 0 and one about 1/3 the size around 1. This hammers home the point that this outcome should not be modelled with a standard linear regression.

*Part Two*

Moving beyond the linear regression, I decided to instead base the model off of the binomial outcome distribution with a logistic regression. I used the same predictors as before and ran the model. The model results are summarized in Table 2 below, and like before, there were no significant estimates. Again, the general direction of the estimates indicated a positive relationship, with higher GRE scores and GPA resulting in a higher probability of acceptance.

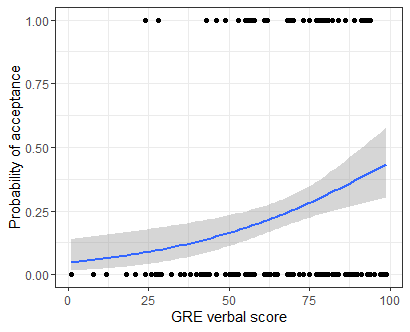
**Table 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | B | SE | t | p |
| Intercept | -6.04 | 1.81 | -3.33 | <.001 |
| GREQ | .017 | .01 | 1.80 | .071 |
| GREV | .016 | .009 | 1.85 | .064 |
| GREAW | .005 | .007 | .73 | .462 |
| GPA | .720 | .503 | 1.43 | .152 |

*Note.* Model results for the logistic regression using GRE section scores and GPA to predict probability of being accepted into the program. Estimates and standard errors are the log odds of being accepted compared to being rejected. Positive values indicate higher odds of acceptance and negative values indicate lower odds.

The results of the model are displayed in Figure 1 below.

**Figure 2**



*Figure 2* – Plotted model fit over the original binomial data. The blue line is the model fit created by stat\_smooth(). This model does a much better job at predicting the binomial data by avoiding making non-sensical predictions about negative probabilities of acceptance. Error ribbons represent +/- SE.

The logistic model provides an accurate estimation of the constraints of binomial outcome data. It does not predict past floors in the outcome and does not violate assumptions like the previous linear regression.

*Part Three*

To explore this data further, I wanted to see if these quantitative measures of applications were differentially weighted by the applied program within the department. So, I ran another logistic regression with the same predictors and outcome as the previous models and added in which program they applied to as a dummy-coded categorical predictor. The results of the model are summarized in Table 3, and there were significant estimates that emerged from this model. The estimate for an industrial-organizational application (*B* = -1.25, *SE* = .494, *p* = .011), social/personality application (*B* = -1.56, *SE* = .5, *p* = .002), GRE quantitative score (*B* = .021, *SE* = .009, *p* = .027), and GPA (*B* = 1.04, *SE* = .525, *p* = .048) were all significant. Both the I/O and S/P applicants experienced lower odds of acceptance in their respective programs, indicating that these may be more competitive spots for applicants compared to cognitive and behavioral neuroscience (Figure 3).

**Table 3**

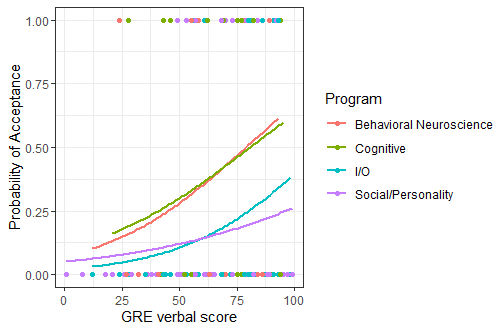
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | B | SE | t | p |
| Intercept | -6.56 | 1.94 | -3.39 | <.001 |
| GREQ | .021 | .009 | 2.22 | .027 |
| GREV | .013 | .01 | 1.29 | .197 |
| GREAW | .01 | .01 | 1.29 | .197 |
| GPA | 1.04 | .525 | 1.98 | .048 |
| Cognitive | .19 | .513 | .37 | .711 |
| I/O | -1.25 | .494 | -2.53 | .011 |
| S/P | -1.56 | .5 | -3.13 | .002 |

*Note*. Model results summarized for the logistic regression predicting acceptance including program as a categorical predictor. The estimates and standard errors are log odds of acceptance.

Comparing this model to the previous logistic regression without program as a predictor, the AIC was much lower (AIC = 255) than the previous model (AIC = 270). I also compared the BIC values of the two models and found a similar result albeit the model fit discrepancy was reduced (BIC without program = 287, BIC with program = 283) likely due to the added complexity of mixing in a categorical predictor to this model.

A visual representation of the relationship between program and GRE verbal scores and the original data can be found in Figure 3.

**Figure 3**

****

*Figure 3* – Logistic regression results plotted over the original decision binomial outcome data by program. As expected from the model results table, there is a significant difference between programs with cognitive and BN students more likely to be accepted than I/O and S/P regardless of GRE verbal scores.

# intercept: -6.56

# cog: .19

# i/o: -1.25

# s/p: -1.56

# grev: .013

# greq: .021

# greaw: .01

# gpa: 1.04

#

# b/c this is logistic, have to apply that function to result

#

# E(decision(accept)) = -6.56 + .19\*"cog" - 1.25\*"i/o" - 1.56\*"s/p" + .013\*grev + .021\*greq + .01\*greaw + 1.04\*gpa

#

# predict acceptance for each program w/ 50th percentile score on GRE, 3.5 gpa, **dummy coded program**

#

# cog:

cog <- -6.56 + .19\*1 - 1.25\*0 - 1.56\*0 + .013\*50 + .021\*50 + .01\*50 + 1.04\*3.5

# i/o:

io <- -6.56 + .19\*0 - 1.25\*1 - 1.56\*0 + .013\*50 + .021\*50 + .01\*50 + 1.04\*3.5

# s/p:

sp <- -6.56 + .19\*0 - 1.25\*0 - 1.56\*1 + .013\*50 + .021\*50 + .01\*50 + 1.04\*3.5

# bn

bn <- -6.56 + .19\*0 - 1.25\*0 - 1.56\*0 + .013\*50 + .021\*50 + .01\*50 + 1.04\*3.5

# now need to use logistic to turn results into proper output log odds

# use inv.logit() from boot library to do this calculation

programs <- c(cog, io, sp, bn)

# apply logistic to program vector

map(programs, inv.logit)

# cog: .37, io: .12, sp: .09, bn: .33

# probabilities for acceptance by program

*Part Four*

For the next model, I wanted to see what factors in this experiment would predict stop responses during a go-no-go task, particularly the ratio of stop to go trials and the inter-trial interval (ITI). I started again with an exploratory linear model with ratio, ratio quadratic, ITI, and the interaction between ratio and ITI as predictors.

(*Part Six*)

I used this initial model to investigate whether this variable of summed stop responses needed to be transformed as an exercise. I used the check\_model() command and found a couple issues: first, the posterior predicted model distributions predicted values below zero and had a normal shape (as expected for this model), but the observed data were peaked around 0 and did not fall below that value. Second, there were severe multicollinearity issues for the quadratic, interaction, and even the ratio term. Lastly, the error terms were not normally distributed and showed heterogeneity along the predicted values of the outcome.

*Part Four cont…*

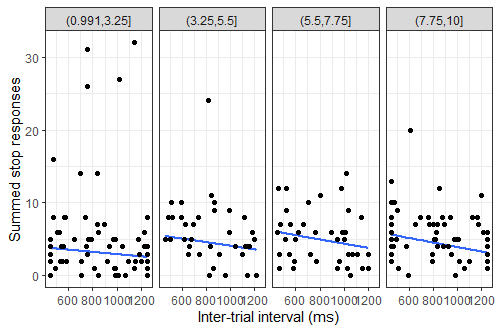
To address these issues with the linear model, I opted to log transform the outcome of summed stop responses (adding 1 to prevent issues with zero values) and center the predictors. I reran the adjusted model, and the results are summarized in Table 4. Each of the predictors aside from the interaction term were significant. For every millisecond of inter-trial interval (ITI), fewer stop responses were predicted by the model. This was the opposite for the ratio term, which an increase of ratio of stop to go trials predicted an increase of stop responses in that trial (Figure 4).

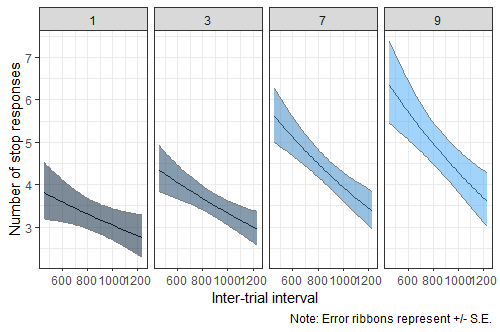
Upon second inspection of the assumptions, this model addressed most issues including multicollinearity, normality of errors, and homogeneity of errors. The only issue that remained was the posterior predictive check revealed a bimodal distribution of observed data with a small peak still appearing around 0. This is likely due to the fact that this is not the proper model for the type of outcome that is being measured here. To bring home this point, I plotted the relationship between different values of ratio of go to stop trials and the inter-trial interval in Figure 4. There are still a lot of values that pile around zero that will be hard to address with the linear model.

**Table 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | B | SE | t | p |
| Intercept | 4.68 | .078 | 23.2 | <.001 |
| ITI | -.0005 | .0002 | -2.57 | .011 |
| Ratio | .040 | .017 | 2.38 | .018 |
| Ratio\*Ratio | -.015 | .006 | -2.29 | .023 |
| Ratio\*ITI | -.0000 | .0001 | -.69 | .492 |

*Note.* Model results using the above predictors to predict log-transformed summed stop responses in a go-no-go task. The estimates and standard errors have been back transformed back into summed stop responses.

**Figure 4**



*Figure 4* – Two plots of four panels representing various continuous or sliced breaks along ratio by inter-trial interval to predict the number of stop responses. The upper plot includes the original data with model fits along continuous bins of breaks. The lower plot uses specific breaks of ratio (1, 3, 7, 9) and includes error ribbons to model uncertainty.

*Part Five*

In order to properly model the sum of stop responses (count data), I need to use a Poisson regression. So, using the same predictors and outcome as the previous model, I ran the Poisson regression, and the results are summarized in Table 5 below. The main difference between this and the previous model was the individual weight ratio fell out of significance (*B* = .014, *SE* = .019, *p* = .462), while the other predictors and their weights and errors remained relatively unchanged. Before I continued down this path, I needed to check my assumptions for this model. After using the check\_model() command, I noticed two issues: one with the posterior predictive check showing an inflation of zeroes for the observed data compared to the predicted Poisson distribution, and two, there appeared to be an issue with the dispersion. I conducted a follow-up overdispersion test with the check\_overdispersion() command from the performance library and found significant overdispersion (ratio = 4.64, *p* < .001).

So, instead of continuing with this model, I created another model using the same predictor and outcome, but run with a negative binomial distribution, which has been recommended to deal with overdispersed Poisson models. The results of the model are summarized in Table 6. This model outperformed the previous model in terms of fit and overdispersion. The main difference is that the interaction I got the general direction of before has now been found to be significant (*B* = -.0001, *SE* = .0001, *p* = .029). The weight is small like the weight for the ITI predictor, but it is likely the result of the scale units being so small (ms). When plotted over 100s of milliseconds, the relationship becomes clear (see Figure 5). The posterior predicted negative binomial distribution matched the observed data much better than before, and a test of overdispersion indicated there was none detected (ratio = 1.23, *p* = .168). AIC and BIC values for each of the models are compared in Table 6, with this model vastly outperforming the previous model.

To visualize the relationship between the predictors and outcome, I

**Table 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | B | SE | t | p |
| Intercept | 1.81 | .084 | 21.5 | <.001 |
| ITI | -.0004 | .0001 | -3.62 | <.001 |
| Ratio | .012 | .011 | 1.14 | .256 |
| Ratio\*Ratio | -.015 | .004 | -3.80 | <.001 |
| Ratio\*ITI | -.0001 | .0000 | -1.83 | .068 |

*Note.* Model results for the standard Poisson regression ran using inter-trial interval, ratio, ratio quadratic, and the interaction between ITI and ratio to predict the number of stop responses. Weights and errors represent summed stop responses.

**Table 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | B | SE | t | p |
| Intercept | 1.81 | .084 | 21.5 | <.001 |
| ITI | -.0004 | .0002 | -1.99 | .047 |
| Ratio | .014 | .019 | .735 | .462 |
| Ratio\*Ratio | -.016 | .007 | -2.19 | .029 |
| Ratio\*ITI | -.0001 | .0001 | -1.06 | .029 |

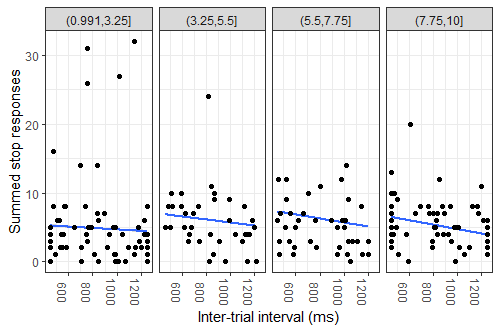
*Note.* Model results summarized for the negative binomial Poisson regression of the summed stop responses. Weights and errors should be interpreted as summed stop responses.

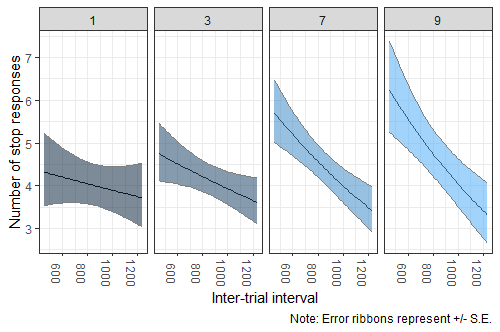
**Table 7**

|  |  |  |
| --- | --- | --- |
| Model | AIC | BIC |
| Poisson | 1407 | 1423 |
| Negative Binomial | 1136 | 1155 |

*Note.* Comparison of model fit indices between the generalized linear regressions ran with different distributional assumptions for the go-no-go data.

**Figure 5**





*Figure 5* – Model results for the negative binomial Poisson regression predicting stop responses. Same formatting as the previous figure for the original log-transformed linear model results (Figure 3). There does not appear to be much of difference between these results and the previous model, as ratio increases, there is a steeper drop in stop responses as ITI increases.

# predict stop responses

#

# E(sum\_stop) =

# 1.81 - .0004\*c.iti + .014\*c.ratio - .016\*ratio2 - .0001 \* c.iti \* c.ratio

#

pred\_sum\_stop <- 1.81 -

.0004\*(1000 - mean(go\_no\_go\_summed\_resp$iti)) +

.014\*(3 - mean(go\_no\_go\_summed\_resp$ratio)) -

.016\*(3 - mean(go\_no\_go\_summed\_resp$ratio))^2 -

.0001\*(1000 - mean(go\_no\_go\_summed\_resp$iti))\*(3 - mean(go\_no\_go\_summed\_resp$ratio))

# stop responses = 1.66 -> 2