Logistic Regression

Andrew Zieffler
April 02, 2021

Preparation

In this set of notes, you will learn how to use logistic regression models to model dichotomous categorical outcome variables (e.g., dummy coded outcome). We will use data from the file *graduation.csv* (see the data codebook (http://zief0002.github.io/epsy-8252/codebooks/graduation.html)) to explore predictors of college graduation.

```
# Load libraries
library(AICcmodavg)
library(broom)
library(corrr)
library(tidyverse)

# Read in data
grad = read_csv(file = "~/Documents/github/epsy-8252/data/graduation.csv")

# View data
head(grad)
```

```
# A tibble: 6 x 6
  degree   act scholarship    ap firstgen nontrad
   <dbl> <dbl>       <dbl> <dbl>    <dbl>   <dbl>
1      1    21         0        0        0       0
2      1    19         0        0        0       0
3      1    27         0        0        1       0
4      1    25         0.5      0        1       0
5      0    28         0       17        1       0
6      1    21         0        0        0       1
```

In the last set of notes, we saw that using the linear probability model leads to direct violations of the linear model's assumptions. If that isn't problematic enough, it is possible to run into severe issues when we make predictions. For example, given the constant effect of X in these models it is possible to

have an X value that results in a predicted proportion that is either greater than 1 or less than 0. This is a problem since proportions are constrained to the range of [0, 1].

Since the predicted outcome in our model is the proportion of students who graduate, before we consider any alternative models, let's actually examine the empirical proportions of students who graduate at different ACT scores.

```
# Obtain the proportion of graduates for each ACT score
graduates = grad %>%
  group_by(act, degree) %>%
  summarize( N = n() ) %>%
  mutate( Prop = N / sum (N) ) %>%
  filter(degree == 1) %>%
  ungroup() #Makes the resulting tibble regular

# View data
head(graduates, 10)
```

```
# A tibble: 10 x 4
     act degree     N  Prop
   <dbl>  <dbl> <int> <dbl>
 1    11      1     1 0.25
 2    13      1     4 0.5
 3    14      1     6 0.5
 4    15      1    13 0.448
 5    16      1     6 0.24
 6    17      1    18 0.429
 7    18      1    26 0.531
 8    19      1    50 0.685
 9    20      1    74 0.679
10    21      1    97 0.678
```

```
# Plot proportions
ggplot(data = graduates, aes(x = act, y = Prop)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  theme_light() +
  xlab("ACT score") +
  ylab("Proportion of graduates")
```

Proportion of graduates conditioned on ACT score. The loess smoother suggests that the proportion of students who graduate is a non-linear function of ACT scores.

Alternative Models to the Linear Probability Model

Many of the non-linear models that are typically used to model dichotomous outcome data are "S"-shaped models. Below is a plot of one-such "S"-shaped model.

The non-linear "S"-shaped model has many attractive features. First, the predicted Y values are bounded between 0 and 1. Furthermore, as X gets smaller, the proportion of $Y = 1$, approaches 0 at a slower rate. Similarly, as X gets larger, the proportion of $Y = 1$, approaches 1 at a slower rate. Lastly, this model curve is monotonic; smaller values of X are associated with smaller proportions of $Y = 1$ (or if the "S" were backwards, larger values of X would be associated with smaller proportions of $Y = 1$). The key is that there are no bends in the curve; it is always growing or always decreasing.

In our graduation example, the empirical data maps well to this curve. Higher ACT scores are associated with a higher proportion of students who graduate (monotonic). The effect of ACT, however, is not constant, and seems to diminish at higher ACT scores. Lastly, we want to bound the proportion at every ACT score to lie between 0 and 1.

How do we fit such an "S"-shaped curve? We apply a transformation function, call it Λ (Lambda), to the predicted values. Mathematically,

$$\Lambda(\pi_i) = \Lambda\Big[\beta_0 + \beta_1(X)\Big]$$

The specific transformation function used is any mathematical function that can fit the criteria we had before (monotonic, nonlinear, maps to [0, 1] space). There are several mathematical functions that do this. One common function that meets these specifications is the *logistic function*. Mathematically, the logistic function is

$$\Lambda(w) = \frac{1}{1 + e^{-w}}$$

where w is the value fed into the logistic function. For example, to logistically transform $w = 3$, we use

$$\begin{aligned}\Lambda(3) &= \frac{1}{1 + e^{-3}} \\ &= 0.953\end{aligned}$$

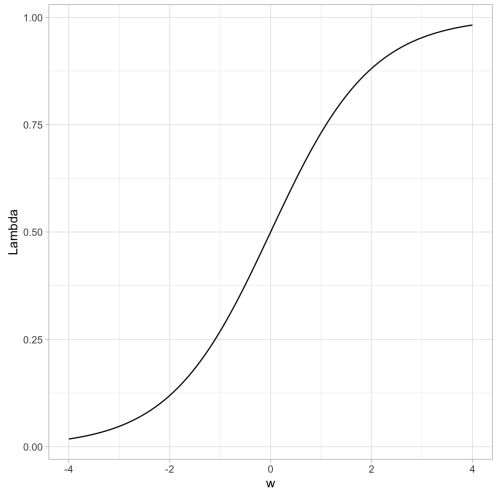
Below we show how to transform many such values using R.

```
# Create w values and transformed values
example = tibble(
  w = seq(from = -4, to = 4, by = 0.01)  # Set up values
  ) %>%
  mutate(
    Lambda = 1 / (1 + exp(-w))  # Transform using logistic function
  )

# View data
example
```

```
# A tibble: 801 x 2
       w Lambda
   <dbl>  <dbl>
 1 -4    0.0180
 2 -3.99 0.0182
 3 -3.98 0.0183
 4 -3.97 0.0185
 5 -3.96 0.0187
 6 -3.95 0.0189
 7 -3.94 0.0191
 8 -3.93 0.0193
 9 -3.92 0.0195
10 -3.91 0.0196
# … with 791 more rows
```

```
# Plot the results
ggplot(data = example, aes(x = w, y = Lambda)) +
  geom_line() +
  theme_light()
```



Plot of the logistically transformed values for a sequence of values from -4 to 4.

You can see that by using this transformation we get a monotonic "S"-shaped curve. Now try substituting a really large value of w into the function. This gives an asymptote at 1. Also substitute a really "large"" negative value in for w. This gives an asymptote at 0. So this function also bounds the output between 0 and 1.

How does this work in a regression? There, we are transforming the *predicted values*, the π_i values, which we express as a function of the predictors. Since we transform the left-side of that equation, we also need to transform the right-side.

$$\begin{aligned}\Lambda(\hat{\pi}_i) &= \Lambda\Big[\beta_0 + \beta_1(X_i)\Big] \\ &= \frac{1}{1 + e^{-\left[\beta_0+\beta_1(X_i)\right]}}\end{aligned}$$

Since we took a linear model ($\beta_0 + \beta_1(X_i)$) and applied a logistic transformation, the resulting model is the *linear logistic model* or more simply, the *logistic model*.

Re-Expressing a Logistic Transformation

The logistic model expresses the proportion of 1s (π_i) as a function of the predictor X. It can be mathematically expressed as

$$\pi_i = \frac{1}{1 + e^{-\left[\beta_0+\beta_1(X_i)\right]}}$$

We can re-express this using algebra and rules of logarithms.

$$
\begin{aligned}
\pi_i &= \frac{1}{1 + e^{-\left[\beta_0 + \beta_1(X_i)\right]}} \\
\pi_i \times (1 + e^{-\left[\beta_0 + \beta_1(X_i)\right]}) &= 1 \\
\pi_i + \pi_i(e^{-\left[\beta_0 + \beta_1(X_i)\right]}) &= 1 \\
\pi_i(e^{-\left[\beta_0 + \beta_1(X_i)\right]}) &= 1 - \pi_i \\
e^{-\left[\beta_0 + \beta_1(X_i)\right]} &= \frac{1 - \pi_i}{\pi_i} \\
e^{\left[\beta_0 + \beta_1(X_i)\right]} &= \frac{\pi_i}{1 - \pi_i} \\
\ln\left(e^{\left[\beta_0 + \beta_1(X_i)\right]}\right) &= \ln\left(\frac{\pi_i}{1 - \pi_i}\right) \\
\beta_0 + \beta_1(X_i) &= \ln\left(\frac{\pi_i}{1 - \pi_i}\right)
\end{aligned}
$$

Or,

$$
\ln\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1(X_i)
$$

The logistic model expresses the natural logarithm of $\frac{\pi_i}{1-\pi_i}$ as a linear function of X. Note that there is no error term on this model. This is because the model is for the mean structure only (the proportions), we are not modeling the actual Y_i values (i.e., the 0s and 1s) with the logistic regression model.

Log-Odds or Logits

The ratio that we are taking the logarithm of, $\frac{\pi_i}{1-\pi_i}$, is referred to as *odds*. Odds are the ratio of two probabilities. Namely the chance an event occurs (π_i) versus the chance that same event does not occur ($1 - \pi_i$). As such, it gives the *relative chance* of that event occurring. To understand this better, we will look at a couple examples.

Let's assume that the probability of getting an "A" in a course is 0.7. Then we know the probability of NOT getting an "A" in that course is 0.3. The odds of getting an "A" are then

$$
\text{Odds} = \frac{0.7}{0.3} = 2.33
$$

That the probability of getting an "A" in the class is 2.33 times as likely as NOT getting an "A" in the class. This is the relative probability of getting an "A."

As another example, Fivethirtyeight.com computed the probability that a Canadian hockey team would win the Stanley Cup in 2018 as 0.17 (https://fivethirtyeight.com/features/will-canada-end-its-stanley-cup-drought-well-its-not-impossible/). The odds of a Canadian team winning the Stanley Cup is then

$$\text{Odds} = \frac{0.17}{0.83} = 0.21$$

The probability that a Canadian team wins the Stanley Cup is 0.21 times as likely as a Canadian team NOT winning the Stanley Cup. (Odds less than 1 indicate that is is more likely for an event NOT to occur than to occur. Invert the fraction to compute how much more like the event is not to occur.)

In the logistic model, we are predicting the log-odds (also referred to as the *logit*. When we get these values, we typically transform the logits to odds by inverting the log-transformation (take e to that power.)

Binomially Distributed Errors

The logistic transformation fixed two problems: (1) the non-linearity in the conditional mean function, and (2) bounding any predicted values between 0 and 1. However, just fitting this transformation does not fix the problem of non-normality. Remember from the previous notes we learned that at each X_i there were only two potential values for Y_i; 0 or 1. Rather than use a normal (or Gaussian) distribution to model the conditional distribution of Y_i, we will use the *binomial distribution*.

The binomial distribution is a discrete probability distribution that gives the probability of obtaining exactly k successes out of n Bernoulli trials (where the result of each Bernoulli trial is true with probability π and false with probability $1 - \pi$). This is appropriate since at each value of X we can posit n Bernoulli trials, k of which are 1 (successes).

Fitting the Binomial Logistic Model in R

To fit a logistic regression model with binomial errors, we use the `glm()` function.[1] The syntax to fit the logistic model using `glm()` is:

```
glm(y ~ 1 + x, data = dataframe, family = binomial(link = "logit"))
```

The formula depicting the model and the `data=` arguments are specified in the same manner as in the `lm()` function. We also need to specify the distribution for the conditional Y_i values (binomial) and the link function (logit) via the `family=` argument.

For our example,

```
glm.1 = glm(degree ~ 1 + act, data = grad, family = binomial(link =
        "logit"))
```

The coefficient-level output of the model can be printed using `tidy()`.

```
tidy(glm.1)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   -1.61     0.283      -5.70 1.21e- 8
2 act            0.108    0.0116      9.25 2.20e-20
```

The fitted equation for the model is

$$\ln\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = -1.61 + 0.11(\text{ACT Score}_i)$$

We interpret the coefficients in the same manner as we interpret coefficients from a linear model, with the caveat that the outcome is now in log-odds (or logits):

- The predicted log-odds of graduating for students with an ACT score of 0 are -1.61.
- Each one-point difference in ACT score is associated with a difference of 0.11 in the predicted log-odds of graduating, on average.

Back-Transforming to Odds

For better interpretations, we can back-transform log-odds to odds. This is typically a better metric for interpretation of the coefficients. To back-transform to odds, we exponentiate both sides of the fitted equation and use the rules of exponents to simplify:

$$\ln\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = -1.61 + 0.11(\text{ACT Score}_i)$$

$$e^{\ln\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right)} = e^{-1.61+0.11(\text{ACT Score}_i)}$$

$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{-1.61} \times e^{0.11(\text{ACT Score}_i)}$$

When ACT score = 0, the *predicted odds of graduating* are

$$\begin{aligned}\frac{\hat{\pi}_i}{1-\hat{\pi}_i} &= e^{-1.61} \times e^{0.11(0)} \\ &= e^{-1.61} \times 1 \\ &= e^{-1.61} \\ &= 0.2\end{aligned}$$

For students with an ACT score of 0, their odds of graduating is 0.2. That is, for these students, the probability of graduating is 0.2 times that of not graduating. (It is far more likely these students will not graduate!)

To interpret the effect of ACT on the odds of graduating, we will compare the odds of graduating for students that have ACT score that differ by one point. Say ACT = 0 and ACT = 1.

We already know the predicted odds for students with ACT = 0, namely $e^{-1.61}$. For students with an ACT of 1, their predicted odds of graduating are

$$\begin{aligned}\frac{\hat{\pi}_i}{1-\hat{\pi}_i} &= e^{-1.61} \times e^{0.11(1)} \\ &= e^{-1.61} \times e^{0.11}\end{aligned}$$

These students odds of graduating are $e^{0.11}$ times greater than students with an ACT score of 0. Moreover, this increase in the odds, on average, is the case for every one-point difference in ACT score. In general,

- The predicted odds for $X = 0$ are $e^{\hat{\beta}_0}$.
- Each one-unit difference in X is associated with a $e^{\hat{\beta}_1}$ times increase (decrease) in the odds.

We can obtain these values in R by using the `coef()` function to obtain the fitted model's coefficients and then exponentiating them using the `exp()` function.

```
exp(coef(glm.1))
```

```
(Intercept)         act
  0.1997102   1.1135342
```

From these values, we interpret the coefficients in the odds metric as

- The predicted odds of graduating for students with an ACT score of 0 are 0.20.
- Each one-unit difference in ACT score is associated with 1.11 times greater odds of graduating.

To even further understand and interpret the fitted model, we can plot the predicted odds of graduating for a range of ACT scores. Recall, the general fitted equation for the logistic regression model is written as:

$$\ln\left[\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right] = \hat{\beta}_0 + \hat{\beta}_1(x_i)$$

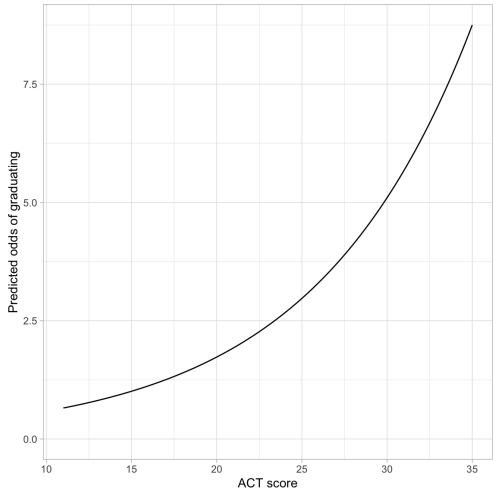
We need to predict odds rather than log-odds on the left-hand side of the equation. To do this we exponentiate both sides of the equation:

$$e^{\ln\left[\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right]} = e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}$$

$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}$$

We include the right-side of this in the argument `fun=` of the `geom_function()` layer, substituting in the values for $\hat{\beta}_0$ and $\hat{\beta}_1$. Below we plot the results from our fitted logistic model.

```
# Plot the fitted equation
ggplot(data = grad, aes(x = act, y = degree)) +
  geom_point(alpha = 0) +
  geom_function(
    fun = function(x) {exp(-1.611 + 0.108*x)}
    ) +
  theme_light() +
  xlab("ACT score") +
  ylab("Predicted odds of graduating")
```



Predicted odds of graduating college as a function of ACT score.

The monotonic increase in the curve indicates the positive effect of ACT score on the odds of graduating. The exponential growth curve indicates that students with higher ACT scores have increasingly higher odds of graduating.

Back-Transforming to Probability

We can also back-transform from odds to probability. To do this, we will again start with the logistic fitted equation and use algebra to isolate the probability of graduating (π_i) on the left-hand side of the equation.

$$\ln\left[\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right] = \hat{\beta}_0 + \hat{\beta}_1(x_i)$$

$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}$$

$$\hat{\pi}_i = e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}(1-\hat{\pi}_i)$$

$$\hat{\pi}_i = e^{\hat{\beta}_0+\hat{\beta}_1(x_i)} - e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}(\hat{\pi}_i)$$

$$\hat{\pi}_i + e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}(\hat{\pi}_i) = e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}$$

$$\hat{\pi}_i(1 + e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}) = e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}$$

$$\hat{\pi}_i = \frac{e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}}$$

$$\hat{\pi}_i = \frac{e^{\hat{Y}_i}}{1 + e^{\hat{Y}_i}}$$

That is, to obtain the probability of graduating, we can transform the fitted values (i.e., the predicted log-odds) from the logistic model. For example, the intercept from the logistic fitted equation, -1.61 was the predicted log-odds for students with an ACT of 0. To obtain the predicted probability of graduating for students with ACT of 0:

```
exp(-1.61) / (1 + exp(-1.61))
```

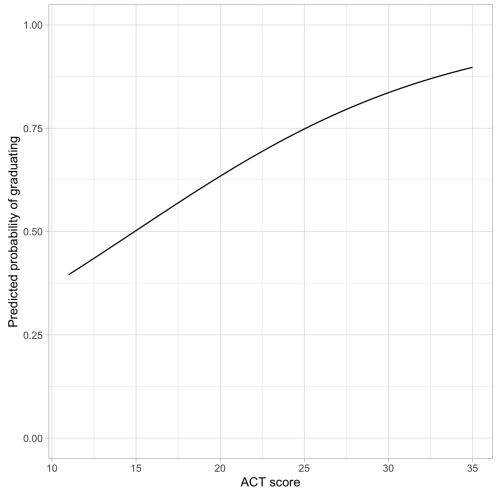
```
[1] 0.1665886
```

For students with ACT of 0, the predicted probability of graduating is 0.17.

This transformation from log-odds to probability, is non-linear, which means that there is not a clean interpretation of the effects of ACT (i.e., the slope) on the probability of graduating. To understand this effect we can plot the probability of graduating across the range of ACT scores. To do this, we use `geom_function()` and input the transformation to probability with the fitted equation:

$$\hat{\pi}_i = \frac{e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1(x_i)}}$$

```
# Plot the fitted equation
ggplot(data = grad, aes(x = act, y = degree)) +
  geom_point(alpha = 0) +
  geom_function(
    fun = function(x) {exp(-1.611 + 0.108*x) / (1 + exp(-1.611 + 0.108*x))}
    ) +
  theme_light() +
  xlab("ACT score") +
  ylab("Predicted probability of graduating") +
  ylim(0, 1)
```



Predicted probability of graduating college as a function of ACT score.

The effect of ACT on the probability of graduating follows a monotonic increasing "S"-curve. While there is always an increasing effect of ACT, the magnitude of this effect depends on ACT score. For lower ACT scores there is a larger effect of ACT score on the probability of graduating than for higher ACT scores.

One interesting point on the plot is the ACT score where the probability of graduating is 0.5. For us this is approximately 15. This implies that students who score less than 15 are more likely to not graduate than to graduate (on average), and those that score higher than 15 are more likely to graduate than not (on average).

Model-Level Summaries

The `glance()` output for the GLM model also included model-level information. For the model we fitted, the model-level output was:

```
# Model-level output
glance(glm.1)
```

```
# A tibble: 1 x 8
  null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
1         2723.    2343 -1317. 2637. 2649.    2633.        2342  2344
```

The metric of measuring residual fit is the deviance (remember the deviance was $-2\times$ log-likelihood). The value in the `null.deviance` column is the residual deviance from fitting the intercept-only model. It acts as a baseline to compare other models.

```
# Fit intercept-only model
glm.0 = glm(degree ~ 1, data = grad, family = binomial(link = "logit"))

# Compute deviance
-2 * logLik(glm.0)[[1]]
```

```
[1] 2722.546
```

The value in the `deviance` column is the residual deviance from fitting whichever model was fitted, in our case the model that used ACT score as a predictor.

```
-2 * logLik(glm.1)[[1]]
```

```
[1] 2633.236
```

Recall that deviance is akin to the sum of squared residuals (SSE) in conventional linear regression; smaller values indicate less error. In our case, the model that includes ACT score as a predictor has less error than the intercept only model; its deviance is 90 less than the intercept-only model.

There are two ways to determine whether this decrease in deviance is statistically significant. The first is to examine the *p*-value associated with the ACT predictor in the fitted model. Since that is the only predictor included above-and-beyond the intercept, the *p*-value associated with it indicates whether the ACT predictor is statistically relevant.

The second method to test the improvement in deviance is a test of nested models. Since the intercept-only model is nested in the model that includes ACT as a predictor, we can use a *Likelihood Ratio Test* to examine this. To do so, we use the `anova()` function with the added argument `test="LRT"` .

```
anova(glm.0, glm.1, test = "LRT")
```

```
Analysis of Deviance Table

Model 1: degree ~ 1
Model 2: degree ~ 1 + act
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2343     2722.6
2      2342     2633.2  1    89.31 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis of this test is that there is NO improvement in the deviance. The results of this test, $\chi^2(1) = 89.3, p < .001$, indicate that the observed difference of 89.3 is more than we would expect if the null hypothesis was true. In practice, this implies that the more complex model has significantly less error than the intercept-only model and should be adopted.

We could have also used model evidence for making decisions about effects.

```
aictab(
  cand.set = list(glm.0, glm.1),
  modnames = c("Intercept-Only", "Effect of ACT")
)
```

```
Model selection based on AICc:

               K    AICc Delta_AICc AICcWt Cum.Wt       LL
Effect of ACT  2 2637.24       0.00      1      1 -1316.62
Intercept-Only 1 2724.55      87.31      0      1 -1361.27
```

Here, given the data and the candidate set of models, there is overwhelming evidence to support the model that includes ACT score.

Pseudo R-Squareds

We can also use the residual deviance to compute a pseudo R^2 value for the model. As with the LMER models, we compare a model's residual deviance to the baseline residual deviance (that from the intercept-only model).

```
# Baseline residual deviance: 2722.6
# Model residual deviance: 2633.2

# Compute pseudo R-squared
(2722.6 - 2633.2) / 2722.6
```

```
[1] 0.03283626
```

Interpreting pseudo R^2 values are somewhat problematic. A rough interpretation is that differences in ACT scores explains 3.28% of the variation in graduation status. However, this interpretation is a bit sketchy. Pseudo R^2 values mimic R^2 values in that they are generally on a similar scale, ranging from 0 to 1 (though remember pseudo R^2 values can be negative). Moreover, higher pseudo R^2 values, like R^2 values, indicate better model fit. So while I wouldn't offer the earlier interpretation of the value of 0.0328, this does suggest that ACT scores are not perhaps incredibly predictive of the log-odds of graduating.

::note In logistic regression, several pseudo R^2 values have been proposed. See https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/ (https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/) for more information. :::

Including Covariates

Including covariates in the logistic model is done the same way as for `lm()` models. For example, say we wanted to examine the effect of ACT score on probability of graduating, after controlling for whether or not a student was first generation college student. We fit that model as

```
# Fit model
glm.2 = glm(degree ~ 1 + act + firstgen, data = grad, family = binomial(link
        = "logit"))
```

To evaluate this model, we will examine the model evidence comparing this model to both the baseline model (intercept-only) and the model that included the main-effect of ACT. For completeness, we will also fit and include (for comparison) a model that only includes the `firstgen` predictor.

```
# Fit model with only firstgen
glm.3 = glm(degree ~ 1 + firstgen, data = grad, family = binomial(link =
        "logit"))

# Model evidence
aictab(
  cand.set = list(glm.0, glm.1, glm.2, glm.3),
  modnames = c("Intercept-Only", "ACT", "ACT + First Gen.", "First Gen.")
)
```

```
Model selection based on AICc:

                 K    AICc Delta_AICc AICcWt Cum.Wt       LL
ACT + First Gen. 3 2615.22       0.00      1      1 -1304.61
ACT              2 2637.24      22.02      0      1 -1316.62
First Gen.       2 2666.09      50.87      0      1 -1331.04
Intercept-Only   1 2724.55     109.32      0      1 -1361.27
```

Given the data and candidate models fitted, the empirical evidence overwhelmingly supports including both ACT scores and first generation status in the model. Adopting this model, we next look at the coefficient-level output:

```
# Coefficient-level output
tidy(glm.2)
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  -1.48      0.285      -5.20 1.99e- 7
2 act           0.0881    0.0123      7.18 6.73e-13
3 firstgen      0.516     0.104       4.94 7.94e- 7
```

The fitted equation is

$$\ln\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = -1.48 + 0.09(\text{ACT Score}_i) + 0.52(\text{First Generation}_i)$$

Using the logit/log-odds metric, we interpret the coefficients as:

- For students who are not first generation college students with an ACT score of 0, the predicted log-odds of graduating are -1.48, on average.
- Each one-point difference in ACT score is associated with a difference of 0.09 in the predicted log-odds of graduating, on average, after controlling for whether o not the students are first

generation college students.

- First generation college students, on average, have a predicted log-odds of graduating that is 0.52 higher than students who are not first generation students, after controlling for differences in ACT scores.

Back-Transforming to Odds

If we back-transform the coefficients to facilitate interpretations using the odds metric,

```
exp(coef(glm.2))
```

```
(Intercept)         act    firstgen
  0.2275236   1.0921328   1.6745588
```

The fitted equation is:

$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{-1.48} \times e^{0.09(\text{ACT Score}_i)} \times e^{0.52(\text{First Generation}_i)}$$

The interpretations are:

- For students with an ACT score of 0 who are not first generation college students, the predicted odds of graduating are $e^{-1.48} = 0.23$, on average.
- Each one-point difference in ACT score is associated with improving the odds of graduating 1.09 times, on average, after controlling for whether or not the students are first generation college students.
- First generation college students, on average, predicted odds of graduating are 1.67 times that of students who are not first generation students, after controlling for differences in ACT scores.

We can also write the fitted equations for both first generation and non-first generation students and then use multiple `geom_function()` layers to plot the fitted curves.

Non-First Generation :

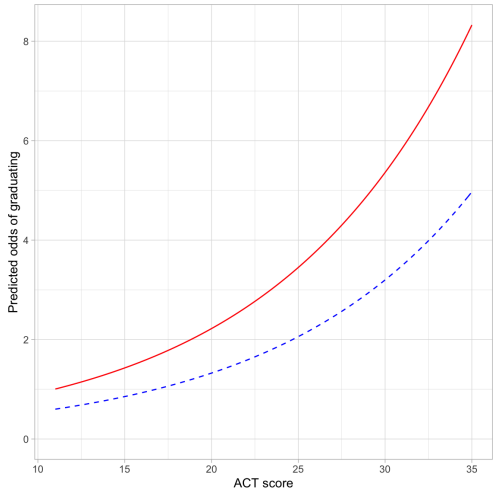
$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{-1.48} \times e^{0.09(\text{ACT Score}_i)} \times e^{0.52(0)}$$
$$= 0.227 \times e^{0.09(\text{ACT Score}_i)}$$

First Generation :

$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{-1.48} \times e^{0.09(\text{ACT Score}_i)} \times e^{0.52()}$$
$$= 0.227 \times e^{0.09(\text{ACT Score}_i)} \times 1.674$$
$$= 0.381 \times e^{0.09(\text{ACT Score}_i)}$$

Below I use the exponentiated fitted equation and substitute the values of `firstgen` into the respective functions to create the plot.

```
# Plot the fitted equation
ggplot(data = grad, aes(x = act, y = degree)) +
  geom_point(alpha = 0) +
  # Non-first generation students
  geom_function(
    fun = function(x) {exp(-1.48 + 0.0881*x + 0.516*0)},
    linetype = "dashed",
    color = "blue"
    ) +
  # First generation students
  geom_function(
    fun = function(x) {exp(-1.48 + 0.0881*x + 0.516*1)},
    linetype = "solid",
    color = "red"
    )+
  theme_light() +
  xlab("ACT score") +
  ylab("Predicted odds of graduating")
```



Predicted odds of graduating college as a function of ACT score first generation (solid, red line) and non-first generation (dashed, blue line) students.

Here we see that the odds of graduating increase exponentially at higher ACT scores for both first generation and non-first generation students, on average. This rate of increase, however, is higher fo first generation students. Moreover, first generation students have higher odds of graduating than non-first generation students, on average, regardless of ACT score.

Back-Transforming to Probability

We can also plot the predicted probability of graduating as a function of ACT score. Algebraically manipulating the fitted equation,

$$\hat{\pi}_i = \frac{e^{-1.48+0.088(\text{ACT}_i)+0.515(\text{First Generation}_i)}}{1 + e^{-1.48+0.088(\text{ACT}_i)+0.515(\text{First Generation}_i)}}$$

We can then produce the fitted equations for non-first generation and first generation students by substituting either 0 or 1, respectively, into the `firstgen` variable. These equations are:

Non-First Generation Students

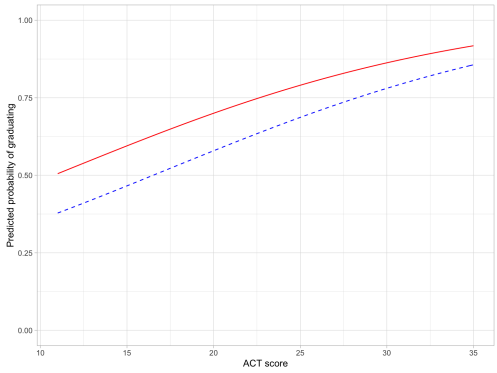
$$\begin{aligned}\hat{\pi}_i &= \frac{e^{-1.48+0.088(\text{ACT}_i)+0.515(0)}}{1 + e^{-1.48+0.088(\text{ACT}_i)+0.515(0)}} \\ &= \frac{e^{-1.48+0.088(\text{ACT}_i)}}{1 + e^{-1.48+0.088(\text{ACT}_i)}}\end{aligned}$$

First Generation Students

$$\begin{aligned}\hat{\pi}_i &= \frac{e^{-1.48+0.088(\text{ACT}_i)+0.515(1)}}{1 + e^{-1.48+0.088(\text{ACT}_i)+0.515(1)}} \\ &= \frac{e^{-0.965+0.088(\text{ACT}_i)}}{1 + e^{-0.965+0.088(\text{ACT}_i)}}\end{aligned}$$

We can include each of these in a `geom_function()` layer in our plot.

```
# Plot the fitted equations
ggplot(data = grad, aes(x = act, y = degree)) +
  geom_point(alpha = 0) +
  # Non-first generation students
  geom_function(
    fun = function(x) {exp(-1.48 + 0.0888*x) / (1 + exp(-1.481 + 0.088*x))},
    linetype = "dashed",
    color = "blue"
    ) +
  # First generation students
  geom_function(
    fun = function(x) {exp(-0.965 + 0.0888*x) / (1 + exp(-0.965 +
        0.088*x))},
    linetype = "solid",
    color = "red"
    ) +
  theme_light() +
  xlab("ACT score") +
  ylab("Predicted probability of graduating") +
  ylim(0, 1)
```



Predicted probability of graduating college as a function of ACT score for first generation (solid, red line) and non-first generation (dashed, blue line) students.

Here we see that the probability of graduating increase is positively associated with ACT score for both first generation and non-first generation students, on average. The magnitude of the effect of ACT depends on ACT score for both groups. Although first generation students have higher probability of graduating than non-first generation students, on average, regardless of ACT score, the magnitude of this difference decreases at higher ACT scores.

Presenting a Table of Regression Models

As with the linear models and linear mixed-effects models, it is important to present the results from the models fitted.

```
# Load stargazer library
library(stargazer)

# Table
stargazer(
  glm.0, glm.1, glm.3, glm.2,
  type = "html",
  title = "Coefficients and standard errors for a taxonomy of models fitted
        to predict graduation status. All models were fitted using a
        logistic regression and assuming binomial errors.",
  column.labels = c("Model A", "Model B", "Model C", "Model D"),
  colnames = FALSE,
  model.numbers = FALSE,
  dep.var.caption = "Outcome: Dummy-Coded Indicator of Graduation",
  dep.var.labels.include = FALSE,
  covariate.labels = c("ACT score", "First Generation Indicator"),
  keep.stat = NULL,
  notes.align = "l",
  add.lines = list(
    c("Residual Deviance", 2722.6, 2633.2, 2662.1, 2609.2),
    c("Corrected AIC", round(AICc(glm.0), 1), round(AICc(glm.1), 1),
      round(AICc(glm.3), 1), round(AICc(glm.2), 1))
    ),
  star.cutoffs = NA, # Omit stars
  omit.table.layout = "n" #Don't show table notes
)
```

Coefficients and standard errors for a taxonomy of models fitted to predict graduation status. All models were fitted using a logistic regression and assuming binomial errors.

Predictor	Model A	Model B	Model C	Model D
		Coefficient-level estimates		
ACT score		0.11		0.09
		(0.01)		(0.012)
First Generation Indicator			0.77	0.52
			(0.10)	(0.10)
Constant	1.01	-1.61	0.50	-1.48
	(0.05)	(0.28)	(0.08)	(0.29)
		Model-level estimates		
Residual Deviance	2722.6	2633.2	2662.1	2609.2
AICc	2724.5	2637.2	2666.1	2615.2

1. The logistic regression model is from a family of models referred to as *Generalized Linear Regression* models. The General Linear Model (i.e., fixed-effects regression model) is also a member of the Generalized Linear Model family.↩