**Logistic Regression output interpretation**

glm(formula = leaves.presence ~ Area, family = binomial(link="logit"), data = n)

Let's walk through this output (notice that I changed the name of the variable in the second line under Coefficients):

Deviance Residuals:

Min 1Q Median 3Q Max

-1.213 -1.044 -1.023 1.312 1.344

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.3877697 0.0282178 -13.742 < 2e-16 \*\*\*

Area 0.0008166 0.0002472 3.303 0.000956 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 16662 on 12237 degrees of freedom

Residual deviance: 16651 on 12236 degrees of freedom

(314 observations deleted due to missingness)

AIC: 16655

Number of Fisher Scoring iterations: 4

Just as there are [residuals](http://en.wikipedia.org/wiki/Errors_and_residuals_in_statistics) in linear (OLS) regression, there can be residuals in logistic regression and other generalized linear models. They are more complicated when the response variable is not continuous, however. GLiMs can have five different types of residuals, but what comes listed standard are the deviance residuals. ([Deviance](http://en.wikipedia.org/wiki/Deviance_%28statistics%29) and deviance residuals are more advanced, so I'll be brief here; if this discussion is somewhat hard to follow, I wouldn't worry too much, you can skip it):

Deviance Residuals:

Min 1Q Median 3Q Max

-1.213 -1.044 -1.023 1.312 1.344

For every data point used in your model, the deviance associated with that point is calculated. Having done this for each point, you have a set of such residuals, and the above output is simply a non-parametric description of their distribution.

Next we see the information about the covariates, which is what people typically are primarily interested in:

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.3877697 0.0282178 -13.742 < 2e-16 \*\*\*

Area 0.0008166 0.0002472 3.303 0.000956 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

For a simple logistic regression model like this one, there is only one covariate (Area here) and the intercept (also sometimes called the 'constant'). If you had a multiple logistic regression, there would be additional covariates listed below these, but the interpretation of the output would be the same. Under Estimate in the second row is the coefficient associated with the variable listed to the left. It is the estimated amount by which the log odds of leaves.presence would increase if Areawere one unit higher. The log odds of leaves.presence when Area is 00 is just above in the first row. (If you are not sufficiently familiar with log odds, it may help you to read my answer here: [interpretation of simple predictions to odds ratios in logistic regression](https://stats.stackexchange.com/a/34638/7290).) In the next column, we see the [standard error](http://en.wikipedia.org/wiki/Standard_error) associated with these estimates. That is, they are an estimate of how much, on average, these estimates would bounce around if the study were re-run identically, but with new data, over and over. (If you are not very familiar with the idea of a standard error, it may help you to read my answer here: [how to interpret coefficient standard errors in linear regression](https://stats.stackexchange.com/a/18213/7290).) If we were to divide the estimate by the standard error, we would get a quotient which is assumed to be normally distributed with large enough samples. This value is listed in under z value. Below Pr(>|z|) are listed the two-tailed [p-values](http://en.wikipedia.org/wiki/P-value) that correspond to those z-values in a standard normal distribution. Lastly, there are the traditional significance stars (and note the key below the coefficients table).

The Dispersion line is printed by default with GLiMs, but doesn't add much information here (it is more important with count models, e.g.). We can ignore this.

Lastly, we get information about the model and its goodness of fit:

Null deviance: 16662 on 12237 degrees of freedom

Residual deviance: 16651 on 12236 degrees of freedom

(314 observations deleted due to missingness)

AIC: 16655

Number of Fisher Scoring iterations: 4

The line about missingness is often, um, missing. It shows up here because you had 314 observations for which either leaves.presence, Area, or both were missing. Those partial observations were not used in fitting the model.

The Residual deviance is a measure of the lack of fit of your model taken as a whole, whereas the Null deviance is such a measure for a reduced model that only includes the intercept. Notice that the degrees of freedom associated with these two differs by only one. Since your model has only one covariate, only one additional parameter has been estimated (the Estimate for Area), and thus only one additional degree of freedom has been consumed. These two values can be used in conducting a test of the model as a whole, which would be analogous to the global FF-test that comes with a multiple linear regression model. Since you have only one covariate, such a test would be uninteresting in this case.

The [AIC](http://en.wikipedia.org/wiki/Akaike%27s_Information_Criterion) is another measure of goodness of fit that takes into account the ability of the model to fit the data. This is very useful when comparing two models where one may fit better but perhaps only by virtue of being more flexible and thus better able to fit any data. Since you have only one model, this is uninformative.

The reference to Fisher scoring iterations has to do with how the model was estimated. A linear model can be fit by solving closed form equations. Unfortunately, that cannot be done with most GLiMs including logistic regression. Instead, an iterative approach (the [Newton-Raphson algorithm](http://en.wikipedia.org/wiki/Newton-Raphson) by default) is used. Loosely, the model is fit based on a guess about what the estimates might be. The algorithm then looks around to see if the fit would be improved by using different estimates instead. If so, it moves in that direction (say, using a higher value for the estimate) and then fits the model again. The algorithm stops when it doesn't perceive that moving again would yield much additional improvement. This line tells you how many iterations there were before the process stopped and output the results.