Tutorial #2: Gaussian GLM - categorical predictor X

Guy F. Sutton

Centre for Biological Control, Rhodes University

2. Gaussian GLM (normal distribution)

The first GLM that we are going to look at is a *gaussian GLM*, or a GLM specified with a normal data distribution. As we saw in video 1, a linear regression (one continuous predictor) is a Gaussian GLM, and an ANOVA (one categorical predictor) is also a type of gaussian GLM.

The gaussian GLM is usually an appropriate starting point for your analysis when your response variabe is a continuous number (e.g. body mass, nitrogen concentration, leg length), but not when your response variable is a count (e.g. abundances) or a proportion.

An example:

Let's consider a study where we measure the body mass of 20 insects (`body_mass`), their body lengths (`body_length`), at each of two sites (`site`). Ultimately, we could be interested in asking whether (`body_mass`) is correlated with (`body_length`), and whether this relationship differs between sites (`site`).

```
head(data)
```

```
##    body_mass body_length site
## 1 18.7754002    15.28778    A
## 2 25.5245663    19.07157    A
## 3 23.4864950    14.49438    A
## 4 23.5963224    26.04058    A
## 5 28.9805369    11.87531    A
## 6  0.7743048    20.52689    A
```

2.1. Gaussian GLM - single categorical predictor

Let's consider a simple model where we look at whether the body mass of an insect (`body_mass`) differs between two sites (`site`). Here, we have a single categorical predictor variable (`site`).

$\mathbf{H_0}$: Insect body mass is not different between the two sites.

$\mathbf{H_1}$: Insect body mass is different between the two sites.

Fitting the model

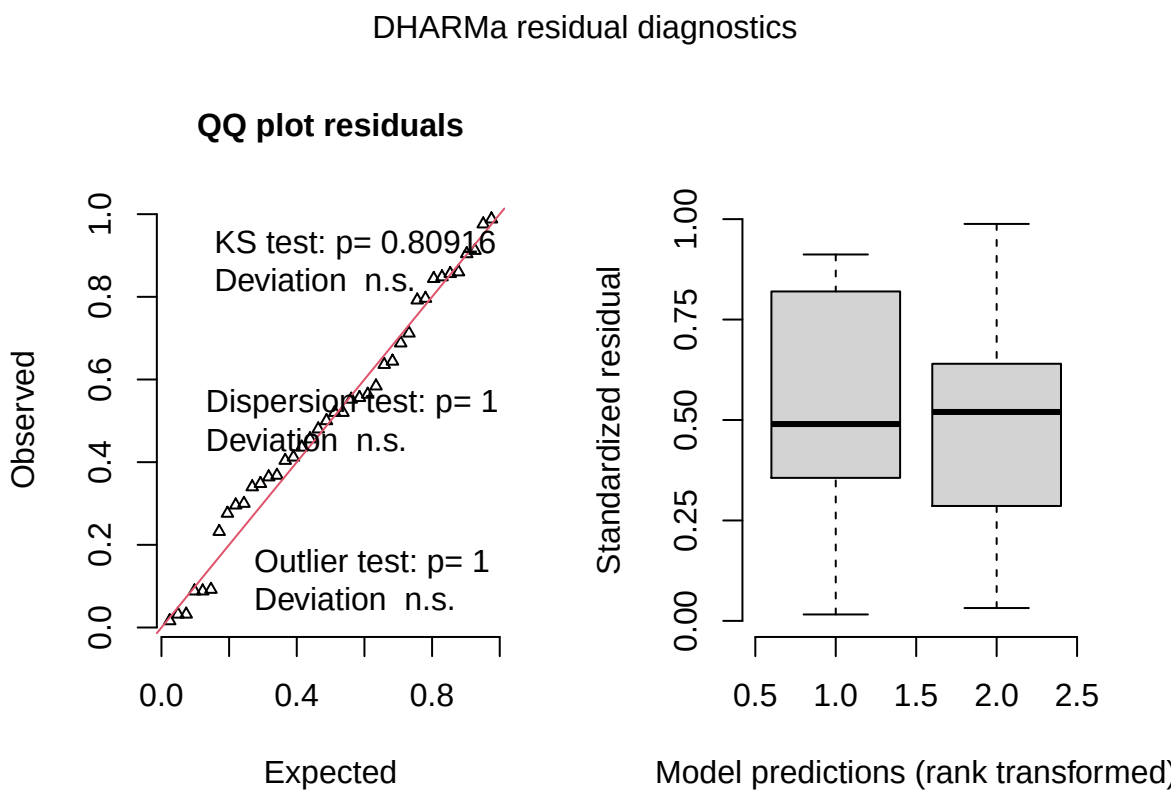
Fitting a GLM is simple in R. All we need to do is tell it what our response variable is (`body_mass`), and then we specify our predictor variables to the right-hand side of this weird ~ (tilde) symbol. Here, our predictor variable was `site`. We need to tell R where these data are stored (`data`), and that we want a gaussian GLM (`family = gaussian()`).

```
mod1 <- glm(body_mass ~ site,
            data = data,
            family = gaussian(link = "identity"))
```

Evaluate model fit

Before we look at the results from our model, we must first check whether the GLM that we fit was an appropriate choice for our data. We do that by looking at model diagnostics. The easiest and most informative way to do this in R is using the `DHARMa` package.

```
mod1_diagnostis <- DHARMa::simulateResiduals(mod1)
plot(mod1_diagnostis)
```



LHS plot (QQplot): This plot tells us whether our data conforms to a normal (or gaussian) data distribution. If our GLM is a good fit to the data, the white triangles will fall approximately on the red 1:1 line and the KS test P-value will be greater than 0.05. The Kolmorogorov-Smirnoff test (KS test) evaluates whether our data follow the distribution we specified in the `family` argument in the GLM call above (remember: we said "family = gaussian").

- Interpreting this plot tells us that the gaussian data distribution seemed appropriate.

RHS plot (Fitted vs Residuals): This plot tells us whether there are issues with homogeneity of variances. GLM's, like ANOVA and linear regression, assume that the variance in the different groups, or across the range of x values, are approximately equal. We want to see the bold black lines for the different groups fall approximately at y = 0.50, and the range of the boxplots should be relatively similar.

- Interpreting this plot we can happily see that the bold black lines for the two boxplots are approximately at y = 0.50 and the range of the boxplots is almost equal.

Take home: The GLM that we fit seems like a good fit to the data. The inferences that we draw and results we obtain should be reasonable. If either of these two plots produced issues (we will see some examples of models that are bad fits later), you DO NOT want to use the results. The p-values and inferences you draw will be biased, or just wrong.

Results

Now we get to the bit we are interested in: assessing statistical significance and calculating p-values. To do this, we will use a Likelihood Ratio Test (LRT). When we only have 1 predictor variable (here: `site`), we need to calculate p-values using type I sum-of-squares (SOS). SOS's are just different ways that we ask `R` to calculate p-values.

PLEASE DO NOT USE SUMMARY() - THIS WILL PRODUCE THE WRONG P-VALUES WHEN YOU HAVE MORE THAN 1 PREDICTOR VARIABLE

```
# Perform LRT  with type I sum-of-squares
car::Anova(mod1,
           # Specify we want a Likelihood Ratio Test
           test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: body_mass
##      LR Chisq Df Pr(>Chisq)
## site   8.3721  1    0.00381 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, the LRT tells us that insect `body_mass` was significantly different `sites` ($\chi^2 = 8.37$, d.f. $= 1$, P < 0.001).

In the next section, we will look at how much different insect body mass was at site A versus site B. There are a few ways to do this:

```
# Extract parameter estimate
summary(mod1)
```

```
##
## Call:
## glm(formula = body_mass ~ site, family = gaussian(link = "identity"),
##     data = data)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -23.1669   -5.4187   -0.3998    6.0128   22.2782
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   23.941      2.439   9.816 5.71e-12 ***
## siteB          9.981      3.449   2.893  0.00628 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 118.9812)
##
##     Null deviance: 5517.4  on 39  degrees of freedom
## Residual deviance: 4521.3  on 38  degrees of freedom
## AIC: 308.62
##
## Number of Fisher Scoring iterations: 2
```

Notice how `siteA` is missing? `R` uses the first level (alphabetically) of our predictor variable as the reference level, and then compares the rest of the levels to the reference.

The estimate for `(Intercept)` is actually just the mean value for our reference level when we have a categorical predictor. This interpretation is different when our predictor variable is continous (we will see this in our next model looking at `body_length`). So, the estimate value here tells us that the mean body mass at siteA is 23.941.

What is the mean body mass of insects from `siteB`? We simply add the estimate for `siteB` (9.98) to the mean for `siteA` (23.941) = 33.92 mg per insect at `siteB`.

If the estimate for `siteB` was -9.981, the mean body mass at `siteB` would be 23.941 - 9.981 = 13.96.

```
# Calculate mean +- standard error
data %>%
  dplyr::group_by(site) %>%
  dplyr::summarise(
    mean_body_mass = mean(body_mass),
    sd_body_mass = sd(body_mass),
    n = n(),
    se_body_mass = sd_body_mass / sqrt(n))
```

```
## # A tibble: 2 x 5
##   site  mean_body_mass sd_body_mass     n se_body_mass
## * <chr>          <dbl>        <dbl> <int>        <dbl>
## 1 A               23.9         10.9    20         2.43
## 2 B               33.9         10.9    20         2.44
```

Make a figure

Now we are going to make a figure to summarise our findings and that you can include in your thesis or paper.

```
# Calculate mean +- standard error
data %>%
  ggplot(data = ., aes(x = site,
```

```
                      y = body_mass)) +
geom_boxplot() +
# Add significance letters
scale_x_discrete("Site ",
                 labels = c("A", "B")) +
annotate("text", x = 1, y = 40, label = "a") +
annotate("text", x = 2, y = 60, label = "b") +
# Change axis labels
labs(y = "Insect body mass (mg)")
```