Workflow Linear Models II

Chris Howden Stanislaus Stadlmann

Table of contents


```
set.seed(171974)
project <- "Linear Models II"
version <- "1"
date    <- format(Sys.Date(), "%d-%m-%Y")
title   <- paste(project, " v", version, " ", date, sep = "")
```

The title of this project is Linear Models II v1 28-05-2024. The slides for the presentation accompanying this workflow can be found here.

In this workflow we focus on practical data analysis for two of the more common GLMs: Logistic regression for binary data (using a Binomial distribution); and Poisson/count regression for count data (using a Poisson distribution). The GLM framework is also described in detail.

Acknowledgements

If you used this workflow, please don't forget to acknowledge us. You can use the following sentence:

> The authors acknowledge the Statistical Workshops and Workflows provided by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney.

Libraries

Before we delve into the content, it is necessary to load certain libraries (the code is collapsed here). If you want to use them, please install them beforehand with `install.packages("library_name")`.

```
# No warnings
options(warn = -1)

suppressPackageStartupMessages({
  library("tibble")
  library("magrittr")
  library("statmod")
  library("ggplot2")
  library("emmeans")
  library("lme4")
  library("lmerTest")
  library("gglm")
  library("patchwork")
  library("haven")
  library("gplots")
  library("MASS")
  library("emmeans")
  library("DHARMa")
})

# GGplot theme
theme_set(theme_bw())
```

Logistic Regression

In this part, we will be exploring Logistic Regression Analysis. Let's start by creating some data. This part in code will also be collapsed, but you can always have a look by clicking the

"Code" button.

```
set.seed(171974)
success <- 100
failure <- 500
cc1 <- data.frame(lung_cancer = factor(c(rep("Lung Cancer", success), rep("No Lung Cancer"
cc1$healthy <- ifelse(cc1$lung_cancer=="Lung Cancer", rnorm(success, mean=40, sd=10), rnor
writexl::write_xlsx(cc1, "lung_cancer_dataset.xlsx")
```

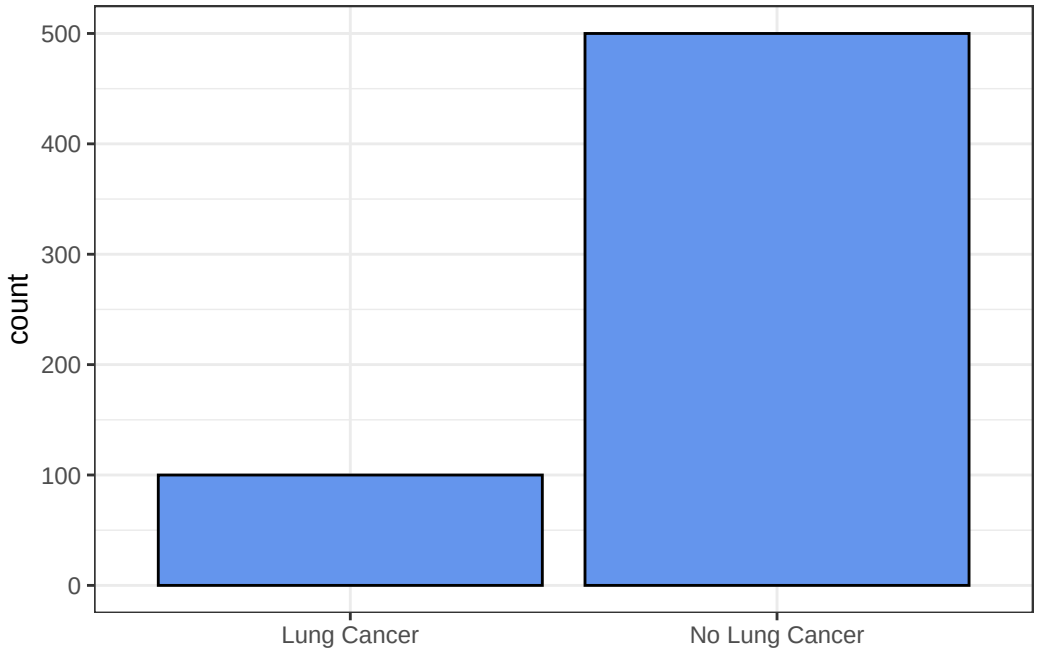
We now have a `data.frame` object called `cc1` with two variables:

- `lung_cancer`: A binary variable with two categories: `Lung Cancer` and `No Lung Cancer`
- `healthy`: This is a metric variable indicating how healthy a person is. The higher, the healthier.

Step 1: Exploratory Data Analysis

Let's first explore this dataset a little and look at the distribution of our dependent variable categories:

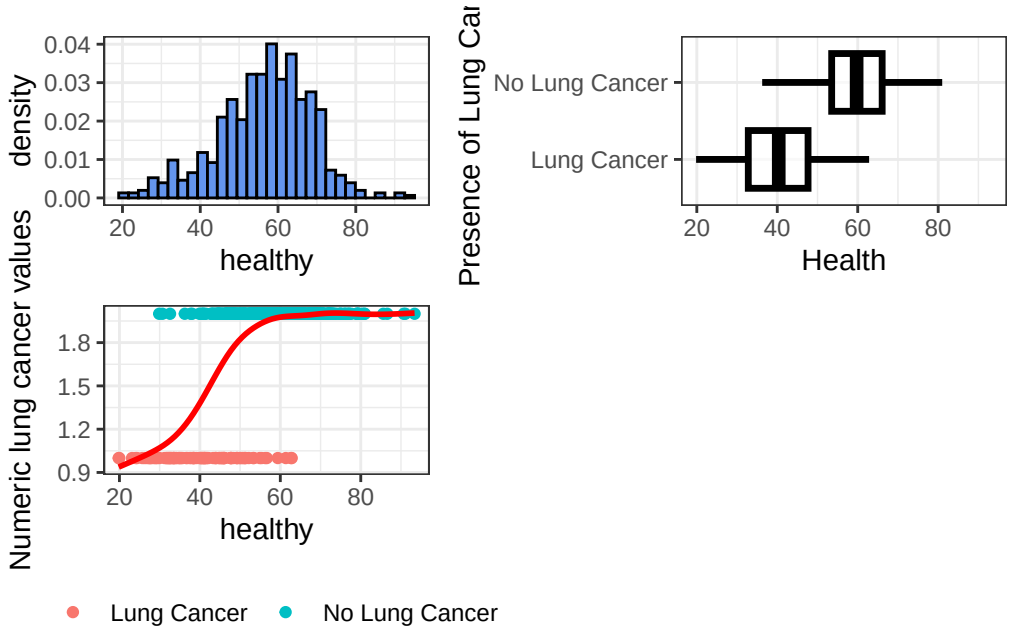
```
ggplot(cc1, aes(x = lung_cancer)) +
  geom_bar(fill = "cornflowerblue", col = "black") +
  labs(x = NULL)
```



How does it look like if we connect our dependent variable to the explanatory variable **healthy**? We are assuming that healthier people have a lower probability of lung cancer. Is this true?

```
# Plot 1
ggplot(cc1, aes(x = healthy, y = after_stat(density))) +
  geom_histogram(fill = "cornflowerblue", col = "black") +
# Plot 2
ggplot(cc1, aes(x = healthy, y = lung_cancer)) +
  # geom_violin(alpha=0.4, position = position_dodge(width = .75),size=1,color="black") +
  geom_boxplot(
    notch = FALSE,
    outlier.size = -1,
    color = "black",
    lwd = 1.2,
    alpha = 0.7
  ) +
  labs(x = "Health", y = "Presence of Lung Cancer") +
# Plot 3
ggplot(cc1, aes(x = healthy, y = as.numeric(lung_cancer), col = lung_cancer)) +
  geom_point() +
  geom_smooth(se = FALSE, col = "red", method = "gam", formula = y ~ s(x, bs = "tp")) +
  labs(y = "Numeric lung cancer values", col = NULL) +
  theme(legend.position = "bottom") +
# Plot layout
  plot_layout(ncol = 2)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In the above graphs, we see the values of `healthy` divided up by the categories of `lung_cancer`, in separate boxplots (top right). Also, we look at the distribution of `health` as well as the numeric dependency (bottom left).

We can indeed see signs of our previous hypothesis. As we have a binary response and a metric explanatory variable (`healthy`), a logistic regression analysis is the right way to go. All 3 plots tells us there are no outliers or other data problems with `healthy`.

Step 2: Fit Model

We fit the model here. Keep in mind that the "logistic regression" uses a binomial distribution, which is why we specify that here in the code. The word "logistic" comes from the link function used to connect explanatory variables to the parameter of interest, the event probability.

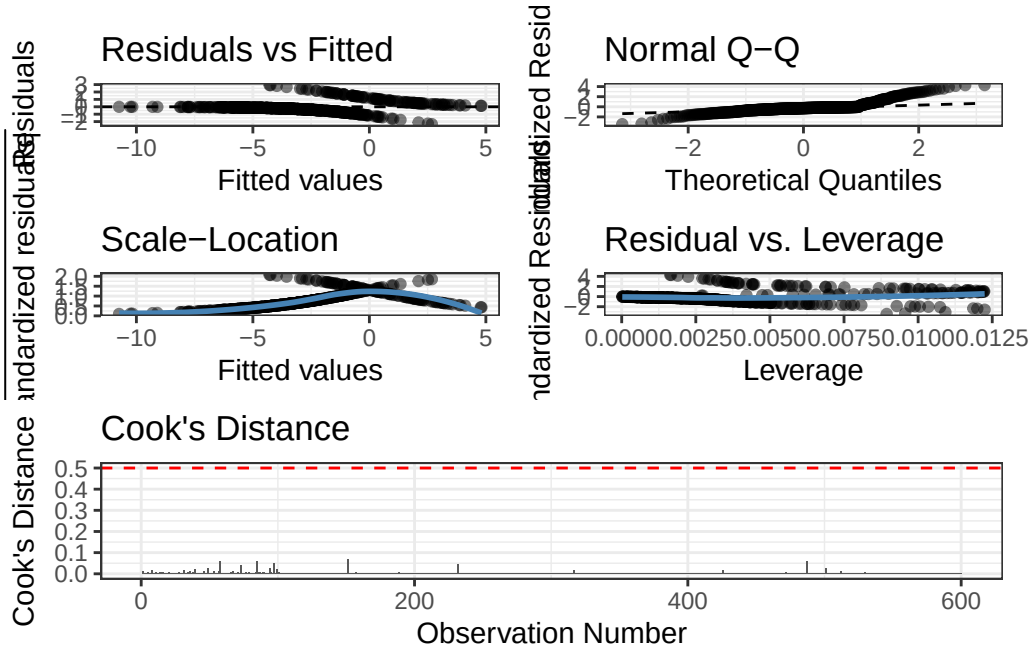
```
cc1$lung_cancer <- relevel(cc1$lung_cancer, ref = "No Lung Cancer")
logistic_reg <- glm(
  lung_cancer ~ healthy,
  data = cc1,
  family = "binomial")
```

Step 3: Check model assumptions

Residual Analysis

In generalised linear models, our assumptions are a little different. We don't have assumptions for residuals, but we assume that the conditional values of Y $\mathbf{X}$ has a exponential family distribution (which in this case is the binomial one). As such, our classic five-plot diagnostic graph is not as informative as in linear models. Let's have a look anyway:

```
gglm(logistic_reg, theme = theme_bw()) +
  ggplot(data = logistic_reg) +
  stat_cooks_obs() +
  geom_hline(yintercept = 0.5, linetype = "dashed", col = "red") +
  plot_layout(nrow = 2, heights = c(2, 1))
```

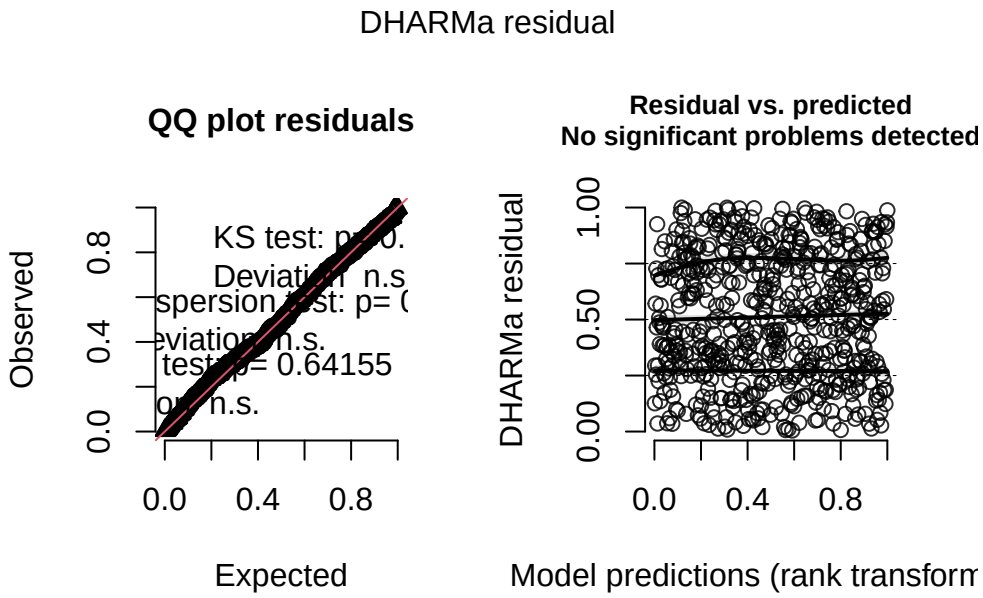


None of these graphs really work for us (except the last one), because they are built on normality assumptions (which are violated by design), but they are useful to check for outliers.

The package `DHARMa` can help in this case. It "uses a simulation-based approach to create readily interpretable scaled (quantile) residuals for fitted (generalized) linear mixed models." That means it can tell us how far off our residuals are from what we expect.

Let's have a look:

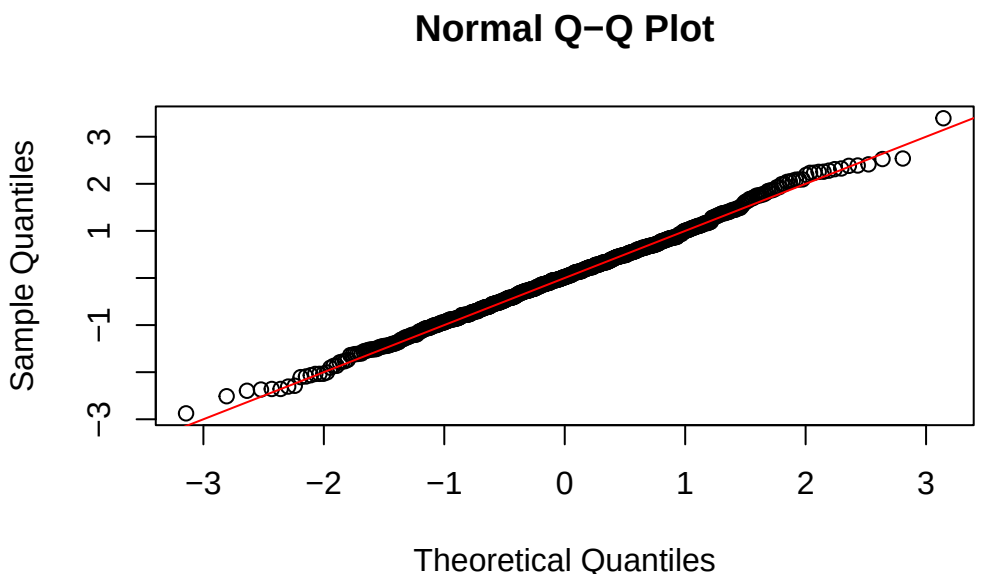
```
dharma_resids <- simulateResiduals(logistic_reg)
plot(dharma_resids)
```



DHARMa residuals simulate what the residuals would look like, given the modeling assumptions, and then compare them to our actual residuals. They even tell us in the plot whether there are any issues with the current residuals. According to these plots, our residuals are all fine.

We can also have a look at the "randomized quantile" residuals, which, according to Cindy Feng et al. (2020) are very effective in detecting false model specifications:

```
quantile_res <- qresiduals(logistic_reg)
qqnorm(quantile_res)
abline(0, 1, col = "red")
```



Step 4: Goodness of fit

Residual deviance compared to degrees of freedom

Evaluating goodness of fit involves assessing the residual deviance in comparison to degrees of freedom (DF). While considered a basic GLM test, it's less effective for binomial and Poisson families due to the asymptotic nature of the deviance's Chi-Squared distribution, particularly challenging for Bernoulli models with $n = 1$ (refer to MASS Section 7.2, page 195, Figure 7.3). Despite its limitations, this test, detailed in MASS (Section 7.1, page 186), is a good initial assessment.

A discrepancy in fit may imply issues like missing predictors, outliers, or overdispersion (Faraway Section 2.11, page 43). Notably, if the deviance is significantly larger, it signals a potential misspecification in the model, requiring further scrutiny in the assumption testing phase.

Let's calculate the metric:

```
deviance(logistic_reg)
```

```
[1] 285.1721
```

```
logistic_reg$df.residual
```

```
[1] 598
```

This looks good!

Test for separation

If any of the standard errors are larger than the coefficient estimates or we are getting any error messages about a cell being all 0 or 1 then we likely have seperation.

```
summary(logistic_reg)
```

```
Call:
glm(formula = lung_cancer ~ healthy, family = "binomial", data = cc1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  9.00166    0.97307   9.251   <2e-16 ***
healthy     -0.21145    0.02029 -10.422   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 540.67  on 599  degrees of freedom
Residual deviance: 285.17  on 598  degrees of freedom
AIC: 289.17

Number of Fisher Scoring iterations: 6
```

Also looks good.

Comparison with the Null model

This is a classic test in GLM - we compare the fitted model to a model without any predictor values, and check whether the model is a significant improvement.

```
null <- glm(lung_cancer ~ 1, data = cc1, family = binomial)
anova(null, logistic_reg, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: lung_cancer ~ 1
Model 2: lung_cancer ~ healthy
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       599     540.67
2       598     285.17  1    255.5 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our fitted model is significantly better than the Null model.

Explained deviance

This is similar to the R-squared in linear regression: % deviance explained.

```
((deviance(null) - deviance(logistic_reg)) / deviance(null)) * 100
```

```
[1] 47.25613
```

We have about 47% deviance explained, pretty good!

Step 5: Interpret model parameters and reach conclusion

For Simple Linear models we can simply look at the parameter estimate summary and CI's. However, in logistic regression these are hard to interpret as they are still on the logit scale.

The only really useful part of this 'raw' output is the p-value associated with the parameters. Which in this case shows strong evidence of being associated with `healthy` ($p < 2e-16$).

```
summary(logistic_reg)
```

```
Call:
glm(formula = lung_cancer ~ healthy, family = "binomial", data = cc1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  9.00166    0.97307   9.251   <2e-16 ***
healthy     -0.21145    0.02029 -10.422   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 540.67  on 599  degrees of freedom
Residual deviance: 285.17  on 598  degrees of freedom
AIC: 289.17

Number of Fisher Scoring iterations: 6
```

The parameters can be made more interpretable by taking their exponential since this turns them into odds ratios (which is explained in the presentation).

```
exp(coef(logistic_reg))
```

```
 (Intercept)      healthy
8116.5446454    0.8094061
```

```
exp(confint(logistic_reg))
```

```
Waiting for profiling to be done...
```

```
                   2.5 %       97.5 %
(Intercept) 1355.6456124 6.237704e+04
healthy        0.7755573 8.400245e-01
```

We get the above OR=0.81 for the continuous variable Health, which tells us that for each 1 point increase on the Health index the odds of getting lung cancer are 0.8 compared to the lower score (95%CI = 0.78-0.84).

Refer to the presentation here for more detail.

Step 6: Reporting

> There is strong evidence to show that being healthy is associated with lower odds of Lung Cancer (p<2e-16). For each 1 point increase on the Health index the Odds of getting lung Cancer are 0.8 compared to the lower score (95%CI odds ratio = 0.79-0.86). This effect on lung cancer has been estimated very accurately [as 95% CI is quite narrow]

The model is an acceptable fit to the data with a pseudo R2=45%. There were no outliers or unexplained structure. The model fit was a GLM with binomial distribution and logit link function"

Poisson regression

In order to explore this chapter, we have to load a dataset first.

```
load("tom scat DATA v9 24-11-2017.Rdata")
data2 <- data
data2$BeforeAfter <- relevel(data$BeforeAfter, ref = "After")
```

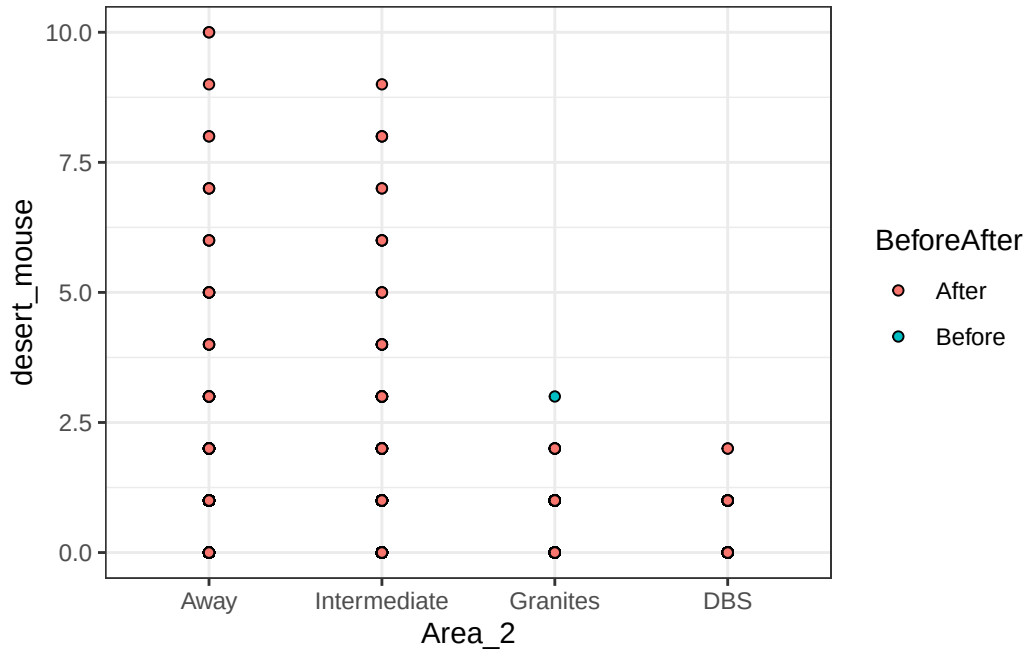
The variables of interest in this dataset are

- `desert_mouse`: A count variable (integer)
- `Area_2`: A categorical variable with four different levels: Away, Intermediate, Granites, DBS
- `BeforeAfter`: A categorical variable with two levels: After, Before

Step 1: Pick a suitable model via EDA

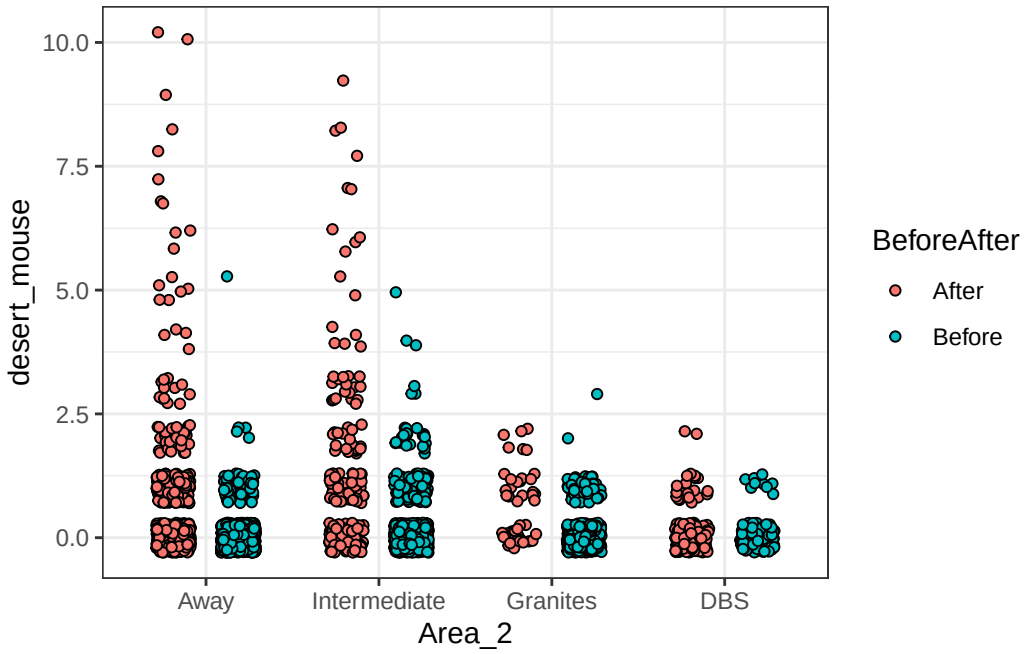
First, we display the dataset graphically:

```
ggplot(data = data2, aes(x = Area_2, y = desert_mouse, fill = BeforeAfter)) +
  geom_point(pch = 21)
```



So above is a plot for each of the 4 sites. But it's not very good since all the scats are overlayed on each other. Let's do some random scattering:

```
ggplot(data = data2, aes(x = Area_2, y = desert_mouse, fill = BeforeAfter)) +
  geom_point(pch = 21,
             position = position_jitterdodge(jitter.width = 0.4, jitter.height = 0.3))
```



In this graph, the observations are separated by both the `Area_2` and `BeforeAfter` categories and then "jittered" horizontally (and a little vertically as well). This random jitter makes it easier to observe patterns in the data. We can see that each category has a very different pattern with regard to their count, but that generally the "After" category has a higher count value. This pattern is something we can exploit using our Count regression.

Step 2: Fit the model

Let's fit the model next:

```
poisson_reg <-
  glm(desert_mouse ~ Area_2 * BeforeAfter, data = data2, family = "poisson")
# nb_model <- glm.nb(desert_mouse ~ Area_2 * BeforeAfter, data = data2)
```

Step 3: Check model assumptions via diagnostics

Zero inflation

Zero inflation in count data occurs when there are too many zeros for a Poisson distribution to handle, often indicating two underlying processes: the occurrence of an event and the frequency if it occurs. Testing for zero inflation involves comparing the simulated number of zeros based on the overall average with the observed data:

```r
test.0i.theory <- rpois(mean(data2$desert_mouse), n = 10000)
prop.table(table(test.0i.theory)) * 100
```

```
test.0i.theory
    0     1     2     3     4
69.57 25.18  4.58  0.62  0.05
```

```r
round(prop.table(table(data2$desert_mouse)) * 100, 2)
```

```
    0     1     2     3     4     5     6     7     8     9    10
75.87 18.60  2.89  1.18  0.38  0.35  0.24  0.17  0.17  0.07  0.07
```

The deviations (bigger 0 proportion for our data) may suggest the need for a ZIP model, though the presented case is not concerning.

Overdispersion

For the same reasons explained in logistic regression Poisson distributions can be over dispersed i.e. there is too much variance for the single parameter in the Poisson distribution to handle.

We test this using a function from http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#overdispersion. There is ongoing research on this topic so more recent information and solutions may be available here.

Function definition here:

```r
overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model, type = "pearson")
  Pearson.chisq <- sum(rp^2)
  prat <- Pearson.chisq / rdf
  pval <- pchisq(Pearson.chisq, df = rdf, lower.tail = FALSE)
  c(chisq = Pearson.chisq, ratio = prat, rdf = rdf, p = pval)
}
```

```r
overdisp_fun(poisson_reg)
```

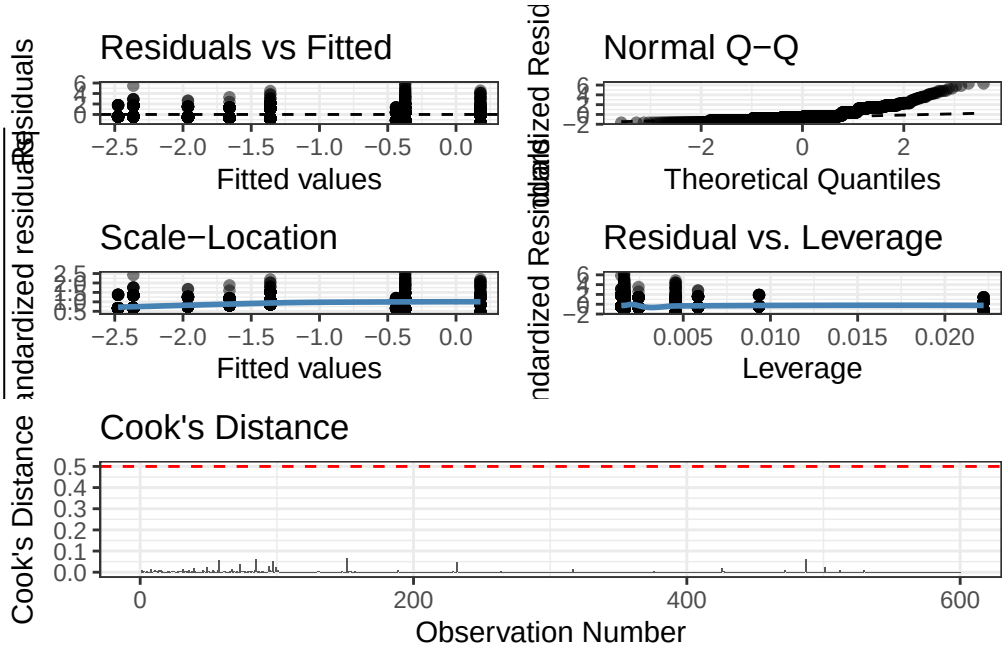
```
      chisq       ratio         rdf           p
4.467379e+03 1.557663e+00 2.868000e+03 9.576844e-74
```

This model is only marginally overdispersed.

Residual Analysis

Let's create our five-plot diagnosis graph:

```
gglm(poisson_reg, theme = theme_bw()) +
  ggplot(data = logistic_reg) +
  stat_cooks_obs() +
  geom_hline(yintercept = 0.5, linetype = "dashed", col = "red") +
  plot_layout(nrow = 2, heights = c(2, 1))
```



Due to the discrete-ness of the Poisson distribution, a lot of these plots show vertical lines of observations. This is to be expected. The "normal QQ" plot is not informative for us, because we don't assume a normal distribution of residuals.

Step 4: Goodness of fit

First, we compare the fitted model to our "Null" model, a model without any covariates.

```
null <- glm(desert_mouse ~ 1, data = data2, family = "poisson")
anova(null, poisson_reg, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: desert_mouse ~ 1
Model 2: desert_mouse ~ Area_2 * BeforeAfter
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2875     3477.5
2      2868     2743.8  7   733.69 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see, our model is significantly different from 0. Next, we'll check our Pseudo R^2 to see how much variance is explained.

```
((deviance(null) - deviance(poisson_reg)) / deviance(null)) * 100
```

```
[1] 21.09826
```

21 per cent.

Step 5: Interpret model parameters

First, let's look at the model output:

```
summary(poisson_reg)
```

```
Call:
glm(formula = desert_mouse ~ Area_2 * BeforeAfter, family = "poisson",
    data = data2)

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -0.37221    0.04834  -7.700 1.36e-14 ***
Area_2Intermediate                 0.55148    0.07844   7.030 2.06e-12 ***
Area_2Granites                    -0.06716    0.19188  -0.350 0.726340
Area_2DBS                         -1.59140    0.20977  -7.586 3.29e-14 ***
BeforeAfterBefore                 -1.99154    0.13312 -14.960  < 2e-16 ***
Area_2Intermediate:BeforeAfterBefore  0.45179    0.16705   2.704 0.006842 **
Area_2Granites:BeforeAfterBefore   0.77146    0.25500   3.025 0.002483 **
Area_2DBS:BeforeAfterBefore        1.47955    0.41292   3.583 0.000339 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3477.5  on 2875  degrees of freedom
Residual deviance: 2743.8  on 2868  degrees of freedom
AIC: 4299.9

Number of Fisher Scoring iterations: 6
```

We can see that most combinations of categories have a significant effect on the outcome variable.

Refer to the presentation here for more detail.

Effect size and comparisons

Let's look at the estimated marginal means (emmeans). Emmeans are a great way to calculate the predicted average value of the target distribution, given certain covariate combinations.

```
emm <- emmeans(poisson_reg,
               by = "Area_2",
               specs = "BeforeAfter",
               type = "response"
)
emm
```

```
Area_2 = Away:
 BeforeAfter   rate     SE  df asymp.LCL asymp.UCL
 After       0.6892 0.0333 Inf    0.6269     0.758
 Before      0.0941 0.0117 Inf    0.0738     0.120

Area_2 = Intermediate:
 BeforeAfter   rate     SE  df asymp.LCL asymp.UCL
 After       1.1963 0.0739 Inf    1.0599     1.350
 Before      0.2565 0.0205 Inf    0.2194     0.300

Area_2 = Granites:
 BeforeAfter   rate     SE  df asymp.LCL asymp.UCL
 After       0.6444 0.1197 Inf    0.4478     0.927
 Before      0.1902 0.0215 Inf    0.1524     0.238
```

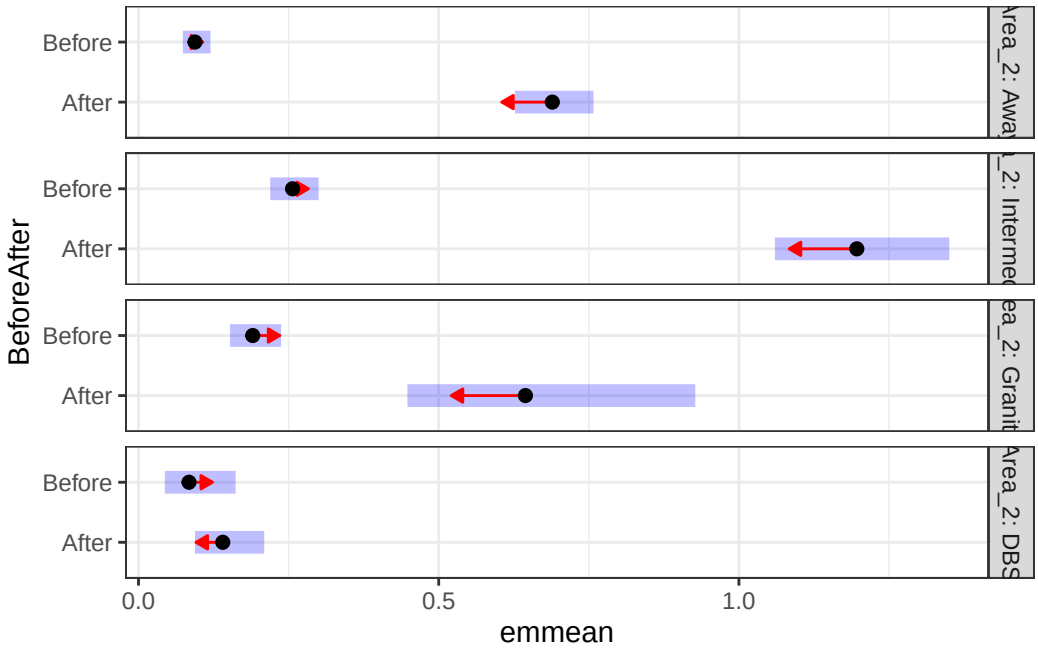
```
Area_2 = DBS:
 BeforeAfter   rate     SE  df asymp.LCL asymp.UCL
 After       0.1404 0.0286 Inf    0.0941     0.209
 Before      0.0841 0.0280 Inf    0.0438     0.162

Confidence level used: 0.95
Intervals are back-transformed from the log scale
```

Interestingly, the computed expected rates are highly different. Let's do a marginal mean comparison graphically and see which categories have the highest expected mean rate.

```
#were not getting the same result as Chris so before we show this we need to align them
#pairs(emmeans(poisson_reg,
#               specs = c("Area_2", "BeforeAfter")
#), transform = "response")
```

```
emm_plot <- plot(emm, comparisons = TRUE)
emm_plot
```



The plot above shows predicted conditional average values of the target distribution, but also allows for pairwise comparison. Wherever the red arrows overlap, a significant difference

between the categories is not present. Where they don't overlap, we have a significant category difference on the $\alpha = 0.05$ level. The top and bottom arrows only point into one direction, as they are the highest and lowest predicted value. Confidence intervals are given in blue (but not meant for cross comparison). Pairwise compared p values are available with the `pairs()` function.

Step 6: Overall conclusion suitable for publication

Because all covariates are categorical variables, the model is essentially a Two-Way ANOVA model. It therefore makes sense to test each whole effect (and not just the parameters) as well as the interaction:

```
anova(poisson_reg, test = "Chisq")
```

```
Analysis of Deviance Table

Model: poisson, link: log

Response: desert_mouse

Terms added sequentially (first to last)

                   Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                2875     3477.5
Area_2              3   126.67      2872     3350.8 < 2.2e-16 ***
BeforeAfter         1   587.66      2871     2763.1 < 2.2e-16 ***
Area_2:BeforeAfter  3    19.35      2868     2743.8 0.0002316 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both the individual `Area_2` and `BeforeAfter` effects are significantly different from 0 with a p value of $p < 2.2e{-}16$. The interaction effect is also statistically significant at $p < 0.0002316$.

The model is a good fit to the data with a pseudo $R^2 = 57\%$. There were no outliers or unexplained structure.

The model fit was a GLM with Poisson distribution and log link function. There was little evidence of over dispersion or zero inflation.