Lecture 6: More on Model Parameters

T. Neeman

19th August 2022

Model Parameters

```
library(tidyverse)
library(ggbeeswarm)
library(equatiomatic)
library(ggResidpanel)
library(emmeans)
library(GGally)
library(flextable)
```

In the last lecture, we set up a workflow for a data analysis of data of a simple experiment. In this lecture, we'll illustrate how a statistical model is very flexible framework for assessing associations and patterns in our data.

A model is defined by a set of parameters. The **parameters** are the constants in the equations or expressions that link the experimental factors to the mean response. For example, the (rain, yield) data, the form of the model was yield = a + b*rain + c*rain^2. The model parameters are a, b, and c. The data are used to **estimate** the model parameters. As with all estimates, one also estimates the **uncertainty** (SE) of the model parameter estimates.

The parameters in the seed orchard model were (1) the **mean** in each group, and (2) the **mean difference**. The variation around the means was also estimated, and is of interest to us, especially if we plan to do another study. For example, in a follow-up study, we'll need to estimate the signal:noise ratio; the larger the noise (variation) relative to the signal, the more samples we'll need to distinguish the groups.

We estimate 3 parameters, but there is some redundancy here, since mean_diff $= \mu_{SO} - \mu_P$. On the other hand, the mean difference is the parameter of greatest interest to us, because it estimates the association between treatment and outcome. So whilst this model has two parameters, there is more than one way to define the model parameterisation.

R parameterises the seed orchard model as follows:

- Parameter 1: mean of the reference group. The reference group is the group with factor level 1; in this case group 'SO'.
- Parameter 2: mean difference between two groups.

Notice that the mean of the P group is Parameter 1 + Parameter 2.

The parameter estimates are obtained via the **summary** function applied to the model object.

```
seed <- read_csv("../Data/seed orchard data.csv") %>%
        mutate(seedlot = factor(seedlot, levels = c("P","SO")))
```

```
## Rows: 16 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): seedlot
## dbl (2): plot, dbh
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
model.seed <- lm(dbh ~ seedlot, data = seed)

summary(model.seed)
```

```
##
## Call:
## lm(formula = dbh ~ seedlot, data = seed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0725 -0.9200 -0.2769  0.9887  3.8975
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  28.6925     0.7405  38.749 1.21e-15 ***
## seedlotSO     1.8888     1.0472   1.804   0.0928 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.094 on 14 degrees of freedom
## Multiple R-squared:  0.1886, Adjusted R-squared:  0.1306
## F-statistic: 3.253 on 1 and 14 DF,  p-value: 0.09284
```

Compare the parameter estimates with the mean estimates from the **emmeans** function

```
emmeans(model.seed, ~seedlot) %>%
  as_tibble() %>%
  flextable::flextable() %>%
  colformat_double(j = c(2:3,5:6), digits = 1)
```

seedlot	emmean	SE	df	lower.CL	upper.CL
P	28.7	0.7	14	27.1	30.3
SO	30.6	0.7	14	29.0	32.2

A model for multiple groups

Pea growth data: Peas were grown under 5 different growth media which differed in the type of sugar used (pea data.csv). The different types of sugar (including a no-sugar control) were: control, glucose, fructose, g&f and sucrose. The experimenter recorded the lengths of pea sections.

How many parameters in this model? What is the reference group for this experiment?

- Parameter 1: mean of the reference group ("control"): $\mu_{control}$
- Parameters 2 - 5: mean difference in pea length between each sugar treatment and control: $\beta_1, \beta_2, \beta_3, \beta_4$

Notice that $\mu_{fructose} = \mu_{control} + \beta_1(sugar_{fructose})$, $\mu_{glucose} = \mu_{control} + \beta_2(sugar_{glucose})$, etc.

When there are no differences amongst the varieties, then Parameters 2 - 5 = 0. The ANOVA table for variety has 4 degrees of freedom, and the associated p-value is a measure of evidence against the hypothesis that these 4 parameters are 0 (i.e. no treatment effect).

```
pea <- read_csv("../Data/pea data.csv")
```

```
## Rows: 50 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): sugar
## dbl (2): sampleNo, length
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(pea)
```

```
## spec_tbl_df [50 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ sampleNo: num [1:50] 1 2 3 4 5 6 7 8 9 10 ...
##  $ sugar   : chr [1:50] "control" "control" "control" "control" ...
##  $ length  : num [1:50] 75 67 70 75 65 71 67 67 76 68 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   sampleNo = col_double(),
##   ..   sugar = col_character(),
##   ..   length = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
model.pea <- lm(length ~ sugar, data = pea)
extract_eq(model.pea)
```

$$\text{length} = \alpha + \beta_1(\text{sugar}_{\text{fructose}}) + \beta_2(\text{sugar}_{\text{g\&f}}) + \beta_3(\text{sugar}_{\text{glucose}}) + \beta_4(\text{sugar}_{\text{sucrose}}) + \epsilon \tag{1}$$

```
summary(model.pea)
```

```
##
## Call:
## lm(formula = length ~ sugar, data = pea)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.100 -1.825 -0.150  0.975  5.900
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    70.1000     0.7386  94.907  < 2e-16 ***
## sugarfructose -11.9000     1.0446 -11.392 7.50e-15 ***
## sugarg&f      -12.1000     1.0446 -11.584 4.27e-15 ***
## sugarglucose  -10.8000     1.0446 -10.339 1.81e-13 ***
## sugarsucrose   -6.0000     1.0446  -5.744 7.48e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.336 on 45 degrees of freedom
## Multiple R-squared:  0.8144, Adjusted R-squared:  0.7979
## F-statistic: 49.37 on 4 and 45 DF,  p-value: 6.737e-16
```

Compare the model parameter estimates with the mean estimates obtained from the **emmeans** function.

```
emmeans(model.pea, ~sugar)
```

```
##  sugar    emmean    SE df lower.CL upper.CL
##  control    70.1 0.739 45     68.6     71.6
##  fructose   58.2 0.739 45     56.7     59.7
##  g&f        58.0 0.739 45     56.5     59.5
##  glucose    59.3 0.739 45     57.8     60.8
##  sucrose    64.1 0.739 45     62.6     65.6
##
## Confidence level used: 0.95
```

Models for associations between continuous variables

Breast cancer density data: Cases (Case = 1) were women who developed breast cancer after their first follow-up mammogram and Controls were still breast cancer-free at the time. Age (years) at first mammogram, body mass index at first mammogram. The researcher wants to assess whether breast density was a risk factor in this case-control study. She is also interested in the relationship between breast density and age and body mass index.

Let's explore the relationship between age, BMI and breast density

```
breast <- read_csv("../Data/breast cancer density.csv") %>%
  mutate(case = factor(case, labels = c("control","case")))
```

```
## Rows: 1065 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (5): case, ARM, AGE, BMI, density
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
summary(breast)
```

```
##       case          ARM             AGE             BMI            density
##  control:942   Min.   :1.000   Min.   :35.00   Min.   :17.60   Min.   :  0.00
##  case   :123   1st Qu.:1.000   1st Qu.:46.00   1st Qu.:23.20   1st Qu.: 15.00
##                Median :1.000   Median :49.00   Median :25.70   Median : 40.00
##                Mean   :1.476   Mean   :50.17   Mean   :26.72   Mean   : 44.45
##                3rd Qu.:2.000   3rd Qu.:54.00   3rd Qu.:29.40   3rd Qu.: 70.00
##                Max.   :2.000   Max.   :70.00   Max.   :50.40   Max.   :100.00
##                                                NA's   :16
```

Here are exploratory plots showing the relationship between age, BMI on breast density

```
ggpairs(breast,columns = c("AGE", "BMI", "density"), aes(col = factor(case), alpha = 0.1) )
```

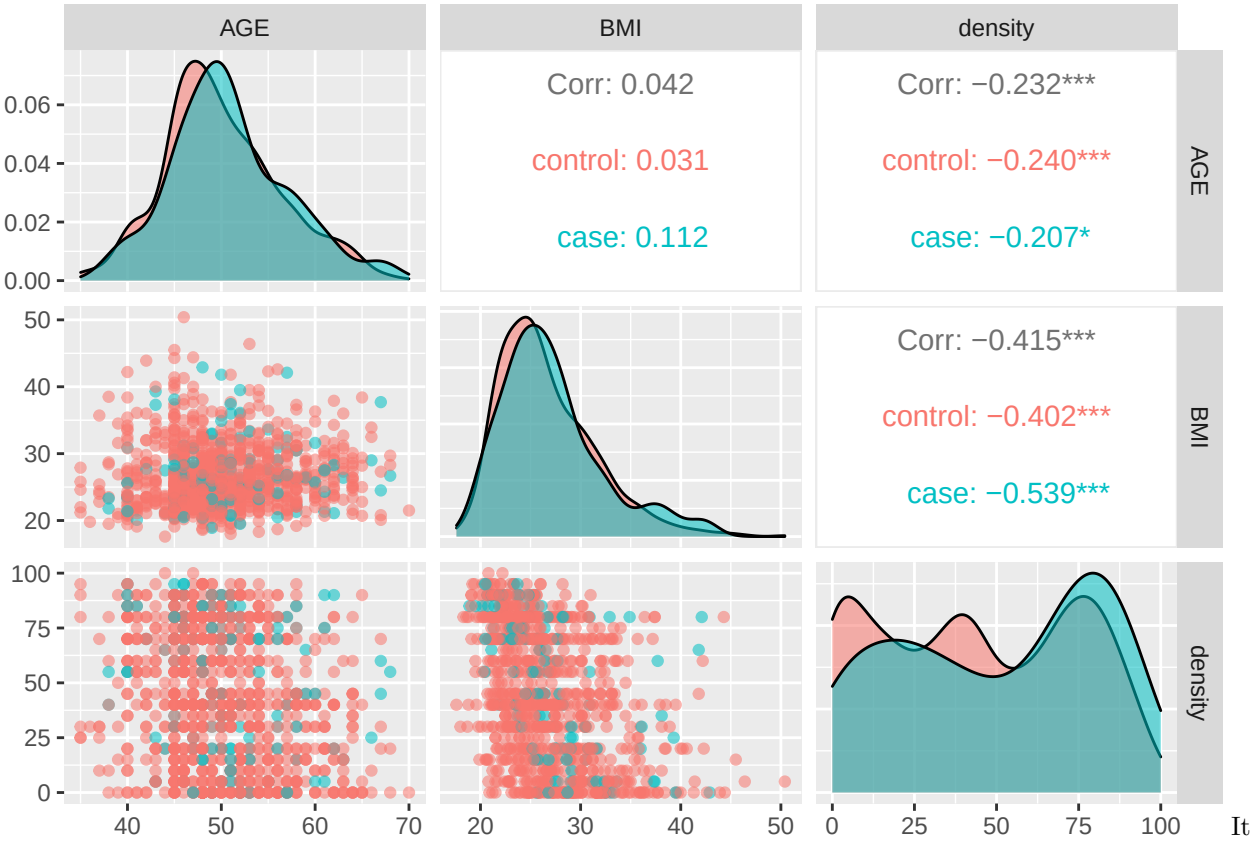
```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values

## Warning: Removed 16 rows containing missing values (geom_point).

## Warning: Removed 16 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values

## Warning: Removed 16 rows containing missing values (geom_point).
```



It looks like cases have on average higher breast density. There is a lot of variation between patients, but there

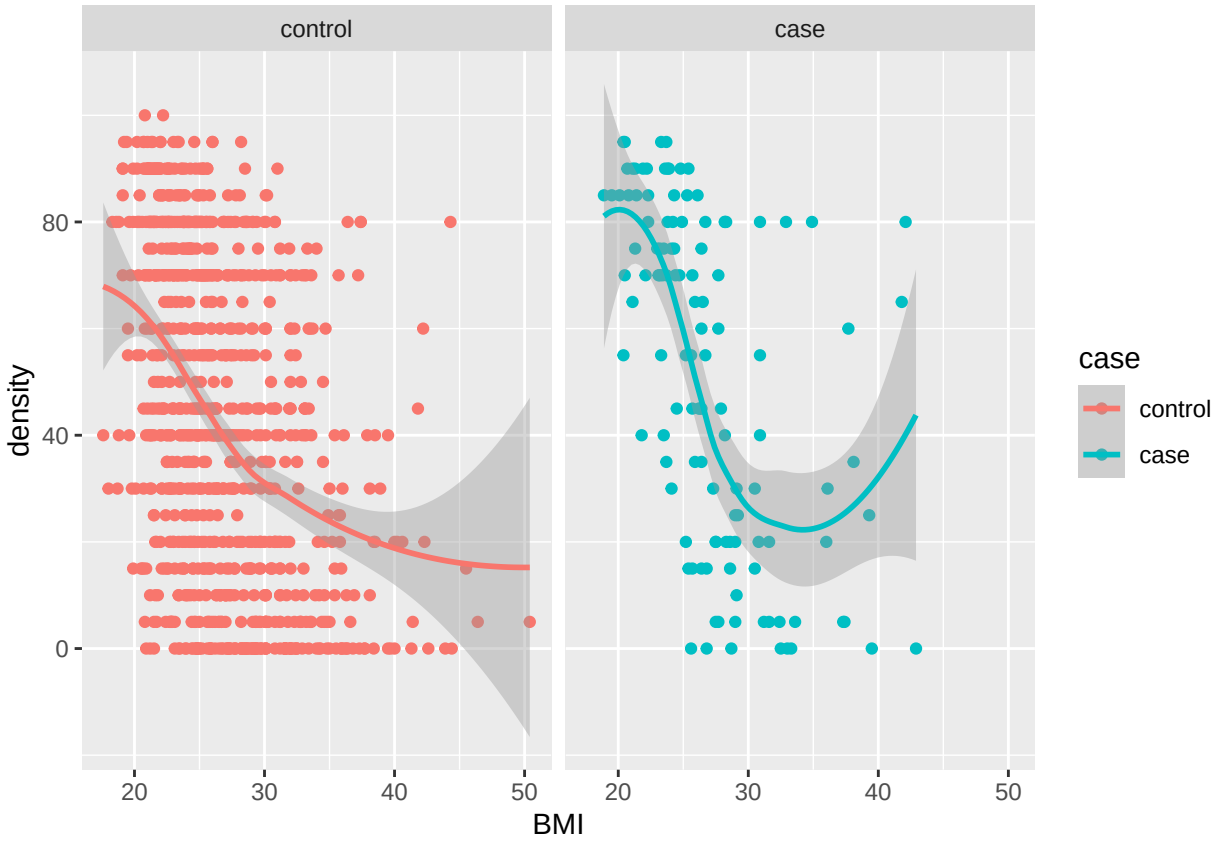
is significant negative correlation between age and density and between bmi and density, in both cases and controls. There is little correlation between bmi and age.

We can also look at these relationships using geom_smooth(). The code below looks at the association of BMI and breast density. You can repeat this code for age and breast density:

```
ggplot(breast, aes(BMI, density, col = case))+
  geom_point()+
  geom_smooth() +
  facet_wrap(~case)
```

```
## Warning: Removed 16 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 16 rows containing missing values (geom_point).
```



Let's fit 3 separate models to these data, interpret and compare the parameter estimates. In the equations below, parameter estimates are denoted by a, b, c, d, e, f and g. The lm() will also estimate the variance components (sigmas), but our focus will be on a:g.

Model 1: Density = a + b * AGE + N(0,sigma1) Model 2: Density = c + d * BMI + N(0, sigma2) Model 3: Density = e + f * AGE + g * BMI + N(0,sigma3)

```
mod_age <- lm(density~AGE, data = breast)
mod_BMI <- lm(density~BMI, data = breast)
mod_AGE_BMI <- lm(density ~ AGE + BMI, data = breast)
extract_eq(mod_age)
```

$$\text{density} = \alpha + \beta_1(\text{AGE}) + \epsilon \tag{2}$$

```
extract_eq(mod_BMI)
```

$$\text{density} = \alpha + \beta_1(\text{BMI}) + \epsilon \tag{3}$$

```
extract_eq(mod_AGE_BMI)
```

$$\text{density} = \alpha + \beta_1(\text{AGE}) + \beta_2(\text{BMI}) + \epsilon \tag{4}$$

The summary function gives the parameter estimates. Compare the parameter estimates of the first two models with the third model. What is R-squared? How does it change between models? What is the intercept? Is it a meaningful parameter?

```
summary(mod_age)
```

```
##
## Call:
## lm(formula = density ~ AGE, data = breast)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.183 -26.895  -0.766  26.492  62.295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 101.0775     7.3237  13.801  < 2e-16 ***
## AGE          -1.1288     0.1449  -7.791 1.57e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.36 on 1063 degrees of freedom
## Multiple R-squared:  0.05402,    Adjusted R-squared:  0.05313
## F-statistic:  60.7 on 1 and 1063 DF,  p-value: 1.575e-14
```

```
summary(mod_BMI)
```

```
##
## Call:
## lm(formula = density ~ BMI, data = breast)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -59.65 -22.93  -0.64  23.39  81.06
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 113.8651     4.7686   23.88   <2e-16 ***
## BMI          -2.5941     0.1756  -14.77   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.49 on 1047 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.1724, Adjusted R-squared:  0.1716
## F-statistic: 218.2 on 1 and 1047 DF,  p-value: < 2.2e-16
```

```
summary(mod_AGE_BMI)
```

```
##
## Call:
## lm(formula = density ~ AGE + BMI, data = breast)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.639 -21.639   0.591  21.479  81.736
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 165.8420     7.9658  20.819  < 2e-16 ***
## AGE          -1.0664     0.1330  -8.019 2.84e-15 ***
## BMI          -2.5367     0.1707 -14.860  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.69 on 1046 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.2204, Adjusted R-squared:  0.2189
## F-statistic: 147.8 on 2 and 1046 DF,  p-value: < 2.2e-16
```

To make the intercept more interpretable, one can centre the covariates, ie subtract off the mean. Then re-fit the model using centred age and centred BMI. Compare the parameter estimates. How would we interpret the intercept parameter?

```
breast <- breast %>%
  mutate(AGE_C = scale(AGE, scale = FALSE),
         BMI_C = scale(BMI, scale = FALSE))


mod_AGE_BMI_C <- lm(density ~ AGE_C + BMI_C,
                    data = breast)
summary(mod_AGE_BMI_C)
```

```
##
## Call:
## lm(formula = density ~ AGE_C + BMI_C, data = breast)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.639 -21.639   0.591  21.479  81.736
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  44.5726     0.8242  54.082  < 2e-16 ***
## AGE_C        -1.0664     0.1330  -8.019 2.84e-15 ***
## BMI_C        -2.5367     0.1707 -14.860  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.69 on 1046 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.2204, Adjusted R-squared:  0.2189
## F-statistic: 147.8 on 2 and 1046 DF,  p-value: < 2.2e-16
```

Nitrate availability and plant growth

Plants respond to external nitrate availability in the soil by altering their root mass ratio (RMR). Under low nitrogen conditions, plants allocate relatively more biomass to the root. Legumes have the ability to form root nodules in symbiosis with N_2-fixing rhizobia, and this may impact on the soil nitrogen - RMR relationship characteristic of other plants.

In this glasshouse experiment, researchers grew legumes under 6 different soil nitrogen conditions: 0.01, 0.1, 1,2, 10, and 100 mM. Seedlings were grown for 4 weeks, then harvested, dried and RMR measured.

Import data

```
legume<-read_csv("../Data/legume nitrate experiment.csv")
```

```
## Rows: 144 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (5): ID, tray, nitrate, RMR, log_RMR
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

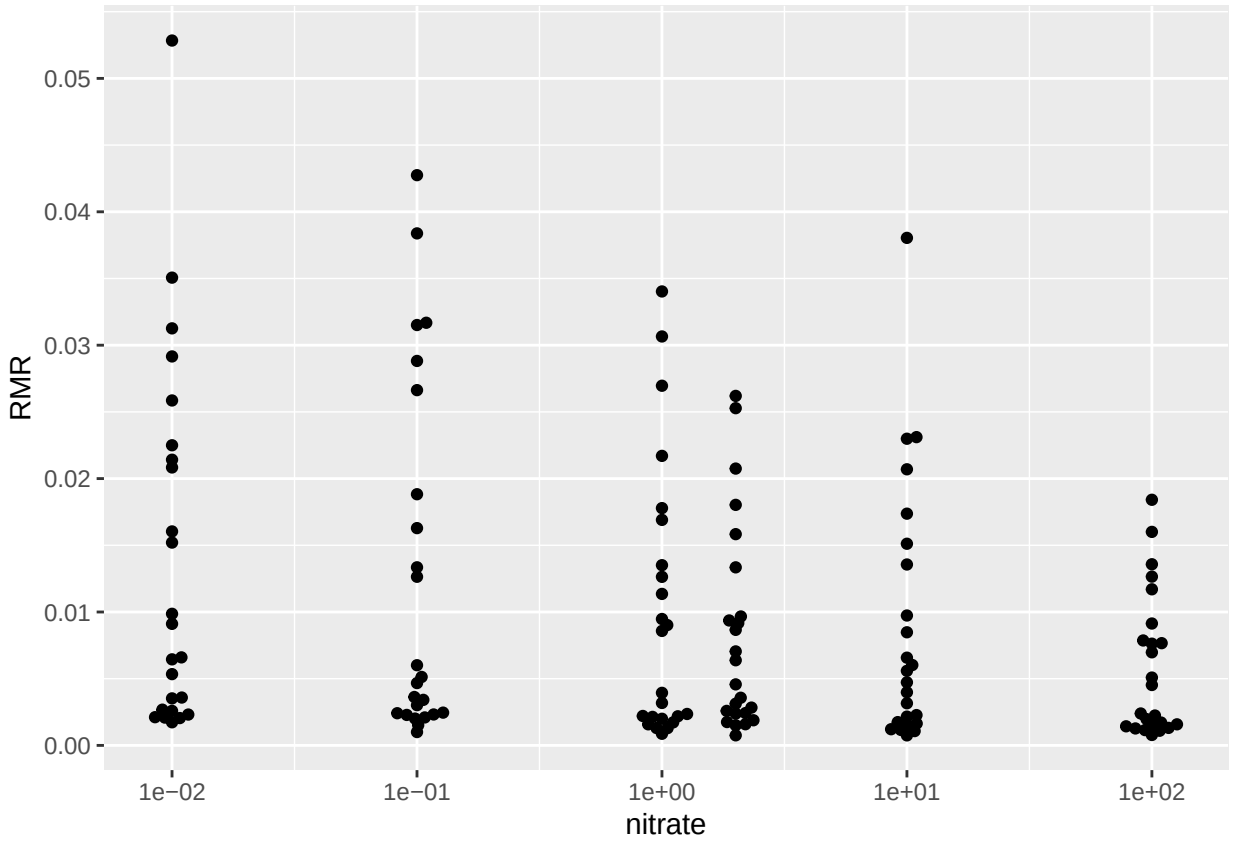
```
str(legume)
```

```
## spec_tbl_df [144 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ID     : num [1:144] 1 2 3 4 5 6 7 8 9 10 ...
##  $ tray   : num [1:144] 1 1 1 1 1 1 1 1 1 1 ...
##  $ nitrate: num [1:144] 0.01 0.01 0.01 0.01 0.1 0.1 0.1 0.1 1 1 ...
##  $ RMR    : num [1:144] 0.00268 0.00534 0.00211 0.00173 0.00301 ...
##  $ log_RMR: num [1:144] -5.92 -5.23 -6.16 -6.36 -5.8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ID = col_double(),
##   ..   tray = col_double(),
##   ..   nitrate = col_double(),
##   ..   RMR = col_double(),
##   ..   log_RMR = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

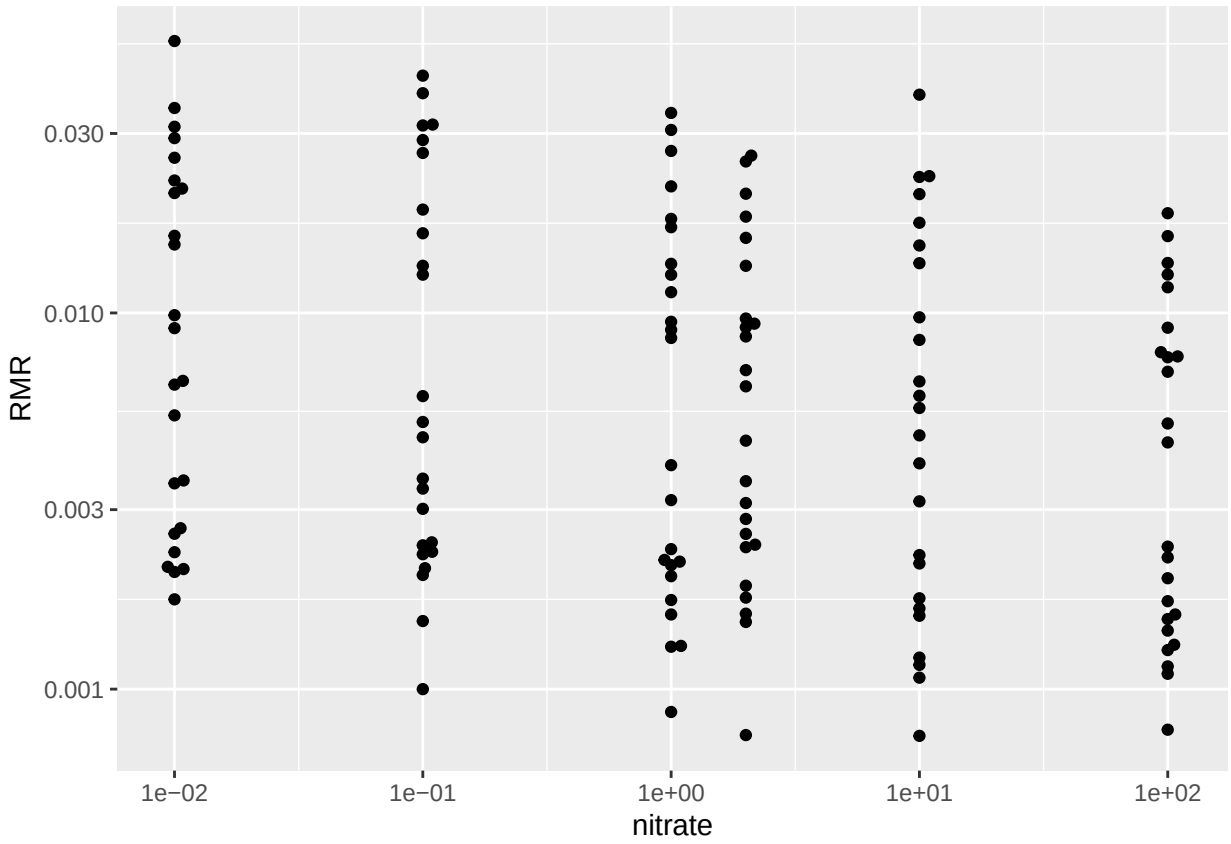
Data exploration

We look at the relationship between treatment (nitrate concentration) and response (RMR). Typically, one plots log(concentration) on the x-axis. We present both RMR and log(RMR) as potential response variables.

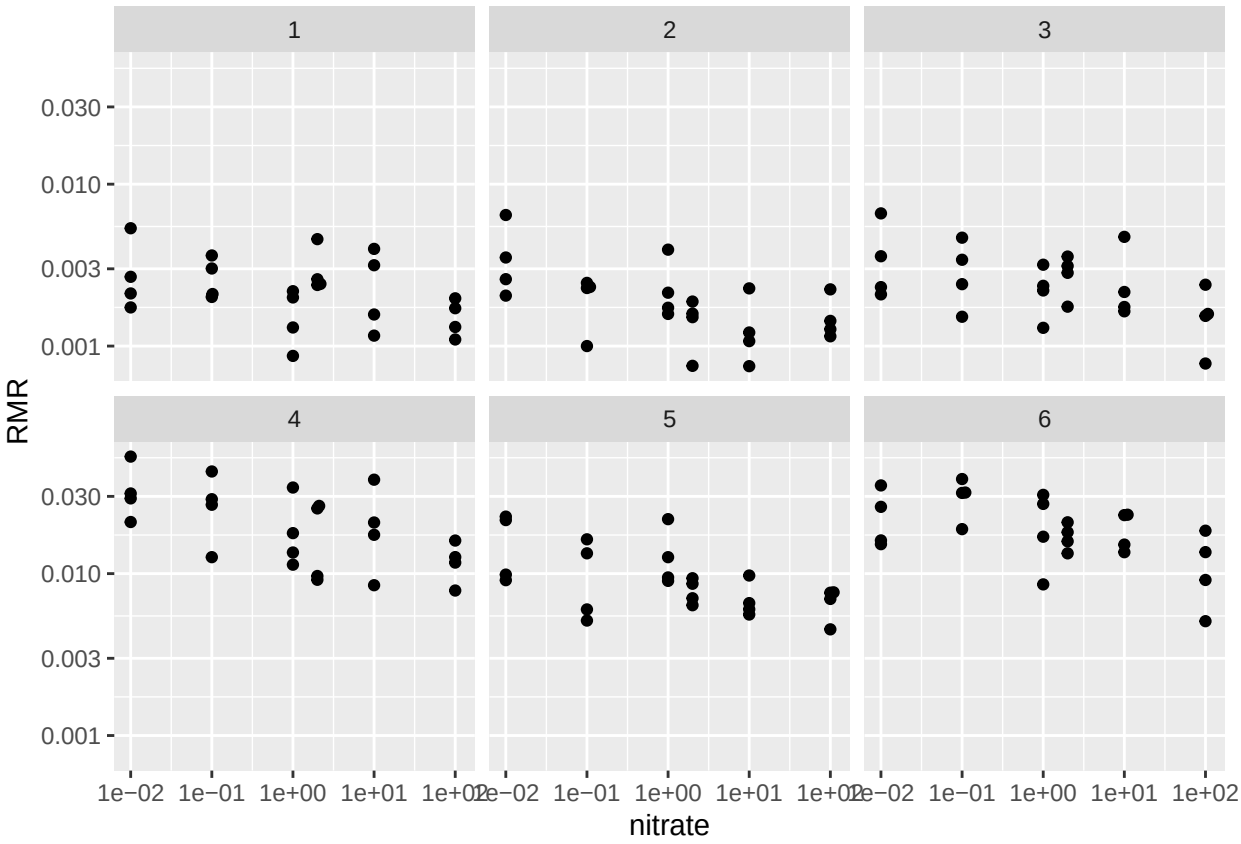
```
ggplot(legume, aes(x=nitrate, y=RMR))+
  geom_beeswarm()+
  scale_x_log10()
```




```
ggplot(legume, aes(x=nitrate, y= RMR))+
  geom_beeswarm()+
  scale_x_log10()+
  scale_y_log10()
```




```
ggplot(legume, aes(x=nitrate, y= RMR))+
  geom_beeswarm()+
  scale_x_log10()+
  scale_y_log10()+
  facet_wrap(~tray)
```



We consider our model assumptions that the data should be normally distributed around its mean. The RMR distribution is clearly non-normal; it has long tails. On the other hand, the log(RMR) distribution looks more "normal". This is a relatively common phenomenon with biological data, and visualising the data before modelling it will help us choose a reasonable model.

We have a couple of options regarded how we include nitrate in the model. On the one hand, we have 6 treatments, so nitrate can be a *factor* with 6 levels. On the other hand, we anticipate a dose-response relationship, with decreasing RMR for increasing log(nitrate concentration).

Option 1: Treat nitrate concentration as a factor with 6 levels. This model has *6 parameters*, namely the mean log(RMR) for each concentration level. One can estimate the mean differences in log(RMR) between any two nitrate concentration levels, but we might miss the real story, which is how does log(RMR) change for every 10-fold inrease in dose.

Option 2: Treat log10(nitrate concentration) as a continuous variable. The simplest model to consider is log(RMR) = a + b * log10(nitrate). This model has 2 parameters: a the intercept, and b the slope. b measures the association between log10(nitrate) and log(RMR). For every 10-fold increase in concentration, log(RMR) increases by b units.

We'll go with option 2, given the data and our research question. We include tray in the model because we notice that tray impacts on log_RMR.

Fit statistical model to data

```
model.legume<-lm(log(RMR) ~ log10(nitrate) + factor(tray), data = legume)
anova(model.legume)
```

```
## Analysis of Variance Table
```

```
##
## Response: log(RMR)
##                 Df  Sum Sq Mean Sq F value    Pr(>F)
## log10(nitrate)   1   8.634  8.6338   49.00 1.042e-10 ***
## factor(tray)     5 145.744 29.1488  165.43 < 2.2e-16 ***
## Residuals      137  24.139  0.1762
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model.legume)
```

```
##
## Call:
## lm(formula = log(RMR) ~ log10(nitrate) + factor(tray), data = legume)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91263 -0.26403  0.00374  0.28274  0.89101
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -6.13486    0.08569 -71.590  < 2e-16 ***
## log10(nitrate) -0.18896    0.02699  -7.000 1.04e-10 ***
## factor(tray)2  -0.17782    0.12117  -1.467    0.145
## factor(tray)3   0.10174    0.12117   0.840    0.403
## factor(tray)4   2.18102    0.12117  17.999  < 2e-16 ***
## factor(tray)5   1.44273    0.12117  11.906  < 2e-16 ***
## factor(tray)6   2.14183    0.12117  17.676  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4198 on 137 degrees of freedom
## Multiple R-squared:  0.8648, Adjusted R-squared:  0.8589
## F-statistic:   146 on 6 and 137 DF,  p-value: < 2.2e-16
```

The anova() function shows us the ANOVA table for the model. The important line is the first line, which indicates strong evidence that log10(nitrate concentration) is associated with log(RMR). We don't yet know the direction of the association. But the ANOVA table provides the inference that indicates that the "signal" we thought we might be seeing is probably "real".

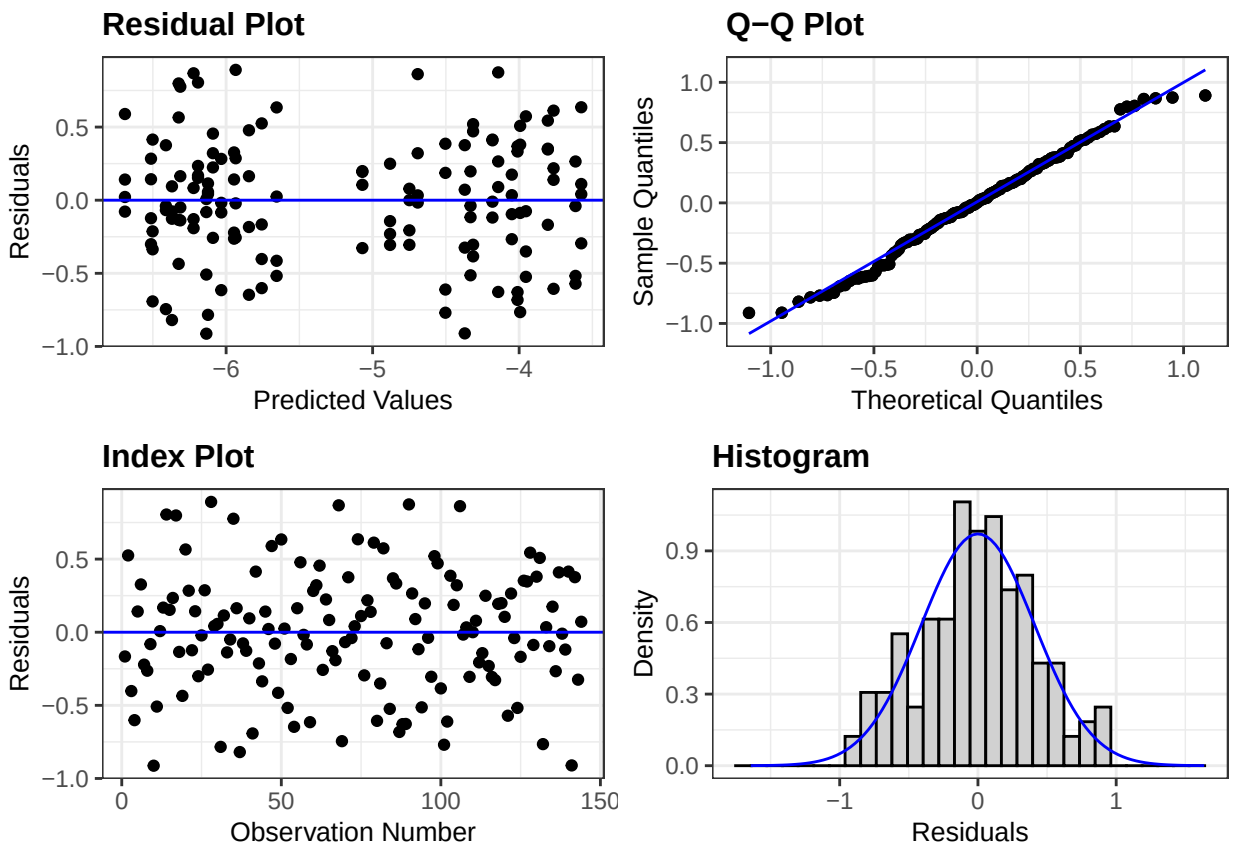
The summary() function provides the parameter estimates and SE(uncertainty). The two rows under Coefficients are the estimates for *a (Intercept)* and *b (slope)*. The slope estimate *-0.19* is negative, meaning that as log(concentration) increases, log(RMR) decreases. The standard error *0.027* is a measure of our uncertainty around the estimated slope. The t-value *-7.0* is the ratio of the estimate to the SE: -0.19 / 0.027. I call this the *signal-to-noise* ratio, as it measures the strength of the slope "signal" relative to our uncertainty of the magnitude of the signal. The further t is from 0, the stronger our evidence that the "signal" is real.

Finally, the p-value is derived from the t-value, and has an easily recognisable interpretation.

Check model assumptions

Following our standard workflow, we assess our model assumptions with residual plots.

```
resid_panel(model.legume)
```



Summarise model graphically

A summary of the model can be shown together with the data. We need something equivalent to means and standard errors. Instead of the emmeans() function, we'll use the predict() function to get estimated mean log_RMR for a range of nitrate concentrations. The predict() function includes an option for 95% confidence intervals around the estimated means. The confidence interval is mean +- 1.96*SE.

```
results1 <- emmeans(model.legume, ~nitrate+tray, type = "response",
                    at = list(nitrate = c(0.01, .1, 1, 2, 10,100),
                              tray = 5)) %>%
            as_tibble()
results1
```

```
## # A tibble: 6 x 7
##   nitrate  tray response       SE    df lower.CL upper.CL
##     <dbl> <dbl>    <dbl>    <dbl> <dbl>    <dbl>    <dbl>
## 1    0.01     5  0.0134  0.00136    137  0.0109   0.0164
## 2    0.1      5  0.0111  0.000999   137  0.00926  0.0132
## 3    1        5  0.00917 0.000786   137  0.00774  0.0109
## 4    2        5  0.00866 0.000744   137  0.00731  0.0103
## 5   10        5  0.00759 0.000679   137  0.00636  0.00906
## 6  100        5  0.00628 0.000632   137  0.00515  0.00766
```

```
ggplot(data=results1, aes(x=nitrate, y=response))+
  geom_line()+
  geom_ribbon(data=results1, aes(x=nitrate, ymin=response-SE, ymax=response+SE), alpha = 0.2)+
  geom_point(data = legume, aes(x=nitrate, y=RMR),col="darkgreen")+
  scale_x_log10()+
  scale_y_log10()+
  theme_classic()
```