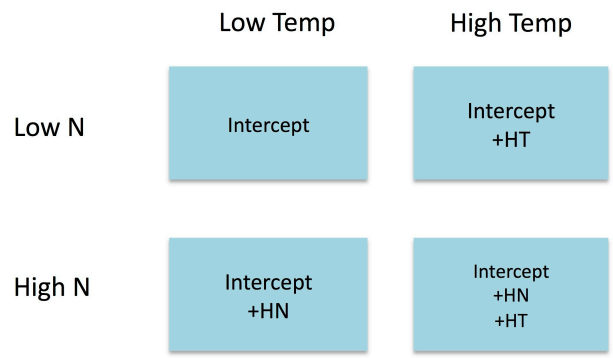
Lecture 11: Using and interpreting multiple terms; Contrasts and multiple comparisons

In the last lecture we discussed how the design of experiments, surveys, and statistical models affects our ability to draw valid inferences. In this lecture we will consider two further issues that frequently arise when we try to interpret complex (or even simple) models: (1) How to think about multiple predictors in observational datasets, and our ability to draw conclusions about them; (2) How to test whether levels of a grouping factor are different from each other.

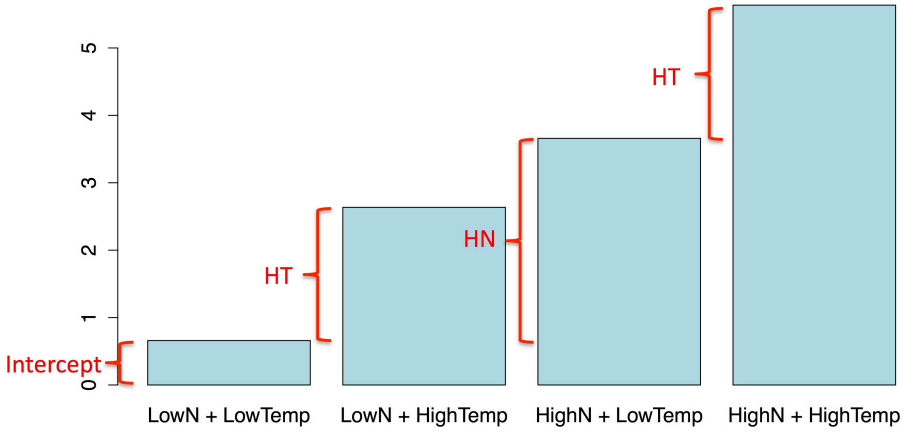
As discussed in the previous lecture, it's helpful to compare experimental manipulations to observational datasets. So far in the course, when introducing statistical techniques, I often talk about experiments and observational data interchangeably. The reason is that *the structure of the model is the same*, regardless of where the data came from. If you have an experiment with two or more treatments, the treatment variable will be entered into a linear model or a GLM as a factor. Likewise, if you did an experiment at multiple sites or in multiple years, you will account for variation among sites or among years with a factor. And if you have a purely observational dataset where the response might differ by some grouping variable (sites, years, species, etc.) then you will account for that with a factor. The statistical framework doesn't differ among these cases, because in each case we just want to model the difference in the response variable among groups of data, and it doesn't matter whether those groups are generated by the experimenter, or by nature, or by survey design. Later in the course we'll talk about using random vs. fixed effects for grouping variables, but the basic point will remain the same.

Although experiments and observational data are modeled the same way, there are complexities that arise more commonly for observational data, especially when we put multiple terms in the model. Let's think about how multiple model parameters are estimated, and what they mean. It will be helpful to compare a well-designed experiment to an observational dataset. Let's imagine that we want to quantify how temperature affects the growth of a particular phytoplankton species, perhaps in the context of global change. And maybe we also want to quantify how the effect of temperature compares to the effect of nitrogen supply. We use the following experimental design:



This is a nice factorial design, where we have all combinations of two treatments. And we replicate all combinations as well (just one replicate shown). The text inside the boxes shows how a linear model (or GLM) in R will estimate the treatments effects. Low N + Low Temp gets the baseline Intercept, the difference between High N and Low N is the parameter 'HN', and the difference between High Temp and Low Temp is the parameter 'HT'.

This design illustrates what a coefficient in a linear model means, when there are other coefficients in the model. The coefficient 'HighN' quantifies the effect of the N treatment, *holding temperature constant*. Likewise the coefficient 'HighTemp' quantifies the effect of the temperature treatment, *holding N constant*. The factorial experimental design is the ideal way to estimate these coefficients, because we have created treatments where N varies but temperature is constant, as well as treatments where temperature varies but N is constant. This kind of design is called *orthogonal*, because the two treatments are varied independently. Here is an example of what the fitted model might look like:

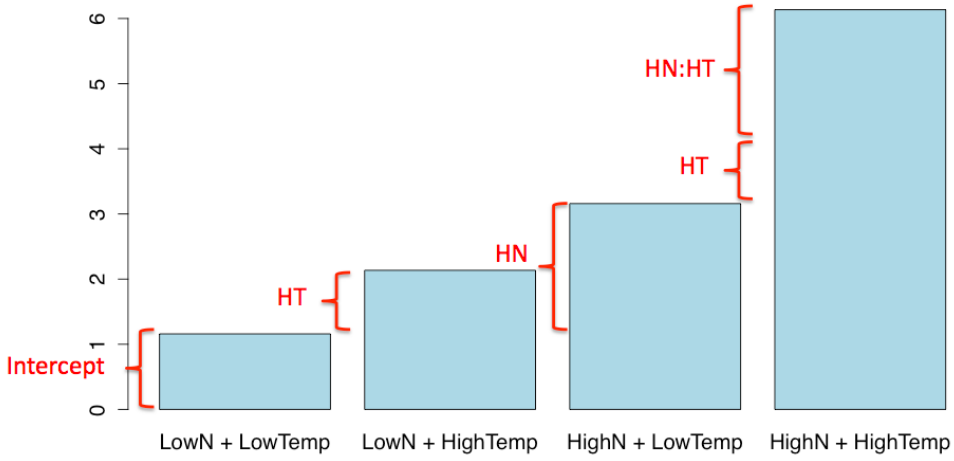


I've plotted the fitted model means, and the intervals in red show the meaning of the model parameters. Note that you get HighN+HighTemp by just adding the separate effects of HN and HT.

Because we have all combinations of both treatments, we can also quantify any interaction between N and temperature:

	Low Temp	High Temp
Low N	Intercept	Intercept +HT
High N	Intercept +HN	Intercept +HN +HT +HN:HT

Now the High N + High Temp treatment gets an extra parameter, 'HN:HT'. This quantifies how much the High Temp + High N treatment differs from what is predicted by just adding 'HighN' + 'HighTemp'. If 'HighN:HighTemp' = 0 there is no interaction, but if this parameter is nonzero then the effect of High N + High Temp is different from just adding the effect of the two treatments. It's easier to see on an example graph of an interaction:

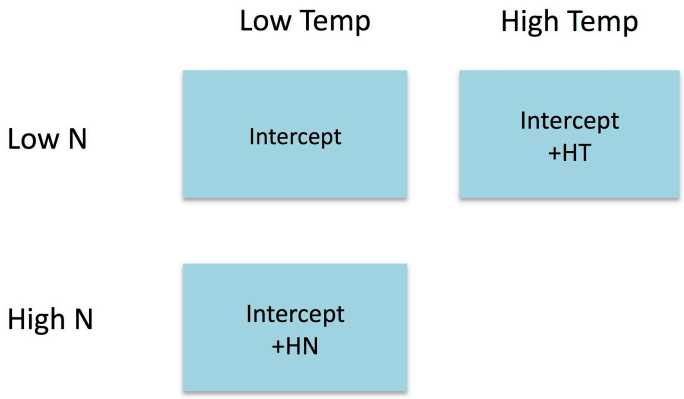


Now to get the HighN+HighTemp treatment you have to add HT, HN, and HN:HT, because the effects of HT and HN alone do not account for what happens when both N and temperature are increase.

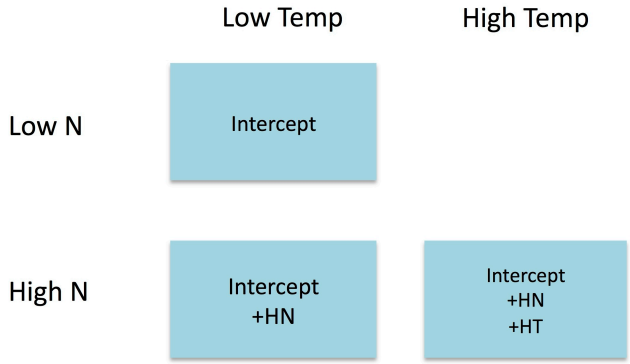
An important thing to note here is that *with the interaction parameter in the model, the meaning of the other parameters changes*. Now the parameter 'HN' is not the effect of N while holding temperature constant; rather it is the effect of N when temperature is low. Likewise, the parameter 'HT' is the effect of temperature when N is low. So when you add interactions into a model the meaning of the main effects changes. This is because with an interaction present it doesn't make sense to

say “What is the effect of N while holding temperature constant?”, because the interaction by definition means that the effect of N depends on the effect of temperature.

Now let’s think about what we could estimate if our design wasn’t as smart. For example, maybe we just made these three treatment combinations:

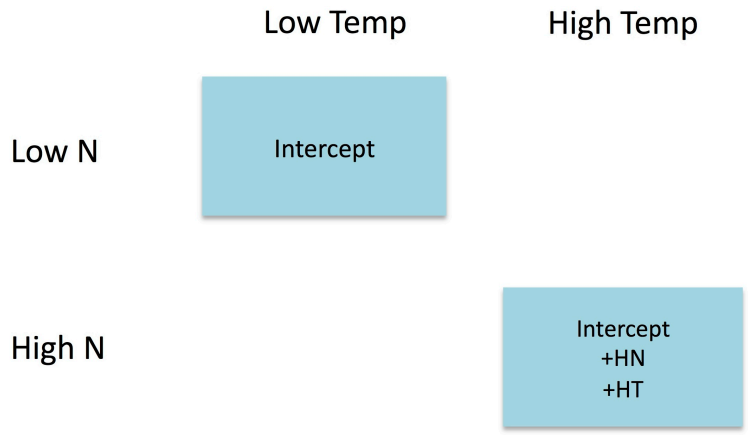


Now we can’t estimate the interaction term, because the treatments do not include high vs. low N at different temperatures, or vice versa. However, we can still estimate the main effects HighN and HighTemp. That is because we still have variation among the treatments that lets us ask ‘what is the effect of N, holding temperature constant’, and vice versa. Note that this is the case no matter which three treatment combinations we have, e.g.:



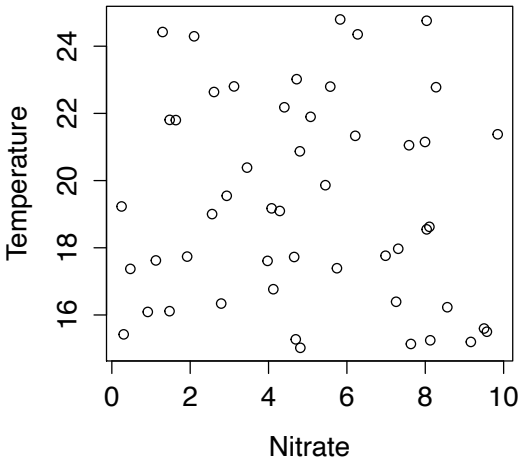
this still lets us estimate the two main effects. Biologically it may be suspect, because we might think the interaction is important to measure, but statistically this is legit.

What happens if we lose another treatment combination?



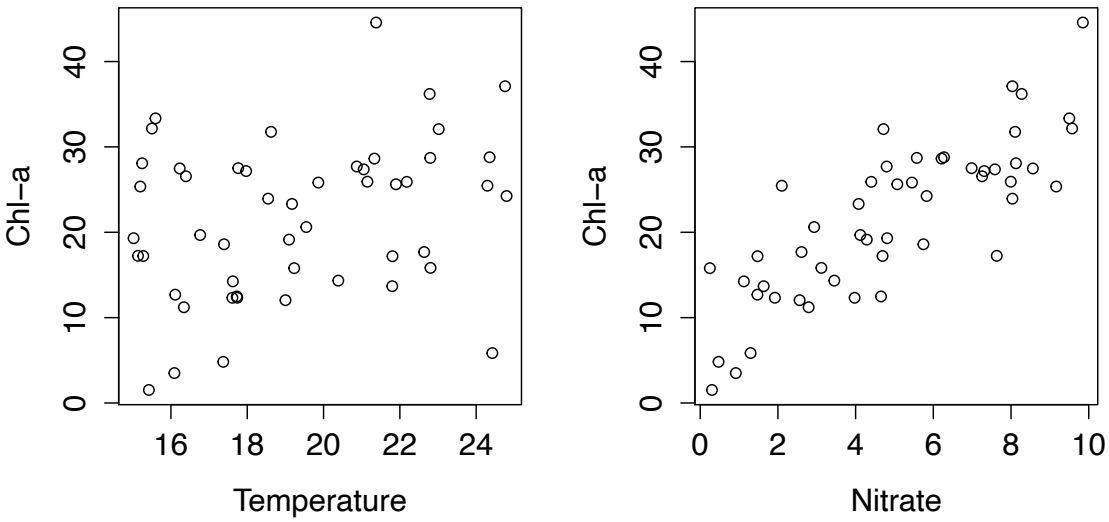
Now we have a problem. We'd like to estimate the effect of N, and the effect of temperature, but we can't because they are totally confounded. We can't tell whether the difference between High Temp + High N, and Low Temp + Low N, is due to N or to temperature. In statistical lingo these two predictors are perfectly *collinear*. So not only is this experimental design biologically weird, it is statistically unfittable.

Now that we understand what model coefficients mean for categorical factors and their interaction, let's apply similar reasoning to observational data. Maybe we want to see if phytoplankton abundance (approximated as Chl-a) is predicted by temperature and nitrogen (e.g. nitrate) in the ocean, using a bunch of samples compiled from various cruises. First we should look at our two predictors variables, to see how well we can separately estimate their effects. Ideally they would look something like this:

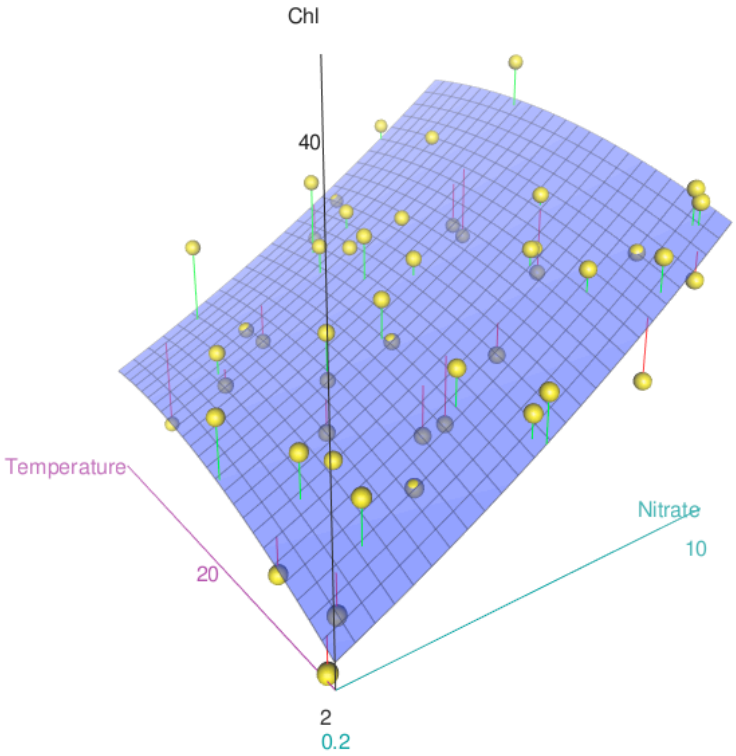


In this case there is a full range of combinations of both variables. So we can see what is the effect of nitrate while holding temperature constant, and vice versa, and we can see whether the effect of nitrate changes as temperature changes, and vice

versa. Now let's look at some simulated data for how chlorophyll might respond to temperature and nitrate:



Both plots show a positive relationship, though the correlation with temperature is weak. We can also plot this as a 3D scatterplot, with the function scatter3d():



This function also fits a quadratic surface to help you see any patterns. It looks like both variables cause chlorophyll to increase, and maybe there is an interaction between the variables. Let's fit a linear model with an interaction to the data:

```
Call:
lm(formula = chlorophyll ~ temperature * nitrate)

Residuals:
     Min       1Q   Median       3Q      Max
-10.1905  -3.1169   0.5854   2.7280   7.3873

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -8.55227    8.18351  -1.045   0.3015
temperature        0.84589    0.42606   1.985   0.0531 .
nitrate            1.93105    1.36724   1.412   0.1646
temperature:nitrate  0.04523    0.07253   0.624   0.5360
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.232 on 46 degrees of freedom
Multiple R-squared:  0.7979,    Adjusted R-squared:  0.7848
F-statistic: 60.55 on 3 and 46 DF,  p-value: 5.249e-16
```

Looks like none of the coefficients are significant, although temperature is nearly so. But our scatterplots clearly showed some patterns, plus the R^2 for the whole model is pretty large (~0.8). What's going on here? This is where it becomes important that the meaning of the main effects, 'temperature', and 'nitrate', change when the interaction 'temperature:nitrate' is in the model. What 'temperature' is quantifying here is 'what is the effect of temperature, when holding nitrate and temperature*nitrate constant?'. This idea of varying temperature while holding temperature*nitrate constant is pretty confusing, but it is possible if nitrate=0, because then temperature*nitrate equals zero regardless of what value temperature has. So the parameter 'temperature' tells you the effect of temperature when nitrate is zero. Likewise the parameter 'nitrate' tells you the effect of nitrate when temperature is zero. Neither of these is likely to be of much biological interest. In addition, none of the temperature data are anywhere near zero, and the nitrate data are all above zero, so the model doesn't have much information about the region of parameter space that these terms represent. However, the estimation of these parameters is necessary for fitting the linear model with the interaction.

How can we make these results more sensible? This is a case where standardizing the predictors would be a good idea, i.e. subtracting the mean from both temperature and nitrate. Then the parameter nitrate would estimate 'what is the effect of nitrate when temperature is at its mean?', and vice versa. But in this case the interaction is not significant, so we can just remove it from the model. If it were significant, then we would focus on that term and visualizing what it means (plotting with the effects package is useful). But in this case we should remove the interaction from the model, so that the model parameters are interpretable in terms of the main effects:

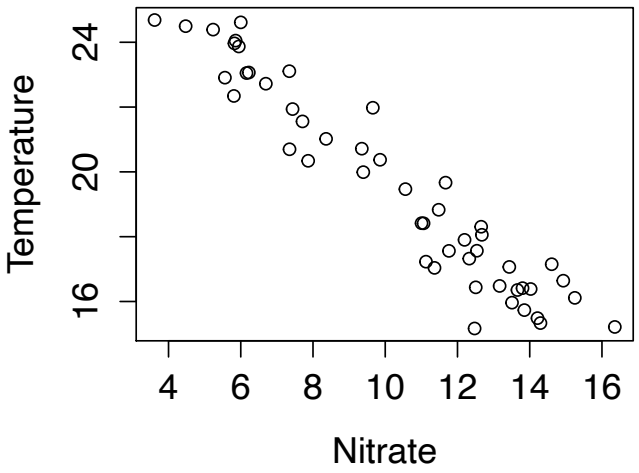
```
Call:
lm(formula = chlorophyll ~ temperature + nitrate)

Residuals:
     Min       1Q   Median       3Q      Max
-11.1748  -3.1328   0.8243   2.6383   7.2869

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -12.9545     4.1115  -3.151  0.00283 **
temperature   1.0804     0.1989   5.432 1.93e-06 ***
nitrate       2.7727     0.2163  12.817  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.204 on 47 degrees of freedom
Multiple R-squared:  0.7962,    Adjusted R-squared:  0.7876
F-statistic: 91.82 on 2 and 47 DF,  p-value: < 2.2e-16
```

Without the interaction the effects of temperature and nitrate are both highly significant. And like I just explained, they don't mean the same thing as they did I the previous model. Now they mean 'what is the effect of temperature while holding nitrate constant', and vice versa. It is also worth comparing to a model with just temperature:

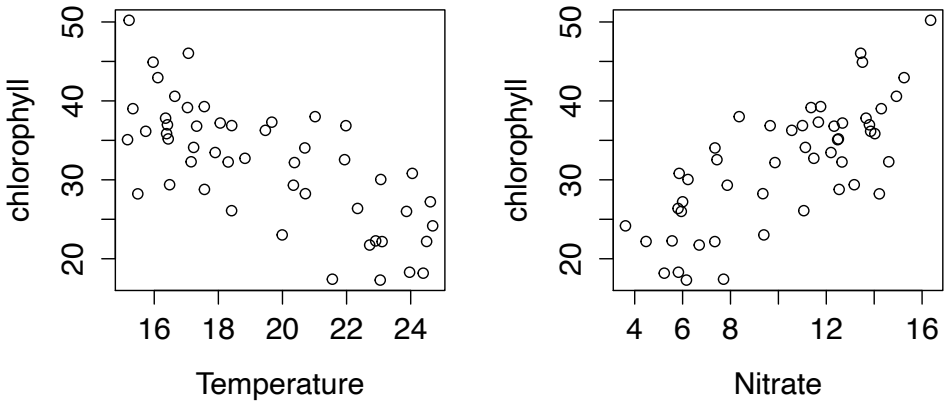
```
Call:
lm(formula = chlorophyll ~ temperature)

Residuals:
     Min       1Q   Median       3Q      Max
-20.2206  -6.7715   0.8612   6.3595  21.1588

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.7600     8.1238   0.586   0.5607
temperature   0.8726     0.4159   2.098   0.0412 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.821 on 48 degrees of freedom
Multiple R-squared:  0.08401,   Adjusted R-squared:  0.06493
F-statistic: 4.403 on 1 and 48 DF,  p-value: 0.04117
```

The coefficient for temperature is somewhat smaller, and its standard error and p-value are much larger. Why is that? From the plots we know that nitrate explains a lot of variation in chlorophyll, and when that variation is accounted for with a 'nitrate' parameter, then the effect of temperature has much less noise around it. So this is an important point: *including multiple terms in a model can make the detection of each individual term easier*.

Now let's look at an example where multiple terms in a model is problematic. The example I just gave had two explanatory variables (temperature and nitrate) that were uncorrelated. This is the direct analogue of an orthogonal experimental design, and is the optimal situation for a multiple regression. Often the predictors in a model are somewhat correlated with each other, and this can cause problems. For example, in the fake dataset I made above, temperature and nitrate were uncorrelated, but in the ocean they are often correlated due to stratification (high temp = high stratification = low nutrient supply). So the relationship among the predictors might look more like this:



So this situation is like the example in the experiment where there was just two treatment combinations: the variables are highly collinear. They are not perfectly collinear, but the model is going to have a hard time estimating the effect of nitrate while holding temperature constant, and vice versa, because there is not much independent variation in the predictors. I took these predictors and simulated chlorophyll data using the same exact parameters as I used in the previous example:



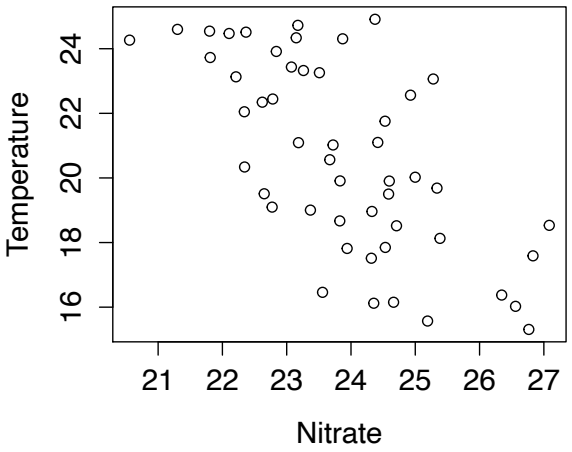
Uh-oh, chlorophyll increases with nitrate, but it appears to decrease with temperature. This is despite the fact that I generated this data from a model where chlorophyll increases with temperature! The problem is that the effect of nitrate is stronger than the effect of temperature (that's how I coded it), and temperature is negatively correlated with nitrate, so it makes it look like chlorophyll declines with temperature. A multiple regression will estimate both effects at once, so it is possible it will still get the correct answer:

```
Call:
lm(formula = chlorophyll ~ temperature + nitrate)

Residuals:
     Min       1Q   Median       3Q      Max
-10.4080  -3.3559   0.4297   3.5808   9.2459

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 16.24556   24.08314   0.675   0.5033
temperature -0.05511    0.84518  -0.065   0.9483
nitrate      1.63479    0.76200   2.145   0.0371 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.184 on 47 degrees of freedom
Multiple R-squared:  0.5601,    Adjusted R-squared:  0.5413
F-statistic: 29.92 on 2 and 47 DF,  p-value: 4.168e-09
```

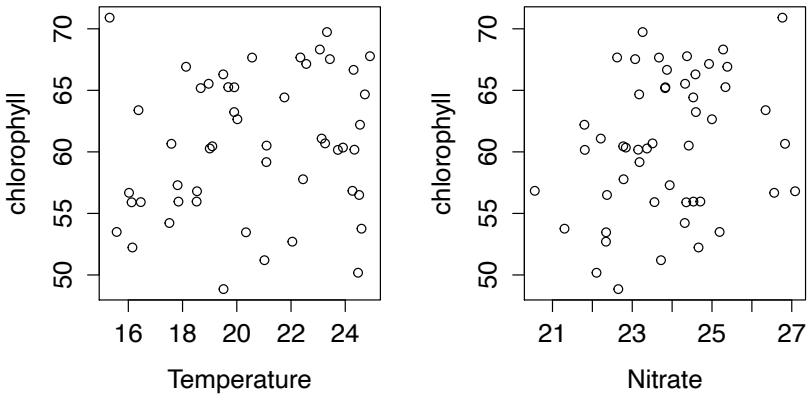
The model says that there is a positive effect of nitrate, and no effect of temperature. So at least we didn't get a result for temperature that was in the wrong direction, but we also don't have enough information in the data to estimate the correct temperature effect. This example illustrates a challenge of working with observational data, namely that *correlated predictors will tend to obscure each other's effects*. In this case we probably shouldn't have made this model at all, after

looking at the scatterplot of temperature vs. nitrate. However, the multiple regression is still better than a simple regression of chlorophyll vs. temperature, which would tell us the wrong answer. To get a more accurate answer we would need to collect data from sites where temperature and nitrate are more decoupled (e.g. equatorial upwelling sites), or do a manipulated experiment to separate out the effects of temperature and nitrate. A similar phenomenon will happen if predictors are positively correlated. For example, maybe phytoplankton are limited mostly by nitrate, but nitrate and phosphate tend to be positively correlated in the ocean. If we made a regression with both nitrate and phosphate, the model would have a hard time distinguishing the separate effects of the two predictors.

I don't want this example to give the impression that you can't use correlated predictors in a model. It is often inevitable that predictors are somewhat correlated, and if the correlation is not too strong then including multiple predictors will actually yield better results. For example, if temperature and nitrate are more weakly correlated:



Then the raw data may show no pattern with chlorophyll:



But the regression correctly detects the effects of both variables:

```
Call:
lm(formula = chlorophyll ~ temperature + nitrate)

Residuals:
     Min       1Q   Median       3Q      Max
-10.2908  -3.5630   0.7421   2.3345   9.1052

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -27.2178    20.1703  -1.349 0.183674
temperature   1.1985     0.3168   3.783 0.000438 ***
nitrate       2.6402     0.6374   4.142 0.000142 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.871 on 47 degrees of freedom
Multiple R-squared:  0.2886,    Adjusted R-squared:  0.2584
F-statistic: 9.535 on 2 and 47 DF,  p-value: 0.0003342
```

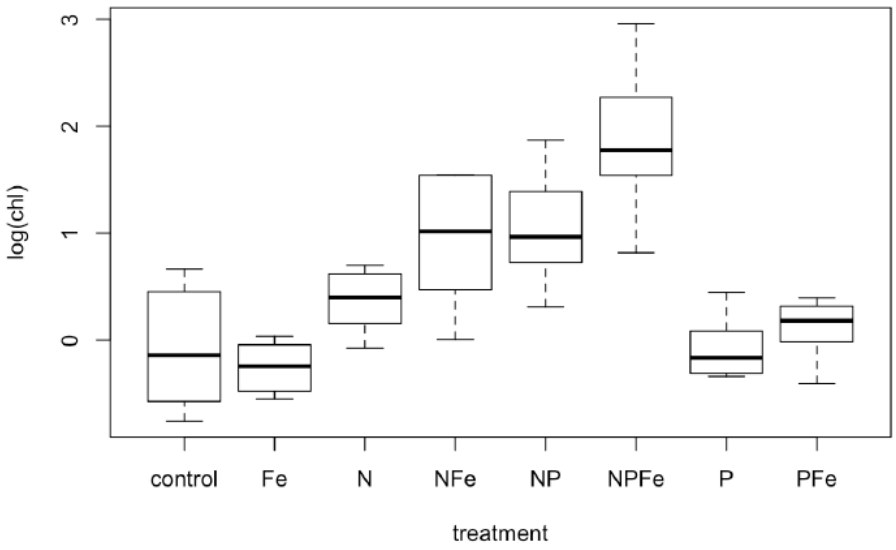
So my general advice would be: for weak-moderate correlation among predictors, including multiple predictors is OK but keep an eye on whether the results change a lot when predictors are add/removed from the model, and think about what this might be indicating. For strong correlation (e.g., $r > 0.7$), including multiple predictors is probably going to be problematic. Instead you can do separate analyses with different predictors, and think about what the results mean, or you can combine predictors into composite variables with something like principal coordinates analysis, which we will discuss later.

Contrasts and multiple comparisons

The first topic in this lecture was essentially about the correlation-causation problem – in the absence of experimental manipulation, it is challenging to establish causal relationships between predictors and response variables, and if the

predictors are mutually correlated (or correlated with an unobserved driver), then we can even get answers that are in the wrong direction. Now we will focus on a different inference problem that applies equally to experimental and observational studies – how to make many statistical comparisons while accounting for the fact that 'significant' results may be due to chance.

When we have a factor in a model that has more than two levels or groups, we may be interested in comparing specific groups. For example, imagine that we want to test whether phytoplankton production is limited or co-limited by different nutrients, and so we perform a series of incubations where we add potentially limiting nutrients to community samples in bottles. Imagine that we add nitrogen (N), phosphorus (P), or iron (Fe), as well as N+P, N+Fe, P+Fe, and N+P+Fe. We also have control bottles with no nutrients added. Our response variable is chlorophyll concentration after 24 hours, and the data look like this:

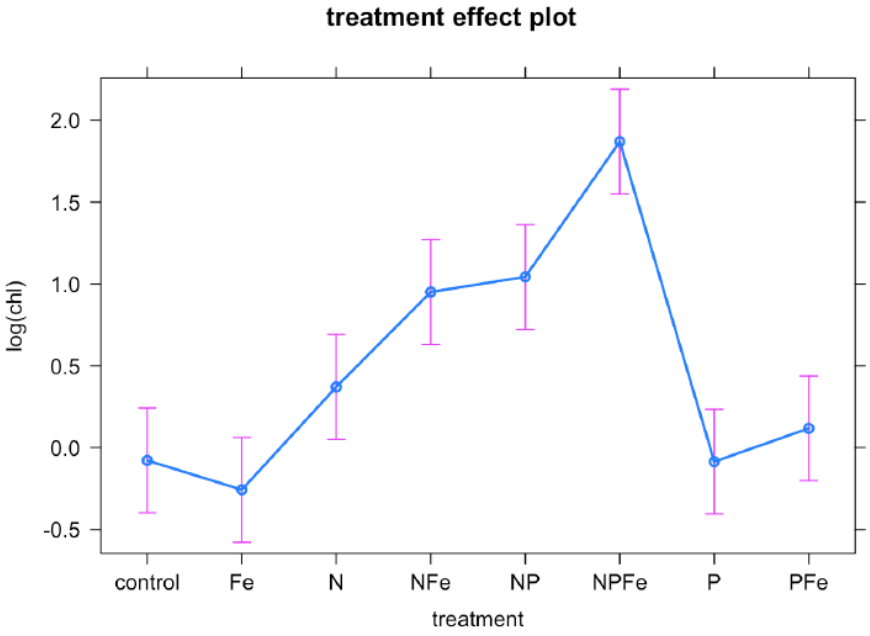


Using this data, we want to make a statistical model and ask a number of questions. We want to know whether any nutrient is limiting, whether one nutrient is more limiting than others, and whether pairs of nutrients are co-limiting. If we make a linear model with the formula log(chl) ~ treatment, and perform a standard F-test on the treatment factor, we get these results:

```
mod = lm(log(chl) ~ treatment, data = chl.data)
Anova(mod)
```

```
## Anova Table (Type II tests)
##
## Response: log(chl)
##           Sum Sq Df F value   Pr(>F)
## treatment 30.309  7  21.169 1.25e-13 ***
## Residuals 11.454 56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(allEffects(mod))
```



This F-test is a *global* test, in the sense that it quantifies all variation explained by the treatment factor, and asks whether treatment explains more variation than expected by chance. If we want to perform further tests, such as whether the N treatment has more chlorophyll than the control (evidence of N limitation), or whether the N+Fe treatment has more chlorophyll than the N treatment (evidence that Fe becomes limiting once N is added), we need to perform further comparisons.

To test whether a particular pair of treatments, such as control vs. N addition, have different means, we can perform a null hypothesis test called a *contrast*. Contrasts can also involve more than two groups (we will see examples in the homework assignment).

What contrasts should we perform? There are various philosophical approaches to hypothesis testing using contrasts, which I will not describe in detail here. From my perspective the important guidelines are (1) focus on hypotheses you are interested

in, (2) avoid creating a proliferation of tests if possible (but sometimes this is not possible), (3) keep in mind that p-value thresholds like 0.05 are just rules of thumb, not magical boundaries, (4) remember that effect sizes (the actual magnitude of differences) are just as important as p-values (which only tell you whether a particular difference is unlikely to be due to chance).

We can calculate contrasts of our choosing using the handy package 'emmeans'. The name of the package refers to *estimated marginal means*, which have also been referred to as least square means in the literature. Estimated marginal means are just the means for each group/level of a factor, calculated while holding any covariates constant at their mean value (or some other predefined value). These are the same as the 'effects' we have been plotting with the effects or ggeffects packages. So, different names proliferate, but the bottom line is that we want to plot and test the whether our model predicts that groups are different from each other.

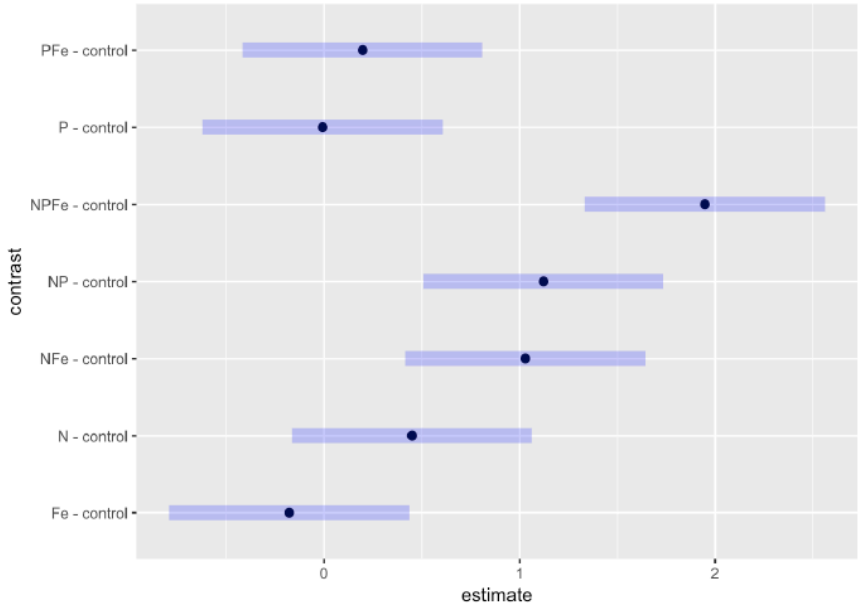
For example, if we want to compare all nutrient amendments to the control treatment, there is a built in option to do this:

```
trt.control.contrast = emmeans(mod, specs = trt.vs.ctrl ~ treatment)
trt.control.contrast

## $emmeans
##  treatment      emmean        SE df    lower.CL   upper.CL
##  control   -0.07817387 0.1598982 56 -0.39848853 0.24214079
##  Fe        -0.25750319 0.1598982 56 -0.57781785 0.06281147
##  N          0.37076561 0.1598982 56  0.05045095 0.69108026
##  NFe        0.95125675 0.1598982 56  0.63094209 1.27157141
##  NP         1.04337669 0.1598982 56  0.72306203 1.36369135
##  NPFe       1.86985033 0.1598982 56  1.54953567 2.19016499
##  P         -0.08571840 0.1598982 56 -0.40603306 0.23459626
##  PFe        0.11791477 0.1598982 56 -0.20239989 0.43822943
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate        SE df t.ratio p.value
##  Fe - control   -0.17932932 0.2261303 56  -0.793  0.8972
##  N - control     0.44893947 0.2261303 56   1.985  0.2401
##  NFe - control   1.02943062 0.2261303 56   4.552  0.0002
##  NP - control    1.12155056 0.2261303 56   4.960  <.0001
##  NPFe - control  1.94802420 0.2261303 56   8.615  <.0001
##  P - control    -0.00754453 0.2261303 56  -0.033  1.0000
##  PFe - control   0.19608864 0.2261303 56   0.867  0.8677
##
## Results are given on the log (not the response) scale.
## P value adjustment: dunnettx method for 7 tests
```

The output first shows us the estimated marginal means, and their standard errors, degrees of freedom, and confidence intervals. Then it shows us contrasts where the control treatment is subtracted from each nutrient treatment. E.g., 'Fe – control' is the mean log(chl) in the Fe treatment minus the mean log(chl) in the control

treatment, and the value of that contrast is -0.18. The SE and degrees of freedom for that contrast is used to calculate a t-test, and that is where the p-value (0.8972) comes from. So for this particular contrast, the observed difference between treatments could very easily be due to chance, and we have little evidence that there is iron limitation (or more precisely, iron limitation without co-limitation by other nutrients). If we want to plot the contrast themselves, as opposed to the group means, we can use the plot() function from emmeans:

```
plot(trt.control.contrast$contrast)
```



Several of the contrasts, particularly those involving both N and Fe or N and P, are clearly different from zero and have very small p-values. When comparing all the contrasts it seems like adding N alone may increase chl concentration, but in this experiment the difference is not large enough to be confidently different from chance. However, adding both N and another nutrient results in a larger, detectable difference in chl, suggesting that multiple nutrients co-limit production, or that alleviating N limitation quickly results in limitation by other nutrients.

Now let's look again at the output from emmeans and note the line highlighted in red:

```
trt.control.contrast = emmeans(mod, specs = trt.vs.ctrl ~ treatment)
trt.control.contrast
```

```
## $emmeans
##  treatment      emmean        SE df    lower.CL   upper.CL
##  control   -0.07817387 0.1598982 56 -0.39848853 0.24214079
##  Fe        -0.25750319 0.1598982 56 -0.57781785 0.06281147
##  N          0.37076561 0.1598982 56  0.05045095 0.69108026
##  NFe        0.95125675 0.1598982 56  0.63094209 1.27157141
##  NP         1.04337669 0.1598982 56  0.72306203 1.36369135
##  NPFe       1.86985033 0.1598982 56  1.54953567 2.19016499
##  P         -0.08571840 0.1598982 56 -0.40603306 0.23459626
##  PFe        0.11791477 0.1598982 56 -0.20239989 0.43822943
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate        SE df t.ratio p.value
##  Fe - control   -0.17932932 0.2261303 56  -0.793  0.8972
##  N - control     0.44893947 0.2261303 56   1.985  0.2401
##  NFe - control   1.02943062 0.2261303 56   4.552  0.0002
##  NP - control    1.12155056 0.2261303 56   4.960  <.0001
##  NPFe - control  1.94802420 0.2261303 56   8.615  <.0001
##  P - control    -0.00754453 0.2261303 56  -0.033  1.0000
##  PFe - control   0.19608864 0.2261303 56   0.867  0.8677
##
## Results are given on the log (not the response) scale.
## P value adjustment: dunnettx method for 7 tests
```

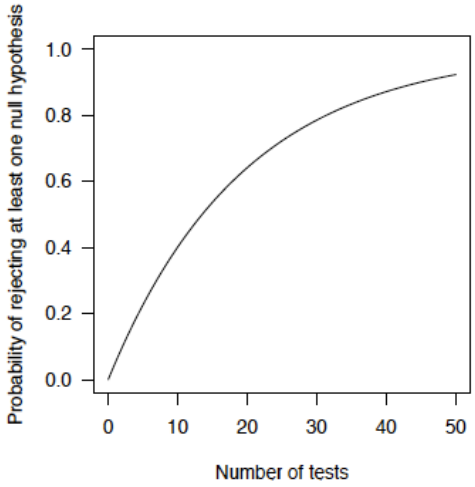
By default, the emmeans function has adjusted the p-values of our contrasts, using the dunnettx methods for 7 tests. What does this mean, and why have the p-values been adjusted? To understand this we need introduce the concept of a *family-wise error rate*.

Multiple comparisons and the family-wise error rate

Let's what review what a p-value means. For a particular test, such as the 'N – control' contrast, the p-value is *the probability that we would see a contrast this large, or larger, if the true difference between treatments is actually zero*. For this to make sense, we have to imagine that our experiment is just one instance out of an endless universe of possible experiments. Each time we perform an experiment, we are sampling chlorophyll values from a certain number of bottles, and the chlorophyll concentration varies across bottles for a variety of reasons that we can't control. Sometimes, by chance, that variation will cause the control and nitrogen addition treatments to have significantly different means, even if the experimental treatment is *not* the cause of the difference. The p-value tells us how likely it is that a particular difference could be due to chance. If we encounter a p-value that, by chance, exceeds a significance threshold, this will lead us to 'reject the null hypothesis', where the null hypothesis is that the treatments are the same. This paradigm for intepreting statistical results is commonly called *frequentist*, because we are thinking about the *frequency* of a particular result in an imagined universe of repeated experiments. I often refer to 'rejection of the null, even though it is true'

as a 'false positive', because the situation is analogous to getting tested for something like COVID-19 and getting a positive test result even though you are uninfected.

Now let's consider the fact that we are performing multiple hypothesis tests at the same time, such as the treatment vs. control contrasts performed earlier. If we perform k tests then we will find a false positive *at least once* with probability $\pi = 1 - (1 - 0.05)^k$. This is the *family-wise error rate* (FWER): the probability that we will make at least one error (treating a false positive as a real difference), when conduct a whole family of tests. This plot shows how FWER increases as you perform more tests:



Here the x axis is k and the y axis is π or FWER. The important thing to note here is that *if you perform a large number of tests, it is almost guaranteed that at least one will be a false positive*. And even for more modest numbers of tests it is more likely than not that you will get a false positive. For example, if you perform 10 tests then there is a ~40% chance that one will return a false positive, and if you perform 20 tests there is a ~65% chance.

Now here is the tricky and ultimately philosophical question: what should we do about this?

One approach is to adjust p-values to control the FWER. For example, if you are using $p = 0.05$ as a threshold to call something 'significant', but you do 10 tests, then you can make all of the p-values larger, so that there is only a 5% chance that at least one of the tests is a false positive. The Dunnett method used by the emmeans function is a standard FWER correction for the case when you are comparing multiple treatments to a control.

Is controlling the FWER the best approach? This is a debated question for which I can't give a consensus answer. One thing to know is that many statisticians don't

like it when p-values are arbitrarily divided into 'significant' and 'non-significant' by an arbitrary threshold. The p-value is a continuous number, and it matters if it is extremely small vs. 0.045. In addition, the p-value itself will change if you repeat an experiment, for the same reason that false positives sometimes occur – the phenomenon you are trying to measure is variable for reasons you can't control. So an alternative approach to adjusting p-values is to *adjust your expectations*. Be mindful of the fact that any pattern you see could be due to chance, and be extra mindful of the fact that if you make a large number of comparisons, you are likely to encounter false positives. If you really have 10 hypotheses to test, then there is a 40% chance you will get a false positive. That's just the reality of trying to measure how the world works. We will think some more about false positives and false negatives in the next lecture, when we learn about simulation.

What do the results of the nutrient experiments look like without p-value adjustment? We can see this by changing the adjustment option in emmeans:

```
trt.control.contrast.noadjust = emmeans(mod, specs = trt.vs.ctrl ~ treatment,
adjust = 'none')
trt.control.contrast.noadjust
```

```
## $emmeans
##  treatment      emmean        SE df    lower.CL   upper.CL
##  control   -0.07817387 0.1598982 56 -0.39848853 0.24214079
##  Fe        -0.25750319 0.1598982 56 -0.57781785 0.06281147
##  N          0.37076561 0.1598982 56  0.05045095 0.69108026
##  NFe        0.95125675 0.1598982 56  0.63094209 1.27157141
##  NP         1.04337669 0.1598982 56  0.72306203 1.36369135
##  NPFe       1.86985033 0.1598982 56  1.54953567 2.19016499
##  P         -0.08571840 0.1598982 56 -0.40603306 0.23459626
##  PFe        0.11791477 0.1598982 56 -0.20239989 0.43822943
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate        SE df t.ratio p.value
##  Fe - control   -0.17932932 0.2261303 56  -0.793  0.4311
##  N - control     0.44893947 0.2261303 56   1.985  0.0520
##  NFe - control   1.02943062 0.2261303 56   4.552  <.0001
##  NP - control    1.12155056 0.2261303 56   4.960  <.0001
##  NPFe - control  1.94802420 0.2261303 56   8.615  <.0001
##  P - control    -0.00754453 0.2261303 56  -0.033  0.9735
##  PFe - control   0.19608864 0.2261303 56   0.867  0.3896
##
## Results are given on the log (not the response) scale.
```

If we compare the p-values to the contrasts to what we obtained before, using the Dunnett correction, then three of them are still very small (NFe – control, NP – control, and NPFe – control), and three of them are still large (Fe – control, P – control, and PFe – control). The only one that has changed in a potentially meaningful way is 'N – control': with the correction p = 0.24, and without the correction p = 0.052. This gives you a good sense for when adjustments will have an important effect – for tests where there is not exceedingly strong evidence in the

first place. At the same time, the magnitude of the adjustment will depend on the number of tests. If you perform 10,000 tests then the adjustment for FWER will be extreme for all p-values.

Another common form of contrast is one where you compare all possible pairs of treatments to each other. This is often called a 'post hoc' test because it does not focus on particular hypotheses that you had in advance. Instead, you are deciding after the fact (post hoc means 'after the fact' in Latin) to see which treatments are significantly different. This is what it looks like for the nutrient experiment example:

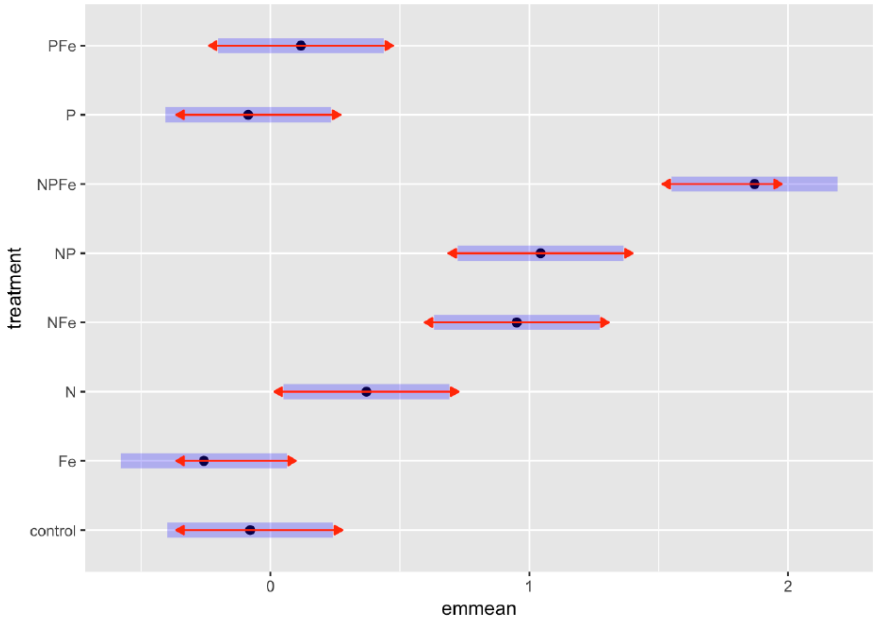
```
pairwise.posthoc = pairs(emmeans(mod, "treatment"))
pairwise.posthoc
```

```
##  contrast          estimate        SE df t.ratio p.value
##  control - Fe    0.17932932 0.2261303 56   0.793  0.9928
##  control - N    -0.44893947 0.2261303 56  -1.985  0.5007
##  control - NFe  -1.02943062 0.2261303 56  -4.552  0.0007
##  control - NP   -1.12155056 0.2261303 56  -4.960  0.0002
##  control - NPFe -1.94802420 0.2261303 56  -8.615  <.0001
##  control - P     0.00754453 0.2261303 56   0.033  1.0000
##  control - PFe  -0.19608864 0.2261303 56  -0.867  0.9878
##  Fe - N         -0.62826880 0.2261303 56  -2.778  0.1214
##  Fe - NFe       -1.20875994 0.2261303 56  -5.345  <.0001
##  Fe - NP        -1.30087988 0.2261303 56  -5.753  <.0001
##  Fe - NPFe      -2.12735352 0.2261303 56  -9.408  <.0001
##  Fe - P         -0.17178479 0.2261303 56  -0.760  0.9945
##  Fe - PFe       -0.37541796 0.2261303 56  -1.660  0.7120
##  N - NFe        -0.58049115 0.2261303 56  -2.567  0.1900
##  N - NP         -0.67261108 0.2261303 56  -2.974  0.0770
##  N - NPFe       -1.49908473 0.2261303 56  -6.629  <.0001
##  N - P           0.45648401 0.2261303 56   2.019  0.4790
##  N - PFe         0.25285083 0.2261303 56   1.118  0.9501
##  NFe - NP       -0.09211994 0.2261303 56  -0.407  0.9999
##  NFe - NPFe     -0.91859358 0.2261303 56  -4.062  0.0036
##  NFe - P         1.03697515 0.2261303 56   4.586  0.0006
##  NFe - PFe       0.83334198 0.2261303 56   3.685  0.0113
##  NP - NPFe      -0.82647364 0.2261303 56  -3.655  0.0124
##  NP - P          1.12909509 0.2261303 56   4.993  0.0002
##  NP - PFe        0.92546192 0.2261303 56   4.093  0.0033
##  NPFe - P        1.95556873 0.2261303 56   8.648  <.0001
##  NPFe - PFe      1.75193556 0.2261303 56   7.747  <.0001
##  P - PFe        -0.20363317 0.2261303 56  -0.901  0.9848
##
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 8 estimates
```

This is a lot of contrasts! Is this kind of all-possible-pairs comparison really useful or necessary? This is another question that is hard to answer and will depend on the specifics of your study. In general I would avoid it if you can satisfactorily address your hypotheses without it, just because the proliferation of tests means false positives are pretty likely. In the case of this particular experiment, the most important comparisons are whether the treatments are different from the controls, and whether adding a second nutrient is different from the first nutrient – for example, is N+Fe different from N or Fe alone? This is still a large number of comparisons, but it is at least fewer than looking at all possible pairs. In addition, you may be satisfied to say 'I did a global F-test, and there is significant variation among treatments, and by looking at the treatment effects and confidence intervals I can interpret how the different nutrients affect productivity'. This is common

advice from people who don't like the obsession with p-values: focus on effect sizes and confidence intervals instead. Nonetheless, the reviewers of your manuscript may request p-values.

If you do choose to compare all pairs, emmeans has a nice plot to do this visually:



Here the blue bars are confidence intervals of the emmeans, and the red arrows show which treatments are different from each other: if the red arrows don't overlap, then the treatments are different at a 0.05 level (with a Tukey correction by default).

Custom contrasts

emmeans has many nice functions for computing common contrasts, but it can be helpful to specify custom contrasts of your choosing. For example, if we don't want to compare all pairs, but we do want to compare the N+P treatment to the N treatment, then we need to set up a contrast where the mean for the first treatment gets a multipler of '1', the mean for the second treatment gets a multiplier of '-1', and all other treatments get multipliers of '0'. The contrast() function will then multiply each treatment mean by the specified multipliers, add them up, and calculate whether they are different from zero. To make the vector of multipliers, refer to the order of the emmeans returned by emmeans(), which is the order of the levels of the factor:

```
> levels(chl.data$treatment)
[1] "control" "Fe"      "N"       "NFe"     "NP"      "NPFe"    "P"
     "PFe"
> N.colimitation = contrast(emmeans(mod, specs = ~ treatment), method
 = list("NP - N" = c(0,0,-1,0,1,0,0,0), "NFe - N" = c(0,0,-1,1,0,0,0,
0)))
> N.colimitation
 contrast estimate    SE df t.ratio p.value
 NP - N      0.673 0.226 56   2.974  0.0043
 NFe - N     0.580 0.226 56   2.567  0.0130
```

So what does this mean? The first contrast, which I named "NP – N", is equal to:

(0)*control + (0)*Fe + (-1)*N + (0)*Nfe + (1)*NP + (0)*NPFe + (0)*P + (0)*Pfe

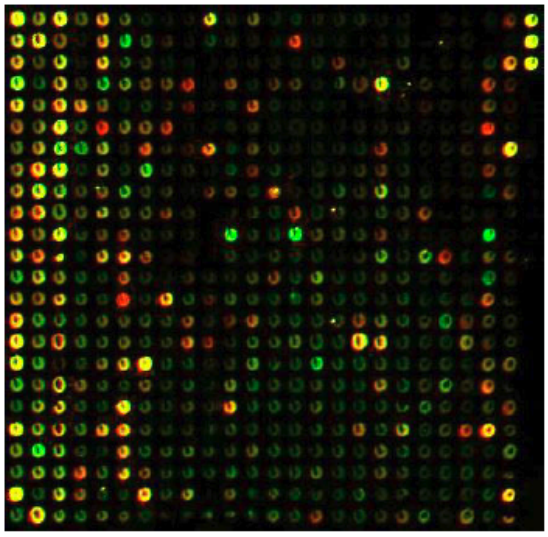
= (-1)*N + (1)*NP, or equivalently, NP – N.

We have specific multipliers for each of the treatment means because we can create a custom contrast from any linear combination of the treatment means. In the homework assignment we will work on a more complex example, but in this case I've just set up the multipliers so that the contrast is testing whether the N+P treatment has more chlorophyll than the N treatment. Why am I testing this? If the community shows a greater response to N+P than N, it means that either P becomes limiting once N is added, or that they are co-limiting (where we only see an increase in chlorophyll when both are added). The second contrast, names "NFe – N", is asking the same question for nitrogen and iron. Both of the contrasts are significantly different from zero. We also saw previously that the N treatment has modestly more chlorophyll than the control treatment, but it is not very clear whether this is a real difference or not. So we could interpret these results to mean that N may be the primary limiting nutrient, but once it becomes more available the community quickly becomes limited by P and/or Fe. Does it make sense that adding P or Fe after adding N can increase chlorophyll? Maybe – this could happen if different members of the phytoplankton community are limited by different nutrients, due to difference physiological strategies.

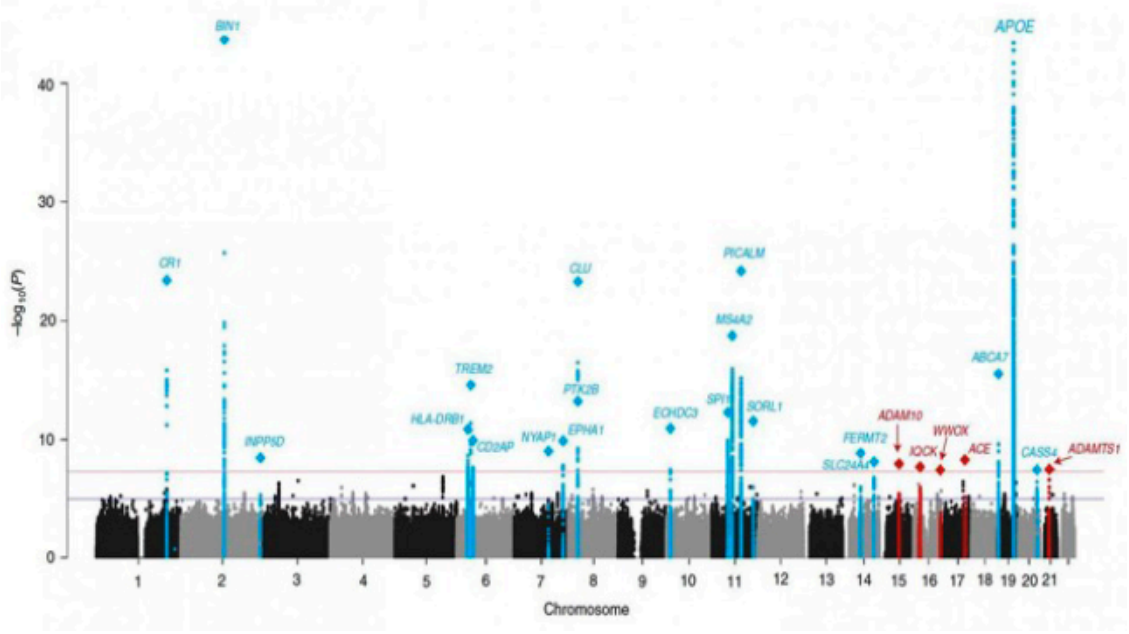
Here's another interesting thing about this example. The data are from a simulation, where I defined in advance what the true treatment means are, and then 'observed' replicates of the treatments where drawn randomly using those means and a pre-specified residual variation. When I chose the true means I made the contrast "N – control" the same as "NP – N" and "Nfe – N". However, due to the variability inherent in the experiment these three contrasts ended up with p-values of 0.052, 0.0043, and 0.013. This is important: variability among experimental replicates doesn't just lead to unexplained variation, it also makes p-values variable as well!

Multiple comparisons with vast numbers of comparisons

To finish this topic let's consider cases where the nature of the study requires us to make a very large number of comparisons/contrasts. One of the first areas where this problem arose was in the use of microarrays to look at transcriptional differences between experimental treatments.



Genomes have many genes, and these experiments are interested in which genes exhibit changes in their expression level (mRNA concentration) under different conditions. To answer this question we inevitably have to do a very large number of tests. Based on the formula for the family-wise error rate I introduced earlier, we know that with a large number of tests there will be false positives. So how can we attempt to distinguish real experimental differences from noise? The problem is not specific to microarrays. If one wants to correlate allele frequencies with phenotypes (GWAS), or correlate metagenome gene functions with environmental conditions, or correlate microbial OTU frequencies with environmental conditions, the same issue arises. This figure illustrates the problem:



On the x-axis is the position of a genomic locus, and on the y-axis is $-\log_{10}$(P value) for the difference at that locus between treatments. So, a high value on the y-axis means a small p-value. The horizontal lines show different p-value cutoffs that are derived using different criteria for distinguishing real results from false positives. Loci with very small p-values are less likely to be false positives, but nonetheless we will never truly know which effects are real and which are not from a single study. Replicating the study and combining this design with other approaches will be necessary.

If we perform such a study, what is the best way to sort out all the p-values and decide which loci to focus on? A popular method has been to focus not on FWER but on the *false discovery rate* (FDR). The false discovery rate is the *proportion of discoveries (significant p-values) that are false*. We can decide what we consider to be an acceptable FDR, and then adjust p-values so that this FDR is obtained. For example, if we choose 5% this means that *5% of the significant contrasts will be false positives*. So in the figure above, imagine that we will say that all the p-values above the line are 'discoveries', and in doing so we will acknoweldge that 5% of them are not real, just do to noise.

One reason this method has become popular is because it is not as strict as methods that control the FWER (e.g. the Dunnett method). The downside is that you will have more false positives. So an important aspect of using this method is acknowledging that the results are somewhat exploratory, rather than confirmatory. If a set of potential marker genes are identified using a FDR procedure, it will be important to follow-up and study those genes in more detail to determine whether the effect is real.

For researchers with a more ecological focus, one area where multiple comparisons and the false discovery rate may be important is microbial community analysis. The affordability of sequencing has made it relatively easy to compare microbial community composition or activity across environments or across experimental treatments, but because these communities are often highly diverse, it is important keep false discoveries in mind. For example, I have read a paper where the authors wanted to test which microbial taxa exhibited significant diel cycles. They tested hundreds of taxa and reported which had a diel signal with $p < 0.05$. However, they applied no corrections to control FDR, and in total about 10% of taxa had a diel signal with $p < 0.05$. Just due to chance we would expect that 5% of taxa will have a diel signal with $p < 0.05$ (indeed, that is the definition of $p < 0.05$). So that means the false discovery rate in this study is likely 50%: half of the reported significant effects are likely false!