Week 8: Categorical Variables

Bob O'Hara

Contents

This Week: Categorical Variables, aka Factors

There is a video for this, about half way through. If you are just looking for that, then click here.

So far we have been looking at using continuous variables to explain a response, i.e. regression. This week we are going to change to looking at categorical variables, i.e. variables that can only take a discrete set of values, e.g. "green", "blue", "red"; or "Control" and "treated with a vaccine". This sort of data can be quite common.

A1. Think of three problems, e.g. from practicals you have done, where you might get this sort of data.

The type of model used to fit to this data is usually called an ANOVA model (ANOVA = ANalysis Of VAriance). If there is a single categorical variable, it is called a one-way ANOVA, if there are two it is called a two-way ANOVA. The reason for this terminology is that the models were developed with the approach of trying to partition the variation into different parts of the model, so crop variety might explain some of the variation, and fungicide treatment might explain some more: the aim was to find out what crop varieies grew best, and what was the optimal treatment to apply. Nowadays the same approach is used for a much wider range of problems, often where this partitioning of the variation does not make any sense (it only really makes sense if you have a well designed experiment, otherwise it is not possible to find only one way to partition the variation).

We will see that the way to build models for this sort of data is to, essentially, use multople regression, and to write the design matrix in cunning ways to make the inferences from the models and data that we want.

Part A - All about data

Hoosfield Barley Yield Data

This data comes from Rothamsted, from an experiment started in 1852 that is still running. A lot of the ideas surounding ANOVA models and how to design experiments were developed at Rothamsted (early on by R.A. Fisher). So this has some historical interest, as well as giving us a simple enough data structure to see how to build, fit and interpret these models.

The experiment has been to grow Spring barley on the site (Hoosfield) continuously and to look at the yield (i.e. how much seed is produced). Each plot had one of 4 treatments:

- **Control**: unfertilised control
- **Fertilised**: Fertilised with chemical fertiliser (P, K, Mg, N)
- **Manure**: Fertilised with farmyard manure
- **Stopped**: Fertilised with farmyard manure up to 1871, unfertilised since then

The response is yield, i.e. how much barley was harvested from the field: a higher yield is obviously better. The units are tonnes per hecatare (t/ha). We have data that are means over about 10 years: we will treat these as replicates.

This reads in the data.

```
Yields <- read.csv("https://www.math.ntnu.no/emner/ST2304/2021v/Week08/Hoosfield_Yields.csv")
```

A1. Think of, and write down, a biological question you could ask using these data.

Hint

There are a few possibilities, mostly of the “compare X with Y” variety.

Answers

Does fertiliser improve yields compared to the unfertilised control? Does farmyard manure improve yields compared to the unfertilised control? Does fertiliser improve yields more than farmyard manure? Did continuing farmyard manure improve yields compared to stopping in 1871?

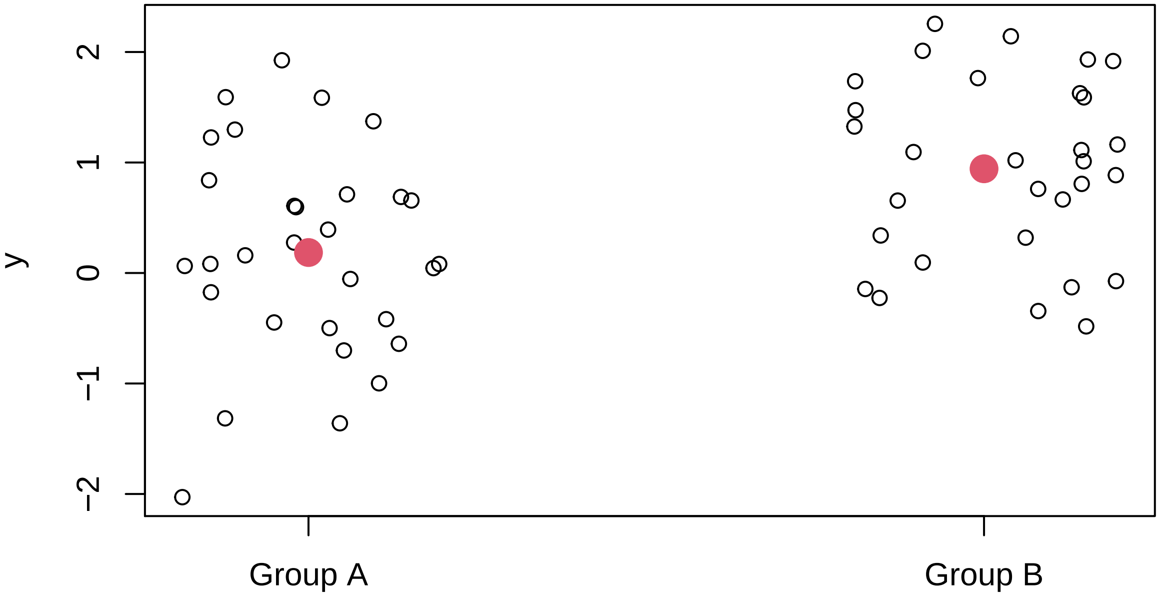
Part B - Analysis

Now we move on to the modelling. We will start by comparing pairs of treatments. We will use a t-test, as you have already come across these. We will then re-write the test as a regression, as a way to get into the more general models.

Here is some (very fake) data:

```r
x <- rep(c("A", "B"), each=30)
xIsB <- as.numeric(x=="B") # use this to make different means
y = rnorm(length(x), xIsB, 1)
# Put the two groups into 2 vectors
yA <- y[x=="A"]; yB <- y[x=="B"];

plot(jitter(xIsB), y, xaxt="n", xlab="")
points(c(0,1), pch=16, c(mean(yA), mean(yB)), cex=2, col=2)
axis(1, c("Group A", "Group B"), at=c(0,1), las=1)
```



We want to comapre the means, the red dots. The difference between the means is $D = \mu_A - \mu_B$, which we estimate with $\hat{D} = \hat{\mu}_A - \hat{\mu}_B = \bar{y}_A - \bar{y}_B$. But we need to know the uncertainty in this estimate.

We can get to it by assuming that our data are normally distributed with different means and the same variance, i.e. $y_i^A \sim N(\mu^A, \sigma^2)$ and $y_j^B \sim N(\mu^B, \sigma^2)$. From this, it turns out that $\hat{D}$ follows a t-distribution, with variance equal to the standard error. So

$$t = \frac{\bar{y}_A - \bar{y}_B}{\sqrt{s^2/n}} \sim t_{n-2}$$

where $n-2$ is the degrees of freedom. Thus we can calculate 95% confidence intervals as $(\hat{D}+t_{n-2}(0.025)\frac{\hat{\sigma}}{\sqrt{n}}, \hat{D}+t_{n-2}(0.975)\frac{\hat{\sigma}}{\sqrt{n}})$. Or, if we want to do it R:

```
Dhat <- mean(yA) - mean(yB)

# Estimate s^2
Resids <- c(yA - mean(yA), yB - mean(yB))
sigma2hat <- (var(yA)+var(yB))/2
StdErr <- sqrt(sigma2hat) * sqrt(1/length(yA)+1/length(yB))
df <- length(yA)+length(yB)-2# degrees of freedom
(CI <- Dhat + qt(c(0.025, 0.975), df)*StdErr)
```

```
## [1] -1.2144013 -0.3016394
```

Of course, R provides functions to do this

```
t.test(yA, yB, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  yA and yB
## t = -3.3247, df = 58, p-value = 0.001538
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.2144013 -0.3016394
## sample estimates:
## mean of x mean of y
## 0.1853541 0.9433744
```

t-tests in R

Now we move on to the analysis part. We will want you to pick a question that is closest to your answer to A1.

- Does fertiliser improve yields compared to the unfertilised control?
- Does farmyard manure improve yields compared to the unfertilised control?
- Does fertiliser improve yields more than farmyard manure?
- Did continuing farmyard manure improve yields compared to stopping in 1871?

All of these questions can be answered with a t-test, For the data set you have chosen:

- **do a t-test**, using `t.test()`. Is there an effect of treatment? How large and what direction?
- Think about the output, which bit tells us if there is an effect? How do we quantify the effect?

You will need to split the yield data so that each treatment is in a different vector. You can do it like this.:

```
Control <- Yields$yield[Yields$Treatment=="Control"]
Fertilised <- Yields$yield[Yields$Treatment=="Fertilised"]
Manure <- Yields$yield[Yields$Treatment=="Manure"]
Stopped <- Yields$yield[Yields$Treatment=="Stopped"]
```

Hint

For the coding, you just need to use the right vector that you have just created.

Once you have done that, work out what the imprtant numbers are and what they mean. The output gives you the means for the two treatments, which can help you work out which one is larger.

Answers

One at a time. The first test can also be used to answer the second question.

As an aside, I have assigned the test to an object, so I can use the values in the answers, using the magic of Markdown. I then put brackets around the line of code so R will show us the output.

```
(CtrlFert.t <- t.test(x=Control, y=Fertilised, var.equal = TRUE))
```

```
##
##  Two Sample t-test
##
## data:  Control and Fertilised
## t = -12.399, df = 34, p-value = 3.621e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.283197 -1.640136
## sample estimates:
## mean of x mean of y
## 0.9122222 2.8738889
```

First, to answer the second question (which bit tells us if there is an effect and how do we quantify it):

- the t-statistic (= -12.4) gives us a standardised estimate of the effect. The t test is a test of whether this number *or one more extreme* is likely if the difference were actually zero: here it is 3.62×10^{-14}, This is extremely small, so we can infer that the difference is almost certainly not zero.
- We can quantify the size of the effect by the difference in the means, i.e. 2.87 - 0.91 = 1.96 t/ha. The confidence interval give the range of likely values we might observe if this were the correct difference, i.e. 2.28 t/ha - 1.64 t/ha.

Note that the order of `x` and `y` in the t-test only changes the sign. Compare the output from `t.test(x=Control, y=Fertilised, var.equal = TRUE)` and `t.test(x=Fertilised, y=Control, var.equal = TRUE)` if you do not believe me (yes, seriously, try things out and see what happens).

Now to the first question for each hypothesis:

- Does fertiliser improve yields compared to the unfertilised control?

The answer is yes, fertilizer does increase yields (shocking, I know):

- The mean of the control treatment, x, is 0.91t/ha, and for the manure it is 2.87 t/ha, so the fertiliser roughly tripled the yield.
- The 95% confidence interval is 2.28 t/ha - 1.64 t/ha, so if the true difference was 1.96t/ha, we are very unlikely to observe a value close to zero: the p-value is 3.62×10^{-14}.

Does farmyard manure improve yields compared to the unfertilised control?

```
(CtrlManure.t <- t.test(x=Control, y=Manure, var.equal = TRUE))
```

```
##
##  Two Sample t-test
##
## data:  Control and Manure
## t = -9.0134, df = 34, p-value = 1.555e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.691386 -2.333059
## sample estimates:
## mean of x mean of y
## 0.9122222 3.9244444
```

The answer is yes, manure does increase yields:

- The mean of the control treatment, x, is 0.91t/ha, and for the manure it is 3.92 t/ha, so the fertiliser roughly quadrupled the yield.
- The 95% confidence interval is 3.69 t/ha - 2.33 t/ha, so if the true difference was 3.01t/ha, we are very unlikely to observe a value close to zero: the p-value is 1.56×10^{-10}.

Does fertiliser improve yields more than farmyard manure?

```
(FertManure.t <- t.test(x=Fertilised, y=Manure, var.equal = TRUE))
```

```
##
##  Two Sample t-test
##
## data:  Fertilised and Manure
## t = -2.9117, df = 34, p-value = 0.006306
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.7838043 -0.3173068
## sample estimates:
## mean of x mean of y
##  2.873889  3.924444
```

The answer is no: manure actually increased yields compared to the fertiliser:

- The mean of the fertiliser treatment, x, is 2.87t/ha, and for the manure it is 3.92 t/ha, so the fertiliser increased yield by roughly a third.
- The 95% confidence interval is 0.32 t/ha - 1.78 t/ha, so if the true difference was 1.05t/ha, we are unlikely to observe a value close to zero: the p-value is 0.00631.

Did continuing farmyard manure improve yields compared to stopping in 1871?

```
(StopManure.t <- t.test(x=Stopped, y=Manure, var.equal = TRUE))
```

```
##
##  Two Sample t-test
##
## data:  Stopped and Manure
## t = -6.0941, df = 34, p-value = 6.503e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.937354 -1.468201
## sample estimates:
## mean of x mean of y
##  1.721667  3.924444
```

The answer is yes, continuing to add manure did lead to a continued increase in yields:

- The mean of the treatment that stopped in 1871, x, is 1.72t/ha, and for the treatment that continued to add manure it is 3.92 t/ha.
- The 95% confidence interval is 2.94 t/ha - 1.47 t/ha, so if the true difference was 2.2t/ha, we are very unlikely to observe a value close to zero: the p-value is 6.5×10^{-7}.

First interpretation

B1. What did you find out from your analysis? (what was the main *biological* conclusion)

B3. Do you think your analysis gave the whole picture of the results of the experiment?

Hint

You may have already answered the first question.

Answers

Depending on which question you asked, you could get the following answers:

- Fertiliser improved yields over unfertilised control
- Farmyard manure improve yields compared to the unfertilised control
- Yields from the Fertiliser were lower than farmyard manure.
- Continuing farmyard manure improved yields compared to stopping in 1871

Each one of these answers does not give the whle story. e.g. just saying that manure gave larger yields than fertiliser does not say that manure is actually better: it could (in principle) still be that both had lower yields than doing nothing. You need all of these effects (and possibly more) to get a full picture.

Part C - t-tests as linear models

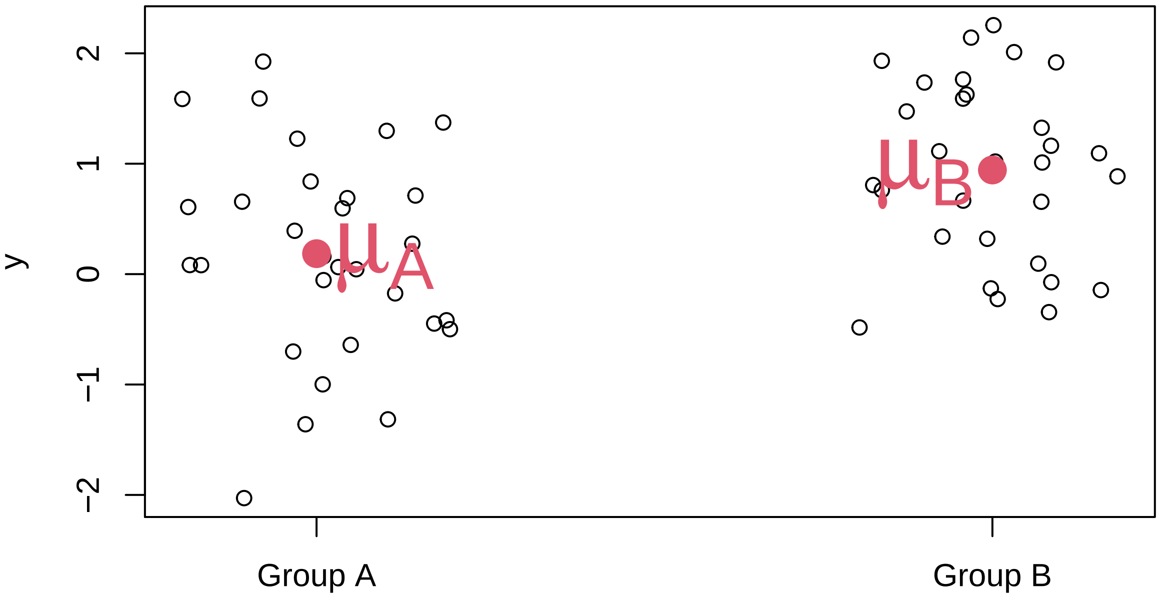
Here we have 4 treatments, and comparing them all separately will be a mess. We might also have more than one type of treatment (e.g. we can decide to look at applying fertiliser and fungicide in the same experiment). It is easier to look at everything in one model: this also also improves the estimates, because the overall error is smaller (this is similar to last week, where a multiple regression improved the estimates of the effects of each variable). Because the models get more complicated, we need a general way of writing them, which is what this section builds towards.

First, we will write a t-test in three ways. All three are the same model, but have different advantages and disadvantages: the first is easy to understnad but limited, the second is more genral, and the third is an extension of what we have already learned about.

t-tests as t-tests

First, t-test as we already know them. y_i^A and y_j^B are vectors with the response in them. They have means μ^A and μ^B and a common variance (σ^2). The t-test asks if $\mu^A = \mu^B$.

```
plot(jitter(xIsB), y, xaxt="n", xlab="")
points(c(0,1), c(mean(yA), mean(yB)), pch=16, cex=2, col=2)
axis(1, c("Group A", "Group B"), at=c(0,1), las=1)
text(0.1,  mean(yA)-0.04, expression(mu[A]), cex=3, col=2)
text(0.9,  mean(yB)-0.04, expression(mu[B]), cex=3, col=2)
```



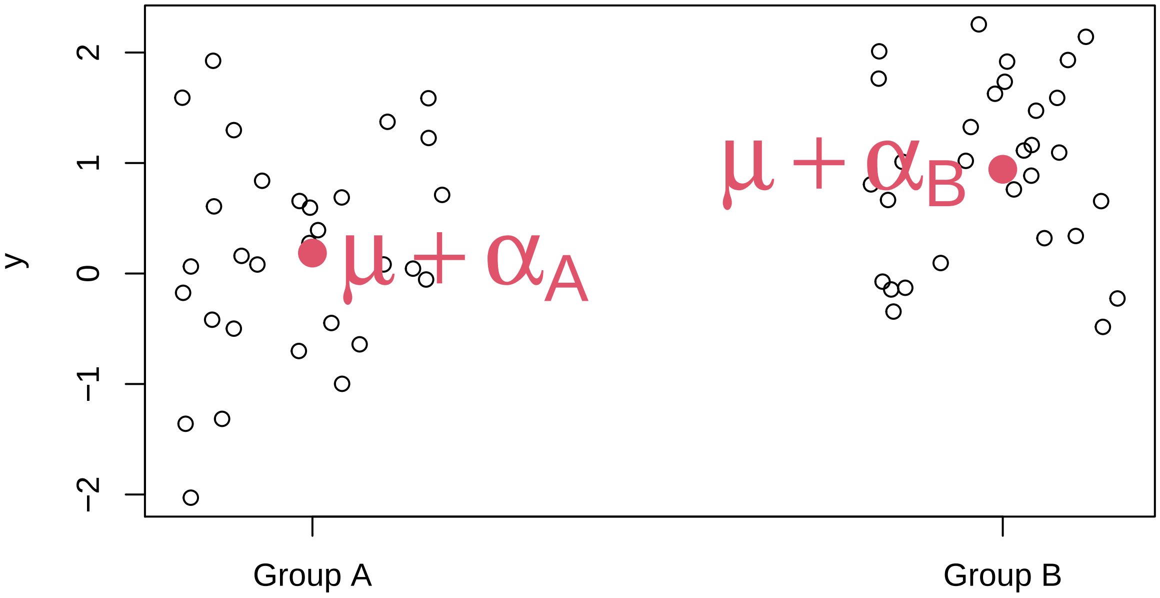
t-tests as an ANOVA

We have one response, y_{ij}, where i says which group y_{ij} is in (i.e. A or B), and j is the j^{th} observation in group i.

$$y_{ij} \sim N(\mu + \alpha_i, \sigma^2)$$

There is a common mean, μ and effects α_i. If the α_i's are different, there is an effect. This is an analysis of variance because the approach to it has been to look at $\sum \alpha_i^2$, which (conveniently) follows a χ^2 distribution.

```
plot(jitter(xIsB), y, xaxt="n", xlab="")
points(c(0,1), c(mean(yA), mean(yB)), pch=16, cex=2, col=2)
axis(1, c("Group A", "Group B"), at=c(0,1), las=1)
text(0.04,  mean(yA)-0.14, expression(mu + alpha[A]), cex=3, col=2, adj=0)
text(0.95, mean(yB)-0.04, expression(mu + alpha[B]), cex=3, col=2, adj=1)
```



This is the same as the t-test version because the mean of group A is $\mu + \alpha_A$, which is μ_A in the t-test version, and the mean of group B is $\mu + \alpha_B$, which is μ_B.

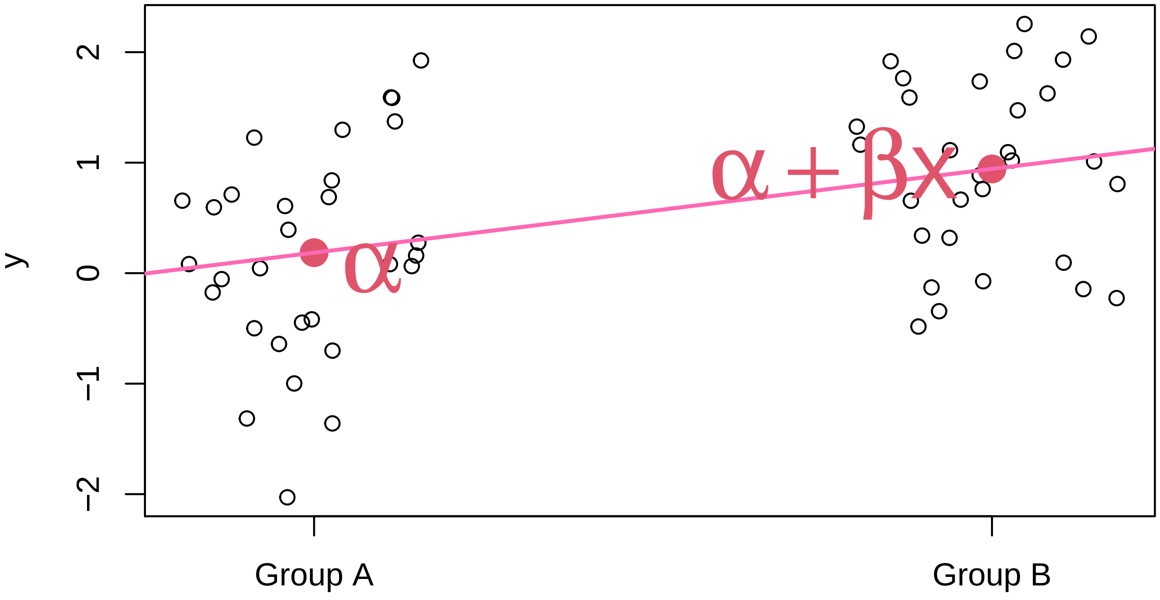
t-tests as a regression model

Again we have one response, y_i, where i denotes the i^{th} observation. It has a covariate X_i, where

$$X_i = \begin{cases} 0 & \text{if } X_i = A \\ 1 & \text{if } X_i = B \end{cases}$$

then $y_i \sim N(\alpha + \beta X_i, \sigma^2)$. We thus simply do a regression aginst X_i. We call X_i a **dummy variable**. It can only take values 0 and 1, and later we will build more complicated analyses out of several dummy variables.

```
plot(jitter(xIsB), y, xaxt="n", xlab="")
points(c(0,1), c(mean(yA), mean(yB)), pch=16, cex=2, col=2)
axis(1, c("Group A", "Group B"), at=c(0,1), las=1)
abline(mean(yA), mean(yB)-mean(yA), col="hotpink", lwd=2)
text(0.04,  mean(yA)-0.14, expression(alpha), cex=3, col=2, adj=0)
text(0.95, mean(yB)-0.04, expression(alpha + beta*x), cex=3, col=2, adj=1)
```



This works because

$$y_i = \begin{cases} \alpha & \text{if } X_i = A \\ \alpha + \beta & \text{if } X_i = B \end{cases}$$

All of the models are the same

For all of the models we have the same variance, and means for the two groups. But we write the means in different ways:

- For the t-test we have 2 vectors, each with a mean
- For the ANOVA we have 1 vector a covariate which says whether the mean is $\mu + \alpha_1$ or $\mu + \alpha_2$
- for the linear model the mean is $\alpha + \beta X_i$, which is α or $\alpha + \beta$, depending on X_i

Identifiability: making sure the model is unique

Part of the discussion abiut multiple regression was about centring the regression. Some of the same issues are about to appear with categorical varaibles. Essentially, we can centre the models in different places. If we write the model badly, it is not clear where we have centred the model: indeed any centre will do.

For example, for the ANOVA model we have $\mu_A = \mu + \alpha_A$ and $\mu_B = \mu + \alpha_B$. What if we add a constant, C, to μ, and subtract the same constant from each α_i? We would get

$\mu_i = \mu + C + \alpha_i - C = \mu + \alpha_i$

In other words, we can add any constant, C, and get the same model. So $\alpha_1 = 100$ is just as correct as $\alpha_1 = -2$, as long as we correct μ. So we need to "fix" something to make the model make sense.

One way to fix this is to say $\sum_i n_i \alpha_i = 0$, so μ is the overall mean of the data. Another way is to say $\alpha_A = 0$, so $\mu_A = \mu$ and $\mu_B = \mu + \alpha_B$. This is the linear model formulation. Although we could just as well fix $\alpha_B = 0$, so $\mu_A = \mu + \alpha_A$. The linear model formulation lets us focus on the difference, $\mu_A - \mu_B = \beta$, and switching to $\alpha_B = 0$ just switched the sign of the difference, as it is now$\mu_B - \mu_A$.

We will come back to this later. For now it is worth nothing that we write the models in different ways, depending on what we want to do. We have to make the choice (even if R has a default), and whilst it can be annoying att imes, it can also make the analyses easier to interprete, because we can write them to make interpretation easier.

Fitting t-tests as linear models

We can do a t-test using `lm()`. First we will do it the hard way: this way you can see how R sees the data. Then we will do it the easy way, where R does the hard stuff.

First we create a variable, `xIsB`, which is 1 if `x=B` and 0 if `x=A`. We do this first by testing `x=="B"`, i.e. does each value of `x` equal the value B. If it does, we get `TRUE`, if not we get `FALSE`. Then we convert this to numbers, using `as.numeric()`: `TRUE` becomes 1 and `FALSE` becomes zero (this is a convention in computing).

Some terminology: a variable like this which is set to take values 0 or 1 to represent something categorical is called a *dummy variable*, or an *indicator variable*.

once we have our dummy variable, `xIsB`, we regress `y` against it:

```
xIsB <- as.numeric(x=="B")
mod0 <- lm(y ~ xIsB)
round(summary(mod0)$coefficients,2)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.19       0.16    1.15     0.25
## xIsB            0.76       0.23    3.32     0.00
```

The intercept is where `xIsB` can be 0, i.e. when `x` is A (i.e. not B). The slope is the change when `xIsB` changes by a value of 1, but as it can only be 0 or 1, it has to be the change from 0 to 1, i.e. the difference between the means for A and B.

Writing the model this way means we have to convert our categorical variable into a number. But R can do this for us (which is a lot more convenient for more complicated situations). It calls categorical variables *factors*. Factors can only take specific values, which we call *levels*. We can give the different possible values of the factor sensible names, e.g. "Control", "Fertilised", and let R do the conversion to numbers internally. But we have to understand something about this process if we want to interpret the output of the analysis:

```
x.Factor <- factor(x)
mod0F <- lm(y ~ x.Factor)
round(summary(mod0F)$coefficients,2)
```

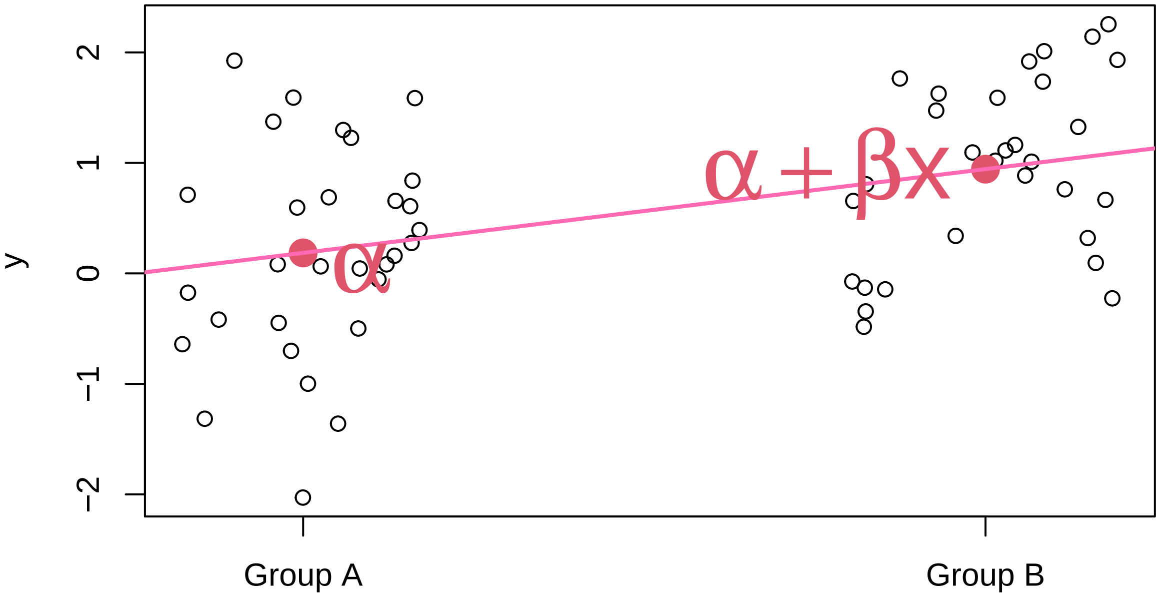
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.19        0.16     1.15      0.25
## x.FactorB        0.76        0.23     3.32      0.00
```

Here `xB` means that there is the variable `x`, and the estimate is of the level B, If `x.Factor=A`, it gets coded internally as 0. If `x.Factor=B`, it gets coded internally as 1. So internally R is creating the dummy variable.

The output needs a bit of explanation. The `(Intercept)` is obviously the intercept, which is when the dummy variable equals 0, i.e. factor level A. The other row in the summary is called `x.FactorB`. This is the name of the variable (`x.Factor`) and then the level that it represents (`B`). This is the difference between the level B and the intercept (level A)[1].

Calculating Predictions

So we know that the `x.FactorB` term is the difference between levels A and B. But what are the predictions for each level? In practice we can get these with `predict()` but if we know how we get them, the output from more complicated models becomes understandable.



Because we have a linear model, the basic model is $y_i = \mu_i + \varepsilon_i$. i.e. each observation has a mean, μ_i, and an error, ε_i. The error follows a normal distirbution: $\varepsilon_i \sim N(0, \sigma^2)$. But we are interested in the mean, μ_i. Because this is a linear model, it is of the form $\mu_i = \alpha + \beta x_i$, where x_i can only be 0 or 1.

We know that α is the intercept, so the coefficient for `(Intercept)`: for our example it is 0.19. This is the mean for the intercept level, i.e. level A.

For level B we have $x_i = 1$, so the mean is $\mu_i = \alpha + \beta \times 1$. β is the `x.FactorB` term: the difference between levels A and B. So the mean is $\mu_i = 0.19 + 0.76 \times 1 = 0.94$.

Exercise: Fit the models with lm()

For the question you looked at before, use `lm()` to fit the model (i.e. to do the t-test).

You will have to create the correct data frame to do this. For example:

[1] You may wonder which level R choses to use as the intercept. It can't understand if something is a control or some other sensible intercept, so it sorts the levels into alphabetical order, and uses the first level as the intercept level. So here A comes before B, and thus is becomes the intercept. If you want another level as the intercept, it can be changed with the `relevel()` function, which we will see soon. In practice, always check which level is the intercept (i.e. which level is missing in the `summary()` or `coef()` output).

```
ManureStop.data <- Yields[Yields$Treatment=="Manure" | Yields$Treatment=="Stopped" ,]
ManureStop.data$Treatment <- factor(ManureStop.data$Treatment)

modMS <-lm(yield ~ Treatment, data=ManureStop.data)
coef(modMS)
summary(modMS)
```

C1. What did you find out from your analysis with the new model?

C2. Do you reach the same conclusion as you did before?

Hint

Fit the model and work out what the coefficients mean, in particular what is the treatment effect?

Answers

One at a time.

Does fertiliser improve yields compared to the unfertilised control?

```
FertControl.data <- Yields[Yields$Treatment=="Fertilised" | Yields$Treatment=="Control" ,]
FertControl.data$Treatment <- factor(FertControl.data$Treatment)

modFC <-lm(yield ~ Treatment, data=FertControl.data)
summary(modFC)
```

```
##
## Call:
## lm(formula = yield ~ Treatment, data = FertControl.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23389 -0.25597 -0.05222  0.28278  0.98611
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.9122     0.1119   8.154 1.64e-09 ***
## TreatmentFertilised   1.9617     0.1582  12.399 3.62e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4746 on 34 degrees of freedom
## Multiple R-squared:  0.8189, Adjusted R-squared:  0.8136
## F-statistic: 153.7 on 1 and 34 DF,  p-value: 3.621e-14
```

From the summary we can see that the Fertilised effect is larger than the intercept: the estimate is 1.96t/ha. The standard error is 0.16t/ha, so the effect is clearly a long way from 0. Indeed the confidence interval is 1.64 t/ha - 2.28 t/ha.

Does farmyard manure improve yields compared to the unfertilised control?

```
ManureControl.data <- Yields[Yields$Treatment=="Manure" | Yields$Treatment=="Control" ,]
ManureControl.data$Treatment <- factor(ManureControl.data$Treatment)

modMC <-lm(yield ~ Treatment, data=ManureControl.data)
summary(modMC)
```

```
##
```

```
## Call:
## lm(formula = yield ~ Treatment, data = ManureControl.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0444 -0.5744 -0.1122  0.2428  2.5356
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.9122     0.2363   3.860 0.000482 ***
## TreatmentManure   3.0122     0.3342   9.013 1.56e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.003 on 34 degrees of freedom
## Multiple R-squared:  0.705,  Adjusted R-squared:  0.6963
## F-statistic: 81.24 on 1 and 34 DF,  p-value: 1.555e-10
```

From the summary we can see that the Manure effect is larger than the intercept: the estimate is 3.01t/ha. The standard error is 0.33t/ha, so the effect is clearly a long way from 0. Indeed the confidence interval is 2.33 t/ha - 3.69 t/ha.

Does fertiliser improve yields more than farmyard manure?

```
FertManure.data <- Yields[Yields$Treatment=="Fertilised" | Yields$Treatment=="Manure" ,]
FertManure.data$Treatment <- factor(FertManure.data$Treatment)

modFM <-lm(yield ~ Treatment, data=FertManure.data)
summary(modFM)
```

```
##
## Call:
## lm(formula = yield ~ Treatment, data = FertManure.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0444 -0.6440 -0.3042  0.7111  2.5356
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.8739     0.2551  11.264 5.08e-13 ***
## TreatmentManure   1.0506     0.3608   2.912  0.00631 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.082 on 34 degrees of freedom
## Multiple R-squared:  0.1996, Adjusted R-squared:  0.176
## F-statistic: 8.478 on 1 and 34 DF,  p-value: 0.006306
```

From the summary we can see that the Manure effect is larger than the intercept: the estimate is 1.05t/ha, so the manure has a larger effect (the intercept is the Fertilised level here). The standard error is 0.36t/ha, so the effect is clearly a long way from 0. Indeed the confidence interval is 0.32 t/ha - 1.78 t/ha.

Did continuing farmyard manure improve yields compared to stopping in 1871?

```
ManureStopped.data <- Yields[Yields$Treatment=="Manure" | Yields$Treatment=="Stopped" ,]
ManureStopped.data$Treatment <- factor(ManureStopped.data$Treatment)
```

```
modMS <-lm(yield ~ Treatment, data=ManureStopped.data)
summary(modMS)
```

```
##
## Call:
## lm(formula = yield ~ Treatment, data = ManureStopped.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0444 -0.6569 -0.2531  0.4833  2.5356
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.9244     0.2556  15.354  < 2e-16 ***
## TreatmentStopped  -2.2028     0.3615  -6.094  6.5e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.084 on 34 degrees of freedom
## Multiple R-squared:  0.5221, Adjusted R-squared:  0.508
## F-statistic: 37.14 on 1 and 34 DF,  p-value: 6.503e-07
```

From the summary we can see that the Stopped effect is smaller than the intercept: the estimate is -2.2t/ha, so stopping the manure treatment reduced the yield.. The standard error is 0.36t/ha, so the effect is clearly a long way from 0. Indeed the confidence interval is -2.94 t/ha to -1.47 t/ha.

C2. Do you reach the same conclusion as you did before?

Yes, hopefully. You should get the same statistics too. If you did not, check that you haven't just got the sign the other way, because the intercept was not the level you though it was.

A Vdeo

Now here's a video of me trying to explain all this, i.e. how to build these models using dummy variables, and how to extends them to more than 2 levels and more than 1 fqctor.

Click here or watch below

Part D - Factors with More than 2 Levels

Now we want to look at all of the treatments together: running a single model is better than several individual models. This is easy to do, but the interpretation of the coefficients needs another step up in complexity.

With more than 2 levels we still the problem of identifiability, i.e. the model can be written in many ways and we have to fix it to one way. For example, with 3 levels we have μ_A, μ_B, and μ_C. We can write these as $mu + \alpha_i$, but the same problem appears: we can add C to mu, and subtract it from each α_i and still get the same mean.

The default method R uses to fix this it by setting one level to be a baseline, and the others are a **contrast** to that level. So we will see that the TreatmentFertilised effect is $\mu_{\text{Fertilised}} - \mu_{\text{Control}}$.

We will use all of the data, and make `Yields$Treatment` a factor:

```
Yields$Treatment <- factor(Yields$Treatment)
levels(Yields$Treatment)
```

```
## [1] "Control"    "Fertilised" "Manure"     "Stopped"
```

Now it has 4 levels. Internally, R is going to use the dummy variable trick again, i.e. converting these to 0s and 1s, so that it can regress the response against a column of 0s and 1s. We can see what it does with a toy example:

```
(A.Factor <- rep(c("A", "B", "C"), each=2))
```

```
## [1] "A" "A" "B" "B" "C" "C"
```

```
model.matrix(~A.Factor)[1:6,]
```

```
##   (Intercept) A.FactorB A.FactorC
## 1           1         0         0
## 2           1         0         0
## 3           1         1         0
## 4           1         1         0
## 5           1         0         1
## 6           1         0         1
```

The variables `A.factorB` and `A.factorC` can be thought of as "Is it B?" and "Is it C?". If it is not either of these, it must be A. So Level A will be the intercept, and the B and C effects will be the differences from level A.

For the yield data we can fit the model (make sure `Treatment` is a factor):

```
mod.Treatments <- lm(yield ~ Treatment, data=Yields)
summary(mod.Treatments)
```

Exercise

You should fit the model with the code above.

D1. How should you interpret the estimates of the Treatment effect?

D2. What is the estimated mean of each Treatment?

Hint

(1) what level is missing in the summary table?
(2) Compare the estimates to those fron the t-tests above
(3) For the means, you need to work out what terms have to be added up. For each treatment there is only one or two.

Answers

```
mod.Treatments <- lm(yield ~ Treatment, data=Yields)
summary(mod.Treatments)
```

```
##
## Call:
## lm(formula = yield ~ Treatment, data = Yields)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0444 -0.4646 -0.1422  0.3724  2.5356
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.9122     0.1973   4.624 1.74e-05 ***
## TreatmentFertilised   1.9617     0.2790   7.031 1.25e-09 ***
## TreatmentManure       3.0122     0.2790  10.796  < 2e-16 ***
## TreatmentStopped      0.8094     0.2790   2.901    0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.837 on 68 degrees of freedom
## Multiple R-squared:  0.6633, Adjusted R-squared:  0.6485
## F-statistic: 44.66 on 3 and 68 DF,  p-value: 4.582e-16
```

D1. How should you interpret the estimates of the Treatment effect?

The missing level is Control, so this is the intercept level. The other effects are contrasts to this level. e.g. the `TreatmentFertilised` effect is the difference between the Fertilised and Control effects: the model estimates that the Fertilised treatment has a yield that is, on average, 1.96t/ha higher than the control. Similarly, the `Stopped` treatment has a yield that is 0.81t/ha higher than the control. Note that the comparison is always with the `Control`, i.e. the intercept level.

D2. What is the estimated mean of each Treatment?

The Control level is the intercept, so this is justnthat. The other levels are the intercept plus their effects:

- Control: 0.91
- Fertilised: 0.91 + 1.96 = 2.87
- Manure: 0.91 + 3.01 = 3.92
- Stopped: 0.91 + 0.81 = 1.72

Part E - Models with Two categorical variables

The treatments in the yields experiment changed over time. Some particularly large changes were made to the experiments around 1970, so we want to know if these had an effect. Later we will ask if the effect changes with the treatments.

Before fitting the model, we need to make sure the `After1970` variable is in the right format:

```
Yields$After1970 <- factor(Yields$After1970) # Make After1970 a factor
Yields$After1970 <- relevel(Yields$After1970, ref="Before") # Make before the intercept
```

The first line just makes it a factor. The second line changes which level will be the intercept. `relevel()` simply changes this so that the level specified by `ref=` becomes the reference level (you can look at the output of the model to see what happens: if you want to play, try fitting different models after `relevel()`ing with `ref="Before"` and `ref="After"`).

The model is like a multiple regression, so we can write it like this:

```
mod.2way <- lm(yield ~ Treatment + After1970, data=Yields)
summary(mod.2way)
```

But what does all this mean? You can work that out for yourselves! If you can do this, all other models are built up the same way.

Exercise: Interpeting the coefficients

Oh, perhaps it might help to walk you through interpreting the output. You might also want to refer back to the Calculating Predictions section to remind yourself how to calculate means: here we are doing the same thing, but with more terms.

First, run the code above and get the summary (you will just need the coefficients, so use those if they are easier to work with).

- What combination of levels is the Intercept (it is one Treatment and one After1970 level)?

- Ignoring the `After1970` factor for a moment, how would you calculate the mean of the other treatment effects? e.g. the Fertilised effect
- Ignoring the `Treatment` factor for a moment, how would you calculate the mean of the before/after 1970 effects?
- For the data with the `Fertilised` treatment and `Before1970`, how would you calculate the mean?
- For the data with the `Fertilised` treatment and `After1970`, how would you calculate the mean?

If it helps (and it may or may not), these are the unique comnination of factor levels in the design matrix. Each row is one observation (but in the data there are several observations with the same 1s and 0s as row 31, for example).

```
knitr::kable(unique(model.matrix(yield ~ Treatment + After1970, data=Yields)))
```

	(Intercept)	TreatmentFertilised	TreatmentManure	TreatmentStopped	After1970After
1	1	0	0	0	0
13	1	0	0	0	1
19	1	1	0	0	0
31	1	1	0	0	1
37	1	0	1	0	0
49	1	0	1	0	1
55	1	0	0	1	0
67	1	0	0	1	1

(`knitr::kable()` is a function to create tables. If I don't use this here, the columns don't align very well)

Hint

What terms are missing from the coefficients? Calculate out some of the means - e.g the Fertilised, before 1970 and the Fertilised After 1970

Answers

First, here is the summary:

```
coef(mod.2way)
```

```
##         (Intercept) TreatmentFertilised     TreatmentManure    TreatmentStopped
##           0.6110417           1.9616667           3.0122222           0.8094444
##      After1970After
##           0.9035417
```

- What combination of levels is the Intercept (it is one Treatment and one After1970 level)?

We can see that for the Treatment the `Control` level is missing, so this must be the intercept. And for the After1970After factor, `Before` is missing, so the intercept must be for data from before 1970 with the Control treatment.

- Ignoring the `After1970` factor for a moment, how would you calculate the mean of the other treatment effects? e.g. the Fertilised effect

These are all something like $\alpha + \beta x_i$:

- Control: This is the intercept, so α, i.e. 0.61.
- Fertilised: This is the intercept plus the Fertilised effect, i.e. $0.61 + 1.96 = 2.57$.
- Manure: This is the intercept plus the Manure effect, i.e. $0.61 + 3.01 = 3.62$.
- Stopped: This is the intercept plus the Fertilised effect, i.e. $0.61 + 0.81 = 1.42$.

- Ignoring the `Treatment` factor for a moment, how would you calculate the mean of the before/after 1970 effects?
- Before: This is the intercept, i.e. 0.61.
- After: This is the intercept plus the After effect, i.e. $0.61 + 0.9 = 1.51$.
- For the data with the `Fertilised` treatment and `Before1970`, how would you calculate the mean?

Now we need both the Fertilised and Before 1970 effects. But we know that Before 1970 is the intercept level, so is set to 0. The (full) model is

$$\mu_i = \beta_0 + \beta_{Fertilised}x_{1i} + \beta_{Manure}x_{2i} + \beta_{Stopped}x_{3i} + \beta_{After1970}x_{4i}$$

where x_{1i} to x_{4i} are the dummy variables. So for Fertilised and Before 1970, $x_{1i} = 1$, $x_{2i} = 0$, $x_{3i} = 0$ and $x_{4i} = 0$. So the mean reduces to

$$\mu_i = \beta_0 + \beta_{Fertilised} = 0.61 + 1.96 = 2.57$$

- For the data with the `Fertilised` treatment and `After1970`, how would you calculate the mean?

Now we need both the Fertilised and After 1970 effects. Following on from the previous question, we have the following dummy variables: $x_{1i} = 1$, $x_{2i} = 0$, $x_{3i} = 0$ and $x_{4i} = 1$. So the mean is now

$$\mu_i = \beta_0 + \beta_{Fertilised} + \beta_{After} = 0.61 + 1.96 + 0.9 = 4.53$$

Phew.

Notice that there is at most one coefficient from the Treatment factor and one from the After1970 factor. This will always be true for these models.

Footnote