Linear Models

Jarrod Hadfield

University of Edinburgh

Linear Models

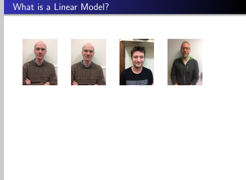
Linear Models

Jarrod Hadfield

University of Edinburgh

OK - yesterday we used the function `lm` to fit a very basic linear model. Today we'll look at linear models more generally. We'll see what makes a linear model linear, and tomorrow we'll see how we can generalise it to non-normal response variables.

What is a Linear Model?

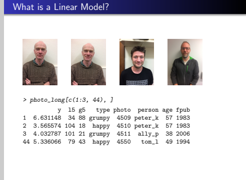


We'll start by looking at our grumpy scores again, but we'll also analyse a new data set as these faces are starting to get a bit tedious.

What is a Linear Model?

```
> photo_long[c(1:3, 44), ]
          y  l5 g5   type photo  person age fpub
1  6.631148  34 88 grumpy  4509 peter_k  57 1983
2  3.565574 104 18  happy  4510 peter_k  57 1983
3  4.032787 101 21 grumpy  4511  ally_p  38 2006
44 5.336066  79 43  happy  4550   tom_l  49 1994
```



What is a Linear Model?

These are the first three lines and the final line of our data frame. We have our average grumpy scores for the 44 photos (y). The next two columns we haven't spoken about yet - these are the number of respondents that gave the photo a grump score less than or equal to 5 (l5) and the number of respondents that gave the photo a grump score greater than 5 (g5). We have the conditions under which the photo was taken (type), the name of the photo (photo), the name of the person photographed (person), their age (age) when photographed (2017) and the year in which they published their first paper (fpub).

What is a Linear Model?

- Model Syntax `y ~ type + fpub`

What is a Linear Model?
Model Syntax y ~ type + fpub

So we might start with a model like this: the average grumpy score is a function of the type of photo and when the person photographed published their first paper. I'm sure you have an intuitive idea of what the model consists of, but what actually does the mathematical model look like?

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[\texttt{y[1]}] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\beta_3\\
E[\texttt{y[2]}] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\beta_3\\
E[\texttt{y[3]}] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\beta_3\\
\vdots\quad & \quad\vdots\\
E[\texttt{y[44]}] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\beta_3
\end{aligned}
$$

What is a Linear Model?
Model Syntax y ~ type + fpub
Set of Simultaneous Equations

It looks overwhelming, but that's mainly because there's just a lot of it. In blue we have data that we've gone out and collected and in red we have the parameters we'd like to estimate using those data. On the left hand side we have the expected value of each observation and on the right hand side we have our predictors in blue: an intercept of all ones, categorical predictors such as type are expanded into a series of binary variables of the form 'is the photo of `type` 'grumpy', yes or no?' and continuous predictors such as `fpub` remain unchanged. The key thing is that although you can do what you want with the predictor variables, the blue things on the right, you are never multiplying or dividing the parameters, the things in red, by each other. That's what makes a linear model linear.

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[y[1]] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\beta_3 + \texttt{I(fpub[1]\^{}2)}\beta_4\\
E[y[2]] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\beta_3 + \texttt{I(fpub[2]\^{}2)}\beta_4\\
E[y[3]] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\beta_3 + \texttt{I(fpub[3]\^{}2)}\beta_4\\
\vdots \quad & \quad \vdots\\
E[y[44]] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\beta_3 + \texttt{I(fpub[44]\^{}2)}\beta_4
\end{aligned}
$$

So for example you could also include the square of the year since first publication (`fpub^2`) and include this as a predictor. This would allow a quadratic relationship between the response variable and `fpub` - a non-linear *relationship* if you like, but the model is still a linear model. You are still taking your data (`fpub`) or some function of your data (`fpub^2` or `type=="grumpy"`) and multiplying them by a parameter and adding them together.

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[\texttt{y[1]}] &= 1\beta_1 + \texttt{(type[1]=="grumpy")}\beta_2 + \texttt{fpub[1]}\beta_3 + \texttt{I(fpub[1]\^{}2)}\beta_4 \\
E[\texttt{y[2]}] &= 1\beta_1 + \texttt{(type[2]=="grumpy")}\beta_2 + \texttt{fpub[2]}\beta_3 + \texttt{I(fpub[2]\^{}2)}\beta_4 \\
E[\texttt{y[3]}] &= 1\beta_1 + \texttt{(type[3]=="grumpy")}\beta_2 + \texttt{fpub[3]}\beta_3 + \texttt{I(fpub[3]\^{}2)}\beta_4 \\
\vdots \quad & \quad \vdots \\
E[\texttt{y[44]}] &= 1\beta_1 + \texttt{(type[44]=="grumpy")}\beta_2 + \texttt{fpub[44]}\beta_3 + \texttt{I(fpub[44]\^{}2)}\beta_4
\end{aligned}
$$

Do what you want with your data

What is a Linear Model?

So you are free to do what ever you wish to your data, you could square it, you could take its absolute value, you could - if it made sense, which it probably wouldn't - take the loop de loop of it.

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[\texttt{y[1]}] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\beta_3 + \texttt{I(fpub[1]\^{}2)}\beta_4 \\
E[\texttt{y[2]}] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\beta_3 + \texttt{I(fpub[2]\^{}2)}\beta_4 \\
E[\texttt{y[3]}] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\beta_3 + \texttt{I(fpub[3]\^{}2)}\beta_4 \\
\vdots \quad & \quad \vdots \\
E[\texttt{y[44]}] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\beta_3 + \texttt{I(fpub[44]\^{}2)}\beta_4
\end{aligned}
$$

Do what you want with your data but a number you have collected should *never* appear on both the left and right hand side *in any form*.

What is a Linear Model?

The only thing you are not allowed to do unless you really know what you are doing is to use the same numbers to calculate something on both the right and on the left.

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[y\texttt{[1]}] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\beta_3 + \texttt{I(fpub[1]\^{}2)}\beta_4\\
E[y\texttt{[2]}] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\beta_3 + \texttt{I(fpub[2]\^{}2)}\beta_4\\
E[y\texttt{[3]}] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\beta_3 + \texttt{I(fpub[3]\^{}2)}\beta_4\\
\vdots \quad & \quad \vdots\\
E[y\texttt{[44]}] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\beta_3 + \texttt{I(fpub[44]\^{}2)}\beta_4
\end{aligned}
$$

Do what you want with your data but a number you have collected should *never* appear on both the left and right hand side *in any form*.

Science AAAS

Bateman in Nature: Predation on Offspring Reduces the Potential for Sexual Selection

To illustrate the point, a few years ago a paper was published in Science on pronghorns (a strange antelope-like animal from North America).

What is a Linear Model?

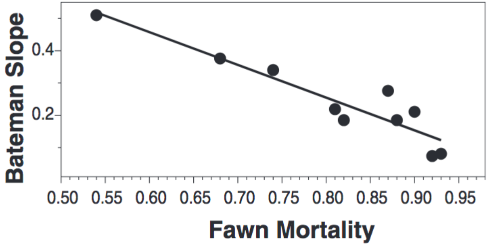
- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

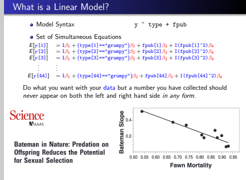
$$
\begin{aligned}
E[y[1]] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\beta_3 + \texttt{I(fpub[1]\^{}2)}\beta_4\\
E[y[2]] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\beta_3 + \texttt{I(fpub[2]\^{}2)}\beta_4\\
E[y[3]] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\beta_3 + \texttt{I(fpub[3]\^{}2)}\beta_4\\
\vdots &\quad \vdots\\
E[y[44]] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\beta_3 + \texttt{I(fpub[44]\^{}2)}\beta_4
\end{aligned}
$$

Do what you want with your data but a number you have collected should *never* appear on both the left and right hand side *in any form*.

Science AAAS

Bateman in Nature: Predation on Offspring Reduces the Potential for Sexual Selection





The key relationship in this paper is depicted here. On the x-axis we have annual fawn mortality over 11 years and on the y-axis we have something called the Bateman slope. It's not important to know what the Bateman slope is, but it is important to know that in this particular instance the Bateman slope is calculated using fawn mortality. So fawn mortality is being used directly as a predictor and indirectly in the response. If you see relationships like this in biology where the relationship is super strong it is nearly always because the same numbers have been used to calculate both the quantities on the y and x axis. The relationship is bogus.

I never want to see anyone do this unless they really know what they're doing. To drive it home, generate 100 random data points (call them y1) and then another random data points (call them y2). Let's imagine these were the size of an organism at two time points and we would like to know whether animals that were large at time 1 grow slower than animals that were small. You might then look at the relationship between growth (y2-y1) and starting size (y1). Try it.

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[\texttt{y[1]}] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\beta_3 + \texttt{I(fpub[1]\^{}2)}\beta_4\\
E[\texttt{y[2]}] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\beta_3 + \texttt{I(fpub[2]\^{}2)}\beta_4\\
E[\texttt{y[3]}] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\beta_3 + \texttt{I(fpub[3]\^{}2)}\beta_4\\
\vdots \quad & \quad \vdots\\
E[\texttt{y[44]}] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\beta_3 + \texttt{I(fpub[44]\^{}2)}\beta_4
\end{aligned}
$$

Ok - let's assume this hasn't been done.

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[\texttt{y[1]}] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\,\beta_3 \\
E[\texttt{y[2]}] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\,\beta_3 \\
E[\texttt{y[3]}] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\,\beta_3 \\
\vdots \quad & \quad \vdots \\
E[\texttt{y[44]}] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\,\beta_3
\end{aligned}
$$

What is a Linear Model?
Model Syntax y ~ type + fpub
Set of Simultaneous Equations

and let's not fit a quadratic term for now. Now, we've only looked at these equations for four data points, and the model only contains three parameters, but that's bad enough.

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[y[1]] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\beta_3\\
E[y[2]] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\beta_3\\
E[y[3]] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\beta_3\\
\vdots \quad & \quad \vdots\\
E[y[44]] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\beta_3
\end{aligned}
$$

- Compact representation: design matrix and parameter vector

$$E[\mathbf{y}] = \mathbf{X}\boldsymbol{\beta}$$

We can, however, represent this whole system of equations very compactly in terms of matrices and vectors. This neat little equation is doing all this: the **X** matrix we call a design matrix it has three columns: the first is all ones, the second is all this blue information here (e.g. `type=="grumpy"`) and so on. $\boldsymbol{\beta}$ is a parameter vector with three elements: β_1, β_2 and β_3. We can matrix multiply these two things together and by doing this we are carrying out this set of operations - multiply $\boldsymbol{\beta}$ by the relevant bit of information and then summing over all terms.

What is a Linear Model?

- Model Syntax `y ~ type + fpub`
- Set of Simultaneous Equations

$$
\begin{aligned}
E[y[1]] &= 1\beta_1 + (\texttt{type[1]=="grumpy"})\beta_2 + \texttt{fpub[1]}\,\beta_3\\
E[y[2]] &= 1\beta_1 + (\texttt{type[2]=="grumpy"})\beta_2 + \texttt{fpub[2]}\,\beta_3\\
E[y[3]] &= 1\beta_1 + (\texttt{type[3]=="grumpy"})\beta_2 + \texttt{fpub[3]}\,\beta_3\\
\vdots \quad & \quad \vdots\\
E[y[44]] &= 1\beta_1 + (\texttt{type[44]=="grumpy"})\beta_2 + \texttt{fpub[44]}\,\beta_3
\end{aligned}
$$

- Compact representation: design matrix and parameter vector

$$E[\mathbf{y}] = \mathbf{X}\beta$$

```
> X <- model.matrix(y ~ type + fpub, data = photo_long)
> X[c(1, 2, 3, 44), ]
   (Intercept) typegrumpy fpub
1            1          1 1983
2            1          0 1983
3            1          1 2006
44           1          0 1994
```


What is a Linear Model?

In R you can generate this design matrix using the function `model.matrix`: and you can see how it corresponds to the data: the 3rd observation is for someone under grumpy conditions who first published a paper in 2006, the 44th observation is for someone under 'not grumpy' (i.e. happy) conditions who first published a paper in 1994. I find it is often helpful to look at the design matrix if I'm not sure exactly what the parameters are relating to.

What is a Linear Model?

$$E[\mathbf{y}] = \mathbf{X}\beta$$

What is a Linear Model?
$E[\mathbf{y}] = \mathbf{X}\beta$

OK- so we've got our set of simultaneous equations and the natural thing to do (if we can remember school) would be to solve them for the β's. So lets take our first three observations where the scores (to the nearest integer) are 7, 4 and 4. The equations are then

$$\begin{aligned} 7 &= 1\beta_1 + 1\beta_2 + 1983\beta_3 \\ 4 &= 1\beta_1 + 0\beta_2 + 1983\beta_3 \\ 4 &= 1\beta_1 + 1\beta_2 + 2006\beta_3 \end{aligned}$$

If we take Equation 1 from Equation 3 we have:

$$-3 = 2006\beta_3 - 1983\beta_3$$

so β_3 must be $-3/(2006 - 1983) = -0.13$. If we substitute β_3 into Equation 2 we have

$$4 = \beta_1 + 1983 \times -0.13$$

so β_1 must be $4 - 1983 \times -0.13 = 262$. If we substitute β_1 and β_3 into equation 1 (or 3):

$$7 = 262 + \beta_2 + 1983 \times -0.13$$

so β_2 must be equal to $7 - 262 - 1983 \times -0.13 = 3$. Tedious perhaps, but simple!

The problem of course, is that to solve them we need to know the expected grumpy score given the predictors. What is the expected score for a photo of a person with these properties? And we don't know the expected score, all that we know is the actual score we have for this particular photo. Which means that we have to do one more bit of modelling - we need to model how the actual scores will deviate from the expected value: what will the noise look like.

What is a Linear Model?

$$E[\mathbf{y}] = \mathbf{X}\beta$$

- The full model

$$\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2_e \mathrm{I})$$

What is a Linear Model?

$E[\mathbf{y}] = \mathbf{X}\beta$

- The full model

$\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2_e \mathrm{I})$

In a standard linear model we assume that the error around the expected value is normally distributed, and that the variance of these errors (the residual variance) is to be estimated.

What is a Linear Model?

$$E[\mathbf{y}] = \mathbf{X}\beta$$

- The full model

$$\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2_e\mathrm{I})$$

- Error structure

$$\sigma^2_e\mathrm{I} = \sigma^2_e \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

What is a Linear Model?
$E[\mathbf{y}] = \mathbf{X}\beta$
The full model
$\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2_e\mathrm{I})$
Error structure
$\sigma^2_e\mathrm{I} = \sigma^2_e \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Now remember that y here is not a single observation: this is a vector (the column) of all the 44 scores. The mean vector (the prediction) is also a vector with 44 elements: one for each data point, and the noise term is a 44 by 44 covariance matrix. **I** is called an identity matrix it has ones along the diagonal, and zero everywhere else.

What is a Linear Model?

$$E[\mathbf{y}] = \mathbf{X}\beta$$

- The full model

$$\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2_e\mathrm{I})$$

- Error structure

$$\sigma^2_e\mathrm{I} = \sigma^{\mathbf{2}}_{\mathbf{e}} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sigma^2_e & 0 & \dots & 0 \\ 0 & \sigma^2_e & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma^2_e \end{bmatrix}$$

What is a Linear Model?

$E[\mathbf{y}] = \mathbf{X}\beta$

- The full model

$\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2_e\mathrm{I})$

- Error structure

$\sigma^2_e\mathrm{I} = \sigma^{\mathbf{2}}_{\mathbf{e}} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sigma^2_e & 0 & \dots & 0 \\ 0 & \sigma^2_e & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma^2_e \end{bmatrix}$

When we multiply it by our residual variance we get the error structure for our model. The key things to note is that first: all off-diagonal terms are zero - this means that we are assuming that the errors around the predicted values are uncorrelated. If photo 1 has a higher score than predicted, you would not expect that photo 2 also had a higher score. This is an assumption of the model and it is easy to see why this might not be true (we'll deal with this in the mixed model lectures). The second thing to note is that we expect the error to be equally variable for each data point. In fact, yesterday we saw that it was probably a poor assumption and we might like to change it.

Linear Model

```
> photo_m5 <- lm(y ~ type + fpub, data = photo_long)
```


Linear Model
> photo_m5 <- lm(y ~ type + fpub, data = photo_long)

so we can fit our model, which you should be familiar with,

Linear Model

```
> photo_m5 <- lm(y ~ type + fpub, data = photo_long)
> summary(photo_m5)
Residuals:
    Min      1Q  Median      3Q     Max 
-3.3639 -0.7954 -0.0344  0.7624  2.8804 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept) 35.99446   30.48944   1.181   0.2446   
typegrumpy   1.22834    0.36994   3.320   0.0019 **
fpub        -0.01597    0.01529  -1.045   0.3023   
---
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.227 on 41 degrees of freedom
Multiple R-squared:  0.2281,	Adjusted R-squared:  0.1904 
F-statistic: 6.058 on 2 and 41 DF,  p-value: 0.004954
```


and the results are.... And again our eyes are dragged to the final column and the stars rewarding us for our effort. But let's think what the model is actually telling us. The intercept is a score of 36 which seems a bit bonkers given we know that the photos were scored on a scale of 1 to 10. It also has a massive standard error: the true value could plausibly be as high as 100 or as low as -30! But what does this number, 36, actually mean? Well the intercept is the score for someone under happy conditions ... but who published their first paper in the year Christ was born (`fpub=0`). Deborah Charlesworth is old, but she's not that old, so for now lets not worry too much about this issue and return to it shortly.

Photos taken of people under grumpy conditions do seem to get a higher grump score - our best estimate is a little over 1 unit higher, and the standard error tells us it is unlikely our true value is less than 0.5. The p-value tells us it is very likely greater than zero.

And finally, the estimate associated with `fpub` tells us that people are being scored a little happier (by 0.016 units) for each year they waited to start publishing. Is this a big or small value? I'm not sure immediately, but `fpub` spans about 40 years (Deborah first published in 1969 (Honky Tonk Women - The Rolling Stones) and Alex Twyford first published in 2011 (Adele - Rolling in the Deep)) and so if we multiply this number by 40 we get -0.639. Not a tremendously big change, and indeed the sampling distribution overlaps zero and the effect is non-significant. However, the standard error is about as big as the estimate so the true change could be as big as -1.862 units. I think Alex would be pretty sad to look 2 units grumpier when he's been publishing as long as Deborah so perhaps we should collect some more data before drawing firm conclusions about the importance of `fpub`.

Linear Model

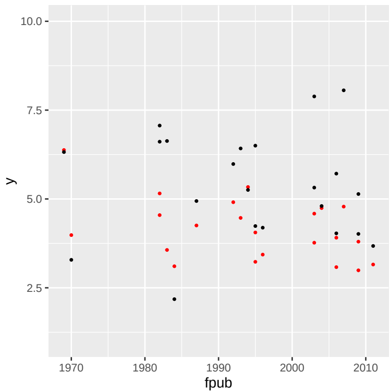
```
> coef(summary(photo_m5))
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```


Linear Model

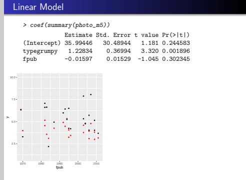
```
> coef(summary(photo_m5))
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```


Of course it's often easier, particularly when there are few terms in the model,

```
> coef(summary(photo_m5))
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```



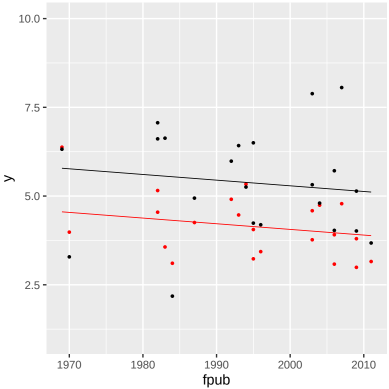
└Linear Model

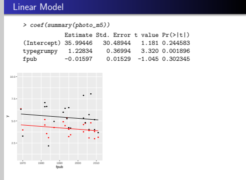


to graph the relationships. So these are our data with fpub along the x-axis and the score on the right axis, and the photo type in different colours (black is grump, red is happy).

Linear Model

```
> coef(summary(photo_m5))
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```

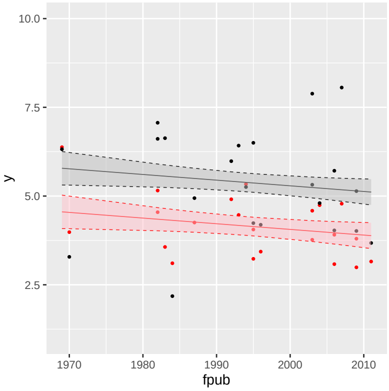


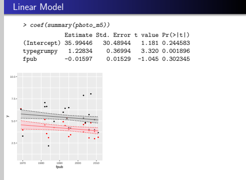


These lines are our best estimates of how the expected score changes as function of fpub and photo type. Note that the lines are parallel; we only have one parameter associated with fpub and therefore we expect the same relationship to hold irrespective of whether the photo type is grumpy or happy.

Linear Model

```
> coef(summary(photo_m5))
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```

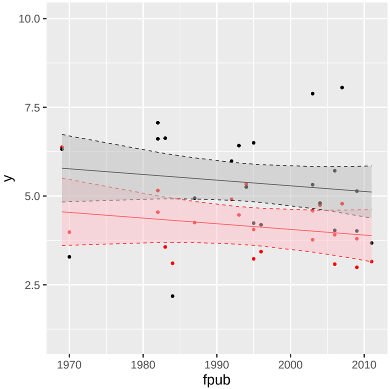


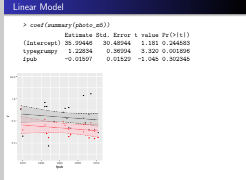


We can also overlay our standard errors around these expectations. The key thing to notice about them is that the standard errors on the predictions are flaring out as we move toward more extreme values of `fpub`, particularly for very low values of `fpub`. It makes sense that this *should* happen; we have quite a lot of people that started publishing in the mid-90's and it makes sense that we can estimate their expected grumpiness more accurately *if* grumpiness does depend on `fpub`. You can also see that by the time we extrapolate down to the year Christ was born the standard errors would be huge. Another way to understand why this happens is to think about what would happen if you took plausible values of the `fpub` slope from its sampling distribution and recalculated the line. You would get a see-saw pattern around 1995 where small differences in slope have magnified effects at extreme values.

Linear Model

```
> coef(summary(photo_m5))
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```

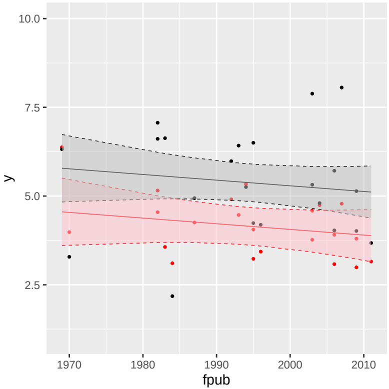




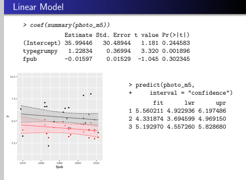
We can also display the 95% confidence intervals, which should be about twice as wide as the standard errors. You might now start worrying that there's conflict between what the coefficient table tells you and what the graph tells you. The confidence intervals overlap; is there really a difference between the scores of grumpy and happy photos? The confidence intervals overlap more at extreme values of `fpub`; does this mean that I am less confident that there would be a difference between grumpy and happy photos for people that started publishing a long time ago - the model output doesn't seem to suggest this? It is important to understand that these are the confidence intervals of the predicted values values not the confidence intervals of the parameters themselves. We'll return to this a little later but for now I think its useful just to know

Linear Model

```
> coef(summary(photo_m5))
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```



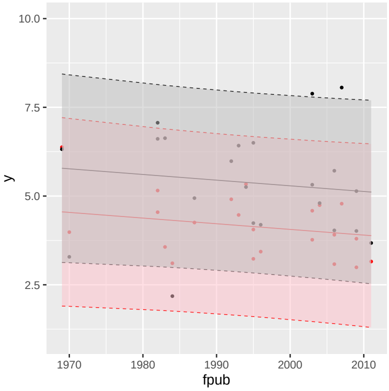

```
> predict(photo_m5,
+       interval = "confidence")
       fit      lwr      upr
1 5.560211 4.922936 6.197486
2 4.331874 3.694599 4.969150
3 5.192970 4.557260 5.828680
```

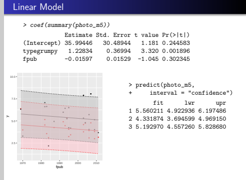
that you can get them using the predict function

Linear Model

```
> coef(summary(photo_m5))
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```

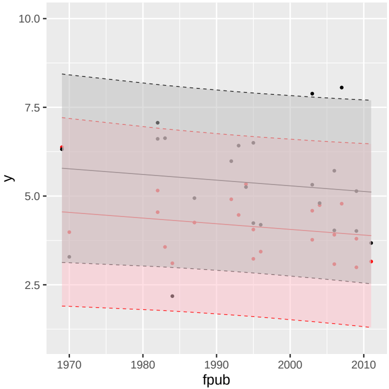



```
> predict(photo_m5,
+     interval = "confidence")
       fit      lwr      upr
1 5.560211 4.922936 6.197486
2 4.331874 3.694599 4.969150
3 5.192970 4.557260 5.828680
```

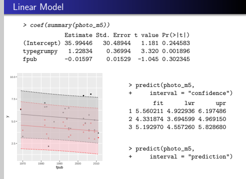



The other type of interval that is useful is the prediction interval. So these intervals should contain 95% of the observations, and you can see that 2 out of 44 observations lie outside the 95% prediction interval, which is about what you expect. However, you can probably also see that 3 out of 22 grumpy photos lie outside their prediction interval whereas no happy photos did so, and in fact there's quite a deficit of red points close to the prediction boundary. As we saw yesterday this is probably because the grumpy scores are more variable than the happy scores but in this model we've estimated a common variance.

```
> coef(summary(photo_m5))
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.99446   30.48944   1.181 0.244583
typegrumpy   1.22834    0.36994   3.320 0.001896
fpub        -0.01597    0.01529  -1.045 0.302345
```




```
> predict(photo_m5,
+       interval = "confidence")
       fit      lwr      upr
1 5.560211 4.922936 6.197486
2 4.331874 3.694599 4.969150
3 5.192970 4.557260 5.828680

> predict(photo_m5,
+       interval = "prediction")
```

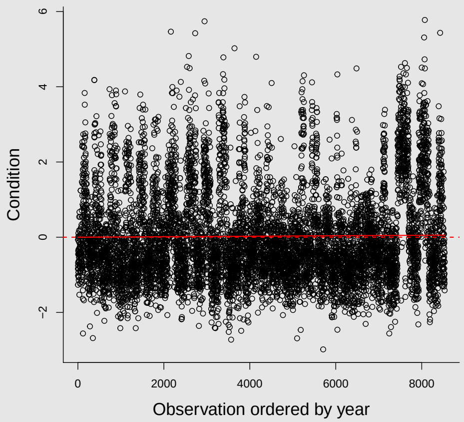



The prediction intervals can also be obtained using the predict function but `prediction` must be passed to the `interval` argument.

Estimates, then standard errors, then p-values.

└ Estimates, then standard errors, then p-values.

Estimates, then standard errors, then p-values.

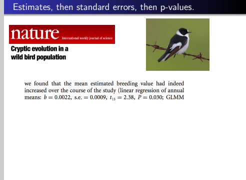


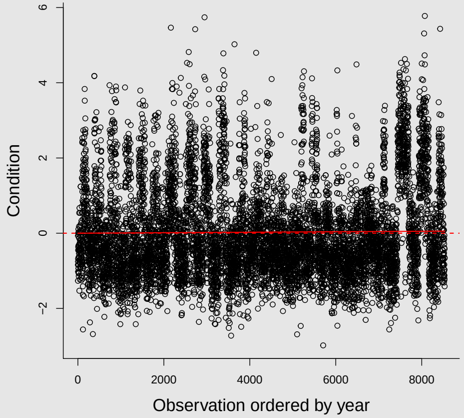
Estimates, then standard errors, then p-values.

nature International weekly journal of science

Cryptic evolution in a wild bird population

we found that the mean estimated breeding value had indeed increased over the course of the study (linear regression of annual means: $b = 0.0022$, s.e. $= 0.0009$, $t_{15} = 2.38$, $P = 0.030$; GLMM





Estimates, then standard errors, then p-values.

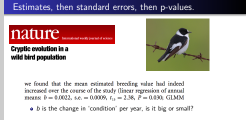
nature International weekly journal of science

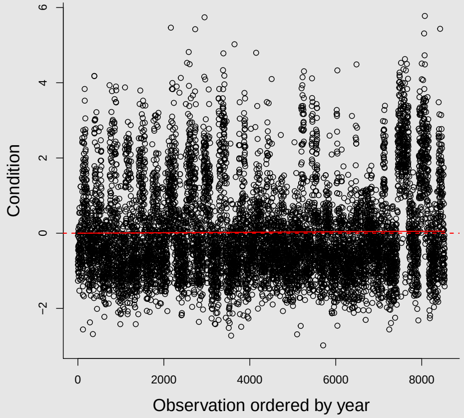
Cryptic evolution in a wild bird population

we found that the mean estimated breeding value had indeed increased over the course of the study (linear regression of annual means: $b = 0.0022$, s.e. $= 0.0009$, $t_{15} = 2.38$, $P = 0.030$; GLMM

- b is the change in 'condition' per year, is it big or small?

Estimates, then standard errors, then p-values.





Estimates, then standard errors, then p-values.

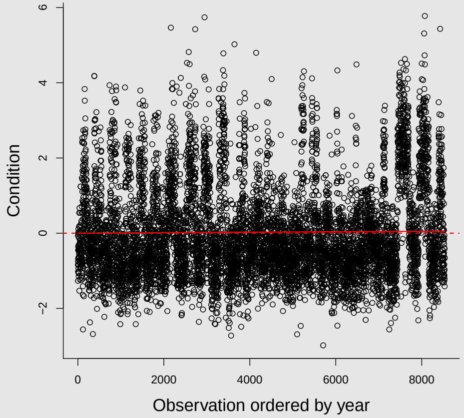
nature International weekly journal of science

Cryptic evolution in a wild bird population

we found that the mean estimated breeding value had indeed increased over the course of the study (linear regression of annual means: $b = 0.0022$, s.e. $= 0.0009$, $t_{15} = 2.38$, $P = 0.030$; GLMM

- b is the change in ‘condition’ per year, is it big or small?
- ‘Condition’ is the residual from a regression of body mass on tarsus length.

Estimates, then standard errors, then p-values.
nature International weekly journal of science
Cryptic evolution in a wild bird population
we found that the mean estimated breeding value had indeed increased over the course of the study (linear regression of annual means: $b = 0.0022$, s.e. $= 0.0009$, $t_{15} = 2.38$, $P = 0.030$; GLMM
- b is the change in ‘condition’ per year, is it big or small?
- ‘Condition’ is the residual from a regression of body mass on tarsus length.



Estimates, then standard errors, then p-values.

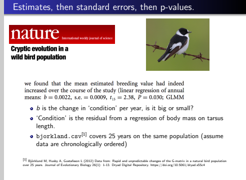
nature International weekly journal of science

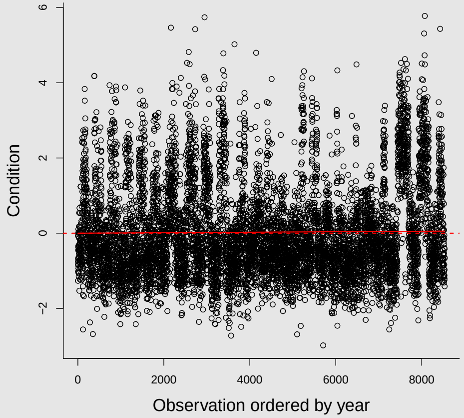
Cryptic evolution in a wild bird population

we found that the mean estimated breeding value had indeed increased over the course of the study (linear regression of annual means: $b = 0.0022$, s.e. $= 0.0009$, $t_{15} = 2.38$, $P = 0.030$; GLMM

- b is the change in 'condition' per year, is it big or small?
- 'Condition' is the residual from a regression of body mass on tarsus length.
- `bjorkland.csv`[1] covers 25 years on the same population (assume data are chronologically ordered)

[1] Björklund M, Husby A, Gustafsson L (2012) Data from: Rapid and unpredictable changes of the G-matrix in a natural bird population over 25 years. Journal of Evolutionary Biology 26(1): 1-13. Dryad Digital Repository. https://doi.org/10.5061/dryad.s55c4





Estimates, then standard errors, then p-values.

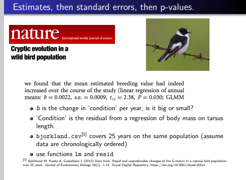
nature International weekly journal of science

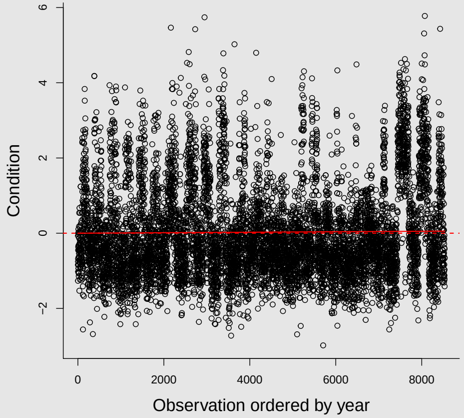
Cryptic evolution in a wild bird population

we found that the mean estimated breeding value had indeed increased over the course of the study (linear regression of annual means: $b = 0.0022$, s.e. $= 0.0009$, $t_{15} = 2.38$, $P = 0.030$; GLMM

- b is the change in 'condition' per year, is it big or small?
- 'Condition' is the residual from a regression of body mass on tarsus length.
- `bjorkland.csv`[1] covers 25 years on the same population (assume data are chronologically ordered)
- use functions `lm` and `resid`

[1] Björklund M, Husby A, Gustafsson L (2012) Data from: Rapid and unpredictable changes of the G-matrix in a natural bird population over 25 years. Journal of Evolutionary Biology 26(1): 1-13. Dryad Digital Repository. https://doi.org/10.5061/dryad.s55c4





Linear Model

One issue that students worry about, I think, is the global intercept. Why have it, why is there no coefficient associated with happy, and why is the coefficient associated with grumpy `type` the difference between happy `type` and grumpy `type`? Surely I just want to know what the underlying mean (or intercept) is for the two types of photo?

Linear Model

```
> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)
```


Linear Model
> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)

Well we are free to remove the global intercept by adding −1 to the model formula. − removes the following term and a 1 stands for the global intercept in R. The global intercept is automatically included so if you don't want it you have to explicitly remove it.

Linear Model

```
> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)

> X <- model.matrix(formula(photo_m6), data = photo_long)
> X[c(1, 2, 3, 44), ]
```


Linear Model

> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)

> X <- model.matrix(formula(photo_m6), data = photo_long)
> X[c(1, 2, 3, 44),]

If we look at the design matrix for this new model

```
> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)

> X <- model.matrix(formula(photo_m6), data = photo_long)
> X[c(1, 2, 3, 44), ]

   typehappy typegrumpy fpub
1          0          1 1983
2          1          0 1983
3          0          1 2006
44         1          0 1994
```


Linear Model

> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)
> X <- model.matrix(formula(photo_m6), data = photo_long)
> X[c(1, 2, 3, 44),]
typehappy typegrumpy fpub
1 0 1 1983
2 1 0 1983
3 0 1 2006
44 1 0 1994

we can see that the first column of 1's has been removed, and has been replaced with a new binary variable 'was the photo taken under happy conditions or not?'.

Linear Model

```
> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)

> X <- model.matrix(formula(photo_m6), data = photo_long)
> X[c(1, 2, 3, 44), ]
```

	typehappy	typegrumpy	fpub
1	0	1	1983
2	1	0	1983
3	0	1	2006
44	1	0	1994

(Intercept)	typegrumpy	fpub
1	1	1983
1	0	1983
1	1	2006
1	0	1994

We can compare our new design matrix with the original design matrix (in blue) where we included a global intercept. You can see that the design matrix for happy photos hasn't changed: if we wanted to work out the expected score for a happy photo in the year of Christ (the intercept) that would simply be our first coefficient in both cases. However, the design matrix for grumpy photos has changed. Before we would have to take the global intercept and add it to the grumpy coefficient to get the expected score for a grumpy photo in the year of Christ. Now, we would just take the grumpy coefficient. So although there are coefficients called `typegrumpy` in both models, the coefficients are actually different things. In our new model it is the intercept (ie. when `fpub=0`) for grumpy photos, and in the original model it was the *difference* in intercept between grumpy and happy photos.

Linear Model

```
> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)

> X <- model.matrix(formula(photo_m6), data = photo_long)
> X[c(1, 2, 3, 44), ]
```

	typehappy	typegrumpy	fpub	(Intercept)	typegrumpy	fpub
1	0	1	1983	1	1	1983
2	1	0	1983	1	0	1983
3	0	1	2006	1	1	2006
44	1	0	1994	1	0	1994

```
> coef(summary(photo_m6))
```

	Estimate	Std. Error	t value	Pr(>|t|)
typehappy	35.99446	30.48944	1.181	0.2446
typegrumpy	37.22280	30.48944	1.221	0.2291
fpub	-0.01597	0.01529	-1.045	0.3023

Linear Model
> photo_m6 <- lm(y ~ type - 1 + fpub, data = photo_long)
> X <- model.matrix(formula(photo_m6), data = photo_long)
> X[c(1, 2, 3, 44),]
> coef(summary(photo_m6))

If we fit our new model then we can see that our model reflects this. The `typegrumpy` coefficient is now similar to the `typehappy` coefficient because it represents the expected score at the birth of Christ. The difference between these two coefficients is about 1.23 and is *exactly* equal to the `typehappy` coefficient in our original (blue) model. This is an important point. The two models are identical it is just they are *reparameterisations* of each other and we're free to use the parameterisation that we feel is most informative. The reason that the default is to have a global intercept is that we're usually interested in the difference between groups or treatment levels. We can calculate it easy enough from these numbers (37.22-35.99) but its not possible from this summary to work out the standard error of the difference, nor is it possible from this summary to test whether the difference is significant. The p-value associated with `typegrumpy` is now testing whether grumpy photos have a score significantly different from zero in the year Christ was born. Not a very relevant hypothesis to be testing.

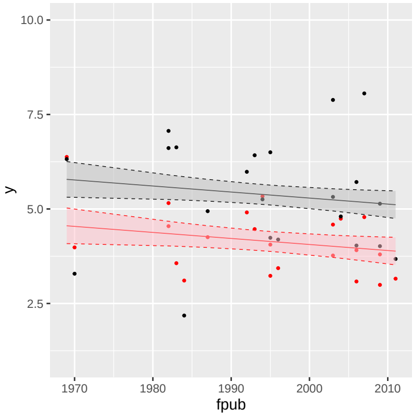
```
> coef(summary(photo_m6))
              Estimate Std. Error t value Pr(>|t|)
typehappy     35.99446   30.48944   1.181   0.2446
typegrumpy    37.22280   30.48944   1.221   0.2291
fpub          -0.01597    0.01529  -1.045   0.3023
```

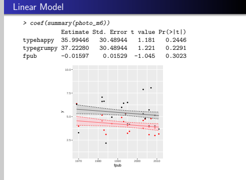

Linear Model
> coef(summary(photo_m6))
Estimate Std. Error t value Pr(>|t|)
typehappy 35.99446 30.48944 1.181 0.2446
typegrumpy 37.22280 30.48944 1.221 0.2291
fpub -0.01597 0.01529 -1.045 0.3023

Just to assure you that the two models really are equivalent and that they are different parameterisations of the same underlying model

Linear Model

```
> coef(summary(photo_m6))
            Estimate Std. Error t value Pr(>|t|)
typehappy   35.99446   30.48944   1.181   0.2446
typegrumpy  37.22280   30.48944   1.221   0.2291
fpub        -0.01597    0.01529  -1.045   0.3023
```





we can also draw our model together with the standard errors around the expected values. Identical to that we saw before.

Interactions

In the previous model we assumed that the effect of `fpub` on grumpiness scores was the same irrespective of whether the photos were taken under grumpy or happy conditions. It seems a perfectly reasonable assumption and I can think of no reason why it would be any different. But people do like to fit interactions because a) they think they should b) because they really want `fpub` to explain something and perhaps it only does under grumpy conditions, or under grumpy conditions if the person is standing on one-leg etc or c) because the interaction is biologically likely or is the focus of the study.

a) and b) are disastrous if the results are not treated with caution. If you test a lot of terms in your model some will be significant just by chance even if the true value of the coefficients is zero. You'll get many false positives and you'll waste a lot of time coming up with some cock and bull story to explain the finding. This is as likely to happen with main effects as it is with interactions, but the problem is that there is generally more possible interactions than main effects. With 6 main effects there are 15 two-way interactions. So I urge you to think before jumping into the murky world of interactions. Decide before you fit the model which interactions, if any, are plausible and/or of primary interest. Don't bung all the two-way and three-way interactions into a model and hope to get something sensible out the other end.

Interactions

```
> photo_m7 <- lm(y ~ type + fpub + type:fpub, data = photo_long)
```


Interactions
> photo_m7 <- lm(y ~ type + fpub + type:fpub, data = photo_long)

Lets presume we have thought carefully (I haven't) and we'd like to fit the interaction. In R we can do this by having a colon between the two terms we'd like to interact. In this case we've also fitted main effects (we have `fpub` and `type` alone in the model formula too) and this is usually what you would like to do.

Interactions

```
> photo_m7 <- lm(y ~ type * fpub, data = photo_long)
```


Interactions
> photo_m7 <- lm(y ~ type * fpub, data = photo_long)

We can also define the model more compactly by just having our two terms and a star between them. This star is shorthand for fit the two main effects and the interaction between them. We can then look at the model summary

Interactions

```
> photo_m7 <- lm(y ~ type * fpub, data = photo_long)

                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      64.30779   43.19188  1.4889   0.1444
typegrumpy      -55.39831   61.08254 -0.9069   0.3699
fpub             -0.03016    0.02165 -1.3929   0.1713
typegrumpy:fpub   0.02839    0.03062  0.9271   0.3595
```


Interactions
> photo_m7 <- lm(y ~ type * fpub, data = photo_long)
Estimate Std. Error t value Pr(>|t|)
(Intercept) 64.30779 43.19188 1.4889 0.1444
typegrumpy -55.39831 61.08254 -0.9069 0.3699
fpub -0.03016 0.02165 -1.3929 0.1713
typegrumpy:fpub 0.02839 0.03062 0.9271 0.3595

and I can still see everybody's eyes going to the p-value column. You're gutted! The interaction between type and fpub is not significant, but even worse, grumpy photos are no longer significantly different from happy photos. The one significant effect you had has disappeared, so what do you do now? Drop the interaction - it's not significant after all - and pretend you never did it? But is this honest - doesn't the difference between grumpy photos and happy photos depend on the interaction not being there?

Well lets think what the model looks like, and what hypotheses we're actually testing first.

Interactions

```
> photo_m7 <- lm(y ~ type * fpub, data = photo_long)

                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      64.30779   43.19188  1.4889   0.1444
typegrumpy      -55.39831   61.08254 -0.9069   0.3699
fpub             -0.03016    0.02165 -1.3929   0.1713
typegrumpy:fpub   0.02839    0.03062  0.9271   0.3595

> X <- model.matrix(formula(photo_m7), data = photo_long)
> X[c(1, 2, 3, 44), ]
   (Intercept) typegrumpy fpub typegrumpy:fpub
1            1          1 1983            1983
2            1          0 1983               0
3            1          1 2006            2006
44           1          0 1994               0
```

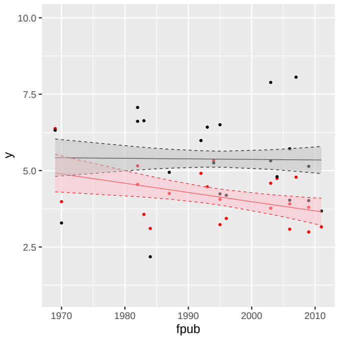

Interactions
> photo_m7 <- lm(y ~ type * fpub, data = photo_long)
Estimate Std. Error t value Pr(>|t|)
(Intercept) 64.30779 43.19188 1.4889 0.1444
typegrumpy -55.39831 61.08254 -0.9069 0.3699
fpub -0.03016 0.02165 -1.3929 0.1713
typegrumpy:fpub 0.02839 0.03062 0.9271 0.3595
> X <- model.matrix(formula(photo_m7), data = photo_long)
> X[c(1, 2, 3, 44),]
(Intercept) typegrumpy fpub typegrumpy:fpub
1 1 1 1983 1983
2 1 0 1983 0
3 1 1 2006 2006
44 1 0 1994 0

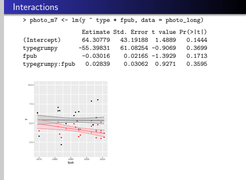
Again, I can find it helpful to inspect the design matrix if I'm not sure exactly what these coefficients refer to. Lets try and sketch it by hand. If we set everything to zero expect the intercept we have the expected score of a happy photo at the birth of Christ (64.31). If we add the `typegrumpy` coefficient to this we get the expected score of a happy photo at the birth of Christ (64.31 + -55.4 = 8.91). Next we need to work out how the scores change with `fpub`, so lets start by looking what the design matrix looks like for a happy photo (so the second row). The only column which involves `fpub` is the third column, so the coefficient `fpub` is referring to the slope for happy photos: The score is expected to change by -0.03 units for every year. If we wanted to do the same for grumpy photos (for example those in the 1st and 3rd row), well `fpub` appears twice and so what we would have to do is sum the two coefficients `fpub` and `typegrumpy:fpub` in order to get the slope for grumpy photos (-0.03 + 0.03 = 0). The `typegrumpy:fpub` coefficient is therefore the *difference* between the two slopes.

So the p-value for the `typegrumpy` coefficient is whether the two types of photos are expected to differ for people that started publishing 2000 years ago. When we did not have an interaction we had fairly precise information on whether there would be a difference for those people publishing 2000 years ago because we assumed the difference that we observed amongst our peers would also hold back then. If we allow the slopes to differ then the difference between the scores of happy and grumpy photos is allowed to differ amongst people that started publishing at different times. We can only really know how this difference might change across the range of `fpub` we have sampled, and outside of this range we have to extrapolate. Accordingly, the further outside the range we look the more unreliable our extrapolation is expected to become, to the point where we may no longer be able to confidently say what the difference between the two photographs would be for someone publishing 2000 years ago.

Interactions

```
> photo_m7 <- lm(y ~ type * fpub, data = photo_long)

                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      64.30779   43.19188  1.4889   0.1444
typegrumpy      -55.39831   61.08254 -0.9069   0.3699
fpub             -0.03016    0.02165 -1.3929   0.1713
typegrumpy:fpub   0.02839    0.03062  0.9271   0.3595
```





We can see this graphically. The standard errors overlap before about 1980, but are quite far apart after this. In special cases, non-overlapping standard errors would indicate that the difference between the two effects is significant at the 5% level, but this is not always the case, and indeed this is not one of those special cases. A more reliable way to test for significance is to redefine the null hypothesis so it makes sense. So one possibility would be to start testing hypotheses about differences in the range of fpub we have sampled. We could, for example, take 1969 from everyone's fpub so that the new intercept is now 1969, when Deborah started publishing. That might make more sense.

Mean-centring

```
> photo_long$mcfpub <- photo_long$fpub - mean(photo_long$fpub)
```


Mean-centring
> photo_long$mcfpub <- photo_long$fpub - mean(photo_long$fpub)

A more common approach is to mean centre the variable. So recalculate everybody's `fpub` as a deviation from the mean `fpub` in the sample (around 1995).

Mean-centring

```
> photo_long$mcfpub <- photo_long$fpub - mean(photo_long$fpub)

> photo_m8 <- lm(y ~ type * mcfpub, data = photo_long)

                   Estimate Std. Error t value  Pr(>|t|)
(Intercept)         4.14753    0.26204 15.8279 8.059e-19
typegrumpy          1.22834    0.37058  3.3147 1.957e-03
mcfpub             -0.03016    0.02165 -1.3929 1.713e-01
typegrumpy:mcfpub   0.02839    0.03062  0.9271 3.595e-01
```


Mean-centring
> photo_long$mcfpub <- photo_long$fpub - mean(photo_long$fpub)
> photo_m8 <- lm(y ~ type * mcfpub, data = photo_long)
Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.14753 0.26204 15.8279 8.059e-19
typegrumpy 1.22834 0.37058 3.3147 1.957e-03
mcfpub -0.03016 0.02165 -1.3929 1.713e-01
typegrumpy:mcfpub 0.02839 0.03062 0.9271 3.595e-01

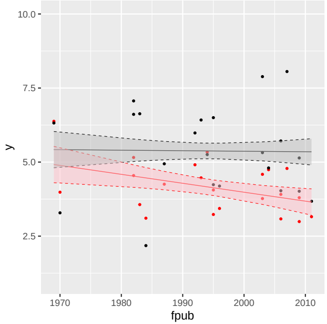
If we refit the model but with `fpub` mean-centred we can see that the coefficients have changed, as have the standard errors and the p-values. The intercept now looks reasonable; this is the expected grumpy score for a happy photo of someone who first started publishing around 1995. The standard errors are quite tight because we are not having to extrapolate way beyond our data, and the difference between happy and grumpy scores for these people is about 1 unit and this difference is well estimated and significantly different from zero.

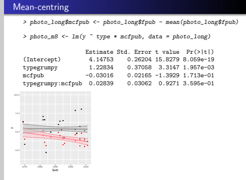
It is important to remember that this model is identical to the model that we fitted where `fpub` wasn't mean centred. All's we've done is reparameterised the model by shifting the value of `fpub` so the intercept is *interpreted* differently. The slope parameters associated with `fpub` haven't changed. It's important that you report the mean values of covariates if you mean centre otherwise people can't compare your conclusions with theirs if the mean `fpub` differed, which it most likely will.

Mean-centring

```
> photo_long$mcfpub <- photo_long$fpub - mean(photo_long$fpub)

> photo_m8 <- lm(y ~ type * mcfpub, data = photo_long)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.14753	0.26204	15.8279	8.059e-19
typegrumpy	1.22834	0.37058	3.3147	1.957e-03
mcfpub	-0.03016	0.02165	-1.3929	1.713e-01
typegrumpy:mcfpub	0.02839	0.03062	0.9271	3.595e-01



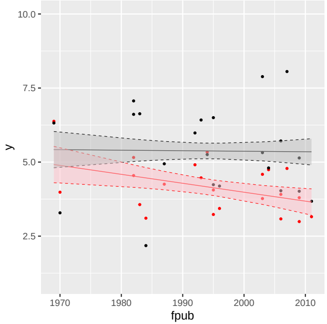


We can see this if we plot our predictions and their standard errors. It's identical to the previous plot. We can also see this if we compare this model with the previous one

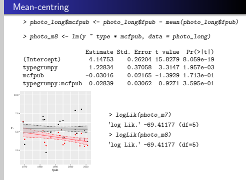
Mean-centring

```
> photo_long$mcfpub <- photo_long$fpub - mean(photo_long$fpub)

> photo_m8 <- lm(y ~ type * mcfpub, data = photo_long)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.14753	0.26204	15.8279	8.059e-19
typegrumpy	1.22834	0.37058	3.3147	1.957e-03
mcfpub	-0.03016	0.02165	-1.3929	1.713e-01
typegrumpy:mcfpub	0.02839	0.03062	0.9271	3.595e-01




```
> logLik(photo_m7)
'log Lik.' -69.41177 (df=5)
> logLik(photo_m8)
'log Lik.' -69.41177 (df=5)
```

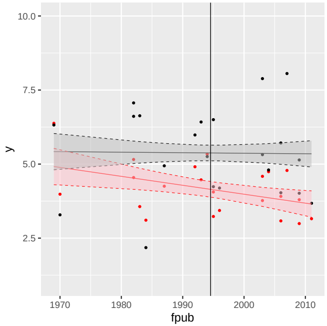



The likelihoods are the same.

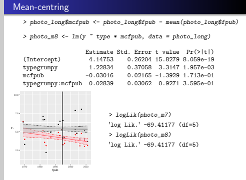
Mean-centring

```
> photo_long$mcfpub <- photo_long$fpub - mean(photo_long$fpub)

> photo_m8 <- lm(y ~ type * mcfpub, data = photo_long)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.14753	0.26204	15.8279	8.059e-19
typegrumpy	1.22834	0.37058	3.3147	1.957e-03
mcfpub	-0.03016	0.02165	-1.3929	1.713e-01
typegrumpy:mcfpub	0.02839	0.03062	0.9271	3.595e-01




```
> logLik(photo_m7)
'log Lik.' -69.41177 (df=5)
> logLik(photo_m8)
'log Lik.' -69.41177 (df=5)
```

All that we have done is reparameterised our model so the value of `fpub` that corresponds to the intercept is here (the vertical line). You will sometimes come across the word *contrasts*: different contrasts are essentially different parameterisations of the same model and they are often used so that the estimates are easier to interpret and any hypothesis tests have a more natural meaning.
You might however worry - rightly so - that being able to shift our null hypothesis (in this case the difference between happy and grumpy photos at the birth of Christ is zero, to the difference is zero in 1995) is open to abuse. It is. I would decide before hand not to fit the interaction if I thought it was implausible or if I did think it was plausible I would decide *prior* to the analysis to either a) mean centre the covariate (or perform a type-II test, more of which later) or b) drop the interaction if non-significant (mean centring doesn't effect the estimate of the slope or the associated p-value) or c) test the joint null hypothesis that both the main term and the interaction (`type` and `fpub:type`) are zero. This latter test asks whether there is evidence that grumpy and happy photos are scored differently at *any* value of `fpub` rather than some *specific* value. We'll see how to do this a little later.

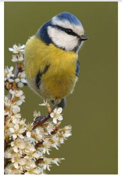
Is there really an interaction?

Is there really an interaction?

As an aside, you often see people claim significant differences between groups, or significant differences between groups in the effect of a covariate (a group by covariate interaction) based on flawed logic. Sometimes you will see people do it within papers (or particularly in talks); for example they might find a significant effect of some experimental treatment in males, but a non-significant effect in females, and then claim this is evidence that the two sexes respond to the treatment differently. More commonly you see it done when someone compares the results of their study to a previous one. For example, one study might apply an experimental treatment and find a significant response, while another study applies the same treatment and does not find a significant response. By claiming that the treatment has different effects in different populations the authors are essentially claiming a population by treatment interaction, and you can often find large chunks of discussion dedicated to explaining why it exists. In many cases, the case for a difference existing at all is weak.

Is there really an interaction?

'*We found higher heritabilities overall than Hadfield et al.(2006a), thereby illustrating that the genetic determinism of colouration can vary across populations and requires further quantitative genetic investigations.*'



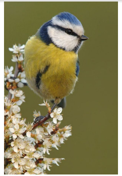
Is there really an interaction?

'*We found higher heritabilities overall than Hadfield et al.(2006a), thereby illustrating that the genetic determinism of colouration can vary across populations and requires further quantitative genetic investigations.*'

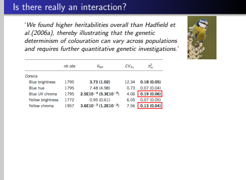
This is a quote from a paper written by people with good quantitative skills. In their study they estimated the heritability of plumage colouration in a Corsican population of blue tits and found that the heritability was significantly different from zero. I had estimated the same parameters previously and could not reject heritability values of zero, leading to the authors claim that heritabilities vary from population to population.

Is there really an interaction?

'We found higher heritabilities overall than Hadfield et al.(2006a), thereby illustrating that the genetic determinism of colouration can vary across populations and requires further quantitative genetic investigations.'



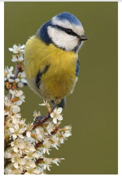
	nb obs	V_{Am}	CV_{A_m}	h^2_m
Corsica				
Blue brightness	1795	**3.73 (1.02)**	12.34	**0.18 (0.05)**
Blue hue	1795	7.48 (4.98)	0.73	0.07 (0.04)
Blue UV chroma	1795	**2.5E10^{-4} (5.3E10^{-5})**	4.06	**0.19 (0.06)**
Yellow brightness	1772	0.95 (0.61)	6.05	0.07 (0.05)
Yellow chroma	1957	**3.6E10^{-3} (1.2E10^{-3})**	7.56	**0.13 (0.04)**



This is their estimates for two plumage regions, the blue head and the yellow chest, showing that 19% of the variation in the blue colouration and 13% of the yellow colouration is genetic. You can see that the standard errors are less than half the estimate and so if the sampling distributions of these parameters were normal (generally you need very large sample sizes before the sampling distribution of a heritability starts to look normal) you would be able to claim that they are significantly different from zero.

Is there really an interaction?

'We found higher heritabilities overall than Hadfield et al.(2006a), thereby illustrating that the genetic determinism of colouration can vary across populations and requires further quantitative genetic investigations.'

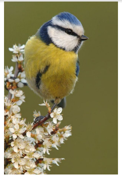


	nb obs	V_{Am}	CV_{A_m}	h^2_m
Corsica				
Blue brightness	1795	**3.73 (1.02)**	12.34	**0.18 (0.05)**
Blue hue	1795	7.48 (4.98)	0.73	0.07 (0.04)
Blue UV chroma	1795	**2.5E10^{-4} (5.3E10^{-5})**	4.06	**0.19 (0.06)**
Yellow brightness	1772	0.95 (0.61)	6.05	0.07 (0.05)
Yellow chroma	1957	**3.6E10^{-3} (1.2E10^{-3})**	7.56	**0.13 (0.04)**

	cap colour	chest colour
heritability	0.10 ± 0.11	0.07 ± 0.09

These are my previous estimates, and you can see that their *estimates* of heritability are higher than mine (almost twice is large), but does this imply they're significantly higher? How would you test whether they're significantly different? A very useful result is that the variance of $a - b$ is equal to the variance of a plus the variance of b minus twice the covariance. Because the two studies are independent the sampling errors on the pair of estimates (mine and theirs) are independent and so the covariance is zero. So our best estimate of the difference (for the blue colour) is $0.19 - 0.10 = 0.09$ and the sampling variance around the difference is $0.06^2 + 0.11^2 = 0.016$ giving a standard error of $\sqrt{0.016} = 0.13$. So what's the chance we see a difference of 0.09, or bigger, just by chance? How do we perform a one-tailed test on the hypothesis that heritabilities in their population are larger than in mine? `1-pnorm(0.09, 0, 0.13)`=0.244. Not very convincing!

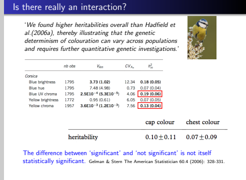
‘*We found higher heritabilities overall than Hadfield et al.(2006a), thereby illustrating that the genetic determinism of colouration can vary across populations and requires further quantitative genetic investigations.*’



	nb obs	V_{Am}	CV_{A_m}	h^2_m
Corsica				
Blue brightness	1795	**3.73 (1.02)**	12.34	**0.18 (0.05)**
Blue hue	1795	7.48 (4.98)	0.73	0.07 (0.04)
Blue UV chroma	1795	**2.5E10^{-4} (5.3E10^{-5})**	4.06	**0.19 (0.06)**
Yellow brightness	1772	0.95 (0.61)	6.05	0.07 (0.05)
Yellow chroma	1957	**3.6E10^{-3} (1.2E10^{-3})**	7.56	**0.13 (0.04)**

	cap colour	chest colour
heritability	0.10 ± 0.11	0.07 ± 0.09

The difference between ‘significant’ and ‘not significant’ is not itself statistically significant. Gelman & Stern The American Statistician 60.4 (2006): 328-331.



This is such a common mistake to make that prominent statisticians have repeatedly tried to caution scientists about it. It is obvious once its explained, but unfortunately this type of misinterpretation doesn’t have a catchy memorable name so you can’t say ’Oh you’ve made an x mistake’ but you could direct a person to this paper by Gelman & Stern (2006), where it’s nicely explained.

Confounding

So hopefully you're starting to get a feel for the underlying model you're constructing for your data and an understanding of the hypotheses that are being tested when you summarise a model in R. The model you choose to fit, and the hypotheses you choose to test, should be dictated by how you think your data came to be and the questions you want to ask of it. However, sometimes the data you have collected aren't up to the task of answering the questions you would like to ask of them. The information the data provide about a parameter might be so small that that parameter can't be estimated precisely enough to be useful. This might be because you haven't collected enough data per se, or you haven't collected enough data in the right way. Sometimes this is unavoidable.

Let's imagine that I was really interested in whether the length of time you had spent in academia made you grumpy, but I also felt like age may also play some role.

Confounding

```
> photo_long$ypub <- 2017 - photo_long$fpub
```


Confounding
> photo_long$ypub <- 2017 - photo_long$fpub

The first thing I'm going to do is transform `fpub` (the year in which the person published their first paper) because I find it confusing. Instead I've calculated the number of years from when the photo was taken (2017) since the person started publishing - time in academia if you like. Again, the model would be identical if we fitted `fpub` or `ypub` I've just reparameterised it so I can make better sense of it. Now if I want to test whether age and/or time in academia makes you grumpy the natural thing to do

Confounding

```
> photo_long$ypub <- 2017 - photo_long$fpub
> photo_m9 <- lm(y ~ type + ypub + age, data = photo_long)
```


Confounding
> photo_long$ypub <- 2017 - photo_long$fpub
> photo_m9 <- lm(y ~ type + ypub + age, data = photo_long)

would be to add age to the model.

Confounding

```
> photo_long$ypub <- 2017 - photo_long$fpub
> photo_m9 <- lm(y ~ type + ypub + age, data = photo_long)

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.48779     3.2926  0.4519 0.654420
typegrumpy   1.28533     0.4362  2.9467 0.005948
ypub        -0.08073     0.1288 -0.6266 0.535349
age          0.09424     0.1280  0.7363 0.466935
```


Confounding

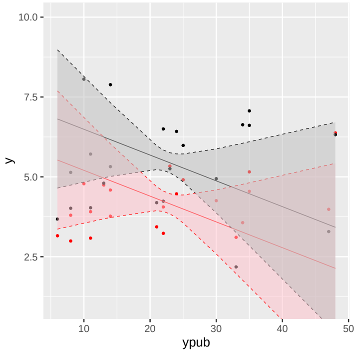
```
> photo_long$ypub <- 2017 - photo_long$fpub
> photo_m9 <- lm(y ~ type + ypub + age, data = photo_long)

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.48779     3.2926  0.4519 0.654420
typegrumpy   1.28533     0.4362  2.9467 0.005948
ypub        -0.08073     0.1288 -0.6266 0.535349
age          0.09424     0.1280  0.7363 0.466935
```

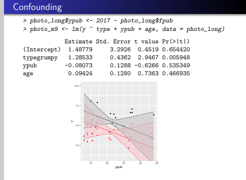

We've gone to the p-values again! Nothing doing. But wait, the coefficient for ypub is negative and is quite larger in magnitude. If we multiply our coefficient by 40 we get -3.23: If Alex had been publishing as long as Deborah we would expect him to look around 3 units happier. That's our best estimate and its a big effect. But the standard errors are so large that the confidence intervals suggest that he could be up to 14 units happier or 7 units grumpier. Previously, we suggested that our estimate of the effect of time in academia was perhaps a little bit imprecise and we might want to collect some more data before making firm conclusions. By adding another covariate we're now saying that our estimate is so noisy we might as well ignore it and state we have no real idea what the effect of time since publishing is.

Confounding

```
> photo_long$ypub <- 2017 - photo_long$fpub
> photo_m9 <- lm(y ~ type + ypub + age, data = photo_long)
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.48779     3.2926  0.4519 0.654420
typegrumpy     1.28533     0.4362  2.9467 0.005948
ypub          -0.08073     0.1288 -0.6266 0.535349
age            0.09424     0.1280  0.7363 0.466935
```

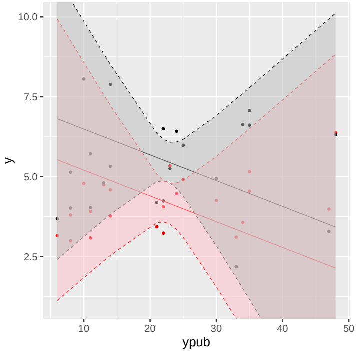


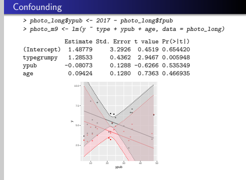


We can see this graphically too if we plot our predictions and their standard errors.

Confounding

```
> photo_long$ypub <- 2017 - photo_long$fpub
> photo_m9 <- lm(y ~ type + ypub + age, data = photo_long)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.48779     3.2926  0.4519 0.654420
typegrumpy   1.28533     0.4362  2.9467 0.005948
ypub        -0.08073     0.1288 -0.6266 0.535349
age          0.09424     0.1280  0.7363 0.466935
```

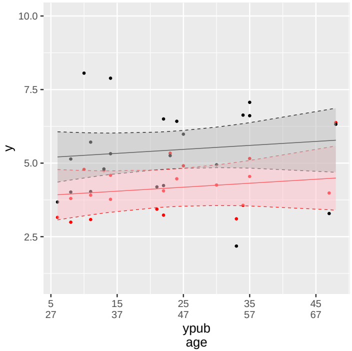


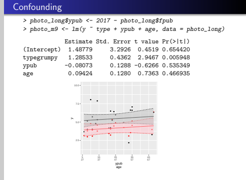


If we plot the confidence intervals it looks even worse. When making this plot I've calculated the expected grumpy score for people that started publishing at a range of dates, but I've held their age constant. I've assumed their age is 48; the mean age of those photographed. This is what the coefficient in a linear model is telling us. If we held age constant and photo type constant what would be the effect of ypub on the grumpy score: it is *trying* to estimate the causal effect of ypub (and I emphasise the word *trying*, it's not an experiment). It makes sense then that it is hard to estimate this effect, because if we held age constant there would not be much variation in first publication date. For example, there are four people aged 38 but they have been publishing for a restricted range of years (10-14) and so we only have a tiny bit of variation in time in academia to work with. If two variables are strongly correlated it can be hard to estimate the independent effects of each on the response and so the standard errors on the coefficients are large. The problem is that you might come to the conclusion from this summary table that a) neither variable has an effect on the response (if you look at the p-values) or that b) the uncertainty on the coefficients are so large that we cannot really say whether either variable has an effect (if you look at the standard errors). In fact, although you might not have much power to estimate their independent effects you may have quite a bit of power to estimate their combined effect. For example, rather than predicting the expected score holding the age constant at 48 lets let age vary at the same time.

Confounding

```
> photo_long$ypub <- 2017 - photo_long$fpub
> photo_m9 <- lm(y ~ type + ypub + age, data = photo_long)

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.48779     3.2926  0.4519 0.654420
typegrumpy   1.28533     0.4362  2.9467 0.005948
ypub        -0.08073     0.1288 -0.6266 0.535349
age          0.09424     0.1280  0.7363 0.466935
```





So these are our estimates of expected values and confidence intervals for someone who started publishing 5 years ago and who is aged 27, and over here for someone who started publishing 45 years ago and is aged 67. You can see that the confidence intervals are reasonably tight now, and perhaps you might be happy claiming that years in academia probably doesn't have major effects on grumpiness, the confidence intervals don't include big shifts in grumpiness across the range of values on the x-axis. So what should we do when we have a situation like this and how do we know we have a situation like this?

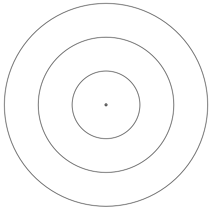
Accuracy and Precision

└ Accuracy and Precision

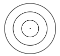
Accuracy and Precision

Before we address this issue I want to briefly discuss two important concepts, accuracy and precision. In every day speech these two words have pretty much the same meaning but in statistics they have very different meanings.

Accuracy and Precision

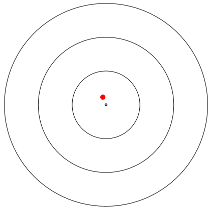


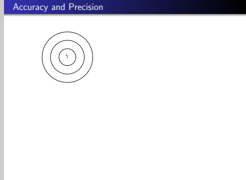
Accuracy and Precision



Imagine you are trying to estimate a pair of parameters, and the centre of this bull's eye represents their true values.

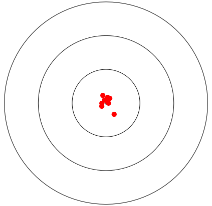
Accuracy and Precision

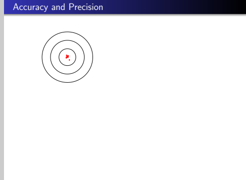




You've then collected some data and obtained estimates of the two parameters. You only have one estimate,

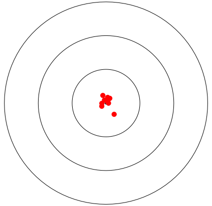
Accuracy and Precision



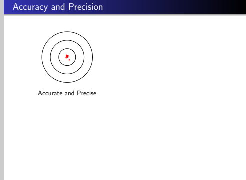


but you could imagine a distribution of possible estimates: the sampling distribution. Its a little different from what we saw before, because previously we thought about the sampling distribution for a single parameter rather than the sampling distribution of a pair of parameters but I hope the plot makes intuitive sense. In this example the sampling distribution is clustered tightly around the true values

Accuracy and Precision

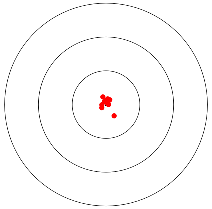


Accurate and Precise

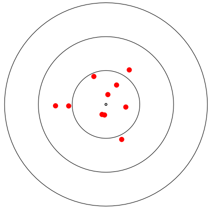


the combination of data you have and the model you have used have resulted in estimates that are accurate (they are actually unbiased - they are on-target) and they are also precise (there's not much variability in the estimates). This is a good position to be in.

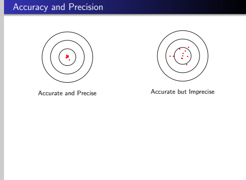
Accuracy and Precision



Accurate and Precise

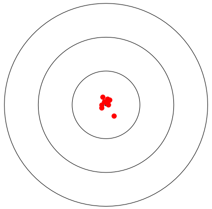


Accurate but Imprecise

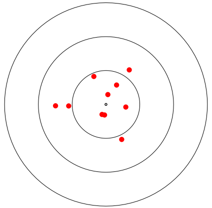


Alternatively, we could have accurate estimates in that the centre of the sampling distribution is aligned with the true values, but the estimates are quite imprecise (there's quite a bit of variability)

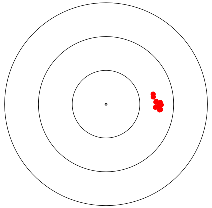
Accuracy and Precision



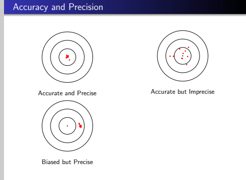
Accurate and Precise



Accurate but Imprecise

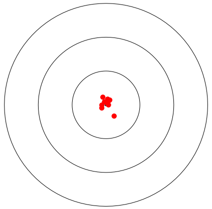


Biased but Precise

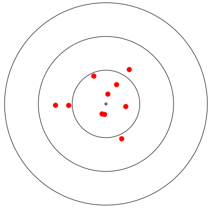


The opposite scenario is that the estimates are inaccurate - they're biased - but they have high precision. In this example the estimates are only biased for the parameter on the x-axis, where as the estimate for the y-axis are both accurate and precise.

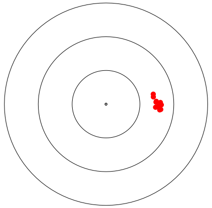
Accuracy and Precision



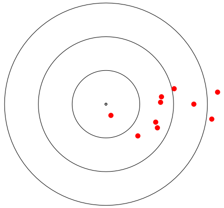
Accurate and Precise



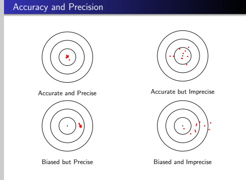
Accurate but Imprecise



Biased but Precise

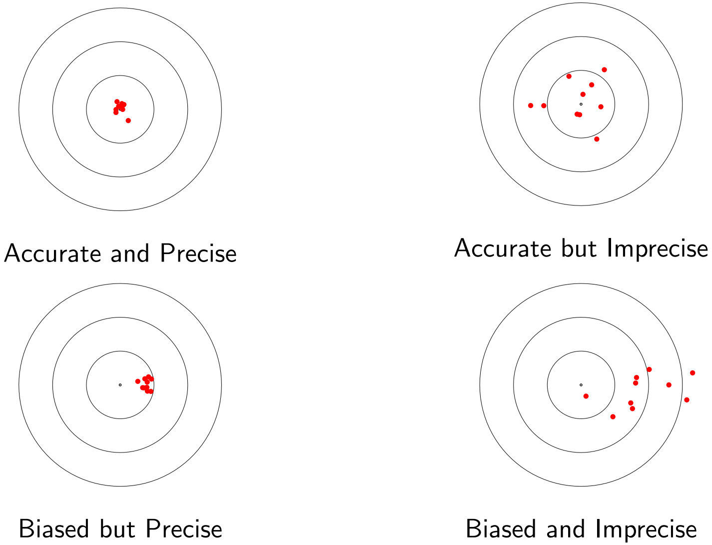


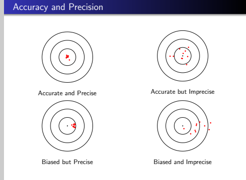
Biased and Imprecise



The worst case scenario is when we have estimates that are both biased and imprecise. Now its easy to see that the top left scenario is ideal and the bottom right scenario is the worst of the four. But what about the other two: what do we care more about - accuracy or precision. In this example, I think most of you would prefer the accurate but imprecise scenario over the biased but precise scenario.

Accuracy and Precision





But what about this? A difficult choice!

Confounding

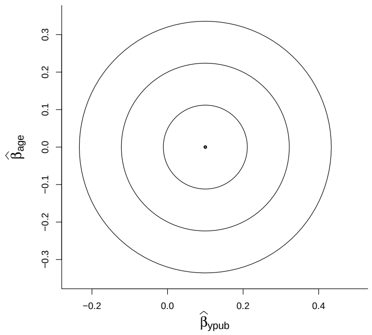
- Imagine that the true slope was 0.1 for `ypub` and 0 for age.

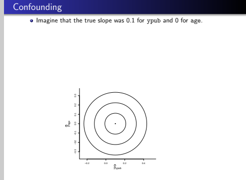
Confounding
- Imagine that the true slope was 0.1 for ypub and 0 for age.

So let's think about our example where age and time in academia are confounded, and let's imagine that in reality there was no effect of age but for every year you have been in academia you are scored 0.1 units more grumpy.

Confounding

- Imagine that the true slope was 0.1 for ypub and 0 for age.

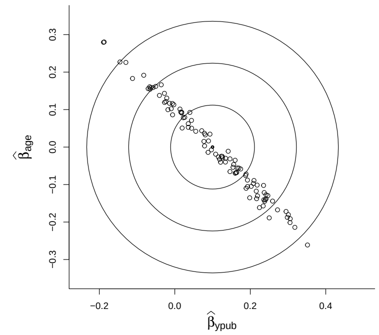


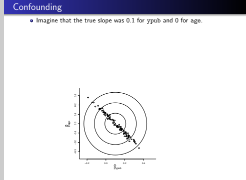


Our bull's eye is at 0 on the y-axis and 0.1 on the x-axis. What do you think the sampling distribution of this pair of parameters looks like?

Confounding

- Imagine that the true slope was 0.1 for ypub and 0 for age.

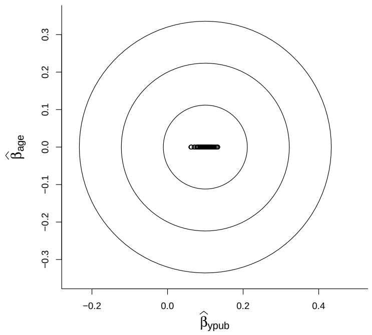


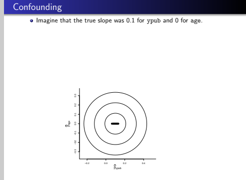


Here I've sampled 100 replicate data sets according to our model and the ML parameter estimates and then I've plotted the estimates made for each replicate data set. You can see that the sampling distribution is strongly negatively correlated. To see why this is the case imagine that the two variables were completely confounded - everybody starts publishing when they are 25 and so there is one to one relationship between age and time in academia. Its clearly impossible then to say which of the two variables is having an effect on the response, but we could estimate their aggregate effect.

Imagine the aggregate effect was represented by a bit of string and there is a mark on it representing how the aggregate effect is actually partitioned between age and time and academia. There's quite a bit of information to estimate the length of the string, so let's imagine we know the aggregate effect (and therefore the length of the string) exactly. Let's also imagine that in this case 1/3 of the effect is due to age and 2/3rds is due to time in academia. We don't have much information to partition the effects of the two variables so our estimates are not going to hit this mark exactly, they have poor precision. Now if you underestimate the mark, let's say you estimate the contribution of age to be 1/6 rather than 1/3 this means that you have overestimated the effect of time in academia by the same margin: $2/3 + 1/6 = 5/6$. So the sampling errors are negatively correlated. If you underestimate the effect of age you will overestimate the effect of time in academia, and vice versa.

- Imagine that the true slope was 0.1 for ypub and 0 for age.

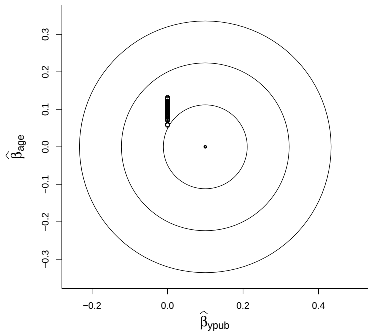


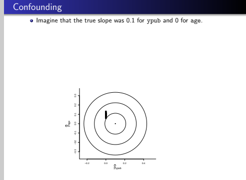


Now let's imagine that I drop age from the model. Wonderful. The estimates are still clustered around their true values but the precision is now much better.

Confounding

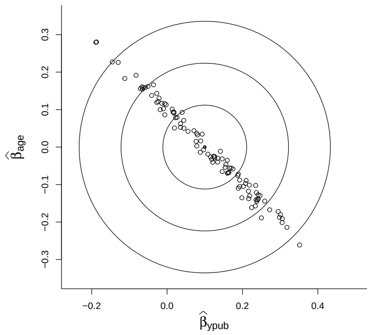
- Imagine that the true slope was 0.1 for ypub and 0 for age.

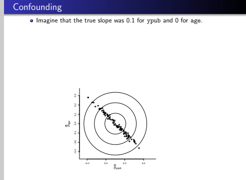




If, on the other hand, we drop ypub from the model its a bit of a disaster. The estimates are nice and precise but they're tremendously biased. We're incorrectly interpreting the ypub effects as age effects.

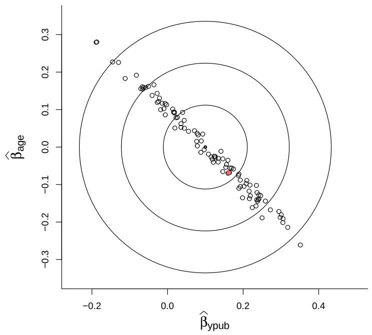
- Imagine that the true slope was 0.1 for ypub and 0 for age.

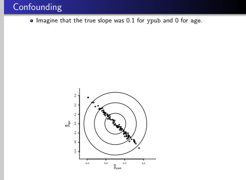




In reality you haven't repeated the study 100 times

- Imagine that the true slope was 0.1 for ypub and 0 for age.

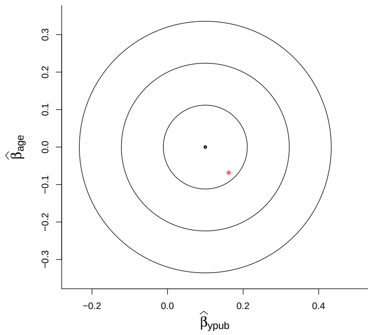


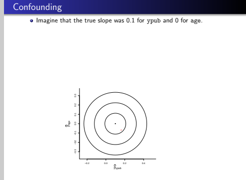


You've just done it once. Let's say this study in red here.

Confounding

- Imagine that the true slope was 0.1 for `ypub` and 0 for age.



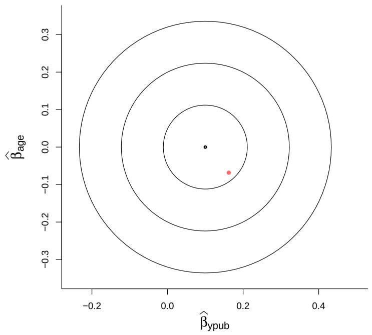


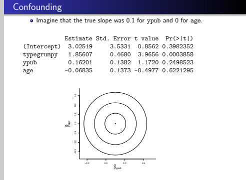
We just have a pair of point estimates.

Confounding

- Imagine that the true slope was 0.1 for ypub and 0 for age.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.02519	3.5331	0.8562	0.3982352
typegrumpy	1.85607	0.4680	3.9656	0.0003858
ypub	0.16201	0.1382	1.1720	0.2498523
age	-0.06835	0.1373	-0.4977	0.6221295



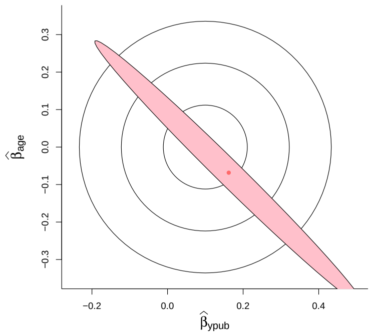


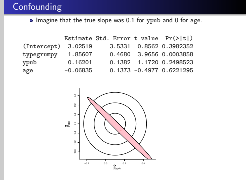
That we can see in the coefficient table. We also have the standard errors that tells us the sampling standard deviation along each axis. There isn't anything in the summary that tells us whether the sampling errors for the two parameters are correlated or not, but we can find that information out.

Confounding

- Imagine that the true slope was 0.1 for ypub and 0 for age.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.02519	3.5331	0.8562	0.3982352
typegrumpy	1.85607	0.4680	3.9656	0.0003858
ypub	0.16201	0.1382	1.1720	0.2498523
age	-0.06835	0.1373	-0.4977	0.6221295



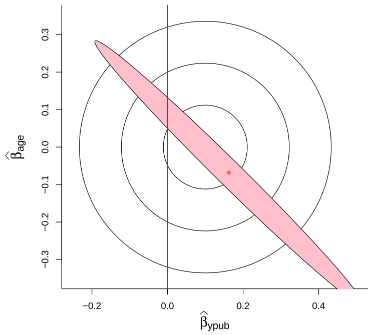

Confounding
Imagine that the true slope was 0.1 for ypub and 0 for age.

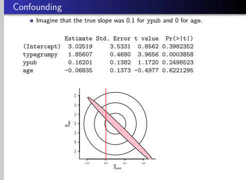
This red ellipse here is a graphical representation of the expected sampling distribution obtained from the model fit. I expect 95% of the estimate to lie within this ellipse if the true values were equal to their maximum likelihood estimates. You can see that its shape is nearly identical to the sampling distribution I obtained via simulation.

Confounding

- Imagine that the true slope was 0.1 for ypub and 0 for age.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.02519	3.5331	0.8562	0.3982352
typegrumpy	1.85607	0.4680	3.9656	0.0003858
ypub	0.16201	0.1382	1.1720	0.2498523
age	-0.06835	0.1373	-0.4977	0.6221295



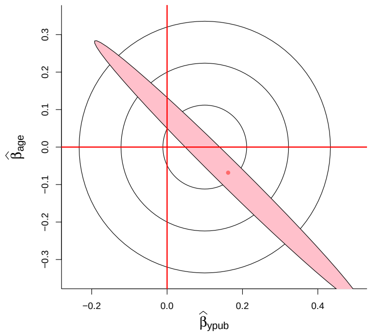


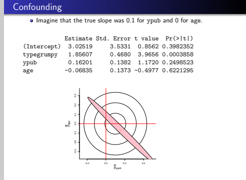
When we test whether the effect of ypub is significant or not, we are asking whether our estimate is likely to overlapped zero, and we can see this to be the case: large fractions of the ellipse lies either side of the vertical red line.

Confounding

- Imagine that the true slope was 0.1 for ypub and 0 for age.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.02519	3.5331	0.8562	0.3982352
typegrumpy	1.85607	0.4680	3.9656	0.0003858
ypub	0.16201	0.1382	1.1720	0.2498523
age	-0.06835	0.1373	-0.4977	0.6221295





Likewise when we test whether the effect of age is significant or not, we are asking whether our estimate is likely to overlap zero, and we can see this is likely: : large fractions of the ellipse lies either side of the horizontal red line.
So what are you going to do in this situation?

Confounding: Diagnosis

- Variance Inflation Factor

Confounding: Diagnosis
- Variance Inflation Factor

The first thing you need to do is to diagnose whether confounding is likely to be an issue or not. Sometimes its obvious - of course there has to be a strong correlation between how old you are and how long you've been publishing - but sometimes its not so obvious. You could have a focal predictor that is only moderately correlated with several other predictors, but in aggregate those other predictors explain a lot of variation in the focal predictor. Variance inflation factors are a good place to start.

Confounding: Diagnosis

- Variance Inflation Factor

```
> car::vif(m1)
typegrumpy       ypub        age
   1.00000   57.61506   57.61506
```


Confounding: Diagnosis
Variance Inflation Factor
> car::vif(m1)
typegrumpy ypub age
1.00000 57.61506 57.61506

They are implemented in the car package

Confounding: Diagnosis

- Variance Inflation Factor

```
> car::vif(m1)
typegrumpy        ypub         age
   1.00000    57.61506    57.61506
```

- Compares the sampling variance to those that would have been observed had the predictors been uncorrelated

Confounding: Diagnosis
- Variance Inflation Factor
> car::vif(m1)
typegrumpy ypub age
1.00000 57.61506 57.61506
- Compares the sampling variance to those that would have been observed had the predictors been uncorrelated

and they say by what factor the sampling variances are increased due to partial confounding with other variables. We can see that the inflation for `ypub` and age is high (with complete confounding they would be infinite) and that the sampling variances are 57 times higher than what they would be had all variables been uncorrelated. The square root of this number ($\sqrt{57} = 7.5$) tells us that our standard errors could be reduced by a factor of 7.5 had we been able to achieve this. The variance inflation factor for grumpy is one, which is the ideal scenario, and this arises because we have used an experimental design; every person was assessed under grumpy and non-grumpy conditions. The one issue with variance inflation factors is that they don't tell you what other variables are causing the inflation. Here we know that the variance inflation for `ypub` is caused by its strong association with age but with more complicated models this might not be so obvious.

Confounding: Diagnosis

- Variance Inflation Factor

```
> car::vif(m1)
typegrumpy        ypub         age
   1.00000    57.61506    57.61506
```

- Compares the sampling variance to those that would have been observed had the predictors been uncorrelated
- Sampling correlations

Confounding: Diagnosis

- Variance Inflation Factor

> car::vif(m1)
typegrumpy ypub age
1.00000 57.61506 57.61506

- Compares the sampling variance to those that would have been observed had the predictors been uncorrelated
- Sampling correlations

The other possibility is to look at the expected sampling correlations between pairs of estimates; strong correlations (either positive or negative) between a pair of estimates tell us that it is hard to separate the effects of those variables on the response.

Confounding: Diagnosis

- Variance Inflation Factor

```
> car::vif(m1)
typegrumpy       ypub        age
   1.00000   57.61506   57.61506
```

- Compares the sampling variance to those that would have been observed had the predictors been uncorrelated

- Sampling correlations

```
> sC <- summary(m1)$cov.unscaled * summary(m1)$sigma^2
> cov2cor(sC)
            (Intercept) typegrumpy ypub    age
(Intercept)  1.0000     -0.0662     0.9651 -0.9889
typegrumpy  -0.0662      1.0000     0.0000 -0.0000
ypub         0.9651      0.0000     1.0000 -0.9913
age         -0.9889     -0.0000    -0.9913  1.0000
```


Confounding: Diagnosis
Variance Inflation Factor
> car::vif(m1)
typegrumpy ypub age
1.00000 57.61506 57.61506
Compares the sampling variance to those that would have been observed had the predictors been uncorrelated
Sampling correlations
> sC <- summary(m1)$cov.unscaled * summary(m1)$sigma^2
> cov2cor(sC)
(Intercept) typegrumpy ypub age
(Intercept) 1.0000 -0.0662 0.9651 -0.9889
typegrumpy -0.0662 1.0000 0.0000 -0.0000
ypub 0.9651 0.0000 1.0000 -0.9913
age -0.9889 -0.0000 -0.9913 1.0000

The ellipse I plotted earlier was essentially a graphical representation of the sampling variances and covariances, which can be extracted from most models. Because the sampling covariances depend on the scale of the predictors[1] it is easier to interpret the correlations,

[1] If the predictor is measured in grams, then the sampling variances are in units of (units of the response per gram)^2 and so may differ a lot between predictors. For example, if the same predictor was measured in kilos, the sampling variance would go down by a factor of a million.

Confounding: Diagnosis

- Variance Inflation Factor

```
> car::vif(m1)
```

```
typegrumpy        ypub         age
   1.00000    57.61506    57.61506
```

- Compares the sampling variance to those that would have been observed had the predictors been uncorrelated
- Sampling correlations

```
> sC <- summary(m1)$cov.unscaled * summary(m1)$sigma^2
> cov2cor(sC)
```

	(Intercept)	typegrumpy	ypub	age
(Intercept)	1.0000	-0.0662	0.9651	-0.9889
typegrumpy	-0.0662	1.0000	0.0000	-0.0000
ypub	0.9651	0.0000	1.0000	-0.9913
age	-0.9889	-0.0000	-0.9913	1.0000

- Correlations large in magnitude indicate pairs of effects that are hard to separate

Confounding: Diagnosis
Variance Inflation Factor
> car::vif(m1)
typegrumpy ypub age
1.00000 57.61506 57.61506
Compares the sampling variance to those that would have been observed had the predictors been uncorrelated
Sampling correlations
> sC <- summary(m1)$cov.unscaled * summary(m1)$sigma^2
> cov2cor(sC)
(Intercept) typegrumpy ypub age
(Intercept) 1.0000 -0.0662 0.9651 -0.9889
typegrumpy -0.0662 1.0000 0.0000 -0.0000
ypub 0.9651 0.0000 1.0000 -0.9913
age -0.9889 -0.0000 -0.9913 1.0000
Correlations large in magnitude indicate pairs of effects that are hard to separate

and we can see that the sampling errors for the grumpy effects are not correlated at all with the `ypub` and `age` (by design) but the sampling errors for `ypub` and `age` are strongly correlated. We can also see that they're strongly correlated with the intercept - and we saw that at the start of this lecture. This is because the intercept is the expected score for happy photos when both `ypub` and `age` are zero, and because the actual joint distribution of `ypub` and `age` are far from these values, small shifts in the slopes drive big changes in the expected values at `ypub`=`age`=0.
The question then is what do we do if we have variables that we think are heavily confounded? There is no silver bullet, but

Confounding: Diagnosis

- Variance Inflation Factor

```
> car::vif(m1)
typegrumpy       ypub        age
   1.00000   57.61506   57.61506
```

- Compares the sampling variance to those that would have been observed had the predictors been uncorrelated
- Sampling correlations

```
> sC <- summary(m1)$cov.unscaled * summary(m1)$sigma^2
> cov2cor(sC)
            (Intercept) typegrumpy ypub    age
(Intercept)  1.0000     -0.0662     0.9651 -0.9889
typegrumpy  -0.0662      1.0000     0.0000 -0.0000
ypub         0.9651      0.0000     1.0000 -0.9913
age         -0.9889     -0.0000    -0.9913  1.0000
```

- Correlations large in magnitude indicate pairs of effects that are hard to separate

└Confounding: Diagnosis

Confounding: Diagnosis
Variance Inflation Factor
> car::vif(m1)
typegrumpy ypub age
1.00000 57.61506 57.61506
Compares the sampling variance to those that would have been observed had the predictors been uncorrelated
Sampling correlations
> sC <- summary(m1)$cov.unscaled * summary(m1)$sigma^2
> cov2cor(sC)
(Intercept) typegrumpy ypub age
(Intercept) 1.0000 -0.0662 0.9651 -0.9889
typegrumpy -0.0662 1.0000 0.0000 -0.0000
ypub 0.9651 0.0000 1.0000 -0.9913
age -0.9889 -0.0000 -0.9913 1.0000
Correlations large in magnitude indicate pairs of effects that are hard to separate

and we can see that the sampling errors for the grumpy effects are not correlated at all with the ypub and age (by design) but the sampling errors for ypub and age are strongly correlated. We can also see that they're strongly correlated with the intercept - and we saw that at the start of this lecture. This is because the intercept is the expected score for happy photos when both ypub and age are zero, and because the actual joint distribution of ypub and age are far from these values, small shifts in the slopes drive big changes in the expected values at ypub=age=0.
The question then is what do we do if we have variables that we think are heavily confounded? There is no silver bullet, but

Confounding: Solutions

Select age or `fpub` effects

Confounding: Solutions
Select age or `fpub` effects

The first possibility is to only use one of the two predictors from a pair that are strongly correlated. If I knew in advance of fitting the model that two predictors are likely to be so strongly correlated that separating their effects is not worth attempting, I would probably make a decision in advance of model fitting and choose the variable that I think is most important biologically.

Select age or `fpub` effects

- Retain the most biologically plausible variable and be honest ('we could not reliably separate the effects of `ypub` from age')

Confounding: Solutions
Select age or fpub effects
Retain the most biologically plausible variable and be honest ('we could not reliably separate the effects of ypub from age')

You could of course do this after fitting the model - so you could retain the effect of years publishing if you think this is more likely to be the driver than age - but in both cases I would be honest and say the effect could also be driven by age but you didn't have the power to separate them.

Confounding: Solutions

Select age or fpub effects

- Retain the most biologically plausible variable and be honest ('we could not reliably separate the effects of ypub from age')

```
      Estimate Std. Error t value   Pr(>|t|)
ypub   0.09382      0.018   5.211 9.895e-06
```

- Fit both independently and retain the model with highest likelihood and be honest (because you could have selected the wrong term)

Confounding: Solutions
Select age or fpub effects
Retain the most biologically plausible variable and be honest ('we could not reliably separate the effects of ypub from age')
Estimate Std. Error t value Pr(>|t|)
ypub 0.09382 0.018 5.211 9.895e-06
Fit both independently and retain the model with highest likelihood and be honest (because you could have selected the wrong term)

Rather than selecting a variable based on biological intuition you could let the computer do it for you. So you could fit two models, one containing age as a predictor and one containing ypub as a predictor and select the model with the highest likelihood. However, as before you have to be honest about the difficulty of separating the two effects because the likelihoods might be very similar and it would be easy to select the wrong model just by chance. For example, in our simulated data set we set the ypub coefficient to 0.1 and the age coefficient to 0, yet here the model returning the highest likelihood is by chance actually the one with age fitted.

Select age or fpub effects

- Retain the most biologically plausible variable and be honest ('we could not reliably separate the effects of ypub from age')

```
     Estimate Std. Error t value  Pr(>|t|)
ypub  0.09382      0.018   5.211 9.895e-06
```

- Fit both independently and retain the model with highest likelihood and be honest (because you could have selected the wrong term)

```
    Estimate Std. Error t value  Pr(>|t|)
age  0.09121     0.0182   5.013 1.777e-05
```


Confounding: Solutions

Select age or fpub effects

Retain the most biologically plausible variable and be honest ('we could not reliably separate the effects of ypub from age')

Estimate Std. Error t value Pr(>|t|)
ypub 0.09382 0.018 5.211 9.895e-06

Fit both independently and retain the model with highest likelihood and be honest (because you could have selected the wrong term)

Estimate Std. Error t value Pr(>|t|)
age 0.09121 0.0182 5.013 1.777e-05

Rather than selecting a variable based on biological intuition you could let the computer do it for you. So you could fit two models, one containing age as a predictor and one containing ypub as a predictor and select the model with the highest likelihood. However, as before you have to be honest about the difficulty of separating the two effects because the likelihoods might be very similar and it would be easy to select the wrong model just by chance. For example, in our simulated data set we set the ypub coefficient to 0.1 and the age coefficient to 0, yet here the model returning the highest likelihood is by chance actually the one with age fitted.

Confounding: Solutions

Be agnostic about age or ypub effects

Confounding: Solutions
Be agnostic about age or ypub effects

The second option is to retain both predictors, and test whether either predictor has an effect on the response, without caring which one is the driving variable.

Confounding: Solutions

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$

Confounding: Solutions

Be agnostic about age or ypub effects

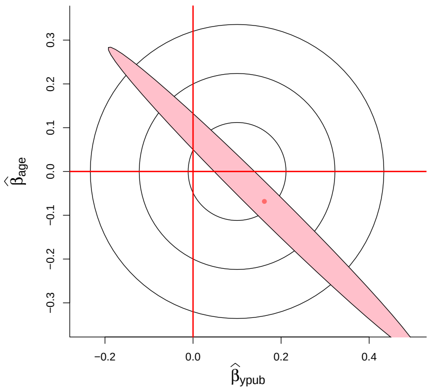
- Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$

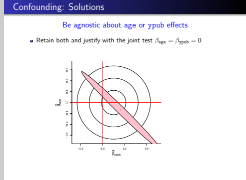
The null hypothesis is then that both regression coefficients are zero.

Confounding: Solutions

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$

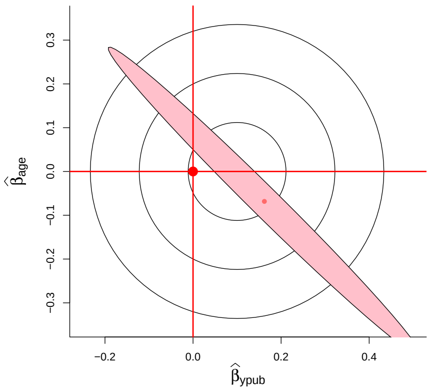


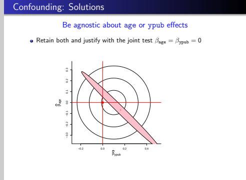


We plotted this graph earlier; the small red dot is our estimate from the simulated data, and the red ellipse depicted the sampling distribution of the estimates around the estimate. 95% of estimates should fall within the ellipse if the true value was equal to the estimated value.

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$

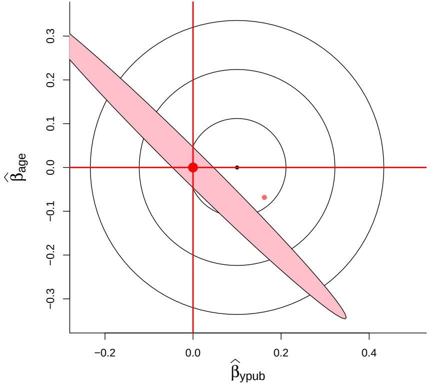


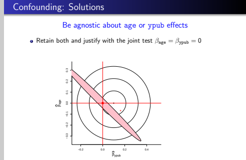


Our null-hypothesis is that both coefficients are zero, which is this larger red dot. In a linear model the shape of the sampling distribution does not change with the mean (with other types of model this is only true as the sample size becomes large, hence the tests are approximate)

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$





and so this ellipse now describes the sampling distribution had the true effects been zero. You can see that our estimated value lies outside of the 95% probability region and is therefore significant at the 5% level.

Confounding: Solutions

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
- F-test: Multi-parameter version of the t-test.

Confounding: Solutions

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
- F-test: Multi-parameter version of the t-test.

When testing a single parameter we saw that the t-test is exact when the response variable is normal, but the z-test (which ignores estimation uncertainty in the residual standard deviation) is usually very accurate unless sample sizes are pitiful. The multi-parameter analogue of the t-test is the F-test,

Confounding: Solutions

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
- F-test: Multi-parameter version of the t-test.

```
> anova(update(m1, . ~ . - age - ypub), m1)
      Pr(>F)
5.944669e-05
```


Confounding: Solutions
Be agnostic about age or ypub effects
Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
F-test: Multi-parameter version of the t-test.
> anova(update(m1, . ~ . - age - ypub), m1)
Pr(>F)
5.944669e-05

which can be performed using the function `anova` and comparing the full model with a model with the terms to be tested deleted. I've done this using the function update which takes our original model and fits a new model including everything in the original model (the dot) but with `ypub` and `age` removed (by having a minus sign).

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
- F-test: Multi-parameter version of the t-test.

```
> anova(update(m1, . ~ . - age - ypub), m1)
     Pr(>F)
5.944669e-05
```

- Wald test: Multi-parameter version of the z-test.

Confounding: Solutions
Be agnostic about age or ypub effects
Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
F-test: Multi-parameter version of the t-test.
> anova(update(m1, . ~ . - age - ypub), m1)
Pr(>F)
5.944669e-05
Wald test: Multi-parameter version of the z-test.

For more complicated problems the sampling distribution for a set of parameters is not known, but we might know that as sample sizes increase the sampling distribution will start to look multivariate normal, with known (co)variances. In the context of a t-test this would be like setting the degrees of freedom to be very very high indicating that we know the residual standard deviation exactly. When testing a single parameter this is known as a Z-test, and the Wald-test is the multiple parameter equivalent.

Confounding: Solutions

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$
- F-test: Multi-parameter version of the t-test.

```
> anova(update(m1, . ~ . - age - ypub), m1)
      Pr(>F)
5.944669e-05
```

- Wald test: Multi-parameter version of the z-test.

```
> aod::wald.test(sC, coef(m1), Terms = 3:4)
            P
1.526501e-06
```


Confounding: Solutions
Be agnostic about age or ypub effects
Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$
F-test: Multi-parameter version of the t-test.
> anova(update(m1, . ~ . - age - ypub), m1)
Pr(>F)
5.944669e-05
Wald test: Multi-parameter version of the z-test.
> aod::wald.test(sC, coef(m1), Terms = 3:4)
P
1.526501e-06

We can fit a Wald test using the function `wald.test`[1] from the aod package. We give it the matrix of sampling (co)variances for our parameters, which we obtained earlier (`sC`), and our point estimates (using the function `coef`) and then the positions of the effects we want to test (positions 3 and 4 refer to `ypub` and `age` respectively). In relative terms the p-values are quite discrepant (the p-value of the Wald-test is about 39 times lower) but this is because the t and normal distribution differ most in their tails. Had the estimates been less extreme, lying in the main body of the sampling distribution under the null-hypothesis, their p-values would be less discrepant.

For example, if the estimates were halved in magnitude then we would have obtained the p-values 0.05 and 0.04 for the F-test and the Wald test respectively.

[1] Note that the function `wald.test` can also perform F-tests if the residual degrees of freedom (the sample size minus the number of coefficients in the model) is specified and the `anova` function can also perform Wald-tests when the argument `test="Chisq"` is passed. However, in some cases an `anova` method might not be written for the model fitting function you use, and so it is good to see how you can do it 'by hand'.

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
- F-test: Multi-parameter version of the t-test.

```
> anova(update(m1, . ~ . - age - ypub), m1)
      Pr(>F)
5.944669e-05
```

- Wald test: Multi-parameter version of the z-test.

```
> aod::wald.test(sC, coef(m1), Terms = 3:4)
            P
1.526501e-06
```

- Likelihood-ratio test:

Confounding: Solutions
Be agnostic about age or ypub effects
Retain both and justify with the joint test $\beta_{\texttt{age}} = \beta_{\texttt{ypub}} = 0$
F-test: Multi-parameter version of the t-test.
> anova(update(m1, . ~ . - age - ypub), m1)
Pr(>F)
5.944669e-05
Wald test: Multi-parameter version of the z-test.
> aod::wald.test(sC, coef(m1), Terms = 3:4)
P
1.526501e-06
Likelihood-ratio test:

We could also compare the two models using a likelihood-ratio test, which again is an approximation that improves as the information in a data-set about the parameter to be tested increases.

Confounding: Solutions

Be agnostic about age or ypub effects

- Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$
- F-test: Multi-parameter version of the t-test.

```
> anova(update(m1, . ~ . - age - ypub), m1)
      Pr(>F)
5.944669e-05
```

- Wald test: Multi-parameter version of the z-test.

```
> aod::wald.test(sC, coef(m1), Terms = 3:4)
           P
1.526501e-06
```

- Likelihood-ratio test:

```
> anova(update(m1, . ~ . - age - ypub), m1, test = "LRT")
     Pr(>Chi)
1.526501e-06
```


Confounding: Solutions
Be agnostic about age or ypub effects
Retain both and justify with the joint test $\beta_{\text{age}} = \beta_{\text{ypub}} = 0$
F-test: Multi-parameter version of the t-test.
> anova(update(m1, . ~ . - age - ypub), m1)
Pr(>F)
5.944669e-05
Wald test: Multi-parameter version of the z-test.
> aod::wald.test(sC, coef(m1), Terms = 3:4)
P
1.526501e-06
Likelihood-ratio test:
> anova(update(m1, . ~ . - age - ypub), m1, test = "LRT")
Pr(>Chi)
1.526501e-06

We can view this as a form of model comparison, and again we can pass our full model and reduced model to `anova` and specify that we want to to a likelihood ratio test (`test="LRT"`)[1].

[1] You can see that the likelihood ratio test (`test="LRT"`) and z-test (`test="chisq"`) give identical p-values. When the parameter to be tested is a regression coefficient in a linear model, the tests are equivalent but this is because `anova` doesn't actually fit what I would call a standard likelihood ratio test. Lets say we we were fitting a simple model with an intercept β and residual standard deviation σ. Lets subscript parameter estimates from the null model with a zero, so $\hat{\sigma}_0$ and $\hat{\beta}_0$ which is fixed at zero. We'll subscript with a one the parameters from the full model: $\hat{\sigma}_1$ and $\hat{\beta}_1$ where $\hat{\beta}_1$ is free to take any value. The 'standard' likelihood ratio test compares the likelihood of the data under $\hat{\sigma}_0$ and $\hat{\beta}_0 = 0$ (so `dnorm(data, 0,` $\hat{\sigma}_0$`))`) with that under $\hat{\sigma}_1$ and $\hat{\beta}_1$ (so `dnorm(data,` $\hat{\beta}_1$, $\hat{\sigma}_1$`))`). The likelihood ratio test performed by `anova` actually uses the likelihood under $\hat{\sigma}_0$ and $\hat{\beta}_1$ (so `dnorm(data,` $\hat{\beta}_1$, $\hat{\sigma}_0$`))`). Doing a standard likelihood ratio test gives a slightly different answer:

Confounding: Sequential tests

Confounding: Sequential tests

A word of caution is required at this point, because the F-test that we have done asks whether age and/or `ypub` explain significant variation in grumpiness scores after accounting for *all* other terms in the model. The single-parameter version of this is the t-test results presented in the summary table, and we saw earlier that age explains no additional variation after accounting for `ypub` and vice-versa. Sometimes people refer to this test as a type-III test.

Confounding: Sequential tests

```
> anova(m1)
           Df Sum Sq Mean Sq F value    Pr(>F)
typegrumpy  1 31.005  31.005 15.7258 0.0003858
ypub        1 52.321  52.321 26.5374 1.279e-05
age         1  0.488   0.488  0.2477 0.6221295
Residuals  32 63.091   1.972
```


Confounding: Sequential tests

```
> anova(m1)
           Df Sum Sq Mean Sq F value    Pr(>F)
typegrumpy  1 31.005  31.005 15.7258 0.0003858
ypub        1 52.321  52.321 26.5374 1.279e-05
age         1  0.488   0.488  0.2477 0.6221295
Residuals  32 63.091   1.972
```


However, if you just pass a model to the function `anova` without an accompanying simplified model, it will perform a sequential test which asks whether a predictor explains significant variation after accounting for any *previous* terms in the model. Sometimes people refer to this as an incremental or type-I test. So in our model `ypub` appeared in the formula prior to `age`, and so the sequential test first tests for the effect `ypub` after accounting for the effect of being grumpy or not, and then second tests whether `age` has an effect after accounting for the effect `ypub`. You can see that `ypub` has a significant effect, but after accounting for it, `age` does not.

Confounding: Sequential tests

```
> anova(m1)
           Df Sum Sq Mean Sq F value    Pr(>F)
typegrumpy  1 31.005  31.005 15.7258 0.0003858
ypub        1 52.321  52.321 26.5374 1.279e-05
age         1  0.488   0.488  0.2477 0.6221295
Residuals  32 63.091   1.972
```

```
> anova(update(m1, . ~ . - ypub - age + age + ypub))
           Df Sum Sq Mean Sq F value    Pr(>F)
typegrumpy  1 31.005  31.005 15.7258 0.0003858
age         1 50.102  50.102 25.4115 1.764e-05
ypub        1  2.708   2.708  1.3736 0.2498523
Residuals  32 63.091   1.972
```


You could also reverse the order of the terms in the model, so here I've updated our original model by including everything in the original model (the dot) then removed `ypub` and `age` (by having a minus sign) and then added them back in (by having a plus sign) but in reverse order. Now we test for the effect of `age` after accounting for the effect of being grumpy or not, and then after accounting for the effect of `age` we test whether `ypub` has an effect. We come to the opposite conclusion, but at least we understand why.

In R, Type-I tests will always test the main effects prior to any interaction terms, no matter which order they are specified in (for example, applying anova to the model `grumpy+fpub+grumpy:fpub` gives the same output as `grumpy:fpub+grumpy+fpub` with the interaction being tested last. An intermediate type of test is a type-II test which is implemented in the function `Anova` from the `car` package. This is a useful but underused test. Here, all main effects are added simultaneously, and all two way interactions are added simultaneously, then three-ways and so on. This offers a nice way of testing for the main effects without making arbitrary choices about which value of the covariate to evaluate them at, which we saw earlier. However, it doesn't force you to sequentially test main effects. For example, a Type-II test of the model `grumpy+ypub+age+ypub:grumpy` first tests `grumpy`, `ypub` and `age` simultaneously (and presumably finds that neither have a significant effect) and then tests whether the interaction term can explain additional variation in the response.

Accuracy and Precision

Accuracy and Precision

By analysing a model where two of the predictor variables are heavily confounded, we touched on a number of themes that might influence the accuracy and precision of our results. More generally,

Accuracy and Precision

Low Precision

- Small sample size

Accuracy and Precision
Low Precision
- Small sample size

We can give a summary of those things that will reduce the precision of our estimates, the most obvious being small sample sizes.

Low Precision

- Small sample size
- Predictors not very variable

Accuracy and Precision
Low Precision
- Small sample size
- Predictors not very variable

Also, if our predictor variable wasn't very variable it would be hard to get accurate estimates of the effect of that variable. For example imagine I wanted to test whether the height of people affects some outcome, and the heights of the people I chose to study only ranged from 179cm to 180cm. It would be hard to get accurate estimates of the effect compared to picking people with a wider range of heights.

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented

Accuracy and Precision
Low Precision
- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented

In the context of categorical predictors lack of variability implies most observations are only in one group. For example if we were interested in whether a new diet had some impact a study would not be very powerful if we only had 5 people on the diet, irrespective of whether we had a million controls.

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded

Accuracy and Precision
Low Precision
Small sample size
Predictors not very variable
Little variation in continuous predictors
Levels of a categorical predictor not equally represented
Predictors confounded

As we saw confounding can severely affect precision

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented

Accuracy and Precision
Low Precision
Small sample size
Predictors not very variable
Little variation in continuous predictors
Levels of a categorical predictor not equally represented
Predictors confounded
Little independent variation in continuous predictors
Combinations of levels not equally represented

because essentially it reduces the amount of *independent* variation in the predictor variables, so if they are strongly correlated in the case of continuous variables, or in the case of categorical variables if combinations of levels are not equally represented. For example, if those on the diet were nearly all men, but most of the controls were women.

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation

Accuracy and Precision
Low Precision
- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation

Finally, high residual variation reduces precision because its hard to detect differences when there is a lot of noise.

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Accuracy and Precision
Low Precision
Small sample size
Predictors not very variable
Little variation in continuous predictors
Levels of a categorical predictor not equally represented
Predictors confounded
Little *independent* variation in continuous predictors
Combinations of levels not equally represented
High residual variation
Conditions not standardised experimentally
Conditions not standardised statistically

In an experimental setting you might be able to lower the noise by trying to control conditions as carefully as possible, but sometimes you can try and control for the noise statistically. For example,if people were measured before being put on the diet and then again after, looking for *differences* between time-points controls for any noise that affects an individual at all time points. Later we'll see how this can be done with mixed-effect models.

Accuracy and Precision

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Bias

Accuracy and Precision
Low Precision
- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically
Bias

Bias is often a more difficult problem to fix, particularly in a non-experimental setting.

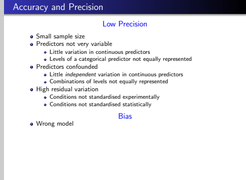
Accuracy and Precision

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Bias

- Wrong model



For example, you could have measured all relevant variables but had fitted the wrong model to the data.

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Bias

- Wrong model
- Unmeasured variables

Accuracy and Precision
Low Precision
Small sample size
Predictors not very variable
Little variation in continuous predictors
Levels of a categorical predictor not equally represented
Predictors confounded
Little *independent* variation in continuous predictors
Combinations of levels not equally represented
High residual variation
Conditions not standardised experimentally
Conditions not standardised statistically
Bias
Wrong model
Unmeasured variables

More commonly, there are probably predictor variables out there that affect the response variable and that you haven't measured which are also correlated with a predictor of interest. The effect of the predictor is then biased by this unmeasured variable, as we saw in the simulated data when we fitted `age` instead of `ypub`.

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Bias

- Wrong model
- Unmeasured variables
  - No Control
  - No Randomisation

Accuracy and Precision
Low Precision
Small sample size
Predictors not very variable
Little variation in continuous predictors
Levels of a categorical predictor not equally represented
Predictors confounded
Little independent variation in continuous predictors
Combinations of levels not equally represented
High residual variation
Conditions not standardised experimentally
Conditions not standardised statistically
Bias
Wrong model
Unmeasured variables
No Control
No Randomisation

This is much less likely to happen in an experimental setting, where employing controls and randomisation can ensure that an experimental treatment is not correlated with some unmeasured variable.

Accuracy and Precision

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Bias

- Wrong model
- Unmeasured variables
  - No Control
  - No Randomisation
- Poorly measured variables

Accuracy and Precision
Low Precision
- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically
Bias
- Wrong model
- Unmeasured variables
  - No Control
  - No Randomisation
- Poorly measured variables

Lastly it can happen when either the data or predictor variables have been measured poorly.

Accuracy and Precision

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Bias

- Wrong model
- Unmeasured variables
  - No Control
  - No Randomisation
- Poorly measured variables
  - Predictors measured with error

Accuracy and Precision
Low Precision
Small sample size
Predictors not very variable
Little variation in continuous predictors
Levels of a categorical predictor not equally represented
Predictors confounded
Little *independent* variation in continuous predictors
Combinations of levels not equally represented
High residual variation
Conditions not standardised experimentally
Conditions not standardised statistically
Bias
Wrong model
Unmeasured variables
No Control
No Randomisation
Poorly measured variables
Predictors measured with error

If predictors have been measured with error then we tend to underestimate the true effect

Accuracy and Precision

Low Precision

- Small sample size
- Predictors not very variable
  - Little variation in continuous predictors
  - Levels of a categorical predictor not equally represented
- Predictors confounded
  - Little *independent* variation in continuous predictors
  - Combinations of levels not equally represented
- High residual variation
  - Conditions not standardised experimentally
  - Conditions not standardised statistically

Bias

- Wrong model
- Unmeasured variables
  - No Control
  - No Randomisation
- Poorly measured variables
  - Predictors measured with error
  - Predictors/response missing not at random

Accuracy and Precision
Low Precision
Small sample size
Predictors not very variable
Little variation in continuous predictors
Levels of a categorical predictor not equally represented
Predictors confounded
Little *independent* variation in continuous predictors
Combinations of levels not equally represented
High residual variation
Conditions not standardised experimentally
Conditions not standardised statistically
Bias
Wrong model
Unmeasured variables
No Control
No Randomisation
Poorly measured variables
Predictors measured with error
Predictors/response missing not at random

and we get both positive and negative bias if either the response variable and/or the predictors haven't been measured on some individuals and the probability of not being measured depends on what their response would have been.