Model selection

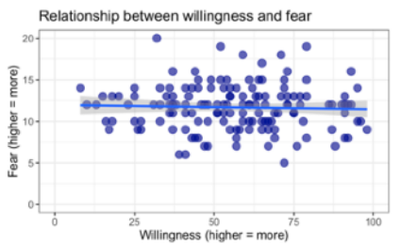
Research Methods for Human Inquiry
Andrew Perfors

Remember our problem

What's going on here? This doesn't seem to make sense

```
> modelWFK <- lm(willingness ~ fear + knowledge, data=db)
> summary(modelWFK)

Call:
lm(formula = willingness ~ fear + knowledge, data = db)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 68.28414    6.07014  11.249  < 2e-16 ***
fear        -3.65566    0.60976  -5.995  1.6e-08 ***
knowledge    0.69486    0.07221   9.622  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘~’ 0.1 ‘’ 1

Residual standard error: 16.42 on 142 degrees of freedom
Multiple R-squared:  0.3956,  Adjusted R-squared:  0.3871
F-statistic: 46.47 on 2 and 142 DF,  p-value: 2.977e-16
```

??



Remember our problem

What's going on here? This doesn't seem to make sense

One possibility is that our pictures were for each pair of variables independent of the others, but our regression is finding the best-fit line across *all of them at once*

Let's look at different regression models then!

modelWFK

Outcome: willingness
Predictors: fear*** and knowledge***, no interaction

```
> modelWFK <- lm(willingness ~ fear + knowledge, data=db)
> summary(modelWFK)

Call:
lm(formula = willingness ~ fear + knowledge, data = db)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 68.28414    6.07014  11.249  < 2e-16 ***
fear        -3.65566    0.60976  -5.995  1.6e-08 ***
knowledge    0.69486    0.07221   9.622  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘˜’ 0.1 ‘’ 1

Residual standard error: 16.42 on 142 degrees of freedom
Multiple R-squared:  0.3956,  Adjusted R-squared:  0.3871
F-statistic: 46.47 on 2 and 142 DF,  p-value: 2.977e-16
```

modelWF

Outcome: willingness
Predictors: fear alone

```
> modelWF <- lm(willingness ~ fear, data=db)
> summary(modelWF)

Call:
lm(formula = willingness ~ fear, data = db)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  59.6807     7.6899   7.761 1.46e-12 ***
fear         -0.3001     0.6407  -0.468     0.64
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.03 on 143 degrees of freedom
Multiple R-squared:  0.001532,  Adjusted R-squared:  -0.00545
F-statistic: 0.2195 on 1 and 143 DF,  p-value: 0.6402
```

modelWK

Outcome: willingness
Predictors: knowledge*** alone

```
> modelWK <- lm(willingness ~ knowledge, data=db)
> summary(modelWK)

Call:
lm(formula = willingness ~ knowledge, data = db)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.46242    3.28526  11.099  < 2e-16 ***
knowledge    0.44725    0.06608   6.768 3.11e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.31 on 143 degrees of freedom
Multiple R-squared:  0.2426,  Adjusted R-squared:  0.2373
F-statistic: 45.81 on 1 and 143 DF,  p-value: 3.112e-10
```

modelWFKi

Outcome: willingness

Predictors: fear*** and knowledge**, with interaction

```
> modelWFKi <- lm(willingness ~ fear*knowledge, data=db)
> summary(modelWFKi)

Call:
lm(formula = willingness ~ fear * knowledge, data = db)

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    68.6976414 11.2852526   6.087 1.03e-08 ***
fear           -3.6925733  1.0456995  -3.531 0.000559 ***
knowledge       0.6856926  0.2227172   3.079 0.002499 **
fear:knowledge  0.0007657  0.0175915   0.044 0.965343
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.47 on 141 degrees of freedom
Multiple R-squared:  0.3956,  Adjusted R-squared:  0.3828
F-statistic: 30.76 on 3 and 141 DF,  p-value: 2.316e-15
```

We have four models… which is right?

modelWFK

```
willingness ~ fear + knowledge

fear*** (p<.0001)
knowledge** (p=0.002)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3871

F(2,142) = 30.76, p<.0001
```

modelWFKi

```
willingness ~ fear*knowledge

fear*** (p<.0001)
knowledge** (p=.002)
Interaction (p=.965)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3828

F(3,141)=30.76, p<.0001
```

modelWF

```
willingness ~ fear

fear (p=0.64)

Multiple R-squared:  0.001532
Adjusted R-squared:  -0.00545

F(1,143)=0.22, p=0.64
```

modelWK

```
willingness ~ knowledge

knowledge*** (p<.0001)

Multiple R-squared:  0.2426
Adjusted R-squared:  0.2373

F(1,143)=45.81, p<.0001
```

Maybe whichever explains the most variance?

modelWFK

```
willingness ~ fear + knowledge

fear*** (p<.0001)
knowledge** (p=0.002)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3871

F(2,142) = 30.76, p<.0001
```

modelWFKi

```
willingness ~ fear*knowledge

fear*** (p<.0001)
knowledge** (p=.002)
Interaction (p=.965)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3828

F(3,141)=30.76, p<.0001
```

modelWF

```
willingness ~ fear

fear (p=0.64)

Multiple R-squared:  0.001532
Adjusted R-squared:  -0.00545

F(1,143)=0.22, p=0.64
```

modelWK

```
willingness ~ knowledge

knowledge*** (p<.0001)

Multiple R-squared:  0.2426
Adjusted R-squared:  0.2373

F(1,143)=45.81, p<.0001
```

Maybe whichever explains the most variance?

No — in general, adding predictors to a model will *always* increase R^2 (the worst they can do is to keep it the same) so choosing the model with the highest R^2 means always choosing the one with the most predictors

```
> db$v <- rnorm(n=145,mean=10,sd=5)
```

Created additional variable v which is literally random with respect to any of the others

```
> modelWFiKiVi <- lm(willingness ~ fear*knowledge*v, data=db)
> summary(modelWFiKiVi)
```

This is a model that now contains v in it plus all interactions

Maybe whichever explains the most variance?

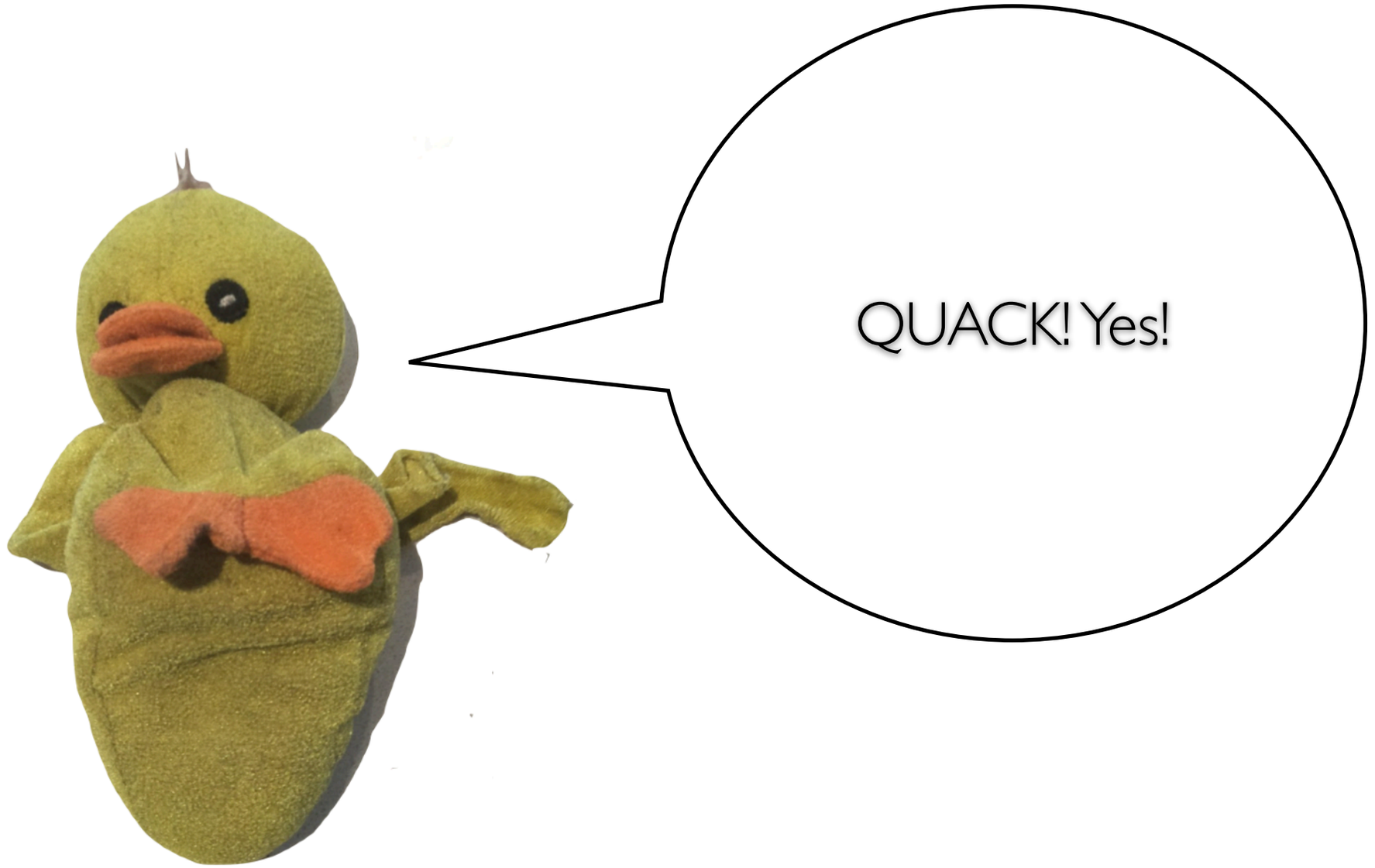
This has an even higher R^2 even though the additional predictor was random and is not significant!

```
lm(formula = willingness ~ fear * knowledge * v, data = db)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     103.035496  25.199251   4.089 7.35e-05 ***
fear             -6.367513   2.309542  -2.757  0.00663 **
knowledge         0.375644   0.435073   0.863  0.38942
v                -2.958544   2.233550  -1.325  0.18751
fear:knowledge    0.023103   0.033273   0.694  0.48863
fear:v            0.216834   0.201787   1.075  0.28446
knowledge:v       0.022145   0.045239   0.490  0.62527
fear:knowledge:v -0.001283   0.003524  -0.364  0.71632
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.47 on 137 degrees of freedom
Multiple R-squared:  0.413,   Adjusted R-squared:  0.383
F-statistic: 13.77 on 7 and 137 DF,  p-value: 2.042e-13
```

Previous best: 0.3956

What's going on?

And why is this bad? Shouldn't we always want the model that explains the most variance? i.e., that fits the data the best?

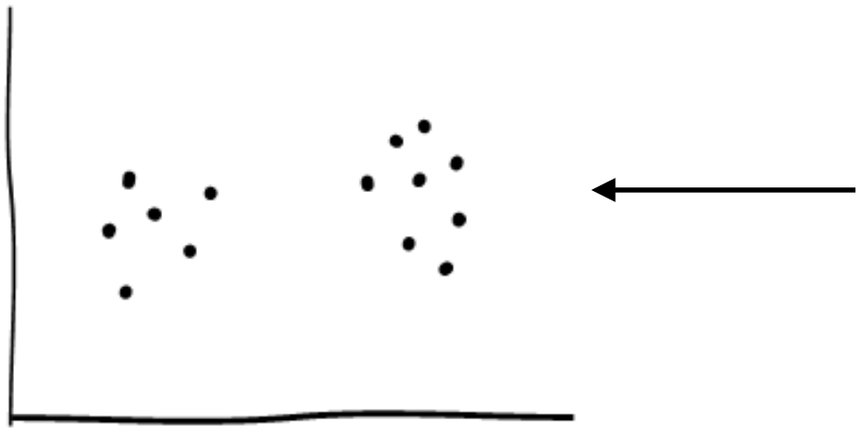


What's going on?

And why is this bad? Shouldn't we always want the model that explains the most variance? i.e., that fits the data the best?

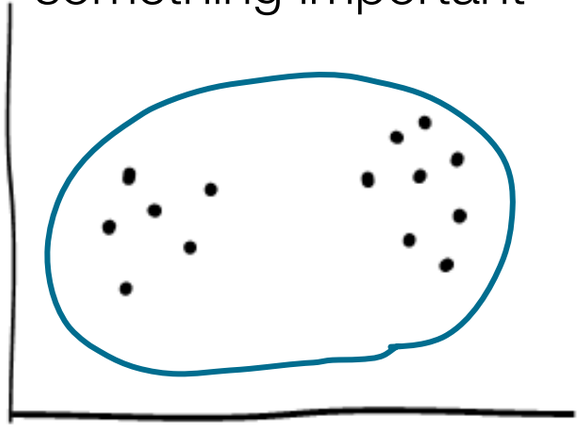
The problem with overfitting

The problem with "too complex" models is that they **overfit** — intuitively, the additional parameters are just capturing some of the noise in the data. That means they aren't really describing the *actual* relationships (which is what we want to do)

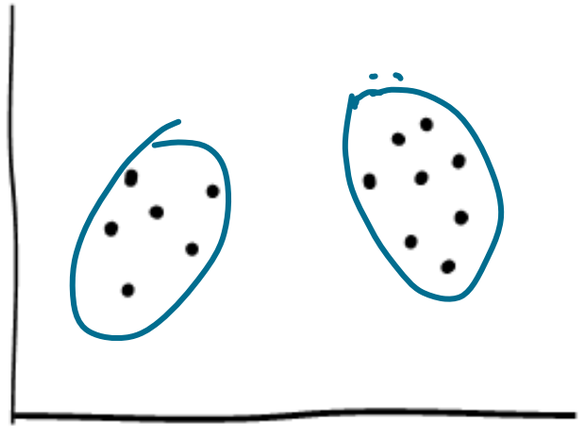


Suppose your data looks like this, and you want to know what kind of underlying process generated it

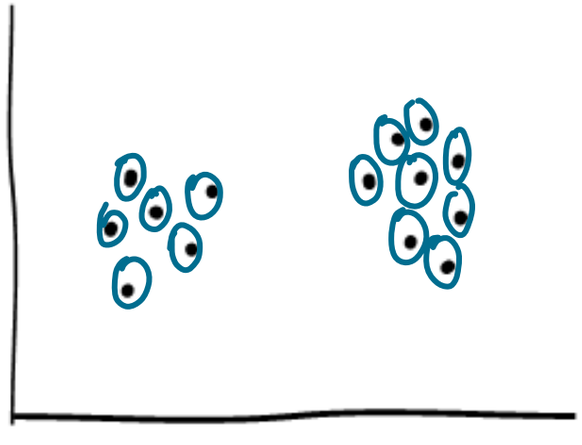
This seems a bit simple, like it's missing something important



This seems about right



This fits best of all but is also missing important regularities

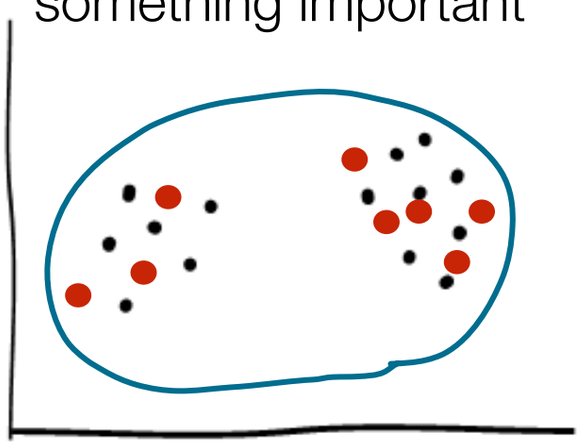


The problem with overfitting

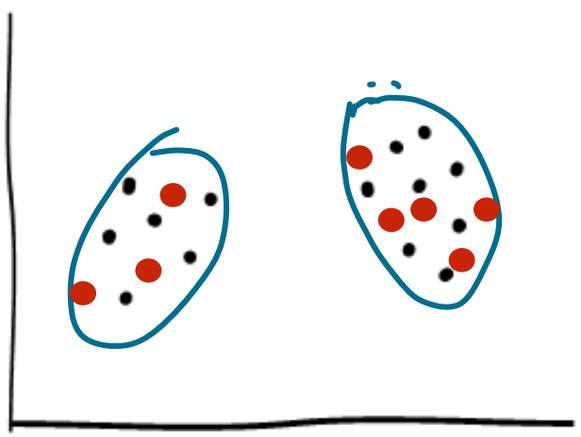
The problem with "too complex" models is that they **overfit** — intuitively, the additional parameters are just capturing some of the noise in the data. That means they aren't really describing the *actual* relationships (which is what we want to do)

And when you generate **new data**, the overfit model can't explain it

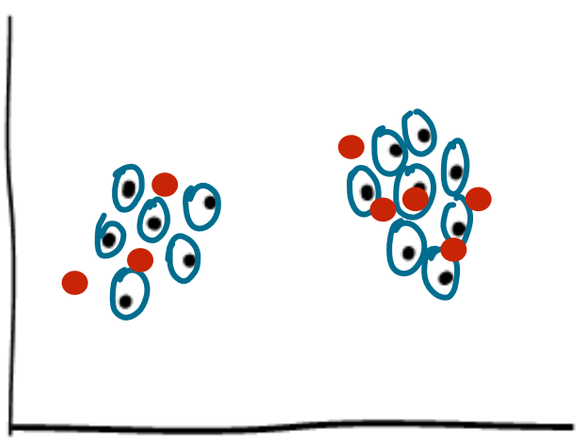
This seems a bit simple, like it's missing something important



This seems about right



This fits best of all but is also missing important regularities



How do we know how complex is too complex?

This is a hugely difficult problem and there is no one obvious solution! The best is to actually test on new data, but we don't always have that, and it's a bit tricky to do right. A technique called **cross-validation** lets us approximate it but is beyond the scope of this subject.

Something similar that often works is to **penalise models for additional parameters**.

Penalising model complexity: adjusted R^2

Penalises additional parameters: $$\text{adj. } R^2 = 1 - \left(\frac{\text{SS}_{res}}{\text{SS}_{tot}} \times \frac{N-1}{N-K-1} \right)$$

modelWFK

```
willingness ~ fear + knowledge

fear*** (p<.0001)
knowledge** (p=0.002)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3871

F(2,142) = 30.76, p<.0001
```

modelWFKi

```
willingness ~ fear*knowledge

fear*** (p<.0001)
knowledge** (p=.002)
Interaction (p=.965)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3828

F(3,141)=30.76, p<.0001
```

modelWF

```
willingness ~ fear

fear (p=0.64)

Multiple R-squared:  0.001532
Adjusted R-squared:  -0.00545

F(1,143)=0.22, p=0.64
```

modelWK

```
willingness ~ knowledge

knowledge*** (p<.0001)

Multiple R-squared:  0.2426
Adjusted R-squared:  0.2373

F(1,143)=45.81, p<.0001
```

Penalising model complexity: adjusted R^2

Seems reasonable in this case, but I don't like it as much as the other two I'm going to show you because I don't think it penalises enough

modelWFK

```
willingness ~ fear + knowledge

fear*** (p<.0001)
knowledge** (p=0.002)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3871

F(2,142) = 30.76, p<.0001
```

modelWFKi

```
willingness ~ fear*knowledge

fear*** (p<.0001)
knowledge** (p=.002)
Interaction (p=.965)

Multiple R-squared:  0.3956
Adjusted R-squared:  0.3828

F(3,141)=30.76, p<.0001
```

modelWF

```
willingness ~ fear

fear (p=0.64)

Multiple R-squared:  0.001532
Adjusted R-squared:  -0.00545

F(1,143)=0.22, p=0.64
```

modelWK

```
willingness ~ knowledge

knowledge*** (p<.0001)

Multiple R-squared:  0.2426
Adjusted R-squared:  0.2373

F(1,143)=45.81, p<.0001
```

Penalising model complexity: AIC and BIC

Both of these are widely used. They just penalise slightly differently. I prefer BIC because I find in general it more often performs in accordance to my intuitions, I think because it scales better with sample size

AIC: Akaike Information Criterion

$$\mathrm{AIC} = \frac{\mathrm{SS}_{res}}{\hat{\sigma}}^2 + 2K$$

How big the residuals are relative to your variance

Two times the # of parameters in your model

BIC: Bayesian Information Criterion

$$\mathrm{BIC} = k\ln(n) - 2\ln(\widehat{L}).$$

The number of parameters times the natural log of the sample size

Essentially a measure of how well the model fits (similar to the residual term in AIC)

* Actually the one I usually recommend is LOOIC, which is the same idea but uses a leave-one-out approach, but it's complicated to do and BIC is a good simple approximation so that's where we'll focus here

Penalising model complexity: AIC and BIC

They are easy to use in R! The functions are `AIC()` and `BIC()`. For each, you give them the model objects and the one with the **lowest value is best**

```
> AIC(modelWF,modelWK,modelWFK,modelWFKi,modelWFiKiVi)
             df      AIC
modelWF       3 1298.736
modelWK       3 1258.664
modelWFK      4 1227.946
modelWFKi     5 1229.944
modelWFiKiVi  9 1233.718
```

```
> BIC(modelWF,modelWK,modelWFK,modelWFKi,modelWFiKiVi)
             df      BIC
modelWF       3 1307.666
modelWK       3 1267.594
modelWFK      4 1239.853
modelWFKi     5 1244.828
modelWFiKiVi  9 1260.508
```

What does this tell us?

All of these analyses suggest that the “best” model is the one with knowledge and fear as predictors, but no interaction. So we’ll go ahead and interpret our results using that model

```
> modelWFK <- lm(willingness ~ fear + knowledge, data=db)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 68.28414    6.07014  11.249  < 2e-16 ***
fear        -3.65566    0.60976  -5.995  1.6e-08 ***
knowledge    0.69486    0.07221   9.622  < 2e-16 ***

> standardCoefs(modelWFK)
                     b       beta
fear      -3.6556640 -0.4768069
knowledge  0.6948593  0.7652595
```

This suggests that fear contributes independently from knowledge to the willingness to work together. Knowledge accounts for so much variance that if it’s not included in the model, it swamps the effect of fear. But when it is, the effect of fear can be picked up.

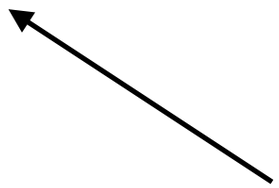
As it happens, this is very close to the actual model I used to generate the data, so our regression worked pretty well!

Doing model comparisons

Note: you probably don't want to throw every possible model into your `AIC()` or `BIC()` functions. They aren't infallible, and you don't want to overinterpret tiny differences in AIC or BIC.

In particular, resist the temptation to include overly complex models that you don't know how to interpret. AIC and BIC "break" more with more complexity. And the whole reason for doing this is to understand what's going on. So if you can't make sense of the model, don't bother.

```
modelWFiKiVi <- lm(willingness ~ fear*knowledge*v, data=db)
```



This is a great example of an uninterpretable model we should probably not have bothered with

Writing this up

As with writing up regressions in general, there is no simple formula. You need to think about what information is relevant based on what you did and the results you obtained. Often model comparisons are reported in a table, and then the winning model is interpreted in the text.

We compared four linear regression models which used willingness as an outcome variable, but varied in the predictor variables (fear, knowledge, and the presence of absence of an interaction). Table 1 shows each model along with the obtained AIC and BIC values, which penalise model complexity. The best-performing model, shown in bold, contained both fear and knowledge as predictor variables but had no interaction term.

Model name	Model description	AIC	BIC
WF	willingness ~ fear	1298.7	1307.7
WK	willingness ~ knowledge	1258.7	1267.6
WFK	**willingness ~ fear + knowledge**	**1227.9**	**1239.8**
WFKi	willingness ~ fear + knowledge + fear:knowledge	1229.9	1244.8

Writing this up

As with writing up regressions in general, there is no simple formula. You need to think about what information is relevant based on what you did and the results you obtained. Often model comparisons are reported in a table, and then the winning model is interpreted in the text.

In the best-performing model, the outcome variable was each person's willingness to work with people from the other place, and the two predictors were that person's level of fear and that person's level of knowledge about people from the other place. This model was statistically significant, $F(2,142)=46.47$, $p<.0001$, and accounted for 39.6% of the variance in the willingness to work. Fear was a significant predictor, $t(142)=-5.99$, $p<.0001$, with a raw coefficient of -3.66, suggesting that holding knowledge constant, each additional unit of fear was associated with a decrease in willingness of 3.66 units. Knowledge as also significant, $t(142)=9.62$, $p<.0001$, with a raw coefficient of 0.69, suggesting that holding fear constant, each additional unit of knowledge was associated with an increase in willingness of 0.69 units. Standardised coefficients (fear: -0.48, knowledge: 0.77) suggested that the effect of knowledge was nearly twice as large as that of fear.

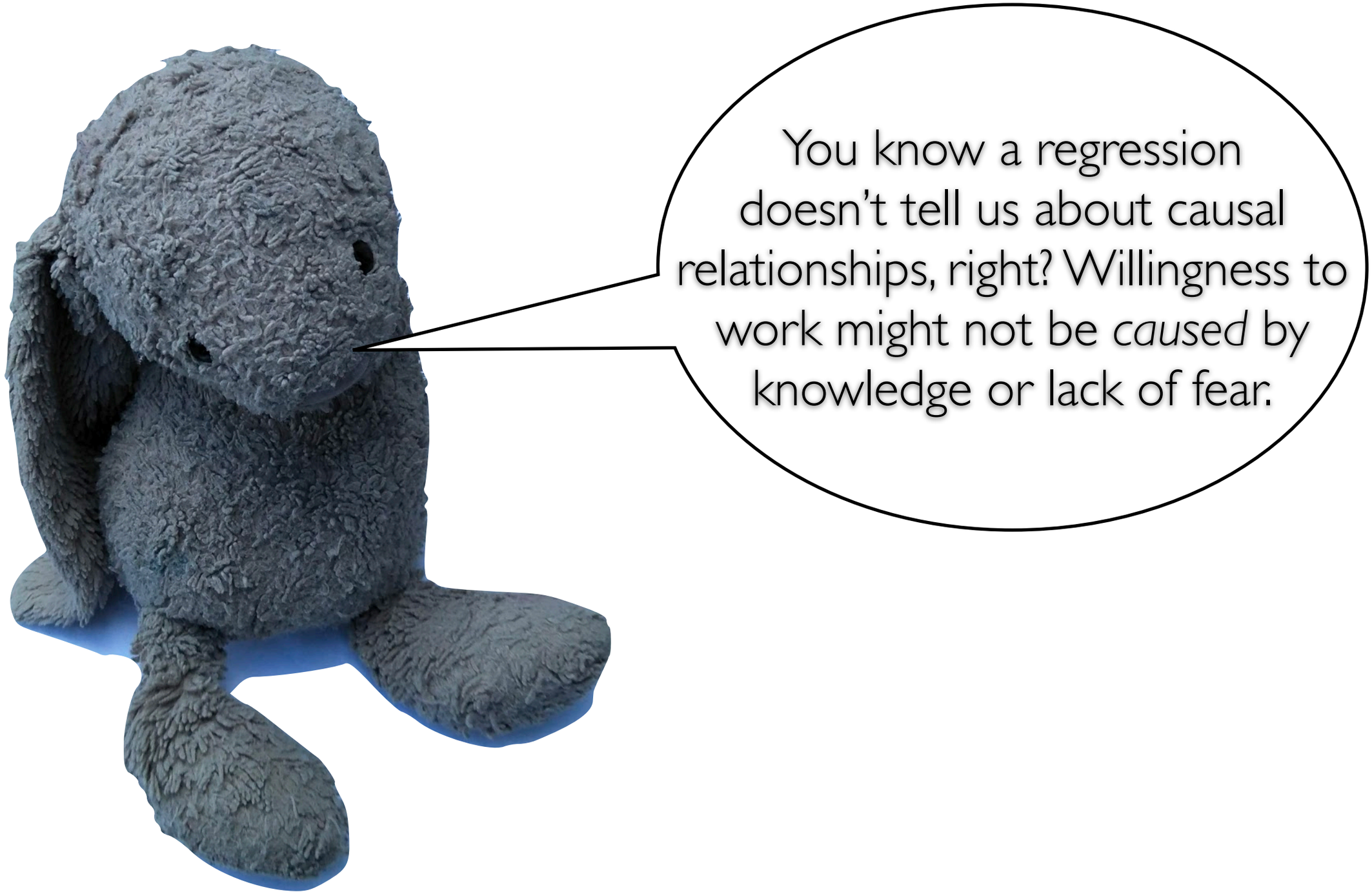
Writing this up

(It's also good to mention somewhere what tests you did to test model assumptions, but often that will go into a footnote or supplemental materials or something).

So what now?

If willingness to work together is affected by both fear and knowledge, maybe we can improve things by learning about each other

So what now?


You know a regression doesn't tell us about causal relationships, right? Willingness to work might not be *caused* by knowledge or lack of fear.

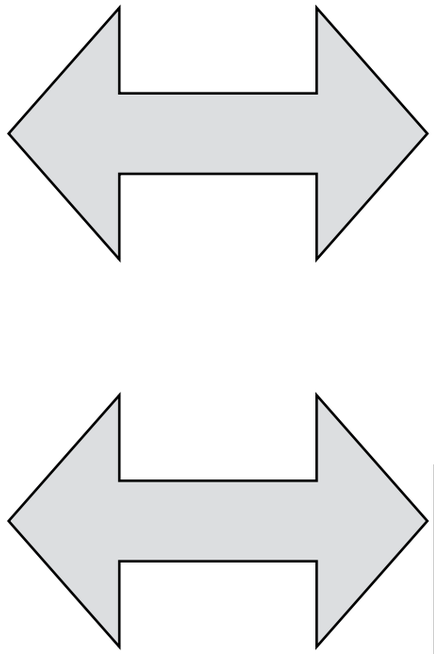
So what now?

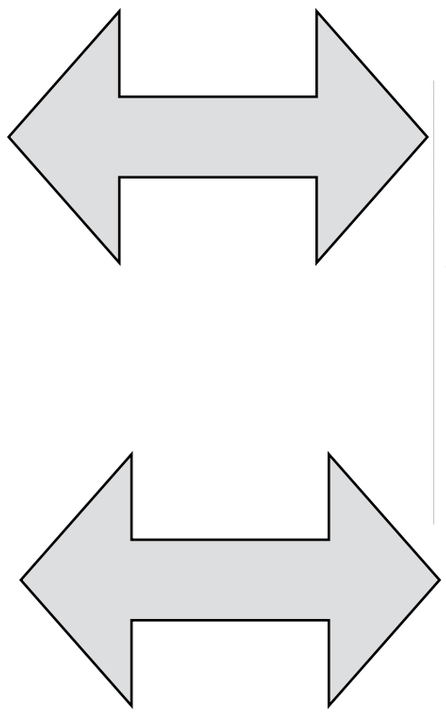
True. But it doesn't mean there *isn't* one either. We can at least *try* an intervention. It might work, and it also might tell us something about the causality.

So they make a “person exchange”

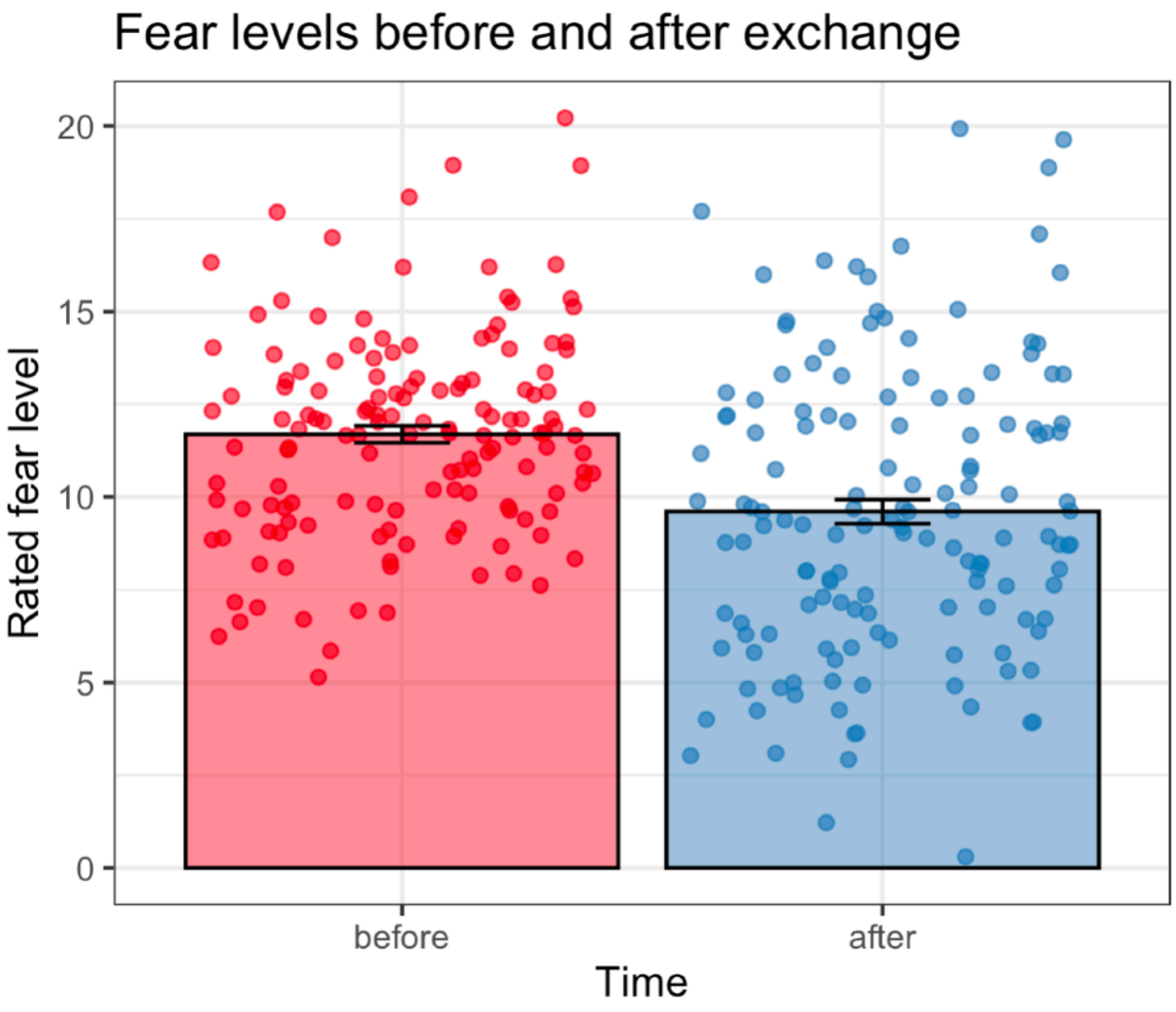
Bunnyland

Otherland

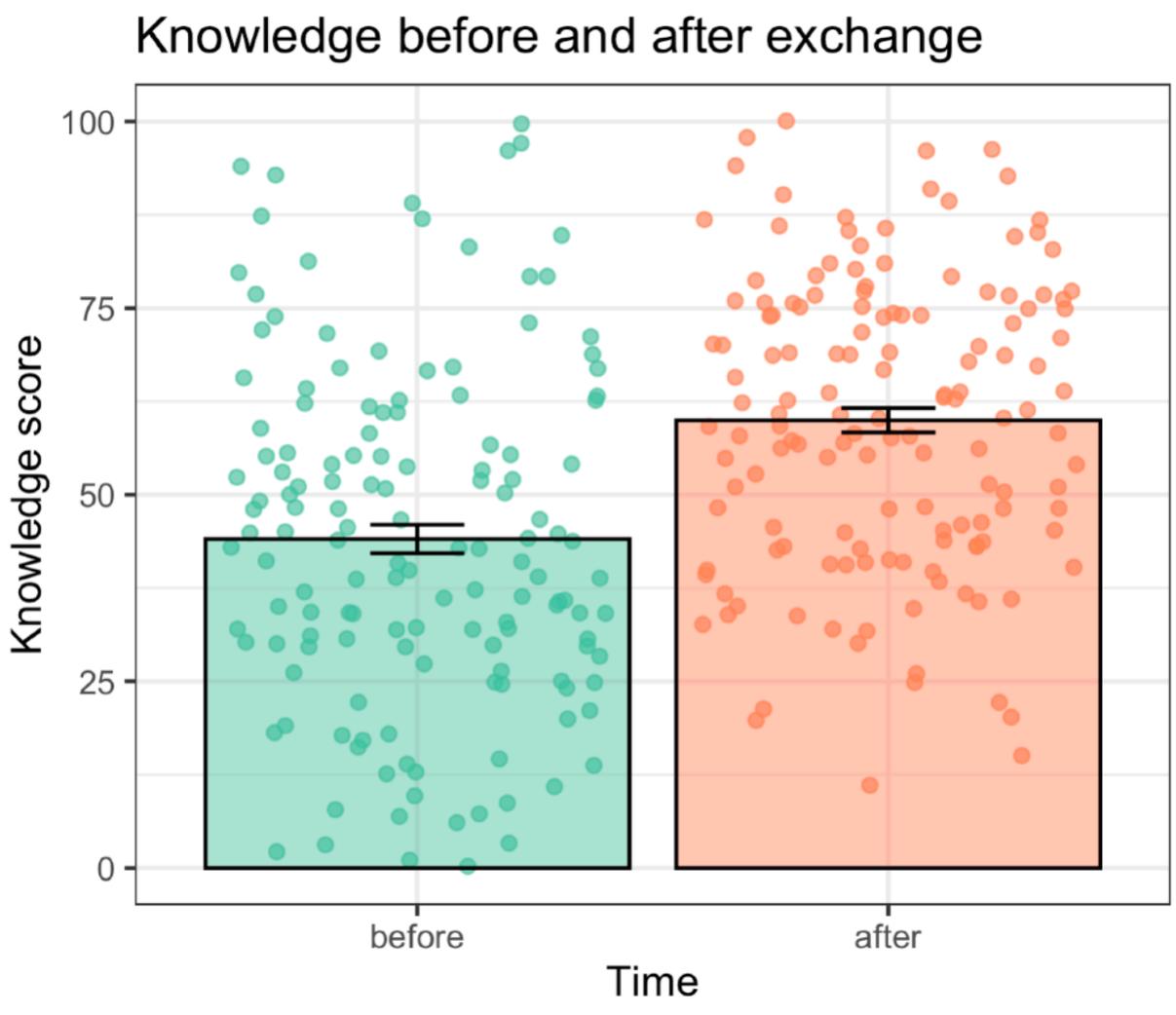




Afterward, things have changed!

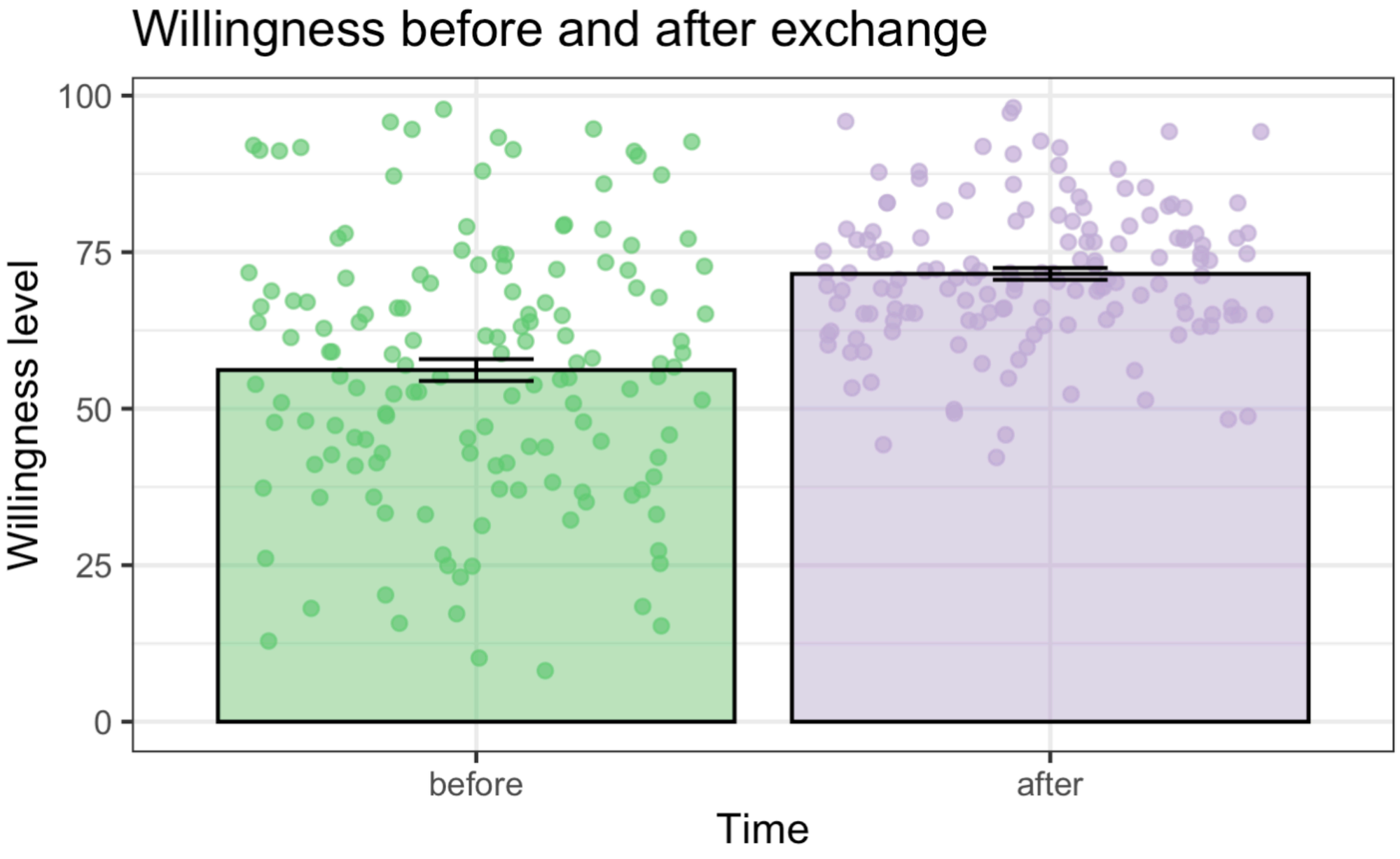


Fear has gone down



Knowledge has gone up

Afterward, things have changed!



And willingness to work together has gone way up!

You'll analyse that data in…

Exercises are in `w10daylexercises.Rmd`