Regression assumptions

Research Methods for Human Inquiry
Andrew Perfors

Last time

Learned a lot about how to do and interpret regressions, but glossed over assumption checking. That's the topic here

Warning: A lot of this can't be turned into a tidy procedure. You have to use your judgment. I'm also skipping over some things you could do. Chapter 15 in *Learning Statistics with R* has more. The point of this is to tell you some of the main things!

Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!

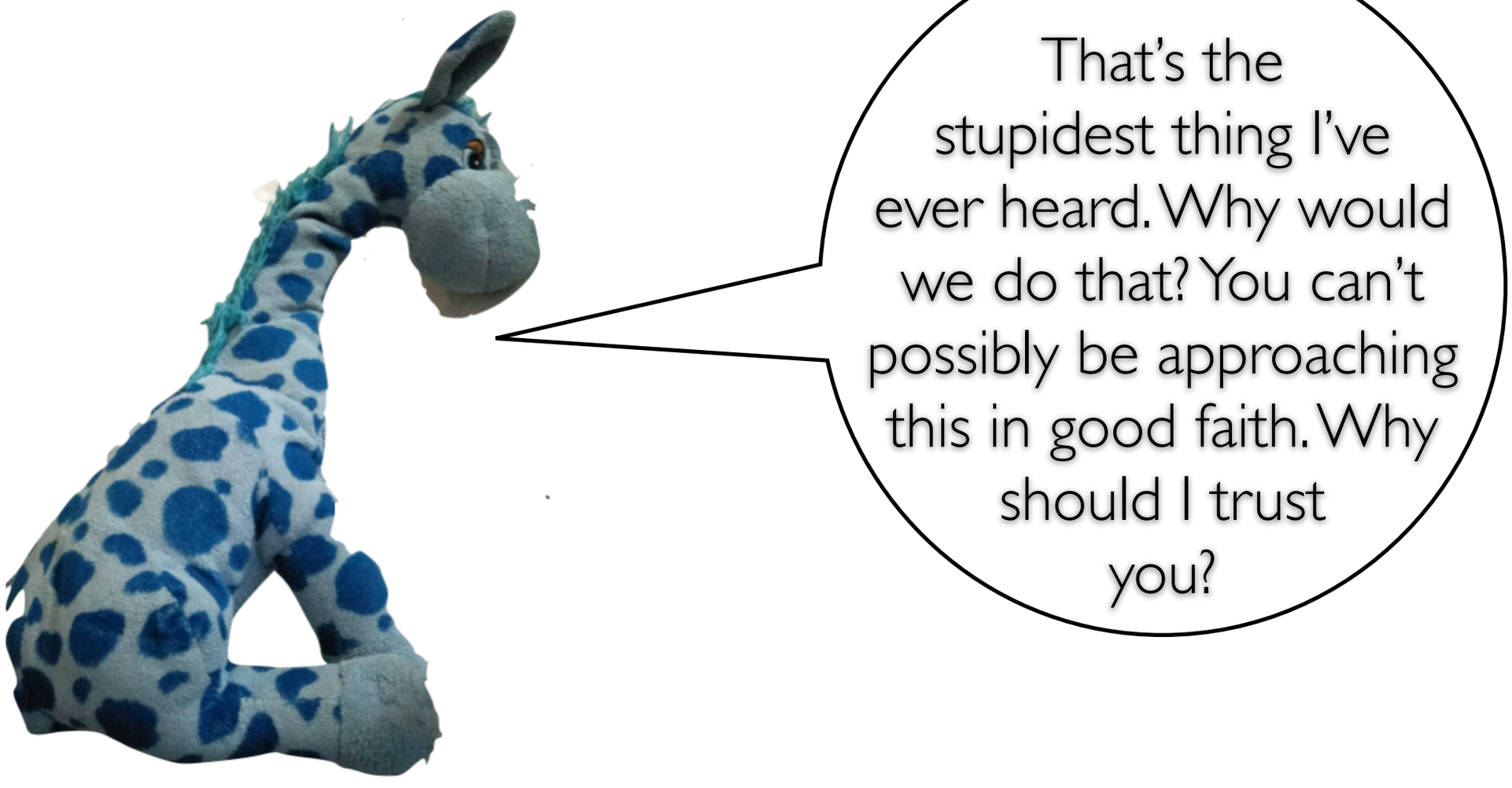
Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!

That's a total lie, but I've heard that people in Bunnyland sneak across the border and steal our babies. That's probably what LFB and Foxy were actually doing!

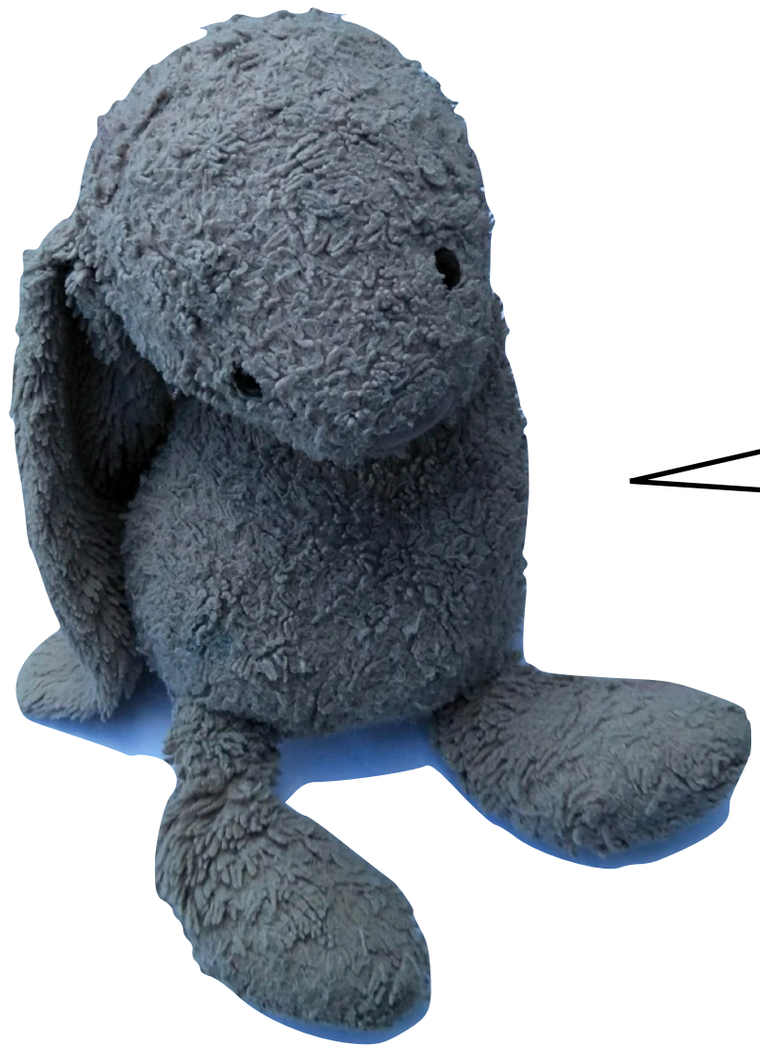
Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!



Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!



Oh please stop fighting!!!!!!

Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!

Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!

I was really scared of the Others before I met them. Maybe fear is driving a lot of this

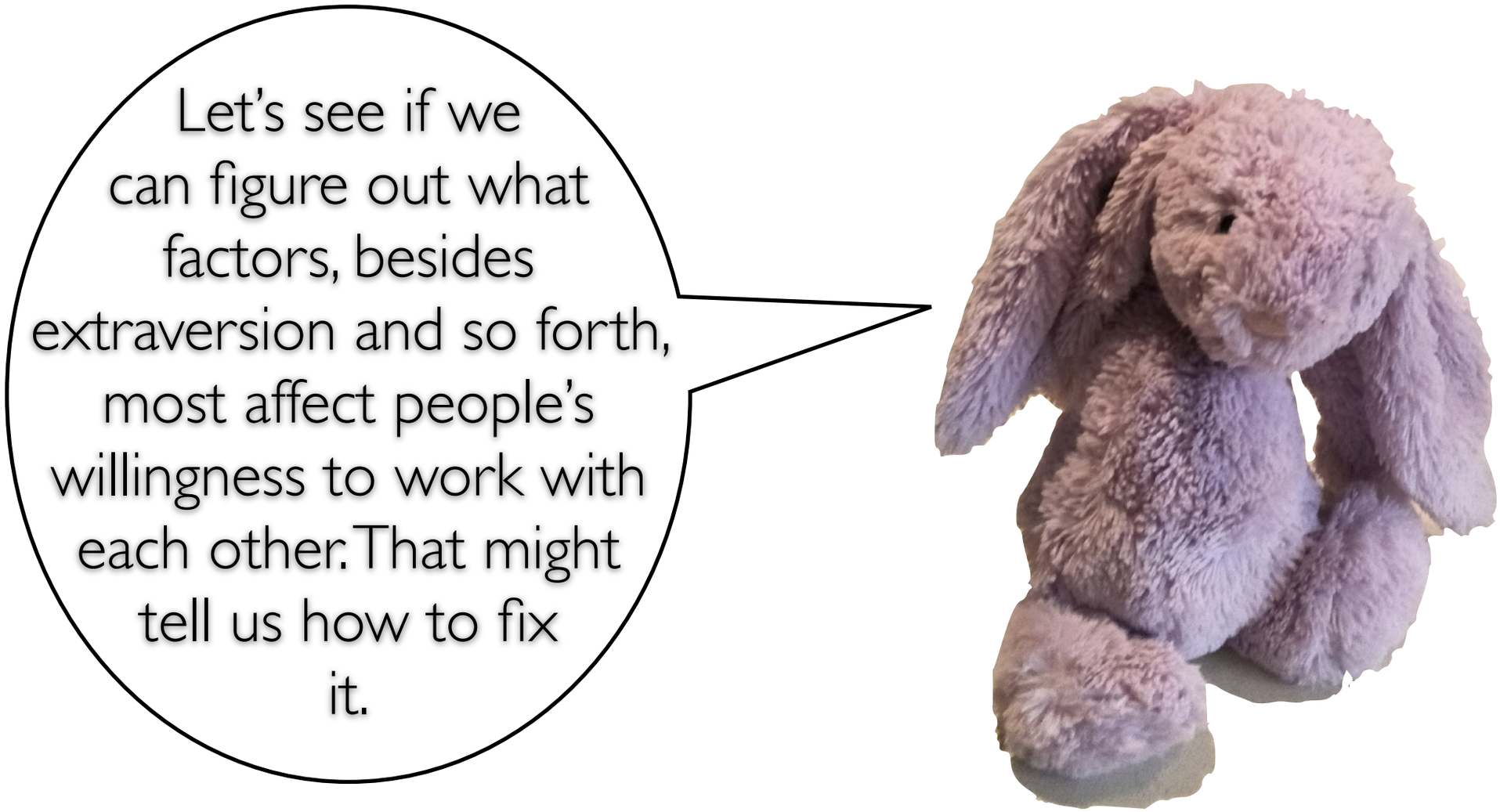
Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!

Yeah. Also, I didn't know anything about all of you, so that made me distrust as well. Now that I know you, it's different, but most people in our lands don't know each other.

Back to our story…

Our friends have figured out that they probably have to work together, between Bunnyland and Otherland, in order to solve the food crisis. But that doesn't mean they'll be able to persuade everyone else of this!

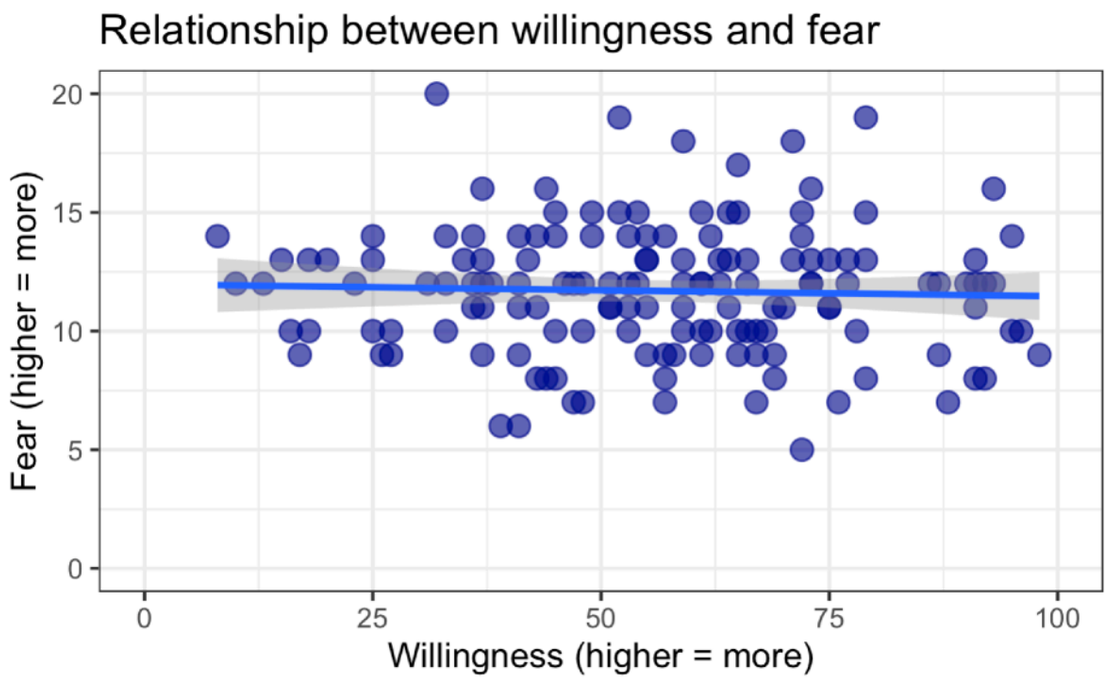


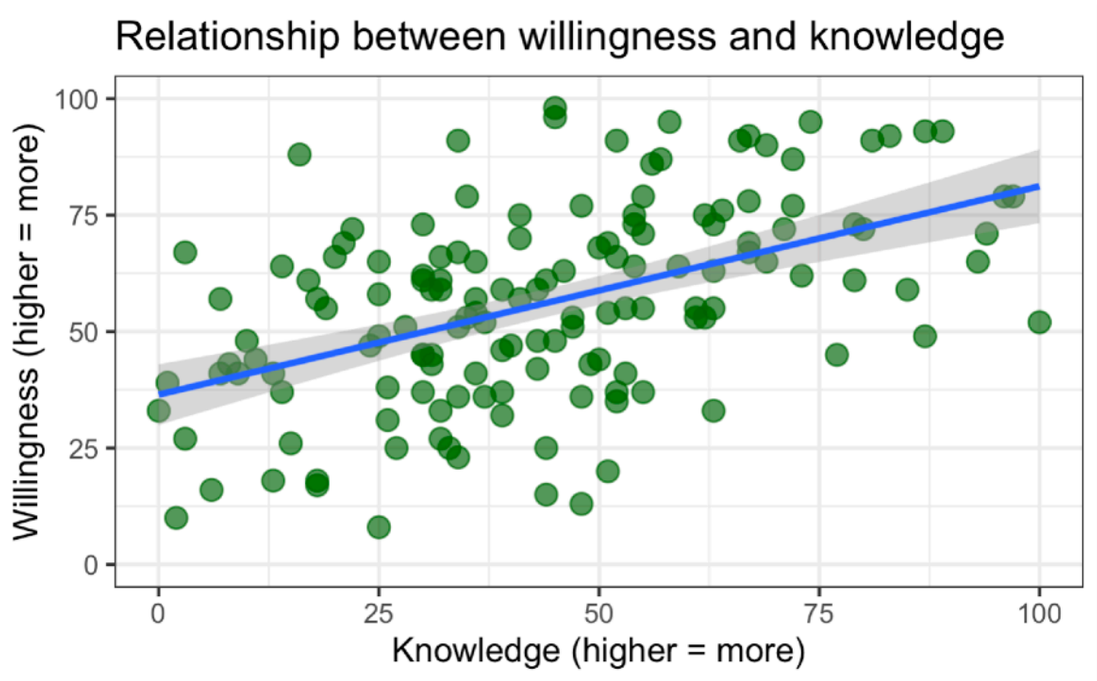
The data

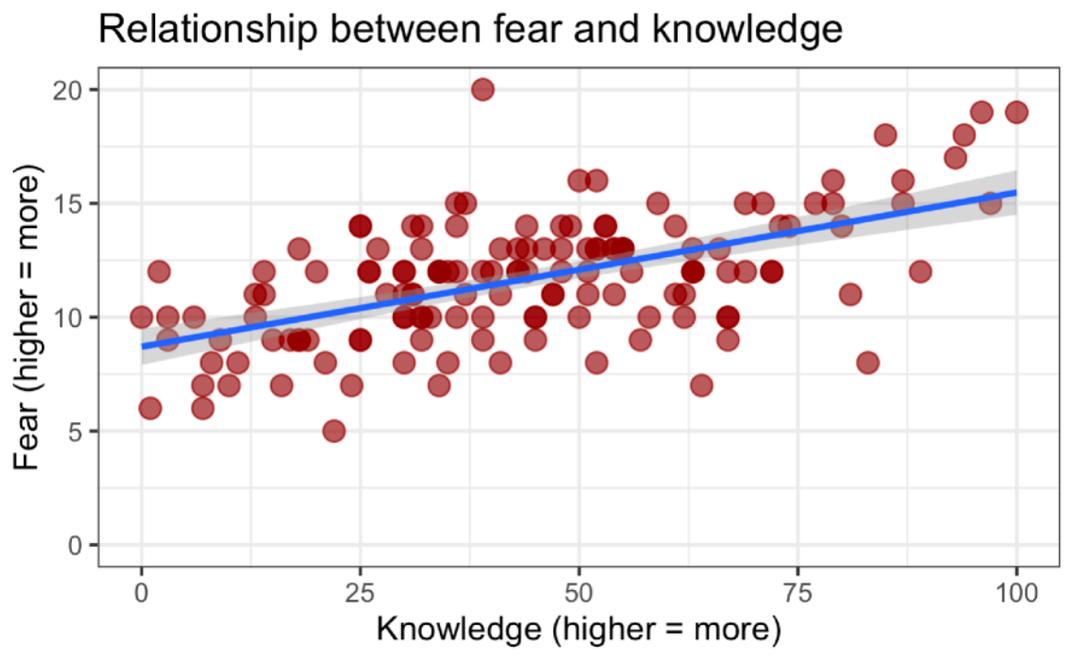
- One tibble, `db`, with seven variables
  - `person`... each person who took our surveys (82 from Bunnyland, 63 from Otherland)
  - `home`… whether from Bunnyland or Otherland
  - `willingness`… willingness to work with someone from the other place, on a scale of 0 (no way) to 100 (totally)
  - `knowledge`… score on a multiple choice quiz about the other place, on a scale of 0 (none correct) to 100 (all correct)
  - `fear`… self-reported fear of people from the other place, on a scale of 0 (no fear) to 20 (maximum fear)
  - `anger`… self-reported anger toward people from the other place, on a scale of 0 (no anger) to 20 (maximum anger)
  - `size`… size of the person (small, medium, or large)

What predicts willingness to work with each other?

Let's first just look at the roles of fear and knowledge







What predicts willingness to work with each other?

Let's look at a regression with knowledge and fear as predictors, but no interaction (just for simplicity)

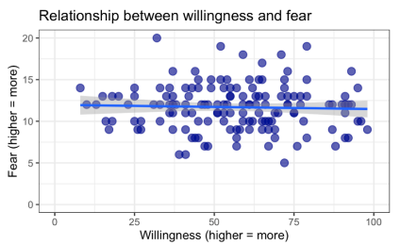
```
> modelWFK <- lm(willingness ~ fear + knowledge, data=db)
> summary(modelWFK)

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
```

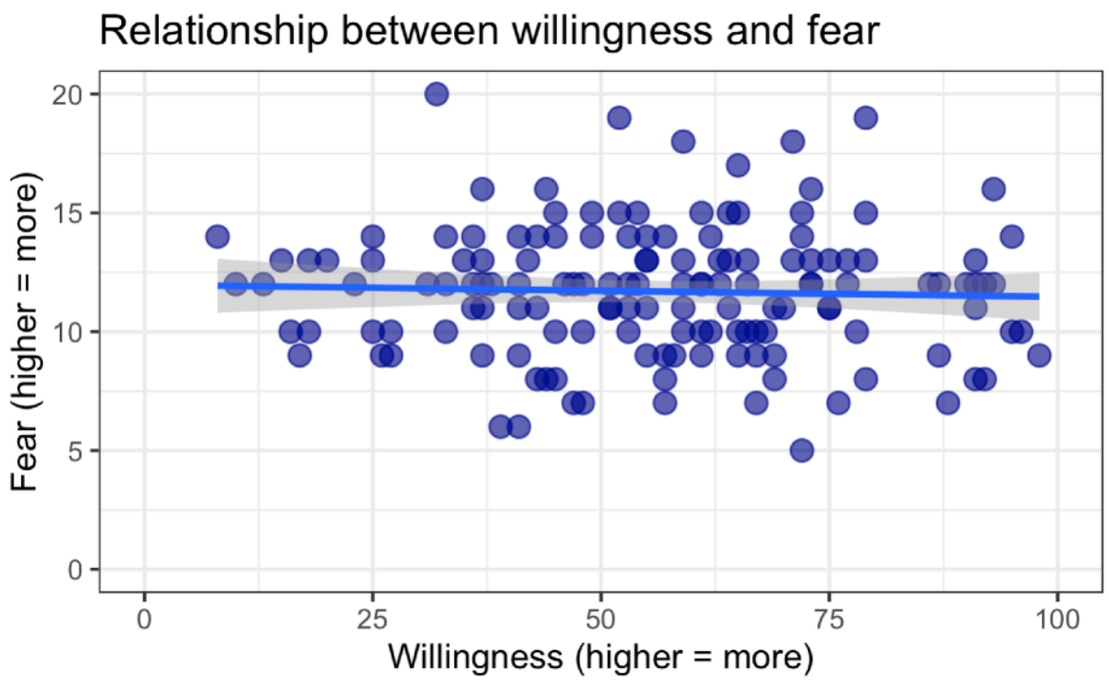
??

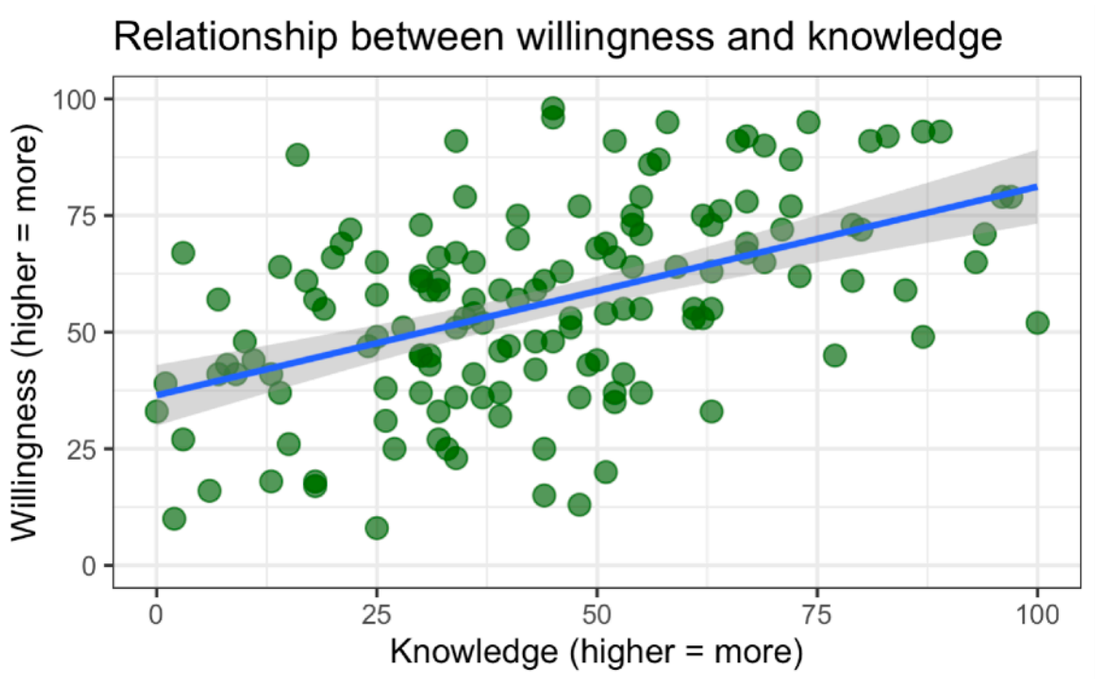


Hmm. Let's check some stuff

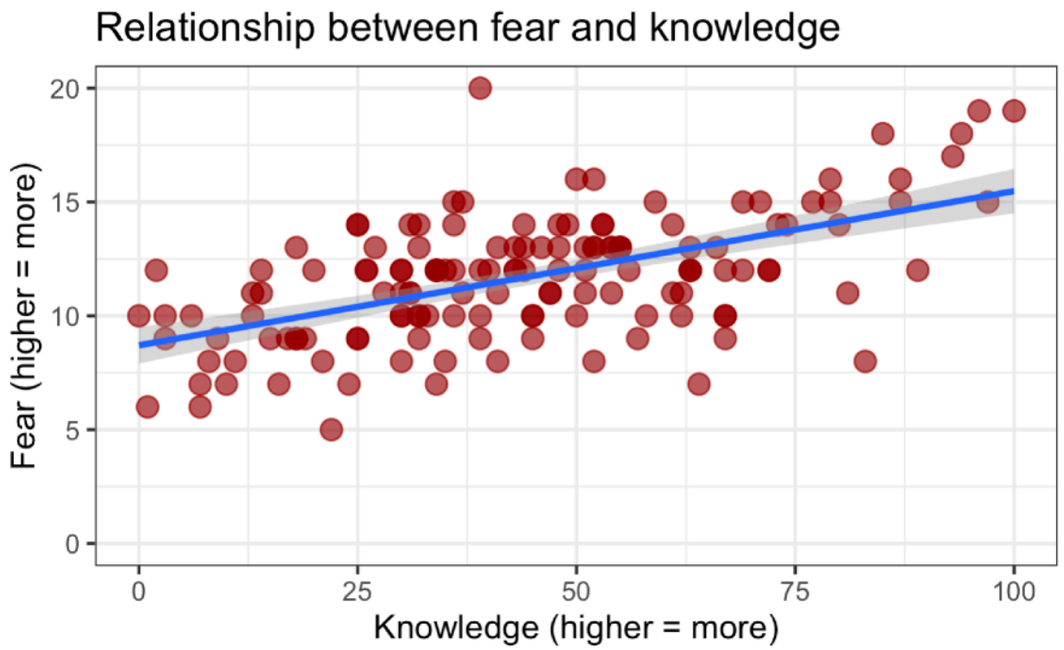
- Linearity
- Normality of residuals
- High-influence points
- Collinearity

Linearity



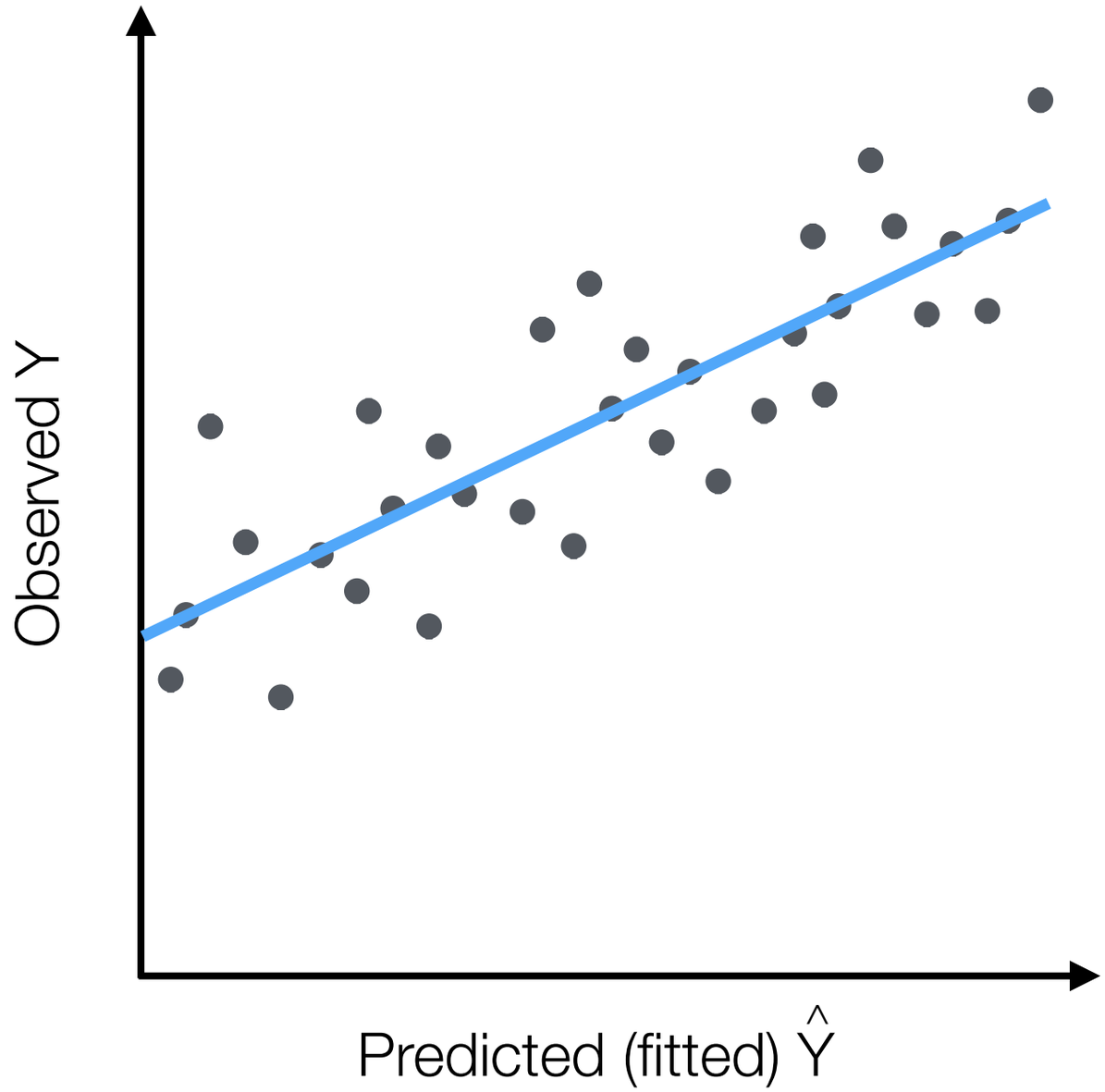


Looks linear, but we don't want each variable individually; we also want the fitted line. And maybe less vague too



Linearity

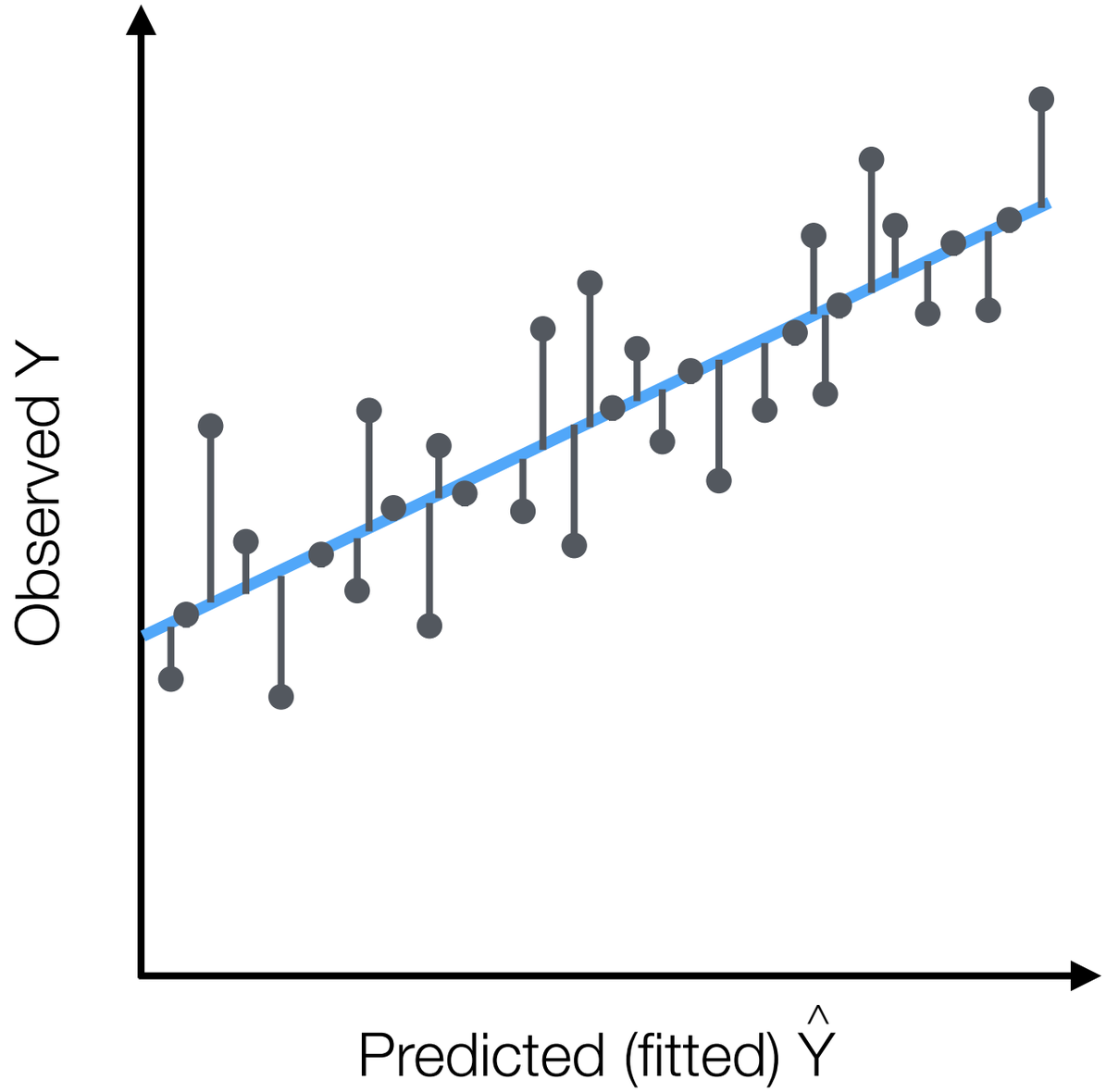
If it is linear, then there is a linear relationship between the predicted (fitted) values $\hat{Y}$ and the actual outcome data Y



Linearity

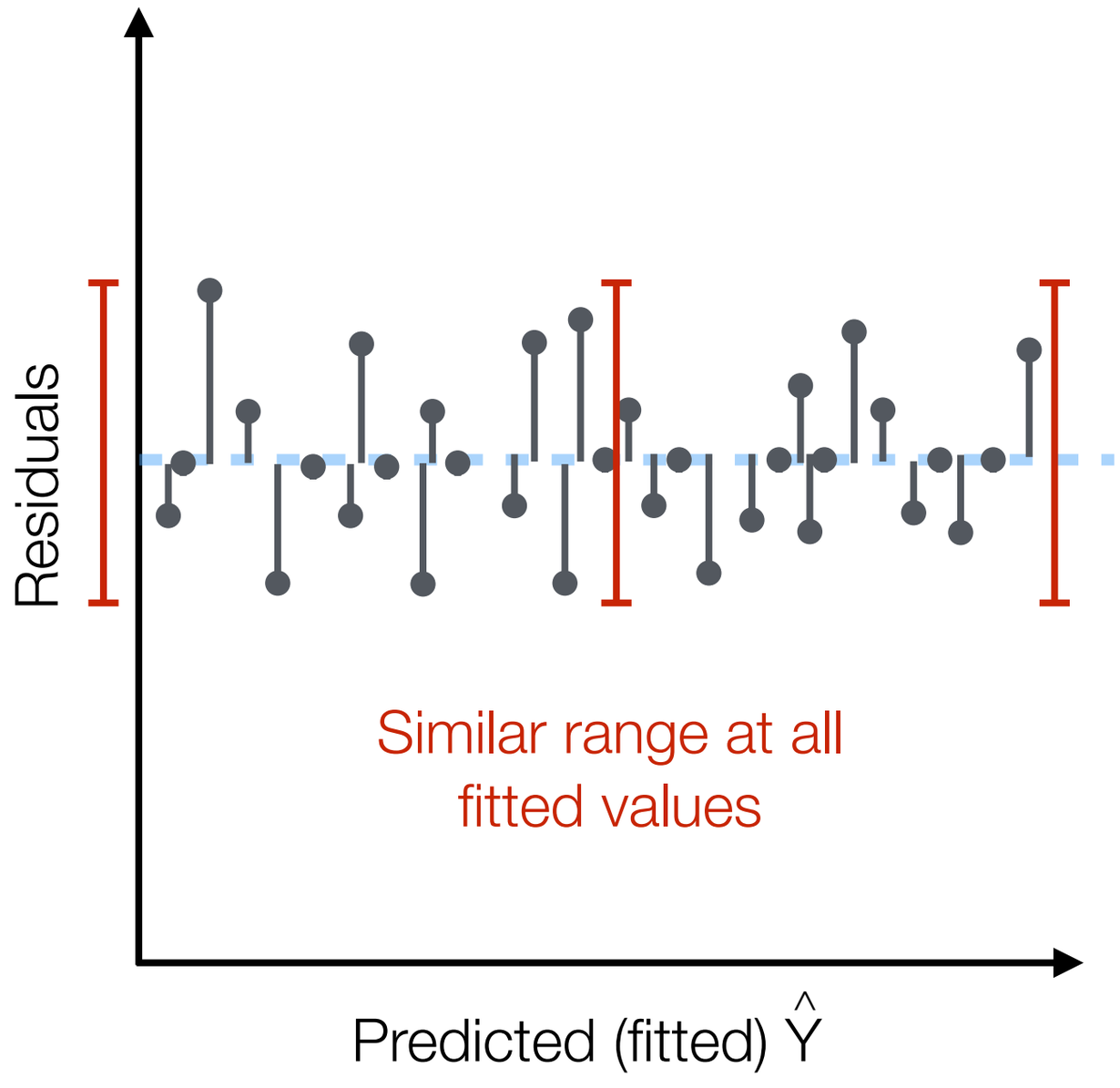
Look at the residuals. Intuition: if it is linear, they should be similar for all of the fitted predictor values.

$$\epsilon_i = Y_i - \hat{Y}_i$$



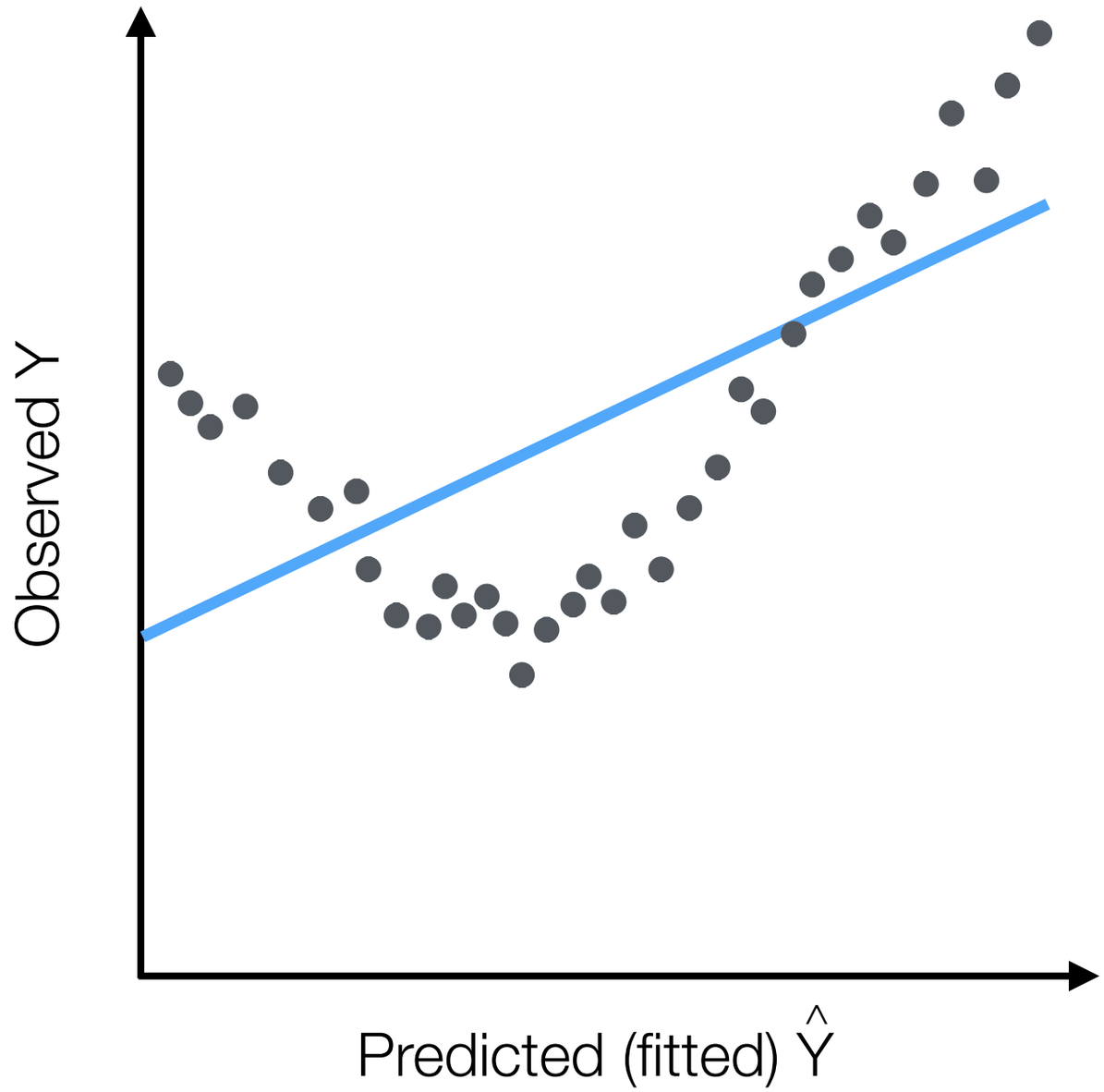
Linearity

Look at the residuals. Intuition: if it is linear, they should be similar for all of the fitted predictor values.



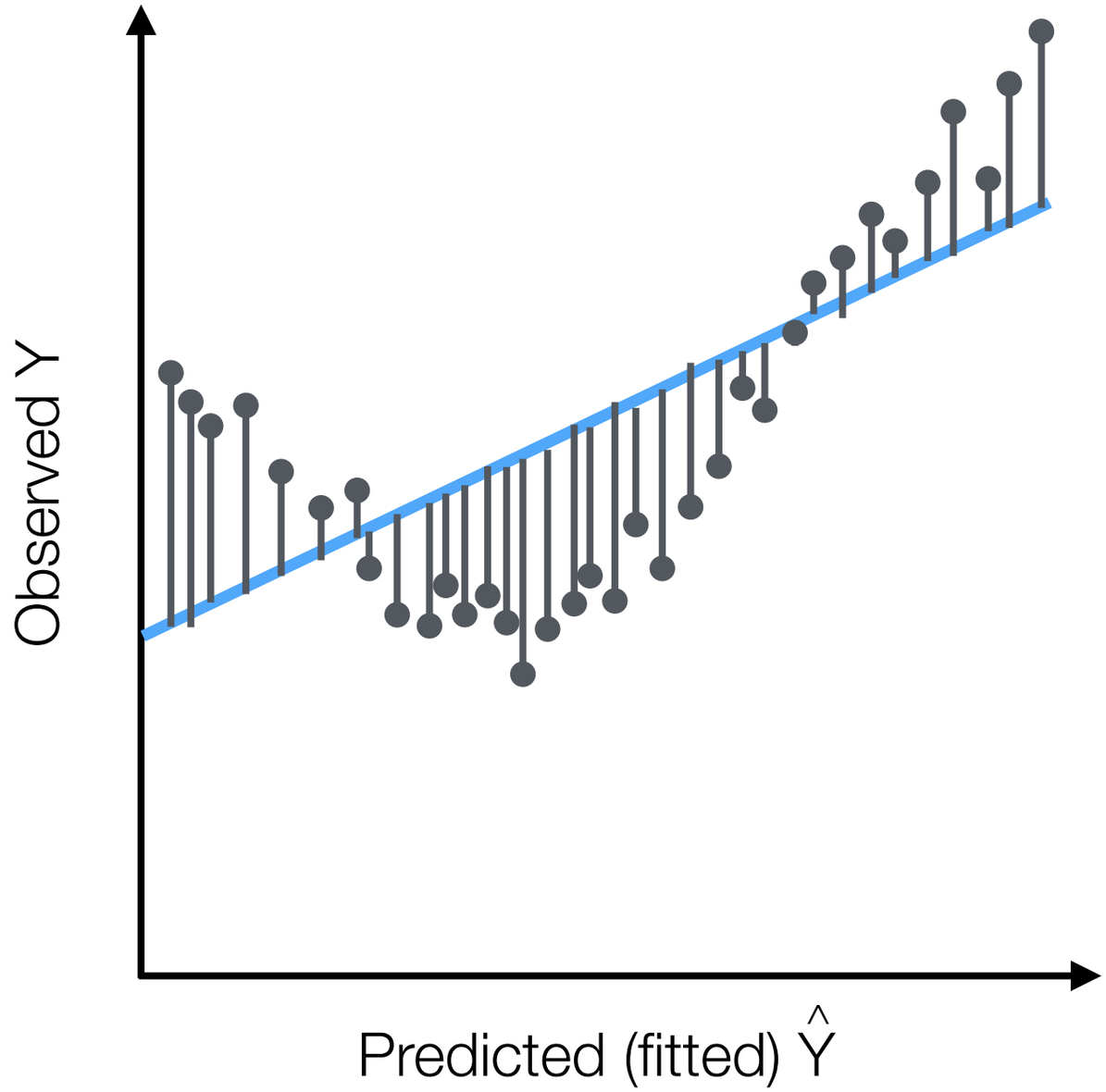
Linearity

How about if it's not linear?



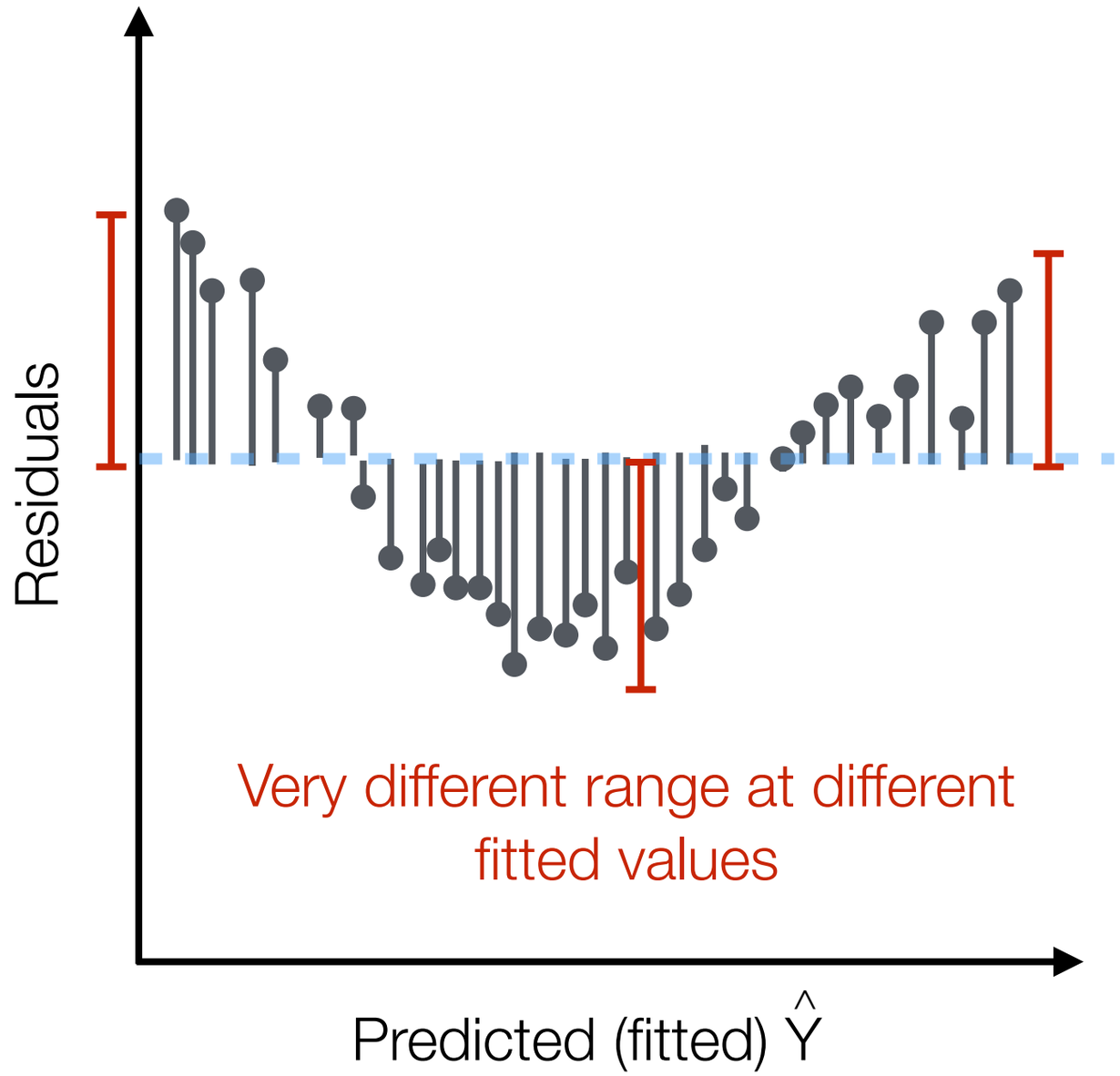
Linearity

Again, look at the residuals. Intuition: if it is not linear, they should not be similar throughout



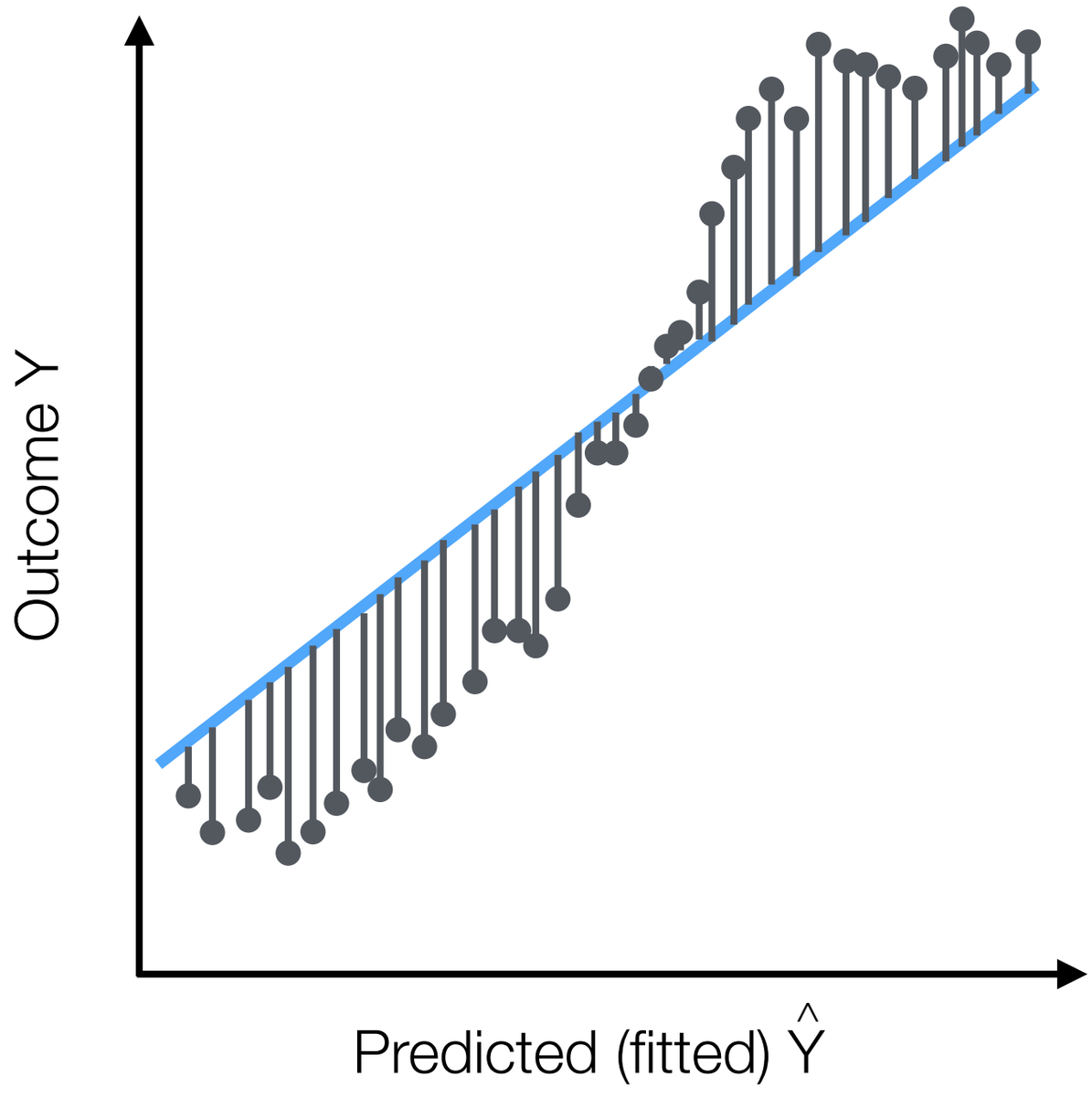
Linearity

Again, look at the residuals. Intuition: if it is not linear, they should not be similar throughout



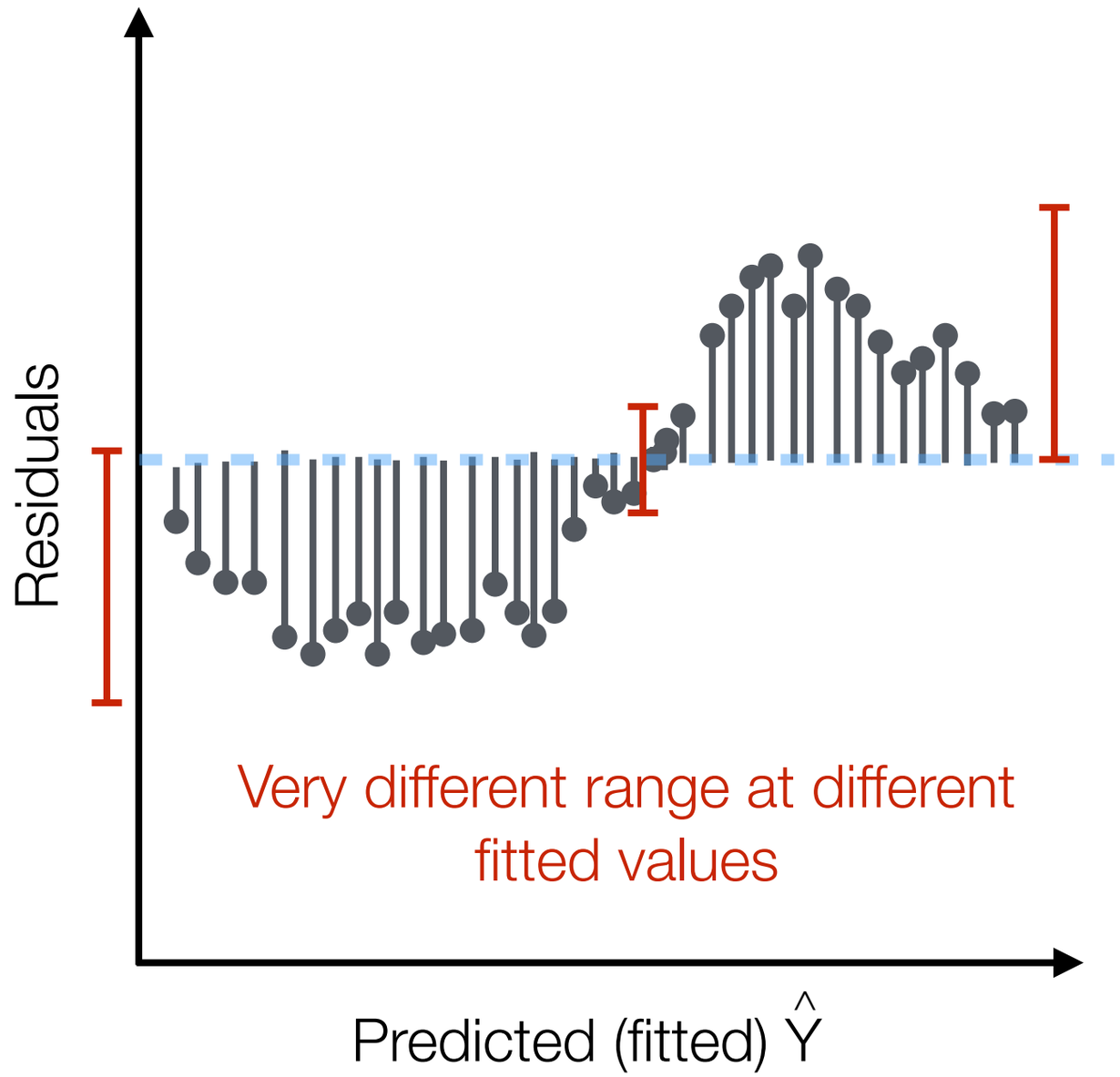
Linearity

This works for all different kinds of nonlinearity



Linearity

This works for all different kinds of nonlinearity



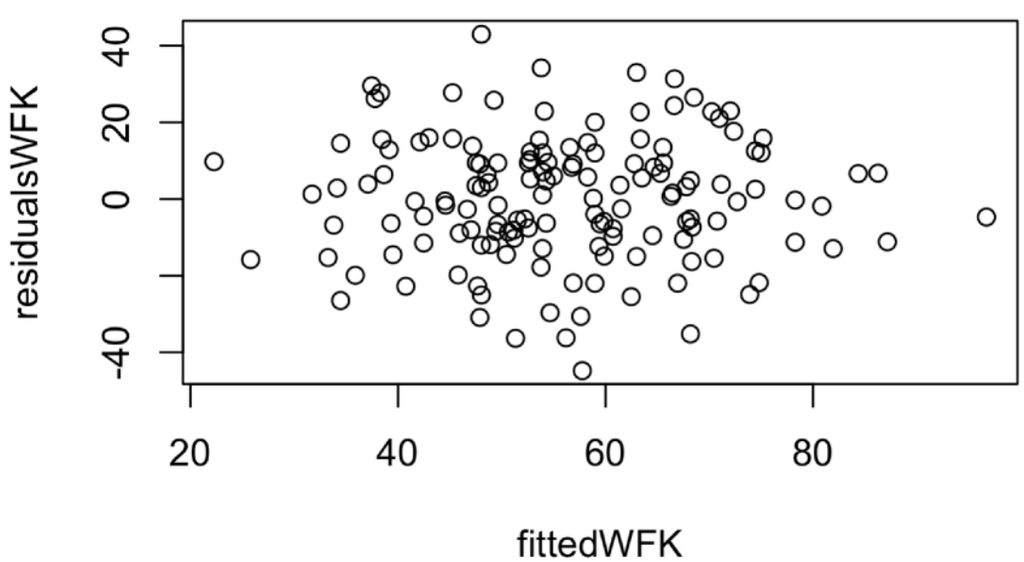
Analysing linearity in R

Could extract the fitted values and residuals from the model object using functions `residuals()` and `fitted.values()`

```
> residualsWFK <- residuals(modelWFK)
> fittedWFK <- fitted.values(modelWFK)
```

Then make a quick plot using the `plot()` command:

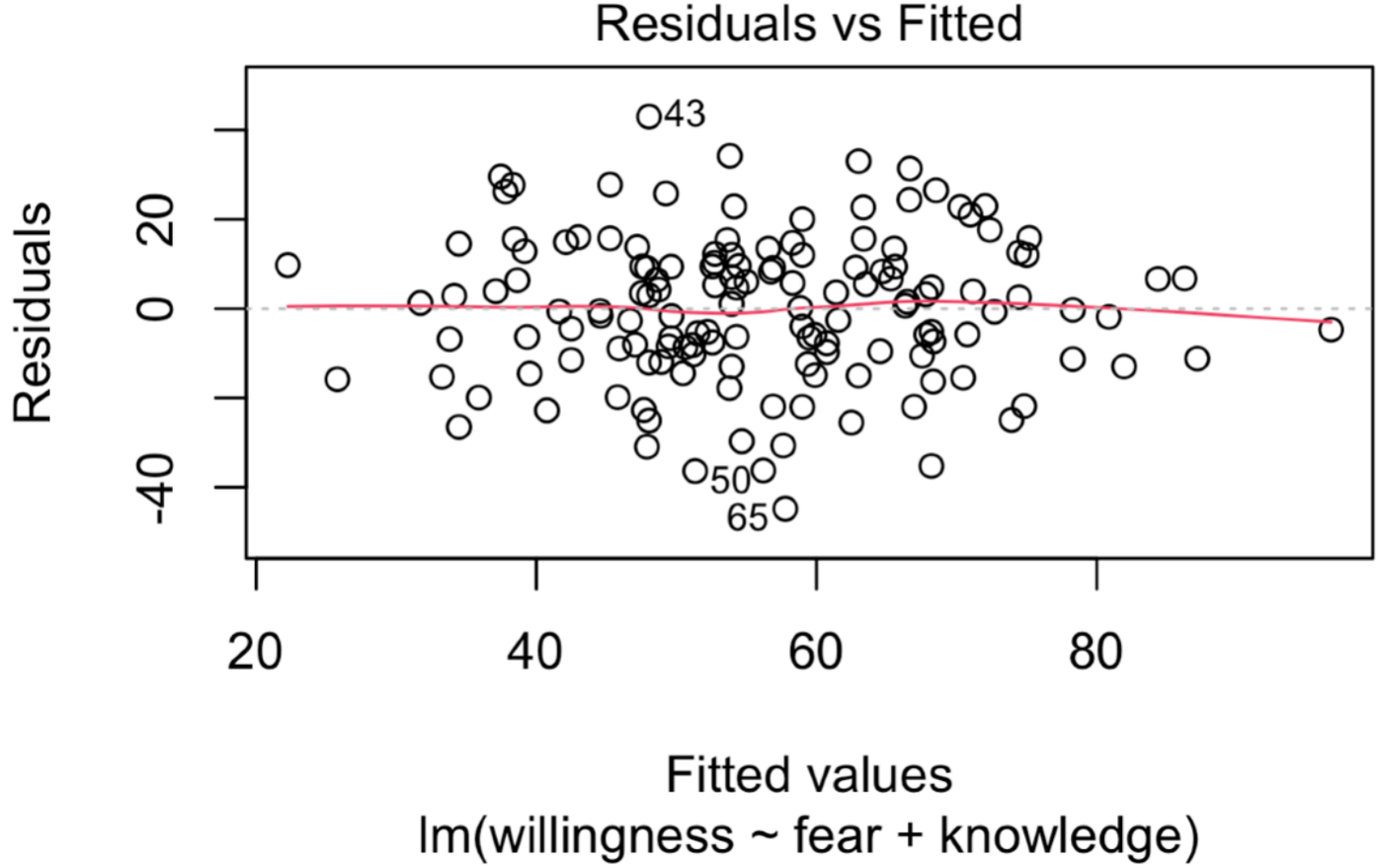
```
> plot(fittedWFK,residualsWFK)
```



Analysing linearity in R

A faster way to do it is to give the entire model object to the `plot()` command, along with the argument `which=1`

```
> plot(modelWFK,which=1)
```



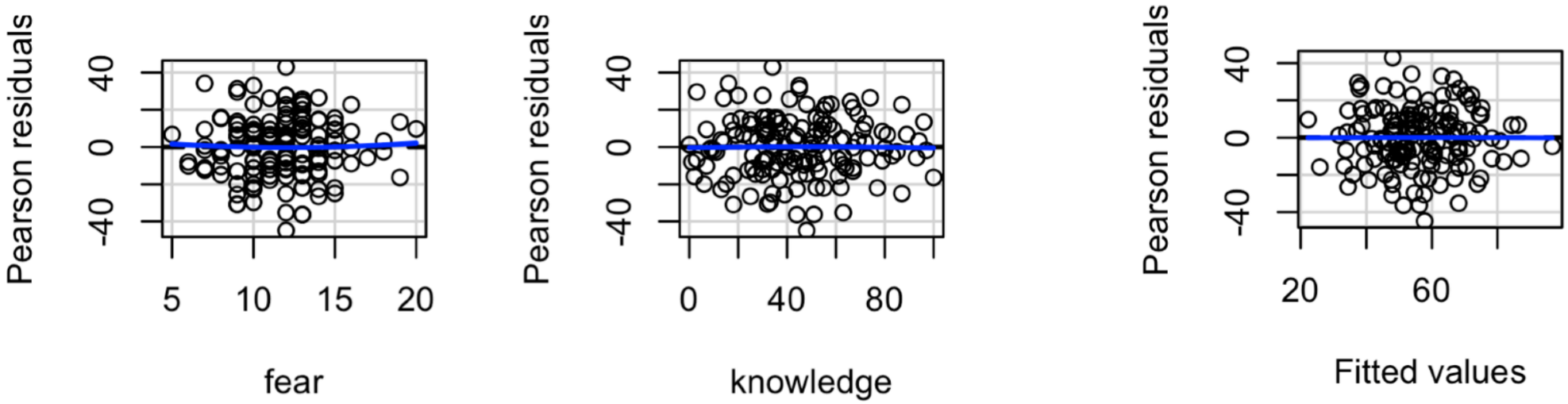
This looks pretty darn linear!

Analysing linearity in R

We can do the same thing for each predictor individually, using the `residualPlots()` command from the `car` library

```
> library(car)
> residualPlots(modelWFK)
               Test stat Pr(>|Test stat|)
fear             0.3111           0.7562
knowledge       -0.1076           0.9144
Tukey test      -0.0181           0.9855
```

If p<.05, that variable is probably not linear



Seems safe to assume linearity!

What do we do if it's not linear?

Beyond the scope of this class, but usually the solution is to do a different kind of regression (e.g. logistic) or to transform your variable (see Chapter 15). Relatively small deviations from linearity are not usually a huge problem though.

Normality of the residuals

Just like the ANOVA, regressions assume that the residuals are normal. There are actually two different kinds of residuals we might care about:

Ordinary residuals. This is what we've been talking about so far

$$\epsilon_i = Y_i - \hat{Y}_i$$

Standardised residuals. If your predictors are on different scales, these are nice because they are essentially what you'd get if you converted the ordinary residuals to z scores

$$\epsilon'_i = \frac{\epsilon_i}{\hat{\sigma}\sqrt{1 - h_i}}$$

This is called a hat-value and I'll explain it later

Normality of the residuals

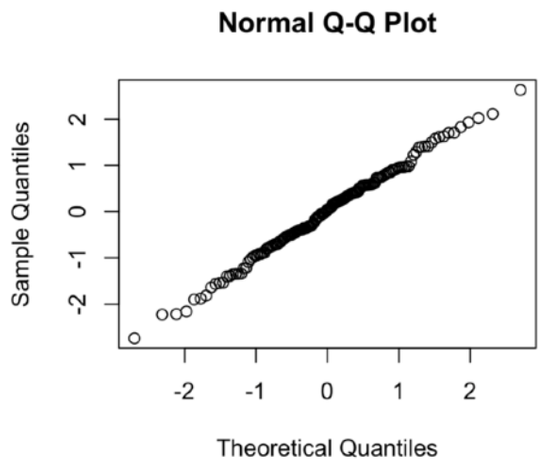
We can test normality the same way we have done before: QQ plots and the Shapiro-Wilk test (on the standardised residuals)

First get the standardised residuals using `rstandard()`

```
> rstandardWFK <- rstandard(modelWFK)
```

QQ plot

```
> qqnorm(rstandardWFK)
```



Shapiro-Wilk

```
> shapiro.test(rstandardWFK)

   Shapiro-Wilk normality test

data:  rstandardWFK
W = 0.9963, p-value = 0.9758
```

Seems safe to assume normality!

What do we do if they aren't normal?

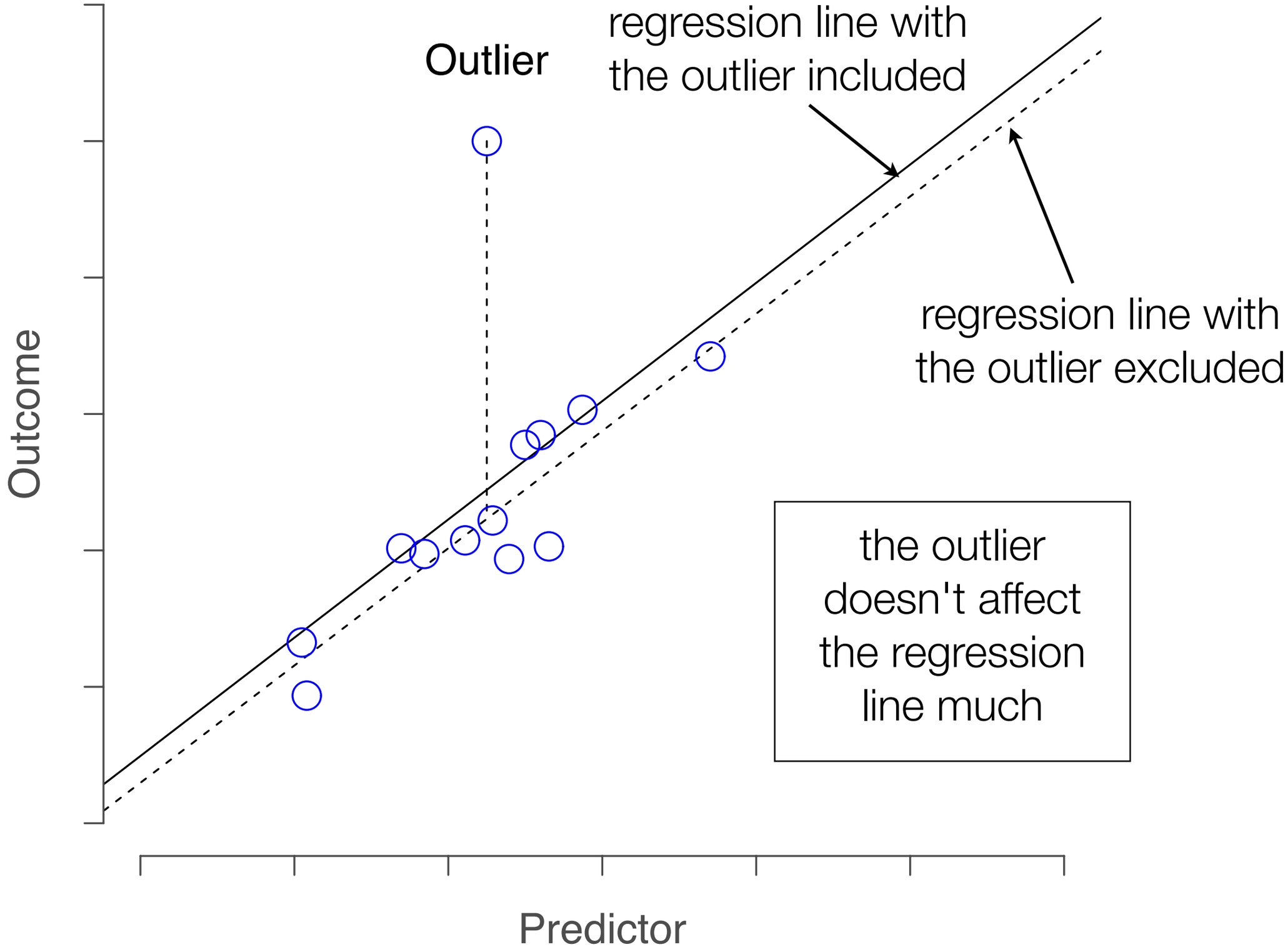
If relatively minor deviations, just interpret the results with a grain of salt and be a little less certain of your conclusions

If larger, this also is beyond the scope of this class, but usually the solution is to do a different kind of regression (e.g. logistic) or to transform your variable (see Chapter 15).

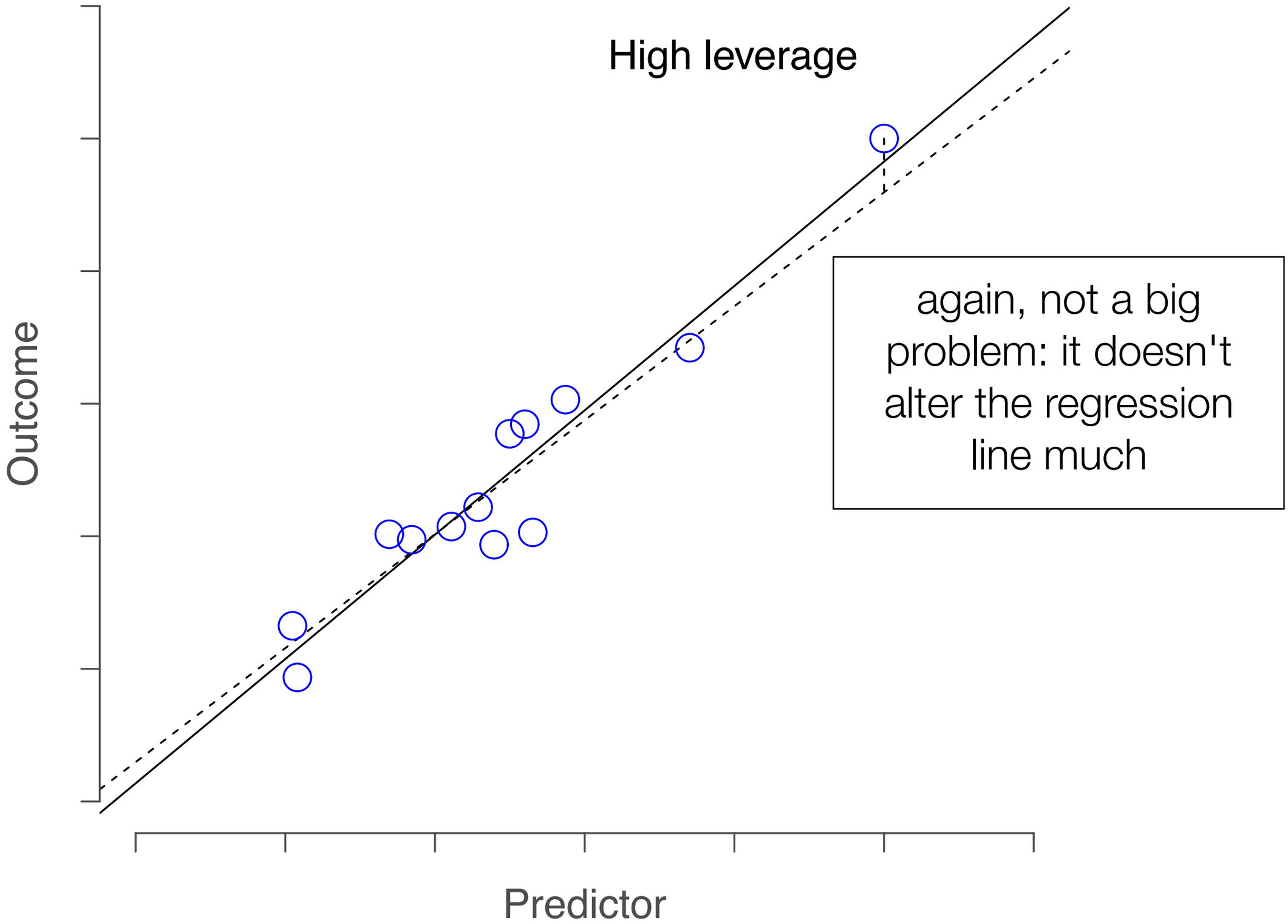
Identifying high-influence points

- Outlier:
  - an observation that has a large residual
  - i.e., model prediction is a long way off
- High leverage point:
  - an observation that has different values on the predictors than the other ones
  - residual might still be small though
- High influence point:
  - an outlier with high leverage
  - these are dangerous!

An outlier alone is usually okay

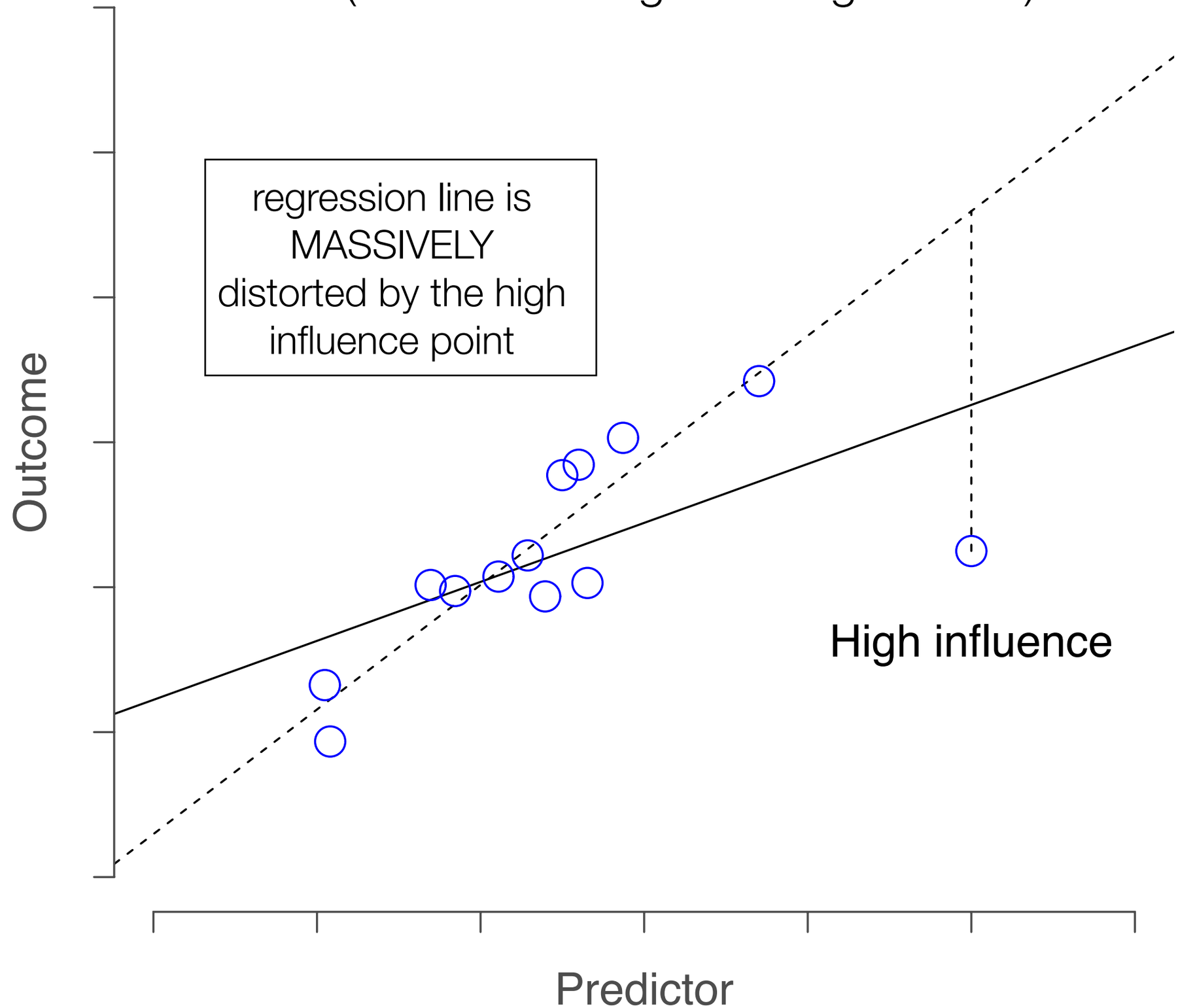


A high leverage point alone is usually okay



A high influence point is dangerous

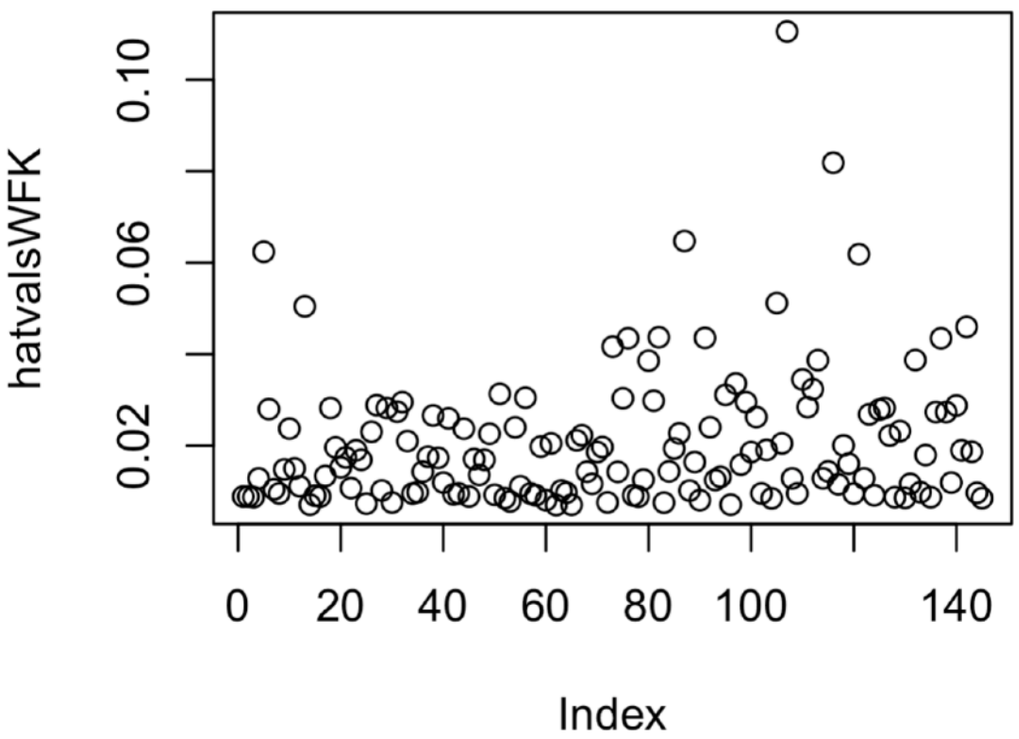
(influence = high leverage outlier)



Identifying high-influence points

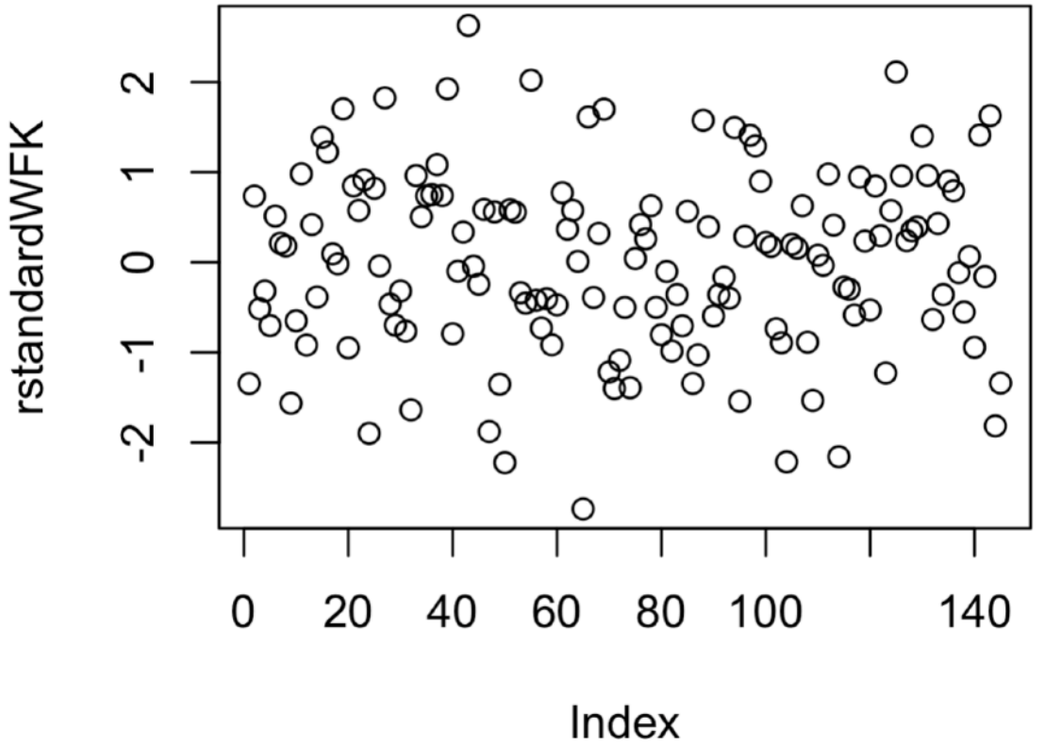
Leverage can be quantified using what is called a hat value h_i which measures the extent to which the i-th observation "controls" the regression line

```
> hatvalsWFK <- hatvalues(modelWFK)
> plot(hatvalsWFK)
```



Outliers are points with large standardised residuals

```
> rstandardWFK <- rstandard(modelWFK)
> plot(rstandardWFK)
```



Identifying high-influence points

High-influence points have a high leverage *and* high influence.
They are quantified with something called **Cook's Distance**

Measures outlier-ness

$$D_i = \frac{\epsilon_i^{*2}}{K+1} \times \frac{h_i}{1-h_i}$$

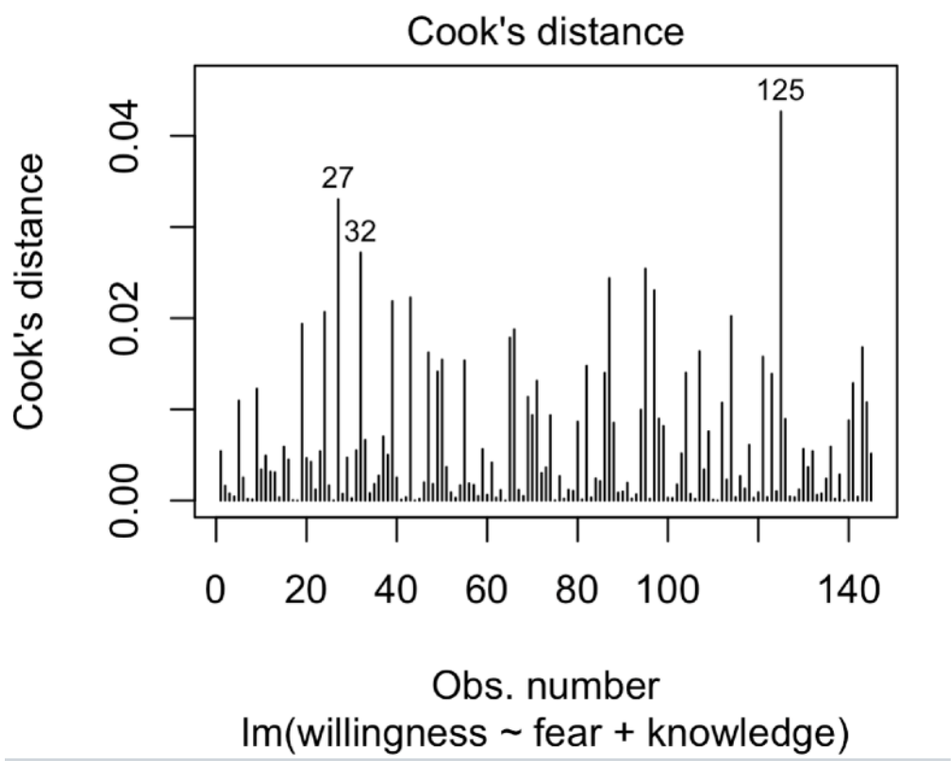
Measures leverage

Give the entire model object to the `cooks.distance()` command

```
> cookdWFK <- cooks.distance(modelWFK)
> plot(cookdWFK)
```

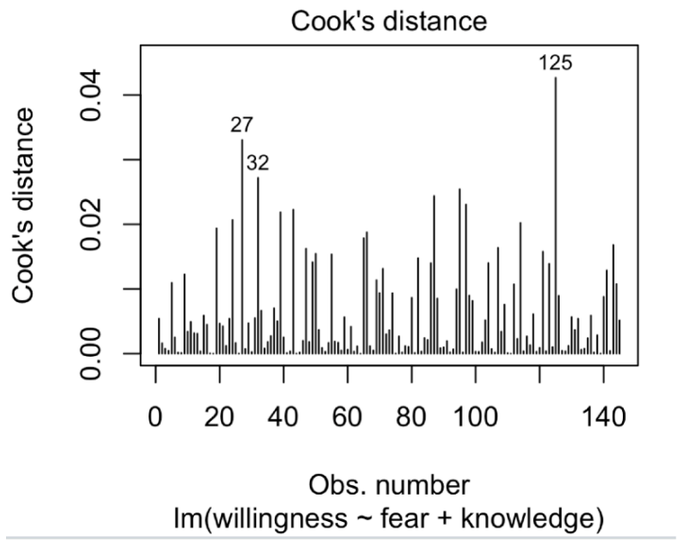
or

```
> plot(modelWFK,which=4)
```



Which ones are high leverage?

There is a lot of disagreement about how to determine this!



Here, k=3 (b_0, b_1, b_2) and N=145

- Some say it shouldn't be $>2k/N$, where k=# of coefficients

```
> 2*3/145
[1] 0.04137931
```

→ One datapoint, pt 125

- Others say it shouldn't be $> 4/(N-k-1)$

```
> 4/(145-3-1)
[1] 0.02836879
```

→ Two datapoints, pt 125 and 27

- Others say it shouldn't be > 3 times the mean Cook's Distance

```
> 3*mean(cookWFK)
[1] 0.01775223
```

→ 13 datapoints, too many to list

- Others say it shouldn't be > 0.1 or 1

→ No datapoints

In real life: pre-register. In my experience most of these are too conservative, and I often go with 0.1 (and always do analyses with and without exclusion). In this class: use $2k/N$.

What do you do if you have high influence points?

Difficult. You don’t want to just remove data willy-nilly. Would be good to pre-register your exclusion criteria before analysing data

Sometimes might remove those datapoints and make sure the qualitative results are similar. Or keep them and interpret results knowing they are there

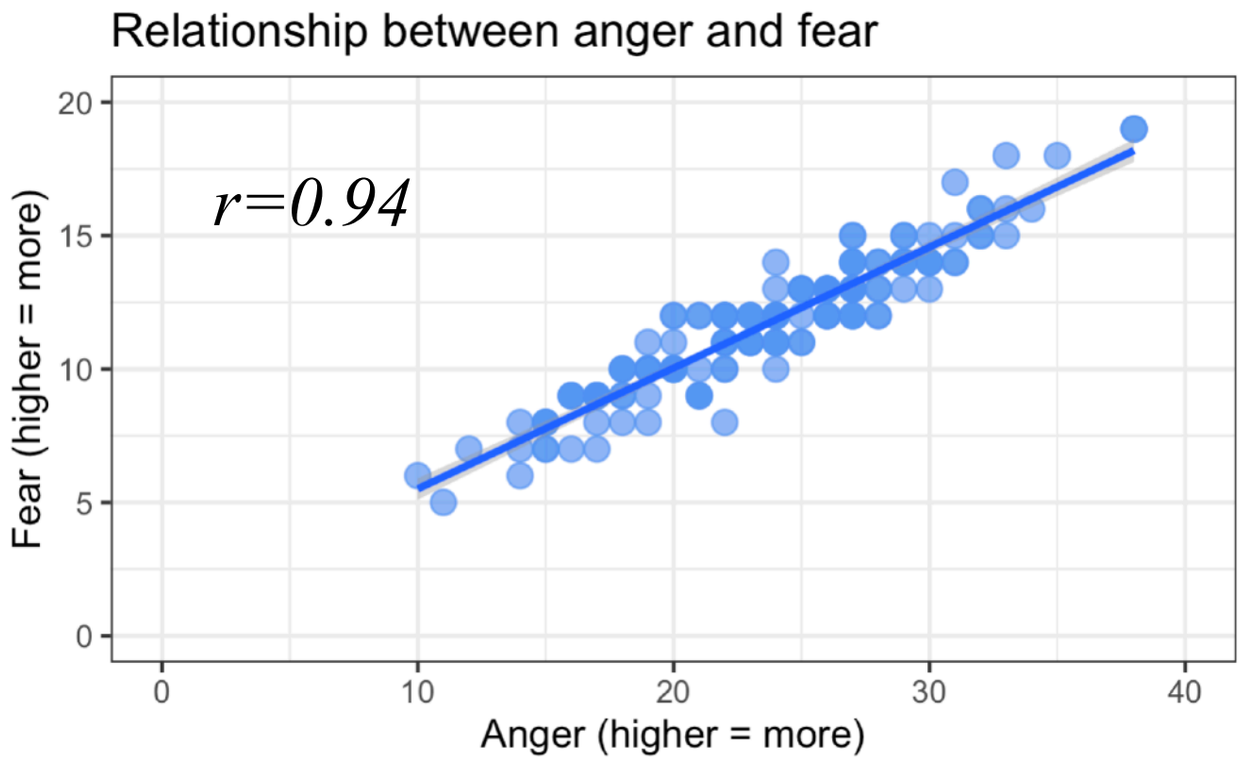
Collinearity

This is when your predictors are highly correlated with each other. When this happens, it's tough sensibly interpret the results, and the correlation causes a lot of uncertainty about the coefficients

```
> cor.test(db$fear,db$anger)

   Pearson's product-moment correlation

data:  db$fear and db$anger
t = 33.959, df = 143, p-value < 2.2e-16
95 percent confidence interval:
 0.9219757 0.9588139
sample estimates:
      cor
0.9432268
```



How large of a correlation is "too large"?

Collinearity

The biggest problem is that when you have highly correlated variables, the regression has a lot of uncertainty about the coefficients of each of them. So we quantify with a Variance Inflation Factor (VIF) which captures how badly the correlation is messing up your confidence intervals around the coefficients

$$\mathrm{VIF}_k = \frac{1}{1 - R^2_{(-k)}}$$

Variance accounted for by the model where variable *k* is the outcome and the others are predictors

VIF is high if adding that variable is highly predicted by the others

The square root of VIF is about how much larger the confidence interval around b_k is than you'd expect if everything was uncorrelated

VIF = 1 is great!
Much larger than 2 or 3.. possible problem?

Collinearity

Calculate it in R using the `vif()` function in the `car` package

```
> vif(modelWFK)
     fear knowledge
 1.486072  1.486072
```

By contrast, consider a model with fear and anger in it…

```
> modelWFKA <- lm(willingness ~ fear + knowledge + anger, data=db)
> vif(modelWFKA)
     fear knowledge     anger
 9.660628  1.487103  9.070571
```

Note that it doesn't make sense to use VIF on a model with interaction terms! Your question is about the collinearity of *each variable alone*

What do you do if you have collinearity?

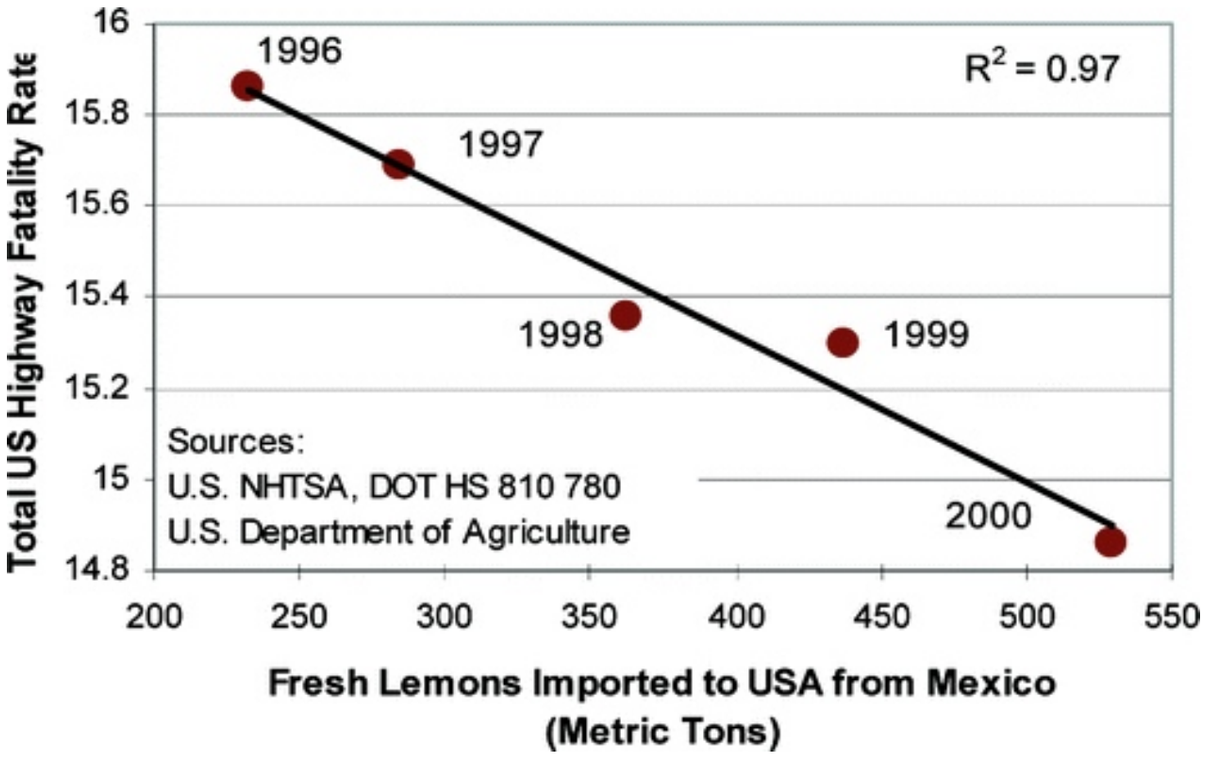
Generally, you want to take out some of the collinear variables (probably also report that they were collinear, because that is relevant and also may be theoretically interesting).

Which ones? Guide by your theory.

Finally: remember correlation ≠ causation

- You know this, but it's often easy to forget when doing a regression, because you usually use them in a non-experimental design.
- The variables are not controlled and no intervention has been done, so causation cannot be inferred

Lemons cause road deaths!



Don't be this person!

Back to our question…

What's going on here? Nothing seemed to be wrong when we checked assumptions… we'll talk about that in the next video

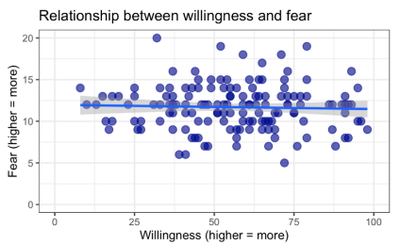
```
> modelWFK <- lm(willingness ~ fear + knowledge, data=db)
> summary(modelWFK)

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
```

??



Exercises are in `w10daylexercises.Rmd`