Model validation

Models behaving badly

Samuel Robinson, Ph.D.

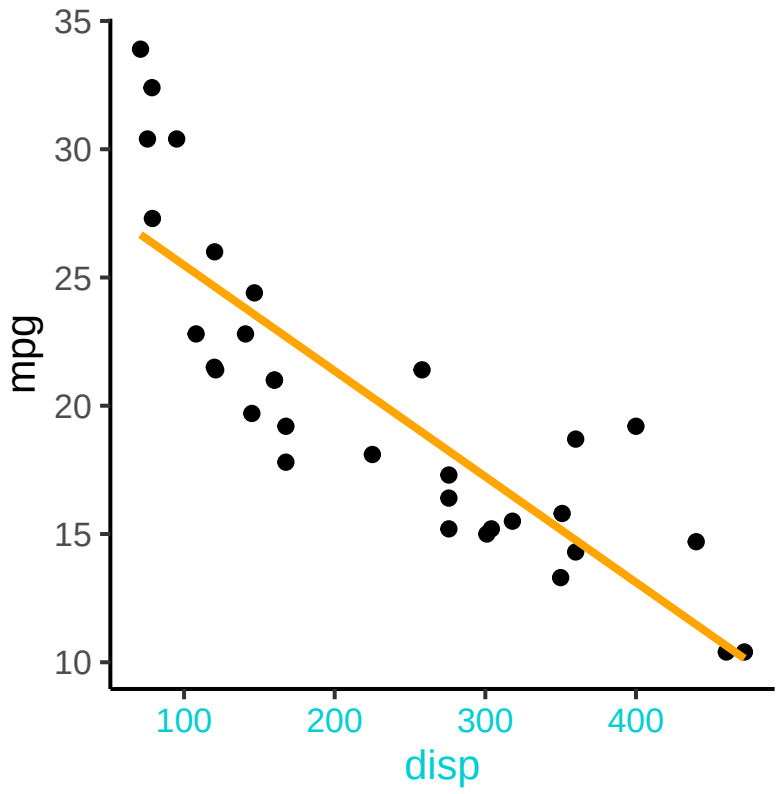
October 23, 2020

Motivation

Are my model results reliable?

- Residual checks
- Transformations
- Collinearity
- How much stuff should I put into my model?

Assumptions of linear regression



There are 3 main assumptions to this model:

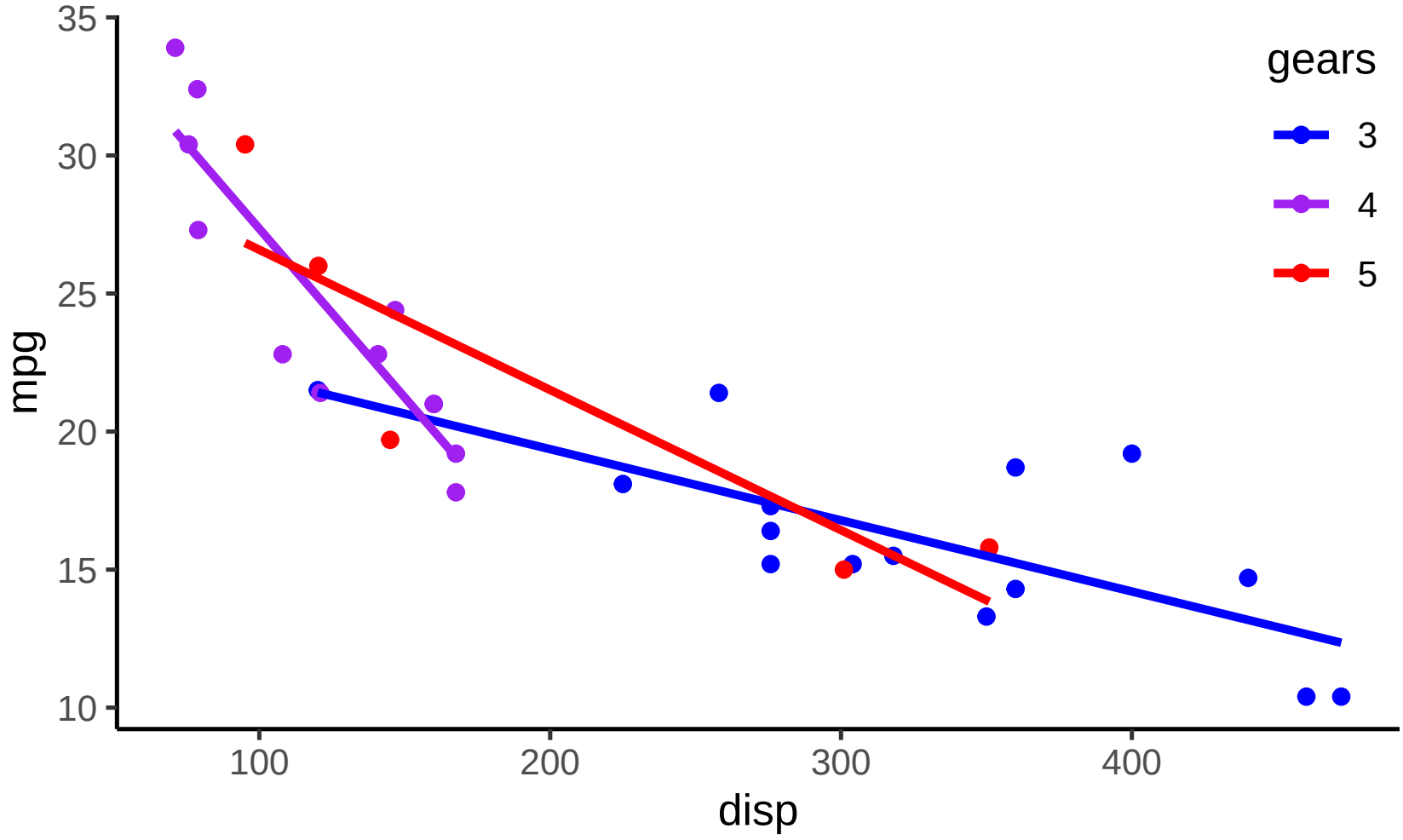
1. The relationship between *disp* and *mpg* is linear
2. *mpg* (the data) is Normally distributed around $\hat{mpg}$ (the line)
3. σ is the same everywhere

This is pretty easy to see if you only have 1 variable, but...

$$\hat{mpg} = b_0 + b_1 disp$$

$$mpg \sim Normal(\hat{mpg}, \sigma)$$

What if I have many variables?

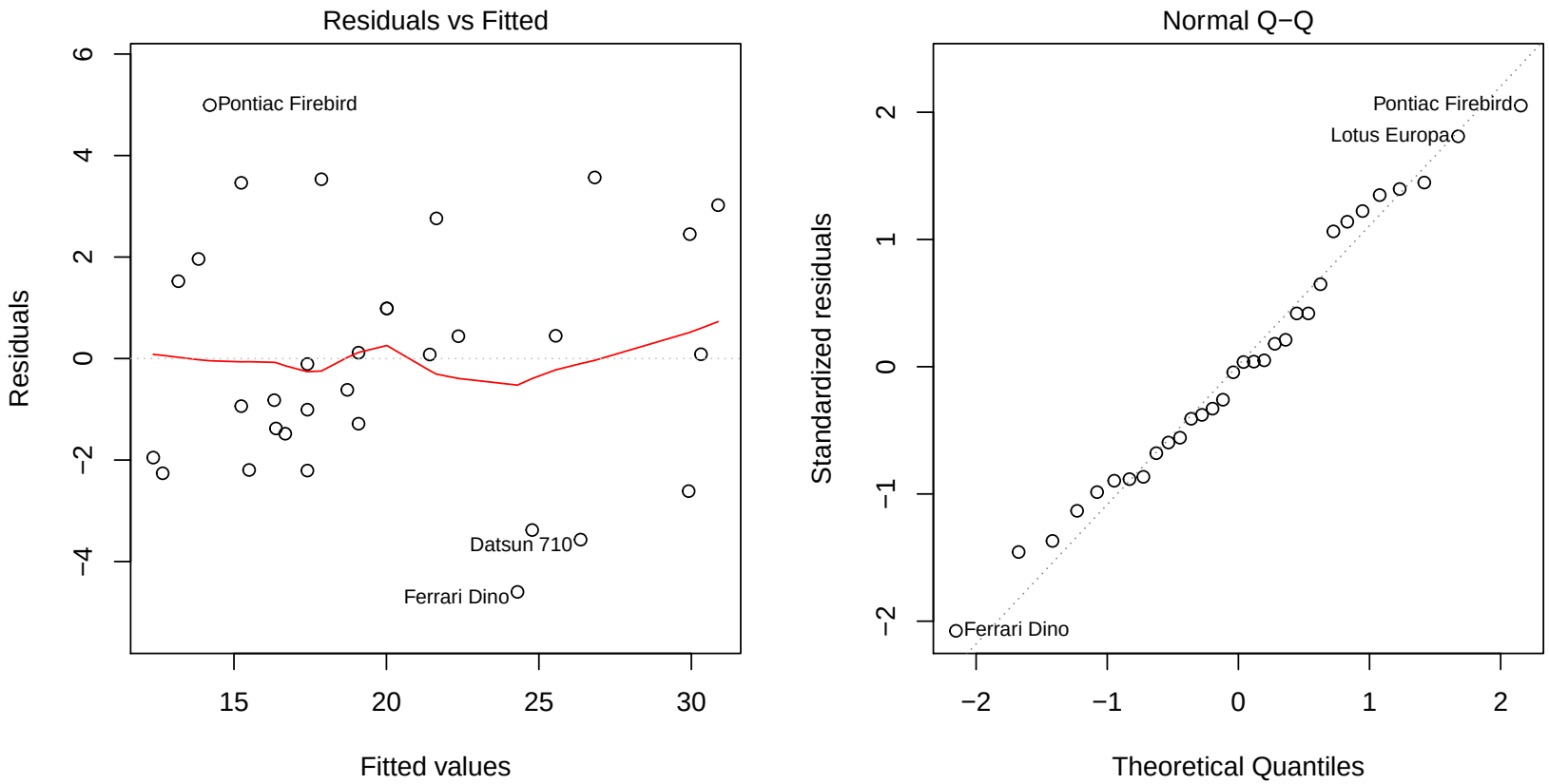


Difficult to see if the assumptions are met

Solution: residual checks

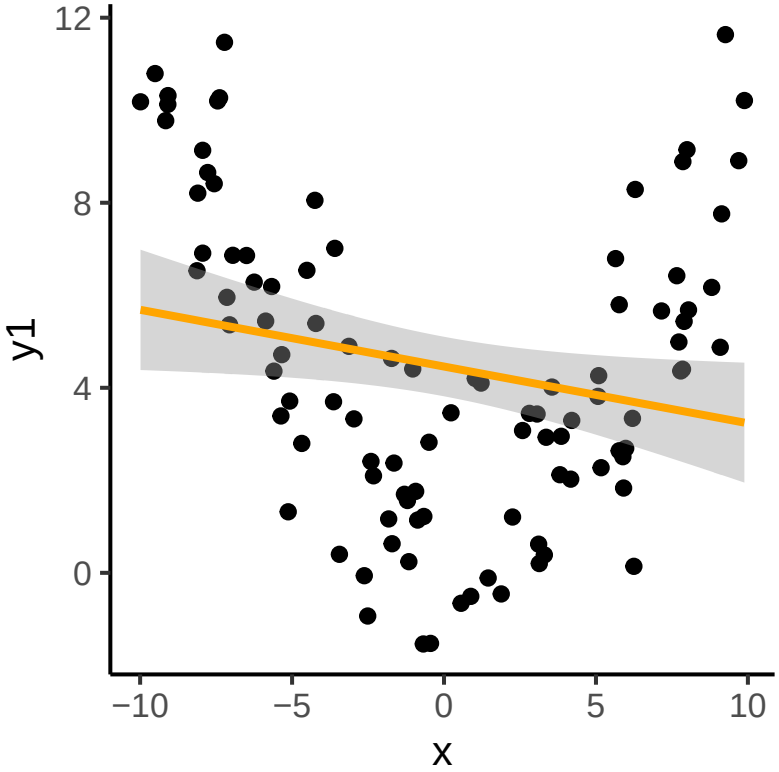
Some common ways of checking the assumptions: **residual plots**

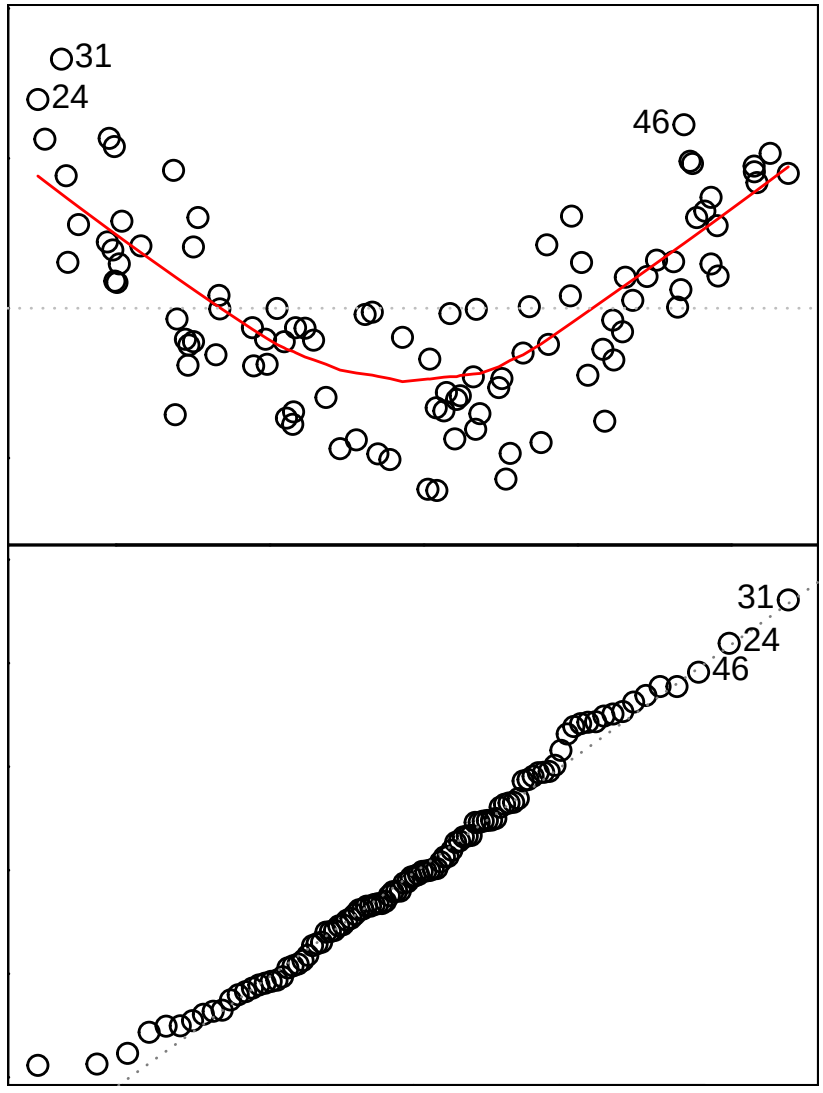
```
mod1 <- lm(mpg~disp*factor(gear),data=mtcars) #Fits model
par(mfrow=c(1,2),mar=c(3,3,1,1)+1) #Splits plot into 2
plot(mod1, which=c(1,2)) #1st and 2nd residual plots
```



1. Points in Plot 1 should show *no pattern* (shotgun blast)
2. Points in Plot 2 should be *roughly* on top of the 1:1 line

Problem 1: Non-linear relationship

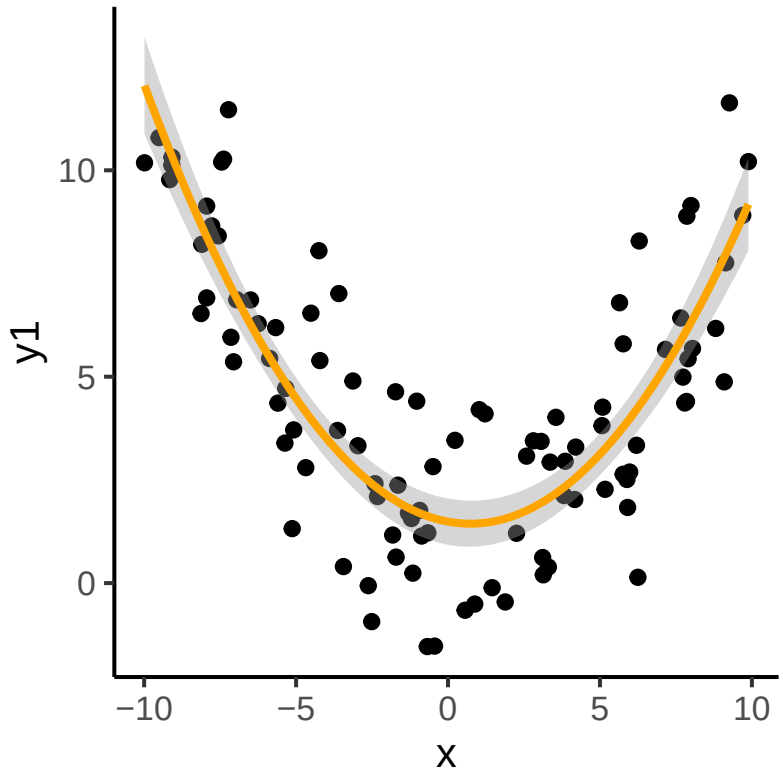


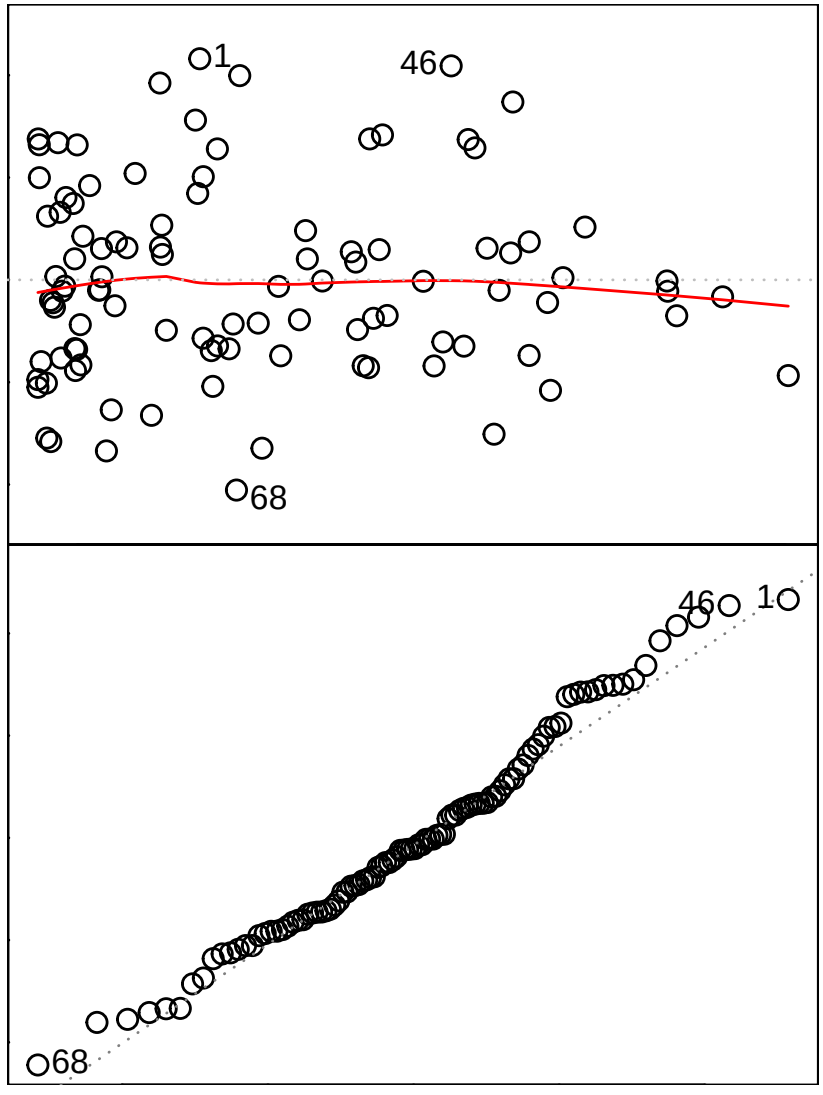



```
lm(y1~x,data=d1)
```

$y1$ clearly follows a hump-shaped relationship, not a linear one

Solution: transform predictors



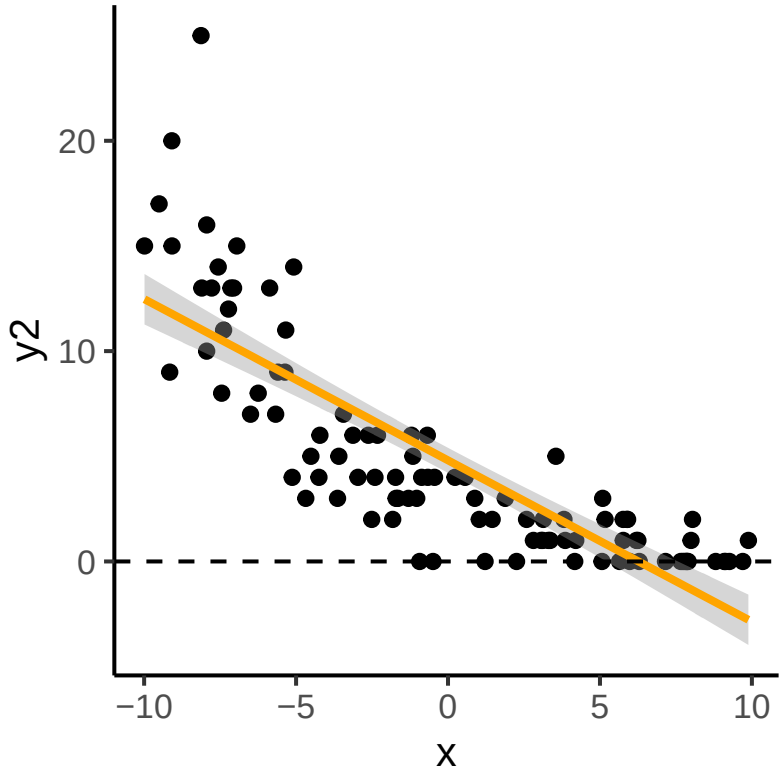


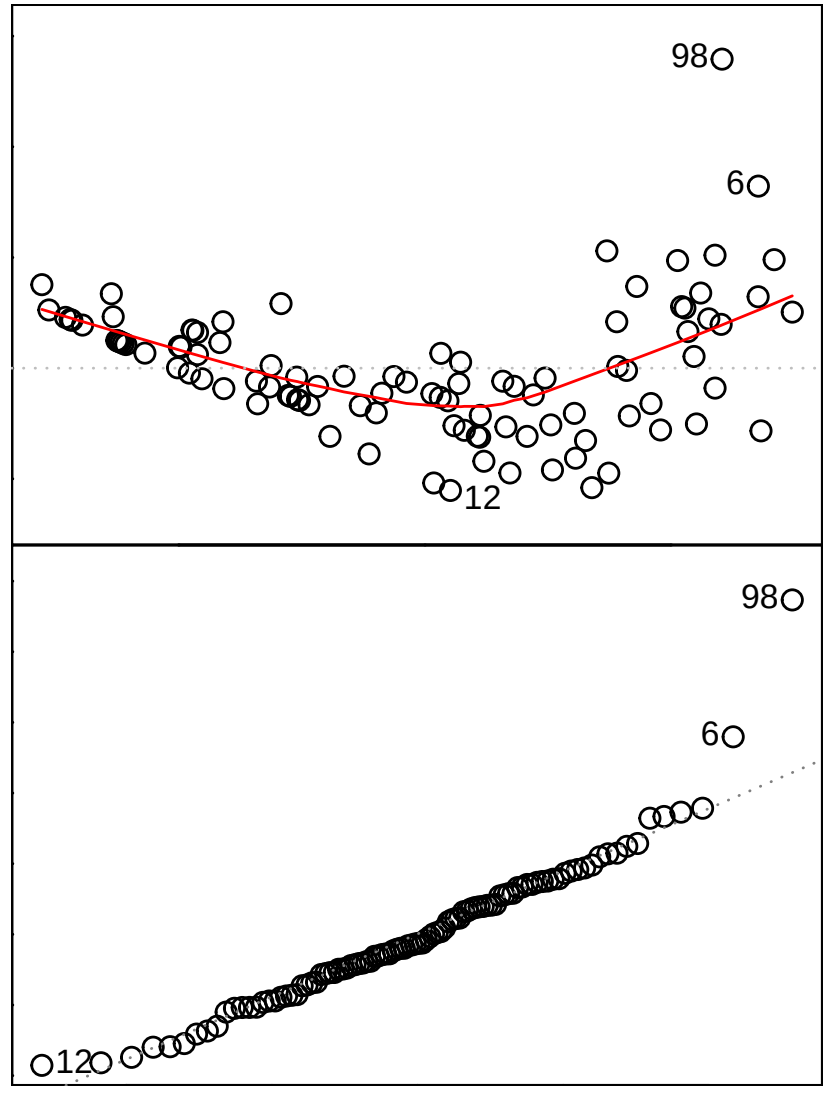

```
lm(y1~poly(x,2),data=d1)
```

log and *square-root* transformations are common

- Warning: Polynomials can do weird things; consider whether this is biologically reasonable!

Problem 2a: Non-normal response

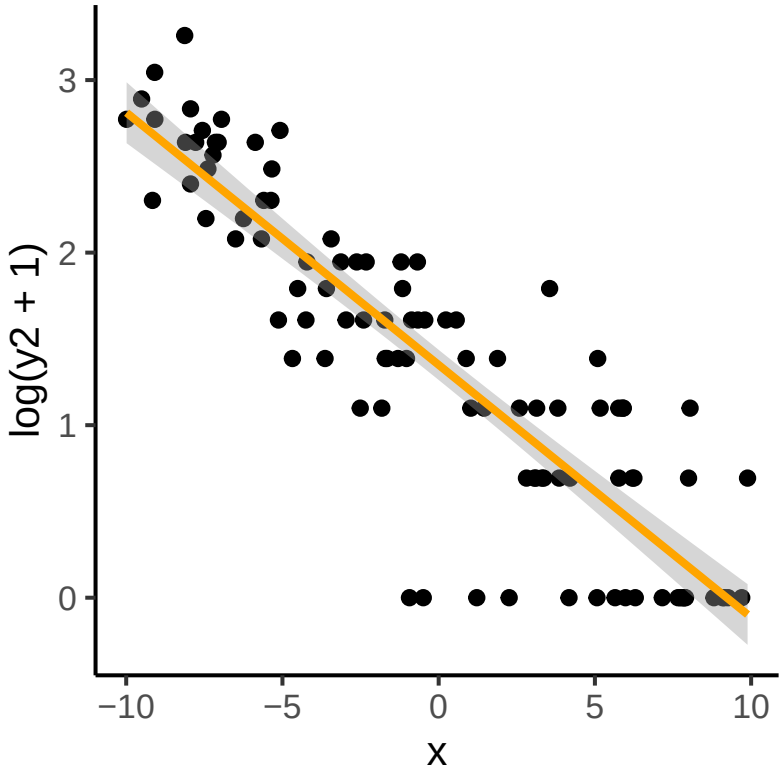


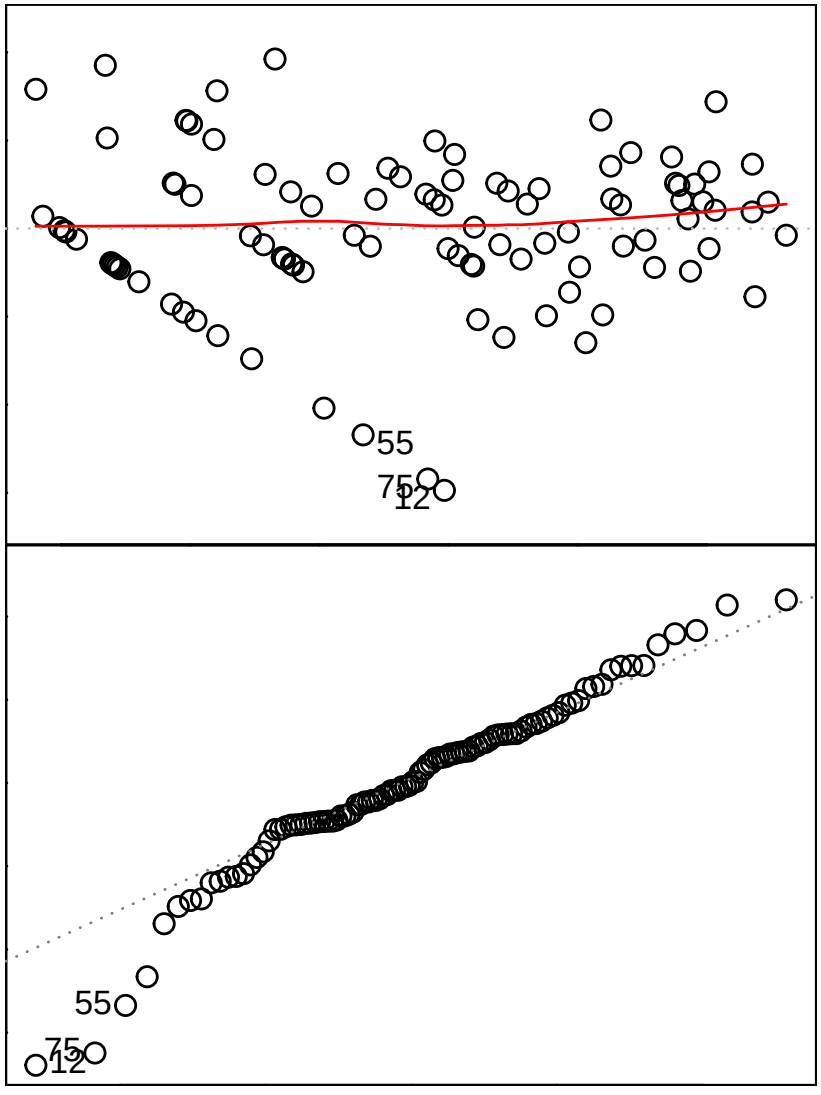



```
lm(y2~x,data=d1)
```

$y2$ is count data (integers ≥ 0). *Very* common in ecological data.

Solution: transform data to meet assumptions

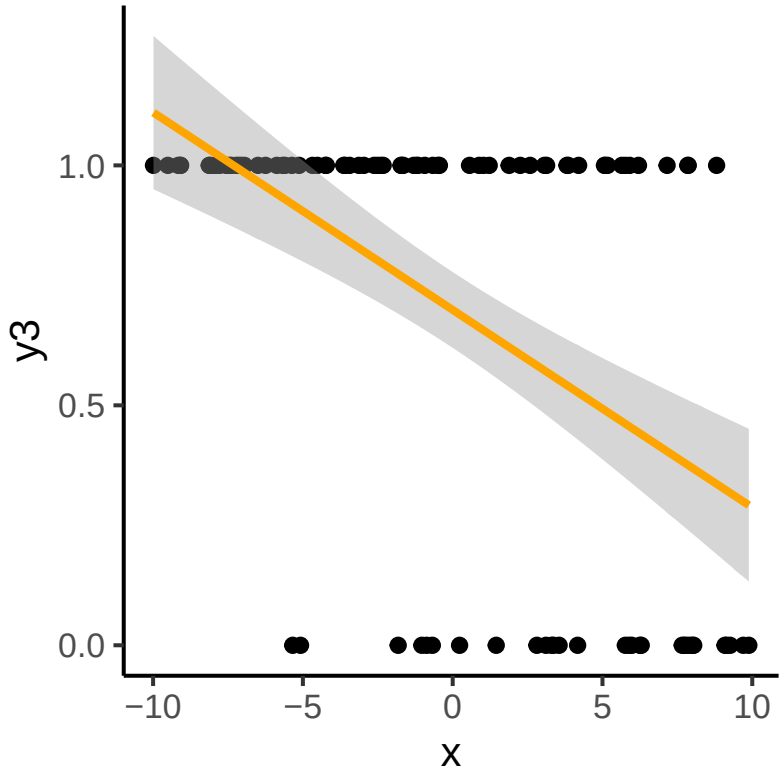


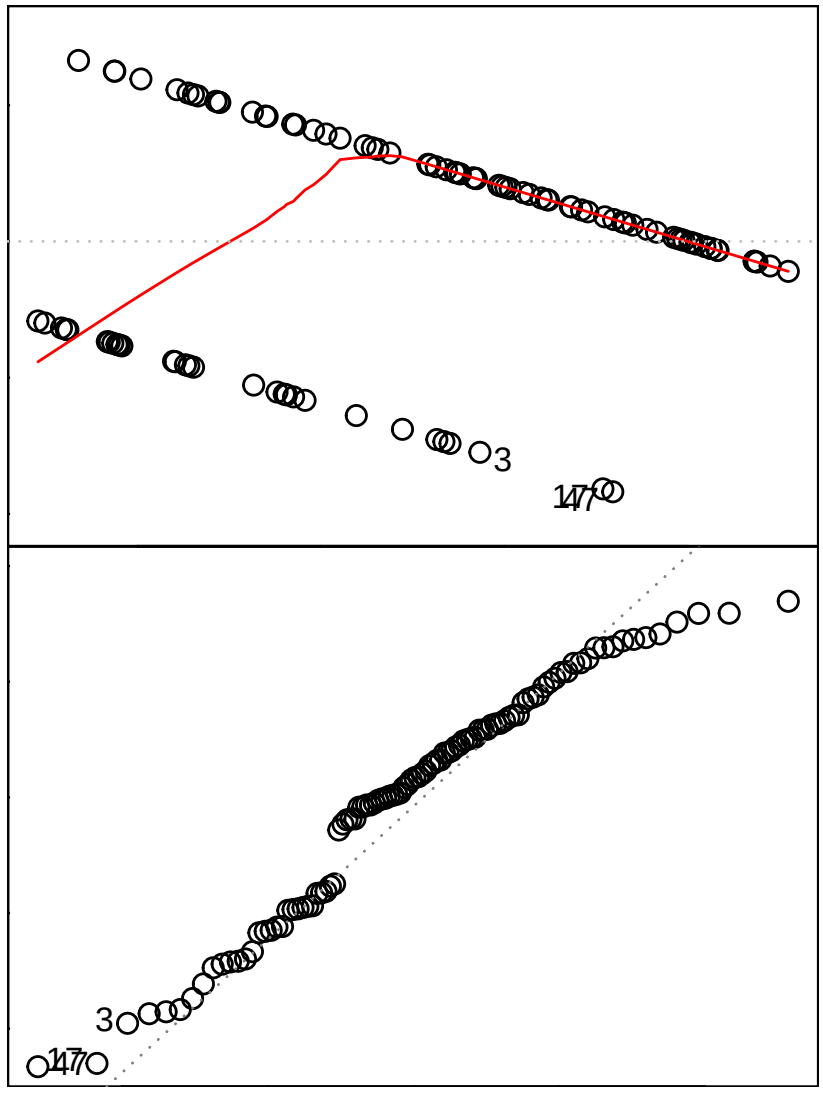



```
lm(log(y2+1)~x,data=d1)
```

Square-root transformations are also common

Problem 2b: Non-normal response

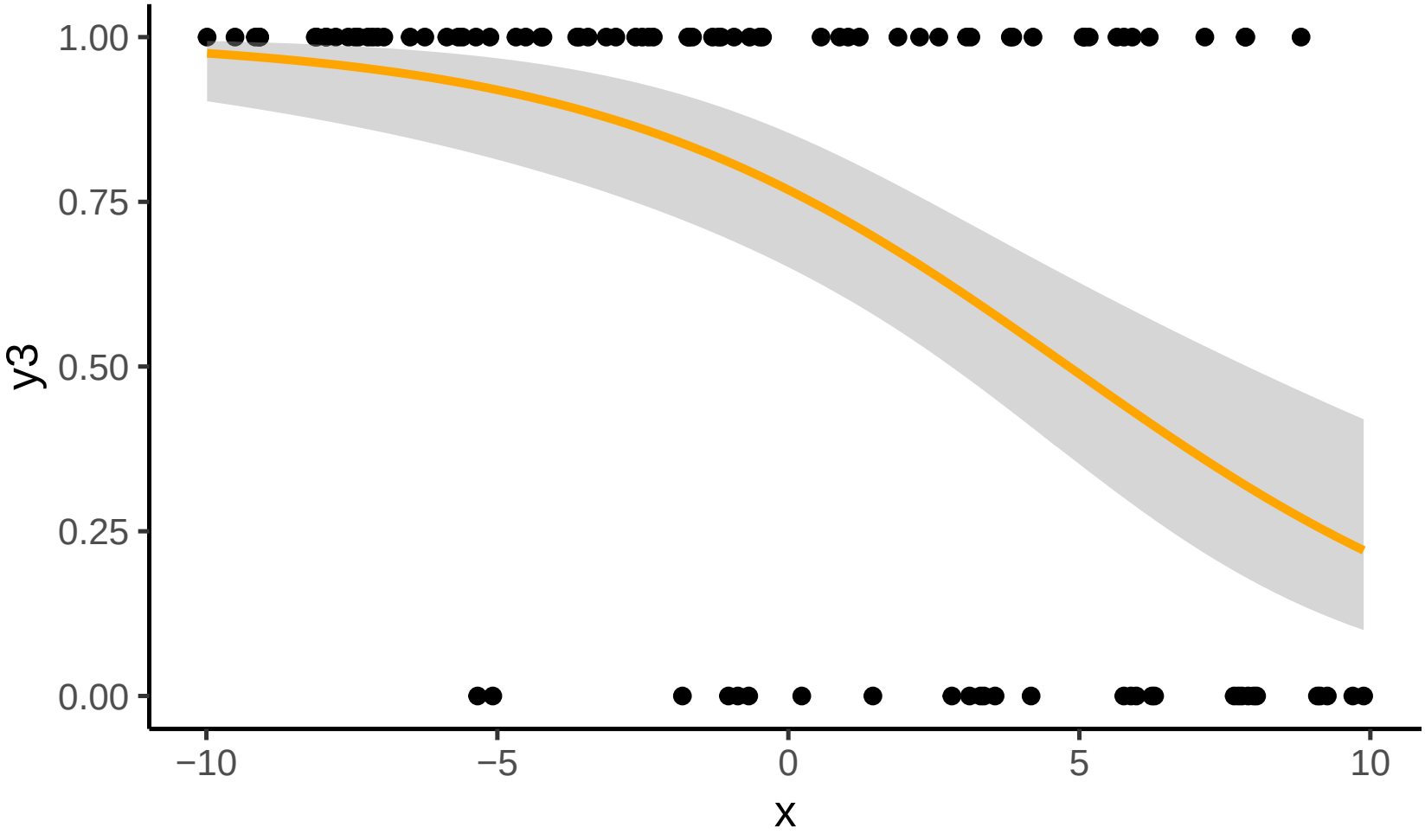





```
lm(y3~x,data=d1)
```

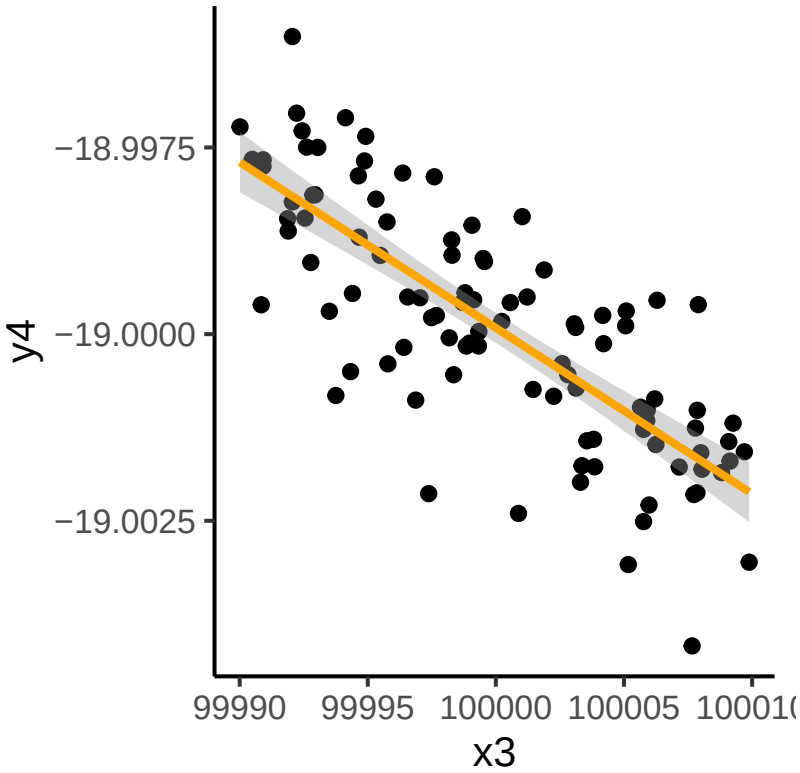
$y3$ is binomial data (success/failure, 0 or 1). *Very* common in ecological data.

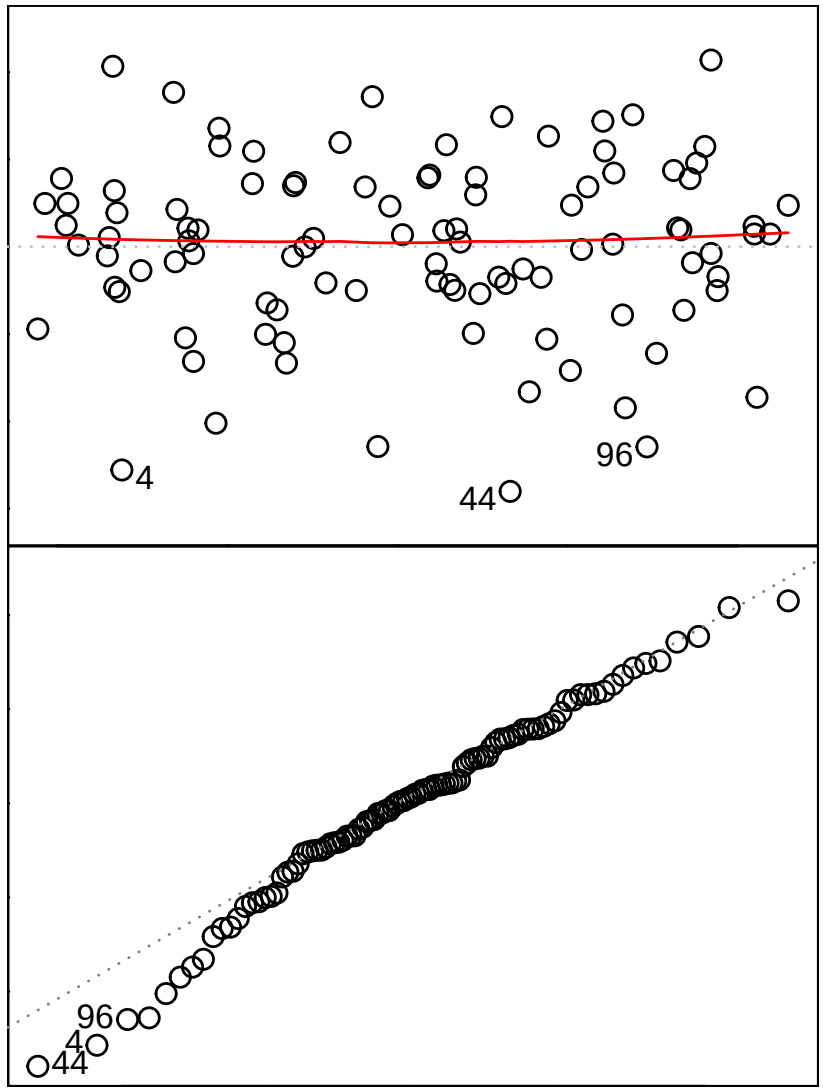
Solution: use a Generalized Linear Model (GLM)



- This is a topic for another lecture. Hold tight!

Problem: variables are on different scales

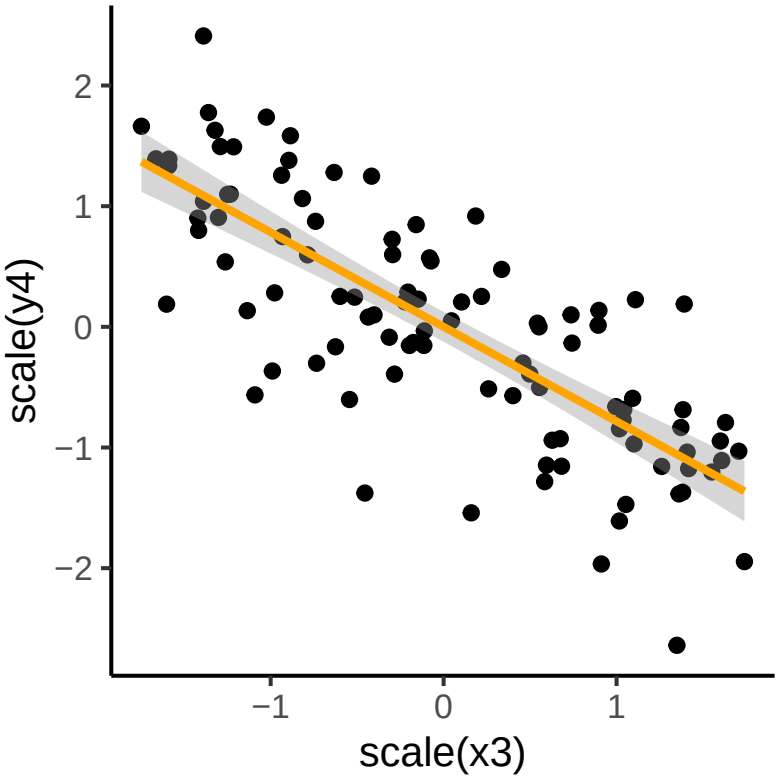


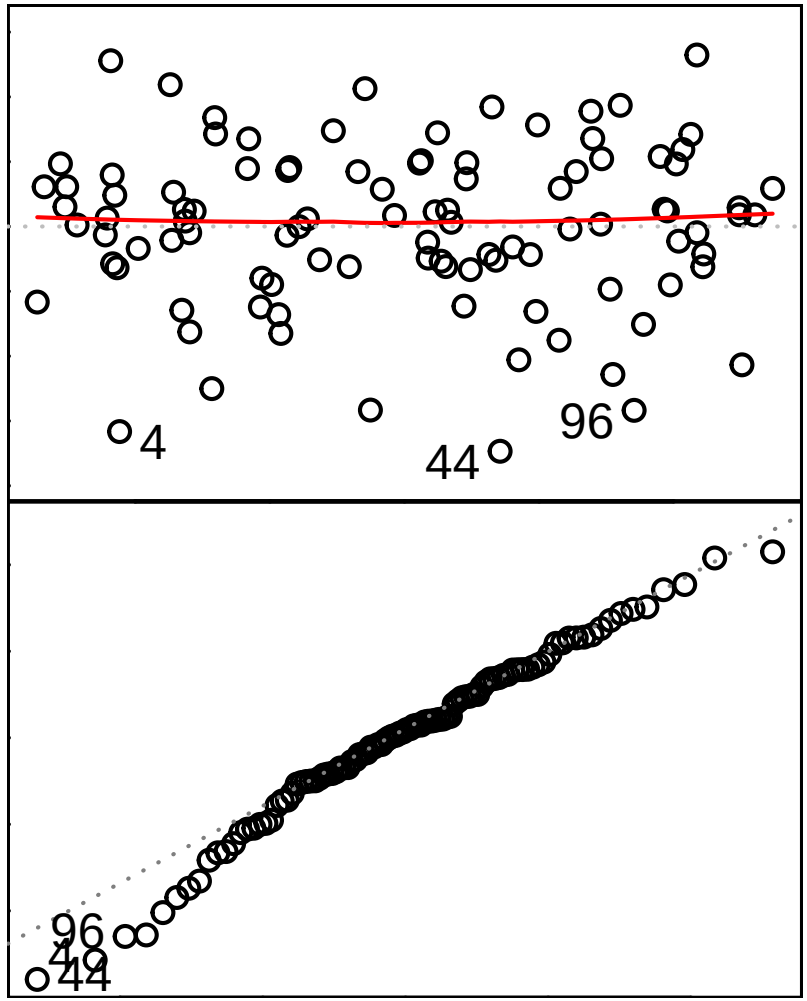



```
lm(y4~x3,data=d1)
```

- $y4$ is tiny, while $x3$ is huge
- OK for now, but can cause problems when fitting complicated models (GLMs)

Solution: scale data/predictors before fitting






```
#Subtracts mean, divides by SD
d1$s.y4 <- scale(y4)
d1$s.x3 <- scale(x3)
lm(s.y4~s.x3,data=d1) #Refit
```

- Residuals are the same as before
- Coefficients are now related to *scaled* data and predictor

But wait... there's more (assumptions)!

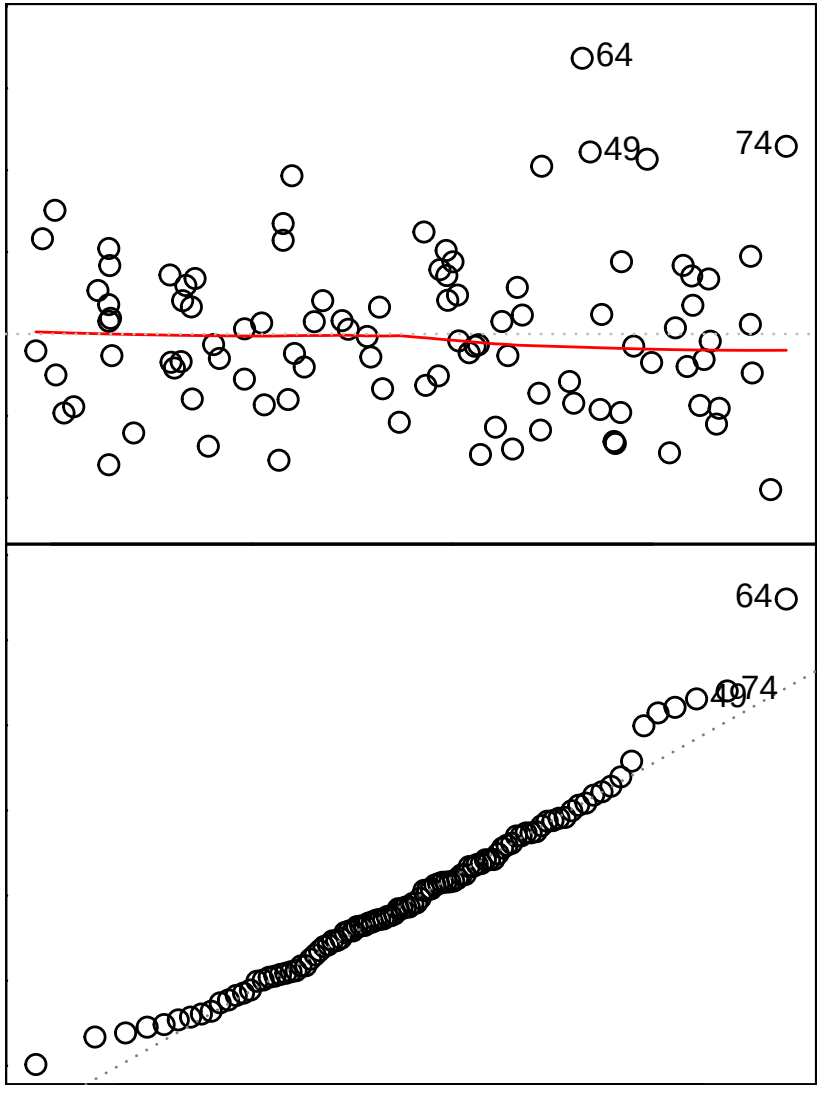
One more assumption:

4. If you have 2+ predictors in your model, the predictors are not related to each other

- Say we have 2 predictors, x and $x2$:

```
lm(y0~x+x2,data=d1)
```

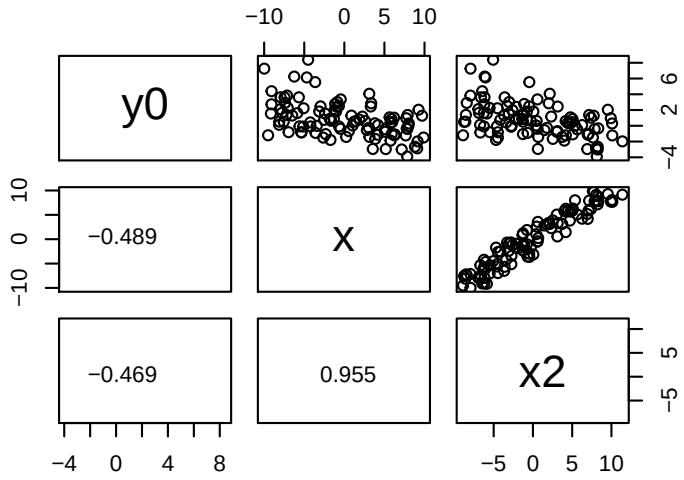
- Model fits, and residuals look OK, but there's trouble ahead!



Uh oh! Collinearity!

```
#Function to print correlation (r) value
corText <- function(x,y){
  text(0.5,0.5,round(cor(x,y),3))
}

#Pairplot of y0, x, and x2
pairs(d1[,c('y0','x','x2')],lower.panel=corText)
```



`pairs()` is useful for looking at relations among your data

- x and $x2$ mean basically the same thing!
- Also revealed using variance-inflation factors (VIFs):

```
library(car)
#VIF scores:
# 1 = no problem
# 1-5 = some problems
# 5+ = big problems!
vif(m2)
```

```
##          x         x2
## 11.31812 11.31812
```

Is collinearity really that bad?

```
#Correct model
m1 <- lm(y0~x,data=d1)
```

	Estimate	Std. Error	Pr(>|t|)
(Intercept)	0.7851936	0.1943002	0.0001059
x	-0.1900346	0.0342596	0.0000002

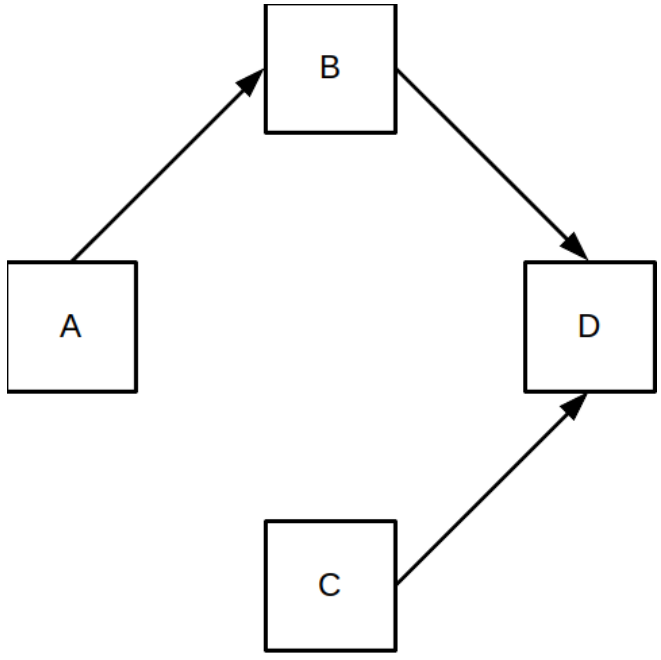
```
#Incorrect model
m2 <- lm(y0~x+x2,data=d1)
```

	Estimate	Std. Error	Pr(>|t|)
(Intercept)	0.7860300	0.1955770	0.0001155
x	-0.1812556	0.1158464	0.1209288
x2	-0.0094931	0.1196074	0.9369028

- Increases SE of each term, so model may "miss" important terms
- Gets worse with increasing correlation, or if many terms are correlated!

How do we fix this? Depends on your goals:

1. I care about predicting things
   - Use dimensional reduction (e.g. PCA) and re-run model
2. I care about what's causing things
   - Design experiment to separate cause and effect
   - Think about what is causing what. *Graphical models* are helpful for this
     - Not all variables have to be included!



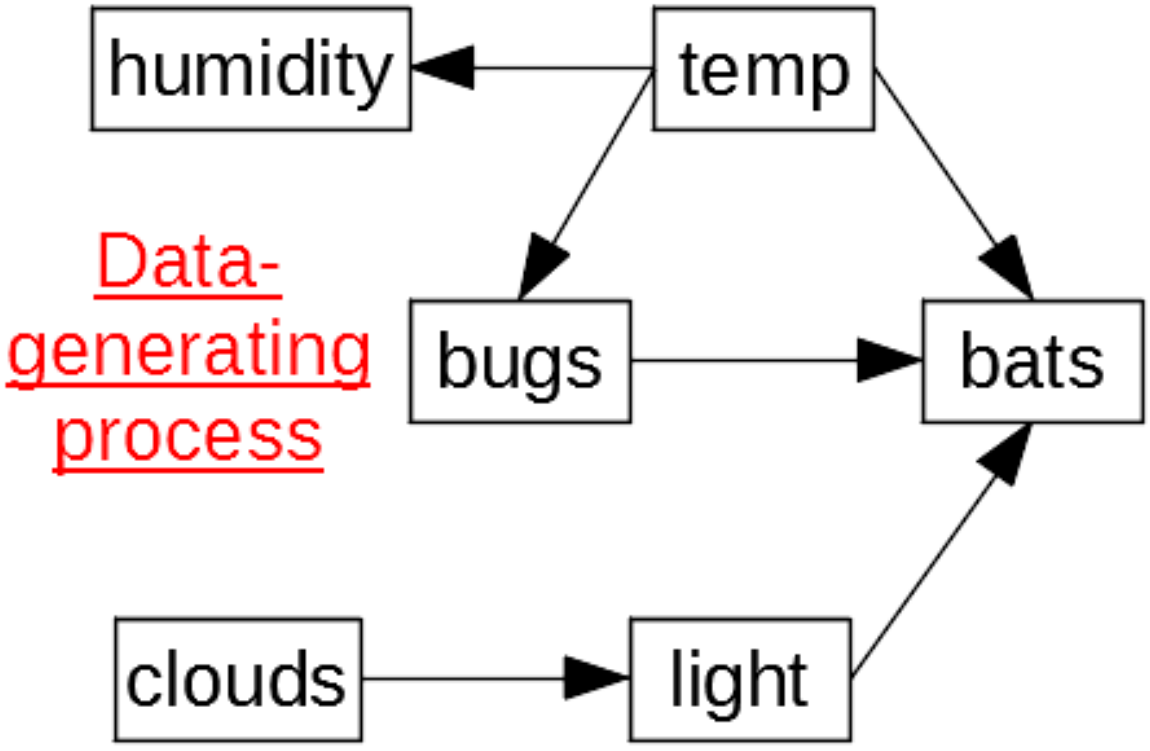
- Simple graphical model, where the effect of A on D is *mediated* by B.
- "Correct" `lm` model of D:

```
lm(D ~ B + C)
```

A challenger approaches!

- Guess what... more bat data! This time there are 6 variables that were measured. We're interested in predicting `bats` (counts of bats per night).
- Formulate a causal model that seems reasonable
    - Draw it out on paper/in PowerPoint using flow diagrams
- Fit an `lm` model of `bats` from your causal model, check the assumptions, and update as necessary

Here's the answer



This is the **true** process that generated the data. Model for `bats` should look like:

```
lm(log(bats+0.1)~poly(temp,2)+light+bugs,data=dat)
```