Linear models 3

Models behaving badly

Samuel Robinson, Ph.D.

October 23, 2020

Motivation

1. Are my model results reliable?

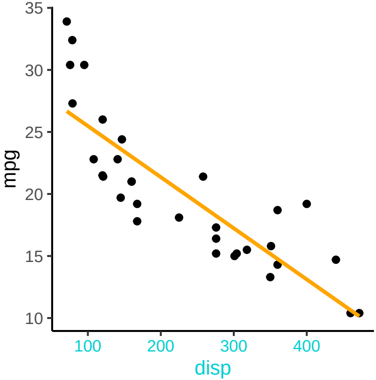
- Residual checks
- Transformations
- Collinearity

2. How do I tell if terms are important or not?

- Drop-1 ANOVA
- Wald t-tests
- How much stuff should I put into my model?

Are my model results reliable?

Assumptions of linear regression



There are 3 main assumptions to this model:

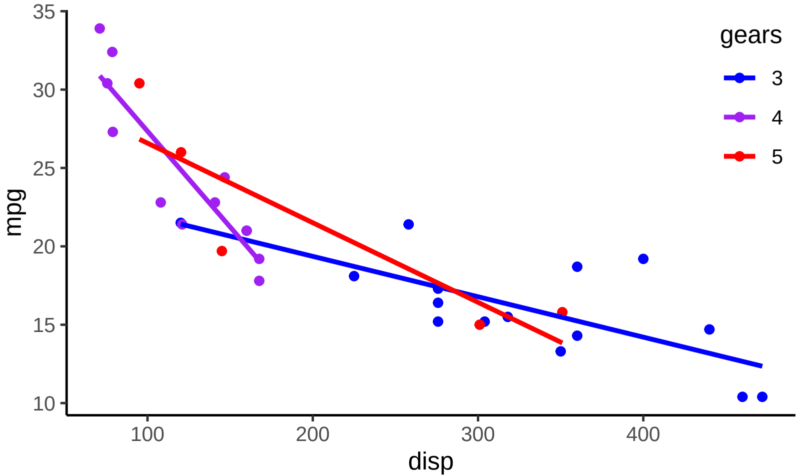
1. The relationship between *disp* and *mpg* is linear
2. *mpg* (the data) is Normally distributed around $\hat{mpg}$ (the line)
3. σ is the same everywhere

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?

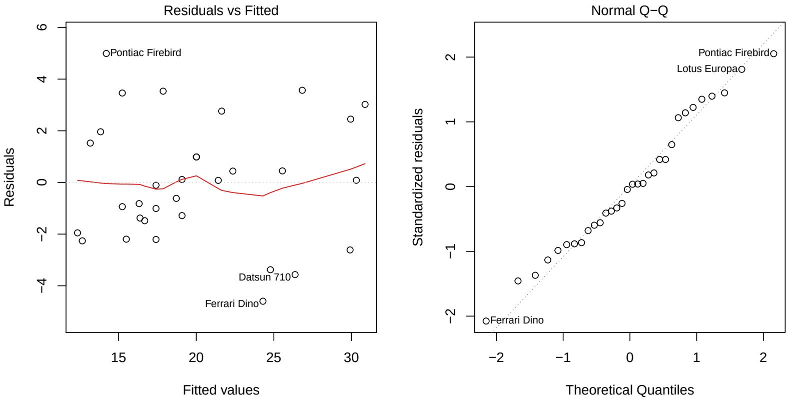


Difficult to see if the assumptions are met

Solution: residual checks

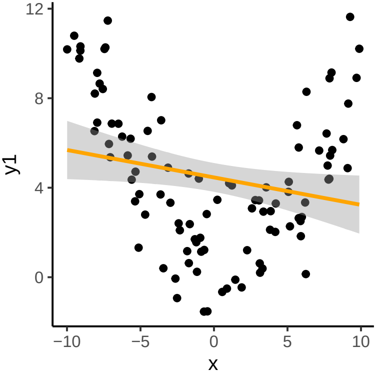
Some common ways of checking the assumptions: **residual plots**

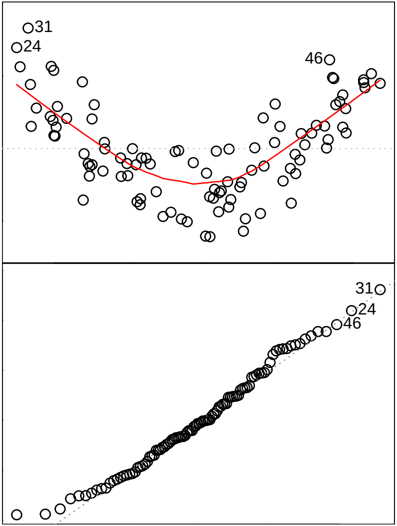
```
mod1 <- lm(mpg~disp*factor(gear),data=mtcars)
par(mfrow=c(1,2),mar=c(3,3,1,1)+1)
plot(mod1, which=c(1,2))
```



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

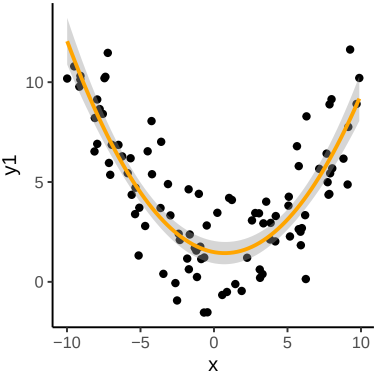


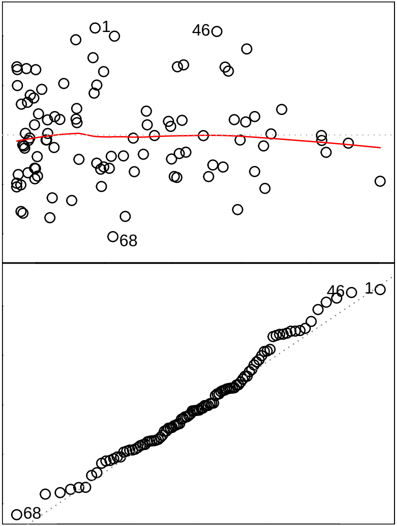


```r
lm(y1~x,data=d1)
```

$y1$ clearly follows a hump-shaped relationship

Solution: use a polynomial model

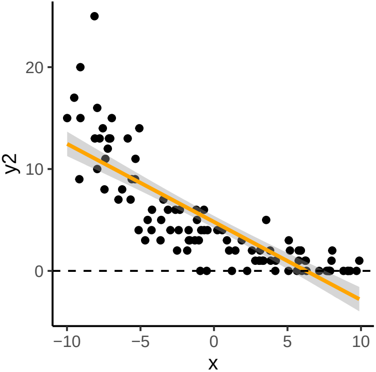


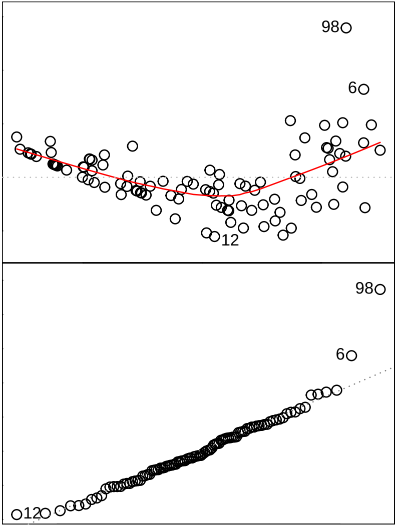


```
lm(y1~poly(x,2),data=d1)
```

Warning: Polynomials can do weird things, especially at the edges of the distribution. Consider whether this is biologically reasonable!

Problem 2a: Non-normal response

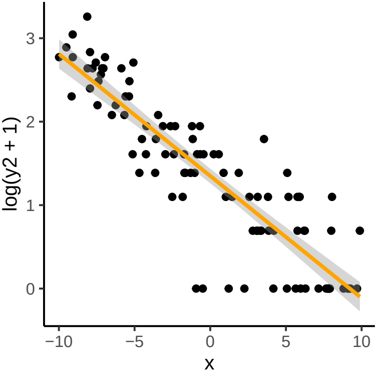


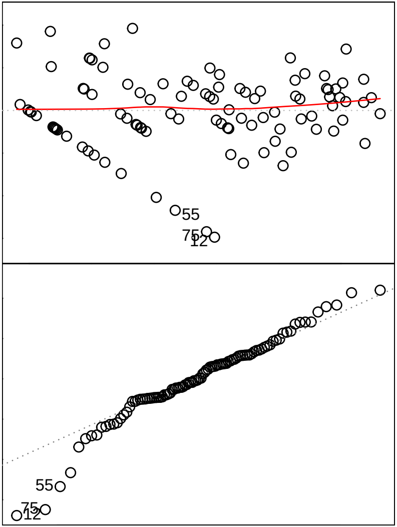


```
lm(y2~x,data=d1)
```

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

Solution: transform data to meet assumptions

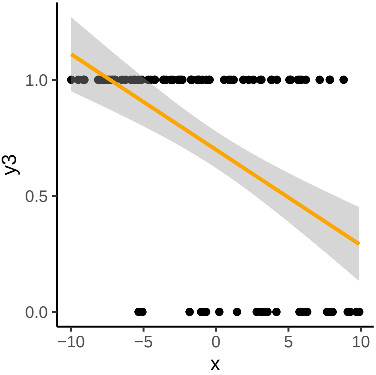




```
lm(log(y2+1)~x,data=d1)
```

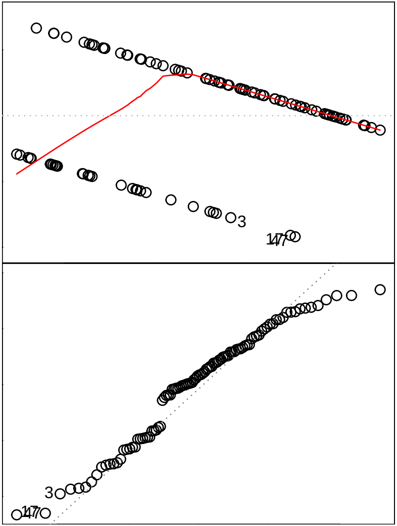
Square-root transformations are also common

Problem 2b: Non-normal response

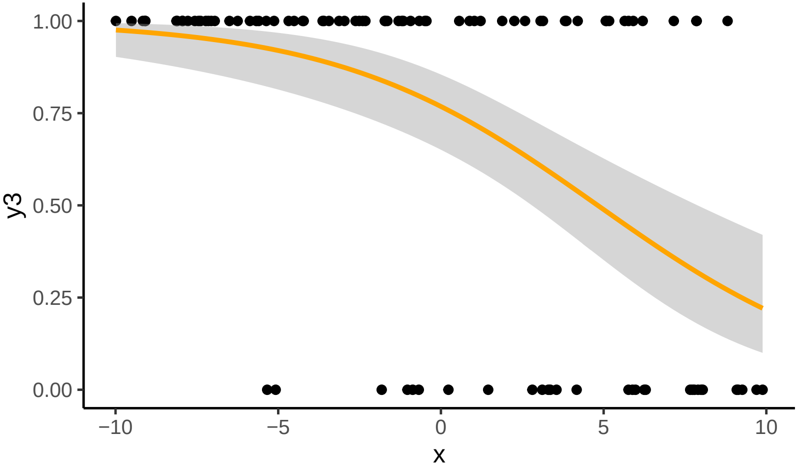


```r
lm(y3~x,data=d1)
```

*y*3 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!

But wait… there's more (assumptions)!

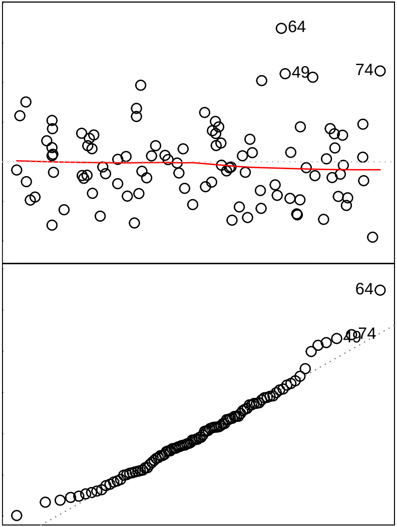
- Additional assumption for models with many predictors:

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)
```

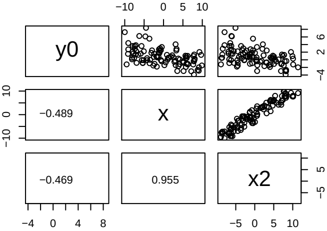
- Model fits, and residuals look OK, but there's trouble ahead!



Here comes trouble!

```
#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)
```



- 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 this problem 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

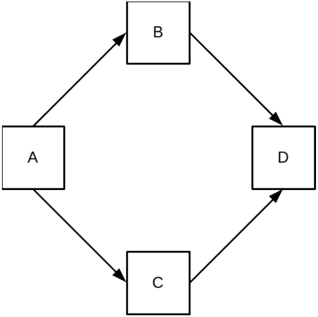


Figure 1: A simple graphical model

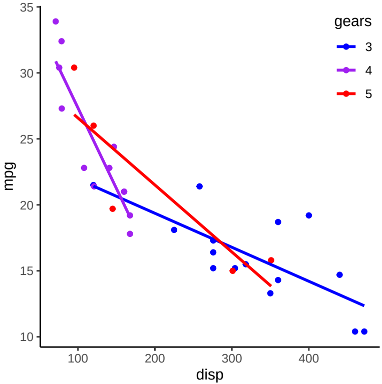
How do I tell if terms are important or not?

The mpg model, once again:

```
m2 <- lm(mpg~disp*factor(gear)
         data=mtcars)
```

	Estimate	Std. Error	Pr(>|t|)
(Intercept)	24.5156	2.4624	0.0000
disp	-0.0258	0.0073	0.0015
factor(gear)4	15.0520	3.5580	0.0003
factor(gear)5	7.1454	3.5359	0.0537
disp:factor(gear)4	-0.0964	0.0213	0.0001
disp:factor(gear)5	-0.0250	0.0133	0.0717

- This tells us about individual coefficients (slopes and intercepts), but. . .
- What if we're interested in entire factors?



- e.g. "Is *gears* important as a group for predicting *mpg*?"

Relative strength of terms:

How do I check if the things that I put in my model are useful for predicting the thing that I'm interested in?

1. drop-1 (Type III) ANOVA for *entire factors*
   - e.g. "Does adding *gear* matter for predicting *mpg*?"
   - Tests for changes in sum of squares with factor
2. Wald t-scores for *levels of factors*
   - e.g. "Is the coefficient for *gear3* different from *gear4?*"
   - Tests whether a coefficient = 0, given the estimated value (mean) and the variablity (SE) of the coefficient

p-values are only meaningful if the model assumptions are valid!

drop-1 ANOVA

```
#mpg depends on gears
mod1 <- lm(mpg ~ factor(gear), data = mtcars)
drop1(mod1,test=’F’) #Effect of gears is very strong
```

```
## Single term deletions
##
## Model:
## mpg ~ factor(gear)
##              Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                     642.8 102.00
## factor(gear)  2    483.24 1126.0 115.94  10.901 0.0002948 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
#mpg depends on disp
mod2 <- lm(mpg ~ disp, data = mtcars)
drop1(mod2,test=’F’) #Effect of disp is also very strong
```

```
## Single term deletions
##
## Model:
## mpg ~ disp
##        Df Sum of Sq     RSS     AIC F value   Pr(>F)
## <none>               317.16  77.397
## disp    1    808.89 1126.05 115.943  76.513 9.38e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

drop-1 ANOVA

```
#mpg depends on disp and gear
mod3 <- lm(mpg ~ disp + factor(gear), data = mtcars)
drop1(mod3,test=’F’) #Effect of disp is very strong, and erases the effect of gear
```

```
## Single term deletions
##
## Model:
## mpg ~ disp + factor(gear)
##              Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>                    317.01  81.383
## disp          1    325.79 642.80 102.003 28.7755 1.025e-05 ***
## factor(gear)  2      0.15 317.16  77.397  0.0065    0.9935
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
#mpg depends on disp interacted with gear
mod4 <- lm(mpg ~ disp*factor(gear), data = mtcars)
drop1(mod4,test=’F’) #Interaction effect is strong. Why are disp and gear not shown?
```

```
## Single term deletions
##
## Model:
## mpg ~ disp * factor(gear)
##                   Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                         172.87 65.978
## disp:factor(gear)  2    144.14 317.01 81.383  10.839 0.0003771 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Wald t-scores

- Wald t-scores are shown in model `summary`
- t-score = mean÷SD
- p-value comes from Student’s t-distribution (similar to Normal, but has longer tails depending on sample size)

```
summary(mod1)
```

```
##
## Call:
## lm(formula = mpg ~ factor(gear), data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7333 -3.2333 -0.9067  2.8483  9.3667
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     16.107      1.216  13.250 7.87e-14 ***
## factor(gear)4    8.427      1.823   4.621 7.26e-05 ***
## factor(gear)5    5.273      2.431   2.169   0.0384 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.708 on 29 degrees of freedom
## Multiple R-squared:  0.4292, Adjusted R-squared:  0.3898
## F-statistic:  10.9 on 2 and 29 DF,  p-value: 0.0002948
```

Comparing between intercepts

- If you've found that *gear* is important, are the levels different from each other?
- If number of levels = 3+, then you need to account for *multiple comparisons*
- One common method: Bonferroni correction

```
library(multcomp) #Loads the multcomp package (needs to be installed first)
mod1Comp <- glht(mod1, linfct = mcp('factor(gear)'='Tukey')) #Fits multcomp object using gear
summary(mod1Comp,test=adjusted('bonferroni')) #gear4 different from gear3 only
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = mpg ~ factor(gear), data = mtcars)
##
## Linear Hypotheses:
##            Estimate Std. Error t value Pr(>|t|)
## 4 - 3 == 0    8.427      1.823   4.621 0.000218 ***
## 5 - 3 == 0    5.273      2.431   2.169 0.115267
## 5 - 4 == 0   -3.153      2.506  -1.258 0.654971
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```