

# Linear models

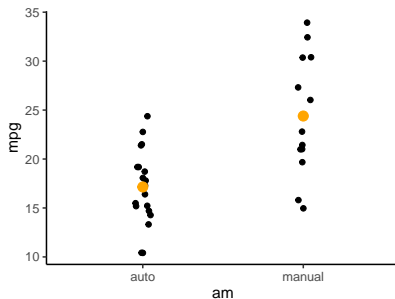
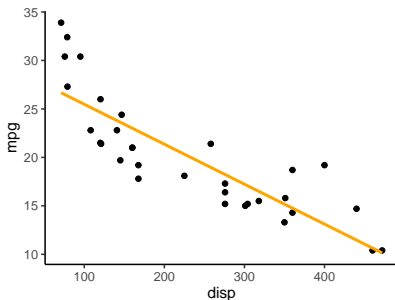
## How do they work?

Samuel Robinson, Ph.D.

October 8, 2020

# Motivation

- *I have some bivariate data (2 things measured per row), and I want to know if they're related to each other*
- *I have 2+ groups of data, and I want to know whether the means are different*



# Model terminology

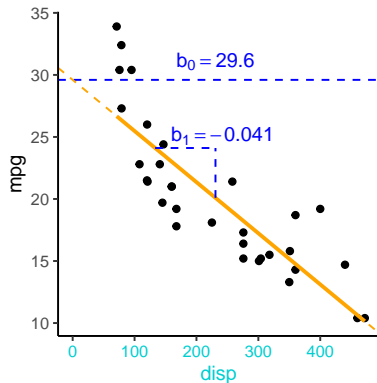
- All linear models take the form:

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 \dots + b_jx_j$$
$$y \sim \text{Normal}(\hat{y}, \sigma)$$

- $y$  is the thing you're interested in predicting
- $\hat{y}$  is the *predicted value* of  $y$
- $x_1 \dots x_j$  are *predictors* of  $y$
- $b_1 \dots b_j$  are *coefficients* for each predictor  $x_i$
- $b_0$  is the *intercept*, a coefficient that doesn't depend on predictors
- $y \sim \text{Normal}(\hat{y}, \sigma)$  means:
  - “ $y$  follows a Normal distribution with mean  $\hat{y}$  and SD  $\sigma$ ”

This may look terrifying, but let's use a simple example:

## Example



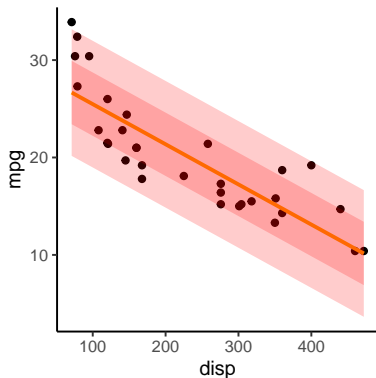
$$\hat{mpg} = b_0 + b_1 disp$$

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

- $mpg$  is the thing you're interested in predicting
- $\hat{mpg}$  is the *predicted value* of  $mpg$
- $disp$  is the *predictor* of  $mpg$
- $b_0$  is the *intercept*,  $b_1$  is the *coefficient* for  $disp$
- $mpg \sim Normal(\hat{mpg}, \sigma)$  means:
  - “ $mpg$  follows a Normal distribution with mean  $\hat{mpg}$  and SD  $\sigma$ ”
- $\sigma$  isn't displayed on the figure. Where is it?

## Example (cont.)

$\sigma$  isn't displayed on the figure. Where is it?



- $\sigma$  is the “leftover” or “residual” variance
- i.e. variation between samples that the model couldn't explain
- Since  $y \sim \text{Normal}(\hat{y}, \sigma)$ , this means that points are normally distributed around the *entire line* of  $\hat{y}$

# How do I get R to fit this model?

lm is one of the main functions used for linear modeling:

```
#Formula structure: y ~ x  
mod1 <- lm(mpg ~ disp, #mpg depends on disp  
           data = mtcars) #Name of the dataframe containing mpg & disp  
summary(mod1)
```

```
##  
## Call:  
## lm(formula = mpg ~ disp, data = mtcars)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -4.8922 -2.2022 -0.9631  1.6272  7.2305   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept) 29.599855   1.229720  24.070 < 2e-16 ***  
## disp        -0.041215   0.004712  -8.747 9.38e-10 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 3.251 on 30 degrees of freedom  
## Multiple R-squared:  0.7183, Adjusted R-squared:  0.709   
## F-statistic: 76.51 on 1 and 30 DF, p-value: 9.38e-10
```

For a detailed breakdown of lm's output, [click here](#)

# Simulate data

Now that we know how linear models work, we can simulate our own data:

```
#Parameters:
```

```
b0 <- 1 #Intercept
```

```
b1 <- 2 #Slope
```

```
sigma <- 3 #SD
```

```
#Make up some data:
```

```
x <- 0:30 #Predictor values
```

```
#Predicted y values
```

```
pred_y <- b0 + b1*x
```

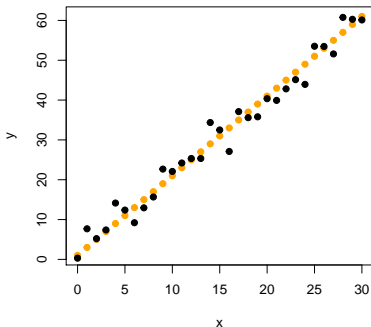
```
#Add "noise" around pred_y
```

```
actual_y <- rnorm(n = length(pred_y),  
                  mean = pred_y,  
                  sd= sigma)
```

```
#Plot the data we just made
```

```
plot(x,pred_y,col='orange',pch=19,  
      ylab='y')
```

```
points(x,actual_y,col='black',pch=19)
```



# Fit a model from simulated data

How does R do at finding the coefficients?

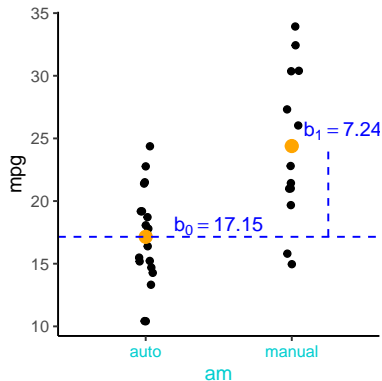
Remember:  $b_0 = 1$ ,  $b_1 = 2$ ,  $\sigma = 3$

```
#Put the simulated data into a dataframe
fakeDat <- data.frame(x = x, y = actual_y, pred = pred_y)
mod1sim <- lm(y ~ x, data = fakeDat) #Fit a linear model
summary(mod1sim)
```

```
##
## Call:
## lm(formula = y ~ x, data = fakeDat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.7568 -1.7623 -0.2176  1.9419  5.3572
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.02974    1.00445    2.021  0.0526 .
## x            1.92670    0.05751   33.499 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.864 on 29 degrees of freedom
## Multiple R-squared:  0.9748, Adjusted R-squared:  0.9739
## F-statistic: 1122 on 1 and 29 DF, p-value: < 2.2e-16
```



# What about categorical data?



This uses *exactly the same* math!

- $mpg$  is the thing you're interested in predicting
- $\hat{mpg}$  is the *predicted value* of  $mpg$
- $am$  is the *predictor* of  $mpg$ 
  - set of 0s and 1s, not continuous
- $b_0$  is the *intercept*,  $b_1$  is the *coefficient* for  $am$

Where is  $\sigma$ ?

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

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

# How do I get R to fit this model?

Syntax is exactly the same for this model

```
#Formula structure: y ~ x  
mod2 <- lm(mpg ~ am, #mpg depends on am  
           data = mtcars) #Name of the dataframe containing mpg & am  
summary(mod2)
```

```
##  
## Call:  
## lm(formula = mpg ~ am, data = mtcars)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -9.3923 -3.0923 -0.2974  3.2439  9.5077   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)   17.147      1.125   15.247 1.13e-15 ***  
## am              7.245      1.764    4.106 0.000285 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 4.902 on 30 degrees of freedom  
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3385   
## F-statistic: 16.86 on 1 and 30 DF, p-value: 0.000285
```

## A challenger approaches!

- Simulate your own data with 2 discrete levels
  - Steal Borrow my code, and change the predictor from continuous to discrete
  - Useful command: `rep` (replicate)
- Use `lm` to fit a model to the data you just simulated
  - How does R do at guessing your coefficients?