

Linear regression

Lines, least squares, and what the output means

*"All models are wrong –
but some are useful."*

-- George Box

Week 1 recap

Kahoot-style warm-up

12 quick questions on Week 1.

Not graded.

Use a pseudonym if you want.

Fast and correct wins.



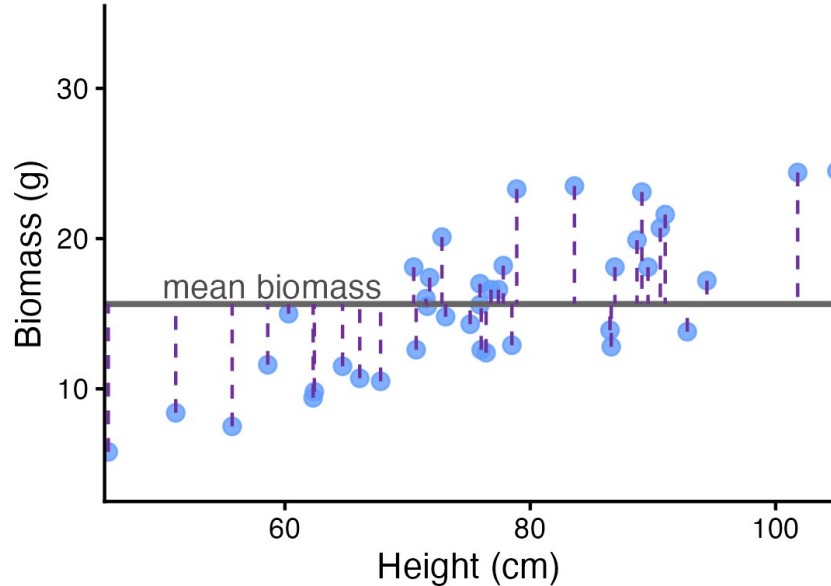
Scan with your phone

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w2-week1-recap](https://greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w2-week1-recap)

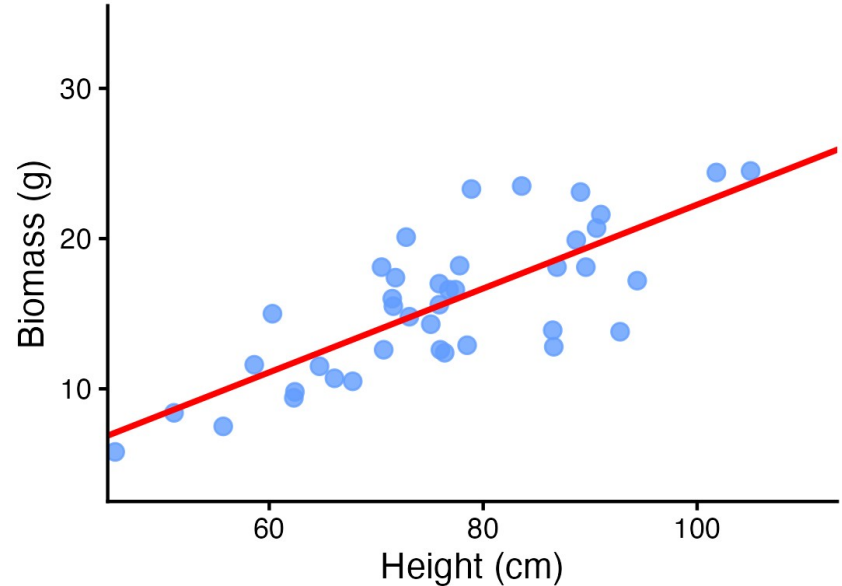
[host screen](#)

Where we are

No information: the mean



Know the height: a line

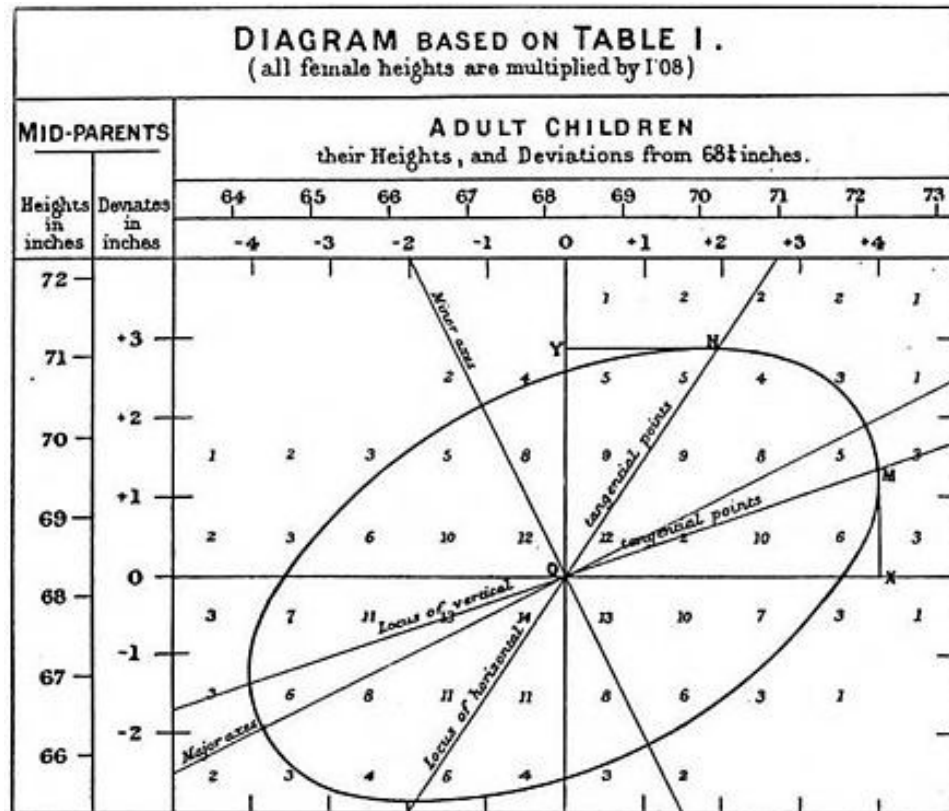


Day 1: “what if you knew one thing about the plant?” Regression = the expected y, **given** **x**.

Goals for Today

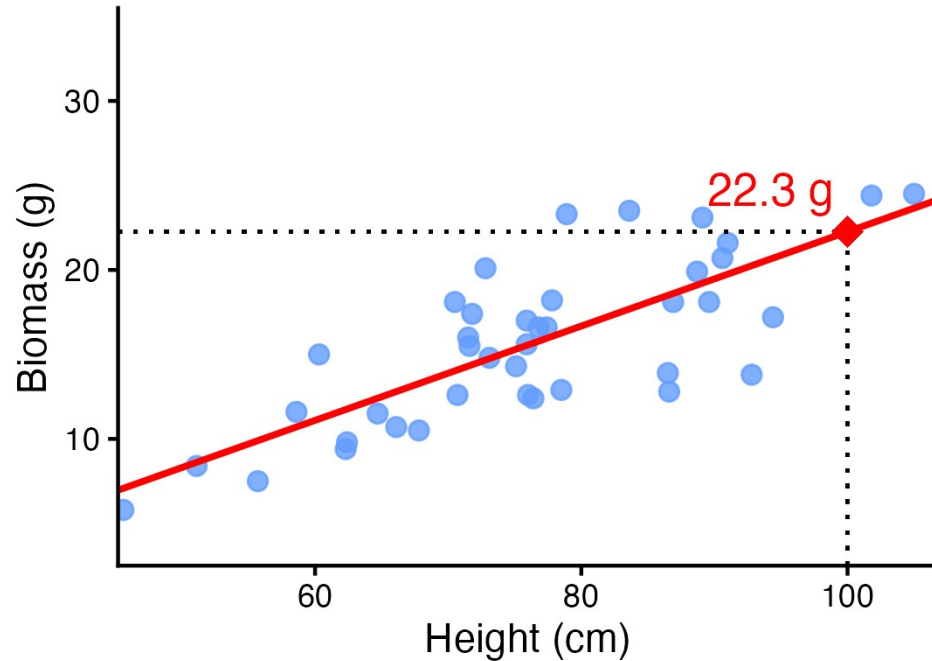
- A line, by least squares
- `lm()`: fit, read, predict
- R^2
- t-tests, F-tests, confidence intervals
- Assumptions, and the two classic fixes

“Regression” described in 1886



Children's heights sat closer to the population mean than their parents': they “**regressed**” to it.

Two jobs for one line



Predict

a 100 cm plant?

Test

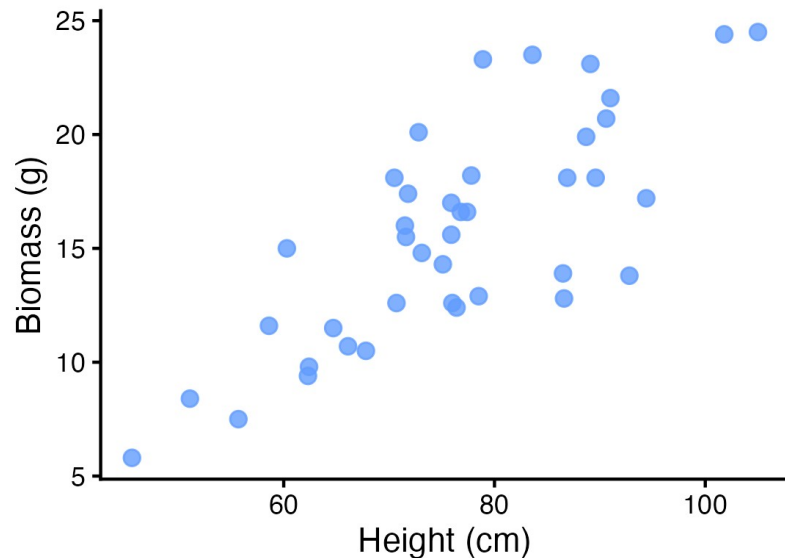
does height matter at all?

Part 1: Fitting a line

Residuals and least squares

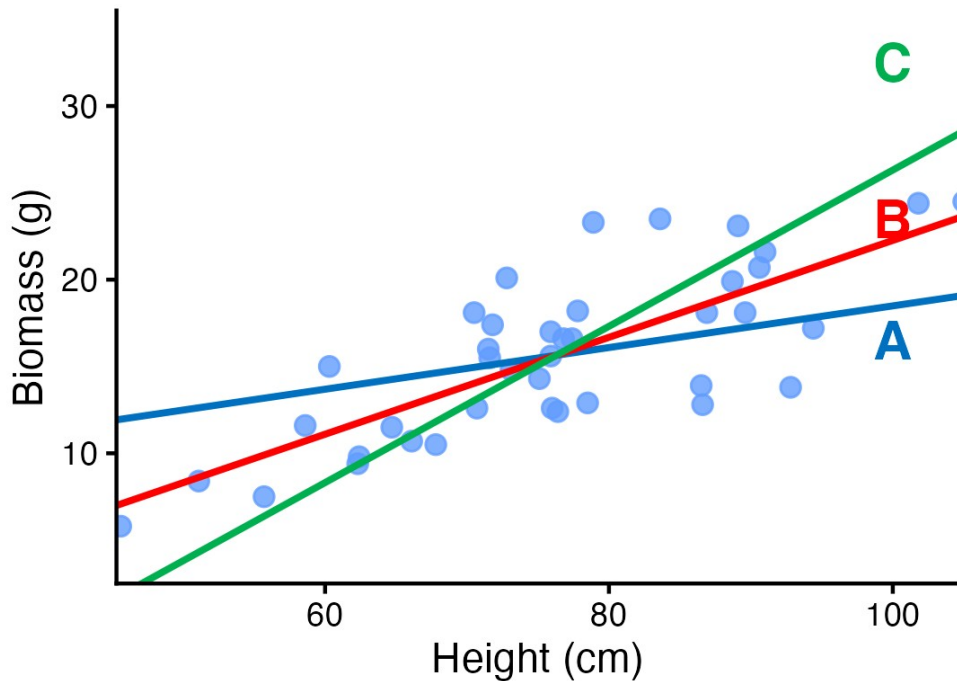
40 wheat plants

```
> wheat <- read.csv("data/wheat_plants.csv")
> head(wheat)
  plant water nitrogen height biomass
1     1    45      89   86.9    18.1
2     2    41       6   71.6    15.5
3     3    21     66   73.1    14.8
4     4    31     70   75.9    15.6
5     5    49    115   91.0    21.6
6     6    49     65   94.4    17.2
> nrow(wheat)
[1] 40
```



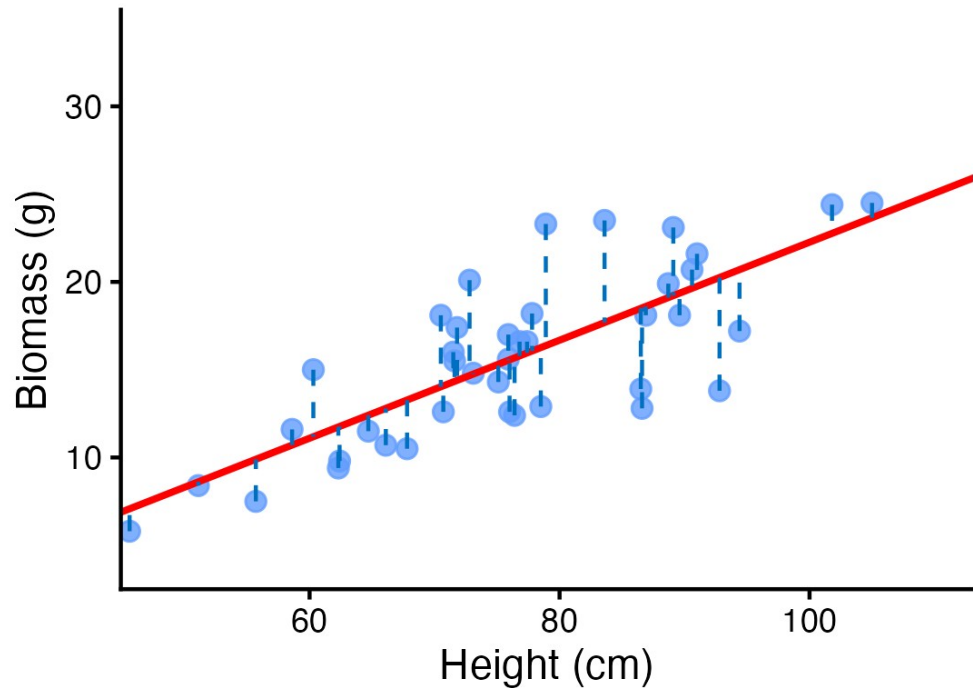
wheat_plants.csv: biomass (g) as a function of height (cm)

Three lines, one question



All three go through the middle of the cloud. Which is **closest to the points**?

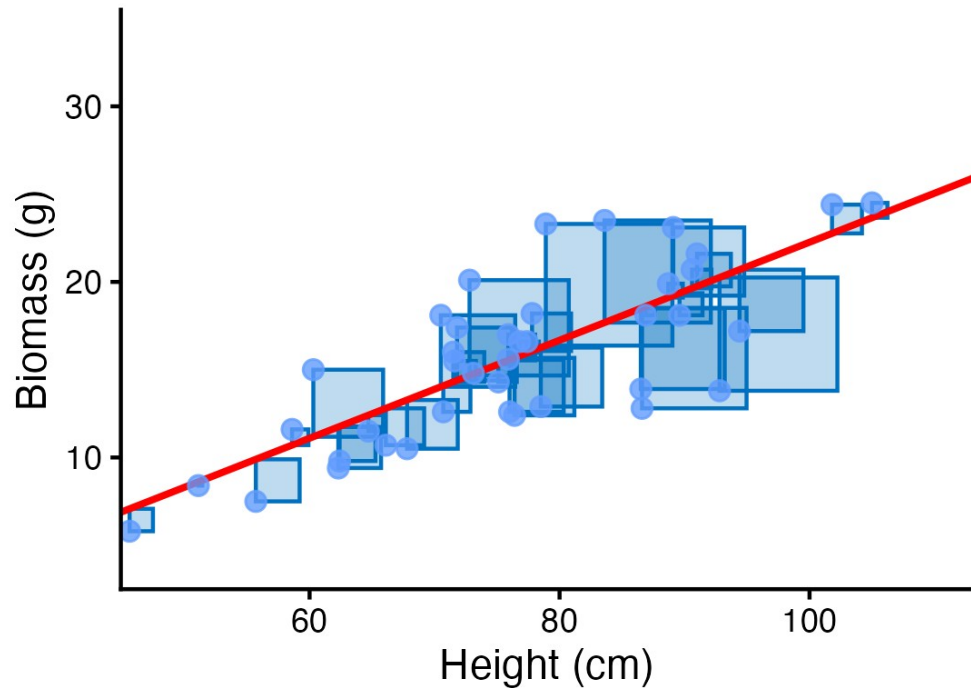
Residuals



$$\hat{y}_i = -5.65 + 0.279 x_i \quad \text{residual} = y_i - \hat{y}_i$$

Each **dashed line**: how far one plant is from the line.

Square them, add them up

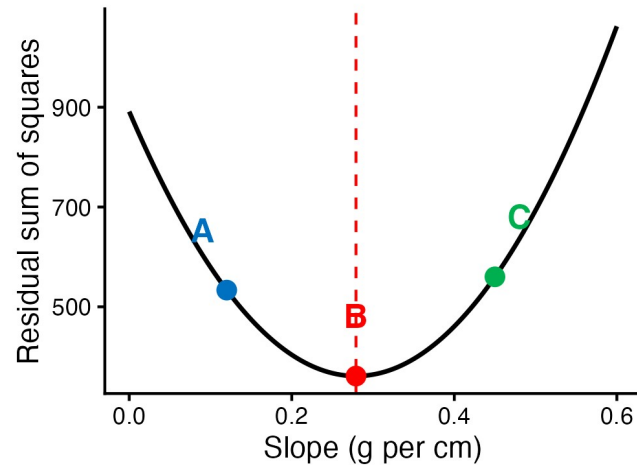


$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n \varepsilon_i^2$$

Total area = **residual sum of squares**

Score the three lines

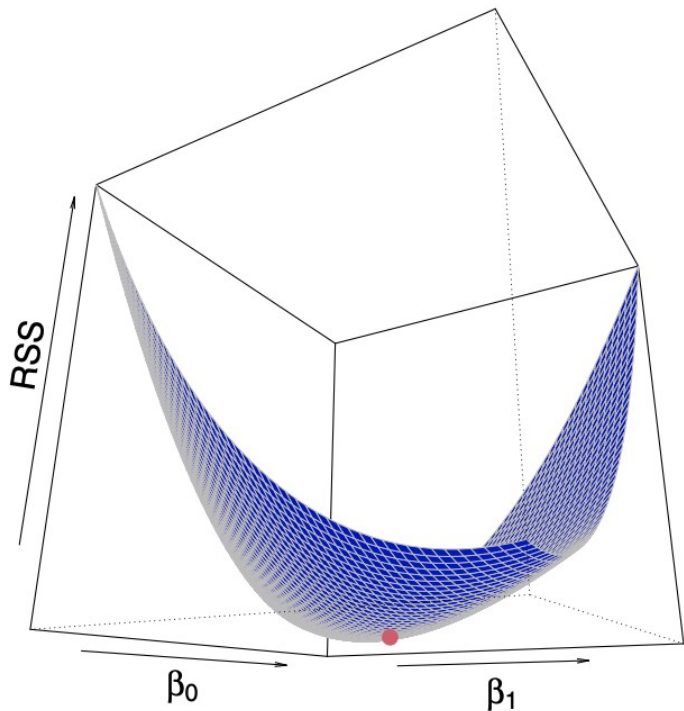
```
> # score a line through the middle of the data by its RSS
> model_line <- function(slope) {
+   b0 <- mean(wheat$biomass) - slope * mean(wheat$height)
+   predicted <- b0 + slope * wheat$height
+   sum((wheat$biomass - predicted)^2)
+ }
> model_line(0.12)      # line A
[1] 533.4545
> model_line(0.279)    # line B
[1] 361.2284
> model_line(0.45)     # line C
[1] 560.0627
```



B: the smallest RSS

$$RSS_A = 533 \quad RSS_B = 361 \quad RSS_C = 560$$

Least squares



Regression finds the line with the **least sum of squares**.

The model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

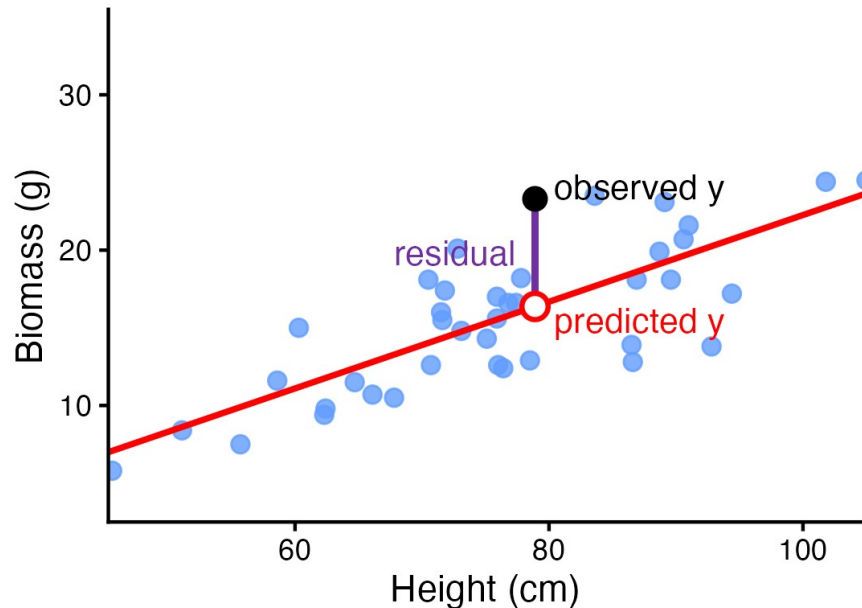
y_i biomass of plant i (response)

x_i height of plant i (predictor)

β_0 intercept: y when $x = 0$

β_1 slope: change in y per unit x

ε_i residual: what the line misses



The slope by hand

$$\hat{\beta}_1 = \frac{\text{cov}(x, y)}{\text{var}(x)} = r \frac{s_y}{s_x}$$

$$\hat{\beta}_1 = \frac{48.7}{174.5} = 0.279 \text{ g per cm}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\beta}_0 = 15.64 - 0.279 \times 76.3 = -5.65 \text{ g}$$

```
> # slope
> b1 <- cov(wheat$height, wheat$biomass) / var(wheat$height)
> b1
[1] 0.2790764
> # intercept
> b0 <- mean(wheat$biomass) - b1 * mean(wheat$height)
> b0
[1] -5.647833
> # the same slope, via r
> r <- cor(wheat$height, wheat$biomass)
> r * sd(wheat$biomass) / sd(wheat$height)
[1] 0.2790764
```

Syntax: the lm() function

```
model <- lm(biomass ~ height, data = wheat)
```

response (y)

“as a function of”

the data frame

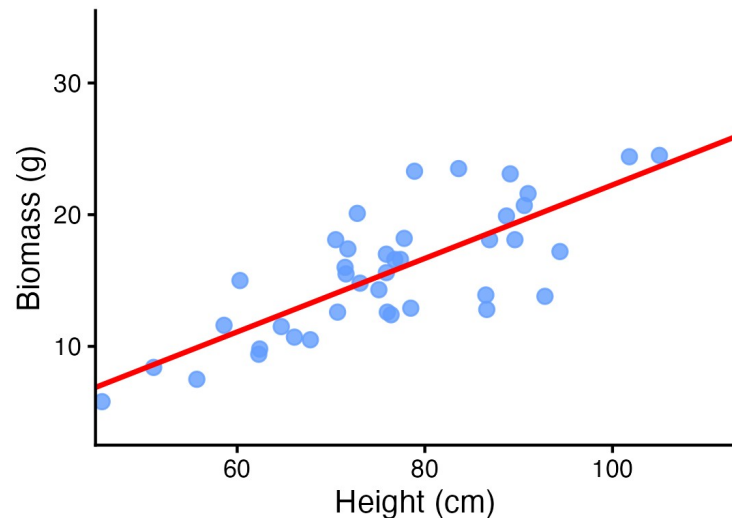
predictor (x)

lm() finds the same line

```
> model <- lm(biomass ~ height, data = wheat)
> model
```

```
Call:
lm(formula = biomass ~ height, data = wheat)
```

```
Coefficients:
(Intercept)      height
   -5.6478      0.2791
```



+0.28 g of biomass per cm of height, on average

Live poll

Number: live histogram

Predict: what is the biomass (g) of a 100 cm wheat plant?



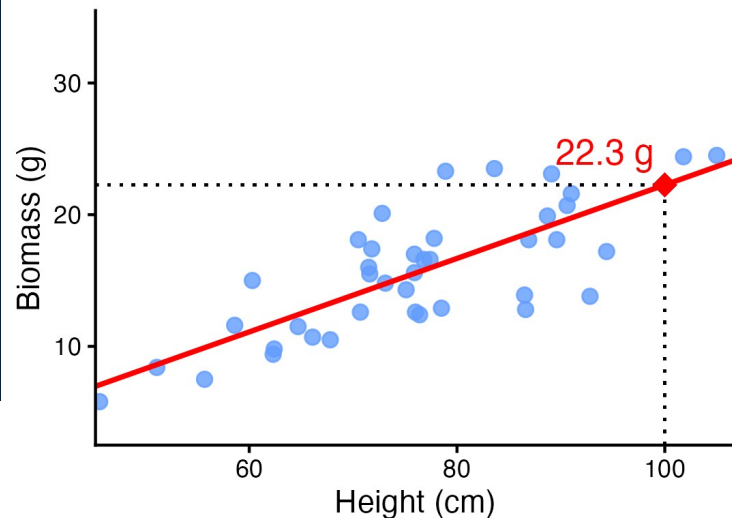
Scan with your phone

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w2-predict](https://greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w2-predict)

[host screen](#)

Predictions: predict()

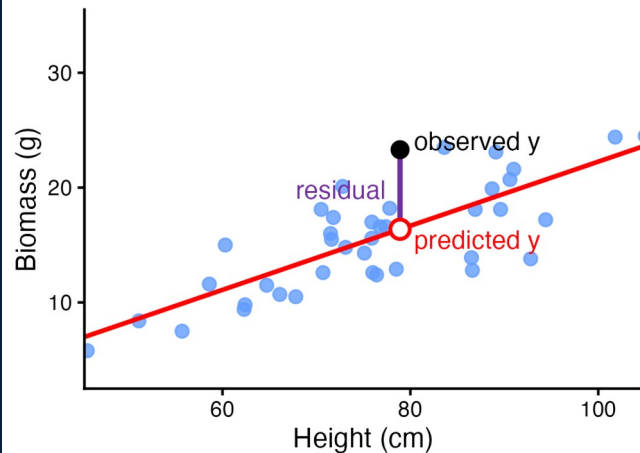
```
> predict(model, newdata = data.frame(height = 100))
      1
22.25981
> -5.65 + 0.279 * 100
[1] 22.25
> new <- data.frame(height = c(60, 80, 100))
> predict(model, newdata = new)
      1      2      3
11.09675 16.67828 22.25981
```



$$\hat{y} = -5.65 + 0.279 \times 100 = 22.3 \text{ g}$$

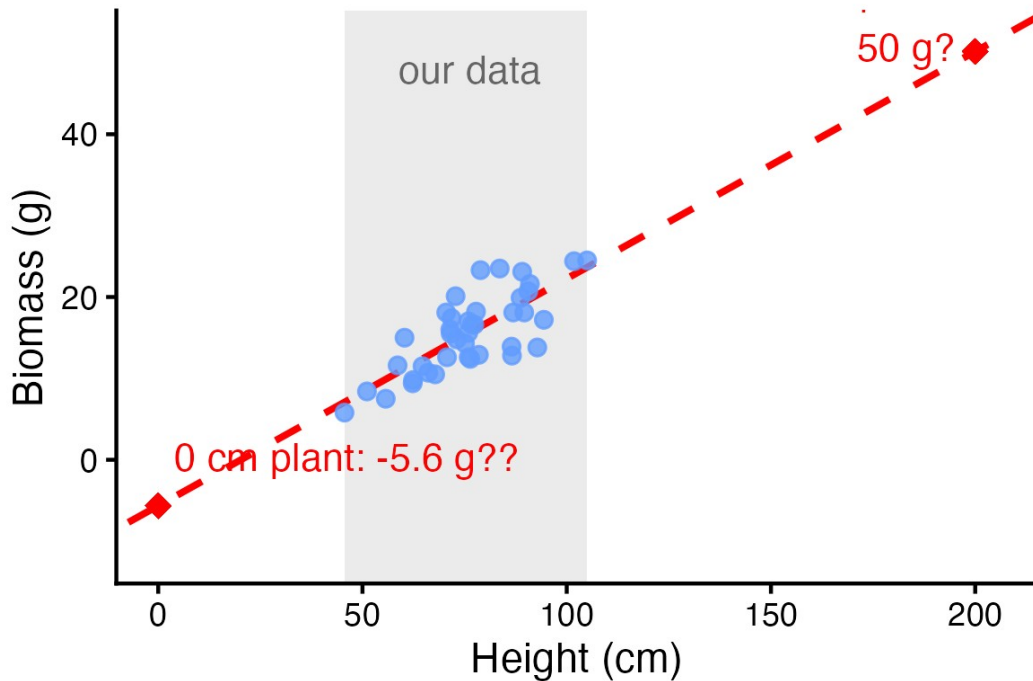
Observed = predicted + residual

```
> wheat$predicted <- predict(model) # fitted values
> wheat$residual <- resid(model)
> head(wheat[, c("height", "biomass", "predicted", "residual")])
  height biomass predicted residual
1   86.9   18.1  18.60391 -0.50390773
2   71.6   15.5  14.33404  1.16596148
3   73.1   14.8  14.75265  0.04734685
4   75.9   15.6  15.53407  0.06593288
5   91.0   21.6  19.74812  1.85187896
6   94.4   17.2  20.69698 -3.49698087
> # observed = predicted + residual
> head(wheat$predicted + wheat$residual)
[1] 18.1 15.5 14.8 15.6 21.6 17.2
```



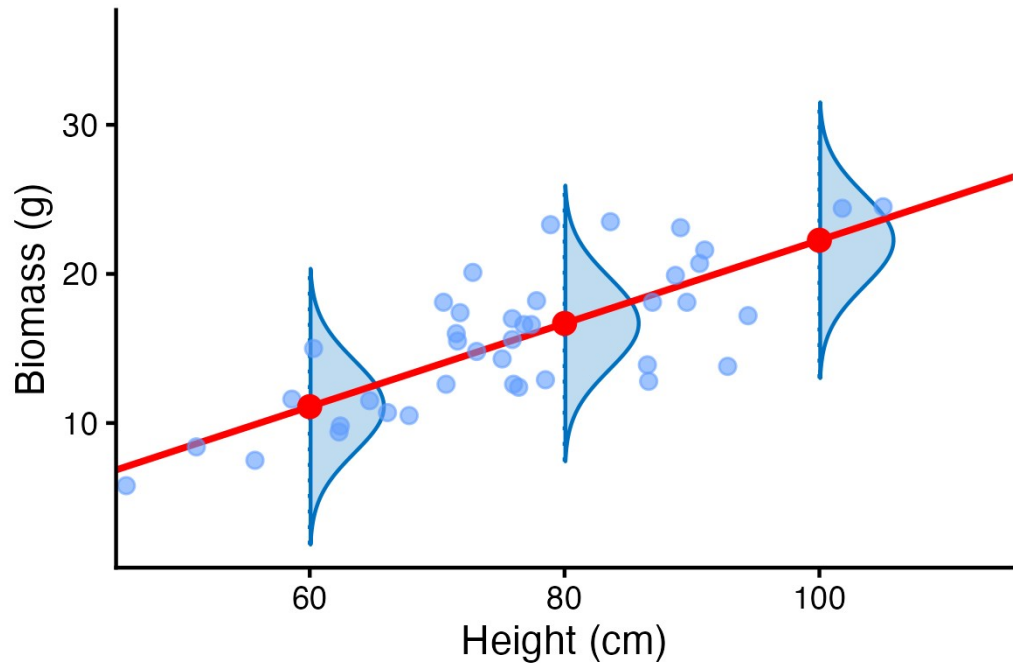
fitted (the line) + **residual** (what it misses) = the data

Interpolate, don't extrapolate



Inside the data: trust the line. Outside: **the biology may not follow.**

The line is a conditional mean



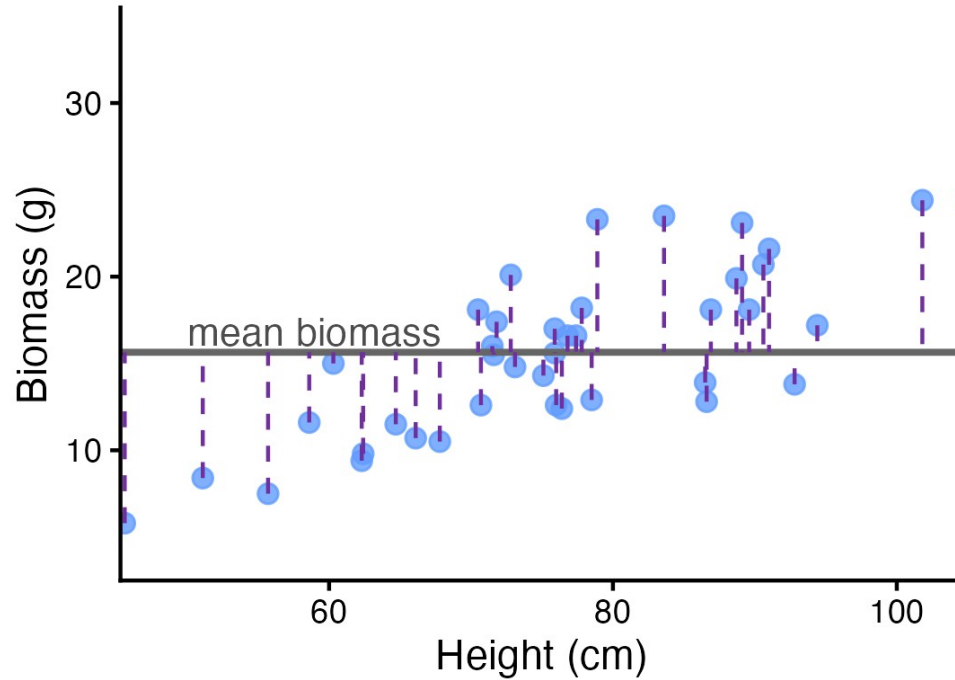
$$E[y | x] = \beta_0 + \beta_1 x \quad y | x \sim N(\beta_0 + \beta_1 x, \sigma^2)$$

At each height, biomass is a distribution. **The line is its mean.**

Part 2: How good is the line?

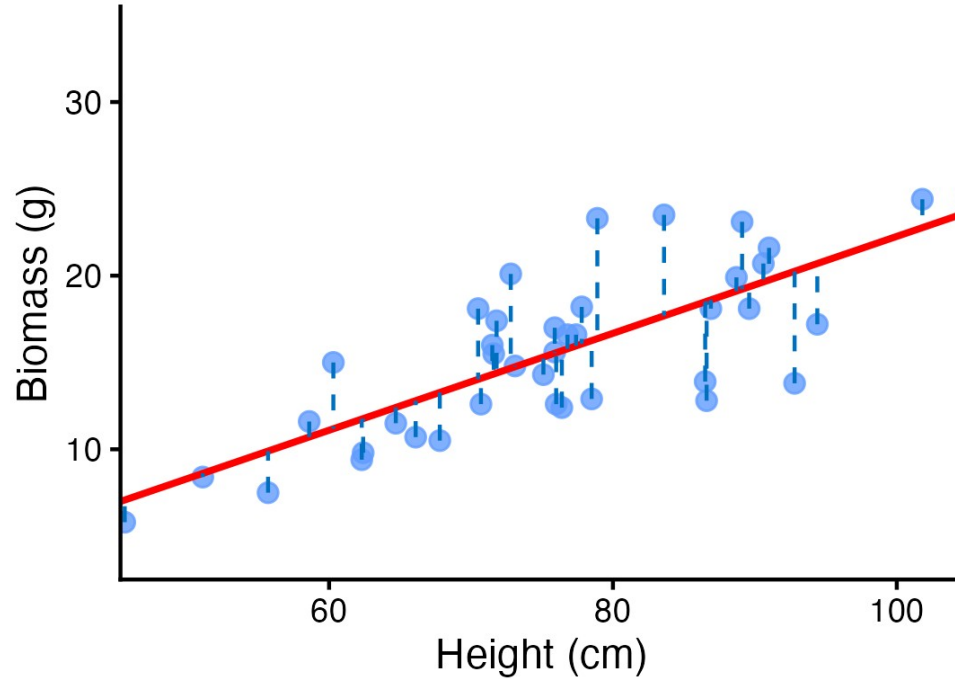
R^2 , t-tests, F-tests, confidence intervals

Variation around the mean



SS total: every plant's distance from the mean, squared and summed

Variation around the line



SS residual: every plant's distance from the line, squared and summed

R²: the share explained

$$SS_{\text{total}} = \sum_i (y_i - \bar{y})^2 \quad \text{spread around the mean}$$

$$SS_{\text{residual}} = \sum_i (y_i - \hat{y}_i)^2 \quad \text{spread around the line}$$

$$R^2 = \frac{SS_{\text{total}} - SS_{\text{residual}}}{SS_{\text{total}}} = 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}}$$

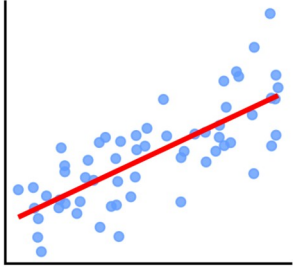
$$R^2 = \frac{891 - 361}{891} = 0.59$$

Height explains 59% of the variation in biomass

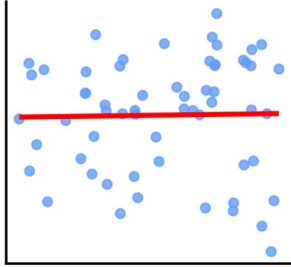
```
> ss_total <- sum((wheat$biomass - mean(wheat$biomass))^2)
> ss_resid <- sum(resid(model)^2)
> ss_total
[1] 891.299
> ss_resid
[1] 361.2284
> 1 - ss_resid / ss_total      # R2 by hand
[1] 0.594717
> summary(model)$r.squared
[1] 0.594717
> cor(wheat$height, wheat$biomass)^2  # = r2
[1] 0.594717
```

What R^2 looks like

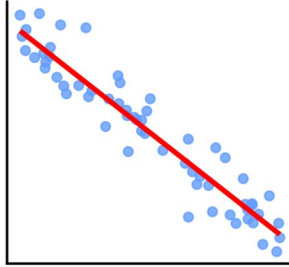
A: $R^2 = 0.58$



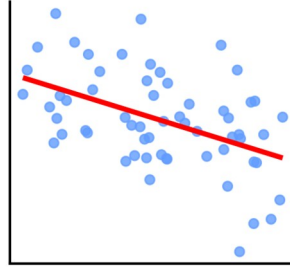
B: $R^2 = 0.00$



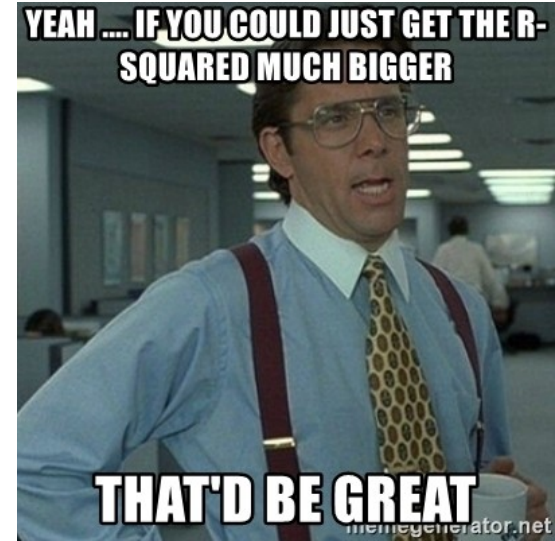
C: $R^2 = 0.91$



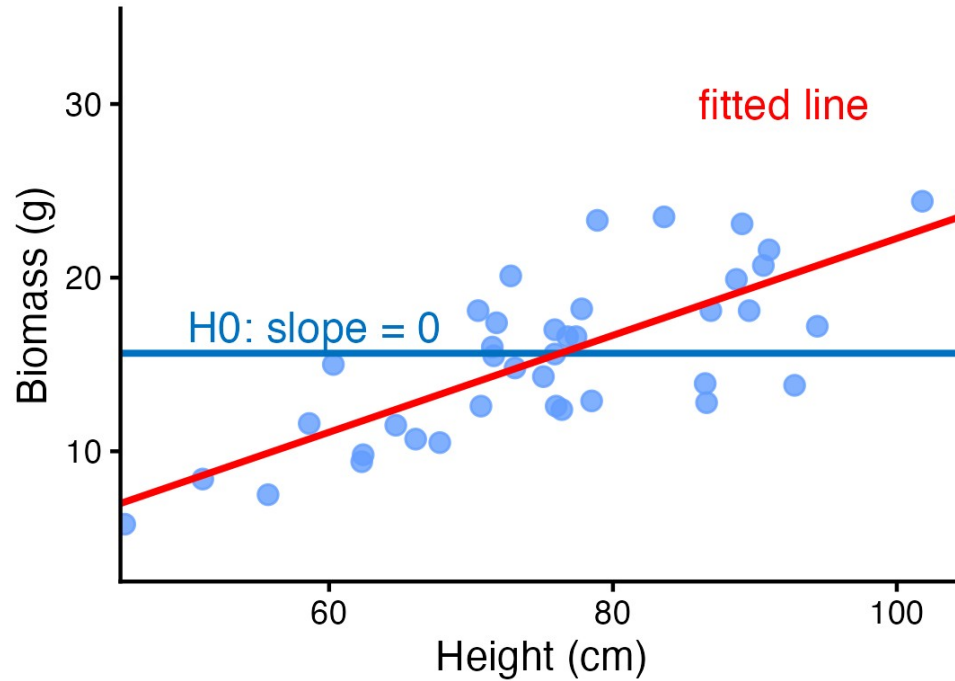
D: $R^2 = 0.22$



Tightness around the line: **not steepness, not direction**. R^2 is never negative.



Is the slope real?



H₀: the slope is zero: height tells us nothing, the best bet is the mean

summary(model)

```
> summary(model)
```

```
Call:
```

```
lm(formula = biomass ~ height, data = wheat)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-6.4505	-2.1590	0.0566	1.6519	6.9287

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5.64783	2.89282	-1.952	0.0583 .
height	0.27908	0.03737	7.467	5.79e-09 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 3.083 on 38 degrees of freedom
```

```
Multiple R-squared:  0.5947,    Adjusted R-squared:  0.5841
```

```
F-statistic: 55.76 on 1 and 38 DF,  p-value: 5.785e-09
```

The coefficient table

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-5.64783	2.89282	-1.952	0.0583	.
height	0.27908	0.03737	7.467	5.79e-09	***

estimate

its **standard error**

t = estimate / SE

p-value

The bottom lines

```
Residual standard error: 3.083 on 38 degrees of freedom  
Multiple R-squared:  0.5947,    Adjusted R-squared:  0.5841  
F-statistic: 55.76 on 1 and 38 DF,  p-value: 5.785e-09
```

$\sigma = 3.1$ g: the typical miss

$R^2 = 0.59$: the share explained

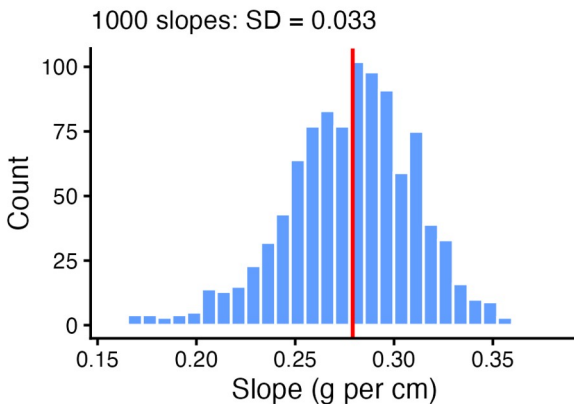
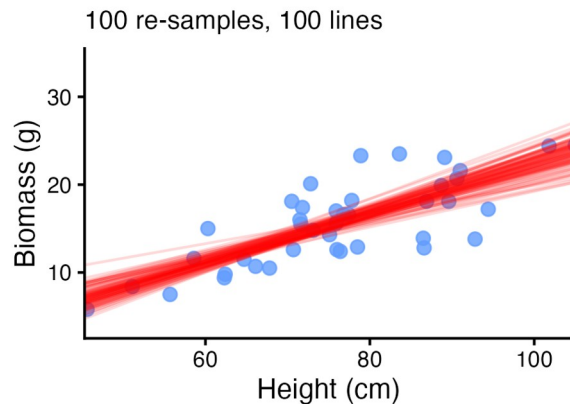
F : the whole model against the flat line

The slope's t-test

$$t = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)} \quad df = n - 2$$

$$t = \frac{0.279}{0.0374} = 7.47 \quad p = 5.8e - 09$$

The slope is **7.5 standard errors** from zero.



Confidence interval for the slope

$$\hat{\beta}_1 \pm t_{0.975, 38} \times SE = 0.279 \pm 2.02 \times 0.0374 = [0.20, 0.35]$$

```
> confint(model)
```

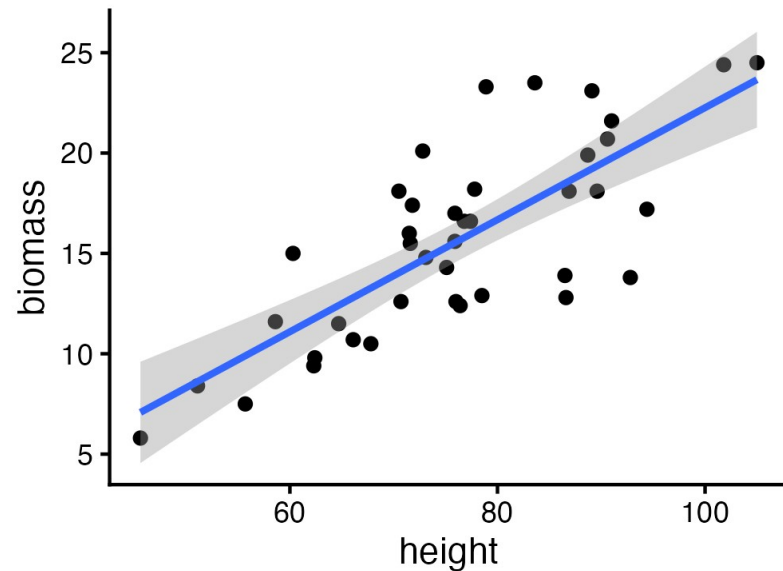
		2.5 %	97.5 %
(Intercept)	-11.5040436	0.2083775	
height	0.2034192	0.3547337	

```
> set.seed(206)
> slopes <- replicate(1000, {
+   i <- sample(40, replace = TRUE) # re-sample the plants
+   coef(lm(biomass ~ height, data = wheat[i, ])) [2]
+ })
> sd(slopes) # spread of the slope
[1] 0.03301875
> summary(model)$coefficients[2, "Std. Error"]
[1] 0.03737278
```

Report this: 0.28 g per cm, 95% CI 0.20 to 0.35

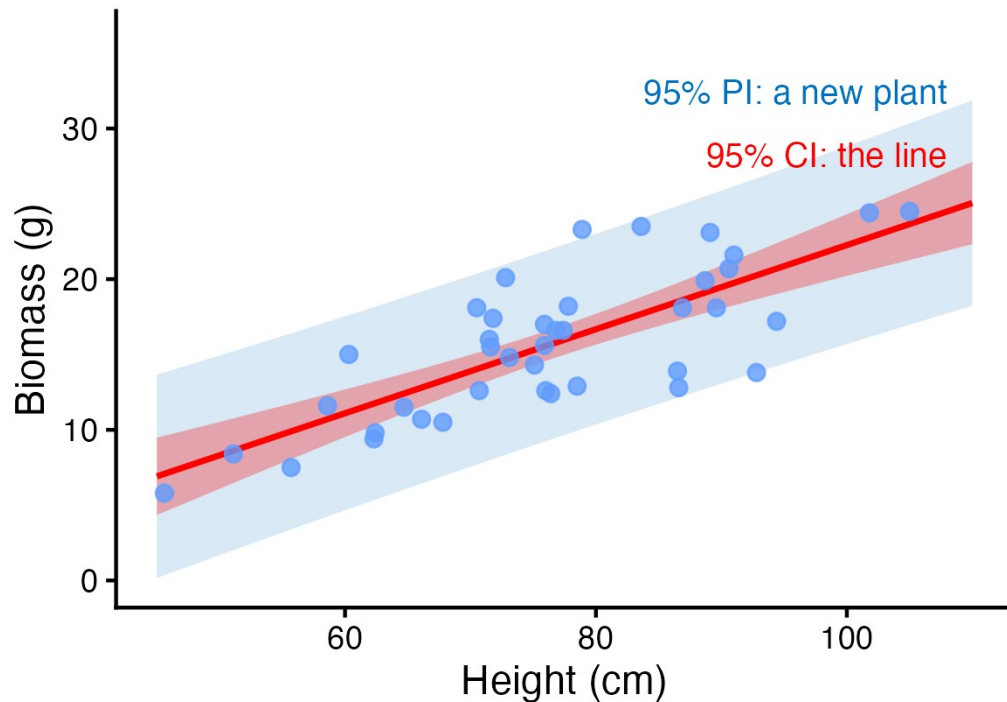
The band: `geom_smooth(method = "lm")`

```
> library(ggplot2)
> ggplot(wheat, aes(x = height, y = biomass)) +
+   geom_point() +
+   geom_smooth(method = "lm") +
+   theme_classic()
```



The band is the 95% CI **for the line**, not for where new plants fall

Two intervals: the line, or a new plant?



```
> new <- data.frame(height = c(60, 80, 100))
> predict(model, newdata = new, interval = "confidence")
      fit      lwr      upr
1 11.09675  9.517423 12.67608
2 16.67828 15.652415 17.70415
3 22.25981 20.212925 24.30669
> predict(model, newdata = new, interval = "prediction")
      fit      lwr      upr
1 11.09675  4.658465 17.53504
2 16.67828 10.352962 23.00360
3 22.25981 15.691172 28.82845
```

CI: where the mean biomass of 100 cm plants is. **PI:** where one new 100 cm plant will land.

The F-test: the whole model

$$F = \frac{R^2/k}{(1 - R^2)/(n - k - 1)} \quad df_1 = k, \quad df_2 = n - k - 1$$

$$F = \frac{0.595/1}{0.405/38} = 55.8 \quad = t^2 = 7.47^2$$

```
> r2 <- summary(model)$r.squared
> k <- 1 # predictors
> n <- nrow(wheat) # plants
> f <- (r2 / k) / ((1 - r2) / (n - k - 1))
> f
[1] 55.76163
> pf(f, k, n - k - 1, lower.tail = FALSE) # p-value
[1] 5.785277e-09
> summary(model)$fstatistic
  value  numdf  dendf
55.76163 1.00000 38.00000
> 7.467^2 # one predictor: F = t^2
[1] 55.75609
```

H_0 : all slopes are zero. One predictor: **F = t²**

The intercept's p-value

```
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) -5.64783    2.89282  -1.952   0.0583 .
```

It tests whether a 0 cm plant has 0 g of biomass.

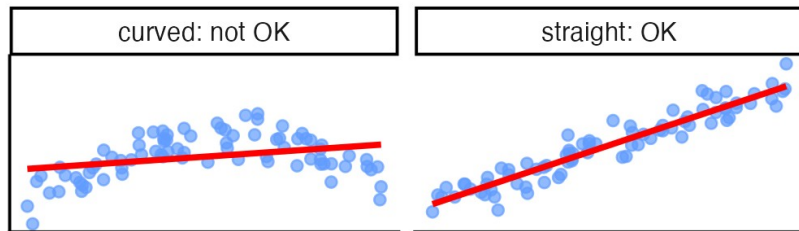
Usually not your question. Keep the intercept; ignore its p-value.

Part 3: What can go wrong

Assumptions and diagnostic plots

Assumptions: L • I • N • E

L Linearity

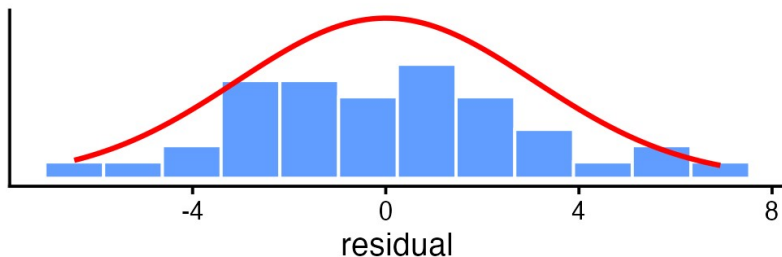


I Independence

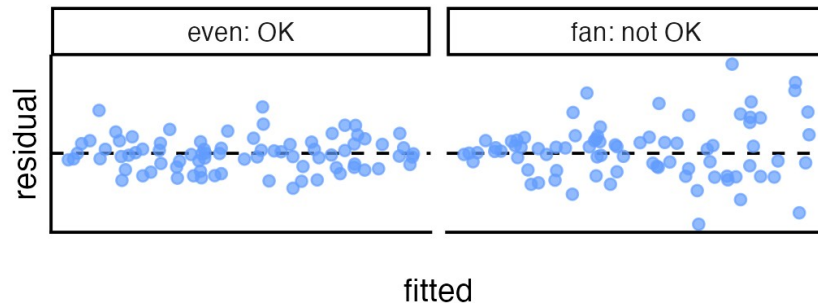
10 tillers x 8 pots = 8 replicates, not 80



N Normal residuals



E Equal variance



About the **residuals**, never the raw y. “In practice these are rarely met perfectly.”

plot(model): four diagnostic plots

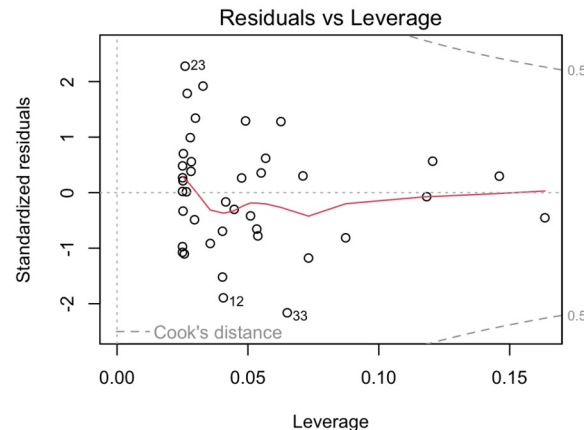
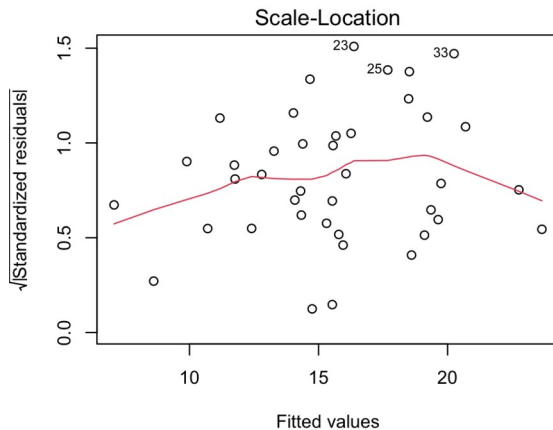
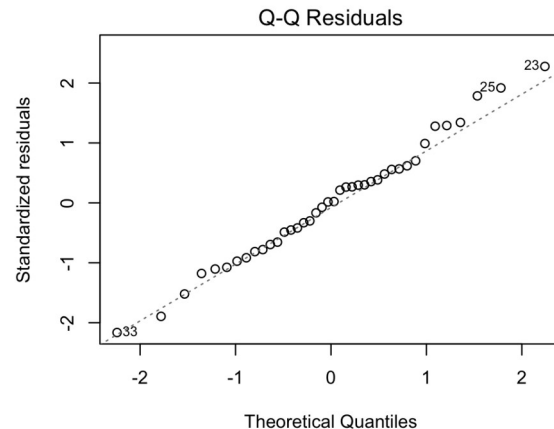
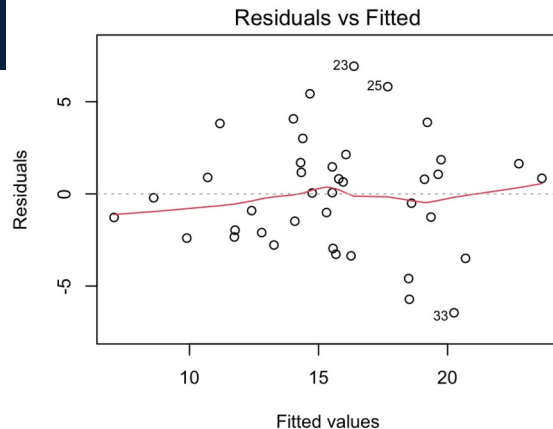
```
> par(mfrow = c(2, 2)) # four panels in one window  
> plot(model)
```

1 linear?

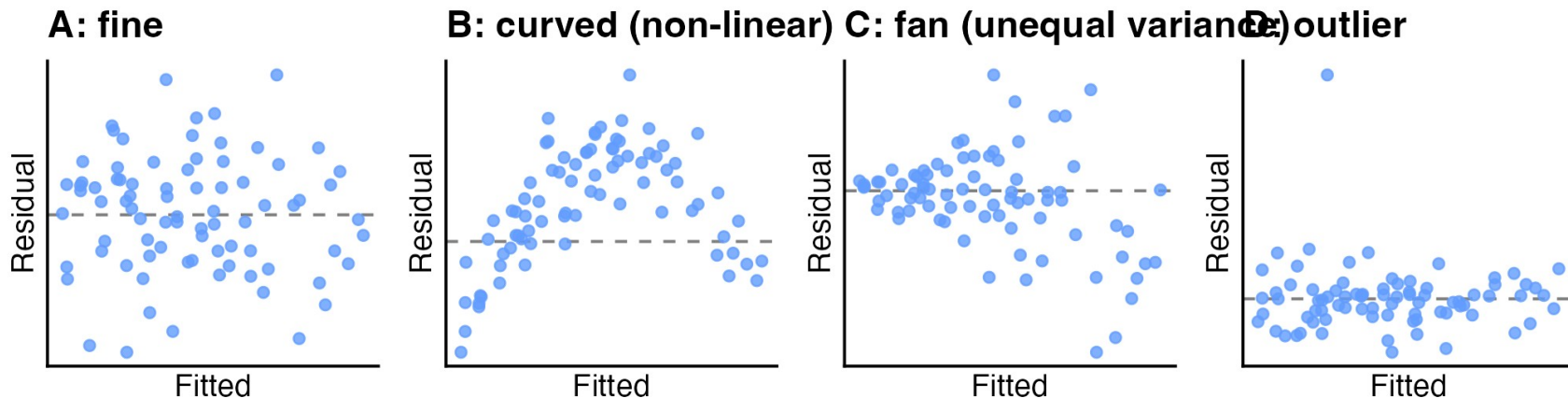
2 normal?

3 equal variance?

4 any plant pulling the line?



Four patterns, four fixes



A nothing to do

B add x^2

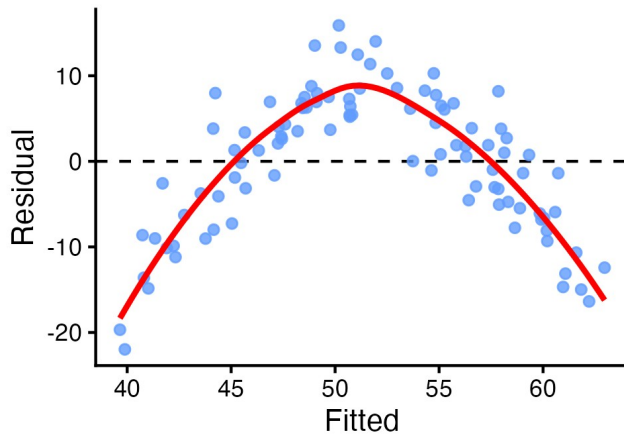
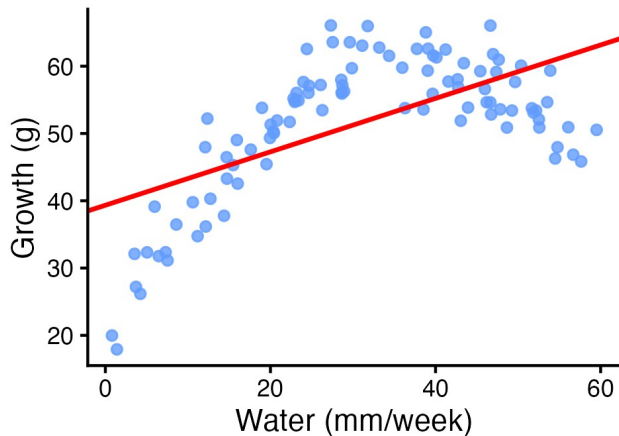
C $\log(y)$

D check the data

Violation 1: a curve

```
> set.seed(1)
> water <- runif(100, 0, 60) # mm per week
> growth <- 20 + 2.2 * water - 0.03 * water^2 +
+   rnorm(100, sd = 4)
> pots <- data.frame(water, growth)
> model_w <- lm(growth ~ water, data = pots)
> summary(model_w)$r.squared
[1] 0.38397
```

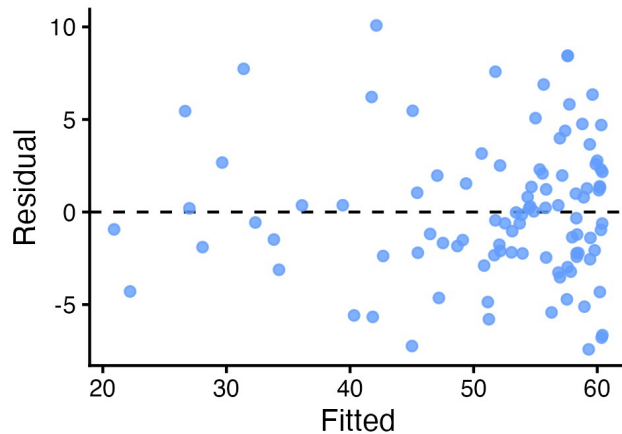
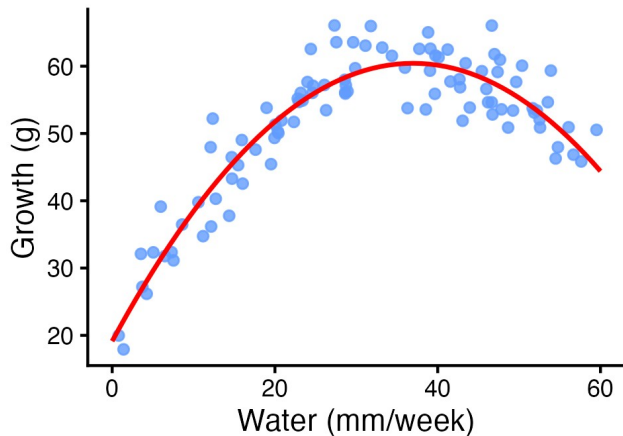
Too much water drowns the plant: the residuals **curve**.



Fix: add a squared term

```
> model_w2 <- lm(growth ~ water + I(water^2), data = pots)
> round(coef(model_w2), 3)
(Intercept)      water  I(water^2)
      19.137      2.234      -0.030
> summary(model_w2)$r.squared
[1] 0.8674312
```

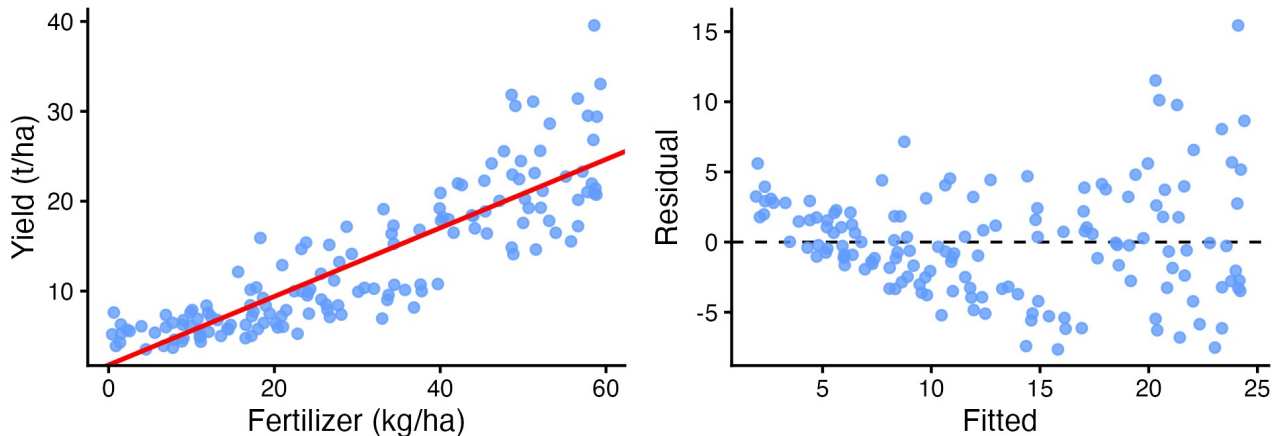
R^2 **0.38** $\rightarrow$ **0.87**. Still a linear model: linear in the β s.



Violation 2: a fan

```
> set.seed(2)
> dose <- runif(150, 0, 60) # fertilizer, kg/ha
> yield <- exp(1.5 + 0.03 * dose + rnorm(150, sd = 0.25))
> plots <- data.frame(dose, yield)
> model_f <- lm(yield ~ dose, data = plots)
```

The residuals **fan out**: every SE and p-value is wrong.

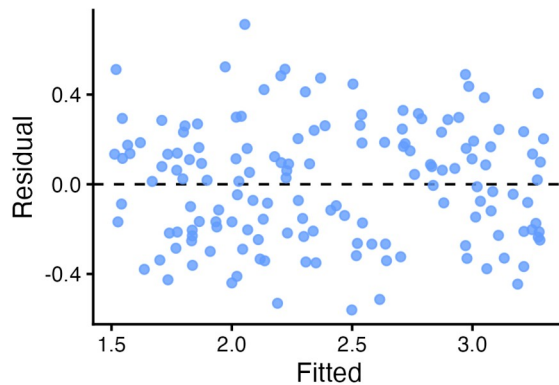
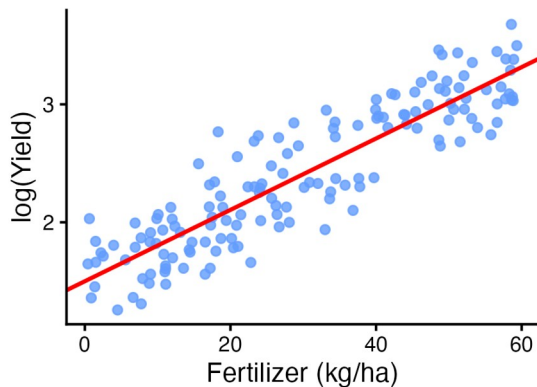


Fix: log the response

```
> model_log <- lm(log(yield) ~ dose, data = plots)
> round(coef(model_log), 3)
(Intercept)      dose
         1.50         0.03
> exp(0.030)      # x 1.03 per kg/ha: +3%
[1] 1.030455
```

The slope is now a **percent change**: ×
1.03 per kg/ha

$$\log(\hat{y}) = a + 0.030x \quad \Rightarrow \quad \hat{y} = e^a \times e^{0.030x} \quad \Rightarrow \quad \times 1.03 \text{ per kg/ha}$$



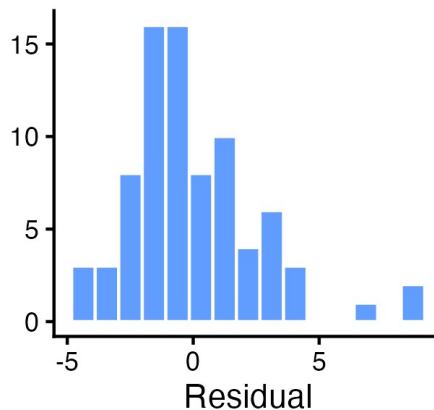
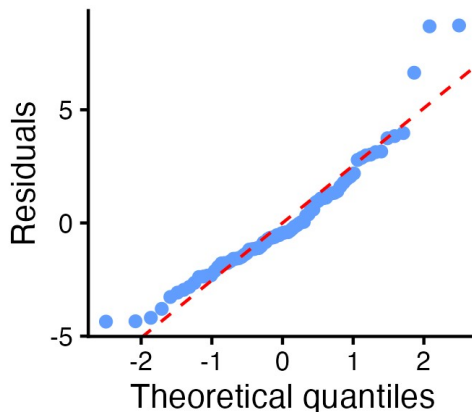
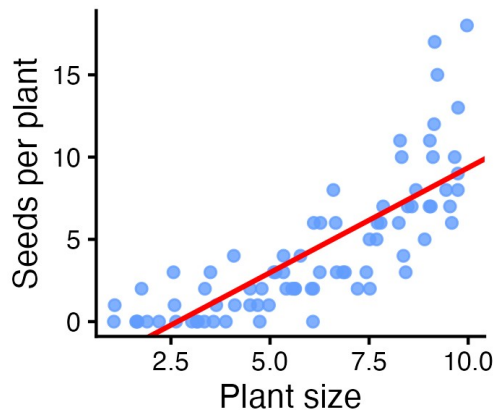
Violation 3: residuals that are not normal

```
> set.seed(4)
> size <- runif(80, 1, 10) # plant size
> # seeds per plant: a count, many zeros
> seeds <- rpois(80, lambda = exp(-1 + 0.35 * size))
> pods <- data.frame(size, seeds)
> model_s <- lm(seeds ~ size, data = pods)
> shapiro.test(resid(model_s))
```

Shapiro-Wilk normality test

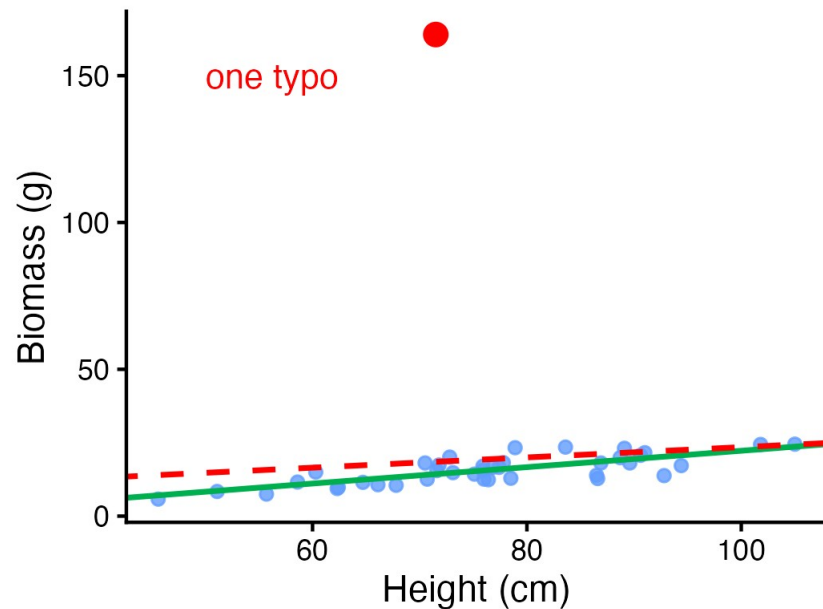
```
data: resid(model_s)
W = 0.92704, p-value = 0.0002078
```

Counts: skewed residuals, a bent Q-Q plot, Shapiro $p < 0.001$. **Next week: a Poisson GLM.**



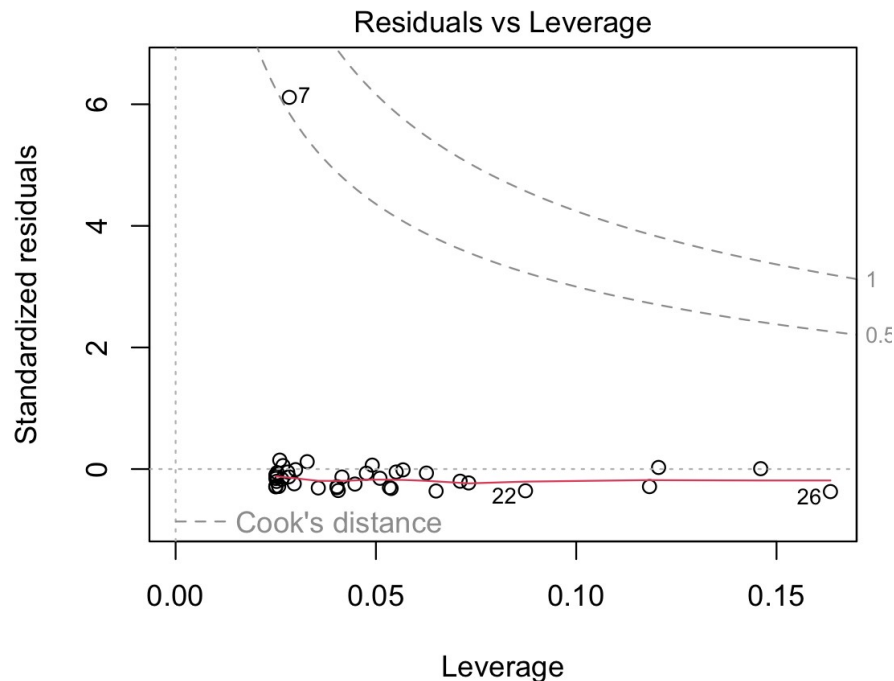
Violation 4: one bad point

```
> typo <- wheat
> typo$biomass[7] <- 164 # 16.4 typed as 164
> model_t <- lm(biomass ~ height, data = typo)
> coef(model) # the real slope
(Intercept) height
-5.6478330 0.2790764
> coef(model_t) # with the typo
(Intercept) height
6.0119191 0.1747512
> which.max(cooks.distance(model_t))
7
7
> max(cooks.distance(model_t))
[1] 0.5460262
```



One typo (16.4 → 164) moves the slope from 0.28 to 0.17. **Check the data before the statistics.**

Finding it: residuals vs leverage

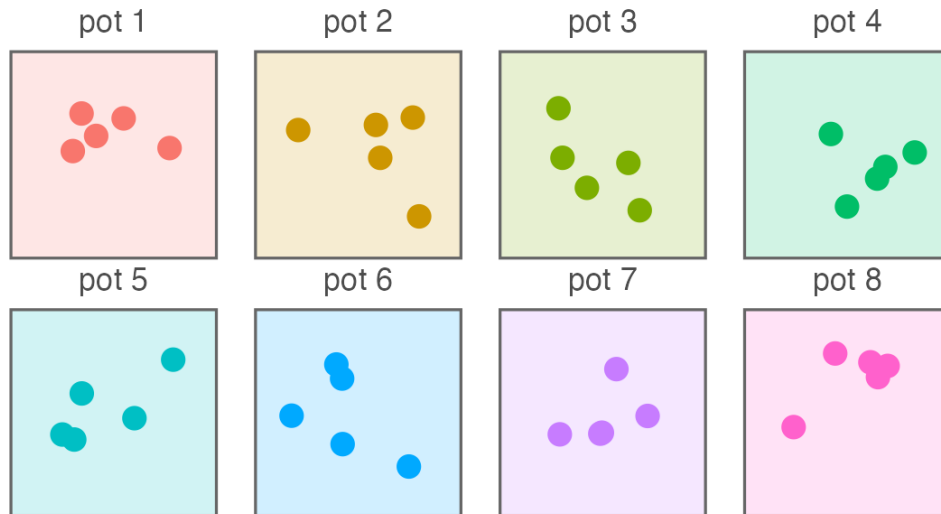


Leverage: how unusual the x value is

Residual: how far from the line

Cook's distance: both at once; past the dashed 0.5 and 1 contours, one point is steering the model

Violation 5: not independent

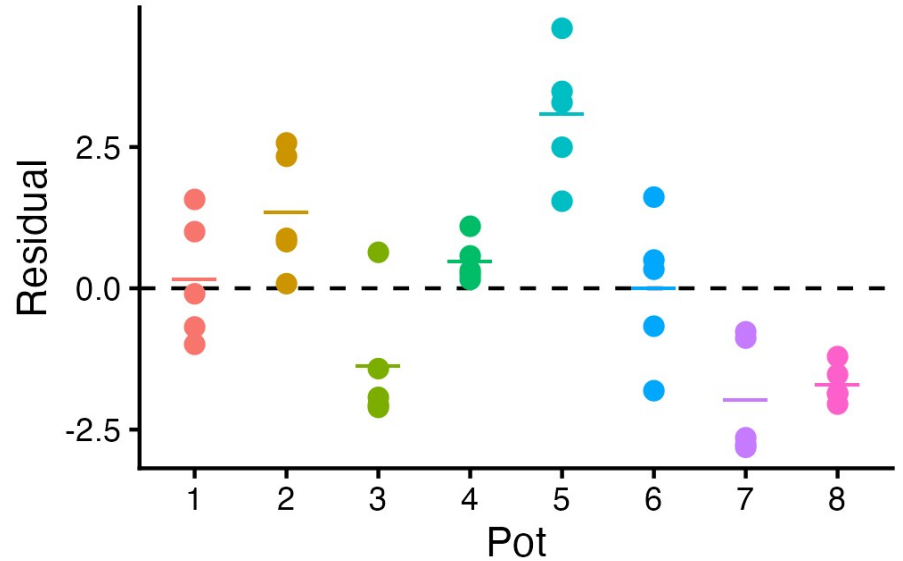
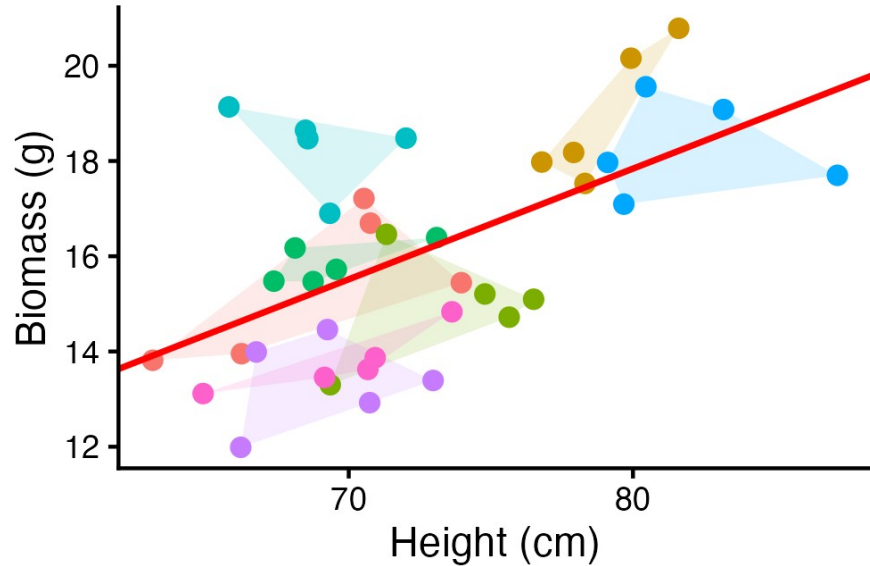


8 pots, 5 tillers each: 40 measurements.

Tillers in one pot share its soil, water, and light. **They are not 40 independent plants.**

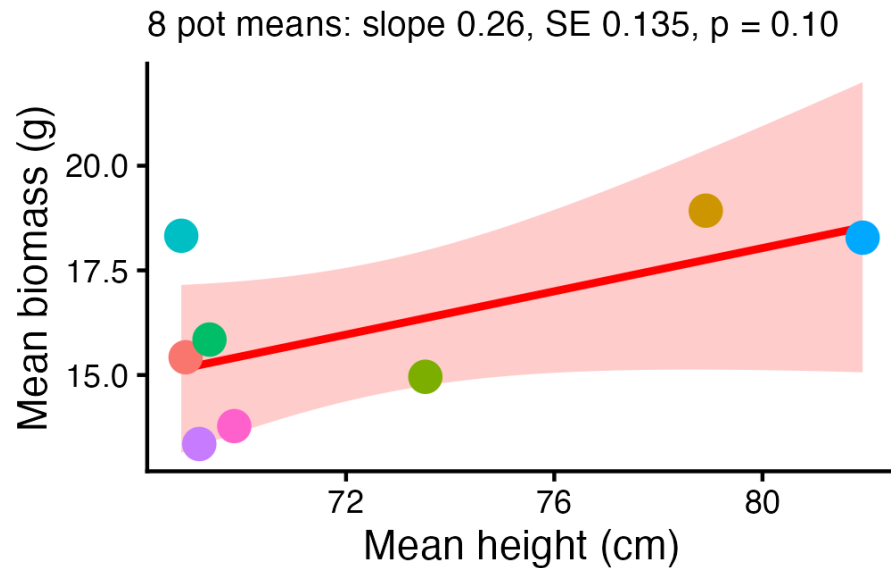
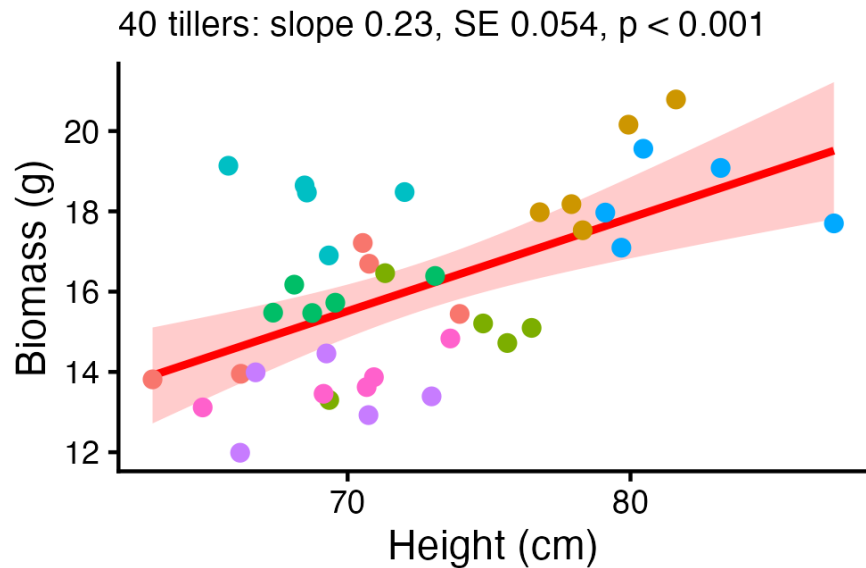
Independence is about the **observations**, not about x and y: does one tiller's residual tell you about its pot-mates'?

The signature: residuals that come in clumps



Left: each pot's tillers sit together. Right: residuals grouped by pot: **whole pots sit above or below the line.**

The consequence: an SE that is too small



Same slope. Pretending $n = 40$ makes the SE **2.5× too small** and turns $p = 0.10$ into $p < 0.001$.

Missing data: lm() drops rows silently

```
> wheat_na <- wheat
> # three plants were not measured
> wheat_na$nitrogen[c(3, 11, 25)] <- NA
> colSums(is.na(wheat_na))
  plant      water  nitrogen    height    biomass predicted
    0         0         3         0         0         0
residual    noise
    0         0

> model_na <- lm(biomass ~ height + nitrogen, data = wheat_na)
> nobs(model_na)                # rows actually used
[1] 37
> length(resid(model_na))       # shorter than the data frame!
[1] 37
> sum(complete.cases(wheat_na))
[1] 37
```

```
> model_na <- lm(biomass ~ height + nitrogen, wheat_na,
+   na.action = na.exclude)
> # padded with NA: lines up with the data
> length(resid(model_na))
[1] 40
> wheat_na$residual <- resid(model_na)
```

37 plants used, 40 residuals expected: **na.action = na.exclude** keeps them lined up. Always report n.

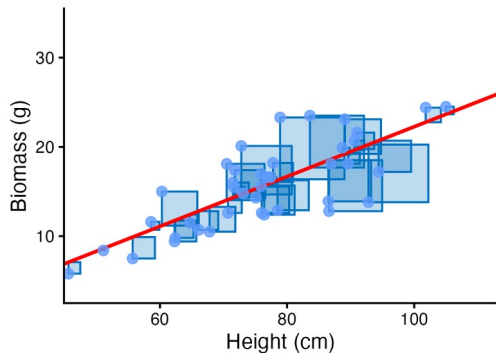
**I DON'T ALWAYS MAKE A LINEAR
REGRESSION MODEL**



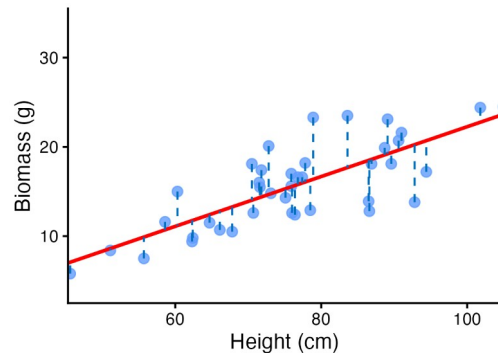
**BUT WHEN I DO, I CHECK IT WITH A SCATTER
PLOT & RESIDUAL PLOT**

So far, in four pictures

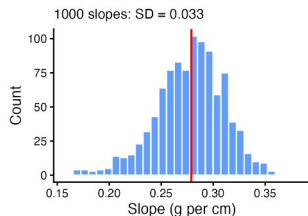
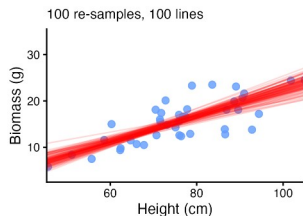
Least
squares



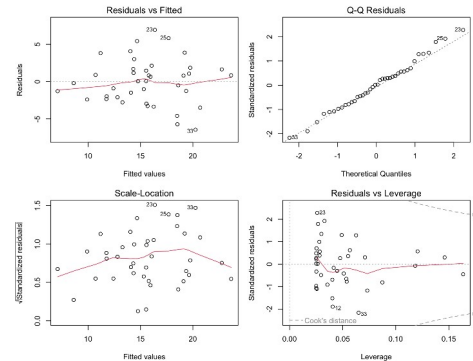
R^2



SE, t, CI



plot(model)



Next: more than one predictor, and the three ways a second variable can fool you

Part 4: More than one predictor

Multiple linear regression

Recap so far

Kahoot-style warm-up

13 quick questions on what we've covered so far.

Not graded.

Use a pseudonym if you want.

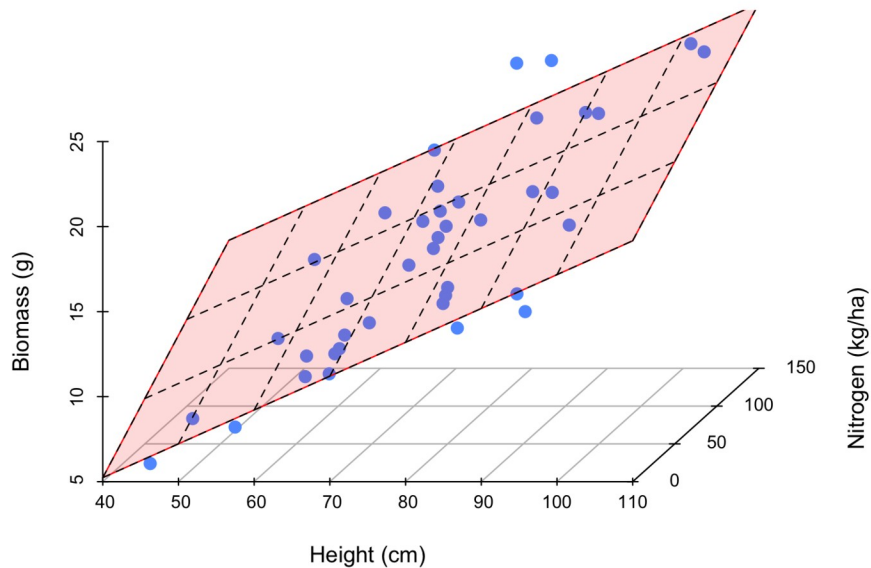
Fast and correct wins.

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w2-monday-recap](https://greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w2-monday-recap)



Scan with your phone

From a line to a plane



$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$

Two predictors: a plane. Same rule:
least squares.

lm() with two predictors

```
> model2 <- lm(biomass ~ height + nitrogen, data = wheat)
> summary(model2)

Call:
lm(formula = biomass ~ height + nitrogen, data = wheat)

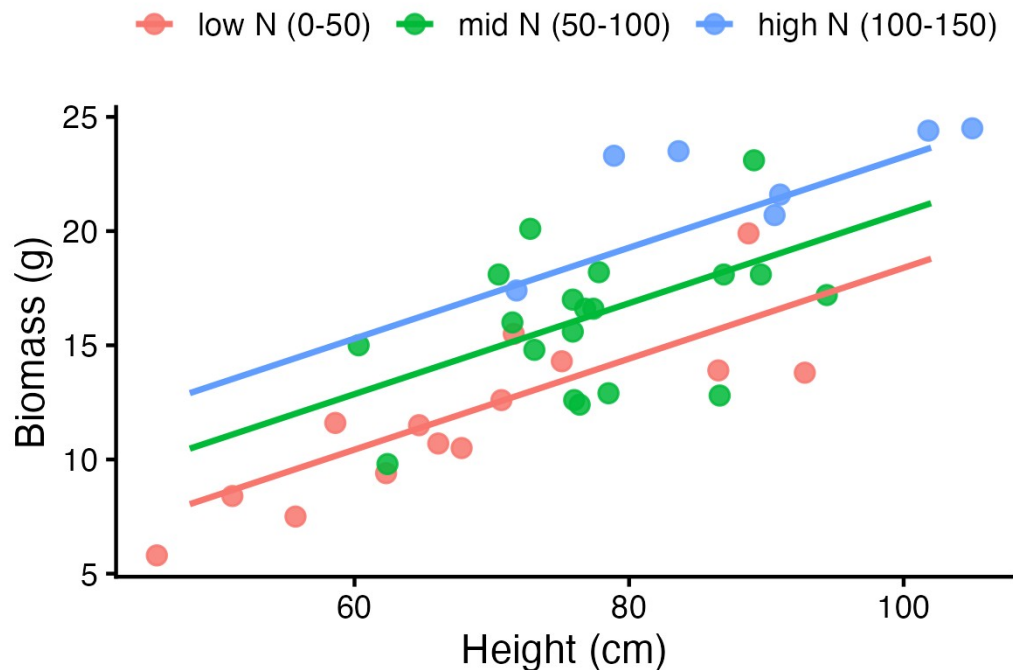
Residuals:
    Min       1Q   Median       3Q      Max
-5.2589 -1.3734 -0.1355  1.2666  4.4950

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.72580    2.45074  -1.112   0.273
height       0.19902    0.03538   5.625 2.02e-06 ***
nitrogen     0.04862    0.01087   4.472 7.11e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.517 on 37 degrees of freedom
Multiple R-squared:  0.7369,    Adjusted R-squared:  0.7227
F-statistic: 51.82 on 2 and 37 DF,  p-value: 1.87e-11
```

$$\widehat{\text{biomass}} = -2.73 + 0.199 \text{ height} + 0.0486 \text{ nitrogen}$$

“Holding the other fixed”

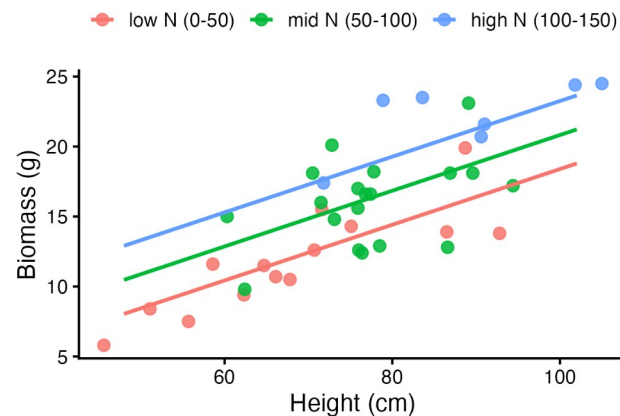


$\beta_1 = +0.20$ g per cm, among plants with the same nitrogen

$\beta_2 = +0.049$ g per kg/ha, among plants of the same height

Why the height slope changed

```
> coef(model)      # biomass ~ height
(Intercept)      height
-5.6478330      0.2790764
> coef(model2)     # biomass ~ height + nitrogen
(Intercept)      height      nitrogen
-2.72579601      0.19902030      0.04862335
> cor(wheat$height, wheat$nitrogen)
[1] 0.5060092
```



0.28 → 0.20: height alone was taking credit for some of nitrogen's work

R² only goes up: use adjusted R²

```
> set.seed(206)
> wheat$noise <- rnorm(40) # a predictor that is pure noise
> r2 <- function(m) summary(m)$r.squared
> r2(lm(biomass ~ height, wheat))
[1] 0.594717
> r2(lm(biomass ~ height + nitrogen, wheat))
[1] 0.7369115
> r2(lm(biomass ~ height + nitrogen + noise, wheat))
[1] 0.7489497
> adj <- function(m) summary(m)$adj.r.squared
> adj(lm(biomass ~ height + nitrogen + noise, wheat))
[1] 0.7280289
```

$$R_{\text{adj}}^2 = 1 - (1 - R^2) \frac{n - 1}{n - k - 1}$$

A **pure-noise** predictor still raised R². Adjusted R² charges for it.

Does nitrogen earn its place?

```
> anova(model, model2) # does nitrogen improve the fit?
```

Analysis of Variance Table

Model 1: biomass ~ height

Model 2: biomass ~ height + nitrogen

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	38	361.23				
2	37	234.49	1	126.74	19.998	7.112e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
> coef(summary(model2))["nitrogen", "t value"]^2 # = F
```

```
[1] 19.99782
```

F = 20.0, p < 0.001

= t^2 for nitrogen: one term added, same test

Nested F-test: compare a model with and without a **set** of terms. One term: it is the t-test.

anova(model2): the order matters

```
> anova(model2) # sequential: height, then nitrogen
Analysis of Variance Table

Response: biomass
      Df Sum Sq Mean Sq F value    Pr(>F)
height  1  530.07   530.07  83.639 4.941e-11 ***
nitrogen 1  126.74   126.74  19.998 7.112e-05 ***
Residuals 37  234.49     6.34
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> # the other order
> anova(lm(biomass ~ nitrogen + height, wheat))
Analysis of Variance Table

Response: biomass
      Df Sum Sq Mean Sq F value    Pr(>F)
nitrogen 1  456.26   456.26  71.992 3.306e-10 ***
height  1  200.55   200.55  31.645 2.020e-06 ***
Residuals 37  234.49     6.34
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> # each term given the others (= the t-tests)
> drop1(model2, test = "F")
Single term deletions

Model:
biomass ~ height + nitrogen
      Df Sum of Sq    RSS   AIC F value    Pr(>F)
<none>                 234.49  76.741
height  1     200.55  435.04  99.463   31.645 2.020e-06 ***
nitrogen 1     126.74  361.23  92.025   19.998 7.112e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

anova(): sequential, each term given the ones before it.

drop1() / summary(): each term given all the others.

Height's sum of squares changes with the order; nitrogen's does not when it is last. **Report drop1() or summary().**

F-test with $k = 2$

$$F = \frac{R^2/k}{(1 - R^2)/(n - k - 1)} \quad df_1 = k, \quad df_2 = n - k - 1$$

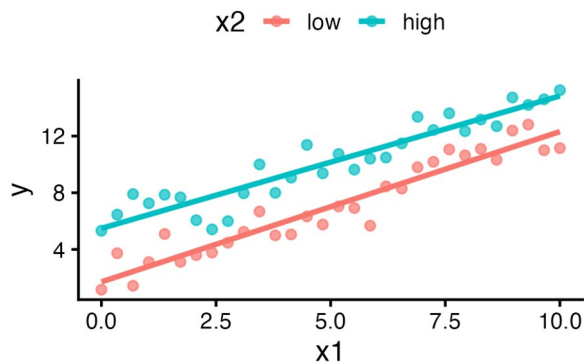
$$F = \frac{0.737/2}{0.263/37} = 51.8$$

```
> r2 <- summary(model2)$r.squared
> k <- 2; n <- 40
> f <- (r2 / k) / ((1 - r2) / (n - k - 1))
> f
[1] 51.81853
> pf(f, k, n - k - 1, lower.tail = FALSE)
[1] 1.870191e-11
```

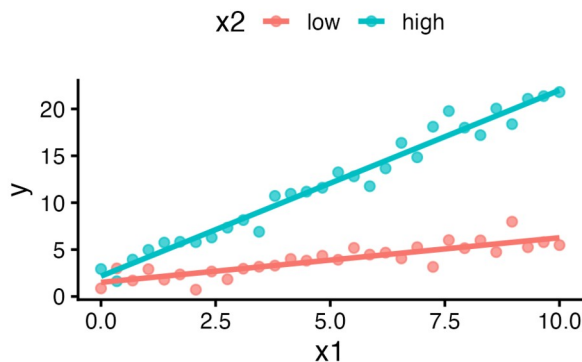
H_0 : both slopes are zero. One F for the model,
one t per predictor.

Interactions

No interaction: same slope



Interaction: slope depends on x2



$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \varepsilon$$

$$\text{slope of } x_1 = \beta_1 + \beta_3 x_2$$

```
> model_int <- lm(biomass ~ height * nitrogen, data = wheat)
> round(coef(summary(model_int)), 4)
```

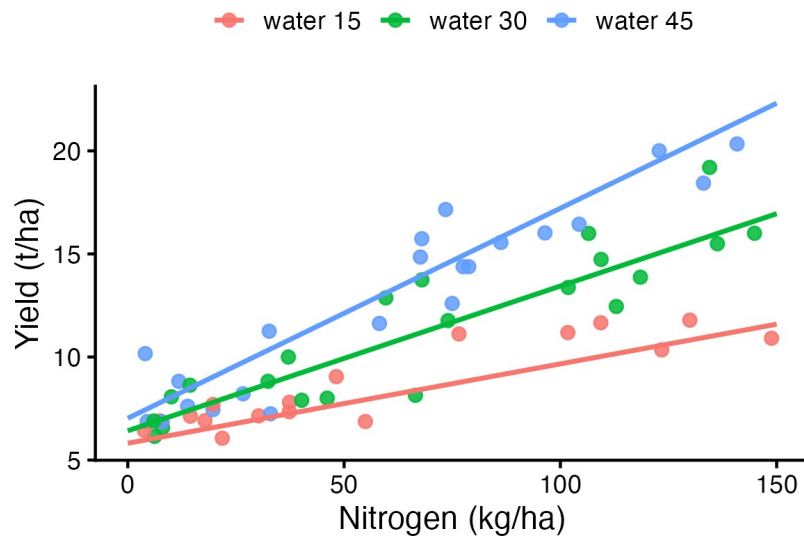
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.9686	3.8008	-0.7810	0.4399
height	0.2022	0.0523	3.8688	0.0004
nitrogen	0.0532	0.0549	0.9679	0.3395
height:nitrogen	-0.0001	0.0007	-0.0844	0.9332

$y \sim x_1 * x_2$: the slope of x_1 depends on x_2 . Here $p = 0.93$:
none.

A real interaction: nitrogen × water

```
> set.seed(18)
> trial <- data.frame(nitrogen = runif(60, 0, 150),
+                     water     = runif(60, 10, 50))
> trial$yield <- with(trial, 5 + 0.01 * nitrogen + 0.05 * water
+                     + 0.002 * nitrogen * water + rnorm(60, sd = 1.2))
> model_nw <- lm(yield ~ nitrogen * water, data = trial)
> round(coef(summary(model_nw)), 4)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.2123	0.6944	7.5061	0.0000
nitrogen	0.0068	0.0091	0.7506	0.4560
water	0.0403	0.0211	1.9144	0.0607
nitrogen:water	0.0021	0.0003	7.5646	0.0000



Nitrogen adds 0.038 t/ha per kg at water 15, **0.102 at water 45**: the lines fan out.

Interactions and centring

```
> library(car)
> vif(model_nw)      # the product term overlaps its parts
      nitrogen      water nitrogen:water
      8.978014      2.850287      10.761527
> trial$nitrogen_c <- trial$nitrogen - mean(trial$nitrogen)
> trial$water_c     <- trial$water - mean(trial$water)
> model_nwc <- lm(yield ~ nitrogen_c * water_c, data = trial)
> vif(model_nwc)     # centred: the overlap is gone
      nitrogen_c      water_c nitrogen_c:water_c
      1.013733      1.001760      1.015259
> # same interaction; main effects now at the mean
> round(coef(model_nwc), 4)
      (Intercept)      nitrogen_c      water_c
      11.1270      0.0741      0.1725
nitrogen_c:water_c
      0.0021
```

The product term overlaps its parts: VIF up to 11.

Centre both predictors: VIF ~ 1, the same interaction, and main effects that mean something (the slope at average water).

Why bother? Hidden relationships

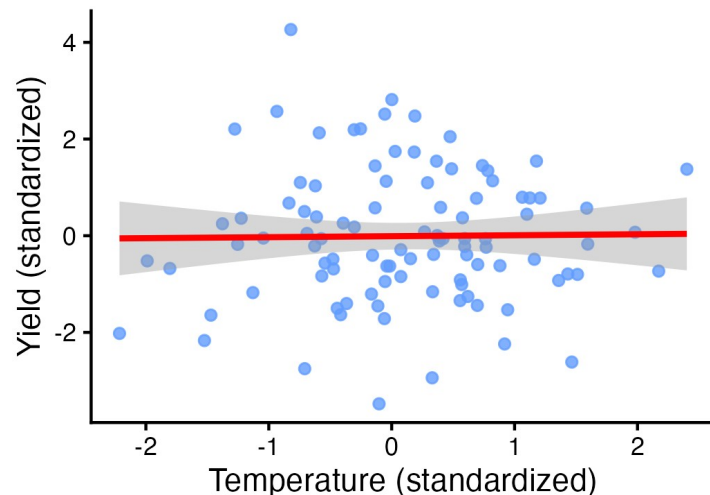
What a second variable can reveal, or fake

Case 1: does temperature affect yield?

```
> set.seed(1)
> temperature <- rnorm(100)
> insects      <- rnorm(100) - temperature # heat: fewer
> yield        <- temperature + insects + rnorm(100)
```

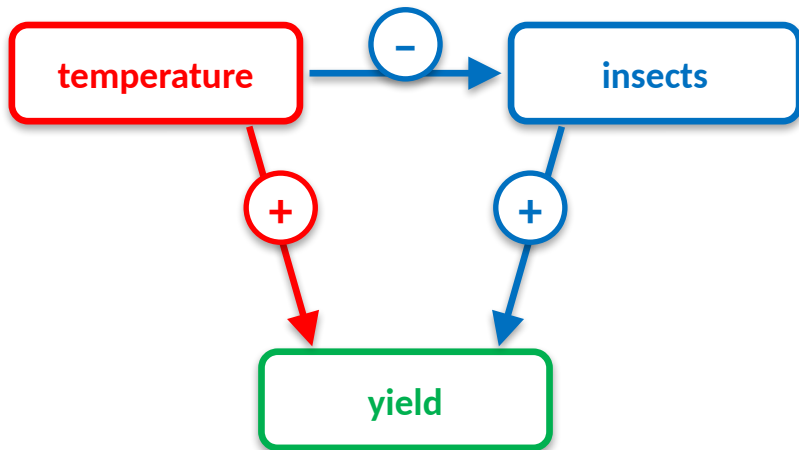
```
> round(coef(summary(lm(yield ~ temperature))), 3)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.01	0.139	-0.074	0.941
temperature	0.02	0.155	0.130	0.897



Flat line, $p = 0.9$. **Hands up: does temperature matter?**

Reveal: two paths that cancel



```
> round(coef(summary(lm(yield ~ temperature + insects))), 3)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.025	0.105	0.241	0.81
temperature	0.968	0.160	6.043	0.00
insects	0.947	0.109	8.646	0.00

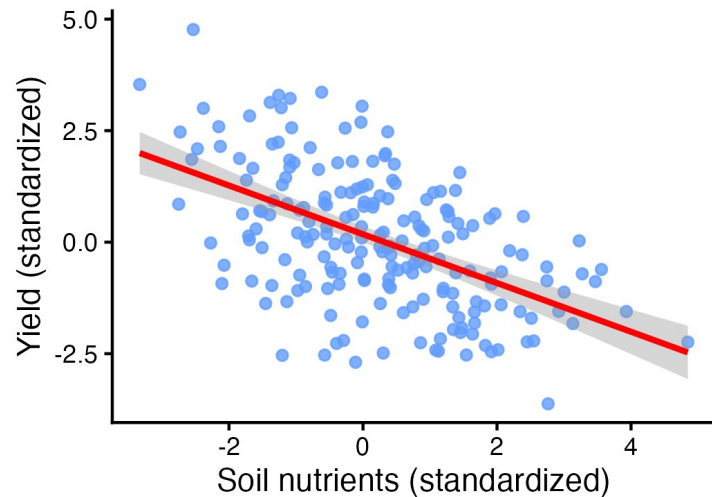
```
insects <- rnorm(100) - temperature  
yield <- temperature + insects + rnorm(100)
```

With insects in the model: temperature **+1, $p < 0.001$** . A suppressor.

Case 2: do nutrients reduce yield?

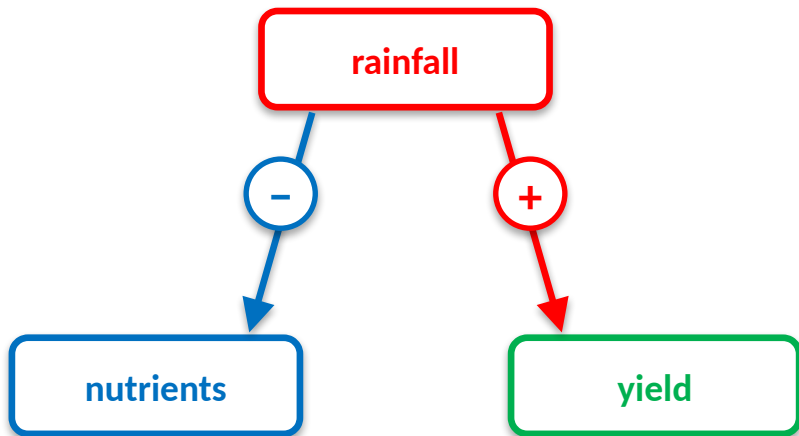
```
> set.seed(2)
> rainfall <- rnorm(200)
> nutrients <- -rainfall + rnorm(200) # rain leaches them
> yield <- rainfall + rnorm(200) # nutrients: no effect
```

```
> round(coef(summary(lm(yield ~ nutrients))), 3)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.177      0.093     1.915   0.057
nutrients    -0.546      0.063    -8.729   0.000
```



Strongly negative, $p < 0.001$. **Hands up: do nutrients hurt yield?**

Reveal: a confounder



```
nutrients <- -rainfall + rnorm(200)
yield <- rainfall + rnorm(200)
```

```
> round(coef(summary(lm(yield ~ rainfall + nutrients))), 3)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.041	0.074	0.551	0.582
rainfall	1.105	0.100	11.088	0.000
nutrients	0.053	0.073	0.722	0.471

With rainfall in the model: nutrients **p = 0.5**. Rainfall drives both.

Confounders you didn't measure

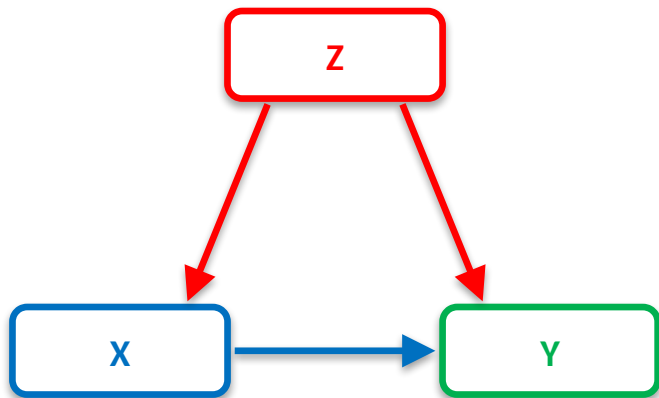


Find a spurious correlation.

tylervigen.com/spurious-correlations

The usual confounder: **time**.

Confounder: adjust for it

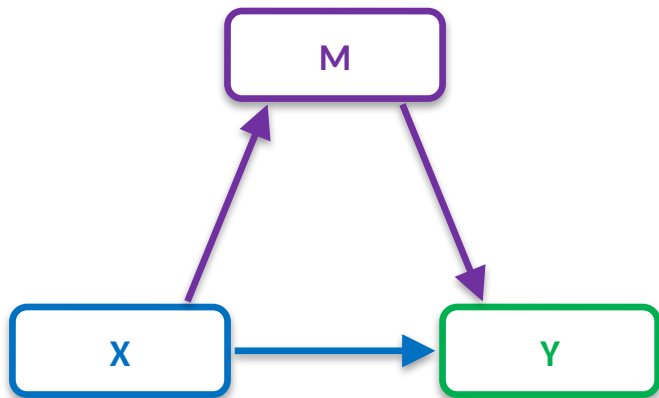


Z causes both X and Y.

Put Z in the model.

```
> without <- lm(yield ~ nutrients)
> with    <- lm(yield ~ rainfall + nutrients)
> round(coef(summary(without))["nutrients", ], 3)
Estimate Std. Error  t value  Pr(>|t|)
   -0.546    0.063   -8.729    0.000
> round(coef(summary(with))["nutrients", ], 3)
Estimate Std. Error  t value  Pr(>|t|)
   0.053    0.073    0.722    0.471
```

Mediator: depends on the question



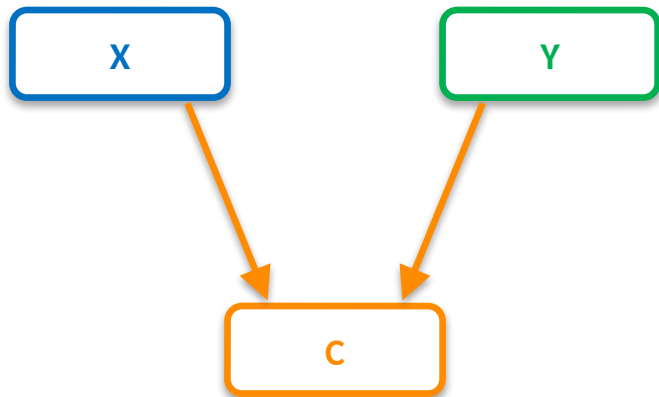
M is on the path from X to Y.

With M: X's direct effect.

Without M: X's total effect.

```
> total <- lm(yield ~ temperature)
> direct <- lm(yield ~ temperature + insects)
> round(coef(summary(total))["temperature", ], 3)
      Estimate Std. Error    t value    Pr(>|t|)
      0.020      0.155      0.130      0.897
> round(coef(summary(direct))["temperature", ], 3)
      Estimate Std. Error    t value    Pr(>|t|)
      0.968      0.160      6.043      0.000
```

Collider: leave it out



X and Y both cause C.

Adjusting for C **invents** an X-Y association.

```
> without <- lm(y ~ x)
> with    <- lm(y ~ x + c)
> round(coef(summary(without))["x", ], 3)
      Estimate Std. Error   t value   Pr(>|t|)
      0.022     0.057      0.389     0.697
> round(coef(summary(with))["x", ], 3)
      Estimate Std. Error   t value   Pr(>|t|)
      -0.740     0.034     -21.737     0.000
```

Multicollinearity

When two predictors say the same thing

Case 3: predictors that overlap

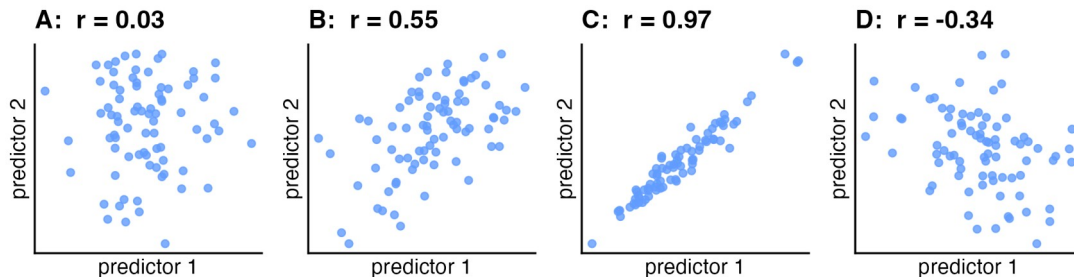
```
> set.seed(4)
> temperature <- rnorm(200)
> precipitation <- rnorm(200)
> microbiome <- temperature + precipitation +
+   rnorm(200, sd = 0.1) # almost their sum
> growth <- temperature + precipitation + rnorm(200)
> model_c <- lm(growth ~ temperature + precipitation +
+   microbiome)
```

```
> round(coef(summary(model_c)), 3)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.031      0.069   -0.445    0.657
temperature     0.447      0.714    0.626    0.532
precipitation   0.423      0.720    0.587    0.558
microbiome     0.471      0.711    0.663    0.508
> summary(model_c)$r.squared
[1] 0.6268291
> summary(model_c)$fstatistic
      value      numdf      dendf
109.7428    3.0000   196.0000
```

A great model ($F = 110$)...

...and no coefficient is significant.

Detect it: the variance inflation factor



$$VIF_j = \frac{1}{1 - R_j^2} \quad R_j^2 : \text{how well the other predictors predict } x_j$$

VIF > 5: look. VIF > 10: drop or combine.

```
> library(car)
> vif(model_c)
temperature precipitation microbiome
100.2172 101.5519 204.0255
> model_c2 <- update(model_c, . ~ . - microbiome) # drop it
> vif(model_c2)
temperature precipitation
1.000266 1.000266
> round(coef(summary(model_c2)), 3)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.032	0.069	-0.464	0.643
temperature	0.917	0.071	12.886	0.000
precipitation	0.897	0.071	12.582	0.000

Perfect collinearity: R gives up

```
> microbiome2 <- temperature + precipitation # their exact sum
> coef(lm(growth ~ temperature + precipitation + microbiome2))
(Intercept)    temperature precipitation    microbiome2
-0.03187741     0.91730146     0.89737751             NA
```

An exact combination of other predictors: **NA**

Real data: bird density in forest patches

```
> birds <- read.csv("data/birds.csv")[, -1] # drop column 1
> head(round(birds, 2), 4)
      bd grzinv  dist height  peri mammal  area  leg
1  9.94   0.25  0.22   0.28 -0.47  -0.01 -0.76 -0.30
2 10.25  -0.35 -1.54  -1.12  0.41   1.41  0.20  0.78
3 10.01  -2.17 -0.13  -0.65 -0.63  -1.57  0.51 -0.82
4 15.82   0.64  1.26   1.81  1.80   2.27  1.45  1.55
> model_birds <- lm(bd ~ ., data = birds) # "." = all others
> round(vif(model_birds), 1)
      grzinv      dist      height      peri      mammal
1256.7 11425669.7  7898629.8  5253300.6  12131.7
      area      leg
939.5  8767479.7
```

VIFs in the millions: dist is almost a function of the other variables

Drop it and refit

```
> model_birds2 <- update(model_birds, . ~ . - dist)
> round(vif(model_birds2), 1)
grzinv height peri mammal area leg
  1.0   1.6   1.6   1.4   1.0   1.8
> round(coef(summary(model_birds2)), 3)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	9.800	0.180	54.388	0.000
grzinv	0.612	0.181	3.375	0.001
height	1.481	0.221	6.690	0.000
peri	0.505	0.220	2.299	0.024
mammal	-0.108	0.217	-0.498	0.620
area	3.321	0.182	18.214	0.000
leg	-1.180	0.253	-4.656	0.000

update(model, . ~ . - dist)

VIFs: 1 to 1.8

Under the hood: a matrix

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad \hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

```
> head(model.matrix(model2), 4)
      (Intercept) height nitrogen
1             1    86.9         89
2             1    71.6          6
3             1    73.1         66
4             1    75.9         70
```

The column of 1s is the intercept's **dummy variable**. Next week:
0/1 columns for groups too.

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$
$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$
$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Reporting a regression

“Biomass increased with height (**0.28 g per cm, 95% CI 0.20 to 0.35**; $t_{38} = 7.5$, $p < 0.001$). Height explained 59% of the variation in biomass ($R^2 = 0.59$, $n = 40$).”

```
> model_std <- lm(scale(biomass) ~ scale(height) +  
+   scale(nitrogen), data = wheat)  
> round(coef(model_std), 2)  # slopes in SD units  
      (Intercept)    scale(height) scale(nitrogen)  
           0.00           0.55           0.44
```

Slopes in SD units: which predictor matters more

Effect size + CI + R^2 + n . **A p-value alone is not a result.**

Rough summary and guide

slope in units + CI

share explained

chance?

a new plant?

trust the p-values?

overlapping predictors?

a third variable?

`coef()`, `confint()`

R^2 , adjusted R^2

t per coefficient, F for the model

`predict()`, inside the data

`plot(model)`: curve $\rightarrow x^2$, fan $\rightarrow \log(y)$

`vif()` > 10: drop or combine

draw the diagram

Week 2 showdown

Kahoot-style showdown

12 questions on this week.

Not graded.

Use a pseudonym if you want.

Fast and correct wins.



Scan with your phone

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w2-showdown](https://greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w2-showdown)

[host screen](#)

Next week: General(ized) linear models

Problem Set 2 due Monday, October 12, 1:10 PM • Friday: data.table how-to

