Model selection

Which predictors belong in the model?

1 Too many models
2 Scoring models
3 Testing ideas
4 Exploring
5 Traps
6 Predicting



Week 4 recap

Kahoot-style warm-up

10 quick questions on last week.

Not graded.

Use a pseudonym if you want.

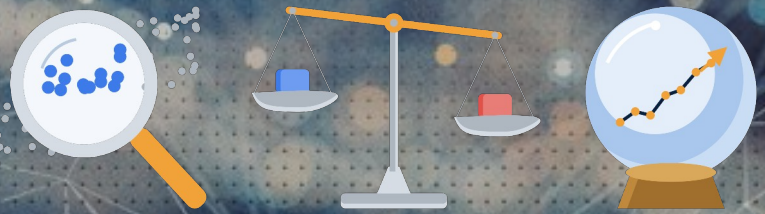
Fast and correct wins.

Scan with your phone

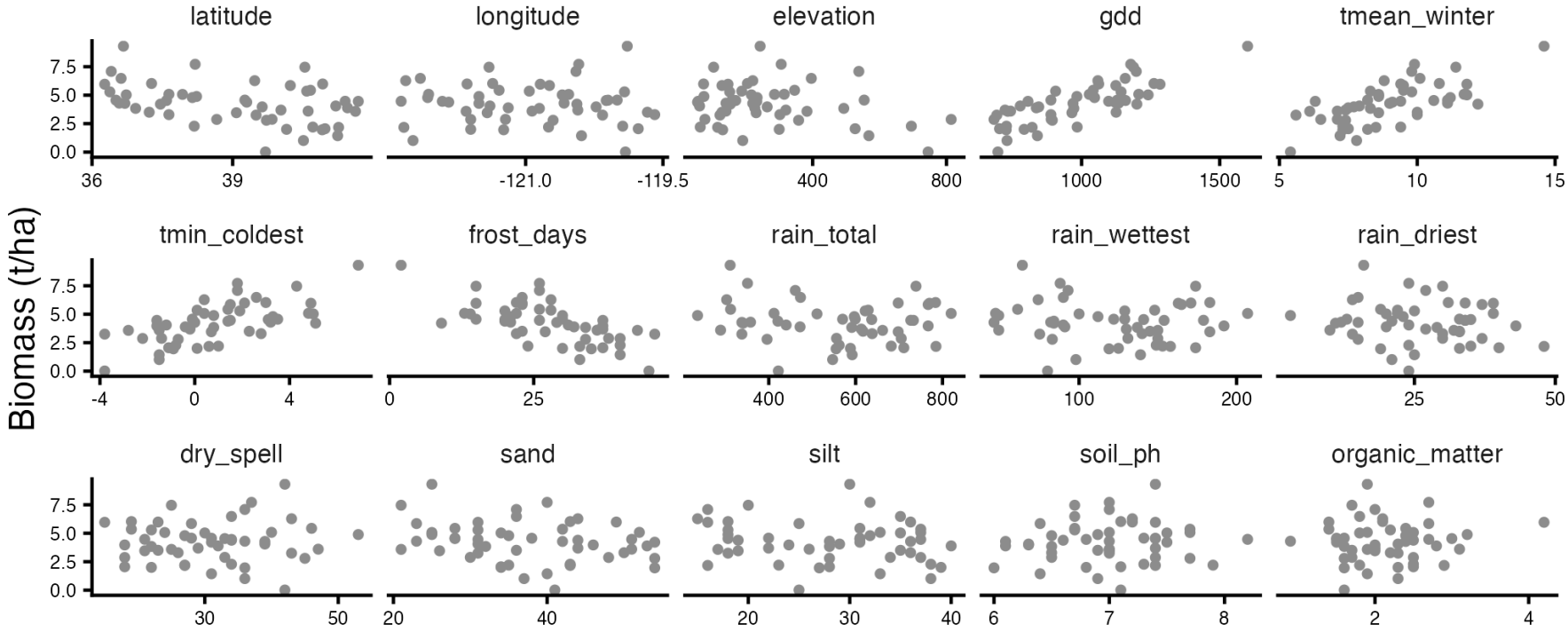
[Join on your phone or laptop: greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w5-week4-recap](greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w5-week4-recap)

[host screen]

1 Too many models
2 Scoring models
3 Testing ideas
4 Exploring
5 Traps
6 Predicting



The cover-crop farms are back



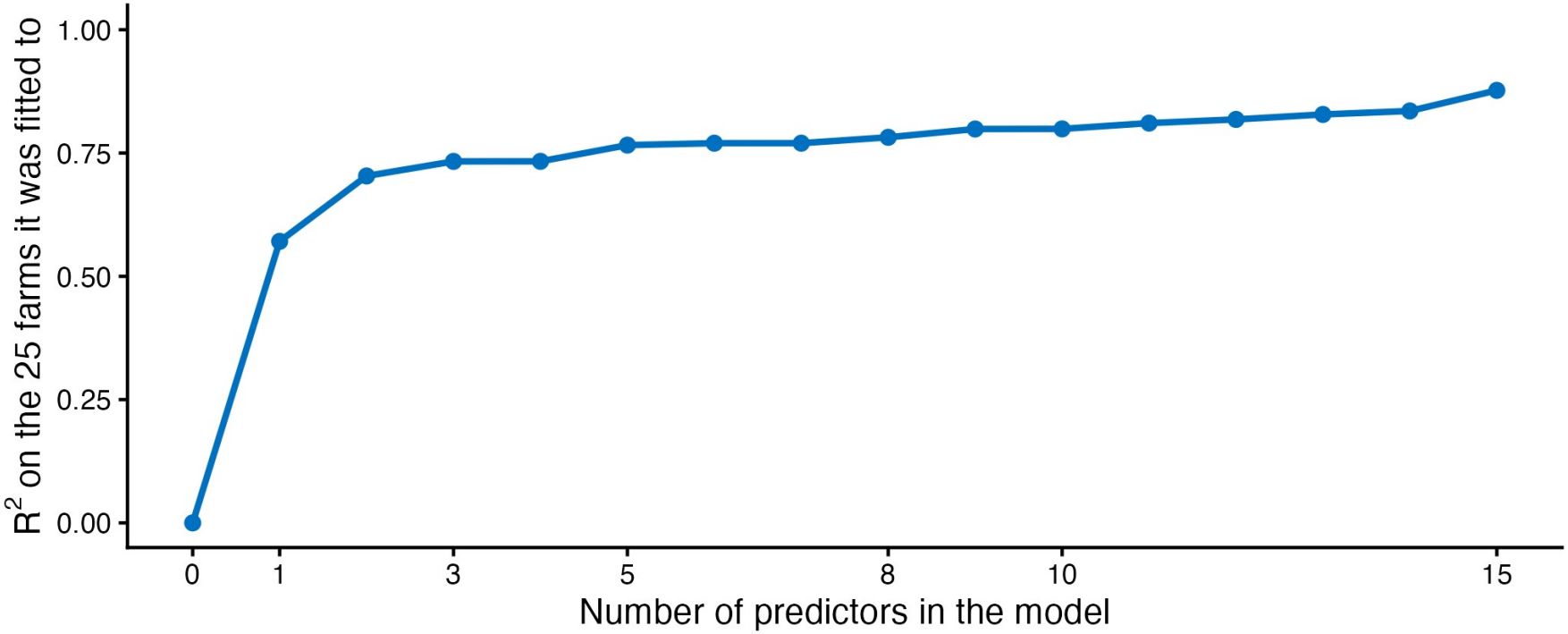
Biomass on **50 farms** against 15 candidate predictors

How many models could we fit?

```
> farms <- read.csv("https://greymonroe.github.io/PLS_206/data/cover_crop.csv")
> farms <- subset(farms, select = -c(farm, clay))
> ncol(farms) - 1          # candidate predictors
[1] 15  ← candidate predictors
> 2^15                     # possible models (each one in or out)
[1] 32768  ← possible models
```

Each predictor is in or out: **2^15 = 32,768 models**

Add predictors: R^2 only goes up



Fitted on 25 farms: R^2 rises from **0.57** to **0.88**

Live poll

Multiple choice

Fitted on 25 farms. Which model predicts the other 25 farms best?

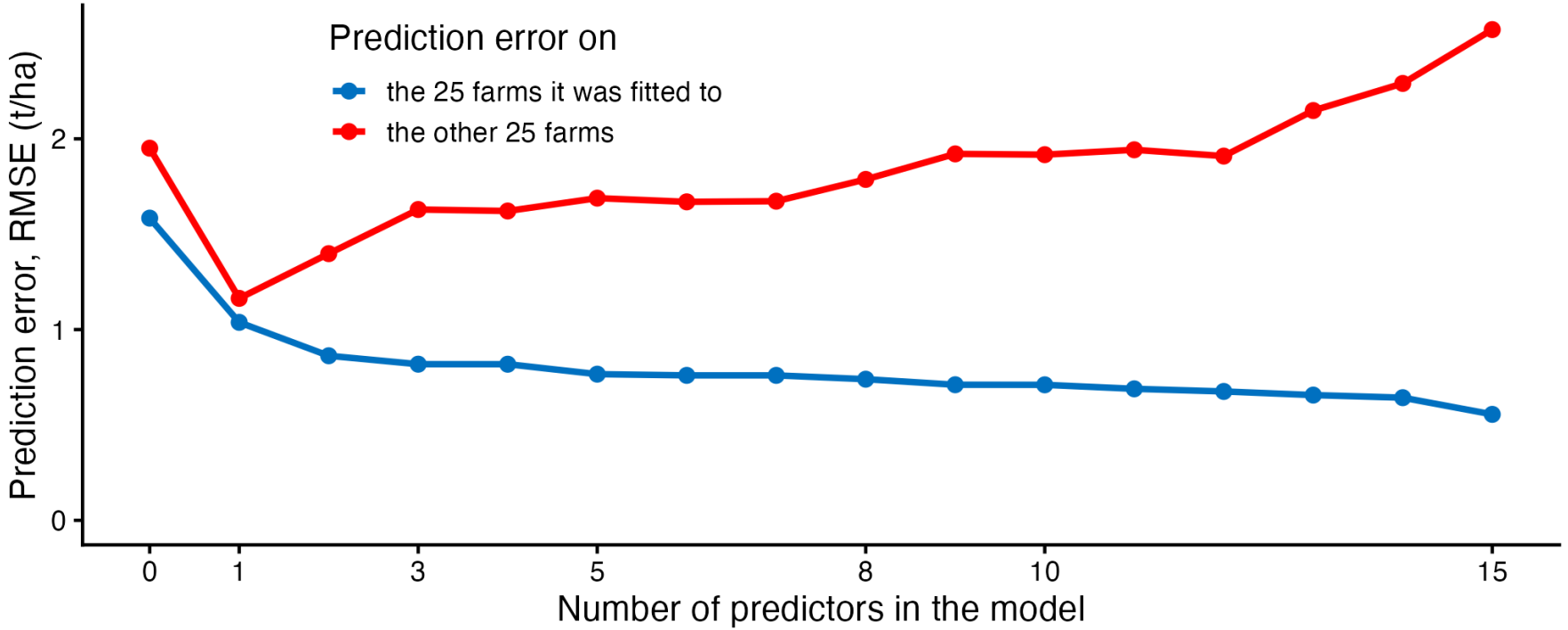
A 1 predictor (gdd)
B 3 predictors
C 8 predictors
D all 15 predictors

Scan with your phone

Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w5-best-k

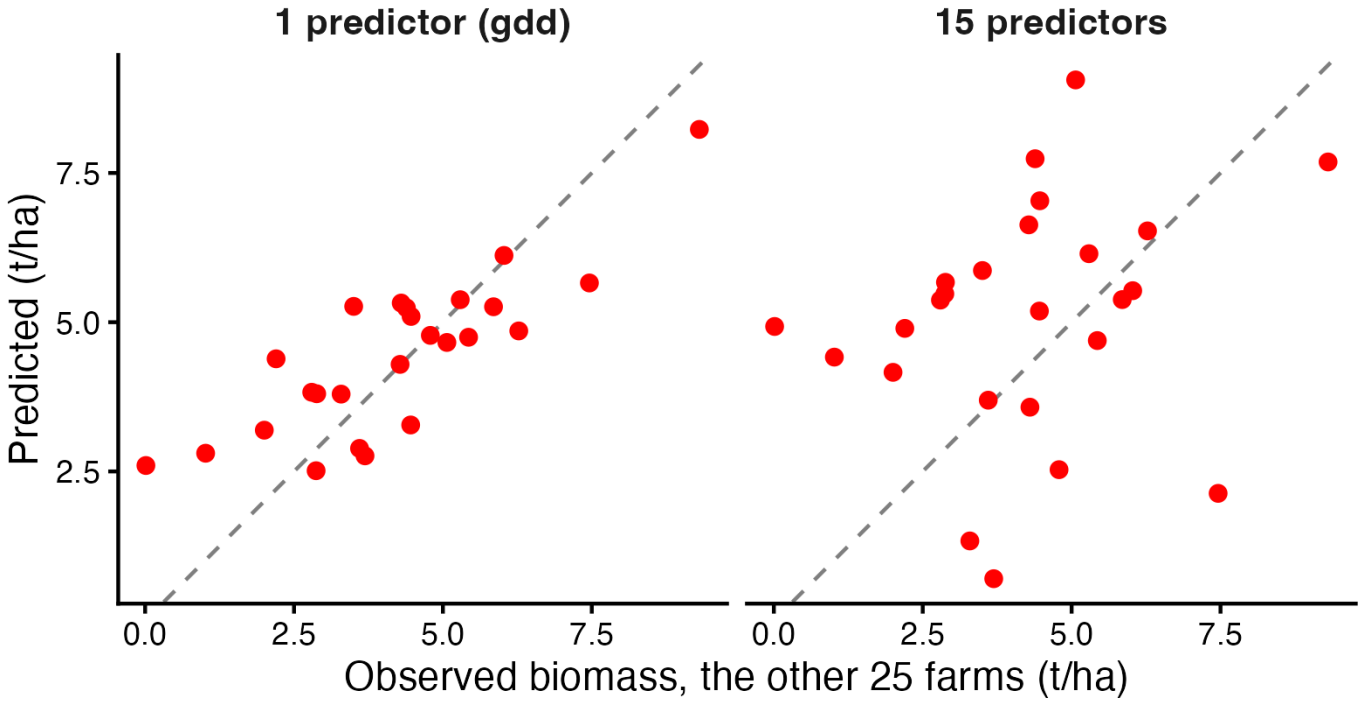
host screen

...but the other 25 farms get worse



Error on new farms: **1.16 t/ha** with gdd alone, **2.57 t/ha** with all 15

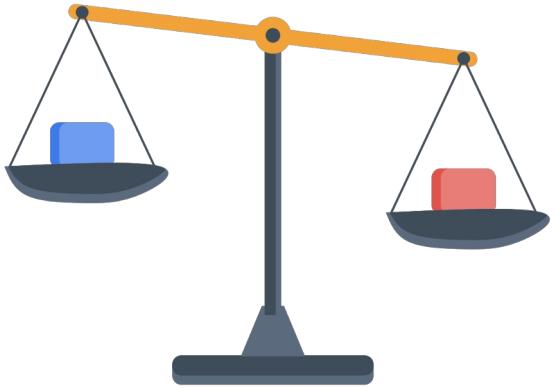
Overfitting: the model learned the noise



Too simple = bias (wrong on average). **Too flexible** = variance (unstable): overfitting

What is the model for?







Explore **Test ideas** **Predict**

Each goal has **its own way to choose**

A tool for each goal

Explore	**search all subsets; findings = new hypotheses**
Test ideas	**a few models written first; AIC weights**
Predict	**held-out data, cross-validation (Week 6)**

Today: the scoring tool and **testing ideas**. Wednesday: **exploring** and **predicting**

1 Too many models

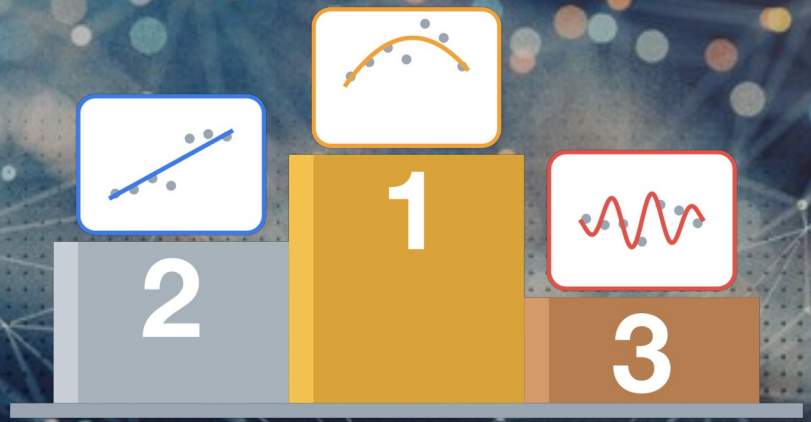
2 **Scoring models**

3 Testing ideas

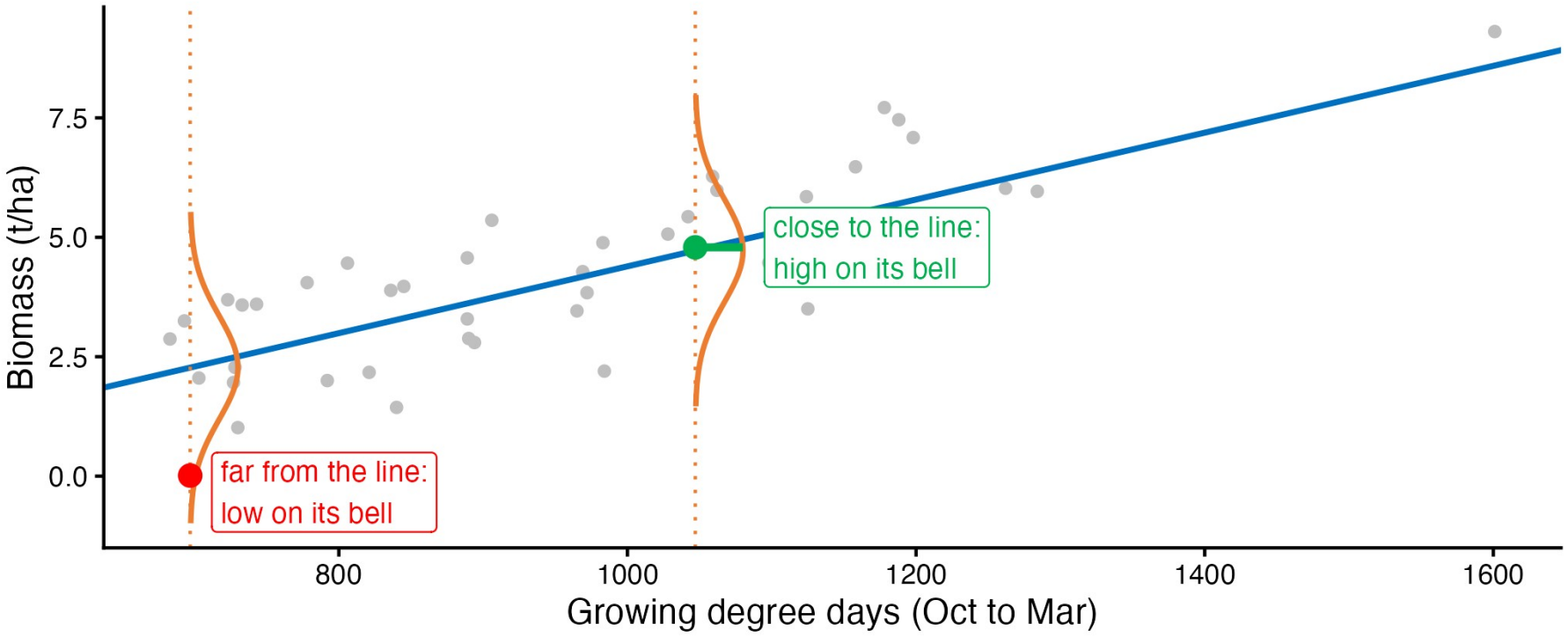
4 Exploring

5 Traps

6 Predicting



Likelihood, one farm at a time



The model puts a **bell around the line**. Likelihood = how high each farm sits on its bell

The calculation: the height of the bell at y

$$L_i = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \hat{y}_i)^2}{2\sigma^2}\right)$$

green: $y - \hat{y} = 4789 - 4721 = 68$ $L = 3.67 \times 10^{-4} \times e^{-0.002} = 3.67 \times 10^{-4}$ $\log L = -7.91$

red: $y - \hat{y} = 13 - 2273 = -2260$ $L = 3.67 \times 10^{-4} \times e^{-2.16} = 0.42 \times 10^{-4}$ $\log L = -10.07$

y_i: farm i's biomass $\hat{y}_i$: the line's prediction there σ: width of the bell, 1,086 kg/ha

$1/(\sigma\sqrt{2\pi})$: the height of the bell's peak $\exp(\ldots)$: how far down the bell the farm sits

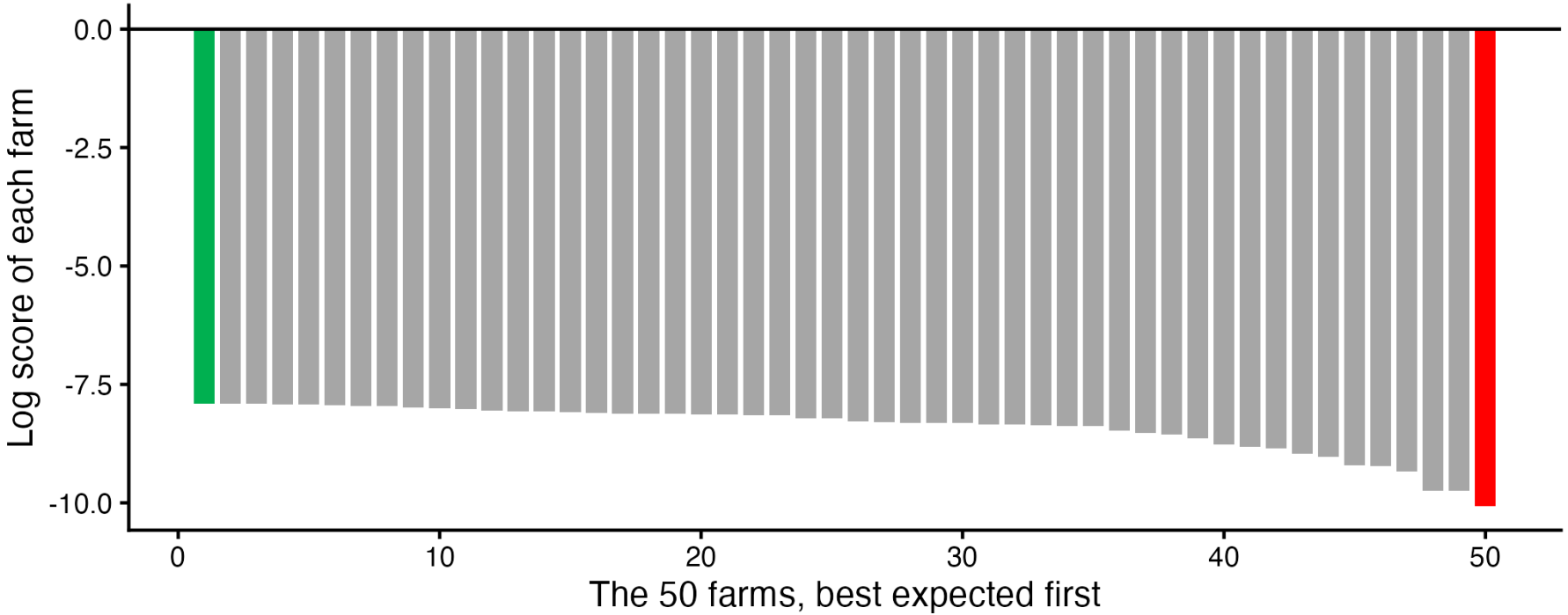
On the line: $e^0 = 1$, the full peak. Two σ away: $e^{-2} \approx 0.13$ of the peak

One farm in R: dnorm() is that formula

```
> y     <- farms$biomass[4]     # green farm
> y_hat <- fitted(m_temp)[[4]]  # predicted there
> sigma <- sqrt(mean(resid(m_temp)^2))
> 1 / (sigma * sqrt(2 * pi)) * exp(-(y - y_hat)^2 / (2 *
sigma^2))
[1] 0.0003665691  ← by hand
> dnorm(y, mean = y_hat, sd = sigma)
[1] 0.0003665691  ← dnorm(): the same
> log(dnorm(y, mean = y_hat, sd = sigma))
[1] -7.911323  ← its log score
```

dnorm(y, mean = ŷ, sd = σ) = the height of the bell at y. Take the log: **-7.91**

50 farms, one score: add up the logs



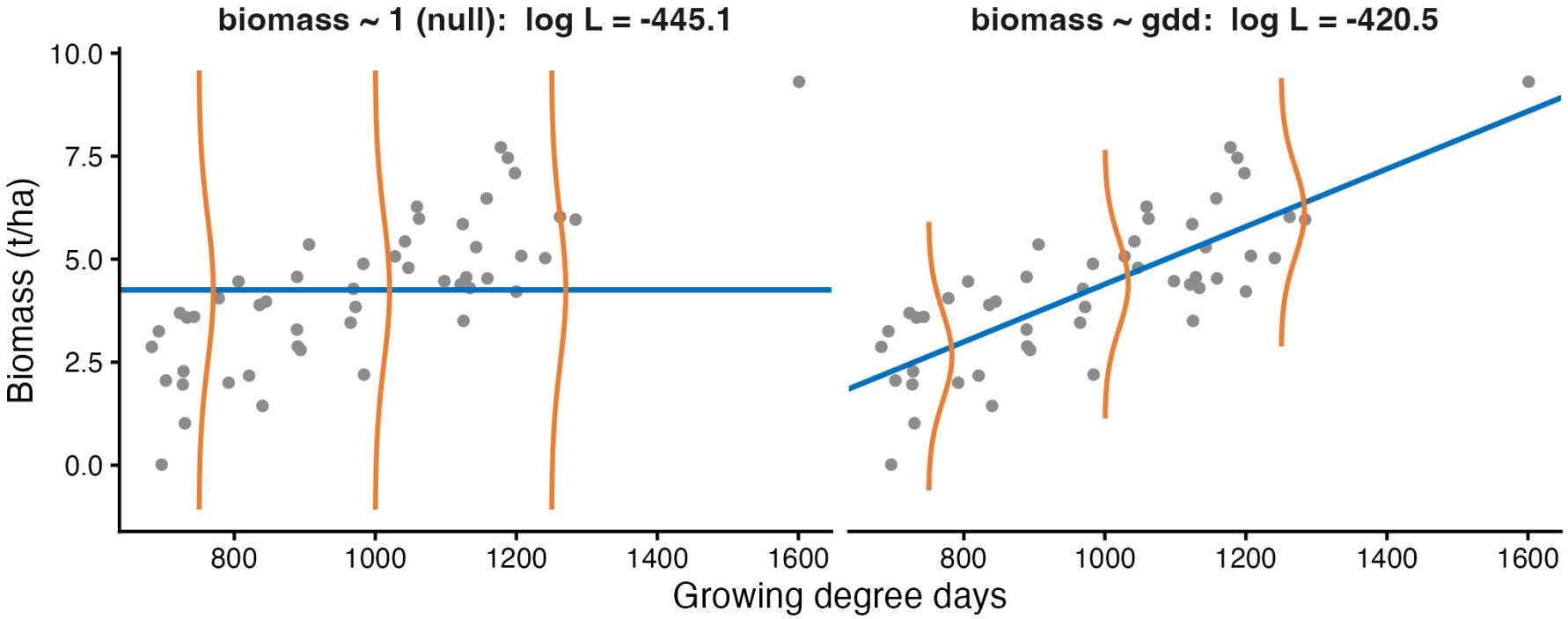
Multiply the 50 heights; tiny numbers, so **add their logs**: log L = -420.47

By hand: dnorm() for each farm, then sum()

```
> sigma <- sqrt(mean(resid(m_temp)^2))    # width of the bell
> scores <- dnorm(farms$biomass, mean = fitted(m_temp),
+                 sd = sigma, log = TRUE) # log height, each
farm
> round(range(scores), 2)
[1] -10.07  -7.91  ← worst and best farm
> sum(scores)
[1] -420.4669  ← add them up
> logLik(m_temp)
'log Lik.' -420.4669 (df=3)  ← what R reports
```

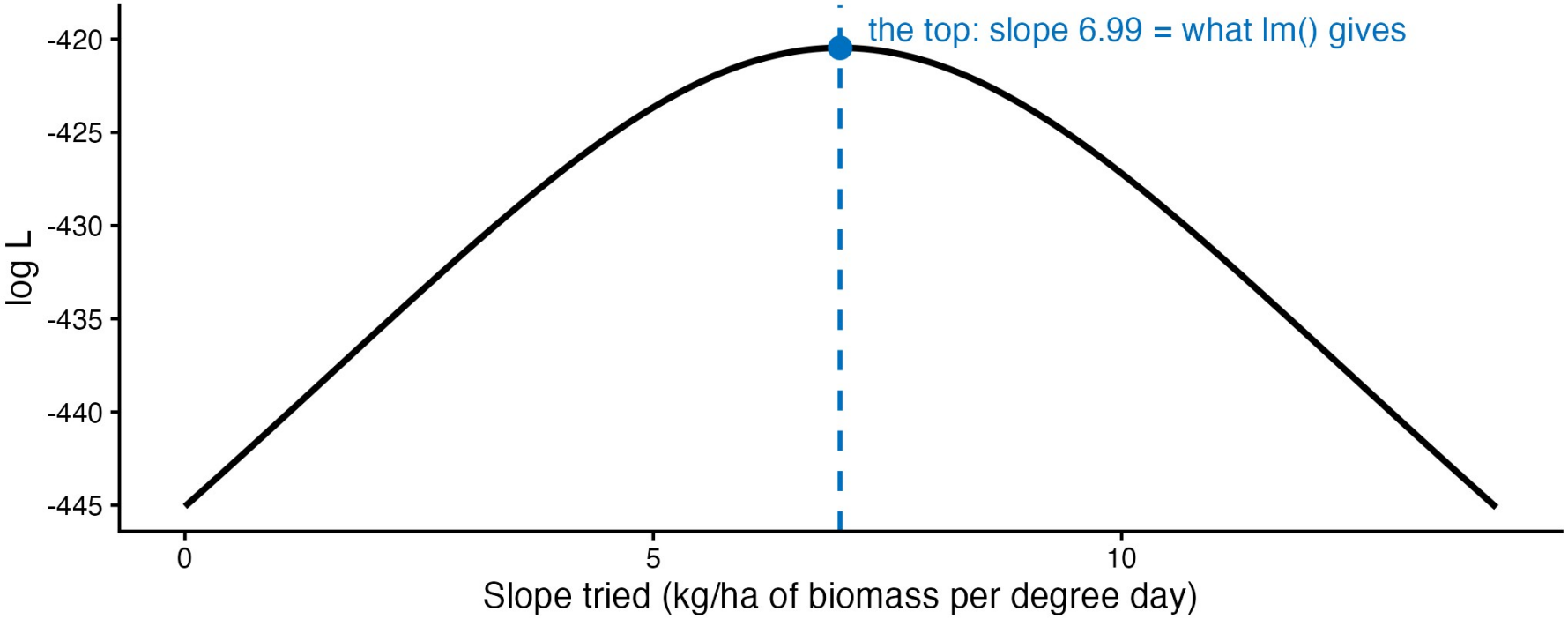
sum(scores) = **logLik()**. df = 3: intercept, slope, **and σ**

A worse model expects the data less



No predictor: one wide bell for every farm. **log L = -445.1** vs -420.5

lm() picks the line with the highest likelihood



Maximum likelihood: try every slope, keep the one where log L peaks

AIC = misfit + a price for each parameter

$$\text{AIC} = -2 \log L + 2k$$

$$= 840.93 + 2 \times 3 = 846.93$$

$-2 \log L$: misfit (lower = closer to the data) k: parameters estimated

L: the likelihood lower AIC = better

Lower AIC = **closer to the data for its size**

AIC() does the same sum

```
> AIC(m_temp)
[1] 846.9337  ← AIC()
> -2 * as.numeric(logLik(m_temp)) + 2 * 3     # k = 3:
intercept, slope, sigma
[1] 846.9337  ← by hand: the same
```

k = 3: **σ counts** as a parameter

Live poll

Number: live histogram

The null model, biomass ~ 1, has log L = -445.07 and k = 2.

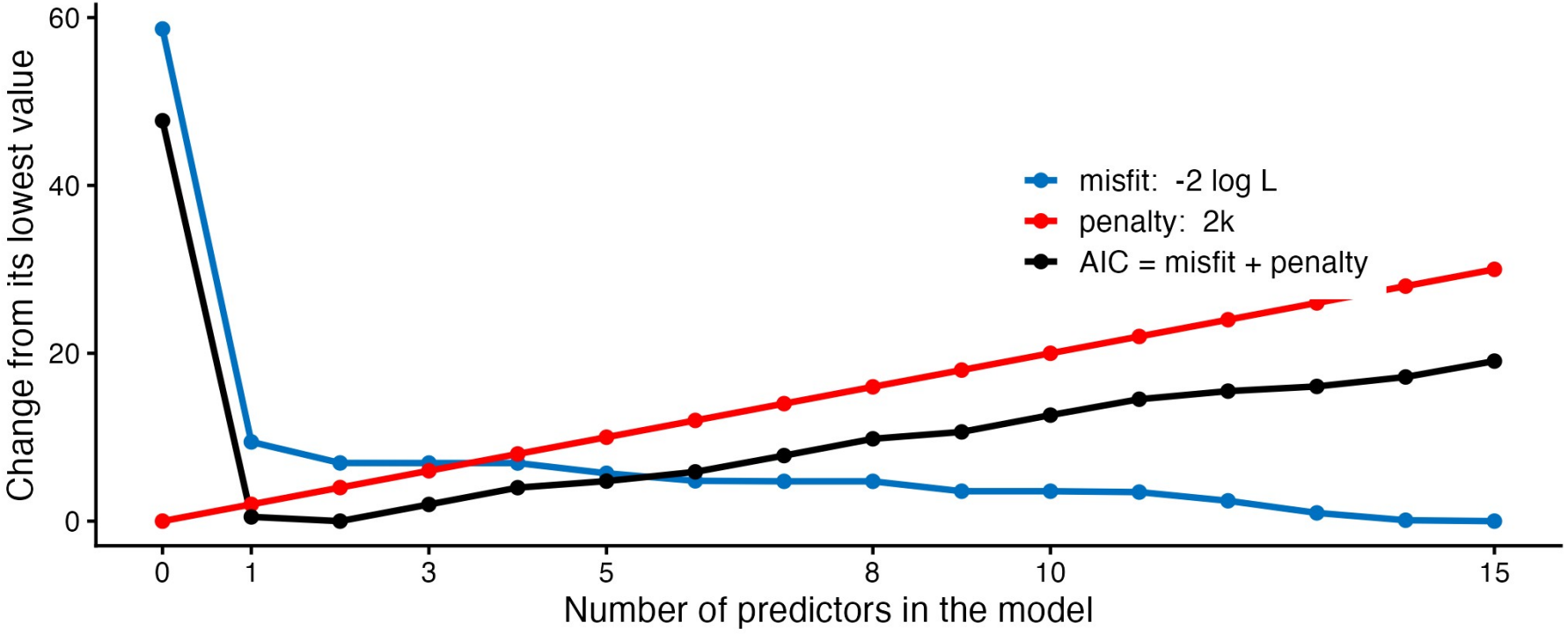
What is its AIC?

Scan with your phone

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w5-aic-null](greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w5-aic-null)

[host screen]

Misfit falls, the penalty rises, AIC finds the balance



All 50 farms: AIC is lowest at **2 predictors**

Only differences matter: ΔAIC

ΔAIC 0 to 2	**about equally good**
ΔAIC 4 to 7	**clearly less support**
ΔAIC above 10	**essentially no support**

ΔAIC = this model's AIC **minus the best AIC in the set**

Akaike weights: share of the support

$$w_i = \frac{e^{-\Delta_i/2}}{\sum_j e^{-\Delta_j/2}}$$

$$\Delta = 0,\ 2.11,\ 5.89 \ \rightarrow\ w = 0.71,\ 0.25,\ 0.04$$

Δ_i: AIC of model i minus the lowest AIC in the set w_i: its Akaike weight (they sum to 1)

Weights sum to 1 across **the models in the set**

AICc: a correction for small samples

$$\text{AICc} = \text{AIC} + \frac{2k(k+1)}{n-k-1}$$

$$= 846.93 + \frac{2\times3\times4}{50-3-1} = 847.46$$

n: observations k: parameters the extra term vanishes as n grows

Use **AICc** when n is less than about 40 times k; it becomes AIC for big n

BIC: a stiffer price

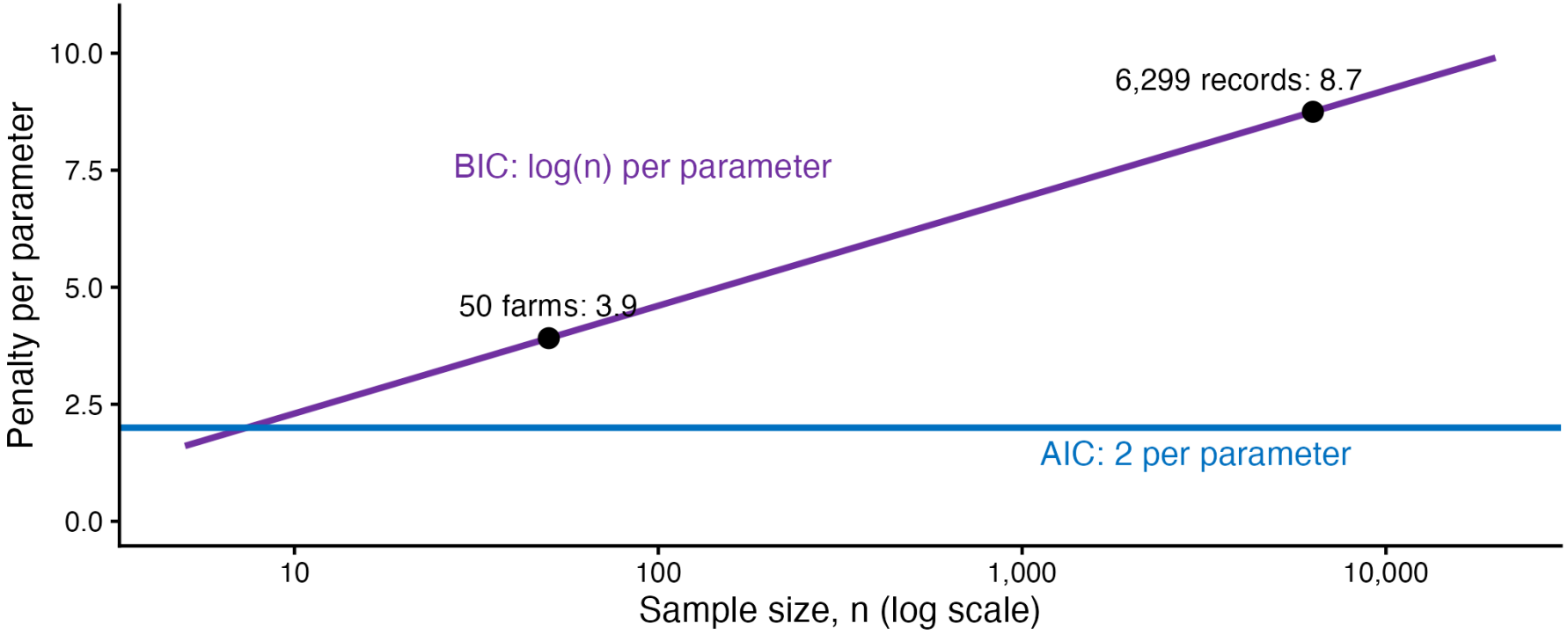
$$\text{BIC} = -2 \log L + \log(n)\, k$$

$$= 840.93 + 3.91 \times 3 = 852.67$$

$\log(n)$: natural log of the sample size, $\log(50) = 3.91$, bigger than 2 once $n > 7$

```
> AICc(m_temp)      # MuMIn
[1] 847.4555  ← AICc
> BIC(m_temp)
[1] 852.6698  ← BIC
```

AIC or BIC?



AIC: best predictions. **BIC**: smaller models, finds the true one if it is in the set

1 Too many models
2 Scoring models
3 **Testing ideas**
4 Exploring
5 Traps
6 Predicting



Ideas first, then models

Temperature

```
biomass ~ gdd
```

Moisture

```
biomass ~ rain_total
```

Soil

```
biomass ~
organic_matter
      + soil_ph
```

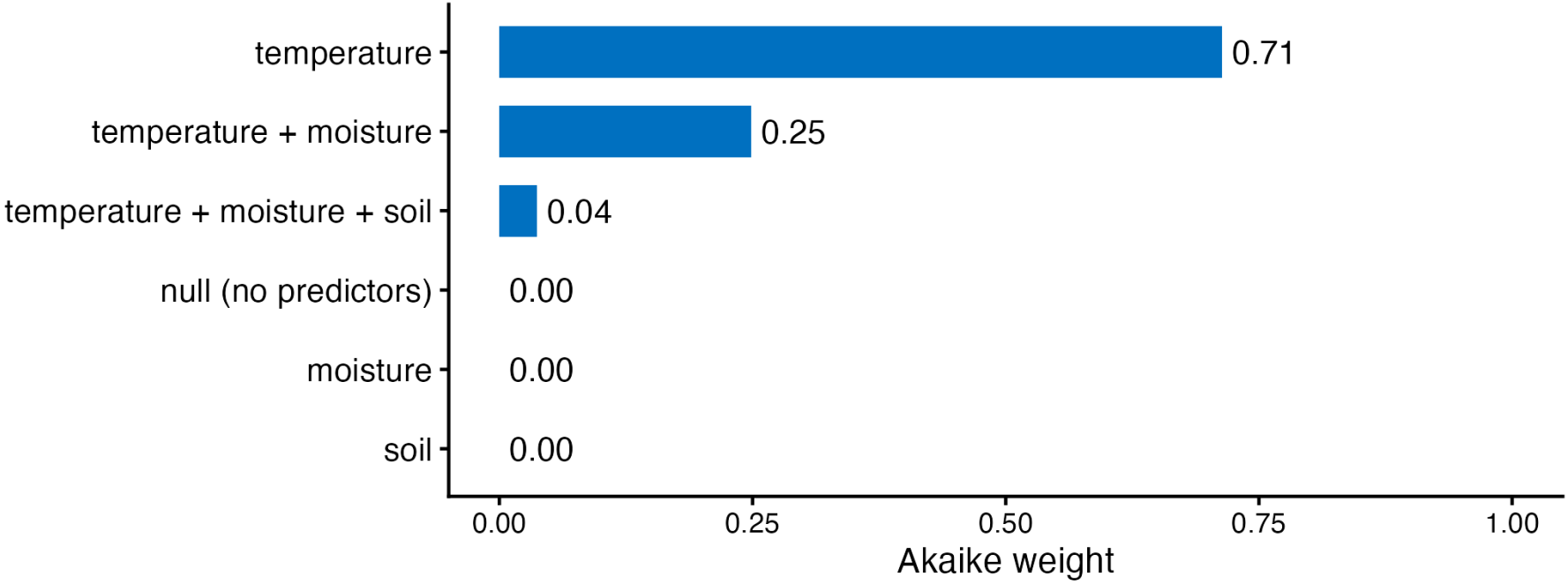
plus: temperature + moisture, all three, and the null

Six models, **written before looking** at the data

model.sel(): the six models, ranked

```
> m_null  <- lm(biomass ~ 1, data = farms)
> m_temp  <- lm(biomass ~ gdd, data = farms)
> m_moist <- lm(biomass ~ rain_total, data = farms)
> m_soil  <- lm(biomass ~ organic_matter + soil_ph, data =
farms)
> m_tm    <- lm(biomass ~ gdd + rain_total, data = farms)
> m_all   <- lm(biomass ~ gdd + rain_total + organic_matter +
soil_ph, data = farms)
> ms <- model.sel(m_null, m_temp, m_moist, m_soil, m_tm, m_all)
> round(as.data.frame(ms)[, c("df", "logLik", "AICc", "delta",
"weight")], 2)
        df  logLik   AICc delta weight
m_temp   3 -420.47 847.46  0.00   0.71  ← weight
m_tm     4 -420.34 849.56  2.11   0.25  ← delta
m_all    6 -419.70 853.35  5.89   0.04
m_null   2 -445.07 894.39 46.94   0.00
m_moist  3 -444.68 895.88 48.43   0.00
m_soil   4 -444.83 898.55 51.10   0.00
```

The evidence, as weights



Temperature has 71% of the support; moisture and soil add nothing

Nested models: anova() and AIC agree

```
> anova(m_temp, m_tm)      # does adding moisture help?
Analysis of Variance Table

Model 1: biomass ~ gdd
Model 2: biomass ~ gdd + rain_total
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1     48 58986591
2     47 58681162  1    305428 0.2446 0.6232  ← p: no help
```

Moisture adds nothing: **p = 0.62**, ΔAICc = **2.11**

Not nested? Only AIC can compare them

```
> anova(m_moist, m_soil)   # moisture vs soil: not nested
Analysis of Variance Table

Model 1: biomass ~ rain_total
Model 2: biomass ~ organic_matter + soil_ph
  Res.Df       RSS Df Sum of Sq F Pr(>F)
1     48 155377480
2     47 156318053  1   -940573  ← no test: not nested
```

Moisture vs soil: no F test exists. **AIC compares any models** fitted to the same data

Some terms stay, no matter what

The design

blocks, plots, plates, repeated measures

Known controls

sampling effort, body size, planting date

Main effects

any variable in an interaction you keep

Selection is for the **uncertain** terms. The rest are in every candidate model.

Same response: log(y) and y cannot be compared

```
> m_raw <- lm(biomass ~ gdd, data = farms)
> m_log <- lm(log(biomass) ~ gdd, data = farms)
> AIC(m_raw, m_log)
       df      AIC
m_raw   3 846.9337
m_log   3 123.1217  ← not comparable!
```

Different response = different likelihood. Compare on one scale only.

Same rows: missing values change n

```
> farms_na <- farms
> farms_na$soil_ph[c(3, 17, 29)] <- NA    # suppose 3 soil
tests were lost
> m_a <- lm(biomass ~ gdd, data = farms_na)
> m_b <- lm(biomass ~ gdd + soil_ph, data = farms_na)
> c(nobs(m_a), nobs(m_b))
[1] 50 47  ← different rows
> AIC(m_a, m_b)
    df      AIC
m_a  3 846.9337
m_b  4 799.1344  ← not comparable!
```

Drop the NA rows first, then fit every candidate model on the same data

Report the table, not just the winner

Model	k	AICc	ΔAICc	weight
biomass ~ gdd	3	847.5	0.0	0.71
+ rain_total	4	849.6	2.1	0.25
+ rain + soil	6	853.3	5.9	0.04
null	2	894.4	46.9	0.00

Then the winner's **estimates with CIs** and a plot of the fit

Live coding: scoring models



`Week5_live_1_aic.R`

Part 1

1 Overfitting: 25 farms vs the other 25

2 Likelihood and AIC by hand

3 AICc, BIC

4 Six ideas, model.sel()

5 The two traps

Model selection, part 2

1 Too many models
2 Scoring models
3 Testing ideas
▶ **4 Exploring**
5 Traps
6 Predicting



Monday recap

Kahoot-style warm-up

8 quick questions on Monday.

Not graded.

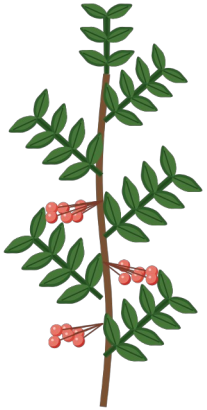
Fast and correct wins.

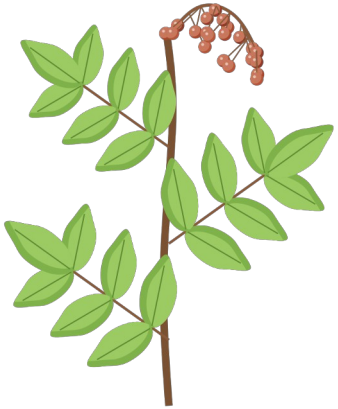
Scan with your phone

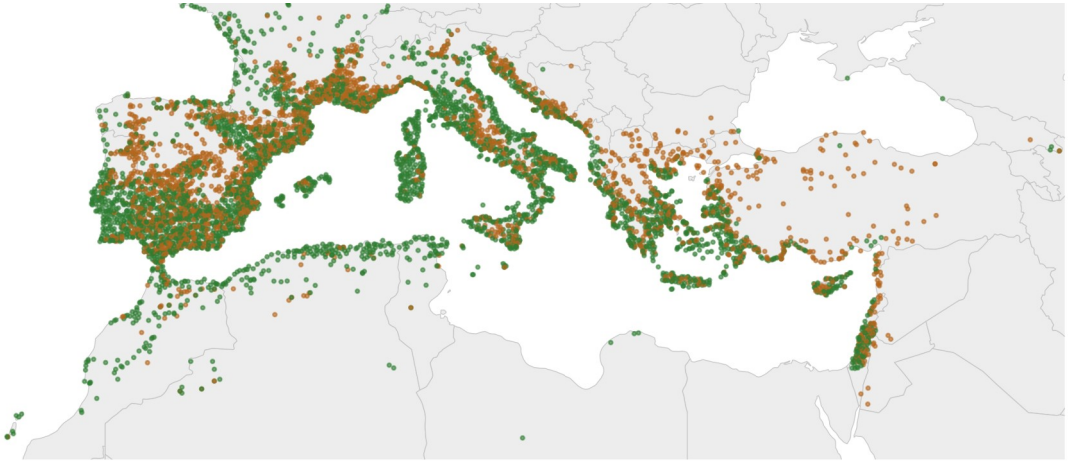
Join on your phone or laptop: greymonroe.github.io/live-poll-test/quiz-play.html?quiz=pls206-w5-monday-recap

host screen

Two wild pistachios







P. lentiscus
mastic tree, evergreen

P. terebinthus
turpentine tree, deciduous

6,299 records: **3,678 lentiscus**, **2,621 terebinthus**. Their ranges overlap.

Where each row comes from

A sighting → **The climate there** → **One row**

GBIF: someone recorded a plant, its species, and where

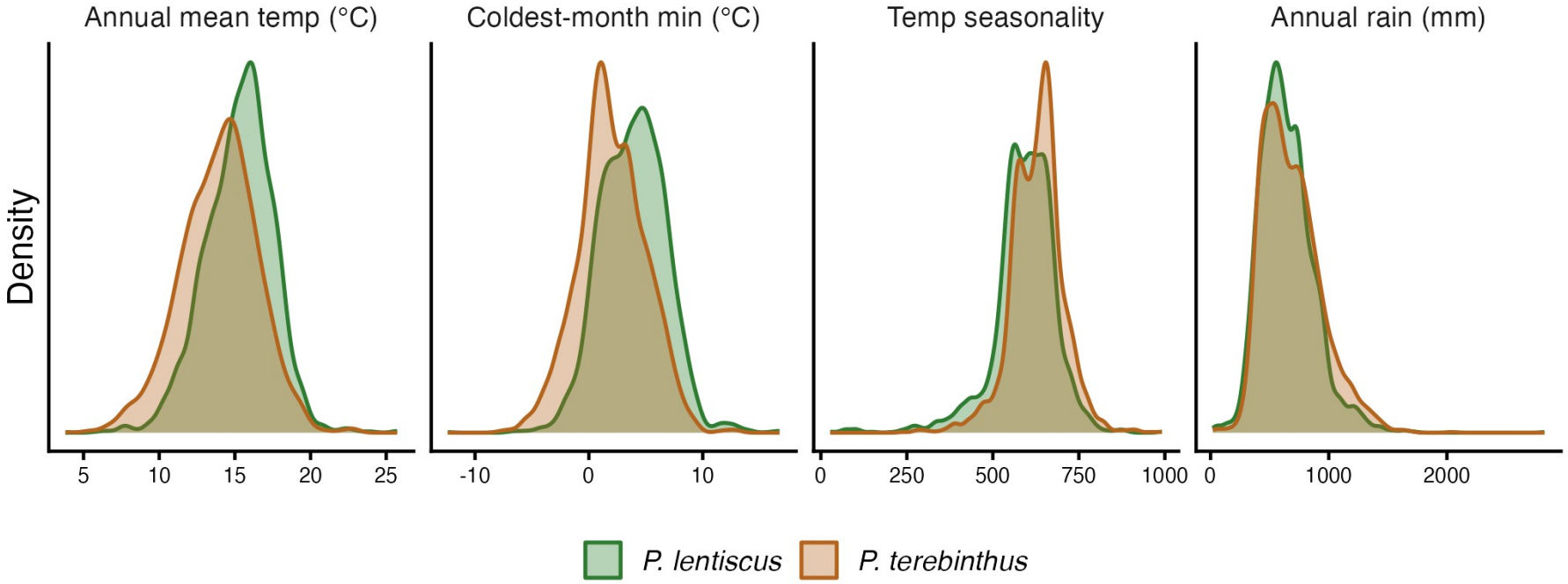
WorldClim: 19 climate variables, averaged over 1970 to 2000

species + location + 19 climate numbers

```
> rows <- c(1, 2000, 4000, 6000)
> round(pis[rows, c("lat", "lon", "bio1", "bio4", "bio12")], 1)
      lat  lon bio1  bio4 bio12
1    32.6 35.1 18.7 569.0   594
2000 42.1 24.4 11.6 799.2   542
4000 39.0 -1.9 14.3 717.1   378
6000  5.3 39.6 20.2 101.8   719
> pis$species_short[rows]
[1] "lentiscus"   "terebinthus" "terebinthus" "lentiscus"
```

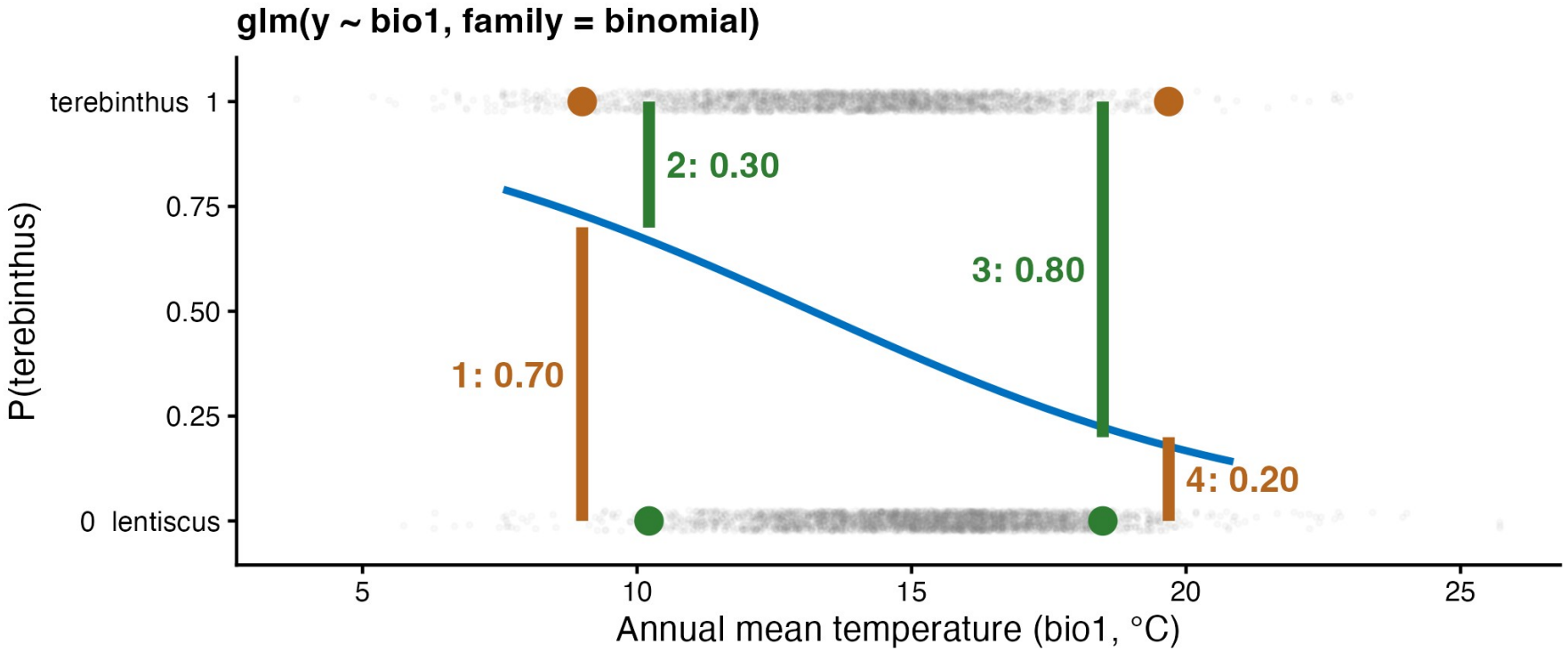
Question: from the climate, **which species is it?** y = 1 terebinthus, 0 lentiscus

19 climate variables, all tangled



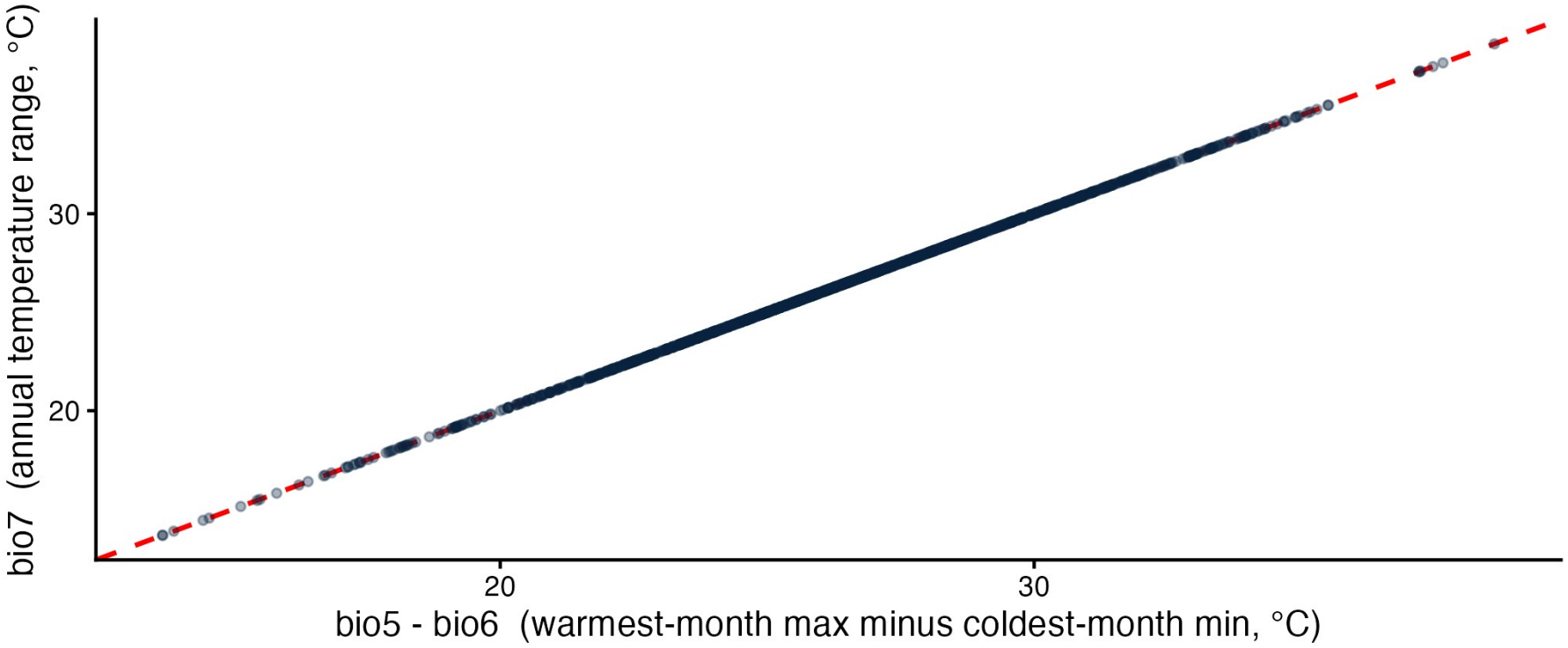
The species differ, but **overlap a lot**; the 19 variables are built from the same months

First model: species ~ temperature



Each record's likelihood: **height of the curve** if terebinthus, **1 – height** if lentiscus

Throw everything in? bio7 is bio5 – bio6



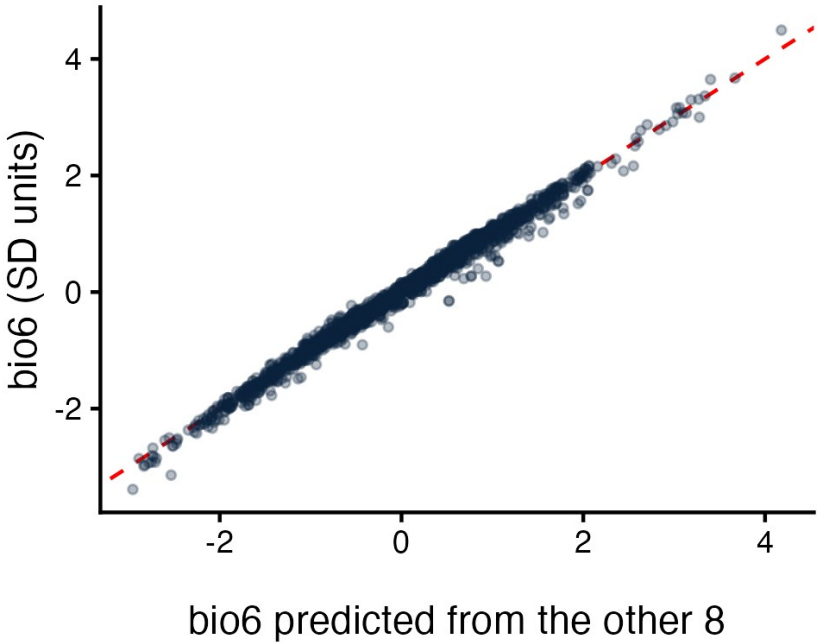
All 19 in one model: coefficients of **±29,776**. Drop bio7.

No pair is correlated, yet one is redundant

Every pair: |r| < 0.7

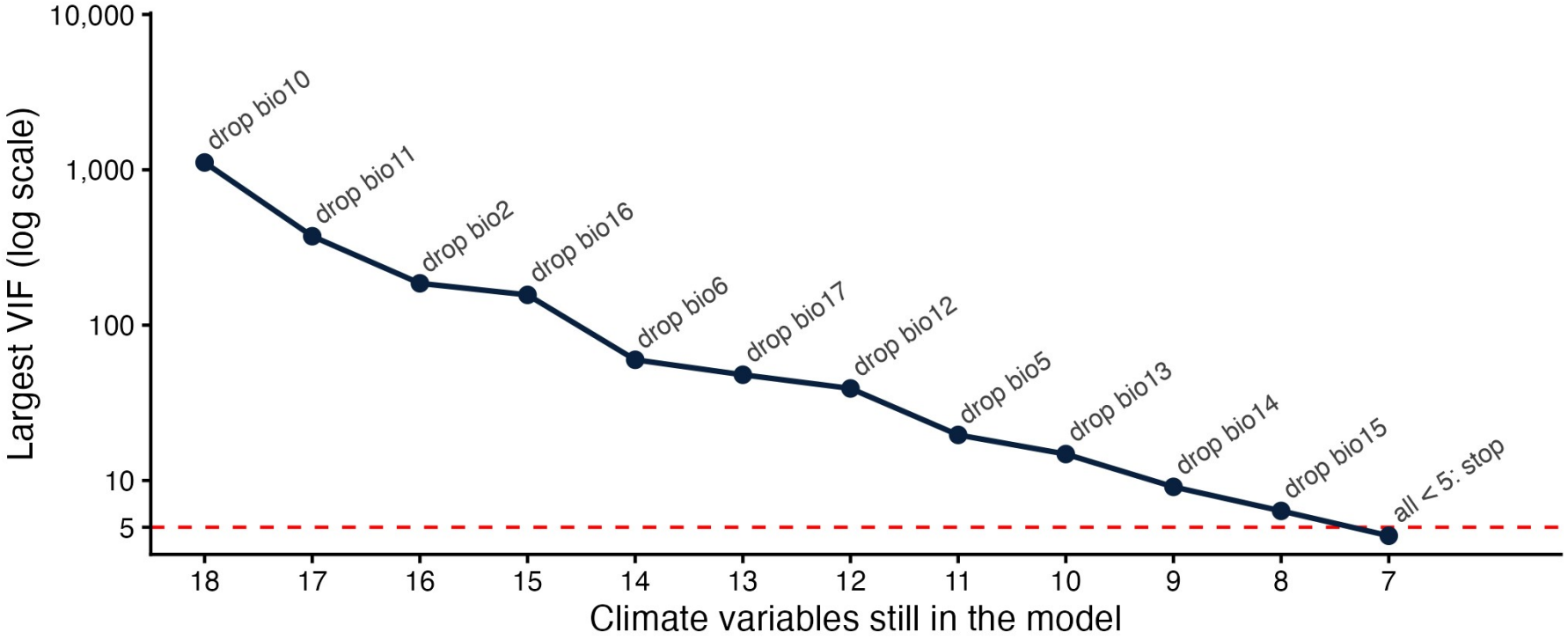
	bio8	bio4	bio10	bio15	bio9	bio2	bio12	bio3
bio6	0.37	0.63	0.58	0.66	0.53	0.41	0.15	0.11
bio8		0.22	0.29	0.08	0.12	0.13	0.25	0.06
bio4			0.18	0.23	0.09	0.36	0.14	0.43
bio10				0.63	0.61	0.17	0.47	0.02
bio15					0.55	0.02	0.17	0.18
bio9						0.04	0.25	0.08
bio2							0.35	0.66
bio12								0.17

But together: $R^2 = 0.989$



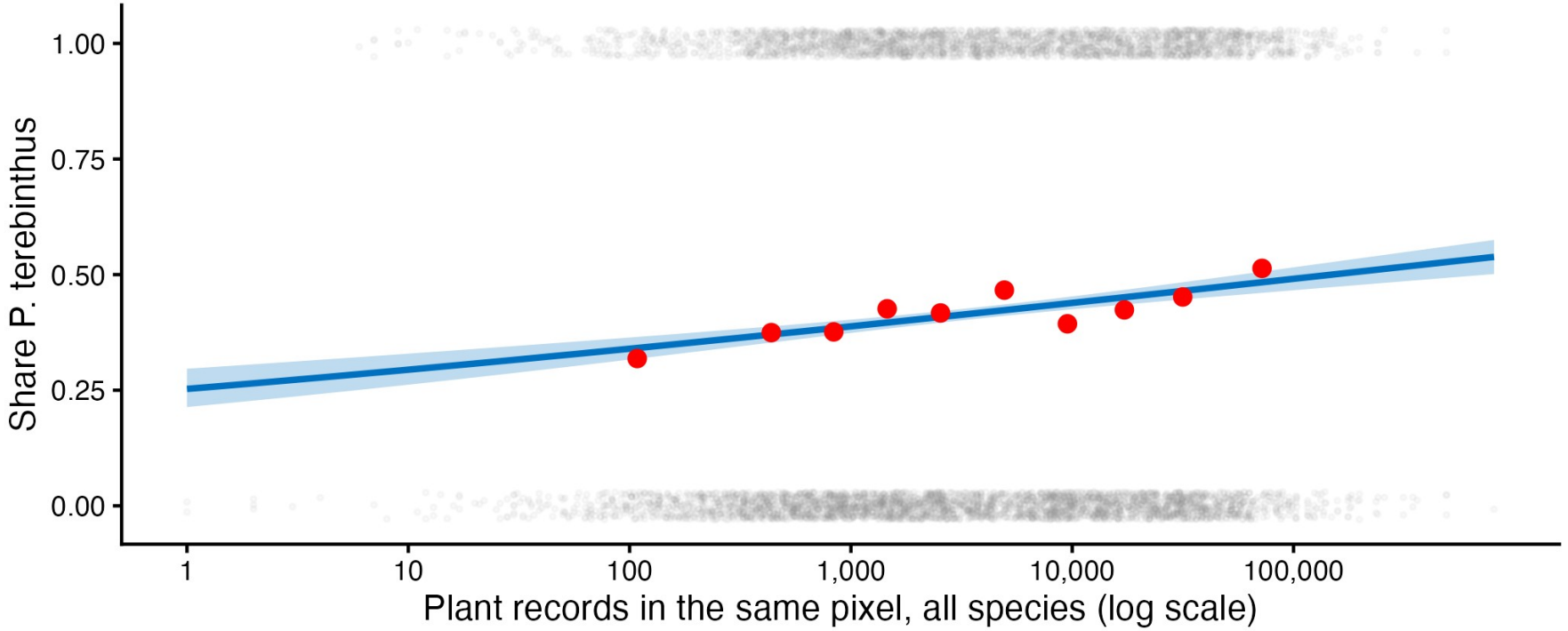
bio6 is predicted almost perfectly by the other 8 together: **VIF = 1 / (1 − R^2)**, far above 5

Drop the worst VIF, refit, repeat



18 → 7 variables: bio1, bio3, bio4, bio8, bio9, bio18, bio19, every VIF under 5

Sampling effort: a control we keep



effort goes in every model: dredge(full, fixed = ~ effort)

dredge(): fit every subset



7 climate variables, each in or out:

2^7 = 128 models

Each row = one model

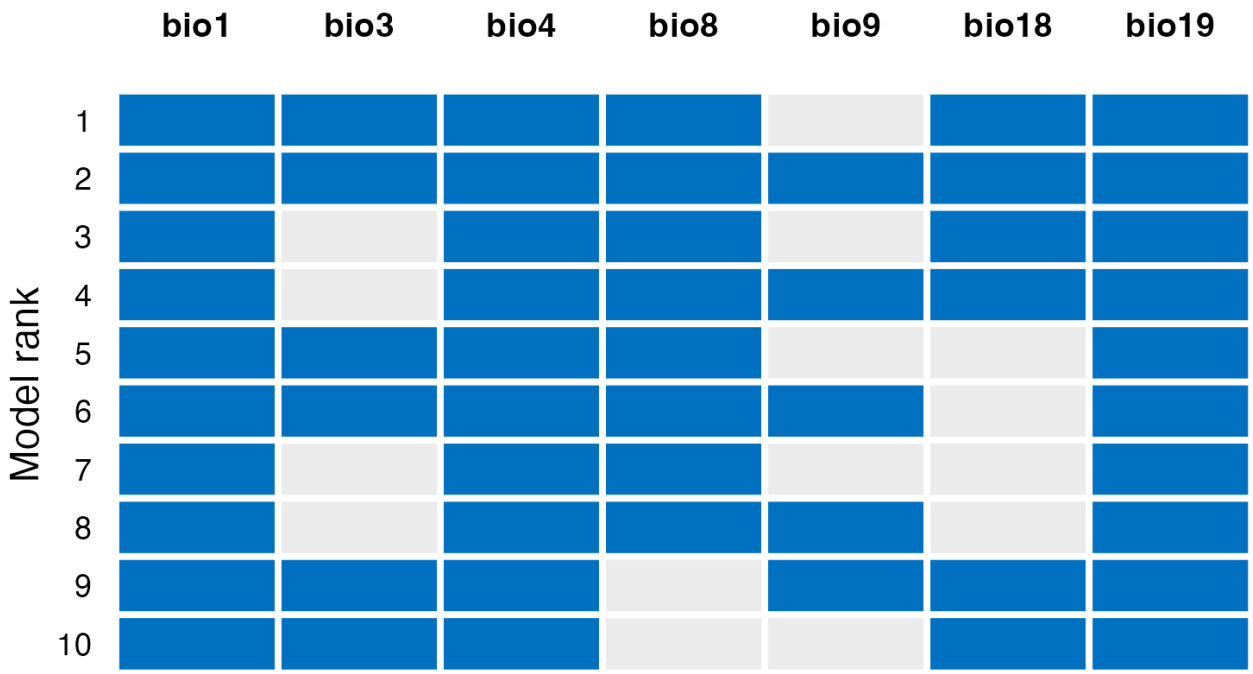
blue = variable in it

ranked best (top) to worst

```
dredge(full, fixed = ~ effort)
```

effort is in all 128 (fixed). Only the climate variables are searched.

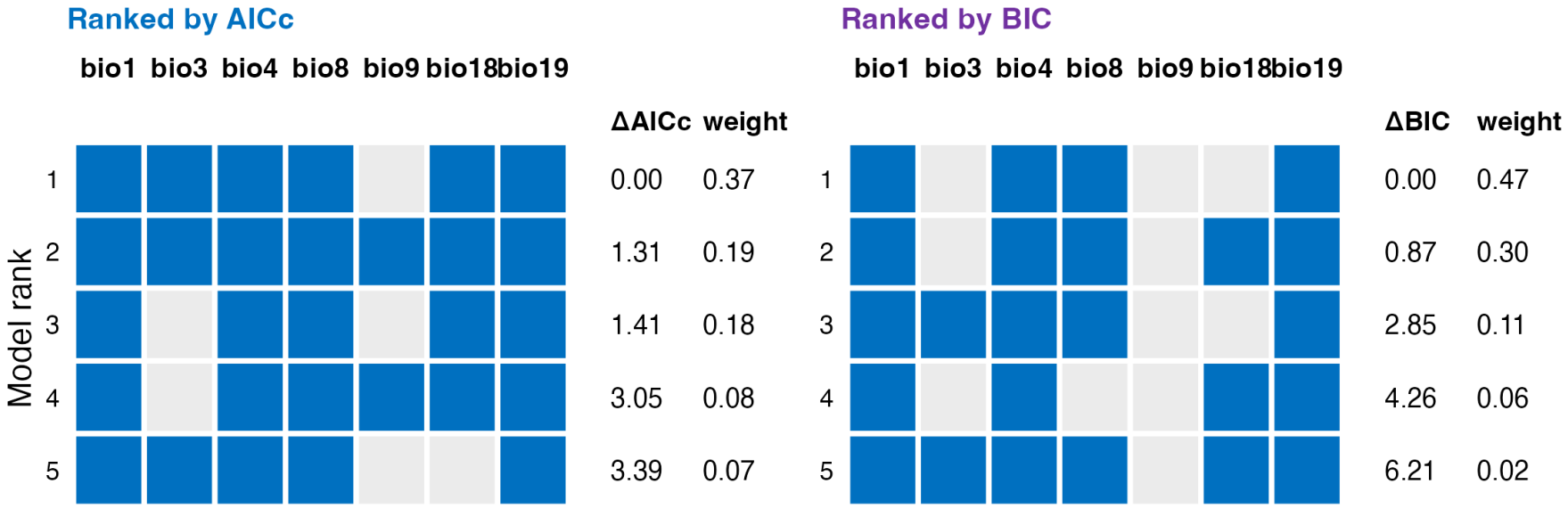
The top 10 of 128



	ΔAICc	weight
1	0.00	0.37
2	1.31	0.19
3	1.41	0.18
4	3.05	0.08
5	3.39	0.07
6	3.51	0.06
7	7.28	0.01
8	7.82	0.01
9	8.01	0.01
10	9.19	0.00

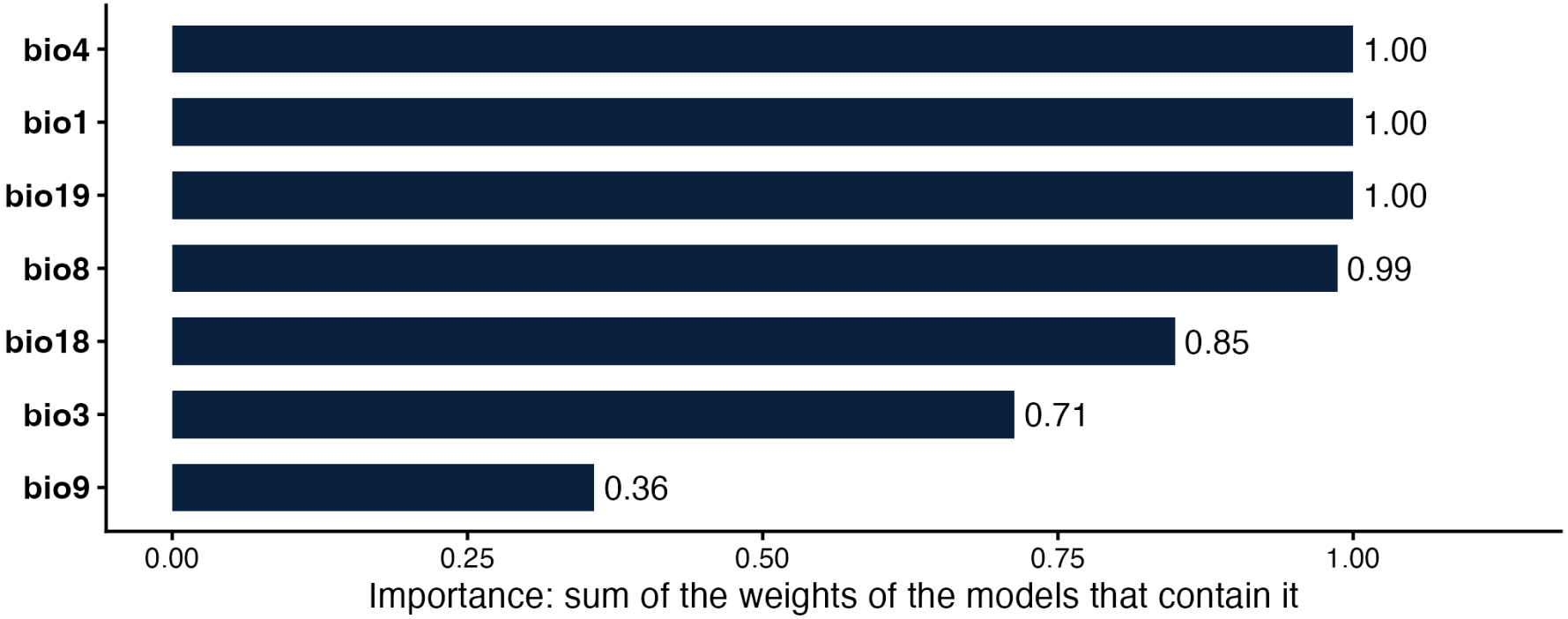
3 models within ΔAICc 2; the top one has only **37% of the weight**

AICc vs BIC with 6,299 records



BIC charges 8.7 per parameter here: a smaller top model

Importance: summed weights



1.00 = in every model with real support. Not effect size, not a p-value.

Live poll

Multiple choice

bio4 has importance 1.00. What does that tell you?

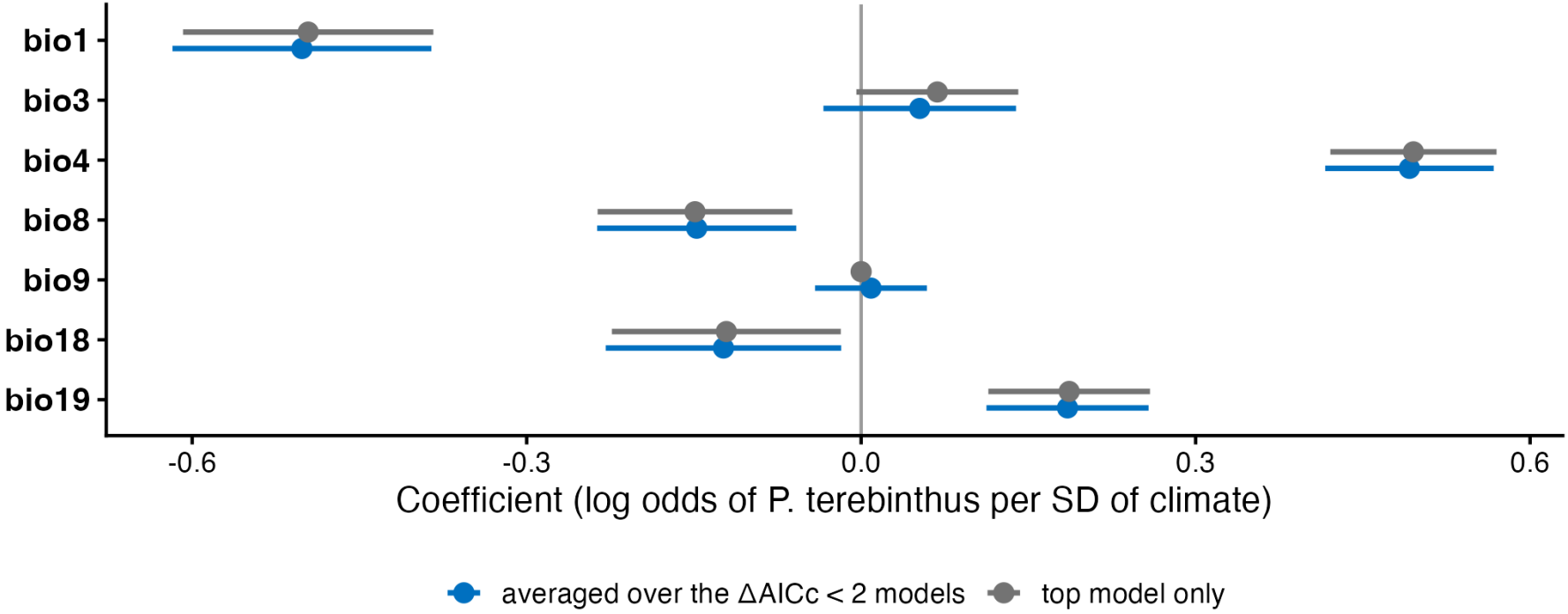
A Its effect is large
B It is in every well-supported model
C Its p-value is below 0.05
D It causes the difference between the species

Scan with your phone

Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w5-importance

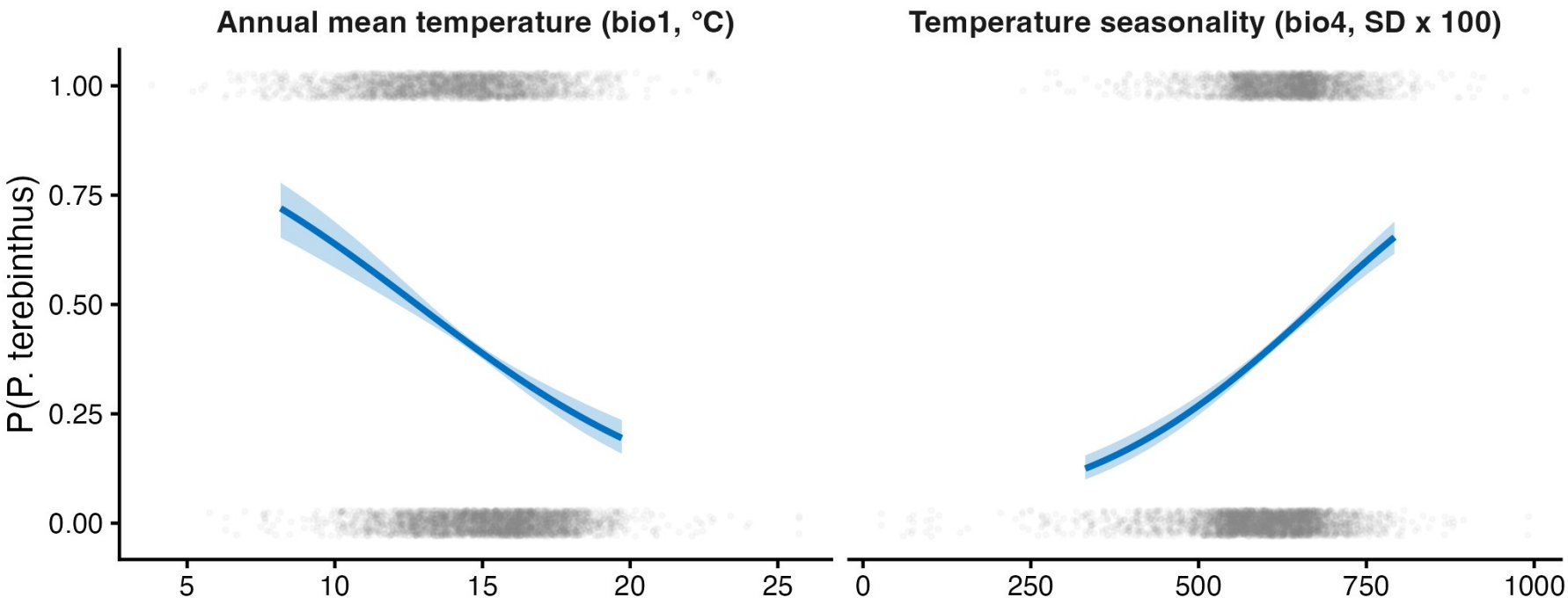
host screen

Model averaging: estimates across the top models



Averaged estimates **shrink** terms that are only in some of the models

What we found: hypotheses, not results



Terebinthus: cooler, more seasonal places. **Test it on new data.**

1 Too many models

2 Scoring models

3 Testing ideas

4 Exploring

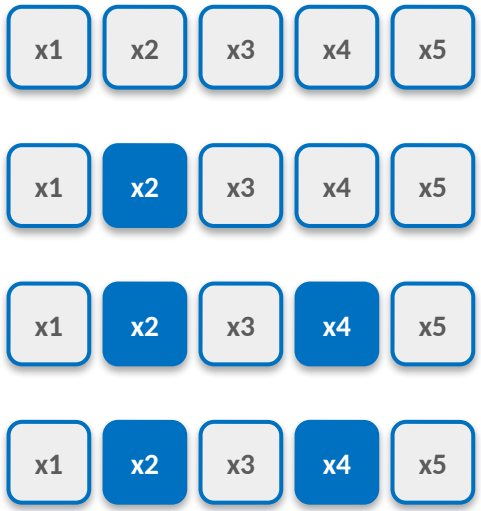
▶ **5 Traps**

6 Predicting

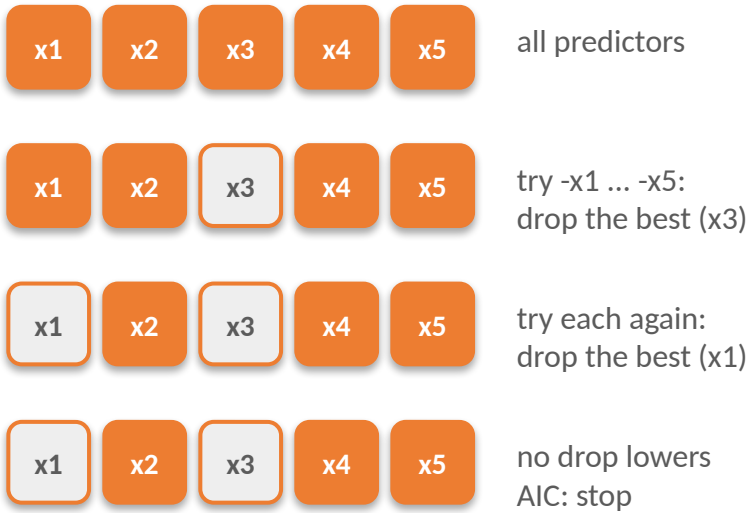


Stepwise selection: the recipe

Forward: start with none, add

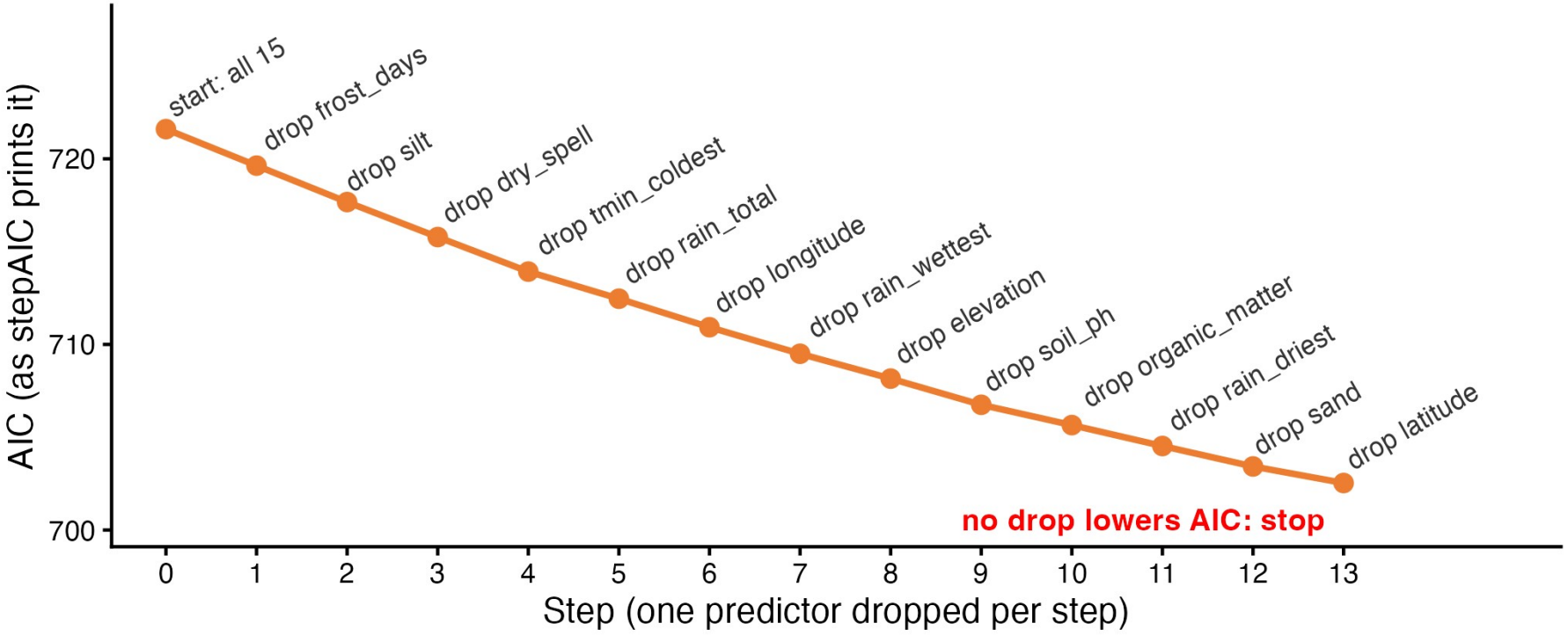


Backward: start with all, drop



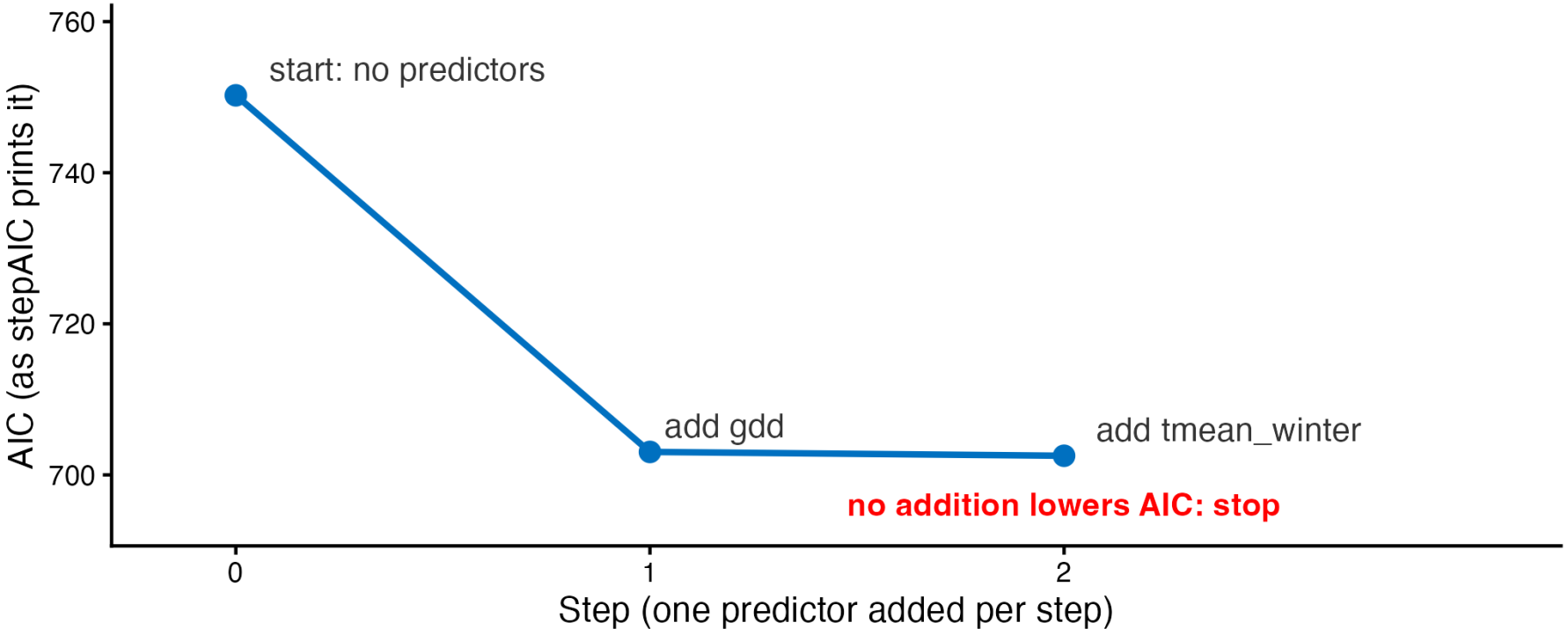
Each step: try every single change, **keep the one that lowers AIC most**. Stop.

Stepwise selection: drop one at a time



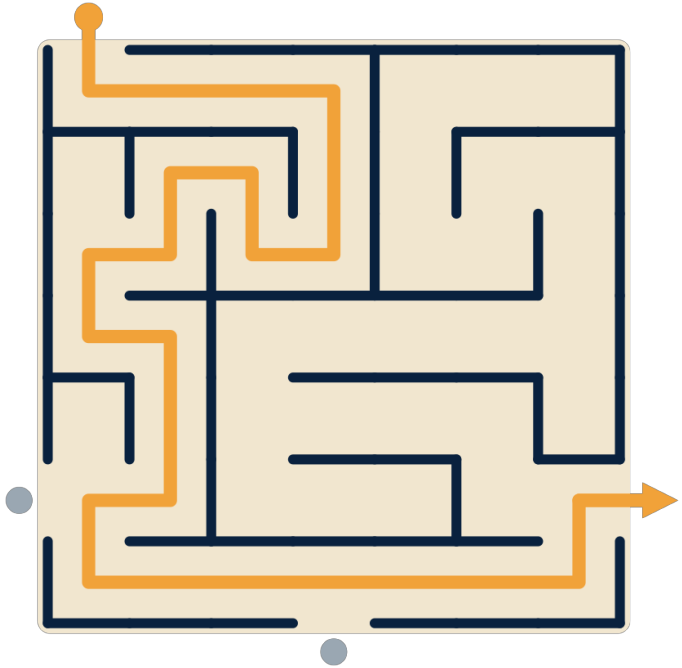
Start with all 15. **Drop whichever lowers AIC most**. Repeat until no drop helps.

Forward on the same farms



Add gdd, then tmean_winter, stop. **Same model as backward here**; often they differ.

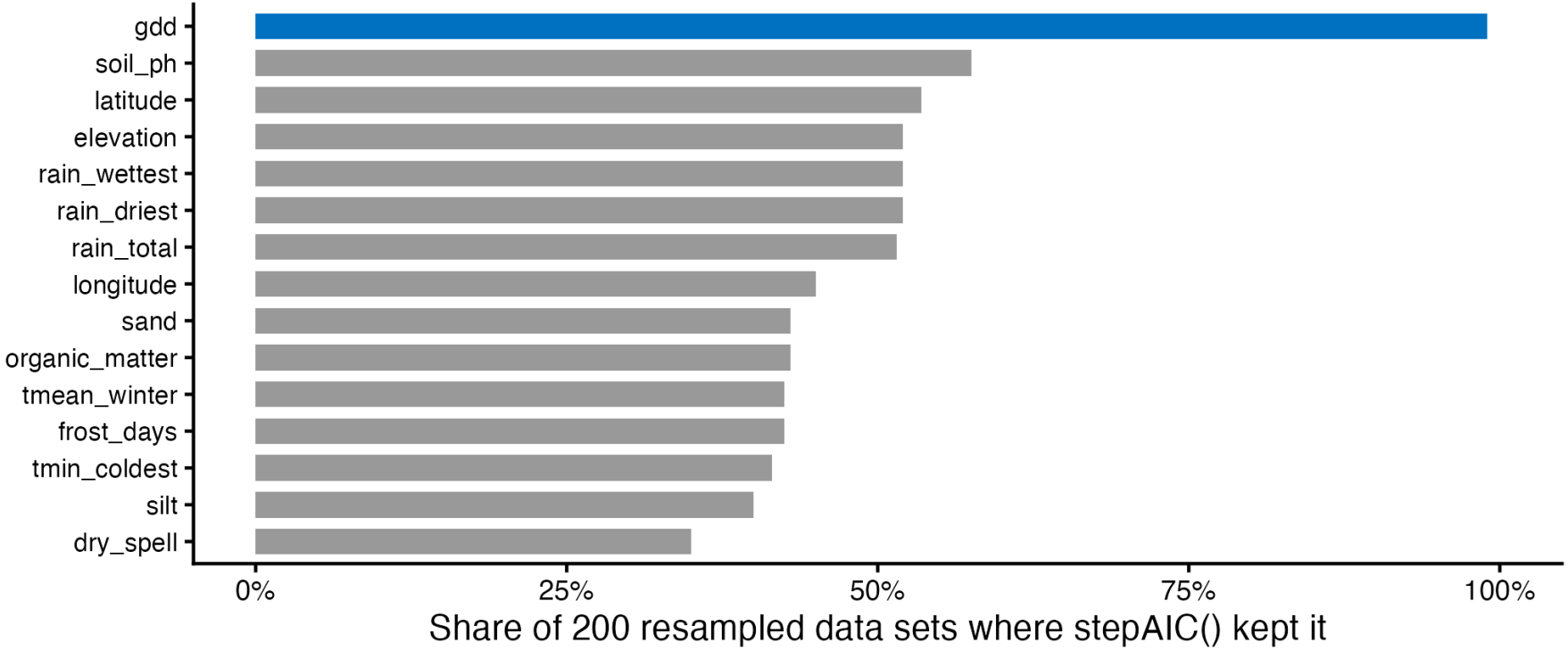
Stepwise: one path through the maze



```
> full_f <- lm(biomass ~ ., data = farms)
> step_f <- stepAIC(full_f, trace = 0)     # MASS
> attr(terms(step_f), "term.labels")   # what it kept
[1] "gdd"          "tmean_winter"  ← junk kept
```

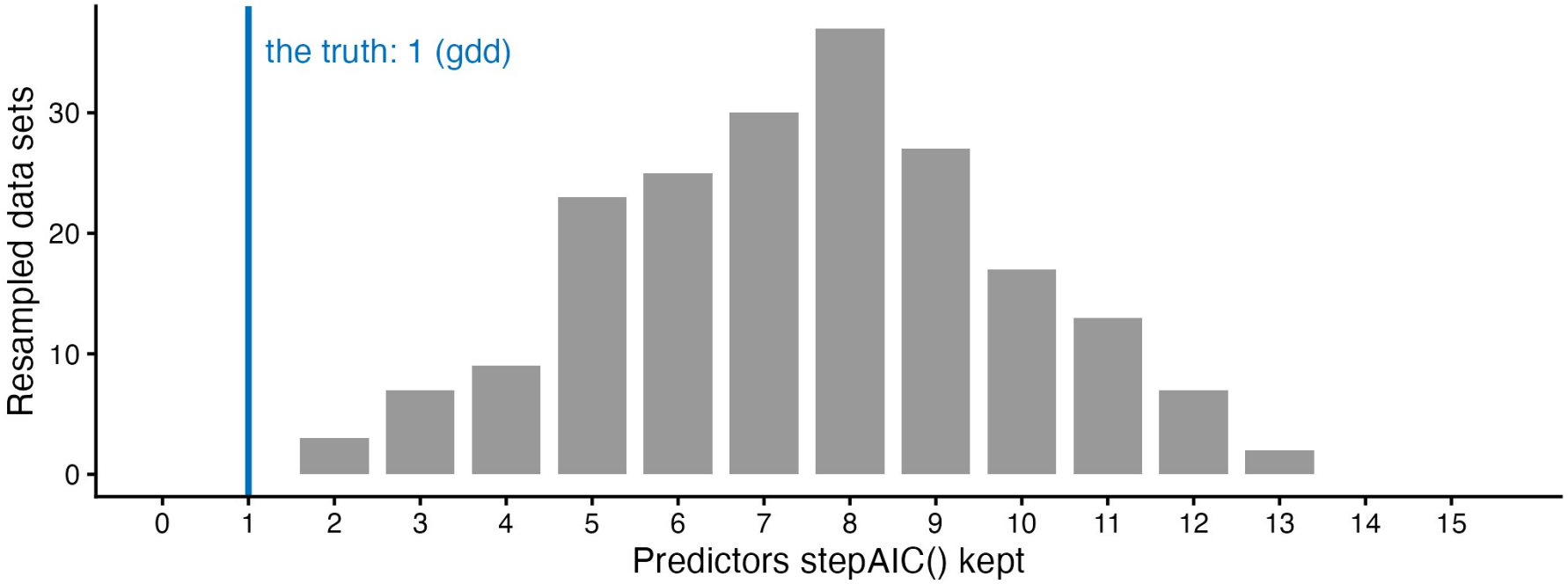
Kept gdd and **tmean_winter** (which does nothing)

Resample the farms, rerun stepAIC



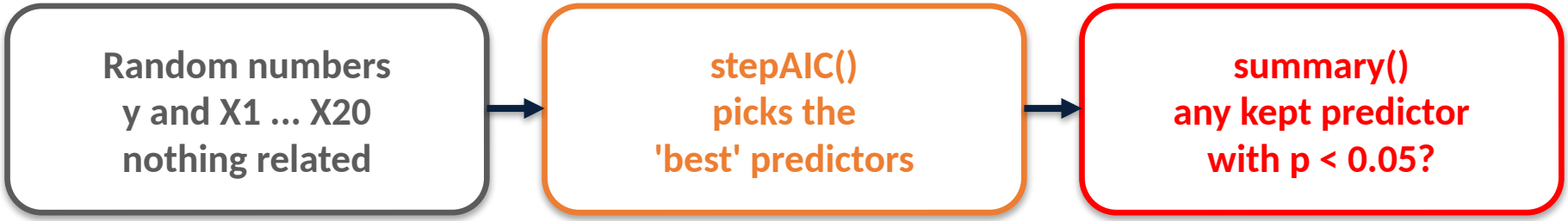
gdd: 99%. Every junk variable: 35% to 57% of the time

How many did it keep? The truth is 1



Median **8 predictors**. gdd alone: 0 times out of 200

An experiment: selection on pure noise




```
noise <- data.frame(y = rnorm(100), matrix(rnorm(100 * 20), 100))
step_n <- stepAIC(lm(y ~ ., data = noise), trace = 0)
summary(step_n)
```

[Run it yourself: greymonroe.github.io/PLS_206/Week 5/Week5_noise_experiment.R](greymonroe.github.io/PLS_206/Week 5/Week5_noise_experiment.R)

Every true slope is **zero**. Any 'significant' predictor is a false positive.

Pure noise, 'significant' results

```
> set.seed(41)
> noise <- data.frame(y = rnorm(100), matrix(rnorm(100 * 20),
100))
> step_n <- stepAIC(lm(y ~ ., data = noise), trace = 0)
> round(summary(step_n)$coefficients, 3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.211      0.097   2.183    0.032
X1            -0.200      0.107  -1.867    0.065
X11           -0.269      0.096  -2.813    0.006  ← noise!
X17            0.165      0.093   1.782    0.078
X18            0.131      0.088   1.488    0.140
X20           -0.367      0.107  -3.445    0.001  ← noise!
```

p = 0.006 and **p = 0.001** for columns of random numbers

Live poll

Number: your guess, live histogram

Now repeat that experiment 1,000 times, with fresh random numbers each time.

In what % of runs does at least one kept predictor come out 'significant' (p < 0.05)?

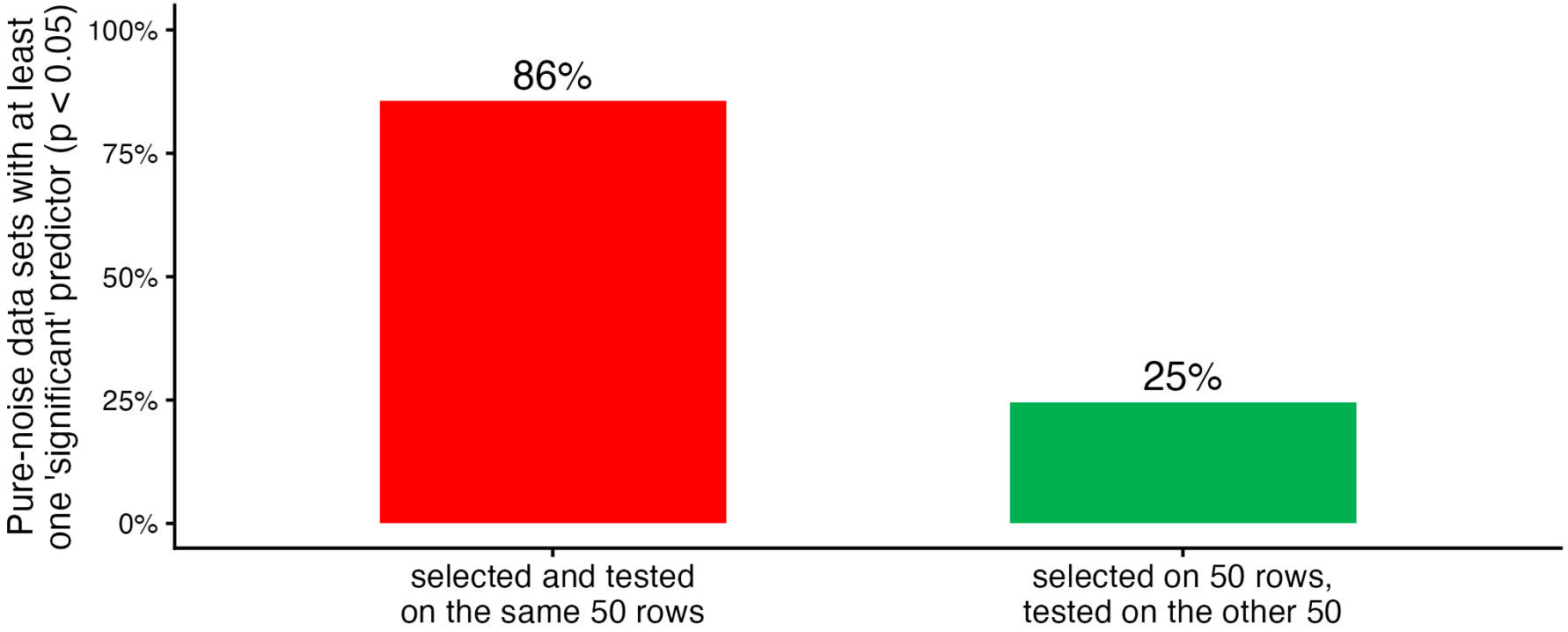


Scan with your phone

[Join on your phone or laptop: greymonroe.github.io/live-poll-test/submit.html?poll=pls206-w5-noise]

[host screen]

Fix: choose on one half, test on the other



86% with the same rows; **25%** on fresh rows, about what chance gives

Selection finds predictors, not causes

```
> m_lat <- lm(biomass ~ latitude, data = farms)
> round(as.data.frame(model.sel(m_null, m_lat))[, c("df",
"AICc", "delta", "weight")], 3)
       df    AICc delta weight
m_lat   3 887.453 0.000   0.97  ← weight
m_null  2 894.391 6.938   0.03
```



Latitude gets **97% of the weight** vs the null, but only through gdd

1 Too many models

2 Scoring models

3 Testing ideas

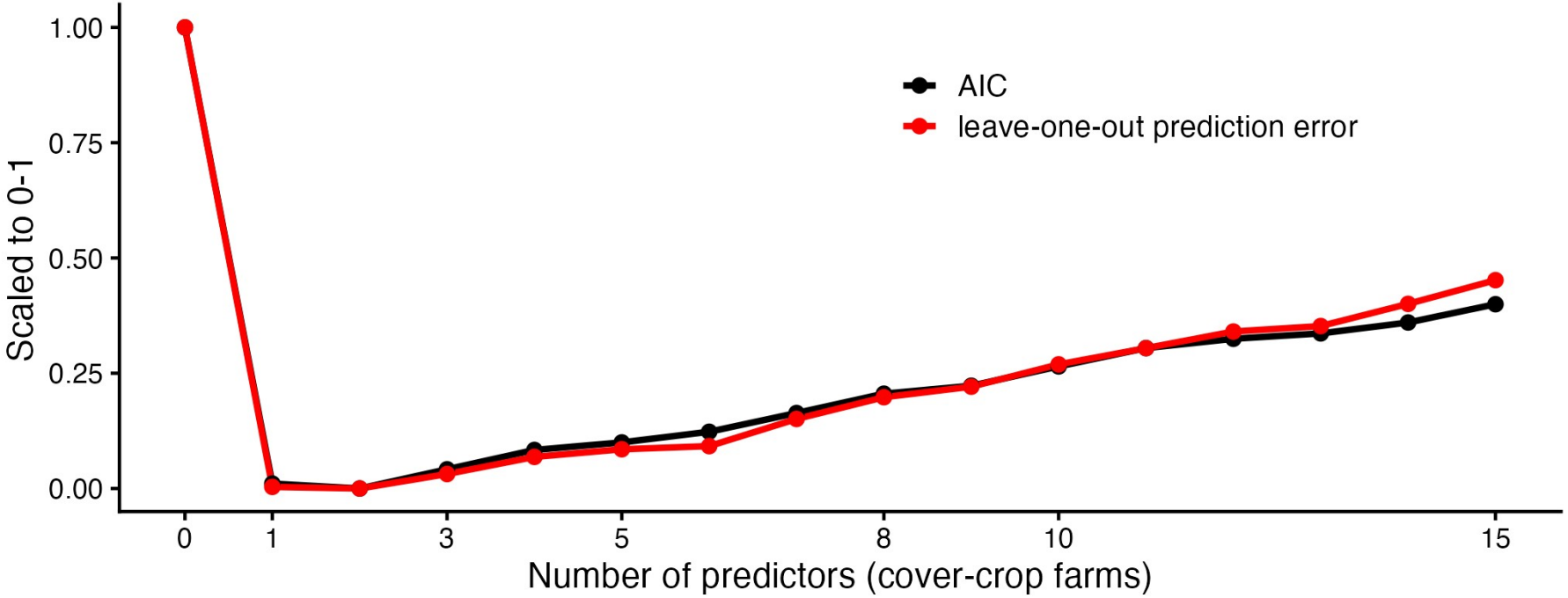
4 Exploring

5 Traps

▶ **6 Predicting**



AIC approximates leave-one-out prediction



AIC and **leave-one-out error** pick the same model on the cover-crop farms

Next week: shrink instead of choose

This week	**each predictor in or out**
Next week	**keep them all, shrink their slopes (ridge, lasso)**
Both	**judged on data the model did not see**

Lasso shrinks some slopes to exactly zero: selection without the 32,768 models

Which tool for which goal

Test ideas	**write models first; model.sel(); report the table**
Explore	**dredge(); results are hypotheses**
Predict	**held-out data, cross-validation**
Always	**same rows, same response**
Always	**keep design terms and controls**
Avoid	**stepwise; p-values after selection**

Problem Set 5 due Monday 2 November, 1:10 PM

Live coding: explore, then check



`Week5_live_2_explore.R`

Part 2

1 Pistacia: bio7, VIF
2 dredge(), importance, averaging
3 stepAIC on resampled farms
4 Pure noise
5 AIC vs leave-one-out