

Ulysses' Compass: Regularization

EES 4891-06/5891-01

Bayesian Statistical Methods

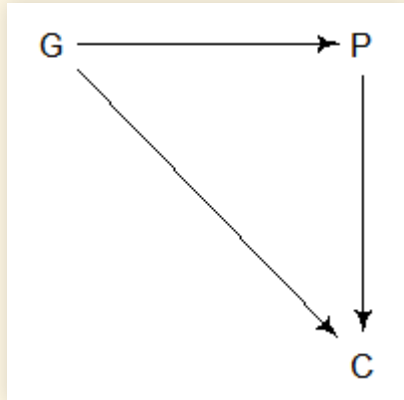
Jonathan Magnolia Gilligan

Class #9: Wednesday February 11, 2026

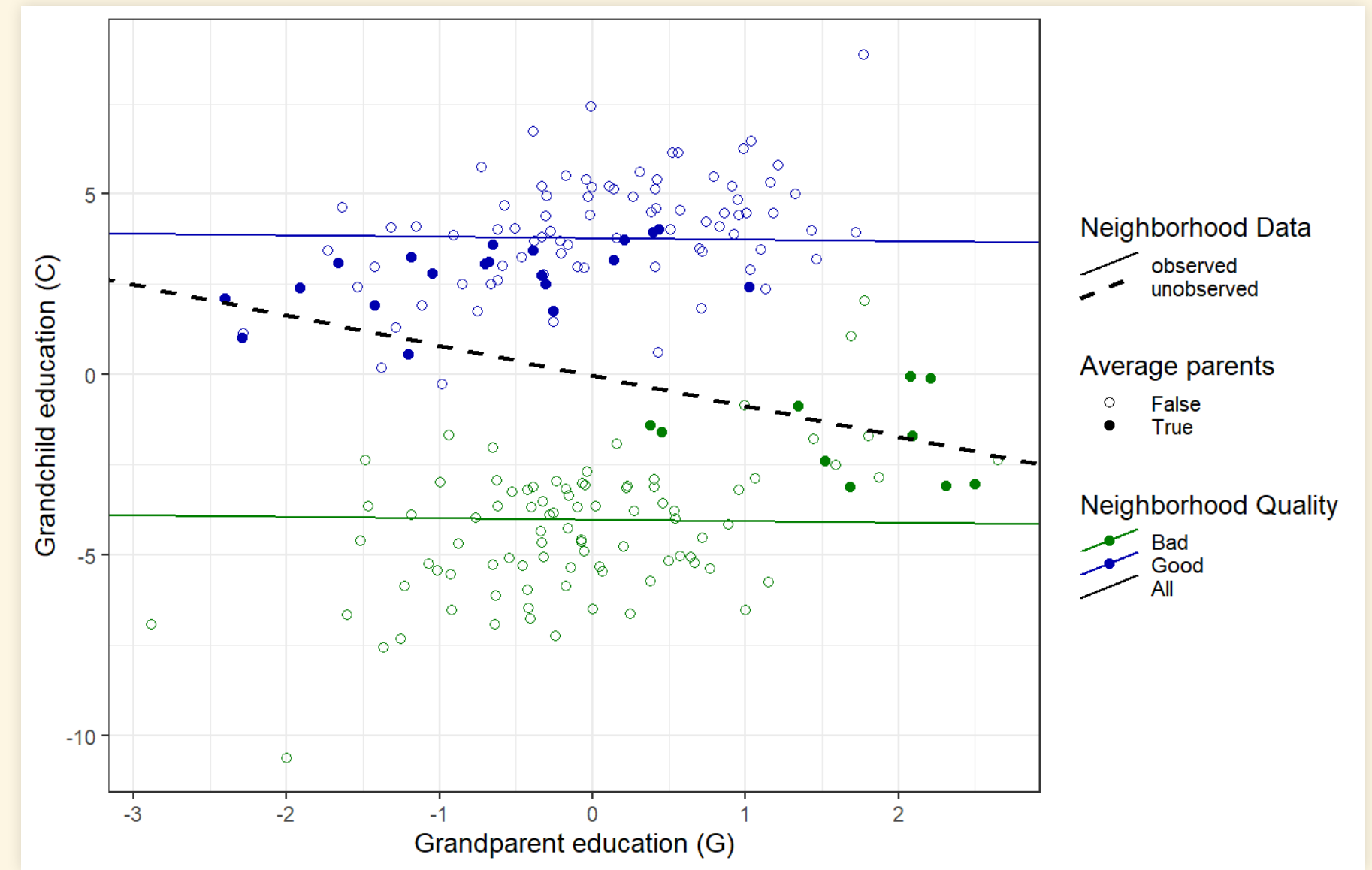
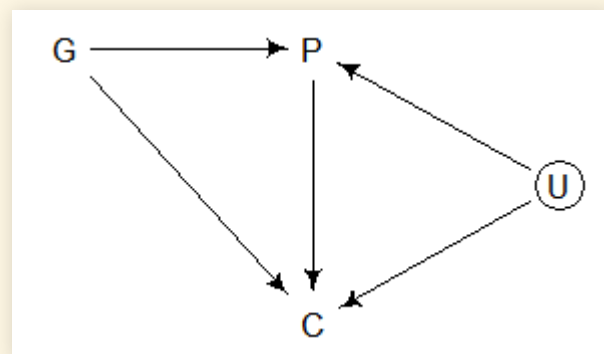
Examples of confounders

Example: Haunted DAG

- How do parents' P and grandparents' G educational attainment influence educational attainment of children C ?



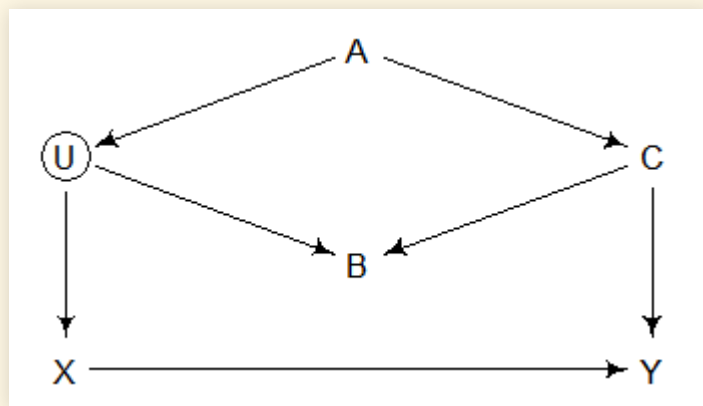
- But there are unmeasured effects here, such as the character of the neighborhood.
 - Grandparents moved into the neighborhood after they finished school,
 - Parents and children grew up in the neighborhood and are affected by it.



- There is a no correlation between G and C in each neighborhood
 - This is the correct answer.
- But when we don't account for the neighborhood effect, the collider bias makes it look like there's a negative correlation
 - more educated grandparents have less educated grandchildren

Backdoor Effects

- In the age and happiness example, conditioning on the marriage variable created bias,
- But in the grandparent, parent, and children example, we needed to condition on the neighborhood to avoid bias.
 - How can we tell when to condition on a variable?
- Consider this DAG:



- How does X affect Y ?

- Backdoor paths:
 1. $X \leftarrow U \leftarrow A \rightarrow C \rightarrow Y$
 2. $X \leftarrow U \rightarrow B \leftarrow C \rightarrow Y$
- Which backdoor path is open?
 1. This path is open because it has no internal collider
 2. This path is closed because B is a collider.
 - If we condition on B , it will open the backdoor and introduce a collider effect.
- Closing backdoors:
 - We don't observe U , so we can't condition on it.
 - To close the backdoor path #1, condition on A or C .

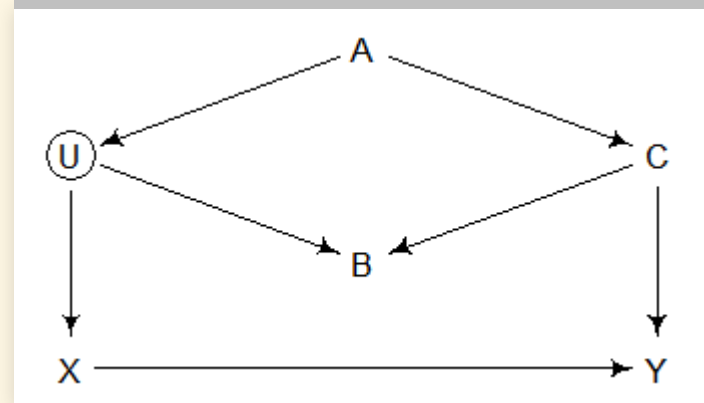
Automated Analysis

- Define the DAG

```
library(dagitty)
dag_two_roads <- dagitty("dag {
  U [unobserved]
  X -> Y
  X <- U <- A -> C -> Y
  U -> B <- C
}")
```

- Optionally, draw the DAG diagram

```
coordinates(dag_two_roads) <- list(
  x = c(U = 0, X = 0, A = 1, B = 1, C = 2, Y = 2),
  y = c(U = 0, X = 1, A = -0.5, B = 0.5, C = 0, Y = 1)
)
drawdag(dag_two_roads)
```



- Analyze to identify which variables to condition on

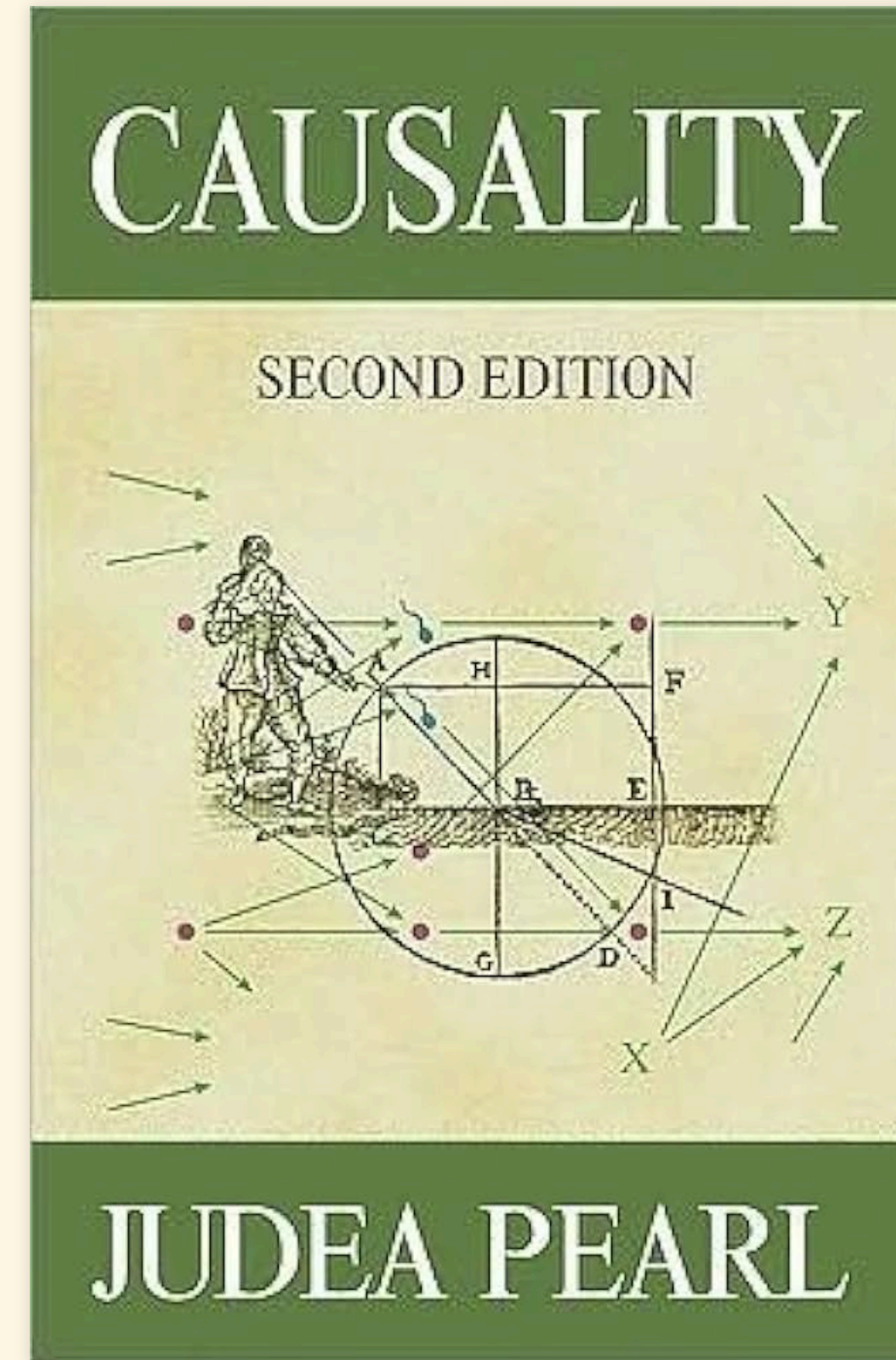
```
adjustmentSets(dag_two_roads, exposure = "X", outcome = "Y")
```

```
## { C }
## { A }
```

- Condition on *A* or *C*

Further Reading

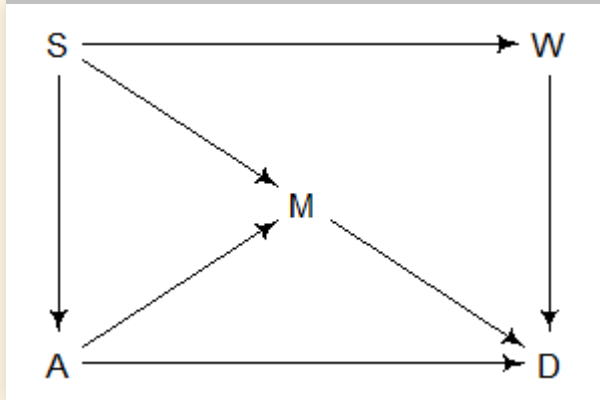
- The textbook's use of DAGs to represent causal relationships comes from
 - Judea Pearl, *Causality: Models, Reasoning, and Inference*



Backdoors in Waffle-House and Divorce

- Waffle-House and Divorce

```
dag_waffles <- dagitty("dag {  
A -> D  
A -> M -> D  
A <- S -> M  
S -> W -> D  
}")
```



S = state, W = waffle-house restaurants,
 A = median age at marriage, M = marriage rate,
and D = divorce rate.

- Identify which variables to condition on

```
adjustmentSets(dag_waffles, exposure="W", outcome="D")
```

```
## { A, M }  
## { S }
```

- What does this mean?

- Backdoors:

- $W \leftarrow S \rightarrow M \rightarrow D$
- $W \leftarrow S \rightarrow A \rightarrow D$
- $W \leftarrow S \rightarrow A \rightarrow M \rightarrow D$

- All of these pass through S .
- To close the backdoors, either
 - Condition on S , or
 - Condition on both A and M .

- Further analysis: *conditional independencies*

```
impliedConditionalIndependencies(dag_waffles)
```

```
## A _||_ W | S  
## D _||_ S | A, M, W  
## M _||_ W | S
```

- If we condition on S , then A and M should both be independent of W
- If we simultaneously condition on A , M , and W , then D should be independent of S .

Bayes's Theorem and Ockham's Razor

Bayes's Theorem and Ockham's Razor

Everything should be made as simple as possible, but no simpler
— Einstein

- **Ockham's razor:** *Models with fewer hypotheses are to be preferred*
 - But we also prefer models that make better predictions
- How do we find the right balance between simplicity and completeness?
 - *Overfitting versus underfitting*
 - *Confounding:* Incorrect causal relationships can produce better predictions.
- Bayesian methods allow us to take a systematic formal approach to finding the best balance,
 - But we need some additional tools
- Tools for finding a good balance:
 - **Regularization:** Use a *regularizing prior* to avoid overfitting
 - Also called *penalized likelihood*.
 - **Cross-validation and information criteria**
 - **Cross-validation:** Fit parameters to part of your data and predict the rest.
 - **Information criteria:** Use *information theory* to measure how much value additional complexity adds.

Overfitting

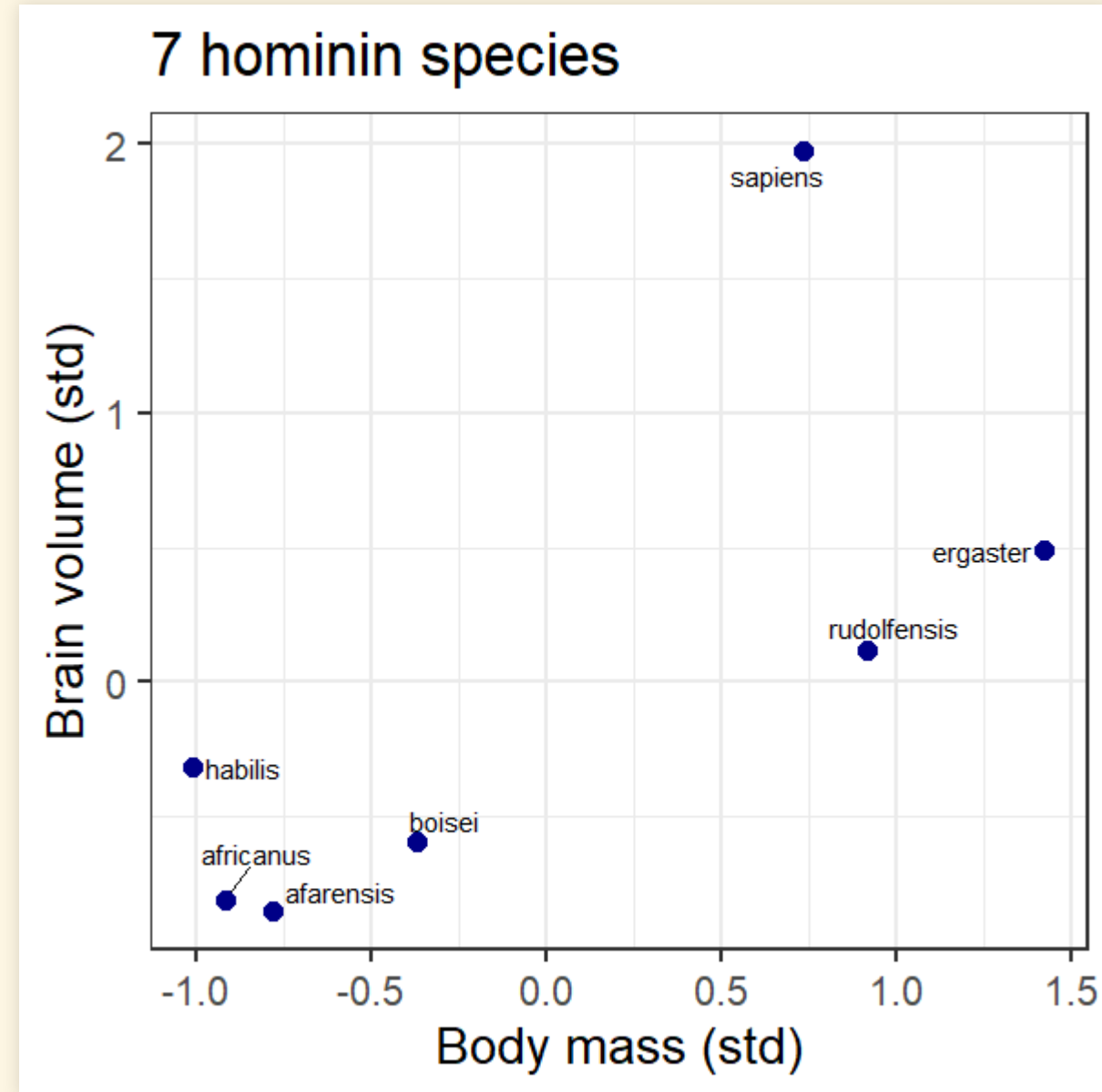
- Correlation: R^2

$$R^2 = \frac{\text{var}(\text{data}) - \text{var}(\text{residuals})}{\text{var}(\text{data})} = 1 - \frac{\text{var}(\text{residuals})}{\text{var}(\text{data})}$$

- R^2 increases the more parameters you add. Favors extreme overfitting.
 - Adding parameters almost always improves fit to current data
 - **Overfitting** happens when improving the fit to current data makes predictions of new data worse.

Example

- Data: relationship between brain volume and body mass for 7 hominin species.



- Fit 6 models:

1. $\mu = \alpha + \beta M$

2. $\mu = \alpha + \beta_1 M + \beta_2 M^2$

3. $\mu = \alpha + \beta_1 M + \beta_2 M^2 + \beta_3 M^3$

4. $\mu = \alpha + \beta_1 M + \beta_2 M^2 + \beta_3 M^3 + \beta_4 M^4$

5. $\mu = \alpha + \beta_1 M + \beta_2 M^2 + \dots + \beta_5 M^5$

6. $\mu = \alpha + \beta_1 M + \beta_2 M^2 + \dots + \beta_6 M^6$

- Quality of fit:

1. Model # 1 : $R^2 = 0.486$

2. Model # 2 : $R^2 = 0.538$

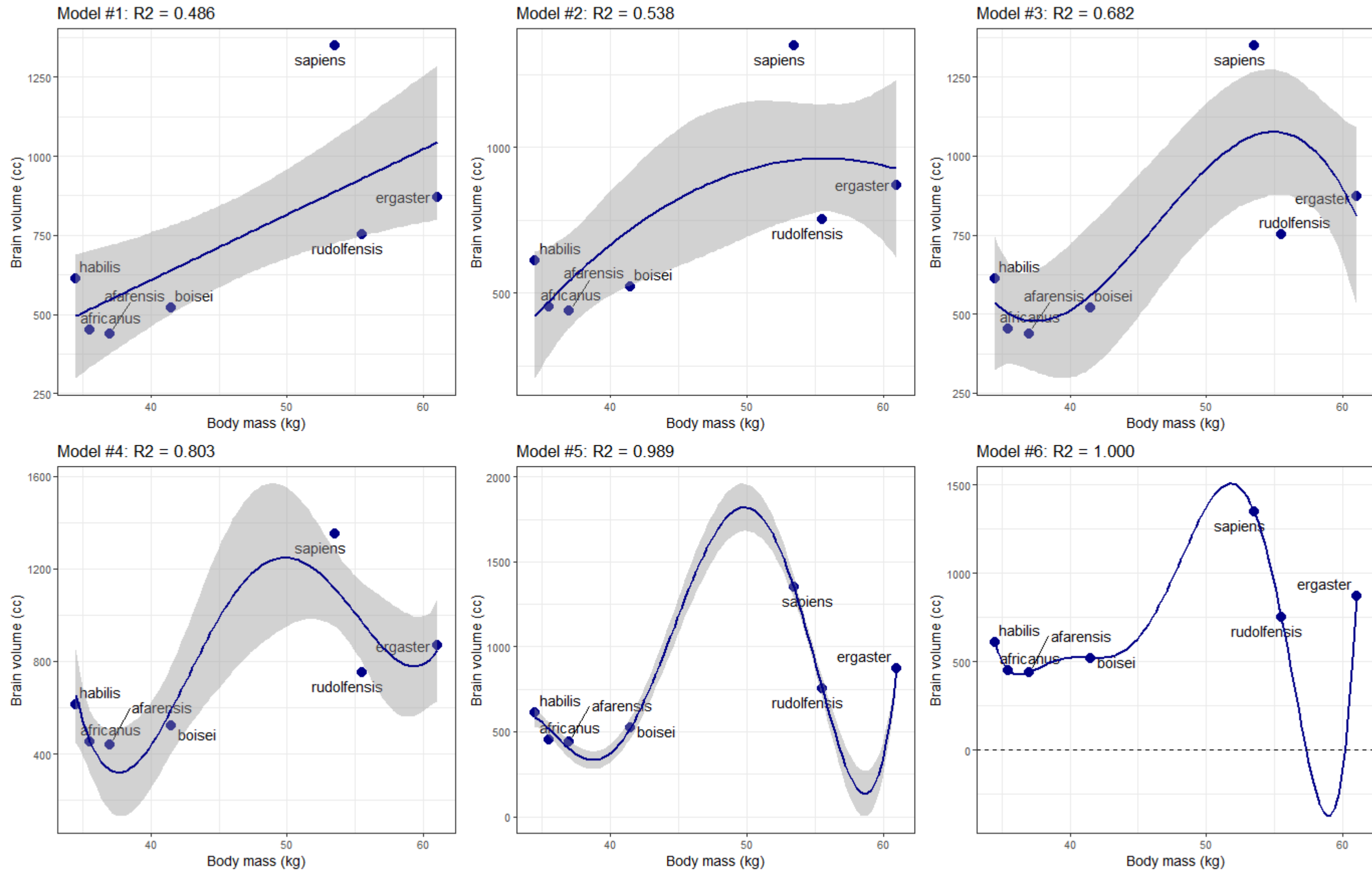
3. Model # 3 : $R^2 = 0.682$

4. Model # 4 : $R^2 = 0.803$

5. Model # 5 : $R^2 = 0.989$

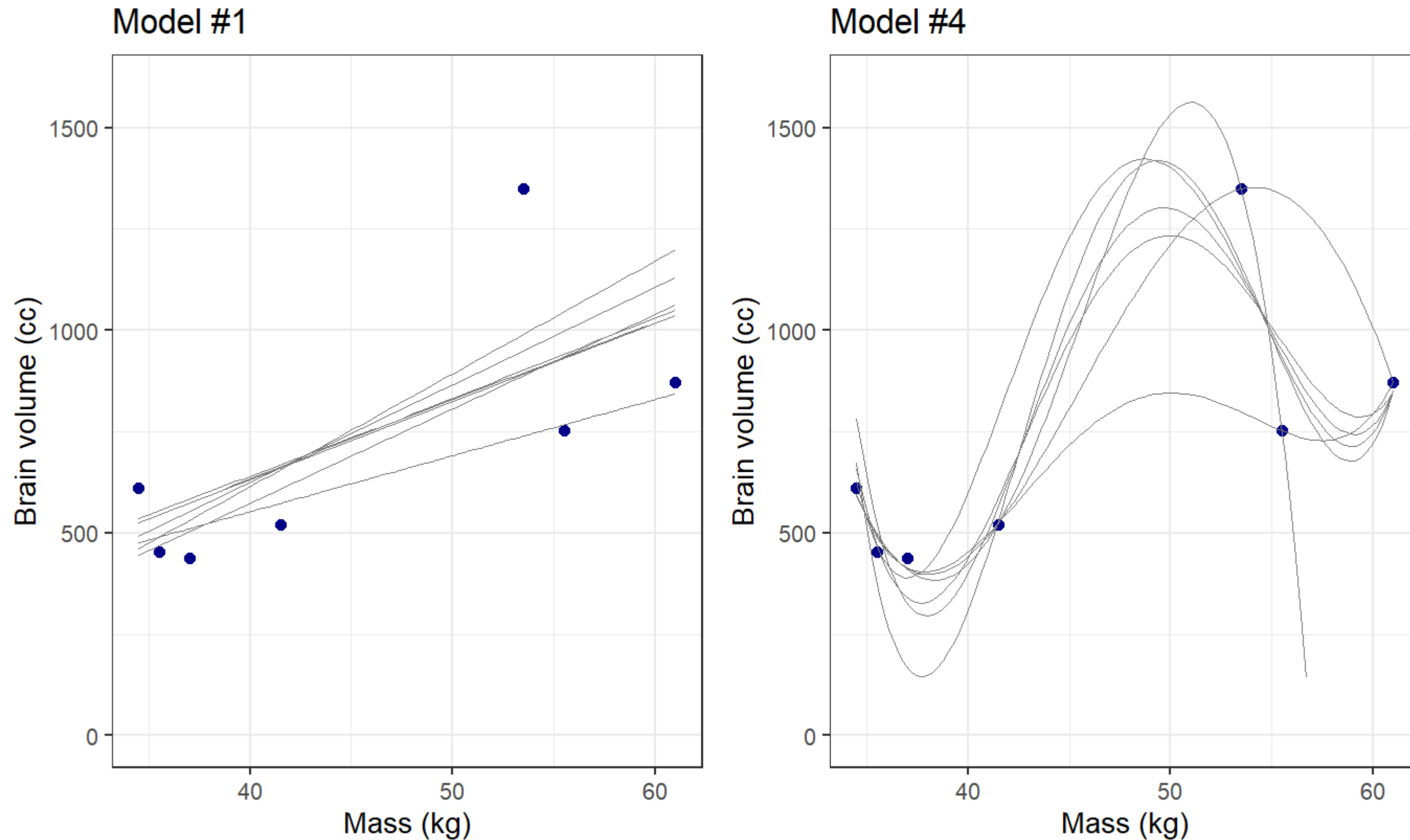
6. Model # 6 : $R^2 = 1.000$

Examining models



Underfitting vs. Overfitting

- Sensitivity to dropping one measurement:



Information Theory

Entropy & Accuracy

- There isn't a universal, objective standard for judging the best balance of overfitting and underfitting.
- But there are systematic procedures for arriving at the best balance
 1. Pick a **target**: what you want the model to do well at.
 2. Develop a measurement of **deviance**: How close does model come to the *target*?
- We use *information theory* to develop a systematic way of measuring *deviance*.
 - What will matter is only the *deviance* of *out-of-sample* predictions.

Assessing Accuracy

- Weather prediction:
 - On average, it rains 30% of the time and is sunny the rest of the time.
 - A naive model would predict 30% chance of rain every day
 - This model would be *well-calibrated* but fairly useless.
 - A model that says it never rains will be correct 70% of the time.
- Do we care more about some kinds of errors than others?
 - It's worse not to have an umbrella when it rains than to carry one and not need it.
- **Joint likelihood:** probability of getting day 1 right *and* day 2 right *and* day 3 right ...

$$P(\text{day}_1 \ \& \ \text{day}_2 \ \& \ \text{day}_3 \ \& \ \dots \ \text{day}_N) = P(\text{day}_1) \times P(\text{day}_2) \times P(\text{day}_3) \times \dots \times P(\text{day}_N)$$

- **Log scoring rule:**

$$\log(P(\text{day}_1 \ \& \ \text{day}_2 \ \& \ \text{day}_3 \ \& \ \dots \ \text{day}_N)) = \sum_{i=1}^N \log(P(\text{day}_i))$$

- So we use the log of the probability to score accuracy.

Information

- **Information** is the *reduction in uncertainty when we learn an outcome*.
 - **Information entropy**: If there are n possible events with probabilities $p_1 \dots p_n$, then

$$H(p) = - \sum_{i=1}^n p_i \log(p_i)$$

- \log is the *natural logarithm* (also called \ln)
- Example: if $p_{\text{sun}} = 0.7$ and $p_{\text{rain}} = 0.3$, then

$$H(p) = -(0.7 \log(0.7) + 0.3 \log(0.3)) = 0.61.$$

- In another place, where $p_{\text{rain}} = 0.01$ and $p_{\text{sun}} = 0.99$,

$$H(p) = 0.06.$$

- Because there is so much more certainty about the weather on an average day, you learn a lot less from new data.

Divergence

- *Information entropy* measures uncertainty about the world (data).
- *Divergence* compares what we know from data versus what our model predicts.
 - **Divergence:** is the *additional uncertainty in using probabilities from one distribution to describe another distribution*.
 - The uncertainty in using observed data to make predictions about new data
 - Kullback-Liebler divergence:

$$D_{\text{KL}} = \sum_i p_i (\log(p_i) - \log(q_i)) = \sum_i \left(\frac{p_i}{q_i} \right),$$

- p_i are the true probabilities, q_i are our model's estimates of the probabilities.
- If the model probabilities are correct, then $q_i = p_i$ and $D_{\text{KL}} = 0$.

Measuring divergence

- The point of making a model is that we don't know the true probabilities p_i , and we want to estimate them with the model's q_i , so how can we measure the divergence?
- We can't measure p_i , but we can still use divergence to compare two models q and r :

$$D_q = \sum_i p_i (\log(p_i) - \log(q_i))$$

$$D_r = \sum_i p_i (\log(p_i) - \log(r_i))$$

$$\begin{aligned} D_q - D_r &= \sum_i p_i (\log(p_i) - \log(q_i)) - p_i (\log(p_i) - \log(r_i)) \\ &= \sum_i p_i (\log(r_i) - \log(q_i)) \end{aligned}$$

Divergence and Entropy

- The difference in divergence between two models q and r is

$$D_q - D_r = \sum_i p_i (\log(r_i) - \log(q_i))$$

- We can approximate this as

$$S(r) - S(q),$$

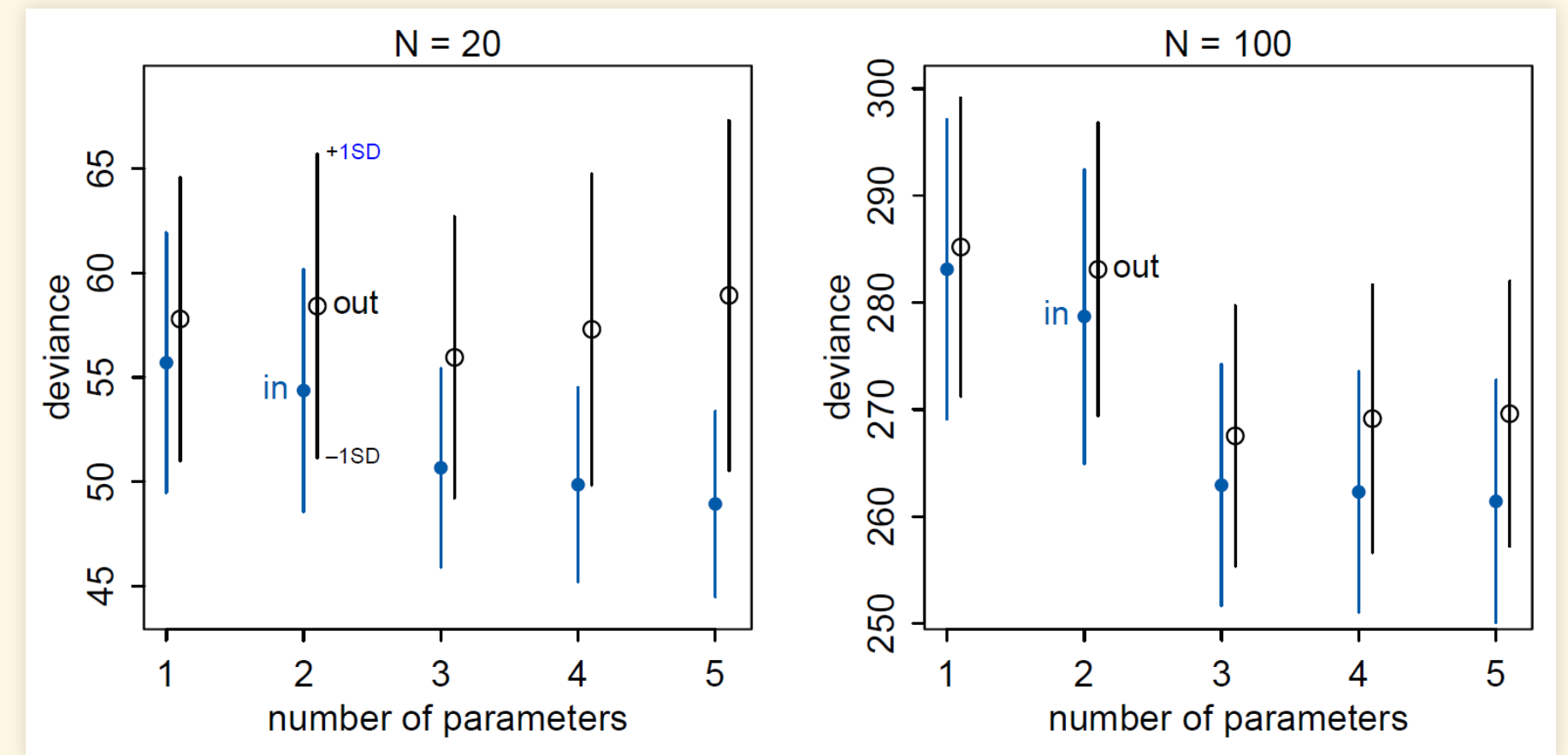
where

$$S(q) = \sum_i \log(q_i)$$

- Use the function `lppd()` from the `rethinking` package to calculate the **log-pointwise-predictive density** from a `quap` model. `lppd(md1, n = 1E4)` will calculate the log of the posterior probability of the model at 10,000 points. You can then use `sum()` to add these up and calculate the entropy $S(\text{md1})$
- We generally define **deviance** as $-2S(q)$.
 - Larger values of deviance are worse.

Using Entropy to Test Models

- *Training data vs. test data:*
 - Divide your data into two parts.
 - Use the *training data* to train your models
 - Use your models to predict the *test data*
 - Compare the KL-divergence of the models using the test data predictions.
- Example:
 - Generate data using a process with 3 parameters
 - *Training* set with N samples
 - *Test* set with N examples
 - Fit models with 1 to 5 parameters, using *training* data.
 - Measure deviance:
 - *In-sample* (training data)
 - *Out-of-sample* (test data)

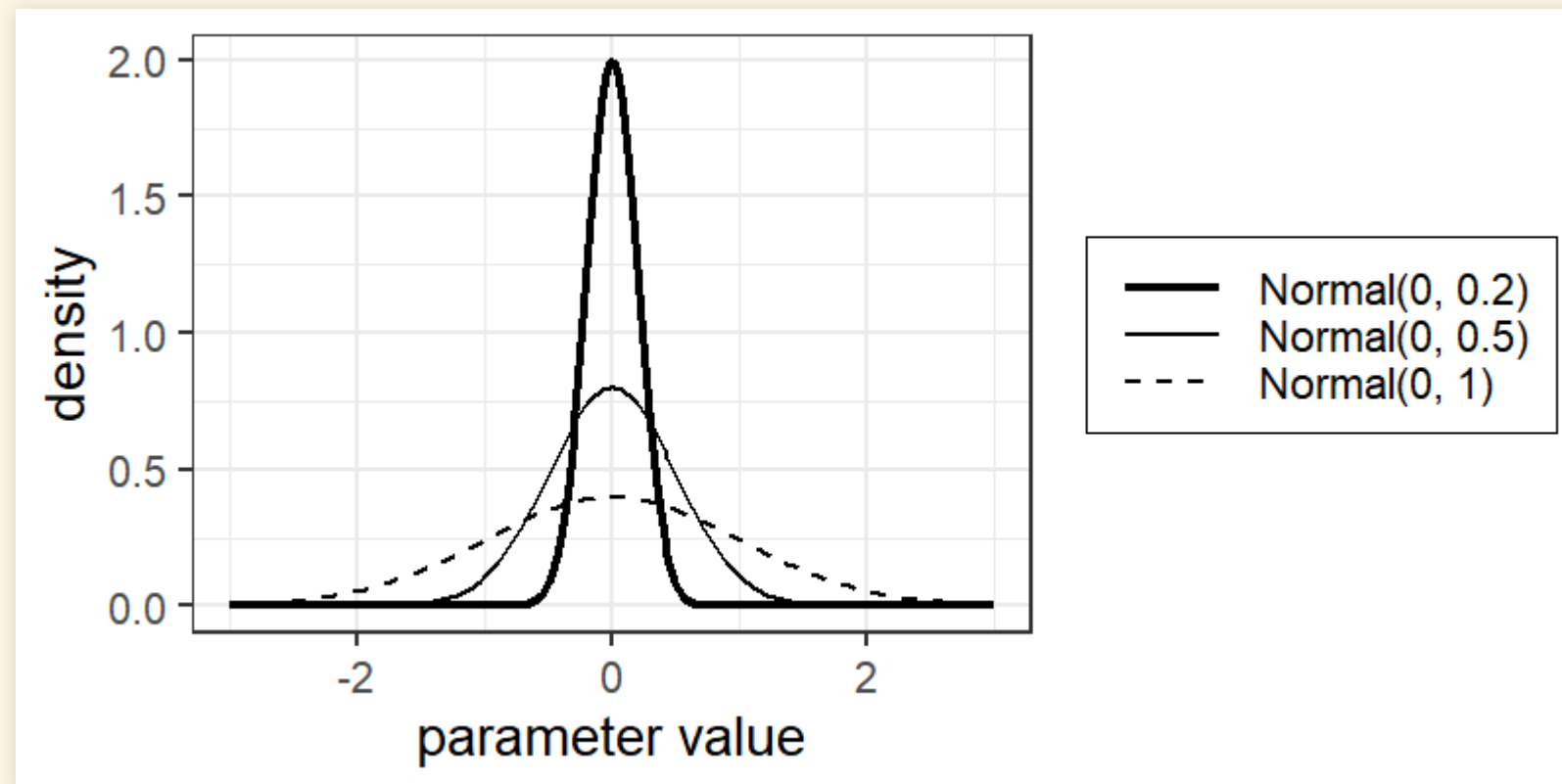


- The 5-parameter model has the smallest *in-sample* deviance.
- The 3-parameter model has the smallest *out-of-sample* deviance.
- Deviance estimates are more reliable for larger N (number of measurements).

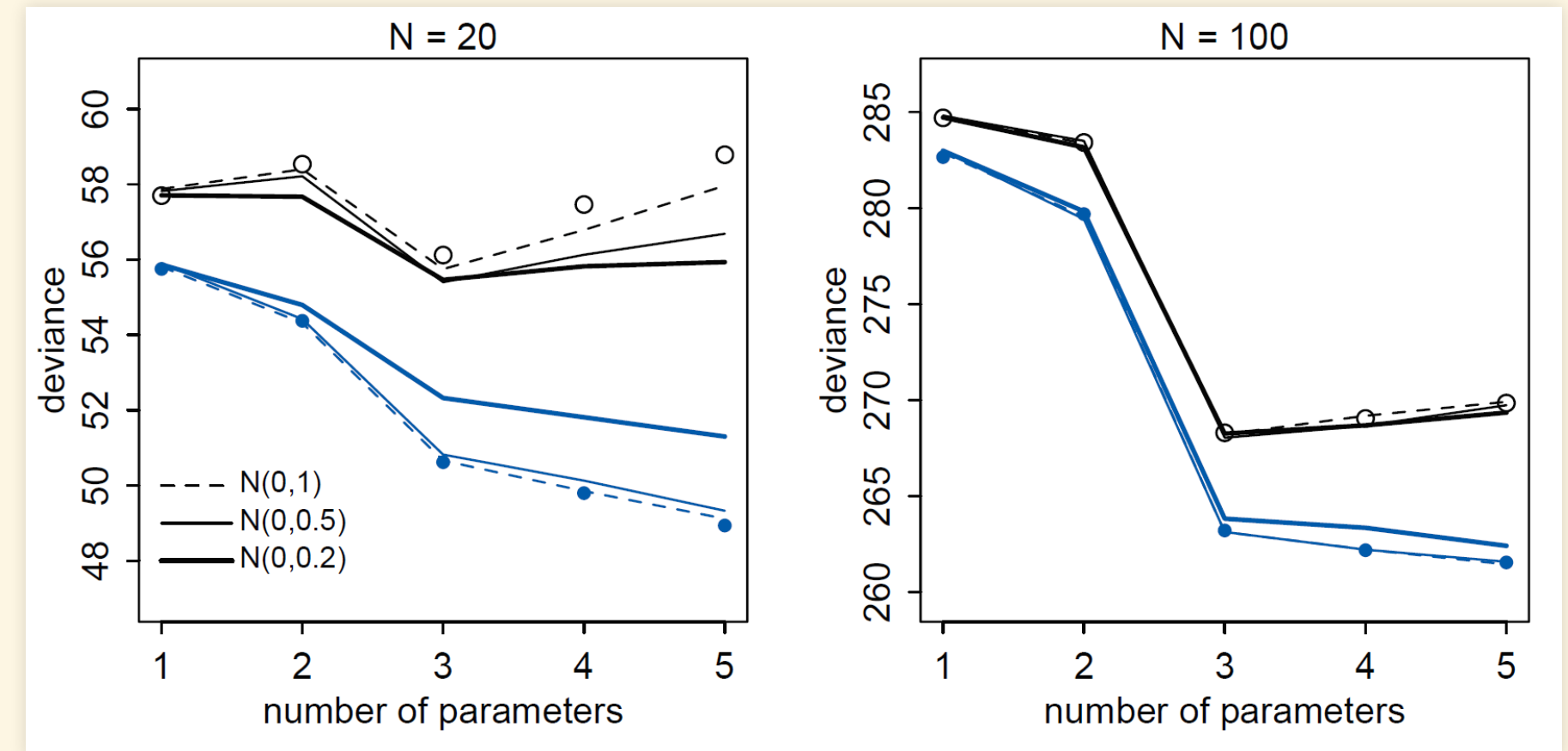
Regularization

Regularizing Priors

- Alternate approach: *regularizing priors*
 - Widely used in Machine Learning
 - Making the model worse at fitting *training data* can make it better at predicting *test data*.
 - Regularizing prior



Normal prior for β parameters:



- blue = in-sample, black = out-of-sample
- When there is a lot of data ($N = 100$), the regularizing priors keep the out-of-sample deviance small, even with many parameters.
- Regularizing priors tend to force unnecessary parameters to small (near-zero) values.
- Fancier regularizing priors set a threshold and push parameters to be either far from zero or else very close to zero.

Predicting Predictive Accuracy

Cross-Validation

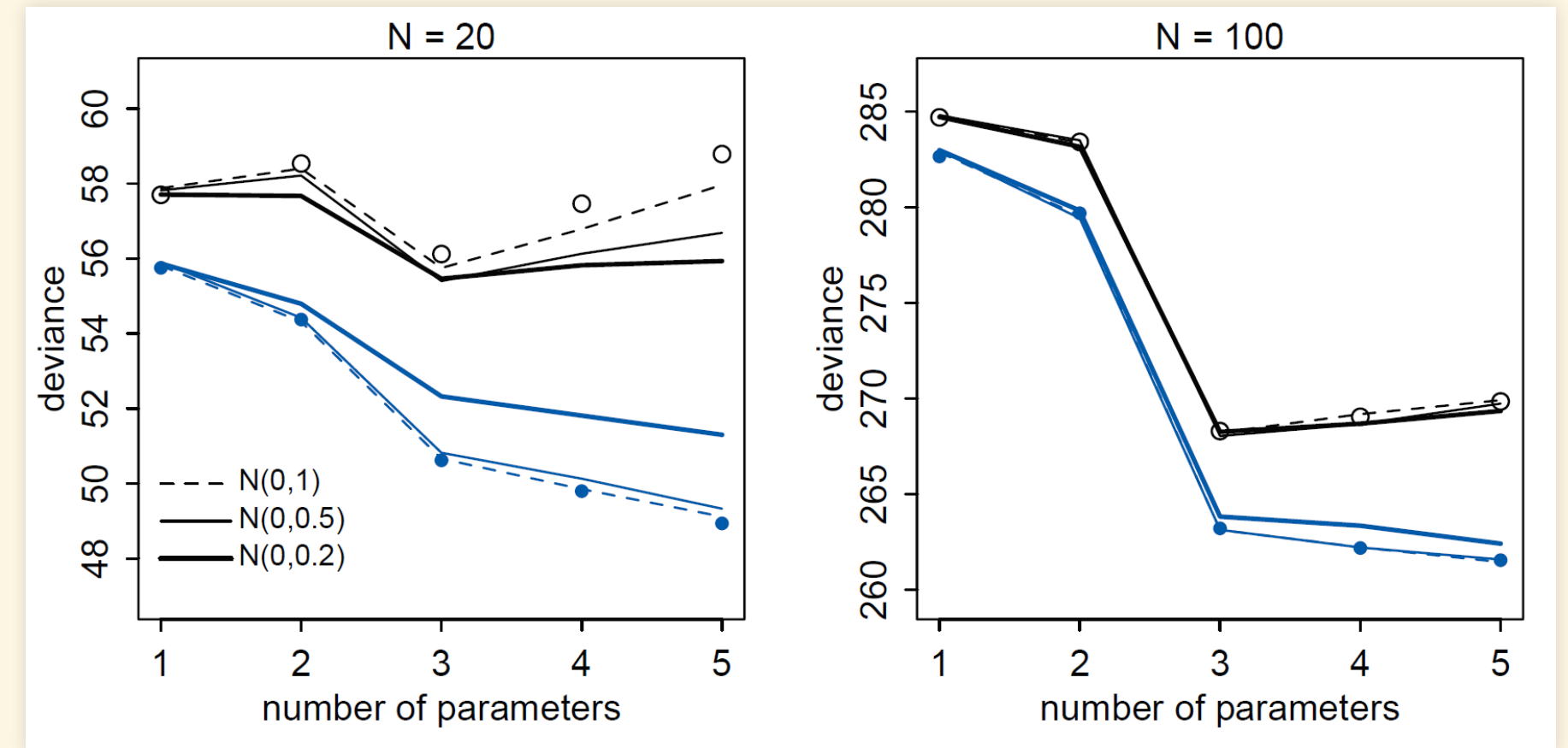
- How can we get a sense of how well our model will work with out-of-sample predictions?
- We started by splitting our data in half: *training* and *test* data.
- Sometimes it's not efficient to split our data in half.
- Can we do better?
- k -fold cross-validation:
 - Split data into k equal parts (example: $k = 5$)
 - For each part i (called a “fold”), fit the model to the other $k - 1$ parts and then predict part i .
 - Repeat this for all k parts.
 - Use all k folds to assess model performance
- Leave-one-out cross-validation (LOOCV):
 - An extreme form of K -fold cross-validation, where $k = N$, the size of the data.
 - For each data point, fit the model to all the others and then predict that one point.
- Problem: If you have N observations, then you have to fit your model N times. If N is large, this can be very slow.
- Pareto-Smoothed Importance Sampling (PSIS) is a fancy technique that lets us estimate LOOCV while we fit the model one time, without actually having to do real cross-validation.

Information Criteria

- As an alternative to cross-validation, use information theory to estimate the out-of-sample KL divergence.
- Examine the differences between in-sample and out-of-sample divergence in the figure
 - The difference is roughly twice the number of parameters.
 - In general, for relatively flat priors, the overfitting penalty is about twice the number of parameters.
 - Akaike Information Criterion (AIC)

$$AIC = D_{\text{train}} + 2p = -2\text{lppd} + 2p,$$

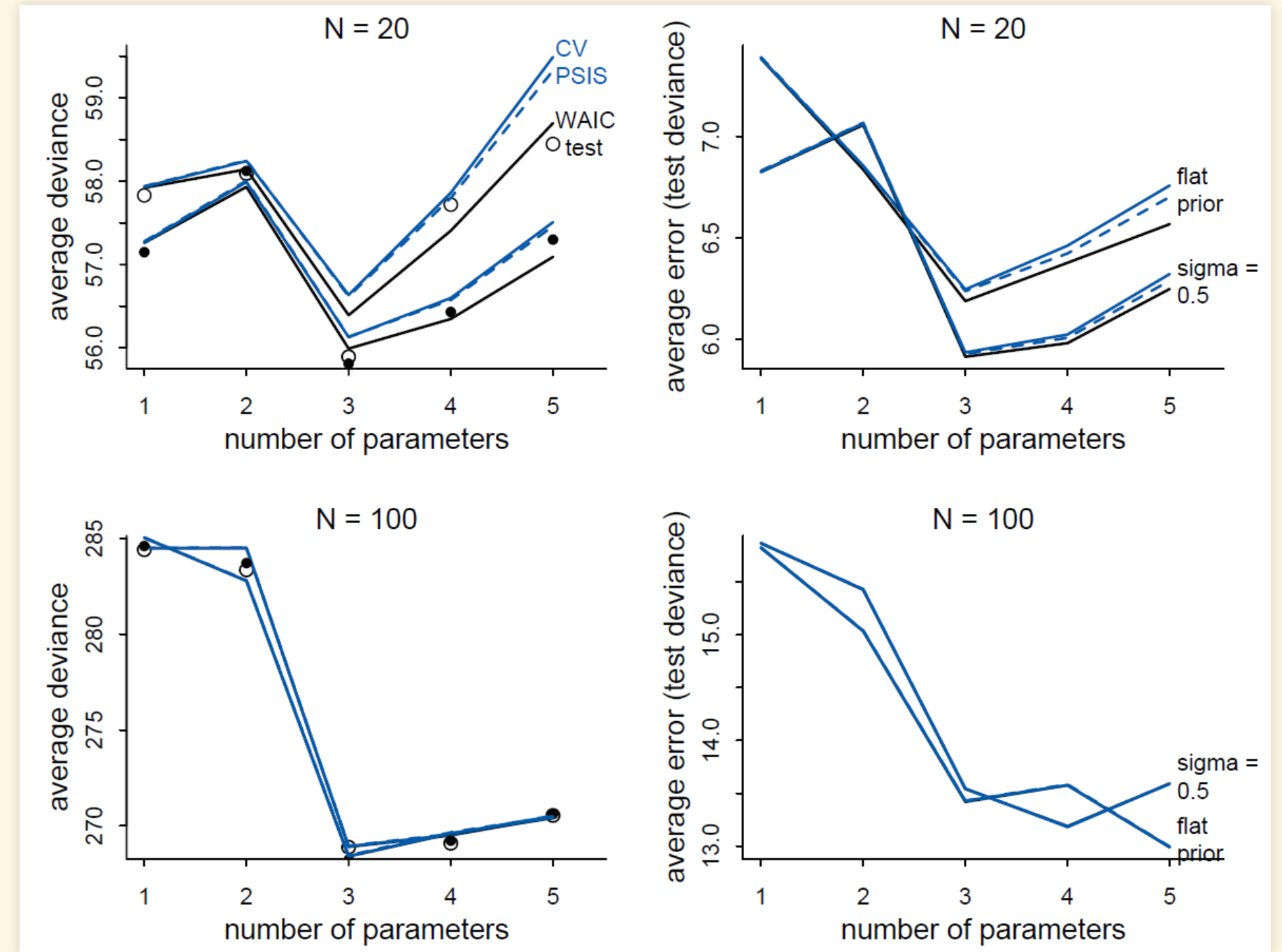
where lppd is the log-pointwise-predictive density (basically a sample of the posterior).



- Conditions for validity:
 - Priors are flat, or dominated by likelihood (data).
 - Posterior distribution is approximately Gaussian for each parameter.
 - The data sample size N is much greater than the number of parameters k .

Other Information Criteria

- AIC is only valid under these conditions:
 - Priors are flat, or dominated by likelihood (data).
 - Posterior distribution is approximately Gaussian for each parameter.
 - The data sample size N is much greater than the number of parameters k .
- Flat priors are usually not a good choice.
- DIC (Deviance Information Criteria) works with informative priors, but the other two criteria still apply.
- Watanabe-Akaike Information Criteria (WAIC, also called Widely Applicable Information Criterion) is more broadly applicable.
 - We won't go into details of calculating WAIC. The rethinking package will do it for us, and so will most other Bayesian analysis packages.
- General principle:
 - For all the information criteria we're examining, the smaller (more negative) they are, the better the model performs.



- Comparison of different measures

How to Compare Models

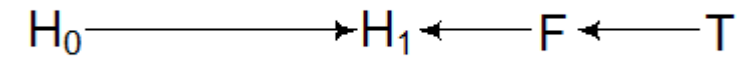
Comparison vs. Selection

- Many people use CV, PSIS, Deviance, or Information Criteria to select models
 - Use whatever model has the smallest score
- This is not wise. It only looks at what model is smallest, but doesn't consider how great the differences are between models.
 - This is like only looking at the mode (maximum) of the posterior and ignoring the rest of it.
 - The width the posterior matters too. It tells us about how uncertain the estimate is.
- When we compare models, look at how great the differences are between them.
- Remember that these criteria tell us about predictive power, but we have seen that predictive power doesn't tell us about causality.
 - Backdoor paths can have useful information, even though it's not causal.
 - But backdoor predictions only work if we don't interfere with the system.
 - In other words, if the future is just like the past.
 - In the plant-growth model, knowing about the fungus was a better predictor of plant growth than knowing about the anti-fungus treatment
 - but knowing about the fungus doesn't help us predict the effect of treating a field.

Example Using WAIC

- Plant growth experiment:

- DAG



H_0 = height before, H_1 = height after,
 T = anti-fungal treatment, F = fungus

- Three models:

1. mdl_0 : $\mu \sim \text{log-Normal}(0, 0.25)$

2. mdl_T : $\mu = \alpha + \beta_T T$

3. mdl_TF : $\mu = \alpha + \beta_T T + \beta_F F$

```
set.seed(11)
round(WAIC(mdl_TF), 2)
```

```
##      WAIC      lppd penalty std_err
## 1 361.45 -177.17      3.55   14.17
```

```
set.seed(77)
round(compare(mdl_0, mdl_T, mdl_TF, func = WAIC), 2)
```

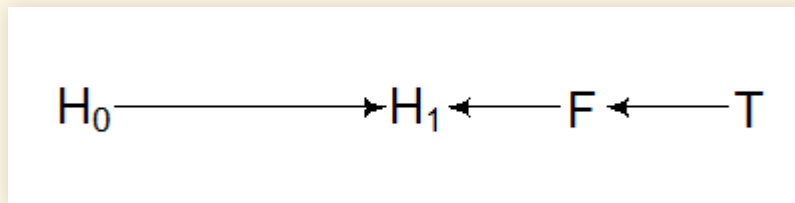
```
##      WAIC      SE dWAIC      dSE pWAIC weight
## mdl_TF 361.81 14.26  0.00      NA   3.74      1
## mdl_T  402.65 11.20 40.84 10.44   2.58      0
## mdl_0  405.91 11.65 44.10 12.22   1.58      0
```

- Best predictions on top.
- “d” are differences from the best model.
- “SE” are standard errors.
- pWAIC is prediction penalty (estimate of *out-of-sample* vs. *in-sample*).
- weight gives the relative support for each model, given the data.
 - Useful for model-averaging.

Example Using WAIC

- Plant growth experiment:

- DAG



H_0 = height before, H_1 = height after,
 T = anti-fungal treatment, F = fungus

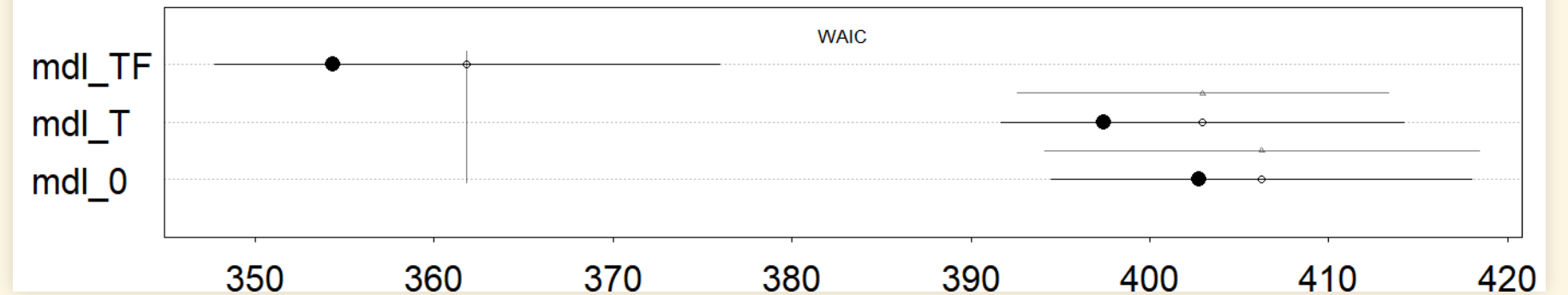
- Three models:

1. mdl_0 : $\mu \sim \text{log-Normal}(0, 0.25)$

2. mdl_T : $\mu = \alpha + \beta_T T$

3. mdl_TF : $\mu = \alpha + \beta_T T + \beta_F F$

```
plot(compare(mdl_0, mdl_T, mdl_TF, func = WAIC), cex=2, lwd=2)
```



- Plot:

- Line is range of estimated out-of-sample deviance
- Gray point is best estimate of out-of-sample deviance
- Black point is in-sample deviance
- Light lines over models are differences from best model

- TF model is clearly the best for predictions
 - We can't tell which of the others is better
- TF model has post-treatment confounder
 - WAIC can't tell us about causation

