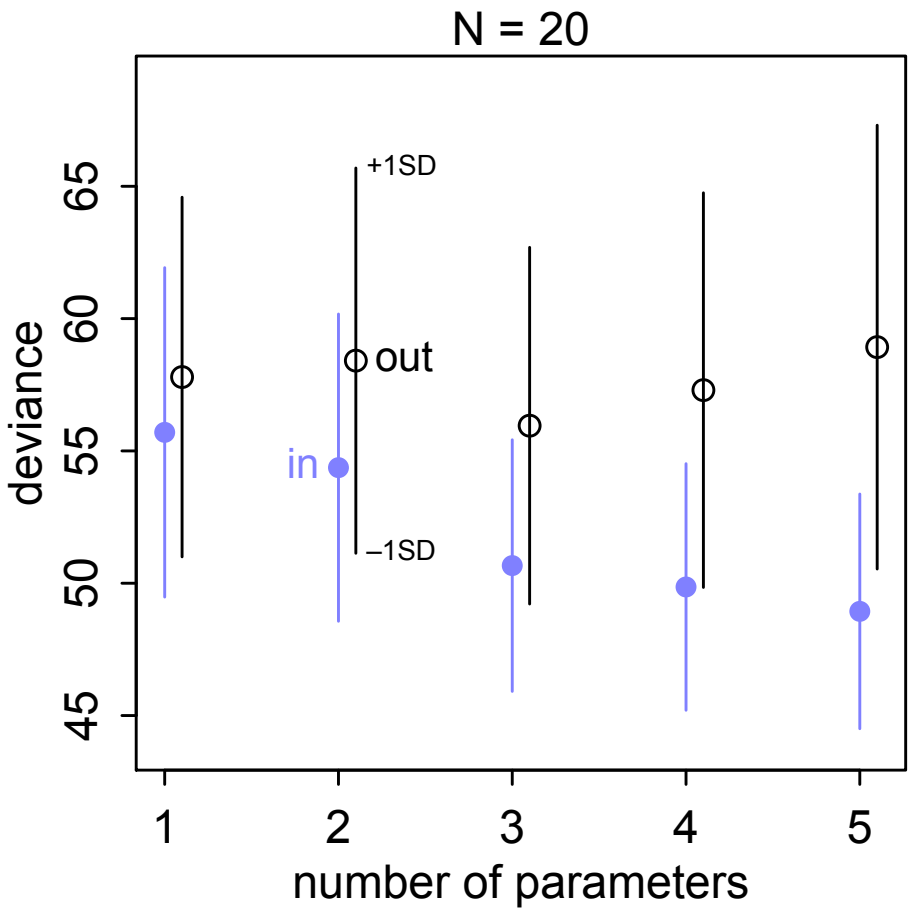
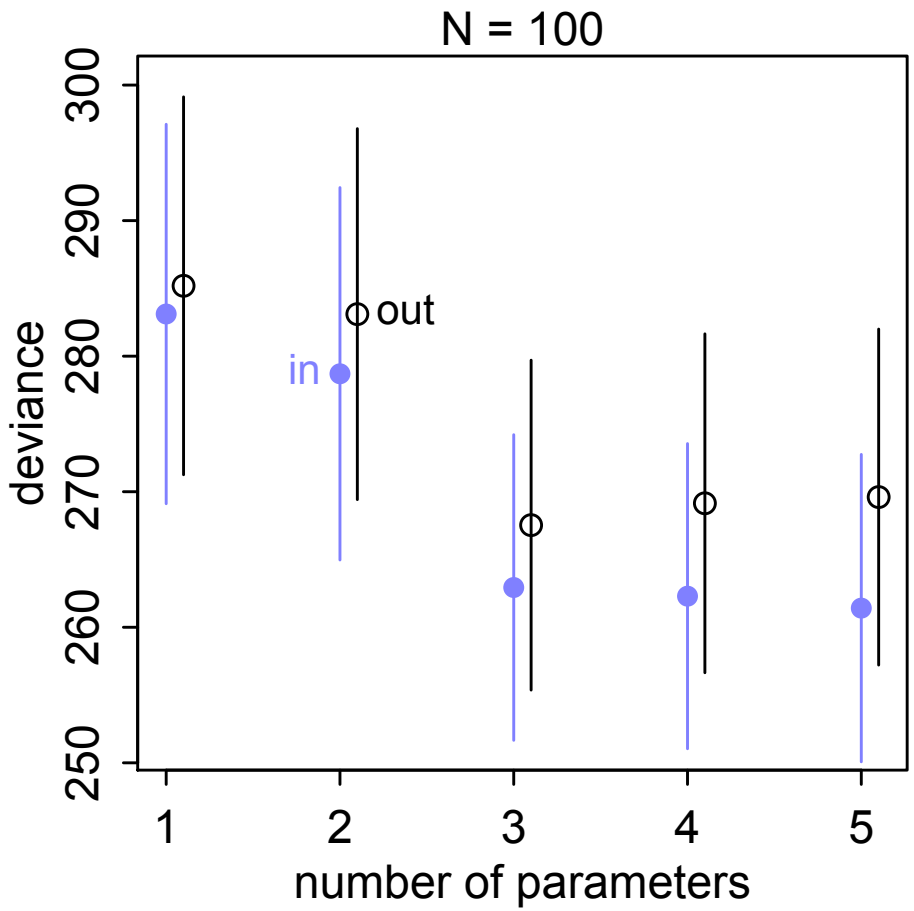
Statistical Rethinking

Week 4: Ockham, Ulysses, and the Model

Richard McElreath

Everybody overfits



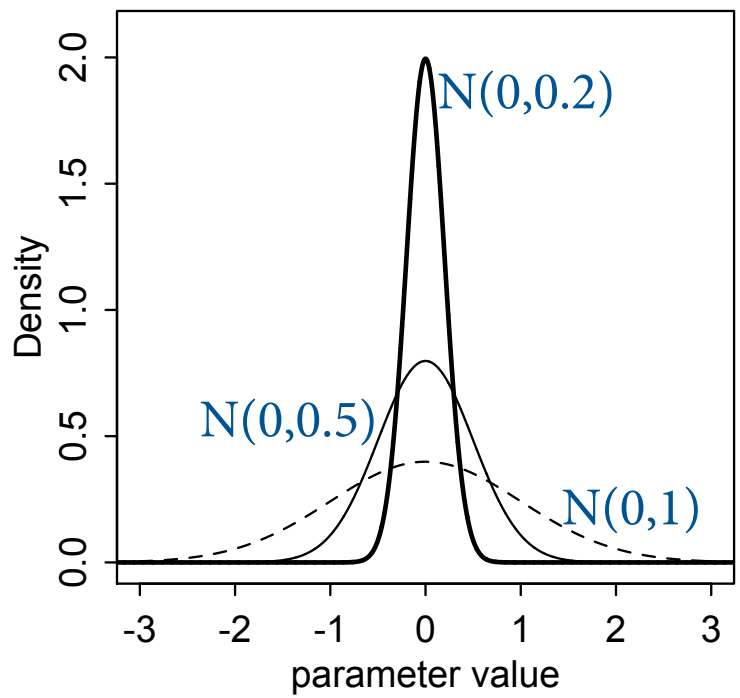


Regularization

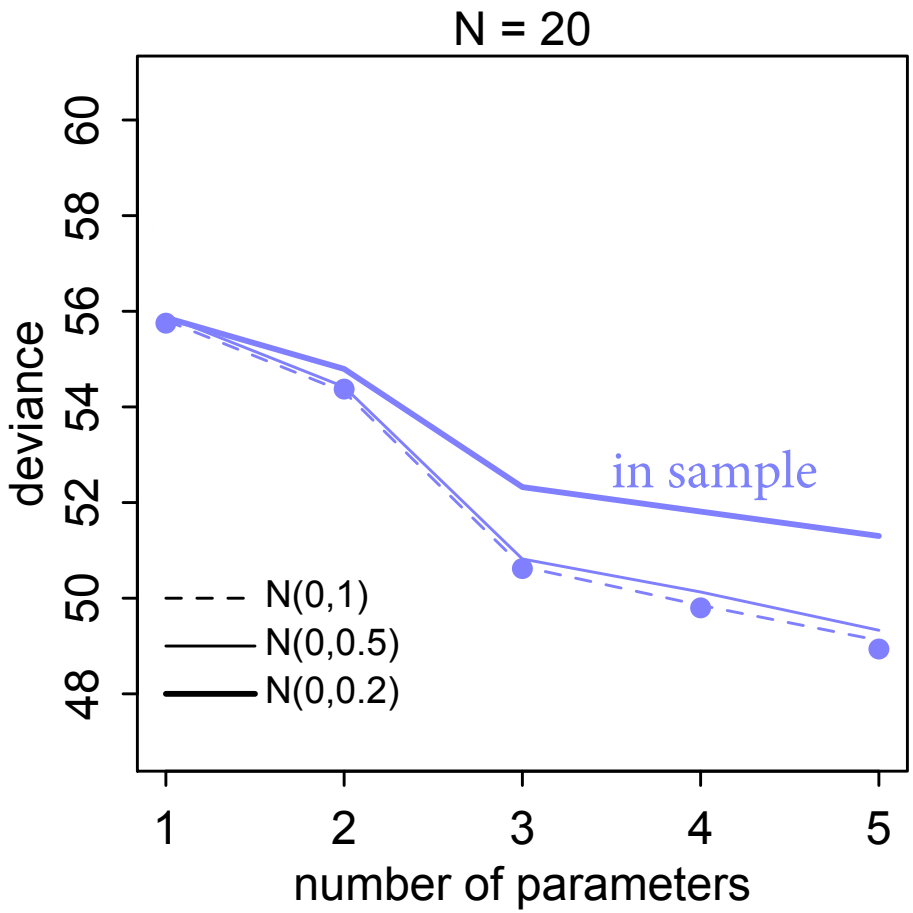
- Use informative, conservative priors to reduce overfitting => model learns less from sample
- But if too informative, model learns too little
- Such priors are *regularizing*

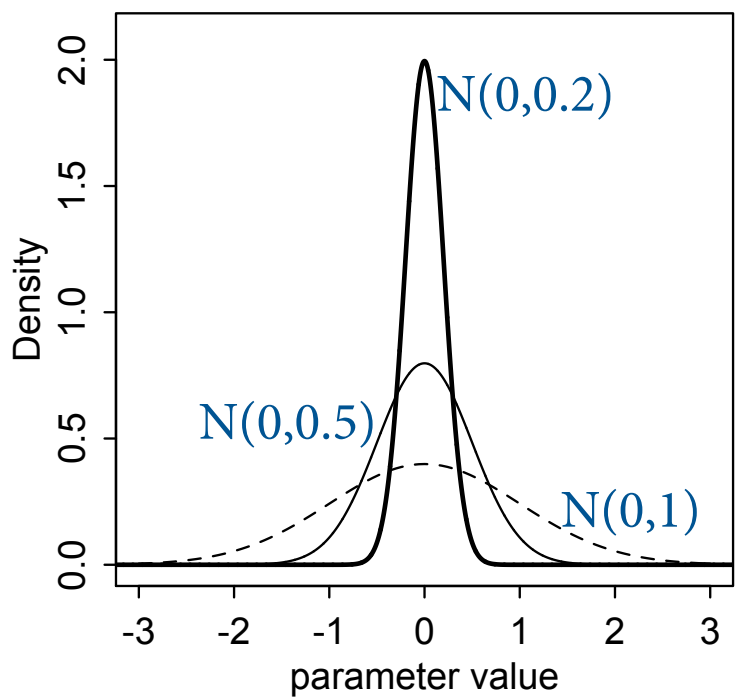
$$y_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$
$$\alpha \sim \text{Normal}(0, 100)$$
$$\beta \sim \text{Normal}(0, 1)$$
$$\sigma \sim \text{Uniform}(0, 10)$$

regularizing prior

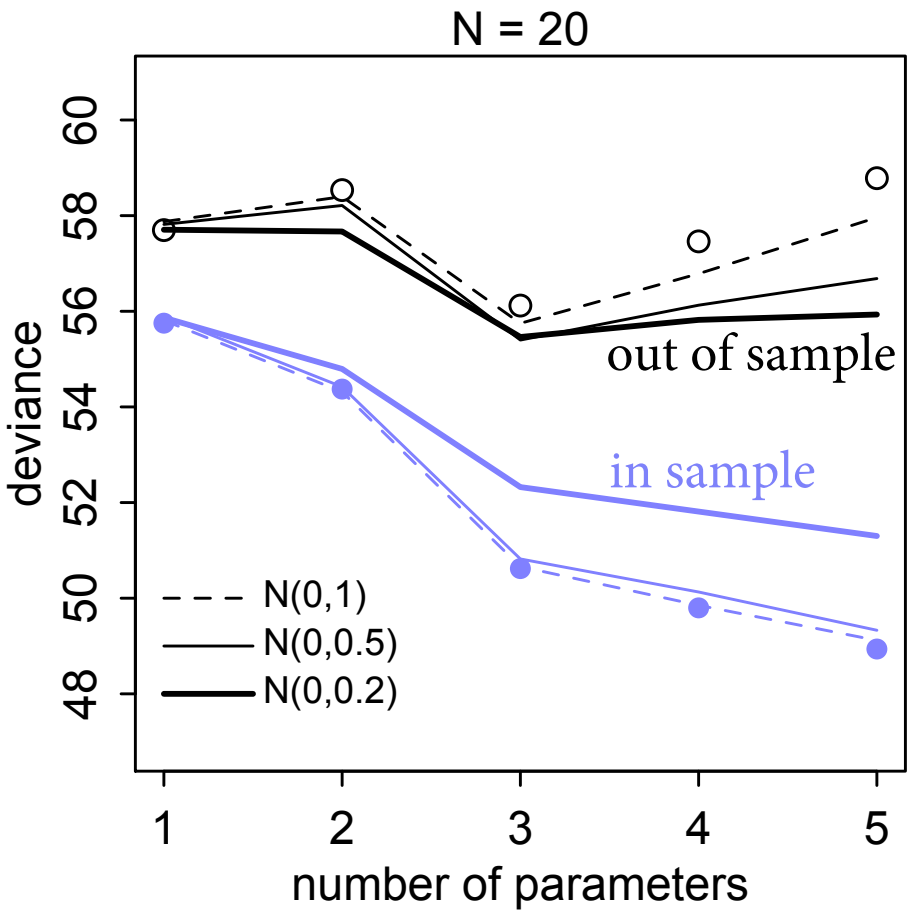


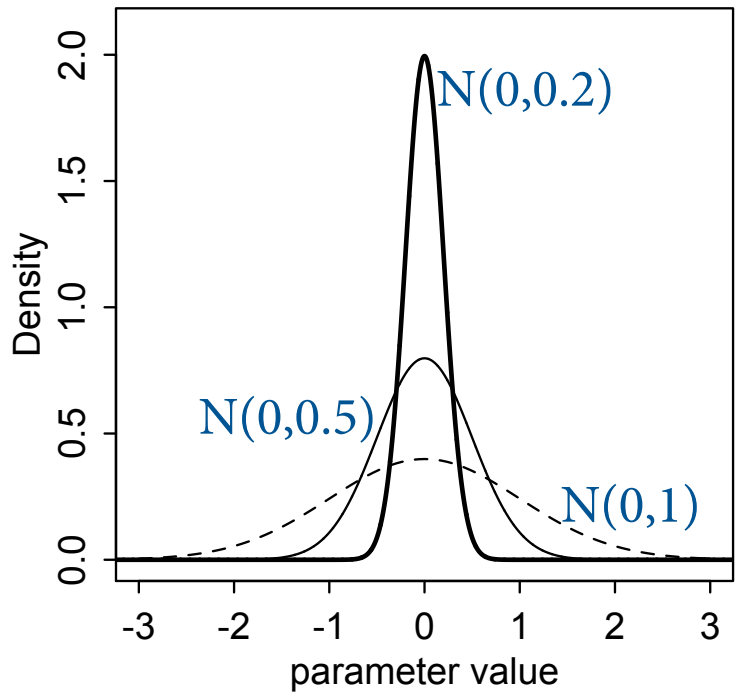
Regularization



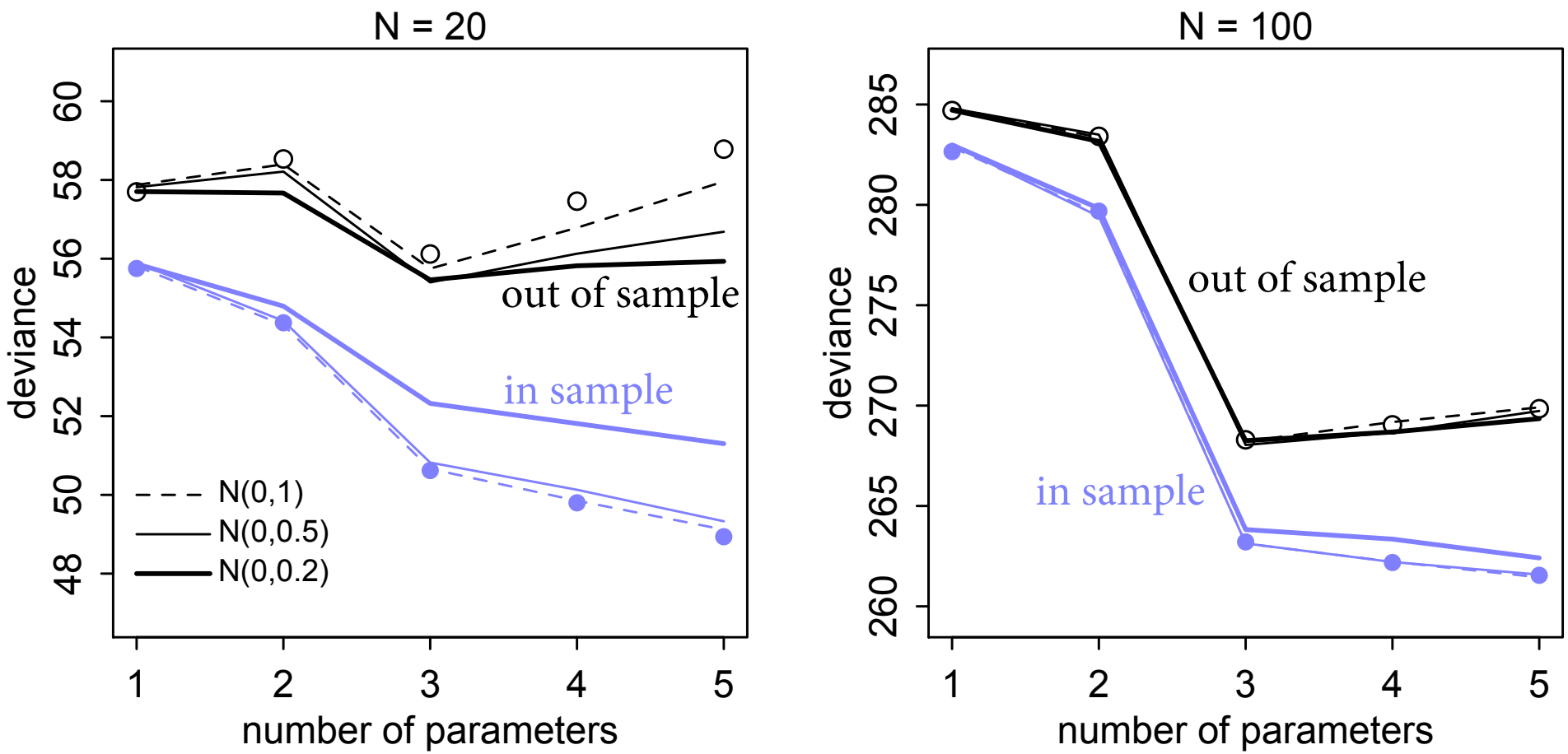


Regularization





Regularization



Information criteria

- Can we estimate out-of-sample deviance?
- In theory: Cross-validation
- Also in theory: *Information criteria*
  - *Information*, because use of deviance based on information theoretic analysis
  - *Criteria*, because used to compare models
- Information criteria estimate *relative* out of sample error
  - AIC, DIC, WAIC, many others

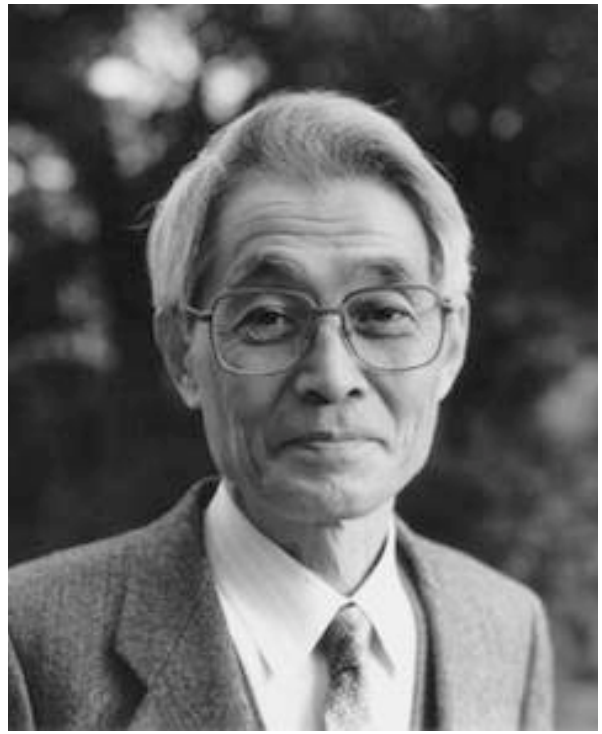
Akaike information criterion

[ah–ka–ee–kay]

- A meta-model of forecasting:
  - Two samples: *training* and *testing*, size N
  - Fit model to *training* sample, get D_{train}
  - Use fit to *training* to compute D_{test}
  - Difference $D_{\text{test}} - D_{\text{train}}$ is overfitting
- Under some strict conditions:

$$\text{AIC} = D_{\text{train}} + 2k \approx \text{E}\, D_{\text{test}}$$

k is parameter count



Hirotugu Akaike
(1927–2009)

Akaike information criterion

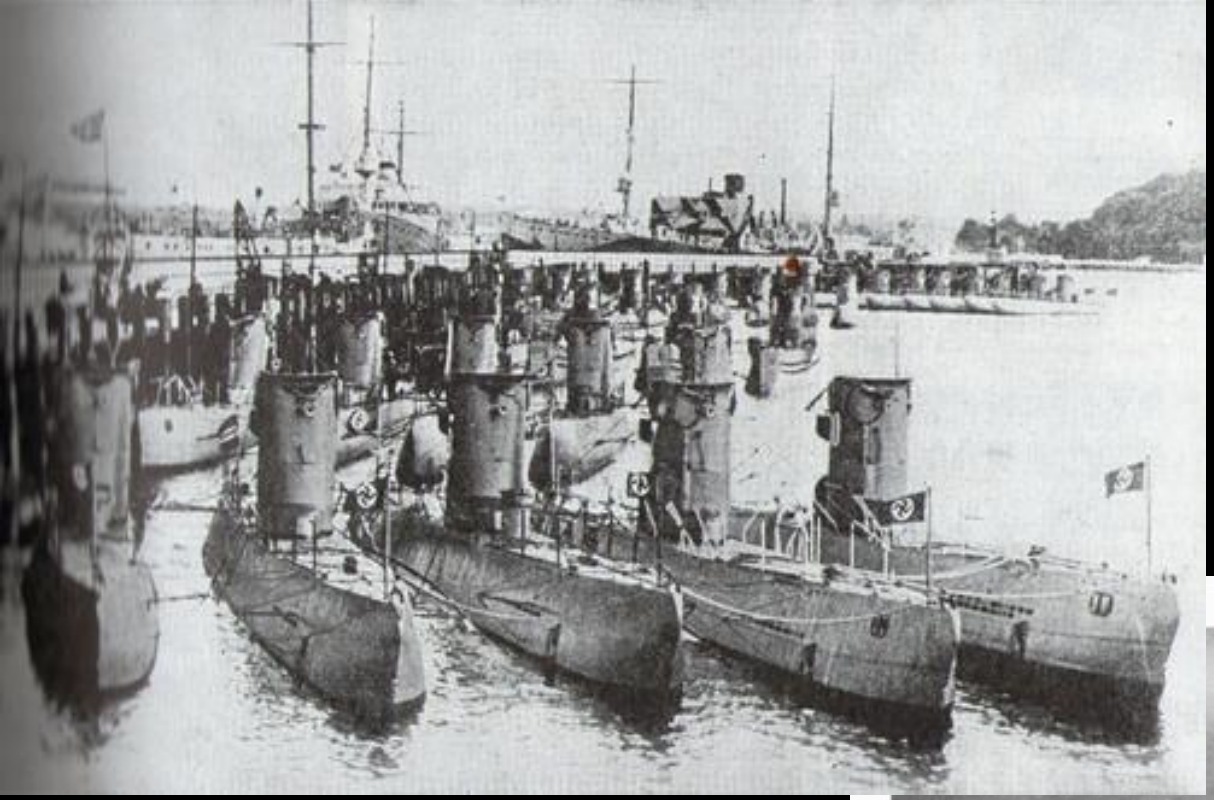
$$\mathrm{AIC} = D_{\text{train}} + 2k \approx \mathrm{E}\, D_{\text{test}}$$

- Conditions:
  - You like the AIC forecasting model
  - Flat priors
  - No varying/mixed/random effects
  - Gaussian posterior distribution
  - $k << N$; as k approaches N:

$$\mathrm{AICc} = D_{\text{train}} + \frac{2k}{1 - (k+1)/N}$$

Akaike information criterion

- Prediction/forecasting task matters
- Suppose we care about accumulated error over learning, aka *prequential* error
- Consider the humble wurst
  - Grill-only or boil-then-grill?
  - Want to consume each wurst
  - How to learn and eat well at same time?
  - AIC not the right scenario



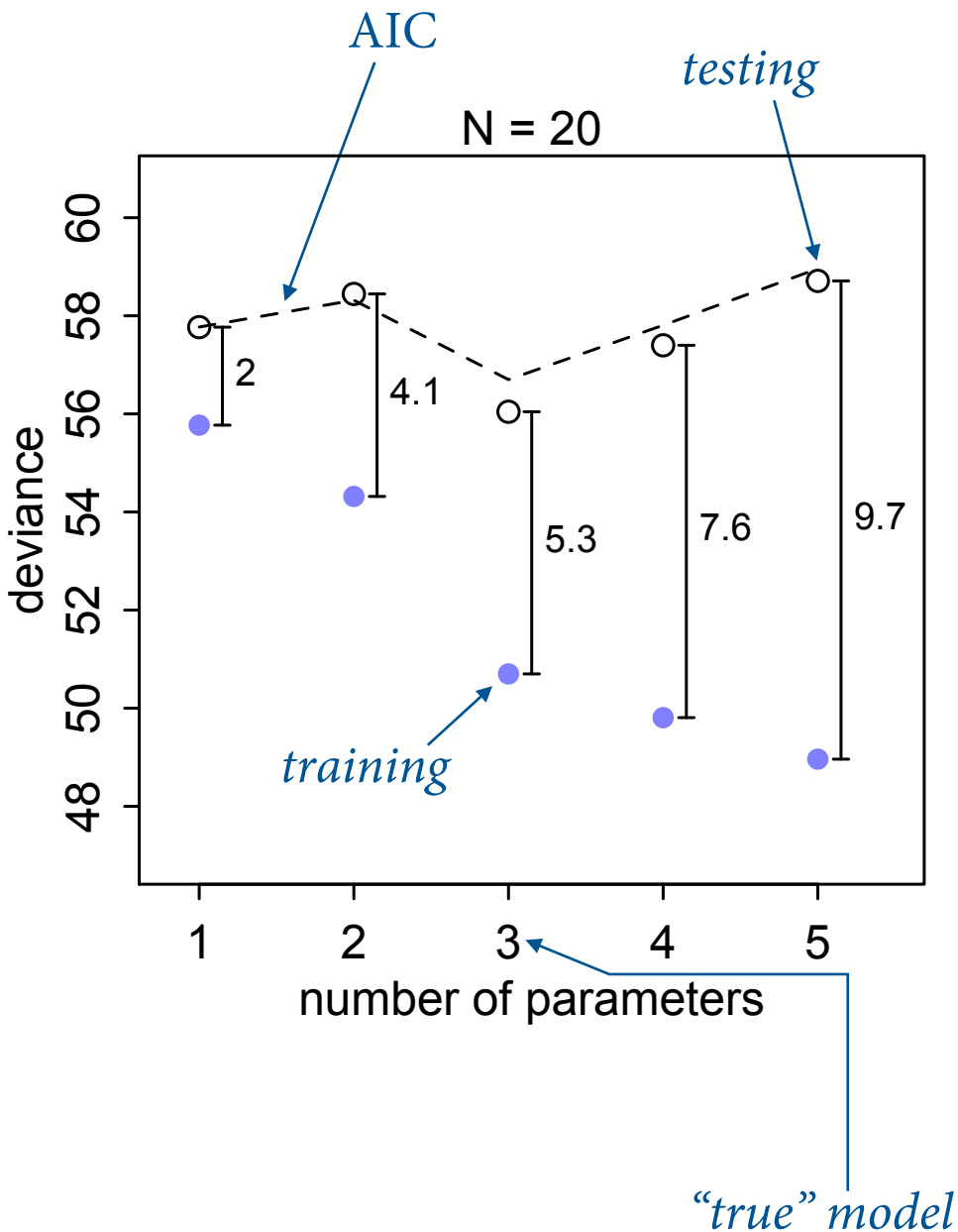

AIC
testing
N = 20
deviance
60
58
56
54
52
50
48
2
4.1
5.3
7.6
9.7
training
1
2
3
4
5
number of parameters
"true" model

Figure 6.10

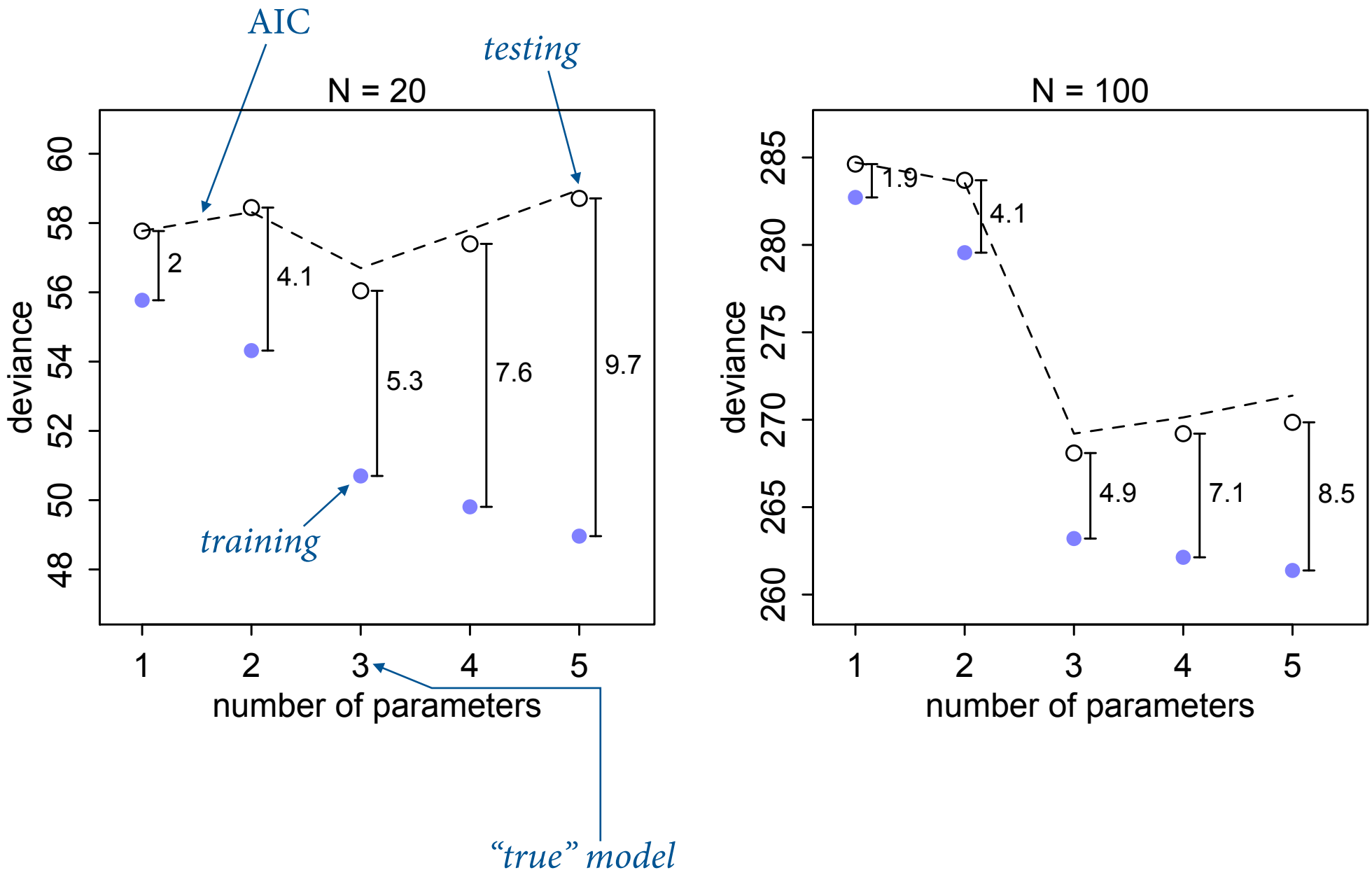
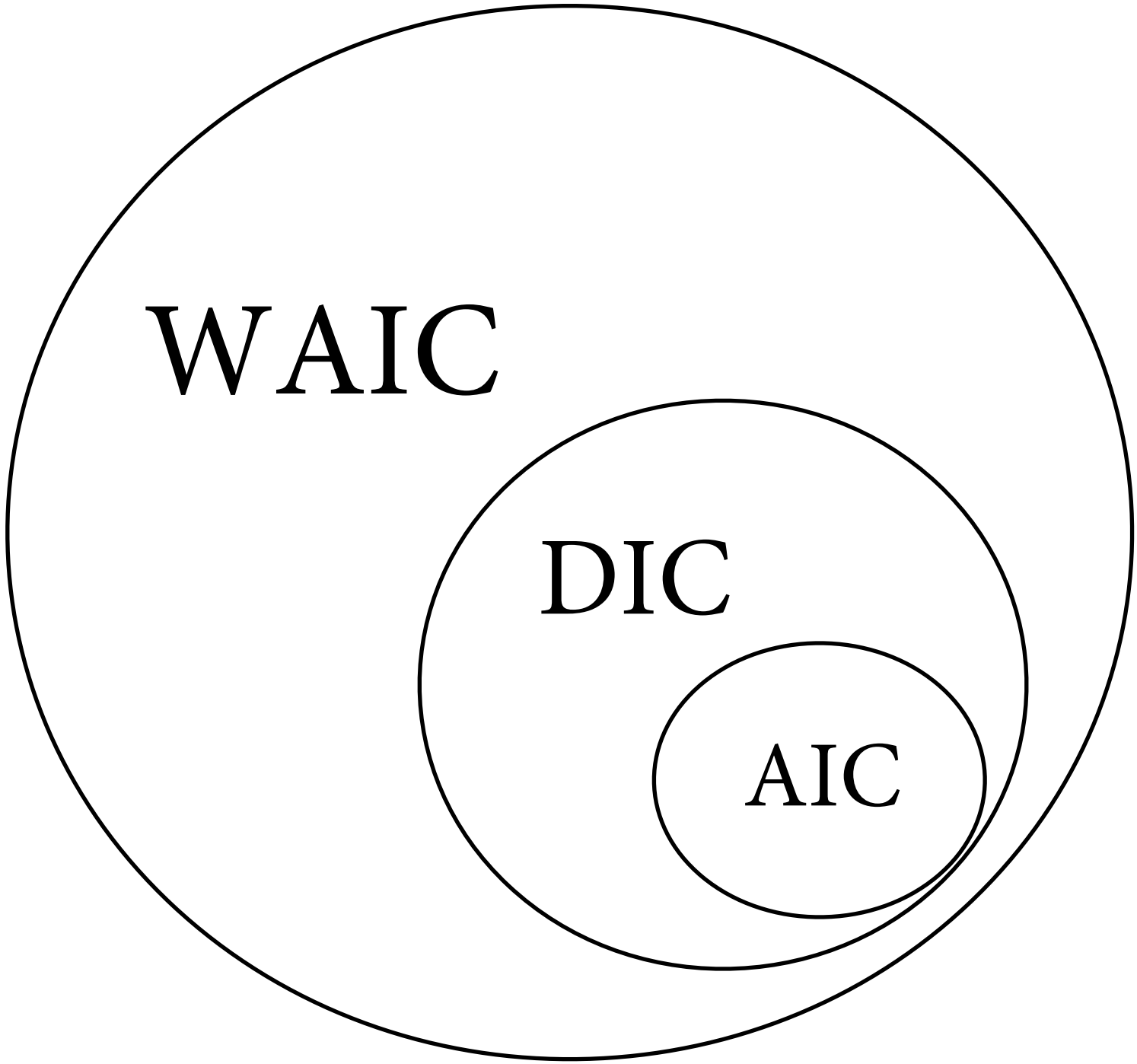

AIC
testing
N = 20
deviance
48
50
52
54
56
58
60
2
4.1
5.3
7.6
9.7
training
1
2
3
4
5
number of parameters
"true" model
N = 100
deviance
260
265
270
275
280
285
1.9
4.1
4.9
7.1
8.5
1
2
3
4
5
number of parameters

Figure 6.10

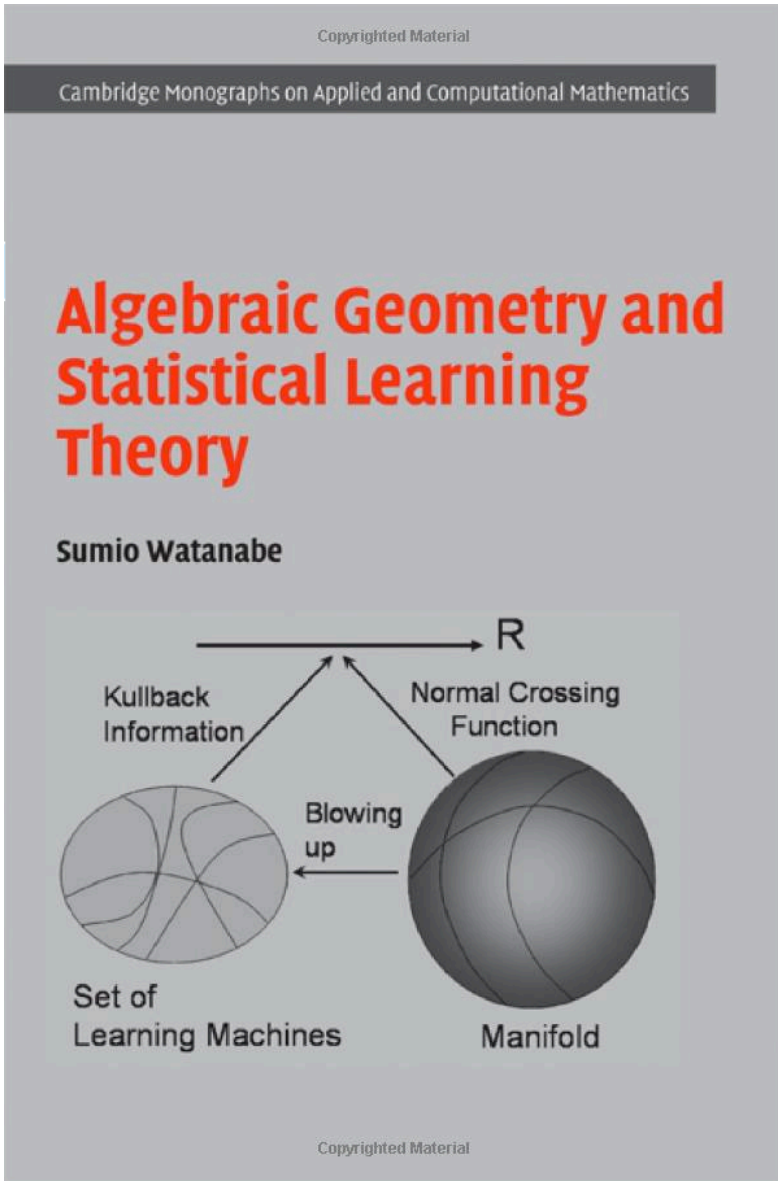

WAIC
DIC
AIC

Widely Applicable IC

- Widely Applicable Information Criterion (WAIC)
  - Sumio Watanabe 2010
  - Sometimes called "Watanabe-Akaike Information Criterion"

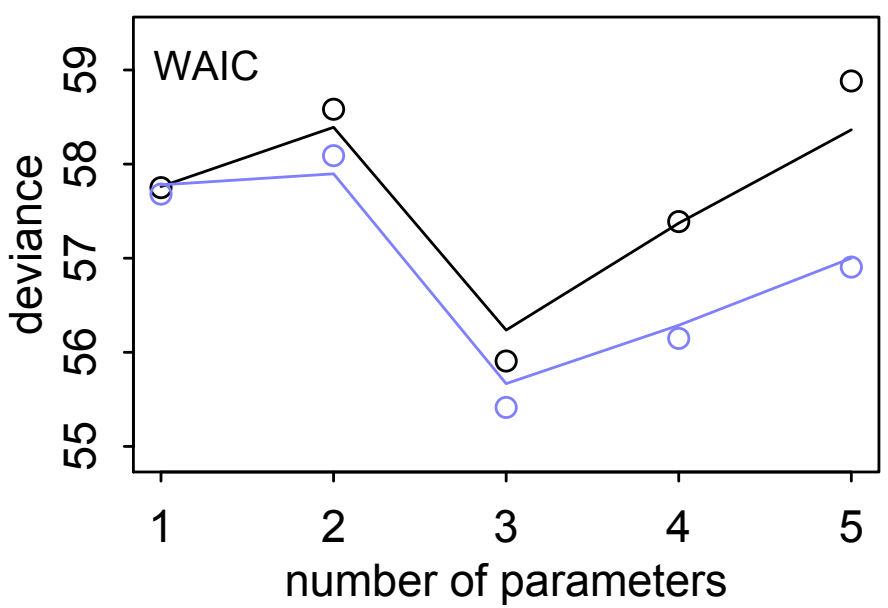
$$\mathrm{WAIC} = -2\mathrm{lppd} + 2p_{\mathrm{WAIC}}$$

- Does not assume Gaussian posterior
- `WAIC` function in rethinking



At the beach, finally

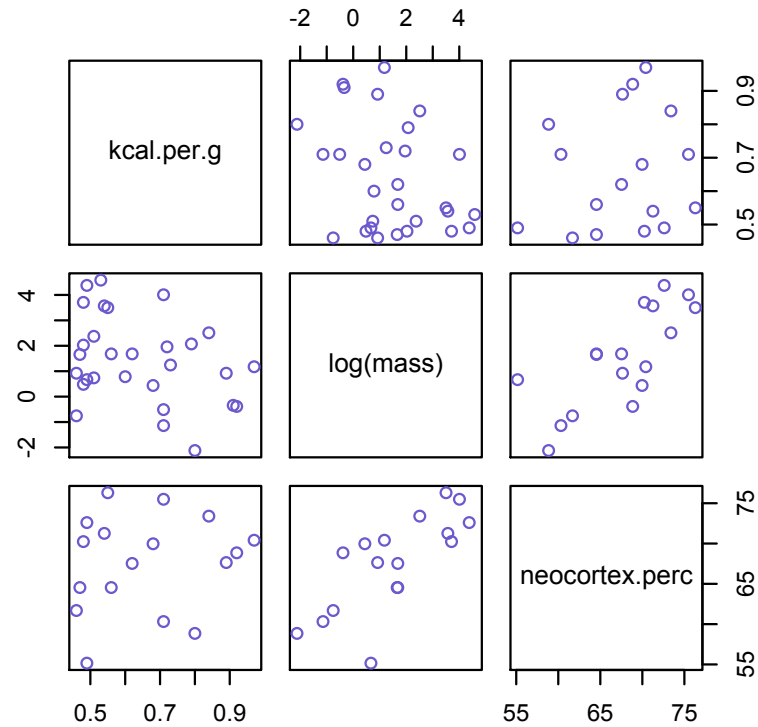
- Underfitting possible; overfitting inevitable
- Regularizing priors *reduce* it
- Information criteria *measure* it
- Taste great together



Using AIC/DIC/WAIC

- Avoid *model selection*
- *Model comparison*: quantify uncertainty about models, in addition to uncertainty about parameters
- *Model averaging*: Simulate predictions, averaging over uncertainty about models
  - don't average *parameters*, but only *predictions*

Primate milk again



R code 6.21

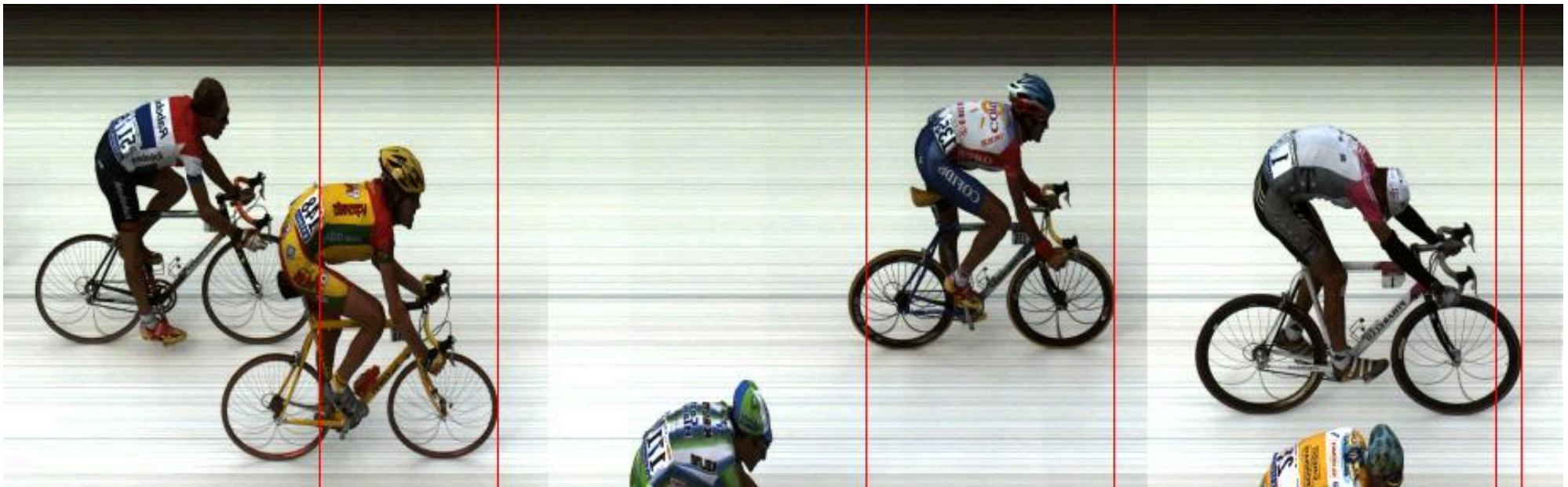
```
data(milk)
d <- milk[ complete.cases(milk) , ]
d$neocortex <- d$neocortex.perc / 100
dim(d)
```

```
[1] 17  9
```

Primate milk again

- Fit four different models:

  m6.11: kcal ~ 1
  m6.12: kcal ~ 1 + neocortex
  m6.13: kcal ~ 1 + log(mass)
  m6.14: kcal ~ 1 + neocortex + log(mass)



Comparing

- What is expected out-of-sample deviance for each model?

R code 6.24

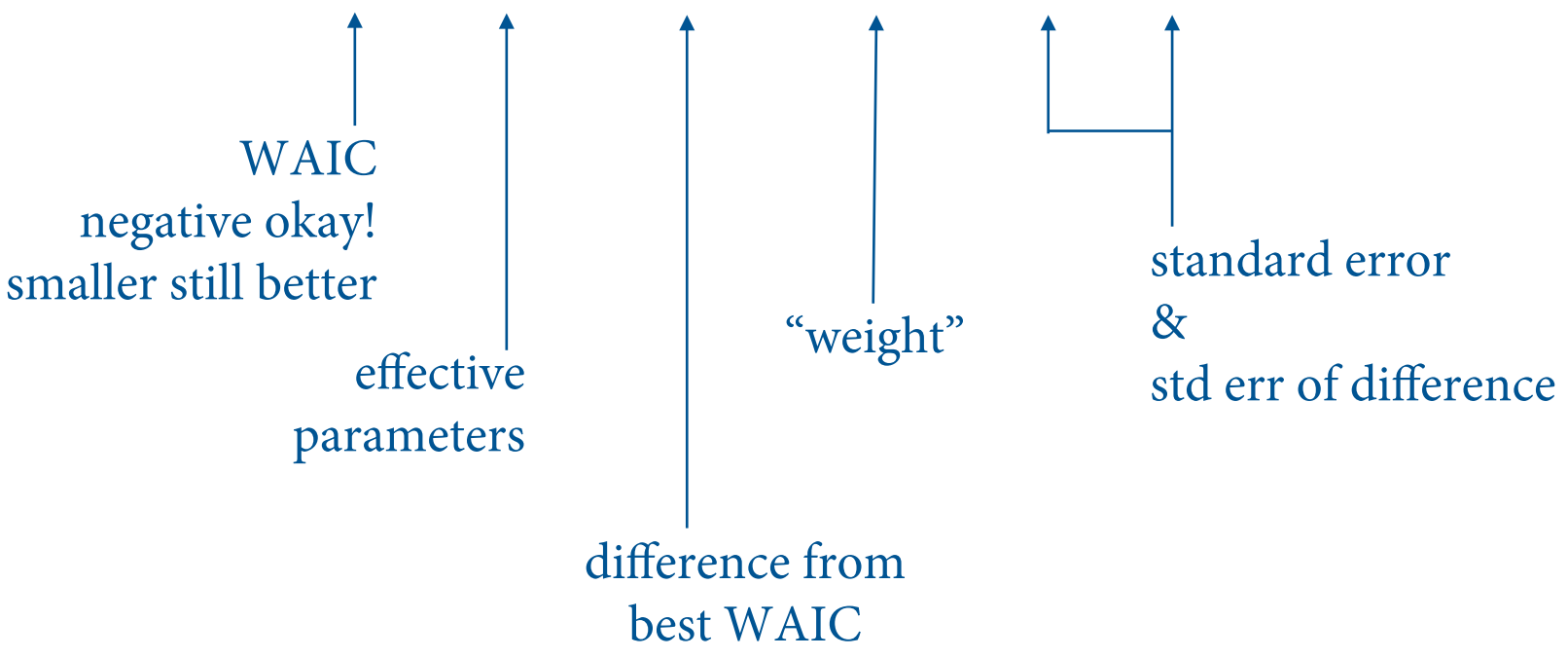
```
( milk.models <- compare( m6.11 , m6.12 , m6.13 , m6.14 ) )
```

```
        WAIC pWAIC dWAIC weight   SE  dSE
m6.14 -15.0   4.8   0.0   0.93 7.54   NA
m6.11  -8.3   1.8   6.7   0.03 4.52 7.26
m6.13  -7.9   3.0   7.1   0.03 5.67 5.33
m6.12  -6.2   2.9   8.9   0.01 4.34 7.57
```

R code 6.24

```
( milk.models <- compare( m6.11 , m6.12 , m6.13 , m6.14 ) )
```

```
        WAIC pWAIC dWAIC weight   SE  dSE
m6.14 -15.0   4.8   0.0   0.93 7.54   NA
m6.11  -8.3   1.8   6.7   0.03 4.52 7.26
m6.13  -7.9   3.0   7.1   0.03 5.67 5.33
m6.12  -6.2   2.9   8.9   0.01 4.34 7.57
```



Weights

R code 6.24

```
( milk.models <- compare( m6.11 , m6.12 , m6.13 , m6.14 ) )
```

```
        WAIC pWAIC dWAIC weight   SE  dSE
m6.14 -15.0   4.8   0.0   0.93 7.54   NA
m6.11  -8.3   1.8   6.7   0.03 4.52 7.26
m6.13  -7.9   3.0   7.1   0.03 5.67 5.33
m6.12  -6.2   2.9   8.9   0.01 4.34 7.57
```

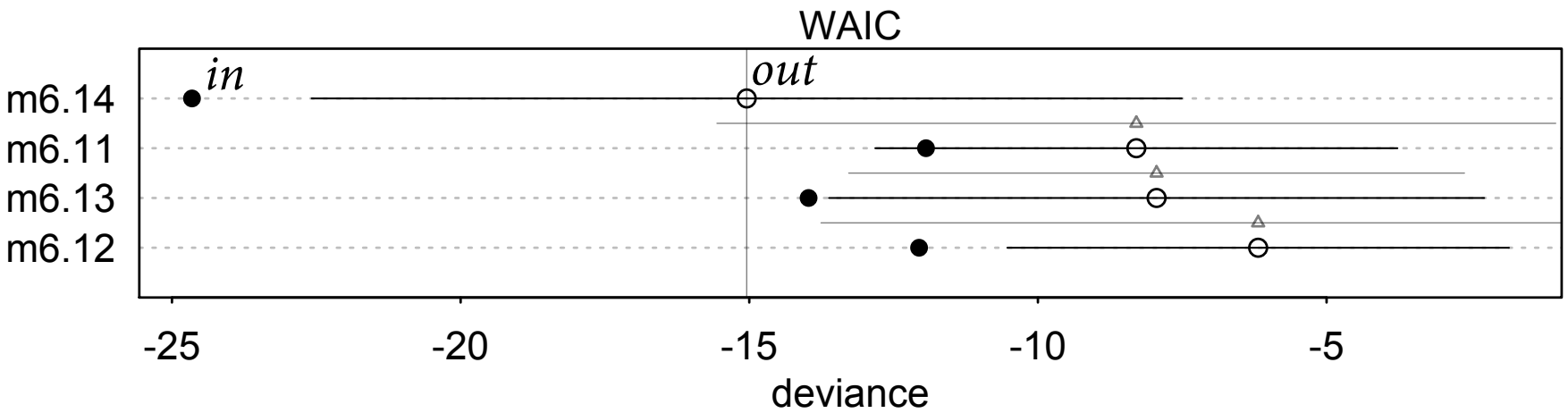
- deviance estimate of relative divergence
- convert to probability scale, standardize => “weight”
- each weight is estimated probability model is best for prediction
- BUT just a central estimate; need to look at std err...

Standard errors

R code 6.24

```
( milk.models <- compare( m6.11 , m6.12 , m6.13 , m6.14 ) )
```

	WAIC	pWAIC	dWAIC	weight	SE	dSE
m6.14	-15.0	4.8	0.0	0.93	7.54	NA
m6.11	-8.3	1.8	6.7	0.03	4.52	7.26
m6.13	-7.9	3.0	7.1	0.03	5.67	5.33
m6.12	-6.2	2.9	8.9	0.01	4.34	7.57

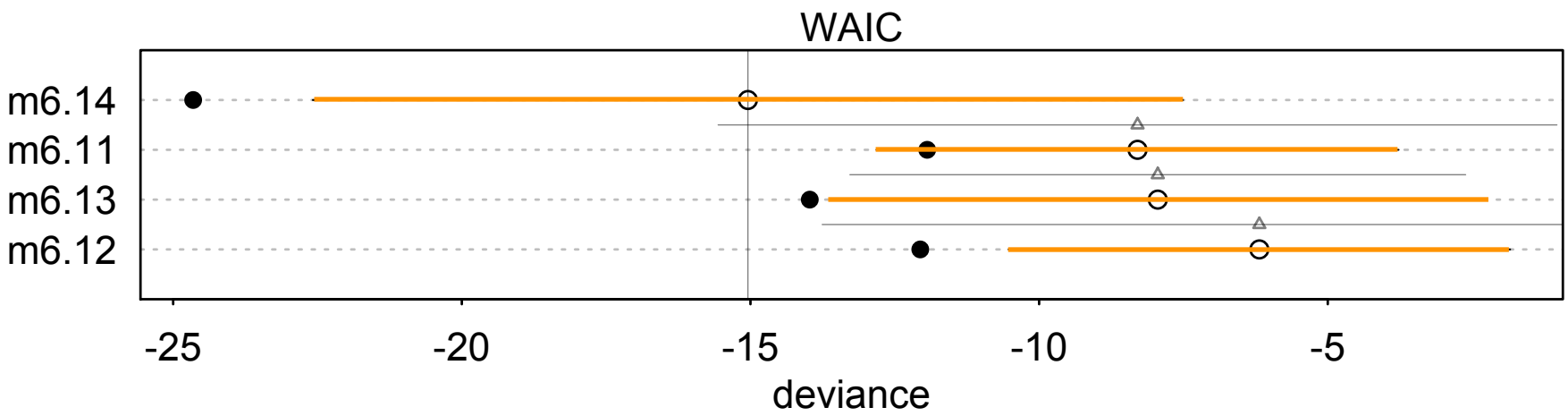


Standard errors

R code 6.24

```
( milk.models <- compare( m6.11 , m6.12 , m6.13 , m6.14 ) )
```

```
        WAIC pWAIC dWAIC weight   SE  dSE
m6.14 -15.0   4.8   0.0   0.93 7.54   NA
m6.11  -8.3   1.8   6.7   0.03 4.52 7.26
m6.13  -7.9   3.0   7.1   0.03 5.67 5.33
m6.12  -6.2   2.9   8.9   0.01 4.34 7.57
```

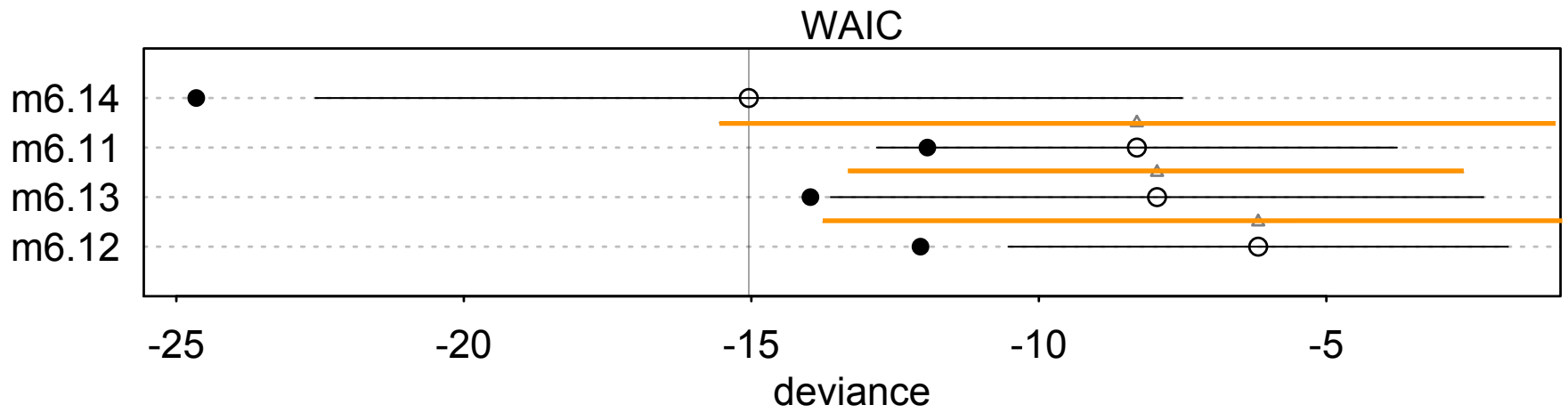


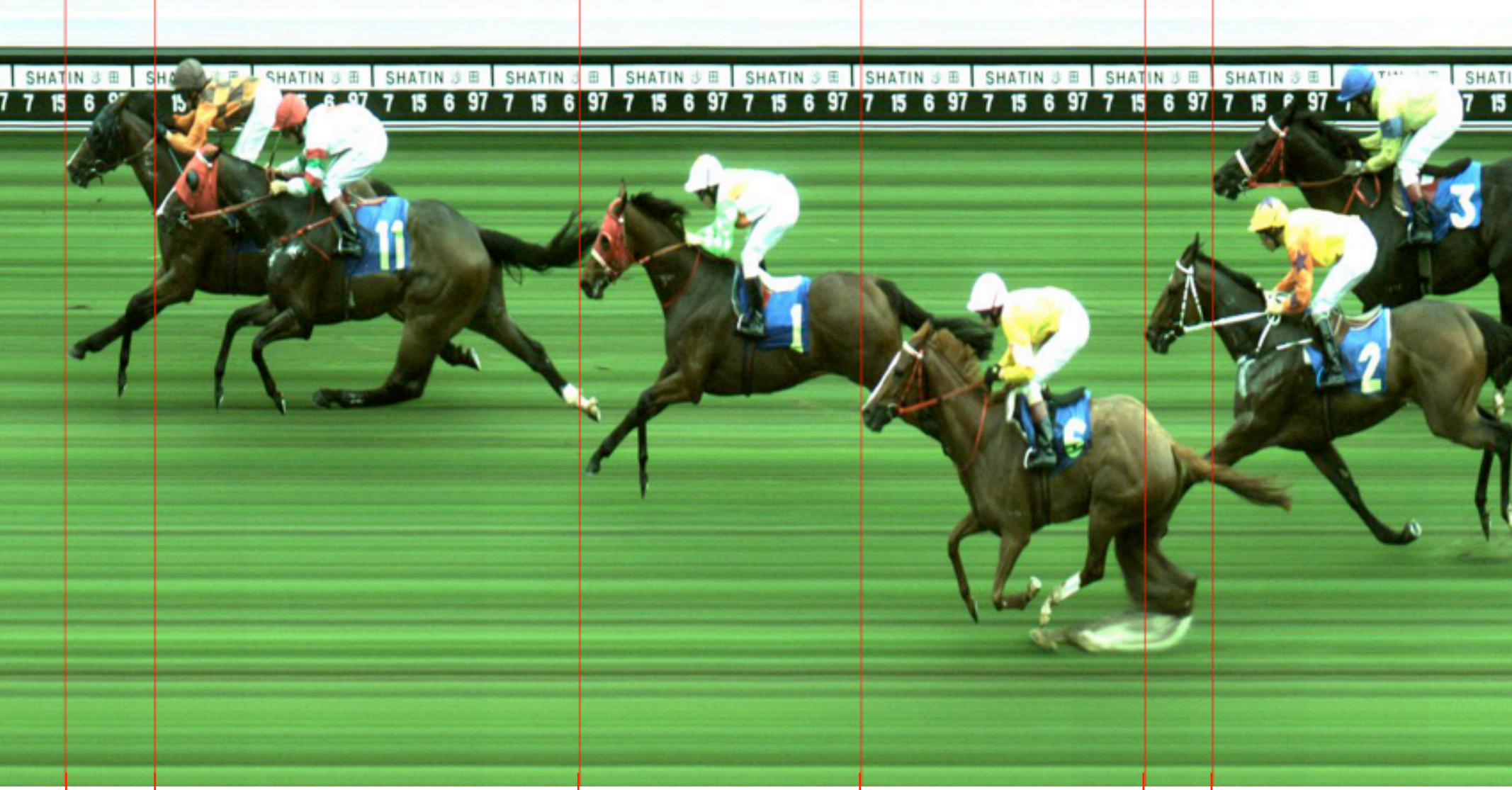
Standard errors

R code 6.24

```
( milk.models <- compare( m6.11 , m6.12 , m6.13 , m6.14 ) )
```

```
        WAIC pWAIC dWAIC weight   SE  dSE
m6.14 -15.0   4.8   0.0   0.93 7.54   NA
m6.11  -8.3   1.8   6.7   0.03 4.52 7.26
m6.13  -7.9   3.0   7.1   0.03 5.67 5.33
m6.12  -6.2   2.9   8.9   0.01 4.34 7.57
```





$WAIC_A$ $WAIC_B$ $WAIC_C$ $WAIC_D$ $WAIC_E$ $WAIC_F$

Comparing estimates

- Always learn more from set of models than any one model
- Compare estimates to help understand differences in model performance

R code 6.27

```
coeftab(m6.11,m6.12,m6.13,m6.14)
```

	m6.11	m6.12	m6.13	m6.14
a	0.66	0.35	0.71	-1.09
log.sigma	-1.79	-1.80	-1.85	-2.16
bn	NA	0.45	NA	2.79
bm	NA	NA	-0.03	-0.10
nobs	17	17	17	17

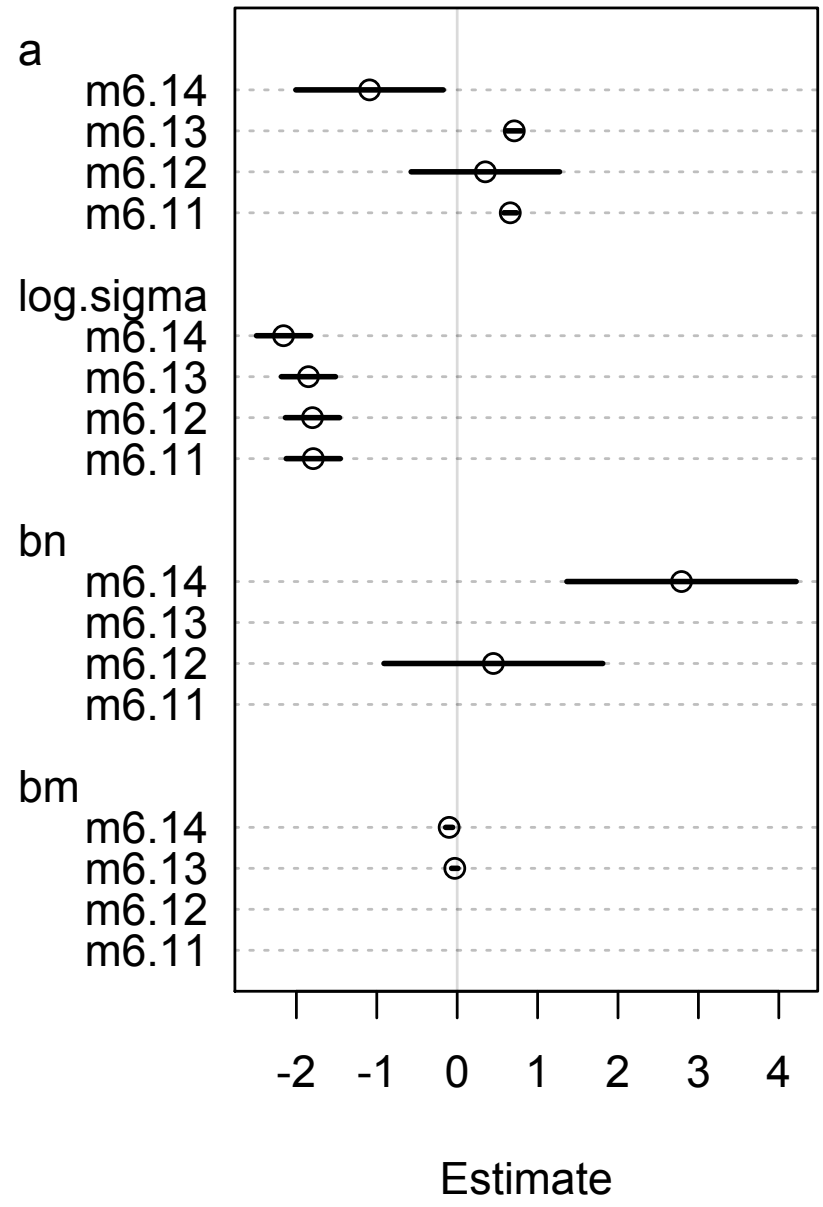


Figure 6.12

R code 6.27

```
coeftab(m6.11,m6.12,m6.13,m6.14)
```

	m6.11	m6.12	m6.13	m6.14
a	0.66	0.35	0.71	-1.09
log.sigma	-1.79	-1.80	-1.85	-2.16
bn	NA	0.45	NA	2.79
bm	NA	NA	-0.03	-0.10
nobs	17	17	17	17

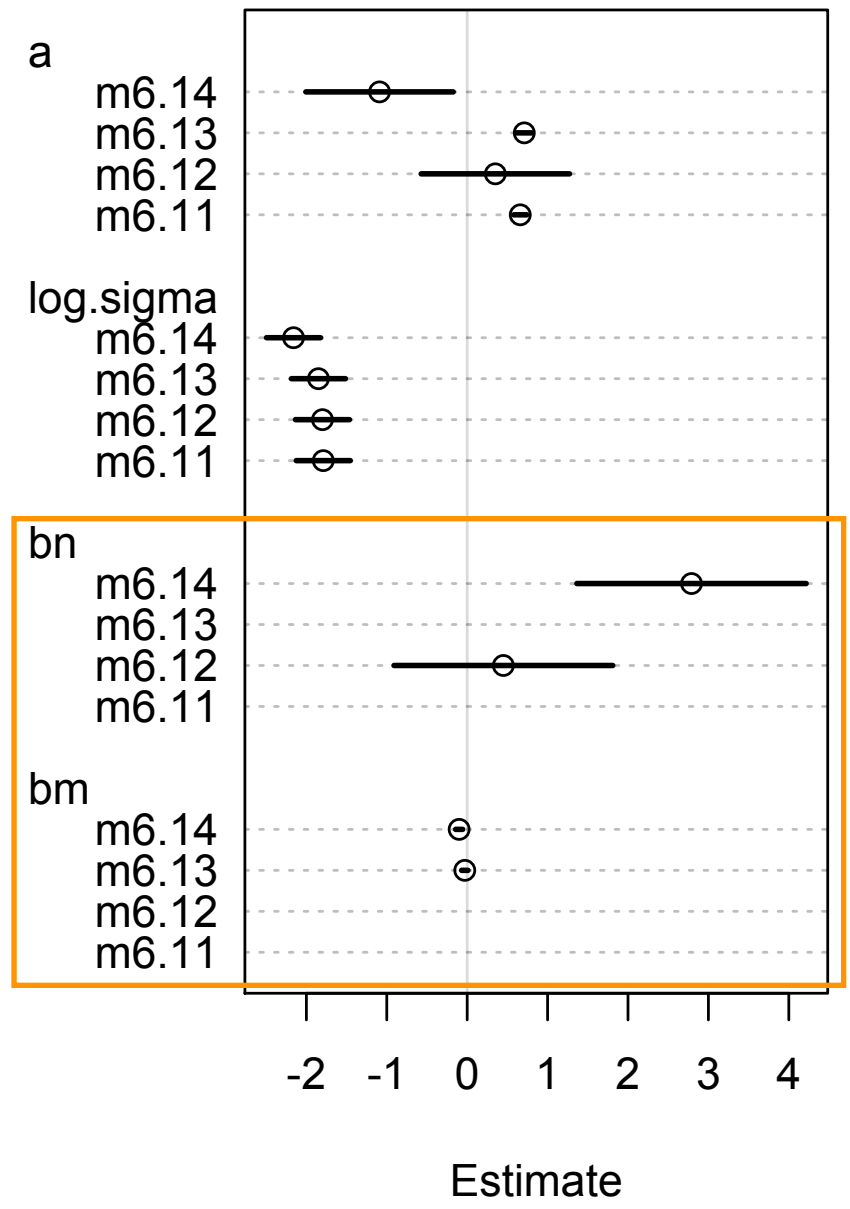
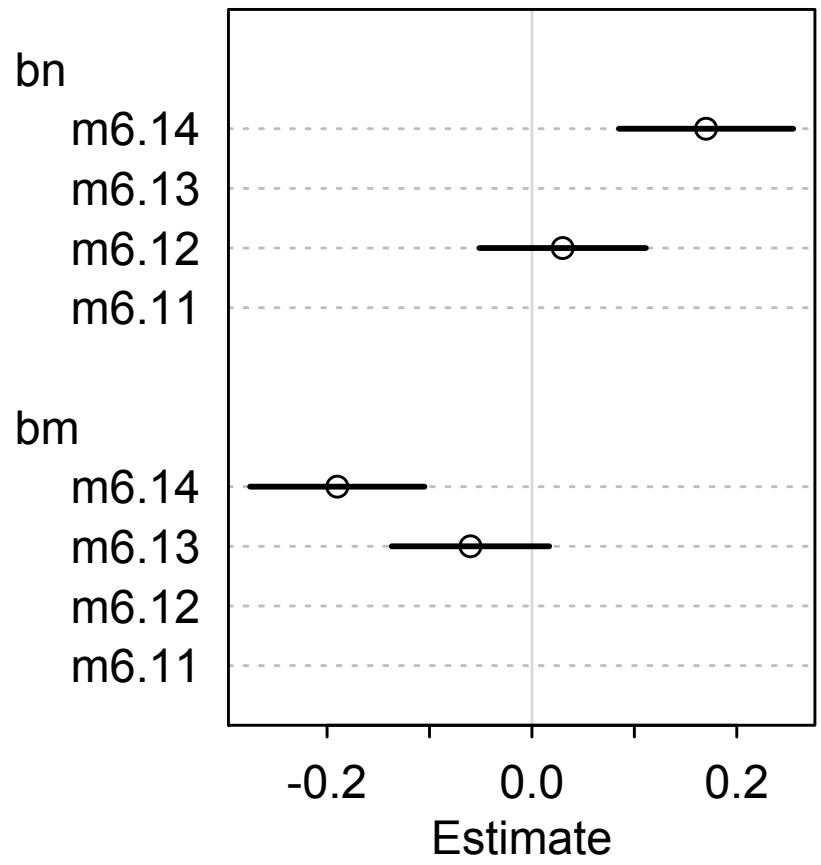


Figure 6.12

Standardized predictors help

```
plot( coeftab(m6.11,m6.12,m6.13,m6.14),pars=c("bn","bm") )
```



Still better to contrast predictions, not estimates

Model averaging

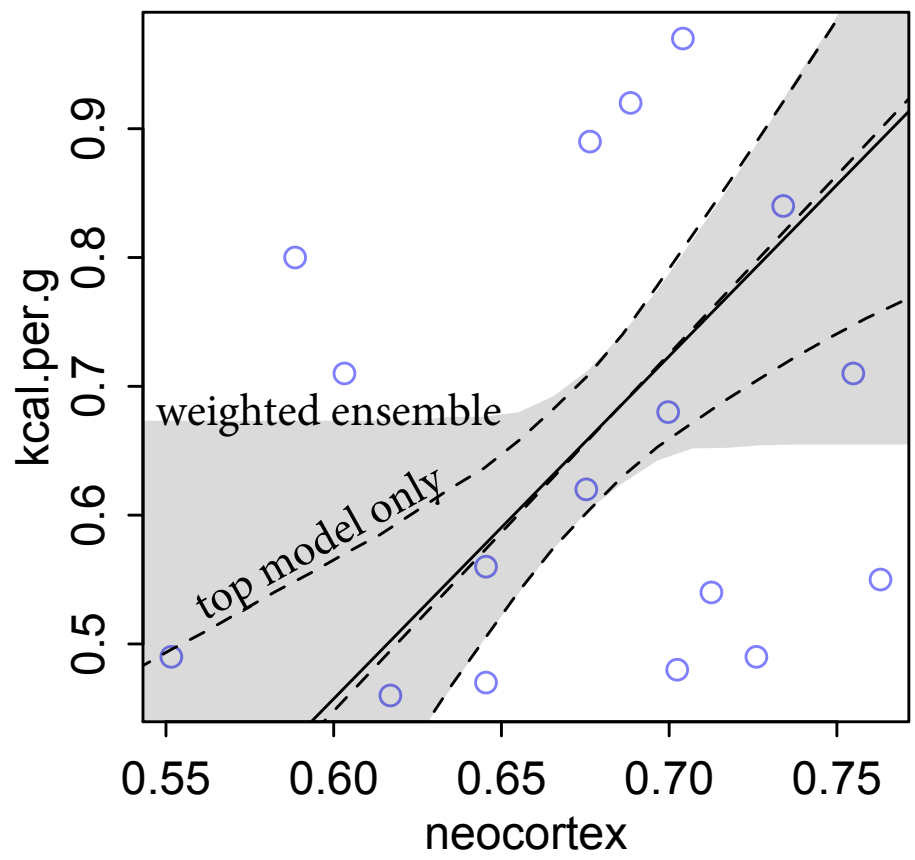
- When computing predictions, average over posterior
- For more than one model, can average the averages
- Do not average parameter estimates, just predictions
  - Because parameters in different models live in different small worlds => don't mean same thing, even if named same thing
  - But predictions reference common large world

Model averaging

- Model averaging procedure
  - Compute information weight for each model
  - Compute distribution of predictions for each model
  - Mix predictions using model weights
- Result is one kind of *prediction ensemble*
- Such ensembles can outperform single-model predictions

```
# compute counterfactual predictions
# neocortex from 0.5 to 0.8
nc.seq <- seq(from=0.5,to=0.8,length.out=30)
d.predict <- list(
    kcal.per.g = rep(0,30), # empty outcome
    neocortex = nc.seq,     # sequence of neocortex
    mass = rep(4.5,30)      # average mass
)
pred.m6.14 <- link( m6.14 , data=d.predict )
mu <- apply( pred.m6.14 , 2 , mean )
mu.PI <- apply( pred.m6.14 , 2 , PI )

# plot it all
plot( kcal.per.g ~ neocortex , d , col=rangi2 )
lines( nc.seq , mu , lty=2 )
lines( nc.seq , mu.PI[1,] , lty=2 )
lines( nc.seq , mu.PI[2,] , lty=2 )
```

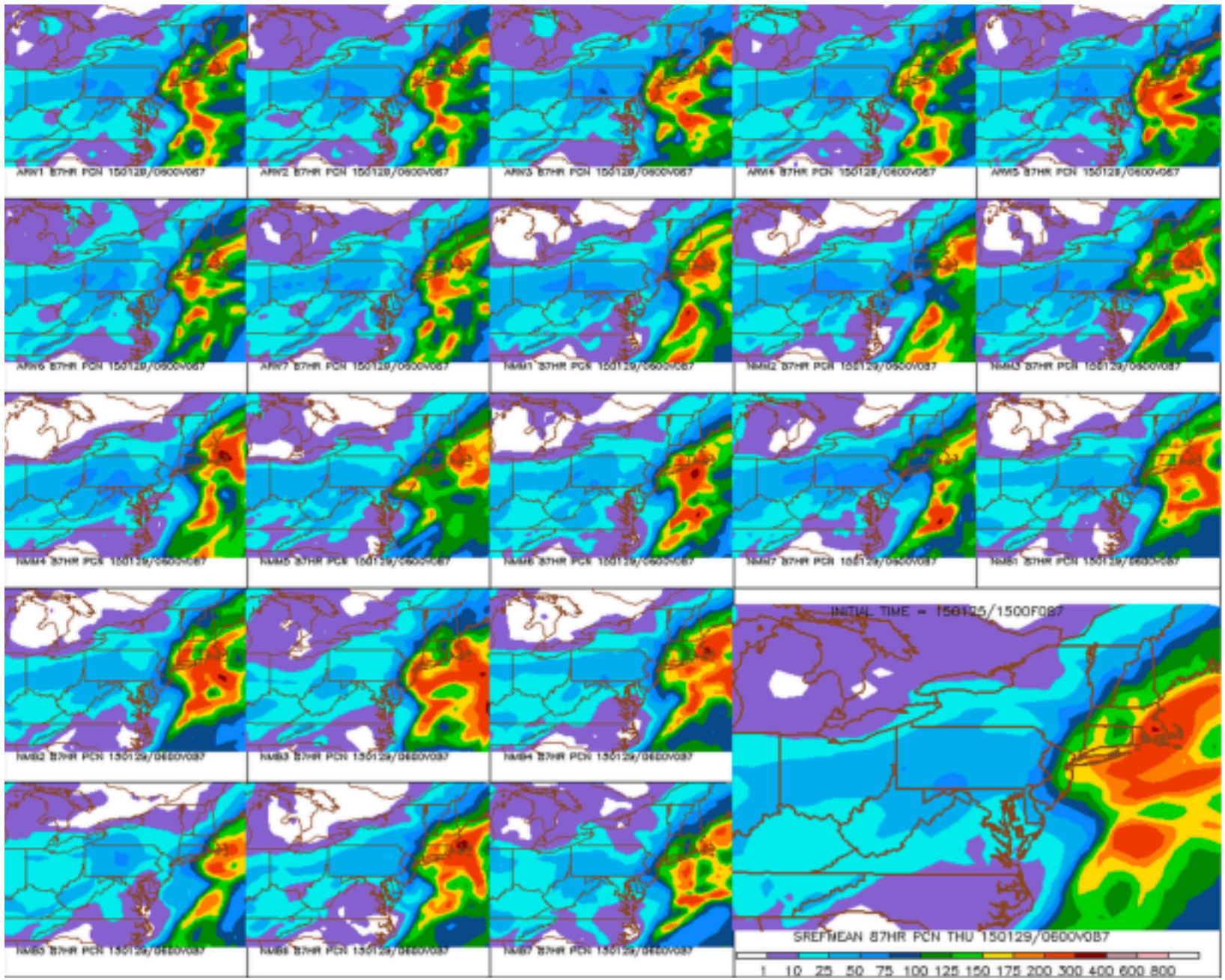


R code 6.30

```
milk.ensemble <- ensemble( m6.11 , m6.12 , m6.13 , m6.14 , data=d.predict )
mu <- apply( milk.ensemble$link , 2 , mean )
mu.PI <- apply( milk.ensemble$link , 2 , PI )
lines( nc.seq , mu )
shade( mu.PI , nc.seq )
```

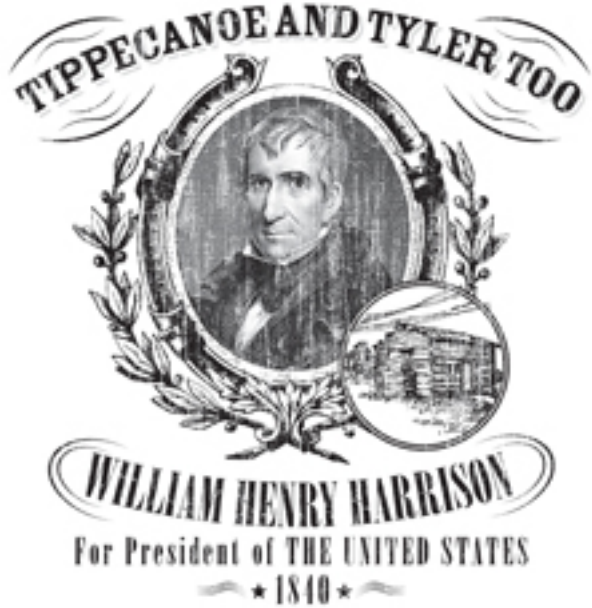
Figure 6.13

World leader in global medium-range numerical weather prediction



Curse of Tippecanoe

- 1840–1960: Every US president elected in year ending in digit "0" died in office
  - W. H. Harrison first, "Old Tippecanoe"
  - Lincoln, Garfield, McKinley, Harding, FD Roosevelt
  - J. F. Kennedy last, assassinated in 1963
  - Reagan broke the curse!
- Trying all possible models: A formula for overfitting
  - Be thoughtful
  - Model averaging mitigates the curse
  - Admit data exploration



Complexity can be good

- Good reasons to use more complex models than AIC/DIC/WAIC recommend
  - Theory says predictor important, so estimate it
  - Lots of sources of variation, but *IC not focused right
  - Simpler model better may mean only that estimate should be smaller => average
- *Consistency* critique has blunt teeth
  - Sometimes noted: As N –> infinity, *IC favors most complex model
  - But as N –> infinity, estimates infinitely precise
  - In hierarchical models, no coherent way N –> infinity?

On the horizon

- Homework: 6H1, 6H2, 6H3
- Next week: Interactions, practicing model comparison
- Week 6: Markov chain Monte Carlo, Maximum entropy, and generalized linear models