Stat 5421 Lecture Notes: Model Selection and Model Averaging

Charles J. Geyer

August 24, 2022

Contents

1 License

2 R

- The version of R used to make this document is 4.2.1.
- The version of the `rmarkdown` package used to make this document is 2.15.
- The version of the `glmbb` package used to make this document is 0.5.1.

3 Information Criteria

3.1 AIC

In the early 1970's Akaike proposed the first information criterion. Later many others were proposed, so Akaike's is now called the Akaike information criterion (AIC).

The "information" in AIC is Kullback-Leibler information, which is the concept of Shannon information imported into statistics, which in turn is the concept of entropy imported from statistical physics into communications theory. It is expected log likelihood, which is what the maximized value of the log likelihood is trying to estimate.

What Akaike discovered is that the maximized value of the log likelihood is a biased estimate of Kullback-Leibler information. It overestimates it by p the dimension of the model (number of parameters). Or, what is the same thing, the likelihood ratio test (LRT) statistic, which is minus twice the (maximized value of the) log likelihood, underestimates its expectation by $2p$. So

$$\text{AIC} = \text{LRT} + 2p$$

All of this is a "large sample size" result based on the "usual" asymptotics of maximum likelihood, which is not valid for all statistical models. But it is always valid for exponential family models in general and categorical data analysis in particular (when sample size is "large").

3.2 AIC Versus Hypothesis Tests

The central dogma of hypothesis testing is "do only one test" or if you do more than one test, you must correct P-values to account for doing more than one test.

So the theory of hypothesis tests is the Wrong Thing for comparing many models. And AIC is the Right Thing or at least *a* Right Thing.

Approaches based on hypothesis testing, such as that implemented by R function `step` come with no guarantees of doing anything correct. They are undeniably TTD (things to do) but have no theoretical justification.

3.3 BIC

In the late 1970's Schwarz proposed another information criterion, which is now usually called the Bayesian information criterion (BIC). Its formula is

$$\text{BIC} = \text{LRT} + \log(n) \cdot p$$

Since $\log(n) \geq 2$ for $n \geq 8$, BIC penalizes larger models more than AIC. BIC always selects smaller models than AIC.

The reason BIC is called "Bayesian" is that, if $\text{BIC}(m)$ denotes the BIC for model m and $g(m)$ denotes the prior probability for model m, then

$$\Pr(m \mid \text{data}) \approx \frac{\exp(-\frac{1}{2}\,\text{BIC}(m))g(m)}{\sum_{m^* \in \mathcal{M}} \exp(-\frac{1}{2}\,\text{BIC}(m^*))g(m^*)} \tag{1}$$

where $\mathcal{M}$ is the class of models under consideration.

This is a "large sample size" result based on the "usual" asymptotics of Bayesian inference (the Bernstein–von Mises theorem), which is not valid for all statistical models. But it is always valid for exponential family models in general and categorical data analysis in particular (when sample size is "large").

When we use a flat prior (g is a constant function of m), the prior g cancels out of the formula, and we obtain

$$\text{BIC}(m) \approx -2 \log \Pr(m \mid \text{data}) + \text{a constant}$$

Clearly, BIC is defined the way it is to be comparable to AIC, not to produce the simplest Bayesian formulas.

3.4 AIC Versus BIC

In model selection AIC and BIC do two different jobs. No selection criterion can do both jobs (Yang, 2005, DOI:10.1093/biomet/92.4.937).

- BIC provides consistent model selection when the true unknown model is among the models under consideration.
- AIC is minimax-rate optimal for estimating the regression function and other probabilities and expectations. It does not need the true unknown model to be among the models under consideration.

Assuming the true unknown model to be among the models under consideration, and Bayesians have to assume this — not among the models under consideration means prior probability zero and posterior probability zero — selecting the model with smallest BIC will select the true unknown model with probability that goes to one as n goes to infinity. Of course, that does not mean BIC is guaranteed to select the correct model at any finite sample size.

If we do not assume the true unknown model is among the models under consideration, then we only have AIC as an option. It generally does not do consistent model selection. However, it does give the best predictions of probabilities and expectations of random variables in the model. It is using the models under consideration to give the best predictions of probabilities and expectations under the true unknown model (which need not be among the models under consideration).

In short,

- use BIC when the true unknown model is assumed to be among the models under consideration, but
- use AIC when we do not want to assume this.

A shorthand often used is "sparsity". The *sparsity* assumption is that the true unknown model has only a few parameters and is one of the models under consideration. Under sparsity, BIC does consistent model selection.

If you do not want to assume sparsity, then use AIC.

3.5 Other Information Criteria

Many other information criteria have been proposed. We will not cover any of them except to note that R function `glmbb` in R package `glmbb` also allows for so-called corrected AIC, denoted AIC_c. The help page for that function says that this has no justification for categorical data analysis, so we will not use it.

4 Considering All Possible Models

The branch and bound algorithm allows consideration of all possible models (or all models in some large class) without actually fitting all of the models. The trick is in the "bounds". Sometimes one can tell that all models either having or lacking a certain term in their formula will fail to be anywhere near the best model found so far. Thus they can be ignored. This is what R function `glmbb` (for GLM branch and bound) does. Even when there are thousands of models, it can consider all of them in a reasonable amount of computer time.

The branch and bound algorithm is not magic, however. When there are billions and billions of models, it may be too slow.

5 Model Selection Versus Model Averaging

Model selection is a mug's game. When there are many models under consideration, the probability of selecting the correct model may be very small, even when the correct model (true unknown model) is among the models under consideration.

True, BIC selects the correct model with probability going to one as sample size goes to infinity, but for any finite sample size, this probability may be small. And AIC does not even guarantee that.

Also, model selection is just the Wrong Thing if you think like a Bayesian. Bayesians apply Bayes' rule, and that provides posterior probabilities, which are approximated by (1). So Bayesians should not select models, they should average over all models according to posterior probability. This is called *Bayesian model averaging* (BMA). For a good introduction see the paper by Hoeting, et al. (1999, doi:10.1214/ss/1009212519, unfortunately this paper had a "printing malfunction" that caused a lot of minus signs and left parentheses to be omitted, a corrected version is available at http://www.stat.washington.edu/www/research/online/hoeting1999.pdf).

To do full BMA one usually needs to do MCMC with multiple models. That can be done with R function `temper` in R package `mcmc` but we are not going to cover that. The package vignette bfst.pdf shows how to do that, but we will not cover it in this course.

A fairly good approximation (if the sample size is large) to BMA uses BIC. Let $\hat{\theta}_m$ denote the maximum likelihood estimate parameter vector for model m, and let $E(W \mid m, \hat{\theta}_m)$ be the expected value of some function W of the data assuming model m is correct and the true unknown parameter value is $\hat{\theta}_m$. Then

$$E(W \mid \text{data}) \approx \frac{\sum_{m \in \mathcal{M}} E(W \mid m, \hat{\theta}_m) \exp(-\frac{1}{2} \operatorname{BIC}(m)) g(m)}{\sum_{m^* \in \mathcal{M}} \exp(-\frac{1}{2} \operatorname{BIC}(m^*)) g(m^*)} \tag{2}$$

for large n. The logic is that for large sample sizes the posterior distribution of the within model parameter θ_m will be concentrated near $\hat{\theta}_m$. So $E(W \mid m, \hat{\theta}_m)$ is a good approximation to $E(W \mid m)$.

Frequentists can also play the model averaging game. Even though they do not buy Bayesian arguments, they can see that model selection is a mug's game. If model selection is very unlikely to pick the best model (which is always the case when there are many models and the sample size is not humongous), then averaging over good models is better. There are many proposed ways to do frequentist model averaging (FMA) in the literature, but one simple way (and the only way we will cover in this course is to replace BIC with AIC in (2) giving

$$\hat{w}_{\text{FMA}} \approx \frac{\sum_{m \in \mathcal{M}} E(W \mid m, \hat{\theta}_m) \exp(-\frac{1}{2} \operatorname{AIC}(m)) g(m)}{\sum_{m^* \in \mathcal{M}} \exp(-\frac{1}{2} \operatorname{AIC}(m^*)) g(m^*)} \tag{3}$$

As an alternative to summing over all possible models in (2) and (3). Madigan and Raftery (1992, DOI:10.1080/01621459.1994.10476894) proposed what they called "Occam's window" in which we only sum over a subset of the models under consideration. Here we only sum over those that have the highest BIC values. This is convenient because that is what R function `glmbb` outputs (only those models within a "cutoff" of the minimum criterion value). The user can choose the "cutoff" as desired. The default cutoff of 10 results in a "weights" $\exp(-\frac{1}{2} \operatorname{BIC}(m))$ that are no less than $\exp(-\frac{1}{2}\text{cutoff})$, which is $\exp(-5) = 0.0067379$ for the default cutoff, relative to the maximum weight.

6 Examples

We will redo some of the examples from other handouts that already used R function `glmbb`.

6.1 High School Student Survey

In the high school student survey data from Chapter 9 in Agresti we saw there were two relatively good models according to AIC

```
library(CatDataAnalysis)
data(table_9.3)
library(glmbb)
out <- glmbb(count ~ a * c * m, data = table_9.3, family = poisson)
summary(out)
```

```
##
## Results of search for hierarchical models with lowest AIC.
## Search was for all models with AIC no larger than min(AIC) + 10
## These are shown below.
##
##   criterion  weight  formula
##   63.42      0.6927  count ~ a*c + a*m + c*m
##   65.04      0.3073  count ~ a*c*m
```

Now we know how to interpret the weights. They are $\exp(-\frac{1}{2}\,\mathrm{AIC}(m))$ normalized to sum to one.

```
sout <- summary(out)
sout$results
```

```
##   criterion    weight                formula
## 1  63.41741 0.6927499 count ~ a*c + a*m + c*m
## 2  65.04343 0.3072501           count ~ a*c*m
```

```
aic <- sout$results$criterion
w <- exp(- aic / 2)
w <- w / sum(w)
all.equal(sout$results$weight, w)
```

```
## [1] TRUE
```

If we treat these as vaguely like posterior probabilities (they are the FMA competitor to posterior probabilities), then they say the "best" model is not so much better than the second "best" that we should not assume the "best" model is correct.

We can also use BIC.

```
out.bic <- glmbb(count ~ a * c * m, data = table_9.3, family = poisson,
    criterion = "BIC", BIC.option = "sum")
summary(out.bic)
```

```
##
## Results of search for hierarchical models with lowest BIC.
## Search was for all models with BIC no larger than min(BIC) + 10
## These are shown below.
##
##   criterion  weight   formula
##   103.5      0.97535  count ~ a*c + a*m + c*m
##   110.9      0.02465  count ~ a*c*m
```

We see that BIC is much more favorable to the smaller (more parsimonious) model than AIC is. BIC puts probability 0.97535 on the "best" model but AIC puts only probability 0.69275 on this model. In this example they agree on the "best" model, but generally these two criteria do not agree.

The optional argument `BIC.option = "sum"` to R function `glmbb` is explained in the help for that function. We should always use it when doing categorical data analysis. It perhaps should be the default, but is not for reasons of backward compatibility.

6.2 Seat Belt Use

In the seat belt use data from Chapter 9 in Agresti we saw there were many relatively good models according to AIC

```
# clean up R global environment
rm(list = ls())
```

```
count <- c(7287, 11587, 3246, 6134, 10381, 10969, 6123, 6693,
            996,   759,  973,  757,   812,   380, 1084,  513)
injury <- gl(2, 8, 16, labels = c("No", "Yes"))
gender <- gl(2, 4, 16, labels = c("Female", "Male"))
location <- gl(2, 2, 16, labels = c("Urban", "Rural"))
seat.belt <- gl(2, 1, 16, labels = c("No", "Yes"))

library(glmbb)
out.aic <- glmbb(count ~ seat.belt * injury * location * gender,
    family = "poisson")
summary(out.aic)
```

```
##
## Results of search for hierarchical models with lowest AIC.
## Search was for all models with AIC no larger than min(AIC) + 10
## These are shown below.
##
##   criterion  weight   formula
##   182.8      0.24105  count ~ seat.belt*injury*location + seat.belt*location*gender + injury*locatio
##   183.1      0.21546  count ~ injury*gender + seat.belt*injury*location + seat.belt*location*gender
##   184.0      0.13742  count ~ seat.belt*injury + seat.belt*location*gender + injury*location*gender
##   184.8      0.09055  count ~ seat.belt*injury*location + seat.belt*injury*gender + seat.belt*locati
##   184.9      0.08446  count ~ seat.belt*injury + injury*location + injury*gender + seat.belt*locatio
##   185.0      0.08042  count ~ seat.belt*injury*location + seat.belt*injury*gender + seat.belt*locati
##   185.5      0.06462  count ~ seat.belt*injury*location*gender
##   185.8      0.05365  count ~ seat.belt*injury*gender + seat.belt*location*gender + injury*location*g
##   186.8      0.03237  count ~ injury*location + seat.belt*injury*gender + seat.belt*location*gender
```

Now we know how to interpret the weights. We see that no model gets a majority of the weight. None is highly probable to be the best model. There are just too many models under consideration for that to happen.

This, of course, assumes that in FMA we want to use a flat "prior" (because frequentists don't like priors), that is, we want $g(m) = 1$ for all m in (3).

We can also use BIC.

```
out.bic <- glmbb(count ~ seat.belt * injury * location * gender,
    family = "poisson", criterion = "BIC", BIC.option = "sum")
summary(out.bic)
```

```
##
## Results of search for hierarchical models with lowest BIC.
## Search was for all models with BIC no larger than min(BIC) + 10
## These are shown below.
##
##   criterion  weight   formula
##   294.6      0.87998  count ~ seat.belt*injury + injury*location + injury*gender + seat.belt*locatio
##   299.3      0.08189  count ~ seat.belt*injury + seat.belt*location + injury*location + seat.belt*ge
##   301.8      0.02328  count ~ injury*gender + seat.belt*injury*location + seat.belt*location*gender
##   302.7      0.01485  count ~ seat.belt*injury + seat.belt*location*gender + injury*location*gender
```

There is a big difference between the analyses.

- BIC does put the majority of the posterior probability on one model (assuming flat prior on models).
- That best model according to BIC weight is number 5 according to AIC weight. Conversely, the best

model according to AIC weight is below the cutoff according to BIC weight and so does not appear on the BIC list.

- BIC plus Occam's window spreads out the posterior probability over fewer models than AIC + Occam's window.

If we compute expected cell counts according to FMA and BMA, they should be rather different.

It is not very obvious how to do FMA without refitting all the models, but all the model fits are stored in the R environment `out.aic$envir`

```
ls(envir = out.aic$envir)
```

```
##  [1] "min.crit"
##  [2] "sha1.0b21bc716a36c9104619cad165cd8e1ee59eaa67"
##  [3] "sha1.0bc71e279edad3fdfd08de9a0b9e0ae111052d6e"
##  [4] "sha1.1178bbd4b48158d0044b3a1347e41003680ed4cb"
##  [5] "sha1.1394de7910da632a1171a5370b549df8b89df21c"
##  [6] "sha1.17fd74be284b4fb9f9500b30852ef16f55a5951d"
##  [7] "sha1.2d0982bf246f88e37a97b49bdff7be4da5f4e1e2"
##  [8] "sha1.34b3afacb24ed6a05f656993097cd2466e194c33"
##  [9] "sha1.34b64a787d14cbac65556ecb290973c935a63910"
## [10] "sha1.364675bd9f91f1dd81e9b4f78723c7c893d02fb8"
## [11] "sha1.39061969f2e696a74694e6e643c158b68d571886"
## [12] "sha1.39cd3d0327e6002b8be8fad301dfbaafb2c24eaa"
## [13] "sha1.4855a747811e1cb6a94a5275898fb6497b0bebe4"
## [14] "sha1.4d6583bbf0d0862415f7bacafd6b00679c515c0c"
## [15] "sha1.527e582a9d862405ddf2c05b6f09c890212c3c0d"
## [16] "sha1.66b77d3f388eae1ce33b62b466e8f311bffa469a"
## [17] "sha1.66cfba8ba57104e87efbfc4ba75ca67551816543"
## [18] "sha1.6aa8694584f0ebadef108c9e6684b32fc9aef678"
## [19] "sha1.6f40df9c7712ee3618c90d56f68d7162b5d06e1a"
## [20] "sha1.724e48559c4e25c9dff6641a26069457726036c2"
## [21] "sha1.7ac914cd162e41bc044ba9fcdb9bddb00182abcf"
## [22] "sha1.7f9bf47ef1ae0aecfaa935614a3164c4cf1d385f"
## [23] "sha1.893a4536650ea9f4ec0f81782104f20e335738a9"
## [24] "sha1.8bca01f0d7b4ba675cb4e25bacfb13f06ce1bf29"
## [25] "sha1.932904ab5c59667c0edab04b0aff8e87d5eef820"
## [26] "sha1.a1f53c04447266ee85267edfcd62af50dedafa41"
## [27] "sha1.a449f95d9878915ee202f828977904a33819389b"
## [28] "sha1.a9eb2e4d8dbbfd65464afdc18c87c99ef16585aa"
## [29] "sha1.bd24727fa981859346a3b3ebf6fd8f1935d7f5c7"
## [30] "sha1.cba6840ac60dcc46de77bbf74a4c06e5e485ebb4"
## [31] "sha1.cbe1125be84ccb49411ef716ffe59218bb01436f"
## [32] "sha1.d21240256fc4cc009cb9b6ecd2ae465d2bc920b3"
## [33] "sha1.d9dd41b6107367ab849386daddf27f856bac8e82"
## [34] "sha1.db6cf35ff8bebb4c28e8baec502ae4e5c1365d35"
## [35] "sha1.e2f8395790cec12e6b92e36a338b2dbba7093932"
## [36] "sha1.f02e9c53c00058688f875f16b7024ce5d063e84c"
## [37] "sha1.f24e60e6005914cdeda0fdc45cabfcb700e2c567"
## [38] "sha1.f7cc3ef833144b019cbd8726f7865841afce6b64"
```

All of the R objects whose names begin with `sha1.` are the model fits. The object `min.crit` is the same as `out.aic$min.crit`

```
identical(out.aic$min.crit, out.aic$envir$min.crit)
```

```
## [1] TRUE
```

And, although this is not documented, we can look in the code for R function `summary.glmbb` to see how to extract the criteria from these models.

```
min.crit <- out.aic$min.crit
cutoff <- out.aic$cutoff
e <- out.aic$envir
rm("min.crit", envir = e)
criterion <- unlist(eapply(e, "[[", "criterion"))
is.in.window <- criterion <= min.crit + cutoff
w <- criterion[is.in.window]
w <- exp(- w / 2)
w <- w / sum(w)
moo <- eapply(e, "[[", "fitted.values")
moo <- moo[is.in.window]
moo <- as.data.frame(moo)
moo <- as.matrix(moo)
mu.hat <- drop(moo %*% w)
foo <- data.frame(injury = injury, gender = gender, location = location,
    seat.belt = seat.belt, observed = count, expected.fma = mu.hat)
print(foo, digits=5)
```

```
##    injury gender location seat.belt observed expected.fma
## 1      No Female    Urban        No     7287      7277.39
## 2      No Female    Urban       Yes    11587     11608.43
## 3      No Female    Rural        No     3246      3252.80
## 4      No Female    Rural       Yes     6134      6115.39
## 5      No   Male    Urban        No    10381     10380.36
## 6      No   Male    Urban       Yes    10969     10957.83
## 7      No   Male    Rural        No     6123      6126.46
## 8      No   Male    Rural       Yes     6693      6701.35
## 9     Yes Female    Urban        No      996      1005.61
## 10    Yes Female    Urban       Yes      759       737.57
## 11    Yes Female    Rural        No      973       966.20
## 12    Yes Female    Rural       Yes      757       775.61
## 13    Yes   Male    Urban        No      812       812.64
## 14    Yes   Male    Urban       Yes      380       391.17
## 15    Yes   Male    Rural        No     1084      1080.54
## 16    Yes   Male    Rural       Yes      513       504.65
```

Note this is frequentist model averaging. The expected cell counts reported in the last column do not correspond to the prediction of any model. They are a weighted average of the predictions of all models that pass through "Occam's window" (that are reported by `summary(out.aic)`). And the weights are the weights reported by `summary(out.aic)`.

And, if we redo the same calculation using BMA rather than FMA, we get

```
min.crit <- out.bic$min.crit
cutoff <- out.bic$cutoff
e <- out.bic$envir
rm("min.crit", envir = e)
criterion <- unlist(eapply(e, "[[", "criterion"))
is.in.window <- criterion <= min.crit + cutoff
w <- criterion[is.in.window]
w <- exp(- w / 2)
w <- w / sum(w)
moo <- eapply(e, "[[", "fitted.values")
```

```
moo <- moo[is.in.window]
moo <- as.data.frame(moo)
moo <- as.matrix(moo)
mu.hat <- drop(moo %*% w)
foo <- data.frame(foo, expected.bma = mu.hat)
print(foo, digits=5)
```

```
##    injury gender location seat.belt observed expected.fma expected.bma
## 1      No Female    Urban        No     7287      7277.39      7264.64
## 2      No Female    Urban       Yes    11587     11608.43     11641.34
## 3      No Female    Rural        No     3246      3252.80      3262.64
## 4      No Female    Rural       Yes     6134      6115.39      6085.38
## 5      No   Male    Urban        No    10381     10380.36     10368.87
## 6      No   Male    Urban       Yes    10969     10957.83     10949.15
## 7      No   Male    Rural        No     6123      6126.46      6140.85
## 8      No   Male    Rural       Yes     6693      6701.35      6707.13
## 9     Yes Female    Urban        No      996      1005.61      1008.24
## 10    Yes Female    Urban       Yes      759       737.57       714.78
## 11    Yes Female    Rural        No      973       966.20       966.48
## 12    Yes Female    Rural       Yes      757       775.61       795.50
## 13    Yes   Male    Urban        No      812       812.64       834.25
## 14    Yes   Male    Urban       Yes      380       391.17       389.73
## 15    Yes   Male    Rural        No     1084      1080.54      1056.03
## 16    Yes   Male    Rural       Yes      513       504.65       508.99
```

Even though the `fitted.values` components of the model fits are the same in FMA and BMA, the weighted averages are different because the weights are different and because different numbers of models are given any weight at all, for FMA and for BMA.

If one wanted to go on and calculate other things that are functions of these expected values, like conditional odds ratios that Agresti computes, then they should be based on the FMA or BMA values computed above.

If one is doing FMA, then standard errors for these FMA estimators can be computed as described by Burnham and Anderson, Section 4.3.2. But we will not give examples of that here.

If one wants Bayesian posterior distributions for these expectations, one would have to do full BMA via MCMC as described in the aforementioned vignette bfst.pdf in R package `mcmc`. But, as also aformentioned, we will not be discussing that in this course.

6.3 Alligator Food Choice

6.3.1 AIC

In the alligator food choice data from Chapter 8 in Agresti we saw there were many relatively good models according to AIC

```
# clean up R global environment
rm(list = ls())

library(CatDataAnalysis)
data(table_8.1)
foo <- transform(table_8.1,
    lake = factor(lake,
        labels = c("Hancock", "Oklawaha", "Trafford", "George")),
    gender = factor(gender, labels = c("Male", "Female")),
    size = factor(size, labels = c("<=2.3", ">2.3")),
    food = factor(food,
```

```
        labels = c("Fish", "Invertebrate", "Reptile", "Bird", "Other")))
out.aic <- glmbb(big = count ~ lake * gender * size * food,
    little = ~ lake * gender * size + food,
    family = poisson, data = foo)
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```
## Warning: glm.fit: fitted rates numerically 0 occurred
```

```
summary(out.aic)
```

```
##
## Results of search for hierarchical models with lowest AIC.
## Search was for all models with AIC no larger than min(AIC) + 10
## These are shown below.
##
##   criterion  weight    formula
##   288.0      0.903304  count ~ lake*food + size*food + lake*gender*size
##   293.7      0.050072  count ~ lake*food + gender*food + size*food + lake*gender*size
##   294.9      0.028388  count ~ lake*gender*size + lake*size*food
##   296.8      0.010643  count ~ size*food + lake*gender*size + lake*gender*food
##   297.5      0.007593  count ~ gender*food + lake*gender*size + lake*size*food
```

6.3.2 BIC

And now we can also do BIC.

```
out.bic <- glmbb(big = count ~ lake * gender * size * food,
    little = ~ lake * gender * size + food,
    family = poisson, data = foo,
    criterion = "BIC", BIC.option = "sum")
summary(out.bic)
```

```
##
## Results of search for hierarchical models with lowest BIC.
## Search was for all models with BIC no larger than min(BIC) + 10
## These are shown below.
##
##   criterion  weight   formula
##   388.0      0.96096  count ~ food + lake*gender*size
##   394.4      0.03904  count ~ size*food + lake*gender*size
```

This is quite a shock. It says that the empty model that says `food` is not associated with `lake`, `gender`, or `size` or any interaction of them is the best model (according to BIC). If you think like a Bayesian, perhaps there is nothing much going on in these data.