Finite Mixture Models

Joshua Landman

20 November 2018

Finite Mixture Models

- What are they used for?
- How do they work?
- Why do we care about them?
- Let's find out!

What are mixture models used for?

1. Predictive inference (not causal)
2. Comparing a finite number of M different models
3. Choosing the best model(s) for a set of observations

The Main Idea

Key points:

1. We don't initially know which model an observation is generated from
2. We have a set of M models we want to compare and choose from
3. The models can be parametrized any way that makes sense (i.e., we can test theories whose models have different underlying distributions)

The Model

$f_m(y|x, \theta_m) \rightarrow$ model m, where y is its outcome, x its covariate vector, and θ_m its parameters.

We have latent variable $Z_i \in \{1, 2, \ldots, M\}$. The value of Z_i represents the model an observation i is generated from.

The DGP is given by $Y_i|X_i, Z_i \sim f_{Z\,i}(Y_i, X_i, \theta_{Z\,i})$. Note the similarity between the DGP and model formulation.

How do we select the correct model for an observation?

Assumption: conditional independence over all observations given X and Z.

Observed data likelihood function:

$$L_{\text{obs}}(\Theta, \Pi | \{X_i, Y_i\}_{i=1}^N) = \prod_{i=1}^{N} \left\{ \sum_{m=1}^{M} \pi_m f_m(Y_i | X_i, \theta_m) \right\}$$

Θ is the set of all model parameters. Π is the set of all model probabilities. $\pi_m = \Pr(Z_i = m)$ is the population proportion of observations generated by model m.

π_m measures relative performance of model (theory) m

Fitting a Mixture Model - MLE

- Maximum likelihood estimation via Expectation-Maximization (EM): compute all π_m; maximize weighted log-likelihood functions for each model; repeat until convergence

Fitting a Mixture Model - MLE

- Maximum likelihood estimation via Expectation-Maximization (EM): compute all π_m; maximize weighted log-likelihood functions for each model; repeat until convergence

- Good: L_{obs} is non-decreasing

Fitting a Mixture Model - MLE

- Maximum likelihood estimation via Expectation-Maximization (EM): compute all π_m; maximize weighted log-likelihood functions for each model; repeat until convergence

- Good: L_{obs} is non-decreasing
- Bad: standard errors must be computed separately, computationally expensive to fit, potentially multimodal likelihood for MLE

Expectation-Maximization and Mixture Models

- Alternating E (expectation) and M (maximization) steps until convergence

- E-step: computes conditional expectation of latent variable Z_i:

$$Q(\Theta, \Pi | \Theta^{(t-1)}, \Pi^{(t-1)}, \{X_i, Y_i, Z_i\}_{i=1}^{N}) =$$

$$\sum_{i=1}^{N} \sum_{m=1}^{M} \zeta_{i,m}^{(t-1)} \{\log \pi_m + \log f_m(Y_i | X_i, \theta_m)\}$$

Expectation-Maximization and Mixture Models

- Alternating E (expectation) and M (maximization) steps until convergence

- E-step: computes conditional expectation of latent variable Z_i:

$$Q(\Theta, \Pi | \Theta^{(t-1)}, \Pi^{(t-1)}, \{X_i, Y_i, Z_i\}_{i=1}^{N}) =$$

$$\sum_{i=1}^{N} \sum_{m=1}^{M} \zeta_{i,m}^{(t-1)} \{\log \pi_m + \log f_m(Y_i | X_i, \theta_m)\}$$

- M-step: maximizes the Q function by separately maximizing log-likelihood functions for each model (updating $\pi_m^{(t)}$)

What is $\zeta_{i,m}$?

$$\zeta_{i,m} = Pr(Z_i = m|\Theta^{(t-1)}, \Pi^{(t-1)}, \{X_i, Y_i\}_{i=1}^N)$$
$$= \frac{\pi_m^{(t-1)} f_m(Y_i|X_i, \theta_m)}{\sum_{m=1}^M \pi_m^{(t-1)} f_m(Y_i|X_i, \theta_m)}$$

- Intuitively, this is the ratio of the likelihood that observation i was generated from model m to the sum of all likelihoods for each model.

What is $\zeta_{i,m}$?

$$\zeta_{i,m} = Pr(Z_i = m | \Theta^{(t-1)}, \Pi^{(t-1)}, \{X_i, Y_i\}_{i=1}^{N})$$
$$= \frac{\pi_m^{(t-1)} f_m(Y_i | X_i, \theta_m)}{\sum_{m=1}^{M} \pi_m^{(t-1)} f_m(Y_i | X_i, \theta_m)}$$

- Intuitively, this is the ratio of the likelihood that observation i was generated from model m to the sum of all likelihoods for each model.

- In other words, the posterior probability of observation i having been generated from m.

Fitting a Mixture Model - MCMC

- Markov chain Monte Carlo (Dirichlet priors)

Fitting a Mixture Model - MCMC

- Markov chain Monte Carlo (Dirichlet priors)
- Good: standard errors and uncertainty terms for, e.g., π_m arise naturally

Fitting a Mixture Model - MCMC

- Markov chain Monte Carlo (Dirichlet priors)
- Good: standard errors and uncertainty terms for, e.g., π_m arise naturally
- Bad: mixing can be poor, convergence of MCMC can be very slow

Further Issues with Fitting Mixture Models

- Even if all submodels are identifiable, there is no guarantee that their mixture will be

Further Issues with Fitting Mixture Models

- Even if all submodels are identifiable, there is no guarantee that their mixture will be
- When too many models are included, results are less meaningful

Further Issues with Fitting Mixture Models

- Even if all submodels are identifiable, there is no guarantee that their mixture will be
- When too many models are included, results are less meaningful
- Overfitting ($\pi_m \approx 0$, multiple models with the same parameters)

Further Issues with Fitting Mixture Models

- Even if all submodels are identifiable, there is no guarantee that their mixture will be
- When too many models are included, results are less meaningful
- Overfitting ($\pi_m \approx 0$, multiple models with the same parameters)
- Missing theories

Model Comparison and Observation Consistency

- π_m is the population proportion (MLE or Bayesian) of observations consistent with a model

Model Comparison and Observation Consistency

- π_m is the population proportion (MLE or Bayesian) of observations consistent with a model
- $\sum_{i=1}^{N} \frac{\hat{\zeta}_{i,m}}{N}$ is the sample proportion

Model Comparison and Observation Consistency

- π_m is the population proportion (MLE or Bayesian) of observations consistent with a model
- $\sum_{i=1}^{N} \frac{\hat{\zeta}_{i,m}}{N}$ is the sample proportion
- If we assume that each observation is consistent with exactly one theory, we can use statistically significant consistency:

$$\lambda_m^* = \inf \left\{ \frac{\sum_{i=1}^{N} (1 - \hat{\zeta}_{i,m}) \mathbf{1}\{\hat{\zeta}_{i,m} \geq \lambda_m\}}{\sum_{i=1}^{N} \mathbf{1}\{\hat{\zeta}_{i,m} \geq \lambda_m\} + \prod_{i=1}^{N} \mathbf{1}\{\hat{\zeta}_{i,m} < \lambda_m\}} \leq \alpha_m \right\}$$

Model Comparison and Observation Consistency

- π_m is the population proportion (MLE or Bayesian) of observations consistent with a model
- $\sum_{i=1}^{N} \frac{\hat{\zeta}_{i,m}}{N}$ is the sample proportion
- If we assume that each observation is consistent with exactly one theory, we can use statistically significant consistency:

$$\lambda_m^* = \inf \left\{ \frac{\sum_{i=1}^{N} (1 - \hat{\zeta}_{i,m}) \mathbf{1}\{\hat{\zeta}_{i,m} \geq \lambda_m\}}{\sum_{i=1}^{N} \mathbf{1}\{\hat{\zeta}_{i,m} \geq \lambda_m\} + \prod_{i=1}^{N} \mathbf{1}\{\hat{\zeta}_{i,m} < \lambda_m\}} \leq \alpha_m \right\}$$

- Given a predefined false discovery rate α_m (e.g., 0.05), find the largest set of observations such that this inequality is true.

Model Comparison and Observation Consistency

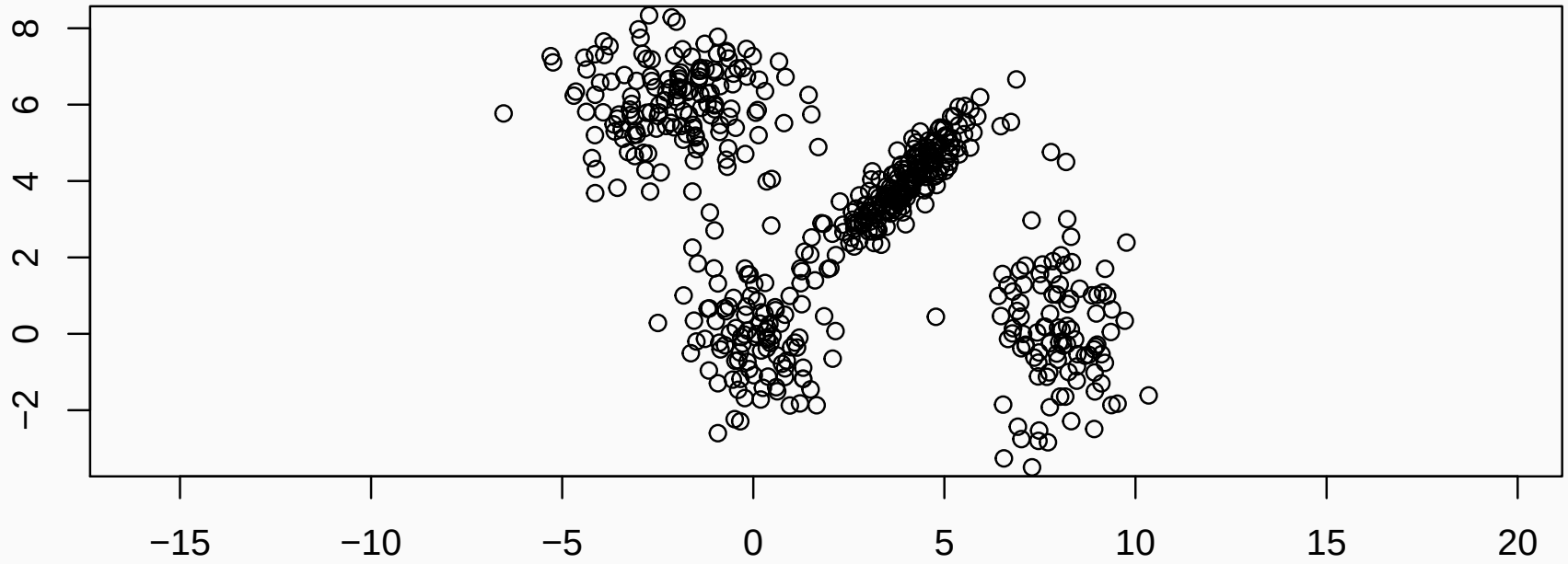
- π_m is the population proportion (MLE or Bayesian) of observations consistent with a model
- $\sum_{i=1}^{N} \frac{\hat{\zeta}_{i,m}}{N}$ is the sample proportion
- If we assume that each observation is consistent with exactly one theory, we can use statistically significant consistency:

$$\lambda_m^* = \inf \left\{ \frac{\sum_{i=1}^{N} (1 - \hat{\zeta}_{i,m}) \mathbf{1}\{\hat{\zeta}_{i,m} \geq \lambda_m\}}{\sum_{i=1}^{N} \mathbf{1}\{\hat{\zeta}_{i,m} \geq \lambda_m\} + \prod_{i=1}^{N} \mathbf{1}\{\hat{\zeta}_{i,m} < \lambda_m\}} \leq \alpha_m \right\}$$

- Given a predefined false discovery rate α_m (e.g., 0.05), find the largest set of observations such that this inequality is true.
- Allows for accounting for theories not included in the mixture model

Example: Model Selection and Fitting

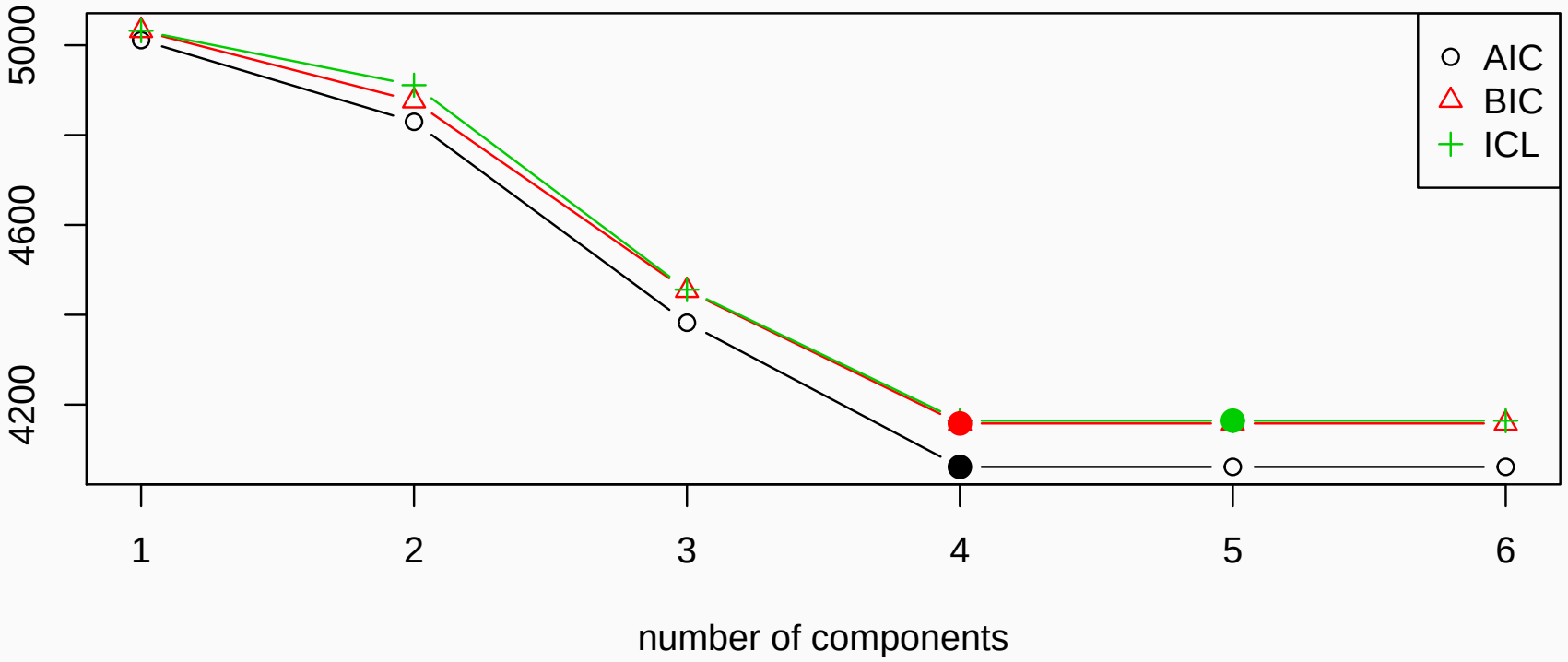
- Goal: fit a mixture model where each of the four submodels is Gaussian
- Tools: `flexmix`, `Nclus`



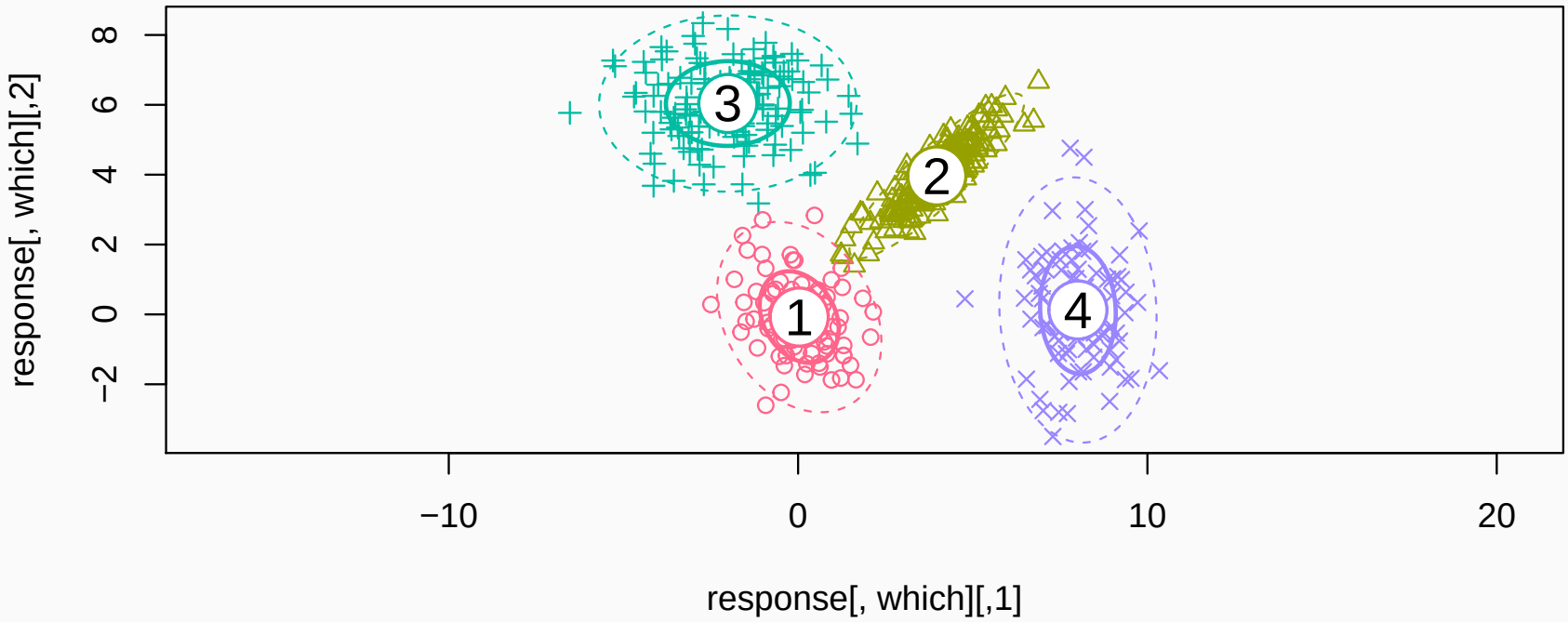
- What if we don't know *a priori* how many competing models/subpopulations there are?

Choosing the Number of Components

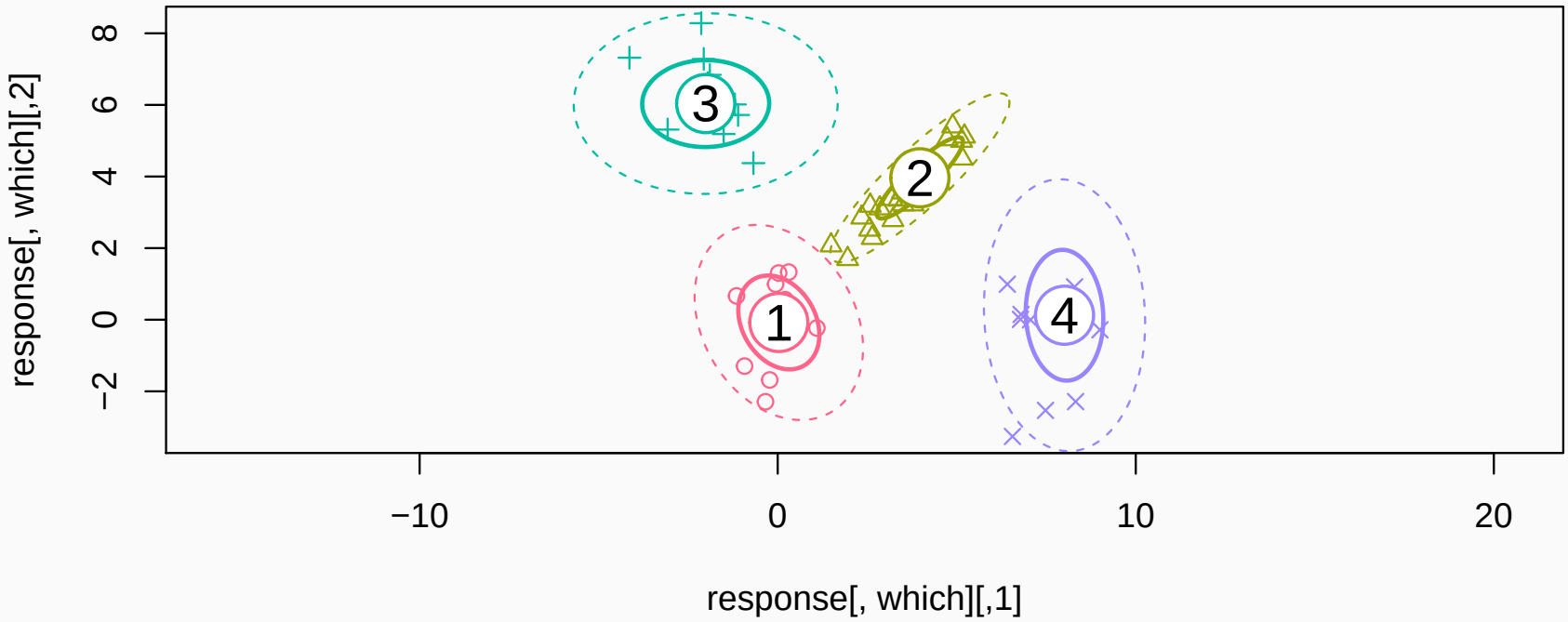
```
model = stepFlexmix(training ~ 1, k = 1:6,
  model = FLXMCmvnorm(diagonal = FALSE))
```



Model Fit Results - 4 Components

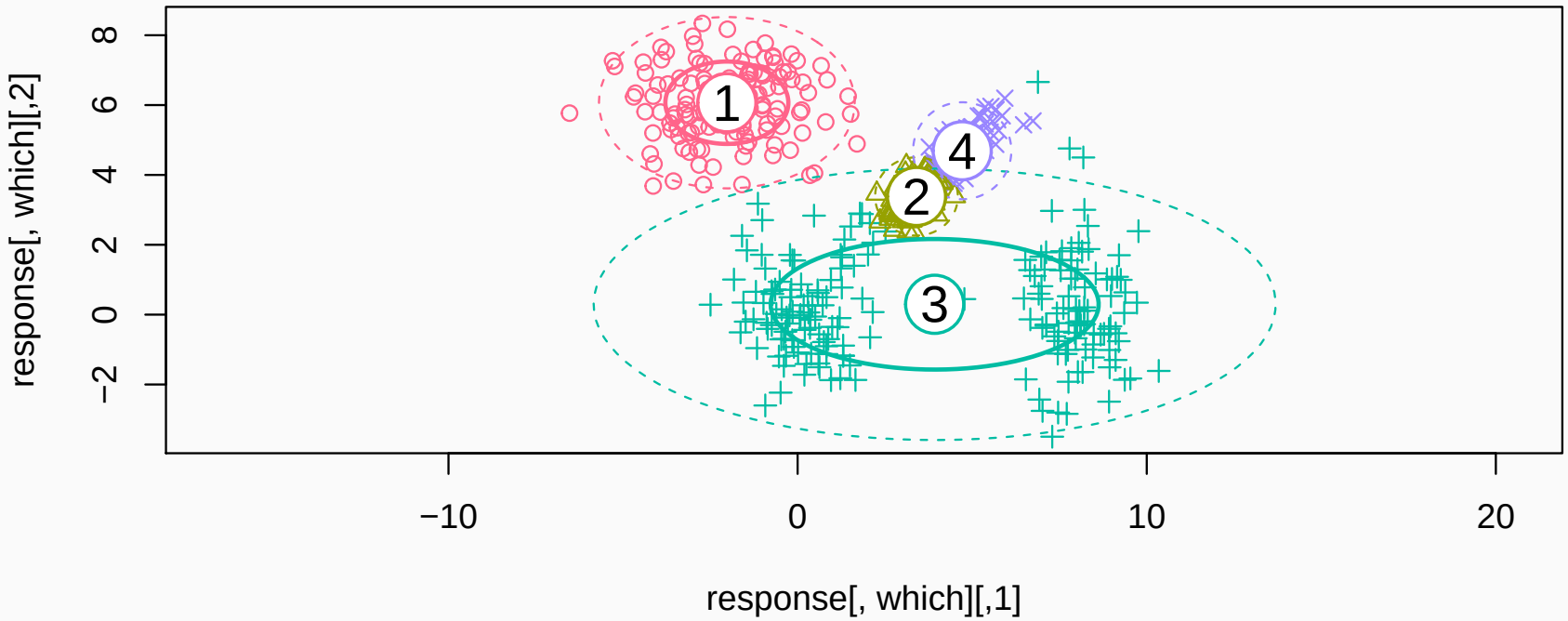


Prediction Results



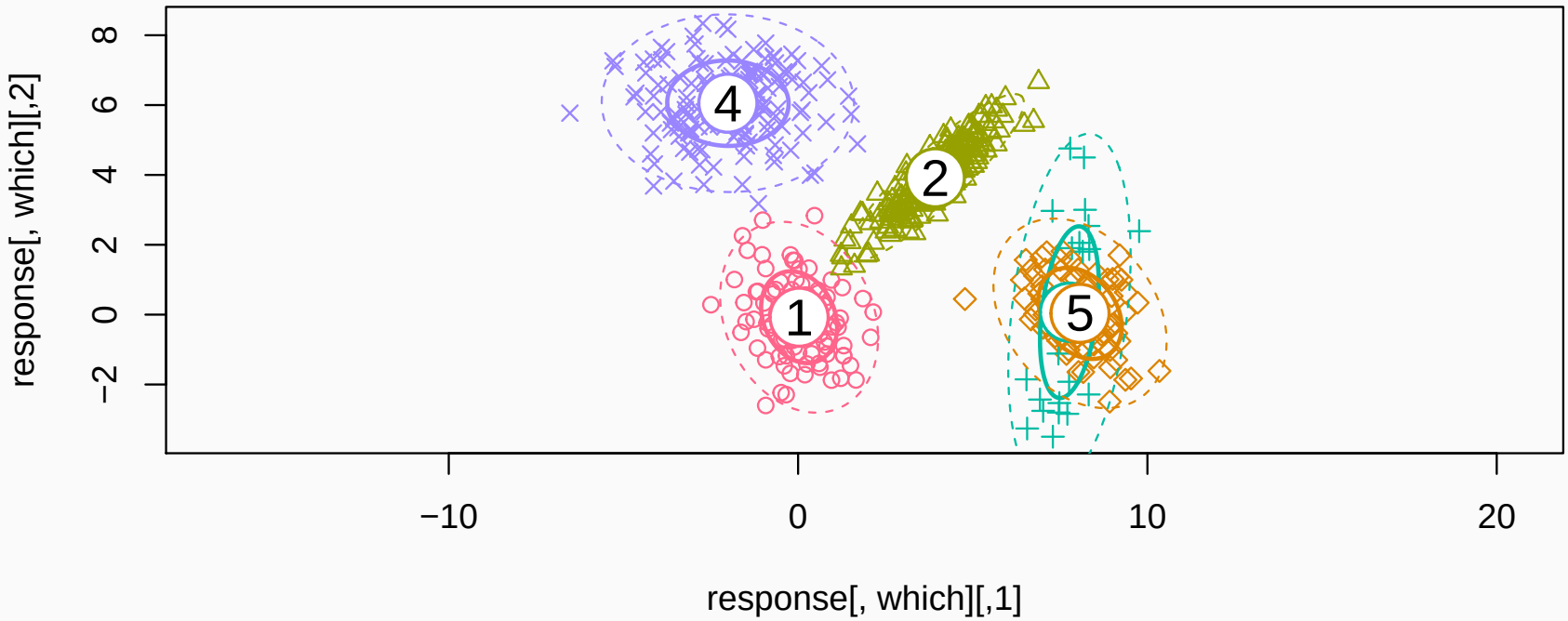
What if We Chose the Wrong Model?

- Correct number of components but diagonal covariance matrix



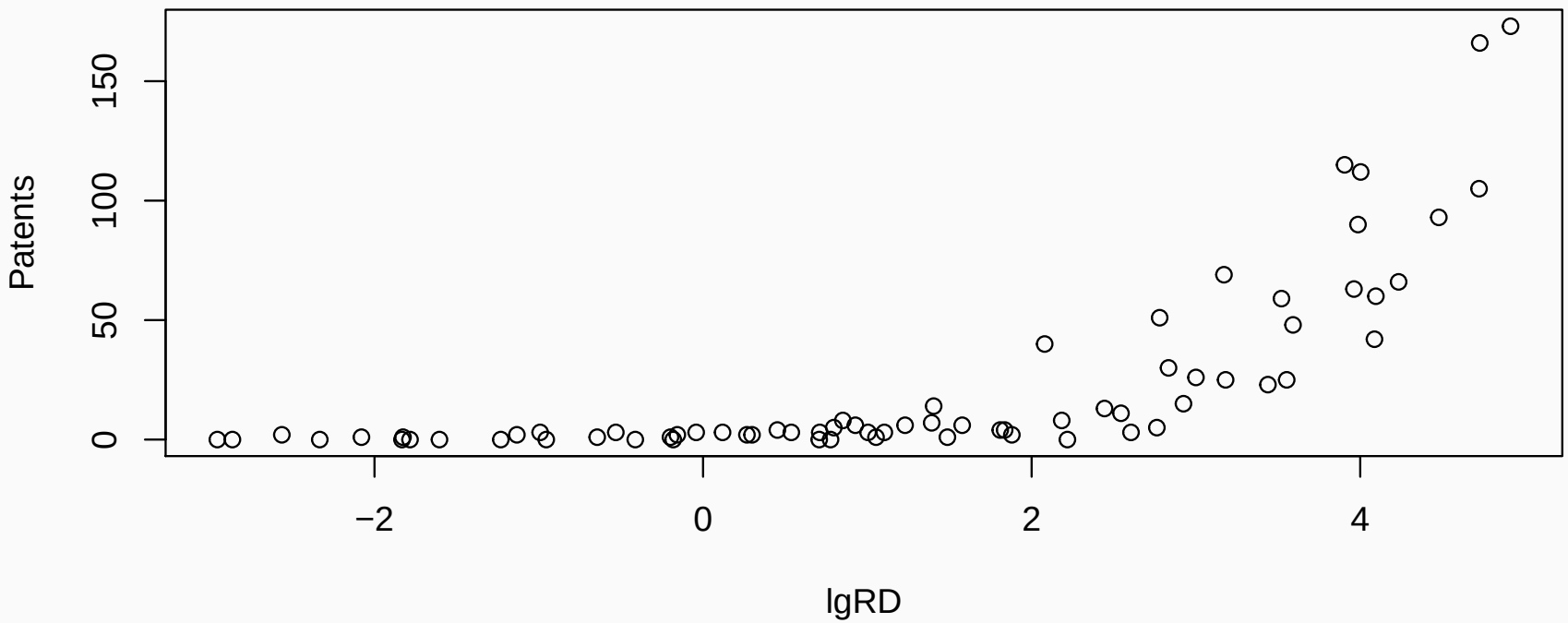
What if We Chose the Wrong Model?

- Incorrect number of components but correct model structure



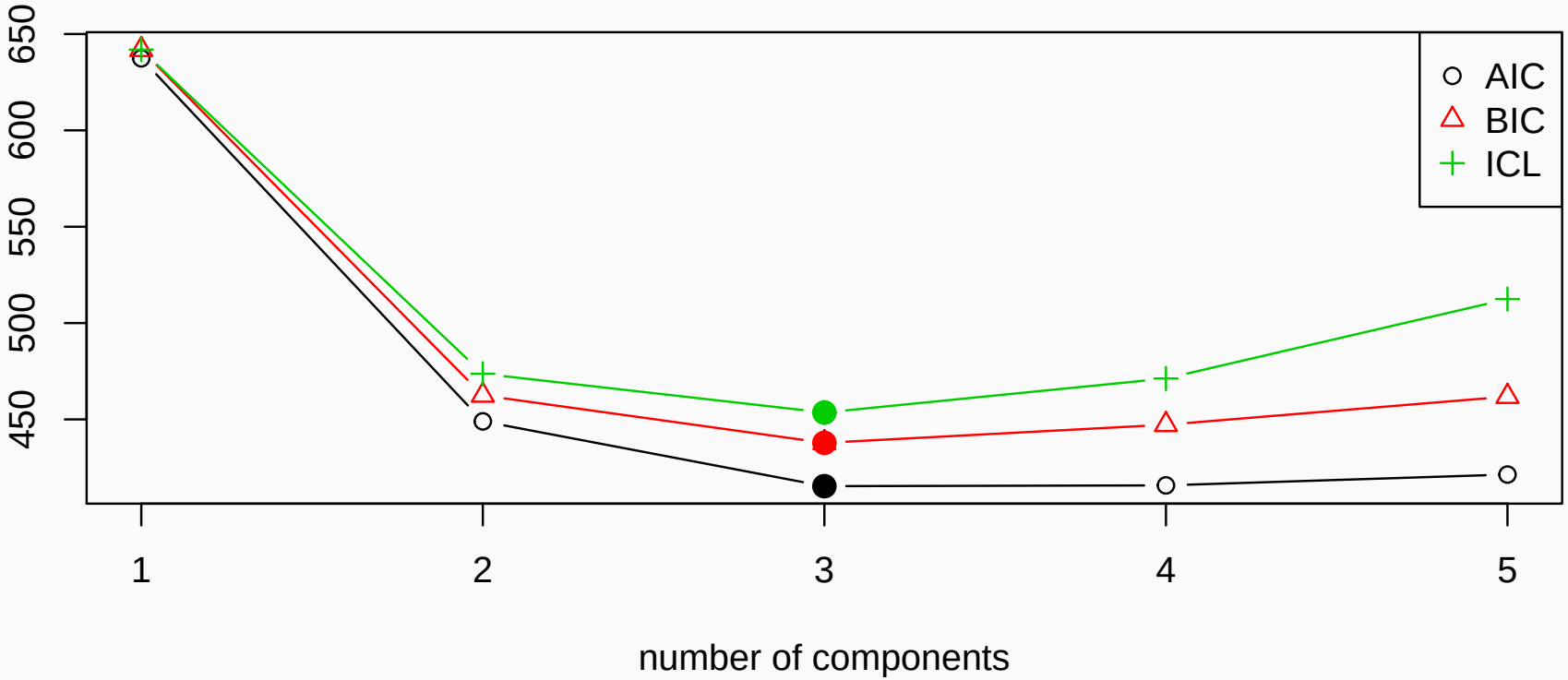
Another Example: Poisson Regression

- Number of approved patents as a function of money spent on research and development



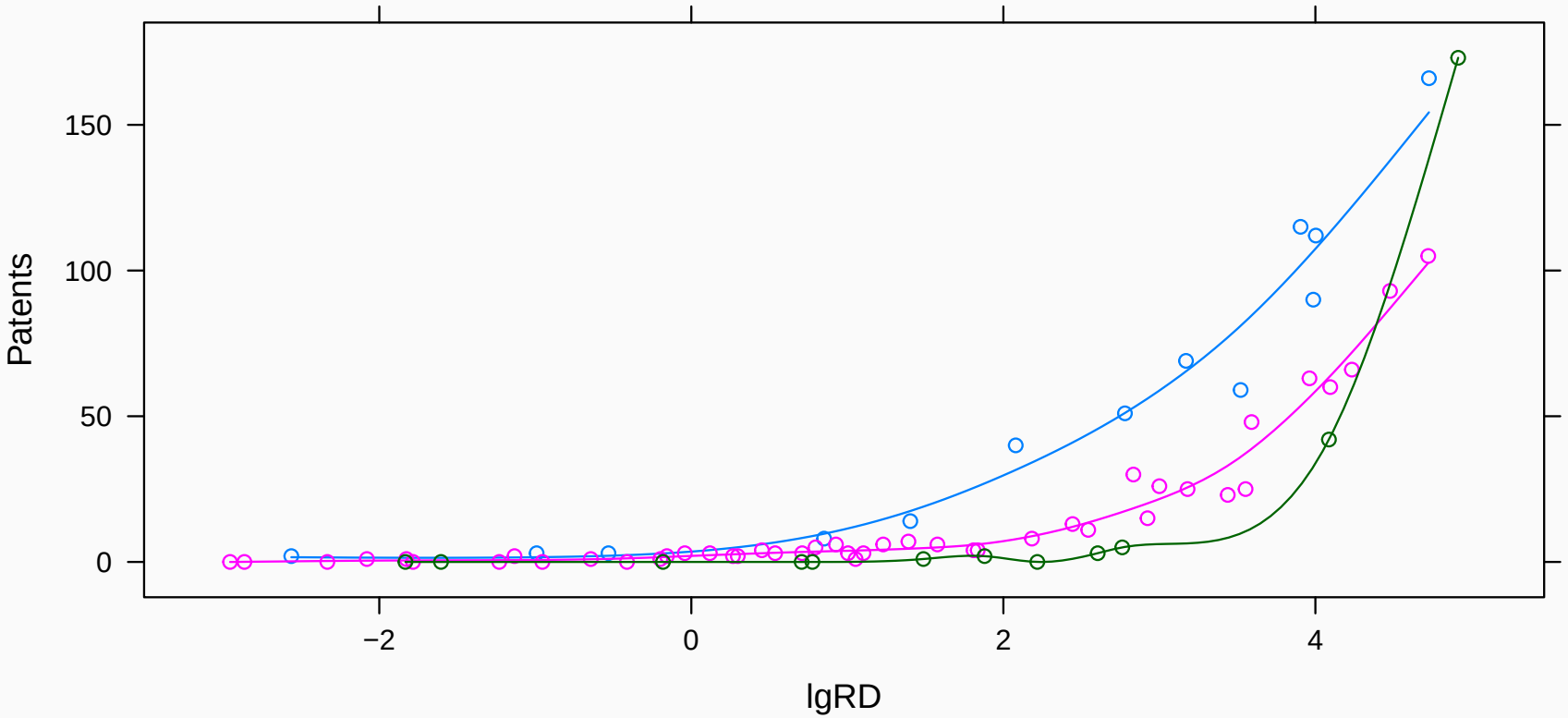
What is the true number of component subpopulations?

```
pat_mix = stepFlexmix(Patents ~ lgRD, k = 1:5,
  data = df_patent,
  model = FLXMRglm(family = "poisson"),
  concomitant = FLXPmultinom(~RDS))
```



Interpreting the Model

```
xyplot(Patents ~ lgRD, groups = factor(clusters(model)),
       df, type=c('p', 'spline'))
```



Interpreting the Model

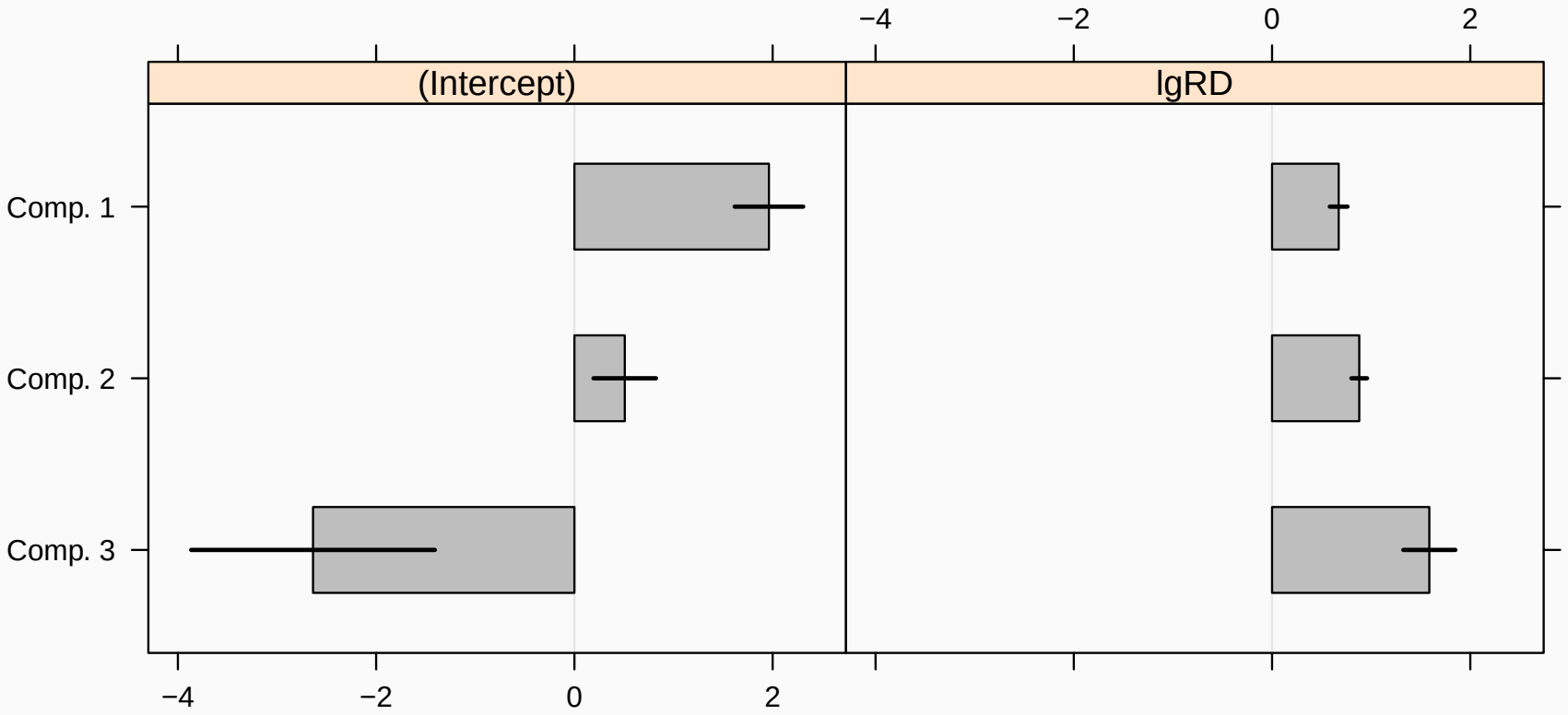
```
> summary(model)

       prior size post>0 ratio
Comp.1 0.184   13     47 0.277
Comp.2 0.615   45     63 0.714
Comp.3 0.201   12     48 0.250

'log Lik.' -197.6752 (df=10)
AIC: 415.3504   BIC: 437.8354
```

Interpreting the Model

```
plot(refit(model), bycluster=F)
```



Interpreting the Model

```
> summary(refit(model))
$Comp.1
             Estimate Std. Error z value  Pr(>|z|)
(Intercept) 1.962183   0.176430  11.122 < 2.2e-16 ***
lgRD        0.671868   0.045622  14.727 < 2.2e-16 ***
---
$Comp.2
(Intercept) 0.508128   0.160890  3.1582  0.001587 **
lgRD        0.879663   0.040248 21.8560 < 2.2e-16 ***
---
$Comp.3
(Intercept) -2.63689    0.62706 -4.2052 2.609e-05 ***
lgRD         1.58653    0.13400 11.8402 < 2.2e-16 ***
```

Final Thoughts

- Mixture model components can take many forms, ranging from simple Gaussians to more complex regressions
- Choosing model parameters wisely is very important
- They are a fantastic tool for choosing from among competing models/theories as long as their limitations are understood