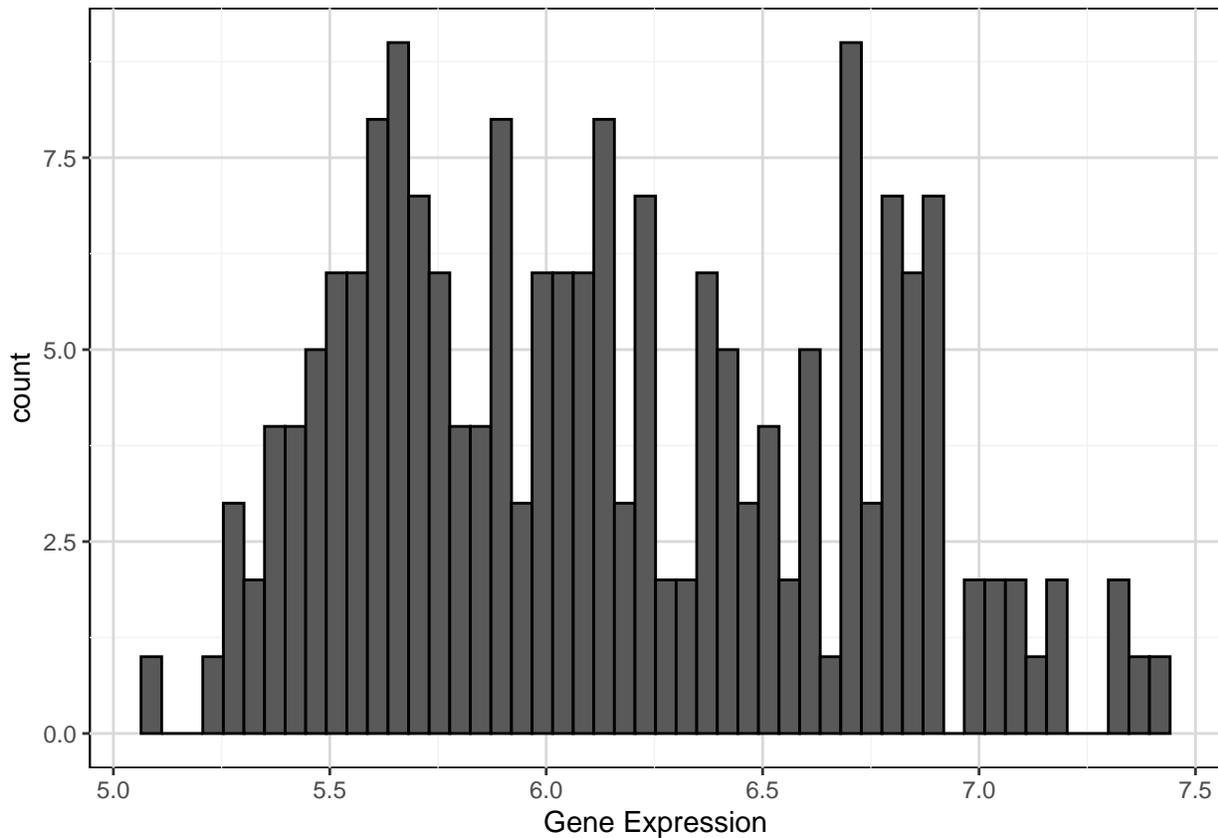


Mixture models

Sometimes having a simple named distribution to model your data is inadequate. This can occur when the data items are comprised of multiple subsets which behave differently.



Question 1

Example. Suppose we are looking at gene expression in individuals, some of which have had lung cancer and some of which do not have lung cancer. Some genes may be expressed differently between these two subsets so we can model gene expression with a *mixture model*.

In the tutorial we fit a Gaussian mixture model with two components to the above data. Reproduce the results from the tutorial using the code provided (i.e., combine the code provided in the tutorial into a single R script, and make sure you can run the script and get the same Figures).

Model specification

In a mixture model we assume that there are $K \geq 2$ components. The distribution of individuals across these components are given by their *mixing proportions* $\alpha_1, \dots, \alpha_K$, such that $\sum_k \alpha_k = 1$. An individual belonging

to the k -th sub-population behaves according to a distribution with density function f_k . These *component distributions* are parameterized with parameters $\vec{\theta}_k$.

So a mixture model is specified by its:

- mixing proportions: $\alpha_1, \dots, \alpha_K$
- component distributions: f_1, \dots, f_K
- component parameters: $\vec{\theta}_1, \dots, \vec{\theta}_K$

The distribution used to describe our data is then written as

$$y_i \sim \sum_{k=1}^K \alpha_k f_k(\vec{\theta}_k)$$

Question 2

Example. We can assume that those with lung cancer occur with probability α_1 and those without lung cancer occur with probability α_2 . The gene expression for these two components we can model with a Gaussian distribution $\mathcal{N}(\mu_k, \sigma)$ for $k = 1, 2$.

In this question we'll examine how the learned clusters stratify lung cancer condition. Recall from the tutorial that Stan does not estimate cluster assignments (see slide 16). However, given an estimate of $\vec{\mu}$ and σ , we can calculate the posterior cluster probabilities for each sample. If we define z_i to be the cluster assignment for observation i , then the posterior cluster probabilities are given by:

$$\pi(z_i = k | y_i, \mu_k, \sigma) \propto \mathcal{N}(y_i | \mu_k, \sigma)$$

Find the maximum a posteriori (MAP) component assignment for each sample. How many lung cancer patients are in cluster 1? How many lung cancer patients are in cluster 2?

Bonus. Are the MAP cluster assignments different than what we may observe if we randomly assigned lung cancer patients to the two clusters? Fisher's exact test offers a simple test to answer this question; use it to determine if the MAP cluster assignments are statistically significant.

Bayesian model

In a Bayesian data analysis we quantify our uncertainty about model parameters through probability distributions. We do so by placing a prior distribution on our parameters.

Example. In our example we can place the following prior distributions on our parameters:

- $\vec{\alpha} \sim \text{Dirichlet}(1, K)$ - Uniform/uninformative prior over a simplex
- $\mu_k \sim \mathcal{N}(10, 10)$ - Normal distribution with mean 10 and standard deviation 10
- $\sigma^2 \sim \text{IG}(1, 1)$ - Inverse gamma distribution with shape 1 and scale 1

Note: The parameters of our prior distributions are called our hyperparameters

The Stan code below encodes the Gaussian mixture model that we've specified, but for a model with K components, where K is provided as a part of the data.

```
// The input data required to fit the model
data {
  int<lower=1> n; // number of observations
  int<lower=2> K;
  vector[n] y; // observations
}
```

```

// The parameters of our model
parameters {
  simplex[K] alpha;    // mixing proportions
  ordered[K] mu;      // means of components
  real<lower=0> sigma; // standard deviations of components
}

// The posterior specification
model {
  // priors
  target += inv_gamma_lpdf(sigma^2 | 1, 1);
  target += normal_lpdf(mu | 10, 10);
  target += dirichlet_lpdf(alpha | rep_vector(1, K));

  // likelihood
  for(i in 1:n){
    vector[K] temp = log(alpha);

    for(k in 1:K){
      temp[k] += normal_lpdf(y[i] | mu[k], sigma);
    }

    target += log_sum_exp(temp);
  }
}

```

Question 3

In the tutorial we assumed that there were $K = 2$ clusters. However, in many cases we don't know how many clusters there are in the population. Modify your code to fit the gene expression data to Gaussian mixtures with different numbers of components (i.e., rerun the code from the slides with different settings of K , recording and reporting the posterior mean μ_k and σ_k for each setting of K , for $k = 1, \dots, K$).

Bonus. In these instances where we have multiple models to fit a single dataset we often choose a “best” model. Formal Bayesian model comparison techniques are too computationally expensive to cover here; in lieu of these, use the posterior credible intervals about the model parameters to draw a conclusion about a suitable value for K .

Hierarchical model

In a hierarchical model, we add additional layer(s) of prior distributions. This can reflect general similarities in the population

In our example, we may want to assume that individuals with and without lung cancer have a mean expression level that itself is a realization of an overall distribution over mean expression levels.

$$\begin{aligned}y_i | \vec{\mu}, \vec{\sigma}, \vec{\alpha} &\sim \text{GMM}(\vec{\mu}, \vec{\sigma}, \vec{\alpha}) \\ \vec{\sigma} &\sim \text{IG}(3, 10) \\ \vec{\alpha} &\sim \text{Dirichlet}(1, 2) \\ \vec{\mu} | \mu_0, \sigma_0 &\sim \mathcal{N}(\mu_0, \sigma_0) \\ \mu_0 &\sim \mathcal{N}(10, 10) \\ \sigma_0 &\sim \text{IG}(3, 10)\end{aligned}$$

Question 4

Modify the Stan file for a Gaussian mixture model so that there are hierarchical priors on the μ_k 's. Reference Stan's manual for distribution functions for the [normal](#), [inverse gamma](#), and [dirichlet](#) distributions.

Bonus. Often in Bayesian analyses we want to evaluate the sensitivity of our results to our hyperparameters, however, editing the Stan code for every new set of hyperparameters can be tedious. How could you modify the code to take hyperparameters as input to avoid constant re-writing of the Stan file?