MIXTURE MODELS IN R

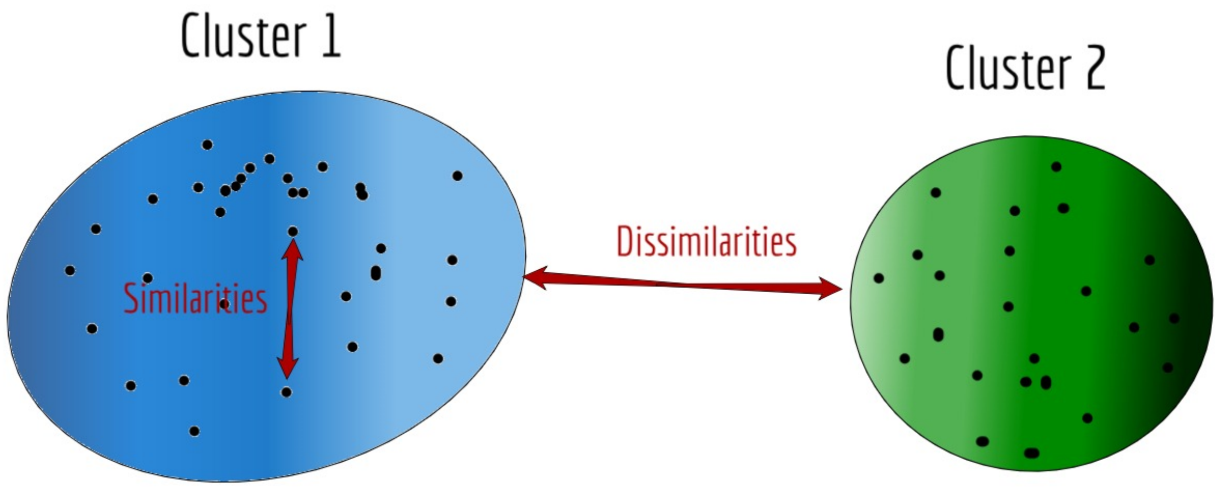
Introduction to model-based clustering

Victor Medina
Researcher at SBIF

What is clustering?

The procedure of partitioning a set of observations into a set of meaningful subclasses

→ Help to explore and understand the natural structure in a dataset



Applications of clustering

- Medicine
  - Ex. In medical imaging to distinguish between different types of tissue
- Business
  - Ex. To discover distinctive groups of customers to develop targeted marketing programs
- Social Sciences
  - Ex. To identify zones in a city by the type of committed crimes to manage law enforcement resources more effectively

Clustering methods

- Partitioning techniques
  - Find centers of clusters among the observations and each one is assigned to the cluster that has the closest center. Ex. **Kmeans**
- Hierarchical techniques
  - Connect the observations based on their similarity to form clusters. Ex. **Hierarchical clustering**
- Model-base methods
  - Use probabilistic distributions to create the clusters. Ex. **Mixture models**

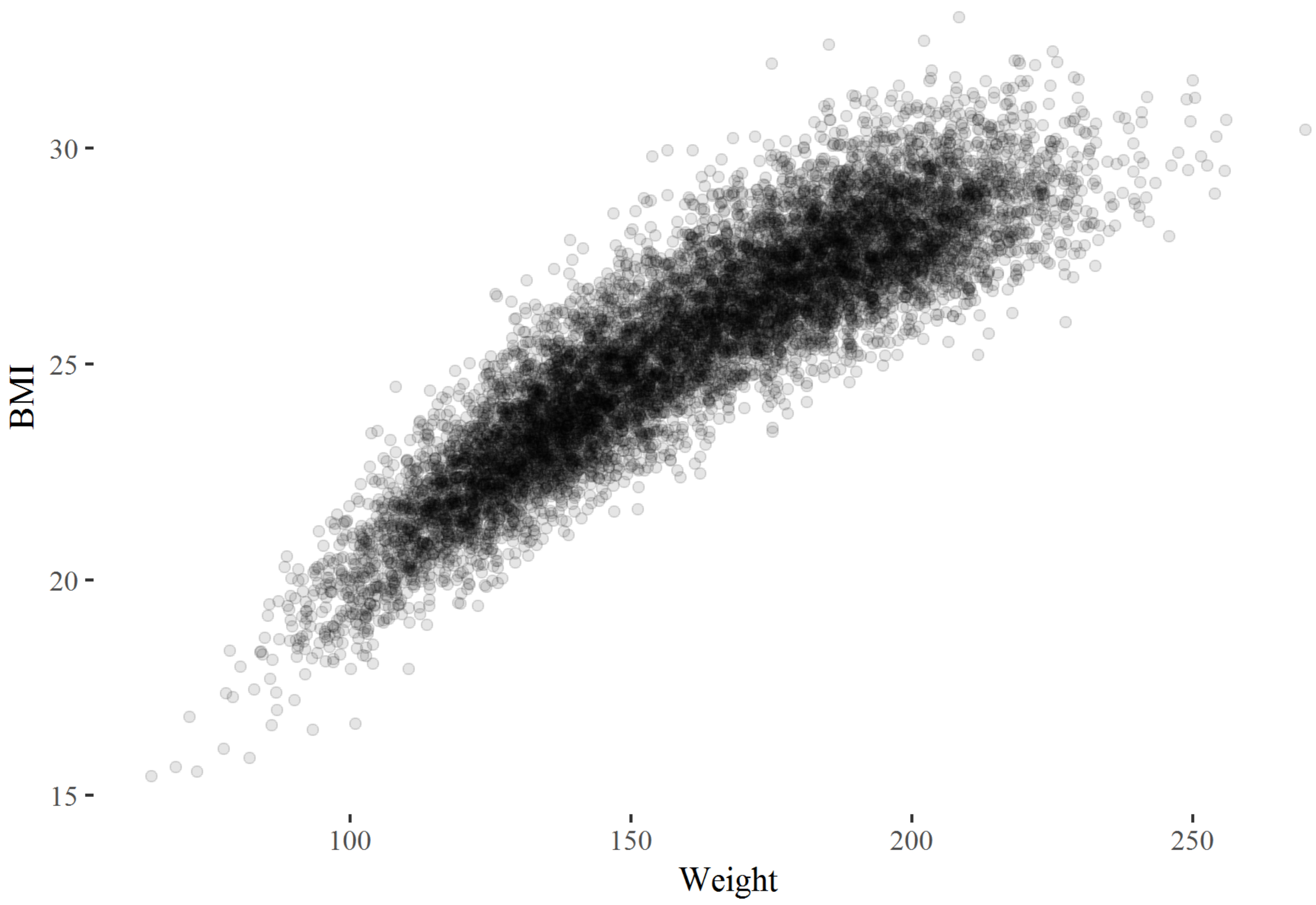
Gender dataset

```
gender <- read.csv("gender.csv")
head(gender)
```

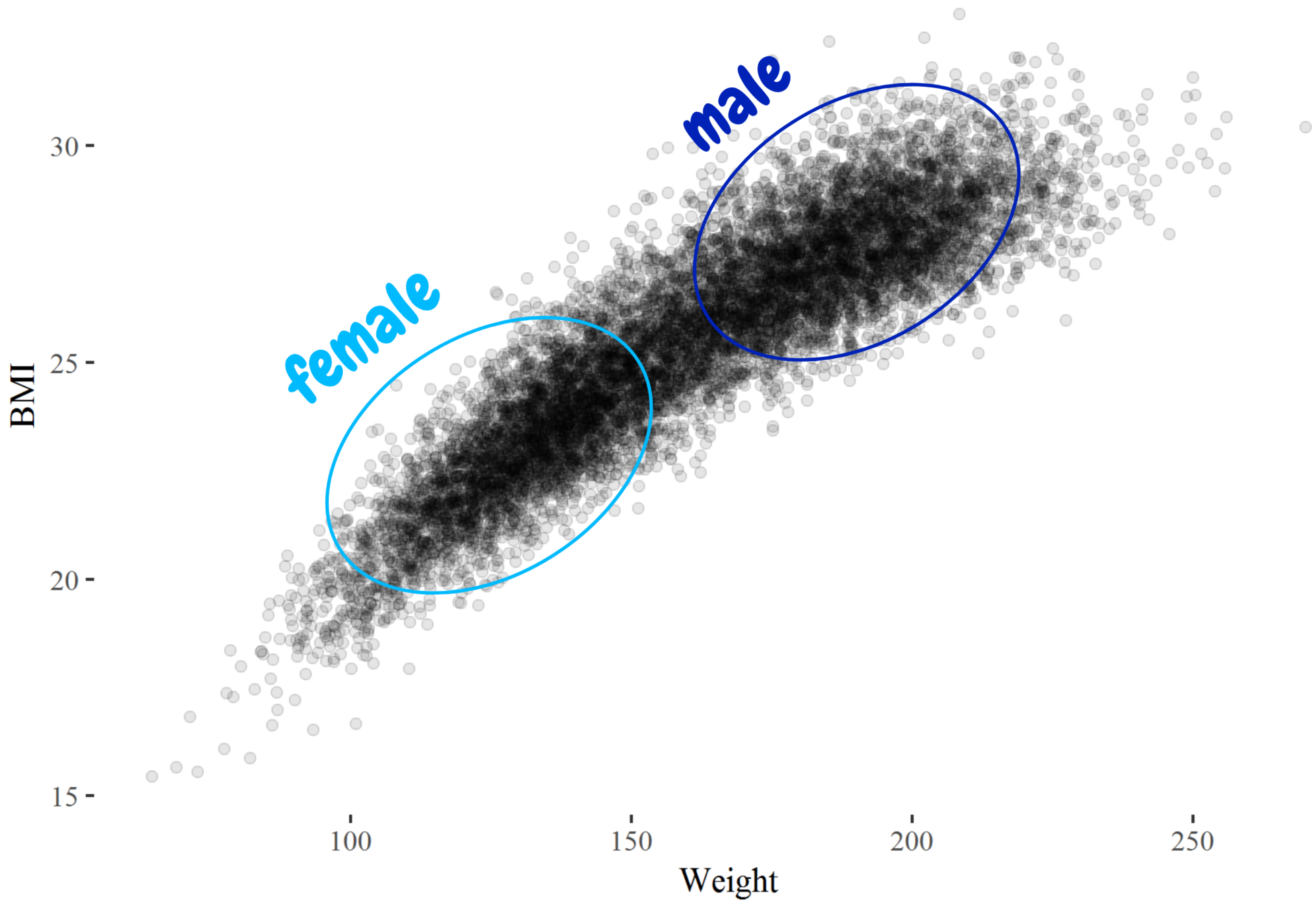
```
    Height   Weight      BMI
1 73.84702 241.8936 31.18576
2 68.78190 162.3105 24.12104
3 74.11011 212.7409 27.23291
4 71.73098 220.0425 30.06706
5 69.88180 206.3498 29.70803
6 67.25302 152.2122 23.66049
```

Gender dataset: Can you guess the gender?

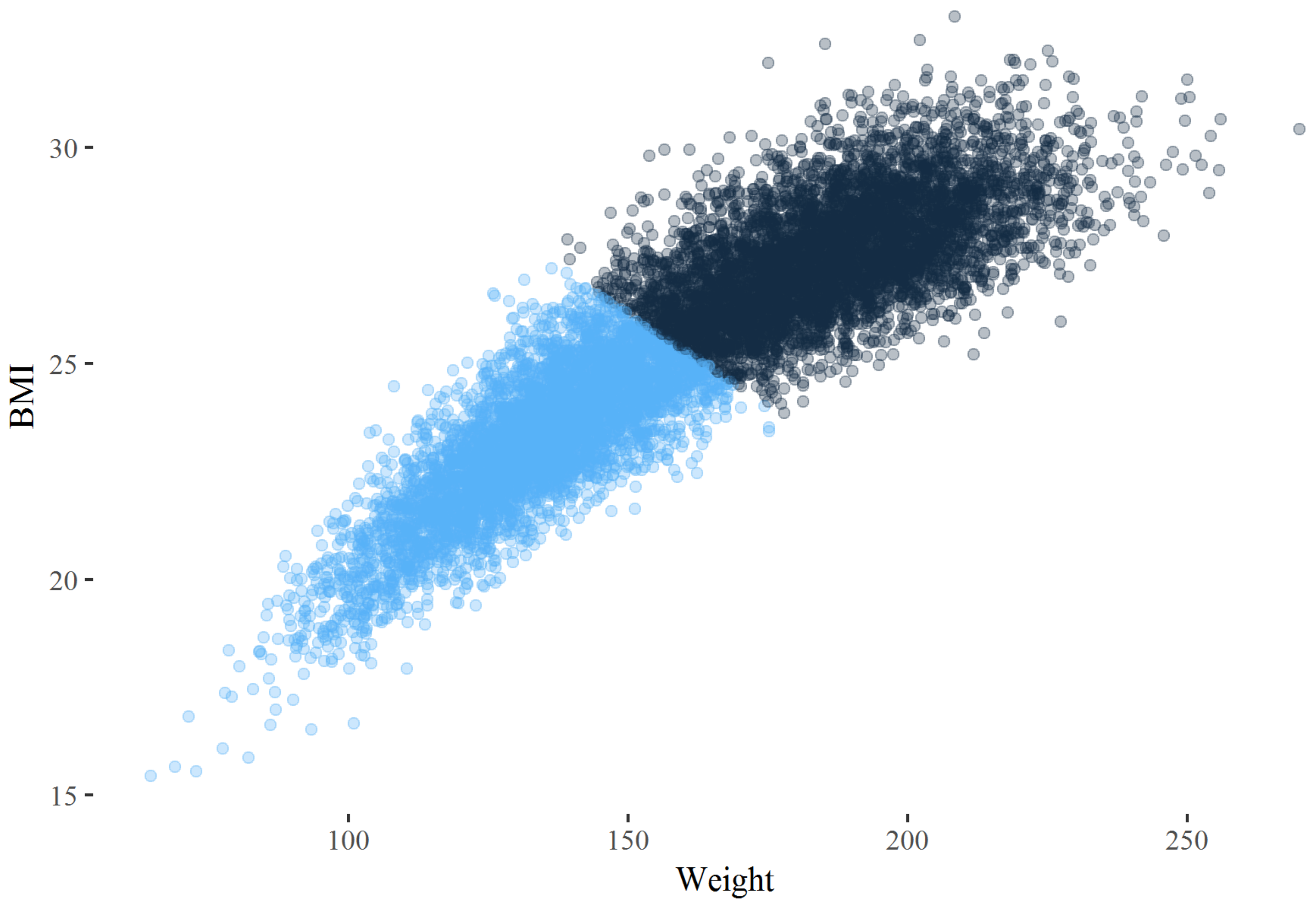
```
library(ggplot2)
ggplot(gender, aes(x = Weight, y = BMI)) + geom_points()
```



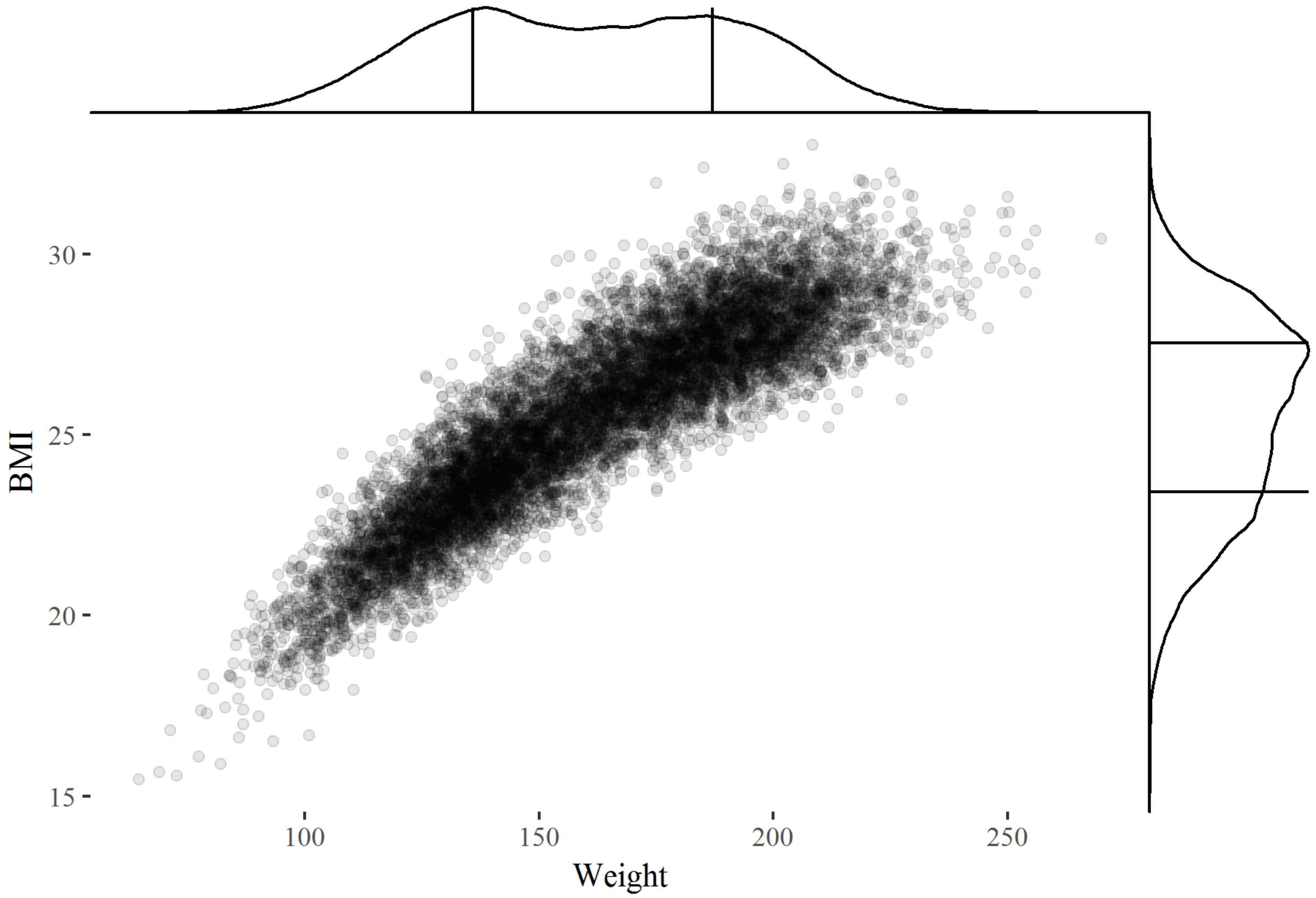
Gender dataset: Can you guess the gender?



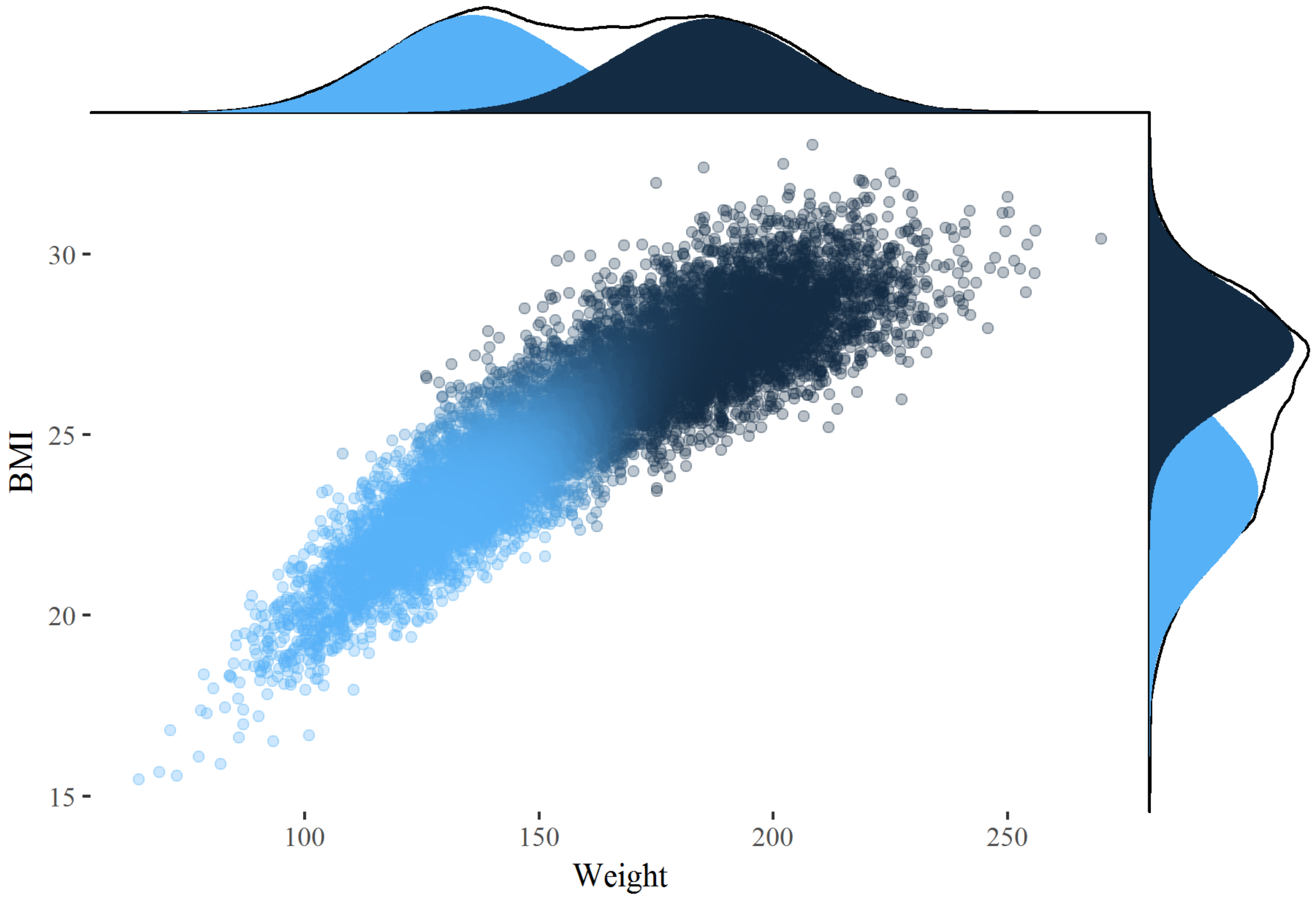
Under traditional cluster approaches



Model-based clustering



Model-based clustering



MIXTURE MODELS IN R

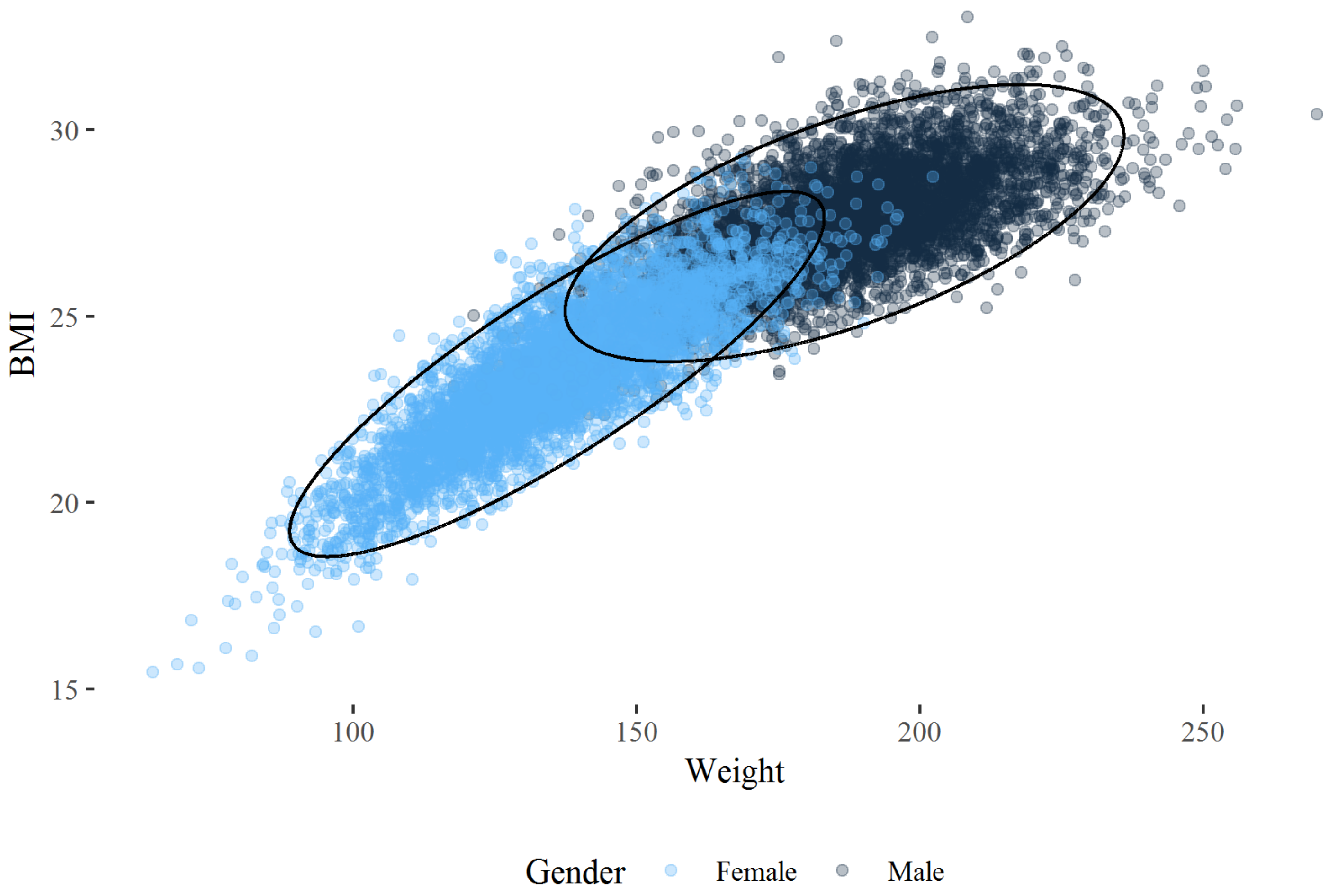
Let's practice!

MIXTURE MODELS IN R

Gaussian distribution

Victor Medina
Researcher at SBIF

Mixture model to Gender dataset

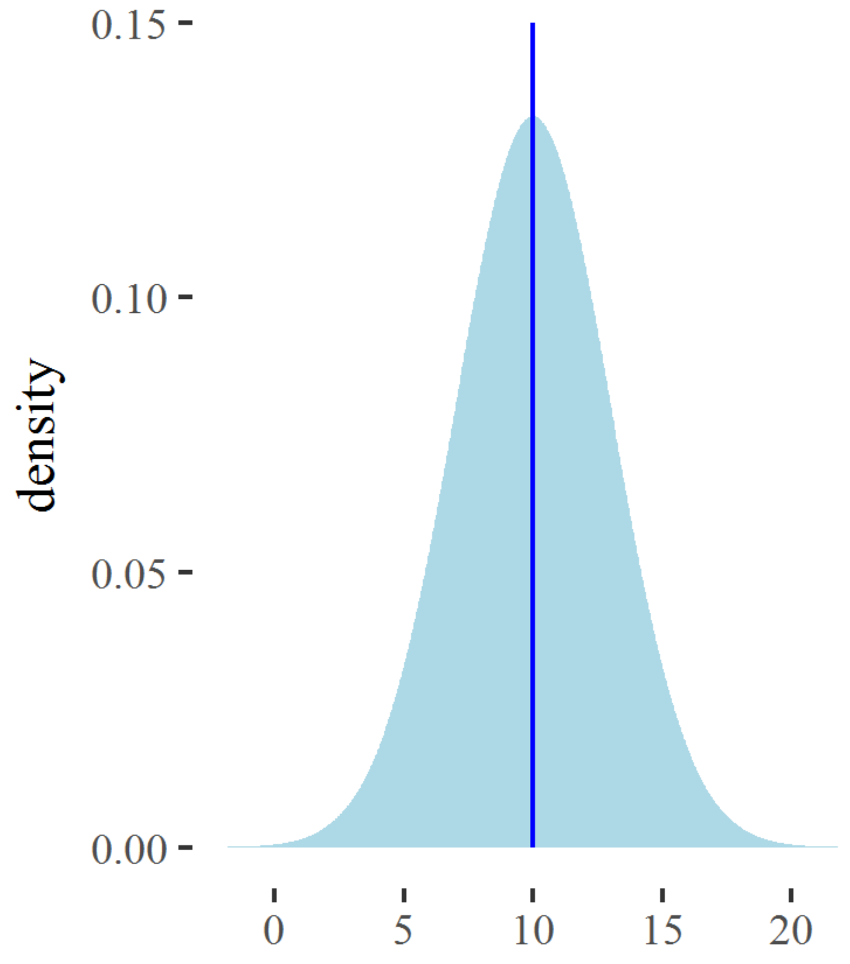


Packages for fitting Mixture Models

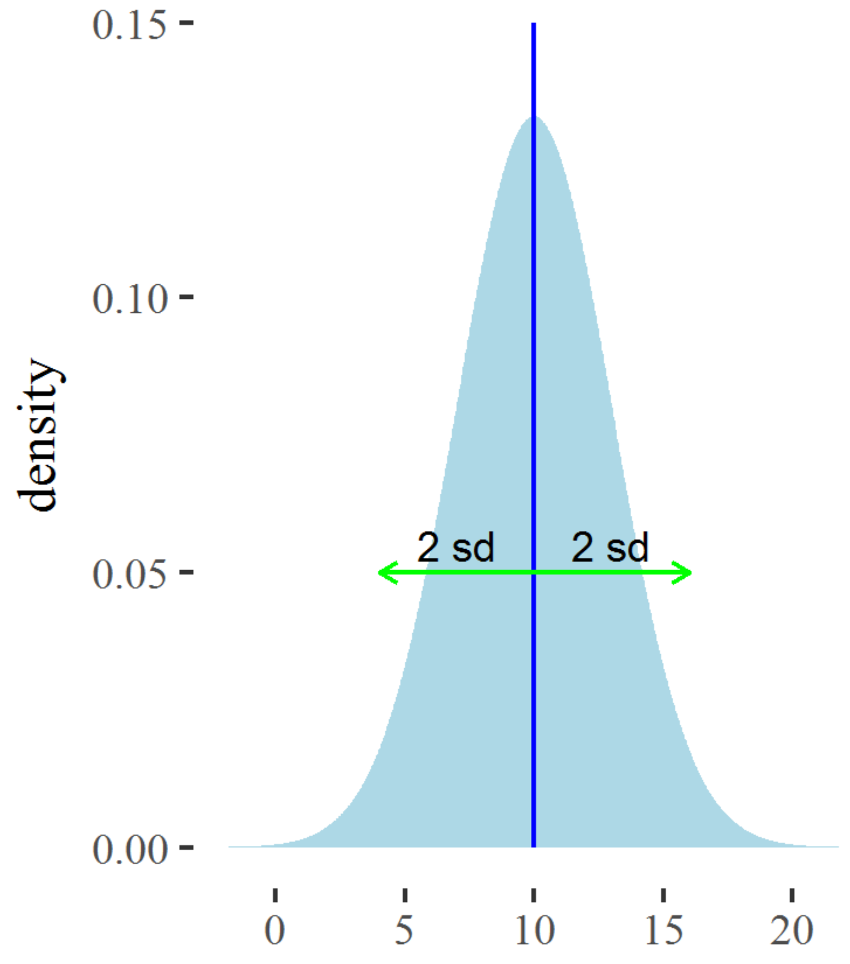
- `mixtools`
  - The Poisson distribution is not implemented.
- `bayesmix`
  - Bayesian inference is outside the scope of the course.
- `EMCluster`
  - Only Gaussian distributions.
- **`flexmix`**
  - Has all the distributions we need and gives you the flexibility to perform more complex models.

Properties of Gaussian distribution

Mean



Standard deviation



Sample from a Gaussian distribution

To generate samples from a Gaussian distribution:

- `rnorm(n, mean, sd)`

Example: Generate 100 values from a Gaussian distribution with a mean of 10 and a standard deviation of 5

```
> population_sample <- rnorm(n = 100, mean = 10, sd = 5)
> head(population_sample)
```

```
[1]  6.248874  9.564190 16.006521  9.139647 10.114969 16.423538
```

Estimation of the Mean

- Don't know the mean and the standard deviation, only know the observations
  - Need to be **estimated from the observations**
- To estimate the mean, we can calculate the **sample mean**

```
> mean_estimate <- mean(population_sample)
```

```
[1] 10.35759
```

Estimation of the Standard Deviation (sd)

- To estimate the `sd`, we perform the following procedure

$$value_i \rightarrow (. - mean\_estimate) \rightarrow (.)^2 \rightarrow mean(.) \rightarrow \sqrt{(.)}$$

```
> population_sample %>%
+   subtract(mean_estimate) %>%
+   raise_to_power(2) %>%
+   mean() %>%
+   sqrt()
[1] 5.318641
```

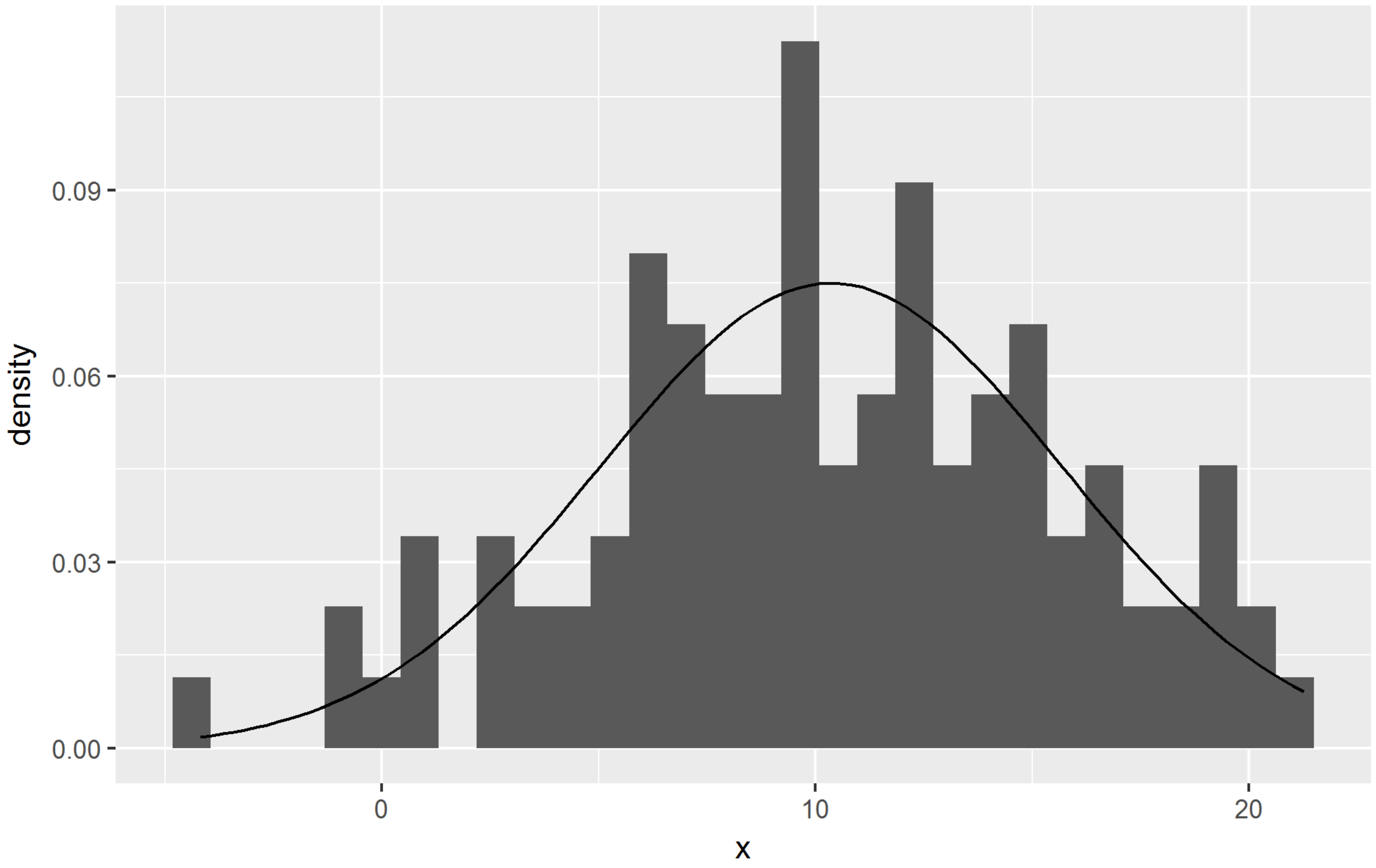
- Using the `sd` function

```
> standard_deviation_estimate <- sd(population_sample)
> standard_deviation_estimate
```

```
[1] 5.345435
```

Visualizing the sample with estimated Gaussian distribution

```
> # Transform the sample into a data frame
> population_sample <- data.frame(x = population_sample)
> # Plot the histogram
> ggplot(data = population_sample) +
+   geom_histogram(aes(x = x, y = ..density..)) +
+   stat_function(geom = "line",
+                 fun = dnorm,
+                 args = list(mean = mean_estimate,
+                             sd = standard_deviation_estimate))
```


density
0.09
0.06
0.03
0.00
0
10
20
x

MIXTURE MODELS IN R

Let's practice!

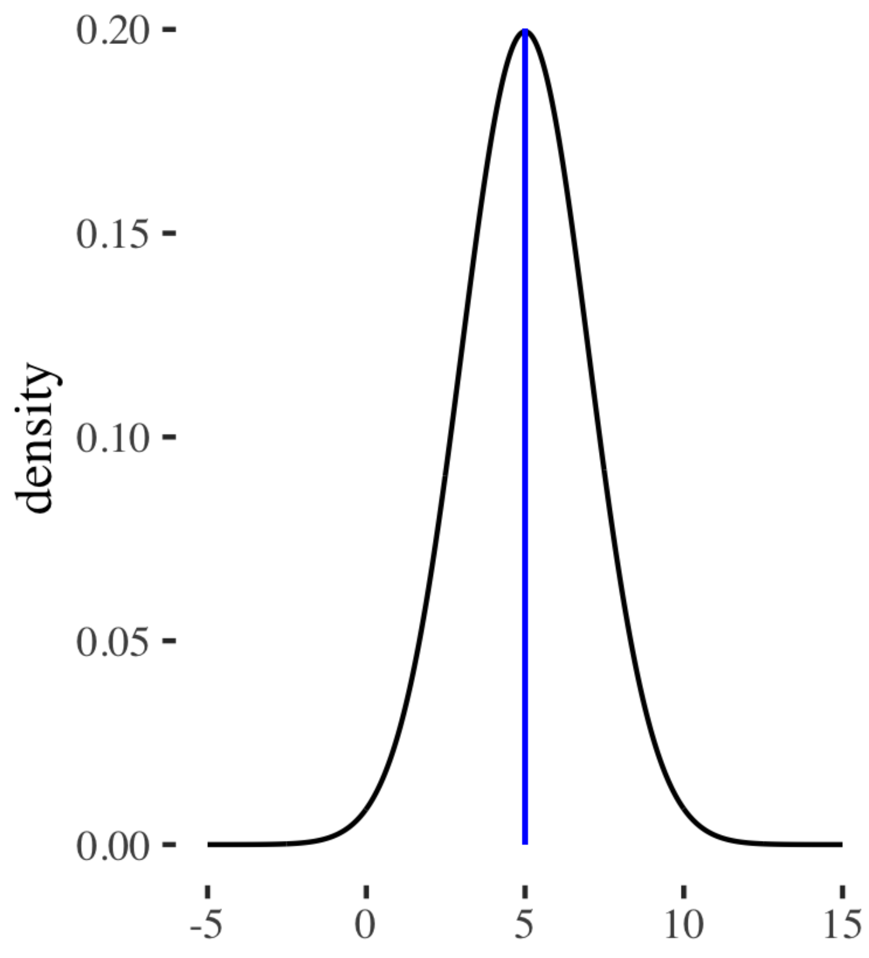
MIXTURE MODELS IN R

Gaussian mixture models (GMM)

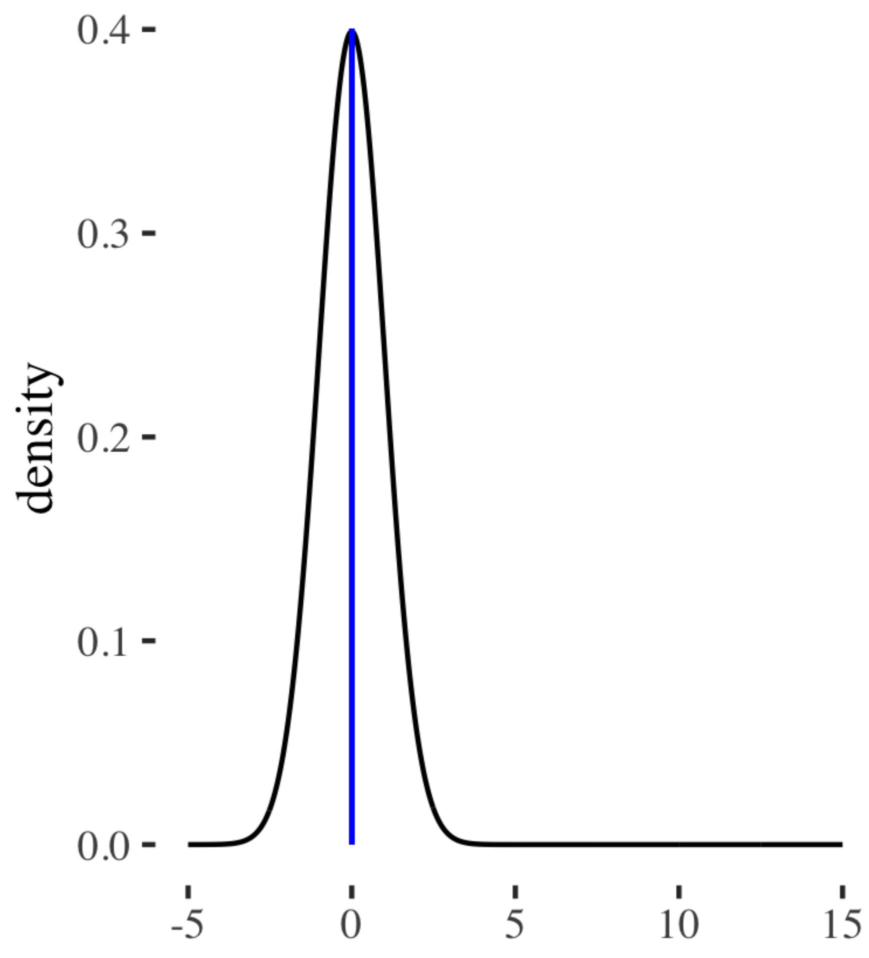
Victor Medina
Researcher at SBIF

Flipping and sampling

- Heads



- Tails



Flipping the coin

```
> # The number of observations
> number_of_obs <- 500
> # Simulate the coin
> coin <- sample(c(0,1), size = number_of_obs,
+                replace = TRUE, prob = c(0.5, 0.5))
> head(coin)
```

```
[1] 0 1 0 1 0 0
```

```
> table(coin)
```

```
coin
  0   1
239 261
```

Sampling and simulate the mixture

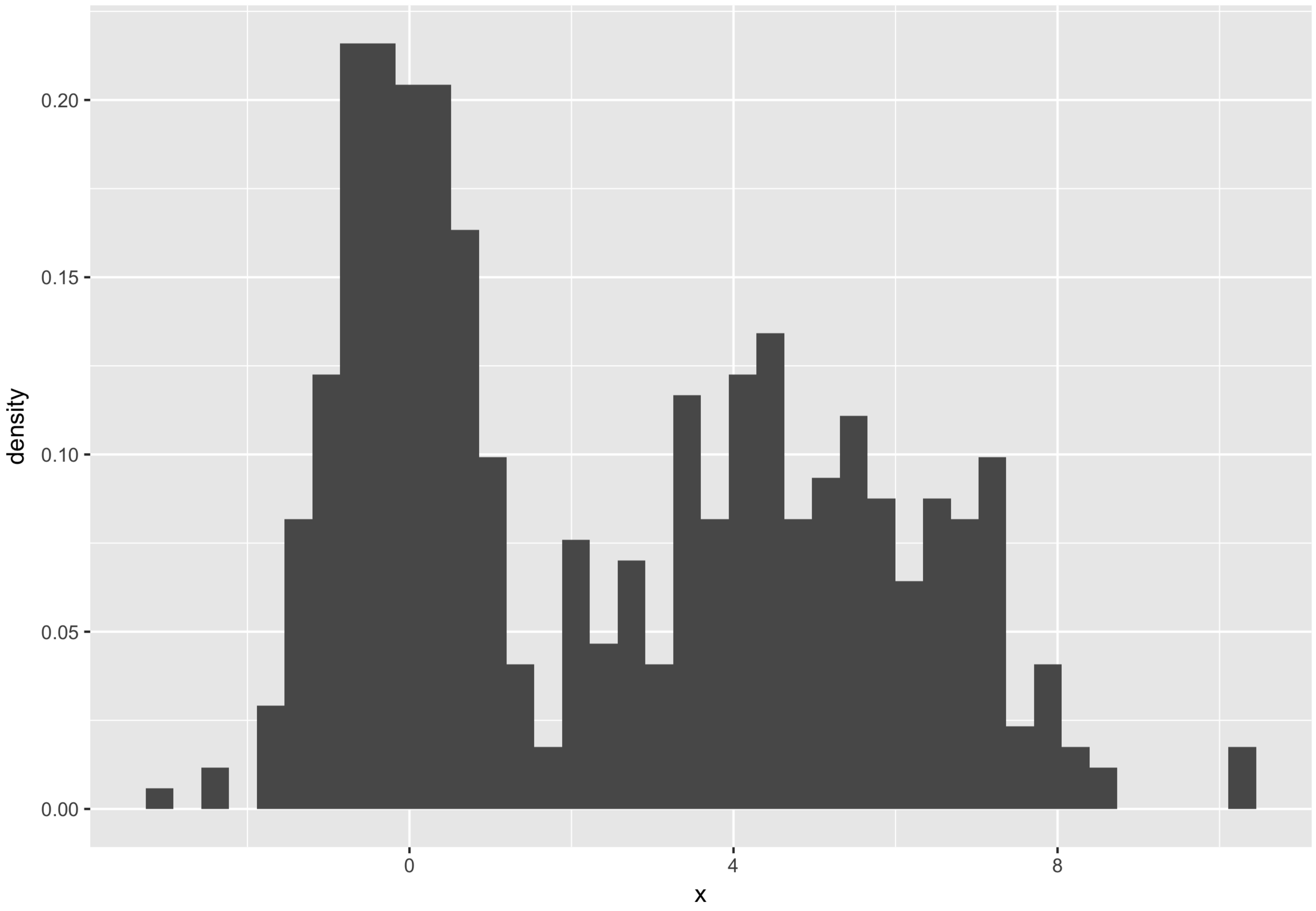
```
> # Gaussian 1 "heads"
> gauss_1 <- rnorm(n = number_of_obs, mean = 5, sd = 2)
> # Gaussian 2 "tails"
> gauss_2 <- rnorm(n = number_of_obs)

> # Simulate the mixture
> mixture_simulation <- ifelse(coin, gauss_1, gauss_2)
> head(cbind(coin, gauss_1, gauss_2, mixture_simulation))
```

```
     coin  gauss_1    gauss_2 mixture_simulation
[1,]    0 7.378712 -0.4559596         -0.4559596
[2,]    1 6.102770  3.3595880          6.1027696
[3,]    0 5.707269 -0.0731496         -0.0731496
[4,]    1 3.592059 -1.2407104          3.5920586
[5,]    0 5.236851 -0.5110058         -0.5110058
[6,]    0 4.152619 -0.5124031         -0.5124031
```

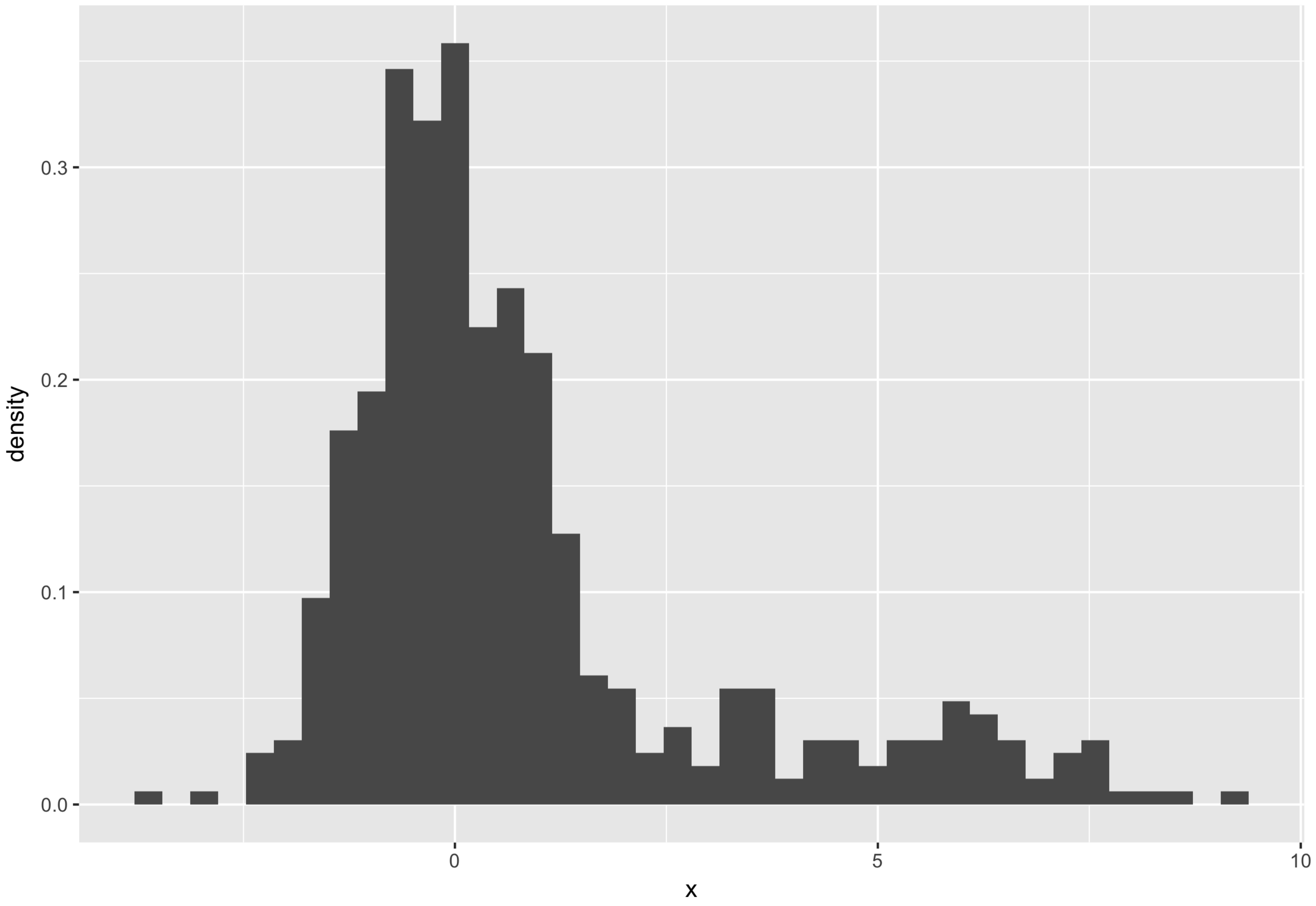
Plot the mixture

```
> # Transform to a data frame
> mixture_simulation <- data.frame(x = mixture_simulation)
> # Create the histogram
> ggplot(mixture_simulation) +
+     geom_histogram(aes(x = x, ..density..), bins = 40)
```


0.20
0.15
0.10
0.05
0.00
density
0
4
8
x

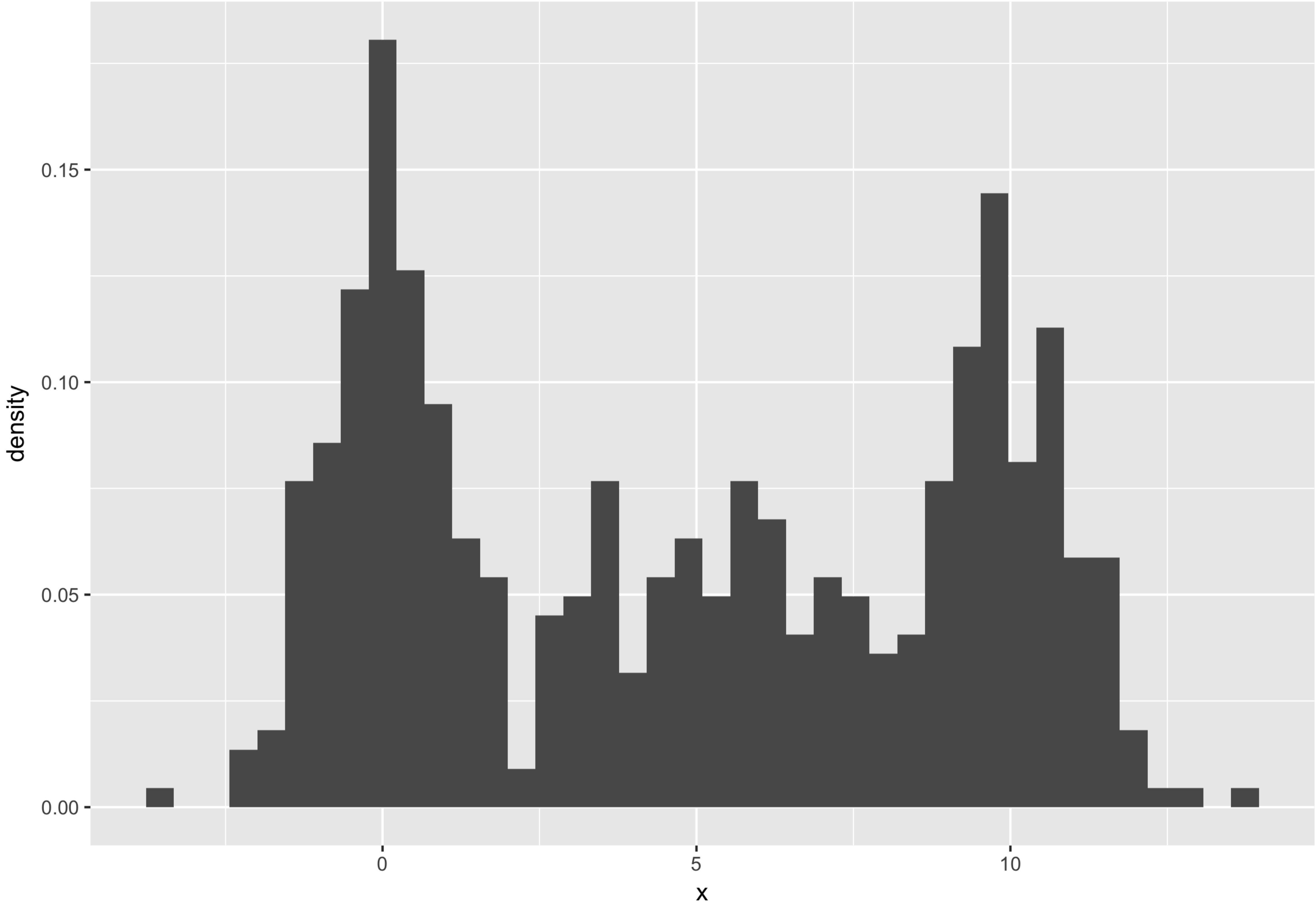
Changing the proportions

```
> # Simulate the coin with different proportions
> coin <- sample(c(0,1), size = number_of_obs,
+                replace = TRUE, prob = c(0.8, 0.2))
> # Simulate the mixture
> mixture_simulation <- data.frame(x = ifelse(coin, gauss_1, gauss_2))
> # Create the histogram
> ggplot(mixture_simulation) +
+   geom_histogram(aes(x = x, ..density..), bins = 40)
```


density
0.0
0.1
0.2
0.3
0
5
10
x

Mixture of three distributions

```
> proportions <- sample(c(0, 1, 2), number_of_obs,
+                       replace = TRUE, prob = c(1/3, 1/3, 1/3))
> gauss_3 <- rnorm(n = number_of_obs, mean = 10, sd = 1)
> mixture_simulation <- data.frame(x = ifelse(proportions == 0, gauss_1,
+                                     ifelse(proportions == 1, gauss_2, gauss_3)))
> ggplot(mixture_simulation) +
+   geom_histogram(aes(x = x, ..density..), bins = 40)
```


0.15
0.10
0.05
0.00
density
0
5
10
x

MIXTURE MODELS IN R

Let's practice!