MIXTURE MODELS IN R

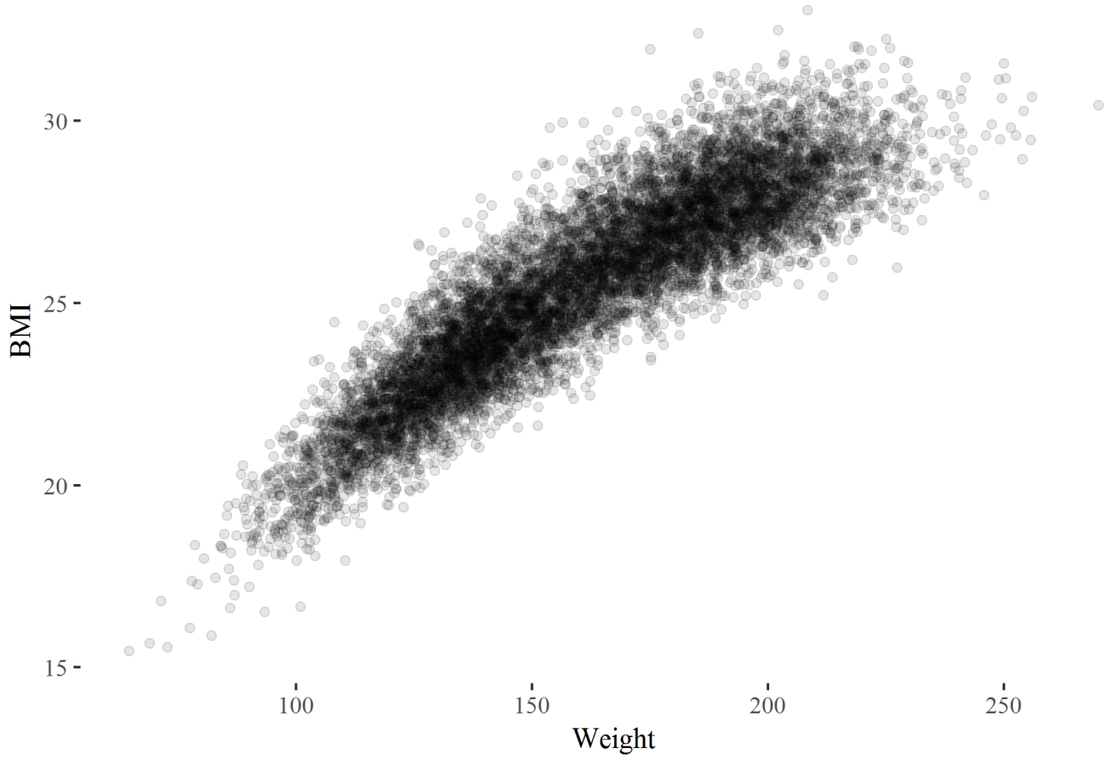
Structure of mixture models

Victor Medina
Researcher at SBIF

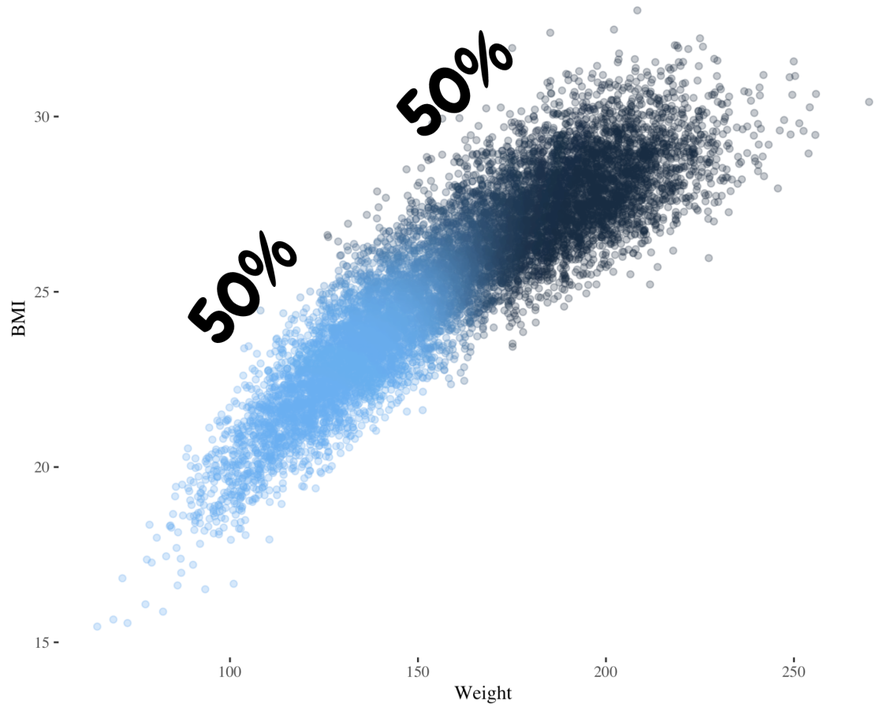
Description of mixture models

1. Which is the suitable probability distribution?
    - Get familiar with different probability distributions.
2. How many sub-populations should we consider?
    - Data scientist or statistical criteria.
3. What are the parameters and their estimations?
    - Awesome method called EM algorithm!

Example 1: Gender data set

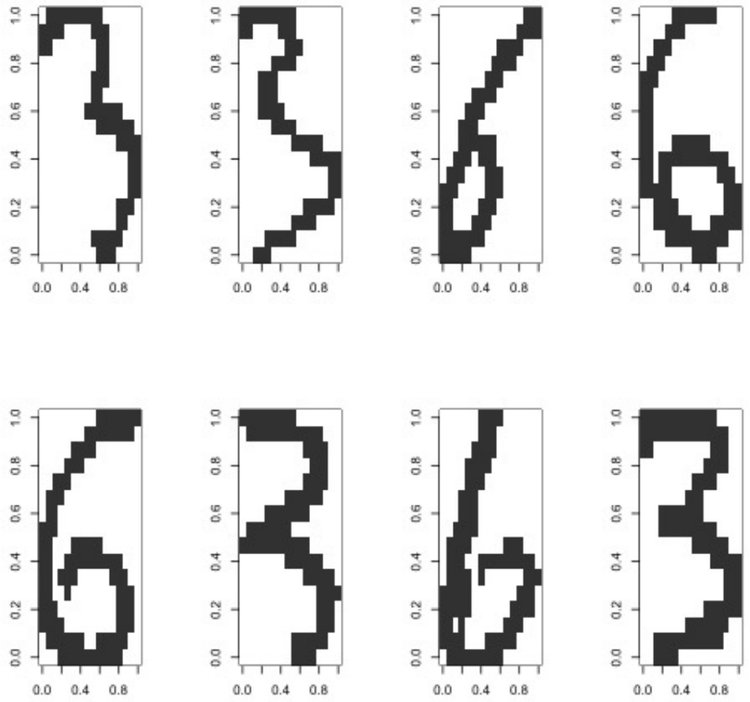


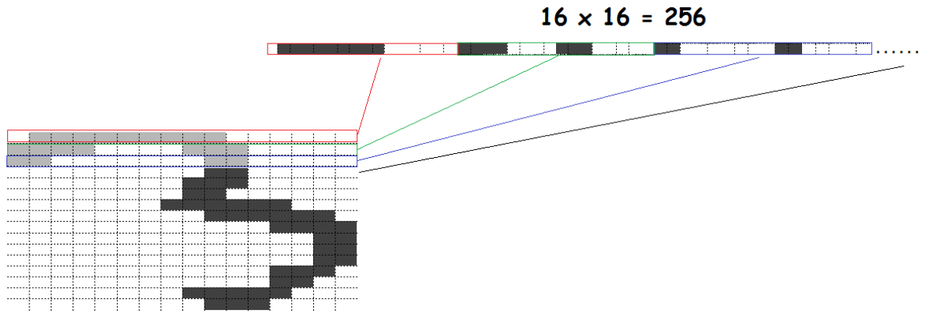
Example 1: Gender dataset results



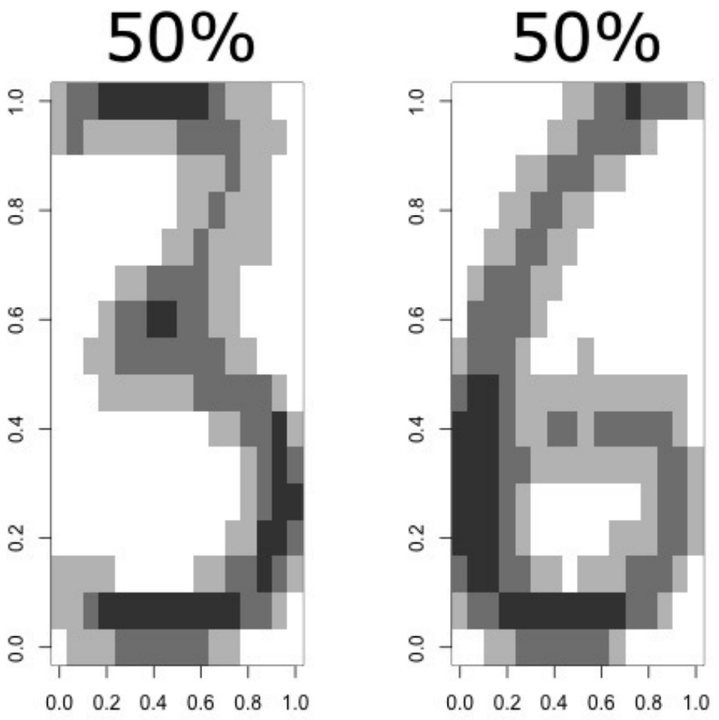
1. Which distribution? **Bivariate Gaussian distribution**
2. How many clusters? **Two clusters**
3. What are the estimates? **Means, Standard deviations and proportions**

Example 2: Handwritten digits





Example 2: Handwritten digits results

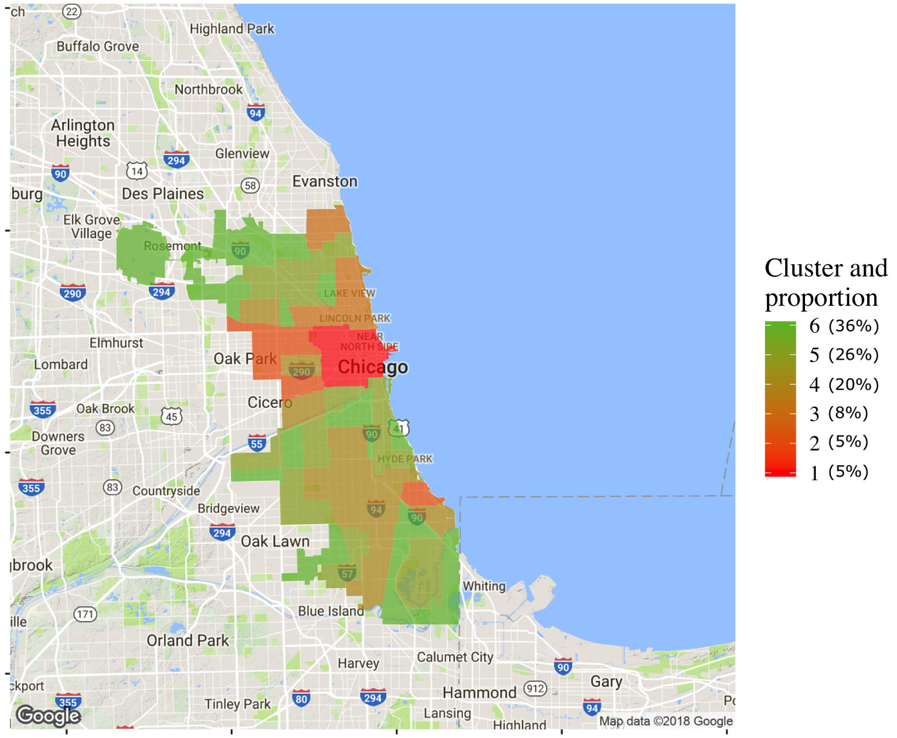


1. Which distribution? **Bernoulli distribution**
2. How many clusters? **Two clusters**
3. What are the estimates? **The mean probability of being 1 for every dot and proportions**

Example 3: Crime types

COMMUNITY	ASSAULT	BATTERY	BURGLARY	CRIMINAL DAMAGE	CRIMINAL TRESPASS	DECEPTIVE PRACTICE
ALBANY PARK	123	429	147	287	38	137
ARCHER HEIGHTS	51	134	92	114	23	67
ARMOUR SQUARE	74	184	55	99	56	59
ASHBURN	169	448	194	379	43	178
AUBURN GRESHAM	708	1681	339	859	228	310
AUSTIN	1198	3347	517	1666	265	767
AVALON PARK	118	280	76	150	29	73
AVONDALE	135	350	145	310	36	200
BELMONT CRAGIN	337	850	327	528	88	314
BEVERLY	63	97	64	129	29	89
BRIDGEPORT	105	217	133	205	27	96
BRIGHTON PARK	182	477	131	309	35	95
BURNSIDE	37	80	26	53	1	15
CALUMET HEIGHTS	108	230	86	197	34	84
CHATHAM	540	1282	320	681	219	366
CHICAGO LAWN	459	1227	420	742	126	288
CLEARING	54	154	68	168	14	69
DOUGLAS	205	540	72	220	123	195
DUNNING	104	296	119	210	37	140
EAST GARFIELD PARK	376	1144	129	430	191	148

Example 3: Crime types results



1. Which distribution? **Multivariate Poisson distribution**
2. How many clusters? **Six clusters**
3. What are the estimates? **Average number of crimes by type and proportions**

MIXTURE MODELS IN R

Let's practice!

MIXTURE MODELS IN R

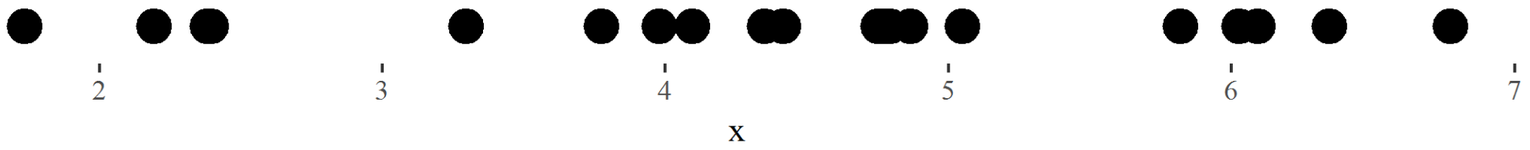
Parameters estimation

Victor Medina
Researcher at SBIF

The problem

```
> head(data)
```

```
          x
1 3.294453
2 5.818586
3 2.380493
4 4.415913
5 5.048659
6 4.750195
```



Assumptions

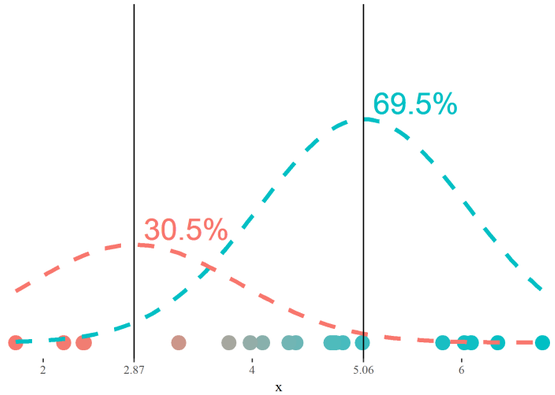
1. Which distribution? $\rightarrow$ Gaussian distribution ✓
2. Number of clusters? $\rightarrow$ 2 clusters ✓
3. **What parameters**?
    - 2 means
    - 2 proportions
    - 2 sd $\rightarrow$ both equal 1 ✓

$\Rightarrow$ 4 parameters to be estimated! (2 means and 2 proportions)

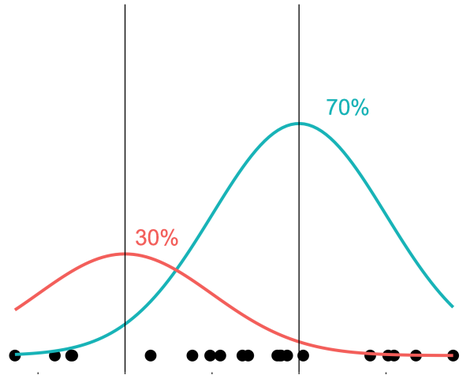
Two steps

1 Known probabilities →

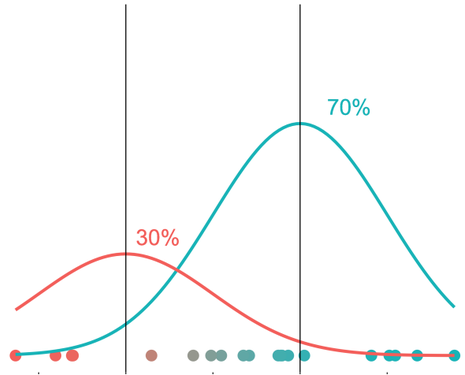
Estimate means and proportions



2 Known means and proportions →



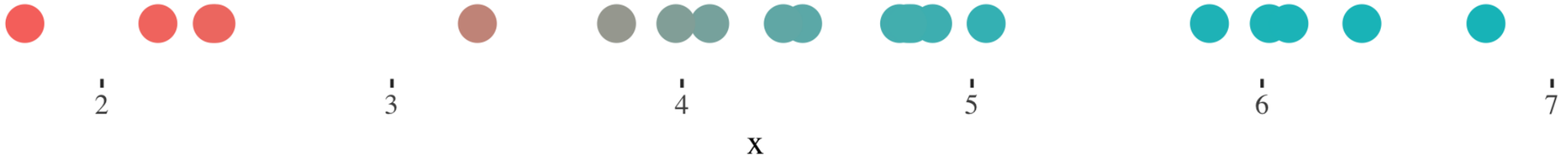
Estimate probabilities



Step 1: Known probabilities

```
> head(data_with_probs)
```

```
          x prob_red prob_blue
1 3.294453     0.64      0.36
2 5.818586     0.01      0.99
3 2.380493     0.92      0.08
4 4.415913     0.16      0.84
5 5.048659     0.05      0.95
6 4.750195     0.09      0.91
```



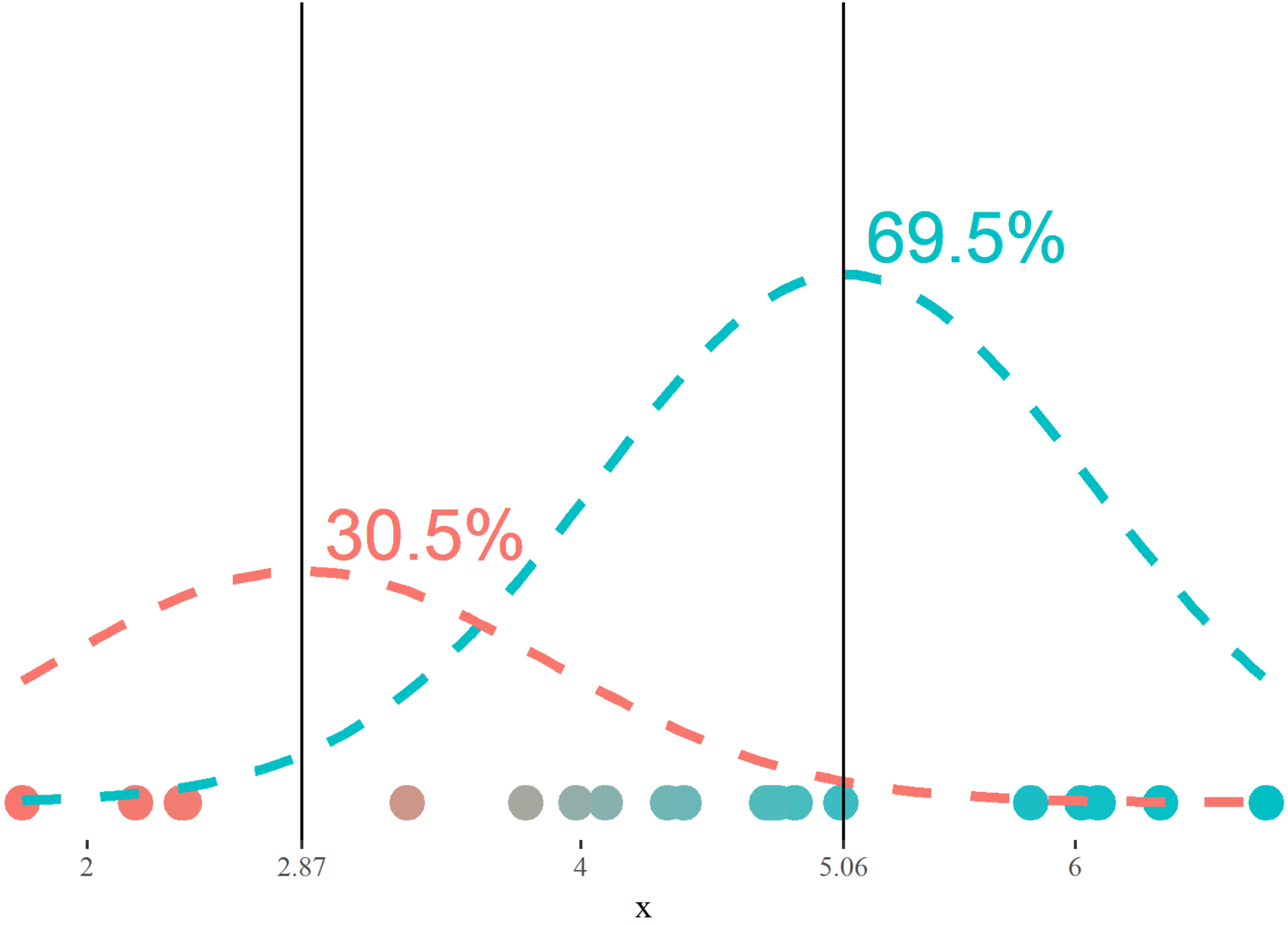
Step 1: Known probabilities

For the means

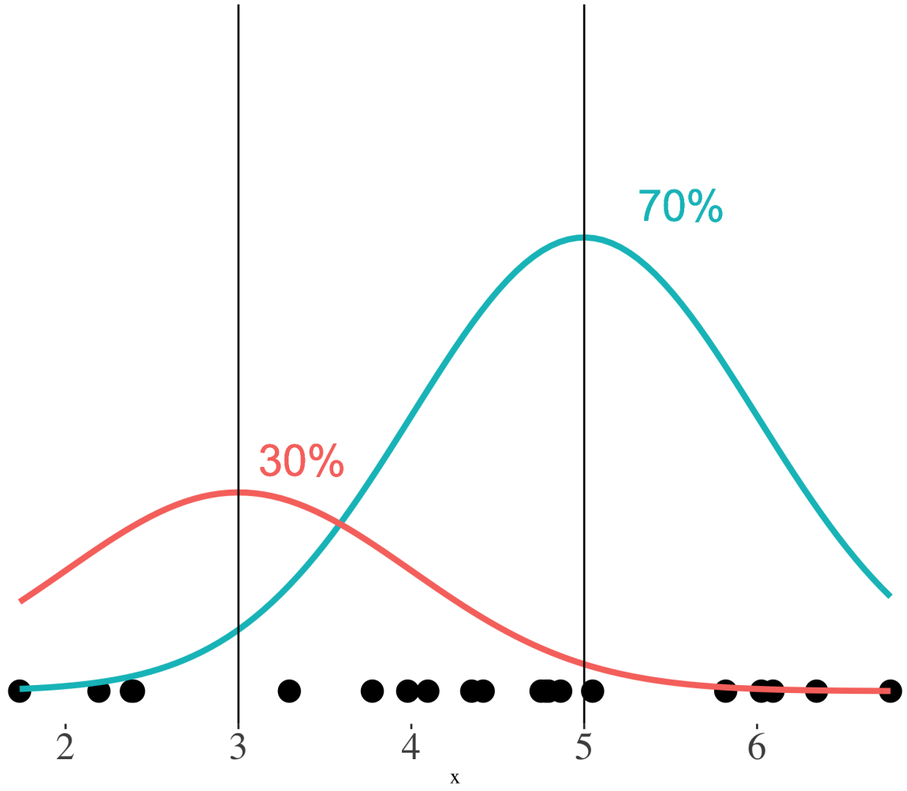
```
> means_estimates <- data_with_probs %>%
+   summarise(mean_red = sum(x * prob_red) / sum(prob_red),
+             mean_blue = sum(x * prob_blue) / sum(prob_blue))
> means_estimates

  mean_red mean_blue
1  2.86925  5.062976
```

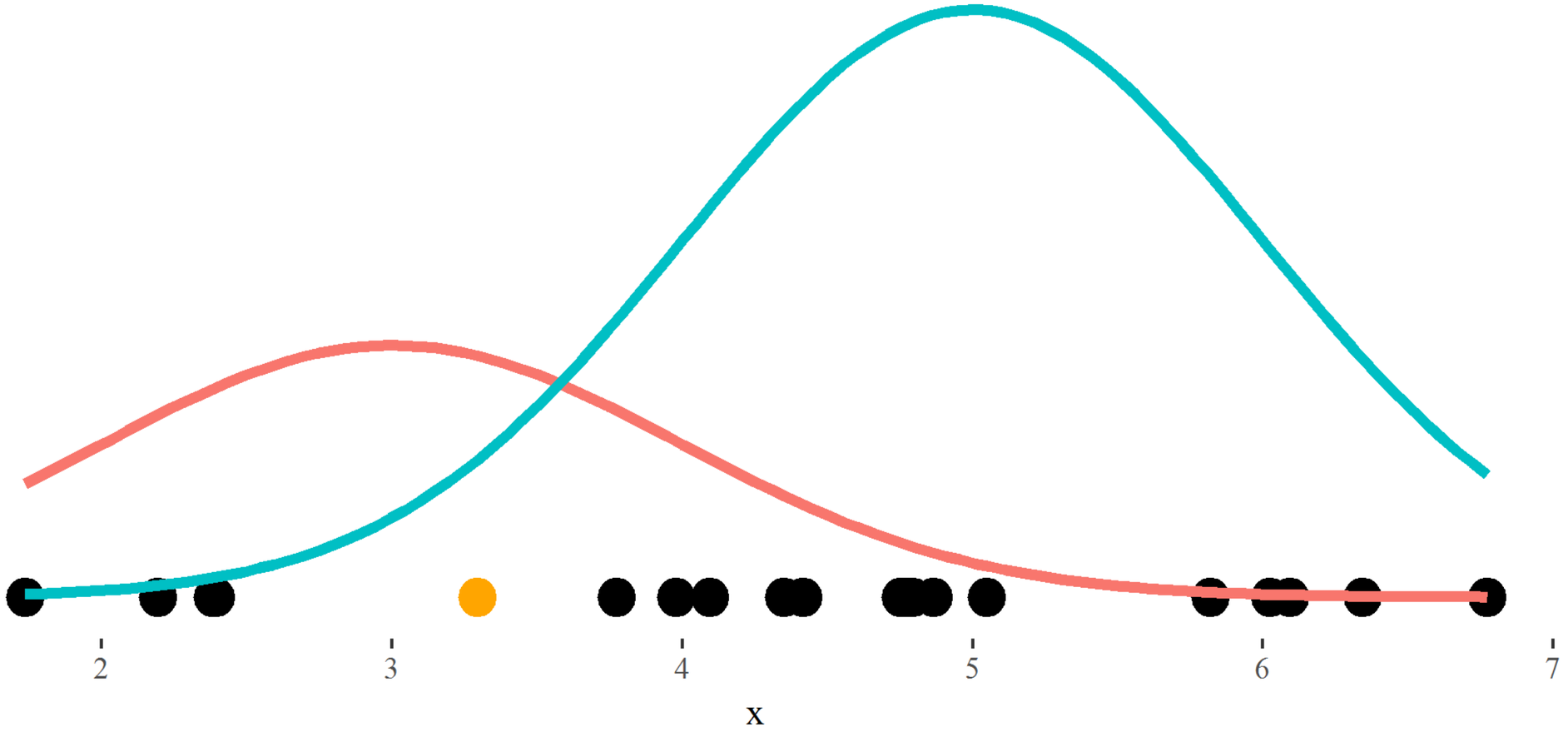
For the proportions

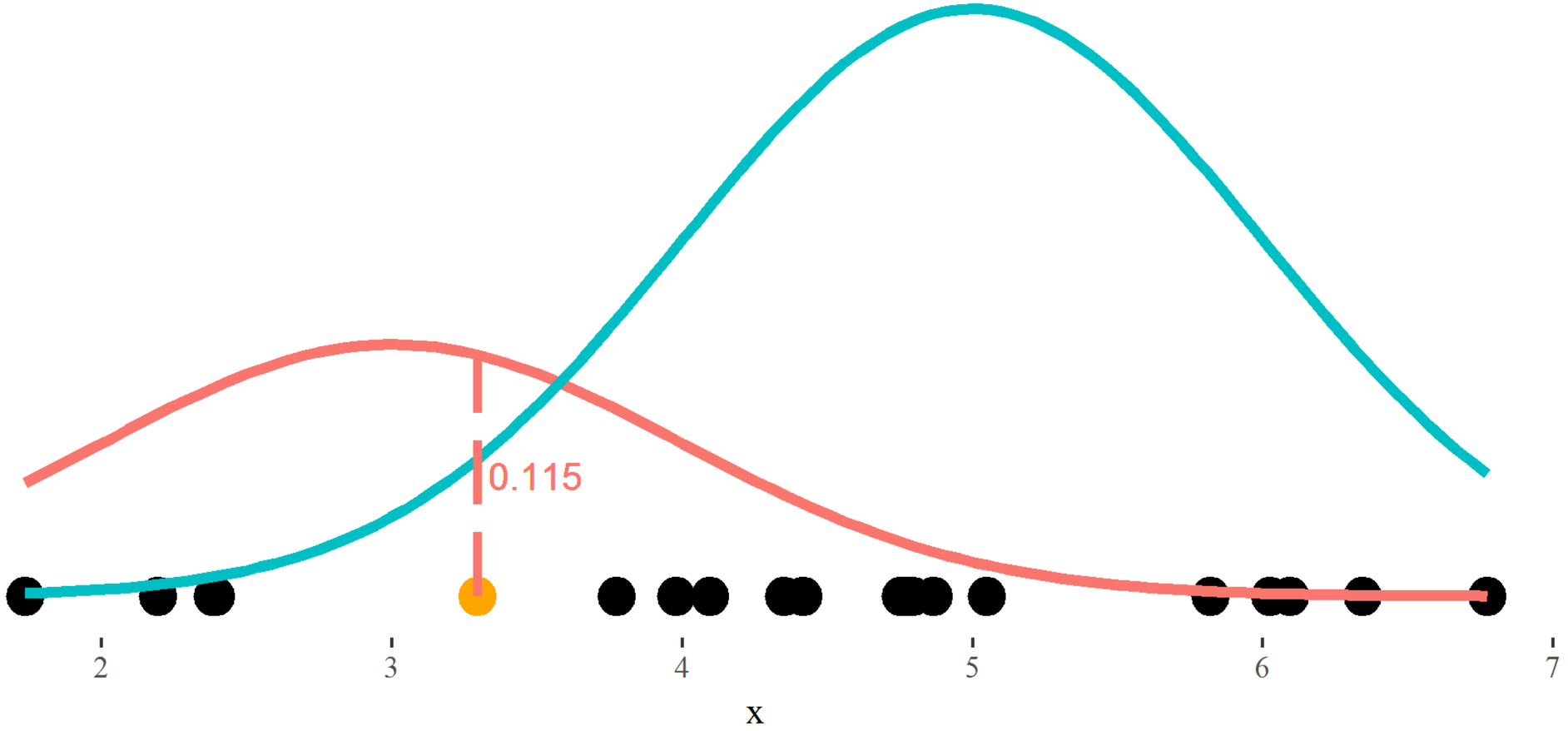
```
> proportions_estimates <- data_with_probs %>%
+   summarise(proportion_red = mean(prob_red),
+             proportion_blue = 1 - proportion_red)
> proportions_estimates

  proportion_red proportion_blue
1          0.305           0.695
```

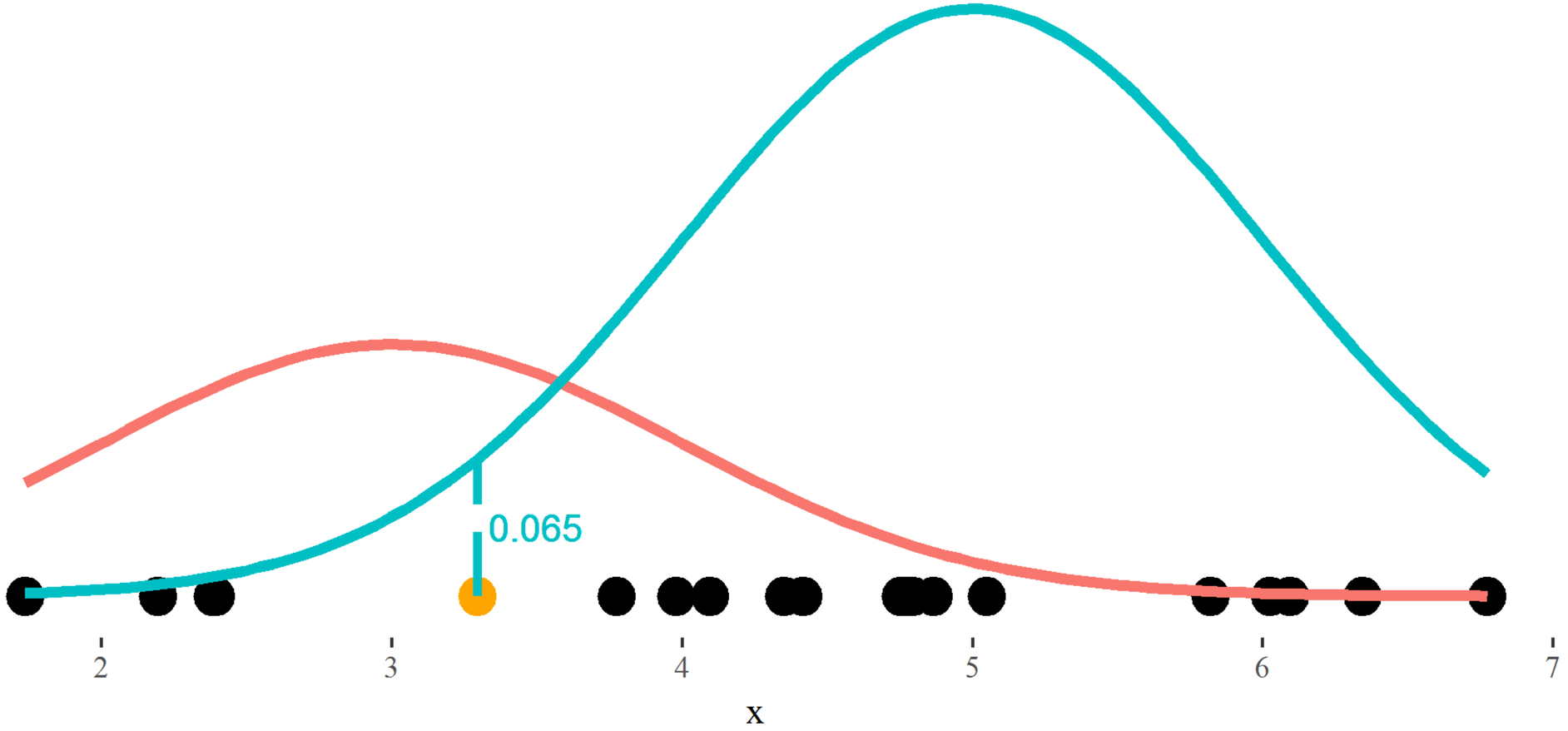

69.5%
30.5%
2
2.87
4
5.06
6
x

Step 2: Known means and proportions




2
3
4
5
6
7
x


0.115
2
3
4
5
6
7
x


0.065
2
3
4
5
6
7
x

Step 2: Scaled probabilities

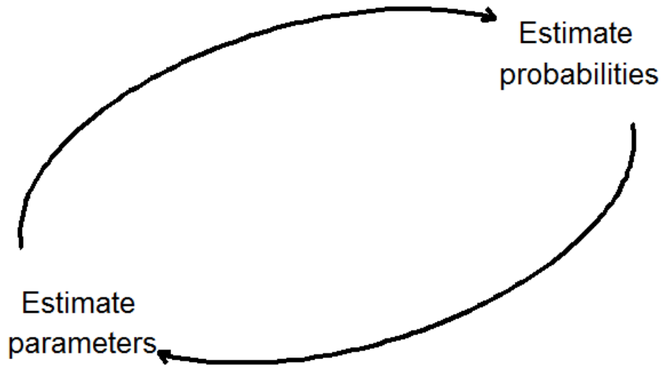
$$\text{Probability}_{\text{blue}} = \frac{0.065}{0.115+0.065} = 0.36$$

```
> data %>%
+   mutate(prob_from_red = 0.3 * dnorm(x, mean = 3),
+          prob_from_blue = 0.7 * dnorm(x,mean = 5),
+          prob_red = prob_from_red/(prob_from_red + prob_from_blue),
+          prob_blue = prob_from_blue/(prob_from_red + prob_from_blue)) %>%
+   select(x, prob_red, prob_blue) %>%
+   head()
```

```
         x   prob_red  prob_blue
1 3.294453 0.63733037 0.36266963
2 5.818586 0.01115698 0.98884302
3 2.380493 0.91619343 0.08380657
4 4.415913 0.15721146 0.84278854
5 5.048659 0.04999159 0.95000841
6 4.750195 0.08724975 0.91275025
```

Summary

- When we know the probabilities $\rightarrow$ **estimate means and proportions**
- When we know the means and proportions $\rightarrow$ **estimate the probabilities**



MIXTURE MODELS IN R

Let's practice!

MIXTURE MODELS IN R

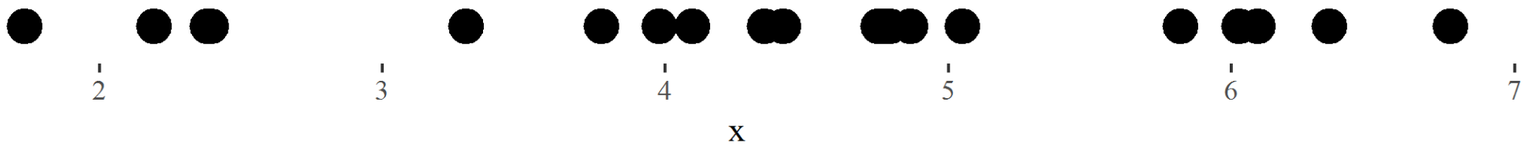
EM algorithm

Victor Medina
Researcher at SBIF

Same problem, this time for real

```
> head(data)
```

```
          x
1 3.294453
2 5.818586
3 2.380493
4 4.415913
5 5.048659
6 4.750195
```



Iteration 0: Initial parameters

Initial means

```
> means_init <- c(1, 2)
> means_init
```

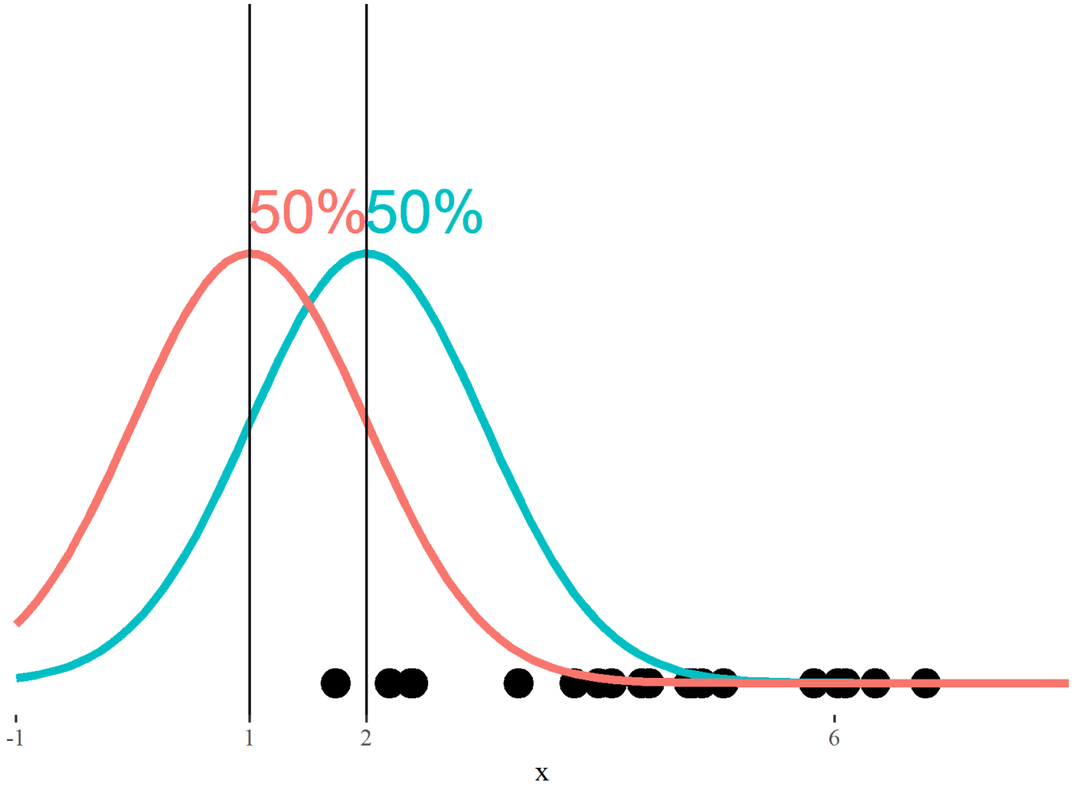
```
[1] 1 2
```

Initial proportions

```
> props_init <- c(0.5, 0.5)
> props_init
```

```
[1] 0.5 0.5
```

Iteration 0: Initial parameters



Iteration 1: Estimate probabilities (Expectation)

```
> data_with_probs <- data %>%
+   mutate(prob_from_red = props_init[1] * dnorm(x, mean = means_init[1]),
+          prob_from_blue = props_init[2] * dnorm(x, mean = means_init[2]),
+          prob_red = prob_from_red/(prob_from_red + prob_from_blue),
+          prob_blue = prob_from_blue/(prob_from_red + prob_from_blue)) %>%
+   select(x, prob_red, prob_blue)
> head(data_with_probs)
```

```
         x   prob_red prob_blue
1 3.294453 0.14252762 0.8574724
2 5.818586 0.01314364 0.9868564
3 2.380493 0.29307562 0.7069244
4 4.415913 0.05137250 0.9486275
5 5.048659 0.02795899 0.9720410
6 4.750195 0.03731988 0.9626801
```

Iteration 1: Estimate parameters (Maximization)

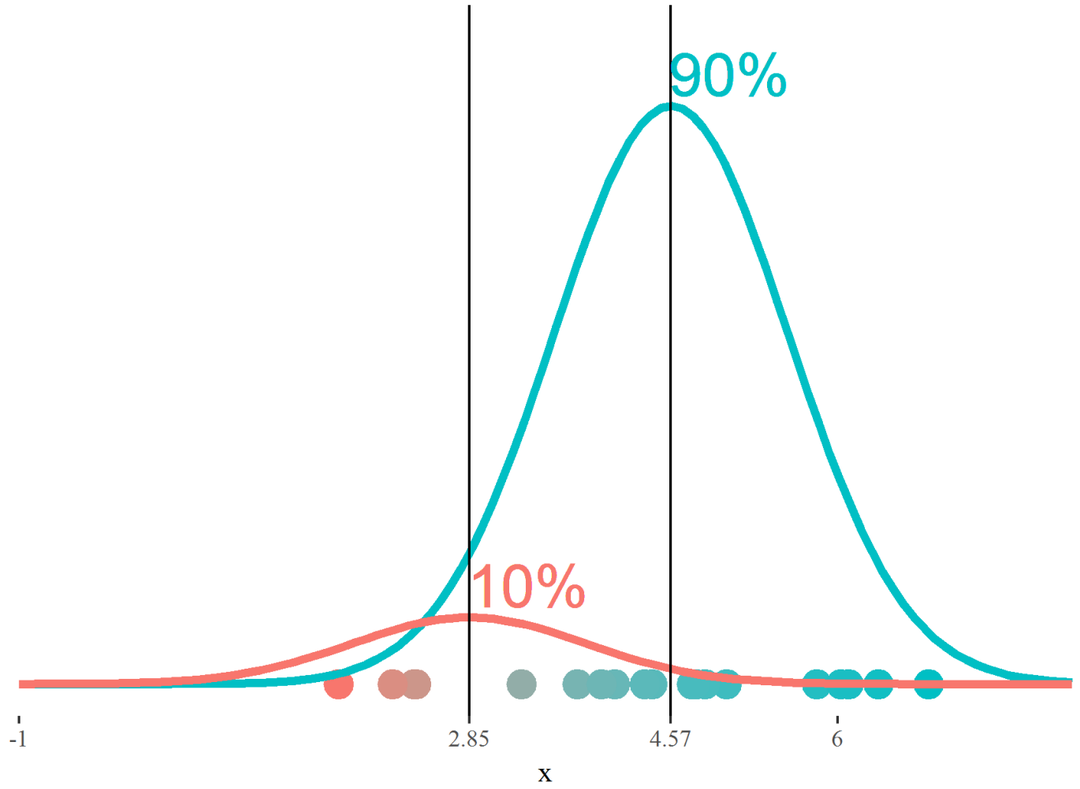
```
> means_estimates <- data_with_probs %>%
+   summarise(mean_red = sum(x * prob_red) / sum(prob_red),
+             mean_blue = sum(x * prob_blue) / sum(prob_blue)) %>%
+   as.numeric()
> means_estimates
```

```
[1] 2.848001 4.572862
```

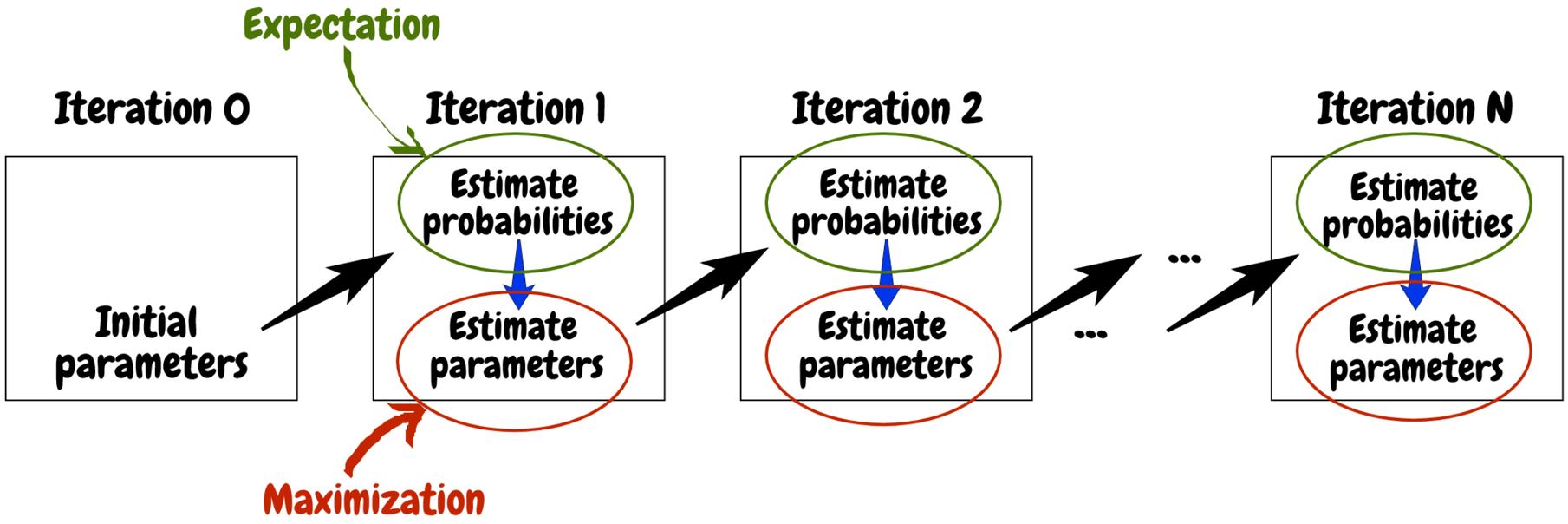
```
> props_estimates <- data_with_probs %>%
+   summarise(proportion_red = mean(prob_red),
+             proportion_blue = 1- proportion_red) %>%
+   as.numeric()
> props_estimates
```

```
[1] 0.1032487 0.8967513
```

Iteration 1: Estimate parameters (Maximization)



Expectation-Maximization algorithm



Expectation function

```
> # Expectation (known means and proportions)
> expectation <- function(data, means, proportions){
+
+   # Estimate the probabilities
+   data <- data %>%
+     mutate(prob_from_red = proportions[1] * dnorm(x, mean = means[1]),
+            prob_from_blue = proportions[2] * dnorm(x, mean = means[2]),
+            prob_red = prob_from_red/(prob_from_red + prob_from_blue),
+            prob_blue = prob_from_blue/(prob_from_red + prob_from_blue)) %>%
+     select(x, prob_red, prob_blue)
+
+   # Return data with probabilities
+   return(data)
+ }
```

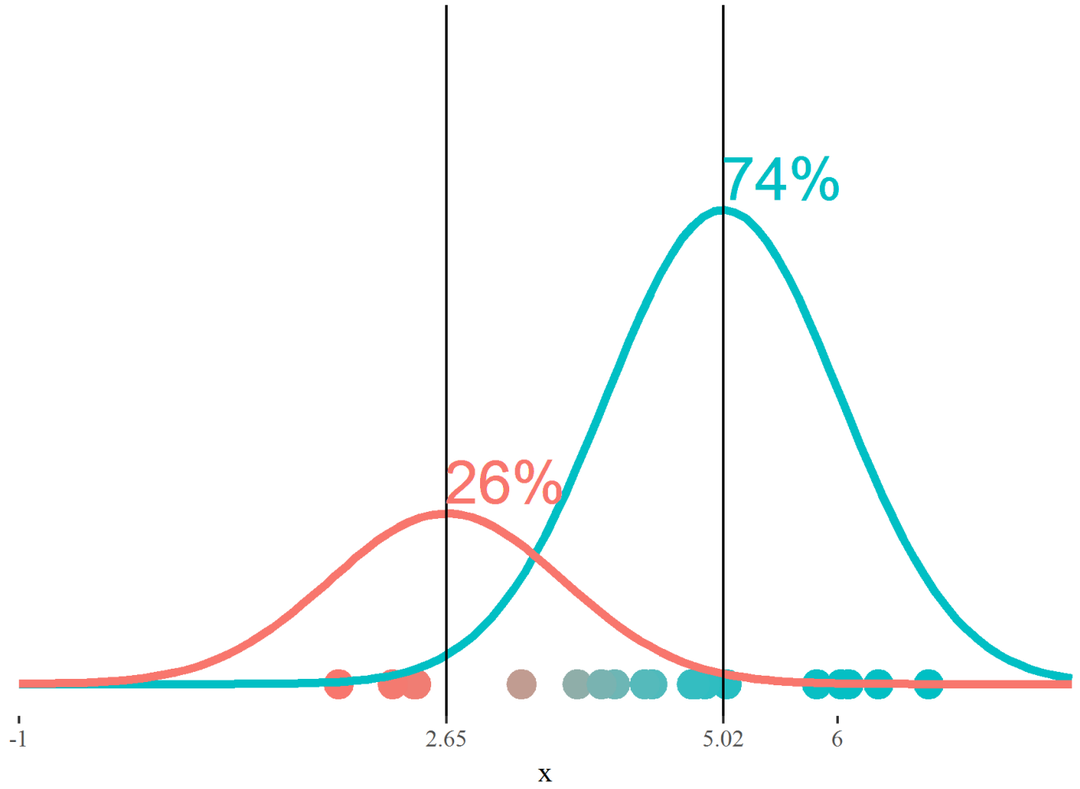
Maximization function

```
> # Maximization (known probabilities)
> maximization <- function(data_with_probs){
+
+   # Estimate the means
+   means_estimates <- data_with_probs %>%
+     summarise(mean_red = sum(x * prob_red) / sum(prob_red),
+               mean_blue = sum(x * prob_blue) / sum(prob_blue)) %>%
+     as.numeric()
+
+   # Estimate the proportions
+   proportions_estimates <- data_with_probs %>%
+     summarise(proportion_red = mean(prob_red),
+               proportion_blue = 1 - proportion_red) %>%
+     as.numeric()
+
+   # Return the results
+   list(means_estimates, proportions_estimates)
+ }
```

Iteratively

```
> # Iterative process
> for(i in 1:10){
+   # Expectation-Maximization
+   new_values <- maximization(expectation(data, means_init, props_init))
+
+   # New means and proportions
+   means_init <- new_values[[1]]
+   props_init <- new_values[[2]]
+
+   # Print results
+   cat(c(i, means_init, proportions_init),"\n")
+ }

1 2.848001 4.572862 0.1032487 0.8967513
2 2.469715 4.736764 0.1508531 0.8491469
3 2.411235 4.863675 0.1911983 0.8088017
4 2.455946 4.929702 0.2162419 0.7837581
5 2.511132 4.96399 0.232063 0.767937
6 2.556729 4.984427 0.2428862 0.7571138
7 2.59167  4.998099 0.2507144 0.7492856
8 2.618177 5.007884 0.2565634 0.7434366
9 2.638406 5.015153 0.261021 0.738979
10 2.653982 5.020675 0.264463 0.735537
```

After 10 iterations



MIXTURE MODELS IN R

Let's practice!