00_simulated_data

Frances Lin

11/11/2020

Description

This .Rmd file creates simulated data for the two-component Gaussian mixture. All related results can be found in the results folder.

Load packages

```
library(here)
library(tidyverse)
library(tibble)
library(ggplot2)
library(gridExtra)
```

Simulate data

Specify the values and parameters for the two-component Gaussian mixture

```
# Specify the #s of obversations for each component
n <- 100
k <- 2 # might want to extend it to kth mixture

# Specify mean and sd of component 1 (or A)
mu_1 <- 0
sd_1 <- 1

# Specify mean and sd of component 2 (or B)
mu_2 <- 2 #4 #2
sd_2 <- 1
```

Create a dataframe to store the simulated data

```
set.seed(1234) # for reproducibility
data <- tibble(
  component = c(rep("A", n), rep("B", n)),
  value = c(rnorm(n, mean = mu_1, sd = sd_1), rnorm(n, mean = mu_2, sd = sd_2))
)
head(data)
```

```
## # A tibble: 6 x 2
##   component  value
##   <chr>      <dbl>
## 1 A         -1.21
## 2 A          0.277
```

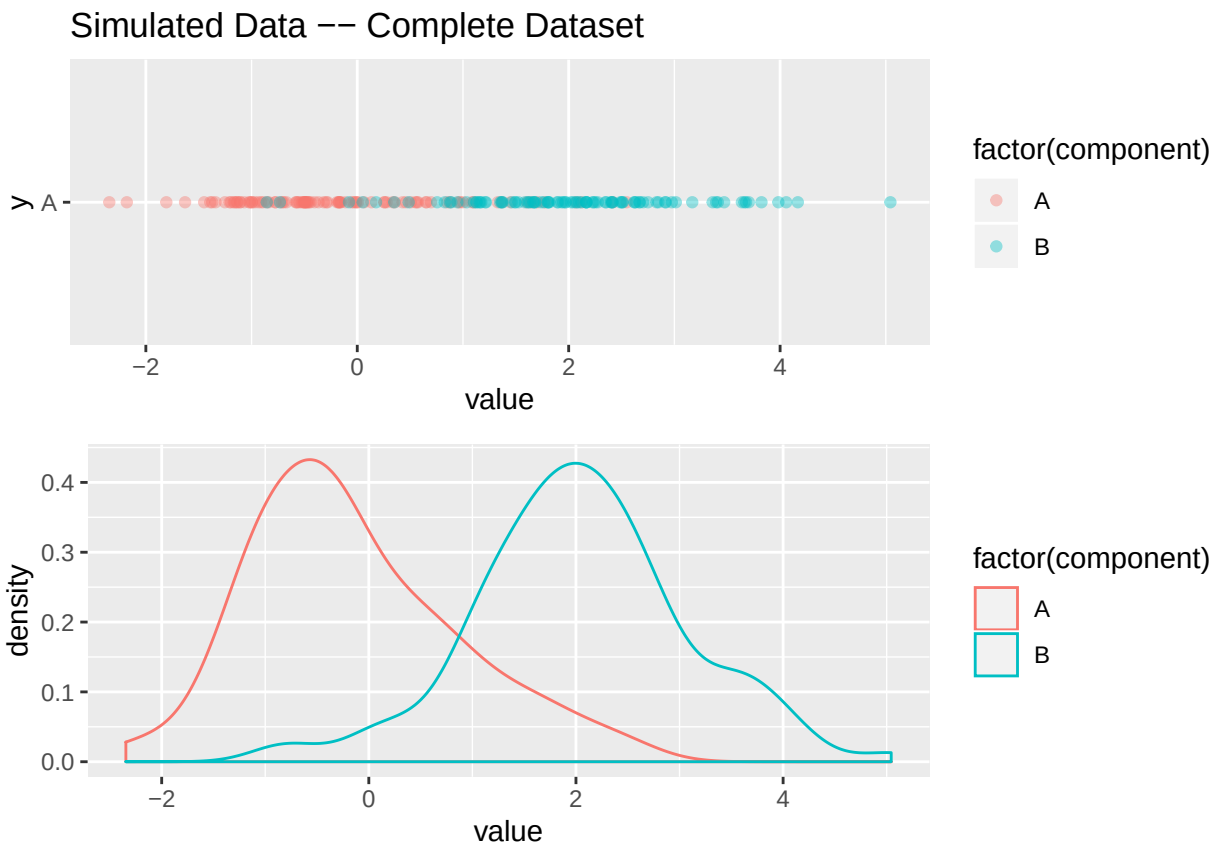
```
## 3 A          1.08
## 4 A         -2.35
## 5 A          0.429
## 6 A          0.506
```

Plot simulated data

```
data_flat_plot <- data %>%
  ggplot(aes(x = value, y = "A", color = factor(component))) +
  geom_point(alpha = 0.4) +
  ggtitle("Simulated Data -- Complete Dataset")
#data_flat_plot
```

```
data_density_plot <- data %>%
  ggplot(aes(x = value, color = factor(component))) +
  geom_density()
#data_density_plot
```

```
data_plot <- grid.arrange(data_flat_plot, data_density_plot)
```




```
data_plot
```

```
## TableGrob (2 x 1) "arrange": 2 grobs
```

```
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
```

Estimate the mean, var, and sd for this complete dataset (component A and B) using MLE

```
data_summary <- data %>%
  group_by(component) %>%
  summarize(
    mean = mean(value),
#    var = var(value), # never mind dnorm() takes sd
    var = var(value),
    sd = sd(value)
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
data_summary
```

```
## # A tibble: 2 x 4
##   component   mean   var    sd
##   <chr>      <dbl> <dbl> <dbl>
## 1 A         -0.157  1.01  1.00
## 2 B          2.04   1.07  1.03
```

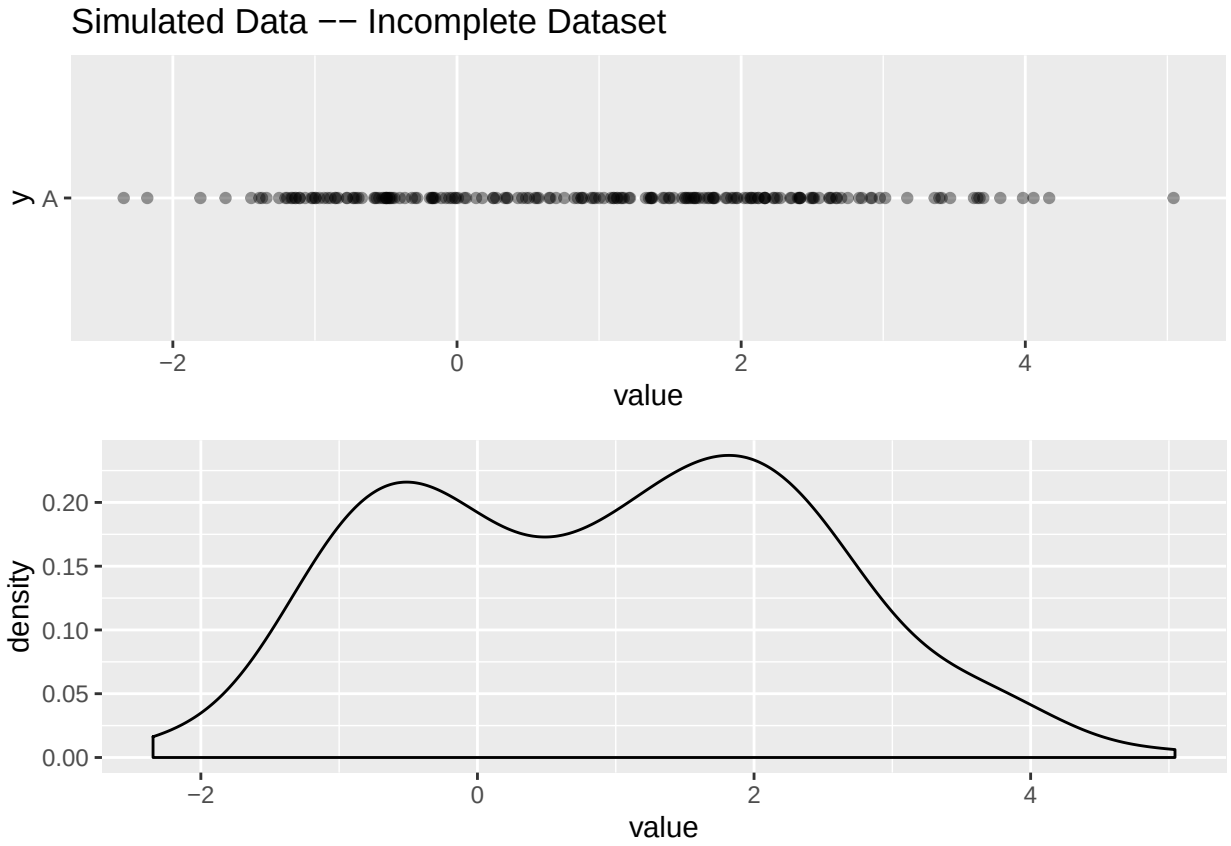
Plot simulated data

Assume there is a latent (or missing) variable

```
data_flat_plot_latent <- data %>%
  ggplot(aes(x = value, y = "A")) +
  geom_point(alpha = 0.4) +
  ggtitle("Simulated Data -- Incomplete Dataset")
#data_flat_plot_latent
```

```
data_density_plot_latent <- data %>%
  ggplot(aes(x = value)) +
  geom_density()
#data_density_plot_latent
```

```
data_plot_latent <- grid.arrange(data_flat_plot_latent, data_density_plot_latent)
```




```
data_plot_latent
```

```
## TableGrob (2 x 1) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
```

We'll see how do we estimate the mean and sd for this incomplete dataset (with components removed) next

Save out results

```
write_rds(data, here("results", "data.rds"))
write_rds(data_plot, here("results", "data_plot.rds"))
write_rds(data_summary, here("results", "data_summary.rds"))
write_rds(data_plot_latent, here("results", "data_plot_latent.rds"))
```