# HW 1

## Question 1: EM for the PANSS data

Load the `Surrogate` package in R and load the dataset `Schizo_PANSS`:

```
library(Surrogate)
data("Schizo_PANSS")
```

The dataset combines five clinical trials aimed at determining if risperidone decreases the Positive and Negative Syndrome Score (PANSS) over time compared to a control treatment for patients with schizophrenia. These are longitudinal trials where patients are assessed at weeks 1, 2, 4, 6 and 8 after being assigned to a treatment arm.

Each row in the dataset is a different trial participant, and `Week1`, `Week2`, `Week4`, `Week6`, `Week8` records the change in PANSS from baseline. The variable `Treat` represents whether the patient was enrolled in the control (-1) or if the patient was in the risperidone arm (1).

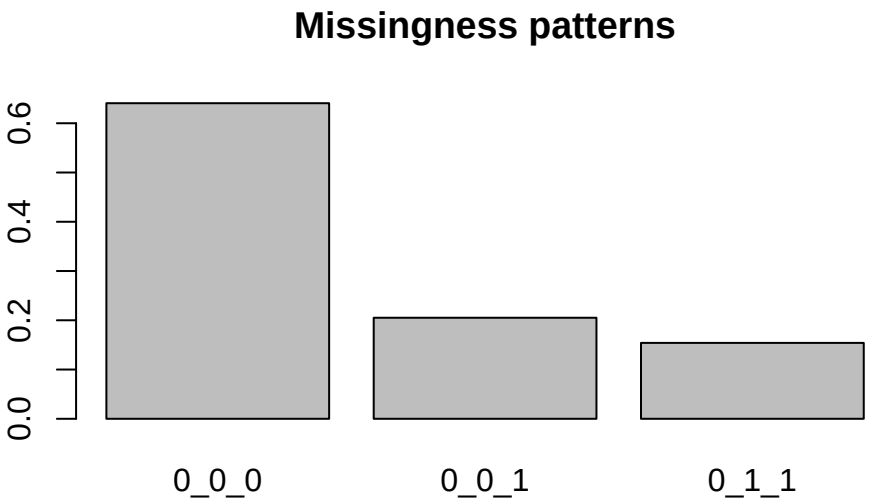
Subset the data to include only `Week1`, `Week4` and `Week8`, and including only complete cases, dropouts between week 4 and week 8, and dropouts between weeks 1 and weeks 4.

```
hw_data <- Schizo_PANSS[,c("Id","Treat","Week1","Week4","Week8")] |>
  subset((!is.na(Week1) & !is.na(Week4) & !is.na(Week8))
         | (!is.na(Week1) & !is.na(Week4) & is.na(Week8))
         | (!is.na(Week1) & is.na(Week4) & is.na(Week8)))
```

Double-checking we did the subsetting correctly:

```
gen_miss_patterns <- function(mat) {
  patterns <- mat |>
    is.na() |>
    apply(2,as.integer) |>
    apply(1, \(row) {
      paste(row,collapse = "_")
    })
  return(patterns)
```

```
}
y_data <- hw_data[,c("Week1","Week4","Week8")]
miss_patterns <- gen_miss_patterns(y_data)
tab_miss <- table(miss_patterns)
prop_miss <- prop.table(tab_miss)
prop_miss <- sort(prop_miss,decreasing = TRUE)
barplot(prop_miss, main = "Missingness patterns")
```




Of the remaining data, 20% of cases dropped out before the final week, and 15% of cases dropped out between weeks 1 and 4.

We're going to fit the following model to this dataset:

$$y_i \mid \text{Treat}_i \sim \text{Normal}(\mu + \beta_1 \text{Treat}_i + \beta_2 t + \beta_3 \text{Treat}_i t, \Sigma)$$

where $t$ is the vector $1, 4, 8$ indicating at what time points the measurements were taken and $\mu$ is a scalar mean, under the somewhat dubious assumption of ignorable dropout.

In order to do so, we can use an Expectation-Conditional-Maximization algorithm:

1. Initialize with $\beta^{(1)}, \Sigma^{(1)}$, and a value $\epsilon$
2. For $t = 1, 2, ...$
    a. Compute $\mathbb{E}\left[y_i \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$ and $\mathbb{E}\left[y_i y_i^T \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$ for all $i$
    b. Update $\beta^{(t)}$ to $\beta^{(t+1)}$

$$\beta^{(t+1)} = \left(\sum_i X_i^T (\Sigma^{(t)})^{-1} X_i\right)^{-1} \sum_i X_i^T (\Sigma^{(t)})^{-1} \mathbb{E}\left[y_i \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$$

    c. Update $\Sigma^{(t)}$ to $\Sigma^{(t+1)}$

$$\Sigma^{(t+1)} = \frac{1}{n}\sum_i \mathbb{E}\left[(y_i - X_i\beta^{(t+1)})(y_i - X_i\beta^{(t+1)})^T \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$$

d. If $Q(\beta^{(t+1)}, \Sigma^{(t+1)} \mid \beta^{(t)}, \Sigma^{(t)}) - Q(\beta^{(t)}, \Sigma^{(t)} \mid \beta^{(t)}, \Sigma^{(t)}) < \epsilon$, stop, otherwise, return to step a.

### Part a

Fit your model to simulated data

```
set.seed(123)
n <- 1000
p <- 5
K <- 3
X <- list()
beta <- rnorm(p)
L <- matrix(rnorm(K * K),K,K)
Sigma <- L %*% t(L)
phi <- rnorm(p)
y <- list()
### d will hold our indicators for which group each patient is in.
### d = 1: dropout between weeks 1 and 4
### d = 2: dropout between weeks 4 and 8
### d = 3: all values are observed
d <- rep(NA_integer_,n)
for (i in 1:n) {
  X[[i]] <- matrix(rnorm(p * K),K,p)
  y[[i]] <- X[[i]] %*% beta + MASS::mvrnorm(mu = rep(0,K), Sigma = Sigma)
  logit_p_m <- X[[i]] %*% phi + c(-2, -1, 1)
  p_m <- exp(logit_p_m) / sum(exp(logit_p_m))
  d_i <- rmultinom(1, 1, p_m)
  d[i] <- which(as.logical(d_i))
  y[[i]] <- y[[i]][1:d[i]]
}
```

For this model, we will assume that even if you drop out, we still observe your covariates for all time points.

Use the expressions from lecture 11's notes to derive asymptotic standard errors from the observed information for the observed data for your estimates for beta. Compute marginal asymptotic confidence intervals for each element of beta and test whether the true values lie in those intervals.

Report how many steps it took for your model to converge.

### Part b

Now fit your model to the real data, report the MLEs and the standard errors for your $\beta$ coefficients.

### Part c

Instead of ECM, change your algorithm to an EM algorithm which will involve running your maximization to convergence for each EM step.

1. Initialize with $\beta^{(1)}, \Sigma^{(1)}$, and a value $\epsilon$
2. For $t = 1, 2, ...$
   a. Compute $\mathbb{E}\left[y_i \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$ and $\mathbb{E}\left[y_i y_i^T \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$ for all $i$
   b. For $s = 1, 2, ...$
      i. At $s = 1$, set $\beta^{(s)} = \beta^{(t)}, \Sigma^{(s)} = \Sigma^{(t)}$
      ii. Update $\beta^{(s)}$ to $\beta^{(s+1)}$

$$\beta^{(s+1)} = \left(\sum_i X_i^T (\Sigma^{(s)})^{-1} X_i\right)^{-1} \sum_i X_i^T (\Sigma^{(s)})^{-1} \mathbb{E}\left[y_i \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$$

      iii. Update $\Sigma^{(s)}$ to $\Sigma^{(s+1)}$

$$\Sigma^{(s+1)} = \frac{1}{n} \sum_i \mathbb{E}\left[(y_i - X_i\beta^{(s+1)})(y_i - X_i\beta^{(s+1)})^T \mid y_{i(0)}, \beta^{(t)}, \Sigma^{(t)}\right]$$

      iv. Iterate until $\beta^{(s)}, \Sigma^{(s)}$ reach a stationary point
   c. Set $\beta^{(t+1)} = \beta^{(s)}, \Sigma^{(t+1)} = \Sigma^{(s)}$
   d. If $Q(\beta^{(t+1)}, \Sigma^{(t+1)} \mid \beta^{(t)}, \Sigma^{(t)}) - Q(\beta^{(t)}, \Sigma^{(t)} \mid \beta^{(t)}, \Sigma^{(t)}) < \epsilon$, stop, otherwise, return to step a.

Fit this model to your simulated dataset, and report how many steps it took for this EM algorithm to converge versus the ECM algorithm.