3 **(EM for Factor analysis with Sparse Loading)** (a) For the algorithm in Question 2, suppose that we add a lasso penalty into the loglikelihood function to estimate the sparsity structure of the loading matrix. Derive and implement the EM algorithm with such lasso penalty. Organize the algorithm as a Python class so that it maximally reuses your implementation in Question 2. Credit will be given proportionally to the amount of code duplicated (the less, the better).

$$Q_{lasso}(\theta) = \tilde{E}_{X|Y}[\log p(Y,X|\theta)] - \lambda \sum_{j=1}^{p}\sum_{k=1}^{L}|\Lambda_{jk}|$$

E step still same to 2 because penalty only depends on parameters

M step: In the updates of Λ:

for j-th row Λ_j, here minimize $f(\Lambda_j) = \frac{1}{2}\Lambda_j^T A \Lambda_j - \Lambda_j^T B_j + \lambda \psi_{jj} \|\Lambda_j\|_1$,

$A = \sum_{i=1}^{n} E[X_i X_i^T | Y_i]$, $B_j = \sum_{i=1}^{n} Y_{ij} E[X_i | Y_i]^T$

By taking subgradient to single element Λ_{jk} and setting it to zero:

$\Lambda_{jk} = \frac{1}{A_{kk}} \text{sign}(C_{jk}) \max(0, |C_{jk}| - \lambda \psi_{jj})$, $C_{jk} = B_{jk} - \sum_{m \neq k} A_{km} B_{jm}$

μ and ψ still same to 2