# Overview

This package implements Expectation-Maximization (EM) and Monte Carlo EM (MCEM) algorithms for missing data imputation under a multivariate and Poisson assumption.

The MCEM is available under multivariate assumption only, which allows Monte Carlo sampling when the analytical expectation in the E-step becomes intractable.

# Installation instruction.

|  |  |  |
| --- | --- | --- |
| Step | Action | Code |
| 1 | Install required library | **library("ggplot2")** |
| 2 | Install the package from GITHUB. | remotes::install\_github("ngamihimihi/DATA501\_Project",  subdir = "DATA501Package",  build\_vignettes = TRUE,  INSTALL\_opts = c("--install-tests")) |
| 3 | Load required package for testing. | library(“DATA501Package”) |
| 4 | Run available test. | testthat::test\_package("DATA501Package", "tap") |

# Test plan

## Objective

To test and validate the functionality of an R package that implements the EM and MCEM algorithms for missing data imputation under the multivariate normal and poisson assumption. Please note that MCEM algorithms is only avaialble for multivariate assumption.

## Test scope

Function ready for testing.

* e\_step\_nvnorm\_em
* e\_step\_nvnorm\_mcem
* e\_step\_poisson\_em
* em\_engine (available distribution: poisson and mvnorm)
* initialise\_parameters
* initialise\_parameters\_poisson
* initilise\_parameters\_mvnorm
* log\_likelihood\_nvnorm
* log\_likelihood\_poisson
* m\_step\_nvnorm
* m\_step\_poisson
* run\_em\_algorithm

Method ready for testing:

* plot(result,what="loglik")
* summary(result)

## Test Categories and Cases

### a. Initialization functions.

- Functions: initialize\_parameters\_nvnorm()

|  |  |  |
| --- | --- | --- |
| Test ID | Desciption | Expected outcome |
| 1.1 | Input valid numeric matrix with some NAs | Returns valid mu and sigma, no error |
| 1.2 | Input with a column entirely NA | Throws error |
| 1.3 | Input with a row entirely NA | Issues warning, continues |
| 1.4 | Covariance matrix not PD | Issues message, applies jitter |
| 1.5 | Invalid input (e,g.: dataframe or character matrix) | Throws error |

### b. E-step functions

e\_step\_nvnorm\_em(), e\_step\_nvnorm\_mc(),e\_step\_poisson\_em()

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Applicable function | Desciption | Expected outcome |
| 2.1 | All three | Inputed matrix matches the shape of input | Dimension equal |
| 1.2 | All three | Observed values remain unchanged | Same as input |
| 1.3 | All three | Missing values are imputed | NA replaced |
| 1.4 | e\_step\_nvnorm\_mc() | For MC version,accept\_rate attribute exists | Attribute exists and is numeric |

### c. Log-likelihood function.

- Function: log\_likelihood\_nvnorm(),log\_likelihood\_poisson()

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Applicable functions | Description | Expected outcome |
| 3.1 | Both functions | Expected output data type | Output is of double type and is a finite value |
| 3.2 | Both functions | No error thrown with correct input | No error |
| 3.3 | log\_likelihood\_nvnorm() | Function is able to handle single row data | Correct data type output and no error thrown |
| 3.4 | log\_likelihood\_poisson() | Function is able to handles NA values | Loglikelihood is calculated successfully |
| 3.5 | log\_likelihood\_poisson() | Function throw errors with mismatched lambda and input data length  e.g: lambda vector has length 3 while data matrix has 2 columns. | Error throw |
| 3.6 | log\_likelihood\_poisson() | Function returns NA for invalid lambda | Output returns is NA |
| 3.7 | log\_likelihood\_poisson() | Function returns NA for invalid data values | Output returns is NA |
| 3.8 | log\_likelihood\_poisson | Function works for edge case zeros | Output is calculated successfully |

### d. Main Engine and EM Algorithm

Function: em\_engine(), run\_em\_algorithm()

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Applicable function | Description | Expected Outcome |
| 4.1 | Both | Run without error for valid data | Return updated em\_model |
| 4.2 | Both | parameter\_history length == number of iterations | Confirmed |
| 4.3 | Both | Early stopping triggers | Stop before max\_iter if tolerance met |
| 4.4 | Both | MCEM accept Monte Carlo parameters | Return updated em\_model; |

## Instruction to test submission.

#### 1. Install dependencies.

Make sure all the following packages are installed:

install.packages(c("devtools", "testthat", "rmarkdown", "knitr"))

#### 2. Install the package.

Install package from GITHUB

#### 3. Run all unit tests:

Current unit tests are prepared for the 2 main object and function.

To run the unit test:

devtools::test()

#### 4. Use Test data

Dependency: dplyr, data needs to be converted to matrix before passing on to run\_em\_algorithm

Code to import and test:

data<-read.csv("kc\_house\_data.csv",skip=1,header = FALSE)

head(data,5)

data<-data[,-c(1,2)]

data <- as.matrix(data)

model <- em\_model(data,distribution = "nvnorm",method = "EM")

model\_em <- em\_model(data,distribution = "nvnorm",method = "EM")

model\_mcem<- em\_model(data,distribution = "nvnorm",method = "EM")

#View result

#Standard EM

model\_em$data

model\_em$method

model\_em$early\_stop

model\_em$loglik\_history

model\_em$distribution

model\_em$parameters

model\_em$parameter\_history

head(model\_em$imputed,5)

#Monte Carlo EM

model\_mcem$data

model\_mcem$method

model\_mcem$early\_stop

model\_mcem$loglik\_history

model\_mcem$distribution

model\_mcem$parameters

model\_mcem$parameter\_history

head(model \_mcem$imputed,5)