Math 517 - Main Project

Federico Di Gennaro, Elsa Farinella, Marco Scialanga

Table of contents

1 Introduction

The Expectation-Maximization (EM) algorithm is an iterative method used to perform maximum likelihood estimation in the presence of missing data, or when it might be helpful to think of our data as if there were some latent variables (e.g., when estimating mixture distributions) [1]. The algorithm first estimates the unobserved values (the *expectation* step), then optimizes the likelihood (the *maximization* step), repeating these two steps until convergence to a stationary point.

In this project, our goal is to investigate when the EM algorithm is a good option for statistical inference in the presence of missing data. To answer this question, we will consider different percentages and mechanisms of missing data and apply the EM algorithm for various tasks. In Section 2, we generate observations from the multivariate normal distribution, introduce NA's and then apply the EM algorithm to estimate the mean vector μ, the covariance matrix Σ and the weight vector β in linear and logistic regression settings. In Section 3, we generate data in a similar way and then compare the performances of the EM algorithm in terms of parameter estimates with those obtained with maximum likelihood after imputation.

1.1 Missing Data Mechanisms

The most well-known missing data mechanisms are the three introduced in [2]: *Missing Completely at Random* (MCAR), *Missing at Random* (MAR), *Missing not at Random* (MNAR).

When dealing with missing values, we divide our data matrix X in its observed part X_{OBS} and its missing part X_{MIS}: $X = (X_{OBS}, X_{MIS})$. For any vector of data X (i.e. the column of a dataset), we can also introduce the random variable $M \in \{0, 1\}^n$ such that $M_i = 1$ if X_i is missing and $M_i = 0$ otherwise. The mechanism will depend on the distribution of M. We can now present the definitions of the three missing data mechanisms mentioned above by describing for each of them the relationship between X_{OBS}, X_{MIS}, M.

MCAR: We have MCAR data when the probability of a value being missing is independent from (X_{OBS}, X_{MIS}), hence if: $\mathbb{P}_M(M|X_{OBS}, X_{MIS}) = \mathbb{P}_M(M)\ \forall X_{OBS}, X_{MIS}$.
MAR: We have MAR data when the probability of a value being missing is dependent on the observed values X_{OBS}, but not on X_{MIS}. Formally, $\mathbb{P}_M(M|X_{OBS}, X_{MIS}) = \mathbb{P}_M(M|X_{OBS})\ \forall X_{MIS}$.
MNAR: We have MNAR data in all other cases, i.e., when the probability of a value being missing depends on the unobserved data.

In Figure 1, we can see how data can change after introducing NA's in the first variable with the MCAR, MAR, and MNAR mechanisms: for MCAR, the distribution is not heavily affected; for MAR and MNAR, on the other hand, the data looks very different from its original shape (note: there are infinitely many ways to produce MAR and MNAR data, in this project we used the functions from the `missMethods` library [3] and the function `produce_na`, which internally calls `ampute` from the `mice` package [4]). This will cause the EM algorithm to perform more poorly when faced with these last two mechanisms.

In this project, we will investigate the behavior and performance of EM when faced with all three of these missing data mechanisms.

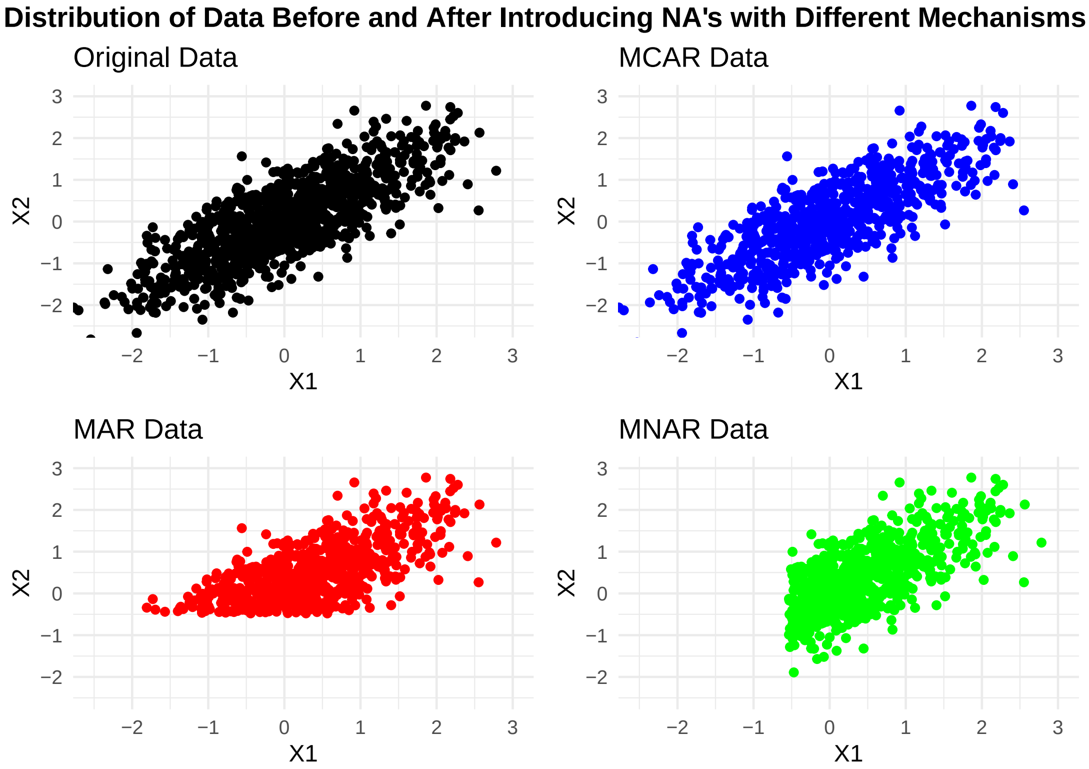

Figure 1: Visualization of data distribution before and after introducing missing data with the MCAR, MAR, and MNAR mechanisms.

2 Performance of EM with Missing Data on Different Tasks

As previously mentioned, the EM algorithm is an effective technique that is often used to perform MLE with missing data. In this section, we analyze how EM behaves when dealing with different proportions of missing values in the design matrix X, in three different frameworks:

1) *EM for Gaussian data:* Estimate the parameters μ and Σ of a multivariate normal distribution.

2) *EM for Linear Regression:* Estimate the parameters β of a linear regression model where the design matrix X is such that $X_i \sim \mathcal{N}_p(\mu, \Sigma),\ i = 1, ..., n$.

3) *EM for Logistic Regression:* Estimate the parameters β of a logistic regression model where the design matrix X is such that $X_i \sim \mathcal{N}_p(\mu, \Sigma),\ i = 1, ..., n$.

In all three cases, we evaluate the performance of the EM algorithm with the root mean squared error (RMSE), defined, taking $\beta \in \mathbb{R}^d$ as the true parameter and denoting the EM estimate by β_{EM}, as:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(\beta_{EM,i} - \beta_i)^2},$$

and analogously for matrices in $\mathbb{R}^{m \times n}$ (as in the case of Σ).

One aspect that we noticed when we began performing the experiments described above is that the accuracy of EM estimates varies significantly even for the same missing data percentage and mechanism. This can be easily explained: for example, with MCAR data and mean estimation for multivariate normal, we could have data missing symmetrically around the mean, so that mean estimation won't be affected much. However, it could also happen that a lot of the missing data is on one side of the true mean, thus heavily worsening the performance of mean estimation. Consequently, in order to generate informative visualizations, we repeat EM estimation several times (ranging from 15 to 600 depending on the experiment) for each incomplete data setting and average the $RMSE$'s obtained in each simulation. To conclude the introduction, it is important to note that the results obtained with the MAR and MNAR mechanisms should not be considered valid *for all MAR and MNAR settings*. As mentioned in the introduction, there are infinitely many ways to produce MAR and MNAR data. Some ways, of course, could cause the EM algorithm to perform worse or better than others. For example, in the case of mean estimation with MNAR data, if the missing values are introduced symmetrically around the mean by systematically removing the lowest and highest values, mean estimation won't be affected much. Thus, the right way to interpret results for MAR and MNAR data in Section 2 is that, for incomplete data generated with these mechanisms, the performance of EM can be *at least as bad* as the one we present in this project.

2.1 EM for Mean and Covariance Estimation

In this subsection, we will use the EM algorithm to estimate the mean μ and covariance matrix Σ of multivariate normal data X with missing values generated with the three mechanisms described in Section 1.1. We begin by generating $n = 300$ observations from the multivariate normal distribution with $\mu = (1, 2)^T$, $\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$. We will analyze how interesting patterns arise when dealing with more than two variables in Section 2.1.1.

Below we plot the densities generated with the estimated μ and Σ when 25% of the data is missing and the $RMSE$ obtained with the EM algorithm when dealing with the three different mechanisms for both μ and Σ and percentages of missing data ranging from 5 to 70%. We also plot the errors obtained with sample mean and sample covariance (dashed lines). We computed the latter by only keeping complete rows. Recall that these graphs are obtained by averaging over several (in this case: 200) repetitions of the same experiment. Thus, here and wherever else in the project the overall appearance of the plot is not negatively affected, we also plot the 95% confidence intervals (CI's) to show that the curves we obtain are representatitve of the true expected value of the results.

In Figure 2, we can see how MAR and MNAR data make the EM algorithm perform much more poorly, with their estimated density already being far from the original one. On the other hand, the density generated by μ_{EM} and Σ_{EM} after introducing NA's through the MCAR mechanism is quite similar to the true one.

Figure 3 and Figure 4 reveal distinct behaviors of the EM algorithm under the three missing data mechanisms: MCAR, MAR, and MNAR. Under the MCAR mechanism, where data is missing indiscriminately, the algorithm's performance is quite satisfactory at all percentages of missing data up to 70%, for both mean and covariance estimation. For mean estimation, we can see that the EM algorithm only slightly outperforms simply taking the sample mean; on the other hand, for covariance estimation, especially for larger (> 0.2) percentages of missing data, the EM algorithm clearly tops the simpler approach. One should keep in mind that the EM algorithm is quite more expensive, in computational resources, than simply taking the sample mean and sample covariance. For this reason, for MCAR data and especially for large n, if computational resources are an issue, it would seem advisable to only use the EM algorithm for covariance estimation and large percentages of missing data. If efficiency is not a matter of concern, however, the EM algorithm is clearly more accurate for both applications.

Transitioning to the MAR mechanism, the EM algorithm's struggle is more evident, due to the nature of missing data. The RMSE climbs steadily as the percentage of missing data rises. This linear increase hints at

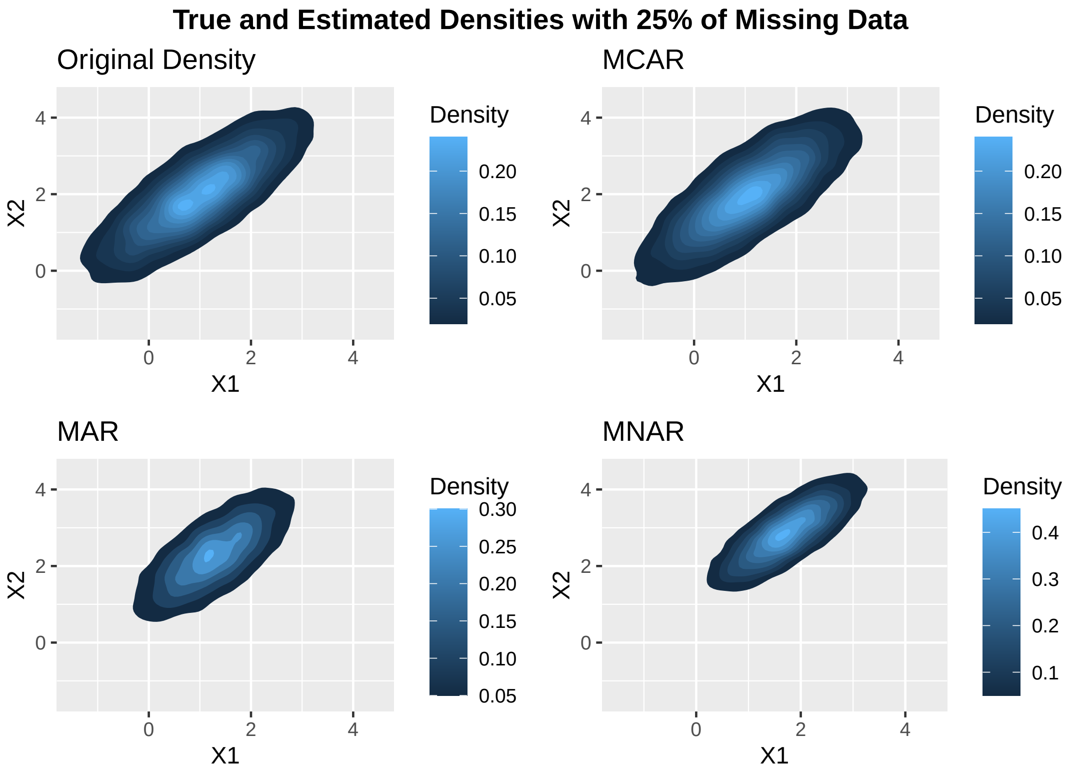

Figure 2: Original and estimated densities with EM applied on incomplete datasets generated by the three mechanisms with 25% of missing data.

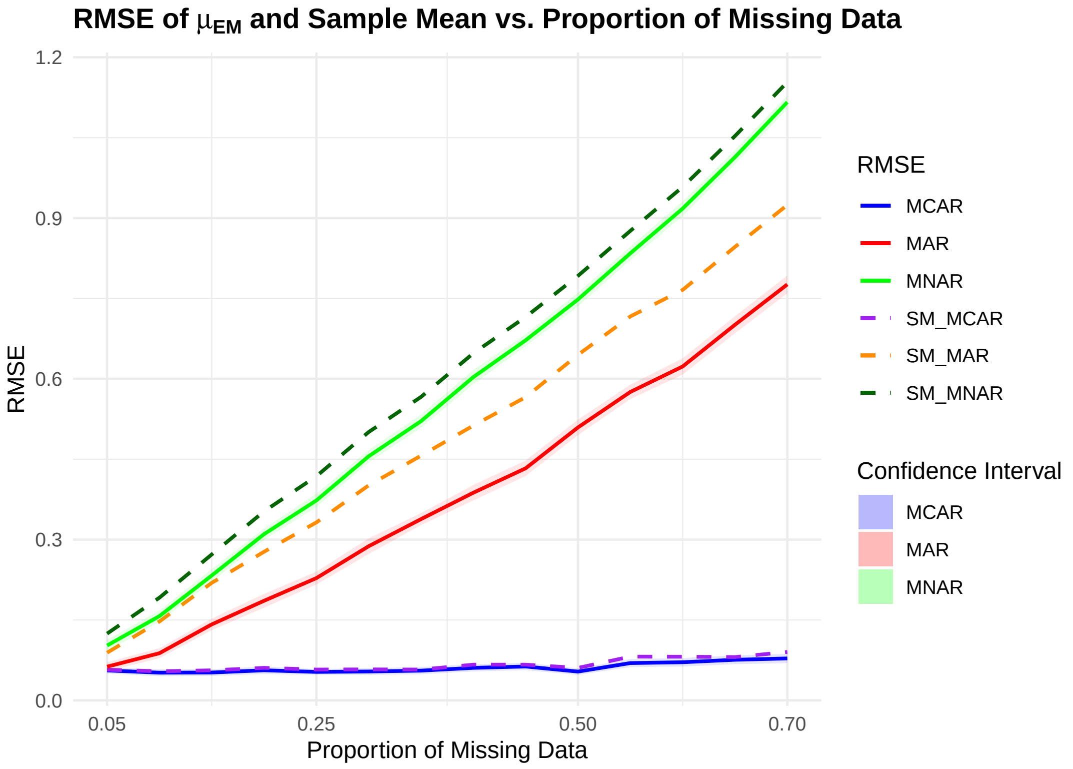

Figure 3: RMSE of μ_{EM} (solid) and sample mean (dashed) for the three different mechanisms as the percentage of missing data grows, with 95% confidence intervals.

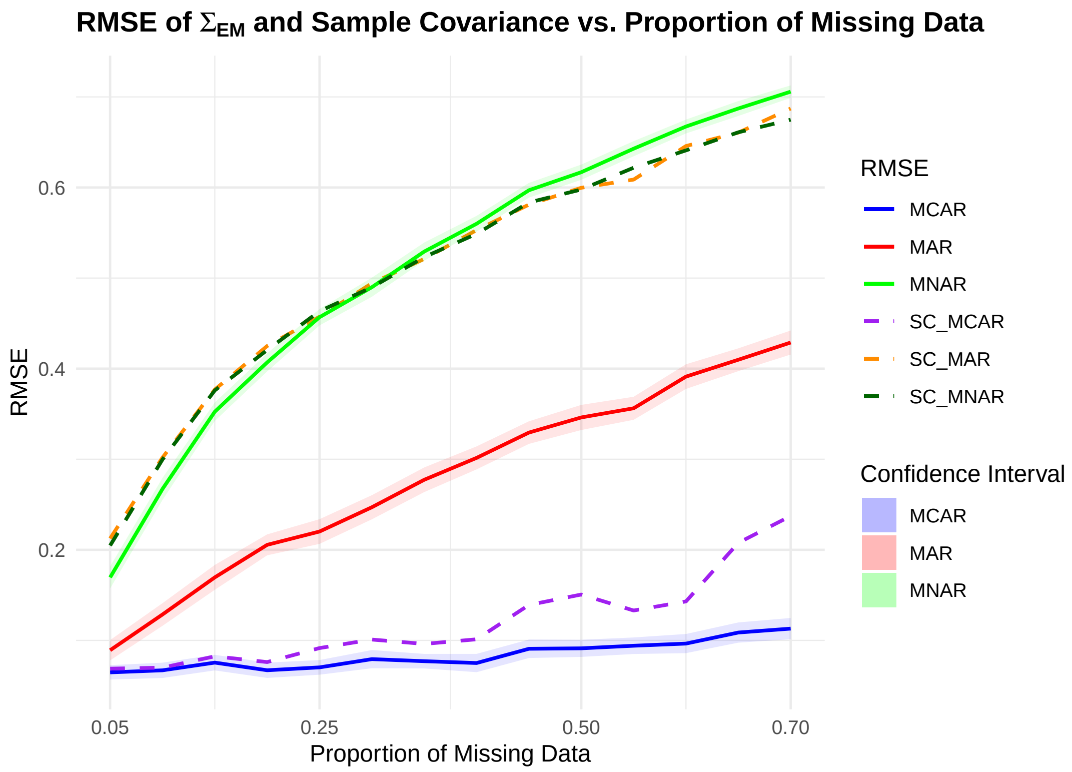

Figure 4: RMSE of Σ_{EM} (solid) and sample covariance (dashed) for the three different mechanisms as the percentage of missing data grows, with 95% confidence intervals.

a consistent degradation in the quality of the EM estimation as the missing data is related to other observed variables. However, the EM algorithm still does much better than the simpler approaches outlined above: the MAR setting is where the EM algorithm most clearly surpasses simply taking the sample mean and covariance. Furthermore, it is interesting to note that when taking the sample covariance after removing incomplete rows, the trend of the error is equivalent to that of the MNAR case. This is because in this experiment, we only have two columns that condition on each other to generate NA's. Then, eliminating the rows with any of the two columns being empty gives the same result as MNAR data, due to the definition of the functions in the `missMethods` package as well.

The graph produced using the MNAR mechanism underscores a more severe condition. From the onset, the RMSE for both mean and covariance estimation is elevated and ascends rapidly. This trend illustrates the algorithm's heightened sensitivity when the missingness is related to the unobserved data itself. As the proportion of missing information increases, the RMSE grows substantially, clearly exceeding the errors observed in the other two mechanisms. On the other hand, the error obtained by taking the sample mean and covariance as estimates is not that much higher for low percentages of missing data, and, for covariance estimation, it is even lower for proportions > 0.3. This would suggest that, even in the MNAR case, one could also consider sacrificing some accuracy for lower percentages of missing data, in favor of efficiency. For higher percentages, and for covariance estimation in particular, the experiment seems to show that one should avoid the EM algorithm, as it cannot even outperform the other much simpler approach we considered.

When comparing the results, it is clear that the missing data mechanism significantly influences the accuracy of the estimates obtained with EM. The MAR and MNAR conditions, in particular, expose the limitations of the EM algorithm, where the RMSE suggests a more pronounced inaccuracy in the estimation process.

On the other hand, the EM algorithm shows resilience under MCAR conditions, even for high percentages of incompleteness. This analysis showcases the critical need to understand and correctly identify the missing data mechanism in statistical modeling to ensure the reliability of the estimates produced.

2.1.1 A More in Depth Look at how Σ Affects EM Performance in Mean Estimation with MAR Data

Limiting our analysis to two variables, however, could prevent us from discovering interesting patterns that only arise when considering the dynamics at play between multiple columns with incomplete information. Thus, we generate multivariate random data with $\mu = (0, 0, 0, 0)^T$ and $\Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.5 & 0.9 \\ 0 & 0.5 & 1 & 0.7 \\ 0 & 0.9 & 0.7 & 1 \end{bmatrix}$. Note that the first variable is independent of all the others. Then, we generate MAR data in column 1 conditioned on column 2 (covariance 0), on column 2 conditioned on column 3 (covariance 0.5), on column 3 conditioned on column 4 (covariance 0.7) and on column 4 conditioned on column 2 (covariance 0.9). We are interested to see how the error of each coordinate of μ_{EM} will behave, in particular when compared to the more preferable setting of MCAR data.

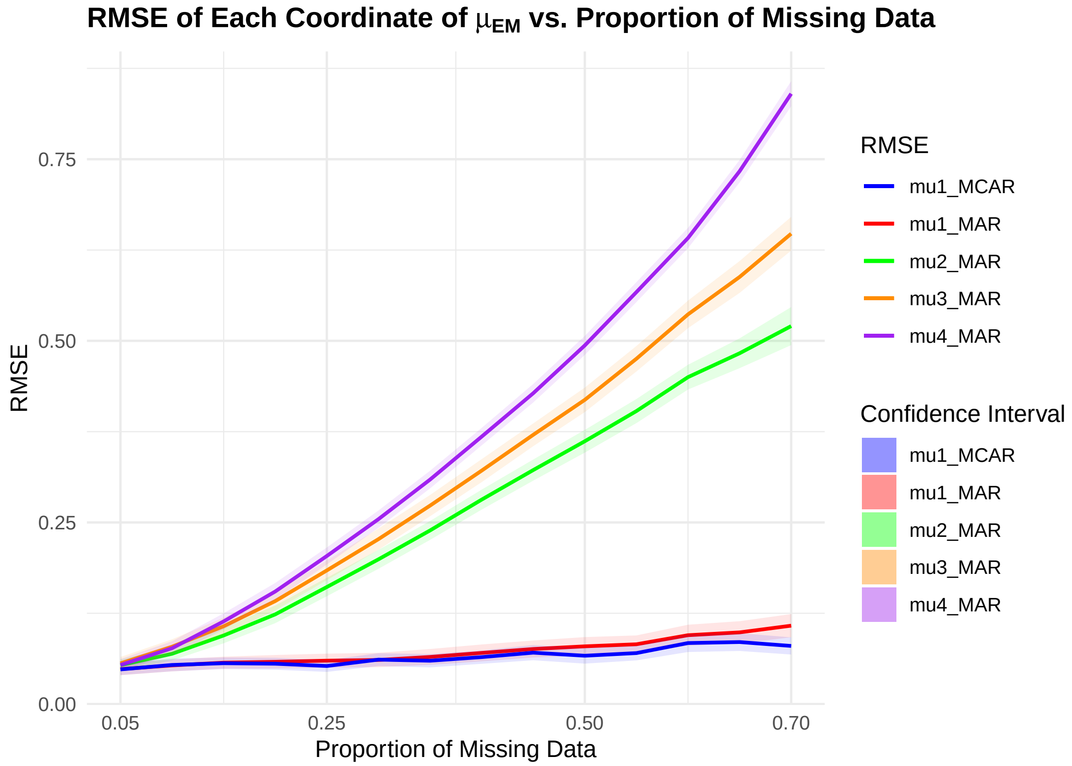

Figure 5: RMSE of $\mu_i,\ i = 1, \ldots, 4$ for MAR data and of μ_1 for MCAR data with 95% confidence intervals.

From Figure 5, we can see that the performance of the EM algorithm in this setting heavily depends on the correlation between the variables used to generate MAR data, especially for percentages of missing data higher than 20%. The pattern is quite clear: the higher the correlation between the variables when generating MAR data, the higher the error. The difference is more and more evident as the percentage of missing data increases. On the other hand, when we have a column with MAR data, that is, however, uncorrelated with all the others, the estimation of its mean through the EM algorithm is equivalent to that of the same method applied to MCAR data. From these observations, we can deduce that when performing mean estimation with the EM algorithm, one should think very carefully about the correlations between variables, especially when faced with MAR data.

2.2 EM for Linear Regression

Let us now consider the following linear regression framework:

$$Y = \hat{X}\beta + \epsilon,$$

$\hat{X} = (1, X)$ and observations drawn from a multivariate normal distribution with $\mu = (-2, -1, 0, 1)^T$ and $\Sigma = \begin{bmatrix} 1 & 0.4 & 0.5 & 0.2 \\ 0.4 & 2 & 0.6 & 0.4 \\ 0.5 & 0.6 & 3 & 0.9 \\ 0.2 & 0.4 & 0.9 & 4 \end{bmatrix}$, the noise is $\epsilon \sim \mathcal{N}(0, 1)$, and $\beta = (2, 3, -1, 4, 3)$. Then, we introduce missing data in the design matrix X and use the EM algorithm to find estimates β_{EM}, measuring their accuracy with the RMSE. This investigation offers valuable insights into the algorithm's effectiveness in handling missing data within the context of linear regression modeling.

The plot below displays the RMSE of linear regression estimates β_{EM} across various levels of missing data for the three mechanisms.

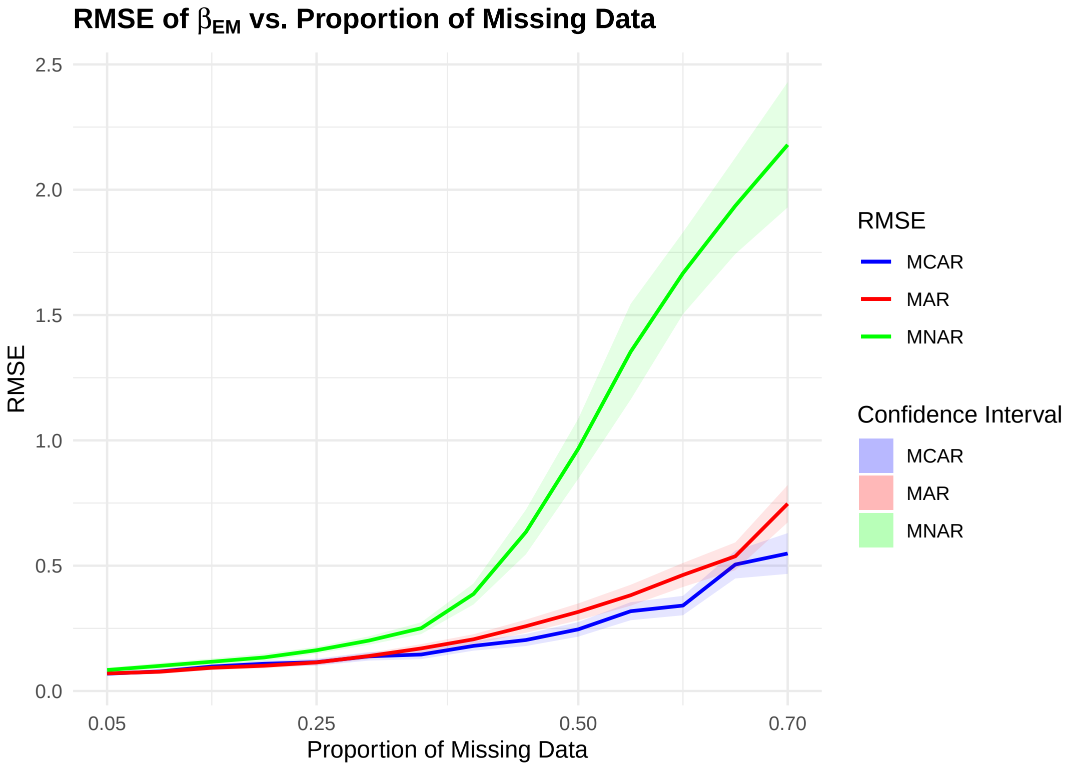

Figure 6: RMSE of β_{EM} for the three different mechanisms as the percentage of missing data grows, with 95% confidence intervals for linear regression.

Figure 6 clearly delineates two distinct error trends for different missing data mechanisms. The first trend concerns the RMSE for both MCAR and MAR mechanisms, which follow a similar pattern over the range of missing data proportions. Moreover, the error associated with MCAR consistently remains only marginally lower (unlike for mean and covariance estimation) than that for MAR. The errors associated with the MNAR mechanism follow a different trend, escalating more sharply, especially after the 25% missing data threshold. This marked hike suggests EM's greater sensitivity to the proportion of missing data in the MNAR case when compared to MCAR and MAR.

2.2.1 A More in Depth Look at how Σ Affects EM Performance in Linear Regression with MAR Data

We now generate multivariate random data with $\mu = (0,0,0,0)^T$ and $\Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.5 & 0.7 \\ 0 & 0.5 & 1 & 0.9 \\ 0 & 0.7 & 0.9 & 1 \end{bmatrix}$. Note that the first variable is independent of all the others. Then, we generate MAR data in column 1 conditioned on column 2 (covariance 0), on column 2 conditioned on column 3 (covariance 0.5), on column 3 conditioned on column 4 (covariance 0.7) and on column 4 conditioned on column 2 (covariance 0.9). We are interested to see how the error of each coordinate of β_{EM} will behave, in particular when compared to the more preferable setting of MCAR data. Will we see the same phenomenon as in Section 2.1.1, or will the more complex (when compared to mean estimation) nature of linear regression reveal different patterns?

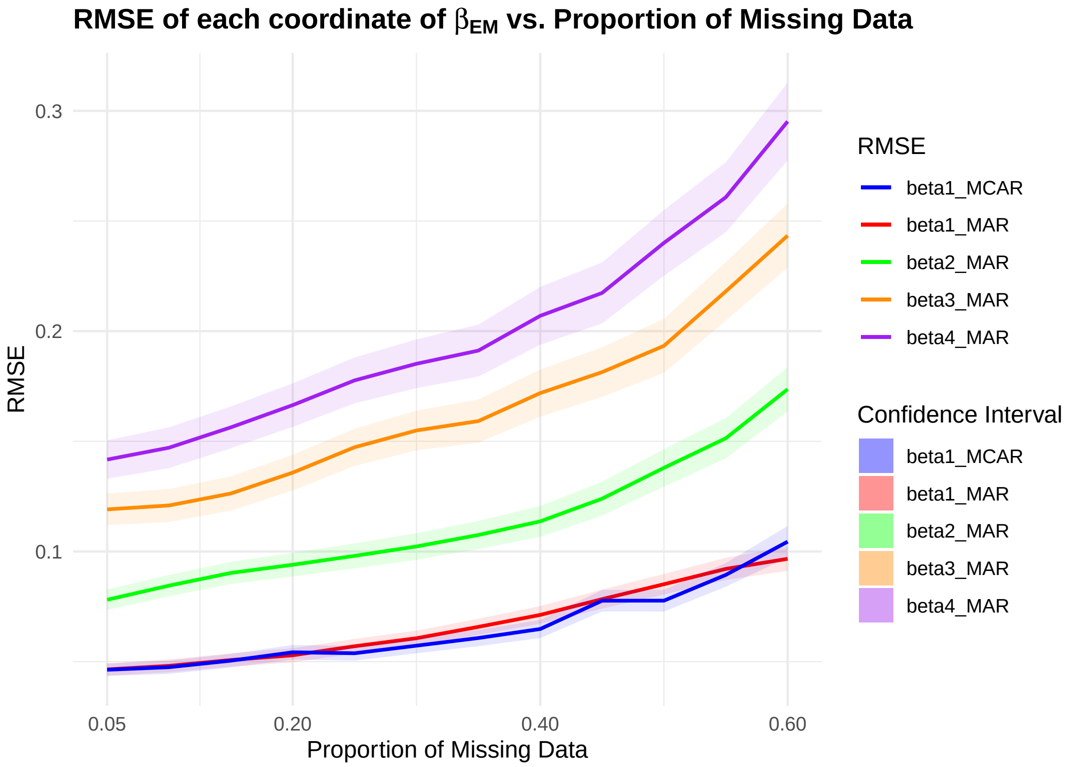

Figure 7: RMSE of $\beta_i,\ i = 1, \dots, 4$ for MAR data and of β_1 for MCAR data in the setting of linear regression, with 95% confidence intervals.

Figure 7 reveals a different behavior than that found in Section 2.1.1. While the error for β_1 evolves similarly for the MAR (uncorrelated column) and MCAR scenarios, the other errors are not in order of correlation between the target and control columns when producing NA's (if they were, the errors associated with β_2 and β_3 would be switched). Rather, the errors, already from low percentages of missing data, are ordered by the sum of the covariances of the target column with the other variables. To clarify, columns $1-4$ are ordered by the sum of the covariances with the other variables and this causes the errors associated with

$\beta_1, \ldots, \beta_4$ to be ordered in the same way. This experiment, again, suggests that when using EM on MAR data it is crucial to carefully think about the correlations between the variables, as these heavily affect the performance of the algorithm.

Note: for this last experiment, we introduced up to 60% of missing data for visualization purposes, since the results for higher percentages were characterized by high variance.

2.3 EM for Logistic Regression

We now dive deep into another application of the EM algorithm in the context of missing data: logistic regression.

Let (y_i, X_i) be n i.i.d. observation with $y_i \in \{0,\ 1\}$ binary response. As in the context of linear regression, we generate our data X from a multivariate normal distribution. If we also define the unknown parameters $\beta = (\beta_0, ..., \beta_p)^T$, for notation purposes, we can group the various quantities of the model as follows $\theta := (\mu, \Sigma, \beta)$.

Recall that the logistic regression model for binary classification can be written as:

$$\mathbb{P}(y_i = 1 | X_i, \beta) = \frac{exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right)}{1 + exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right)},\ i = 1, ..., n.$$

Analogously to the linear regression framework, our goal is to use the EM algorithm to estimate β when there are missing values in the design matrix X.

Unlike in the case of linear regression, for logistic regression there is not a closed-form expression for the expectation in the E-step of the EM algorithm. Therefore, we are going to use a Monte Carlo version of EM, first proposed in [5]: to calculate the above-mentioned expectation, a large number of samples of missing data from $p(X_{MIS}|X_{OBS}, y; \theta)$ are generated and the expectation is then replaced by the empirical mean. An accurate estimation of this expectation requires a significant computational effort. Thus, to reduce such complexity, a Stochastic Approximation EM (SAEM) [6] is often used instead.

SAEM replaces the E-step by a stochastic approximation based on a single simulation of X_{MIS} and the t^{th} iteration consists of the following three steps (starting point $\theta^{(0)}$):

- For $i = 1, ..., n$ draw uniformly at random a single sample $X_{MIS}^{(t)}$ from the conditional distribution of missing variables $p(X_{MIS}|X_{OBS}, y; \theta^{(t-1)})$.
- Update the function Q: $Q(\theta, \theta^{(t)}) = Q(\theta, \theta^{(t-1)}) + \gamma_t \left(l(\theta; X_{OBS}, X_{MIS}^{(t)}, y) - Q(\theta, \theta^{(t-1)}) \right)$.

  Where $l(\theta; X, y)$ is the log-likelihood for the complete data.
- Maximization step: update the estimation of θ.

$$\theta^{(t+1)} = argmax_{\theta} Q(\theta, \theta^{(t)}).$$

This methodology is implemented in the `R` package `misaem`. In the paragraphs below, we are going to illustrate some examples and report experimental results produced by the application of the algorithm for logistic regression.

As suggested in [7], we generate a design matrix $X \in \mathbb{R}^{n \times p}$ with $n = 300$ (observations) and $p = 2$ (covariates) drawing each observation as we mentioned before from a multivariate normal distribution $\mathcal{N}_p(\mu, \Sigma)$ with:

$$\mu = (1, 2)^T, \Sigma = diag(\sigma) C diag(\sigma),$$

where $\sigma = (1, 2)^T$ and the correlation matrix C is:

$$C = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

For the logistic model, we use $\beta = (0, 1, -1)$. Below, we compute β_{EM} for the three different mechanisms and percentages of missing data ranging from 5 to 70%, compute the RMSE for each estimate, and plot the results. Note that this method is quite expensive in terms of computing resources. Thus, to avoid running the code for an excessive amount of time, we had to limit ourselves to averaging over only 10 experiments: this caused the error to be a bit volatile. Still, we can advance some statements about the performances of EM in this scenario by analyzing the plots below. For brevity, throughout the rest of the project, in the contest of logistic regression we will refer to the SAEM algorithm just by EM.

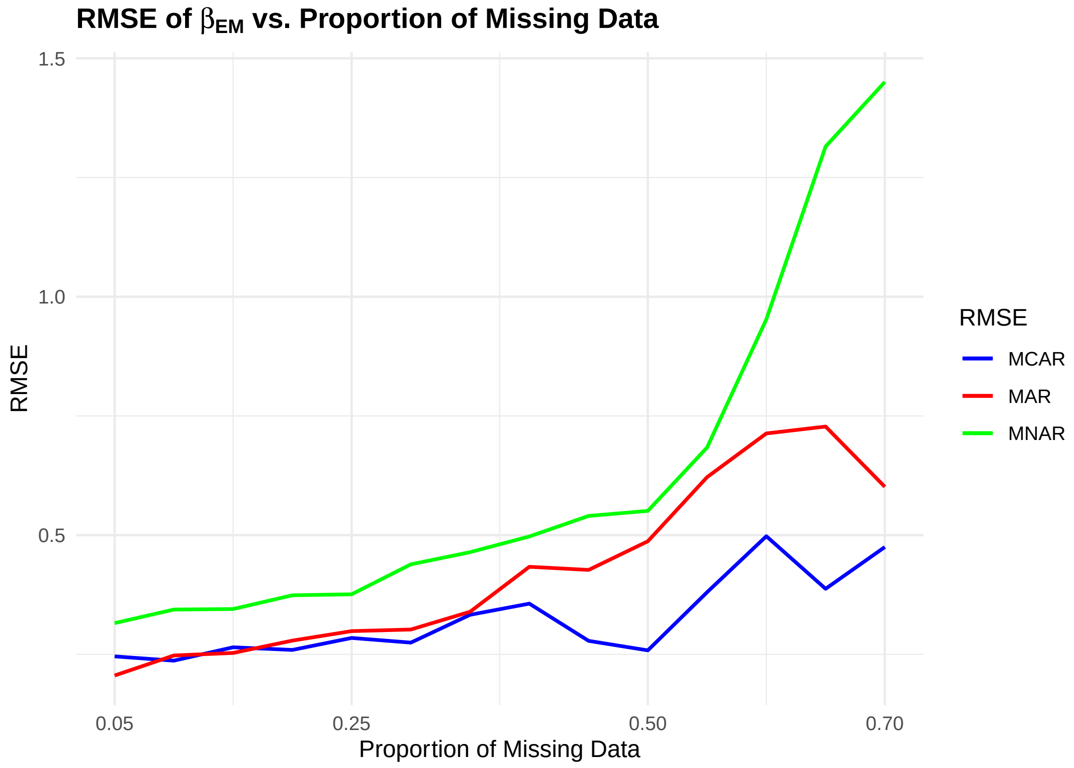

Figure 8: RMSE of β_{EM} for the three different mechanisms as the percentage of missing data grows for logistic regression.

Similarly to the linear regression setting, Figure 8 reveals a distinct pattern for MCAR and MAR, and a separate one for MNAR. For MCAR and MAR, we can observe a slight increase on average, of the RMSE of β as the percentage of missing data increases. For the MAR mechanism, this increase is slightly more pronounced. Unlike mean and covariance estimation, the MCAR case is not exempt from an error increment for higher percentages. For MNAR, as in the experiments above, the RMSE of β increases the quickest as the missing percentage in the data increases. In terms of order of magnitude, from the above figure, it is clear that the setting in which the EM algorithm performs the worst is the MNAR setting. This is also observed in the linear regression problem studied above. Again, as expected, the EM algorithm seems not to be able to get reliable estimates when there are more complex underlying mechanisms under which missing data are generated.

In conclusion, we can argue that even in this case the EM algorithm is a good choice in the MCAR setting and a decent one even for the MAR setting, specifically when the missing percentage is not very high (<0.4). On the other hand, we need to be more careful when the missing data are generated not at random. In this case, especially for larger proportions of missing data, the EM algorithm seems to be not a great choice to perform logistic regression.

3 Comparison of EM Estimates Against Imputation + Maximum Likelihood Estimation

In this section, we compare the results obtained with EM applied to both the linear and logistic regression tasks against those attained by performing data imputation followed by standard maximum likelihood estimation.

3.1 Imputation Methods

We consider four different imputation methods [8] readily available in the following `R` packages.

softImpute: The `softImpute` package fits a low-rank matrix approximation to a matrix with missing values via nuclear-norm regularization. `softImpute` offers two different variants to compute such approximation. One iteratively computes the soft-thresholded SVD of a filled in matrix - an algorithm described in [9]. This is option `type="svd"` in the call to `softImpute()`. The other uses alternating ridge regression, at each stage filling in the missing entries with the latest estimates. This we believe is the faster option, and is option `type="als"` in the call to `softImpute()` . We computed the best value of the nuclear-norm regularization parameter λ using cross-validation and then we used this λ_{CV} when calling `softImpute()` .

mice: The mice package implements a multiple imputation methods for multivariate missing data. The mice function computes, based on an incomplete dataset, multiple imputations by chained equations and thus returns m *(*= 5 in our experiments, the default value*)* imputations, of which we take the mean to generate the final dataset.

missForest: The missForest function imputes missing values iteratively by training random forests on the observed values to predict the missing ones.

missMDA: The impute PCA function imputes missing values applying principal component methods. The missing values are predicted using the iterative PCA algorithm for a predefined number of dimensions (see [10] for further details).

3.2 Experiments

We now design some experiments to better understand in which settings it is worth using the EM algorithm and in which ones it is instead preferable to opt for data imputation followed by standard maximum likelihood estimation.

3.2.1 Linear Regression

MCAR. First, we perform the experiment for linear regression on datasets generated with the MCAR mechanism. The complete dataset, in this case as well as in the MAR and MNAR cases, was generated exactly as in Section 2.2.

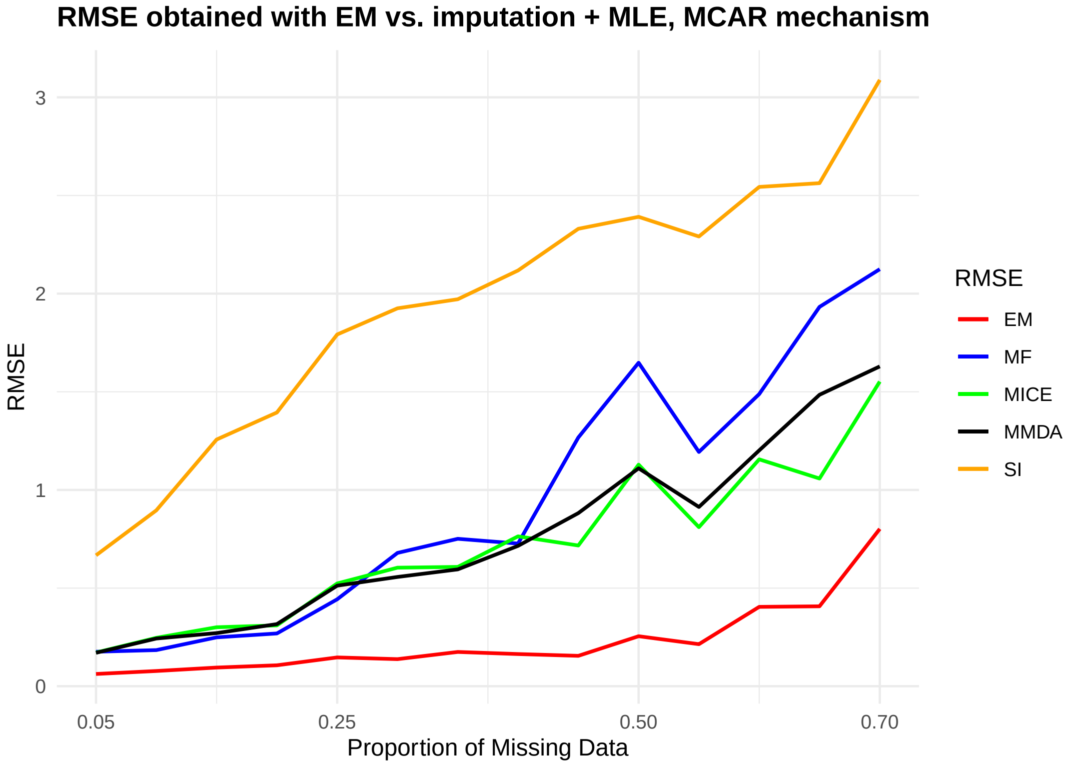

Figure 9: EM vs imputation + MLE for linear regression on MCAR data.

From the plot in Figure 9, we can observe that the method that obtains a lower RMSE for the entire range of the missing percentages is the EM algorithm. We can also notice that this difference is quite marked when increasing the percentage of missing data. Across the imputation methods, the `SoftImpute`method is the one that performs worst; `MDA`, `MissForest`, `MICE` have similar performances, expecially for low percentages of missingness.

MAR. Now we perform the experiment for datasets generated with the MAR mechanism.

As we can see in Figure 10, for the MAR mechanism, the results are very similar to those obtained in MCAR setting. In particular, the EM algorithm is again the best method. Between the four imputation packages, the worst-performing one for most percentages is `softImpute`, with the other three mostly showing very similar trends.

MNAR. Now we perform the same experiment for datasets generated with the MNAR mechanism.

Figure 11 shows that, even in the case of the MNAR mechanism, the method that performs better in terms of RMSE is again the EM algorithm. For percentages of missingness up to around 40%, `softImpute` is again the package performing the worst. Later on, the errors associated with the MLE estimates obtained after imputation methods plateau, apart from `missMDA`, for which the error keeps increasing until the 70% mark. Note that, despite EM being better than the rest, it still suffers from a higher error than in the MCAR and MAR cases, especially for higher percentages.

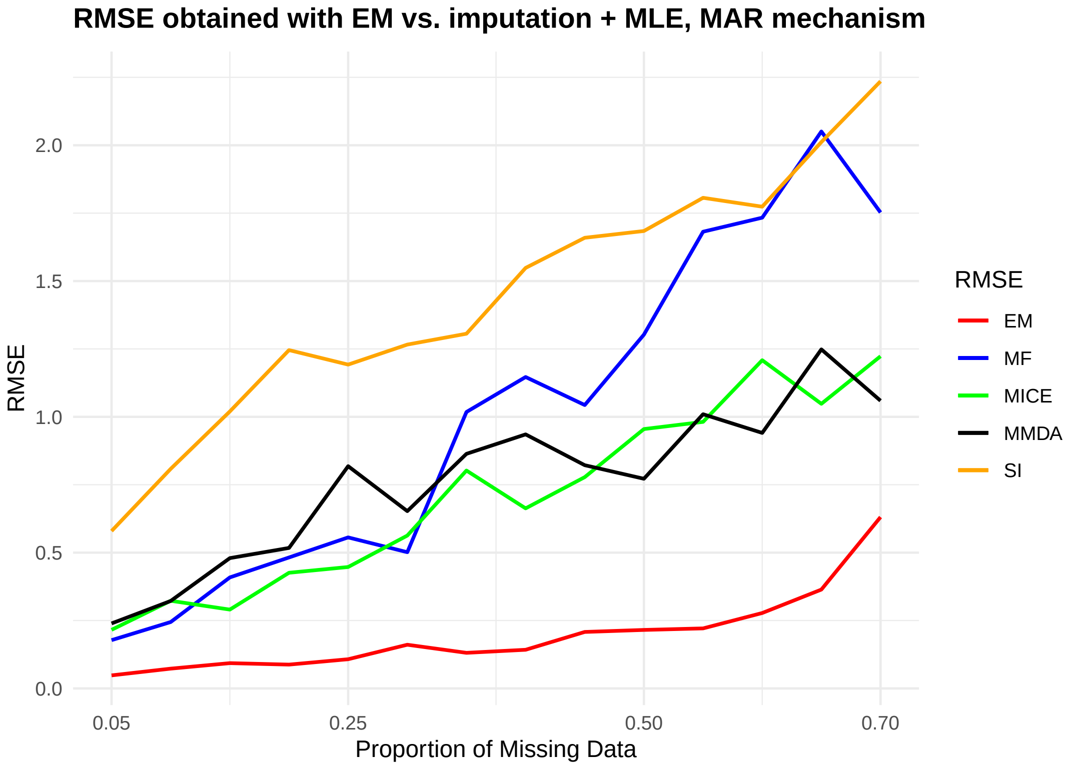

Figure 10: EM vs imputation + MLE for linear regression on MAR data.

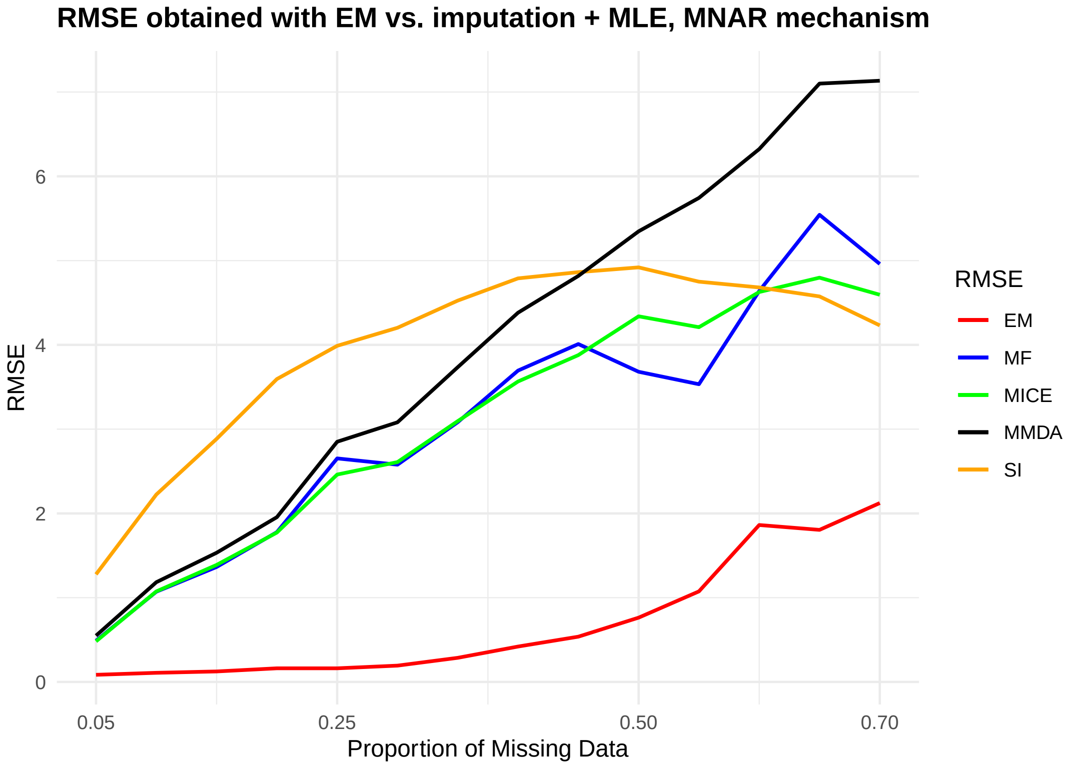

Figure 11: EM vs imputation + MLE for linear regression on MNAR data.

Final observations. From the experimental results presented above, it is clear that in the linear regression setting with missing data, using the EM algorithm to estimate β leads to better estimates compared to the those obtained with MLE after imputation, at least using the four packages we tried above. As we observed, in fact, for all the three mechanisms that generate NA's, EM outperforms the other approaches. Hence, we can argue that this is the best method to use for such a problem independently from the nature of the missingness in our data.

3.2.2 Logistic Regression

Then, we proposed the same analysis for the logistic regression setting. We will see that the results will be quite different from the ones obtained in the context of linear regression. For these last experiments, we limit ourselves to $5\% - 40\%$ of missing data to reduce the computational burden of this section. As we will see below, we can still obtain meaningful results in this range.

MCAR. We now perform the experiment for logistic regression on a dataset generated with the MCAR mechanism. Here and for the next two experiments, the $n = 150$ observations (we went down from 300 to 150 to cut computational costs) are generated from a multivariate normal distribution with $\mu = (1, 2, 3, 4)^T, \Sigma =$
$$\begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix}$$

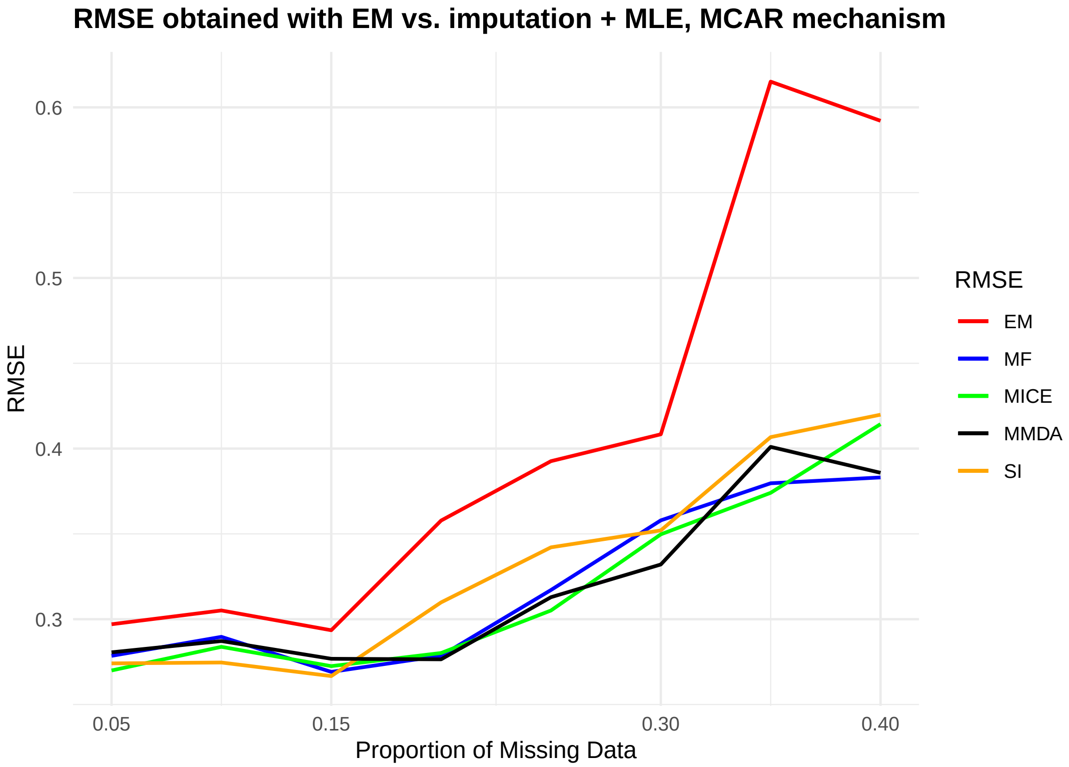

Figure 12: EM vs imputation + MLE for logistic regression on MCAR data.

Figure 12 shows that the EM algorithm is the worst method to use in this setting, since all four imputation methods we tried performed better (on average) for all percentages. The four imputation methods gave similar results in terms of RMSE. We now proceed with the same experiments with MAR and MNAR data to see whether the same behavior holds, and later make some final remarks about the logistic regression setting.

MAR.

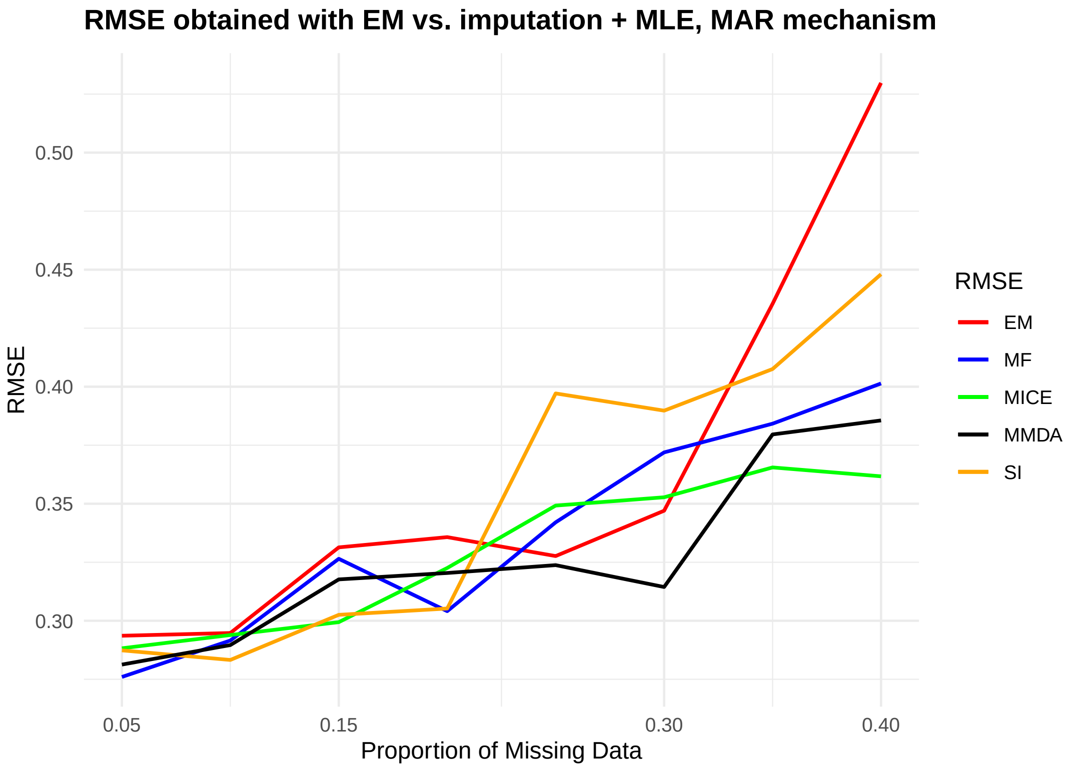

Figure 13: EM vs imputation + MLE for logistic regression on MAR data.

From Figure 13, we can observe that in this case, for percentages up to 30%, the performance of the EM algorithm (on average) is comparable with that of MLE after imputation methods, but never better. For higher percentages $(35\%, 40\%)$, we get instead a much worse RMSE.

MNAR.

Figure 14 indicates that the performance of EM algorithm is again similar to that of imputation + MLE, apart from when imputation is performed with `softImpute`, in which case we get the best results by a margin.

Final observations. From the experimental results presented above, it is clear that in the logistic regression setting with missing data, using the stochastic approximation of the EM algorithm is not the ideal choice. In fact, independently of the mechanism and percentages that generate missingness, EM is never the best choice in terms of RMSE obtained. This difference from the linear regression case could be due to the fact that, as mentioned in Section 2.3, in this case we don't have a closed form solution for the expectation step and thus we resort to use a Monte Carlo version of EM. This algorithm is also quite expensive, and thus it seems not advisable to use it, at least when imputation options are available.

3.2.3 Comparison and observations

The results obtained in this section are interesting. We observed that the performances of EM compared to the one of maximum likelihood estimation after imputation is more task-dependent than mechanism-

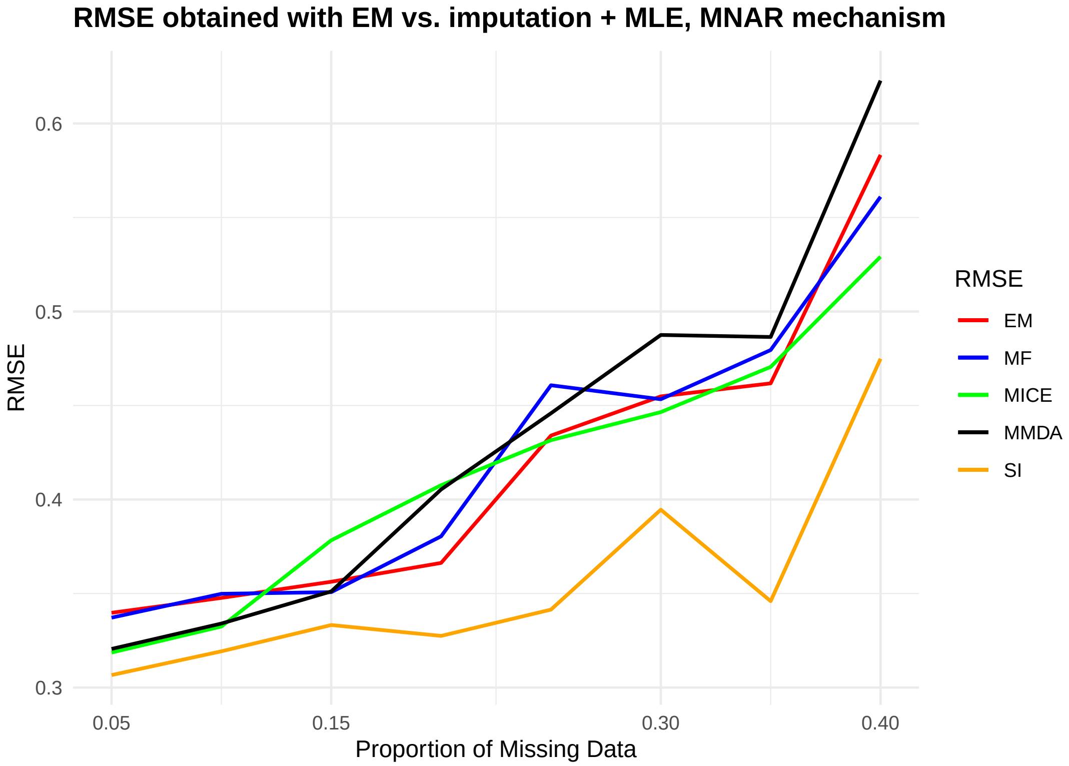

Figure 14: EM vs imputation + MLE for logistic regression on MNAR data.

dependent. In the previous two subsections, in fact, we observed that EM was the best method for linear regression but one of the worst one for logistic regression.

4 Conclusion

In this project, we performed several numerical experiments to investigate how the Expectation-Maximization algorithm behaves in various contexts with missing data. The incomplete datasets were generated using three different mechanisms - MCAR, MAR, and MNAR - and varying percentages of missing data.

In Section 2 and Section 3 we tackled the following question: *When you are faced with missing data, when is the EM algorithm a good option for statistical inference (the estimation process)?*

In Section 2, we studied this question in three different frameworks: mean and covariance estimation of multivariate normal distribution, linear regression, and logistic regression. Through the experiments above, we observed that the EM algorithm's performance varies widely depending on the mechanism through which the missing data is generated. In particular, we argue that the EM algorithm is quite effective in the case MCAR data, independently of the framework we are working in. In fact, in this case the RMSE stays relatively low for all the tasks we tried and even for high percentage of missing data. The performance is a bit worse for the MAR setting, especially for large proportion of missingness. Finally, the MNAR mechanism is the one that hurts the estimation process the most, with the RMSE being way higher than the other two settings. Again, it is crucial to consider that the different ways of generating MAR and MNAR data are endless. What we showed is that if data is MAR and MNAR, performance can be heavily affected. We do not claim, on the other hand, that it would be affected in the same way for all MAR and MNAR scenarios.

As we showed in Section 2.1.1, in fact, performance in the case of MAR data can be similar to MCAR when the correlations between variables is low.

Another interesting question that arose from this study of the EM algorithm in the context of missing data is whether this algorithm outperforms usual imputation methods combined with maximum likelihood estimation. We observed that the algorithm outperforms maximum likelihood estimation after imputation in terms of RMSE in the linear regression problem. On the other hand, when dealing with logistic regression, the SAEM algorithm seems to be the worst choice you can make among the ones analyzed. Thus, we argue that there is no method that gets the lowest error for all tasks in which missing data is present. Instead, a case-by-case analysis needs to be done, and the result can change depending on the problem.

Summarizing, we observed that the EM algorithm is in general a good choice when dealing with missing values in a design matrix X, but the error obtained by its estimates depends on the missing data generating process we are dealing with. We also showed that EM is not always preferable to imputation followed by standard maximum likelihood estimation. Further directions of research could be to try the EM algorithm on incomplete datasets generated with different MAR and MNAR mechanisms and / or apply the EM algorithm to different frameworks.

References

[1] P. L. Mhalla, “EM algorithm.” https://math-517.github.io/math_517_website/notes/week_06.html, 2023.

[2] D. B. Rubin, “Inference and missing data,” *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976, Accessed: Dec. 17, 2023. [Online]. Available: http://www.jstor.org/stable/2335739

[3] T. Rockel, “Package ‘missMethods’.” https://cran.r-project.org/web/packages/missMethods/missMethods.pdf, 2022.

[4] I. M. Teresa Alves de Sousa, “How to generate missing values?” https://rmisstastic.netlify.app/how-to/generate/misssimul#2_Use_of_produce_NA_with_default_settings, 2021.

[5] J. G. Ibrahim, M.-H. Chen, and S. R. Lipsitz, “Monte Carlo EM for Missing Covariates in Parametric Regression Models,” *Biometrics*, vol. 55, no. 2, pp. 591–596, 1999, Available: https://ideas.repec.org/a/bla/biomet/v55y1999i2p591-596.html

[6] M. Lavielle, *Mixed Effects Models for the Population Approach: Models, Tasks, Methods and Tools*. Chapman and Hall/CRC, 2014. Available: https://hal.science/hal-01122873

[7] W. Jiang, “Linear regression and logistic regression with missing covariates.” https://cran.r-project.org/web/packages/misaem/vignettes/misaem.html, 2021.

[8] A. S. Genevieve Robin Imke Mayer, “How to impute missing values?” https://rmisstastic.netlify.app/how-to/impute/missimp, 2021.

[9] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral regularization algorithms for learning large incomplete matrices,” *Journal of Machine Learning Research*, vol. 11, no. 80, pp. 2287–2322, 2010, Available: http://jmlr.org/papers/v11/mazumder10a.html

[10] J. Josse and F. Husson, “missMDA: A package for handling missing values in multivariate data analysis,” *Journal of Statistical Software*, vol. 70, no. 1, pp. 1–31, 2016, doi: 10.18637/jss.v070.i01.