Radboud University Nijmegen

and

Jeroen Bosch Hospital

# Handling Missing Data in Electronic Health Record Laboratory Data:

## Evaluation and Comparison of Feasible Methods

Thesis MSc Artificial Intelligence : Cognitive Computing

Faculty of Social Sciences


*Author:*
K. van de Poll
s4695194

*Supervisors:*
Prof. dr. E. de Vries MD PhD, JBZ
D.H. van Dalen RN MSc, JBZ
Dr. J.H.P Kwisthout, Radboud University

*Second reader:*
M.L.P. Bueno, Radboud University


October 2023

# Contents

# Introduction

With the aging of the population, the prevalence of frailty is increasing [1]. There are different ways to define frailty but in the present context it is defined following Clegg et al.: `frailty is a state of vulnerability to poor resolution of homeostasis after a stressor event and is a consequence of cumulative decline in many physiological systems during a lifetime' [2]. Frailty is associated with adverse health outcomes [3] and increased healthcare costs [1]. For example, frail patients have an increased risk of postoperative complications and may require longer and more intensive medical care, which necessitates the consideration of alternative treatment paths [4]. The gold standard for assessing frailty is the Comprehensive Geriatric Assessment (CGA) [5]. However, this assessment is characterised by its multidimensional, holistic nature leading to substantial time requirements. These inherent challenges, coupled with the significant impact of frailty on healthcare outcomes has led to the development of more than 50 frailty screening tools for a variety of settings [6]. However, a large number of screening tools measure the same construct of frailty while the agreement between these screening tools has been shown to vary [7]. Thereby, reviews of frailty screening tools have highlighted the need for better reliability and validity testing [8]. Besides, most of these screening tools are labour and time intensive as well because they are based on surveys or questionnaires that require new information to be gathered. Automated assessment of the probability of being frail utilizing already existing routinely acquired data confers a notable benefit by obviating the need for additional time and effort [9]. Therefore, the demand for developing reliable and efficient methods for predicting frailty in a clinical setting using existing patient data is growing.

One of the ongoing research projects at the Jeroen Bosch Hospital (JBZ) is the development of an automated prediction model to estimate the probability of being frail in elderly patients on the basis of routine hospital Electronic Health Record (EHR) data. EHRs have the potential to serve as a valuable data source for the development of such clinical prediction models [10].

As EHR systems become more widely adopted, they generate increasingly large amounts of clinical data, prompting a growing interest among researchers, healthcare administrators, and clinicians in leveraging this data for secondary use. The ability to analyse EHR data has the potential to improve clinical knowledge and enhance patient care by providing valuable insights into healthcare delivery and patient outcomes. As such, the secondary use of EHR data has become a key focus for those seeking to optimise healthcare practices and improve patient outcomes [11]. EHRs provide an opportunity to create clinical prediction models that can assist healthcare professionals in making informed decisions regarding initiation or transitioning treatments for individualised patient care [12]. However, developing such a model with the use of EHRs is not without its challenges. EHR data can be categorised into two categories: structured and unstructured (textual) data [13]. Structured data is especially suitable for statistical analyses as it is stored in organised databases that allow for dependable data retrieval. EHR data often lack a standardised format and may contain errors introduced by human input. Moreover, the data of individual patients are recorded at irregular time intervals with varying frequencies. Also, missing data are frequently encountered in healthcare data, including the collected dataset of the ongoing frailty project of the JBZ. To ensure the accuracy and reliability of the prediction models built with EHR datasets, appropriate techniques must be implemented to address these missing data, as numerous machine learning (ML) algorithms assume the completeness of the dataset. When it comes to missing structured EHR data, it can result from a lack of collection or a lack of documentation [14].

The dataset in this study will be used to develop a prediction model for the probability of being frail. It contains multiple variables with diverse percentages of missingness; for this master thesis, only the laboratory data will be taken into account. The presence of missing data could reduce the accuracy and reliability of the prediction models, highlighting the need for strategies to handle these missing data [15]. Thereby, many machine learning techniques are not capable to deal with missing data, and therefore, this needs to be resolved beforehand. However, there is no universally accepted optimal solution to address the issue of missing data.

Approaches to address missing data require making assumptions about the underlying mechanism responsible for the missingness. The underlying mechanisms for incomplete data can be classified into three types: MCAR, MAR, and MNAR [16]. Understanding the differences between these mechanisms is crucial when handling missing data in research studies.

Missing Completely At Random (MCAR):
MCAR occurs when the missing data is completely unrelated to the observed and unobserved data. This means that the probability of missing data does not depend on the missing values themselves or on any other variables in the dataset. For example, a blood sample that is accidentally damaged in the laboratory. MCAR is considered the most favourable missing data mechanism since it does not introduce any bias in the analysis. The missing data can be easily ignored or imputed by various methods.

Missing At Random (MAR):
MAR occurs when the probability of missing data depends only on the observed data and not on the missing data itself. In other words, the missingness can be explained by observed data. This means that if the missing data were observed, it would not be different from the non-missing data. For example, an elderly person cannot attend a blood test due to illness. Predicting such an occurrence may be possible from other data available for the elderly person, but it would not be related to what would have been measured had they not been ill. MAR is a less favourable missing data mechanism than MCAR, but it is still considered manageable. MAR is considered to be plausible in many EHR datasets since there may be observable patterns or factors that influence whether or not a certain variable is missing, such as demographics or disease severity.

Missing Not At Random (MNAR):
MNAR occurs when the probability of missing data depends on the missing values themselves or on other variables not included in the dataset. In other words, the missingness is related to unobserved or unknown data. For example, an elderly person living alone may be less likely to answer a questionnaire about nutritional behaviour, because they know that their nutritional behaviour is not sufficient. In this case, the missing data is MNAR because it's not missing due to random chance. Instead, it is influenced by the patient’s living situation which is unobserved data. Therefore, the missing data is non-ignorable. Handling MNAR requires assumptions about the missing data mechanism, and the estimates obtained from the analysis may be biased. This means that the missing data is systematically different from the non-missing data and ignoring that can lead to biased estimates. MNAR is the most problematic missing data mechanism as it can introduce significant bias in the analysis.

Another highly probable mechanism in EHRs is informative missingness or I-MNAR, a special case of MNAR [17, 18]. I-MNAR occurs when the probability of missing data depends on the missing values themselves or on other variables included in the dataset. For example, the doctor’s decision to perform or to not perform specific blood tests on patients because they are (not) clinically indicated. In this case, the decision to perform the blood test is influenced by both unobserved data and observed data. The absence of a particular test result may be informative in itself and the reasons for the missing data should be carefully considered in the interpretation of the EHR data. The absence of laboratory test results can be indicative of various factors, such as disease severity, test availability, or clinician preferences.

In the context of this thesis lies the significance of Explainable Artificial Intelligence (XAI) [19]. The core theme of this master's thesis centres on the decision-making process concerning imputation methods for an EHR dataset. This dataset is instrumental in the development of a model for predicting the probability of being frail. In the context of XAI, the detailed and transparent explanation of the decision processes, even in the early stage of prediction model development, holds significant importance. Moreover, trust is the cornerstone of successful model adoption within the medical community, where healthcare professionals serve as its primary users. This trust hinges on healthcare professionals' ability to comprehend and interpret the model's underlying mechanisms. While exploring the complex data aspects may not be entirely comprehensible to them, the significance lies in the model's medical rationale being explainable, interpretable, and transparent. Consequently, this thesis aligns itself with the critical paradigm of XAI, wherein the aim is not to create a predictive black box, but rather a robust, interpretable, and transparent tool that empowers healthcare practitioners to confidently embrace and utilise it in their daily work.

This work differs from other research because it does not just find the 'best' imputation method using generated datasets with different missingness patterns. Instead, it seeks the most optimal solutions for handling the missing data in the EHR dataset, which will serve as the foundation for a predictive model aiming to predict the probability of being frail. Central to this approach is the recognition that different laboratory tests exhibit different distributions of test results as well as specific patterns of missing data. This acknowledgement set the need for tailored imputation techniques that align with these individual patterns, steering away from a one-size-fits-all solution. The thought processes and choices employed in this work are intended to shed light on the extensive level of deliberation and decision-making required before one can assert the 'best' imputation approach tailored to their specific objective. This work serves as an eye-opener, emphasizing the necessity of such thorough processes and underscores the importance of repeating this exploration for every new research goal that emerges.

# Theoretical comparisons

Several methods for addressing missing data have been proposed, however, identifying the most suitable approach depends heavily on the specific nature of the problem being addressed. In the following sections, various methods for handling missing data will be discussed and evaluated for their feasibility in completing an EHR dataset for prediction modelling purposes. A range of methods can be employed, including complete case analysis, imputation methods such as mean imputation, median imputation, Last Observation Carried Forward, regression imputation, and multiple imputation. The advantages and limitations of each method will be evaluated in the context of the laboratory data in the EHR dataset to determine the most effective approaches for handling the missing data.

## Complete case analysis

The most frequently used and simplest approach is to eliminate cases or individual variables that contain missing data, i.e., complete case analysis. This could be a great method when there is a small amount of missing data in a study. However, conducting complete case analysis is appropriate only when the missing data are MCAR, ensuring unbiased estimated are obtained [20, 21]. Then, if the dataset is sufficiently large, with statistical power not being a concern, complete case analysis can be a viable option.

In the present study, none of the patients are complete cases due to missing values in one or more laboratory tests. However, there are significant differences in the frequency of laboratory tests performed. The range of missingness is between 3.21% and 99.97%.

In our dataset, we have identified several laboratory tests with a missingness of more than 80%. Having more than 80% missingness for a variable in the dataset can pose a challenge for imputation methods, and may not be feasible in a reliable way, especially for MNAR data [22]. Imputation methods typically require a certain amount of available variable values to be successful, and when the missingness is too high, imputation can introduce bias and result in unreliable estimates. Given the limited amount of available variable values when missingness is more than 80%, imputation is not deemed feasible for such variables [23]. These variables were removed from the final dataset for further analysis. However, they have been assigned a missing indicator value, a concept that will be elucidated in a subsequent paragraph. This indicator is utilised because the presence of a result for a rarely used laboratory test is informative in itself. The presence of a measurement of a laboratory test with high missingness may serve as a potential indicator of frailty being present, as elaborated in the later section on the missing indicator method.

## Mean imputation

Standard mean imputation involves replacing each missing value with the mean value of the non-missing values in the same variable [24]. However, it is important to note that mean imputation can also be tailored by calculating the mean based on certain personal characteristics, such as age and gender. This personalised approach accounts for a more specific and individualised mean value, which may provide a more accurate representation of the missing data for certain cases. Although mean imputation is easy and fast to implement and preserves the sample size and structure of the data, it has two major limitations. First, it assumes that the missing values are MCAR, which may not be true in many EHR datasets. Second, mean imputation can lead to biased estimates of variance and covariance, and can distort the distributional properties of the data, especially if the variable being imputed has a skewed distribution [24]. As the number of missing values increases, the standard deviation becomes smaller after imputation compared to before the imputation [25].

The assumption of imputed data being accurate can lead to an overstatement of the statistical precision of a model. Besides, mean imputation does not add new information to the dataset, it only increases the sample size and leads to a bias in the standard error [26]. The over-representation of the mean value and the reduction in the variability of the data also results in an overblown representation of the correlation between variables [27].

Despite these limitations, mean imputation remains a popular method for handling missing data. The appropriateness of using mean imputation as a method for handling missing data in EHR datasets remains a matter of debate. Some researchers argue that mean imputation should not be used under any circumstances [28]. Conversely, others suggest that mean imputation can produce fairly accurate results when the proportion of missing data is less than 5% [21]. Other researchers suggests that the limitations of mean imputation can be negligible when the proportion of missing data is less than 10% and when there is low correlation among the variables [29]. As such, the decision to use mean imputation should be carefully considered in the context of the specific research question, dataset characteristics, and limitations of the method.

When building a prediction model for the probability of being frail based on EHR data, the assumption of the data being missing completely at random cannot be met. Besides, the distributions of many of the laboratory tests are skewed and this could distort the distributional properties of the data. Imputing missing values with the mean can distort the true distribution of the variable, resulting in biased estimates of the parameters and incorrect conclusions. This could potentially be resolved by using median imputation, as elaborated in the subsequent paragraph. However, when a laboratory test value is missing, it is likely that there was no medical indication to perform that particular test. Consequently, the likelihood of the missing value being an outlier is small, while the probability of it being close to the mean or median is higher. This is particularly true if there are no indications suggesting the missing value to be a potential outlier based on correlations with other laboratory tests. Hence, the application of mean imputation in our dataset could prove beneficial when dealing with missing laboratory test values that exhibit a non-skewed distribution and a low percentage of missingness.

## Median imputation

Standard median imputation involves replacing each missing value with the median value of the non-missing values in the same variable [23]. Similar to mean imputation, for median imputation, you can also calculate the median based on different personal characteristics. This will also account for a more specific and individualised median value. Median imputation was developed to address imputation of non-normally distributed data. However, it shares similar limitations with mean imputation. The main advantage of median imputation over mean imputation is that it is less sensitive to outliers. Since the median is a robust measure of central tendency, it is not influenced by extreme values in the same way the mean is. Median imputation is more useful compared to mean imputation when the variable has a (highly) skewed distribution (in case of a normal distribution, there will be no relevant difference between mean and median imputation). However, like mean imputation, median imputation does not incorporate any additional information in the dataset, and the variability of the imputed data is underestimated, which can lead to biased parameter estimates and standard errors.

The majority of the laboratory tests in our dataset exhibit a (highly) skewed distribution. The application of median imputation in our dataset can prove beneficial when dealing with missing laboratory test values that exhibit a skewed distribution and a low percentage of missingness.

## Last Observation Carried Forward

Last Observation Carried Forward (LOCF) is a commonly used imputation method in longitudinal clinical studies to handle missing data. This method involves carrying forward the last observed value from the same patient for a variable in the case of missing data [26]. However, the main limitation of LOCF is that it assumes that the value of the outcome remains constant despite missing data, which is often an unrealistic assumption [30]. For example, if a patient had a heart attack during the period of missing data, this event would not be reflected in the imputed data, as LOCF assumes the outcome remains constant.

LOCF is especially problematic in the context of imputing EHR datasets when the missing laboratory test data for a patient cannot be carried forward as there are no prior instances of the test. Therefore, LOCF is not suitable for the completion of an EHR dataset for prediction modelling. Also, frailty is recognised as a dynamic construct marked by frequent transitions between different frailty states over the course of time [31]. LOCF assumes that missing values remain constant over time and that the last available observation is representative of the entire time period. However, considering the dynamic nature of frailty, laboratory outcomes are expected to vary over time. Therefore, utilizing LOCF to complete a dataset for the prediction of frailty would fail to capture the inherent dynamics of frailty.

## Missing indicator method

The missing indicator method involves replacing missing values in a dataset with binary or quaternary indicators. Binary indicators indicate whether the value is present or absent, with a value of 0 indicating that a laboratory test has not been performed and a value of 1 indicating that it has been performed [32]. On the other hand, quaternary indicators not only indicate the presence or absence of the value but also capture the value itself. For example, they can be set to low if the test outcome falls within the lower range, normal if it falls within the normal range, high if it falls within the higher range, and not conducted if the test has not been performed. Quaternary indicators provide a more comprehensive representation of the data. This additional information can be utilised in the prediction model.

As a result, including each participant and each variable in the analysis becomes feasible, thereby preserving the statistical power. However, this method can lead to biased estimates, even when data is MCAR or when the amount of missing data is relatively small [33]. For that reason, it is generally advisable to exercise caution when considering the utilization of the missing indicator method [33].

A lot of potentially valuable information is lost when the laboratory values are replaced with the binary or quaternary indicators. To still use the potential of this method, the binary missing indicator method could also be used in combination with other imputation methods, to ensure that the information is not lost. In this combination method, an auxiliary matrix is created to differentiate between observed and imputed values. This matrix includes binary indicators, where 1 indicates the presence of an observed value, and 0 indicates the absence of the observed value. This method helps to distinguish between imputed values and those that were observed in the original dataset, facilitating analysis and interpretation. Thereby, the missing indicator method is interesting because the absence of certain variable measurements can offer valuable clinical information [34].

In our case, there are laboratory values which have missing percentages above 80%. When the percentage of missing data is very high, it becomes increasingly difficult to impute these values feasibly. In such cases, the missing indicator method may be useful. In situations where there is a high degree of missingness, the mere presence or absence of a laboratory value can be informative enough for a prediction model. In fact, the scarcity of such laboratory tests may indicate the existence of specific clinical conditions or risk factors, which can be leveraged to enhance the predictive accuracy of the frailty model. Moreover, incorporating the auxiliary matrix containing missing indicators proves to be highly advantageous in subsequent stages of the prediction model. By including the auxiliary matrix, the informative MNAR relationship is retained, as it enables differentiation between the observed values and the imputed values.

## Regression imputation

Regression imputation involves estimating values of a continuous variable using a regression model with other variables as predictors. It leverages the information from the (nearly) complete variables when the variables are correlated. Regression imputation has several advantages. Firstly, it is a straightforward method that is easy to implement. Secondly, it can handle missing data from both continuous and categorical variables. Thirdly, it can preserve the sample size and reduce the loss of information due to missing data. However, regression imputation assumes that the relationships between the variables are linear and that the missing values are MAR or MCAR. If these assumptions are met, regression imputation can provide accurate imputations and improve the validity and power of subsequent analyses. This method has been shown to perform well when the variables are missing at random (MAR) and the imputation model is correctly specified [34]. However, as in mean imputation, it does not add new information to the dataset, it only increases the sample size [35]

The idea of borrowing information from observed variables is also used in other imputation methods, such as multiple imputation (MI). In a later section, the methodology of MI will be explained in detail. MI is the more advanced option as it takes the uncertainty introduced by imputation into account. Therefore, it is not advised to use regression imputation when more sophisticated techniques, such as MI, are available [36].

Regression imputation offers several advantages compared to mean/median imputation. However, it is important to note that regression imputation relies on the assumptions of linearity in the relationships between variables and the missingness being MAR or MCAR. In the context of completing our dataset, these assumptions do not hold. The missing values are not MAR or MCAR, and there is no guarantee that the relationships between variables are exclusively linear.

## Stochastic regression imputation

Stochastic regression imputation is a more advanced form of regression imputation that adds random noise to the imputed values. The random noise serves to account for the uncertainty in the imputed values and helps to produce a wider range of plausible imputations [36].

However, there are also limitations to stochastic regression imputation. It can be computationally intensive, which may make it impractical for large EHR datasets. Additionally, it may not be appropriate for datasets with highly non-linear relationships between variables, as it assumes a linear relationship between observed and missing values. Stochastic regression and regression imputation share common limitations, including assumptions of MAR or MCAR and linearity. Consequently, stochastic regression is likely not an appropriate approach for completing our EHR dataset.

## Random Forest

Random Forest (RF) imputation is an alternative method for handling missing data that overcomes some of the limitations of regression imputation. RF imputation involves constructing multiple decision trees, each using a different subset of the data, and averaging their predictions to generate imputed values [37, 38]. Unlike regression imputation, RF imputation does not rely on assumptions of linearity in the relationships between variables or the missingness mechanism. Instead, it leverages the power of ensemble learning to impute missing values based on a combination of decision trees.

RF is able to handle mixed data types, including categorical , numerical, and ordinal variables. This makes it particularly useful for imputing missing values in datasets with diverse data types, such as EHR datasets, where a wide range of variables may be present.

The advantages of RF imputation include its ability to capture non-linear relationships between variables, handle complex data structures, and incorporate uncertainty in the imputation process [39]. In their study, Tang and Ishwaran even showed that the performance of RF was good under moderate (50%) to high (75%) missingness, moreover, the performance was even promising in certain cases where the data was MNAR.

It is important to note that RF imputation, like other imputation methods, is not without limitations. The accuracy of imputations can be influenced by factors such as the amount of missing data and the representation of the underlying data distribution. Besides, RF imputation can be computationally intensive, especially with a large number of variables and missing values. Imputing a massive EHR dataset may require substantial computational resources and time, which is may not be the ideal approach when working on completing an EHR dataset for a prediction model. New data and new patients will enter the EHR dataset and the imputation model has to run multiple times instead of one time. A method with high computational demands may pose practical challenges in our upcoming applications.

Single imputation methods were generally preferred over complete case and available case analyses [27], however, recent studies have compared traditional single imputation methods to newer methods, indicating the existence of more effective alternatives [40–42].

## Multiple imputation

MI is a technique used to handle missing data by creating multiple plausible imputed datasets [43, 44]. It involves creating multiple plausible values for every missing data point, relying on the observed data and a specified imputation model.

MI involves three steps:

1. Initially, MI involves the creation of imputations, for the missing values within the dataset. This process creates a number of complete datasets (*m*), where the gaps left by missing data are filled with random selections from a distribution of plausible values.
2. In the subsequent phase, every imputed dataset is separately subjected to statistical analysis. Consequently, this step generates *m* distinct analyses.
3. The final step in MI involves the pooling of the *m* estimates into one estimate. This pooling takes into account the variability present both within and among the *m* imputed datasets. This results in statistically sound estimates that account for the uncertainty arising from missing data [44].

The main advantages of MI over any other single imputation method are that MI preserves variability, it keeps complex relationships, and it reduces the bias. However, MI involves creating multiple imputed datasets, which increases computational complexity compared to other imputation methods. Generating and analysing multiple datasets can be time-consuming and computationally demanding, especially for large EHR datasets. It is important to note that MI techniques still operate under assumptions regarding the nature of missingness in the data. Specifically, MI assumes that the missing data mechanism is either MAR or MCAR to provide an unbiased estimate [45].

MI requires specifying an imputation model that describes the relationships between variables. The accuracy of imputed values depends on the quality and appropriateness of the imputation model. Building a robust imputation model can be challenging, especially when dealing with complex EHR datasets with numerous variables. When building an MI-based imputation model, various choices have to be made. These choices include determining the imputation model, specifying the number of imputations, selecting appropriate imputation methods, and handling variable transformations, among others. When these choices are made correctly, MI could be beneficial for completing an EHR dataset.

In our dataset, all the laboratory tests have different distributions. One key advantage of MI is its ability to handle variables with different distributions simultaneously. By incorporating this flexibility, MI can produce imputations that are customised to each laboratory test's distribution. Additionally, MI enables the imputation of missing values while preserving the relationships between variables. This is particularly important in an EHR dataset where different laboratory tests are often interrelated or exhibit dependencies. Theoretically, MI could be advantageous for completing an EHR dataset when correct modelling choices have been made.

# Dataset

The dataset utilised in this study was obtained from the EHR data of the JBZ. Initially, the dataset comprised 23,858 patients. The inclusion criteria for the cohort were patients aged 70 years and older who had undergone some form of frailty screening in the JBZ. It is important to note that patients who explicitly objected to the use of their information for scientific research purposes were not included in the dataset.

In the initial dataset, also patients with the CGA as the frailty assessment were included. However, because of the unstructured nature of the CGA outcome it was not feasible within the timeframe of this thesis project to extract and utilise the outcomes of the CGA. Only frailty screeners with a structured outcome have been utilised. The structured frailty screeners present in the dataset are the APOP Screening [46], the Clinical Frailty Scale [47], the Tilburg Frailty Indicator [48], the Groningen Frailty Indicator [49] , and the G8 frailty instrument [50]. To ensure standardised and consistent data in our dataset a filtering process was implemented to select patients based on only having undergone one frailty screener with a structured outcome.
We established a six-month data window for each patient, encompassing the period starting six months prior to the frailty screener date and including the day of the frailty screener assessment. The rationale behind selecting patients with only one frailty screener assessment was driven by the target audience of the frailty prediction model – the elderly population visiting the hospital of whom the frailty status is unknown. We therefore focussed on individuals for whom we lacked prior knowledge or evidence of frailty. By including patients with only one frailty screener assessment, we aimed to capture a representative sample of the target population for whom frailty prediction is important. In order to maintain focus on the main objective of investigating imputation methods for the laboratory data, patients who lacked any laboratory values were excluded from the analysis.

Initially, the dataset comprised 23,858 patients; 16,247 patients were excluded due to having undergone more than one frailty screener or the assessment tool being the CGA, and an additional 1,285 patients were removed because no laboratory values were available for them. Consequently, the final dataset included 6,326 patients, ranging from 70 to 98 years of age (Median = 78, IQR = 73-82); 50% of the patients were female (Fig. 1). Among these patients, 15% were classified as frail based on the outcome of the frailty screener (Fig. 2). Notably, Figure 2 illustrates an increase in the number of frail patients as age advances. This is coherent with the fact that frailty prevalence increases with age [51].

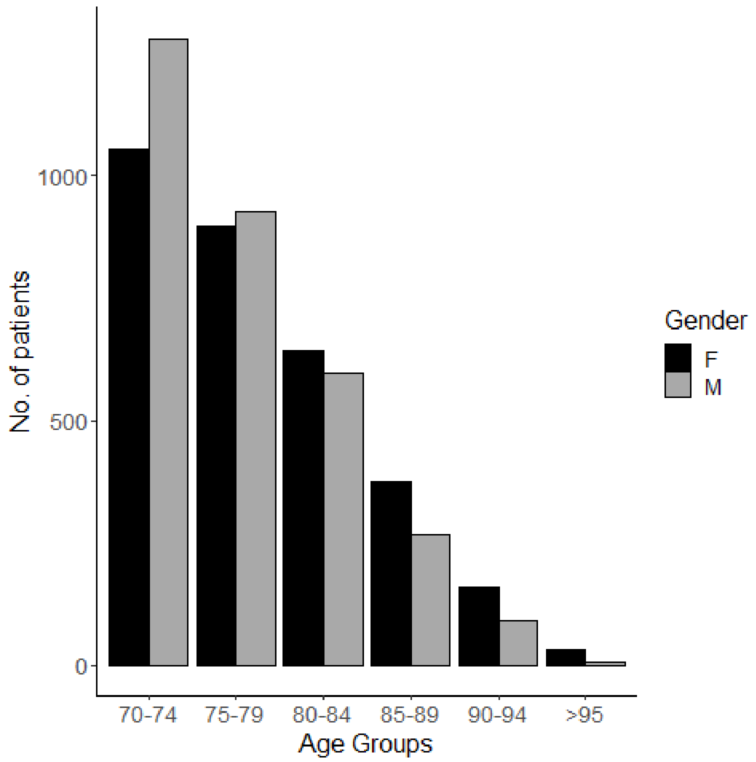



***Fig 1.*** *The distribution of the age groups and gender among the 6,326 patients in the final dataset.*

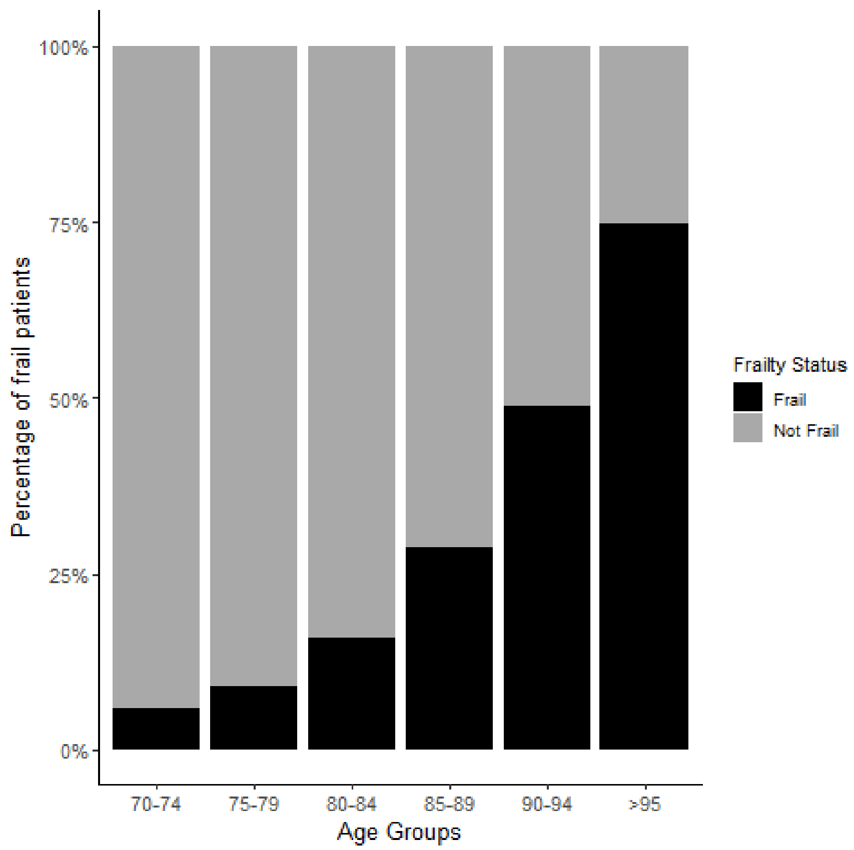



***Fig 2.*** *The percentage of frail patients filtered by age group among the 6,326 patients in the final dataset.*

# Pre-processing

Pre-processing of raw EHR data is crucial in ensuring data quality and suitability for subsequent analysis. This section outlines the key pre-processing steps undertaken, including data merging, handling measurement units, and outlier removal.

## Filtering and selection of patient records

The pre-processing stage involved the removal of patients who lacked any recorded laboratory values of blood samples or did not have a structured frailty assessment outcome. This data filtering process was performed to ensure the dataset's integrity and focus on patients with complete information relevant to the study objectives.

## Data merging for comprehensive analysis

To create a comprehensive dataset, the point-of-care testing (POCT) and non-POCT data pertaining to specific laboratory values were merged. POCT occurs directly at the bedside of the patient, is often conducted by nursing staff, while the non-POCT is conducted in the laboratory by laboratory staff. Even though the non-POCT measurements are regarded as more precise due to the laboratory protocols and oversight, statistical tests demonstrated no noteworthy differences in measurements. Moreover, the occurrence of outliers displayed uniformity in both variable types. The merging process involved combining the POCT and non-POCT data for five laboratory tests: GLC, INR, Hb, CRP, and eGFR.

## Handling measurement units for consistency

It was observed that certain laboratory values exhibited inconsistencies in their measurement units, due to a fault in the export process from the EHR. It became clear this was due to the presence of urine samples alongside blood samples within the dataset. To address this issue, values associated with urine samples were systematically removed, focusing solely on blood samples for the relevant laboratory values.

## Meticulous outlier removal with expert validation

The pre-processed data underwent a meticulous outlier removal procedure. Outliers, characterised as values that significantly deviated from the expected or biologically plausible range, were identified and carefully evaluated. This step was conducted with the utmost care to ensure that only outliers displaying biologically implausible values were removed. It is important to note that outliers were considered to be outside the biologically plausible range only when other related values measured at the same time did not exhibit such extreme values. To reinforce the reliability of this process, a domain expert with specialised expertise in the field was actively involved. Her invaluable insights and validation helped confirm the biological implausibility of the outliers, thereby upholding the integrity of the data cleaning process.

The values that were removed during the pre-processing steps, such as the measurement unit standardization and outlier removal, were replaced with "missing" values. This replacement signifies that these measurements were not conducted or recorded for the corresponding data points. Consequently, there is no available information regarding these specific measurement points within the final dataset.

## Transformation of string representations to numeric values

An additional data transformation step involved handling values represented as strings, particularly those containing "<" (smaller than) and “>” (greater than) symbols preceding the numerical value. To prevent the loss of valuable information, these string representations, such as ">90," were converted to their corresponding numeric values (e.g., 90). The interpretation of the ">" and "<" symbols can vary in two ways: either indicating that all values above or below a certain threshold are considered normal, or representing a cut-off point where the value becomes extremely high or low. In either case, the converted value is set to the upper or lower bound of the normal or extreme range. This conversion influences the distribution of the data points, resulting in a peak at the specific threshold value instead of multiple data points above or below it. The conversion from such string to numeric values was paramount to prevent the exclusion of information. Retaining the original string representations would have resulted in the loss of valuable data points, potentially leading to biased or incomplete results.

In this thesis, the focus was on retaining only the last observation of each patient for each laboratory test because addressing the challenges associated with a large amount of temporal data was beyond the scope of this thesis. This approach resulted in a dataset mostly consisting of data recorded on the day of the frailty screener. By doing so, a manageable dataset was created for this thesis project, which facilitated the investigation of the most effective imputation method for handling the missing data. In Table 1, the specifics of the final dataset are shown.

*Table 1. Final dataset specifications after all the pre-processing steps*

| Abbr. | Full name | Unit | Range (min-max) | Median (IQR) | Reference values | Missing in X% of the patients |
|---|---|---|---|---|---|---|
| **ALAT** | Alanine aminotransferase | U/L | X ∈ [9, 2011] | 20 (15-29) | F: X ∈ [0, 34] and M: X ∈ [0, 44] | 20.38% |
| **ALB** | Albumin | g/L | X ∈ [12, 53] | 37 (33-39) | X ∈ [35, 50] | 27.39% |
| **ALP** | Alkaline Phosphatase | U/L | X ∈ [21, 6510] | 80 (66-102) | X ∈ [40, 120] | 27.57% |
| **ASAT** | Aspartate aminotransferase | U/L | X ∈ [8, 2496] | 23 (18-32) | F: X ∈ [0, 29] and M: X ∈ [0, 34] | 30.76% |
| **BILD** | Direct bilirubin | µmol/L | X ∈ [2, 422] | 3 (2-5) | X ∈ [0, 4] | 33.43% |
| **BILT** | Total bilirubin | µmol/L | X ∈ [2, 489] | 9 (6-13) | X ∈ [0, 16] | 31.38% |
| **Ca** | Calcium | mmol/L | X ∈ [1.34, 3.67] | 2.35 (2.26-2.44) | X ∈ [2.15, 2.6] | 31.55% |
| **CD4** | Percentage CD4 lymphocytes | % | X ∈ [19.9, 87.5] | 53.6 (38.0-64.8) | X ∈ [60, 80] | 99.87% |
| **CK** | Creatine kinase | U/L | X ∈ [15, 29403] | 89 (59-142) | F: X ∈ [0, 144] and M: X ∈ [0, 170] | 31.72% |
| **Cl** | Chloride | mmol/L | X ∈ [17, 122] | 103 (100-105) | X ∈ [97, 107] | 35.44% |
| **CRE** | Creatinine | µmol/L | X ∈ [22, 1681] | 80 (66-100) | F: X ∈ [49, 90] and M: X ∈ [64, 104] | 3.21% |
| **CRP** | C-reactive protein | mg/L | X ∈ [1, 608] | 10 (3-100) | X ∈ [0, 8] | 19.85% |
| **eGFR** | Estimated glomerular filtration rate | mL/min/ 1.73m2 | X ∈ [3, 90] | 70 (54-84) | X > 90 | 30.29% |
| **ESR** | Erythrocyte sedimentation rate | mm/uur | X ∈ [2, 120] | 9 (3-21) | F: X ∈ [0, 30] and M: X ∈ [0, 20] | 86.45% |
| **FA** | Folic acid | nmol/l | X ∈ [2.5, 54] | 14 (10-23) | X ∈ [6, 39] | 90.37% |
| **FER** | Ferritin | µg/l | X ∈ [1.2, 13000] | 140 (47-353) | F: X ∈ [15, 270] and M: X ∈ [22, 300] | 78.96% |
| **FGLC** | Fasting glucose | mmol/L | X ∈ [3.2, 37.4] | 5.9 (5.2-7.1) | X ∈ [4, 6] | 83.65% |
| **FT4** | Thyroxine | pmol/L | X ∈ [6.8, 34] | 16.0 (14.3-19.0) | X ∈ [9, 24] | 90.99% |
| **GGT** | Gamma glutamyl transferase | U/L | X ∈ [6, 3252] | 28 (19-51) | F: X ∈ [0, 39] and M: X ∈ [0, 54] | 26.28% |
| **GLC** | Glucose | mmol/L | X ∈ [2.7, 41.3] | 6.8 (5.8-8.4) | X ∈ [4, 7.8] | 23.57% |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Hb** | Hemoglobin | mmol/L | X ∈ [2.2, 14.7] | 8.3 (7.6-9.0) | F: X ∈ [7.5, 10] and M: X ∈ [8.5, 11] | 9.12% |
| **HDL** | High-density lipoproteins | mmol/L | X ∈ [0.2, 3.5] | 1.3 (1.0-1.6) | F: X < 1.1 and M: X > 0.9 | 56.39% |
| **Ht** | Hematocrit | L/L | X ∈ [0.11, 0.79] | 0.40 (0.47-0.43) | F: X ∈ [0.35, 0.45] and M: X ∈ [0.4, 0.5] | 9.15% |
| **INR** | International normalised ratio | INR | X ∈ [0.8, 10] | 1.1 (1.0-1.8) | X ∈ [0.8, 1.3] | 76.96% |
| **K** | Potassium | mmol/L | X ∈ [2.2, 8.2] | 4.1 (3.8-4.4) | X ∈ [3.5, 4.8] | 8.80% |
| **LDH** | Lactate dehydrogenase | U/L | X ∈ [77, 5621] | 225 (196-266) | X ∈ [0, 249] | 34.70% |
| **LDL** | Low-density lipoproteins | mmol/L | X ∈ [1, 7.5] | 2.4 (1.9-3.1) | X < 2.5 | 56.51% |
| **MCH** | Mean corpuscular hemoglobin | fmol/L | X ∈ [0.82, 2.77] | 1.89 (1.81-1.96) | X ∈ [1.7, 2.1] | 51.26% |
| **MCV** | Mean corpuscular volume | fL | X ∈ [52, 130] | 90 (87-94) | X ∈ [80, 100] | 9.15% |
| **Mg** | Magnesium | mmol/L | X ∈ [0.16, 1.57] | 0.78 (0.72-0.85) | X ∈ [0.7, 1.1] | 68.00% |
| **Na** | Sodium | mmol/L | X ∈ [106, 164] | 139 (136-141) | X ∈ [135, 145] | 9.17% |
| **Neut** | Neutrophil | x10^9/L | X ∈ [0.1, 71.6] | 6.3 (4.4-9.2) | X ∈ [1.5, 7.5] | 22.54% |
| **P** | Phosphate | mmol/L | X ∈ [0.4, 3.6] | 1.10 (0.95-1.30) | X ∈ [0.8, 1.5] | 93.32% |
| **PLT** | Thrombocytes | x10^9/L | X ∈ [5, 1705] | 244 (196-304) | X ∈ [140, 400] | 9.20% |
| **PT** | Prothrombin time | sec | X ∈ [11, 100] | 115 (13-17) | X ∈ [11, 14] | 91.46% |
| **PTH** | Parathyroid hormone | pmol/L | X ∈ [0.5, 170] | 11.0 (6.9-21.0) | X ∈ [2, 9.3] | 97.80% |
| **RBC** | Red blood cell | x10^12/L | X ∈ [1.24, 8.28] | 4.45 (4.07-4.79) | F: X ∈ [4, 5] and M: X ∈ [4.5, 5.5] | 9.25% |
| **TCHOL** | Total cholesterol | mmol/L | X ∈ [1.6, 10.2] | 4.5 (3.8-5.3) | X < 5 | 55.64% |
| **TG** | Triglyceride | mmol/L | X ∈ [0.3, 13] | 1.3 (1.0-1.9) | X ∈ [0, 2.2] | 56.09% |
| **TP** | Total protein | g/L | X ∈ [43, 114] | 71.5 (68.0-75.0) | X ∈ [63, 83] | 33.83% |
| **TSH** | Thyroid stimulating hormone | mU/L | X ∈ [0.008, 62.0] | 1.9 (1.1-3.0) | X ∈ [0.55, 4.8] | 73.84% |
| **UR** | Urea | mmol/L | X ∈ [1.6, 70.6] | 6.7 (5.3-8.7) | X ∈ [2.5, 6.4] | 20.36% |
| **VitB12** | Vitamin B12 | pmol/L | X ∈ [81, 1476] | 357 (267-523) | X ∈ [156, 672] | 85.05% |
| **VitD25** | 25-Hydroxy vitamin D | nmol/L | X ∈ [11, 190] | 61 (46-77) | X ∈ [50, 200] | 90.33% |
| **VitD3** | 1,25 dihydroxy vitamin D3 | pmol/L | X ∈ [35,3, 120] | 78 (56.5-98.8) | X ∈ [47.0, 130.3] | 99.97% |
| **WBC** | White blood cell | x10^9/L | X ∈ [0.3, 208.5] | 8.8 (6.9-11.6) | X ∈ [4, 10] | 9.15% |

*Min = minimum, max = maximum, IQR = Interquartile range.*

# Feature selection

In the context of developing a model to predict the probability of being frail, it is crucial to carefully select the features that will be included in the analysis. The primary focus of this thesis is not on selecting features, but rather on effectively addressing the challenge posed by a large amount of missing data within the laboratory variables of the dataset. The goal is to construct a dataset that effectively captures the necessary information for accurate prediction, while also addressing the issue of missing data. Not all of the 46 laboratory tests present in the original dataset will be utilised in the subsequent building of the model. This section aims to provide a comprehensive explanation of the rationale behind the inclusion or exclusion of specific laboratory tests from the perspective of handling missingness. Various factors were considered in this decision-making process, including the extent of missingness and high correlations observed among the laboratory tests.

## High percentage of missingness

Laboratory tests with a missingness exceeding 80%, being CD4, ESR, FA, FGLC, FT4, P, PT, PTH, VitB12, VitD25, and VitD3, will not be incorporated in the subsequent imputation comparison analysis. This exclusionary decision is grounded in the acknowledgment that variables exhibiting such a high degree of missingness present a significant challenge for reliable imputation methods [22]. Consequently, including these variables in the imputation analysis would introduce uncertainty and potential bias, compromising the integrity and validity of the results. However, despite their high missingness, certain laboratory tests will be retained in the final dataset employing the missing indicator method. The rationale behind this decision will be elaborated upon in the subsequent section.

## Significant relationship with frailty

Among the remaining laboratory tests, a thorough investigation was conducted to determine the presence of a significant relationship between the numeric values of these tests and the frailty status, this is shown in Table 2.A. The objective was to identify variables that would be informative and useful for the frailty prediction model. In cases where no significant difference was found in the test results between the frail and non-frail individuals, the respective laboratory tests were deemed not suitable for inclusion in the prediction model, and therefore not useful to include in further imputation analysis. Additionally, for laboratory tests where no significant relationship was observed, an additional assessment was carried out to verify the sufficiency of the sample size. Furthermore, the expertise of a domain expert was solicited to validate the significance of these tests based on their knowledge. No significant relationship between the numeric values of TG ($p$ = 9.97E-01), BILIT ($p$ = 1.05E-01), ALAT ($p$ = 3,07E-01), CRP ($p$ = 3,76E-01), Mg ($p$ = 7,07E-01), FER ($p$ = 1,86E-01), TSH ($p$ = 3,06E-01) and the frailty status was observed. However, the domain expert thought that ALAT and CRP would be valuable for predicting frailty. Considering the expert's input, ALAT and CRP will be retained in this thesis for the imputation comparison analysis. As a result, TG, BILIT, Mg, FER, and TSH will not be incorporated in the imputation comparison analysis.

Besides, the presence of a result for any of these laboratory tests alone could potentially serve as a valuable indicator or predictor of frailty. To investigate this assumption further, an analysis was conducted to examine the frequency of laboratory test occurrences among frail and non-frail patients. The percentage of frail patients among the total number of patients for each laboratory test was calculated and is presented in Tables 2.A and 2.B. Despite the high missingness observed in the laboratory tests, specifically for P, PT, FT4, ESR, PTH, and FA, it is noteworthy that the percentage of conducted laboratory tests on frail patients is relatively higher compared to the overall prevalence of frail patients in the dataset of 14.98%. For these laboratory tests, the percentage of conducted tests among frail patients ranges from 19,10% up to 30.28%, this is shown in Table 2.B. This is also the case for BILIT, Mg, FER, and TSH, as shown in Table 2.A. These findings suggest that there may be a higher likelihood of conducting these specific laboratory tests in frail compared to non-frail patients, indicating a potential association between these tests being performed and frailty status. Based on these findings, BILIT, Mg, FER, and TSH will be retained in the final dataset for the prediction model, utilizing the missing indicator method.

*Table 2.A. Significance between laboratory tests and frailty status*

| Lab test | Coefficient | p-value | Presence of a significant relationship with frailty, $p < 0.05$ | Frail patient percentage* |
|---|---|---|---|---|
| **LDL** | -2.41E-01 | 4.21E-04 | Yes, negative | 12.21% |
| **HDL** | -7.55E-01 | 1.24E-06 | Yes, negative | 12.21% |
| **TCHOL** | -2.43E-01 | 8.19E-06 | Yes, negative | 12.32% |
| **TG** | -1.72E-03 | 9.79E-01 | No | 13.14% |
| **Cl** | -2.80E-02 | 4.97E-04 | Yes, negative | 14.25% |
| **ALB** | -8.28E-02 | 4.92E-24 | Yes, negative | 14.48% |
| **eGFR** | -1.63E-02 | 1.92E-15 | Yes, negative | 14.50% |
| **BILID** | 4.48E-03 | 2.40E-02 | Yes, positive | 14.63% |
| **CK** | 1.34E-04 | 4.83E-02 | Yes, positive | 14.72% |
| **MCH** | -4.28E-01 | 8.78E-03 | Yes, negative | 14.76% |
| **UR** | 5.37E-02 | 2.31E-17 | Yes, positive | 14.91% |
| **Ca** | -5.56E+00 | 4.98E-02 | Yes, negative | 14.92% |
| **Neut** | 2.73E-02 | 1.52E-03 | Yes, positive | 14.94% |
| **BILIT** | 2.76E-03 | 1.05E-01 | No | 15.02% |
| **TP** | -3.14E-02 | 1.26E-05 | Yes, negative | 15.05% |
| **CRE** | 4.66E-03 | 5.18E-05 | Yes, positive | 15.08% |
| **K** | -1.36E+00 | 4.33E-03 | Yes, negative | 15.12% |
| **Na** | -8.17E-01 | 5.99E-04 | Yes, negative | 15.18% |
| **Hb** | -3.22E-01 | 1.20E-28 | Yes, negative | 15.30% |
| **MCV** | 1.24E-02 | 4.17E-02 | Yes, positive | 15.31% |
| **WBC** | 8.90E-03 | 3.90E-02 | Yes, positive | 15.31% |
| **Ht** | -6.08E+00 | 2.44E-20 | Yes, negative | 15.31% |
| **PLT** | 1.09E-03 | 9.69E-04 | Yes, positive | 15.31% |
| **GLC** | 4.83E-02 | 7.54E-06 | Yes, positive | 15.35% |
| **ASAT** | 7.77E-04 | 1.67E-02 | Yes, positive | 15.35% |
| **LDH** | 6.03E-04 | 4.79E-04 | Yes, positive | 15.39% |
| **GGT** | 1.03E-03 | 4.39E-02 | Yes, positive | 15.43% |
| **ALP** | 1.25E-03 | 5.42E-04 | Yes, positive | 15.54% |
| **ALAT** | 4.48E-04 | 3.07E-01 | No | 15.61% |
| **CRP** | 4.51E-04 | 3.76E-01 | No | 15.63% |
| **RBC** | -5.67E-01 | 1.34E-22 | Yes, negative | 16.45% |
| **Mg** | -1.75E-01 | 7.07E-01 | No | 19.99% |
| **FER** | -1.64E-04 | 1.86E-01 | No | 20.42% |
| **INR** | 2.41E-01 | 9.75E-06 | Yes, positive | 21.05% |
| **TSH** | -2.58E-02 | 3.06E-01 | No | 21.23% |

*LDL = Low-density lipoproteins. HDL = High-density lipoproteins. TCHOL = Total cholesterol. TG = Triglyceride. Cl = Chloride. ALB = Albumin. eGFR = Estimated glomerular filtration rate. BILID = Direct bilirubin. CK = Creatinine kinase. MCH = Mean corpuscular hemoglobin. UR = Urea. Ca = Calcium. Neut = Neutrophil. BILIT = Total bilirubin. TP = Total protein. CRE = Creatinine. K = Potassium. Na = Sodium. Hb = Hemoglobin. MCV = Mean corpuscular volume. WBC = White blood cell. Ht = Hematocrit. PLT = Thrombocytes. GLC = Glucose. ASAT = Aspartate aminotransferase. LDH = Lactate dehydrogenase. GGT = Gamma glutamyl transferase. ALP = Alkaline Phosphatase. ALAT = Alanine aminotransferase. CRP = C-reactive protein. RBC = Red blood cell. Mg = Magnesium. FER = Ferritin. INR = International normalised ratio. TSH = Thyroid stimulating hormone.*
** Frail patient percentage signifies the fraction of frail patients who underwent the indicated lab test, relative to the total number of patients who took the same test. It offers insights into the test's frequency among frail individuals as a percentage of all test recipients. The P-values presented in this table were obtained through logistic regression analyses.*

*Table 2.B. Significance between laboratory tests and frailty status of laboratory tests with a missingness exceeding 80%*

| Lab test | Coefficient | p-value | Significant or not, p < 0.05 | Frail patient percentage* |
|---|---|---|---|---|
| **CD4** | -2,93E-16 | 1,00E+00 | No | 0,00% |
| **VitD3** | -7,19E-17 | 1,00E+00 | No | 0,00% |
| **FGLC** | 3,12E-01 | 1,01E-01 | No | 6,65% |
| **VitD25** | 7,45E-04 | 9,58E-01 | No | 14,98% |
| **P** | 2,14E-01 | 4,89E-01 | No | 19,10% |
| **PT** | 1,66E-02 | 6,51E-02 | Yes, positive | 19,22% |
| **FT4** | 4,98E-02 | 8,82E-02 | Yes, positive | 19,93% |
| **ESR** | 1,53E-02 | 3,09E-04 | Yes, positive | 20,61% |
| **PTH** | 5,02E+02 | 9,52E-01 | No | 23,02% |
| **VitB12** | 2,92E-03 | 2,14E-03 | Yes, positive | 25,37% |
| **FA** | -6,77E-04 | 9,15E-01 | No | 30,28% |

*CD4 = Percentage CD4 lymphocytes. VitD3 = 1,25 dihydroxy vitamin D3. FGLC = Fasting glucose. VitD25 = 25-Hydroxy vitamin D. P = Phosphate. PT = Prothrombin time. FT4 = Thyroxine. ESR = Erythrocyte sedimentation rate. PTH = Parathyroid hormone. VitB12 = Vitamin B12. FA = Folic acid.*
** Frail patient percentage signifies the fraction of frail patients who underwent the indicated lab test, relative to the total number of patients who took the same test. It offers insights into the test's frequency among frail individuals as a percentage of all test recipients.*
*The P-values presented in this table were obtained through logistic regression analyses.*

## Derived variables and high correlations

The correlations between all laboratory tests are detailed in Appendix A. The laboratory test LDL will be excluded from the dataset for imputation comparison analysis. LDL is derived using the Martin Hopkins formula, which relies on the values of TOTCHOL, HDL, and TG. eGFR is calculated using an equation that takes the serum creatinine levels, age, and gender into account. Therefore, eGFR is also excluded from the dataset. Furthermore, MCH exhibits a strong correlation of 0.88 with MCV; MCV is highly prevalent in the dataset, whereas MCH is relatively less prevalent. Retaining MCV alone would provide sufficient information regarding the corpuscular characteristics of the erythrocytes. Also, Ht and Hb exhibit a strong correlation of 0.97, and Hb and RBC have a correlation of 0.89. Given the high correlation and the similar interpretation of these variables (higher Hb and higher RBC conveying similar meanings), the decision has been made to remove Ht and RBC from the dataset. Figure 3 presents the distribution of missing data for the remaining 25 laboratory tests, which will be employed throughout the rest of this thesis.

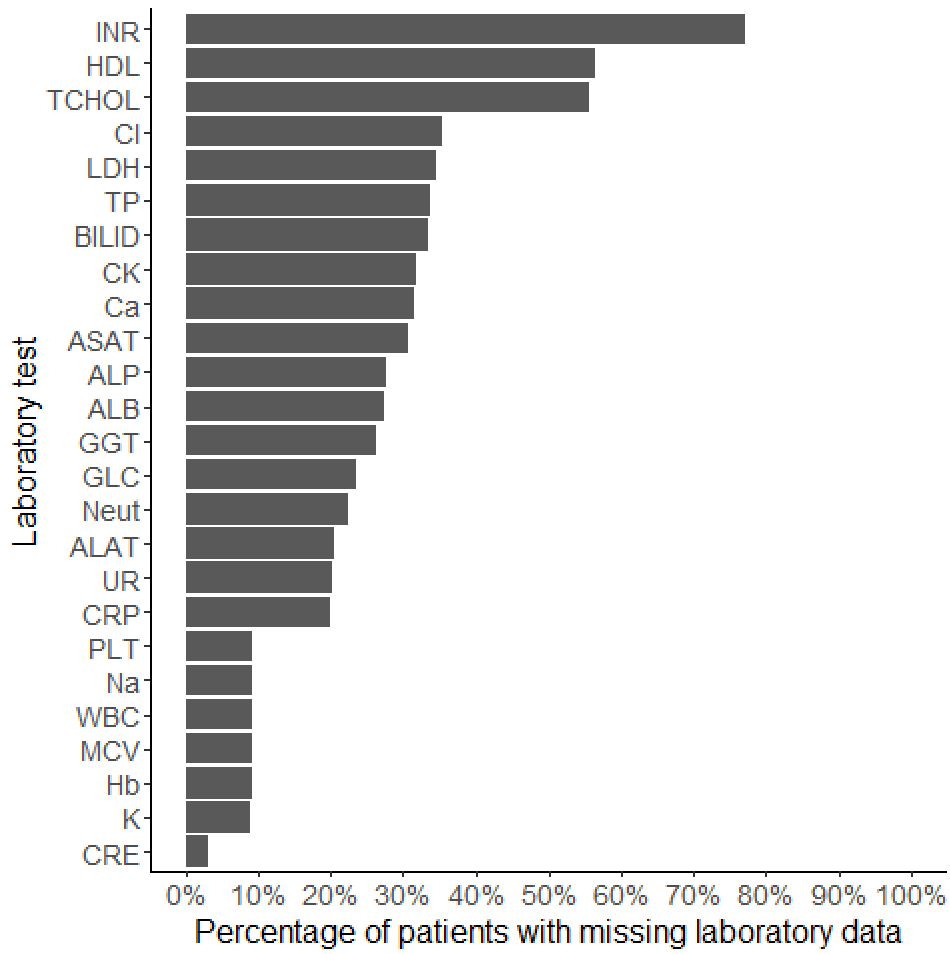



***Fig 3***. *Percentage of missingness of the remaining 25 laboratory tests. ALAT = Alanine aminotransferase, ALB = Albumin, ALP = Alkaline Phosphatase. ASAT = Aspartate aminotransferase, BILID = Direct bilirubin, Ca = Calcium, CK = Creatine kinase, Cl = Chloride, CRE = Creatinine, CRP = C-reactive protein, GGT = Gamma glutamyl transferase, GLC = Glucose, Hb = Hemoglobin, HDL = High-density lipoproteins, INR = International normalised ratio, K = Potassium, LDH = Lactate dehydrogenase, MCV = Mean corpus volume, Na = Sodium, Neut = Neutrophil, PLT = Thrombocytes, TCHOL = Total cholesterol, TP = Total protein, UR = Urea, WBC = White blood cell.*

# Method

The final 25 laboratory test variables were each imputed using nine different imputation methods; the nine resulting new variables were compared with each other for each original feature. These methods were:

1. Standard mean imputation: missing values were replaced with the mean of the available laboratory test values.
2. Mean imputation based on gender and age: the mean values were imputed separately for different gender and age group combinations; the age groups were 70-74, 75-79, 80-84, 85-89, 90-94, and ≥95.
3. Standard median imputation: missing values were replaced with the median of the available laboratory test values.
4. Median imputation based on gender and age: the median values were imputed separately for different gender and age group combinations; the age groups are 70-74, 75-79, 80-84, 85-89, 90-94, and ≥95.
5. RF imputation: the missForest package in R was employed to perform imputation using RF algorithms, which provide an imputation technique based on the relationships between variables.
6. Predictive mean matching (pmm) using single imputation: pmm imputation was applied to impute missing values using the mice package in R, utilizing the information from one single imputed dataset.
7. Bayesian linear regression (norm) using single imputation: a Bayesian linear regression technique was used to impute missing data from one single imputed dataset.
8. Predictive mean matching (pmm) using MI: pmm imputation was applied to impute missing values using the mice package in R, utilizing the information from multiple imputed datasets to generate more accurate estimates.
9. Bayesian linear regression (norm) using MI: a Bayesian linear regression technique was used to impute missing data from multiple imputed datasets, considering uncertainty in the imputed values.
10. The missing indicator method: Impute categories such as "low," "normal," "high," for available data and "not conducted" for missing data. Although the missing indicator method cannot be directly compared with the numeric imputation methods, it still holds potential as an alternative solution.

The final data frame comprised 6,326 rows representing individual patients, and 27 columns which encompassed age group, gender, and 25 different laboratory tests. The imputation methods mentioned were applied to this dataset. The utilised R code can be found in Appendix F and G, and the code specifications for the imputation methods can be found in Appendix H.

# Results

The comparison of the different imputation methods was approached from a multidisciplinary perspective, taking into account technical and biological considerations, and a combination of a data-centric evaluation and the medical considerations. Firstly, the distribution pattern of the values was analysed per variable. This involved assessing whether the distribution appeared to follow a (nearly) normal pattern or exhibited an extended tail at either the upper or lower end, or both. The data-centric evaluation also entailed an examination of the suitability of imputation methods for this project. Subsequently, an exploration into the medical significance of the laboratory test and the degree of missingness was undertaken. This medical perspective included considerations such as the informative missingness of laboratory tests, whether the tests are solely executed when a doctor observes relevant medical indicators or if they are performed more routinely. It also involved verifying whether the imputed values fell within the biologically plausible range.

As a first step, a selection of imputation methods was made based on a combination of technical and biological considerations.

The application of MI methods presents challenges for completing an EHR dataset that necessitates regular updates due to the inflow of new information on a daily basis. MI involves generating multiple datasets with imputed values, which is computationally intensive and time-consuming. With the dynamic nature of an EHR dataset requiring continuous updates, the repeated implementation of MI methods becomes impractical and resource intensive. Additionally, the differences between the pmm MI and norm MI, and their single imputation versions were minimal, this is shown in Appendix B-D. Consequently, **pmm MI** and **norm MI** were not used any further in this dataset.

A critical aspect across all laboratory tests is the avoidance of imputing very extreme values, because of the assumption of informative missingness. Even in cases where the extreme nature of a value is not observable for the medical doctors, it remains essential to refrain from imputing such values for predictive modelling purposes. The rationale behind this is that imputing extreme values could potentially lead to the prediction model being built upon inaccurately imputed extreme data points. Imputation methods pmm single and norm single have the tendency to impute values that mirror the distribution of the observed values, which can include a substantial number of extreme values. If so, these methods introduce numerous extreme values during imputation. As these extreme values hold significant influence, relying on wrongly imputed data could adversely affect the prediction model's outcomes. Furthermore, norm single imputes negative values when the observed values are not normally distributed. Negative values are not possible in these blood tests. Consequently, imputation methods **pmm single** and **norm single** were not used any further in this dataset. (The statistics and the distributions of the imputed values of pmm single and pmm norm for the included laboratory tests are shown in Appendix B-D.)

The standard mean and median imputation methods will also be excluded due to the availability of their enhanced counterparts, mean imputation based on gender and age, and median imputation based on gender and age. These refined versions offer more informative and accurate results since the values of diverse laboratory tests tend to vary with age and differ by gender. For instance, Hb levels tend to decrease with age, and they are generally lower in females compared to males [52, 53]. Illustratively, in our dataset the median Hb value for male patients aged 70-74 is 8.70, while the median Hb value for female patients aged ≥95 is 7.35. Therefore, to ensure a more accurate imputation process, **standard mean and median imputation** methods will also not be used any further in this dataset. (The statistics and the distributions of the imputed values of standard mean and median imputation for all laboratory tests are shown in Appendix B.)

After eliminating pmm MI, norm MI, pmm single, norm single, and the standard mean and median imputation techniques, the remaining methods are mean and median imputation based on gender and age, RF imputation, and the missing indicator method.

As a second step, all variables were categorised into three groups, based on a combination of data-centric and medical information (domain knowledge).

The first group comprised laboratory tests for which healthcare professionals can more easily determine the necessity of the test based on the observation of medical indicators. For instance, a significantly elevated level of BILID will manifest itself as yellowing of the skin, while CRP tests are conducted selectively when there are clinical indications of infection. Most of these laboratory tests showed a skewed distribution of the values and their value was either in the normal range or (very) extreme. In such cases, informative missingness is more easily justifiable, as these tests are typically ordered when the expectation is that extreme values will be detected. These tests are conducted based on the observation of specific symptoms in the patient. This held true for ALB, ALAT, ASAT, ALP, BILID, CK, CRP, GGT, GLC, Hb, LDH, Na, Neut, PLT, UR, and WBC. The objective was to ensure that values imputed for missing data did not deviate from the normal reference values because of the informative missingness. As a result, the chosen imputation method was **median imputation based on gender and age**. This choice was based on the skewed distributions observed in these laboratory tests. Median imputation is preferred over mean imputation in this context due to its capacity to account for outliers. (The statistics and the distributions of the imputed values of the different imputation methods for UR are shown as a representative example in Table 4, Figures 4 and 5, and for the other laboratory tests in Appendix B.)

*Table 4. Statistics of imputed values of the different imputation methods for UR.*

| UR | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 7.95 | 6.70 | 5.10 | 1.60 | 70.60 |
| **Mean** | 7.95 | 7.95 | 0.00 | 7.95 | 7.95 |
| **Mean gender-age** | 7.86 | 7.99 | 1.03 | 6.70 | 11.36 |
| **Median** | 6.70 | 6.70 | 0.00 | 6.70 | 6.70 |
| **Median gender-age** | 6.64 | 6.50 | 0.74 | 5.70 | 8.65 |
| **RF** | 6.88 | 6.35 | 2.27 | 4.50 | 30.30 |
| **pmm MI** | 7.81 | 6.60 | 4.85 | 1.60 | 70.60 |
| **pmm single** | 7.38 | 6.60 | 3.70 | 1.60 | 35.20 |
| **norm MI** | 7.83 | 6.70 | 4.95 | -7.84 | 70.60 |
| **norm single** | 7.48 | 7.42 | 4.22 | -5.19 | 26.76 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for UR (Urea). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*

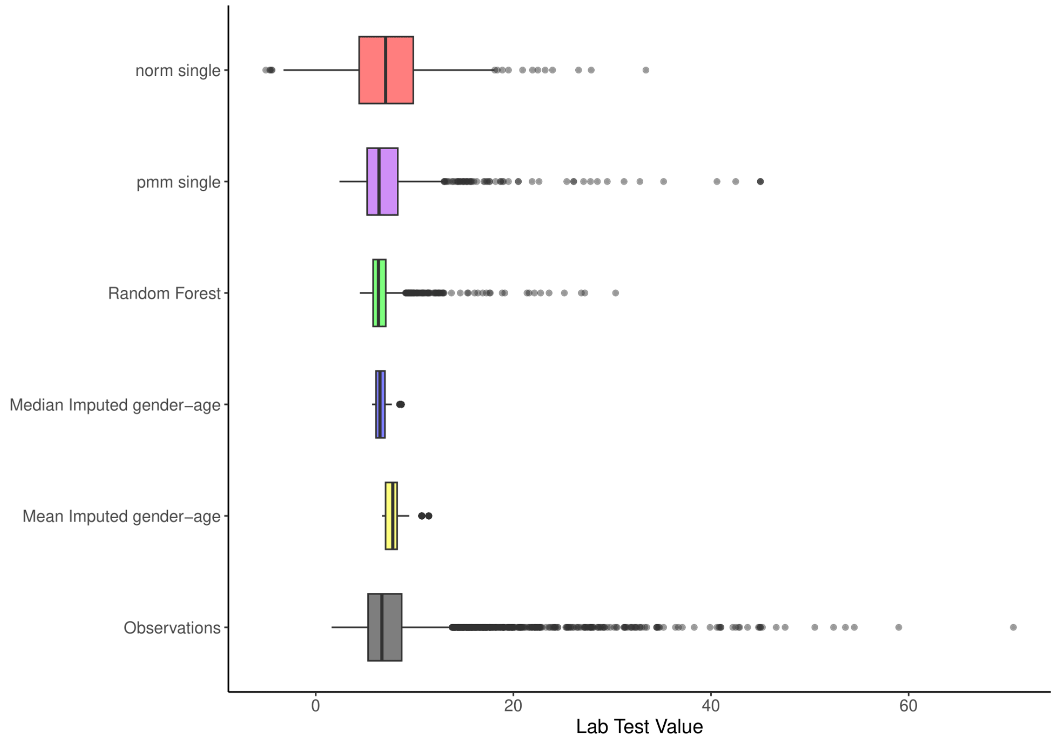



***Fig 3****. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, RF, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for UR (Urea).*

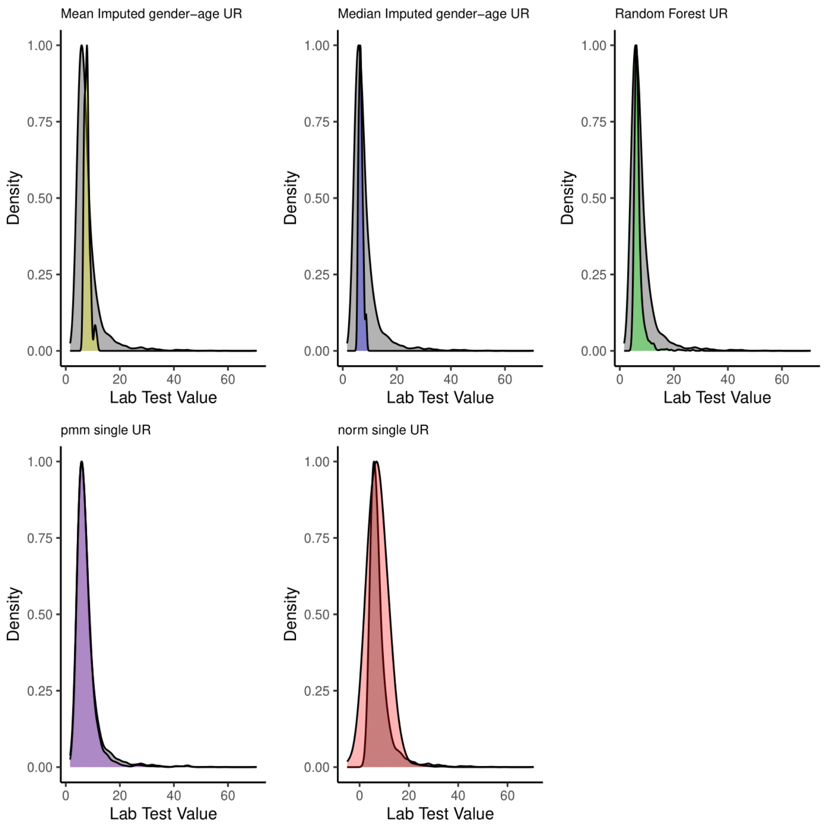



***Fig 4.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, RF, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for UR (Urea).*

For the second group of laboratory tests, externally identifying patients with extreme values was more difficult. Most of them showed an (almost) normal distribution of the values. For these variables, a somewhat broader range of imputed values was needed to get a better representation of the expected true nature of the missing values. This meant moving beyond the sole imputation of medians, while also avoiding excessive distribution of values, as imputing very extreme values is undesirable. Therefore, **RF** imputation was the selected method for handling missing data in these tests. These laboratory tests encompassed: Ca, Cl, CRE, HDL, K, MCV, TCHOL, and TP. RF imputation achieved a more extensive spectrum of values than mean or median imputation while it also refrained from excessive value imputation. In the example case below for MCV, RF imputed a broader range of values compared to mean or median imputation but the imputed values stayed in the normal reference value range. (The statistics and the distributions of the imputed values of the different imputation methods for MCV are shown in Table 5, Figures 6 and 7, and for the other laboratory tests in Appendix C.)

*Table 5. Statistics of imputed values of the different imputation methods for MCV.*

| MCV | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 90.38 | 90.00 | 6.02 | 52.00 | 130.00 |
| **Mean** | 90.38 | 90.38 | 0.00 | 90.38 | 90.38 |
| **Mean gender-age** | 90.32 | 90.22 | 0.46 | 89.88 | 92.43 |
| **Median** | 90.00 | 90.00 | 0.00 | 90.00 | 94.00 |
| **Median gender-age** | 90.36 | 90.00 | 0.52 | 52.00 | 130.00 |
| **RF** | 91.05 | 91.00 | 1.21 | 87.00 | 94.00 |
| **pmm MI** | 90.41 | 90.00 | 5.99 | 52.00 | 130.00 |
| **pmm single** | 90.54 | 90.00 | 5.43 | 59.00 | 106.00 |
| **norm MI** | 90.41 | 90.00 | 6.01 | 52.00 | 130.00 |
| **norm single** | 90.73 | 90.58 | 5.87 | 72.42 | 108.19 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for MCV (Mean Corpuscular Volume). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*

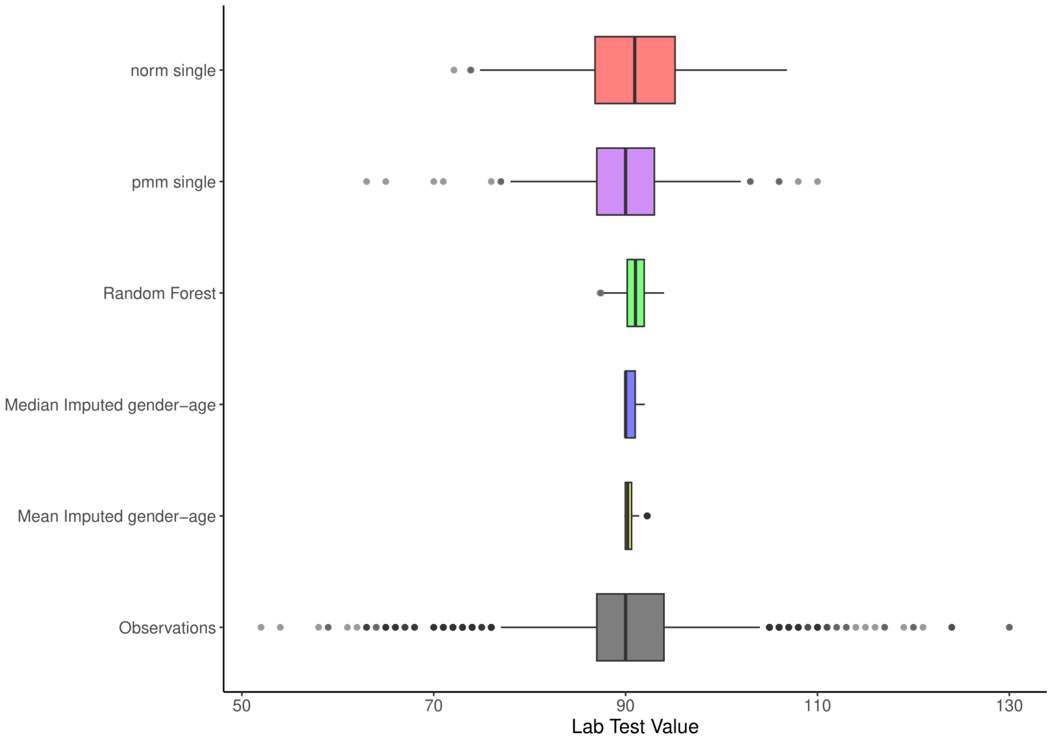



***Fig 5.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, RF, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for MCV (Mean Corpuscular Volume).*

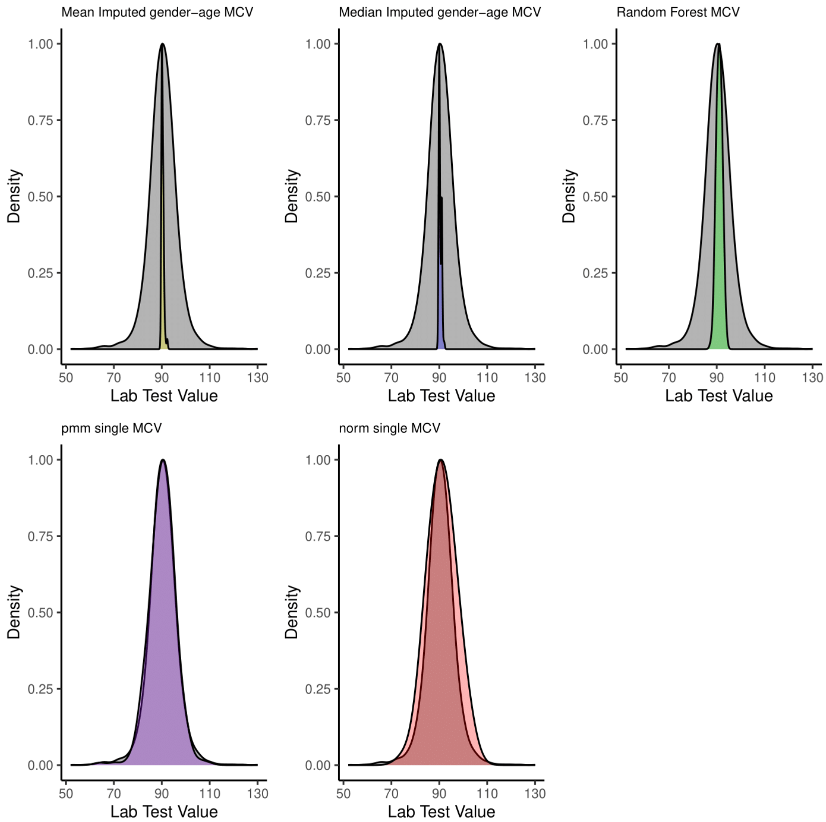



***Fig 6.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, RF, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for MCV (Mean Corpuscular Volume).*

For the third group and in this case the last remaining laboratory test, INR, the decision was made to employ **a ternary missing indicator method** (below normal range, above normal range, and not conducted). This laboratory test had a substantial missingness percentage (76.96%) and possesses a distinct nature: it is performed exclusively when patients are on specific medications. Given this context, having information on whether the test was conducted and if so, whether the outcome falls within the extremes of high or low values is inherently valuable. Moreover, the percentage of frail patients who undergo INR testing was higher compared to the overall percentage of frail patients (Table 2.B), further emphasising the clinical significance of this laboratory test in the context of this project. The statistics and the distributions of the imputed values of the different imputation methods for INR are shown in Appendix D)

In Appendix E, tables and figures of UR and MCV comparing the observed values with the observed + imputed values are shown. However, due to the large number of measurements, the graphs show significant overlap. The very high and low imputed values are no longer clearly visible, impeding a detailed assessment of the impact of each imputation technique. Consequently, in the decision-making process, the focus was primarily on the graphs comparing observed values to imputed values alone.

Table 6 displays the computation times for the different imputation methods. RF had a relatively long computation time compared to the other imputation techniques. However, the benefits derived from this imputation method (see above) outweighed the computational costs since the prediction model algorithm doesn't require continuous learning after each single patient, and with the utilisation of high-performance computers for building the final prediction model, the feasibility of using RF remains intact.

*Table 6. The computation times in seconds for all imputation methods*

| Imputation method | Computation time (sec) |
|---|---|
| **Mean** | 2 |
| **Mean gender-age** | 3 |
| **Median** | 2 |
| **Median gender-age** | 3 |
| **RF** | 1312 |
| **pmm MI** | 209 |
| **pmm single** | 19 |
| **norm MI** | 173 |
| **norm single** | 16 |
| **Missing indicator** | 5 |

*MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*

# Discussion

We conducted a comprehensive analysis of imputation methods for handling missing data in a dataset containing numerical laboratory test results extracted from a hospital EHR. Our approach considered technical, biological, and a combination of data-centric and medical perspectives. We assessed the distribution patterns of values and their medical significance. We highlighted the need to avoid imputing extreme values, particularly for laboratory tests which are generally ordered based on clinical indications. Enhanced mean and median imputation methods based on gender and age were preferred over standard mean and median imputations. Median imputation based on gender and age was chosen for tests with clear clinical indications and skewed distributions, while RF imputation was applied to tests with near-normal distributions where the clinical indication is not clear. For laboratory tests with high missingness percentages and unique nature, a ternary missing indicator method was used. Our multidisciplinary perspective not only enhances the robustness of our approach but also ensures that our early-stage decisions are in line with the principles of XAI. We emphasise the significance of transparent decision-making to establish trust and facilitate the acceptance of predictive models by healthcare professionals. They are, after all, the end-users of these models in clinical practice.

Our results show that the steps undertaken and the thoughtful deliberations conducted are essential. This not only holds true for this project but also for all future projects involving substantial (EHR) datasets containing significant numbers of missing data, as well as for assessing the value of previously published work in this field. In 2017, Goldstein and colleagues conducted a review article focusing on EHR prediction model studies [54]. Their analysis of 107 reviewed studies revealed that a mere 58 of them acknowledged the presence of missing data, with MI being the most frequently employed method for addressing this issue. To shed further light on this matter, Yasrebi-De Kom and colleagues conducted a systematic review aimed at critically evaluating prediction models developed based on EHR data [55]. Their findings echoed those of Goldstein et al., indicating that 36% of the studies did not provide detailed information regarding their approach to handling missing data. The majority of the reviewed studies that did broach the subject of missing data handling, opted for the MI method or mean imputation. Nevertheless, it could be possible that some studies may have considered missing data but did not report this in their paper. Our results show that each project and each variable therein require careful, multidisciplinary consideration to determine the most suitable approach for dealing with missing values. It is important to recognise that the decisions made and the challenges faced in this particular project may differ from those in future projects. Thus, the process of handling missing values will always need to be repeated and if necessary adapted to the specific circumstances of each subsequent project.

## Strengths and weaknesses

While our study has provided valuable insights and contributions to the field, it is essential to acknowledge certain limitations that have influenced our findings and interpretations.

The first limitation of our study pertains to the age distribution within the final dataset. Due to our deliberate exclusion of patients who underwent multiple frailty screenings, which primarily affected the oldest individuals who frequented the hospital more often, the age distribution shifted a little to the lower side. In the initial dataset, the age spanned from 70 to 101 years (Median = 82 IQR = 76-87) in the final dataset the age spanned from 70 to 98 years of age (Median = 78, IQR = 73-82). This exclusionary criterion not only influenced the age distribution but also implied that the proportion of frail patients in our dataset is likely to be a bit lower compared to the original dataset. However, the underlying rationale and decision-making processes remain unchanged.

Another limitation of our study is our use of only the last recorded value for each laboratory test. In the majority of patients, this last value was obtained on the day of the frailty test. When abnormal results were observed, further tests may have been conducted on subsequent days. These test outcomes were not included in the analysis. However, since the prediction model will be built to make a prediction on the first day of contact between the elderly patient and the hospital this should not have any negative consequences.

After addressing the limitations and constraints in our study, let's now shift our attention to the strengths that characterise our research. The focus of this thesis was evaluating various imputation methods for numerical laboratory data extracted from a hospital EHR. Also for different data types like binary and textual data, the core principles and rigorous approach in this thesis stay relevant in today's data-rich landscape. day's data-rich landscape, information comes in various forms, including binary data and unstructured (textual) data. The latter format introduces unique challenges, distinct from those posed by structured data, such as grammatical and spelling errors, ambiguities, and abbreviations [56]. Thereby, the challenges associated with missing data are not limited to numerical values alone; they permeate all facets of data analysis. As such, the core principles elucidated in this thesis, emphasizing transparency, multidisciplinary perspectives, and methodological rigor, remain profoundly relevant for ensuring data integrity, reliability, and meaningful insights across diverse data types. The central theme of this thesis, characterised by its thoroughness and multidisciplinary perspective, taking into account both biological, technical and a combination of data-centric and medical considerations should continue to guide decision-making regarding these diverse data types.

Furthermore, the application of median imputation based on gender and age, RF imputation, and the ternary missing indicator method showcased in this research represents more than a solution for the specific laboratory tests scrutinised. It underscores the adaptability and forward-thinking nature of our approach. These methods not only meet the specific needs of the laboratory tests under investigation but also lay a versatile and robust foundation for addressing future scenarios. As we anticipate developing the prediction model further, especially in contexts where analogous challenges surface, these methods should be foremost in our consideration for evaluation and potential adoption. This adaptability positions this research as a dynamic and forward-looking contribution, offering practical solutions not just for the present dataset but as a valuable resource for addressing similar challenges that may arise in future applications and studies.

In conclusion, our comprehensive analysis of imputation methods for managing missing data within the context of numerical laboratory test results extracted from a hospital EHR yields several key insights. Throughout our study, we have emphasised the importance of transparency and multidisciplinary perspectives, aligning with the integration of data-centric and medical considerations. These principles, which have guided our approach, not only addressed current challenges but also position our research as a valuable resource for future endeavours. Our research journey, marked by meticulous decision-making and methodological rigor, resonates with the broader context of predictive modelling in healthcare. It underlines the significance of comprehensive and transparent approaches, essential for nurturing trust among healthcare professionals—the ultimate end-users of predictive tools in real-world clinical practice.

As we reflect on our study's strengths and acknowledge its limitations, we recognise the value of our contributions. Our research not only addresses immediate challenges but also lays the groundwork for future applications. By weaving together the elements discussed in this discussion section, we emphasise the enduring relevance of our work, bridging the gap between current practices and the ever-evolving landscape of healthcare data analytics.

## References


[1] E. O. Hoogendijk, J. Afilalo, K. E. Ensrud, P. Kowal, G. Onder, and L. P. Fried, 'Frailty 1 Frailty: implications for clinical practice and public health', 2019. [Online]. Available: www.thelancet.com

[2] A. Clegg, J. Young, S. Iliffe, M. O. Rikkert, and K. Rockwood, 'Frailty in elderly people', in *The Lancet*, Elsevier B.V., 2013, pp. 752–762. doi: 10.1016/S0140-6736(12)62167-9.

[3] S. Vermeiren *et al.*, 'Frailty and the Prediction of Negative Health Outcomes: A Meta-Analysis', *Journal of the American Medical Directors Association*, vol. 17, no. 12. Elsevier Inc., pp. 1163.e1-1163.e17, Dec. 01, 2016. doi: 10.1016/j.jamda.2016.09.010.

[4] C. L. Chen *et al.*, 'Frailty is Associated with an Increased Risk of Major Adverse Outcomes in Elderly Patients Following Surgical Treatment of Hip Fracture', *Sci Rep*, vol. 9, no. 1, Dec. 2019, doi: 10.1038/s41598-019-55459-2.

[5] A. E. Stuck, A. L. Siu, G. D. Wieland, L. Z. Rubenstein, and J. Adams, 'Comprehensive geriatric assessment: a meta-analysis of controlled trials', *The Lancet*, vol. 342, no. 8878, pp. 1032–1036, 1993.

[6] J. W. Faller, D. do Nascimento Pereira, S. de Souza, F. K. Nampo, F. de Souza Orlandi, and S. Matumoto, 'Instruments for the detection of frailty syndrome in older adults: A systematic review', *PLoS One*, vol. 14, no. 4, Apr. 2019, doi: 10.1371/journal.pone.0216166.

[7] G. A. Aguayo *et al.*, 'Agreement between 35 published frailty scores in the general population', *Am J Epidemiol*, vol. 186, no. 4, pp. 420–434, Aug. 2017, doi: 10.1093/aje/kwx061.

[8] K. Bouillon *et al.*, 'Measures of frailty in population-based studies: An overview', *BMC Geriatr*, vol. 13, no. 1, 2013, doi: 10.1186/1471-2318-13-64.

[9] D. A. Lekan, D. C. Wallace, T. P. McCoy, J. Hu, S. G. Silva, and H. E. Whitson, 'Frailty Assessment in Hospitalized Older Adults Using the Electronic Health Record', *Biol Res Nurs*, vol. 19, no. 2, pp. 213–228, Mar. 2017, doi: 10.1177/1099800416679730.

[10] B. A. Goldstein, A. M. Navar, M. J. Pencina, and J. P. A. Ioannidis, 'Opportunities and challenges in developing risk prediction models with electronic health records data: A systematic review', *Journal of the American Medical Informatics Association*, vol. 24, no. 1, pp. 198–208, Jan. 2017, doi: 10.1093/jamia/ocw042.

[11] Z. Hu, G. B. Melton, E. G. Arsoniadis, Y. Wang, M. R. Kwaan, and G. J. Simon, 'Strategies for handling missing clinical data for automated surgical site infection detection from the electronic health record', *J Biomed Inform*, vol. 68, pp. 112–120, Apr. 2017, doi: 10.1016/j.jbi.2017.03.009.

[12] A. D. Sniderman, R. B. D'Agostino Sr, and M. J. Pencina, 'The role of physicians in the era of predictive analytics', *JAMA*, vol. 314, no. 1, pp. 25–26, 2015.

[13] B. J. Wells, A. S. Nowacki, K. Chagin, and M. W. Kattan, 'Strategies for Handling Missing Data in Electronic Health Record Derived Data', *eGEMs (Generating Evidence & Methods to improve patient outcomes)*, vol. 1, no. 3, p. 7, Dec. 2013, doi: 10.13063/2327-9214.1035.

[14] C. D. Newgard and R. J. Lewis, 'Missing data: how to best account for what is not known', *JAMA*, vol. 314, no. 9, pp. 940–941, 2015.

[15] S. W. J. Nijman *et al.*, 'Missing data is poorly handled and reported in prediction model studies using machine learning: a literature review', *Journal of Clinical Epidemiology*, vol. 142. Elsevier Inc., pp. 218–229, Feb. 01, 2022. doi: 10.1016/j.jclinepi.2021.11.023.

[16] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 2019.

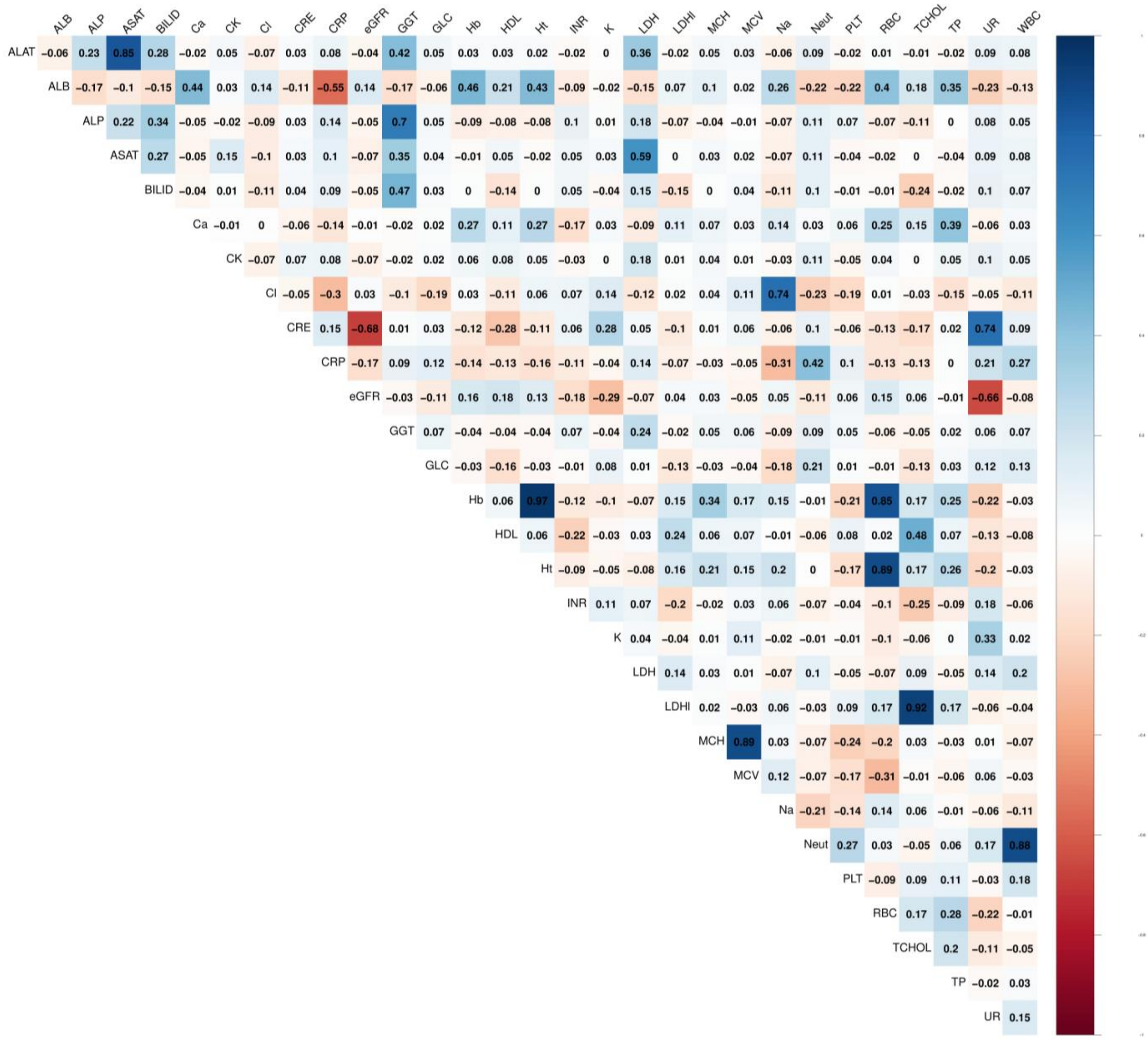
[17] A. L. M. Tan *et al.*, 'Informative missingness: What can we learn from patterns in missing laboratory data in the electronic health record?', *J Biomed Inform*, vol. 139, Mar. 2023, doi: 10.1016/j.jbi.2023.104306.

[18] R. H. H. Groenwold, ‘Informative missingness in electronic health record systems: the curse of knowing’, *Diagn Progn Res*, vol. 4, no. 1, Dec. 2020, doi: 10.1186/s41512-020-00077-0.
[19] A. B. Arrieta *et al.*, ‘Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI’, *Information fusion*, vol. 58, pp. 82–115, 2020.
[20] D. B. Rubin, ‘Inference and missing data’, *Biometrika*, vol. 63, no. 3, pp. 581–592, Dec. 1976, doi: 10.1093/biomet/63.3.581.
[21] J. L. Schafer, ‘Multiple imputation: a primer’, *Stat Methods Med Res*, vol. 8, no. 1, pp. 3–15, Feb. 1999, doi: 10.1177/096228029900800102.
[22] J. H. Lee and J. Charles Huber, ‘Evaluation of Multiple Imputation with Large Proportions of Missing Data: How Much Is Too Much?’, 2021. [Online]. Available: https://creativecommons.org/licenses/by-nc/4.0/
[23] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 2019.
[24] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 2019.
[25] A. Jadhav, D. Pramod, and K. Ramanathan, ‘Comparison of Performance of Data Imputation Methods for Numeric Dataset’, *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913–933, Aug. 2019, doi: 10.1080/08839514.2019.1637138.
[26] H. Kang, ‘The prevention and handling of the missing data’, *Korean Journal of Anesthesiology*, vol. 64, no. 5. pp. 402–406, May 2013. doi: 10.4097/kjae.2013.64.5.402.
[27] J. L. Schafer and J. W. Graham, ‘Missing data: Our view of the state of the art’, *Psychol Methods*, vol. 7, no. 2, pp. 147–177, 2002, doi: 10.1037/1082-989X.7.2.147.
[28] T. D. Pigott, ‘A Review of Methods for Missing Data’, 2001.
[29] N. Tsikriktsis, ‘A review of techniques for treating missing data in OM survey research’, *Journal of Operations Management*, vol. 24, no. 1, pp. 53–62, 2005, doi: 10.1016/j.jom.2005.03.001.
[30] D. Mavridis, G. Salanti, T. A. Furukawa, A. Cipriani, A. Chaimani, and I. R. White, ‘Allowing for uncertainty due to missing and LOCF imputed outcomes in meta-analysis’, *Stat Med*, vol. 38, no. 5, pp. 720–737, Feb. 2019, doi: 10.1002/sim.8009.
[31] T. M. Gill, E. A. Gahbauer, H. G. Allore, and L. Han, ‘Transitions between frailty states among community-living older persons’, *Arch Intern Med*, vol. 166, no. 4, pp. 418–423, 2006.
[32] R. H. H. Groenwold, I. R. White, A. R. T. Donders, J. R. Carpenter, D. G. Altman, and K. G. M. Moons, ‘Missing covariate data in clinical research: when and when not to use the missing-indicator method for analysis’, *Cmaj*, vol. 184, no. 11, pp. 1265–1269, 2012.
[33] M. J. Knol *et al.*, ‘Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example’, *J Clin Epidemiol*, vol. 63, no. 7, pp. 728–736, Jul. 2010, doi: 10.1016/j.jclinepi.2009.08.028.
[34] S. Van Buuren, *Flexible imputation of missing data*. CRC press, 2018.
[35] H. Kang, ‘The prevention and handling of the missing data’, *Korean Journal of Anesthesiology*, vol. 64, no. 5. pp. 402–406, 2013. doi: 10.4097/kjae.2013.64.5.402.
[36] C. K. Enders, *Applied missing data analysis*. Guilford Publications, 2022.
[37] L. Breiman, ‘Random Forests’, 2001.
[38] D. J. Stekhoven and P. Bühlmann, ‘Missforest-Non-parametric missing value imputation for mixed-type data’, *Bioinformatics*, vol. 28, no. 1, pp. 112–118, Jan. 2012, doi: 10.1093/bioinformatics/btr597.
[39] F. Tang and H. Ishwaran, ‘Random forest missing data algorithms’, *Stat Anal Data Min*, vol. 10, no. 6, pp. 363–377, Dec. 2017, doi: 10.1002/sam.11348.
[40] P. D. Allison, ‘Missing Data Techniques for Structural Equation Modeling’, *J Abnorm Psychol*, vol. 112, no. 4, pp. 545–557, 2003, doi: 10.1037/0021-843X.112.4.545.
[41] S. Van Buuren, *Flexible imputation of missing data*. CRC press, 2018.
[42] X. H. Zhou, G. J. Eckert, and W. M. Tierney, ‘Multiple imputation of public health research’, *Stat Med*, vol. 20, no. 9–10, pp. 1541–1549, May 2001, doi: 10.1002/sim.689.

[43] S. van Buuren, ‘Multiple imputation of discrete and continuous data by fully conditional specification’, *Stat Methods Med Res*, vol. 16, no. 3, pp. 219–242, Jun. 2007, doi: 10.1177/0962280206074463.
[44] D. B. Rubin, *Multiple imputation for nonresponse in surveys*, vol. 81. John Wiley & Sons, 2004.
[45] A. B. Pedersen *et al.*, ‘Missing data and multiple imputation in clinical epidemiological research’, *Clin Epidemiol*, vol. 9, pp. 157–166, Mar. 2017, doi: 10.2147/CLEP.S129785.
[46] J. de Gelder *et al.*, ‘Predicting adverse health outcomes in older emergency department patients: the APOP study’, *Neth J Med*, vol. 74, no. 8, pp. 342–352, 2016.
[47] K. Rockwood *et al.*, ‘A global clinical measure of fitness and frailty in elderly people’, *CMAJ. Canadian Medical Association Journal*, vol. 173, no. 5, pp. 489–495, Aug. 2005, doi: 10.1503/cmaj.050051.
[48] R. J. J. Gobbens, M. A. L. M. van Assen, K. G. Luijkx, M. T. Wijnen-Sponselee, and J. M. G. A. Schols, ‘The tilburg frailty indicator: Psychometric properties’, *J Am Med Dir Assoc*, vol. 11, no. 5, pp. 344–355, 2010, doi: 10.1016/j.jamda.2009.11.003.
[49] N. Steverink, ‘Measuring frailty: developing and testing the GFI (Groningen Frailty Indicator)’, *Gerontologist*, vol. 41, p. 236, 2001.
[50] C. A. Bellera *et al.*, ‘Screening older cancer patients: First evaluation of the G-8 geriatric screening tool’, *Annals of Oncology*, vol. 23, no. 8, pp. 2166–2172, Aug. 2012, doi: 10.1093/annonc/mdr587.
[51] E. O. Hoogendijk, J. Afilalo, K. E. Ensrud, P. Kowal, G. Onder, and L. P. Fried, ‘Frailty 1 Frailty: implications for clinical practice and public health’, 2019. [Online]. Available: www.thelancet.com
[52] C. W. Choi *et al.*, ‘Prevalence and characteristics of anemia in the elderly: Cross-sectional study of three urban Korean population samples’, *Am J Hematol*, vol. 77, no. 1, pp. 26–30, Sep. 2004, doi: 10.1002/ajh.20140.
[53] M. E. Salive *et al.*, ‘Anemia and Hemoglobin Levels in Older Persons: Relationsh7p with Age, Gender, and Health Status’, 1992.
[54] B. A. Goldstein, A. M. Navar, M. J. Pencina, and J. P. A. Ioannidis, ‘Opportunities and challenges in developing risk prediction models with electronic health records data: A systematic review’, *Journal of the American Medical Informatics Association*, vol. 24, no. 1, pp. 198–208, 2017, doi: 10.1093/jamia/ocw042.
[55] I. A. R. Yasrebi-De Kom *et al.*, ‘Electronic health record-based prediction models for in-hospital adverse drug event diagnosis or prognosis: a systematic review’, *Journal of the American Medical Informatics Association*, vol. 30, no. 5. Oxford University Press, pp. 978–988, May 01, 2023. doi: 10.1093/jamia/ocad014.
[56] M. Tayefi *et al.*, ‘Challenges and opportunities beyond structured data in analysis of electronic health records’, *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 13, no. 6. John Wiley and Sons Inc, Nov. 01, 2021. doi: 10.1002/wics.1549.

# Appendices

## Appendix A. Correlation graph.




***Appendix Figure 1.*** *The correlation heatmap of the remaining laboratory tests. Calculated using the Pearson correlation coefficient. Strong positive and negative correlations can be observed among several tests, revealing potential associations within this patient population.*

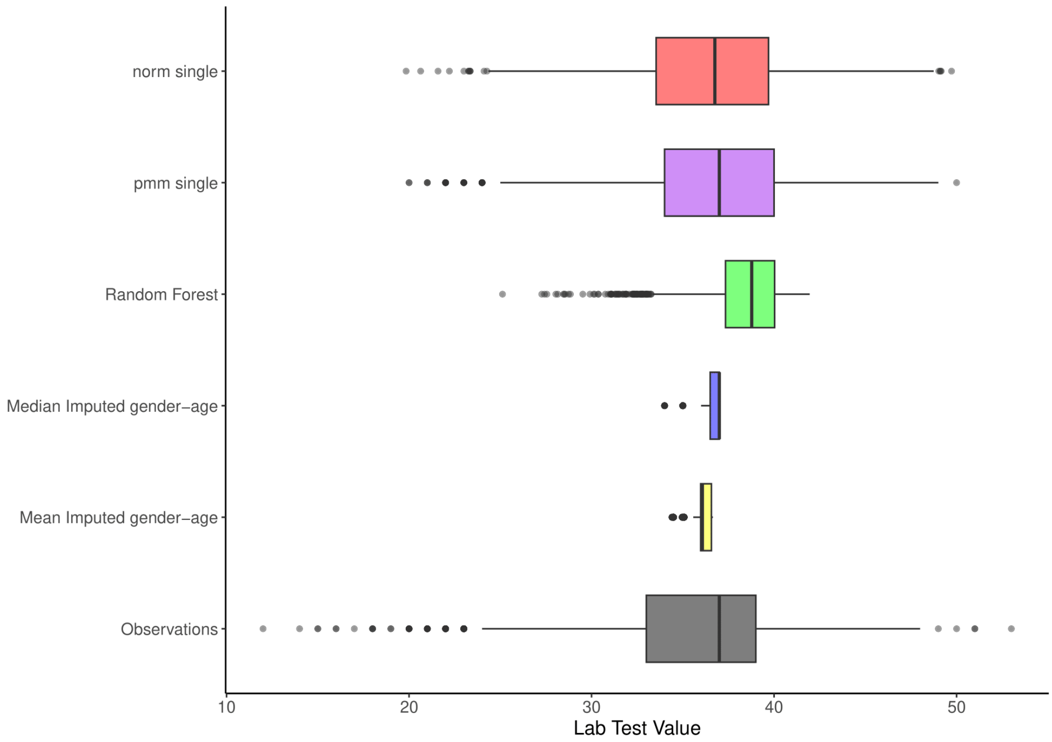
## Appendix B. Tables and figures of the laboratory tests for which median imputation based on gender and age is the chosen imputation method.

### ALB

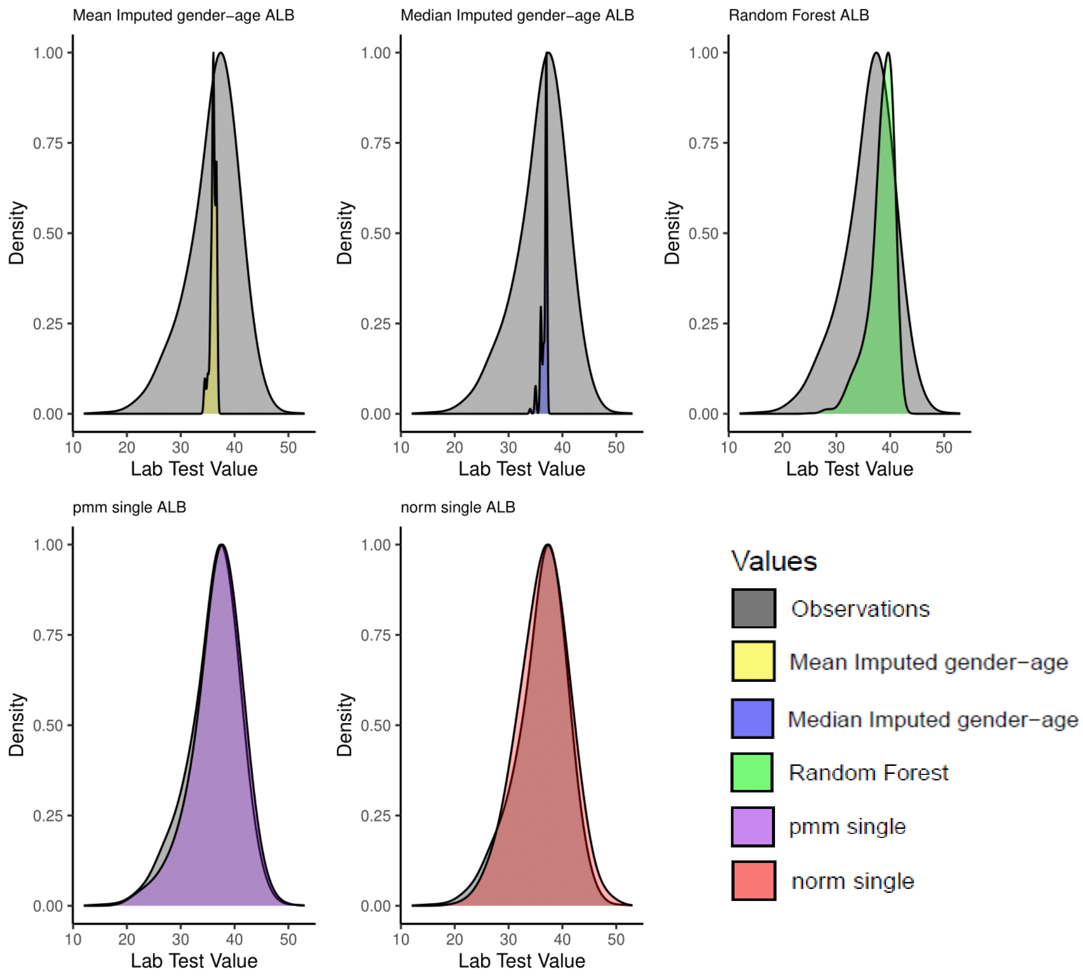
*Appendix Table 1. Statistics of imputed values of the different imputation methods for ALB.*

| ALB | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 36.00 | 37.00 | 4.84 | 12.00 | 53.00 |
| **Mean** | 36.00 | 36.00 | 0.00 | 36.00 | 36.00 |
| **Mean gender-age** | 36.04 | 36.07 | 0.53 | 34.41 | 36.65 |
| **Median** | 37.00 | 37.00 | 0.00 | 37.00 | 37.00 |
| **Median gender-age** | 36.63 | 37.00 | 0.6 | 34.00 | 37.00 |
| **Random Forest** | 38.29 | 38.78 | 2.42 | 25.00 | 42.00 |
| **pmm MI** | 36.22 | 37.00 | 4.73 | 12.00 | 53.00 |
| **pmm single** | 36.68 | 37.00 | 4.46 | 12.00 | 48.00 |
| **norm MI** | 36.18 | 37.00 | 4.78 | 11.95 | 54.82 |
| **norm single** | 36.54 | 36.62 | 4.68 | 15.18 | 54.01 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for ALB (Albumin). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 1.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ALB (Albumin).*



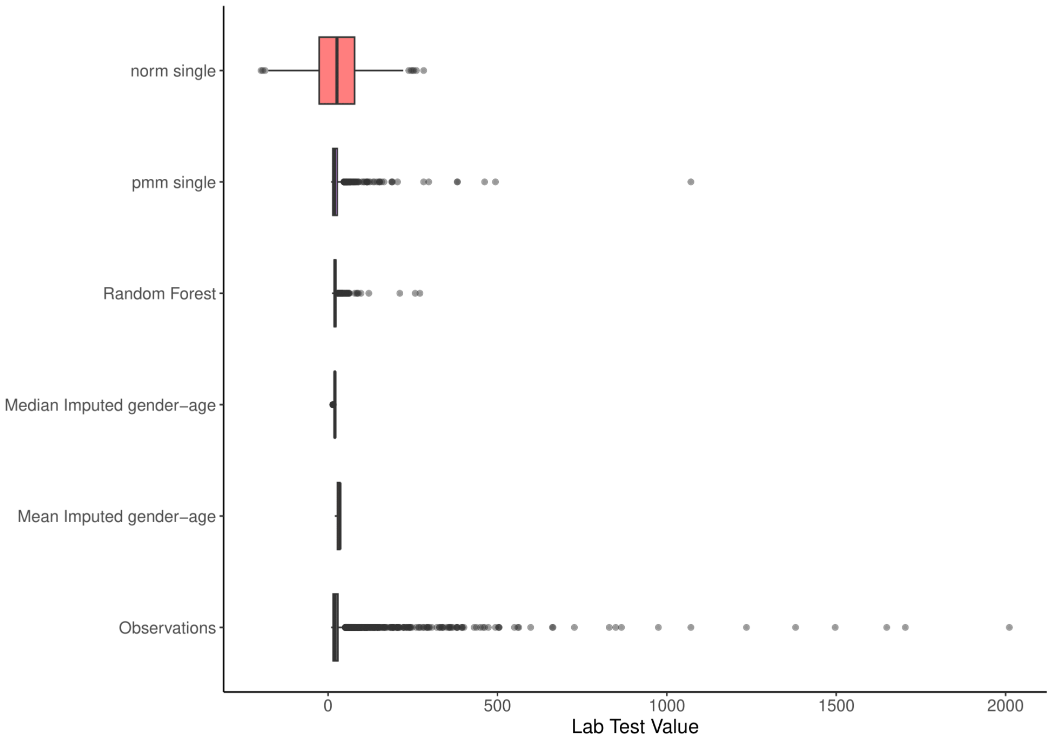

***Appendix Figure 2.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ALB (Albumin).*

## ALAT

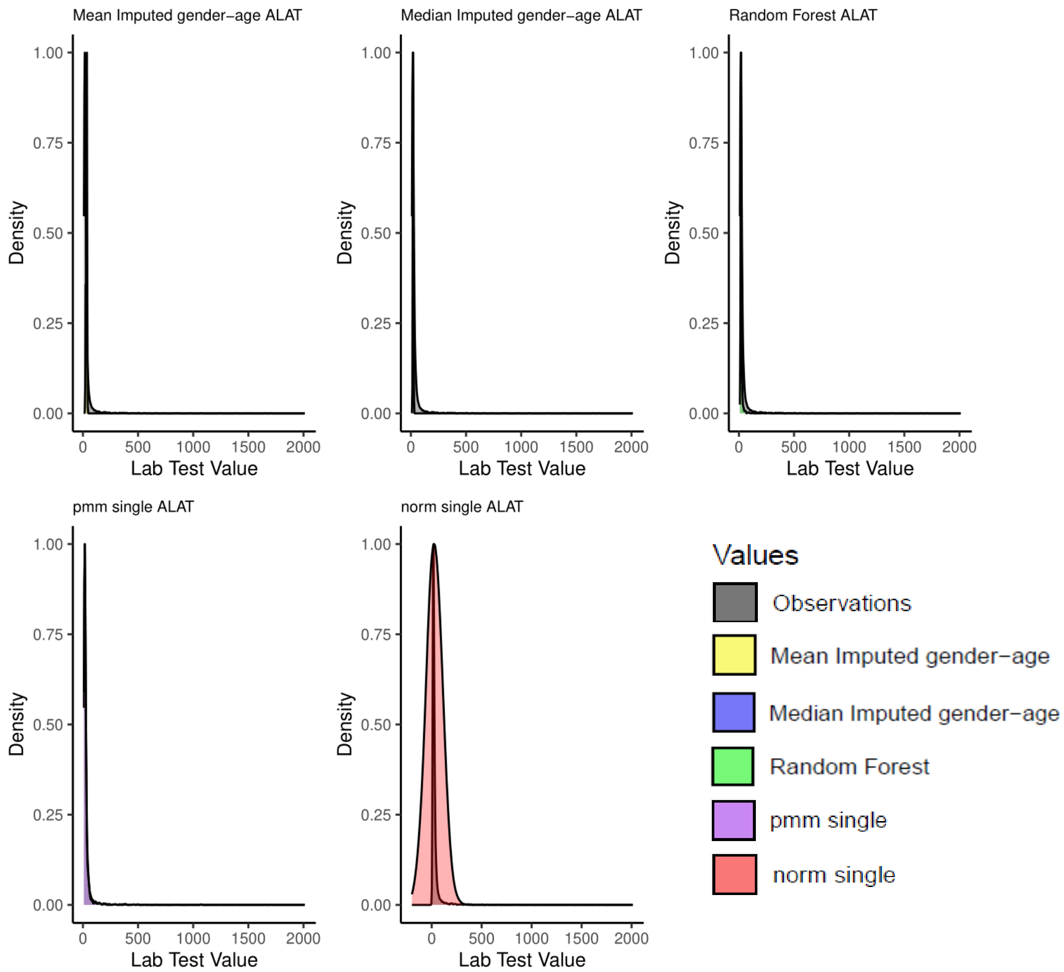
*Appendix Table 2. Statistics of imputed values of the different imputation methods for ALAT.*

| ALAT | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 33.03 | 20.00 | 76.52 | 9.00 | 2011.00 |
| **Mean** | 33.03 | 33.03 | 0.00 | 33.03 | 33.03 |
| **Mean gender-age** | 32.96 | 34.65 | 4.05 | 19.75 | 37.19 |
| **Median** | 20.00 | 20.00 | 0.00 | 20.00 | 20.00 |
| **Median gender-age** | 20.00 | 20.00 | 1.97 | 14.00 | 23.00 |
| **Random Forest** | 22.00 | 20.2 | 13.52 | 12.00 | 271.00 |
| **pmm MI** | 31.95 | 20.00 | 71.52 | 9.00 | 2011.00 |
| **pmm single** | 26.19 | 19.00 | 34.34 | 9.00 | 727.00 |
| **norm MI** | 32.19 | 20.00 | 76.43 | -288.15 | 2011.00 |
| **norm single** | 29.06 | 30.73 | 75.71 | -248.91 | 258.27 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for ALAT (Alanine aminotransferase). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 3.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ALAT (Alanine aminotransferase).*



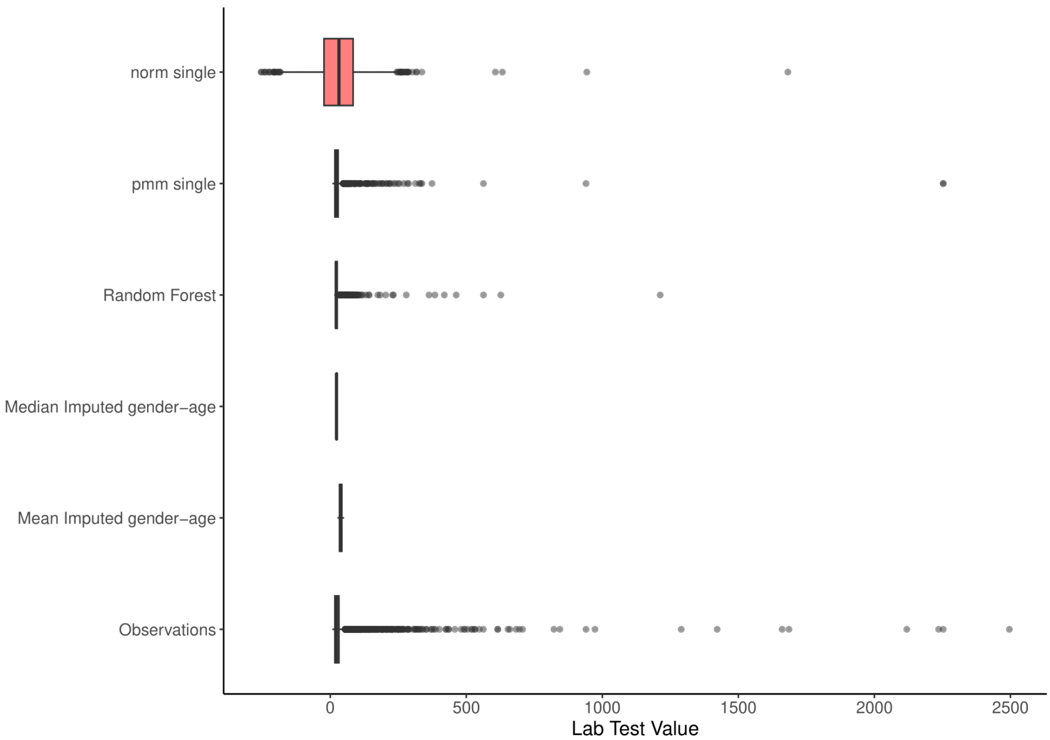

***Appendix Figure 4.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ALAT (Alanine aminotransferase).*

## ASAT

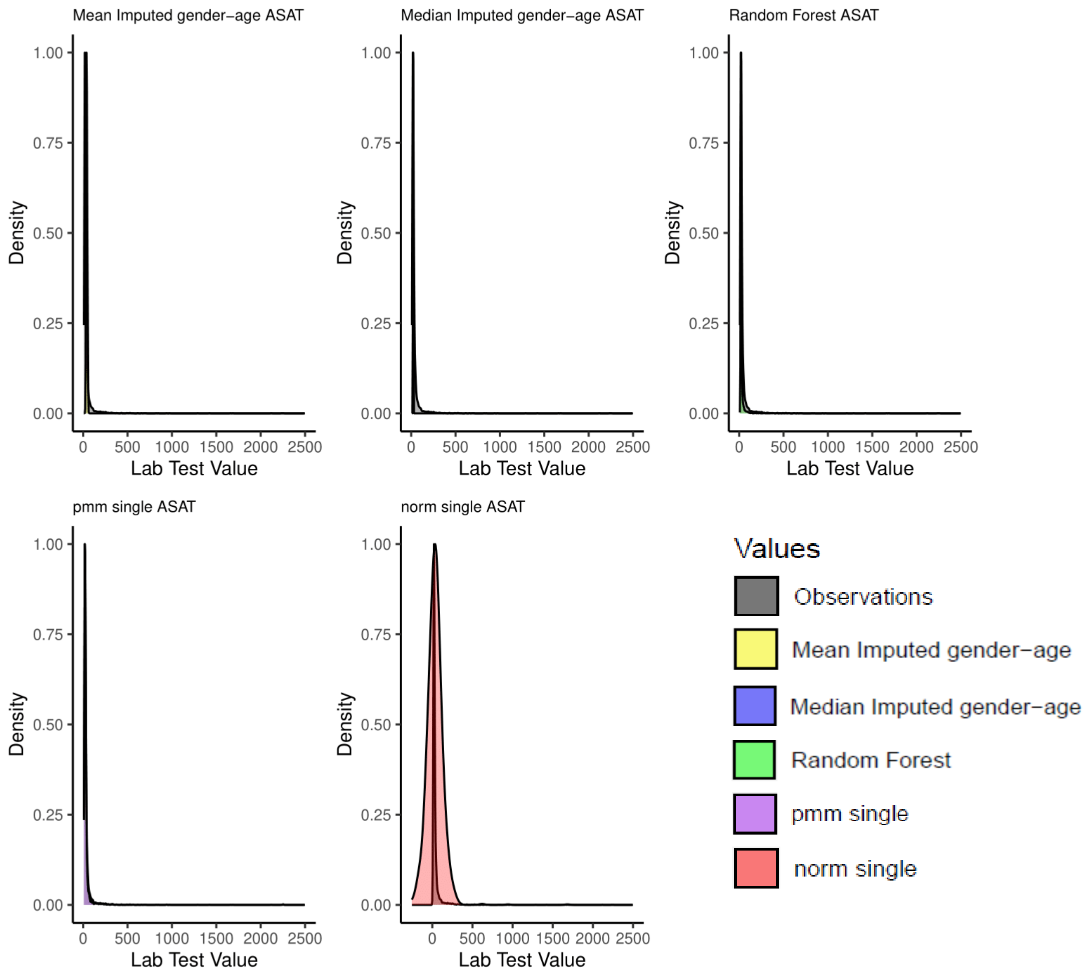
*Appendix Table 3. Statistics of imputed values of the different imputation methods for ASAT.*

| ASAT | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 39.69 | 23.00 | 100.16 | 8.00 | 2496.00 |
| **Mean** | 39.69 | 39.69 | 0.00 | 39.69 | 39.69 |
| **Mean gender-age** | 39.68 | 39.46 | 5.82 | 26.35 | 50.76 |
| **Median** | 23.00 | 23.00 | 0.00 | 23.00 | 23.00 |
| **Median gender-age** | 23.04 | 23.00 | 0.73 | 22.00 | 25.00 |
| **Random Forest** | 27.58 | 21.96 | 40.6 | 14.00 | 1213.00 |
| **pmm MI** | 37.26 | 23.00 | 92.05 | 8.00 | 2496.00 |
| **pmm single** | 10.77 | 22.00 | 61.8 | 8.00 | 2119.00 |
| **norm MI** | 37.68 | 23.00 | 99.69 | -413.76 | 2496.00 |
| **norm single** | 30.95 | 29.08 | 97.05 | -263.38 | 1746.43 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for ASAT (Aspartate aminotransferase). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 5****. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ASAT (Aspartate aminotransferase).*



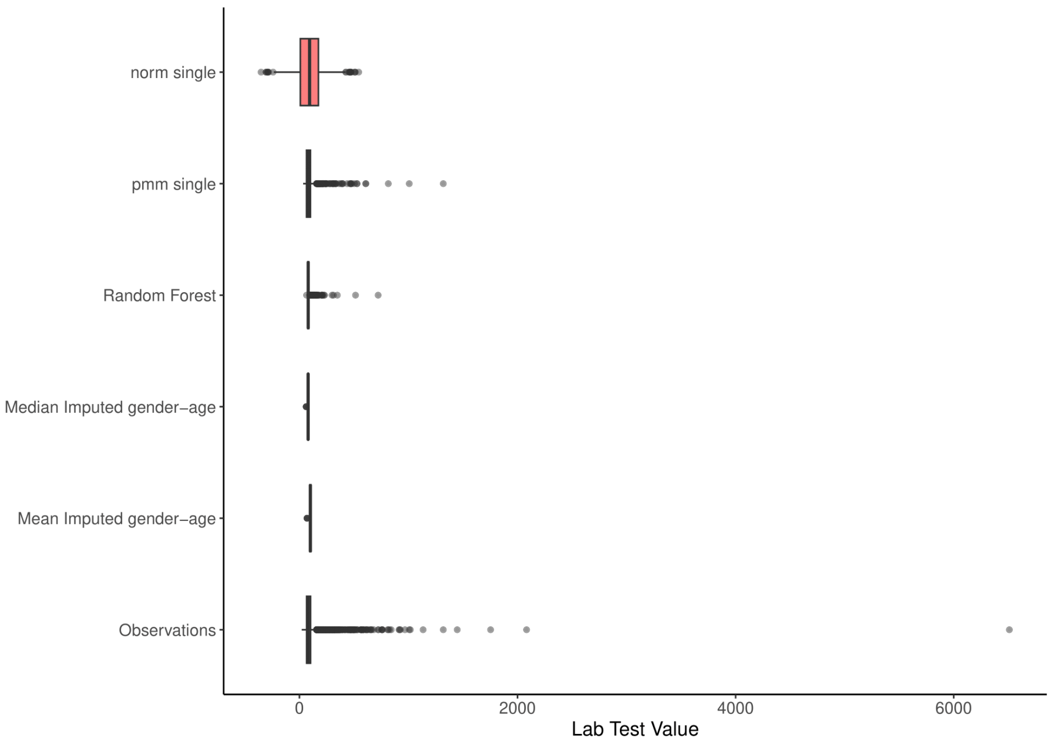

***Appendix Figure 6.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ASAT (Aspartate aminotransferase).*

## ALP

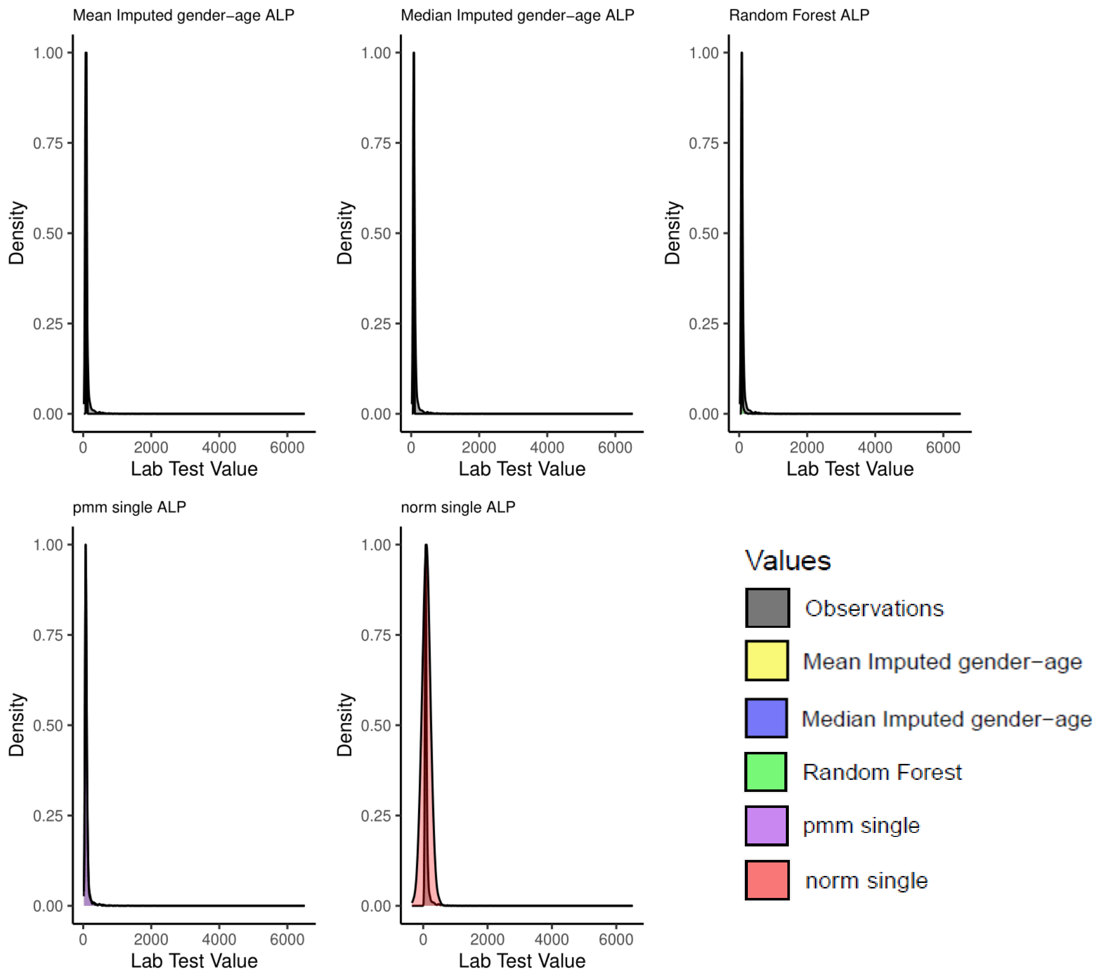
*Appendix Table 4. Statistics of imputed values of the different imputation methods for ALP.*

| ALP | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 101.08 | 80.00 | 132.05 | 21.00 | 6510.00 |
| Mean | 101.08 | 101.08 | 0.00 | 101.08 | 101.08 |
| **Mean gender-age** | 100.63 | 100.6 | 5.72 | 69.67 | 111.98 |
| **Median** | 80.00 | 80.00 | 0.00 | 80.00 | 80.00 |
| Median gender-age | 80.3 | 80.00 | 2.85 | 62.00 | 87.00 |
| Random Forest | 84.09 | 80.66 | 25.37 | 65.00 | 722.00 |
| pmm MI | 98.42 | 80.00 | 119.96 | 21.00 | 6510.00 |
| pmm single | 92.02 | 79.00 | 68.36 | 21.00 | 1007.00 |
| norm MI | 98.94 | 81.00 | 131.45 | -399.85 | 6510.00 |
| norm single | 92.9 | 95.49 | 131.25 | -360.68 | 529.82 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for ALP (Alkaline Phosphatase). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 7***. *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-age, Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ALP (Alkaline Phosphatase).*



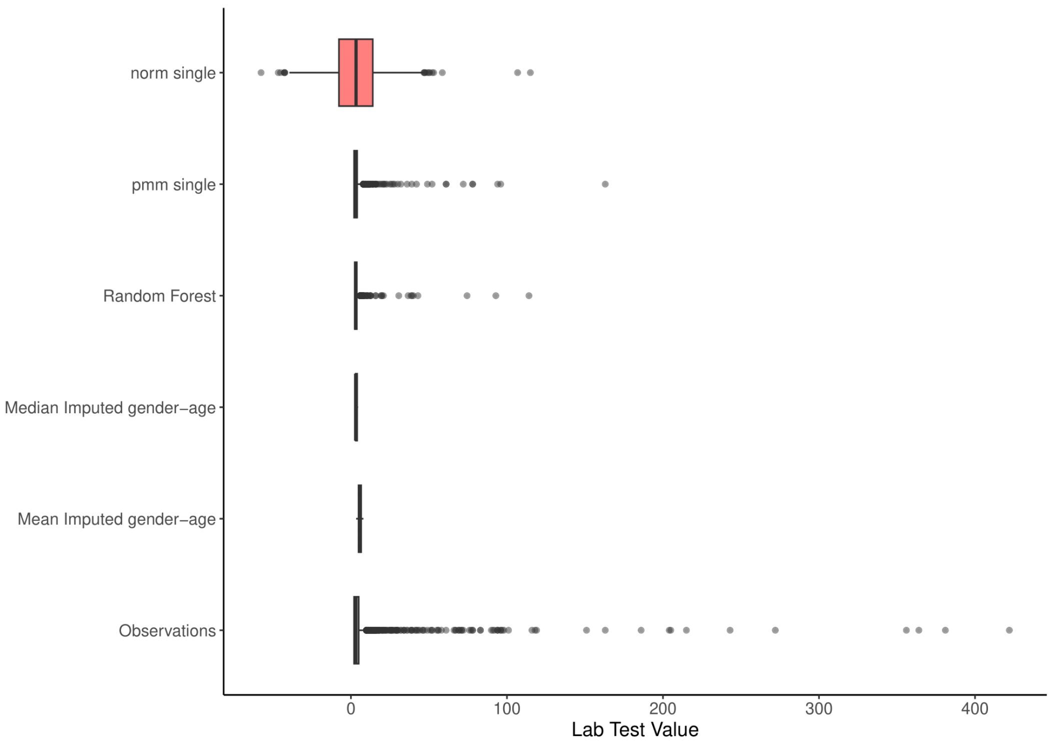

***Appendix Figure 7.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for ALP (Alkaline Phosphatase)*

## BILID

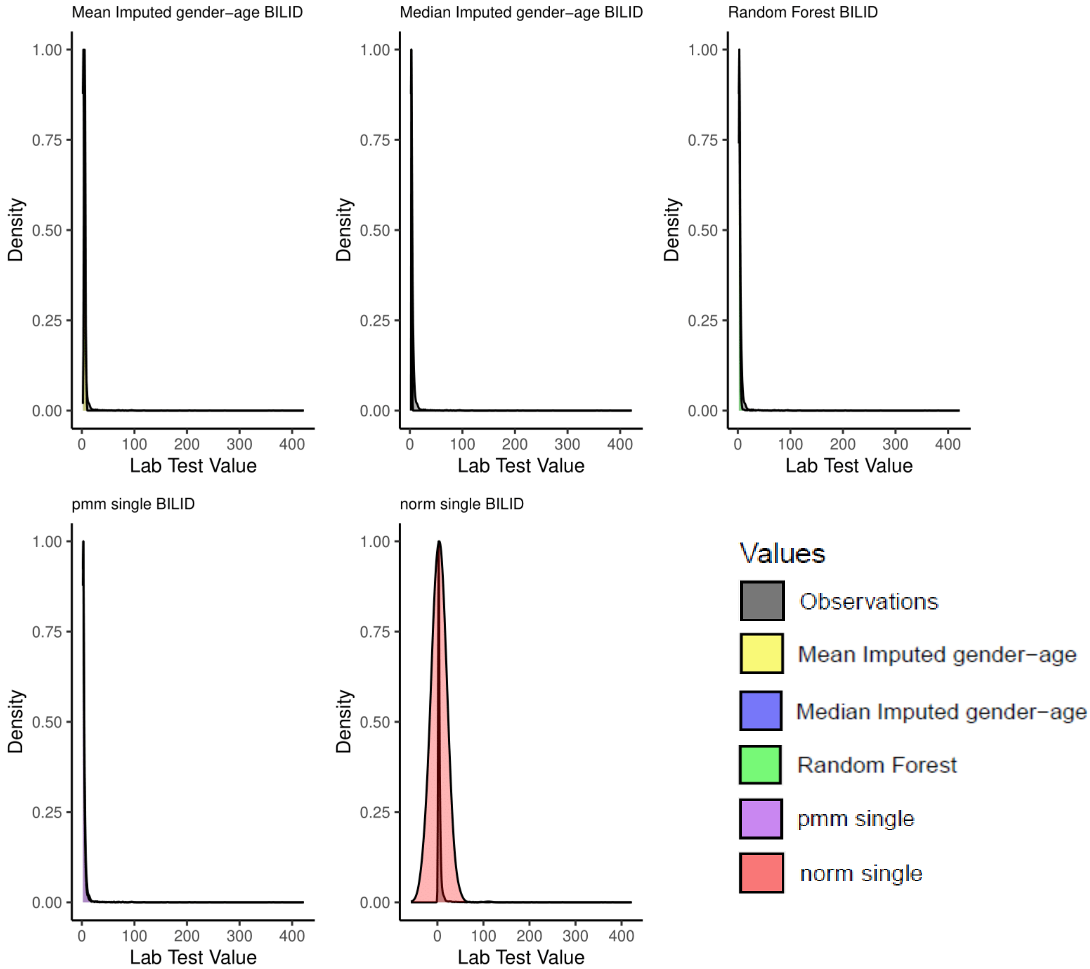
*Appendix Table 5. Statistics of imputed values of the different imputation methods for BILID.*

| BILID | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 5.6 | 3.00 | 16.72 | 2.00 | 422.00 |
| **Mean** | 5.6 | 5.6 | 0.00 | 5.6 | 5.6 |
| **Mean gender-age** | 5.48 | 5.53 | 1.2 | 3.33 | 7.99 |
| **Median** | 3.00 | 3.00 | 0.00 | 3.00 | 3.00 |
| **Median gender-age** | 3.28 | 3.00 | 0.46 | 3.00 | 4.5 |
| **Random Forest** | 3.52 | 2.9 | 4.21 | 2.00 | 114.1 |
| **pmm MI** | 5.16 | 3.00 | 14.66 | 2.00 | 422.00 |
| **pmm single** | 4.53 | 3.00 | 9.23 | 2.00 | 204.00 |
| **norm MI** | 4.94 | 3.00 | 16.64 | -58.64 | 422.00 |
| **norm single** | 3.38 | 3.29 | 16.68 | -52.91 | 93.21 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for BILID (Direct Bilirubin). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 7.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for BILID (Direct bilirubin).*



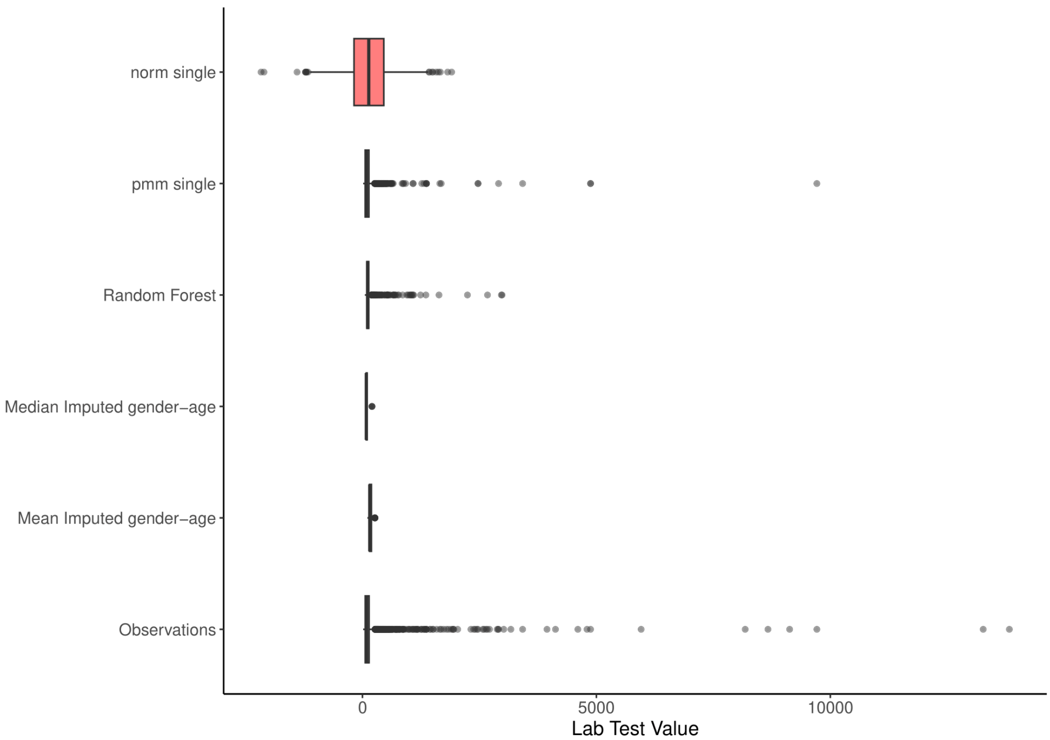

***Appendix Figure 8.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for BILID (Direct bilirubin).*

CK

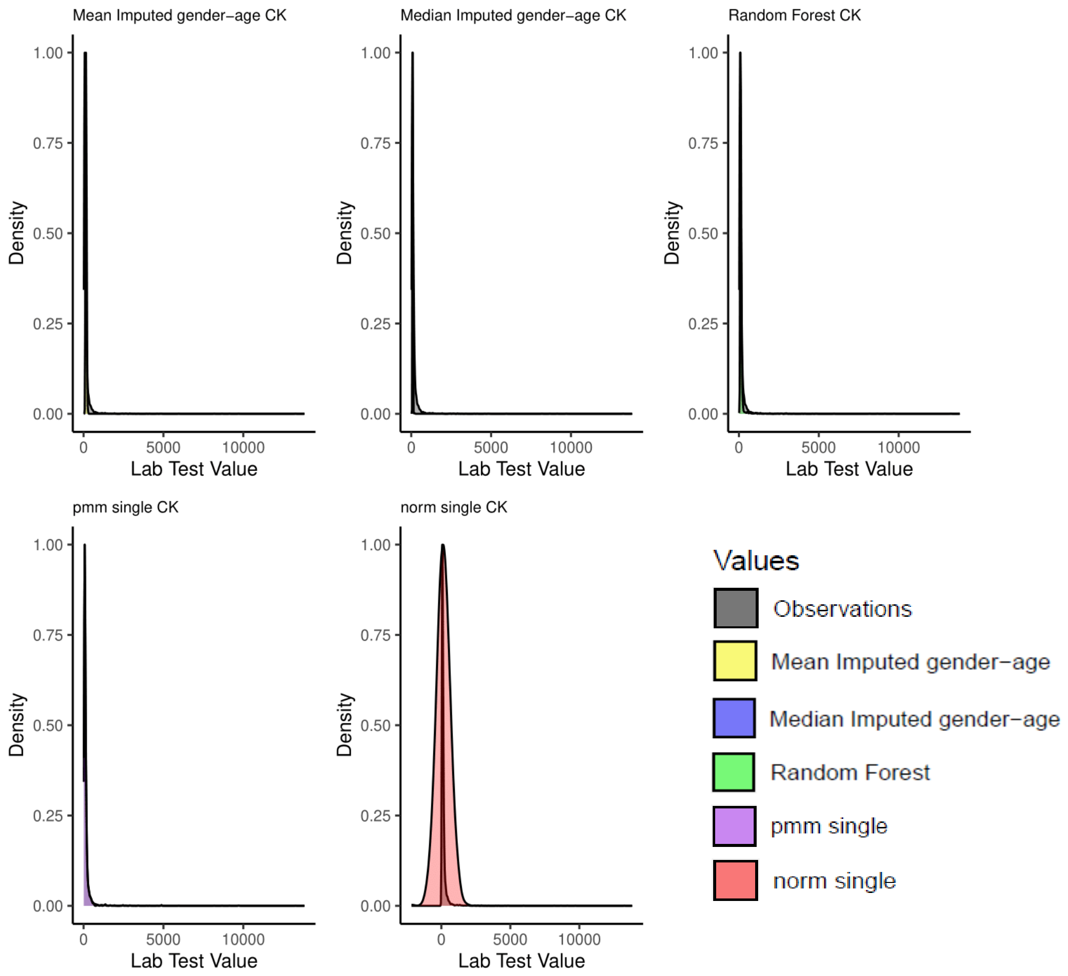
*Appendix Table 6. Statistics of imputed values of the different imputation methods for CK.*

| CK | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 161.3 | 89.00 | 485.5 | 15.00 | 29403.00 |
| **Mean** | 161.3 | 161.3 | 0.00 | 161.3 | 161.3 |
| **Mean gender-age** | 160.36 | 155.53 | 30.26 | 105.42 | 266.00 |
| **Median** | 89.00 | 89.00 | 0.00 | 89.00 | 89.00 |
| **Median gender-age** | 89.67 | 81.5 | 11.85 | 74.00 | 203.00 |
| **Random Forest** | 135.34 | 112.2 | 154.93 | 50.93 | 2983.59 |
| **pmm MI** | 152.13 | 87.00 | 439.62 | 15.00 | 13826.00 |
| **pmm single** | 119.89 | 82.00 | 186.76 | 73.00 | 2909.00 |
| **norm MI** | 152.75 | 91.00 | 488.92 | -4137.63 | 13826.00 |
| **norm single** | 127.17 | 130.62 | 499.23 | -2175.3 | 1802.82 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for CK (Creatine kinase). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 9.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for CK (Creatine kinase).*



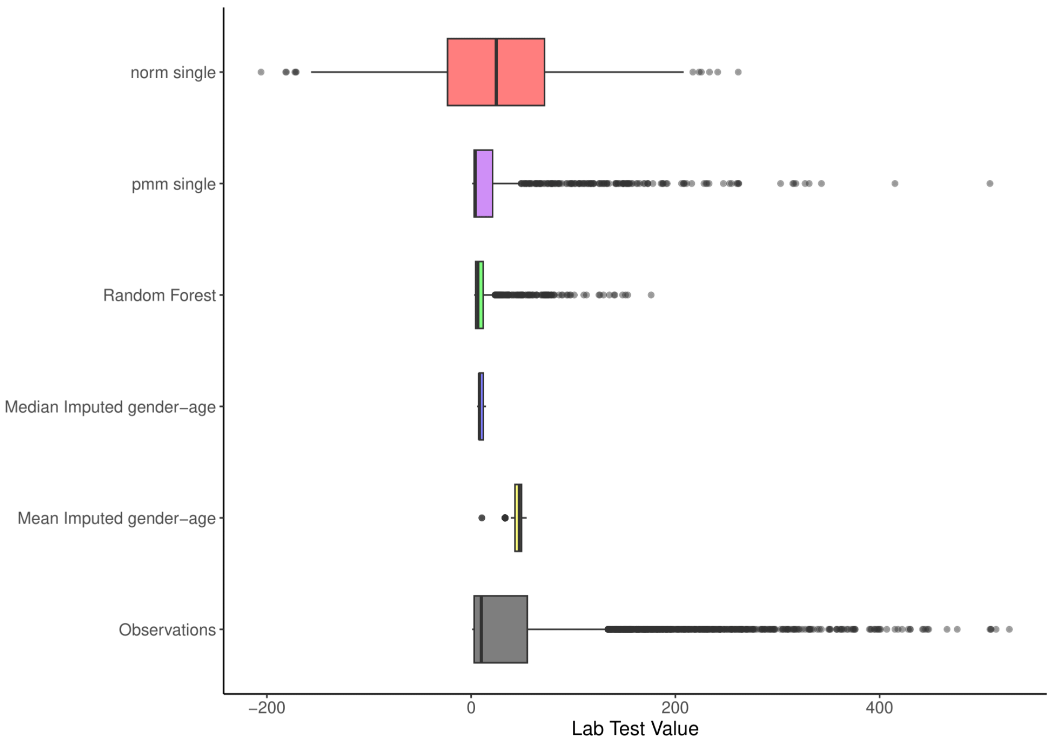

***Appendix Figure 10.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for CK (Creatine kinase).*

## CRP

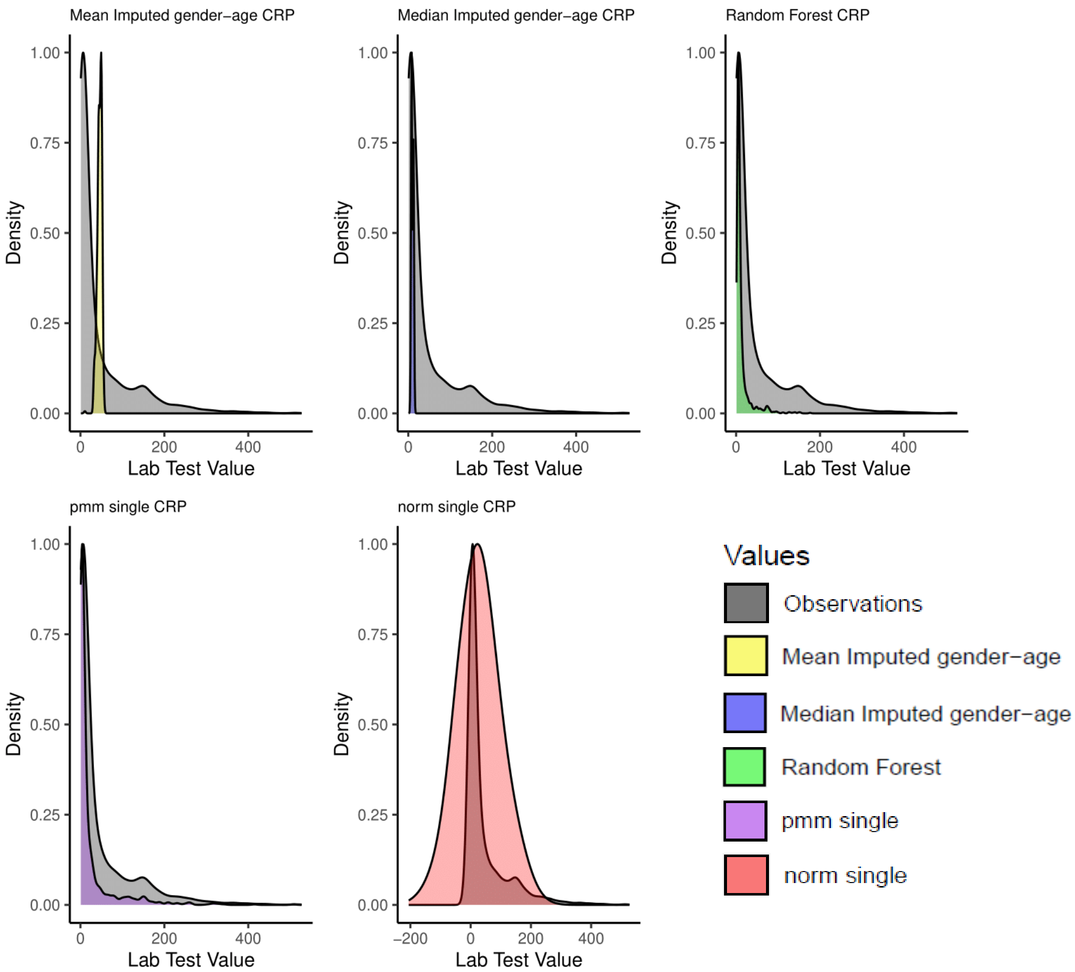
*Appendix Table 7. Statistics of imputed values of the different imputation methods for CRP.*

| CRP | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 45.92 | 10.00 | 73.86 | 1.00 | 527.00 |
| **Mean** | 45.92 | 45.92 | 0.00 | 45.92 | 45.92 |
| **Mean gender-age** | 45.43 | 47.13 | 5.35 | 10.5 | 54.34 |
| **Median** | 10.00 | 10.00 | 0.00 | 10.00 | 10.00 |
| **Median gender-age** | 9.89 | 8.00 | 2.36 | 6.00 | 14.5 |
| **Random Forest** | 13.21 | 6.58 | 19.78 | 2.9 | 176.00 |
| **pmm MI** | 41.95 | 8.3 | 70.65 | 1.00 | 527.00 |
| **pmm single** | 25.78 | 5.00 | 51.52 | 1.00 | 514.00 |
| **norm MI** | 41.81 | 11.00 | 73.88 | -291.36 | 527.00 |
| **norm single** | 24.57 | 21.26 | 71.42 | -187.55 | 288.36 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for CRP (C-reactive protein). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 11.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for CRP (C-reactive protein).*



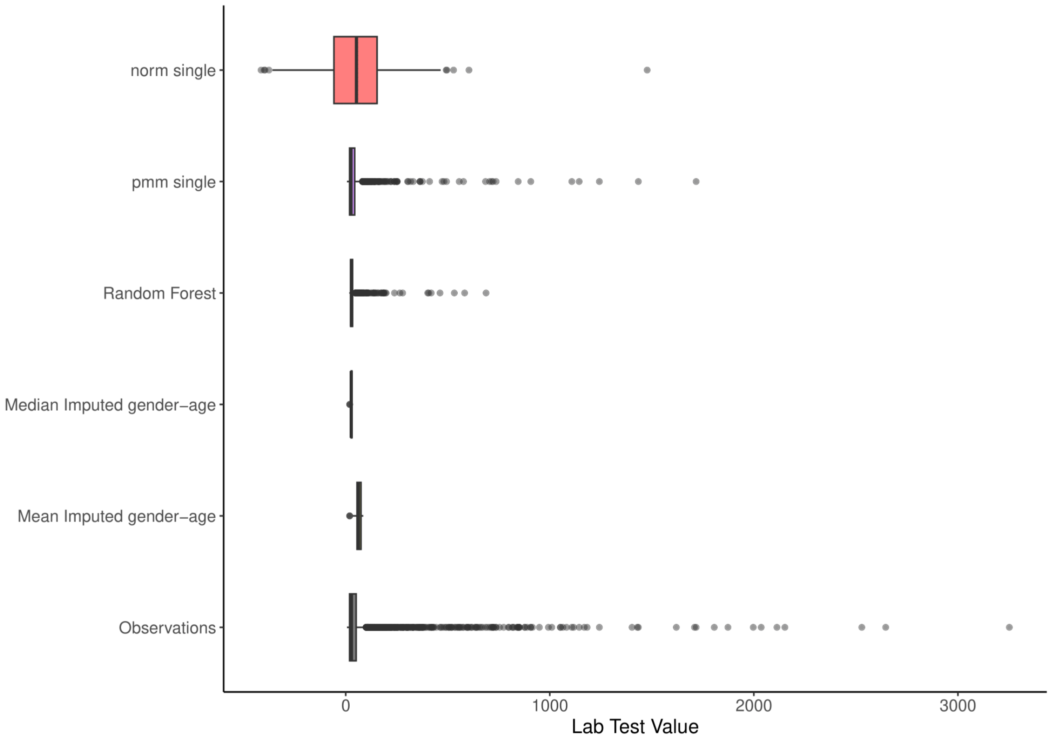

***Appendix Figure 12***. *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for CRP (C-reactive protein).*

## GGT

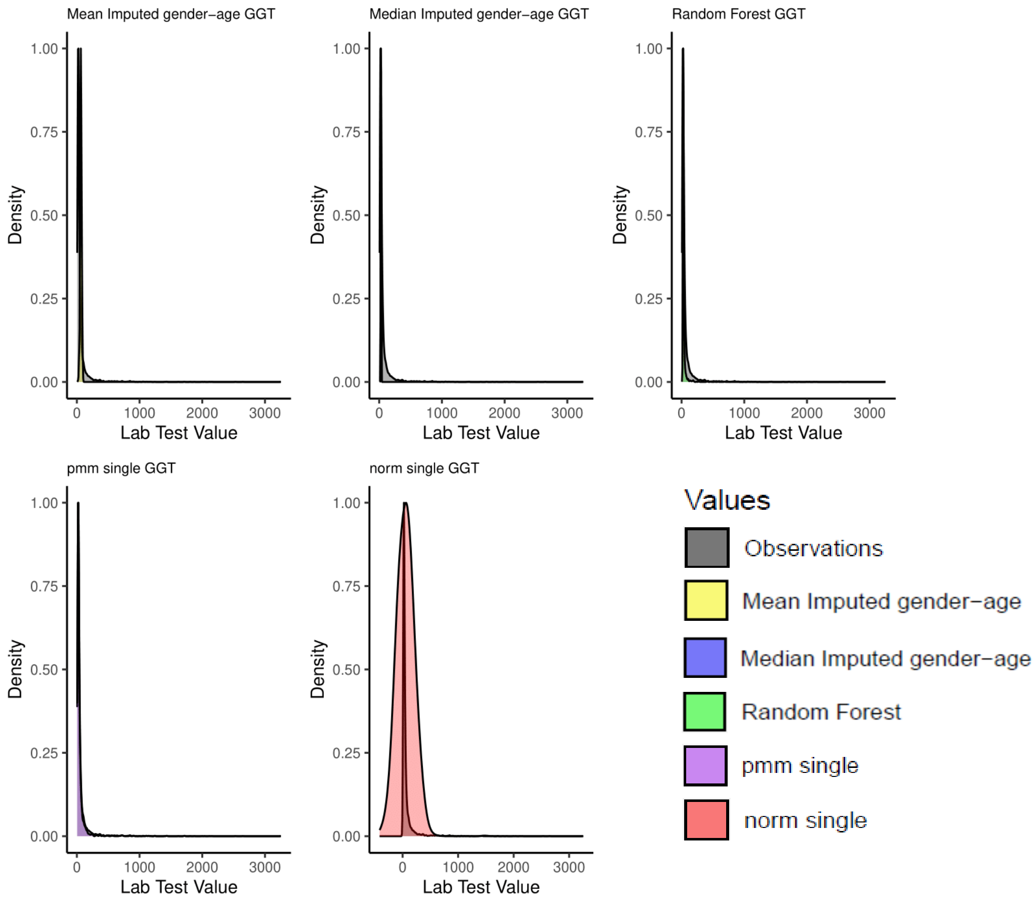
*Appendix Table 8. Statistics of imputed values of the different imputation methods for GGT.*

| GGT | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 65.65 | 28.00 | 156.93 | 6.00 | 3252.00 |
| **Mean** | 65.65 | 65.65 | 0.00 | 65.65 | 65.65 |
| **Mean gender-age** | 64.7 | 63.1 | 11.41 | 19.25 | 86.09 |
| **Median** | 28.00 | 28.00 | 0.00 | 28.00 | 28.00 |
| **Median gender-age** | 28.26 | 27.00 | 2.52 | 19.5 | 32.00 |
| **Random Forest** | 35.27 | 28.43 | 38.02 | 18.00 | 688.00 |
| **pmm MI** | 61.93 | 28.00 | 147.63 | 6.00 | 3252.00 |
| **pmm single** | 51.52 | 27.00 | 125.51 | 7.00 | 3252.00 |
| **norm MI** | 62.65 | 30.00 | 157.44 | -574.62 | 3252.00 |
| **norm single** | 55.15 | 52.44 | 160.64 | -484.00 | 1489.23 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for GGT (Gamma glutamyl transferase). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 13.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for GGT (Gamma glutamyl transferase).*



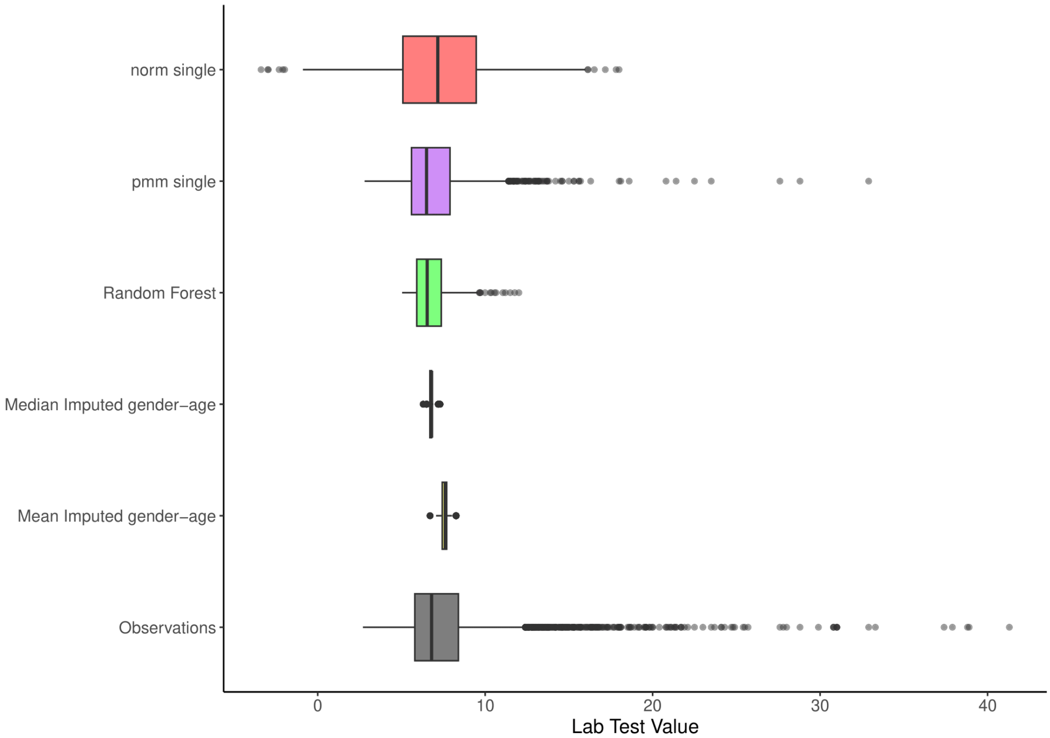

***Appendix Figure 14.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for GGT (Gamma glutamyl transferase).*

GLC

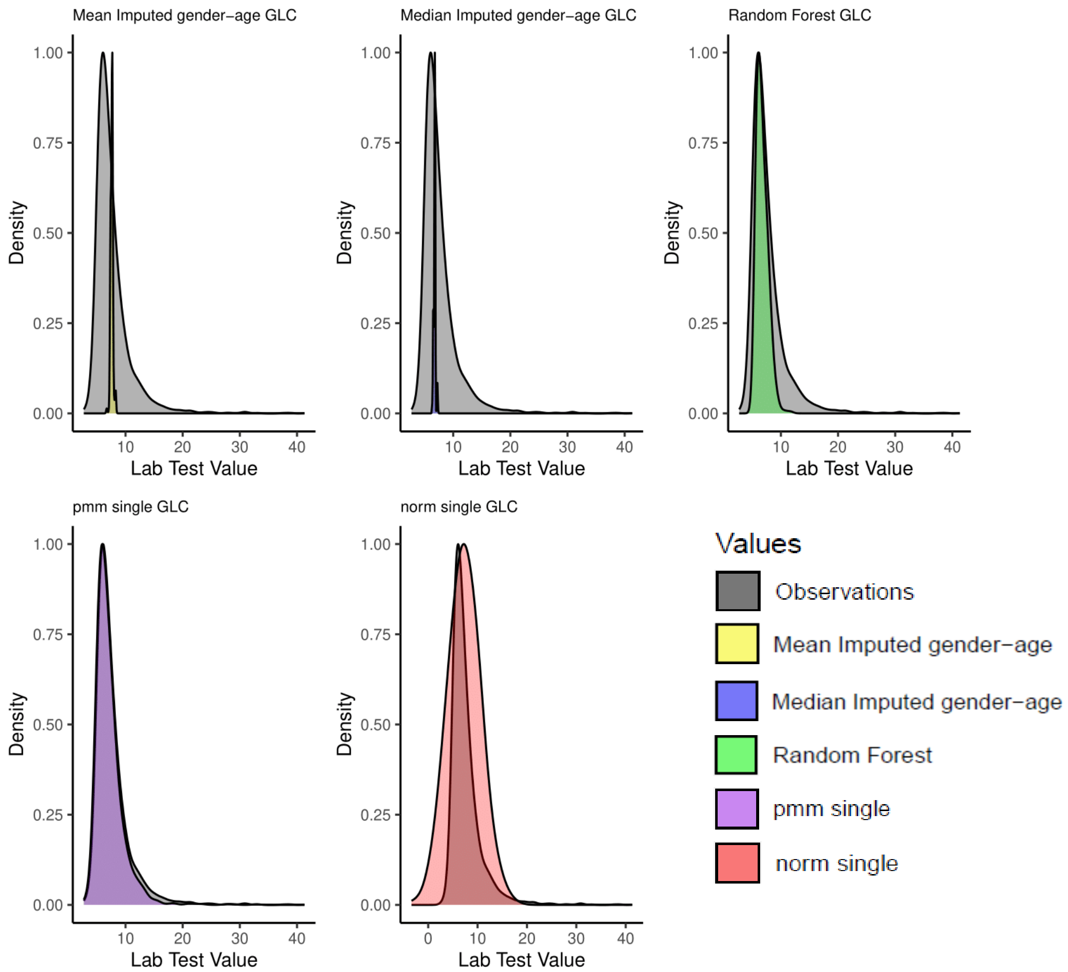
*Appendix Table 9. Statistics of imputed values of the different imputation methods for GLC.*

| GLC | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 7.61 | 6.8 | 3.16 | 2.7 | 41.3 |
| **Mean** | 7.61 | 7.61 | 0.00 | 7.61 | 7.61 |
| **Mean gender-age** | 7.6 | 7.62 | 0.21 | 6.71 | 8.26 |
| **Median** | 6.8 | 6.8 | 0.00 | 6.8 | 6.8 |
| **Median gender-age** | 6.75 | 6.8 | 0.17 | 6.3 | 7.3 |
| **Random Forest** | 6.73 | 6.54 | 1.00 | 5.00 | 12.00 |
| **pmm MI** | 7.52 | 6.7 | 3.05 | 2.7 | 41.3 |
| **pmm single** | 7.15 | 6.6 | 2.43 | 2.7 | 24.7 |
| **norm MI** | 7.53 | 6.8 | 3.17 | -6.4 | 41.3 |
| **norm single** | 7.21 | 7.17 | 3.12 | -2.91 | 16.25 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for GLC (Glucose). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 15.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for GLC (Glucose).*



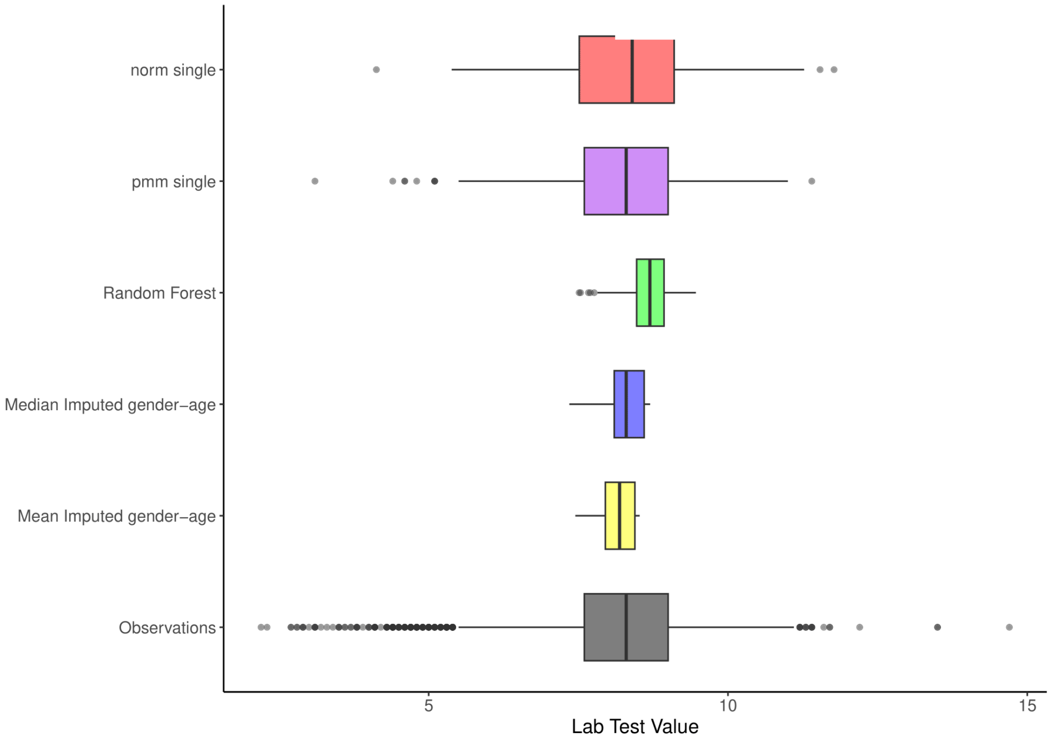

***Appendix Figure 16.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for GLC (Glucose).*

## Hb

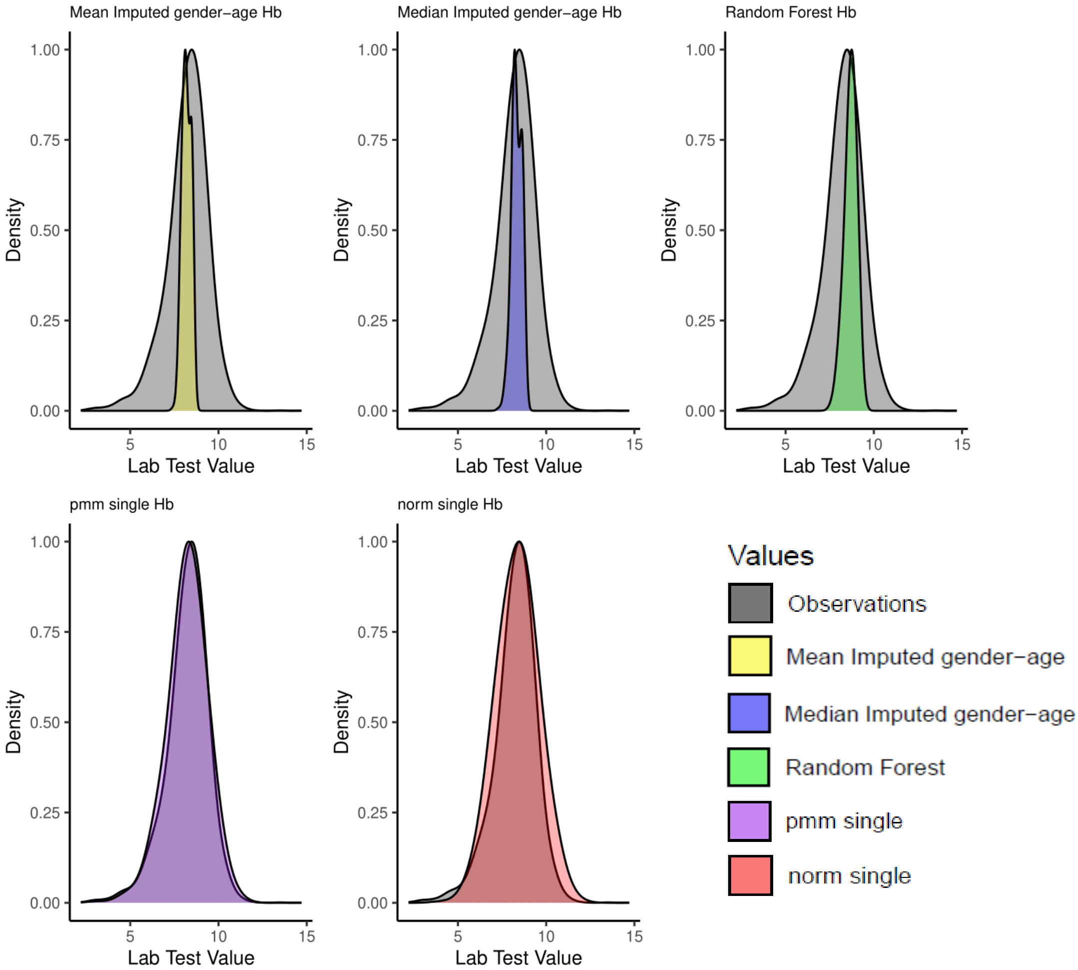
*Appendix Table 10. Statistics of imputed values of the different imputation methods for Hb.*

| Hb | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 8.21 | 8.3 | 1.19 | 2.2 | 14.7 |
| **Mean** | 8.21 | 8.21 | 0.00 | 8.21 | 8.21 |
| **Mean gender-age** | 8.19 | 8.2 | 0.24 | 7.45 | 8.52 |
| **Median** | 8.3 | 8.3 | 0.00 | 8.3 | 8.3 |
| **Median gender-age** | 8.33 | 8.3 | 0.27 | 7.35 | 8.7 |
| **Random Forest** | 8.68 | 8.69 | 0.35 | 7.5 | 9.5 |
| **pmm MI** | 8.22 | 8.3 | 1.19 | 2.2 | 14.7 |
| **pmm single** | 8.25 | 8.4 | 1.17 | 2.8 | 12.2 |
| **norm MI** | 8.22 | 8.3 | 1.19 | 2.2 | 14.7 |
| **norm single** | 8.25 | 8.29 | 1.23 | 2.4 | 12.7 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for Hb (Hemoglobin). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 17.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Hb (Hemoglobin).*



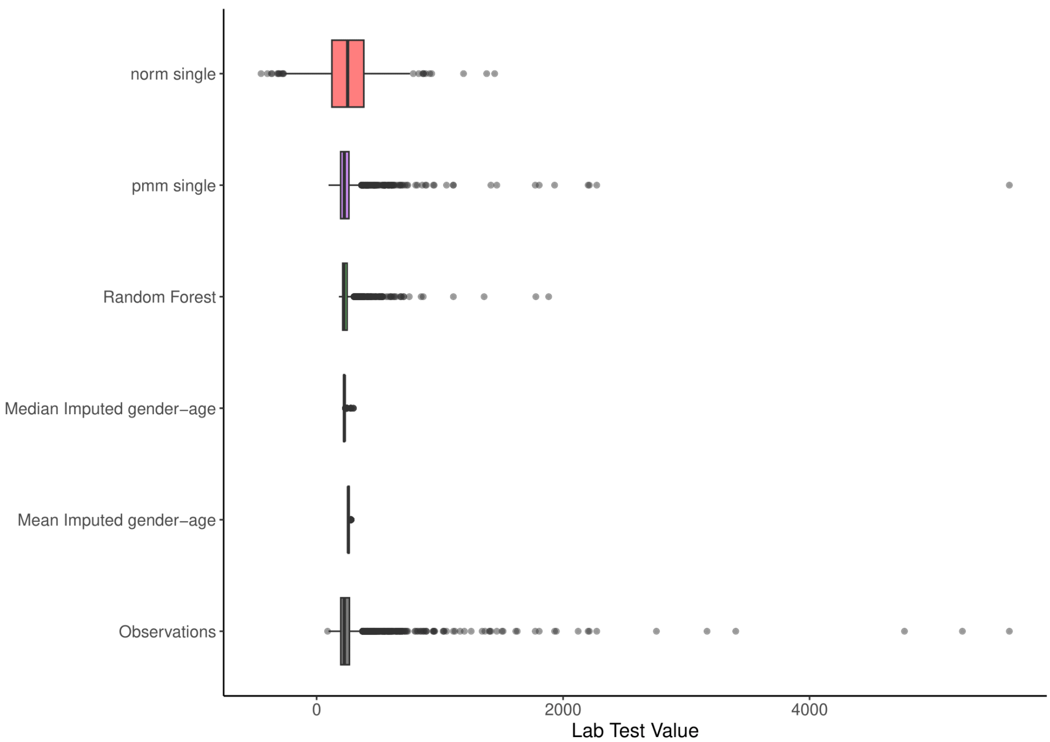

***Appendix Figure 18.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Hb (Hemoglobin).*

## LDH

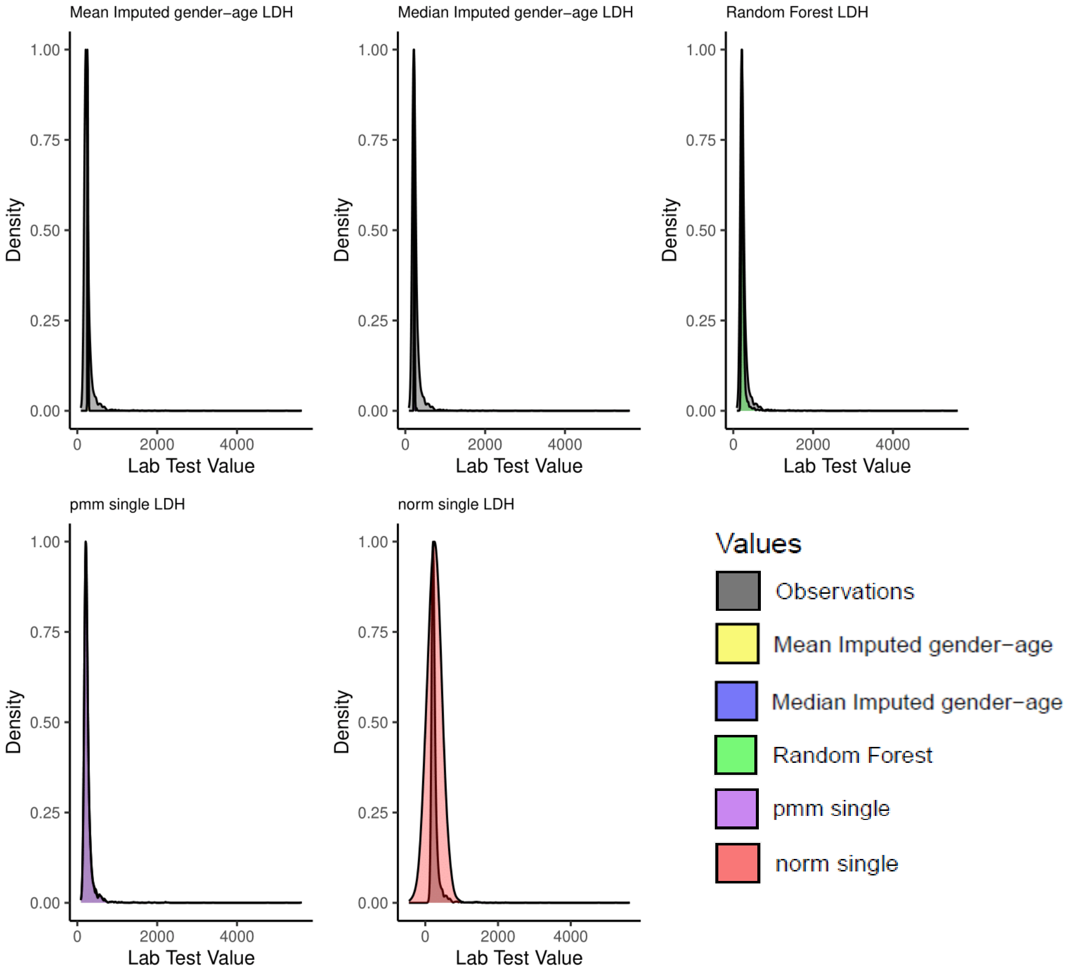
*Appendix Table 11. Statistics of imputed values of the different imputation methods for LDH.*

| LDH | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 261.38 | 225.00 | 209.63 | 77.00 | 5621.00 |
| **Mean** | 261.38 | 261.38 | 0.00 | 261.38 | 261.38 |
| **Mean gender-age** | 261.03 | 257.25 | 7.43 | 252.8 | 279.3 |
| **Median** | 225.00 | 225.00 | 0.00 | 225.00 | 225.00 |
| **Median gender-age** | 224.96 | 225.5 | 7.41 | 220.00 | 298.00 |
| **Random Forest** | 243.54 | 223.31 | 83.48 | 181.00 | 1883.00 |
| **pmm MI** | 256.44 | 224.00 | 189.5 | 89.00 | 5621.00 |
| **pmm single** | 250.4 | 223.00 | 173.16 | 97.00 | 5621.00 |
| **norm MI** | 257.53 | 228.00 | 207.63 | -554.96 | 5621.00 |
| **norm single** | 246.58 | 245.67 | 206.41 | -539.96 | 1797.88 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for LDH (Lactate dehydrogenase). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 19****. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for LDH (Lactate dehydrogenase).*



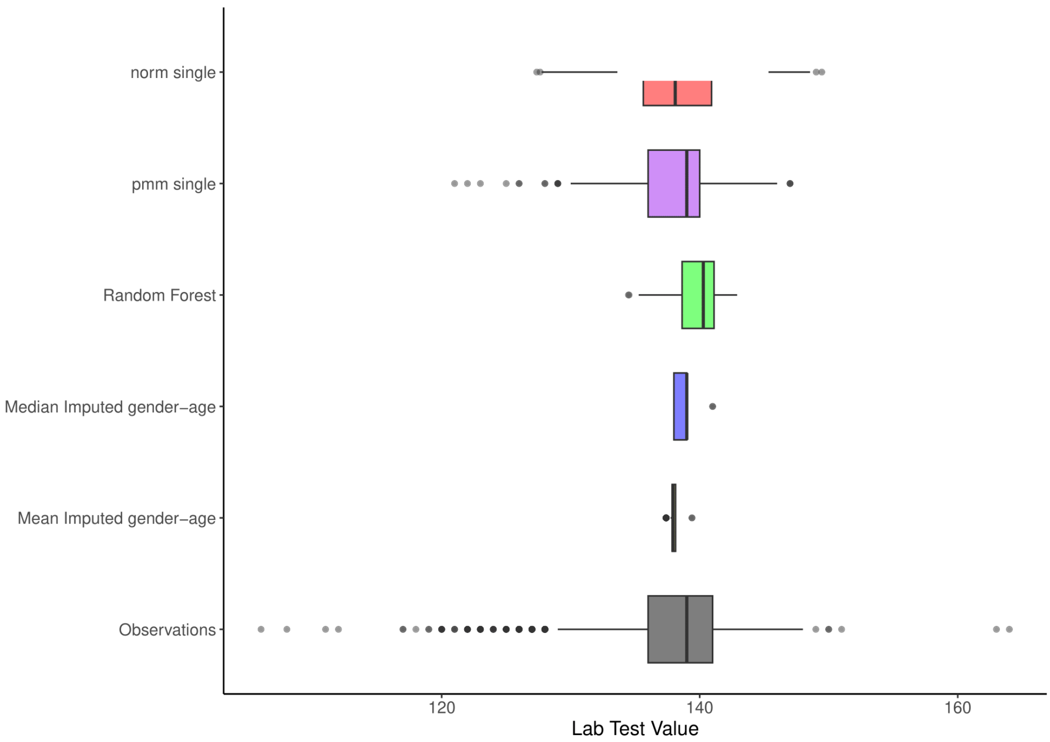

***Appendix Figure 20.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for LDH (Lactate dehydrogenase).*

Na

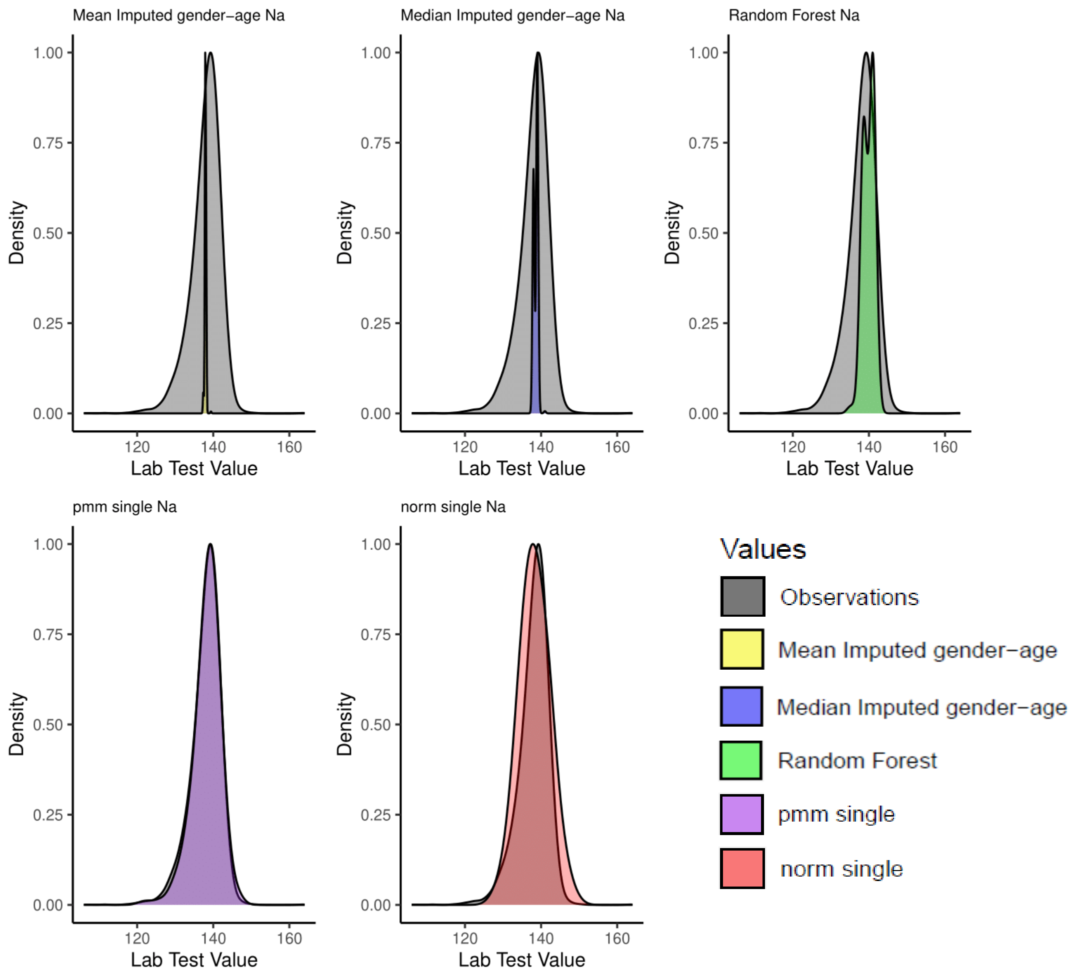
*Appendix Table 12. Statistics of imputed values of the different imputation methods for Na.*

| Na | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 137.95 | 139.00 | 3.92 | 106.00 | 164.00 |
| **Mean** | 137.95 | 137.95 | 0.00 | 137.95 | 137.95 |
| **Mean gender-age** | 137.95 | 137.94 | 0.19 | 137.4 | 139.4 |
| **Median** | 139.00 | 139.00 | 0.00 | 139.00 | 139.00 |
| **Median gender-age** | 138.61 | 139.00 | 0.51 | 138.00 | 141.00 |
| **Random Forest** | 139.89 | 140.3 | 1.45 | 134.00 | 143.00 |
| **pmm MI** | 137.98 | 139.00 | 3.91 | 106.00 | 164.00 |
| **pmm single** | 138.33 | 139.00 | 3.78 | 121.00 | 148.00 |
| **norm MI** | 137.97 | 139.00 | 3.93 | 106.00 | 164.00 |
| **norm single** | 138.29 | 137.98 | 4.01 | 126.69 | 149.71 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for Na (Sodium). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 21**. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Na (Sodium).*



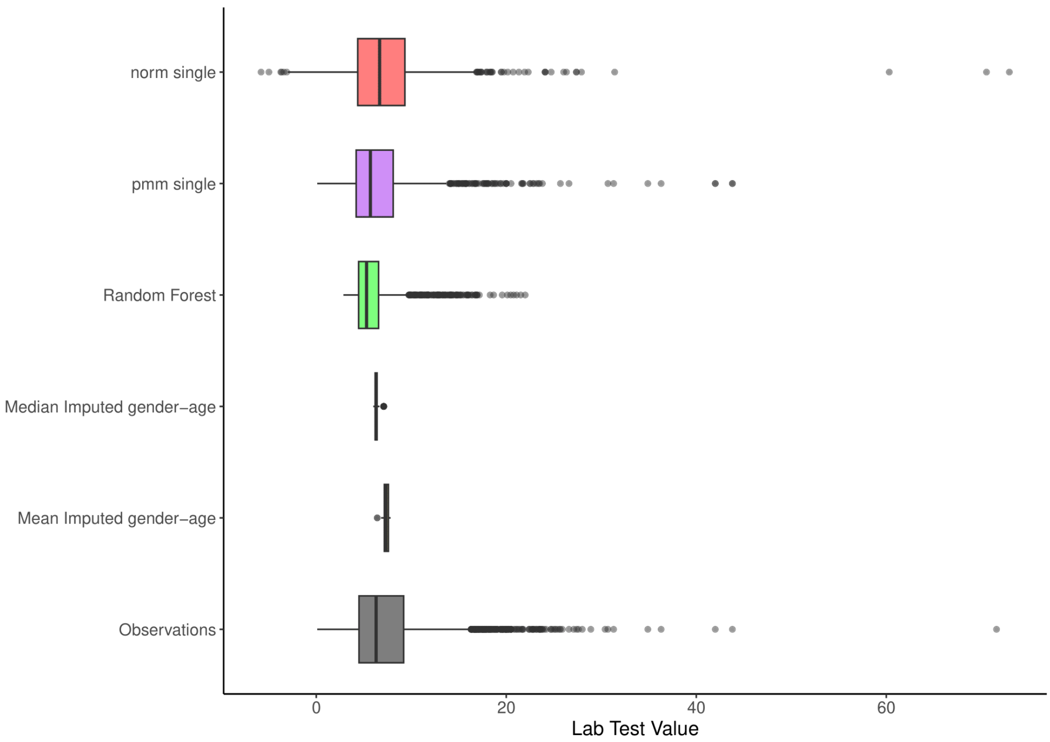

***Appendix Figure 22****. Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Na (Sodium).*

## Neut

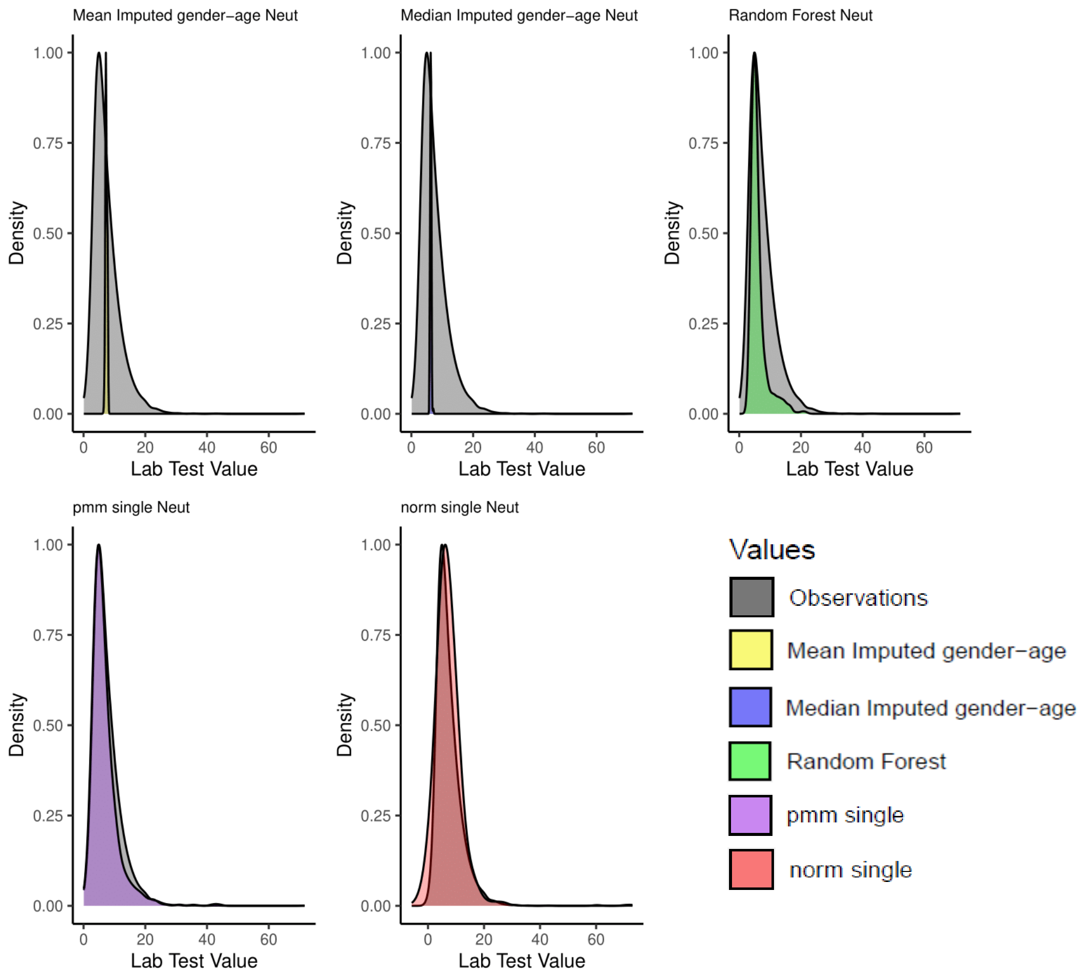
*Appendix Table 13. Statistics of imputed values of the different imputation methods for Neut.*

| Neut | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 7.39 | 6.3 | 4.29 | 0.1 | 71.6 |
| **Mean** | 7.39 | 7.39 | 0.00 | 7.39 | 7.39 |
| **Mean gender-age** | 7.37 | 7.33 | 0.28 | 6.42 | 7.82 |
| **Median** | 6.3 | 6.3 | 0.00 | 6.3 | 6.3 |
| **Median gender-age** | 6.28 | 6.3 | 0.19 | 6.00 | 7.1 |
| **Random Forest** | 6.11 | 5.3 | 2.77 | 2.9 | 22.00 |
| **pmm MI** | 7.25 | 6.1 | 4.32 | 0.1 | 71.6 |
| **pmm single** | 6.63 | 5.5 | 4.32 | 0.1 | 43.8 |
| **norm MI** | 7.32 | 6.4 | 4.51 | -9.14 | 78.18 |
| **norm single** | 6.97 | 6.47 | 5.34 | -6.48 | 77.68 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for Neut (Neutrophil). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 23***. *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Neut (Neutrophil).*



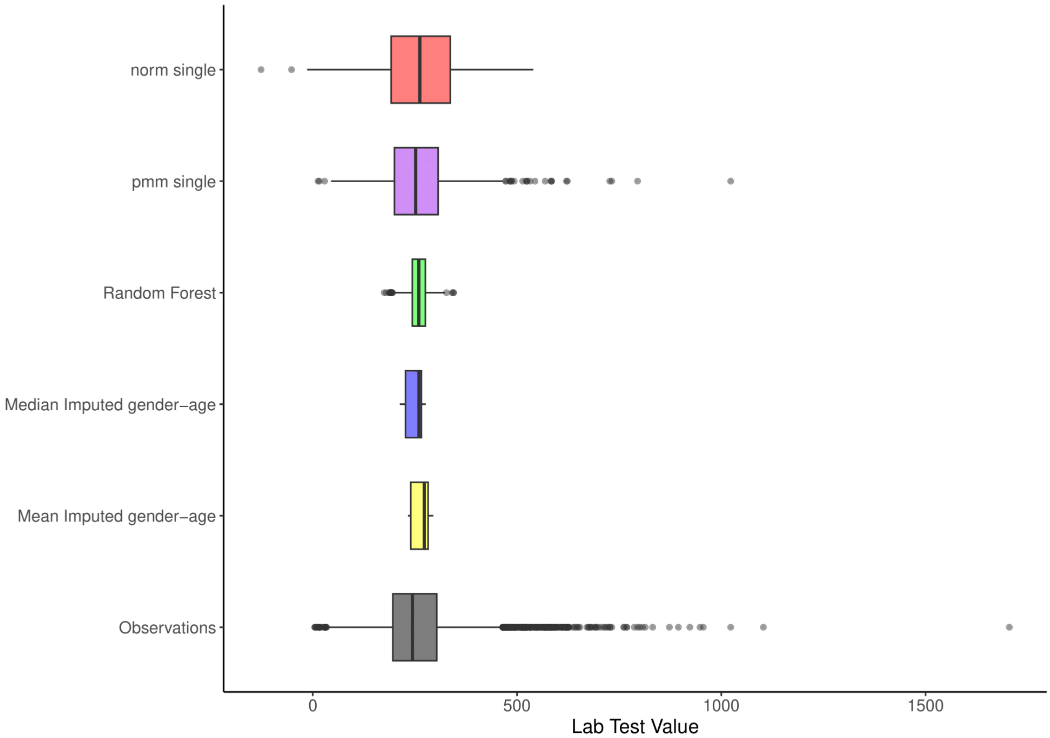

***Appendix Figure 24.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Neut (Neutrophil).*

PLT

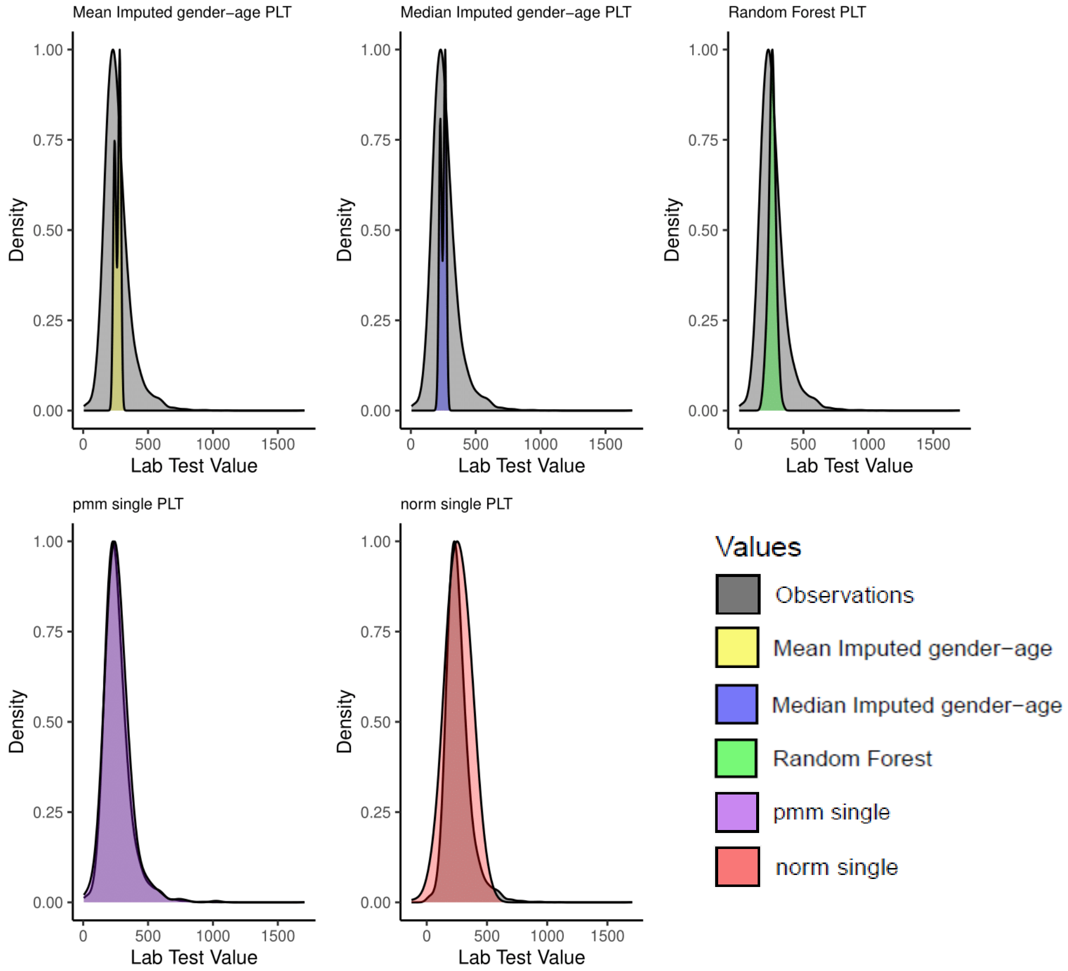
*Appendix Table 14. Statistics of imputed values of the different imputation methods for PLT.*

| PLT | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 261.12 | 244.00 | 103.42 | 5.00 | 1705.00 |
| **Mean** | 261.12 | 261.12 | 0.00 | 261.12 | 261.12 |
| **Mean gender-age** | 264.24 | 272.92 | 19.82 | 210.3 | 295.3 |
| **Median** | 244.00 | 244.00 | 0.00 | 244.00 | 244.00 |
| **Median gender-age** | 247.39 | 260.00 | 19.23 | 211.00 | 276.5 |
| **Random Forest** | 258.9 | 259.7 | 27.2 | 175.00 | 345.00 |
| **pmm MI** | 260.65 | 244.00 | 102.98 | 5.00 | 1705.00 |
| **pmm single** | 254.25 | 234.00 | 106.36 | 13.00 | 1023.00 |
| **norm MI** | 261.04 | 245.00 | 103.31 | -137.45 | 1705.00 |
| **norm single** | 257.65 | 260.-2 | 95.23 | -5.83 | 507.65 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for PLT (Thrombocytes). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 25**. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for PLT (Thrombocytes).*



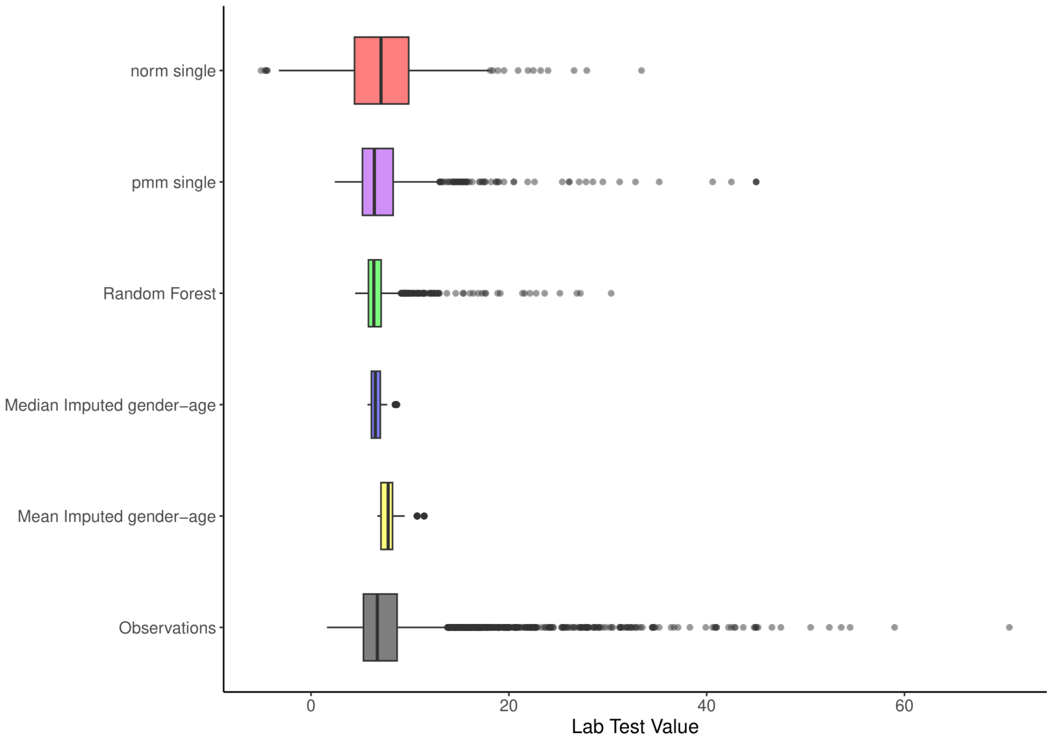

***Appendix Figure 26.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for PLT (Thrombocytes).*

UR

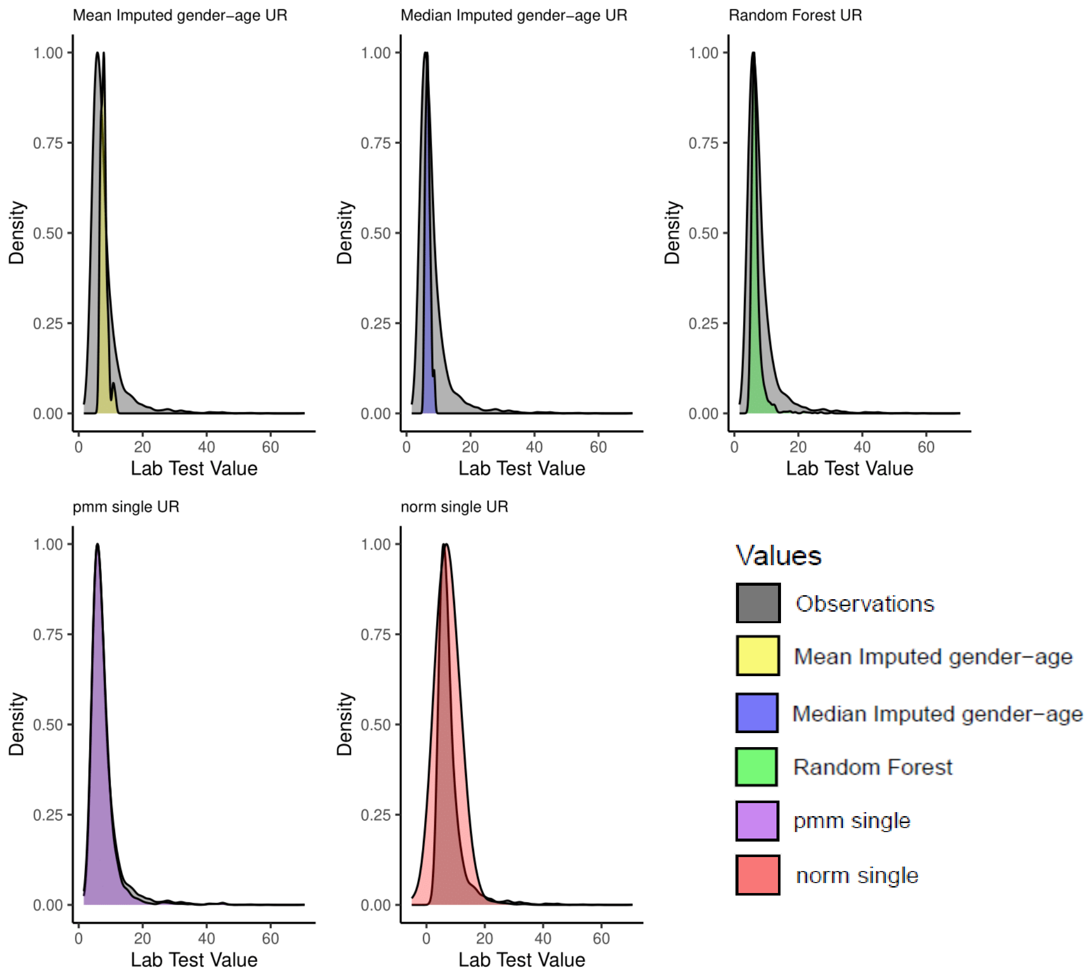
*Appendix Table 15. Statistics of imputed values of the different imputation methods for UR.*

| UR | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 7.95 | 6.7 | 5.1 | 1.6 | 70.6 |
| **Mean** | 7.95 | 7.95 | 0.00 | 7.95 | 7.95 |
| **Mean gender-age** | 7.86 | 7.99 | 1.03 | 6.7 | 11.36 |
| **Median** | 6.7 | 6.7 | 0.00 | 6.7 | 6.7 |
| **Median gender-age** | 6.64 | 6.5 | 0.74 | 5.7 | 8.65 |
| **Random Forest** | 6.88 | 6.35 | 2.27 | 4.5 | 30.3 |
| **pmm MI** | 7.81 | 6.6 | 4.85 | 1.6 | 70.6 |
| **pmm single** | 7.38 | 6.6 | 3.7 | 1.6 | 35.2 |
| **norm MI** | 7.83 | 6.7 | 4.95 | -7.84 | 70.6 |
| **norm single** | 7.48 | 7.42 | 4.22 | -5.19 | 26.76 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for UR (Urea). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 27**. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for UR (Urea).*



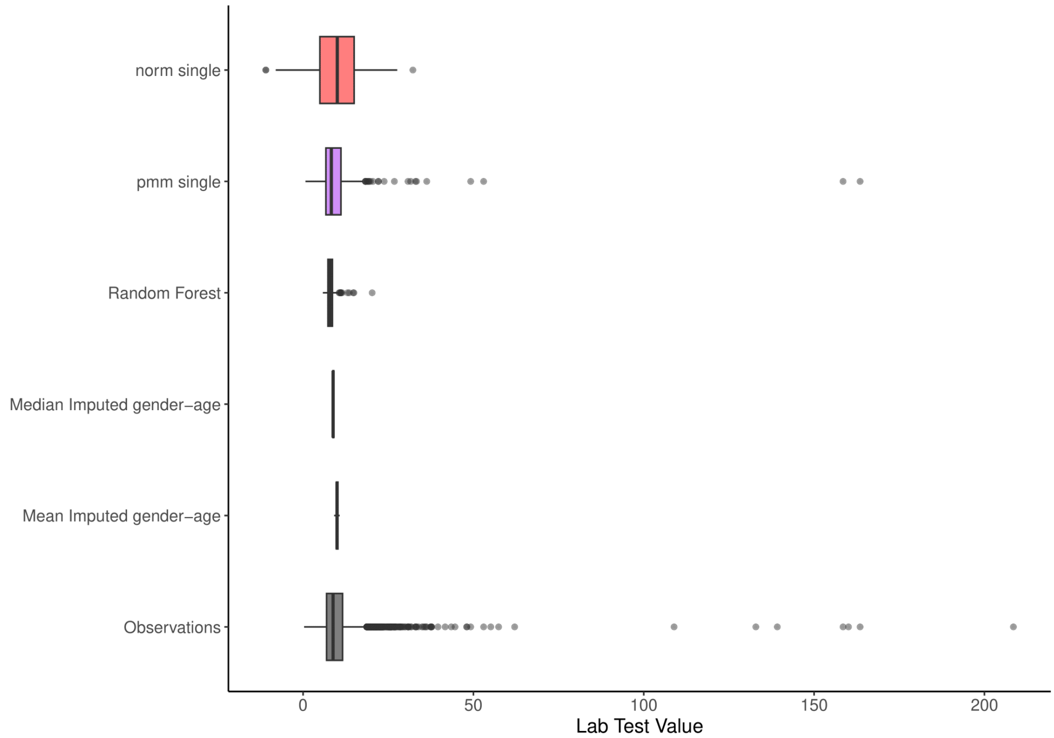

***Appendix Figure 28.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for UR (Urea).*

## WBC

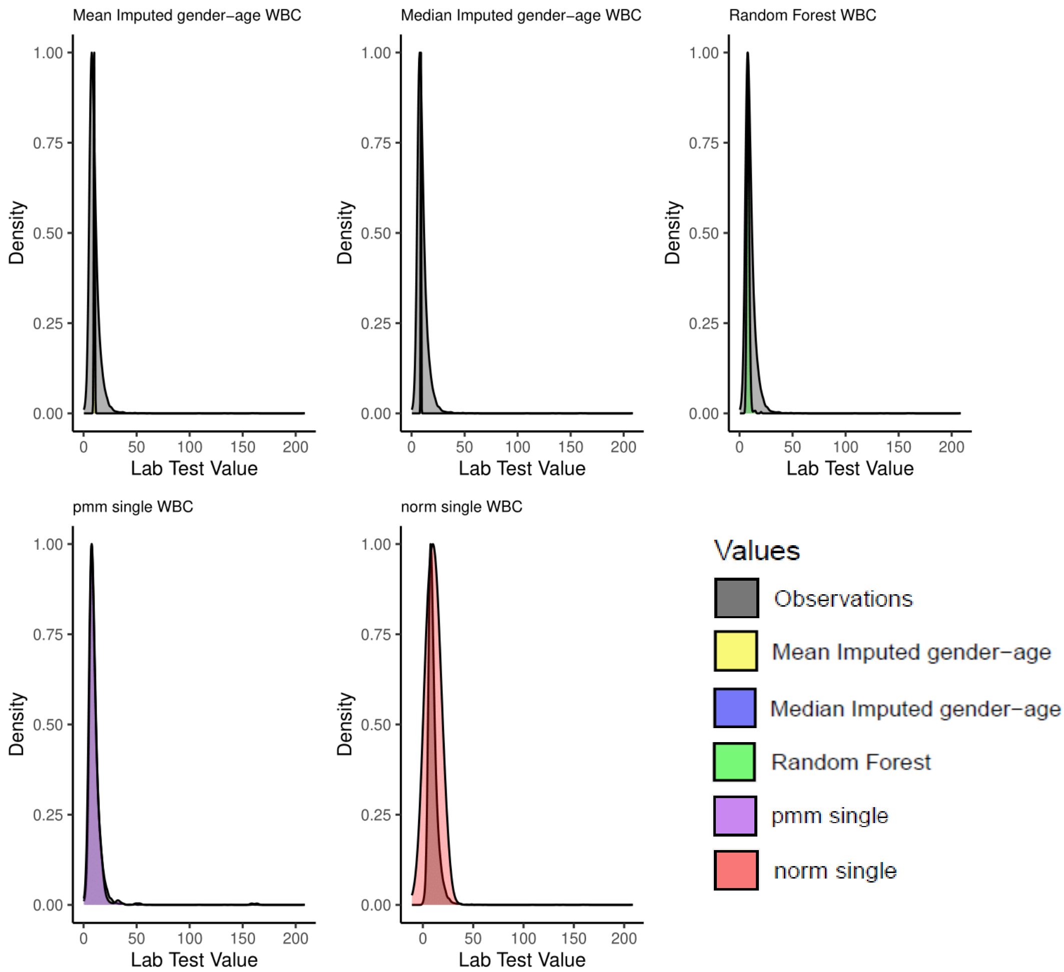
*Appendix Table 16. Statistics of imputed values of the different imputation methods for WBC.*

| WBC | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 10.09 | 8.8 | 6.97 | 0.3 | 208.5 |
| **Mean** | 10.09 | 10.09 | 0.00 | 10.09 | 10.09 |
| **Mean gender-age** | 10.07 | 9.96 | 0.36 | 9.00 | 10.74 |
| **Median** | 8.8 | 8.8 | 0.00 | 8.8 | 8.8 |
| **Median gender-age** | 8.8 | 8.8 | 0.15 | 8.2 | 9.00 |
| **Random Forest** | 8.07 | 7.93 | 1.24 | 5.8 | 20.3 |
| **pmm MI** | 10.03 | 8.8 | 6.96 | 0.3 | 208.5 |
| **pmm single** | 9.17 | 8.00 | 7.65 | 2.3 | 160.1 |
| **norm MI** | 10.05 | 8.8 | 6.97 | -14.53 | 208.5 |
| **norm single** | 9.4 | 9.24 | 6.63 | -10.88 | 29.76 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for WBC (White blood cell). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 29**. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for WBC (White blood cell).*




***Appendix Figure 30.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for WBC (White blood cell).*

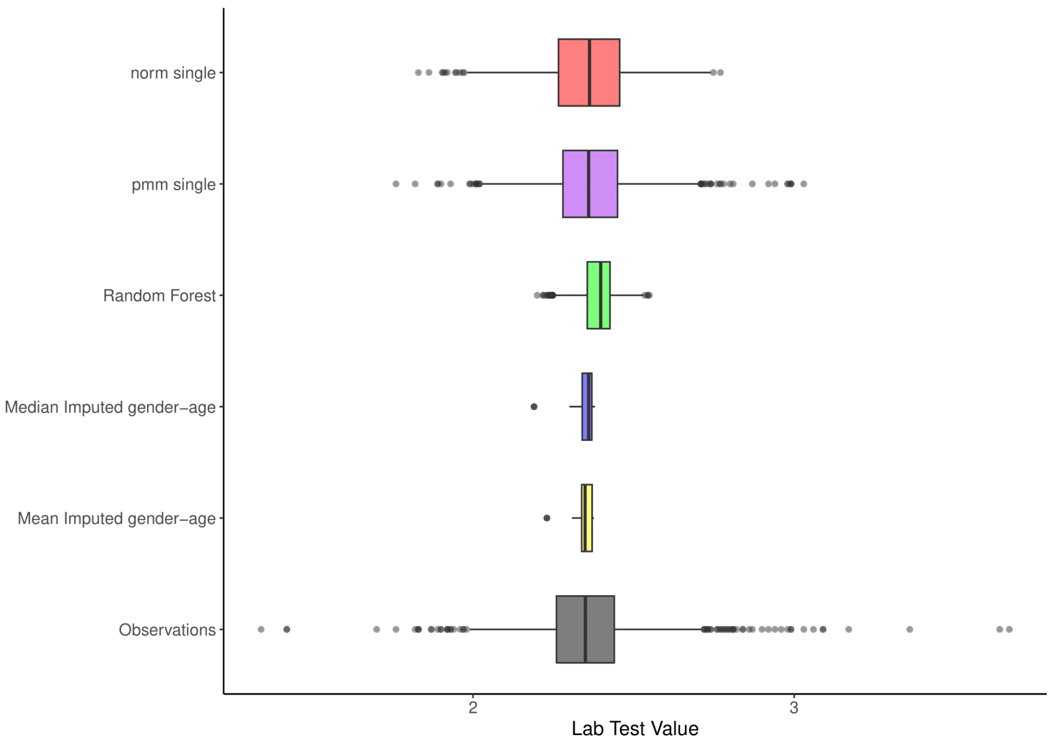
## Appendix C. Tables and figures of the laboratory tests for which Random Forest is the chosen imputation method.

### Ca

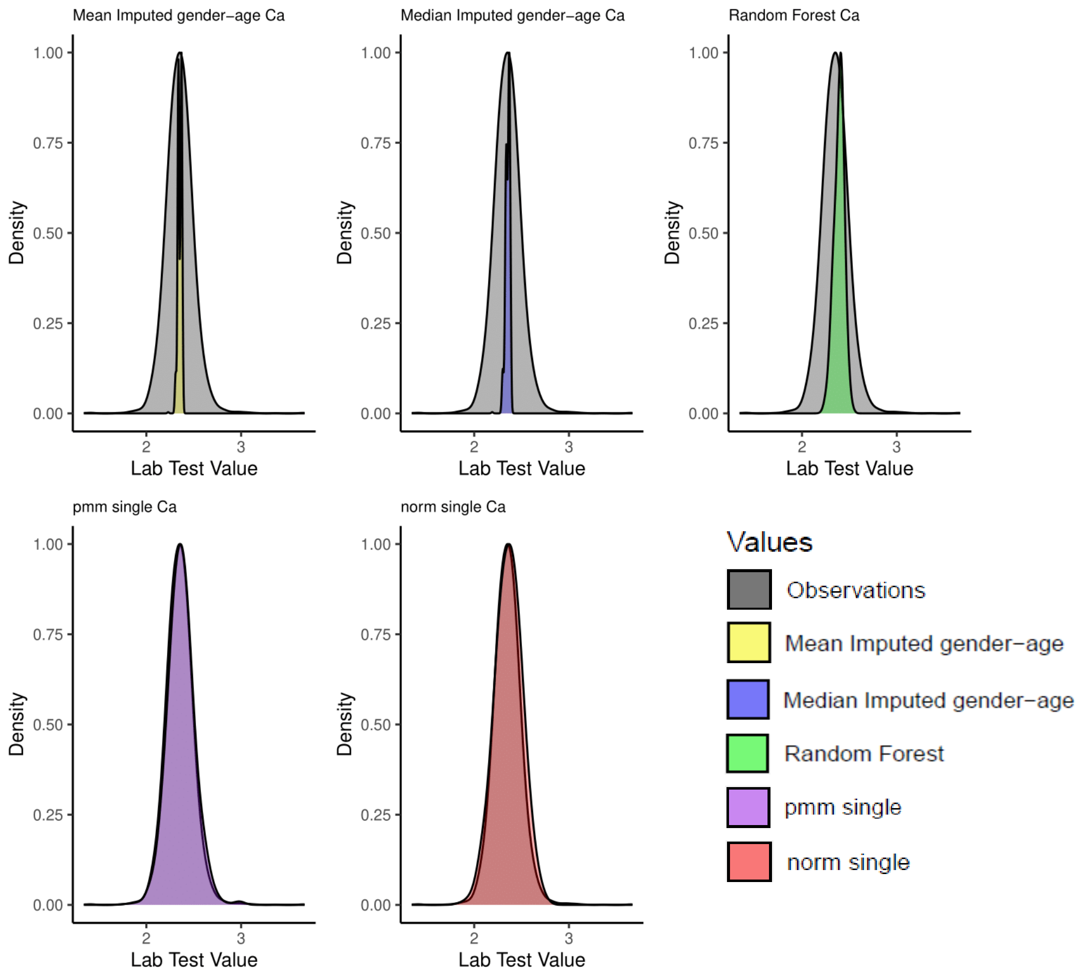
*Appendix Table 17. Statistics of imputed values of the different imputation methods for Ca.*

| Ca | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 2.35 | 2.35 | 0.14 | 1.34 | 3.67 |
| **Mean** | 2.35 | 2.35 | 0.00 | 2.35 | 2.35 |
| **Mean gender-age** | 2.35 | 2.35 | 0.02 | 2.23 | 2.38 |
| **Median** | 2.35 | 2.35 | 0.00 | 2.35 | 2.35 |
| **Median gender-age** | 2.35 | 2.36 | 0.02 | 2.19 | 2.38 |
| **Random Forest** | 2.39 | 2.4 | 0.05 | 2.2 | 2.5 |
| **pmm MI** | 2.36 | 2.4 | 0.14 | 1.34 | 3.67 |
| **pmm single** | 2.36 | 2.36 | 0.14 | 1.82 | 3.09 |
| **norm MI** | 2.36 | 2.4 | 0.14 | 1.34 | 3.67 |
| **norm single** | 2.36 | 2.36 | 0.14 | 1.89 | 2.78 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for Ca (Calcium). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 31.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Ca (Calcium).*



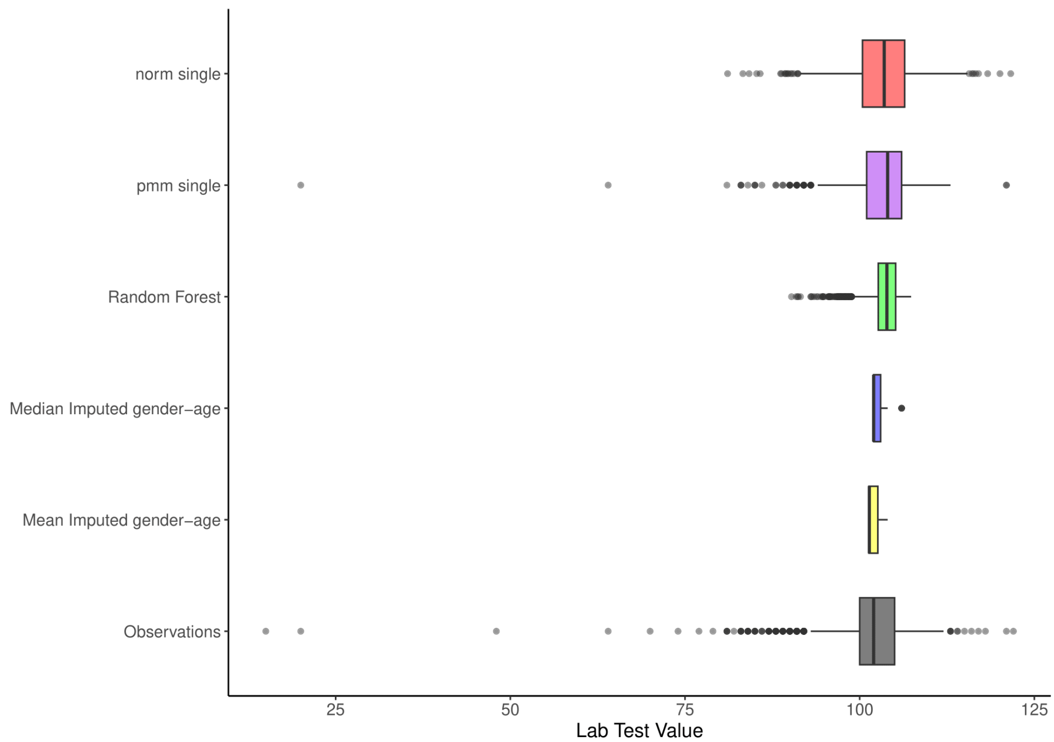

***Appendix Figure 32***. *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Ca (Calcium).*

## Cl

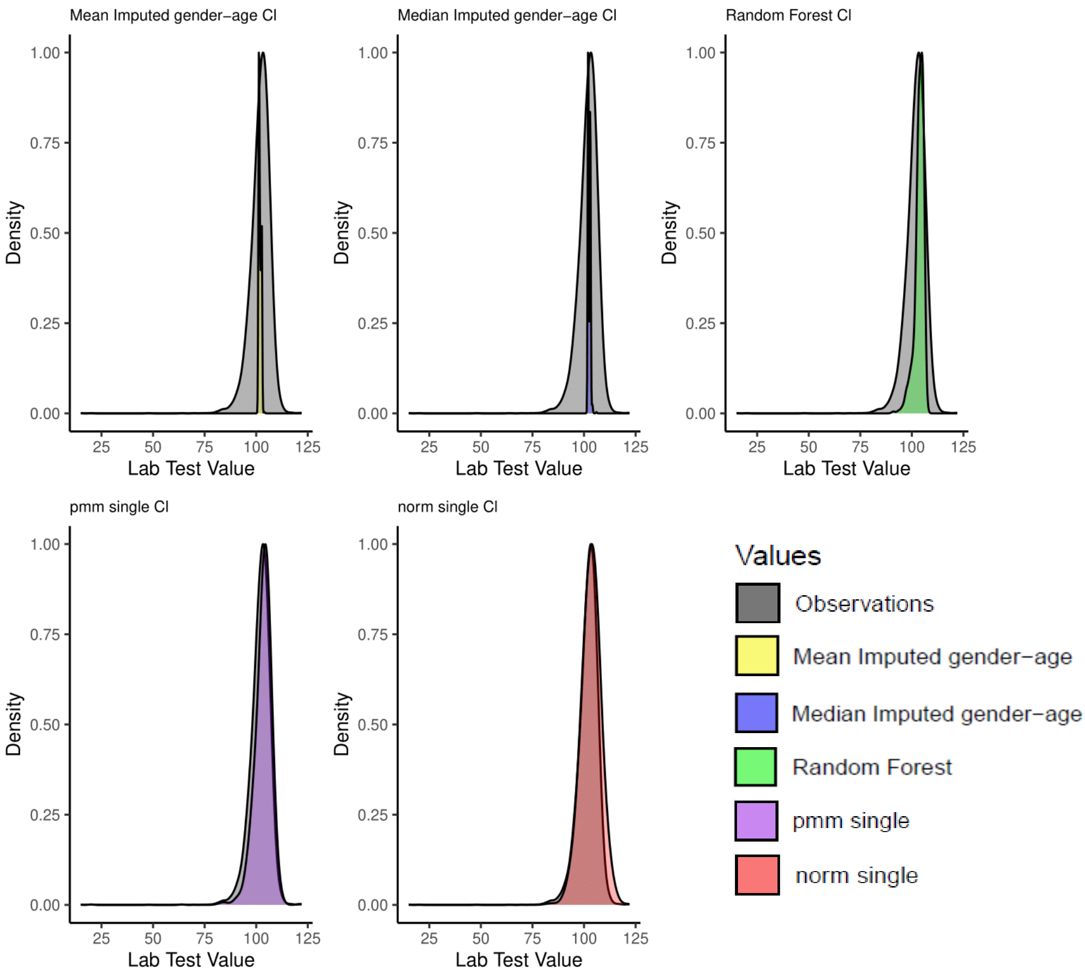
*Appendix Table 18. Statistics of imputed values of the different imputation methods for Cl.*

| Cl | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 101.88 | 102.00 | 16.72 | 15.00 | 122.00 |
| **Mean** | 101.88 | 101.88 | 0.00 | 101.88 | 101.88 |
| **Mean gender-age** | 101.82 | 101.38 | 0.5 | 101.23 | 104.00 |
| **Median** | 102.00 | 102.00 | 0.00 | 102.00 | 102.00 |
| **Median gender-age** | 102.49 | 102.00 | 0.547 | 102.00 | 106.00 |
| **Random Forest** | 103.52 | 103.9 | 2.27 | 90.00 | 107.00 |
| **pmm MI** | 102.35 | 103.00 | 4.88 | 15.00 | 122.00 |
| **pmm single** | 103.13 | 104.00 | 4.61 | 15.00 | 117.00 |
| **norm MI** | 102.4 | 103.00 | 4.99 | 15.00 | 131.44 |
| **norm single** | 103.49 | 103.62 | 4.73 | 80.81 | 121.71 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for Cl (Chloride). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 33.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Cl (Chloride).*



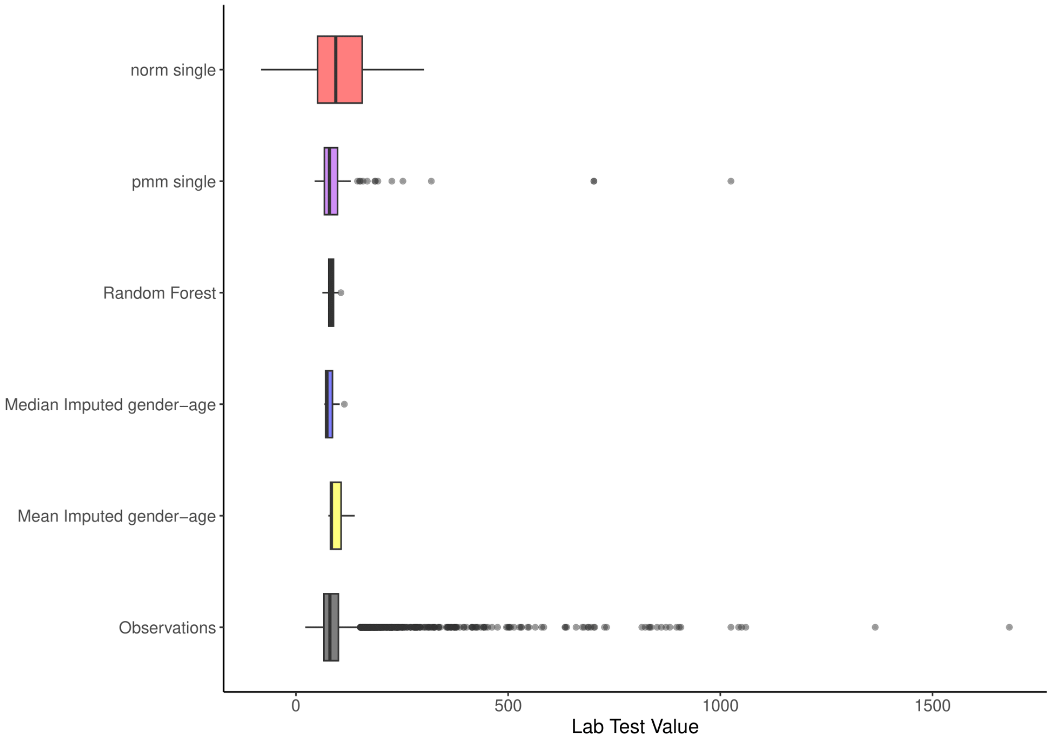

***Appendix Figure 34***. *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for Cl (Chloride).*

## CRE

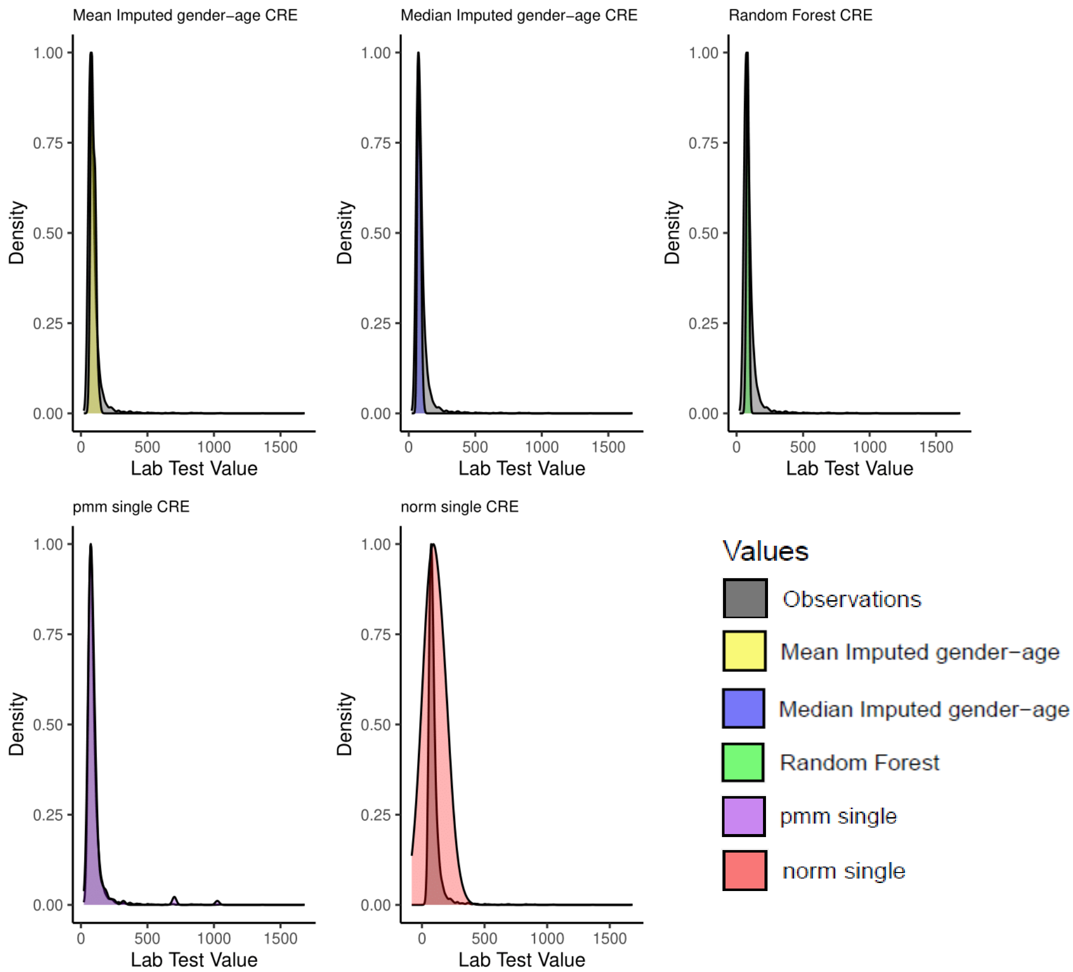
*Appendix Table 19. Statistics of imputed values of the different imputation methods for CRE.*

| CRE | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 95.32 | 80.00 | 74.42 | 22.00 | 1681.00 |
| **Mean** | 95.32 | 95.32 | 0.00 | 95.32 | 95.32 |
| **Mean gender-age** | 92.23 | 83.53 | 30.26 | 67.27 | 138.45 |
| **Median** | 80.00 | 80.00 | 0.00 | 80.00 | 80.00 |
| **Median gender-age** | 78.32 | 73.00 | 10.87 | 67.00 | 114.00 |
| **Random Forest** | 83.06 | 83.33 | 8.00 | 62.00 | 106.00 |
| **pmm MI** | 95.11 | 80.00 | 73.82 | 22.00 | 1681.00 |
| **pmm single** | 87.42 | 78.00 | 30.78 | 43.00 | 215.00 |
| **norm MI** | 95.25 | 80.00 | 74.41 | -134.16 | 1681.00 |
| **norm single** | 98.57 | 95.62 | 71.44 | -113.32 | 298.59 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for CRE (Creatinine). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 35****. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for CRE (Creatinine).*



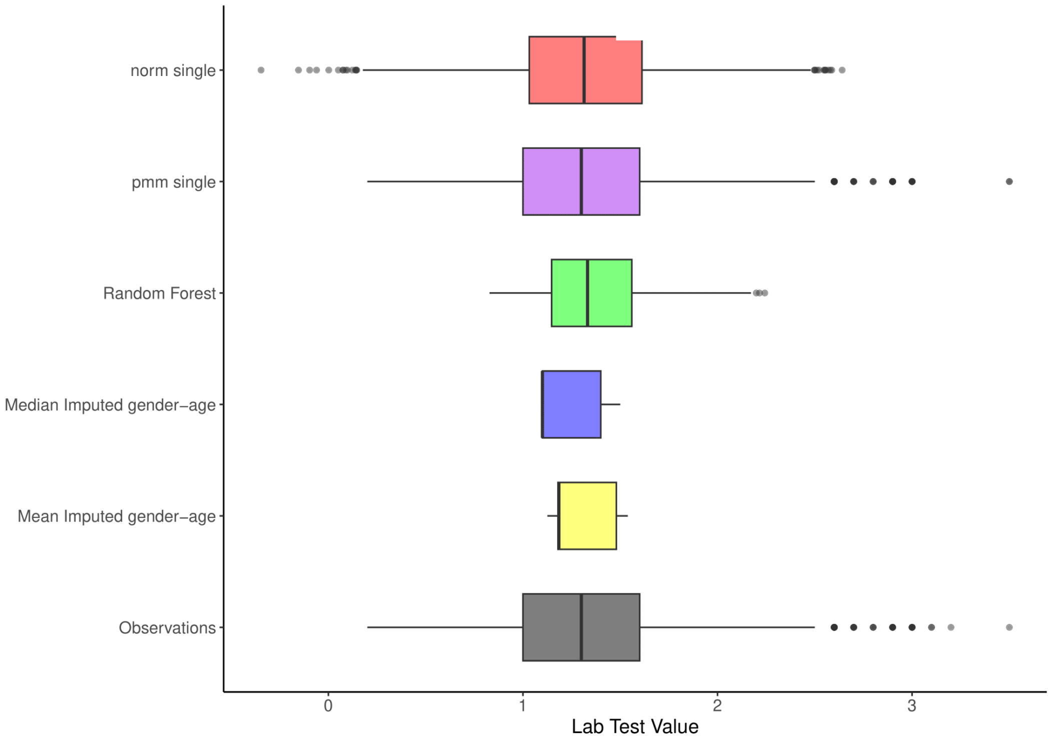

***Appendix Figure 36.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for CRE (Creatinine).*

## HDL

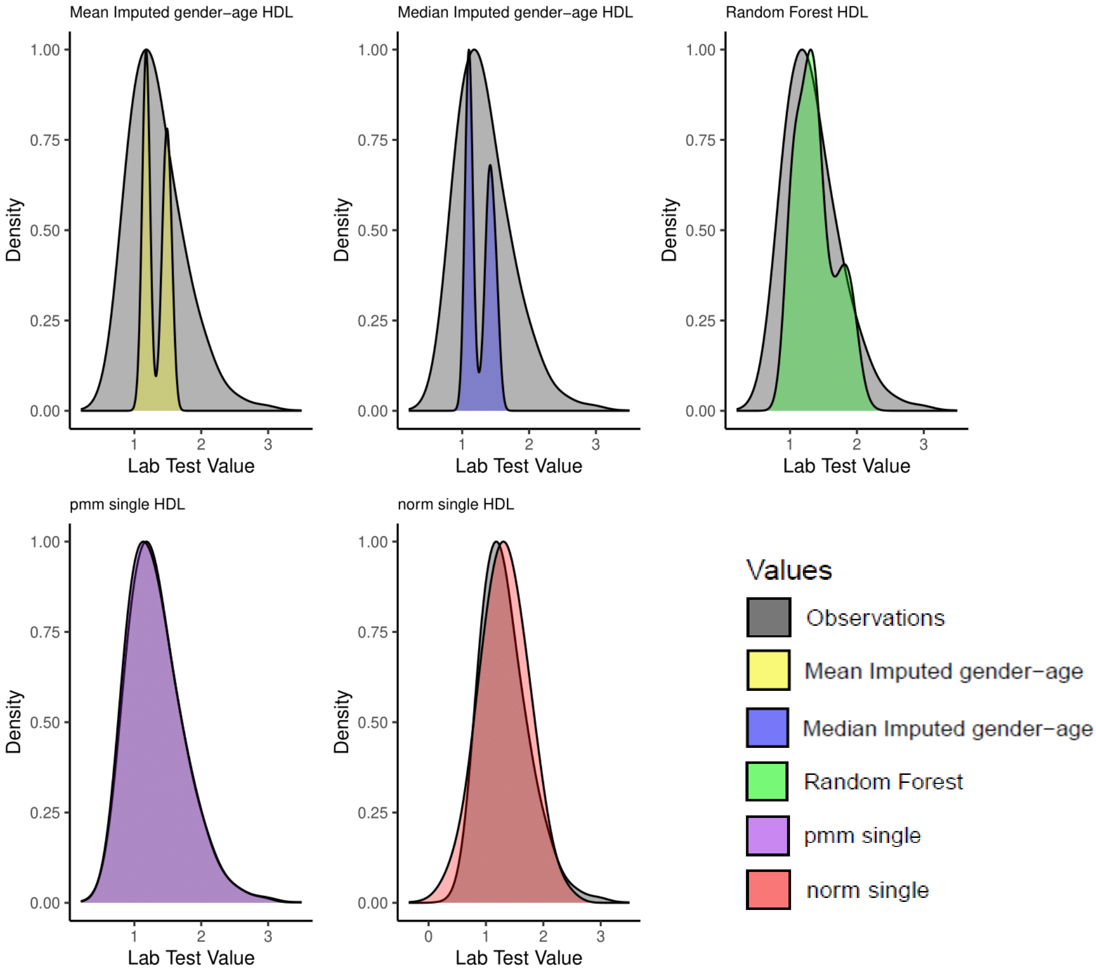
*Appendix Table 20. Statistics of imputed values of the different imputation methods for HDL.*

| HDL | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 1.33 | 1.3 | 0.42 | 0.2 | 3.5 |
| **Mean** | 1.33 | 1.33 | 0.00 | 1.33 | 1.33 |
| **Mean gender-age** | 1.18 | 1.18 | 0.16 | 1.13 | 1.54 |
| **Median** | 1.3 | 1.3 | 0.00 | 1.3 | 1.3 |
| **Median gender-age** | 1.26 | 1.1 | 0.17 | 1.1 | 1.5 |
| **Random Forest** | 1.38 | 1.33 | 0.29 | 0.83 | 2.2 |
| **pmm MI** | 1.32 | 1.3 | 0.42 | 0.2 | 3.5 |
| **pmm single** | 1.33 | 1.3 | 0.43 | 0.4 | 3.5 |
| **norm MI** | 1.32 | 1.3 | 0.42 | -0.78 | 3.5 |
| **norm single** | 1.32 | 1.32 | 0.43 | -0.43 | 2.89 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for HDL (High-density lipoproteins). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 37.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for HDL (High-density lipoproteins).*



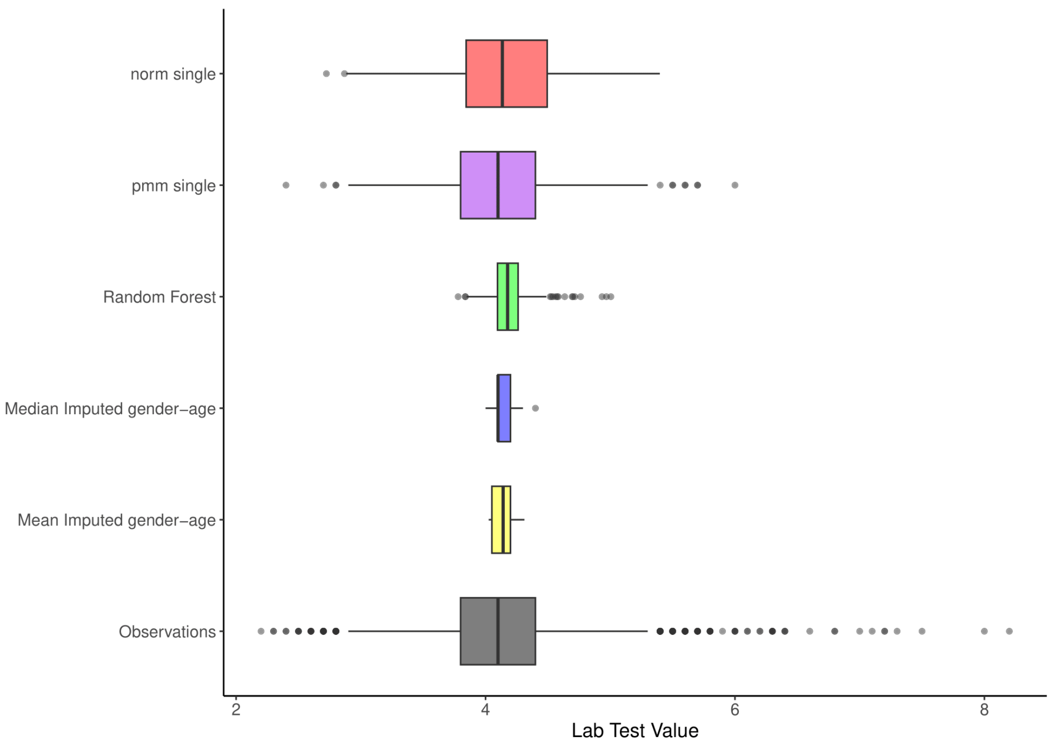

***Appendix Figure 38.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for HDL (High-density lipoproteins).*

K

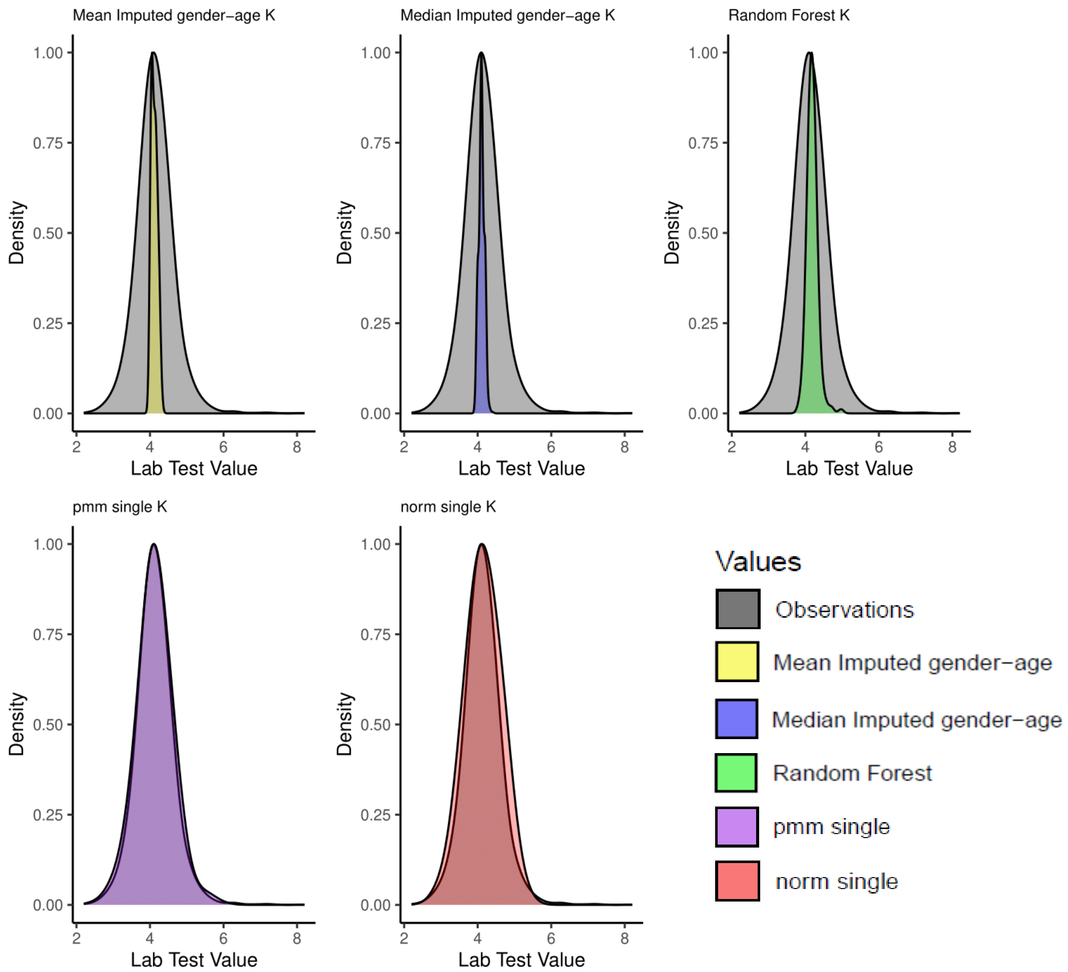
*Appendix Table 21. Statistics of imputed values of the different imputation methods for K.*

| K | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 4.13 | 4.1 | 0.51 | 2.2 | 8.2 |
| **Mean** | 4.13 | 4.13 | 0.00 | 4.13 | 4.13 |
| **Mean gender-age** | 4.12 | 4.14 | 0.08 | 4,00 | 4.3 |
| **Median** | 4.1 | 4.1 | 0.00 | 4.1 | 4.1 |
| **Median gender-age** | 4.11 | 4.1 | 0.07 | 4,00 | 4.4 |
| **Random Forest** | 4.19 | 4.18 | 0.15 | 3.8 | 5.00 |
| **pmm MI** | 4.12 | 4.1 | 0.5 | 2.2 | 8.2 |
| **pmm single** | 4.07 | 4.1 | 0.48 | 2.5 | 5.4 |
| **norm MI** | 4.12 | 4.1 | 0.5 | 2.2 | 8.2 |
| **norm single** | 4.12 | 4.09 | 0.5 | 2.78 | 5.76 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for K (Potassium). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 39****. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for K (Potassium).*



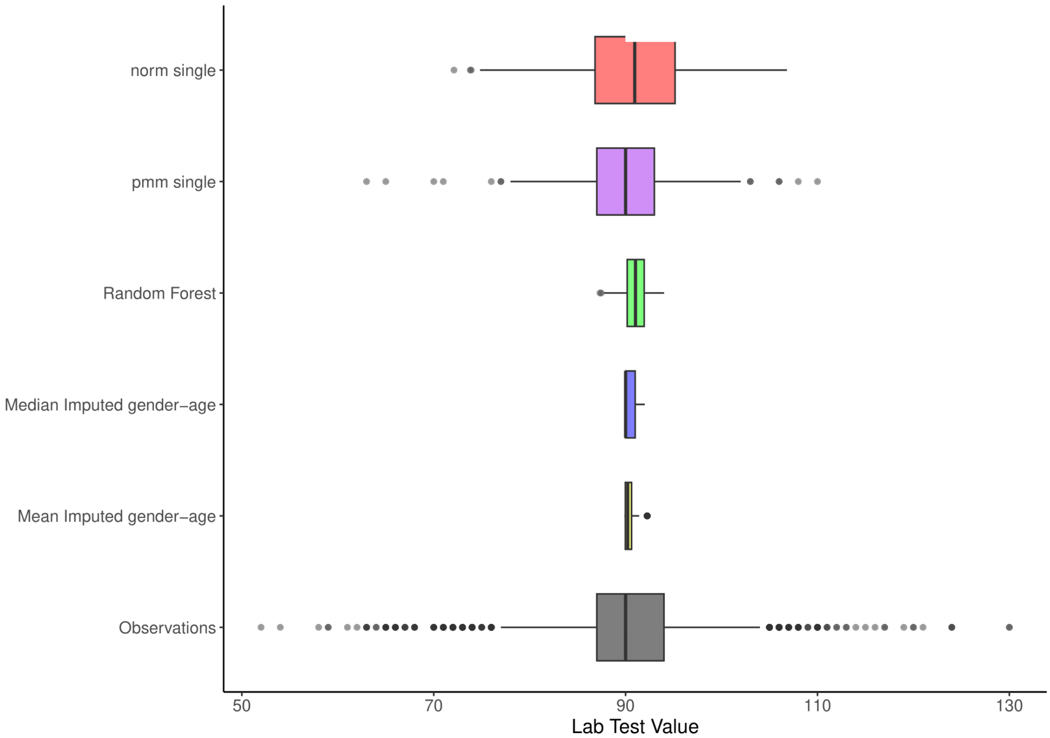

***Appendix Figure 40.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for K (Potassium).*

## MCV

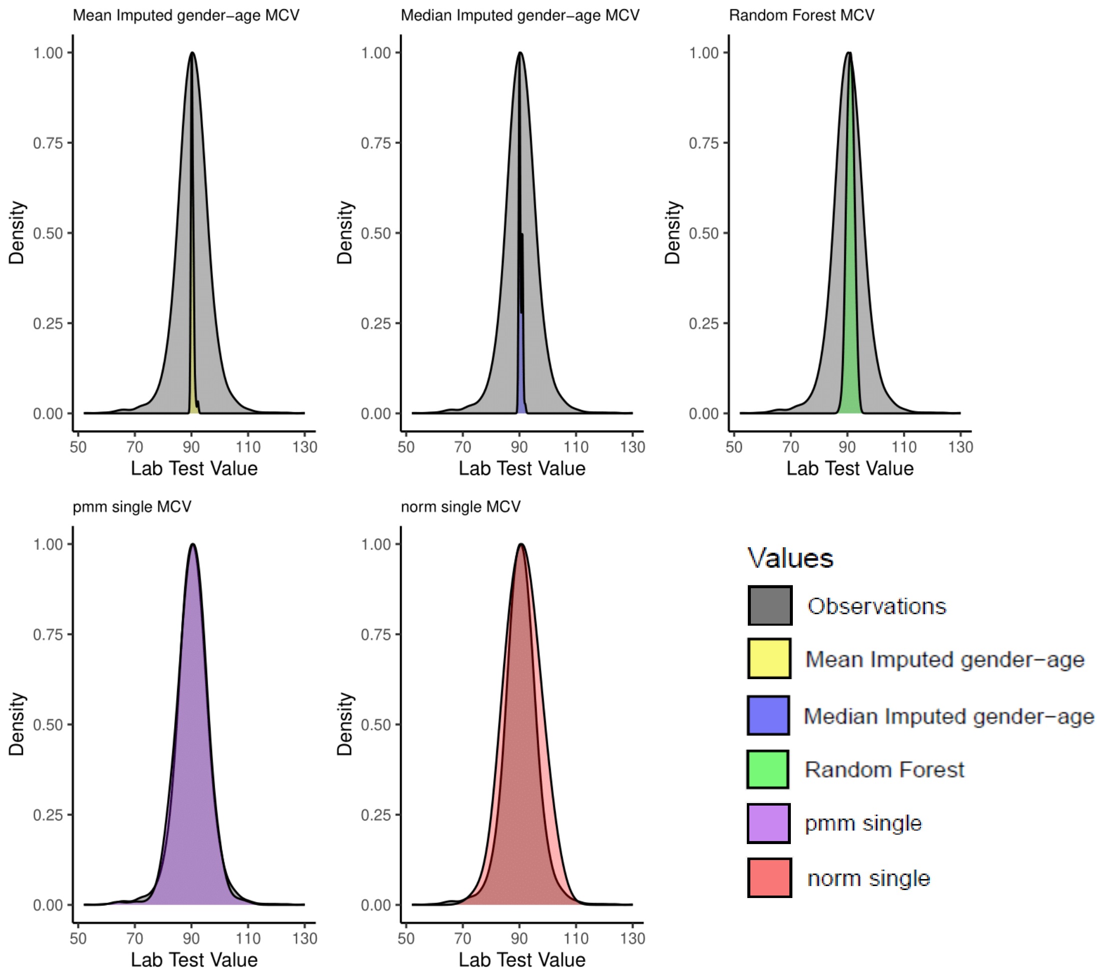
*Appendix Table 22. Statistics of imputed values of the different imputation methods for MCV.*

| MCV | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 90.38 | 90.00 | 6.02 | 52.00 | 130.00 |
| **Mean** | 90.38 | 90.38 | 0.00 | 90.38 | 90.38 |
| **Mean gender-age** | 90.32 | 90.22 | 0.46 | 89.88 | 92.43 |
| **Median** | 90.00 | 90.00 | 0.00 | 90.00 | 94.00 |
| **Median gender-age** | 90.36 | 90.00 | 0.52 | 52.00 | 130.00 |
| **Random Forest** | 91.05 | 91.00 | 1.21 | 87.00 | 94.00 |
| **pmm MI** | 90.41 | 90.00 | 5.99 | 52.00 | 130.00 |
| **pmm single** | 90.54 | 90.00 | 5.43 | 59.00 | 106.00 |
| **norm MI** | 90.41 | 90.00 | 6.01 | 52.00 | 130.00 |
| **norm single** | 90.73 | 90.58 | 5.87 | 72.42 | 108.19 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for MCV (Mean corpuscular volume). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 41.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for MCV (Mean corpuscular volume).*




***Appendix Figure 42***. *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for MCV (Mean corpuscular volume).*

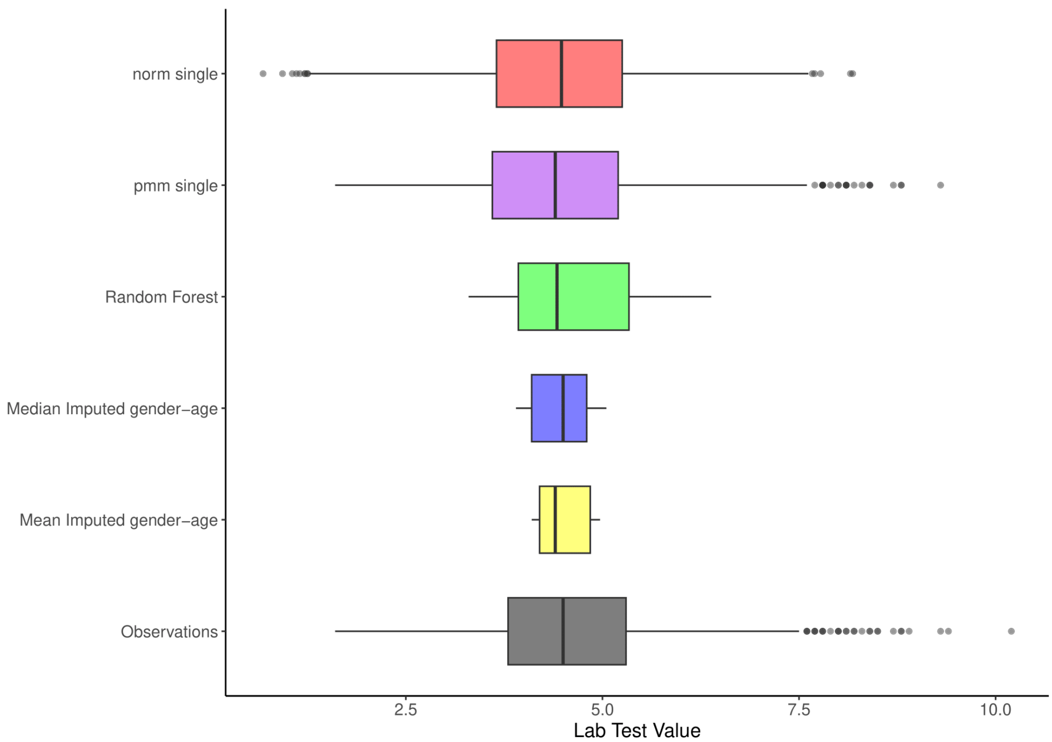
.

## TCHOL

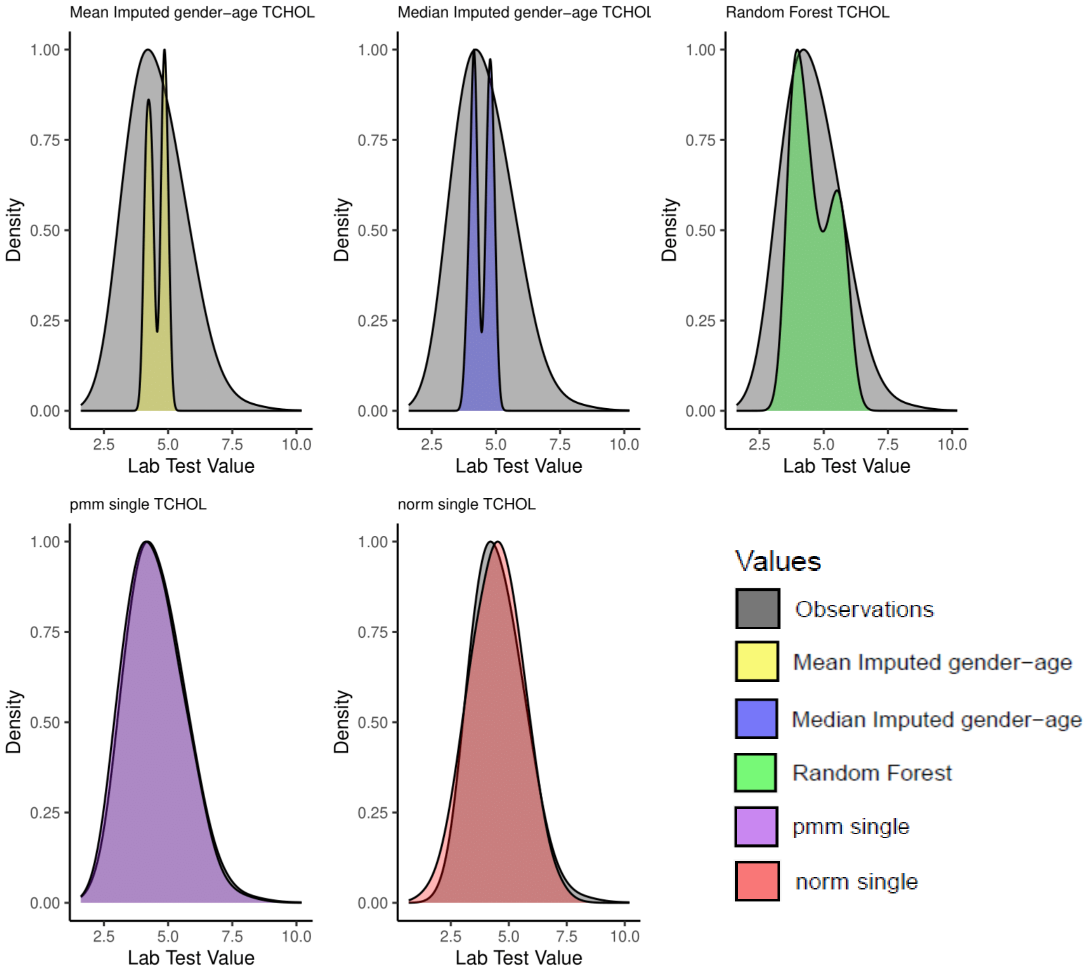
*Appendix Table 23. Statistics of imputed values of the different imputation methods for TCHOL.*

| TCHOL | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 4.56 | 4.5 | 1.13 | 1.6 | 10.2 |
| **Mean** | 4.56 | 4.56 | 0.00 | 4.56 | 4.56 |
| **Mean gender-age** | 4.55 | 4.4 | 0.32 | 4.1 | 5.00 |
| **Median** | 4.5 | 4.5 | 0.00 | 4.5 | 4.5 |
| **Median gender-age** | 4.45 | 4.5 | 0.35 | 3.9 | 5.1 |
| **Random Forest** | 4.61 | 4.42 | 0.77 | 3.3 | 6.4 |
| **pmm MI** | 4.52 | 4.5 | 1.13 | 1.6 | 10.2 |
| **pmm single** | 4.47 | 4.4 | 1.11 | 1.6 | 10.2 |
| **norm MI** | 4.53 | 4.5 | 1.13 | -2.39 | 10.2 |
| **norm single** | 4.46 | 4.44 | 1.13 | -2.38 | 8.39 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for TCHOL (Total cholesterol). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 43.*** *Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for TCHOL (Total cholesterol).*



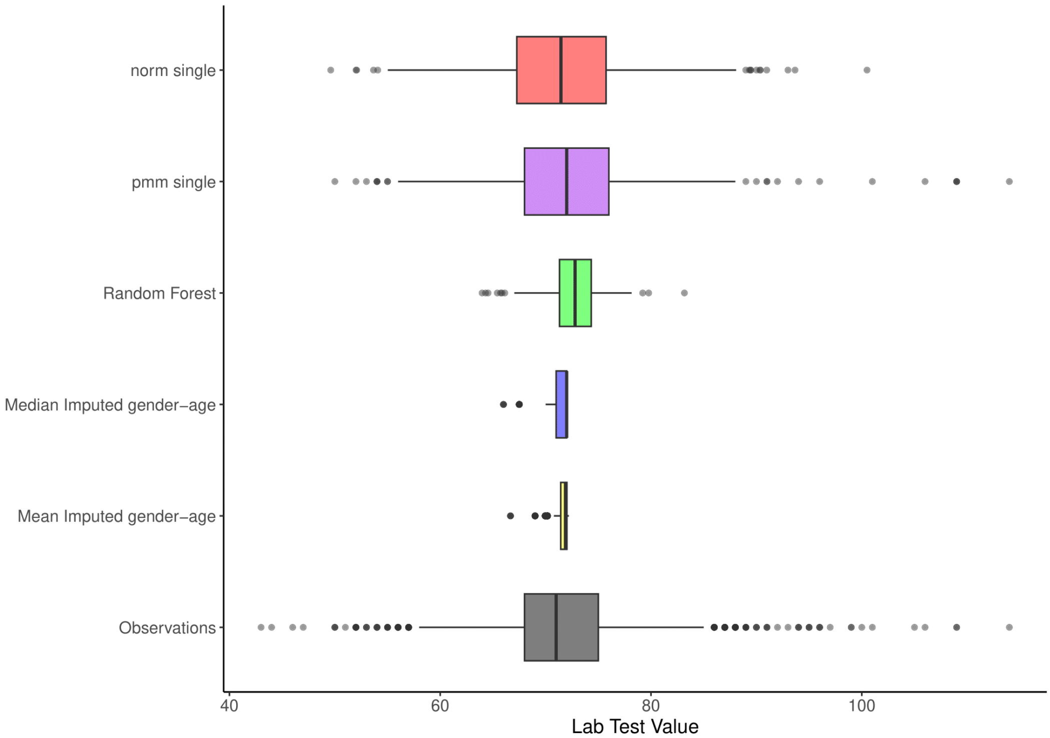

***Appendix Figure 44.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for TCHOL (Total cholesterol).*

TP

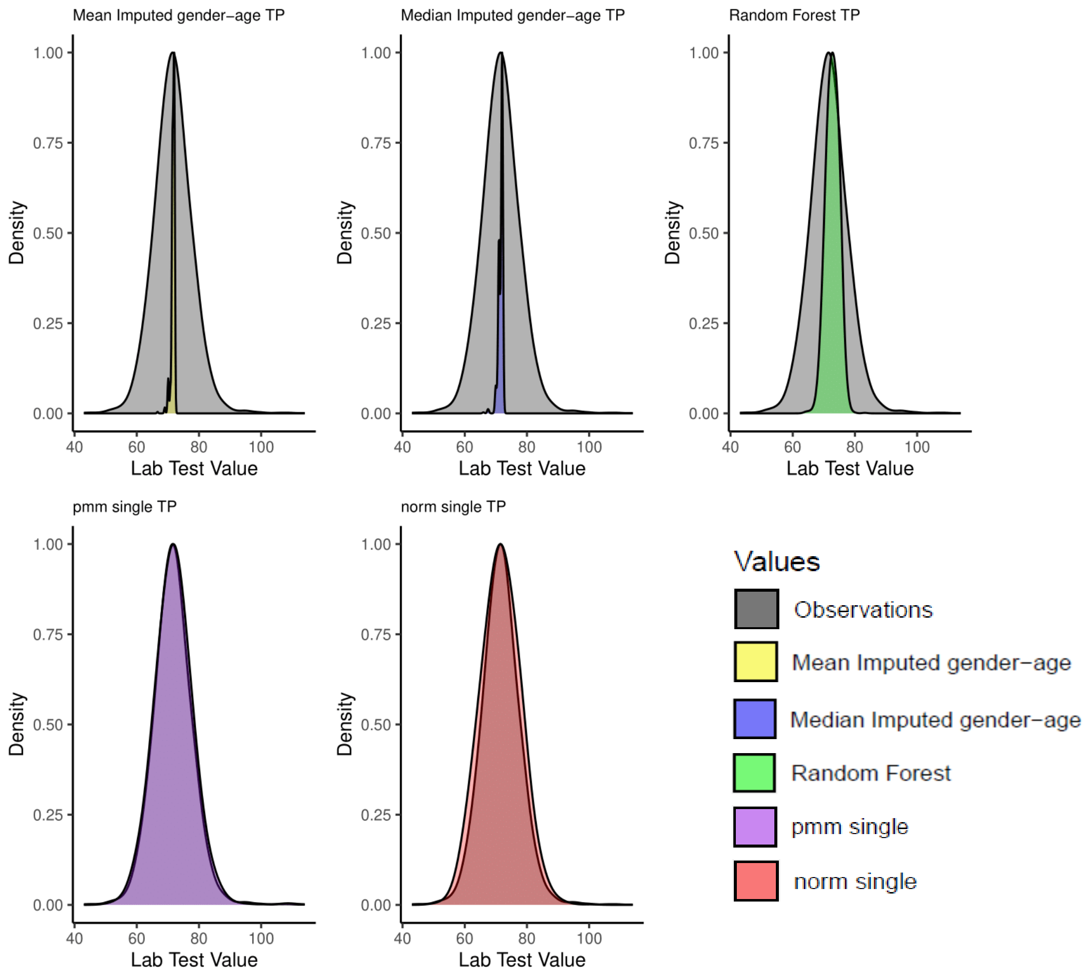
*Appendix Table 24. Statistics of imputed values of the different imputation methods for TP.*

| TP | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 71.63 | 71.00 | 6.15 | 43.00 | 114.00 |
| **Mean** | 71.63 | 71.63 | 0.00 | 71.63 | 71.63 |
| **Mean gender-age** | 71.64 | 71.88 | 0.58 | 66.67 | 72.2 |
| **Median** | 71.00 | 71.00 | 0.00 | 71.00 | 71.00 |
| **Median gender-age** | 71.55 | 72.00 | 0.72 | 66.00 | 72.00 |
| **Random Forest** | 72.76 | 72.8 | 2.1 | 64.00 | 83.00 |
| **pmm MI** | 71.66 | 71.00 | 6.12 | 43.00 | 114.00 |
| **pmm single** | 71.61 | 72.00 | 6.08 | 43.00 | 105.00 |
| **norm MI** | 71.68 | 71.00 | 6.21 | 43.00 | 114.00 |
| **norm single** | 71.67 | 71.8 | 6.23 | 49.93 | 97.04 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for TP (Total protein). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 45**. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for TP (Total protein).*




***Appendix Figure 46***. *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for TP (Total protein).*

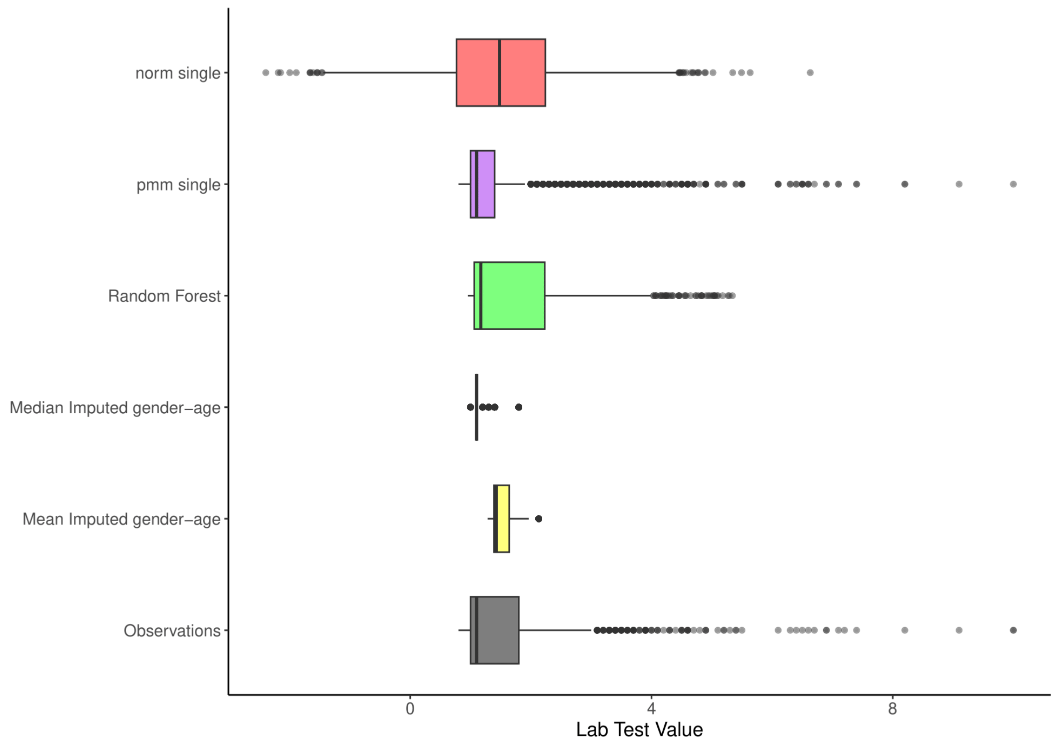
## Appendix D. Tables and figures of the laboratory tests for which the missing indicator method is the chosen imputation method.

### INR

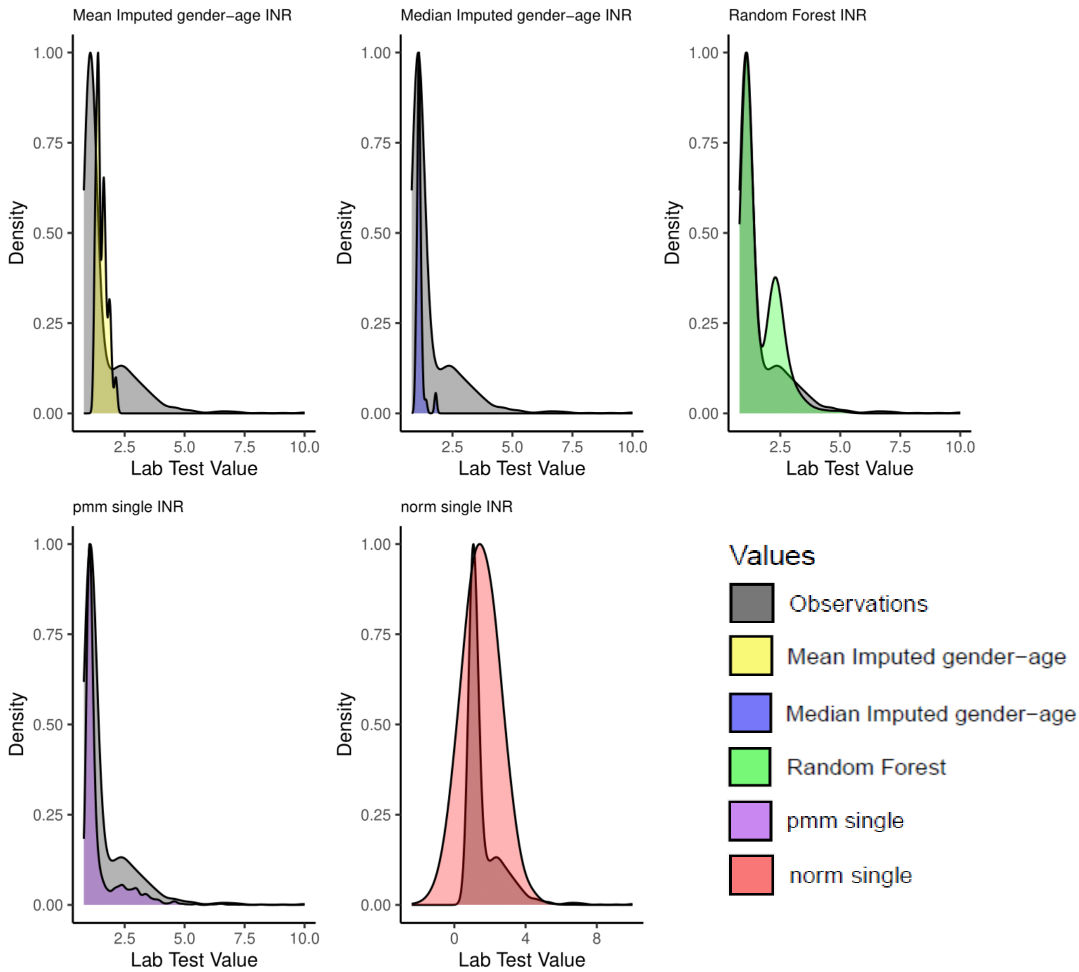
*Appendix Table 25. Statistics of imputed values of the different imputation methods for INR.*

| INR | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 1.59 | 1.1 | 1.05 | 0.8 | 10.00 |
| **Mean** | 1.59 | 1.59 | 0.00 | 1.59 | 1.59 |
| **Mean gender-age** | 1.54 | 1.43 | 0.22 | 1.28 | 2.13 |
| **Median** | 1.1 | 1.1 | 0.00 | 1.1 | 1.1 |
| **Median gender-age** | 1.13 | 1.1 | 0.15 | 1.00 | 1.8 |
| **Random Forest** | 1.63 | 1.71 | 0.75 | 0.96 | 5.3 |
| **pmm MI** | 1.51 | 1.1 | 0.96 | 0.8 | 10.00 |
| **pmm single** | 1.5 | 1.1 | 0.92 | 0.8 | 10.00 |
| **norm MI** | 1.5 | 1.3 | 1.05 | -3.29 | 10.00 |
| **norm single** | 1.48 | 1.49 | 1.03 | -2.07 | 5.19 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values of the different imputation methods for INR (International normalized ratio). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 47**. Boxplots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for INR (International normalized ratio).*




***Appendix Figure 48.*** *Density plots for the observed values and the imputed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for INR (International normalized ratio).*

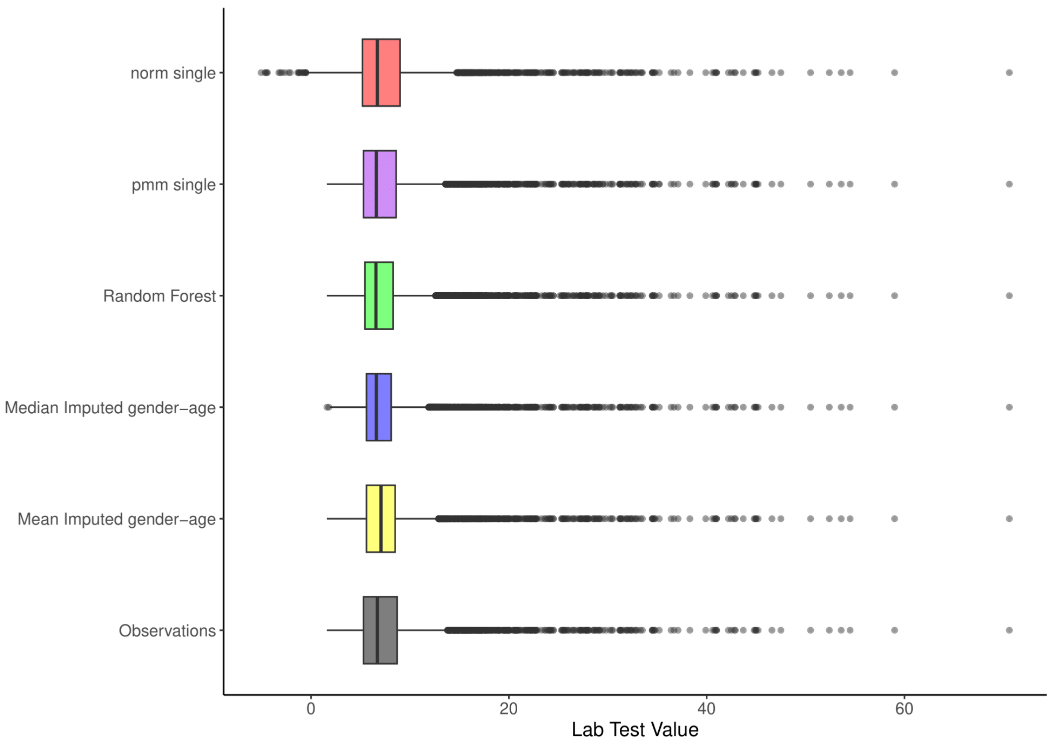
## Appendix E. Table and figures of MCV and UR combined with the observed values.

### UR

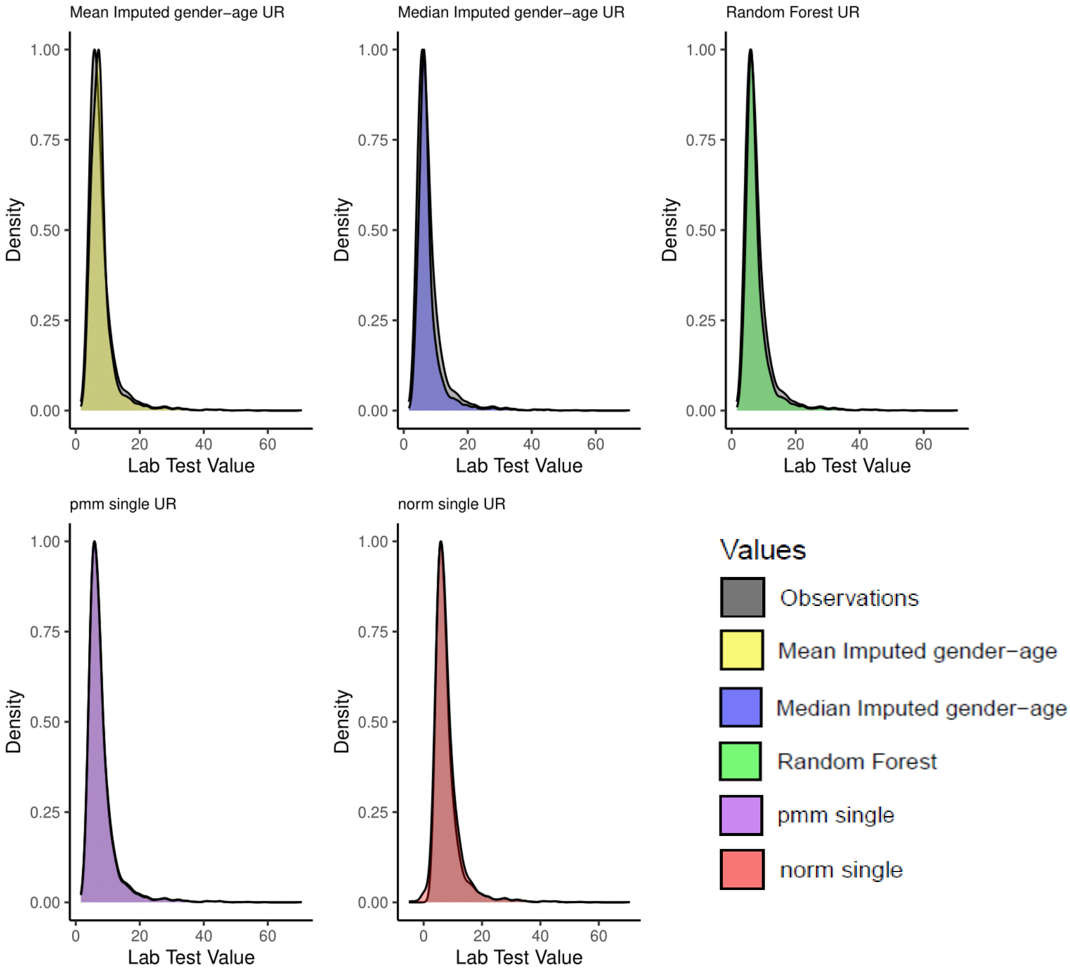
*Appendix Table 26. Statistics of imputed values of the different imputation methods for UR.*

| UR | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 7.95 | 6.7 | 5.1 | 1.6 | 70.6 |
| **Mean** | 7.95 | 7.5 | 4.57 | 1.6 | 70.6 |
| **Mean gender-age** | 7.93 | 7.07 | 4.59 | 1.6 | 70.6 |
| **Median** | 7.71 | 6.7 | 4.59 | 1.6 | 70.6 |
| **Median gender-age** | 7.69 | 6.6 | 4.6 | 1.6 | 70.6 |
| **Random Forest** | 7.74 | 6.56 | 4.7 | 1.6 | 70.6 |
| **pmm MI** | 7.85 | 6.7 | 4.89 | 1.6 | 70.6 |
| **pmm single** | 7.83 | 6.7 | 4.86 | 1.6 | 70.6 |
| **norm MI** | 7.88 | 6.8 | 4.92 | -5.12 | 70.6 |
| **norm single** | 7.86 | 6.8 | 4.94 | -5.19 | 70.6 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values combined with the observed values of the different imputation methods for UR (Urea). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 49**. Boxplots for the observed values and the imputed values combined with the observed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for UR (Urea).*



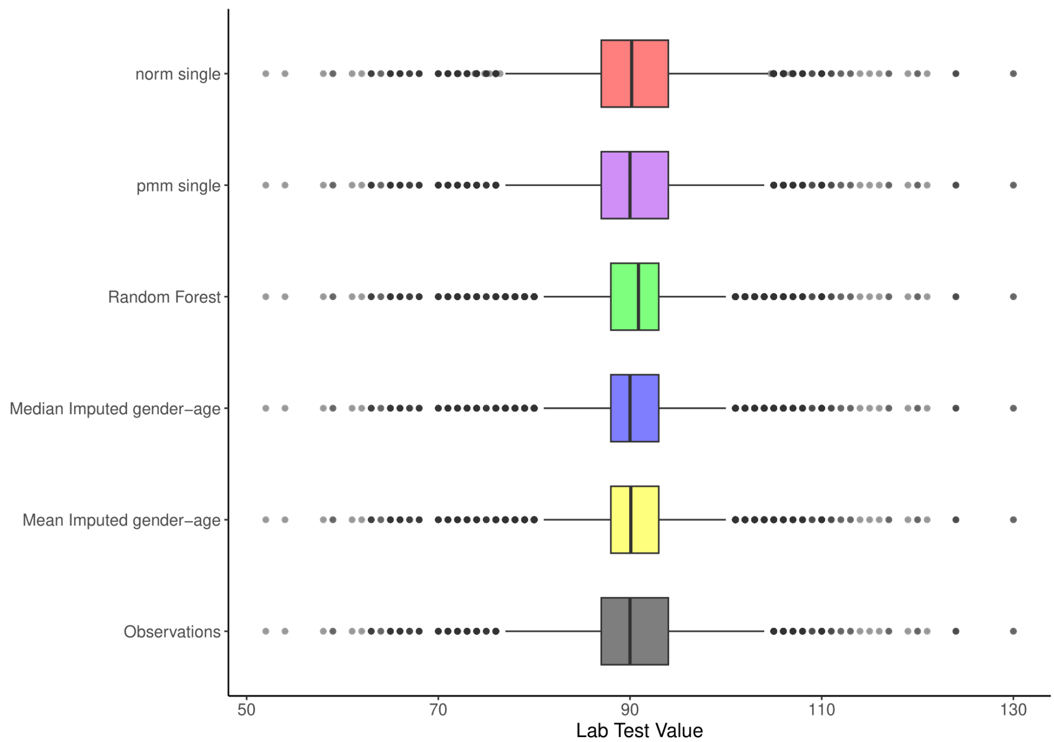

***Appendix Figure 50.*** *Density plots for the observed values and the imputed values combined with the observed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for UR (Urea).*

MCV

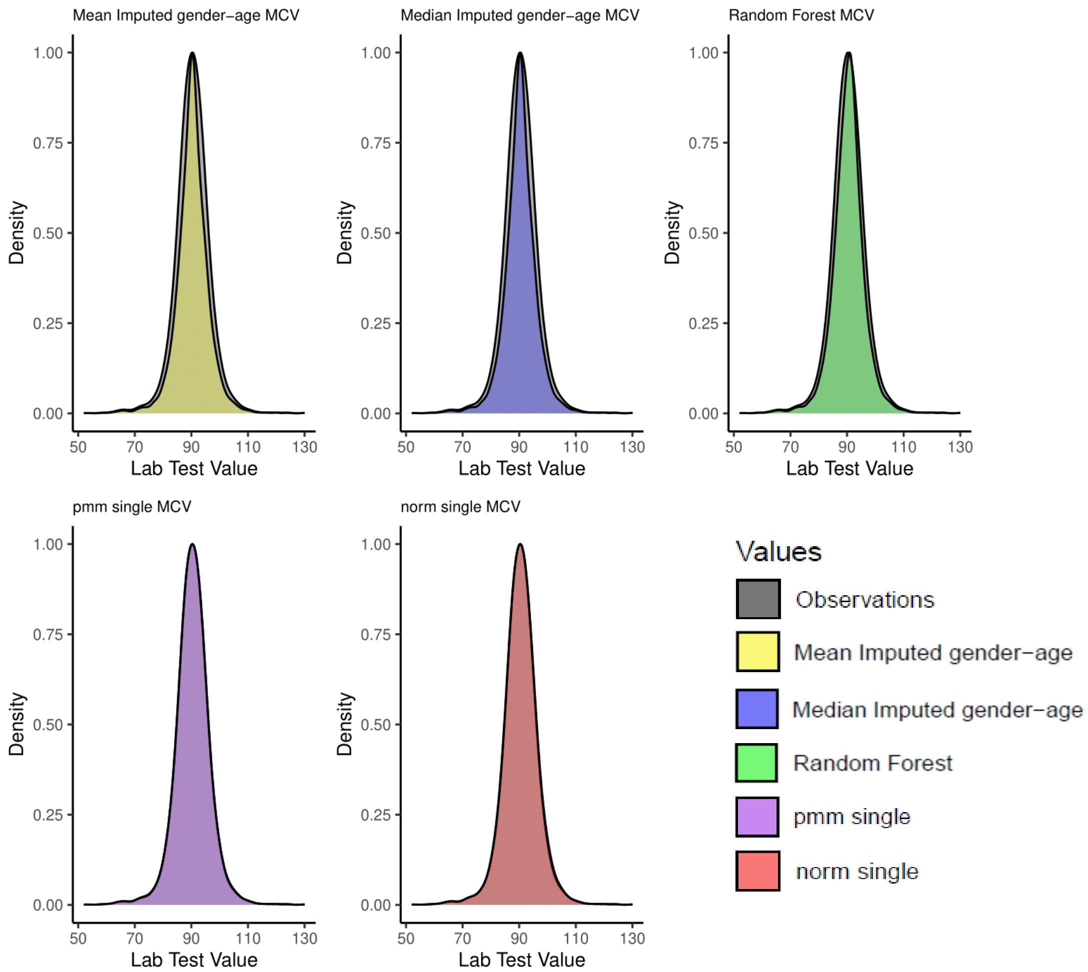
*Appendix Table 27. Statistics of imputed values of the different imputation methods for MCV*

| MCV | mean | median | sd | min | max |
|---|---|---|---|---|---|
| **Observations** | 90.38 | 90,00 | 6.02 | 52.00 | 130.00 |
| **Mean** | 90.38 | 90.38 | 5.76 | 52.00 | 130.00 |
| **Mean gender-age** | 90.38 | 90.09 | 5.76 | 52.00 | 130.00 |
| **Median** | 90.35 | 90.00 | 5.76 | 52.00 | 130.00 |
| **Median gender-age** | 90.38 | 90.00 | 5.76 | 52.00 | 130.00 |
| **Random Forest** | 90.43 | 90.89 | 5.76 | 52.00 | 130.00 |
| **pmm MI** | 90.39 | 90.00 | 6.01 | 52.00 | 130.00 |
| **pmm single** | 90.39 | 90.00 | 5.97 | 52.00 | 130.00 |
| **norm MI** | 90.41 | 90.00 | 6.00 | 52.00 | 130.00 |
| **norm single** | 90.41 | 90.00 | 6.00 | 52.00 | 130.00 |

*The mean, median, sd (standard deviation), min (minimum), and max (maximum) of the imputed values combined with the observed values of the different imputation methods for MCV (Mean corpuscular volume). MI = Multiple imputation, pmm = predictive mean matching, norm = Bayesian linear regression.*




***Appendix Figure 51.*** *Boxplots for the observed values and the imputed values combined with the observed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for MCV (Mean corpuscular volume).*




***Appendix Figure 52.*** *Density plots for the observed values and the imputed values combined with the observed values of Mean Imputed gender-age, Median Imputed gender-ag,. Random Forest, pmm (predictive mean matching) single, and norm (Bayesian linear regression) single for MCV (Mean corpuscular volume).*

# Appendix F. Main R code

````
---
title: ""
author: "Kai van de Poll"
format: html
editor: visual
date: "16-08-2023"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## Set Library
##It is necessary to set .libPaths by hand before running this .qmd file:

```{r set path to library}
.libPaths("H:/R/Library")
```

## Load required Packages

```{r load packages}
library(knitr)
library(markdown)
library(rlang)
library(vctrs)
library(tibble)
library(ggplot2)
library(tidyr)
library(clipr)
library(tidyverse)
library(tidymodels)
library(readxl)
library(summarytools)
library(skimr)
library(corrplot)
library(purrr)
library(plyr)
library(naniar)
library(RColorBrewer)
library(gridExtra)
library(data.table)
library(rpart)
library(rpart.plot)
library(fitdistrplus)
library(mice)
library(dplyr)
library(randomForest)
library(patchwork)
library(openxlsx)
library(pastecs)
library(missMDA)
```

## Set path to load dataset

```{r set path to load dataset}
path = "H:/R/Code Kai/voorspelmodel-kwetsbaarheid-bij-ouderen-lab-20230512.xlsx"
```

## Import all sheets

```{r import data}
sheet_names = excel_sheets(path)
tibble_frailty_1 = lapply(sheet_names, function(x) {as_tibble(read_excel(path, sheet = x))})
names(tibble_frailty_1) = sheet_names

```

### Filter patients based on the condition that the newest screener is also the oldest
screener, and exclude patients whose screener was a consult. And update the patients' age to
their age at the time of the last screener.

```{r}
````
````

~~~~
oldest_newest_df <-
tibble_frailty_1$overview[as.Date(tibble_frailty_1$overview$screening_newest_form_start_date)
!= as.Date(tibble_frailty_1$overview$screening_oldest_form_start_date) |
is.na(tibble_frailty_1$overview$screening_newest_form_start_date)|
is.na(tibble_frailty_1$overview$screening_oldest_form_start_date) |
tibble_frailty_1$overview$screening_newest_form_description == "Consult" |
tibble_frailty_1$overview$screening_oldest_form_description == "Consult", "pseudo_id"]

pseudo_ids <- oldest_newest_df$pseudo_id
tibble_frailty_1 <- lapply(tibble_frailty_1, function(df) {
  df[!(df[[1]] %in% pseudo_ids), ]
})

# Replace "age" column with "screening_newest_form_age_at_time_of_event" column
tibble_frailty_1$overview$age <-
tibble_frailty_1$overview$screening_newest_form_age_at_time_of_event

# Replace empty cells in "age" column with corresponding values from
"screening_newest_report_age_at_time_of_event" column
tibble_frailty_1$overview$age[is.na(tibble_frailty_1$overview$age)] <-
tibble_frailty_1$overview$screening_newest_report_age_at_time_of_event[is.na(tibble_frailty_1$
overview$age)]

# Replace empty cells in "age" column with corresponding values from
"cga_care_activities_single_newest_procedure_age_at_time_of_event" column
tibble_frailty_1$overview$age[is.na(tibble_frailty_1$overview$age)] <-
tibble_frailty_1$overview$cga_care_activities_single_newest_procedure_age_at_time_of_event[is.
na(tibble_frailty_1$overview$age)]

```
~~~~

### rename sheets and merging the _poct with the non poct in the overview.

~~~~
```{r rename sheets}
sheet_names_df = as.data.frame(sheet_names)
sheet_names_df[sheet_names_df == "metadata"] <- "1-metadata"
sheet_names_df[sheet_names_df == "overview"] <- "2-overview"
sheet_names_df = sheet_names_df %>% separate(sheet_names, c("varnumber", "varname"), "-")

sheet_names_df[sheet_names_df == "albumin"] <- "ALB"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("albumin","ALB",colnames(x))))
sheet_names_df[sheet_names_df == "alat"] <- "ALAT"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("alat","ALAT",colnames(x))))
sheet_names_df[sheet_names_df == "asat"] <- "ASAT"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("asat","ASAT",colnames(x))))
sheet_names_df[sheet_names_df == "bili_direct"] <- "BILID"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("bili_direct","BILID",colnames(x))))
sheet_names_df[sheet_names_df == "bili_total"] <- "BILIT"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("bili_total","BILIT",colnames(x))))
sheet_names_df[sheet_names_df == "calcium"] <- "Ca"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("calcium","Ca",colnames(x))))
sheet_names_df[sheet_names_df == "cd_4"] <- "CD4"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("cd_4","CD4",colnames(x))))
sheet_names_df[sheet_names_df == "chloride"] <- "Cl"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("chloride","Cl",colnames(x))))
sheet_names_df[sheet_names_df == "ck"] <- "CK"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("ck","CK",colnames(x))))
sheet_names_df[sheet_names_df == "creatinine"] <- "CRE"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("creatinine","CRE",colnames(x))))
sheet_names_df[sheet_names_df == "crp"] <- "CRP"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("crp","CRP",colnames(x))))
sheet_names_df[sheet_names_df == "crp_poct"] <- "CRP_poct"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("crp_poct","CRP_poct",colnames(x))))
sheet_names_df[sheet_names_df == "egfr_ckd_epi"] <- "eGFR"
~~~~

```
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("egfr_CKd_epi","eGFR",colnames(x))))
sheet_names_df[sheet_names_df == "egfr_mdrd"] <- "eGFR_mdrd"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("egfr_mdrd","eGFR_mdrd",colnames(x))))
sheet_names_df[sheet_names_df == "erythrocytes"] <- "RBC"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("erythrocytes","RBC",colnames(x))))
sheet_names_df[sheet_names_df == "esr"] <- "ESR"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("esr","ESR",colnames(x))))
sheet_names_df[sheet_names_df == "ferritin"] <- "FER"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("ferritin","FER",colnames(x))))
sheet_names_df[sheet_names_df == "folate"] <- "FA"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("folate","FA",colnames(x))))
sheet_names_df[sheet_names_df == "gamma_gt"] <- "GGT"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("gamma_gt","GGT",colnames(x))))
sheet_names_df[sheet_names_df == "glucose_fasting"] <- "FGLC"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("glucose_fasting","FGLC",colnames(x))))
sheet_names_df[sheet_names_df == "glucose"] <- "GLC"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("glucose","GLC",colnames(x))))
sheet_names_df[sheet_names_df == "glucose_poct"] <- "glu_poct"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("glucose_poct","glu_poct",colnames(x))))
sheet_names_df[sheet_names_df == "hdl"] <- "HDL"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("hdl","HDL",colnames(x))))
sheet_names_df[sheet_names_df == "hematocrit"] <- "Ht"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("hematocrit","Ht",colnames(x))))
sheet_names_df[sheet_names_df == "hemoglobin"] <- "Hb"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("hemoglobin","Hb",colnames(x))))
sheet_names_df[sheet_names_df == "hemoglobin_poct"] <- "Hb_poct"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("hemoglobin_poct","Hb_poct",colnames(x))))
sheet_names_df[sheet_names_df == "inr"] <- "INR"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("inr","INR",colnames(x))))
sheet_names_df[sheet_names_df == "inr_poct"] <- "INR_poct"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("inr_poct","INR_poct",colnames(x))))
sheet_names_df[sheet_names_df == "ld"] <- "LDH"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("ld","LDH",colnames(x))))
sheet_names_df[sheet_names_df == "ldl"] <- "LDL"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("ldl","LDL",colnames(x))))
sheet_names_df[sheet_names_df == "leucocytes"] <- "WBC"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("leucocytes","WBC",colnames(x))))
sheet_names_df[sheet_names_df == "magnesium"] <- "Mg"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("magnesium","Mg",colnames(x))))
sheet_names_df[sheet_names_df == "mch"] <- "MCH"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("mch","MCH",colnames(x))))
sheet_names_df[sheet_names_df == "mcv"] <- "MCV"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("mcv","MCV",colnames(x))))
sheet_names_df[sheet_names_df == "nadir_cd_4_count"] <- "nadir_CD4"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("nadir_cd_4_count","nadir_CD4",colnames(x))))
sheet_names_df[sheet_names_df == "neutrophils"] <- "Neut"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("neutrophils","Neut",colnames(x))))
sheet_names_df[sheet_names_df == "potassium"] <- "K"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("potassium","K",colnames(x))))
sheet_names_df[sheet_names_df == "pth"] <- "PTH"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("pth","PTH",colnames(x))))
```

```
sheet_names_df[sheet_names_df == "pt"] <- "PT"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("pt","PT",colnames(x))))
sheet_names_df[sheet_names_df == "phospatase_alkaline"] <- "ALP"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("phospatase_alkaline","ALP",colnames(x))))
sheet_names_df[sheet_names_df == "phosphorus"] <- "P"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("phosphorus","P",colnames(x))))
sheet_names_df[sheet_names_df == "platelet_count"] <- "PLT"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("platelet_count","PLT",colnames(x))))
sheet_names_df[sheet_names_df == "psa"] <- "psa"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("psa","psa",colnames(x))))
sheet_names_df[sheet_names_df == "psa_ratio"] <- "ratiopsa"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("psa_ratio","ratiopsa",colnames(x))))
sheet_names_df[sheet_names_df == "protein_total"] <- "TP"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("protein_total","TP",colnames(x))))
sheet_names_df[sheet_names_df == "sodium"] <- "Na"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("sodium","Na",colnames(x))))
sheet_names_df[sheet_names_df == "t_4_free"] <- "FT4"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("t_4_free","FT4",colnames(x))))
sheet_names_df[sheet_names_df == "total_cholesterol"] <- "TCHOL"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("total_cholesterol","TCHOL",colnames(x))))
sheet_names_df[sheet_names_df == "triglycerides"] <- "TG"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("triglycerides","TG",colnames(x))))
sheet_names_df[sheet_names_df == "tsh"] <- "TSH"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("tsh","TSH",colnames(x))))
sheet_names_df[sheet_names_df == "urea"] <- "UR"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("urea","UR",colnames(x))))
sheet_names_df[sheet_names_df == "vitamin_b_12"] <- "VitB12"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("vitamin_b_12","VitB12",colnames(x))))
sheet_names_df[sheet_names_df == "1_25_vit_d_nmol"] <- "VitD25"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("1_25_vit_d_nmol","VitD25",colnames(x))))
sheet_names_df[sheet_names_df == "1_25_vit_d_pmol"] <- "VitD3"
tibble_frailty_1 <- lapply(tibble_frailty_1, function(x) setNames (x,
gsub("1_25_vit_d_pmol","VitD3",colnames(x))))

names(tibble_frailty_1) = sheet_names_df$varname
tibble_frailty_1$overview <- tibble_frailty_1$overview %>% dplyr::select(-
contains(c("screening_ol", "cga")))

tibble_frailty_1$overview$GLC_collection_count <-
tibble_frailty_1$overview$GLC_collection_count +
tibble_frailty_1$overview$GLC_poct_collection_count

tibble_frailty_1$overview$Hb_collection_count <- tibble_frailty_1$overview$Hb_collection_count
+ tibble_frailty_1$overview$Hb_poct_collection_count

tibble_frailty_1$overview$INR_collection_count <-
tibble_frailty_1$overview$INR_collection_count +
tibble_frailty_1$overview$INR_poct_collection_count

tibble_frailty_1$overview$CRP_collection_count <-
tibble_frailty_1$overview$CRP_collection_count +
tibble_frailty_1$overview$CRP_poct_collection_count

tibble_frailty_1$overview$eGFR_collection_count <-
tibble_frailty_1$overview$eGFR_collection_count +
tibble_frailty_1$overview$eGFR_mdrd_collection_count
tibble_frailty_1$overview <-tibble_frailty_1$overview %>% dplyr::select(-contains(c('poct',
'mdrd')))
```

```
## Discard empty columns
```

### Ctcue exports a default set of columns, even if data is not requested

````
```{r discard empty columns}
tibble_frailty_2 = lapply(tibble_frailty_1, function(xi) xi[!sapply(xi, function(xii)
all(is.na(xii)))])
```
````

### Mutate column classes

###Overview, removing the selection count and validation label columns in overview.
###Removes all patients where age \<70, because of new export some patients have age 0 (no more information).

````
```{r mutate column classes}
tibble_frailty_2$overview = tibble_frailty_2$overview %>%
  mutate(gender = as.factor(gender)) %>% mutate(age = as.numeric(age))
tibble_frailty_2$overview <- tibble_frailty_2$overview %>% dplyr:: select(-
contains("_selection_count"), -contains("_validation_label"), -contains("nadir"))
tibble_frailty_2 <- tibble_frailty_2[-which(names(tibble_frailty_2) == "nadir_CD4")]
tibble_frailty_2$overview <- tibble_frailty_2$overview[!is.na(tibble_frailty_2$overview$age) &
tibble_frailty_2$overview$age >= 70, ]

```
````

###3 = Remove all the values which have a different (the least occurring) measurement unit. This removes most of the urine samples. Removed specific values originated from urine samples rather than blood samples.

````
```{r}
tibble_frailty_3 = lapply(tibble_frailty_2, function(x) if(ncol(x) ==8) subset (x, x[,c(6)] ==
names(which.max(table(x[, c(6)])))) else x)
tibble_frailty_3 = lapply(tibble_frailty_3, function(x) if(ncol(x) ==7) subset (x, x[,c(5)] ==
names(which.max(table(x[, c(5)])))) else x)

###Removes the specific cases where there are still urine measurements.
tibble_frailty_3$K <- tibble_frailty_3$K %>% filter(.[[3]] %in% c('3.5 - 4.8', '3.5 - 5.0'))
tibble_frailty_3$P <- tibble_frailty_3$P %>% filter(!is.na(.[[3]]))
tibble_frailty_3$Na <- tibble_frailty_3$Na %>% filter(grepl('135 - 145', .[[3]]))
tibble_frailty_3$UR <- tibble_frailty_3$UR %>% filter(!is.na(.[[3]]))

tibble_frailty_3$BILIT <- tibble_frailty_3$BILIT %>% filter(!is.na(.[[3]]))
tibble_frailty_3$Ca <- tibble_frailty_3$Ca %>% filter(!is.na(.[[3]]))
tibble_frailty_3$CRE <- tibble_frailty_3$CRE %>% filter(!is.na(.[[3]]))
tibble_frailty_3$FA <- tibble_frailty_3$FA %>% filter(!is.na(.[[3]]))
tibble_frailty_3$GLC <- tibble_frailty_3$GLC %>% filter(!is.na(.[[3]]))
tibble_frailty_3$LDH <- tibble_frailty_3$LDH %>% filter(!is.na(.[[3]]))
tibble_frailty_3$VitB12 <- tibble_frailty_3$VitB12 %>%  filter(!is.na(.[[3]]) & .[[3]] != '24
- 181')
tibble_frailty_3$ALB <- tibble_frailty_3$ALB %>% filter(!is.na(.[[3]]))

###Removes columns that are not useful anymore.
strings_to_remove <- c("_flag", "_limit")
tibble_frailty_3 <- lapply(tibble_frailty_3, function(df) {
  df[, !(endsWith(names(df), strings_to_remove[1]) | endsWith(names(df),
strings_to_remove[2])), drop = FALSE]
})

```
````

###3 = Changed "X_measurement_start_date" to "start_date" in 6th column of dataframes with 6 columns. To later join the dataframes by pseudo_id and start_date.

###4 = Changed the names of the columns by removing "_measurement_test_value" from the column names; for less typing and better overview.

###5 = Removed all the empty dataframes.

###7 = merge the poct and non poct dataframes together.

````
```{r}
tibble_frailty_3 = lapply(tibble_frailty_3, function(x) if(ncol(x)==6) {colnames(x)[6] <-
'start_date'; x} else x)

tibble_frailty_4 = lapply(tibble_frailty_3, function(x) setNames (x,
gsub("_measurement_test_value","",colnames(x))))

tibble_frailty_5 = keep(tibble_frailty_4,  ~ nrow(.x) > 0)
````
````

````
tibble_frailty_7 = tibble_frailty_5

tibble_frailty_7$CRP <- rbind(tibble_frailty_7$CRP, setNames(tibble_frailty_7$CRP_poct,
names(tibble_frailty_7$CRP)))
tibble_frailty_7$Hb <- rbind(tibble_frailty_7$Hb, setNames(tibble_frailty_7$Hb_poct,
names(tibble_frailty_7$Hb)))
tibble_frailty_7$INR <- rbind(tibble_frailty_7$INR, setNames(tibble_frailty_7$INR_poct,
names(tibble_frailty_7$INR)))
tibble_frailty_7$GLC <- rbind(tibble_frailty_7$GLC, setNames(tibble_frailty_7$glu_poct,
names(tibble_frailty_7$GLC)))
tibble_frailty_7$eGFR <- rbind(tibble_frailty_7$eGFR, setNames(tibble_frailty_7$eGFR_mdrd,
names(tibble_frailty_7$eGFR)))

```
````

### There are at least 1285 patients with 0 lab values. They are removed from the dataframe. Together with the X amount of patients without an frailty outcome.

````
```{r}
cols_of_interest <- tibble_frailty_7$overview[c(1, 9:56)]
zero_rows_pseudo_ids <- c()

for (i in 1:nrow(cols_of_interest)) {
  if (all(cols_of_interest[i, -1] == 0)) {
    zero_rows_pseudo_ids <- c(zero_rows_pseudo_ids, cols_of_interest[i, 1])
  }
}

zero_rows_pseudo_ids_df <- data.frame(pseudo_id = zero_rows_pseudo_ids)
zero_rows_pseudo_ids_df <- t(zero_rows_pseudo_ids_df)
pseudo_ids_to_remove <- zero_rows_pseudo_ids_df[, 1]

tibble_frailty_7 <- map(tibble_frailty_7, function(df) {
  # Remove rows with matching pseudo_ids
  df[!(df[[1]] %in% pseudo_ids_to_remove), ]
})
```
````

### Put all pseudo_ids of 6731 patients into excel file

````
```{r}
pseudo_ids <- data.frame(tibble_frailty_7$overview$pseudo_id)
file_path <- "H:/R/Code Kai/pseudo_ids.xlsx"

write.xlsx(pseudo_ids, file_path)
```
````

### Get the frailty_status of the patients from the pseudo_id_frailty.xlsx which comes from the other R code where the frailty status was connected to the patient_id.

````
```{r}
file_path_excel <- "H:/R/Code Kai/pseudo_id_frailty.xlsx"
data_frail_status <- read.xlsx(file_path_excel, sheet = 1)

pseudo_ids_frailty_status_df <- data.frame(pseudo_id = data_frail_status[, 1], frailty_status
= data_frail_status[, 2])
pseudo_ids_frailty_status_df$frailty_status <-
factor(pseudo_ids_frailty_status_df$frailty_status)

pseudo_ids_to_keep <- pseudo_ids_frailty_status_df[, 1]

tibble_frailty_7 <- map(tibble_frailty_7, function(df) {
  # Remove rows with matching pseudo_ids
  df[(df[[1]] %in% pseudo_ids_to_keep), ]
})
```
````

### Age and gender distribution plot

````
```{r}
tibble_age_gender <- tibble_frailty_7$overview
tibble_age_gender<- tibble_age_gender [c("pseudo_id","age","gender")]

tibble_age_gender["age_group"] = cut(tibble_age_gender$age, c(70, 74, 79, 84, 89, 94, Inf),
c("70-74", "75-79", "80-84", "85-89", "90-94",">95"), include.lowest=TRUE)
````

````
png("age_gender_0108_correct.png")
ggplot(tibble_age_gender, aes(x = age_group, fill = gender))+ geom_bar(position = "dodge",
color= "black") + labs(title = "Age groups by gender", x="Age Groups", y="No. of patients",
fill = "Gender") +scale_fill_grey()
dev.off()

summary(tibble_age_gender)
```
````

### 8 = remove the 'lab comment, metadata and screeningoldestentries dataframes.

### 9 = remove the overview dataframe to deal with only the lab test values.

````
```{r}
tibble_frailty_8 = keep(tibble_frailty_7, ~ ncol(.x) >= 6)
tibble_frailty_8$overview <- tibble_frailty_8$overview %>% dplyr:: select(-
contains("comment"), -contains("cohort"))
tibble_frailty_9 <- tibble_frailty_8[setdiff(names(tibble_frailty_8), c('overview',
'glu_poct', 'INR_poct', 'Hb_poct', 'CRP_poct', 'eGFR_mdrd'))]
```
````

### Dealing with \> and \<, duplicate string column and remove the \< and \>, make into numeric and copy to the old numeric column. Warnings are fine.

````
```{r}
for (i in 1:length(tibble_frailty_9)) {
  # Get the name of the current data frame
  df_name <- names(tibble_frailty_9)[i]

  # Duplicate the 4th column and add it to the data frame with a new name
  tibble_frailty_9[[i]][paste0(df_name, "_string_without")] <- tibble_frailty_9[[i]][[3]]
}

remove_chars <- function(x) {
  gsub("[<>]", "", x)
}

tibble_frailty_10 <- lapply(tibble_frailty_9, function(df) {
  df %>% mutate_at(vars(7), ~remove_chars(.))
})

convert_col7_numeric <- function(df){
  df <- df %>%
    mutate_at(vars(7), as.numeric)
  return(df)
}

tibble_frailty_10 <- lapply(tibble_frailty_10, convert_col7_numeric)

tibble_frailty_10 <- lapply(tibble_frailty_10, function(df) {
  df[, 4] <- df[, 7]
  df <- df[,-7]
  return(df)
})

tibble_frailty_10 <- lapply(tibble_frailty_10, function(x) drop_na(x))

```
````

### Only keep the last observation of each patient for each lab test.

````
```{r}
filter_last_observed <- function(df) {
  df$start_date <- as.POSIXct(df$start_date)

    filtered_df <- df %>%
    group_by(pseudo_id) %>%
    filter(!is.na(start_date) & start_date == max(start_date)) %>%
    ungroup()

  return(filtered_df)
}

only_last_list <- lapply(tibble_frailty_10, filter_last_observed)

only_last_list <- lapply(only_last_list, function(x) drop_na(x))
````

````
```
````

### Summary and unique counts, graph that shows the percentage of patients in which the laboratory test is missing.

````
```{r}
summary_fun <- function(df) {
  summary(df[[4]])
}

summary_list <- lapply(only_last_list, summary_fun)
summary_df <- do.call(rbind, summary_list)

count_unique <- function(df) {
  n_distinct(df[, 1])
}

unique_counts <- lapply(only_last_list, count_unique)
unique_count_df <- do.call(rbind, unique_counts)
unique_count_df <- data.frame(unique_count_df)
unique_count_df$percentage_present <- (unique_count_df[,1]/6326)*100
unique_count_df$percentage_missing <- (100-unique_count_df[,2])

unique_count_df <- rownames_to_column(unique_count_df, var = "lab_test")

unique_count_df <- unique_count_df[order(unique_count_df$percentage_missing),]

unique_count_df$percentage_missing

pdf("percentage_missing_2008.pdf", width = 6, height = 8)
ggplot(unique_count_df, aes(x=reorder(lab_test, percentage_missing), y=percentage_missing)) +
  geom_bar(stat="identity") +
  xlab("Laboratory test") +
  ylab("Percentage of patients with missing laboratory data") +
  theme_classic() +
   theme(panel.grid.major = element_line(linewidth = 0.2,color = "grey"))+
  scale_y_continuous(limits=c(0, 100), breaks=seq(0, 100, by=10), labels=paste0(seq(0, 100, 
by=10), "%")) + coord_flip()
dev.off()

```
````

###removing the lab tests that are not present in \>80% of the patients. And removing others because they are being calculated with other variables or they have such high correlations.

````
```{r}
lab_test_remove <- c("VitD3", "CD4", "PTH", "P", "FT4", "PT", "FA", "VitD25", "ESR", "VitB12", 
"FGLC", "LDL", "MCH", "TG", "BILIT", "Mg", "FER", "TSH", "Ht", "RBC", "eGFR")
only_last_list <- only_last_list[!(names(only_last_list) %in% lab_test_remove)]
only_last_list <- lapply(only_last_list, function(x) drop_na(x))
```
````

### Merge all the dataframes together into one datatable. Columns are pseudo_id, gender, age, age_group, followed by the 25 remaining lab tests.

````
```{r}
tibble_frailty_11 <- only_last_list
merge_all <- lapply(tibble_frailty_11, function(d) {

  d %>%
    dplyr::select(ends_with(c("id", "_numeric")))
})

merge_all = lapply(merge_all, function(x) drop_na(x))

list_of_dt <- lapply(merge_all, as.data.table)
list_of_dt <- lapply(list_of_dt, distinct)

merged_datatable <- Reduce(function(x, y) merge(x, y, by = "pseudo_id", all = TRUE, 
allow.cartesian = TRUE),  list_of_dt)
merged_datatable <- unique(merged_datatable, by = "pseudo_id")

###Add the age, age-group and gender
merged_datatable <- merge(merged_datatable, tibble_age_gender, by = "pseudo_id", all.x = TRUE)

###Add the frailty status
````

````
merged_datatable <- merge(merged_datatable, pseudo_ids_frailty_status_df, by = "pseudo_id", all.x = TRUE)


###Reorder the columns in the merged data table
setcolorder(merged_datatable, c("pseudo_id", "gender", "age", "age_group", "frailty_status", setdiff(names(merged_datatable), c("pseudo_id", "gender", "age", "age_group", "frailty_status"))))

observed_missing_merged <- merged_datatable
colnames(observed_missing_merged)[colnames(observed_missing_merged) == 'RBS_numeric'] <- 'GLC_numeric'
colnames(observed_missing_merged)[colnames(observed_missing_merged) == 'LD_numeric'] <- 'LDH_numeric'

```
````

### Densityplot of all observed values.

````
```{r}
tibble_numeric <- lapply(only_last_list, function(d) {

  d %>%
    dplyr::select(ends_with(c("_numeric", "pseudo_id")))
})

densityplot_list <- lapply(tibble_numeric, function(df) merge(df, pseudo_ids_frailty_status_df, by = "pseudo_id"))
densityplot_list = lapply(densityplot_list, function(x) drop_na(x))

plotje <- function(df){ df %>% ggplot() +
  geom_density(aes(x =.data[[names(df)[2]]], y = ..ndensity.., group = frailty_status, color = frailty_status), alpha = 0.5)}

pdf("densityplots_all.pdf")
lapply(densityplot_list, plot)
dev.off()

##################################

tail_removal_cols <- c("BILID_numeric", "BILIT_numeric", "CK_numeric", "CRE_numeric", "CRP_numeric", "FER_numeric", "GGT_numeric", "LD_numeric", "WBC_numeric", "ALP_numeric", "TSH_numeric", "ALAT_numeric", "ASAT_numeric" )

pdf("densityplots_remaining_without_tail_0607.pdf")
plot_density_all <- function(df) {
  col <- names(df)[2]
 #colnames <- sub("_numeric", "", col)
  plot_title <- paste("Densityplot of frail and non frail patients of ", col)

  if (col %in% tail_removal_cols) {
    # Compute the lower and upper bounds for the xlim
    col_data <- df[[col]]
    lower_bound <- quantile(col_data, 0.00, na.rm = TRUE)
    upper_bound <- quantile(col_data, 0.98, na.rm = TRUE)

    p <- ggplot() +
      geom_density(data = df, aes(x = .data[[col]], y = ..ndensity.., fill = frailty_status), alpha = 0.5) +
      labs(title = plot_title, x = "Lab Test Value", y = "Density") +
      theme(legend.position = c(.9, .9), plot.title = element_text(size = 12)) +
      xlim(lower_bound, upper_bound)  # Set the xlim to remove the tail
  } else {
    p <- ggplot() +
      geom_density(data = df, aes(x = .data[[col]], y = ..ndensity.., fill = frailty_status), alpha = 0.5) +
      labs(title = plot_title, x = "Lab Test Value", y = "Density") +
      theme(legend.position = c(.9, .9), plot.title = element_text(size = 12))
  }

  print(p)
}

for (df in densityplot_list) {
  plot_density_all(df)
}
````

````
dev.off()

```
````

```
### ########################################IMPUTATION

### Mean imputation
```

````
```{r}
observed_missing_merged <- data.table(observed_missing_merged)
observed_missing_merged <- data.table(observed_missing_merged)
mean_values <- copy(observed_missing_merged)

#skewed_observations <- data.table((skewed_observations))
#mean_values <- copy(skewed_observations)


###The mean of the columns of all patients combined
mean_values_columns <- mean_values[, lapply(.SD, mean, na.rm = TRUE), .SDcols =
6:ncol(mean_values)]
###The mean imputed datatable.
mean_imputed_columns <- copy(mean_values)
for (col in names(mean_values_columns)) {
  mean_imputed_columns[is.na(mean_imputed_columns[[col]]), col] <- mean_values_columns[[col]]
}

pseudo_id_order <- observed_missing_merged$pseudo_id
imputation_indicator <- copy(mean_values)
# Iterate over columns to identify imputed values (1 is present, 0 is imputed)
for (col in names(mean_values)) {
  imputed_rows <- is.na(mean_values[[col]])
  imputation_indicator[[col]] <- ifelse(imputed_rows, 0, 1)
}
imputation_indicator[, 1:5 := 0]

solely_imputed_mean_dt <- copy(mean_imputed_columns)
###The mean imputed datatable, but without the observed values. The reverse of the
observed_missing_merged datatable.
solely_imputed_mean_dt[imputation_indicator == 1] <- NA


###The mean of patient based on gender and age group
mean_values_genderage <- mean_values[, lapply(.SD, mean, na.rm = TRUE), by = .(gender,
age_group), .SDcols = 6:ncol(mean_values)]
###The mean based on gender and age group imputed datatable.
mean_imputed_genderage_dt <- copy(observed_missing_merged)
mean_imputed_genderage_dt <- merge(mean_imputed_genderage_dt, mean_values_genderage, by =
c("gender", "age_group"))
cols_to_replace <- grep("\\.x$", names(mean_imputed_genderage_dt), value = TRUE)
for (col in cols_to_replace) {
  col_y <- sub("\\.x$", ".y", col)
  mean_imputed_genderage_dt[is.na(get(col)), (col) := get(col_y)]
}
mean_imputed_genderage_dt[, grep("\\.y$", names(mean_imputed_genderage_dt)) := NULL]
setnames(mean_imputed_genderage_dt, old = names(mean_imputed_genderage_dt), new = sub("\\.x$",
"", names(mean_imputed_genderage_dt)))
col_names <- colnames(mean_imputed_genderage_dt)
new_col_order <- c(col_names[3], col_names[-3])
mean_imputed_genderage_dt <- mean_imputed_genderage_dt[, ..new_col_order]

#Reorder such that we can compare with the imputation_indicator later.
mean_imputed_genderage_dt <- mean_imputed_genderage_dt[match(pseudo_id_order,
mean_imputed_genderage_dt$pseudo_id),]

solely_imputed_mean_genderage_dt <- copy(mean_imputed_genderage_dt)
###The mean based on gender and age group imputed datatable, but without the observed values.
The reverse of the observed_missing_merged datatable.

solely_imputed_mean_genderage_dt[imputation_indicator == 1] <- NA

common_cols <- intersect(names(mean_imputed_genderage_dt)[6:ncol(mean_imputed_genderage_dt)],
names(observed_missing_merged)[6:ncol(observed_missing_merged)])

```
````

```
### Median imputation
```
```

```{r}
#median_values <- copy(observed_missing_merged)
observed_missing_merged <- data.table((observed_missing_merged))
median_values <- copy(observed_missing_merged)

#median_values <- data.table((skewed_observations))
#median_values <- copy(median_values)

###The median of the columns of all patients combined.
median_values_columns <- median_values[, lapply(.SD, median, na.rm = TRUE), .SDcols = 6:ncol(median_values)]
###The median imputed datatable.
median_imputed_columns <- copy(median_values)
for (col in names(median_values_columns)) {
  median_imputed_columns[is.na(median_imputed_columns[[col]]), col] <- median_values_columns[[col]]
}

solely_imputed_median_dt <- copy(median_imputed_columns)
###The median imputed datatable, but without the observed values. The reverse of the observed_missing_merged datatable.
solely_imputed_median_dt[imputation_indicator == 1] <- NA



###The median of patient based on gender and age group
median_values_genderage <- median_values[, lapply(.SD, median, na.rm = TRUE),by = .(gender, age_group), .SDcols = 6:ncol(median_values)]
###The median based on gender and age group imputed datatable.
median_imputed_genderage_dt <- copy(observed_missing_merged)
median_imputed_genderage_dt <- merge(median_imputed_genderage_dt, median_values_genderage, by = c("gender", "age_group"))
cols_to_replace <- grep("\\.x$", names(median_imputed_genderage_dt), value = TRUE)
for (col in cols_to_replace) {
  col_y <- sub("\\.x$", ".y", col)
  median_imputed_genderage_dt[is.na(get(col)), (col) := get(col_y)]
}
median_imputed_genderage_dt[, grep("\\.y$", names(median_imputed_genderage_dt)) := NULL]
setnames(median_imputed_genderage_dt, old = names(median_imputed_genderage_dt), new = sub("\\.x$", "", names(median_imputed_genderage_dt)))
col_names <- colnames(median_imputed_genderage_dt)
new_col_order <- c(col_names[3], col_names[-3])
median_imputed_genderage_dt <- median_imputed_genderage_dt[, ..new_col_order]

#Reorder such that we can compare with the imputation_indicator later.
median_imputed_genderage_dt <- median_imputed_genderage_dt[match(pseudo_id_order, median_imputed_genderage_dt$pseudo_id),]

solely_imputed_median_genderage_dt <- copy(median_imputed_genderage_dt)
###The median based on gender and age group imputed datatable, but without the observed values. The reverse of the observed_missing_merged datatable.
solely_imputed_median_genderage_dt[imputation_indicator == 1] <- NA

common_cols <- intersect(names(mean_imputed_genderage_dt)[6:ncol(mean_imputed_genderage_dt)], names(observed_missing_merged)[6:ncol(observed_missing_merged)])

```

#random forest

```
```{r}
install.packages("missForest")
library(missForest)

to_impute_dataframe <- data.frame(observed_missing_merged)

#cols_to_impute <- names(to_impute_dataframe)[6:ncol(to_impute_dataframe)]
cols_to_impute <- to_impute_dataframe[, -(1:5)]
terug <- to_impute_dataframe[, 1:5]


start.time <- Sys.time()
# Impute missing values using missForest for the selected columns
imputed_data_missForest <- missForest(cols_to_impute, maxiter = 5, ntree = 100, verbose = TRUE)
end.time <- Sys.time()
```
```

```
time.taken <- round(end.time - start.time,2)
time.taken

solely_forest <- cbind(terug,forest)

imputation_indicator_2 <- copy(to_impute_dataframe)
# Iterate over columns to identify imputed values (1 is present, 0 is imputed)
for (col in names(to_impute_dataframe)) {
  imputed_rows <- is.na(to_impute_dataframe[[col]])
  imputation_indicator_2[[col]] <- ifelse(imputed_rows, 0, 1)
}
imputation_indicator_2[, 1:5 := 0]

solely_forest <- data.table(solely_forest)

solely_forest[imputation_indicator_2 == 1] <- NA

```

### Multiple imputation : pmm

```{r}
common_cols <- names(observed_missing_merged)[6:ncol(observed_missing_merged)]
pseudo_id_order <- observed_missing_merged$pseudo_id

to_impute_dataframe <- observed_missing_merged

mice_obj <- mice(to_impute_dataframe, m = 1,printFlag = FALSE)

predictor_matrix <- mice_obj$pred
predictor_matrix[ ,"age_group"] <- 0
predictor_matrix["age",] <- 0
predictor_matrix["gender",] <- 0
predictor_matrix["age_group",] <- 0
predictor_matrix["pseudo_id",] <- 0
predictor_matrix[,"frailty_status"] <- 0

visit_seq <- mice_obj$visitSequence
visit_seq2 <- visit_seq
visit_seq2 <- c("pseudo_id", "gender", "age", "age_group" ,"frailty_status",
              "ALAT_numeric", "ASAT_numeric",
               "Hb_numeric", "MCV_numeric", "WBC_numeric", "ALB_numeric", "CRP_numeric",
"PLT_numeric",
               "HDL_numeric", "TP_numeric", "TCHOL_numeric",
               "BILID_numeric",
               "CRE_numeric", "UR_numeric", "CK_numeric",
               "Ca_numeric", "Cl_numeric", "eGFR_numeric", "GGT_numeric", "INR_numeric",
"LD_numeric", "Neut_numeric", "K_numeric", "ALP_numeric", "Na_numeric", "RBS_numeric")


num_iterations <- 20
num_datasets <- 10
method_mice <- "pmm"

pmm_data_1 <- mice(
  data = to_impute_dataframe,
  m = num_datasets,
  maxit = num_iterations,
  method = method_mice,
  seed = 123,
  pred = predictor_matrix,
  printFlag = FALSE,
  visitSequence = visit_seq2
)

pmm_data_single <- complete(pmm_data_1)
write.xlsx(pmm_data_single, file = "pmm_data_single_2107.xlsx")


pmm_data_single_incl <- read_excel("pmm_data_single_2107.xlsx")
pmm_data_single_incl$eGFR_numeric <- NULL

pmm_data2 <- pmm_data_single_incl
#Reorder such that we can compare with the imputation_indicator later.
pmm_data2 <- pmm_data2[match(pseudo_id_order, pmm_data2$pseudo_id),]

pmm_data_single_excl <- copy(pmm_data2)
```

```
###The mean based on gender and age group imputed datatable, but without the observed values.
The reverse of the observed_missing_merged datatable.

pmm_data_single_excl[imputation_indicator == 1] <- NA

```
```

###Multiple imputation: Norm

```
```{r}
to_impute_dataframe <- observed_missing_merged

mice_obj <- mice(to_impute_dataframe, m = 1,printFlag = FALSE)

predictor_matrix <- mice_obj$pred
predictor_matrix[ ,"age_group"] <- 0
predictor_matrix["age",] <- 0
predictor_matrix["gender",] <- 0
predictor_matrix["age_group",] <- 0
predictor_matrix["pseudo_id",] <- 0
predictor_matrix[,"frailty_status"] <- 0

visit_seq <- mice_obj$visitSequence
visit_seq2 <- visit_seq
visit_seq2 <- c("pseudo_id", "gender", "age", "age_group" ,"frailty_status",
              "ALAT_numeric", "ASAT_numeric",
               "Hb_numeric", "MCV_numeric", "WBC_numeric", "ALB_numeric", "CRP_numeric",
"PLT_numeric",
               "HDL_numeric", "TP_numeric", "TCHOL_numeric",
               "BILID_numeric",
               "CRE_numeric", "UR_numeric", "CK_numeric",
               "Ca_numeric", "Cl_numeric", "eGFR_numeric", "GGT_numeric", "INR_numeric",
"LD_numeric", "Neut_numeric", "K_numeric", "ALP_numeric", "PT_numeric", "Na_numeric",
"RBS_numeric")


num_iterations <- 20
num_datasets <- 10
method_mice <- "norm"

norm_data_1 <- mice(
  data = to_impute_dataframe,
  m = num_datasets,
  maxit = num_iterations,
  method = method_mice,
  seed = 123,
  pred = predictor_matrix,
  printFlag = FALSE,
  visitSequence = visit_seq2
)

norm_data_single <- complete(norm_data_1)
write.xlsx(norm_data_single, file = "norm_data_single_2107.xlsx")


norm_data_single_incl <- read_excel("norm_data_single_2107.xlsx")
norm_data_single_incl$eGFR_numeric <- NULL

norm_data2 <- norm_data_single_incl
#Reorder such that we can compare with the imputation_indicator later.
norm_data2 <- norm_data2[match(pseudo_id_order, norm_data2$pseudo_id),]

norm_data_single_excl <- copy(norm_data2)
###The mean based on gender and age group imputed datatable, but without the observed values.
The reverse of the observed_missing_merged datatable.

norm_data_single_excl[imputation_indicator == 1] <- NA

```
```

###Separate all the densityplots - INCL imputed values

```
```{r}
# Load necessary packages
library(ggplot2)
library(gridExtra)
```

```
tail_removal_cols <- c("BILID_numeric", "BILIT_numeric", "CK_numeric", "CRE_numeric",
"CRP_numeric", "FER_numeric", "GGT_numeric", "LDH_numeric", "WBC_numeric", "ALP_numeric",
"TSH_numeric", "ALAT_numeric", "ASAT_numeric" )

pdf("Density_INCL_good_2008_3.pdf")

# Initialize an empty list to store the plots
plot_list <- list()

theme_set(
  theme(
    plot.title = element_text(size = 8),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 5)
  )
)

# Loop over the common_cols and create the plots
for (col in common_cols) {
 col_title <- gsub("_numeric", "", col)

  title_p1 <- paste("Mean Imputed gender-age", col_title)  # Separate title for p1
  title_p2 <- paste("Median Imputed gender-age", col_title)  # Separate title for p2
  title_p3 <- paste("Random Forest", col_title)  # Separate title for p3
  title_p4 <- paste("pmm single", col_title)  # Separate title for p4
  title_p5 <- paste("norm single", col_title)  # Separate title for p5

      p1 <- ggplot() +
      geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
        geom_density(data = mean_imputed_genderage_dt, aes(x = .data[[col]], y = ..ndensity..,
fill = "Mean Imputed gender-age"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE)
+
      labs(title = title_p1, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "Mean Imputed gender-
age" = "yellow"))


  ##########Mediaan

      p2 <- ggplot() +
       geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      geom_density(data = median_imputed_genderage_dt, aes(x = .data[[col]], y = ..ndensity..,
fill = "Median Imputed gender-age"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm =
TRUE) +
      labs(title = title_p2, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "Median Imputed
gender-age" = "blue"))
#####FOREST

      p3 <- ggplot() +
       geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      geom_density(data = forest, aes(x = .data[[col]], y = ..ndensity.., fill = "Random
Forest"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      labs(title = title_p3, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "Random Forest" =
"green"))
###pmm

      p4 <- ggplot() + geom_density(data = observed_missing_merged, aes(x = .data[[col]], y =
..ndensity.., fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm =
TRUE) + geom_density(data = pmm_data_single_incl, aes(x = .data[[col]], y = ..ndensity.., fill
= "pmm single"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      labs(title = title_p4, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "pmm single" =
"purple"))

###Norm

      p5 <- ggplot() +
       geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      geom_density(data = norm_data_single_incl, aes(x = .data[[col]], y = ..ndensity.., fill
= "norm single"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      labs(title = title_p5, x = "Lab Test Value", y = "Density") +
```

````
       scale_fill_manual("Values", values = c("Observations" = "black", "norm single" =
"red"))


  # Add the plots to the plot_list
  plot_list[[col]] <- grid.arrange(p1, p2, p3, p4, p5, nrow=2, top = textGrob("Observed values
compared to single imputation values", gp=gpar(fontsize=12,font=8)))
}


for (col in common_cols) {
  print(plot_list[[col]])
}
dev.off()

```
````

###Separate all the densityplots - EXCL imputed values

````
```{r}
library(grid)

pdf("Density_EXCL_good_2008_3.pdf")

# Initialize an empty list to store the plots
plot_list <- list()

theme_set(
  theme(
    legend.position = c(.85, .85),
    plot.title = element_text(size = 8),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 5)
  )
)

# Loop over the common_cols and create the plots
for (col in common_cols) {
 col_title <- gsub("_numeric", "", col)

  title_p1 <- paste("Mean Imputed gender-age", col_title)  # Separate title for p1
  title_p2 <- paste("Median Imputed gender-age", col_title)  # Separate title for p2
  title_p3 <- paste("Random Forest", col_title)  # Separate title for p3
  title_p4 <- paste("pmm single", col_title)  # Separate title for p4
  title_p5 <- paste("norm single", col_title)  # Separate title for p5

      p1 <- ggplot() +
      geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
        geom_density(data = solely_imputed_mean_genderage_dt, aes(x = .data[[col]], y =
..ndensity.., fill = "Mean Imputed gender-age"), alpha = 0.3, show.legend = FALSE, adjust = 2,
na.rm = TRUE) +
      labs(title = title_p1, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "Mean Imputed gender-
age" = "yellow"))


  ##########Mediaan
      p2 <- ggplot() +
       geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      geom_density(data = solely_imputed_median_genderage_dt, aes(x = .data[[col]], y =
..ndensity.., fill = "Median Imputed gender-age"), alpha = 0.3, show.legend = FALSE, adjust =
2, na.rm = TRUE) +
      labs(title = title_p2, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "Median Imputed
gender-age" = "blue"))
#####FOREST

      p3 <- ggplot() +
       geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      geom_density(data = solely_forest, aes(x = .data[[col]], y = ..ndensity.., fill =
"Random Forest"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      labs(title = title_p3, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "Random Forest" =
"green"))
````

```
###pmm

      p4 <- ggplot() +
       geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      geom_density(data = pmm_data_single_excl, aes(x = .data[[col]], y = ..ndensity.., fill =
"pmm single"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      labs(title = title_p4, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "pmm single" =
"purple"))

###Norm
      p5 <- ggplot() +
       geom_density(data = observed_missing_merged, aes(x = .data[[col]], y = ..ndensity..,
fill = "Observations"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      geom_density(data = norm_data_single_excl, aes(x = .data[[col]], y = ..ndensity.., fill
= "norm single"), alpha = 0.3, show.legend = FALSE, adjust = 2, na.rm = TRUE) +
      labs(title = title_p5, x = "Lab Test Value", y = "Density") +
       scale_fill_manual("Values", values = c("Observations" = "black", "norm single" =
"red"))

  # Add the plots to the plot_list
  plot_list[[col]] <- grid.arrange(p1, p2, p3, p4, p5, nrow=2, top = textGrob("Observed values
compared to single imputation values", gp=gpar(fontsize=12,font=8)))

}

for (col in common_cols) {
  print(plot_list[[col]])
}
dev.off()
```

###Boxplot - EXCL imputed values

```{r}
pdf("Boxplot_EXCL_2008.pdf")
for (col in common_cols) {
  plot_title <- paste("Observed values compared to single imputation values", col)
  {
    p <- ggplot() +
      geom_boxplot(data = observed_missing_merged, aes(x = 1, y = .data[[col]], fill =
"Observations"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = solely_imputed_mean_genderage_dt, aes(x = 2, y = .data[[col]], fill
= "Mean Imputed gender-age"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = solely_imputed_median_genderage_dt, aes(x = 3, y = .data[[col]],
fill = "Median Imputed gender-age"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm =
TRUE) +
      geom_boxplot(data = solely_forest, aes(x = 4, y = .data[[col]], fill = "Random Forest"),
width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = pmm_data_single_excl, aes(x = 5, y = .data[[col]], fill = "pmm
single"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = norm_data_single_excl, aes(x = 6, y = .data[[col]], fill = "norm
single"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
     labs(title = plot_title, x = "", y = "Lab Test Value") +
         scale_fill_manual("Values", values = c("Observations" = "black",  "Mean Imputed
gender-age" = "yellow",  "Median Imputed gender-age" = "blue", "Random Forest" = "green", "pmm
single" = "purple", "norm single" = "red"))+
     scale_x_continuous(breaks = c(1, 2, 3, 4, 5,6),
                     labels = c("Observations" ,  "Mean Imputed gender-age" ,  "Median Imputed
gender-age" , "Random Forest", "pmm single", "norm single" )) +
       theme(legend.position = "none", plot.title = element_text(size = 11),  axis.text.x =
element_text(angle = 45, hjust = 1))
  }

  print(p)
}
dev.off()

```

###Boxplot - INCL imputed values

```{r}
pdf("Boxplot_INCL_2008.pdf")
for (col in common_cols) {
```

```
  plot_title <- paste("Observed values compared to single imputation values with observed
values", col)
  {
    p <- ggplot() +
      geom_boxplot(data = observed_missing_merged, aes(x = 1, y = .data[[col]], fill =
"Observations"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = mean_imputed_genderage_dt, aes(x = 2, y = .data[[col]], fill = "Mean
Imputed gender-age"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = median_imputed_genderage_dt, aes(x = 3, y = .data[[col]], fill =
"Median Imputed gender-age"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = forest, aes(x = 4, y = .data[[col]], fill = "Random Forest"), width
= 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = pmm_data_single_incl, aes(x = 5, y = .data[[col]], fill = "pmm
single"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
      geom_boxplot(data = norm_data_single_incl, aes(x = 6, y = .data[[col]], fill = "norm
single"), width = 0.6, alpha = 0.5, show.legend = TRUE, na.rm = TRUE) +
     labs(title = plot_title, x = "", y = "Lab Test Value") +
         scale_fill_manual("Values", values = c("Observations" = "black",  "Mean Imputed
gender-age" = "yellow",  "Median Imputed gender-age" = "blue", "Random Forest" = "green", "pmm
single" = "purple", "norm single" = "red"))+
     scale_x_continuous(breaks = c(1, 2, 3, 4, 5,6),
                     labels = c("Observations" ,  "Mean Imputed gender-age" ,  "Median Imputed
gender-age" , "Random Forest", "pmm single", "norm single" )) +
       theme(legend.position = "none", plot.title = element_text(size = 11),  axis.text.x =
element_text(angle = 45, hjust = 1))
  }

  print(p)
}
dev.off()

```

###Median correlation

```
```{r}

tibble_median <- lapply(only_last_list, function(d) {

  d %>%
     dplyr::select(ends_with(c("_id", "_numeric")))
})

tibble_median = lapply(tibble_median, function(x) setNames (x,
gsub("_numeric","",colnames(x))))

tibble_median = (lapply(tibble_median, function(x) x %>%dplyr::group_by(pseudo_id) %>%
dplyr::summarise_at(.vars = names(.)[2],  median, na.rm=TRUE)))

output_median <- tibble_median %>% reduce(full_join, by = c("pseudo_id"))


output_median[is.na(output_median)] = 0
output_median[output_median == Inf] <- 0
output_median[output_median == -Inf] <- 0

correlation_median <- cor(output_median[sapply(output_median, is.numeric)], use =
"complete.obs")
pdf("correlation_median_file_large_1508.pdf", height = 45, width = 45)
corrplot(correlation_median,order = 'alphabet', method="circle",addCoef.col = 'black',  type =
"upper", na.label=" ")
dev.off()


```
```

###Correlation patient time

```
```{r}
lab_test_remove_cor <- c("VitD3", "CD4", "PTH", "P", "FT4", "PT", "FA", "VitD25", "ESR",
"VitB12", "FGLC", "TG", "BILIT", "Mg", "FER", "TSH")
only_last_list <- only_last_list[!(names(only_last_list) %in% lab_test_remove_cor)]
only_last_list <- lapply(only_last_list, function(x) drop_na(x))

tibble_correlate_patient_time <- lapply(only_last_list, function(d) {
```
```

````
  d %>%
     dplyr::select(ends_with(c("_numeric", "id", "date")))
})

output_patient_time <- join_all(tibble_correlate_patient_time, by = NULL, type = "full", match
= "first")  %>% distinct()

correlation_patient_time <- cor(output_patient_time[sapply(output_patient_time, is.numeric)],
use = "pairwise.complete.obs")

correlation_patient_time2 <- correlation_patient_time
correlation_patient_time2[is.na(correlation_patient_time2)] = 0

colnames(correlation_patient_time2) = gsub("_numeric","",colnames(correlation_patient_time2))
rownames(correlation_patient_time2) = gsub("_numeric","",rownames(correlation_patient_time2))

pdf("correlation_file_patient_time_large_2508.pdf", height = 45, width = 45)
corrplot(correlation_patient_time2, order = "alphabet", method="color",addCoef.col = 'black',
type = "upper", na.label=" ", tl.col="black", tl.srt=45, diag=FALSE, number.cex = 2)
dev.off()

```
````
````

# Appendix G. Frailty status R code

```
---
title: ""
author: "Kai van de Poll"
format: html
editor: visual
date: "17-05-2023"
---
```

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

```
## Set Library
##It is necessary to set .libPaths by hand before running this .qmd file:
```

```
```{r set path to library}
.libPaths("H:/R/Library")
```
```

```
## Load required Packages
```

```
```{r load packages}
library(knitr)
library(markdown)
library(tibble)
library(ggplot2)
library(tidyr)
library(clipr)
library(tidymodels)
library(readxl)
library(summarytools)
library(tidyverse)
library(skimr)
library(corrplot)
library(purrr)
library(plyr)
library(naniar)
library(RColorBrewer)
library(gridExtra)
library(data.table)
library(rpart)
library(rpart.plot)
library(fitdistrplus)
library(mice)
library(dplyr)
library(randomForest)
library(patchwork)
library(openxlsx)
```
```

```
## Set path to load dataset
```

```
```{r set path to load dataset}
path1 = "H:/R/Code Kai/voorspelmodel-kwetsbaarheid-bij-ouderen-frailtylabels-pt1-
20230530.xlsx"
path2 = "H:/R/Code Kai/voorspelmodel-kwetsbaarheid-bij-ouderen-frailtylabels-pt2-
20230602.xlsx"
path3 = "H:/R/Code Kai/voorspelmodel-kwetsbaarheid-bij-ouderen-frailtylabels-pt3-
20230524.xlsx"
path4 = "H:/R/Code Kai/voorspelmodel-kwetsbaarheid-bij-ouderen-frailtylabels-pt4-
20230526.xlsx"
```
```

```
## Import all sheets
```

```
```{r import data}
sheet_names1 = excel_sheets(path1)
sheet_names2 = excel_sheets(path2)
sheet_names3 = excel_sheets(path3)
sheet_names4 = excel_sheets(path4)
tibble_frailty_1 = lapply(sheet_names1, function(x) {as_tibble(read_excel(path1, sheet = x))})
names(tibble_frailty_1) = sheet_names1
tibble_frailty_2 = lapply(sheet_names2, function(x) {as_tibble(read_excel(path2, sheet = x))})
names(tibble_frailty_2) = sheet_names2
```
```

````
tibble_frailty_3 = lapply(sheet_names3, function(x) {as_tibble(read_excel(path3, sheet = x))})
names(tibble_frailty_3) = sheet_names3
tibble_frailty_4 = lapply(sheet_names4, function(x) {as_tibble(read_excel(path4, sheet = x))})
names(tibble_frailty_4) = sheet_names4

sheet_names_df = as.data.frame(sheet_names1)
sheet_names_df[sheet_names_df == "metadata"] <- "1-metadata"
sheet_names_df[sheet_names_df == "overview"] <- "2-overview"
sheet_names_df = sheet_names_df %>% separate(sheet_names1, c("varnumber", "varname"), "-")

names(tibble_frailty_1) = sheet_names_df$varname
names(tibble_frailty_2) = sheet_names_df$varname
names(tibble_frailty_3) = sheet_names_df$varname
names(tibble_frailty_4) = sheet_names_df$varname

merged_list <- map2(tibble_frailty_1, tibble_frailty_2, bind_rows) %>% map2(tibble_frailty_3,
bind_rows) %>%      map2(tibble_frailty_4, bind_rows)

merged_list$metadata = NULL

```
````

### Only keep the remaining 6731 patients

````
```{r}
file_path_excel <- "H:/R/Code Kai/pseudo_ids.xlsx"
pseudo_ids_excel <- read.xlsx(file_path_excel, sheet = 1)$tibble_frailty_7.overview.pseudo_id

pseudo_ids_df <- data.frame(pseudo_id = pseudo_ids_excel)

list_of_all_patients <- lapply(merged_list, function(df) {
  semi_join(df, pseudo_ids_df, by = names(df)[1])
})

```
````

````
```{r}
list_of_all_patients$screening_oldest_entries = NULL
list_of_all_patients$g_8_mijn_jbz_toelichting = NULL
list_of_all_patients$tfi_advies = NULL
list_of_all_patients$gfi_opmerkingen = NULL
list_of_all_patients$g_8_toelichting = NULL
list_of_all_patients$apop_cognitie = NULL
list_of_all_patients$apop_cognitie = NULL
list_of_all_patients$cfs_clinical_frailty_scale = NULL

#Because i removed some dataframes, such that there will only be patients with an 'outcome'
overview_pseudo_ids <- list_of_all_patients[[1]]$pseudo_id
unique_pseudo_ids <- vector()

for (i in 2:length(list_of_all_patients)) {
  current_pseudo_ids <- list_of_all_patients[[i]][[1]]

  intersect_pseudo_ids <- intersect(overview_pseudo_ids, current_pseudo_ids)

  overview_pseudo_ids <- setdiff(overview_pseudo_ids, intersect_pseudo_ids)
}

unique_pseudo_df <- data.frame(pseudo_id = overview_pseudo_ids)

#remove them from overview
list_of_all_patients <- lapply(list_of_all_patients, function(df) {
  df <- anti_join(df, data.frame(pseudo_id = unique_pseudo_df), by = "pseudo_id")
  return(df)
})

```
````

###Plot the age_group and gender

````
```{r}
tibble_age_gender <- list_of_all_patients$overview
tibble_age_gender<- tibble_age_gender [c("pseudo_id","age","gender")]

tibble_age_gender["age_group"] = cut(tibble_age_gender$age, c(70, 74, 79, 84, 89, 94, Inf),
c("70-74", "75-79", "80-84", "85-89", "90-94",">95"), include.lowest=TRUE)
````

```
#pdf("age_gender_3005.pdf")
#ggplot(tibble_age_gender, aes(x = age_group, fill = gender))+ geom_bar(position = "dodge") +
labs(title = "Age groups by gender", x="Age Groups", y="No. of patients", fill = "Gender") +
scale_fill_manual(values=c("#c1272d","#0000a7"))
#dev.off()

summary(tibble_age_gender)
```

### There are patients who have had 2 frailty screeners on the same day, such as the APOP and the CFS. The frailty screener other than the APOP will remain because that one is most likely to be performed after the APOP.

```{r}
frail_or_not <- list_of_all_patients
frail_or_not$overview = NULL

frail_or_not_merged1 <- Reduce(function(df1, df2) merge(df1, df2, by = "pseudo_id", all =
TRUE), frail_or_not)
frail_or_not_merged <- merge(frail_or_not_merged1, tibble_age_gender, by = "pseudo_id")

frail_or_not_merged <- frail_or_not_merged %>%
  mutate(apop_uitkomst_form_entries_start_date =
as.Date(apop_uitkomst_form_entries_start_date)) %>%
  arrange(pseudo_id, desc(apop_uitkomst_form_entries_start_date)) %>%
  distinct(pseudo_id, .keep_all = TRUE)

apop_columns <- names(frail_or_not_merged)[grepl("apop", names(frail_or_not_merged))]

frail_or_not_merged[apop_columns] <- lapply(frail_or_not_merged[apop_columns],
function(column) {
  column[frail_or_not_merged$apop_uitkomst_form_start_date <
frail_or_not_merged$cfs_categorie_form_start_date |
            frail_or_not_merged$apop_uitkomst_form_start_date <
frail_or_not_merged$g_8_mijn_jbz_totaal_score_form_start_date |
            frail_or_not_merged$apop_uitkomst_form_start_date <
frail_or_not_merged$gfi_totaal_score_form_start_date |
            frail_or_not_merged$apop_uitkomst_form_start_date <
frail_or_not_merged$tfi_totaalscore_form_start_date] <- NA
  return(column)
})


```

###Set the boundaries for each frailty screener when it should give frail or not frail.

```{r}
frailty_status <- character(nrow(frail_or_not_merged))

for (i in 1:nrow(frail_or_not_merged)) {
  if (!is.na(frail_or_not_merged[i, "tfi_totaalscore_form_entries_value_number"]) &&
frail_or_not_merged[i, "tfi_totaalscore_form_entries_value_number"] > 4) {
    frailty_status[i] <- "frail"
  } else if (!is.na(frail_or_not_merged[i, "gfi_totaal_score_form_entries_value_number"]) &&
frail_or_not_merged[i, "gfi_totaal_score_form_entries_value_number"] > 3) {
    frailty_status[i] <- "frail"
  } else if (!is.na(frail_or_not_merged[i, "apop_uitkomst_form_entries_value_text"]) &&
grepl("groot|hoog", frail_or_not_merged[i, "apop_uitkomst_form_entries_value_text"])) {
    frailty_status[i] <- "frail"
  } else if (!is.na(frail_or_not_merged[i, "cfs_categorie_form_entries_value_text"]) &&
grepl("5|6|7|8", frail_or_not_merged[i, "cfs_categorie_form_entries_value_text"], ignore.case
= TRUE)) {
    frailty_status[i] <- "frail"
  } else if (!is.na(frail_or_not_merged[i, "g_8_totaal_score_form_entries_value_number"]) &&
frail_or_not_merged[i, "g_8_totaal_score_form_entries_value_number"] < 15) {
    frailty_status[i] <- "frail"
  } else if (!is.na(frail_or_not_merged[i,
"g_8_mijn_jbz_totaal_score_form_entries_value_number"]) && frail_or_not_merged[i,
"g_8_mijn_jbz_totaal_score_form_entries_value_number"] < 15) {
    frailty_status[i] <- "frail"
  } else {
    frailty_status[i] <- "not frail"
  }
}

frail_or_not_merged$frailty_status <- frailty_status
frail_or_not_merged <- frail_or_not_merged %>%
```

````
  relocate(frailty_status, .before = 2)

```
````

### Percentage of patients that have a frail outcome

````
```{r}
frail_count <- sum(frail_or_not_merged$frailty_status == "frail")

frail_percentage <- (frail_count / nrow(frail_or_not_merged)) * 100

cat("Percentage of frail patients:", frail_percentage, "%\n")

```
````

### Frail and age_group plot

````
```{r}
#pdf("age_gender_3005.pdf")
ggplot(frail_or_not_merged, aes(x = age_group, fill = frailty_status))+ geom_bar(position =
"dodge") + labs(title = "Age groups by gender", x="Age Groups", y="No. of patients", fill =
"Frailty status") + scale_fill_manual(values=c("#c1272d","#0000a7"))
#dev.off()

png("percentage_frail_1608.png")
ggplot(frail_or_not_merged, aes(x = age_group, fill = frailty_status)) +
  geom_bar(position = position_fill(reverse = TRUE)) +
  scale_fill_manual(values = c("black", "darkgrey"), labels = c("Frail", "Not Frail")) +
  labs(title = "Percentages of frail patients by age group", x = "Age Groups", y = "Percentage
of frail patients", fill = "Frailty Status") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme_minimal()
dev.off()

```
````

#Put frail information in excel, pseudo_id + frailty status

````
```{r}
pseudo_id_frailty <- frail_or_not_merged[c("pseudo_id","frailty_status")]
file_path <- "H:/R/Code Kai/pseudo_id_frailty.xlsx"

write.xlsx(pseudo_id_frailty, file_path)
```
````
````

# Appendix H. Code specifications for the imputation methods

*Table 28. The code specifications of RF, pmm and norm imputation in R.*

| Goal | Package and function | Arguments |
|---|---|---|
| **Random Forest Imputation** | missForest: version 1.5<br>missForest() | |
| **Predictive mean matching MI** | mice: version 3.16.0<br>mice() | data = data<br>method = "pmm"<br>m=10<br>seed = 123<br>maxit = 20<br>visitSequence = VisitSeq<br>pred = predictor_matrix |
| **Predictive mean matching SI** | mice: version 3.16.0<br>mice() | data = data<br>method = "pmm"<br>m=1<br>seed = 123<br>maxit = 20<br>visitSequence = VisitSeq<br>pred = predictor_matrix |
| **Bayesian linear regression MI** | mice: version 3.16.0<br>mice() | data = data<br>method = "norm"<br>m=10<br>seed = 123<br>maxit = 20<br>visitSequence = VisitSeq<br>pred = predictor_matrix |
| **Bayesian linear regression SI** | mice: version 3.16.0<br>mice() | data = data<br>method = "norm"<br>m=1<br>seed = 123<br>maxit = 20<br>visitSequence = VisitSeq<br>pred = predictor_matrix |

*VisitSeq = c("pseudo_id", "gender", "age", "age_group" ,"frailty_status", "ALAT_numeric", "ASAT_numeric", "Hb_numeric", "MCV_numeric", "WBC_numeric", "ALB_numeric", "CRP_numeric", "PLT_numeric","HDL_numeric", "TP_numeric", "TCHOL_numeric", BILID_numeric","CRE_numeric", "UR_numeric", "CK_numeric", "Ca_numeric", "Cl_numeric", "GGT_numeric", "INR_numeric", "LDH_numeric", "Neut_numeric", "K_numeric", "ALP_numeric", "Na_numeric", "GLC_numeric").*

*Predictor_matrix = predictor_matrix[ ,"age_group"] <- 0. predictor_matrix["age",] <- 0. predictor_matrix["gender",] <- 0. predictor_matrix["age_group",] <- 0. predictor_matrix["pseudo_id",] <- 0 .predictor_matrix[,"frailty_status"] <- 0*