COMPARATIVE ANALYSIS OF SUB-OPTIMAL STATISTICAL METHODS ON LONGITUDINAL ANTIBODY TITRE DATA

MASTER OF SCIENCE (BIOSTATISTICS) THESIS

SUSANNE NTCHAULA BARNABA

UNIVERSITY OF MALAWI

MAY 2024

COMPARATIVE ANALYSIS OF SUB-OPTIMAL STATISTICAL METHODS ON LONGITUDINAL ANTIBODY TITRE DATA

MASTER OF SCIENCE (BIOSTATISTICS) THESIS

BY

SUSANNE NTCHAULA BARNABA

Submitted to the Mathematical Sciences Department, Faculty of Science, in partial fulfilment of the requirements for the degree of Master of Science (Biostatistics)

UNIVERSITY OF MALAWI

MAY 2024

DECLARATION

I, the undersigned, hereby declare that this thesis/dissertation is my own original work and has not been submitted to any other institution for a similar purpose. Where other people's work has been used, acknowledgements have been made.

SUSANNE NTCHAULA BARNABA
Full Legal Name

_____

Signature

_____

Date

CERTIFICATION OF APPROVAL

The undersigned certifies that this thesis represents the student's own work and effort, and has been submitted with my approval.

Signature:_____ Date:_____

Marc Henrion, Ph.D. (Lecturer)

Supervisor

DEDICATION

This thesis is dedicated to my daughter, Peniel Lucinda Kuwala Mphaka. I hope I have shown you the way. To my Dad and Mom, thank you for teaching me the importance of school and prayer at an early age in life. I believe I have made you proud!

ACKNOWLEDGEMENTS

I would like to thank God for His love, care, and guidance.

I would like to thank my supervisor Dr Marc Henrion, Ph.D. for his guidance, constructive ideas and for never giving up on me when I allowed work to take more of my time on this thesis. I really appreciate his encouragement and positive critiques on my work which have helped me structure the entire thesis with understanding. If it were not for him, I wouldn't have been submitting this today! It is all for you dear DR.

Furthermore, this would not be complete if I fail to thank my life support system: Augustine Barnaba, Lucy Barnaba, Peniel Lucinda Mphaka, Luka Mphaka, Patricia Barnaba and Tendai Barnaba. Thank you for your encouragement, prayers and support.

To my grandma Pat thank you for everything, hope you lived. This could have made you proud.

Special thanks to my friends, classmates and workmates: Ellen Gondwe, Chikondi Shaba, Halima Twabi, Angela Masano, Mphatso Luwemba and Ruth Vellemu for their support, encouragement, prayers and friendship during my studies. Lastly, but not least, a lot of gratitude goes to the Malawi Liverpool Wellcome Trust statistical support unit for their generous support in providing this dataset the PCVPA data. This cannot go without mentioning Dr. Todd Swarthout for structuring the dataset for this thesis, I appreciate his assistance.

ABSTRACT

Longitudinal studies provide valuable insights into changes and factors influencing responses over time, but inappropriate methods can lead to erroneous results. This study evaluates longitudinal data analysis methods for estimating antibody titres, focusing on correcting inappropriate commonly used methods and providing recommendations for optimal statistical inference. The study contributes to the knowledge base in Malawi and addresses the gap in appropriate longitudinal modelling techniques. Using inappropriate statistical methods in longitudinal data analysis can yield misleading results, affecting the validity and reliability of research findings. Addressing this issue is crucial for ensuring accurate estimation of antibody titres. In this, study a comparative approach was employed, analyzing both real-world and simulated data to assess the performance of different modelling techniques. A longitudinal censored mixed model used in the simulated data to account for lower limits of detection and contrasted this to imputations of censored values to 0, DL/2, DL, and complete case analysis. Censored regression models and imputations were used for non-linear, non-longitudinal PCVPA data. Raw data used arithmetic means, while log-transformed data used geometric means. The longitudinal aspect of the data is accounted for through random effects. By simulating ELISA data with known vaccination and age effects evaluated the effectiveness of statistical models in estimating antibody concentrations. The analysis of both real-world and simulated data reveals significant insights into the performance of different statistical methods. Findings indicate that certain models perform poorly in capturing the effects of age, exposure, and gender. However, the censored model stands out by providing estimates closer to the true values and narrower confidence intervals, particularly in intercept estimation. The comparison between real-world and simulated data underscores the importance of selecting appropriate statistical methods for longitudinal data analysis. The study's results emphasize the significance of the censored model in improving estimation accuracy and reducing bias. Thereby, enhancing understanding of longitudinal data analysis for antibody titres, contributing to advancing statistical inference in research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ELISA	Enzyme-linked Immuno-Absorbent Assay
PCVPA	Pneumococcal Vaccine for Vulnerable Populations in Africa
LoD	Limit of Detection
LoQ	Limit of Quantification
LoB	Limit of Blank
VT	Vaccine Serotype
PCV	Pneumococcal Vaccine
OD	Optical Density
REML	Restricted Maximum Likelihood
ML	Maximum Likelihood
DL	Detection Limit
GLM	Generalized Linear Models
GEE	Generating Estimates Equations
ABC	Adjustment for Bi-censoring

CHAPTER 1

INTRODUCTION

This chapter presents the brief background of the study, the knowledge gaps that were identified, the objectives of the study and the importance of the study.

1.1 Background

1.1.1 Background to Antibody Titres

Antibodies are specific chemicals that bind to the antigens used for their production (Crowther, 2000) . They are produced in response to antigenic stimuli and are mainly protein in nature. They belong to a group of serum known as globulins, they are also known as immuno-globulins because of their immune response functions. Antibodies are subdivided into five subtypes known as IgA, IgD, IgE, IgG and IgM based on molecular size, structure and function (Crowther, 2000). According to Crowther (2001) antibody titres measure how much antibody for specific pathogens an organism has produced. The enzyme-linked immuno-absorbent assay (ELISA) system is widely used for measuring antibody titres and antigens.

There are two main approaches to estimating antibody titers from ELISA data; that is Standard curve fitting and endpoint titration. In standard curve fitting method it involves fitting a standard curve by plotting the optical density (OD) or other measurement values against known concentrations of the antibody standards. The unknown antibody concentrations can then be determined by interpolating their OD values onto the standard curve (Yang et al., 2016). Whilst in endpoint titration the sample is serially diluted, and the endpoint titer is defined as the reciprocal of the highest dilution that gives a reading above a pre-determined cut-off value or threshold (Yang et al., 2016). The cut off is chosen for each dilution step to determine the endpoint titer.

Recent studies have highlighted some limitations and potential improvements to these antibody titer estimation methods, these limitations include introducing errors which

leads to biases in model fitting and potentially misrepresenting lower concentration limits. Therefore, it is recommended that these lower limits must be validated to ensure the accuracy of antibody titer measurements.

There are different end point titers, which are limit of detection, limit of quantitation and limit of blank all these are used to describe the smallest concentration of an analyte that can be reliably measured by an analytical procedure (Armbruster & Pry, 2008). Limit of blank (LoB) according to Epi 17 protocol guideline it is defined as the highest apparent analyte concentration expected to be found when replicates of blank sample containing no analyte are tested (Armbruster & Pry, 2008) and this is given by LoB = mean blank + 1.645 (standard deviation of blanks). The (lower) limit of detection (LoD) is the lowest analyte concentration likely to be reliably distinguished from LoB and at which detection is feasible. It is determined by using both test replicates of a sample known to contain low concentration and also measured limit of blank which is deduced as LoD = LoB + 1.645 (standard deviation of low concentration sample). The (lower) limit of quantification (LoQ) is defined as the lowest concentration of an analyte that can not only be detected but also measured up to predefined targets of accuracy and precision (Armbruster & Pry, 2008). By definition LoQ ≥ LoD > LoB.

Below is a graph illustrating the difference between limits of blank, detection, and quantification, according to guideline EP17, Protocols for Determination of Limits of Detection and Limits of Quantification which was published by Clinical Laboratory and Standards Institute (Tholen, 2004).

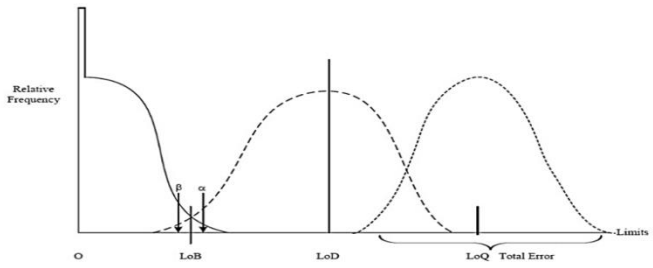


Figure 1: Illustrating the relationship between LoB, LoD and LoQ, the solid line represents the results distribution of LoB, the dashed line represents the results distribution of LoD and the dotted line represents the results distribution of LoQ.

As it has been mentioned, recent studies have made significant advancements in the estimation of antibody titers, particularly focusing on improving accuracy and addressing limitations that have been mentioned above in the existing methods. The key points that were highlighted are; addressing measurements challenges, studies have addressed challenges related to measurements falling below the limit of detection (LOD) in antibody assays. New approaches, like the adjustment for Bi-censoring (ABC) method, have been developed to handle measurements below the LOD effectively, ensuring more robust estimations of antibody titers across various assays, including the hemagglutination inhibition assay (HAI) (Ge et al., 2022). But little has been done on data that is below detection limit. Another is adjusting for censored data and simulation studies.

In adjusting for censored data came up in order to mitigate biases introduced by values below the LOD, recent studies have proposed novel methods to adjust coefficient estimates, accounting for the censored nature of these measurements. By applying these adjustments, researchers can obtain more accurate estimates of antibody titer increases, particularly in the context of vaccine studies (Ge et al., 2022). This is one of the methods that has been taken into account in this study.

Simulation studies, this is a key cornerstone in this thesis. Although several simulation studies have been conducted to mimic real-world scenarios in antibody assays, generating data that closely resembles actual assay results. And also even these simulations have been instrumental in developing and validating new methods for estimating antibody titers, ensuring reliable and unbiased measurements(Ge et al., 2022). Little has been done on data that is censored to the left and have detection limit at the same time longitudinal nature.

1.2 Problem statement

In this thesis longitudinal dataset of IgG antibody data which contained some missing values and subject to a pre-specified LoD was used. Often, antibody titre data are analyzed using imputation techniques and other methods like calculation of the arithmetic mean and fitting simple linear regression without considering the nature of concentration data, censored data, or longitudinal data.

Statistically, data below the LoD / LoQ (identical in the data for the present study) are left censored. There are many methods to handle such censored observations, some based on statistical theory, some on heuristic. One commonly used method involves discarding censored observations; this is used when you know that the data does not contain the information you need. Simple imputation techniques form another approach. Missing data are substituted with fixed values (typically either 0, the LoD/LoQ or a value half-way between 0 and the LoD/LoQ). While simple imputation maintains the sample size, and is easy to use, it artificially reduces variability, thus underestimating the variance and standard deviation of the data.

Another approach is the use of non-parametric methods, which are based on fewer assumptions. For example, non-parametric methods like the Wilcoxon rank-sum test may also work well in antibody titre data. Apart from a marginal reduction in power, these non-parametric methods has some limitations. The main limitation is that non-parametric methods do not extend easily to more sophisticated analyses where you adjust for covariates or confounders and they do not help at all when your goal is estimation or prediction (as opposed to identifying associations).

Furthermore, Censored regression techniques, borrowing ideas from time-to-event analysis have been available for some time for example, Tobit regression which treat data below the lower limit of detection (LOD) as censored data and it accounts for both left and right censoring in the data. and the techniques from survival analysis are mostly used for estimating parameters.

However, in the context of antibody titre or ELISA, it is also important to work on the log scale, given the nature of data generation. For instance, the geometric mean is meaningful compared to the arithmetic mean. This natural log scale arises because it is more important to know when the values are halving or doubling than a change in absolute value.

But what is less clear is how to incorporate these ideas into longitudinal analyses where data are correlated and statistical models need to account for that. So, in summary, there are three main challenges for longitudinal data analysis of antibody titre data which we aim to address in a single analysis framework:

- Left-censored data due to a LoD/LoQ.
- Longitudinal / correlated data.
- Skew data where the focus of analysis is on relative not absolute change.

To this end, this thesis will investigate the impact of ignoring these challenges and using simple, sub-optimal but easier to use techniques.

1.3 Study objectives

The main objective is to assess the impact on statistical inference of sub-optimal analysis methods.

Specifically, the study wants to;

- To quantify the bias and the underestimation of variance due to simple imputation methods.
- To quantify the impact of ignoring the logarithmic scale on data interpretation.
- To evaluate whether the above issues are exacerbated in a longitudinal data setting.

1.4 Significance of the study

There are several impacts if statistical inference is not properly done. For instance, as already discussed above, analysing longitudinal antibody titre data using imputation techniques and using arithmetic mean instead of geometric mean models poses challenges as bias is introduced and variance is underestimated. And also, it affects interpretation, for it is more difficult to interpret absolute value than relative change, and it is also harder to compare two groups. The statistical approaches explored and used in this study reduce the bias mentioned and provide a principled approach to analysing antibody titre data and correct ways to deal with such data.

This study will primarily show the correct ways of dealing with this data and clearly show how wrong results can be if sub-optimal methods are used. The research report will add to the longitudinal analysis knowledge base that is currently available in Malawi. Besides, the study will also correct the inappropriate methods that are mostly used, assess their implications, and make recommendations on the best type of longitudinal model to use for estimating antibody titres.

1.5 Thesis structure

The thesis is structured as follows: Chapter 2 gives a literature review of different model diagnostic statistics. Chapter 3 describes the methodology used for this study. Chapter 4 presents the results. Chapter 5 provides a discussion, a conclusion, and recommendations.

CHAPTER 2

LITERATURE REVIEW ON METHODS

This chapter reviews the literature on statistical analysis methods that are used in 1) longitudinal data analysis, and 2) analysis of antibody titre data. As part of this, the chapter also touches on the concept of geometric and arithmetic means and on censored regression techniques.

2.1. Overview of longitudinal data analysis.

Longitudinal studies involve the repeated measurement of individuals over time to study changes in responses and the factors influencing these changes (Fitzmaurice et al., 2011). Weiss (2005) expands this definition to include the collection of outcomes, treatments, or exposures at multiple follow-up times, emphasizing the importance of temporal ordering in longitudinal data analysis. This type of data is described as multivariate and hierarchical, with observations nested within subjects (Weiss, 2005).
The structure of longitudinal data, characterized by multiple observations within subjects ordered across time, necessitates consideration of its unique properties for analysis (Fitzmaurice et al., 2011). Hedeker & Gibbons (2006) highlight the advantages of longitudinal studies, noting their increased power compared to cross-sectional studies due to the independent information provided by repeated measures. Additionally, each subject acts as their own control, reducing intra-subject variability relative to inter-subject variability.

Longitudinal studies allow for the separation of temporal effects within individuals from cohort effects at baseline (Hedeker & Gibbons, 2006). This distinction is crucial in understanding changes over time and considering different time scales such as cohort and period effects. In contrast, cross-sectional studies may confound aging effects with cohort differences (Brown & Prescott, 2015).

Despite the benefits of longitudinal studies, challenges exist, such as the non-independence of data observations within individuals, requiring sophisticated statistical

methods to address this dependency (Hedeker & Gibbons, 2006). Parameter estimation for certain models can be computationally intensive due to iterative processes and the lack of closed-form solutions.

Attrition, leading to missing data as participants drop out over time, poses a significant challenge in longitudinal studies (Hedeker & Gibbons, 2006). Reasons for attrition, like perceived lack of benefit or adverse effects, can introduce bias and impact the sample's representativeness, potentially affecting the generalizability of findings.

Proper statistical analysis of longitudinal data must account for the intra-subject correlation of response measurements to ensure valid inference (Fitzmaurice et al., 2011). Neglecting this correlation can lead to invalid results, affecting confidence intervals and the outcomes of statistical tests.

In summary, longitudinal data analysis methods account for multiple observations within subjects ordered across time, addressing challenges such as non-independence, missing data, and the need to properly model intra-subject correlation for accurate statistical inference.

2.2 Geometric mean and Arithmetic mean.

The sample mean which is the arithmetic mean is the most frequently used statistic for summarizing research data in applications in which the response of interest is measured on a continuous scale (Olivier et al., 2008). The arithmetic mean (AM) captures the average value for a series of numbers, mathematically it can be represented as;

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \qquad (1)$$

Where n is the number of observations and X_i is the $i-^{th}$ observation of the random variable X, $i = 1, \dots, n$.

The geometric mean (GM) is obtained by calculating the n^{th} root root of the product of a collection of n numbers and is a measure of central tendency (Olivier et al., 2008).

To come up with geometric mean formula let's consider $X_1, X_2, \dots, X_n$ then geometric mean is defined as;

$$GM = \sqrt[n]{x_1 x_2 \dots x_n} \qquad (2)$$

This can also be written as;

$$LogGM = \frac{1}{n}\log(x_1 x_2 \dots x_n) \qquad (3)$$

$$= \frac{1}{n}(\log x_1 + logx_2 + \cdots + logx_n)$$

$$= \frac{\sum logx_i}{n}$$

$$GM = \exp\left(\frac{\sum logx_i}{n}\right) \qquad (4)$$

The main distinction between geometric and arithmetic means is that, in order to determine the geometric mean, all the n numbers in the given data set must be multiplied, and the observed result must be taken as the n^{th} root. The arithmetic mean is determined by adding up the n numbers in the dataset and dividing by n.

2.3. Bootstrap method

Bootstrapping is a resampling procedure that uses data from a sample to generate a sampling distribution by repeatedly taking random samples from the known sample with replacement (Cameron & Pravin, 2005) it is also defined as simulation methods for frequentist. It is one of the widely applicable computer intensive statistical tools that can be used to yield estimates of parameters that are difficult to estimate otherwise. The bootstrap method is commonly used in case where there is complicated statistic and no analytical formula is available (Wehrens et al., 2000).

The fundamental idea in bootstrap is repeatedly draw samples with replacement from the observed data to simulate the variability inherent in the data collection process. For instance, given an observed dataset $X = (x_1, x_2, \dots, x_n)$, the bootstrap procedure involves generating multiple bootstrap samples denoted as $X^* = (x_1^*, x_2^*, \dots, x_n^*)$, by randomly selecting observations with replacement (Davison & Hinkley, 1997). The key concept is using these resampled datasets to approximate the sampling distribution of a statistic of interest.

Given a bootstrap sample X^*, calculating the statistic of interest, denoted as θ^*;

$$\theta^* = g(x_1^*, x_2^*, \dots, x_n^*) \qquad (5)$$

Thereafter, this process is repeated for several times (B iterations) to create an empirical distribution of θ^*. The result is the collection of bootstrap statistic $\theta_1^*, \theta_2^*, \dots, \theta_B^*$ (Davison & Hinkley, 1997).

The aim of this empirical distribution is to use it to make statistical inferences. For example, one can estimate the confidence interval for the population parameter by determining the range between the $\alpha/2$ -th and $1-\alpha/2$-th percentiles of the bootstrap statistic:

$(\theta^*_{\alpha/2}, \theta^*_{1-\alpha/2})$The bootstrap method can be applied in several ways; bootstrap in hypothesis testing, bias reduction, confidence intervals and estimation of standard errors (Cameron & Pravin, 2005). In this thesis bootstrap confidence intervals were used in real world PCVPA data to come up with confidence bands. Bootstrap confidence intervals offer several advantages, they are versatile, applicable to various statistical problems and particularly useful when underlying distribution is unknown or complex. Additionally, bootstrap confidence intervals can be constructed for almost any statistic making them a valuable tool for statistical inference.

In general, bootstrap methods provide a powerful and widely applicable tool in statistics, to assess uncertainty and derive confidence intervals in the absence of distribution information. The methods simplicity and ability to provide reliable estimates for a variety of statistical problems makes it widely used (Davison & Hinkley, 1997).

2.4 Censoring

Censoring occurs when the event of interest is not observed for some subjects because either 1) it occurred before the study started (left censoring), 2) it did not occur before the study is terminated (right censoring) or 3) it occurred between study visits (interval censoring). When censoring occurs, the researcher has only partial information about the subjects for which censoring occurred (Turkson et al., 2021). Censoring is common in survival analysis for it represents a form of missing data (Kleinbaum & Klein, 2005). In survival analysis, almost all censoring is right censoring and there are three broad reasons why right censoring might occur; a person is lost to follow up during the study period, a person withdraws from the study or a person does not experience the event before the study ends (Kleinbaum & Klein, 2005). Longitudinal data are also sometimes censored, for the same reasons as in survival analysis, but also due to study design (e.g. when study visits are scheduled which can lead to interval censoring) or because an event of interest has already occurred before the study started.

As already mentioned above, there are three types of censoring, right censoring, left censoring and interval censoring (Turkson et al., 2021). Right censoring is widely known for it is common in survival analysis as well as longitudinal analysis, it occurs when individuals have not experienced the event of interest by the end of the study or the last available follow-up time. For example, if we assume that there is a time *T* and a censoring time *b*, the*T's* are independently and identically distributed with probability function $f(t)$ and survival function $S(t)$. The exact life time T of a subject will be known, if and only if *T* is less than or equal to *b*; if *T* is greater than *b*, the subject is a survivor and his event time is censored at *b*. The data from this experiment can be represented by pairs of random variables(K, δ), where δindicates whether the lifetime corresponds to an event$(\delta = 1)$ or is censored$(\delta = 0)$, and K is equal to *T* if the lifetime is observed and *b* if it is censored. For a right-censoring $K = \min(T_i, b)$, where K is some time variable and *a* and *b* some points in time (Klein & Moeschberger, 2003) (Klein & Moeschberger, 1997).

Interval censoring occurs when the event of interest is known to have occurred within a specific time interval but the exact timing within that interval is unknown. In other words, it can be by study design ; for example, study visits every 6 months, but that participant experiences an event of interest (say heart attack) between visits, or no exact date known just that it occurred between two schedule visits i.e. (a<T<b) (Turkson et al., 2021) In this type of censoring the observed data consist of intervals $I_1 I_2, \dots, I_m$where for each $k = 1, 2, \dots, m$ where k is the number of time intervals and m is the total number of time intervals. In this case an uncensored observation of an observed death corresponds to an observed interval consisting of a single point (Turkson et al., 2021). This type of censoring commonly occurs when periodic assessments of the outcome event of interest is done at discrete time points rather than continuously.

This thesis focused on this left censoring. Typically, this occurs when the event of interest has already occurred before enrolment. Left censoring is very rare for studies observing events but it is commonly encountered in laboratory data when dealing with continuous data subject to detection limits.

To define left-censoring, for example, a time *X* associated with a specific subject in a study is considered to be left censored if it is less than censoring time *a*. For left censoring to occur the event of interest must occur for the subject before that person is observed in the study at time *a (T<a)*. For such subjects since they have already experienced the event sometime before time *a*, but the exact time is not known. Therefore, the exact time will be known if and only if *X* is greater than or equal to *a*. The data from a left censored sampling scheme can be represented by pairs of random variables(T, δ), where T is equal to X if the lifetime is observed and δindicates whether the lifetime corresponds to an observed event $(\delta = 1)$or is censored$(\delta = 0)$, for left censoring $T = \max(X_i, b)$ (Turkson et al., 2021). In laboratory studies, the measurement of the analyte is equivalent to the event time in event time data. Here, left censoring occurs if there is a lower limit below which the laboratory assay cannot yield a valid measurement anymore. Value below this lower limit of detection are left censored.

Klein et al (2003) concluded that left censoring is a special case of right censoring with the time axis reversed, and it is for this reason there have been few special techniques developed solely for left censored data.

In addition to afore mentioned reasons of censoring, censoring can also be due to events of interest. When this happens, it becomes informative censoring and this introduces bias. In most longitudinal analyses, it is assumed that censoring is non-informative or random. This occurs when the probability of censoring is unrelated to the event of interest or the censoring is considered independent of the underlying survival time or outcome.

2.4.1 Methods for censored data

There are several methods of handling censored data, depending on the nature of censoring. As previously stated, censoring can be of 3 ways; either censoring to the right, censoring to the left and interval censoring. Some of the methods commonly used in handling censored data include likelihood-based approaches, imputation approaches, dichotomizing the data and complete data analysis (Turkson et al., 2021).

Likelihood-based approaches use estimation methods, which involve constructing a likelihood function that models the probability distribution of both observed and censored values., and many of these methods maximizes the likelihood under certain model assumptions, including the censoring mechanism. This type of approaches includes Kaplan- Meier, log-rank test and the Cox regression (Turkson et al., 2021).

The Kaplan-Meier method also called the product limit estimator is the most popular method when dealing with survival analysis for it requires weak assumptions i.e. assumes no distribution but it utilizes all the information i.e. right censored data and fully observed data (Hosmer et al., 2008). It is a non-parametric method used to estimate survival probability S(t) from observed survival times (Hosmer et al., 2008). Let $0 \leq t_1 < \cdots < t_n$ be the observed death times, let n_i be the number of individuals at risk. And let d_i be the number of observed deaths at $t_i, i = 1, \ldots, n$ then the Kaplain-Meier estimator is given by;

$$\hat{S}(t) = \prod_{i:T_i \leq t} \frac{r_i - d_i}{r_i} = \prod_{i:T_i \leq t} (1 - \frac{d_i}{r_i}) \qquad (6)$$

Where r_i is the number of individuals at risk right before the ith death time.

The log-rank test also called the Mantel-Haenszel test (when comparing only 2 curves), is a statistical significance test that is used compare two or more groups. This test is also obtained by constructing a 2x2 table at each distinct death time, and comparing the death rates between the two groups conditional on the number at risks in the groups (Collet, 2004). Considering the null hypothesis there is no difference between survival population curves. i.e. the probability of an event occurring at any time point is the same for each population. The test statistic is calculated as follows

$$\chi^2(logrank) = \frac{(O_1 - E_1)^2}{E_1} + \frac{(O_2 - E_2)^2}{E_2} \qquad (7)$$

Where O_1 and O_2 are the total numbers of observed events in groups 1 and 2, respectively and E_1 and E_2 are total numbers of expected events.

Then the Cox regression or the Cox Proportional Hazard model is a semi-parametric model. It is semi-parametric because it makes no parametric assumptions regarding the baseline hazard. The Cox proportional hazard model makes parametric assumptions concerning the effect of the predictors on the hazard function but makes no assumption

regarding the nature of the hazard function $\lambda(t)$ (Harrell, 2001). Since the only assumption made is on proportionality of the baseline hazard, therefore, it means that the hazard ratio is constant over time (Collett, 2015). PH is the common approach used in research to model the effects of covariates on survival. It can be defined as; let $y_1, \dots, y_j$ be the values of *j* covariates $Y_1, \dots, Y_j$, then the hazard function is given as the following model (Cox regression model);

$$h(t) = h_0(t) \exp\left(\sum_{i=1}^{j} \sigma_i y_i\right) \qquad (8)$$

Where $\sigma_i = \sigma_1, \sigma_2, \dots, \sigma_n$ is a 1 x j of regression coefficients and $h_0(t)$ is the baseline hazard function at time *t*.

Furthermore, when analyzing left-censored data using Kaplan-Meier survival analysis and Cox proportional hazards regression, it is essential to handle the inherent uncertainty about the exact event times. For Kaplan-Meier, left-censored events can be treated as tied events occurring at time zero, assuming they happened at the earliest possible time. This involves considering censored events at time zero as if they occurred simultaneously. In Cox regression, left-censored data can be included by treating it as regular censored data, assuming that censoring at time zero is non-informative. The Cox model assumes that the probability of censoring at a given time is unrelated to the probability of the event occurring.

A common type of method of handling censored data, particularly left-censored concentration data from laboratory experiments, is given by various imputation techniques. These methods have several disadvantages, specifically the introduction of bias and under-estimation of variance. These methods depend on model assumptions that are difficult to check without information (Turkson et al., 2021). Many researchers use imputation techniques because of lack of statistical software packages for analysis because some censored data require sophisticated methods.

This thesis used data that were left-censored due to the detection limit. The common methods for left-censored data due to a detection limit are imputation to zero, imputation to the detection limit and imputation to half the detection limit. Imputation to zero, where events below the detection limit are treated as if they occurred at the earliest possible time, often time zero (Turkson et al., 2021). Although this method is

straightforward, it assumes that all events below the detection limit occur simultaneously, potentially introducing bias and underestimating the true event times.
Imputation to the detection limit this involves assigning all left-censored observations the value of the detection limit itself. This method acknowledges that events occurred but provides limited information about when they occurred. However, imputing events to the detection limit might introduce an upward bias, as it assumes all left-censored events happened precisely at the threshold, ignoring potential variability in their true occurrence times (Turkson et al., 2021).

Imputation to half the detection limit. This method strikes a balance between imputing to zero and imputing to detection limit, thereby acknowledging the uncertainty in the event timing and assuming events are equally likely to occur at any point within the detection range. This method mitigates biases introduced by the other imputation methods, thereby providing a more conservative estimate of event times (Turkson et al., 2021).

In addition to this, sometimes researchers resort to dichotomizing the data, Dichotomizing left-censored data involves transforming continuous survival times into binary outcomes, typically distinguishing between "event" and "non-event" based on a specified threshold (Leung et al., 1997). While this method is utilized to simplify analyses and accommodate left-censored information, it carries inherent limitations. Dichotomization results in a loss of precision and statistical power, as it disregards the continuous nature of survival times. The choice of the threshold becomes crucial, introducing subjectivity and potentially influencing conclusion (Leung et al., 1997). Additionally, dichotomizing may mask important temporal patterns in the data, as it oversimplifies the nuanced information embedded in the left-censored survival times (Leung et al., 1997). When dealing with dichotomization of left-censored data it has to be done with caution, considering its potential impact on the reliability and interpretability of survival analyses.
Finally, another method sometimes used for censored data is complete case analysis. Censored observations are completely ignored and only the uncensored complete observations are included in the analysis. This type of analysis is commonly adopted because of its simplicity but it has several disadvantages. These include estimation bias for inference based on analysing

uncensored data only may be biased. Another reason is loss of efficiency since there is loss in sample size (Leung et al., 1997).

2.4.2. Censored regression models

The basic model that was used in this thesis is the censored regression model which was developed from the Tobit model which was named after Tobin (1958) who applied it to individual expenditure on consumer durable goods (Cameron & Pravin, 2005). These are statistical models designed to handle data where the dependent variable is subject to censoring, meaning that certain observations are only partially observed or limited by some threshold. These models are particularly relevant in scenarios where the outcome variable is only observable within a certain range or under certain conditions (Amemiya, 1984). As it was previously stated censoring can be either right-censored (values above a certain threshold are unobservable) or left-censored (values below a certain threshold are unobservable). Censored regression models address this challenge and provide estimates while accounting for the censored nature of the data.

The censored regression model can be defined by using a latent variable y^* which is only partially observed and assumed to be normally distributed. Let $x_{1,i}, x_{2,i}, x_{3,i}, \dots, x_{p,i}$ be p observed variables for the ith study participant, i=1,..,n. The standard censored regression model (Tobit model) can be written as;

$$y_i^* = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_p x_{p,i} + \varepsilon_i \qquad (9)$$

Then,

$$y_i^* = X_i'\beta + \varepsilon_i \qquad (10)$$

with ε_i are assumed to be independent and identically distributed from $N(0, \sigma^2)$. It is also assumed that y_i and x_i are observed for $= 1,2, \dots, n$, but y_i^* are observed if

$y_i = y_i^*$ if $y_i^* > DL$,

$y_i = 0$ if $y_i^* \leq DL$

To estimate the parameters the likelihood for the above equation needs to be computed and optimised. Let's define X to be an *n X p* matrix whose ith row is x_i', we assume that $lim_{n\to\infty} n^{-1}X'X$ is positive definite. Note that $y_i^* > DL$ and $y_i^* \leq DL$ may be changed to $y_i^* > y_0$ and $y_i^* \leq y_0$ without essentially changing the model, whether y_0 is known or unknown, since y_0 can be absorbed into the constant term of the regression. Then the likelihood function of the standard censored regression model to estimate the parameters is given by;

$$L = \sum_{0} [\Phi(^{x_i'\beta}/_{\sigma})] \sum_{1} \phi[(^{y_i - x_i'\beta}/_{\sigma})] \qquad (11)$$

Censored regression models, offer a powerful framework for handling censored data and provide estimates that account for the limitations imposed by censoring (Amemiya, 1984).

2.5. Longitudinal data analysis models

This subsection explores different models that are commonly used in analysing longitudinal data. There is a class of regression approaches that is commonly considered; mixed effects models which are the major focus of this thesis.

2.5.1 Mixed effects models

Mixed effects models are popular for modelling longitudinal data and a basic characteristic of these models is the inclusion of random effects into the regression models to account for the influence of a grouping variable, e.g. subjects with repeated observations. Such grouping variables are called random factors and they are used to capture differences in the response variables and differences in the effects of covariates (referred to as fixed factors) between different levels of the grouping variable(s) on the response (Hedeker & Gibbons, 2006). Because of the inclusion of both random and fixed factors (and hence the estimation of the associated random and fixed effects / model coefficients), these models are called mixed effects models. Such mixed models also allow to estimate the degree of variation at the level of the grouping variable that exists in the data. For what follows in this review and discussion of mixed effects models, we assume a single grouping variable and that that variable is the subject ID variable.

A key feature of mixed models, in the context of longitudinal data where subjects are followed up over time, is that subjects are not assumed to be measured on same number of time points and this means that all data can be easily included in the analysis. The inclusion of all data has the advantage that it increases statistical power.

2.5.1.1 Linear mixed effects models.

Linear mixed effects (LMMs) models are an extension of the general linear model to include random factors. LMMs make specific assumptions about the variation in observations attributable to variation within subjects and to variation between subjects. These models permit regression analysis with correlated data and also they specify variance components that represents both within-subject and between-subject variation in outcomes and trajectories. Linear mixed model parameters can be estimated using either Maximum Likelihood (ML) or Restricted Maximum Likelihood (REML) methods. Maximum Likelihood estimation aims at maximizing the likelihood function, which measures how well the model explains the observed data. ML estimates the variance components of both fixed and random effects, providing parameter estimates that maximize the probability of observing the given data under the assumed model. However, ML tends to yield biased estimates, especially for random effects, as it can be sensitive to sample size (Fitzmaurice et al., 2011).

While Restricted Maximum Likelihood estimation addresses the bias issue by maximizing the likelihood function, but under the condition that the estimates are consistent with the fixed effects. REML removes the fixed effects and only focuses on the random effects' variance components. This method is particularly useful for estimating the variability associated with random effects without being influenced by fixed effects. REML estimates are often considered more reliable for understanding the underlying variance structure in the data. In summary, ML estimates both fixed and random effects, whereas REML primarily focuses on the variance components of random effects, providing more robust estimates of the underlying variability in the data (Fitzmaurice et al., 2011).

Let's assume a sample of *N* subjects are measured repeatedly overtime, let Y_{ij} denote the response variable for the i^{th} subject on the j^{th} measurement (Fitzmaurice, Laird, & Ware, 2011). Let $\beta_{i0} + \beta_{i1}X_{ij}$ denotes the observation line path for subject *i* where X_{ij} denotes the time of measurement *j* on subject *i.* The within-subject variation is given by $Y_{ij} - (\beta_{i0} + \beta_{i1}X_{ij})$ and the between-subject variation among intercepts is var(β_{i0})and among (Cameron & Pravin, 2005) slopes is var(β_{i1}). Let us also assume that the within-subject intercepts and slopes are normally distributed.

Within subjects:

$$Y_{ij} = \beta_{i0} + \beta_{i1}X_{ij} + \varepsilon_{ij} \qquad (12)$$

where $\varepsilon_{ij} \sim N(0, \sigma^2)$

And between subjects: $\begin{pmatrix}\beta_{i0}\\ \beta_{i1}\end{pmatrix} \sim N[\begin{pmatrix}\beta_{i0}\\ \beta_{i1}\end{pmatrix}', \begin{pmatrix}D_{00} & D_{01}\\ D_{10} & D_{11}\end{pmatrix}]$

Re-writing this can be $b_{i0} = (\beta_{i0} - \beta_0)$ and $b_{i1} = (\beta_{i1} - \beta_1)$

Therefore, this model can be written as;

$$Y_{ij} = \beta_0 + \beta_1 X_{ij} + b_{i0} + b_{i1}X_{ij} + \varepsilon_{ij} \qquad (13)$$

A more general form, with more than one independent variable, can be written as;

$$Y_{ij} = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + b_{i0} + b_{i1}X_{i1} + \cdots + b_{ip}X_{ip} + \varepsilon_{ij} \qquad (14)$$

where $\beta_0, \beta_1, \ldots, \beta_k$ represent fixed effects and $b_{i0}, b_{i1}, \ldots b_{ip}$ represent random effects.

Therefore, $Y_{ij} = X'_{ij}\beta + Z'_{ij}b + \varepsilon_{ij} \qquad (15)$

Where $X'_{ij} = X_{ij1}, X_{ij2}, X_{ij3}, \ldots X_{ijk}$ and $Z'_{ij} = X_{ij1}, X_{ij2}, X_{ij3}, \ldots X_{ijp}$ and it is assumed that the covariates Z_{ij} are a subset of the variables in X_{ij} thus $p<k$.

2.5.1.2 Longitudinal censored models

Literature shows that different types of longitudinal models have been used to deal with censored data, but the more common model was the use of linear mixed models by imputing the data. The data that was used in this thesis was longitudinal and censored to the left, hence the sophisticated linear model employed.

Recall from the previous section on censored or tobit model, this was originated from linear regression analysis. Let y^* be the latent variable that is not censored and assume linear regression (Twisk & Rijmen, 2009).

$$y_i^* = x_i'\beta + \varepsilon_i \qquad (16)$$

Where $\varepsilon_i \sim N(0, \sigma^2)$

Addition to that lets assume that y^* can be observed at a range of (l, p) only that the values of y^* are smaller than *l* or larger than *p*. Hence the observed dependent variable *y* is obtained from y^*.

$$y_i = l \; for \; y_i^* \leq l \qquad (17)$$

$$y_i = y_i^* \; for \; l < y_i^* < p \qquad (18)$$

$$y_i = p \; for \; y_i^* \geq p \qquad (19)$$

Furthermore, for longitudinal censored or tobit model, since $E(y)$ is not eaqual to $E(y^*)$ because of censoring. For distribution of y is not the same as the distribution of y^* (Twisk & Rijmen, 2009).

Therefore, for longitudinal censored model can be defined in a similar way by let y^* be a linear mixed model thus;

$$Y_{ij} = \beta_0 + \beta_1 X_{i1} + b_{i0} + \varepsilon_{ij} \qquad (20)$$

Which can also be written as;

$$Y_{ij} = X'_{ij}\beta + Z'_{ij}b + \varepsilon_{ij} \qquad (21)$$

So, to account for censoring in the data then a longitudinal censored mixed model can be written as;

$$Y^*_{ij}|b_i = X'_{ij}\beta + Z'_{ij}b + \varepsilon_{ij} \qquad (22)$$

Where $\varepsilon_{ij} \sim N(0, \sigma^2)$ and $b_i \sim N(0, D)$

Where *i* denotes subject *i* and X_{ij} denotes the time of measurement *j* on subject *i*.

It is challenging to estimate the longitudinal model's parameters because the likelihood involves integrals over the random effects b_i that are not analytically solvable (Twisk & Rijmen, 2009). When the dimensionality of is too low the integral can be approximated using the Gaussian quadrature. The likelihood of the mixed censored model to estimate parameters can be defined as;

$$Y^*_{ij}|b_i = X'_{ij}\beta + Z'_{ij}b + \varepsilon_{ij} \qquad (23)$$

Where $\varepsilon_{ij} \sim N(0, \sigma^2)$ and $b_i \sim N(0, D)$

Where *i* refers to case *i* and *j* to the *jth* measurement conditional on the case specific parameters b_i, a linear model is assumed with

$$E\left(Y^*_{ij}\middle|b_i\right) = X'_{ij}\beta + Z'_{ij}b + \varepsilon_{ij} \qquad (24)$$

Where *y* is obtained from y^* as;

$$y_{ij} = l \; for \; y^*_{ij} \leq l \qquad (25)$$

$$y_{ij} = y^*_{ij} \; for \; l < y^*_{ij} < p \qquad (26)$$

$$y_{ij} = p \; for \; y^*_{ij} \geq p \qquad (27)$$

Therefore, the density function of *y* is

$$f\left(y_{ij} = l\right) = F\left(y^*_{ij} = l\right) \qquad (28)$$

$$f\left(y_{ij}\right) = f\left(y^*_{ij}\right) for \; l < y^*_{ij} < p \qquad (29)$$

$$f\left(y_{ij} = p\right) = 1 - F\left(y^*_{ij} = p\right) \qquad (30)$$

Hence the contribution to the likelihood of case *i* is obtained as a summation of the *j* measurement for case *i* and integrating this summation over the case specific parameters (Twisk & Rijmen, 2009). Thus

$$L_i = \int_{b_i} \sum_j f(y_{ij}) N(b_i; 0, D) db_i \qquad (31)$$

Therefore, the likelihood of the mixed censored model can be written as

$$L = \sum_i L_i \qquad (32)$$

2.5.2. Generating Estimates Equations (GEE)

GEEs are an estimation approach for generalized linear models (GLM) that accounts for correlated data and clustered data. It provides a flexible framework for modeling the relationship between variables while accounting for correlation within clusters. Most statistical methods often assume independence among observations, which may not hold in the case of repeated measurements or clustered data. GEE addresses this limitation by incorporating correlation structures(Hedeker & Gibbons, 2006).

To estimate GEE let Y_{ij} represent the response for the i^{th} individual in the j^{th}cluster at time *t*. The GEE model is typically expressed through a mean model $E(Y_{ij}) = \mu_{ij}$ where μ_{ij} is a function of covariates.

The working correlation matrix V characterizes the within- cluster correlation. The GEE estimating equations take the form;

$$U(\beta) = \sum_{j=1}^{m} V_j^{-1}.R_j(\beta) = 0 \qquad (33)$$

Where β is the vector of parameters, $U(\beta)$ is the score function, V_j is the working correlation matrix for cluster j, $R_j(\beta)$ is the contribution to the score from the cluster *j* (Hedeker & Gibbons, 2006).

In longitudinal data, GEE is specifically designed for data collected over multiple time points or from clustered units, such as individuals within families or patients within hospitals. While in correlation structures GEE allows for specification of various

correlation structures e.g. exchangeable, autoregressive and unstructured to capture the dependence among observations within clusters(Hedeker & Gibbons, 2006).

In general GEE is a powerful tool for analyzing correlated and longitudinal data.

2.6 Review of Previous Research

The analysis of antibody titre data is crucial in understanding immune responses and vaccine efficacy. Longitudinal studies often encounter censored data, where values fall below detection limits, leading to challenges in accurate estimation. Researchers have addressed this issue by utilizing imputation techniques and linear mixed models to handle censored data effectively. Twisk & Rijmen (2009) emphasized the importance of considering censoring in data analysis for better interpretation.

A summary review of papers on antibody titre data reveals various approaches to handling censored data. Bonate et al. (2009) and Zhao (2017) used imputation techniques for endpoint titer and concentration data, respectively, while Persichetti et al. (2017) and Devanarayan (2017) opted for deletion of censored data. Van Stappen (2015) and Moraschini (2015) employed deletion and imputation methods for optical density and concentration data. These studies highlight the diversity in methods used to address censored data in antibody titre analysis.

The thesis adopted a censored mixed model to assess bias and underestimated variance resulting from inappropriate imputation techniques. The main objective was to compare the impact of sub-optimal analysis methods with principled longitudinal data models. By focusing on accurate estimation and addressing limitations in existing methods, the study aimed to enhance the reliability of antibody titre measurements.

Recent advancements in antibody titre estimation methods have focused on addressing measurement challenges and adjusting for censored data. New approaches, like the Adjustment for Bi-censoring (ABC) method, have been developed to handle measurements below the limit of detection effectively, ensuring robust estimations of antibody titres. Simulation studies have played a crucial role in developing and validating new methods, ensuring reliable and unbiased measurements.

2.6.1 Summary review of papers on antibody titre data.

Table 1 below summarizes review of a few different papers on antibody titre data, how the data was analysed, the antibody data type used and whether the data was longitudinal or not. And the methods used to deal with censored data.

Table 1: Summary review on Antibody titer data

Name of the Paper	Antibody data type	Longitudinal	Method to deal with the Censored Data
Bonate et al (2009)	Endpoint titer	Yes	Imputed to half detection limit
Persichetti et al (2017)	Endpoint titer	No	Deletion
Zhao (2017)	Concentration	Yes	Imputation
Van Stappen (2015)	Optical density	No	Deletion
Moraschini (2015)	Concentration	No	Imputed to half detection limit
Yang (2015)	Optical density	No	Imputed to half detection limit
Devanarayan (2017)	Concentration	No	Deletion

CHAPTER 3

METHODOLOGY

The data and techniques used in the study are covered in detail in this chapter.

3.1 Real-world data

This thesis project used primary real-world serology data from the Pneumococcal Vaccine for Vulnerable Populations in Africa (PCVPA) study (Swarthout et al., 2022). The 13-valent pneumococcal vaccine (PCV13) targets 13 serotypes of the pneumococcal bacterium *Streptococcus pneumoniae*. The PCV13 vaccine was introduced in 2011 using a 3+0 schedule, where one dose at each of 6 weeks, 10 weeks, and 14 weeks of age was given to under 1-year-olds (Swarthout et al., 2020). This has been effective in reducing the prevalence of nasopharyngeal carriage and invasive pneumococcal diseases. For example, a trial study in Malawi, specifically in Karonga, showed that PCV13 was effective compared with 2 years before the introduction of PCV. Vaccine serotype (VT) carriage among young PCV-vaccinated children (1-4 years of age) was 28.2% before vs. 16.5% after PCV introduction (Swarthout et al., 2020). However, even though it has been shown that there is reduced vaccine serotype (VT) carriage in Malawi, there is persistent residual carriage of all 13 vaccine serotypes among children vaccinated with PCV13. One plausible reason for this is the waning of the PCV13 vaccine after the first year of life(Swarthout et al., 2022).

An observational surveillance study targeting under five children from Blantyre using random sampling for pneumococcal carriage and repeated cross-sectional surveys was conducted from 2015–2019(Swarthout et al., 2020). For this thesis project, a subset of samples from PCVPA was used for serological assaying in the ongoing PAVE study, and 638 samples were randomly selected from the larger parent survey(Swarthout et al., 2022). Of these, 556 were primary samples and 82 were secondary samples(Swarthout et al., 2022). The data used in this project are serotype-specific immunoglobin G (IgG) levels of children aged 4 weeks to 60 months, which were

measured via ELISA. Enzyme-linked immunosorbent assay (ELISA) detects and measures antibodies and hormones in the blood. The assay relies on the principle of binding specificity between an antibody and an antigen (Crowther, 2000). When measuring these serotypes using ELISA, a surface is coated with a molecule of interest, such as an antigen or antibody. The sample containing the target is then added, allowing binding to occur. Thereafter, after washing away unbound material, an enzyme-linked secondary molecule is introduced that binds specifically to the target. Subsequent, addition of a substrate for the enzyme induces a colour change of intensity, which is proportional to the amount of the target substance. Then this colour change is measured, providing a quantitative assessment of the targets presence(Crowther, 2000). IgG is a type of antibody produced by plasma B cells in the human body and can serve as a proxy measurement of immunity.

3.2 Descriptive Statistics.

To summarize the results, the means, medians, interquartile ranges and confidence intervals for the estimated parameter for each of the three biomarkers were calculated. The primary interest was on i) bias (which model yields the least biased results, i.e. gets closest to the true value) and ii) the associated estimates of uncertainty (e.g. we would expect the imputation models to underestimate uncertainty associated with the parameter estimates).

3.3 Simulated data

As the focus of this project was an evaluation of different methods for modelling left-censored, cross-sectional IgG data, this project also simulated enzyme-linked immunosorbent assay (ELISA) data where the true effects of vaccination and the trend with age were known. During this simulation, we first simulated study participants and their characteristics. Specifically, we simulated patient IDs, number of visits, exposure statuses to check if participants were exposed to the pathogen, ages and sex. Because the ELISA assays used by the serosurvey had a lower limit of detection (DL) of 0.15 μg/mL, we used the same DL during the simulations, left-censoring any simulated IgG levels below the DL. To explore the performance of different modelling approaches, we simulated data for three hypothetical biomarkers, each representing a different scenario (specifically different levels of missing data, different strengths of effects of vaccine and trend with age). To mimic the real-world analysis process, we also

simulated samples of known concentration (so called standards) for standard curve fitting (ELISA measurements are optical density measurements and using a standard curve, these optical densities can be mapped back to actual antibody concentration levels).

Furthermore, we simulated antibody concentrations using a four-parameter logistic regression for all the three biomarkers. By taking the inverse of the four-parameter logistic curve and using the concentrations we simulated the optical densities. To simulate the entire dataset, we followed the following steps;

- Simulate the study participants and their characteristics $n = 100$. For example, the number of samples to be analysed which was $ns = 5$, probability for a randomly selected participant to have been exposed to the pathogen was set at $P = 0.4$, set the detection limit which was assumed to be the same for all biomarkers $dl = 0.15$.
- Simulate two data frames to retain information about patients and samples.
- Simulate participants at different time points, i.e. number of visits, because the data being simulated was longitudinal. These time points were five different visits and were assumed to be 1 year apart.
- Simulated age using gamma distribution and gender for each patient. In addition, exposure status (whether the patient has been exposed to pathogens) is simulated based on the specified probability of $P = 0.4$.
- Random effects for all three biomarkers were simulated using a normal distribution.
- Set values for the percentage of missing data, effect of age, effect of vaccine, effect of sex.
- Simulate theoretical average concentration measurements based on the above factors, i.e. gender, sex, exposure status and random effects.
- Add random noise to simulate natural fluctuations in IgG levels that were normally distributed. The resulting ‘noisy’ values are referred to as the true concentration levels.
- Random noise was added was in two levels of exponential distribution, because we simulated random variation for the actual concentration and the noise due to the measurement process.

- Set parameters for four parameter logistic curve. Using the inverse of this curve, we translate the true concentrations to true optical densities (OD).
- Add noise to mimic the imperfect measurement process in real life. These values are taken as the simulated IgG optical density measurements.

After the simulation, the data were analyzed using the following steps (replicating the way the real-world optical density data are processed and analyzed):

- Using least-squares estimation, fit a four-parameter logistic regression model to the simulated standard samples.
- Use the fitted curve to convert the simulated optical densities into estimated concentration levels for the simulated patient samples.

Thereafter, we removed the standards used when calculating the measured concentrations using least square estimation and saved the data to fit the various analysis models to investigate how well the different models can estimate the parameters used during the data simulation.

The data we simulated was longitudinal in nature because it involved repeated measurements on a set of subjects and was also left-censored given the detection limit we used. During this simulation, we generated one dataset with 100 participants and five samples per individual simulated at equal time points.

Given that the data were simulated, we set the values for each of the model parameters. From these, we can derive the true values of some parameters estimated by the models, specifically the model intercept. To derive the true values for the intercepts for the three sets of simulated data, we took the expectation of the antibody concentration random variable used during the simulations. Note that the variables can be considered to be random during the simulation process (since we drew random samples from specific parametric distributions, whereas in the models, the predictors are assumed to be fixed and the response variable random). Below are the calculations;

From equation (*) we take an expectation of $E(Y_{ij}) = E\left(\frac{\beta_1 S_{ij}+\beta_2 A_{ij}+\beta_3 E_{ij}+u_i+\varepsilon_{ij}+V_{ij}}{K}\right)$ (34)

Where S, A, and E are the factor variables Sex, Age, and Exposure.

I = index for individuals

j = index of observations within individuals

$\beta_1 \ldots \beta_3$ = parameters of the variables

U_i = Random effect

ε_{ij} , V_{ij} = Exponential distribution cases

K = Scaling parameter

In the above expectation, to derive the true value of the intercept, we only need to consider the terms that do not involve the independent variables A_{ij}, S_{ij} ,and E_{ij}. Therefore, we need to consider only:

$$E\left(\frac{U_i}{K} + \frac{\varepsilon_{ij}}{K} + \frac{V_{ij}}{K}\right) \qquad (35)$$

We note that $E(\frac{U_i}{K}) = 0,$ since $U_i$is normally distributed where U_i~N $(0,\sigma^2)$. There we are left with

$$E\left(\frac{\varepsilon_{ij}}{K} + \frac{V_{ij}}{K}\right) = \frac{\frac{1}{\lambda_1}}{K} + \frac{V_{ij}}{K} = \frac{\frac{1}{\lambda_2}}{K} \qquad (36)$$

Since ε_{ij} , $V_{ij}$are exponential random variables, they are both distributed exponentially with means $\frac{1}{\lambda_1}$, respectively $\frac{1}{\lambda_2}$and variances$\frac{1}{{\lambda_1}^2}$ and $\frac{1}{{\lambda_2}^2}$

Hence, from the simulation for first biomarker, lambda 1 = 0.5, and lambda 2 = 0.75, and K = 1. For second biomarker, rate 1 = 0.3, rate 2= 1, and K= 2, the last biomarker, which was the third one, rate 1= 0.75, rate2 = 0.4, and K = 2. These three biomarkers also differ in the proportion of missing data and all these values were chosen to make the biomarkers differ.

Therefore, substituting in the formulae yields;

Measured concentration of simulated biomarker1;

$$MC1 = \frac{{}^1/_{0.5}}{1} + \frac{{}^1/_{0.75}}{1}$$

$$= 2 + 1.33 = 3.33$$

Measured concentration of simulated biomarker 2;

$$MC2 = \frac{1/0.3}{2} + \frac{1/1}{2}$$

$$= 1.67 + 0.5 = 2.17$$

Measured concentration of simulated biomarker3;

$$MC3 = \frac{1/0.75}{2} + \frac{1/0.4}{2}$$

$$= 0.67 + 1.25 = 1.92$$

3.3.1 Data analysis of simulated data

For the analysis, we used a longitudinal mixed model but accounted for the simulated censored data in five different ways. We considered three different simple imputations (imputation of censored values to zero, imputation to the detection limit and imputation to half the detection limit). We also conducted a complete case analysis. Finally, as a fifth analysis approach, we used a censored regression model.

The linear mixed model is given below. Let Y represent the continuous outcome variable (IgG concentration). We included three fixed predictors; age of the patient, exposure status of the patient and sex of the patient. In addition, the models had a fixed intercept and a random effect for patient ID. During simulation, the data was log transformed; therefore, the longitudinal mixed model for the first scenario is given by

$$Y_{ij} = \beta_0 + \beta_1 Age + \beta_2 Sex + \beta_3 Exposure + \mu_i + \varepsilon_{ij} \qquad (37)$$

Where $\beta_1 \ldots \beta_3$ were fixed effects parameters age, exposure and sex fixed factors, β_0 was the fixed intercept and μ_i was the random effect associated with the individual i, i indexes individuals, j indexes observations within individuals and ε_{ij} was a residual error term assumed to follow a $N(0, \sigma^2)$ distribution.

For the first four analysis methods (imputations to 0, half the detection limit, detection limit and the complete case analysis), the above model was fitted using standard maximum likelihood estimation using the lme4 package and the lmer() function in the R environment for statistical computing (R Core Team, 2023).

For the censored regression model, however, the models were fitted slightly differently because the likelihood contributions from the censored observations were different from those from the observed measurements.

Specifically, due to left censoring, we did not observeY_{ij}, but rather φ_{ij}, where the values below the detection limit were censored. φ_{ij} took the value of $Y_{ij}$for $Y_{ij} > \rho_{ij}$and took the value ρ_{ij}, the known lower limit of detection for the *jth* response on subject *i*.

$$\varphi_{ij} = \begin{cases} y, & y > \rho \\ \rho, & y \leq \rho \end{cases}$$

Where ρ was the lower limit of detection (ρ = 0.15 µg/mL in our data)

Assuming a Gaussian random effect model, this allowed us to write down the log-likelihood below, for an observed data set

$$l(\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2) = \sum\sum (\log\left(f\left(\varphi_{ij}|\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2\right)\right) + \log(g(Y_{ij}|\varphi_{ij}, \sigma^2))) \quad (38)$$

Where; $f\left(\varphi_{ij}|\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2\right)$ represented the density function for the censored observation φ_{ij}

- $(g(Y_{ij}|\varphi_{ij}, \sigma^2)$ represented the density function for the observed measurement Y_{ij} given the censored values φ_{ij}.

These density functions can then be defined as below to explicitly express the log-likelihood;

$$f\left(\varphi_{ij}|\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2\right) = \Phi\left(\frac{(\varphi_{ij} - \beta_0 - \beta_1 Age - \beta_2 Sex - \beta_3 Exposure - \mu_i)}{\sigma^{(l(\varphi_{ij} \leq \rho_{ij}))}}\right)$$

$$\text{and} g\left(Y_{ij}|\varphi_{ij}, \sigma^2\right) = \phi\left(\frac{\left(\frac{Y_{ij} - \varphi_{ij}}{\sigma}\right)}{\sigma}\right) \quad (39)$$

Where; - Φ () represents the CDF of the standard normal distribution.

- Φ () represents the PDF of the standard normal distribution.
- L () represents the indicator function that equals 1 if the condition inside is true and 0 otherwise.

Lastly, the log-likelihood is then expressed as;

$$l(\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2) = \sum \log\left(\Phi\left(\frac{(\varphi_{ij} - \beta_0 - \beta_1 Age - \beta_2 Sex - \beta_3 Exposure - \mu_i)}{\sigma^{(l(\varphi_{ij} \leq \rho_{ij}))}}\right)\right) + \sum log\left(\phi\left(\frac{\left(\frac{Y_{ij} - \varphi_{ij}}{\sigma}\right)}{\sigma}\right)\right) \quad (40)$$

Where $\Phi(.)$ and $\phi(.)$ are the probability and cumulative density function, respectively, of the standard normal distribution.

To estimate the parameter values in this censored regression we maximised the log-likehood function. This was done using the lmer() function, which was used to fit the linear mixed-effects model to the data. The lmer() function optimizes the log-likelihood to find the parameter estimates that best describe the observed data.

3.3.2. Data analysis for PCVPA DATA

For analysis of the real-world PCVPA data apart from the four models, which were imputing to detection limit, imputing to half detection limit, imputing to zero, and using a complete case, we used censored regression model. Because these data had a detection limit of 0.15, to be specific, a lower limit detection means that observations were left censored. For easy interpretation these data were also log transformed, and we assumed a Gaussian linear model for y. Because of these data, we had a single variable, age in months. The natural logarithm of the measured IgG concentration with a single explanatory variable age, A is given by;

$$Y = \beta_1 + \beta_0 A + \varepsilon = f(A, \beta) + \varepsilon \qquad (41)$$

Where $\varepsilon \sim N(0, \sigma^2)$.

However, since the data were left censored with a lower limit of detection, we did not observe Y but rather y_N, where the values below the detection limit were censored. y_N took the value of Y for $Y > \rho$ and took the value ρ, the known lower limit of detection.

$$y_N = \begin{cases} y, & y > \rho \\ \rho, & y \leq \rho \end{cases}$$

Where ρ was the lower limit of detection (ρ = 0.15 µg/mL in our data).For we assumed Gaussian model, the observed data set $\{A_I, y_{N,i}\}_{i=1}^{n}$ log-likelihood can be written as;

$$l(\beta, \sigma^2) = \sum_{y_{N,i} > \rho} log\left(\phi\left(\frac{y_{N,i} - f(A,\beta)}{\sigma}\right)\right) + \sum_{y_{N,i} \leq \rho} log\left(\Phi\left(\frac{f(A,\beta) - \rho}{\sigma}\right)\right) \qquad (42)$$

Where $\phi(.)$ and $\Phi(.)$ are the probability and cumulative density functions of the standard normal distribution.

This thesis is a comparative study, and in simulated data, we incorporated the longitudinal aspect because we were trying to investigate whether the effects of age, gender and exposure status become more pronounced or worsen when working with longitudinal data. In addition, longitudinal studies introduce complexities, i.e. correlated measurements within subjects and potential time-dependent trend.

Specifically, this evaluation involves compares the performance of imputation methods, considering bias and variance metrics in both cross-sectional and longitudinal analyses.

3.4 Variables in the Study

The outcome of interest was the IgG concentration at different ages from 0 to 60 months. Age (in months) was used as the predictor variable for IgG concentrations of the different strains. In this dataset, we measured serotype-specific IgGs against the 13 vaccine serotypes namely; 1, 3, 4, 5, 6A, 6B, 7F, 9V, 14, 18C, 19A, 19F and 23F and two non-vaccine serotypes 12F and 33F, as well as IgGs against three pneumococcal proteins PsaA, NanA, and Ply. The population level was estimated, serotype-specific immunogenicity profiles were obtained using linear regression and censored regression models.

The simulation study used the same outcome variable (level of serotype-specific antibodies as measured by IgG concentration) for all three biomarkers but used three predictor variables: age, sex and exposure status.

Therefore, all outcome variables in both the PCVPA and simulation studies were continuous. The explanatory variables were chosen on the basis of the literature on bioassays and available data from the PCVPA study.

CHAPTER 4

RESULTS AND APPLICATION TO THE DATA

The data used are explained in detail in this chapter. Furthermore, the chapter provides technical details on the analysis methods that were used in the study.

4.1. Implementation

All analyses and simulations were implemented in R version 4.3.2 (R Core Team, 2023) with the use of packages censReg (Henningsen, 2022) for the censored regression likelihood, ggplot2 (Wickham, 2016) for the data and results visualization and boot (Cante & Ripley, 2022) for the bootstrapped confidence intervals.

4.2. Overview

This thesis project largely focuses on simulated data where we simulated IgG concentration data subject to a lower limit of detection, and deployed different methods of dealing with censored data. To show the implications of using imputation in the analysis of data subject to a lower limit of detection, we compared three different method of simple imputations (impution of censored data to 0, the detection limit or half the detection limit) and compared the results from analyses of these imputed data to the results from a complete case analysis and from using a censored regression model. The latter is statistically more principled than the other four methods because it makes full use of all the data and accounts for the uncertainty associated with left-censored observations. In the final set of analysis results, the PCVPA serosurvey data are used to illustrate the same five modelling approaches on real-world research data and validate some of the conclusions from the simulation study.

4.3 Descriptive Statistics for simulated data.

Table 2, summarizes the descriptive statistics for all the three biomarkers and explanatory variables. For example, in biomarker 1, the mean age for those exposed to

the pathogen when using the model Imputing to DL is 0.093. These results demonstrate how these variables differ under various modeling and imputation approaches.

Table 2: Descriptive Statistics for Biomarker 1, 2 and 3

Biomarker	Model	Variable	Mean	Median	Standard deviation
1	Imputed to DL	Age	0.093	0.092	0.045
		Exposed	0.933	0.925	0.329
		Gender	2.372	2.343	0.363
	Imputed to ½ DL	Age	0.102	0.102	0.046
		Exposed	1.011	1.002	0.333
		Gender	2.575	2.548	0.345
	Imputed to Zero	Age	0.111	0.110	0.048
		Exposed	1.089	1.084	0.341
		Gender	2.777	2.761	0.341
	Complete case analysis	Age	0.076	0.076	0.045
		Exposed	0.769	0.765	0.323
		Gender	1.892	1.883	0.322
	Censored regression	Age	0.107	0.107	0.049
		Exposed	1.052	1.035	0.347
		Gender	2.668	2.646	0.398
2	Imputed to DL	Age	0.102	0.237	0.030
		Exposed	1.011	1.219	0.219
		Gender	2.575	0.795	0.190
	Imputed to ½ DL	Age	0.263	0.263	0.031
		Exposed	1.296	1.297	0.219
		Gender	0.902	0.897	0.200
	Imputing to Zero	Age	0.288	0.288	0.033
		Exposed	1.371	1.372	0.223
		Gender	1.002	0.998	0.216

	Complete case analysis	Age	0.212	0.212	0.030
		Exposed	1.136	1.135	0.220
		Gender	0.697	0.690	0.191
	Censored regression	Age	0.258	0.258	0.032
		Exposed	1.281	1.280	0.222
		Gender	0.884	0.876	0.202
3	Imputed to DL	Age	0.111	0.107	0.032
		Exposed	2.420	2.365	0.444
		Gender	0.246	0.239	0.161
	Imputed to ½ DL	Age	0.135	0.132	0.029
		Exposed	2.617	2.584	0.381
		Gender	0.308	0.301	0.180
	Imputing Zero	Age	0.161	0.160	0.031
		Exposed	2.617	2.804	0.330
		Gender	0.308	0.364	0.210
	Complete case analysis	Age	0.177	0.772	0.029
		Exposed	2.086	2.060	0.377
		Gender	0.080	0.175	0.163
	Censored regression	Age	0.140	0.136	0.038
		Exposed	2.621	2.574	0.473
		Gender	0.321	0.308	0.195

4.4 Estimation of the effect of explanatory variables on concentration

As discussed above, we used simulated data to investigate the differences in results obtained from the five different models and to quantify how well these different methods performed.

Results for the parameters for each of the three biomarkers are shown on forest plots and presented in tables. These plots and tables show the true values of the model parameters against their estimates from the different models.

Knowing these values, we then compared these true intercept values with those estimated by the various models.

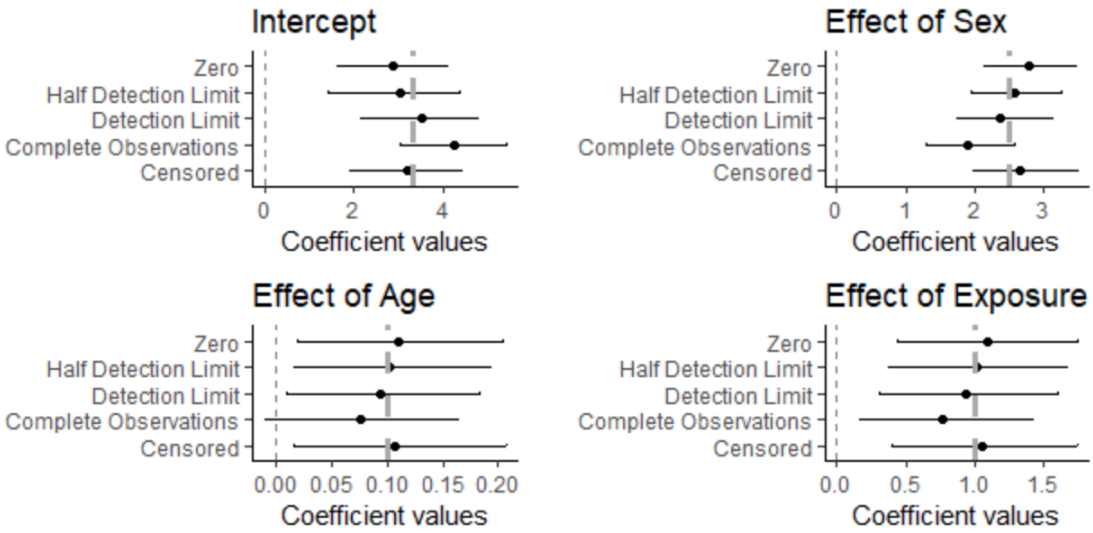


Figure 2: Forest plot of estimated parameters for the first biomarker. True values are indicated by the dashed grey vertical lines

The black dots are the point estimates of the effects of age, exposure, and sex on the first biomarker against the measured concentration, and the black horizontal segments represent the associated confidence intervals. The closer a point estimate is to the vertical true value line, the better (in terms of higher accuracy/lower bias) that particular model estimates the corresponding intercept or effect parameter for the first biomarker. From the graphs above, except for the intercept parameter, for which imputing to the detection limit results in the least bias, imputing to the half detection limit and using a censored model perform best, i.e., result in the least bias. From the graphs above, no substantial differences in the width of the confidence intervals can be seen; in other words, the different methods yield estimates with similar precision.

The above graphs are also summarized in Table 3, which shows the true values for each parameter and the point estimates together with 95% confidence intervals for the different modeling approaches.

Table 3: **Comparison of true values, point estimates values, and uncertainty values of different modelling approaches**

Biomarker	**Variable**	**True Value**	**Parameter estimates from the different modelling approaches**				
Biomarker1			Imputed to DL	Imputed to ½ DL	Imputed to zero	Complete case analysis	Censored regression
	Intercept	3.330	3.549 (2.168-4.765)	3.060 (1.428-4.366)	2.900 (1.609-4.113)	4.276 (3.046-5.432)	3.224 (1.920-4.420)
	Age	0.100	0.093 (0.010-0.183)	0.102 (0.015-0.193)	0.111 (0.019-0.204)	0.076 (-0.011-0.165)	0.107 (0.015-0.208)
	Exposed	1.000	0.933 (0.313-1.597)	1.011 (0.379-1.668)	1.089 (0.444-1.752)	0.769 (0.170-1.421)	1.052 (0.405-1.748)
	Sex Male	2.500	2.372 (1.735-3.120)	2.574 (1.955-3.252)	2.777 (2.138-3.468)	1.892 (1.304-2.566)	2.668 (1.976-3.503)

Table 3 shows the point estimates and confidence intervals to show variation for the intercept and the coefficient for the different predictor variables: age, exposure, and sex, in biomarker 1. Comparing these point estimates and confidence intervals with the true value, we were able to evaluate the accuracy and bias of each modelling approach for the first simulation approach.

The results in the table above show that imputing to half detection limit and using censored model performs the best; as their results are closer to the point estimate. However, imputation to half detection limit is the closest to the true value, which means that the result has the least bias. The table above also shows that imputing to half detection limit confidence interval is narrower whilst using censored model is wider, which means that imputing to half detection limit demonstrates greater degree of precision.

Therefore, these results mean that given the five models, it is good to impute data to half the detection limit despite the fact that using censored model performed well in point estimation, but imputing to the half detection limit was closest, which means that it introduces the least bias. The wider the confidence interval, the more uncertainty or variability is accounted for. Therefore, imputing to the half detection limit has a narrower confidence interval, which signifies a greater degree of precision, highlighting the method's ability to provide estimates with reduced uncertainty.

The graph below shows the effects of the explanatory variables on the second biomarker.

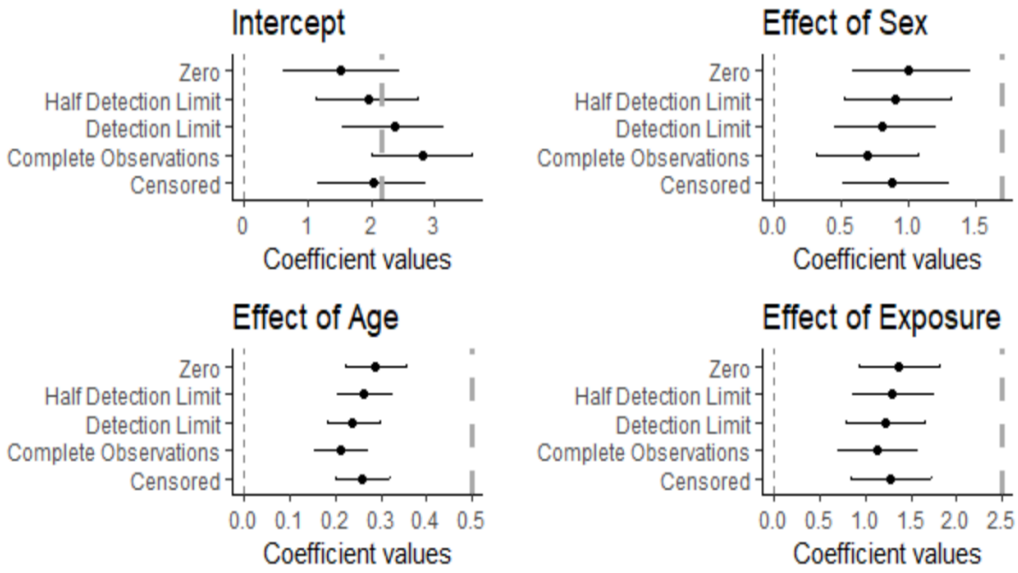


Figure 3: ***Forest plot of the second simulated biomarker***

From the graphs above, except for the intercept parameter, for which using the censored model results in the least bias, all methods performed poorly in estimating the effects of sex, age, and exposure. Moreover, none of the confidence interval from either method cover the true value of the three parameters. This shows that all models introduce bias and underestimate variance regarding explaining the effects of the three variables.

The above graphs are also summarized in Table 4, which shows the true values for each parameter and the point estimates together with 95% confidence intervals for the different modeling approaches.

The table 4 summarizes all the results explained in the graph above.

Table 4: Comparison of true values, point estimates values and uncertainty values of different modelling approaches for biomarker 2.

Biomarker	**Variable**	**True Value**	**Modelling approach**				
Biomarker 2			Imputed to DL	Imputed to ½ DL	Imputed to zero	Complete case analysis	Censored regression
	Intercept	2.17	2.371 (1.543-3.117)	1.950 (1.130-2.729)	1.528 (0.616-2.410)	2.820 (2.017-3.585)	2.028 (1.159-2.844)
	Age	0.50	0.238 (0.183-0.297)	0.263 (0.206-0.324)	0.288 (0.224-0.357)	0.212 (0.153-0.270)	0.258 (0.200-0.321)
	Exposed	2.50	1.223(0.783-1.659)	1.298(0.855-1.735)	1.371(0.942-1.810)	1.136 (0.705-1.557)	1.281 (0.838-1.731)
	Sex Male	1.70	0.802 (0.446-1.202)	0.902 (0.530-1.320)	1.002 (0.589-1.454)	0.697 (0.324-1.070)	0.884 (0.514-1.298)

Table 4 shows a comparison of true values, point estimates and uncertainty estimates in the form of confidence intervals for second biomarker.

From the above table results, it shows that the methods perform poorly with regard to the effects of age, exposure and sex. Except in intercept where censored model is used, the point estimate is closer to the true value and its confidence interval is narrower compared to the rest of the models. Despite the fact that all methods performed poorly for the effects of sex, age and exposure, the point estimate that was closer to the true value is imputation to zero and the confidence interval that includes the true value and is narrower is for the imputation to zero method.

The confidence intervals for imputation to detection limit, imputation to half detection limit, complete case and censored model all include the true value but have varying widths. Despite having different widths in terms of the confidence interval, the confidence interval for most of these methods was wider, thereby indicating more variability or uncertainty in the estimate.

In addition, most of these methods performed poorly because some simulation settings make estimation very difficult. For instance, an increase in exposure and age values in measured concentration two during simulation and a proportion of missing data since all the biomarkers had different proportions, resulted in all methods in this scenario yielding substantial bias and underestimation of uncertainty.

The graph below shows the effects of the explanatory variables on the third biomarker.

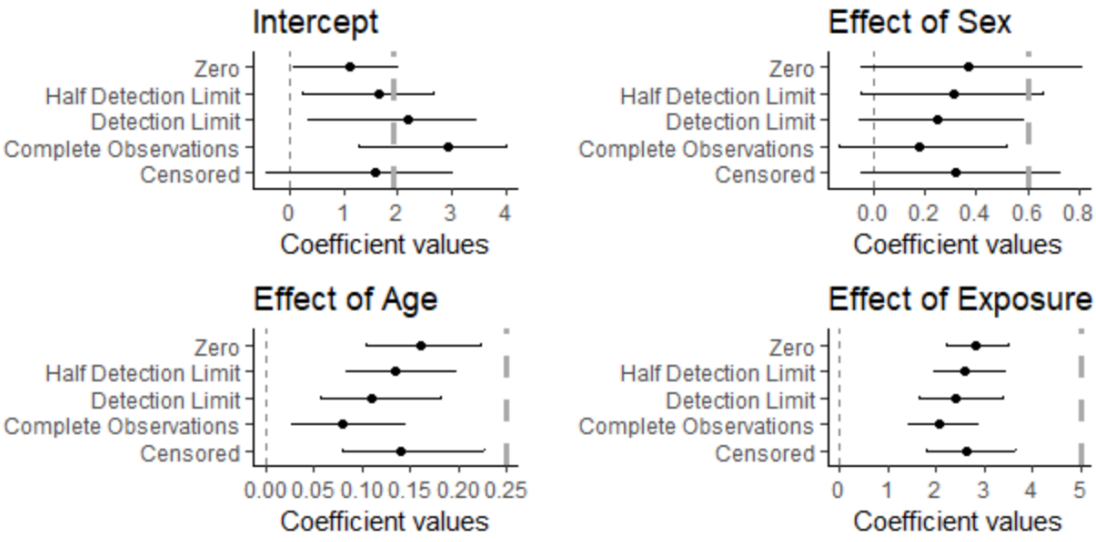


Figure 4: Forest plot of the third biomarker

From the graphs above, the imputing to half detection limit and imputing to detection limit methods prove to be closer to the true value, which means the least bias introduced in the results. On the other hand, graphs using censored model and imputing to zero have the least bias, even though all the models performed poorly in estimating the effects of age and exposure and not great in estimating the effect of sex. From the graphs above, no substantial differences in the width of the confidence intervals can be seen; in general, it is quite difficult to tell how different the confidence intervals are.

The above graphs are also summarized in Table 5, which shows the true values for each parameter and the point estimates together with 95% confidence intervals for the different modeling approaches.

Table 5: Comparison of true values, point estimates values, and uncertainty values of different modelling approaches for biomarker 3.

Biomarker	**Variable**	**True Value**	**Modelling approach**				
Biomarker3			Imputed to DL	Imputed to ½ DL	Imputed to zero	Complete case analysis	Censored regression
	Intercept	1.920	2.196 (0.358-3.415)	1.658(0.252-2.662)	1.120(0.090-1.981)	2.905(1.273-3.996)	1.590(-0.440-2.977)
	Age	0.250	0.110(0.057-0.182)	0.135(0.083-0.197)	0.161(0.105-0.223)	0.080(0.027-0.145)	0.140(0.081-0.226)
	Exposed	5.000	2.410 (1.650-3.379)	2.617(1.950-3.447)	2.824(2.215-3.496)	2.086(1.432-2.883)	2.621(1.819-3.661)
	Sex Male	0.600	0.246 (0.061-0.581)	-0.308 (0.053-0.663)	-0.370 (0.049-0.806)	-0.177(-(0.035-0.521)	0.321(-0.048-0.725)

The above table presents the point estimates and confidence intervals for the intercept and the coefficients for the different predictor variables in biomarker 3.

From the table above, the results show that all methods performed poorly in explaining the effect of gender and exposure and not well in explaining the effect of sex. Except in the intercept where imputing to the detection limit, the point estimate proves to be closer to the true value, but imputing to the zero method its confidence interval is narrower compared to the rest of the models. Despite all methods performed so poorly

to the effect of age and exposure and not great in explain the effect of sex, imputation to zero and using censored regression methods have least bias in all the graph.

In general, imputation to zero on the effect of exposure seems like narrower than the rest of the models, thereby suggesting, less variability in the system. In general, the imputation to zero modelling approach tends to result in narrower confidence intervals for most variables, whereas the censored regression modelling approach tends to result in wider confidence intervals.

Overall, the analysis of the three biomarkers reveals that imputing to half the detection limit and using the censored model demonstrate superior performance, with results closer to the point estimate. Notably, imputation to the half detection limit stands out as the method closest to the true value, indicating minimal bias. In addition, its narrower confidence interval, compared to the wider interval of the censored model, signifies a greater degree of precision.

However, a general observation across all methods revealed poor performance in explaining the effects of age, exposure and sex. An exception is found in the intercept, where the censored model demonstrates a closer point estimate to the true value and a narrower confidence interval.

Furthermore, it was noted that confidence intervals, especially in biomarker 2 and 3, all methods encompass the true value in explaining some variables, but exhibit varying widths. Despite differences in width, most intervals are wider, indicating more variability or uncertainty in the estimates. Differences in simulation settings, such as varying proportions of missing data and increase in exposure and age values for these biomarkers, contribute to substantial bias and underestimation of uncertainty across all methods.

4.5 PCVPA Study Results

The population average IgG age profiles in the PCVPA study were nonlinear. For comparing the methods used in this thesis, linear models were used. The work in this thesis was the groundwork for deciding on the methodology for the wider analysis project of the PCVPA IgG data where non-linear functional forms were used together

with censored, mixed regression models (Swarthout et al., 2022), but the extension to non-linear functional forms was beyond the scope of this thesis project. The assays' lower limit of detection was 0.15 μg/mL, which meant that the observations below this limit were left censored. To account for left censoring, the censored regression model was deployed as a method used.

4.5.1 Descriptive statistics for PCVPA data

638 plasma samples were evaluated in this study, of which 556 were primary samples and 82 secondary samples that were linked to each primary sample. From the table below, most serotype log transformed means are different across imputation methods. The data were log-transformed for easy interpretation; therefore, the means calculated below are geometric means. For instance, serotypes 6A, 19A, 19F and 23F show notable difference in geometric means between imputations.

In addition, larger standard deviations indicate greater variability for, serotypes 6A, 19A, 19F and 23F, which contain higher standard deviations, thereby suggesting increased variability in IgG concentration for these serotypes.

Furthermore, from the table below, it can be easily seen that the confidence intervals for the censored regression approach in most serotypes are wider as compared to the rest of the models

.

Table 6: Descriptive Statistics for all serotypes

Serotype	Model	Mean	Standard deviation	Confidence Interval
1	Imputed to DL	0.311	0.043	0.245 -0.388
	Imputed to Half DL	0.268	0.041	0.205 -0.341
	Imputed to Zero	0.155	0.040	0.098 -0.230
	Censored regression	0.275	0.043	0.171 -0.410
3	Imputed to DL	0.326	0.014	0.303-0.351
	Imputed to Half DL	0.254	0.023	0.218-0.294
	Imputed to Zero	0.105	0.026	0.068-0.153
	Censored regression	0.234	0.027	0.136-0.349
4	Imputed to DL	0.324	0.017	0.298-0.352
	Imputed to Half DL	0.255	0.012	0.236-0.275
	Imputed to Zero	0.127	0.003	0.122-0.133
	Censored regression	0.245	0.012	0.180-0.350
5	Imputed to DL	0.562	0.129	0.375-0.800
	Imputed to Half DL	0.537	0.130	0.349-0.779
	Imputed to Zero	0.465	0.140	0.270-0.733
	Censored regression	0.546	0.130	0.309-0.935
6A	Imputed to DL	0.841	0.288	0.448-1.401
	Imputed to Half DL	0.811	0.290	0.419-1.378
	Imputed to Zero	0.728	0.296	0.340-1.320
	Censored regression	0.820	0.292	0.364-1.717
6B	Imputed to DL	0.620	0.153	0.400-0.903
	Imputed to Half DL	0.587	0.152	0.370-0.871
	Imputed to Zero	0.493	0.153	0.280-0.786

	Censored regression	0.598	0.154	0.323-1.064
7F	Imputed to DL	0.469	0.162	0.249-0.784
	Imputed to Half DL	0.443	0.164	0.223-0.766
	Imputed to Zero	0.370	0.163	0.160-0.700
	Censored regression	0.452	0.166	0.190-0.906
9V	Imputed to DL	0.461	0.080	0.341-0.605
	Imputed to Half DL	0.415	0.083	0.293-0.565
	Imputed to Zero	0.296	0.093	0.167-0.475
	Censored regression	0.423	0.085	0.251-0.699
14	Imputed to DL	1.179	0.018	1.149-1.209
	Imputed to Half DL	1.144	0.012	1.123-1.164
	Imputed to Zero	1.030	0.004	1.024-1.036
	Censored regression	1.149	0.015	0.862-1.437
18C	Imputed to DL	0.330	0.037	0.273-0.394
	Imputed to Half DL	0.260	0.035	0.207-0.321
	Imputed to Zero	0.125	0.025	0.088-0.171
	Censored regression	0.245	0.036	0.153-0.406
19A	Imputed to DL	1.626	0.749	0.674-3.162
	Imputed to Half DL	1.601	0.744	0.657-3.128
	Imputed to Zero	1.525	0.738	0.600-3.054
	Censored regression	1.607	0.747	0.554-3.892
19F	Imputed to DL	1.539	0.496	0.856-2.493
	Imputed to Half DL	1.525	0.493	0.846-2.473
	Imputed to Zero	1.456	0.512	0.762-2.456

	Censored regression	1.530	0.495	0.710-2.981
23F	Imputed to DL	0.381	0.030	0.333-0.432
	Imputed to Half DL	0.308	0.032	0.259-0.363
	Imputed to Zero	0.159	0.025	0.121-0.202
	Censored regression	0.291	0.034	0.190-0.450
33F	Imputed to DL	0.242	0.041	0.181-0.314
	Imputed to Half DL	0.172	0.042	0.109-0.253
	Imputed to Zero	0.064	0.031	0.025-0.128
	Censored regression	0.145	0.054	0.054-0.319

4.5.2 Estimation of effect of age on IgG concentration.

As discussed in the previous chapters, we used censored regression models to account for lower limits of detection and contrast this to imputations of censored values DL, DL/2 and imputed to zero, but because of log transformation 0 could not be used instead 0.0001 given the logarithm was taken of the IgG values. The antibody titre data (IgG data) are naturally skewed, and it is more meaningful to discuss fold changes than absolute differences; therefore, before model fitting, the data were log transformed so that the fitted arithmetic mean corresponded with the log of the geometric mean in the original data scale. We had IgG concentrations for 14 serotypes. For all serotypes, we imputed the data to the detection limit, half detection limit, zero (0.0001), and censored regression models.

We used bootstrapping and the percentile method to calculate the confidence intervals for the best-fit linear model for all serotypes. The results of all serotypes are presented in the graphs below, where the geometric mean at each age point in months was estimated. The results presented in the graphs below have the following colour code: blue represents imputing to zero, red represents imputing to the detection limit, orange represents imputing to the half detection limit, and black represents the censored model. The black dots are geometric means of concentration for all data points within each 3-month age band, and the grey dots are IgG titre data points for each sample, while the

shaded area is the 95% confidence band for each model represented in the colour of the line of model fit.

The main findings of these are summarised with only a handful of selected serotypes, because several serotypes were analysed. The graphs below show the results for serotypes: serotype 1, serotype 6A and serotype 4.

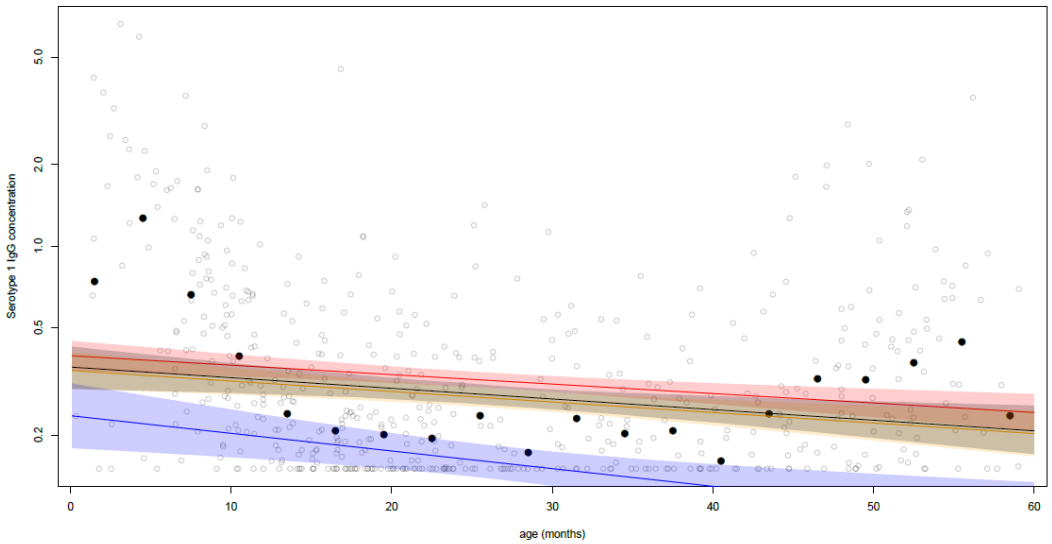


Figure5: Different linear regression models for serotype 1.

Because the use of these data was to compare which model performed better, we fitted all four models to serotype 1 data, which showed the relationship between age and IgG. However, the focus was on different model fits. The above linear graph shows that when you impute to zero, it introduces the least bias because all 20 geometric means lie above the fitted line. When you impute the data to the detection limit, the model fit was biased high, as most (11 out of 20) of the geometric means lie below the fitted line, and it is the model fit that is highest among all four shown approaches. From the graph above, it can also be seen that imputing to zero has a wider confidence band which, means greater variability.

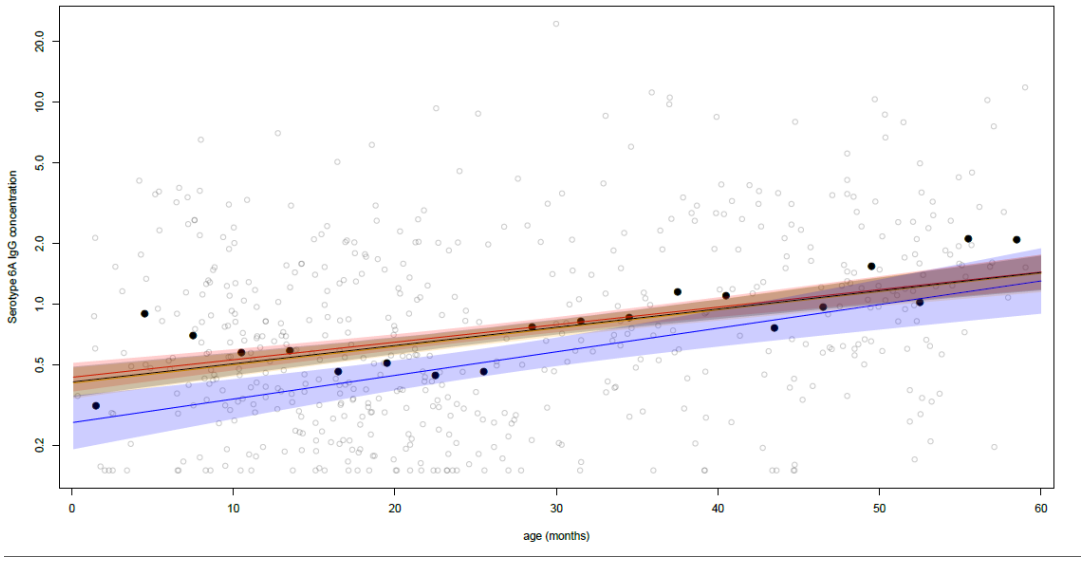


Figure 6: Different linear regression models for serotype 6A

The above graph of serotype 6A showed not much difference in terms of bias because almost all geometric means lie above and within in all methods. However, even though it is like that imputing to zero showed least bias for only three geometric means are above the fitted line. The above graph also shows that imputing to zero has a wider confidence band than the rest of the model, which suggests an increased variability in IgG concentration.

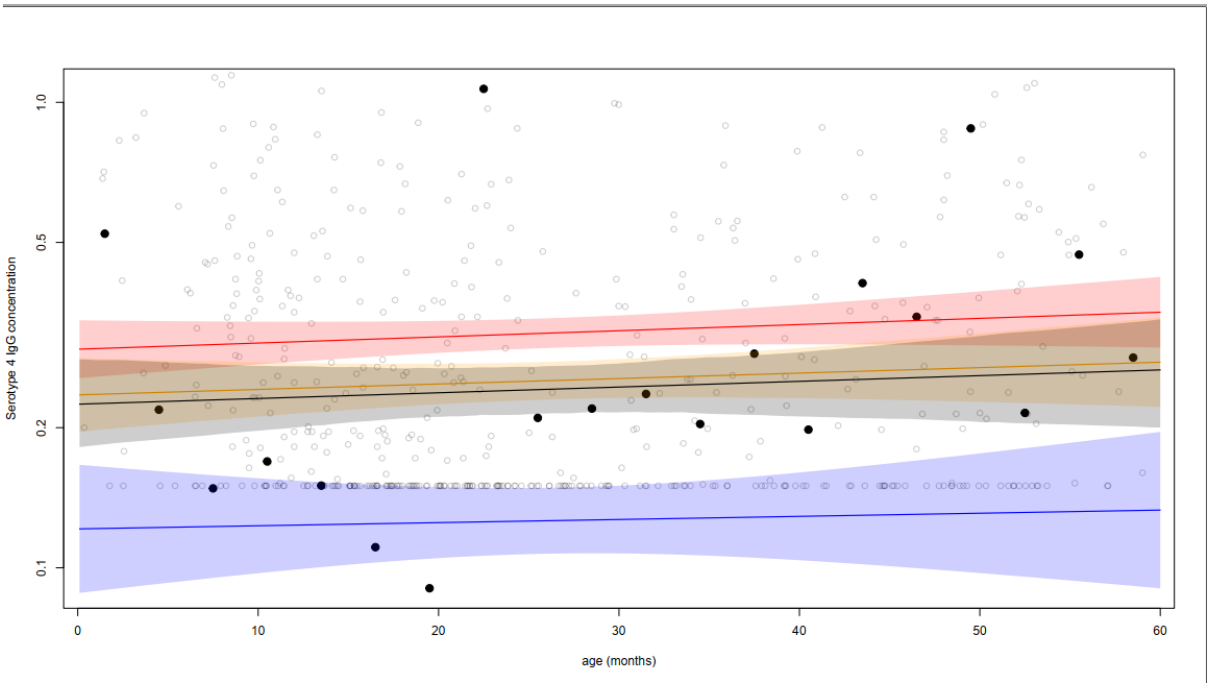


Figure 7: Four linear regression models on serotype 4

When three separate models were utilized, all three graphs of various serotypes (1, 6A, and 4) exhibited a similar pattern. This was also the case since the detection limit utilized was a positive 0.15 g/mL. The above graph also shows how imputation to DL biases high, as it can be seen that 15 out of 20 geometric means lie below the model fit. Moreover, imputation to zero biases low as it can be seen that all geometric means lie above the model fit. From the graph it is hard to tell which model underestimate variance since some of the confidence bands are not clear, for instance, imputing to zero.

In general, imputation approaches are simple but ignore the uncertainty associated with censored observations. In addition, imputing to zero or imputing to DL there is a risk of biasing low or high respectively, and imputing to half detection limit often performs well in terms of bias and is often very similar to the censored regression approach. In principle, the censored regression approach is the most statistically principled, and the only one correctly handling uncertainty associated with the censored observations. However, it is also the least simple to implement and requires more advanced statistical expertise. However, in practice, on the PCVPA data, the confidence intervals from the censored regression model look very similar in terms of width to those from imputing to half detection limit (DL/2); thus, so that this advantage could be limited in practice.

CHAPTER 5

DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS

Discussion of the results of the study as well as conclusion and recommendations are presented in the section below.

5.1. Discussion

The confidence bands of the 95% confidence interval changed depending on the model used. As previously stated, in principle approach confidence intervals from the censored regression turn to be wider as they account for the extra uncertainty of the censored observations which, the imputation method ignores. However, some imputation techniques resulted in wider confidence intervals, for example, imputing to zero in some cases. This was only when the imputation methods impute to values far away from the non-censored observations, which resulted in wider confidence intervals.

Furthermore, in this study, the point estimates increased or decreased depending on the model used in both data sets. The study also showed that the censored regression model performed well as compared to imputation to DL, imputation to half DL, imputation to zero, and the use of complete observational models, for it was less likely to underestimate variance and in many of the simulated data examples, was the one with least bias (though imputation to half the DL performed better in a few cases).

Although in the second and third biomarkers in simulated data, all methods performed so poorly in explaining the effect of variables (age, sex and exposure) and also not so great in explaining the effect of sex in third biomarker. Imputation to zero deemed to be closer to the true value in most of the variables, thereby least bias was introduced. And also, in second biomarker it was seen that imputing to zero had a narrow confidence interval which meant less variability even though it was had to conclude about confidence intervals, since confidence interval (CI) of all methods vary in width.

This study also used linear models in both simulated and PCVPA data despite having population profiles that were non-linear for comparison purposes. However, assessing the impact of functional form on performance was beyond the scope of this thesis project. The author of this thesis contributed to the wider modelling project of the PCVPA data that included non-linear function forms (see (Swarthout et al., 2022) where the author of this thesis is a co-author). Use of the Kaplan-Meier method when data distribution is not known (Canales et al., 2018). The disadvantage of this method is that it does not perform well in multivariate analysis and is also not excellent with left-censored data compared with right-censored data. The advantage of this model is that it uses the entire dataset and is not restricted to using summary statistics for the dataset. Another method is the use of substitution with a limit of detection DL (Canales et al., 2018); this is done by halving the DL with a square root of half. The disadvantage of this method is that it introduces error even though it is minimal, especially when large portions of a data set are below the DL (Canales & etal, 2018), and it does not account for left censoring in the data. Another method for dealing with left-censored data is the use of maximum likelihood estimation (MLE). Since in censored observations likelihoods are derived directly rather than for imputed values. Even though in imputation techniques uses maximum likelihood estimation.

The use of a geometric mean in this study to be specific to the PCVPA study helped in the interpretation. Because the antibody titre data are naturally on a logarithmic scale, the use of an arithmetic mean could have affected the interpretation of the results.

Furthermore, the longitudinal aspect was incorporated in the simulated data; this helped to take correlation into account, and from the results, it has been shown that the longitudinal censored model performed better in all three biomarkers because it did not underestimate variance, and bias was not introduced as it was seen in the first biomarker results. The point estimates in the censored model were closer to the true value, and the confidence interval was wider, although in some biomarkers, it was difficult to draw conclusion on width.

5.2. Conclusion

The aim of this study was to assess the impact of suboptimal analysis methods on statistical inference, mainly by quantifying the bias and underestimation of variance

due to simple imputation methods. We also to assessed the interpretation challenges when the logarithmic scale is ignored and evaluated these issues in a longitudinal data setting. In this study, we looked at the use of four different imputation techniques: imputation to detection limit, imputation to half detection limit, imputation to zero, and the use of complete observations in the data as well as the use of censored models when there is censoring in the data.

The findings suggest that the use of imputation techniques in data analysis introduces bias and underestimates uncertainty. This has been evident in the simulated data for all three biomarkers, where imputation techniques, for example, imputing to the detection limit, have proved to be a worse model than the rest of the models, for it underestimates variance and introduces bias. It has also been noted that despite other imputation methods performing better than the censored regression approach at times this was not consistently so the censored regression method performed best on average across simulation scenarios. Of greater concern is the underestimation of variance, which can lead to type I errors.

The results showed that the use of arithmetic mean in data like IgG data is not a good indicator of central tendency in naturally logarithmic data, and that interpretation is straightforward in log-transformed data or when a log link is used. This study has also revealed that if the logarithmic scale is ignored and the use of simple imputation in longitudinal data settings, it may lead to similar outcomes, such as underestimating variance and introducing bias in the data, as has been seen in simulated data where longitudinal censored mixed models have proven to perform better on average across all three biomarkers as compared to linear mixed models. The use of log-transformed data in PCVPA data also made interpretation easier.

In conclusion, the best method to use is the censored regression model, which accounts for censoring in the data. This comparative analysis study improved our understanding of the methods of data analysis and assessed the impact on parameter estimates and interpretation of inappropriate methods.

5.3. Recommendations

It is important to emphasize the importance of thorough data examination before applying imputation methods, highlighting the need for analysts to carefully assess the dataset to ensure the appropriateness of the chosen technique.

Moreover, when dealing with log-transformed data containing censoring and detection limits, it is better to use censored regression models over imputation techniques. Because of the statistical robustness an accuracy offered by censored regression models in handling the complexities of censored data, ensuring more reliable and precise results in the analysis of such datasets.

The study also suggests that when data exhibits non-linear profile, the use of linear spline regression models can be ideal and more effective. Linear spline regression models are appropriate to be used in order to capture the non-linearity in the data, thus giving a best-fit representation of the underlying pattern and trend. Using such linear spline regression models for non-linear data profiles, researchers can increase the model's power to capture certain complexities in the data. This elevates the quality and totality of the statistical analysis and interpretation.

REFERENCES

Amemiya, T. (1984). Tobit models. *A Survey Journal of Econometrics*.

Armbruster, D. A., & Pry, T. (2008). Limit of blank, limit of detection and limit of quantitation. *The Clinical Biochemist. Reviews*, *29 Suppl 1*(Suppl 1), S49-52.

Brown, H., & Prescott, R. (2015). *Applied mixed models in medicine* (Third edition). John Wiley & Sons Inc.

Cameron, A., & Pravin, K. (2005). *MicoEconomerics Methods and Application*. Cambridge University Press.

Canales, R. A., Wilson, A. M., Pearce-Walker, J. I., Verhougstraete, M. P., & Reynolds, K. A. (2018). Methods for Handling Left-Censored Data in Quantitative Microbial Risk Assessment. *Applied and Environmental Microbiology*, *84*(20), e01203-18. https://doi.org/10.1128/AEM.01203-18

Cante, A., & Ripley, B. (2022). *boot: Bootstrap R (S-Plus) Functions*. *R package version 1.3-28.1*.

Collett, D. (2015). *Modelling survival data in medical research* (Third edition). CRC Press, Taylor & Francis Group.

Crowther, J. R. (2000). *ELISA Guidebook, The* (Vol. 149). Humana Press. https://doi.org/10.1385/1592590497

Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap Methods and their Application* (1st ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511802843

Fitzmaurice, G. M., Laird, N. M., & Ware, J. H. (2011). *Applied longitudinal analysis*. Wiley.

Ge, Y., Zane Billings, W., Skarlupka, A. L., Cao, W., Dobbin, K. K., Ross, T. M., Handel, A., & Shen, Y. (2022). *Improving Estimation of Intervention Impact on Antibody Titer Increase in the Presence of Missing Values: A Proposed Method for Addressing Limit of Detection Issues*. https://doi.org/10.1101/2022.08.25.22279230

Harrell, F. E. (2001). Cox Proportional Hazards Regression Model. In F. E. Harrell, *Regression Modeling Strategies* (pp. 465–507). Springer New York. https://doi.org/10.1007/978-1-4757-3462-1_19

Hedeker, D. R., & Gibbons, R. D. (2006). *Longitudinal data analysis*. Wiley-Interscience.

Henningsen, A. (2022). *censReg: Censored Regression (Tobit) Models*. *0.5-36*. https://CRAN.R-project.org/package=censReg

Hosmer, D. W., Lemeshow, S., & May, S. (2008). *Applied Survival Analysis: Regression Modeling of Time-to-Event Data* (1st ed.). Wiley. https://doi.org/10.1002/9780470258019

Klein, J. P., & Moeschberger, M. L. (2003). *Survival analysis: Techniques for censored and truncated data* (2nd ed). Springer.

Kleinbaum, D. G., & Klein, M. (2005). *Survival analysis: A self-learning text* (2. ed). Springer.

Leung, K.-M., Elashoff, R. M., & Afifi, A. A. (1997). CENSORING ISSUES IN SURVIVAL ANALYSIS. *Annual Review of Public Health*, *18*(1), 83–104. https://doi.org/10.1146/annurev.publhealth.18.1.83

Olivier, J., Johnson, W. D., & Marshall, G. D. (2008). The logarithmic transformation and the geometric mean in reporting experimental IgE results: What are they and when and why to use them? *Annals of Allergy, Asthma & Immunology*, *100*(4), 333–337. https://doi.org/10.1016/S1081-1206(10)60595-9

R Core Team. (2023). *R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing*.

Swarthout, T. D., Fronterre, C., Lourenço, J., Obolski, U., Gori, A., Bar-Zeev, N., Everett, D., Kamng'ona, A. W., Mwalukomo, T. S., Mataya, A. A., Mwansambo, C., Banda, M., Gupta, S., Diggle, P., French, N., & Heyderman, R. S. (2020). High residual carriage of vaccine-serotype Streptococcus pneumoniae after introduction of pneumococcal conjugate vaccine in Malawi. *Nature Communications*, *11*(1), 2222. https://doi.org/10.1038/s41467-020-15786-9

Swarthout, T. D., Henrion, M. Y. R., Thindwa, D., Meiring, J. E., Mbewe, M., Kalizang'Oma, A., Brown, C., Msefula, J., Moyo, B., Mataya, A. A., Barnaba, S., Pearce, E., Gordon, M., Goldblatt, D., French, N., & Heyderman, R. S. (2022). Waning of antibody levels induced by a 13-valent pneumococcal

conjugate vaccine, using a 3 + 0 schedule, within the first year of life among children younger than 5 years in Blantyre, Malawi: An observational, population-level, serosurveillance study. *The Lancet Infectious Diseases*, *22*(12), 1737–1747. https://doi.org/10.1016/S1473-3099(22)00438-8

Tholen, D. W. (2004). *Protocols for determination of limits of detection and limits of quantitation: Approved guideline*. NCCLS.

Turkson, A. J., Ayiah-Mensah, F., & Nimoh, V. (2021). Handling Censoring and Censored Data in Survival Analysis: A Standalone Systematic Literature Review. *International Journal of Mathematics and Mathematical Sciences*, *2021*, 1–16. https://doi.org/10.1155/2021/9307475

Twisk, J., & Rijmen, F. (2009). Longitudinal tobit regression: A new approach to analyze outcome variables with floor or ceiling effects. *Journal of Clinical Epidemiology*, *62*(9), 953–958. https://doi.org/10.1016/j.jclinepi.2008.10.003

Wehrens, R., Putter, H., & Buydens, L. (2000). The Bootstrap; A tutorial. *Chemometrics and Intelligent Laboratory Systems*.

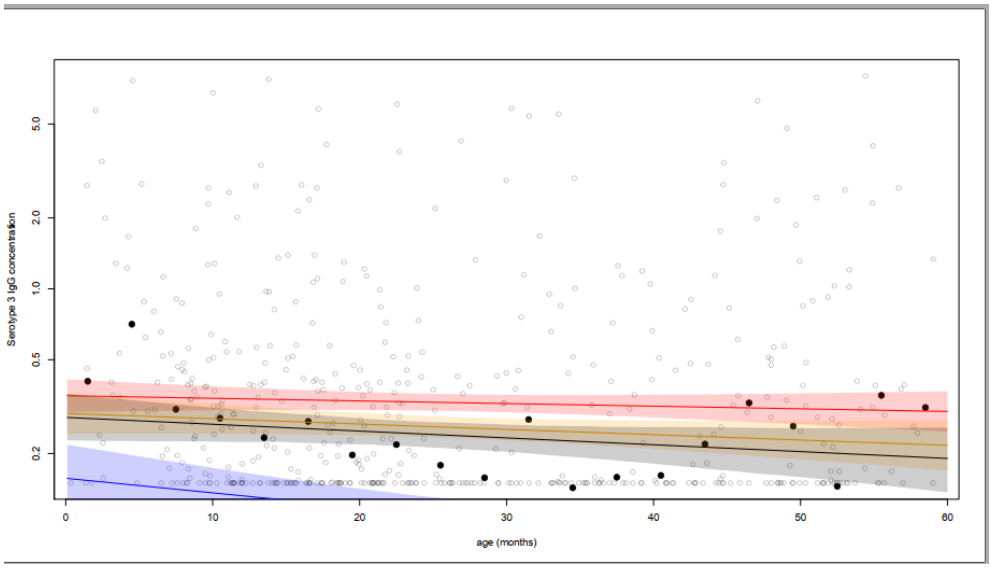
Weiss, R. E. (2005). *Modeling longitudinal data*. Springer.

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. *Springer-Verlag New York,*.

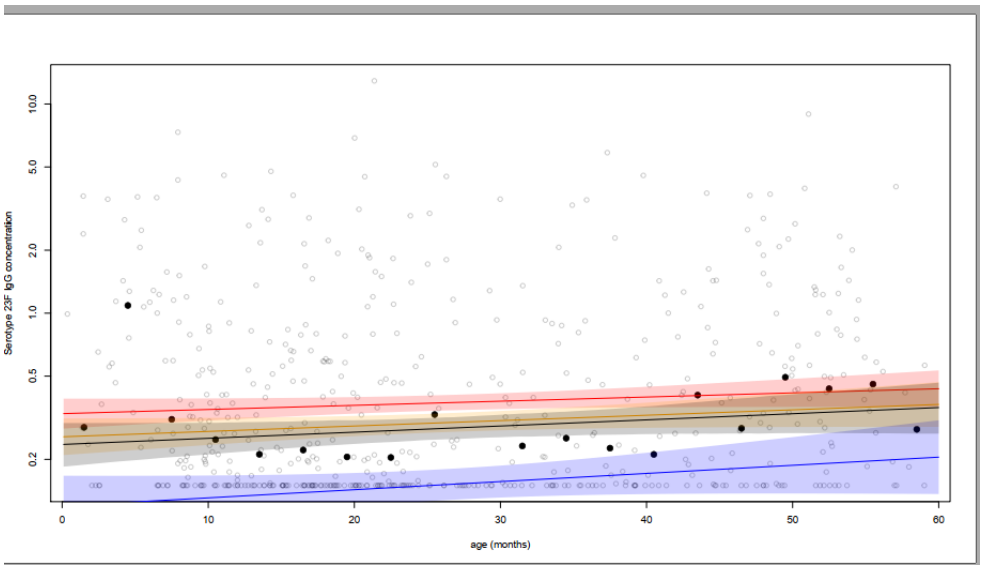
Yang, H., Baker, S. F., González, M. E., Topham, D. J., Martínez-Sobrido, L., Zand, M., Holden-Wiltse, J., & Wu, H. (2016). An improved method for estimating antibody titers in microneutralization assay using green fluorescent protein. *Journal of Biopharmaceutical Statistics*, *26*(3), 409–420. https://doi.org/10.1080/10543406.2015.1052475

APPENDIX

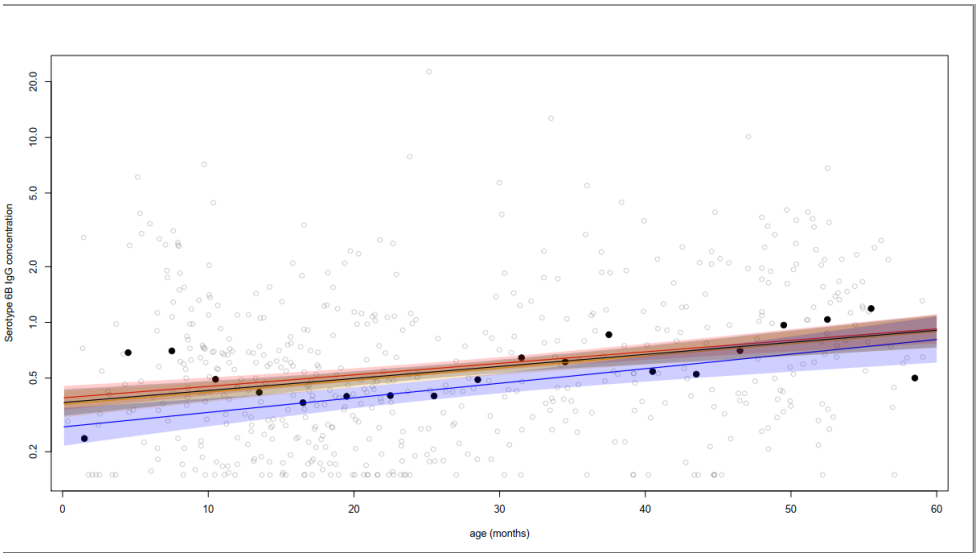
PCVPA DATA GRAPHS



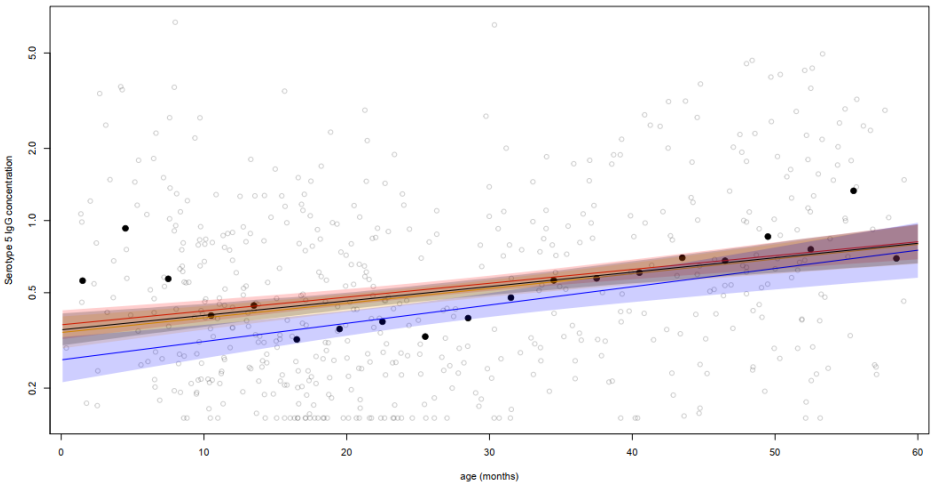
Serotype3 graph



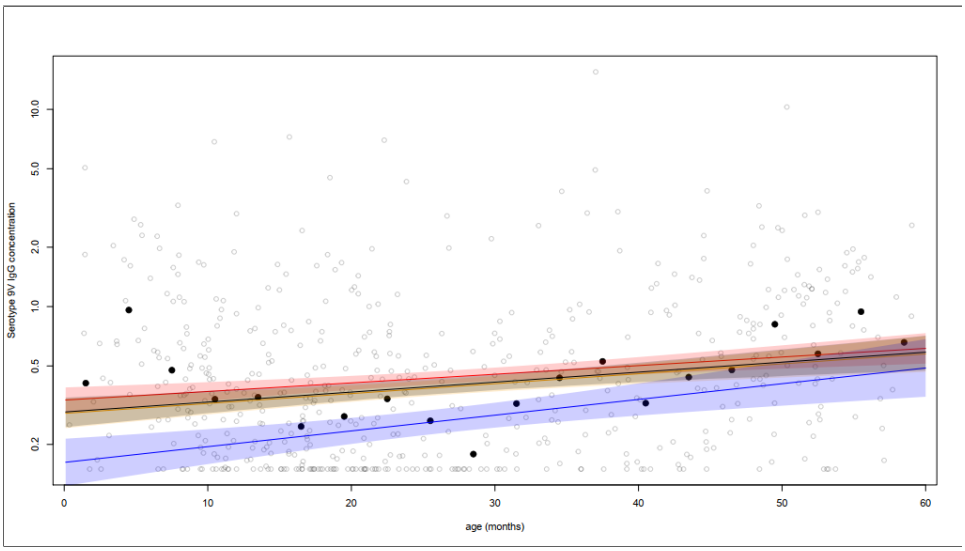
Serotype 23F graph



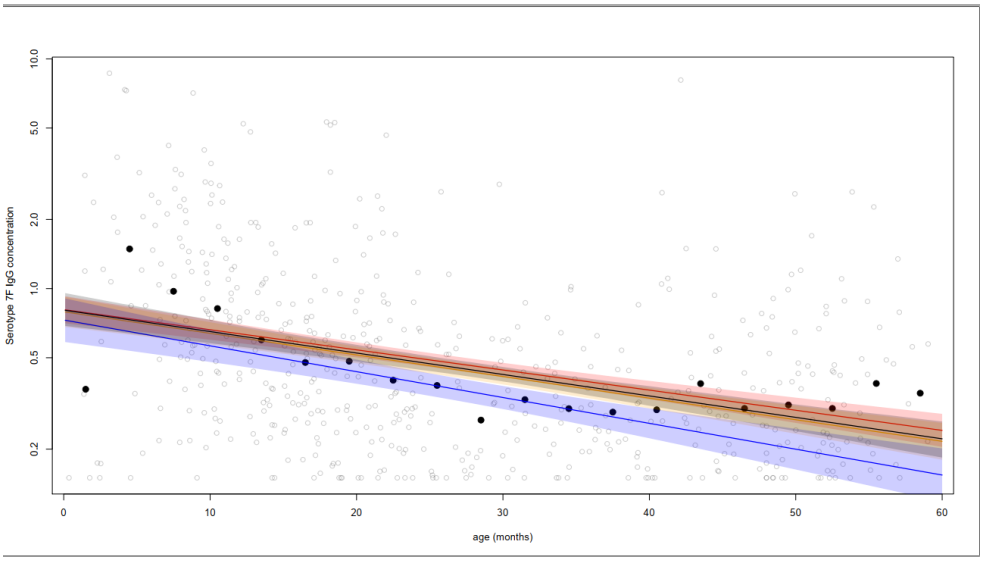
Serotype 6B graph



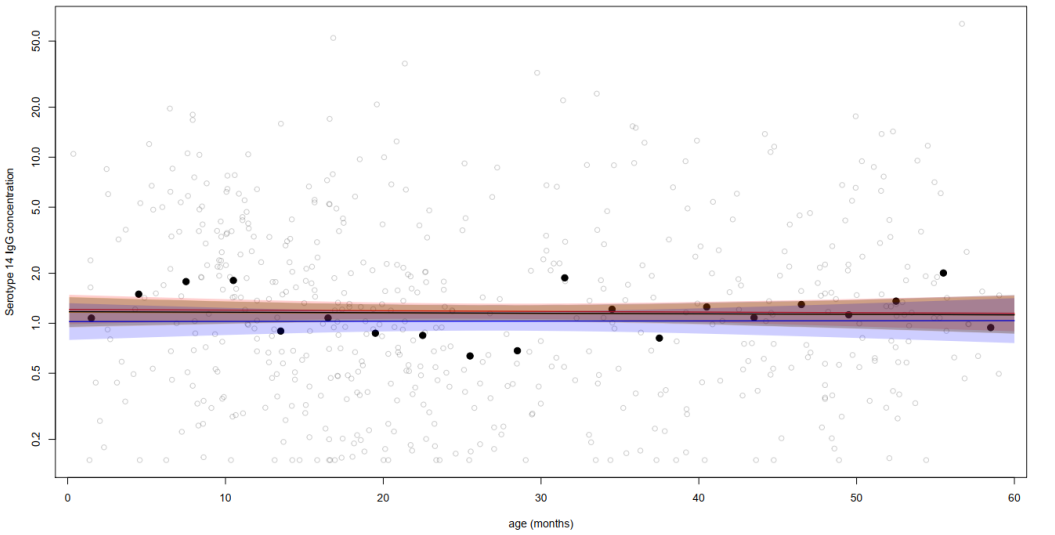
Serotype 5 graph



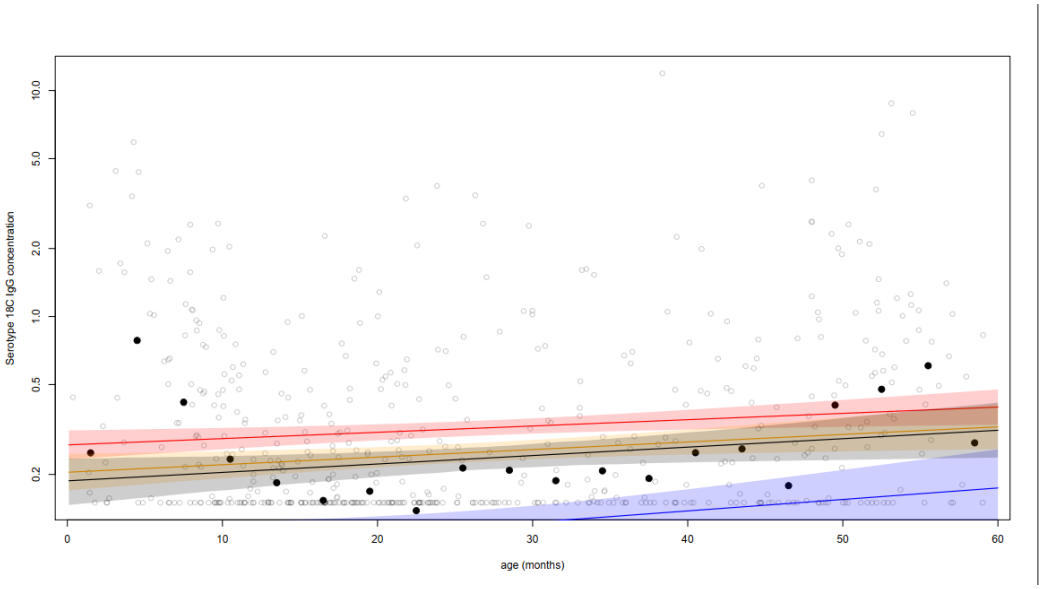
Serotype 9V graph



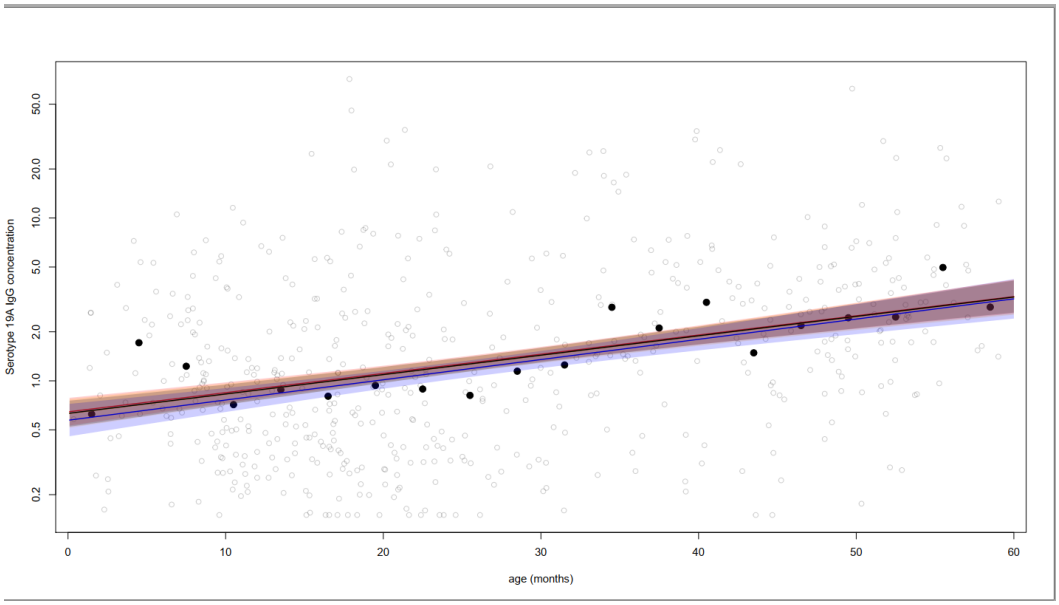
Serotype 7F graph



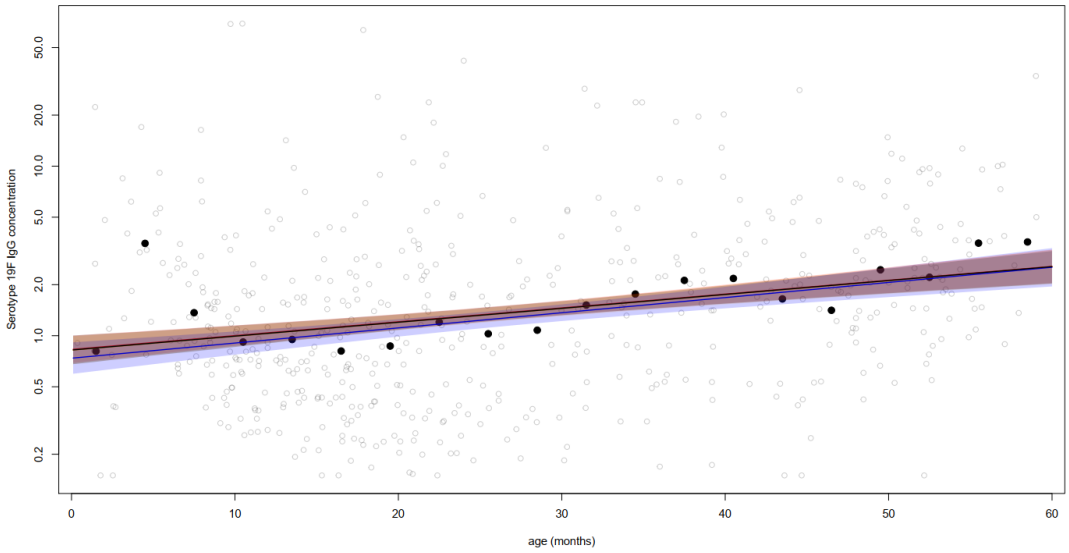
Serotype 14 graph



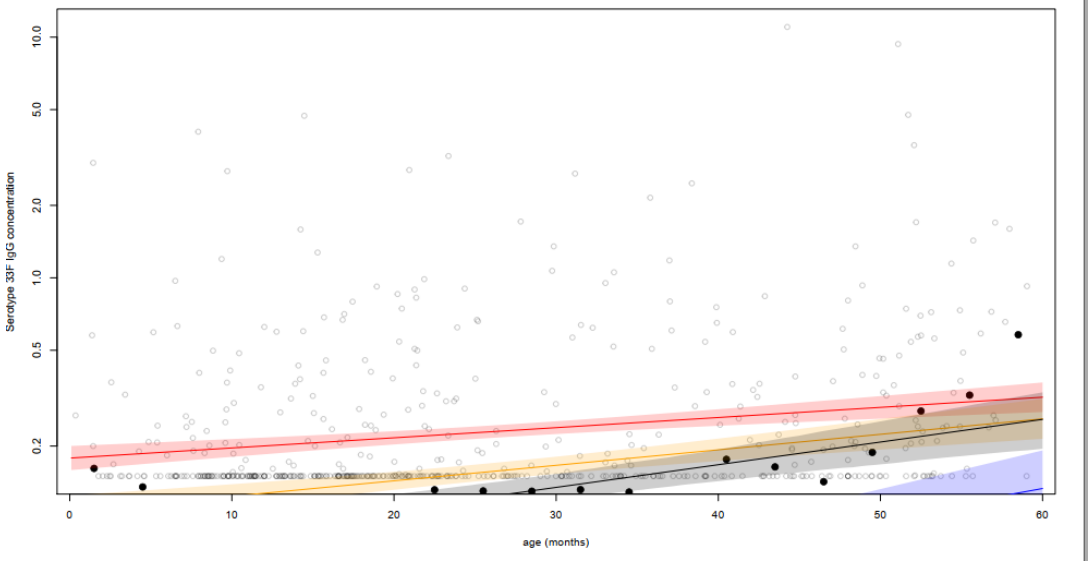
Serotype 18C graph



Serotype 19A graph



Serotype 19F graph



Serotype 33F graph

R PROGRAMS USED TO ANALYSE THE DATA

Programmer: Susanne Ntchaula Barnaba

Program: Biostatistics Master's Thesis – Longitudinal data

Supervisor: DR. Marc Henrion

PCVPA DATA ANALYSIS

```
rm(list=ls())

library(censReg)
library(VGAM)
library(boot)
```

```
# helper functions
geoMean<-function(x,na.rm=T){
  return(exp(sum(log(x[x > 0]), na.rm=na.rm) / length(x)))}

geoMeanCensoring<-function(x,left=0,right=Inf){
  # will return an error message if no censoring in the data
  require(censReg)
  return(exp(coef(censReg(log(x)~1,left=log(left),right=log(right)))["(Intercept)"]))}


# read data and reformat
dat<-read.csv("C:/Users/DELL/Desktop/Susanne Barnaba
BACKUP/Literatureforsusanne/Msc project/data/PCVPA_serology_reformatted.csv")
#View(dat)
#summary(dat)

levels(dat$age_cat_gmc)[levels(dat$age_cat_gmc)=="0-2m"]<-"0-02m"
levels(dat$age_cat_gmc)[levels(dat$age_cat_gmc)=="3-5m"]<-"03-05m"
levels(dat$age_cat_gmc)[levels(dat$age_cat_gmc)=="6-8m"]<-"06-08m"
levels(dat$age_cat_gmc)[levels(dat$age_cat_gmc)=="9-11m"]<-"09-11m"
dat$age_cat_gmc<-factor(as.character(dat$age_cat_gmc))


serotypes<-
c("1","3","4","5","6A","6B","7F","9V","14","18C","19A","19F","23F","33F")

for(st in serotypes){
  dat[,paste(sep="","res",st,"_num_ImpZero")]<-dat[,paste(sep="","res",st,"_num")]

dat[is.na(dat[,paste(sep="","res",st,"_num_ImpZero")]),paste(sep="","res",st,"_num_I
Zero")]<-0.01}


doAnalysis<-
function(dat,st,varNameOrig,varNameNum,varNameImpDL,varNameImpHalfDL,var
NameImpZero,outPrefix,DL=0.15){
  datGeoM<-
data.frame(ageCat=unique(dat$age_cat_gmc),geoM=NA,geoM_better=NA)
  for(i in 1:nrow(datGeoM)){
    datGeoM$geoM[i]<-
geoMean(dat[dat$age_cat_gmc==datGeoM$ageCat[i],varNameImpHalfDL]) # note
that these use the simple DL/2 imputation for calculation

if(sum(dat[dat$age_cat_gmc==datGeoM$ageCat[i],varNameOrig]=="<0.150")>0){
```

```
    datGeoM$geoM_better[i]<-
exp(coef(censReg(as.formula(paste(sep="","log(",varNameImpDL,")~1")),data=dat[d
at$age_cat_gmc==datGeoM$ageCat[i],],left=log(DL)))["(Intercept)"])
   }else{
    datGeoM$geoM_better[i]<-
exp(coef(lm(as.formula(paste(sep="","log(",varNameNum,")~1")),data=dat[dat$age_
cat_gmc==datGeoM$ageCat[i],]))["(Intercept)"])}  }
 levels(datGeoM$ageCat)[levels(datGeoM$ageCat)=="0-3m"]<-">00m"
 levels(datGeoM$ageCat)[levels(datGeoM$ageCat)==">3-6m"]<-">03m"
 levels(datGeoM$ageCat)[levels(datGeoM$ageCat)==">6-9m"]<-">06m"
 levels(datGeoM$ageCat)[levels(datGeoM$ageCat)==">9m"]<-">09m"

 datGeoM$ageCat<-factor(as.character(datGeoM$ageCat))

 datGeoM<-datGeoM[order(as.character(datGeoM$ageCat)),]
 datGeoM$ageCatNum<-seq(1.5,58.5,by=3)

 # remove NR values
 idx<-which(dat[,varNameOrig]=="NR" | dat[,varNameOrig]=="QNS" )
 if(length(idx)>0){datTmp<-dat[-idx,]}else{datTmp<-dat}
 datTmp<-dat

 # fit a linear regression model
 linearModDL<-lm(as.formula(paste(sep="","log(",varNameImpDL,") ~
ageMonths")), data=datTmp)
 linearModHalfDL<-lm(as.formula(paste(sep="","log(",varNameImpHalfDL,") ~
ageMonths")), data=datTmp)
 linearModZero<-lm(as.formula(paste(sep="","log(",varNameImpZero,") ~
ageMonths")), data=datTmp)
 #summary(linearMod)

 #fit a Tobit regression model
 modTobit<-vglm(as.formula(paste(sep="","log(",varNameImpDL,") ~
ageMonths")),
              tobit(Lower = log(0.15)),data=datTmp)
 #summary(modTobit)

 #Plot of linear regression with geometric means
 datNew<-data.frame(ageMonths=seq(0.1,60,length=1000))
 xx<-datNew$ageMonths

 cf<-coef(linearModDL,logSigma=F)
 yyDL<-cf[1]+cf[2]*xx

 predTmp<-predict(linearModDL, newdata = data.frame(ageMonths=xx), interval =
'confidence')
```

```
  yyDLLow <- predTmp[,2]
  yyDLHigh <- predTmp[,3]

  cf<-coef(linearModHalfDL,logSigma=F)
  yyHalfDL<-cf[1]+cf[2]*xx

  predTmp<-predict(linearModHalfDL, newdata = data.frame(ageMonths=xx),
interval = 'confidence')
  yyHalfDLLow <- predTmp[,2]
  yyHalfDLHigh <- predTmp[,3]

  cf<-coef(linearModZero,logSigma=F)
  yyZero<-cf[1]+cf[2]*xx

  predTmp<-predict(linearModZero, newdata = data.frame(ageMonths=xx), interval =
'confidence')
  yyZeroLow <- predTmp[,2]
  yyZeroHigh <- predTmp[,3]

  cf<-modTobit@coefficients
  yyTobit<-cf[1]+cf[3]*xx

  #bootstrap 95% CI for tobit regression model
  bxm<- function(formula, data, indices) {
    d <- data[indices,] # allows boot to select sample
    fit <- vglm(as.formula(paste(sep="","log(",varNameImpDL,") ~ ageMonths" )),
              tobit(Lower = log(0.15)),data=d)
    cf<-fit@coefficients
    xx<-seq(0.1,60,length=200)
    yy<-cf[1]+cf[3]*xx
    return(yy)  }

  # bootstrapping with 1000 replications
  Results <- boot(data=datTmp, statistic=bxm, R=1000, formula=
as.formula(paste(sep="","log(",varNameImpDL,") ~ageMonths")))

  # view results
  Results
  #plot(Results)

  # get 95% confidence interval
  yyLowTobit<-rep(NA,ncol(Results$t))
  yyHighTobit<-rep(NA,ncol(Results$t))
  for(i in 1:ncol(Results$t)){
    tmp<-boot.ci(Results, type="perc", index = i)$perc[4:5]
    yyLowTobit[i]<-tmp[1]
```

```
    yyHighTobit[i]<-tmp[2]  }


  # Create a data frame for summary statistics
  summary_table <- data.frame(
    Model = c("Linear (DL)", "Linear (HalfDL)", "Linear (Zero)", "Tobit"),
    Mean = c(mean(exp(yyDL)), mean(exp(yyHalfDL)), mean(exp(yyZero)),
mean(exp(yyTobit))),
    SD = c(sd(exp(yyDL)), sd(exp(yyHalfDL)), sd(exp(yyZero)), sd(exp(yyTobit))),
    Lower_CI = c(quantile(exp(yyDL), 0.025), quantile(exp(yyHalfDL), 0.025),
quantile(exp(yyZero), 0.025), quantile(exp(yyLowTobit), 0.025)),
    Upper_CI = c(quantile(exp(yyDL), 0.975), quantile(exp(yyHalfDL), 0.975),
quantile(exp(yyZero), 0.975), quantile(exp(yyHighTobit), 0.975))  )

  # Print the summary table
  cat("Summary Table for", st, "\n")
  print(summary_table)

  # Save the table to a CSV file
  write.csv(summary_table, file = paste0(outPrefix, "_summary_table.csv"),
row.names = FALSE)



  pdf(paste(sep="",outPrefix,"_fits.pdf"),width=16,height=9)
  # Calculate dynamic y-axis limits
  y_axis_min <- min(datGeoM$geoM_better, na.rm = TRUE)
  y_axis_max <- max(datGeoM$geoM_better, na.rm = TRUE)

  # Calculate dynamic x-axis limits
  x_axis_min <- min(datGeoM$ageCatNum, na.rm = TRUE)
  x_axis_max <- max(datGeoM$ageCatNum, na.rm = TRUE)

  # Plot the graph with dynamic axis limits
  plot(datGeoM$ageCatNum, datGeoM$geoM_better, log = "y", xlab = "age
(months)", ylab = paste(sep = "", "Serotype ", st, " IgG concentration"), pch = 20,
xlim = c(x_axis_min, x_axis_max), ylim = c(y_axis_min, y_axis_max), cex = 2)

  #plot(datGeoM$ageCatNum,datGeoM$geoM_better,log="y",xlab="age
(months)",ylab=paste(sep="","Serotype ",st," IgG
concentration"),pch=20,ylim=c(DL,max(dat[,varNameImpDL],na.rm=T)), cex=2)

points(dat$ageMonths,dat[,varNameImpDL],col=rgb(red=0,green=0,blue=0,alpha=50
,maxColorValue=255))

  lines(xx,exp(yyDL),col="red",lwd=1.5)
```

```
polygon(x=c(xx,xx[length(xx):1]),y=c(exp(yyDLLow),exp(yyDLHigh)[length(xx):1]
),col=rgb(red=255,green=0,blue=0,alpha=50,maxColorValue=255),border=NA)

  lines(xx,exp(yyHalfDL),col="orange",lwd=1.5)

polygon(x=c(xx,xx[length(xx):1]),y=c(exp(yyHalfDLLow),exp(yyHalfDLHigh)[lengt
h(xx):1]),col=rgb(red=255,green=165,blue=0,alpha=50,maxColorValue=255),border
=NA)

  lines(xx,exp(yyZero),col="blue",lwd=1.5)

polygon(x=c(xx,xx[length(xx):1]),y=c(exp(yyZeroLow),exp(yyZeroHigh)[length(xx):
1]),col=rgb(red=0,green=0,blue=255,alpha=50,maxColorValue=255),border=NA)

  lines(xx,exp(yyTobit),lwd=1.5,col="black")
  xx<-seq(0.1,60,length=200)

polygon(x=c(xx,xx[length(xx):1]),y=c(exp(yyLowTobit),exp(yyHighTobit)[length(xx
):1]),col=rgb(red=0,green=0,blue=0,alpha=50,maxColorValue=255),border=NA)

 # par(xpd=T)
  #lgd <- legend("topleft", legend = c("data", "geometric mean per age band"), pch =
c(1,20), bty="n",
col=c(rgb(red=0,green=0,blue=0,alpha=50,maxColorValue=255),"black"),pt.cex=c(1,
2),inset=c(0,-0.10))
  #legend(lgd$rect$left+lgd$rect$w, 5^(lgd$rect$top), legend = c("model fit", "95%
CI (model fit)"), lwd=c(2,2),
bty="n",col=c("red",rgb(red=255,green=190,blue=190,maxColorValue=255))) # can't
have transparent colors in the legend when in the margin; hence approximating the
transparent red and gray...
  #par(xpd=F)

  dev.off()

  return(list=ls())

#return(list=list(modLinear=modLinear,modTobit=modTobit,bootResults=Results))}
# running serotype 23F

serotypes<-
c("1","3","4","5","6A","6B","7F","9V","14","18C","19A","19F","23F","33F")
fits<-list()
for(st in serotypes){
  print(st)
```

```
  fits[[st]]<-
doAnalysis(dat=dat,st=st,varNameOrig=paste(sep="","res",st),varNameNum=paste(se
p="","res",st,"_num"),varNameImpDL=paste(sep="","res",st,"_num_ImpDL"),varNa
meImpHalfDL=paste(sep="","res",st,"_num_ImpHalfDL"),varNameImpZero
=paste(sep="","res",st,"_num_ImpZero"),outPrefix=paste(sep="","../output/serotype",
st),DL=0.15)}
```

DATA SIMULATION

```
rm(list=ls())
#MscProject starts here
# libraries (put any libraries that you need to load here)
library(gtools)
data(ELISA)
#print(ELISA)

#Changing the whole script into function
simData<-function(seed, np=100, ns=5, pExp=0.4, dl=0.15){
# set a random seed; this can be any number; it's just so you can reproduce it at a later
stage
#seed<-15*3+2019
#print(seed)
set.seed(seed)

#set overall parameters (add more as needed) (since the parameters are in simdata
function the rest have changed them to comment)
#np<-100 # number of participants for which samples are analysed
# ns<-5 # number of samples per participant
# pExp<-0.4 # probability for a randomly selected participant to have been exposed to
the pathogen (I assume the 3 antibodies are related to this pathogen)
# dl<-0.15 # detection limit (assume same one for all markers)

# set up data frames
simDatPatients<-data.frame(
  patientID=paste(sep="","p",1:np),
  exposed=NA,
  exposedDate=NA,
  gender=NA,
  age=NA,
  random1=NA,
  random2=NA,
  random3=NA)
#print(simDatPatients)


tmp<-expand.grid(1:ns,1:np)
```

```
simDatSamples<-data.frame(
  patientID=paste(sep="","p",tmp[,2]),
  sampleID=paste(sep="","p",tmp[,2],"_s",tmp[,1]),
  visit=tmp[,1],
  exposed=NA,
  exposedDate=NA,
  gender=NA,
  random1=NA,
  random2=NA,
  random3=NA,
  age=NA,
  ac1=NA, # actual average concentration for antibies 1-3
  ac2=NA,
  ac3=NA,
  c1=NA, # actual concentrations for antibodies 1-3
  c2=NA,
  c3=NA,
  od1=NA, # perfectly backtransformed optical densities
  od2=NA,
  od3=NA,
  mod1=NA, # measured optical densities for antibodies 1-3 (od1, od2, od3 with
noise)
  mod2=NA,
  mod3=NA,
  mc1=NA, # measured concentrations for antibodies 1-3; this is calculated, not
simulated from the optical densities; this is what researchers would do to convert OD
into concentrations; you will need to choose some of your samples to be standards of
known cocentration.
  mc2=NA,
  mc3=NA
)

rm(tmp)

#print(simDatSamples)

#simulate gender
simDatPatients$gender<-
factor(sample(x=c("male","female"),size=nrow(simDatPatients),replace=T))
simDatSamples$gender<-
simDatPatients$gender[match(simDatSamples$patientID,simDatPatients$patientID)]



#simulate age
simDatPatients$age<-rgamma(nrow(simDatPatients),shape=8,rate=1)
```

```
simDatSamples$age<-
simDatPatients$age[match(simDatSamples$patientID,simDatPatients$patientID)]+si
mDatSamples$visit


# simulate exposure status - have participants been exposed to the pathogen (i.e.
should have had an immune response that led to antibodies being present)
simDatPatients$exposed<-sample(x=0:1,size=np,replace=T,prob=c(1-pExp,pExp))
simDatPatients$exposedDate<-
ifelse(simDatPatients$exposed==0,0,sample(x=1:ns,size=nrow(simDatPatients),repla
ce=T))
simDatSamples$exposedDate<-
simDatPatients$exposedDate[match(simDatSamples$patientID,simDatPatients$patie
ntID)]
for(i in 1:np){
  idxS<-which(simDatSamples$patientID==paste(sep="","p",i))
  idxP<-which(simDatPatients$patientID==paste(sep="","p",i))
  simDatSamples$exposed[idxS]<-ifelse(simDatPatients$exposed[idxP]==1 &
simDatSamples$visit[idxS]>=simDatPatients$exposedDate[idxP],1,0)}

# simulate random patient effect
simDatPatients$random1<-rnorm(np,mean=0,sd=1)
simDatPatients$random2<-rnorm(np,mean=0,sd=1)
simDatPatients$random3<-rnorm(np,mean=0,sd=1)
simDatSamples$random1<-
simDatPatients$random1[match(simDatSamples$patientID,simDatPatients$patientID
)]
simDatSamples$random2<-
simDatPatients$random2[match(simDatSamples$patientID,simDatPatients$patientID
)]
simDatSamples$random3<-
simDatPatients$random3[match(simDatSamples$patientID,simDatPatients$patientID
)]

# simulate concentration data for exposed and unexposed
logistic4Param<-function(x,a=1,b=1,c=1,d=5){
  res<-d+(a-d)/(1+(x/c)^b)
  return(res)}
# for c1

maleOffset<-2.5
changePerYear<-0.1
exposureIncrease<-1

simDatSamples$ac1<-
ifelse(simDatSamples$gender=="male",maleOffset,0)+simDatSamples$age*changeP
```

```
erYear+simDatSamples$exposed*exposureIncrease+rexp(nrow(simDatSamples),rate
=0.5)+simDatSamples$random1
simDatSamples$ac1[simDatSamples$ac1<0]<-0

simDatSamples$c1<-(simDatSamples$ac1+rexp(nrow(simDatSamples),rate=0.75))/1
simDatSamples$c1[simDatSamples$c1<0]<-0

#for c2

maleOffset<-1.7
changePerYear<-0.5
exposureIncrease<-2.5

simDatSamples$ac2<-
ifelse(simDatSamples$gender=="male",maleOffset,0)+simDatSamples$age*changeP
erYear+simDatSamples$exposed*exposureIncrease+rexp(nrow(simDatSamples),
rate=0.3)+simDatSamples$random2
simDatSamples$ac2[simDatSamples$ac2<0]<-0

simDatSamples$c2<-(simDatSamples$ac2+rexp(nrow(simDatSamples),rate=1))/2
simDatSamples$c2[simDatSamples$c2<0]<-0

#for c3

maleOffset<-0.6
changePerYear<-0.25
exposureIncrease<-5

simDatSamples$ac3<-
ifelse(simDatSamples$gender=="male",maleOffset,0)+simDatSamples$age*changeP
erYear+simDatSamples$exposed*exposureIncrease+rexp(nrow(simDatSamples),rate
=0.75)+simDatSamples$random3
simDatSamples$ac3[simDatSamples$ac3<0]<-0

simDatSamples$c3<-(simDatSamples$ac3+rexp(nrow(simDatSamples),rate=0.4))/2
simDatSamples$c3[simDatSamples$c1<0]<-0

#print(simDatSamples)
# simulate optical densities from the concentrations

inverse.logistic4Param<-function(x,a,b,c,d){
  res<-c*((a-d)/(x-d) - 1)^(1/b)
  return(res)}


# add standards of known concentration (adds rows)
```

```
stds<-data.frame(
  patientID=paste(sep="","std",1:5),
  sampleID=paste(sep="","std",1:5),
  visit=NA,
  exposed=NA,
  exposedDate=NA,
  gender=NA,
  random1=NA,
  random2=NA,
  random3=NA,
  age=NA,
  ac1=NA,
  c1=c(5,12.5,18,20,30),
  ac2=NA,
  c2=c(6,9,12,15,27),
  ac3=NA,
  c3=c(4,8,12,20,35),
  od1=NA,
  od2=NA,
  od3=NA,
  mod1=NA,
  mod2=NA,
  mod3=NA,
  mc1=NA,
  mc2=NA,
  mc3=NA)
simDatSamples<-rbind(simDatSamples,stds)

# going from concentration to OD

# For od1 and mod1
# x<-seq(0,40,length=100)
# y<-inverse.logistic4Param(x,a=2,b=1,c=0.5,d=40) # default values
# y2<-inverse.logistic4Param(x,a=0.1,b=1,c=0.5,d=40) # changing a
# y3<-inverse.logistic4Param(x,a=1,b=2,c=0.5,d=40) # changing b
# y4<-inverse.logistic4Param(x,a=1,b=1,c=2,d=40) # changing c
# yFinal<-inverse.logistic4Param(x,a=0.2,b=1.5,c=1,d=40) # changing d
# plot(x,y,type="l",ylim=c(0,5)) # specify ylim=c(0,20) if you want to restrict the
range shown on the y axis to [0,20]
# abline(h=dl,lty=2)
# lines(x,y2,col="red")
# lines(x,y3,col="blue")
# lines(x,y4,col="green")
# lines(x,yFinal,col="orange",lwd=2)
```

```
simDatSamples$od1<-inverse.logistic4Param(simDatSamples$c1,
a=0.2,b=1.5,c=1,d=40)

simDatSamples$mod1<-simDatSamples$od1+rnorm(nrow(simDatSamples),sd=0.01)
simDatSamples$mod1[simDatSamples$mod1<0]<-0

#sum(is.na(simDatSamples$od1))


#For od2 and mod2

# For od2 and mod2
# x<-seq(0,35,length=100)
# y<-inverse.logistic4Param(x,a=2,b=1,c=0.5,d=35) # default values
# y2<-inverse.logistic4Param(x,a=0.1,b=1,c=0.5,d=35) # changing a
# y3<-inverse.logistic4Param(x,a=1,b=2,c=0.5,d=35) # changing b
# y4<-inverse.logistic4Param(x,a=1,b=1,c=2,d=35) # changing c
# yFinal<-inverse.logistic4Param(x,a=0,b=1,c=1.7,d=35) # changing d
# plot(x,y,type="l",ylim=c(0,5)) # specify ylim=c(0,20) if you want to restrict the
range shown on the y axis to [0,20]
# abline(h=dl,lty=2)
# lines(x,y2,col="red")
# lines(x,y3,col="blue")
# lines(x,y4,col="green")
# lines(x,yFinal,col="orange",lwd=2)

simDatSamples$od2<-inverse.logistic4Param(simDatSamples$c2,
a=0,b=1,c=1.7,d=35)

simDatSamples$mod2<-
simDatSamples$od2+rnorm(nrow(simDatSamples),sd=0.005)
simDatSamples$mod2[simDatSamples$mod2<0]<-0

#simDatSamples$mod2<-
rlnorm(nrow(simDatSamples),mean=simDatSamples$od2,sd=0.5)
#For od3 and mod3
# For od3 and mod3
# x<-seq(0,50,length=100)
# y<-inverse.logistic4Param(x,a=2,b=1,c=0.5,d=50) # default values
# y2<-inverse.logistic4Param(x,a=0.1,b=1,c=0.5,d=50) # changing a
# y3<-inverse.logistic4Param(x,a=1,b=2,c=0.5,d=50) # changing b
# y4<-inverse.logistic4Param(x,a=1,b=1,c=2,d=50) # changing c
# yFinal<-inverse.logistic4Param(x,a=0.2,b=1.5,c=1.5,d=50) # changing d
# plot(x,y,type="l",ylim=c(0,5)) # specify ylim=c(0,20) if you want to restrict the
range shown on the y axis to [0,20]
# abline(h=dl,lty=2)
```

```
# lines(x,y2,col="red")
# lines(x,y3,col="blue")
# lines(x,y4,col="green")
# lines(x,yFinal,col="orange",lwd=2)

simDatSamples$od3<-inverse.logistic4Param(simDatSamples$c3,
a=0.2,b=1.5,c=1.25,d=50)

simDatSamples$mod3<-simDatSamples$od3+rnorm(nrow(simDatSamples),sd=0.02)
simDatSamples$mod3[simDatSamples$mod3<0]<-0

#sum(is.na(simDatSamples$mod3))
#hist(simDatSamples$mod1,breaks=50)

#simDatSamples$mod3<-
rlnorm(nrow(simDatSamples),mean=simDatSamples$od3,sd=0.5)

#print(simDatSamples)

# derive (or set) limits of detection and quantification; replace all ODs below the
LOQs by "< LOQ" or some impossble values such as "-9"

simDatSamples$mod1[simDatSamples$mod1<dl]<-NA
#sum(is.na(simDatSamples$mod1))/nrow(simDatSamples) # check the proportion of
missing values; aim for ~4-6% for one of mod1, mod2, mod3, ~10-15% for another
and ~25-30% for another

simDatSamples$mod2[simDatSamples$mod2<dl]<-NA
#sum(is.na(simDatSamples$mod2))/nrow(simDatSamples) # check the proportion of
missing values; aim for ~4-6% for one of mod1, mod2, mod3, ~10-15% for another
and ~25-30% for another

simDatSamples$mod3[simDatSamples$mod3<dl]<-NA
#sum(is.na(simDatSamples$mod3))/nrow(simDatSamples) # check the proportion of
missing values; aim for ~4-6% for one of mod1, mod2, mod3, ~10-15% for another
and ~25-30% for another

#print(simDatSamples)

# going from od to measured concentration: (i) use least squares to estimate values for
a, b, c, d, (ii) use the estimated parameters to obtain mc1, mc2, mc3

idxStds<-grep(simDatSamples$patientID,pattern="std")

ssFun<-function(pars,c,mod){
  res<-sum( (c - logistic4Param(mod,a=pars[1],b=pars[2],c=pars[3],d=pars[4]))^2 )
```

```
 res}

parsC1<-
optim(par=c(0,1,1,40),fn=ssFun,c=simDatSamples$c1[idxStds],mod=simDatSamples
$mod1[idxStds])
simDatSamples$mc1<-logistic4Param(simDatSamples$mod1,
a=parsC1$par[1],b=parsC1$par[2],c=parsC1$par[3],d=parsC1$par[4])
dlMc1<-logistic4Param(dl,
a=parsC1$par[1],b=parsC1$par[2],c=parsC1$par[3],d=parsC1$par[4])

parsC2<-
optim(par=c(0,1,1,35),fn=ssFun,c=simDatSamples$c2[idxStds],mod=simDatSamples
$mod2[idxStds])
simDatSamples$mc2<-logistic4Param(simDatSamples$mod2,
a=parsC2$par[1],b=parsC2$par[2],c=parsC2$par[3],d=parsC2$par[4])
dlMc2<-logistic4Param(dl,
a=parsC2$par[1],b=parsC2$par[2],c=parsC2$par[3],d=parsC2$par[4])
parsC3<-
optim(par=c(0,1,1,50),fn=ssFun,c=simDatSamples$c3[idxStds],mod=simDatSamples
$mod3[idxStds])
simDatSamples$mc3<-logistic4Param(simDatSamples$mod3,
a=parsC3$par[1],b=parsC3$par[2],c=parsC3$par[3],d=parsC3$par[4])
dlMc3<-logistic4Param(dl,
a=parsC3$par[1],b=parsC3$par[2],c=parsC3$par[3],d=parsC3$par[4])


# remove the standards
simDatSamples<-simDatSamples[-idxStds,]

# preview the data
#print(simDatSamples)

# save the data
#save(list=c("simDatSamples","dlMc1","dlMc2","dlMc3"),file="/Users/HP/Desktop/
Literatureforsusanne/Msc project/data/simDat20201022.RData")
# Calculate the proportion of missing data for mod1
prop_missing_mod1 <- sum(is.na(simDatSamples$mod1))/nrow(simDatSamples)

# Calculate the proportion of missing data for mod2
prop_missing_mod2 <- sum(is.na(simDatSamples$mod2))/nrow(simDatSamples)

# Calculate the proportion of missing data for mod3
prop_missing_mod3 <- sum(is.na(simDatSamples$mod3))/nrow(simDatSamples)

# Print the results
cat("Proportion of missing data for mod1:", prop_missing_mod1, "\n")
```

```
cat("Proportion of missing data for mod2:", prop_missing_mod2, "\n")
cat("Proportion of missing data for mod3:", prop_missing_mod3, "\n")
 print(results)
#return(simDatSamples)

return(list(simDatSamples=simDatSamples,
          dlMc1=dlMc1,
          dlMc2=dlMc2,dlMc3=dlMc3))}
```

DATA SIMULATION ANALYSIS

```
rm(list=ls())


source("c:/Users/DELL/Desktop/Susanne Barnaba
BACKUP/Literatureforsusanne/Msc
project/Scripts/Sue_DataSimulation_M_MHFinal.R")
results<-list()
B<-1e3

# helper functions
geoMean<-function(x,na.rm=T){
  return(exp(sum(log(x[x > 0]), na.rm=na.rm) / length(x)))}

geoMeanCensoring<-function(x,left=0,right=Inf){
  # will return an error message if no censoring in the data
  require(censReg)
  return(exp(coef(censReg(log(x)~1,left=log(left),right=log(right)))["(Intercept)"]))}


library(lme4)
#library(lmerTest)
library(censReg)
library(VGAM)
library(boot)
library(GGally)
library(tidyverse)
library(plm)
library(Matrix)

singularModels<-
  data.frame(j=integer(0),model=character(0))

for(j in 1:B){
#print(j)
simDat<-simData(np=100, ns=5, pExp=0.4, dl=0.15, seed=j)
simDatSamples<-simDat$simDatSamples
```

```
dlMc1<-simDat$dlMc1
dlMc2<-simDat$dlMc2
dlMc3<-simDat$dlMc3



 idxMc1<-which(is.na(simDatSamples$mc1))
 idxMc2<-which(is.na(simDatSamples$mc2))
 idxMc3<-which(is.na(simDatSamples$mc3))

 simDatSamples$mc1[idxMc1]<-dlMc1
 simDatSamples$mc2[idxMc2]<-dlMc2
 simDatSamples$mc3[idxMc3]<-dlMc3

 #fit a linear longitudinal mixed model to all the three biomakers to DL

 fm1_DL<-lmer(mc1~age+exposed+gender+ (1|patientID), data = simDatSamples)
 if(isSingular(fm1_DL)){singularModels<-rbind(singularModels,
                                 data.frame(j=j,model="fm1_DL"))}


  #summary(fm1_DL)


 fm2_DL<-lmer(mc2~age+exposed+gender+ (1|patientID), data = simDatSamples)
 if(isSingular(fm2_DL)){singularModels<-rbind(singularModels,
                                  data.frame(j=j,model="fm2_DL"))}



 #summary(fm2_DL)

 fm3_DL<-lmer(mc3~age+exposed+ gender + (1|patientID), data = simDatSamples)
 if(isSingular(fm3_DL)){singularModels<-rbind(singularModels,
                                  data.frame(j=j,model="fm3_DL"))}


 #summary(fm3_DL)

 simDatSamples$mc1[idxMc1]<-dlMc1/2
 simDatSamples$mc2[idxMc2]<-dlMc2/2
 simDatSamples$mc3[idxMc3]<-dlMc3/2

 #fit a linear longitudinal mixed model to all the three biomakers to DL/2
```

```
  fm1_halfDL<-lmer(mc1~age+exposed +gender+ (1|patientID), data =
simDatSamples)
  if(isSingular(fm1_halfDL)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm1_halfDL"))}



  #summary(fm1_halfDL)


  fm2_halfDL<-lmer(mc2~age+exposed+gender +(1|patientID), data =
simDatSamples)
  if(isSingular(fm2_halfDL)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm2_halfDL"))}



  #summary(fm2_halfDL)

  fm3_halfDL<-lmer(mc3~age+exposed+gender+(1|patientID), data =
simDatSamples)
  if(isSingular(fm3_halfDL)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm3_halfDL"))}

  #summary(fm3_halfDL)


  simDatSamples$mc1[idxMc1]<-0
  simDatSamples$mc2[idxMc2]<-0
  simDatSamples$mc3[idxMc3]<-0

  #fit a linear longitudinal mixed model to all the three biomakers to zero

  fm1_zero<-lmer(mc1~age+exposed +gender+ (1|patientID), data = simDatSamples)
  if(isSingular(fm1_zero)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm1_zero"))}

  #summary(fm1_zero)


  fm2_zero<-lmer(mc2~age+exposed+gender+(1|patientID), data = simDatSamples)
  if(isSingular(fm2_zero)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm2_zero"))}

  #summary(fm2_zero)
```

```
fm3_zero<-lmer(mc3~age+exposed+gender+(1|patientID), data = simDatSamples)
if(isSingular(fm3_zero)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm3_zero"))}

#summary(fm3_zero)

#fit a longitudinal mixed model to all the three biomakers to NA

simDatSamples$mc1[idxMc1]<-NA
simDatSamples$mc2[idxMc2]<-NA
simDatSamples$mc3[idxMc3]<-NA


fm1_NA<-lmer(mc1~age+exposed +gender+ (1|patientID), data = simDatSamples)
if(isSingular(fm1_NA)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm1_NA"))}

#summary(fm1_NA)


fm2_NA<-lmer(mc2~age+exposed+gender+(1|patientID), data = simDatSamples)
if(isSingular(fm2_NA)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm2_NA"))}

#summary(fm2_NA)

fm3_NA<-lmer(mc3~age+exposed+gender+(1|patientID), data = simDatSamples)
if(isSingular(fm3_NA)){singularModels<-rbind(singularModels,
                                data.frame(j=j,model="fm3_NA"))}

#summary(fm3_NA)
 #fit a censored longitudinal mixed model to all the three biomakers

simDatSamples$mc1[idxMc1]<-dlMc1
simDatSamples$mc2[idxMc2]<-dlMc2
simDatSamples$mc3[idxMc3]<-dlMc3



simDatSamples<-pdata.frame(simDatSamples, c("patientID", "visit"))
 fm1_C<-censReg(mc1~age+exposed +gender, data = simDatSamples, method
="BHHH", left=dlMc1)
 #isSingular(fm1_C)
#summary(fm1_C)
```

```
 fm2_C<-censReg(mc2~age+exposed+gender, data = simDatSamples, method
="BHHH", left=dlMc2)
 #isSingular(fm2_C)
 #summary(fm2_C)

 fm3_C<-censReg(mc3~age+exposed+gender, data = simDatSamples, method
="BHHH", left=dlMc3)
 #isSingular(fm3_C)
#summary(fm3_C)



resultsTable<-
 data.frame(intercept=numeric(15),interceptSE=numeric(15),
          age=numeric(15),ageSE=numeric(15),
          exposed=numeric(15), exposedSE=numeric(15),
          gender=numeric(15), genderSE=numeric(15))
rownames(resultsTable)<-c(paste(sep="","DL_",1:3),
                    paste(sep="","halfDL_",1:3),
                    paste(sep="","Zero_",1:3),
                    paste(sep="","NA_",1:3),
                    paste(sep="","C_",1:3))


resultsTable["DL_1",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm1_DL)$coefficients[,1:2])))
resultsTable["DL_2",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm2_DL)$coefficients[,1:2])))
resultsTable["DL_3",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm3_DL)$coefficients[,1:2])))
resultsTable["halfDL_1",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm1_halfDL)$coefficients[,1:2
])))
resultsTable["halfDL_2",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm2_halfDL)$coefficients[,1:2
])))
resultsTable["halfDL_3",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm3_halfDL)$coefficients[,1:2
])))
resultsTable["Zero_1",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm1_zero)$coefficients[,1:2]))
)
resultsTable["Zero_2",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm2_zero)$coefficients[,1:2]))
resultsTable["Zero_3",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm3_zero)$coefficients[,1:2]))
```

```
resultsTable["NA_1",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm1_NA)$coefficients[,1:2])))
resultsTable["NA_2",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm2_NA)$coefficients[,1:2])))
resultsTable["NA_3",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm3_NA)$coefficients[,1:2])))
resultsTable["C_1",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm1_C)$estimate[1:4,1:2])))
resultsTable["C_2",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm2_C)$estimate[1:4,1:2])))
resultsTable["C_3",]<-
as.vector(matrix(byrow=T,nrow=2,as.vector(summary(fm3_C)$estimate[1:4,1:2])))


results[[j]]<-resultsTable


}resultsMean <- matrix(NA, nrow=nrow(results[[1]]), ncol=ncol(results[[1]]))
rownames(resultsMean)<-rownames(results[[1]])
colnames(resultsMean)<-colnames(results[[1]])
resultsMedian <- resultsMean
resultsquantiles25<-resultsMean
resultsquantiles75<-resultsMean
resultsStandarddeviation<-resultsMean
resultsquantiles2.5<-resultsMean
resultsquantiles97.5<-resultsMean

for(l in 1:nrow(results[[1]])){
  for(m in 1:ncol(results[[1]])){
    resultsMean[l, m] <- mean(unlist(lapply(results, `[`, i =l, j = m)))
    resultsMedian[l, m] <- median(unlist(lapply(results, `[`, i =l, j = m)))
    resultsquantiles25[l, m]<-quantile(probs=0.25,unlist(lapply(results, `[`, i =l, j =
m)))
    resultsquantiles75[l, m]<-quantile(probs=0.75,unlist(lapply(results, `[`, i =l, j =
m)))
    resultsStandarddeviation[l, m] <- sd(unlist(lapply(results, `[`, i =l, j = m)))
    resultsquantiles2.5[l, m]<-quantile(probs=0.025,unlist(lapply(results, `[`, i =l, j =
m)))
    resultsquantiles97.5[l, m]<-quantile(probs=0.975,unlist(lapply(results, `[`, i =l, j =
m)))}}

# resultsTable
# To Create a forest plot

library(ggplot2)
library(gridExtra)
```

```
#with intercept Mc1, Mc2, Mc3

#Mc1
modNames<-c("DL","C","Zero","NA","halfDL")
markerName<-"1"
parName<-"intercept"
trueMc1Intercept<-3.33

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]) )


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc1Intercept<- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc1Intercept,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Intercept", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()

#This section will be removed
  #df<-data.frame(
  #modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),

  #df$modNames2<-relevel(x=factor(df$modNames),ref = "truth"),
  #mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  #lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  #upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

  #)
```

```
#png("myGraphmc1I.png",width=16,height=9,units="cm",res=300)
 #df$modNames<-factor(df$modNames, levels = df$modNames)
 #ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
 #                xmax=upper))+
 #geom_point()+
 #geom_errorbarh(height=.1)+
 #geom_vline(xintercept = trueMc1Intercept,lty=2,lwd=1.25, col="darkgrey") +
 #scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc1)+
 #labs(title="Effect of Intercept", x="Coefficient values",y = "")+
 #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
 #theme_classic()
#dev.off()

#Mc2

modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"2"
parName<-"intercept"
trueMc2Intercept<-2.17

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
 modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
 mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
 lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
 upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc2Intercept<- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
 geom_point()+
 geom_errorbarh(height=.1)+
 geom_vline(xintercept = trueMc2Intercept,lty=2,lwd=1.25, col="darkgrey") +
 labs(title="Intercept", x="Coefficient values",y = "")+
 geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
 theme_classic()
```

```
#df<-data.frame(
  #modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),


  #mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  #lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  #upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

  #)

#png("myGraphmc2I.png",width=16,height=9,units="cm",res=300)
#Mc2intercept = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
# geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc2Intercept,lty=2,lwd=1.25, col="darkgrey") +
 # scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc2)+
# labs(title="Effect of Intercept", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()

#Mc3
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"3"
parName<-"intercept"
trueMc3Intercept<-1.92

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
```

```
Mc3Intercept<- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc3Intercept,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Intercept", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()

# Below it has to be removed except


#df<-data.frame(
# modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),
#
#
# mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
# lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
# upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

#)


#png("myGraphmc3I.png",width=16,height=9,units="cm",res=300)
#Mc3intercept = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
# geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc3Intercept,lty=2,lwd=1.25, col="darkgrey") +
 # scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc3)+
# labs(title="Effect of Intercept", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()

#with gender Mc1, Mc2, Mc3

#Mc1
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"1"
parName<-"gender"
trueMc1gender<-2.5

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
```

```
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc1gender<- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc1gender,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Sex", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()




# df<-data.frame(
#   modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),

       #df$modNames2<-relevel(x=factor(df$modNames),ref = "truth"),
#   mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
#   lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
#   upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])
#
# )


#png("myGraphmc1G.png",width=16,height=9,units="cm",res=300)
#Mc1gender = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                    xmax=upper))+
#     geom_point()+
#     geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc1gender,lty=2,lwd=1.25, col="darkgrey") +
      #scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc1)+
#     labs(title="Effect of Sex", x="Coefficient values",y = "")
```

```
      #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
      #theme_classic()
#dev.off()

#Mc2
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"2"
parName<-"gender"
trueMc2gender<-1.7

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc2gender<- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc2gender,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Sex", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()


#df<-data.frame(
#   modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),


#   mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
#   lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
#   upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])
#
# )
```

```
#png("myGraphmc2G.png",width=16,height=9,units="cm",res=300)
# Mc2gender = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
#  geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc2gender,lty=2,lwd=1.25, col="darkgrey") +
  # scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc2)+
# labs(title="Effect of Sex", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()

#Mc3
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"3"
parName<-"gender"
trueMc3gender<-0.6

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc3gender<- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc3gender,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Sex", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()


#df<-data.frame(
```

```
#   modNames=c("DL","HalfDL","Zero","Complete Observation","Censored"),
#
#   mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
#   lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
#   upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

# )

#png("myGraphmc3G.png",width=16,height=9,units="cm",res=300)
# Mc3gender = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
# geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc3gender,lty=2,lwd=1.25, col="darkgrey") +
  # scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc3)+
# labs(title="Effect of Sex", x="Cofficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()

#with age Mc1, Mc2, Mc3

#Mc1
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"1"
parName<-"age"
trueMc1age<-0.1

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc1age<-ggplot(data=df, aes(y=modNames, x=mean, xmin=lower, xmax=upper))+
```

```
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc1age,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Age", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()



#df<-data.frame(
# modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),

# mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
# lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
# upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])
#
#)
#png("myGraphmc1A.png",width=16,height=9,units="cm",res=300)
#Mc1age = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
# geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc1age,lty=2,lwd=1.25, col="darkgrey") +
  #scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc1)+
# labs(title="Effect of Age", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()

#Mc2
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"2"
parName<-"age"
trueMc2age<-0.5

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs
```

```
resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc2age <- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower, xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc2age,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Age", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()


#df<-data.frame(
  #modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),


  #mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  #lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  #upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

  #)

#png("myGraphmc2A.png",width=16,height=9,units="cm",res=300)
#Mc2age = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
# geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc2age,lty=2,lwd=1.25, col="darkgrey") +
 # scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc2)+
# labs(title="Effect of Age", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()

#Mc3
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"3"
parName<-"age"
trueMc3age<-0.25

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
```

```
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc3age <- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower, xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc3age,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Age", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()




#df<-data.frame(
# modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),


# mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
# lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
# upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

#)

#png("myGraphmc3A.png",width=16,height=9,units="cm",res=300)
#Mc3age = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
# xmax=upper))+
  #geom_point()+
#geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc3age,lty=2,lwd=1.25, col="darkgrey") +
 # scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc3)+
#labs(title="Effect of Age", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
```

```
#dev.off()

#with Exposed Mc1, Mc2, Mc3

#Mc1
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"1"
parName<-"exposed"
trueMc1exposed<-1

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])
)


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc1exposed <- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc1exposed,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Exposure", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()



#df<-data.frame(
# modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),


# mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
# lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
#upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])
```

```
#)

#png("myGraphmc1E.png",width=16,height=9,units="cm",res=300)
#Mc1exposed = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
# xmax=upper))+
#geom_point()+
#geom_errorbarh(height=.1)+
#geom_vline(xintercept = trueMc1exposed,lty=2,lwd=1.25, col="darkgrey") +
  #scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc1)+
# labs(title="Effect of Exposure", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()


#Mc2
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"2"
parName<-"exposed"
trueMc2exposed<-2.5

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable


# Use the data frame to create the graph
df <- coeffs
Mc2exposed <- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc2exposed,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Exposure", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
```

```
  theme_classic()


#df<-data.frame(
# modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),


# mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
# lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
# upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

#)

#png("myGraphMC2E.png",width=16,height=9,units="cm",res=300)
#Mc2exposed = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
# geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc2exposed,lty=2,lwd=1.25, col="darkgrey") +
  #scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc1)+
# labs(title="Effect of Exposure", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()
#dev.off()

#Mc3
modNames<-c("DL","halfDL","Zero","NA","C")
markerName<-"3"
parName<-"exposed"
trueMc3exposed<-5

# Create a data frame with the mean, lower, and upper values of the coefficients
coeffs <- data.frame(
  modNames=c("Detection Limit","Half Detection Limit","Zero","Complete
Observations","Censored"),
  mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
  lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
  upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName]))


# Use the data frame to create the table for intercept
resultsTable <- coeffs

resultsTable
```

```
# Use the data frame to create the graph
df <- coeffs
Mc3exposed <- ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
xmax=upper))+
  geom_point()+
  geom_errorbarh(height=.1)+
  geom_vline(xintercept = trueMc3exposed,lty=2,lwd=1.25, col="darkgrey") +
  labs(title="Effect of Exposure", x="Coefficient values",y = "")+
  geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  theme_classic()



#df<-data.frame(
# modNames=c("DL","Half DL","Zero","Complete Observations","Censored"),


# mean=c(resultsMean[paste(sep="_",modNames,markerName),parName]),
# lower=c(resultsquantiles2.5[paste(sep="_",modNames,markerName),parName]),
# upper=c(resultsquantiles97.5[paste(sep="_",modNames,markerName),parName])

#)

#png("myGraphmc3E.png",width=16,height=9,units="cm",res=300)
#Mc3exposed = ggplot(data=df, aes(y=modNames, x=mean, xmin=lower,
#                 xmax=upper))+
# geom_point()+
# geom_errorbarh(height=.1)+
# geom_vline(xintercept = trueMc3exposed,lty=2,lwd=1.25, col="darkgrey") +
  # scale_y_discrete(name ="", breaks=1:nrow(df),labels=df$Mc3)+
# labs(title="Effect of Exposure", x="Coefficient values",y = "")
  #geom_vline(xintercept =0, color = "black", linetype = "dashed", alpha= .5)+
  #theme_classic()

#dev.off()

grid.arrange(Mc1Intercept, Mc1gender,Mc1age, Mc1exposed, nrow = 2, ncol =2)

grid.arrange(Mc2Intercept, Mc2gender,Mc2age, Mc2exposed, nrow = 2, ncol =2)

grid.arrange(Mc3Intercept, Mc3gender,Mc3age, Mc3exposed, nrow = 2, ncol =2)
```