Bayesian and Bootstrap Methods for Estimating Population Attributable Risk: Development of an R Package for Epidemiological Inference

University of Turku
Department of Computing
Master of Science Thesis
Data Analytics
June 2025
Peppi-Lotta Saari

Supervisors:
Leo Lahti
Ville Laitinen

UNIVERSITY OF TURKU
Department of Computing

Peppi-Lotta Saari: Bayesian and Bootstrap Methods for Estimating Population Attributable Risk: Development of an R Package for Epidemiological Inference

Master of Science Thesis, 61 p.
Data Analytics
June 2025

This thesis explores the calculation of Population Attributable Risk (PAR) and Population Attributable Fraction (PAF), focusing on constructing confidence intervals using both Bayesian and Bootstrap methods. The result is an R package designed for usability, enabling users to compute PAR and PAF from 2x2 contingency tables and construct confidence intervals using either a fully Bayesian approach, as described by Pirikahu et al., or a Bootstrap method.
Comparative evaluations show that while Bootstrap generally produces shorter intervals, its performance diminishes in cases of low or high exposure rates or small sample sizes—conditions where the Bayesian method demonstrates more consistent coverage. The package also includes functionality for adjusted PAR calculations, offering a pathway toward more complex exposure scenarios.
This thesis emphasizes the importance of transparency, reproducibility, and methodological choice in statistical software development. It also highlights the challenges posed by restricted access to scientific literature and limited ongoing maintenance of statistical packages, advocating for more open and sustainable approaches in research tool development.

Keywords: Population attributable risk (PAR), Population attributable fraction (PAF), Attributable fraction (AF), Confidence interval, Bayesian inference, Bootstrap, R, Programming, 2x2 contingency table, Epidemiology, Statistical modeling, Statistical programming

TURUN YLIOPISTO
Tietotekniikan laitos

Peppi-Lotta Saari: Bayesian and Bootstrap Methods for Estimating Population Attributable Risk: Development of an R Package for Epidemiological Inference

Pro gradu -tutkielma, 61 s.
Data Analytics
Kesäkuu 2025

Tässä opinnäytetyössä tarkastellaan väestön riskin (Population Attributable Risk, PAR) ja riskiosuuden (Population Attributable Fraction, PAF) laskemista. Pääpaino on luottamusvälien muodostamisessa käyttäen sekä Bayesilaista että Bootstrap-menetelmää. Työn tuloksena on käytettävyyteen keskittyvä R-paketti, jonka avulla käyttäjät voivat laskea PAR- ja PAF-arvoja 2x2-kontingenssitauluista ja muodostaa niille luottamusvälit joko täysin Bayesilaisen lähestymistavan (Pirikahu et al.) tai Bootstrap-menetelmän avulla.

Menetelmien vertailu osoittaa, että vaikka Bootstrap tuottaa yleensä lyhyempiä luottamusvälejä, sen suorituskyky heikkenee tilanteissa, joissa altistumisaste on matala tai korkea tai otoskoko pieni. Näissä olosuhteissa Bayesilainen menetelmä tarjoaa tasaisemman kattavuuden. Pakettiin sisältyy myös alustava toiminnallisuus säädetyn väestön riskin laskemiseksi, tarjoten mahdollisuuden siirtyä monimutkaisempiin altistumisskenaarioihin.

Opinnäytetyö korostaa läpinäkyvyyden, toistettavuuden ja menetelmällisen valinnan merkitystä tilastollisen ohjelmistokehityksen kontekstissa. Lisäksi työ tuo esiin haasteita, joita tieteellisen kirjallisuuden rajoitettu saatavuus ja tilastopohjaisten ohjelmistopakettien vähäinen ylläpito aiheuttavat, ja puolustaa avoimempia ja kestävämpiä ratkaisuja tutkimustyökalujen kehittämisessä.

Asiasanat: Väestöattribuoitu riski (PAR), Väestöattribuoitu fraktio (PAF), Attribuoitu fraktio (AF), Luottamusväli, Bayesilainen päättely, Bootstrap, R, Ohjelmointi, 2x2 Ristiintaulukointi, Epidemiologia, Tilastomallinnus, Tilasto-ohjelmointi

Contents

1 Introduction

The primary focus of this thesis is to create an R package based on a paper by Pirikahu et al. (2016) titled "Bayesian methods of confidence interval construction for the population attributable risk from cross-sectional studies". [1] The authors propose a fully Bayesian framework for constructing confidence intervals for the population-attributable risk (PAR), a measure that quantifies the percentage of cases in a population that would not have occurred had the exposure not taken place.

Confidence intervals are typically associated with frequentist statistics, and Bayesian methods yield credible intervals. Repeated sampling makes it possible to examine whether a Bayesian credible interval exhibits characteristics of a frequentist confidence interval. A more detailed discussion on the distinction between confidence and credible intervals is provided in sections 2.1, and 2.4.

1.1 Research Question and Objectives

The main objective of this thesis is to implement the statistical model proposed by Pirikahu et al. (2016) in the form of an R package. This package will allow users to construct confidence intervals for PAR in a single exposure scenario using a Bayesian approach and a bootstrap approach.

The paper provides a complete probabilistic model, which I have implemented as an easy-to-use R package. The package is designed with usability and trans-

parency in mind, and the source code includes an example with real, openly and freely available data and evaluation code to simulate all viable epidemiological settings. Documentation is generated using the roxygen2 package, which allows code annotations to be compiled to .Rd files from comments in the code without disrupting the functionality of code itself. [2] This thorough documentation aims to ensure clarity and reduce the risk of user error.

To assess the performance and robustness of the Bayesian model, I have conducted extensive simulation studies using parameter values representative of real-world epidemiological settings, as described in the original paper. Simulations involve generating datasets under known conditions, constructing intervals with Bayesian and Bootstrap approaches, and evaluating the coverage and length of intervals in general and compared between methods. In addition to the R package, another key deliverable of this thesis is a dataset summarizing the simulation results.

Key R repositories for accessing additional packages include CRAN, Bioconductor, and Forge. The `devtools` package allows for installation of packages directly from GitHub. [3] I work with GitHub daily, so I chose to make my package available there, and it can be taken into use with the $devtools :: install\_github("peppi-lotta/par")$ command. R code, and the simulated dataset, are found in my GitHub account https://github.com/peppi-lotta/par.

1.2 Relevance and Significance

PAR and a related measure, population-attributable fraction (PAF), are essential tools in epidemiology. They estimate the potential reduction in disease incidence if an exposure were eliminated. These metrics are critical for policymakers, healthcare professionals, and other stakeholders when making data-based decisions to reduce disease burden at the population level. While a few R packages currently provide tools to compute PAF, many do not incorporate PAR or the latest statistical ad-

vancements, like the fully Bayesian approach introduced in the paper by Pirikahu et al.

This thesis contributes an up-to-date software package that allows users to specify whether they wish to calculate PAR or PAF and select the estimation method, Bayesian or bootstrap. This flexibility offers transparency and control, allowing users to understand the techniques and theory behind their analyses. My research has yielded evidence that bootstrap is the best frequentist approach, so I have included it along with the Bayesian approach and directly compare the two methods.

1.3 Structure of the Thesis

Chapter 2 introduces the fundamental concepts required to understand the paper by Pirikahu et al. (2016). It includes a definition for Bayes' theorem and its role in computing PAR and an overview of key statistical concepts such as prior and posterior distributions and likelihood. The intended audience is students of the Faculty of Technology with a limited background in statistics.

Chapter 4 presents a detailed description of the Bayesian model introduced in the "Bayesian methods of confidence interval construction for the population attributable risk from cross-sectional studies" paper. It includes a mathematical background, code implementation details, usage instructions for the developed R package, and a discussion of model evaluation using simulated data. Visualizations are provided to illustrate performance metrics and outcomes comparing Bayesian and bootstrap methods.

Chapter 5 presents the conclusions drawn from the evaluation results and reflects on the overall success of the R package in terms of usability and functionality. I also discuss the potential future directions for constructing confidence intervals for PAR and PAF.

1.4 Use of AI Tools

While writing this thesis, I used ChatGPT versions 3 and 4 to help with planning and idea generation, particularly when outlining the structure. All content has been written by me, while AI tools such as Grammarly and ChatGPT were only used for grammar and stylistic improvements, the analytical and technical content is original and verified from sources.

2 Fundamental Concepts

In this chapter, I define Bayesian inference, and the statistical terminology used in this thesis. The whole point of the methods explored is to construct a confidence interval, and confidence interval is probably the most used term in this work.

2.1 Confidence Interval

The confidence interval is a concept from frequentist statistics, based on repeated sampling theory. [4] It represents a range within which a parameter is expected to lie a specified percentage of the time a test is repeated. [5] The confidence interval quantifies the uncertainty of an estimate, and it has an upper and a lower limit. The width of the interval depends on two factors: sample size n and the estimate's standard deviation or standard error. In health sciences, a 95% confidence level is commonly used. [6] For a 95% confidence interval, the interval will contain the parameter in 95 out of 100 repeated samples. [4]

Confidence interval does not express the probability that the true value of the parameter lies within an interval. because in frequentist statistics, parameters are not assigned probabilities. [7] However, this is exactly what the Bayesian credible interval represents. A 95% credible interval means that there is a 95% chance the true value is within the interval. This is often how confidence intervals are mistakenly interpreted, making credible intervals more intuitive. [7]

One method for constructing a credible interval or confidence interval is the tail

method, which sets the lower and upper limits by symmetrically excluding the tails of the distribution. These limits are the $\alpha/2$ and $1 - \alpha/2$ percentiles. For a 95% interval, α is 0.05. [8]

2.2 Bayesian Inference

Inference refers to the process of summarizing or characterizing a model. [9] Inference involves finding a suitable model and fitting it to observed data. Bayes' Theorem is at the core of Bayesian inference because the goal is to predict values given some observed data. [10] Bayes' Theorem is an equation for calculating the conditional probability of event A happening given that event B has occurred. Equation for Bayes' Theorem is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{2.1}$$

If A and B are independent, then the conditional probability of A happening given that B has happened is just the probability of A, $P(A|B) = P(A)$. [11] While Bayes' theorem allows calculating conditional probabilities, a person doing calculations can place values, acquired through frequentist approaches, in the equation and get a conditional probability; thus, using Bayes' theorem alone does not constitute Bayesian inference. [12]

Bayesian inference is a broader statistical approach that treats the prior, likelihood, and posterior as distributions rather than fixed values. [12] This allows inference even with limited or uncertain data. [11] Bayesian inference's as well as statistical inference's goal in general is to make predictions about an unobserved set of data y based on an already observed set of data x. [10], [13]

The Bayesian approach to describing the connection between x and y is through conditional probability. Bayes' Theorem is at the core of Bayesian inference because

the goal is to predict sample y given that a set of samples x has been observed, where $x = \{x_1, x_2, ..., x_n\}$ and y is an unobserved sample x_{n+1}. The probability of observing a sample y given the set of samples x is

$$P(y|x) = P(x_{n+1}|\{x_1, x_2, ..., x_n\}). \tag{2.2}$$

When the amount of observation x_i starts to grow, it becomes increasingly complex to calculate the effect of each individual observation on y. If x_i are independent and identically distributed (i.i.d.), equation 2.2 can be simplified by replacing the sample set x with a distribution that is proportional to the sample set's distribution. [10] θ is used to represent the distribution and conditional probability for a new sample $P(y|\theta)$ is

$$P(y|\theta) = \frac{P(\theta|y)P(y)}{P(\theta)}. \tag{2.3}$$

[14]

Sample sets x are data that can be analysed, and inference is done based on data analysis. Bayesian data analysis follows a three-step process:

- Full probability model: Create a model consistent with the underlying scientific knowledge of the problem and, observed and unobserved data.
- Conditioning on observed data: Calculate and interpret the posterior distribution.
- Evaluation: Assess the model fit and the implications of the posterior distribution.

[10]

2.2.1 Prior

Equation 2.2 shows the relationship between belief $P(\theta)$ and data and how belief is changed through observed samples. [14] $P(\theta)$ is a prior distribution representing knowledge and plausibility of parameters before the data has been taken into account. The prior distribution is a way to represent both information and ignorance. [9], [12], [15]

$P(\theta)$ denotes the prior distribution. It may be based on earlier inference or general domain knowledge, but even without previous data, a prior must be chosen. Almost always, some domain knowledge is available and should be used to choose the most suitable prior. [12] Nonetheless, prior selection can be partly arbitrary. [9] Whether derived from past posteriors or heuristics, the prior reflects the beliefs of the analyst. [13] Priors can significantly influence the posterior and through it, the inference. [12] Priors' impact may range from negligible to significant. [9]

Calculating the posterior becomes easier when the prior is selected from the same distribution family as the posterior. Prior is chosen so that the parametric form of the prior is the same as the posterior's. This type of prior is called a conjugate prior because it is a conjugate to the likelihood distribution. [16] Other types of priors include maximum entropy priors, Laplace's prior, Jeffrey's prior, empirical priors, hierarchical priors, matching priors, reference priors, and invariant priors. [9] The prior distribution is not the only one affecting the posterior. The likelihood is another factor in calculating the posterior.

2.2.2 Likelihood

The likelihood function contains the information that the data x_i brings to the inference. [9] Likelihood is derived by excluding all outcomes inconsistent with observed data, and it is the function that describes a distribution of variables. [12] Bayesian inference obeys the likelihood principle. Two probability models with the

same likelihood function will yield the same inference for distribution θ. [10] The information provided by x_1 about distribution θ is contained in distribution $l(\theta|x_1)$. When a new observation x_2 is made, it needs to comply with the equation

$$l_1(\theta|x_1) = cl_2(\theta|x_2). \tag{2.4}$$

[9]

As sample size increases, the influence of the likelihood dominates over the influence of the prior. [12] The prior is modified by the data x trough the likelihood function. In 2.4 c is a constant. Multiplication by a constant leaves the likelihood function unchanged. Only the relative value of the likelihood matters, and multiplying by a constant will not affect the posterior. [15]

2.2.3 Posterior

The posterior is the probability p conditional on observed data. [12] Posterior is what we know of the final distribution given the knowledge of observed data and prior. As outlined in paragraphs 2.2.1 and 2.2.2, we know that likelihood is the only entity modifying the prior, so we get a rule for Bayesian inference: *posterior distribution* $\propto$ *likelihood* $\times$ *prior distribution* where,

$$P(\theta|x) \propto cP(x|\theta)P(\theta). \tag{2.5}$$

[15]

Posterior distributions, $P(\theta|x)$, are often complex and high-dimensional, making exact inference nearly impossible. The posterior function is often an integral that is not solvable analytically, and as a result, sampling methods are used to summarize the posterior. [17] Samples from the posterior are drawn rather than forming a mathematical function of the posterior. [12]

2.2.4 Posterior Sampling

Posterior sampling can be used to summarize and simulate the posterior. Samples are drawn in proportion to their posterior probability. The samples will have the same distribution as the actual posterior, and with a large enough sample set, the distribution can be accurately approximated. Sampling is part of the evaluation phase of Bayesian data analysis. [12] Summarization includes answering questions like: what is the frequency of parameter values in a specific interval, what are the upper and lower limits of credible or confidence intervals, and what are values such as the mean or median of the posterior. Simulations serve to check model behaviour and make predictions. They can also help validate the model by comparing its behaviour to a known distribution. Simulation can also be used to predict possible future observations. [12] Sampling needed in this thesis work is simple because posterior distributions are expressed analytically with the multinomial or Dirichlet distributions.

2.3 Multinomial Distribution

Multinomial distribution has a fixed number of trials that are independent from each other, and samples have a fixed set of outcomes. The probability mass function describes the chance that a discrete random variable is equal to a specific value. The sum of $f(x)$ over all x equals 1, and $f(x) = P(X = x)$. [18] The probability mass function of the multinomial distribution is

$$f(x_1, x_2, ..., x_k) = \frac{n!}{x_1!x_2!...x_k!} p_1^{x_1} p_2^{x_2} ... p_k^{x_k} \tag{2.6}$$

where $n = x_1 + x_2 + ... + x_{k-1} + x_k$. [19], [20]

In Bayesian analysis, the Dirichlet distribution is commonly used as a prior distribution to model a multinomial distribution. [20]

2.4 Bayesian Confidence Intervals

A Bayesian credible interval can be considered a confidence interval if it has the properties of a frequentist confidence interval. This is determined by simulating datasets with fixed parameters and evaluating the proportion of intervals that contain the actual value. [1], [8] The coverage rate indicates whether the interval is nominal. Nominal coverage means the true value lies within the interval, the expected number of times. [1]

2.5 Cross-Sectional Study

A cross-sectional study captures a snapshot of a population at a single point in time. Both exposure and outcome are measured at the same time. Subjects are selected from a population that is relevant to the study. This study method does not consider new cases of the disease that develop over a selected period of time.

Types of cross-sectional studies include:

- A Descriptive cross-sectional study is suitable for evaluating the prevalence of one or more health outcomes in a population.
- Analytical cross-sectional study measures the prevalence of outcomes and exposures. It's challenging to figure out causal relationships based on a cross-sectional study alone.
- A Repeated cross-sectional study conducts a study multiple times on the same population at different points in time. The individuals chosen for the tests at different study instances are not the same individuals. This type of study is suitable for showing changes in a population over time.

[21]

2.6 Population Data

Data about humans or animals often exhibit hierarchical structures, whether deliberately organized or otherwise, and this aspect should not be overlooked. [22] Typically, datasets consist of samples that are the lowest-level units but can be organized into higher-level units. For instance, a dataset containing student information can have students as the lowest-level units, which can then be grouped by class, school, or district. Students from a single school form a distinct cluster. [23] The model defined by Pirikahu et al. only considers one exposure variable and one outcome. In section 4.5 I demonstrate how analysis can be adjusted based on groups formed by one other variable.

2.7 Relative Risk

Relative risk expresses the risk of an outcome for a person with a chosen level of a risk factor compared to another person with a different risk factor exposure level.

$$Relative\ Risk = \frac{Incidence\ of\ disease\ in\ exposed\ group}{Incidence\ of\ disease\ inunexposed\ group}. \tag{2.7}$$

A synonym for relative risk is the risk ratio. Relative risk can be formed between groups that are exposed and unexposed or more exposed and less exposed. [24]

3 Literature on Confidence Interval Construction for Attributable Risk

This review focuses on literature that discusses methods for calculating and constructing confidence intervals for the population attributable risk (PAR) or population attributable fraction (PAF). It excludes papers that merely use PAR and PAF as an analysis component without contributions to methodologies. Relevant literature was obtained through searches of databases ResearchGate, Google Scholar, and PubMed using the keywords: attributable AND (risk OR fraction) AND "confidence interval". Some papers had titles that suggested the content fit the requirements for this review, but were excluded because I couldn't access them.

Drescher and Schill provide a concise historical overview of the development of attributable risk. The concept was first introduced by Levin in 1953, and a variance estimator was proposed by Walter in 1975. The notion of attributable risk was later generalized to accommodate multifactorial and polytomous risk factors by Walter (1976), Ejigou (1979), Walker (1981), Deneman and Schlesselman (1983), and Whittemore (1982, 1983). Alternative estimators based on the Mantel–Haenszel method were proposed by Greenland (1987) and Kuritz and Landis (1987, 1988). A unified approach to calculating attributable risk in a general multivariate setting was developed by Bruzzi et al. (1985), with variance estimators contributed by Benichou and Gail in 1990. [25]

Since its original introduction of the term "attributable proportion", the terminology surrounding this concept has expanded. As many as sixteen different terms have been used to denote the population attributable risk. Common alternatives include "attributable risk", "etiologic fraction", "fraction of etiology" "attributable fraction", "attributable risk percent", "preventable fraction", "prevented fraction", "assigned shares", "excess fraction", "risk fraction", and "rate fraction" [26], [27]

3.1 Role of Attributable Risk in Decision-Making

A high relative risk does not necessarily indicate a significant public health concern if the exposure is rare. A low relative risk may have a substantial impact if exposure is prevalent. The attributable fraction and attributable risk integrate both the relative risk and the proportion of the exposed population when calculating the burden of a risk factor. [25], [28] The true impact of a risk factor depends both on the size of the risk and the proportion of the population exposed to the risk factor. [25] PAR and PAF are tools for estimating the reduction in disease burden achievable through risk factor elimination. [29] PAR nor PAF are substitutions for relative risk as a health hazard appraisal tool, but serve as complementary measures. [30]

Effective disease prevention requires understanding the population-level health impact of risk factors. Most outcomes have multiple risk factors associated with them, and most risk factors are associated with multiple outcomes. Prevention can be done by encouraging healthy behaviour, taxation, financial incentives, campaigns, engineering, and legislation. Risk factors within a population are not immutable and are influenced by factors such as aging, improved healthcare, and vaccination programs. Once some risk factors are removed, others emerge more prominently in the population. [31] When a risk factor is rare and contributes little to the overall disease burden, it may not warrant attention in public health planning. [30]

Some risk factors exist on a continuous scale rather than as binary exposures,

meaning the factor is not dichotomous. For instance, while hypertension is typically defined as blood pressure exceeding 140 mmHg, many individuals with blood pressure below the hypertension threshold still have elevated blood pressure and face increased risk. Restricting analysis to dichotomous definitions may give an incomplete picture of a risk factor's total effect on a population. [31] While PAF and PAR assume complete elimination of exposure, such interventions are often unrealistic. For example, even with legal restrictions and cessation programmes, many people continue to smoke. [32]

3.2 Population Attributable Fraction

The term "attributable" implies causality, and population attributable fraction represents the fraction of all cases (exposed and unexposed) that would not have occurred if exposure to a risk factor had not occurred. [1], [32] PAF quantifies the proportion of cases attributable to a specific risk factor. [29] The total number of deaths, disease, and injury attributable to a risk factor is quantified by applying the population attributable fraction to the total number of outcomes. [31] The formula 3.1 that Pirikahu et al. use in their paper represents the PAF with probabilities and was originally proposed by MacManon and Pugh. [29] The equation used to mathematically describe PAF is

$$PAF = \frac{P(D^+) - P(D^+|E^-)}{P(D^+)}. \tag{3.1}$$

[1]

Here $P(D^+)$ denotes the overall probability of disease occurrence in the population, and $P(D^+|E^-)$ denotes the probability of disease in the unexposed population. [25]

3.3 Population Attributable Risk

There is a lot of confusion around the terms related to attributable risk and attributable fraction.[1] Greenland and Robin state, "The number of terms for attributable fractions is perhaps the largest of any concept in epidemiology". They also state, "While the concept (attributable fraction) is known by many names, we would think this variety would cause no problem as long as the conceptual and algebraic formulations were unambiguous." They argue that the concept of attributable fraction requires separation into the concepts of excess fraction, etiologic fraction, and incidence-density fraction. [33]

Pirikahu et al. also quote this paper by Greenland and Robins, and I can't find clear definitions or distinctions for population attributable risk in the papers Pirikahu et al. cite. I assume they give separate definitions to PAR and PAF with "unambiguous conceptual and algebraic formulations". I presume they have done this to bypass the ambiguity surrounding the terminology. The definition Pirikahu et al. give for PAR is "Population Attributable Risk allows us to determine the overall effect of a risk factor on society, by quantifying the reduction in the risk that could be achieved, if it were possible to completely remove the known risk factor from the population". The mathematical formulation for this is

$$PAR = P(D^+) - P(D^+|E^-). \tag{3.2}$$

Here, D is the disease status, and E is the exposure status. PAR is a probability distribution calculated by removing the probability distribution of disease cases in the non-exposed population from the probability distribution of all disease occurrences. [1]

Equations for PAR and PAF are very similar and consist of the same elements. Most papers consider attributable fraction when examining variance and confidence

intervals. The introduced methods apply to PAR as well as PAF. Now that we have determined equations for PAR and PAF, we can understand how variance and confidence intervals are estimated. Walter's 1976 paper, *The Estimation and Interpretation of Attributable Risk in Health Research*, was among the first to examine variance estimation for attributable risk in three different study designs, one of these being cross-sectional studies. The paper focuses on the simplest situation of dichotomous disease outcome and exposure. Combinations of disease and exposure are represented in a 2x2 contingency table. [30] With binomial exposure and disease outcome, the contingency table has 4 unique cells with no overlap between the cells. Example table 3.1 has four groups a, b, c, and d. Proportions are also presented in table 3.1. If total number of samples is n then proportion of a is $p_{11} = \frac{a}{n}$

Table 3.1: 2 x 2 Contingency Table For n Samples

Exposed	D^+ (has disease)	D^- (no disease)
E^+	a, p_{11}	b, p_{10}
E^-	c, p_{01}	d, p_{00}

PAF is calculated from the contingency table cell proportions using the equation

$$PAF = 1 - \frac{1}{(p_{11} + p_{10})(\psi - 1) + 1}, \tag{3.3}$$

where p_{xj} are the proportions of the 3.1 contingency table's cells and ψ is [30]

$$\psi = \frac{p_{11}(p_{01} + p_{00})}{(p_{11} + p_{10})p_{01}}. \tag{3.4}$$

[30]

3.4 Adjusted Attributable Risk

Originally, the attributable fraction was defined for a single dichotomous risk factor, a simplification that does not account for interactions with other exposures. [29]

PAF and PAR calculations can be extended to the joint effects of multiple exposures. PAF is generally less than the sum of individual PAFs, as people exposed to multiple risk factors should not be counted twice. The way risk factors work together is not straightforward, and the sum of attributable fractions could exceed 100% if calculated separately. [29], [32] These estimates rely on strong assumptions; unbiased study design and data analysis, appropriate adjustment for all confounders, and the assumption that removing the exposure does not influence other risk factors. [32] Risk factors can form causal chains that may involve indirect effects on the outcome, when one factor is a risk factor for another risk factor. This means that reducing any risk factor may lead to the prevention of an outcome. [31]

Sometimes it is impossible or ineffective to get matching case and control groups in a study. Sometimes the case group could be older than the control group, and the chosen risk factor is not independent of another factor. Higher age could mean longer exposure, as is the case for smoking, for example. One solution to account for different ages, in case and control groups, is to standardize estimates by calculating a weighted sum. In weighted sum, the weight is the proportion of cases in the attribute group. In the case of age, attribute groups could be 18-25, 26-35, 36-45, etc. [30] If weight ω_i is chosen as the count of samples in the adjustment group, the weight sum PAF is formulated as

$$PAF_s = \frac{\sum_i \omega_i PAF_i}{\sum_i \omega_i}. \tag{3.5}$$

We can consider age as the adjustment group, for example. PAF could be calculated for each age group separately, where PAF is weighted by a value that describes the age group's relative size among all the samples. [30] This method of setting the weight is called case-load weighting. DiMaso et al. give a simplified version of the weighted-sum equation, which is

$$PAF = \sum_{k=1}^{K} \omega_k PAF_k. \tag{3.6}$$

Here the sum of weights equals to 1, $\sum_{k=1}^{K} W_k = 1$. Weights could also be chosen to increase precision in what is called "precision-weighing". [29]

Risk factors are not always dichotomous and could be found on $(k+1)$ levels. Some level or levels are a baseline, and all other levels are associated with increased risk. n_1 is the total count of the exposure group $a+c$ and n_2 is the total count of the control group $b+d$. Attributable risk at exposure level i is $\hat{PAF}_i = \frac{(a_i d - b_i c)}{n_1 d}$ and the attributable risk for all levels

$$1 - \hat{PAF} = 1 - \sum_{i=1}^{k} \hat{PAF}_i = 1 - \frac{cn_2}{dn_1}. \tag{3.7}$$

Confounding over all exposure levels is the same as calculating exposure as dichotomous. [30] When equation 3.7 is written in the form of distributions and conditional distributions, it becomes

$$PAF = 1 - \sum_{j=1}^{k} \frac{P(C=j)P(D=1|E=0, C=j)}{P(D=0)}. \tag{3.8}$$

Here C is an adjustment group variable with k categories. Cell frequencies are $\hat{p} = (\hat{p}_{11j}, ..., \hat{p}_{22k})$ generated from a multinomial distribution $p = (p_{11j}, ..., p_{00k})$. [34]

DiMaso et al. go even further and explain a way to calculate a weighted sum where the confounding effect of an adjustment factor is accounted for, as well as different levels of exposure to the risk factor. [29]

$$_{adj}PAF_E = 1 - \sum_{k=1}^{K} \sum_{q=0}^{Q} \frac{\rho_{q,k}}{RR_{q|k}}. \tag{3.9}$$

[29]

First sum is taken over all adjustment groups, and the second sum is taken over

all risk factor levels in equation 3.9. $\rho_{q,k}$ is the proportion of cases qth risk factor level and kth adjustment group. $RR_{q|k}$ represents the relative risks for the qth risk factor level given the kth adjustment group. $\rho_{q,k}$ is replaced by the observed proportion of cases and by replacing $RR_{q|k}$ by the maximum-likelihood estimate. [29]

3.5 Constructing Confidence Interval

Variance estimation is essential for constructing confidence intervals around attributable risk estimates. [34] Attributable risk expressed with p, the proportion exposed to the risk factor and r, the relative risk, is

$$PAR = \frac{p(r-1)}{1+p(r-1)} \tag{3.10}$$

. PAR varies between 0 and 1. The maximum likelihood (ML) based interval for PAR is

$$\{\frac{\hat{p}(\hat{r}-1)exp(-u)}{1+\hat{p}(\hat{r}-1)exp(-u)}, \frac{\hat{p}(\hat{r}-1)exp(u)}{1+\hat{p}(\hat{r}-1)exp(u)}\}, \tag{3.11}$$

where $u = Z_{1-\frac{1}{2}\alpha}v$ and Z is a standard normal random variable such that $P(Z \leq Z_{1-\frac{1}{2}\alpha}) = 1 - \frac{1}{2}\alpha$.

Leung and Kupper expand upon Water's work and show that the logarithmic transformation (LT) method leads to a confidence interval with better coverage and shorter interval length than Walter's ML-based interval. For a cross-sectional study design, the variance of PAR is [28]

$$v^2 = \{\frac{(a+c)(c+d)}{ad-bc}\}^2\{\frac{ad(N-c)+bc^2}{Nc(a+c)(c+d)}\} \tag{3.12}$$

,

and $100(1-a)\%$ confidence interval is

$$\{\frac{(ad-bc)exp(-u)}{Nc+(ad-bc)exp(-u)}, \frac{(ad-bc)exp(u)}{Nc+(ad-bc)exp(u)}\} \tag{3.13}$$

, where $u = Z_{1-\frac{1}{2}\alpha}v$. [28]

If $|\hat{PAR} - \frac{1}{2}| \leq \frac{\sqrt{3}}{6}$, the Logarithmic Transformation produces a shorter interval than the Maximum likelihood-based approach. [28] Transformations can improve the estimated confidence intervals when they are normally distributed. Logit-transformations produce better results than Log-transformations as log-transformed intervals tend to be wider without improvement to coverage. [34] In addition to Walter's ML-based interval and Leung and Kupper LT-based intervals, there are newer and improved methods for constructing confidence intervals: Delta, Bootstrap, and Jackknife.

3.5.1 Delta

The Delta method is a standard approach for variance estimation for PAR and PAF. Estimates of the adjusted attributable risk are often obtained using the Delta method, but it tends to underestimate the standard error, leading to biased confidence intervals. [34] Bootstrap method generally outperforms the Delta method in terms of both coverage and interval length. [1]

3.5.2 Bootstrap

The bootstrap method was first introduced by Efron in 1979 and gained popularity as computing power increased. The unknown PAR is estimated by repeatedly sampling with replacement. For each sample $\hat{PAR}$ is computed. This process is repeated B times, and the distribution of the replications is the estimated distribution of the true parameter. [34] Samples may be drawn from a dataset (non-parametric bootstrap) or from a fitted model (parametric bootstrap) with replacement, where samples

are placed back into the sample pool. [1] A dataset with one exposure and one outcome can be represented in a 2x2 contingency table with four unique classification cells. The probability of selecting a classification is the same as the estimated value in a parametric model. The parametric and non-parametric bootstraps for a 2x2 contingency table are the same when the sample size is the same as the dataset size. [1]

Weighted methods can be used to reduce the computational burden of generating a large number of samples. The selection of the random weight vector is determined by the version of the bootstrap that is used. In nonparametric bootstrap random variables are taken from a multinomial distribution. A variant of the nonparametric bootstrap is the Bayesian bootstrap, where random variables are taken from the Dirichlet distribution. [34]

PAR values are between 0 and 1. Bootstrap confidence interval upper and lower limits are calculated by summing $P\hat{A}R$ values for each sample. Mathematically, the interval is

$$CI_{normal} = [\frac{1}{B}\sum_{b=1}^{B} P\hat{A}R_b^{*boot} - Z_{1-\alpha} \times \hat{se}(P\hat{A}R^{*boot}); \frac{1}{B}\sum_{b=1}^{B} P\hat{A}R_b^{*boot} + Z_{1-\alpha} \times \hat{se}(P\hat{A}R^{*boot})]. \tag{3.14}$$

The sum is weighted by the sample count $\frac{1}{B}$ to standardize the values between 0 and 1. The base for upper and lower limit is the same and what sets them apart from each other is adding or subtracting variance $Z_{1-\alpha} \times \hat{se}(P\hat{A}R^{*boot})$, where $Z_{1-\alpha}$ is the $100 \times (1-\alpha)$ percentile of the standard normal distribution and $\hat{se}(P\hat{A}R^{*boot})$ is

$$\hat{se}(P\hat{A}R^{*boot}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(P\hat{A}R_b^{*boot} - \frac{1}{B}\sum_{b=1}^{B} P\hat{A}R_b^{*boot})^2}. \tag{3.15}$$

[34]

3.5.3 Jackknife

The jackknife is another variant of computer-intensive methods that were suggested by Quenouille in 1949. In this method, the data set is changed systematically by leaving out every observation of the data one at a time. [34] The jackknife normal-based confidence interval looks almost the same as the mathematical notation 3.14 for Bootstrap. The jackknife normal-based confidence interval is

$$CI_{normal} = [\frac{1}{n}\sum_{i=1}^{n} P\hat{A}R_i^{*jack} - Z_{1-\alpha} \times \hat{se}(P\hat{A}R^{*jack}); \frac{1}{n}\sum_{i=1}^{n} P\hat{A}R_i^{*jack} + Z_{1-\alpha} \times \hat{se}(P\hat{A}R^{*jack})], \tag{3.16}$$

where $Z_{1-\alpha}$ is the $100 \times (1-\alpha)$ percentile of the standard normal distribution and $\hat{se}(P\hat{A}R^{*jack})$ is

$$\hat{se}(P\hat{A}R^{*jack}) = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}(P\hat{A}R_i^{*jack} - \frac{1}{n}\sum_{i=1}^{n} P\hat{A}R_i^{*jack})^2}. \tag{3.17}$$

[34]

3.5.4 Comparison of Methods

Lehnert-Batar et al. used simulation studies to compare Delta, Jackknife, Bootstrap, and Bayesian Bootstrap methods. The Bayesian and non-parametric bootstraps performed best, followed by the Jackknife. The Delta method generally underperformed, except in some cases, where a logit-transformation was applied. Bootstrap outperformed Jackknife in most scenarios, though both struggled with small sample sizes. [34]

Lee et al. compared the Green, Delta, and Monte Carlo methods for estimating 95% confidence intervals of the attributable fraction, and found no significant differences between them. [35] Lui (2001) compares five interval estimators for at-

tributable risk: two by Leung and Kupper, one based on logarithmic transformation by Fleiss, one based on Wald's test statistic by Walter, and one using an approach similar to Fieller's theorem. These were assessed using coverage probability and average interval length. All estimators produced less-than-nominal coverage. [36]

3.6 Software

Lehnert-Batar et al. used R to compare the Delta, Jackknife, Bootstrap, and Bayesian Bootstrap methods. Their software is available via ISSAN, the IMBE Statistical Software Archive Network: http://www.imbe.med.uni-erlangen.de/issan/issan.htm. [34] Several R packages allow estimation of different forms of the attributable fraction. Packages include epiR, AF, averisk, and pifpaf. [29] The AF package, introduced by Dahlqwist et al., has the broadest functionality. It provides functions for estimating confounder-adjusted attributable fractions in cross-sectional, case–control, and cohort studies. It supports binary exposures and is demonstrated using openly available real-world datasets. Other packages mentioned include epiR, attribrisk, and paf, though the authors noted in their 2015 paper that these were less up to date. [37]

4 The Bayesian Approach to Confidence Interval Construction for Population Attributable Risk (PAR)

In this chapter, I present the methodology proposed in the paper "Bayesian Methods for Confidence Interval Construction of Population Attributable Risk from Cross-Sectional Studies" by Pirikahu et al. (2016). I present an R package written based on the mathematical model explained earlier in this chapter, assess the model's performance with simulated data, and provide a sample workflow demonstrating the use of the R package on real data.

4.1 Mathematical Model

Pirikahu et al. explore a fully Bayesian approach to constructing a confidence interval for PAR. The model is straightforward to extend to other measures such as the PAF. The model is created to do inference in scenarios where there is one exposure and one outcome. These scenarios can be represented in a 2x2 contingency table format. The table has four unique cells that don't overlap. Contingency table 4.1 defines cell counts a, b, c, and d, where a represent a situation where $P(D^+ \cap E^+)$ a population specimen is exposed AND has disease.

n is total sample size, $n = a + b + c + d$. The probability of exposure is $P(E^+) =$

Table 4.1: 2 x 2 Contingency Table For n Samples

Exposed	D^+ (has disease)	D^- (no disease)	Total
E^+	a	b	a + b
E^-	c	d	c + d
Total	a + c	b + d	n

$\frac{a+b}{n}$, the probability of being unexposed is $P(E^-) = \frac{c+d}{n}$, and the probability of having the disease, regardless of exposure status, is $P(D^+) = \frac{a+c}{n}$.

The PAR is calculated using the formula 3.2 first introduced in section 3.2. By aplying Bayes' Theorem to component $P(D^+|E^-)$ the equation can be writen in to another form

$$\begin{aligned} PAR &= P(D^+) - P(D^+|E^-) \\ &= P(D^+) - \frac{P(E^-|D^+)P(D^+)}{P(E^-)}. \end{aligned} \tag{4.1}$$

Values observable from the contingency table values are placed into the PAR equation. Probability of $P(A \cap B)$ is equal to $P(A|B) \times P(B)$ so

$$P(E^-|D^+)P(D^+) = P(E^- \cap D^+) = c. \tag{4.2}$$

PAR equation in terms of the contingency table counts is

$$\begin{aligned} PAR &= P(D^+) - P(D^+|E^-) \\ &= P(D^+) - \frac{P(E^-|D^+)P(D^+)}{P(E^-)} \\ &= \frac{a+c}{n} - \frac{\frac{c}{n}}{\frac{c+d}{n}} \\ &= \frac{a+c}{n} - \frac{c}{n} \times \frac{n}{c+d} \\ &= \frac{a+c}{n} - \frac{c}{c+d} \\ &= \frac{a+c}{a+b+c+d} - \frac{c}{c+d}. \end{aligned} \tag{4.3}$$

2x2 contingency table forms 4 mutually exclusive outcomes. These groups can be modeled with the multinomial distribution, $(a, b, c, d) \sim Multinomial(n, p_{11}, p_{10}, p_{01}, p_{00})$.

Using the Bayes' theorem notation, the posterior distribution is

$$P(a, b, c, d|p_{11}, p_{10}, p_{01}, p_{00}) = P(\theta|x) \propto f(x|\theta)P(\theta). \tag{4.4}$$

Given the conjugacy between the multinomial and Dirichlet distributions, the posterior distribution is expressed analytically as

$$\theta|x \ Dirichlet(a + 1, b + 1, c + 1, d + 1). \tag{4.5}$$

Representing posteriors analytically is computationally less expensive than using, for example, MCMC simulation. The confidence interval is constructed by drawing samples from the posterior and calculating PAR for each sample. The fully Bayesian approach produces a credible interval. The credible interval is a confidence interval because it exhibits frequentist coverage. Evaluation code tests show that this is true. The results are discussed in 4.4. In the evaluation test, I create 1000 samples that represent contingency tables corresponding to a known PAR value. The code generates the interval for that imaginary contingency table and compares whether the actual PAR value lies within the generated interval at least 95% of the sample cases.

4.2 R Code

Implementation in R becomes straightforward once the mathematics is well understood. The construction of a confidence interval involves four key steps; extracting values from data sets and forming contingency tables, simulating contingency tables, calculating PAR for a contingency table, and constructing the confidence interval for a contingency table.

4.2.1 Extracting Contingency Table Values

The R method 4.2.1 *extract_abcd* extracts values a, b, c, and d from a data frame provided as a parameter. The function returns a, b, c, and d as an ordered vector. Values a, b, c, and d are the cell values described in contingency table 4.1. All the functions created for this package use these category values, and the values are always expected to be in this same order. This standardized output ensures compatibility with other functions in the package. This helper function is provided to reduce user-made errors.

```
extract_abcd <- function(
    data,
    exposure_col,
    outcome_col)
{
    x_0e0d <- sum(data[[exposure_col]] == 0
        & data[[outcome_col]] == 0)
    x_0e1d <- sum(data[[exposure_col]] == 0
        & data[[outcome_col]] == 1)
    x_1e0d <- sum(data[[exposure_col]] == 1
        & data[[outcome_col]] == 0)
    x_1e1d <- sum(data[[exposure_col]] == 1
        & data[[outcome_col]] == 1)

    return(c(
    x_1e1d = x_1e1d,
    x_1e0d = x_1e0d,
    x_0e1d = x_0e1d,
    x_0e0d = x_0e0d
```

```
    ))
}
```

Listing 4.1: R function to extract a, b, c, and d from a data frame given exposure and outcome columns

In function 4.2.1, the user provides a dataset and the names of exposure and outcome columns. This function only works for dichotomous data where exposure and outcome are represented by 0 and 1. 0 indicates the absence of exposure or outcome, and 1 indicates the subject has been exposed or has the outcome. The user has to preprocess their data to fit this format.

4.2.2 Calculating PAR

The function in 4.2.2 computes PAR from a vector containing a, b, c, and d. The input vector can be obtained using the function *extract_abcd*. By using the *extract_abcd* function, the user can save time because they don't have to spend time figuring out what order the values should be in. Function 4.2.2 only takes a vector as input and extracts the vector's values for further calculations.

```
calculate_par <- function(x) {
    x <- as.numeric(x)
    a <- x[1]
    b <- x[2]
    c <- x[3]
    d <- x[4]

    if (c + d == 0) {
        return(0)
    }
```

```
    par <- (a + c) / (a + b + c + d) - c / (c + d)


    return(par)
}
```

Listing 4.2: R function to calculate PAR from a contingency table vector

The logic for calculating PAR is taken straight from the last line of 4.3. Especially when the total number of n is small, the values for c and d might be 0. Since zero cannot be a divisor, I have opted to have the function return 0 as the value for PAR. Having no disease means there is no disease attributable to any risk factor, so it makes sense that attributable risk is then 0.

The function 4.2.2 *calculate_par* returns a single value, and it is the PAR. The package has a similar function for calculating the PAF. These functions were created separately to ensure similar functions could be added to the package later without disrupting existing functionality.

4.2.3 Constructing the Bayesian Confidence Interval

Like the 4.2.2 *calculate_par* function, the *calculate_bayesian_ci* method requires the user to provide a vector of values as a parameter. It also accepts optional parameters: interval, a prior distribution vector, and the number of samples. Vectors x and *prior* values must be ordered as a, b, c, and d. The sample size is set to 10,000, which is generally considered sufficient based on Pirikahu et al. Prior defaults to a vector of ones, representing a non-informative uniform distribution. The default interval coverage is 0.95, a commonly used value. The user has the freedom to tweak all values used in the functions. This was done to achieve the goal of usability by increasing user freedom. The user can also specify the "type" of confidence interval they are calculating. Currently, PAR and PAF are supported.

```
calculate_bayesian_ci <- function(
    type,
    x,
    interval = 0.95,
    prior = c(1, 1, 1, 1),
    sample_count = 10000
) {
  x <- as.numeric(x)
  a <- x[1]
  b <- x[2]
  c <- x[3]
  d <- x[4]
  n <- a + b + c + d
  prior <- as.numeric(prior)
```

Listing 4.3: R function to construct a Bayesian confidence interval for PAR or PAF from a contingency table vector

The core logic of the code is shown in 4.2.3. In the main part of the function, samples of contingency tables are generated using the Dirichlet distribution. The number of generated contingency tables is specified by *sample_count* . Generation is done by calling the *rdirichlet* function from the *MCMCpack* package. The resulting tables are stored in the *samples* variable. Using the apply function, *calculate_par* is applied to each generated table producing PAR values. The confidence interval is then computed from these PAR values using the *quantile* function. The function returns a matrix with two values: the lower and upper bounds of the confidence interval.

```
  samples <- MCMCpack::rdirichlet(
```

```
    sample_count,
    c(a + prior[1],
      b + prior[2],
      c + prior[3],
      d + prior[4],
      n
    )
  )
  samples <- apply(samples, 2, function(x) x * n)

  if (type == "par") {
    par_samples <- apply(samples, 1, calculate_par)
  } else if (type == "paf") {
    par_samples <- apply(samples, 1, calculate_paf)
  } else {
    stop("Invalid type. Please use 'par' or 'paf'")
  }

  # Calculate the confidence interval
  confidence_interval <- quantile(
    par_samples,
    c(
      (1 - interval) / 2,
      1 - (1 - interval) / 2
    )
  )
  return(matrix(c(
```

```
    confidence_interval[1],
    confidence_interval[2]
  )))
```

Listing 4.4: Core functionality for constructing the Bayesian confidence interval for PAR or PAF

Despite multinomial simulations generally being more efficient than MCMC simulations, conducting evaluations is resource-intensive. When the *compiler* is loaded, the *compile_all* function is called and all functions within the package are converted from human-readable code to machine code, enhancing execution speed. For bootstrap and bayesian the function is almost the same, except samples are generated without any priors using the rmultinom function as shown in example 4.2.3.

```
    samples <- t(rmultinom(
        sample_count,
        n,
        c(p_11, p_10, p_01, p_00)))
```

Listing 4.5: R code to generate multinomial samples for confidence interval construction

The ‘cmpfun‘ function from the Byte Code Compiler package compiles functions into machine code. Function 4.2.3 *compile_all* compiles the body of a closure and returns a new closure with the same parameters while replacing the original body with the compiled expression. This makes the functions run faster. [38]

```
    compile_all <- function() {
        calculate_bayesian_ci <-
            cmpfun(calculate_bayesian_ci)
        calculate_bootstrap_ci <-
            cmpfun(calculate_bootstrap_ci)
```

```
        calculate_par <-
            cmpfun(calculate_par)
        calculate_paf <-
            cmpfun(calculate_paf)
        extract_abcd <-
            cmpfun(extract_abcd)
    }
```

Listing 4.6: Compiling functions to byte code for improved performance

This optimization is necessary for running multiple simulations, as is done in the evaluation code in the next section 4.3.

4.3 Evaluation of the Model

Performance of the Bayesian approach is evaluated using simulated data created based on selected known values for parameters p, q, e and n.

- $p = P(D^+|\ E^+)$, the probability of having the disease given exposure.
- $q = P(D^+|\ E^-)$, the probability of having the disease given no exposure.
- $e = P(E^+)$, the probability of exposure in population.
- n, the total number of samples.

$P(E^-) = 1 - P(E^+) = 1 - e$. because exposure either has happened or not. Same way the disease is dichotomous and $P(D^-|E^-) = 1 - P(D^+|E^-) = 1 - q$. and $P(D^+|E^+) = 1 - P(D^-|E^+) = 1 - p$. Probabilities for a is

$$a = p_{11} \times n = P(D^+ \cap\ E^+) \times n = P(D^+|\ E^+) \times P(E+) = p \times e \times n. \qquad (4.6)$$

In a similar way b, c, and d can be expressed in terms of p, q, e, and n:

- $b = p_{10} \times n = (1 - p) \times e \times n$
- $c = p_{01} \times n = q \times (1 - e) \times n$
- $d = p_{00} \times n = (1 - q) \times (1 - e) \times n$

Rate of dicease ocurance is $P(D^+)$ and is calculated as

$$\begin{aligned} P(D^+) &= P(D^+ \cap\ E^+) + P(D^+ \cap\ E^-) \\ &= P(D^+|E^+)P(E^+) + P(D^+|E^-)P(E^-) \\ &= p \times e + q \times (1 - e). \end{aligned} \tag{4.7}$$

1000 contingency tables are generated using the multinomial distribution. The generated tables correspond to selected variables p, q, e, and n. The parameter values for the simulation are as follows

Table 4.2: Parameters for the simulation

p	0.001	0.01	0.05	0.1	0.2	0.3	0.35	0.4	0.45	0.5
q	0.001	0.01	0.05	0.1	0.2	0.3	0.35	0.4	0.45	0.5
e	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9

We can expand this evaluation matrix to compare different sample sizes: small, small-medium, large-medium, and large. Counts are in table 4.3

Table 4.3: Sample size parameters for the simulation

n	16	64	256	1024

4.3.1 Code for Evaluating the Model

Probabilities for a, b, c, and d need to be reverse-engineered from the known parameters given in table 4.3. From the probabilities, contingency table samples are

generated and confidence intervals constructed. Confidence intervals are constructed with two different methods: the fully Bayesian approach proposed by Pirikahu et al. and the bootstrap.

The function 4.3.1 *get_probabilities_2x2_table* computes and returns the probabilities for a, b, c, and d as p_11, p_10, p_01 and p_00. The probability calculations are done with the equation from list 4.3.

```
get_probabilities_2x2_table <- function( p, q, e ) {
    p_11 <- p * e
    p_10 <- ( 1 - p ) * e
    p_01 <- q * ( 1 - e )
    p_00 <- ( 1 - q ) * ( 1 - e )

    return(list(
        p_11 = p_11,
        p_10 = p_10,
        p_01 = p_01,
        p_00 = p_00)
    )
}
```

Listing 4.7: R function to compute cell probabilities for a 2x2 table given p, q, and e

Multiplying the probabilities with total sample size n, 1000 contingency tables are generated that have values a, b, c, and d that corespond to parameters p, q, e. For each of these tables, the confidence interval is computed using the *calculate_bayesian_ci* function. Pirikahu et al. state that 10,000 simulations would be ideal for constructing the confidence interval. Due to resource constraints, I have reduced the number of simulations to 1000. Simulated contingency tables are saved

to *samples* variable. Samples are generated with the rmultinom function.

```
samples <- rmultinom(
    1000,
    row$n,
    c(row$p_11, row$p_10, row$p_01, row$p_00)
)
```

Constructing the Confidence Interval

All 1000 generated tables are looped through. To calculate a confidence interval, the $calculate_bayesian_ci$ function is called.

```
bayes_cis <- apply(samples, 2, function(sample) {
  a <- sample[1]
  b <- sample[2]
  c <- sample[3]
  d <- sample[4]
  n <- a + b + c + d
  calculate_bayesian_ci(
    "par",
    c(a, b, c, d),
    interval,
    prior,
    1000
    )
})
```

Listing 4.8: Applying the Bayesian confidence interval calculation to each simulated contingency table

Bootstrap method is applied to the same set of samples to get a confidence interval for each sample

```
boot_cis <- apply(samples, 2, function(sample) {
  a <- sample[1]
  b <- sample[2]
  c <- sample[3]
  d <- sample[4]
  n <- a + b + c + d
  calculate_bootstrap_ci(
    "par",
    c(a, b, c, d),
    interval,
    10000
    )
})
```

Listing 4.9: Applying the Bootstrap confidence interval calculation to each simulated contingency table

Metrics

The coverage is considered nominal when the actual PAR falls within the lower and upper bounds of the interval in 95% of the simulations. Calculating the actual PAR from p, q and e is done in order to calculate the coverage percentage. Values from 4.3 are placed into 3.2 to get an equation to calculate PAR from the p, q, and e parameters.

$$\begin{aligned} PAR &= P(D^+) - P(D^+|E^-) \\ &= p * e + q * (1 - e) - q. \end{aligned} \tag{4.8}$$

The second row of equation 4.8 is directly implemented in code. Coverage percentage is calculated by checking if the actual PAR value lies within the upper and lower bounds of the constructed confidence interval and dividing the cases where PAR is in the interval by the total number of samples.

```
    bayes_coverage <- mean(
        bayes_cis[1, ] <= row$actual_par
        & bayes_cis[2, ] >= row$actual_par
    )
```

Listing 4.10: Calculate the coverage percentage for the Bayesian confidence interval

The mean length of the interval across all simulations is computed along with the coverage percentage.

```
bayes_mean_length <- mean(
    bayes_cis[2, ] - bayes_cis[1, ]
)
```

Listing 4.11: Calculate the mean length of the Bayesian confidence interval

Coverage percentage and mean interval length are two metrics that are used to compare different models. If the coverage percentages of all models meet the chosen 95% level the model has nominal coverage, and the model with the narrowest mean length is considered the most effective.

Coverage percentage and mean interval length were calculated for both Bayesian and Bootstrap methods, and results were saved in a CSV file for further analysis. The CSV file has columns described in table 4.3.1. The file is included in the examples directory in the par package along with the complete evaluation code.

Table 4.4: Description of columns in the evaluation results CSV file

Column	Description
V1	An identifier for each simulation run.
p	Probability of disease given exposure, $P(D^+\|E^+)$.
q	Probability of disease given no exposure, $P(D^+\|E^-)$.
e	Probability of being exposed, $P(E^+)$.
n	Total sample size used in the simulation.
p_11	Joint probability, $P(D^+ \cap E^+)$.
p_10	Joint probability, $P(D^- \cap E^+)$.
p_01	Joint probability, $P(D^+ \cap E^-)$.
p_00	Joint probability, $P(D^- \cap E^-)$.
actual_par	The true Population Attributable Risk (PAR) used for simulation.
bayes_ci_mean_length	Mean length of the Bayesian credible interval over simulations.
bayes_ci_coverage	Proportion of simulations where the Bayesian credible interval covered the true PAR.
boot_ci_mean_length	Mean length of the bootstrap confidence interval over simulations.
boot_ci_coverage	Proportion of simulations where the bootstrap interval covered the true PAR.
interval	Confidence level used for interval construction (e.g., 0.95).
prior	Prior vector used for the Dirichlet distribution, expressed as a string of four comma-separated values.

Optimizing the Evaluation Code

The steps outlined in paragraph 4.3.1 are computationally expensive. Generating 1000 contingency tables and constructing of the confidence interval requires 1000 simulations for both Bayes and Bootstrap separately. This amounts to $1000 * 1000 * 2 = 2,000,000$ simulations. This would be even more if the 10,000 samples deemed sufficient by Pirikahu et al. were used.

The values and calculations that do not require simulations are computed outside of a loop and output to a CSV file. Calculations done at this phase include the actual PAR and the probabilities p_11, p_10, p_01, p_00. Then the simulations are executed in a loop, utilizing values from the CSV file. Stored values are used instead of calculating the same values multiple times. This approach also allows

simulation to be divided into smaller subsets if needed. To further speed up the evaluation, the functions we compiled to machine code as demonstrated in 4.2.3. Machine code is faster to run than uncompiled code. Additionally, I utilize the *parallel* package to execute multiple samples in parallel, allowing evaluation code to leverage the available cores on a machine. The simulation can utilize all cores if no other processes are running; otherwise, it will run on any unused cores. The variables 'start' and 'end' are used to define the first and last rows of a subset.

4.4 Comparison of Fully Bayesian Method with Bootstrap Method

All visualizations indicate the same result, which is; a fully Bayesian approach yields coverage that is close to the nominal level across all scenarios, while the confidence intervals constructed with the Bayesian method are generally wider than those from the Bootstrap method. The difference in interval lengths becomes negligible when the total sample size reaches 64.

The goal interval coverage is 95% although this is not always achieved. Figure 4.1 shows how often Bayesian gets closer to nominal compared to the bootstrap method. Figure 4.2 shows that with both methods. Figure plot are divided based on sample size n and prior used when generating the Bayesian confidence interval. From the figure it is seen nominal coverage is achieved less than 50% of the time.

Figure 4.3 shows how many percent points the interval falls short of nominal coverage. Differences between methods are most noticeable when the sample size is small, 16 samples. The Bayesian method tends to achieve coverage around 90%-98% whereas the bootstrap never achieves nominality and rarely gets coverage over 80%. The difference in coverage percentage gets smaller when sample sizes increase.

Figure 4.4 shows how often Bayesian confidence intervals are shorter than boot-

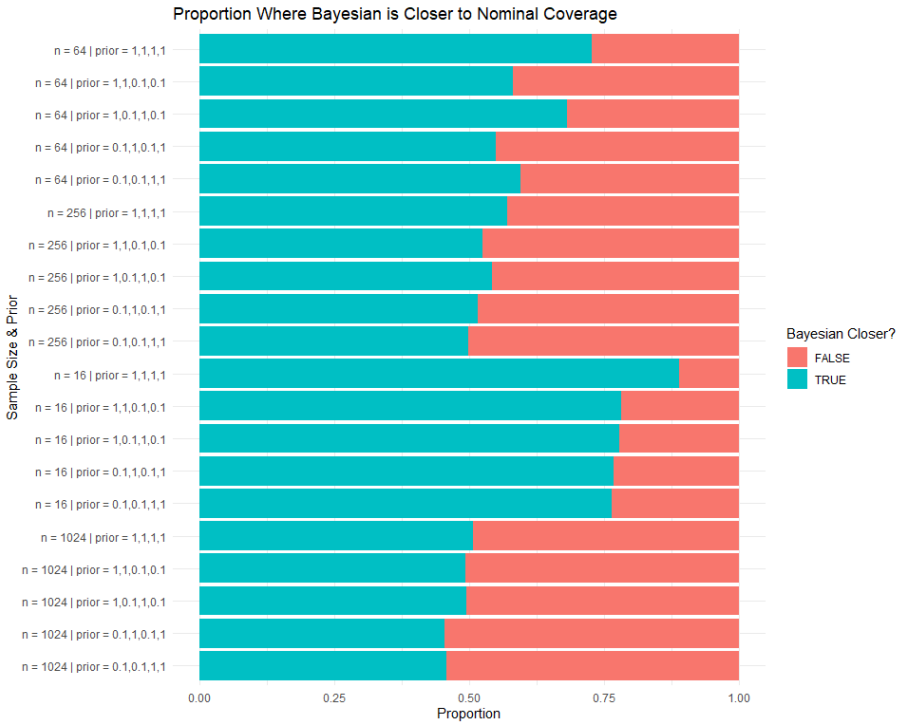


Figure 4.1: Compare Bayesian and Bootstrap methods on which is closer to nominal coverage. Sections are divided by sample size and prior distribution.

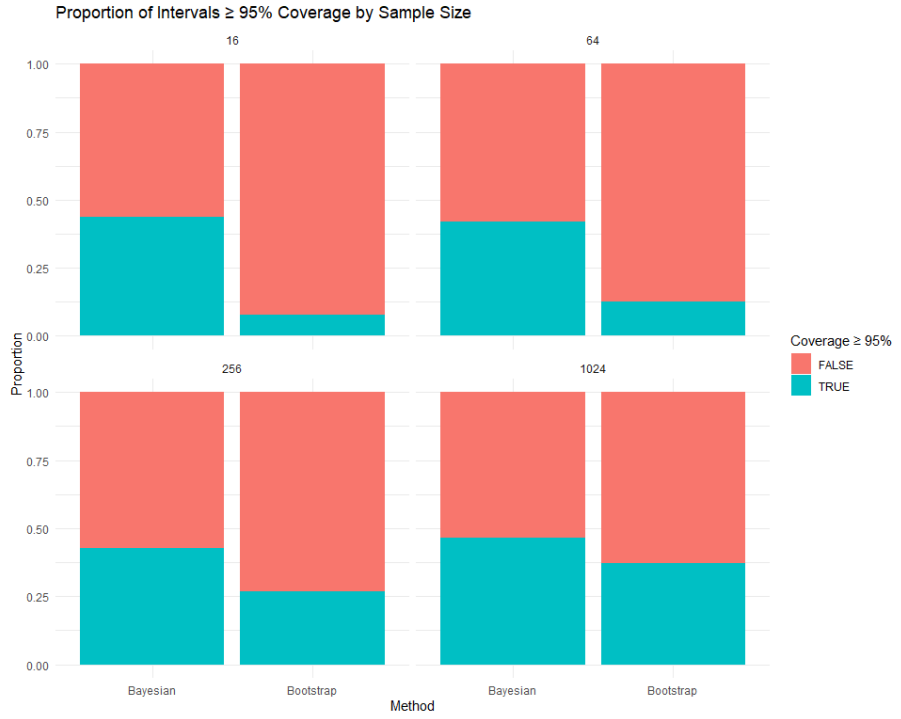


Figure 4.2: Compare Bayesian and Bootstrap methods based on which achieves nominal coverage. Sections are divided by sample size.

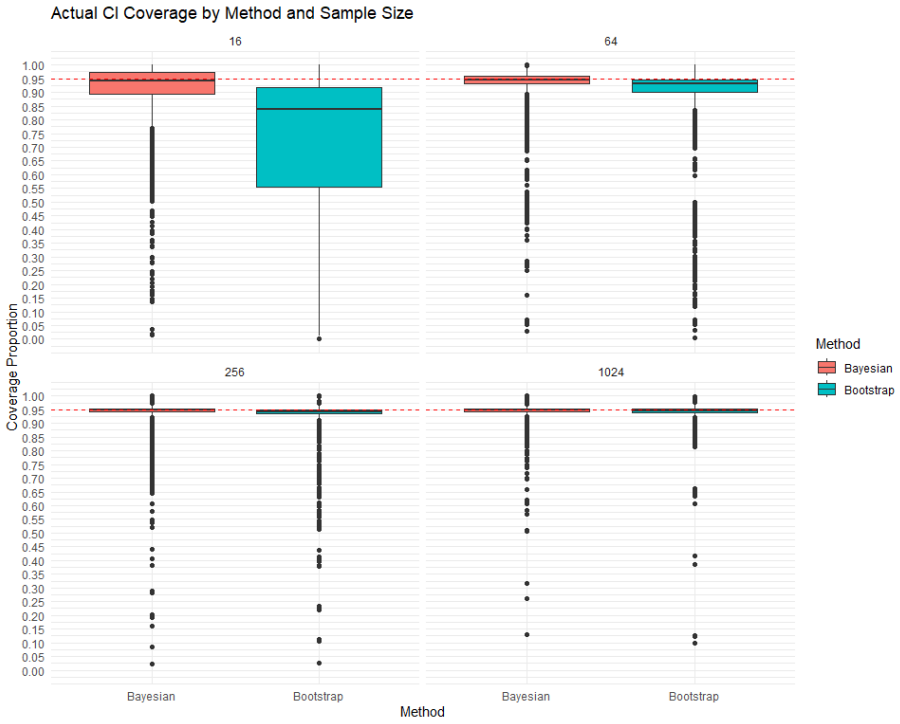


Figure 4.3: Compare Bayesian and Bootstrap methods on which method achieves nominal coverage box plot. Sections are divided by sample size.

strap intervals, which is rarely. Figure 4.5 shows the difference in intervals. Visuals are sectioned by sample size. Red represents Bayesian, blue represents bootstrap, and gray represents the overlap in the boxplot. The Box plot indicates different results from figure 4.4. Figure 4.5 indicates there is no significant difference in interval length when the sample size is large, and not much of a difference between methods when sample sizes are small. The Bayesian method produces longer intervals, and that probably explains why the coverage tends to be better as well.

Figure 4.6 demonstrates that using a uniform prior performs well in all evaluated conditions, and a Bayesian approach is especially advantageous when sample sizes are small. In contrast, the Bootstrap method struggles in situations where the contingency table values c and d are small and resampling can lead to zero values for these parameters. As shown in 4.3, since c and d are used as denominators in the calculations, the method fails mathematically when these values are zero. This

Figure 4.4: Compare Bayesian and Bootstrap methods based on which produces shorter intervals. Sections are divided by sample size and prior distribution.

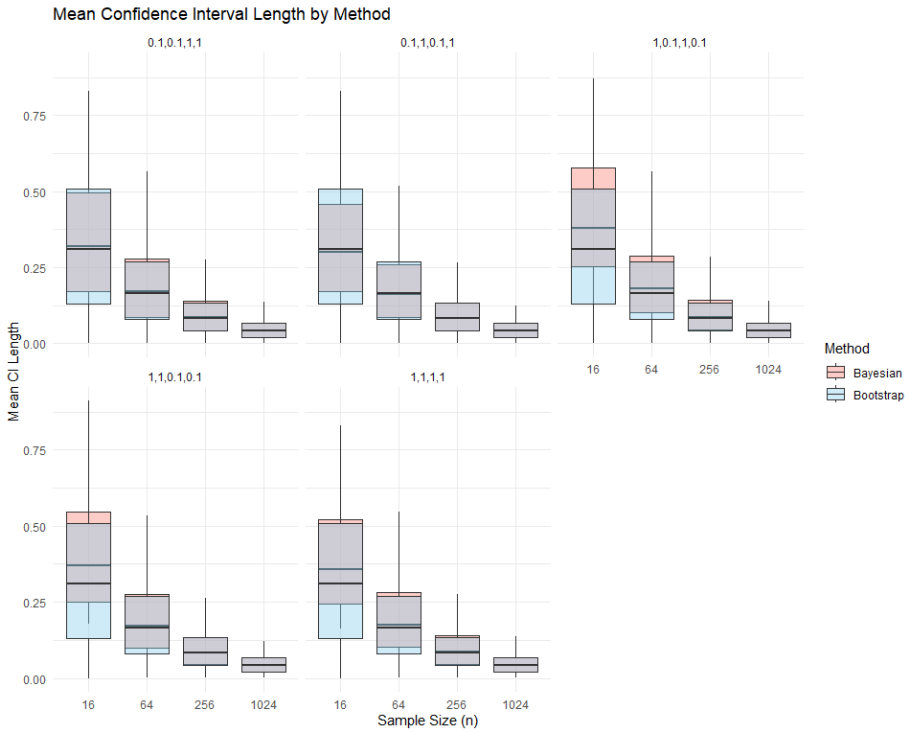


Figure 4.5: Box plot comparison of interval length. Sections are divided by sample size and prior distribution.

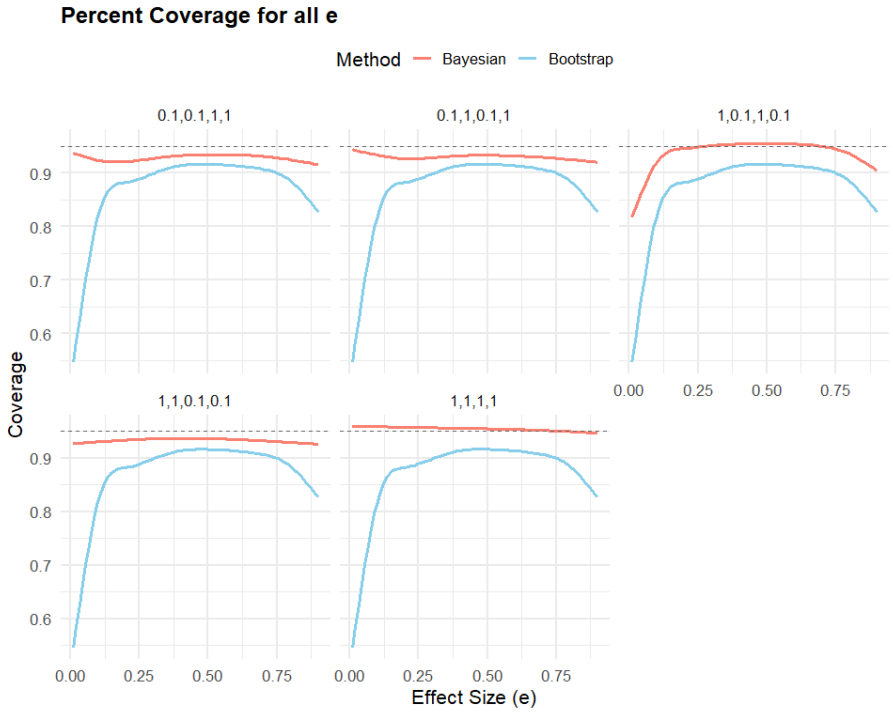


Figure 4.6: Coverage percentage over all e values

mathematical limitation explains why the Bootstrap approach is unreliable when c and d are zero or near-zero. In contrast, the Bayesian method incorporates a prior distribution that effectively prevents these values from being zero, thereby ensuring the mathematical validity of the estimates. While zero values for c and d are unlikely in large-sample scenarios and generally not problematic, in small-sample settings, zero values do not necessarily reflect true probabilities. The Bayesian prior accounts for this uncertainty in the construction of the interval, resulting in better performance compared to Bootstrap. In summary, the Bayesian approach offers greater stability across a range of conditions. However, when sample sizes are sufficiently large, the Bootstrap method remains a valid and efficient alternative.

4.5 Example with Real Data

To demonstrate the code developed in this thesis, I have utilized the dataset "Risk Factors for Cardiovascular Heart Disease", curated by Kuzak Dempsy and made available on Kaggle. [39] This dataset provides a comprehensive collection of health-related variables known to influence cardiovascular disease risk, aligning closely with risk factors identified by the Centers for Disease Control and Prevention. [40] The dataset includes the following features described in table 4.5

Table 4.5: Description of variables in the cardiovascular disease dataset

Variable	**Description**
Age	Age of the individual, recorded in days (integer)
Gender	Gender of the individual (male = 1, female = 2)
Height	Height in centimeters (integer)
Weight	Weight in kilograms (integer)
ap_hi	Systolic blood pressure reading (integer)
ap_lo	Diastolic blood pressure reading (integer)
Cholesterol	Cholesterol level, categorized into ordinal groups (integer)
Gluc	Blood glucose level, categorized into ordinal groups (integer)
Smoke	Smoking status (boolean)
Alco	Alcohol consumption status (boolean)
Active	Physical activity status (boolean)
Cardio	Presence (1) or absence (0) of cardiovascular disease (target variable)

4.5.1 Code

This section describes the demo code used to compute the Population Attributable Risk (PAR) and its corresponding confidence intervals using Bayesian and bootstrap methods, along with age group-based standardisation. Required libraries for calling the function used are

```
library(MCMCpack)
library(dplyr)
library(data.table)
library(devtools)
```

```
# Optional
library(compiler)
```

Listing 4.12: Required R libraries for Bayesian and bootstrap PAR estimation

The custom `par` package, used for estimating the population attributable risk and Bayesian confidence intervals, is installed from GitHub.

```
devtools::install_github("peppi-lotta/par")
library(par)
```

Listing 4.13: Install and load the custom `par` package from GitHub

The dataset is read from a CSV file, and a random subset of 1000 observations is selected. The Body Mass Index (BMI) is calculated, and individuals with a BMI greater than 24.99 are classified as overweight. Being overweight is a risk factor for cardiovascular disease, and being overweight is marked with 1, and not being overweight is marked with 0. Being overweight is a known risk factor; this is why I've chosen it as the example risk factor. This is the preprocessing phase of the data analysis workflow. The package can only handle dichotomous data, so having raw weight data will not produce any results.

```
file_path <- "./data.csv"
data <- read.csv(file_path)
data <- data[sample(nrow(data), 1000), ]
data <- data %>%
    mutate(over_weight = ifelse(
        weight/((height/100)^2) > 24.99, 1, 0
    ))
```

Listing 4.14: Preprocessing the cardiovascular dataset: creating the overweight variable

Calculate the Population Attributable Risk and Confidence Interval

Values from data are extracted with *extract_abcd* function, and the output is given as input to *calculate_par*.

```
# Extract the values of a, b, c, and d from the data
x <- extract_abcd(data, exposure_col, outcome_col)


# Calculate the population attributable risk
par <- calculate_par(x)
cat("PAR: ", par, "\n")
```

Listing 4.15: Extracting contingency table values and calculating PAR from the dataset

```
PAR:  0.1048598
```

Calculating a confidence interval is done by calling the function *calculate_bayesian_ci*

```
interval = 0.95
prior = c(1, 1, 1, 1)
sample_count = 10000


bay_ci <- calculate_bayesian_ci(
    "par",
    x,
    interval,
    prior,
    sample_count
)
cat("Confidence Interval:\n")
```

```
print(bay_ci)
```

Listing 4.16: Calculating the Bayesian confidence interval for PAR from the dataset

Output is:

```
Confidence Interval:
          [,1]
[1,] 0.04895299
[2,] 0.13142518
```

Bootstrap confidence interval is calculated with the function *calculate_bootstrap_ci*. Only differences between *calculate_bootstrap_ci* are function names, and that *calculate_bootst* function does not accept a vector with priors as a parameter.

```
boot_ci <- calculate_bootstrap_ci(
    "par",
    x,
    interval,
    10000
    )
```

Listing 4.17: Calculating the Bootstrap confidence interval for PAR from the dataset

Output is:

```
Confidence Interval:
          [,1]
[1,] 0.04963707
[2,] 0.13127293
```

Crude single-exposure PARs and PAFs are biased due to lack of confounder adjustment. The adjusted attributable fraction quantifies the impact of removing a

single risk factor while controlling for others. For two dichotomous risk factors $E1$ and $E2$, the crude and adjusted attributable fractions coincide only when $E1$ and $E2$ are independent or when $E2$ does not increase disease risk. The weighted-sum method accounts for confounders and effect modifiers, but can be biased when data is sparse. This happens because risk factors are not independent, and multi-exposed cases will be counted more than once. Adjusted attributable fractions should not be used to partition joint risks into exposure-specific effects. [29]

To account for age-related differences in the population, the data is divided based on age groups. The example data has ages in days, so preprocessing the ages into categorical groups is needed.

```
unique_ages <- unique(data[["age"]])
unique_ages <- sort(unique_ages)
print(unique_ages)


data <- data %>%
    mutate(
    Age_group = case_when(
        age/365  < 20 ~ "0-19",
        age/365 >= 20 & age/365  < 30 ~ "20-29",
        age/365  >= 30 & age/365  < 40 ~ "30-39",
        age/365  >= 40 & age/365  < 50 ~ "40-49",
        age/365  >= 50 & age/365  < 60 ~ "50-59",
        age/365  >= 60 & age/365  < 70 ~ "60-59",
        age/365  >= 70 ~ "70+"
    )
    )
```

```
age_groups <- unique(data$Age_group)
```

Listing 4.18: Preprocessing: Creating age groups from age in days

The groups are looped through and PAR is calculated for each subgroup. The PAR values are summed together and each summed PAR is weighted with a value that is relative to the group size.

```
par <- 0
bay_lower_bound <- 0
bay_upper_bound <- 0

for (age_group in age_groups) {
    exposure_col <- "over_weight"
    outcome_col <- "cardio"

    age_data <- data[data$Age_group == age_group, ]
    weight <- nrow(age_data)

    x <- extract_abcd(age_data, exposure_col, outcome_col)
    par <- par + calculate_par(x) * weight

    bay_ci <- calculate_bayesian_ci(
    "par",
    x,
    interval,
    prior,
    sample_count
    )
```

```
}
lower_bound <- bay_lower_bound + bay_ci[1] * weight
upper_bound <- bay_upper_bound + bay_ci[2] * weight


cat("par:", par/nrow(data), "\n")
cat("Bayes␣CI:\n")
print(c(bay_lower_bound/nrow(data),
bay_upper_bound/nrow(data)))
```

Listing 4.19: Manual calculation of age-standardized PAR and Bayesian confidence interval

Output is

```
par: 0.07397856
Bayes CI:
[1] 0.005072823 0.144337473
Bootstrap CI:
[1] 0.003624141 0.148473531
```

Same adjusted calculation can be done with the function *calculate_standardized_bayesian_* where in addition to *exposure_col* and *outcome_col* a *standarisation_col* needs to be specified.

```
# Define parameters
type <- "par"
exposure_col <- "over_weight"
outcome_col <- "cardio"
standarisation_col <- "Age_group"
interval <- 0.95
prior <- c(1, 1, 1, 1)
```

```
sample_count <- 1000


# Call the function
result <- calculate_standardized_bayesian_ci(
  type = type,
  data = data,
  exposure_col = exposure_col,
  outcome_col = outcome_col,
  standarisation_col = standarisation_col,
  interval = interval,
  prior = prior,
  sample_count = sample_count
)


# Print the result
cat("Standardized␣Bayesian␣Confidence␣Interval:\n")
print(result)
```

Listing 4.20: Example: Calculating age-standardized Bayesian confidence interval for PAR using a function from the custom R package

Output:

```
Standardized Bayesian Confidence Interval:
           [,1]
[1,] 0.003972383
[2,] 0.141892267
```

Functions' outputs are made as simple as possible. The user completely defines titles and formatting; the functions themselves only output the results. 1000 sam-

ples were selected randomly from the cardiovascular disease dataset in demo code 4.5.1. PAR is about 10.5%, and the Bayesian confidence interval lower limit is 4.9% and upper limit 13.1%. In this demo the bootstrap interval is slightly shorter than the Bayesian interval. The difference is so small that it's negligible. This is consistent with the simulated evaluation results. The age adjusted PAR is 7.4% and the confidence interval's lower and upper limits are 0.5% and 14.4 for bootstrap and 0.4 and 14.8 for Bootstrap. Both methods produce significantly longer intervals than the simple method that is not using adjustment. Adjusting for age seems to add uncertainty to the inference. Understanding this better would require simulation studies to determine if this is a typical result. In this case, Bootstrap produced a longer interval, which seems interesting because in one variable situations, Bootstrap tends to produce shorter intervals than Bayesian.

Table 4.6 has some more example cases. Instead of overweight (BMI $>$ 24.99) the analysis could be done so that obese is the exposure level. A BMI over 29.99 defines obesity and BMI value is changes in code 4.5.1 to get this. Even though obesity poses a greater threat than being overweight, the overall occurrence of obesity is much less than being overweight, and that can be seen in the interval. The rows where obesity is the exposure threshold have smaller lower and upper bounds than the rows where the exposure level is being overweight. These limited samples also have longer intervals when the confidence interval is constructed with age adjustment. An exception to this is the 70000 sample cases. Here, the sample count is so high that intervals are very short, and adjustment moves the interval, keeping it the same length instead of making it longer.

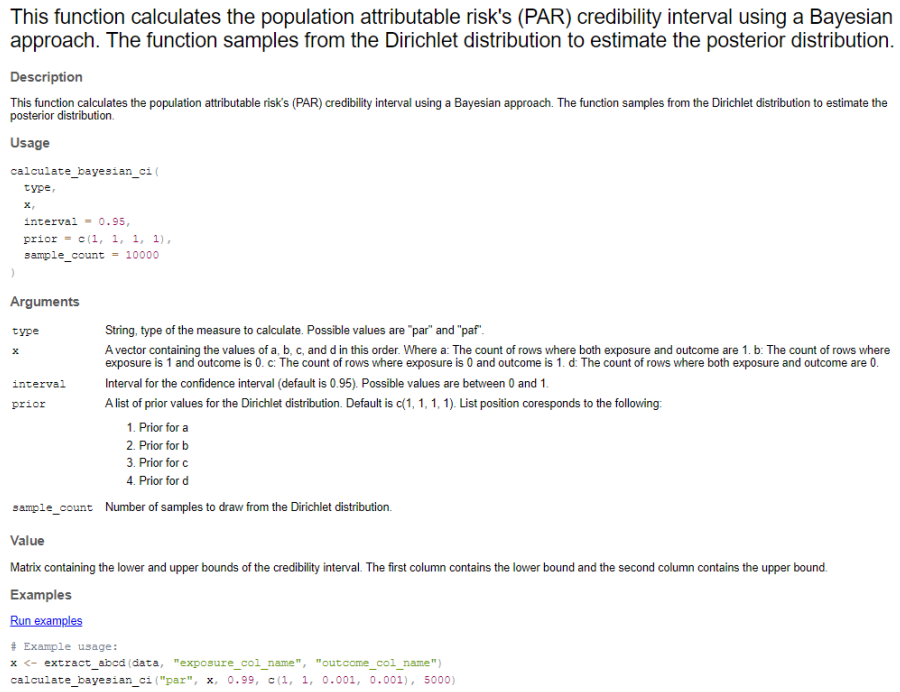
All function documentations are found in the man directory of the https://github.com/peppi-lotta/par GitHub package. The Roxygen-formatted code comments for *calculate_par* are shown in code 4.5.1. Automatically generated documentation looks like figure 4.7 when previewed. The Documentation is lacking. The

Table 4.6: Comparison of Bootstrap and Bayesian Confidence Intervals for PAR

Sample Size	PAR Value	Bayes Interval	Bootstrap Interval	Age-Adj. Bayes Int.	Exposure
64	0.078	[-0.043, 0.191]	[-0.045, 0.209]	[-0.131, 0.253]	Overweight
64	0.068	[-0.004, 0.140]	[0.00, 0.141]	[-0.069, 0.194]	Obese
1000	0.102	[0.063, 0.140]	[0.063, 0.140]	[0.025, 0.160]	Overweight
1000	0.039	[0.021, 0.057]	[0.021, 0.057]	[0.000, 0.063]	Obese
70000	0.102	[0.097, 0.107]	[0.097, 0.106]	[0.082, 0.100]	Overweight
70000	0.047	[0.045, 0.049]	[0.045, 0.049]	[0.039, 0.047]	Obese

title is the same as the summary. The title should be something more concise and clear, for example, the function's name. The name "Value" is not precise enough for the subtitle; a better heading would be "Return Value".

```
#'
#' This function calculates the population
#' attributable risk (PAR)
#' for a binary exposure and outcome.
#' PAR = P(D+) - P(D+| E-)
#' = (a + c)/(a + b + c + d) - (c + d)/c
#' @param x A vector containing the
#' values of a, b, c, and d
#' in this order. Where
#' a: The count of rows where
#' both exposure and outcome are 1.
#' b: The count of rows where
#' exposure is 1 and outcome is 0.
#' c: The count of rows where
#' exposure is 0 and outcome is 1.
#' d: The count of rows where
#'both exposure and outcome are 0.
#'
```



This function calculates the population attributable risk's (PAR) credibility interval using a Bayesian approach. The function samples from the Dirichlet distribution to estimate the posterior distribution.

Description

This function calculates the population attributable risk's (PAR) credibility interval using a Bayesian approach. The function samples from the Dirichlet distribution to estimate the posterior distribution.

Usage

```
calculate_bayesian_ci(
  type,
  x,
  interval = 0.95,
  prior = c(1, 1, 1, 1),
  sample_count = 10000
)
```

Arguments

`type`	String, type of the measure to calculate. Possible values are "par" and "paf".
`x`	A vector containing the values of a, b, c, and d in this order. Where a: The count of rows where both exposure and outcome are 1. b: The count of rows where exposure is 1 and outcome is 0. c: The count of rows where exposure is 0 and outcome is 1. d: The count of rows where both exposure and outcome are 0.
`interval`	Interval for the confidence interval (default is 0.95). Possible values are between 0 and 1.
`prior`	A list of prior values for the Dirichlet distribution. Default is c(1, 1, 1, 1). List position coresponds to the following: 1. Prior for a 2. Prior for b 3. Prior for c 4. Prior for d
`sample_count`	Number of samples to draw from the Dirichlet distribution.

Value

Matrix containing the lower and upper bounds of the credibility interval. The first column contains the lower bound and the second column contains the upper bound.

Examples

Run examples

```
# Example usage:
x <- extract_abcd(data, "exposure_col_name", "outcome_col_name")
calculate_bayesian_ci("par", x, 0.99, c(1, 1, 0.001, 0.001), 5000)
```

Figure 4.7: Preview of generated documentation

```
#' @return The population attributable risk (PAR).
#'
#' @examples
#' # Example usage:
#' x <- c(1, 2, 3, 4)
#' PAR <- calculate_par(x)
#' @export
calculate_par <- function(x)
```

Listing 4.21: R function to calculate the population attributable risk (PAR) from a contingency table vector

5 Conclusion

In today's technological landscape, anyone could provide Pirikahu et al.'s paper to ChatGPT or any large language model and request an R package that performs the described procedures. However, using AI is not beneficial without an understanding of mathematics and the basics of Bayesian inference. The generated code is useless if the user doesn't understand how to use it, what it does, or why it works. While users don't necessarily need to know all these details themselves, they must be confident that the package was written by someone who does. This thesis provides the necessary background to understand the developed package and serves as proof that I possess the knowledge required to write a reliable tool for constructing confidence intervals for PAR and PAF.

The technical objective was to create an R package that is available for future use, transparent in implementation, user-friendly, and well-documented. A side product of this work is a dataset of evaluation results, published alongside the R package. The package requires only a basic knowledge of R from the user. User testing could determine whether the package is truly easy to use. The `par` package is well documented using the `roxygen2` format. This thesis also serves as documentation for the foundational concepts guiding development and explains the intended workflow. Both the workflow and evaluation code are included with the `par` package on GitHub.

Although the original goal was to calculate PAR, the package also includes support for PAF. Several helper functions provide additional functionality beyond just

calculating PAR and constructing intervals. Inputs are kept as simple as possible, allowing users to directly utilize the return values. The package also provides a function for calculating adjusted PAR or PAF, extending the work of Pirikahu et al.

Some improvements could be made to prevent user errors. For instance, function 4.2.2 calculates a value even if there are negative values in the input vector. Negative values are not realistic input values because the inputs represent counts, where the lowest possible value is 0. Future improvements could include validating user input and providing clear error messages upon validation failure. The package is designed for 2x2 contingency tables that consists of four unique cells, and functions expect a vector of four values coresponding to the cells. For example, the function *calculate_bayesian_ci* will not raise an error for vectors longer than four elements; instead, it silently uses the first four. This could confuse users if incorrect input still produces results without warning. The `roxygen2`-generated documentation could be improved slightly. For instance, the title is currently identical to the summary. Title could be made more concise by using the function name instead. Also, the heading "Value" is vague; "Return Value" would be more precise. Nonetheless, `roxygen2` was an excellent choice, streamlining the documentation and development process.

Another limitation is that this work was created solely by one person and published on GitHub. Although the code is publicly available, its continued availability depends on my maintaining the repository. If I remove it, the code will be lost. Further development also depends on my willingness to accept contributions from others. As discussed previously, I have listed the limitations of the package and do not plan to address them after the completion of this thesis. The code is free for others to use and adapt for future development under the MIT license, the most permissive license option. While my package provides a functional solution for calculating PAR and PAF, I do not intend to continue development or fix bug that might surfase when dependencies are not updated. I encourage others to copy or fork the

package and use it as a reference or base for implementing a Bayesian approach.

Initially, the package was intended solely for calculating PAR. However, further research showed that the specific values are less important than the method used to construct the confidence interval. As a result, the R package was designed to include dedicated functions for both PAR and PAF. Additional functions for values derived from contingency tables could be easily added in the future by including a function that outputs a single value. Added function could then be incorporated into the existing framework for both Bayesian and bootstrap confidence interval construction.

The best alternative for calculating PAF that I found is the `AF` package. Its last update was in 2020 and does not include a fully Bayesian approach. Compared to `AF`, the `par` package allows users to explicitly choose between PAR and PAF and is more flexible in permitting users to define their preferred approach. The `AF` package does offer more comprehensive support for adjusted and multivariate analyses than the `par` package.

I implemented Bootstrap in addition to the Bayesian approach to enable comparison with a well-established method. Research shows that Bootstrap is among the best-performing methods, making it an ideal benchmark to evaluate the strengths and limitations of Bayesian methods. The code used for evaluation is included in the `par` package's examples folder, along with the corresponding results. Anyone can reproduce these results or use the data for visualization or comparative analysis.

Pirikahu et al. provide both mathematical and verbal definitions for PAR and PAF but primarily focus on PAR in their paper. In epidemiology, "population attributable risk" and "population attributable fraction" are often used interchangeably. Pirikahu et al. made a deliberate effort to define these concepts clearly. The literature contains as many as 16 different terms for similar concepts, which made the research process challenging. Understanding PAR and PAF is essential

for evidence-based decision-making. I found it particularly frustrating how many relevant papers were behind paywalls, which hindered my research and could similarly impede future development. Lack of open-access literature on constructing confidence intervals limited the scope of my literature review.

Bootstrap performs poorly when either the exposure rate or sample size is low. Its coverage also declines at high exposure rates. In contrast, Bayesian coverage remains stable when a uniform prior is used. Bootstrap tends to produce shorter confidence intervals, making it preferable when its coverage is adequate. With large sample sizes, the difference in interval length becomes negligible. When the exposure rate e is below 0.25 or above 0.75, the Bayesian method is preferred. In the midrange 0.25<e<0.75 0.25<e<0.75, Bootstrap generally performs better. Bayesian methods are also superior in small-sample contexts because they incorporate prior knowledge and better account for uncertainty.

The dichotomous model explored by Pirikahu et al. involves a single binary variable and a binary outcome and is considered simplistic. Extensive literature exists on extending PAF confidence interval construction to multiple exposures and multible levels of exposure. Expanding Bayesian methods to support multivariate analysis and different exposure levels is a logical next step. I implemented an experimental version with an additional risk factor. PAR was calculated for each subgroup defined by the adjustment variable, then weighted by the group's relative population proportion. While this method was informed by the literature, no formal evaluation was conducted. Preliminary results presented in table 4.6 indicate that the adjusted method produced longer intervals in general. With a large sample size (70,000), the interval length was the same as in single-variable case but was slightly shifted lower. This approach used case-count-based weights, which diverge from the Bayesian principle that parameters should be distributions in stead of singe values.

Computational limitations also influenced the work. My personal computer had

just enough resources to evaluate the dichotomous case. Initially, simulations were estimated to take around 300 hours, which I reduced to approximately 40 hours through code optimization, simulation constraints, and parallel processing using all four CPU cores of my machine. Simulating adjusted PAR values with its greater complexity would require significantly more computing power.

References

[1] S. Pirikahu, G. Jones, M. L. Hazelton, and C. Heuer, "Bayesian methods of confidence interval construction for the population attributable risk from cross-sectional studies", *Statistics in Medicine*, vol. 35, pp. 3117–3130, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:3497293.

[2] *roxygen2-basics*, https://r-pkgs.org/man.html#roxygen2-basics/, Accessed: 2024-12-31.

[3] F. M. Giorgi, C. Ceraolo, and D. Mercatelli, "The r language: An engine for bioinformatics and data science", *Life*, vol. 12, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248442073.

[4] R. van de Schoot and S. Depaoli, "Bayesian analyses : Where to start and what to report", *The European health psychologist*, vol. 16, pp. 75–84, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:142633945.

[5] B. Illowsky and S. T. Dean, "Introductory statistics: Openstax", 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:126293670.

[6] L. C. Hespanhol, C. S. Vallio, L. da Cunha Menezes Costa, and B. T. Saragiotto, "Understanding and interpreting confidence and credible intervals around effect estimates.", *Brazilian journal of physical therapy*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:58581702.

[7] I. Fornacon-Wood, H. B. Mistry, C. Johnson-Hart, C. Faivre-Finn, J. P. O'Connor, and G. Price, "Understanding the differences between bayesian and frequentist statistics.", *International journal of radiation oncology, biology, physics*, vol. 112 5, pp. 1076–1082, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247420107.

[8] L. Shi, H. Sun, and P. Bai, "Bayesian confidence interval for difference of the proportions in a 2×2 table with structural zero", *Journal of Applied Statistics*, vol. 36, pp. 483–494, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:119871041.

[9] C. P. Robert, "The bayesian choice : From decision-theoretic foundations to computational implementation", in *The Bayesian choice : from decision-theoretic foundations to computational implementation*, 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:50937448.

[10] A. Gelman, "Bayesian data analysis", in *Bayesian Data Analysis*, 2014.

[11] A. Gut, "Probability: A graduate course", in *Probability: A Graduate Course*, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:117844972.

[12] R. Mcelreath, "Statistical rethinking: A bayesian course with examples in r and stan", in *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*, 2015.

[13] D. Lindley, "The 1988 wald memorial lectures: The present position in bayesian statistics", *Statistical Science*, vol. 5, pp. 44–65, 1990. [Online]. Available: https://api.semanticscholar.org/CorpusID:117797898.

[14] Y. Pawitan, "In all likelihood : Statistical modelling and inference using likelihood", *The Mathematical Gazette*, vol. 86, pp. 375–376, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:117422783.

[15] G. E. P. Box and G. C. Tiao, "Bayesian inference in statistical analysis", *International Statistical Review*, vol. 43, p. 242, 1973. [Online]. Available: https://api.semanticscholar.org/CorpusID:122028907.

[16] M. Sugiyama, "Chapter 17 - bayesian inference", in *Introduction to Statistical Machine Learning*, M. Sugiyama, Ed., Boston: Morgan Kaufmann, 2016, pp. 185–196, ISBN: 978-0-12-802121-7. DOI: https://doi.org/10.1016/B978-0-12-802121-7.00028-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128021217000285.

[17] R. van de Schoot *et al.*, "Bayesian statistics and modelling", *Nature Reviews Methods Primers*, vol. 1, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:268753577.

[18] *Probability Mass Functions*, https://online.stat.psu.edu/stat414/lesson/7/7.2, Accessed: 2024-10-14.

[19] *he Multinomial Distribution*, https://online.stat.psu.edu/stat504/book/export/html/6672, Accessed: 2024-10-14.

[20] S. Sinharay, "Continuous probability distributions", in *International Encyclopedia of Education (Third Edition)*, P. Peterson, E. Baker, and B. McGaw, Eds., Third Edition, Oxford: Elsevier, 2010, pp. 98–102, ISBN: 978-0-08-044894-7. DOI: https://doi.org/10.1016/B978-0-08-044894-7.01720-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080448947017206.

[21] X. Wang and Z. Cheng, "Cross-sectional studies: Strengths, weaknesses, and recommendations.", *Chest*, vol. 158 1S, S65–S71, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220520566.

[22] H. Goldstein, "Multilevel statistical models: Goldstein/multilevel statistical models", 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:156992120.

[23] *Hierarchical (multilevel) models for survey data*, https://www.hcp.med.harvard.edu/statistics/survey-soft/hierarchical.html, Accessed: 2024-12-31.

[24] N. G. Bruce, D. Pope, and D. Stanistreet, "Quantitative methods for health research: A practical interactive guide to epidemiology and statistics", 2008, pp. 199–200. [Online]. Available: https://api.semanticscholar.org/CorpusID:79185595.

[25] K. Drescher and W. Schill, "Attributable risk estimation from case-control data via logistic regression.", *Biometrics*, vol. 47 4, pp. 1247–56, 1991. [Online]. Available: https://api.semanticscholar.org/CorpusID:44991418.

[26] S. Greenland, "Concepts and pitfalls in measuring and interpreting attributable fractions, prevented fractions, and causation probabilities.", *Annals of epidemiology*, vol. 25 3, pp. 155–61, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:23780866.

[27] W. Ahrens and I. Pigeot, "Handbook of epidemiology", 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:161853324.

[28] H. M. Leung and L. L. Kupper, "Comparisons of confidence intervals for attributable risk.", *Biometrics*, vol. 37 2, pp. 293–302, 1981. [Online]. Available: https://api.semanticscholar.org/CorpusID:46619391.

[29] M. D. Maso *et al.*, "Attributable fraction for multiple risk factors: Methods, interpretations, and examples", *Statistical Methods in Medical Research*, vol. 29, pp. 854–865, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:149445588.

[30] S. D. Walter, "The estimation and interpretation of attributable risk in health research.", *Biometrics*, vol. 32 4, pp. 829–49, 1976. [Online]. Available: https://api.semanticscholar.org/CorpusID:37957769.

[31] W. H. Organization, Ed., *Global Health Risks: Mortality and Burden of Disease Attributable to Selected Major Risks*. Albany: World Health Organization, 2009, Available from: ProQuest Ebook Central. Accessed: 2025-03-05.

[32] M. A. Mansournia and D. G. Altman, "Population attributable fraction", *BMJ*, vol. 360, 2018, Published 22 February 2018. DOI: 10.1136/bmj.k757. [Online]. Available: https://www.bmj.com/content/360/bmj.k757.

[33] S. Greenland and J. M. Robins, "Conceptual problems in the definition and interpretation of attributable fractions.", *American journal of epidemiology*, vol. 128 6, pp. 1185–97, 1988. [Online]. Available: https://api.semanticscholar.org/CorpusID:17417604.

[34] A. Lehnert-Batar, A. B. Pfahlberg, and O. Gefeller, "Comparison of confidence intervals for adjusted attributable risk estimates under multinomial sampling", *Biometrical Journal*, vol. 48, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:9751534.

[35] S. Lee *et al.*, "A comparison of green, delta, and monte carlo methods to select an optimal approach for calculating the 95% confidence interval of the population-attributable fraction: Guidance for epidemiological research", *Journal of Preventive Medicine and Public Health*, vol. 57, pp. 499–507, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272567357.

[36] K.-J. Lui, "Notes on interval estimation of the attributable risk in cross-sectional sampling", *Statistics in Medicine*, vol. 20, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:22083806.

[37] E. Dahlqwist, J. Zetterqvist, Y. Pawitan, and A. Sjölander, "Model-based estimation of the attributable fraction for cross-sectional, case–control and cohort studies using the r package af", *European Journal of Epidemiology*, vol. 31, pp. 575–582, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:3482697.

[38] *Compile: Byte code compiler*, https://www.rdocumentation.org/packages/compiler/versions/3.6.2/topics/compile, Accessed: 2025-02-21.

[39] K. Dempsy, *Risk factors for cardiovascular heart disease*, Accessed: 2025-04-27, 2022. [Online]. Available: https://www.kaggle.com/datasets/thedevastator/exploring-risk-factors-for-cardiovascular-diseas/.

[40] Centers for Disease Control and Prevention, *Heart disease and stroke facts*, Accessed: 2025-04-27, 2023. [Online]. Available: https://www.cdc.gov/chronicdisease/resources/publications/factsheets/heart-disease-stroke.htm.