Figuring out the population attributable risk (PAR) with a bayesian approach; single and multivariatete analysis on cross-sectional study data.

Turun yliopisto
Tietotekniikan laitos
TkK-tutkielma
Labra
Joulukuu 2024
Peppi-Lotta Saari

TURUN YLIOPISTO
Tietotekniikan laitos

Peppi-Lotta Saari: Figuring out the population attributable risk (PAR) with a bayesian approach; single and multivariatete analysis on cross-sectional study data.

TkK-tutkielma, 33 s., 2 liites.
Labra
Joulukuu 2024

Here be abstract

Asiasanat: tähän, lista, avainsanoista

Sisällys

1 Introduction

The primary focus of this thesis is a paper written by *Pirikahu et al. (2016)* by the title, *Bayesian methods of confidence interval construction for the population attributable risk from cross-sectional studies.* This paper proposes a fully Bayesian approach to constructing a confidence interval for the population-attributable risk (PAR). The PAR value represents the percentage of cases in a population that would not have occurred had the exposure not taken place [1].

Confidence intervals are typically associated with Frequentist statistics. Bayesian approaches do not directly produce confidence intervals but credibility intervals. We can create a credibility interval with Bayesian approach and use repeated sampling to determine whether the credibility interval exhibits the Frequentist properties of a confidence interval. I provide a more in-depth explanation of the differences between confidence and credibility intervals in sections 2.2.3, 2.2.4, and 2.2.5.

1.1 Task and Goal

As a part of this thesis work, I will create an R package designed to construct a confidence interval for PAR based on the theoretical framework presented in the paper by *Pirikahu et al. (2016).*

Ideally, Bayesian data analysis follows a three-step process, which I will adhere to in this work. The steps are as follows

- Full probability model: Create a model that is consistent with 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.

[2]

The paper by *Pirikahu et al. (2016)* provides a complete probability model for constructing a confidence interval in a single exposure scenario. I intend to implement this model within an R package and demonstrate with a workflow and real data how the code can be used. I aim to create an easy-to-use and efficient R package that researchers can utilize. I'm focusing on thorough documentation to minimize user errors. Documentation is created by adhering to the roxygen2 structure.

Roxygen automatically generates a .Rd file from the comments in the R script without affecting code packageing and use [3]. I've chosen to use Roxygen because it allowes me to maintain documentation alongside the code.

I will evaluate the model and its expansions using simulated data. Evaluation happens by selecting specific parameters with known values and simulating data that correspond to these specified values. The *Pirikahu et al. (2016)* paper offers realistic values commonly employed in epidemiology.

In addition to the R package, another concrete outcome of this work is a data table containing evaluation results. Running evaluation with simulations is very resource-intensive. I have created a structure in the evaluation code that allows me to run in subsections and output the results in a CSV file.

Furthermore, I will explore potential expansions to the model. The model outlined in the paper is limited to single-exposure scenarios. I will extend the model to a hierarchical model. The hierarchical model will consist of two levels, allowing two variables to be taken into account instead of just one when calculating an outcome. I will demonstrate and evaluate how the expansions code works using simulated data.

1.2 Structure

Population-attributable risk describes the potential reduction in disease occurrences if a specific risk factor is eliminated from a population. PAR is a conditional value that can be derived with Bayes' theorem, which provides a mathematical foundation for this thesis work. In the next chapter, chapter **??**, I will begin by discussing Bayes' theorem and its relevance to the *Pirikahu et al. (2016)* paper while providing sufficient context for understanding the underlying theory.

I will outline fundamental statistical terms crucial for later sections of this work. I will also present an overview of Bayesian inference, highlighting key characteristics and concepts such as prior and posterior distributions and likelihood. I assuming the target audience of this thesis are students of Faculty of technology and have limited prior knowledge of statistics.

In chapter 3, I will delve into the model described in by *Pirikahu et al. (2016)*, providing a detailed explanation and the necessary mathematical background. This chapter will also include descriptions of the code I created as part of this thesis, instructions on how to use it, and code for evaluating the model with simulated data. I also discuss the results of the evaluation in this chapter and provide figures and tables to illustrate the outcomes.

In chapter **??**, I discuss expanding the model to a two-level hierarchical model.

1.3 AI Disclaimer

I have utilized ChatGPT versions 3 and 4 to generate ideas for the structure of this thesis. All content has been composed by me from sources I have explicitly cited.

While I found the AI helpful for brainstorming, I did not rely on it to generate the content. I have used Grammarly to enhance grammar and structure sentences in my text.

2 Key Consepts

This chapter will provide an understanding of the Bayesian approach. It will lead the reader through Bayes' theorem to Bayesian inference and data analytics. The meaning of population-attributable risk and relevant concepts to statistics, in general, will be explored in this chapter.

2.1 Bayes' theorem

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

$P(A|B)$ is the conditional probability [4] $P(A|B)$ is the probability that A happens given that B has happened. If A and B are independent events, then $P(A|B) = P(A)$. [4]

A person using 2.1 formula can place values acquired through Frequentist approaches and get a conditional probability; thus, using Bayes' theorem alone is insufficient to make a Bayesian inference.

Bayesian inference is a broader philosophical and statistical approach to statistical inference.[5] In the Bayesian approach, the values of the posterior, prior, and likelihood are not set but random values of a distribution.[4] This allows us to make inferences even when the exact values are unknown and when data is limited.

2.2 Key Statistics consepts

This chapter defines some key statistical concepts that are relevant to the approach defined by *Pirikahu et al. (2016)* and Bayesian inference.

2.2.1 Probability mass function

The probability mass function describes the probability that a discrete random variable is equal to a certain value: $f(x) = P(X = x)$. $f(x)$ over all x equals to 1.[6]

2.2.2 Multinomial distribution

Properties of the multinomial distribution are

- fixed number of trials independent from each other
- samples have a fixed set of outcomes

Probability mass function of the multinomial distribution is:

$$f(x_1, x_2, ..., x_k) = \frac{n!}{x_1!x_2!...x_k!} p_1^{x_1} p_2^{x_2} ... p_k^{x_k} \tag{2.2}$$

, where $n = x_1 + x_2 + ... + x_{k-1} + x_k$[7][8]

In Bayesian analysis, Dirichlet distribution is often used as a prior distribution to model a multinomial distribution. [8] This is considered a standard reference prior.[1]

2.2.3 Confidence intervals

Confidence interval is a frequentist term and it is based on repeated sampling theory.[9] Confidence interval is a range of numbers likely to include the unknown population parameter being estimated.[10] The confidence interval measures the uncertainty of an estimate. The interval has an upper and a lower limit. The true estimate lies within that interval some chosen percent of the time. The width of the confidence interval is determined by two factors: sample size n and standard deviation or standard error of the estimate. It is standard in health sciences to set the interval to be 95%.[11]

Confidence interval is also known as the regression coefficient, a term that describes the relationship between the predictor variable and outcome.[12] For 95% CI, the interval will contain the regression coefficient 95 out of 100 times when repeated.[9]

Confidence interval is differs from having a 95% probability that the regression coefficient is in the interval. In Frequentist statistics, parameters are not assigned probability values. [13] This hower is not the case in Bayesian statistics.

2.2.4 Credibility intervals

Credibility intervals describes a probability that the true value is within the chosen interval. 95% credibility interval means that there is a 95% probability that the true value is in the interval. The Frequentist confidence interval is often misinterpreted this way. And that is why the credibility interval is said to be more intuitive than the confidence interval.[13]

The tail method is a way of constructing the credibility interval. It sets the lower and upper limits by symmetrically cutting the interval between both tails of a distribution. The limit are α / 2 and 1 $-$ α / 2 percentage points of the distribution. If the selected interval is 95%, then α is 0.05. [14]

2.2.5 Bayesian confidence intervals

Bayesian credible interval is a confidence interval if it demonstrates the Frequentist properties of a confidence interval. Whether an interval demonstrates Frequentist characteristics can be observed by calculating the overall percentage coverage for simulated data where various fixed parameters are set. [1] [14] Coverage percentage will show whether the interval is nominal. [1]

2.3 Bayesian Inference

Inference means summary or characterization[15]. It is the process of finding an appropriate model and fitting it to a data set.[2] Statistical inference's goal is to make predictions about an unobserved set of data y based on an already observed set of data x.[16][2] The Bayesian approach to describing this connection is through probability that y happens given that x has happened. If $x = \{x_1, x_2, ..., x_n\}$ and $y = x_{n+1}$ then probability $P(y|x)$ is

$$P(y|x) = P(x_{n+1}|\{x_1, x_2, ..., x_n\}) \tag{2.3}$$

When the amount of observation x_i starts to grow, it becomes increasingly difficult to calculate the effect of each observation on y.

Rormula 2.3 can be simplified to $P(y|\theta)$ where θ is a distribution where x_i are independent and identically distributed, i.e., iid. x_i depends only on θ and not on other x_j.[16] This way of expressing uncertainty via a distribution leans on the concept of exchangeability. Exchangeability is a basic concept of statistical analysis. [2]

There is no operational difference between θ which describes belief and $P(x|\theta)$ which is a measurable quantity. They both describe uncertainty. $P(x|\theta)$ is an updated version of the prior. With prior knowlwdge added the Bayes' theorem becomes:

$$P(x|\theta) = \frac{P(\theta|x)P(x)}{P(\theta)} \tag{2.4}$$

[17]

2.3.1 Prior

Prior distribution is the knowledge we have of the subject matter before we take into account the data.[18] Prior contains the plausibility for all parameters before the data has been taken into account. [5][15] Prior distribution is the best way to summarize information and lack of it.[15]

θ is the prior distribution. θ can be chosen based on some previous inference. Even if no prior distribution is available, a prior must be selected. Almost always, some domain knowledge is available and should be used to help choose the most suitable prior.[5] Regardless of prior knowledge, choosing a prior is often at least partly arbitrary.[15]

Whether the prior is a posterior distribution or a distribution selected by some other means, it doesn't make a difference qualitatively. The prior distribution is a distribution that describes the prior beliefs of the person making the inference.[16] The prior can significantly affect the posterior. Bad priors can lead to misleading posteriors and bad inference.[5] Priors can have a negligible, moderate, or highly noticeable effect on the posterior.[15]

Calculating the posterior becomes easier when the prior is selected from the same family as the posterior. Prior is chosen so that the parametric form of the prior is the same as the posterior's. The prior distribution is conjugate to the likelihood that the prior belongs to the same distribution family as the posterior distribution. [19] The priors mainly used in this thesis are the conjugate priors.

Some other types of priors are maximum entropy prior, parametric approximations, Laplace's prior, Jeffrey's prior, empirical, hierarchical, matching priors, reference priors, and invariant priors. [15]

The prior distribution is not the only distribution that affects the posterior. The likelihood is another factor in calculating the posterior.

2.3.2 Likelihood

Likelihood or likelihood function has meaning outside of the Bayesian approach. Here, I will define likelihood in the context of Bayesian inference. Every time likelihood is mentioned in this thesis work, this definition applies.

Likelihood function contains the information that x_i brings to the inference. [15] Likelihood is derived by forming all data sequences and removing the ones inconsistent with the observed data. It usually is the distribution function assigned to a variable or distribution of variables. [5] The information provided by X_1 about θ is contained in the distribution $l(\theta|x_1)$. When a new observation x_2 is made, it needs to comply with equation

$$l_1(\theta|x_1) = cl_2(\theta|x_2) \tag{2.5}$$

[15].

Equation 2.5 is called the likelihood principle, and it is valid when the θ in both sides of the equation is the same. θ includes every unknown factor of the model. [15]. Bayesian inference obeys the likelihood principle. Two probability models that have the same likelihood function lead to the same inference for θ given a sample of data.[2]

When sample size increases, the meaning of the likelihood increases in relation to it. [5] The prior is modified by the data x trough the likelihood function. It represents the information about θ coming from the data. In 2.5 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 have no effect on the posterior θ. The constant will be canceled by normalizing the right side of 2.6.[18]

2.3.3 Posterior

The simplest definition for a posterior is the probability p conditional on data.[5] Posterior is what we know of the distribution θ given the knowledge of some observed data. As outlined in paragraphs 2.3.1 and 2.3.2, we know that likelihood is the only entity modifying the prior, so we get the simplified version of Bayes rule: *posterior distribution* $\propto$ *likelihood* $\times$ *prior distribution* where,

$$P(\theta|x) \propto cP(x|\theta)P(\theta) \tag{2.6}$$

[18]

Posterior is often mathematically complex, high-dimensional and direct inference is impossible. The number of dimensions is equal to the number of parameters. The posterior function is often an integral that is not solvable analytically, and the posterior distribution can not be evaluated exactly. Sampling the posterior provides a solution for this.[20] Some leading sampling methods, like ones based on Markov chain Monte Carlo (MCMC), only produce samples of the posterior instead of a mathematical function of the posterior.[5]

2.3.4 Posterior Sampling

Posterior sampling can be used to summarize and simulate the posterior. Samples are drawn from a "bucket"where values are present in proportion to their posterior probability. The samples will have the same distribution as the actual posterior. The more samples are drawn, the more exact the distribution will be. Sampling is part of the evaluation phase of Bayesian data analysis. [5]

Summarize

Summarization is divided into three categories

- Intervals of defined boundaries: questions about the frequency of the parameters in the posterior in chosen intervals.
- Intervals of defined mass: questions about confidence and credibility intervals.
- Point estimates: questions about single points in the posterior distribution.

Simulation

Simulations are done to check the model and make predictions. Simulation can be done on a prior distribution to understand it better. Sampling from a known distribution will allow testing of whether a model is working correctly. Simulation can also be used to predict possible future observations. [5]

2.4 PAR, Population attributable risk

Population-attributable risk measures the impact of completely removing a risk factor in a population.

$$PAR = P(D^+) - P(D^+|E^-) \tag{2.7}$$

where 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, given that exposure has not happened from the probability distribution of all disease occurrences. [1]

2.5 Cross-sectional study

A cross-sectional study is a snapshot of a population at a certain point in time. Both the exposure and outcome can be observed at the same time. Individuals for observation are chosen from the population that is relevant for the study. This study method does not consider new cases of the disease that develop over a selected period of time.

Subtypes of cross-sectional studies are

- A Descriptive cross-sectional study: good for evaluating the prevalence of one or more health outcomes in a population.
- Analytical cross-sectional study: measures the prevalence of outcomes end exposures. It's challenging to figure out causal relationships based on 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 good for showing changes in a population over time.

[21]

2.6 Frequentist approach to confidence interval construction for PAR

The Delta method is the standard approach to variance estimation for PAR. The bootstrap method outperforms the delta method in terms of coverage and interval length. [1] As the bootstrap performs better than the delta method, I will compare the Bayesian approach to the bootstrap method.

2.6.1 Bootstrap method

The Bootstrap method approximates the distribution of a statistic by repeated sampling. The samples are drawn from a fitted model or from a dataset with replacement, AKA placing the sample back into the "sample bucket. Drawing from a fitted model is parametric, and drawing from a dataset with replacement is non-parametric.

A contingency table has a cell for each classification. A dataset with one exposure and one outcome can be represented in a 2x2 contingency table with four classifications. 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]

2.7 R language

R is a language and environment for statistical computing and graphics. R is available as Free Software under the Free Software Foundation's GNU General Public License terms in source code form. R is a GNU project. R is an environment where statistical techniques are implemented. It can be extended with the usage of packages. [22]

The base R has functions to perform all major statistical tests, plotting and matrix operations. It is provided as a combination of 14 different core packages: base, compiler, datasets, grDevices, graphics, grid, methods, parallel, splines, stats, stats4, tcltk, tools, and utils. Once installed on a machine, the package can then be loaded with the library() command.

R's functionality can be divided into data interaction, analysis, and result visualization. There are three major repositories for additional R packages: CRAN, Bioconductor, and R-Forge. CRAN package, devtools, provides a convenient function for installing packages directly from a GitHub repository. [23]

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

In this chapter, I will examine the approach proposed in the paper titled *Bayesian Methods for Confidence Interval Construction of Population Attributable Risk from Cross-Sectional Studies* by *Pirikahu et al. (2016)*

3.1 Mathematical Model

Taulukko 3.1: 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

Let n denotes the total sample size, where $a+b+c+d = n$. From the contingency table stucture it is evident that a cross-sectional study incorporating one exposure variable and one disease variable can be characterized as a multinomial distribution with four independent possible outcomes. These outcomes can be described using the multinomial distribution as follows:

$$(a, b, c, d) \sim Multinomial(n, p_{11}, p_{10}, p_{01}, p_{00}) \tag{3.1}$$

- $P(E^+) = \frac{a+b}{n}$: The probability of the exposed in a population.
- $P(E^-) = \frac{c+d}{n}$: The probability of the non-exposed in a population.
- $P(D^+) = \frac{a+c}{n}$: The probability of having the disease in a population.
- $p_{11} = P(D^+ \cap E^+) = P(D^+|E^+)P(E^+) = P(E^+|D^+)P(D^+) = \frac{a}{n}$: The probability of being exposed and having the disease.
- $p_{10} = P(D^- \cap E^+) = P(D^-|E^+)P(E^+) = P(E^+|D^-)P(D^-) = \frac{b}{n}$: The probability of being exposed and not having the disease.
- $p_{01} = P(D^+ \cap E^-) = P(D^+|E^-)P(E^-) = P(E^-|D^+)P(D^+) = \frac{c}{n}$: The probability of not being exposed and having the disease.
- $p_{00} = P(D^- \cap E^-) = P(D^-|E^-)P(E^-) = P(E^-|D^-)P(D^-) = \frac{d}{n}$: The probability of not being exposed and not having the disease.

The population-attributable risk (PAR) refers to the proportion of disease within a population that can be attributed to a specific exposure. The PAR can be calculated using the formula 2.7. By applying Bayes' theorem and by incorporating the values listed in 3.1, we obtain the maximum likelihood estimation function for PAR.

$$\begin{aligned} PAR &= P(D^+) - P(D^+|E^-) \\ &= P(D^+) - \frac{P(E^-|D^+)P(D^+)}{P(E^-)} \\ &= \frac{a+c}{n} - \frac{\frac{c}{n}}{\frac{c+d}{n}} \\ &= \frac{a+c}{n} - \frac{c}{n} \times \frac{n}{c+d} \\ &= \frac{a+c}{n} - \frac{c}{c+d} \\ &= \frac{a+c}{a+b+c+d} - \frac{c}{c+d} \end{aligned} \tag{3.2}$$

A prior distribution that estimates all the situations described in a contingency table is: $\theta = (p_{11}, p_{10}, p_{01}, p_{00})$ Observed values or samples are: $x = (a, b, c, d)$. A probability mass function denotes the likelihood in respect to p_k as

$$f(x|\theta) = \frac{n!}{a!b!c!d!} p_{11}^a p_{10}^b p_{01}^c p_{00}^d \tag{3.3}$$

The posterior distribution is:

$$p(a, b, c, d|p_{11}, p_{10}, p_{01}, p_{00}) = p(\theta|x) \propto f(x|\theta)p(\theta) \tag{3.4}$$

Due to the conjugacy relationship, the posterior can be found analytically in relation to the prior. Posterior is

$$\theta|x \; Dirichlet(a + 1, b + 1, c + 1, d + 1). \tag{3.5}$$

Representing posteriors analytically is computationally less expensive than using MCMC simulation. The confidence interval is a Frequentist concept; however, we can determine the Frequentist coverage of the credibility interval through simulated data.

3.2 R Code

Implementing the model in R is quite straightforward once the underlying mathematical model is grasped. Constructing a confidence interval for a dataset involves four key steps

- Extracting the contingency table values from data
- Generating new contingency tables by simulation
- Calculating the PAR for each simulated table
- Constructing the confidence interval from the simulated PAR values

3.2.1 Extracting the Contingency Table Values from Data

3.2.1 method takes a data frame and extracts the values for a, b, c, and d, returning them in a single vector. a, b, c, and d are the count for the different categories and align with categories given in 3.1. All the function that I've created for this package, that use these category values, are expecting a vector with a, b, c, and d values in this order. I've provided this helper function so that the user can use this an trust that the values from a data set are extracted and saved to the correct order.

```
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
    ))
}
```

3.2.2 Calculate PAR

I've created a function to calculate Population Attributable Risk from contingency table cell values. 3.2.2 function accepts a vector of values and returns the corresponding PAR value. The vector must consist of the values a, b, c, and d in that precise order.

If these values are obtained using the method $extract_abcd$, the vector will be properly sequenced and can be directly passed to the 3.2.2 function.

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

The logic behind calculating par is derived from equation 2.7. In cases where the total number of samples n where $n = a + b + c + d$ is small and the exposure rates $a + b$ are low, the function may return zero values for c and d. Since zero cannot be a divisor, I have opted to handle this situation by returning a value of 0 if either the sum of c and d is zero.

```
if (c + d == 0) {
    return(0)
}
par <- (a + c) / (a + b + c + d) - c / (c + d)
```

The function $calculate_par$ will return a single value, that is, the PAR.

```
calculate_par <- function(x) {
...
    return(par)
}
```

3.2.3 Code for Constructing the Confidence Interval

Similar to the $calculate_par$ function, the $calculate_bayesian_ci$ method requires a vector of values as a parameter, which must be specified by the user. Additionally, this method can accept values for interval coverage, a vector for the prior distribution, and a value for the number of samples; however, these additional parameters are optional and have default settings. The function expects the x and $prior$ vectors to be ordered as a, b, c, and d. The default setting for the number of samples is 10000, which is considered sufficiently large according to *Pirikahu et al. (2016)*. The *prior* defaults to a vector of ones, indicating a non-informative uniform prior. The standard default value for the interval coverage is 0.95, which is commonly used.

```
calculate_bayesian_ci <- function(
    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)
    ...
}
```

3.2.3 is the main logic of the code. Samples of contingency tables are generated using the Dirichlet distribution. 3.2.3 is calling the *rdirichlet* function from the *MCMCpack* package to form new contingency tables and saves them to *samples* variable. *Samples* contains *sample_count* number of tables. With vector operation *apply calculate*$_p$*ar* is applied to each table and we get "*sample_count*"of PAR values. The confidence interval is calculated using the *quantile* function. The function returns a matrix with the lower bound of the confidence interval as the first value and the upper bound as the second value.

```
calculate_bayesian_ci <- function(
...
```

```
    samples <- 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)

    par_samples <- apply(samples, 1, calculate_par)

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

Finally, the function returns a matrix with the lower bound of the confidence interval That are extracted from the quantile function as the first value and the upper bound as the second value.

```
calculate_bayesian_ci <- function
```

```
...
    return(matrix(c(
        confidence_interval[1],
        confidence_interval[2]
    )))
```

Despite the fact that multinomial simulations are generally more efficient than MCMC simulations, conducting evaluations can be resource-intensive. When the *compiler* is loaded, the *compile*$_{a}$*ll* function can be called and all functions within the package are converted from human-readable code to machine code, enhancing execution speed.

The ‘cmpfun‘ function from the Byte Code Compiler can be utilized to compile a function into machine code. This function compiles the body of a closure and returns a new closure with the same formal parameters while replacing the original body with the compiled expression. [24]

```
    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)
    }
```

3.3 Evaluation of the Model

We will run simulation based on selected known values for parameters p, q, e and n to explore performance.

- $p = P(D^+|\ E^+)$, 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.
- n, the total number of samples.

Because exposure either has happened or not, we can deduce that $P(E^-) = 1 - P(E^+) = 1 - e$. And because a person can either have the disease or not, we can deduce that $P(D^-|E^-) = 1 - P(D^+|E^-) = 1 - q$. and $P(D^+|E^+) = 1 - P(D^-|E^+) = 1 - p$. We can use this knowledge to form the probabilities for the different categories

- $a = p_{11} \times n = P(D^+ \cap\ E^+) \times n = P(D^+|\ E^+) \times P(E+) = p \times e \times n$
- $b = p_{10} \times n = P(D^- \cap\ E^+) \times n = P(D^-|\ E^+) \times P(E+) = (1 - p) \times e \times n$
- $c = p_{01} \times n = P(D^+ \cap\ E^-) \times n = P(D^+|\ E^-) \times P(E-) = q \times (1 - e) \times n$
- $d = p_{00} \times n = P(D^- \cap\ E^+) \times n = P(D^-|\ E^-) \times P(E-) = (1 - q) \times (1 - e) \times n$

Rate of decease occurance is $P(D^+)$ and can be calculated as

$$\begin{aligned} P(D^+) &= P(D^+ \cap\ E^+) + P(D^+ \cap\ E^-) \\ &= P(D^+|E^+)P(E^+) + P(D^+|E^-)P(E^-) \\ &= p \times e + q \times (1 - e) \end{aligned} \tag{3.6}$$

Taulukko 3.2: 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 need to generate 10,000 contingency tables that correspond to selected variables p, q, e, and n, using the multinomial distrubution 3.1. The parameter values for the simulation are as follows

We can expand this evaluation matrix to compare different sample sizes

Taulukko 3.3: Parameters for the simulation

n	16	32	64	128	256	512	1024	4096	16384	65536

3.3.1 Code for Evaluating the Model

Table 3.3 provides the parameters for p, q, and e. We need to reverse engineer the probabilities for a, b, c, and d based on the selected parameters. Once we have the probabilities, we can generate contingency tables and construct confidence intervals for PAR employing two different methods: the Bayesian approach proposed by *Pirikahu et al. (2016)* and bootstrap. I loop through all the different combinations of parameters and o the following steps for each combination.

subsubsectionCalculating Probabilities for a, b, c, and d From p, q, e

First we start by calculating the probabilities associated with selected p, q and e. The function *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 function returns a list containing these probabilities in order.

```
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)
  )
 }
```

By utilizing the probabilities for categories and total sample size n, we can simulate contingency tables that align with the chosen parameters p, q, e and n. For each of these tables, we can compute the confidence interval (CI) using the *calculate_bayesian_ci* function.

subsubsectionSimulating Contingency Tables

While *Pirikahu et al. (2016)* gives that 10,000 simulations would be ideal. Due to resource constraints I have, I have reduced the number of simulations to a 1000. Simulated contingency tables are saved to *samples* variable.

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

subsubsectionConstructing the Confidence Interval

The confidence interval is computed for each generated contingency table in *samples*. All of these tables represent the same parameters: p, q, e, and n, and share the same PAR. PAR calculated with the Bayesian method to get a set of confidence intervals

```
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(
    c(a, b, c, d),
    interval,
    prior,
    10000
    )
})
```

Bootstrap method is applied to the same set of samples to get a set of confidence intervals

```
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_bayesian_ci(
    c(a, b, c, d),
    interval,
    10000
    )
    })
```

subsubsectionCalculating Metrics The coverage is considered nominal if the actual PAR falls within the lower and upper bounds of the interval in 95% of the simulations, or at least $1000 * 0.95 = 950$ times. Calculating the actual PAR from p, q and e has to be done so that we can calculate coverage percentage. I have given definitions for p, q, and e in 3.3. Values from 3.3 can be placed into 2.7 to get an equation to calculate PAR from the parameters.

$$\begin{aligned} PAR &= P(D^+) - P(D^+|E^-) \\ &= p * e + q * (1 - e) - q \end{aligned} \tag{3.7}$$

The second row of equation 3.7 can be directly implemented in code. We can calculate the coverage percentage by checking if the actual Par value is with in the upper and lower bounds of the confidence interval.

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

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

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

Coverage percentage and mean interval legth are two metrics that can be used to compare different models. If the coverage percentages of all models meet the nominal criteria, meaning they are equal to or exceed the specified interval value, the model with the narrowest mean length is considered the most effective.

I have calculated coverage precentage and mean interval lenth for both Bayesian and bootstrap methods. I have save the result in a CSV file for further analysis.

subsubsectionCSV File Output

After the steps out lined in previous section. I have generated a CSv file with the following columns

$p, q, e, n, p\_11, p\_10, p\_01, p\_00, actual\_par, bayes\_ci\_mean\_length, bayes\_ci\_covera$

$boot\_ci\_mean\_length$ and $boot\_ci\_coverage$. The file can be found in the data folder of the created R package.

subsubsectionOptimizing the Evaluation Code

The steps I've outlined in paragraph 3.3.1 are computationally expensive. I generate a 1000 contingency tables and constructions of the conffidence interval requires 10000 simulations for Bayes and Bootstrap each. This amounts to $1000 * 10000 * 2 = 20,000,000$ simulations.

The values and calculations that do not require simulations are computed outside of a loop and subsequently outputted into the CSV file. These calculations include the actual PAR and the probabilities p_11, p_10, p_01, p_00. The simulations can then be executed in a loop, utilizing values from the file for each run. This approach allows me to divide the simulation into smaller subsets, enabling me to select a sufficiently large subset to run on my machine.

To further enhance the speed of the simulation, I have implemented several optimizations. For instance, I compile the functions, as demonstrated in 3.2.3. Machine code is faster to run then un compiled code. Compiling can be done by calling the $compile_all$ function.

Additionally, I utilize the *parallel* package to execute the multible samples in parallel, allowing it to leverage the available cores on my 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' represent the starting and ending rows of the file that define the subset to be processed. I am enabling parallel execution with the future package as follows

```
plan(multisession)
results <- future_map(start:end, function(i) {
...
```

3.4 Comparison of Pirikahu et al.'s Fully Bayesian method with Bootstrap Method

Run the eval code and print some figures here fron the CSV file.

3.4.1 Different Priors

Run eval code with different priors and print some figures here.

4 Expanding the Pirikahu Approach

Data pertaining to humans or animals often exhibit hierarchical structures, whether deliberately organized or otherwise, and this aspect should not be overlooked [25]. To enhance the approach introduced by *Pirikahu et al. (2016)*, it is essential to adopt a hierarchical model. This expansion will enable us to calculate the variability in the PAR across different subgroups, allowing for a more precise understanding of how subgroup-specific characteristics contribute to this variability.

4.1 Hierarchical Model

Hierarchical models, commonly referred to as multilevel models, leverage the knowledge gained from previous clusters when addressing new ones. These clusters may consist of individuals, groups, locations, or, in the context of this work, populations. The advantages of multilevel models include improved estimations, as well as the mitigation of the bias caused by over-sampled clusters dominating the inferences. Furthermore, these models implicitly account for variation and better preserve uncertainty, thereby avoiding unnecessary data transformation [5].

Data can exhibit hierarchical, nested, or clustered structures. A hierarchy comprises units organized at varying levels, with groupings occurring even in random formations. The membership of these groups, and vice versa, influences the characteristics of their members [25]. Typically, datasets consist of samples that serve as 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 [26].

4.1.1 2-Level Model

4.2 Code

The code implementation for the basic Pirikahu approach primarily utilizes the rdirichlet function from the MCMC library. However, implementing the multilevel model requires a more specialized approach.

The brms library offers an effective framework for hierarchical models, standing for Bayesian regression models using Stan [27].

5 Conclusion

5.1 Future work

Lähdeluettelo

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

[2] A. Gelman, ”Bayesian Data Analysis”, teoksessa *Bayesian Data Analysis*, 2014.

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

[4] A. Gut, ”Probability: A Graduate Course”, teoksessa *Probability: A Graduate Course*, 2005. url: `https://api.semanticscholar.org/CorpusID:117844972`.

[5] R. Mcelreath, ”Statistical Rethinking: A Bayesian Course with Examples in R and Stan”, teoksessa *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*, 2015.

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

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

[8] S. Sinharay, ”Continuous Probability Distributions”, teoksessa *International Encyclopedia of Education (Third Edition)*, P. Peterson, E. Baker ja B. Mc-

Gaw, toim., Third Edition, Oxford: Elsevier, 2010, s. 98–102, ISBN: 978-0-08-044894-7. DOI: https://doi.org/10.1016/B978-0-08-044894-7.01720-6. url: https://www.sciencedirect.com/science/article/pii/B9780080448947017206.

[9] R. van de Schoot ja S. Depaoli, ”Bayesian analyses : where to start and what to report”, *The European health psychologist*, vol. 16, s. 75–84, 2014. url: https://api.semanticscholar.org/CorpusID:142633945.

[10] B. Illowsky ja S. T. Dean, ”Introductory Statistics: OpenStax”, 2013. url: https://api.semanticscholar.org/CorpusID:126293670.

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

[12] *he Multinomial Distribution*, https://statisticsbyjim.com/glossary/regression-coefficient/, Accessed: 2024-10-14.

[13] I. Fornacon-Wood, H. B. Mistry, C. Johnson-Hart, C. Faivre-Finn, J. P. O'Connor ja G. Price, ”Understanding the Differences Between Bayesian and Frequentist Statistics.”, *International journal of radiation oncology, biology, physics*, vol. 112 5, s. 1076–1082, 2022. url: https://api.semanticscholar.org/CorpusID:247420107.

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

[15] C. P. Robert, ”The Bayesian choice : from decision-theoretic foundations to computational implementation”, teoksessa *The Bayesian choice : from decision-*

theoretic foundations to computational implementation, 2007. url: https://api.semanticscholar.org/CorpusID:50937448.

[16] D. Lindley, ”The 1988 Wald Memorial Lectures: The Present Position in Bayesian Statistics”, *Statistical Science*, vol. 5, s. 44–65, 1990. url: https://api.semanticscholar.org/CorpusID:117797898.

[17] Y. Pawitan, ”In all likelihood : statistical modelling and inference using likelihood”, *The Mathematical Gazette*, vol. 86, s. 375–376, 2002. url: https://api.semanticscholar.org/CorpusID:117422783.

[18] G. E. P. Box ja G. C. Tiao, ”Bayesian inference in statistical analysis”, *International Statistical Review*, vol. 43, s. 242, 1973. url: https://api.semanticscholar.org/CorpusID:122028907.

[19] M. Sugiyama, ”Chapter 17 - Bayesian Inference”, teoksessa *Introduction to Statistical Machine Learning*, M. Sugiyama, toim., Boston: Morgan Kaufmann, 2016, s. 185–196, ISBN: 978-0-12-802121-7. DOI: https://doi.org/10.1016/B978-0-12-802121-7.00028-5. url: https://www.sciencedirect.com/science/article/pii/B9780128021217000285.

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

[21] X. Wang ja Z. Cheng, ”Cross-Sectional Studies: Strengths, Weaknesses, and Recommendations.”, *Chest*, vol. 158 1S, S65–S71, 2020. url: https://api.semanticscholar.org/CorpusID:220520566.

[22] *R web page*, https://www.r-project.org/about.html, Accessed: 2024-11-4.

[23] F. M. Giorgi, C. Ceraolo ja D. Mercatelli, ”The R Language: An Engine for Bioinformatics and Data Science”, *Life*, vol. 12, 2022. url: https://api.semanticscholar.org/CorpusID:248442073.

[24] *compile: Byte Code Compiler*, `https://www.rdocumentation.org/packages/compiler/versions/3.6.2/topics/compile`, Accessed: 2025-02-21.

[25] H. Goldstein, "Multilevel Statistical Models: Goldstein/Multilevel Statistical Models", 2010. url: `https://api.semanticscholar.org/CorpusID:156992120`.

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

[27] *Bayesian regression models using Stan*, `http://paulbuerkner.com/brms/`, Accessed: 2024-12-31.

Liite A Liitedokumentti

Tähän tulee liitteeksi dokumentaatio R koodista, joka on julkaistu käyttöön.

Liitteen ohjelmakoodi 1 kuvaa matemaattisen monadirakenteen pohjalta rakentuvan Haskellin tyyppiluokan. Tyyppiluokan voi nähdä eräänlaisena abstraktina ohjelmointirajapintana (API), joka muodostaa ohjelmoijalle abstraktin ohjelmointikielen käyttöliittymän (UI).

Ohjelmalistaus 1 Tyyppiluokka 'Monad'.

```
{haskell}
class Monad m where
    ( >>= )         :: m a -> (a -> m b) -> m b
    return          :: a                 -> m a

    fail            :: String            -> m a
    (>>)            :: m a -> m b        -> m b
    m >> k          =  m >>= \_ -> k       -- default

instance Monad IO where  ...               -- omitted
```

Ensimmäisen liitteen toinen sivu. Ohjelmalistaus 2 demonstroi vielä monadin käyttöä.

Ohjelmalistaus 2 Monadin käyttöä.

```
{haskell}
main =
return "Your name:" >>=
putStr >>=
\_ -> getLine >>=
\n -> putStrLn ("Hey " ++ n)
```