The Bootstrap

Ashley I Naimi

Spring 2022

Contents

1 Introduction

After obtaining a point estimate, standard practice is to quantify its statistical uncertainty. This quantification is usually accomplished via a confidence interval estimate, which is meant to provide information on a plausible range of point estimate values compatible with the unknown parameter of interest (**?**). There are several ways to estimate confidence intervals of an estimate. The most common technique is to use maximum likelihood theory, which provides an estimate of the standard error of the parameter of interest directly from the likelihood function (known as the observed Fisher information) (Cole et al., 2013, Pawitan (2001),Therneau and Grambsch (2000)), which can be used to obtain confidence intervals. Ninety-five percent confidence intervals are computed by assuming the estimate follows a normal distribution and invoking the empirical rule, which states that 95% of the mass of the estimate's distribution falls within $1.96 \times SE(\hat{\beta})$ units of the point estimate (Wackerly et al., 2002). This approach is based on asymptotic approximations that assume the sample size is sufficiently large.

There are generally three situations in which an alternative is preferred: when an analytic expression for the standard error or confidence interval of an estimator is unknown, or when such an expression exists, but is too complex to implement manually with standard software. For example, marginal standardization (i.e., g computation or the parametric g formula) is increasingly being used in applied research, yet analytic expressions for the standard error of an estimate from marginal standardization are typically intractable (Muller and MacLehose, 2014, Naimi and Whitcomb (2020)). A well-known alternative to variance estimation in such situations is the bootstrap (Efron, 1979).

Several bootstrap estimators exist (Efron and Tibshirani, 1993), however epidemiologists often rely on a few simple versions, namely the normal interval (Wald) bootstrap, and the percentile bootstrap. Furthermore, with the advent of programs for fitting advanced machine learning techniques, researchers are increasingly using the bootstrap to obtain measures of uncertainty for effect estimates obtain from machine learning algorithms (Lee et al., 2010, Oulhote et al. (2019)). Unfortunately, important questions remain as to whether the bootstrap can generally provide "honest" standard errors or confidence intervals[1] when machine learning methods are used (Wasserman, 2006).

[1] Note that "honest" confidence intervals are technically defined as those with a "minimum coverage probability over a rich class of (nonparametric) regression functions with no less than nominal coverage" Li (1989).

Here, we'll illustrate how to use bootstrap confidence interval estimators. We first provide a brief conceptual overview of the procedure, focusing on three particular bootstrap estimators. We then show how the bootstrap can be used to obtain measures of uncertainty when modeling data from the NHEFS data.

2 The Bootstrap

Standard errors and confidence intervals are meant to capture the variation that would be observed in a point estimate in the exact same conditions (same population, sampling scheme, and statistical model) under repeated random sampling. In addition to the approach using the observed Fisher Information, several semi- and non-parametric techniques can be used to obtain standard errors or confidence intervals, including influence function-based methods (Huber and Ronchetti, 2009), balanced repeated replications (Kish and Frankel, 1970), the delta method (van der Vaart, 2000), the jackknife (Tukey, 1958), and the bootstrap (Efron, 1979). The bootstrap is by far the most common alternative. Both parametric and non-parametric bootstrap variance estimators exist (Carpenter and Bithell, 2000). We briefly touch on the logic behind the parametric bootstrap, but focus our attention primarily on the non-parametric bootstrap. We clarify the distinction between the nonparametric bootstrap, and bootstrapping a nonparametric (e.g., machine learning) estimator in a subsequent section.

The bootstrap is a technique that employs the Monte Carlo method and the substitution (or plug-in) principle to obtain standard error or confidence interval estimates (Efron, 2003). The Monte Carlo method is a general approach for solving a broad class of problems using computation to generate random numbers. The Monte Carlo method was introduced in the early 20th century (Metropolis and Ulam, 1949). The basic idea behind this method is to use chance to solve problems that would either be intractable, or just too difficult to solve analytically. Because of it's use of chance, it is named after the famed Monte Carlo casino in Las Vegas, NV.

Technical Note:

To give a simple example of the Monte Carlo method at work, suppose you didn't know, but had to estimate $\pi = 3.1415926\ldots$ Suppose further that all you knew was the following:

$$A_S = L \times W$$

and

$$A_C/4 = (\pi \times r^2)/4$$

If you chose $L = W = r = 1$, you could take the ratio of the area of the quarter circle to the area of the unit square to give:

$$\pi = 4 \times \frac{A_C/4}{A_S}$$

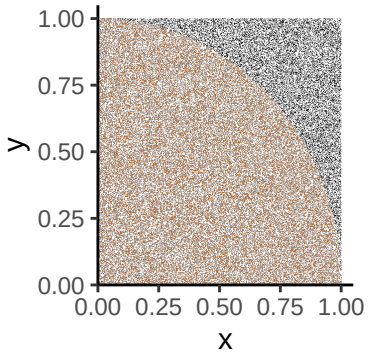
You could then randomly spread points over the entire unit square. Taking four times the proportion of points that fall in the quarter circle relative to the unit square will give you an estimate of π, as with the following code:

```
remotes::install_github("exaexa/scattermore")
library(scattermore)

n <- 1e+05
x <- runif(n, 0, 1)
y <- runif(n, 0, 1)
circle <- (x^2 + y^2 < 1)

plot_dat <- tibble(x, y, circle)

plot_dat %>%
    print(n = 5)
```

```
## # A tibble: 100,000 x 3
##       x     y circle
##   <dbl> <dbl> <lgl>
## 1 0.556 0.749 TRUE
## 2 0.359 0.151 TRUE
## 3 0.678 0.856 FALSE
## 4 0.687 0.690 TRUE
## 5 0.738 0.802 FALSE
## # ... with 99,995 more rows
```

Technical Note (cont.):

```
mc_plot <- ggplot(plot_dat) + geom_scattermore(aes(x,
    y, color = circle)) + scale_x_continuous(expand = c(0,
    0), limits = c(0, 1.05)) + scale_y_continuous(expand = c(0,
    0), limits = c(0, 1.05)) + scale_color_manual(values = c("#000000",
    "#D55E00")) + theme(legend.position = "none")

ggsave(here("figures", "2022_03_02-mc_plot.pdf"),
    plot = mc_plot, width = 5, height = 5,
    units = "cm")
```



We can begin to compute π by computing the ratio of points in the quarter circle to the ratio of points in the square:

```
mean(circle) * 4
```

```
## [1] 3.14256
```

Not π, but not far. To get a more accurate number, we can increase the number of points used to implement the Monte Carlo method:

```
mc_num <- c(100, 1000, 10000, 1e+05, 1e+06,
    1e+07)
```

Technical Note (cont.):

```
pi_estimate <- NULL
for (i in mc_num) {
    n <- i
    x <- runif(n, 0, 1)
    y <- runif(n, 0, 1)
    circle <- (x^2 + y^2 < 1)
    pi_estimate <- rbind(pi_estimate, mean(circle) *
        4)
}

cbind(mc_num, pi_estimate)
```

```
##      mc_num
## [1,]  1e+02 2.920000
## [2,]  1e+03 3.156000
## [3,]  1e+04 3.135200
## [4,]  1e+05 3.146360
## [5,]  1e+06 3.141456
## [6,]  1e+07 3.141798
```

This Monte Carlo method is used for both parametric and non-parametric bootstrap estimators. For the parametric bootstrap, the Monte Carlo method is used to generate residuals from a statistical model defining the relation between the exposure, confounders, and outcome of interest. The estimated parameters from this model are used, along with the exposure and confounder data, and the residuals generated from the Monte Carlo process, to generate a bootstrapped outcome. This process is repeated B times, giving B datasets with a bootstrapped outcome, and the original exposure and confounder data. One then fits a separate model to each of these B datasets to obtain a distribution of bootstrapped parameter estimates that can be use to quantify the uncertainty around the original estimates. In this setting, the bootstrapping process is "parametric' ' in that a parametric (e.g., logistic regression) model is used to generate the bootstrapped outcome data. Violations of the model's

assumptions can lead to biased estimates of statistical uncertainty for the original parameter estimates (Carpenter and Bithell, 2000).[2] Generally, the parametric bootstrap is much less commonly used compared to its counterpart.

[2] While true, the same is true for the nonparametric bootstrap: using a misspecified model can lead to biased variance estimators.

The non-parametric bootstrap uses the random numbers generated from the Monte Carlo process to select random samples with replacement from the original data. The number of bootstrap samples chosen by the researcher is limited only by the available computing power. For each bootstrap sample, one can obtain a "bootstrap replicate' ' of the point estimate for the parameter of interest. With a large enough number of bootstrap replicates, one can use the distribution of bootstrap estimates to obtain information on the degree of uncertainty associated with the point estimate of interest. In effect, one can substitute (or plug-in) the empirical distribution of the estimates from each bootstrap resample for the unknown distribution of the point estimate. For a number of estimators, this empirical distribution can be used to estimate features (such as the standard error or percentiles) of the unknown distribution of the parameter estimate.

This version of the bootstrap is referred to as non-parametric because the data are not assumed to follow a specified parametric model. Rather, they are resampled based on their (nonparametric) empirical distribution (Carpenter and Bithell, 2000). However, this does not imply that the nonparametric bootstrap will work equally well when the model used to generate the parameter estimate of interest is itself nonparametric (e.g., a machine learning based estimator). Indeed, this is the result of the fact that both the parametric and nonparametric bootstrap estimators require that the underlying estimation model meets certain conditions (e.g., regularity, smoothness of the underlying regression function) (Longford, 2008), which are not guaranteed to hold when nonparametric methods are used (Wasserman, 2006).

3 Example Demonstration

Let's use the NHEFS to implement the bootstrap. We'll start by importing the data and, focusing on a few select covariates, we'll estimate the covariate adjusted association between quitting smoking and weight change between 1972 and 1981 (note, the continuous outcome):

```
#' Load relevant packages
packages <- c("broom", "here", "tidyverse",
    "skimr", "rlang", "sandwich", "boot",
    "kableExtra")

for (package in packages) {
    if (!require(package, character.only = T,
        quietly = T)) {
        install.packages(package, repos = "http://lib.stat.cmu.edu/R/CRAN")
    }
}

for (package in packages) {
    library(package, character.only = T)
}

#' Define where the data are
file_loc <- url("https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/1268/20/nhefs.csv")

#' This begins the process of cleaning and formatting the data
nhefs <- read_csv(file_loc) %>%
    select(qsmk, wt82_71, sex, age, race) %>%
    na.omit(.)

factor_names <- c("sex", "race")
nhefs[, factor_names] <- lapply(nhefs[, factor_names],
    factor)

#' Define outcome
nhefs <- nhefs %>%
    mutate(id = row_number(), .before = qsmk)
```

Here are the first 15 rows of these data.

```
#' Quick summary of data
nhefs %>%
    print(n = 15)
```

```
## # A tibble: 1,566 x 6
##       id  qsmk wt82_71 sex     age race
##    <int> <dbl>   <dbl> <fct> <dbl> <fct>
##  1     1     0 -10.1   0        42 1
##  2     2     0   2.60  0        36 0
##  3     3     0   9.41  1        56 1
##  4     4     0   4.99  0        68 1
##  5     5     0   4.99  0        40 0
##  6     6     0   4.42  1        43 1
##  7     7     0  -4.08  1        56 0
##  8     8     0   0.227 1        29 0
##  9     9     0  -2.72  0        51 0
## 10    10     0   9.86  0        43 0
## 11    11     1  15.9   0        43 0
## 12    12     0  -1.82  0        34 0
## 13    13     0   0.562 1        54 0
## 14    14     0   0.110 1        51 1
## 15    15     1 -18.9   1        71 0
## # ... with 1,551 more rows
```

We can use these data to estimate the adjusted mean difference for the association between `qsmk` and `wt82_71`:

```
mod_obj <- lm(wt82_71 ~ qsmk + race + sex +
    age, data = nhefs)

summary(mod_obj)
```

```
##
## Call:
## lm(formula = wt82_71 ~ qsmk + race + sex + age, data = nhefs)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.954  -3.904  -0.033   4.150  46.044
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.22138    0.76368  12.075  < 2e-16 ***
## qsmk         3.06123    0.44255   6.917 6.69e-12 ***
## race1       -0.06851    0.56731  -0.121    0.904
## sex1        -0.38660    0.38397  -1.007    0.314
## age         -0.16407    0.01609 -10.199  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.563 on 1561 degrees of freedom
## Multiple R-squared:  0.08129,    Adjusted R-squared:  0.07894
## F-statistic: 34.53 on 4 and 1561 DF,  p-value: < 2.2e-16
```

Our goal now is to quanfity a standard error for the association between `qsmk` and `wt82_71`. We can use the simple bootstrap to do this. The bootstrap relies on a random resample of these data, with replacement. One simple way to do this is with the `sample` function in R:

```
# create an index for the nhefs data
index <- sample(1:nrow(nhefs), nrow(nhefs),
    replace = T)

# first 10 indices
index[1:10]
```

```
##  [1]  638  190  309  142  449 1237 1092  332  502 1037
```

```
# use index to resample nhefs with
# replacement
nhefs_resample <- nhefs[index, ]
```

```
# look at the resample data
nhefs_resample %>%
    arrange(id) %>%
    print(n = 15)
```

```
## # A tibble: 1,566 x 6
##       id  qsmk wt82_71 sex     age race
##    <int> <dbl>   <dbl> <fct> <dbl> <fct>
##  1     1     0  -10.1  0        42 1
##  2     1     0  -10.1  0        42 1
##  3     1     0  -10.1  0        42 1
##  4     1     0  -10.1  0        42 1
##  5     2     0    2.60 0        36 0
##  6     2     0    2.60 0        36 0
##  7     3     0    9.41 1        56 1
##  8     3     0    9.41 1        56 1
##  9     5     0    4.99 0        40 0
## 10     5     0    4.99 0        40 0
## 11     6     0    4.42 1        43 1
## 12     7     0   -4.08 1        56 0
## 13     7     0   -4.08 1        56 0
## 14    10     0    9.86 0        43 0
## 15    12     0   -1.82 0        34 0
## # ... with 1,551 more rows
```

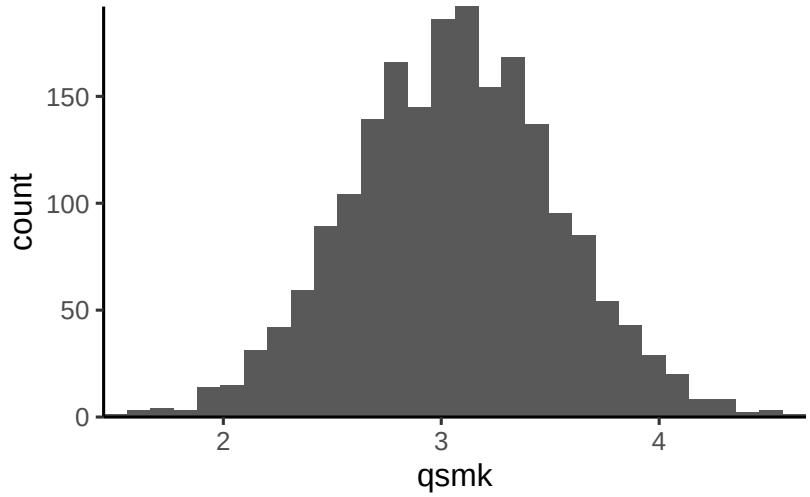
Note that, in the above output, there are several IDs that are repeated. Continually resampling these data with replacement yields a large number of datasets, each of which provides a variation of the original point estimate. This variation depends on the underlying variation in the data, and can thus be used to quantify the sampling variation of the point estimate of interest. In the simplest procedure, we can use a `for` loop to construct the bootstrap:

```
replicates <- 2000
bootstrap_estimates <- NULL
```

```
for (i in 1:replicates) {

    # set the seed, so we get a
    # different sample each time
    set.seed(i)

    # create an index for the nhefs
    # data
    index <- sample(1:nrow(nhefs), nrow(nhefs),
        replace = T)

    # first 10 indices
    index[1:10]

    # use index to resample nhefs with
    # replacement
    nhefs_resample <- nhefs[index, ]

    # estimate the association in the
    # resample
    mod_obj_boot <- lm(wt82_71 ~ qsmk + race +
        sex + age, data = nhefs_resample)

    bootstrap_estimates <- rbind(bootstrap_estimates,
        coef(mod_obj_boot)[2])

}
```

We can look at the distribution of the estimates to get a sense of variability:

```
bootstrap_estimates <- data.frame(bootstrap_estimates)

ggplot(bootstrap_estimates) + geom_histogram(aes(qsmk)) +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0))
```



The simplest way to use the bootstrap would be to obtain the standard deviation of this distribution, and use it as the standard error of the point estimate for the association between `qsmk` and `wt82_71`:

```
se_estimate <- sd(bootstrap_estimates$qsmk)

lcl_qsmk <- coef(mod_obj)[2] - 1.96 * se_estimate
ucl_qsmk <- coef(mod_obj)[2] + 1.96 * se_estimate

coef(mod_obj)[2]
```

```
##     qsmk
## 3.061233
```

```
lcl_qsmk
```

```
##     qsmk
## 2.135977
```

```
ucl_qsmk
```

```
##    qsmk
## 3.98649
```

While many different types of nonparametric bootstrap confidence interval estimators exist, the above features comprise the central characteristics of the approach. Different bootstrap estimators are distinguished by what information is extracted (and how) from the sample of bootstrap replicates $\{\hat{\psi}_1^*, \hat{\psi}_2^*, \ldots, \hat{\psi}_B^*\}$. Three bootstrap confidence interval estimators are arguably best suited to epidemiologic research due to their relative ease of implementation, established theoretical properties, and robustness to violations of a varying range of assumptions (Greenland, 2004). These are the Wald (or normal-interval estimator), percentile, and Bias-Corrected and Accelerated (BC_a) bootstrap estimators.

The Wald, or normal interval, bootstrap estimator is one of the simplest of bootstrap confidence interval estimators. The approach requires the assumption that the parameter estimator $\hat{\psi}$ is normally distributed. After obtaining a sample of bootstrap estimates as outlined above, one can implement the method by simply estimating the standard deviation of this sample. For the example data with marginal standardization, 95% Wald-type bootstrap confidence intervals for the estimated odds ratio $\exp(\hat{\psi})$ can be obtained as $\exp\{\hat{\psi} \pm 1.96 \times SD(\hat{\psi}^*)\}$, where $SD(\hat{\psi}^*)$ is the standard deviation of the bootstrap replicates (values other than 1.96 can be used for different α-level confidence intervals). The standard deviation of the distribution of bootstrap replicates $SD(\hat{\psi}^*)$ can be used to quantify the standard error of the point estimate $SE(\hat{\psi})$ (Altman and Bland, 2005) The use of $\pm 1.96 \times SE(\hat{\psi})$ assumes that $\hat{\psi}$ follows a normal distribution, and is only valid in reasonably large samples. As a consequence, Wald-type bootstrap confidence intervals will usually have a less than nominal coverage probability in small samples (Efron and Tibshirani, 1993, DiCiccio and Efron (1996)). Finally, Efron suggests that between 50 and 200 replicates is sufficient to obtain a good estimate of $SE(\hat{\psi})$ [Efron and Tibshirani (1993)}, however with modern computing implementing more resamples becomes a trivial exercise.

The percentile bootstrap estimator is as simple to implement as the Wald estimator, but does not require the assumption that $\hat{\beta}$ follows a normal distribution. To obtain two-sided percentile bootstrap confidence intervals, one simply selects the bootstrap replicate (i.e., the estimate based on bootstrap resample) corresponding to the $100 \times \alpha/2$ and $100 \times (1 - \alpha/2)$ percentile of the distribution of bootstrap replicates. For example, with 2,000 bootstrap

replicates $\{\hat{\beta}_1^*, \hat{\beta}_2^*, \ldots, \hat{\beta}_{2,000}^*\}$ and a nominal coverage of 95%, the 2.5th and 97.5th percentile points representing the lower and upper confidence limits would correspond to $\hat{\beta}_{50}^*$ and $\hat{\beta}_{1,950}^*$, respectively. The percentile method is both *transformation* and *range respecting*: for example, percentile confidence interval estimates can be obtained on either the log-scale and transformed to the exponential scale or vice versa (Efron and Tibshirani, 1993). Moreover, they respect the boundedness of the estimator in that they will not provide confidence interval estimates that fall outside of the allowable range of the parameter estimate. Percentile intervals (when obtained using the non-parametric bootstrap) are completely non-parametric. As a consequence, this method is subject to anti-conservative properties in that its coverage probability is usually less than nominal (DiCiccio and Efron, 1996, Greenland (2004),Efron and Tibshirani (1993),Carpenter and Bithell (2000)). Finally, because percentile-based methods require estimates of the tails of the distribution of bootstrap replicates, more bootstrap resamples are required than what is technically required for the normal-interval bootstrap. Although the specific number may depend on the scenario, 1,000 to 2,000 re-samples is often seen in practice.

Early recognition of the poor coverage probabilities of the Wald-type and percentile bootstrap confidence intervals led to two modifications of the percentile method (DiCiccio and Efron, 1996). The resulting "bias-corrected and accelerated'' confidence intervals are meant to improve the performance of the percentile confidence interval estimator. The BC_a confidence interval estimator is a percentile-based estimator in that the confidence interval end-points selected are percentiles of distribution of bootstrap replicates. This interval estimator is also transformation and range respecting. It differs from the standard percentile method in that the percentiles corresponding to the upper and lower interval estimates are chosen as a function of a bias correction factor and an acceleration factor that are determined by the data. The bias-correction factor is meant to account for the discrepancy between the median of the sample of bootstrap replicates and the point estimate. This factor can be calculated as a function of the total number of bootstrap replicates less than the point estimate obtained from the original data (Efron and Tibshirani, 1993). The acceleration factor is meant to compensate for possible heterogeneity in the standard error of the estimator as a function of the true parameter value, and can be calculated using the jackknife procedure (Tukey, 1958). One important

feature to note is that the BC_a estimator requires that the number of resamples is no smaller than the sample size, due to the need to estimate the acceleration factor.

References

Douglas G Altman and J Martin Bland. Standard deviations and standard errors. *BMJ*, 331(7521):903, 2005.

James Carpenter and John Bithell. Bootstrap confidence intervals: when, which, what? a practical guide for medical statisticians. *Stat Med*, 19(9): 1141—1164, 2000.

Stephen R Cole, Haitao Chu, and Sander Greenland. Maximum likelihood, profile likelihood, and penalized likelihood: A primer. *Am J Epidemiol*, 179(2): 252–260, 2013.

T.J. DiCiccio and B. Efron. Bootstrap confidence intervals. *Stat Sci*, 11(3): 189—212, 1996.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

Bradley Efron. Second thoughts on the bootstrap. *Stat Sci*, 18(2):135—140, 2003.

Bradley Efron and Robert Tibshirani. *Introduction to the Bootstrap*. Chapman & Hall/CRC, Boca Raton, FL, 1993.

Sander Greenland. Interval estimation by simulation as an alternative to and extension of confidence intervals. *Int J Epidemiol*, 33(6):1389–1397, 2004.

Peter J. Huber and Elvezio. Ronchetti. *Robust statistics*. Wiley, Hoboken, N.J., 2009.

Leslie Kish and Martin R. Frankel. Balanced repeated replications for standard errors. *J Am Stat Assoc*, 65(331):1071—1094, 1970.

Brian K. Lee, Justin Lessler, and Elizabeth A. Stuart. Improving propensity score weighting using machine learning. *Stat Med*, 29(3):337–346, 2010.

Ker-Chau Li. Honest confidence regions for nonparametric regression. *The Annals of Statistics*, 17(3):1001–1008, 1989.

NT Longford. *Studying Human Populations: An Advanced Course in Statistics*. Springer, New York, 2008.

N Metropolis and S Ulam. The Monte Carlo method. *J Am Stat Assoc*, 44(247): 335–341, 1949.

Clemma J Muller and Richard F MacLehose. Estimating predicted probabilities from logistic regression: different methods correspond to different target populations. *Int J Epidemiol*, 43(3):962–970, 2014.

Ashley I Naimi and Brian W Whitcomb. Estimating risk ratios and risk differences using regression. *American Journal of Epidemiology*, 189(6):508–510, 2020.

Youssef Oulhote, Brent Coull, Marie-Abele Bind, Frodi Debes, Flemming Nielsen, Ibon Tamayo, Pal Weihe, and Philippe Grandjean. Joint and independent neurotoxic effects of early life exposures to a chemical mixture: A multipollutant approach combining ensemble learning and g-computation. *Environmental Epidemiology*, 3(5):e063, 2019.

Yudi. Pawitan. *In all likelihood : statistical modelling and inference using likelihood*. Clarendon Press ; Oxford University Press, Oxford; New York, 2001.

Terry M. Therneau and Patricia M. Grambsch. *Modeling survival data : extending the Cox model*. Springer, New York, 2000.

J.W. Tukey. Bias and confidence in not quite large samples. *Annals of Mathematical Statistics*, 29(2):614–, 1958.

A. W. van der Vaart. *Asymptotic statistics*. Cambridge University Press, Cambridge, 2000.

Dennis D. Wackerly, William. Mendenhall, and Richard L. Scheaffer. *Mathematical statistics with applications*. Duxbury, Pacific Grove, CA, 2002.

Larry Wasserman. *All of nonparametric statistics*. Springer, New York; London, 2006.