Statistical Tools for Causal Inference

The SKY Community

2019-08-23

Contents

Introduction

Tools of causal inference are the basic statistical building block behind most scientific results. It is thus extremely useful to have an open source collectively aggreed upon resource presenting and assessing them, as well as listing the current unresolved issues. The content of this book covers the basic theoretical knowledge and technical skills required for implementing staistical methods of causal inference. This means:

- Understanding of the basic language to encode causality,
- Knowledge of the fundamental problems of inference and the biases of intuitive estimators,
- Understanding of how econometric methods recover treatment effects,
- Ability to compute these estimators along with an estimate of their precision using the statistical software R.

This book is geared for teaching causal inference to graduate students that want to apply statistical tools of causal inference. The demonstration of theoretical results are provided, but the final goal is not to have students reproduce them, but mostly to enable them to grasp a better understanding of the fundations for the tools that they will be using. The focus is on understanding the issues and solutions more than understanding the maths that are behind, even though the maths are there and are used to convey the notions rigorously. All the notions and estimators are introduced using a numerical example and simulations, so that each notion is illustrated and appears more intuitive to the students. The second version of this book will contain examples using real applications. The third version will contain exercises.

This book is written in Rmarkdown using the bookdown package. It is available both as a web-book and as a pdf book.

This book is a collaborative effort that is part of the Social Science Knowledge Accumulation Initiative (SKY). The code behind this book is publically available on GitHub and you can propose corrections and updates. How to make contributions to this book is explained on the SKY website. Do not hesitate to make suggestions, modifications and extensions. This way this book will grow and become the living open source collaborative reference for methodological work that it could be.

Part I

The Two Fundamental Problems of Inference

Introduction: the Two Fundamental Problems of Inference

When trying to estimate the effect of a program on an outcome, we face two very important and difficult problems: the Fundamental Problem of Causal Inference (FPCI) and the Fundamental Problem of Statistical Inference (FPSI).

In its most basic form, the FPCI states that our causal parameter of interest (TT, short for Treatment on the Treated, that we will define shortly) is fundamentally unobservable, even when the sample size is infinite. The main reason for that is that one component of TT, the outcome of the treated had they not received the program, remains unobservable. We call this outcome a counterfactual outcome. The FPCI is a very dispiriting result, and is actually the basis for all of the statistical methods of causal inference. All of these methods try to find ways to estimate the counterfactual by using observable quantities that hopefully approximate it as well as possible. Most people, including us but also policymakers, generally rely on intuitive quantities in order to generate the counterfactual (the individuals without the program or the individuals before the program was implemented). Unfortunately, these approximations are generally very crude, and the resulting estimators of TT are generally biased, sometimes severely.

The Fundamental Problem of Statistical Inference (FPSI) states that, even if we have an estimator E that identifies TT in the population, we cannot observe E because we only have access to a finite sample of the population. The only thing that we can form from the sample is a sample equivalent $\hat{E}$ to the population quantity E, and $\hat{E} \neq E$. Why is $\hat{E} \neq E$? Because a finite sample is never perfectly representative of the population. What can we do to deal with the FPSI? I am going to argue that there are mainly two things that we might want to do: estimating the extent of sampling noise and decreasing sampling noise.

Chapter 1

The Fundamental Problem of Causal Inference

In order to state the FPCI, we are going to describe the basic language to encode causality set up by Rubin, and named Rubin Causal Model (RCM). RCM being about partly observed random variables, it is hard to make these notions concrete with real data. That's why we are going to use simulations from a simple model in order to make it clear how these variables are generated. The second virtue of this model is that it is going to make it clear the source of selection into the treatment. This is going to be useful when understanding biases of intuitive comparisons, but also to discuss the methods of causal inference. A third virtue of this approach is that it makes clear the connexion between the treatment effects literature and models. Finally, a fourth reason that it is useful is that it is going to give us a source of sampling variation that we are going to use to visualize and explore the properties of our estimators.

I use X_i to denote random variable X all along the notes. I assume that we have access to a sample of N observations indexed by $i \in \{1, \ldots, N\}$. ''i'' will denote the basic sampling units when we are in a sample, and a basic element of the probability space when we are in populations. Introducing rigorous measure-theoretic notations for the population is feasible but is not necessary for comprehension.

When the sample size is infinite, we say that we have a population. A population is a very useful fiction for two reasons. First, in a population, there is no sampling noise: we observe an infinite amount of observations, and our estimators are infinitely precise. This is useful to study phenomena independently of sampling noise. For example, it is in general easier to prove that an estimator is equal to TT under some conditions in the population. Second, we are most of the time much more interested in estimating the values of parameters in the population rather than in the sample. The population parameter, independent of sampling noise, gives a much better idea of the causal parameter for the population of interest than the parameter in the sample. In general, the estimator for both quantities will be the same, but the estimators for the effetc of sampling noise on these estimators will differ. Sampling noise for the population parameter will generally be larger, since it is affected by another source of variability (sample choice).

1.1 Rubin Causal Model

The RCM is made of three distinct building blocks: a treatment allocation rule, that decides who receives the treatment; potential outcomes, that measure how each individual reacts to the treatment; the switching equation that relates potential outcomes to observed outcomes through the allocation rule.

1.1.1   Treatment allocation rule

The first building block of the RCM is the treatment allocation rule. Throughout this class, we are going to be interested in inferring the causal effect of only one treatment with respect to a control condition. Extensions to multi-valued treatments are in general self-explanatory.

In the RCM, treatment allocation is captured by the variable D_i. $D_i = 1$ if unit i receives the treatment and $D_i = 0$ if unit i does not receive the treatment and thus remains in the control condition.

The treatment allocation rule is critical for several reasons. First, because it switches the treatment on or off for each unit, it is going to be at the source of the FPCI. Second, the specific properties of the treatment allocatoin rule are going to matter for the feasibility and bias of the various econometric methods that we are going to study.

Let's take a few examples of allocation rules. These allocation rules are just examples. They do not cover the space of all possible allocation rules. They are especially useful as concrete devices to understand the sources of biases and the nature of the allocation rule. In reality, there exists even more complex allocation rules (awareness, eligibility, application, acceptance, active participation). Awareness seems especially important for program participation and has only been tackled recently by economists.

First, some notation. Let's imagine a treatment that is given to individuals. Whether each individual receives the treatment partly depends on the level of her outcome before receiving the treatment. Let's denote this variable Y_i^B, with B standing for "Before". It can be the health status assessed by a professional before deciding to give a drug to a patient. It can be the poverty level of a household used to assess its eligibilty to a cash transfer program.

1.1.1.1   Sharp cutoff rule

The sharp cutoff rule means that everyone below some threshold $\bar{Y}$ is going to receive the treatment. Everyone whose outcome before the treatment lies above $\bar{Y}$ does not receive the treatment. Such rules can be found in reality in a lot of situations. They might be generated by administrative rules. One very simple way to model this rule is as follows:

$$D_i = \mathbb{1}[Y_i^B \leq \bar{Y}], \tag{1.1}$$

where $\mathbb{1}[A]$ is the indicator function, taking value 1 when A is true and 0 otherwise.

Example 1.1 (Sharp cutoff rule). Imagine that $Y_i^B = \exp(y_i^B)$, with $y_i^B = \mu_i + U_i^B$, $\mu_i \sim \mathcal{N}(\bar{\mu}, \sigma_\mu^2)$ and $U_i^B \sim \mathcal{N}(0, \sigma_U^2)$. Now, let's choose some values for these parameters so that we can generate a sample of individuals and allocate the treatment among them. I'm going to switch to R for that.

```
param <- c(8,.5,.28,1500)
names(param) <- c("barmu","sigma2mu","sigma2U","barY")
param
```

```
##    barmu sigma2mu  sigma2U     barY
##     8.00     0.50     0.28  1500.00
```

Now, I have choosen values for the parameters in my model. For example, $\bar{\mu} = 8$ and $\bar{Y} = 1500$. What remains to be done is to generate Y_i^B and then D_i. For this, I have to choose a sample size ($N = 1000$) and then generate the shocks from a normal.

```
# for reproducibility, I choose a seed that will give me the same random sample each time I run the pro
set.seed(1234)
N <-1000
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
```
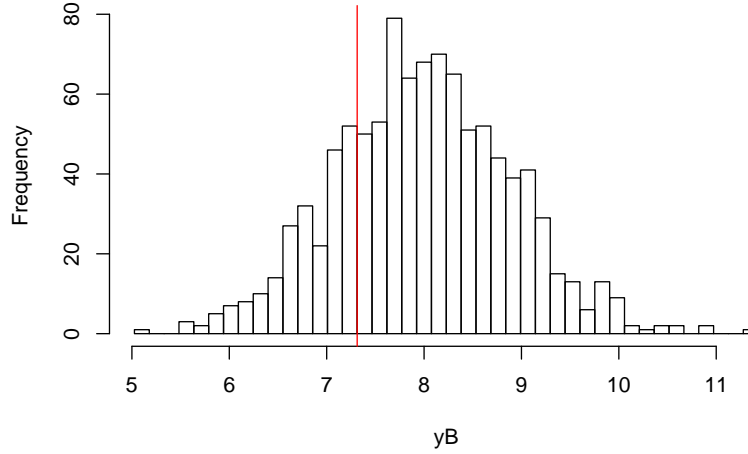
Figure 1.1: Histogram of y_B

Table 1.1: Treatment allocation with sharp cutoff rule

0	771
1	229

```r
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- ifelse(YB<=param["barY"],1,0)
```

Let's now build a histogram of the data that we have just generated.

```r
# building histogram of yB with cutoff point at ybar
# Number of steps
Nsteps.1 <- 15
#step width
step.1 <- (log(param["barY"])-min(yB[Ds==1]))/Nsteps.1
Nsteps.0 <- (-log(param["barY"])+max(yB[Ds==0]))/step.1
breaks <- cumsum(c(min(yB[Ds==1]),c(rep(step.1,Nsteps.1+Nsteps.0+1))))
hist(yB,breaks=breaks,main="")
abline(v=log(param["barY"]),col="red")
```

You can see on Figure 1.1 a histogram of y_i^B with the red line indicating the cutoff point: $\bar{y} = \ln(\bar{Y}) = 7.3$. All the observations below the red line are treated according to the sharp rule while all the one located above are not. In order to see how many observations eventually receive the treatment with this allocation rule, let's build a contingency table.

```r
table.D.sharp <- as.matrix(table(Ds))
knitr::kable(table.D.sharp,caption='Treatment allocation with sharp cutoff rule',booktabs=TRUE)
```

We can see on Table 1.1 that there are 229 treated observations.

1.1.1.2 Fuzzy cutoff rule

This rule is less sharp than the sharp cutoff rule. Here, other criteria than Y_i^B enter into the decision to allocate the treatment. The doctor might measure the health status of a patient following official guidelines, but he might also measure other factors that will also influence his decision of giving the drug to the patient.

The officials administering a program might measure the official income level of a household, but they might also consider other features of the household situation when deciding to enroll the household into the program or not. If these additional criteria are unobserved to the econometrician, then we have the fuzzy cutoff rule. A very simple way to model this rule is as follows:

$$D_i = \mathbb{1}[Y_i^B + V_i \leq \bar{Y}], \tag{1.2}$$

where V_i is a random variable unobserved to the econometrician and standing for the other influences that might drive the allocation of the treatment. V_i is distributed according to a, for the moment, unspecified cumulative distribution function F_V. When V_i is degenerate (*i.e.* it has only one point of support: it is a constant), the fuzzy cutoff rule becomes the sharp cutoff rule.

1.1.1.3 Eligibility + self-selection rule

It is also possible that households, once they have been made eligible to the treatment, can decide whether they want to receive it or not. A patient might be able to refuse the drug that the doctor suggests she should take. A household might refuse to participate in a cash transfer program to which it has been made eligible. Not all programs have this feature, but most of them have some room for decisions by the agents themselves of whether they want to receive the treatment or not. One simple way to model this rule is as follows:

$$D_i = \mathbb{1}[D_i^* \geq 0]E_i, \tag{1.3}$$

where D_i^* is individual i's valuation of the treatment and E_i is whether or not she is deemed eligible for the treatment. E_i might be choosen according to the sharp cutoff rule of to the fuzzy cutoff rule, or to any other eligibility rule. We will be more explicit about D_i^* in what follows.

SIMULATIONS ARE MISSING FOR THESE LAST TWO RULES

1.1.2 Potential outcomes

The second main building block of the RCM are potential outcomes. Let's say that we are interested in the effect of a treatment on an outcome Y. Each unit i can thus be in two potential states: treated or non treated. Before the allocation of the treatment is decided, both of these states are feasible for each unit.

Definition 1.1 (Potential outcomes). For each unit i, we define two potential outcomes:

- Y_i^1: the outcome that unit i is going to have if it receives the treatment,
- Y_i^0: the outcome that unit i is going to have if it **does not** receive the treatment.

Example 1.2. Let's choose functional forms for our potential outcomes. For simplicity, all lower case letters will denote log outcomes. $y_i^0 = \mu_i + \delta + U_i^0$, with δ a time shock common to all the observations and $U_i^0 = \rho U_i^B + \epsilon_i$, with $|\rho| < 1$. In the absence of the treatment, part of the shocks U_i^B that the individuals experienced in the previous period persist, while some part vanish. $y_i^1 = y_i^0 + \bar{\alpha} + \theta \mu_i + \eta_i$. In order to generate the potential outcomes, one has to define the laws for the shocks and to choose parameter values. Let's assume that $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ and $\eta_i \sim \mathcal{N}(0, \sigma_\eta^2)$. Now let's choose some parameter values:

```
l <- length(param)
param <- c(param,0.9,0.01,0.05,0.05,0.05,0.1)
names(param)[(l+1):length(param)] <- c("rho","theta","sigma2epsilon","sigma2eta","delta","baralpha")
param
```

```
##          barmu      sigma2mu       sigma2U           barY           rho
##           8.00          0.50          0.28        1500.00          0.90
```
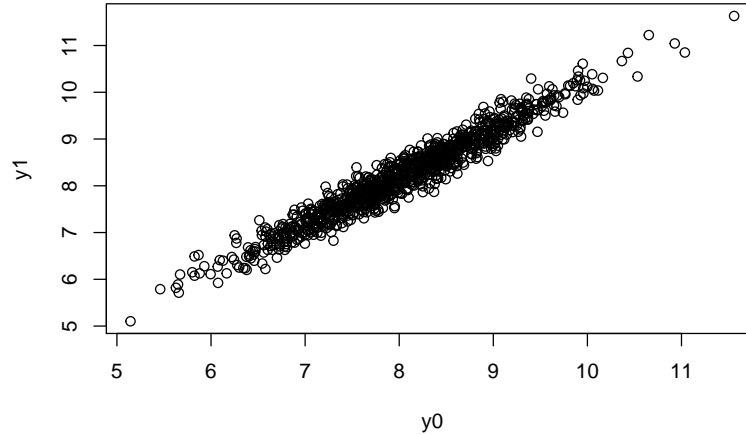
Figure 1.2: Potential outcomes

```
##          theta sigma2epsilon    sigma2eta        delta      baralpha
##           0.01          0.05         0.05         0.05          0.10
```

We can finally generate the potential outcomes;

```
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
```

Now, I would like to visualize my potential outcomes:

```
plot(y0,y1)
```

You can see on the resulting Figure 1.2 that both potential outcomes are positively correlated. Those with a large potential outcome when untreated (*e.g.* in good health without the treatment) also have a positive health with the treatment. It is also true that individuals with bad health in the absence of the treatment also have bad health with the treatment.

1.1.3   Switching equation

The last building block of the RCM is the switching equation. It links the observed outcome to the potential outcomes through the allocation rule:

$$Y_i = \begin{cases} Y_i^1 & \text{if } D_i = 1 \\ Y_i^0 & \text{if } D_i = 0 \end{cases} \tag{1.4}$$
$$= Y_i^1 D_i + Y_i^0 (1 - D_i)$$

Example 1.3. In order to generate observed outcomes in our numerical example, we simply have to enforce the switching equation:

```
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```
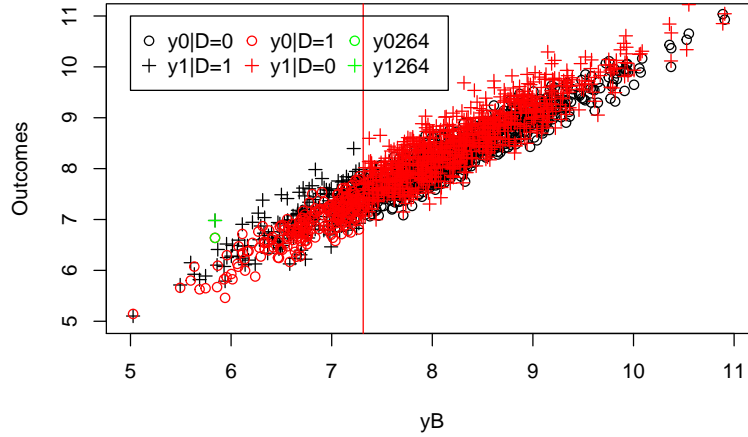
Figure 1.3: Potential outcomes

What the switching equation (1.4) means is that, for each individual i, we get to observe only one of the two potential outcomes. When individual i belongs to the treatment group (*i.e.* $D_i = 1$), we get to observe Y_i^1. When individual i belongs to the control group (*i.e.* $D_i = 0$), we get to observe Y_i^0. Because the same individual cannot be at the same time in both groups, we can NEVER see both potential outcomes for the same individual at the same time.

For each of the individuals, one of the two potential outcomes is unobserved. We say that it is a **counterfactual**. A counterfactual quantity is a quantity that is, according to Hume's definition, contrary to the observed facts. A counterfactual cannot be observed, but it can be conceived by an effort of reason: it is the consequence of what would have happened had some action not been taken.

Remark. One very nice way of visualising the switching equation has been proposed by Jerzy Neyman in a 1923 prescient paper. Neyman proposes to imagine two urns, each one filled with N balls. One urn is the treatment urn and contains balls with the id of the unit and the value of its potential outcome Y_i^1. The other urn is the control urn, and it contains balls with the value of the potential outcome Y_i^0 for each unit i. Following the allocation rule D_i, we decide whether unit i is in the treatment or control group. When unit i is in the treatment group, we take the corresponding ball from the first urn and observe the potential outcome on it. But, at the same time, the urns are connected so that the corresponding ball with the potential outcome of unit i in the control urn disappears as soon as we draw ball i from the treatment urn.

The switching equation works a lot like Schrodinger's cat paradox. Schrodinger's cat is placed in a sealed box and receives a dose of poison when an atom emits a radiation. As long as the box is sealed, there is no way we can know whether the cat is dead or alive. When we open the box, we observe either a dead cat or a living cat, but we cannot observe the cat both alive and dead at the same time. The switching equation is like opening the box, it collapses the observed outcome into one of the two potential ones.

Example 1.4. One way to visualize the inner workings of the switching equation is to plot the potential outcomes along with the criteria driving the allocation rule. In our simple example, it simply amounts to plotting observed (y_i) and potential outcomes (y_i^1 and y_i^0) along y_i^B.

```
plot(yB[Ds==0],y0[Ds==0],pch=1,xlim=c(5,11),ylim=c(5,11),xlab="yB",ylab="Outcomes")
points(yB[Ds==1],y1[Ds==1],pch=3)
points(yB[Ds==0],y1[Ds==0],pch=3,col='red')
points(yB[Ds==1],y0[Ds==1],pch=1,col='red')
test <- 5.8
i.test <- which(abs(yB-test)==min(abs(yB-test)))
points(yB[abs(yB-test)==min(abs(yB-test))],y1[abs(yB-test)==min(abs(yB-test))],col='green',pch=3)
points(yB[abs(yB-test)==min(abs(yB-test))],y0[abs(yB-test)==min(abs(yB-test))],col='green')
abline(v=log(param["barY"]),col="red")
legend(5,11,c('y0|D=0','y1|D=1','y0|D=1','y1|D=0',paste('y0',i.test,sep=''),paste('y1',i.test,sep='')),
```
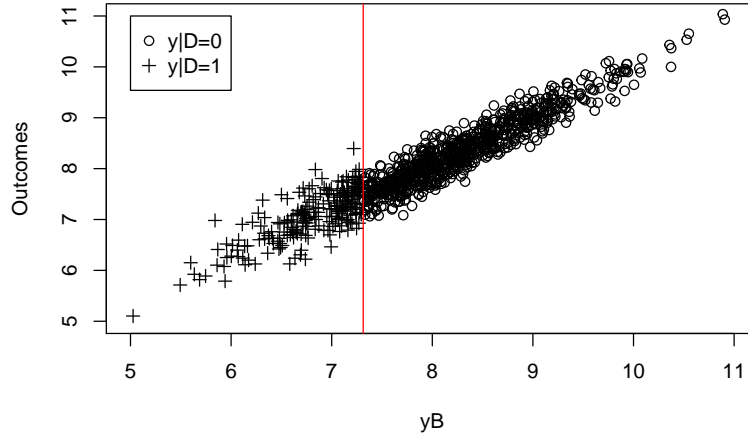
Figure 1.4: Observed outcomes

```r
plot(yB[Ds==0],y0[Ds==0],pch=1,xlim=c(5,11),ylim=c(5,11),xlab="yB",ylab="Outcomes")
points(yB[Ds==1],y1[Ds==1],pch=3)
legend(5,11,c('y|D=0','y|D=1'),pch=c(1,3))
abline(v=log(param["barY"]),col="red")
```

Figure 1.3 plots the observed outcomes y_i along with the unobserved potential outcomes. Figure 1.3 shows that each individual in the sample is endowed with two potential outcomes, represented by a circle and a cross. Figure 1.4 plots the observed outcomes y_i that results from applying the switching equation. Only one of the two potential outcomes is observed (the cross for the treated group and the circle for the untreated group) and the other is not. The observed sample in Figure 1.4 only shows observed outcomes, and is thus silent on the values of the missing potential outcomes.

1.2 Treatment effects

The RCM enables the definition of causal effects at the individual level. In practice though, we generally focus on a summary measure: the effect of the treatment on the treated.

1.2.1 Individual level treatment effects

Potential outcomes enable us to define the central notion of causal inference: the causal effect, also labelled the treatment effect, which is the difference between the two potential outcomes.

Definition 1.2 (Individual level treatment effect)**.** For each unit i, the causal effect of the treatment on outcome Y is: $\Delta_i^Y = Y_i^1 - Y_i^0$.

Example 1.5. The individual level causal effect in log terms is: $\Delta_i^y = \alpha_i = \bar{\alpha} + \theta\mu_i + \eta_i$. The effect is the sum of a part common to all individuals, a part correlated with μ_i: the treatment might have a larger or a smaller effect depending on the unobserved permanent ability or health status of individuals, and a random shock. It is possible to make the effect of the treatment to depend on U_i^B also, but it would complicate the model.

In Figure 1.3, the individual level treatment effects are the differences between each cross and its corresponding circle. For example, for observation 264, the two potential outcomes appear in green in Figure 1.3. The effect of the treatment on unit 264 is equal to:

$$\Delta_{264}^y = y_{264}^1 - y_{264}^0 = 6.98 - 6.64 = 0.34.$$

Since observation 264 belongs to the treatment group, we can only observe the potential outcome in the presence of the treatment, y_{264}^1.

The RCM allows for heterogeneity of treatment effects. The treatment has a large effect on some units and a much smaller effect on other units. We can even have some units that benefit from the treatment and some units that are harmed by the treatment. The individual level effect of the treatment is itself a random variable (and not a fixed parameter). It has a distribution, F_{Δ^Y}.

Heterogeneity of treatment effects seems very natural: the treatment might interact with individuals' different backgrounds. The effect of a drug might depend on the genetic background of an individual. An education program might only work for children that already have sufficient non-cognitive skills, and thus might depend in turn on family background. An environmental regulation or a behavioral intervention might only trigger reactions by already environmentally aware individuals. A CCT might have a larger effect when indiviuals are credit-constrained or face shocks.

Example 1.6. In our numerical example, the distribution of $\Delta_i^y = \alpha_i$ is a normal: $\alpha_i \sim \mathcal{N}(\bar{\alpha} + \theta\bar{\mu}, \theta^2\sigma_\mu^2 + \sigma_\eta^2)$. We would like to visualize treatment effect heterogeneity. For that, we can build a histogram of the individual level causal effect.

On top of the histogram, we can also draw the theoretical distribution of the treatment effect: a normal with mean 0.18 and variance 0.05.

```
hist(alpha,main="",prob=TRUE)
curve(dnorm(x, mean=(param["baralpha"]+param["theta"]*param["barmu"]), sd=sqrt(param["theta"]^2*param["
```
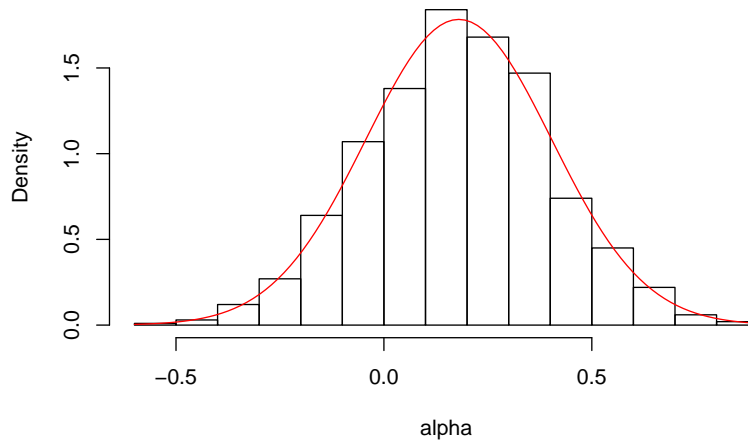

Figure 1.5: Histogram of Δ^y

The first thing that we can see on Figure 1.5 is that the theoretical and the empirical distributions nicely align with each other. We also see that the majority of the observations lies to the right of zero: most people experience a positive effect of the treatment. But there are some individuals that do not benefit from the treatment: the effect of the treatment on them is negative.

1.2.2   Average treatment effect on the treated

We do not generally estimate individual-level treatment effects. We generally look for summary statistics of the effect of the treatment. By far the most widely reported causal parameter is the Treatment on the Treated parameter (TT). It can be defined in the sample at hand or in the population.

Definition 1.3 (Average and expected treatment effects on the treated)**.** The Treatment on the Treated parameters for outcome Y are:

- The average Treatment effect on the Treated in the sample:

$$\Delta_{TT_s}^Y = \frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N (Y_i^1 - Y_i^0) D_i,$$

- The expected Treatment effect on the Treated in the population:

$$\Delta_{TT}^Y = \mathbb{E}[Y_i^1 - Y_i^0 | D_i = 1].$$

The TT parameters measure the average effect of the treatment on those who actually take it, either in the sample at hand or in the popluation. It is generally considered to be the most policy-relevant parameter since it measures the effect of the treatment as it has actually been allocated. For example, the expected causal effect on the overall population is only relevant if policymakers are considering implementing the treatment even on those who have not been selected to receive it. For a drug or an anti-poverty program, it would mean giving the treatment to healthy or rich people, which would make little sense.

TT does not say anything about how the effect of the treatment is distributed in the population or in the sample. TT does not account for the heterogneity of treatment effects. In Lecture 7, we will look at other parameters of interest that look more closely into how the effect of the treatment is distributed.

Example 1.7. The value of TT in our sample is:

$$\Delta_{TT_s}^y = 0.168.$$

Computing the population value of TT is slightly more involved: we have to use the formula for the conditional expectation of a censored bivariate normal random variable:

$$
\begin{aligned}
\Delta_{TT}^y &= \mathbb{E}[\alpha_i | D_i = 1] \\
&= \bar{\alpha} + \theta \mathbb{E}[\mu_i | \mu_i + U_i^B \le \bar{y}] \\
&= \bar{\alpha} + \theta \left(\bar{\mu} - \frac{\sigma_\mu^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}} \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} \right) \\
&= \bar{\alpha} + \theta\bar{\mu} - \theta \left(\frac{\sigma_\mu^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}} \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} \right),
\end{aligned}
$$

where ϕ and Φ are respectively the density and the cumulative distribution functions of the standard normal. The second equality follows from the definition of α_i and D_i and from the fact that η_i is independent from μ_i and U_i^B. The third equality comes from the formula for the expectation of a censored bivariate normal random variable. In order to compute the population value of TT easily for different sets of parameter values, let's write a function in R:

```
delta.y.tt <- function(param){return(param["baralpha"]+param["theta"]*param["barmu"]
                        -param["theta"]*((param["sigma2mu"]*dnorm((log(param["barY"])-param
                            /(sqrt(param["sigma2mu"]+param["sigma2U"])
                                *pnorm((log(param["barY"])-param["barmu"])/(sqr
```

The population value of TT computed using this function is: $\Delta_{TT}^y = 0.172$. We can see that the values of TT in the sample and in the population differ slightly. This is because of sampling noise: the units in the sample are not perfectly representative of the units in the population.

1.3 Fundamental problem of causal inference

At least in this lecture, causal inference is about trying to infer TT, either in the sample or in the population. The FPCI states that it is impossible to directly observe TT because one part of it remains fundamentally unobserved.

Theorem 1.1 (Fundamental problem of causal inference). *It is impossible to observe TT, either in the population or in the sample.*

Proof. The proof of the FPCI is rather straightforward. Let me start with the sample TT:

$$\Delta^Y_{TT_s} = \frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N (Y_i^1 - Y_i^0) D_i$$

$$= \frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N Y_i^1 D_i - \frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N Y_i^0 D_i$$

$$= \frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N Y_i D_i - \frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N Y_i^0 D_i.$$

Since Y_i^0 is unobserved whenever $D_i = 1$, $\frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N Y_i^0 D_i$ is unobserved, and so is $\Delta^Y_{TT_s}$. The same is true for the population TT:

$$\Delta^Y_{TT} = \mathbb{E}[Y_i^1 - Y_i^0 | D_i = 1]$$

$$= \mathbb{E}[Y_i^1 | D_i = 1] - \mathbb{E}[Y_i^0 | D_i = 1]$$

$$= \mathbb{E}[Y_i | D_i = 1] - \mathbb{E}[Y_i^0 | D_i = 1].$$

$\mathbb{E}[Y_i^0 | D_i = 1]$ is unobserved, and so is Δ^Y_{TT}. $\square$

The key insight in order to understand the FPCI is to see that the outcomes of the treated units had they not been treated are unobservable, and so is their average or expectation. We say that they are counterfactual, contrary to what has happened.

Definition 1.4 (Couterfactual). Both $\frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N Y_i^0 D_i$ and $\mathbb{E}[Y_i^0 | D_i = 1]$ are counterfactual quantities that we will never get to observe.

Example 1.8. The average counterfactual outcome of the treated is the mean of the red circles in the y axis on Figure 1.3:

$$\frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N y_i^0 D_i = 6.91.$$

Remember that we can estimate this quantity only because we have generated the data ourselves. In real life, this quantity is hopelessly unobserved.

$\mathbb{E}[y_i^0 | D_i = 1]$ can be computed using the formula for the expectation of a censored normal random variable:

$$\begin{aligned}
\mathbb{E}[y_i^0|D_i = 1] &= \mathbb{E}[\mu_i + \delta + U_i^0|D_i = 1] \\
&= \mathbb{E}[\mu_i + \delta + \rho U_i^B + \epsilon_i|D_i = 1] \\
&= \delta + \mathbb{E}[\mu_i + \rho U_i^B|y_i^B \le \bar{y}] \\
&= \delta + \bar{\mu} - \frac{\sigma_\mu^2 + \rho\sigma_U^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}} \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}.
\end{aligned}$$

We can write a function in R to compute this value:

```r
esp.y0.D1 <- function(param){
  return(param["delta"]+param["barmu"]
        -((param["sigma2mu"]+param["rho"]*param["sigma2U"])
         *dnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"]))))
        /(sqrt(param["sigma2mu"]+param["sigma2U"])*pnorm((log(param["barY"])-param["barmu"])
                                          /(sqrt(param["sigma2mu"]+param["sigma2U"])))))
}
```

The population value of TT computed using this function is: $\mathbb{E}[y_i^0|D_i = 1] = 6.9$.

1.4 Intuitive estimators, confounding factors and selection bias

In this section, we are going to examine the properties of two intuitive comparisons that laypeople, policymakers but also ourselves make in order to estimate causal effects: the with/wihtout comparison (WW) and the before/after comparison (BA). WW compares the average outcomes of the treated individuals with those of the untreated individuals. BA compares the average outcomes of the treated after taking the treatment to their average outcomes before they took the treatment. These comparisons try to proxy for the expected counterfactual outcome in the treated group by using an observed quantity. WW uses the expected outcome of the untreated individuals as a proxy. BA uses the expected outcome of the treated before they take the treatment as a proxy.

Unfortunately, both of these proxies are generally poor and provide biased estimates of TT. The reason that these proxies are poor is that the treatment is not the only factor that differentiates the treated group from the groups used to form the proxy. The intuitive comparisons are biased because factors, other than the treatment, are correlated to its allocation. The factors that bias the intuitive comparisons are generally called confouding factors or confounders.

The treatment effect measures the effect of a ceteris paribus change in treatment status, while the intuitive comparisons capture both the effect of this change and that of other correlated changes that spuriously contaminate the comparison. Intuitive comparisons measure correlations while treatment effects measure causality. The old motto "correlation is not causation" applies vehemently here.
Remark. A funny anecdote about this expression "correlation is not causation". This expression is due to Karl Pearson, the father of modern statistics. He coined the phrase in his famous book "The Grammar of Science." Pearson is famous for inventing the correlation coefficient. He actually thought that correlation was a much superior, much more rigorous term, than causation. In his book, he actually used the sentence to argue in favor of abandoning causation altogether and focusing on the much better-defined and measurable concept of correlation. Interesting turn of events that his sentence is now used to mean that correlation is weaker than causation, totally reverting the original intended meaning.

In this section, we are going to define both comparisons, study their biases and state the conditions under which they identify TT. This will prove to be a very useful introduction to the notion of identification. It is

also very important to be able to understand the sources of bias of comparisons that we use every day and that come very naturally to policy makers and lay people.

Remark. In this section, we state the definitions and formulae in the population. This is for two reasons. First, it is simpler, and lighter in terms of notation. Second, it emphasizes that the problems with intuitive comparisons are independent of sampling noise. Most of the results stated here for the population extend to the sample, replacing the expectation operator by the average operator. I will nevertheless give examples in the sample, since it is so much simpler to compute. I will denote sample equivalents of population estimators with a hat.

1.4.1   With/Without comparison, selection bias and cross-sectional confounders

The with/without comparison (WW) is very intuitive: just compare the outcomes of the treated and untreated individuals in order to estimate the causal effect. This approach is nevertheless generally biased. We call the bias of WW selection bias (SB). Selection bias is due to unobserved confounders that are distributed differently in the treatment and control group and that generate differences in outcomes even in the absence of the treatment. In this section, I define the WW estimator, derives its bias, introduces the confounders and states conditions under which it is unbiased.

1.4.1.1   With/Without comparison

The with/without comparison (WW) is very intuitive: just compare the outcomes of the treated and untreated individuals in order to estimate the causal effect.

Definition 1.5 (With/without comparison). The with/without comparison is the difference between the expected outcomes of the treated and the expected outcomes of the untreated:

$$\Delta_{WW}^{Y} = \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i|D_i = 0].$$

Example 1.9. In the population, WW can be computed using the traditional formula for the expectation of a truncated normal distribution:

$$
\begin{aligned}
\Delta_{WW}^{y} &= \mathbb{E}[y_i|D_i = 1] - \mathbb{E}[y_i|D_i = 0] \\
&= \mathbb{E}[y_i^1|D_i = 1] - \mathbb{E}[y_i^0|D_i = 0] \\
&= \mathbb{E}[\alpha_i|D_i = 1] + \mathbb{E}[\mu_i + \rho U_i^B|\mu_i + U_i^B \leq \bar{y}] - \mathbb{E}[\mu_i + \rho U_i^B|\mu_i + U_i^B > \bar{y}] \\
&= \bar{\alpha} + \theta \left(\bar{\mu} - \frac{\sigma_\mu^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}} \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} \right) - \frac{\sigma_\mu^2 + \rho\sigma_U^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}} \left(\frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} + \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{1 - \Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} \right).
\end{aligned}
$$

In order to compute this parameter, we are going to set up a R function. For reasons that will become clearer later, we will define two separate functions to compute the first and second part of the formula. In the first part, you should have recognised TT, that we have already computed in Lecture 1. We are going to call the second part SB, for reasons that will become explicit in a bit.

```
delta.y.tt <- function(param){
  return(param["baralpha"]+param["theta"]*param["barmu"]-param["theta"]
        *((param["sigma2mu"]*dnorm((log(param["barY"])-param["barmu"])
                              /(sqrt(param["sigma2mu"]+param["sigma2U"]))))
          /(sqrt(param["sigma2mu"]+param["sigma2U"])*pnorm((log(param["barY"])-param["barmu"])
                                /(sqrt(param["sigma2mu"]+param["sigma2U"]))
```

```
}
delta.y.sb <- function(param){
  return(-(param["sigma2mu"]+param["rho"]*param["sigma2U"])/sqrt(param["sigma2mu"]+param["sigma2U"])
         *dnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))
         *(1/pnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))
           +1/(1-pnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"]))))))
}
delta.y.ww <- function(param){
  return(delta.y.tt(param)+delta.y.sb(param))
}
```

As a conclusion of all these derivations, WW in the population is equal to -1.298. Remember that the value of TT in the population is 0.172.

In order to compute the WW estimator in a sample, I'm going to generate a brand new sample and I'm going to choose a seed for the pseudo-random number generator so that we obtain the same result each time we run the code. I use `set.seed(1234)` in the code chunk below.

```
param <- c(8,.5,.28,1500)
names(param) <- c("barmu","sigma2mu","sigma2U","barY")
set.seed(1234)
N <-1000
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
Ds[YB<=param["barY"]] <- 1
l <- length(param)
param <- c(param,0.9,0.01,0.05,0.05,0.05,0.1)
names(param)[(l+1):length(param)] <- c("rho","theta","sigma2epsilon","sigma2eta","delta","baralpha")
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

In this sample, the average outcome of the treated in the presence of the treatment is

$$\frac{1}{\sum_{i=1}^{N} D_i} \sum_{i=1}^{N} D_i y_i = 7.074.$$

It is materialized by a circle on Figure 1.6. The average outcome of the untreated is

$$\frac{1}{\sum_{i=1}^{N}(1-D_i)} \sum_{i=1}^{N}(1-D_i)y_i = 8.383.$$

It is materialized by a plus sign on Figure 1.6.
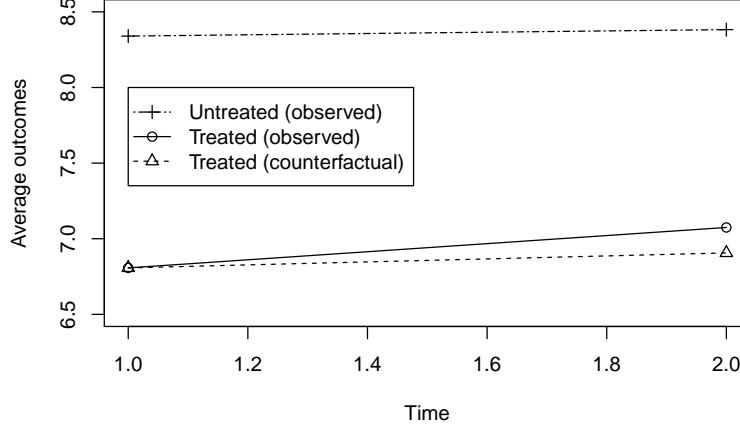
The estimate of the WW comparison in the sample is thus:

Figure 1.6: Evolution of average outcomes in the treated and control group before (Time =1) and after (Time=2) the treatment

$$\Delta\hat{Y}_{WW} = \frac{1}{\sum_{i=1}^{N} D_i} \sum_{i=1}^{N} Y_i D_i - \frac{1}{\sum_{i=1}^{N}(1 - D_i)} \sum_{i=1}^{N} Y_i(1 - D_i).$$

We have $\Delta\hat{y}_{WW}$ = -1.308. Remember that the value of TT in the sample is $\Delta^{y}_{TT_s} = 0.168$.

Overall, WW severely underestimates the effect of the treatment in our example. WW suggests that the treatment has a negative effect on outcomes whereas we know by construction that it has a positive one.

1.4.1.2   Selection bias

When we form the with/without comparison, we do not recover the TT parameter. Instead, we recover TT plus a bias term, called **selection bias**:

$$\Delta^{Y}_{WW} = \Delta^{Y}_{TT} + \Delta^{Y}_{SB}.$$

Definition 1.6 (Selection bias)**.** Selection bias is the difference between the with/without comparison and the treatment on the treated parameter:

$$\Delta^{Y}_{SB} = \Delta^{Y}_{WW} - \Delta^{Y}_{TT}.$$

WW tries to approximate the counterfactual expected outcome in the treated group by using $\mathbb{E}[Y_i^0|D_i = 0]$, the expected outcome in the untreated group . Selection bias appears because this proxy is generally poor. It is very easy to see that selection bias is indeed directly due to this bad proxy problem:

Theorem 1.2 (Selection bias and counterfactual)**.** *Selection bias is the difference between the counterfactual expected potential outcome in the absence of the treatment among the treated and the expected potential outcome in the absence of the treatment among the untreated.*

$$\Delta^{Y}_{SB} = \mathbb{E}[Y_i^0|D_i = 1] - \mathbb{E}[Y_i^0|D_i = 0].$$

Proof.

$$\Delta_{SB}^Y = \Delta_{WW}^Y - \Delta_{TT}^Y$$
$$= \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i|D_i = 0] - \mathbb{E}[Y_i^1 - Y_i^0|D_i = 1]$$
$$= \mathbb{E}[Y_i^0|D_i = 1] - \mathbb{E}[Y_i^0|D_i = 0].$$

The first and second equalities stem only from the definition of both parameters. The third equality stems from using the switching equation: $Y_i = Y_i^1 D_i + Y_i^0(1 - D_i)$, so that $\mathbb{E}[Y_i|D_i = 1] = \mathbb{E}[Y_i^1|D_i = 1]$ and $\mathbb{E}[Y_i|D_i = 0] = \mathbb{E}[Y_i^0|D_i = 0]$. □

Example 1.10. In the population, SB is equal to

$$\Delta_{SB}^y = \Delta_{WW}^y - \Delta_{TT}^y$$
$$= -1.298 - 0.172$$
$$= -1.471$$

We could have computed SB directly using the formula from Theorem 1.2:

$$\Delta_{SB}^y = \mathbb{E}[y_i^0|D_i = 1] - \mathbb{E}[y_i^0|D_i = 0]$$

$$= -\frac{\sigma_\mu^2 + \rho\sigma_U^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}} \left(\frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} + \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{1 - \Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} \right).$$

When using the R function for SB that we have defined earlier, we indeed find: $\Delta_{SB}^y = $ -1.471.

In the sample, $\Delta_{SB}^{\hat{y}} = $ -1.308-0.168 = -1.476. Selection bias emerges because we are using a bad proxy for the counterfactual. The average outcome for the untreated is equal to $\frac{1}{\sum_{i=1}^N (1-D_i)} \sum_{i=1}^N (1 - D_i)y_i = 8.383$ while the counterfactual average outcome for the treated is $\frac{1}{\sum_{i=1}^N D_i} \sum_{i=1}^N D_i y_i^0 = 6.906$. Their difference is as expected equal to SB: $\Delta_{SB}^{\hat{y}} = 6.906 - 8.383 = $ -1.476. The counterfactual average outcome of the treated is much smaller than the average outcome of the untreated. On Figure 1.6, this is materialized by the fact that the plus sign is located much above the triangle.

Remark. The concept of selection bias is related to but different from the concept of sample selection bias. With sample selection bias, we worry that selection into the sample might bias the estimated effect of a treatment on outcomes. With selection bias, we worry that selection into the treatment itself might bias the effect of the treatment on outcomes. Both biases are due to unbserved covariates, but they do not play out in the same way.

For example, estimating the effect of education on women's wages raises both selection bias and sample selection bias issues. Selection bias stems from the fact that more educated women are more likely to be more dynamic and thus to have higher earnings even when less educated. Selection bias would be positive in that case, overestimating the effect of education on earnings.

Sample selection bias stems from the fact that we can only use a sample of working women in order to estimate the effect of education on wages, since we do not observe the wages on non working women. But, selection into the labor force might generate sample selection bias. More educated women participate more in the labor market, while less educated women participate less. As a consequence, less educated women that work are different from the overall sample of less educated women. They might be more dynamic and work-focused. As a consequence, their wages are higher than the average wages of the less educated women.

Comparing the wages of less educated women that work to those of more educated women that work might understate the effect of education on earnings. Sample selection bias would generate a negative bias on the education coefficient.

1.4.1.3   Confounding factors

Confounding factors are the factors that generate differences between treated and untreated individuals even in the absence of the treatment. The confounding factors are thus responsible for selection bias. In general, the mere fact of being selected for receiving the treatment means that you have a host of characteristics that would differentiate you from the unselected individuals, even if you were not to receive the treatment eventually.

For example, if a drug is given to initially sicker individuals, then, we expect that they will be sicker that the untreated in the absence of the treatment. Comparing sick individuals to healthy ones is not a sound way to estimate the effect of a treatment. Obviously, even if our treatment performs well, healthier individuals will be healthier after the treatment has been allocated to the sicker patients. The best we can expect is that the treated patients have recovered, and that their health after the treatment is comparable to that of the untreated patients. In that case, the with/without comparison is going to be null, whereas the true effect of the treatment is positive. Selection bias is negative in that case: in the absence of the treatment, the average health status of the treated individuals would have been smaller than that of the untreated individuals. The confounding factor is the health status of individuals when the decision to allocate the drug has been taken. It is correlated to both the allocation of the treatment (negatively) and to health in the absence of the treatment (positively).

Example 1.11. In our example, μ_i and U_i^B are the confounding factors. Because the treatment is only given to individuals with pre-treament outcomes smaller than a threshold ($y_i^B \leq \bar{y}$), participants tend to have smaller μ_i and U_i^B than non participants, as we can see on Figure 1.7.
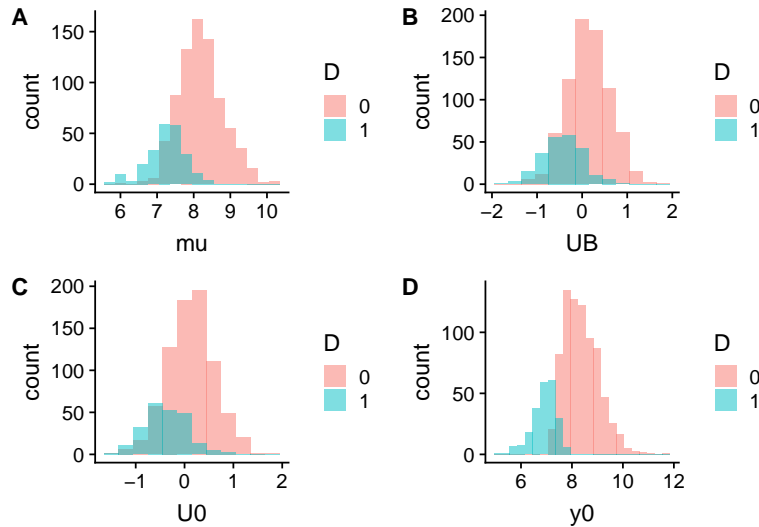
Figure 1.7: Distribution of confounders in the treated and control group

Since confounding factors are persistent, they affect the outcomes of participants and non participants after the treatment date. μ_i persists entirely over time, and U_i^B persists at a rate ρ. As a consequence, even in the absence of the treatment, participants have lower outcomes than non participants, as we can see on Figure 1.7.

We can derive the contributions of both confouding factors to overall SB:

$$\mathbb{E}[Y_i^0|D_i = 1] = \mathbb{E}[\mu_i + \delta + U_i^0|\mu_i + U_i^B \leq \bar{y}]$$

$$= \delta + \mathbb{E}[\mu_i|\mu_i + U_i^B \leq \bar{y}] + \rho\mathbb{E}[U_i^B|\mu_i + U_i^B \leq \bar{y}]$$

$$\Delta_{SB}^y = \mathbb{E}[\mu_i|\mu_i + U_i^B \leq \bar{y}] - \mathbb{E}[\mu_i|\mu_i + U_i^B > \bar{y}]$$

$$+ \rho\left(\mathbb{E}[U_i^B|\mu_i + U_i^B \leq \bar{y}] - \mathbb{E}[U_i^B|\mu_i + U_i^B > \bar{y}]\right)$$

$$= -\frac{\sigma_\mu^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}}\left(\frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} + \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{1 - \Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}\right)$$

$$- \frac{\rho\sigma_U^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}}\left(\frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)} + \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}{1 - \Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma_\mu^2+\sigma_U^2}}\right)}\right)$$

In order to evaluate these quantities, let's build two R functions:

```
delta.y.sb.mu <- function(param){
  return(-(param["sigma2mu"])/sqrt(param["sigma2mu"]+param["sigma2U"])
         *dnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))
         *(1/pnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))
           +1/(1-pnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"]))))))
}
delta.y.sb.U <- function(param){
  return(-(param["rho"]*param["sigma2U"])/sqrt(param["sigma2mu"]+param["sigma2U"])
         *dnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))
         *(1/pnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))
           +1/(1-pnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"]))))))
}
```

The contribution of μ_i to selection bias is -0.978 while that of U_i^0 is of -0.493.

1.4.1.4   When does WW identify TT?

Are there conditions under which WW identify TT? The answer is yes: when there is no selection bias, the proxy used by WW for the counterfactual quantity is actually valid. Formally, WW identifies TT when the following assumption holds:

Definition 1.7 (No selection bias)**.** We assume the following:

$$\mathbb{E}[Y_i^0|D_i = 1] = \mathbb{E}[Y_i^0|D_i = 0].$$

Under Assumption 1.7, the expected counterfactual outcome of the treated is equal to the expected potential outcome of the untreated in the absence of the treatment. This yields to the following result:

Theorem 1.3. *Under Assumption 1.7, WW identifies the TT parameter:*

$$\Delta_{WW}^Y = \Delta_{TT}^Y.$$

Proof.

$$\begin{aligned}
\Delta_{WW}^Y &= \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i|D_i = 0] \\
&= \mathbb{E}[Y_i^1|D_i = 1] - \mathbb{E}[Y_i^0|D_i = 0] \\
&= \mathbb{E}[Y_i^1|D_i = 1] - \mathbb{E}[Y_i^0|D_i = 1] \\
&= \Delta_{TT}^Y,
\end{aligned}$$

where the second equation uses the switching equation and the third uses Assumption 1.7.   □

So, under Assumption 1.7, the WW comparison actually identifies the TT parameter. We say that Assumption 1.7 is an **identification assumption**: it serves to identify the parameter of interest using observed data. The intuition for this result is simply that, under Assumption 1.7, there are no confounding factors and thus no selection bias. under Assumption 1.7, the factors that yield individuals to receive or not the treatment are mean-independent of the potential outcomes in the absence of the treatment. In this case, the expected outcome in the untreated group actually is a perfect proxy for the counterfactual expected outcome of the treated group.

Obviously, Assumption 1.7 is extremely unlikely to hold in real life. For Assumption 1.7 to hold, it has to be that **all** the determinants of D_i are actually unrelated to Y_i^0. One way to enforce Assumption 1.7 is to randomize treatment intake. We will see this in the Lecture on RCTs. It might also be possible that Assumption 1.7 holds in the data in the absence of an RCT. But this is not very likely, and should be checked by every mean possible.

One way to test for the validity of Assumption 1.7 is to compare the values of observed covariates in the treated and untreated group. For Assumption 1.7 to be credible, observed covariates should be distributed in the same way.

Another nice way to test for the validity of Assumption 1.7 with observed data is to implement a **placebo test**. A placebo test looks for an effect where there should be none, if we believe the identification assumptions. For example, under Assumption 1.7 it should be (even though it is not rigorously implied) that outcomes before the treatment are also mean-independent of the treatment allocation. And actually, since a future treatment cannot have an effect today (unless people anticipate the treatment, which we assume away here), the WW comparison before the treatment should be null, therefore giving a zero effect of the placebo treatment "will receive the treatment in the future."

Example 1.12. When the allocation rule defining D_i is the eligibility rule that we have used so far, we have already seen that Assumption 1.7 does not hold and the placebo test should not pass either.

One way of generating Assumption 1.7 from the eligibility rule that we are using is to mute the persistence in outcome dynamics. For example, one could set $\rho = 0$ and $\sigma_\mu^2 = 0$.

```
param <- c(8,0,.28,1500,0,0.01,0.05,0.05,0.05,0.1)
names(param) <- c("barmu","sigma2mu","sigma2U","barY","rho","theta","sigma2epsilon","sigma2eta","delta"
param
```

```
##        barmu      sigma2mu        sigma2U           barY            rho
##         8.00          0.00           0.28        1500.00           0.00
##        theta sigma2epsilon       sigma2eta          delta       baralpha
##         0.01          0.05           0.05           0.05           0.10
```

In that case, outcomes are not persistent and Assumption 1.7 holds:

$$
\begin{aligned}
\mathbb{E}[y_i^0|D_i = 1] &= \mathbb{E}[\mu_i + \delta + U_i^0|y_i^B \leq \bar{y}] \\
&= \mathbb{E}[\bar{\mu} + \delta + \epsilon_i|\bar{\mu} + U_i^B \leq \bar{y}] \\
&= \bar{\mu} + \delta + \mathbb{E}[\epsilon_i|\bar{\mu} + U_i^B \leq \bar{y}] \\
&= \bar{\mu} + \delta + \mathbb{E}[\epsilon_i|\bar{\mu} + U_i^B > \bar{y}] \\
&= \mathbb{E}[\mu_i + \delta + U_i^0|y_i^B > \bar{y}] \\
&= \mathbb{E}[y_i^0|D_i = 0],
\end{aligned}
$$

where the second equality follows from $\sigma_\mu^2 = 0$ and $\rho = 0$ and the fourth from $\epsilon_i \perp\!\!\!\perp U_i^B$. Another direct way to see this is to use the formula for selection bias that we have derived above. It is easy to see that with $\rho = 0$ and $\sigma_\mu^2 = 0$, $\Delta_{SB}^y = 0$. To be sure, we can compute Δ_{SB}^y with the new parameter values: $\Delta_{SB}^y = 0$. As a consequence, $\Delta_{TT}^y = 0.18 = 0.18 = \Delta_{WW}^y$.

Remark. You might have noticed that the value of Δ_{TT}^y is different than before. It is normal, since it depends on the values of parameters, and especially on σ_μ^2 and ρ.

Let's see how these quantities behave in the sample.

```
set.seed(1234)
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
Ds[YB<=param["barY"]] <- 1
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

We can see that $\mathbb{E}[Y_i^0|\hat{D}_i = 1] = 8.038 \approx 8.055 = \mathbb{E}[Y_i^0|\hat{D}_i = 0]$. This means that WW should be close to TT: $\Delta_{TT}^{\hat{y}} = 0.198 \approx 0.182 = \Delta_{WW}^{\hat{y}}$. Note that $\hat{W}W$ in the sample is not exactly, but only approximately, equal to TT in the population and in the sample. This is an instance of the Fundamental Problem of Statistical Inference that we will study in the next chapter.

Under these restrictions, the placebo test would unfortunately conclude against Assumption 1.7 even though it is valid:

$$
\begin{aligned}
\mathbb{E}[y_i^B|D_i = 1] &= \mathbb{E}[\mu_i + U_i^B|y_i^B \leq \bar{y}] \\
&= \mathbb{E}[\bar{\mu} + U_i^B|\bar{\mu} + U_i^B \leq \bar{y}] \\
&= \bar{\mu} + \mathbb{E}[U_i^B|\bar{\mu} + U_i^B \leq \bar{y}] \\
&\neq \bar{\mu} + \mathbb{E}[U_i^B|\bar{\mu} + U_i^B > \bar{y}] \\
&= \mathbb{E}[\mu_i + U_i^0|y_i^B > \bar{y}] \\
&= \mathbb{E}[y_i^B|D_i = 0].
\end{aligned}
$$

In the sample, we indeed have that $\mathbb{E}[Y_i^B|\hat{D}_i = 1] = 7.004 \neq 8.072 = \mathbb{E}[Y_i^B|\hat{D}_i = 0]$. The reason for the failure of the placebo test to conclude that Ww is actually correct is that the U_i^B shock enters both into the selection equation and the outcome equation for y_i^B, generating a wage at period B between the outcomes of the treated and of the untreated. Since it is not persistent, this wedge does not generate selection bias. This wedge would not be detected if we could perform it further back in time, before the selection period.

Another way to make Assumption 1.7 work is to generate a new allocation rule where all the determinants of treatment intake are indeed orthogonal to potential outcomes and to outcomes before the treatment. Let's assume for example that $D_i = \mathbb{1}[V_i \leq \bar{y}]$, with $V_i \sim \mathcal{N}(\bar{\mu}, \sigma_\mu^2 + \sigma_U^2)$ and $V_i \perp\!\!\!\perp (Y_i^0, Y_i^1, Y_i^B, \mu_i, \eta_i)$. In that case, Assumption 1.7 holds and the placebo test does work. Indeed, we have:

$$
\begin{aligned}
\Delta_{TT}^y &= \mathbb{E}[Y_i^1 - Y_i^0|D_i = 1] \\
&= \mathbb{E}[\alpha_i|D_i = 1] \\
&= \mathbb{E}[\bar{\alpha} + \theta\mu_i + \eta_i|V_i \leq \bar{y}] \\
&= \bar{\alpha} + \theta\bar{\mu} \\
&= \Delta_{ATE}^y \\
\Delta_{WW}^y &= \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i|D_i = 0] \\
&= \mathbb{E}[Y_i^1|D_i = 1] - \mathbb{E}[Y_i^0|D_i = 0] \\
&= \mathbb{E}[Y_i^1|V_i \leq \bar{y}] - \mathbb{E}[Y_i^0|V_i > \bar{y}] \\
&= \mathbb{E}[Y_i^1] - \mathbb{E}[Y_i^0] \\
&= \Delta_{ATE}^y
\end{aligned}
$$

ATE is the Average Treatment Effect in the population. It is the expected effect of the treatment on all the members of the population, not only on the treated. When the treatment is randomly allocated, both TT and ATE are equal, since the treated are a random subset of the overall population. I prefer to use ATE for my definition of the R function in order not to erase the definition of the TT function:

```
delta.y.ate <- function(param){
  return(param["baralpha"]+param["theta"]*param["barmu"])
}
```

In the population, WW identifies TT: $\Delta_{TT}^y = 0.18 = \Delta_{WW}^y$. Let's see how these quantities behave in the sample:

```
set.seed(1234)
N <-1000
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
Ds[V<=log(param["barY"])] <- 1
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
```

```
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

In the sample, the counterfactual is well approximated by the outcomes of the untreated: $\mathbb{E}[Y_i^0|\hat{D}_i = 1] = 8.085 \approx 8.054 = \mathbb{E}[Y_i^0|\hat{D}_i = 0]$. As a consequence, WW should be close to TT: $\Delta_{TT}^{\hat{y}} = 0.168 \approx 0.199 = \Delta_{WW}^{\hat{y}}$. The placebo test is also valid in that case: $\mathbb{E}[Y_i^B|\hat{D}_i = 1] = 7.95 \approx 7.99 = \mathbb{E}[Y_i^B|\hat{D}_i = 0]$.

1.4.2 The before/after comparison, temporal confounders and time trend bias

The before/after comparison (BA) is also very intuitive: it consists in looking at how the outcomes of the treated have changed over time and to attribute this change to the effect of the treatment. The problem is that other changes might have affected outcomes in the absence of the treatment, thereby biasing BA. The bias of BA is called time-trend bias. It is due to confounders that affect the outcomes of the treated over time. This section defines the BA estimator, derives its bias, describes the role of the confounders and states conditions under which BA identifies TT.

Example 1.13. Before computing any estimates, we need to reset all our parameter values and generated sample it their usual values:

```
param <- c(8,.5,.28,1500)
names(param) <- c("barmu","sigma2mu","sigma2U","barY")
set.seed(1234)
N <-1000
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
Ds[YB<=param["barY"]] <- 1
l <- length(param)
param <- c(param,0.9,0.01,0.05,0.05,0.05,0.1)
names(param)[(l+1):length(param)] <- c("rho","theta","sigma2epsilon","sigma2eta","delta","baralpha")
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

1.4.2.1 The before/after comparison

The before/after estimator (BA) compares the outcomes of the treated after taking the treatment to the outcomes of the treated before taking the treatment. It is also sometimes called a "pre-post comparison."

Definition 1.8 (Before/after comparison). The before/after comparison is the difference between the expected outcomes in the treated group after the treatment and the expected outcomes in the same group before the treatment:

$$\Delta_{BA}^Y = \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i^B|D_i = 1].$$

Example 1.14. In the population, the BA estimator has the following shape:

$$
\begin{aligned}
\Delta^y_{BA} &= \mathbb{E}[y_i|D_i = 1] - \mathbb{E}[y_i^B|D_i = 1] \\
&= \mathbb{E}[y_i^1 - y_i^B|D_i = 1] \\
&= \mathbb{E}[\alpha_i|D_i = 1] + \delta + (\rho - 1)\mathbb{E}[U_i^B|\mu_i + U_i^B \le \bar{y}] \\
&= \Delta^y_{TT} + \delta + (1 - \rho)\left(\frac{\sigma^2_U}{\sqrt{\sigma^2_\mu + \sigma^2_U}} \frac{\phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma^2_\mu+\sigma^2_U}}\right)}{\Phi\left(\frac{\bar{y}-\bar{\mu}}{\sqrt{\sigma^2_\mu+\sigma^2_U}}\right)} \right).
\end{aligned}
$$

In order to compute BA in the population, we can again use a R function, combining the value of TT and that of the second part of the formula, that we are going to denote TB for reasons that are going to become clear in a bit.

```
delta.y.tb <- function(param){
  return(param["delta"]
         +(1-param["rho"])*((param["sigma2U"])/sqrt(param["sigma2mu"]+param["sigma2U"]))
         *dnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))
         /pnorm((log(param["barY"])-param["barmu"])/(sqrt(param["sigma2mu"]+param["sigma2U"])))))
}
delta.y.ba <- function(param){
  return(delta.y.tt(param)+ delta.y.tb(param))
}
```

The value of BA in the population is thus $\Delta^y_{BA} = 0.265$. Remember that the true value of TT in the population is 0.172. In the sample, the value of BA is $\Delta^{\hat{y}}_{BA} = 0.267$. Remember that the value of TT in the sample is $\Delta^y_{TT_s} = 0.168$.

1.4.2.2   Time trend bias

When we form the before/after comparison, we do not recover the TT parameter. Instead, we recover TT plus a bias term, called **time trend bias**:

$$
\Delta^Y_{BA} = \Delta^Y_{TT} + \Delta^Y_{TB}.
$$

Definition 1.9 (Time trend bias)**.** Time trend bias is the difference between the before/after comparison and the treatment on the treated parameter:

$$
\Delta^Y_{TB} = \Delta^Y_{BA} - \Delta^Y_{TT}.
$$

BA uses the expected outcome in the treated group before the treatment as a proxy for the expected counterfactual outcome in the absence of the treatment in the same group. TB is due to the fact that BA uses an imperfect proxy for the counterfactual expected outcome of the treated:

Theorem 1.4. *Time trend bias is the difference between the counterfactual expected potential outcome in the absence of the treatment among the treated and the expected outcome before the treatment in the same group.*

$$
\Delta^Y_{TB} = \mathbb{E}[Y_i^0|D_i = 1] - \mathbb{E}[Y_i^B|D_i = 1].
$$

Proof.

$$\Delta_{TB}^Y = \Delta_{BA}^Y - \Delta_{TT}^Y$$
$$= \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i^B|D_i = 1] - \mathbb{E}[Y_i^1 - Y_i^0|D_i = 1]$$
$$= \mathbb{E}[Y_i^0|D_i = 1] - \mathbb{E}[Y_i^B|D_i = 1].$$

The first and second equalities stem from the definition of both parameters. The third equality stems from using the switching equation: $Y_i = Y_i^1 D_i + Y_i^0(1 - D_i)$, so that $\mathbb{E}[Y_i|D_i = 1] = \mathbb{E}[Y_i^1|D_i = 1]$. $\square$

Example 1.15. In the population, TB is equal to

$\Delta_{TB}^y = \Delta_{BA}^y - \Delta_{TT}^y = 0.265 - 0.172 = 0.093$. We could have computed this result using Theorem1.4:

$$\Delta_{TB}^y = \mathbb{E}[y_i^0|D_i = 1] - \mathbb{E}[y_i^B|D_i = 1]$$

$$= \delta + (1 - \rho) \left(\frac{\sigma_U^2}{\sqrt{\sigma_\mu^2 + \sigma_U^2}} \frac{\phi\left(\frac{\bar{y} - \bar{\mu}}{\sqrt{\sigma_\mu^2 + \sigma_U^2}}\right)}{\Phi\left(\frac{\bar{y} - \bar{\mu}}{\sqrt{\sigma_\mu^2 + \sigma_U^2}}\right)} \right).$$

Using the R function that we have defined previously, this approach gives $\Delta_{TB}^y = 0.093$.

In the sample $\Delta_{BA}^{\hat{y}} = 0.267$ while $\Delta_{TT}^{\hat{y}} = 0.168$, so that $\Delta_{TB}^{\hat{y}} = 0.099$. Time trend bias emerges because we are using a bad proxy for the counterfactual average outcomes of the treated. The average outcome of the treated before the treatment takes place is $\mathbb{E}[y_i^B|\hat{D}_i = 1] = 6.807$ while the true counterfactual average outcome for the treated after the treatment is $\mathbb{E}[y_i^0|\hat{D}_i = 1] = 6.906$. Outcomes would have increased in the treatment group even in the absence of the treatment. As a consequence, the BA comparison overestimates the true effect of the treatment. TB estimated using Theorem 1.4 is equal to: $\Delta_{TB}^{\hat{y}} = 6.906 - 6.807 = 0.099$. This can be seen on Figure 1.6: the triangle in period 2 is higher than in period 1.

1.4.2.3 Temporal confounders

Temporal confounders make the outcomes of the treated change at the same time as the treatment status changes, thereby confounding the effect of the treatment. Temporal confounders are responsible for time trend bias.

Over time, there are other things that change than the treatment status. For example, maybe sick individuals naturally recover, and thus their counterfactual health status is better than ther health status before taking the treatment. As a results, BA might overstimate the effect of the treatment. It might also be that the overall level of health in the country has increased, because of increasing GDP, for example.

Example 1.16. In our example, δ and U_i^B are the confounders. δ captures the overall changes in outcomes over time (business cycle, general improvement of health status). U_i^B captures the fact that transitorily sicker individuals tend at the same time to receive the treatment and also to recover naturally. The BA comparison incorrectly attributes both of these changes to the effect of the treatment. We can compute the relative contributions of both sources to the overall time-trend bias in the population.

δ contributes for 0.05 while U_i^B contributes for 0.043.

1.4.2.4 When does BA identify TT?

Are there conditions under which BA actually identifies TT? The answer is yes, when there are no temporal confounders. When that is the case, the variation of outcomes over time is only due to the treatment and it identifies the treatment effect.

More formally, we make the following assumption:

Definition 1.10 (No time trend bias)**.** We assume the following:

$$\mathbb{E}[Y_i^0|D_i = 1] = \mathbb{E}[Y_i^B|D_i = 1].$$

Under Assumption 1.10, the expected counterfactual outcome of the treated is equal to the expected potential outcome of the untreated in the absence of the treatment. This yields to the following result:

Theorem 1.5. *Under Assumption 1.10, BA identifies the TT parameter:*

$$\Delta_{BA}^Y = \Delta_{TT}^Y.$$

Proof.

$$\begin{aligned}
\Delta_{BA}^Y &= \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i^B|D_i = 1]\\
&= \mathbb{E}[Y_i^1|D_i = 1] - \mathbb{E}[Y_i^B|D_i = 1]\\
&= \mathbb{E}[Y_i^1|D_i = 1] - \mathbb{E}[Y_i^0|D_i = 1]\\
&= \Delta_{TT}^Y,
\end{aligned}$$

where the second equation uses the switching equation and the third uses Assumption 1.10.   □

Under Assumption 1.10 the outcomes of the treated before the treatment takes place are a good proxy for the counterfactual. As a consequence, BA identifies TT. Under Assumption 1.10is very unlikely to hold in real life. Indeed, it requires tha nothing happens to the outcomes of the treated in the absence of the treatment. Assumption 1.10 rules out economy-wide shocks but also mean-reversion, such as when sick people naturally recover from an illness.

A good way to test for the validity of Assumption 1.10 is to perform a placebo test. Two of these tests are possible. One placebo test would be to apply the BA estimator between two pre-treatment periods where nothing should happen since the treatment status does not vary and, by assumption, nothing else should vary. A second placebo test would be to apply the BA estimator to a group that does not receive the treatment. The untreated group is a perfectly suitable candidate for that. Assumption 1.10 does not imply that there should be no change in the untreated outcomes, but detecting such a change would cast a serious doubt on the validity of Assumption 1.10.

Example 1.17. One way to generate a population in which Assumption 1.10 holds is to shut down the two sources of confounders in our original model by setting both $\delta = 0$ and $\rho = 1$.

```r
param <- c(8,0.5,.28,1500,1,0.01,0.05,0.05,0,0.1)
names(param) <- c("barmu","sigma2mu","sigma2U","barY","rho","theta","sigma2epsilon","sigma2eta","delta"
param
```

```
##         barmu      sigma2mu       sigma2U          barY           rho
##          8.00          0.50          0.28       1500.00          1.00
##         theta sigma2epsilon      sigma2eta         delta      baralpha
##          0.01          0.05          0.05          0.00          0.10
```

In that case, according to the formula we have derived for TB, we have: $\Delta_{TB}^y = 0$. Let's see how these quantities behave in the sample:

```r
set.seed(1234)
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
```

```
YB <- exp(yB)
Ds <- rep(0,N)
Ds[YB<=param["barY"]] <- 1
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu + U0 + param["delta"]
alpha <- param["baralpha"]+ param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

In the sample, the value of BA is $\Delta^{\hat{y}}_{BA} = 0.173$ while the value of TT in the sample is $\Delta^y_{TT_s} = 0.168$. We cannot perform a placebo test using two periods of pre-treatment outcomes for the treated since we have generated only one period of pre-treatment outcome. We will be able to perform this test later in the DID lecture. We can perfom the placebo test that applies the BA estimator to the untreated. The value of BA for the untreated is $\Delta^y_{\widehat{BA|D=0}} = 0.007$, which is reasonably close to zero.

Chapter 2

The Fundamental Problem of Statistical Inference

The Fundamental Problem of Statistical Inference (FPSI) states that, even if we have an estimator E that identifies TT in the population, we cannot observe E because we only have access to a finite sample of the population. The only thing that we can form from the sample is a sample equivalent $\hat{E}$ to the population quantity E, and $\hat{E} \neq E$. For example, the sample analog to WW is the difference in means between treated and untreated units $\hat{W}W$. As we saw in the last lecture, $\hat{W}W$ is never exactly equal to WW.

Why is $\hat{E} \neq E$? Because a finite sample is never perfectly representative of the population. In a sample, even in a random sample, the distribution of the observed and unobserved covariates deviates from the true population one. As a consequence, the sample value of the estimator is never precisely equal to the population value, but fluctuates around it with sampling noise. The main problem with the FPSI is that if we find an effect of our treatment, be it small or large, we cannot know whether we should attribute it to the treatment or to the bad or good luck of sampling noise.

What can we do to deal with the FPSI? I am going to argue that there are mainly two things that we might want to do: estimating the extent of sampling noise and decreasing sampling noise.

Estimating sampling noise means measuring how much variability there is in our estimate $\hat{E}$ due to the sampling procedure. This is very useful because it enables us to form a confidence interval that gauges how far from $\hat{E}$ the true value E might be. It is a measure of the precision of our estimation and of the extent to which sampling noise might drive our results. Estimating sampling noise is very hard because we have only access to one sample and we would like to know the behavior of our estimator over repeated samples. We are going to learn four ways to estimate the extent of sampling noise using data from one sample.

Because sampling noise is such a nuisance and makes our estimates imprecise, we would like to be able to make it as small as possible. We are going to study three ways of decreasing sampling noise, two that take place before collecting the data (increasing sample size, stratifying) and one that takes place after (conditioning).

Maybe you are surprised not to find statistical significance tests as an important answer to the FPSI. I argue in this lecture that statistical tests are misleading tools that make us overestimate the confidence in our results and underestimate the scope of sampling noise. Statistical tests are not meant to be used for scientific research, but were originally designed to make decisions in industrial settings where the concept of successive sampling made actual sense. Statistical tests also generate collective behaviors such as publication bias and specification search that undermine the very foundations of science. A general movement in the social sciences, but also in physics, is starting to ban the reporting of p-values.

2.1   What is sampling noise? Definition and illustration

In this section, I am going to define sampling noise and illustrate it with a numerical example. In Section 2.1.1, I define sampling noise. In section 2.1.2, I illustrate how sampling noise varies when one is interested in the population treatment effect. In section 2.1.3, I illustrate how sampling noise varies when one is interesetd in the sample treatment effect. Finally, in section 2.1.4, I show how confidence intervals can be built from an estimate of sampling noise.

2.1.1   Sampling noise, a definition

Sampling noise measures how much sampling variability moves the sample estimator $\hat{E}$ around. One way to define it more rigorously is to make it equal to the width of a confidence interval:

Definition 2.1 (Sampling noise)**.** Sampling noise is the width of the symmetric interval around TT within which $\delta * 100\%$ of the sample estimators fall, where δ is the confidence level. As a consequence, sampling noise is equal to 2ϵ where ϵ is such that:

$$\Pr(|\hat{E} - TT| \le \epsilon) = \delta.$$

This definition tries to capture the properties of the distribution of $\hat{E}$ using only one number. As every simplification, it leaves room for dissatisfaction, exactly as a 2D map is a convenient albeit arbitrary betrayal of a 3D phenomenon. For example, there is nothing sacred about the symmetry of the interval. It is just extremely convenient. One might prefer an interval that is symmetric in tail probabilities instead. Feel free to explore with different concepts if you like.

A related concept to that of sampling noise is that of precision: the smaller the sampling noise, the higher the precision. Precision can be defined for example as the inverse of sampling noise $\frac{1}{2\epsilon}$.

Finally, a very useful concept is that of signal to noise ratio. It is not used in economics, but physicists use this concept all the time. The signal to noise ratio measures the treatment effect in multiple of the sampling noise. If they are of the same order of magnitude, we have a lot of noise and little confidence in our estimates. If the signal is much larger than the noise, we tend to have a lot of confidence in our parameter estimates. The signal to noise ratio can be computed as follows: $\frac{E}{2\epsilon}$ or $\frac{\hat{E}}{2\epsilon}$.

Remark. A very striking result is that the signal to noise ratio of a result that is marginally significant at the 5% level is very small, around one half, meaning that the noise is generally double the signal in these results. We will derive this result after studying how to estimate sampling noise with real data.

There are two distinct ways of understanding sampling noise, depending on whether we are after the population treatment effect (Δ_{TT}^Y) or the sample treatment effect ($\Delta_{TT_s}^Y$). Sampling noise for the population treatment effect stems from the fact that the sample is not perfectly representative of the population. The sample differs from the population and thus the sample estimates differs from the population estimate. Sampling noise for the sample parameter stems from the fact that the control group is not a perfect embodiment of the counterfactual. Discrepancies between treated and control samples are going to generate differences between the WW estimate and the TT effect in the sample.

2.1.2   Sampling noise for the population treatment effect

Sampling noise for the population treatment effect stems from the fact that the sample is not perfectly representative of the population.

Example 2.1. In order to assess the scope of sampling noise for our population treatment effect estimate, let's first draw a sample. In order to be able to do that, I first have to define the parameter values:

```r
param <- c(8,.5,.28,1500,0.9,0.01,0.05,0.05,0.05,0.1)
names(param) <- c("barmu","sigma2mu","sigma2U","barY","rho","theta","sigma2epsilon","sigma2eta","delta"
param
```

```
##       barmu     sigma2mu      sigma2U         barY          rho
##        8.00         0.50         0.28      1500.00         0.90
##       theta sigma2epsilon     sigma2eta        delta     baralpha
##        0.01         0.05         0.05         0.05         0.10
```

```r
set.seed(1234)
N <-1000
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
Ds[V<=log(param["barY"])] <- 1
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)

delta.y.ate <- function(param){
  return(param["baralpha"]+param["theta"]*param["barmu"])
}
```

In this sample, the WW estimator yields an estimate of $\Delta^{\hat{y}}_{WW} = 0.133$. Despite random assignment, we have $\Delta^{\hat{y}}_{WW} \neq \Delta^{y}_{TT} = 0.18$, an instance of the FPSI.

In order to see how sampling noise varies, let's draw another sample. In order to do so, I am going to choose a different seed to initialize the pseudo-random number generator in R.

```r
set.seed(12345)
N <-1000
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
Ds[V<=log(param["barY"])] <- 1
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
```

```
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

In this sample, the WW estimator yields an estimate of $\Delta^{\hat{y}}_{WW} = 0.179$. Again, despite random assignment, we have $\Delta^{\hat{y}}_{WW} \neq \Delta^{y}_{TT} = 0.18$, an instance of the FPSI. Furthermore, the estimate of the population treatment effect in this sample differs from the previous one, a consequence of sampling noise.

Let's now visualize the extent of sampling noise by repeating the procedure multiple times with various sample sizes. This is called Monte Carlo replications: in each replication, I choose a sample size, draw one sample from the population and compute the $\hat{W}W$ estimator. At each replication, the sample I'm using is different, reflecting the actual sampling process and enabling me to gauge the extent of sampling noise. In order to focus on sampling noise alone, I am running the replications in the model in which selection into the treatment is independent on potential outcomes, so that $WW = TT$ in the population. In order to speed up the process, I am using parallelized computing: I send each sample to a different core in my computer so that several samples can be run at the same time. You might want to adapt the program below to the number of cores you actually have using the `ncpus` variable in the beginning of the `.Rmd` file that generates this page.. In order to parallelize computations, I use the Snowfall package in R, that gives very simple and intuitive parallelization commands. In order to save time when generating the graph, I use the wonderful "cache" option of knitr: it stores the estimates from the code chunk and will not rerun it as long as the code inside the chunk has not been altered nor the code of the chunks that it depends on (parameter values, for example).

```
monte.carlo.ww <- function(s,N,param){
  set.seed(s)
  mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
  UB <- rnorm(N,0,sqrt(param["sigma2U"]))
  yB <- mu + UB
  YB <- exp(yB)
  Ds <- rep(0,N)
  V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
  Ds[V<=log(param["barY"])] <- 1
  epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
  eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
  U0 <- param["rho"]*UB + epsilon
  y0 <- mu +  U0 + param["delta"]
  alpha <- param["baralpha"]+  param["theta"]*mu + eta
  y1 <- y0+alpha
  Y0 <- exp(y0)
  Y1 <- exp(y1)
  y <- y1*Ds+y0*(1-Ds)
  Y <- Y1*Ds+Y0*(1-Ds)
  return(c((1/sum(Ds))*sum(y*Ds)-(1/sum(1-Ds))*sum(y*(1-Ds)),var(y[Ds==1]),var(y[Ds==0]),mean(Ds)))
}

simuls.ww.N <- function(N,Nsim,param){
  simuls.ww <- as.data.frame(matrix(unlist(lapply(1:Nsim,monte.carlo.ww,N=N,param=param)),nrow=Nsim,nco
  colnames(simuls.ww) <- c('WW','V1','V0','p')
  return(simuls.ww)
}

sf.simuls.ww.N <- function(N,Nsim,param){
  sfInit(parallel=TRUE,cpus=ncpus)
  sim <- as.data.frame(matrix(unlist(sfLapply(1:Nsim,monte.carlo.ww,N=N,param=param)),nrow=Nsim,ncol=4,
```

```
  sfStop()
  colnames(sim) <- c('WW','V1','V0','p')
  return(sim)
}

simuls.ww <- lapply(N.sample,sf.simuls.ww.N,Nsim=Nsim,param=param)

par(mfrow=c(2,2))
for (i in 1:4){
  hist(simuls.ww[[i]][,'WW'],main=paste('N=',as.character(N.sample[i])),xlab=expression(hat(Delta^yWW)))
  abline(v=delta.y.ate(param),col="red")
}
```


Figure 2.1: Distribution of the *WW* estimator over replications of samples of different sizes

Figure 2.1 is essential to understanding statistical inference and the properties of our estimators. We can see on Figure 2.1 that the estimates indeed move around at each sample replication. We can also see that the estimates seem to be concentrated around the truth. We also see that the estimates are more and more concentrated around the truth as sample size grows larger and larger.

How big is sampling noise in all of these examples? We can compute it by using the replications as approximations to the true distribution of the estimator after an infinite number of samples has been drawn. Let's first choose a confidence level and then compute the empirical equivalent to the formula in Definition 2.1.

```
delta<- 0.99
delta.2 <- 0.95
samp.noise <- function(estim,delta){
  return(2*quantile(abs(delta.y.ate(param)-estim),prob=delta))
}
samp.noise.ww <- sapply(lapply(simuls.ww,`[`,,1),samp.noise,delta=delta)
names(samp.noise.ww) <- N.sample
samp.noise.ww
```

```
##        100       1000      10000      1e+05
## 1.09916429 0.39083801 0.11582492 0.03527744
```

Let's also compute precision and the signal to noise ratio and put all of these results together in a nice table.

Table 2.1: Sampling noise of $\hat{WW}$ for the population treatment effect with $\delta = 0.99$ for various sample sizes

	Sampling noise	Precision	Signal to noise ratio
100	1.10	0.91	0.16
1000	0.39	2.56	0.46
10000	0.12	8.63	1.55
1e+05	0.04	28.35	5.10

```r
precision <- function(estim,delta){
  return(1/samp.noise(estim,delta))
}
signal.to.noise <- function(estim,delta,param){
  return(delta.y.ate(param)/samp.noise(estim,delta))
}
precision.ww <- sapply(lapply(simuls.ww,`[`,,1),precision,delta=delta)
names(precision.ww) <- N.sample
signal.to.noise.ww <- sapply(lapply(simuls.ww,`[`,,1),signal.to.noise,delta=delta,param=param)
names(signal.to.noise.ww) <- N.sample
table.noise <- cbind(samp.noise.ww,precision.ww,signal.to.noise.ww)
colnames(table.noise) <- c('Sampling noise', 'Precision', 'Signal to noise ratio')
knitr::kable(table.noise,caption=paste('Sampling noise of $\\hat{WW}$ for the population treatment effe
```

Finally, a nice way to summarize the extent of sampling noise is to graph how sampling noise varies around the true treatment effect, as shown on Figure 2.2.

```r
colnames(table.noise) <- c('sampling.noise', 'precision', 'signal.to.noise')
table.noise <- as.data.frame(table.noise)
table.noise$N <- as.numeric(rownames(table.noise))
table.noise$TT <- rep(delta.y.ate(param),nrow(table.noise))
ggplot(table.noise, aes(x=as.factor(N), y=TT)) +
  geom_bar(position=position_dodge(), stat="identity", colour='black') +
  geom_errorbar(aes(ymin=TT-sampling.noise/2, ymax=TT+sampling.noise/2), width=.2,position=position_dod
  xlab("Sample Size")+
  theme_bw()
```

With $N = 100$, we can definitely see on Figure 2.2 that sampling noise is ridiculously large, especially compared with the treatment effect that we are trying to estimate. The signal to noise ratio is 0.16, which means that sampling noise is an order of magnitude bigger than the signal we are trying to extract. As a consequence, in 22.2% of our samples, we are going to estimate a negative effect of the treatment. There is also a 20.4% chance that we end up estimating an effect that is double the true effect. So how much can we trust our estimate from one sample to be close to the true effect of the treatment when $N = 100$? Not much.

With $N = 1000$, sampling noise is still large: the signal to noise ratio is 0.46, which means that sampling noise is double the signal we are trying to extract. As a consequence, the chance that we end up with a negative treatment effect has decreased to 0.9% and that we end up with an effect double the true one is 1%. But still, the chances that we end up with an effect that is smaller than three quarters of the true effect is 25.6% and the chances that we end up with an estimator that is 25% bigger than the true effect is 26.2%. These are nontrivial differences: compare a program that increases earnings by 13.5% to one that increases them by 18% and another by 22.5%. They would have completely different cost/benefit ratios. But we at least trust our estimator to give us a correct idea of the sign of the treatment effect and a vague and imprecise idea of its magnitude.

With $N = 10^{4}$, sampling noise is smaller than the signal, which is encouraging. The signal to noise ratio is 1.55. In only 1% of the samples does the estimated effect of the treatment become smaller than 0.125

Figure 2.2: Sampling noise of $\hat{WW}$ (99% confidence) around TT for various sample sizes

or bigger than 0.247. We start gaining a lot of confidence in the relative magnitude of the effect, even if sampling noise is still responsible for economically significant variation.

With $N = 10^{5}$, sampling noise has become trivial. The signal to noise ratio is 5.1, which means that the signal is now 5 times bigger than the sampling noise. In only 1% of the samples does the estimated effect of the treatment become smaller than 0.163 or bigger than 0.198. Sampling noise is not any more responsible for economically meaningful variation.

2.1.3 Sampling noise for the sample treatment effect

Sampling noise for the sample parameter stems from the fact that the treated and control groups are not perfectly identical. The distribution of observed and unobserved covariates is actually different, because of sampling variation. This makes the actual comparison of means in the sample a noisy estimate of the true comparison that we would obtain by comparing the potential outcomes of the treated directly.

In order to understand this issue well and to be able to illustrate it correctly, I am going to focus on the average treatment effect in the whole sample, not on the treated: $\Delta^Y_{ATE_s} = \frac{1}{N}\sum_{i=1}^{N}(Y_i^1 - Y_i^0)$. This enables me to define a sample parameter that is independent of the allocation of D_i. This is without important consequences since these two parameters are equal in the population when there is no selection bias, as we are assuming since the beginning of this lecture. Furthermore, if we view the treatment allocation generating no selection bias as a true random assignment in a Randomized Controlled Trial (RCT), then it is still possible to use this approach to estimate TT if we view the population over which we randomise as the population selected for receiving the treatment, as we will see in the lecture on RCTs.

Example 2.2. In order to assess the scope of sampling noise for our sample treatment effect estimate, we first have to draw a sample:

```
set.seed(1234)
N <-1000
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
Ds[V<=log(param["barY"])] <- 1
```

```
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

In this sample, the treatment effect parameter is $\Delta^y_{ATE_s} = 0.171$. The WW estimator yields an estimate of $\Delta^{\hat{y}}_{WW} = 0.133$. Despite random assignment, we have $\Delta^y_{ATE_s} \neq \Delta^{\hat{y}}_{WW}$, an instance of the FPSI.

In order to see how sampling noise varies, let's draw a new treatment allocation, while retaining the same sample and the same potential outcomes.

```
set.seed(12345)
N <-1000
Ds <- rep(0,N)
V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
Ds[V<=log(param["barY"])] <- 1
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

In this sample, the treatment effect parameter is still $\Delta^y_{ATE_s} = 0.171$. The WW estimator yields now an estimate of $\Delta^{\hat{y}}_{WW} = 0.051$. The WW estimate is different from our previous estimate because the treatment was allocated to a different random subset of people.

Why is this second estimate so imprecise? It might because it estimates one of the two components of the average treatment effect badly, or both. The true average potential outcome with the treatment is, in this sample, $\frac{1}{N}\sum_{i=1}^{N} y_i^1 = 8.207$ while the WW estimate of this quantity is $\frac{1}{\sum_{i=1}^{N} D_i}\sum_{i=1}^{N} D_i y_i = 8.113$. The true average potential outcome without the treatment is, in this sample, $\frac{1}{N}\sum_{i=1}^{N} y_i^0 = 8.036$ while the WW estimate of this quantity is $\frac{1}{\sum_{i=1}^{N}(1-D_i)}\sum_{i=1}^{N}(1-D_i)y_i = 8.062$. It thus seems that most of the bias in the estimated effect stems from the fact that the treatment has been allocated to individuals with lower than expected outcomes with the treatment, be it because they did not react strongly to the treatment, or because they were in worse shape without the treatment. We can check which one of these two explanations is more important. The true average effect of the treatment is, in this sample, $\frac{1}{N}\sum_{i=1}^{N}(y_i^1 - y_i^0) = 0.171$ while, in the treated group, this quantity is $\frac{1}{\sum_{i=1}^{N} D_i}\sum_{i=1}^{N} D_i(y_i^1 - y_i^0) = 0.18$. The true average potential outcome without the treatment is, in this sample, $\frac{1}{N}\sum_{i=1}^{N} y_i^0 = 8.036$ while, in the treated group, this quantity is $\frac{1}{\sum_{i=1}^{N} D_i}\sum_{i=1}^{N} D_i y_i^0 = 7.933$. The reason for the poor performance of the WW estimator in this sample is that individuals with lower counterfactual outcomes were included in the treated group, not that the treatment had lower effects on them. The bad counterfactual outcomes of the treated generates a bias of -0.103, while the bias due to heterogeneous reactions to the treatment is of 0.009. The last part of the bias is the one due to the fact that the individuals in the control group have slightly better counterfactual outcomes than in the sample: -0.026. The sum of these three terms yields the total bias of our WW estimator in this second sample: -0.12.

Let's now assess the overall effect of sampling noise on the estimate of the sample treatment effect for various sample sizes. In order to do this, I am going to use parallelized Monte Carlo simulations again. For the sake of simplicity, I am going to generate the same potential outcomes in each replication, using the same seed, and only choose a different treatment allocation.

```r
monte.carlo.ww.sample <- function(s,N,param){
  set.seed(1234)
  mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
  UB <- rnorm(N,0,sqrt(param["sigma2U"]))
  yB <- mu + UB
  YB <- exp(yB)
  epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
  eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
  U0 <- param["rho"]*UB + epsilon
  y0 <- mu +  U0 + param["delta"]
  alpha <- param["baralpha"]+  param["theta"]*mu + eta
  y1 <- y0+alpha
  Y0 <- exp(y0)
  Y1 <- exp(y1)
  set.seed(s)
  Ds <- rep(0,N)
  V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
  Ds[V<=log(param["barY"])] <- 1
  y <- y1*Ds+y0*(1-Ds)
  Y <- Y1*Ds+Y0*(1-Ds)
  return((1/sum(Ds))*sum(y*Ds)-(1/sum(1-Ds))*sum(y*(1-Ds)))
}

simuls.ww.N.sample <- function(N,Nsim,param){
  return(unlist(lapply(1:Nsim,monte.carlo.ww.sample,N=N,param=param)))
}

sf.simuls.ww.N.sample <- function(N,Nsim,param){
  sfInit(parallel=TRUE,cpus=ncpus)
  sim <- sfLapply(1:Nsim,monte.carlo.ww.sample,N=N,param=param)
  sfStop()
  return(unlist(sim))
}

simuls.ww.sample <- lapply(N.sample,sf.simuls.ww.N.sample,Nsim=Nsim,param=param)

monte.carlo.ate.sample <- function(N,s,param){
  set.seed(s)
  mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
  UB <- rnorm(N,0,sqrt(param["sigma2U"]))
  yB <- mu + UB
  YB <- exp(yB)
  epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
  eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
  U0 <- param["rho"]*UB + epsilon
  y0 <- mu +  U0 + param["delta"]
  alpha <- param["baralpha"]+  param["theta"]*mu + eta
  y1 <- y0+alpha
  Y0 <- exp(y0)
  Y1 <- exp(y1)
  Ds <- rep(0,N)
  V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
  Ds[V<=log(param["barY"])] <- 1
```

```r
  y <- y1*Ds+y0*(1-Ds)
  Y <- Y1*Ds+Y0*(1-Ds)
  return(mean(alpha))
}

par(mfrow=c(2,2))
for (i in 1:4){
  hist(simuls.ww.sample[[i]],main=paste('N=',as.character(N.sample[i])),xlab=expression(hat(Delta^yWW))
  abline(v=monte.carlo.ate.sample(N.sample[[i]],1234,param),col="red")
}
```
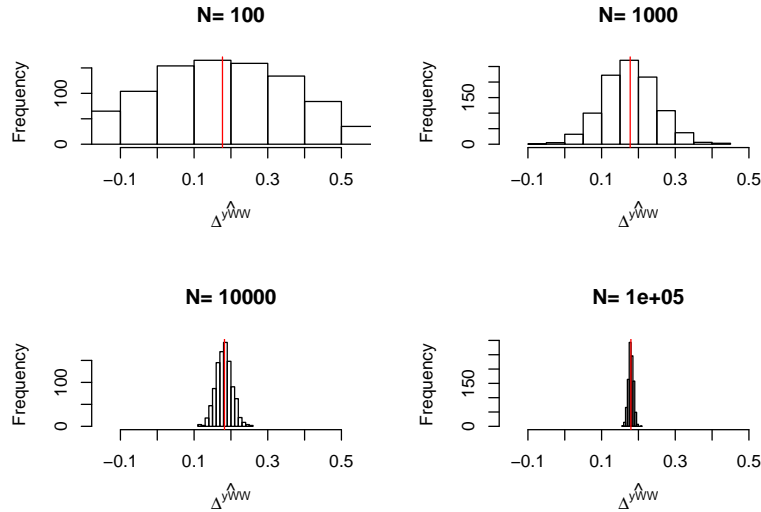

Figure 2.3: Distribution of the WW estimator over replications of treatment allocation for samples of different sizes

Let's also compute sampling noise, precision and the signal to noise ratio in these examples.

```r
samp.noise.sample <- function(i,delta,param){
  return(2*quantile(abs(monte.carlo.ate.sample(1234,N.sample[[i]],param)-simuls.ww.sample[[i]]),prob=del
}
samp.noise.ww.sample <- sapply(1:4,samp.noise.sample,delta=delta,param=param)
names(samp.noise.ww.sample) <- N.sample

precision.sample <- function(i,delta,param){
  return(1/samp.noise.sample(i,delta,param=param))
}
signal.to.noise.sample <- function(i,delta,param){
  return(monte.carlo.ate.sample(1234,N.sample[[i]],param)/samp.noise.sample(i,delta,param=param))
}
precision.ww.sample <- sapply(1:4,precision.sample,delta=delta,param=param)
names(precision.ww.sample) <- N.sample
signal.to.noise.ww.sample <- sapply(1:4,signal.to.noise.sample,delta=delta,param=param)
names(signal.to.noise.ww.sample) <- N.sample
table.noise.sample <- cbind(samp.noise.ww.sample,precision.ww.sample,signal.to.noise.ww.sample)
colnames(table.noise.sample) <- c('Sampling noise', 'Precision', 'Signal to noise ratio')
knitr::kable(table.noise.sample,caption=paste('Sampling noise of $\\hat{WW}$ for the sample treatment e
```

Finally, let's compare the extent of sampling noise for the population and the sample treatment effect parameters.

Table 2.2: Sampling noise of $\hat{W}W$ for the sample treatment effect with $\delta = 0.99$ and for various sample sizes

	Sampling noise	Precision	Signal to noise ratio
100	1.208	0.828	0.149
1000	0.366	2.729	0.482
10000	0.122	8.218	1.585
1e+05	0.033	30.283	5.453

```r
colnames(table.noise.sample) <- c('sampling.noise', 'precision', 'signal.to.noise')
table.noise.sample <- as.data.frame(table.noise.sample)
table.noise.sample$N <- as.numeric(rownames(table.noise.sample))
table.noise.sample$TT <- sapply(N.sample,monte.carlo.ate.sample,s=1234,param=param)
table.noise.sample$Type <- 'TTs'
table.noise$Type <- 'TT'
table.noise.tot <- rbind(table.noise,table.noise.sample)
table.noise.tot$Type <- factor(table.noise.tot$Type)

ggplot(table.noise.tot, aes(x=as.factor(N), y=TT,fill=Type)) +
  geom_bar(position=position_dodge(), stat="identity", colour='black') +
  geom_errorbar(aes(ymin=TT-sampling.noise/2, ymax=TT+sampling.noise/2), width=.2,position=position_dodg
  xlab("Sample Size")+
  theme_bw()+
  theme(legend.position=c(0.85,0.88))
```
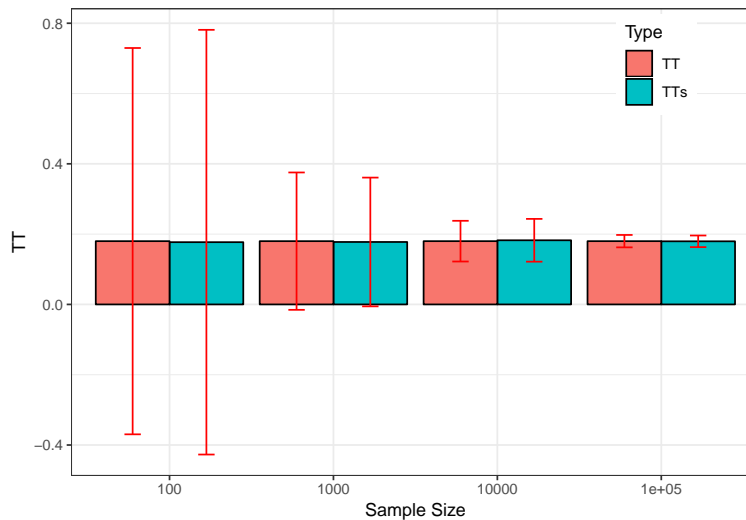

Figure 2.4: Sampling noise of $\hat{W}W$ (99% confidence) around TT and TT_s for various sample sizes

Figure 2.3 and Table 2.2 present the results of the simulations of sampling noise for the sample treatment effect parameter. Figure 2.4 compares sampling noise for the population and sample treatment effects.
For all practical purposes, the estimates of sampling noise for the sample treatment effect are extremely close to the ones we have estimated for the population treatment effect. I am actually surprised by this result, since I expected that keeping the potential outcomes constant over replications would decrease sampling noise. It seems that the variability in potential outcomes over replications of random allocations of the treatment in a given sample mimics very well the sampling process from a population. I do not know if this result of similarity of sampling noise for the population and sample treatment effect is a general one, but considering them as similar or close seems innocuous in our example.

2.1.4   Building confidence intervals from estimates of sampling noise

In real life, we do not observe TT. We only have access to $\hat{E}$. Let's also assume for now that we have access to an estimate of sampling noise, 2ϵ. How can we use these two quantities to assess the set of values that TT might take? One very useful device that we can use is the confidence interval. Confidence intervals are very useful because they quantify the zone within which we have a chance to find the true effect TT:

Theorem 2.1 (Confidence interval). *For a given level of confidence δ and corresponding level of sampling noise 2ϵ of the estimator $\hat{E}$ of TT, the confidence interval $\left\{\hat{E} - \epsilon, \hat{E} + \epsilon\right\}$ is such that the probability that it contains TT is equal to δ over sample replications:*

$$\Pr(\hat{E} - \epsilon \leq TT \leq \hat{E} + \epsilon) = \delta.$$

Proof. From the definition of sampling noise, we know that:

$$\Pr(|\hat{E} - TT| \leq \epsilon) = \delta.$$

Now:

$$\begin{aligned}
\Pr(|\hat{E} - TT| \leq \epsilon) &= \Pr(TT - \epsilon \leq \hat{E} \leq TT + \epsilon) \\
&= \Pr(-\hat{E} - \epsilon \leq -TT \leq -\hat{E} + \epsilon) \\
&= \Pr(\hat{E} - \epsilon \leq TT \leq \hat{E} + \epsilon),
\end{aligned}$$

which proves the result. $\square$

It is very important to note that confidence intervals are centered around $\hat{E}$ and not around TT. When estimating sampling noise and building Figure 2.2, we have centered our intervals around TT. The interval was fixed and $\hat{E}$ was moving across replications and 2ϵ was defined as the length of the interval around TT containing a proportion δ of the estimates $\hat{E}$. A confidence interval cannot be centered around TT, which is unknown, but is centered around $\hat{E}$, that we can observe. As a consequence, it is the interval that moves around across replications, and δ is the proportion of samples in which the interval contains TT.

Example 2.3. Let's see how confidence intervals behave in our numerical example.

```
N.plot <- 40
plot.list <- list()

for (k in 1:length(N.sample)){
  set.seed(1234)
  test <- sample(simuls.ww[[k]][,'WW'],N.plot)
  test <- as.data.frame(cbind(test,rep(samp.noise(simuls.ww[[k]][,'WW'],delta=delta)),rep(samp.noise(si
  colnames(test) <- c('WW','sampling.noise.1','sampling.noise.2')
  test$id <- 1:N.plot
  plot.test <- ggplot(test, aes(x=as.factor(id), y=WW)) +
      geom_bar(position=position_dodge(), stat="identity", colour='black') +
      geom_errorbar(aes(ymin=WW-sampling.noise.1/2, ymax=WW+sampling.noise.1/2), width=.2,position=posi
      geom_errorbar(aes(ymin=WW-sampling.noise.2/2, ymax=WW+sampling.noise.2/2), width=.2,position=posi
      geom_hline(aes(yintercept=delta.y.ate(param)), colour="#990000", linetype="dashed")+
      #ylim(-0.5,1.2)+
      xlab("Sample id")+
      theme_bw()+
```

```
      ggtitle(paste("N=",N.sample[k]))
  plot.list[[k]] <- plot.test
}
plot.CI <- plot_grid(plot.list[[1]],plot.list[[2]],plot.list[[3]],plot.list[[4]],ncol=1,nrow=length(N.sa
print(plot.CI)
```
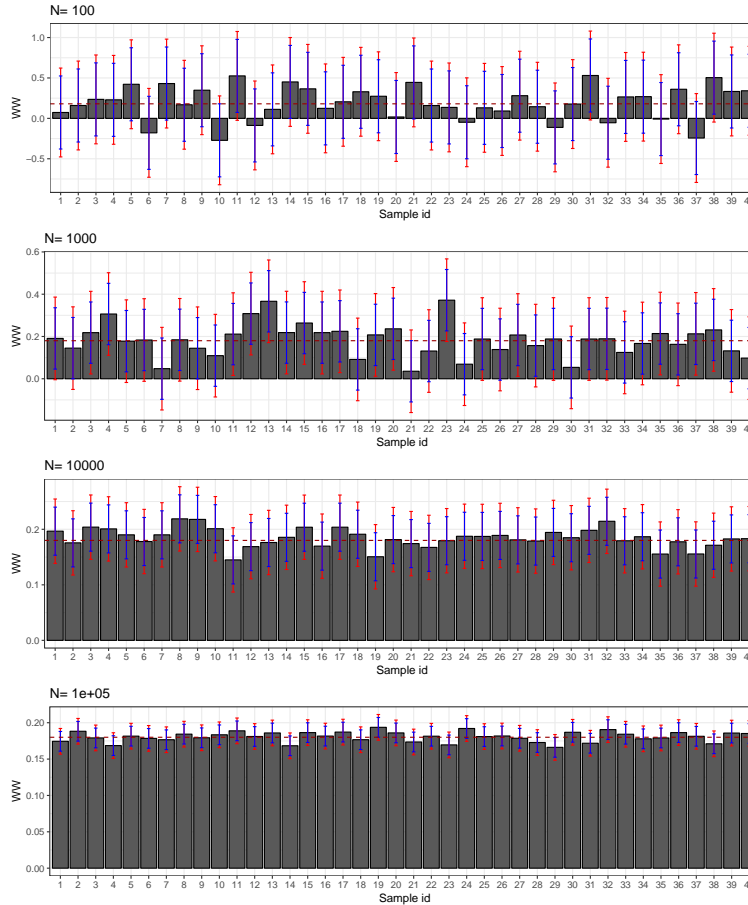

Figure 2.5: Confidence intervals of $\hat{WW}$ for $\delta = 0.99$ (red) and 0.95 (blue) over sample replications for various sample sizes

Figure 2.5 presents the 99% and 95% confidence intervals for 40 samples selected from our simulations. First, confidence intervals do their job: they contain the true effect most of the time. Second, the 95% confidence interval misses the true effect more often, as expected. For example, with $N = 1000$, the confidence intervals in samples 13 and 23 do not contain the true effect, but it is not far from their lower bound. Third, confidence intervals faithfully reflect what we can learn from our estimates at each sample size. With $N = 100$, the confidence intervals make it clear that the effect might be very large or very small, even strongly negative. With $N = 1000$, the confidence intervals suggest that the effect is either positive or null, but unlikely to be strongly negative. Most of the time, we get the sign right. With $N = 10^{4}$, we know that the true effect is bigger than 0.1 and smaller than 0.3 and most intervals place the true effect somewhere between 0.11 and 0.25. With $N = 10^{5}$, we know that the true effect is bigger than 0.15 and smaller than 0.21 and most intervals place the true effect somewhere between 0.16 and 0.20.

2.1.5   Reporting sampling noise: a proposal

Once sampling noise is measured (and we'll see how to get an estimate in the next section), one still has to communicate it to others. There are many ways to report sampling noise:

- Sampling noise as defined in this book ($2 * \epsilon$)
- The corresponding confidence interval
- The signal to noise ratio
- A standard error
- A significance level
- A p-value
- A t-statistic

The main problem with all of these approaches is that they do not express sampling noise in a way that is directly comparable to the magnitude of the TT estimate. Other ways of reporting sampling noise such as p-values and t-stats are nonlinear transforms of sampling noise, making it difficult to really gauge the size of sampling noise as it relates to the magnitude of TT.

My own preference goes to the following format for reporting results: $TT \pm \epsilon$. As such, we can readily compare the size of the noise to the sizee of the TT estimate. We can also form all the other ways of expressing sampling noise directly.

Example 2.4. Let's see how this approach behaves in our numerical example.

```
test.all <- list()
for (k in 1:length(N.sample)){
  set.seed(1234)
  test <- sample(simuls.ww[[k]][,'WW'],N.plot)
  test <- as.data.frame(cbind(test,rep(samp.noise(simuls.ww[[k]][,'WW'],delta=delta)),rep(samp.noise(si
  colnames(test) <- c('WW','sampling.noise.1','sampling.noise.2')
  test$id <- 1:N.plot
  test.all[[k]] <- test
}
```

With $N = 100$, the reporting of the results for sample 1 would be something like: "we find an effect of 0.07 ± 0.55." Note how the choice of δ does not matter much for the result. The previous result was for $\delta = 0.99$ while the result for $\delta = 0.95$ would have been: "we find an effect of 0.07 ± 0.45." The precise result changes with δ, but the qualitative result stays the same: the magnitude of sampling noise is large and it dwarfs the treatment effect estimate.

With $N = 1000$, the reporting of the results for sample 1 with $\delta = 0.99$ would be something like: "we find an effect of 0.19 ± 0.2." With $\delta = 0.95$: "we find an effect of 0.19 ± 0.15." Again, although the precise quantitative result is affected by the choice of δ, but hte qualitative message stays the same: sampling noise is of the same order of magnitude as the estimated treatment effect.

With $N = 10^{4}$, the reporting of the results for sample 1 with $\delta = 0.99$ would be something like: "we find an effect of 0.2 ± 0.06." With $\delta = 0.95$: "we find an effect of 0.2 ± 0.04." Again, see how the qualitative result is independent of the precise choice of δ: sampling noise is almost one order of magnitude smaller than the treatment effect estimate.

With $N = 10^{5}$, the reporting of the results for sample 1 with $\delta = 0.99$ would be something like: "we find an effect of 0.17 ± 0.02." With $\delta = 0.95$: "we find an effect of 0.17 ± 0.01." Again, see how the qualitative result is independent of the precise choice of δ: sampling noise is one order of magnitude smaller than the treatment effect estimate.

Remark. What I hope the example makes clear is that my proposed way of reporting results gives the same importance to sampling noise as it gives to the treatment effect estimate. Also, comparing them is easy, without requiring a huge computational burden on our brain.

Remark. One problem with the approach that I propose is when you have a non-symetric distribution of sampling noise, or when $TT \pm \epsilon$ exceeds natural bounds on TT (such as if the effect cannot be bigger than

one, for example). I think these issues are minor and rare and can be dealt with on a case by case basis. The advantage of having one simple and directly readable number comparable to the magnitude of the treatment effect is overwhelming and makes this approach the most natural and adequate, in my opinion.

2.1.6 Using effect sizes to normalize the reporting of treatment effects and their precision

When looking at the effect of a program on an outcome, we depend on the scaling on that outcome to appreciate the relative size of the estimated treatment effect.

It is often difficult to appreciate the relative importance of the size of an effect, even if we know the scale of the outcome of interest. One useful device to normalize the treatment effects is called Cohen's d, or effect size. The idea is to compare the magnitude of the treatment effect to an estimate of the usual amount of variation that the outcome undergoes in the population. The way to build Cohen's d is by dividing the estimated treatment effect by the standard deviation of the outcome. I generally prefer to use the standard deviation of the outcome in the control group, so as not to include the additional amoiunt of variation due to the heterogeneity in treatment effects.

Definition 2.2 (Cohen's d)**.** Cohen's d, or effect size, is the ratio of the estimated treatment effect to the standard deviation of outcomes in the control group:

$$d = \frac{\hat{TT}}{\sqrt{\frac{1}{N^0}\sum_{i=1}^{N^0}(Y_i - \bar{Y^0})^2}}$$

where $\hat{TT}$ is an estimate of the treatment effect, N^0 is the number of individuals in the treatment group and $\bar{Y^0}$ is the average outcome in the treatment group.

Cohen's d can be interpreted in terms of magnitude of effect size:

- It is generally considered that an effect is large when its d is larger than 0.8.
- An effect size around 0.5 is considered medium
- An effect size around 0.2 is considered to be small
- An effect size around 0.02 is considered to be very small.

There probably could be a rescaling of these terms, but that is the actual state of the art.

What I like about effect sizes is that they encourage an interpretation of the order of magnitude of the treatment effect. As such, they enable to include the information on precision by looking at which orders of magnitude are compatible with the estimated effect at the estimated precision level. Effect sizes and orders of magnitude help make us aware that our results might be imprecise, and that the precise value that we have estimated is probably not the truth. What is important is the range of effect sizes compatible with our results (both point estimate and precision).

Example 2.5. Let's see how Cohen's d behaves in our numerical example.

The value of Cohen's d (or effect size) in the population is equal to:

$$ES = \frac{TT}{\sqrt{V^0}} = \frac{\bar{\alpha} + \theta\bar{\mu}}{\sqrt{\sigma_\mu^2 + \rho^2\sigma_U^2 + \sigma_\epsilon^2}}$$

We can write a function to compute this parameter, as well as functions to implement its estimator in the simulated samples:

```
V0 <- function(param){
  return(param["sigma2mu"]+param["rho"]^2*param["sigma2U"]+param["sigma2epsilon"])
}
```

```r
ES <- function(param){
  return(delta.y.ate(param)/sqrt(V0(param)))
}

samp.noise.ES <- function(estim,delta,param=param){
  return(2*quantile(abs(delta.y.ate(param)/sqrt(V0(param))-estim),prob=delta))
}

for (i in 1:4){
  simuls.ww[[i]][,'ES'] <- simuls.ww[[i]][,'WW']/sqrt(simuls.ww[[i]][,'V0'])
}
```

The true effect size in the population is thus 0.2. It is considered to be small according to the current classification, although I'd say that a treatment able to move the outcomes by 20% of their usual variation is a pretty effective treatment, and this effect should be labelled at least medium. Let's stick with the classification though. In our example, the effect size does not differ much from the treatment effect since the standard deviation of outcomes in the control group is pretty close to one: it is equal to 0.88. Let's now build confidence intervals for the effect size and try to comment on the magnitudes of these effects using the normalized classification.

```r
N.plot.ES <- 40
plot.list.ES <- list()

for (k in 1:length(N.sample)){
  set.seed(1234)
  test.ES <- sample(simuls.ww[[k]][,'ES'],N.plot)
  test.ES <- as.data.frame(cbind(test.ES,rep(samp.noise.ES(simuls.ww[[k]][,'ES'],delta=delta,param=param
  colnames(test.ES) <- c('ES','sampling.noise.ES.1','sampling.noise.ES.2')
  test.ES$id <- 1:N.plot.ES
  plot.test.ES <- ggplot(test.ES, aes(x=as.factor(id), y=ES)) +
      geom_bar(position=position_dodge(), stat="identity", colour='black') +
      geom_errorbar(aes(ymin=ES-sampling.noise.ES.1/2, ymax=ES+sampling.noise.ES.1/2), width=.2,position
      geom_errorbar(aes(ymin=ES-sampling.noise.ES.2/2, ymax=ES+sampling.noise.ES.2/2), width=.2,position
      geom_hline(aes(yintercept=delta.y.ate(param)/sqrt(V0(param))), colour="#990000", linetype="dashed"
      #ylim(-0.5,1.2)+
      xlab("Sample id")+
      ylab("Effect Size")+
      theme_bw()+
      ggtitle(paste("N=",N.sample[k]))
  plot.list.ES[[k]] <- plot.test.ES
}
plot.CI.ES <- plot_grid(plot.list.ES[[1]],plot.list.ES[[2]],plot.list.ES[[3]],plot.list.ES[[4]],ncol=1,
print(plot.CI.ES)
```

Figure 2.6 presents the 99% and 95% confidence intervals for the effect size estimated in 40 samples selected from our simulations. Let's regroup our estimate and see how we could present their results.

```r
test.all.ES <- list()
for (k in 1:length(N.sample)){
  set.seed(1234)
  test.ES <- sample(simuls.ww[[k]][,'ES'],N.plot)
  test.ES <- as.data.frame(cbind(test.ES,rep(samp.noise.ES(simuls.ww[[k]][,'ES'],delta=delta,param=param
  colnames(test.ES) <- c('ES','sampling.noise.ES.1','sampling.noise.ES.2')
  test.ES$id <- 1:N.plot.ES
  test.all.ES[[k]] <- test.ES
```
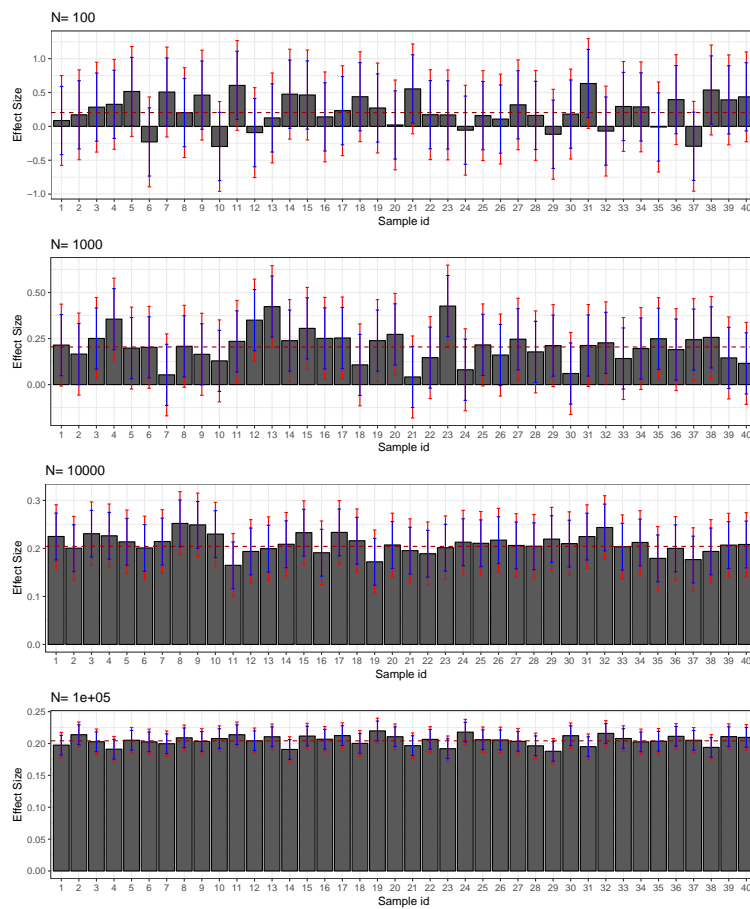
Figure 2.6: Confidence intervals of $\hat{ES}$ for $\delta = 0.99$ (red) and 0.95 (blue) over sample replications for various sample sizes

```
}
```

With $N = 100$, the reporting of the results for sample 1 would be something like: "we find an effect size of 0.09 ± 0.66" with $\delta = 0, 99$. With $\delta = 0.95$ we would say: "we find an effect of 0.09 ± 0.5." All in all, our estimate is compatible with the treatment having a large positive effect size and a medium negative effect size. Low precision prevents us from saying much else.

With $N = 1000$, the reporting of the results for sample 1 with $\delta = 0.99$ would be something like: "we find an effect size of 0.21 ± 0.22." With $\delta = 0.95$: "we find an effect size of 0.21 ± 0.17." Our estimate is compatible with a medium positive effect or a very small positive or even negative effect (depending on the choice of δ).

With $N = 10^{4}$, the reporting of the results for sample 1 with $\delta = 0.99$ would be something like: "we find an effect size of 0.22 ± 0.07." With $\delta = 0.95$: "we find an effect size of 0.22 ± 0.05." Our estimate is thus compatible with a small effect of the treatment. We can rule out that the effect of the treatment is medium since the upper bound of the 99% confidence interval is equal to 0.29. We can also rule out that the effect of the treatment is very small since the lower bound of the 99% confidence interval is equal to 0.16. With this sample size, we have been able to reach a precision level sufficient enough to pin down the order of magnitude of the effect size of our treatment. There still remains a considerable amount of uncertainty about the true effect size, though: the upper bound of our confidence interval is almost double the lower bound.

With $N = 10^{5}$, the reporting of the results for sample 1 with $\delta = 0.99$ would be something like: "we find an effect size of 0.2 ± 0.02." With $\delta = 0.95$: "we find an effect size of 0.2 ± 0.02." Here, the level of precision of our result is such that, first, it does not depend on the choice of δ in any meaningful way, and second, we can do more than pinpoint the order of magnitude of the effect size, we can start to zero in on its precise value. From our estimate, the true value of the effect size is really close to 0.2. It could be equal to 0.18 or 0.22, but not further away from 0.2 than that. Remember that is actually equal to 0.2.

Remark. One issue with Cohen's d is that its magnitude depends on the dispersion of the outcomes in the control group. That means that for the same treatment, and same value of the treatment effect, the effect size is larger in a population where oucomes are more homogeneous. This is not an attractive feature of a normalizing scale that its size depends on the particular application. One solution would be, for each outcome, to provide a standardized scale, using for example the estmated standard deviation in a reference population. This would be similar to the invention of the metric system, where a reference scale was agreed uppon once and for all.

Remark. Cohen's d is well defined for continuous outcomes. For discrete outcomes, the use of Cohen's d poses a series of problems, and alternatives such as relative risk ratios and odds ratios have been proposed. I'll comment on that in the last chapter.

2.2   Estimating sampling noise

Gauging the extent sampling noise is very useful in order to be able to determine how much we should trust our results. Are they precise, so that the true treatment effect lies very close to our estimate? Or are our results imprecise, the true treatment effect maybe lying very far from our estimate?

Estimating sampling noise is hard because we want to infer a property of our estimator over repeated samples using only one sample. In this lecture, I am going to introduce four tools that enable you to gauge sampling noise and to choose sample size. The four tools are Chebyshev's inequality, the Central Limit Theorem, resampling methods and Fisher's permutation method. The idea of all these methods is to use the properties of the sample to infer the properties of our estimator over replications. Chebyshev's inequality gives an upper bound on the sampling noise and a lower bound on sample size, but these bounds are generally too wide to be useful. The Central Limit Theorem (CLT) approximates the distribution of $\hat{E}$ by a normal distribution, and quantifies sampling noise as a multiple of the standard deviation. Resampling methods use the sample as a population and draw new samples from it in order to approximate sampling noise. Fisher's permutation method, also called randomization inference, derives the distribution of $\hat{E}$ under the assumption that all treatment effects are null, by reallocating the treatment indicator among the treatment and control group.

Both the CLT and resampling methods are approximation methods, and their approximation of the true extent of sampling noise gets better and better as sample size increases. Fisher's permutation method is exact-it is not an approximation-but it only works for the special case of the *WW* estimator in a randomized design.

The remaining of this section is structured as follows. Section 2.2.1 introduces the assumptions that we will need in order to implement the methods. Section 2.2.2 presents the Chebyshev approach to gauging sampling noise and choosing sample size. Section 2.2.3 introduces the CLT way of approximating sampling noise and choosing sample size. Section 2.2.4 presents the resampling methods. Section 2.2.5 introduces the Fisher's exact permutation approach.

Remark. I am going to derive the estimators for the precision only for the *WW* estimator. In the following lectures, I will show how these methods adapt to other estimators.

2.2.1 Assumptions

In order to be able to use the theorems that power up the methods that we are going to use to gauge sampling noise, we need to make some assumptions on the properties of the data. The main assumptions that we need are that the estimator identifies the true effect of the treatment in the population, that the estimator is well-defined in the sample, that the observations in the sample are independently and identically distributed (i.i.d.), that there is no interaction between units and that the variances of the outcomes in the treated and untreated group are finite.

We know from last lecture that for the *WW* estimator to identify *TT*, we need to assume that there is no selection bias, as stated in Assumption 1.7. One way to ensure that this assumption holds is to use a RCT.

In order to be able to form the *WW* estimator in the sample, we also need that there is at least one treated and one untreated in the sample:

Definition 2.3 (Full rank)**.** We assume that there is at least one observation in the sample that receives the treatment and one observation that does not receive it:

$$\exists i, j \leq N \text{ such that } D_i = 1 \& D_j = 0.$$

One way to ensure that this assumption holds is to sample treated and untreated units.

In order to be able to estimate the variance of the estimator easily, we assume that the observations come from random sampling and are i.i.d.:

Definition 2.4 (i.i.d. sampling)**.** We assume that the observations in the sample are identically and independently distributed:

$$\forall i, j \leq N, \, i \neq j, \, (Y_i, D_i) \perp\!\!\!\perp (Y_j, D_j),$$
$$(Y_i, D_i) \& (Y_j, D_j) \sim F_{Y,D}.$$

We have to assume something on how the observations are related to each other and to the population. Identical sampling is natural in the sense that we are OK to assume that the observations stem from the same population model. Independent sampling is something else altogether. Independence means that the fates of two closely related individuals are assumed to be independent. This rules out two empirically relevant scenarios:

1. The fates of individuals are related because of common influences, as for example the environment, etc,
2. The fates of individuals are related because they directly influence each other, as for example on a market, but also for example because there are diffusion effects, such as contagion of deseases or technological adoption by imitation.

We will address both sources of failure of the independence assumption in future lectures.

Finally, in order for all our derivations to make sense, we need to assume that the outcomes in both groups have finite variances, otherwise sampling noise is going to be too extreme to be able to estimate it using the methods developed in this lecture:

Definition 2.5 (Finite variance of $\Delta_{WW}^{\hat{Y}}$). We assume that $\mathbb{V}[Y^1|D_i = 1]$ and $\mathbb{V}[Y^0|D_i = 0]$ are finite.

2.2.2   Using Chebyshev's inequality

Chebyshev's inequality is a fundamental building block of statistics. It relates the sampling noise of an estimator to its variance. More precisely, it derives an upper bound on the samplig noise of an unbiased estimator:

Theorem 2.2 (Chebyshev's inequality). *For any unbiased estimator $\hat{\theta}$, sampling noise level 2ϵ and confidence level δ, sampling noise is bounded from above:*

$$2\epsilon \leq 2\sqrt{\frac{\mathbb{V}[\hat{\theta}]}{1 - \delta}}.$$

Remark. The more general version of Chebyshev's inequality that is generally presented is as follows:

$$\Pr(|\hat{\theta} - \mathbb{E}[\hat{\theta}]| > \epsilon) \leq \frac{\mathbb{V}[\hat{\theta}]}{\epsilon^2}.$$

The version I present in Theorem 2.2 is adapted to the bouding of sampling noise for a given confidence level, while this version is adapted to bounding the confidence level for a given level of sampling noise. In order to go from this general version to Theorem 2.2, simply remember that, for an unbiased estimator, $\mathbb{E}[\hat{\theta}] = \theta$ and that, by definition of sampling noise, $\Pr(|\hat{\theta} - \theta| > \epsilon) = 1 - \delta$. As a result, $1 - \delta \leq \mathbb{V}[\hat{\theta}]/\epsilon^2$, hence the result in Theorem 2.2.

Using Chebyshev's inequality, we can obtain an upper bound on the sampling noise of the WW estimator:

Theorem 2.3 (Upper bound on the sampling noise of $\hat{W}W$). *Under Assumptions 1.7, 2.3 and 2.4, for a given confidence level δ, the sampling noise of the $\hat{W}W$ estimator is bounded from above:*

$$2\epsilon \leq 2\sqrt{\frac{1}{N(1 - \delta)}\left(\frac{\mathbb{V}[Y_i^1|D_i = 1]}{\Pr(D_i = 1)} + \frac{\mathbb{V}[Y_i^0|D_i = 0]}{1 - \Pr(D_i = 1)}\right)} \equiv 2\bar{\epsilon}.$$

Proof. See in appendix **??**   □

Theorem 2.3 is a useful step forward for estimating sampling noise. Theorem 2.3 states that the actual level of sampling noise of the $\hat{W}W$ estimator (2ϵ) is never bigger than a quantity that depends on sample size, confidence level and on the variances of outcomes in the treated and control groups. We either know all the components of the formula for $2\bar{\epsilon}$ or we can estimate them in the sample. For example, $\Pr(D_i = 1)$, $\mathbb{V}[Y_i^1|D_i = 1]$ and $\mathbb{V}[Y_i^0|D_i = 0]$ by can be approximated by, respectively:

$$\Pr(\hat{D_i} = 1) = \frac{1}{N} \sum_{i=1}^{N} D_i$$

$$\mathbb{V}[Y_i^1 | \hat{D_i} = 1] = \frac{1}{\sum_{i=1}^{N} D_i} \sum_{i=1}^{N} D_i(Y_i - \frac{1}{\sum_{i=1}^{N} D_i} \sum_{i=1}^{N} D_i Y_i)^2$$

$$\mathbb{V}[Y_i^0 | \hat{D_i} = 0] = \frac{1}{\sum_{i=1}^{N}(1 - D_i)} \sum_{i=1}^{N}(1 - D_i)(Y_i - \frac{1}{\sum_{i=1}^{N}(1 - D_i)} \sum_{i=1}^{N}(1 - D_i)Y_i)^2.$$

Using these approximations for the quantities in the formula, we can compute an estimate of the upper bound on sampling noise, $\hat{2\epsilon}$.

Example 2.6. Let's write an R function that is going to compute an estimate for the upper bound of sampling noise for any sample:

```r
samp.noise.ww.cheb <- function(N,delta,v1,v0,p){
  return(2*sqrt((v1/p+v0/(1-p))/(N*(1-delta))))
}
```

Let's estimate this upper bound in our usual sample:

```r
set.seed(1234)
N <-1000
delta <- 0.99
mu <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]))
UB <- rnorm(N,0,sqrt(param["sigma2U"]))
yB <- mu + UB
YB <- exp(yB)
Ds <- rep(0,N)
V <- rnorm(N,param["barmu"],sqrt(param["sigma2mu"]+param["sigma2U"]))
Ds[V<=log(param["barY"])] <- 1
epsilon <- rnorm(N,0,sqrt(param["sigma2epsilon"]))
eta<- rnorm(N,0,sqrt(param["sigma2eta"]))
U0 <- param["rho"]*UB + epsilon
y0 <- mu +  U0 + param["delta"]
alpha <- param["baralpha"]+  param["theta"]*mu + eta
y1 <- y0+alpha
Y0 <- exp(y0)
Y1 <- exp(y1)
y <- y1*Ds+y0*(1-Ds)
Y <- Y1*Ds+Y0*(1-Ds)
```

In our sample, for $\delta = 0.99$, $\hat{2\epsilon} = 1.35$. How does this compare with the true extent of sampling noise when $N = 1000$? Remember that we have computed an estimate of sampling noise out of our Monte Carlo replications. In Table 2.2, we can read that sampling noise is actually equal to 0.39. The Chebyshev upper bound overestimates the extent of sampling noise by 245%.

How does the Chebyshev upper bound fares overall? In order to know that, let's compute the Chebyshev upper bound for all the simulated samples. You might have noticed that, when running the Monte Carlo simulations for the population parameter, I have not only recovered $\hat{WW}$ for each sample, but also the estimates of the components of the formula for the upper bound on sampling noise. I can thus easily compute the Chebyshev upper bound on sampling noise for each replication.

```r
for (k in (1:length(N.sample))){
  simuls.ww[[k]]$cheb.noise <- samp.noise.ww.cheb(N.sample[[k]],delta,simuls.ww[[k]][,'V1'],simuls.ww[[
}
```

```r
par(mfrow=c(2,2))
for (i in 1:4){
  hist(simuls.ww[[i]][,'cheb.noise'],main=paste('N=',as.character(N.sample[i])),xlab=expression(hat(2*ba
  abline(v=table.noise[i,colnames(table.noise)=='sampling.noise'],col="red")
}
```
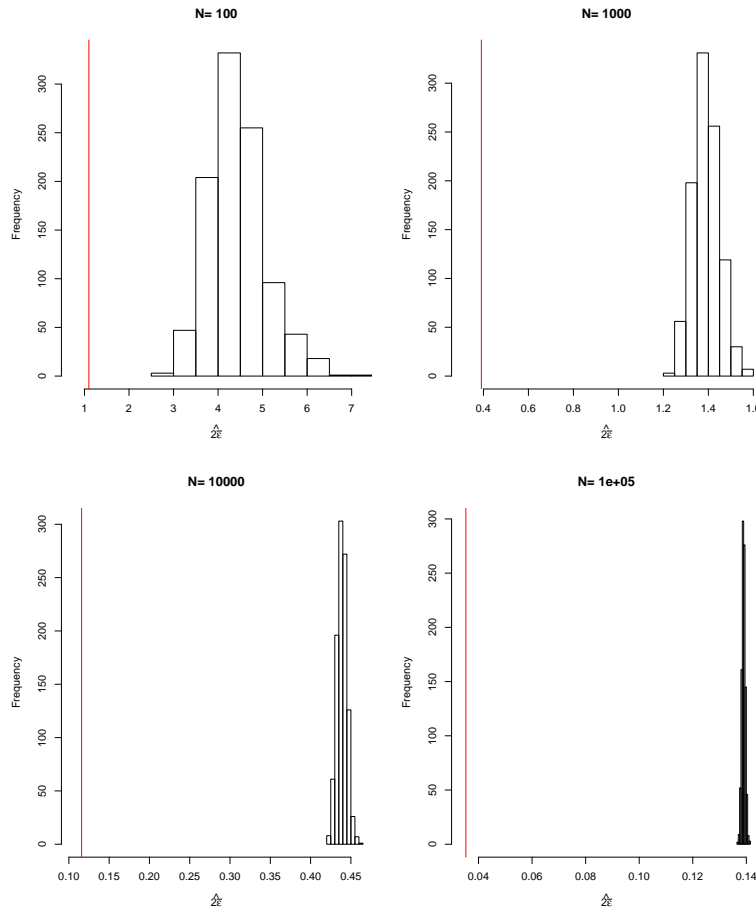

Figure 2.7: Distribution of the Chebyshev upper bound on sampling noise over replications of samples of different sizes (true sampling noise in red)

Figure 2.7 shows that the upper bound works: it is always bigger than the true sampling noise. Figure 2.7 also shows that the upper bound is large: it generally is of an order of magnitude bigger than the true sampling noise, and thus offers a blurry and too pessimistic view of the precision of an estimator. Figure 2.8 shows that the average Chebyshev bound gives an inflated estimate of sampling noise. Figure 2.9 shows that the Chebyshev confidence intervals are clearly less precise than the true unknown ones. With $N = 1000$, the true confidence intervals generally reject large negative effects, whereas the Chebyshev confidence intervals do not rule out this possibility. With $N = 10^{4}$, the true confidence intervals generally reject effects smaller than 0.1, whereas the Chebyshev confidence intervals cannot rule out small negative effects.

As a conclusion on Chebyshev estimates of sampling noise, their advantage is that they offer an upper bound on the noise: we can never underestimate noise if we use them. A downside of Chebyshev sampling noise is their lower precision, which makes it harder to pinpoint the true confidence intervals.

```r
for (k in (1:length(N.sample))){
  table.noise$cheb.noise[k] <- mean(simuls.ww[[k]]$cheb.noise)
}
```

```
ggplot(table.noise, aes(x=as.factor(N), y=TT)) +
  geom_bar(position=position_dodge(), stat="identity", colour='black') +
  geom_errorbar(aes(ymin=TT-sampling.noise/2, ymax=TT+sampling.noise/2), width=.2,position=position_dod
  geom_errorbar(aes(ymin=TT-cheb.noise/2, ymax=TT+cheb.noise/2), width=.2,position=position_dodge(.9),c
  xlab("Sample Size")+
  theme_bw()
```
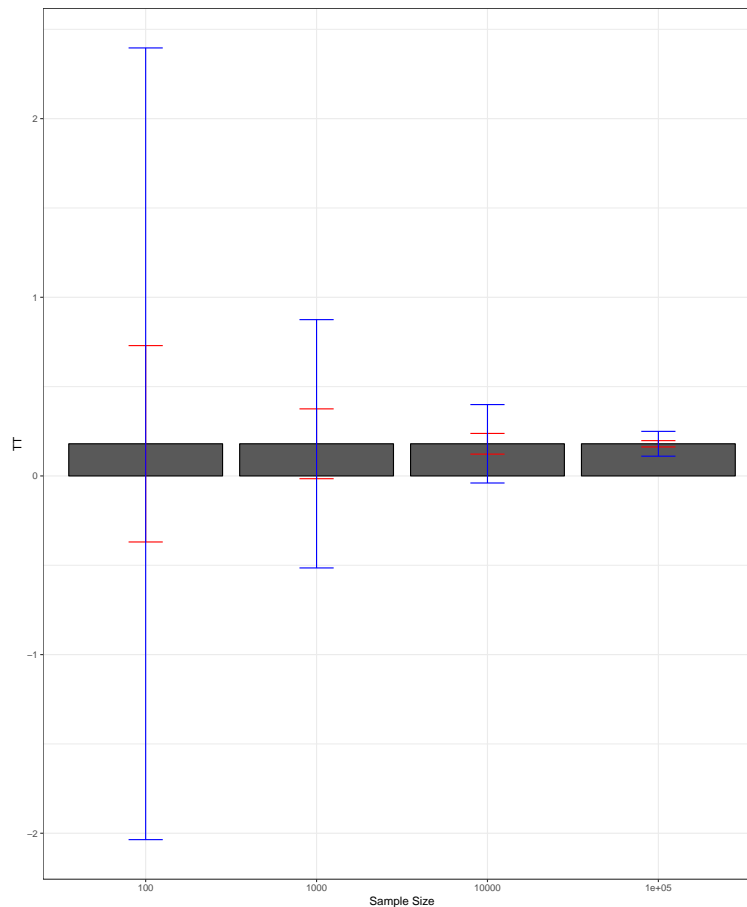

Figure 2.8: Average Chebyshev upper bound on sampling noise over replications of samples of different sizes (true sampling noise in red)

```
N.plot <- 40
plot.list <- list()

for (k in 1:length(N.sample)){
  set.seed(1234)
  test.cheb <- simuls.ww[[k]][sample(N.plot),c('WW','cheb.noise')]
  test.cheb <- as.data.frame(cbind(test.cheb,rep(samp.noise(simuls.ww[[k]][,'WW'],delta=delta),N.plot))
  colnames(test.cheb) <- c('WW','cheb.noise','sampling.noise')
  test.cheb$id <- 1:N.plot
  plot.test.cheb <- ggplot(test.cheb, aes(x=as.factor(id), y=WW)) +
      geom_bar(position=position_dodge(), stat="identity", colour='black') +
      geom_errorbar(aes(ymin=WW-sampling.noise/2, ymax=WW+sampling.noise/2), width=.2,position=position_
      geom_errorbar(aes(ymin=WW-cheb.noise/2, ymax=WW+cheb.noise/2), width=.2,position=position_dodge(.9
      geom_hline(aes(yintercept=delta.y.ate(param)), colour="#990000", linetype="dashed")+
```

```
      xlab("Sample id")+
      theme_bw()+
      ggtitle(paste("N=",N.sample[k]))
  plot.list[[k]] <- plot.test.cheb
}
plot.CI <- plot_grid(plot.list[[1]],plot.list[[2]],plot.list[[3]],plot.list[[4]],ncol=1,nrow=length(N.sa
print(plot.CI)
```
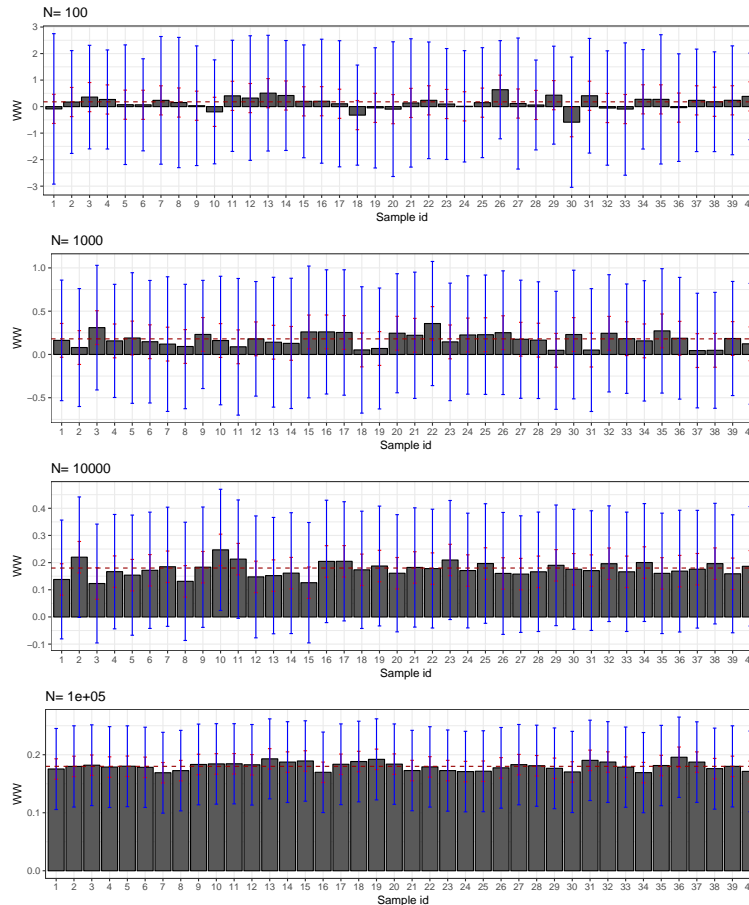

Figure 2.9: Chebyshev confidence intervals of $\hat{WW}$ for $\delta = 0.99$ over sample replications for various sample sizes (true confidence intervals in red)

2.2.3   Using the Central Limit Theorem

2.2.4   Using resampling methods

2.2.5   Using randomization inference