Applied Propensity Score Analysis with R

Jason Bryer, Ph.D.

2023-04-04

Contents

Preface

Last updated: April 04, 2023

I was first introduced to propensity score analysis (PSA) by my late dissertation advisor Robert Pruzek in 2006 when I entered graduate school. The notion that you could get reasonable causal estimates without the need of randomization was foreign to me and at first, I was skeptical. Many years later having used PSA for many projects, not only am I convinced it is possible, I believe there are instances where this may be preferred over the randomized control trial. I have been the Principal Investigator for two Federal grants to develop and test the Diagnostic Assessment and Achievement of College Skills (DAACS) where have attempted to conduct large scale randomized control trials (RCT) involving thousands of students. I have found through my experiences conducting these large scale RCTs that there are numerous compromises made in delivering an intervention that compromise the generalizability of the results. Moreover, RCTs assume a single, homogenous, causal effect for everyone. In reality this is rarely true. Not all interventions are equally effective for everyone. With PSA, particularly in the stratification section, it is possible to tease out how an intervention may vary by the observed covariates.

I have taught PSA many times over the years. This "book" is my attempt to collect my notes and experiences on conducting PSA. For the most part I will emphasize the applied and provide links to references if the reader wishes to explore the theoretical in more details. The `psa` R package that accompanies this book is available on Github and can be installed using the `remotes` package with the command below. By setting the `dependencies = 'Enhances'` parameter will ensure that all the R packages used in this book are installed as well. The `psa` package contains a number of datasets and utility functions used throughout the book. But it also contains a Shiny application designed to conduct PSA using a graphical user interface. Details on using the application are provided in the appendix.

```
remotes::install_github('jbryer/psa',
                        build_vignettes = TRUE,
                        dependencies = 'Enhances')
```

Contributing

This books is a work in progress and contributions are welcome. Please adhere to the code of conduct. Each page has an edit link which will take you directly to the source file on Github. You can also submit feedback using the Github Issues tracker.

Acknowledgements

This website was created using bookdown and is hosted by Github pages.

Colophon

```
devtools::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.2.1 (2022-06-23)
##  os       macOS Ventura 13.2.1
##  system   aarch64, darwin20
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       America/New_York
##  date     2023-04-04
##  pandoc   3.1.1 @ /opt/homebrew/bin/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------
##  package       * version  date (UTC) lib source
##  abind           1.4-5    2016-07-21 [1] CRAN (R 4.2.0)
##  backports       1.4.1    2021-12-13 [1] CRAN (R 4.2.0)
##  bookdown        0.32     2023-01-17 [1] CRAN (R 4.2.0)
##  boot            1.3-28.1 2022-11-22 [1] CRAN (R 4.2.0)
##  cachem          1.0.7    2023-02-24 [1] CRAN (R 4.2.0)
##  callr           3.7.3    2022-11-02 [1] CRAN (R 4.2.0)
##  car           * 3.1-1    2022-10-19 [1] CRAN (R 4.2.0)
##  carData       * 3.0-5    2022-01-06 [1] CRAN (R 4.2.0)
##  cli             3.6.1    2023-03-23 [1] CRAN (R 4.2.0)
##  codetools       0.2-18   2020-11-04 [1] CRAN (R 4.2.1)
##  coin            1.4-2    2021-10-08 [1] CRAN (R 4.2.0)
```

```
##  colorspace      2.1-0    2023-01-23 [1] CRAN (R 4.2.0)
##  crayon          1.5.2    2022-09-29 [1] CRAN (R 4.2.0)
##  devtools        2.4.5    2022-10-11 [1] CRAN (R 4.2.0)
##  digest          0.6.31   2022-12-11 [1] CRAN (R 4.2.0)
##  dplyr         * 1.1.1    2023-03-22 [1] CRAN (R 4.2.1)
##  ellipsis        0.3.2    2021-04-29 [1] CRAN (R 4.2.0)
##  evaluate        0.20     2023-01-17 [1] CRAN (R 4.2.0)
##  ez            * 4.4-0    2016-11-02 [1] CRAN (R 4.2.0)
##  fansi           1.0.4    2023-01-22 [1] CRAN (R 4.2.0)
##  fastmap         1.1.1    2023-02-24 [1] CRAN (R 4.2.0)
##  fs              1.6.1    2023-02-06 [1] CRAN (R 4.2.0)
##  generics        0.1.3    2022-07-05 [1] CRAN (R 4.2.0)
##  ggplot2       * 3.4.1    2023-02-10 [1] CRAN (R 4.2.0)
##  ggthemes        4.2.4    2021-01-20 [1] CRAN (R 4.2.0)
##  glue            1.6.2    2022-02-24 [1] CRAN (R 4.2.0)
##  granova       * 2.2      2023-03-22 [1] CRAN (R 4.2.0)
##  granovaGG     * 1.4.0    2015-12-18 [1] CRAN (R 4.2.0)
##  gridExtra       2.3      2017-09-09 [1] CRAN (R 4.2.0)
##  gtable          0.3.3    2023-03-21 [1] CRAN (R 4.2.0)
##  htmltools       0.5.4    2022-12-07 [1] CRAN (R 4.2.0)
##  htmlwidgets     1.6.1    2023-01-07 [1] CRAN (R 4.2.0)
##  httpuv          1.6.9    2023-02-14 [1] CRAN (R 4.2.0)
##  knitr         * 1.42     2023-01-25 [1] CRAN (R 4.2.0)
##  later           1.3.0    2021-08-18 [1] CRAN (R 4.2.0)
##  lattice         0.20-45  2021-09-22 [1] CRAN (R 4.2.1)
##  libcoin         1.0-9    2021-09-27 [1] CRAN (R 4.2.0)
##  lifecycle       1.0.3    2022-10-07 [1] CRAN (R 4.2.0)
##  lme4            1.1-32   2023-03-14 [1] CRAN (R 4.2.0)
##  magrittr        2.0.3    2022-03-30 [1] CRAN (R 4.2.0)
##  MASS          * 7.3-58.2 2023-01-23 [1] CRAN (R 4.2.0)
##  Matching      * 4.10-8   2022-11-03 [1] CRAN (R 4.2.0)
##  MatchIt       * 4.5.1    2023-02-23 [1] CRAN (R 4.2.1)
##  Matrix          1.5-3    2022-11-11 [1] CRAN (R 4.2.0)
##  matrixStats     0.63.0   2022-11-18 [1] CRAN (R 4.2.0)
##  memoise         2.0.1    2021-11-26 [1] CRAN (R 4.2.0)
##  mgcv            1.8-41   2022-10-21 [1] CRAN (R 4.2.0)
##  mime            0.12     2021-09-28 [1] CRAN (R 4.2.0)
##  miniUI          0.1.1.1  2018-05-18 [1] CRAN (R 4.2.0)
##  minqa           1.2.5    2022-10-19 [1] CRAN (R 4.2.0)
##  mnormt          2.1.1    2022-09-26 [1] CRAN (R 4.2.0)
##  modeltools      0.2-23   2020-03-05 [1] CRAN (R 4.2.0)
##  multcomp        1.4-23   2023-03-09 [1] CRAN (R 4.2.0)
##  multilevelPSA * 1.2.5    2018-03-22 [1] CRAN (R 4.2.0)
##  munsell         0.5.0    2018-06-12 [1] CRAN (R 4.2.0)
##  mvtnorm         1.1-3    2021-10-08 [1] CRAN (R 4.2.0)
##  nlme            3.1-161  2022-12-15 [1] CRAN (R 4.2.0)
```

```
##  nloptr          2.0.3    2022-05-26 [1] CRAN (R 4.2.0)
##  party           1.3-13   2023-03-17 [1] CRAN (R 4.2.0)
##  pillar          1.9.0    2023-03-22 [1] CRAN (R 4.2.1)
##  pkgbuild        1.4.0    2022-11-27 [1] CRAN (R 4.2.0)
##  pkgconfig       2.0.3    2019-09-22 [1] CRAN (R 4.2.0)
##  pkgload         1.3.2    2022-11-16 [1] CRAN (R 4.2.0)
##  plyr            1.8.8    2022-11-11 [1] CRAN (R 4.2.0)
##  prettyunits     1.1.1    2020-01-24 [1] CRAN (R 4.2.0)
##  processx        3.8.0    2022-10-26 [1] CRAN (R 4.2.0)
##  profvis         0.3.7    2020-11-02 [1] CRAN (R 4.2.0)
##  promises        1.2.0.1  2021-02-11 [1] CRAN (R 4.2.0)
##  ps              1.7.3    2023-03-21 [1] CRAN (R 4.2.0)
##  PSAboot       * 1.3.6    2023-03-22 [1] local
##  PSAgraphics   * 2.1.2    2023-03-21 [1] local
##  psych           2.3.3    2023-03-18 [1] CRAN (R 4.2.0)
##  purrr           1.0.1    2023-01-10 [1] CRAN (R 4.2.0)
##  R6              2.5.1    2021-08-19 [1] CRAN (R 4.2.0)
##  randomForest    4.7-1.1  2022-05-23 [1] CRAN (R 4.2.0)
##  RColorBrewer    1.1-3    2022-04-03 [1] CRAN (R 4.2.0)
##  Rcpp            1.0.10   2023-01-22 [1] CRAN (R 4.2.0)
##  remotes         2.4.2    2021-11-30 [1] CRAN (R 4.2.0)
##  reshape         0.8.9    2022-04-12 [1] CRAN (R 4.2.0)
##  reshape2      * 1.4.4    2020-04-09 [1] CRAN (R 4.2.0)
##  rlang           1.1.0    2023-03-14 [1] CRAN (R 4.2.0)
##  rmarkdown       2.20     2023-01-19 [1] CRAN (R 4.2.0)
##  rpart         * 4.1.19   2022-10-21 [1] CRAN (R 4.2.0)
##  rstudioapi      0.14     2022-08-22 [1] CRAN (R 4.2.0)
##  sandwich        3.0-2    2022-06-15 [1] CRAN (R 4.2.0)
##  scales        * 1.2.1    2022-08-20 [1] CRAN (R 4.2.0)
##  sessioninfo     1.2.2    2021-12-06 [1] CRAN (R 4.2.0)
##  shiny           1.7.4    2022-12-15 [1] CRAN (R 4.2.0)
##  stringi         1.7.12   2023-01-11 [1] CRAN (R 4.2.0)
##  stringr         1.5.0    2022-12-02 [1] CRAN (R 4.2.0)
##  strucchange     1.5-3    2022-06-15 [1] CRAN (R 4.2.0)
##  survival        3.5-0    2023-01-09 [1] CRAN (R 4.2.0)
##  TH.data         1.1-1    2022-04-26 [1] CRAN (R 4.2.0)
##  tibble          3.2.1    2023-03-20 [1] CRAN (R 4.2.1)
##  tidyselect      1.2.0    2022-10-10 [1] CRAN (R 4.2.0)
##  TriMatch      * 0.9.9    2017-12-06 [1] CRAN (R 4.2.0)
##  urlchecker      1.0.1    2021-11-30 [1] CRAN (R 4.2.0)
##  usethis         2.1.6    2022-05-25 [1] CRAN (R 4.2.0)
##  utf8            1.2.3    2023-01-31 [1] CRAN (R 4.2.0)
##  vctrs           0.6.1    2023-03-22 [1] CRAN (R 4.2.1)
##  withr           2.5.0    2022-03-03 [1] CRAN (R 4.2.0)
##  xfun            0.37     2023-01-31 [1] CRAN (R 4.2.0)
##  xtable        * 1.8-4    2019-04-21 [1] CRAN (R 4.2.0)
```

```
##  yaml            2.3.7    2023-01-23 [1] CRAN (R 4.2.0)
##  zoo             1.8-11   2022-09-17 [1] CRAN (R 4.2.0)
##
##  [1] /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library
##
## ------------------------------------------------------------------------------
```

License

Chapter 1

Introduction

The use of propensity score methods (**?**) for estimating causal effects in observational studies or certain kinds of quasi-experiments has been increasing over the last couple of decades (see Figure 1.1), especially in the social sciences (**?**) and medical research (**?**). Propensity score analysis (PSA) attempts to adjust selection bias that occurs due to the lack of randomization. Analysis is typically conducted in three phases where in phase I, the probability of placement in the treatment is estimated to identify matched pairs or clusters so that in phase II, comparisons on the dependent variable can be made between matched pairs or within clusters. Lastly, phase III involves testing the robustness of estimates to any unobserved confounders. R (**?**) is ideal for conducting PSA given its wide availability of the most current statistical methods vis-à-vis add-on packages as well as its superior graphics capabilities.

This book will provide a theoretical overview of propensity score methods as well as illustrations and discussion of implementing PSA methods in R. Chapter 1 provides an overview of all three phases of PSA with minimal R code. Chapters **??**, **??**, and **??** will discuss the details of implementing the three major approaches to PSA. Chapter **??** provides some strategies to conducting PSA when there is missing data. Chapters **??** and **??** provide details for phase III of PSA using sensitivity analysis and bootstrapping, respectively. Lastly, chapter **??** provides methods for implementing PSA with non-binary treatments and chapter **??** discusses methods for PSA with cluster, or Hierarchical, data. The appendices contain additional details regarding the PSA Shiny application (Appendix **??**), limitations of interpreting fitted values from logistic regression (Appendix **??**), and additional methods and packages for estimating propensity scores (Appendix **??**).

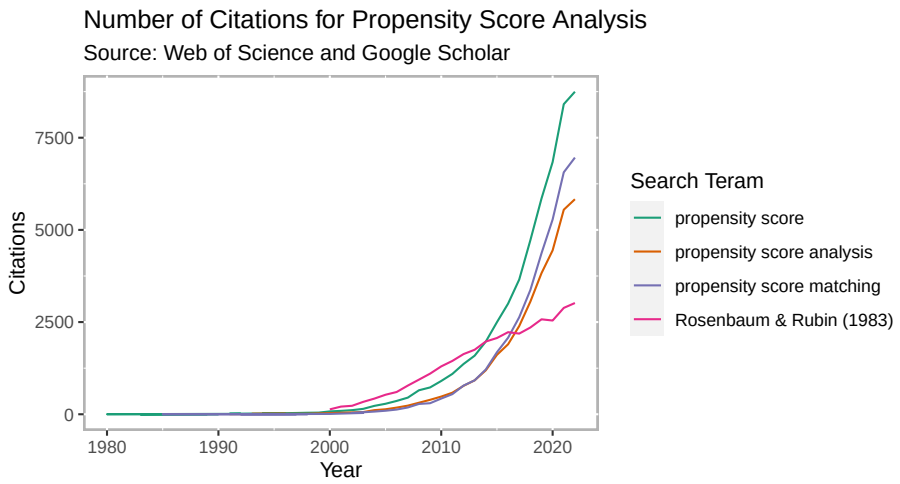

Figure 1.1: PSA Citations per year

1.1 Counterfactual Model for Causality

In order to understand how propensity score analysis allows us to make causal estimates from observational data, we must first understand the basic principals of causality, particularly the counterfactual model. Figure 1.2 depicts a conterfactual model. We begin with our research subject. This can be a student, patient, mouse, asteroid, or any other object we wish to know whether some condition has an effect on. Consider two parallel universes: one where the subject receives condition A and another where they receive condition B. Typically one condition is some treatment whereas the other condition is the absence of that treatment (also referred to as the control). We will use treatment and control throughout this book to refer to these two conditions. Once the individual has been exposed to the two conditions, the outcome is measured. The difference between these outcomes is the true causal effect. However, unless your Dr. Strange living in the Marvell multiverse, it is impossible for an object to exist in two universes at the same time, therefore we can never actually observe the true causal effect. **?** referred to this as the *Fundamental Problem of Causal Inference*.

1.2 Randomized Control Trials: "The Gold Standard"

The randomized control trials (RCT) has been the gold standard for estimating causal effects. Effects can be estimated using simple means between groups,

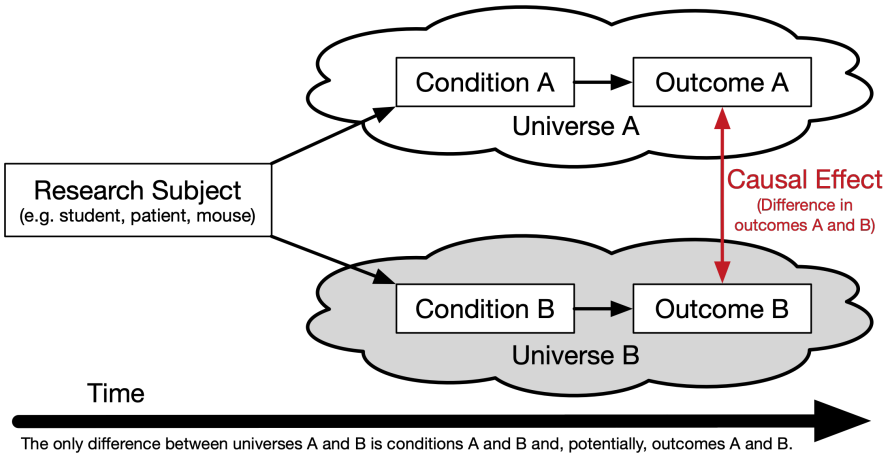

Figure 1.2: Theoretical Causal Model

or blocks in randomized block design. Randomization presumes unbiasedness and balance between groups. However, randomization is often not feasible for many reasons, especially in educational contexts. Although the RCT is the gold standard, it is important to recognize that it only *estimates* the causal effect. We will look at an example of where the RCT can be wrong and why on average it provides good estimates of the true causal effect so we can build a model to closely mimick the RCT with non-randomized data.

The Intelligence Quotient (IQ) is a common measure of intelligence. It is designed such that the mean is 100 and the standard deviation is 15. Consider we have developed an intervention that is known to increase anyone's IQ by 4.5 points (or a standardized effect size of 0.3). Figure 1.3 represents such a scenario with 30 individuals. The left panel has the individual's outcome if they were assigned to the control condition (in blue) and to the treatment condition (in red). The distance between the red and blue points for any individual is 4.5, our stipulated counterfactual difference. For RCTs we only ever get to observe one outcome for any individual. The right pane represents one possible set of outcomes from an RCT. That is, we randomly selected one outcome for each individual from the left pane.

Figure 1.4 includes the mean differences between treatment and control as vertical lines in blue and red, respectively. On the left where we observe the true counterfactuals the difference between the treatment (in red) and control (in blue) vertical lines is 4.5. However, on the right the difference between treatment and control is -5.3!

In this example not only did the RCT not estimate the true effect, it estimated in the wrong direction. However, Figure 1.5 represents the distribution of effects after conducting 1,000 RCTs from the 30 individuals above. The point here

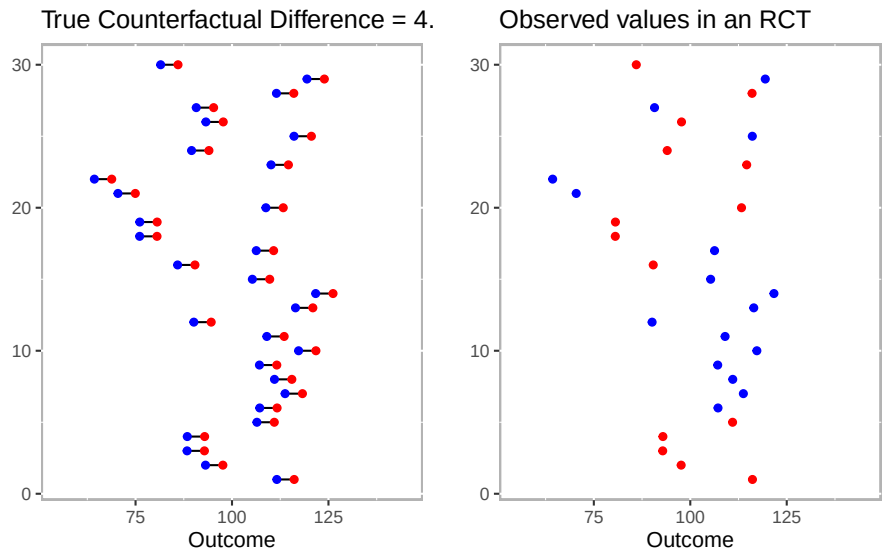

Figure 1.3: Example conterfactuals (left panel) with one possible randomized control trial.

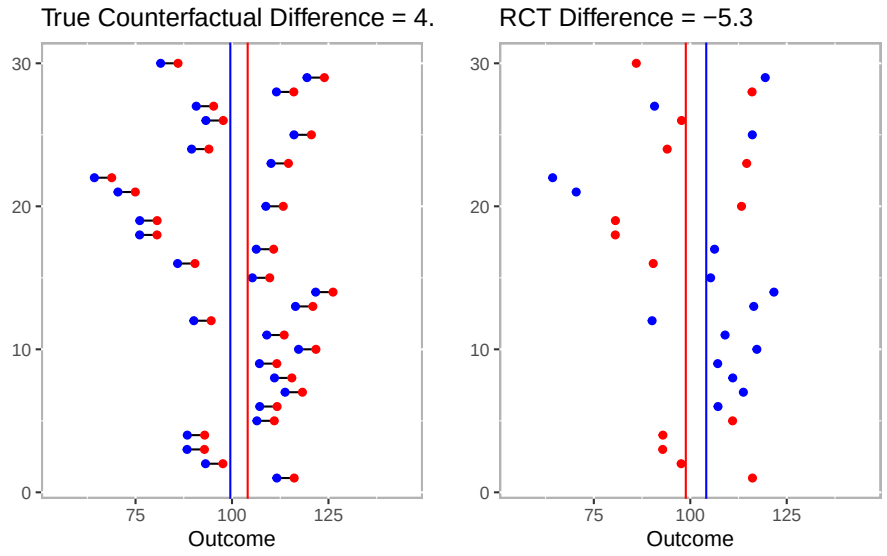

Figure 1.4: Estimated differences for full counterfactual model and one RCT.

is that the RCT is already compromise to estimating the true counterfactual (i.e. causal effect). It is consider the gold standard because over many trials it will nearly approximate the true counterfactual.

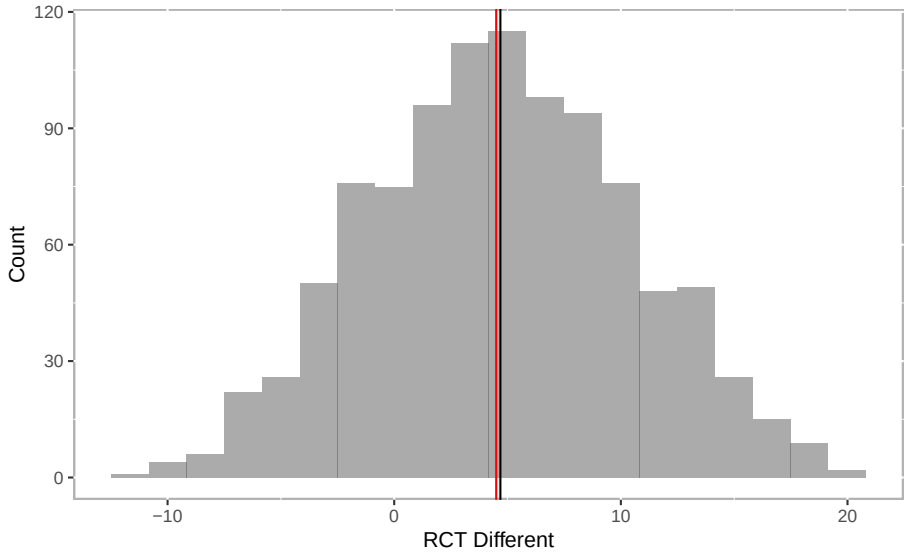

Figure 1.5: Distribution of differences across many RCTs

The RCT works because the probability of anyone being in the treatment is 50%. Statistically, we call this the strong ignorability assumption. The strong ignorability assumption states that an outcome is independent of any observed or unobserved covariates[1] under randomization. This is represented mathematically as:

$$\left(Y_i\left(1\right), Y_i\left(0\right)\right) \perp T_i \tag{1.1}$$

For all X_i Here, Y is our outcome of interest and i is an individual response such that $Y_i(1)$ is the outcome for subject i if assigned to the treatment group and $Y_i(0)$ is the outcome for subject i if assigned to the control group. The $\perp$ means independent and T_i is assignment indicator subject i. Therefore, it follows that the causal effect of a treatment is the difference in an individual's outcome under the situation they were given the treatment and not (referred to as a counterfactual).

$$\delta_i = Y_{i1} - Y_{i0} \tag{1.2}$$

[1]Covariates used in this book and in the context of propensity score analysis are the independent variables that influence statistical models for predicting treatment placement and outcomes.

However, it is impossible to directly observe \({ }_{i}\) (referred to as The Fundamental Problem of Causal Inference, Holland 1986). Rubin framed this problem as a missing data problem and the details will be discussed in the next section.

1.2.1 Rubin's Causal Model

Returning to Figure 1.2, the problem with getting a true causal effect is that we only observe outcome A **or** outcome B, never both. As a result, we are missing data to estimate the causal effect. **?** first coined the term *potential outcomes* when referring to randomized trials. However, Donald Rubin extended Neyman's idea to include both observational and experimental data. Rubin's student **?** later coined this the Rubin Causal Model.

? discussed an example of the effect of aspirin on a headache:

> "Intuitively, the causal effect of one treatment, E, over another, C, for a particular unit and an interval of time from t_1 to t_2 is the difference between what would have happened at time t_2 if the unit had been exposed to E initiated at t_1 and what would have happened at t_2 if the unit had been exposed to C initiated at t_1: 'If an hour ago I had taken two aspirins instead of just a glass of water, my headache would now be gone,' or 'because an hour ago I took two aspirins instead of just a glass of water, my headache is now gone.' Our definition of the causal effect of the E versus C treatment will reflect this intuitive meaning."

Under the Rubin Causal Model, whether or not you have a headache is the cause of whether or not your took aspirin one hour ago, but we can only observe one. The key to estimating the causal effect has to do with understanding the mechanism for the selecting whether or nor to take the aspirin. Imagine you get chronic headaches so you need to decide many times whether or not to take an aspirin. Let's also stimulate that the aspirin is more likely to be effective if you take it in the morning than the afternoon. If you decide to flip a coin to decide whether or not to take the aspirin there should be balance between observed headaches in morning and afternoon. That is, even though there is a difference between morning and afternoon, that does not influence the observed outcomes. However, you decide that you will take the aspirin only if it is above 50 degrees outside. Since it is more likely to be warmer in the afternoon then the morning, comparing the outcomes will provide a bias estimate, in part because deciding whether to take the aspirin is no longer 50%. But if we observed the weather we can potentially determine the probability of taking the aspirin or not. With enough observations, we compare situations where the probability of taking the aspirin was low, but there were some observations with and without aspirin all the way across the spectrum to where there was a high probability fo taking the aspirin.