2022

Correlation Does Not Imply Correlation: A Thesis on Causal Influence and Simpson's Paradox

Emily Naitoh

SCRIPPS
THE WOMEN'S COLLEGE
• CLAREMONT •

Correlation Does Not Imply Correlation: A Thesis on Causal Influence and Simpson's Paradox

Emily Naitoh

Christina Edholm, First Reader
Christopher Towse, Second Reader

Submitted to Scripps College in Partial Fulfillment
of the Degree of Bachelor of Arts

December 13, 2022

Department of Mathematics

Abstract

In our data-driven world, it has become commonplace to attempt to find causal relationships. One of the themes of this thesis is to show methods of determining causation. The second theme follows a saying in mathematics, "correlation does not imply causation". We will also discuss situations where correlation does not even imply correlation itself. These cases are described by Simpson's paradox in an exploration of different areas of mathematics and computer coding.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Professor Christina Edholm for guiding me through this thesis. We worked together every week to make progress toward the goal of completing my thesis. I also will thank Professor Christopher Towse for being my second thesis reader and giving helpful tips on the process. I'd also like to thank my family for financially assisting me in my education and encouraging me to graduate. Lastly, I would like to thank my boyfriend for supporting me every day and bringing me the necessary sustenance to complete the countless hours of sitting at my desk working on this thesis.

Chapter 1

Introduction

1.1 Importance

In this section we will go through an example of an issue caused by the ignorance of Simpson's paradox. The most famous incident of Simpson's paradox is from an instance at the University of California, Berkeley(20). The problem here stemmed from the accusation that admissions to the University had a gender bias. Overall, a greater percentage of males were accepted into the school, however, in each facility the opposite was true. **Table 1.1** shows example data involved in this case.

	Male		Female		Percent Male	Percent Female
	Accept	Reject	Accept	Reject		
Facility A	820	80	680	20	0.91	0.97
Facility B	20	80	100	200	0.2	0.33
Combined	840	160	780	220	0.84	0.78

Table 1.1 Example Table of Berkeley Admissions (7)

In this case, we have the same total number of females and males applying for the school overall. In both Facility A and Facility B there is a higher percentage of females accepted. In aggregate, however, there is a higher acceptance of males. The reasoning here is that female students choose the more difficult facility (Facility B) at a high rate. Overall, the implications here are that if Simpson's paradox was not detected UC Berkeley would be at risk of a lawsuit for gender discrimination (7). Such a lawsuit was attempted, fortunately, before it occurred they were able to figure out the true cause of this supposed discrimination. This example is mathematically easy to understand but cases further in this thesis become more complicated.

The thesis is organized as follows. We begin with a refresher on probability and statistics in Section 1.2. That is followed by Chapter 2 on causal influence. This chapter is split into three subsections. First is an introductory explanation on linear models in Section 2.1. The second section, Section 2.2, defines causal structures graphically using directed acyclic graphs. Then Section 2.3 goes through methods of intervention to find causality. Chapter 3 is on Simpson's paradox. It's first section, Section 3.1, defines Simpson's paradox through population and sub-population rates. Then Section 3.2 has definitions through probability. The last section, Section 3.3, has graphical examples of Simpson's paradox. Chapter 4 includes two algorithms for finding Simpson's paradox in data sets. Conclusions and future work are in Section 5.1 and Section 5.2 of Chapter 5. We then end the thesis with our Appendix A including source code and a bibliography.

In this thesis, a literature review was conducted, but only so many sources could be directly used and cited in the text. For further reading, we suggest the sources below. If you are interested in code go to source (1). For definitions of Simpson's paradox by rates you can read sources (3), (9), (18), and (20). Sources with probability include (4), (6), (9), (11), (13), (17), (18), (19), and (20). Graphical source include (1), (11), (18), (19), and (21). Lastly, here are sources using vectors for Simpsons's paradox (8), (9), (13), and (21). Other useful sources can be found by searching for Judea Pearl whose work at UCLA has been an integral part of my research.

If you would like to follow along with the code it is posted on github at *https://github.com/emnaitoh/Simpsons-Paradox-Senior-Thesis.*

1.2 Probability and Statistics

This section is a refresher on the statistics and probability needed to understand some of the definitions in further chapters. The majority of this section comes from ideas in Chapter 1 of the textbook *"Causal inference in statistics: A Primer"* by Judea Pearl, et al. (12) Here, we will go through basic probability. Probability is the mathematical way to describe uncertainty. For this thesis, we will need to understand some core ideas in probability and statistics. Let's quickly go through some definitions.

1.2.1 Probability

This section is on probability. It will be useful for both Chapters 2 and 3. It will be useful for rules and theorems in causality and our definitions of Simpson's paradox.

Definition 1.2.1 (Independence)**.** *Given two events, A and B, if*

$$P(A|B) = P(A)$$

and

$$P(A, B) = P(A)P(B)$$

then A and B are independent.

Independence means that the observation of event B does not change the probability of event A. The opposite of independence is dependence.

Definition 1.2.2 (Conditional Independence)**.** *Given three events, A, B, and C, if*

$$P(A|B, C) = P(A|C)$$

and

$$P(B|A, C) = P(B|C)$$

then A and B are conditionally independent given C.

Conditional Independence means that two dependent events become independent when there is a third event. Another way to think of it is if you only included data with event C then if A and B are independent then they are conditionally independent. Furthermore, if A and B are also independent with the whole dataset then they are marginally independent. Marginal probability is the probability of an even disregarding other outcomes of another variable.

Definition 1.2.3 (Probability Distribution)**.** *A probability distribution is a function (i.e. a curve or graph) that shows the probability of different values for a variable.*

In other words, it is how the probability of different events is distributed. Further, the sum of the probabilities is 1 and thus the area under the curve given by the probability distribution is 1.
Density function is similar but it is for continuous functions and takes the form of an integral. For multiple variables you can have joint distributions.

Definition 1.2.4 (Mutually Exclusive). *Given A and B are mutually exclusive events then;*

$$P(A \cup B) = P(A) + P(B)$$

Another way to understand this is that they are sets that do not overlap. For example, being a cat or a dog are mutually exclusive events since it is not possible to be in both groups. This is clear since the probability of both of them is the same as each individual combined i.e. $P(female \cup male) = P(female) + P(male)$.

The following definitions will help us understand the steps taken to get to Bayes' theorem (Definition 1.2.8).

Definition 1.2.5 (Partition). *A partition of events, $B_1, B_2, ...B_n$, is a set events that are mutually exclusive and include all probable outcomes.*

Definition 1.2.6 (Law of Total Probability). *For any two events A and B;*

$$P(A) = P(A, B) + P(A, \neg B)$$

Where ¬ means "not".
Furthermore, if we have a partition, $B_1, B_2, ...B_n$, then it is true that;

$$P(A) = P(A, B_1) + P(A, B_2) + ... + P(A, B_n)$$

and

$$P(A) = \sum_n P(A|B_n)P(B_n)$$

Through division we can get the following definition.

Definition 1.2.7 (Conditional Probability).

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

Putting the laws together we get Bayes' rule.

Definition 1.2.8 (Bayes' Rule)**.**

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

Bayes' rule is helpful for finding conditional probability given the reversed condition and the original probabilities.

Another important idea in probability is expected value or mean.

Definition 1.2.9 (Expected Value)**.**

$$E(X) = \sum_x xP(X = x)$$

The expected value of a variable X is the sum of values multiplied by their respective probabilities.

The expected value is exactly how it sounds, it is the value that you expect to get based on the outcomes and probabilities. An example is lets say you are at an arcade and ask you dad for some money for a claw machine that cost either 1 dollar or 50 cents. Claw machines are more likely to be 1 dollar than 50 cents, so lets say 75% to 25% respectively. Then the expected value of the amount your dad gives you is $0.75 \cdot 1 + 0.25 \cdot 0.5 = 0.875$, so about 88 cents.
Here are some more cases of expected value.

Definition 1.2.10 (Expected Value of a Function)**.** *If $g(X)$ is a function of X then,*

$$E[g(X)] = \sum_x g(x)P(x)$$

Definition 1.2.11 (Expected Value of Conditional)**.** *Lets say we have the conditional probability $P(Y = y|X = x)$ then the expected value is*

$$E(Y|X = x) = \sum_y yP(Y = y|X = x)$$

1.2.2 Statistics

Statistics is another necessary topic for this thesis. This subsection will be a refresher of basic statistics. One main use of statistics is comparing samples to populations. A population is an entire group that we are trying to understand. The sample is a subgroup randomly chosen from the population. A large part of statistics is trying to find information about the population using just a sample.

Let's begin with the simple concept of variance. Variance is a measure that describes the spread of all data points. Variance can be for a population or for a sample and there are more than one way to find variance.

Definition 1.2.12 (Population Variance)**.** *For population variance we have,*

$$Var(X) = \sigma_X^2 = E((X-\mu)^2) = \frac{\sum(x_i-\mu)^2}{N}$$

Where μ is the population mean and N is the population size. Variance is always positive.

Sample Variance is similar but with a slight change.

Definition 1.2.13 (Sample Variance)**.** *For sample variance we have,*

$$Var(X) = s^2 = \frac{\sum(x_i-\bar{x})^2}{n-1}$$

Where $n-1$ is called the degrees of freedom.

Definition 1.2.14 (Standard Deviation)**.** *Standard deviation for a population is denoted σ_X it is the square root of the variance, it has the same units as the variable X and is always positive. Similarly, $s=\sqrt{s^2}$ for the sample standard deviation.*

Standard deviation is a very useful tool in statistics because it can show an estimate of the distribution of a random variable X. It is known that in a normal distribution, like the one shown in **Figure1.1**, we have about 68% of the population within one standard deviation of the mean and 95% within two standard deviations and the large majority of 99.7% are within three.

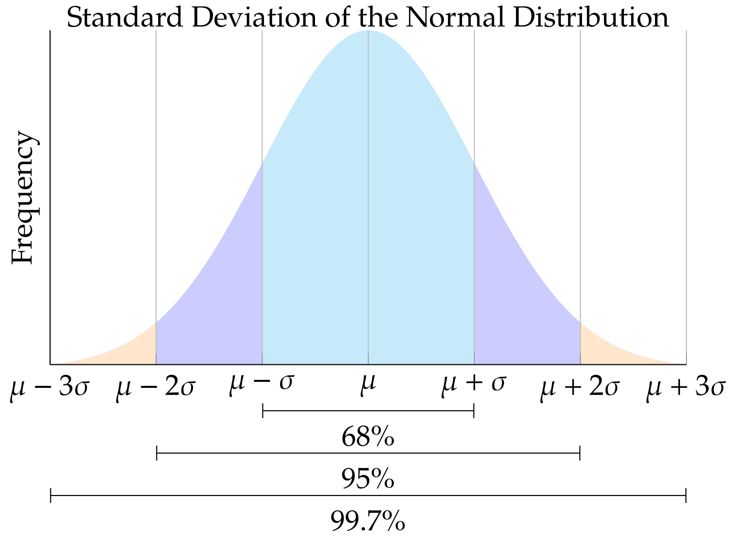

Figure 1.1 Frequency plotted by probability where μ is the mean and σ is the standard deviation(5)

Next we have covariance, it shows the association of how two variables X, Y vary together. The measure is how the variables linearly covary which is how close their relationship is to linear.

Definition 1.2.15 (Covariance)**.**

$$\sigma_{XY} = E[(X - E(X))(Y - E(Y))]$$

A normalized covariance is the correlation coefficient.

Definition 1.2.16 (Correlation Coefficient)**.**

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

The value above is for the population. The correlation coefficient for a sample is r. The correlation coefficient is between positive and negative 1 which is the slope of a line representing X and Y. This line is in the simplest form, $y = a + bx$. Here r represents b and the value a is the intercept. The figures below show example lines with differing correlation coefficients (r).

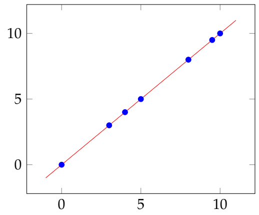

Figure 1.2 Positive Correlation $r = 1$

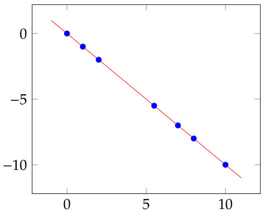

Figure 1.3 Negative Correlation $r = -1$

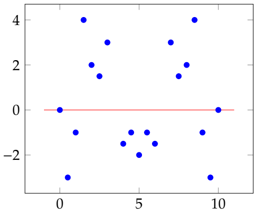

Figure 1.4 No Correlation $r = 0$

Another way of thinking about correlation is the amount that the values change together (i.e. do they both increase or decrease together, do they grow or shrink opposite of each other, is there no relationship). An issue with the correlation coefficient is that it is only useful for linear relationships. Of course, there are many other important relationships that are not linear, for example exponential which increases at a much faster rate. It is important to recognize nonlinear relationships before using the correlation coefficient.

A multiple linear regression is a regression that takes multiple predictor variables. It still returns a linear model but it is more complex. It still has slopes (partial regression coefficients) multiplied by predictor variables (X's) and an intercept (r_0).

Definition 1.2.17 (Partial Regression Coefficient)**.** *$R_{YX \cdot Z}$ is the slope of Y on X when we hold constant Z.*

To compute the *partial regression coefficient* you can show Y as a linear combination of Xs and add noise.

$$Y = r_0 + r_1X_1 + r_2X_2 + \cdots + r_kX_k + \epsilon$$

ϵ must be uncorrelated to the regressors X to obtain the best least-square coefficients (least squares is explained later in **Chapter 2.1)**.

$$Cov(\epsilon, X_i) = 0 \quad for \quad i = 1, 2 \cdots, k$$

The orthogonality principle helps compute $X = \alpha + \beta Y + \epsilon$

$$E[X] = \alpha + \beta E[Y]$$

$$E[XY] = \alpha E[Y] + \beta E[Y^2] + E[Y\epsilon]$$

Orthogonality principle says $E[Y\epsilon] = 0$. These result in the equations for α and β.

$$\alpha = E(X) - E(Y)\frac{\sigma_{XY}}{\sigma_Y^2}$$

$$\beta = \frac{\sigma_{XY}}{\sigma_Y^2}$$

The next section will deal with models that are more directly relevant to causality, including types of structural causal models(12).

Chapter 2

Casual Inference

This chapter goes through the theory of causal inference. Causality is an important concept in this thesis because it will lead us into the understanding that data can easily be misinterpreted. The chapter is based on chapters 2 and 3 of *"Causal inference in statistics: A Primer"* by Judea Pearl (12). Formulas and equations are directly from the text but the contents are expanded and reworded by me. We begin by introducing ideas in linear modeling. We then move on to structural causal modeling and directed acyclic graphs. Further on we look into d-separation. Lastly, we discuss intervention methods, such as fixing variables, the 'do' method, the backdoor criterion, and the front-door criterion.

2.1 Linear Models

Before we go into causal inference we must discuss models. There are many types of models, in this section, we will go into regression models, structural causal models, and graphical models like directed acyclic graphs. A useful statistical model is the linear regression model. It is a simple yet powerful model that can create a line to show relations and predict values. A basic linear model can be in the form $y = mx + b$. In this y is the response, m is the slope, x is the predictor variable, and b is the intercept. The least-squared regression tries to minimize the residual sum of squares.

Definition 2.1.1 (Residual Sum of Squares)**.**

$$RSS = \sum (y_i - \hat{y})^2$$

Here y_i is the y value at i^{th} data point and $\hat{y}$ is the value that the regression predicted.

The residual sum of squares shows the error between the real data points and the regression line (10). Looking at **Figure 2.1** you can get a general idea of the graphical meaning of residual sum of squares. Here RSS is the sum of vertical distance from the blue points to the orange regression line squared. This distance is the error since it is the difference of our prediction and the actual data. These error values are squared before summing together to avoid them canceling out. So from the RSS we can get a general idea of the error. This issues with RSS are that the magnitude relies on the sample size and there is not a standardized unit for distance. Overall, this can still be useful for creating models. Minimizing RSS is used in the least-squares criterion. BY minimizing RSS you reduce the distance of the points to the regression line and end up with the line of best fit.

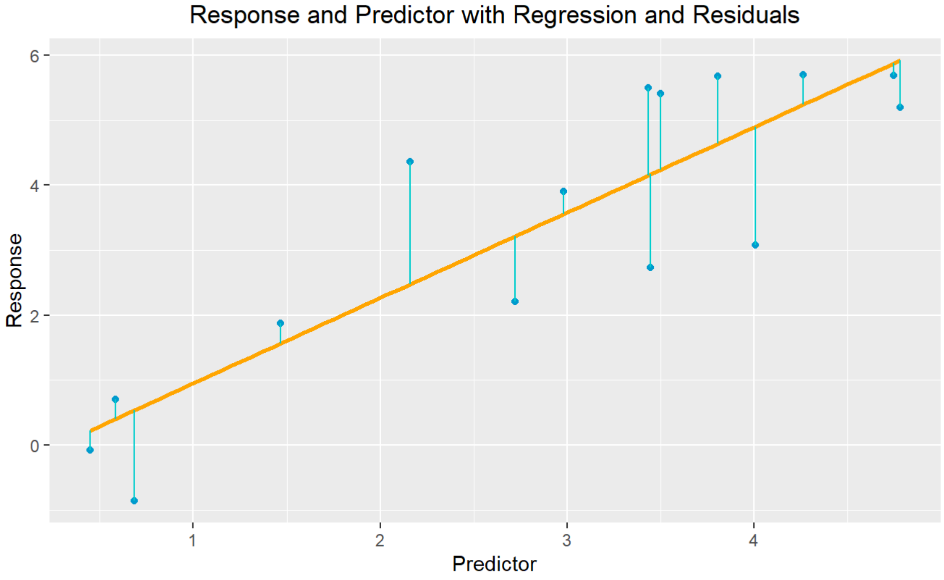

Figure 2.1 Graph of RSS: Orange diagonal line is Regression, blue vertical lines are residuals, blue points are data

While this value is helpful it can be used to calculate a much more useful value. First note that RSS or SS_{Res} represents error meaning it is the variation of the data that is not explained by the model. The opposite of this is the regression sum of squares, SS_{Reg}. This value is the variation explained by the model. The equation for it is $RSS = \sum(\hat{y} - \bar{y})^2$ where $\bar{y}$ is the mean. Together we get $SS_{tot} = SS_{Reg} + SS_{Res} = \sum(y_i - \bar{y})^2$ this value represents the total variation i.e. how far the data is from the mean. From these values you get one of the most important concepts in linear modeling, the coefficient of determination or R^2. Where $R^2 = \frac{SS_{Reg}}{SS_{tot}}$. R^2 is a measure that tells you how well your model fits the observed data. Now that we explained R^2 we can continue with other definitions.

2.2 Causality

Structural causal models, or SCMs, are models that attempt to describe the causal influences in a system. Here by system, we mean a group of interacting variables which may have causal relations. SCMs separate variables into two categories. Exogenous variables, U, are external to the system and endogenous variables, V, are internal. Endogenous variables are descendants of exogenous variables. A descendant of a variable is one that is further down the path of that variable in the direction of the arrows. Since exogenous variables have no ancestors they are root nodes.

Graphical causal models are graphical representations of SCMs. The nodes and edges are representations of variables and functions. Directed acyclic graphs (DAGs) can show causation. A child of a variable is directly caused by that variable. A descendant is potentially caused by the ancestor except for intransitive cases which will be explained later.

2.2.1 Chains

In this subsection we will go through the directed acyclic graphs of chains, forks, and colliders (12). First, chains can generally be shown as three (or more) nodes connected to each other by two edges that follow the same direction. A chain is shown in **Figure 2.2**.

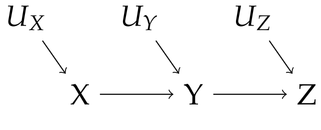

Figure 2.2 Chain Diagram With Exogenous Variables

Below are some rules for likely dependencies in chains (**Figure 2.2**) between variables X, Y, Z.

- Z and Y, are likely dependent for z, y values
  $P(Z = z|Y = y) \neq P(Z = z)$
- Y and X, are likely dependent for y, x values
  $P(Y = y|X = x) \neq P(Y = y)$
- Z and X, are likely dependent for z, x values
  $P(Z = z|X = x) \neq P(Z = z)$
- Z and X are conditionally independent on Y if for all x, y, z
  $P(Z = z|X = x, Y = y) = P(Z = z|Y = y)$

In a chain, all of the variables are likely dependent when not conditioned. The likely dependency between Z and X is because Z depends on Y which depends on X. The reason Z and X are only likely dependent and not certainly is because of exogenous variables. In **Figure 2.2** these are shown as U_X, U_Y, U_Z. An example of an exogenous variable is something that cannot be measured.

Definition 2.2.1 (The Case of Intransitive Dependence)**.** $V = \{X, Y, X\}$, $U = \{U_X, U_Y, U_Z\}$, $F = \{f_x, f_Y, f_Z\}$.
$f_X : X = U_X$

$$f_Y : Y = \begin{cases} a \textit{ if } X = 1 \textit{ and } U_Y = 1 \\ b \textit{ if } X = 2 \textit{ and } U_Y = 1 \\ c \textit{ if } U_Z = 2 \end{cases}$$

$$f_Z : Z = \begin{cases} i \textit{ if } Y = c \textit{ or } U_Z = 1 \\ j \textit{ if } U_Z = 2 \end{cases}$$

In the intransitive case above, X does not affect Z regardless of the exogenous variables. Only the values of a, b for Y depend on X and Y only affects Z if $Y = c$. So X and Z are independent since any changes to X do not change Z.

Furthermore, if we look at the cases where $Y = a$ is held constant. Then we can have X and Z as independent since changing X does not change Y and thus does not affect Z. This means that X is independent of Z conditional on Y.

2.2.2 Forks

Next, we have forks. These are graphically shown as three nodes connected by two edges, however, here we have two directions. The figure (2.3) below shows a fork.

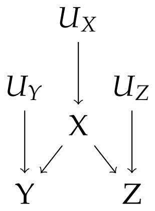

Figure 2.3 Fork Diagram With Exogenous Variables

The likely dependencies for **Figure 2.3** are shown below.

- X and Y, are likely dependent for x, y values
  $P(X = x|Y = y) \neq P(X = x)$
- X and Z, are likely dependent for x, z values
  $P(X = x|Z = z) \neq P(X = x)$
- Z and Y, are likely dependent for z, y values
  $P(Z = z|Y = y) \neq P(Z = z)$
- Y and Z are conditionally independent on X for all x, y, z
  $P(Y = y|Z = z, X = x) = P(Y = y|X = x)$

Theorem 2.1 (Conditional Independence of Forks). *If X is a variable that causes both Y and Z and the only path goes through X, then Y and Z are conditionally independent on X.*

This causes an issue in causality. If X is unobserved then we may make false assumptions that Z and Y are dependent on each other in the way of causality. Given X causes Y and Z it is clear that the changes in Y and Z are not from causation. In real data, however, you often don't get all the causes of each variable, and thus you may end up not realizing that there exists an X variable that creates a path between Z and Y. So without the knowledge of X, you cannot find the conditional independence, and the variables may look dependent on each other.

2.2.3 Colliders

This subsection is about colliders. Colliders look similar to forks but backward. They can have three nodes and two edges that connect in the opposite way as forks. The figure below shows a general collider.

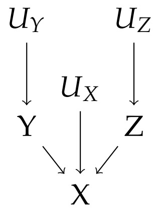

Figure 2.4 Collider Diagram With Exogenous Variables

The likely dependencies are quite different from the previous. They are shown below.

- Z and X, are likely dependent for z, x values
  $P(Z = z|X = x) \neq P(Z = z)$
- Y and X, are likely dependent for y, x values
  $P(Y = y|X = x) \neq P(Y = y)$
- Z and Y, are likely independent for z, y values
  $P(Z = z|Y = y) = P(Z = z)$
- Z and Y are likely conditionally dependent on X for x, y, z
  $P(Z = z|Y = y, X = x) \neq P(Z = z|X = x)$

The last item in this can create confusion. It is clear that Z and Y are independent in this system since one is not a descendant of the other and they are not the descendant of a shared ancestor. The confusion comes from the fact that they can be made dependent on each other.

A simple way of looking at this is a situation in which $X = Y + Z$. If we know the value of X it tells us nothing about the values of Y and Z, but if we are given the value of Y beforehand then the information of X tells us what Z is. So we know $Z = X - Y$ which makes Z dependent on Y conditional on X.

Theorem 2.2 (Conditional Independence in Colliders)**.** *If a variable Z is the collision node between two variables X and Y, and there is only one path between X and Y, then X and Y are unconditionally independent but are dependent conditional on Z and any descendants of Z.*

This theorem is the result of the ideas before it and shows why colliders are an interesting topic for dependence.

2.2.4 d-separation

Real-world casualty is usually much more complicated than these three node systems. In this subsection, we will take a look at d-separation and d-connection which helps with the understanding of independence.

Definition 2.2.2 (d-separation and d-connection)**.** *d-separated variables, X and Y, are nodes with either no paths or only blocked paths. If the nodes have any paths they are d-connected.*

Paths can be blocked by nodes such as colliders, chains, and forks. d-separation and d-connection can tell us about the dependencies of variables.

Theorem 2.3 (d-separation)**.** *A path is blocked by a set of nodes, Z, if and only if either of the below conditions are met*

1. *The path contains a chain or fork where the middle node is in the set Z (the middle node is conditioned on)*

2. *The path contains a collider where the collision node and its descendants are not in set Z.*

The importance of this is to tell dependence and independence. Variables that are d-separated are definitely independent and d-connected are most likely dependent. Furthermore, d-separation can be conditional; if every path between two nodes X and Y are blocked by Z, then X and Y are d-separated and independent conditional on Z.

2.3 Interventions

This section is based on Chapter 3 of *"Causal inference in statistics: A Primer"* by Judea Pearl (12). Here we will look into interventions that can help us understand causality. The real world has aspects that are impossible to randomize for experiments. Even if it is possible to control a variable it may be too expensive or difficult to be reasonable. Instead of controlling variables in circumstances like these, it is more beneficial to just record data than to create an experiment to analyze for causation. The issue this brings is the difficulty in determining if a relation is causal or only correlative.

Experimentation is usually the most reliable way of showing causation. With experimentation, we have the ability to compare control groups with one or more treatment groups made up of randomly selected individuals. In a model based on pre-collected data, we are not able to control any factors, so intervention on a variable is done by fixing its value and considering the corresponding data of the remaining variables. Fixing the value of a variable works to basically remove it from the system.

Let's look at an example system: there appears to be a correlation between trending searches of snickerdoodle cookies and the movie franchise Home Alone. This is shown in **Figure** 2.5.

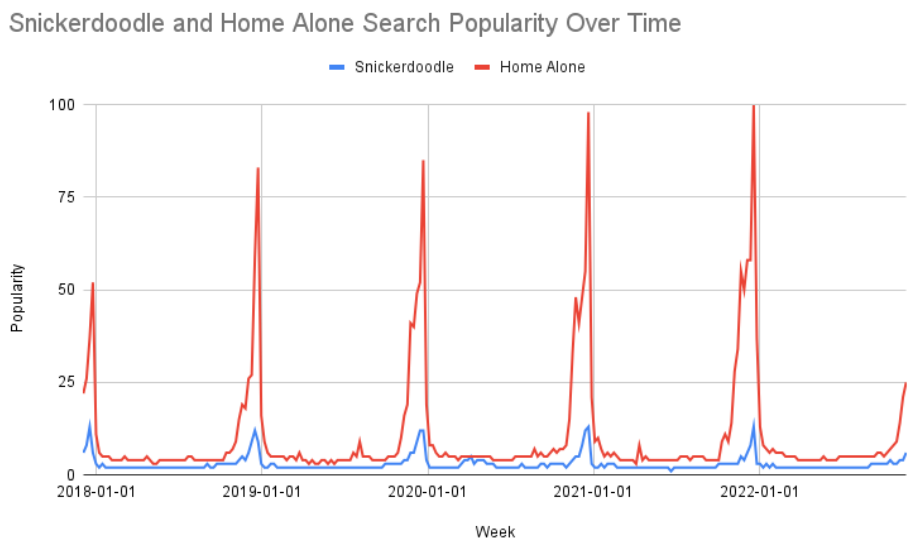

Figure 2.5 Google Search Trends Of Snickerdoodle and Home Alone Overtime

Snickerdoodles are not a core part of any of the Home Alone movies so the logic of causation is not clear. If you think a little harder you may come up with the third variable of Christmas. Many people watch Home Alone during the holidays and snickerdoodles are a sweet holiday treat. Of course, there may be some people who just need to eat snickerdoodles when they watch holiday-related films but the relationship with this is not obvious.

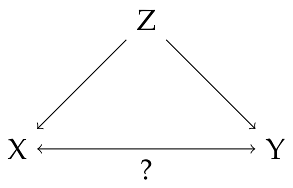

Figure 2.6 Unknown Causation Between X and Y

We can apply **Figure 2.6** to our snickerdoodle/Home Alone example. Here we have variable X as Home Alone, Y as snickerdoodles, and Z as Christmas. With our example, we can see an obvious correlation but without our prior knowledge, it would be difficult to determine the actual relation shown in **Figure** 2.6. When we fix the variable X (Home Alone) we get a completely different directed acyclic graph shown in the figure. The new DAG does not have an edge between X and Y. The reasoning for this is that if you fix variable X then any changes in Y do not affect X.

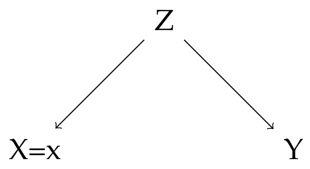

Figure 2.7 Modified DAG With Fixed X

In an experiment, we could fix the variable by banning Home Alone, but this is not reasonable in real life. Instead, we can just condition the data we already have. The next section will show us a method that we can use to do this.

2.3.1 Fixing Variables with 'do'

Continuing with chapter 3 of Pearl's book, *"Causal inference in statistics: A Primer"* (12), we now look into fixing variables using the method 'do'. For notation, we will use $do(X = x)$ to denote cases where we fix variable X with value x. So the probability that when we intervene on X we get outcome Y is $P(Y = y|do(X = x))$. This is different than our original probability because it represents the entire population if their X variable was fixed with value x. The probability $P(Y = y|X = x)$ on the other hand, only includes individuals that naturally have $X = x$.

As an example, if we have a population we can hypothetically intervene by uniformly applying treatment $do(X = 1)$ (such as giving a drug) or not applying treatment $do(X = 0)$ (giving a placebo). When we estimate the difference between these interventions we get the average causal effect (ACE), $P(Y = 1|do(X = 1)) - P(Y = 1|do(X + 0))$ (difference between outcomes i.e. recovery). If X and Y take on more values we call this the general causal effect.

Let's say we have the relation shown below in **Figure 2.9**.

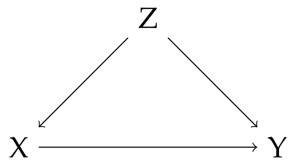

Figure 2.8 DAG of Mediator X

Here we have the variable X is called a mediator variable. In this, Y responds to both X and Z while X also is influenced by Z. By intervening $P(Y = y|do(X = x))$ we can modify the relation to become a fork structure like that of **Figure** 2.7. This DAG is the manipulated model representing the intervention. The manipulated conditional probability $P_m(Y = y|X = x)$ is equal to the causal effect $P(Y = y|do(X = x))$.

Our original probability has two properties shared with this manipulated probability. One is that the marginal probability does not change with intervention. This means that $P(Z = z)$ is not changed by removing the arrow between Z and X. The other shared property is that the conditional probability $P(Y = y|Z = z, X = x)$ is unchanged. This means that manipulation of X does not change the way that Y responds to both variables X and Z. These properties can be shown by the equations below.

$$P_m(Z = z) = P(Z = z)$$
$$\text{and}$$
$$P_m(Y = y|Z = z, X = x) = P(Y = y|Z = z, X = x)$$

Since our manipulated model no longer has the causal connection between X and Z they are now d-separated. This means they are independent and thus $P_m(Z = z|X = x) = P_m(Z = z) = P(Z = z)$. Next, we have the adjustment formula shown below.

Definition 2.3.1 (Adjustment Formula)**.**

$$P(Y = y|do(X = x)) = \sum_z P(Y = y|X = x, Z = z)P(Z = z)$$

This equation calculates the association of X and Y for all z values of variable Z. This is called *adjusting/controlling for* Z. It now is possible to solve for $P(Y = y|do(X = x))$ using the data since the right-hand side can be calculated with only conditional probabilities! In a controlled experiment this adjusting is unnecessary however in many cases it is unreasonable to do experimentation. Experimenters may use adjustment for minimizing variations in samples but this is not needed.

To show adjustment we will use a generic example. Let $X = 1$ be treatment, $Z = 1$ be subgroup 1, and $Y = 1$ be success. Then we get

$$P(Y = 1|do(X = 1)) = P(Y = 1|X = 1, Z = 1)P(Z = 1) + P(Y = 1|X = 1, Z = 0)P(Z = 0)$$

and

$$P(Y = 1|do(X = 0)) = P(Y = 1|X = 0, Z = 1)P(Z = 1) + P(Y = 1|X = 0, Z = 0)P(Z = 0)$$

Note that the above equations look quite similar to the expected value. With these, the effect of the treatment is as follows:

$$ACE = P(Y = 1|do(X = 1)) - P(Y = 1|do(X = 0))$$

In the next chapter, we will use this to show Simpson's paradox.

Theorem 2.4 (Causal Effect Rule)**.** *Given the variable parents of X called PA, the causal effect of X on Y is shown below.*

$$P(Y = y|do(X = x)) = \sum_z P(Y = y|X = x, PA = z)P(PA = z)$$

where z is the set of all combinations of values of PA

From this we can get a different equation.

$$P(Y = y|do(X = x)) = \sum_z \frac{P(X = x, Y = y, PA = z)}{P(X = x|PA = z)}$$

For the next part, we first need to know the rule of product decomposition.

Theorem 2.5. *For any acyclic graph, the joint distribution of variables can be found by the product of the conditional probabilities of children given parents $P(child|parents)$. This rule is given by the equation below.*

$$P(x_1, x_2, ..., x_n) = \prod_i P(x_i|pa_i)$$

X_i is the child variable and pa_i is the parent.

An example of this from a chain of three variables is below.

$$P(X = x, Y = y, Z = z) = P(X = x)P(Y = y|X = x)P(Z = z|Y = y)$$

To break this down, the first part $P(X = x)$ is the first node with no parents. Then $P(Y = y|X = x)$ is the child Y given the parent X. Lastly $P(Z = z|Y = y)$ is the child Z given the parent Y.

Now that we have this we can see that before the intervention of our original DAG we get

$$P(x, y, z) = P(z)P(x|z)P(y|x, z)$$

and after intervention

$$P(z, y|do(x)) = P_m(z)P_m(y|x, z) = P(z)P(y|x, z)$$

The reason that we don't have $P(x|z)$ after the intervention is because we fix $X = x$. If we want y given multiple values of z we use

$$P(y|do(x)) = \sum_z P(z)P(y|x, z)$$

Further, we can get the truncated product formula (g-formula)

$$P(x_1, x_2, ..., x_n|do(x)) = \prod_i P(x_i|pa_i) \text{ for all } i \text{ where } X_i \notin X$$

Finally, the relationship between pre-intervention and post-intervention is

$$P(z, y|do(x)) = \frac{P(x, y, z)}{P(x|z)}$$

This is really important because it means that with non-experimental data we can find the effect of intervention $do(x)$ by using conditional probability $P(x|z)$ (i.e. the effect of the treatment on the recovery).

2.3.2 Backdoor Criterion

This subsection will go over the backdoor criterion. Let's start with a definition.

Definition 2.3.2 (The Backdoor Criterion)**.** *Given a DAG with (X, Y) as an ordered pair of variables, then a set of variables Z satisfies the backdoor criterion if Z blocks all (if any) paths from Y to X and no node in Z is a descendant of X.*

Figure 2.9 shows an example where Z satisfies the backdoor criterion.

Theorem 2.6. *If the backdoor criterion is satisfied by the set of variables Z the causal effect of X on Y is*

$$P(y|do(x)) = \sum_{z} P(y|x,z)P(z)$$

The logic here is that we block erroneous paths between X and Y while leaving the direct pathways from X to Y. To find the causal effect of X on Y we need the set of backdoor variables Z to be conditioned so that no variable in Z has an arrow into X. Meaning Z cannot be an ancestor of X. This is to ensure X and Y are independent and not confounded. We must be careful to not condition descendants of X so as to not affect Y by blocking the path between X and Y. For this requirement, we cannot condition on colliders that may unblock a path between X and Y.

A generic example is shown in the DAG below.

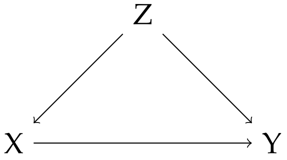

Figure 2.9 DAG of Mediator X

In **Figure 2.9** we want to determine the causal effect of treatment X on success Y with a backdoor Z relative to (X, Y), where Z is an ancestor to variable W which also affects the success Y. We can see that Z is a backdoor that affects both X and W directly and is an ancestor to Y. In actual data, we may not have the variable Z (Christmas from **Figure** 2.5). but since we are given W that can also work as a backdoor since it also satisfies the backdoor criterion. Now we can use the adjustment formula to come to a conclusion.

$$P(Y = y|do(X = x)) = \sum_{w} P(Y = y|X = x, W = w)P(W = w)$$

If we have a case with no backdoor variables then we simply have the empty set satisfying the backdoor criterion and thus our result is $P(y|do(x)) = P(y|x)$. In the case where we want a specific W value, we would need to change this a bit. Let's say we have the example below.

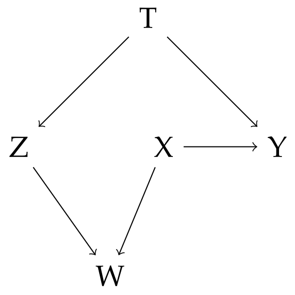

Figure 2.10 Backdoor Example

Here we would have to condition on T to remove the path ($Z \longleftarrow T \longrightarrow Y$). The result would be

$$P(y|do(x), w) = \sum_{T} P(y|x, w, t)P(t|x, w)$$

Using this we can do a method called *moderation or effect modification*. Here we can compare different values of W. It would be the simple comparison of $P(y|do(x), w)$ and $P(y|do(x), w')$.

2.3.3 Front-Door Criterion

In cases with unobserved confounders, you may not be able to use the backdoor criterion. Unobserved variables cannot block erroneous paths and thus are not able to be used for this. With hidden confounders (unobserved), we may be unable to determine the causality between X and Y since this confounder could potentially be the main reason for causality. To fix this we can use an intermediate variable. **Figures** 2.11 and 2.12 show a generic case of this.

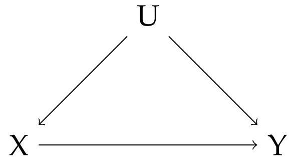

Figure 2.11 DAG where U is unobserved

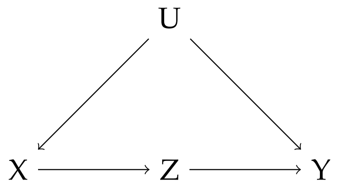

Figure 2.12 Front-Door Criterion: Modified Figure 2.11

Figure 2.12 shows that there is no backdoor from X to Z because U is unobserved, thus the effect of X on Z is able to be identified. We can also identify the effect of Z on Y by conditioning on X. Now from the adjustment formula, we have the two equations below.

$$P(Z = z|do(X = x)) = P(Z = z|X = x)$$

and

$$P(Y = y|do(Z = z)) = \sum_x P(Y = y|Z = z, X = x)P(X = x)$$

Since we want to know how X affects Y we can combine the effects above. The probability we want to find is $P(y|do(x))$ so we must start with setting $X = x$. Here we get $P(z|do(x)) = P(z|x)$ since there is no backdoor path. From this, we need to look at the arrow between Z and Y. To get this we must condition on X. We end up with

$$P(y|do(z)) = \sum_x P(y|z, x)P(x)$$

Now we join the effects together, so we get

$$P(y|do(x)) = \sum_z P(y|do(z))P(z|do(X))$$

Lastly, we plug in the equations using the notation x' for the summation. We finally get

$$P(y|do(x)) = \sum_z \sum_{x'} P(y|z, x')P(x')P(z|x)$$

This result is called the *front-door formula* and is extremely useful since we can use it to find causal effect when the backdoor method is not possible. Similarly to the backdoor formula, there are criteria that must be met.

Definition 2.3.3 (Front-Door Criterion)**.** *The set of variables Z satisfies the front-door criterion relative to (X, Y) an ordered pair of variables if the following are true.*

1. *All paths from X to Y are intercepted by a variable in Z*
2. *X has no backdoor paths to Z*
3. *All (if any) backdoor paths from Z to Y are blocked by X*

Now that we defined the front-door criterion we can look at one more important theorem.

Theorem 2.7 (Front-Door Adjustment)**.** *If $P(x, z) > 0$ and Z satisfies the front-door criterion in respect to (X, Y) then the causal effect of X on Y is identifiable by the formula below*

$$P(y|dx(x)) = \sum_{z} P(z|x) \sum_{x'} P(z|x', z)P(x')$$

With that theorem we can now finish this chapter with the tools necessary to manipulate data to show causal relations. If you would like to go further into causality I suggest looking into *do-calculus* which is beyond the scope of this thesis.

Chapter 3

Simpson's Paradox

This chapter discusses different definitions of Simpson's paradox. We begin with a definition by rates. There we look at the differences between an aggregate rate of an entire population and the rates of individual subgroups. Next, we define Simpson's paradox through probability. Lastly, we look at graphical examples.

3.1 Population and Sub-population Rates

In this chapter (**Chapter 3**), we will look at definitions of Simpson's paradox from the perspectives of different areas of mathematics. We will start with an example case involving adoption rates.

Consider two animal shelters trying to get people to adopt their animals. Below are their adoption rates for cats, dogs, and in total.

	Shelter 1	**Shelter 2**
Dog Adopted	9	31
Dog Not Adopted	4	16
Cat Adopted	94	70
Cat Not Adopted	76	60
Dog Adopted Percent	69.2%	66.0%
Cat Adopted Percent	55.3%	53.8%
Total Adopted Percent	56.3%	57.1%

Table 3.1 Adoption Rates for Two Shelters

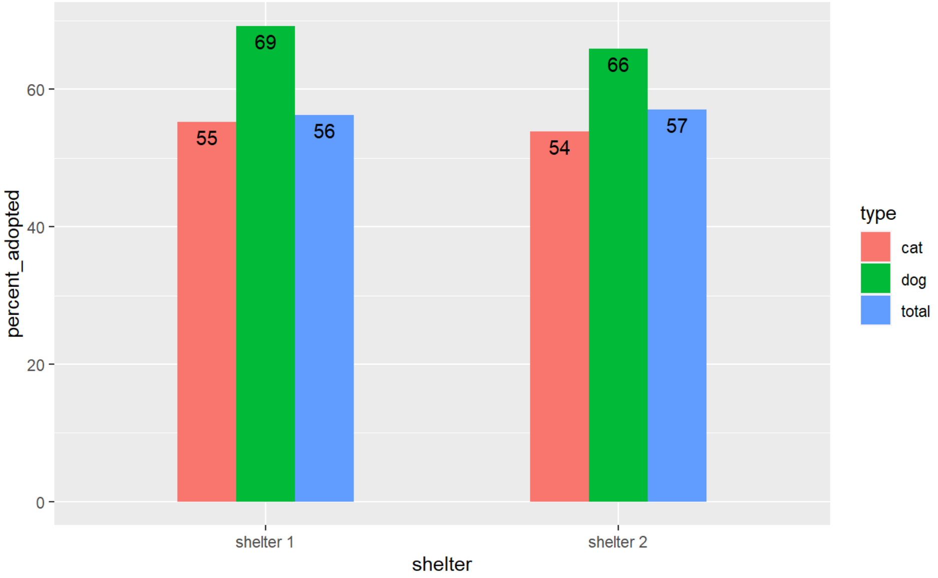

Figure 3.1 Bar Chart of Adoption Rates for Two Shelters

For total adoptions shelter 2 has a higher percentage but separately Shelter 1 has more dog adoptions and cat adoptions. Basic arithmetic says that $0.6923 = \frac{9}{13} > \frac{31}{47} = 0.6596$ and $0.5529 = \frac{94}{170} > \frac{70}{130} = 0.5385$, but we also have $0.5628 = \frac{9+94}{13+170} < \frac{31+70}{47+130} = 0.5706$. As you can see the left side of the equation is greater for the first two equations which are the subgroups but it is less for the last equation which is the aggregate.

This case is not very extreme and the implications are minimal but in the real world there are cases that are more dramatic and have significant implications.

In more mathematical terms, consider $[A, B]$ mutually exclusive and jointly exhaustive populations with rates $[\alpha, \beta]$ (rates for two populations) when partitioned we get rates $[A_1, A_2, B_1, B_2]$ (rates for two populations subdivided) so the overall rates are $\alpha = A_1 + A_2$ and $\beta = B_1 + B_2$.(2)
We define

$$C_1 \equiv A_1 \geq B_1$$

$$C_2 \equiv A_2 \geq B_2$$

$$C_3 \equiv \beta \geq \alpha$$

$C \equiv (C_1 \& C_2 \& C_3)$

Let $\theta = (A_1 - B_1) + (A_2 - B_2) + (\beta - \alpha)$, then Simpson's paradox occurs when

i) $C \equiv (C_1 \& C_2 \& C_3)$ and

ii) $C_4 \equiv \theta > 0$

Going through the conditions we can see that condition i) implies Simpson's directly except for the case $(A_1 = B_1 \ \& A_2 = B_2 \ \& \beta = \alpha)$. To fix this we need condition ii) to remove this case, so we have $\theta = (A_1 - B_1) + (A_2 - B_2) + (\beta - \alpha) > 0$. This means the values can't all be equal, thus combined the conditions suffice in the i) → ii) direction.

Next we must show that condition ii) must also include condition i). Conversely, $A_1 > B_1, B_2 > A_2, \beta > \alpha$ satisfies ii) but is not Simpson's paradox. So we need the i) condition. Now since A_2 must be greater than or equal to B_2 we remove this case. Thus, the conditions suffice in the ii) → i) direction.

Now that we showed that both conditions are necessary and sufficient to show Simpson's paradox we can go through a few theorems.

Theorem 3.1. *Simpson's Paradox requires* $A_1 \neq A_2(2)$

Proof Using the notation above we can find a contradiction in the case where $A_1 = A_2$. Let $a = \frac{\text{members of A in partition 1}}{\text{total members of A}}$. Then we have the rate

$$\alpha = aA_1 + (1 - a)A_2$$

Since $A_1 = A_2$ we get

$$aA_1 + (1 - a)A_2 = aA_1 + (1 - a)A_1 = aA_1 + A_1 - aA_1 = A_1$$

So $\alpha = A_1$, but by definition $\alpha = A_1 + A_2$ so for $\alpha = A_1$ we need $A_2 = 0$. That implies that $A_1 = 0$ as well. So now we have $\alpha = A_1 = A_2 = 0$. From our original condition, we need $A_1 \geq B_1$, $A_2 \geq B_2$, and $\beta \geq \alpha$. Since the values are rates they must be non-negative. Together we have $0 = A_1 \geq B_1 \geq 0$ so B_1 must be 0. Similarly, B_2 must be 0. So now $B_1 = B_2 = 0$, thus $\beta = B_1 + B_2 = 0$. Finally this violates condition ii) since $\theta = (0 - 0) + (0 - 0) + (0 - 0) \not> 0$■

Theorem 3.2. *Simpson's Paradox requires* $B_1 \neq B_2(2)$

Proof If $B_1 = B_2$ then $\beta - B_1 - B_2 = 0$. Since $\theta = (A_1 - B_1) + (A_2 - B_2) + (\beta - \alpha)$ we get $\theta = A_1 + A_2 - \alpha - 0$. By definition $\alpha = A_1 + A_2$ so $\theta = 0$. This contradicts condition ii) since it requires $\theta > 0$■

Next, we will look at another way we can define Simpson's Paradox. This definition looks at successes and failures of two sub-populations.

Consider a population D with sub-populations D_1 and D_2. Let sub-population D_1 have A_i trials with a_i successes and D_2 have B_i trials with b_i successes. Then Simpson's paradox occurs if

$$\frac{a_i}{A_i} \geq \frac{b_i}{B_i} \text{ for all } i = 1, 2, ..., n \text{ and } \frac{\sum a_i}{\sum A_i} \leq \frac{\sum b_i}{\sum B_i}(16)$$

An example of this is shown at the beginning of the section (3.1). This is one of the simplest ways of showing Simpson's paradox and my favorite of the non-graphical explanations .

3.2 Definition by Probability

In this section, we will define Simpson's paradox through probability. Let $T = treatment, \neg T = no\ treatment, S = success, \neg S = failure, M = male$, and $\neg M = female$. In this definition, treatment is not in the sense of medicine but as a treatment in an experiment (no treatment is control). Further, success only means a desired outcome. As well, male and female can be substituted by other sub-groupings. With this notation, we have Simpson's paradox if

i) $P(S|T) \leq P(S|\neg T)$

ii) $P(S|T, M) > P(S|\neg T, M)$

iii) $P(S|T, \neg M) > P(S|\neg T, \neg M)(16)$

Condition i) means that overall success is more probable without treatment. Condition ii) means success for male participants is more probable with treatment. Condition iii) means success for female participants is more probable with treatment. The paradox comes from the fact that the treatment is less than or equally successful as no treatment in aggregate but for each sub-population the treatment has more successes. The example at the beginning of this chapter (3.1) can be seen from this perspective since the percent adopted can represent the probability of adoption. Success is adoption, treatment is shelter, and the sub-populations are cats and dogs.

3.3 Graphics of Simpson's Paradox

This last section will go through graphical representations of Simpson's paradox. The first data set we will look at is the iris dataset(15). This dataset is a classic dataset used in many data science and machine learning classes. The iris data frame has 150 observations (rows) of individual iris flowers with 5 variables (columns) recorded. These variables include the sepal length and width, petal length and width, and species. When looking at data we often like to try and see relationships and correlations. A great way to start with this is graphically. In **Figure** 3.2 we graphed sepal width and sepal length for the aggregate data.

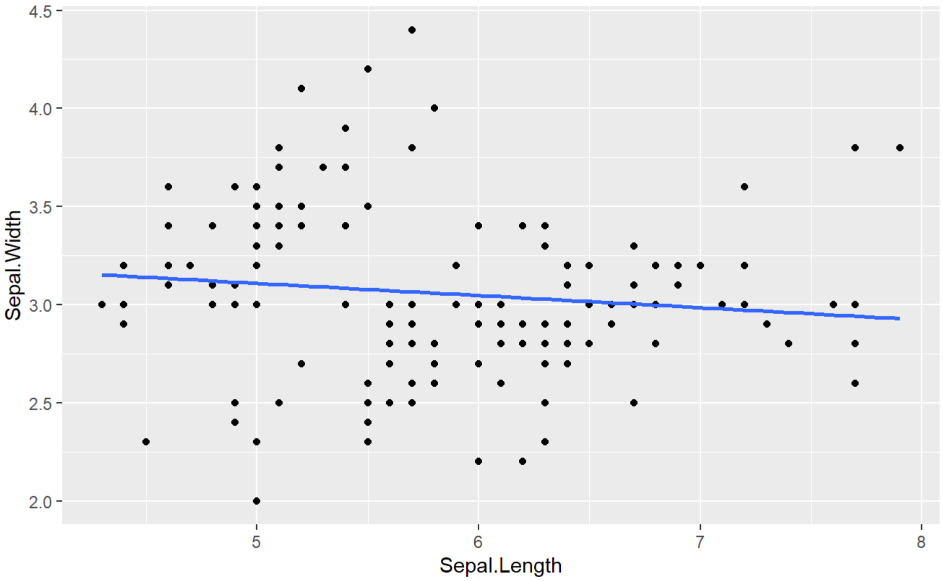

Figure 3.2 Graph of Aggregate Data

The linear regression line shown in blue has a slightly negative slope (-0.1176) but is close enough to 0 to say it is possibly not correlated. When we look at the data separated by species we get a different result. This is shown in **Figure** 3.3 below.

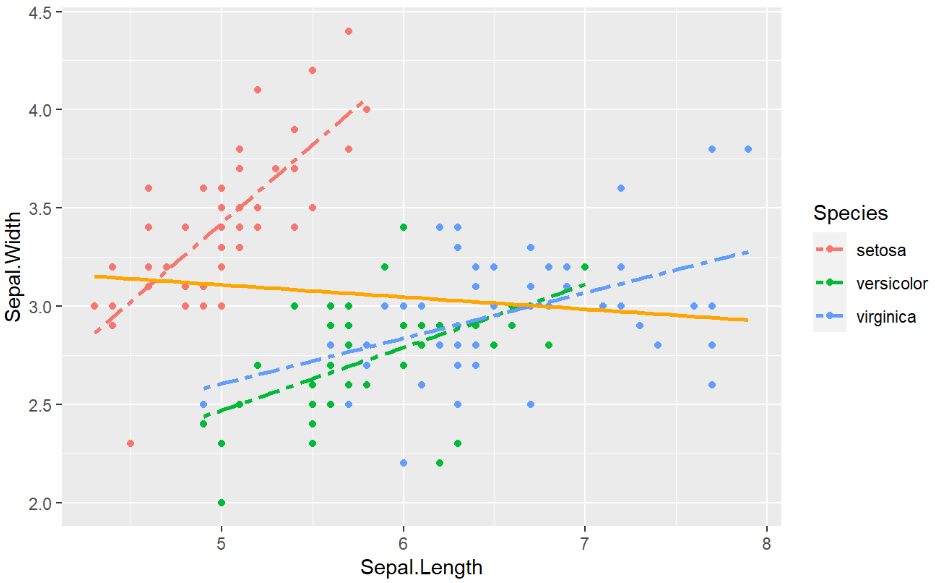

Figure 3.3 Graph of Subgroups

Here we have positive correlations for each species. This change in correlation is a great visualization of Simpson's paradox. The aggregate data has opposing results from the divided group data. Without the regression lines it is still possible to visually see the reversed trends in this graph. This is not always the case. Some data sets have much more overlaps in the data that may make it difficult to tell. The next chapter (4) will attempt to help with this issue.

Now we will go through randomly generated data in order to see how we can create Simpson's paradox. Before we begin, you can find the code to follow along in both the appendix and the GitHub. Do note that running this code will give different results every time due to randomness. Starting with an example of non-Simpson's paradox data, we have the figure below.

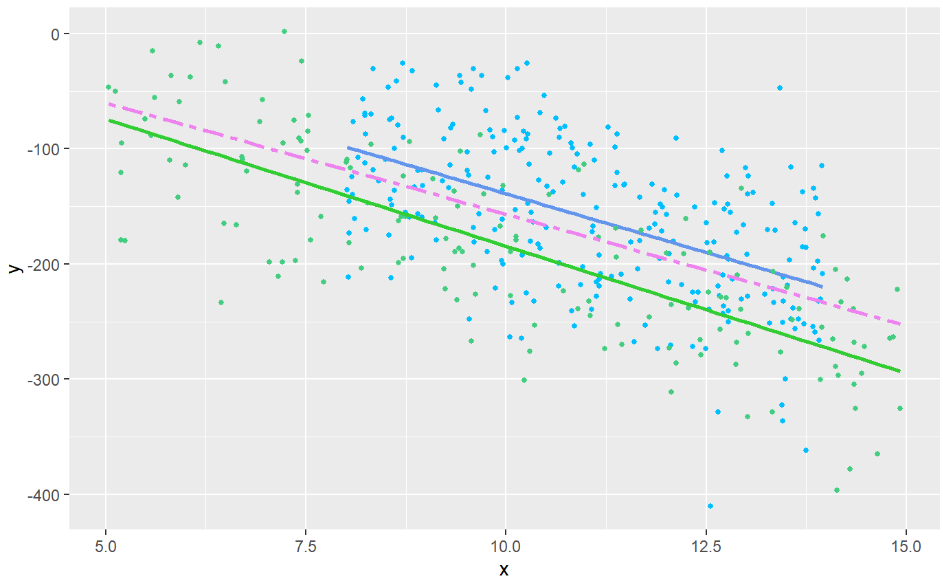

Figure 3.4 Generated Data Without Simpson's

This data was generated in a way that did not cause Simpson's paradox. This was done by keeping the clusters close enough and with the same general trend so that the aggregate regression would just become like an average. We generated another graph with Simpson's paradox shown somewhat but not extremely.

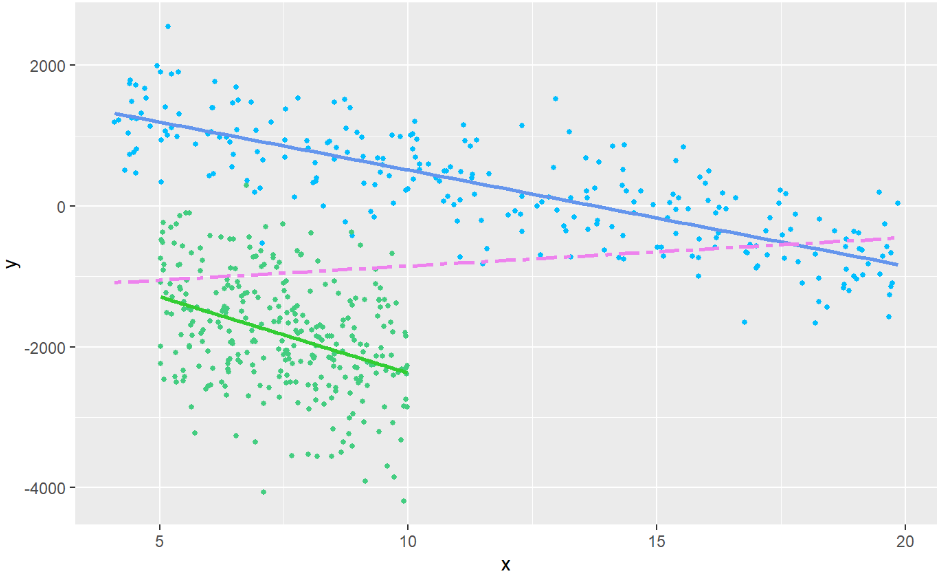

Figure 3.5 Generated Data with Moderate Simpson's

Here we have a subgroup that is more condensed on the x-axis and another that is spread out. The vertical distance between the regressions has spread further away and the result is a graph that shows Simpson's paradox somewhat. Our last graph is an obvious example.

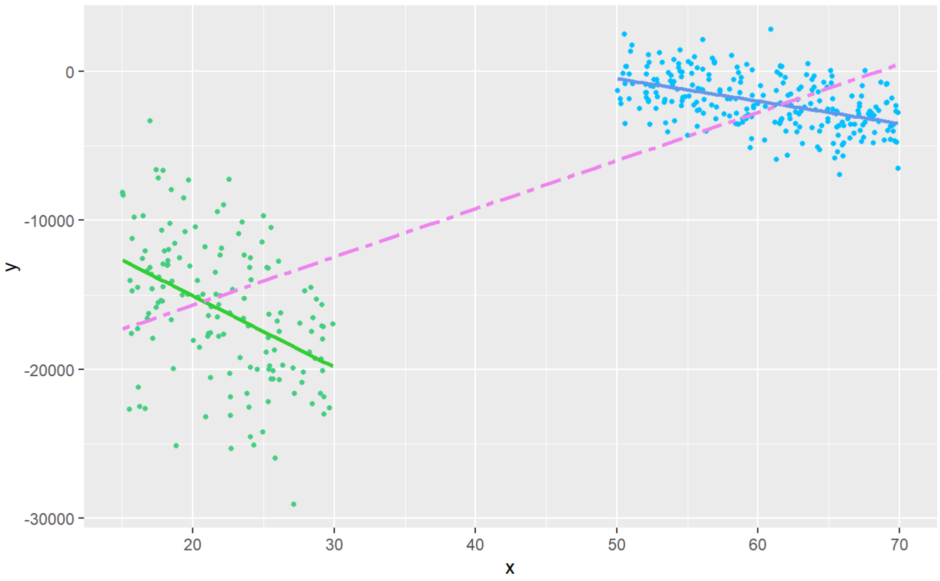

Figure 3.6 Generated Data with Obvious Simpson's

This last example of generated data has two subgroups that are densely clustered relative to the scale. These clusters are far apart to the point that Simpson's paradox occurs. The regression line of the aggregate becomes Simpson's paradox since the subgroups are so distant that the regression is more accurate with a reversed slope than one in between the two subgroups. This last example makes it clear what graphically causes Simpson's paradox. In this chapter, we learned how to define Simpson's paradox by rates, with probability, and visually. The next chapter (4) will look into determining Simpson's with code.

Chapter 4

Coding Examples

4.1 Population and Sub-population Correlation

This chapter (4) will go through the creation of code to detect Simpson's paradox. When you have a large set of data it may seem impossible to determine if Simpson's paradox takes place. This code will attempt to automate the process of checking for Simpson's paradox. The code is in the appendix and on GitHub if you would like to follow along.

4.2 Function 1

In this first function, the detection method is through analysis of correlation (14). For this, you will need to have a data set with a predictor variable, $xVar$, and a response variable, $yVar$. The process starts by calculating the correlation of the aggregated data. That means the correlation using the entire columns. From here you go through each column that are not the predictor or response columns. Then you must check the values in each column to see if they are categorical. In general, that means they are non-numeric values. Next, you split the data into subsets of the categories. Then you calculate the correlation of the predictor and response values of just the data of each subgroup. Lastly, you need to compare the aggregated correlation to the subgroups correlations. Below is pseudo-code that describes this process.

Algorithm 1 Detection by Correlation of Categorical Subgroups

```
Input: data, xVar, yVar
/* data is a dataset and xVar and yVar are string values of the column names for the
predictor and response variables*/

Algorithm:
xData = data [xVar] /* Column data of x values*/
yData = data [yVar] /* Column data of y values*/
aggCor = Correlation(xData,yData)/* Pearson's correlation of x and y values*/
print("Aggregated Correlation", aggCor) /* Prints the aggregated correlation*/
cols = data[data columns not xVar, yVar] /* data frame without x and y variables*/

/*Initializing lists*/
corCoef = empty list
allSubgroups = empty list

/*Find the correlation of subgroups of categorical variables*/
for each column in cols do: /*Go through each column*/
   if class of column is factordo: /*factor class is categorical data*/
     subgroups = unique values of column
     for each sub in subgroups do: /*Go through each subgroup*/
       allSubgroups = allSubgroups + sub /*Add sub to the list of subgroups*/
       subdf = filter data /*Keep only values in subgroup*/
       subX = subdf [xVar] /*x values only in subgroup*/
       subY = subdf [yVar] /*y values only in subgroup*/
       subCor = cor(subX,subY) /*correlation in subgroup*/
       corCoef = corCoef + subCor /*Add subCor to the list*/
       end
     end
   end

/*Test subgroup correlations for reversal*/
ind = 1 /*index to keep track of location in list*/
sp = FALSE/*Default value Simpsonś paradox is false*/

if length of corCoef > 0 do: /*Test for reversal of items in the list*/
   if aggCor > 0 do: /*Checking the sign of aggregate correlation*/
     for each cc in corCoef do: /*Goes through all subgroup correlations*/
       if cc < 0 do: /*Checks for reversal*/
         print("Simpson's Paradox", corCoef[ind], allSub[ind])/*Results*/
         sp = TRUE/*Saves SP result*/
       end
     end
   end
   /*Test for negative aggCor*/
```

```
  if aggCor < 0 do: /*Checking the sign of aggregate correlation*/
    for each cc in corCoef do: /*Goes through all subgroup correlations*/
      if cc > 0 do: /*Checks for reversal*/
        print("Simpson’s Paradox", corCoef[ind], allSub[ind])/*Results*/
        sp = TRUE/*Saves SP result*/
      end
    end
  end
if sp = FALSE do:
  print("Simpson’s not detected") end
```

Assumptions in this algorithm are that all categorical variables are in the type factor. This however is not the case. Numerical values can be used as a replacement for categorical values. For example, a car has a certain number of cylinders; these cylinders are numeric but they represent subgroups. In order to fix this we can create a new algorithm that determines if a variable is a hidden factor by testing if it is an integer with an arbitrary number of levels or subgroups.

4.3 Function 2

The code in this second function itself is similar however the input also includes a value for the arbitrary max number of subgroups. This max number will help determine if a column has subgroups or if the labels are most likely names or discrete numerical values. This issue is also in the common case where values are labeled with IDs. This edited code goes through the ways subgroups can be contained by variable type to make sure all cases of subgroups are tested. The pseudo-code is below.

Algorithm 2 Detection by Correlation of All Subgroups

```
Input: data, xVar, yVar, maxSubs
/* data is a dataset and xVar and yVar are string values of the column names for the
predictor and response variables maxSubs is an integer value*/

Algorithm:
xData = data [xVar] /* Column data of x values*/
yData = data [yVar] /* Column data of y values*/
aggCor = Correlation(xData,yData)/* Pearson's correlation of x and y values*/
print("Aggregated Correlation", aggCor) /* Prints the aggregated correlation*/
cols = data[data columns not xVar, yVar] /* data frame without x and y variables*/

cNamesAll = names of columns/*Keeps the names of each column*/
/*Initializing lists*/
corCoef = empty list
allSubgroups = empty list indName = 1 /*Keeps track of column name*/ corName
= empty list

/*Find the correlation of subgroups of categorical variables*/
for each column in cols do: /*Go through each column*/
    subgroups = unique values of column
  continue = FALSE/*Default FALSE becomes true if the current col is grouped*/
   if class of column is factordo:
    continue = TRUE/*Change continue to true to continue to correlation stage*/
  end
   if class of column is integer AND number of subgroups < maxSubsdo:
    continue = TRUE/*Change continue to true to continue to correlation stage*/
  end
   if class of column is character AND number of subgroups < maxSubsdo:
    continue = TRUE/*Change continue to true to continue to correlation stage*/
  end
  if continue = TRUE
    for each sub in subgroups do: /*Go through each subgroup*/
      allSubgroups = allSubgroups + sub /*Add sub to the list of subgroups*/
      subdf = filter data /*Keep only values in subgroup*/
      subX = subdf [xVar] /*x values only in subgroup*/
      subY = subdf [yVar] /*y values only in subgroup*/
      subCor = cor(subX,subY) /*correlation in subgroup*/
      corCoef = corCoef + subCor /*Add subCor to the list*/
      end
    end
  end

/*Test subgroup correlations for reversal*/
ind = 1 /*index to keep track of location in list*/
sp = FALSE/*Default value Simpsonś paradox is false*/
```

```
if length of corCoef > 0 do: /*Test for reversal of items in the list*/
  if aggCor > 0 do: /*Checking the sign of aggregate correlation*/
    for each cc in corCoef do: /*Goes through all subgroup correlations*/
      if cc < 0 do: /*Checks for reversal*/
        print("Simpson's Paradox", corCoef[ind], allSub[ind])/*Results*/
        sp = TRUE/*Saves SP result*/
      end
    end
  end
  /*Test for negative aggCor*/
  if aggCor < 0 do: /*Checking the sign of aggregate correlation*/
    for each cc in corCoef do: /*Goes through all subgroup correlations*/
      if cc > 0 do: /*Checks for reversal*/
        print("Simpson's Paradox", corCoef[ind], allSub[ind])/*Results*/
        sp = TRUE/*Saves SP result*/
      end
    end
  end
if sp = FALSE do:
   print("Simpson's not detected")
```

This second algorithm did a better job of detecting Simpson's paradox. It removed the issue of hidden categorical data. Overall, these algorithms are useful in detecting Simpson's paradox but still need human supervision. Of course, there will be edge cases that we have not considered which will not be found by these algorithms. Nonetheless, this is a good start. In the end, it must be the job of a data scientist or someone similar to check for themselves if the results make sense.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis we went through the ideas of causal influence and Simpson's paradox. We used basic probability and statistics along with graph theory to understand issues in causality. We used directed acyclic graphs to visualize structural causal models. We used probability to manipulate data structures to show causality in precollected data. We went through definitions of Simpson's paradox from the perspectives of differing mathematical subjects.

In Chapter one, we looked at one example of Simpson's paradox in the real world. We also went through the basic probability and statistics needed to understand the following chapters. In the second chapter, we discussed causal influence to understand problems and solutions to working with pre-collected data. We looked at linear models, structural causal models, and directed acyclic graphs. We defined d-separation to better understand dependence and independence. Then we discussed intervention methods, like fixing variables in the 'do' method, and the backdoor and the front-door criteria. In chapter three, we defined Simpson's paradox using different methods. These methods included definitions by rates, probability, and graphical exploration. Lastly, in chapter four, we created two algorithms that worked to help determine if a data set has Simpson's paradox. This code was limited, however, it can still be a useful tool for data scientists.

5.2 Future Work

Future work may include creating more code methods to determine if a dataset satisfies the conditions of Simpson's paradox. It may be graphically or using statistics or probability. For causality, we may want to look further into 'do' calculus from the end of Chapter 2. Further research may also be done about the implications of these results in the real world.

Appendix A

Source Code

```
---
title: "Senior Thesis"
author: "Emily Naitoh"
date: "2022-11-02"
output: html_document
---
# Code language R

library(ggplot2)
library(tidyverse)
library(ascentTraining)


n = 15
b0 = 0
b1 = 1.3
x.rss = runif(n,min=0,max=5)
e = rnorm(n, mean = 0, sd =1)
y.rss = b0 + b1*x.rss + e
d.rss = data.frame(cbind(x.rss,y.rss))
fit=lm(y.rss~x.rss,d.rss)
d.rss$p.rss = predict(fit)
d.rss$r.rss = residuals(fit)
dsave = d.rss
```

```
plt = ggplot(dsave, aes(x.rss,y.rss)) +
  geom_point(color="deepskyblue3")

plt +
  geom_smooth(se=FALSE,method='lm',color="Orange")+
  geom_segment(aes(x=x.rss,y=y.rss,xend = x.rss, yend = p.rss),
               data = dsave,color="cyan3")+
  xlab("Predictor")+ylab("Response")+
  ggtitle("Response and Predictor with Regression and Residuals")+
  theme(plot.title = element_text(hjust = 0.5))


# a is adopted
# n is not adopted
# t is total

#shelter 1
dog1a = 9
dog1n = 4
dog1t = dog1a+dog1n
cat1a = 94
cat1n = 76
cat1t = cat1a+cat1n
shelter1a = dog1a+cat1a
shelter1n = dog1n+cat1n
shelter1t = shelter1a+shelter1n

#shelter 2
dog2a =31
dog2n =16
dog2t =dog2a+dog2n
cat2a =70
cat2n =60
cat2t =cat2a+cat2n
shelter2a =dog2a+cat2a
shelter2n =dog2n+cat2n
shelter2t =shelter2a+shelter2n

# percent adopted
```

```
d1 = 100*(dog1a/dog1t)
c1 = 100*(cat1a/cat1t)
t1 = 100*(shelter1a/shelter1t)

d2 = 100*(dog2a/dog2t)
c2 = 100*(cat2a/cat2t)
t2 = 100*(shelter2a/shelter2t)

shelter = c(rep("shelter 1" , 3) , rep("shelter 2" , 3))
type = rep(c("dog" , "cat", 'total') , 2)
percent_adopted= c(d1,c1,t1,d2,c2,t2)
adoptdata = data.frame(shelter,type,percent_adopted)

# group bar chart
ggplot(adoptdata, aes(fill=type, y=percent_adopted, x=shelter)) +
  geom_col(width = 0.5, position = 'dodge')+
  geom_text(label=round(percent_adopted),vjust = 1.5,
            position = position_dodge(.5))


data(iris)
head(iris)
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width))+
  geom_point()+
  geom_smooth(method='lm',se=FALSE)
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,color=Species))+
  geom_point()+
  geom_smooth(method='lm',se=FALSE,linetype="twodash")+
  geom_smooth(aes(x=Sepal.Length,y=Sepal.Width),method='lm',
              se=FALSE,color="Orange")


## Not SP Random Data
# Subgroup A
nA = 250
beta0A = 30
beta1A = -18
XA = runif(nA,min=8,max=14)
epsilonA = rnorm(nA, mean = 0, sd =60)
```

```
YA = beta0A + beta1A*XA + epsilonA
dfA = data.frame(cbind(XA,YA))

# Subgroup B
nB = 150
beta0B = 5
beta1B = -20
XB = runif(nB,min=5,max=15)
epsilonB = rnorm(nB, mean = 2, sd =52)
YB = beta0B + beta1B*XB + epsilonB
dfB = data.frame(cbind(XB,YB))

# Together
XAB = append(XA,XB)
YAB = append(YA,YB)
dfAB = data.frame(cbind(XAB,YAB))

# Plot
ggplot(data=dfA,aes(x=XA,y=YA))+
  geom_point(color='deepskyblue',size=1)+
  geom_smooth(method='lm',se=FALSE,color="cornflowerblue")+
  geom_point(data=dfB,aes(x=XB,y=YB),color='seagreen3',size=1)+
  geom_smooth(data=dfB,aes(x=XB,y=YB),method='lm',se=FALSE,
              color='limegreen')+
  geom_smooth(data=dfAB,aes(x=XAB,y=YAB),method='lm',se=FALSE,
              color='violet',linetype="twodash")+
  xlab('x')+ylab('y')

## SP Random Data
# Subgroup A
nA = 250
beta0A = 2000
beta1A = -150
XA = runif(nA,min=4,max=20)
epsilonA = rnorm(nA, mean = 0, sd =500)
YA = beta0A + beta1A*XA + epsilonA
dfA = data.frame(cbind(XA,YA))

# Subgroup B
```

```
nB = 300
beta0B = -1
beta1B = -230
XB = runif(nB,min=5,max=10)
epsilonB = rnorm(nB, mean = -10, sd =750)
YB = beta0B + beta1B*XB + epsilonB
dfB = data.frame(cbind(XB,YB))

# Together
XAB = append(XA,XB)
YAB = append(YA,YB)
dfAB = data.frame(cbind(XAB,YAB))

# Plot
ggplot(data=dfA,aes(x=XA,y=YA))+
  geom_point(color='deepskyblue',size=1)+
  geom_smooth(method='lm',se=FALSE,color="cornflowerblue")+
  geom_point(data=dfB,aes(x=XB,y=YB),color='seagreen3',size=1)+
  geom_smooth(data=dfB,aes(x=XB,y=YB),method='lm',se=FALSE,
              color='limegreen')+
  geom_smooth(data=dfAB,aes(x=XAB,y=YAB),method='lm',se=FALSE,
              color='violet',linetype="twodash")+
  xlab('x')+ylab('y')

## Obvious SP Random Data
# Subgroup A
nA = 250
beta0A = 7000
beta1A = -150
XA = runif(nA,min=50,max=70)
epsilonA = rnorm(nA, mean = 100, sd =1500)
YA = beta0A + beta1A*XA + epsilonA
dfA = data.frame(cbind(XA,YA))

# Subgroup B
nB = 150
beta0B = -5000
beta1B = -500
XB = runif(nB,min=15,max=30)
```

```
epsilonB = rnorm(nB, mean = -50, sd =4300)
YB = beta0B + beta1B*XB + epsilonB
dfB = data.frame(cbind(XB,YB))

# Together
XAB = append(XA,XB)
YAB = append(YA,YB)
dfAB = data.frame(cbind(XAB,YAB))

# Plot
ggplot(data=dfA,aes(x=XA,y=YA))+
  geom_point(color='deepskyblue',size=1)+
  geom_smooth(method='lm',se=FALSE,color="cornflowerblue")+
  geom_point(data=dfB,aes(x=XB,y=YB),color='seagreen3',size=1)+
  geom_smooth(data=dfB,aes(x=XB,y=YB),method='lm',se=FALSE,
              color='limegreen')+
  geom_smooth(data=dfAB,aes(x=XAB,y=YAB),method='lm',se=FALSE,
              color='violet',linetype="twodash")+
  xlab('x')+ylab('y')


simpDetector = function(data,xVar,yVar){
  data = na.omit(data) # remove NA values

  ## Determine Aggregated Correlation
  a=data[[xVar]]
  b=data[[yVar]]
  aggCor = cor(a,b)
  print(paste0('Aggregated Correlation: ',aggCor))
  cs = data[names(data) %in% c(xVar,yVar)== FALSE] # data frame

  # Find correlation of subgroups with categorical variables
  corCoefs = list() # initialize list]
  allSub = list() # initialize list
  for (currentCol in cs){
    if (class(currentCol)=="factor"){
        subs = unique(currentCol)
        for (s in subs){
          allSub=c(allSub,s)
          subDf = data %>% filter(currentCol == s)
          subx=subDf[[xVar]]
          suby=subDf[[yVar]]
          subCor = cor(subx,suby)
          corCoefs = c(corCoefs, subCor)
        }
    }
  }

  # Test if reversal
  ind = 1 # index to find string
  sp = FALSE
  if (length(corCoefs>0)){
    if (aggCor > 0){
        for (cc in corCoefs){
          if(cc<0){
            print(paste0(paste0(paste0("Simpson\'s Paradox detected: ",
                  corCoefs[[ind]]),", "),allSub[[ind]]))
            sp=TRUE
          }
        ind = ind+1
        }
    }
    else if (aggCor < 0){
      for (cc in corCoefs){
        if(cc>0){
          print(paste0(paste0(paste0("Simpson\'s Paradox detected: ",
                corCoefs[[ind]]),", "),allSub[[ind]]))
          sp=TRUE
        }
        ind = ind+1
      }
    }
  }
  if (sp == FALSE){
    print("Simpson\'s Paradox is not detected.")
  }
}
```

```
# test
data(iris)
simpDetector(iris,'Sepal.Length','Sepal.Width')
simpDetector(auto_mpg,'mpg','acceleration')#false b/c numeric subs


simpDetector2 = function(data,xVar,yVar,maxSubs){
  data = na.omit(data) # remove NA values

  ## Determine Aggregated Correlation
  a=na.omit(data[[xVar]])
  b=na.omit(data[[yVar]])
  aggCor = cor(a,b)
  print(paste0('Aggregated Correlation: ',aggCor))
  cs = data[names(data) %in% c(xVar,yVar)== FALSE] # data frame
  cNamesAll = names(cs)

  # Find correlation of subgroups with categorical variables
  corCoefs = list() # initialize list
  allSub = list() # initialize list
  indName = 1
  corName = list()
    for (currentCol in cs){
      continue = FALSE
      currentCol = na.omit(currentCol)
      subs = unique(currentCol) # unique values in column
      if (class(subs)=="integer"){
        if (length(subs) < maxSubs){
          continue = TRUE
        }
      }else if (class(currentCol)=="factor"){
        continue = TRUE
      }else if (class(currentCol)=="character"){
        if (length(subs) < maxSubs){
          continue = TRUE
        }
      }else if (class(currentCol)=="numeric"){
        if (length(subs) < maxSubs){
          if (all(floor(currentCol)== currentCol)){
```

```
        continue = TRUE
      }
    }
  }
  if (continue){
    for (s in subs){
      allSub=c(allSub,s)
      subDf = data %>% filter(currentCol == s)
      subx=subDf[[xVar]]
      suby=subDf[[yVar]]
      subCor = cor(subx,suby)
      corCoefs = c(corCoefs, subCor)
      corName = c(corName, cNamesAll[indName])
    }
  }
  indName = indName + 1
}

# Test if reversal
ind = 1 # index to find string
sp = FALSE
if (length(corCoefs>0)){
  if (aggCor > 0){
    for (cc in corCoefs){
      if(!is.na(cc < 0) && cc<0){
        print(paste0(paste0(paste0(paste0(paste0("Simpson\'s Paradox
              detected: ",corCoefs[[ind]]),", "),allSub[[ind]])," "),
              corName[ind]))
        sp=TRUE
      }
      ind = ind+1
    }
}else if (aggCor < 0){
  for (cc in corCoefs){
    if(!is.na(cc > 0) && cc>0){
      print(paste0(paste0(paste0(paste0(paste0("Simpson\'s Paradox
            detected: ",corCoefs[[ind]]),", "),allSub[[ind]])," "),
            corName[ind]))
      sp=TRUE
```

```
        }
        ind = ind+1
      }
    }
  }

  # not SP
  if (sp == FALSE){
    print("Simpson\'s Paradox is not detected.")
  }
}

# test
data(auto_mpg)
simpDetector2(auto_mpg,'acceleration','mpg',15)
data(iris)
simpDetector2(iris,'Sepal.Length','Sepal.Width',1)
```

Bibliography

[1] Nazanin Alipourfard, Peter G Fennell, and Kristina Lerman. Can you trust the trend? discovering simpson's paradoxes in social data. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 19–27, 2018.

[2] Prasanta S Bandyoapdhyay, Davin Nelson, Mark Greenwood, Gordon Brittan, and Jesse Berwald. The logic of simpson's paradox. *Synthese*, 181(2):185–208, 2011.

[3] Prasanta S Bandyopadhyay, Mark Greenwood, Don Wallace F Dcruz, and Venkata Raghavan R. Simpson's paradox and causality. *American Philosophical Quarterly*, pages 13–25, 2015.

[4] Colin R Blyth. On simpson's paradox and the sure-thing principle. *Journal of the American Statistical Association*, 67(338):364–366, 1972.

[5] John Canning. Normal distribution curve in latex, Dec 2012.

[6] Clelia Di Serio, Yosef Rinott, and Marco Scarsini. Simpson's paradox in survival models. *Scandinavian journal of statistics*, 36(3):463–480, 2009.

[7] Rogier A Kievit, Willem E Frankenhuis, Lourens J Waldorp, and Denny Borsboom. Simpson's paradox in psychological science: a practical guide. *Frontiers in psychology*, 4:513, 2013.

[8] Nick Lord. 74.11 from vectors to reversal paradoxes. *The Mathematical Gazette*, 74(467):55–58, 1990.

[9] Gary Malinas and John Bigelow. Simpson's paradox. 2004.

[10] Douglas D. Mooney and Randall J. Swift. *Empirical Modeling*. Mathematical Association of America, 1999.

[11] Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2), 2000.

[12] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. Causal inference in statistics: A primer. 2016. 2016.

[13] Myra L Samuels. Simpson's paradox and related phenomena. *Journal of the American Statistical Association*, 88(421):81–88, 1993.

[14] Rahul Sharma, Huseyn Garayev, Minakshi Kaushik, Sijo Arakkal Peious, Prayag Tiwari, and Dirk Draheim. Detecting simpson's paradox: A machine learning perspective. In *International Conference on Database and Expert Systems Applications*, pages 323–335. Springer, 2022.

[15] Rahul Sharma, Minakshi Kaushik, Sijo Arakkal Peious, Markus Bertl, Ankit Vidyarthi, Ashwani Kumar, and Dirk Draheim. Detecting simpson's paradox: A step towards fairness in machine learning. In *European Conference on Advances in Databases and Information Systems*, pages 67–76. Springer, 2022.

[16] Rahul Sharma, Minakshi Kaushik, Sijo Arakkal Peious, Mahtab Shahin, Ankit Vidyarthi, Prayag Tiwari, and Dirk Draheim. Why not to trust big data: Discussing statistical paradoxes. In *International Conference on Database Systems for Advanced Applications*, pages 50–63. Springer, 2022.

[17] Aris Spanos. Yule–simpson's paradox: the probabilistic versus the empirical conundrum. *Statistical Methods & Applications*, 30(2):605–635, 2021.

[18] Shyam Sunder. Simpson's reversal paradox and cost allocation. *Journal of Accounting Research*, pages 222–233, 1983.

[19] Julius von Kügelgen, Luigi Gresele, and Bernhard Schölkopf. Simpson's paradox in covid-19 case fatality rates: a mediation analysis of age-related causal effects. *IEEE transactions on artificial intelligence*, 2(1):18–27, 2021.

[20] Clifford H Wagner. Simpson's paradox in real life. *The American Statistician*, 36(1):46–48, 1982.

[21] Jeff Witmer. Simpson's paradox, visual displays, and causal diagrams. *The American Mathematical Monthly*, 128(7):598–610, 2021.