Methods

Brain-Powered. Insight Driven. Data Analytics.

Causality

Jeremy Albright

November 13, 2018

Causality

Statistics class in the 20th Century

- "You can never prove causality."
- So researchers published results saying "x is *associated* with y."
- But then, implicitly, suggest that the association is causal.
- Example: Voters policy positions are *congruent* with positions of parties.
    - Do parties respond to the demands of voters? (Think High School Civics-view of democracy)
    - Do voters update their policy positions to match their preferred party (think of Fox News viewers evaluations of North Korea)
    - Who cares!?!?! The two seem to move together! We have *congruence*!

Causality

21st century views have evolved away from fatalism to reframe the question:

- What assumptions need to be met in order to state that an association is causal?
- Under what conditions are those assumptions met?
- Can the assumptions be met even when we can't perform randomization?

Two different approaches to the problem:

- Rubin's (elaboration on Neyman's) *potential outcomes* framework.
- Pearl's (elaboration on Sewall's) *structural causal models*.

The former is the dominant approach in applied statistics, but the latter has some important insights to add.

Potential Outcomes

Take a binary treatment $D_i \in \{0, 1\}$. Represent the outcome received by subject *i* as Y_{iD}. Then Y_{i0} and Y_{i1} are the potential outcomes.

A subject is *either* Y_{i0} *or* Y_{i1}, we don't observe both. Yet we want to determine:

$$Y_{i1} - Y_{i0}$$

which is the causal effect of the intervention.

Although subjects receive *either* 0 *or* 1, but not the other, we may be able to identify the Average Treatment Effect (ATE).

$$\text{ATE} = \mathbb{E}\left[Y_{i1} - Y_{i0}\right]$$

To derive appropriate estimators for the ATE we need to make a few assumptions. Particularly important is that the treatment is independent of potential outcomes, written as:

$$Y_{i0}, Y_{i1} \perp\!\!\!\perp D_i$$

Potential Outcomes

Finding ways to make D_i independent is at the heart of the potential outcomes framework. This leads to a few methodologies now commonplace in applied statistics:

1. Randomized experiments by definition make D_i independent.
2. Propensity score matching or weighting make the treated and control look the same on possible confounders so that the only differences must be random error.
3. Regression discontinuity designs where a cut-off on a continuous variable separate treated and control units.
4. Instrumental variables, where compliance is non-random but treatment *assignment* is random.
5. Longitudinal designs that use fixed effects or first differences to remove unit-level confounders affecting the treatment.

Potential Outcomes

The key assumption is $Y_{i0}, Y_{i1} \perp\!\!\!\perp D_i$.

From Pearl (2018, pg. 279-280):

> "Unfortunately, I have yet to find a single person who can explain what ignorability means in a language spoken by those who need to make this assumption or assess its plausibility in a given problem…If you think this sounds circular, I agree with you!"

Structural Causal Models

A different orientation - though one that often leads to similar estimators - is attributed to Judea Pearl and based on *structural causal models* (SCMs).

SCMs are graphs with nodes, directed edges, and functions mapping exogenous variables to endogenous ones. Denote U as the set of exogenous variables, V as the set of endogenous variables, and F as the set of functions mapping U to V.

A concrete example is:

$U = \{X, Y\}$

$V = \{Z\}$

$F = \{f_z\}$

where f_z is the function mapping X and Y onto Z.

This definition implies the following graph:

$$\begin{tikzcd}
X \arrow[dr] & & Y \arrow[dl] \\
& Z &
\end{tikzcd}$$

The arrows represent a generic causal relationship only, the actual function mapping X and Y onto Z can be anything we like.

These types of figures should be familiar to anybody who has previously encounted structural equation models (SEMs) in applied statistics. The primary difference is that SEMs are parametric, typically assuming a linear relationship:

$$Z = b_0 + b_1 X + b_2 Y$$

but SCMs are defined without committing to a particular functional form.

We get around functional forms by talking about the variables in terms of joint probability functions and taking advantage of well-known rules for converting between joint, conditional, and marginal probabilities.

Take the following graph:

$$\begin{tikzcd} X \arrow[r] & Y \arrow[r] & Z \end{tikzcd}$$

Any (acyclic) graph has a joint distribution that is defined by multiplying all conditional probabilities, where conditioning is performed on the direct parent.

For example The joint distribution for the variables in the model is

$$P(X, Y, Z) = P(X)P(Y|X)P(Z|Y)$$

Understanding the conditional probabilities implied by a model will enable us to generate some rules for determining how causal effects can be identified from observational data. These rules fundamentally change how statistical modeling should be approached.

"You should control for everything you can. That is, after all, why we do regression." - A prominant political science methodologist in the early 2000s.

No, you should not control for everything. In fact, depending on the causal model, some variables should explicitly *not* be controlled for. We'll start out with when you *should* control for a non-treatment variable. Take the following graph:

$$\begin{tikzcd}
Y \arrow[r] \arrow[dr] & & X \arrow[dl] \\
& Z &
\end{tikzcd}$$

We wish to know the effect of X on Z, but Y is a common cause. Let's say we could intervene in the world to set X at a given value. By doing so, we'd be removing the effect of Y on X and would be left with:

$$\begin{tikzcd}
Y \arrow[dr] & & X \arrow[dl] \\
& Z &
\end{tikzcd}$$

We can identify the causal effect by comparing the world in which we have control with the world in which we do not.

In both scenarios defined by the SCMs on the previous slide, the probability that Z takes on a value is conditioned only on Y and X, $P(Z = z \mid Y, X)$, and the probability that Y takes on a given value is not conditional on anything.

We want to know the effect of X on Z if we could intervene on X and set its value.

$$P(Z = z \mid do(X = x))$$

Based on the intervention SCM,

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

This is true because $P(Z = z \mid do(X = x))$ is what we get after integrating out Y.

But we know from comparing the graphs that $P(Z = z \mid Y = y, X = x)$ and $P(Y = y)$ are the same in both worlds. Thus, we have all the information we need to calculate a causal effect such as.

$$P(Z = z \mid do(X = 1)) - P(Z = z \mid do(X = 0))$$

Take a slightly more complicated model:

$$\begin{tikzcd}
& W \arrow[dl] \arrow[dr] & \\
X \arrow[dr] & & Y \arrow[dl] \\
& Z &
\end{tikzcd}$$

There are now two paths from X to Z:

1. $X \rightarrow Z$
2. $X \leftarrow W \rightarrow Y \rightarrow Z$

These are read from left to right regardless of the direction of the arrows. However, the arrows identify the second path as a *backdoor path* because there is an arrow leading into X.

Backdoor paths are essential for identifying causal effects because they represent spurious associations.

Blocking Backdoor Paths

Pearl shows that causal effects can be identified if we can *block* the backdoor path. We do this by conditioning on any of the variables the lay on the backdoor path, meaning the conditioning set can be any of the following:

1. {W}
2. {Y}
3. {W, Y}

We don't necessarily have to control for both, though we can.

The key is that, by blocking a backdoor path, we remove the spurious association between the outcome and X. After blocking, subsequent variables on the backdoor path may not need to be controlled for.

Now let's flip the top arrows.

$$\begin{tikzcd}
& W & \\
X \arrow[ur] \arrow[dr] & & Y \arrow[ul] \arrow[dl] \\
& Z &
\end{tikzcd}$$

This fundamentally changes the conditioning set, which now *only* contains Y.

This occurs because W is a *collider variable*, which is defined as a variable that lies along a backdoor path with arrows pointing into it from multiple directions. We would write this backdoor path as

$X \rightarrow W \leftarrow Y \rightarrow Z.$

When we write out the path in this manner we can immediately identify collider variables as those with arrows pointing to the node from both directions.

A collider variable blocks a backdoor path. The counter-intuitive result is that *conditioning on a collider opens the backdoor path*.

To identify the causal effect we need to block all backdoor paths from X to Z. The *backdoor criterion* can be defined as (Pearl, Glymour, & Powell, 2016, p. 61):

> Given an ordered pair of variables (X, Z) in a directed acyclic graph G, a set of variables V satisfies the backdoor criterion relative to (X, Z) if no node in V is a descendant of X, and V blocks every path between X and Z that contains an arrow into X.

That is, we identify a set of nodes in $\{V\}$ to condition on such that:

1. We block all spurious paths from X to Z.
2. We leave all directed paths from X to Z unperturbed.
3. We do not inadvertantly create new spurious paths via conditioning on colliders or their descendants.

Another example is *mediation*, as in the following figure:

$$\begin{tikzcd}
 & M \arrow[dr] & \\
X \arrow[ur] \arrow[rr] & & Z
\end{tikzcd}$$

We can get the *direct effect* of X on Z if we average over levels of M, which is the standard approach to mediation.

But what if we add a variable as follows?:

$$\begin{tikzcd}
X \arrow[dr] \arrow[drr, bend left] & & \\
 & M \arrow[r] & Z \\
W \arrow[ur] \arrow[urr, bend right] & &
\end{tikzcd}$$

Now M is a collider, and we know that conditioning on a collider causes problems.

Conditioning on M opens the path $X \rightarrow M \leftarrow W \rightarrow Z$, allowing an indirect effect to interfere with the direct effect.

Not conditioning on M leaves the indirect path $X \rightarrow M \rightarrow Z$ open.

How do we deal with this in a manner that allows us to recover the direct effect of X on Z? We now intervene on *both* X and M.

$$P(Z = z \mid do(X = x), do(M = m)).$$

Intervening and conditioning are not the same thing. Conditioning averages over values of M, intervening sets its value such that there are no longer the arrows $X \rightarrow M$ and $W \rightarrow M$.

The *conditional direct effect* is

$$CDE = P(Z = z \mid do(X = x), do(M = m)) - P(Z = z \mid do(X = x'), do(M = m))$$

The *conditional* refers to the fact that the direct effect $X \rightarrow Z$ may differ depending on the value to which the mediator is set.

The $do(\cdot)$ operator is equivalent to removing an arrow from a graph.

Given:

$$\begin{tikzcd}
X \arrow[dr] \arrow[drr, bend left] & & \\
& M \arrow[r] & Z \\
W \arrow[ur] \arrow[urr, bend right] & &
\end{tikzcd}$$

There is no path to X, so $do(X) = x$, and the CDE is

$$CDE = P(Z = z \mid X = x, do(M = m)) - P(Z = z \mid X = x', do(M = m)).$$

The last step is to rewrite the $do(M = m)$ in terms of the observed world. To block the backdoor path $M \leftarrow W \rightarrow Z$ we need to condition on W. We are left with:

$$CDE = \sum_i \big[P(Z = z \mid X = x, M = m, W = w) - \\ P(Z = z \mid X = x', M = m, W = w)\big]\, P(W = w)$$

Mediation

There is a general result behind this (Pearl, Glymour, & Jewell, 2016, pg. 77):

The CDE of X on Z can be identified when a mediation variable M is present given:

1. There exists a set V_1 of variables that blocks all backdoor paths from M to Z.
2. There exists a set V_2 of variables that blocks all backdoor paths from X to Z after deleting all arrows entering M.

The second of these was met automatically given the lack of parents for X.

These general rules make it possible to identify direct causal effects in contexts that were previously intractable, even if the researchers did not realize they were dealing with an intractable problem (i.e. failed to realize the direct effect was different depending on levels of M).

These models are all very simple, but graphs can be far more complex. Consider the following (adapted from Morgan & Winship, 2015, pg. 135):

$$\begin{tikzcd}
 & & & X \arrow[ddr] & \\
S \arrow[urrr] \arrow[dr] & & V \arrow[ur] \arrow[drr] & & \\
 & T \arrow[ur] \arrow[drr] & & & Z \\
U \arrow[ur] \arrow[drr] & & & W \arrow[ur] & \\
 & & Y \arrow[uurr, bend right] & &
\end{tikzcd}$$

A general approach to approaching these diagrams is to employ a tool called *d*-separation, defined as follows (Pearl, Glymour, & Powell, 2016, p. 47):

A path p is blocked by a set of nodes N iif:

1. p contains a chain of nodes $A \rightarrow B \rightarrow C$ or fork $A \leftarrow B \rightarrow C$ such that the middle node B is conditioned on, or
2. p contains a collider $A \rightarrow B \leftarrow C$ such that the collision node B is not conditioned on, nor are any descendents of B conditioned on.

Fortunately, there is software that can help us algorithmically determine which variables are *d*-separated. The software (and R package) is called `dagitty`.

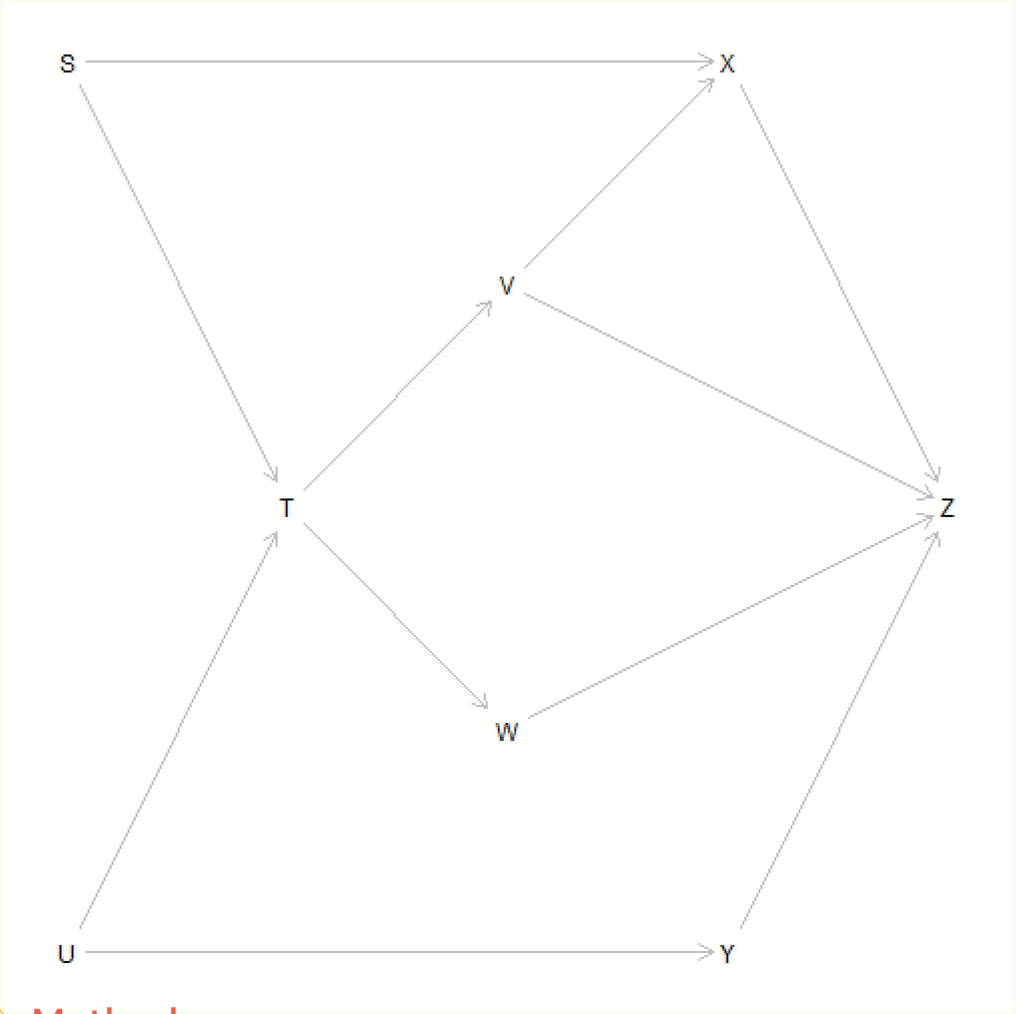
Declare the SEM:

```
g <- dagitty('dag {
                S [pos="0,0"]
                T [pos="1,2"]
                U [pos="0,4"]
                V [pos="2,1"]
                W [pos="2,3"]
                X [pos="3,0"]
                Y [pos="3,4"]
                Z [pos="4,2"]

  S -> X -> Z
  S -> T
  T -> V -> X -> Z
  T -> V -> Z
  T -> W -> Z
  U -> Y -> Z
  U -> T
  Y -> Z
}')
```

Confirm it looks good.

```
plot(g)
```



We can now make some queries on the graph. For example, what are the paths from X to Z?

```
paths(g, "X", "Z")
```

```
## $paths
## [1] "X -> Z"                          "X <- S -> T -> V -> Z"
## [3] "X <- S -> T -> W -> Z"           "X <- S -> T <- U -> Y -> Z"
## [5] "X <- V -> Z"                     "X <- V <- T -> W -> Z"
## [7] "X <- V <- T <- U -> Y -> Z"
##
## $open
## [1]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
```

We can quickly see that there are seven paths, six of which are backdoor paths, linking X to Z.

Only the fourth is blocked by the collider at T.

We wish to predict Z on the basis of X. Using the rules for d-separation to remove spurious dependencies, what set of variables can we condition on to get the true causal effect of X on X?

```r
adjustmentSets(g, "X", "Z", type = "all") %>%
  head(15)
```

```
##  { S, V }
##  { S, T, V }
##  { S, U, V }
##  { T, U, V }
##  { S, T, U, V }
##  { S, V, W }
##  { S, T, V, W }
##  { U, V, W }
##  { S, U, V, W }
##  { T, U, V, W }
##  { S, T, U, V, W }
##  { S, V, Y }
##  { T, V, Y }
##  { S, T, V, Y }
##  { S, U, V, Y }
```

Notice that T is in some of these sets. If we unblock the path $X \leftarrow S \rightarrow T \leftarrow U \rightarrow Y \rightarrow Z$, we need to reblock it by conditioning on another variable such as U or Y.

This is a lot of options. Can we get something simpler?

```
adjustmentSets(g, "X", "Z", type = "minimal")
```

```
## { V, W, Y }
## { T, V, Y }
## { U, V, W }
## { T, U, V }
## { S, V }
```

Note two important points.

1. We don't *have* to condition on all possible causes of Y.
2. There are some combinations of variables we should *not* use as adjustors.

We'll illustrate by generating some data consistent with the model.

```r
lavaan_model <- "Z ~ .8*X + .6*V + .6*W + .6*Y
                 X ~ .5*S + .5*V
                 Y ~ .5*U
                 V ~ .5*T
                 W ~ .5*T
                 T ~ .5*S + .5*U"

set.seed(12345)
g_tbl <- simulateData(lavaan_model, sample.nobs=1000)
```

We're using the `lavaan` package to generate a `data.frame` with 1000 observations.

The effect of each exogenous variable on the endogenous variables are set to be non-zero.

This is a traditional SEM, meaning that the set of functions F in the SCM are all linear.

We can verify that our data conform to the model by first specifying the model without the known coefficients.

```
lavaan_model <- "Z ~ X + V + W + Y
                 X ~ S + V
                 Y ~ U
                 V ~ T
                 W ~ T
                 T ~ S + U"
```

Next fit the model using traditional SEM.

```
lavaan_fit <- sem(lavaan_model, data = g_tbl)
```

Now look at the coefficients and verify that the path $X \to Z$ has a coefficient of approximately .8.

```r
parameterEstimates(lavaan_fit)
```

```
##    lhs op rhs    est    se      z pvalue ci.lower ci.upper
## 1    Z  ~   X  0.801 0.028 28.992      0    0.747    0.856
## 2    Z  ~   V  0.602 0.032 18.661      0    0.538    0.665
## 3    Z  ~   W  0.574 0.029 19.461      0    0.516    0.632
## 4    Z  ~   Y  0.582 0.030 19.568      0    0.523    0.640
## 5    X  ~   S  0.556 0.033 16.973      0    0.492    0.620
## 6    X  ~   V  0.485 0.028 17.251      0    0.430    0.540
## 7    Y  ~   U  0.489 0.031 15.980      0    0.429    0.549
## 8    V  ~   T  0.505 0.027 18.699      0    0.452    0.558
## 9    W  ~   T  0.489 0.026 18.971      0    0.439    0.540
## 10   T  ~   S  0.510 0.030 16.987      0    0.452    0.569
## 11   T  ~   U  0.488 0.030 16.038      0    0.428    0.548
## 12   Z ~~   Z  1.028 0.046 22.361      0    0.938    1.118
## 13   X ~~   X  1.052 0.047 22.361      0    0.959    1.144
## 14   Y ~~   Y  0.939 0.042 22.361      0    0.857    1.022
## 15   V ~~   V  1.035 0.046 22.361      0    0.944    1.126
## 16   W ~~   W  0.945 0.042 22.361      0    0.862    1.028
## 17   T ~~   T  0.928 0.041 22.361      0    0.846    1.009
## 18   S ~~   S  1.028 0.000     NA     NA    1.028    1.028
## 19   S ~~   U -0.030 0.000     NA     NA   -0.030   -0.030
## 20   U ~~   U  1.002 0.000     NA     NA    1.002    1.002
```

We want to estimate the effect of X on Z. What do we get without adjustment?

```
lm(Z ~ X, data = g_tbl) %>%
  tidy
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   0.0939    0.0500      1.88 6.06e-  2
## 2 X             1.19      0.0369     32.2  1.26e-156
```

That's NQR, the effect should be .8. What do we get if we also adjust on the collider T?

```
lm(Z ~ X + T, data = g_tbl) %>%
  tidy
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   0.0841    0.0456      1.85 6.52e-  2
## 2 X             0.974     0.0369     26.4  6.78e-117
## 3 T             0.597     0.0419     14.2  4.68e- 42
```

What if we condition using the sets we were told `dagitty` told us to?

```
lm(Z ~ X + S + V, data = g_tbl) %>%
  tidy
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   0.0675    0.0433      1.56 1.20e- 1
## 2 X             0.807     0.0422     19.1  1.31e-69
## 3 S             0.0997    0.0495      2.01 4.44e- 2
## 4 V             0.776     0.0427     18.2  5.03e-64
```

```
lm(Z ~ X + V + W + Y, data = g_tbl) %>%
  tidy
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   0.0227    0.0322     0.704 4.81e-  1
## 2 X             0.801     0.0278    28.8   3.81e-133
## 3 V             0.602     0.0322    18.7   4.64e- 67
## 4 W             0.574     0.0298    19.3   1.25e- 70
## 5 Y             0.582     0.0298    19.5   5.87e- 72
```

```r
lm(Z ~ X + U + V + W, data = g_tbl) %>%
  tidy
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   0.0568    0.0364      1.56 1.19e-  1
## 2 X             0.800     0.0316     25.3  1.49e-109
## 3 U             0.337     0.0384      8.77 7.37e- 18
## 4 V             0.585     0.0371     15.8  3.62e- 50
## 5 W             0.557     0.0344     16.2  1.67e- 52
```

```r
lm(Z ~ X + T + U + V, data = g_tbl) %>%
  tidy
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   0.0657    0.0403      1.63 1.03e- 1
## 2 X             0.820     0.0356     23.0  1.63e-94
## 3 T             0.248     0.0427      5.80 9.10e- 9
## 4 U             0.384     0.0442      8.68 1.60e-17
## 5 V             0.580     0.0430     13.5  3.48e-38
```

A few observations:

1. This approach tells us what we need to control for and when we should not control for something. Propensity scores from the potential outcomes framework offer an alternative means of adjusting. *The variables used to determine the propensity score model should follow the same principles as those outlined above*.
2. There may be some variable that affects the outcome Z that we cannot observe. This may not matter at all depending on its location in the SCM.
3. SCMs have testable implications that make it possible to determine if our SCM is bad.

For an example of the first, see Pearl, Glymour, & Jewell, 2016, pages 72-75.

The second two can be explored using `dagitty`.

Once again, take our model:

$$\begin{tikzcd}
 & & & X \arrow[ddr] & \\
S \arrow[urrr] \arrow[dr] & & V \arrow[ur] \arrow[drr] & & \\
 & T \arrow[ur] \arrow[drr] & & & Z \\
U \arrow[ur] \arrow[drr] & & & W \arrow[ur] & \\
 & & Y \arrow[uurr, bend right] & &
\end{tikzcd}$$

Let's say that we can't actually observe W or Y. An old-school regressionista would say we are SOL. A modern causal-aware practitioner would not.

We can tell `dagitty` that these variables are unobserved, or *latent*.

```r
g_unobs <- g
latents(g_unobs) <- c("W", "Y")
```

```r
adjustmentSets(g, "X", "Z", type = "minimal")
```

```
##  { V, W, Y }
##  { T, V, Y }
##  { U, V, W }
##  { T, U, V }
##  { S, V }
```

```r
adjustmentSets(g_unobs, "X", "Z", type = "minimal")
```

```
##  { T, U, V }
##  { S, V }
```

We're still okay!

How do we know our SCM is correct? This raises an important concern.

Pearl claims SCMs are unfamiliar to statisticians. Although this is true in their nonparametric form, linear SEMs have been popular ever since the software LISREL was released by a couple of Swedes in 1972.

These models are maligned by many smart statisticians, myself included, because they have been so thoroughly abused that it's become hard to take them seriously.

A typical approach:

- The effect of X on Z isn't signifcant, my dissertation (or publication needed for tenure) is a failure!
- I know, I'll add a variable M_1 between X and Z, maybe there's a mediated effect!
- Damn, no mediated effect. What if I add M_2 and M_3 to the model and keep moving around the directed arrows.
- Hey, something is eventually significant!

In other words, these models are rife with p-hacking.

A careful analysis of SCMs, however, closes off some of the models we may want to try out of desperation. This is because the conditioning we perform should render certain associations to be independent.

Take our familiar model we just analyzed. As it stands, W and V are dependent because they have a common ancestor:

$$V \leftarrow T \rightarrow W$$

But by the definition of *d*-separation, we know that conditioning on T should render W independent of V. That is,

$$P(W = w \mid V = v) = P(W = w).$$

We can test this with a regression of W on V and T. If the model is correct, the association between W and V should be zero.

Start by showing an association exists between W and V.

```
lm(W ~ V, data = g_tbl) %>%
  tidy() %>%
  mutate_if(is.numeric, funs(round(., 3)))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)    0.017     0.035     0.485   0.628
## 2 V              0.258     0.029     8.81    0
```

Controlling for T should render this association statistically indistinguishable from zero.

```r
lm(W ~ V + T, data = g_tbl) %>%
  tidy() %>%
  mutate_if(is.numeric, funs(round(., 3)))
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)    0.016     0.031     0.522   0.602
## 2 V              0.009     0.03      0.288   0.774
## 3 T              0.485     0.03     16.2     0
```

In fact, we can get all conditional independencies implied by the model.

```r
impliedConditionalIndependencies(g) %>%
  head(20)
```

```
## S _||_ U
## S _||_ V | T
## S _||_ W | T
## S _||_ Y
## S _||_ Z | V, W, X, Y
## S _||_ Z | T, V, X, Y
## S _||_ Z | U, V, W, X
## S _||_ Z | T, U, V, X
## T _||_ X | S, V
## T _||_ Y | U
## T _||_ Z | V, W, X, Y
## T _||_ Z | U, V, W, X
## T _||_ Z | S, V, W, Y
## T _||_ Z | S, U, V, W
## U _||_ V | T
## U _||_ W | T
## U _||_ X | S, V
## U _||_ X | S, T
## U _||_ Z | V, W, X, Y
## U _||_ Z | S, V, W, Y
```

Methods
Brain-Powered. Insight Driven. Data Analytics.

We generated our data to intentionally be consistent with the model, so testing these conditional independencies will confirm them.

When we don't know if the model is correct, we can generate the conditional independencies and check each of them. When they're not correct, our model is wrong.

Counterfactuals

SCMs, and the implied probabilities, can be used to address seemingly intractable questions. Specifically, they can address unit-specific *counterfactuals*. Whereas interventions, and determining ATEs, can be performed by averaging across a group of cases, specific counterfactuals relate to an individual case.

At first, counterfactuals seem entirely intractable. Think of a court case where there is an assertion that taking a drug caused a person's death. There are two (potential) outcomes:

1. Z_0, the outcome when the person did not take the drug, i.e. $X = 0$.
2. Z_1, the outcome when the person did take the drug, i.e. $X = 1$.

The person took the drug and died, so we know $Z_1 = 1$ (1 = death, 0 = no death).

The defense would like to know $P(Z_0 \mid X = 1, Y = 1)$. But this seems like nonesense. We want to know the probability of an event under one hypothetical world while conditioning on another world, the one we observed.

Methods
Brain-Powered. Insight Driven. Data Analytics.

Counterfactuals

The solution relies on establishing an SCM that explicitly includes error terms.

$$\begin{tikzcd}
 & & UZ \arrow[dr] & \\
 & UY \arrow[dr] & & Z \\
 & & Y \arrow[ur] & \\
UX \arrow[r] & X \arrow[ur] & &
\end{tikzcd}$$

Each of the $U \in \{UX, UY, UZ\}$ is an individual-specific value. After fitting the model using the observed data, we can get these values for a specific person.

We then alter the graph by setting the value of X or Y to the counterfactual value and solve for Z using the error term value identified by the full regression.

SCMs and ML

Pearl (2018) makes the audacious claim that current machine learning models cannot ever assert causality because they cannot deal with interventions, let alone counterfactuals.

A machine learning model takes a set of features $V = \{v_1, v_2, \ldots, v_k\}$ and finds a function f_z mapping this set onto an outcome Z.

Using variations on statistical modeling, this amounts to modeling the joint distribution of all variables. But a joint distribution *changes* when we intervene on a variable.

For example, if we are given a data set without knowing where it came from, we can fit a regression model using the joint distribution. But we wouldn't know that X is randomized or not.

Causality requires knowing which conditional probabilities are invariant to changes in the structural model. ML is blind to this.

ML as currently practiced throws a bunch of stuff into a blender and sees what comes out, akin to 20th century regression modeling that taught us to "control for everything."

This may not matter when we want to predict the presence of a dog, cat, or hot dog in a picture. It *will* matter if we want to:

1. Tell policymakers whether or not to increase the minimum wage.
2. Determine if admissions criteria at a university are racially biased.
3. Find a defendant guilty in a criminal trial.
4. Determine a counterfactual for an individual for whom existing data are not representative.

ML models are akin to the underwear gnome problem:

1. Features.
2. dot dot dot.
3. Prediction!

The black box hides the answer we need.

Limitations of the Pearlian Weltanshauung

At the same time, Pearl's dismissal of non-SCM approaches to modeling (potential outcomes, ML) are based on finding specific cases where these approaches fail, but he does not give a sense as to how often they fail.

Take, for example, our apparently complicated model:

$$\begin{tikzcd}
S \arrow[rrr] \arrow[ddr] & & & X \arrow[ddr] & \\
& & V \arrow[ur] \arrow[drr] & & \\
& T \arrow[ur] \arrow[drr] & & & Z \\
U \arrow[ur] \arrow[drr] & & & W \arrow[ur] & \\
& & Y \arrow[uurr, bend right] & &
\end{tikzcd}$$

We can identify the *canonical set* of adjuster variables, which will be valid if any valid set exists.

```
adjustmentSets(g, "X", "Z", type = "canonical")
```

```
##  { S, T, U, V, W, Y }
```

```
lm(Z ~ ., data = g_tbl) %>%
  tidy
```

```
## # A tibble: 8 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   0.0236    0.0322     0.732 4.64e-  1
## 2 X             0.815     0.0314    26.0   4.54e-114
## 3 Y             0.562     0.0332    16.9   1.39e- 56
## 4 V             0.604     0.0349    17.3   7.55e- 59
## 5 W             0.583     0.0332    17.5   4.11e- 60
## 6 T            -0.0466    0.0395    -1.18  2.38e-  1
## 7 S            -0.0106    0.0398    -0.266 7.90e-  1
## 8 U             0.0692    0.0395     1.75  8.05e-  2
```

We didn't do too bad. The problem, of course, is that there are SCMs that do not have all IVs or features as a proper adjustment set. How bad our conclusions are will depend on how well our representation of reality is.

This, of course, begs the question of how we draw our SCM in the first place.

When $\{V\}$ is large, the possible set of connections may not all be clearly dictated by theory, and the number of possible combinations of arrows is two large to test via a grid-search.

These examples assume acyclicity, as due most of Pearl's introductory examples. Observational data is rife with reverse causation. Instrumental variables assist in simple situations, but what about bi-directional causality affecting variables needed to black a backdoor path?

Not all models are identified, especially when stepping away from the world of linearity.