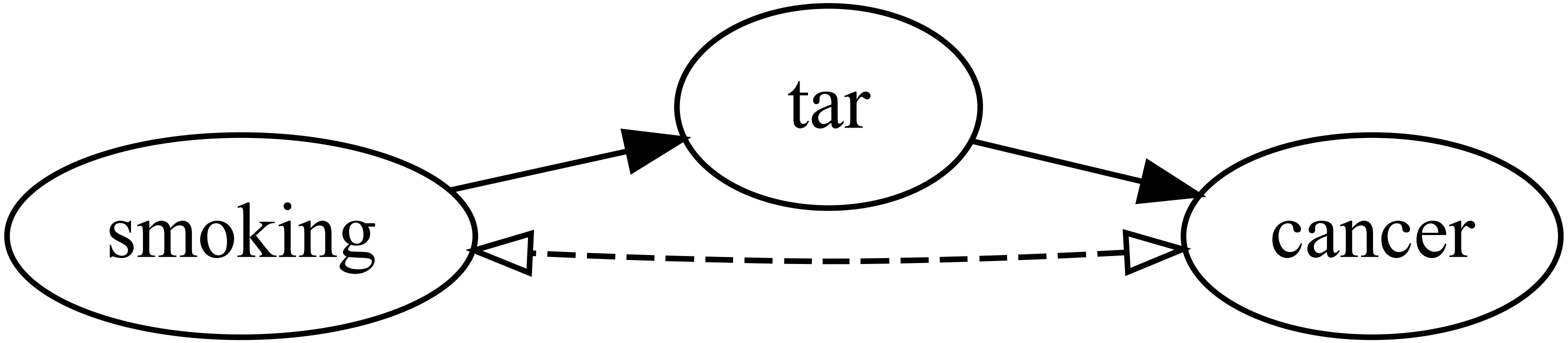
Causal Programming

Joshua Brulé

Smoking/cancer structural causal model

$$\text{smoking} = f_1(\epsilon_1)$$
$$\text{tar} = f_2(\text{smoking}, \epsilon_2)$$
$$\text{cancer} = f_3(\text{tar}, \epsilon_3)$$
$$\epsilon_1 \not\perp\!\!\!\perp \epsilon_3$$



Causal calculus (Pearl 1995)

$$P(y \mid \hat{x}, z, w) = P(y \mid \hat{x}, w) \text{ if } (Y \perp\!\!\!\perp Z \mid X, W)_{G_{\overline{X}}}$$

$$P(y \mid \hat{x}, \hat{z}, w) = P(y \mid \hat{x}, z, w) \text{ if } (Y \perp\!\!\!\perp Z \mid X, W)_{G_{\overline{X}\underline{Z}}}$$

$$P(y \mid \hat{x}, z, w) = P(y \mid \hat{x}, w) \text{ if } (Y \perp\!\!\!\perp Z \mid X, W)_{G_{\overline{X}, \overline{Z(W)}}}$$

- W, X, Y, Z - nodes in a causal DAG G
- $G_{\overline{X}}$ delete edges pointing into X
- $G_{\underline{X}}$ denotes delete edges emanating from X
- $Z(W)$ Z-nodes that are not ancestors of any W-node
- **Note: $\mathbf{P(y \mid do(x))}$ abbreviated $\mathbf{P(y \mid \hat{x})}$**

Example proof

$$
\begin{aligned}
P(y \mid \hat{x}) &= \sum_z P(y \mid z, \hat{x}) P(z \mid \hat{x}) && \text{(law of total probability)} \\
&= \sum_z P(y \mid \hat{z}, \hat{x}) P(z \mid \hat{x}) && \text{(rule 2)} \\
&= \sum_z P(y \mid \hat{z}) P(z \mid \hat{x}) && \text{(rule 3)} \\
&= \sum_z \left[\sum_x P(y \mid x, \hat{z}) P(x \mid \hat{z}) \right] P(z \mid \hat{x}) && \text{(law of total probability)} \\
&= \sum_z \left[\sum_x P(y \mid x, \hat{z}) P(x) \right] P(z \mid \hat{x}) && \text{(rule 3)} \\
&= \sum_z \left[\sum_x P(y \mid x, z) P(x) \right] P(z \mid \hat{x}) && \text{(rule 2)} \\
&= \sum_z \left[\sum_x P(y \mid x, z) P(x) \right] P(z \mid x) && \text{(rule 2)}
\end{aligned}
$$

Causation coefficient

Correlation is not causation

"Correlation is not causation but it sure is a hint."

"Empirically observed covariation is a necessary but not sufficient condition for causality."

—Edward Tufte

Correlation coefficient

$$\rho = \frac{cov(X,Y)}{\sqrt{Var[X]Var[Y]}}$$

$$\rho = \frac{\sum_x \sum_y xyP(x,y) - \sum_x xP(x) \sum_y yP(y)}{\sqrt{(\sum_x x^2 P(x) - (\sum_x xP(x))^2)(\sum_y y^2 P(y) - (\sum_y yP(y))^2)}}$$

Correlation coefficient (rewritten)

$$Var[X] = \sum_x x^2 P(x) - (\sum_x xP(x))^2$$

$$Var[Y] = \sum_x \sum_y y^2 P(y|x)P(x) - (\sum_x \sum_y yP(y|x)P(x))^2$$

$$\rho = \frac{\sum_x \sum_y xyP(y|x)P(x) - \sum_x xP(x) \sum_x \sum_y yP(y|x)P(x)}{\sqrt{Var[X]Var[Y]}}$$

Defining the causation coefficient

- Substitute $P(y \mid do(x))$, **abbreviated $\mathbf{P(y \mid \hat{x})}$** for $P(y \mid x)$
  - i.e. Replace observational distribution with interventional distribution
- Substitute $\hat{P}(x)$ for $P(x)$
  - 'Distribution of interventions'
  - Interpret as the relative cohort sizes in an experimental study
  - Natural causation coefficient: $\hat{P}(x) = P(x)$

Causation coefficient

$$Var[\hat{X}] = \sum_x x^2 \hat{P}(x) - (\sum_x x\hat{P}(x))^2$$

$$Var_{\hat{X}}[Y] = \sum_x \sum_y y^2 P(y|\hat{x})\hat{P}(x) - (\sum_x \sum_y yP(y|\hat{x})\hat{P}(x))^2$$

$$\gamma_{X \to Y} = \frac{\sum_x \sum_y xyP(y|\hat{x})\hat{P}(x) - \sum_x x\hat{P}(x) \sum_x \sum_y yP(y|\hat{x})\hat{P}(x)}{\sqrt{Var[\hat{X}]Var_{\hat{X}}[Y]}}$$

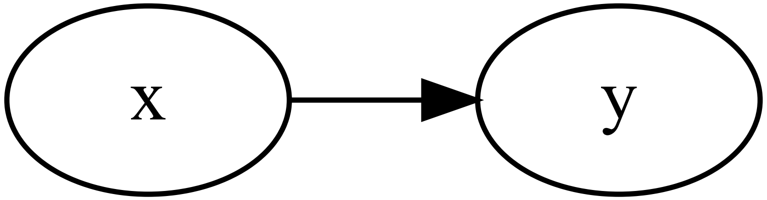
Interpretation of γ

- $\rho = \pm 1$ - perfect positive/negative linear correlation
- $\gamma = \pm 1$ - perfect positive/negative linear causation
- $\rho = 0$ - "linearly uncorrelated"
- $\gamma = 0$ - "linearly acausal"

No-confounding

$P(y \mid x) = P(y \mid \hat{x})$ implies $\gamma_{X \to Y} = \rho$

Converse holds for Bernoulli (binary) random variables



Independence and Invariance

Definitions:

- X and Y are *independent* iff $P(y \mid x) = P(y), \forall x, y$
- Y is *invariant* to X iff $P(y \mid \hat{x}) = P(y), \forall x, y$

Lemmas:

- For Bernoulli $X, Y, \rho = 0$ iff X and Y are independent
- For Bernoulli $X, Y, \gamma_{X \to Y} = 0$ iff Y is invariant to X

Average treatment effect

For Bernoulli random variables:

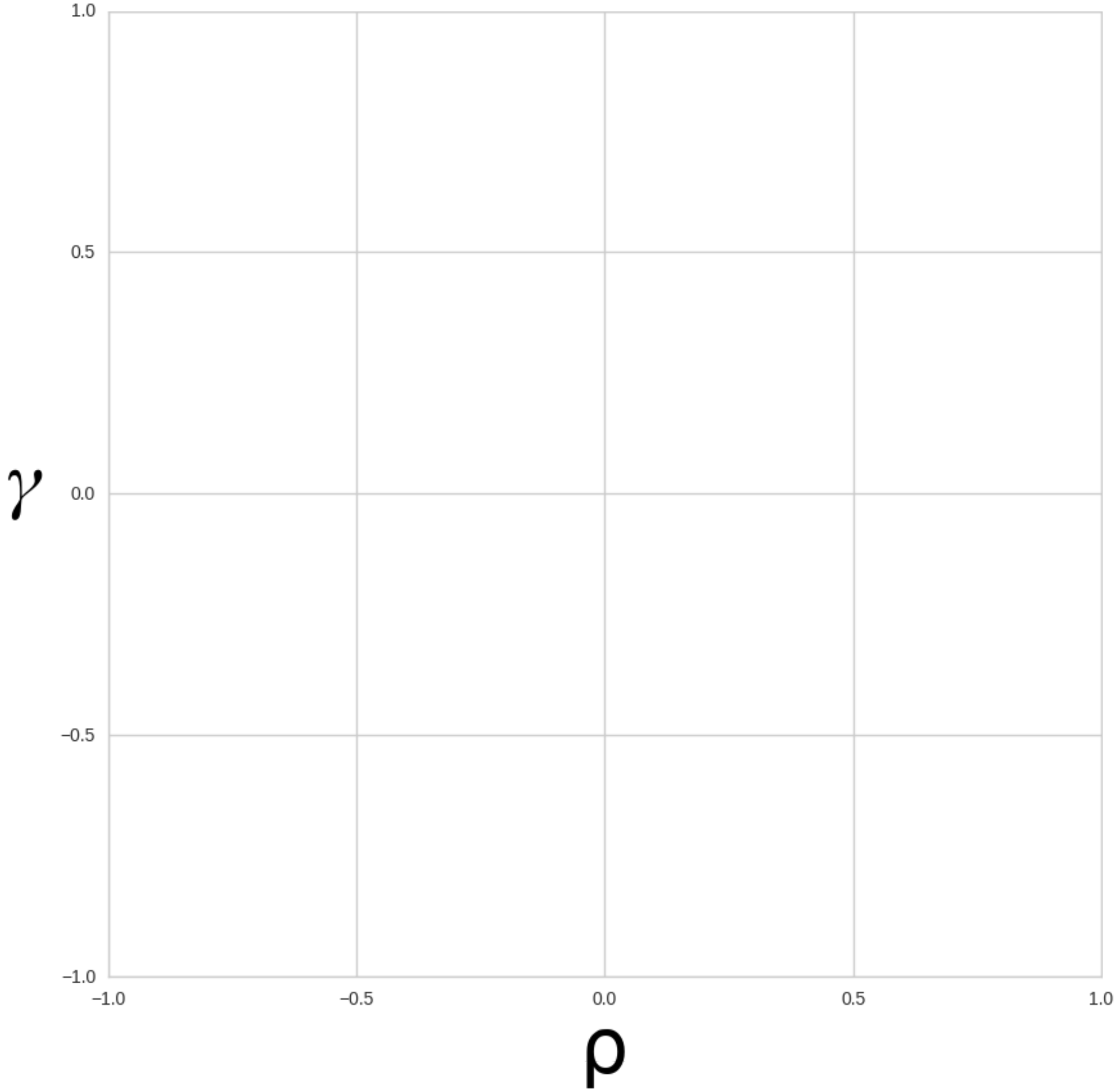
$$ATE(X \to Y) \equiv P(Y = 1 \mid do(X = 1)) - P(Y = 1 \mid do(X = 0))$$

$$\gamma_{X \to Y} = \mathrm{ATE}(X \to Y)\sqrt{\frac{Var[\hat{X}]}{Var_{\hat{X}}[Y]}}$$

- γ has the same sign as $\mathrm{ATE}(X \to Y)$
- $\mathrm{ATE}(X \to Y)$ > 0 - treatment is more effective
- $\mathrm{ATE}(X \to Y)$ < 0 - treatment is less effective

Plot causation vs correlation

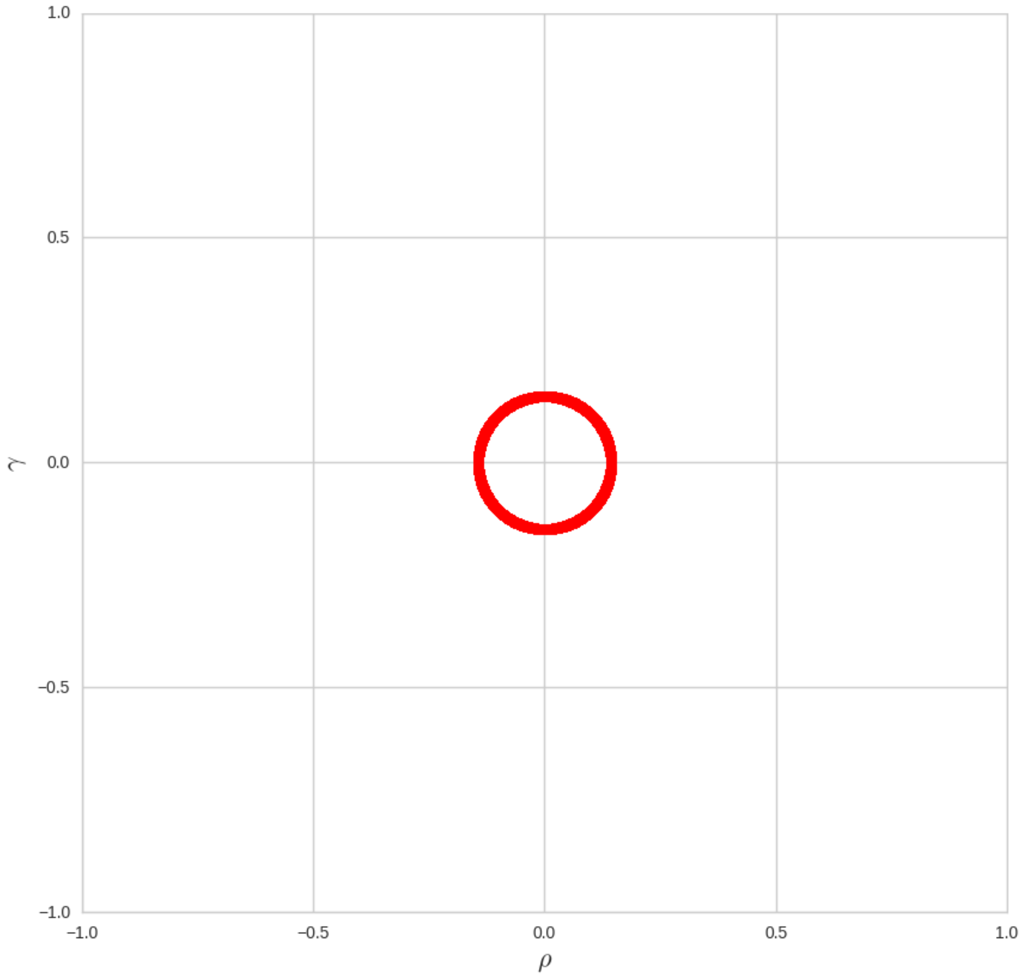
Every point on a $\gamma\rho$ plot is a structural causal model



1.0
0.5
0.0
−0.5
−1.0
γ
−1.0
−0.5
0.0
0.5
1.0
ρ

Invariant and independent

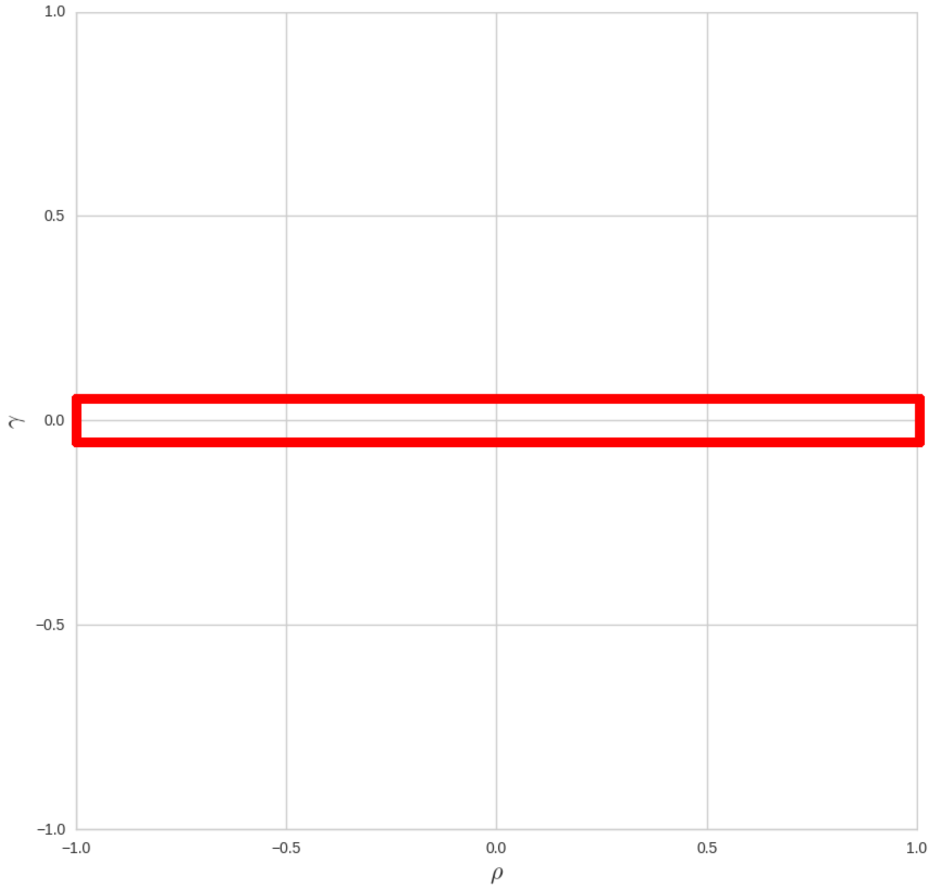
- Neither manipulation nor observation of X changes/provides information about Y
    - e.g. Two events outside each other's past and future light cone



1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
γ
ρ

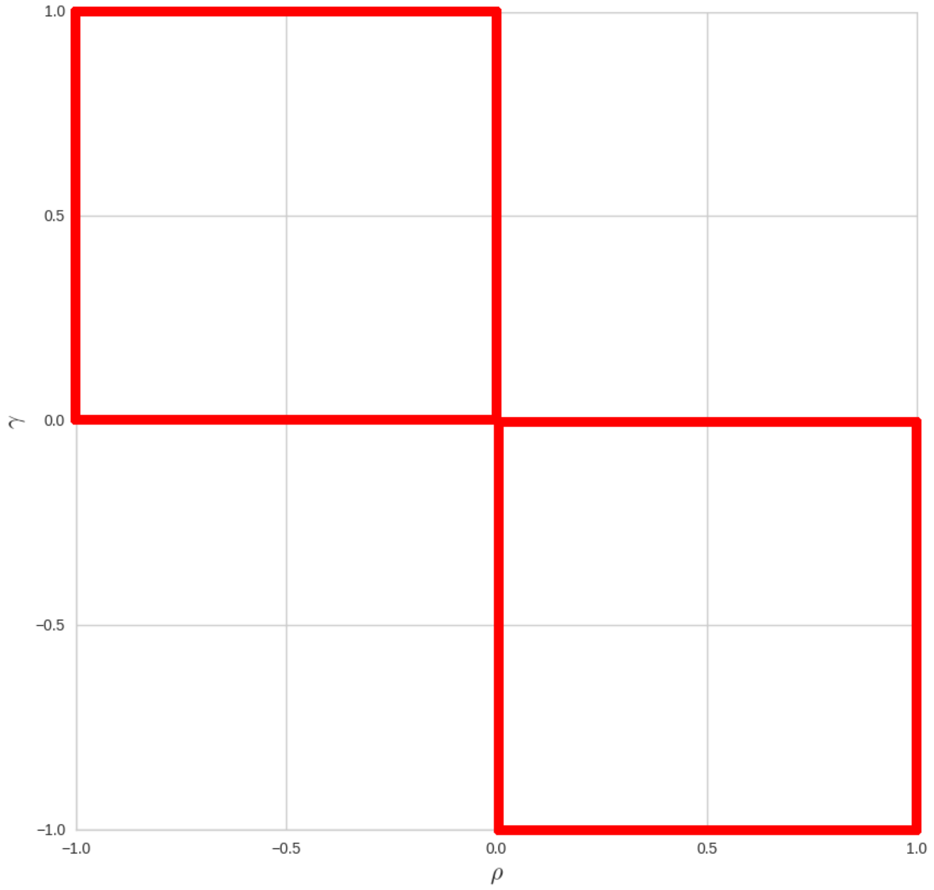
Causation vs. correlation: common causation

- "If an improbable coincidence has occurred, there must exist a common cause" (Reichenbach 1956)
  - e.g. Myopia and ambient lighting at night (Quinn et al. 1999)



Inverse causation

- ρ and γ have the opposite sign
  - e.g. Tuberculosis in Arizona (Gardner 1982)



Example model: inverse causation

Let $\epsilon_Z \sim \mathrm{Bernoulli}(1/2)$ and $\epsilon_Y \sim \mathrm{Bernoulli}(3/4)$. The following model exhibits inverse causation:

$$
\begin{aligned}
Z &= \epsilon_Z \\
X &= Z \\
Y &= \begin{cases} \neg Z & \text{if } \epsilon_Y = 1 \\ X & \text{if } \epsilon_Y = 0 \end{cases}
\end{aligned}
$$

Inverse causation probability distributions

Table 3.8: Observational distribution of inverse causation model

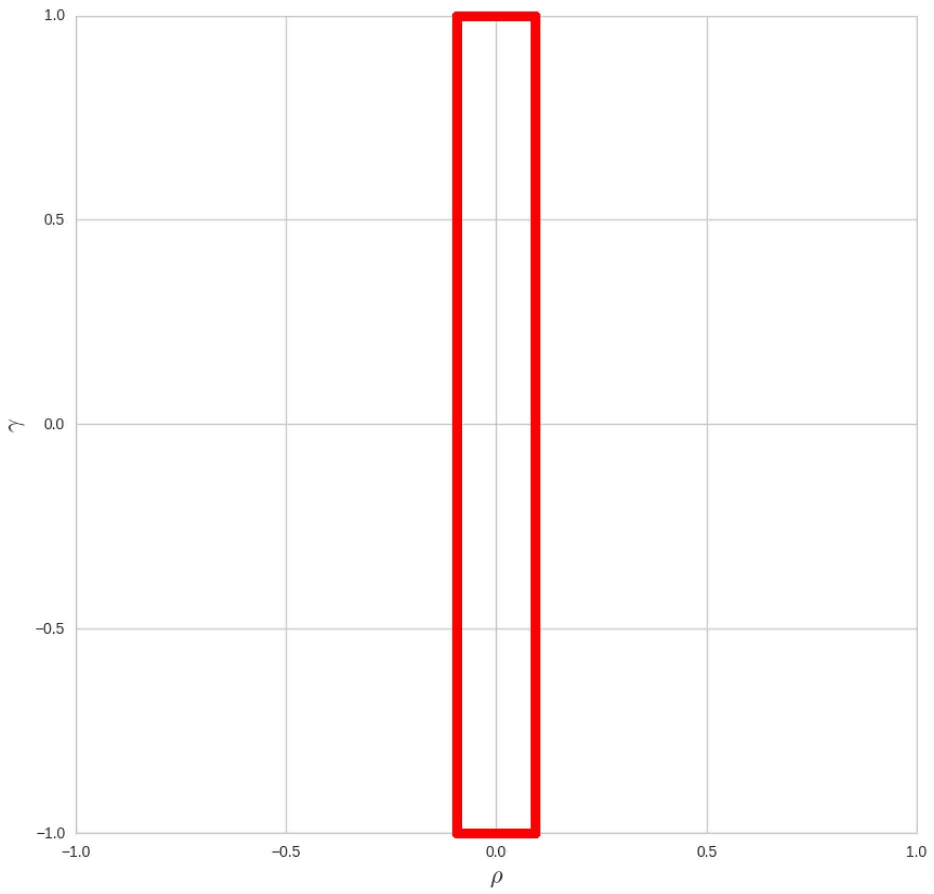
$P(x,y)$	y=0	y=1	$P(x)$
x=0	1/8	3/8	1/2
x=1	3/8	1/8	1/2
$P(y)$	1/2	1/2	

Table 3.9: Interventional distributions of inverse causation model

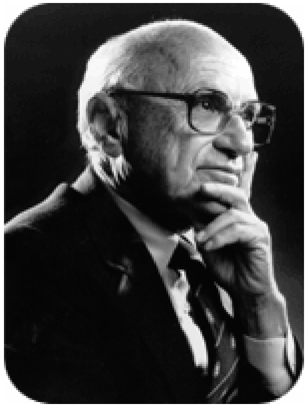
$P(y \mid do(x))$	y=0	y=1
x=0	5/8	3/8
x=1	3/8	5/8

Causation vs. correlation: unfaithfulness

- X and Y are *unfaithful* if they are independent but not invariant
  - I define this as a 'local' version of unfaithful distribution (Spirtes et al. 1993)



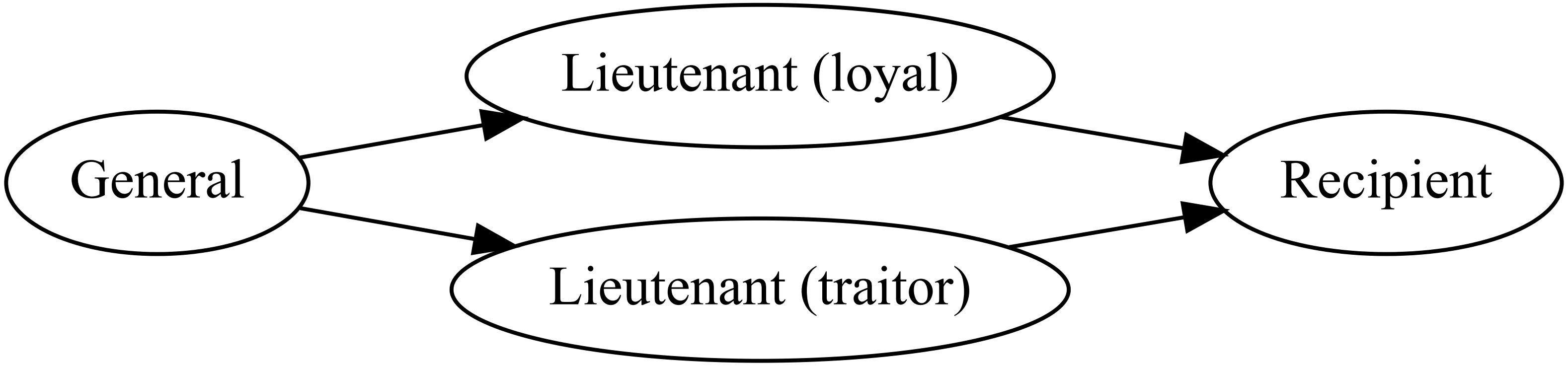
”Friedman’s thermostat”



- Observe correlation between furnace and outside temperature
- Observe **no** correlation between furnace and inside temperature
- Observe **no** correlation between inside and outside temperature

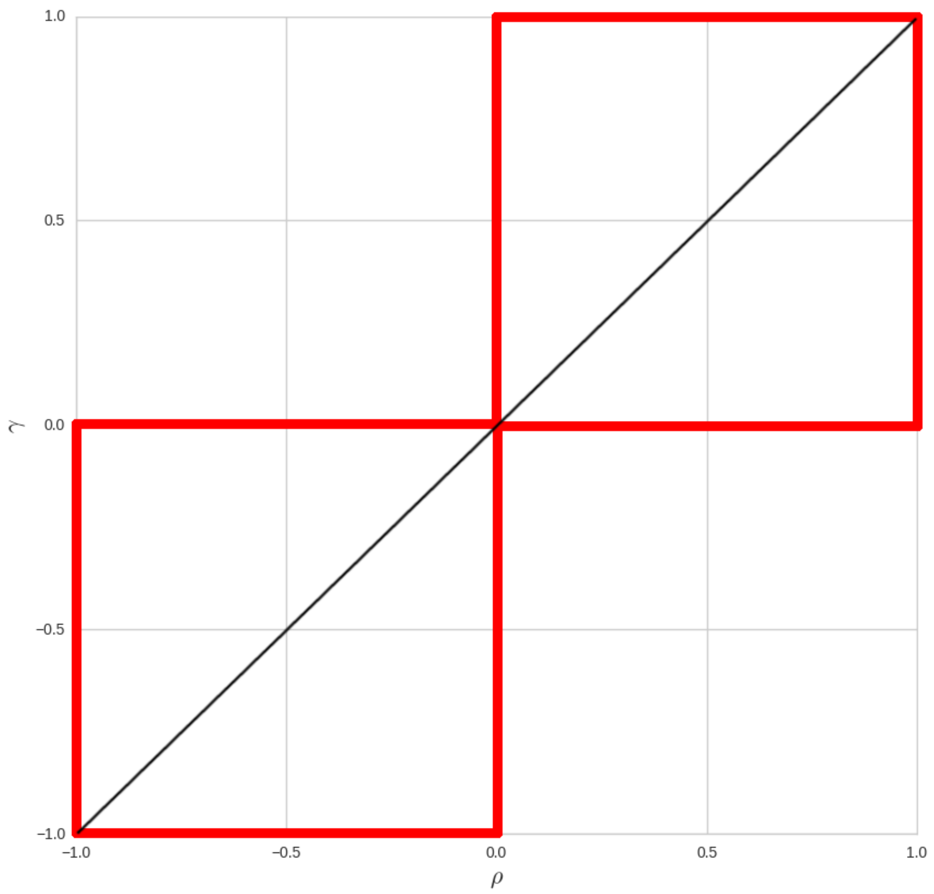
"Traitorous lieutenant"

- General wishes to send one bit, recipient XORs bits
- For `1`, send `(0, 1)` or `(1, 0)` with equal probability
- For `0`, send `(1, 1)` or `(0, 0)` with equal probability



Genuine causation and confounding bias

- ρ and γ have the same sign
  - May be biased by confounders

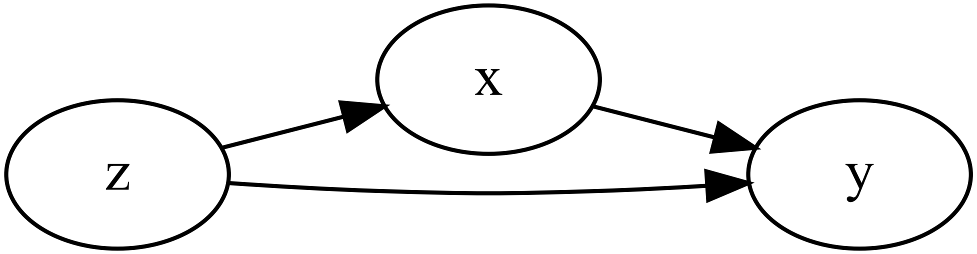


Recovering intuition: Why do we think correlation $\approx$ causation?

- Need a way to analyze behavior of 'typical' models
- Don't draw samples from a model, draw *models* from a space of models
- How to parameterize that space?

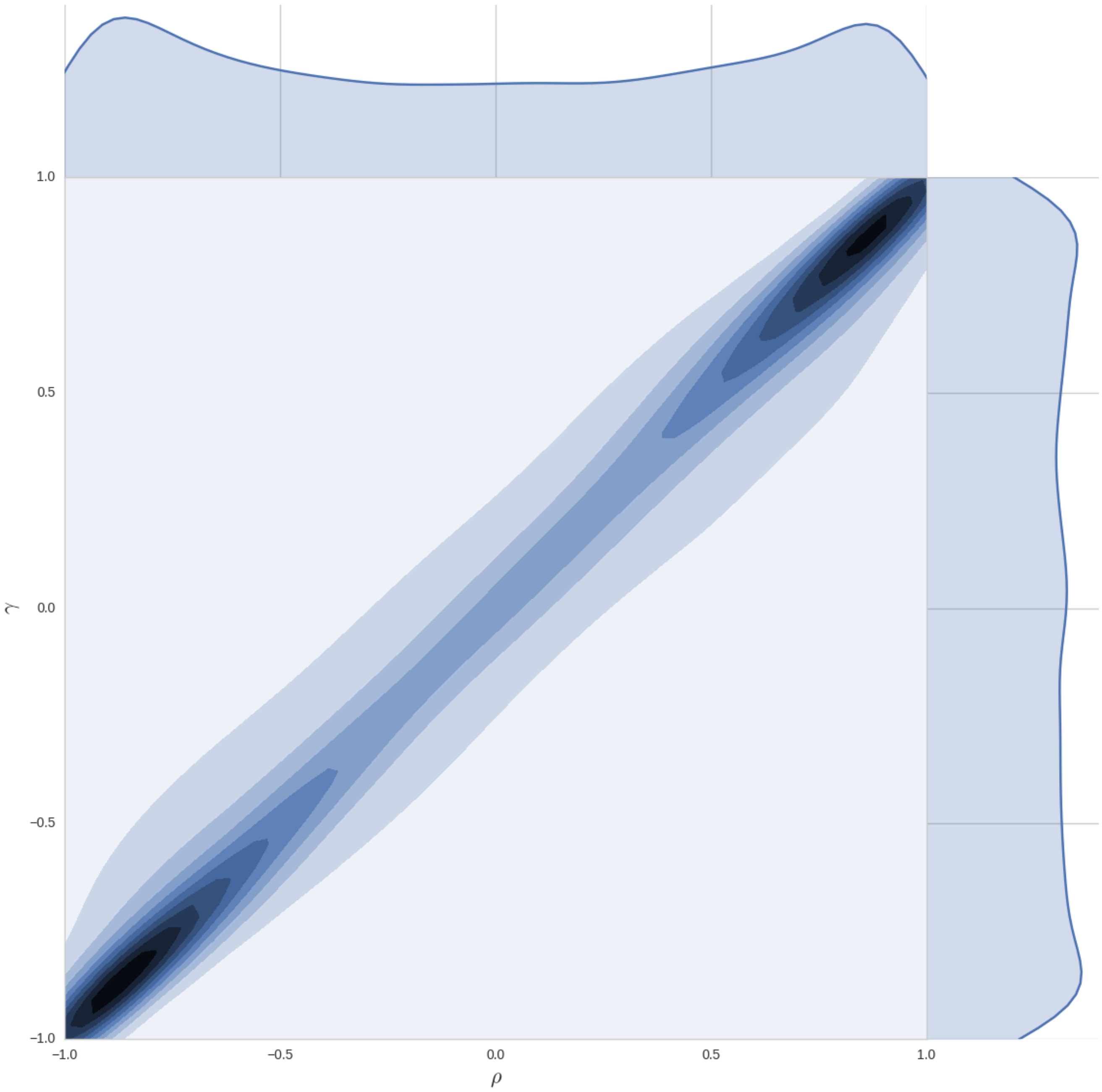
Parameterization

$$Z = \epsilon_Z$$
$$X = \alpha_Z Z + \epsilon_X$$
$$Y = \beta_X X + \beta_Z Z + \epsilon_Y$$



- Draw a sample model M from maximum entropy distribution over the parameters
- Compute (ρ, γ) for M
- Plot a kernel density estimate

Causation vs correlation ($\approx$ 12% inverse causation)



1.0
0.5
0.0
−0.5
−1.0
γ
−1.0
−0.5
0.0
0.5
1.0
ρ

Correlation/causation relationships

- Most of these effects were known, not all were named
- γ, ρ provides unified framework (population, acyclic)
- Intuition for why correlation $\approx$ causation
- Other relationships:
    - Spurious correlation (population vs sample distribution)
    - Mutual causation (not in acyclic models)
    - Reverse causation (confusing $X \rightarrow Y$ for $Y \rightarrow X$)

No substitute for proper causal analysis

Causal programming

Declarative programming (”what” instead of ”how”)

- (Purely) functional programming
  - Functions, algebraic data types
  - Function application
- Logic programming
  - First-order horn clauses
  - Resolution
- Linear programming
  - Linear objective function, linear constraints
  - Optimize
- Probabilistic programming
  - Various
  - Conditional sampling

Causal inference relation

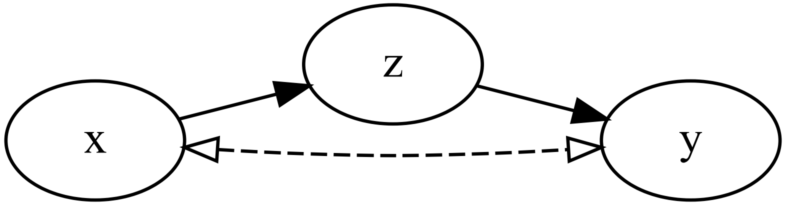
$$\langle M, D, Q, F \rangle_V$$

- M - set of structural causal models
- D - set of distributions; known probability functions
- Q - query from the causal hierarchy (Shpitser 2008), e.g. $P(y \mid x), P(y \mid do(x))$
- F - formula that computes Q as a function of D for every model in M

- V - set of endogenous variables (usually implicit)

Identification (find F)

Model, $M =$



Distribution, Query

$D = P(x, y, z), Q = P(y \mid do(x))$

Formula

$\sum_z P(z \mid x) \sum_{x'} P(y \mid x', z) P(x')$

Causal discovery (find M)

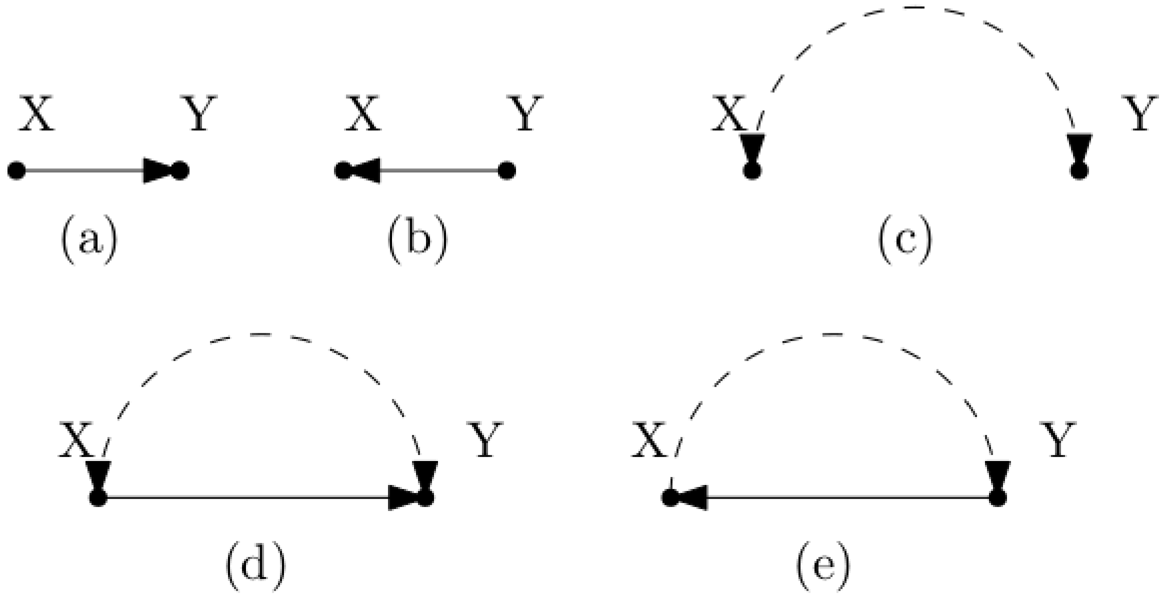
Distribution, Query

$D = P(x, y)$, where $X \not\perp\!\!\!\perp Y$
$Q = P(y \mid do(x))$

Solutions

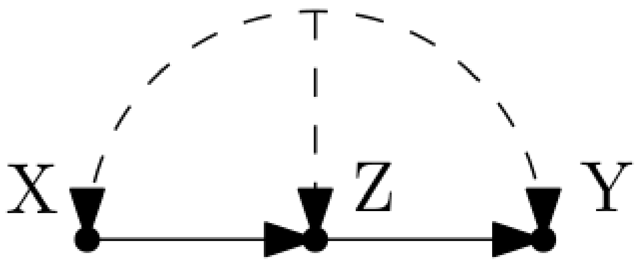
$\langle M_1, D, Q, F_1 \rangle, \langle M_2, D, Q, F_2 \rangle$, where: $M_1 = (a), F_1 = P(y \mid x)$ $M_2 = (b)$, $F_2 = P(y)$

Models



Context matters

There always exist compatible models where identification is impossible

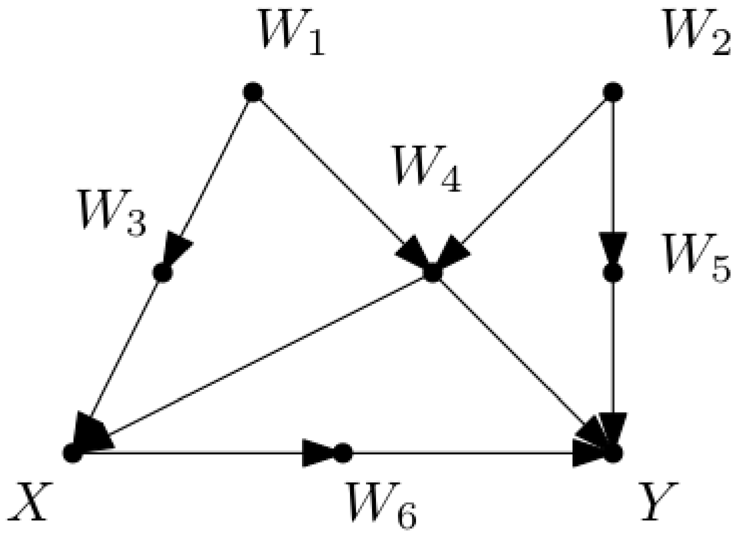


Research design (find D)

Solutions

$\langle M, D_1, Q, F_1\rangle, \langle M, D_2, Q, F_2\rangle$ $F_1 = \sum_{w_3,w_4} P(y \mid w_3, w_4, x)P(w_3, w_4)$
$F_2 = \sum_{w_4,w_5} P(y \mid w_4, w_5, x)P(w_4, w_5)$ $D_1 = P(x, y, w_3, w_4)$
$D_2 = P(x, y, w_4, w_5)$

Model

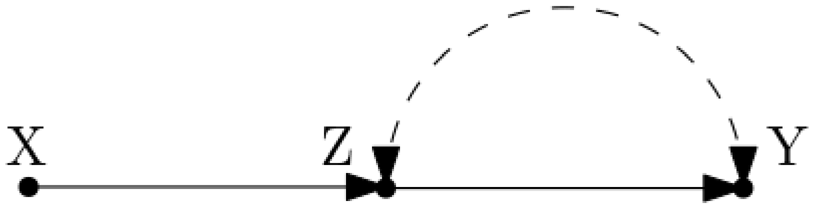


Query

$Q = P(y \mid do(x))$

Query generation (find Q)

"Testable implications"



e.g. Can identify $P(y \mid do(x))$ and $P(z \mid do(x))$, but not $P(y \mid do(z))$

Optimization problems

Cost function over M, D, Q

- M - favor simple models (Occam's razor)
- D - optimal research design
- Q - (inverse) value of information

”Meta-theory” / ”Framework”

- Sensitive to domains of M, D, Q
- Specify domains to get usable/implementable theory
- Framework to classify existing methods/problems

(Some) Prior work / existing algorithms

Identification

- ID (Shpitser 2006): M = (causal diagrams), D = $P(v)$, Q = $P(y \mid do(x))$
- IDC* (Shpitser & Pearl 2007): M = "", D = $P(v \mid do(z)) \forall Z \subseteq V$, Q = $P(\alpha \mid \beta)$
- zID (Bareinboim 2012): M = "", D = $P(v \mid do(z))$, Q = $P(y \mid do(x))$
- Selection bias (Bareinboim 2014): M = "", D = $P(v \mid S = 1)$, Q = $P(y \mid do(x))$

Causal discovery

- Inductive causation based algorithms, e.g. PC, FCI

Research design / query generation (research opportunity?)

- Informally studied, no formal algorithms?

Causal programming language

Learn Lisp in < 1 minute

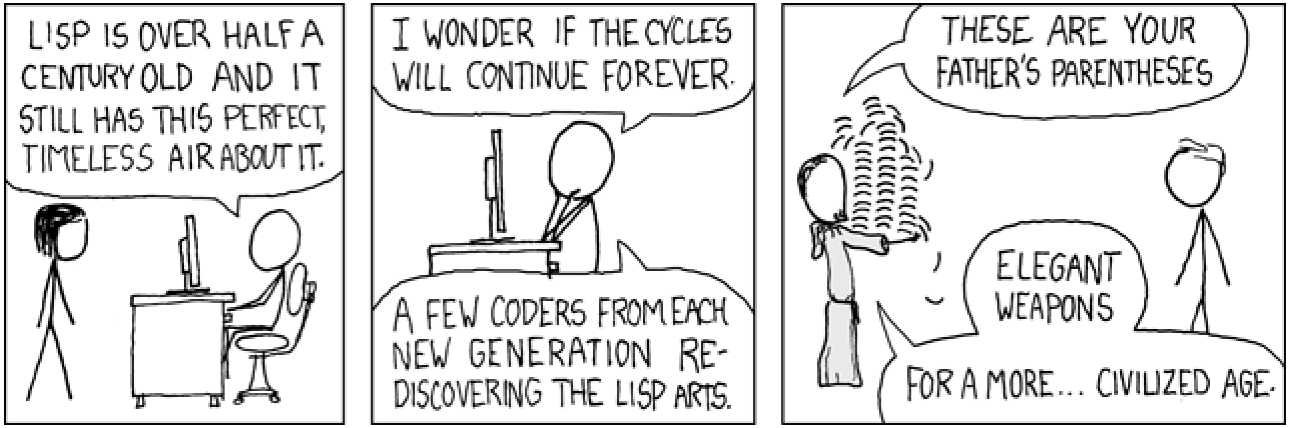
Everything is a function call

Move the left parentheses one word to the left

```
load_image("xkcd-297.png")
```

In [2]:
```
(load-image "xkcd-297.png")
```

Out[2]:



"Core" Whittemore

- `(model {:x [], :y [:x]})` - a (set of) structural causal model(s)
- `(data [:x :y])` - the "signature" of a distribution, e.g. $P(x, y)$
- `(q [:y] :do [:x])` - a query, e.g. $P(y \mid do(x))$
- `(identify m d q)` - returns a formula
- `(estimate distribution formula)` - applies formula to distribution

Example: Treatment of renal calculi (Charig et al. 1986)

Load data

In [3]:

```
(def kidney-dataset
  (read-csv "data/renal-calculi.csv"))

(count kidney-dataset)
```

Out[3]:

```
700
```

In [4]:

```
(head kidney-dataset)
```

Out[4]:

:size	:success	:treatment
"small"	"yes"	"surgery"
"large"	"yes"	"nephrolithotomy"
"small"	"yes"	"surgery"
"small"	"yes"	"surgery"
"large"	"yes"	"nephrolithotomy"
"large"	"yes"	"surgery"
"small"	"yes"	"nephrolithotomy"
"small"	"yes"	"surgery"
"large"	"no"	"nephrolithotomy"
"large"	"yes"	"nephrolithotomy"

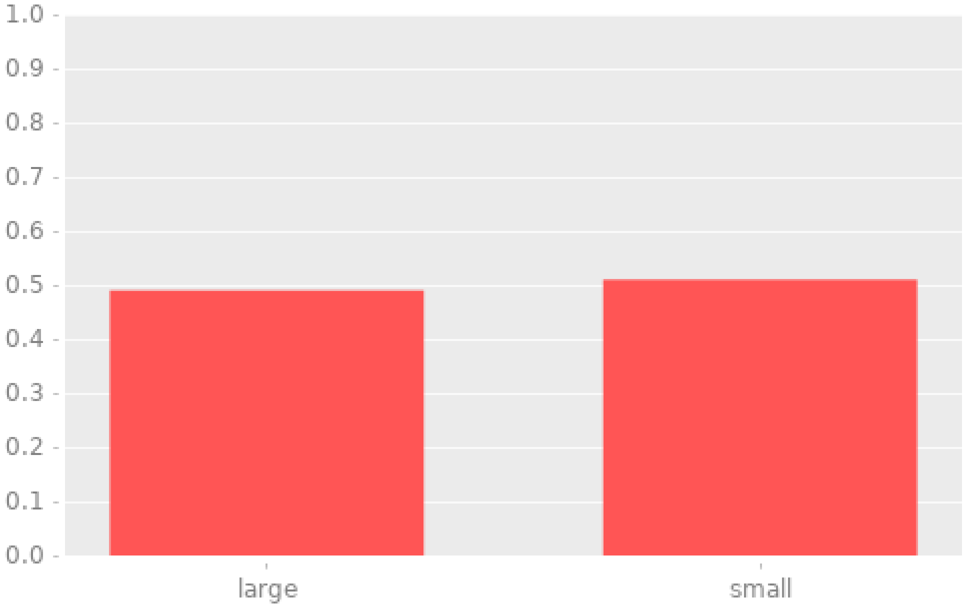
Categorical distribution

In [5]:

```
(def kidney-distribution
  (categorical kidney-dataset))

(plot-univariate kidney-distribution :size)
```

Out[5]:



Simpson's paradox

P(success=yes | treatment=surgery) < P(success=yes | treatment=nephrolithotomy)

In [7]:
```
(estimate kidney-distribution
  (q {:success "yes"} :given {:treatment "surgery"}))
```

Out[7]:
```
0.78
```

In [8]:
```
(estimate kidney-distribution
  (q {:success "yes"} :given {:treatment "nephrolithotomy"}))
```

Out[8]:
```
0.8257142857142857
```

P(success=yes | treatment, size=small)

In [9]:
```
(estimate kidney-distribution
  (q {:success "yes"} :given {:treatment "surgery" :size "small"}))
```

Out[9]: 0.9310344827586207

In [10]:
```
(estimate kidney-distribution
  (q {:success "yes"} :given {:treatment "nephrolithotomy" :size "small"}))
```

Out[10]: 0.8666666666666667

P(success=yes | treatment, size=large)

In [11]:
```
(estimate kidney-distribution
  (q {:success "yes"} :given {:treatment "surgery" :size "large"}))
```

Out[11]: 0.7300380228136882

In [12]:
```
(estimate kidney-distribution
  (q {:success "yes"} :given {:treatment "nephrolithotomy" :size "large"}))
```

Out[12]: 0.6875

Model assumptions

$$\text{size} = f_{\text{size}}(\epsilon_{\text{size}})$$

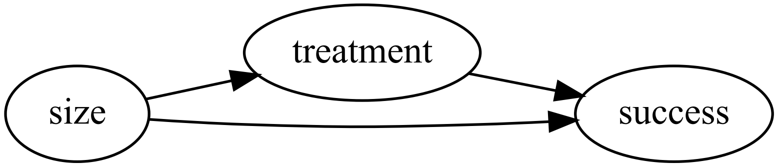
$$\text{treatment} = f_{\text{treatment}}(\text{size}, \epsilon_{\text{treatment}})$$

$$\text{success} = f_{\text{success}}(\text{treatment}, \text{size}, \epsilon_{\text{success}})$$

In [13]:

```
(define charig1986
  (model
    {:size []
     :treatment [:size]
     :success [:treatment :size]}))
```

Out[13]:



Identify

In [14]:

```
(define f
  (identify charig1986
    (data [:treatment :success :size])
    (q [:success] :do {:treatment "surgery"})))
```

Out[14]:

$$\left[\sum_{\text{size}} P(\text{size})P(\text{success} \mid \text{size}, \text{treatment})\right]$$

$$\text{where: treatment} = \text{"surgery"}$$

In [15]:

```
(identify charig1986
  (data [:treatment :success])
  (q [:success] :do {:treatment "surgery"}))
```

Out[15]:

```
#whittemore.core.Fail{:cause #{{:hedge #whittemore.core.Model{:pa {:treatment
#{}, :success #{:treatment}}, :bi #{#{:treatment :success}}, :s #{:succes
s}}}}}
```

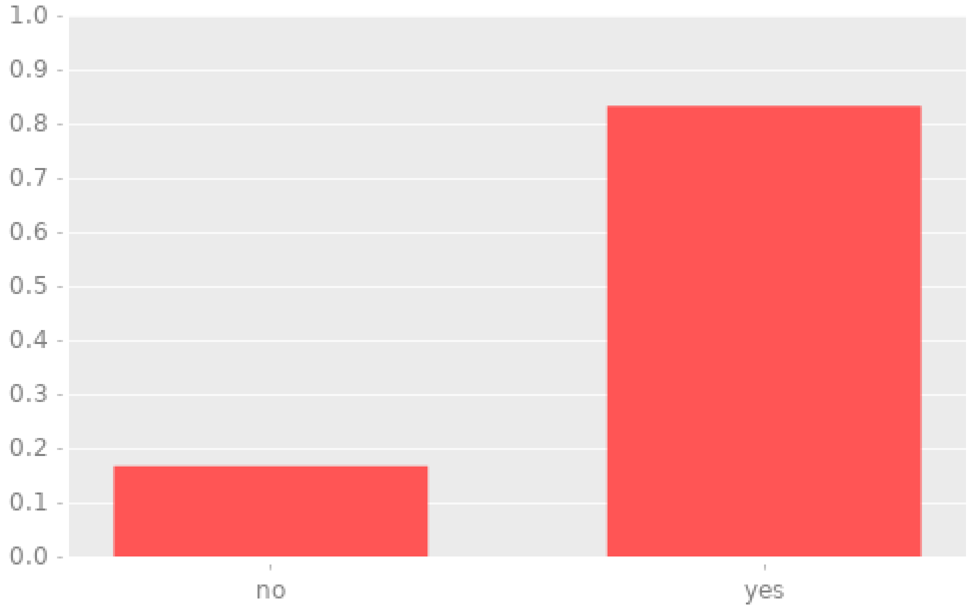
Estimate

In [16]:
```
(estimate kidney-distribution f)
```

Out[16]:
```
#whittemore.core.Categorical{:pmf {{:success "yes"} 0.8325462173856037, {:succ
ess "no"} 0.16745378261439622}}
```

In [17]:
```
(plot-univariate (estimate kidney-distribution f))
```

Out[17]:



Problem: $\boldsymbol{P}()$ notation is overloaded

- $P(Y = y \mid X = x)$; real number in the range [0, 1]
- $P(y \mid X = x)$; conditional *distribution* of Y
- $P(y \mid x)$; function from domain of X to conditional distributions of Y

Solution: syntactic sugar

In [18]:
```
(infer
  charig1986
  kidney-distribution
  (q {:success "yes"} :do {:treatment "surgery"}))
```

Out[18]: 0.8325462173856037

In [19]:
```
(infer
  charig1986
  kidney-distribution
  (q {:success "yes"} :do {:treatment "nephrolithotomy"}))
```

Out[19]: 0.778875

Infer and plot

In [20]:

```
(def associational-plot
  (plot-p-map
    {"P(success | nephro...)"
     (estimate kidney-distribution
       (q {:success "yes"} :given {:treatment "nephrolithotomy"}))

     "P(success | surgery)"
     (estimate kidney-distribution
       (q {:success "yes"} :given {:treatment "surgery"}))}))

(def interventional-plot
  (plot-p-map
    {"P(success | do(nephro...))"
     (infer charig1986 kidney-distribution
       (q {:success "yes"} :do {:treatment "nephrolithotomy"}))

     "P(success | do(surgery))"
     (infer charig1986 kidney-distribution
       (q {:success "yes"} :do {:treatment "surgery"}))}))
```

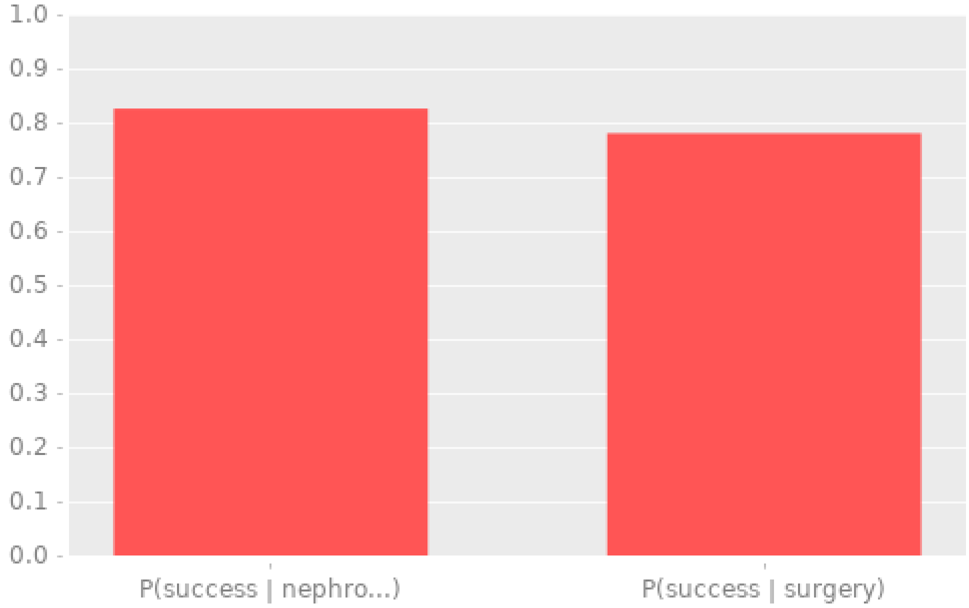
Out[20]:

```
#'user/interventional-plot
```

In [21]:

```
associational-plot
```

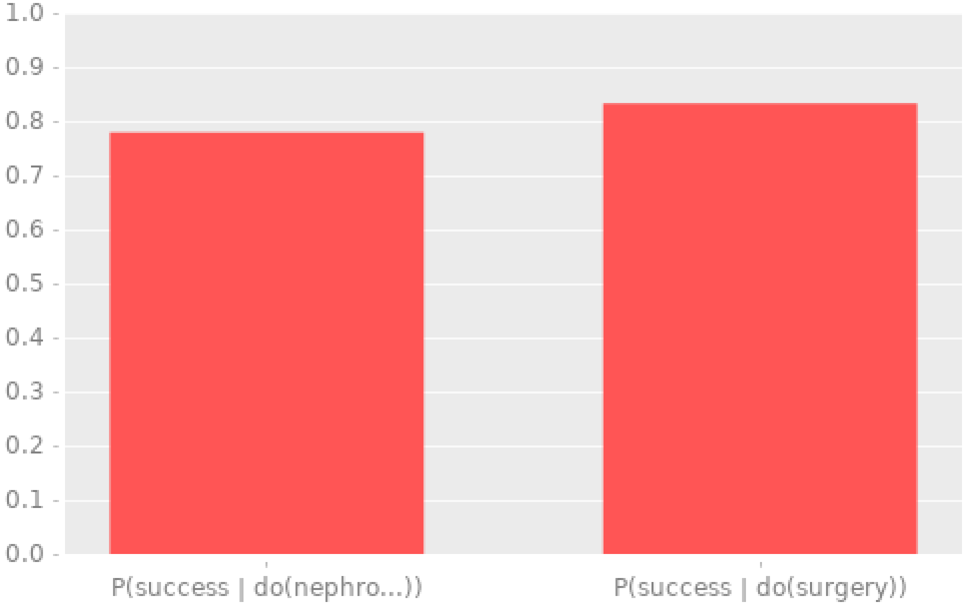
Out[21]:



In [22]:

```
interventional-plot
```

Out[22]:



Nonstandard adjustments

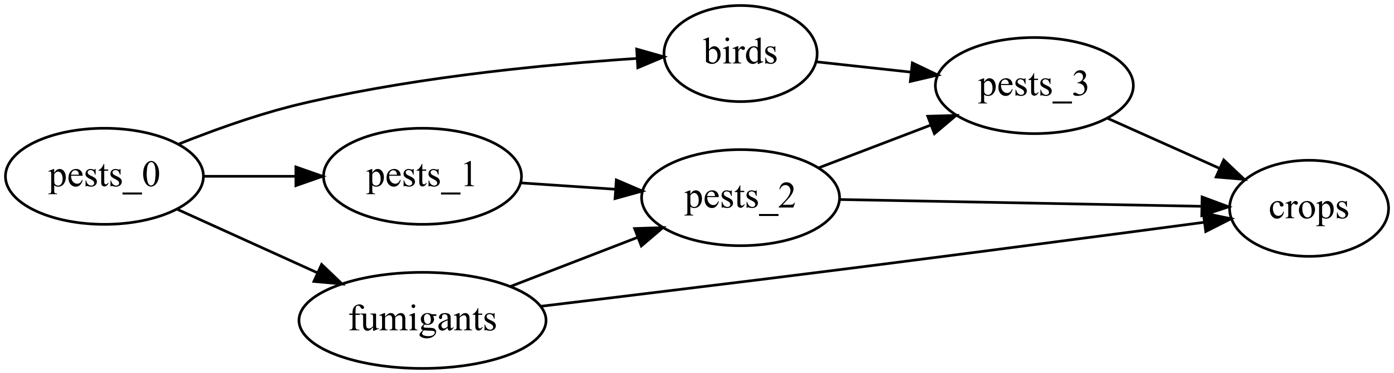
> *This article provides the most systematic account to date of the problems with and solutions to a recurring problem in experimental political science: conditioning on posttreatment variables.*
>
> *...we recommend avoiding selecting on or controlling for posttreatment covariates.*

"How Conditioning on Posttreatment Variables Can Ruin Your Experiment and What to Do about It" (Montgomery et al. 2018)

In [23]:

```
(define wainer1989
  (model
    {:pests_0 []
     :birds [:pests_0]
     :pests_1 [:pests_0]
     :fumigants [:pests_0]
     :pests_2 [:pests_1 :fumigants]
     :pests_3 [:pests_2 :birds]
     :crops [:fumigants :pests_2 :pests_3]}))
```

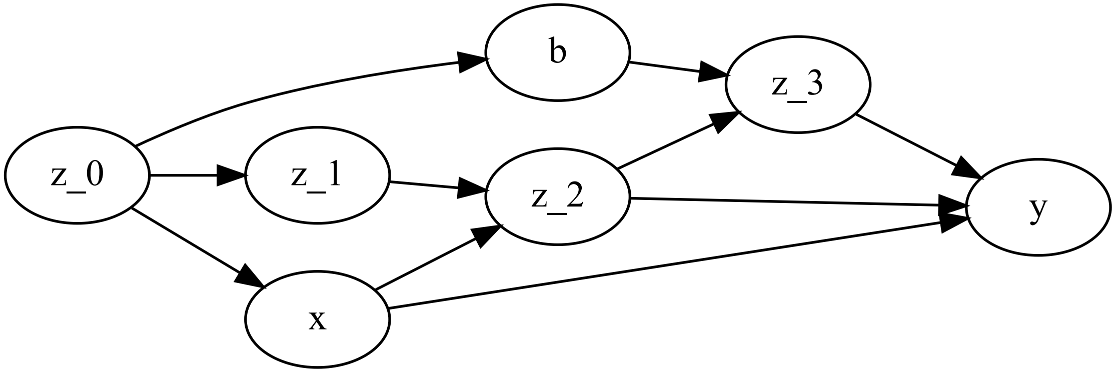
Out[23]:



In [25]:

```
(define wainer-short
  (model
    {:z_0 []
     :b [:z_0]
     :z_1 [:z_0]
     :x [:z_0]
     :z_2 [:z_1 :x]
     :z_3 [:z_2 :b]
     :y [:x :z_2 :z_3]}))
```

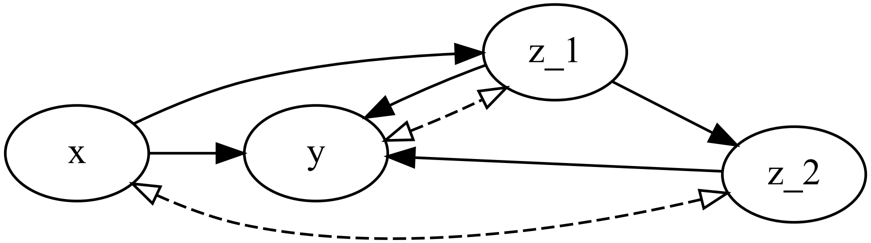
Out[25]:



In [27]:

```
(define concomitant-example
  "Figure 3.8 (f) from (Shpitser 2008)"
  (model
    {:y [:x :z_1 :z_2]
     :z_2 [:z_1]
     :z_1 [:x]
     :x []}
    #{:y :z_1}
    #{:x :z_2}))
```

Out[27]:



In [28]:

```
(identify
  concomitant-example
  (data [:x :y :z_1 :z_2])
  (q [:y] :do [:x]))
```

Out[28]:

$$\left[\sum_{z_1,z_2}\left[\sum_{x} P(x)P(z_2 \mid x, z_1)\right] P(z_1 \mid x)P(y \mid x, z_1, z_2)\right]$$

where: (unbound)

Distribution protocol

- (estimate *this formula*)
- (measure *this event*)
- (signature *this*)

User extensible; potential for integration with probabilistic programming

"Nanopass" simplification

- Tikka and Karvanen modify the ID algorithm to simplify formulas
- Whittemore seperates identification and simplification steps
- "Pattern matching" rules to simplify formulas
    - Marginalize rule $\sum_x P(x, y) \to P(y)$
    - Conditional rule $\frac{P(x,y)}{P(y)} \to P(x \mid y)$
    - Not *currently* user extensible

Install (Ubuntu)

```
$ sudo apt install leiningen
$ pip3 install jupyter

$ lein new whittmore demo
$ cd demo
$ lein jupyter notebook
```

Source

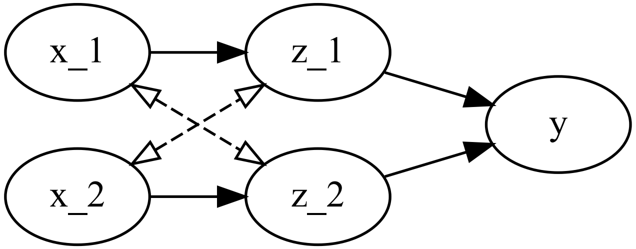
github.com/jtcbrule/whittemore

Questions?

In [29]:

```
(define butterfly
  (model
    {:x_1 []
     :z_1 [:x_1]
     :y [:z_1 :z_2]
     :x_2 []
     :z_2 [:x_2]}
    #{:x_1 :z_2}
    #{:z_1 :x_2}))
```

Out[29]:



In [30]:

```
(identify butterfly (q [:y] :do [:x_1 :x_2]))
```

Out[30]:

$$\left[\sum_{z_1,z_2} P(y \mid x_1, x_2, z_1, z_2) P(z_2 \mid x_2) P(z_1 \mid x_1)\right]$$

where: (unbound)