SOCI 40258

Causal Mediation Analysis

Week 2: Graphical Causal Models

Outline

- Elements of directed acyclic graphs (DAGs)
- Statistical association, d-separation, and d-connection
- DAGs as nonparametric structural equation models (NPSEMs)
- DAGs and the potential outcomes framework
- Markov and truncated Markov factorizations
- Identification with DAGs

Overview

- Overview
  - DAGs are visual representations of causal models that are governed by simple rules that map the model onto statements about observed probability distributions

- Utility
  - DAGs are useful in *identification analyses*, which seek to determine whether a causal effect or set of causal effects could be computed with data from an infinite sample

  - Second, DAGs are useful for guiding the selection of an appropriate estimator for the causal effect or set of causal effects of interest

  - Third, DAGs can also be used to derive a comprehensive set of testable hypotheses about observed probability distributions given the causal model

Elements of DAGs

- Variables (or nodes), typically denoted by letters

- Arrows (or directed edges), which represent direct causal effects
  - An arrow from one variable to another signifies that the variable at the origin of the arrow causes the variable at the terminus

- Missing arrows, which represent the absence of a direct causal effect
  - The absence of an arrow from one variable to another signifies that there is no direct causal effect between the two variables for every member of the population

Elements of DAGs

- Paths
  - Sequences of adjacent arrows that traverse any given variable at most once
- Causal (or directed) paths
  - Paths in which all the adjacent arrows point in the same direction
- Non-causal paths
  - Paths in which all arrows do not point in the same direction

Elements of DAGs

- Children and descendants
  - Variables directly caused by another variable are called its *children*
  - All variables directly or indirectly caused by another variable are called its *descendants*

- Parents and ancestors
  - Variables that directly cause another variable are called its *parents*
  - Variables that directly or indirectly cause another variable are called its *ancestors*

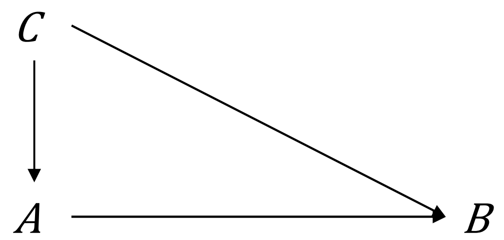
- Colliders
  - A variable on a path with two arrows pointing into it—that is, a variable on a path with at least two parents

Excluded elements of DAGs

- DAGs may not contain cycles (i.e., they are *acyclic)*
  - A cycle is a directed path that begins and terminates at the same variable
  - In a DAG, no directed paths emanating from a causal variable may also terminate at the same causal variable, or in other words, no variable may be its own descendent
  - This restriction precludes the possibility of simultaneous causality and encodes the basic condition that causes must temporally precede outcomes
  - Note, however, that DAGs can accommodate many different forms of causal feedback by carefully specifying the temporal sequence of variables or measurements of the same variable

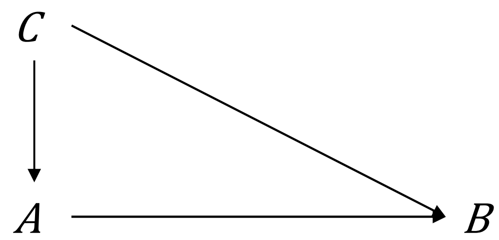
An example of a DAG

- A and B are children of C, and C is a parent of A and B
- B is also a child of A, and A is also a parent of B



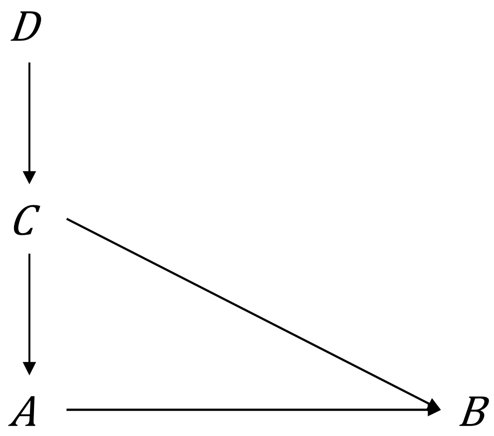
An example of a DAG

- $C \rightarrow A$, $C \rightarrow B$, $A \rightarrow B$, and $C \rightarrow A \rightarrow B$ are causal, or directed, paths
- $C \rightarrow A \rightarrow B$ is a causal path called a *chain*
- $A \leftarrow C \rightarrow B$ is a non-causal path called a *fork*
- $A \rightarrow B \leftarrow C$ is a non-causal path called an *inverted fork*
- B is a *collider* on the path $A \rightarrow B \leftarrow C$



Another example of a DAG

- A, B, and C are descendants of D
- D is an ancestor of A, B, and C
- $D \rightarrow C$, $D \rightarrow C \rightarrow A$, $D \rightarrow C \rightarrow B$, and $D \rightarrow C \rightarrow A \rightarrow B$ are new causal paths
- $D \rightarrow C \rightarrow A$, $D \rightarrow C \rightarrow B$, and $D \rightarrow C \rightarrow A \rightarrow B$ are all chains



Three sources of association

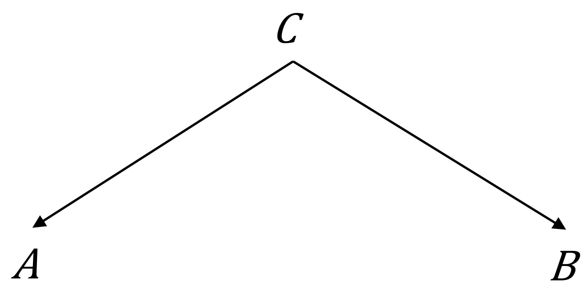
- Statistical association and independence can be easily derived from the causal models encoded by DAGs
- The three basic causal structures that generate statistical associations are:
  - Causation, which involves causal paths and chains
  - Confounding, which involves forks
  - Endogenous selection (AKA collider stratification), which involves inverted forks

Causation

$$A \longrightarrow C \longrightarrow B$$

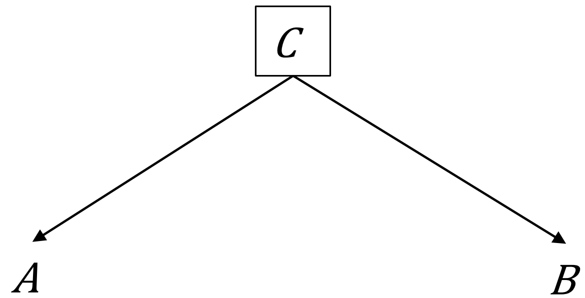
- $P(A, B) \neq P(A)P(B)$ because A causes C and C causes B
- The path $A \rightarrow C \rightarrow B$ is a *chain*
- Mediation is defined in terms of chains, and mediation analysis is the study of causal chains

Confounding



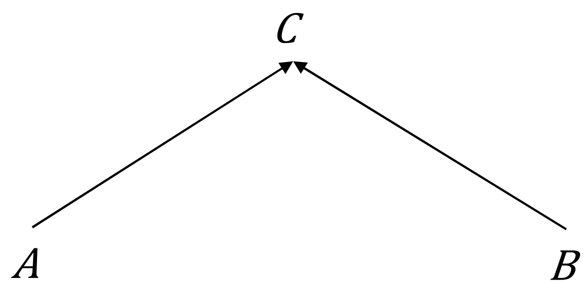
- $P(A,B) \neq P(A)P(B)$ because C causes A and C causes B
- The path $A \leftarrow C \rightarrow B$ is a *fork*

Confounding



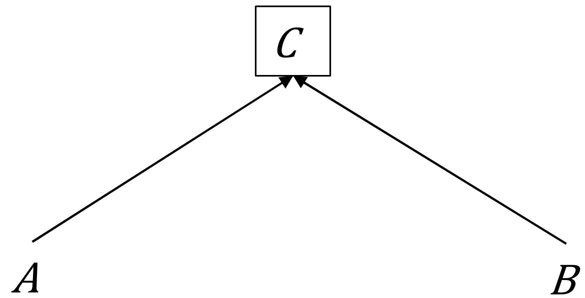
- $P(A, B|C) = P(A|C)P(B|C)$, that is, A and B are independent given C
- A box drawn around a variable denotes conditioning (i.e., perfect, or complete, stratification by levels of that variable)

Endogenous selection



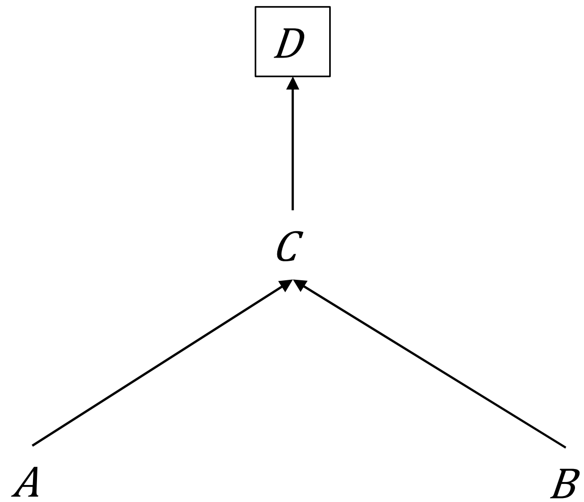
- $P(A, B) = P(A)P(B)$, that is, A and B are marginally, or unconditionally, independent
- The path $A \rightarrow C \leftarrow B$ is an *inverted fork*, and C is a collider

Endogenous selection



- $P(A, B|C) \neq P(A|C)P(B|C)$, that is, A and B are not independent given C
- The path $A \rightarrow C \leftarrow B$ is an *inverted fork*, and C is a collider on which we've conditioned
- Endogenous selection is also commonly referred to as "collider stratification"

Endogenous selection



- $P(A, B|D) \neq P(A|D)P(B|D)$, that is, A and B are not independent given D, a descendent of the collider C
- Conditioning on a descendant of a collider has the same associational consequences as conditioning on the collider itself

D-separation

- The concept of "d-separation" consolidates these different sources of statistical association into a simple graphical rule that determines whether a path in a DAG generates a statistical association between two variables

- A path between two variables, A and B, is d-separated, or "blocked," if:
  - The path contains a non-collider that has been conditioned on
  - The path contains a collider, possibly with descendants, that have not been conditioned on

- If two variables, A and B, are d-separated along all paths by conditioning on a variable or set of variables C, which may be empty, then $P(A, B|C) = P(A|C)P(B|C)$, that is, A and B are independent given C

D-connection

- A path between two variables, A and B, is d-connected, or "unblocked," if it is not d-separated
  - That is, the path contains a non-collider that has not been conditioned on, or the path contains a collider, or descendants of this collider, that have been conditioned on

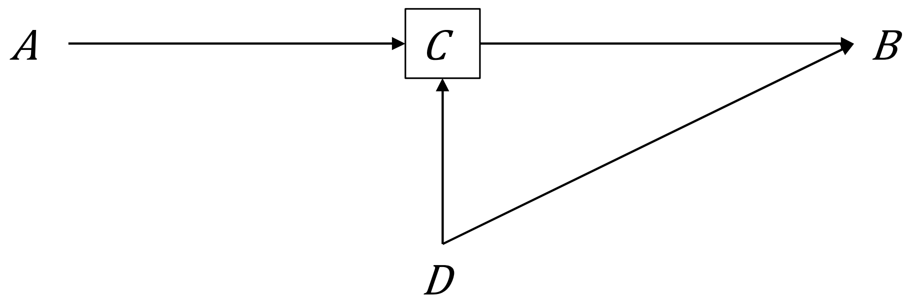
- If two variables, A and B, are not d-separated by C along all paths, and thus are d-connected, then $P(A, B|C) \neq P(A|C)P(B|C)$, that is, A and B are not independent given C

D-separation

$$A \longrightarrow \boxed{C} \longrightarrow B$$

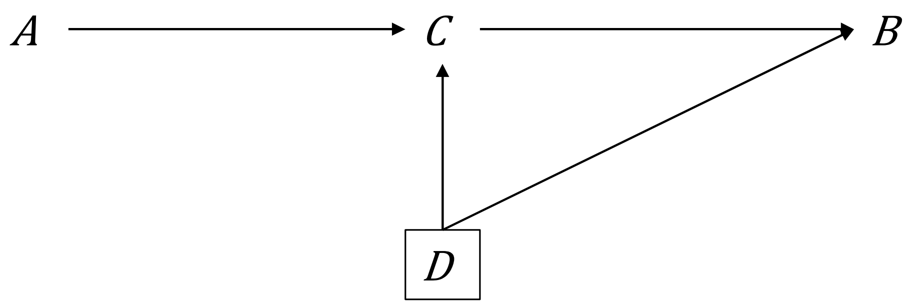
- Are A and B d-separated?
- Recall that a box drawn around a variable denotes conditioning

D-separation



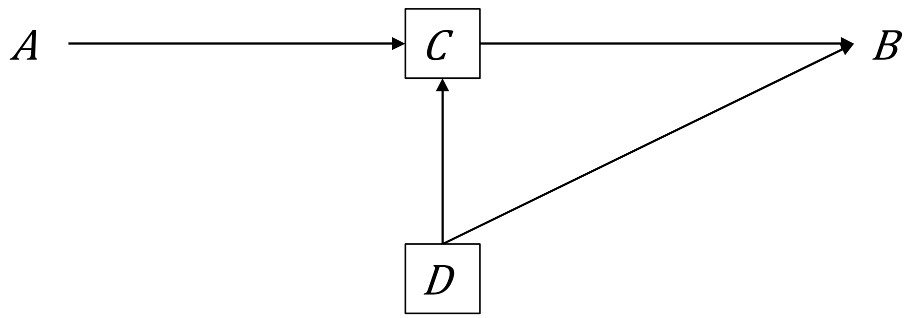
- What about now – are A and B d-separated?

D-separation



- Are C and B d-separated?

D-separation

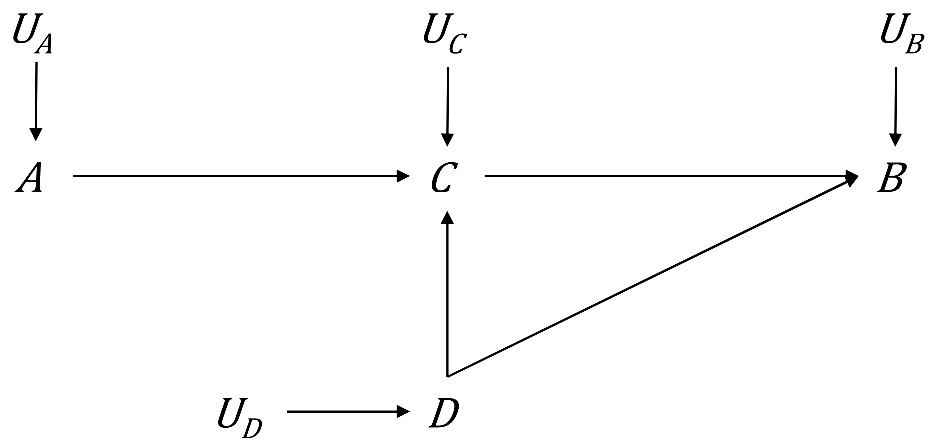


- Are A and B d-separated?

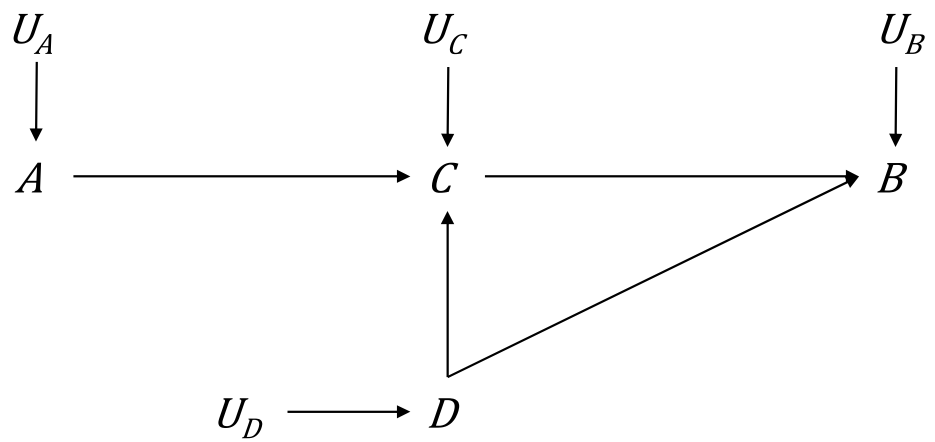
DAGs as NPSEMs

- DAGs can be interpreted as non-parametric structural equation models (NPSEMs)
  - "Nonparametric" here refers to the absence of any assumptions about the probability distributions of the variables or the functional form of the structural equations

- Consider the following DAG…



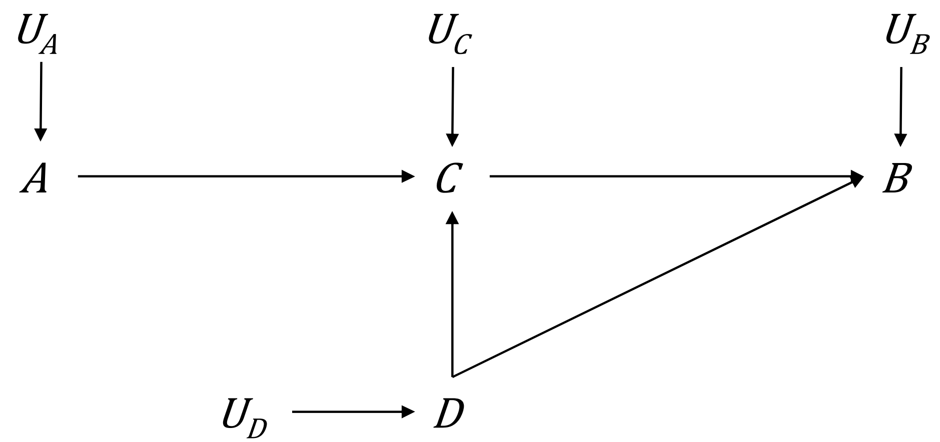
DAGs and NPSEMs



- …which implies the following nonparametric structural equations:

$$A := f_A(U_A),\ D := f_D(U_D),\ C := f_C(A, D, U_C), \text{ and } B := f_B(C, D, U_B)$$

DAGs and NPSEMs



- …where the U terms are random disturbances that are typically suppressed from the DAG for visual simplicity

DAGs as NPSEMs

- What does $C := f_C(A, D, U_C)$, for example, actually denote?
  - It means that C takes $A, D,$ and U_C as causal inputs—that is, the values of C are assigned, or realized, as a function of the previously assigned or realized values of $A, D,$ and U_C
  - It does not imply any specific distribution for C or its inputs, nor does it imply any specific functional form for $f_C(A, D, U_C)$

- In other words, $C := f_C(A, D, U_C)$ is consistent with a variety of different parametric specifications, including, for example, the conventional linear regression model

$$C := f_C\ (A, D, U_C) = \beta_0 + \beta_1 A + \beta_2 D + U_C \qquad U_C \sim N(0, \sigma^2)$$

but also much more complicated models

DAGs and potential outcomes

- DAGs and the potential outcome framework are complimentary frameworks, even though DAGs do not explicitly incorporate potential outcomes
- Within the graphical framework, the "$do(\cdot)$ operator" provides an analog to potential outcomes for conceptualizing and defining causal effects
- The $do(\cdot)$ operator is a mathematical device that defines an "ideal experiment" or an "atomic intervention" where all units in a target population are assigned to different causal states

The do-operator

- In this framework, then, the average causal effect of a treatment, A, on an outcome, Y, is defined as:

$$ATE = E\big(Y|do(A = 1)\big) - E\big(Y|do(A = 0)\big)$$

- The ATE is here defined as the difference in the mean of the outcome if all individuals in the target population were assigned to treatment rather than not

- In this sense, $E\big(Y(1) - Y(0)\big)$ and $E\big(Y|do(A = 1)\big) - E\big(Y|do(A = 0)\big)$ are equivalent estimands that have merely been defined using different notation

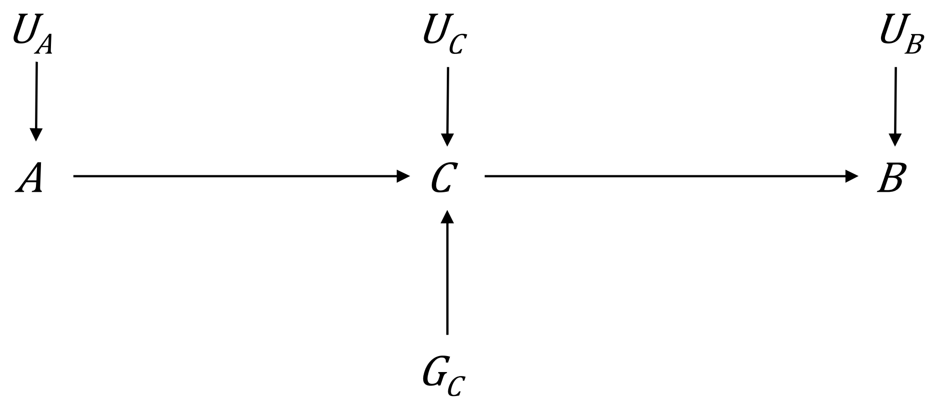
DAGs and the do-operator

- The $do(\cdot)$ operator is not visible in standard representations of DAGs, but additional, related graphs are available that demonstrate the connection explicitly

- Specifically, the $do(\cdot)$ operator is made explicit in *augmented graphs* through the use of "forcing" variables that prescribe a specific intervention, or lack thereof, on the variable or variables of interest

- In practice, forcing variables are rarely depicted in DAGs, and augmented graphs are rarely used in practice, but it is important to keep in mind that this framework presumes such variables are present "in the background"

Augmented graphs



- G_C is a "forcing" variable that takes on three possible values: $do(C = 1)$, $do(C = 0)$, or $idle$

Augmented graphs

$$\begin{tikzcd} U_A \arrow[d] & U_C \arrow[d] & U_B \arrow[d] \\ A \arrow[r] & C \arrow[r] & B \\ & G_C \arrow[u] & \end{tikzcd}$$

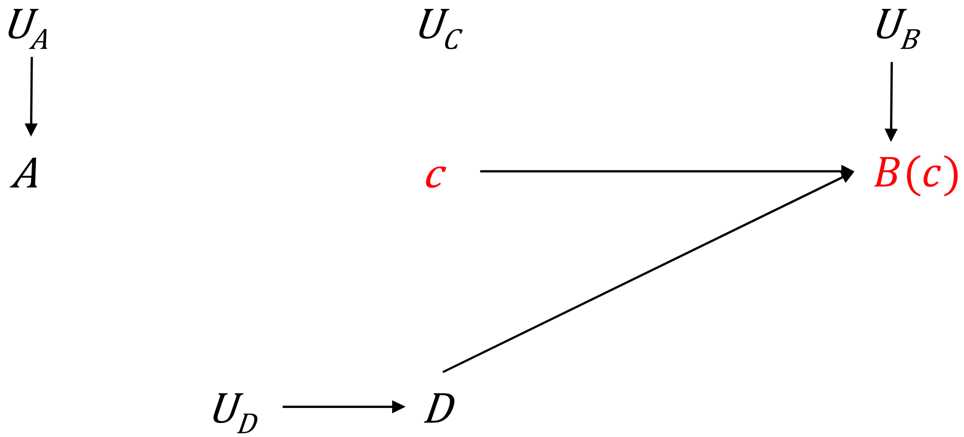
- And the NPSEM for C in the augmented graph is defined as:

$$C = \begin{cases} 1 \text{ if } G_C = do(C = 1) \\ 0 \text{ if } G_C = do(C = 0) \\ f_C(A, U_c) \text{ if } G_C = idle \end{cases}$$

Mutilated graphs

- Interventions – the result of a $do(\cdot)$ operation – are represented in DAGs by "mutilating" them
- Mutilating a DAG involves removing all incoming edges to the variable or variables subject to intervention, and then setting these nodes to prescribed values
- Descendants of the variables intervened upon are then transformed into potential outcomes—that is, the value of the descendant that would occur if its ancestor were set to some specific value

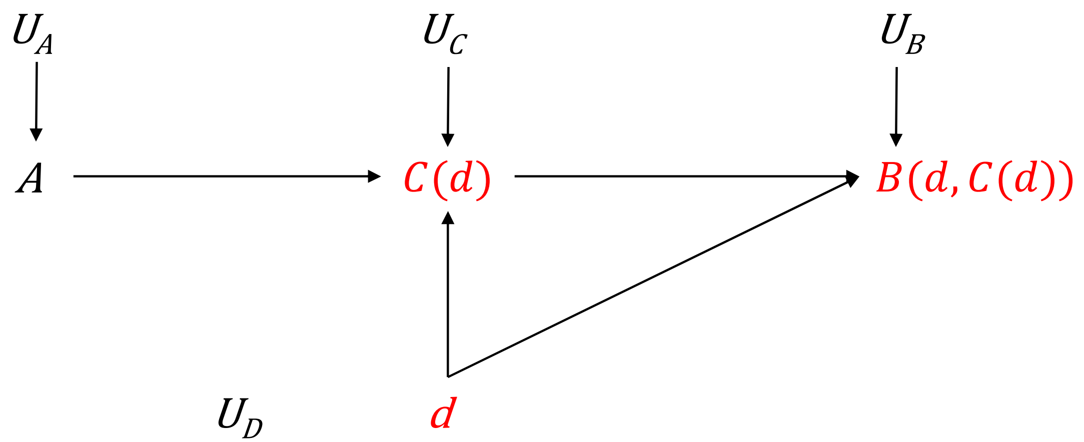
Mutilated graph for $do(C = c)$



- …which implies the following mutilated structural equations:

$$A := f_A(U_A), D := f_D(U_D), C \coloneqq c, \text{ and } B(c) := f_B(c, D, U_B)$$

Mutilated graph for $do(D = d)$



- …which implies the following nonparametric structural equations:

$$A := f_A(U_A),\, D \coloneqq d,\, C(d) := f_C(A, d, U_C), \text{ and } B\big(d, C(d)\big) := f_B(d, C(d), U_B)$$

The Markov factorization

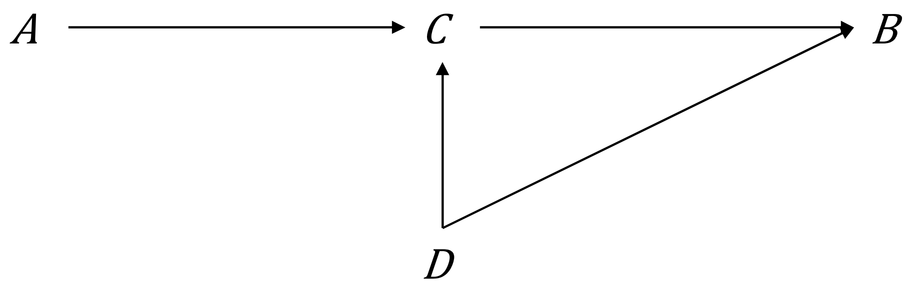
- The dependence and independence relations implied by a DAG imply a particular factorization of the joint distribution for the observed data, known as the Markov factorization

- In general, the joint distribution of any set of k variables, ordered arbitrarily and denoted by $\mathbf{X} = \{X_1, X_2, \dots, X_k\}$, can be factorized as follows using the product rule of joint probability:

$$P(x_1, x_2, \dots, x_k) = P(x_1)P(x_2|x_1) \dots P(x_k|x_1, \dots, x_{k-1})$$

- The Markov factorization of the joint distribution decomposes it into a product of conditional probabilities, with each depending only on the parents of the variable in question

The Markov factorization



$$P(a,d,c,b) = P(a)P(d)P(c|a,d)P(b|c,d)$$
$$= \prod_{j=1}^{k} P\left(\mathrm{v}_j \middle| \mathbf{v}_j^p\right),$$

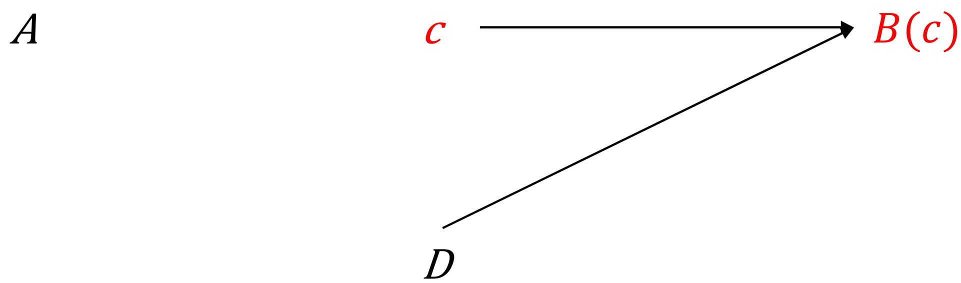
where $\mathrm{v}_j \in \{a,d,c,b\}$ and $\mathbf{v}_j^p$ denotes the parents of v_j

The truncated Markov factorization

- The dependence and independence relations given by a mutilated DAG imply a particular factorization of the interventional joint distribution, known as the truncated Markov factorization

- The truncated Markov factorization is obtained by setting the conditional probability for the variable subject to intervention equal to one and then appropriately conditioning the remaining probabilities on this intervention

The truncated Markov factorization for $do(C = c)$



$$P\big(a, d, c, b \mid do(C = c)\big) = P(a)P(d)(1)P(b \mid c, d)$$

$$= P(a)P(d)P(b \mid c, d)$$

Identification analysis with DAGs

- DAGs, and the concepts of d-separation and d-connection, are extremely useful for *identification analyses*
- *Identification analyses* seek to determine whether a causal effect or set of causal effects of interest could be computed with data from an infinite sample
  - A causal effect is said to be identified if it is possible to compute it with data from an infinite sample, and it is said to be unidentified if it cannot be computed
- Identification is distinct from estimation, which refers to computing estimates for an effect or set of effects from sample data

The back-door criterion

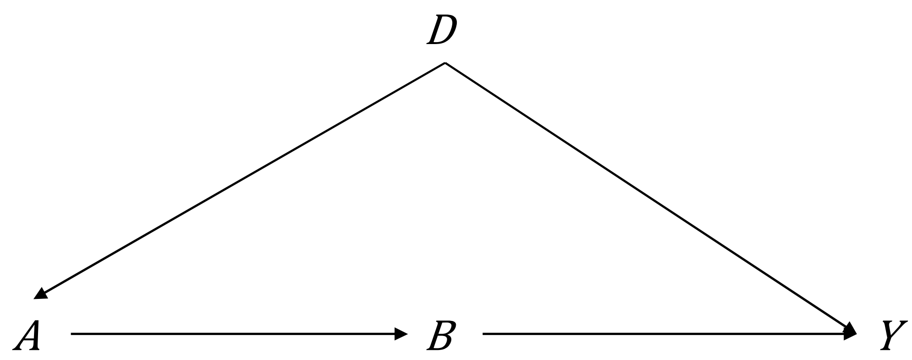
- The backdoor criterion is a set of rules that can be applied to a DAG to determine whether conditioning on a given set of variables will identify a causal effect of interest
- The ATE of an exposure on an outcome is identified if there exists a set of observed variables, C, which may be empty, such that:
  - No element of C is a descendent of treatment
  - Conditioning on C blocks all *back-door paths* from treatment to the outcome
- A *back-door path* is a path between treatment and the outcome that begins with an arrow pointing into treatment

The back-door criterion

- The back-door criterion can be implemented as follows:
  - Write down all back-door paths from treatment to the outcome
  - Determine which paths are blocked (d-separated) or unblocked (d-connected)
  - Search for a set of conditioning variables that will block all unblocked back-door paths
  - Inspect the patterns of descent in the graph to verify that none of the variables in the conditioning set are descendants of treatment

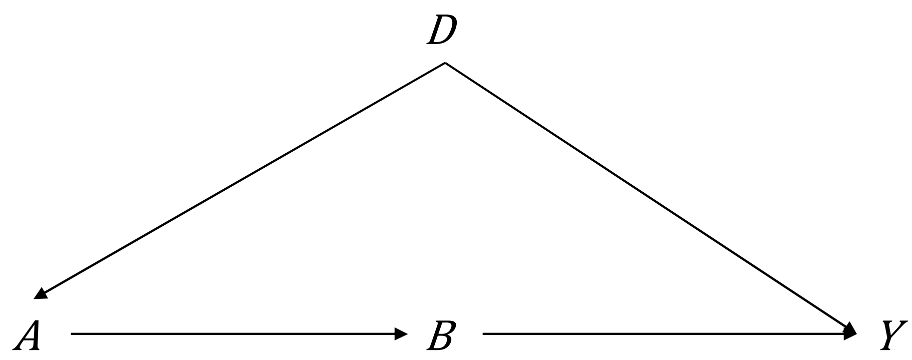
The back-door criterion

- Assume all variables are observed
- Is the ATE of A on B identified by the back-door criterion?
  - If so, what variables are in the conditioning set?
- What about the ATE of A on Y – is this effect identified by the back-door criterion?
  - If so, what variables are in the conditioning set?



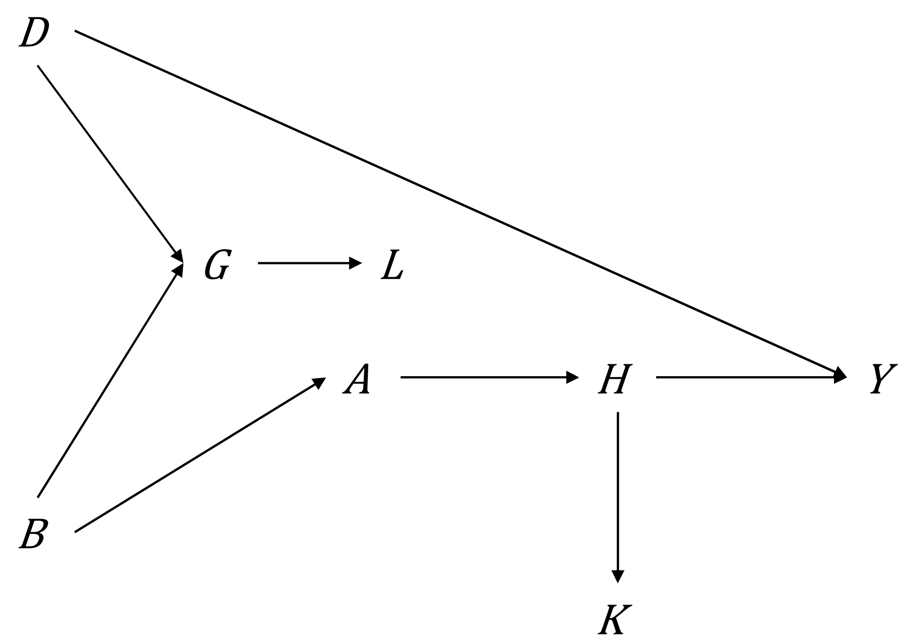
The back-door criterion

- How about the ATE of B on Y – is it identified by the back-door criterion?
  - If so, what variables are in the conditioning set?
  - Are there multiple sets of variables on which we could condition to satisfy the back-door criterion here?



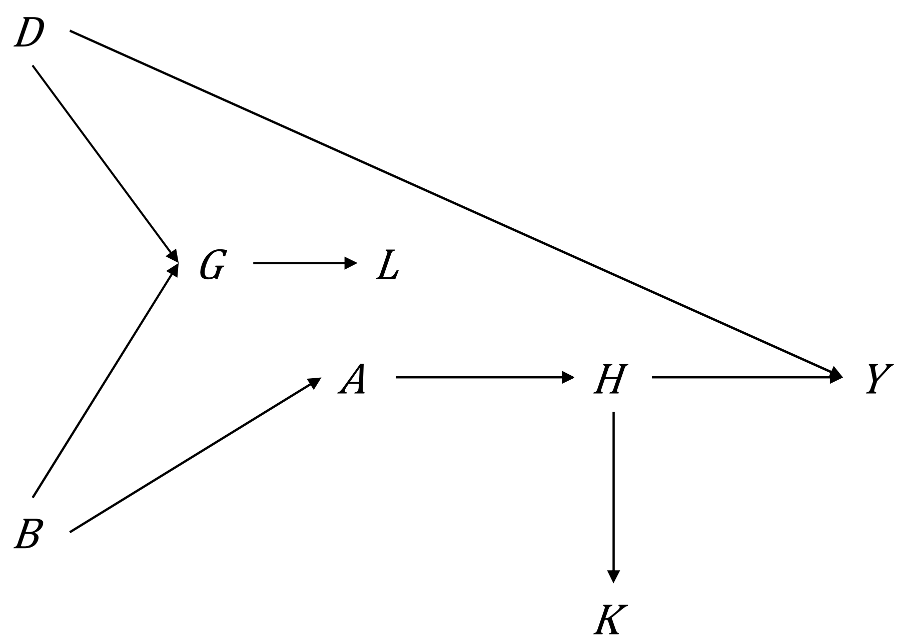
The back-door criterion

- Now let's consider a more complex graph, assuming again that all variables are observed
- Is the ATE of A on Y identified by the back-door criterion?
  - If so, what variables are in the conditioning set?
  - Are there multiple sets of variables on which we could condition to satisfy the backdoor criteria here?



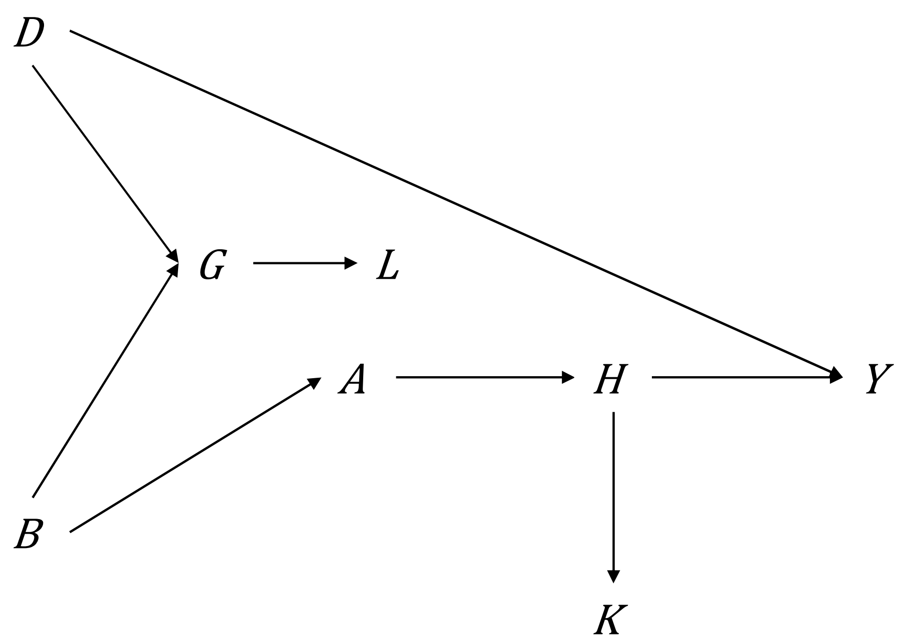
The back-door criterion

- Is the ATE of A on Y identified if we include H in the conditioning set?
  - Why or why not?
- Is the ATE of A on Y identified if we include K in the conditioning set?
  - Why or why not?



The back-door criterion

- Is the ATE of A on Y identified if the conditioning set includes only G?
  - Why or why not?
- Is the ATE of A on Y identified if the conditioning set includes only L?
  - Why or why not?



A "back door" into potential outcomes…

- The back-door criterion provides a direct connection from the graphical to the potential outcomes framework

- If the back-door criterion for the ATE of an exposure A on an outcome Y is satisfied by conditioning on a set of variables C, then the potential outcomes are independent of treatment conditional on C

- In other words, $\{Y(0), Y(1)\} \perp A|C$ when C satisfies the back-door criterion, and when C is empty, then the stronger marginal independence condition, $\{Y(0), Y(1)\} \perp A$, is satisfied

…and the NP identification formula

- When the back-door criterion is satisfied by a set of variables C, then the ATE of a binary exposure A on an outcome Y is equal to

$$
\begin{aligned}
ATE &= E(Y|do(A=1)) - E(Y|do(A=0)) \\
&= E\big(Y(1)\big) - E\big(Y(0)\big) \\
&= \textstyle\sum_c \big(E(Y|A=1, C=c) - E(Y|A=0, C=c)\big) P(C=c)
\end{aligned}
$$

Example: NLSY79

- 1979 National Longitudinal Study of Youth
  - Exposure (D)
    - sample member attended college before age 22
  - Outcome (Y):
    - standardized scores on the CES-D at age 40
  - Covariates (C):
    - Race, gender, parental education, occupation, income, household size, and AFQT scores
  - A potential mediator (M)
    - unemployment between age 35-40

Example: NLSY79

- Draw a DAG that best represents the causal relations among these variables, based on theory and prior knowledge

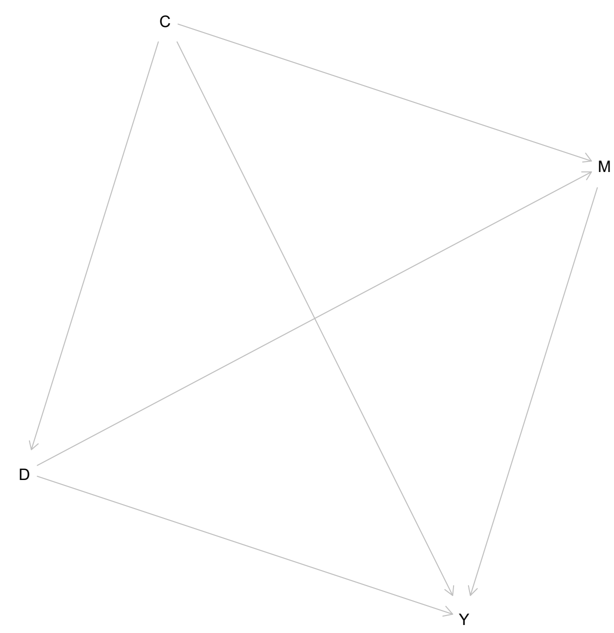
```
### wk 2 nlsy tutorial ###
rm(list=ls())

# load/install libraries #
packages<-c("dagitty")
install.packages(packages)

for (package.i in packages) {
    suppressPackageStartupMessages(library(package.i, character.only=TRUE))
    }

# specify DAG #
nlsyDAG <- dagitty( "dag {
    C -> D -> M -> Y
    C -> M
    C -> Y
    D -> Y
}")

plot(graphLayout(nlsyDAG))
```



Example: NLSY79

- What are the dependence/independence implications of the assumed DAG?

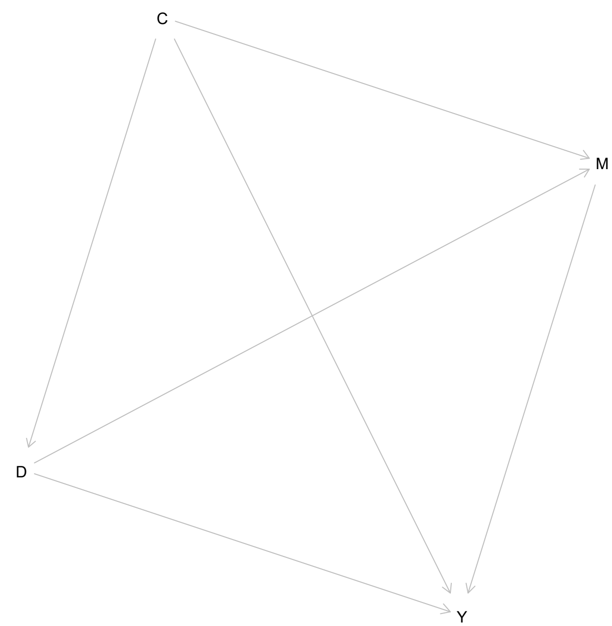
```r

# list dependence/independence implications #
impliedConditionalIndependencies(nlsyDAG)

dconnected(nlsyDAG, "D", "Y")
dconnected(nlsyDAG, "D", "Y", "M")
dconnected(nlsyDAG, "D", "Y", c("C","M"))
dconnected(nlsyDAG, "M", "Y", c("D","C"))

```

```
> # list dependence/independence implications #
> impliedConditionalIndependencies(nlsyDAG)
>
> dconnected(nlsyDAG, "D", "Y")
[1] TRUE
> dconnected(nlsyDAG, "D", "Y", "M")
[1] TRUE
> dconnected(nlsyDAG, "D", "Y", c("C","M"))
[1] TRUE
> dconnected(nlsyDAG, "M", "Y", c("D","C"))
[1] TRUE
>
```



Example: NLSY79

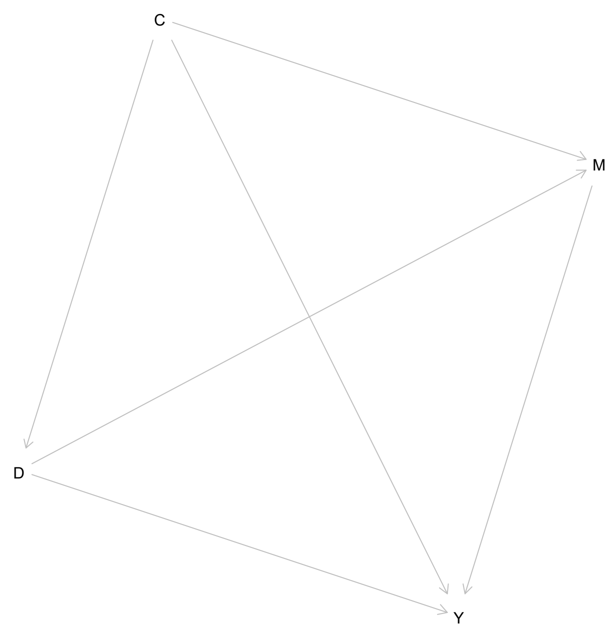
- What are the adjustment sets necessary to nonparametrically identify the ATE of college attendance on depression? What about the ATE of unemployment on depression?

```
# list adjustment sets for ATE of D on Y #
adjustmentSets(nlsyDAG, "D", "Y")

# list adjustment sets for ATE of M on Y #
adjustmentSets(nlsyDAG, "M", "Y")

```

```
> # list adjustment sets for ATE of D on Y #
> adjustmentSets(nlsyDAG, "D", "Y")
{ C }
>
> # list adjustment sets for ATE of M on Y #
> adjustmentSets(nlsyDAG, "M", "Y")
{ C, D }
>
```

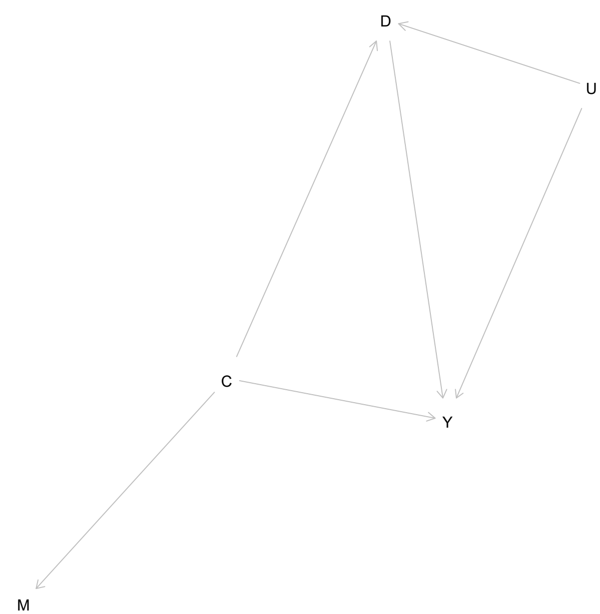


Example: NLSY79

- Re-draw the DAG without mediation (i.e., without a $D \rightarrow M \rightarrow Y$ chain) but with an unobserved confounder for the $D \rightarrow Y$ relationship

```
# repeat with unobserved D-Y confounder and no mediation
nlsyDAGwithU <- dagitty( "dag {
    C -> D
    C -> M
    C -> Y
    D -> Y
    U -> D
    U -> Y
}")

plot(graphLayout(nlsyDAGwithU))
```



Example: NLSY79

- What are the implications for the statistical associations among observed variables and identification of causal effects?

```r
print(impliedConditionalIndependencies(nlsyDAGwithU))

dconnected(nlsyDAGwithU, "D", "Y")
dconnected(nlsyDAGwithU, "D", "Y", "M")
dconnected(nlsyDAGwithU, "D", "Y", c("C","M"))
dconnected(nlsyDAGwithU, "M", "Y", c("D","C"))

print(adjustmentSets(nlsyDAGwithU, "D", "Y"))

print(adjustmentSets(nlsyDAGwithU, "M", "Y"))
```

```
> print(impliedConditionalIndependencies(nlsyDAGwithU))
C _||_ U
D _||_ M | C
M _||_ U
M _||_ Y | C
>
> dconnected(nlsyDAGwithU, "D", "Y")
[1] TRUE
> dconnected(nlsyDAGwithU, "D", "Y", "M")
[1] TRUE
> dconnected(nlsyDAGwithU, "D", "Y", c("C","M"))
[1] TRUE
> dconnected(nlsyDAGwithU, "M", "Y", c("D","C"))
[1] FALSE
>
> print(adjustmentSets(nlsyDAGwithU, "D", "Y"))
{ C, U }
>
> print(adjustmentSets(nlsyDAGwithU, "M", "Y"))
{ C }
>
```

DAGitty web-based GUI

- https://www.dagitty.net/dags.html