

EPI13 - Seminários de Doutorado III

Oficina de DAGs

Rodrigo Citton P. dos Reis
citton.padilha@ufrgs.br

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DEPARTAMENTO DE ESTATÍSTICA E PROGRAMA DE PÓS-GRADUAÇÃO EM EPIDEMIOLOGIA

Vitória da Conquista, 2019



Relebrando

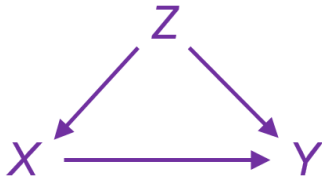
Como duas variáveis podem estar associadas?



- ▶ X e Y serão **associadas na população** se:
 - ▶ X causa Y.
 - ▶ Y causa X.
 - ▶ existe uma Z que é causa comum de X e Y.
- ▶ X e Y serão **associadas em subpopulações definadas por Z** se Z é um **efeito** de X e Y.

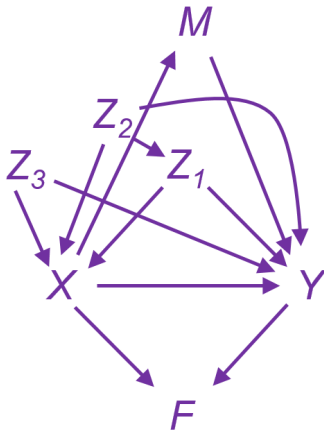
Como duas variáveis podem estar associadas?

- O que podemos concluir do diagrama abaixo?



Como duas variáveis podem estar associadas?

- E neste outro caso? O que podemos concluir?

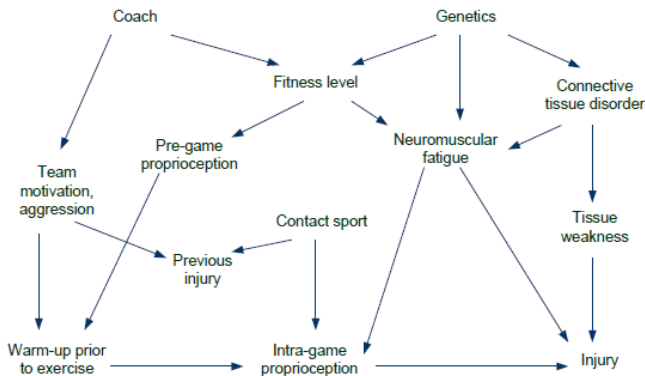


Grafos acíclicos dirigidos

- ▶ Os **grafos acíclicos dirigidos** (DAGs), ou **diagramas causais**, oferecem uma linguagem para especificação de relações (causais) entre as variáveis do quadro conceitual.
- ▶ **Critério back-door**: (1) verifica a existência de **viés de confusão**; (2) em caso afirmativo, verifica a existência de um conjunto de variáveis **suficiente para o controle** (ajuste) do viés de confusão.

Ferramentas computacionais para a construção de DAGs

Um exemplo



- A análise de DAGs pode ser tediosa na prática, e se presta bem à automatização por um programa de computador.

DAGitty

Welcome to DAGitty!

Launch



[Launch DAGitty
online in your
browser](#)

Download



[Download
DAGitty's source
for offline use](#)

Learn



[Learn more about
DAGs and
DAGitty](#)

Code



The R package
"dagitty" is
available on
[CRAN](#) or [github](#)

DAGitty

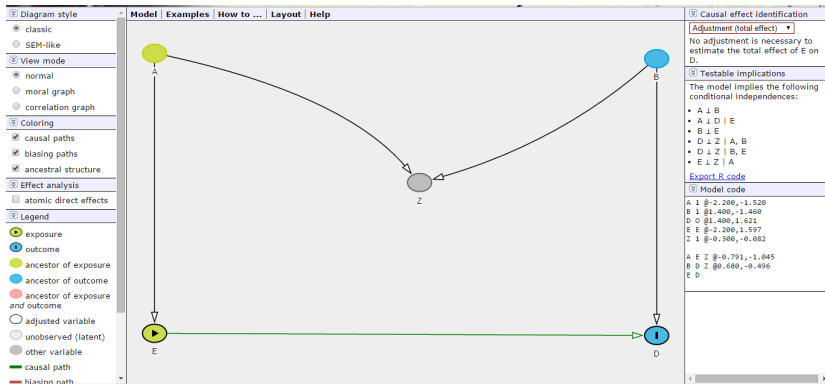
- ▶ O DAGitty é um ambiente baseado em navegador para criar, editar e analisar modelos causais (DAGs).
 - ▶ O foco está no uso de diagramas causais para minimizar o viés em estudos empíricos em epidemiologia e outras disciplinas.



- ▶ O DAGitty é desenvolvido e mantido por **Johannes Textor** (Tumor Immunology Lab and Institute for Computing and Information Sciences, Radboud University Nijmegen).
- ▶ <http://dagitty.net/>

DAGitty

Uma visão geral



DAGitty

Uma visão geral

The screenshot displays the DAGitty software interface, which is used for constructing and analyzing Directed Acyclic Graphs (DAGs). The interface is divided into several panels:

- Diagram style:** Includes options for 'classic' and 'SEM-like' styles.
- View mode:** Includes options for 'normal', 'moral graph', and 'correlation graph'.
- Coloring:** Includes checkboxes for 'causal paths', 'biasing paths', and 'ancestral structure'.
- Effect analysis:** Includes a checkbox for 'atomic direct effects'.
- Legend:** Defines node types (exposure, outcome, ancestor of exposure, ancestor of outcome, ancestor of exposure and outcome, adjusted variable, unobserved (latent), other variable) and path types (causal path, biasing path).

The central panel shows a DAG with nodes A, B, Z, E, and D. Node A is a yellow circle (exposure), node B is a blue circle (outcome), node Z is a grey circle (unobserved/latent), node E is a yellow circle with a play button (exposure), and node D is a blue circle with an exclamation mark (outcome). Edges connect A to Z, B to Z, A to E, and E to D. A green line connects E to D, indicating a causal path.

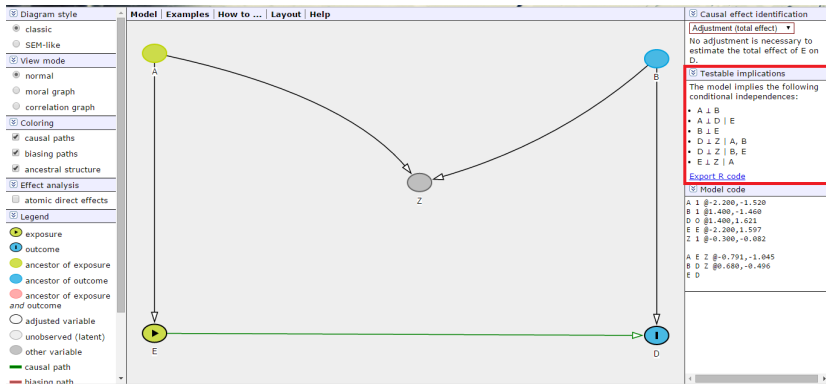
The right panel displays analysis results:

- Causal effect identification:** Shows 'Adjustment (total effect)' with a dropdown menu. Below it, a text box states: 'No adjustment is necessary to estimate the total effect of E on D.'
- Testable implications:** States 'The model implies the following conditional independences:' and lists:
 - $A \perp B$
 - $A \perp D \mid E$
 - $B \perp E$
 - $D \perp Z \mid A, B$
 - $D \perp Z \mid B, E$
 - $E \perp Z \mid A$
- Export R code:** Provides R code for the model, including:


```
A 1 @-2.200,-1.520
B 1 @1.400,-1.460
D 0 @1.400,1.621
E 0 @-2.200,1.597
Z 1 @-0.300,-0.082
A E Z @-0.791,-1.045
B D Z @0.680,-0.496
E D
```

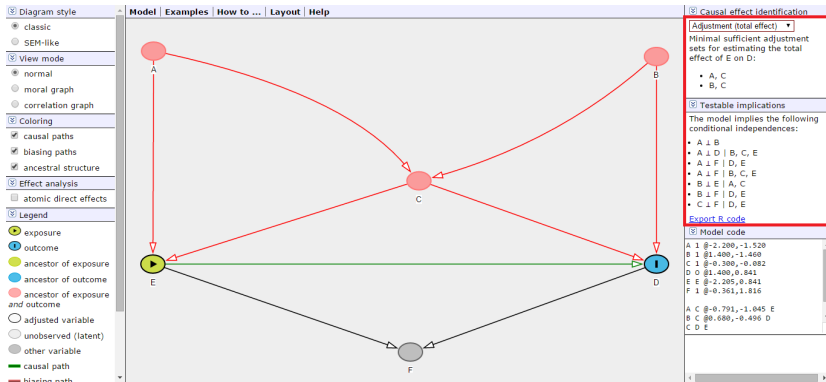
DAGitty

Uma visão geral



DAGitty

Uma visão geral



Criando um DAG no DAGitty

- ▶ No menu **Model**, clique em **New model**.
- ▶ O DAGitty irá solicitar o nome da variável de **exposição**, e logo em seguida o nome da variável de **desfecho**.
 - ▶ Estas variáveis serão criadas no grafo com o caminho $E \rightarrow D$.
- ▶ Para acrescentar uma nova variável dê um **duplo-clique** na área do grafo e dê um nome para esta variável.
- ▶ Para especificar uma relação entre duas variáveis, dê um duplo-clique na **variável de origem** e um duplo-clique na **variável de destino**.
 - ▶ O mesmo procedimento serve para remover uma relação entre duas variáveis já existente.
- ▶ Para remover uma variável do grafo, clique sobre a variável pressionando a **tecla D**.
- ▶ Para renomear uma variável, clique sobre a variável pressionando a **tecla R**.
- ▶ Na dúvida, consulte os menus **How to ...** e **Help**.

Criando um DAG no DAGitty

- ▶ No menu **Model** é possível:
 - ▶ exportar o grafo
 - ▶ publicar o grafo
- ▶ Uma vez publicado, o grafo pode ser:
 - ▶ modificado
 - ▶ apagado

Atividade 2

Atividade 2

- ▶ Utilize o DAGitty para construir o DAG do “**exemplo das Catecolaminas**”.
- ▶ Verifique se as conclusões obtidas concordam com as que você obteve com o critério back-door “feito a mão”.

DAGitty encontra o R

```
# Instala o pacote dagitty
# install.packages("dagitty")

# Carrega o pacote dagitty
library(dagitty)

# Carrega o DAG a partir do DAGitty
dag1 <- downloadGraph(x = "dagitty.net/mBYpOXW")
```

DAGitty encontra o R

```
plot(dag1)
```



DAGitty encontra o R

```
dag1.bd <- backDoorGraph(dag1)
plot(dag1.bd)
```



DAGitty encontra o R

```
print(adjustmentSets(dag1,  
                      effect = "total"))
```

```
## { ESTRESSE, IDADE, TABAGISMO }
```

```
# impliedConditionalIndependencies(dag1,  
#                                 max.results = 2)
```

ggdag: dagitty encontra o ggplot2

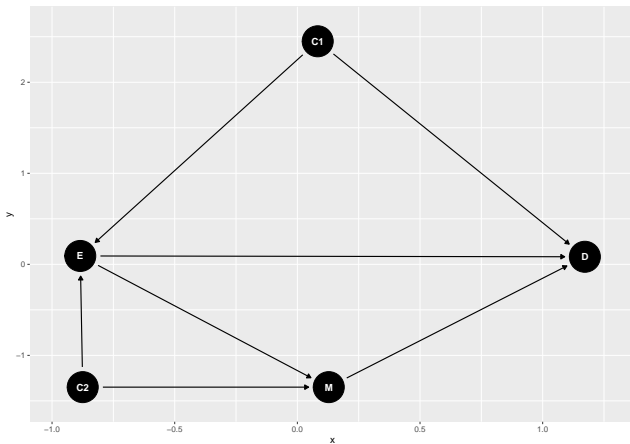
```
# Instala o pacote ggdag
# install.packages("ggdag")

# Carrega o pacote ggdag
library(ggdag)

# Carrega o DAG a partir do DAGitty
dag2 <- downloadGraph(x = "dagitty.net/mQLajCg")
```

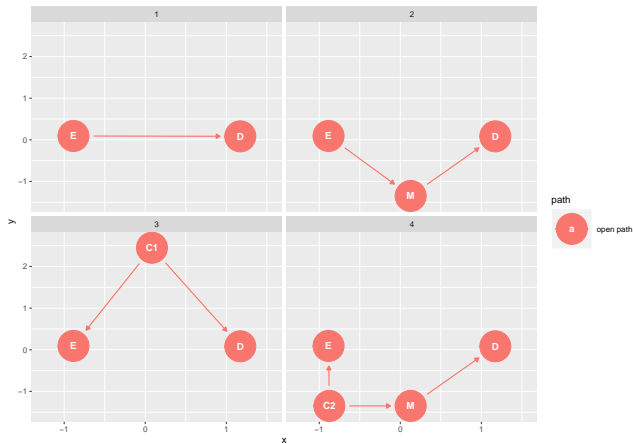
ggdag: dagitty encontra o ggplot2

```
ggdag(dag2)
```



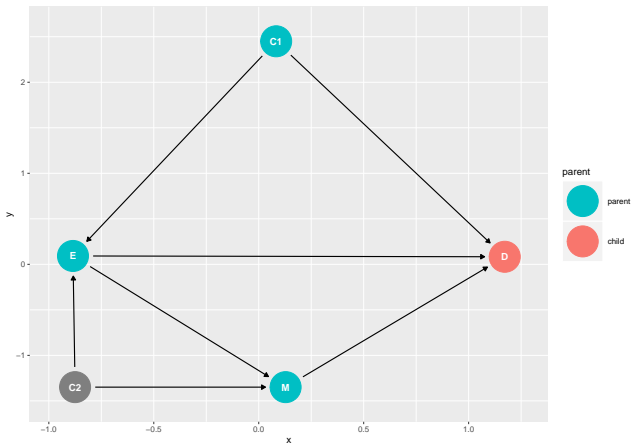
ggdag: dagitty encontra o ggplot2

```
ggdag_paths(dag2)
```



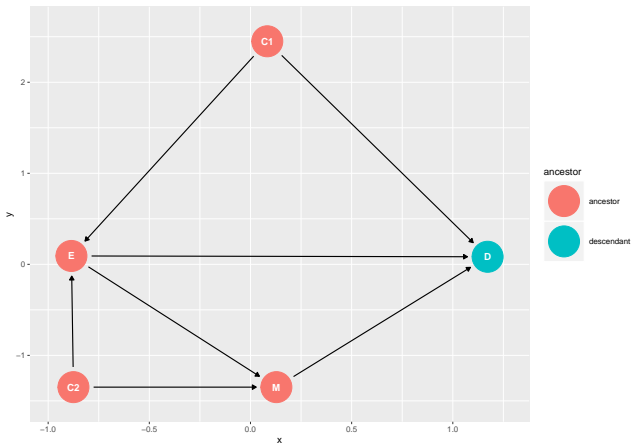
ggdag: dagitty encontra o ggplot2

```
ggdag_parents(dag2, "D")
```



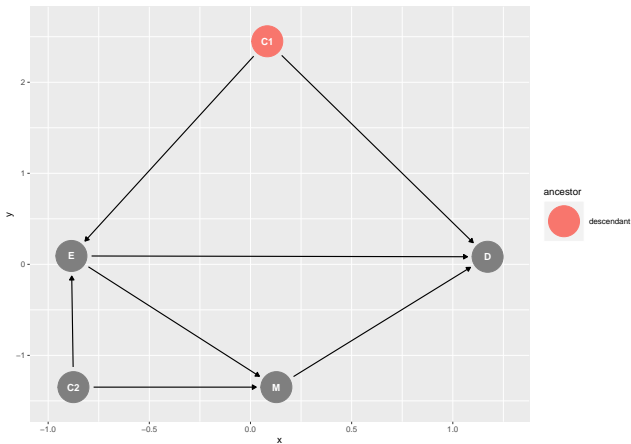
ggdag: dagitty encontra o ggplot2

```
ggdag_ancestors(dag2, "D")
```



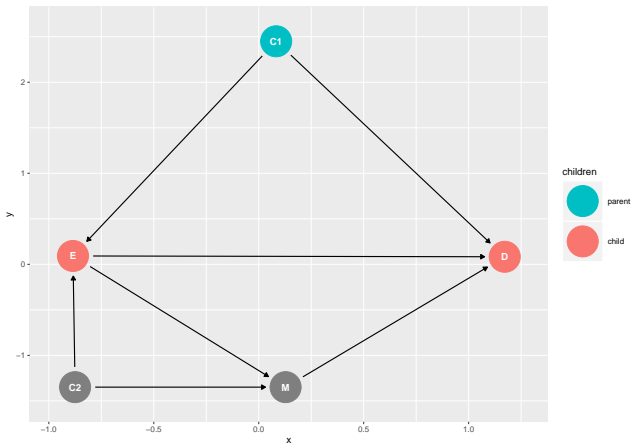
ggdag: dagitty encontra o ggplot2

```
ggdag_ancestors(dag2, "C1")
```



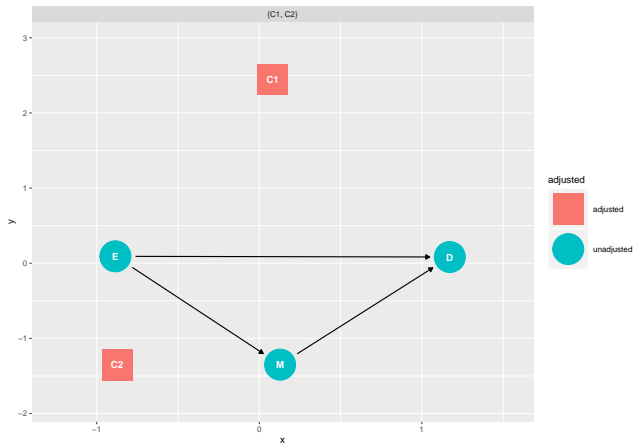
ggdag: dagitty encontra o ggplot2

```
ggdag_children(dag2, "C1")
```



ggdag: dagitty encontra o ggplot2

```
ggdag_adjustment_set(dag2)
```



Atividade 3

DAGitty

- ▶ Utilize um ou mais recursos computacionais para construir um DAG do quadro conceitual relacionado à sua questão de pesquisa.

Obrigado!

