back-door

yousri elfattah

2024-07-28

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

library(dagitty)  
library(ggdag)

##   
## Attaching package: 'ggdag'

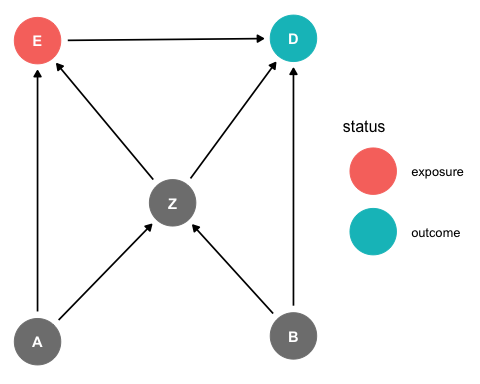
## The following object is masked from 'package:stats':  
##   
## filter

# Chapter 9

## Back-Door Criterion

First we load the causal model named “confounding” from the dagitty library using the function **dagitty::getExample**. We then use the function **dagitty::dagitty** to construct a **dagitty** graph object from a textual description and display the causal graph using **ggdag**.

g <- dagitty::getExample("confounding")  
g2 <- dagitty::dagitty(g)   
ggdag::ggdag\_status(g2) + ggdag::theme\_dag()



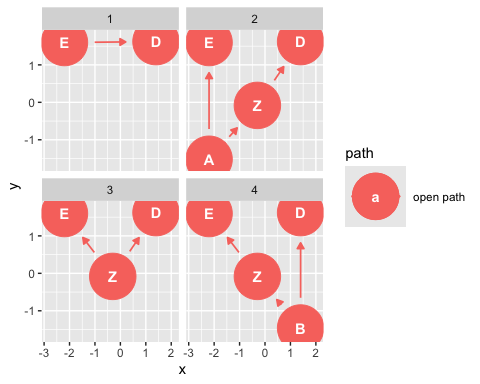
We can list all paths between “E” and “D” using the function **dagitty::paths**. There are five paths from the exposure “E” to the outcome “D” and the other four are biasing backdoor paths.

dagitty::paths(g, "E", "D", directed= FALSE)$paths

## [1] "E -> D" "E <- A -> Z -> D" "E <- A -> Z <- B -> D"  
## [4] "E <- Z -> D" "E <- Z <- B -> D"

We can visualize all paths between “E” and “D” using the function **ggdag::ggdag\_paths**

ggdag::ggdag\_paths(g2)



We can use the function **dagitty::adjustmentSets** to get the backdoor adjustment sets

dagitty::adjustmentSets(g2)

## { B, Z }  
## { A, Z }

We verify that the adjustment sets are backdoor adjustments satisfying the backdoor criterion; that is each adjustment set blocks all backdoor paths. It is already clear that the adjustment sets are not descendants of the exposure “E”. To verify the adjustment set blocks the backdoor paths we use the function **dagitty::dseparation** to check dseparation between “E” and “D” for the dag after removing the causal link “E -> D”

dagitty::dseparated( "dag {A -> E ; A -> Z ; B -> D ; B -> Z ; Z -> D ; Z -> E}","E","D",list("A","Z"))

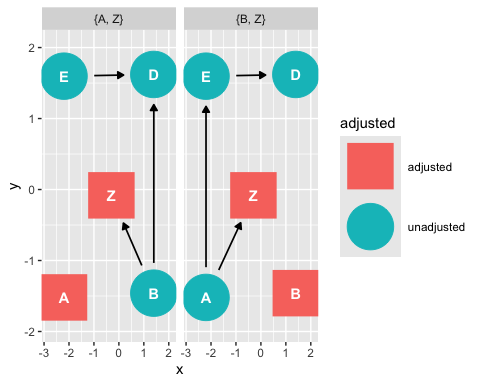
## [1] TRUE

dagitty::dseparated( "dag {A -> E ; A -> Z ; B -> D ; B -> Z ; Z -> D ; Z -> E}","E","D",list("B","Z"))

## [1] TRUE

We can visualize the backdoor adjustments using the function **ggdag::ggdag\_adjustment\_set**

ggdag::ggdag\_adjustment\_set(g2)



## Simulation data

We generate simulated observational data for the linear Gaussion SCM

# simulate data (linear Gaussain back-door model)  
n <- 1e4  
A <- rnorm(n)  
B <- rnorm(n)  
Z <- 0.3 \* A + 0.7 \* B + rnorm(n)  
E <- 0.5 \* Z + 0.35 \* A + rnorm(n)  
D <- 1.2 \* E + 0.9 \* B + 0.65 \* Z + rnorm(n)  
model.data <- data.frame (  
 A <- A,  
 B <- B,  
 Z <- Z,  
 E <- E,  
 D <- D  
)

## Linear regression

Adjusting for {“A”, “Z”}

coef(lm(D ~ E + A + Z, model.data))

## (Intercept) E A Z   
## 0.02274868 1.20442048 -0.11863110 1.06994194

Adjusting for {“B”, “Z”}

coef(lm(D ~ E + B + Z, model.data))

## (Intercept) E B Z   
## 0.004957568 1.217314671 0.912211514 0.643561678

## Drawing a scatter plot

In this script, the scatter plot is generated using **geom\_point()**, and the linear regression line is added with **geom\_smooth()** where method = “**lm**” specifies a linear model. The caption includes the estimated slope from the linear regression.

library(ggplot2)  
  
# Recreate model and extract coefficients  
model <- lm(D ~ E + A + Z, model.data)  
model.coefs <- coef(model)  
  
# Create scatter plot of D versus E  
ggplot(model.data, aes(x = E, y = D)) +  
 geom\_point(alpha = 0.5, size = 1) + # Draw points  
 geom\_smooth(method = "lm", col = "red") + # Add linear regression line  
 labs(  
 title = "Scatter plot of D versus E with linear regression line",  
 x = "E",  
 y = "D",  
 caption = paste0("Slope: ", round(model.coefs["E"],2))  
 ) +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'

