Exercises for Graphical Causal Models part 2

Gertjan Verhoeven, Dutch Healthcare Authority

November 8, 2018

```r
library(dagitty)
library(ggplot2)
```

PART 2

Exercise 4: D-separation

(adapted from Study question 2.4.1 from Pearl, Glymour and Jewell 2016)

First, visit (http://dagitty.net/learn/dsep/index.html) for an interactive tutorial on d-separation.

then:

Figure 2.9 below represents a causal graph from which the error terms have been deleted. Assume that all those errors are mutually independent.

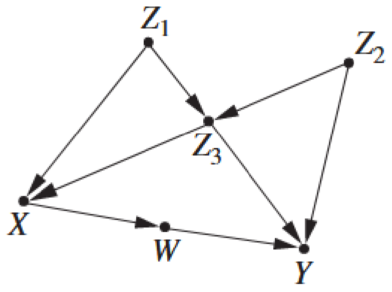
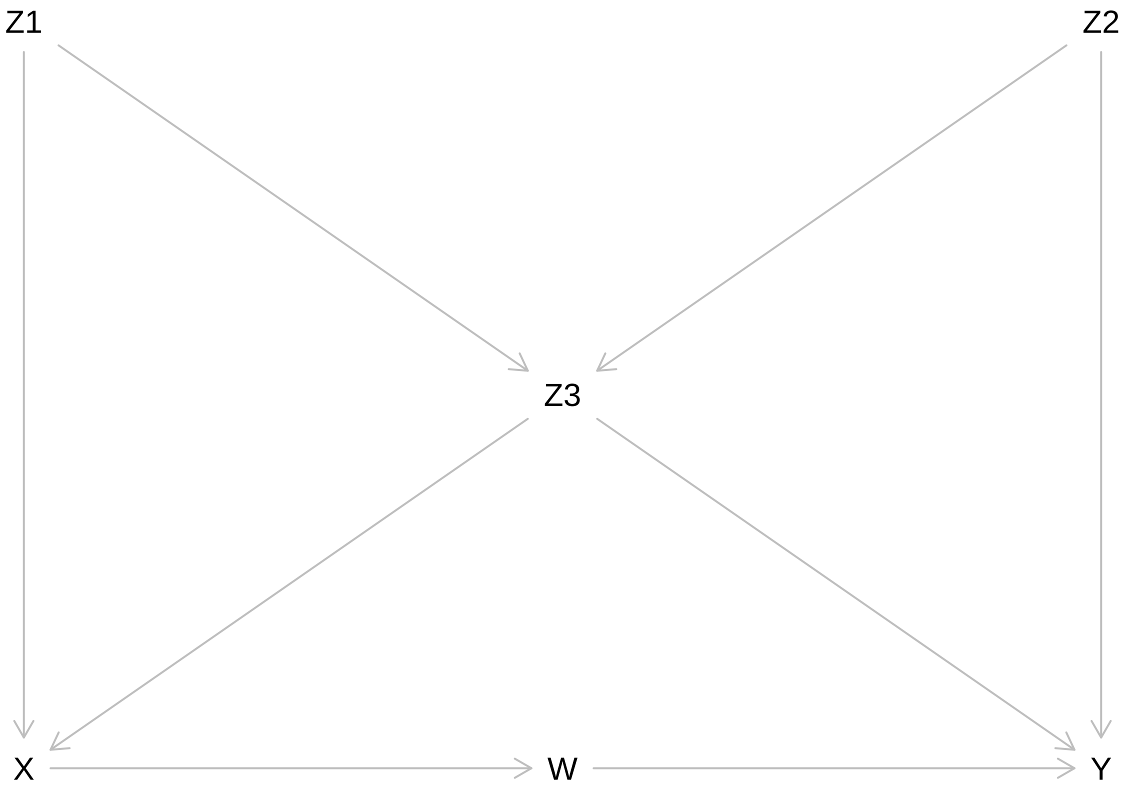


Figure 2.9 A causal graph used in study question 2.4.1, all *U* terms (not shown) are assumed independent

Explore the functionality of the dagitty package for these exercises. Check functions `impliedConditionalIndependencies()`, `dseparated()`, `paths()`.

(a) Draw the DAG using dagitty.

```r
g1 <- dagitty( 'dag {
    X [pos="0,2"]
    Y [pos="2,2"]
    Z1 [pos="0,0"]
    Z2 [pos="2,0"]
    Z3 [pos="1,1"]
    W [pos="1,2"]
    Z1 -> Z3 -> Y
    Z2 -> Y
    Z2 -> Z3 -> X
  X -> W -> Y
  Z1 -> X
}')

plot(g1)
```



b) List all conditional independencies in this graph.

```r
print( impliedConditionalIndependencies( g1 ) )
```

```
## W _||_ Z1 | X
## W _||_ Z2 | Z1, Z3
## W _||_ Z2 | X
## W _||_ Z3 | X
## X _||_ Y | W, Z2, Z3
## X _||_ Y | W, Z1, Z3
## X _||_ Z2 | Z1, Z3
## Y _||_ Z1 | X, Z2, Z3
## Y _||_ Z1 | W, Z2, Z3
## Z1 _||_ Z2
```

c) List all paths between X and Y.

```r
paths( g1, "X", "Y")
```

```
## $paths
## [1] "X -> W -> Y"             "X <- Z1 -> Z3 -> Y"
## [3] "X <- Z1 -> Z3 <- Z2 -> Y" "X <- Z3 -> Y"
## [5] "X <- Z3 <- Z2 -> Y"
##
```

```
## $open
## [1]  TRUE  TRUE FALSE  TRUE  TRUE
```

d) Which set of variables d-separates this pair of variables? Explain why using the list of all paths between X and Y.

Solution: we start with blocking the shortest open paths. This leads us to conclude that W and Z3 are always required. However, conditioning on Z3 OPENS up a a previously closed path (`X <- Z1 -> Z3 <- Z2 -> Y`). We can either add Z1 or Z2 to close this path again. Adding either one of these does not re-open previously closed paths. We end up with the two implied conditional independencies between X an Y we already found at a).

e) List all conditional independencies in this graph, assuming that only variables in the set {Z3,W,X,Z1} can be measured.

Use `latents()`.

```r
latents( g1 ) <- setdiff( names(g1), c("Z3", "W", "X", "Z1") )
latents( g1 )
```

```
## [1] "Y"  "Z2"
```

```r
impliedConditionalIndependencies( g1 )
```

```
## W _||_ Z1 | X
## W _||_ Z3 | X
```

(f) For pairs {Z1, W} and {Z1, Z2}, determine whether they are independent conditional on all other variables in the graph. (Remember to turn `latents()` off again.)

```r
latents(g1) <- NULL
# {Z1.W}
dseparated(g1, "Z1", "W", c("X", "Y", "Z2", "Z3"))
```

```
## [1] TRUE
```

```r
# {Z1, Z2}
dseparated(g1, "Z1", "Z2", c("X", "Y", "W", "Z3"))
```

```
## [1] FALSE
```

Open question: why do we not find the c. indepency Z1 indep W given X, Y, Z2 and Z3 as output in a)

(g) Suppose we wish to estimate the value of Z2 from measurements taken on all other variables in the model. Find the smallest set of variables that would yield as good an estimate of Z2 as when we measured all variables.

```r
markovBlanket(g1, "Z2")
```

```
## [1] "Y"  "Z3" "W"  "Z1"
```

Exercise 5: Valid adjustment sets

In this exercise, we will use various Graphical Identification criteria to identify causal effects in DAGs.

We use the DAG in Figure 13.8 from Elwert.

a) Code the DAG in R using `dagitty`.

```
g <- dagitty('dag {
    A -> Y
    A -> C -> Y
    B -> C
    B -> T -> F -> Y
    T -> E -> Y
    D -> E
    B -> D -> Y
}')
```

b) List all paths between T and Y. which paths are open?

```
paths(g, "T", "Y")
```

```
## $paths
## [1] "T -> E -> Y"                     "T -> E <- D -> Y"
## [3] "T -> E <- D <- B -> C -> Y"      "T -> E <- D <- B -> C <- A -> Y"
## [5] "T -> F -> Y"                     "T <- B -> C -> Y"
## [7] "T <- B -> C <- A -> Y"           "T <- B -> D -> E -> Y"
## [9] "T <- B -> D -> Y"
##
## $open
## [1]  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
```

c) For all open paths between T and Y, which are the causal paths? Which are open backdoor paths?

There are five open paths.

Causal paths: T -> E -> Y T -> F -> Y Backdoor paths: “T <- B -> C -> Y” “T <- B -> D -> E -> Y” “T <- B -> D -> Y”

d) What is the definition of the back-door criterion? Explain why Z = (B) satisfies this criterion.

- No variable in Z is a descendant of T.
- Z blocks all backdoor paths from T to Y.

From c) we see that B blocks all backdoor paths, and B does not lie on either causal path.

e) What is the definition of the adjustment criterion? How does it differ from the back-door criterion?

- Z blocks all noncausal paths from T to Y.
- No variable in Z lies on the causal path, or descends from a variable on the causal path.

The backdoor criterion tells us to ignore noncausal paths that start with an arrow out of treatment.

f) List all sets of variables that satisfy the “Adjustment criterion” for this DAG. Use the dagitty function `adjustmentSets()` (Important: by default this function only returns the minimal sufficient adjustment set).

What is the smallest adjustment set for this causal effect?

```
adjustmentSets(g, exposure = "T", outcome = "Y", type = "all", effect = "total")
```

```
##  { B }
##  { A, B }
##  { B, C }
##  { A, B, C }
##  { B, D }
##  { A, B, D }
##  { A, C, D }
##  { B, C, D }
##  { A, B, C, D }
```

B is smallest set.

g) Show that Z = (A, C, D) satisfies the adjustment criterion.

Z does not contain E or F, this satisfies the second part. And conditioning on (A,C,D) blocks all noncausal open paths.

h) What is the definition of the Parents of the treatment criterion? Apply the “Parents of the treatment criterion” to identify a valid adjustment set for the causal effect of T on Y.

Conditioning on all parents of treatment T identifies the total causal effect of T on any outcome. The parents of T are “B”.

i) What is the definition of the Parents of the outcome criterion? Can the “Parents of the outcome criterion” be applied to this graph?

If no backdoor path shares a variable with any causal path other than T and Y, conditioning on the parents of the outcome that do not lie on a causal path from T to Y identifies the total causal effect of T on Y.

Variable/node E is both in a causal path AND in a backdoor path, so does not work here.

Exercise 6: Adjustment sets: estimation and precision

We expect that by adding more predictors that explain variance in the outcome Y, we get a better estimate of the effect of T on Y.

The following code simulates data from the DAG from the previous exercise (Fig 13.8 Elwert)

The structural equations all consist of linear relationships and Gaussian noise for all variables.

```r
set.seed(123)
N <- 1000 # sample size
Ua <- rnorm( N ); Ub <- rnorm( N ); Uc <- rnorm( N );
Ud <- rnorm( N ); Ue <- rnorm( N ); Uf <- rnorm( N );
Ut <- rnorm( N ); Uy <- rnorm( N );

A <- Ua
B <- Ub
C <- 3 * A - 2 * B + Uc
D <- 0.5 * B + Ud
T <- -3 * B + Ut
E <- T + 3.5 * D + Ue
F <- 0.8 * T + Uf
Y <- 1.1 * A - 2 * C + 3 * D + 0.4 * E + 0.7 * F + Uy

df <- data.frame(A, B, C, D, T, E, F, Y)
```

a) Derive from the equations what the total causal effect is of T on Y. I.e. how much does the value of Y change if we change T with one unit.

We expect for the causal effect of T on Y:

Causal path T -> F -> Y: 0.8 x 0.7 = 0.56 Causal path T -> E -> Y: 1.0 x 0.4 = 0.4

Total effect: 0.96

b) Estimate this causal effect from the data using `lm()`, using the *smallest* adjustment set identified in the previous exercise.

```r
lmfit <- lm(Y ~ T + B)
summary(lmfit)
```

```
##
## Call:
## lm(formula = Y ~ T + B)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.7945  -4.4088   0.0454   4.7058  19.0208
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03899    0.21838  -0.179    0.858
## T            1.06309    0.21948   4.844 1.48e-06 ***
## B            6.03830    0.68800   8.777  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.896 on 997 degrees of freedom
## Multiple R-squared:  0.1674, Adjusted R-squared:  0.1657
## F-statistic: 100.2 on 2 and 997 DF,  p-value: < 2.2e-16
```

c) Estimate this causal effect from the data using `lm()`, using the *largest* adjustment set identified in the previous exercise.

```r
lmfit <- lm(Y ~ T + A + B + C + D)
summary(lmfit)
```

```
##
## Call:
## lm(formula = Y ~ T + A + B + C + D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8462 -0.8938 -0.0196  0.8772  3.8663
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01561    0.03964   0.394    0.694
## T            0.98114    0.03986  24.615   <2e-16 ***
## A            1.24420    0.12723   9.779   <2e-16 ***
## B           -0.03571    0.14948  -0.239    0.811
## C           -2.05921    0.04054 -50.798   <2e-16 ***
## D            4.42843    0.03995 110.847   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.251 on 994 degrees of freedom
## Multiple R-squared:  0.9727, Adjusted R-squared:  0.9725
## F-statistic:  7075 on 5 and 994 DF,  p-value: < 2.2e-16
```

Compare both estimates, which is more precise (i.e. smallest std errors) and why?

d) Is the largest adjustment set also the set that provides the most precise estimate (i.e. smallest std errors) for the effect of T on Y? Or can you find a better adjustment set?

No, B is screened by A, C and D. Leave out B.

```
lmfit <- lm(Y ~ T + A + C + D)
summary(lmfit)
```

```
## 
## Call:
## lm(formula = Y ~ T + A + C + D)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.8238 -0.8890 -0.0202  0.8682  3.8798 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.01586    0.03960    0.40    0.689    
## T            0.98864    0.02456   40.26   <2e-16 ***
## A            1.22875    0.10951   11.22   <2e-16 ***
## C           -2.05400    0.03416  -60.13   <2e-16 ***
## D            4.42707    0.03952  112.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.251 on 995 degrees of freedom
## Multiple R-squared:  0.9727, Adjusted R-squared:  0.9726 
## F-statistic:  8853 on 4 and 995 DF,  p-value: < 2.2e-16
```

Exercise 7: NPSEM: Estimation of causal effects and effect heterogeneity

NPSEM stands for non-parametric structural equation models. This means that our causal reasoning around DAG's (causal graphs) does not make any parametric assumptions on the relationships between the variables. It does assume that the relationship between the parents of a variable and the variable itself can be described by a structural equation.

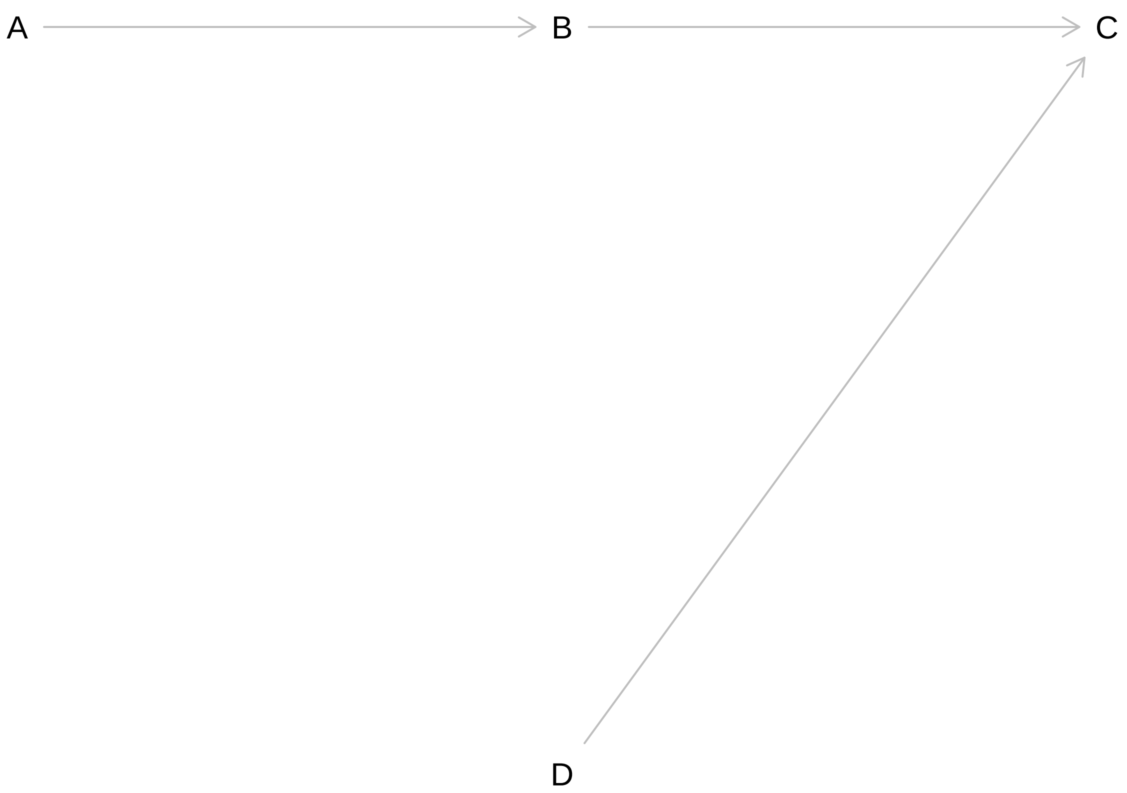
[Data is generated according to causal model Fig. 13.7 from Elwert 2013. The specific parametric equations are not supplied but are to be discovered by estimation from data.]

```
set.seed(123)
N <- 10000 # sample size
Ua <- rnorm( N ); Ub <- rnorm( N ); Uc <- rnorm( N );Ud  <- rnorm( N );
A <- Ua
B <- -2 * A + Ub
D <- 3 + B + Ud
C <- (B*D)^2 + Uc

df <- data.frame(A, B, C, D)
saveRDS(df, "fig13.7_data.rds")
```

A paper is published that claims a causal model depicted in Figure 13.7. The causal model is used to draw inferences from an observational dataset. The dataset is published as open data with the paper (Awesome!). You decide to check whether the data is consistent with their causal model.

a) Code the DAG in R.

```
g <- dagitty('dag {
    A [pos="0,0"]
    B [pos="1,0"]
    D [pos="1,1"]
    C [pos="2,0"]

    A -> B -> C
    D -> C
}')
plot(g)
```



b) Read in the dataset (fig13.7_data.rds). List all implied dependencies, both marginal and conditional. Test these dependencies in the data, for example by calculating correlation coefficients (linear: Pearson, non-linear: MIC/ dCor https://m-clark.github.io/docs/CorrelationComparison.pdf), or simple scatterplots with geom_smooth() or using the dagitty function `localTests()`.

```
impliedConditionalIndependencies(g)
```

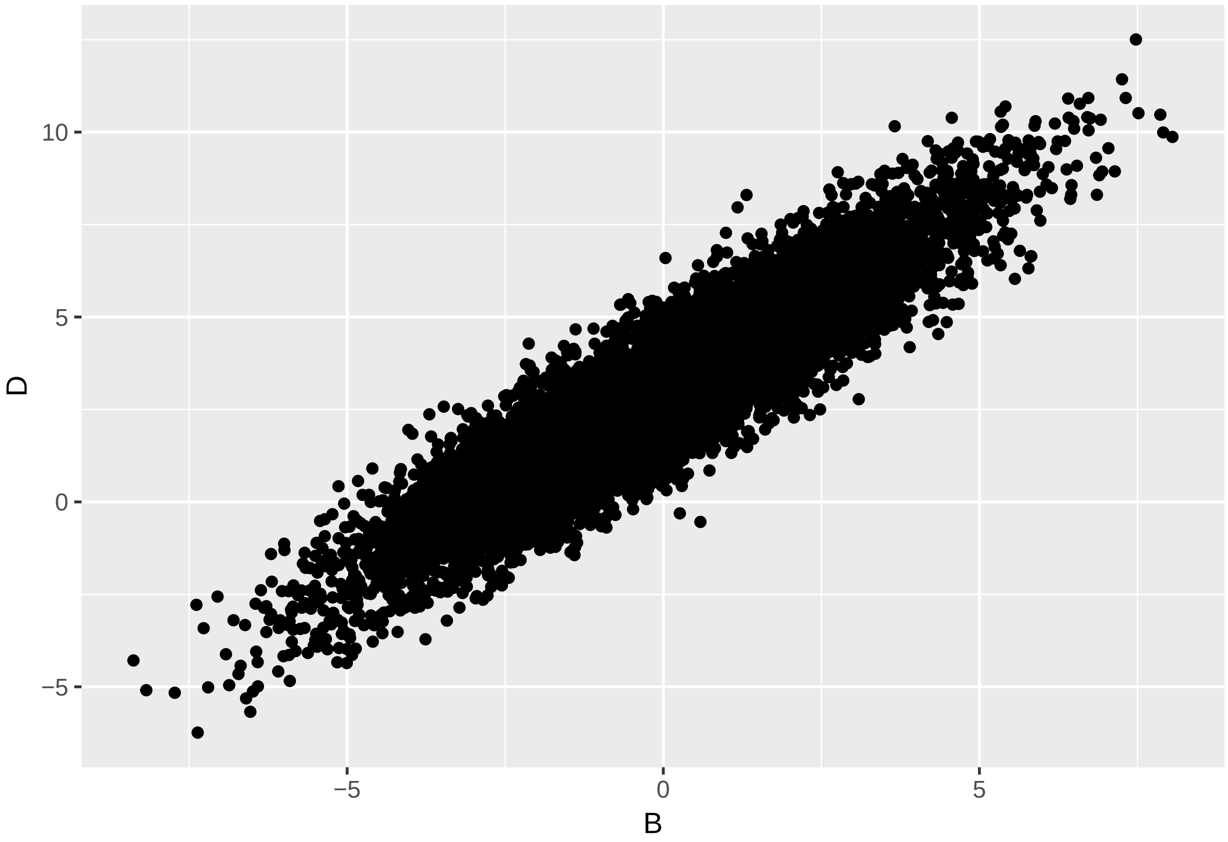
```
## A _||_ C | B
## A _||_ D
## B _||_ D
```

```
localTests(g, df)
```

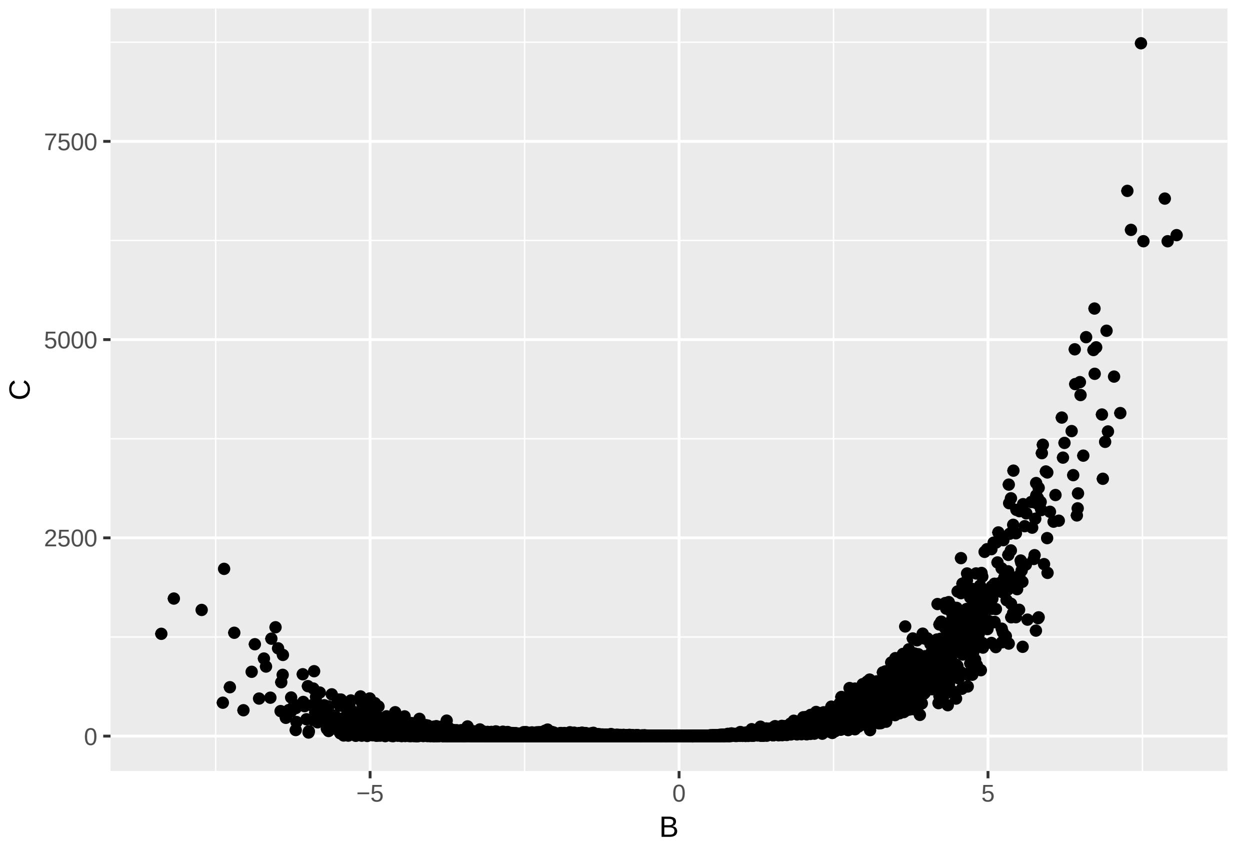
```
##                    estimate    std.error   p.value          2.5%
## A _||_ C | B  1.517889e-05 0.0000125516 0.2265689 -9.424775e-06
## A _||_ D     -3.330494e-01 0.0023370222 0.0000000 -3.376304e-01
## B _||_ D      8.294528e-01 0.0036965944 0.0000000  8.222067e-01
##                        97.5%
## A _||_ C | B  3.978255e-05
## A _||_ D     -3.284684e-01
## B _||_ D      8.366989e-01
```

Are there any dependencies in the data that are in strong contradiction with the causal model?

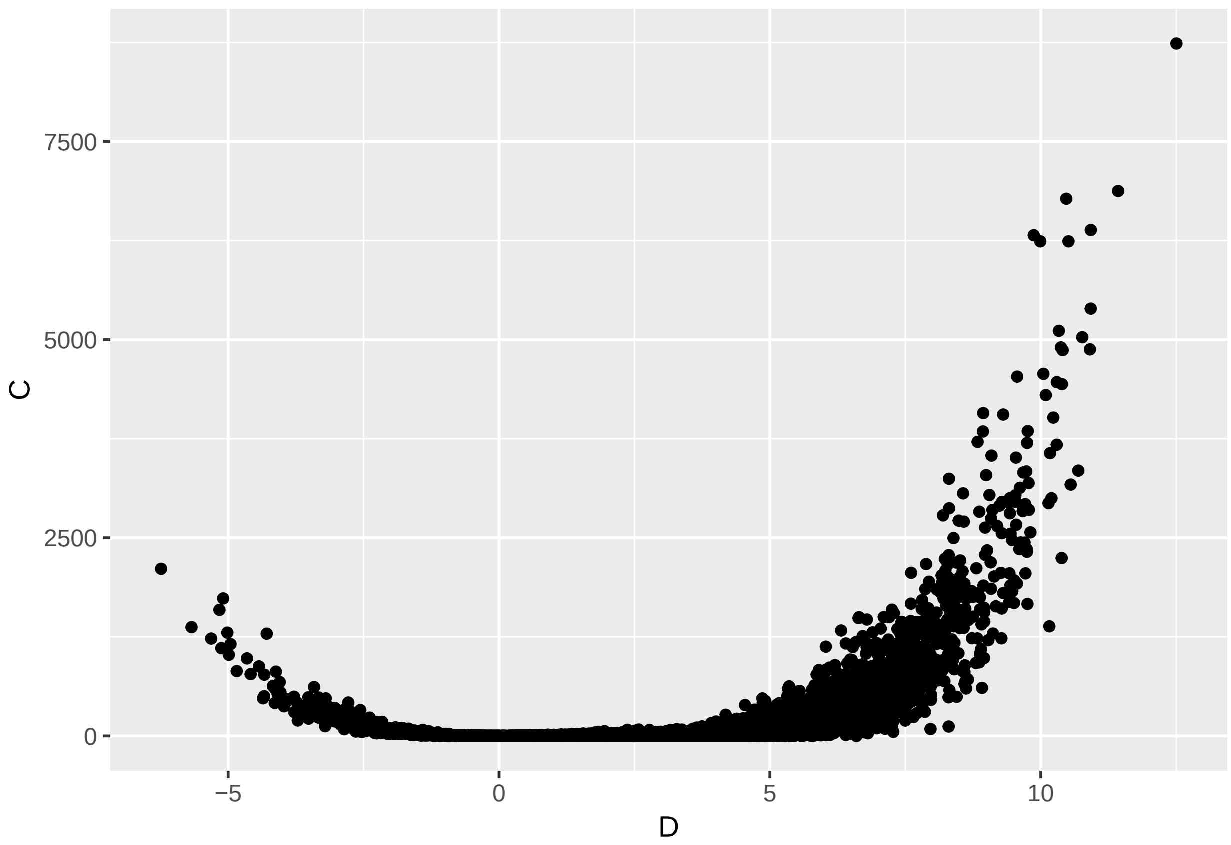
```
ggplot(df, aes(x = B, y = D)) + geom_point()
```



```
ggplot(df, aes(x = B, y = C)) + geom_point()
```

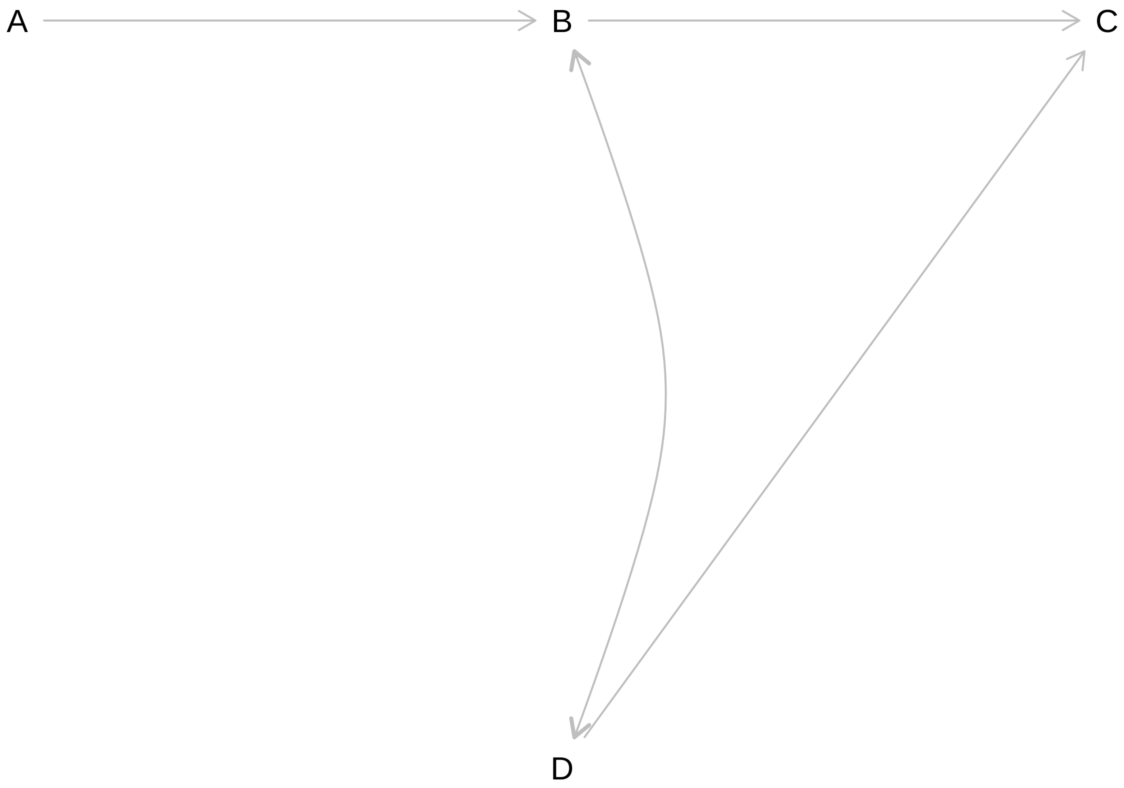


```
ggplot(df, aes(x = D, y = C)) + geom_point()
```



Update the graph:

```
g <- dagitty('dag {
    A [pos="0,0"]
    B [pos="1,0"]
    D [pos="1,1"]
    C [pos="2,0"]

    A -> B -> C
    B <-> D
    D -> C
}')
plot(g)
```



c) Use Random forest to model function $C = f_C(B, D, U_C)$ from the data. This allows non-linearities and interactions in the predictors B and D.

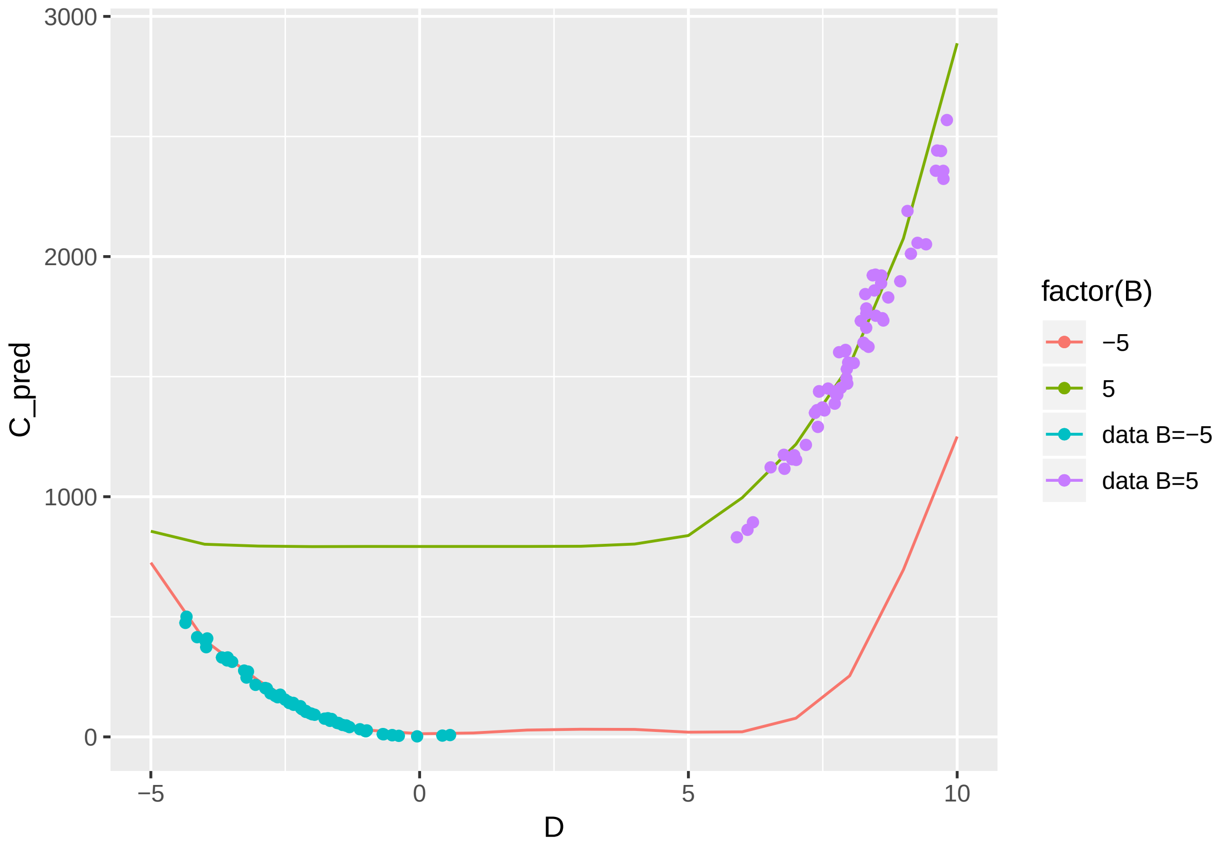
You can use the R package `ranger` for this.

```r
library(ranger)
rf.fit <- ranger("C ~ B + D", data = df)
```

d) predict C over the data range of D for both B=-5 and B= +5. Plot both curves in one graph. Is the causal effect of B on C homogeneous, or does it depend on D?

(Hint: `ranger` has a `predict` method that can work on new data, the predictions are in `predict_output$predictions`)

```r
newdata <- rbind(data.frame(B = -5, D = c(-5:10)),
                 data.frame(B = 5, D = c(-5:10)))
newdata$C_pred <- predict(rf.fit, data = newdata)$predictions

ggplot(newdata, aes(x = D, y = C_pred, group = factor(B), col = factor(B))) + geom_line() +
  geom_point(data = subset(df, abs(B+5) < 0.2), aes(x = D, y = C, group = "bla", col = "data B=-5")) +
  geom_point(data = subset(df, abs(B-5) < 0.2), aes(x = D, y = C, group = "bla", col = "data B=5"))
```



e) Given a particular value of D, do we have data where B is in the region around -5, as well as data where B is in the region around 5?

(I.e. For a given counfounder D, do we have overlap of treatment types (value of B) in the data.)

What does this mean for our causal effect estimates of B on C?

```r
# around D = -2, check over what range we have data on B
summary(subset(df, abs(D+2) < 0.2)$B)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -6.192  -4.520  -3.986  -3.907  -3.241  -2.444
```

For D around -2, we only have B values between -6 and -2.

So, we do not have both data. So we do not have counterfactual observations, relying on extrapolation.