# Solutions to: Exercises for Graphical Causal Models part 1

*Gertjan Verhoeven, Dutch Healthcare Authority*

*October 24, 2018*

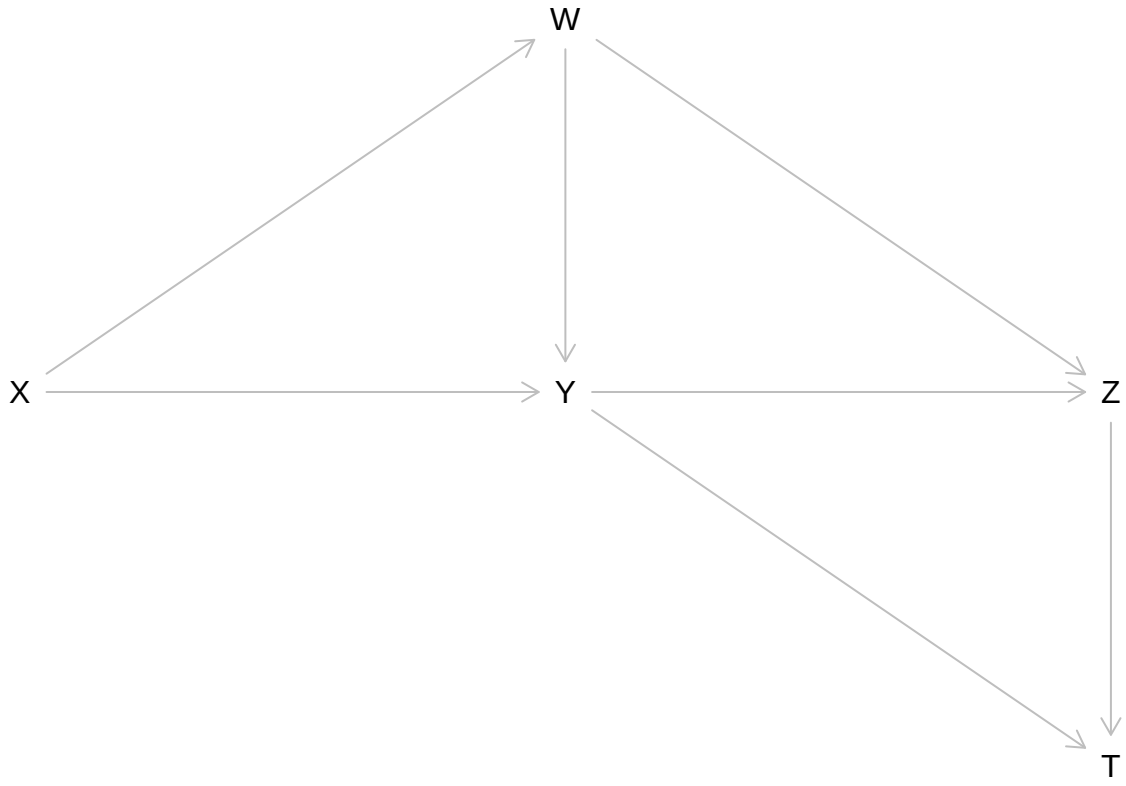
## Exercise 1: DAG Terminology

(From Causal Inference in Statistics: A Primer" by Pearl, Glymour, and Jewell, 2016.)

We first define the graph in this exercise using dagitty syntax and plot it.

```
g <- dagitty('dag {
    X [pos="0,1"]
    Y [pos="1,1"]
    Z [pos="2,1"]
    W [pos="1,0"]
    T [pos="2,2"]

    X -> Y -> Z -> T
    X -> W -> Y -> T
    W -> Z
}')
plot(g)
```




a) Name all of the parents of Z.

These are the direct causes of Z, i.e. W and Y.

b) Name all of the ancestors of Z.

This means all parents, and all parents of parents etc. So W, Y and X.

c) Name all of the children of W.

All vars for which W is a parent. So Y and Z.

d) Name all of the descendants of W.

This means all children, and all children of children etc. So Y, Z, T

e) Draw all (simple) paths between X and T.

Paths are sequences of adjacent arrows that traverse each var at most once. Arrows may point in any direction. So:

$X \to W \to Z \to T$
$X \to W \to Y \to Z \to T$
$X \to W \to Z \leftarrow Y \to T$
$X \to Y \leftarrow W \to Z \to T$
$X \to Y \to Z \to T$
$X \to Y \to T$
$X \to W \to Y \to T$

Check our answer using dagitty:

```
paths( g, "X", "T" )$paths
```

```
## [1] "X -> W -> Y -> T"      "X -> W -> Y -> Z -> T" "X -> W -> Z -> T"
## [4] "X -> W -> Z <- Y -> T" "X -> Y -> T"           "X -> Y -> Z -> T"
## [7] "X -> Y <- W -> Z -> T"
```

It lists 7 paths. We haves 7 paths.

f) Draw all the directed paths between X and T.

We should now drop all paths that contain a collider. This holds for two paths:

$X \to W \to Z \leftarrow Y \to T$ $X \to Y \leftarrow W \to Z \to T$

Check our answer using dagitty:

```
paths( g, "X", "T", directed=TRUE )$paths
```

```
## [1] "X -> W -> Y -> T"      "X -> W -> Y -> Z -> T" "X -> W -> Z -> T"
## [4] "X -> Y -> T"           "X -> Y -> Z -> T"
```

## Exercise 2: Elementary causal structures

For each of the questions below: simulate a dataset of 10.000 datapoints from the causal model using `rnorm()`. Assume for $U_x$, $U_y$ and $U_z$ normally distributed noise with mean zero and standard deviation 5. Dependence or independence can be empirically verified by scatterplots with a smoother (geom_smooth()) or by calculating a correlation coefficient (for linear dependencies).

a) chain:

$X = U_x$
$Y = 2X + U_y$
$Z = \sqrt{|Y|} + U_z$

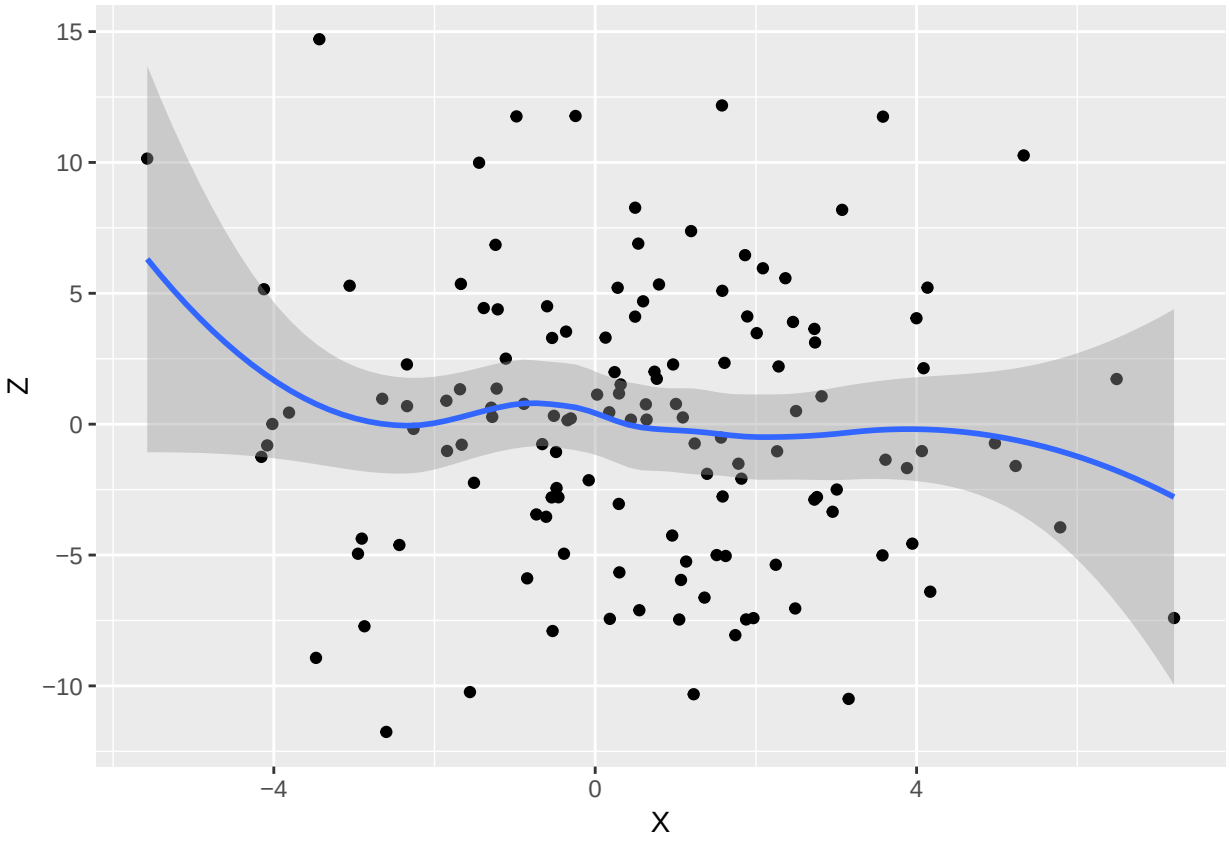
Verify that exposure X and outcome Z are independent (uncorrelated) given mediator Y by subsetting the data on a small interval of Y (say 0.8 and 1.2).

```
set.seed(123)
n <- 10000
X <- rnorm(n, 0, 5)
Y <- 2 * X + rnorm(n, 0, 5)
Z <- sqrt(abs(Y)) + rnorm(n, 0, 5)

df <- data.table(X, Y, Z)

ggplot(df[Y %between% c(0.8, 1.2)], aes(x = X, y = Z)) +
  geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




b) fork:

$Z = U_z$
$X = -0.5Z + U_x$
$Y = 2Z + U_y$

Here Z is the common cause of X and Y.

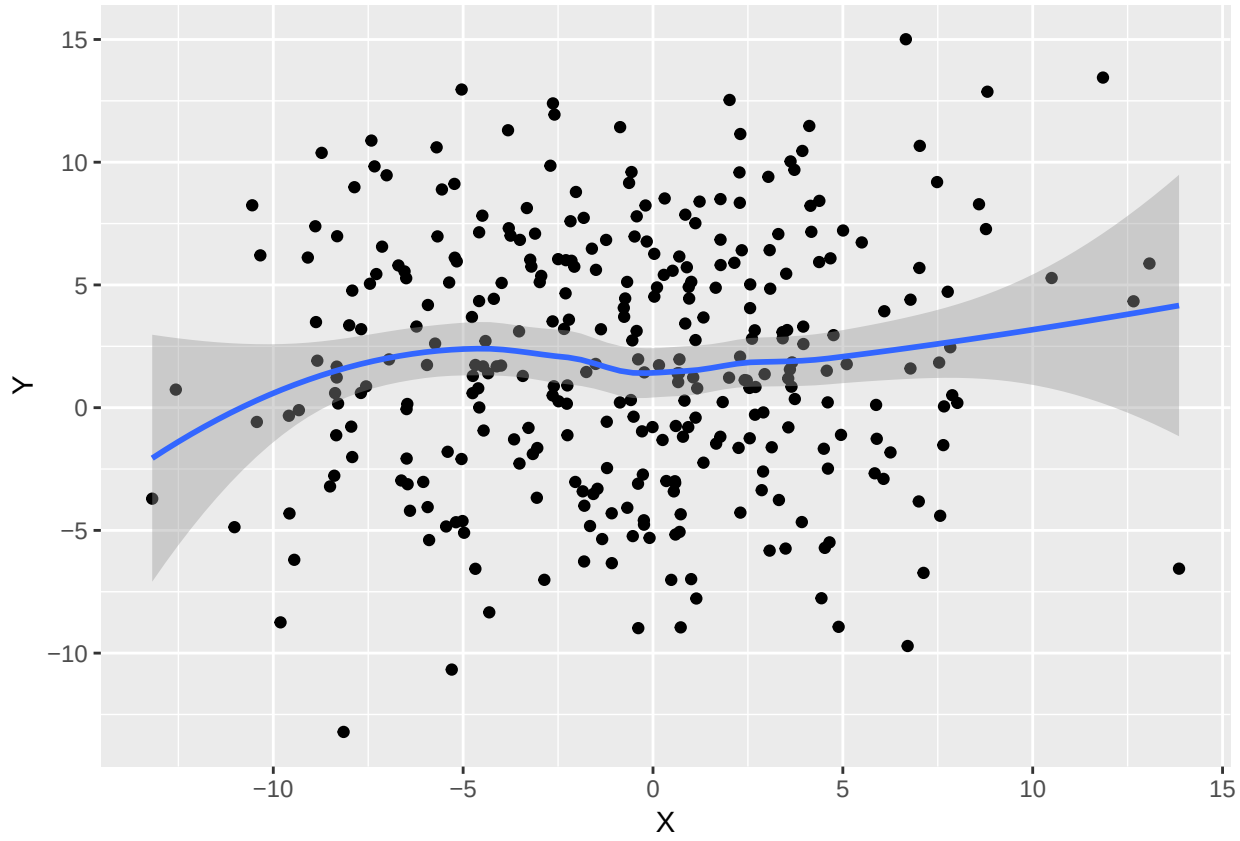
Empirically verify that X and Y are independent given Z. Do this by subsetting the data on a small interval of Z.

```
set.seed(123)
n <- 10000
Z <- rnorm(n, 0, 5)
X <- -0.5 * Z + rnorm(n, 0, 5)
Y <- 2 * Z + rnorm(n, 0, 5)


df <- data.table(X, Y, Z)

ggplot(df[Z %between% c(0.8, 1.2)], aes(x = X, y = Y)) +
  geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




c) inverted fork (collider):

$X = U_x$
$Y = U_y$
$Z = 2X - 4Y + U_z$

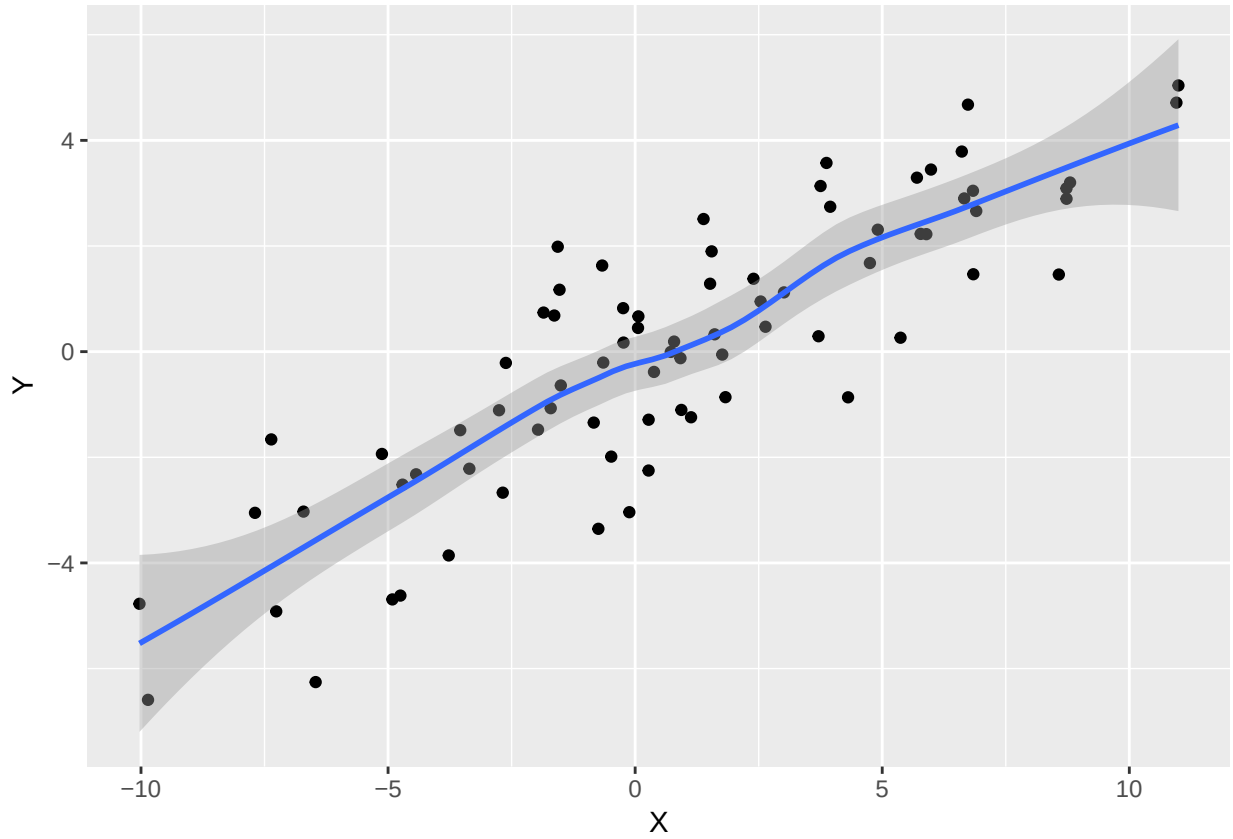
Verify that X and Y are dependent given Z after subsetting the data on a small interval of Z. Here Z is a descendant of both X and Y.

```
set.seed(123)
n <- 10000
X <- rnorm(n, 0, 5)
Y <- rnorm(n, 0, 5)
Z <- 2*X + -4*Y + rnorm(n, 0, 5)

df <- data.table(X, Y, Z)

ggplot(df[Z %between% c(0.8, 1.2)], aes(x = X, y = Y)) +
  geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




## Exercise 3: Playing around with SCM's

(Study question 1.5.1 From Pearl, Glymour and Jewell)

Suppose we have the following SCM. Assume all exogenous variables are independent and that the expected value of each is 0.

$X = U_x$
$Y = 1/3X + U_y$
$Z = 1/16Y + U_z$

First, let us generate some data by directly implementing the model specification.

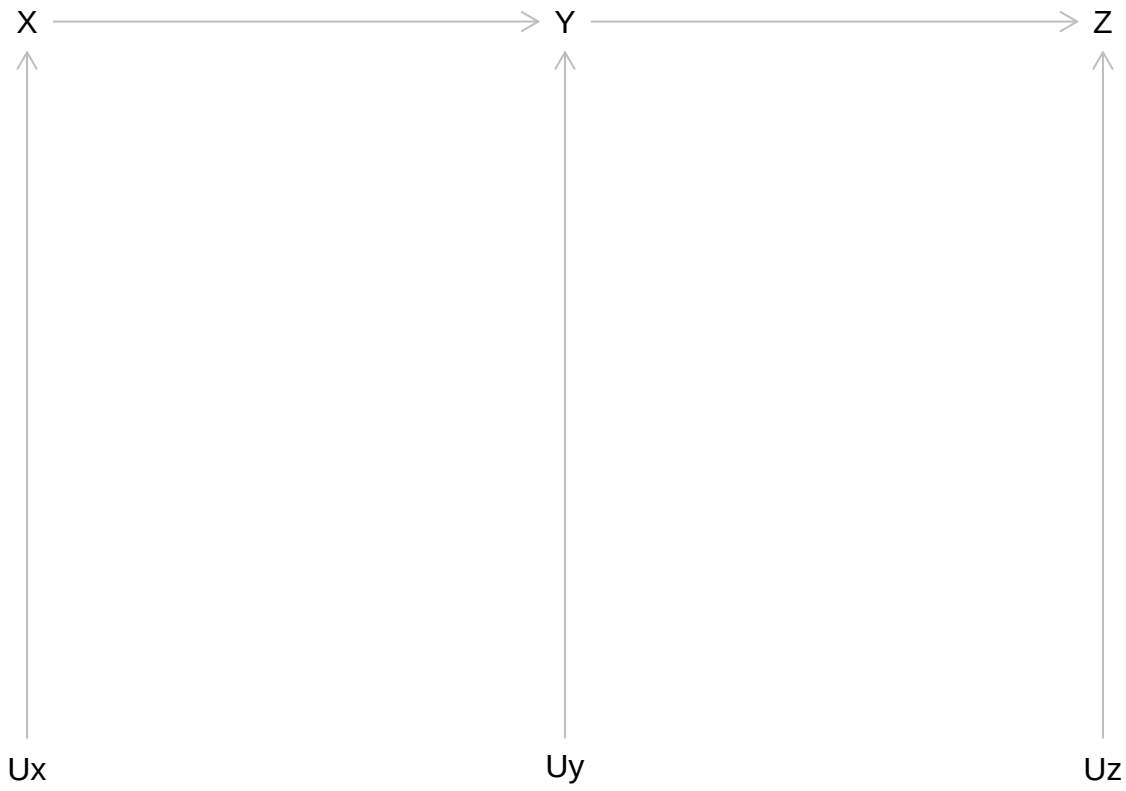
```
N <- 10000 # sample size
Ux <- rnorm( N ); Uy <- rnorm( N ); Uz <- rnorm( N )
X <- Ux
Y <- 1/3*X + Uy
Z <- 1/16*Y + Uz
d <- data.frame(X=X,Y=Y,Z=Z)
```

a) Use `dagitty` to draw the graph that complies with the model.

```
#We use this exercise to introduce another option to plot graphs in dagitty: specify the graph itself #

g <- dagitty("dag {
    Ux -> X -> Y -> Z <- Uz
    Uy -> Y
```

```
}")
coordinates(g) <- list(
    x=c(Ux=1,Uy=2,Uz=3,X=1,Y=2,Z=3),
    y=c(Ux=1,Uy=1,Uz=1,X=0,Y=0,Z=0) )
plot(g)
```




b) Determine the best guess of the value (expected value) of Z, given that we observe Y=3.

We take the parametric function for Z. In expectation the error is 0. So we just plug in Y=3.

```
#The answer should be 3/16. So the following expression should return a value close to 3/16:
3/16
```

```
## [1] 0.1875
```

```
predict(lm(Z~Y,d),list(Y=3),interval="confidence")
```

```
##         fit       lwr       upr
## 1 0.1948093 0.1355339 0.2540846
```

c) Determine the best guess of the value of Z, given that we observe X=3.

The expectations of Uy and Uz are 0, so just plug in X=3 and propagate to Z.

```
#This should be 1/16, so the following should give us a value close to 1/16
predict(lm(Z~X,d),list(X=3),interval="confidence")
```

```
##          fit         lwr       upr
## 1 0.03304151 -0.02988408 0.0959671
```

d) Determine the best guess of the value of Z, given that we observe X=1 and Y=3.

A: If we observe that Y=3, then the fact that X=1 becomes irrelevant for predicting Z. So the answer should be 3, just like in (b).

```
predict(lm(Z~X+Y,d),list(X=1,Y=3),interval="confidence")
```

```
##         fit       lwr       upr
## 1 0.1935221 0.1342266 0.2528176
```

```
##        fit      lwr      upr
## 1 2.936292 2.001441 3.871143
```

e) Determine the best guess of X, given that we observed Y=2.

```
lm(X~Y,d)
```

```
##
## Call:
## lm(formula = X ~ Y, data = d)
##
## Coefficients:
## (Intercept)            Y
##   -0.009382     0.309759
```

```
predict(lm(X~Y,d),list(Y=2),interval="confidence")
```

```
##        fit       lwr       upr
## 1 0.610136 0.5708103 0.6494617
```

```
mean(subset(d, Y > 1.9 & Y < 2.1)$X)
```

```
## [1] 0.5649688
```

This is actually quite surprising. At first sight one expects, since Y = 1/3 X, that X should be 6. However, since X is normally distributed around zero, values where X=6 are extremly rare. The effect of Uy is much more important in getting a value Y = 2

(Textor & co use the covariance matrix of X and Y)

f) Determine the best guess of Y, given that we observed X=1 and Z=3.

Finding this value requires an involved calculation, which gives an answer close to (but not equal to) 0.5. In case you wish to verify your result, here's how:

```
predict(lm(Y~X+Z,d),list(X=1,Z=3),interval="confidence")
```

```
##         fit       lwr       upr
## 1 0.5556835 0.4917326 0.6196344
```