Causal inference: how to analyse causal scenarios correctly, and repercusions of a wrong analysis

Nermina Logo Lendo, Rodrigo Jiménez and Inés García Ortiz
Bioinformatics and Computational Biology MSc
Universidad Autónoma de Madrid
2021 - 2022

January 2022

Contents

1 Objective

Causal inference is a key element in statistics. It help us reach valuable conclusions about how do variables relate to one another, and help us make decisions in order to mantain our health, combat disease, adjust habits, etc. The problem comes when data is misinterpreted, and correlation is mistaken by causalty. It is very different to say 'ice cream causes cancer' rather than 'in the same season of the year, both ice cream sales and number of melanoma diagnosis increase'. A wrong conclusion can have serious repercusions, that may go from administrating a wrong treatment to ruining the ice cream economy. Even if our field of study is not statistics, it is interesting to understand some basic concepts to prevent us from being fooled by sensational news and develop the so called 'critical thinking'. The objective of this project is to show what changes when data is modelled in the wrong way and how to interpret it correctly. The code can be accessed from the GitHub repository Causalinference - GitHub repository . Along the document, we will explain basic concepts with examples, vaguely based in real life events. It is present all the code necessary to perform each of the cases.

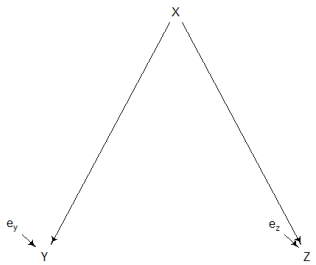
2 What to do when there is a common cause

In this section, we will cover the difficulties that may come when we want to analyse the cause of an outcome variable, Z, when it is affected by X. X is a variable that doesn't only affect Z, but also a second variable, Y: for this reason, X is a **common cause** of both X and Y. We have used the following modules to make the analysis:

```
library(dagitty)
library(car)
library(rethinking)
if(!suppressWarnings(require("rethinking",
quietly = TRUE))) {drawdag <- plot}
```

The variable Y can have an effect on X or not. This gives us two basic scenarios to work on. The first scenario is illustrated in the following DAG:

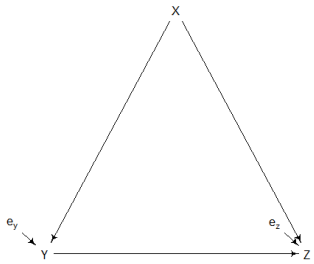
```
scenario1.DAG <- dagitty("dag {
X -> Y
X -> Z
e_y -> Y
e_z -> Z
}")

coordinates(scenario1.DAG) <- list(x = c(Y = 1, X = 2, Z
    = 3, e_y = 0.75, e_z = 2.75), y = c(Y = 3, X = 1, Z =
    3, e_y = 2.75, e_z = 2.75))

drawdag(scenario1.DAG)
```



In this first case, Y is not related to Z. The second scenario would be:

```
scenario2.DAG <- dagitty("dag {
X -> Y
X -> Z
```

```
Y -> Z
e_y -> Y
e_z -> Z}")
coordinates(scenario2.DAG) <- list(x = c(Y = 1, X = 2, Z
    = 3, e_y = 0.75, e_z = 2.75), y = c(Y = 3, X = 1, Z =
    3, e_y = 2.75, e_z = 2.75))
drawdag(scenario2.DAG)
```



In this second case, Y does have an effect over Z. It is important to tell the difference between both of them, as the correct model to apply will be different. How can we tell if our data corresponds to one scenario or another? First of all, we have to address one key problem: **how do we simulate data?** To simulate data in different scenarios, we have used two options: vectors and a function to create datsets. To use vectors is a quick method and very versatile, but if you need to change the coefficients that relate one variable to another it is necessary to create new vectors. Meanwhile, to have a function that creates data frames is useful to make different trials in which the relation between variables is maintained and the coefficients change: the main disadvantage of this method is that anytime the structure of the DAG changes, the function is no longer valid. Both methods are valuable, and each has its advantages and disadvantages. We will use data frames in this chapter and vectors in the next one to show different ways to simulate the data. This first function to create dataframes creates three columns: X, Y and Z. We can change the parameters that relate one varaible to another: if the estimate that multiplies Y for Z is 0, we are representing the scenario 1.

```
create.dataset <- function(b_yz, N = 500, b_xy = 3, b_xz
    = 3,
                                e_x = 1, e_y = 1, e_z = 1) {
  name_df <- data.frame(X = runif(N, 1, 100) + rnorm(N,
      sd = e_x))
  name_df$Y <- name_df$X * b_xy + rnorm(N, sd = e_y)
  name_df$Z <- name_df$X * b_xz + name_df$Y * b_yz +
      rnorm(N, sd = e_z)
```

```
  return(name_df)}
Ynoinfluences <- create.dataset(0)
Yinfluences <- create.dataset(-2)
non.influences <- create.dataset(0, b_xz = 0)
```

Once this is introduce, we can go over the issue that matters. How do we know if our data belongs to scenario 1 or scenario 2?

```
Y_check <- function (dataset, conflevel = 0.01) {
  colnames(dataset) <- c('X','Y','Z')
  model_with_Y <- lm(Z~X+Y, data = dataset)
  p.v.X <-(summary(model_with_Y)$coefficients['X','Pr(>|t
      |)'])
  p.v.Y <- (summary(model_with_Y)$coefficients['Y', 'Pr
      (>|t|)'])

  if ((p.v.X <= conflevel)&(p.v.Y > conflevel))
  {cat("The variable of analysis is not influenced by Y\n
      ")
    cat('See plot\n')
    scenario1.DAG <- dagitty("dag {
    X -> Y
    X -> Z
    e_y -> Y
    e_z -> Z}")
    coordinates(scenario1.DAG) <- list(x = c(Y = 1, X =
        2, Z = 3, e_y = 0.75, e_z = 2.75),y = c(Y = 3, X =
         1, Z = 3, e_y = 2.75, e_z = 2.75))
    drawdag(scenario1.DAG)
    return(invisible(1))}

  if ((p.v.X <= conflevel)&(p.v.Y <= conflevel))
  {cat("The variable of analysis is influenced by both X
      and Y\n")
    cat('See plot\n')
    scenario2.DAG <- dagitty("dag {
    X -> Y
    X -> Z
    Y -> Z
    e_y -> Y
    e_z -> Z}")
    coordinates(scenario2.DAG) <- list(x = c(Y = 1, X =
        2, Z = 3, e_y = 0.75, e_z = 2.75),y = c(Y = 3, X =
         1, Z = 3, e_y = 2.75, e_z = 2.75))
    drawdag(scenario2.DAG)
    return(invisible(2))}
```

```
  if ((p.v.X > conflevel)&(p.v.Y > conflevel))
  {cat("It seems that neither X or Y affect Z\nYou may
      want to review your working model\n")
  return(invisible(0))}

  if ((p.v.Y <= conflevel)&(p.v.X > conflevel))
  {cat('It looks like Y is related to Z, but not Z\nYou
      may want to revisit the hypothesis \'X = common
      cause of Y and Z\'')
  return(invisible(0))}
  }
```

An example of its use is:

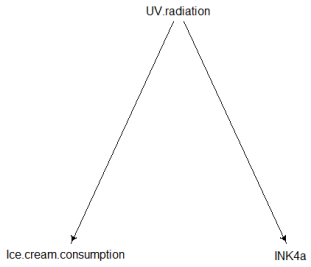
```
a <- Y_check(non.influences)

## It seems that neither X or Y affect Z
## You may want to review your working model

b <- Y_check(Yinfluences)

##The variable of analysis is influenced by both X and Y
##See plot (scenario2 DAG)
```

This function could be optimized in multiple ways, but it shows that the p value of the linear model can be used to classify a data set in one scenario or another. It is necessary to have an idea beforehand of which variable may be the common cause. If by mistake Y is actually the common cause, and X is the mediator, the function wouldn't notice it because **this couldn't be done by p-values**. This concept, of knowing the 'structure of the DAG' or how variables are related, is crucial in causal inference, and has to be based in real facts. It is similar to which came first, the hen or the egg? Which came first, the melanoma or the exposure to UV radiation?

We will now present a example to illustrate the problems of a bad modeling of scenario 1. We will study the expression of gene INK4a, key for melanoma development. It will be the outcome variable, Z. It is directly affected by UV radiation, X. UV radiation can come from sunbathing, which increases the appetite for ice cream consumption, measured in ml of consumed ice cream, Y. You collect data of potentially cancerous tissue from 100 people, from which you know the hours they have spent in the sun the last year, the amount of consumed ice cream and the expression of INK4a.

```
# Theoretical coefficeints between variables
b_xy_i_uv_i <- 5
b_xz_i_uv_i <- 10
b_yz_i_uv_i <- 0
samplesize <- 100
```

```
i_uv_i.DAG <- dagitty("dag {
UV.radiation -> Ice.cream.consumption
UV.radiation -> INK4a}")
coordinates(i_uv_i.DAG) <- list(x = c(Ice.cream.
    consumption = 1, UV.radiation = 2, INK4a = 3),y = c(
    Ice.cream.consumption = 3, UV.radiation = 1, INK4a =
    3))

drawdag(i_uv_i.DAG)
```




```
sc1.comm <- function(b_yz, N, b_xz, b_xy, reps = 100,
    ...){
  onlyY_pv <- rep(NA, reps)
  both_pv <- rep(NA,reps)
  onlyX_coefX <- rep(NA,reps)
  both_coefX <- rep(NA, reps)

  for (i in 1:reps) {
    dataset <- create.dataset(b_yz, N = N, b_xz = b_xz, b
        _xy = b_xy, ...)

    both <- lm(Z~X+Y, data = dataset)
    onlyY <- lm(Z~Y, data = dataset)
    onlyX <- lm(Z~X, data = dataset)
    onlyY_pv[i] <- summary(onlyY)$coefficients["Y", "Pr
        (>|t|)"]
    both_pv[i] <- summary(both)$coefficients["Y", "Pr(>|t
        |)"]

    onlyX_coefX[i] <- summary(onlyX)$coefficients["X", "
        Estimate"]
    both_coefX[i] <- summary(both)$coefficients["X", "
        Estimate"]
    rm(dataset)}
```

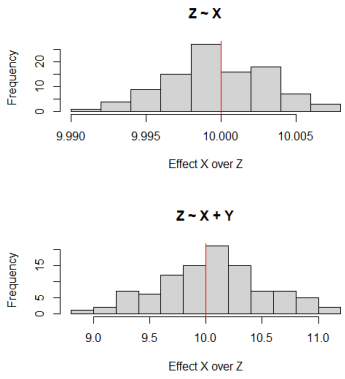
```
  cat('\n Change in relevance of Y on Z\n')
  cat('\nWhen Z ~ Y: \nThe p value of Y is ', mean(onlyY_
      pv),'\n')
  cat('\nWhen Z ~Y + X: \nThe p value of Y is ', mean(
      both_pv), '\n')
  cat('\n Change in effect of X over Z')
  cat('\nWhen Z ~ X: \nThe estimate for X is ', mean(
      onlyX_coefX),'and its s.d. is', sd(onlyX_coefX),'\n'
      )
  cat('\nWhen Z ~Y + X: \nThe estimate for X is ', mean(
      both_coefX),
       'and its s.d. is', sd(both_coefX),'\n')
  cat('\nBeing input x -> z: ', b_xz)
  #This illustrates how, even if the estimate of the
      coefficient for X is similar in both cases, the
      variance is higher in the presence of Y
  op <- par(mfrow= c(2,1), mar = rep(3,4))
  hist(onlyX_coefX, main = 'Z ~ X', xlab = 'Effect X over
       Z')
  abline(v = b_xz, col = 'red')
  hist(both_coefX, main = 'Z ~ X + Y', xlab = 'Effect X
      over Z')
  abline(v = b_xz, col = 'red')
  par(op)}
```

If we call this function with the data for this particular case, this is the output:

```
sc1.comm(b_yz = b_yz_i_uv_i, N = samplesize, b_xz = b_xz_
    i_uv_i, b_xy = b_xy_i_uv_i)
```

```
## Change in relevance of Y on Z
## When Z ~ Y:
## The p value of Y is  1.888825e-202
## When Z ~Y + X:
## The p value of Y is  0.5211384
## Change in effect of X over Z
## When Z ~ X:
## The estimate for X is  10.00005 and its s.d. is
    0.003373088
## When Z ~Y + X:
## The estimate for X is  9.982808 and its s.d. is
    0.444773
## Being input x -> z:  10
```

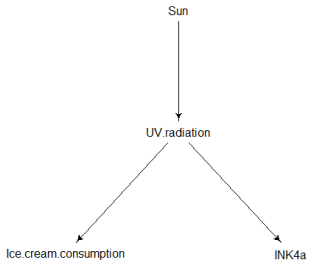
As can be seen, Y is only relevant when we are not condicioning on X. This means that if we condition on ice cream sales but not on UV radiation, we



will see association with INK4a. This association is not causation, but if it is mistaken, will result in a quite silly conclusion.

```
impliedConditionalIndependencies(i_uv_i.DAG)
## INK4 _||_ Ic.. | UV.r
\begin{lstlisting}

On the other hand, we see that the calculated coefficient
    for X, thus, UV radiation, is quite similar to the
    input estimate in both models. What is interesting to
    see is that the variance of the estimate increases
    when conditioning on Y.
In this type of situation, it is not advised to condition
    on Y, ice cream, because it won't give more
    information about melanoma and INK4a and will affect
    negatively to our estimate of UV radiation, that is a
    more interesting variable to study. It is important to
    condition on UV radiation, which is called a \textbf{
    confounder}.
One interesting variation of scenario 1, among all
    variations that can be done, is if there is a variable
    , A, that is the cause of the confounder. Should we
    condition on it? Let's see it with the example:
\begin{lstlisting}
b_ax_i_uv_i2 <- 2
b_xy_i_uv_i2 <- 5
b_xz_i_uv_i2 <- 10
b_yz_i_uv_i2 <- 0

i_uv_i_sun.DAG <- dagitty("dag {
Sun -> UV.radiation
UV.radiation -> Ice.cream.consumption
```

```
UV.radiation -> INK4a}")

coordinates(i_uv_i_sun.DAG) <- list(x = c(Ice.cream.
    consumption = 1, UV.radiation = 2, Sun = 2, INK4a = 3)
    , y = c(Ice.cream.consumption = 3, UV.radiation = 2,
    Sun = 1, INK4a = 3))
drawdag(i_uv_i_sun.DAG)
```



As the significance of ice cream, Y, was covered in the previous function, it will be skipped in this one. We are interested in knowing if the sun, A, plays a role in the value of INK4a, Z, and how does it affect X.

```
#Necessary to create another function to create a dataset
     with other structure
create.datasetv2 <- function(b_ax, b_yz=0, N = 500, b_xy
    = 3, b_xz = 3,
                                 e_x = 1, e_y = 1, e_z = 1) {
  name_df <- data.frame(A = runif(N, 1, 100) + rnorm(N))
  name_df$X <- name_df$A * b_ax + rnorm(N, sd = e_x)
  name_df$Y <- name_df$X * b_xy + rnorm(N, sd = e_y)
  name_df$Z <- name_df$X * b_xz + name_df$Y * b_yz +
     rnorm(N, sd = e_z)
  return(name_df)}

sc1.comm.plusancestor <- function(b_yz, N, b_xz, b_xy, b_
    ax, reps = 30, e_x= 1, ...){
  onlyA_pvA <- rep(NA, reps)
  bothXA_pvA <- rep(NA,reps)
  three_pvA <- rep(NA, reps)

  onlyA_coefA <- rep(NA,reps)
  bothXA_coefA <- rep(NA, reps)
  three_coefA <- rep(NA, reps)

  onlyX_coefX <- rep(NA,reps)
```

```
bothXA_coefX <- rep(NA, reps)
three_coefX <- rep(NA, reps)

onlyX_pvX <- rep(NA, reps)
bothXA_pvX <- rep(NA, reps)
three_pvX <- rep(NA, reps)

#set.seed(13) #can be uncommented for reproducibility
for (i in 1:reps) {
  dataset <- create.datasetv2(b_yz= b_yz, N = N, b_xz =
      b_xz, b_ax = b_ax,
                                  b_xy = b_xy, e_x =e_x,
                                     ...)
  three <- lm(Z~X+A+Y, data = dataset)
  bothXA <- lm(Z~X+A, data = dataset)
  onlyA <- lm(Z~A, data = dataset)
  onlyX <- lm(Z~X, data = dataset)

  onlyA_pvA[i] <- summary(onlyA)$coefficients["A", "Pr
      (>|t|)"]
  bothXA_pvA[i] <- summary(bothXA)$coefficients["A", "
      Pr(>|t|)"]
  three_pvA[i] <- summary(three)$coefficients["A", "Pr
      (>|t|)"]

  onlyA_coefA[i] <- summary(onlyA)$coefficients["A", '
      Estimate']
  bothXA_coefA[i] <- summary(bothXA)$coefficients["A",
      "Estimate"]
  three_coefA[i] <- summary(three)$coefficients["A", "
      Estimate"]

  onlyX_coefX[i] <- summary(onlyX)$coefficients["X", '
      Estimate']
  bothXA_coefX[i] <- summary(bothXA)$coefficients["X",
      "Estimate"]
  three_coefX[i] <- summary(three)$coefficients["X", "
      Estimate"]

  onlyX_pvX[i] <- summary(onlyX)$coefficients["X", "Pr
      (>|t|)"]
  bothXA_pvX[i] <- summary(bothXA)$coefficients["X", "
      Pr(>|t|)"]
  three_pvX[i] <- summary(three)$coefficients["X", "Pr
      (>|t|)"]
  rm(dataset)}
```

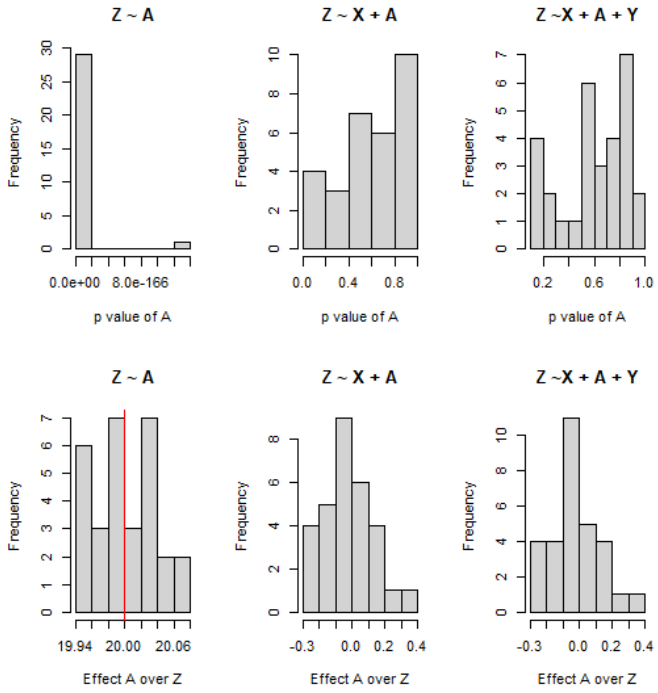
```
cat('\n____Change in p value of A on Z\n')
cat('\nWhen Z ~ A: \nThe p value of A is ', mean(onlyA_
    pvA),'\n')
cat('\nWhen Z ~ X + A: \nThe p value of A is ', mean(
    bothXA_pvA), '\n')
cat('\nWhen Z ~ Y + X + A: \nThe p value of A is ',
    mean(three_pvA), '\n')

cat('\n____Effect of A over Z\n')
cat('Input A -> X: ', b_ax,'\nInput X -> Z:', b_xz,'\
    nTotal effect A -> Z', b_xz * b_ax, '\n')
cat('\nWhen Z ~ A: \nCoefficient of A is ', mean(onlyA_
    coefA),'and its s.d. is',sd(onlyA_coefA),'\n')
cat('\nWhen Z ~ X + A: \nCoefficient of A is ', mean(
    bothXA_coefA),'and its s.d. is',sd(bothXA_coefA),'\n
    ')
cat('\nWhen Z ~ Y + X + A: \nCoefficient of A is ',
    mean(three_coefA),'and its s.d. is',sd(three_coefA),
     '\nSee plots:\n')
op <- par(mfrow= c(2,3))
hist(onlyA_pvA, main = 'Z ~ A', xlab = 'p value of A')
hist(bothXA_pvA, main = 'Z ~ X + A', xlab = 'p value of
     A')
hist(three_pvA, main='Z ~X + A + Y', xlab = 'p value of
     A')

hist(onlyA_coefA, main = 'Z ~ A', xlab = 'Effect A over
     Z')
abline(v = b_xz*b_ax, col = 'red')
hist(bothXA_coefA, main = 'Z ~ X + A', xlab = 'Effect A
     over Z')
abline(v = b_xz*b_ax, col = 'red')
hist(three_coefA, main='Z ~X + A + Y', xlab = 'Effect A
     over Z')
abline(v = b_xz*b_ax, col = 'red')
par(op)

#p value of X    ######it is very obvious that it will
    always have a significant value
#cat('\n____Change in p value of X on Z\n\nWhen Z ~ A:
    \nThe p value of X is ', mean(onlyX_pvX),'\n\nWhen Z
     ~ X +A: \nThe p value of X is ', mean(bothXA_pvX),
    '\n\nWhen Z ~ Y + X + A: \nThe p value of X is ',
    mean(three_pvX), '\n')
```

```
  cat('\n____Effect of X over Z\n\n')
  cat('Input X -> Z: ', b_xz,'\n')
  cat('\nWhen Z ~ A: \nCoefficient of X is ', mean(onlyX_
      coefX),'and its s.d. is',sd(onlyX_coefX),'\n')
  cat('\nWhen Z ~ X +A: \nCoefficient of X is ', mean(
      bothXA_coefX),'and its s.d. is', sd(bothXA_coefX),'\
      n')
  cat('\nWhen Z ~ Y + X + A: \nCoefficient of X is ',
      mean(three_coefX),'and its s.d. is', sd(three_coefX)
      ,'\nSee plots:\n')

  op <- par(mfrow= c(2,3))
  hist(onlyX_pvX, main = 'Z ~ X', xlab = 'p value of X')
  hist(bothXA_pvX, main = 'Z ~ X + A', xlab = 'p value of
       X')
  hist(three_pvX, main='Z ~X + A + Y', xlab = 'p value of
       X')

  hist(onlyX_coefX, main = 'Z ~ X', xlab = 'Effect X over
       Z')
  abline(v = b_xz, col = 'red')
  hist(bothXA_coefX, main = 'Z ~ X + A', xlab = 'Effect X
       over Z')
  abline(v = b_xz, col = 'red')
  hist(three_coefX, main='Z ~X + A + Y', xlab = 'Effect X
       over Z')
  abline(v = b_xz, col = 'red')

  par(op)}
```

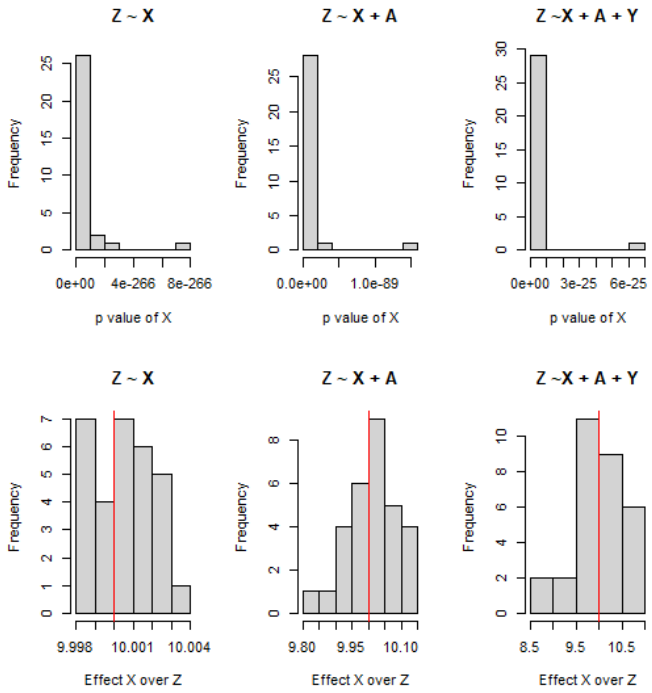
Introducing the data of the problem, the output is:

```
sc1.comm.plusancestor(b_yz = b_yz_i_uv_i2, N = samplesize
    , b_xz=b_xz_i_uv_i2, b_ax = b_ax_i_uv_i2, b_xy = b_xy_
    i_uv_i2)

#____Change in p value of A on Z
#When Z ~ A:
#The p value of A is  3.806132e-168
#When Z ~ X + A: The p value of A is  0.4372251
#When Z ~ Y + X + A:  The p value of A is  0.4176931

#____Effect of A over Z
#Input A -> X:  2
#Input X -> Z: 10
#Total effect A -> Z 20
#When Z ~ A: Coefficient of A is  20.00201 and its s.d.
```

```
    is 0.03505331
#When Z ~ X + A: Coefficient of A is  0.005650556 and its
     s.d. is 0.2130753
#When Z ~ Y + X + A: Coefficient of A is  0.005779907 and
     its s.d. is 0.2167012
#See plots:
```




```
#____Effect of X over Z
#Input X -> Z:  10
#When Z ~ X: Coefficient of X is  10.00001 and its s.d.
    is 0.002016263
#When Z ~ X + A: Coefficient of X is  10.00989 and its s.
    d. is 0.1031397
#When Z ~ Y + X + A: Coefficient of X is  10.17693 and
    its s.d. is 0.5715534 #See plots:
```

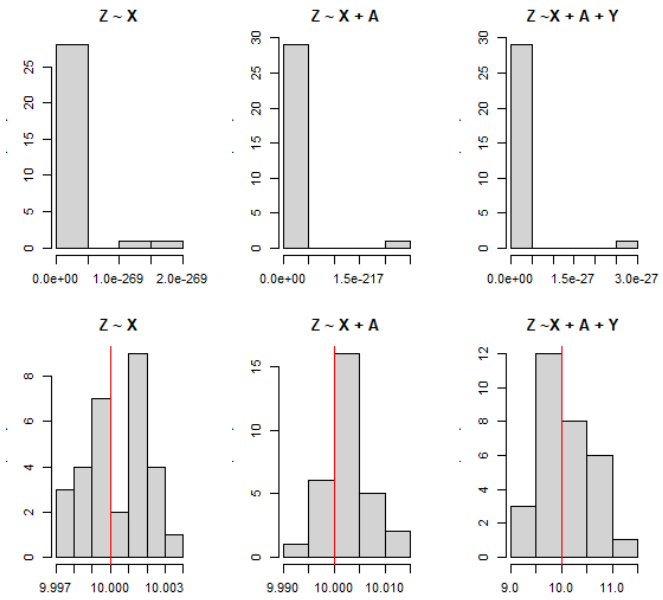


Let's first talk about A. It only has a significant p value when it is alone in the model, so the coefficients when Z X + A or Z X + A + Y are not relevant. The effect of A on Z can only be appreciated when Z A. This is supported by:

```
impliedConditionalIndependencies(i_uv_i_sun.DAG)
#INK4 _||_ Ic.. | UV.r
#INK4 _||_ Sun | UV.r
#Ic.. _||_ Sun | UV.r
```

The p value of X increases with the complexity of the model, but in any case is less significant. The estimate of X is around the expected even if complexity is increased, but its standard error gets higher. In other words, if the cause of study is UV radiation, conditioning on sun is detrimental as the variance of the coefficient increases. As seen in the cause of the cause previous work from Ramon Diaz Uriarte, when the standard error of X increases, the variance of its coefficient when Z X + A is reduced.

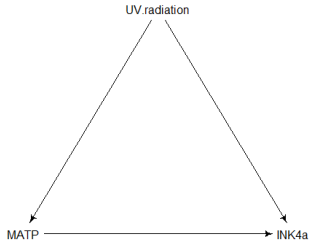
```
sc1.comm.plusancestor(b_yz = b_yz_i_uv_i2, N = samplesize
    , b_xz=b_xz_i_uv_i2, b_ax = b_ax_i_uv_i2, b_xy=b_xy_i_
    uv_i2, e_x =10)

## When Z ~ X + A: Coefficient of X is  9.999956 and its
    s.d. is 0.004412072
```


Z ~ X
Z ~ X + A
Z ~X + A + Y
0.0e+00
1.0e-269
2.0e-269
0.0e+00
1.5e-217
0.0e+00
1.5e-27
3.0e-27
Z ~ X
Z ~ X + A
Z ~X + A + Y
9.997
10.000
10.003
9.990
10.000
10.010
9.0
10.0
11.0

This function also illustrates a key property of causal inference: **same rules for simple models can be applied to more complex models**.

Let's move on to the scenario 2. We will illustrate with another example what to expect conditioning on the different possibilities. We have concluded that INK4a overexpression is caused by UV radiation. A recent study shows that there is also intervention of MATP in the French population in this process. It seems to follow the following DAG:

```
#Theoretical data
samplesize <- 100
b_xz_m_uv_i <- 4
b_yz_m_uv_i <- (-3)
b_xy_m_uv_i <- 2

m_uv_i.DAG <- dagitty("dag {
UV.radiation -> MATP
UV.radiation -> INK4a
MATP -> INK4a}")

coordinates(m_uv_i.DAG) <- list(x = c(MATP = 1, UV.
    radiation = 2, INK4a = 3),y = c(MATP = 3, UV.radiation
     = 1, INK4a = 3))
drawdag(m_uv_i.DAG)
```

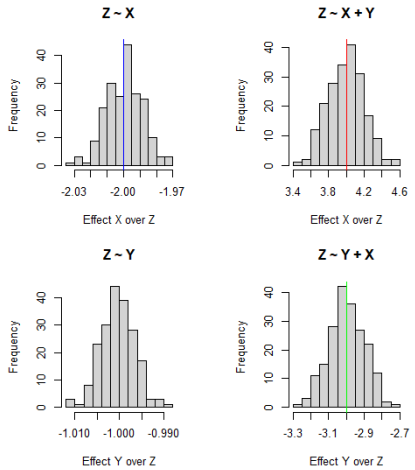


In this case, we are not just asking the question how UV radiation, X, influences INK4a, Z, but we may also be interested in how MATP, Y, affects INK4a.

```
sc2.comm <- function(b_xz, b_yz, b_xy, N, reps = 200,
    ...) {
  onlyY_pvY <- rep(NA, reps)
  both_pvY <- rep(NA, reps)
  onlyX_coefX <- rep(NA, reps)
  both_coefX <- rep(NA, reps)
  onlyY_coefY <- rep(NA, reps)
  both_coefY <- rep(NA, reps)
```

```
for (i in 1:reps){
  dataset <- create.dataset(N=N, b_xz = b_xz, b_yz = b_
      yz, b_xy = b_xy)
  both <- lm(Z~X + Y, dataset)
  onlyY <- lm(Z~Y, dataset)
  onlyX <- lm(Z~X, dataset)

  onlyY_pvY[i] <- summary(onlyY)$coefficients['Y', 'Pr
      (>|t|)']
  both_pvY[i] <- summary(both)$coefficients['Y', 'Pr(>|
      t|)']
  onlyX_coefX[i] <- summary(onlyX)$coefficients['X', '
      Estimate']
  both_coefX[i] <- summary(both)$coefficients['X', '
      Estimate']
  onlyY_coefY[i] <- summary(onlyY)$coefficients['Y', '
      Estimate']
  both_coefY[i] <- summary(both)$coefficients['Y', '
      Estimate']
  rm(dataset)}

cat('p value of Y')
cat('\nWhen Z~Y:', mean(onlyY_pvY))
cat('\nWhen Z~Y+X', mean(both_pvY),'\n\n')
cat('___Changes in X\n')
cat('When Z ~ X:\nX coefficient:', mean(onlyX_coefX), '
    s.d:', sd(onlyX_coefX))
cat('\nWhen Z ~ X + Y:\nX coefficient:', mean(both_
    coefX), 's.d:', sd(both_coefX))
cat('\n\n___Changes in Y\n')
cat('When Z ~ Y:\nY coefficient:', mean(onlyY_coefY), '
    s.d:', sd(onlyY_coefY))
cat('\nWhen Z ~ Y + X:\nY coefficient:', mean(both_
    coefY), 's.d:', sd(both_coefY))
##legend: red, direct effect X, blue total effect X,
    green effect Y
op <- par(mfrow= c(2,2))
hist(onlyX_coefX, main = 'Z ~ X', xlab = 'Effect X over
     Z')
abline(v = b_xz + b_yz*b_xy, col = 'blue')#total effect
    , it takes into account both sources of effect
abline(v = b_xz, col = 'red')#direct effect

hist(both_coefX, main = 'Z ~ X + Y', xlab = 'Effect X
    over Z')
abline(v = b_xz + b_yz*b_xy, col = 'blue')#total effect
```

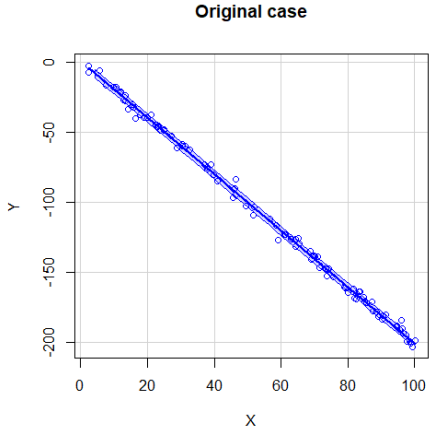
```
  abline(v = b_xz, col = 'red')#direct effect

  hist(onlyY_coefY, main = 'Z ~ Y', xlab = 'Effect X over
      Z')
  abline(v = b_yz, col = 'green')
  hist(both_coefY, main = 'Z ~ Y + X', xlab = 'Effect X
     over Z')
  abline(v = b_yz, col = 'green')
  par(op)}

sc2.comm(b_xz = b_xz_m_uv_i, b_yz = b_yz_m_uv_i, b_xy = b
    _xy_m_uv_i,
         N = samplesize)

#p value of Y
#When Z~Y: 1.157204e-134
#When Z~Y+X 7.789572e-38
#___Changes in X
#When Z ~ X: X coefficient: -2.000495 s.d: 0.01156517
#When Z ~ X + Y: X coefficient: 4.008798 s.d: 0.2082812
#___Changes in Y
#When Z ~ Y: Y coefficient: -1.00081 s.d: 0.00404317
#When Z ~ Y + X: Y coefficient: -3.004562 s.d: 0.1041344
```



In this case MATP, Y, has a significant p value in both models, in presence and absence of the common cause X, UV radiation. This makes sense and was expected. On blue it is shown the total effect of X over Z, as it takes into account the effect of X over Y as well. On red, it is shown the direct effect of X over Z. On green, the effect of Y over Z. When the mediator Y is out of the model,

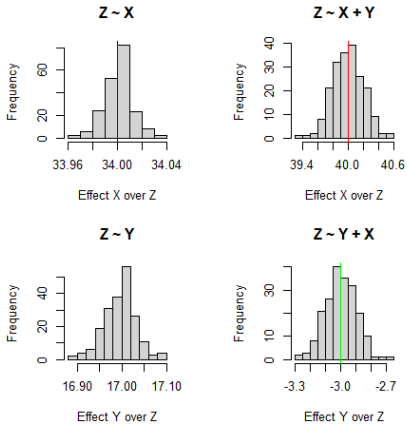
the X estimates the total effect over Z, including Y contribution. Only when Y is included it is possible to discern what is the direct effect of X. Depending on the case it would be more interesting to study the total or the direct effect of X. For this case, we argue that to know the total effect would be better because in the human body MATP expression is unavoidable. In any case, when there are two covariates in the model the variance of the estimates increase. To know the effect of Y over Z, X has to be taken into account. When UV radiation is not considered, the estimate for MATP is biased; for this reason in this kind of causal structures X is called a confounder, as in the scenario 1. To condition on Y or not may give unexpected outcomes when looking at X over Z. In this case, if MATP is not considered, it seems that the total effect is negative. If we input a higher X->Z value:

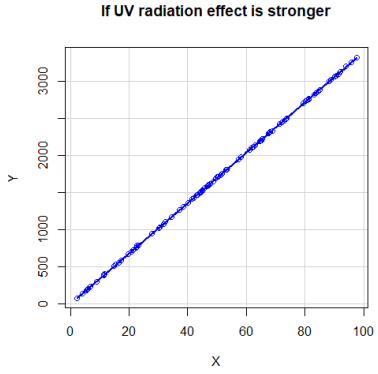
```
when_xz_4 <- create.dataset(b_xz = b_xz_m_uv_i, b_yz = b_
    yz_m_uv_i, b_xy = b_xy_m_uv_i,N = samplesize)
scatterplot(Z~X, data = when_xz_4, main ='Original case',
     regLine=TRUE)
```




```
sc2.comm(b_xz = b_xz_m_uv_i*10, b_yz = b_yz_m_uv_i, b_xy
    = b_xy_m_uv_i, N = samplesize)
when_xz_40 <- create.dataset(b_xz = b_xz_m_uv_i*10, b_yz
    = b_yz_m_uv_i, b_xy = b_xy_m_uv_i, N = samplesize)
scatterplot(Z~X, data = when_xz_40, main = 'If UV
    radiation effect is stronger', regLine=TRUE)

#p value of Y
#When Z~Y: 1.620428e-161
#When Z~Y+X 2.566037e-43
#___Changes in X
```

```
#When Z ~ X: X coefficient: 34.00068 s.d: 0.01156865
#When Z ~ X + Y: X coefficient: 39.99528 s.d: 0.2036934
#___Changes in Y
#When Z ~ Y: Y coefficient: 16.99545 s.d: 0.03612757
#When Z ~ Y + X: Y coefficient: −2.997494 s.d: 0.1018777
```

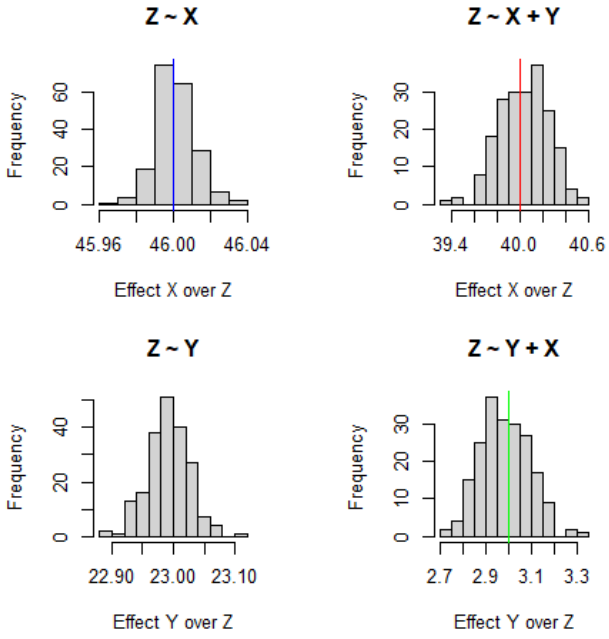




This is because the X contribution has a higher impact over Z than Y in this second case. The estimate for X in the simpler model isn't negative, it's total effect is lower than the direct effect. If the Y -> Z value wasn't negative:

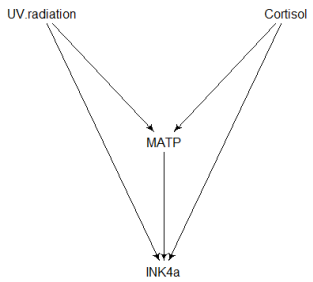
```
sc2.comm(b_xz = b_xz_m_uv_i*10, b_yz = b_yz_m_uv_i*(-1),
    b_xy = b_xy_m_uv_i, N = samplesize)
when_xz_40andnegative <- create.dataset(b_xz = b_xz_m_uv_
    i*10, b_yz = b_yz_m_uv_i*(-1), b_xy = b_xy_m_uv_i,N =
    samplesize)
scatterplot(Z~X, data = when_xz_40andnegative, main = 'If
    MATP enhances INK4a', regLine=TRUE)
```

```
#p value of Y
#When Z~Y: 2.258331e-173
#When Z~Y+X 1.924975e-41
#___Changes in X
#When Z ~ X: X coefficient: 46.00096 s.d: 0.0106993
#When Z ~ X + Y: X coefficient: 40.03247 s.d: 0.213261

#___Changes in Y
#When Z ~ Y: Y coefficient: 22.9915 s.d: 0.03411146
#When Z ~ Y + X: Y coefficient: 2.983863 s.d: 0.1064528
```



Then the effect of X, UV radiation, over Z, INK4a, is increased, as it should be obvious. The study goes on and we discovers that both the expression of MATP and INK4a is also influenced by another key factor, the cortisol level. This leaves us with the following DAG:

```
samplesize <- 100
b_by_m_uv_i_c <- 3
b_bz_m_uv_i_c <- 2
b_xz_m_uv_i_c <- 4
b_xy_m_uv_i_c <- 2
b_yz_m_uv_i_c <- (-3)

m_uv_i_c.DAG <- dagitty("dag {
UV.radiation -> MATP
UV.radiation -> INK4a
Cortisol -> MATP
Cortisol -> INK4a
MATP -> INK4a}")

coordinates(m_uv_i_c.DAG) <- list(x = c(UV.radiation = 1,
    MATP = 2, INK4a = 2, Cortisol = 3), y = c(UV.
    radiation = 1, MATP = 2, INK4a = 3, Cortisol = 1))
drawdag(m_uv_i_c.DAG)
```

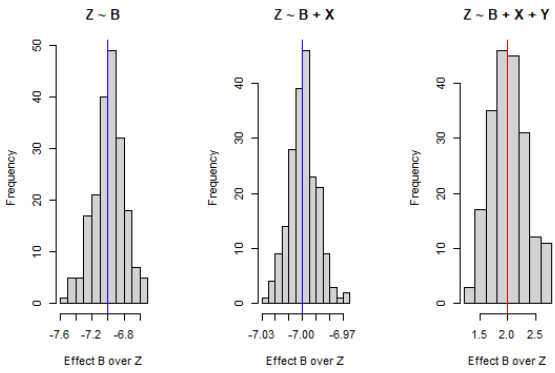


From the previous function we have learnt that the condition on the mediator variable Y, MATP, would allow us to know the direct effect of UV.radiation. When it is not present in the model, what we can see is the total effect. The reasoning behing the UV.radiation-MATP-INK4a set also apply to the Cortisol-MATP-INK4a set.

```
create.datasetv3 <- function(b_by, b_bz, b_yz=(-3), N =
    500, b_xy = 3, b_xz = 3, e_x = 1, e_y = 1, e_z = 1, e_
    b = 1) {
  name_df <- data.frame(B = runif(N, 1, 100) + rnorm(N,
      sd = e_b))
  name_df$X <- runif(N, 1, 100) + rnorm(N, sd = e_x)
```

```
  name_df$Y <- name_df$X * b_xy + name_df$B * b_by +rnorm
      (N, sd = e_y)
  name_df$Z <- name_df$X * b_xz + name_df$Y * b_yz +
    name_df$B * b_bz+ rnorm(N, sd = e_z)
  return(name_df)}

sc2.comm.extraoverY <- function(b_by, b_bz, b_xz, b_xy, b
    _yz, N, reps = 200, ...) { onlyB_coefB <- rep(NA,reps)
  bothBX_coefB <- rep(NA, reps)
  three_coefB <- rep(NA,reps)

  for (i in 1:reps){
    dataset <- create.datasetv3(N=N, b_xz = b_xz, b_yz =
        b_yz, b_xy = b_xy, b_by = b_by, b_bz = b_bz, ...)
    onlyB <- lm(Z ~ B, dataset)
    bothBX <- lm(Z ~ X + B, dataset)
    three <- lm(Z ~ Y + X + B, dataset)

    onlyB_coefB[i] <- summary(onlyB)$coefficients['B', '
        Estimate']
    bothBX_coefB[i] <- summary(bothBX)$coefficients['B',
        'Estimate']
    three_coefB[i] <- summary(three)$coefficients['B', '
        Estimate']
    rm(dataset)}

  cat('___Changes in B\n')
  cat('When Z ~ B:\nB coefficient:', mean(onlyB_coefB), '
      s.d:', sd(onlyB_coefB))
  cat('\nWhen Z ~ X + B:\nB coefficient:', mean(bothBX_
      coefB), 's.d:', sd(bothBX_coefB))
  cat('\nWhen Z ~ Y + X + B:\nB coefficient:', mean(three
      _coefB), 's.d:', sd(three_coefB))
  #legend: blue, total efffect X, red direct effect X,
      green effect Y
  op <- par(mfrow= c(1,3), mar = rep(4,4)
  hist(onlyB_coefB, main = 'Z ~ B', xlab = 'Effect B over
      Z')
  abline(v = b_bz, col = 'red')#direct effect
  abline(v = b_bz + b_yz*b_by, col = 'blue')#total effect
      of B
  hist(bothBX_coefB, main = 'Z ~ B + X', xlab = 'Effect B
      over Z')
  abline(v = b_bz, col = 'red')#direct effect
  abline(v = b_bz + b_yz*b_by, col = 'blue')#total effect
      of B
```

```
  hist(three_coefB, main = 'Z ~ B + X + Y', xlab = '
      Effect B over Z')
  abline(v = b_bz, col = 'red')#direct effect
  abline(v = b_bz + b_yz*b_by, col = 'blue')#total effect
       of B
  par(op)}

sc2.comm.extraoverY(b_by = b_by_m_uv_i_c, b_bz = b_bz_m_
    uv_i_c,
                    b_xz = b_xz_m_uv_i_c, b_xy = b_xy_m_
                        uv_i_c,
                    b_yz = b_yz_m_uv_i_c, N = samplesize
                        )
#___Changes in B
# When Z ~ B: B coefficient: -6.988085 s.d: 0.1896504
# When Z ~ X + B: B coefficient: -6.999463 s.d:
    0.01037498
# When Z ~ Y + X + B: B coefficient: 1.999542 s.d:
    0.3236159
```



The total effect of cortisol, B, is well reflected when it is on its own in the model or when UV radition, X, is considered, because they are independent from each other as can be seen in:

```
impliedConditionalIndependencies(m_uv_i_c.DAG)
# Crts _||_ UV.r
```

The variance of the coefficient when it is found along X is smaller than when B, cortisol, is checked on its own or when it also considers the collider Y, MATP. Therefore in this type of graph it would be prefered to consider both B and X on the model: the variance of the estimates is smaller and the total effect is calculated. As in the previous case, condiotioning on Y may be counterproductive. The absence of unmeasured confounding for the influence of both the exposure and the intermediate variable on the outcome is required for estimating direct

effects. If both of these requirements are not satisfied, no approach can offer unbiased estimates of exposure's direct effects.