Causal Inference - Assignment Part1

Emilia Löscher 8470014 Kyuri Park 5439043

06 March, 2022

1. Draw a DAG representing a simple and fairly plausible causal system from your preferred topic of choice. Describe briefly the substantive motivation behind your DAG.

Many people with Obsessive-compulsive disorder (OCD) become depressed. According to Millet et al. (2004), the lifetime rates of major depression in OCD patients is about 81.2%. Several studies attempted to identify the causal mechanism in the comorbidity of OCD and depression (McNally et al.2017; Zandberg et al. 2015) and speculated that OCD symptoms often precede and correspondingly activate the depression symptoms.

Given this background, here we proceed to look into the causal relationship between the symptoms of OCD and depression. We hypothesize that *distress associated with obsession* (*'ocdis'*) causes *feeling of guilt* (*'guilt'*) via several paths. Having been inspired by the DAG model from McNally et al. (2017), we construct our DAG with total 7 variables (nodes), which consist of OCD symptoms as well as depression symptoms. The specifics of the variables in our DAG is as follows:

< OCD symptoms >

- **ocdis**: distress caused by obsessions/compulsions
- **ocint**: interference due to obsessions/compulsions
- **occon**: difficulty controlling obsessions/compulsions

< Depression symptoms >

- **sad**: sadness
- **insom**: insomnia/sleeping problems
- **concen**: concentration/decision-making impairment
- **guilt**: guilt and self-blame

See *Figure 1* below for our DAG.[1]

[1] ***Note:*** All the code for the figures can be found in the *Appendix* at the end of the document.

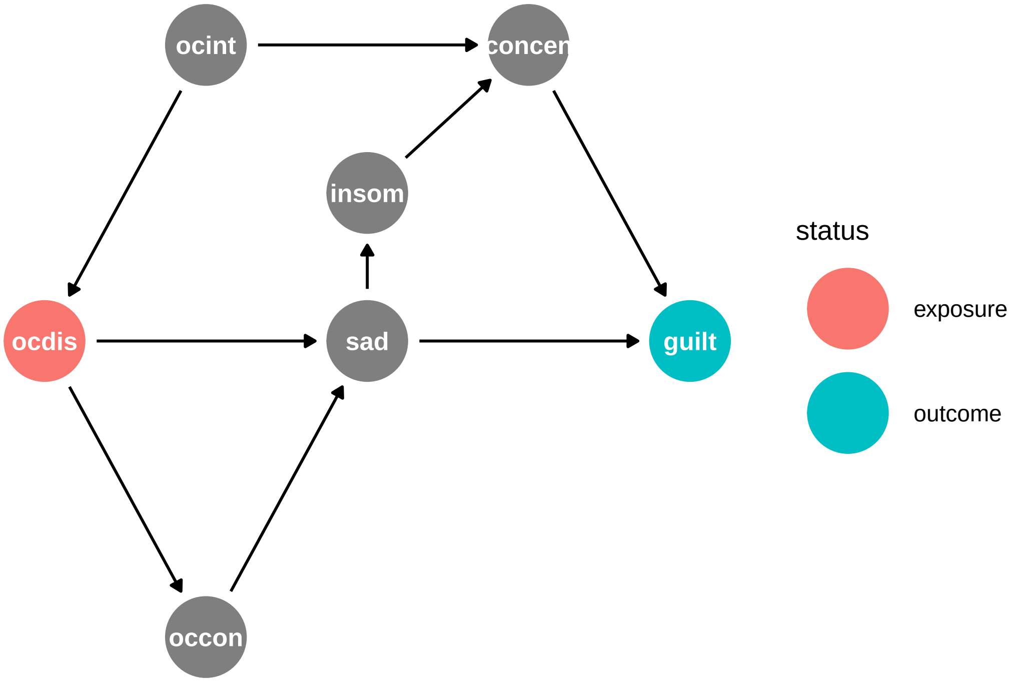

Figure 1: OCD-Depression DAG

2. Specify a structural causal model for your DAG. Assume all relationships between the variables are linear, and all the variables have normally distributed residuals.

In the following, the structural causal model for our DAG is specified:

$$
\begin{aligned}
&ocint := 2.69 + \epsilon_{ocint}\\
&ocdis := 2.81 + 3.52 \cdot ocint + \epsilon_{ocdis}\\
&occon := 2.67 + 3.38 \cdot ocdis + \epsilon_{occon}\\
&sad := 1.55 + 4.33 \cdot ocdis + 2.98 \cdot occon + \epsilon_{sad}\\
&insom := 0.81 + 2.17 \cdot sad + \epsilon_{insom}\\
&concen := 1.48 + 2.54 \cdot ocint + 3.46 \cdot insom + \epsilon_{concen}\\
&guilt := 1.56 + 1.57 \cdot concen + 3.31 \cdot sad + \epsilon_{guilt},
\end{aligned}
$$

where $\epsilon_{ocint}, \ldots, \epsilon_{guilt} \overset{iid}{\sim} \mathcal{N}(0, \sigma_i^2)$ with $i \in ocint, \ldots, guilt$ and $\sigma_i \in \{0.82, 0.76, 0.76, 0.94, 1.07, 0.87, 1.17\}$.

We base our estimates of the intercepts and residual standard deviations (SD) on the data from McNally et al. (2017). They interviewed 408 patients about their OCD and depression symptoms. Intercepts and residual SDs are set to the mean and SD values of the corresponding symptom item that they found. The regression coefficients are decided such that it could represent the relative strength (effect) of each symptom we have thought of, considering the size of SD for each item.

3. Generate data from your SCM with a sample size of 500 units.

```r
## set the seed
set.seed(1000)

## sample size (n) = 500
n <- 500

## generate the data
ocint <- 2.69 + rnorm(n, 0, 0.82)
ocdis <- 2.81 + 3.52 * ocint + rnorm(n, 0, 0.76)
occon <- 2.67 + 3.38 * ocdis + rnorm(n, 0, 0.76)
sad <- 1.55 + 4.33 * ocdis + 2.98 * occon + rnorm(n, 0, 0.94)
insom <- 0.81 + 2.17 * sad + rnorm(n, 0, 1.07)
concen <-  1.48 + 2.54 * ocint + 3.46 * insom + rnorm(n, 0, 0.87)
guilt <- 1.56 + 1.57 * concen + 3.31 * sad + rnorm(n, 0, 1.17)

## create a data frame called "OCDDEP"
OCDDEP <- data.frame(ocint, ocdis, occon, sad, insom, concen, guilt)
```

4. Use the PC-algorithm on the generated data to 'discover' the structure of your causal system

4a. Is your true DAG covered in the Markov equivalence class provided by the algorithm?

No, as seen in *Figure 2*, our true DAG is not covered in the Markov equivalence class. One of the edges is missing (i.e., *ocint* → *concen*). Among the given Markov equivalence class set, DAG *(l)* – the one in the right bottom in *Figure 2* – is the closest to the true DAG structure, as it has all edges with the correct direction, except for the one missing edge between *ocint* → *concen*.

4b. Provide the CP-DAG. To what extent did the procedure correctly recover which relationship were absent/present/directed?

As shown in *Figure 3* (CP-DAG), the procedure correctly recovered most of the relationships (edges) present in our true DAG. In other words, the procedure estimated the skeleton of the true DAG pretty well. The only relationship that was absent in the CP-DAG was the one connecting *ocint* and *concen*. Yet, the procedure did not identify the complete set of colliders in the true DAG and correspondingly failed to orient some of the edges. It was successful in doing so for *guilt* (*sad* → *guilt* ← *concern*). That is, out of the three colliders (i.e., *sad*, *concen*, *guilt*) in the true DAG, only *guilt* has been successfully recognized by the procedure.

We assume the reason that the procedure failed to recover the entire true DAG is mainly due to the small sample size (i.e., $n = 500$), which resulted in lack of power. Based on trial and error, it was revealed that the procedure was able to pick the correct DAG structure including the direction of the edges, when n approached 2500. Therefore, we conclude that the appropriate sample size to estimate the causal effects for our DAG model with 7 nodes is probably around 2500 (minimum), and in this case there were simply not enough samples to properly identify the whole structure of the true DAG.

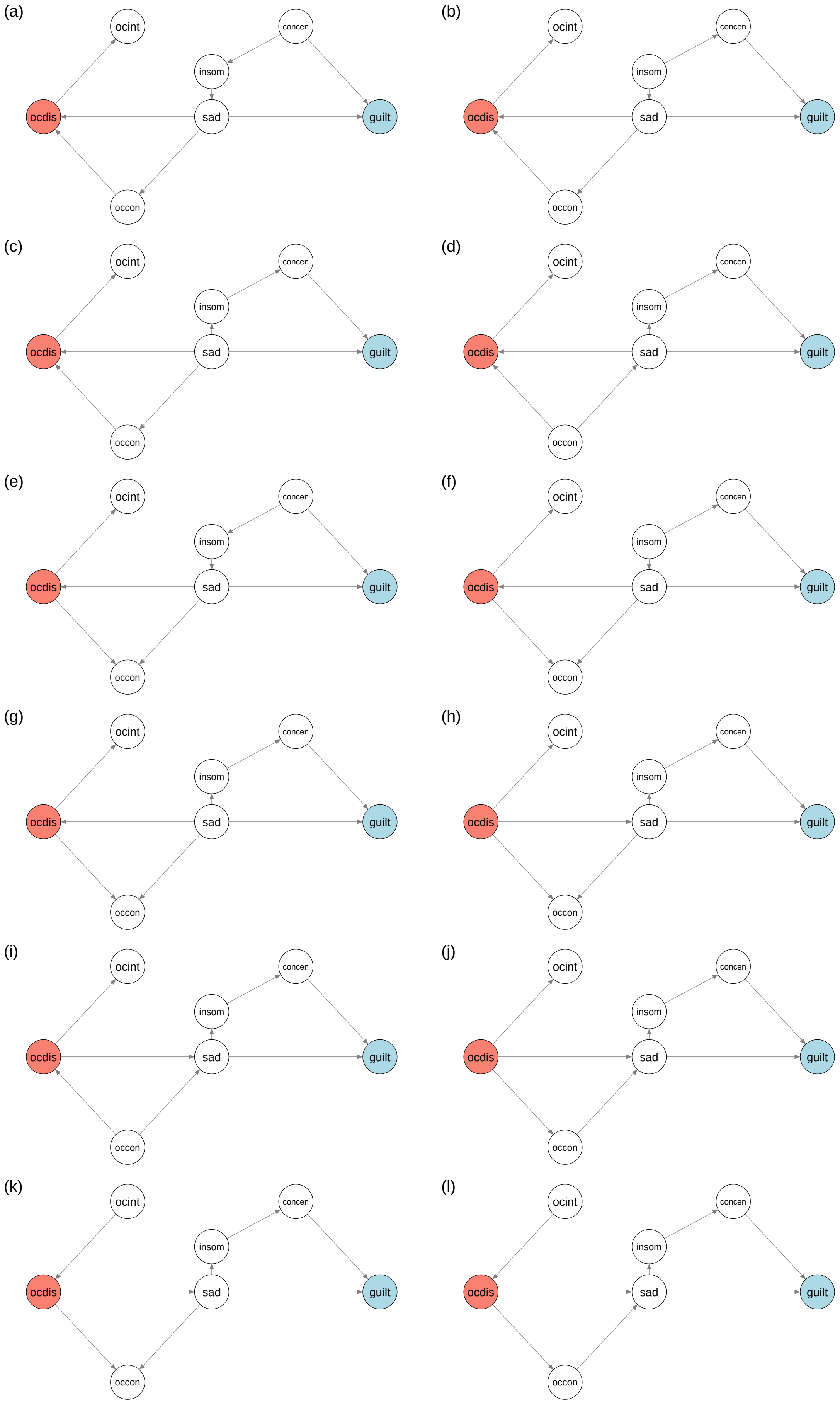

Figure 2: Estimated Markov-Equivalence Class

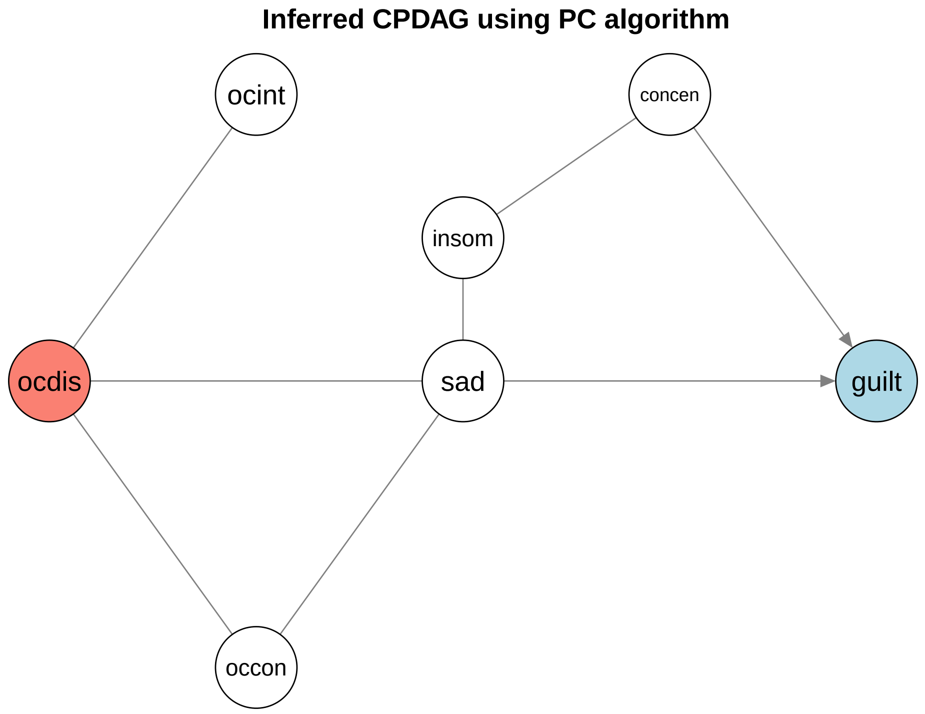

Figure 3: Completed Partially Directed Acyclic Graph (CPDAG)

5. Choose two variables for which they should estimate the causal relationship

5a. Cause variable: `ocdis`
5b. Outcome variable: `guilt`

6. For the causal effect specified in 5:

6a. What is the true causal effect of the cause on the outcome variable based on your SCM?

We use two different ways to find the true causal effect:
1) Summing the effects of all valid paths between *ocdis* and *guilt*.
2) Mimicking the intervention on *ocdis*: fixing *ocdis* to a certain value and see the change in *guilt* when increasing *ocdis* by one point.
As seen below, the true causal effect for both methods are found to be around *217*.

– **Method 1**) There are total four valid causal paths between *ocdis* and *guilt*. And summing the effect of each valid path would give us the true causal effect estimate.

The valid causal paths are listed below:

- path1) $ocdis \rightarrow sad \rightarrow guilt$
- path2) $ocdis \rightarrow occon \rightarrow sad \rightarrow guilt$
- path3) $ocdis \rightarrow sad \rightarrow insom \rightarrow concen \rightarrow guilt$
- path4) $ocdis \rightarrow occon \rightarrow sad \rightarrow insom \rightarrow concen \rightarrow guilt$

```r
## Method 1: Analytical solution
# path 1
path1 <- 4.33 * 3.31
# path 2
path2 <- 3.38 * 2.98 * 3.31
# path 3
path3 <- 4.32 * 2.17 * 3.46 * 1.57
# path 4
path4 <- 3.38 * 2.98 * 2.17 * 3.46 * 1.57

# true causal effect
TCE1 = path1 + path2 + path3 + path4
cat("True causal effect =", TCE1)
```

```
## True causal effect = 217.3277
```

– **Method 2**) For this method, we compute the true causal effect by taking the difference between the expected value of *guilt* when *ocdis* is set to 12 and when it is one point higher, at 13 (it is not necessary but we set it to 12, because the mean value of *ocdis* is around 12 in the generated data). This is essentially mimicking an intervention by fixing *ocdis* to a certain value and and check the corresponding effect on *guilt* when *ocids* is increased by 1 point, while the other variables are held constant.

```r
## Method 2: Mimicking intervention on ocdis
# 'ocdis' is set to 12
ocint2 <-  2.69
ocdis2 <- 12     # fix "ocdis" to 12
occon2 <- 2.67 + 3.38 * ocdis2
sad2 <- 1.55 + 4.33 * ocdis2 + 2.98 * occon2
insom2 <- 0.81 + 2.17* sad2
concen2 <-  1.48 + 2.54 * ocint2 + 3.46 * insom2
guilt2 <- 1.56 + 1.57 * concen2 + 3.31 * sad2

# 'ocdis' is set to 13: one point higher
ocint3 <-  2.69
ocdis3 <- 13     # fix "ocdis" to 13
occon3 <- 2.67 + 3.38 * ocdis3
sad3 <- 1.55 + 4.33 * ocdis3 + 2.98 * occon3
insom3 <- 0.81 + 2.17* sad3
concen3 <-  1.48 + 2.54 * ocint3 + 3.46 * insom3
guilt3 <- 1.56 + 1.57 * concen3 + 3.31 * sad3

# true causal effect = change in the expected value of 'guilt' when 'ocdis' increases by 1
TCE2 <- guilt3 - guilt2
cat("True causal effect =", TCE2)
```

```
## True causal effect = 217.4456
```

6b. Based on the true DAG, what linear regression model should be used to estimate the causal effect correctly?

```
# find variables needed to be controlled for.
adjustmentSets(ocddep)
```

```
## { ocint }
```

We need to block the backdoor path by controlling for *ocint* in the regression model.
The linear regression model to estimate the causal effect of *ocdis* on *guilt* can, therefore, be written as:

$$guilt_i = \beta_0 + \beta_1 ocdis_i + \beta_2 ocint_i + \epsilon_i, \text{ with } \epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2), \; \forall\, i = 1, \dots, 500$$

Note: here the σ is the residual standard deviation that is different from all of the σ_i specified in the SCM from Q2.

6c. Estimate the causal effect with this regression model based on your generated data. To what extent is the true effect recovered?

Based on the linear regression model specified in *6b*, the causal effect of *ocdis* on *guilt* is estimated to be around 216 as shown below. Given that the true causal effect is approximately 217, it is concluded that the true causal effect is mostly recovered. This slight discrepancy between the estimated and true causal effect can be even reduced by increasing the sample size (e.g., with the sample size of 3000 for example, the causal effect of *ocdis* on *guilt* is estimated to be 217, which is the same as true causal effect).

```
# linear model estimating causal effect of 'ocdis' on 'guilt' while controlling for 'ocint'
mod <- lm(guilt ~ ocdis + ocint, data = OCDDEP)
round(summary(mod)$coefficients, 4)
```

```
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 152.1915     8.4568  17.9963    0.000
## ocdis       216.1705     2.1567 100.2325    0.000
## ocint         9.8732     7.9526   1.2415    0.215
```

7. For assignment Part II, we will need a dichotomous cause variable.

7a. Make a dichotomized version of the cause variable from question 5 & 6 in your dataset, for example by assigning scores lower than the mean a 0 and the rest a 1.

We dichotomized *ocdis* based on its mean value in such a way that assigning *0* for the scores *lower* than the mean, and assigning *1* for the scores *higher* than the mean. See *Figure 4* for the distribution of the dichotomized variable.

```r
# dichotomized version of 'ocdis' = 'dicho_ocdis'
OCDDEP$dicho_ocdis <- ifelse(ocdis < mean(ocdis), 0, 1)
# the number of observations in each category
table(OCDDEP$dicho_ocdis)
```

```
##
##   0   1
## 254 246
```

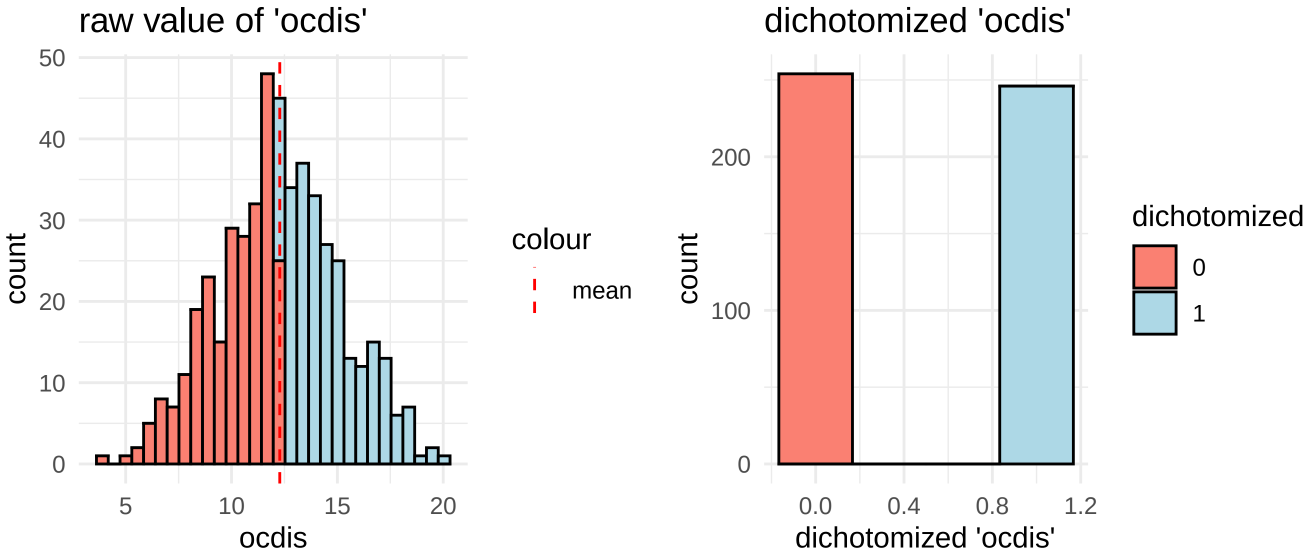

Figure 4: Dichotomizing 'ocdis' based on the mean value

7b. Use the correct model for the causal effect (the one from 6b) to estimate the causal effect again, but now with your dichotomized cause variable. Discuss the results.

As shown below, when we use the dichotomized cause variable (`dicho_ocdis`), then the causal effect of `ocdis` on `guilt` is estimated to be around 174, which is smaller than the estimated causal effect in *6c* with the continuous cause variable. Also, the estimate of the average effect of `ocint` (confounder) on `guilt` becomes larger, and it correspondingly becomes significant in this model. Hence, it is concluded that when dichotomizing the cause variable, the true causal effect is somewhat underestimated, while the effect of the variable being controlled for (e.g., confounder) is overestimated.

```
# linear model with the dichotomized 'ocdis'
mod_dichotomized <- lm(guilt ~ dicho_ocdis + ocint, data = OCDDEP)
round(summary(mod_dichotomized)$coefficients, 4)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 885.6599    32.6305 27.1421        0
## dicho_ocdis 174.3989    22.4867  7.7556        0
## ocint       690.2090    14.6139 47.2297        0
```

8. Prepare the data for the second causal inference assignment.

8a. Create a data frame with all your variables, including the continuous and the dichotomized versions of your cause variable. Save it in an .Rdata file.

```
# create a RData file
save(OCDDEP, file = "5439043_8470014.RData")
```

8b. Make a brief .txt file with in it a description what each of the variables in the dataframe is.

See `5439043_8470014.txt`.

References

McNally, R.J., Mair, P., Mugno, B.L., & Riemann, B.C. (2017). Co-Morbid Obsessive–Compulsive Disorder and Depression: A Bayesian Network Approach. *Psychological Medicine. 47*(7): 1204–14. https://doi.org/10.1017/S0033291716003287.

Millet, B., Kochman, F., Gallarda, T., Krebs, M.O., Demonfaucon, F., Barrot, I., Bourdel, M.C., Olie, J.P., Loo, H., & Hantouche, E.G. (2004). Phenomenological and Comorbid Features Associated in Obsessive–Compulsive Disorder: Influence of Age of Onset. *Journal of Affective Disorders. 79*(1-3): 241–46.

Zandberg, L.J., Zang, Y., McLean, C.P., Yeh, R., Simpson, H.B., & Foa, E.B. (2015). Change in Obsessive-Compulsive Symptoms Mediates Subsequent Change in Depressive Symptoms During Exposure and Response Prevention. *Behaviour Research and Therapy. 68*:76–81.

Appendix

Code for Figure 1

```
## plot our DAG using 'dagify'
ocddep <- dagify(
  ocdis ~ ocint,
  occon ~ ocdis,
  sad ~ ocdis + occon,
  insom ~ sad,
  concen ~ ocint + insom,
  guilt ~ concen + sad,
  exposure = "ocdis", # cause variable we are interested in
  outcome = "guilt", # effect variable we are interested in
  # set the coordinates
  coords = list(x = c(ocdis = -1, ocint=-0.5, occon=-0.5, sad = 0,
                      insom=0, concen=0.5, guilt = 1),
              y = c(ocdis = 0, ocint=1, occon=-1, sad = 0, insom=0.5,
                    concen=1, guilt = 0))
)

ggdag_status(ocddep) + theme_dag()
```

Code for PC-algorithm (Figure 2)

```
## sufficient statistics: correlation matrix & sample size
suffStat <- list(C= cor(OCDDEP), n = nrow(OCDDEP))

## run the pc algorithm
varnames <- colnames(OCDDEP)
pc_fit <- pc(suffStat = suffStat, indepTest = gaussCItest, alpha = 0.01,
              labels = varnames)

## plot the markov equivalence class
# extract the adjacency matrix of the cpdag
cpdag_mat <- as(pc_fit, "matrix")

# extract the DAG adjacency matrix in a vector form (by rows)
res1 <- pdag2allDags(cpdag_mat)

# get the adjacency matrix of an individual DAG
res1_dags <- list()
for(i in 1 :nrow(res1$dags)){
  res1_dags[[i]] <- t(matrix(res1$dags[i,], 7, 7, byrow = T))
}

# specify a layout matrix
laymat <- rbind(c(-0.5, 1),
                c(-1, 0),
                c(-0.5, -1),
                c(0, 0),
                c(0, 0.5),
                c(0.5, 1),
                c(1, 0))


# plot the DAG for each adj.matrix
par(mfrow=c(6,2))
for (i in 1:length(res1_dags)){
  qgraph(res1_dags[[i]], bidirectional=TRUE, color=c("white","salmon", "white",
                                                 "white", "white", "white", "lightblue"),
       layout = laymat, labels = varnames, asize = 5, vsize =14,
       title = paste0("(",letters[i],")"), title.cex=3)
}
```

Code for Figure 3

```
## plot the CP-DAG
# adjust the plotting margin
par(mar=c(5, 4, 3.5, 2) + 0.1)

# plot the CP-DAG using qgraph
qgraph(pc_fit, color=c("white","salmon", "white",
                                                 "white", "white", "white", "lightblue"),
       layout = laymat, labels = varnames)
title("Inferred CPDAG using PC algorithm", line=3, cex.main = 1)
```

Code for Figure 4

```
## plot the histogram of 'ocdis' and the reference line at the mean value
p1 <- ggplot(OCDDEP, aes(ocdis, fill=ocdis < mean(ocdis))) +
  geom_histogram(color="black", show.legend = FALSE) +
  scale_fill_manual(values=c("lightblue", "salmon")) +
  ggtitle("raw value of 'ocdis'") +
  geom_vline(aes(xintercept = mean(ocdis), colour = "mean"),linetype = "dashed") +
  scale_color_manual(values = c("mean" = "red")) +
  theme_minimal()
```

```r
## plot the dichotomized version of 'ocdis' based on the mean value
p2 <- ggplot(OCDDEP, aes(dicho_ocdis, fill=as.factor(dicho_ocdis))) +
  geom_histogram(color="black",binwidth=1/3, show.legend = TRUE) +
  scale_fill_manual(values=c("salmon", "lightblue")) +
  ggtitle("dichotomized 'ocdis'") +
  labs(fill='dichotomized', x = "dichotomized 'ocdis'") +
  theme_minimal()

## put two plots (p1 & p2) together side by side
ggarrange(p1, p2, ncol =2)
```