Causal Relationships in Observational Data - Simulations Benchmark

Claudinei Biazoli

February 2022

This report is supplementary to the manuscript entitled “Causal Relationships in Longitudinal Observational Data: An Integrative Modelling Approach”. Here, we report the outline of the simulated scenarios used to test the causal discovery properties of the integrative approach alongside findings of conventional multiple regression. All codes, including the complete simulations scripts and the application to the Millenium Cohort Study, are available at osf.io/pmw6c/.

The algorithm is specified in the function bellow and follow the steps:

1 - train a model in part of the sample using the XGBoost algorithm

2 - use the model to predict the outcome for each individual in the remaining of the sample (a cross-validation procedure has to be added when analyzing empirical data)

3 - calculate the prediction errors for each individual prediction defined as the absolute difference between predicted and observed scores

```
gcxboost<-function(df,predictors_cols,outcome_col){
d<-as.matrix(df)
bst<-xgboost(data=d[1:n_train,predictors_cols],label =d[1:n_train,outcome_col],max.depth=6, eta=0.3, nrounds=100
,objective="reg:squarederror")
pred<-predict(bst,d[(n_train+1):(n_train+n_test),predictors_cols])   #### ESSE PODE SER MEU PRIOR!!!
err<-abs(d[(n_train+1):(n_train+n_test),outcome_col]-pred)
return(err)
}

boot_test<-function(vector1,vector2,R){   ### Alternative hypothesis :  VECTOR1 > VECTOR2
  l<-length(vector1)
  D<-array(0,R)
  D_org<-mean(vector1-vector2)
  for(i in 1:R){
    resample<-2*rbinom(l,1,0.5)-1
    D[i]<-mean(resample*(vector1-vector2))
  }
 p<-length(which(D>D_org))/R
  return(p)
}
```

Then, as implemented in the main code for each simulated scenario:

4 - Perform steps 1-3 with the whole set of predictors (X)

5 - Perform steps 1-3 with the whole set of predictors except one (X-i)

6 - Use a non-parametric bootstrap to hypothesis testing (Alternative hypothesis: Prediction error of X>X-i)

A causal link is presumed between a given variable and the outcome if the null hypothesis is rejected, i.e., if the past of the variable significantly increase the accuray of prediction of the outcome.

For the simulations, simulated datasets with known causal structures were generated by sampling from different continuous and discrete distributions. Causal structures were represented by Directed Acyclic Graphs (DAGs). Each simulated dataset comprised 15000 subjects, with 10000 used for training the model and 5000 for prediction. The sample size was chosen to match the Millenium Cohort Study, which was used to illustrate the application of data discovery methods to the analysis of the long-term determinants of cognitive development. For the simulation scenarios, only continuous outcomes were modeled, but the approach can be used to discrete outcomes. Continuous outcomes were chosen to test the machine-learning algorithm in a more challenging scenario than classification.

```
set.seed(1234)
N <- 15000
n_train <- 10000
n_test  <- 5000
boot_num<-1000
```

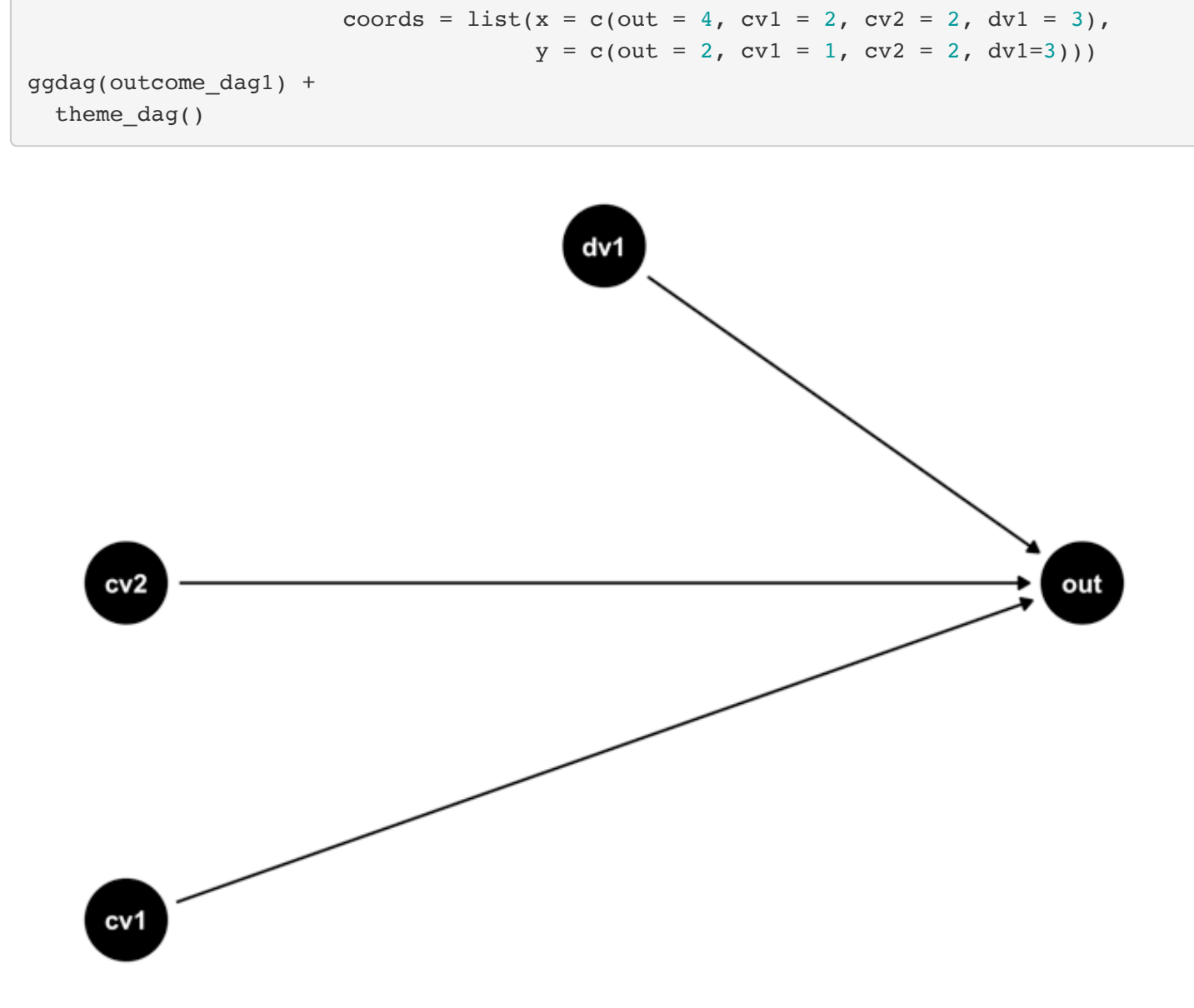
Baseline datasets, without relations between variables or between predictors and outcomes, were generated by random sampling:

```
 null_data<-function(n_people){
 nulld<-tibble(
  id = 1:n_people,
  outcome = rbeta(n_people, shape1 = 5, shape2 = 3),
  con_variable_1 = rnorm(n_people, 1.0, 0.2),
  con_variable_2 = rnorm(n_people, 1.5, 0.3),
  con_variable_3 = rnorm(n_people, 0.5, 0.3),
  dis_variable_1 = sample(c(0, 1), n_people, replace = TRUE),
  dis_variable_2 = rpois( n_people, lambda = 1))
return(nulld)
 }
null_sim<-null_data(N)
```

Scenario 1: Independent causes

Two continuous variables and one discrete variable independently causing the outcome were modeled as represented in the graph:

```
outcome_dag1 <- dagify(out ~ cv1 + cv2 + dv1,
                       coords = list(x = c(out = 4, cv1 = 2, cv2 = 2, dv1 = 3),
                                     y = c(out = 2, cv1 = 1, cv2 = 2, dv1=3)))
ggdag(outcome_dag1) +
  theme_dag()
```



```
sim_independent_effect <- null_sim %>%
mutate(outcome_modified = outcome + (0.3 * con_variable_1) + (0.3 * con_variable_2)+ (0.3 * dis_variable_1))
```

Relations are properly captured by multiple regression,

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_2 +
##     con_variable_3 + dis_variable_1 + dis_variable_2, data = sim_independent_effect)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56855 -0.11066  0.00971  0.12092  0.36624
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.615064   0.009907  62.085   <2e-16 ***
## con_variable_1  0.308816   0.006597  46.815   <2e-16 ***
## con_variable_2  0.303405   0.004393  69.071   <2e-16 ***
## con_variable_3 -0.006973   0.004439  -1.571    0.116
## dis_variable_1  0.301955   0.002628 114.914   <2e-16 ***
## dis_variable_2 -0.001424   0.001302  -1.094    0.274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1609 on 14994 degrees of freedom
## Multiple R-squared:  0.5757, Adjusted R-squared:  0.5756
## F-statistic:  4070 on 5 and 14994 DF,  p-value: < 2.2e-16
```

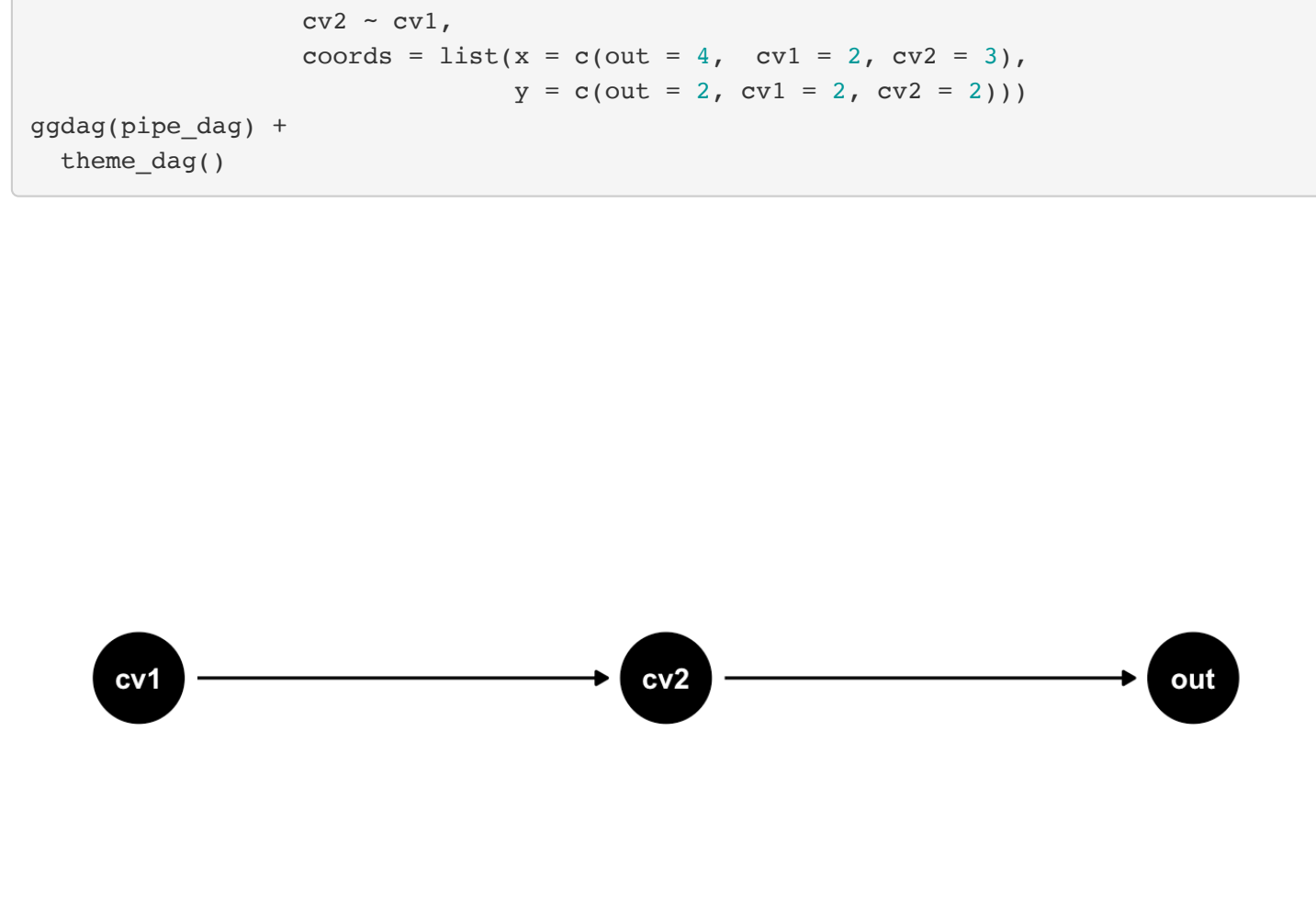
and by the GC-XGBoost algorithm :

```
##                 [,1]
## con_variable_1 0.000
## con_variable_2 0.000
## con_variable_3 0.990
## dis_variable_1 0.000
## dis_variable_2 0.259
```

Scenario 2: The pipe

The first fundamental simple DAG structure considered was “the pipe”.

```
pipe_dag <- dagify(out ~ cv2 ,
                   cv2 ~ cv1,
                   coords = list(x = c(out = 4,  cv1 = 2, cv2 = 3),
                                 y = c(out = 2, cv1 = 2, cv2 = 2)))
ggdag(pipe_dag) +
  theme_dag()
```



```
sim_pipe <- null_sim %>%
  mutate(con_variable_2 = con_variable_2 + (0.3 * con_variable_1)) %>%
  mutate(outcome_modified = outcome +  (0.3 * con_variable_2) )
```

If all variables are included in a multiple regression, the true associations are found:

The structure is also captured by the CGXBoost approach:

```
##                 [,1]
## con_variable_1 0.692
## con_variable_2 0.000
## con_variable_3 0.611
## dis_variable_1 0.094
## dis_variable_2 0.157
```

However, a spurious relation between CV1 and the outcome is introduced if CV2 is not included in the regression model:

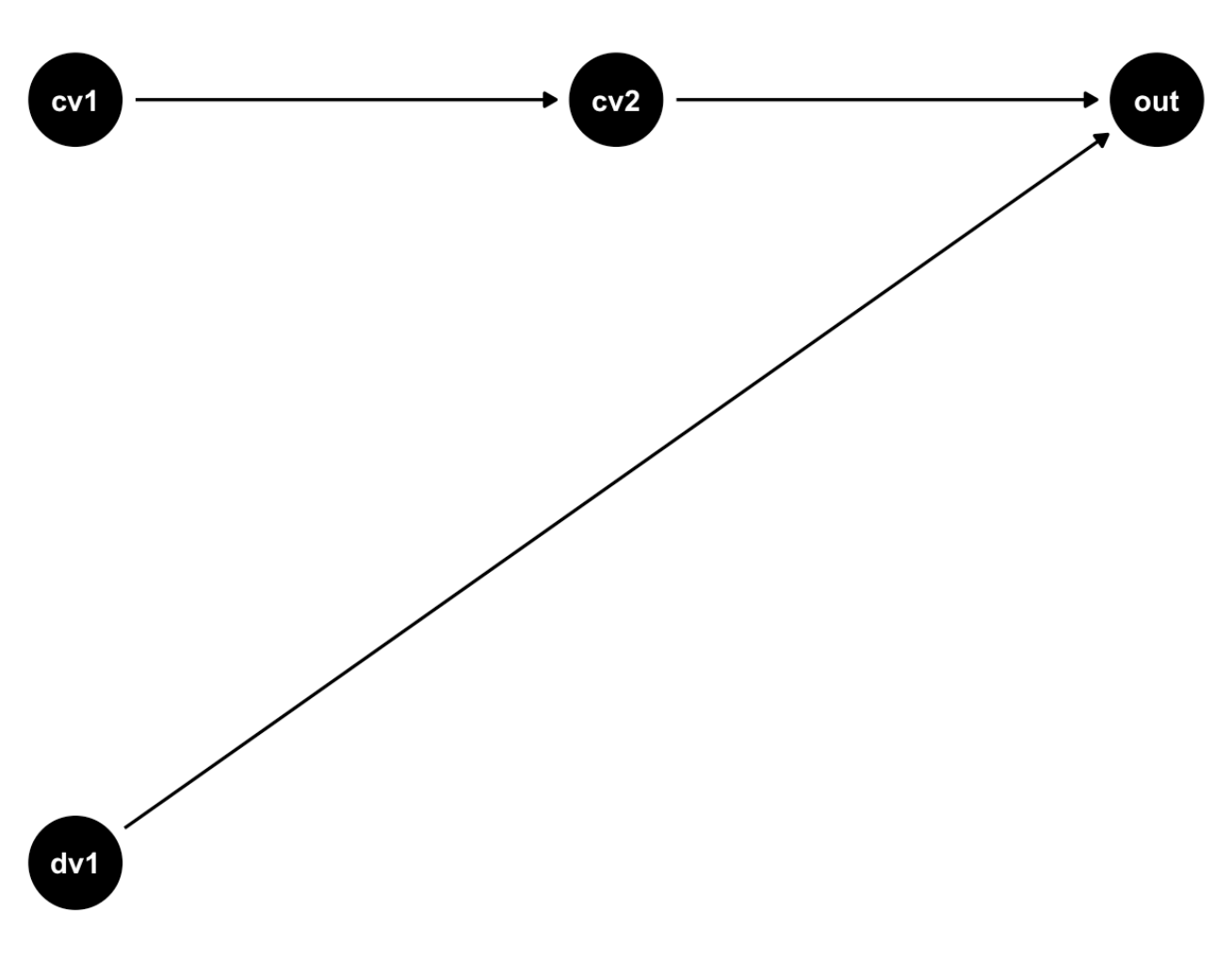
```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_3 +
##     dis_variable_1 + dis_variable_2, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79968 -0.12863  0.00507  0.13405  0.54935
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.074770   0.008425 127.571   <2e-16 ***
## con_variable_1  0.093483   0.007573  12.344   <2e-16 ***
## con_variable_3 -0.007709   0.005097  -1.513    0.130
## dis_variable_1  0.004538   0.003016   1.505    0.132
## dis_variable_2 -0.001960   0.001494  -1.312    0.190
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1847 on 14995 degrees of freedom
## Multiple R-squared:  0.01046,   Adjusted R-squared:  0.0102
## F-statistic: 39.65 on 4 and 14995 DF,  p-value: < 2.2e-16
```

However, XGBoost procedure does not falsely detect such a direct link:

```
##                 [,1]
## con_variable_1 0.275
## con_variable_3 0.890
## dis_variable_1 0.215
## dis_variable_2 0.366
```

Including an independent causal link, the regression an GC-XGBoost results are in agreement with the structural ground truth if all variables are included:

```
pipe_dag <- dagify(out ~ cv2 + dv1 ,
                   cv2 ~ cv1,
                   coords = list(x = c(out = 4, dv1 = 2, cv1 = 2, cv2 = 3),
                                 y = c(out = 2, dv1 = 1, cv1 = 2, cv2 = 2)))
ggdag(pipe_dag) +
  theme_dag()
```



```
sim_pipe_1 <- null_sim %>%
  mutate(con_variable_2 = con_variable_2 + (0.3 * con_variable_1)) %>%
  mutate(outcome_modified = outcome +  (0.3 * con_variable_2)+ (0.3 * dis_variable_1) )


model_all <- lm(outcome_modified ~ con_variable_1 + con_variable_2 + con_variable_3 + dis_variable_1 +dis_variabl
e_2, data = sim_pipe)  #which is controlled by including con_variable_2
summary(model_all)
```

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_2 +
##     con_variable_3 + dis_variable_1 + dis_variable_2, data = sim_pipe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56855 -0.11066  0.00971  0.12092  0.36624
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.615064   0.009907  62.085   <2e-16 ***
## con_variable_1  0.007795   0.006712   1.161    0.246
## con_variable_2  0.303405   0.004393  69.071   <2e-16 ***
## con_variable_3 -0.006973   0.004439  -1.571    0.116
## dis_variable_1  0.001955   0.002628   0.744    0.457
## dis_variable_2 -0.001424   0.001302  -1.094    0.274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1609 on 14994 degrees of freedom
## Multiple R-squared:  0.2493, Adjusted R-squared:  0.2491
## F-statistic:   996 on 5 and 14994 DF,  p-value: < 2.2e-16
```

```
df<-sim_pipe_1
```

```
##                 [,1]
## con_variable_1 0.933
## con_variable_2 0.000
## con_variable_3 0.722
## dis_variable_1 0.000
## dis_variable_2 0.702
```

However, a relation between CV1 and the outcome is introduced if CV2 is not included in the regression model:

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_3 +
##     dis_variable_1 + dis_variable_2, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79968 -0.12863  0.00507  0.13405  0.54935
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.074770   0.008425 127.571   <2e-16 ***
## con_variable_1  0.093483   0.007573  12.344   <2e-16 ***
## con_variable_3 -0.007709   0.005097  -1.513     0.13
## dis_variable_1  0.304538   0.003016 100.959   <2e-16 ***
## dis_variable_2 -0.001960   0.001494  -1.312     0.19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1847 on 14995 degrees of freedom
## Multiple R-squared:  0.4085, Adjusted R-squared:  0.4083
## F-statistic:  2589 on 4 and 14995 DF,  p-value: < 2.2e-16
```

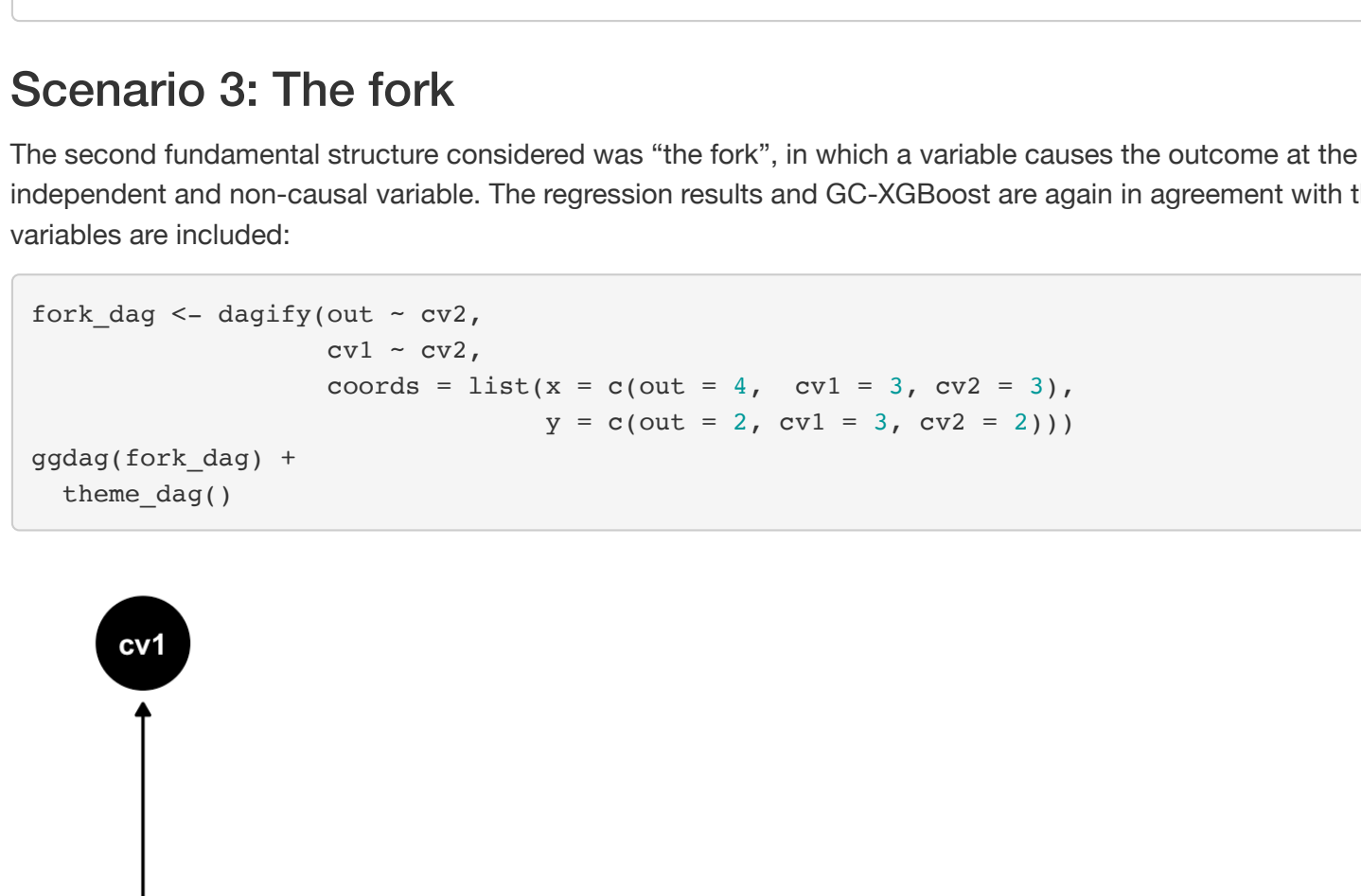
Which is not observed for the GC-XGBoost:

```
##                 [,1]
## con_variable_1 0.602
## con_variable_3 0.848
## dis_variable_1 0.000
## dis_variable_2 0.709
```

Scenario 3: The fork

The second fundamental structure considered was “the fork”, in which a variable causes the outcome at the same time that it causes another independent and non-causal variable. The regression results and GC-XGBoost are again in agreement with the structural ground-truth if all variables are included:

```
fork_dag <- dagify(out ~ cv2,
                   cv1 ~ cv2,
                   coords = list(x = c(out = 4,  cv1 = 3, cv2 = 3),
                                 y = c(out = 2, cv1 = 3, cv2 = 2)))
ggdag(fork_dag) +
  theme_dag()
```



```
sim_fork <- null_sim %>%
  mutate(con_variable_1 = con_variable_1 + (0.3 * con_variable_2)) %>%
  mutate(outcome_modified = outcome +  (0.3 * con_variable_2) )


model_all <- lm(outcome_modified ~ con_variable_1 + con_variable_2 + con_variable_3 + dis_variable_1 +dis_variabl
e_2, data = sim_fork)  #which is controled by including con_variable_2
summary(model_all)
```

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_2 +
##     con_variable_3 + dis_variable_1 + dis_variable_2, data = sim_fork)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56855 -0.11066  0.00971  0.12092  0.36624
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.615064   0.009907  62.085   <2e-16 ***
## con_variable_1  0.008816   0.006597   1.336    0.181
## con_variable_2  0.300760   0.004797  62.702   <2e-16 ***
## con_variable_3 -0.006973   0.004439  -1.571    0.116
## dis_variable_1  0.001955   0.002628   0.744    0.457
## dis_variable_2 -0.001424   0.001302  -1.094    0.274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1609 on 14994 degrees of freedom
## Multiple R-squared:  0.2417, Adjusted R-squared:  0.2415
## F-statistic: 955.9 on 5 and 14994 DF,  p-value: < 2.2e-16
```

```
df<-sim_fork
```

```
##                 [,1]
## con_variable_1 0.870
## con_variable_2 0.000
## con_variable_3 0.821
## dis_variable_1 0.241
## dis_variable_2 0.257
```

A spurious correlation between CV1 and the outcome is introduced when CV2 is not included in the regression model:

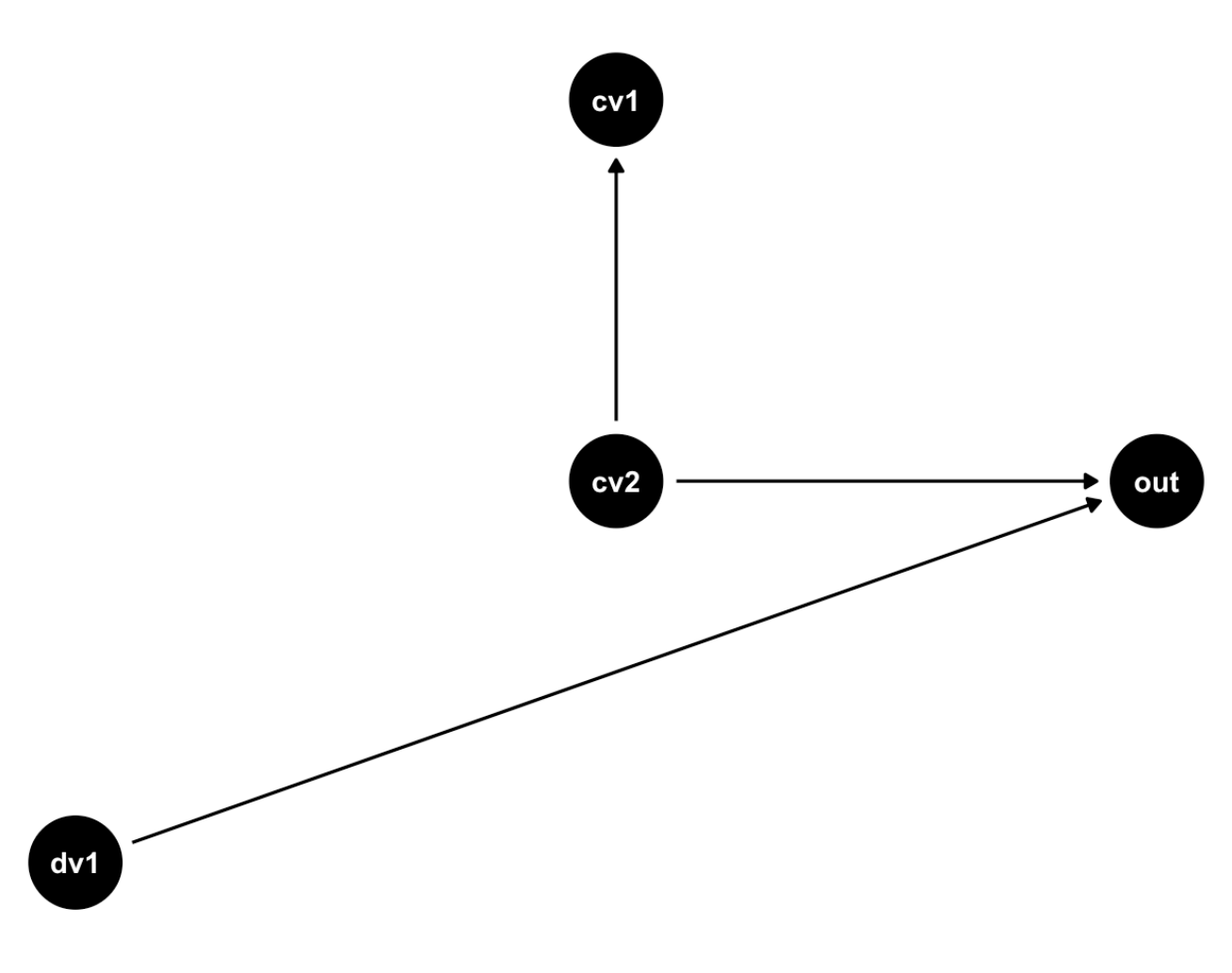
```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_3 +
##     dis_variable_1 + dis_variable_2, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77378 -0.12484  0.00577  0.13040  0.51936
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.824192   0.010480  78.644   <2e-16 ***
## con_variable_1  0.175029   0.006786  25.792   <2e-16 ***
## con_variable_3 -0.008505   0.004987  -1.705   0.0881 .
## dis_variable_1  0.003976   0.002952   1.347   0.1780
## dis_variable_2 -0.001721   0.001462  -1.177   0.2391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1808 on 14995 degrees of freedom
## Multiple R-squared:  0.04288,    Adjusted R-squared:  0.04262
## F-statistic: 167.9 on 4 and 14995 DF,  p-value: < 2.2e-16
```

But not in GC-XGBoost:

```
##                 [,1]
## con_variable_1 0.120
## con_variable_3 0.980
## dis_variable_1 0.889
## dis_variable_2 0.687
```

The results are not affected by adding an extra independent causal link:

```
fork_dag <- dagify(out ~ cv2 + dv1 ,
                   cv1 ~ cv2,
                   coords = list(x = c(out = 4, dv1 = 2, cv1 = 3, cv2 = 3),
                                 y = c(out = 2, dv1 = 1, cv1 = 3, cv2 = 2)))
ggdag(fork_dag) +
  theme_dag()
```



```
sim_fork_1 <- null_sim %>%
  mutate(con_variable_1 = con_variable_1 + (0.3 * con_variable_2)) %>%
  mutate(outcome_modified = outcome +  (0.3 * con_variable_2)+ (0.3 * dis_variable_1) )


model_all <- lm(outcome_modified ~ con_variable_1 + con_variable_2 + con_variable_3 + dis_variable_1 +dis_variabl
e_2, data = sim_fork_1)  #which is controled by including con_variable_2
summary(model_all)
```

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_2 +
##     con_variable_3 + dis_variable_1 + dis_variable_2, data = sim_fork_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56855 -0.11066  0.00971  0.12092  0.36624
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.615064   0.009907  62.085   <2e-16 ***
## con_variable_1  0.008816   0.006597   1.336    0.181
## con_variable_2  0.300760   0.004797  62.702   <2e-16 ***
## con_variable_3 -0.006973   0.004439  -1.571    0.116
## dis_variable_1  0.301955   0.002628 114.914   <2e-16 ***
## dis_variable_2 -0.001424   0.001302  -1.094    0.274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1609 on 14994 degrees of freedom
## Multiple R-squared:  0.5485, Adjusted R-squared:  0.5483
## F-statistic:  3643 on 5 and 14994 DF,  p-value: < 2.2e-16
```

```
df<-sim_fork_1
```

```
##                 [,1]
## con_variable_1 0.988
## con_variable_2 0.000
## con_variable_3 0.998
## dis_variable_1 0.000
## dis_variable_2 1.000
```

A spurious correlation between CV1 and the outcome is introduced when CV2 is not included in the model:

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_3 +
##     dis_variable_1 + dis_variable_2, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77378 -0.12484  0.00577  0.13040  0.51936
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.824192   0.010480  78.644   <2e-16 ***
## con_variable_1  0.175029   0.006786  25.792   <2e-16 ***
## con_variable_3 -0.008505   0.004987  -1.705   0.0881 .
## dis_variable_1  0.303976   0.002952 102.980   <2e-16 ***
## dis_variable_2 -0.001721   0.001462  -1.177   0.2391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1808 on 14995 degrees of freedom
## Multiple R-squared:  0.4301, Adjusted R-squared:  0.4299
## F-statistic:  2829 on 4 and 14995 DF,  p-value: < 2.2e-16
```

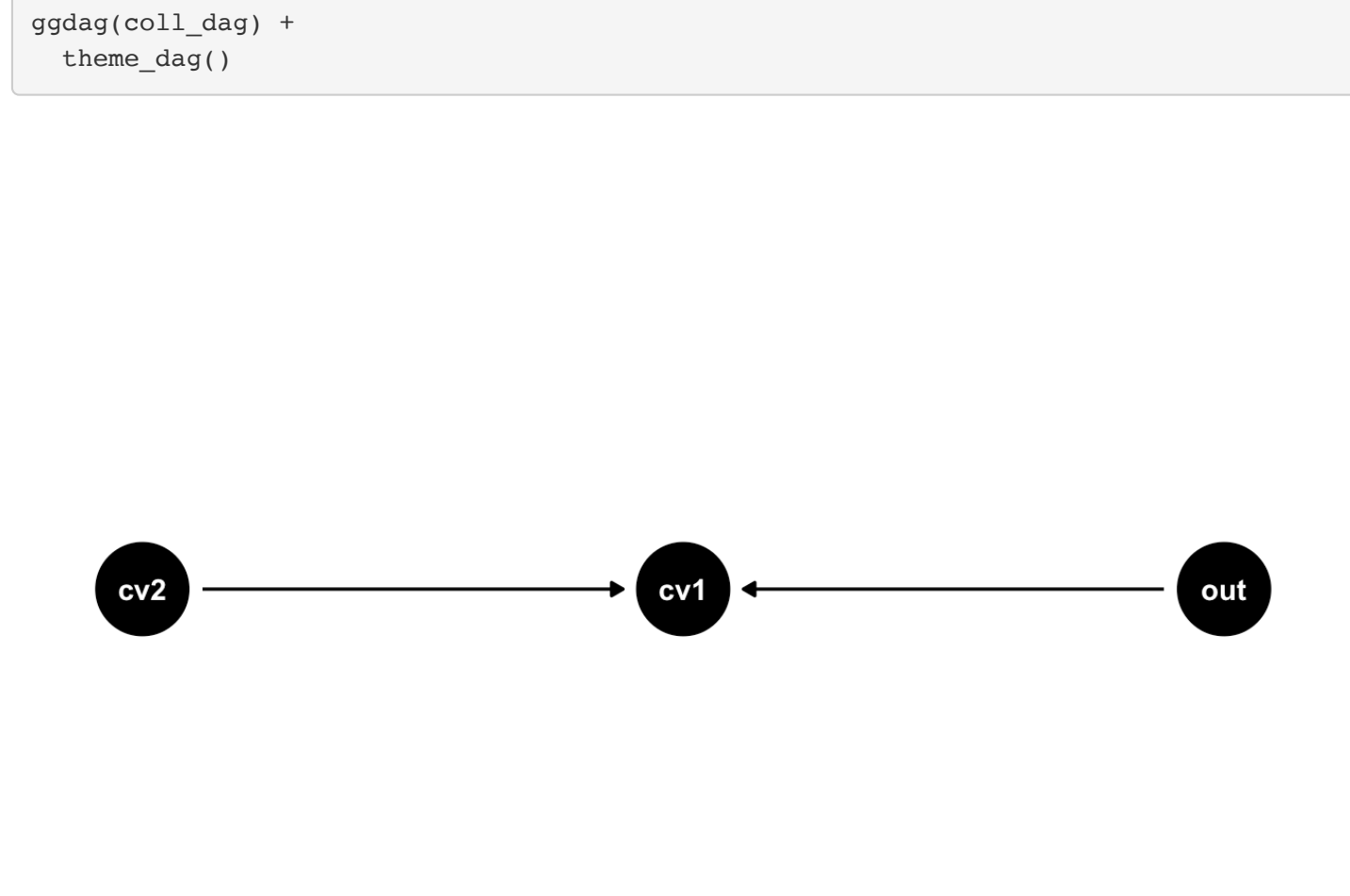
but not in the GC-XGBoost approach :

```
##                 [,1]
## con_variable_1 0.228
## con_variable_3 0.989
## dis_variable_1 0.000
## dis_variable_2 0.298
```

Scenario 4: The collider

The third fundamental structure investigated was “the collider” graph motive.

```
coll_dag <- dagify(cv1 ~ out + cv2,
                   coords = list(x = c(out = 4, cv1 = 3, cv2 = 2),
                                 y = c(out = 2, cv1 = 2, cv2 = 2)))
ggdag(coll_dag) +
  theme_dag()
```



```
sim_coll_1 <- null_sim %>%
  mutate(con_variable_1 = con_variable_1 + (0.3 * outcome) + (0.3 * con_variable_2)) %>%
  mutate(outcome_modified = outcome )


model_all <- lm(outcome_modified ~ con_variable_1 + con_variable_2 + con_variable_3 + dis_variable_1 +dis_variabl
e_2, data = sim_coll_1)  #which is controled by including con_variable_2
summary(model_all)
```

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_1 + con_variable_2 +
##     con_variable_3 + dis_variable_1 + dis_variable_2, data = sim_coll_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57562 -0.10701  0.00988  0.11486  0.43375
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.393051   0.010279  38.240   <2e-16 ***
## con_variable_1  0.192269   0.006200  31.013   <2e-16 ***
## con_variable_2 -0.053038   0.004630 -11.455   <2e-16 ***
## con_variable_3 -0.007563   0.004303  -1.757   0.0789 .
## dis_variable_1  0.001706   0.002547   0.670   0.5030
## dis_variable_2 -0.001182   0.001262  -0.937   0.3489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.156 on 14994 degrees of freedom
## Multiple R-squared:  0.06058,    Adjusted R-squared:  0.06027
## F-statistic: 193.4 on 5 and 14994 DF,  p-value: < 2.2e-16
```

```
df<-sim_coll_1
```

Including all variables, the collider blocks the relation.

```
##                 [,1]
## con_variable_1 0.000
## con_variable_2 0.240
## con_variable_3 0.651
## dis_variable_1 0.467
## dis_variable_2 0.293
```

When the collider variable is NOT included in the regression, no spurious relation is introduced:

```
##
## Call:
## lm(formula = outcome_modified ~ con_variable_2 + con_variable_3 +
##     dis_variable_1 + dis_variable_2, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56799 -0.11072  0.00984  0.12093  0.36646
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.624004   0.007308  85.386   <2e-16 ***
## con_variable_2  0.003337   0.004392   0.760    0.447
## con_variable_3 -0.006926   0.004439  -1.560    0.119
## dis_variable_1  0.001962   0.002628   0.746    0.455
## dis_variable_2 -0.001432   0.001302  -1.100    0.271
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1609 on 14995 degrees of freedom
## Multiple R-squared:  0.0003205,  Adjusted R-squared:  5.387e-05
## F-statistic: 1.202 on 4 and 14995 DF,  p-value: 0.3076
```

And the same is observed for XG-BOOST:

```
##                 [,1]
## con_variable_1 0.903
## con_variable_3 0.998
## dis_variable_1 0.762
## dis_variable_2 0.968
```