covdepGE demo

10/31/2021

```
library(covdepGE)
library(ggplot2)
library(latex2exp)
`?`(covdepGE())
```

```
## starting httpd help server ... done
```

```
# script for generating the data for the discrete and continuous
# covariate model
source("generate_data.R")
cont <- generate_continuous()
disc <- generate_discrete(same = F)

# dimensions of the data
dim(cont$data)
```

```
## [1] 180   5
```

```
dim(disc$data)
```

```
## [1] 100  11
```

```
# apply covdepGE to the continuous data
out.cont <- covdepGE(cont$data, cont$covts, print_time = T, tau = 0.56)
```

```
## Time difference of 2.368989 secs
```

```
# apply covdepGE to the discrete data
out.disc <- covdepGE(disc$data, disc$covts, print_time = T)
```

```
## Time difference of 6.06228 secs
```

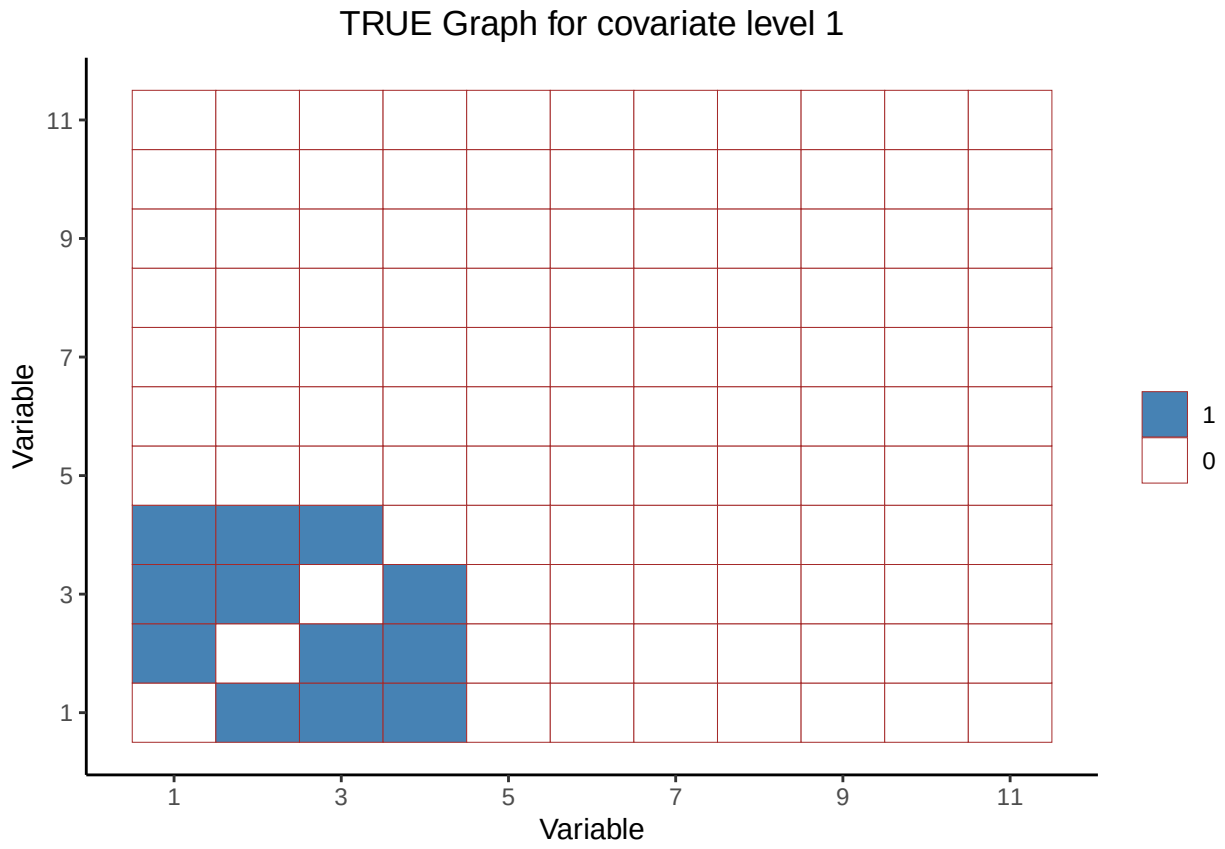
```
# time trials:
microbenchmark::microbenchmark(covdepGE(cont$data, cont$covts, tau = 0.56),
    covdepGE(disc$data, disc$covts), times = 10)
```

```
## Unit: seconds
##                                       expr      min       lq     mean
##  covdepGE(cont$data, cont$covts, tau = 0.56) 1.249733 1.390201 1.481883
##              covdepGE(disc$data, disc$covts) 5.165367 5.598895 6.044160
##    median       uq      max neval
##  1.513278 1.563111 1.668632    10
##  6.143843 6.402495 6.549847    10
```

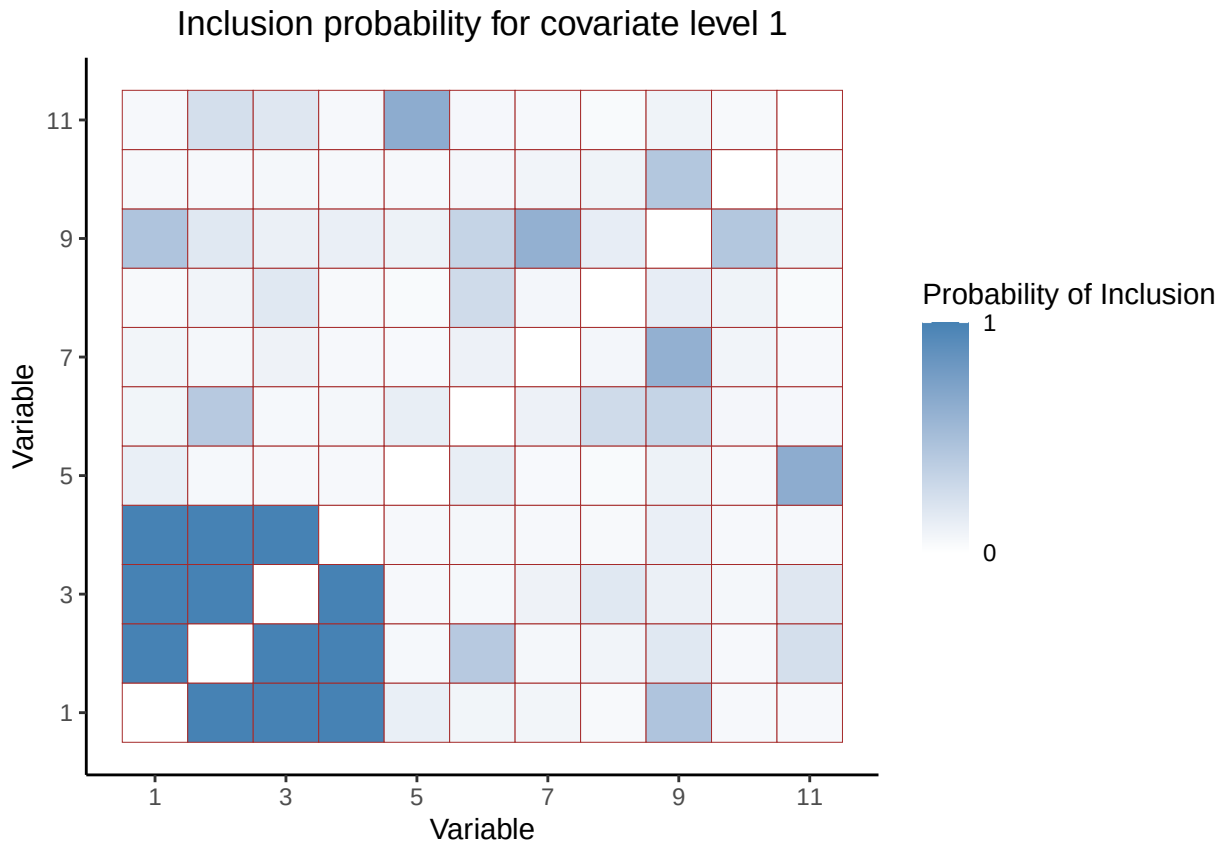
```r
# get the predicted discrete probabilities of inclusion
incl.probs.disc <- out.disc$inclusion_probs

# get the true covariance structure
incl.probs.disc.tr1 <- (disc$true_covariance[[1]] - diag(diag(disc$true_covariance[[1]])) !=
    0) * 1
incl.probs.disc.tr2 <- (disc$true_covariance[[2]] - diag(diag(disc$true_covariance[[2]])) !=
    0) * 1

# get the discrete graphs
graphs.disc <- out.disc$graphs

# visualize the probabilities of inclusion for the first
# individual (covariate level 1)
round(incl.probs1 <- incl.probs.disc[[1]], 2)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
##  [1,] 0.00 1.00 1.00 1.00 0.12 0.08 0.08 0.05 0.46  0.05  0.05
##  [2,] 1.00 0.00 1.00 1.00 0.06 0.41 0.06 0.08 0.17  0.05  0.25
##  [3,] 1.00 1.00 0.00 1.00 0.05 0.06 0.10 0.17 0.12  0.06  0.18
##  [4,] 1.00 1.00 1.00 0.00 0.05 0.06 0.05 0.05 0.12  0.05  0.05
##  [5,] 0.12 0.06 0.05 0.05 0.00 0.13 0.05 0.04 0.10  0.05  0.64
##  [6,] 0.08 0.41 0.06 0.06 0.13 0.00 0.10 0.28 0.33  0.06  0.06
##  [7,] 0.08 0.06 0.10 0.05 0.05 0.10 0.00 0.07 0.62  0.08  0.05
##  [8,] 0.05 0.08 0.17 0.05 0.04 0.28 0.07 0.00 0.14  0.09  0.04
##  [9,] 0.46 0.17 0.12 0.12 0.10 0.33 0.62 0.14 0.00  0.43  0.09
## [10,] 0.05 0.05 0.06 0.05 0.05 0.06 0.08 0.09 0.43  0.00  0.05
## [11,] 0.05 0.25 0.18 0.05 0.64 0.06 0.05 0.04 0.09  0.05  0.00
```

```r
# true (lvl1)
(ggplot(reshape2::melt(incl.probs.disc.tr1), aes(x = Var1, y = Var2,
    fill = value)) + geom_tile(color = "brown") + scale_fill_gradient(low = "white",
    high = "steelblue", breaks = c(1, 0)) + guides(fill = guide_legend(title = "")) +
    theme_classic() + xlab("Variable") + ylab("Variable") + scale_x_continuous(breaks = seq(1,
    ncol(disc$data), 2)) + scale_y_continuous(breaks = seq(1, ncol(disc$data),
    2)) + ggtitle("TRUE Graph for covariate level 1") + theme(plot.title = element_text(hjust = 0.5)))
```



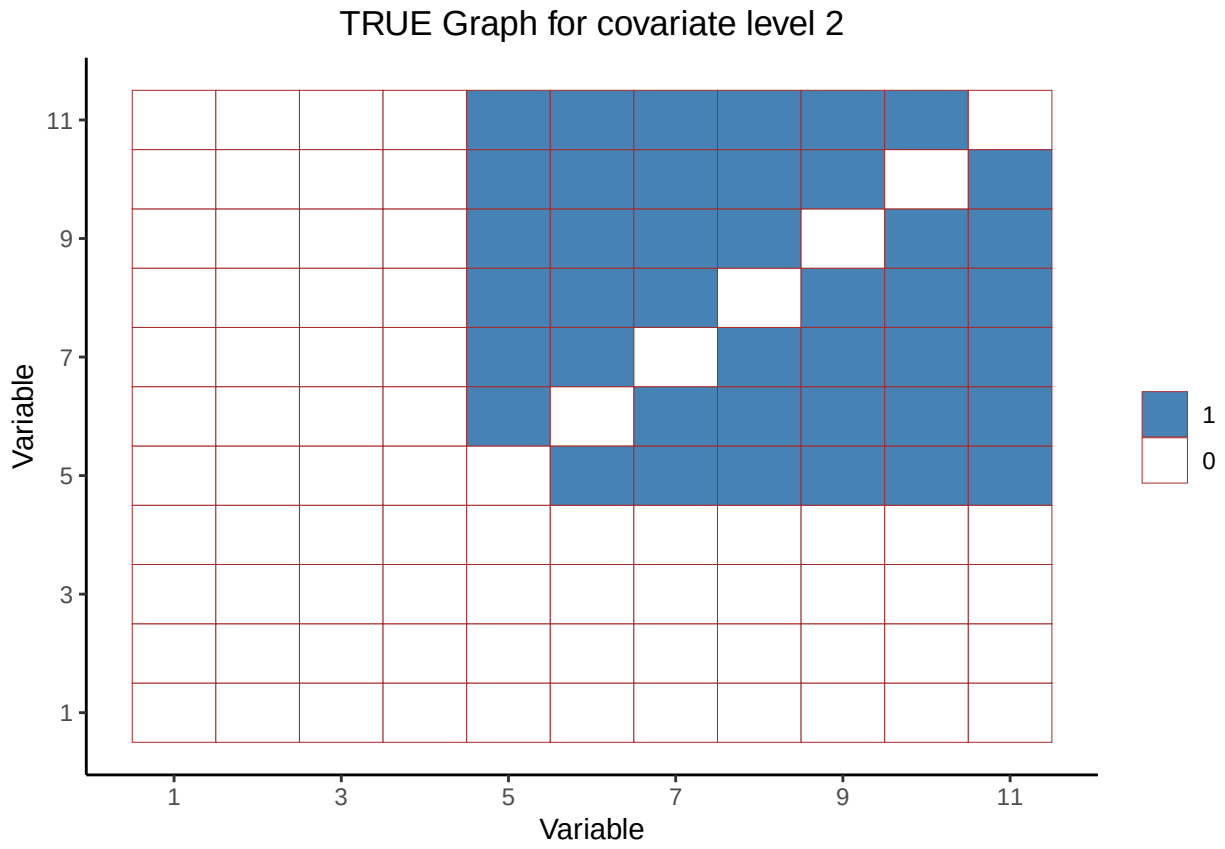

```
# predicted (lvl1)
(ggplot(reshape2::melt(incl.probs1), aes(x = Var1, y = Var2, fill = value)) +
    geom_tile(color = "brown") + scale_fill_gradient(low = "white",
    high = "steelblue", breaks = c(1, 0)) + labs(fill = "Probability of Inclusion") +
    theme_classic() + xlab("Variable") + ylab("Variable") + scale_x_continuous(breaks = seq(1,
    ncol(disc$data), 2)) + scale_y_continuous(breaks = seq(1, ncol(disc$data),
    2)) + ggtitle("Inclusion probability for covariate level 1") +
    theme(plot.title = element_text(hjust = 0.5)))
```



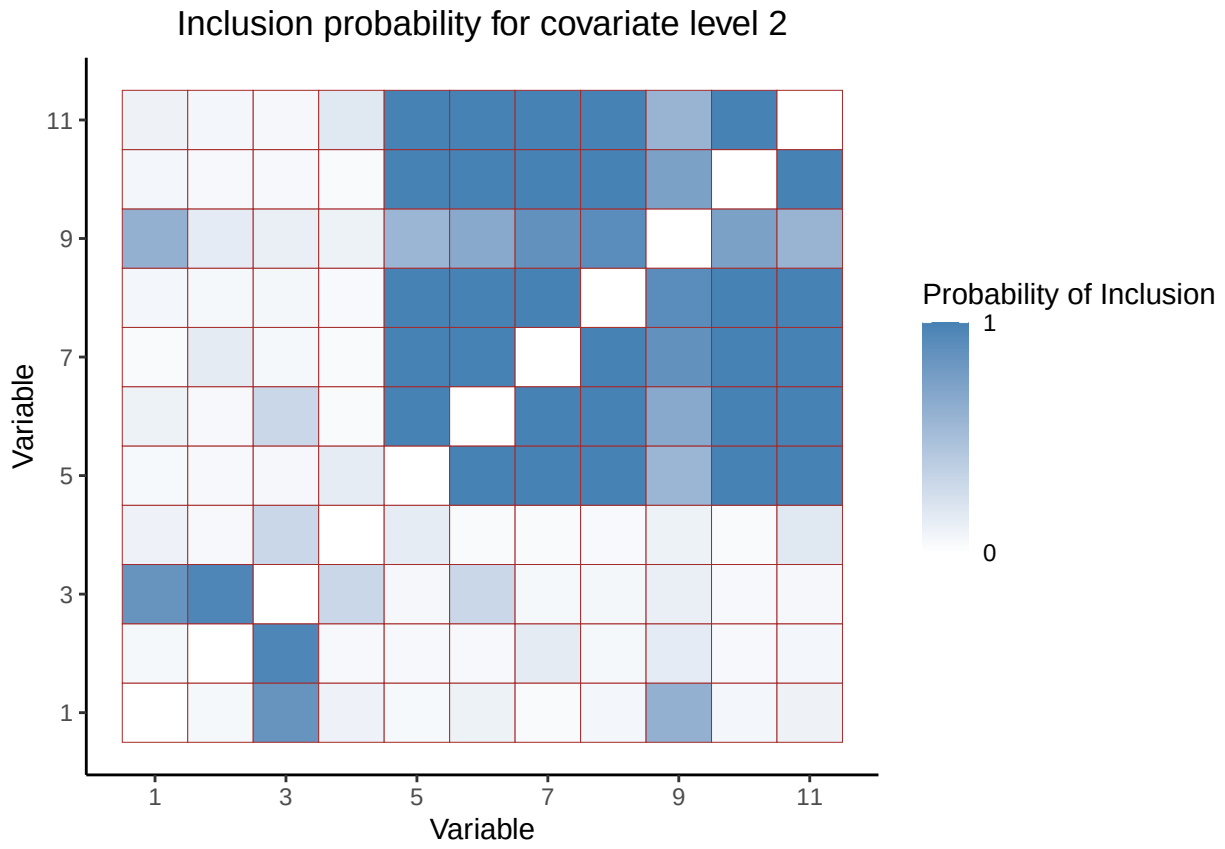

```
# visualize the probabilities of inclusion for the first
# individual (covariate level 1)
round(incl.probs2 <- incl.probs.disc[[length(incl.probs.disc)]], 2)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
##  [1,] 0.00 0.05 0.85 0.10 0.05 0.10 0.04 0.07 0.61  0.07  0.10
##  [2,] 0.05 0.00 0.97 0.05 0.05 0.05 0.15 0.06 0.15  0.05  0.07
##  [3,] 0.85 0.97 0.00 0.30 0.06 0.31 0.06 0.06 0.12  0.05  0.06
##  [4,] 0.10 0.05 0.30 0.00 0.14 0.04 0.04 0.04 0.10  0.04  0.18
##  [5,] 0.05 0.05 0.06 0.14 0.00 1.00 1.00 1.00 0.57  1.00  1.00
##  [6,] 0.10 0.05 0.31 0.04 1.00 0.00 1.00 1.00 0.68  1.00  1.00
##  [7,] 0.04 0.15 0.06 0.04 1.00 1.00 0.00 1.00 0.87  1.00  1.00
##  [8,] 0.07 0.06 0.06 0.04 1.00 1.00 1.00 0.00 0.91  1.00  1.00
##  [9,] 0.61 0.15 0.12 0.10 0.57 0.68 0.87 0.91 0.00  0.74  0.58
## [10,] 0.07 0.05 0.05 0.04 1.00 1.00 1.00 1.00 0.74  0.00  1.00
## [11,] 0.10 0.07 0.06 0.18 1.00 1.00 1.00 1.00 0.58  1.00  0.00
```

```
# true (lvl2)
(ggplot(reshape2::melt(incl.probs.disc.tr2), aes(x = Var1, y = Var2,
    fill = value)) + geom_tile(color = "brown") + scale_fill_gradient(low = "white",
    high = "steelblue", breaks = c(1, 0)) + guides(fill = guide_legend(title = "")) +
    theme_classic() + xlab("Variable") + ylab("Variable") + scale_x_continuous(breaks = seq(1,
    ncol(disc$data), 2)) + scale_y_continuous(breaks = seq(1, ncol(disc$data),
    2)) + ggtitle("TRUE Graph for covariate level 2") + theme(plot.title = element_text(hjust = 0.5)))
```



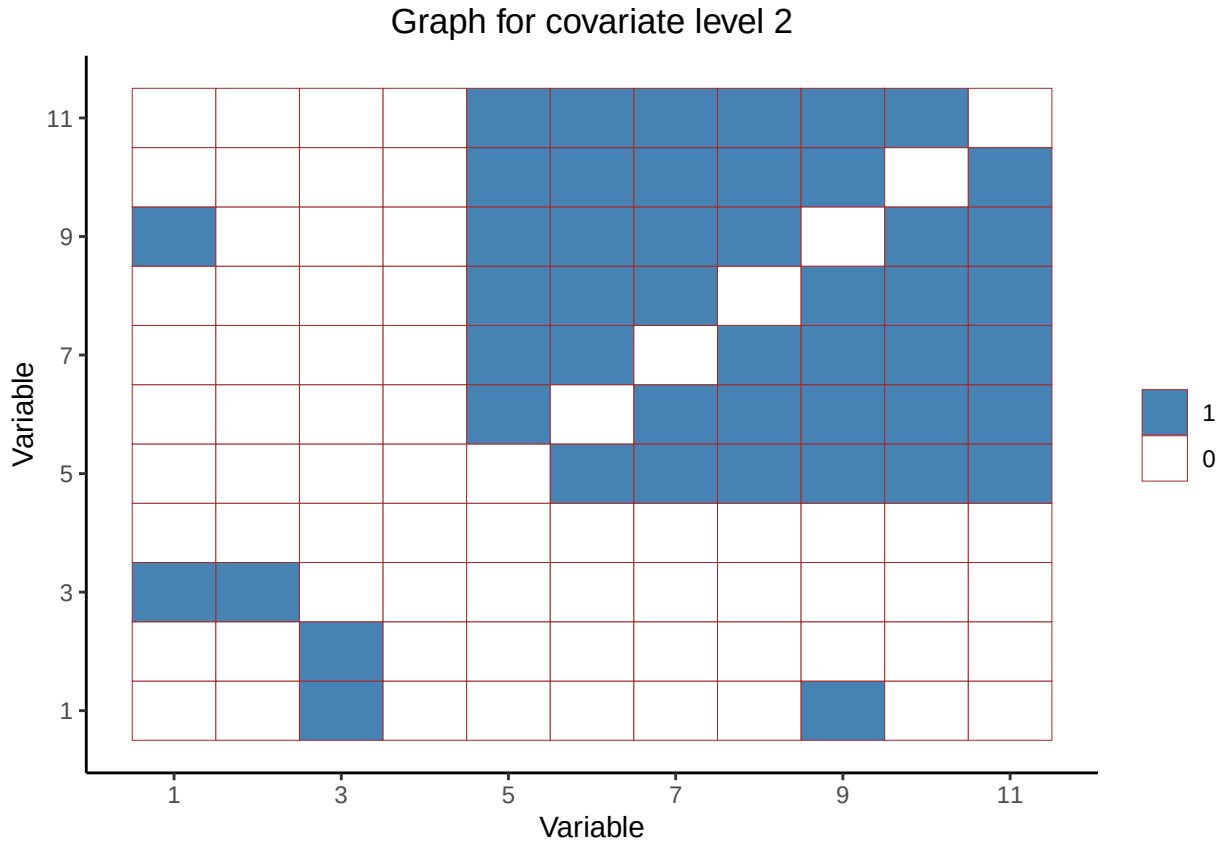

```
# predicted (lvl2)
(ggplot(reshape2::melt(incl.probs2), aes(x = Var1, y = Var2, fill = value)) +
    geom_tile(color = "brown") + scale_fill_gradient(low = "white",
    high = "steelblue", breaks = c(1, 0)) + labs(fill = "Probability of Inclusion") +
    theme_classic() + xlab("Variable") + ylab("Variable") + scale_x_continuous(breaks = seq(1,
    ncol(disc$data), 2)) + scale_y_continuous(breaks = seq(1, ncol(disc$data),
    2)) + ggtitle("Inclusion probability for covariate level 2") +
    theme(plot.title = element_text(hjust = 0.5)))
```



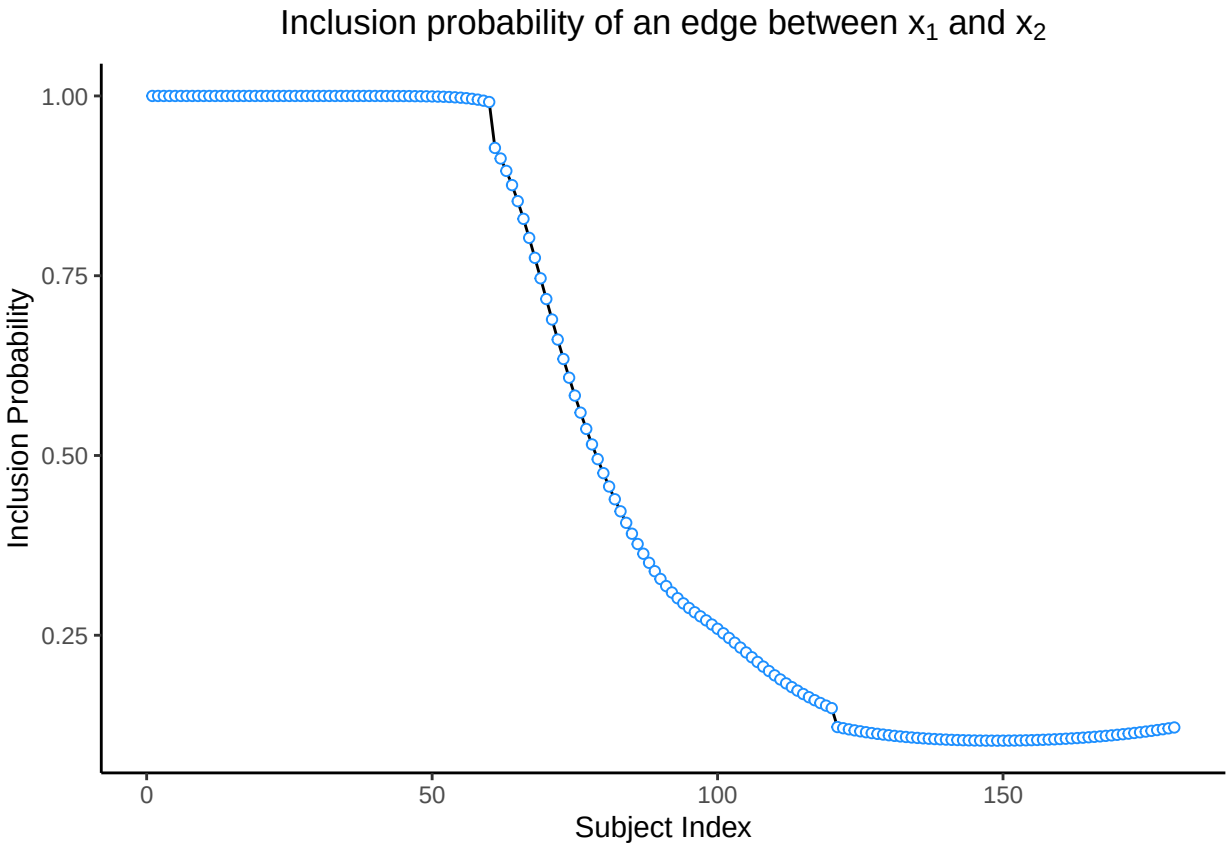

```
# visualize the graph for the last individual (covariate level
# 2)
(graph100 <- graphs.disc[[length(graphs.disc)]])
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
##  [1,]    0    0    1    0    0    0    0    0    1     0     0
##  [2,]    0    0    1    0    0    0    0    0    0     0     0
##  [3,]    1    1    0    0    0    0    0    0    0     0     0
##  [4,]    0    0    0    0    0    0    0    0    0     0     0
##  [5,]    0    0    0    0    0    1    1    1    1     1     1
##  [6,]    0    0    0    0    1    0    1    1    1     1     1
##  [7,]    0    0    0    0    1    1    0    1    1     1     1
##  [8,]    0    0    0    0    1    1    1    0    1     1     1
##  [9,]    1    0    0    0    1    1    1    1    0     1     1
## [10,]    0    0    0    0    1    1    1    1    1     0     1
## [11,]    0    0    0    0    1    1    1    1    1     1     0
```

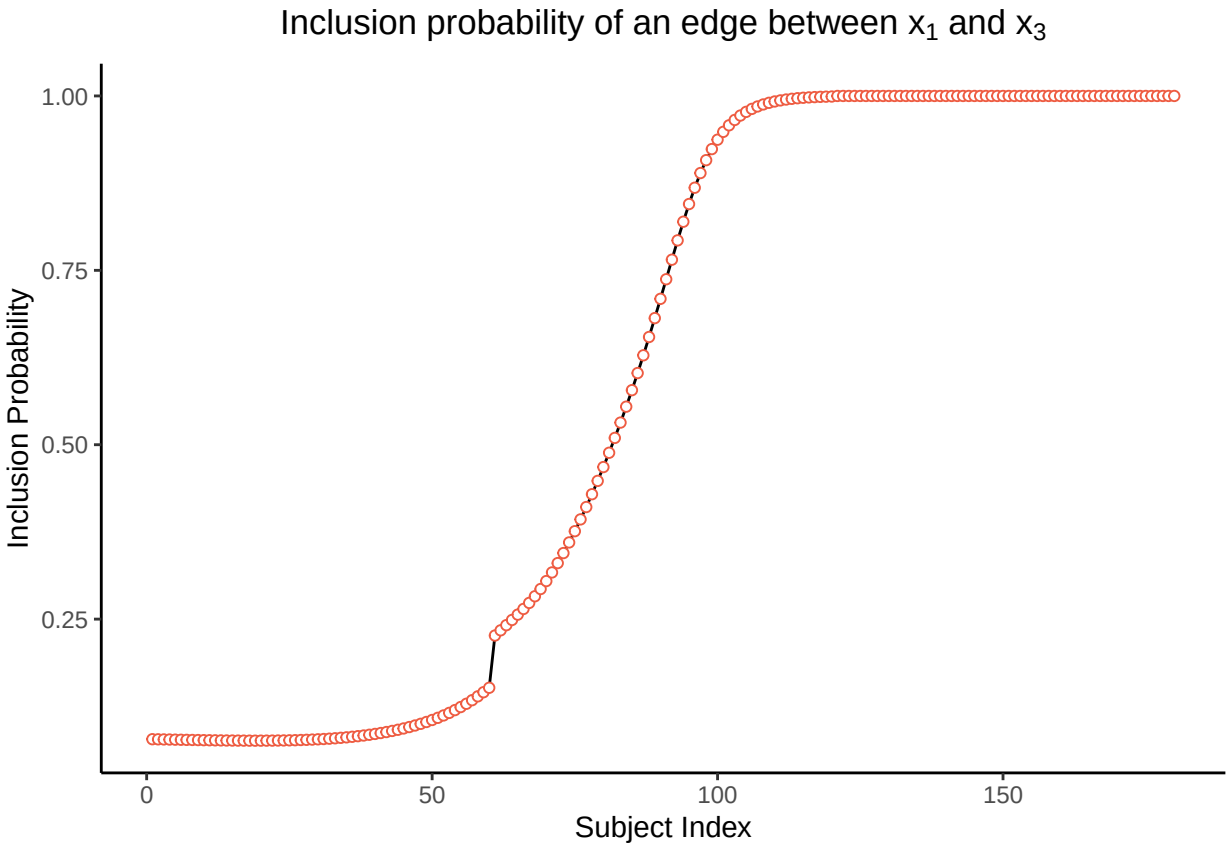
```
(ggplot(reshape2::melt(graph100), aes(x = Var1, y = Var2, fill = value)) +
    geom_tile(color = "brown") + scale_fill_gradient(low = "white",
    high = "steelblue", breaks = c(1, 0)) + guides(fill = guide_legend(title = "")) +
    theme_classic() + xlab("Variable") + ylab("Variable") + scale_x_continuous(breaks = seq(1,
    ncol(disc$data), 2)) + scale_y_continuous(breaks = seq(1, ncol(disc$data),
    2)) + ggtitle("Graph for covariate level 2") + theme(plot.title = element_text(hjust = 0.5)))
```




```
# get the continuous probabilities of inclusion
incl.probs.cont <- out.cont$inclusion_probs

# get continuous probabilities of inclusion for x_1 to x_2 and
# x_1 to x_3
probs12 <- unlist(lapply(incl.probs.cont, function(x) x[1, 2]))
probs13 <- unlist(lapply(out.cont$inclusion_probs, function(x) x[1,
    3]))

# visualize them
(ggplot(data.frame(subj = 1:length(probs12), prob = probs12), aes(subj,
    prob)) + geom_line() + geom_point(color = "dodgerblue", fill = "white",
    shape = 21) + theme_classic() + xlab("Subject Index") + ylab("Inclusion Probability") +
    ggtitle(TeX("Inclusion probability of an edge between $x_1$ and $x_2$")) +
    theme(plot.title = element_text(hjust = 0.5)))
```




```
(ggplot(data.frame(subj = 1:length(probs13), prob = probs13), aes(subj,
    prob)) + geom_line() + geom_point(color = "tomato2", fill = "white",
    shape = 21) + theme_classic() + xlab("Subject Index") + ylab("Inclusion Probability") +
    ggtitle(TeX("Inclusion probability of an edge between $x_1$ and $x_3$")) +
    theme(plot.title = element_text(hjust = 0.5)))
```


Inclusion probability of an edge between x_1 and x_3
Inclusion Probability
1.00
0.75
0.50
0.25
0
50
100
150
Subject Index