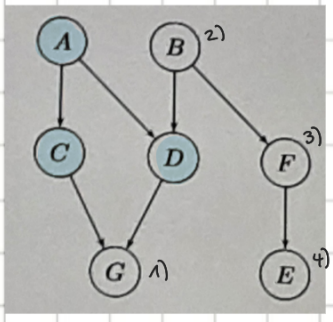
Project 9

Anna Toidze, Marie-Claire Indilewitsch, Sophia Houhamdi, Ieva Pudžiuvelytė

Problem 23

(i)



1) $A \perp G \mid \{C, D\}$ G is d-seperated from A given $\{C, D\}$

→ case a) from slides: head-to-tail at →(C)→ and C is in the set $\{C, D\}$

2) 3) 4) B, F, E are not d-seperated from A given $\{C, D\}$ (case a)

case b) → head-to-head but D is part of $\{C, D\}$ & F, E not in $\{C, D\}$

(ii) (a) $B \perp C \mid D$: False D is a collider on the path $B \to C$ ($C \leftarrow A \to D \leftarrow B$) because case b) head-to-head meeting at D but D is part of set $\{D\}$ =° so B not conditionally independent of C given D

(b) $G \perp E \mid D$: False would be true for conditioning on D on $G_1 \leftarrow D \leftarrow B \to F \to E$ (blocked) but due to the second path $G_1 \leftarrow C \leftarrow A \to D \leftarrow B \to F \to E$ (unblocked) G is not conditionally independent of E given D as case b) head-to-head meeting at D but D is part of set $\{D\}$

(c) $C \perp F \mid A$: True both paths I) $C \leftarrow A \to D \leftarrow B \to F$ (blocked by A & D)

II) $C \to G_1 \leftarrow D \leftarrow B \to F$ (blocked by G_1) are blocked.

(d) $C \perp E \mid MB(C) = \{A, G_1, D\}$: True both paths I) $C \leftarrow A \to D \leftarrow B \to F \leftarrow E$

II) $C \to G_1 \leftarrow D \leftarrow B \to F \leftarrow E$ are blocked.

+ given its Markov blanket C is conditionally independent of all other nodes in the graph

Project 9

```
library(ggplot2)
library(pcalg)
library(BiDAG)
library(graph)
```

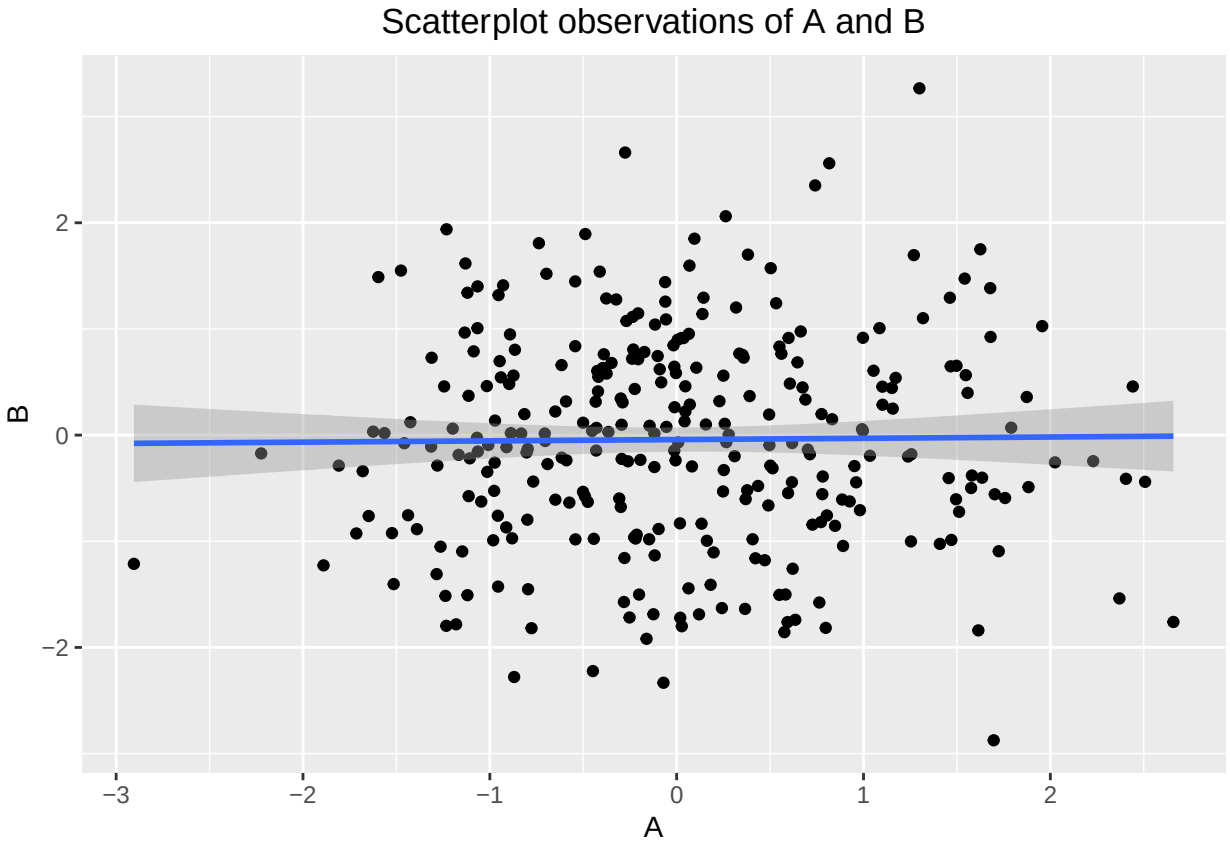
Problem 24: Testing for marginal correlation

Using the data from MVN DAG.rds, display the observations of A and B in a scatterplot. What does the plot suggest about their (marginal) correlation? Does it agree with Figure 2? Use the function cor.test() to test the null hypothesis of no correlation between A and B. What is your conclusion?

```
# load the data
data <- readRDS("MVN_DAG.rds")

# plot the scatterplot
ggplot(data, aes(x = A, y = B)) +
  geom_point()+
  theme(plot.title = element_text(hjust = 0.5))+
  ggtitle("Scatterplot observations of A and B") +
  xlab("A") +
  ylab("B")+
  geom_smooth(method = 'lm')
```

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



The plot suggests that A and B are uncorrelated which is also seen in the added linear regression line as its almost horizontal around 0. This agrees with Figure 2 as the graph structure also suggests a marginal correlation of 0, so independence.

```
# hypothesis testing

cor.test(data$A,data$B)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$A and data$B
## t = 0.20194, df = 298, p-value = 0.8401
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1016784  0.1247727
## sample estimates:
##        cor
## 0.01169715
```

Based on the calculated p-value of 0.8401 there seems to be no correlation between A and B and the null hypothesis can be accepted. This agrees with both the scatterplot and the graph, A and B seem to be (marginally) independent/uncorrelated.

Problem 25: Testing for partial correlation

Linearly regress A on C (that is, with A as the response variable and C as the explanatory variable). Compute and store the residuals.

```
residuals_AC <- residuals(lm(A ~ C, data= data))
```

Linearly regress B on C. Compute and store the residuals.

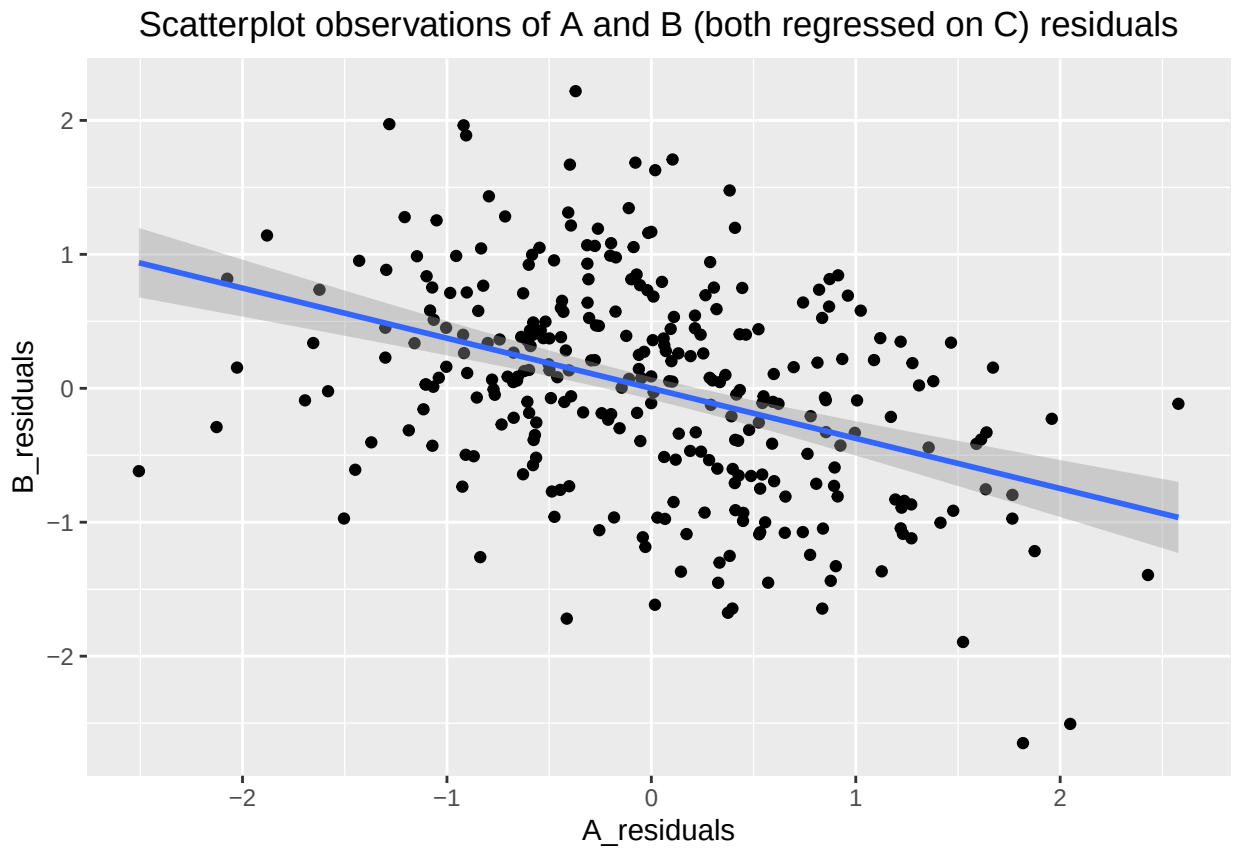
```
residuals_BC <- residuals(lm(B ~ C, data= data))
```

Plot the residuals of A (regressed on C) against the residuals of B (regressed on C). What do you see?

```
residuals_df <- data.frame('A_residuals' = residuals_AC, 'B_residuals' = residuals_BC)

ggplot(residuals_df, aes(x = A_residuals, y = B_residuals)) +
  geom_point()+
  theme(plot.title = element_text(hjust = 0.5))+
  ggtitle("Scatterplot observations of A and B (both regressed on C) residuals") +
  xlab("A_residuals") +
  ylab("B_residuals")+
  geom_smooth(method = 'lm')
```

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



One can observe a negative linear relationship between the residuals of A and B in the scatterplot, indicating a negative correlation between the residuals of A (regressed on C) and residuals of B (regressed on C).

Use the function cor.test() to test the null hypothesis of no correlation between the residuals of A (regressed on C) and the residuals of B (regressed on C). What is your conclusion? Does this agree with your expectation

based on the underlying DAG in Figure 2?

```
# hypothesis testing
cor.test(residuals_df$A_residuals, residuals_df$B_residuals)
```

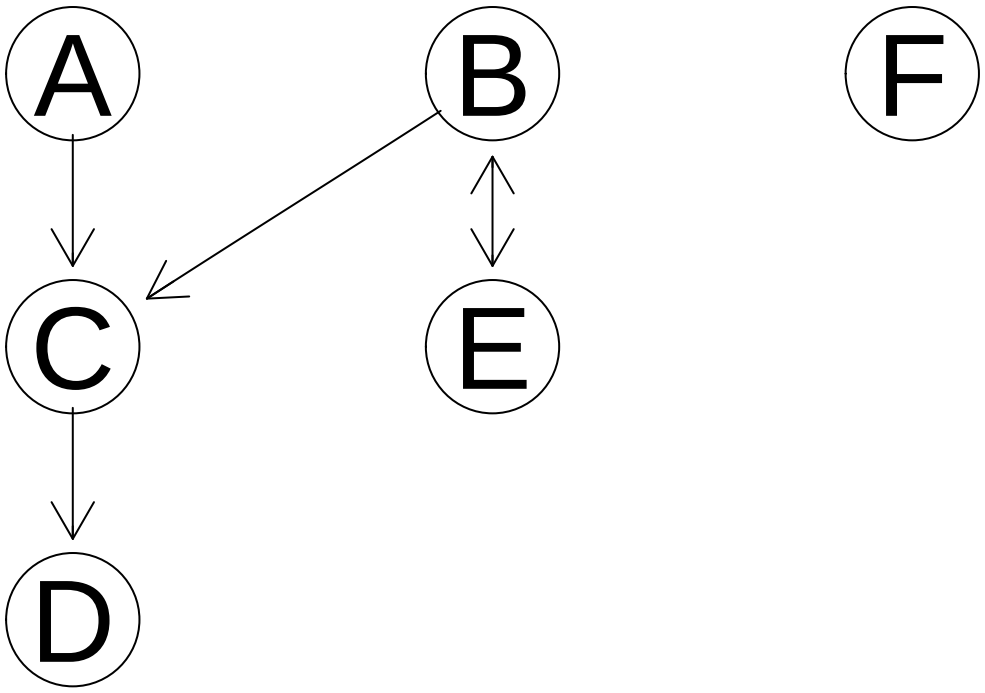
```
##
##  Pearson's product-moment correlation
##
## data:  residuals_df$A_residuals and residuals_df$B_residuals
## t = -7.5173, df = 298, p-value = 6.6e-13
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4903245 -0.2995546
## sample estimates:
##        cor
## -0.3992521
```

With the calculated p-value of 6.6e-13 which is « 5% the null hypothesis of no correlation between residuals of A and B can be rejected. This suggests that A and B residuals regressed on C are (negatively) correlated wich is in line with the scatter plot as well as the underlying DAG as A and B are not conditionally independent given C (head-to-head and C part of set C, v-structure A -> C <- B).

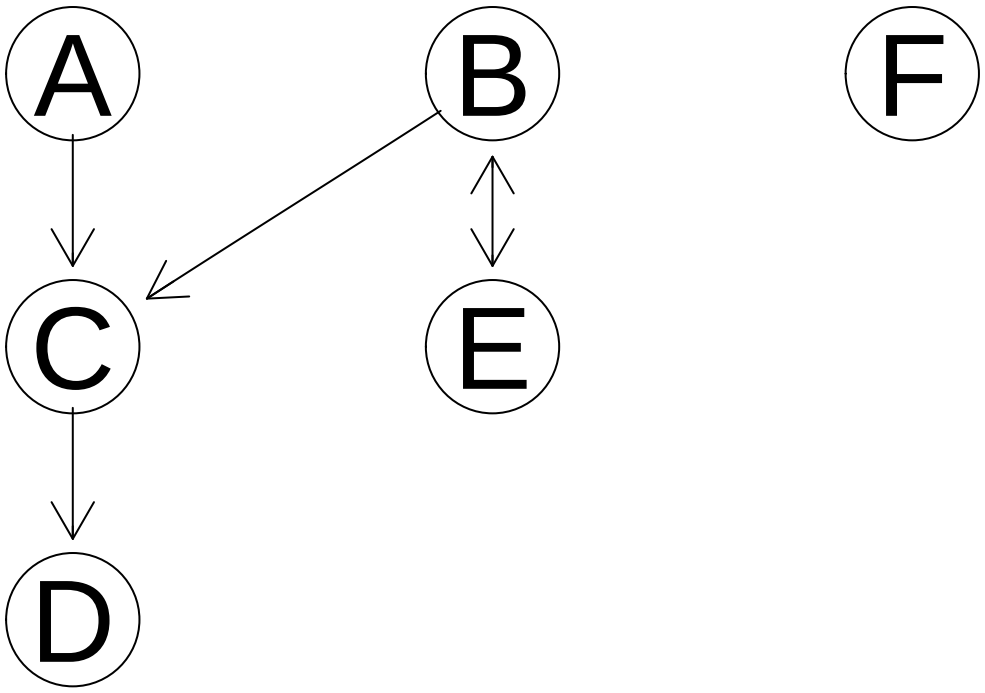
Problem 26: Running the PC algorithm

Install and load the R package pcalg. Use the function pc() to run the PC algorithm on the data in MVN DAG.rds, and plot the result. For hints, see footnote 4. Does the algorithm successfully learn the structure of the data-generating graph in Figure 2? How is the result affected by the significance level for the conditional independence tests?

```
# run PC algorithm
pc <- pc(list(C = cor(data), n = nrow(data)),
         indepTest = gaussCItest,
         alpha = 0.05,
         labels = colnames(data)
         )
plot(pc, main = "Graph structure alpha = 0.05")
```

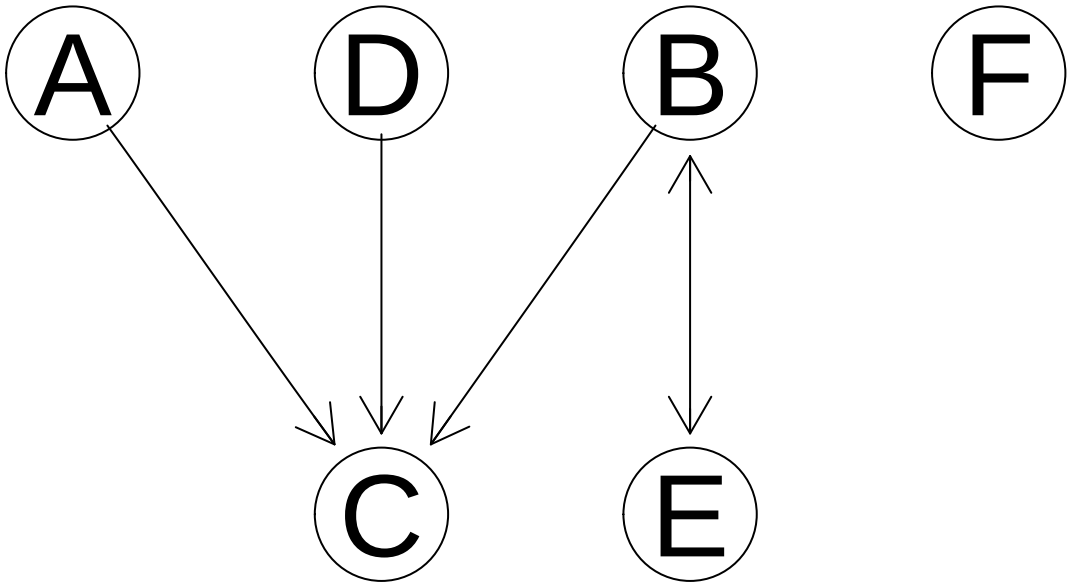



```
# try out different alphas
pc_0.01 <- pc(list(C = cor(data), n = nrow(data)),
         indepTest = gaussCItest,
         alpha = 0.01,
         labels = colnames(data))
plot(pc_0.01, main = "Graph structure alpha = 0.01")
```

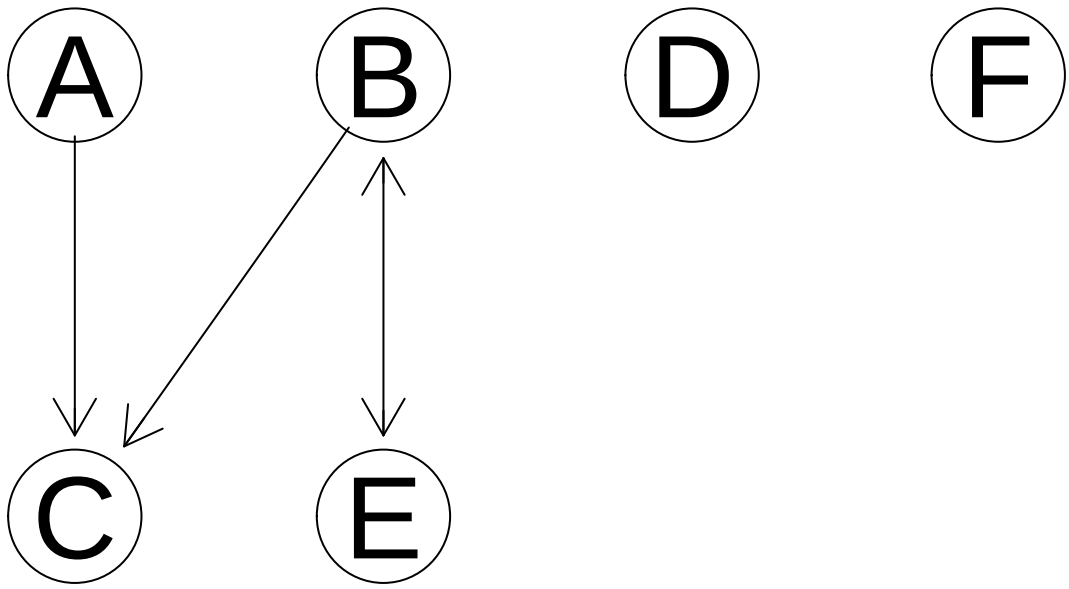



```
pc_0.001 <- pc(list(C = cor(data), n = nrow(data)),
         indepTest = gaussCItest,
         alpha = 0.001,
         labels = colnames(data))
plot(pc_0.001, main = "Graph structure alpha = 0.001")
```

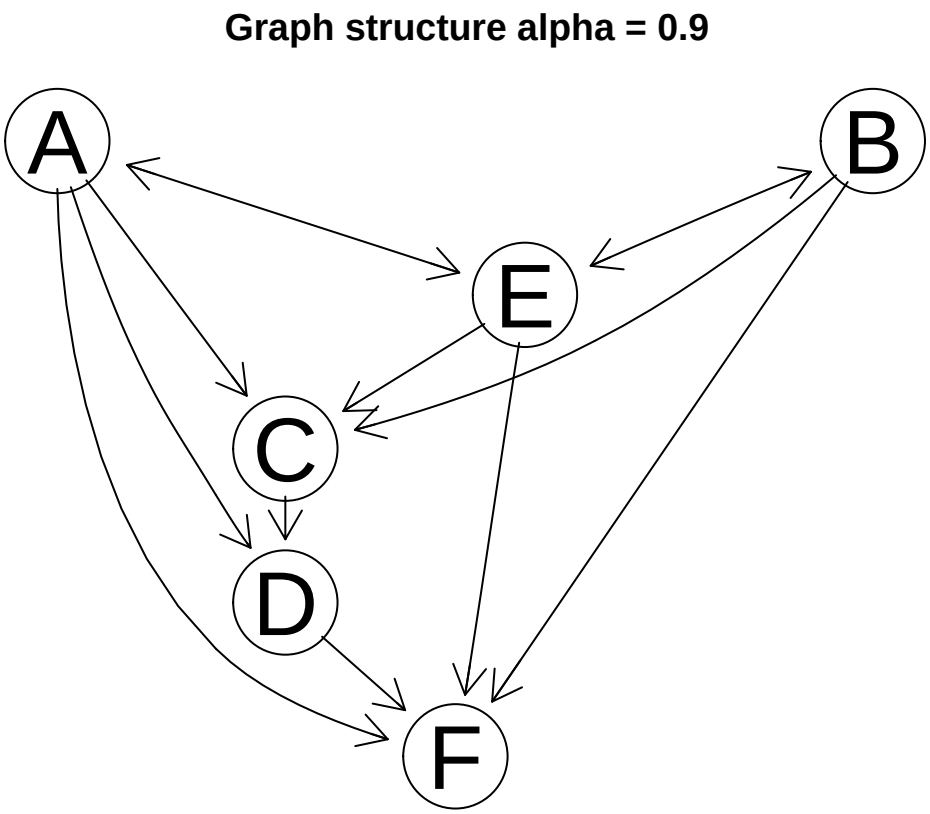
Graph structure alpha = 0.001




```
pc_0.00001 <- pc(list(C = cor(data), n = nrow(data)),
         indepTest = gaussCItest,
         alpha = 0.00001,
         labels = colnames(data))
plot(pc_0.00001, main = "Graph structure alpha = 0.00001")
```




```
pc_0.9 <- pc(list(C = cor(data), n = nrow(data)),
         indepTest = gaussCItest,
         alpha = 0.9,
         labels = colnames(data))
plot(pc_0.9, main = "Graph structure alpha = 0.9")
```



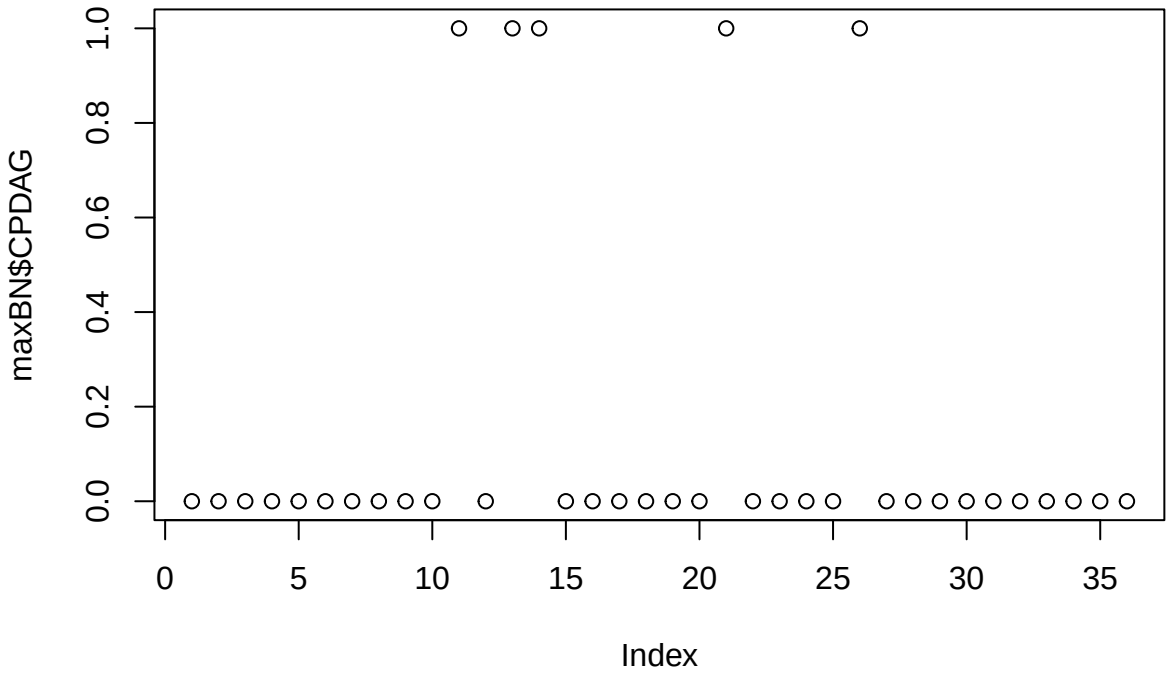
With the standard alpha = 0.05 the algorithm successfully learns the structure of the DAG in Figure 2 except the edge between B and E is wrong/ not determined. The alpha affects number of detected edges, the higher it is the more edges are possibly added to the learned DAG (see alpha = 0.9). With lower alphas up to a certain point nothing changes (see alpha = 0.01) but if we go even lower, see alpha = 0.001, the learned structure also is changed. If we would go even smaller, less edges are added (see 0.00001).

Problem 27: Running the partition MCMC algorithm

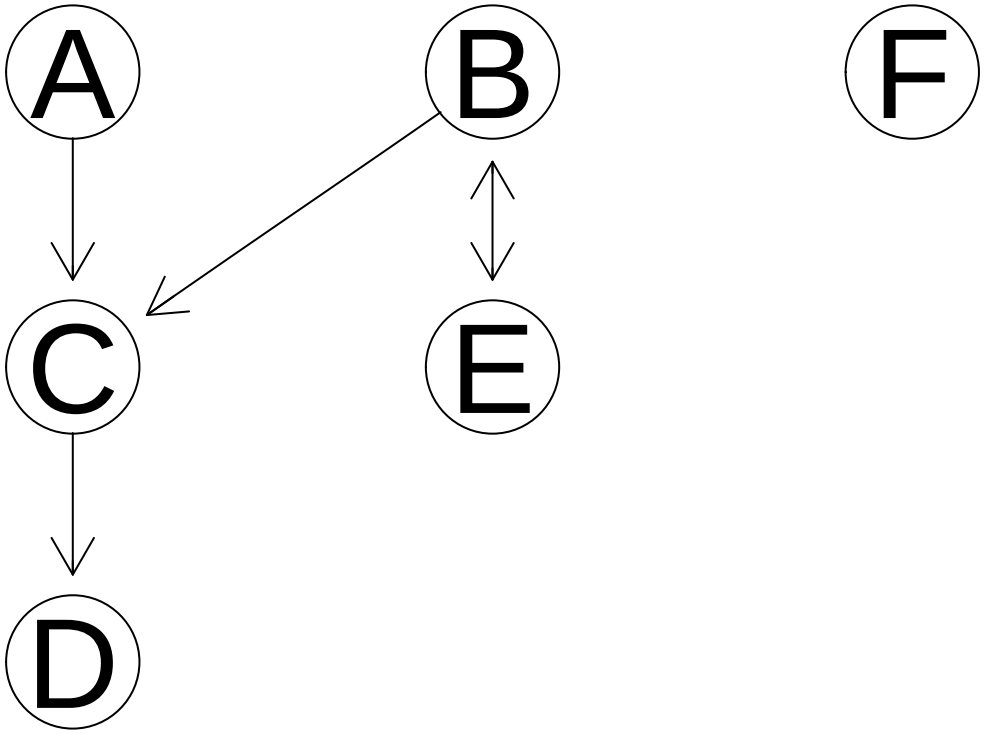
Install and load the R package BiDAG. Initialize the parameters with Score <- scoreparameters("bge", data) on the data in MVN DAG.rds using the Bayesian Gaussian equivalent (BGe) score. Run the iterative MCMC algorithm with maxBN <- learnBN(Score, algorithm = "orderIter") to learn the maximum scoring DAG and plot its equivalence class maxBN$CPDAG. How is the result affected by the hyper-parameter $d\mu$? Run the partition MCMC algorithm with partitionsample <- sampleBN(Score, algorithm = "partition", startspace = maxBNendspace) in order to sample from the posterior distribution over graph structures, where the search space is determined from the maximum scoring DAG. Compute the marginal posterior probabilities of the edges with edgesposterior <-edgep(partitionsample, pdag=TRUE) and plot them in a heatmap.

```
Score <- scoreparameters("bge", data)
maxBN <- learnBN(Score, algorithm = "orderIter")
```

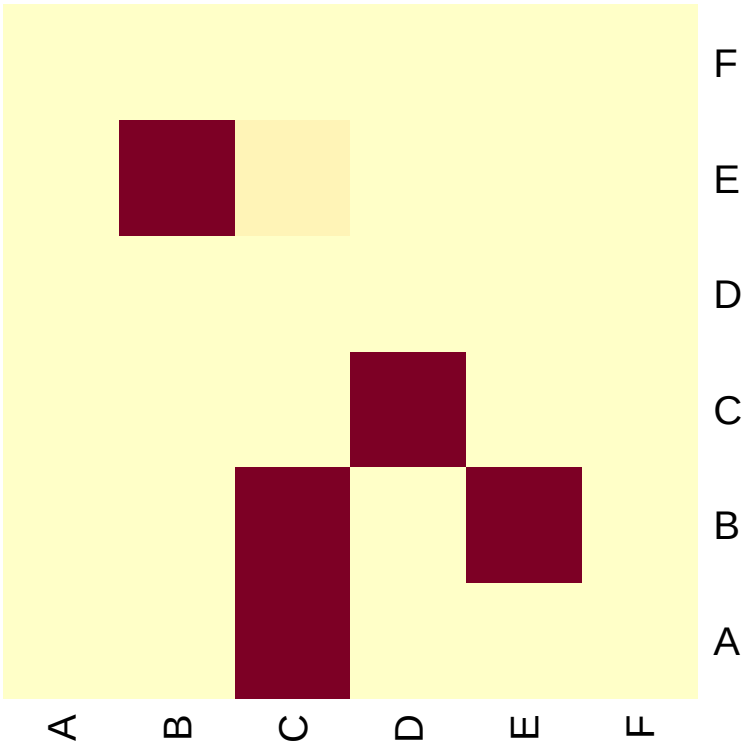
```
plot(maxBN$CPDAG)
```



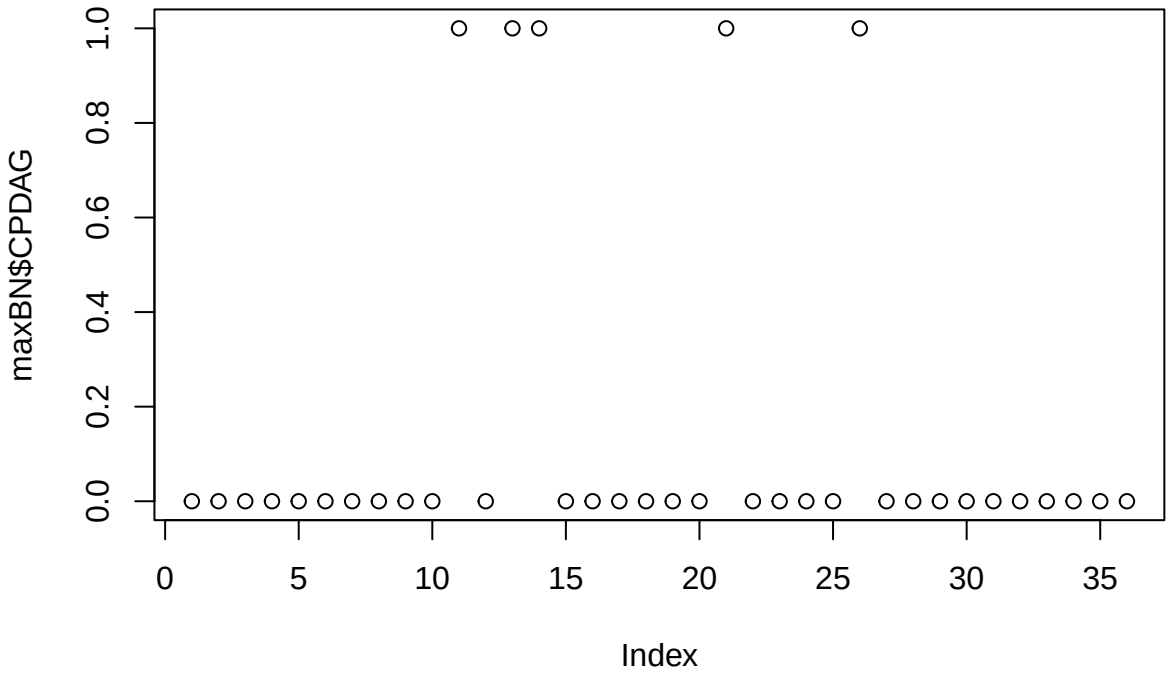

```
plot(graphAM(as.matrix(maxBN$CPDAG),edgemode = "directed"))
```



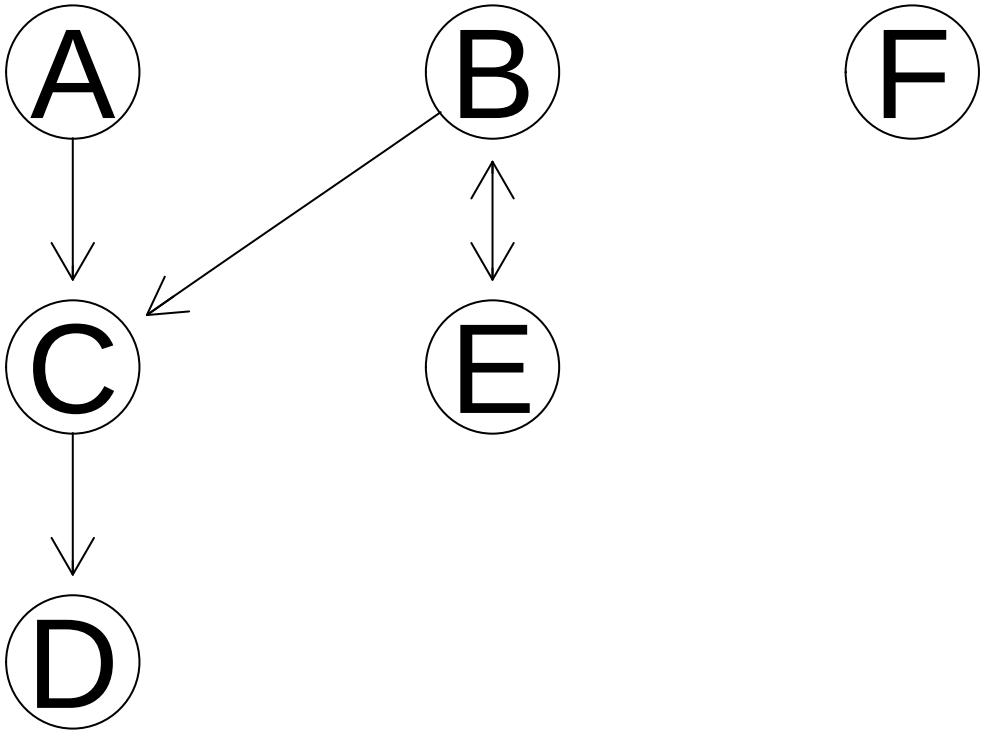

```
partitionsample <- sampleBN(Score, algorithm = "partition", startspace = maxBN$endspace)
edgesposterior <- edgep(partitionsample, pdag=TRUE)
heatmap(edgesposterior, Colv=NA, Rowv=NA, scale='none')
```



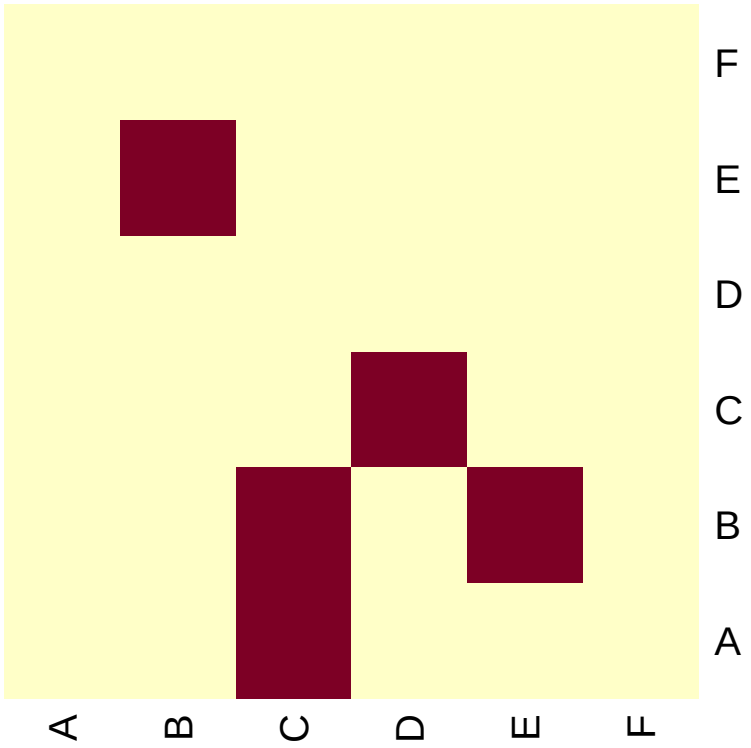

```
#try out for different _
Score <- scoreparameters("bge", data,  bgepar = list(am = 0.1, aw = NULL))
maxBN <- learnBN(Score, algorithm = "orderIter")
plot(maxBN$CPDAG)
```



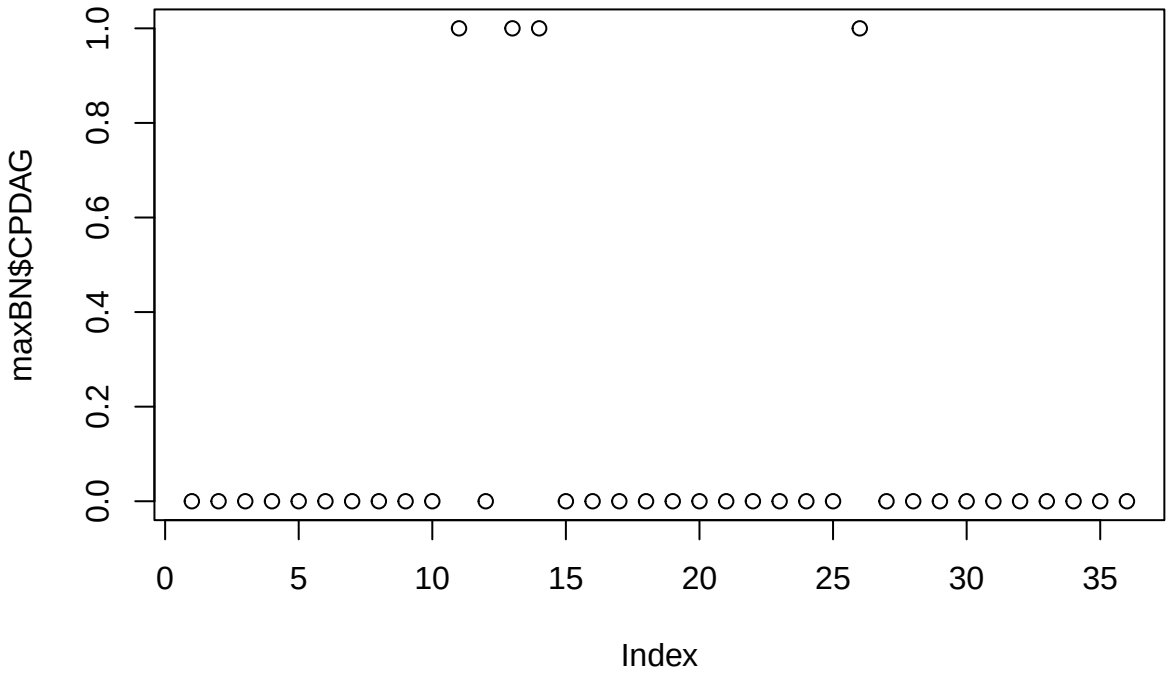

```
plot(graphAM(as.matrix(maxBN$CPDAG),edgemode = "directed"))
```



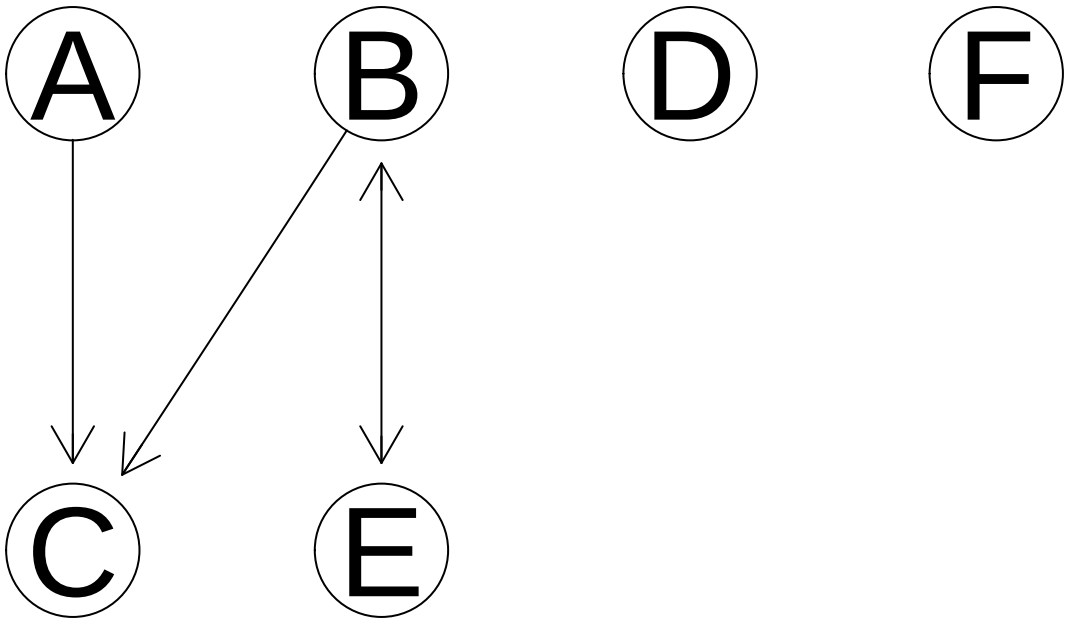

```
partitionsample <- sampleBN(Score, algorithm = "partition", startspace = maxBN$endspace)
edgesposterior <-edgep(partitionsample, pdag=TRUE)
heatmap(edgesposterior, Colv=NA, Rowv=NA, scale='none')
```



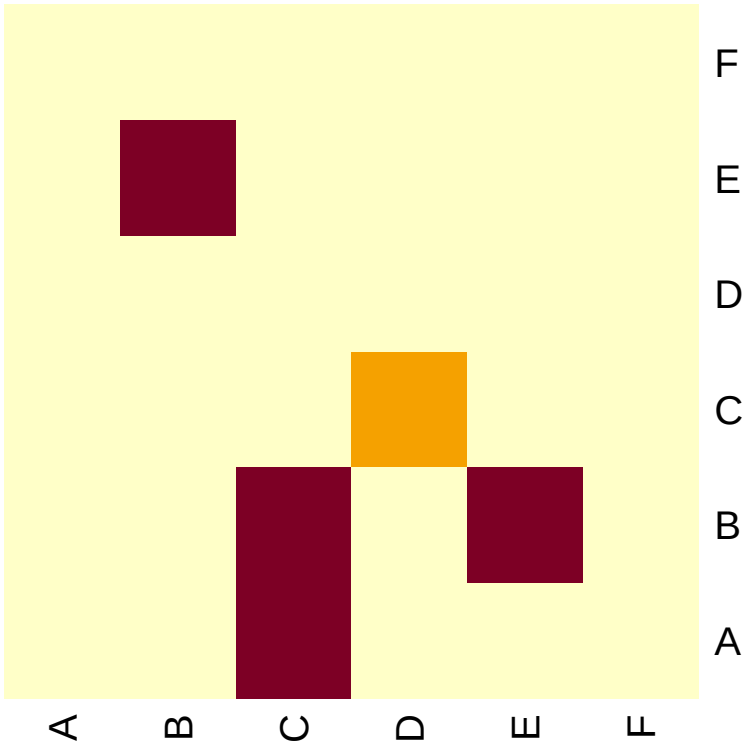

```
Score <- scoreparameters("bge", data,  bgepar = list(am = 0.001, aw = NULL))
maxBN <- learnBN(Score, algorithm = "orderIter")
plot(maxBN$CPDAG)
```



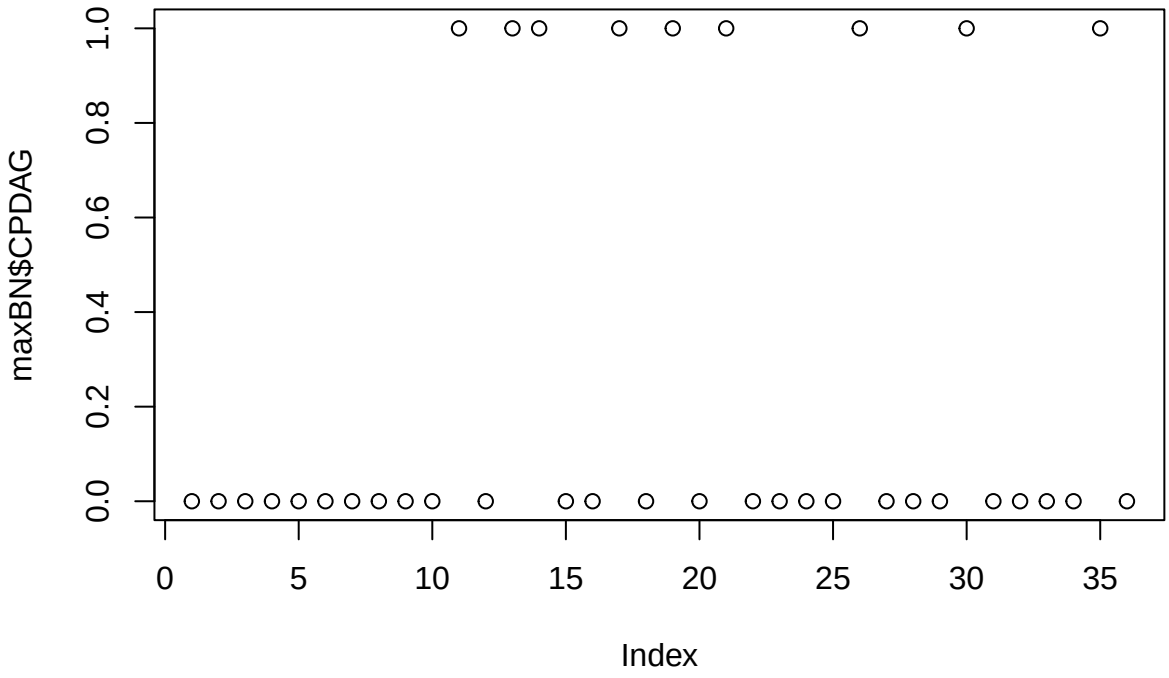

```
plot(graphAM(as.matrix(maxBN$CPDAG),edgemode = "directed"))
```



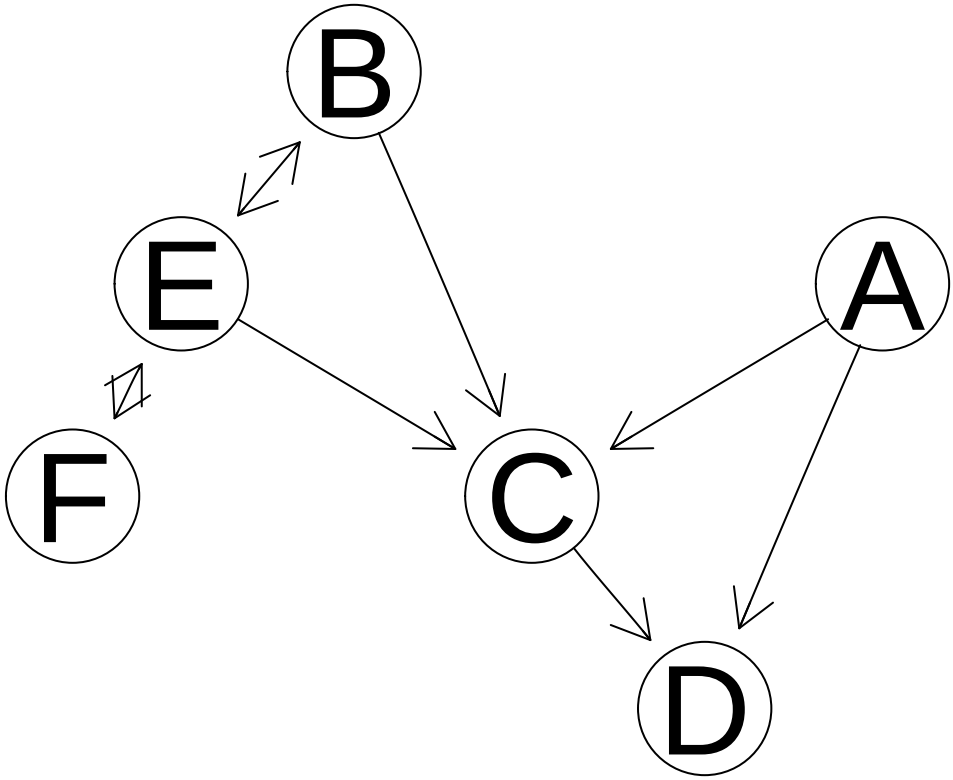

```
partitionsample <- sampleBN(Score, algorithm = "partition", startspace = maxBN$endspace)
edgesposterior <-edgep(partitionsample, pdag=TRUE)
heatmap(edgesposterior, Colv=NA, Rowv=NA, scale='none')
```



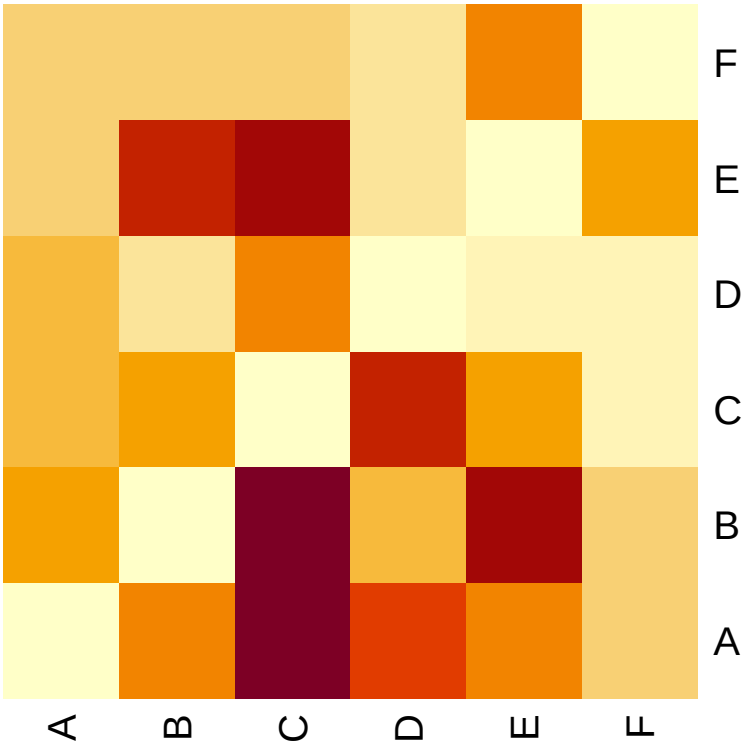

```
Score <- scoreparameters("bge", data, bgepar = list(am = 100, aw = NULL))
maxBN <- learnBN(Score, algorithm = "orderIter")
plot(maxBN$CPDAG)
```




```
plot(graphAM(as.matrix(maxBN$CPDAG),edgemode = "directed"))
```




```
partitionsample <- sampleBN(Score, algorithm = "partition", startspace = maxBN$endspace)
edgesposterior <-edgep(partitionsample, pdag=TRUE)
heatmap(edgesposterior, Colv=NA, Rowv=NA, scale='none')
```



The higher the α_μ, the denser the resulting graph and also the higher the marginal posterior probabilites of the edges (seen in darker rectangles in the heatmap).

```
library(rmarkdown)
rmarkdown::render("project9.Rmd", output_format = "pdf_document")
```