Problem 1 : Consider the "ksl" data in the "deal" package. Eliminate surveyyear.

```
library(deal)
library(bnlearn)
```

```
##
## Attaching package: 'bnlearn'
```

```
## The following objects are masked from 'package:deal':
##
##     modelstring, nodes, nodes<-, score
```

```
data("ksl")
head(ksl)
```

```
##   FEV Kol Hyp   logBMI Smok Alc Work Sex Year
## 1 116 745   1 3.360375    1   2    2   2    2
## 2 252 680   1 3.477232    2   2    2   1    2
## 3 205 669   1 3.285038    2   1    2   2    2
## 4  66 636   1 3.117950    2   1    2   2    2
## 5 255 669   1 3.213260    2   1    2   1    2
## 6 135 555   0 3.474138    2   1    2   2    2
```

```
dim(ksl)
```

```
## [1] 1083    9
```

Let us remove the survey year from the dataset :

```
ksl <- ksl[,-9]
head(ksl)
```

```
##   FEV Kol Hyp   logBMI Smok Alc Work Sex
## 1 116 745   1 3.360375    1   2    2   2
## 2 252 680   1 3.477232    2   2    2   1
## 3 205 669   1 3.285038    2   1    2   2
## 4  66 636   1 3.117950    2   1    2   2
## 5 255 669   1 3.213260    2   1    2   1
## 6 135 555   0 3.474138    2   1    2   2
```

```
dim(ksl)
```

```
## [1] 1083    8
```

(a) Discretize the data – justify your choice of discretization.

Let's first use log transformation to convert the integer columns in the data into continuous values before discretizing it.

```
#Forced Ejection Volume : Convert integer into continous :
ksl$FEV <- log(ksl$FEV)

#Cholesterol : Convert integer into continous :
ksl$Kol <- log(ksl$Kol)

#Hypertension : Convert integer into factor
ksl$Hyp <- as.factor(ksl$Hyp)
```

```
head(ksl)
```

```
##        FEV      Kol Hyp   logBMI Smok Alc Work Sex
## 1 4.753590 6.613384   1 3.360375    1   2    2   2
## 2 5.529429 6.522093   1 3.477232    2   2    2   1
## 3 5.323010 6.505784   1 3.285038    2   1    2   2
## 4 4.189655 6.455199   1 3.117950    2   1    2   2
## 5 5.541264 6.505784   1 3.213260    2   1    2   1
## 6 4.905275 6.318968   0 3.474138    2   1    2   2
```

Let us apply discretize function to Discretize the data , Let us use “interval” method so that we have two groups of intervals so that we can later rename them into high and low.

```
discrete_df1 <- discretize(ksl, method = "interval", breaks = 2)
head(discrete_df1)
```

```
##                  FEV                Kol Hyp            logBMI Smok Alc Work Sex
## 1 [3.63759,4.78624] (6.43318,7.20341]   1 (3.30433,3.84738]    1   2    2   2
## 2 (4.78624,5.93489] (6.43318,7.20341]   1 (3.30433,3.84738]    2   2    2   1
## 3 (4.78624,5.93489] (6.43318,7.20341]   1 [2.76127,3.30433]    2   1    2   2
## 4 [3.63759,4.78624] (6.43318,7.20341]   1 [2.76127,3.30433]    2   1    2   2
## 5 (4.78624,5.93489] (6.43318,7.20341]   1 [2.76127,3.30433]    2   1    2   1
## 6 (4.78624,5.93489] [5.66296,6.43318]   0 (3.30433,3.84738]    2   1    2   2
```

```
levels(discrete_df1$FEV) <- c("low","high")
levels(discrete_df1$Kol) <- c("low","high")
levels(discrete_df1$logBMI) <- c("low","high")
```

Transformed Data :

```
head(discrete_df1)
```

```
##    FEV  Kol Hyp logBMI Smok Alc Work Sex
## 1  low high   1   high    1   2    2   2
## 2 high high   1   high    2   2    2   1
## 3 high high   1    low    2   1    2   2
## 4  low high   1    low    2   1    2   2
## 5 high high   1    low    2   1    2   1
## 6 high  low   0   high    2   1    2   2
```

(b) Ensure all the variables are “factors”. Fit a multinomial Bayesian Network. Describe yourapproach. Report the structure and the parameterization.

Let us fit a Bayesian Network to the data :

Structure Learning:

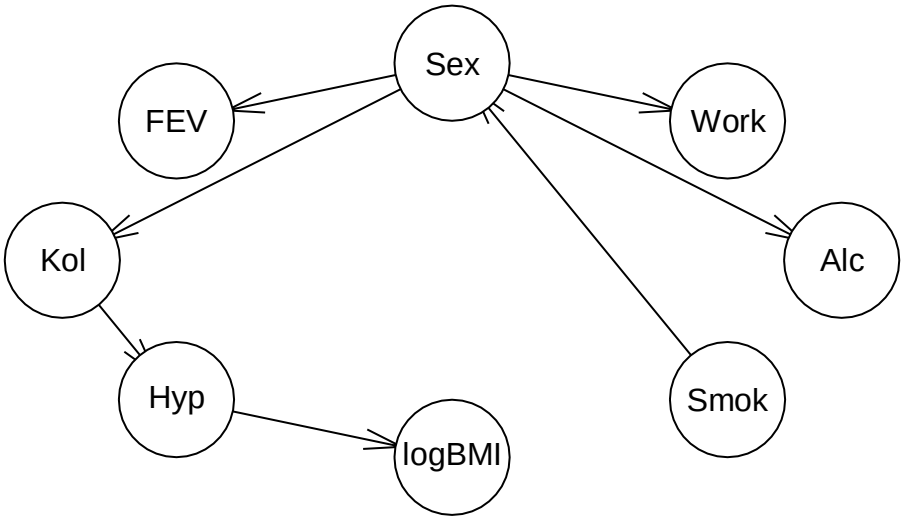
```
BN1 <- hc(discrete_df1, score = "bic")
BN1
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [Smok][Sex|Smok][FEV|Sex][Kol|Sex][Alc|Sex][Work|Sex][Hyp|Kol][logBMI|Hyp]
##   nodes:                                 8
##   arcs:                                  7
##     undirected arcs:                     0
```

```
##     directed arcs:                          7
##   average markov blanket size:              1.75
##   average neighbourhood size:               1.75
##   average branching factor:                 0.88
##
##   learning algorithm:                       Hill-Climbing
##   score:                                    BIC (disc.)
##   penalization coefficient:                 3.493745
##   tests used in the learning procedure:     77
##   optimized:                                TRUE
```

The following is the best structure found using the BIC score-based indexing and the hill climbing method:

```
plot(BN1)
```



Parametre Leaning :

Let's use the maximum likelihood estimation approach to parametrize the trained network:

```
bn_fit1 <- bn.fit(BN1,data = discrete_df1,method = "mle")
```

Parameters of the model fitted :

```
bn_fit1
```

```
##
##   Bayesian network parameters
##
##   Parameters of node FEV (multinomial distribution)
##
## Conditional probability table:
##
##       Sex
## FEV            1         2
##   low  0.1113074 0.2340426
##   high 0.8886926 0.7659574
##
##   Parameters of node Kol (multinomial distribution)
##
## Conditional probability table:
##
```

```
##       Sex
## Kol            1         2
##   low  0.4328622 0.2050290
##   high 0.5671378 0.7949710
##
##   Parameters of node Hyp (multinomial distribution)
##
## Conditional probability table:
##
##    Kol
## Hyp       low      high
##   0 0.5128205 0.4193989
##   1 0.4871795 0.5806011
##
##   Parameters of node logBMI (multinomial distribution)
##
## Conditional probability table:
##
##       Hyp
## logBMI         0         1
##   low  0.8172485 0.6174497
##   high 0.1827515 0.3825503
##
##   Parameters of node Smok (multinomial distribution)
##
## Conditional probability table:
##          1         2
## 0.2825485 0.7174515
##
##   Parameters of node Alc (multinomial distribution)
##
## Conditional probability table:
##
##    Sex
## Alc         1         2
##   1 0.3533569 0.6460348
##   2 0.6466431 0.3539652
##
##   Parameters of node Work (multinomial distribution)
##
## Conditional probability table:
##
##     Sex
## Work          1          2
##    1 0.21554770 0.07736944
##    2 0.78445230 0.92263056
##
##   Parameters of node Sex (multinomial distribution)
##
## Conditional probability table:
##
##    Smok
## Sex         1         2
##   1 0.2222222 0.6409266
```

```
##    2 0.7777778 0.3590734
```

The above reports the conditional probability tables for all the nodes in the network.

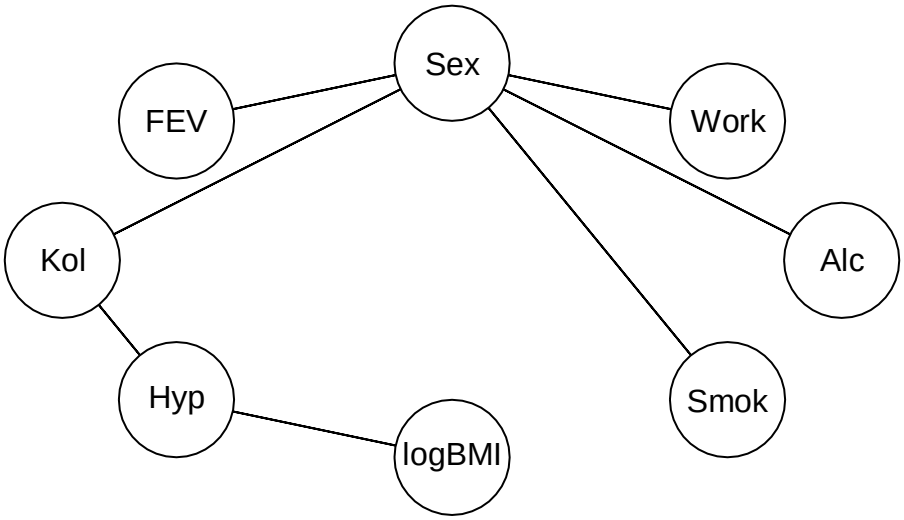
(c) Find the CPDAG for your network in B.

To find the CPDAG of the network :

```
CP_DAG1 <- cpdag(bn_fit1)
CP_DAG1
```

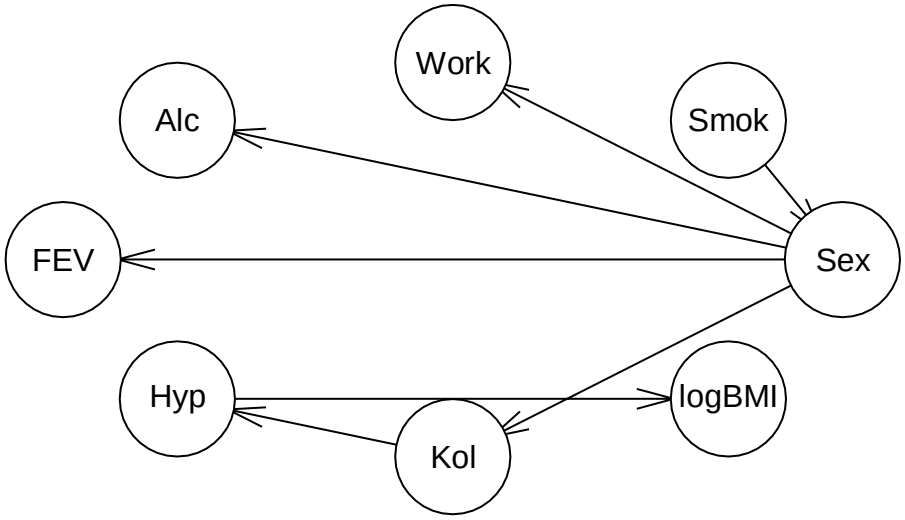
```
##
##   Random/Generated Bayesian network
##
##   model:
##     [undirected graph]
##   nodes:                                8
##   arcs:                                 7
##     undirected arcs:                    7
##     directed arcs:                      0
##   average markov blanket size:          1.75
##   average neighbourhood size:           1.75
##   average branching factor:             0.00
##
##   generation algorithm:                 Empty
```

```
plot(CP_DAG1)
```

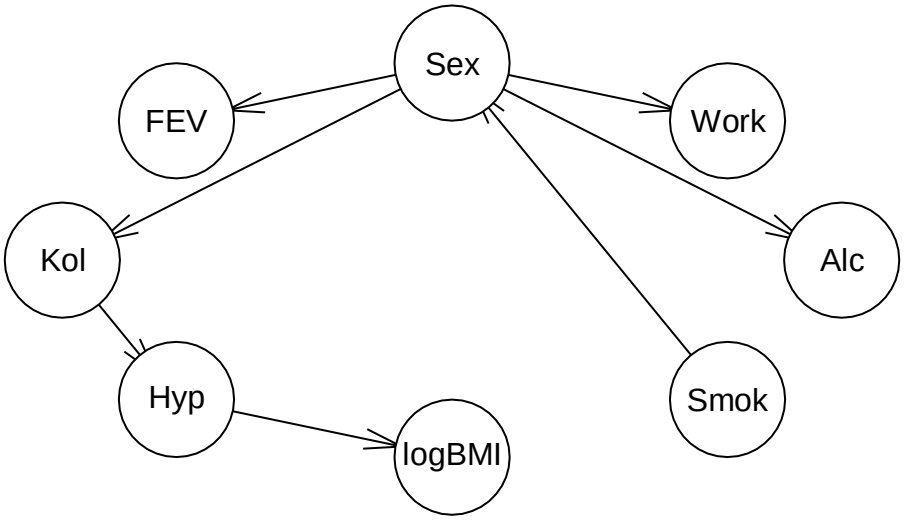


(d) Find an alternative DAG generated from the CPDAG that is a member of the equivalence same equivalence class as your DAG in b.

```
Alternative_DAG <- model2network("[Smok][Sex|Smok][FEV|Sex][Kol|Sex][Alc|Sex][Work|Sex][Hyp|Kol][logBMI
plot(Alternative_DAG)
```




```
alt <- cextend(BN1)
plot(alt)
```



(e) **According to your BN, are individuals that consume alcohol and smoke more likely to have higher forced ejection volumes? Justify your answer.**

In this case ,

The event is *FEV = High* The evidence is *Alcohol = 2 (Frequent) and smoke = 2 (Yes)*

```
cpquery(bn_fit1 , event = (FEV == "high") , evidence = (Alc == 2) & (Smok == 2))
```

```
## [1] 0.8502004
```

According to the information above, there is an 84.3 percent likelihood that someone who regularly uses tobacco and alcohol will have a larger forced ejection volume.
so that the response can be supported Yes.

(f) According to your BN, are individuals that consume alcohol and have hypertension morelikely to have low forced ejection volumes? Justify your answer

In this case ,

The event is *FEV = low* The evidence is *Alcohol = 2 (Frequent) and hypertension = 1 (Yes)*

```
cpquery(bn_fit1 , event = (FEV == "low") , evidence = (Alc == 2) & (Hyp == 1))
```

```
## [1] 0.1638341
```

The reply is no. According to the information above, there is only a 14.6% likelihood that someone who drinks and has hypertension will have poor forced ejection volumes.

(g) Discretize the data using an alternative technique (vs what was done in part A). Learn the BN using your approach in part B. How does it differ from the BN learned in Part B? Is the CPDAG different?

Instead of using interval method, we will use the hartemink method

```
discrete_df2 <- discretize(ksl, method = "hartemink", breaks = 2)
head(discrete_df2)
```

```
##                FEV               Kol Hyp            logBMI Smok Alc Work Sex
## 1 [3.63759,5.4161] [5.66296,6.70401]   1 [2.76127,3.43978]    1   2    2   2
## 2 (5.4161,5.93489] [5.66296,6.70401]   1 (3.43978,3.84738]    2   2    2   1
## 3 [3.63759,5.4161] [5.66296,6.70401]   1 [2.76127,3.43978]    2   1    2   2
## 4 [3.63759,5.4161] [5.66296,6.70401]   1 [2.76127,3.43978]    2   1    2   2
## 5 (5.4161,5.93489] [5.66296,6.70401]   1 [2.76127,3.43978]    2   1    2   1
## 6 [3.63759,5.4161] [5.66296,6.70401]   0 (3.43978,3.84738]    2   1    2   2
```

```
levels(discrete_df2$FEV)  <- c("low","high")
levels(discrete_df2$Kol)  <- c("low","high")
levels(discrete_df2$logBMI)  <- c("low","high")
head(discrete_df2)
```

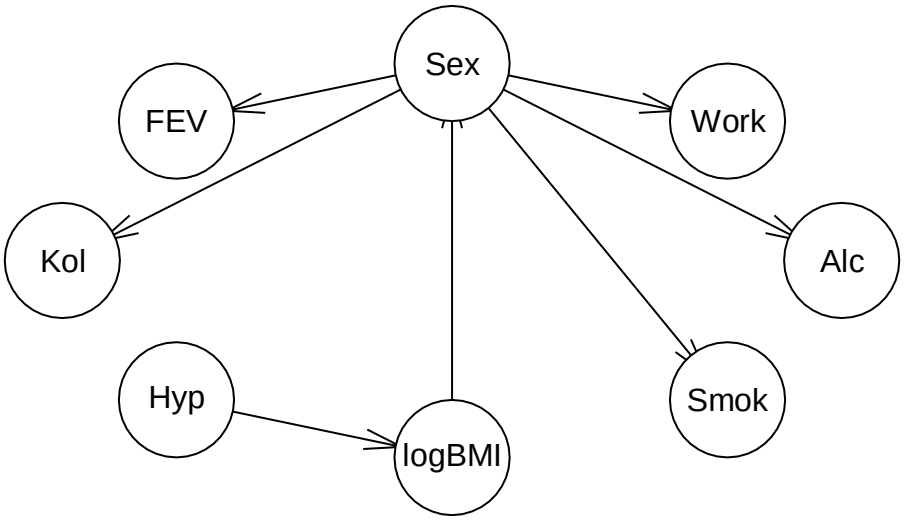
```
##    FEV Kol Hyp logBMI Smok Alc Work Sex
## 1  low low   1    low    1   2    2   2
## 2 high low   1   high    2   2    2   1
## 3  low low   1    low    2   1    2   2
## 4  low low   1    low    2   1    2   2
## 5 high low   1    low    2   1    2   1
## 6  low low   0   high    2   1    2   2
```

```
BN2 <- hc(discrete_df2, score = "bic")
BN2
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [Hyp][logBMI|Hyp][Sex|logBMI][FEV|Sex][Kol|Sex][Smok|Sex][Alc|Sex][Work|Sex]
##   nodes:                                 8
##   arcs:                                  7
##     undirected arcs:                     0
##     directed arcs:                       7
##   average markov blanket size:           1.75
##   average neighbourhood size:            1.75
##   average branching factor:              0.88
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 BIC (disc.)
##   penalization coefficient:              3.493745
##   tests used in the learning procedure:  91
```

```
##   optimized:                              TRUE
```

```
plot(BN2)
```

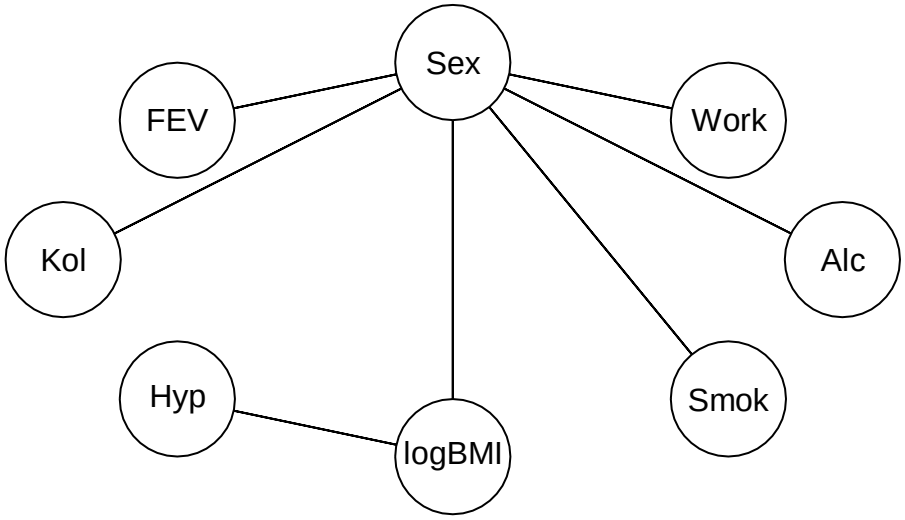



```
bn_fit2 <- bn.fit(BN2,data = discrete_df1,method = "mle")
bn_fit2
```

```
##
##   Bayesian network parameters
##
##   Parameters of node FEV (multinomial distribution)
##
## Conditional probability table:
##
##       Sex
## FEV            1         2
##   low  0.1113074 0.2340426
##   high 0.8886926 0.7659574
##
##   Parameters of node Kol (multinomial distribution)
##
## Conditional probability table:
##
##       Sex
## Kol            1         2
##   low  0.4328622 0.2050290
##   high 0.5671378 0.7949710
##
##   Parameters of node Hyp (multinomial distribution)
##
## Conditional probability table:
##          0         1
## 0.4496768 0.5503232
##
##   Parameters of node logBMI (multinomial distribution)
##
## Conditional probability table:
##
```

```
##       Hyp
## logBMI         0         1
##   low  0.8172485 0.6174497
##   high 0.1827515 0.3825503
##
##   Parameters of node Smok (multinomial distribution)
##
## Conditional probability table:
##
##     Sex
## Smok         1         2
##    1 0.1201413 0.4603482
##    2 0.8798587 0.5396518
##
##   Parameters of node Alc (multinomial distribution)
##
## Conditional probability table:
##
##    Sex
## Alc         1         2
##   1 0.3533569 0.6460348
##   2 0.6466431 0.3539652
##
##   Parameters of node Work (multinomial distribution)
##
## Conditional probability table:
##
##     Sex
## Work          1          2
##    1 0.21554770 0.07736944
##    2 0.78445230 0.92263056
##
##   Parameters of node Sex (multinomial distribution)
##
## Conditional probability table:
##
##    logBMI
## Sex       low      high
##   1 0.5522193 0.4511041
##   2 0.4477807 0.5488959
```

```r
CP_DAG2 <- cpdag(bn_fit2)
plot(CP_DAG2)
```



The model trained and the cpdag are the identical for both scenarios even though the data were discretized using two distinct ways.

```
compare(BN1,BN2)
```

```
## $tp
## [1] 5
##
## $fp
## [1] 2
##
## $fn
## [1] 2
```

Problem 2 : Consider the "marks" data in the "bnlearn" package

```
library(bnlearn)
df <- bnlearn::marks
dim(df)
```

```
## [1] 88  5
```

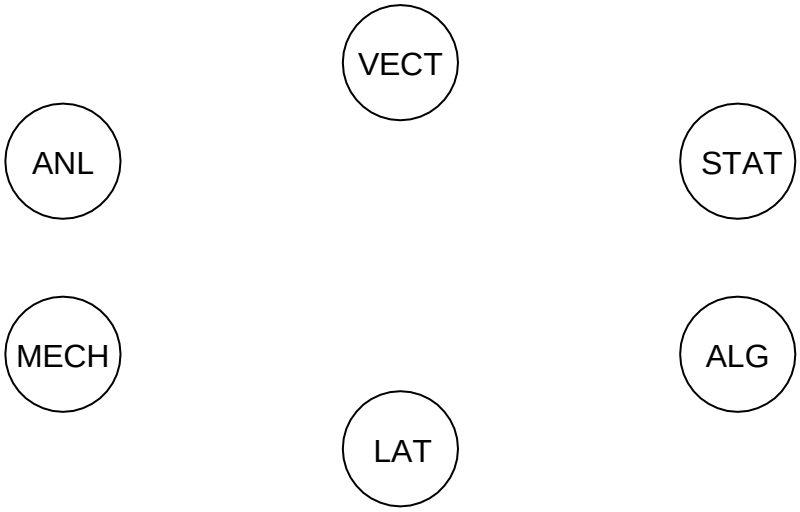
```
head(df)
```

```
##   MECH VECT ALG ANL STAT
## 1   77   82  67  67   81
## 2   63   78  80  70   81
## 3   75   73  71  66   81
## 4   55   72  63  70   68
## 5   63   63  65  70   63
## 6   53   61  72  64   73
```

(a) Create a bn object describing the below graph:

We establish an empty graph with the designated nodes in order to generate the bayesian network.

```
dag <- empty.graph(nodes = c("ANL","MECH","LAT","ALG","STAT","VECT"))
plot(dag)
```



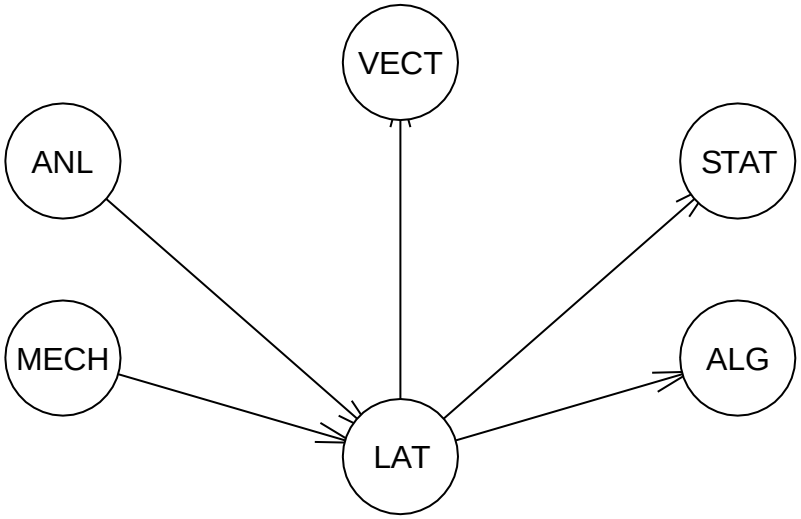
Let us specify the edges :

```
dag <- set.arc(dag, from = "MECH", to = "LAT")
dag <- set.arc(dag, from = "ANL", to = "LAT")
dag <- set.arc(dag, from = "LAT", to = "ALG")
dag <- set.arc(dag, from = "LAT", to = "STAT")
dag <- set.arc(dag, from = "LAT", to = "VECT")
dag
```

```
##
##   Random/Generated Bayesian network
##
##   model:
##    [ANL][MECH][LAT|ANL:MECH][ALG|LAT][STAT|LAT][VECT|LAT]
##   nodes:                                 6
##   arcs:                                  5
##     undirected arcs:                     0
```

```
##     directed arcs:                      5
##   average markov blanket size:          2.00
##   average neighbourhood size:           1.67
##   average branching factor:             0.83
##
##   generation algorithm:                 Empty
```

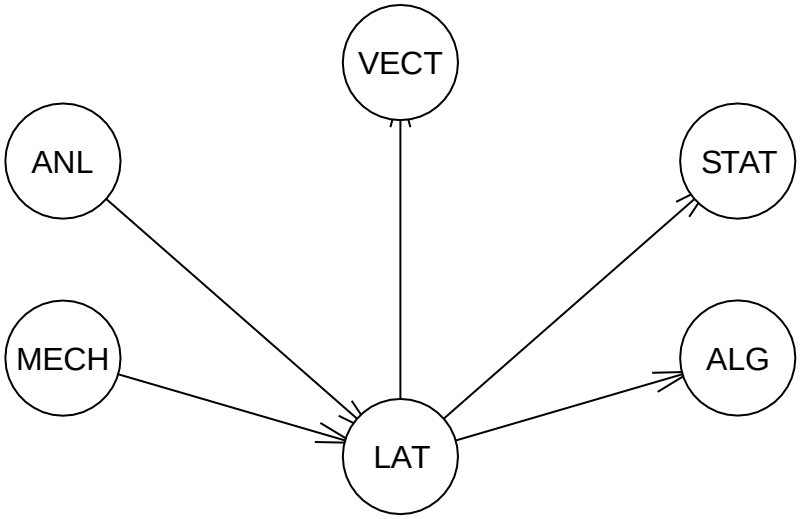
```
plot(dag)
```



(b) Find the CPDAG

To generate a CPDAG , we can use the following :

```
CP_DAG <- cpdag(dag)
plot(CP_DAG)
```




```
all.equal(dag, CP_DAG)
```

```
## [1] TRUE
```

We can see that the original network and the completed partially directed acyclic graph (CPDAG) are identical.

(c) Use hc to find the most likely structure. How does it differ from the above DAG?

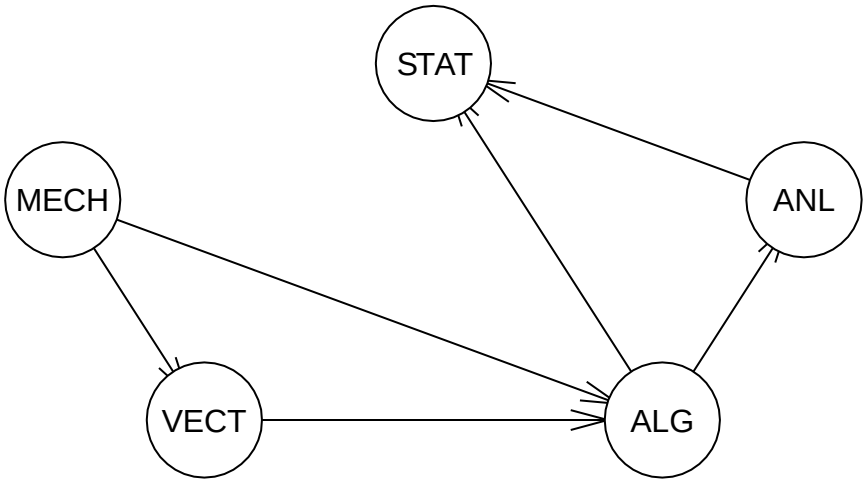
Let us apply hill climbing algorithm using score based method :

```
mls <- hc(df)
mls
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [MECH][VECT|MECH][ALG|MECH:VECT][ANL|ALG][STAT|ALG:ANL]
##   nodes:                                 5
##   arcs:                                  6
##     undirected arcs:                     0
##     directed arcs:                       6
##   average markov blanket size:           2.40
##   average neighbourhood size:            2.40
##   average branching factor:              1.20
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 BIC (Gauss.)
##   penalization coefficient:              2.238668
##   tests used in the learning procedure:  34
##   optimized:                             TRUE
```

Let us plot the most likely structure obtained from above :

```
plot(mls)
```



From the foregoing, we can conclude that the structure is most likely devoid of the node that changed the graph's edges..

```
nodes(dag)
```

```
## [1] "ANL"  "MECH" "LAT"  "ALG"  "STAT" "VECT"
```

```
nodes(mls)
```

```
## [1] "MECH" "VECT" "ALG"  "ANL"  "STAT"
```

```
modelstring(dag)
```

```
## [1] "[ANL][MECH][LAT|ANL:MECH][ALG|LAT][STAT|LAT][VECT|LAT]"
```

```
modelstring(mls)
```

```
## [1] "[MECH][VECT|MECH][ALG|MECH:VECT][ANL|ALG][STAT|ALG:ANL]"
```

It is possible to see above how the distributions differ. Additionally, we can detect some fluctuation in the arc measurement.

Problem 3 : The "carcass" data from the package "gRbase" contains data on meat. Specifically, the data describes the thickness of meat and fat layers in different regions on the back of a pig together with the lean meat percentage on each of 344 carcasses. The data has been used for prediction of lean meat percentage based on carcass thickness.

```
library(gRbase)
```

```
##
## Attaching package: 'gRbase'
## The following objects are masked from 'package:bnlearn':
##
##     ancestors, children, nodes, parents
## The following object is masked from 'package:deal':
##
##     nodes
```

```
data("carcass")
head(carcass)
```

```
##   Fat11 Meat11 Fat12 Meat12 Fat13 Meat13 LeanMeat
## 1    17     51    12     51    12     61 56.52475
## 2    17     49    15     48    15     54 57.57958
## 3    14     38    11     34    11     40 55.88994
## 4    17     58    12     58    11     58 61.81719
## 5    14     51    12     48    13     54 62.95964
## 6    20     40    14     40    14     45 54.57870
```

let us transform the data into log scale to make it continous :

```
carcass$Fat11 <- log(carcass$Fat11)
carcass$Meat11 <- log(carcass$Meat11)
carcass$Fat12 <- log(carcass$Fat12)
carcass$Meat12 <- log(carcass$Meat12)
carcass$Fat13 <- log(carcass$Fat13)
carcass$Meat13 <- log(carcass$Meat13)
```

```
head(carcass)
```

```
##      Fat11   Meat11    Fat12   Meat12    Fat13   Meat13 LeanMeat
## 1 2.833213 3.931826 2.484907 3.931826 2.484907 4.110874 56.52475
## 2 2.833213 3.891820 2.708050 3.871201 2.708050 3.988984 57.57958
## 3 2.639057 3.637586 2.397895 3.526361 2.397895 3.688879 55.88994
## 4 2.833213 4.060443 2.484907 4.060443 2.397895 4.060443 61.81719
## 5 2.639057 3.931826 2.484907 3.871201 2.564949 3.988984 62.95964
## 6 2.995732 3.688879 2.639057 3.688879 2.639057 3.806662 54.57870
```

(a) Create a BN using score-based structural learning and ensure that "Lean Meat" is at the bottom of the network.

Learn the structure of the network using score based method :

```
BN3 <- hc(carcass, score = "bge")
BN3
```

```
##
```

```
##   Bayesian network learned via Score-based methods
##
##   model:
##    [Fat11][Meat11][Meat12|Fat11:Meat11][Meat13|Fat11:Meat11:Meat12]
##    [LeanMeat|Fat11:Meat11:Meat12][Fat13|Fat11:Meat11:Meat12:Meat13:LeanMeat]
##    [Fat12|Fat11:Meat11:Meat12:Fat13:Meat13:LeanMeat]
##   nodes:                                 7
##   arcs:                                  19
##     undirected arcs:                     0
##     directed arcs:                       19
##   average markov blanket size:           6.00
##   average neighbourhood size:            5.43
##   average branching factor:              2.71
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Bayesian Gaussian (BGe)
##   graph prior:                           Uniform
##   imaginary sample size (normal):        1
##   imaginary sample size (Wishart):       9
##   tests used in the learning procedure:  159
##   optimized:                             TRUE
```

(b) Create a BN using conditional independence tests for structural learning and ensure that "Lean Meat" is at the bottom of the network.

```{r}
data(carcass)
carcass <- data.frame(lapply(carcass, as.numeric))
bn_ci <- hc(carcass, target = "LeanMeat", algorithm = "tabu")
graphviz.plot(bn_ci)
```

(c) How do the networks in A-B compare?

The networks found in parts (a) and (b) have various structural compositions. The conditional independence tests for structural learning algorithm (part b) estimates conditional independence relationships, whereas the score-based structural learning algorithm (part a) employs a scoring function to discover the optimal structure. As we can see from the aforementioned pictures, the resulting networks are completely unique, have various arc structures, and represent various assumptions on the conditional dependencies in the data. While figure 2 has a nested structure, figure 1 has a more linear structure.

(d) How does the model compare with A?

```{r}
bn_structure <- empty.graph(nodes = colnames(carcass))
bn_fit <- bn.fit(bn_structure, data = carcass)
simulated_data <- rbn(bn_fit, n = 25)
learned_bn <- hc(simulated_data)
graphviz.plot(learned_bn)
```

The comparison results will provide insights into the differences between the learned structure of the simulated dataset and the original model in terms of edge presence, direction, and other structural characteristics.

(e) Simulate a dataset from your BN in part A with 100 samples, then learn the structure, how does the model compare with A

```r
bn_structure <- empty.graph(nodes = colnames(carcass))
bn_fit <- bn.fit(bn_structure, data = carcass)
simulated_data <- rbn(bn_fit, n = 100)
learned_bn <- hc(simulated_data)
graphviz.plot(learned_bn)
```

Problem 4 : Blue baby syndrome (infant methemoglobinemia) occurs when there is not enough exygen in the blood. The aim of the following network that leverages both clinical expertise and historic data. The below DAG represents the incidence and presentation of six possible diseases that would lead to a blue baby syndrome.

(a) Construct the BN structure in R, what is the joint distribution written in compact factored form (modelstring use is acceptable)

Let us create the network as follows :

```
dag4 <- empty.graph(nodes =c("n1","n2","n3","n4","n5","n6","n7","n8","n9","n10","n11","n12","n13","n14"


dag4 <- set.arc(dag4, from = "n1", to = "n2")
dag4 <- set.arc(dag4, from = "n2", to = "n3")
dag4 <- set.arc(dag4, from = "n2", to = "n4")
dag4 <- set.arc(dag4, from = "n2", to = "n5")
dag4 <- set.arc(dag4, from = "n2", to = "n6")
dag4 <- set.arc(dag4, from = "n2", to = "n7")
dag4 <- set.arc(dag4, from = "n2", to = "n8")
dag4 <- set.arc(dag4, from = "n2", to = "n9")
dag4 <- set.arc(dag4, from = "n4", to = "n15")
dag4 <- set.arc(dag4, from = "n5", to = "n10")
dag4 <- set.arc(dag4, from = "n6", to = "n10")
dag4 <- set.arc(dag4, from = "n6", to = "n11")
dag4 <- set.arc(dag4, from = "n7", to = "n11")
dag4 <- set.arc(dag4, from = "n7", to = "n12")
dag4 <- set.arc(dag4, from = "n7", to = "n13")
dag4 <- set.arc(dag4, from = "n7", to = "n14")
dag4 <- set.arc(dag4, from = "n8", to = "n13")
dag4 <- set.arc(dag4, from = "n9", to = "n13")
dag4 <- set.arc(dag4, from = "n9", to = "n14")
dag4 <- set.arc(dag4, from = "n10", to = "n16")
dag4 <- set.arc(dag4, from = "n11", to = "n16")
dag4 <- set.arc(dag4, from = "n11", to = "n17")
dag4 <- set.arc(dag4, from = "n12", to = "n18")
dag4 <- set.arc(dag4, from = "n13", to = "n19")
dag4 <- set.arc(dag4, from = "n14", to = "n20")
```

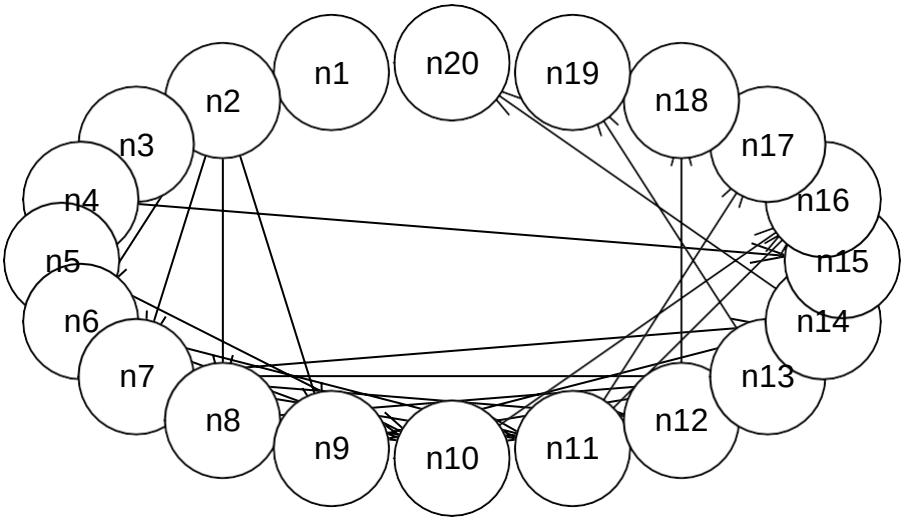
Let us see the description of network :

```
dag4
```

```
##
##   Random/Generated Bayesian network
##
##   model:
##    [n1][n2|n1][n3|n2][n4|n2][n5|n2][n6|n2][n7|n2][n8|n2][n9|n2][n10|n5:n6]
##    [n11|n6:n7][n12|n7][n13|n7:n8:n9][n14|n7:n9][n15|n4][n16|n10:n11][n17|n11]
##    [n18|n12][n19|n13][n20|n14]
##   nodes:                                 20
##   arcs:                                  25
##     undirected arcs:                     0
##     directed arcs:                       25
##   average markov blanket size:           3.10
```

```
##   average neighbourhood size:           2.50
##   average branching factor:             1.25
##
##   generation algorithm:                 Empty
```

```
plot(dag4)
```



The joint disturbution of the network is :

```
modelstring(dag4)
```

```
## [1]
```

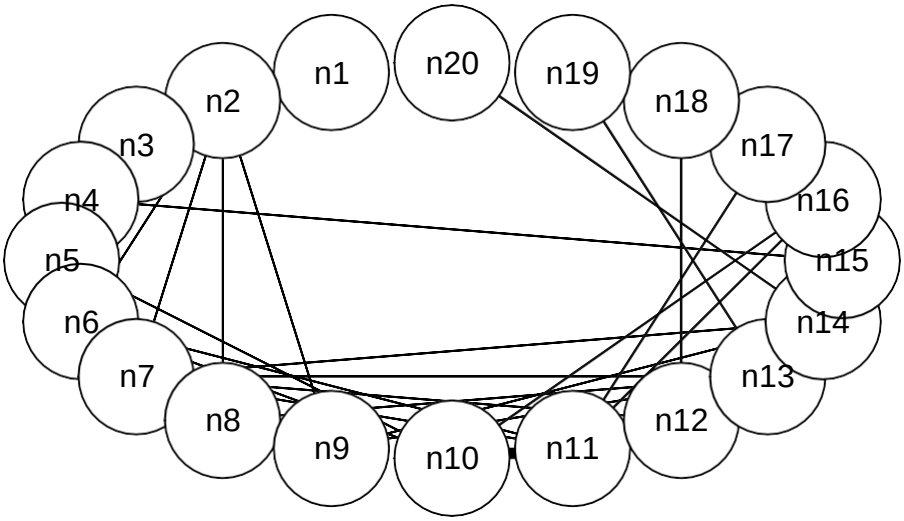
```
"[n1][n2|n1][n3|n2][n4|n2][n5|n2][n6|n2][n7|n2][n8|n2][n9|n2][n10|n5:n6][n11|n6:n7][n12|n7][n13|
```

(b) Draw the moral graph.

```
moral_graph <- moral(dag4)
moral_graph
```

```
##
##   Random/Generated Bayesian network
##
##   model:
##     [undirected graph]
##   nodes:                                20
##   arcs:                                 31
##     undirected arcs:                    31
##     directed arcs:                      0
##   average markov blanket size:          3.10
##   average neighbourhood size:           3.10
##   average branching factor:             0.00
##
##   generation algorithm:                 Empty
```

```
plot(moral_graph)
```



(c) Parametrize using the data posted with the assignment

```
df4<-  read.csv("child_network.csv")
head(df4)
```

```
##   BirthAsphyxia HypDistrib HypoxiaInO2    CO2 ChestXray Grunting LVHreport
## 1            no      Equal    Moderate Normal Oligaemic       no        no
## 2            no      Equal      Severe Normal Asy/Patch      yes        no
## 3            no      Equal      Severe Normal    Normal       no       yes
## 4            no      Equal    Moderate Normal Asy/Patch      yes        no
## 5            no      Equal    Moderate Normal    Normal       no        no
## 6            no      Equal    Moderate Normal Oligaemic       no        no
##   LowerBodyO2 RUQO2 CO2Report XrayReport Disease GruntingReport       Age LVH
## 1       5-12  5-12     >=7.5  Grd_Glass   PAIVS             no  0-3_days  no
## 2        12+    <5      <7.5 Asy/Patchy     TGA             no  0-3_days  no
## 3        12+  5-12      <7.5     Normal     TGA             no  0-3_days yes
## 4       5-12  5-12     >=7.5 Asy/Patchy    Lung            yes  0-3_days  no
## 5       5-12  5-12      <7.5     Normal  Fallot             no 4-10_days  no
## 6       5-12    <5      <7.5  Oligaemic  Fallot            yes 4-10_days  no
##   DuctFlow CardiacMixing LungParench LungFlow Sick
## 1 Lt_to_Rt      Complete      Normal      Low  yes
## 2     None       Transp.    Abnormal      Low  yes
## 3     None       Transp.      Normal     High  yes
## 4     None          Mild    Abnormal   Normal  yes
## 5 Lt_to_Rt      Complete      Normal      Low   no
## 6 Lt_to_Rt          Mild      Normal      Low   no
```

Convert the data into factors :

```
for(i  in colnames(df4)){
  df4[,i] = as.factor(df4[,i])
}
head(df4)
```

```
##  BirthAsphyxia HypDistrib HypoxiaInO2    CO2 ChestXray Grunting LVHreport
## 1            no      Equal    Moderate Normal Oligaemic       no        no
## 2            no      Equal      Severe Normal Asy/Patch      yes        no
## 3            no      Equal      Severe Normal    Normal       no       yes
```

```
## 4            no      Equal    Moderate Normal Asy/Patch      yes        no
## 5            no      Equal    Moderate Normal    Normal       no        no
## 6            no      Equal    Moderate Normal Oligaemic       no        no
##   LowerBodyO2 RUQO2 CO2Report XrayReport Disease GruntingReport       Age LVH
## 1        5-12  5-12     >=7.5  Grd_Glass   PAIVS             no  0-3_days  no
## 2         12+    <5      <7.5 Asy/Patchy     TGA             no  0-3_days  no
## 3         12+  5-12      <7.5     Normal     TGA             no  0-3_days yes
## 4        5-12  5-12     >=7.5 Asy/Patchy    Lung            yes  0-3_days  no
## 5        5-12  5-12      <7.5     Normal  Fallot             no 4-10_days  no
## 6        5-12    <5      <7.5  Oligaemic  Fallot            yes 4-10_days  no
##   DuctFlow CardiacMixing LungParench LungFlow Sick
## 1 Lt_to_Rt      Complete      Normal      Low  yes
## 2     None       Transp.    Abnormal      Low  yes
## 3     None       Transp.      Normal     High  yes
## 4     None          Mild    Abnormal   Normal  yes
## 5 Lt_to_Rt      Complete      Normal      Low   no
## 6 Lt_to_Rt          Mild      Normal      Low   no
```

```
mapping <- c("BirthAsphyxia" = "n1",
             "Disease" = "n2",
             "Age" = "n3",
             "LVH" = "n4",
             "DuctFlow" = "n5",
             "CardiacMixing" = "n6",
             "LungParench" = "n7",
             "LungFlow" = "n8",
             "Sick" = "n9",
             "HypDistrib" = "n10",
             "HypoxiaInO2" = "n11",
             "CO2" = "n12",
             "ChestXray" = "n13",
             "Grunting" = "n14",
             "LVHreport" = "n15",
             "LowerBodyO2" = "n16",
             "RUQO2" = "n17",
             "CO2Report" = "n18",
             "XrayReport" = "n19",
             "GruntingReport" = "n20")
names(df4) <- sapply(names(df4), function(x) mapping[[x]])
head(df4)
```

```
##   n1   n10      n11    n12       n13 n14 n15  n16  n17   n18        n19     n2
## 1 no Equal Moderate Normal Oligaemic  no  no 5-12 5-12 >=7.5  Grd_Glass  PAIVS
## 2 no Equal   Severe Normal Asy/Patch yes  no  12+   <5  <7.5 Asy/Patchy    TGA
## 3 no Equal   Severe Normal    Normal  no yes  12+ 5-12  <7.5     Normal    TGA
## 4 no Equal Moderate Normal Asy/Patch yes  no 5-12 5-12 >=7.5 Asy/Patchy   Lung
## 5 no Equal Moderate Normal    Normal  no  no 5-12 5-12  <7.5     Normal Fallot
## 6 no Equal Moderate Normal Oligaemic  no  no 5-12   <5  <7.5  Oligaemic Fallot
##   n20        n3  n4       n5       n6       n7     n8  n9
## 1  no  0-3_days  no Lt_to_Rt Complete   Normal    Low yes
## 2  no  0-3_days  no     None  Transp. Abnormal    Low yes
## 3  no  0-3_days yes     None  Transp.   Normal   High yes
## 4 yes  0-3_days  no     None     Mild Abnormal Normal yes
## 5  no 4-10_days  no Lt_to_Rt Complete   Normal    Low  no
## 6 yes 4-10_days  no Lt_to_Rt     Mild   Normal    Low  no
```

Parametrize the network using the dataset :

```
bn_fit4 <- bn.fit(dag4,data = df4,method = "mle")
```

(d) What is the CPT for n13 ?

The cpt for n13 is :

```
bn_fit4$n13
```

```
##
##   Parameters of node n13 (multinomial distribution)
##
## Conditional probability table:
##
## , , n8 = High, n9 = no
##
##            n7
## n13             Abnormal   Congested      Normal
##   Asy/Patch 0.048991354 0.120481928 0.010821133
##   Grd_Glass 0.345821326 0.349397590 0.031190325
##   Normal    0.244956772 0.060240964 0.163590070
##   Oligaemic 0.334293948 0.030120482 0.007001910
##   Plethoric 0.025936599 0.439759036 0.787396563
##
## , , n8 = Low, n9 = no
##
##            n7
## n13             Abnormal   Congested      Normal
##   Asy/Patch 0.697594502 0.158163265 0.021129326
##   Grd_Glass 0.051546392 0.500000000 0.021493625
##   Normal    0.046391753 0.061224490 0.159562842
##   Oligaemic 0.146048110 0.224489796 0.778870674
##   Plethoric 0.058419244 0.056122449 0.018943534
##
## , , n8 = Normal, n9 = no
##
##            n7
## n13             Abnormal   Congested      Normal
##   Asy/Patch 0.801498127 0.091666667 0.031914894
##   Grd_Glass 0.044943820 0.675000000 0.010638298
##   Normal    0.056179775 0.058333333 0.903073286
##   Oligaemic 0.059925094 0.025000000 0.021276596
##   Plethoric 0.037453184 0.150000000 0.033096927
##
## , , n8 = High, n9 = yes
##
##            n7
## n13             Abnormal   Congested      Normal
##   Asy/Patch 0.042735043 0.090361446 0.008902077
##   Grd_Glass 0.363247863 0.439759036 0.032640950
##   Normal    0.217948718 0.060240964 0.169139466
##   Oligaemic 0.346153846 0.024096386 0.010385757
##   Plethoric 0.029914530 0.385542169 0.778931751
##
```

```
## , , n8 = Low, n9 = yes
##
##             n7
## n13              Abnormal   Congested      Normal
##   Asy/Patch 0.746938776 0.156862745 0.023133544
##   Grd_Glass 0.040816327 0.490196078 0.017875920
##   Normal    0.061224490 0.009803922 0.146161935
##   Oligaemic 0.106122449 0.264705882 0.793901157
##   Plethoric 0.044897959 0.078431373 0.018927445
##
## , , n8 = Normal, n9 = yes
##
##             n7
## n13              Abnormal   Congested      Normal
##   Asy/Patch 0.807142857 0.081081081 0.022222222
##   Grd_Glass 0.057142857 0.628378378 0.013888889
##   Normal    0.046428571 0.067567568 0.908333333
##   Oligaemic 0.050000000 0.020270270 0.030555556
##   Plethoric 0.039285714 0.202702703 0.025000000
```

(e) What is the CPT for n14 ?

The CPT for n14 is :

```
bn_fit4$n14
```

```
##
##   Parameters of node n14 (multinomial distribution)
##
## Conditional probability table:
##
## , , n9 = no
##
##      n7
## n14      Abnormal  Congested     Normal
##   no  0.38879599 0.78630705 0.94730725
##   yes 0.61120401 0.21369295 0.05269275
##
## , , n9 = yes
##
##      n7
## n14      Abnormal  Congested     Normal
##   no  0.19104084 0.60096154 0.79949622
##   yes 0.80895916 0.39903846 0.20050378
```

(f) Suppose "lower body O2 >5" and "X-ray report = pleothoric", what can we deduce about the disease? Visualize this information on the network.

Based on the given info , The probability of having different diseases is as follows :

```
for(i in c("Fallot" , "Lung" , "PAIVS" , "PFC" , "TAPVD" , "TGA")){
 print( cpquery(bn_fit4 , event = (n2 == i) , evidence = (((n16 == "5-12")|(n16 == "12+")) & (n19 == "P
}
```

```
## [1] 0.1478261
## [1] 0.03198781
```

```
## [1] 0.1094605
## [1] 0.02446982
## [1] 0.08230769
## [1] 0.6454678
```

The child has a likelihood of 0.611 for TGA disease, which is high, and 0.021 for PFC disease, which is low.

(g) Suppose "lower body O2 < 5" and "X-ray report = oligamic" but the child is "not grunting",what can we deduce about the disease? Visualize this information on the network.

```
for(i in c("Fallot" , "Lung" , "PAIVS" , "PFC" , "TAPVD" , "TGA")){
 print( cpquery(bn_fit4 , event = (n2 == i) , evidence = ((n16 == "<5") & (n19 == "Oligaemic") & (n14 =
}
```

```
## [1] 0.452765
## [1] 0.002358491
## [1] 0.3416009
## [1] 0.05350773
## [1] 0.008373206
## [1] 0.09145608
```

The child is more susceptible to PAVIS (0.34 probability) and Fallot (0.434 probability).

(h) Baby Julie is "grunting" with "mild cardiac mixing". Baby George is "not grunting" with"complete cardiac mixing". Which is most at risk for the disease, and why?

For Baby Julie :

```
for(i in c("Fallot" , "Lung" , "PAIVS" , "PFC" , "TAPVD" , "TGA")){
 print( cpquery(bn_fit4 , event = (n2 == i) , evidence = ((n6 == "Mild") & (n14 == "yes"))))
}
```

```
## [1] 0.1873536
## [1] 0.4447236
## [1] 0.01711491
## [1] 0.1458886
## [1] 0.01923077
## [1] 0.1600985
```

For Baby George :

```
for(i in c("Fallot" , "Lung" , "PAIVS" , "PFC" , "TAPVD" , "TGA")){
 print( cpquery(bn_fit4 , event = (n2 == i) , evidence = ((n6 == "Complete") & (n14 == "no"))))
}
```

```
## [1] 0.4499405
## [1] 0.001877494
## [1] 0.4128761
## [1] 0.0106615
## [1] 0.06929316
## [1] 0.0547682
```

According to the aforementioned statistics, Baby George is at a higher risk of contracting the disease than Baby Julie because he has a 45.8 percent chance of contracting Fallot and a 41.1 percent chance of being affected by PAVIS.