

# Wehrmuller-219369109-Ass2

*Gabor Wehrmuller*

*22/09/2019*

## Q1)

### 1.1

$$P(J, W, S, A, P, V, D, U, M) = P(J)P(A)P(W|J)P(P|W)P(U|P, D)P(M|U)P(S|J)P(V|S, A)P(D|V)$$

### 1.2

For each node:  $(n_v - 1) \prod_u n_u$

$$P(J) = 3$$

$$P(A) = 3$$

$$P(W|J) = 1 * 4 = 4$$

$$P(P|W) = 1 * 2 = 2$$

$$P(U|P, D) = 2 * 8 = 16$$

$$P(M|U) = 3 * 3 = 9$$

$$P(V|S, A) = 1 * 16 = 16$$

$$P(D|V) = 3 * 2 = 6$$

Total: 59

### 1.3

Without the assumption:

$$(2^3)(3^1)(4^5) - 1 = 24575$$

Thanks to the assumptions about the independence among the variables, the numbers of necessary parameters can drastically be reduced. This makes inference less complex and more intuitive.

### 1.4

#### 1.4 a)

$$W \perp \emptyset$$

possible paths:

$$\{W, J, S, V\}, \{W, P, U, D, V\}$$

The first path is unblocked, the second is by U. Therefore they are not d-separated and W is not marginally independent of V. The statement is **false**.

#### 1.4 b)

$A \perp M | \{D, W\}$

possible paths:

$\{A, V, S, J, W, P, U, M\}, \{A, V, D, U, M\}$

The first path is blocked, since W is observed and blocks the path. The second path is blocked, since D is observed and blocks the path. The statements is **true**.

#### 1.4 c)

$\{A, W\} \perp D | V$

possible paths:

$A : \{A, V, D\}, \{A, V, S, J, W, P, U, D\}$

$W : \{W, P, U, D\}, \{W, J, S, V, D\}$

The first path for A is blocked, since V is observed and blocks the path. The second path for A is blocked by U, since there the paths are head-to-head. The head-to-head at V is unblocked, since V is observed.

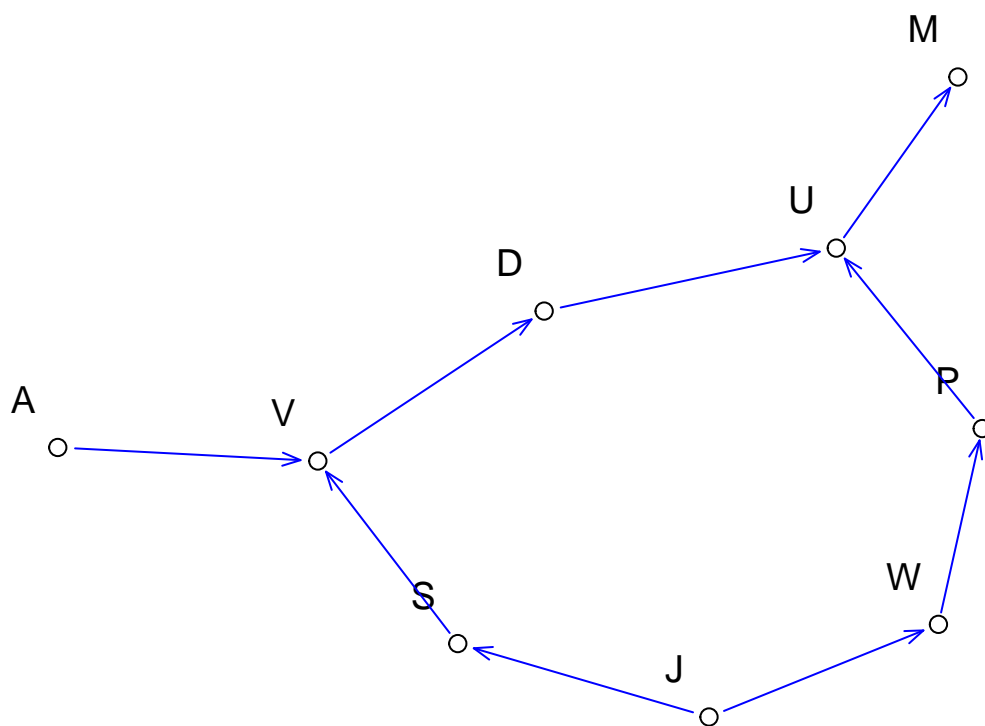
The first path for W is blocked by U since there the paths are head-to-head. The second path is blocked, since V is observed and changes the path from unblocked to blocked.

The statement is **true**.

#### 1.5

```
dag <- DAG(W~J, P~W, U~P+D,M~U, S~J, V~S+A, D~V)
```

```
drawGraph(dag, adjust = F)
```



```

# a)
dSep(dag,first = "W", second = "V", cond = c())

```

```
## [1] FALSE
```

```

# b)
dSep(dag,first = "A", second = "M", cond = c("D","W"))

```

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

```

# c)
dSep(dag,first = c("A","W"), second = "D", cond = c("V"))

```

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

## 1.6

$P(M|S = 3000 - 6000, V = Yes, P = Peakhour, D = < 1)$

Order : J, W, A, U

$P(M|S = 3000 - 6000, V = Yes, P = Peakhour, D = < 1) = \frac{P(M, S=3000-6000, V=Yes, P=Peakhour, D=<1)}{P(S=3000-6000, V=Yes, P=Peakhour, D=<1)}$

$P(M, S, V, P, D) = \sum_{J, W, A, U} P(J)P(A)P(W|J)P(P|W)P(U|P, D)P(M|U)P(S|J)P(V|S, A)P(D|V)$   
 $= \sum_{J, W, A, U} f_0(A)f_1(J)f_2(W, J)f_3(P, W)f_4(V, P, D)f_5(M, U)f_6(S, J)f_7(V, S, A)f_8(D, V)$

Observe  $S = 3000 - 6000, V = Yes, P = Peakhour, D = < 1$

$= \sum_{J, W, A, U} f_0(A)f_1(J)f_2(W, J)f_9(W)f_5(M, U)f_{10}(J)f_{11}(A)$

Eliminate J

$= \sum_{W, A, U} f_0(A)f_9(W)f_5(M, U)f_{11}(A) \sum_J f_1(J)f_2(W, J)f_{10}(J)$

$= \sum_{W, A, U} f_0(A)f_9(W)f_5(M, U)f_{11}(A)f_2(W)$

Eliminate W

$= \sum_{A, U} f_0(A)f_5(M, U)f_{11}(A) \sum_W f_9(W)f_2(W)$

$= \sum_{A, U} f_0(A)f_5(M, U)f_{11}(A)$

Eliminate A

$= \sum_U f_5(M, U) \sum_A f_0(A)f_{11}(A)$

$= \sum_U f_5(M, U)$

Eliminate U

$= f_5(M)$

The posterior distribution :

$P(M|S = 3000 - 6000, V = Yes, P = Peakhour, D = < 1) = \frac{f_5(M)}{\sum_M f_5(M)}$

## Q2)

### 2.1 a)

```
# biocLite is not available for r 3.6 - therefore usage of BiocManager to install the packages
library(RBGL)
library(gRbase)
library(gRain)
library(Rgraphviz)

lh <- c("low", "high")
lmmh <- c("low", "lower middle", "upper middle", "high")

eh <- cptable(~eh, values=c(3, 7), levels=lh)

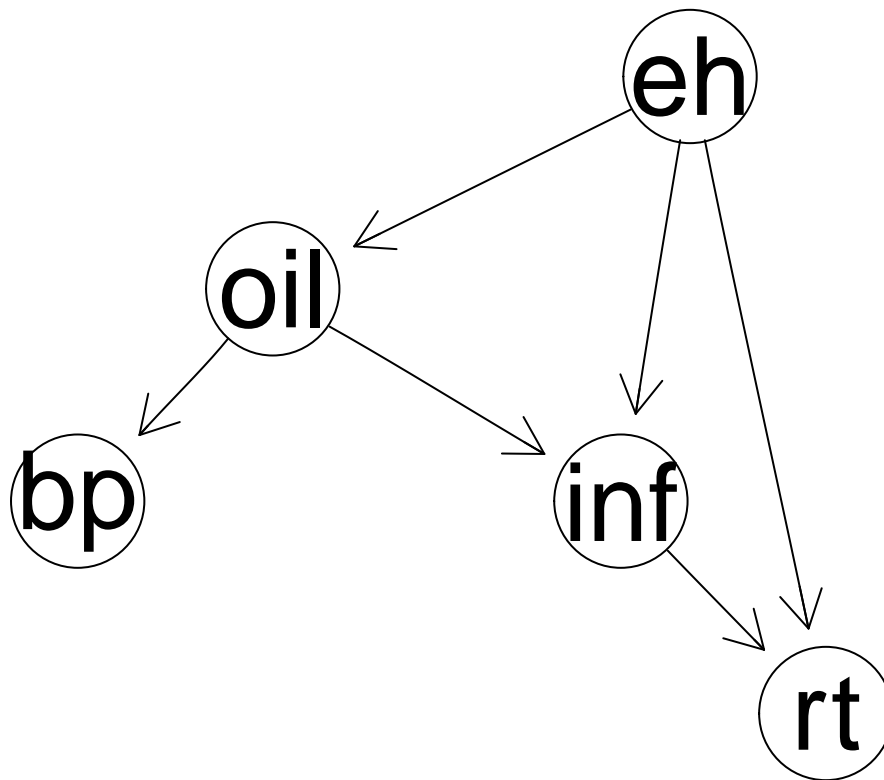
oil.eh <- cptable(~oil|eh, values = c(3,7,6,4), levels = lh)

bp.oil <- cptable(~bp|oil, values = c(20,45,30,5, 20,50,20,10), levels = lmmh)

inf.oil.eh <- cptable(~inf|oil:eh, values = c(1,9, 6,4, 5,5, 4,6), levels = lh)

rt.inf.eh <- cptable(~rt|inf:eh, values = c(2,8, 6,4, 7,3, 2,8), levels = lh)

plist <- compileCPT(list(eh, oil.eh, bp.oil, inf.oil.eh, rt.inf.eh))
net <- grain(plist)
plot(net)
```



2.1 b)

```

for (i in plist){
  print("-----")
  print(i)
}

```

```

## [1] "-----"
## eh
## low high
## 0.3 0.7
## attr("class")
## [1] "parray" "array"
## [1] "-----"
##      eh
## oil   low high
## low  0.3 0.6
## high 0.7 0.4
## attr("class")
## [1] "parray" "array"
## [1] "-----"
##      oil
## bp      low high
## low      0.20 0.2
## lower middle 0.45 0.5
## upper middle 0.30 0.2

```

```
##      high      0.05  0.1
## attr("class")
## [1] "parray" "array"
## [1] "-----"
## , , eh = low
##
##      oil
## inf      low high
## low  0.1  0.6
## high 0.9  0.4
##
## , , eh = high
##
##      oil
## inf      low high
## low  0.5  0.4
## high 0.5  0.6
##
## attr("class")
## [1] "parray" "array"
## [1] "-----"
## , , eh = low
##
##      inf
## rt      low high
## low  0.2  0.6
## high 0.8  0.4
##
## , , eh = high
##
##      inf
## rt      low high
## low  0.7  0.2
## high 0.3  0.8
##
## attr("class")
## [1] "parray" "array"
```

They are the same as in the table.

## 2.2 a)

```
net_2 <- setEvidence(net,nodes = c("oil","ret"),states = c("low","low"))
querygrain(net_2,nodes = "bp")
```

```
## $bp
## bp
##      low lower middle upper middle      high
##      0.20      0.45      0.30      0.05
```

The most likely state for the British petroleum stock price is “middle” with a probability of 45%.

## 2.2 b)

```
net_3 <- setEvidence(net,nodes = "inf",states = "low")
querygrain(net_3,nodes = "rt")
```

```
## $rt
## rt
##      low      high
## 0.5522976 0.4477024
```

the probability that the retailer stock pris is high is 44.77%

## 2.2 c)

```
querygrain(net, nodes = "oil", type = "marginal")
```

```
## $oil
## oil
##  low high
## 0.51 0.49
```

## 2.2 d)

```
querygrain(net, nodes = c("inf", "bp"), type = "joint")
```

```
##      bp
## inf      low lower middle upper middle      high
##  low 0.0914      0.21755      0.1133 0.03475
##  high 0.1086      0.25695      0.1377 0.03975
## attr(,"class")
## [1] "parray" "array"
```

## Q3

### 3.1)

$$\begin{aligned} P(A, B) &= \sum_{C, E, D} P(A, B, C, D, E) \\ &= \sum_{C, E, D} P(A)P(B)P(C|A, B)P(E|C)P(D|C) \\ &= \sum_{E, D} P(A)P(B) \sum_C P(C|A, B)P(E|C)P(D|C) \\ &= \sum_{E, D} P(A)P(B)P(E)P(D) \\ &= P(A)P(B) \sum_E P(E) \sum_D P(D) \\ &= P(A)P(B) \end{aligned}$$

A and B are d-separated by C, since the path is blocked by head-to-head edges.

### 3.2)

$$\begin{aligned} \frac{P(E=1, A=1, B=0)}{P(A=1, B=0)} &= \frac{\sum_{C,D} P(E=1, A=1, B=0, C, D)}{P(A=1, B=0)} \\ \sum_{C,D} P(E=1, A=1, B=0, C, D) &= P(A=1), P(B=0)P(C|A=1, B=0)P(E=1|C)P(D|C) \\ &\text{summarize through } C \text{ and } D : \\ &P(A=1)P(B=0)[ \\ &P(C=0|A=1, B=0)P(E=1|C=0)P(D=0|C=0) + \\ &P(C=0|A=1, B=0)P(E=1|C=0)P(D=1|C=0) + \\ &P(C=1|A=1, B=0)P(E=1|C=1)P(D=0|C=1) + \\ &P(C=1|A=1, B=0)P(E=1|C=1)P(D=1|C=1)] \\ &= (1-\alpha)(\theta)[ \\ &(0.2)(1-\beta)(\gamma) + \\ &(0.2)(1-\beta)(1-\gamma) + \\ &(0.8)(0.7)(0.6) + \\ &(0.8)(0.7)(0.4)] \\ &= (1-\alpha)(\theta)(0.76 - 0.2\beta) \end{aligned}$$

In 3.1 it is shown that:

$$\begin{aligned} P(A, B) &= P(A)P(B), \text{ therefore :} \\ P(A=1, B=0) &= P(A=1)P(B=0) \\ &= (1-\alpha)(\theta) \end{aligned}$$

Hence,

$$P(E=1|A=0, B=0) = \frac{(1-\alpha)(\theta)(0.76-0.2\beta)}{(1-\alpha)(\theta)} = \underline{0.76 - 0.2\beta}$$

### 3.3)

$$\begin{aligned} \alpha &= P(A=0) = \frac{\#(A=0)}{N} = \frac{3}{20} = 0.15 \\ \beta &= P(E=0|C=0) = \frac{\#(C=0, E=0)}{\#(C=0)} = \frac{5}{12} = 0.416 \\ \gamma &= P(D=0|C=0) = \frac{\#(D=0, C=0)}{\#(C=0)} = \frac{2}{12} = 0.16 \\ \theta &= P(B=0) = \frac{\#(B=0)}{N} = \frac{4}{20} = 0.2 \end{aligned}$$

### 3.4)

$$\begin{aligned} P(E=1|A=1, B=0) &= 0.76 - 0.2\beta \\ &= 0.76 - 0.2 * 0.416 = 0.6768 \end{aligned}$$

## Q4

```
ChildData <- read.csv(file="CHILD10k.csv", header=TRUE, sep=",")
```

```
library(bnlearn)
```

```
## Warning: package 'bnlearn' was built under R version 3.6.1
```

```
##
```

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

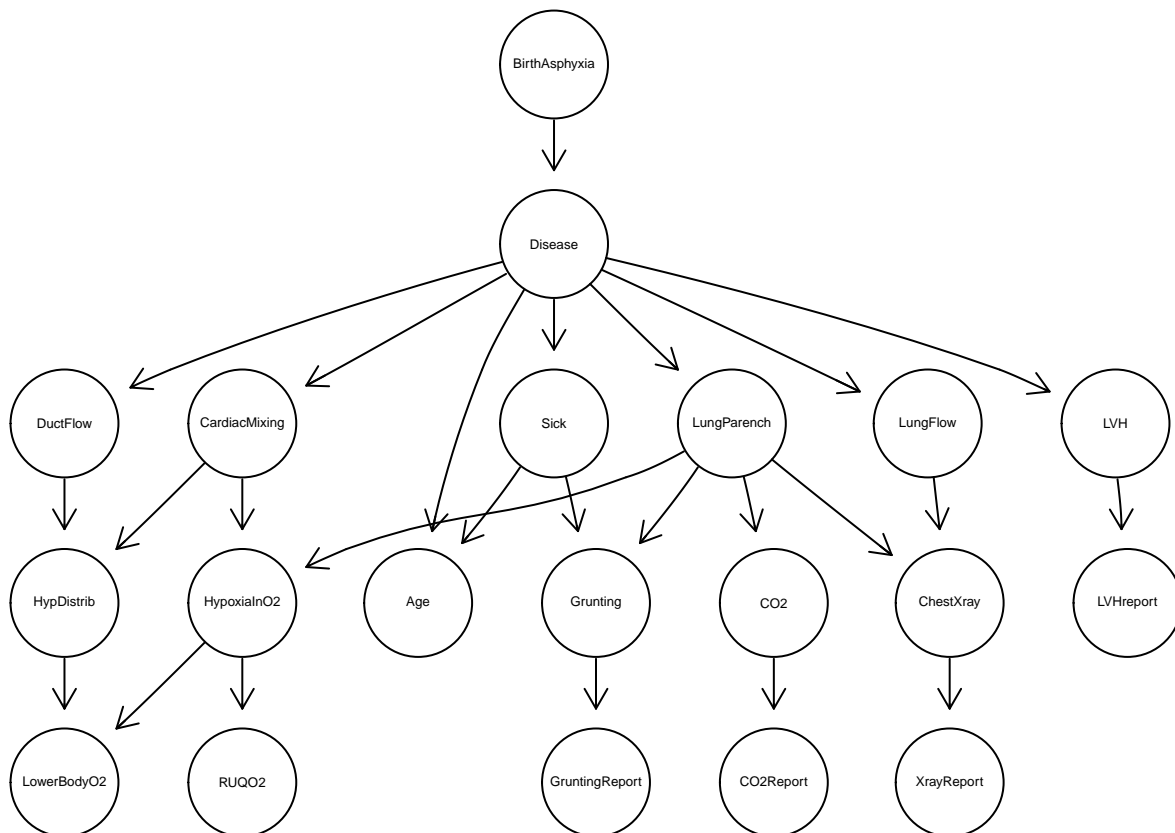
```
## The following objects are masked from 'package:gRbase':
```

```
##
```

```
## ancestors, children, parents
```



```
## The following objects are masked from 'package:BiocGenerics':
##
##   path, score
## The following objects are masked from 'package:igraph':
##
##   compare, path, subgraph
## The following object is masked from 'package:stats':
##
##   sigma
#create and plot the network structure.
modelstring = paste0("[BirthAsphyxia|Disease|BirthAsphyxia] [LVH|Disease] [DuctFlow|Disease]",
  "[CardiacMixing|Disease] [LungParench|Disease] [LungFlow|Disease] [Sick|Disease]",
  "[HypDistrib|DuctFlow:CardiacMixing] [HypoxiaInO2|CardiacMixing:LungParench]",
  "[CO2|LungParench] [ChestXray|LungParench:LungFlow] [Grunting|LungParench:Sick]",
  "[LVHreport|LVH] [Age|Disease:Sick] [LowerBodyO2|HypDistrib:HypoxiaInO2]",
  "[RUQO2|HypoxiaInO2] [CO2Report|CO2] [XrayReport|ChestXray] [GruntingReport|Grunting]")
dag = model2network(modelstring)
par(mfrow = c(1,1))
#source("https://bioconductor.org/biocLite.R")
#biocLite("Rgraphviz")
graphviz.plot(dag)
```

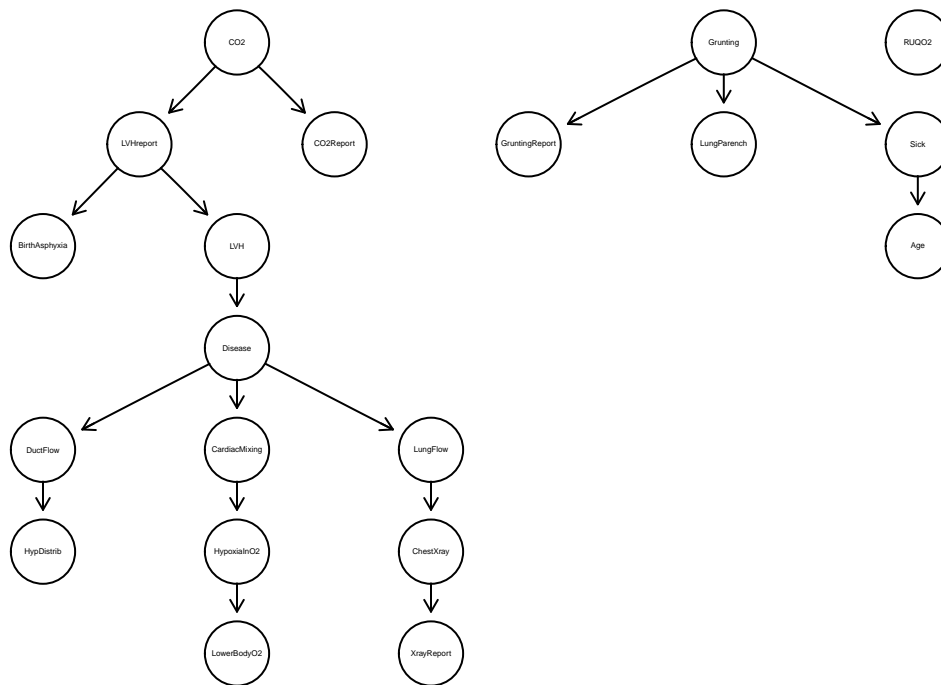


4.1)

```
for (i in c(100,500,1000,5000)){
  bnet_bic = hc(ChildData[1:i,], score = "bic")
  bnet_bde = hc(ChildData[1:i,], score = "bde")
  score_bic = bnlearn::score(bnet_bic, ChildData[1:i,], type = "bic")
  score_bde = bnlearn::score(bnet_bde, ChildData[1:i,], type = "bde")

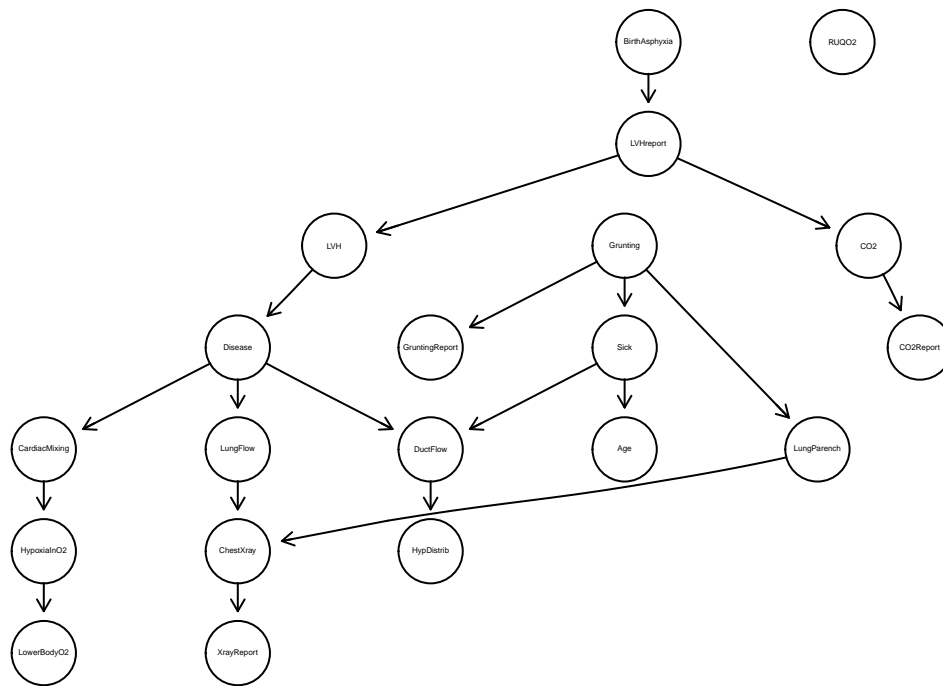
  graphviz.plot(bnet_bic,
    main = paste("\nBayesian Network (score = BIC) with the first", i, "Observations"),
    sub = paste("BIC-Score:",score_bic))
  graphviz.plot(bnet_bde, main = paste("\nBayesian Network (score = BDE) with the first", i, "Observations"),
    sub = paste("BDE-Score:",score_bde))
}
```

Bayesian Network (score = BIC) with the first 100 Observations



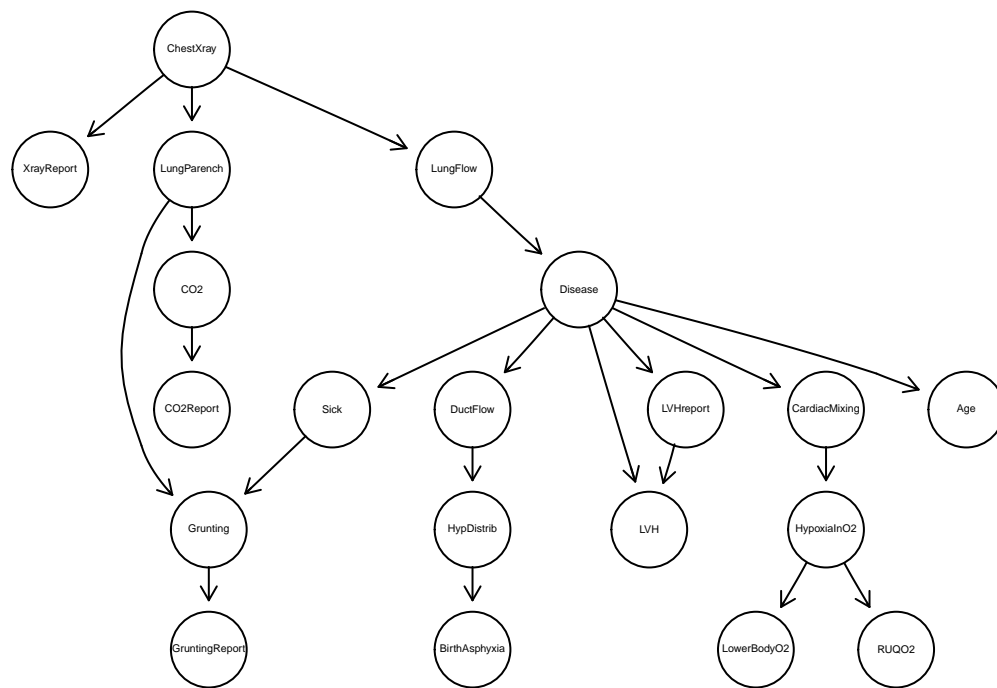
BIC-Score: -1530.05894923122

# Bayesian Network (score = BDE) with the first 100 Observations



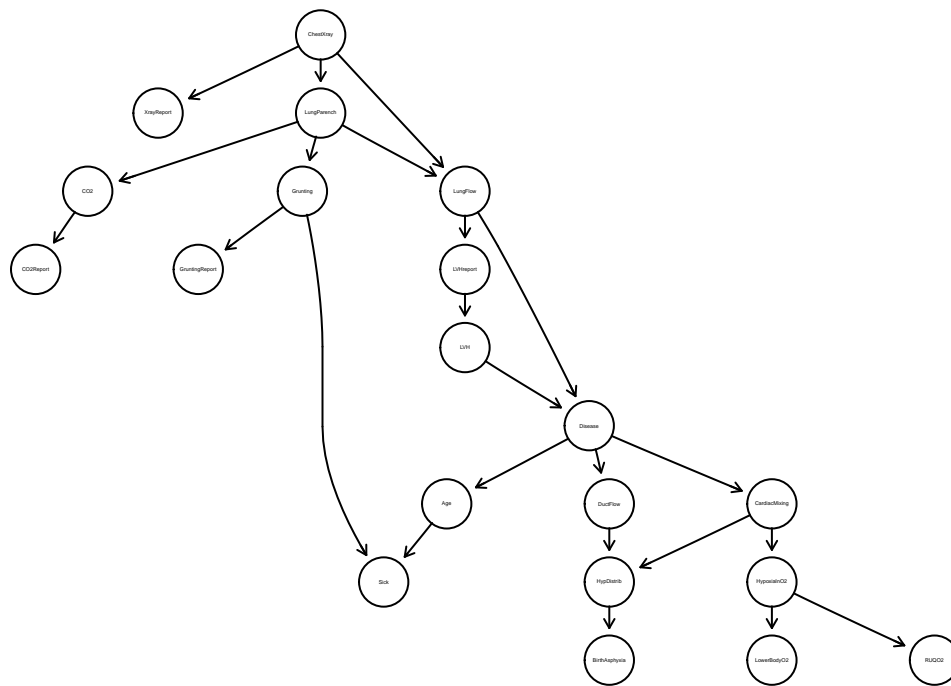
BDE-Score: -1485.44685246671

## Bayesian Network (score = BIC) with the first 500 Observations



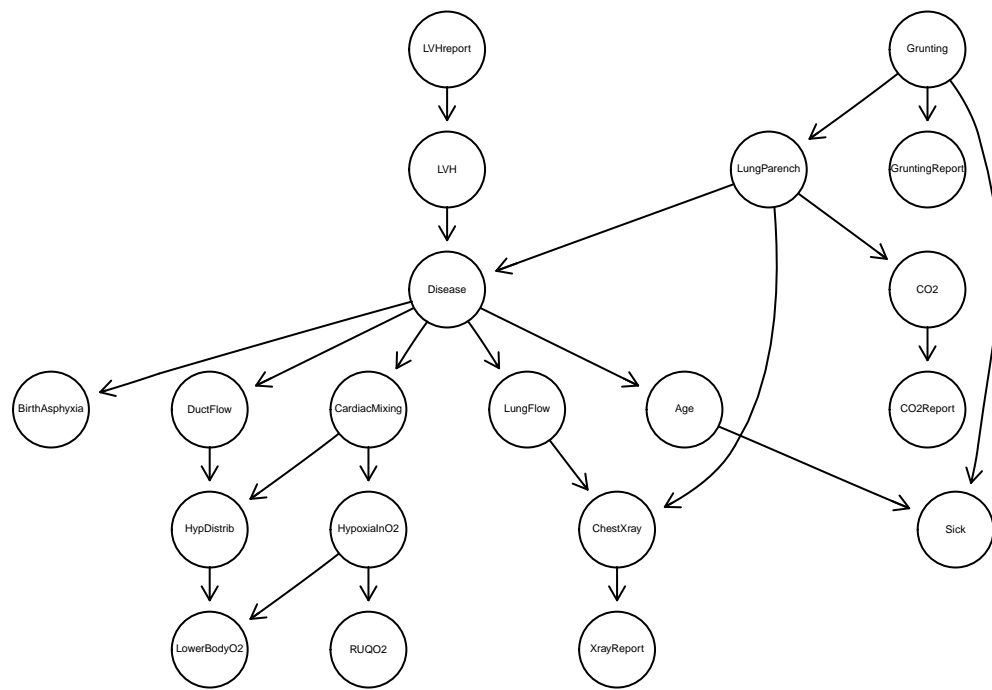
BIC-Score: -6682.96503738455

# Bayesian Network (score = BDE) with the first 500 Observations



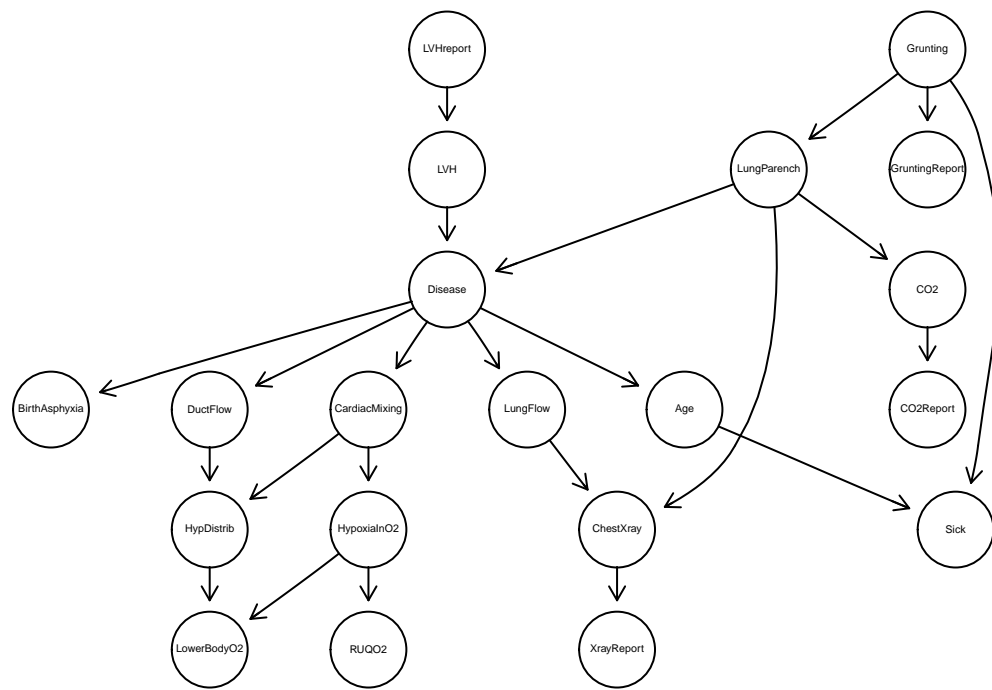
BDE-Score: -6698.14148777467

## Bayesian Network (score = BIC) with the first 1000 Observations



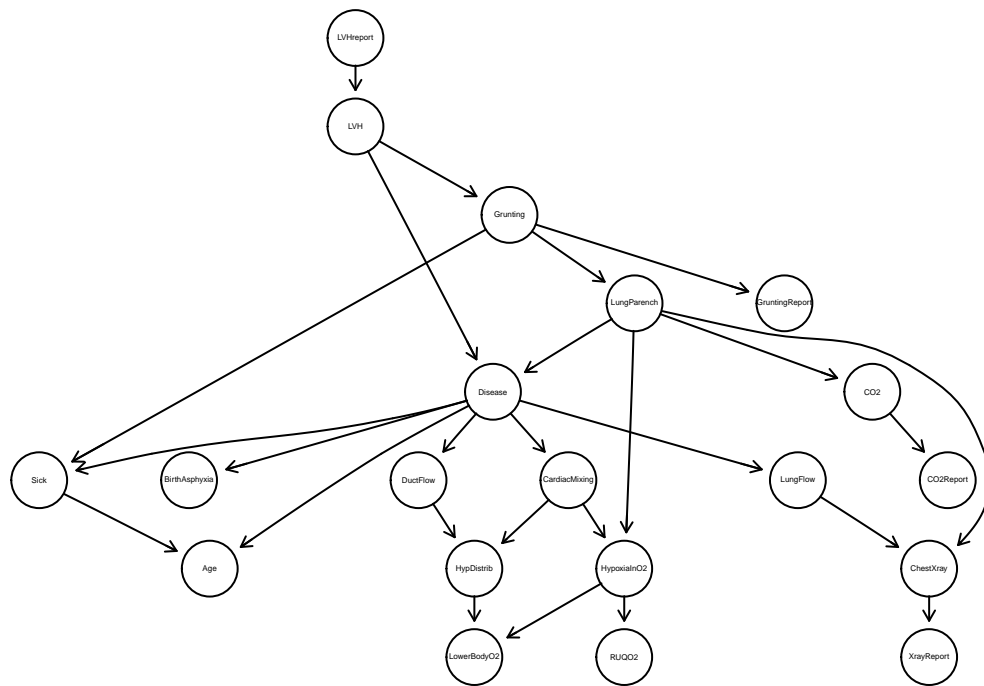
BIC-Score: -13010.8076261982

## Bayesian Network (score = BDE) with the first 1000 Observations



BDE-Score: -13043.7394836404

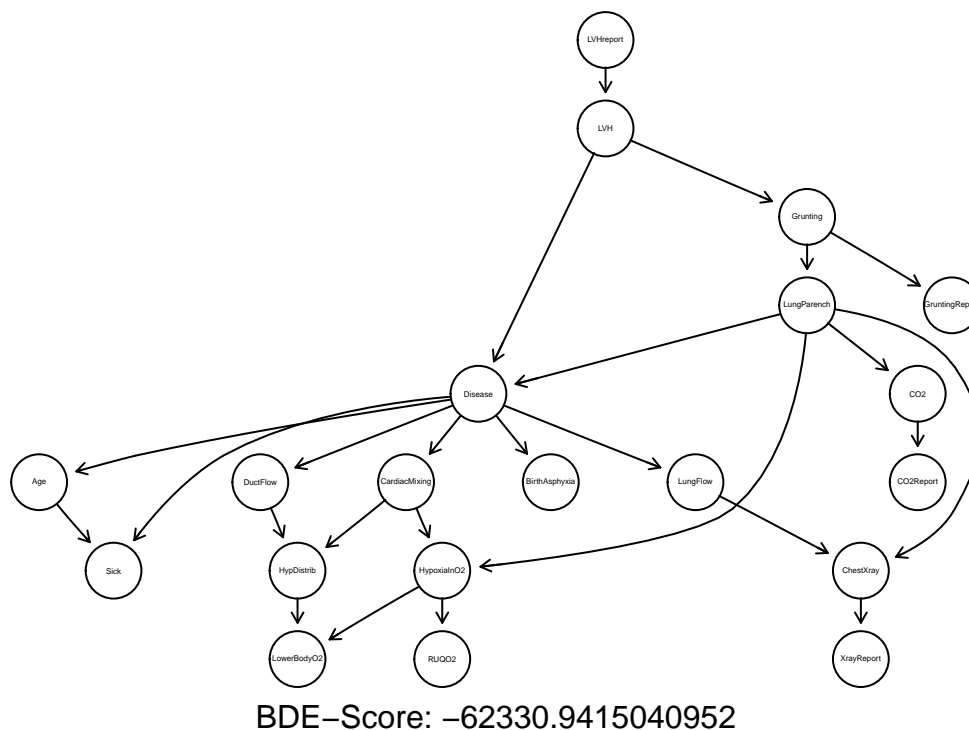
## Bayesian Network (score = BIC) with the first 5000 Observations



BIC-Score: -62210.9938502949



## Bayesian Network (score = BDE) with the first 5000 Observations



### 4.2)

The BIC score increases linearly with sample size. With more samples, the incentive to fit the exact structure of the data increases. This means that with increasing data size, the structure gets more complex, which lowers the score again. This can be observed as well in the plots.

The BDE scores increases as well with sample size. With low sample size, the score tends to favor simpler structures. As more data can be used, more complex structures are also taken into consideration. This can be seen that again with more data, the structure gets more complex.

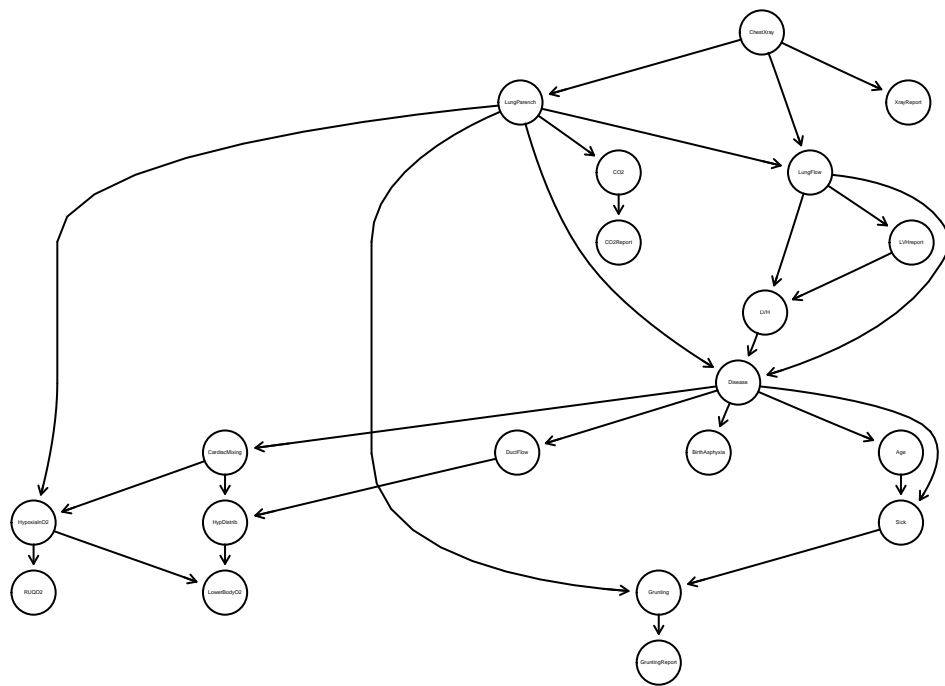
### 4.3 a)

```
bnet.bic.full <- hc(ChildData, score = "bic")
bnet.bde.full <- hc(ChildData, score = "bde")

score.bic.full = bnlearn::score(bnet.bic.full, ChildData, type = "bic")
score.bde.full = bnlearn::score(bnet.bde.full, ChildData, type = "bde")

graphviz.plot(bnet.bic.full, main = paste("\nBayesian Network (score = BIC)"), sub = paste("BIC-Score:"
```

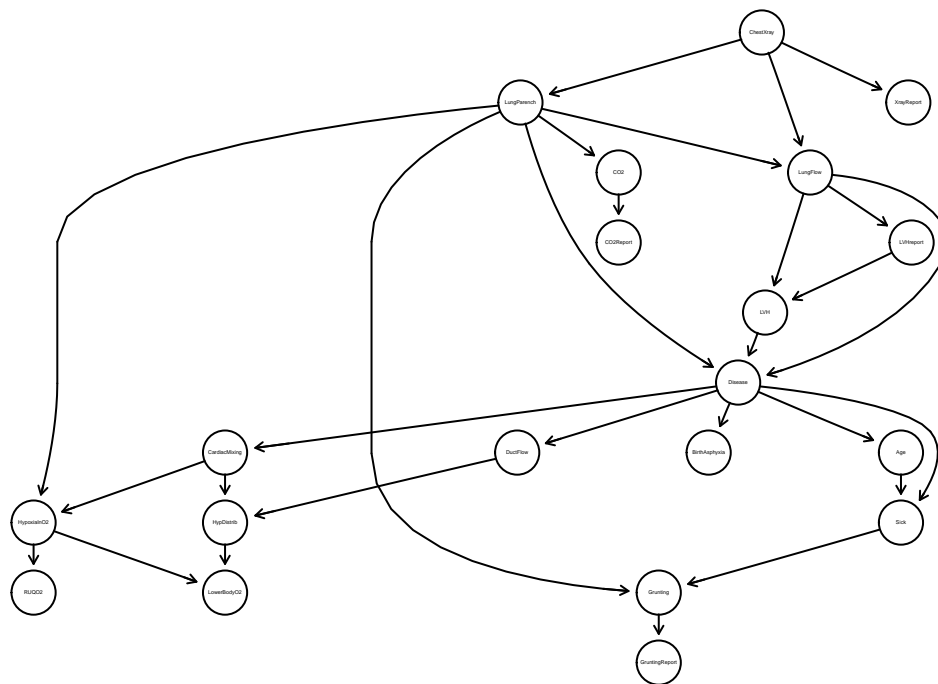
## Bayesian Network (score = BIC)



BIC-Score: -123627.270817048

```
graphviz.plot(bnet.bde.full, main = paste("\nBayesian Network (score = BDE)", sub = paste("BDE-Score:"
```

## Bayesian Network (score = BDE)



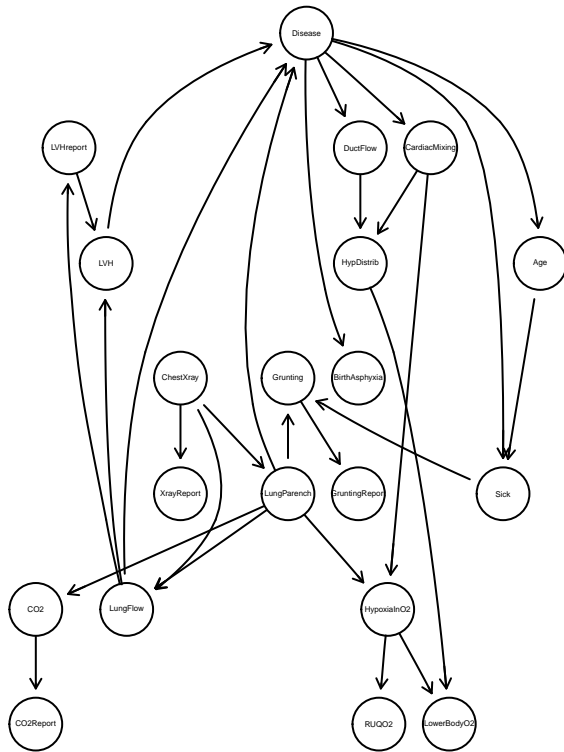
BDE-Score: -123710.318117267

4.3 b)

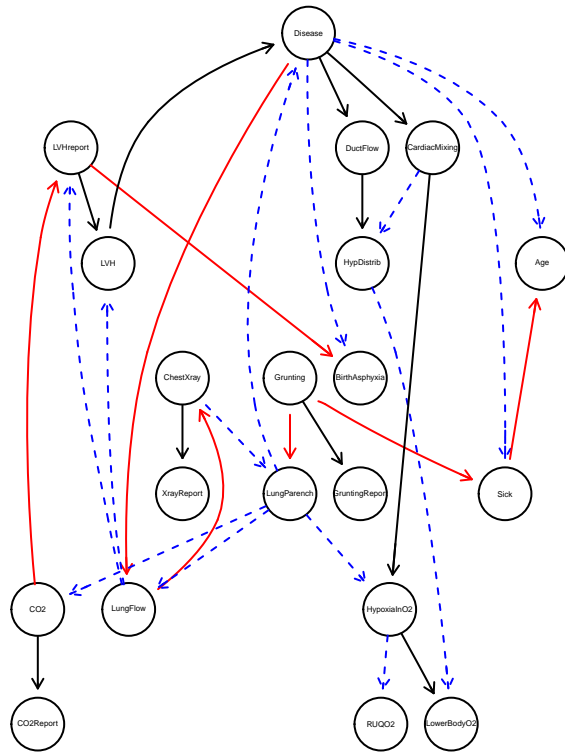
```
par(mfrow = c(1,2))
for (i in c(100,500,1000,5000)){
  assign(paste0("bnet.bic",i), hc(ChildData[1:i,], score = "bic"))
  assign(paste0("bnet.bde",i),hc(ChildData[1:i,], score = "bde"))
}

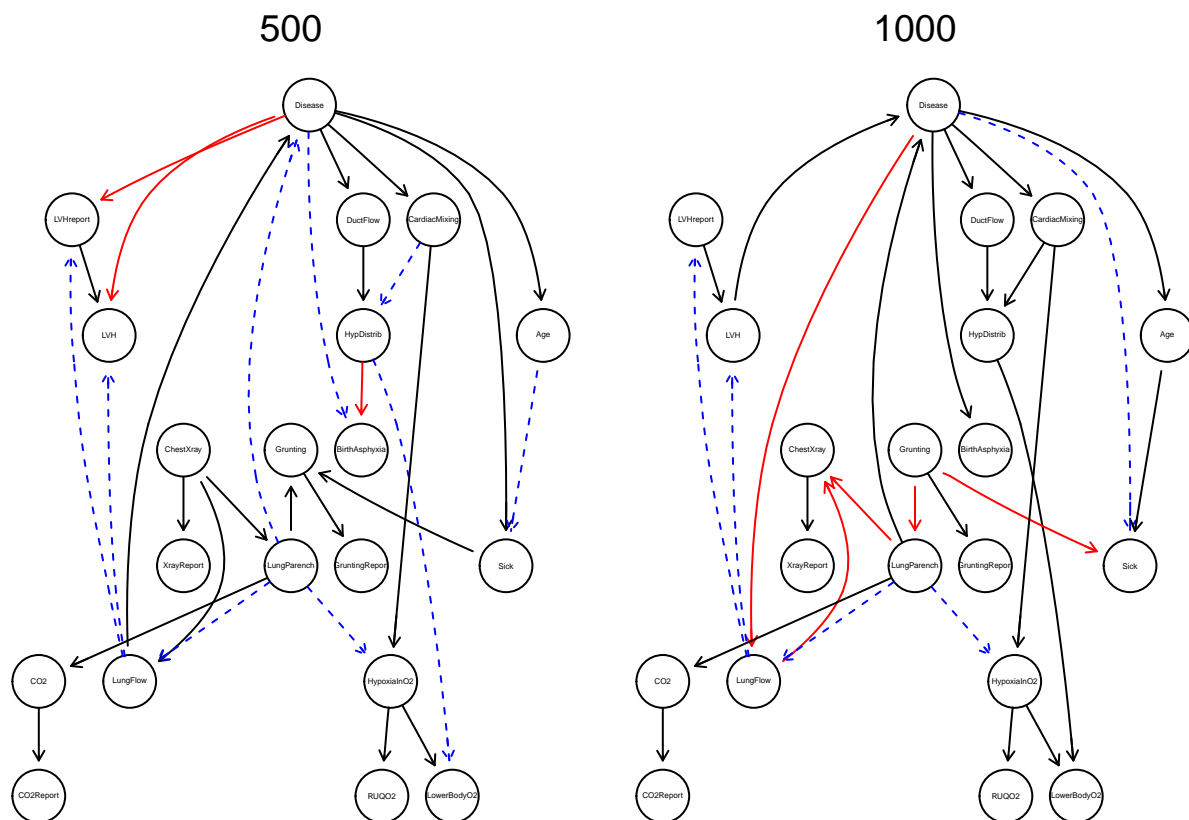
# compare(bnet.bic.full,bnet.bic)
graphviz.compare(bnet.bic.full,bnet.bic100,bnet.bic500,bnet.bic1000,bnet.bic5000,
  main = c("Full (BIC)", "100", "500","1000","5000"))
```

Full (BIC)



100





```
# compare(bnet.bde.full,bnet.bde)
graphviz.compare(bnet.bde.full,bnet.bde100,bnet.bde500,bnet.bde1000,bnet.bde5000,
  main = c("Full (BNE)", "100", "500", "1000", "5000"))
```

The graph illustrates a complex network of causal relationships between medical concepts. The nodes are arranged in a hierarchical manner, with 'Disease' at the top. The edges are color-coded: black for general causal links, red for specific causal links, and blue dashed for potential or inferred links. The graph shows how various reports and clinical findings (like 'LVHreport', 'ChestXray', 'Grunting') lead to specific medical conditions ('LVH', 'Grunting', 'LungParench') and ultimately to physiological states ('CO2', 'LungFlow', 'HypoxatrCO2', 'RUQO2', 'LowerBodyO2').

```

graph TD
    Disease((Disease)) -- black --> LVHreport((LVHreport))
    Disease -- black --> LVH((LVH))
    Disease -- black --> DuctFlow((DuctFlow))
    Disease -- black --> CardiacMixing((CardiacMixing))
    Disease -- black --> Age((Age))
    Disease -- black --> Sick((Sick))
    Disease -- black --> Grunting((Grunting))
    Disease -- black --> BirthAsphyxia((BirthAsphyxia))
    Disease -- black --> LungParench((LungParench))
    Disease -- black --> HypoxatrCO2((HypoxatrCO2))
    Disease -- black --> LowerBodyO2((LowerBodyO2))
    
    LVHreport -- black --> LVH
    LVH -- red --> Disease
    LVH -- red --> Grunting
    LVH -- red --> LungParench
    LVH -- red --> Sick
    
    ChestXray((ChestXray)) -- black --> Grunting
    ChestXray -- black --> XrayReport((XrayReport))
    Grunting -- red --> Disease
    Grunting -- red --> LungParench
    Grunting -- red --> Sick
    
    BirthAsphyxia -- black --> Grunting
    BirthAsphyxia -- black --> Sick
    
    XrayReport -- black --> LungParench
    LungParench -- red --> Disease
    LungParench -- red --> Sick
    
    GruntingReport((GruntingReport)) -- black --> LungParench
    GruntingReport -- black --> HypoxatrCO2
    
    LungParench -- black --> CO2((CO2))
    LungParench -- black --> LungFlow((LungFlow))
    LungParench -- black --> HypoxatrCO2
    LungParench -- black --> LowerBodyO2
    
    CO2 -- black --> CO2Report((CO2Report))
    HypoxatrCO2 -- black --> RUQO2((RUQO2))
    HypoxatrCO2 -- black --> LowerBodyO2
    
    LVH -- blue dashed --> LVHreport
    LVH -- blue dashed --> LungFlow
    LungFlow -- blue dashed --> CO2
    LungFlow -- blue dashed --> LungParench
  
```

[illegible]

The diagram is a causal graph with the following nodes and edges:

- GlobalWarming** (red oval) has a red solid arrow to **GreenhouseGases** and a red solid arrow to **Lungs**.
- GreenhouseGases** (black oval) has a black solid arrow to **CO2**.
- CO2** (black oval) has a black solid arrow to **CO2Rise**.
- CO2Rise** (black oval) has a red solid arrow to **LungFlow**.
- Lungs** (black oval) has a black solid arrow to **LungHealth** and a blue dashed arrow to **LungFlow**.
- LungHealth** (black oval) has a red solid arrow to **ChestInjury** and a blue dashed arrow to **LungFlow**.
- ChestInjury** (black oval) has a black solid arrow to **Angioplasty** and a blue dashed arrow to **Disease**.
- Angioplasty** (black oval) has a blue dashed arrow to **Disease**.
- Disease** (black oval) has a black solid arrow to **Age** and a blue dashed arrow to **Sick**.
- Age** (black oval) has a red solid arrow to **Sick** and a blue dashed arrow to **Suffer**.
- Sick** (black oval) has a red solid arrow to **Suffer** and a blue dashed arrow to **HighChole**.
- Suffer** (black oval) has a black solid arrow to **HighChole**.
- HighChole** (black oval) has a blue dashed arrow to **LowDensityCO2**.
- LowDensityCO2** (black oval) has a blue dashed arrow to **LiverDisease**.
- LiverDisease** (black oval) has a black solid arrow to **LiverDiseaseCO2**.
- LiverDiseaseCO2** (black oval) has a blue dashed arrow to **RUOCD** and a black solid arrow to **LiverDisease**.
- RUOCD** (black oval) has a blue dashed arrow to **LiverDisease**.
- MultipleMyeloma** (red oval) has a red solid arrow to **Lungs** and a blue dashed arrow to **Disease**.

[illegible]





```

## Parameters of node HypDistrib (multinomial distribution)
##
## Conditional probability table:
##
## , , CardiacMixing = Complete
##
##           DuctFlow
## HypDistrib  Lt_to_Rt      None  Rt_to_Lt
##   Equal    0.95405856 0.94011299 0.94809689
##   Unequal  0.04594144 0.05988701 0.05190311
##
## , , CardiacMixing = Mild
##
##           DuctFlow
## HypDistrib  Lt_to_Rt      None  Rt_to_Lt
##   Equal    0.96487985 0.94701987 0.51190476
##   Unequal  0.03512015 0.05298013 0.48809524
##
## , , CardiacMixing = None
##
##           DuctFlow
## HypDistrib  Lt_to_Rt      None  Rt_to_Lt
##   Equal    0.94482759 0.94444444 0.04032258
##   Unequal  0.05517241 0.05555556 0.95967742
##
## , , CardiacMixing = Transp.
##
##           DuctFlow
## HypDistrib  Lt_to_Rt      None  Rt_to_Lt
##   Equal    0.94301994 0.94962335 0.54063604
##   Unequal  0.05698006 0.05037665 0.45936396
##
##
## Parameters of node HypoxiaInO2 (multinomial distribution)
##
## Conditional probability table:
##
## , , LungParench = Abnormal
##
##           CardiacMixing
## HypoxiaInO2  Complete      Mild      None      Transp.
##   Mild      0.10163934 0.09511568 0.70000000 0.01160093
##   Moderate  0.50163934 0.62724936 0.18636364 0.15081206
##   Severe    0.39672131 0.27763496 0.11363636 0.83758701
##
## , , LungParench = Congested
##
##           CardiacMixing
## HypoxiaInO2  Complete      Mild      None      Transp.
##   Mild      0.03703704 0.11034483 0.17721519 0.09701493
##   Moderate  0.63703704 0.73103448 0.77215190 0.32089552
##   Severe    0.32592593 0.15862069 0.05063291 0.58208955
##
## , , LungParench = Normal

```

```

##
##          CardiacMixing
## HypoxiaInO2  Complete      Mild      None      Transp.
## Mild      0.09464286 0.09547739 0.89075630 0.02371181
## Moderate 0.69872449 0.80276382 0.08823529 0.16689466
## Severe   0.20663265 0.10175879 0.02100840 0.80939352
##
##
## Parameters of node CO2 (multinomial distribution)
##
## Conditional probability table:
##
##          LungParench
## CO2      Abnormal  Congested      Normal
## High    0.49462916 0.29287305 0.10116133
## Low     0.05217391 0.05345212 0.10018189
## Normal  0.45319693 0.65367483 0.79865678
##
## Parameters of node ChestXray (multinomial distribution)
##
## Conditional probability table:
## Asy/Patch Grd_Glass      Normal Oligaemic Plethoric
## 0.1299    0.0887    0.2295    0.3364    0.2155
##
## Parameters of node Grunting (multinomial distribution)
##
## Conditional probability table:
##
## , , Sick = no
##
##          LungParench
## Grunting  Abnormal  Congested      Normal
## no       0.38879599 0.78630705 0.94730725
## yes      0.61120401 0.21369295 0.05269275
##
## , , Sick = yes
##
##          LungParench
## Grunting  Abnormal  Congested      Normal
## no       0.19104084 0.60096154 0.79949622
## yes      0.80895916 0.39903846 0.20050378
##
##
## Parameters of node LVHreport (multinomial distribution)
##
## Conditional probability table:
##
##          LungFlow
## LVHreport  High      Low      Normal
## no       0.8404053 0.6096246 0.7738743
## yes      0.1595947 0.3903754 0.2261257
##
## Parameters of node LowerBodyO2 (multinomial distribution)
##

```

```

## Conditional probability table:
##
## , , HypoxiaInO2 = Mild
##
##           HypDistrib
## LowerBodyO2      Equal      Unequal
##      <5    0.09552600 0.35593220
##      12+    0.59008464 0.05508475
##      5-12    0.31438936 0.58898305
##
## , , HypoxiaInO2 = Moderate
##
##           HypDistrib
## LowerBodyO2      Equal      Unequal
##      <5    0.31324278 0.49038462
##      12+    0.10165535 0.05288462
##      5-12    0.58510187 0.45673077
##
## , , HypoxiaInO2 = Severe
##
##           HypDistrib
## LowerBodyO2      Equal      Unequal
##      <5    0.49052269 0.59633028
##      12+    0.11143021 0.04281346
##      5-12    0.39804710 0.36085627
##
##
## Parameters of node RUQO2 (multinomial distribution)
##
## Conditional probability table:
##
##           HypoxiaInO2
## RUQO2      Mild Moderate      Severe
##      <5    0.1251176 0.2948518 0.4964558
##      12+    0.5813735 0.1023791 0.1079023
##      5-12    0.2935089 0.6027691 0.3956419
##
## Parameters of node CO2Report (multinomial distribution)
##
## Conditional probability table:
##
##           CO2
## CO2Report      High      Low      Normal
##      <7.5    0.09728623 0.89030023 0.90628046
##      >=7.5    0.90271377 0.10969977 0.09371954
##
## Parameters of node XrayReport (multinomial distribution)
##
## Conditional probability table:
##
##           ChestXray
## XrayReport      Asy/Patch Grd_Glass      Normal Oligaemic Plethoric
## Asy/Patchy 0.70207852 0.17474634 0.06840959 0.05291320 0.06310905
## Grd_Glass  0.09853734 0.63585118 0.01960784 0.02318668 0.02041763

```

```

## Normal      0.08160123 0.06877114 0.79433551 0.09601665 0.09559165
## Oligaemic   0.01924557 0.01465614 0.05925926 0.80707491 0.02134571
## Plethoric   0.09853734 0.10597520 0.05838780 0.02080856 0.79953596
##
## Parameters of node Disease (multinomial distribution)
##
## Conditional probability table:
##
## , , LungParench = Abnormal, LungFlow = High
##
##      LVH
## Disease      no      yes
## Fallot 0.0435643564 0.0394736842
## Lung 0.1009900990 0.1052631579
## PAIVS 0.0000000000 0.1710526316
## PFC 0.0059405941 0.0000000000
## TAPVD 0.1643564356 0.0657894737
## TGA 0.6851485149 0.6184210526
##
## , , LungParench = Congested, LungFlow = High
##
##      LVH
## Disease      no      yes
## Fallot 0.0194174757 0.0434782609
## Lung 0.0711974110 0.0869565217
## PAIVS 0.0000000000 0.0869565217
## PFC 0.0064724919 0.0000000000
## TAPVD 0.5501618123 0.2608695652
## TGA 0.3527508091 0.5217391304
##
## , , LungParench = Normal, LungFlow = High
##
##      LVH
## Disease      no      yes
## Fallot 0.0501829587 0.0421686747
## Lung 0.0005227392 0.0000000000
## PAIVS 0.0057501307 0.2771084337
## PFC 0.0052273915 0.0030120482
## TAPVD 0.0167276529 0.0000000000
## TGA 0.9215891270 0.6777108434
##
## , , LungParench = Abnormal, LungFlow = Low
##
##      LVH
## Disease      no      yes
## Fallot 0.6254826255 0.0938511327
## Lung 0.0675675676 0.0226537217
## PAIVS 0.0637065637 0.8349514563
## PFC 0.1833976834 0.0226537217
## TAPVD 0.0154440154 0.0064724919
## TGA 0.0444015444 0.0194174757
##
## , , LungParench = Congested, LungFlow = Low
##

```

```

##          LVH
## Disease          no          yes
## Fallot 0.5212765957 0.0636363636
## Lung   0.0744680851 0.0181818182
## PAIVS  0.0265957447 0.8909090909
## PFC    0.1382978723 0.0181818182
## TAPVD  0.2234042553 0.0090909091
## TGA    0.0159574468 0.0000000000
##
## , , LungParench = Normal, LungFlow = Low
##
##          LVH
## Disease          no          yes
## Fallot 0.7859154930 0.1162196679
## Lung   0.0018779343 0.0012771392
## PAIVS  0.0755868545 0.8627075351
## PFC    0.0807511737 0.0127713921
## TAPVD  0.0014084507 0.0000000000
## TGA    0.0544600939 0.0070242656
##
## , , LungParench = Abnormal, LungFlow = Normal
##
##          LVH
## Disease          no          yes
## Fallot 0.1108742004 0.0641025641
## Lung   0.4776119403 0.3205128205
## PAIVS  0.0085287846 0.3717948718
## PFC    0.1023454158 0.0512820513
## TAPVD  0.1236673774 0.0512820513
## TGA    0.1769722814 0.1410256410
##
## , , LungParench = Congested, LungFlow = Normal
##
##          LVH
## Disease          no          yes
## Fallot 0.0854700855 0.1764705882
## Lung   0.3632478632 0.2058823529
## PAIVS  0.0000000000 0.3529411765
## PFC    0.0299145299 0.0588235294
## TAPVD  0.4059829060 0.1470588235
## TGA    0.1153846154 0.0588235294
##
## , , LungParench = Normal, LungFlow = Normal
##
##          LVH
## Disease          no          yes
## Fallot 0.3450164294 0.1296928328
## Lung   0.0142387733 0.0034129693
## PAIVS  0.0197152245 0.6621160410
## PFC    0.0865279299 0.0307167235
## TAPVD  0.0131434830 0.0000000000
## TGA    0.5213581599 0.1740614334
##
##

```

```

## Parameters of node GruntingReport (multinomial distribution)
##
## Conditional probability table:
##
##           Grunting
## GruntingReport      no      yes
##           no 0.90098497 0.20227671
##           yes 0.09901503 0.79772329
##
## Parameters of node Age (multinomial distribution)
##
## Conditional probability table:
##
##           Disease
## Age      Fallot      Lung      PAIVS      PFC      TAPVD
## 0-3_days 0.34474412 0.87673956 0.80008768 0.90554415 0.77756654
## 11-30_days 0.43291840 0.02982107 0.04647085 0.03696099 0.07794677
## 4-10_days 0.22233748 0.09343936 0.15344147 0.05749487 0.14448669
##           Disease
## Age      TGA
## 0-3_days 0.73391725
## 11-30_days 0.08607671
## 4-10_days 0.18000604
##
## Parameters of node LVH (multinomial distribution)
##
## Conditional probability table:
##
## , , LungFlow = High
##
##           LVHreport
## LVH      no      yes
## no 0.97965335 0.25198413
## yes 0.02034665 0.74801587
##
## , , LungFlow = Low
##
##           LVHreport
## LVH      no      yes
## no 0.92208234 0.06695005
## yes 0.07791766 0.93304995
##
## , , LungFlow = Normal
##
##           LVHreport
## LVH      no      yes
## no 0.97442455 0.20131291
## yes 0.02557545 0.79868709
##
## Parameters of node DuctFlow (multinomial distribution)
##
## Conditional probability table:
##

```

```

##           Disease
## DuctFlow      Fallot      Lung      PAIVS      PFC      TAPVD
##   Lt_to_Rt 0.79218534 0.19880716 1.00000000 0.15605749 0.32509506
##   None      0.20781466 0.37574553 0.00000000 0.04106776 0.31558935
##   Rt_to_Lt 0.00000000 0.42544732 0.00000000 0.80287474 0.35931559
##           Disease
## DuctFlow      TGA
##   Lt_to_Rt 0.09634551
##   None      0.79432196
##   Rt_to_Lt 0.10933253
##
## Parameters of node CardiacMixing (multinomial distribution)
##
## Conditional probability table:
##
##           Disease
## CardiacMixing      Fallot      Lung      PAIVS      PFC      TAPVD
##   Complete 0.795643154 0.037773360 0.954844367 0.162217659 0.952471483
##   Mild      0.159059474 0.550695825 0.017974573 0.420944559 0.034220532
##   None      0.022130014 0.391650099 0.009206488 0.400410678 0.005703422
##   Transp.   0.023167358 0.019880716 0.017974573 0.016427105 0.007604563
##           Disease
## CardiacMixing      TGA
##   Complete 0.089700997
##   Mild      0.099365751
##   None      0.017215343
##   Transp.   0.793717910
##
## Parameters of node LungParench (multinomial distribution)
##
## Conditional probability table:
##
##           ChestXray
## LungParench Asy/Patch Grd_Glass      Normal      Oligaemic      Plethoric
##   Abnormal 0.81293303 0.30777903 0.08976035 0.10047562 0.03805104
##   Congested 0.08083141 0.51071026 0.02178649 0.02556480 0.09466357
##   Normal    0.10623557 0.18151071 0.88845316 0.87395957 0.86728538
##
## Parameters of node LungFlow (multinomial distribution)
##
## Conditional probability table:
##
## , , LungParench = Abnormal
##
##           ChestXray
## LungFlow Asy/Patch Grd_Glass      Normal      Oligaemic      Plethoric
##   High    0.025568182 0.750915751 0.660194175 0.582840237 0.195121951
##   Low      0.557765152 0.146520147 0.203883495 0.328402367 0.548780488
##   Normal  0.416666667 0.102564103 0.135922330 0.088757396 0.256097561
##
## , , LungParench = Congested
##
##           ChestXray
## LungFlow Asy/Patch Grd_Glass      Normal      Oligaemic      Plethoric

```

```

##   High    0.33333333 0.289183223 0.400000000 0.104651163 0.671568627
##   Low     0.447619048 0.326710817 0.260000000 0.825581395 0.093137255
##   Normal 0.219047619 0.384105960 0.340000000 0.069767442 0.235294118
##
## , , LungParench = Normal
##
##           ChestXray
## LungFlow  Asy/Patch  Grd_Glass      Normal  Oligaemic  Plethoric
##   High    0.166666667 0.440993789 0.181951937 0.006122449 0.942750134
##   Low     0.579710145 0.472049689 0.282981854 0.984013605 0.037453184
##   Normal 0.253623188 0.086956522 0.535066209 0.009863946 0.019796683
##
##
## Parameters of node Sick (multinomial distribution)
##
## Conditional probability table:
##
## , , Age = 0-3_days
##
##           Disease
## Sick      Fallot      Lung      PAIVS      PFC      TAPVD      TGA
##   no  0.60581745 0.26984127 0.68602740 0.55555556 0.29828851 0.67489712
##   yes 0.39418255 0.73015873 0.31397260 0.44444444 0.70171149 0.32510288
##
## , , Age = 11-30_days
##
##           Disease
## Sick      Fallot      Lung      PAIVS      PFC      TAPVD      TGA
##   no  0.94089457 0.46666667 0.69811321 0.77777778 0.56097561 0.80000000
##   yes 0.05910543 0.53333333 0.30188679 0.22222222 0.43902439 0.20000000
##
## , , Age = 4-10_days
##
##           Disease
## Sick      Fallot      Lung      PAIVS      PFC      TAPVD      TGA
##   no  0.86469673 0.46808511 0.71142857 0.78571429 0.35526316 0.76845638
##   yes 0.13530327 0.53191489 0.28857143 0.21428571 0.64473684 0.23154362

```

#### 4.3 d)

```

cpquery(fitted.param,
        event = (Disease == "Lung"),
        evidence = ((C02 == "High") & (LungParench == "Abnormal")))

```

```
## [1] 0.1760417
```

The probability is 0.17

## Q5)

A Bayesian Network is used by the US Military for Combat Equipment Diagnostics. The military, or the more precise the man in the field, face the problem that at certain times they need to act quick and need to fully rely on their equipment. So, in case a vital element, like a truck, needs maintenance, this needs to be done as quickly as possible. An experienced technician, that can easily repair this might not arrive



within days. So, a causal Bayesian Network (DAG) that holds expert knowledge was developed. It helps and gives guidance for the soldiers to repair the equipment. This method was chosen, since it allows to easily illustrate complex relationships and visualises the causal path structures easily understandable for the soldiers Aebischer et al. (2017).

In the second application, the Bayes theorem is used to evaluate the performance of depression tests. The problem is that these diagnoses need to be as correct as possible. But these tests are often not that correct. To improve the accuracy of these test, multiples are combined. However, It is described that the usage of frequentists tests underestimates the possibility of misclassifications based on cut-offs. They only consider the risk of False Positives. By using the Bayes theorem, Aebischer et al. evaluate the tests newly. The result is that all besides one, reaches a satisfactory level of diagnostic accuracy. This best test uses a newer procedure to select the items. Thanks to the Bayes Theorem it could be shown that this method should be used to select the items, since they give a higher diagnostic accuracy Tommasi, Ferrara, and Saggino (2018).

## References

- Aebischer, David, John Vatterott, Michael Grimes, Andrew Vatterott, Roderick Jordan, Carlo Reinoso, Bradford Alex Baker, et al. 2017. "Bayesian networks for combat equipment diagnostics." *Interfaces* 47 (1): 85–105. <https://doi.org/10.1287/inte.2016.0883>.
- Tommasi, Marco, Grazia Ferrara, and Aristide Saggino. 2018. "Application of bayes' theorem in valuating depression tests performance." *Frontiers in Psychology* 9 (JUL). <https://doi.org/10.3389/fpsyg.2018.01240>.