# BAYESIAN NETWORKS 3rd Exercise Block 2

*Jeremy Williams*

*January 16th, 2018*

```r
#install.packages("bnlearn")
#install.packages("gRain")
#source("http://bioconductor.org/biocLite.R")
#biocLite("Rgraphviz")
#biocLite("RBGL")

suppressMessages(suppressWarnings(library("bnlearn")))
suppressMessages(suppressWarnings(library("gRain")))
suppressMessages(suppressWarnings(library("gRbase")))
suppressMessages(suppressWarnings(library("Rgraphviz")))
suppressMessages(suppressWarnings(library("RBGL")))


#### Model 1  KIT A



tf<-c("false","true")

# Specify the CPTs:
node.E<- cptable(~ E, values=c(8,2),levels=tf)
node.B<- cptable(~ B, values=c(1,9), levels=tf)
node.S<- cptable(~ S + E + B, values=c(1,0,1,0,99,1,3,97), levels=tf)
node.S2<- cptable(~ S2 + E + B, values=c(1,0,1,0,99,1,3,97), levels=tf)

plist<-compileCPT(list(node.E,node.B,node.S,node.S2))
plist
```

```
## CPTspec with probabilities:
##  P( E )
##  P( B )
##  P( S | E B )
##  P( S2 | E B )
```

```r
plist$E
```

```
## E
## false  true
##   0.8   0.2
## attr(,"class")
## [1] "parray" "array"
```

```r
plist$B
```

```
## B
## false  true
##   0.1   0.9
## attr(,"class")
## [1] "parray" "array"
```

```r
plist$S
```

```
## , , B = false
##
##        E
## S       false true
##   false     1    1
##   true      0    0
##
## , , B = true
##
##        E
## S       false true
##   false  0.99 0.03
##   true   0.01 0.97
##
## attr(,"class")
## [1] "parray" "array"
```

```r
plist$S2
```

```
## , , B = false
##
##        E
## S2      false true
##   false     1    1
##   true      0    0
##
## , , B = true
##
##        E
## S2      false true
##   false  0.99 0.03
##   true   0.01 0.97
##
## attr(,"class")
## [1] "parray" "array"
```

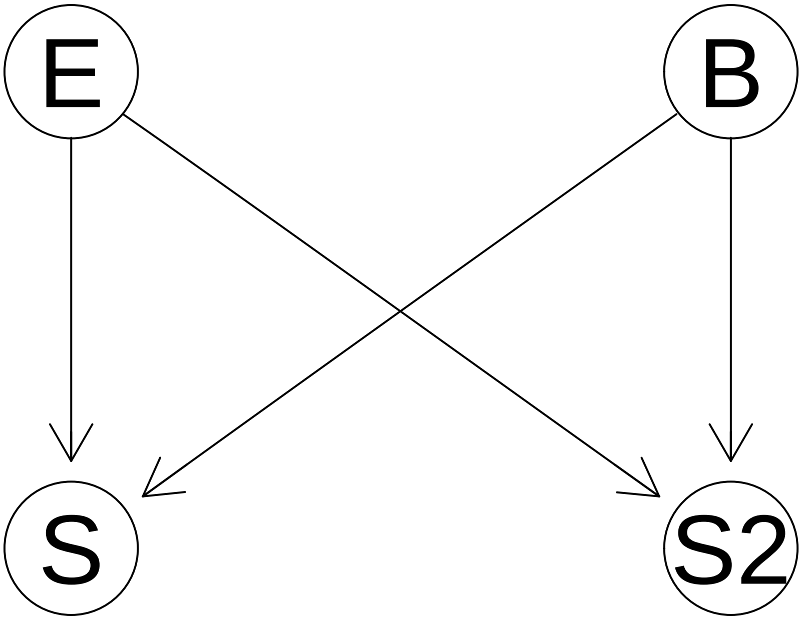
```r
# Create a network of name "Norman.net", for instance:

Norman.net<-grain(plist)
summary(Norman.net)
```

```
## Independence network: Compiled: FALSE Propagated: FALSE
##  Nodes : chr [1:4] "E" "B" "S" "S2"
```

```r
# The graph:

plot1=plot(Norman.net)
```



```
plot1
```

```
## [1] "A graph with 4 nodes."
```

```
# We can compute the marginal probability

# of each variable

# These probabilities are EXACT!!

querygrain(Norman.net,nodes=c("E", "B","S", "S2"),type="marginal")
```

```
## $E
## E
## false  true
##   0.8   0.2
##
## $B
## B
## false  true
##   0.1   0.9
##
## $S
## S
##  false   true
## 0.8182 0.1818
##
```

```
## $S2
## S2
##  false   true
## 0.8182 0.1818
```

```
querygrain(Norman.net,nodes=c("S","S2"), type="joint")
```

```
##        S2
## S          false     true
##   false 0.805834 0.012366
##   true  0.012366 0.169434
## attr(,"class")
## [1] "parray" "array"
```

```
#Question 1

Norman.net.1<-setEvidence(Norman.net,nodes=c("S","S2"),states=c("false","false"))

Norman.net.1
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##   Nodes: chr [1:4] "E" "B" "S" "S2"
##   Evidence:
##   nodes is.hard.evidence hard.state
## 1     S             TRUE      false
## 2    S2             TRUE      false
##   pEvidence: 0.805834
```

```
predOT<-querygrain(Norman.net.1,nodes=c("E"), type="marginal")
predOT
```

```
## $E
## E
##      false       true
## 0.97497996 0.02502004
```

```
predOT$E[["true"]]
```

```
## [1] 0.02502004
```

```
#Question 2
Norman.net.2<-setEvidence(Norman.net,nodes=c("S","S2"),states=c("true","true"))

Norman.net.2
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##   Nodes: chr [1:4] "E" "B" "S" "S2"
##   Evidence:
##   nodes is.hard.evidence hard.state
## 1     S             TRUE       true
## 2    S2             TRUE       true
##   pEvidence: 0.169434
```

```
predOT<-querygrain(Norman.net.2,nodes=c("E"), type="marginal")
predOT
```

```
## $E
## E
##        false         true
## 0.0004249442 0.9995750558
```

```
predOT$E[["true"]]
```

```
## [1] 0.9995751
```

```
#Question 3

Norman.net.3<-setEvidence(Norman.net,nodes=c("S","S2"),states=c("true","false"))

Norman.net.3
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##   Nodes: chr [1:4] "E" "B" "S" "S2"
##   Evidence:
##   nodes is.hard.evidence hard.state
## 1     S             TRUE       true
## 2    S2             TRUE      false
##   pEvidence: 0.012366
```

```
predOT<-querygrain(Norman.net.3,nodes=c("E"), type="marginal")
predOT
```

```
## $E
## E
##     false      true
## 0.5764192 0.4235808
```

```
predOT$E[["true"]]
```

```
## [1] 0.4235808
```

```
#### Model 2  KIT B


tf<-c("false","true")

# Specify the CPTs:
node.E<- cptable(~ E, values=c(8,2),levels=tf)
node.B<- cptable(~ B, values=c(1,9), levels=tf)
node.B1<- cptable(~ B1, values=c(1,9), levels=tf)
node.S<- cptable(~ S + E + B, values=c(1,0,1,0,99,1,3,97), levels=tf)
node.S2<- cptable(~ S2 + E + B1, values=c(1,0,1,0,99,1,3,97), levels=tf)

plist<-compileCPT(list(node.E,node.B,node.B1,node.S,node.S2))
plist
```

```
## CPTspec with probabilities:
##  P( E )
##  P( B )
##  P( B1 )
##  P( S | E B )
##  P( S2 | E B1 )
```

```
plist$E
```

```
## E
## false  true
##   0.8   0.2
## attr(,"class")
## [1] "parray" "array"
```

```
plist$B
```

```
## B
## false  true
##   0.1   0.9
## attr(,"class")
## [1] "parray" "array"
```

```
plist$B1
```

```
## B1
## false  true
##   0.1   0.9
## attr(,"class")
## [1] "parray" "array"
```

```
plist$S
```

```
## , , B = false
##
##         E
## S        false true
##   false      1    1
##   true       0    0
##
## , , B = true
##
##         E
## S        false true
##   false   0.99 0.03
##   true    0.01 0.97
##
## attr(,"class")
## [1] "parray" "array"
```

```
plist$S2
```

```
## , , B1 = false
##
##         E
## S2       false true
##   false      1    1
##   true       0    0
##
## , , B1 = true
##
##         E
## S2       false true
##   false   0.99 0.03
##   true    0.01 0.97
##
## attr(,"class")
## [1] "parray" "array"
```

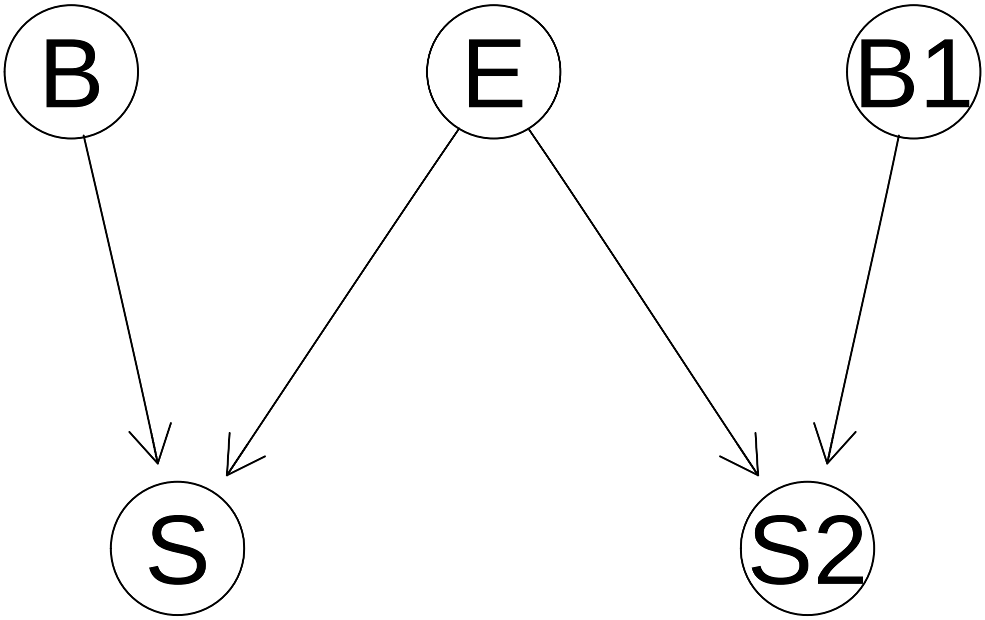
```
# Create a network of name "Norman.net", for instance:

Norman.net<-grain(plist)
summary(Norman.net)
```

```
## Independence network: Compiled: FALSE Propagated: FALSE
##  Nodes : chr [1:5] "E" "B" "B1" "S" "S2"
```

```
# The graph:

plot1=plot(Norman.net)
```



```
plot1
```

```
## [1] "A graph with 5 nodes."
```

```
# We can compute the marginal probability

# of each variable

# These probabilities are EXACT!!

querygrain(Norman.net,nodes=c("E", "B","B1","S", "S2"),type="marginal")
```

```
## $E
## E
## false  true
##   0.8   0.2
```

```
##
## $B
## B
## false  true
##   0.1   0.9
##
## $S
## S
##  false   true
## 0.8182 0.1818
##
## $B1
## B1
## false  true
##   0.1   0.9
##
## $S2
## S2
##  false   true
## 0.8182 0.1818
```

```
querygrain(Norman.net,nodes=c("S","S2"), type="joint")
```

```
##        S2
## S           false      true
##   false 0.7888906 0.0293094
##   true  0.0293094 0.1524906
## attr(,"class")
## [1] "parray" "array"
```

```
#Question 1

Norman.net.1<-setEvidence(Norman.net,nodes=c("S","S2"),states=c("false","false"))

Norman.net.1
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##   Nodes: chr [1:5] "E" "B" "B1" "S" "S2"
##   Evidence:
##   nodes is.hard.evidence hard.state
## 1     S              TRUE      false
## 2    S2              TRUE      false
##   pEvidence: 0.788891
```

```
predOT<-querygrain(Norman.net.1,nodes=c("E"), type="marginal")
predOT
```

```
## $E
## E
##       false        true
## 0.995910967 0.004089033
```

```
predOT$E[["true"]]
```

```
## [1] 0.004089033
```

```
#Question 2
Norman.net.2<-setEvidence(Norman.net,nodes=c("S","S2"),states=c("true","true"))
Norman.net.2
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##   Nodes: chr [1:5] "E" "B" "B1" "S" "S2"
##   Evidence:
##   nodes is.hard.evidence hard.state
## 1     S              TRUE       true
## 2    S2              TRUE       true
##   pEvidence: 0.152491
```

```
predOT<-querygrain(Norman.net.2,nodes=c("E"), type="marginal")
predOT
```

```
## $E
## E
##        false          true
## 0.0004249442 0.9995750558
```

```
predOT$E[["true"]]
```

```
## [1] 0.9995751
```

```
#Question 3

Norman.net.3<-setEvidence(Norman.net,nodes=c("S","S2"),states=c("true","false"))

Norman.net.3
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##   Nodes: chr [1:5] "E" "B" "B1" "S" "S2"
##   Evidence:
##   nodes is.hard.evidence hard.state
## 1     S              TRUE       true
## 2    S2              TRUE      false
##   pEvidence: 0.029309
```

```
predOT<-querygrain(Norman.net.3,nodes=c("E"), type="marginal")
predOT
```

```
## $E
## E
##     false      true
## 0.2434441 0.7565559
```

```
predOT$E[["true"]]
```

```
## [1] 0.7565559
```