drangapu_hw5

2024-05-09

1) Elements of Statistical Learning #17.2

```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
cond_idpt <- function(graph,n1,n2,nodes) {
  subgraph <- delete_vertices(graph,nodes)
  paths <- all_simple_paths(subgraph,from=n1,to =n2)
  return(length(paths)== 0)
}
edges_a <- matrix(c("X", "Y", "Y", "Z"), ncol=2, byrow=TRUE)
edges_b <- matrix(c("X", "Y", "Y", "W", "Z", "Z"), ncol=2, byrow=TRUE)
edges_c <- matrix(c("X", "Y", "Y", "Z", "Z", "W", "X", "W"), ncol=2, byrow=TRUE)
edges_d <- matrix(c("X", "Y", "Y", "Z", "Z", "W"), ncol=2, byrow=TRUE)

graph_a <- graph_from_edgelist(edges_a, directed = FALSE)
graph_b <- graph_from_edgelist(edges_b, directed = FALSE)
graph_c <- graph_from_edgelist(edges_c, directed = FALSE)
graph_d <- graph_from_edgelist(edges_d, directed = FALSE)

cat("Graph (a): Are X and Z conditionally independent given Y?", cond_idpt(graph_a, "X", "Z", "Y"), "\n
```

```
## Graph (a): Are X and Z conditionally independent given Y? TRUE
```

```
cat("Graph (b): Is Z conditionally independent given X,Y,W?  ",cond_idpt(graph_b, "Z", "X", c("Y","W"))
```

```
## Graph (b): Is Z conditionally independent given X,Y,W?   TRUE
```

```
cat("Graph (c): Are X and Z conditionally independent given Y? ",
    cond_idpt(graph_c, "X", "Z", "Y"), "\n")
```

```
## Graph (c): Are X and Z conditionally independent given Y?  FALSE
```

```r
cat("Graph (c): Are X and W conditionally independent given Y and Z? ",
    cond_idpt(graph_c, "X", "W", c("Y", "Z")), "\n")
```

```
## Graph (c): Are X and W conditionally independent given Y and Z?  FALSE
```

```r
cat("Graph (d): Are X and W conditionally independent given Y and Z? ",
    cond_idpt(graph_d, "X", "W", c("Y", "Z")), "\n")
```

```
## Graph (d): Are X and W conditionally independent given Y and Z?  TRUE
```

Consider the following webgraphs. (a) Compute the PageRank vector of Webgraph A for damping constants p = 0.05, 0.25, 0.50, 0.75, and 0.95. How sensitive is the PageRank vector, and overall ranking of importance, to the damping constant? Does the relative ranking of importance according to PageRank support your intuition? (b) Compute the PageRank vector of Webgraph B for damping constant p = 0.15. Interpret your results in terms of the relationship between the number of incoming links that each node has. Does the relative ranking of importance according to PageRank support your intuition?

```r
library(igraph)
edges_A <- c('C', 'A', 'B', 'C', 'D', 'B', 'B', 'E', 'E', 'D', 'D', 'E', 'F', 'C')
graph_A <- graph(edges_A, directed = TRUE)
edges_B <- c('B', 'A', 'C', 'A', 'D', 'B', 'E', 'B', 'F', 'C', 'G', 'C', 'H', 'C')
graph_B <- graph(edges_B, directed = TRUE)

dmp_factors <- c(0.05, 0.25, 0.50, 0.75, 0.95)
pagernk_A <- lapply(dmp_factors, function(d) {
    page_rank(graph_A, directed = TRUE, damping = d)$vector
})
print("PageRank for Webgraph A with given damping factors:")
```

```
## [1] "PageRank for Webgraph A with given damping factors:"
```

```r
print(pagernk_A)
```

```
## [[1]]
##         C         A         B         D         E         F
## 0.1718214 0.1683271 0.1639395 0.1681380 0.1680380 0.1597361
##
## [[2]]
##         C         A         B         D         E         F
## 0.1848587 0.1786588 0.1544288 0.1758772 0.1737324 0.1324441
##
## [[3]]
##          C          A          B          D          E          F
## 0.18583257 0.19227231 0.14719411 0.19135235 0.18399264 0.09935603
##
## [[4]]
##          C          A          B          D          E          F
## 0.17077331 0.19399617 0.14778661 0.21832113 0.20320659 0.06591619
##
## [[5]]
##          C          A          B          D          E          F
## 0.14454445 0.17305017 0.15761096 0.25658531 0.23247617 0.03573294
```

```r
pagernk_B <- page_rank(graph_B, directed = TRUE, damping = 0.15)$vector

print("PageRank for Webgraph B with damping factor 0.15:")
```

```
## [1] "PageRank for Webgraph B with damping factor 0.15:"
```

```r
print(pagernk_B)
```

```
##         B         A         C         D         E         F         G         H 
## 0.1418827 0.1541610 0.1582538 0.1091405 0.1091405 0.1091405 0.1091405 0.1091405
```

Webgraph A:

- Node C consistently shows one of the highest PageRank scores as a central node.
- Node A increased importance with higher damping factors due to more random iterations.
- Nodes D and E are tightly connected and also shows increase in rank with higher damping factors with close interactions
- Node B decreases in importance as the damping factor rises with lesser centrality.
- Node F is the least important because of minimal direct influence.

  The PageRank in Webgraph A is sensitive to the damping factor benefiting nodes in dense connections (D, E) or central nodes (C) as the factor increases.

Webgraph B:

- Nodes B and C are key connectors directing towards Node A and due to their intermediary roles.
- Node A is the most crucial because it is collecting the influence passed from B and C.
- Nodes D, E, F, G, H all are lower and equal ranks, each of them are contributing same to B and C, showing no additional structure.

Conclusion:

The PageRank values align well with network structures, where central nodes (like C in A and A in B) are higher importance. The damping factor's influence is crucial in importance of the roles of nodes based on their connectivity and position within the graph.

Specify the structure of a Bayesian Network that contains four nodes {W,X,Y,Z} and has satisfies the following set of independencies.

Given Independencies:

1. W and X are independent.
2. W and Z are not independent given X.
3. Z and W are independent given Y.
4. W and Y are not independent.

5. X and Y are not independent.

6. W and X are not independent given Z.

7. X and Z are independent given W and Y.

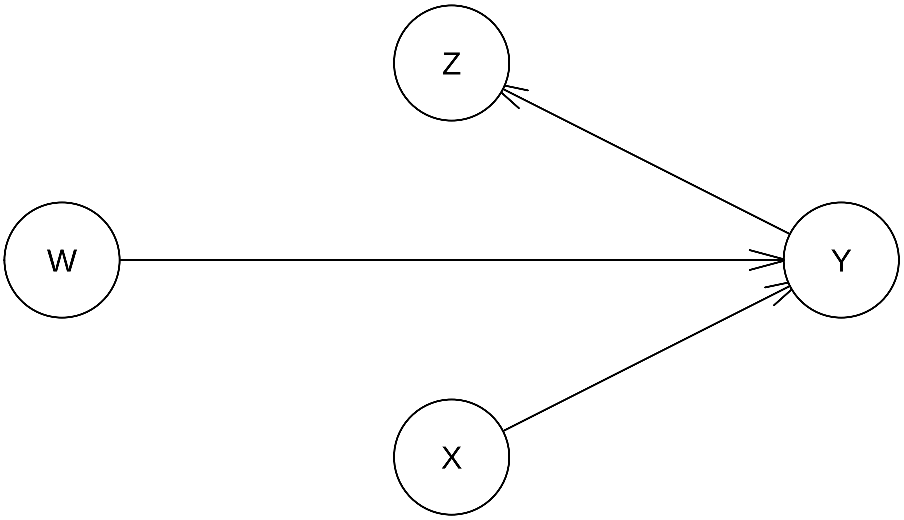
Network Structure:

- .Y influence's both W and X to maintain their dependence on Y but independence from each other when Y is not conditioned.
- Z is influenced by Y, and have a conditional dependency with W that can be overridden by conditioning on Y.
- There is no direct path between X and Z that is either through W or Y and their connection must be mediated.
- Y → W, Y → X: Directly influences from Y to W and X to establish their dependency on Y.
- Y → Z: Z's dependency on Y helps in controlling the influence of Z on W when Y is conditioned.
- Z → W: This creates a pathway where Z can influence W but only when X is not conditioned.

```
library(bnlearn)
```

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

## The following objects are masked from 'package:igraph':
##
##     as.igraph, compare, degree, subgraph
```

```
dag <- model2network("[X][W][Y|X:W][Z|Y]")
plot(dag)
```



```r
cat("W independent of X:", dsep(dag, "W", "X"), "\n")
```

```
## W independent of X: TRUE
```

```r
cat("W not independent of Z given X:", !dsep(dag, "W", "Z", c("X")), "\n")
```

```
## W not independent of Z given X: TRUE
```

```r
cat("Z independent of W given Y:", dsep(dag, "Z", "W", "Y"), "\n")
```

```
## Z independent of W given Y: TRUE
```

```r
cat("W not independent of Y:", !dsep(dag, "W", "Y"), "\n")
```

```
## W not independent of Y: TRUE
```

```r
cat("X not independent of Y:", !dsep(dag, "X", "Y"), "\n")
```

```
## X not independent of Y: TRUE
```

```
cat("W not independent of X given Z:", !dsep(dag, "W", "X", "Z"), "\n")
```

```
## W not independent of X given Z: TRUE
```

```
cat("X independent of Z given W, Y:", dsep(dag, "X", "Z", c("W", "Y")), "\n")
```

```
## X independent of Z given W, Y: TRUE
```

Data released from the US department of Commerce, Bureau of the Census is available in R.

>data(state)

>?state

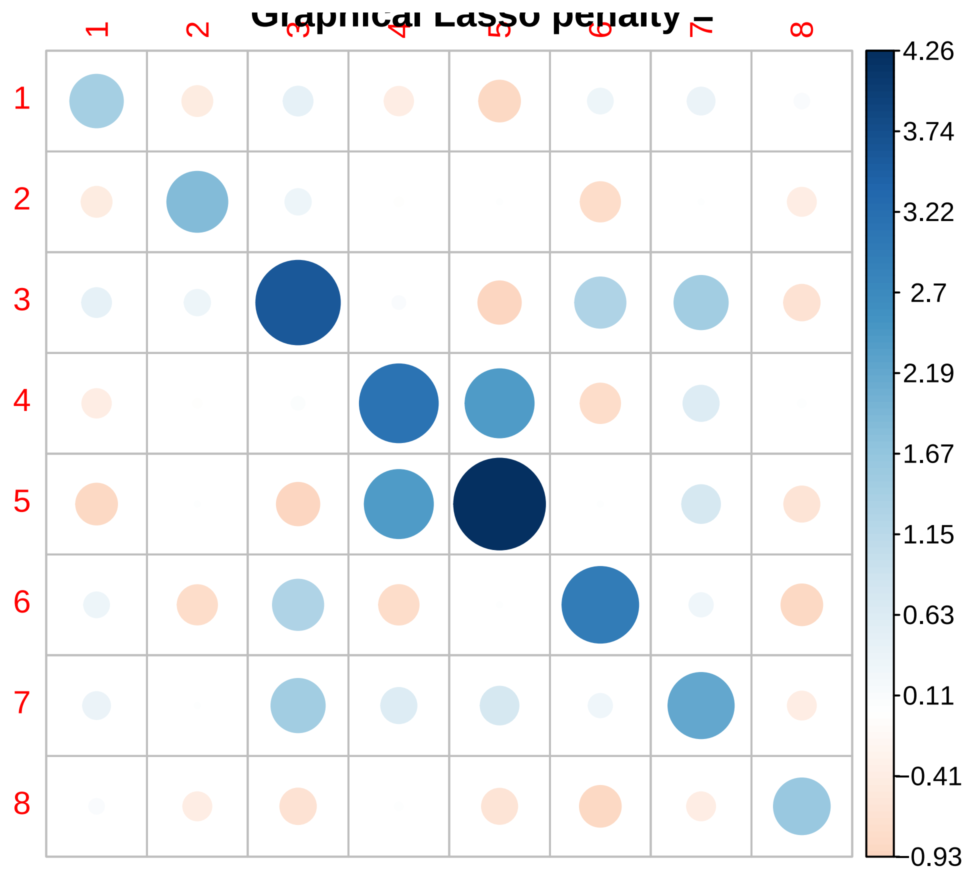
Build a Gaussian Graphical Model using the Graphical Lasso for the 8 predictors (Population, Income, Illiteracy, Life Exp, Murder, HS Grad, Frost, Area) using a range of penalties. What do you find for different penalties, and how does it compliment (and/or contradict) a model fit with SOM?

```
library(glasso)
library(kohonen)
library(corrplot)
```

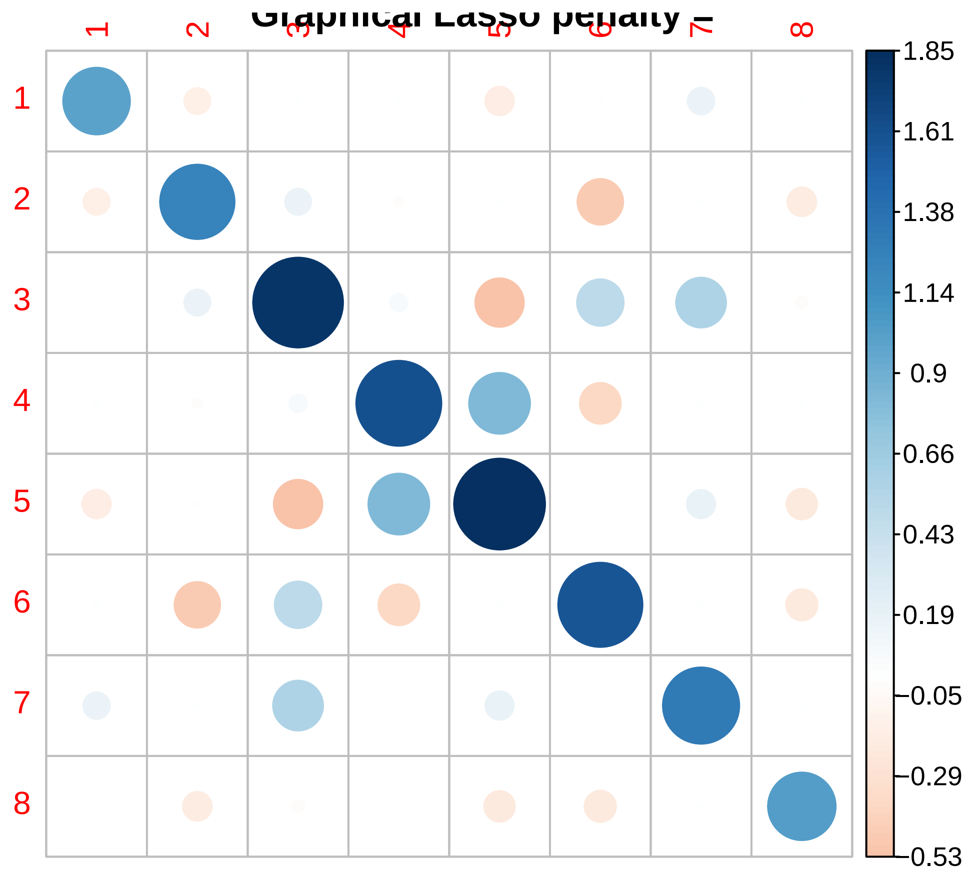
```
## corrplot 0.92 loaded
```

```
data(state)
states <- data.frame(state.x77)
predictors <- states[, c("Population", "Income", "Illiteracy", "Life.Exp", "Murder", "HS.Grad", "Frost"
predictors <- na.omit(predictors)
predictors <- scale(predictors)

penalties <- c(0.01, 0.1, 1, 10)
res <- lapply(penalties, function(pen) glasso(cov(predictors), rho=pen))

lapply(res, function(model) {print(paste("Penalty:", model$rho))
  corrplot(model$wi,is.corr = FALSE, title= paste("Graphical Lasso penalty =", model$rho))
})
```

```
## [1] "Penalty: "
```

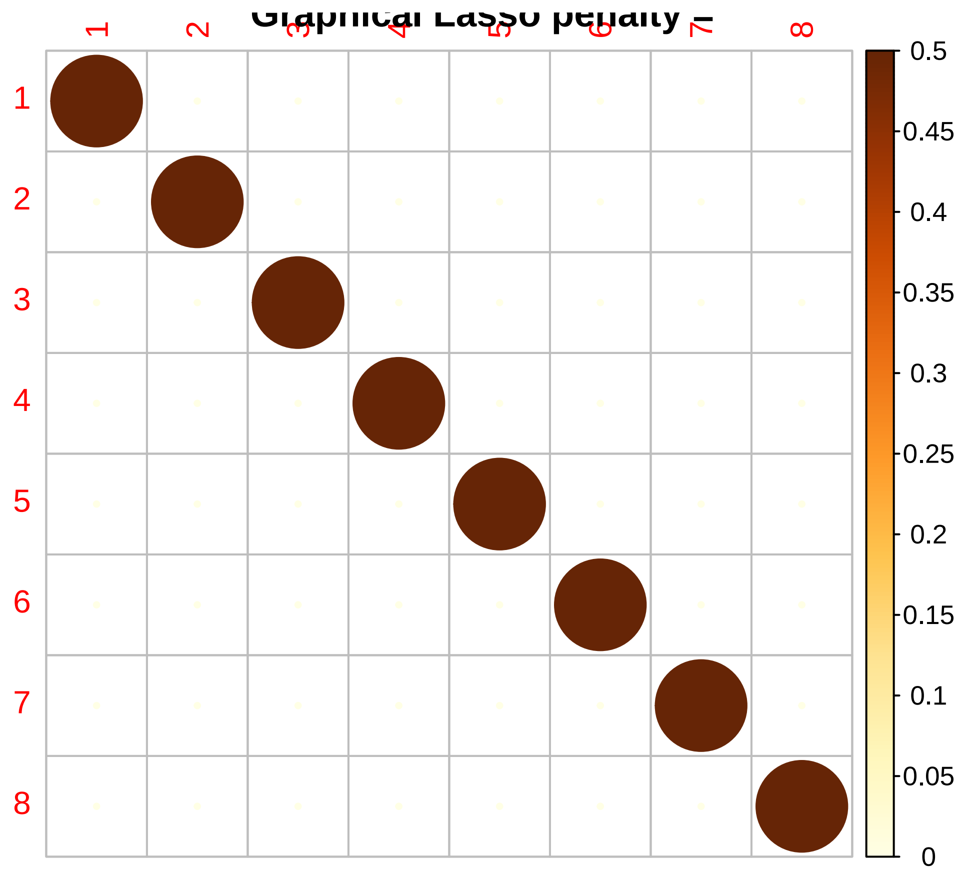

Graphical Lasso penalty =
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7
8
4.26
3.74
3.22
2.7
2.19
1.67
1.15
0.63
0.11
−0.41
−0.93

```
## [1] "Penalty: "
```

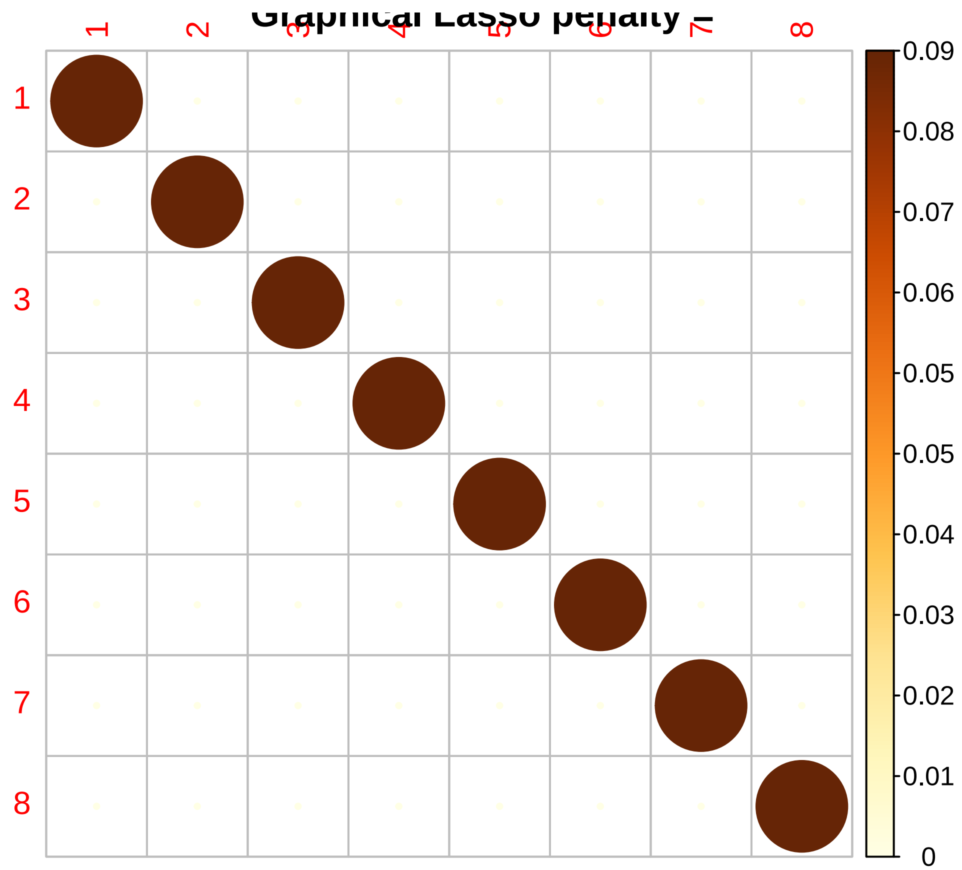


Graphical Lasso penalty =
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7
8
1.85
1.61
1.38
1.14
0.9
0.66
0.43
0.19
-0.05
-0.29
-0.53

```
## [1] "Penalty: "
```


Graphical Lasso penalty =
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7
8
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0

```
## [1] "Penalty: "
```



```
## [[1]]
## [[1]]$corr
##             1            2           3            4          5           6
## 1  0.33473537 -0.108728019  0.10098295 -0.098202272 -0.2017036  0.07482857
## 2 -0.10872656  0.436069781  0.07825520 -0.009548604  0.0000000 -0.18676735
## 3  0.10097723  0.078258405  0.84615835  0.020003561 -0.2176566  0.30599329
## 4 -0.09820514 -0.009541918  0.01997066  0.735460567  0.5620805 -0.18851456
## 5 -0.20169401  0.000000000 -0.21768507  0.562052710  1.0000000  0.00000000
## 6  0.07482895 -0.186771089  0.30600188 -0.188512304  0.0000000  0.69446814
## 7  0.08789702  0.000000000  0.34391871  0.149993407  0.1725359  0.06598240
## 8  0.02641673 -0.095342735 -0.15277161  0.007591715 -0.1494768 -0.20130212
##             7            8
## 1  0.08789563  0.026417079
## 2  0.00000000 -0.095344534
## 3  0.34393079 -0.152779347
## 4  0.14998914  0.007597016
## 5  0.17253199 -0.149471694
## 6  0.06598755 -0.201307915
## 7  0.51577289 -0.094867117
## 8 -0.09486631  0.377216849
##
## [[1]]$corrPos
##    xName yName x y         corr
## 1      1     1 1 8  0.334735366
## 2      1     2 1 7 -0.108726555
## 3      1     3 1 6  0.100977227
## 4      1     4 1 5 -0.098205135
## 5      1     5 1 4 -0.201694014
## 6      1     6 1 3  0.074828954
## 7      1     7 1 2  0.087897025
## 8      1     8 1 1  0.026416729
## 9      2     1 2 8 -0.108728019
## 10     2     2 2 7  0.436069781
## 11     2     3 2 6  0.078258405
## 12     2     4 2 5 -0.009541918
## 13     2     5 2 4  0.000000000
## 14     2     6 2 3 -0.186771089
## 15     2     7 2 2  0.000000000
## 16     2     8 2 1 -0.095342735
## 17     3     1 3 8  0.100982951
## 18     3     2 3 7  0.078255202
## 19     3     3 3 6  0.846158348
## 20     3     4 3 5  0.019970662
## 21     3     5 3 4 -0.217685072
## 22     3     6 3 3  0.306001882
## 23     3     7 3 2  0.343918710
## 24     3     8 3 1 -0.152771612
## 25     4     1 4 8 -0.098202272
## 26     4     2 4 7 -0.009548604
## 27     4     3 4 6  0.020003561
## 28     4     4 4 5  0.735460567
## 29     4     5 4 4  0.562052710
## 30     4     6 4 3 -0.188512304
## 31     4     7 4 2  0.149993407
## 32     4     8 4 1  0.007591715
## 33     5     1 5 8 -0.201703564
## 34     5     2 5 7  0.000000000
## 35     5     3 5 6 -0.217656607
## 36     5     4 5 5  0.562080485
## 37     5     5 5 4  1.000000000
## 38     5     6 5 3  0.000000000
## 39     5     7 5 2  0.172535910
## 40     5     8 5 1 -0.149476811
## 41     6     1 6 8  0.074828574
## 42     6     2 6 7 -0.186767349
## 43     6     3 6 6  0.305993291
## 44     6     4 6 5 -0.188514560
## 45     6     5 6 4  0.000000000
## 46     6     6 6 3  0.694468139
## 47     6     7 6 2  0.065982403
## 48     6     8 6 1 -0.201302117
## 49     7     1 7 8  0.087895629
## 50     7     2 7 7  0.000000000
## 51     7     3 7 6  0.343930790
## 52     7     4 7 5  0.149989145
## 53     7     5 7 4  0.172531992
## 54     7     6 7 3  0.065987555
## 55     7     7 7 2  0.515772891
## 56     7     8 7 1 -0.094866310
## 57     8     1 8 8  0.026417079
## 58     8     2 8 7 -0.095344534
## 59     8     3 8 6 -0.152779347
## 60     8     4 8 5  0.007597016
## 61     8     5 8 4 -0.149471694
## 62     8     6 8 3 -0.201307915
## 63     8     7 8 2 -0.094867117
## 64     8     8 8 1  0.377216849
##
## [[1]]$arg
## [[1]]$arg$type
## [1] "full"
##
##
##
## [[2]]
## [[2]]$corr
##             1           2           3           4            5            6
## 1  0.53859221 -0.08142357  0.00000000  0.00000000 -0.098443504  0.000000000
## 2 -0.08142422  0.66904269  0.08190232 -0.01035670  0.000000000 -0.251867731
## 3  0.00000000  0.08190598  0.97726539  0.03766237 -0.284256237  0.261963921
## 4  0.00000000 -0.01035774  0.03766393  0.87563754  0.448479421 -0.201709450
## 5 -0.09844523  0.00000000 -0.28425228  0.44847897  1.000000000  0.006584715
## 6  0.00000000 -0.25186709  0.26196123 -0.20171014  0.006583755  0.858339787
## 7  0.08593637  0.00000000  0.30128149  0.00000000  0.096707622  0.000000000
## 8  0.00000000 -0.10018845 -0.01720994  0.00000000 -0.113493661 -0.118739286
##           7           8
## 1 0.0859368  0.00000000
## 2 0.0000000 -0.10018842
## 3 0.3012874 -0.01720786
## 4 0.0000000  0.00000000
## 5 0.0967055 -0.11349330
## 6 0.0000000 -0.11873888
## 7 0.7040157  0.00000000
## 8 0.0000000  0.55272274
##
## [[2]]$corrPos
##    xName yName x y         corr
## 1      1     1 1 8  0.538592209
## 2      1     2 1 7 -0.081424219
## 3      1     3 1 6  0.000000000
## 4      1     4 1 5  0.000000000
## 5      1     5 1 4 -0.098445226
## 6      1     6 1 3  0.000000000
## 7      1     7 1 2  0.085936370
## 8      1     8 1 1  0.000000000
## 9      2     1 2 8 -0.081423574
## 10     2     2 2 7  0.669042686
## 11     2     3 2 6  0.081905981
## 12     2     4 2 5 -0.010357740
## 13     2     5 2 4  0.000000000
## 14     2     6 2 3 -0.251867092
## 15     2     7 2 2  0.000000000
## 16     2     8 2 1 -0.100188446
## 17     3     1 3 8  0.000000000
## 18     3     2 3 7  0.081902321
## 19     3     3 3 6  0.977265386
## 20     3     4 3 5  0.037663933
## 21     3     5 3 4 -0.284252275
## 22     3     6 3 3  0.261961229
## 23     3     7 3 2  0.301281490
## 24     3     8 3 1 -0.017209939
## 25     4     1 4 8  0.000000000
## 26     4     2 4 7 -0.010356695
## 27     4     3 4 6  0.037662367
## 28     4     4 4 5  0.875637540
## 29     4     5 4 4  0.448478966
## 30     4     6 4 3 -0.201710141
## 31     4     7 4 2  0.000000000
## 32     4     8 4 1  0.000000000
## 33     5     1 5 8 -0.098443504
## 34     5     2 5 7  0.000000000
## 35     5     3 5 6 -0.284256237
## 36     5     4 5 5  0.448479421
## 37     5     5 5 4  1.000000000
## 38     5     6 5 3  0.006583755
## 39     5     7 5 2  0.096707622
## 40     5     8 5 1 -0.113493661
## 41     6     1 6 8  0.000000000
## 42     6     2 6 7 -0.251867731
## 43     6     3 6 6  0.261963921
## 44     6     4 6 5 -0.201709450
## 45     6     5 6 4  0.006584715
## 46     6     6 6 3  0.858339787
## 47     6     7 6 2  0.000000000
## 48     6     8 6 1 -0.118739286
## 49     7     1 7 8  0.085936799
## 50     7     2 7 7  0.000000000
## 51     7     3 7 6  0.301287385
## 52     7     4 7 5  0.000000000
## 53     7     5 7 4  0.096705497
## 54     7     6 7 3  0.000000000
## 55     7     7 7 2  0.704015675
## 56     7     8 7 1  0.000000000
## 57     8     1 8 8  0.000000000
## 58     8     2 8 7 -0.100188418
## 59     8     3 8 6 -0.017207855
## 60     8     4 8 5  0.000000000
## 61     8     5 8 4 -0.113493298
## 62     8     6 8 3 -0.118738875
## 63     8     7 8 2  0.000000000
## 64     8     8 8 1  0.552722736
##
## [[2]]$arg
## [[2]]$arg$type
## [1] "full"
##
##
##
```

```
## [[3]]
## [[3]]$corr
##   1 2 3 4 5 6 7 8
## 1 1 0 0 0 0 0 0 0
## 2 0 1 0 0 0 0 0 0
## 3 0 0 1 0 0 0 0 0
## 4 0 0 0 1 0 0 0 0
## 5 0 0 0 0 1 0 0 0
## 6 0 0 0 0 0 1 0 0
## 7 0 0 0 0 0 0 1 0
## 8 0 0 0 0 0 0 0 1
##
## [[3]]$corrPos
##    xName yName x y corr
## 1      1     1 1 8    1
## 2      1     2 1 7    0
## 3      1     3 1 6    0
## 4      1     4 1 5    0
## 5      1     5 1 4    0
## 6      1     6 1 3    0
## 7      1     7 1 2    0
## 8      1     8 1 1    0
## 9      2     1 2 8    0
## 10     2     2 2 7    1
## 11     2     3 2 6    0
## 12     2     4 2 5    0
## 13     2     5 2 4    0
## 14     2     6 2 3    0
## 15     2     7 2 2    0
## 16     2     8 2 1    0
## 17     3     1 3 8    0
## 18     3     2 3 7    0
## 19     3     3 3 6    1
## 20     3     4 3 5    0
## 21     3     5 3 4    0
## 22     3     6 3 3    0
## 23     3     7 3 2    0
## 24     3     8 3 1    0
## 25     4     1 4 8    0
## 26     4     2 4 7    0
## 27     4     3 4 6    0
## 28     4     4 4 5    1
## 29     4     5 4 4    0
## 30     4     6 4 3    0
## 31     4     7 4 2    0
## 32     4     8 4 1    0
## 33     5     1 5 8    0
## 34     5     2 5 7    0
## 35     5     3 5 6    0
## 36     5     4 5 5    0
## 37     5     5 5 4    1
## 38     5     6 5 3    0
## 39     5     7 5 2    0
## 40     5     8 5 1    0
## 41     6     1 6 8    0
## 42     6     2 6 7    0
## 43     6     3 6 6    0
## 44     6     4 6 5    0
## 45     6     5 6 4    0
## 46     6     6 6 3    1
## 47     6     7 6 2    0
## 48     6     8 6 1    0
## 49     7     1 7 8    0
## 50     7     2 7 7    0
## 51     7     3 7 6    0
## 52     7     4 7 5    0
## 53     7     5 7 4    0
## 54     7     6 7 3    0
## 55     7     7 7 2    1
## 56     7     8 7 1    0
## 57     8     1 8 8    0
## 58     8     2 8 7    0
## 59     8     3 8 6    0
## 60     8     4 8 5    0
## 61     8     5 8 4    0
## 62     8     6 8 3    0
## 63     8     7 8 2    0
## 64     8     8 8 1    1
## 
## [[3]]$arg
## [[3]]$arg$type
## [1] "full"
## 
## 
## 
## [[4]]
## [[4]]$corr
##   1 2 3 4 5 6 7 8
## 1 1 0 0 0 0 0 0 0
## 2 0 1 0 0 0 0 0 0
## 3 0 0 1 0 0 0 0 0
## 4 0 0 0 1 0 0 0 0
## 5 0 0 0 0 1 0 0 0
## 6 0 0 0 0 0 1 0 0
## 7 0 0 0 0 0 0 1 0
## 8 0 0 0 0 0 0 0 1
## 
## [[4]]$corrPos
##    xName yName x y corr
## 1      1     1 1 8    1
## 2      1     2 1 7    0
## 3      1     3 1 6    0
## 4      1     4 1 5    0
## 5      1     5 1 4    0
## 6      1     6 1 3    0
## 7      1     7 1 2    0
## 8      1     8 1 1    0
## 9      2     1 2 8    0
## 10     2     2 2 7    1
## 11     2     3 2 6    0
## 12     2     4 2 5    0
## 13     2     5 2 4    0
## 14     2     6 2 3    0
## 15     2     7 2 2    0
## 16     2     8 2 1    0
## 17     3     1 3 8    0
## 18     3     2 3 7    0
## 19     3     3 3 6    1
## 20     3     4 3 5    0
## 21     3     5 3 4    0
## 22     3     6 3 3    0
## 23     3     7 3 2    0
## 24     3     8 3 1    0
## 25     4     1 4 8    0
## 26     4     2 4 7    0
## 27     4     3 4 6    0
## 28     4     4 4 5    1
## 29     4     5 4 4    0
## 30     4     6 4 3    0
## 31     4     7 4 2    0
## 32     4     8 4 1    0
## 33     5     1 5 8    0
## 34     5     2 5 7    0
## 35     5     3 5 6    0
## 36     5     4 5 5    0
## 37     5     5 5 4    1
## 38     5     6 5 3    0
## 39     5     7 5 2    0
## 40     5     8 5 1    0
## 41     6     1 6 8    0
## 42     6     2 6 7    0
## 43     6     3 6 6    0
## 44     6     4 6 5    0
## 45     6     5 6 4    0
## 46     6     6 6 3    1
## 47     6     7 6 2    0
## 48     6     8 6 1    0
## 49     7     1 7 8    0
## 50     7     2 7 7    0
## 51     7     3 7 6    0
## 52     7     4 7 5    0
## 53     7     5 7 4    0
## 54     7     6 7 3    0
## 55     7     7 7 2    1
## 56     7     8 7 1    0
## 57     8     1 8 8    0
## 58     8     2 8 7    0
## 59     8     3 8 6    0
## 60     8     4 8 5    0
## 61     8     5 8 4    0
## 62     8     6 8 3    0
## 63     8     7 8 2    0
## 64     8     8 8 1    1
##
## [[4]]$arg
## [[4]]$arg$type
## [1] "full"
```

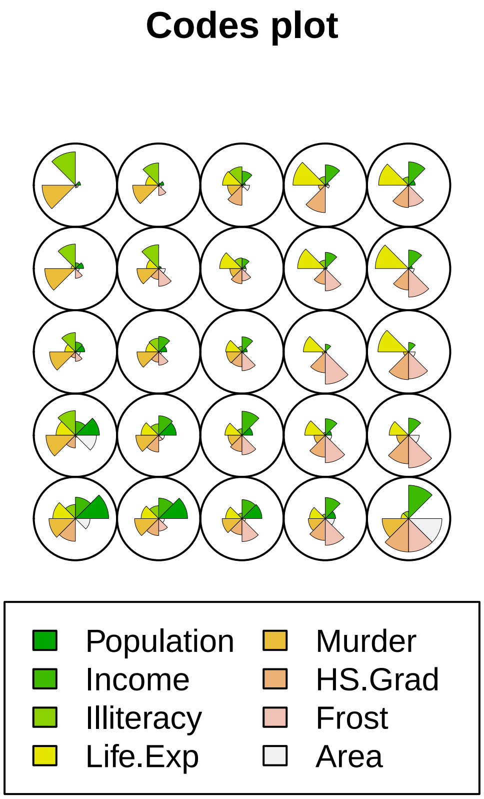
```r
non_zr_counts <- lapply(res, function(model) sum(model$wi != 0, na.rm = TRUE))
print("Non-zero counts in precision matrices for each penalty:")
```

```
## [1] "Non-zero counts in precision matrices for each penalty:"
```

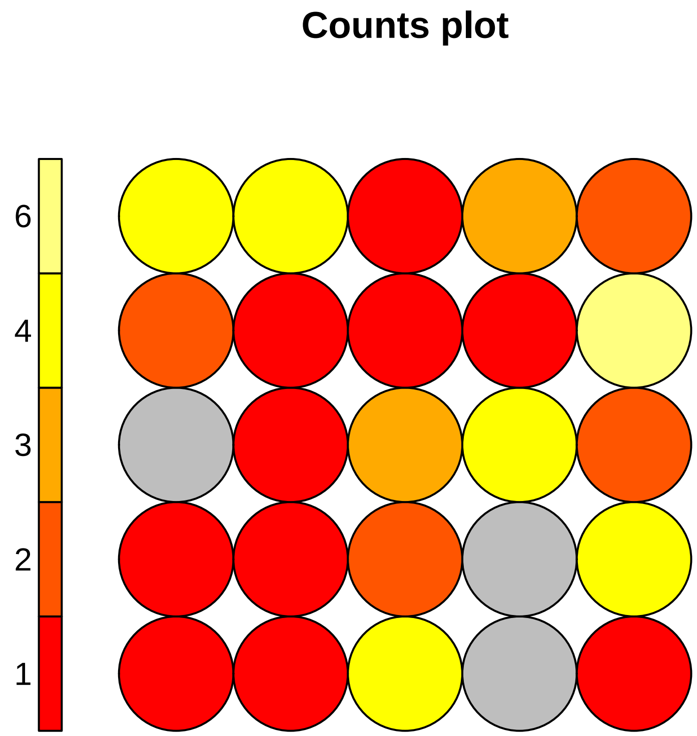
```r
print(non_zr_counts)
```

```
## [[1]]
## [1] 58
##
## [[2]]
## [1] 44
##
## [[3]]
## [1] 8
##
## [[4]]
## [1] 8
```

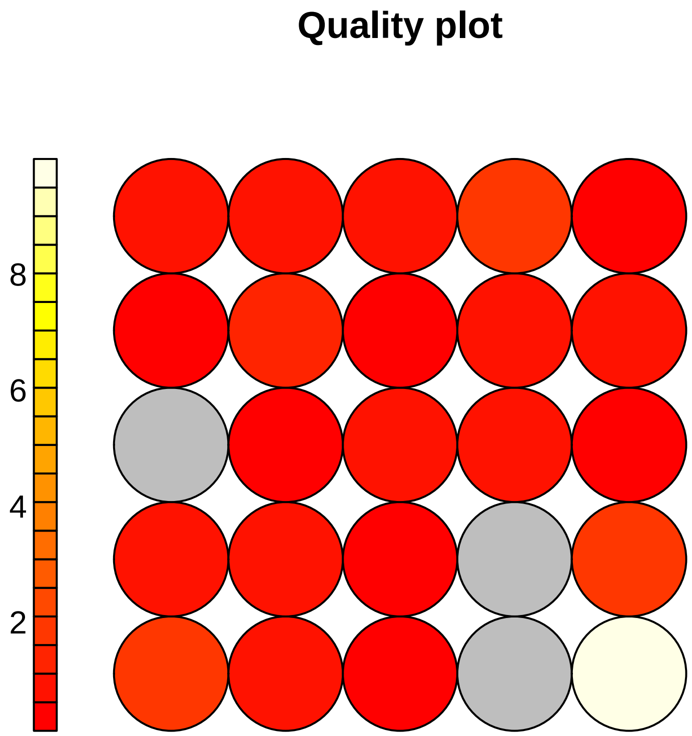
```r
set.seed(43)
grid <- somgrid(xdim = 5, ydim = 5, topo = "rectangular")
som_model <- som(predictors, grid = grid, rlen = 120)


plot(som_model, type = "codes")
```



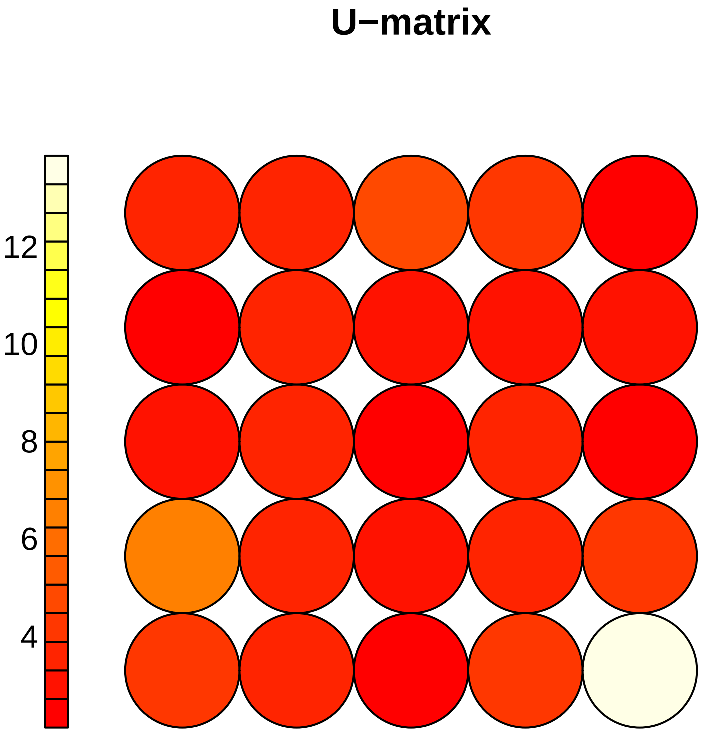
```
plot(som_model, type = "count")
```



```r
plot(som_model, type = "quality")
```



```r
plot(som_model, type = "dist.neighbours", main = "U-matrix")
```



GLASSO :

- **Lower Penalties (0.01 and 0.1):** These models produce precision matrices with more non-zero entries (58 and 44, respectively), indicating a larger number of conditional dependencies between variables. This indicates that at lower penalties, the model considers many variables as interconnected.
- **Higher Penalties (1 and 10):** With only 8 non-zero entries in the precision matrices, the model shows high sparsity, indicating that most variables are conditionally independent except for a few strong dependencies.

SOM:

- The SOM displays clusters of variables across different nodes showing how variables group based on similarities in their data patterns across states. This method captures relationships and can highlight similarities by non-linear interactions and complex patterns not detected by the Graphical Lasso.

Complementary :

- **GLASSO**: The sparse structure in GLASSO particularly at higher penalties identifies the strongest and most direct relationships between variables. For example, if Income strongly related with Life Expectancy in a sparse GLASSO model, it suggests a direct dependency.
- **SOM**: The SOM on the other hand provides clustering states based on overall similarity across multiple metrics. It shows how states group based on similar characteristics.

Contradictory :

- **Dependencies:** If variables are close in the SOM but appear independent in high-penalty Graphical Lasso settings this indicate that their proximity in the SOM is driven by external factors that do not constitute a direct dependency.

Consider the "cad1" data set in the package gRbase. These observations are from individuals in the Danish Heart Clinic.
(a) Learn a Bayesian Network using a structural learning knowledge, and prior knowledge obtained through the definitions of the variables in the help files. You do not have to use all of the variables. Make sure to detail your network construction process.
(b) Construct the above network in R, and infer the Conditional Probability Tables using the cad1 data. Identify any d-separations in the graph.
(c) Suppose it is known that a new observation is female with Hypercholesterolemia (high cholesterol). Absorb this evidence into the graph and revise the probabilities. How does the probability of heart-failure and coronary artery disease (CAD) change after this information is considered?

```r
library(gRbase)
```

```
##
## Attaching package: 'gRbase'

## The following objects are masked from 'package:bnlearn':
##
##     ancestors, children, nodes, parents

## The following objects are masked from 'package:igraph':
##
##     edges, is_dag, topo_sort
```

```r
library(bnlearn)

data(cad1, package = "gRbase")
model <- model2network("[Sex][Hyperchol|Sex][CAD|Hyperchol][Heartfail|CAD:Hyperchol]")
fitted_model<- bn.fit(model, data = cad1[, c("Sex", "Hyperchol", "CAD", "Heartfail")])

print("Fitted Bayesian Network Structure:")
```

```
## [1] "Fitted Bayesian Network Structure:"
```

```r
print(model)
```

```
##
##   Random/Generated Bayesian network
##
##   model:
##    [Sex][Hyperchol|Sex][CAD|Hyperchol][Heartfail|CAD:Hyperchol]
##   nodes:                                  4
##   arcs:                                   4
##     undirected arcs:                       0
##     directed arcs:                         4
##   average markov blanket size:             2.00
##   average neighbourhood size:              2.00
##   average branching factor:                1.00
##
##   generation algorithm:                    Empty
```

```
print("Conditional Probability Tables for each node:")
```

```
## [1] "Conditional Probability Tables for each node:"
```

```
print(fitted_model$Sex)
```

```
##
##   Parameters of node Sex (multinomial distribution)
##
## Conditional probability table:
##     Female      Male
## 0.1991525 0.8008475
```

```
print(fitted_model$Hyperchol)
```

```
##
##   Parameters of node Hyperchol (multinomial distribution)
##
## Conditional probability table:
##
##          Sex
## Hyperchol    Female      Male
##       No  0.5319149 0.4391534
##       Yes 0.4680851 0.5608466
```

```
print(fitted_model$CAD)
```

```
##
##   Parameters of node CAD (multinomial distribution)
##
## Conditional probability table:
##
##      Hyperchol
## CAD      No   Yes
##   No  0.750 0.375
##   Yes 0.250 0.625
```

```
print(fitted_model$Heartfail)
```

```
##
##   Parameters of node Heartfail (multinomial distribution)
##
```

```
## Conditional probability table:
##
## , , Hyperchol = No
##
##          CAD
## Heartfail        No       Yes
##       No  0.6296296 0.8518519
##       Yes 0.3703704 0.1481481
##
## , , Hyperchol = Yes
##
##          CAD
## Heartfail        No       Yes
##       No  0.7083333 0.8625000
##       Yes 0.2916667 0.1375000
```

```
print("D-separation between 'Heartfail' and 'Sex' given 'CAD' and 'Hyperchol':")
```

```
## [1] "D-separation between ’Heartfail’ and ’Sex’ given ’CAD’ and ’Hyperchol’:"
```

```
print(dsep(fitted_model, "Heartfail", "Sex", c("CAD", "Hyperchol")))
```

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

```
data <- rbn(fitted_model, n = 10000)

data_with_evidence <- subset(data, Sex == "Female" & Hyperchol == "Yes")
pb_cad_yes <- mean(data_with_evidence$CAD == "Yes")
pb_heartfail_yes <- mean(data_with_evidence$Heartfail == "Yes")

print(paste(" P(CAD='Yes' | Sex='Female', Hyperchol='Yes'):", pb_cad_yes))
```

```
## [1] " P(CAD=’Yes’ | Sex=’Female’, Hyperchol=’Yes’): 0.642933049946865"
```

```
print(paste("Simulated P(Heartfail='Yes' | Sex='Female', Hyperchol='Yes'):", pb_heartfail_yes))
```

```
## [1] "Simulated P(Heartfail=’Yes’ | Sex=’Female’, Hyperchol=’Yes’): 0.188097768331562"
```

Based on the above Bayesian Network and its Conditional Probability Tables reveal that the probability of Coronary Artery Disease (CAD) for a female with Hypercholesterolemia is approximately 64.29%, indicating a strong link between Hypercholesterolemia and CAD. Heart Failure’s probability under the same conditions is around 18.81%, showing a medium but significant risk increase due to CAD influenced by Hypercholesterolemia.

This analysis highlights the chain effect of risk factors where Hypercholesterolemia increases the risk of CAD, which increases the likelihood of Heart Failure. The d-separation confirms that the sex of the individual is not useful and not beneficial about Heart Failure risk when CAD and Hypercholesterolemia statuses are known.