
Answers part 2

(a) 126758

(b) 9*2*2*2*3*6*4*3*5*2=155520 number of cells
    8*1*1*1*2*5*3*2*4*1=1920 number of parameters

(c)

```
> gm.restart(nstart=1, prob=0, seed=2, table(rhc.dat), forward=T,
backward=T, score="bic")

$model
$model[[1]]
[1] 4 6

$model[[2]]
[1] 9 6

$model[[3]]
[1] 9 2

$model[[4]]
[1] 2 8

$model[[5]]
[1] 8 1

$model[[6]]
[1] 5 6

$model[[7]]
[1] 6 7

$model[[8]]
[1] 10  1  3


$score
[1] 15990.19
```

Independance graph is in graph-c.pdf.

(d)

gender ⊥ income | ninsclas

Which means that gender and income are independent given ninclas. Which means that when you pick some value for ninclas then gender and income are independent. Some other formulation is that

gender and income looks related but that the reason is ninclas.
If you are interested in predicting whether or not someone survives, you must look at the direct dependencies of the variable 'death', which means that it depends on cancer status (ca) and age.

(e)

```
> gm.restart(nstart=1, prob=1, seed=2, table(rhc.dat), forward=T,
backward=T, score="bic")

$search$model[[1]]
[1] 7 6

$search$model[[2]]
[1] 4 3

$search$model[[3]]
[1]  3  2 10

$search$model[[4]]
[1] 3 2 9

$search$model[[5]]
[1] 3 2 8

$search$model[[6]]
[1] 3 1 8

$search$model[[7]]
[1]  5  2 10

$search$model[[8]]
[1] 5 2 6


$search$score
[1] 19054.53
```

Results:
There are more dependencies between the different variables. This will give a better model with better score.

Independance graph is in graph-e.pdf.

(f)

```
> gm.restart(nstart=1, prob=0, seed=2, table(rhc.dat), forward=T,
backward=T, score="aic")

$search
$search$model
```

```
$search$model[[1]]
[1] 10  1  2

$search$model[[2]]
[1] 10  1  3

$search$model[[3]]
[1] 2 1 8

$search$model[[4]]
[1] 2 1 9

$search$model[[5]]
[1] 4 9 1

$search$model[[6]]
[1] 4 9 6

$search$model[[7]]
[1] 5 6 7

$search$model[[8]]
[1] 5 6 9


$search$score
[1] 14472.39

$search$trace
      forward start end    score
 [1,]       1     9   6 19813.67
 [2,]       1     8   1 18128.65
 [3,]       1     7   6 16781.55
 [4,]       1     3   1 16361.47
 [5,]       1     6   5 16055.14
 [6,]       1     9   1 15769.28
 [7,]       1     8   2 15515.52
 [8,]       1    10   3 15269.86
 [9,]       1    10   1 15139.77
[10,]       1     2   1 15011.42
[11,]       1     9   2 14771.25
[12,]       1     9   5 14678.39
[13,]       1     7   5 14602.40
[14,]       1     6   4 14558.35
[15,]       1    10   2 14518.42
[16,]       1     9   4 14483.99
[17,]       1     4   1 14472.39

$search$call
gm.search(table = table, adjm = best.neighbor$adjm, forward =
forward,
    backward = backward, score = score)

$call
```

```
gm.restart(nstart = 1, prob = 0, seed = 2, table = table(rhc.dat),
    forward = T, backward = T, score = "aic")
```

```
> gm.restart(nstart=1, prob=1, seed=2, table(rhc.dat), forward=T,
backward=T, score="aic")

$search
$search$model
$search$model[[1]]
[1]  5 10  7

$search$model[[2]]
[1]  3  1 10

$search$model[[3]]
[1] 4 2 9 6

$search$model[[4]]
[1] 4 2 7 6

$search$model[[5]]
[1]  4  2  7 10

$search$model[[6]]
[1]  4  2  8 10

$search$model[[7]]
[1]  8  1  2 10


$search$score
[1] 15142.41

$search$trace
      forward start end    score
 [1,]       0     6   1 84115.56
 [2,]       0     9   1 50204.87
 [3,]       0     9   7 31217.39
 [4,]       0     7   1 25432.24
 [5,]       0     9   8 21713.07
 [6,]       0     8   6 18696.14
 [7,]       0     5   1 17597.67
 [8,]       0     9   5 16838.52
 [9,]       0     6   5 16280.73
[10,]       0     8   5 15979.36
[11,]       0    10   9 15800.04
[12,]       0    10   6 15634.05
[13,]       0     4   1 15483.13
[14,]       0     9   3 15421.26
[15,]       0     5   4 15361.84
[16,]       0     6   3 15310.53
[17,]       0     8   7 15278.97
```

```
[18,]       0     5   3 15257.17
[19,]       0     5   2 15238.89
[20,]       0     7   3 15236.75
[21,]       0     4   3 15229.19
[22,]       0     8   3 15147.28
[23,]       0     3   2 15142.41

$search$call
gm.search(table = table, adjm = best.neighbor$adjm, forward =
forward,
    backward = backward, score = score)

$call
gm.restart(nstart = 1, prob = 1, seed = 2, table = table(rhc.dat),
    forward = T, backward = T, score = "aic")
```

The models are not the same.

(g)

AIC(M) = dev(M) + 2 × dim(M),
BIC(M) = dev(M) + ln N × dim(M).

N = total number observations = ln(5735) = 8.654

Since 8.65 > 2 the BIC scoring formula favours models with fewer parameters more than the AIC formula. This results in the BIC formula finding a less complex formula than the AIC.

(h) We have tried several combinations of input parameters, but we did not obtain better results than already given above. We noticed that around the probability of 0.5 there were no neighbor models which satisfy the chordal constraint.