Probabilistic Modeling Assignment 2

Shweta Singhal

March 11, 2016

```r
## Function to create a conditional probability table
## Conditional probability is of the form p(x1 | x2, ..., xk)
## varnames: vector of variable names (strings)
## -- NOTE: first variable listed will be x1, remainder will be parents, x2, ..., xk
## probs: vector of probabilities for the flattened probability table
## levelsList: a list containing a vector of levels (outcomes) for each variable
## See the BayesNetExamples.r file for examples of how this function works
createCPT = function(varnames, probs, levelsList){
  if(length(probs) != prod(sapply(levelsList, FUN=length)))
    return(NULL)
  m = length(probs)
  n = length(varnames)
  g = matrix(0, m, n)
  g = as.data.frame(g)
  names(g) = varnames
  k = 1
  for(i in n:1){
    levs = levelsList[[i]]
    g[,i] = rep(levs, each = k, times = m / (k * length(levs)))
    k = k * length(levs)
  }
  return(data.frame(probs = probs, g))
}

## Build a CPT from a data frame
## Constructs a conditional probability table as above, but uses frequencies
## from a data frame of data to generate the probabilities.
createCPT.fromData = function(x, varnames){
  levelsList = list()
  for(i in 1:length(varnames)){
    name = varnames[i]
    levelsList[[i]] = sort(unique(x[,name]))
  }
  m = prod(sapply(levelsList, FUN=length))
  n = length(varnames)
  g = matrix(0, m, n)
  g = as.data.frame(g)
  names(g) = varnames
  k = 1
  for(i in n:1){
    levs = levelsList[[i]]
    g[,i] = rep(levs, each = k, times = m / (k * length(levs)))
    k = k * length(levs)
  }
  probs = numeric(m)
  numLevels = length(levelsList[[1]])
  skip = m / numLevels
  fact = numeric(ncol(x))
  lastfact = 1
  for(i in length(varnames):1){
    j = which(names(x) == varnames[i])
    fact[j] = lastfact
    lastfact = lastfact * length(levelsList[[i]])
  }
  a = as.matrix(x - 1) %*% fact + 1
  for(i in 1:m)
    probs[i] = sum(a == i)
  for(i in 1:skip){
    denom = 0 ## This is the normalization
    for(j in seq(i, m, skip))
      denom = denom + probs[j]
    for(j in seq(i, m, skip)){
      if(denom != 0)
        probs[j] = probs[j] / denom
    }
  }
  return(data.frame(probs = probs, g))
}

##Risk Net
createRiskNet = function(){
  y = read.csv("./RiskFactors.csv")
  var = c("income")
  A = createCPT.fromData(y,var)
  var =  c("smoke","income")
  B = createCPT.fromData(y,var)
  var = c("cholesterol", "income","smoke","exercise")
  C = createCPT.fromData(y,var)
  var = c("bp", "income","smoke","exercise")
  D = createCPT.fromData(y,var)
  var = c("exercise", "income")
  E = createCPT.fromData(y,var)
  var = c("bmi", "income","exercise")
  F = createCPT.fromData(y,var)
  var = c("diabetes", "bmi")
  G = createCPT.fromData(y,var)
  var = c("stroke", "bmi","bp","cholesterol")
  H = createCPT.fromData(y,var)
  var = c("attack", "bmi","bp","cholesterol")
  I = createCPT.fromData(y,var)
  var = c("angina", "bmi","bp","cholesterol")
  J = createCPT.fromData(y,var)
  riskNet = list("a" = A, "b" = B, "c" = C, "d" = D, "e" = E, "f" = F, "g" = G, "h" = H, "i" = I, "j" =
  return(riskNet)
}

##Risk Net for Ques 4)
createRiskNet1 = function(){
  y = read.csv("./RiskFactors.csv")
  var = c("income")
  A = createCPT.fromData(y,var)
  var =  c("smoke","income")
  B = createCPT.fromData(y,var)
  var = c("cholesterol", "income","smoke","exercise")
  C = createCPT.fromData(y,var)
  var = c("bp", "income","smoke","exercise")
  D = createCPT.fromData(y,var)
  var = c("exercise", "income")
  E = createCPT.fromData(y,var)
  var = c("bmi", "income","exercise")
  F = createCPT.fromData(y,var)
  var = c("diabetes", "bmi","smoke","exercise")
  G = createCPT.fromData(y,var)
  var = c("stroke", "bmi","bp","cholesterol","smoke","exercise")
  H = createCPT.fromData(y,var)
  var = c("attack", "bmi","bp","cholesterol","smoke","exercise")
  I = createCPT.fromData(y,var)
  var = c("angina", "bmi","bp","cholesterol","smoke","exercise")
  J = createCPT.fromData(y,var)
  riskNet = list("1" = A, "2" = B, "3" = C, "4" = D, "5" = E, "6" = F, "7" = G, "8" = H, "9" = I, "10" =
  return(riskNet)
}

##Risk Net for Ques 5)
createRiskNet2 = function(){
  y = read.csv("./RiskFactors.csv")
  var = c("income")
  A = createCPT.fromData(y,var)
  var =  c("smoke","income")
  B = createCPT.fromData(y,var)
  var = c("cholesterol", "income","smoke","exercise")
  C = createCPT.fromData(y,var)
  var = c("bp", "income","smoke","exercise")
  D = createCPT.fromData(y,var)
  var = c("exercise", "income")
  E = createCPT.fromData(y,var)
  var = c("bmi", "income","exercise")
  F = createCPT.fromData(y,var)
  var = c("diabetes", "bmi","smoke","exercise")
  G = createCPT.fromData(y,var)
  var = c("stroke", "bmi","bp","cholesterol","smoke","exercise","diabetes")
  H = createCPT.fromData(y,var)
  var = c("attack", "bmi","bp","cholesterol","smoke","exercise")
  I = createCPT.fromData(y,var)
  var = c("angina", "bmi","bp","cholesterol","smoke","exercise")
  J = createCPT.fromData(y,var)
  riskNet = list("1" = A, "2" = B, "3" = C, "4" = D, "5" = E, "6" = F, "7" = G, "8" = H, "9" = I, "10" =
  return(riskNet)
}

## Product of two factors
## A, B: two factor tables
##
```

```
## Should return a factor table that is the product of A and B.
## You can assume that the product of A and B is a valid operation.
productFactor = function(A, B){
  x = names(A)
  y = names(B)
  x = x[-which(x=="probs")]
  y = y[-which(y=="probs")]
  AB = merge(A, B, by=intersect(x, y))
  AB = transform(AB, probs = probs.x * probs.y)
  AB = subset(AB, select = -c(probs.x,probs.y))
  return(AB)
}

## Marginalize a variable from a factor
## A: a factor table
## margVar: a string of the variable name to marginalize
##
## Should return a factor table that marginalizes margVar out of A.
## You can assume that margVar is on the left side of the conditional.
marginalizeFactor = function(X, margVar){
  name = names(X)
  name = name[-which(name==margVar)]
  nameNew = name[-which(name=="probs")]
  margeNet = X
  if(length(nameNew) > 0){
    rows = subset(X,select = nameNew)
    margeNet <- aggregate(X$probs, by=rows,FUN=sum)
    nameSet <- union(nameNew,"probs")
    colnames(margeNet) <- nameSet
  }
  return(margeNet)
}

marginal_1 <- function(bayesNet, var){
  marginalizeBayesNet <- {}
  i <- 0
  factorlist<- {}
  lapply(bayesNet,function(x){
    intersectCol <- intersect(colnames(x), var)
    i <<- i +1
    if(length(intersectCol)>0){
      factorlist <<- append(factorlist, list(x))
      bayesNet <<- bayesNet[-i]
      i <<- i-1
    }
  })
  if(length(factorlist) > 0){
    marginalizeBayesNet<- factorlist[[1]]
    if(length(factorlist) > 1){
      for(j in 2:length(factorlist)){
        marginalizeBayesNet <- productFactor(factorlist[[j-1]], factorlist[[j]])
        factorlist[[j]] <- marginalizeBayesNet
      }
    }
    marginalizeBayesNet <- marginalizeFactor(marginalizeBayesNet, var)
    bayesNet <- append(bayesNet, list(marginalizeBayesNet))
  }
  return(bayesNet)
}

## Marginalize a list of variables
## bayesnet: a list of factor tables
## margVars: a vector of variable names (as strings) to be marginalized
##
## Should return a Bayesian network (list of factor tables) that results
## when the list of variables in margVars is marginalized out of bayesnet.
marginalize = function(bayesnet, margVars){
  n = length(margVars)
  for(i in 1:n){
    bayesnet <- marginal_1(bayesnet, margVars[i])
  }
  return(bayesnet)
}

## Observe values for a set of variables
## bayesnet: a list of factor tables
## obsVars: a vector of variable names (as strings) to be observed
## obsVals: a vector of values for corresponding variables (in the same order)
##
## Set the values of the observed variables. Other values for the variables
## should be removed from the tables. You do not need to normalize the factors
## to be probability mass functions.
observe = function(bayesnet, obsVars, obsVals){
  return(lapply(bayesnet, function(x){
    listVars <- intersect(colnames(x), obsVars)
    locations <- match(listVars, obsVars)
    listVals <- obsVals[locations]
    tempNet <- x
    n <- length(listVars)
    if(n > 0){
      for(i in 1:n){
        tempNet <- tempNet[tempNet[, listVars[i]] == listVals[i],]
      }
    }
    tempNet
  }))
}

## Run inference on a Bayesian network
## bayesnet: a list of factor tables
## margVars: a vector of variable names to marginalize
## obsVars: a vector of variable names to observe
## obsVals: a vector of values for corresponding variables (in the same order)
##
## This function should run marginalization and observation of the sets of
## variables. In the end, it should return a single joint probability table. The
## variables that are marginalized should not appear in the table. The variables
## that are observed should appear in the table, but only with the single
## observed value. The variables that are not marginalized or observed should
## appear in the table with all of their possible values. The probabilities
## should be normalized to sum to one.
infer = function(bayesnet, margVars, obsVars, obsVals){
  bayesnet <- observe(bayesnet,obsVars, obsVals)
  bayesnet <- marginalize(bayesnet, margVars)
  if(length(bayesnet) > 1){
    for(i in 2:length(bayesnet)){
      productNet <- productFactor(bayesnet[[i-1]], bayesnet[[i]])
      bayesnet[[i]] <- productNet
    }
  }
  productNet <- bayesnet[[length(bayesnet)]]
  sum <- sum(productNet$probs)
  productNet$probs = productNet$probs/sum
  return(productNet)
}

varNames = c("income", "exercise", "smoke", "bmi", "bp", "cholesterol","angina", "stroke", "attack", "d

riskNet <- createRiskNet()
```

Ans 1.)

```
sum = 0
for(i in 1:10){
  sum = nrow(riskNet[[i]])+sum
}
print(sum)
```

```
## [1] 504
```

```
#Ans 1.b)
size <- nrow(infer(riskNet, NULL, NULL, NULL))
print(size)
```

```
## [1] 32768
```

Ans 2.)

```
#Ans 2.a)
infer1 <- infer(riskNet, setdiff(varNames, c("diabetes","smoke","exercise")), c("smoke","exercise"), c(
infer1 <- infer1[!(names(infer1) %in% c("smoke", "exercise"))]

infer2 <- infer(riskNet, setdiff(varNames, c("diabetes","smoke","exercise")), c("smoke","exercise"), c(2
infer2 <- infer2[!(names(infer2) %in% c("smoke", "exercise"))]

infer3 <- infer(riskNet, setdiff(varNames, c("stroke","smoke","exercise")), c("smoke","exercise"), c(1,2
infer3 <- infer3[!(names(infer3) %in% c("smoke", "exercise"))]
```

```
infer4 <- infer(riskNet, setdiff(varNames, c("stroke","smoke","exercise")), c("smoke","exercise"), c(2,
infer4 <- infer4[!(names(infer4) %in% c("smoke", "exercise"))]

infer5 <- infer(riskNet, setdiff(varNames, c("attack","smoke","exercise")), c("smoke","exercise"), c(1,
infer5 <- infer5[!(names(infer5) %in% c("smoke", "exercise"))]

infer6 <- infer(riskNet, setdiff(varNames, c("attack", "smoke","exercise")), c("smoke","exercise"), c(2
infer6 <- infer6[!(names(infer6) %in% c("smoke", "exercise"))]

infer7 <- infer(riskNet, setdiff(varNames, c("angina", "smoke","exercise")), c("smoke","exercise"), c(1
infer7 <- infer7[!(names(infer7) %in% c("smoke", "exercise"))]

infer8 <- infer(riskNet, setdiff(varNames, c("angina", "smoke","exercise")), c("smoke","exercise"), c(2
infer8 <- infer8[!(names(infer8) %in% c("smoke", "exercise"))]
```

Diabetes with Bad Habit:

diabetes	probs
1	0.1557
2	0.00878
3	0.81569
4	0.01982

Diabetes with Good Habit:

	diabetes	probs
1	1	0.13179
2	2	0.00866
3	3	0.84418
4	4	0.01774

Stroke in Bad Habit:

	stroke	probs
1	1	0.05095
2	2	0.9491

Stroke in Good Health:

	stroke	probs
1	1	0.0372
2	2	0.9628

Attack in Bad Habit:

	attack	probs
1	1	0.0756
2	2	0.9244

Attack in Good Habit:

Angina in Bad Habit:

Angina in Good Habit:

Ans 2.b)

	attack	probs
1	1	0.0534
2	2	0.9466

	angina	probs
1	1	0.0801
2	2	0.9199

	angina	probs
1	1	0.0539
2	2	0.9461

```
##Ans.2.b)
infer9 <- infer(riskNet, setdiff(varNames, c("diabetes","bp","cholesterol","bmi")), c("bp","cholesterol
infer9 <- infer9[!(names(infer9) %in% c("bp","cholesterol","bmi"))]

infer10 <- infer(riskNet, setdiff(varNames, c("diabetes","bp","cholesterol","bmi")), c("bp","cholesterol
infer10 <- infer10[!(names(infer10) %in% c("bp","cholesterol","bmi"))]

infer11 <- infer(riskNet, setdiff(varNames, c("stroke","bp","cholesterol","bmi")), c("bp","cholesterol"
infer11 <- infer11[!(names(infer11) %in% c("bp","cholesterol","bmi"))]

infer12 <- infer(riskNet, setdiff(varNames, c("stroke","bp","cholesterol","bmi")), c("bp","cholesterol"
infer12 <- infer12[!(names(infer12) %in% c("bp","cholesterol","bmi"))]

infer13 <- infer(riskNet, setdiff(varNames, c("attack","bp","cholesterol","bmi")), c("bp","cholesterol"
infer13 <- infer13[!(names(infer13) %in% c("bp","cholesterol","bmi"))]

infer14 <- infer(riskNet, setdiff(varNames, c("attack","bp","cholesterol","bmi")), c("bp","cholesterol"
infer14 <- infer14[!(names(infer14) %in% c("bp","cholesterol","bmi"))]

infer15 <- infer(riskNet, setdiff(varNames, c("angina","bp","cholesterol","bmi")), c("bp","cholesterol"
infer15 <- infer15[!(names(infer15) %in% c("bp","cholesterol","bmi"))]

infer16 <- infer(riskNet, setdiff(varNames, c("angina","bp","cholesterol","bmi")), c("bp","cholesterol"
infer16 <- infer16[!(names(infer16) %in% c("bp","cholesterol","bmi"))]
```

```
#Queries to use in Ques 5)
infer33 <- infer(riskNet, setdiff(varNames, c("stroke","diabetes")), c("diabetes"), c(1))
infer33 <- infer33[!(names(infer33) %in% c("bp","cholesterol","bmi"))]

infer36 <- infer(riskNet, setdiff(varNames, c("stroke","diabetes")), c("diabetes"), c(3))
infer36 <- infer36[!(names(infer36) %in% c("bp","cholesterol","bmi"))]
```

Diabetes in poor Health:

	diabetes	probs
1	1	0.1199
2	2	0.00747
3	3	0.85491
4	4	0.01772

Diabetes in Good Health:

	diabetes	probs
1	1	0.06088
2	2	0.00931
3	3	0.91915
4	4	0.01066

Stroke in Poor Health:

	stroke	probs
1	1	0.0838
2	2	0.9162

Stroke in Good Health:

	stroke	probs
1	1	0.0138
2	2	0.9862

Attack in Poor Health:

	attack	probs
1	1	0.14
2	2	0.86

Attack in Good Health:

	attack	probs
1	1	0.0154
2	2	0.9846

Angina in Poor Health:

	angina	probs
1	1	0.155
2	2	0.845

Angina in Good Health:

	angina	probs
1	1	0.0128
2	2	0.9972

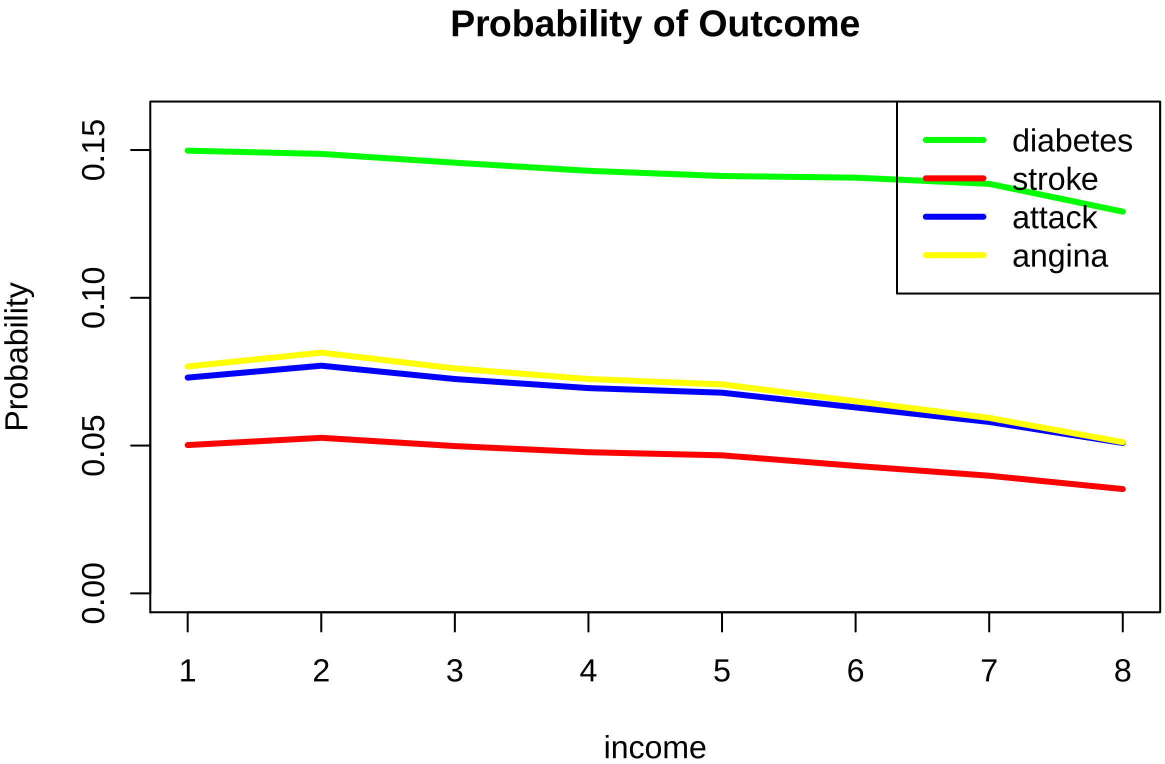
Ans.3)

```r
anginaList<-{}
diabetesList<-{}
strokeList<-{}
attackList<-{}
for(i in 1:8){
  temp <-infer(riskNet, setdiff(varNames, c("diabetes","income")), c("income"), c(i))
  temp <- temp[temp[, "diabetes"] == 1,"probs"]
  diabetesList <- append(diabetesList, list(temp))
  temp <-infer(riskNet, setdiff(varNames, c("stroke","income")), c("income"), c(i))
  temp <- temp[temp[, "stroke"] == 1,"probs"]
  strokeList <- append(strokeList, list(temp))

  temp <-infer(riskNet, setdiff(varNames, c("attack","income")), c("income"), c(i))
  temp <- temp[temp[, "attack"] == 1,"probs"]
  attackList <- append(attackList, list(temp))

  temp <-infer(riskNet, setdiff(varNames, c("angina","income")), c("income"), c(i))
  temp <- temp[temp[, "angina"] == 1,"probs"]
  anginaList <- append(anginaList, list(temp))
}

plot(c(1:8),diabetesList, type='l', ylim = c(0,0.16), lwd=3, col="green", main = "Probability of Outcom
lines(c(1:8),strokeList, col='red', lwd=3)
lines(c(1:8),attackList, col='blue', lwd=3)
lines(c(1:8),anginaList, col='yellow', lwd=3)
legend('topright', c("diabetes", "stroke","attack", "angina"), col = c("green", "red", "blue", "yellow")
```



From the graph, we can conclude that as the income increases the probability of having any disease decreases. Although the decreasing ratio with the income is small but its significant.

For Diabetes, as the income increases from 1 to 8, the chances of diabetes decreases from 0.15% to 0.129%. The relative difference of disease with income is significant.

For Stroke, as the income increases from 1 to 8, the chances of having stroke decreases from 0.0502% to 0.0353%. Which is also significant.

For Attack, as the income increases from 1 to 8, the chances of having stroke decreases from 0.073% to 0.0509%. Which is also significant.

For Angina, as the income increases from 1 to 8, the chances of having stroke decreases from 0.0768% to

0.0511%. Which is also significant.

Ans.4)

```
riskNet = createRiskNet1()

infer17 <- infer(riskNet, setdiff(varNames, c("diabetes","smoke","exercise")), c("smoke","exercise"), c
infer17 <- infer17[!(names(infer17) %in% c("smoke", "exercise"))]

infer18 <- infer(riskNet, setdiff(varNames, c("diabetes","smoke","exercise")), c("smoke","exercise"), c
infer18 <- infer18[!(names(infer18) %in% c("smoke", "exercise"))]

infer19 <- infer(riskNet, setdiff(varNames, c("stroke","smoke","exercise")), c("smoke","exercise"), c(1
infer19 <- infer19[!(names(infer19) %in% c("smoke", "exercise"))]

infer20 <- infer(riskNet, setdiff(varNames, c("stroke","smoke","exercise")), c("smoke","exercise"), c(2
infer20 <- infer20[!(names(infer20) %in% c("smoke", "exercise"))]

infer21 <- infer(riskNet, setdiff(varNames, c("attack","smoke","exercise")), c("smoke","exercise"), c(1
infer21 <- infer21[!(names(infer21) %in% c("smoke", "exercise"))]

infer22 <- infer(riskNet, setdiff(varNames, c("attack", "smoke","exercise")), c("smoke","exercise"), c(2
infer22 <- infer22[!(names(infer22) %in% c("smoke", "exercise"))]

infer23 <- infer(riskNet, setdiff(varNames, c("angina", "smoke","exercise")), c("smoke","exercise"), c(1
infer23 <- infer23[!(names(infer23) %in% c("smoke", "exercise"))]

infer24 <- infer(riskNet, setdiff(varNames, c("angina", "smoke","exercise")), c("smoke","exercise"), c(2
infer24 <- infer24[!(names(infer24) %in% c("smoke", "exercise"))]

infer25 <- infer(riskNet, setdiff(varNames, c("diabetes","bp","cholesterol","bmi")), c("bp","cholesterol
infer25 <- infer25[!(names(infer25) %in% c("bp","cholesterol","bmi"))]

infer26 <- infer(riskNet, setdiff(varNames, c("diabetes","bp","cholesterol","bmi")), c("bp","cholesterol
infer26 <- infer26[!(names(infer26) %in% c("bp","cholesterol","bmi"))]

infer27 <- infer(riskNet, setdiff(varNames, c("stroke","bp","cholesterol","bmi")), c("bp","cholesterol"
infer27 <- infer27[!(names(infer27) %in% c("bp","cholesterol","bmi"))]

infer28 <- infer(riskNet, setdiff(varNames, c("stroke","bp","cholesterol","bmi")), c("bp","cholesterol"
infer28 <- infer28[!(names(infer28) %in% c("bp","cholesterol","bmi"))]

infer29 <- infer(riskNet, setdiff(varNames, c("attack","bp","cholesterol","bmi")), c("bp","cholesterol"
infer29 <- infer29[!(names(infer29) %in% c("bp","cholesterol","bmi"))]

infer30 <- infer(riskNet, setdiff(varNames, c("attack","bp","cholesterol","bmi")), c("bp","cholesterol"
infer30 <- infer30[!(names(infer30) %in% c("bp","cholesterol","bmi"))]

infer31 <- infer(riskNet, setdiff(varNames, c("angina","bp","cholesterol","bmi")), c("bp","cholesterol"
infer31 <- infer31[!(names(infer31) %in% c("bp","cholesterol","bmi"))]

infer32 <- infer(riskNet, setdiff(varNames, c("angina","bp","cholesterol","bmi")), c("bp","cholesterol"
infer32 <- infer32[!(names(infer32) %in% c("bp","cholesterol","bmi"))]
```

Diabetes with bad habit:

	diabetes	probs
1	1	0.21919
2	2	0.00666
3	3	0.75116
4	4	0.02299

Diabetes with Good Habit:

	diabetes	probs
1	1	0.1004
2	2	0.00931
3	3	0.87509
4	4	0.01515

Stroke with Bad Habit:

	stroke	probs
1	1	0.0802
2	2	0.9198

Stroke in Good habit:

	stroke	probs
1	1	0.0256
2	2	0.9744

Attack in Bad Habit:

	attack	probs
1	1	0.121
2	2	0.879

Attack in Good Habit:

	attack	probs
1	1	0.0319
2	2	0.9681

Angina in Bad Habit:

	angina	probs
1	1	0.117
2	2	0.883

Angina in Good Habit:

	angina	probs
1	1	0.0357
2	2	0.9643

Diabetes in Poor Health:

Diabetes in Good Health:

	diabetes	probs
1	1	0.12882
2	2	0.00745
3	3	0.84529
4	4	0.01845

	diabetes	probs
1	1	0.05703
2	2	0.00949
3	3	0.92331
4	4	0.01017

	stroke	probs
1	1	0.0854
2	2	0.9146

Stroke in Poor Health:

Stroke in Good Health:

	stroke	probs
1	1	0.0132
2	2	0.9868

Attack in Poor Health:

	attack	probs
1	1	0.142
2	2	0.858

Attack in Good Health:

	attack	probs
1	1	0.0148
2	2	0.9852

Angina in Poor Health:

	angina	probs
1	1	0.157
2	2	0.843

Angina in Good Health:

	angina	probs
1	1	0.0126
2	2	0.9874

From the above table and the tables from the Question.2), by adding the edges from smoke and exercise to the diseases.

In case of diabetes, the probability of having diabetes with bad habit increases and good habit decreases. Hence our assumption was **not valid**.

In case of Stroke, the probability of having stroke with bad habit increases and in good habit decreases. Hence our assumption was **not valid**.

In case of Attack, the probability of having attack with bad habit increases and in good habit decreases. Hence our assumption was **not valid**.

In case of Angina, the probability of having angina with bad habit increases and in good habit decreases. Hence our assumption was **not valid**.

In case of diabetes, the probability of having diabetes with poor health increases and good health decreases. Hence our assumption was **not valid**.

In case of Stroke, the resultant probabilities are almost same. Hence our assumption was **valid**.

In case of Attack, there is a slight increase in probability of having attack considering smoking and exercise causes (edges) attack given bad health, and a slight decrease in probability of having attack given good health. Hence our assumption was **valid**.

In case of Angina, there is not much difference in the probabilities before and after adding the edge from smoke and excercise to the disease. Hence our assumption was **almost valid**.

Ans. 5)

```
##Ans.5)
riskNet2 = createRiskNet2()

infer34 <- infer(riskNet2, setdiff(varNames, c("stroke","diabetes")), c("diabetes"), c(1))
infer35 <- infer(riskNet2, setdiff(varNames, c("stroke","diabetes")), c("diabetes"), c(3))
```

Probability of having stroke give Diabetes is true with any edge from Diabetes to Stroke:

	diabetes	stroke	probs
1	1	1	0.0414
2	1	2	0.9586

Probability of having Stroke given Diabetes is true after adding an edge from Diabetes to Stroke :

$$P(Stroke = 1|Diabetes|1)$$

	diabetes	stroke	probs
1	1	1	0.077
2	1	2	0.923

Probability of having stroke give Diabetes is false without any edge from Diabetes to Stroke:

	diabetes	stroke	probs
1	3	1	0.0426
2	3	2	0.9574

Probability of having Stroke given Diabetes is false after adding an edge from Diabetes to Stroke :

$$P(Stroke = 1|Diabetes|3)$$

From the above tables we can conclude that, on considering diabetes as one of the causes of stroke in Bayesian network, the probability of having stroke given diabetes increases i.e if a person have diabetes there is greater possibility that the person have stroke too if diabetes is one of the causes of stroke. The absolute difference in probabilities is small, but relatively they are high. Hence here our assumption was **invalid**.

	diabetes	stroke	probs
1	3	1	0.0362
2	3	2	0.9638