Assignment 2

Yash Gangrade

March 4, 2018

Coding Part

The code with all the functions completed is shown below. Please see the 'BayesianNetworks-template.r' file to see this same code. I have completed the code completion for the different functions there itself.

BayesianNetworks-template.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)
{
  ## Check dimensions agree
  if(length(probs) != prod(sapply(levelsList, FUN=length)))
    return(NULL)

  ## Set up table with appropriate dimensions
  m = length(probs)
  n = length(varnames)
  g = matrix(0, m, n)

  ## Convert table to data frame (with column labels)
  g = as.data.frame(g)
  names(g) = varnames

  ## This for loop fills in the entries of the variable values
  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.
createCPTfromData = function(x, varnames)
```

```r
{
  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)

  ## Convert table to data frame (with column labels)
  g = as.data.frame(g)
  names(g) = varnames

  ## This for loop fills in the entries of the variable values
  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)
  }

  ## This is the conditional probability column
  probs = numeric(m)
  numLevels = length(levelsList[[1]])
  skip = m / numLevels

  ## This chunk of code creates the vector "fact" to index into probs using
  ## matrix multiplication with the data frame x
  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]])
  }
  ## Compute unnormalized counts of subjects that satisfy all conditions
  a = as.matrix(x - 1) %*% fact + 1
  for(i in 1:m)
    probs[i] = sum(a == i)

  ## Now normalize the conditional probabilities
  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))
```

```r
    {
      if(denom != 0)
        probs[j] = probs[j] / denom
    }
  }

  return(data.frame(probs = probs, g))
}

## 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)
{
  names_in_A = names(A);
  names_in_B = names(B);

  common_variable_list = intersect(names_in_A, names_in_B);
  common_variable_list = common_variable_list[common_variable_list != 'probs'];

  #inner join of the two tables
  new_table_merged = merge(A, B, by = common_variable_list);
  new_table_merged$probs <- new_table_merged$probs.x * new_table_merged$probs.y;
  ## Never use ->, made this mistake and spent an hour on debugging the code ##
  new_table_merged <- subset(new_table_merged, select = -c(probs.y, probs.x));
  # select = c(a, b) keeps the column a and b to make a table, on the other hand -c(a,b) will remove a
  return(new_table_merged)
}

## 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)
{
  p1 = setdiff(names(X), c(margVar));
  if(isTRUE(all.equal(p1, names(X)))){
    return (X);
  }
  variables = setdiff(names(X), c("probs", margVar));
  templist = list();
  for(i in 1:length(variables)){
    templist[[i]] = X[, variables[i]];
  }
  X = aggregate(X$probs, by = templist, FUN = "sum");
  variables[length(variables)+1] = "probs";
  names(X) = variables;
  return (X)
}
```

```r
## 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(bayesnet);
  if(n>1){
    temp_p = productFactor(bayesnet[[1]], bayesnet[[2]]);
  }
  for(i in 3:length(bayesnet)){
    temp_p = productFactor(temp_p, bayesnet[[i]]);
  }
  bayesnet_new = marginalizeFactor(temp_p, margVars);
  return(bayesnet_new)
}

## 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)
{
  n = length(bayesnet);
  for(i in 1:n){
    p1 = bayesnet[[i]];
    intersecting_variables = intersect(names(p1), obsVars);
    n2 = length(intersecting_variables);
    if(n2 != 0){
      for(j in 1:n2){
        p2 = p1[, intersecting_variables[j]];
        idx = match(intersecting_variables[j], obsVars);
        p1 = p1[p2 == obsVals[idx], ];
      }
    }
    bayesnet[[i]] = p1;
  }
  return(bayesnet)
}

## 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)
{
  observed = observe(bayesnet, obsVars, obsVals);
  marginalized = marginalize(observed, margVars);
  marginalized$probs = marginalized$probs / sum(marginalized$probs);
  return(marginalized);
}
```

Bayesian Network creation (From Diagram 1 used in Question 1). Fig: Bayesnet.r

The code for generating the Bayesian Network from the RiskFactors.csv data is as follows. The Bayesian Network created is printed at the end. It's for answering question 1,2, and 3. Please find the same code in Bayenet.r file.

```
source("BayesianNetworks-template.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)
file = read.csv("RiskFactors.csv", header = TRUE)

#defining in order of the definitions in HW pdf
variables = c("income", "exercise", "smoke", "bmi", "bp", "cholesterol", "angina", "stroke", "attack",

# Indexing used to populate different probability tables. Kind of like a map from integer to string.
#
#1 -> income
#2 -> exercise
#3 -> smoke
#4 -> bmi
#5 -> bp
#6 -> cholesterol
#7 -> angina
#8 -> stroke
#9 -> attack
#10 ->diabetes
#

# Using level order traversal to populate individual conditional probability tables.
#
# Naming Convention:
# Example: Variables are A, B, C.
# 1) A = createCPTfromData(...) -> creates the probability distribution of A from the given csv file
# 2) A_B = createCPTfromData(...) -> creates the probability distribution of A|B from the given csv fil
# 3) A_B_C = createCPTfromData(...) -> creates the probability distribution of A|B,C from the given csv
# And as follows
#
income = createCPTfromData(x = file, varnames = c("income"));
smoke_income = createCPTfromData(x = file[, c(3,1)], varnames = c("smoke", "income"))
bmi_income_excercise = createCPTfromData(x = file[, c(4,1,2)], varnames = c("bmi", "income", "exercise")
exercise_income = createCPTfromData(x = file[, c(2,1)], varnames = c("exercise", "income"))
```

```r
bp_income_exercise_smoke = createCPTfromData(x = file[, c(5,1,2,3)], varnames = c("bp", "income", "exerc
cholesterol_income_exercise_smoke = createCPTfromData(x = file[, c(6,1,2,3)], varnames = c("cholesterol
diabetes_bmi = createCPTfromData(x = file[, c(10,4)], varnames = c("diabetes","bmi"))
stroke_bmi_bp_cholesterol = createCPTfromData(x = file[, c(8,4,5,6)], varnames = c("stroke", "bmi", "bp
attack_bmi_bp_cholesterol = createCPTfromData(x = file[, c(9,4,5,6)], varnames = c("attack", "bmi", "bp
angina_bmi_bp_cholesterol = createCPTfromData(x = file[, c(7,4,5,6)], varnames = c("angina", "bmi", "bp

#create bayesnet with number to name mapping
bayesnet = list("1" = income, "2" = smoke_income, "3" = bmi_income_excercise, "4" = exercise_income, "5
                "6" = cholesterol_income_exercise_smoke, "7" = diabetes_bmi, "8" = stroke_bmi_bp_choles
                "10" = angina_bmi_bp_cholesterol)

#Uncomment the following two ines to see what the bayesian network is
#sprintf("The Bayesian Networks is made as follows:")
#print(bayesnet)

## Method to calculate the size of the bayesian network
l = 0;
for(i in 1:10){
  l = l + nrow(bayesnet[[i]]);
}
#(bayesnet[[2]])
#((nrow(bayesnet[[2]])))
print(sprintf("The Size of the Bayesian Network is %d", l))
```

```
## [1] "The Size of the Bayesian Network is 504"
```

Written Part

Question 1

What is the size (in terms of the number of probabilities needed) of this network? Alternatively, what is the total number of probabilities needed to store the full joint distribution?

Ans:

All the calculations neeeded are performed above. Please see the last few lines of Bayesian Network generation code above.

Size of the Baysian Network i.e. total number of probabilities needed to store the full joint distribution is: 504. Number of Probabilities computed in order to get the bayesian network: 504 - 10 = 494 (since for each variable, you can compute one value from the rest probabilities using Total Probability Theorem.)

Question 2

For each of the four health outcomes (diabetes, stroke, heart attack, angina), answer the following by querying your network (using your infer function):

a) What is the probability of the outcome if I have bad habits (smoke and don't exercise)? How about if I have good habits (don't smoke and do exercise)?

Ans:

We will show the result for each of the health outcomes through code. Then, after that it can be found organized properly in a table.

All the calculations are performed below. The output in form of tables can be seen here for different health outcome asked. After the code ends, result can be found in form of tables. I was facing difficulties making the tables directly in R, so I have added png files (screenshots of tables created in the word file) of these tables.

```
source("Bayesnet.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)

#### DIABETES ####
# Bad Habits -> Diabetes with Smoke and Don't exercise i.e we need to find P(diabetes | smoke = 1, exer

diabetes_smoke_noexercise = infer(bayesnet, setdiff(variables, c("diabetes", "smoke", "exercise")), c("s
(diabetes_smoke_noexercise)
```

```
##   exercise smoke diabetes       probs
## 1        2     1        1 0.159330402
## 2        2     1        2 0.007881463
## 3        2     1        3 0.812632416
## 4        2     1        4 0.020155719
```

```
# Good Habits -> Diabetes with No Smoke and Exercise i.e we need to find P(diabetes | smoke = 2, exerci

diabetes_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("diabetes", "smoke", "exercise")), c("s
(diabetes_nosmoke_exercise)
```

```
##   exercise smoke diabetes       probs
## 1        1     2        1 0.135062355
## 2        1     2        2 0.007673509
## 3        1     2        3 0.839391578
## 4        1     2        4 0.017872557
```

```
#### STROKE ####
# Bad Habits -> Stroke with Smoke and Don't exercise i.e we need to find P(stroke | smoke = 1, exercise

stroke_smoke_noexercise = infer(bayesnet, setdiff(variables, c("stroke", "smoke", "exercise")), c("smoke
(stroke_smoke_noexercise)
```

```
##   exercise smoke stroke      probs
## 1        2     1      1 0.05012672
## 2        2     1      2 0.94987328
```

```
# Good Habits -> Stroke with No Smoke and Exercise i.e we need to find P(stroke | smoke = 2, exercise =

stroke_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("stroke", "smoke", "exercise")), c("smoke
(stroke_nosmoke_exercise)
```

```
##   exercise smoke stroke      probs
## 1        1     2      1 0.03680824
## 2        1     2      2 0.96319176
```

```
#### HEART ATTACK ####
# Bad Habits -> Heart attack with Smoke and Don't exercise i.e we need to find P(attack | smoke = 1, ex

heartattack_smoke_noexercise = infer(bayesnet, setdiff(variables, c("attack", "smoke", "exercise")), c(
(heartattack_smoke_noexercise)
```

Diabetes	P(Diabetes | Smoke, Exercise)	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	**Good Habit** Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.1593	0.1351
2 (Only during pregnancy)	0.0079	0.0077
3 (No)	0.8127	0.8394
4 (Pre - Diabetic)	0.0201	0.0179

Figure 1:

```
##   exercise smoke attack      probs
## 1        2     1      1 0.07241543
## 2        2     1      2 0.92758457
```

```
# Good Habits -> Heart attack with No Smoke and Exercise i.e we need to find P(attack | smoke = 2, exer

heartattack_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("attack", "smoke", "exercise")), c(
(heartattack_nosmoke_exercise)
```

```
##   exercise smoke attack     probs
## 1        1     2      1 0.0510272
## 2        1     2      2 0.9489728
```

```
#### ANGINA ####
# Bad Habits -> Angina with Smoke and Don't exercise i.e we need to find P(angina | smoke = 1, exercise

angina_smoke_noexercise = infer(bayesnet, setdiff(variables, c("angina", "smoke", "exercise")), c("smok
(angina_smoke_noexercise)
```

```
##   exercise smoke angina      probs
## 1        2     1      1 0.07775608
## 2        2     1      2 0.92224392
```

```
# Good Habits -> Angina with No Smoke and Exercise i.e we need to find P(angina | smoke = 2, exercise =

angina_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("angina", "smoke", "exercise")), c("smok
(angina_nosmoke_exercise)
```

```
##   exercise smoke angina     probs
## 1        1     2      1 0.0523189
## 2        1     2      2 0.9476811
```

Final Results for Part a

1)**Diabetes**

2)**Stroke**

Stroke	P(Stroke | Smoke, Exercise)	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	Good Habit Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.0501	0.0368
3 (No)	0.9499	0.9632

Figure 2:

Attack	P(Attack | Smoke, Exercise)	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	Good Habit Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.0724	0.0510
3 (No)	0.9276	0.9490

Figure 3:

3)**Attack**

4)**Angina**

From the results above, we can see that for each of the health outcomes, the probability of having it reduces by some amount if you follow good habits (do exercise and don't smoke) as compared to the probability of having it if you have bad habits (smoke and don't exercise).

b) What is the probability of the outcome if I have poor health (high blood pressure, high cholesterol, and overweight)? What if I have good health (low blood pressure, low cholesterol, and normal weight)?

Ans:

Angina	P(Angina | Smoke, Exercise)	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	Good Habit Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.0778	0.0523
3 (No)	0.9222	0.9477

Figure 4:

These health symptoms are essentially ‘bp’, ‘cholesterol’, and ‘bmi’ in the Bayesian Networks. We will show the result for each of the health outcomes through code. Then, after that it can be found organized properly in a table.

All the calculations are performed below in the R snippet. The output in form of tables can be seen here for different health outcome asked. After the code ends, result can be found in form of tables. I was facing difficulties making the tables directly in R, so I have added png files (screenshots of tables created in the word file) of these tables.

```
source("Bayesnet.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)

#### DIABETES ####
# Bad Health -> Diabetes with poor health i.e. High BP, High Cholesterol, Overweight. We need to find P

diabetes_bp_chol_over = infer(bayesnet, setdiff(variables, c("diabetes", "bp", "cholesterol", "bmi")), c
(diabetes_bp_chol_over)
```

```
##   bmi bp cholesterol diabetes       probs
## 1   3  1           1        1 0.122279398
## 2   3  1           1        2 0.006717897
## 3   3  1           1        3 0.854002910
## 4   3  1           1        4 0.016999795
```

```
# Good Health -> Diabetes with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to find

diabetes_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("diabetes", "bp", "cholesterol", "bmi
(diabetes_nobp_nochol_normal)
```

```
##   bmi bp cholesterol diabetes       probs
## 1   2  3           2        1 0.061634465
## 2   2  3           2        2 0.007800312
## 3   2  3           2        3 0.919896796
## 4   2  3           2        4 0.010668427
```

```
#### STROKE ####
# Bad Health -> Stroke with poor health i.e. High BP, High Cholesterol, Overweight. We need to find P(s

Stroke_bp_chol_over = infer(bayesnet, setdiff(variables, c("stroke", "bp", "cholesterol", "bmi")), c("b
(Stroke_bp_chol_over)
```

```
##   bmi bp cholesterol stroke      probs
## 1   3  1           1      1 0.08397486
## 2   3  1           1      2 0.91602514
```

```
# Good Health -> Stroke with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to find P

Stroke_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("stroke", "bp", "cholesterol", "bmi"))
(Stroke_nobp_nochol_normal)
```

```
##   bmi bp cholesterol stroke      probs
## 1   2  3           2      1 0.01387042
## 2   2  3           2      2 0.98612958
```

```
#### HEART ATTACK ####
# Bad Health -> Heart Attack with poor health i.e. High BP, High Cholesterol, Overweight. We need to fi

attack_bp_chol_over = infer(bayesnet, setdiff(variables, c("attack", "bp", "cholesterol", "bmi")), c("b
(attack_bp_chol_over)
```

Diabetes	P(Diabetes | BP, Cholesterol, BMI)	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.1223	0.0616
2 (Only during pregnancy)	0.0067	0.0078
3 (No)	0.8540	0.9199
4 (Pre - Diabetic)	0.0170	0.1067

Figure 5:

```
##   bmi bp cholesterol attack   probs
## 1   3  1           1      1 0.13433
## 2   3  1           1      2 0.86567
```

```
# Good Health -> Heart Attack with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to f

attack_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("attack", "bp", "cholesterol", "bmi"))
(attack_nobp_nochol_normal)
```

```
##   bmi bp cholesterol attack      probs
## 1   2  3           2      1 0.01588794
## 2   2  3           2      2 0.98411206
```

```
#### ANGINA ####
# Bad Health -> Angina with poor health i.e. High BP, High Cholesterol, Overweight. We need to find P(a

angina_bp_chol_over = infer(bayesnet, setdiff(variables, c("angina", "bp", "cholesterol", "bmi")), c("bp
(angina_bp_chol_over)
```

```
##   bmi bp cholesterol angina     probs
## 1   3  1           1      1 0.1531853
## 2   3  1           1      2 0.8468147
```

```
# Good Health -> Angina with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to find P

angina_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("angina", "bp", "cholesterol", "bmi"))
(angina_nobp_nochol_normal)
```

```
##   bmi bp cholesterol angina      probs
## 1   2  3           2      1 0.01283874
## 2   2  3           2      2 0.98716126
```

Final Results for Part b

Diabetes

Stroke

Stroke	P(Stroke | BP, Cholesterol, BMI)	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.0840	0.0139
2 (No)	0.9160	0.9861

Figure 6:

Attack	P(Attack | BP, Cholesterol, BMI)	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.1343	0.0159
2 (No)	0.8657	0.9841

Figure 7:

Attack

Angina

From the above results, it can be concluded that the probability of having a health problem decreases by a significant amount if the health conditions (bmi, bp, cholesterol) falls in the Good Health region.

Angina	P(Angina | BP, Cholesterol, BMI)	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.1532	0.0128
2 (No)	0.8468	0.9872

Figure 8:

Question 3

Evaluate the effect a person's income has on their probability of having one of the four health outcomes (diabetes, stroke, heart attack, angina). For each of these four outcomes, plot their probability given income status (your horizontal axis should be i = 1, 2, . . . , 8, and your vertical axis should be P(y = 1 | income = i), where y is the outcome). What can you conclude?

Ans:

Here, we have 4 outcomes that we need to analyze with respect to income. Also, we need to evaluate the 4 probabilties i.e. P(diabetes = 1 | income = i), P(stroke = 1 | income = i), P(heart attack = 1 | income = i), and P(angina = 1 | income = i). The procedure is done below:

```r
source("Bayesnet.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)

#create arrays with 8 elements (1 to 8 in order)
inc = (1:8)
diabetes_variation = (1:8)
stroke_variation = (1:8)
attack_variation = (1:8)
angina_variation = (1:8)

# Here in all the cases, income is the observed variable
for(i in 1:8){
  p1 <- infer(bayesnet, setdiff(variables, c("diabetes", "income")), c("income"), i);
  diabetes_variation[i] = p1$probs[1];
  p2 <- infer(bayesnet, setdiff(variables, c("stroke", "income")), c("income"), i);
  stroke_variation[i] = p2$probs[1];
  p3 <- infer(bayesnet, setdiff(variables, c("attack", "income")), c("income"), i);
  attack_variation[i] = p3$probs[1];
  p4 <- infer(bayesnet, setdiff(variables, c("angina", "income")), c("income"), i);
  angina_variation[i] = p4$probs[1];
}

#Final variation of probabilities with income
sprintf("The Diabetes variation with income is:")
```

```
## [1] "The Diabetes variation with income is:"
```

```r
(diabetes_variation)
```

```
## [1] 0.1519546 0.1523172 0.1491232 0.1469863 0.1454104 0.1448793 0.1424329
## [8] 0.1330912
```

```r
sprintf("The Stroke variation with income is:")
```

```
## [1] "The Stroke variation with income is:"
```

```r
(stroke_variation)
```

```
## [1] 0.04952057 0.05213295 0.04884293 0.04719256 0.04557574 0.04323211
## [7] 0.04034332 0.03568678
```

```r
sprintf("The Heart Attack variation with income is:")
```

```
## [1] "The Heart Attack variation with income is:"
```

```r
(attack_variation)
```

```
## [1] 0.07034089 0.07459773 0.06918517 0.06668115 0.06433355 0.06099468
## [7] 0.05688049 0.04970372
```

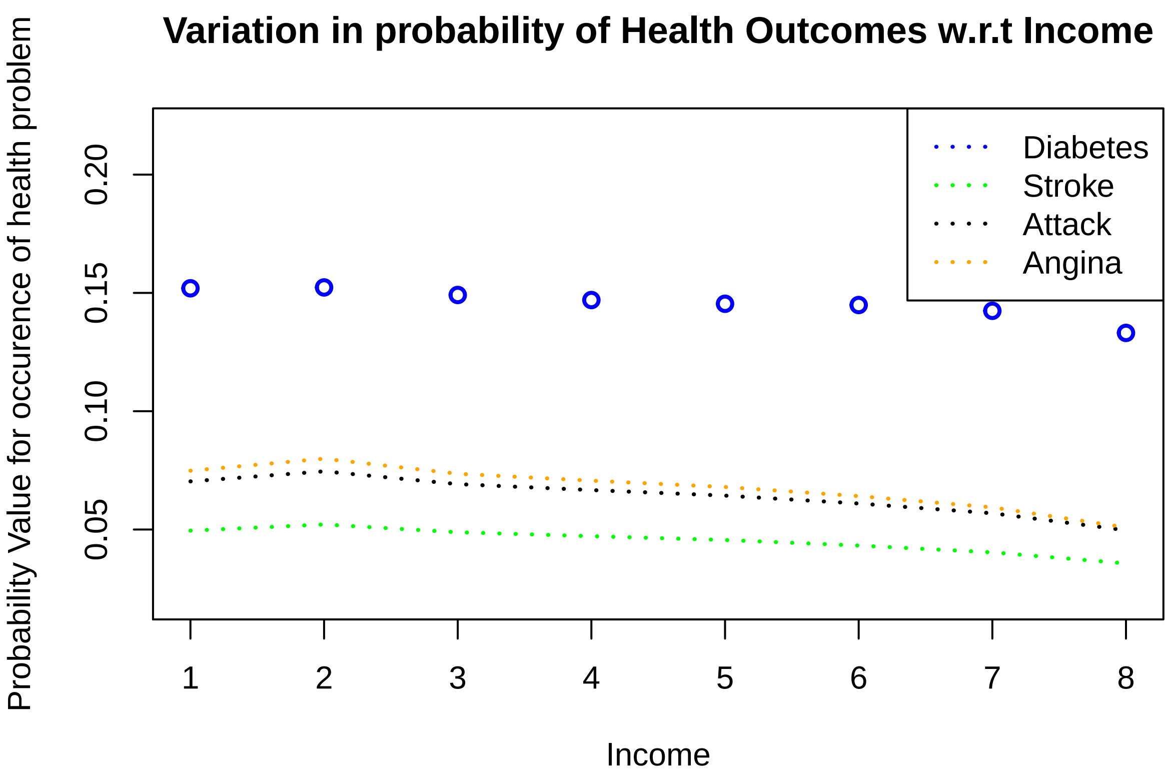
```r
sprintf("The Angina variation with income is:")
```

```
## [1] "The Angina variation with income is:"
```

```r
(angina_variation)
```

```
## [1] 0.07485571 0.07993036 0.07357066 0.07064398 0.06796010 0.06411896
## [7] 0.05936749 0.05088737
```

```r
#Plotting on the Diagram
plot(inc, diabetes_variation, main = "Variation in probability of Health Outcomes w.r.t Income", xlab =
lines(inc, stroke_variation, col="green", lty = 3, lwd = 2)
lines(inc, attack_variation, col="black", lty = 3, lwd = 2)
lines(inc, angina_variation, col="orange", lty = 3, lwd = 2)
legend('topright', c("Diabetes", "Stroke", "Attack", "Angina"), col = c("blue", "green", "black", "orang
```



From the above variation arrays and the plots, we can conclude that as the income of a person increases, the probability of having any of the adverserial health outcomes reduces. This seems a bit intuitive because if you are rich, you are taking care of yourself throughout the life. One important thing to note is that although the absolute difference in the percentages is tiny, but the relative differences between the percentages is large and is fit with the intuition.

For Diabetes, as the income group changes from 1 to 8, we observe that the probability of having diabetes decreased from 15.2% to 13.3%. It is a 1.9% absolute but nearly 13.5% relative decrease which is significant. Essentially, a person whose income is >75000 has 13.5% better chance of not having diabetes than the person whose income is <10000.

Also, for Stroke, as the income group changes from 1 to 8, we observe that the probability of having diabetes decreased from 4.95% to 3.56%. It is a 1.4% absolute but nearly 28% relative dip in the percentage which is significant. Essentially, a person whose income is >75000 has 28% better chance of not having diabetes than the person whose income is <10000.

On the same note, for Heart Attack, as the income group changes from 1 to 8, we observe that the probability of having diabetes decreased from 7% to 5%. It is a 2% absolute but nearly 30% relative dip in the percentage which is significant. Essentially, a person whose income is >75000 has 30% better chance of not having diabetes than the person whose income is <10000.

Similarly, for Angina, as the income group changes from 1 to 8, we observe that the probability of having diabetes decreased from 7.5% to 5%. It is a 1.4% absolute but nearly 33% relative dip in the percentage which is significant. Essentially, a person whose income is >75000 has 33% better chance of not having diabetes than the person whose income is <10000.

So, having higher income is in an inverse relationship with the health outcomes.

Question 4

Notice there are no links in the graph between the habits (smoking and exercise) and the outcomes. What assumption is this making about the effects of smoking and exercise on health problems? Let's test the validity of these assumptions. Create a second Bayesian network as above, but add edges from smoking to each of the four outcomes and edges from exercise to each of the four outcomes. Now redo the queries in Question 2. What was the effect, and do you think the assumptions of the first graph were valid or not?

Solution:

Here we are redoing all the queries given in Question 2 to check whether the assumptions made in the inital graph were valid or not. The initial graph had no links between the habits (smoking and exercise) and the health outcomes. This assumption considers that habits doesn't affect the health outcomes. We are going to check whether that assumption is valid or not.

a) What is the probability of the outcome if I have bad habits (smoke and don't exercise)? How about if I have good habits (don't smoke and do exercise)?

Ans:

We will show the result for each of the health outcomes in through code. Then, after that it can be found organized properly in a table.

```
source("Bayesnet_Q4.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)

#### DIABETES ####
# Bad Habits -> Diabetes with Smoke and Don't exercise i.e we need to find P(diabetes | smoke = 1, exer

diabetes_smoke_noexercise = infer(bayesnet, setdiff(variables, c("diabetes", "smoke", "exercise")), c("
(diabetes_smoke_noexercise)
```

```
##   exercise smoke diabetes       probs
## 1        2     1        1 0.226742871
## 2        2     1        2 0.006208261
## 3        2     1        3 0.742920993
## 4        2     1        4 0.024127875
```

```r
# Good Habits -> Diabetes with No Smoke and Exercise i.e we need to find P(diabetes | smoke = 2, exerci

diabetes_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("diabetes", "smoke", "exercise")), c("
(diabetes_nosmoke_exercise)
```

```
##   exercise smoke diabetes       probs
## 1        1     2        1 0.102527552
## 2        1     2        2 0.008883719
## 3        1     2        3 0.873671491
## 4        1     2        4 0.014917239
```

```r
#### STROKE ####
# Bad Habits -> Stroke with Smoke and Don't exercise i.e we need to find P(stroke | smoke = 1, exercise

stroke_smoke_noexercise = infer(bayesnet, setdiff(variables, c("stroke", "smoke", "exercise")), c("smok
(stroke_smoke_noexercise)
```

```
##   exercise smoke stroke      probs
## 1        2     1      1 0.07902771
## 2        2     1      2 0.92097229
```

```r
# Good Habits -> Stroke with No Smoke and Exercise i.e we need to find P(stroke | smoke = 2, exercise =

stroke_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("stroke", "smoke", "exercise")), c("smok
(stroke_nosmoke_exercise)
```

```
##   exercise smoke stroke      probs
## 1        1     2      1 0.02533856
## 2        1     2      2 0.97466144
```

```r
#### HEART ATTACK ####
# Bad Habits -> Heart attack with Smoke and Don't exercise i.e we need to find P(attack | smoke = 1, ex

heartattack_smoke_noexercise = infer(bayesnet, setdiff(variables, c("attack", "smoke", "exercise")), c(
(heartattack_smoke_noexercise)
```

```
##   exercise smoke attack     probs
## 1        2     1      1 0.1175417
## 2        2     1      2 0.8824583
```

```r
# Good Habits -> Heart attack with No Smoke and Exercise i.e we need to find P(attack | smoke = 2, exer

heartattack_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("attack", "smoke", "exercise")), c(
(heartattack_nosmoke_exercise)
```

```
##   exercise smoke attack      probs
## 1        1     2      1 0.03038452
## 2        1     2      2 0.96961548
```

```r
#### ANGINA ####
# Bad Habits -> Angina with Smoke and Don't exercise i.e we need to find P(angina | smoke = 1, exercise

angina_smoke_noexercise = infer(bayesnet, setdiff(variables, c("angina", "smoke", "exercise")), c("smok
(angina_smoke_noexercise)
```

```
##   exercise smoke angina     probs
## 1        2     1      1 0.1142626
## 2        2     1      2 0.8857374
```

Diabetes	P(Diabetes | Smoke, Exercise)	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	**Good Habit** Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.2267	0.1025
2 (Only during pregnancy)	0.0062	0.0089
3 (No)	0.7429	0.8736
4 (Pre - Diabetic)	0.0241	0.0149

Figure 9:

Stroke	P(Stroke | Smoke, Exercise)	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	**Good Habit** Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.0790	0.0253
3 (No)	0.9210	0.9747

Figure 10:

```
# Good Habits -> Angina with No Smoke and Exercise i.e we need to find P(angina | smoke = 2, exercise =

angina_nosmoke_exercise = infer(bayesnet, setdiff(variables, c("angina", "smoke", "exercise")), c("smok
(angina_nosmoke_exercise)
```

```
##   exercise smoke angina      probs
## 1        1     2      1 0.03596445
## 2        1     2      2 0.96403555
```

Results in form of table are:

Final Results for Part A

1)**Diabetes**

2)**Stroke**

3)**Attack**

4)**Angina**

Attack	**P(Attack | Smoke, Exercise)**	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	**Good Habit** Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.1175	0.0303
3 (No)	0.8825	0.9697

Figure 11:

Angina	**P(Angina | Smoke, Exercise)**	
	Bad Habit Smoke = 1 i.e. Yes Exercise = 2 i.e. No	**Good Habit** Exercise = 1 i.e. Yes Smoke = 2 i.e. No
1 (Yes)	0.1143	0.0359
3 (No)	0.8857	0.9640

Figure 12:

From the results in Question 2 and question 4, we can observe that the resulting probabilities differ in values. In question 2 results, we clearly saw that the reduction in probability if you have good habits than you have bad habits is very small. In question 4 results, it is a bit significant.

For ex. Diabetes: In question 2: P(Diabetes = 1 | Smoke and No Exercise) = 16% and P(Diabetes = 1 | No Smoke and Do Exercise) = 13.5%. This is a very small difference and doesn't fit very well with the intuition.

On the other hand, in Question 4, P(Diabetes = 1 | Smoke and No Exercise) = 22.67% and P(Diabetes = 1 | No Smoke and Do Exercise) = 10.25%. Now, this is a significant difference and it is exactly what we have as an intuition.

Similar arguments goes for other health outcomes i.e. Stroke, Attack, and Angina as well.

Hence, we can conclude that the assumptions made in question 2 about the graph were invalid. There should be a link between Habits and Health Outcomes.

b) What is the probability of the outcome if I have poor health (high blood pressure, high cholesterol, and overweight)? What if I have good health (low blood pressure, low cholesterol, and normal weight)?

Ans:

These health symptoms are essentially 'bp', 'cholesterol', and 'bmi' in the Bayesian Networks. We will show the result for each of the health outcomes in through code. Then, after that it can be found organized properly in a table.

```
source("Bayesnet_Q4.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)

#### DIABETES ####
# Bad Health -> Diabetes with poor health i.e. High BP, High Cholesterol, Overweight. We need to find P
```

```r
diabetes_bp_chol_over = infer(bayesnet, setdiff(variables, c("diabetes", "bp", "cholesterol", "bmi")), c
(diabetes_bp_chol_over)
```

```
##   bmi bp cholesterol diabetes       probs
## 1   3  1           1        1 0.131073667
## 2   3  1           1        2 0.006500468
## 3   3  1           1        3 0.844574233
## 4   3  1           1        4 0.017851632
```

```r
# Good Health -> Diabetes with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to find

diabetes_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("diabetes", "bp", "cholesterol", "bmi
(diabetes_nobp_nochol_normal)
```

```
##   bmi bp cholesterol diabetes      probs
## 1   2  3           2        1 0.05765442
## 2   2  3           2        2 0.00796897
## 3   2  3           2        3 0.92400937
## 4   2  3           2        4 0.01036724
```

```r
#### STROKE ####
# Bad Health -> Stroke with poor health i.e. High BP, High Cholesterol, Overweight. We need to find P(s

Stroke_bp_chol_over = infer(bayesnet, setdiff(variables, c("stroke", "bp", "cholesterol", "bmi")), c("bp
(Stroke_bp_chol_over)
```

```
##   bmi bp cholesterol stroke      probs
## 1   3  1           1      1 0.08575396
## 2   3  1           1      2 0.91424604
```

```r
# Good Health -> Stroke with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to find P

Stroke_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("stroke", "bp", "cholesterol", "bmi"))
(Stroke_nobp_nochol_normal)
```

```
##   bmi bp cholesterol stroke      probs
## 1   2  3           2      1 0.01335442
## 2   2  3           2      2 0.98664558
```

```r
#### HEART ATTACK ####
# Bad Health -> Heart Attack with poor health i.e. High BP, High Cholesterol, Overweight. We need to fi

attack_bp_chol_over = infer(bayesnet, setdiff(variables, c("attack", "bp", "cholesterol", "bmi")), c("bp
(attack_bp_chol_over)
```

```
##   bmi bp cholesterol attack     probs
## 1   3  1           1      1 0.1361617
## 2   3  1           1      2 0.8638383
```

```r
# Good Health -> Heart Attack with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to

attack_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("attack", "bp", "cholesterol", "bmi"))
(attack_nobp_nochol_normal)
```

```
##   bmi bp cholesterol attack      probs
## 1   2  3           2      1 0.01526521
## 2   2  3           2      2 0.98473479
```

Diabetes	P(Diabetes | BP, Cholesterol, BMI)	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.1311	0.0577
2 (Only during pregnancy)	0.0065	0.0080
3 (No)	0.8446	0.9240
4 (Pre - Diabetic)	0.0179	0.1037

Figure 13:

```
#### ANGINA ####
# Bad Health -> Angina with poor health i.e. High BP, High Cholesterol, Overweight. We need to find P(a

angina_bp_chol_over = infer(bayesnet, setdiff(variables, c("angina", "bp", "cholesterol", "bmi")), c("bp
(angina_bp_chol_over)
```

```
##   bmi bp cholesterol angina     probs
## 1   3  1           1      1 0.1548693
## 2   3  1           1      2 0.8451307
```

```
# Good Health -> Angina with good health i.e. Low BP, Low Cholesterol, Normal Weight. We need to find P

angina_nobp_nochol_normal = infer(bayesnet, setdiff(variables, c("angina", "bp", "cholesterol", "bmi"))
(angina_nobp_nochol_normal)
```

```
##   bmi bp cholesterol angina     probs
## 1   2  3           2      1 0.0124368
## 2   2  3           2      2 0.9875632
```

Results in form of table are:

Final Results for Part B

1)**Diabetes**

2)**Stroke**

3)**Attack**

4)**Angina**

From the results in Question 2b and question 4b, we can observe that the resulting probabilities differ in values. In question 2 results, we clearly saw that the reduction in probability if you have good health than you have bad health is very small. In question 4 results, it is a bit significant.

Stroke	**P(Stroke | BP, Cholesterol, BMI)**	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.0858	0.0134
2 (No)	0.9142	0.9866

Figure 14:

Attack	**P(Attack | BP, Cholesterol, BMI)**	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.1362	0.0153
2 (No)	0.8638	0.9847

Figure 15:

Angina	**P(Angina | BP, Cholesterol, BMI)**	
	Bad Health BP = 1 i.e. Yes Cholesterol = 1 i.e. Yes BMI = 3 i.e. Overweight	**Good Health** BP = 3 i.e. No Cholesterol = 2 i.e. No BMI = 2 i.e. Normal
1 (Yes)	0.1549	0.0124
2 (No)	0.8451	0.9876

Figure 16:

For ex. Diabetes: In question 2: P(Diabetes = 1 | Bad Health: High BP, High Cholesterol, Overweight) = 12.23% and P(Diabetes = 1 | Bad Health: High BP, High Cholesterol, Overweight) = 6.16%. This is a reasonable difference and fits with the intuition but not very well.

On the other hand, in Question 4, P(Diabetes = 1 | Bad Health: High BP, High Cholesterol, Overweight) = 13.11% and P(Diabetes = 1 | Bad Health: High BP, High Cholesterol, Overweight) = 5.77%. Now, this is a significant difference and it is exactly what we have as an intuition.

Similar arguments goes for other health outcomes i.e. Stroke, Attack, and Angina as well.

Finally, we can see that the addition of edges from Smoking and Exercise to Health Outcomes affects the part (A) queries a lot and a little bit to part (b) queries. Removing the assumptions made in the initial graph, we get a much more robust Bayesian Network.

Hence, we can conclude that the assumptions made in the initial graph (question 1 and 2) were invalid. There should be a link between Habits and Health Outcomes.

Question 5

Also notice there are no edges between the four outcomes. What assumption is this making about the interactions between health problems? Make a third network, starting from the network in Question 4, but adding an edge from diabetes to stroke. For both networks, evaluate the following probabilities:

P(stroke = 1 | diabetes = 1) and P(stroke = 1 | diabetes = 3)

Again, what was the effect, and was the assumption about the interaction between diabetes and stroke valid?

Solution:

Here, we are testing the hypothesis that whether the stroke and diabetes variable are related or not. Till this point, we have assumed no direct edge between the two variables.

Required Inferences for Question 4 Bayesian Network

```r
source("Bayesnet_Q4.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)
stroke_diabetes1 = infer(bayesnet, setdiff(variables, c("stroke", "diabetes")), c("diabetes"), c(1))
stroke_diabetes3 = infer(bayesnet, setdiff(variables, c("stroke", "diabetes")), c("diabetes"), c(3))
p1 = stroke_diabetes1$probs[1]
p2 = stroke_diabetes3$probs[1]

# P(stroke | diabetes = 1)
(stroke_diabetes1)
```

```
##   diabetes stroke      probs
## 1        1      1 0.04510135
## 2        1      2 0.95489865
```

```r
# P(stroke = 1 | diabetes = 1)
(p1)
```

```
## [1] 0.04510135
```

```r
# P(stroke | diabetes = 3)
(stroke_diabetes3)
```

Stroke	Before adding the edges	After adding the edges
P(Stroke = 1 | Diabetes = 1)	0.0451	0.0764
P(Stroke = 1 | Diabetes = 3)	0.0412	0.0359

Figure 17:

```
##   diabetes stroke      probs
## 1        3      1 0.04122912
## 2        3      2 0.95877088
```

```
# P(stroke = 1 | diabetes = 3)
(p2)
```

```
## [1] 0.04122912
```

Required Inferences for Question 5 updated Bayesian Network

```
source("Bayesnet_Q5.r", echo = FALSE, keep.source = FALSE, max.deparse.length=10000)
stroke_diabetes1 = infer(bayesnet, setdiff(variables, c("stroke", "diabetes")), c("diabetes"), c(1))
stroke_diabetes3 = infer(bayesnet, setdiff(variables, c("stroke", "diabetes")), c("diabetes"), c(3))
p1 = stroke_diabetes1$probs[1]
p2 = stroke_diabetes3$probs[1]

# P(stroke | diabetes = 1)
(stroke_diabetes1)
```

```
##   diabetes stroke      probs
## 1        1      1 0.07642781
## 2        1      2 0.92357219
```

```
# P(stroke = 1 | diabetes = 1)
(p1)
```

```
## [1] 0.07642781
```

```
# P(stroke | diabetes = 3)
(stroke_diabetes3)
```

```
##   diabetes stroke      probs
## 1        3      1 0.03586261
## 2        3      2 0.96413739
```

```
# P(stroke = 1 | diabetes = 3)
(p2)
```

```
## [1] 0.03586261
```

From the values that we have calculated above for the Baysian network of Q4 and Q5. They are summarized below.

From these values, we can conclude that stroke and diabetes are in some way related. It is evident from the fact that the probability of a person having a stroke given he/she has diabetes relatively increases by nearly

40% when the edge between them is added. Thus, the assumption that the stroke and diabetes variable are related is valid.

Question 6

Finally, make sure that your code runs correctly on all of the examples in BayesNetExamples.r. Your code will be graded for correctness on these also.

Ans:

The answers from the Bayesian Network Examples are added below. The results for each question can be found in the included source itself. I have checked, it matches with the correct answers with an error margin of less than 2-3%.

```
source("./BayesNetworkExamples.r", echo = TRUE, keep.source = TRUE, max.deparse.length=10000)
```

```
##
## > source("BayesianNetworks-template.r");
##
## > ####################################
## > ## Simple chain example: x -> y -> z
## > ####################################
## > x = createCPT(list("x"), probs = c(0.3, 0.7), levelsList = list(c("T", "F")))
##
## > yx = createCPT(list("y", "x"), probs = c(0.8, 0.4, 0.2, 0.6),
## +                levelsList = list(c("T", "F"), c("T", "F")))
##
## > zy = createCPT(list("z", "y"), probs = c(0.5, 0.6, 0.5, 0.4),
## +                levelsList = list(c("T", "F"), c("T", "F")))
##
## > (xyzNet = list("x" = x, "y" = yx, "z" = zy))
## $x
##   probs x
## 1   0.3 T
## 2   0.7 F
##
## $y
##   probs y x
## 1   0.8 T T
## 2   0.4 T F
## 3   0.2 F T
## 4   0.6 F F
##
## $z
##   probs z y
## 1   0.5 T T
## 2   0.6 T F
## 3   0.5 F T
## 4   0.4 F F
##
##
## > ## Some simple operations you might try to check your code
## > productFactor(x, yx)
##   x y probs
```

```
## 1 F T  0.28
## 2 F F  0.42
## 3 T T  0.24
## 4 T F  0.06
##
## > productFactor(productFactor(x, yx), zy)
##   y x z probs
## 1 F F T 0.252
## 2 F F F 0.168
## 3 F T T 0.036
## 4 F T F 0.024
## 5 T F T 0.140
## 6 T F F 0.140
## 7 T T T 0.120
## 8 T T F 0.120
##
## > marginalizeFactor(productFactor(x, yx), "x")
##   y probs
## 1 F  0.48
## 2 T  0.52
##
## > marginalizeFactor(productFactor(yx, zy), "z")
##   y x probs
## 1 F F   0.6
## 2 T F   0.4
## 3 F T   0.2
## 4 T T   0.8
##
## > ## Notice in the observe function, you just need to delete rows that are
## > ## inconsistent with the given observations. Factors do not need to be combined
## > ## or normalized in this step.
## > observe(xyzNet, "x", "T")
## $x
##   probs x
## 1   0.3 T
##
## $y
##   probs y x
## 1   0.8 T T
## 3   0.2 F T
##
## $z
##   probs z y
## 1   0.5 T T
## 2   0.6 T F
## 3   0.5 F T
## 4   0.4 F F
##
##
## > observe(xyzNet, c("x", "y"), c("T", "T"))
## $x
##   probs x
## 1   0.3 T
##
```

```
## $y
##   probs y x
## 1   0.8 T T
##
## $z
##   probs z y
## 1   0.5 T T
## 3   0.5 F T
##
##
## > ## Marginalize must first combine all factors involving the variable to
## > ## marginalize. Again, this operation may lead to factors that aren't
## > ## probabilities.
## > marginalize(xyzNet, "x")
##   y z probs
## 1 F F 0.192
## 2 T F 0.260
## 3 F T 0.288
## 4 T T 0.260
##
## > marginalize(xyzNet, "y")
##   x z probs
## 1 F F 0.308
## 2 T F 0.144
## 3 F T 0.392
## 4 T T 0.156
##
## > marginalize(xyzNet, "z")
##   y x probs
## 1 F F  0.42
## 2 T F  0.28
## 3 F T  0.06
## 4 T T  0.24
##
## > marginalize(xyzNet, c("x", "z"))
##   y probs
## 1 F  0.48
## 2 T  0.52
##
## > ############################
## > ## Bishop book (Ch 8) example
## > ############################
## > b = createCPT(list("battery"), probs = c(0.9, 0.1), levelsList = list(c(1, 0)))
##
## > f = createCPT(list("fuel"), probs = c(0.9, 0.1), levelsList = list(c(1, 0)))
##
## > gbf = createCPT(list("gauge", "battery", "fuel"),
## +                 probs = c(0.8, 0.2, 0.2, 0.1, 0.2, 0.8, 0.8, 0.9),
## +                 levelsList = list(c(1, 0), c(1, 0), c(1, 0)))
##
## > carNet = list("battery" = b, "fuel" = f, "gauge" = gbf)
##
## > ## Some examples:
## > ## Notice that different order of operations give the same answer
```

```
## > ## (rows/columns may be permuted)
## > productFactor(productFactor(b, f), gbf)
##   battery fuel gauge probs
## 1       0    0     1 0.001
## 2       0    0     0 0.009
## 3       0    1     1 0.018
## 4       0    1     0 0.072
## 5       1    0     1 0.018
## 6       1    0     0 0.072
## 7       1    1     1 0.648
## 8       1    1     0 0.162
##
## > productFactor(productFactor(gbf, f), b)
##   battery fuel gauge probs
## 1       0    0     1 0.001
## 2       0    0     0 0.009
## 3       0    1     1 0.018
## 4       0    1     0 0.072
## 5       1    0     1 0.018
## 6       1    0     0 0.072
## 7       1    1     1 0.648
## 8       1    1     0 0.162
##
## > marginalizeFactor(productFactor(gbf, b), "gauge")
##   battery fuel probs
## 1       0    0   0.1
## 2       1    0   0.9
## 3       0    1   0.1
## 4       1    1   0.9
##
## > productFactor(marginalizeFactor(gbf, "gauge"), b)
##   battery fuel probs
## 1       0    0   0.1
## 2       0    1   0.1
## 3       1    0   0.9
## 4       1    1   0.9
##
## > productFactor(marginalizeFactor(productFactor(gbf, b), "battery"), f)
##   fuel gauge probs
## 1    0     0 0.081
## 2    0     1 0.019
## 3    1     0 0.234
## 4    1     1 0.666
##
## > marginalizeFactor(productFactor(productFactor(gbf, f), b), "battery")
##   fuel gauge probs
## 1    0     0 0.081
## 2    1     0 0.234
## 3    0     1 0.019
## 4    1     1 0.666
##
## > marginalizeFactor(productFactor(marginalizeFactor(productFactor(gbf, b), "battery"), f), "gauge")
##   fuel probs
## 1    0   0.1
```

```
## 2    1   0.9
##
## > marginalizeFactor(productFactor(marginalizeFactor(productFactor(gbf, b), "battery"), f), "fuel")
##   gauge probs
## 1     0 0.315
## 2     1 0.685
##
## > ## Examples computed in book (see pg. 377)
## > infer(carNet, c("battery", "fuel"), NULL, NULL)     ## (8.30)
##   gauge probs
## 1     0 0.315
## 2     1 0.685
##
## > infer(carNet, c("battery"), "fuel", 0)              ## (8.31)
##   fuel gauge probs
## 1    0     0  0.81
## 2    0     1  0.19
##
## > infer(carNet, c("battery"), "gauge", 0)             ## (8.32)
##   fuel gauge     probs
## 1    0     0 0.2571429
## 2    1     0 0.7428571
##
## > infer(carNet, NULL, c("gauge", "battery"), c(0, 0)) ## (8.33)
##   battery fuel gauge     probs
## 1       0    0     0 0.1111111
## 2       0    1     0 0.8888889
##
## > ###########################################################################
## > ## Kevin Murphy's Example: http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html
## > ###########################################################################
## > c = createCPT(list("cloudy"), probs = c(0.5, 0.5),
## +               levelsList = list(c("F", "T")))
##
## > rc = createCPT(list("rain", "cloudy"), probs = c(0.8, 0.2, 0.2, 0.8),
## +                levelsList = list(c("F", "T"), c("F", "T")))
##
## > sc = createCPT(c("sprinkler", "cloudy"), probs = c(0.5, 0.9, 0.5, 0.1),
## +                levelsList = list(c("F", "T"), c("F", "T")))
##
## > wsr = createCPT(list("wet", "sprinkler", "rain"),
## +                 probs = c(1, 0.1, 0.1, 0.01, 0, 0.9, 0.9, 0.99),
## +                 levelsList = list(c("F", "T"), c("F", "T"), c("F", "T")))
##
## > grassNet = list("cloudy" = c, "rain" = rc, "sprinkler" = sc, "wet" = wsr)
##
## > ## Test your infer() method by replicating the computations on the website!!
## > p1 = infer(grassNet, c("cloudy", "rain"), c("wet"), c("T"));
##
## > (p1$probs[2])
## [1] 0.4297636
##
## > p2 = infer(grassNet, c("cloudy", "sprinkler"), "wet", "T");
##
```

```
## > (p2$probs[2])
## [1] 0.7079277
##
## > p3 = infer(grassNet, c("cloudy", "rain", "sprinkler"), NULL, NULL)
##
## > (p3$probs[2])
## [1] 0.6471
```