Project 1

Discrete Bayesian networks in RStudio

In this project we will be introduced to discrete Bayesian networks with RStudio.

1. Introduction

A Bayesian network (BN) is a type of graphical model that contains 2 components:

1. A **directed acyclic graph** (DAG).
2. A set of **conditional probability distributions** (CPDs).

Such that:

- Each node of the DAG represents a random variable of the network, which can be discrete, categorical or continuous. When all variables are discrete and/or categorical, the BN is said to be discrete.
- A relation (edge) between 2 nodes X and Y, symbolized as $X \to Y$, represents a direct dependence of X on Y. In this sense, we will say that X is the parent of Y (so Y is the child of X).
- The DAG is *directed* because the relationships between nodes have directional sense ($X \to Y$ or $Y \to X$); and it is *acyclic* because the graph cannot contain cycles.
- Each node X_i of the graph has a conditional probability distribution associated with it, denoted $P(X_i \mid \mathrm{Pa}_{X_i})$, where Pa_{X_i} are the parent nodes of node X_i. This distribution quantifies the effect of the parents on node X_i. For discrete Bayesian networks, these distributions are **conditional probability tables** (CPTs).

2. Example of a discrete BN: Burglary Network

The figure 1 shows a famous example of a discrete Bayesian network (*Burglary Network*), created by Judea Pearl.

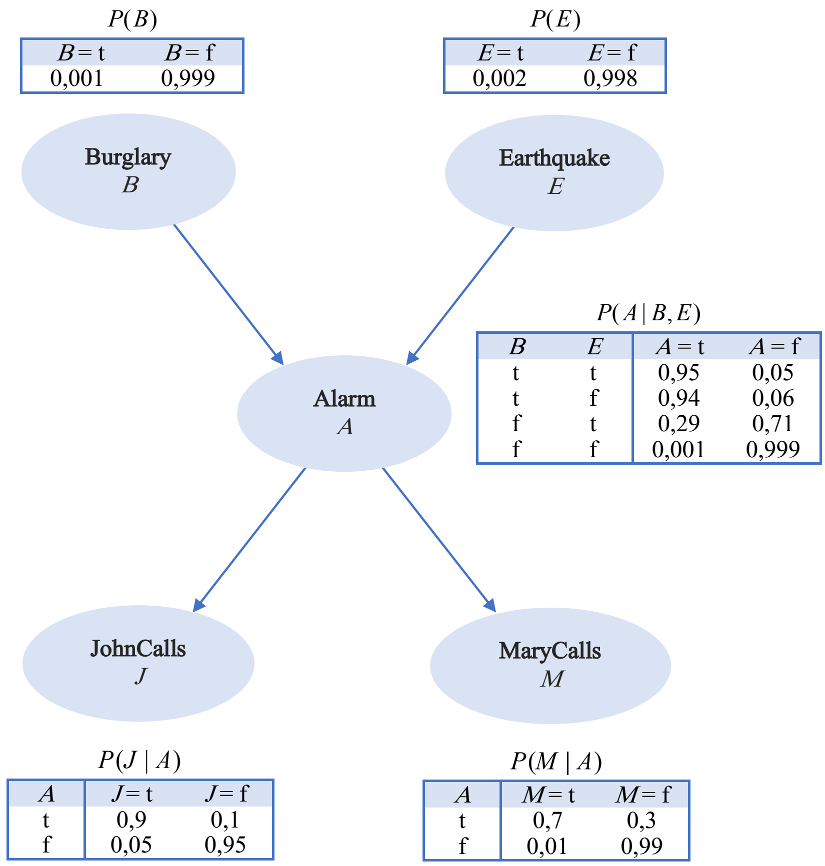

Figura 1: Example of discrete Bayesian network (*Burglary Network*, by J. Pearl)

This network models an anti-theft system installed in a home. As we can see, there are 5 variables or nodes in this network: Burglary (B), Earthquake (E), Alarm (A), JohnCalls (J) and MaryCalls (M).
The phenomenon modeling the network would be as follows:
When a burglary occurs in the tenant's house, the alarm will start to sound (the variable B influences E). Also, since the house is located in an earthquake-prone region of Los Angeles, earthquakes could also trigger the alarm (E influences A). Finally, in case the tenant is away from home, his neighbors John and Mary have promised to call him on his cell phone in case they hear the alarm (A influences J and M).

On the other hand, all variables appearing in this network are categorical, with 2 possible values: true (t) or false (f), that is, the variable happens or does not happen, respectively. Consequently, $B = \mathrm{t}$ would mean that a theft occurs, while $B = \mathrm{f}$ would indicate that a theft does not occur. Similarly, $A = \mathrm{t}$ would symbolize that the alarm sounds, while $A = \mathrm{f}$ would denote that the alarm does not sound. This reasoning would apply to all other variables.

We can also see that the graph in figure 1 constitutes a DAG, since all its relations are directional (have arrow) and we do not find any cycle in it.

In this discrete BN, in addition to the corresponding DAG, we can highlight that each node includes its conditional probability table or CPT. The CPT of a node X_i will be different, depending on whether it has parents or not:

- If the node X_i has no parents (variables B and E), the CPT will contain only one row, with probabilities $P(X_i)$ (one probability for each value of X_i). In the example, the CPT of the variable *Burgalry* consists of the probabilities $P(B)$, i.e., the probability that a theft occurs, $P(B = \mathrm{t}) = 0.001$, and the probability that it does not occur, $P(B = \mathrm{f}) = 0.999$. These probabilities $P(X_i)$ are known as a priori probabilities.
- If node X_i has parents (variables A, J and M), the CPT will store the conditional probabilities $P(X_i \mid \mathrm{Pa}_{X_i})$ (one probability for each value of X_i, and for each combination of parent values). For example, the CPT of the variable *Alarm* includes the probabilities $P(A \mid B, E)$, since the parents of A are B and E. Each row of the table has the probability that the alarm will or will not sound, conditional on a combination of values of *Burgalry* and *Earthquake*. Thus, the 1st row would include 2 probabilities: (1) the probability that the alarm sounds, knowing that a burglary and earthquake occur, $P(A = \mathrm{t} \mid B = \mathrm{t}, E = \mathrm{t})$ =0.95; and (2) the probability that the alarm does not sound, knowing that a burglary and earthquake occur, $P(A = \mathrm{f} \mid B = \mathrm{t}, E = \mathrm{t})$ =0.05.

Whether the node has parents or not, we should note that all rows of the CPTs should sum to 1.

CPTs are also mechanisms to account for uncertainty. For example, the alarm might not sound even if a burglary occurs, because the power supply has been cut off. This would be specified by $P(A = \mathrm{f} \mid B = \mathrm{t}, E = \mathrm{t}) = 0.05$ and $P(A = \mathrm{f} \mid B = \mathrm{t}, E = \mathrm{f}) = 0.06$. Another example might be John calling the tenant when the alarm doesn't go off (because he mistakes it for a car on the street), represented with $P(J = \mathrm{t} \mid A = \mathrm{f}) = 0.05$.

3. Global and local distributions

One of the great advantages of BNs is that they allow us to encode, in a compact form, the complete joint probability distribution, also called *global distribution*.

The **global distribution** (for a discrete BN), denoted $P(X_1, X_2, \dots, X_n)$, consists of a table with the probabilities for each combination of values of the network variables. Thus, each row of the global distribution table holds a probability of the form:

$$P(X_1 = x_1, X_2 = x_2, X_n = x_n) \tag{1}$$

Where:

- $X_1, X_2, \dots, X_n$ are the variables or nodes of the BN.
- $x_1, x_2, \dots, x_n$ is a possible combination of values of the variables or nodes of the BN.

Now, we can see the form that the global distribution table would have for our example:

B	E	A	J	M	$P(B,E,A,J,M)$	
t	t	t	t	t	0,000001197	□ $P(B = \text{t}, E = \text{t}, A = \text{t}, J = \text{t}, M = \text{t})$
t	t	t	t	f	0,000000513	□ $P(B = \text{t}, E = \text{t}, A = \text{t}, J = \text{t}, M = \text{f})$
t	t	t	f	t	0,000000133	□ $P(B = \text{t}, E = \text{t}, A = \text{t}, J = \text{f}, M = \text{t})$
t	t	t	f	f	0,000000057	□ $P(B = \text{t}, E = \text{t}, A = \text{t}, J = \text{f}, M = \text{f})$
⋮	⋮	⋮	⋮	⋮	⋮	
f	f	f	f	t	0,009462	□ $P(B = \text{f}, E = \text{f}, A = \text{f}, J = \text{f}, M = \text{t})$
f	f	f	f	f	0,9367427	□ $P(B = \text{f}, E = \text{f}, A = \text{f}, J = \text{f}, M = \text{f})$

Figura 2: Table of the global distribution for the BN of *Burglary Network*.

We could think that to define this global distribution we would need a total of $2^5 = 32$ probabilities, corresponding to all the combinations of values of the variables (2 possible values for the 5 variables: 2^5 probabilities). However, since the sum of all probabilities is equal to 1, we would only need $2^5 - 1 = 31$ probability values, since the last value would be fixed. These indispensable probability values are called **parameters**.

Therefore, the number of parameters P_G for the global distribution of a discrete BN will be:

$$P_G = \left(\prod_{i=1}^{n} \left| \mathrm{Val}(X_i) \right| \right) - 1 \tag{2}$$

Where $\left|\text{Val}(X_i)\right|$ is the number of values/categories of each variable X_i in the BN.

However, working with the global distribution could be tedious. Even for a BN with few variables, we would need a large amount of memory to store such a distribution. For example, if we had a BN of 20 variables, with 2 values each, we would need $2^{20} - 1 = 1048575$ parameters.

Fortunately, we can use the information encoded in the BN to decompose the global distribution into a set of smaller **local distributions**, one for each variable. Assuming that variables unrelated by an arc are conditionally independent, we can factor the global distribution as follows:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid \text{Pa}_{X_i}\right) \tag{3}$$

Where each factor $P\left(X_i | \text{Pa}_{X_i}\right)$ corresponds to the local distribution of the variable X_i. For the case of discrete BNs, the local distributions would be the CPTs of the variables.

If in our example we factor the global distribution by local distributions, we would have that:

$$P(B, E, A, J, M) = P(B) \cdot P(E) \cdot P\left(A \mid B, E\right) \cdot P\left(J \mid A\right) \cdot P\left(M \mid A\right) \tag{4}$$

So, if we wanted to calculate the probability that Mary and John call when the alarm sounds because there is a burglary (but not an earthquake), we would have:

$$\begin{aligned} P(B = t, E = f, A = t, J = t, M = t) = \\ P(B = t) \cdot P(E = f) \cdot P\left(A = t \mid B = t, E = f\right) \cdot \\ P\left(J = t \mid A = t\right) \cdot P\left(M = t \mid A = t\right) = \\ 0.001 \cdot 0.998 \cdot 0.94 \cdot 0.9 \cdot 0.7 = 0.0005910156 \end{aligned} \tag{5}$$

Likewise, the number of parameters P_L of the local distributions of a discrete BN will be:

$$P_L = \sum_{i=1}^{n} \left[\left(\left|\text{Val}(X_i)\right| - 1\right) \cdot \prod_{j=1}^{m} \left|\text{Val}(\text{Pa}_{j,X_i})\right| \right] \tag{6}$$

Where:

- n: Total number of variables in the BN.
- $\left|\text{Val}(X_i)\right|$: Number of values/categories of the variable X_i.

- m: Total number of parents of the variable X_i.
- $\left|\text{Val}(\text{Pa}_{j,X_i})\right|$: Number of values/categories of the parent j of the variable X_i.

Note: If the variable X_i does not have parents, we substitute 1 for its corresponding product operator.

For the example of *Burglary Network* the number of parameters of the local distributions would be:

$$\begin{aligned} P_L = \underbrace{(2-1)}_{\left|\text{Val}(B)\right|-1} + \underbrace{(2-1)}_{\left|\text{Val}(E)\right|-1} + \underbrace{(2-1)}_{\left|\text{Val}(A)\right|-1} \cdot \underbrace{2\cdot 2}_{\left|\text{Val}(B)\right|\cdot\left|\text{Val}(E)\right|} + \\ \underbrace{(2-1)}_{\left|\text{Val}(J)\right|-1} \cdot \underbrace{2}_{\left|\text{Val}(A)\right|} + \underbrace{(2-1)}_{\left|\text{Val}(M)\right|-1} \cdot \underbrace{2}_{\left|\text{Val}(A)\right|} = 10 \text{ parameters} \end{aligned} \tag{7}$$

As we can see, by factoring the global distribution we can reduce the number of parameters needed (in this example, from 31 parameters we reduce to 10).

4. Exercises

In this section we will perform different exercises to introduce us to discrete BNs with RStudio.

Exercise 1 A hypothetical survey of 500 people has been carried out to investigate possible patterns in the use of different means of transportation, such as cars and trains (among others)[a]. In the survey, 6 variables have been measured, each with different values, namely:

- **Age** (A): *young* (if the person is younger than 30 years old), *adult* (between 30 and 60 years old) and *old* (over 60 years old).
- **Sex** (S): male M or female F.
- **Education** (E): *high school* (if the person completed at most secondary education) or *university* (if the person has university studies).
- **Occupation** (O): if the person is *employed* or *self-employed*.
- **Residence** (R): if the city where the person resides is *small* or *big*.
- **Transportation** (T): whether the person usually travels by *car*, *train* or other means.

In addition, we assume the following relationships between the variables:

- No variable seems to affect the age and gender of respondents.

- Nowadays, being young seems to influence having a college education.
- Gender seems to influence which college major is chosen. For example, Computer Science has a higher percentage of men enrolled, while Early Childhood Education has a higher percentage of women enrolled.
- In general, a better education influences access to better jobs.
- The locality where a student lives seems to be influenced by the university where he/she studies (many students move close to their school).
- The job position might influence the mode of transportation used. For example, a remote place to work would require the use of a car, whereas if we work 2 blocks from our home, we could walk.
- The size of the city where one lives seems to influence the type of transportation used. For example, in large cities many people choose not to use the car, as the train/metro is a good alternative.

a) From the above relationships, represent by hand the corresponding DAG (without using RStudio). What would be the parents and children of each variable in the DAG?

b) Write the expression of the factored global distribution. Calculate the number of parameters of the global distribution, as well as the number of parameters of the local distributions.

Performs the sections **c)**, **d)** and **(e)** in the same RStudio script:

c) Loads in RStudio the Excel file `survey.xlsx`, attached to this practice, which contains all the survey data. Call the loaded data `survey`. Finally, verify that the file has been loaded correctly.

Note: Find information on how to load Excel files in RStudio.

d) Obtain and print by console the CPTs of the survey variables, following the following considerations:

- First, store each variable in `survey` separately as a factor (since all variables are categorical), with the levels ordered according to the statement. For example, for the variable *Age*:

```
age <- factor(survey$A,levels=c("young", "adult", "old"))
```

- If a variable (e.g. A, with values a_1, a_2, a_3) has no parents, store its CPT in a vector, so that when printed it looks like this:

```
> #CPT of variable A: P(A)
> A
a1  a2  a3
0.3 0.5 0.2
```

  Calculate each $P(a_i)$ of the vector as the ratio of the favorable cases of a_i to the total cases.

- If a variable (e.g. C, with values c_1 and c_2) has one or more parents (e.g. A with values a_1 and a_2; and B with values b_1 and b_2), store its CPT in an array, such that when printed it looks like this:

```
> #CPT of variable C: P(C|B,A)
> C
          c1   c2
b1 & a1 0.23 0.77
b1 & a2 0.40 0.60
b2 & a1 0.56 0.44
b2 & a2 0.37 0.63
```

  Calculate each matrix probability as:

$$P(c_i \mid b_j, a_k) = P(c_i, b_j, a_k)/P(b_j, a_k)$$

- To calculate the probabilities of the CPTs use only the functions: **`sum()`**, **`which()`** and **`length()`**. To access the documentation for a function in RStudio, type **`?functionName`** in the console and press ⏎.

(e) From the above CPTs, calculate the following probabilities:

- $P(A = \textit{young}, S = M, E = \textit{university}, O = \textit{self-employed}, R = \textit{big}, T = \textit{car})$.
- $P(A = \textit{old}, S = F, E = \textit{high school}, O = \textit{employed}, R = \textit{big}, T = \textit{other})$.
- $P(A = \textit{adult}, S = M, E = \textit{university}, O = \textit{self-employed}, R = \textit{small}, T = \textit{train})$.

(f) Based on the slides in Theme 3:

1. Write the local independence relations located in the BN.
2. From the chain rule of the joint distribution, using the local independencies above, obtain the factorization of the joint for the BN.

[a] This hypothetical survey has been proposed by M. Scutari and J.-B. Denis

Exercise 2 Create a script in RStudio to solve the following sections:

(a) Install and load in RStudio the package `bnlearn`, which will allow us to work with BNs. Next, create an object `dag`, which will contain the DAG corresponding to the survey of Exercise 1:

1. Start by adding the survey variables to the DAG, using:
   `dag <- empty.graph(variables)`, where `variables` is a character type vector with the nodes of the DAG separated by commas.

2. Adds to the DAG each relation of the form $X \rightarrow Y$ with:
   `dag <- set.arc(dag, from = "X", to = "Y")`.

3. Print the created DAG by console. From the information that appears when printing the DAG, how can we verify that the created DAG is correct?

(b) What happens if we try to introduce in `dag` the relation $R \rightarrow S$? What functions of `bnlearn` allow us to obtain the nodes, the arcs, the parents and the children of each node of the DAG? Apply these functions to `dag`.

Hint: Enter `? "misc utilities"` in the console.

Finally, find the appropriate function to obtain the factorization of the global distribution of the BN. Verify that the results obtained are identical to those of Exercise 1.

(c) Is also possible to create a DAG by entering all the relations simultaneously with a matrix, instead of using one **set.arc()** for each relation. If we had a DAG with variables A, B, and C, and relations $A \rightarrow B$ and $A \rightarrow C$, you can create the matrix of relations (`arcs`) like this:

```
#First we create the DAG.
#(We can also introduce the nodes with the argument "nodes").
dag <- empty.graph(nodes = c("A", "B", "C"))
#We build the relationship matrix or arcs.
#(2 column matrix filled by rows; 1 relation per row).
arcs <- matrix(c("A", "B",
                 "A", "C"), byrow=TRUE, ncol = 2)
#Assign the relationship matrix to the created DAG.
arcs(dag) <- arcs
```

Creates a DAG `otherDag`, with the same variables and relations as `dag`, but using the argument `nodes` and the corresponding relations matrix. Check that **dag** and **otherDag** are identical with the function **all.equal()**.

(d) Finally, you can create a DAG equivalent to the above from the factorization of the global distribution. Assuming a DAG with variables A, B, C and D; and factorization $P(A) \cdot P(B) \cdot P(C|A) \cdot P(D|A, B)$, we could create the corresponding DAG with the function **model2network**:

```
#Create DAG from the factorization of the global distribution.
 dag <- model2network("[A][B][C|A][D|A:B]")
```

Create by this method a DAG named **lastDag**, identical to the previous DAGs. Print it and verify that it is equal to **dag**. ■

Exercise 3 Create a script in RStudio to solve the following sections:

(a) In Exercise 2 we created the DAG for the survey variables (**dag**). However, to build the complete BN, in addition to the DAG, we need the CPTs for each variable. These CPTs were already calculated in Exercise 1, but the format we use is not suitable for working with **bnlearn**. The following example shows how to calculate the **cpts** of the *Burglary Network* with the correct format (using **array()**):

```
#We assume that the variables "burglary", "earthquake", "alarm"
#"johnCalls" and "maryCalls" have already been read from Excel and
#converted to factors with the corresponding levels
#("t" and "f").

#We created the CPT of Burglary (B), with the necessary formatting
#to be able to working with bnlearn.
cptB <- array(c(0.001, 0.999),
              dim = 2,
              dimnames = list(B = levels(burglary)))

#CPT of Earthquake(E).
cptE <- array(c(0.002, 0.998),
              dim = 2,
              dimnames = list(E = levels(earthquake)))

#CPT of Alarm (A).
#(Each element of "dim" corresponds to the number of levels of
#each variable indicated in "dimnames").
cptA <- array(c(0.95, 0.05, 0.94, 0.06, 0.29, 0.71, 0.001, 0.999),
```

```
             dim = c(2,2,2),
                dimnames = list(A = levels(alarm),
                                E = levels(earthquake),
                                B = levels(burglary)))

#CPT by JohnCalls (J).
cptJ <- array(c(0.9, 0.1, 0.05, 0.95),
      dim = c(2,2),
      dimnames = list(J = levels(alarm),
                             A = levels(alarm)))

#CPT of MaryCalls (M).
cptM <- array(c(0.7, 0.3, 0.01, 0.99),
      dim = c(2,2),
      dimnames = list(M = levels(maryCalls),
                         A = levels(alarm))

#Create a list that groups all CPTs.
cpts <- list(B = cptB, E = cptE, A = cptA, J = cptJ, M = cptM)
```

Adapt the above example to calculate the CPTs of the survey variables with the correct format.

(b) Creates the survey BN from `dag` and `cpts`, by: `bn <- custom.fit(dag, cpts)`. Print the created `bn` object and verify that the probabilities of the CPTs are identical to those obtained in Exercise 1.

Note: You can query the CPT of a `variable` of the BN like this: `bn$variable`.

Which function of `bnlearn` allows you to calculate the number of parameters of local distributions? And the total number of nodes and arcs? Apply them to `bn`.

(c) This package also gives us the possibility to create the BN from the DAG and Excel data, making use of the function `bn.fit()`. The great advantage of this function is that it saves us the calculation of the CPTs.

1. Obtain a BN equivalent to the one created in section `b)`, through the command `otherBn <- bn.fit(dag, survey)`. What is the problem with creating the BN in this way? What do you think is the reason for this?

2. To solve the problem:

(i) Convert the tibble **survey** to data frame with **as.data.frame()**.

(ii) Installs and loads the **dplyr** package.

(iii) Converts all columns of **survey** from character type to factor type, using the function **mutate_if()**.

(iv) Re-execute: **otherBn <- bn.fit(dag, survey)**.

(v) Print **otherBn**, verifying that the probabilities of the CPTs are identical to those already obtained. Use the function **modelstring()** to check that the variables and relations of this BN are correct.

■

Exercise 4 Create a script in RStudio to solve the following sections:

a) In order to graph the DAGs of our BNs, we need to install and load the **Rgraphviz** package. If you try to install it from Packages » Install, you will see that it does not appear in the list. Therefore, you have to install it by executing the following code:

```
#In order to install the Rgraphviz package.
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")

BiocManager::install("Rgraphviz")
```

Once **Rgraphviz** is loaded, you can plot a DAG with the function **graphviz.plot(dag)** (or with **graphviz.plot(bn)**). Type **?graphviz.plot** at the console to investigate the input arguments of this function. Then plot **dag** in 5 different ways, varying the arguments **layout** and **shape**. Add to each represented DAG the title "DAG with layout = 'layout name' and shape = 'shape name' ".

b) Represents again **dag**, but adding to the function **graphviz.plot()** the argument **highlight**, to which a list should be assigned to modify the aesthetics of the DAG. You can run this example and see the result:

```
#DAG from Burglary Network.
dagBurglary <- model2network("[B][E][A|B:E][J|A][M|A]")

#Specify the aesthetics of the DAG.
aesthetics <- list(nodes = parents(dagBurglary, "A"),
                   arcs=matrix(c("B","A","E","A"),
                               byrow = TRUE, ncol = 2),
```

```
                  col = "darkorange2",
                 fill = "orange",
                 textCol = "white",
                 lwd = 2,
                 lty = "solid")

#Plot the DAG.
graphviz.plot(dagBurglary, layout = "dot", shape="rectangle",
              main = "Burglary Network (parents of A)",
              highlight = aesthetics)
```

Create your own aesthetic for **dag**, assigning a color for all nodes, another for all arcs and another for the text. Modify the rest of the parameters as you see fit.

Note: You can see the names of the R colors in this document: http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf.

c) In order to obtain greater flexibility when modifying the aesthetics of your DAG, use **edgeRenderInfo()**, **nodeRenderInfo()** and **renderGraph()**. Here you can see an example of use for the *Burglary Network*:

```
#Create the object "plotDag".
plotDag <- graphviz.plot(dagBurglary).

#Modify the aesthetics of the arcs B -> A and E -> A.
edgeRenderInfo(plotDag) <- list(col = c("B~A" = "darkorange2",
                                         "E~A" = "darkorange2"),
                                lwd = c("B~A" = 3, "E~A" = 3))

#We can also make modifications to the arcs later.
edgeRenderInfo(plotDag) <- list(col = c("A~J" = "dodgerblue2",
                                         "A~M" = "burlywood4"),
                               lwd = c("A~J" = 3, "A~M" = 3))

#Modify the aesthetics of nodes B, E and A.
nodeRenderInfo(plotDag) <- list(col = c("B" = "darkorange2",
                                         "E" = "darkorange2",
                                               "A" = "grey"),
                                   textCol = c("B" = "white",
                                              "E" = "white"),
                                     fill = c("B" = "orange",
                                              "E" = "orange",
```

```
                                     "A" = "antiquewhite2"))

#Continue modifying the aesthetics of the nodes.
nodeRenderInfo(plotDag) <- list(col = c("J" = "white",
                                       "M" = "white"),
                            textCol = c("J" = "white",
                                       "M" = "white"),
                         fill = c("J" = "dodgerblue3",
                                   "M" = "slategray"),
                         lty="solid",
                         fontsize = 9)

#Display the final result in the "Plots" tab of RStudio.
renderGraph(plotDag)
```

Important: If you copy and paste the above code into RStudio, be sure to replace the symbol with the correct symbol for your keyboard: $\boxed{\sim}$.

Use the above example as a basis for coloring **dag** as follows:

- Mark in one color the arcs of the path $A \rightarrow E \rightarrow R \rightarrow T$, with a dashed line stroke of thickness 3.
- Marks of another color the arcs of the path $S \rightarrow E \rightarrow O \rightarrow T$, with a continuous line stroke of thickness 2.
- Assigns a different aesthetic (border color, fill color, etc.) to each following group of nodes: $\{A, S\}$, $\{E\}$, $\{O, R\}$ y $\{T\}$

d) Finally, the function **bn.fit.barchart()** and **bn.fit.dotplot()** allow us to obtain a graphical representation of the CPTs, through a bar chart and a dot plot, respectively. From the following example, it represents all the CPTs of **bn**:

```
#Representation of the CPT of Age with bar chart
#and dot plot.
bn.fit.barchart(bn$A, main = "CPT of Age (A)",
                xlab = "P(A)", ylab = "")

bn.fit.dotplot(bn$A, main = "CPT of Age (A)", xlab = "P(A)",
               xlab = "P(A)", ylab = "")
```

■

Exercise 5 Create a script in RStudio that calculates and prints the number of parameters of the local distributions, from a vector of **nodes**, a list of **levels** and an array of **arcs**. Here you can see how the first and last part of the script would look like:

```
nodes <- c("A", "S", "E", "O", "R", "T")

levels <- list(A=c("young", "adult", "old"),
               S=c("M", "F"),
               E=c("high school", "university"),
               O=c("employee", "self-employed"),
               R=("small", "large"),
               T=c("car", "train", "other"))

arcs <- matrix(c("A", "E",
                 "S", "E",
                 "E", "O",
                 "E", "R",
                 "O", "T",
                 "R", "T"), byrow=TRUE, ncol = 2)

...

#Implementation of equation (6).

...

print(paste("Parameters of the local distributions of the BN: "
, numberParam))

#Execution of the script by console:
...
> print(paste("Parameters of the local distributions of the BN: "
, numberParam))
[1] "Parameters of the local distributions of the BN: 21"
```

Notice that if you enter the, **nodes**, **levels** and **arcs** from the *Burglary Network* example, you get 10 parameters.

■