Question 1

Avant tout, il faut installer et charger les librairies nécessaires si ce n'est pas déjà fait :

```
# Chargement des librairies
library("gRain", "gRbase", "Rgraphviz")
```

1.

Nous cherchons à produire une représentation graphique du réseau bayésien à partir des informations fournies par Claude. Ainsi, nous utiliserons des abréviations, pour chaque noeud, correspondantes aux différentes variables aléatoires renseignées :
- Revenus (Élevé, Moyen, Faibles) -> R
- Actifs (Élevé, Moyen, Faibles) -> Ac
- Ratio dettes vs revenus (Élevé, Moyen, Faible) -> DvsR
- Historique de paiement (Bon,mauvais) -> H
- Âge (moins de 25 ans, entre 25 et 50, entre 50 et 65, 65 et plus) -> Ag
- Fiabilité (Fiable, non fiable) -> Fi
- Revenus futurs (Élevés, Moyens, Faibles) -> Rf
- Solvabilité (Non solvable, Solvable) -> S

Pour la création du graphe, nous utilisons un objet graphNEL que nous appelons dag_solvency. Ce graphe est un réseau bayésien qui va contenir les différents noeuds énoncés précédemment. A l'aide des informations fournies par Claude nous pouvons déduire, notamment grâce aux point 1 à 8, les dépendances qui existent entre les noeuds. Ces dépendances se traduisent dans le graphe sous forme d'arêtes.

Par exemple pour le point 1. : "Un(e) client(e) avec un bon historique de paiement a tendance à être plus fiable". Cela se traduit par le fait que la fiabilité dépend de l'historique de paiemen. On crée donc une arête du noeud historique vers le noeud de fiabilité.

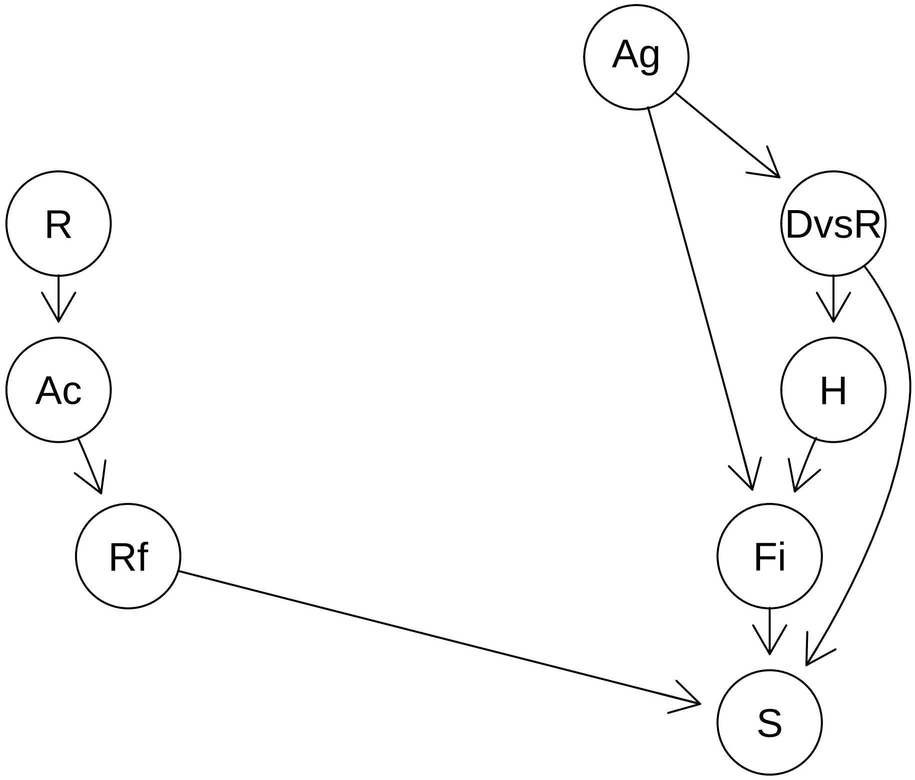
Voici le code R correspondant à la création du réseau bayésien :

```
# Création du grapheNEL
dag_solvency = dag(
  c("R"),
  c("Ac", "R"),
  c("DvsR", "Ag"),
  c("H", "DvsR"),
  c("Ag"),
  c("Fi", "H", "Ag"),
  c("Rf", "Ac"),
  c("S", "Fi", "Rf", "DvsR"),
  result = "graphNEL"
)

# Affichage des caractéristiques du graphe
dag_solvency
```

```
## A graphNEL graph with directed edges
## Number of Nodes = 8
## Number of Edges = 9
```

```
# Affichage du graphe
plot(dag_solvency)
```



2.

Nous souhaitons créer une élicitation des tables de probabilités conditionnelles associées au réseau bayésien créé précédemment. Ces tables de probabilités seront déduites des points 1 à 8, que nous avons en notre possession grâce à l'expérience de Claude.

Par exemple, en tenant compte des abréviations définies précédemment nous pouvons donc déduire des différents points les indications suivantes :
1. Un(e) client(e) avec un bon historique de paiement a tendance à être plus fiable :
P(F = Fiable | H = Bon) > P(F = Fiable | H = Mauvais)
2. Plus un(e) client(e) est âgé(e), plus il/elle a de chance d'être fiable :
P(F = Fiable | Ag = +65) > P(F = Fiable | Ag = -25)
3. Les clients plus âgés ont tendance à avoir un fiable ratio dettes vs revenu :
P(DvsR = Faible | Ag = +65) > P(DvsR = Faible | Ag = -25)
4. La probabilité d'avoir un bon historique de paiement augmente au fur et à mesure que le ratio de dette vs revenus diminue :
P(H = Bon | DvsR = Faible) > P(H = Bon | DvsR = Flevé)
5. Plus les revenus d'une personne sont élevés, plus cette personne a de chance d'avoir des actifs élevés :
P(Ac = Elevé | R = Elevé) > P(Ac = Flevé | R = Faible)
6. Plus une personne a d'actifs, plus cette personne a de chance d'avoir un revenu élevé dans le futur :
P(Rf = Elevé | Ac = Elevé) > P(Rf = Elevé | Ac = Faible)
7. Une personne fiable a tendance à être plus solvable qu'une personne non fiable :
P(S = Solvable | Fi = Fiable) > P(S = Solvable | Fi = Non fiable)
8. Les personnes qui ont des revenus prometteurs ont + de chance d'être solvables que celles dont la perspective des revenus à venir est mauvaise :
P(S = Solvable | Rf = Elevé) > P(S = Solvable | Rf = Faible)

Nous pourrons vérifier très aisément ces exemples via les tables de probabilités que nous créerons.
Pour créer ces tables nous commençons par recréer le réseau bayésien mais cette fois ci en spécifiant les probabilités associées à chaque noeud, à l'aide de la fonction cptable, qui permet de créer les tables de probabilités conditionnelles associées à chaque noeud.
Nous utilisons la fonction cptable avec comme argument le noeud sachant ses parents, les valeurs associées aux tableaux des probabilités conditionnelles à ses parents (indications colonnes par colonnes et tableau par tableau), ainsi que les valeurs possibles du noeuds.

La création d'un objet de type grain (GRAphical Independence Network) nommé grain_solvency va nous permettre à l'aide de la fonction querygrain d'afficher les tables de probabilités conditionnelles associées aux noeuds que l'on spécifie en paramètre.
C'est grâce à ces tables que l'on va pouvoir vérifier que les probabilités que l'on a défini par élicitation des points 1 à 8 sont correctes ou non.

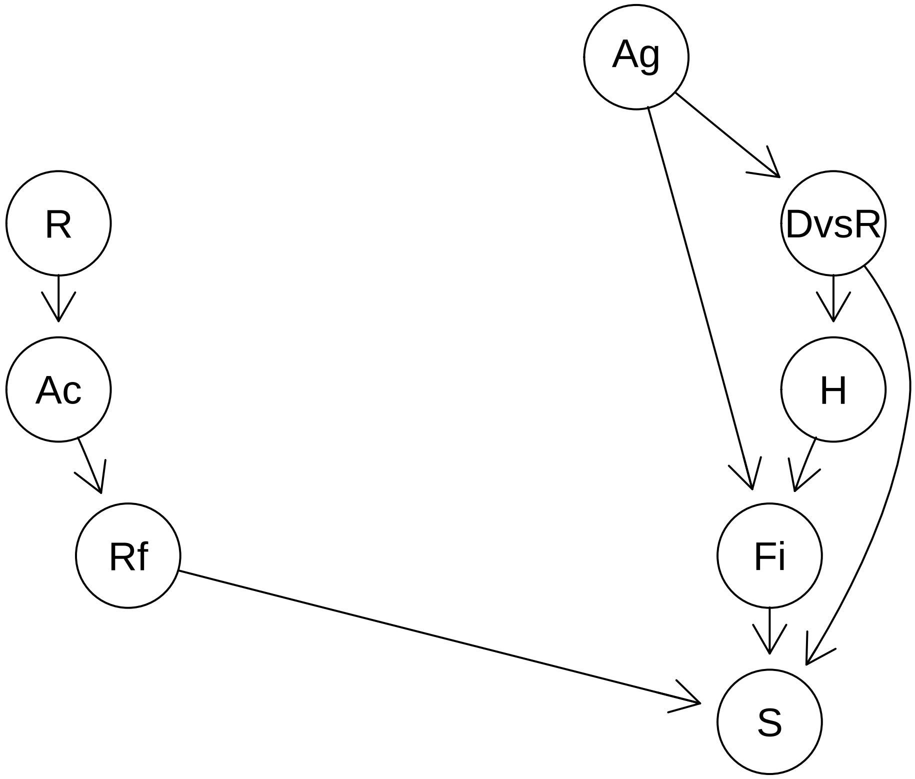
Voici le code R correspondant à l'élicitation des tables de probabilités, respectant les points 1 à 8 susmentionnés :

```
# Création du réseau bayésien en spécifiant les probabilités associés à chaque noeud
val_F_M_E = c("Faible","Moyen", "Elevé")
val_Age = c("-25","25-50", "50-65", "+65")
val_B_M = c("Mauvais", "Bon")
val_F_NF = c("Non fiable", "Fiable")
val_S_NS = c("Non Solvable", "solvable")

cp_R <- cptable(~R,
                values=c(1/3, 1/3, 1/3),
                levels=val_F_M_E)
cp_Ag <- cptable(~Ag,
                 values=c(1/4, 1/4,1/4, 1/4),
                 levels=val_Age)
cp_Ac <- cptable(~Ac|R,
                 values=c(0.8,0.15,0.05, 0.15,0.8,0.05, 0.05,0.15,0.8),
                 levels=val_F_M_E)
cp_Rf <- cptable(~Rf|Ac,
                 values=c(0.7,0.2,0.1, 0.1,0.6,0.3, 0.1,0.2,0.7),
                 levels=val_F_M_E)
cp_DvsR <- cptable(~DvsR|Ag,
                   values=c(0.4,0.3,0.3, 0.5, 0.3, 0.2, 0.6,0.3,0.1, 0.7,0.2,0.1),
                   levels=val_F_M_E)
cp_H <- cptable(~H|DvsR,
                values=c(0.1,0.9, 0.35,0.65, 0.5,0.5),
                levels=val_B_M)
cp_Fi <- cptable(~Fi|Ag+H,
                 values=c(0.6,0.4, 0.3,0.7, 0.2,0.8, 0.1,0.9,
                          0.5,0.5, 0.2,0.8, 0.1,0.9, 0,1),
                 levels=val_F_NF)
cp_S <- cptable(~S|Fi+Rf+DvsR,
                values=c(0.7,0.3, 0.5,0.5,
                         0.6,0.4, 0.4, 0.6,
                         0.5,0.5, 0.3, 0.7,
                         0.6,0.4, 0.4, 0.6,
                         0.6,0.4, 0.3, 0.7,
                         0.5,0.5, 0.2, 0.8,
                         0.7,0.3, 0.4, 0.6,
                         0.5,0.5, 0.2, 0.8,
                         0.4,0.6, 0.1, 0.9),
                levels=val_S_NS)

net_list = compileCPT(list(cp_R,cp_Ag, cp_Ac, cp_Rf, cp_DvsR, cp_H, cp_Fi, cp_S))

# Création d'un objet de type grain (GRAphical Independence Network)
grain_solvency = grain(net_list)
plot(grain_solvency$dag)
```




```
# 1.Un(e) client(e) avec un bon historique de paiement a tendance à être plus fiable :
querygrain(grain_solvency, nodes=c("Fi","H"), type="conditional")
```

```
##              H
## Fi              Mauvais       Bon
##   Non fiable 0.3303665 0.1904762
##   Fiable     0.6696335 0.8095238
```

```
# 2. Plus un(e) client(e) est âgé(e), plus il/elle a de chance d'être fiable :
querygrain(grain_solvency, nodes=c("Fi","Ag"), type="conditional")
```

```
##              Ag
## Fi                -25  25-50  50-65   +65
##   Non fiable 0.5295 0.2255 0.1215 0.019
##   Fiable     0.4705 0.7745 0.8785 0.981
```

```
# 3. Les clients plus âgés ont tendance à avoir un faible ratio dettes vs revenus :
querygrain(grain_solvency, nodes=c("DvsR","Ag"), type="conditional")
```

```
##          Ag
## DvsR      -25 25-50 50-65 +65
##   Faible 0.4   0.5   0.6 0.7
##   Moyen  0.3   0.3   0.3 0.2
##   Elevé  0.3   0.2   0.1 0.1
```

```r
# 4. La probabilité d'avoir un bon historique de paiement augmente au fur et à mesure que le ratio de d
querygrain(grain_solvency, nodes=c("H","DvsR"), type="conditional")
```

```
##          DvsR
## H         Faible Moyen Elevé
##   Mauvais    0.1  0.35   0.5
##   Bon        0.9  0.65   0.5
```

```r
# 5. Plus les revenus d'une personne sont élevés, plus cette personne a de chance d'avoir des actifs él
querygrain(grain_solvency, nodes=c("Ac","R"), type="conditional")
```

```
##         R
## Ac       Faible Moyen Elevé
##   Faible   0.80  0.15  0.05
##   Moyen    0.15  0.80  0.15
##   Elevé    0.05  0.05  0.80
```

```r
# 6. Plus une personne a d'actifs, plus cette personne a de chance d'avoir un revenu élevé dans le futu
querygrain(grain_solvency, nodes=c("Rf","Ac"), type="conditional")
```

```
##         Ac
## Rf       Faible Moyen Elevé
##   Faible    0.7   0.1   0.1
##   Moyen     0.2   0.6   0.2
##   Elevé     0.1   0.3   0.7
```

```r
# 7. Une personne fiable a tendance à être plus solvable qu'une personne non fiable :
querygrain(grain_solvency, nodes=c("S","Fi"), type="conditional")
```

```
##               Fi
## S               Non fiable   Fiable
##   Non Solvable  0.5669671 0.342742
##   solvable      0.4330329 0.657258
```

```r
# 8. Les personnes qui ont des revenus prometteurs ont + de chance d'être solvables que celles dont la p
querygrain(grain_solvency, nodes=c("S","Rf"), type="conditional")
```

```
##               Rf
## S                Faible     Moyen     Elevé
##   Non Solvable 0.50565 0.3951125 0.2951125
##   solvable     0.49435 0.6048875 0.7048875
```

On remarque, grâce aux affichages des tables de probabilités conditionnelles ci-dessus, que toutes les indications fournies par les points 1 à 8 sont respectées.