AprendizajeDeRedesBayesianas

FranciscoPérezHernández

15/3/2017

Parte 1

Vamos a aprender una red bayesiana a partir del dataset:

```
urgencias <- read.table("urgencias11v-class-ms-R.txt", sep = ",", header=TRUE)
```

El algoritmo estará basado en evaluación y búsqueda, empleando el método hc, que implementa un método de búsqueda "Hill climbing" en el espacio de los dags.

```
set.seed(1234)
bn.hc.aic <- hc(urgencias, score = "aic")
bn.hc.aic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [centro][ser_real|centro][tipo_patologia|ser_real][p10|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [horas_estancia|motivo_alta:ser_real][turno|p10:ser_real]
##    [motivo_ingr|tipo_patologia:turno:centro]
##    [tipo_docu|tipofinancia:p10:centro][dia|motivo_ingr:p10:turno]
##   nodes:                                 11
##   arcs:                                  18
##     undirected arcs:                     0
##     directed arcs:                       18
##   average markov blanket size:           4.55
##   average neighbourhood size:            3.27
##   average branching factor:              1.64
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 AIC (disc.)
##   penalization coefficient:              1
##   tests used in the learning procedure:  235
##   optimized:                             TRUE
```

```
bn.hc.bic <- hc(urgencias, score = "bic")
bn.hc.bic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [tipofinancia][tipo_patologia|tipofinancia][motivo_ingr|tipo_patologia]
##    [centro|motivo_ingr:tipo_patologia][tipo_docu|tipofinancia:centro]
##    [turno|motivo_ingr:centro][ser_real|centro][p10|tipo_docu:centro]
##    [motivo_alta|ser_real][horas_estancia|ser_real][dia|p10]
##   nodes:                                 11
##   arcs:                                  14
```

```
##     undirected arcs:                       0
##     directed arcs:                         14
##   average markov blanket size:             2.73
##   average neighbourhood size:              2.55
##   average branching factor:                1.27
##
##   learning algorithm:                      Hill-Climbing
##   score:                                   BIC (disc.)
##   penalization coefficient:                5.18576
##   tests used in the learning procedure:    195
##   optimized:                               TRUE
```

```
bn.hc.k2 <- hc(urgencias, score = "k2")
bn.hc.k2
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [ser_real][motivo_alta|ser_real][centro|ser_real][motivo_ingr|centro]
##    [horas_estancia|motivo_alta:ser_real][tipo_patologia|motivo_ingr:centro]
##    [tipofinancia|tipo_patologia][tipo_docu|tipofinancia:motivo_ingr:centro]
##    [p10|tipo_docu:ser_real]
##    [turno|motivo_ingr:tipo_patologia:tipo_docu:motivo_alta:horas_estancia:centro:ser_real]
##    [dia|motivo_ingr:tipo_patologia:p10:tipo_docu:turno:centro]
##   nodes:                                   11
##   arcs:                                    26
##     undirected arcs:                       0
##     directed arcs:                         26
##   average markov blanket size:             8.00
##   average neighbourhood size:              4.73
##   average branching factor:                2.36
##
##   learning algorithm:                      Hill-Climbing
##   score:                                   Cooper & Herskovits' K2
##   tests used in the learning procedure:    430
##   optimized:                               TRUE
```

```
bn.hc.bde.1 <- hc(urgencias, score="bde", iss = 1)
bn.hc.bde.1
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [p10][dia|p10][centro|p10][ser_real|centro][tipo_patologia|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [motivo_ingr|tipo_patologia:centro][horas_estancia|motivo_alta:ser_real]
##    [tipo_docu|tipofinancia:p10:centro][turno|motivo_ingr:p10:centro]
##   nodes:                                   11
##   arcs:                                    16
##     undirected arcs:                       0
##     directed arcs:                         16
##   average markov blanket size:             3.64
##   average neighbourhood size:              2.91
##   average branching factor:                1.45
```

```
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Bayesian Dirichlet (BDe)
##   graph prior:                           Uniform
##   imaginary sample size:                 1
##   tests used in the learning procedure:  215
##   optimized:                             TRUE
```

```
bn.hc.bde.5 <- hc(urgencias, score="bde", iss = 5)
bn.hc.bde.5
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [p10][dia|p10][centro|p10][ser_real|centro][tipo_patologia|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [horas_estancia|motivo_alta:ser_real][tipo_docu|tipofinancia:p10:centro]
##    [motivo_ingr|tipo_patologia:tipo_docu:centro]
##    [turno|motivo_ingr:p10:centro]
##   nodes:                                 11
##   arcs:                                  17
##     undirected arcs:                     0
##     directed arcs:                       17
##   average markov blanket size:           4.00
##   average neighbourhood size:            3.09
##   average branching factor:              1.55
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Bayesian Dirichlet (BDe)
##   graph prior:                           Uniform
##   imaginary sample size:                 5
##   tests used in the learning procedure:  225
##   optimized:                             TRUE
```

```
bn.hc.bde.10 <- hc(urgencias, score="bde", iss = 10)
bn.hc.bde.10
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [centro][ser_real|centro][tipo_patologia|ser_real][p10|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [motivo_ingr|tipo_patologia:centro][dia|p10]
##    [horas_estancia|motivo_alta:ser_real]
##    [tipo_docu|tipofinancia:motivo_ingr:p10:centro]
##    [turno|motivo_ingr:p10:centro]
##   nodes:                                 11
##   arcs:                                  17
##     undirected arcs:                     0
##     directed arcs:                       17
##   average markov blanket size:           4.18
##   average neighbourhood size:            3.09
##   average branching factor:              1.55
##
```

```
##   learning algorithm:                    Hill-Climbing
##   score:                                 Bayesian Dirichlet (BDe)
##   graph prior:                           Uniform
##   imaginary sample size:                 10
##   tests used in the learning procedure:  225
##   optimized:                             TRUE
```

Parte 2

Vamos a aprender una red bayesiana empleando algún algoritmo basado en test de independencia:

```
set.seed(1234)
bn.iamb <- iamb(urgencias)
```

```
## Warning in FUN(newX[, i], ...): vstructure motivo_ingr -> centro <- turno
## is not applicable, because one or both arcs are oriented in the opposite
## direction.
```

```
## Warning in FUN(newX[, i], ...): vstructure tipo_patologia -> centro <-
## turno is not applicable, because one or both arcs are oriented in the
## opposite direction.
```

```
bn.iamb
```

```
##
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  9
##     undirected arcs:                     5
##     directed arcs:                       4
##   average markov blanket size:           2.00
##   average neighbourhood size:            1.64
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Mutual Information (disc.)
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  296
##   optimized:                             TRUE
```

```
bn.iamb.mi <- iamb(urgencias, test='mi')
```

```
## Warning in FUN(newX[, i], ...): vstructure motivo_ingr -> centro <- turno
## is not applicable, because one or both arcs are oriented in the opposite
## direction.
```

```
## Warning in FUN(newX[, i], ...): vstructure tipo_patologia -> centro <-
## turno is not applicable, because one or both arcs are oriented in the
## opposite direction.
```

```
bn.iamb.mi
```

```
##
```

```
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  9
##     undirected arcs:                     5
##     directed arcs:                       4
##   average markov blanket size:           2.00
##   average neighbourhood size:            1.64
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Mutual Information (disc.)
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  296
##   optimized:                             TRUE
```

```
bn.iamb.x2 <- iamb(urgencias, test='x2')
```

```
## Warning in FUN(newX[, i], ...): vstructure motivo_ingr -> centro <- turno
## is not applicable, because one or both arcs are oriented in the opposite
## direction.

## Warning in FUN(newX[, i], ...): vstructure tipo_patologia -> centro <-
## turno is not applicable, because one or both arcs are oriented in the
## opposite direction.
```

```
bn.iamb.x2
```

```
##
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  11
##     undirected arcs:                     7
##     directed arcs:                       4
##   average markov blanket size:           2.36
##   average neighbourhood size:            2.00
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Pearson's X^2
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  299
##   optimized:                             TRUE
```

Parte 3

Vamos a comparar los resultados obtenidos por los diferentes algoritmos.

Basados en el método Hill Climbing

Vamos a empezar comparando los basados en el método Hill Climbing entre ellos.

aic vs bic

La primera comparación será entre los distintos score, primero entre aic y bic:

```
bn.hc.aic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [centro][ser_real|centro][tipo_patologia|ser_real][p10|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [horas_estancia|motivo_alta:ser_real][turno|p10:ser_real]
##    [motivo_ingr|tipo_patologia:turno:centro]
##    [tipo_docu|tipofinancia:p10:centro][dia|motivo_ingr:p10:turno]
##   nodes:                                 11
##   arcs:                                  18
##     undirected arcs:                     0
##     directed arcs:                       18
##   average markov blanket size:           4.55
##   average neighbourhood size:            3.27
##   average branching factor:              1.64
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 AIC (disc.)
##   penalization coefficient:              1
##   tests used in the learning procedure:  235
##   optimized:                             TRUE
```

```
bn.hc.bic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [tipofinancia][tipo_patologia|tipofinancia][motivo_ingr|tipo_patologia]
##    [centro|motivo_ingr:tipo_patologia][tipo_docu|tipofinancia:centro]
##    [turno|motivo_ingr:centro][ser_real|centro][p10|tipo_docu:centro]
##    [motivo_alta|ser_real][horas_estancia|ser_real][dia|p10]
##   nodes:                                 11
##   arcs:                                  14
##     undirected arcs:                     0
##     directed arcs:                       14
##   average markov blanket size:           2.73
##   average neighbourhood size:            2.55
##   average branching factor:              1.27
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 BIC (disc.)
##   penalization coefficient:              5.18576
##   tests used in the learning procedure:  195
```

```
##   optimized:                          TRUE
```

Vemos como el basado en aic tiene 18 arcos frente a los 14 del basado en bic, viendo también las diferencias entre las medias. El coeficiente de penalización es bastante más alto en bic que en aic, y los tests usados en el procedimiento de aprendizaje son más en aic. Ahora vamos a ver si tienen la misma estructura de red.

```
bnlearn::compare(bn.hc.aic, bn.hc.bic)
```

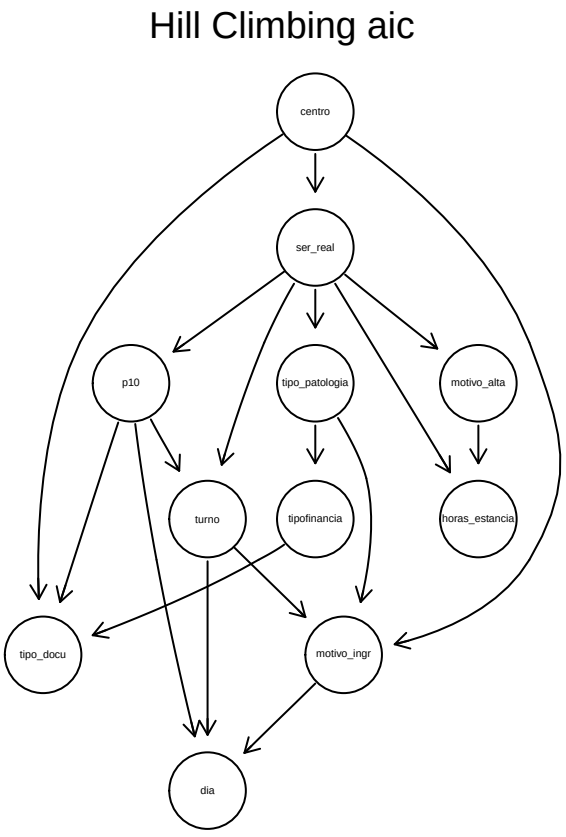
```
## $tp
## [1] 7
##
## $fp
## [1] 7
##
## $fn
## [1] 11
```

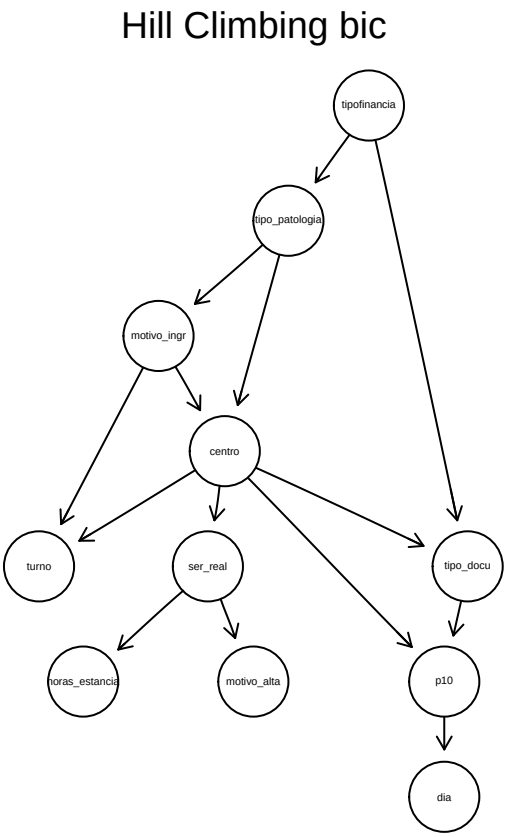
Podemos ver como los verdades positivos (el número de arcos presentes en ambos) son 7, frente a los falsos positivos (el número de arcos no presentes en ambos) que son 7 también, siendo los falsos negativos (el número de arcos no presentes al inicio pero si al final) son 11. Por lo tanto veamos los modelos:

```
par(mfrow = c(1,2))
graphviz.plot(bn.hc.aic, main="Hill Climbing aic")
```

```
## Loading required namespace: Rgraphviz
```

```
graphviz.plot(bn.hc.bic, main="Hill Climbing bic")
```





Donde podemos ver que las redes son bastante distintas a simple vista. Por ejemplo "centro" en el método aic, tiene por debajo a "tipo_docu", "ser_real" y "motivo_ingr", mientras que en el método bic, tiene por debajo a "ser_real", "turno", "tipo_docu" y "p10". Es decir, hay bastantes diferencias. Además, "centro" en el método

aic no tiene nada por encima, mientras que en bic viene de “motivo_ingr” y “tipo_patologia”.

aic vs k2

Vamos a comparar aic y k2:

```
bn.hc.aic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [centro][ser_real|centro][tipo_patologia|ser_real][p10|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [horas_estancia|motivo_alta:ser_real][turno|p10:ser_real]
##    [motivo_ingr|tipo_patologia:turno:centro]
##    [tipo_docu|tipofinancia:p10:centro][dia|motivo_ingr:p10:turno]
##   nodes:                                 11
##   arcs:                                  18
##     undirected arcs:                     0
##     directed arcs:                       18
##   average markov blanket size:           4.55
##   average neighbourhood size:            3.27
##   average branching factor:              1.64
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 AIC (disc.)
##   penalization coefficient:              1
##   tests used in the learning procedure:  235
##   optimized:                             TRUE
```

```
bn.hc.k2
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [ser_real][motivo_alta|ser_real][centro|ser_real][motivo_ingr|centro]
##    [horas_estancia|motivo_alta:ser_real][tipo_patologia|motivo_ingr:centro]
##    [tipofinancia|tipo_patologia][tipo_docu|tipofinancia:motivo_ingr:centro]
##    [p10|tipo_docu:ser_real]
##    [turno|motivo_ingr:tipo_patologia:tipo_docu:motivo_alta:horas_estancia:centro:ser_real]
##    [dia|motivo_ingr:tipo_patologia:p10:tipo_docu:turno:centro]
##   nodes:                                 11
##   arcs:                                  26
##     undirected arcs:                     0
##     directed arcs:                       26
##   average markov blanket size:           8.00
##   average neighbourhood size:            4.73
##   average branching factor:              2.36
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Cooper & Herskovits' K2
##   tests used in the learning procedure:  430
##   optimized:                             TRUE
```

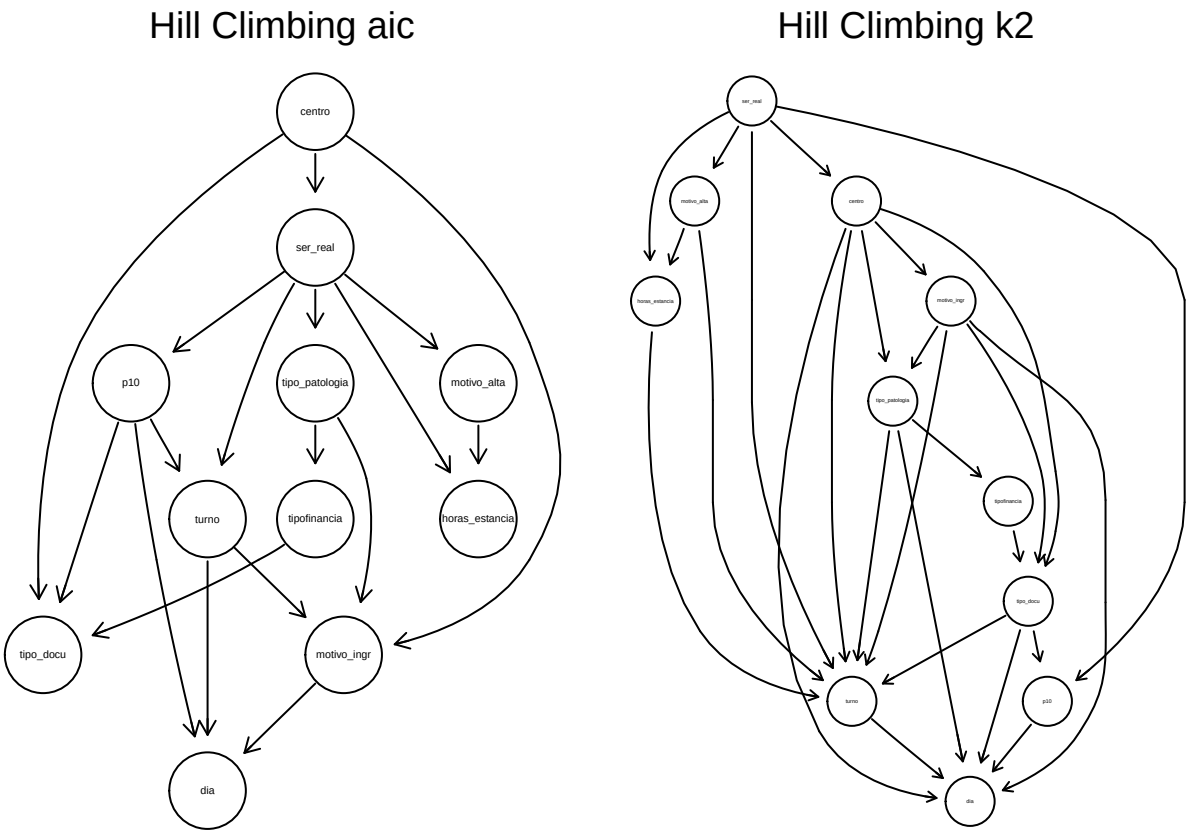
Vemos como el basado en aic tiene 18 arcos frente a los 26 del basado en k2, viendo también las diferencias entre las medias. Los tests usados en el procedimiento de aprendizaje son más en k2 Ahora vamos a ver si tienen la misma estructura de red.

```
bnlearn::compare(bn.hc.aic, bn.hc.k2)
```

```
## $tp
## [1] 12
##
## $fp
## [1] 14
##
## $fn
## [1] 6
```

Podemos ver como los verdades positivos (el número de arcos presentes en ambos) son 12, frente a los falsos positivos (el número de arcos no presentes en ambos) que son 14 también, siendo los falsos negativos (el número de arcos no presentes al inicio pero si al final) son 6. Por lo tanto veamos los modelos:

```
par(mfrow = c(1,2))
graphviz.plot(bn.hc.aic, main="Hill Climbing aic")
graphviz.plot(bn.hc.k2, main="Hill Climbing k2")
```



Donde podemos ver que las redes son bastante distintas a simple vista.

bic vs k2

Vamos a comparar bic y k2:

```
bn.hc.bic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [tipofinancia][tipo_patologia|tipofinancia][motivo_ingr|tipo_patologia]
##    [centro|motivo_ingr:tipo_patologia][tipo_docu|tipofinancia:centro]
##    [turno|motivo_ingr:centro][ser_real|centro][p10|tipo_docu:centro]
##    [motivo_alta|ser_real][horas_estancia|ser_real][dia|p10]
##   nodes:                                 11
##   arcs:                                  14
##     undirected arcs:                     0
##     directed arcs:                       14
##   average markov blanket size:           2.73
##   average neighbourhood size:            2.55
##   average branching factor:              1.27
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 BIC (disc.)
##   penalization coefficient:              5.18576
##   tests used in the learning procedure:  195
##   optimized:                             TRUE
```

```
bn.hc.k2
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [ser_real][motivo_alta|ser_real][centro|ser_real][motivo_ingr|centro]
##    [horas_estancia|motivo_alta:ser_real][tipo_patologia|motivo_ingr:centro]
##    [tipofinancia|tipo_patologia][tipo_docu|tipofinancia:motivo_ingr:centro]
##    [p10|tipo_docu:ser_real]
##    [turno|motivo_ingr:tipo_patologia:tipo_docu:motivo_alta:horas_estancia:centro:ser_real]
##    [dia|motivo_ingr:tipo_patologia:p10:tipo_docu:turno:centro]
##   nodes:                                 11
##   arcs:                                  26
##     undirected arcs:                     0
##     directed arcs:                       26
##   average markov blanket size:           8.00
##   average neighbourhood size:            4.73
##   average branching factor:              2.36
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Cooper & Herskovits' K2
##   tests used in the learning procedure:  430
##   optimized:                             TRUE
```

Vemos como el basado en bic tiene 14 arcos frente a los 26 del basado en k2, viendo también las diferencias entre las medias. Los tests usados en el procedimiento de aprendizaje son más en k2 Ahora vamos a ver si tienen la misma estructura de red.

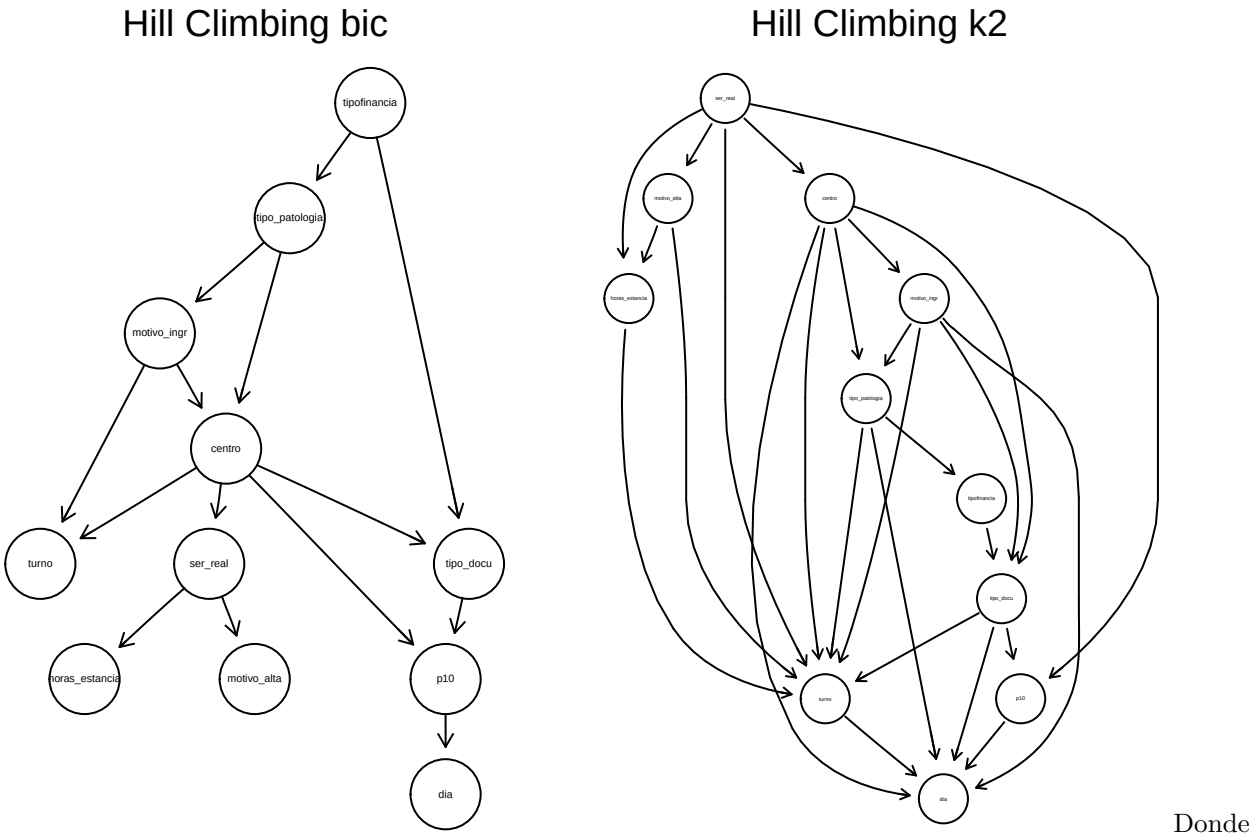
```
bnlearn::compare(bn.hc.bic, bn.hc.k2)
```

```
## $tp
```

```
## [1] 8
##
## $fp
## [1] 18
##
## $fn
## [1] 6
```

Podemos ver como los verdades positivos (el número de arcos presentes en ambos) son 8, frente a los falsos positivos (el número de arcos no presentes en ambos) que son 18 también, siendo los falsos negativos (el número de arcos no presentes al inicio pero si al final) son 6. Por lo tanto veamos los modelos:

```
par(mfrow = c(1,2))
graphviz.plot(bn.hc.bic, main="Hill Climbing bic")
graphviz.plot(bn.hc.k2, main="Hill Climbing k2")
```



Donde podemos ver que las redes son bastante distintas a simple vista.

Distintos bde

Vamos a comparar los 3 diferentes modelos con bde que he realizado:

```
bn.hc.bde.1
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [p10][dia|p10][centro|p10][ser_real|centro][tipo_patologia|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
```

```
##    [motivo_ingr|tipo_patologia:centro][horas_estancia|motivo_alta:ser_real]
##    [tipo_docu|tipofinancia:p10:centro][turno|motivo_ingr:p10:centro]
##   nodes:                                 11
##   arcs:                                  16
##     undirected arcs:                     0
##     directed arcs:                       16
##   average markov blanket size:           3.64
##   average neighbourhood size:            2.91
##   average branching factor:              1.45
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Bayesian Dirichlet (BDe)
##   graph prior:                           Uniform
##   imaginary sample size:                 1
##   tests used in the learning procedure:  215
##   optimized:                             TRUE
```

```
bn.hc.bde.5
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [p10][dia|p10][centro|p10][ser_real|centro][tipo_patologia|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [horas_estancia|motivo_alta:ser_real][tipo_docu|tipofinancia:p10:centro]
##    [motivo_ingr|tipo_patologia:tipo_docu:centro]
##    [turno|motivo_ingr:p10:centro]
##   nodes:                                 11
##   arcs:                                  17
##     undirected arcs:                     0
##     directed arcs:                       17
##   average markov blanket size:           4.00
##   average neighbourhood size:            3.09
##   average branching factor:              1.55
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Bayesian Dirichlet (BDe)
##   graph prior:                           Uniform
##   imaginary sample size:                 5
##   tests used in the learning procedure:  225
##   optimized:                             TRUE
```

```
bn.hc.bde.10
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [centro][ser_real|centro][tipo_patologia|ser_real][p10|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [motivo_ingr|tipo_patologia:centro][dia|p10]
##    [horas_estancia|motivo_alta:ser_real]
##    [tipo_docu|tipofinancia:motivo_ingr:p10:centro]
##    [turno|motivo_ingr:p10:centro]
##   nodes:                                 11
```

```
##   arcs:                                 17
##     undirected arcs:                    0
##     directed arcs:                      17
##   average markov blanket size:          4.18
##   average neighbourhood size:           3.09
##   average branching factor:             1.55
##
##   learning algorithm:                   Hill-Climbing
##   score:                                Bayesian Dirichlet (BDe)
##   graph prior:                          Uniform
##   imaginary sample size:                10
##   tests used in the learning procedure: 225
##   optimized:                            TRUE
```

Vemos como el número de arcos para 1 es de 16 mientras que para 5 y 10 son 17. Además ocurre lo mismo con el número de test usados en el procedimiento de aprendizaje. Ahora vamos a ver si tienen la misma estructura de red.

```
bnlearn::compare(bn.hc.bde.1, bn.hc.bde.5)
```

```
## $tp
## [1] 16
##
## $fp
## [1] 1
##
## $fn
## [1] 0
```

```
bnlearn::compare(bn.hc.bde.1, bn.hc.bde.10)
```

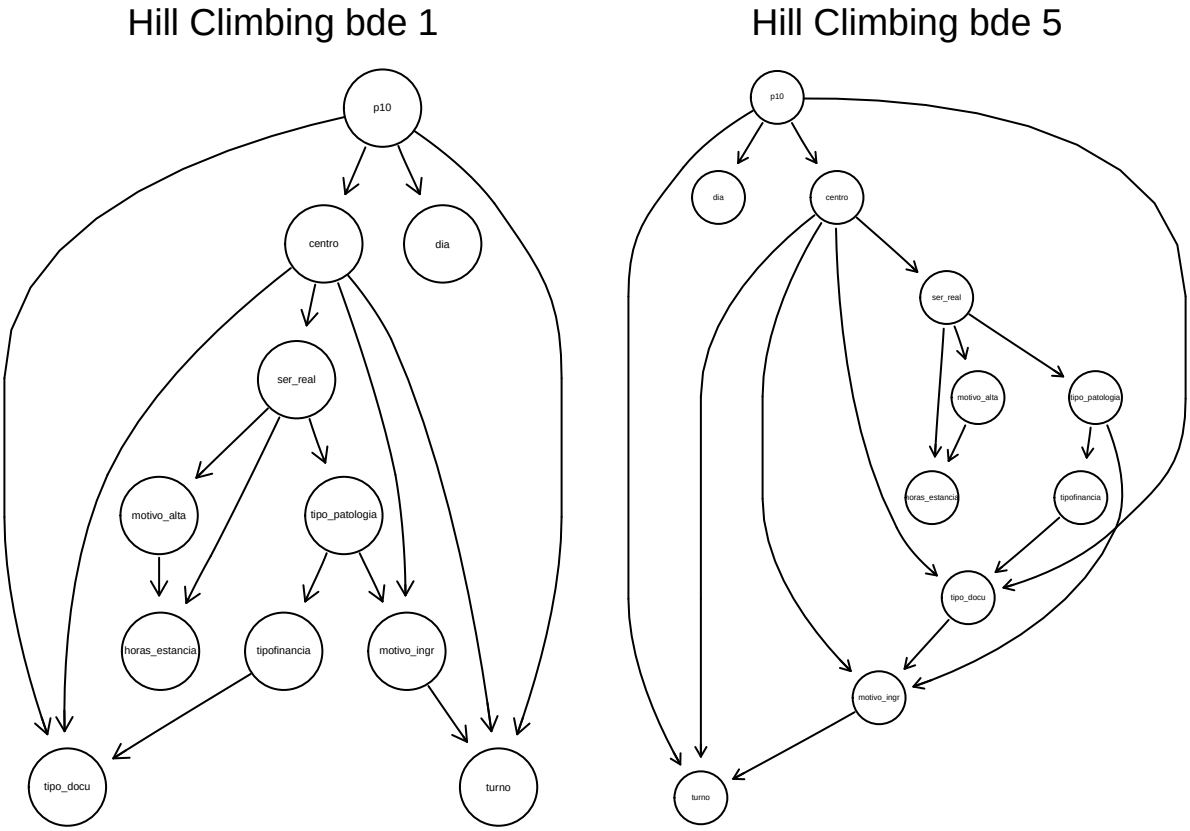
```
## $tp
## [1] 15
##
## $fp
## [1] 2
##
## $fn
## [1] 1
```

```
bnlearn::compare(bn.hc.bde.5, bn.hc.bde.10)
```

```
## $tp
## [1] 15
##
## $fp
## [1] 2
##
## $fn
## [1] 2
```

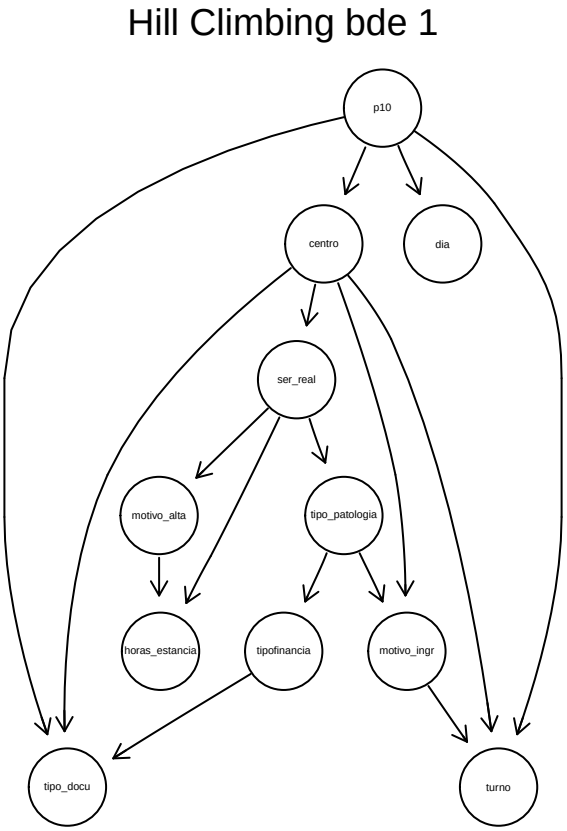
Donde podemos ver que siempre hay alguna diferencia. Vamos a ver los modelos:

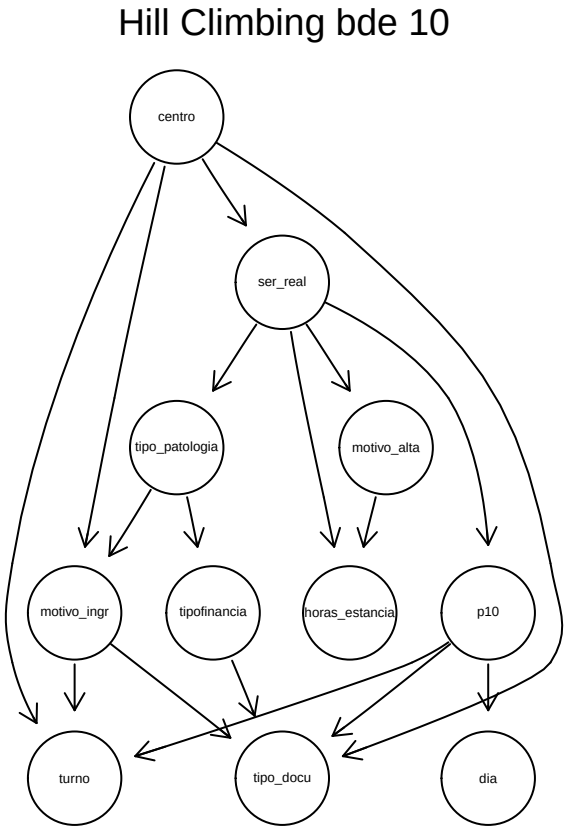
```
par(mfrow = c(1,2))
graphviz.plot(bn.hc.bde.1, main="Hill Climbing bde 1")
graphviz.plot(bn.hc.bde.5, main="Hill Climbing bde 5")
```



Donde podemos ver que salvo en "tipo_docu" en la que hay algunos arcos diferentes, el resto es muy parecido.

```
par(mfrow = c(1,2))
graphviz.plot(bn.hc.bde.1, main="Hill Climbing bde 1")
graphviz.plot(bn.hc.bde.10, main="Hill Climbing bde 10")
```





Vemos como son bastante parecidas pero siguen sin ser tan parecidas como 1 y 5.

aic vs bde

Vamos a comparar aic y bde:

```
bn.hc.aic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [centro][ser_real|centro][tipo_patologia|ser_real][p10|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [horas_estancia|motivo_alta:ser_real][turno|p10:ser_real]
##    [motivo_ingr|tipo_patologia:turno:centro]
##    [tipo_docu|tipofinancia:p10:centro][dia|motivo_ingr:p10:turno]
##   nodes:                                 11
##   arcs:                                  18
##     undirected arcs:                     0
##     directed arcs:                       18
##   average markov blanket size:           4.55
##   average neighbourhood size:            3.27
##   average branching factor:              1.64
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 AIC (disc.)
##   penalization coefficient:              1
```

```
##   tests used in the learning procedure:  235
##   optimized:                             TRUE
```

```
bn.hc.bde.1
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [p10][dia|p10][centro|p10][ser_real|centro][tipo_patologia|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [motivo_ingr|tipo_patologia:centro][horas_estancia|motivo_alta:ser_real]
##    [tipo_docu|tipofinancia:p10:centro][turno|motivo_ingr:p10:centro]
##   nodes:                                 11
##   arcs:                                  16
##     undirected arcs:                     0
##     directed arcs:                       16
##   average markov blanket size:           3.64
##   average neighbourhood size:            2.91
##   average branching factor:              1.45
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Bayesian Dirichlet (BDe)
##   graph prior:                           Uniform
##   imaginary sample size:                 1
##   tests used in the learning procedure:  215
##   optimized:                             TRUE
```

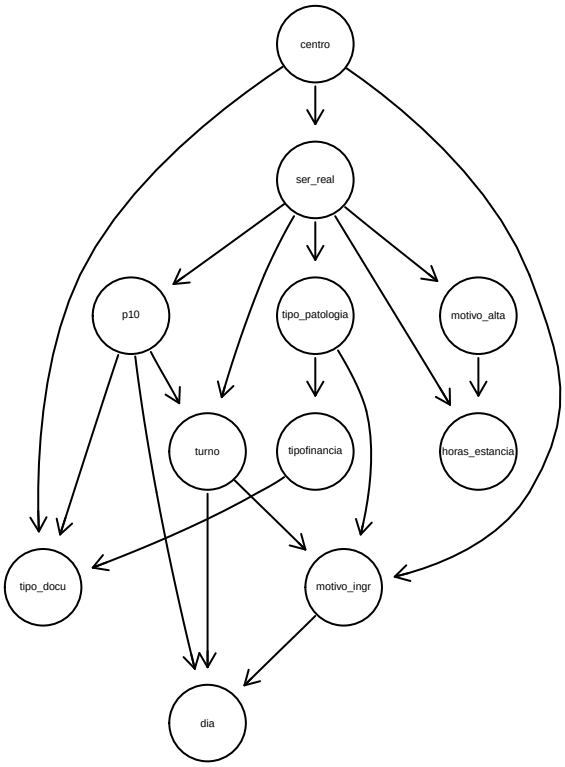
Vemos como el basado en aic tiene 18 arcos frente a los 16 del basado en bde.1, viendo también las diferencias entre las medias. Los tests usados en el procedimiento de aprendizaje son más en aic Ahora vamos a ver si tienen la misma estructura de red.

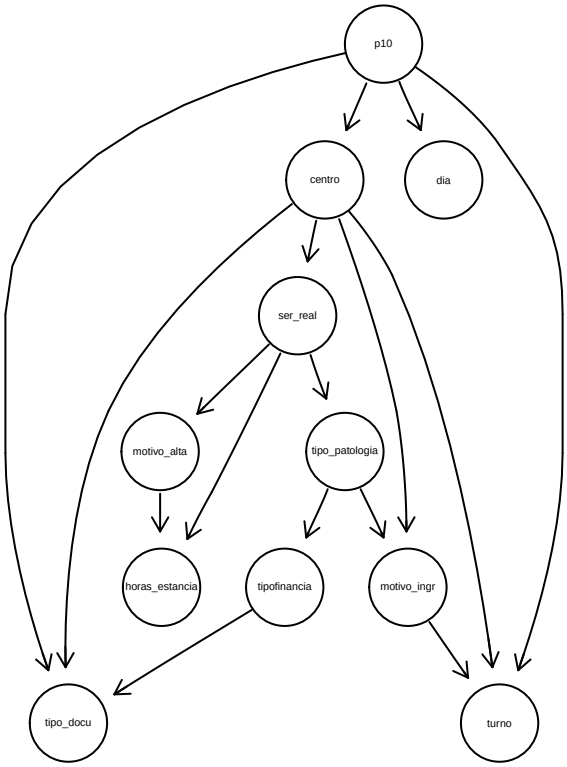
```
bnlearn::compare(bn.hc.aic, bn.hc.bde.1)
```

```
## $tp
## [1] 13
##
## $fp
## [1] 3
##
## $fn
## [1] 5
```

Podemos ver como los verdades positivos (el número de arcos presentes en ambos) son 13, frente a los falsos positivos (el número de arcos no presentes en ambos) que son 3 también, siendo los falsos negativos (el número de arcos no presentes al inicio pero si al final) son 5. Por lo tanto veamos los modelos:

```
par(mfrow = c(1,2))
graphviz.plot(bn.hc.aic, main="Hill Climbing aic")
graphviz.plot(bn.hc.bde.1, main="Hill Climbing bde 1")
```





Donde podemos ver las diferencias entre ambos.

Basados en test de independencia

mi vs x2

Vamos a comparar estos dos test:

```
bn.iamb.mi
```

```
##
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  9
##     undirected arcs:                     5
##     directed arcs:                       4
##   average markov blanket size:           2.00
##   average neighbourhood size:            1.64
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Mutual Information (disc.)
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  296
##   optimized:                             TRUE
```

```
bn.iamb.x2
```

```
##
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  11
##     undirected arcs:                     7
##     directed arcs:                       4
##   average markov blanket size:           2.36
##   average neighbourhood size:            2.00
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Pearson's X^2
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  299
##   optimized:                             TRUE
```

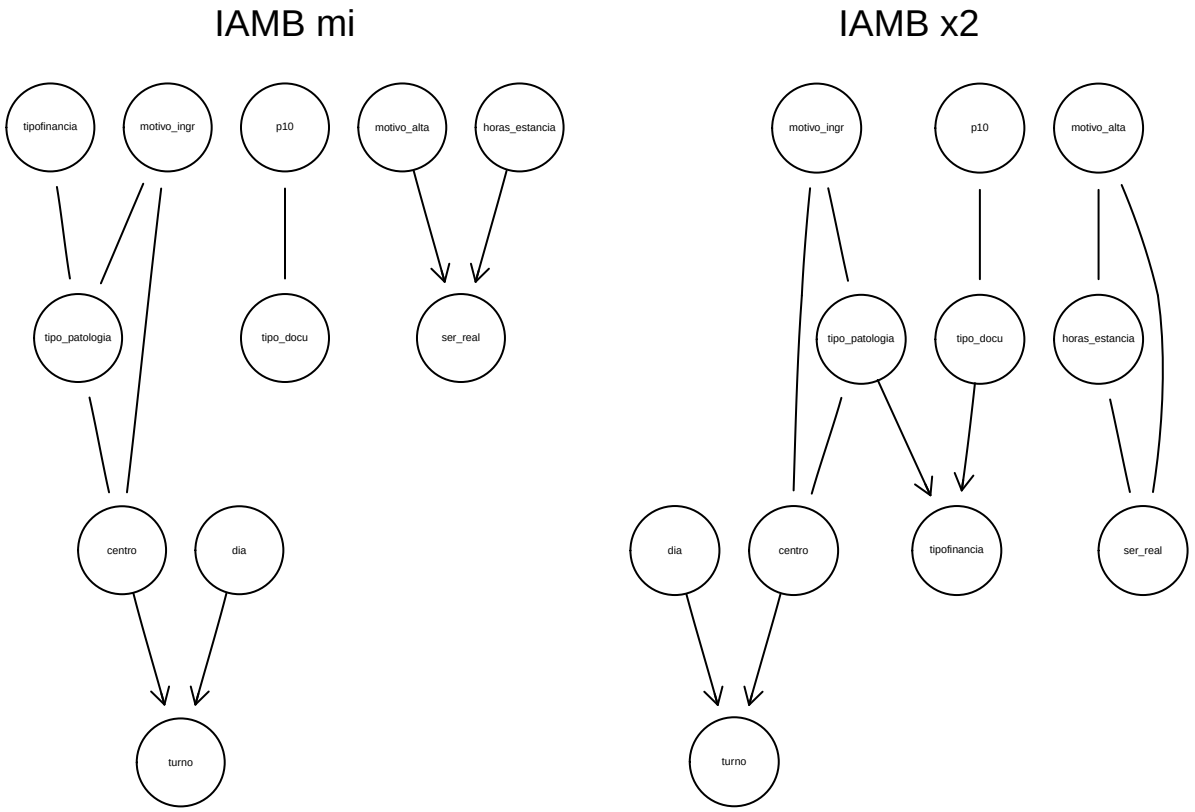
Vemos como el basado en mi tiene 9 arcos frente a los 11 del basado en x2, donde hay arcos directos e indirectos coincidiendo en el número de directos. Los tests usados en el procedimiento de aprendizaje son más en x2. Ahora vamos a ver si tienen la misma estructura de red.

```
bnlearn::compare(bn.iamb.mi, bn.iamb.x2)
```

```
## $tp
## [1] 6
##
## $fp
## [1] 5
##
## $fn
## [1] 3
```

Podemos ver como los verdades positivos (el número de arcos presentes en ambos) son 6, frente a los falsos positivos (el número de arcos no presentes en ambos) que son 5 también, siendo los falsos negativos (el número de arcos no presentes al inicio pero si al final) son 3. Por lo tanto veamos los modelos:

```
par(mfrow = c(1,2))
graphviz.plot(bn.iamb.mi, main="IAMB mi")
graphviz.plot(bn.iamb.x2, main="IAMB x2")
```



Donde podemos ver las diferencias entre ambos y algunas semejanzas.

mi

Vamos a comparar estos dos test:

```
bn.iamb.mi
```

```
##
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  9
##     undirected arcs:                     5
##     directed arcs:                       4
##   average markov blanket size:           2.00
##   average neighbourhood size:            1.64
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Mutual Information (disc.)
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  296
##   optimized:                             TRUE
```

```
bn.iamb
```

```
##
```

```
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  9
##     undirected arcs:                     5
##     directed arcs:                       4
##   average markov blanket size:           2.00
##   average neighbourhood size:            1.64
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Mutual Information (disc.)
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  296
##   optimized:                             TRUE
```

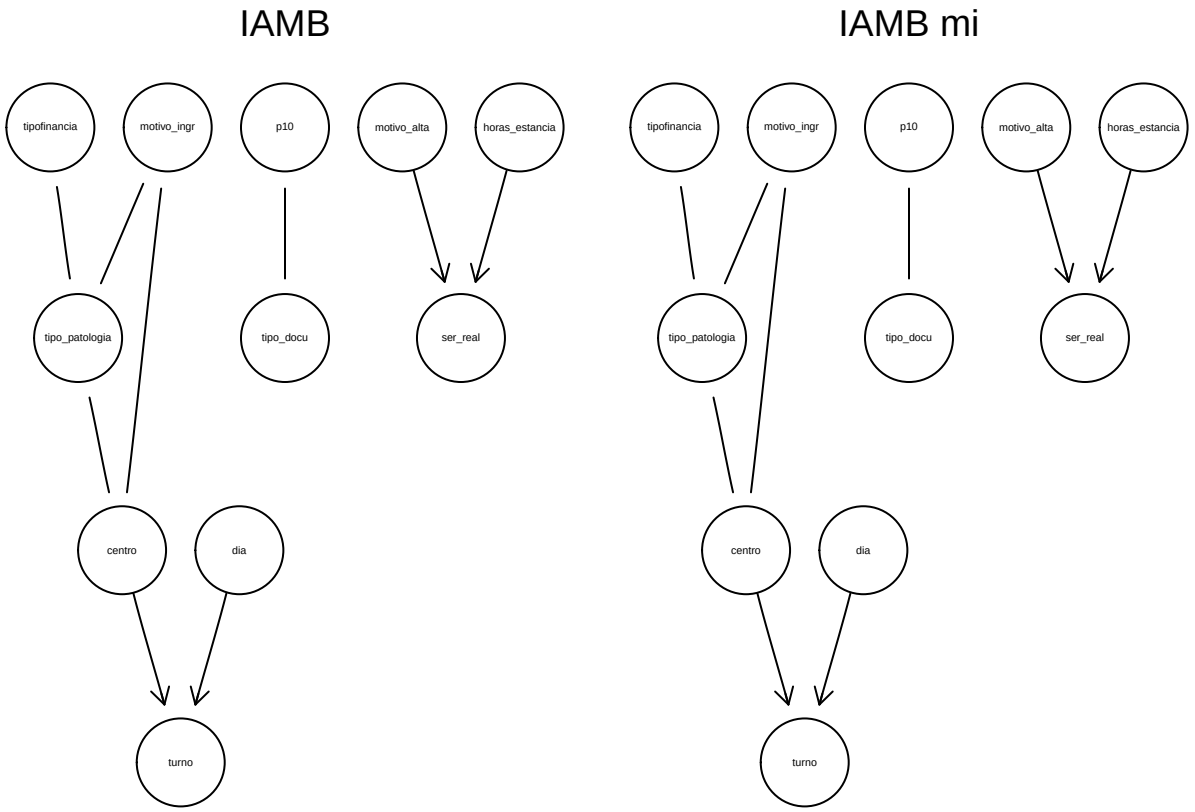
Donde vemos que dan los mismos resultados si usamos test=NULL o test='mi'

```
bnlearn::compare(bn.iamb, bn.iamb.mi)
```

```
## $tp
## [1] 9
##
## $fp
## [1] 0
##
## $fn
## [1] 0
```

Veamos los modelos:

```
par(mfrow = c(1,2))
graphviz.plot(bn.iamb, main="IAMB")
graphviz.plot(bn.iamb.mi, main="IAMB mi")
```



Donde vemos que son iguales.

Hill Climbing vs Basados en test de independencia

aic vs x2

Vamos a comparar estos dos test:

```
bn.hc.aic
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [centro][ser_real|centro][tipo_patologia|ser_real][p10|ser_real]
##    [motivo_alta|ser_real][tipofinancia|tipo_patologia]
##    [horas_estancia|motivo_alta:ser_real][turno|p10:ser_real]
##    [motivo_ingr|tipo_patologia:turno:centro]
##    [tipo_docu|tipofinancia:p10:centro][dia|motivo_ingr:p10:turno]
##   nodes:                                 11
##   arcs:                                  18
##     undirected arcs:                     0
##     directed arcs:                       18
##   average markov blanket size:           4.55
##   average neighbourhood size:            3.27
##   average branching factor:              1.64
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 AIC (disc.)
```

```
##   penalization coefficient:              1
##   tests used in the learning procedure:  235
##   optimized:                             TRUE
```

```
bn.iamb.x2
```

```
##
##   Bayesian network learned via Constraint-based methods
##
##   model:
##     [partially directed graph]
##   nodes:                                 11
##   arcs:                                  11
##     undirected arcs:                     7
##     directed arcs:                       4
##   average markov blanket size:           2.36
##   average neighbourhood size:            2.00
##   average branching factor:              0.36
##
##   learning algorithm:                    IAMB
##   conditional independence test:         Pearson's X^2
##   alpha threshold:                       0.05
##   tests used in the learning procedure:  299
##   optimized:                             TRUE
```

Vemos como el basado en aic tiene 18 arcos, todos directos, frente a los 11 del basado en x2, donde hay arcos directos e indirectos. Los tests usados en el procedimiento de aprendizaje son más en x2. Ahora vamos a ver si tienen la misma estructura de red.

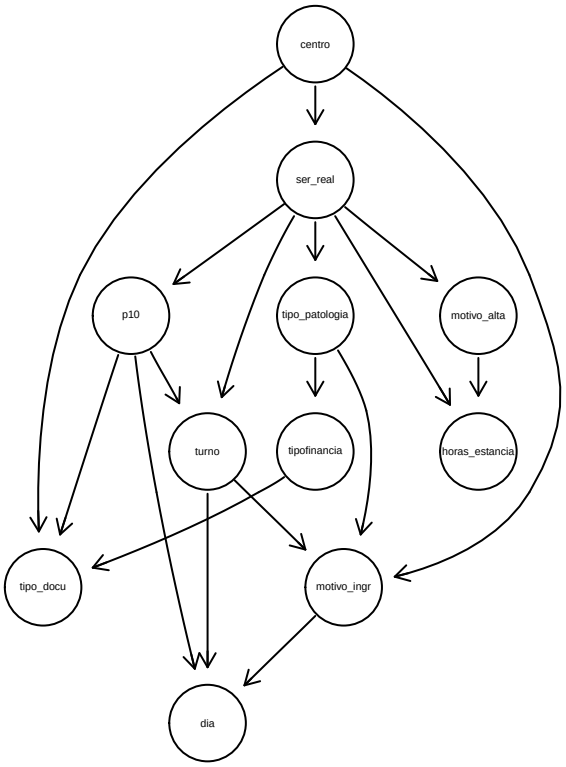
```
bnlearn::compare(bn.hc.aic, bn.iamb.x2)
```

```
## $tp
## [1] 1
##
## $fp
## [1] 10
##
## $fn
## [1] 17
```

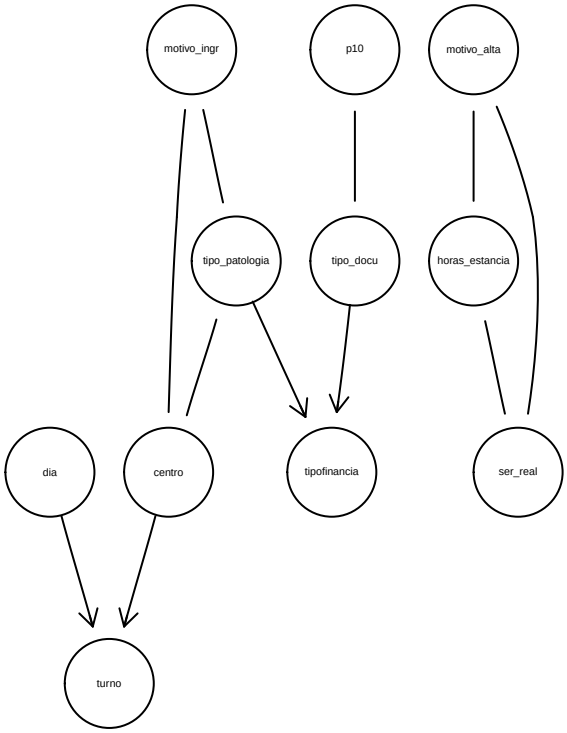
Podemos ver como los verdades positivos (el número de arcos presentes en ambos) son 1, frente a los falsos positivos (el número de arcos no presentes en ambos) que son 10 también, siendo los falsos negativos (el número de arcos no presentes al inicio pero si al final) son 17. Por lo tanto veamos los modelos:

```
par(mfrow = c(1,2))
graphviz.plot(bn.hc.aic, main="Hill Climbing aic")
graphviz.plot(bn.iamb.x2, main="Test de independincia IAMB x2")
```

Hill Climbing aic



Test de independincia IAMB x2



Donde podemos ver que son bastante diferentes.

Conclusiones

Hemos visto diferentes métodos como Hill Climbing y basados en test de independencia, y comparado entre ellos. Hemos podido ver sus modelos, y como en muchos casos las relaciones parecen tener sentido a simple vista, ya que por ejemplo el turno influirá en el “motivo_ingr” y del “dia”.