In [2]:

```
#install.packages('bnlearn')
#install.packages('visNetwork')
```

In [3]:

```
#source("http://bioconductor.org/biocLite.R")
#biocLite(c("graph", "RBGL", "Rgraphviz"))
```

In [4]:

```
#install.packages("gRain")
```

In [1]:

```
library('visNetwork')
library('Rgraphviz')
library('gRain')
library('compiler')
library('bnlearn')
```

```
Loading required package: graph
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: ‘BiocGenerics’

The following objects are masked from ‘package:parallel’:

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

The following objects are masked from ‘package:stats’:

    IQR, mad, sd, var, xtabs

The following objects are masked from ‘package:base’:

    anyDuplicated, append, as.data.frame, cbind, colMeans, colnames,
    colSums, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, lengths, Map, mapply, match,
    mget, order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
    table, tapply, union, unique, unsplit, which, which.max, which.min

Loading required package: grid
Loading required package: gRbase
```

```
Attaching package: ‘compiler’

The following object is masked from ‘package:gRbase’:

    compile


Attaching package: ‘bnlearn’

The following objects are masked from ‘package:gRbase’:

    ancestors, children, parents

The following object is masked from ‘package:BiocGenerics’:

    score

The following object is masked from ‘package:stats’:

    sigma
```

Задача 1.

(8 баллов) Рассмотрим датасет о клиентах банка из предыдущих домашних заданий http://archive.ics.uci.edu/ml/datasets/Bank+Marketing (http://archive.ics.uci.edu/ml/datasets/Bank+Marketing)

(a) По данным постройте граф причинно-следственных связей. Рекомендуется использовать пакет bnlearn в R, описание которого есть на странице курса, однако при желании вы можете использовать другие средства, в том числе на Питоне. Для построения графа можете ограничиться только категориальными признаками и целевой меткой. Помните, что методы построения графа требуют больших выборок (десятки тысяч наблюдений), поэтому используйте файл с полными данными.

(b) Визуализируйте граф. Сколько в нем коллайдров?

(c) Какие выводы можно сделать о причинно-следственных связях влияния признаков на целевую метку? Для ответа на вопрос нужно рассмотреть ребра, присоединенные к вершине, соответствующей целевой метке.

(d) Для некоторых бинарных признаков посчитайте оценку причинно-следственного эффекта с помощью интервенции. Это придется реализовать самостоятельно.

(e) Сделайте выводы. Сравните полученные результаты с результатами из предыдущих заданий.

In [32]:

```
library(caret)
df <- read.table("bank-full.csv", sep=';', header=TRUE)
df <- subset(df, select = -c(duration, age, balance, day, campaign, pdays, previous))
# удалим duration так как сильная корреляция с target, оставим только категориальные при
```

In [33]:

```
options(repr.plot.width = 15, repr.plot.height = 15)
```

In [34]:

```
head(df)
```

job	marital	education	default	housing	loan	contact	month	poutcome	y
management	married	tertiary	no	yes	no	unknown	may	unknown	no
technician	single	secondary	no	yes	no	unknown	may	unknown	no
entrepreneur	married	secondary	no	yes	yes	unknown	may	unknown	no
blue-collar	married	unknown	no	yes	no	unknown	may	unknown	no
unknown	single	unknown	no	no	no	unknown	may	unknown	no
management	married	tertiary	no	yes	no	unknown	may	unknown	no

In [43]:

```
?iamb
```

In [56]:

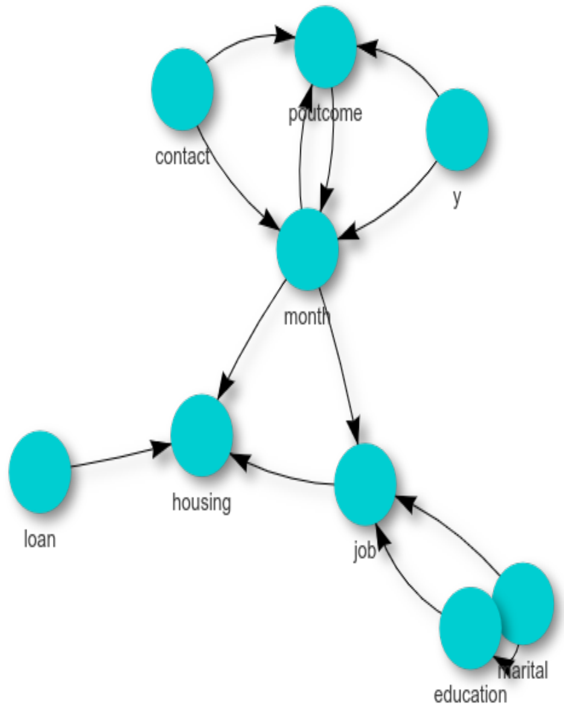
```
bl <- matrix(c("y", "month",
               "y", "poutcome",
               'contact', 'month',
               'poutcome', 'month'), ncol = 2, byrow = TRUE)
bn.iamb <- iamb(df)
bn.iamb
```

```
Warning message in vstruct.apply(arcs = arcs, vs = vs, nodes = nodes, strict = strict,
:
“vstructure housing -> month <- poutcome is not applicable, because one or both arcs ar
e oriented in the opposite direction.”Warning message in vstruct.apply(arcs = arcs, vs
= vs, nodes = nodes, strict = strict, :
“vstructure housing -> month <- y is not applicable, because one or both arcs are orien
ted in the opposite direction.”Warning message in vstruct.apply(arcs = arcs, vs = vs, n
odes = nodes, strict = strict, :
“vstructure housing -> month <- contact is not applicable, because one or both arcs are
oriented in the opposite direction.”Warning message in vstruct.apply(arcs = arcs, vs =
vs, nodes = nodes, strict = strict, :
“vstructure job -> month <- contact is not applicable, because one or both arcs are ori
ented in the opposite direction.”Warning message in vstruct.apply(arcs = arcs, vs = vs,
nodes = nodes, strict = strict, :
“vstructure education -> job <- housing is not applicable, because one or both arcs are
oriented in the opposite direction.”Warning message in vstruct.apply(arcs = arcs, vs =
vs, nodes = nodes, strict = strict, :
“vstructure job -> month <- y is not applicable, because one or both arcs are oriented
in the opposite direction.”Warning message in vstruct.apply(arcs = arcs, vs = vs, nodes
= nodes, strict = strict, :
“vstructure marital -> job <- housing is not applicable, because one or both arcs are o
riented in the opposite direction.”


  Bayesian network learned via Constraint-based methods

  model:
    [partially directed graph]
  nodes:                                 10
  arcs:                                  12
    undirected arcs:                     2
    directed arcs:                       10
  average markov blanket size:           3.40
  average neighbourhood size:            2.40
  average branching factor:              1.00

  learning algorithm:                    IAMB
  conditional independence test:         Mutual Information (disc.)
  alpha threshold:                       0.05
  tests used in the learning procedure:  524
  optimized:                             FALSE
```

In [36]:

```r
plot.network <- function(structure, ht = "400px"){
    nodes.uniq <- unique(c(structure$arcs[,1], structure$arcs[,2]))
    nodes <- data.frame(id = nodes.uniq,
    label = nodes.uniq,
    color = "darkturquoise",
    shadow = TRUE)
    edges <- data.frame(from = structure$arcs[,1],
    to = structure$arcs[,2],
    arrows = "to",
    smooth = TRUE,
    shadow = TRUE,
    color = "black")
    return(visNetwork(nodes, edges, height = ht, width = "100%"))
}
```

In [58]:
```
plot.network(bn.iamb, ht = "600px")
```



```
In [ ]:
```