Andro Rmd

Tapajit Dey

April 20, 2017

```r
setwd("~/Work/release_qual/model/images")
#library
library(plyr)
library(GGally)
library(ggplot2)
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(DataCombine)
library(reshape)
```

```
##
## Attaching package: 'reshape'
```

```
## The following objects are masked from 'package:plyr':
##
##     rename, round_any
```

```r
library(gridExtra)
library(grid)

#Reading .user data
filelist <- list.files(path = "../../data/data1_May16/", pattern = "*4.new.user", full.names = TRUE)
#Reading New data
filelist1 <- list.files(path = "../../data/data1_May16/", pattern = "*4.new$", full.names = TRUE)

UserFile = do.call(rbind, lapply(filelist, function(x) read.csv(file = x,sep=";",na.strings="(not set)"
UserFile[,7] <- as.Date(UserFile[,7],"%Y%m%d")
UserFile[,8] <- as.integer(UserFile[,8])
UserFile[,9] <- as.integer(UserFile[,9])
UserFile[,10] <- as.numeric(UserFile[,10])
UserFile <- UserFile[order(UserFile[,7]),]

NewData <- do.call(rbind, lapply(filelist1, function(x) read.csv(file = x,sep=";")))
NewData[,3] <- as.Date(as.character(NewData[,3]),"%Y%m%d")
NewData <- NewData[order(NewData[,3]),]

total = merge(NewData,UserFile,all=T)
total = total[complete.cases(total),]
total$ga.deviceCategory <- factor(total$ga.deviceCategory)
total$ga.fatalExceptions <- NULL
total$ga.timeOnSite  <- NULL
```

```
#######################################################################################################
#######################################################################################################
library(plyr)
library(GGally)
library(ggplot2)
library(psych)

#2.0.0_317 last day,2.0.0_326 first 2 days,2.0.0_350 last day, 435 last day, 2.1.0_483 last 2, 503 firs
total = total[!rownames(total) %in% c(504,759,761,2010,12248,13245,13244,13825,49530,48963),]


#collapsing all releases to get uniform curve
releases = unique(total$ga.appVersion)
goodrelease = c()
rd = matrix(nrow = 0, ncol=10)

for (r in sort(releases)){
  y= total[total$ga.appVersion == r,]
  z= ddply(y, .(ga.date), summarise,
           nu = sum(ga.newUsers),
           nv = sum(ga.newVisits),
           tu = sum(ga.users),
           tv = sum(ga.visits),
           ex = sum(ga.exceptions))
  #removing suspicious releases
  if (quantile(z$nu,0.1) == quantile(z$nu,0.9) || quantile(z$nv,0.1) == quantile(z$nv,0.9)) next()
  #keeping only releases with >7 days of data or non zero number of exceptions
  if(nrow(z) > 7){
    z$cnu = cumsum(z$nu)
    quantile(z$cnu)
    goodrelease = c(goodrelease,r)
    png(paste0(r,".png"),width = 960, height = 960)
    pairs.panels(z)
    dev.off()
    rd = rbind(rd,(c(r, min(z$ga.date), max(z$ga.date),(z[which(z$cnu > quantile(z$cnu,0.5)),]$ga.date)
                     (z[which(z$cnu > quantile(z$cnu,0.25)),]$ga.date)[1],(z[which(z$cnu > quantile(z$c
                     (z[which(z$cnu >= quantile(z$cnu,0.9)),]$ga.date)[1],sum(z$nu),sum(z$ex),sum(z$nv)
  }
}
```

```
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero

## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
```

```
rd = as.data.frame(rd)
colnames(rd) = c("Release","Mindate","Maxdate","Median","25.Quant","75.Quant","90.Quant","Total.NU","ex
rd$Total.NU = as.numeric(as.character(rd$Total.NU))
Total.NU = as.numeric(as.character(rd$Total.NU))
rd$Total.NV = as.numeric(as.character(rd$Total.NV))

ex = as.numeric(as.character(rd$ex))
rd$ex = ex
rd$Mindate = as.Date(as.integer(as.character(rd$Mindate)),origin="1970-01-01")
rd$Maxdate = as.Date(as.integer(as.character(rd$Maxdate)),origin="1970-01-01")
rd$Median = as.Date(as.integer(as.character(rd$Median)),origin="1970-01-01")
rd$`25.Quant` = as.Date(as.integer(as.character(rd$`25.Quant`)),origin="1970-01-01")
rd$`75.Quant` = as.Date(as.integer(as.character(rd$`75.Quant`)),origin="1970-01-01")
rd$`90.Quant` = as.Date(as.integer(as.character(rd$`90.Quant`)),origin="1970-01-01")

rd$Release = as.character(rd$Release)

refr = rd

tick.col = c()
for (i in 1:nrow(rd)){
  if (ecdf(Total.NU)(rd[i,8]) <0.2){
    rd[i,1] = paste0("",rd[i,1])
  } else if (ecdf(Total.NU)(rd[i,8]) <0.5){
    rd[i,1] = paste0("*",rd[i,1])
  }else if (ecdf(Total.NU)(rd[i,8]) <0.8){
    rd[i,1] = paste0("***",rd[i,1])
  }else {
    rd[i,1] = paste0("******",rd[i,1])
  }

  if (rd[i,9] == 0) {
    tick.col = c(tick.col,"green")
  } else if (rd[i,9] < 50){
    tick.col = c(tick.col,"black")
  } else if (rd[i,9] < 300){
    tick.col = c(tick.col,"blue")
  } else if (rd[i,9] < 750){
    tick.col = c(tick.col,"orange")
  }else {
    tick.col = c(tick.col,"red")
  }
}


rd$`25.Quant` = NULL
rd$`75.Quant` = NULL
rd$ex = NULL
rd$Total.NU = NULL
rd$Total.NV = NULL


# release dates added manually
manrd = data.frame("Release"=as.factor(c("******2.1.2_568","***2.1.2_574","******2.1.4_577","******2.0.
                   "Release Date"=as.Date(c("2015-10-02","2016-01-06","2016-01-22","2014-03-17","2014-0
nrd = merge(rd,manrd,all = T)

r=melt(nrd, id="Release")

png(paste0("Andro_release",".png"),width = 3200, height = 1600)
ggplot(r) + geom_point(aes(x=Release, y=value, colour=variable, shape=variable), size=7) +
  scale_color_manual(values = c("red","blue","green","magenta","black")) + scale_shape_manual(values =
  ylab("Year")+
  theme_bw(base_size = 20) +
  theme(axis.text.y = element_text(colour=tick.col, size=20), axis.title = element_text(size = 20),
        axis.text.x = element_text(colour="black", size=20)) +
  coord_flip()
```

```
## Warning: Removed 23 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
##########################################################
##########################################################
#             Graphical Model                            #
##########################################################
```

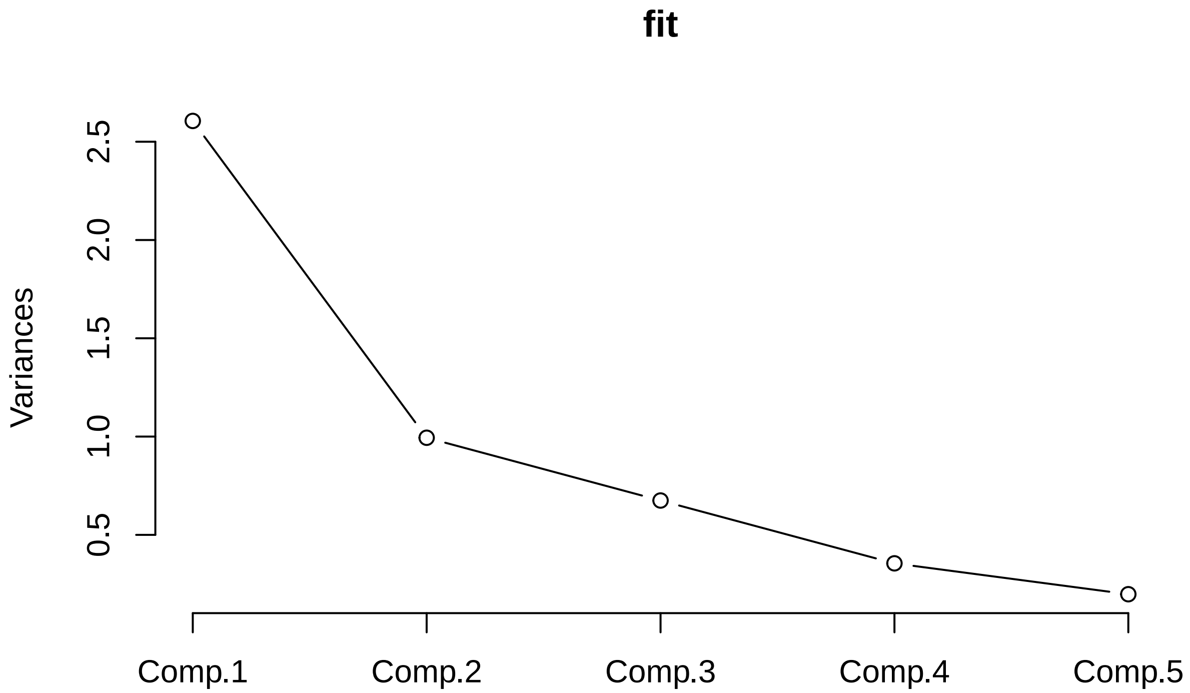
```
##########################################################
setwd("~/Work/release_qual/model/dot_graphs")

library(bnlearn)
```

```
##
## Attaching package: 'bnlearn'

## The following object is masked from 'package:stats':
##
##     sigma
```

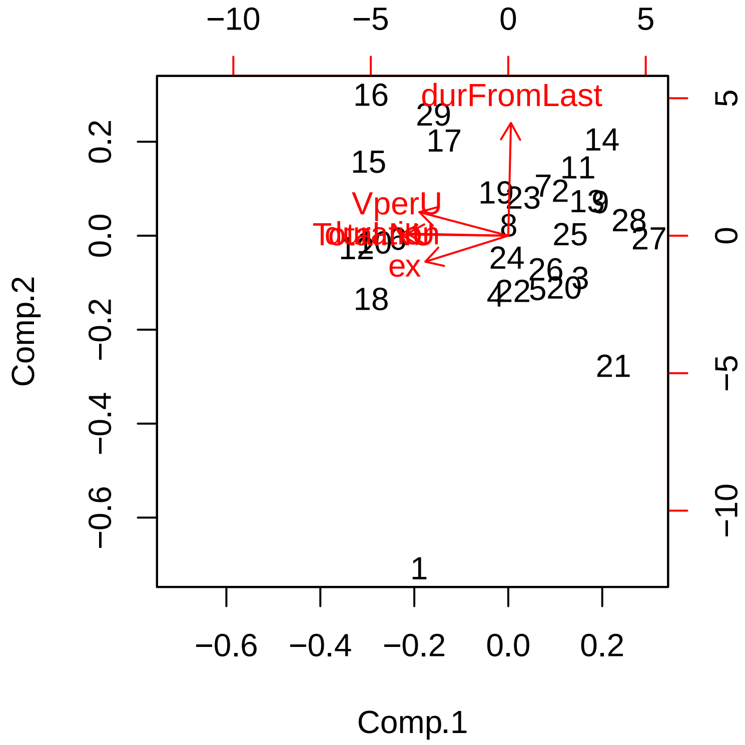
```
refr$Release = as.factor(refr$Release)
refr$duration = as.numeric(refr$`90.Quant` - refr$Mindate)
refr$f.duration = as.numeric(refr$Maxdate - refr$Mindate)
refr[,3:7] = NULL
refr$Release = NULL

refr$VperU = refr$Total.NV / refr$Total.NU
refr$durFromLast = rep(0, nrow(refr))
for (i in 2:nrow(refr)){
  refr[i,8] = refr[i,1] - refr[(i-1),1]
}
refr2 = refr
refr2$Mindate = as.numeric(refr2$Mindate)
refr = refr2

refr$Total.NU = log(refr$Total.NU+1)
refr$duration = log(refr$duration+1)
refr$ex = log(refr$ex + 1)
refr$VperU = log(refr$VperU)
refr$durFromLast = log(abs(refr$durFromLast)+1)

tnv = refr$Total.NV
refr$Total.NV = NULL
refr$f.duration = NULL
refr$Mindate = NULL

refr = data.frame(sapply(refr,function(x) log(x+1)))
refrs = data.frame(sapply(refr,scale))
#PCA and Factor
fit = princomp(refrs)
summary(fit)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     1.6142189 0.9970270 0.8213892 0.59586939 0.44506192
## Proportion of Variance 0.5397527 0.2059130 0.1397552 0.07354821 0.04103088
## Cumulative Proportion  0.5397527 0.7456657 0.8854209 0.95896912 1.00000000
```

```
plot(fit,type="lines")
```



```r
biplot(fit)
```



```r
#psych
fit <- principal(refrs, nfactors=5, rotate="varimax")
fit # print results
```

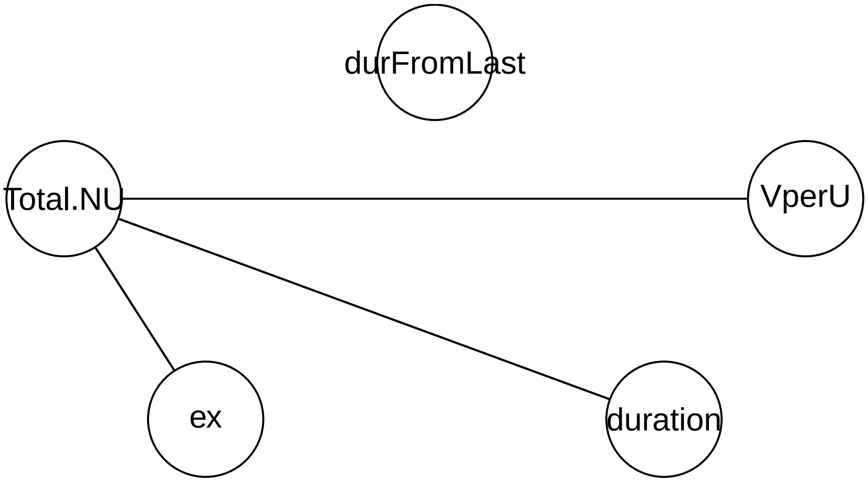
```
## Principal Components Analysis
## Call: principal(r = refrs, nfactors = 5, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##               RC5   RC3  RC4   RC2   RC1 h2       u2 com
## Total.NU    0.42  0.32 0.39 -0.02  0.75  1 -1.8e-15 2.6
## ex          0.10  0.95 0.23 -0.05  0.18  1  1.0e-15 1.2
## duration    0.27  0.27 0.89  0.00  0.25  1  2.2e-16 1.6
## VperU       0.94  0.10 0.23  0.03  0.22  1  6.7e-16 1.3
## durFromLast 0.03 -0.04 0.00  1.00 -0.01  1 -8.9e-16 1.0
##
##                        RC5  RC3  RC4  RC2  RC1
## SS loadings           1.14 1.09 1.05 1.00 0.71
## Proportion Var        0.23 0.22 0.21 0.20 0.14
## Cumulative Var        0.23 0.45 0.66 0.86 1.00
## Proportion Explained  0.23 0.22 0.21 0.20 0.14
## Cumulative Proportion 0.23 0.45 0.66 0.86 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 5 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```r
#Factor
fit <- factanal(refrs, 2, rotation="varimax")
print(fit, digits=2, cutoff=.3, sort=TRUE)
```

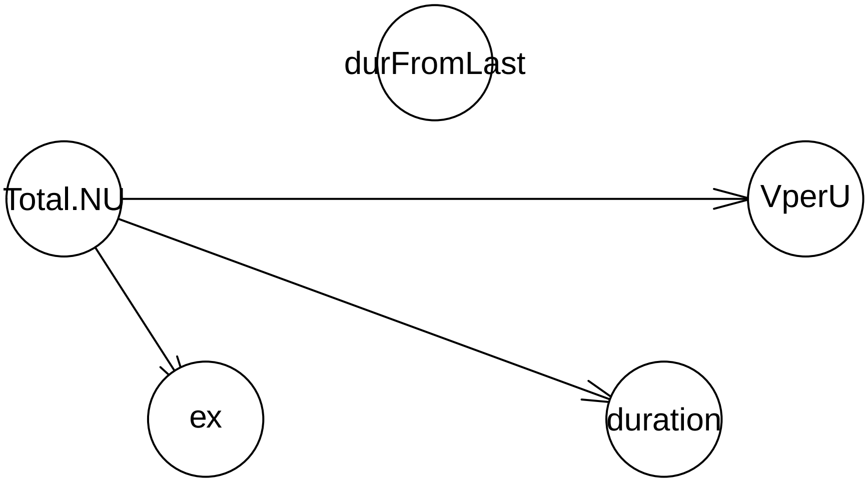
```
##
## Call:
## factanal(x = refrs, factors = 2, rotation = "varimax")
##
## Uniquenesses:
##    Total.NU          ex    duration       VperU durFromLast
##        0.13        0.41        0.37        0.30        0.98
##
## Loadings:
##             Factor1 Factor2
## Total.NU     0.92
## ex           0.55    0.54
## duration     0.76
## VperU        0.79
## durFromLast
##
##                Factor1 Factor2
## SS loadings       2.35    0.46
## Proportion Var    0.47    0.09
## Cumulative Var    0.47    0.56
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 0.07 on 1 degree of freedom.
## The p-value is 0.796
```

```r
#fit <- factor.pa(refrs, nfactors=3, rotation="varimax")
#fit # print results

##Graphical Models##
```

```
pdag1 = iamb(refrs)
plot(pdag1)
```



```
dag1 = cextend(pdag1)
plot(dag1)
```

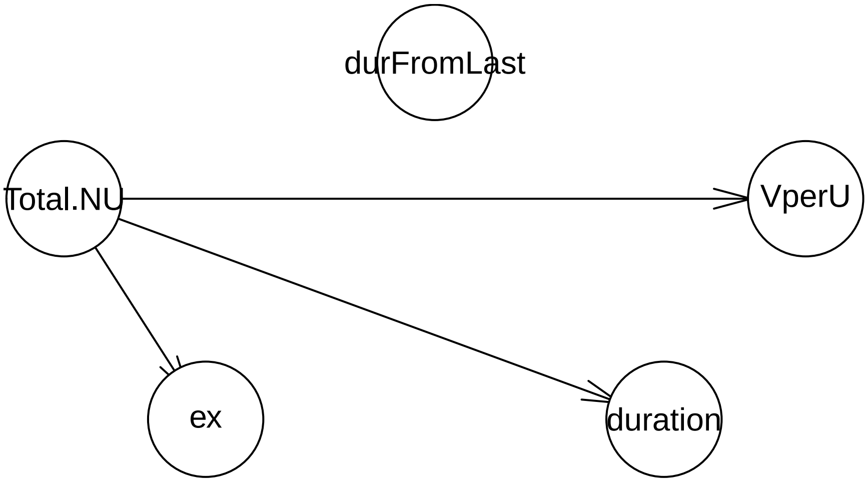


```
(b1 = bn.fit(dag1,refrs))
```

```
##
##   Bayesian network parameters
##
##   Parameters of node Total.NU (Gaussian distribution)
##
## Conditional density: Total.NU
## Coefficients:
##   (Intercept)
## -2.412163e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node ex (Gaussian distribution)
##
## Conditional density: ex | Total.NU
## Coefficients:
```

```
##  (Intercept)      Total.NU
## 5.245867e-17  5.785420e-01
## Standard deviation of the residuals: 0.83062
##
##   Parameters of node duration (Gaussian distribution)
##
## Conditional density: duration | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 5.119953e-16  7.339643e-01
## Standard deviation of the residuals: 0.6916512
##
##   Parameters of node VperU (Gaussian distribution)
##
## Conditional density: VperU | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 1.774154e-16  6.842946e-01
## Standard deviation of the residuals: 0.7425867
##
##   Parameters of node durFromLast (Gaussian distribution)
##
## Conditional density: durFromLast
## Coefficients:
##  (Intercept)
## 1.335139e-16
## Standard deviation of the residuals: 1
```

```
#print(b1$Total.NU)

dag2 = tabu(refrs)
plot(dag2)
```



```
(b2 = bn.fit(dag2, refrs))
```

```
##
##   Bayesian network parameters
##
##   Parameters of node Total.NU (Gaussian distribution)
```

```
##
## Conditional density: Total.NU
## Coefficients:
##   (Intercept)
## -2.412163e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node ex (Gaussian distribution)
##
## Conditional density: ex | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 5.245867e-17  5.785420e-01
## Standard deviation of the residuals: 0.83062
##
##   Parameters of node duration (Gaussian distribution)
##
## Conditional density: duration | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 5.119953e-16  7.339643e-01
## Standard deviation of the residuals: 0.6916512
##
##   Parameters of node VperU (Gaussian distribution)
##
## Conditional density: VperU | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 1.774154e-16  6.842946e-01
## Standard deviation of the residuals: 0.7425867
##
##   Parameters of node durFromLast (Gaussian distribution)
##
## Conditional density: durFromLast
## Coefficients:
##  (Intercept)
## 1.335139e-16
## Standard deviation of the residuals: 1
```

```r
(var2 = b2$Total.NU$sd**2 + b2$ex$sd**2 + b2$duration$sd**2 + b2$VperU$sd**2 + b2$durFromLast$sd**2)
```

```
## [1] 3.719746
```

```r
#write.dot("tabu.dot", b2)


dag3 = rsmax2(refrs, restrict = "si.hiton.pc", maximize = "tabu",
             test = "zf", alpha = 0.1, score = "bic-g")
plot(dag3)
(b3 = bn.fit(dag3, refrs))
```

```
##
##   Bayesian network parameters
##
##   Parameters of node Total.NU (Gaussian distribution)
```

```
##
## Conditional density: Total.NU
## Coefficients:
##   (Intercept)
## -2.412163e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node ex (Gaussian distribution)
##
## Conditional density: ex | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 5.245867e-17  5.785420e-01
## Standard deviation of the residuals: 0.83062
##
##   Parameters of node duration (Gaussian distribution)
##
## Conditional density: duration | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 5.119953e-16  7.339643e-01
## Standard deviation of the residuals: 0.6916512
##
##   Parameters of node VperU (Gaussian distribution)
##
## Conditional density: VperU | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 1.774154e-16  6.842946e-01
## Standard deviation of the residuals: 0.7425867
##
##   Parameters of node durFromLast (Gaussian distribution)
##
## Conditional density: durFromLast
## Coefficients:
##  (Intercept)
## 1.335139e-16
## Standard deviation of the residuals: 1
```

```r
(var3 = b3$Total.NU$sd**2 + b3$ex$sd**2 + b3$duration$sd**2 + b3$VperU$sd**2 + b3$durFromLast$sd**2)
```

```
## [1] 3.719746
```

```r
#write.dot("hyb.dot", b3)



# discrete?
drefr = discretize(refr, method = "hartemink", breaks = 3, ibreaks = 4, idisc = "quantile")
# deal
library(deal)
```

```
##
## Attaching package: 'deal'
```

```
## The following objects are masked from 'package:bnlearn':
##
```

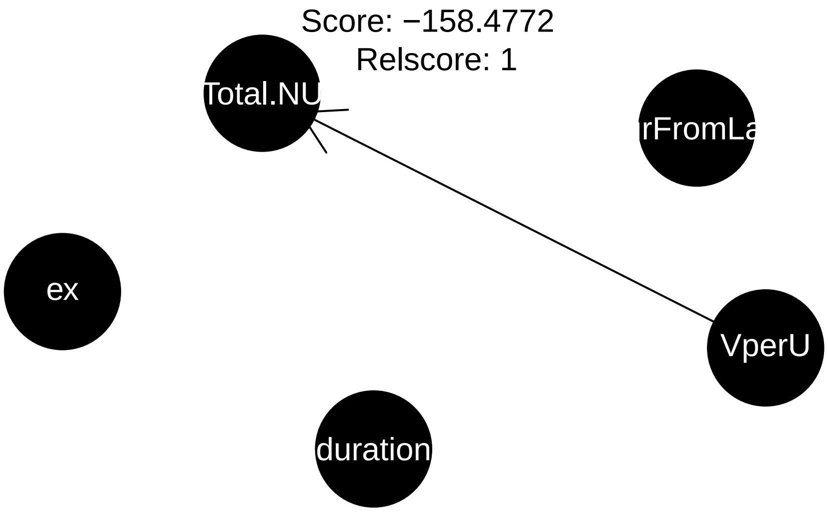
```
##     modelstring, nodes, nodes<-, score
```

```
deal.net = network(drefr)
prior = jointprior(deal.net)
```

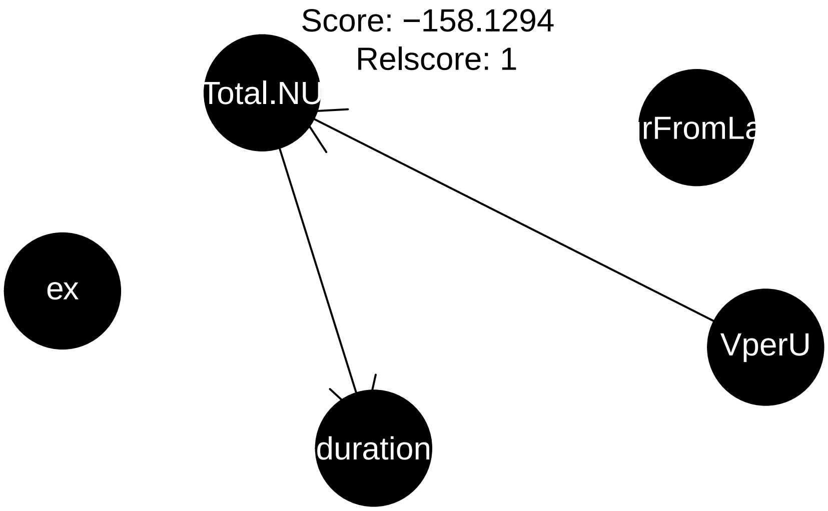
```
## Imaginary sample size: 486
```

```
deal.net = learn(deal.net, drefr, prior)$nw
deal.best = autosearch(deal.net, drefr, prior)
```

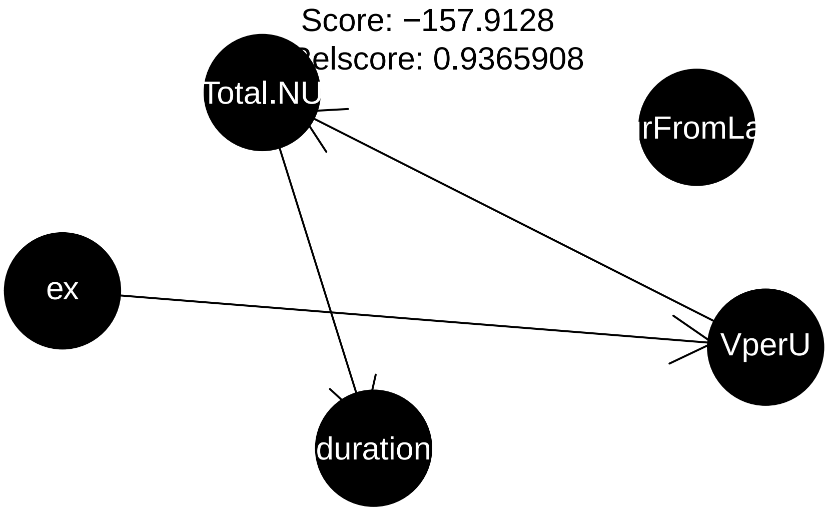
```
## [Autosearch
```



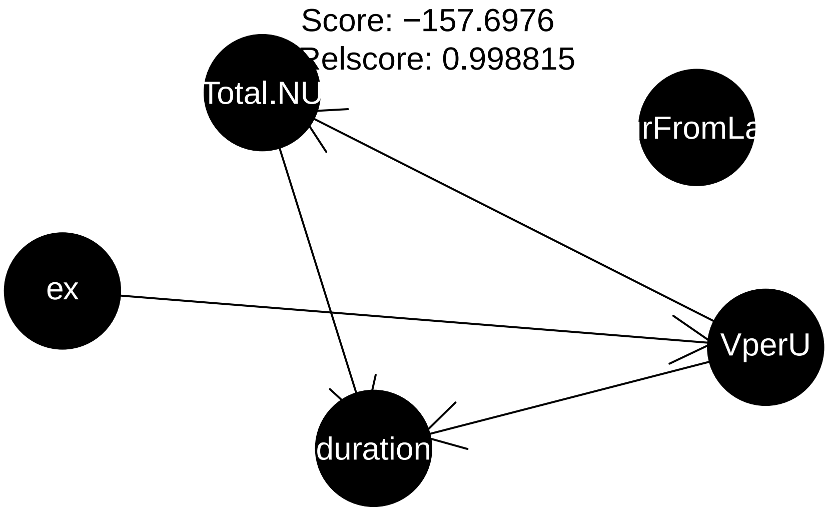
```
## (1) -158.4772 [Total.NU|VperU][ex][duration][VperU][durFromLast]
```



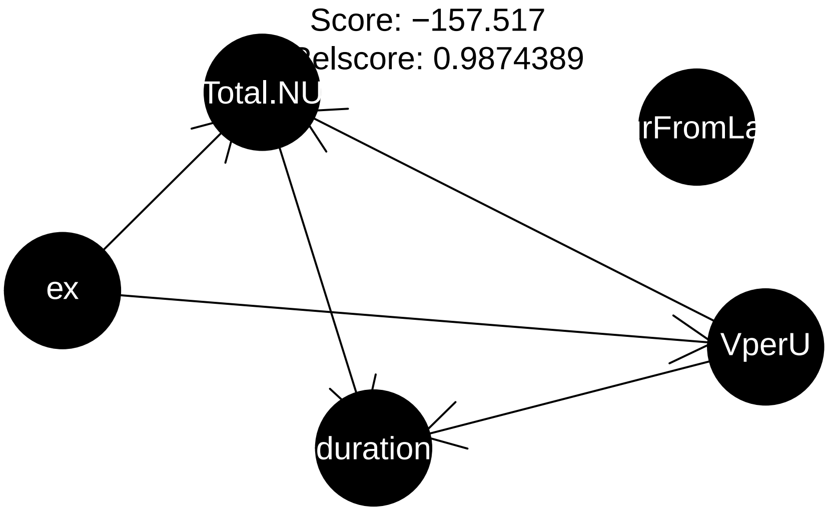
```
## (2) -158.1294 [Total.NU|VperU][ex][duration|Total.NU][VperU][durFromLast]
## .
```



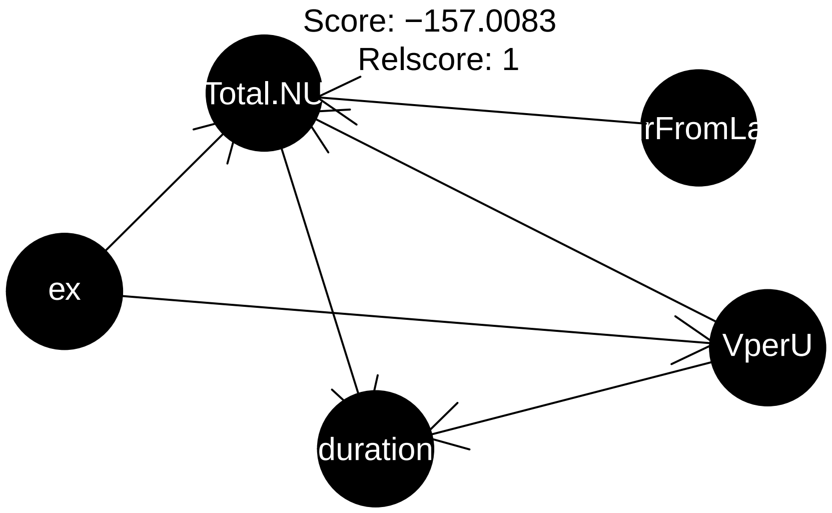
```
## (3) -157.9128 [Total.NU|VperU][ex][duration|Total.NU][VperU|ex][durFromLast]
## .
```



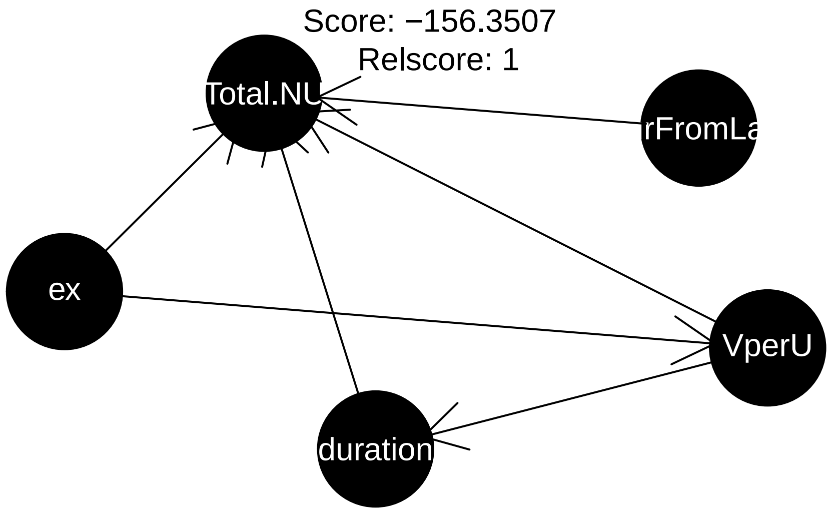
```
## (4) -157.6976 [Total.NU|VperU][ex][duration|Total.NU:VperU][VperU|ex][durFromLast]
## .
```



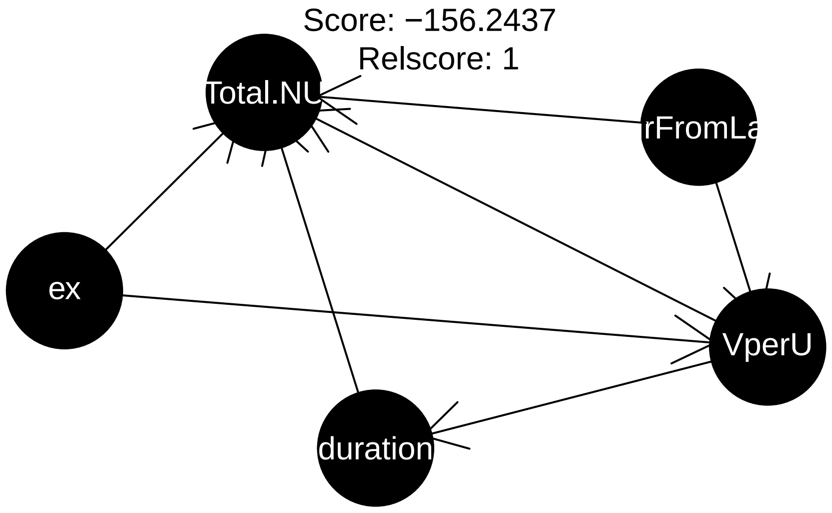
```
## (5) -157.517 [Total.NU|ex:VperU][ex][duration|Total.NU:VperU][VperU|ex][durFromLast]
```



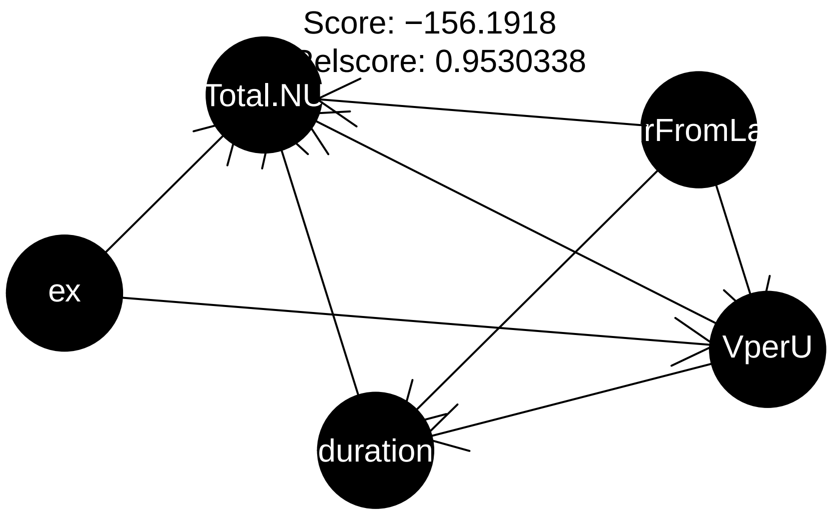
```
## (6) -157.0083 [Total.NU|ex:VperU:durFromLast][ex][duration|Total.NU:VperU][VperU|ex][durFromLast]
```



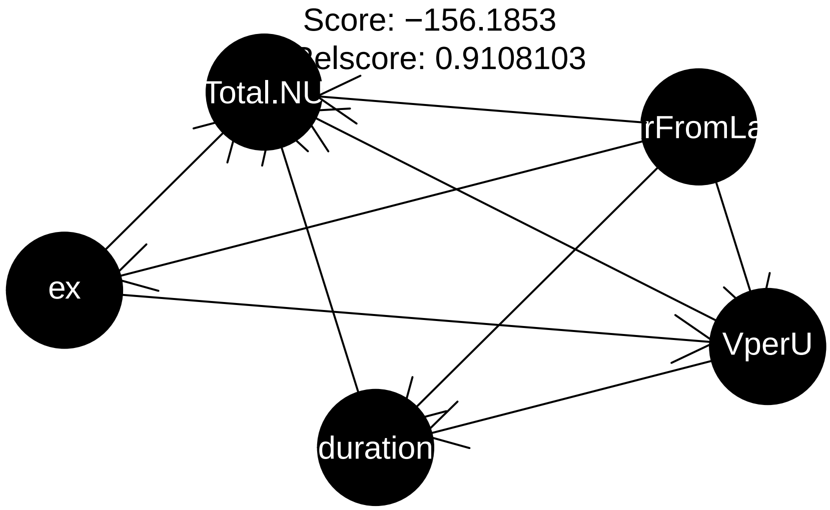
```
## (7) -156.3507 [Total.NU|ex:duration:VperU:durFromLast][ex][duration|VperU][VperU|ex][durFromLast]
```



```
## (8) -156.2437 [Total.NU|ex:duration:VperU:durFromLast][ex][duration|VperU][VperU|ex:durFromLast][durF
## .
```



```
## (9) -156.1918 [Total.NU|ex:duration:VperU:durFromLast][ex][duration|VperU:durFromLast][VperU|ex:durFr
## .
```



```
## (10) -156.1853 [Total.NU|ex:duration:VperU:durFromLast][ex|durFromLast][duration|VperU:durFromLast][V
## .Total 0.124 add 0.052 rem 0.008 turn 0.024 sort 0 choose 0.008 rest 0.032 ]
```

```
bnlearn:::fcat(deal::modelstring(deal.best$nw))
```

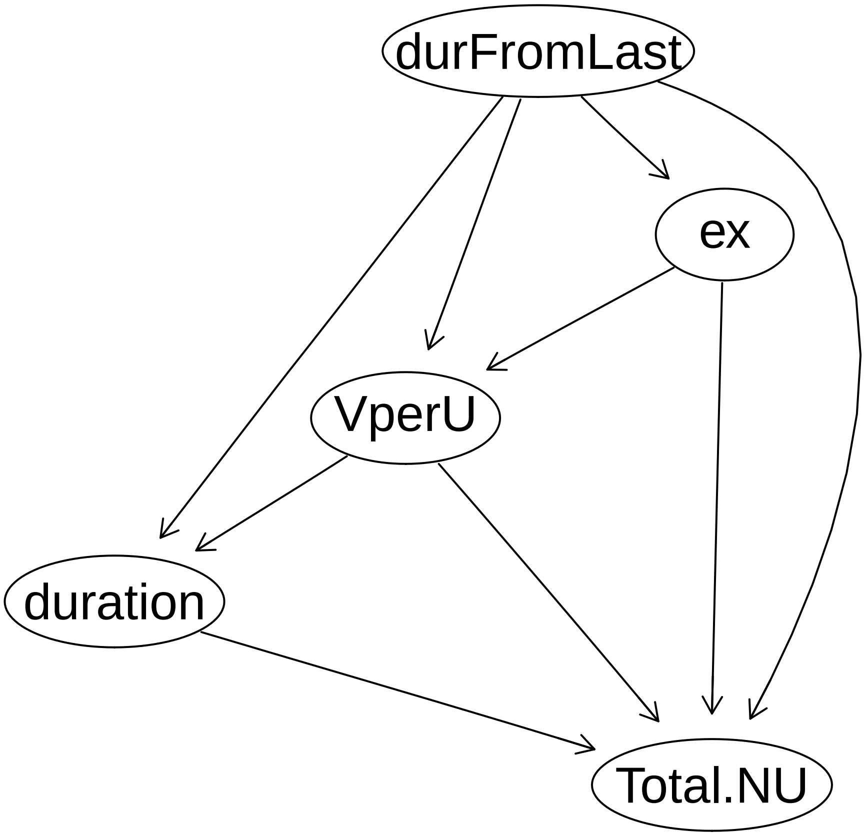
```
##    [Total.NU|ex:duration:VperU:durFromLast][ex|durFromLast]
##    [duration|VperU:durFromLast][VperU|ex:durFromLast][durFromLast]
```

```
dealgr = paste("[Total.NU|ex:duration:VperU:durFromLast][ex|durFromLast][duration|VperU:durFromLast][Vpe
dealnet = model2network(dealgr)
library(Rgraphviz)
```

```
## 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 object is masked from 'package:deal':
##
##     score

## The following object is masked from 'package:bnlearn':
##
##     score

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     IQR, mad, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colnames,
##     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, rownames, sapply, setdiff,
##     sort, table, tapply, union, unique, unsplit, which, which.max,
##     which.min

##
## Attaching package: 'graph'

## The following objects are masked from 'package:deal':
##
##     nodes, nodes<-

## The following objects are masked from 'package:bnlearn':
##
##     degree, nodes, nodes<-

## The following object is masked from 'package:plyr':
##
##     join
```

```r
gr = graphviz.plot(dealnet, shape = "ellipse")
```



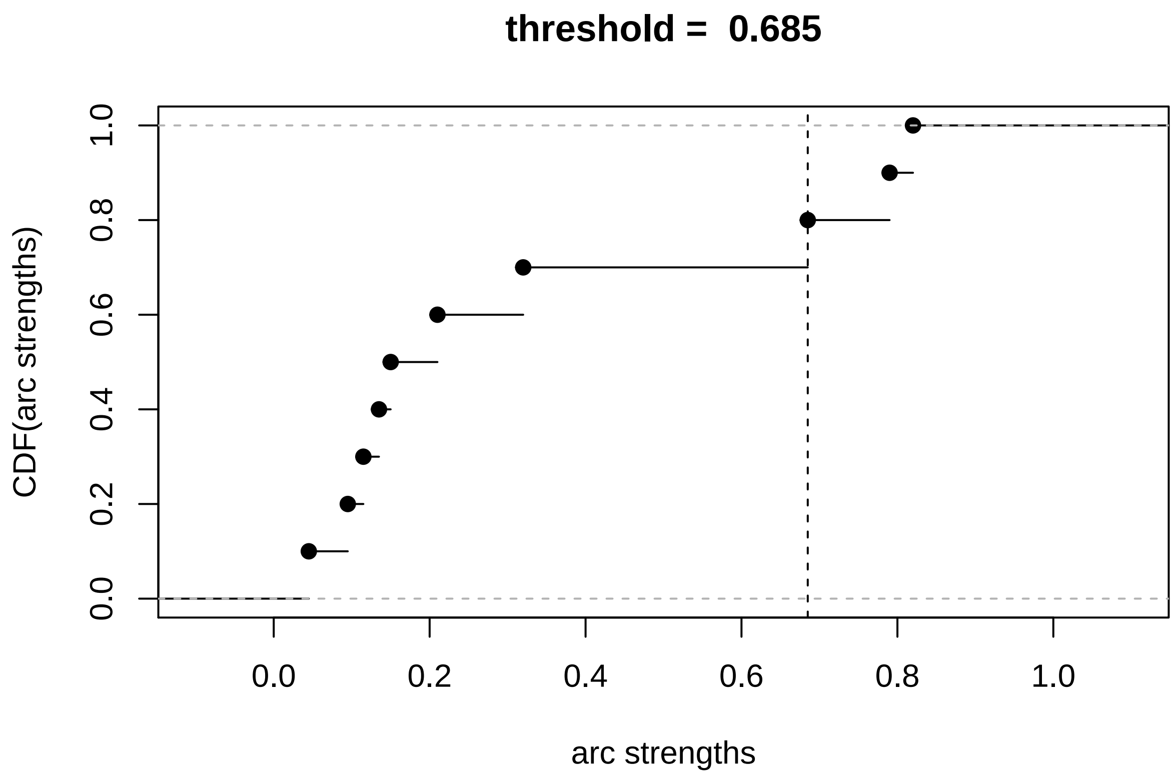
```
(bd = bn.fit(dealnet, refrs))
```

```
##
##   Bayesian network parameters
##
##   Parameters of node duration (Gaussian distribution)
##
## Conditional density: duration | durFromLast + VperU
## Coefficients:
##   (Intercept)    durFromLast          VperU
##  3.978811e-16  -2.703421e-02   5.423634e-01
## Standard deviation of the residuals: 0.8723523
##
##   Parameters of node durFromLast (Gaussian distribution)
##
## Conditional density: durFromLast
## Coefficients:
##  (Intercept)
## 1.335139e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node ex (Gaussian distribution)
##
## Conditional density: ex | durFromLast
## Coefficients:
##   (Intercept)    durFromLast
## -7.616847e-17  -8.183877e-02
## Standard deviation of the residuals: 1.014934
##
##   Parameters of node Total.NU (Gaussian distribution)
##
## Conditional density: Total.NU | duration + durFromLast + ex + VperU
## Coefficients:
##   (Intercept)       duration    durFromLast             ex          VperU
## -2.984053e-16   3.703906e-01  -2.356082e-02   2.652122e-01   4.114062e-01
## Standard deviation of the residuals: 0.5843034
##
##   Parameters of node VperU (Gaussian distribution)
##
## Conditional density: VperU | durFromLast + ex
## Coefficients:
##   (Intercept)    durFromLast             ex
## -1.728833e-17   7.545504e-02   2.842708e-01
## Standard deviation of the residuals: 0.9937538
```

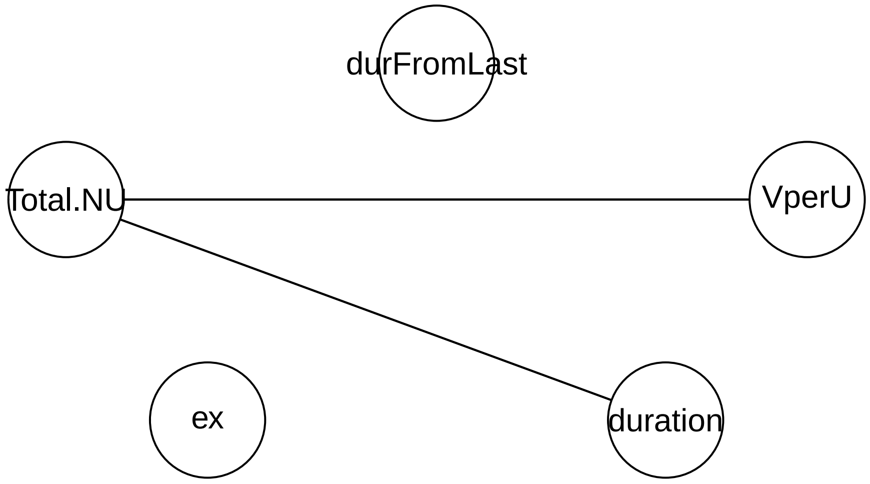
```
(vard = bd$Total.NU$sd**2 + bd$ex$sd**2 + bd$duration$sd**2 + bd$VperU$sd**2 + bd$durFromLast$sd**2)
```

```
## [1] 4.120047
```

```
#write.dot("deal.dot",bd)


#catnet
library(catnet)
netlist = vector(200, mode = "list")
ndata = nrow(drefr)
nodes = names(drefr)
netlist = lapply(netlist, function(net) {
 boot = drefr[sample(ndata, replace = TRUE), ]
 nets = cnSearchOrder(boot)
 best = cnFindBIC(nets, ndata)
 cnMatEdges(best)
 })
sa = custom.strength(netlist, nodes = nodes)
plot(sa)
```



```
avg.catnet = averaged.network(sa, threshold = 0.7)
plot(avg.catnet)
```

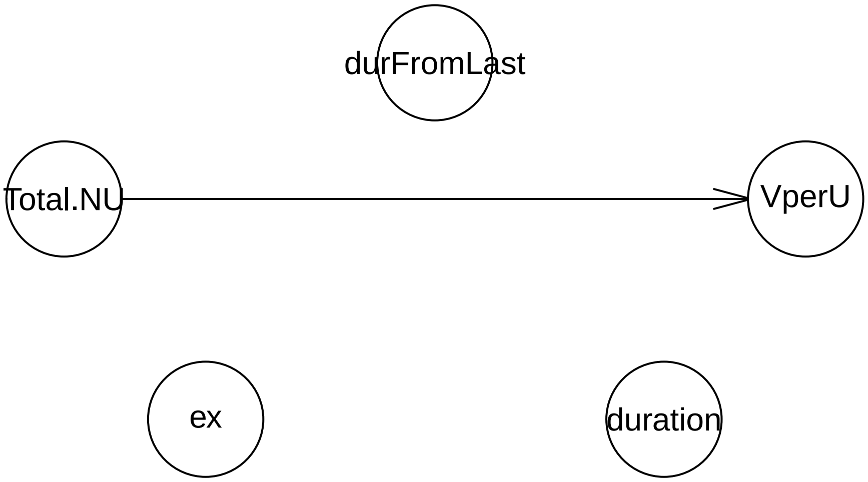


```
#catfit = bn.fit(avg.catnet,drefr)
#boot
(boot = boot.strength(data = drefr, R = 200, algorithm = "hc",
                      algorithm.args = list(score = "bde", iss = 10)))
```

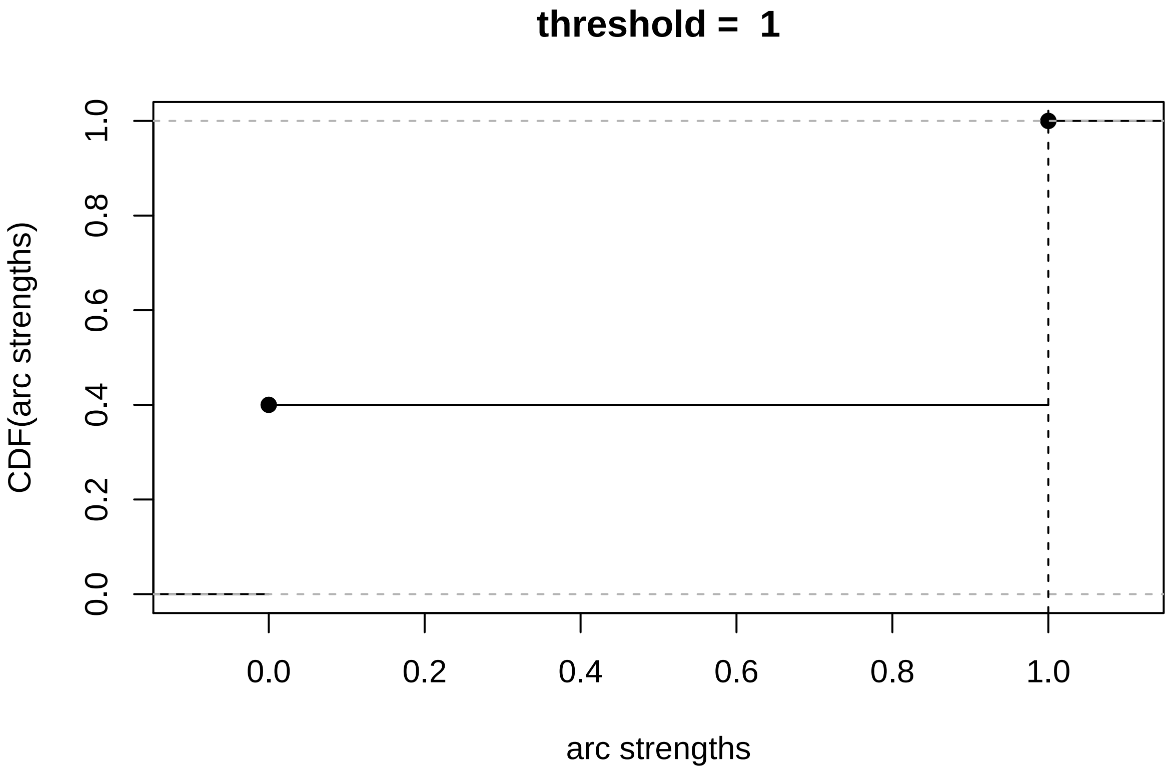
```
## Warning in check.data(x): variable duration has levels that are not
## observed in the data.
```

```
##          from          to strength direction
## 1    Total.NU          ex    0.715 0.6153846
## 2    Total.NU    duration    0.710 0.5387324
## 3    Total.NU       VperU    0.935 0.5534759
## 4     Total.NU durFromLast    0.795 0.6320755
## 5           ex    Total.NU    0.715 0.3846154
## 6           ex    duration    0.660 0.3674242
## 7           ex       VperU    0.845 0.4349112
## 8           ex durFromLast    0.850 0.4941176
## 9     duration    Total.NU    0.710 0.4612676
## 10    duration          ex    0.660 0.6325758
## 11    duration       VperU    0.595 0.4873950
## 12    duration durFromLast    0.630 0.5674603
## 13       VperU    Total.NU    0.935 0.4465241
## 14       VperU          ex    0.845 0.5650888
## 15       VperU    duration    0.595 0.5126050
## 16       VperU durFromLast    0.400 0.5312500
## 17 durFromLast    Total.NU    0.795 0.3679245
## 18 durFromLast          ex    0.850 0.5058824
## 19 durFromLast    duration    0.630 0.4325397
## 20 durFromLast       VperU    0.400 0.4687500
```

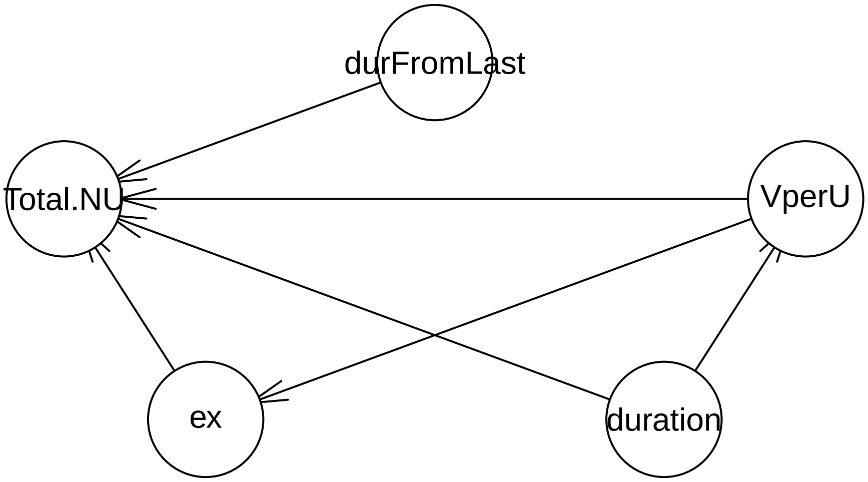
```
avg.boot = averaged.network(boot, threshold = 0.85)
plot(avg.boot)
```



```
#pcalg - doesn't work
# library(pcalg)
# suffStat = list(dm = sapply(drefr,as.integer)-1, nlev = sapply(drefr, nlevels),adaptDF = FALSE)
# pcalg.net = pc(suffStat, indepTest = disCItest, p = ncol(drefr),alpha = 0.05)

#mbde

#refr$Total.NU = log(refr$Total.NU+1)
drefr = discretize(refr, method = "hartemink", breaks = 3, ibreaks = 4, idisc = "quantile")
nodes = names(drefr)

start = random.graph(nodes = nodes, method = "melancon", num = 500,
                     burn.in = 10^5, every = 100)

netlist = lapply(start, function(net) {tabu(drefr, score = "bde",
                                            iss = 1, start = net, tabu = 50) })
arcs = custom.strength(netlist, nodes = nodes, cpdag = F)
plot(arcs)
```



```
bn.mbde = averaged.network(arcs, threshold = 0.12)
plot(bn.mbde)
```



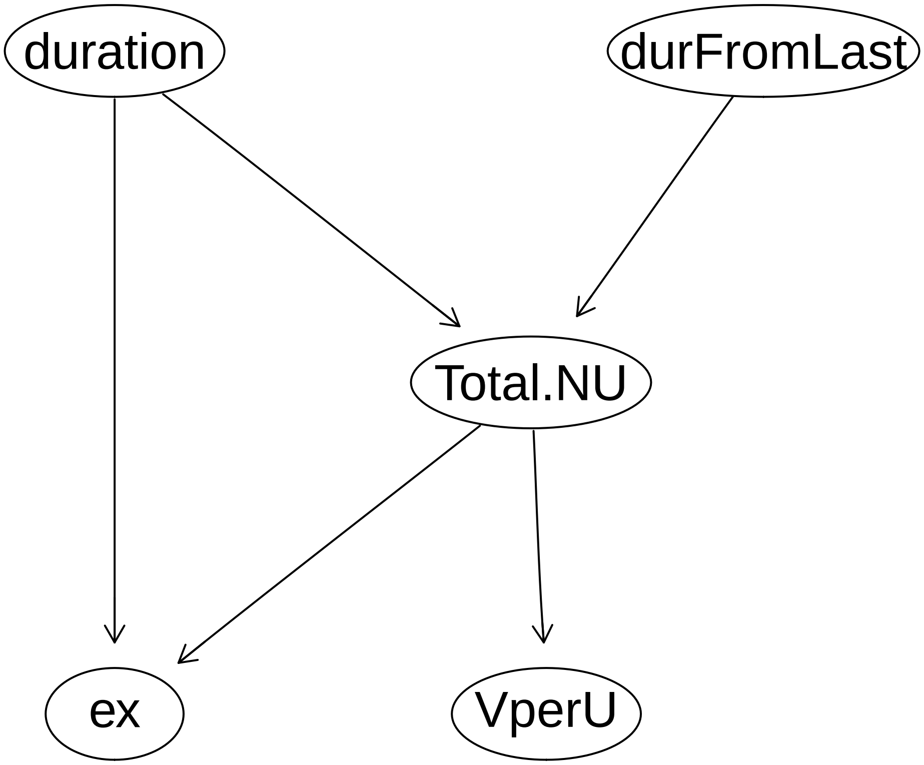
```
(b = bn.fit(bn.mbde, refr))
```

```
##
##   Bayesian network parameters
##
##   Parameters of node Total.NU (Gaussian distribution)
##
## Conditional density: Total.NU | ex + duration + VperU + durFromLast
## Coefficients:
## (Intercept)           ex     duration        VperU  durFromLast
##  0.41196900   0.10949101   0.61044943   0.30106663  -0.02294076
## Standard deviation of the residuals: 0.1983429
##
##   Parameters of node ex (Gaussian distribution)
##
## Conditional density: ex | VperU
## Coefficients:
## (Intercept)        VperU
##   1.0505554    0.4929478
## Standard deviation of the residuals: 0.8042888
##
##   Parameters of node duration (Gaussian distribution)
##
## Conditional density: duration
## Coefficients:
## (Intercept)
##    1.641725
## Standard deviation of the residuals: 0.2059627
##
##   Parameters of node VperU (Gaussian distribution)
##
## Conditional density: VperU | duration
## Coefficients:
## (Intercept)     duration
##   -1.369161     1.218308
## Standard deviation of the residuals: 0.3972895
##
##   Parameters of node durFromLast (Gaussian distribution)
##
## Conditional density: durFromLast
## Coefficients:
## (Intercept)
##    1.390503
## Standard deviation of the residuals: 0.3486269
```

```r
(var4 = b$Total.NU$sd**2 + b$ex$sd**2 + b$duration$sd**2 + b$VperU$sd**2 + b$durFromLast$sd**2)
```

```
## [1] 1.008021
```

```r
#write.dot("mbde.dot",bn.mbde)

########################
custom = paste("[duration][ex|Total.NU:duration][durFromLast][Total.NU|duration:durFromLast][VperU|Tota
custom.net = model2network(custom)
library(Rgraphviz)
gr = graphviz.plot(custom.net, shape = "ellipse")
```



```
(b5 = bn.fit(custom.net, refrs))
```

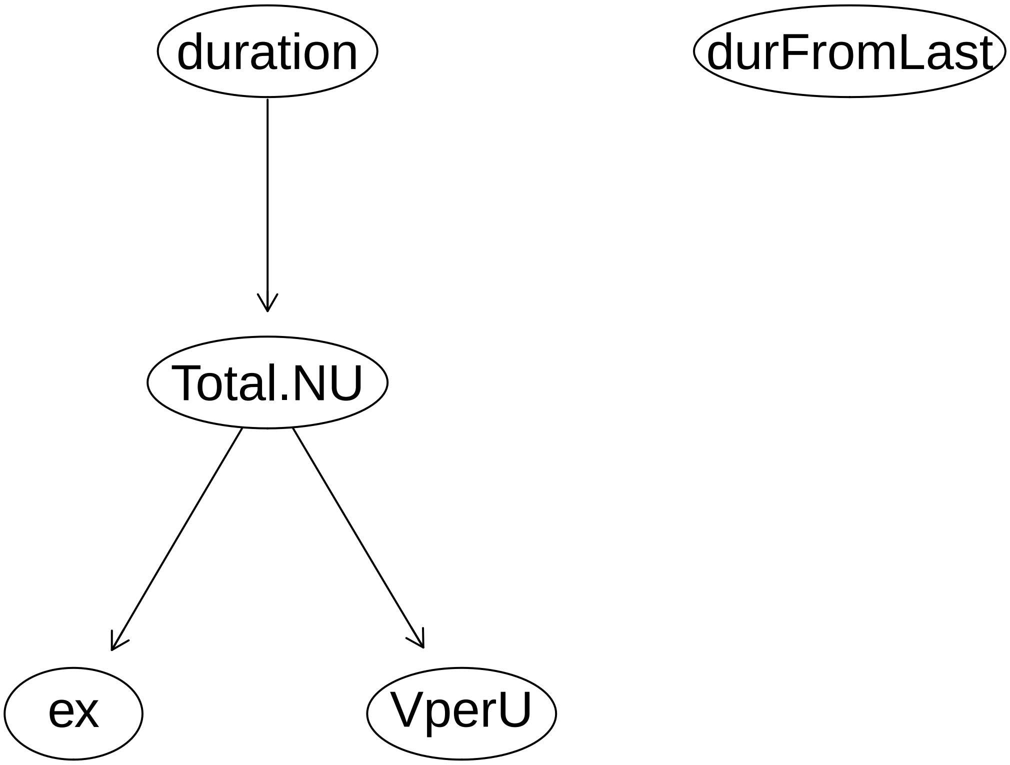
```
##
##   Bayesian network parameters
##
##   Parameters of node duration (Gaussian distribution)
##
## Conditional density: duration
## Coefficients:
##  (Intercept)
## 3.349512e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node durFromLast (Gaussian distribution)
##
## Conditional density: durFromLast
## Coefficients:
##  (Intercept)
## 1.335139e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node ex (Gaussian distribution)
##
## Conditional density: ex | duration + Total.NU
## Coefficients:
##   (Intercept)       duration       Total.NU
## -1.586690e-16   2.324283e-01   4.079479e-01
## Standard deviation of the residuals: 0.8304383
##
##   Parameters of node Total.NU (Gaussian distribution)
```

```
##
## Conditional density: Total.NU | duration + durFromLast
## Coefficients:
##   (Intercept)       duration    durFromLast
## -4.851584e-16   7.339952e-01  -2.425643e-02
## Standard deviation of the residuals: 0.7043771
##
##   Parameters of node VperU (Gaussian distribution)
##
## Conditional density: VperU | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 1.774154e-16  6.842946e-01
## Standard deviation of the residuals: 0.7425867
```

```
(var5 = b5$Total.NU$sd**2 + b5$ex$sd**2 + b5$duration$sd**2 + b5$VperU$sd**2 + b5$durFromLast$sd**2)
```

```
## [1] 3.73721
```

```
#write.dot("custom.dot",b5)

###
### F stat : anova
# anova(lm(VperU ~ Total.NU, data = refrs))

#Final
custom1 = paste("[duration][ex|Total.NU][durFromLast][Total.NU|duration][VperU|Total.NU]")
custom1.net = model2network(custom1)
graphviz.plot(custom1.net, shape = "ellipse")
```



```r
(b51 = bn.fit(custom1.net, refrs))
```

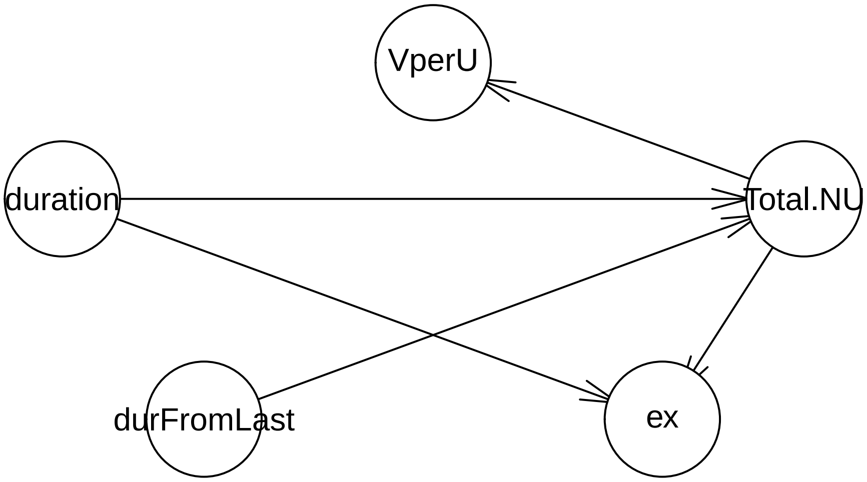
```
##
##   Bayesian network parameters
##
##   Parameters of node duration (Gaussian distribution)
##
## Conditional density: duration
## Coefficients:
##  (Intercept)
## 3.349512e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node durFromLast (Gaussian distribution)
##
## Conditional density: durFromLast
## Coefficients:
##  (Intercept)
## 1.335139e-16
## Standard deviation of the residuals: 1
##
##   Parameters of node ex (Gaussian distribution)
##
## Conditional density: ex | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 5.245867e-17  5.785420e-01
## Standard deviation of the residuals: 0.83062
##
##   Parameters of node Total.NU (Gaussian distribution)
##
## Conditional density: Total.NU | duration
## Coefficients:
##   (Intercept)       duration
## -4.870585e-16   7.339643e-01
## Standard deviation of the residuals: 0.6916512
##
##   Parameters of node VperU (Gaussian distribution)
##
## Conditional density: VperU | Total.NU
## Coefficients:
##  (Intercept)      Total.NU
## 1.774154e-16  6.842946e-01
## Standard deviation of the residuals: 0.7425867
```

```r
(var51 = b51$Total.NU$sd**2 + b51$ex$sd**2 + b51$duration$sd**2 + b51$VperU$sd**2 + b51$durFromLast$sd**
```

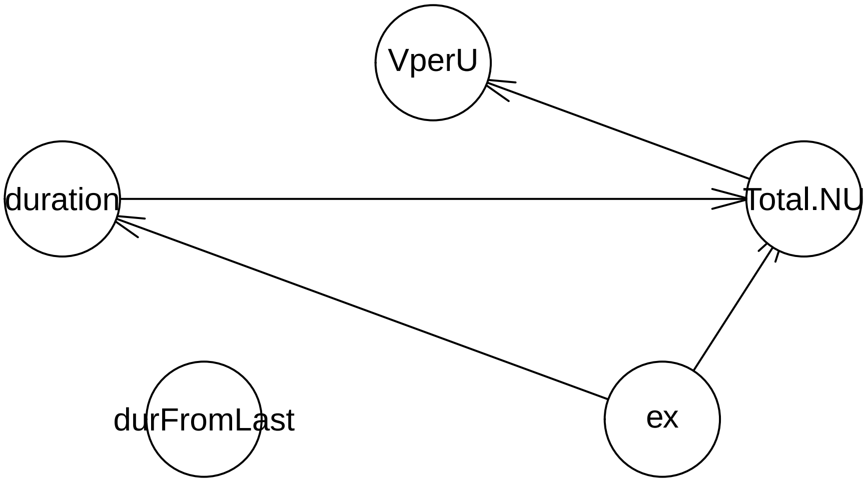
```
## [1] 3.719746
```

```r
#write.dot("final.dot",b51)


#######################
#simulation
#######################
```

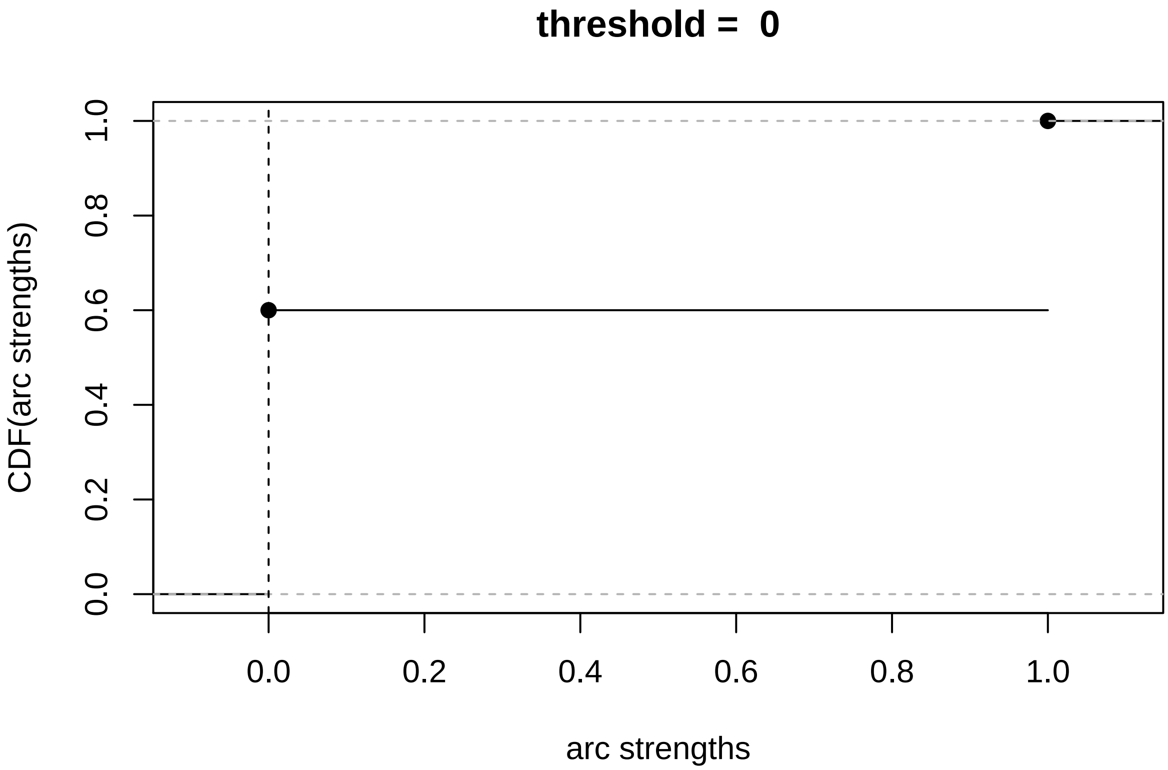
```r
tst = rbn(b5, n=10000, refrs)
dag2s = tabu(tst)
plot(dag2s)
```



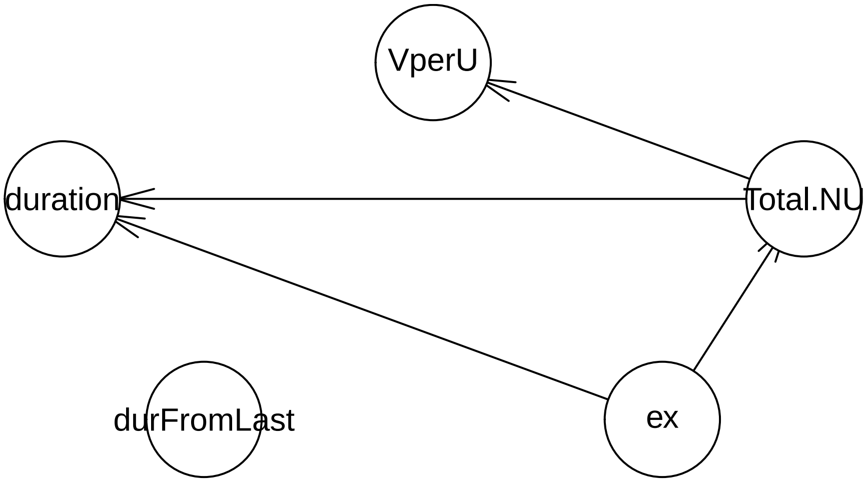
```r
#write.dot("sim2.dot", dag2s)

dag3s = rsmax2(tst, restrict = "si.hiton.pc", maximize = "tabu",
              test = "zf", alpha = 0.1, score = "bic-g")
plot(dag3s)
```



```r
#write.dot("sim3.dot", dag3s)

dtst = discretize(tst, method = "hartemink", breaks = 3, ibreaks = 4, idisc = "quantile")
nodes = names(dtst)

start = random.graph(nodes = nodes, method = "melancon", num = 500,
                     burn.in = 10^5, every = 100)

netlist2 = lapply(start, function(net) {tabu(dtst, score = "bde",
                                            iss = 1, start = net, tabu = 50) })
arcs2 = custom.strength(netlist2, nodes = nodes, cpdag = F)
plot(arcs2)
```



```
bn.mbde2 = averaged.network(arcs2, threshold = 0.12)
plot(bn.mbde2)
```



```
#write.dot("sim4.dot", bn.mbde2)




##############################################################
# plot2 = qplot(y=rd$Release, x=log(Total.NU),xlab = "Release Size", ylab = "")
#
#
#
```

```
# vplayout <- function(x, y) viewport(layout.pos.row = x, layout.pos.col = y)
#
# grid.newpage()
# pushViewport(viewport(layout = grid.layout(1, 5))) #1 rows, 4 columns
# print(plot2, vp = vplayout(1, 1:2))
# print(plot1, vp = vplayout(1, 3:5))
#
#
#
# ########################
# # Looking for generic difference between exception & non exceptions
# Mode <- function(x) {
#    x <- na.omit(x)
#    ux <- unique(x)
#    ux[which.max(tabulate(match(x, ux)))]
#  }
#
# tab <- as.data.frame(matrix(NA,nrow=5,ncol=12))
# names(tab) <- as.vector(t(outer(c("Excep","No.Excep","All"), c("mean","median","mode","sd"), paste, s
# row.names(tab) <- c("New.User","Total.user","New.Visit","Tot.Visit","Session.P.U")
#
# tab[1:5,9] <- round(colMeans(total[,c(11,12,5,6,13)]),2)
# tab[1:5,10] <- round(apply(total[,c(11,12,5,6,13)],2,median),2)
# tab[1:5,11] <- round(apply(total[,c(11,12,5,6,13)],2,Mode),2)
# tab[1:5,12] <- round(apply(total[,c(11,12,5,6,13)],2,sd),2)
#
# y = total[which(total$ga.exceptions == 0),c(11,12,5,6,13)]
# tab[1:5,1] <- round(colMeans(y),2)
# tab[1:5,2] <- round(apply(y,2,median),2)
# tab[1:5,3] <- round(apply(y,2,Mode),2)
# tab[1:5,4] <- round(apply(y,2,sd),2)
#
# y = total[which(total$ga.exceptions != 0),c(11,12,5,6,13)]
# tab[1:5,5] <- round(colMeans(y),2)
# tab[1:5,6] <- round(apply(y,2,median),2)
# tab[1:5,7] <- round(apply(y,2,Mode),2)
# tab[1:5,8] <- round(apply(y,2,sd),2)
#
# tab
#
# ## Aggregated on release level
# filtered = total[which(total$ga.appVersion %in% goodrelease),]
# agg.f = ddply(filtered,.(ga.appVersion), summarise,
#                       nu = sum(ga.newUsers),
#                       nv = sum(ga.newVisits),
#                       tu = sum(ga.users),
#                       tv = sum(ga.visits),
#                       spu = sum(ga.sessionsPerUser),
#                       ex = sum(ga.exceptions))
#
# agg.filtered = data.frame(scale(agg.f[,2:6]))
#
# tab <- as.data.frame(matrix(NA,nrow=5,ncol=12))
# names(tab) <- as.vector(t(outer(c("Excep","No.Excep","All"), c("mean","median","mode","sd"), paste, s
# row.names(tab) <- c("New.User","Total.user","New.Visit","Tot.Visit","Session.P.U")
#
# tab[1:5,9] <- round(colMeans(agg.filtered),2)
# tab[1:5,10] <- round(apply(agg.filtered,2,median),2)
# tab[1:5,11] <- round(apply(agg.filtered,2,Mode),2)
# tab[1:5,12] <- round(apply(agg.filtered,2,sd),2)
#
# y = agg.filtered[which(agg.f$ex == 0),]
# tab[1:5,1] <- round(colMeans(y),2)
# tab[1:5,2] <- round(apply(y,2,median),2)
# tab[1:5,3] <- round(apply(y,2,Mode),2)
# tab[1:5,4] <- round(apply(y,2,sd),2)
#
# y = agg.filtered[which(agg.f$ex != 0),]
# tab[1:5,5] <- round(colMeans(y),2)
# tab[1:5,6] <- round(apply(y,2,median),2)
# tab[1:5,7] <- round(apply(y,2,Mode),2)
# tab[1:5,8] <- round(apply(y,2,sd),2)
#
# tab
# ############
#
# total$binexcep = as.factor(ifelse(total$ga.exceptions>0,1,0))
# mod = glm(binexcep ~ ga.newUsers + ga.newVisits + ga.users + ga.visits + ga.sessionsPerUser, data = t
#           family = binomial)
#
# summary(mod)
```