Assignment 6

Network Analysis 2017

Part 1: Conceptual

Exercise 1 (5 points)
List the three assumptions underlying DAG estimation and explain these in your own words. Describe of each assumption how violations can lead to problems in the estimation of DAGs.

■

Part 2: Practical

For this assignment, we need the following R packages (install them first if you do not have them yet):

```
library("psych")
library("bnlearn")
library("qgraph")
```

In this assignment we revisit the Big 5 dataset we used last time, now including variables on age, gender and education:

Item label	Item description
A1	Am indifferent to the feelings of others
A2	Inquire about others' well-being
A3	Know how to comfort others
A4	Love children
A5	Make people feel at ease
C1	Am exacting in my work
C2	Continue until everything is perfect
C3	Do things according to a plan
C4	Do things in a half-way manner
C5	Waste my time
E1	Don't talk a lot
E2	Find it difficult to approach others
E3	Know how to captivate people
E4	Make friends easily
E5	Take charge
N1	Get angry easily
N2	Get irritated easily
N3	Have frequent mood swings
N4	Often feel blue
N5	Panic easily
O1	Am full of ideas
O2	Avoid difficult reading material
O3	Carry the conversation to a higher level
O4	Spend time reflecting on things
O5	Will not probe deeply into a subject
gender	Males = 1 Females = 2
education	1 = HS, 2 = finished HS, 3 = some college, 4 = college graduate 5 = graduate degree
age	Age in years

As DAG estimation for ordinal variables is not trivial, we will assume all variables are continuous except gender. We can load the data in R as follows:

```
# Load data:
data("bfi")
```

Next, we need to remove all rows with missing data:

```
# Remove NA:
bfiNoNA <- na.omit(bfi)
```

For bnlearn, we have to specify that gender is a categorical variable:

```
# Set gender as catagorical:
bfiNoNA$gender <- factor(bfiNoNA$gender,
                         levels=c(1,2),
                         labels=c("Male","Female"))
```

All other variables we have to set to be numeric (bnlearn otherwise thinks they are of class "integer" and gives an error):

```
# Make everything else numeric:
for (i in c(1:25,27:28)){
  bfiNoNA[,i] <- as.numeric(bfiNoNA[,i])
}
```

Finally, as we cannot readily assume normality, we can apply the nonparanormal transformation to all variables:

```
# Apply non-paranormal transformation:
library("huge")
bfiNoNA[,-26] <- huge.npn(bfiNoNA[,-26])

## Conducting the nonparanormal (npn) transformation via shrunkun ECDF....done.
```

Running all these codes will give you the ‘bfiNoNA‘ dataset you will work with. Your main assignment this week is to select some of the variables you think causally influence each other, think of a hypothetical model, and then estimate the network from data.
For example, I selected the following variables:

```
selection <- c(
  "age", # Age in years
  "education", # Education ranging from 1 = high school to 5 = PhD
  "C5", # Waste my time
  "O1", # Am full of ideas
  "N5" # Panic easily
)
bfiSub <- bfiNoNA[,selection]
```

I expected that the older a person is the less they waste their time and panic easily, and that wasting your time leads to being less full of ideas. A few causal relationships I actually already *know*! I know that age causes in part education, as younger people cannot have finished college yet. I can aide bnlearn by *whitelisting* (forcing to include) this relationship:

```
Whitelist <- matrix(c(
  "age","education"
),,2,byrow=TRUE)
colnames(Whitelist) <- c("from","to")
Whitelist

##      from  to
## [1,] "age" "education"
```

I can add rows to this matrix to whitelist more edges. Next, I also *know* that age cannot be caused by anything! Therefore, I can *blacklist* (forcing to exclude) all edges to age:

```
Blacklist <- matrix(c(
  "education","age",
  "C5","age",
  "O1","age",
  "N5","age"
),,2,byrow=TRUE)
colnames(Blacklist) <- c("from","to")
Blacklist

##      from        to
## [1,] "education" "age"
## [2,] "C5"        "age"
## [3,] "O1"        "age"
## [4,] "N5"        "age"
```

Now everything is in place! I can use one of the many (see `http://www.bnlearn.com/`) algorithms in bnlearn to estimate a network structure. As I wish to interpret my model causally and I have a large sample-size, I opt for a constraint-based algorithm similar to the IC-algorithm. Here, I use the "Incremental Association Markov Blanket" algorithm:

```
Res <- iamb(bfiSub, whitelist = Whitelist, blacklist = Blacklist)
```

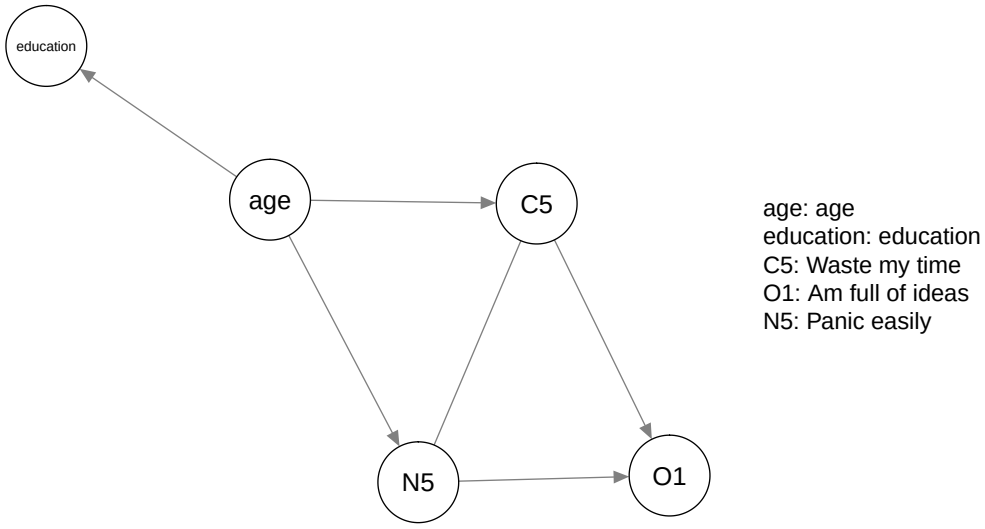
Note, if we want some information on the results I can usually print the results, but there seems to be a dependency problem with qgraph leading to an error. If you wish to print the result, you can use the following code:

```
bnlearn:::print.bn(Res)
```

After estimating the network, we may wish to plot it. The bnlearn package is supported by qgraph, and we can readily feed the object to `qgraph` to plot the results:

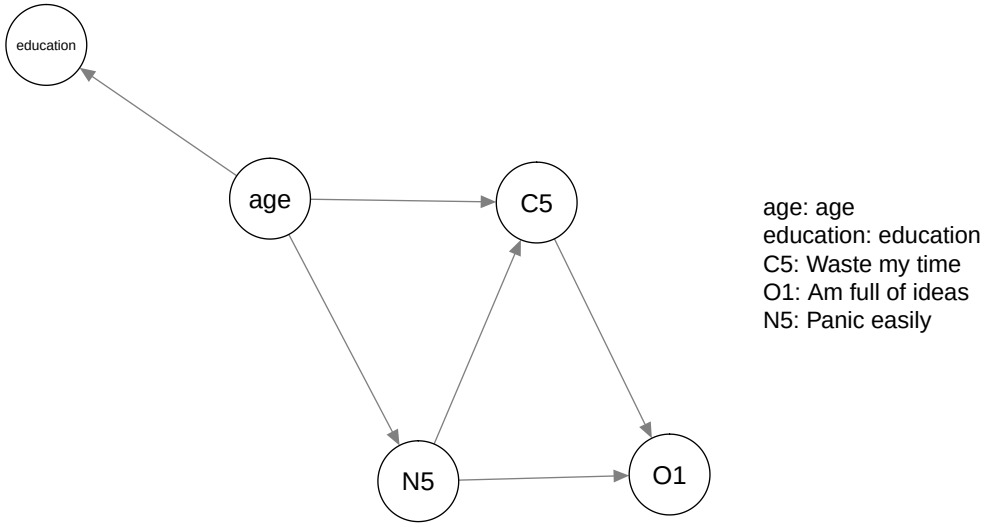
```
Labels <- c(
  "age",
  "education",
  "Waste my time",
  "Am full of ideas",
  "Panic easily"
  )

# Plot network:
qgraph(Res, nodeNames = Labels, legend.cex = 0.5)
```



While interesting, this structure tells us little about *how* the variables are assumed to cause each-other (e.g., positive or negative). To do that, we first need to fit the model and obtain model parameters. However, we have one edge (`N5 - C5`) that is not oriented. We need to orient edges before we can fit the model. From my vast experience in procrastinating, I expect that panicking easily leads to wasting your time:

```
# Set the edge (in bnlearn a directed edge is termed an "arc")::
Res <- set.arc(Res, from = "N5",to="C5")
graph <- qgraph(Res, nodeNames = Labels, legend.cex = 0.5)
```



Now we can fit the model:

```
fit <- bn.fit(Res, bfiSub)
```

We can investigate this model to check the direction of effect. First, for education:

```
fit$education

##
##   Parameters of node education (Gaussian distribution)
##
```

```
## Conditional density: education | age
## Coefficients:
##  (Intercept)           age
## -0.005177882   0.285258956
## Standard deviation of the residuals: 0.9582406
```

As expected, the model shows a positive regression slope of age on education (0.285). Next, for "Panic easily":

```
fit$N5

##
##   Parameters of node N5 (Gaussian distribution)
##
## Conditional density: N5 | age
## Coefficients:
## (Intercept)          age
##  0.02079959  -0.09530428
## Standard deviation of the residuals: 1.015181
```

Older people appear to panic less. Investigating "Waste my time":

```
fit$C5

##
##   Parameters of node C5 (Gaussian distribution)
##
## Conditional density: C5 | age + N5
## Coefficients:
##  (Intercept)           age            N5
##  0.007213843  -0.060814619   0.187782162
## Standard deviation of the residuals: 1.009865
```

Older people waste their time less, and people who panic easily waste their time more. Finally, for "Am full of ideas":

```
fit$O1

##
##   Parameters of node O1 (Gaussian distribution)
##
## Conditional density: O1 | C5 + N5
## Coefficients:
## (Intercept)           C5           N5
## -0.03405320  -0.06376441  -0.13866146
## Standard deviation of the residuals: 0.9873375
```

Both panicking easily and wasting your time seems to cause being less full on ideas!
We may wish to investigate our results for stability. To this end, we can bootstrap the model 1000 times (I get some warnings but I'll ignore them):

```
set.seed(1)
boot <- boot.strength(bfiSub, R = 1000, algorithm = "iamb",
        algorithm.args = list(whitelist = Whitelist, blacklist = Blacklist))
```

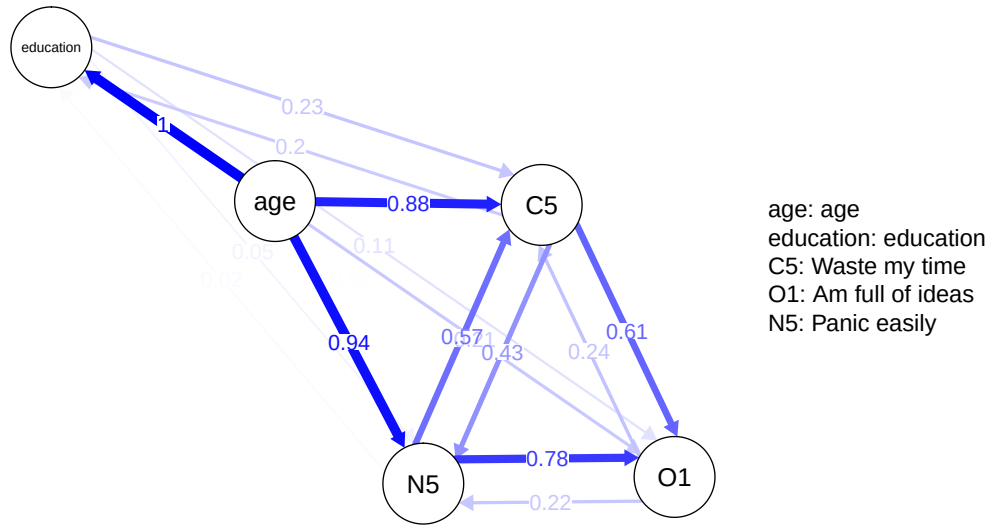
Printing the object gives of every edge that was estimated at least once the probability of including an edge (regardless of direction) between two nodes and the proportion that edge was in the given direction:

```
boot
```

```
##          from        to strength  direction
## 1         age education    1.000 1.00000000
## 2         age        C5    0.876 1.00000000
## 3         age        O1    0.210 1.00000000
## 4         age        N5    0.945 1.00000000
## 5   education       age    1.000 0.00000000
## 6   education        C5    0.431 0.52784223
## 7   education        O1    0.118 0.95338983
## 8   education        N5    0.063 0.74603175
## 9          C5       age    0.876 0.00000000
## 10         C5 education    0.431 0.47215777
## 11         C5        O1    0.847 0.71900826
## 12         C5        N5    1.000 0.43200000
## 13         O1       age    0.210 0.00000000
## 14         O1 education    0.118 0.04661017
## 15         O1        C5    0.847 0.28099174
## 16         O1        N5    1.000 0.22050000
## 17         N5       age    0.945 0.00000000
## 18         N5 education    0.063 0.25396825
## 19         N5        C5    1.000 0.56800000
## 20         N5        O1    1.000 0.77950000
```

This object is also supported by qgraph, which automatically combines the information of presence of an edge and direction into a single weight (proportion of times the specific directed edge was estimated):

```
qgraph(boot, nodeNames = Labels, legend.cex = 0.5,
       layout = graph$layout, edge.labels = TRUE)
```



This tells me that bnlearn was not certain about the direction between N5 and C5 (which we already knew), was a bit uncertain about the direction between C5 and O1, and more certain of other effects. Almost half the times the algorithm estimated an edge between education and C5 too. The edge age → education has a weight of 1, as it was whitelisted.

Exercise 2 (10 points)
Your task is to analyze a dataset using DAG estimation and to write a scientific report (max 1000 words) on your findings. Choose a set of variables (you can use variables that I did, but do not use exactly the

same set) you think causally interact with one-another. You may use the `bfiNoNA` dataset I created in this assignment, but if you have your own data to analyse that is also fine.[a] Select *at least* 4 variables. There is no limit to how many you may include, but note that interpretation becomes much much harder with many variables, so perhaps limit the number of nodes to 8 or less. *Before* analyzing your data, think of a DAG structure you hypothesize underlies your variables, and include a plot of your hypothesized structure in your report. Interpret *and critically discuss* your findings in your report, including how well your results align with your hypothesized structure, the direction of effect, and stability of results using bootstraps.
Some notes:

- You will be graded on writing in addition to the analysis, so write in full sentences. Note that a scientific report should include headers, figure captions and a reference list (if you have references)
- You may use any bnlearn algorithm you want
- Gender is a categorical variable which does not lead to regression coefficients. Instead, it shows regression coefficients of other effects for levels of gender. A higher intercept on one variable for females means that females score higher on that variable! This might be a bit more tricky to interpret though, so you may wish to avoid gender or treat is as continuous
- Do not include R codes in your essay, include them (with comments) as a separate .R file as usual
- Remember to be smart about whitelisting and blacklisting edges!

■

[a] Another similar personality inventory dataset is the HEXACO dataset available as supplementary of `http://www.sciencedirect.com/science/article/pii/S0092656614000701`

Part 3: Challenge (1 bonus point)

I would like to include a function to qgraph that can take an estimated bnlearn object (e.g., `Res`), fits the model to data, extracts the regression parameters as edge weights, and subsequently plots a weighted graph for me. The function should look something like this:

```
qgraph_bnfit <- function(
  bn, # bnlearn object
  data, # Dataset to fit to
  ... # arguments sent to qgraph
){
  # Magic R codes

  qgraph( SOMEGRPAH, ...) # the ... notation sends arguments from the function to qgraph
}
```

Ideally, the function should stop with an informative error if the user supplies a model with undirected edges, or ask the user which direction these should be specified in. For now I'll only look at continuous variables, so the function should also return an error if any variable is categorical. Can you make this function?