Matching students to advisers basic example

This is an R Markdown Notebook.

This is meant as a basic working example for student-adviser matching. As this is not a library, we have to copy the source of the function we use.

```
source("hungar.R")
source("do_match.R")
```

Read in data

At this moment it looks most elegant to do the data entry in a speadsheet and export/import the data via csv files. Those three files have a simple structure: nr, name for students, nr, name and max for advisers and a weights table. Other columns are possible but not used in this algorithm.

The weights table has as header the names (or nr's) of the advisors, and a row for each student with in the first column a student id (name or nr). Cells can be empty, have a positive weight for student preference and can be marked with a single letter. At this moment we have "F" for a forbidden combination and "G" for a combination where student and adviser do not share a common subject.

Here we process example files from the testdata folder.

```
weights<- read.csv("./testdata/testoverzicht.csv", header=TRUE,
                   row.names=1, stringsAsFactors = FALSE)
students<- read.csv("./testdata/students.csv",
                    stringsAsFactors = FALSE)
advisers<- read.csv("./testdata/advisers.csv",
                    stringsAsFactors = FALSE)
```

Post processing of the weights table

The next step is to replace some of the codes in the weights table by numbers and make the whole table numerical.

```
SUBJECT_MISMATCH = -50
TOO_MANY_STUDENTS = -100 ## overflow per advisor
FORBIDDEN_COMBINATION = -1000
SQUEEZE_POWER = 4 ## see below for more exlanations

cost <- weights
cost[ cost==""] <- 0
cost[cost=="G"] <- SUBJECT_MISMATCH
cost[cost=="F"] <- FORBIDDEN_COMBINATION
for (i in 1:ncol(cost))cost[,i] <-
      as.numeric(cost[,i])

if(sum(is.na(cost)) > 0){
  cat("\n\n*** Incomplete conversion")
  stop
}
```

Now we can do the matching. There is one more parameter at play here: SQUEEZE_POWER. Explanation:

Giving almost all students their prefered advisor but one student his/her least prefered advisor night raise some eyebrows, even though it maximizes the sum of values. If that effect is too gross, the high values can be "squeezed down" a little bit so that for instance 8 eights will weight more than 7 nines and 1 one. SQUEEZE_POWER of 1 is normal behaviour, use 2-10 for strong(er) equlizing effects.

In this example 6 students will not have a prefered advisor. After increasing this factor to 4 we have only 2 such students left and setting SQUEEZE_POWER to 9 there is none.

```
match_table<- do_match(cost, advisers$max,
                  penalty = TOO_MANY_STUDENTS,
                  squeeze_power = SQUEEZE_POWER)

final_result<- data.frame(
  nr= students$nr[match_table$nr],
  name= students$name[match_table$nr],
  adviser= advisers$name[match_table$adviser],
  weight = match_table$value
)
```

You might want to print or save the result

```
# print or save final result
# write.csv(final_result, "fr.csv", row.names= FALSE, quote= FALSE)


knitr::kable(final_result, col.names= names(final_result), row.names=FALSE, caption="Best Matching")
```

Table 1: Best Matching

nr	name	adviser	weight
1	Daan	ass3	3
2	Bram	ass5	1
3	Thijs	ass4	3
4	Mees	ass5	3
5	Stijn	ass4	3
6	Siem	ass5	1
7	Gijs	ass3	5
8	Jan	ass3	5
9	Teun	ass4	5
10	Noud	ass4	5
11	Tijn	prof1	5
12	Floris	ass1	3
13	Ties	prof2	-95
14	Joep	ass2	3
15	Niek	prof2	5
16	Pepijn	ass1	5
17	Koen	prof2	5
18	Thijmen	prof2	5
19	Fedde	ass5	0
20	Bas	ass1	5
21	Hidde	ass2	1
22	Pieter	ass3	1
23	Johannes	ass2	3

nr	name	adviser	weight
24	Joris	ass3	3
25	Jelle	ass2	3
26	Jip	ass2	5
27	Hendrik	ass1	5
28	Cornelis	ass1	5
29	Rens	ass2	5
30	Jelte	ass2	5
31	Melle	ass5	0
32	Wout	ass3	3
33	Duuk	ass5	1
34	Loek	ass4	3
35	Gerrit	ass5	3
36	Laurens	ass4	3
37	Matthijs	ass4	5
38	Tijs	ass3	5
39	Wessel	ass3	5
40	Bart	ass4	5
41	Thijn	ass5	1
42	Maarten	prof1	5
43	Dirk	prof2	5
44	Sebastiaan	ass1	3
45	Faas	ass2	3
46	Kees	ass1	3
47	Job	ass1	5
48	Tijmen	prof1	3
49	Nout	prof1	3
50	Moos	ass1	-95

Diagnostics

Next look at some other details of the matching.

```
diag1 <- aggregate(final_result[,3],
                   by=list(final_result$adviser),
                   FUN=length)
names(diag1)<- c("name", "nr_students")
diag1<- merge(advisers, diag1, by="name")

knitr::kable(diag1, caption="Adviser load")
```

Table 2: Adviser load

name	max	nr_students
ass1	8	9
ass2	8	8
ass3	8	8
ass4	8	8
ass5	8	8
prof1	4	4
prof2	4	5

```
diag2<- final_result[final_result$weight %in%
              c (0, SUBJECT_MISMATCH,
                 FORBIDDEN_COMBINATION),]
diag2$message <- c("not a prefered adviser",
                   "subject mismatch",
                   "forbidden combination")[
                     match(diag2$weight,
                             c (0, SUBJECT_MISMATCH,
                                FORBIDDEN_COMBINATION))
                     ]
knitr::kable(diag2, caption= "Unfortunate matches")
```

Table 3: Unfortunate matches

	nr	name	adviser	weight	message
19	19	Fedde	ass5	0	not a prefered adviser
31	31	Melle	ass5	0	not a prefered adviser