final

Hannah Cronin

2022-12-14

Success on a group project is more than just knowledge. It also includes group norms, dynamics, and cohesiveness. Due to this belief I chose the factors group preference, GPA, and current grade as my factors. I chose these three factors, because if given a choice, those who want to work together will choose to and may be happier with their placement and therefor more successful. Working with preferred groups can boost team moral and team chemistry if there is preexisting moral. Next, GPA is indicative of how the student on average has performed in the past. Generally, students who perform worse in classes may have lower GPA's. Finally, the last factor I chose is current grade in that class- this grade indicates knowledge, familiarity, and comfort with the subject matter.

Group preference has the highest weight of the three- I believe that group choice can lead to more cohesive groups especially in a higher learning setting. Students ranked their group preference from 1(lowest) to 4(highest). Next is GPA, GPA is not the highest weight, because a high GPA does indicate the student usually performs well in classes, but does not necessarily indicate that they have background knowledge for this project. If we assume that the project is in a class of students with similar backgrounds, like a specialised class for a specific major, then would GPA would be more indicative of background knowledge. This unknown is why I gave less weight to GPA than student choice. However, GPA has more weight than current grade in the class (GPA added to student coefficient at it's current value- high GPA were more influenced, while lower GPA's were less influenced). Lastly is current class performance, which had the lowest weight when calculating the decision variables coefficients. This is due to the fluid nature of class grades- a student could have an excellent grasp of the class material but may have done poorly on an inital assignment/quiz and therefore have a lower grade.'

To collect data, I would collect data from students in a class via a survey. If the population of the class exceeds 12, I would randomly sample 12 of the students for the model. I would use a software to do this to avoid cherry picking data. I believe using an existing class is a good model as taking a class is voluntary, so class demographics can vary wildly between sections or semesters based of each individual students choice to sigh up for that class or not.

In order to generate data for this project, I used generatedata.com (data.csv folder included in the GitHub as well) to randomly generate 12 students with GPA's ranging from 1.0 to 4.0, a random preference of groups, and class grade ranging from 0.5 to 1 (50-100%).

Coefficients were calculated by adding each of these factors up: preference (1-4) + GPA (assumed to be between 1.0 and 4.0) + Class Grade (Assumed to be between .5 (50%) and 1.00 (100%)). I will also attach the CSV file with the data set used to create the LP model.

Decision variables: Each student and each group

to create 48 decisions

Student1Group1 (S1G1)

Student1Group2 (S1G2)

Student1Group3 (S1G3)

Student1Group4 (S1G4)

Student2Group1 (S2G1)

Student2Group2 (S2G2)

Student2Group3 (S2G3)

Student2Group4 (S2G4)

Student3Group1 (S3G1)

Student3Group2 (S3G2)

Student3Group3 (S3G3)

Student3Group4 (S3G4)

Student4Group1 (S4G1)

Student4Group2 (S4G2)

Student4Group3 (S4G3)

Student4Group4 (S4G4)

Student5Group1 (S5G1)

Student5Group2 (S5G2)

Student5Group3 (S5G3)

Student5Group4 (S5G4)

Student6Group1 (S6G1)

Student6Group2 (S6G2)

Student6Group3 (S6G3)

Student6Group4 (S6G4)

Student7Group1 (S7G1)

Student7Group2 (S7G2)

Student7Group3 (S7G3)

Student7Group4 (S7G4)

Student8Group1 (S8G1)

Student8Group2 (S8G2)

Student8Group3 (S8G3)

Student8Group4 (S8G4)

Student9Group1 (S9G1)

Student9Group2 (S9G2)

Student9Group3 (S9G3)

Student9Group4 (S9G4)

Student10Group1 (S10G1)

Student10Group2 (S10G2)

Student10Group3 (S10G3)

Student10Group4 (S10G4)

Student11Group1 (S11G1)

Student11Group2 (S11G2)

Student11Group3 (S11G3)

Student11Group4 (S11G4)

Student12Group1 (S12G1)

Student12Group2 (S12G2)

Student12Group3 (S12G3)

Student12Group4 (S12G4)

Objective function: Maximize(S) = (see below for coefficients)

There are 16 constraints for this model: one for each student that limits their participation to one group (12) and one for each group that limits its number of participants to 3 (4).

```
library(lpSolveAPI)
```

#making the lp

```
group = make.lp(0,48)
```

#objective function

```
set.objfn(group, c(6.09, 7.09, 8.09, 5.09, 8.44, 6.44, 7.44, 5.44, 3.96, 4.96, 5.96, 6.9
6, 5.01,
                   4.01, 3.01, 6.01, 6.83, 3.83, 5.83, 4.83, 4.33, 5.33, 6.33, 7.33, 5.4
9, 6.49,
                   4.49, 7.49, 6.53, 4.53, 5.53, 3.53, 8.34, 5.34, 7.34, 6.34, 3.08, 4.0
8, 6.08,
                   5.08, 5.45, 6.45, 4.45, 3.45, 4.29, 3.29, 6.29, 5.29)) # 48 decision
variables
lp.control(group, sense = 'max' )
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
```

```
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```
set.type(group, 1:48, type=c("integer"))
```

#constraints

```
add.constraint(group, c(1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student1 can only be in one group
add.constraint(group, c(0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student2 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student3 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student4 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student5 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student6 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student7 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student8 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0), '=', 1) #Student9 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0), '=', 1) #Student10 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0), '=', 1) #Student11 can only be in one group
add.constraint(group, c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1), '=', 1) #Student12 can only be in one group
add.constraint(group, c(1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,
1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0), '=', 3) #Group1 has maximum of three students
add.constraint(group, c(0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,
0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0), '=', 3) #Group2 has maximum of three students
add.constraint(group, c(0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,
0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0), '=', 3) #Group3 has maximum of three students
add.constraint(group, c(0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,
0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1), '=', 3) #Group4 has maximum of three students
```

```
Colnames = c('S1G1','S1G2','S1G3','S1G4','S2G1','S2G2', 'S2G3', 'S2G4', 'S3G1', 'S3G2',
'S3G3', 'S3G4', 'S4G1', 'S4G2','S4G3','S4G4','S5G1', 'S5G2', 'S5G2', 'S5G3', 'S5G4', 'S6
G2', 'S6G3', 'S6G4', 'S7G1', 'S7G2', 'S7G3', 'S7G4', 'S8G1','S8G2', 'S8G3', 'S8G4', 'S9G
1', 'S9G2', 'S9G3', 'S9G4', 'S10G1', 'S10G2', 'S10G3', 'S10G4', 'S11G1', 'S11G2', 'S11G
3', 'S11G4', 'S12G1', 'S12G2', 'S12G3', 'S12G4')
Rownames = c('Student1','Student2','Student3','Student4','Student5','Student6','Student
7', 'Student8', 'Student9', 'Student10', 'Student11', 'Student12', 'Group1', 'Group2',
'Group3', 'Group4')
dimnames(group) = list(Rownames,Colnames)
```

#description of lp

```
group
```

```
## Model name:
##   a linear program with 48 decision variables and 16 constraints
```

#solving lp

```
solve(group)
```

```
## [1] 0
```

```
get.objective(group)
```

```
## [1] 81.84
```

```
get.variables(group)
```

```
##  [1] 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0
## [39] 1 0 0 1 0 0 0 0 1 0
```

```
get.constraints(group)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3
```

Group assignments: Student 1- Group 3 Student 2- Group 1 Student 3- Group 4 Student 4- Group 4 Student 5- Group 1 Student 6- Group 4 Student 7- Group 2 Student 8- Group 2 Student 9- Group 1 Student 10- Group 3 Student 11- Group 2 Student 12- Group 3

Every student except for 2 (students 7 & 8) got their top choice of group.