

final Exam

karthikeyan ramesh

2022-13-11

#I assumed that groups were assembling for a class for the objectives of this project.According to my a

#The total score based on all three criteria. Here, dividing the GPA by four would standardise the GPA,

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

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

```
##
```

```
## filter, lag
```

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

```
##
```

```
## intersect, setdiff, setequal, union
```

```
NR_DATA <- data.frame("Student_Name" = c(1:12), "GPA" = c(3.4, 3.7,3.5, 3.1, 3.1, 2.3, 3 , 3.2, 3.4, 3.1, 3.5, 3.2),  
  "activity" = c(10, 12, 15, 18, 20, 22, 25, 28, 30, 32, 35, 38),  
  "Participation" = c(80, 85, 90, 95, 100, 105, 110, 115, 120, 125, 130, 135),  
  "GPA_Coeff" = c(0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6),  
  "activity_Coeff" = c(0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8),  
  "Participation_Coeff" = c(0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65),  
  "GPA_Coeff" = c(0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6),  
  "activity_Coeff" = c(0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8),  
  "Participation_Coeff" = c(0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65))  
View(NR_DATA_Coeff)
```

#In order for the execution of the problem to make sense, restrictions on group size (three students per

#participation - In order to ensure that each group has high level of participation, a constraint was a

```
getwd()
```

```
## [1] "C:/Users/ASUS/Downloads"
```

```
library(lpSolveAPI)
```

```
NR <- read.lp("C:/Users/ASUS/Downloads/qmm-final.lp")
```

```
NR
```

```
## Model name:
```

```
## a linear program with 48 decision variables and 28 constraints
```

```
solve(NR)
```

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

```
get.objective(NR)
```

```
## [1] 1060
```

```
get.variables(NR)
```

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

```
get.constraints(NR)
```

```
## [1] 9.77 9.49 9.58 10.07 13.22 12.20 12.17 13.24 23.00 21.00 21.00 21.00  
## [13] 3.00 3.00 3.00 3.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00  
## [25] 1.00 1.00 1.00 1.00
```

```
#Optimal Groups based on output below:  
#Group1:Student 1, Student 4, Student 6  
#Group2:Student 3, Student 9, Student 12  
#Group3:Student 2, Student 7, Student 11  
#Group4:Student 5, Student 8, Student 10
```

```
““
```