ETM540 - Homework#5 - Projects Assigned based on Preferences

Mala Daryanani

October 30, 2018

Implement a model for students to do automated team assignment for senior design projects. There are 32 students and 6 projects. Each student has given their score of 0 to 10 to indicate how interested they are in the topic. Projects can have no more than 6 people on a single capstone project.

Solution:

Students preferences for each Project:

```
#n =no. of students
n <- 32
#m =no. of projects
m <- 6

# Each project has at the most 6 students
capacity <- rep.int(6, m)
data_preference <- read.csv("HW5data.csv")
pander(data_preference, caption = "Student to Project Preference")
```

Table 1: Student to Project Preference

X	Proj1	Proj2	Proj3	Proj4	Proj5	Proj6
1	6	2	3	3	1	5
2	2	3	7	6	3	7
3	3	4	8	6	3	4
4	6	0	9	3	3	3
5	7	7	9	6	7	3
6	2	8	9	6	7	3
7	1	4	0	9	8	7
8	7	9	5	9	9	4
9	9	3	5	9	3	3
10	10	3	5	8	9	6
11	3	6	5	6	9	7
12	2	9	3	8	4	8
13	3	0	3	8	4	1
14	6	5	3	8	4	3
15	8	7	3	9	4	3
16	3	6	3	9	8	3
17	8	6	7	5	9	5
18	1	6	7	4	6	5
19	3	8	8	3	6	6
20	1	5	5	1	4	3
21	6	5	5	8	2	3

X	Proj1	Proj2	Proj3	Proj4	Proj5	Proj6
22	9	5	5	5	1	3
23	3	5	6	3	5	3
24	5	5	2	3	3	6
25	3	1	3	3	3	4
26	7	5	3	3	3	3
27	4	5	4	5	5	3
28	3	3	3	6	3	3
29	6	3	4	6	6	3
30	3	4	3	4	6	3
31	1	4	3	3	6	3
32	6	4	1	3	6	3

```
data_preference <- data_preference[ ,2:7]

#function to call project score given by all students
student_pref <- function(student) data_preference[student,1:6]

#Creating matrix score [32,6]
student_i<- lapply(list(rep("Stud",n)),paste0,1:n)
project_j<- lapply(list(rep("Proj",m)),paste0,1:m)
score <- matrix(rep(-1.0, n), nrow=n, ncol=m)
dimnames(score)<-c(student_i,project_j)

#Fill score-matrix with student project ranking 1:6, 1-least preferred., 6-most preferred
for(i in seq(1, n, 1)){
  score[i,] <- rank(student_pref(i), na.last = TRUE, ties.method = "first")
}
```

Bar plot - Displaying individual student's preference/project:

```
library(ggplot2)
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:magrittr':
##
##     set_names
```

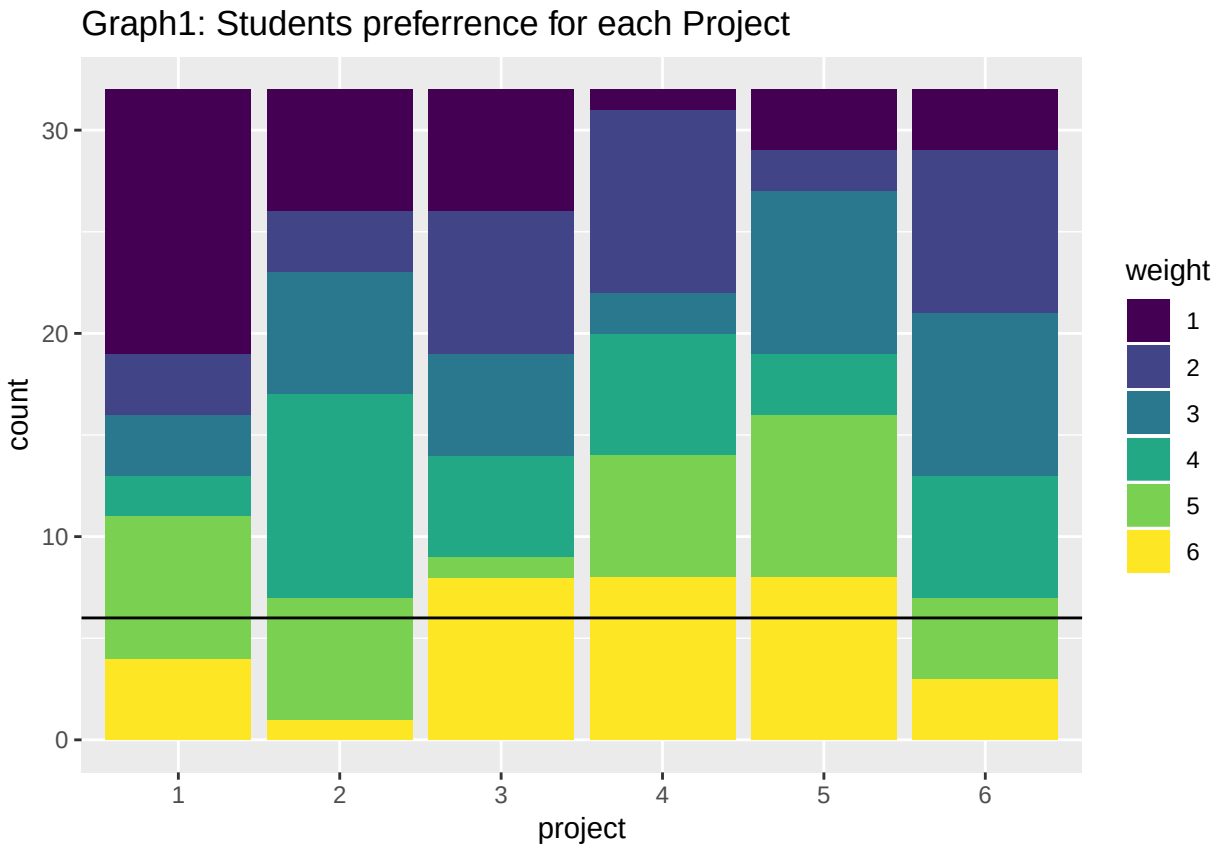
```
library(dplyr)

#Create 'weight' function to plot
weight <- function(student,project) score[student,project]

plot_data <- expand.grid(project = seq_len(m),weight = 1:6) %>%
  rowwise() %>%
  mutate(count = sum(map_int(seq_len(n), ~weight(.x, project) == weight))) %>%
  mutate(project = factor(project), weight = factor(weight))

ggplot(plot_data, aes(x = project, y = count, fill = weight)) +
```

```
  geom_bar(stat = "identity") +
  viridis::scale_fill_viridis(discrete = TRUE) +
  geom_hline(yintercept = 6)+
  ggtitle("Graph1: Students preferrence for each Project")
```



Points to note from above graph:

- X-axis: Refers to the 6 projects
- Y-axis: Total no. of Students (i.e 32)
- Bar-plot: Refers to the score provided by each student for that particular project
- Colors(Bar-plot): Each color represents the weightage (score) given by each student

As we see 'Yellow' color indicates the highest preferred score given by students. Looking at the above graph, we can see most students preferred to have project 3,4 & 5. Project 2 has the least score (indicated by 'Yellow' plot) which implies that very less students would like to do this project.

The horizontal line at y-axis = 6, indicates the restriction of each project to consist of not more than 6 students.

Model built with assigning projects based on individual's preferences:

Mathematical Representation:

$$(\#x_{ij} : Is\ i - Student,\ j - project$$

$$\text{Maximize} \sum_{j=1}^{m}\sum_{i}^{n} weight_{ij} * x_{i,j}$$

$$\text{Subject to} \sum_{i}^{n} x_{ij} \leq Capacity_j \ \forall\ j = 1, 2...m$$

$$\sum_{j}^{m} x_{ij} = 1 \ \forall\ i = 1, 2, ...n$$

$$x_{ij} \in \{0, 1\} \ \forall\ i, j$$

Model built in R

```
model <- MIPModel() %>%
  #binary variable x[student,project], assigned project(=1) or not(=0)
  add_variable(x[i,j], i=1:n, j=1:m, type = "binary") %>%

  #Maximize the preferences
  set_objective(sum_expr(sum_expr(score[i,j] * x[i,j], i=1:n), j=1:m),"max") %>%

  #Cannot exceed capacity of a project (=6 in this case)
  add_constraint(sum_expr(x[i, j], i = 1:n) <= capacity[j], j = 1:m) %>%

  #Each student assigned one project only
  add_constraint(sum_expr(x[i, j], j = 1:m) == 1, i = 1:n)
model
```

```
## Mixed integer linear optimization problem
## Variables:
##   Continuous: 0
##   Integer: 0
##   Binary: 192
## Model sense: maximize
## Constraints: 38
```

```
result <- solve_model(model, with_ROI(solver = "glpk", verbose = TRUE))
```

```
## <SOLVER MSG>  ----
## GLPK Simplex Optimizer, v4.65
## 38 rows, 192 columns, 384 non-zeros
##       0: obj =  -0.000000000e+00 inf =   3.200e+01 (32)
##      37: obj =   1.090000000e+02 inf =   0.000e+00 (0)
## *   122: obj =   1.860000000e+02 inf =   0.000e+00 (0)
## OPTIMAL LP SOLUTION FOUND
## GLPK Integer Optimizer, v4.65
## 38 rows, 192 columns, 384 non-zeros
## 192 integer variables, all of which are binary
## Integer optimization begins...
## Long-step dual simplex will be used
```

```
## +   122: mip =     not found yet <=              +inf        (1; 0)
## +   122: >>>>>   1.860000000e+02 <=   1.860000000e+02   0.0% (1; 0)
## +   122: mip =   1.860000000e+02 <=     tree is empty   0.0% (0; 1)
## INTEGER OPTIMAL SOLUTION FOUND
## <!SOLVER MSG> ----
```

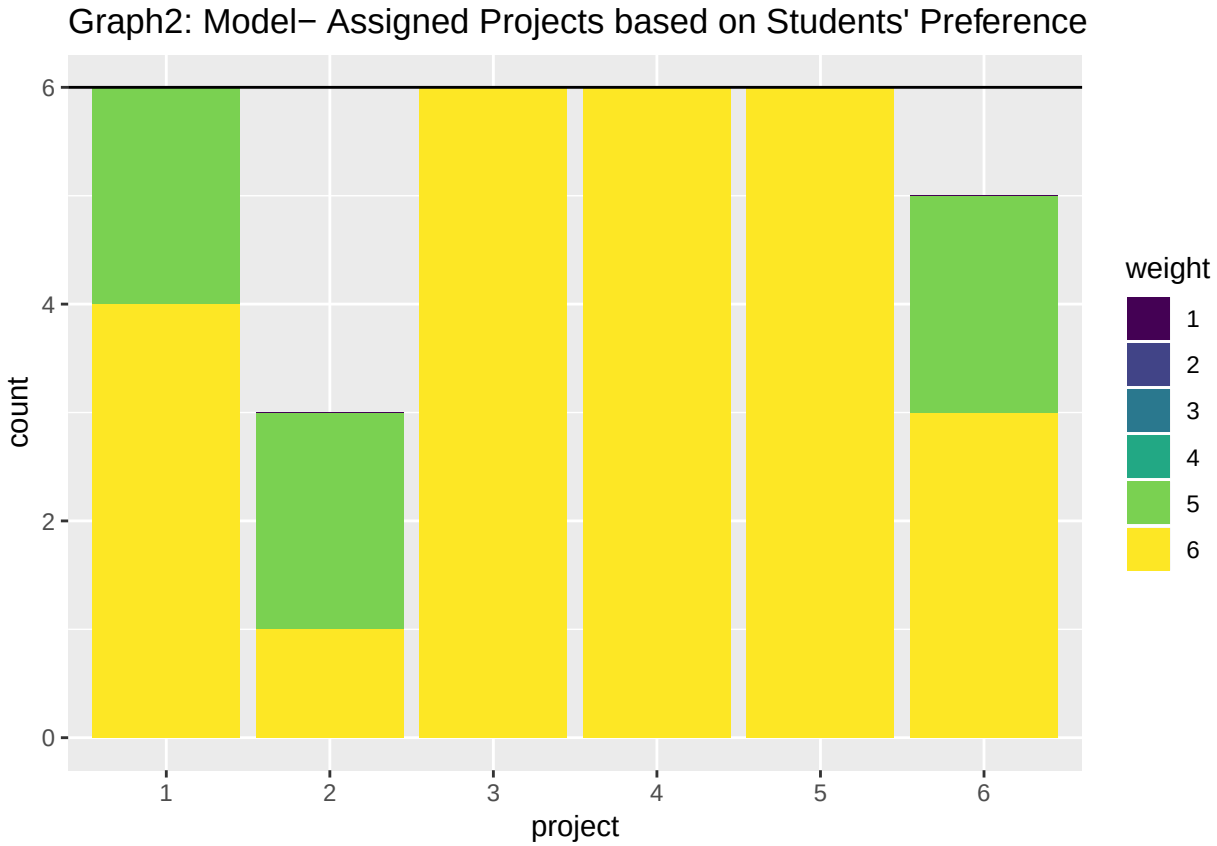
```
result
```

```
## Status: optimal
## Objective value: 186
```

Bar-plot - Model assigned projects based on given data of students' preferences:

```
matching <- result %>%
  get_solution(x[i,j]) %>%
  filter(value > .9) %>%
  select(i, j) %>%
  rowwise() %>%
  mutate(weight = weight(as.numeric(i), as.numeric(j)),
         student_pref = paste0(student_pref(as.numeric(i)), collapse = ",")) %>% ungroup
#head(matching)
#matching %>%
#  group_by(weight) %>%
 # summarise(count = n())

plot_data <- matching %>%
  mutate(project = factor(j), weight = factor(weight, levels = c(1,2,3,4,5,6))) %>%
  group_by(project, weight) %>%
  summarise(count = n()) %>%
  tidyr::complete(weight, fill = list(count = 0))
ggplot(plot_data, aes(x = project, y = count, fill = weight)) +
  geom_bar(stat = "identity") +
  viridis::scale_fill_viridis(discrete = TRUE) +
  geom_hline(yintercept = 6)+
  ggtitle("Graph2: Model- Assigned Projects based on Students' Preference ")
```



Points to note from Graph2:

- X-axis: Refers to the 6 projects
- Y-axis: Total no. of Students assigned per project ($Capacity \leq 6$)
- Bar-plot: Refers to the projects alloted to students based on their weightage (preferred scores)
- No. of students/project: Proj1: 6 students, Proj2: 3 students, Proj3: 6 students, Proj4: 6 students, Proj5: 6 students, Proj6: 5 students

'Yellow'-indicates highest score given by students, then follows 'Green', and so on. The model was designed considering the students preferrence andso the above graph indicates the allocations accordingly. The solution of model has given almost every student what they preferred (indicated in 'Yellow' - weight/score/rank=6).

Once the highest scored projects were exhausted, the model assigned the next most preffered projects (indicated in 'Green' - weight/score/rank=5).

Concluding, the model distributed all 6 projects into 32 students based on what each student preferred. Almost each student is assigned with their choice except for:

- Project1- 2 student were allocated with their second preferrence
- Project2- 2 student were allocated with their second preferrence
- Project6- 2 student were allocated with their second preferrence