Project 1

a.

Inputs: application per seat and accepted application: The product of application per seat and accepted application can be used to indicate how competitive is the program among other candidates. The higher the value the more competitive the program are and less likely for candidate to be selected.

Assume each school wants to optimise their efficiency. University wants to fill seats so that they receive sufficient found. They may issue number of offers higher than planned capacity (number of seats): hens have:

$$totall\ applicant\ \times\ offer\ rate\ \times\ applicant\ acceptance\ rate\ = number\ of\ seats$$

Subtract *number of seats* on both side of equation:

$$applicant\ per\ seat\ \times\ offer\ rate\ \times\ acceptance\ rate = 1$$

Transform equation:

$$offer\ rate = \frac{1}{acceptance\ rate \times applicant\ per\ seat}$$

Offer rate indicate the amount of applicant that are likely to issue an offer. According to classic probability formula:
$$P(being\ issued\ an\ offer) = offer\ rate$$
$$= \frac{1}{acceptance\ rate \times applicant\ per\ seat}$$

One would wish probability as high as possible or, product of acceptance rate and applicate per seat as low as possible. A trait to be considered as input.

Outputs: Average Pay and Employment Rate: The product of those ratio should indicate the expected value salary of an applicant. Having this:

$$E(value\ salary) = average\ salary\ \times\ employment\ rate$$

Above formula reflect the fact that the higher the value, the more desirable. A trait to be considered as output.

b.

Conducting DEA in R to have following efficiency score:

```r
> DEA[,c(1,6)]
# A tibble: 10 x 2
     X1        Efficiency
   <chr>          <dbl>
 1 Chicago          1
 2 Tuck             1
 3 wharton          1
 4 columbia         0.906
 5 NYU              1.
 6 UCLA             0.990
 7 MIT              0.834
 8 berkeley         0.962
 9 harvard          0.730
10 standford        0.651
```

```r
library("rDEA")
library("dplyr")
#install.packages("rDEA")
```

```r
DEA <- read_csv("~/Desktop/Operation/DEA.csv")

DEA$`average pay` <- gsub("\\$", '', DEA$`average pay`)
DEA$`average pay` <- gsub("\\s",'', DEA$`average pay`)
DEA$`average pay` <- gsub(",",'', DEA$`average pay`)
DEA$`average pay` <- as.numeric(DEA$`average pay`)

X <- DEA[,2:3]
Y <- DEA[,4:5]

di_naive = dea(XREF=X, YREF=Y, X=X, Y=Y, model="input", RTS="constant")
DEA$Efficiency <- di_naive$thetaOpt

DEA[,c(1,6)]
```

This is results is the same via *Expected Value* method:

	Best Value			Possibility of Secure a Place		Expected Value	application per seat	accepted applicatic	average pay		employment	Actual Candidate En	Original Ranking
Chicago	$	31,276.14	=	23%	×	$ 133,761.78	7.2	59.4	$ 137,615.00		97.2	8%	8
Tuck	$	29,430.28		22%		$ 133,654.68	8.7	52.2	$ 142,489.00		93.8	6%	9
wharton	$	28,231.31		21%		$ 136,300.74	7.1	68	$ 142,574.00		95.6	10%	5
columbia	$	23,061.35		18%		$ 126,634.47	7.8	70.4	$ 139,006.00		91.1	9%	6
NYU	$	22,329.86		18%		$ 122,883.43	11.3	48.7	$ 135,933.00		90.4	4%	7
UCLA	$	20,036.89		18%		$ 112,996.01	11.7	48.2	$ 127,535.00		88.6	4%	10
MIT	$	18,197.67		14%		$ 132,644.61	11.7	62.3	$ 142,936.00		92.8	5%	3
berkeley	$	16,162.78		13%		$ 122,190.65	14.4	52.5	$ 140,935.00		86.7	4%	4
harvard	$	14,287.06		11%		$ 129,406.50	10.2	88.8	$ 144,750.00		89.4	9%	2
standford	$	9,338.21		7%		$ 131,550.11	17.9	78.7	$ 142,834.00		92.1	4%	1

c.

The efficiency score suggests the most efficient university is Chicago, followed by Tuck and Wharton Business School. However, it is reasonable to recommend university with lower chance of securing a place but higher expected value (*employability × expected salary*)

Chicago: best value. As according to calculation, *Chicago* has the best value. It is more likely secure a place at *Chicago* and have relatively higher expected value. Potential applicant is recommended to apply for *Chicago*.

Warton: best employment prospect. The next is *Wharton* instead of *Tuck*. Wharton has higher expected value. Wharton graduates are more likely to secure a role after graduate and they expect higher salary.

Harvard: highest potential salary. Graduate securing a job would expect receiving the highest average pay graduating from Harvard university. However, comparing with other university, Harvard is among Universities' who least likely to issue an offer.

Stanford: Sandford is not recommended. Although graduate expects higher salary as well as batter change of employment, the probability of securing a place is only 7%, comparing to other
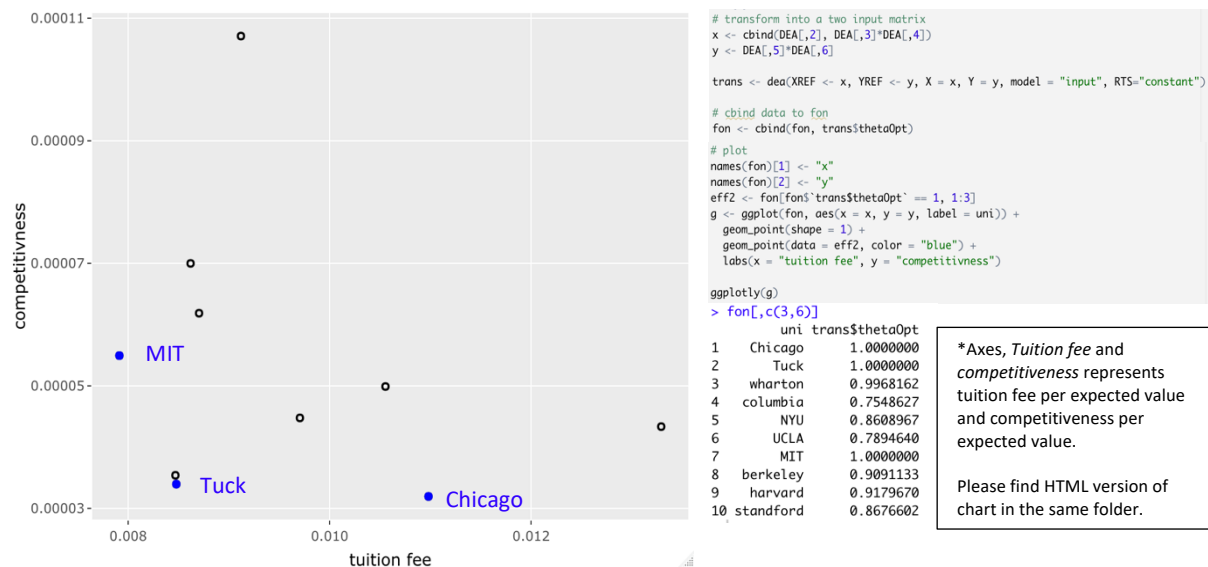
d.

To enhance recommendations, one may consider MBA tuition fees for each business school. Based on analysis in ***a*.**, one can reduce the dimension of the data into 2 input verses 1 output.

Input_1: Tuition fee
Input_2: "Competitiveness" = *Application per seat* × accepted applicant

Output: "Expected value" = employment rate* average pay

Results following output from R:


```
# transform into a two input matrix
x <- cbind(DEA[,2], DEA[,3]*DEA[,4])
y <- DEA[,5]*DEA[,6]

trans <- dea(XREF <- x, YREF <- y, X = x, Y = y, model = "input", RTS="constant")

# cbind data to fon
fon <- cbind(fon, trans$thetaOpt)
# plot
names(fon)[1] <- "x"
names(fon)[2] <- "y"
eff2 <- fon[fon$`trans$thetaOpt` == 1, 1:3]
g <- ggplot(fon, aes(x = x, y = y, label = uni)) +
    geom_point(shape = 1) +
    geom_point(data = eff2, color = "blue") +
    labs(x = "tuition fee", y = "competitivness")

ggplotly(g)
> fon[,c(3,6)]
```

	uni	trans$thetaOpt
1	Chicago	1.0000000
2	Tuck	1.0000000
3	wharton	0.9968162
4	columbia	0.7548627
5	NYU	0.8608967
6	UCLA	0.7894640
7	MIT	1.0000000
8	berkeley	0.9091133
9	harvard	0.9179670
10	standford	0.8676602

*Axes, *Tuition fee* and *competitiveness* represents tuition fee per expected value and competitiveness per expected value.

Please find HTML version of chart in the same folder.

The frontier graph suggests either MIT or Tuck or Chicago are considered as most efficient in terms of competitiveness or tuition fee. If one favours cheaper tuition fee over competitiveness, they MIT may be better choice; conversely if tuition fee is not a problem one can apply for Chicago for being less competitive.

e.

Project 2

a.

Solver shows schedule hour that minimise working hours:

	aerospace	transport	computer	Total		
project manager	0	2	8	10	<=	10
data analyst 1	6	2	0	8	<=	8
data analyst 2	8	0	0	8	<=	8
intern	0	6	0	6	<=	6
	14	10	8			
	>=	>=	>=			
requirements	14	10	8			
Total hour	620	880	480	**1980**		

b.

6 aerospace projects maximum

	aerospace	transport	computer	Total		
project manager	0	2	8	10	<=	10
data analyst 1	**0**	2	0	**2**	<=	**8**
data analyst 2	0	0	0	0	<=	0
intern	0	6	0	6	<=	6
sum	0	10	8			
		>=	>=			
requirements	0	10	8			
Total hour	0	880	480	**1360**		

Setting constraints for data analyst 2 as 0. Optimising total hour shows this will leaves data analyst 1to have 6 (8 – 2) extra space. Both transport and computer jobs are maximised to fit requirement. Analyst 1 can get 6 aerospace report done.

c.

No, it does not change scheduling results.

Ask excel solver to seek minimal value after adding weights wage to column, the solver report similar results:

	aerospace	transport	computer	Total		
project manager	0	2	8	10	<=	10
data analyst 1	6	2	0	8	<=	8
data analyst 2	8	0	0	8	<=	8
intern	0	6	0	6	<=	6
	14	10	8			
	>=	>=	>=			
requirements	14	10	8			
Labour cost for each project	80600	99800	86400			
total Cost	**266800**					

Project 3

a.

Obtain following results by using `igraph` in R (chart visualised using `vizNetwork`.
Maximum capacity is 8. Via 4 routes, as illustrated in following graph. (BEMM461)
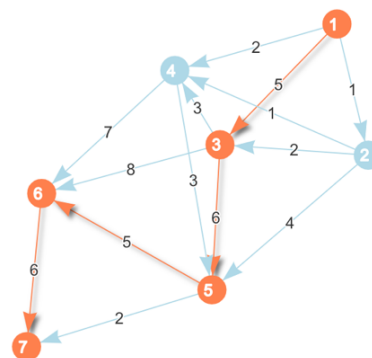Analytics And V...

      1 -> 4 -> 6 -> 7
      1 -> 4 -> 5 - > 7
      1 -> 3 -> 6 ->7
      1 -> 2 -> 5 -> 7
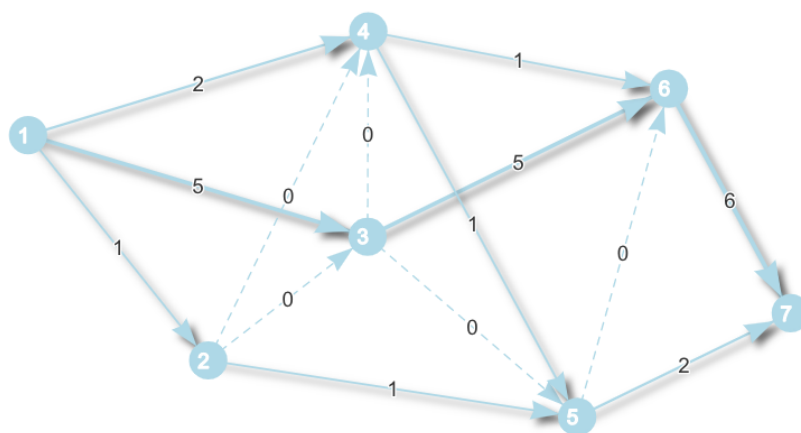
b.

Idea investment sequences would be 1, 3, 5, 6, 7.

Network Structure

origin	destination	flow	capacity
1	2	0.00	1
1	3	1.00	5
1	4	0.00	2
2	5	0.00	4
2	4	0.00	1
2	3	0.00	2
3	4	0.00	3
3	5	1.00	6
3	6	0.00	8
4	5	0.00	3
4	6	0.00	7
5	6	1.00	5
5	7	0.00	2
6	7	1.00	6

Total Single	22.000022

Flow Balance

Note	Netoutflow		Required net ou
1	1.000001	=	1
2	0	=	0
3	0	=	0
4	0	=	0
5	0	=	0
6	0	=	0
7	-1.000001	=	-1

See visualisation

c.