

Assignment Module 9

Riba Khan

2022-11-20

#loading the library and uploading the lp file.

```
AP_HUB <- read.lp("AP_HUB.lp")
print(AP_HUB)
```

## Model name:	x1	x2	x3	x4	x5	x6	x7		
## Minimize	775	800	800	800	800	775	750		
## Sunday	0	1	1	1	1	1	0	>=	18
## Monday	0	0	1	1	1	1	1	>=	27
## Tuesday	1	0	0	1	1	1	1	>=	22
## Wednesday	1	1	0	0	1	1	1	>=	26
## Thursday	1	1	1	0	0	1	1	>=	25
## Friday	1	1	1	1	0	0	1	>=	21
## Saturday	1	1	1	1	1	0	0	>=	19
## Kind	Std	Std	Std	Std	Std	Std	Std		
## Type	Int	Int	Int	Int	Int	Int	Int		
## Upper	Inf	Inf	Inf	Inf	Inf	Inf	Inf		
## Lower	0	0	0	0	0	0	0		

the estimate of number of workers needed each day of the week

```
Number_of_workers_and_days <- matrix(c("Sunday", "Monday", "Tuesday", "Wednesday",
                                         "Thursday", "Friday", "Saturday",
                                         18, 27, 22, 26, 25, 21, 19),
                                       ncol = 2, byrow = F)
```

#setting up the coloumn names

```
colnames(Number_of_workers_and_days) <- c("Days", "Workers_Required")
as.table(Number_of_workers_and_days)
```

##	Days	Workers_Required
## A	Sunday	18
## B	Monday	27
## C	Tuesday	22
## D	Wednesday	26
## E	Thursday	25
## F	Friday	21
## G	Saturday	19

#Package handlers at AP are guaranteed a five-day work week with two consecutive days off. #The base wage for the handlers is \$750 per week. Workers working on Saturday or Sunday receive an #additional \$25 per day.

#The possible shifts and salaries for package handlers are:

#creating a table by entering the necessary information

```
Shift_and_Wages <- matrix(c(1,2,3,4,5,6,7,
                             "Sunday and Monday", "Monday and Tuesday",
                             "Tuesday and Wednesday", "Wednesday and Thursday",
                             "Thursday and Friday", "Friday and Saturday",
                             "Saturday and Sunday",
                             "$775", "$800", "$800", "$800", "$800", "$775", "$750"),
                             ,
                             ncol = 3, byrow = F)
```

#setting up the coloumn names and creating a table

```
colnames(Shift_and_Wages) <- c("Shift", "Days_Off", "Wage")
as.table(Shift_and_Wages)
```

	Shift	Days_Off	Wage
## A	1	Sunday and Monday	\$775
## B	2	Monday and Tuesday	\$800
## C	3	Tuesday and Wednesday	\$800
## D	4	Wednesday and Thursday	\$800
## E	5	Thursday and Friday	\$800
## F	6	Friday and Saturday	\$775
## G	7	Saturday and Sunday	\$750

#solving the problem

```
solve(AP_HUB)
```

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

```
get.objective(AP_HUB)
```

```
## [1] 25675
```

ANSWER: The total cost is \$25675.

#let x1, x2, x3, x4, x5, x6, x7 represents the number of workers required for #all the seven shifts respectively

```
get.variables(AP_HUB)
```

```
## [1] 2 4 5 0 8 1 13
```

#From the above, we can derive:-

#x1 = 2, which implies 2 workers are needed for Shift 1

#x2 = 4, which implies 4 workers are needed for Shift 2.

#x3 = 5, implies 5 workers are needed for Shift 3.

#x4 = 0, implies 0 workers are needed for Shift 4.

#x5 = 8, implies 8 workers are needed for Shift 5.

#x6 = 1, implies 1 worker are needed for Shift 6.

#x7 = 13, implies 13 workers are needed for Shift 7.

#the workers available for each day is

```
Workers_Available_each_day <- matrix(c(0,4,5,0,8,1,0,0,0,5,0,8,1,13,2,0,0,0,
                                         8,1,13,2,4,0,0,8,1,13,2,4,5,0,0,1,13,2
                                         ,
                                         3,4,0,0,0,13,2,4,5,0,8,0,0),
                                     ncol=7,byrow=TRUE)
```

#setting up the coloumn names

```
colnames(Workers_Available_each_day)<- c("Shift1", "Shift2", "Shift3",
                                         "Shift4", "Shift5", "Shift6", "Shift7"
                                         ")")
```

```
row.names(Workers_Available_each_day) <- c('Sunday', 'Monday', 'Tuesday', 'Wednesda', 'Thursday', 'Friday', 'Saturday')
```

Workers_Available_each_day

##	Shift1	Shift2	Shift3	Shift4	Shift5	Shift6	Shift7
## Sunday	0	4	5	0	8	1	0
## Monday	0	0	5	0	8	1	13
## Tuesday	2	0	0	0	8	1	13
## Wednesday	2	4	0	0	8	1	13
## Thursday	2	4	5	0	0	1	13
## Friday	2	3	4	0	0	0	13
## Saturday	2	4	5	0	8	0	0

#showing number of workers available each day

```
rowSums(Workers_Available_each_day)
```

##	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
##	18	27	24	28	25	22	19

#We got the optimal solution.