Integer Programming Assignment

AP is a shipping service that guarantees overnight delivery of packages in the continental US. The company has various hubs at major cities and airports across the country. Packages are received at hubs, and then shipped to intermediate hubs or to their final destination.

The manager of the AP hub in Cleveland is concerned about labor costs, and is interested in determining the most effective way to schedule workers. The hub operates seven days a week, and the number of packages it handles varies from one day to another.

Setting default values to get a clean output

```
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
```

Loading the lpSolveAPI Package

```
library("lpSolveAPI")
```

Loading 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 table below provides an estimate of the number of workers needed each day of the week.

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

colnames(Day_Wise_Workers_Req) <- c("Day_of_the_week", "Workers_Required")

as.table(Day_Wise_Workers_Req)
```

```
##   Day_of_the_week 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:

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

colnames(Day_offs_and_wages) <- c("Shift", "Days_Off", "Wage")

as.table(Day_offs_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
```

Running the lp model

```
solve(ap_hub)
```

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

By getting 0 as the value we get to know that there exists a model.

Total Cost - Objective Function

```
get.objective(ap_hub)
```

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

The total cost to the firm thereby ensuring that the total wage expenses are as low as possible and there are sufficient number of workers available each day to work is "25,675$".

How many workers are available each day to work - Variables

```
get.variables(ap_hub)
```

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

The variables are labeled from x1, x2.....x7 where,
x1 = Number of workers assigned to shift 1 = 2
x2 = Number of workers assigned to shift 2 = 4
x3 = Number of workers assigned to shift 3 = 5
x4 = Number of workers assigned to shift 4 = 0
x5 = Number of workers assigned to shift 5 = 8
x6 = Number of workers assigned to shift 6 = 1
x7 = Number of workers assigned to shift 7 = 13

By the variable values attained we can thereby get to see how many workers are available to work each day with respect to the objective function as well as the constraints framed by the organization,

Sunday = x2 + x3 + x4 + x5 + x6 = 18 Workers
Monday = x3 + x4 + x5 + x6 + x7 = 27 Workers
Tuesday = x4 + x5 + x6 + x7 + x1 = 24 Workers
Wednesday = x5 + x6 + x7 + x1 + x2 = 28 Workers
Thursday = x6 + x7 + x1 + x2 + x3 = 25 Workers
Friday = x7 + x1 + x2 + x3 + x4 = 24 Workers
Saturday = x1 + x2 + x3 + x4 + x5 = 19 Workers