Project 1

Akankshi Mody, Catherine Yu Miao, Rachel Meade

PART 1:

QUESTION 1:

Formulate the budget allocation as a linear programming problem. List and describe the decision variables, the objective, and all the constraints.

ANSWER 1:

Choose (a,b,c,d,e,f,g,h,i,j)

where each of the above represents an amount (in millions) to invest in each advertising category

To Maximize:

(3.1%)a + (4.9%)b + (2.4%)c + (3.9%)d + (1.6%)e + (2.4%)f + (4.6%)g + (2.6%)h + (3.3%)i + (4.4%)j

Subject to the constraints:

1. a,b,c,d,e,f,g,h,i,j >= 0 (Non-negative constraint)
2. a,b,c,d,e,f,g,h,i,j <= 3 (Upper-bound contstraint)
3. a + b + c + d + e + f + g + h + i + j >= 0 (Budget constraint)
4. a + b <= e + j (Print & TV no more than Facebook & Email)
5. e + f + g + h + i >= 2(c + d) (Social Media at least 2x SEO & AdWords)

QUESTION 2:

Use the following (not shown) test case and solve the LP in R.

ANSWER 2:

Refer to .R file for code.

Objective Value = 0.456

Allocation = [0, 3, 0, 1, 0, 0, 3, 0, 0, 3]

QUESTION 3:

Next, we will write a function in R that can construct an allocation for any ROI vector, upper bound, and budget.

ANSWER 3:

```
allocation_g29 <- function(ROI_vec,budget,upper_bound=budget) {
        library(lpSolve)

        A = matrix(0,nrow=23,ncol=10)
        A[1,1:10] = 1
        A[2:11,1:10]=diag(10)
        A[12:21,1:10]=diag(10)
        A[22,1:10]=c(0,0,-2,-2,1,1,1,1,1,0)
        A[23,1:10]=c(1,1,0,0,-1,0,0,0,0,-1)

        b2 = rep(upper_bound,10)
        b3 = rep(0,12)
        B = c(budget,b2,b3)

        C = ROI_vec

        dir1 = rep('<=',11)
        dir2 = rep('>=',11)
        dir3 = '<='
        dir = c(dir1,dir2,dir3)

        alc=lp("max",C,A,dir,B, compute.sens = 1)
        list(objval = alc$objval, sol = alc$solution)
 }
```

QUESTION 4:

Use allocation() function to calculate the optimal objective value without the 3rd constraint.

Report both the optimal objective value and optimal solution, which we call alc2, and compare them with the $3M counterparts.

ANSWER 4:

Use function defined above but without upper_bound constraint.

Objective Value = $0.465
Solution = [0, 5, 0, 0, 0, 0, 0, 0, 0, 5]

PART 2

QUESTION 1: Get the optimal objective value (with $3M upper bound) and the corresponding solution, alc3, using the new ROI vector. Are they the same as their counterparts using the previous ROI vector?

ANSWER 1:

alc1$solution

alc3$sol

No, the solution is not the same. The solution based on the initial ROI Vector was (0 3 0 1 0 0 3 0 0 3), while the solution based on the new ROI vector is (3 0 0 1 3 3 0 0 0 0). However, both of these allocation/ROI combinations result in the same objective value of $0.456 MM.

QUESTION 2: The disappointment of an allocation is defined as the difference between the objective values using the old and new ROI vector. Calculate the disappointment for alc1 and alc2 in the previous section. Do you think the 3rd constraint based on CMO's experience is valuable?

disappoint_alc1 = sum(ROI*alc1$solution) - sum(ROI_new*alc1$solution)

disappoint_alc1 = 0.192

disappoint_alc2 = sum(ROI*alc2$sol) - sum(ROI_new*alc2$sol)

disappoint_alc2 = 0.22

Analysis: Although it may be tempting to use all of the advertising budget on the categories with the highest ROI, companies may experience diminishing returns. This means that although a company might expect a 4.6% ROI on Instagram for the first $3MM spent, they may not be able to capture the same ROI at a different spending level. The level at which these diminishing returns might affect the budget allocation decision is unclear, however. Additionally, companies may not be able to perfectly predict the returns from each advertising category, therefore, they may be better served by diversifying their budget allocations, rather than focusing on just a few platforms.

QUESTION 3: Try to find an allocation that dominates alc1, alc2, and alc3 regarding the average objective values using both the old and new ROI vector. To illustrate, the average objective values of alc1 is the average of $0.456M (ROI old) and $0.264M (ROI new), namely $0.360M. There are at least two possible ways to achieve this allocation.

See .R file for code.

First, we found the average return per optimal allocation using both the old ROI vector and the new ROI vector. Then, we found the various objective values, experimenting with various upper bounds.

The best upper bound was $2.5 MM per platform. Thus, we concluded that the allocation

(0, 2.5, 0, 1.667, 0, 0, 2.5, 0, 0.833, 2.5) dominates alc1, alc2, alc3.

The average return across both ROI vectors is $0.36167 MM.

PART 3

QUESTION 1:

```
allocation <- function(ROI_vec, upper_bound, budget){
  A = rbind(matrix(c(1,0,1,0,0,2,0,2,-1,-1,0,-1,0,-1,0,-1,0,-1,-1,0),2,10),
         matrix(c(rep(1,10)),1,10),diag(10))
  b <- c(0,0,budget,c(rep(upper_bound,10)))
  C = ROI_vec
  dir = rep("<=", 13)
  s = lp("max", C, A, dir, b)
  sol = s$solution
  objval = s$objval
  return(list("sol"=sol, "objval"=objval))
}

Budget = 10
alc_per_month = c()
Return = 0
for (i in c(1:12)){
  result = allocation(ROI_mat[i,],3,Budget)
  R_mat = result$sol*ROI_mat[i,]
  Profit = sum(R_mat)*0.01
  print(Profit)
  Budget = Budget + Profit*0.5
  alc_per_month = rbind(alc_per_month,result$sol)
  Return = Return + result$objval*0.01
}
```

The optimal allocation for the whole year is in the matrix below:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	3.000000	0	0	1.3333333	0.000000	0.0000000	2.666667	0	0.000000	3.000000
[2,]	3.000000	0	0	2.3955000	3.000000	0.0000000	0.000000	0	1.791000	0.000000
[3,]	0.000000	0	0	3.0000000	0.000000	3.0000000	1.389648	0	3.000000	0.000000
[4,]	0.000000	0	0	3.0000000	0.000000	3.0000000	3.000000	0	1.596856	0.000000
[5,]	1.804100	0	0	0.0000000	0.000000	0.0000000	3.000000	0	3.000000	3.000000
[6,]	3.000000	0	0	0.0000000	0.000000	0.0000000	3.000000	0	2.020172	3.000000
[7,]	0.000000	0	0	3.0000000	2.247555	0.0000000	3.000000	0	3.000000	0.000000
[8,]	3.000000	0	0	1.8272941	0.000000	0.6545882	0.000000	0	3.000000	3.000000
[9,]	1.362933	0	0	3.0000000	0.000000	3.0000000	0.000000	0	3.000000	1.362933
[10,]	0.000000	0	0	3.0000000	0.000000	3.0000000	3.000000	0	0.000000	2.955475
[11,]	3.000000	0	0	2.0564210	0.000000	1.1128419	3.000000	0	0.000000	3.000000
[12,]	3.000000	3	0	0.4279507	3.000000	0.0000000	0.000000	0	0.000000	3.000000

Under the optimal allocation, the return for the whole year is: $5.372735 MM.

Question 2:

We are adopting the same strategy in solving the multi-period and single period optimization problem. While resolving the multi-period allocation problem, we assign a new budget to the current month based on the initial budget as well as the return from the prior month. In a word, the budget is dynamic in the multi-period allocation problem.

Question 3:

In order to maintain the relationship with each platform, we need to add another constraint to this optimization problem. Specifically, for each platform, the budget difference for two adjacent months cannot be greater than $1M i.e. the absolute value of budget difference for the same platform for two consecutive months should not be over $1M. Except this constraint, the connection between the multi-period and single-period problem still holds.