Optimization
Project 1-Predicting Advertising Spend

Yash Warty, Jared Hurwit, Sam LaPlatney, Debtanu Bandopadhyay

Problem Statement: Formulating a marketing budget for our company across 10 channels, while maximizing return on investment.

Available Channels: All channels with their current ROI's are shown below:

Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
ROI	3.1%	4.9%	2.4%	3.9%	1.6%	2.4%	4.6%	2.6%	3.3%	4.4%

Key Constraints: We have the following limitations while deciding our budget allocations for the various channels:

1. The total amount that can be invested by us is $10M.
2. The amount invested in print and TV should be no more than the amount spent on Facebook and Email.
3. The total amount used in social media (Facebook, LinkedIn, Instagram, Snapchat, and Twitter) should be at least twice of SEO and AdWords.
4. For each platform, the amount invested should be no more than $3M.

In preparation for running the model, we loaded in the ROI csv file and then created a dictionary of all the available channels, so we can use it to print the allocations by channel at a later stage.

	Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
0	ROI	0.031	0.049	0.024	0.039	0.016	0.024	0.046	0.026	0.033	0.044
1	Second Firms ROI Estimate	0.049	0.023	0.024	0.039	0.044	0.046	0.026	0.019	0.037	0.026

```
#Creating a dictionary with all the available channels so as to print results correctly at a later stage
channel = list(ROI.columns)[1:]
channel_idx = {}
for idx, char in enumerate(channel):
    channel_idx[char] = idx
channel_idx
```

```
{'Print': 0,
 'TV': 1,
 'SEO': 2,
 'AdWords': 3,
 'Facebook': 4,
 'LinkedIn': 5,
 'Instagram': 6,
 'Snapchat': 7,
 'Twitter': 8,
 'Email': 9}
```

With respect to the model, the values in the objective array were derived from the percentages of the ROIs for the 10 different media platforms provided by our file.

```
#Q1
# Defining the objective vector
obj1 = np.array(ROI.iloc[0][1:])

```

Question 3

Since there are 13 constraints and 10 is the number of channels, we set the zeros of our function to these numbers.

```
# initializing & defining the constraint matrix
#We have 13 total constraints across 10 variables
A = np.zeros((13,10))
```

In addition it was necessary for us to create a matrix of the coefficients of each and every coefficient equation. There are 13 constraints and 10 is the number of channels. Every constraint was noted as a ‘less than’ constraint (changing signs where necessary) for ease of modelling.

```
#Constraint 1- The amount invested in print and TV should be no more than the amount spent on Facebook and Email.
A[0,:] = [1,1,0,0,-1,0,0,0,0,-1]
#Constraint 2- The total amount used in social media (Facebook, LinkedIn, Instagram, Snapchat, and Twitter) should be at lea
A[1,:] = [0,0,2,2,-1,-1,-1,-1,-1,0]
#Constraint 3 - Total amount invested across all channels should be less than $10 million
A[2,:]=[1,1,1,1,1,1,1,1,1,1]
#Constraints 4 to 13 - For each platform, the amount invested should be no more than $3M
A[3,:] = [1,0,0,0,0,0,0,0,0,0]
A[4,:] = [0,1,0,0,0,0,0,0,0,0]
A[5,:] = [0,0,1,0,0,0,0,0,0,0]
A[6,:] = [0,0,0,1,0,0,0,0,0,0]
A[7,:] = [0,0,0,0,1,0,0,0,0,0]
A[8,:] = [0,0,0,0,0,1,0,0,0,0]
A[9,:] = [0,0,0,0,0,0,1,0,0,0]
A[10,:] = [0,0,0,0,0,0,0,1,0,0]
A[11,:] = [0,0,0,0,0,0,0,0,1,0]
A[12,:] = [0,0,0,0,0,0,0,0,0,1]
print(A)
```

```
sense = np.array(['<','<','<','<','<','<','<','<','<','<','<','<','<'])
```

Having defined the objective, the constraints and their inequalities in Q1 and Q2, we now move on to running our linear model to find the optimal allocation.

We first tell the model how many variables are present, then add in our constraints and finally add our objective to the model. Then, we run the model to receive the optimized output.

```
#Running the model to find the optimal allocation
trainMod = gp.Model()

Trainx = trainMod.addMVar(len(obj1)) # tell the model how many variables there are
# must define the variables before adding constraints because variables go into the constraints
TrainModCon = trainMod.addMConstrs(A, Trainx, sense, b) # add the constraints to the model
trainMod.setMObjective(None,obj1,0,sense=gp.GRB.MAXIMIZE) # add the objective to the model
trainMod.Params.OutputFlag = 0 # tell gurobi to shut up!!
trainMod.optimize()
```

Upon running the model, we find that our profit is about $456,000.

```
#Finding the profit from our model given the constraints and other conditions
trainMod.objVal
```

```
0.45600000000000007
```

And our optimal allocation is as follows:

```
The optimal revenue we can gain is 0.46 million

Print        0.0
TV           3.0
SEO          0.0
AdWords      1.0
Facebook     0.0
LinkedIn     0.0
Instagram    3.0
Snapchat     0.0
Twitter      0.0
Email        3.0
dtype: float64
```

For maximizing our returns based on the current ROI’s we should spend $3 million on TV, $1 million on Adwords, $3 million on Instagram and $3 million on Email. Our initial allocations agree with the hypothesis that email is a great channel for reaching people, since we have allocated the maximum allowable individual spend to it too.

Question 4:

A second consulting firm reran our estimates and obtained new ROI values. Therefore, we recreated the objective vector as seen below by using the ROI values they provided (shown below):

Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
ROI	4.9%	2.3%	2.4%	3.9%	4.4%	4.6%	2.6%	1.9%	3.7%	2.6%

Using the same constraints and methodology as Question 3, we ran our model once again and obtained the following results:

```
#Running the model to find the optimal allocation
trainMod2 = gp.Model()

Trainx2 = trainMod2.addMVar(len(obj2)) # tell the model how many variables there are
# must define the variables before adding constraints because variables go into the constraints
TrainMod2Con = trainMod2.addMConstrs(A, Trainx2, sense, b) # add the constraints to the model
trainMod2.setMObjective(None,obj2,0,sense=gp.GRB.MAXIMIZE) # add the objective to the model
trainMod2.Params.OutputFlag = 0 # tell gurobi to shut up!!
trainMod2.optimize()
```

Our total profit remains unchanged at $456,000 despite a change in ROI's.

```
#Finding the profit from our model given the constraints and other conditions
trainMod2.objVal
```

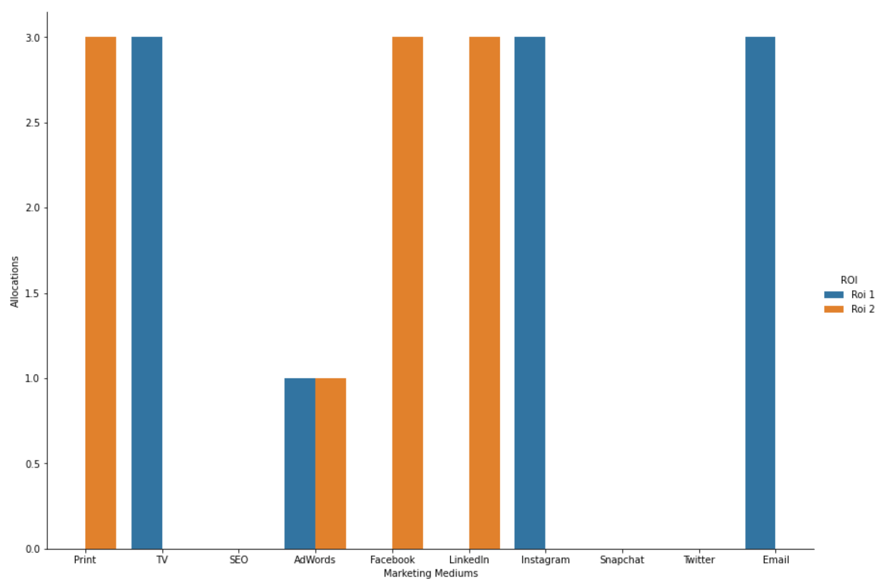
```
0.45600000000000007
```

However, our allocation changed due to the revised ROI's and is now as follows:

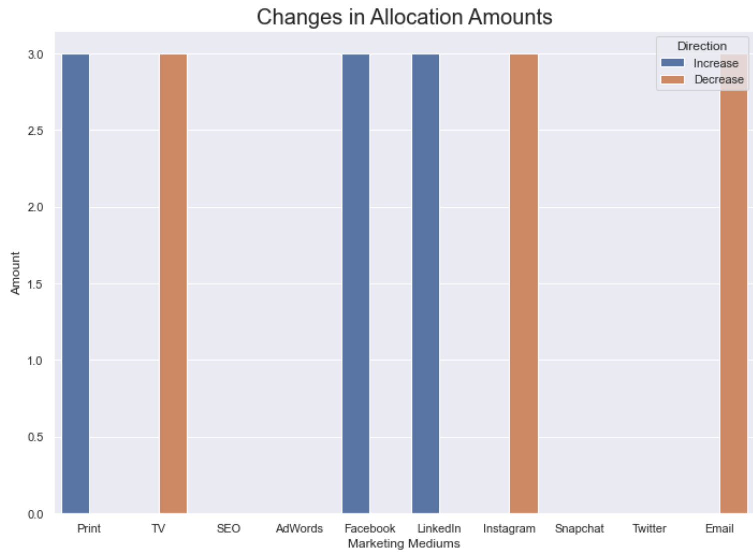
```
The optimal revenue we can gain is 0.46 million

Print        3.0
TV           0.0
SEO          0.0
AdWords      1.0
Facebook     3.0
LinkedIn     3.0
Instagram    0.0
Snapchat     0.0
Twitter      0.0
Email        0.0
dtype: float64
```

For maximizing our returns based on the current ROI's we should spend $3 million on Print, $1 million on Adwords, $3 million on Facebook and $3 million on LinkedIn. Our initial allocations agree with the hypothesis that email is a great channel for reaching people, since we have allocated the maximum allowable individual spend to it too.

A side-by-side comparison of the change in allocations are shown in the graph below,



Visualizing the individual increase/decrease by channel :



<u>Question 5</u>

For Question 5, we first decided to find how much lower our objective would be if we used the first ROI data on the second allocation. This was done by subtracting our objective value from the sum of our objectives applied to the set of second allocations. We found that our objective decreased by $0.204 million in this case.

```
#Q5
#Finding difference between optimal objective and objective used on second allocation
trainMod.objVal - obj1 @ Trainx1.x
```

```
0.20400000000000007
```

We then decided to find how much lower our objective would be if we used the second ROI data on our first allocation. This was done by subtracting our objective value from the sum of our objectives applied to the set of first allocations. We found that our objective decreased by $0.192 million in this case.

```
In [22]: #Finding difference between second allocation objective and objective used on original allocation
         trainMod2.objVal - obj2 @ Trainx.x

Out[22]: 0.19200000000000006
```

To check the relevance of constraint 3, we ran our model by removing constraint 3 and then running it with the first set of ROI’s. This led to the $10 million being divided equally amongst TV and Email channels.

```
In [24]:
print('The optimal revenue we can gain is {:.2f} million'.format(trainMod.objVal))

budget_roi2 = list(Trainx3.x)
budget_dict = {}
for chn, num in zip(channel, budget_roi2):
    budget_dict[chn] = num

pd.Series(budget_dict)
```

```
The optimal revenue we can gain is 0.46 million

Out[24]: Print        0.0
TV           5.0
SEO          0.0
AdWords      0.0
Facebook     0.0
LinkedIn     0.0
Instagram    0.0
Snapchat     0.0
Twitter      0.0
Email        5.0
dtype: float64
```

Subsequently, we ran our model by removing constraint 3 and then running it with the second set of ROI's. This led to the $10 million being divided equally amongst Print and Facebook channels.

```
In [26]:
print('The optimal revenue we can gain is {:.2f} million'.format(trainMod2.objVal))

budget_roi2 = list(Trainx4.x)
budget_dict = {}
for chn, num in zip(channel, budget_roi2):
    budget_dict[chn] = num

pd.Series(budget_dict)
```

```
The optimal revenue we can gain is 0.46 million

Out[26]: Print        5.0
TV           0.0
SEO          0.0
AdWords      0.0
Facebook     5.0
LinkedIn     0.0
Instagram    0.0
Snapchat     0.0
Twitter      0.0
Email        0.0
dtype: float64
```

Based on the allocations observed in both cases above, we concluded that the Third Constraint is required since it encourages portfolio diversity. Otherwise, the model simply chooses the channels with the highest ROI's and splits the money amongst them, exposing us to more risks in case of a change in advertising prices or other market conditions.

Question 6

In this question we were asked to essentially find the tipping point for a change in ROI using the initially provided ROI's and allocations.

To do so, we performed a sensitivity analysis of our allocation and ROI's to find the range of values before a change in allocation takes place.

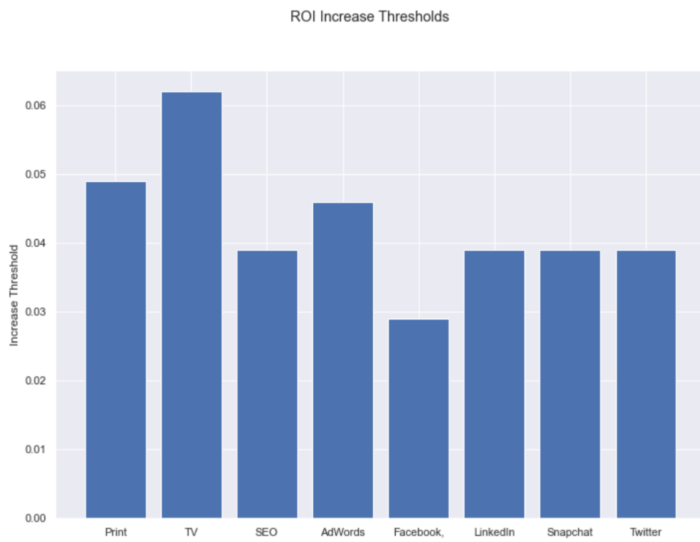
First, we ran the model and obtained the optimal solution using the same process as above:

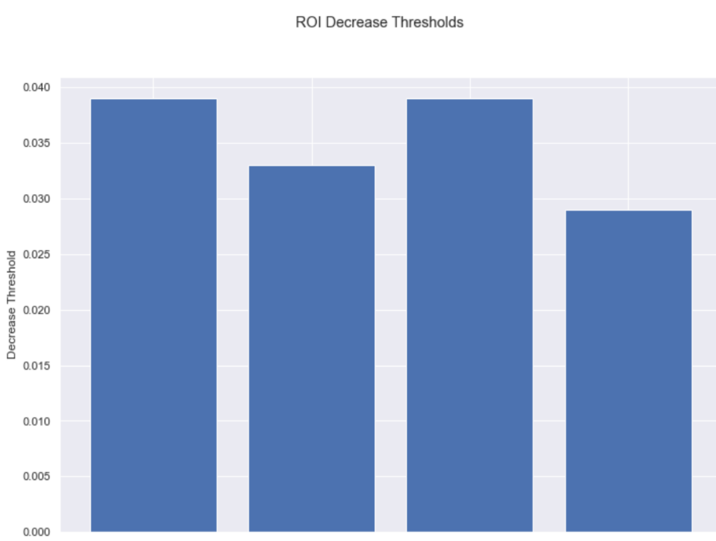
Then we found the upper and lower bounds for our channels using sensitivity analysis

```
trainMod.optimize()

print(Trainx.SAObjLow)
print(Trainx.SAObjUp)
```

	Channel	ChangeDown	ChangeUp
0	Print	-inf	0.049
1	TV	0.039	0.062
2	SEO	-inf	0.039
3	AdWords	0.033	0.046
4	Facebook	-inf	0.029
5	LinkedIn	-inf	0.039
6	Instagram	0.039	inf
7	Snapchat	-inf	0.039
8	Twitter	-inf	0.039
9	Email	0.029	inf

Also, we plotted the following graphs to visually represent both the changes (Increase/Decrease):





Question 7

This question builds upon the basic model we built earlier in the project in Questions 3,4 and 5. To find the monthly returns and allocation, we plan to run our model for every month and obtain this information.

To iterate through the months easily and in order to avoid running the model multiple times for each month, we decided to define a function that takes in the ROI's and monthly budget, runs the model given the constraints and then returns to us an array containing the profit/return for the month and the optimal allocations across channels.

Function:

```
#Q7
def optimization_function_month(roi_matrix, budget):
    obj = roi_matrix # objective vector
    A = np.zeros((13,10)) # initialize constraint matrix



    ############ Defining Constraints
    channel_level_cons = np.identity(10)
    overall_budget_cons = np.ones((1,10))
    A[11,[0,1]] = 1
    A[11,[4,9]] = -1
    A[12,4:9] = 1
    A[12,2:4] = -2
    A[0:10,] = channel_level_cons
    A[10] = overall_budget_cons
    ##################################



    #### Output Part of the constraint####
    b = np.array([3,3,3,3,3,3,3,3,3,3,budget,0,0])
    sense = np.array(['<','<','<','<','<','<','<','<','<','<','<','<','>'])



    ojModel = gp.Model() # initialize an empty model



    ojModX = ojModel.addMVar(10) # tell the model how many variables there are
    # must define the variables before adding constraints because variables go into the constraints
    ojModCon = ojModel.addMConstrs(A, ojModX, sense, b) # add the constraints to the model
    ojModel.setMObjective(None,obj,0,sense=gp.GRB.MAXIMIZE) # add the objective to the model...we'll talk about the None and



    ojModel.Params.OutputFlag = 0 # tell gurobi to shut up!!



    ojModel.optimize() # solve the LP

    objective_value = ojModel.objVal
    budget_allocation = ojModX.x

    return objective_value, budget_allocation
```

We then loaded in the matrix containing the monthly ROI's across channels and processed it through the following steps:

1. Dropped the month name column
2. Set a index column containing numbers 0 through 11 for every month and its respective ROI's
3. Created a list of the columns from the dataframe to use for locating relevant values later

```
monthly_roi = pd.read_csv("roi_mat.csv")
monthly_roi_clean = monthly_roi.drop(monthly_roi.columns[0], axis = 1)
monthly_roi_clean.head()
```

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
0	4.0	3.6	2.4	3.9	3.0	3.5	3.6	2.25	3.5	3.5
1	4.0	3.9	2.7	3.8	4.3	3.2	2.7	1.80	3.7	3.5
2	3.5	2.9	3.1	3.8	2.4	4.1	3.7	2.60	4.2	2.5
3	3.8	3.1	2.4	4.4	2.4	3.8	3.7	2.50	3.6	2.9
4	3.5	3.2	1.9	3.4	2.7	2.7	3.9	2.20	4.5	3.9

```
]:  promotions_list= list(monthly_roi_clean.columns)
```

Subsequently, we defined our initial budget as $10 million and then ran our function while iterating across a list containing all the months and obtained the following key parameters for each month:

1. Monthly ROI
2. Budget Allocation
3. Change in Budget Allocation (Delta)
4. Profit on Actual ROI (From Matrix)

A sample of monthly outputs for January, April, July, October and December are shown below:

```
Month: Jan

Total Budget = 10.00
```

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
Monthly ROI	0.04	0.036	0.024	0.039	0.03	0.035	0.036	0.0225	0.035	0.035
Budget Allocation	0.0	3.0	0.0	1.0	0.0	0.0	3.0	0.0	0.0	3.0
Delta Budget Allocation	0.0	3.0	0.0	1.0	0.0	0.0	3.0	0.0	0.0	3.0
Profit on Actual ROI	0.0	0.108	0.0	0.039	0.0	0.0	0.108	0.0	0.0	0.105

```
Month: Apr

Total Budget = 10.52
```

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
Monthly ROI	0.038	0.031	0.024	0.044	0.024	0.038	0.037	0.025	0.036	0.029
Budget Allocation	0.0	3.0	0.0	1.51	0.0	0.0	3.0	0.0	0.01	3.0
Delta Budget Allocation	0.0	0.0	0.0	0.15	0.0	0.0	0.0	0.0	0.01	0.0
Profit on Actual ROI	0.0	0.093	0.0	0.06644	0.0	0.0	0.111	0.0	0.00036	0.087

```
Month: Jul

Total Budget = 11.10
```

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
Monthly ROI	0.039	0.036	0.02	0.044	0.039	0.037	0.043	0.018	0.04	0.038
Budget Allocation	0.0	3.0	0.0	1.7	0.0	0.0	3.0	0.0	0.4	3.0
Delta Budget Allocation	0.0	0.0	0.0	0.07	0.0	0.0	0.0	0.0	0.14	0.0
Profit on Actual ROI	0.0	0.108	0.0	0.0748	0.0	0.0	0.129	0.0	0.016	0.114

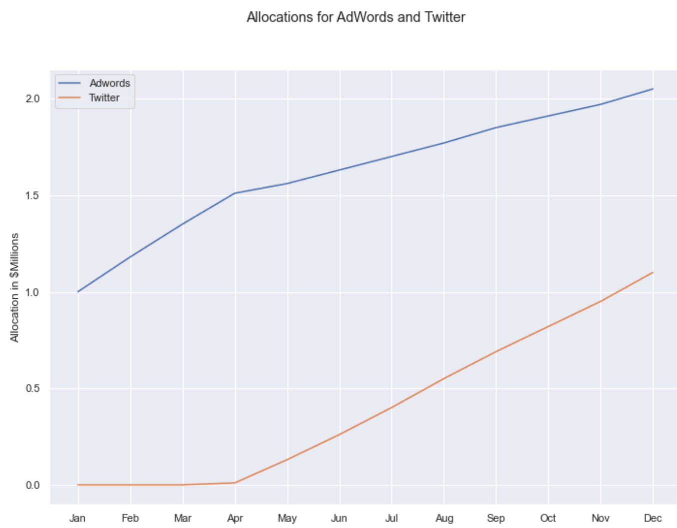
```
Month: Oct

Total Budget = 11.72
```

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
Monthly ROI	0.03	0.03	0.031	0.046	0.031	0.033	0.032	0.023	0.025	0.032
Budget Allocation	0.0	3.0	0.0	1.91	0.0	0.0	3.0	0.0	0.82	3.0
Delta Budget Allocation	0.0	0.0	0.0	0.06	0.0	0.0	0.0	0.0	0.13	0.0
Profit on Actual ROI	0.0	0.09	0.0	0.08786	0.0	0.0	0.096	0.0	0.0205	0.096

```
Month: Dec

Total Budget = 12.15
```

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
Monthly ROI	0.048	0.04	0.019	0.037	0.042	0.036	0.026	0.029	0.036	0.037
Budget Allocation	0.0	3.0	0.0	2.05	0.0	0.0	3.0	0.0	1.1	3.0
Delta Budget Allocation	0.0	0.0	0.0	0.08	0.0	0.0	0.0	0.0	0.15	0.0
Profit on Actual ROI	0.0	0.12	0.0	0.07585	0.0	0.0	0.078	0.0	0.0396	0.111

Finally, we plotted the trends for change in allocations across the two key channels that had changing allocations throughout the year, AdWords and Twitter:



Question 8

As per the information provided, a stable budget is defined as one where monthly allocation such that for each platform the monthly change in spend is no more than $1M. Our budget appears to be extremely robust given this definition, because on reviewing the profits and monthly optimal allocations, we notice the following:

1. The only budgets that are changing in our model from question 7 are those for AdWords and Twitter, and neither of them ever increase by more than $230,000 on a monthly basis.
2. Building on the logic in 1, our monthly changes in allocations are also always less than $1 million.

Hence, we can safely say that our budget estimates are stable since there is no mathematical or logical manner in which we can exceed $1 million in allocation changes.

If the budget was unstable, we would need to add more constraints with respect to allocations and spending limits on specific channels in order to make it stable.