In [15]:
```
import warnings
warnings.filterwarnings('ignore')
from operator import itemgetter
import pandas as pd #dataframe
import numpy as np #mathematical computations
import matplotlib.pyplot as plt #visualization
import matplotlib
import seaborn as sns #visualization
import json #exporting columns
import pickle #saving the model
from sklearn.linear_model import LinearRegression #Linear Regression
from sklearn.linear_model import Lasso #Lasso Regression
from sklearn.tree import DecisionTreeRegressor #Decision Tree Regression
from sklearn.ensemble import RandomForestRegressor #Random Forest Regression
from sklearn.model_selection import train_test_split #Splitting the dataset
from sklearn.model_selection import ShuffleSplit #Random shuffling
from sklearn.model_selection import cross_val_score #Score cross validation
from sklearn.model_selection import GridSearchCV #Hyper parameter tuning
from warnings import simplefilter #Filtering warnings
import seaborn as sns
import missingno as msno
import statsmodels.api as sm
from datetime import datetime
```

Observe the data

Import the data set and show the title

In [16]:
```
Orginal_data = pd.read_csv('./Combined.csv',encoding = "ISO-8859-1")
```

In [17]:
```
print(Orginal_data.columns)
```

```
Index(['Campaign_ID', 'Campagin_Title ', 'Receiving_NPO_name ',
       'Receiving_NPO_Id', 'NPO_Status_orignal', 'NPO_Status',
       'Number_campaigns_NPO', 'Public_Campaign_Access', 'Creator_Type',
       'Creator_Id', 'Campaign_Status', 'Actual_Donation_Amount',
       'Distinct_Donors', 'Campaign_Goal', 'Campaign_Completion_Rate',
       'Days_Left_for_Campaign', 'Campaign_Start_Date', 'Campaign_End_Dat
e',
       'NPO_Tax_Deductibility', 'Campaign_Image1', 'Campaign_Image2',
       'Campaign_Image3', 'Campaign_Image4', 'Campaign_Image5',
       'Campaign_Video', 'Impact_Message1', 'Impact_Message2',
       'Impact_Message3', 'Impact_Message4', 'Impact_Message5',
       'Custom_Amount1', 'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount
4',
       'Description_Campaign', 'Description_NPO'],
      dtype='object')
```

I found there is no "Organizational Causes" and "Campaign Causes" in this data set.

Here are all variables I plan to operate, ignnore other columns temporarily

In [18]:
```
Independ_variable = ["Actual_Donation_Amount","NPO_Tax_Deductibility", "Dist
"Campaign_Image4","Campaign_Image5","Campaign_Video","Impact_Message1","Impa
extract_data = Orginal_data[Independ_variable]
extract_data
```

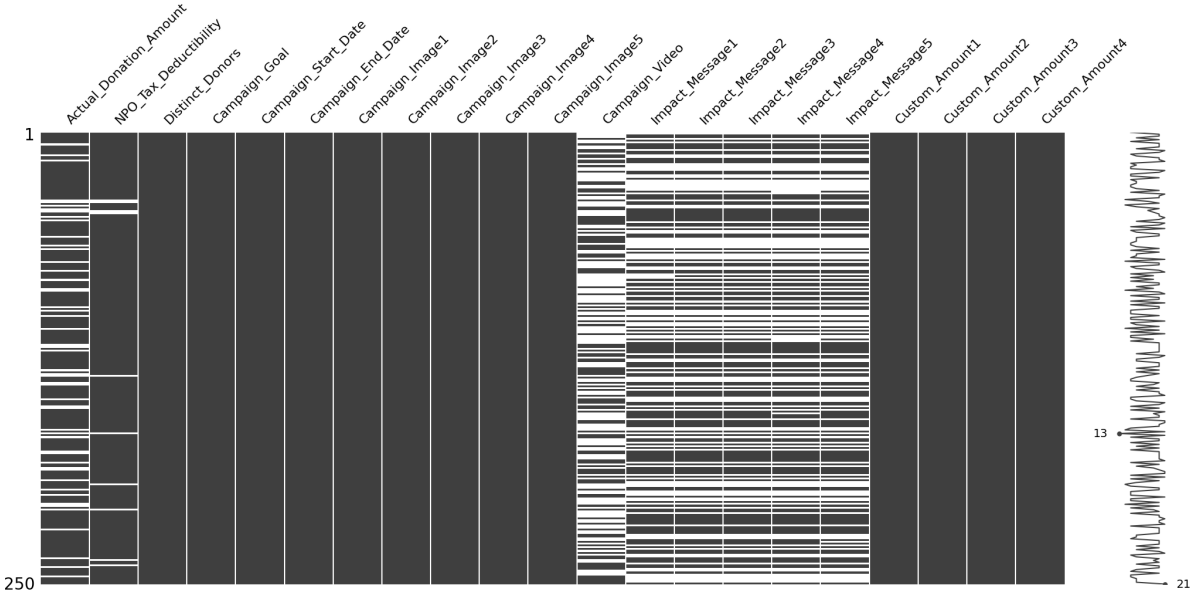
Out[18]:

	Actual_Donation_Amount	NPO_Tax_Deductibility	Distinct_Donors	Campaign_Goal	C
0	5561.0	True	66	50000	
1	2810.0	True	32	20000	
2	1118.0	True	22	30000	
3	2800.0	True	7	2000	
4	2030.0	True	27	5000	
...	...	...	...	...	
15972	10.0	True	1	5000	
15973	150.0	True	4	10000	
15974	1000.0	True	10	1000	
15975	120.0	True	2	3000	
15976	120.0	True	2	40000	

15977 rows × 21 columns

In [19]:
```
msno.matrix(extract_data.sample(250))
```

Out[19]: <AxesSubplot:>



We can see that "Actual_Donation_Amount" "Campaign_Video" "Impact_Message1" "Impact_Message2" "Impact_Message3" "Impact_Message4" and "Impact_Message5" are many missing data, fill them first so that it's more convenient to operate. "NPO_Tax_Deductibility" has been ignore temporarily just like you said in email

In [20]:
```
extract_data['NPO_Tax_Deductibility'] = extract_data['NPO_Tax_Deductibility'
extract_data['Actual_Donation_Amount'] = extract_data['Actual_Donation_Amoun
extract_data['Actual_Donation_Amount'] = pd.to_numeric( extract_data['Actual
extract_data['Distinct_Donors'] = extract_data['Distinct_Donors'].fillna('0'
extract_data['Distinct_Donors'] = pd.to_numeric( extract_data['Distinct_Donc
extract_data['Campaign_Video'] = extract_data['Campaign_Video'].fillna('0')
extract_data['Impact_Message1'] = extract_data['Impact_Message1'].fillna('0'
extract_data['Impact_Message2'] = extract_data['Impact_Message2'].fillna('0'
extract_data['Impact_Message3'] = extract_data['Impact_Message3'].fillna('0'
extract_data['Impact_Message4'] = extract_data['Impact_Message4'].fillna('0'
extract_data['Impact_Message5'] = extract_data['Impact_Message5'].fillna('0'
```

In [21]:
```
extract_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15977 entries, 0 to 15976
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Actual_Donation_Amount  15977 non-null  float64
 1   NPO_Tax_Deductibility   15977 non-null  object 
 2   Distinct_Donors         15977 non-null  int64  
 3   Campaign_Goal           15977 non-null  int64  
 4   Campaign_Start_Date     15977 non-null  object 
 5   Campaign_End_Date       15977 non-null  object 
 6   Campaign_Image1         15977 non-null  int64  
 7   Campaign_Image2         15977 non-null  int64  
 8   Campaign_Image3         15977 non-null  int64  
 9   Campaign_Image4         15977 non-null  int64  
 10  Campaign_Image5         15977 non-null  int64  
 11  Campaign_Video          15977 non-null  object 
 12  Impact_Message1         15977 non-null  object 
 13  Impact_Message2         15977 non-null  object 
 14  Impact_Message3         15977 non-null  object 
 15  Impact_Message4         15977 non-null  object 
 16  Impact_Message5         15977 non-null  object 
 17  Custom_Amount1          15977 non-null  int64  
 18  Custom_Amount2          15977 non-null  int64  
 19  Custom_Amount3          15977 non-null  int64  
 20  Custom_Amount4          15977 non-null  int64  
dtypes: float64(1), int64(11), object(9)
memory usage: 2.6+ MB
```

There is no donations per donor, So add a columns of donations per donor

In [22]:
```
extract_data.columns
```

Out[22]:
```
Index(['Actual_Donation_Amount', 'NPO_Tax_Deductibility', 'Distinct_Donor
s',
       'Campaign_Goal', 'Campaign_Start_Date', 'Campaign_End_Date',
       'Campaign_Image1', 'Campaign_Image2', 'Campaign_Image3',
       'Campaign_Image4', 'Campaign_Image5', 'Campaign_Video',
       'Impact_Message1', 'Impact_Message2', 'Impact_Message3',
       'Impact_Message4', 'Impact_Message5', 'Custom_Amount1',
       'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount4'],
      dtype='object')
```

In [23]:
```
extract_data['NPO_Tax_Deductibility'][0:10]
```

Out[23]:
```
0     True
1     True
2     True
3     True
4     True
5     True
6     True
7    False
8     True
9     True
Name: NPO_Tax_Deductibility, dtype: object
```

In [24]:
```
# I am not sure Distinct_Donors is the total donors or not ?
```

```
extract_data['Donation_per_donor'] = 0
for j in range(len(extract_data["Actual_Donation_Amount"])):
    if extract_data["Distinct_Donors"].iloc[j] != 0:
        extract_data['Donation_per_donor'].iloc[j] = extract_data['Actual_Do
    else:
        extract_data['Donation_per_donor'].iloc[j] = 0

    if extract_data['NPO_Tax_Deductibility'].iloc[j] == True:
        extract_data.loc[j, 'NPO_Tax_Deductibility'] = 1
    else:
        extract_data.loc[j, 'NPO_Tax_Deductibility'] = 0
```

In [25]:
```
print(extract_data['NPO_Tax_Deductibility'][0:10])
```

```
0    1
1    1
2    1
3    1
4    1
5    1
6    1
7    0
8    1
9    1
Name: NPO_Tax_Deductibility, dtype: object
```

Add a columns of numbers of images

In [26]:
```
Add_Campaign_Image_num = lambda x0,x1,x2,x3,x4: (x0 != 0).astype(np.int) +(x
extract_data["Campaign_Image_num"] = Add_Campaign_Image_num(extract_data["Ca
```

Classfy video into “0” and ”1“ two categories

In [27]:
```
Video_or_not = lambda x0: (x0 != '0').astype(np.int)
extract_data["Campaign_Video"] = Video_or_not(extract_data["Campaign_Video"]
extract_data
```

Out[27]:

	Actual_Donation_Amount	NPO_Tax_Deductibility	Distinct_Donors	Campaign_Goal	C
0	5561.0	1	66	50000	
1	2810.0	1	32	20000	
2	1118.0	1	22	30000	
3	2800.0	1	7	2000	
4	2030.0	1	27	5000	
...	...	...	...	...	
15972	10.0	1	1	5000	
15973	150.0	1	4	10000	
15974	1000.0	1	10	1000	
15975	120.0	1	2	3000	
15976	120.0	1	2	40000	

15977 rows × 23 columns

In [ ]:

The format of the date needs to be modified and the duration will be calculated below

In [28]:
```
month_dictionary = {'Jan':'1',
 'Feb':'2',
 'Mar':'3',
 'Apr':'4',
 'May':'5',
 'Jun':'6',
 'Jul':'7',
 'Aug':'8',
 'Sep':'9',
 'Oct':'10',
 'Nov':'11',
 'Dec':'12'}
extract_data['Campaign_Start_Day'] = '0'
extract_data['Campaign_Start_Month'] = '0'
extract_data['Campaign_Start_Year'] = '0'
extract_data['Campaign_End_Day'] = '0'
extract_data['Campaign_End_Month'] = '0'
extract_data['Campaign_End_Year'] = '0'
extract_data['Campaign_Start'] = '0'
extract_data['Campaign_End'] = '0'
extract_data['duration_day'] = '0'
```

```
i = 0
for row in extract_data['Campaign_Start_Date']:
    extract_data.loc[i, 'Campaign_Start_Day'] = extract_data['Campaign_Start
    extract_data.loc[i, 'Campaign_Start_Month'] =month_dictionary[ extract_d
    extract_data.loc[i, 'Campaign_Start_Year'] = '20'+ extract_data['Campaig
    extract_data.loc[i, 'Campaign_End_Day'] = extract_data['Campaign_End_Dat
    extract_data.loc[i, 'Campaign_End_Month'] = month_dictionary[extract_dat
    extract_data.loc[i, 'Campaign_End_Year'] = '20' + extract_data['Campaign
    extract_data.loc[i, 'Campaign_Start'] = extract_data['Campaign_Start_Yea
    extract_data.loc[i, 'Campaign_End'] = extract_data['Campaign_End_Year'].
    extract_data.loc[i, 'duration_day'] = (datetime.strptime(extract_data['C
    if extract_data.loc[i, 'duration_day'] < 0:
        extract_data.loc[i, 'duration_day'] = 0
    i += 1

extract_data.iloc[:,20:]
```

Out[28]:

	Custom_Amount4	Donation_per_donor	Campaign_Image_num	Campaign_Start_Day
0	200	84.257576	5	1
1	200	87.812500	2	1
2	200	50.818182	2	1
3	0	400.000000	5	2
4	0	75.185185	3	1
...	...	...	...	...
15972	0	10.000000	4	31
15973	0	37.500000	3	31
15974	0	100.000000	3	31
15975	0	60.000000	2	31
15976	200	60.000000	3	5

15977 rows × 12 columns

In [29]:
```
extract_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15977 entries, 0 to 15976
Data columns (total 32 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Actual_Donation_Amount  15977 non-null  float64
 1   NPO_Tax_Deductibility   15977 non-null  object 
 2   Distinct_Donors         15977 non-null  int64  
 3   Campaign_Goal           15977 non-null  int64  
 4   Campaign_Start_Date     15977 non-null  object 
 5   Campaign_End_Date       15977 non-null  object 
 6   Campaign_Image1         15977 non-null  int64  
 7   Campaign_Image2         15977 non-null  int64  
 8   Campaign_Image3         15977 non-null  int64  
 9   Campaign_Image4         15977 non-null  int64  
 10  Campaign_Image5         15977 non-null  int64  
 11  Campaign_Video          15977 non-null  int64  
 12  Impact_Message1         15977 non-null  object 
 13  Impact_Message2         15977 non-null  object 
 14  Impact_Message3         15977 non-null  object 
 15  Impact_Message4         15977 non-null  object 
 16  Impact_Message5         15977 non-null  object 
 17  Custom_Amount1          15977 non-null  int64  
 18  Custom_Amount2          15977 non-null  int64  
 19  Custom_Amount3          15977 non-null  int64  
 20  Custom_Amount4          15977 non-null  int64  
 21  Donation_per_donor      15977 non-null  float64
 22  Campaign_Image_num      15977 non-null  int64  
 23  Campaign_Start_Day      15977 non-null  object 
 24  Campaign_Start_Month    15977 non-null  object 
 25  Campaign_Start_Year     15977 non-null  object 
 26  Campaign_End_Day        15977 non-null  object 
 27  Campaign_End_Month      15977 non-null  object 
 28  Campaign_End_Year       15977 non-null  object 
 29  Campaign_Start          15977 non-null  object 
 30  Campaign_End            15977 non-null  object 
 31  duration_day            15977 non-null  object 
dtypes: float64(2), int64(13), object(17)
memory usage: 3.9+ MB
```

See more infomation about every columns

Check whether there are missing data

In [30]:
```
extract_data.isnull().sum()
```

Out[30]:
```
Actual_Donation_Amount    0
NPO_Tax_Deductibility     0
Distinct_Donors           0
Campaign_Goal             0
Campaign_Start_Date       0
Campaign_End_Date         0
Campaign_Image1           0
Campaign_Image2           0
Campaign_Image3           0
Campaign_Image4           0
Campaign_Image5           0
Campaign_Video            0
Impact_Message1           0
Impact_Message2           0
Impact_Message3           0
Impact_Message4           0
Impact_Message5           0
Custom_Amount1            0
Custom_Amount2            0
Custom_Amount3            0
Custom_Amount4            0
Donation_per_donor        0
Campaign_Image_num        0
Campaign_Start_Day        0
Campaign_Start_Month      0
Campaign_Start_Year       0
Campaign_End_Day          0
Campaign_End_Month        0
Campaign_End_Year         0
Campaign_Start            0
Campaign_End              0
duration_day              0
dtype: int64
```

Sentiment Analysis

In [31]:
```
comm_data = pd.DataFrame()
extract_data['Msg1_polarity'] = 0
extract_data['Msg1_subjectivity'] = 0
extract_data['Msg2_polarity'] = 0
extract_data['Msg2_subjectivity'] = 0
extract_data['Msg3_polarity'] = 0
extract_data['Msg3_subjectivity'] = 0
extract_data['Msg4_polarity'] = 0
extract_data['Msg4_subjectivity'] = 0
extract_data['Msg5_polarity'] = 0
extract_data['Msg5_subjectivity'] = 0
```

In [32]:
```
extract_data.columns
```

Out[32]:
```
Index(['Actual_Donation_Amount', 'NPO_Tax_Deductibility', 'Distinct_Donors',
       'Campaign_Goal', 'Campaign_Start_Date', 'Campaign_End_Date',
       'Campaign_Image1', 'Campaign_Image2', 'Campaign_Image3',
       'Campaign_Image4', 'Campaign_Image5', 'Campaign_Video',
       'Impact_Message1', 'Impact_Message2', 'Impact_Message3',
       'Impact_Message4', 'Impact_Message5', 'Custom_Amount1',
       'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount4',
       'Donation_per_donor', 'Campaign_Image_num', 'Campaign_Start_Day',
       'Campaign_Start_Month', 'Campaign_Start_Year', 'Campaign_End_Day',
       'Campaign_End_Month', 'Campaign_End_Year', 'Campaign_Start',
       'Campaign_End', 'duration_day', 'Msg1_polarity', 'Msg1_subjectivity',
       'Msg2_polarity', 'Msg2_subjectivity', 'Msg3_polarity',
       'Msg3_subjectivity', 'Msg4_polarity', 'Msg4_subjectivity',
       'Msg5_polarity', 'Msg5_subjectivity'],
      dtype='object')
```

The polarity item is the positiveness of the text, which is a floating point number in the range of [-1.0, 1.0] The subjectivity item is a subjective score, which is a floating point number in the range of [0.0, 1.0], where 0.0 is very objective and 1.0 is very subjective

In [33]:
```
from textblob import TextBlob
# polarity项为文本积极性，是在[-1.0，1.0]范围内的浮点数
# subjectivity项为主观评分，是在[0.0，1.0]范围内的浮点数，其中0.0是非常客观的，而1.0是
Impact_msg_list = ['Impact_Message1','Impact_Message2','Impact_Message3','Im
Msg_polarity_list = ['Msg1_polarity','Msg2_polarity','Msg3_polarity','Msg4_p
Msg1_subjectivity_list = ['Msg1_subjectivity','Msg2_subjectivity','Msg3_subj
for j in range(len(Impact_msg_list)):
    t=0
    for i in extract_data[Impact_msg_list[j]]:
        blob = TextBlob(i)
        sentiment = blob.sentiment
        extract_data[Msg_polarity_list[j]].iloc[t] = sentiment.polarity
        extract_data[Msg1_subjectivity_list[j]].iloc[t] = sentiment.subjecti
        t+=1
# sum the total five messages polarity and subjectivity
extract_data["Total_Msg_polarity"] = extract_data["Msg1_polarity"]+extract_d
extract_data["Total_Msg_subjectivity"] = extract_data["Msg1_subjectivity"]+e
extract_data.iloc[0:30,28:]
```

Out[33]:

	Campaign_End_Year	Campaign_Start	Campaign_End	duration_day	Msg1_polarity	Msg
0	2017	2017-1-1	2017-9-10	252	0.000000	
1	2017	2017-1-1	2017-3-31	89	0.000000	
2	2017	2017-1-1	2017-2-28	58	0.000000	
3	2017	2017-1-2	2017-3-31	88	0.000000	
4	2017	2017-1-1	2017-2-20	50	0.000000	
5	2018	2017-1-1	2018-1-1	365	-0.200000	
6	2017	2017-1-3	2017-2-7	35	0.000000	
7	2017	2017-1-3	2017-12-31	362	0.000000	
8	2017	2017-1-3	2017-2-14	42	0.000000	
9	2018	2017-1-3	2018-1-3	365	0.000000	
10	2017	2017-1-9	2017-5-31	142	-0.055952	
11	2017	2017-1-4	2017-1-31	27	0.000000	
12	2017	2017-1-1	2017-3-17	75	0.000000	
13	2017	2017-1-5	2017-3-31	85	0.450000	
14	2017	2017-1-14	2017-2-28	45	0.000000	
15	2017	2017-1-5	2017-1-21	16	0.500000	
16	2018	2017-1-5	2018-1-5	365	0.000000	
17	2017	2017-1-5	2017-3-4	58	0.000000	
18	2017	2017-1-6	2017-12-31	359	0.000000	
19	2017	2017-1-6	2017-3-31	84	0.000000	
20	2017	2017-1-6	2017-3-31	84	0.000000	
21	2018	2017-1-8	2018-1-8	365	0.000000	
22	2017	2017-1-8	2017-10-31	296	0.000000	
23	2017	2017-1-8	2017-2-17	40	0.000000	
24	2017	2017-1-9	2017-2-12	34	0.000000	
25	2017	2017-1-11	2017-2-13	33	0.000000	
26	2017	2017-1-10	2017-7-31	202	0.000000	
27	2017	2017-1-12	2017-1-31	19	0.000000	
28	2017	2017-1-16	2017-6-30	165	0.000000	
29	2017	2017-1-18	2017-3-19	60	0.000000	

The method of judging the similarity uses the difflib library It is a score, which in range of [0.0, 1.0]. 0 means this two sentences are totally different and 1 means there are the same.

In [34]:
```
import difflib
def get_equal_rate_1(str1, str2):
   return difflib.SequenceMatcher(None, str1, str2).quick_ratio()
extract_data['Total_similarity'] = 0
Impact_msg_list = ['Impact_Message1','Impact_Message2','Impact_Message3','Im
for j in range(len(Impact_msg_list)-1):
    for i in range(extract_data[Impact_msg_list[j]].shape[0]):
        str1 = extract_data[Impact_msg_list[j]].iloc[i]
        str2 = extract_data[Impact_msg_list[j+1]].iloc[i]
        extract_data['Total_similarity'].iloc[i] += get_equal_rate_1(str1, s
extract_data.iloc[:,28:]
```

Out[34]:

	Campaign_End_Year	Campaign_Start	Campaign_End	duration_day	Msg1_polarity
0	2017	2017-1-1	2017-9-10	252	0.0000
1	2017	2017-1-1	2017-3-31	89	0.0000
2	2017	2017-1-1	2017-2-28	58	0.0000
3	2017	2017-1-2	2017-3-31	88	0.0000
4	2017	2017-1-1	2017-2-20	50	0.0000
...	...	...	...	...	...
15972	2023	2022-10-31	2023-1-1	62	0.0000
15973	2022	2022-10-31	2022-11-30	30	0.0000
15974	2022	2022-10-31	2022-11-30	30	0.0000
15975	2022	2022-10-31	2022-12-31	61	0.0000
15976	2022	2022-9-5	2022-12-31	117	-0.1875

15977 rows × 17 columns

Between two strings, the minimum number of editing operations required to convert one into another, if the distance between them is greater, it means that they are more different

In [35]:
```
import distance
extract_data['Total_distance'] = 0
def edit_distance(s1, s2):
    return distance.levenshtein(s1, s2)

for j in range(len(Impact_msg_list)-1):
    for i in range(extract_data[Impact_msg_list[j]].shape[0]):
        str1 = extract_data[Impact_msg_list[j]].iloc[i]
        str2 = extract_data[Impact_msg_list[j+1]].iloc[i]
        extract_data['Total_distance'].iloc[i] += edit_distance(str1, str2)

extract_data.iloc[0:30,28:]
```

Out[35]:

	Campaign_End_Year	Campaign_Start	Campaign_End	duration_day	Msg1_polarity	Msg
0	2017	2017-1-1	2017-9-10	252	0.000000	
1	2017	2017-1-1	2017-3-31	89	0.000000	
2	2017	2017-1-1	2017-2-28	58	0.000000	
3	2017	2017-1-2	2017-3-31	88	0.000000	
4	2017	2017-1-1	2017-2-20	50	0.000000	
5	2018	2017-1-1	2018-1-1	365	-0.200000	
6	2017	2017-1-3	2017-2-7	35	0.000000	
7	2017	2017-1-3	2017-12-31	362	0.000000	
8	2017	2017-1-3	2017-2-14	42	0.000000	
9	2018	2017-1-3	2018-1-3	365	0.000000	
10	2017	2017-1-9	2017-5-31	142	-0.055952	
11	2017	2017-1-4	2017-1-31	27	0.000000	
12	2017	2017-1-1	2017-3-17	75	0.000000	
13	2017	2017-1-5	2017-3-31	85	0.450000	
14	2017	2017-1-14	2017-2-28	45	0.000000	
15	2017	2017-1-5	2017-1-21	16	0.500000	
16	2018	2017-1-5	2018-1-5	365	0.000000	
17	2017	2017-1-5	2017-3-4	58	0.000000	
18	2017	2017-1-6	2017-12-31	359	0.000000	
19	2017	2017-1-6	2017-3-31	84	0.000000	
20	2017	2017-1-6	2017-3-31	84	0.000000	
21	2018	2017-1-8	2018-1-8	365	0.000000	
22	2017	2017-1-8	2017-10-31	296	0.000000	
23	2017	2017-1-8	2017-2-17	40	0.000000	
24	2017	2017-1-9	2017-2-12	34	0.000000	
25	2017	2017-1-11	2017-2-13	33	0.000000	
26	2017	2017-1-10	2017-7-31	202	0.000000	
27	2017	2017-1-12	2017-1-31	19	0.000000	
28	2017	2017-1-16	2017-6-30	165	0.000000	
29	2017	2017-1-18	2017-3-19	60	0.000000	

In [36]:
```
extract_data['Msg1_category'] = 0
extract_data['Msg2_category'] = 0
extract_data['Msg3_category'] = 0
extract_data['Msg4_category'] = 0
extract_data['Msg5_category'] = 0
Impact_msg_list = ['Impact_Message1','Impact_Message2','Impact_Message3','Im
Msg_category_list = ['Msg1_category','Msg2_category','Msg3_category','Msg4_c
def sentence_length(s):
```

```
    return len([i for i in s.split(' ') if i])

for j in range(len(Impact_msg_list)):
    cnt=0
    for s in extract_data[Impact_msg_list[j]]:
        extract_data[Msg_category_list[j]].iloc[cnt] = 0 if sentence_length(
        cnt += 1
extract_data.iloc[0:30,34:]
```

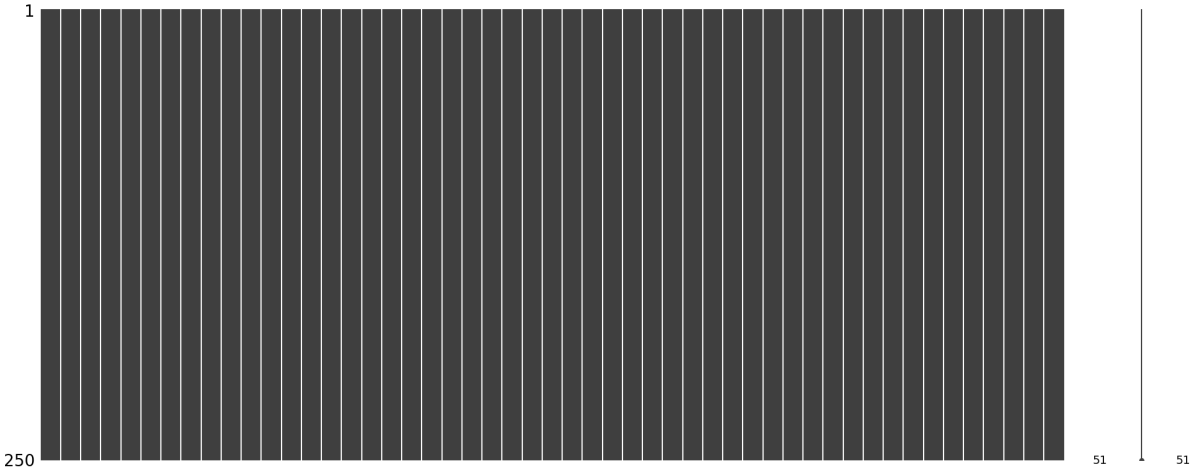
Out[36]:

	Msg2_polarity	Msg2_subjectivity	Msg3_polarity	Msg3_subjectivity	Msg4_polarity	Ms
0	0.350000	0.550000	0.000000	0.000000	0.000000	
1	0.000000	0.000000	0.200000	0.241667	0.000000	
2	0.000000	0.000000	0.200000	0.241667	0.000000	
3	0.000000	0.000000	0.000000	0.000000	0.000000	
4	0.000000	0.000000	0.000000	0.000000	0.000000	
5	-0.200000	0.300000	-0.200000	0.300000	-0.200000	
6	0.000000	0.000000	0.000000	0.000000	0.000000	
7	0.000000	0.000000	0.000000	0.000000	0.000000	
8	0.000000	0.000000	0.000000	0.000000	0.000000	
9	0.000000	0.000000	0.000000	0.000000	0.000000	
10	0.000000	0.000000	0.000000	0.000000	0.100000	
11	0.000000	0.000000	0.000000	0.000000	0.000000	
12	0.000000	0.000000	0.000000	0.000000	0.000000	
13	-0.025000	0.125000	-0.025000	0.125000	0.000000	
14	0.000000	0.000000	0.000000	0.000000	0.000000	
15	0.500000	0.500000	0.000000	0.100000	0.000000	
16	0.000000	0.000000	0.000000	0.000000	0.000000	
17	0.000000	0.000000	0.000000	0.000000	0.000000	
18	0.000000	0.000000	0.200000	0.400000	0.000000	
19	0.000000	0.000000	0.000000	0.000000	0.000000	
20	0.000000	0.000000	0.000000	0.000000	0.000000	
21	0.000000	0.000000	0.000000	0.000000	0.000000	
22	0.000000	0.000000	0.000000	0.000000	0.000000	
23	0.000000	0.000000	0.200000	0.400000	0.000000	
24	0.000000	0.000000	0.000000	0.000000	0.000000	
25	0.136364	0.454545	0.136364	0.454545	0.468182	
26	0.000000	0.000000	0.000000	0.000000	0.000000	
27	0.000000	0.000000	0.000000	0.000000	0.000000	
28	0.000000	0.000000	0.000000	0.000000	0.000000	
29	0.000000	0.000000	0.000000	0.000000	0.000000	

Well done of data cleaning and feature structure

In [37]:
```
msno.matrix(extract_data.sample(250))
```

Out[37]:
```
<AxesSubplot:>
```



In [38]:
```
extract_data.columns
```

Out[38]:
```
Index(['Actual_Donation_Amount', 'NPO_Tax_Deductibility', 'Distinct_Donor
s',
       'Campaign_Goal', 'Campaign_Start_Date', 'Campaign_End_Date',
       'Campaign_Image1', 'Campaign_Image2', 'Campaign_Image3',
       'Campaign_Image4', 'Campaign_Image5', 'Campaign_Video',
       'Impact_Message1', 'Impact_Message2', 'Impact_Message3',
       'Impact_Message4', 'Impact_Message5', 'Custom_Amount1',
       'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount4',
       'Donation_per_donor', 'Campaign_Image_num', 'Campaign_Start_Day',
       'Campaign_Start_Month', 'Campaign_Start_Year', 'Campaign_End_Day',
       'Campaign_End_Month', 'Campaign_End_Year', 'Campaign_Start',
       'Campaign_End', 'duration_day', 'Msg1_polarity', 'Msg1_subjectivit
y',
       'Msg2_polarity', 'Msg2_subjectivity', 'Msg3_polarity',
       'Msg3_subjectivity', 'Msg4_polarity', 'Msg4_subjectivity',
       'Msg5_polarity', 'Msg5_subjectivity', 'Total_Msg_polarity',
       'Total_Msg_subjectivity', 'Total_similarity', 'Total_distance',
       'Msg1_category', 'Msg2_category', 'Msg3_category', 'Msg4_category',
       'Msg5_category'],
      dtype='object')
```

Convert to numeric type

In [39]:
```
extract_data['Total_Msg_subjectivity'] = pd.to_numeric( extract_data['Total_
extract_data['Total_Msg_polarity'] = pd.to_numeric( extract_data['Total_Msg_
extract_data['NPO_Tax_Deductibility'] = pd.to_numeric(extract_data['NPO_Tax_
extract_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15977 entries, 0 to 15976
Data columns (total 51 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Actual_Donation_Amount  15977 non-null  float64
 1   NPO_Tax_Deductibility   15977 non-null  int64  
 2   Distinct_Donors         15977 non-null  int64  
 3   Campaign_Goal           15977 non-null  int64  
 4   Campaign_Start_Date     15977 non-null  object 
 5   Campaign_End_Date       15977 non-null  object 
 6   Campaign_Image1         15977 non-null  int64  
 7   Campaign_Image2         15977 non-null  int64  
 8   Campaign_Image3         15977 non-null  int64  
 9   Campaign_Image4         15977 non-null  int64  
 10  Campaign_Image5         15977 non-null  int64  
 11  Campaign_Video          15977 non-null  int64  
 12  Impact_Message1         15977 non-null  object 
 13  Impact_Message2         15977 non-null  object 
 14  Impact_Message3         15977 non-null  object 
 15  Impact_Message4         15977 non-null  object 
 16  Impact_Message5         15977 non-null  object 
 17  Custom_Amount1          15977 non-null  int64  
 18  Custom_Amount2          15977 non-null  int64  
 19  Custom_Amount3          15977 non-null  int64  
 20  Custom_Amount4          15977 non-null  int64  
 21  Donation_per_donor      15977 non-null  float64
 22  Campaign_Image_num      15977 non-null  int64  
 23  Campaign_Start_Day      15977 non-null  object 
 24  Campaign_Start_Month    15977 non-null  object 
 25  Campaign_Start_Year     15977 non-null  object 
 26  Campaign_End_Day        15977 non-null  object 
 27  Campaign_End_Month      15977 non-null  object 
 28  Campaign_End_Year       15977 non-null  object 
 29  Campaign_Start          15977 non-null  object 
 30  Campaign_End            15977 non-null  object 
 31  duration_day            15977 non-null  object 
 32  Msg1_polarity           15977 non-null  float64
 33  Msg1_subjectivity       15977 non-null  float64
 34  Msg2_polarity           15977 non-null  float64
 35  Msg2_subjectivity       15977 non-null  float64
 36  Msg3_polarity           15977 non-null  float64
 37  Msg3_subjectivity       15977 non-null  float64
 38  Msg4_polarity           15977 non-null  float64
 39  Msg4_subjectivity       15977 non-null  float64
 40  Msg5_polarity           15977 non-null  float64
 41  Msg5_subjectivity       15977 non-null  float64
 42  Total_Msg_polarity      15977 non-null  float64
 43  Total_Msg_subjectivity  15977 non-null  float64
 44  Total_similarity        15977 non-null  float64
 45  Total_distance          15977 non-null  int64  
 46  Msg1_category           15977 non-null  int64  
 47  Msg2_category           15977 non-null  int64  
 48  Msg3_category           15977 non-null  int64  
 49  Msg4_category           15977 non-null  int64  
 50  Msg5_category           15977 non-null  int64  
dtypes: float64(15), int64(20), object(16)
memory usage: 6.2+ MB
```

In [ ]:

In [40]:
```
extract_data[20:].info()
```

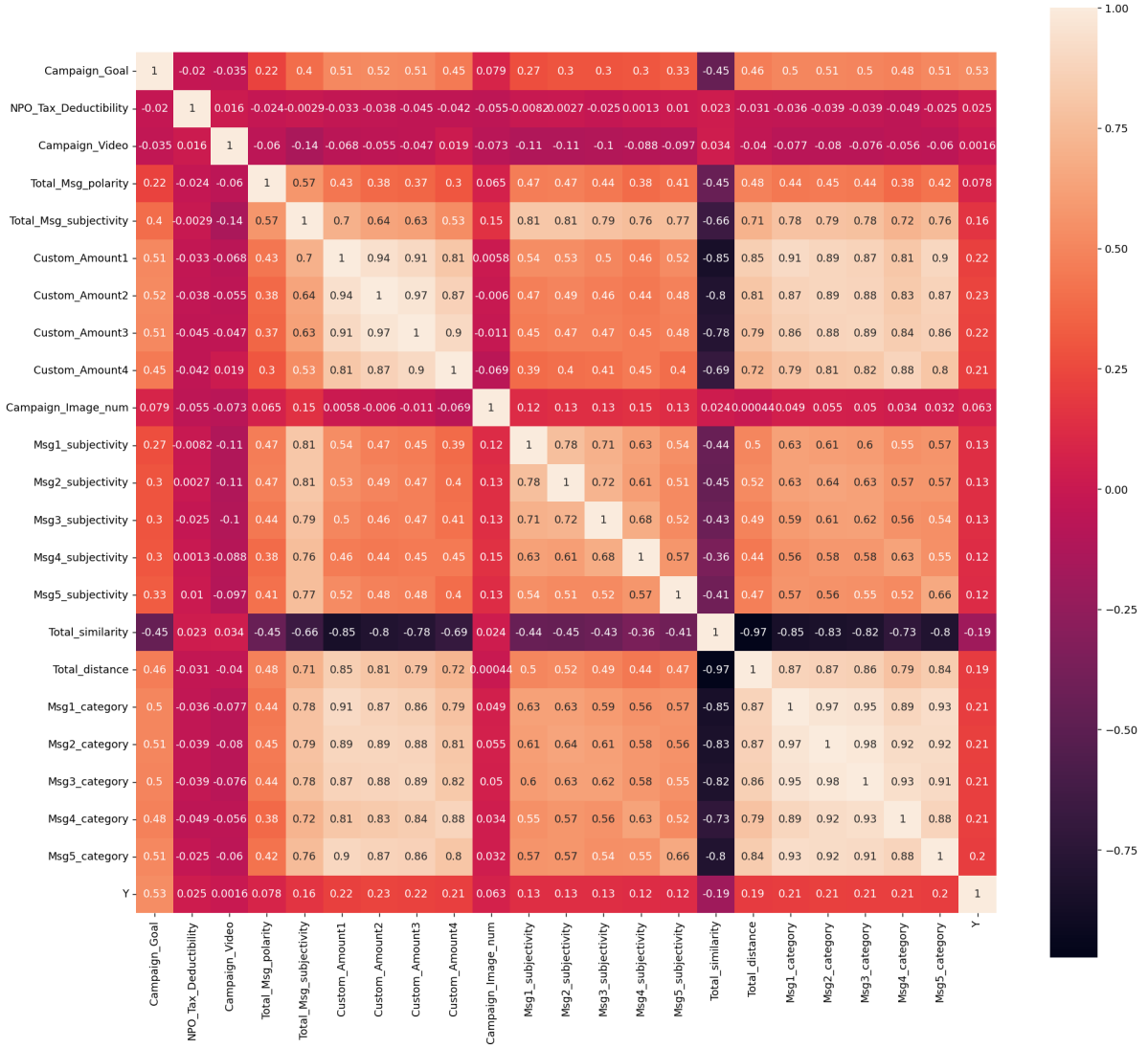
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15957 entries, 20 to 15976
Data columns (total 51 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Actual_Donation_Amount  15957 non-null  float64
 1   NPO_Tax_Deductibility   15957 non-null  int64  
 2   Distinct_Donors         15957 non-null  int64  
 3   Campaign_Goal           15957 non-null  int64  
 4   Campaign_Start_Date     15957 non-null  object 
 5   Campaign_End_Date       15957 non-null  object 
 6   Campaign_Image1         15957 non-null  int64  
 7   Campaign_Image2         15957 non-null  int64  
 8   Campaign_Image3         15957 non-null  int64  
 9   Campaign_Image4         15957 non-null  int64  
 10  Campaign_Image5         15957 non-null  int64  
 11  Campaign_Video          15957 non-null  int64  
 12  Impact_Message1         15957 non-null  object 
 13  Impact_Message2         15957 non-null  object 
 14  Impact_Message3         15957 non-null  object 
 15  Impact_Message4         15957 non-null  object 
 16  Impact_Message5         15957 non-null  object 
 17  Custom_Amount1          15957 non-null  int64  
 18  Custom_Amount2          15957 non-null  int64  
 19  Custom_Amount3          15957 non-null  int64  
 20  Custom_Amount4          15957 non-null  int64  
 21  Donation_per_donor      15957 non-null  float64
 22  Campaign_Image_num      15957 non-null  int64  
 23  Campaign_Start_Day      15957 non-null  object 
 24  Campaign_Start_Month    15957 non-null  object 
 25  Campaign_Start_Year     15957 non-null  object 
 26  Campaign_End_Day        15957 non-null  object 
 27  Campaign_End_Month      15957 non-null  object 
 28  Campaign_End_Year       15957 non-null  object 
 29  Campaign_Start          15957 non-null  object 
 30  Campaign_End            15957 non-null  object 
 31  duration_day            15957 non-null  object 
 32  Msg1_polarity           15957 non-null  float64
 33  Msg1_subjectivity       15957 non-null  float64
 34  Msg2_polarity           15957 non-null  float64
 35  Msg2_subjectivity       15957 non-null  float64
 36  Msg3_polarity           15957 non-null  float64
 37  Msg3_subjectivity       15957 non-null  float64
 38  Msg4_polarity           15957 non-null  float64
 39  Msg4_subjectivity       15957 non-null  float64
 40  Msg5_polarity           15957 non-null  float64
 41  Msg5_subjectivity       15957 non-null  float64
 42  Total_Msg_polarity      15957 non-null  float64
 43  Total_Msg_subjectivity  15957 non-null  float64
 44  Total_similarity        15957 non-null  float64
 45  Total_distance          15957 non-null  int64  
 46  Msg1_category           15957 non-null  int64  
 47  Msg2_category           15957 non-null  int64  
 48  Msg3_category           15957 non-null  int64  
 49  Msg4_category           15957 non-null  int64  
 50  Msg5_category           15957 non-null  int64  
dtypes: float64(15), int64(20), object(16)
memory usage: 6.2+ MB
```

In [41]:
```
#numeric_features Store the following variables that need to draw correlatic
```

```
numeric_features = [ 'Campaign_Goal', 'NPO_Tax_Deductibility',
       'Campaign_Video', 'Total_Msg_polarity', 'Total_Msg_subjectivity',
       'Custom_Amount1', 'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount4
       'Campaign_Image_num', 'duration_day', 'Msg1_subjectivity',
       'Msg2_subjectivity', 'Msg3_subjectivity', 'Msg4_subjectivity',
       'Msg5_subjectivity', 'Total_similarity', 'Total_distance',
       'Msg1_category', 'Msg2_category', 'Msg3_category', 'Msg4_category',
       'Msg5_category']

numeric_features2 = ['Actual_Donation_Amount', 'Campaign_Goal', 'duration_da
       'Campaign_Video',
       'Msg1_category', 'Msg2_category', 'Msg3_category', 'Msg4_category',
       'Msg5_category','Total_Msg_polarity', 'Total_Msg_subjectivity','Total

#Correlation analysis
price_numeric = extract_data[numeric_features]
correlation = price_numeric.corr()
y_train = Orginal_data['Actual_Donation_Amount']
corr = plt.subplots(figsize = (18,16), dpi=128)
corr= sns.heatmap(price_numeric.assign(Y=y_train).corr(method='spearman'), a
```



In [ ]:

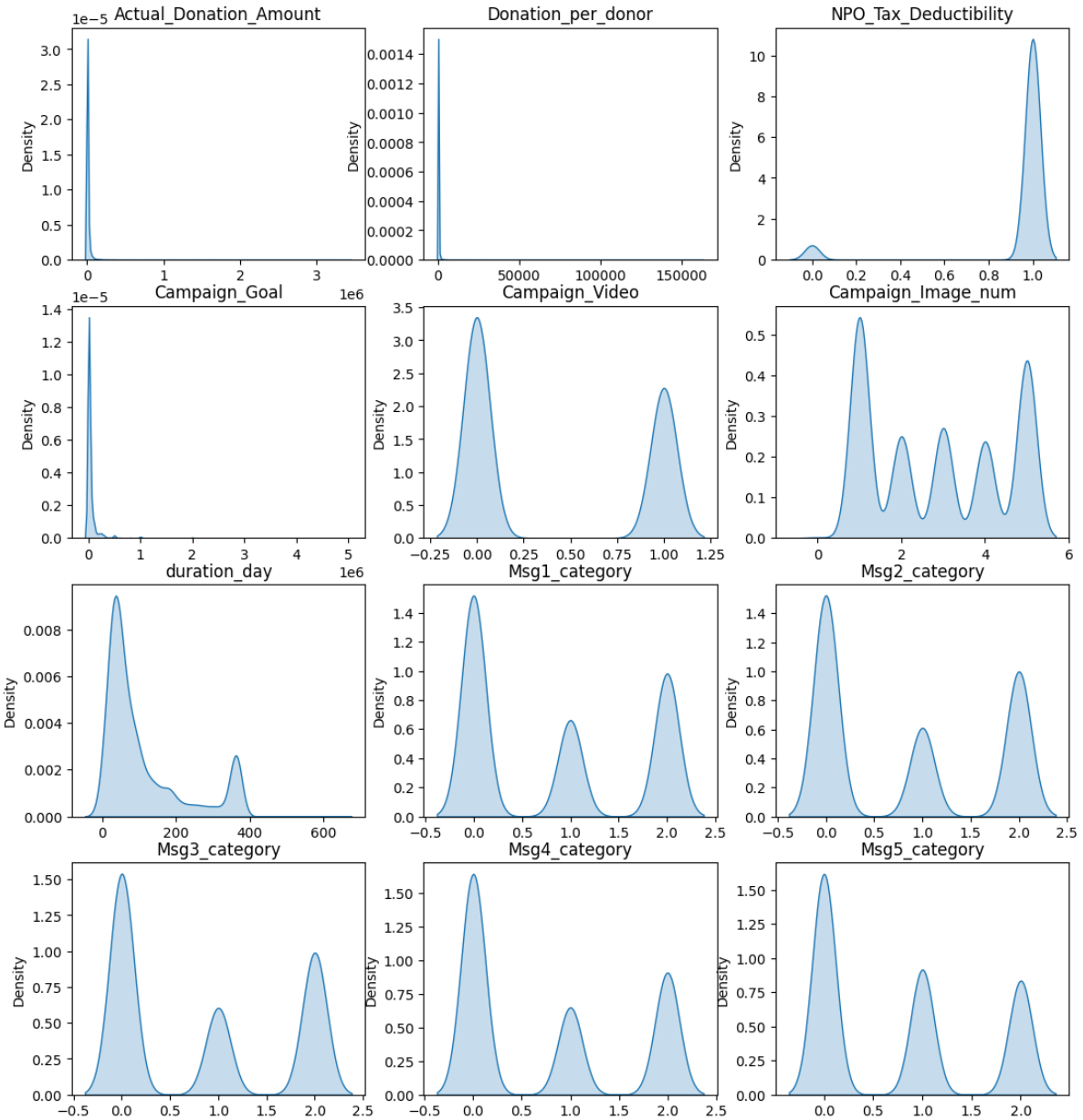
Modeling verfication

In [51]:
```
variable_list1 = ['Actual_Donation_Amount', 'Donation_per_donor', 'NPO_Tax_D
                  'Msg1_category', 'Msg2_category', 'Msg3_category', 'Msg4
                  'Msg5_category']
variables_data1 = extract_data[variable_list1]
```

Variance, Average, Max, Min, Median calculation

In [52]:
```
from scipy import stats
i = 0
plt.figure(figsize=(13, 14))
plt.xticks([])
for title in variable_list1:
    plt.subplot(4,3,i+1)
    plt.title(title)
    sns.kdeplot(extract_data[title], shade=True)
    plt.xlabel(" ")
    i += 1


#plt.hist(extract_data['Campaign_Goal'], bins=80, histtype="stepfilled", alp
```



In [46]:
```
for title in variable_list1:
    extract_data[title] = pd.to_numeric( extract_data[title])
    print( title, "Variance:" ,np.var(extract_data[title]))
    print( title, "Max:" ,np.max(extract_data[title]))
    print( title, "Min:" ,np.min(extract_data[title]))
    print( title, "Median:",np.median(extract_data[title]))
    #print( title, "Average:",np.average(extract_data[title]))
```

```
Actual_Donation_Amount Variance: 3966715831.57211
Actual_Donation_Amount Max: 3431670.0
Actual_Donation_Amount Min: 0.0
Actual_Donation_Amount Median: 1300.0
Donation_per_donor Variance: 2492432.0075160954
Donation_per_donor Max: 163050.0
Donation_per_donor Min: 0.0
Donation_per_donor Median: 84.33734939759036
Campaign_Goal Variance: 23847804493.221405
Campaign_Goal Max: 5000000
Campaign_Goal Min: 100
Campaign_Goal Median: 5000.0
Campaign_Video Variance: 0.24090726682985392
Campaign_Video Max: 1
Campaign_Video Min: 0
Campaign_Video Median: 0.0
Campaign_Image_num Variance: 2.5244628641686444
Campaign_Image_num Max: 5
Campaign_Image_num Min: 0
Campaign_Image_num Median: 3.0
duration_day Variance: 12085.337206651222
duration_day Max: 630
duration_day Min: 0
duration_day Median: 60.0
Msg1_category Variance: 0.7620336235591901
Msg1_category Max: 2
Msg1_category Min: 0
Msg1_category Median: 1.0
Msg2_category Variance: 0.7771878089650285
Msg2_category Max: 2
Msg2_category Min: 0
Msg2_category Median: 1.0
Msg3_category Variance: 0.7761966959381182
Msg3_category Max: 2
Msg3_category Min: 0
Msg3_category Median: 1.0
Msg4_category Variance: 0.7444860229679618
Msg4_category Max: 2
Msg4_category Min: 0
Msg4_category Median: 0.0
Msg5_category Variance: 0.6741166689869394
Msg5_category Max: 2
Msg5_category Min: 0
Msg5_category Median: 1.0
```

In [53]:
```
variable_list2 = ['Actual_Donation_Amount','Donation_per_donor','Campaign_Go
       'Campaign_Video', 'Total_Msg_polarity', 'Total_Msg_subjectivity',
       'Custom_Amount1', 'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount4
       'Campaign_Image_num', 'duration_day', 'Msg1_subjectivity',
       'Msg2_subjectivity', 'Msg3_subjectivity', 'Msg4_subjectivity',
       'Msg5_subjectivity', 'Total_similarity', 'Total_distance',
       'Msg1_category', 'Msg2_category', 'Msg3_category', 'Msg4_category',
       'Msg5_category']
variables_data2 = extract_data[variable_list2]
```

The Linear regression of selected variables

In [56]:
```
import statsmodels.formula.api as smf

model = smf.ols(formula = 'Actual_Donation_Amount ~ Campaign_Goal + Campaign
    Campaign_Image_num  + duration_day + Msg1_category + Msg2_category + Msg

results1 = model.summary()
print(results1)
```

OLS Regression Results

Dep. Variable:	Actual_Donation_Amount	R-squared:	0.304
Model:	OLS	Adj. R-squared:	0.304
Method:	Least Squares	F-statistic:	698.1
Date:	Sun, 13 Nov 2022	Prob (F-statistic):	0.00
Time:	11:32:51	Log-Likelihood:	-1.9633e+05
No. Observations:	15977	AIC:	3.927e+05
Df Residuals:	15966	BIC:	3.928e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1310.9601	1989.760	-0.659	0.510	-5211.113	2589.193
Campaign_Goal	0.2281	0.003	81.990	0.000	0.223	0.234
Campaign_Video	1456.2074	852.903	1.707	0.088	-215.579	3127.994
NPO_Tax_Deductibility	2880.9391	1739.034	1.657	0.098	-527.762	6289.641
Campaign_Image_num	128.8561	264.632	0.487	0.626	-389.852	647.565
duration_day	-10.2333	3.946	-2.593	0.010	-17.967	-2.499
Msg1_category	-2251.8177	1859.567	-1.211	0.226	-5896.779	1393.143
Msg2_category	1032.8411	2549.171	0.405	0.685	-3963.821	6029.503
Msg3_category	-905.2139	2384.096	-0.380	0.704	-5578.310	3767.883
Msg4_category	-314.8923	1297.695	-0.243	0.808	-2858.520	2228.735
Msg5_category	445.6163	1233.503	0.361	0.718	-1972.189	2863.422

Omnibus:	35661.352	Durbin-Watson:	1.942
Prob(Omnibus):	0.000	Jarque-Bera (JB):	500890044.337
Skew:	20.497	Prob(JB):	0.00
Kurtosis:	869.450	Cond. No.	1.24e+06

Notes:

```
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 1.24e+06. This might indicate that there
are
strong multicollinearity or other numerical problems.
```

The Linear regression with more variables

In [60]:
```
model2 = smf.ols(formula = 'Actual_Donation_Amount ~ Campaign_Goal + NPO_Tax
       +Campaign_Image_num+duration_day+Msg1_subjectivity\
       +Msg2_subjectivity+Msg3_subjectivity+Msg4_subjectivity\
       +Msg5_subjectivity+Total_similarity+Total_distance\
       +Msg1_category+ Msg2_category + Msg3_category + Msg4_category +Msg5_c
results2 = model2.summary()
print(results2)
```

OLS Regression Results

Dep. Variable:	Actual_Donation_Amount	R-squared:	0.314
Model:	OLS	Adj. R-squared:	0.313
Method:	Least Squares	F-statistic:	331.6
Date:	Sun, 13 Nov 2022	Prob (F-statistic):	0.00
Time:	11:34:34	Log-Likelihood:	-1.9622e+05
No. Observations:	15977	AIC:	3.925e+05
Df Residuals:	15954	BIC:	3.927e+05
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1619.3993	6606.661	-0.245	0.806	-1.46e+04	1.13e+04
Campaign_Goal	0.2329	0.003	83.219	0.000	0.227	0.238
NPO_Tax_Deductibility	2244.7877	1735.079	1.294	0.196	-1156.162	5645.737
Campaign_Video	1455.8683	850.688	1.711	0.087	-211.576	3123.313
Total_Msg_polarity	-2424.6686	1079.932	-2.245	0.025	-4541.456	-307.881
Total_Msg_subjectivity	1943.4403	708.242	2.744	0.006	555.207	3331.674
Custom_Amount1	-1.4295	0.240	-5.955	0.000	-1.900	-0.959
Custom_Amount2	-0.1177	0.361	-0.326	0.745	-0.826	0.590
Custom_Amount3	-0.0744	0.330	-0.225	0.822	-0.722	0.573
Custom_Amount4	-0.1029	0.152	-0.676	0.499	-0.401	0.196
Campaign_Image_num	40.5187	266.697	0.152	0.879	-482.238	563.276
duration_day	-10.4315	3.927	-2.656	0.008	-18.130	-2.733
Msg1_subjectivity	1.067e+04	3176.407	3.359	0.001	4444.410	1.69e+04
Msg2_subjectivity	-1.124e+04	3233.922	-3.476	0.001	-1.76e+04	-4902.012
Msg3_subjectivity	2.28e+04	3360.905	6.783	0.000	1.62e+04	2.94e+04
Msg4_subjectivity	-1.066e+04	3043.884	-3.504	0.000	-1.66e+04	-4698.320
Msg5_subjectivity	-9617.7378	2506.873	-3.837	0.000	-1.45e+04	-4703.984
Total_similarity	327.5966	1600.996	0.205	0.838	-2810.535	3465.728

```
Total_distance             7.2033      6.477      1.112      0.266      -5.
492      19.899
Msg1_category          -3406.3712   1939.860     -1.756      0.079   -7208.
716     395.974
Msg2_category           1026.4132   2619.997      0.392      0.695   -4109.
075    6161.902
Msg3_category          -4061.3118   2423.154     -1.676      0.094   -8810.
966     688.342
Msg4_category           1008.1028   1627.788      0.619      0.536   -2182.
544    4198.750
Msg5_category           2326.4253   1321.522      1.760      0.078    -263.
907    4916.758
==============================================================================
===
Omnibus:                    35371.049   Durbin-Watson:                   1.
943
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        477276588.
419
Skew:                          20.099   Prob(JB):
0.00
Kurtosis:                     848.772   Cond. No.                     3.63e
+19
==============================================================================
===

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The smallest eigenvalue is 3.13e-25. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Donation per donor Linear regression result by using selected variables

In [ ]:
```
model = smf.ols(formula = 'Donation_per_donor ~ Campaign_Goal + NPO_Tax_Dedu
       +Campaign_Image_num+duration_day+Msg1_subjectivity\
       +Msg2_subjectivity+Msg3_subjectivity+Msg4_subjectivity\
       +Msg5_subjectivity+Total_similarity+Total_distance\
       +Msg1_category+ Msg2_category + Msg3_category + Msg4_category +Msg5_c
results2 = model.summary()
print(results2)
```

OLS Regression Results

Dep. Variable:	Donation_per_donor	R-squared:	0.007
Model:	OLS	Adj. R-squared:	0.005
Method:	Least Squares	F-statistic:	4.780
Date:	Sun, 13 Nov 2022	Prob (F-statistic):	8.99e-13
Time:	02:05:34	Log-Likelihood:	-1.4028e+05
No. Observations:	15977	AIC:	2.806e+05
Df Residuals:	15954	BIC:	2.808e+05
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-65.2901	199.259	-0.328	0.743	-455.860	325.280
Campaign_Goal	0.0005	8.44e-05	5.550	0.000	0.000	0.001
NPO_Tax_Deductibility	-29.5756	52.331	-0.565	0.572	-132.149	72.998
Campaign_Video	14.3132	25.657	0.558	0.577	-35.977	64.604
Total_Msg_polarity	42.1417	32.571	1.294	0.196	-21.701	105.985
Total_Msg_subjectivity	-30.5281	21.361	-1.429	0.153	-72.398	11.342
Custom_Amount1	0.0073	0.007	1.014	0.311	-0.007	0.022
Custom_Amount2	0.0139	0.011	1.278	0.201	-0.007	0.035
Custom_Amount3	0.0104	0.010	1.049	0.294	-0.009	0.030
Custom_Amount4	-0.0091	0.005	-1.971	0.049	-0.018	-5.1e-05
Campaign_Image_num	22.3573	8.044	2.779	0.005	6.591	38.124
duration_day	0.2883	0.118	2.434	0.015	0.056	0.520
Msg1_subjectivity	-41.4160	95.801	-0.432	0.666	-229.198	146.366
Msg2_subjectivity	25.4180	97.536	0.261	0.794	-165.764	216.600
Msg3_subjectivity	107.0225	101.366	1.056	0.291	-91.666	305.711
Msg4_subjectivity	-39.9107	91.805	-0.435	0.664	-219.858	140.036
Msg5_subjectivity	-81.6419	75.608	-1.080	0.280	-229.842	66.559
Total_similarity	30.4887	48.287	0.631	0.528	-64.158	125.136

```
Total_distance            -0.2056      0.195     -1.053      0.293      -0.
589       0.177
Msg1_category             67.0142     58.507      1.145      0.252     -47.
666     181.694
Msg2_category            -17.6539     79.020     -0.223      0.823    -172.
542     137.234
Msg3_category            -30.7882     73.083     -0.421      0.674    -174.
039     112.463
Msg4_category             -5.4594     49.095     -0.111      0.911    -101.
690      90.772
Msg5_category             97.5549     39.858      2.448      0.014      19.
430     175.680
==============================================================================
===
Omnibus:                    58192.128   Durbin-Watson:                   1.
986
Prob(Omnibus):                  0.000   Jarque-Bera (JB):      36028162900.
975
Skew:                          76.283   Prob(JB):
0.00
Kurtosis:                    7358.050   Cond. No.                     3.63e
+19
==============================================================================
===

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The smallest eigenvalue is 3.13e-25. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

test

In [ ]:

```
import nltk
# nltk.download('punkt')
from textblob import TextBlob
texts=["Thank you",'OK!']
for text in texts:
    blob=TextBlob(text)
    emotion=blob.sentiment
    print(emotion)
```

```
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.625, subjectivity=0.5)
```