Marketing Strategy Analysis

Programming Script and Technical Report

LinkedIn: Okino Kamali Leiba (https://www.linkedin.com/in/okinoleiba)

Table of Content

1. Introduction

- What is the impact of each marketing strategy and sales visit on Sales (Amount Collected)?
- Is the same strategy valid for all the different Client Types?

Imports

- Sys module gives access to variables and functions used or maintained by the interpreter
- Pandas has data analysis and manipulation libraries
- Numpy for general array computations
- Matplotlib contains libraries for creating static, animated, and interactive visualizations in Python
- Seaborn for Python data visualization based on Matplotlib
- Scipy provides algorithms for scientific computing in Python

2. Data Loading and Quality Checks

In [ ]:
```
#import modules
import sys, pandas as pd, matplotlib as ml, seaborn as sns, numpy as np, sc
ipy.stats
```

In [ ]:
```
file_path = 'C:/Users/Owner/source/vsc_repo/marketing_prog_cookbook/campaig
n-data.csv'
campaign_data = pd.read_csv(file_path, sep=",", header=0, engine="c", nrows
=2979, keep_default_na=True,
encoding="utf-8")
campaign_data.columns
```

Out[ ]:
```
Index(['Client ID', 'Client Type', 'Number of Customers', 'Montly Target',
       'Zip Code', 'Calendardate', 'Amount Collected', 'Unit Sold',
       'Campaign (Email)', 'Campaign (Flyer)', 'Campaign (Phone)',
       'Sales Contact 1', 'Sales Contact 2', 'Sales Contact 3',
       'Sales Contact 4', 'Sales Contact 5', 'Number of Competition'],
      dtype='object')
```

3. Exploratory Data Analysis

3.1 Exploring and Understanding the Basics Data

1. General Review and Exploration

2. Distribution of Data Across Different Accounts

3. Difference of Sales in Account Types (Using Categorical Mean)

4. Statistical Summary

In [ ]:

```
#data exploration no visualization
#Target/Regressand/Dependent Variable: Amount Collected
#Regressor/Predictors/Indpendent Variables: Campaign (Email), Campaign (Fly
er), Campaign (Phone), Sales Contact 1, Sales Contact 2,
#Sales Contact 3, Sales Contact 4, Sales Contact 5
campaign_data.head(6)
```

Out[ ]:

	Client ID	Client Type	Number of Customers	Montly Target	Zip Code	Calendardate	Amount Collected	Unit Sold	Campa (Em
0	ID-987275	Medium Facility	2800	125	1003	16-01-2014	0	0	0.0
1	ID-987275	Medium Facility	2800	125	1003	16-02-2014	3409460	24	0.0
2	ID-987275	Medium Facility	2800	125	1003	18-03-2014	10228384	75	0.0
3	ID-987275	Medium Facility	2800	125	1003	18-04-2014	17047304	123	0.0
4	ID-987275	Medium Facility	2800	125	1003	19-05-2014	23866224	171	0.0
5	ID-987275	Medium Facility	2800	125	1003	16-06-2014	27275684	198	0.0

In [ ]:

```
#data exploration no visualization
campaign_data.tail(6)
```

Out[ ]:

	Client ID	Client Type	Number of Customers	Montly Target	Zip Code	Calendardate	Amount Collected	Unit Sold	Cam (I
2970	ID-987463	Small Facility	800	20	1003	17-07-2015	0	0	0.0
2971	ID-987463	Small Facility	800	20	1003	16-08-2015	0	0	0.0
2972	ID-987463	Small Facility	800	20	1003	16-09-2015	0	0	0.0
2973	ID-987463	Small Facility	800	20	1003	16-10-2015	0	0	0.0
2974	ID-987463	Small Facility	800	20	1003	16-11-2015	0	0	0.0
2975	ID-987463	Small Facility	800	20	1003	17-12-2015	3409460	24	0.0

In [ ]:
```
#data exploration no visualization
campaign_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2976 entries, 0 to 2975
Data columns (total 17 columns):
Client ID                 2976 non-null object
Client Type               2976 non-null object
Number of Customers       2976 non-null int64
Montly Target             2976 non-null int64
Zip Code                  2976 non-null int64
Calendardate              2976 non-null object
Amount Collected          2976 non-null int64
Unit Sold                 2976 non-null int64
Campaign (Email)          2976 non-null float64
Campaign (Flyer)          2976 non-null float64
Campaign (Phone)          2976 non-null float64
Sales Contact 1           2976 non-null float64
Sales Contact 2           2976 non-null float64
Sales Contact 3           2976 non-null float64
Sales Contact 4           2976 non-null float64
Sales Contact 5           2976 non-null float64
Number of Competition     2976 non-null object
dtypes: float64(8), int64(5), object(4)
memory usage: 395.3+ KB
```

In [ ]:
```
#data exploration no visualization
#take note of Campign (Flyer) and Sales Contact 2
campaign_data[['Client ID', 'Client Type', 'Number of Customers', 'Montly T
arget',
       'Calendardate', 'Amount Collected', 'Unit Sold',
       'Campaign (Email)', 'Campaign (Flyer)', 'Campaign (Phone)',
       'Sales Contact 1', 'Sales Contact 2', 'Sales Contact 3',
       'Sales Contact 4', 'Sales Contact 5', 'Number of Competition']].desc
ribe().round(decimals=2)
```

Out[ ]:

	Number of Customers	Montly Target	Amount Collected	Unit Sold	Campaign (Email)	Campaign (Flyer)	Campa (Pho
count	2976.00	2976.00	2.976000e+03	2976.00	2976.00	2976.00	2976.00
mean	1456.94	75.08	1.700440e+07	121.46	143284.96	685418.60	29777.4
std	1669.85	87.04	3.025803e+07	216.41	723045.16	1727587.37	383213.
min	0.00	5.00	-2.216150e+07	-63.00	0.00	0.00	0.00
25%	240.00	10.00	0.000000e+00	0.00	0.00	0.00	0.00
50%	960.00	47.50	3.409460e+06	24.00	0.00	0.00	0.00
75%	2090.00	101.25	2.045676e+07	147.00	0.00	81482.85	0.00
max	9840.00	510.00	2.079771e+08	1500.00	11446733.30	13593951.20	961738(

In [ ]:
```
#clean data
##modified base dataset: rename columns and changed axis: 1##
campaign_data.dropna(axis=0,how="any",)
campaign_data.duplicated(keep="first")
campaign_data.groupby('Client Type')
campaign_data = campaign_data.rename({'Montly Target':'Monthly Target','Cal
endardate':'Calender Date','Campaign (Email)':'Marketing Channel\
       (Email)','Campaign (Flyer)':'Marketing Channel(Flyer)', 'Campaign (P
hone)':'Marketing Channel(Phone)','Number of Competition':'Level\
        of Competition'},
 axis=1,inplace=False)
#campaign_data = campaign_data.replace({'Montly Target':'Monthly Target','C
alendardate':'Calender Date','Campaign (Email)':'Marketing Channel(Email)',
#'Campaign (Flyer)':'Marketing Channel(Flyer)', 'Campaign (Phone)':'Marketi
ng Channel(Phone)','Number of Competition':'Level of Competition'}, inplace
=False)
campaign_data = campaign_data.set_axis(['Client ID', 'Client Type', 'Number
of Customers', 'Monthly Target',
       'Zip Code', 'Calendar Date', 'Amount Collected', 'Unit Sold',
       'Marketing Channel(Email)', 'Marketing Channel(Flyer)', 'Marketing C
hannel(Phone)',
       'Sales Contact 1', 'Sales Contact 2', 'Sales Contact 3',
       'Sales Contact 4', 'Sales Contact 5', 'Level of Competition'], axis=
1, inplace=False)
campaign_data.head(0)
```

Out[ ]:

	Client ID	Client Type	Number of Customers	Monthly Target	Zip Code	Calendar Date	Amount Collected	Unit Sold	Marketing Channel(Email)

4. Feature Additions and Engineering

In [ ]:
```
#additional date features
##modified base dataset: added new columns##
campaign_data["Calendar Date"]=pd.to_datetime(campaign_data["Calendar Dat
e"],errors="raise",dayfirst=True,yearfirst=True)
campaign_data["Calendar_Month"]=campaign_data["Calendar Date"].dt.month
campaign_data["Calendar_Year"]=campaign_data["Calendar Date"].dt.year
```

5. Statistical Analysis

5.1 Statistical Analysis - Answering the Questions

1. Impact of Marketing Strategy on Sales (Using Correlation and Linear Regression)
2. Impact of Competition on Sales
3. How Different Types of Client Can Have Different Strategies (Catorgize Question 1 and Question 2 Based on Account Type)

5.2 Impact of Marketing Strategy on Sales

Understanding of Distrubtions

In [ ]:
```
campaign_data["Client Type"].value_counts(normalize=True,sort=True,ascendin
g=True).round(decimals=2)
```

Out[ ]:
```
Private Facility    0.09
Medium Facility     0.17
Small Facility      0.28
Large Facility      0.46
Name: Client Type, dtype: float64
```

In [ ]:
```
pd.crosstab(campaign_data["Level of Competition"], campaign_data["Client Ty
pe"],margins=True, normalize="columns").round(decimals=2)
```

Out[ ]:

Client Type	Large Facility	Medium Facility	Private Facility	Small Facility	All
Level of Competition					
High	0.17	0.17	0.17	0.17	0.17
Low	0.83	0.83	0.83	0.83	0.83

In [ ]:
```
campaign_data[['Client ID', 'Client Type', 'Number of Customers', 'Monthly
Target',
       'Calendar Date', 'Amount Collected', 'Unit Sold',
       'Marketing Channel(Email)', 'Marketing Channel(Flyer)', 'Marketing C
hannel(Phone)',
       'Sales Contact 1', 'Sales Contact 2', 'Sales Contact 3',
       'Sales Contact 4', 'Sales Contact 5', 'Level of Competition']].group
by("Level of Competition",).mean().round(decimals=2)
```

Out[ ]:

	Number of Customers	Monthly Target	Amount Collected	Unit Sold	Marketing Channel(Email)	Marketing Channel(Flyer
Level of Competition						
High	1456.94	75.08	29747888.39	213.13	105398.94	994046.72
Low	1456.94	75.08	14455700.99	103.13	150862.17	623692.98

In [ ]:
```
campaign_data[['Client Type', 'Number of Customers', 'Monthly Target',
       'Calendar Date', 'Amount Collected', 'Unit Sold',
       'Marketing Channel(Email)', 'Marketing Channel(Flyer)', 'Marketing C
hannel(Phone)',
       'Sales Contact 1', 'Sales Contact 2', 'Sales Contact 3',
       'Sales Contact 4', 'Sales Contact 5']].groupby("Client Type").mean
().round(2)
```

Out[ ]:

	Number of Customers	Monthly Target	Amount Collected	Unit Sold	Marketing Channel(Email)	Marketing Channel(Flyer)	Ch
Client Type							
Large Facility	1380.84	71.58	19998804.93	143.10	142273.61	819205.63	45
Medium Facility	3940.76	202.86	40759967.67	290.58	437217.10	1552603.27	49
Private Facility	400.73	20.45	5030245.94	35.78	5183.72	227291.88	55
Small Facility	422.51	21.29	1637758.72	11.69	11975.99	91208.75	0.(

In [ ]:
```
campaign_data[['Number of Customers', 'Monthly Target',
       'Calendar Date', 'Amount Collected', 'Unit Sold',
       'Marketing Channel(Email)', 'Marketing Channel(Flyer)', 'Marketing C
hannel(Phone)',
       'Sales Contact 1', 'Sales Contact 2', 'Sales Contact 3',
       'Sales Contact 4', 'Sales Contact 5']].std().round(decimals=3)
```

Out[ ]:
```
Number of Customers         1.669849e+03
Monthly Target              8.704200e+01
Amount Collected            3.025803e+07
Unit Sold                   2.164140e+02
Marketing Channel(Email)    7.230452e+05
Marketing Channel(Flyer)    1.727587e+06
Marketing Channel(Phone)    3.832134e+05
Sales Contact 1             1.034882e+06
Sales Contact 2             3.396991e+06
Sales Contact 3             3.271349e+06
Sales Contact 4             3.869872e+05
Sales Contact 5             8.905955e+04
dtype: float64
```

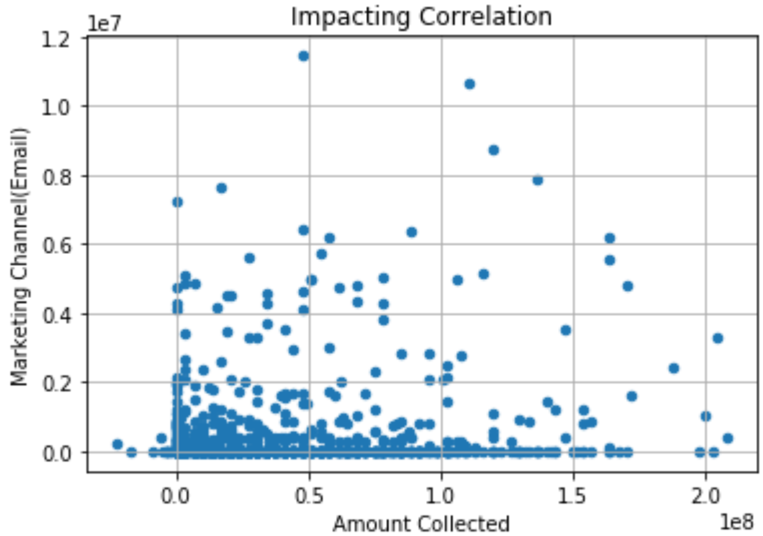
In [ ]:
```
campaign_data[['Number of Customers', 'Monthly Target',
       'Calendar Date', 'Amount Collected', 'Unit Sold',
       'Marketing Channel(Email)', 'Marketing Channel(Flyer)', 'Marketing C
hannel(Phone)',
       'Sales Contact 1', 'Sales Contact 2', 'Sales Contact 3',
       'Sales Contact 4', 'Sales Contact 5']].var().round(decimals=3)
```

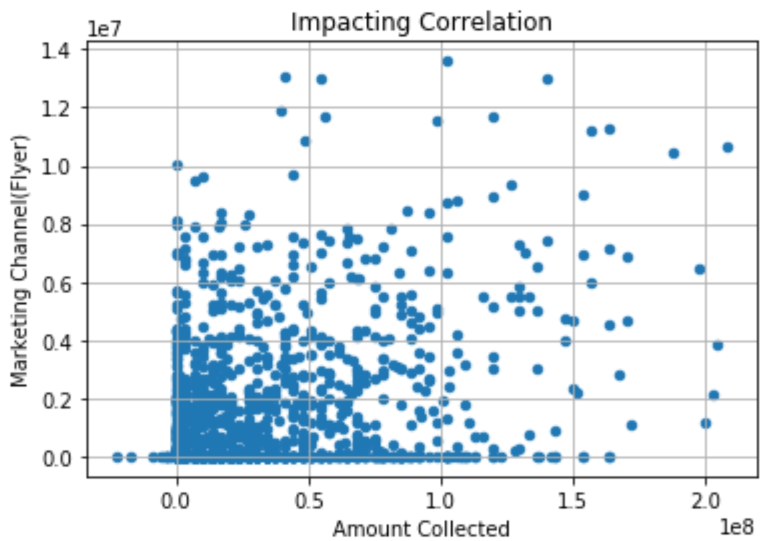
Out[ ]:
```
Number of Customers         2.788395e+06
Monthly Target              7.576330e+03
Amount Collected            9.155484e+14
Unit Sold                   4.683501e+04
Marketing Channel(Email)    5.227943e+11
Marketing Channel(Flyer)    2.984558e+12
Marketing Channel(Phone)    1.468525e+11
Sales Contact 1             1.070981e+12
Sales Contact 2             1.153955e+13
Sales Contact 3             1.070172e+13
Sales Contact 4             1.497591e+11
Sales Contact 5             7.931604e+09
dtype: float64
```

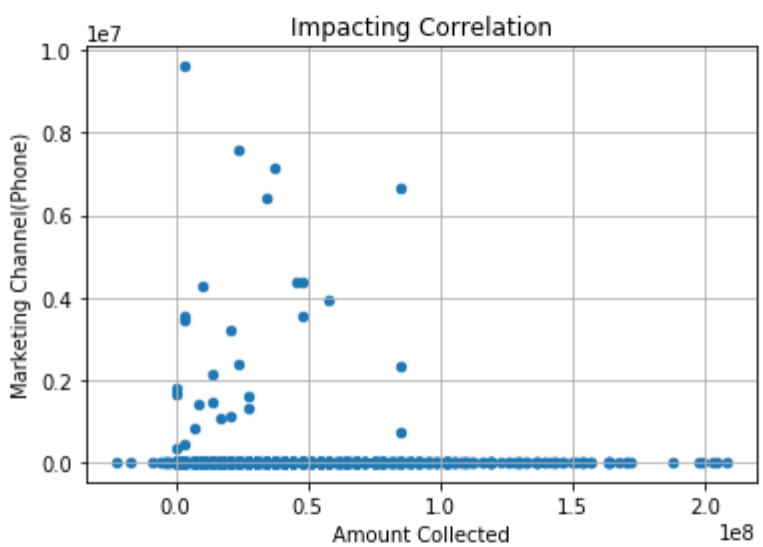
In [ ]:

```
campaign_data.plot(x="Amount Collected", y="Marketing Channel(Email)",kind
="scatter",legend=True,title="Impacting Correlation",grid=True)
campaign_data.plot(x="Amount Collected", y="Marketing Channel(Flyer)",kind
="scatter",legend=True,title="Impacting Correlation",grid=True)
campaign_data.plot(x="Amount Collected", y="Marketing Channel(Phone)",kind
="scatter",legend=True,title="Impacting Correlation",grid=True)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x19f717bc4e0>







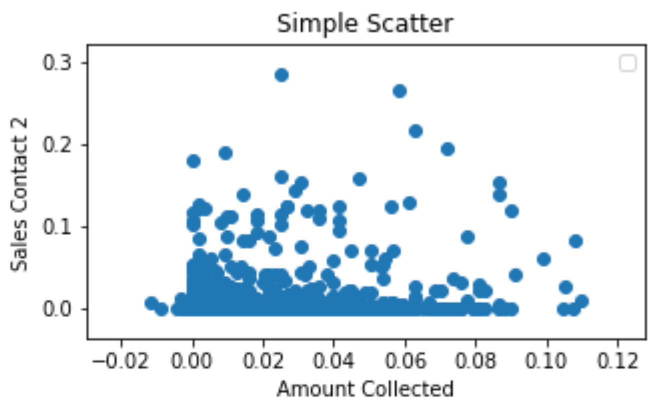
In [ ]:
```
#strong correlation between Amount Collected and Sales Contact 2
from sklearn import preprocessing

x = np.array([element for element in campaign_data["Amount Collected"]]) #n
umpy arrary
y = np.array([element for element in campaign_data["Marketing Channel(Emai
l)"]])
normalize_x = preprocessing.normalize([x] )#normalize data
normalize_y = preprocessing.normalize([y])
fig = ml.pyplot.figure() #matplotlab plot
fig, campaign_plot = ml.pyplot.subplots(figsize=(5, 2.7))
campaign_plot.scatter(normalize_x, normalize_y)
campaign_plot.set_xlabel("Amount Collected")
campaign_plot.set_ylabel("Sales Contact 2")
campaign_plot.set_title("Simple Scatter")
campaign_plot.legend();


# X, Y = np.meshgrid(np.linspace(-3, 3, 128), np.linspace(-3, 3, 128))
# Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)
# co = campaign_plot[0,1].contourf(X, Y, Z, levels=np.linspace(-1.25, 1.25,
11))
# fig.colorbar(co, ax=campaign_plot[0, 1])
```

```
No handles with labels found to put in legend.
```

```
<Figure size 432x288 with 0 Axes>
```



In [ ]:

```
corr_data = campaign_data.corr("pearson")[["Amount Collected"]].dropna(axis
=0, how="any")
corr_data.round(decimals=2)
```

Out[ ]:

	Amount Collected
Number of Customers	0.61
Monthly Target	0.61
Amount Collected	1.00
Unit Sold	1.00
Marketing Channel(Email)	0.25
Marketing Channel(Flyer)	0.44
Marketing Channel(Phone)	0.03
Sales Contact 1	0.28
Sales Contact 2	0.55
Sales Contact 3	0.36
Sales Contact 4	0.24
Sales Contact 5	0.10
Calendar_Month	0.14
Calendar_Year	0.29

Correlation Analysis

In [ ]:

```
#consolidated strategy for targeting
import seaborn as sns, pandas as pd
correlation_data = pd.DataFrame(campaign_data[["Amount Collected","Marketin
g Channel(Email)","Marketing Channel(Flyer)","Marketing Channel(Phone)","Sa
les Contact 1",
"Sales Contact 2","Sales Contact 3","Sales Contact 4","Sales Contact 5"]].c
orr("pearson")["Amount Collected"]).reset_index()
correlation_data.columns = ["Impacting Variable", "Degree of Linear Impact
(Correlation)"]
correlation_data = correlation_data[correlation_data["Impacting Variable"]
!= "Amount Collected"]
correlation_data = correlation_data.sort_values("Degree of Linear Impact (C
orrelation)", axis=0,ascending=False,kind="quicksort",inplace=False,
na_position="first")
correlation_data.style.background_gradient(cmap=sns.light_palette("brown",n
_colors=2,reverse=False,as_cmap=True)).set_precision(2)
#correlation_data.io.Styler.background_color().set_precision(2)
```

Out[ ]:

	Impacting Variable	Degree of Linear Impact (Correlation)
5	Sales Contact 2	0.55
2	Marketing Channel(Flyer)	0.44
6	Sales Contact 3	0.36
4	Sales Contact 1	0.28
1	Marketing Channel(Email)	0.25
7	Sales Contact 4	0.24
8	Sales Contact 5	0.096
3	Marketing Channel(Phone)	0.035

In [ ]:

```
import seaborn as sns, pandas as pd
correlation_data = pd.DataFrame(campaign_data.groupby("Client Type")[["Amou
nt Collected","Marketing Channel(Email)","Marketing Channel(Flyer)","Market
ing Channel(Phone)",
"Sales Contact 1","Sales Contact 2","Sales Contact 3","Sales Contact 4", "S
ales Contact 5"]].corr("pearson")["Amount Collected"]).reset_index()
correlation_data = correlation_data.sort_values(["Client Type", "Amount Col
lected"],axis=0,ascending=False,kind="quicksort",na_position="first",inplac
e=False)
correlation_data.columns=["Account Type", "Variable Impact on Sales", "Impa
ct"]
correlation_data = correlation_data[correlation_data["Variable Impact on Sa
les"] != "Amount Collected"].reset_index(drop=True)
correlation_data.style.background_gradient(cmap=sns.light_palette("purple",
n_colors=4,reverse=False,as_cmap=True)).set_precision(2)
```

```
c:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site
-packages\matplotlib\colors.py:504: RuntimeWarning: invalid value encounter
ed in less
  xa[xa < 0] = -1
```

```
Out[ ]:
```

	Account Type	Variable Impact on Sales	Impact
0	Small Facility	Marketing Channel(Phone)	nan
1	Small Facility	Sales Contact 2	0.22
2	Small Facility	Sales Contact 3	0.068
3	Small Facility	Marketing Channel(Email)	0.06
4	Small Facility	Marketing Channel(Flyer)	0.041
5	Small Facility	Sales Contact 4	0.024
6	Small Facility	Sales Contact 5	0.00093
7	Small Facility	Sales Contact 1	-0.016
8	Private Facility	Sales Contact 2	0.57
9	Private Facility	Marketing Channel(Flyer)	0.28
10	Private Facility	Sales Contact 3	0.18
11	Private Facility	Sales Contact 5	0.13
12	Private Facility	Sales Contact 4	0.096
13	Private Facility	Marketing Channel(Phone)	0.061
14	Private Facility	Sales Contact 1	-0.0075
15	Private Facility	Marketing Channel(Email)	-0.017
16	Medium Facility	Sales Contact 2	0.51
17	Medium Facility	Marketing Channel(Flyer)	0.45
18	Medium Facility	Sales Contact 1	0.27
19	Medium Facility	Marketing Channel(Email)	0.26
20	Medium Facility	Sales Contact 3	0.22
21	Medium Facility	Sales Contact 4	0.15
22	Medium Facility	Sales Contact 5	0.1
23	Medium Facility	Marketing Channel(Phone)	0.021
24	Large Facility	Sales Contact 2	0.42
25	Large Facility	Marketing Channel(Flyer)	0.32
26	Large Facility	Sales Contact 1	0.29
27	Large Facility	Sales Contact 4	0.28
28	Large Facility	Sales Contact 3	0.19

Market Strategy Impact on Sales **(Categorized by Different Account Type)**

In [ ]:

```
##modified base dataset: renamed columns##
import statsmodels.api as sm, statsmodels.formula.api as smf
campaign_data.columns=[mystring.replace(" ", "_") for mystring in campaign_
data.columns]
campaign_data.columns=[mystring.replace("(", "_") for mystring in campaign_
data.columns]
campaign_data.columns=[mystring.replace(")", "") for mystring in campaign_d
ata.columns]
results = smf.ols('Amount_Collected ~ Marketing_Channel_Email + Marketing_C
hannel_Flyer + Marketing_Channel_Phone + Sales_Contact_1 + Sales_Contact_2
+ Sales_Contact_3 + Sales_Contact_4\
        + Sales_Contact_5',data=campaign_data, missing="raise", hasconst=Fa
lse).fit()
print(results.summary())
```

```
                              OLS Regression Results
============================================================================
===
Dep. Variable:      Amount_Collected   R-squared:                        0.
605
Model:                           OLS   Adj. R-squared:                   0.
604
Method:                Least Squares   F-statistic:                      50
4.4
Date:               Tue, 27 Dec 2022   Prob (F-statistic):
0.00
Time:                       00:57:24   Log-Likelihood:                 -545
12.
No. Observations:               2976   AIC:                           1.090
e+05
Df Residuals:                   2967   BIC:                           1.091
e+05
Df Model:                          9
Covariance Type:           nonrobust
============================================================================
================
                              coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------------
----------------
Intercept                1.481e+06   5.12e+05      2.891      0.004    4.77
e+05    2.49e+06
Marketing_Channel_Email     0.7932      0.597      1.329      0.184       -
0.377       1.963
Marketing_Channel_Flyer     3.3376      0.260     12.831      0.000
2.828       3.848
Marketing_Channel_Phone     0.0734      1.053      0.070      0.944       -
1.991       2.137
Sales_Contact_1             4.2368      0.415     10.207      0.000
3.423       5.051
Sales_Contact_2             3.6382      0.129     28.155      0.000
3.385       3.892
Sales_Contact_3             2.3432      0.131     17.925      0.000
2.087       2.600
Sales_Contact_4            10.9478      1.060     10.331      0.000
8.870      13.026
Sales_Contact_5             3.5078      4.549      0.771      0.441       -
5.412      12.428
============================================================================
===
Omnibus:                    1099.749   Durbin-Watson:                    0.
624
Prob(Omnibus):                 0.000   Jarque-Bera (JB):              7733.
226
Skew:                          1.578   Prob(JB):
0.00
Kurtosis:                     10.239   Cond. No.                       5.89
e+06
============================================================================
===
```

```
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 5.89e+06. This might indicate that there
are
strong multicollinearity or other numerical problems.
```

In [ ]:

```
html_frame = pd.read_html(results.summary().tables[1].as_html(),flavor="bs
4",encoding=None,header=0,index_col=0)[0]
```

In [ ]:

```
html_frame = html_frame.reset_index()
html_frame = html_frame[html_frame["P>|t|"] < 0.05][["index","coef"]]
#campaign_data.rename(columns={"index":"Index","coef":"Coef"})
html_frame
```

Out[ ]:

	index	coef
0	Intercept	1.481000e+06
2	Marketing_Channel_Flyer	3.337600e+00
4	Sales_Contact_1	4.236800e+00
5	Sales_Contact_2	3.638200e+00
6	Sales_Contact_3	2.343200e+00
7	Sales_Contact_4	1.094780e+01

Regression Analysis (Market Sales and Strategies - Cateforized for Different Account Types)

In [ ]:

```
consolidated_summary=pd.DataFrame()
for acctype in list(set(list(campaign_data["Client_Type"]))):
    temp_data = campaign_data[campaign_data["Client_Type"]==acctype].copy(d
eep=False)
    results = smf.ols('Amount_Collected ~ Marketing_Channel_Email + Marketi
ng_Channel_Flyer + Marketing_Channel_Phone\
         + Sales_Contact_1 + Sales_Contact_2 + Sales_Contact_3 + Sales_Cont
act_4\
        + Sales_Contact_5',data=temp_data, missing="raise", hasconst=Fals
e).fit()
    consolidated_frame = pd.read_html(results.summary().tables[1].as_html
(),flavor="bs4",encoding=None,
    header=0,index_col=0)[0].reset_index()
    consolidated_frame = consolidated_frame[consolidated_frame["P>|t|"] <
0.05][["index","coef"]]
    consolidated_frame.columns=["Variable", "Coefficient (Impact)"]
    consolidated_frame["Account Type"] = acctype
    consolidated_frame = consolidated_frame.sort_values("Coefficient (Impac
t)", ascending=False,
    kind="quicksort", inplace=False,na_position="first",axis=0)
    consolidated_frame = consolidated_frame[consolidated_frame["Variable"]
!= "Intercept"]
    print(acctype)
    consolidated_summary = consolidated_summary.append(consolidated_frame)
    print(consolidated_frame)
```

```
Private Facility
          Variable  Coefficient (Impact)      Account Type
5  Sales_Contact_2                6.6223  Private Facility
Medium Facility
                  Variable  Coefficient (Impact)     Account Type
2  Marketing_Channel_Flyer                4.1059  Medium Facility
5          Sales_Contact_2                3.5778  Medium Facility
4          Sales_Contact_1                3.1365  Medium Facility
6          Sales_Contact_3                2.1174  Medium Facility
Large Facility
                  Variable  Coefficient (Impact)     Account Type
4          Sales_Contact_1               11.6731  Large Facility
7          Sales_Contact_4               10.6145  Large Facility
5          Sales_Contact_2                4.0031  Large Facility
2  Marketing_Channel_Flyer                2.7204  Large Facility
6          Sales_Contact_3                2.0316  Large Facility
3  Marketing_Channel_Phone               -3.5361  Large Facility
Small Facility
                  Variable  Coefficient (Impact)     Account Type
5          Sales_Contact_2          8.101000e-01  Small Facility
3  Marketing_Channel_Phone         -7.137000e-07  Small Facility
```

In [ ]:

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

consolidated_summary=pd.DataFrame()
for acctype in list(set(list(campaign_data["Client_Type"]))):
    temp_data = campaign_data[campaign_data["Client_Type"]==acctype].copy(d
eep=False)
    results = smf.ols('Amount_Collected ~ Marketing_Channel_Email + Marketi
ng_Channel_Flyer + Marketing_Channel_Phone\
         + Sales_Contact_1 + Sales_Contact_2 + Sales_Contact_3 + Sales_Cont
act_4\
        + Sales_Contact_5',data=temp_data, missing="raise", hasconst=Fals
e).fit()
    consolidated_frame = pd.read_html(results.summary().tables[1].as_html
(),flavor="bs4",encoding=None,
    header=0,index_col=0)[0].reset_index()
    consolidated_frame = consolidated_frame[consolidated_frame["P>|t|"] <
0.05][["index","coef"]]
    consolidated_frame.columns=["Variable", "Coefficient (Impact)"]
    consolidated_frame["Account Type"] = acctype
    consolidated_frame = consolidated_frame.sort_values("Coefficient (Impac
t)", ascending=False,
    kind="quicksort", inplace=False,na_position="first",axis=0)
    consolidated_frame = consolidated_frame[consolidated_frame["Variable"]
!= "Intercept"]
    print(acctype)
    consolidated_summary = consolidated_summary.append(consolidated_frame)
    print(results.summary())
```

Private Facility

```
                            OLS Regression Results
==========================================================================
===
Dep. Variable:      Amount_Collected   R-squared:                       0.
427
Model:                           OLS   Adj. R-squared:                  0.
407
Method:                Least Squares   F-statistic:                     2
1.12
Date:               Tue, 27 Dec 2022   Prob (F-statistic):           1.65e
-26
Time:                       00:57:57   Log-Likelihood:                -465
0.8
No. Observations:                264   AIC:                             93
20.
Df Residuals:                    255   BIC:                             93
52.
Df Model:                          9
Covariance Type:           nonrobust
==========================================================================
================
                              coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------
----------------
Intercept               4.439e+05    8.3e+05      0.535      0.593   -1.19
e+06    2.08e+06
Marketing_Channel_Email     6.6806     17.292      0.386      0.700     -2
7.374      40.735
Marketing_Channel_Flyer     1.6661      1.125      1.482      0.140      -
0.548       3.881
Marketing_Channel_Phone     7.8827     10.196      0.773      0.440     -1
2.197      27.962
Sales_Contact_1           -88.9282     47.497     -1.872      0.062    -18
2.465       4.608
Sales_Contact_2             6.6223      0.718      9.220      0.000
5.208       8.037
Sales_Contact_3            -0.7264      0.893     -0.814      0.417      -
2.485       1.032
Sales_Contact_4            19.4954     19.966      0.976      0.330     -1
9.824      58.815
Sales_Contact_5            21.7693     13.600      1.601      0.111      -
5.013      48.552
==========================================================================
===
Omnibus:                     274.961   Durbin-Watson:                   1.
110
Prob(Omnibus):                 0.000   Jarque-Bera (JB):             9920.
407
Skew:                          4.276   Prob(JB):
0.00
Kurtosis:                     31.788   Cond. No.                        1.79
e+06
==========================================================================
===
```

```
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 1.79e+06. This might indicate that there
are
strong multicollinearity or other numerical problems.
Medium Facility
                            OLS Regression Results
===========================================================================
===
Dep. Variable:      Amount_Collected   R-squared:                        0.
685
Model:                           OLS   Adj. R-squared:                   0.
679
Method:                Least Squares   F-statistic:                      11
9.7
Date:               Tue, 27 Dec 2022   Prob (F-statistic):            3.33e-
118
Time:                       00:58:03   Log-Likelihood:                 -946
1.9
No. Observations:                504   AIC:                           1.894
e+04
Df Residuals:                    495   BIC:                           1.898
e+04
Df Model:                          9
Covariance Type:           nonrobust
===========================================================================
================
                              coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------------
----------------
Intercept                5.682e+06   2.53e+06      2.241      0.025    7.01
e+05    1.07e+07
Marketing_Channel_Email     1.1260      1.287      0.875      0.382      -
1.403       3.655
Marketing_Channel_Flyer     4.1059      0.702      5.851      0.000
2.727       5.485
Marketing_Channel_Phone     2.3077      3.014      0.766      0.444      -
3.615       8.230
Sales_Contact_1             3.1365      0.732      4.284      0.000
1.698       4.575
Sales_Contact_2             3.5778      0.305     11.717      0.000
2.978       4.178
Sales_Contact_3             2.1174      0.295      7.168      0.000
1.537       2.698
Sales_Contact_4            -7.4136      4.497     -1.649      0.100     -1
6.250       1.422
Sales_Contact_5             8.2368     10.845      0.760      0.448     -1
3.071      29.544
===========================================================================
===
Omnibus:                     103.101   Durbin-Watson:                    0.
592
Prob(Omnibus):                 0.000   Jarque-Bera (JB):               197.
```

```
732
Skew:                           1.144   Prob(JB):                     1.16e
-43
Kurtosis:                       5.045   Cond. No.                     1.44
e+07
===========================================================================
===

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 1.44e+07. This might indicate that there
are
strong multicollinearity or other numerical problems.
Large Facility
                            OLS Regression Results
===========================================================================
===
Dep. Variable:        Amount_Collected   R-squared:                      0.
585
Model:                             OLS   Adj. R-squared:                 0.
582
Method:                  Least Squares   F-statistic:                    21
2.8
Date:                 Tue, 27 Dec 2022   Prob (F-statistic):          3.51e-
252
Time:                         00:58:08   Log-Likelihood:               -250
75.
No. Observations:                 1368   AIC:                         5.017
e+04
Df Residuals:                     1359   BIC:                         5.021
e+04
Df Model:                            9
Covariance Type:             nonrobust
===========================================================================
================
                              coef    std err          t      P>|t|
[0.025      0.975]
---------------------------------------------------------------------------
----------------
Intercept                2.812e+06   9.29e+05      3.026      0.003    9.89
e+05    4.63e+06
Marketing_Channel_Email     0.7886      0.936      0.842      0.400      -
1.048       2.625
Marketing_Channel_Flyer     2.7204      0.352      7.739      0.000
2.031       3.410
Marketing_Channel_Phone    -3.5361      1.333     -2.653      0.008      -
6.151      -0.921
Sales_Contact_1            11.6731      1.246      9.368      0.000
9.229      14.117
Sales_Contact_2             4.0031      0.239     16.718      0.000
3.533       4.473
Sales_Contact_3             2.0316      0.246      8.266      0.000
1.549       2.514
Sales_Contact_4            10.6145      1.260      8.423      0.000
8.143      13.086
```

```
Sales_Contact_5            -3.6385      6.703     -0.543      0.587     -1
6.789       9.512
==============================================================================
===
Omnibus:                      334.998   Durbin-Watson:                   0.
655
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1353.
896
Skew:                           1.121   Prob(JB):                    1.01e-
294
Kurtosis:                       7.327   Cond. No.                     6.22
e+06
==============================================================================
===

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 6.22e+06. This might indicate that there
are
strong multicollinearity or other numerical problems.
Small Facility
                            OLS Regression Results
==============================================================================
===
Dep. Variable:       Amount_Collected   R-squared:                       0.
146
Model:                            OLS   Adj. R-squared:                  0.
138
Method:                 Least Squares   F-statistic:                     1
7.79
Date:                Tue, 27 Dec 2022   Prob (F-statistic):           1.17e
-24
Time:                        00:58:17   Log-Likelihood:                 -141
25.
No. Observations:                 840   AIC:                         2.827
e+04
Df Residuals:                     832   BIC:                         2.830
e+04
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
================
                              coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
----------------
Intercept               8.789e+05   2.07e+05      4.252      0.000    4.73
e+05    1.28e+06
Marketing_Channel_Email     1.8882      1.344      1.405      0.160     -
0.750       4.526
Marketing_Channel_Flyer     0.0753      0.268      0.281      0.778     -
0.450       0.600
Marketing_Channel_Phone -7.137e-07   1.68e-07     -4.252      0.000   -1.04
e-06   -3.84e-07
Sales_Contact_1            -0.9764      1.655     -0.590      0.555     -
```

```
4.224       2.271
Sales_Contact_2            0.8101      0.130      6.218      0.000
0.554       1.066
Sales_Contact_3            0.3277      0.198      1.656      0.098      -
0.061       0.716
Sales_Contact_4            0.4837      4.427      0.109      0.913      -
8.206       9.173
Sales_Contact_5           -2.6243      7.996     -0.328      0.743     -1
8.319      13.070
============================================================================
===
Omnibus:                      985.034   Durbin-Watson:                    1.
006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             97146.
151
Skew:                           5.770   Prob(JB):
0.00
Kurtosis:                      54.405   Cond. No.                        9.37
e+16
============================================================================
===

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The smallest eigenvalue is 2.3e-19. This might indicate that there are
```

6. Final Recommendations

Using the table below we can use the coefficient to see how much return we can derive from each dollar we spend, here we can clearly see that for different Account Types and different Campaigns and different Sales Contact are effective for each type of facility

Case Explanation - Small Facility

Small Facility achieved more impact and value on the dollar on average with Sales Contact 2. Despite the weakness in other marketing channels Small Facility is able to offset the gain in returns with Sales Contact 2. It may be advisable to conduct further inward education by developing a comprehensive marketing research plan to determine the factors contributes to the significant losses in the Marketing Channel Phone.

Case Explanation - Medium Facility

Case Explanation - Medium Facility Medium Facility shows decent results with Flyer Campaigns with each dollar spent and a return of four dollars on average. Sales Contact 2 is highly effective followed by Sales Contact 1 and Sales Contact 3. All other marketing strategies shows no significant impact on return on investment and further marketing research may be warranted to determine where improvements can be made or to determine whether to dissolve the other marketing channels.

Case Explanation - Large Facility

Large Facility had no comparative, with the other type of facilities, and significant impact on all of its marketing channels. There is no significant data to determine whether the size of the facility or the availability of resources is a factor for return on investment based on the marketing channel. It is reasonable to assume that there is some segmentation of our target audience based on the size and type of the facility. Additional data would be necessary to determine if there is a correlation between return on investment and the segmentation of the market based on the size and type of the facility.

In [ ]:
```
consolidated_summary
```

Out[ ]:

	Variable	Coefficient (Impact)	Account Type
5	Sales_Contact_2	6.622300e+00	Private Facility
2	Marketing_Channel_Flyer	4.105900e+00	Medium Facility
5	Sales_Contact_2	3.577800e+00	Medium Facility
4	Sales_Contact_1	3.136500e+00	Medium Facility
6	Sales_Contact_3	2.117400e+00	Medium Facility
4	Sales_Contact_1	1.167310e+01	Large Facility
7	Sales_Contact_4	1.061450e+01	Large Facility
5	Sales_Contact_2	4.003100e+00	Large Facility
2	Marketing_Channel_Flyer	2.720400e+00	Large Facility
6	Sales_Contact_3	2.031600e+00	Large Facility
3	Marketing_Channel_Phone	-3.536100e+00	Large Facility
5	Sales_Contact_2	8.101000e-01	Small Facility
3	Marketing_Channel_Phone	-7.137000e-07	Small Facility

In [ ]:
```
consolidated_summary.reset_index(inplace=True,drop=False)
consolidated_summary.drop("index",axis=1,inplace=True)
consolidated_summary["Coefficient (Impact)"] = consolidated_summary["Coeffi
cient (Impact)"].apply(lambda x: round(x,2))
#consolidated_summary.rename({"Coefficient (Impact":"Return on Investmen
t"})
consolidated_summary
```

Out[ ]:

	Variable	Coefficient (Impact)	Account Type
0	Sales_Contact_2	6.62	Private Facility
1	Marketing_Channel_Flyer	4.11	Medium Facility
2	Sales_Contact_2	3.58	Medium Facility
3	Sales_Contact_1	3.14	Medium Facility
4	Sales_Contact_3	2.12	Medium Facility
5	Sales_Contact_1	11.67	Large Facility
6	Sales_Contact_4	10.61	Large Facility
7	Sales_Contact_2	4.00	Large Facility
8	Marketing_Channel_Flyer	2.72	Large Facility
9	Sales_Contact_3	2.03	Large Facility
10	Marketing_Channel_Phone	-3.54	Large Facility
11	Sales_Contact_2	0.81	Small Facility
12	Marketing_Channel_Phone	-0.00	Small Facility

In [ ]:
```
import matplotlib
```

In [ ]:
```
consolidated_summary.columns = ["Variable", "Return on Investment", "Accoun
t Type"]
consolidated_summary["Return on Investment"] = consolidated_summary["Return
on Investment"].apply(lambda x: round(x,2))
consolidated_summary.style.background_gradient(cmap="gist_rainbow")
```

Out[ ]:

	Variable	Return on Investment	Account Type
0	Sales_Contact_2	6.62	Private Facility
1	Marketing_Channel_Flyer	4.11	Medium Facility
2	Sales_Contact_2	3.58	Medium Facility
3	Sales_Contact_1	3.14	Medium Facility
4	Sales_Contact_3	2.12	Medium Facility
5	Sales_Contact_1	11.67	Large Facility
6	Sales_Contact_4	10.61	Large Facility
7	Sales_Contact_2	4	Large Facility
8	Marketing_Channel_Flyer	2.72	Large Facility
9	Sales_Contact_3	2.03	Large Facility
10	Marketing_Channel_Phone	-3.54	Large Facility
11	Sales_Contact_2	0.81	Small Facility
12	Marketing_Channel_Phone	-0	Small Facility

In [ ]:
```
def format(x):
    return "${:.2f}".format(x)
consolidated_summary["Return on Investment"] = consolidated_summary["Return
on Investment"].apply(format)
```

In [ ]:
```
consolidated_summary.columns = ['Variable','Return on Investment','Account
Type']
consolidated_dataframe = pd.DataFrame(consolidated_summary)
consolidated_dataframe.style.background_gradient(cmap='RdYlGn')

# consolidated_summary.columns = ["Variable", "Return on Investment", "Acco
unt Type"]
# consolidated_dataframe = pd.DataFrame(consolidated_summary)
# consolidated_dataframe.style.background_gradient(cmap="YlOrRd")
# consolidated_dataframe["Return on Investment"].style.background_gradient
(cmap="gist_rainbow")
```

Out[ ]:

	Variable	Return on Investment	Account Type
0	Sales_Contact_2	$6.62	Private Facility
1	Marketing_Channel_Flyer	$4.11	Medium Facility
2	Sales_Contact_2	$3.58	Medium Facility
3	Sales_Contact_1	$3.14	Medium Facility
4	Sales_Contact_3	$2.12	Medium Facility
5	Sales_Contact_1	$11.67	Large Facility
6	Sales_Contact_4	$10.61	Large Facility
7	Sales_Contact_2	$4.00	Large Facility
8	Marketing_Channel_Flyer	$2.72	Large Facility
9	Sales_Contact_3	$2.03	Large Facility
10	Marketing_Channel_Phone	$-3.54	Large Facility
11	Sales_Contact_2	$0.81	Small Facility
12	Marketing_Channel_Phone	$-0.00	Small Facility

In [ ]:
```
consolidated_summary.to_csv("okl_consolidated_summary.csv",mode="w",encodin
g="utf-8")
open("okl_consolidated_summary.csv")
```

Out[ ]:
```
<_io.TextIOWrapper name='okl_consolidated_summary.csv' mode='r' encoding='c
p1252'>
```