Media Mix Analysis

In this project, I analyzed the digital marketing campaign data for an insurance company. I used the data visualization and quantitative analysis to check the performance of their different marketing channels and provided suggestions on their media budget allocation.

In [1]:
```
#Import Packages
import csv
import os
from os.path import expanduser

import pandas as pd
import numpy as np
from scipy import random
import seaborn as sns
import math
from math import ceil

from scipy import stats
#import statsmodels.stats.api as sms
from sklearn.linear_model import LogisticRegression
import statsmodels.discrete.discrete_model as sm
from statsmodels.stats.proportion import proportions_ztest

import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import plotly.graph_objects as go
import plotly.express as px
import matplotlib.ticker as mtick

import itertools
%matplotlib inline
```

Import and Merge the Data

In [2]:
```
#Import Data
os.chdir(r'C:\Users\runxi.wang\Desktop\Classes & Training\Homework')

df_conversion = pd.read_csv('Conversions.csv')
df_spend = pd.read_csv("TotalSpend.csv")
```

In [3]:
```
# Count the conversions for each index
conversion_counts = df_conversion.groupby('index')['id'].count()
conversion_counts.name = 'num_conversions'
conversion_counts = conversion_counts.reset_index()
```

In [4]:
```
# Merge the Spend data and the Conversion data
df_full = df_spend.merge(conversion_counts, on='index')
```

Question 1

1. If you are working with the marketing department to gain insight into this data, what 3 metrics might you calculate to assess marketing spend effectiveness? Can you represent these in a visual manner?

I calculated these three metrics to assess marketing spend effectiveness. I check the Spend Effectiveness at both Channel and Source levels.

1. Cost per Traffic
2. Conversion Rate
3. Cost per Conversion

I created several charts to visual the data.

1. I used a stacked bar chart to show the spend (amount and share) of each source.
2. I used bar charts to show the Cost per Traffic (cpt), Conversion Rate (cr) and Cost per Conversion (cpc) at channel and source levels.
3. I created an interactive bubble chart to show a directional correlation between Cost per Traffic and Conversion Rate at source level.

In [5]:
```
# Calculate the CPM, CPC and the Conversion Rate by Source
df_full['cpt'] = (df_full.totalspend / df_full.totaltraffic)
df_full['cr'] = (df_full.num_conversions / df_full.totaltraffic)
df_full['cpc'] = (df_full.totalspend / df_full.num_conversions)
```

In [6]:
```
#Calculate the share of traffic from 'none' source
traffic_total = df_full['totaltraffic'].sum()
non_traffic= df_full.iloc[8].loc['totaltraffic']/traffic_total
#Calculate the share of conversion from 'none' source
con_total = df_full['num_conversions'].sum()
non_convert= df_full.iloc[8].loc['num_conversions']/con_total
# Print the percentage of Traffic and Conversion without any marketing source
print("Share of Traffic from 'none' Source: "+"{:.2%}".format(non_traffic));
print("Share of Conversion from 'none' Source: "+"{:.2%}".format(non_convert
));
```

```
Share of Traffic from 'none' Source: 3.65%
Share of Conversion from 'none' Source: 3.58%
```

In [7]:
```
# Remove the row of source as 'none':
# since it does not have any spend and it contributes to a very small amount o
f traffic and conversion
df_full_new = df_full.loc[~(df_full['source']=='none')]
# Drop the first unnamed column
df_full_new = df_full_new.drop(df_full_new.columns[0], axis=1)
```

In [8]:
```
# Create a dataframe for data visualization
# Add columns with clear data format to the table
df_full_display = df_full_new
df_full_display['spend_format'] = pd.Series(['${0:,.0f}'.format(val * 1) for v
al in df_full_display['totalspend']], index = df_full_display.index)
df_full_display['cpt_format'] = pd.Series(['${0:,.2f}'.format(val * 1) for val
in df_full_display['cpt']], index = df_full_display.index)
df_full_display['cpc_format'] = pd.Series(['${0:,.2f}'.format(val * 1) for val
in df_full_display['cpc']], index = df_full_display.index)
df_full_display['cr%'] = pd.Series(['{0:.1f}%'.format(val * 100) for val in df
_full_display['cr']], index = df_full_display.index)

#Create a column with the label combining channel and source
df_full_display['channel_source'] = df_full_display[['channel', 'source']].agg
('_'.join, axis=1)
```

In [9]:
```
# Print the Dataframe, getting a high level understanding of the dataset
df_full_display
```

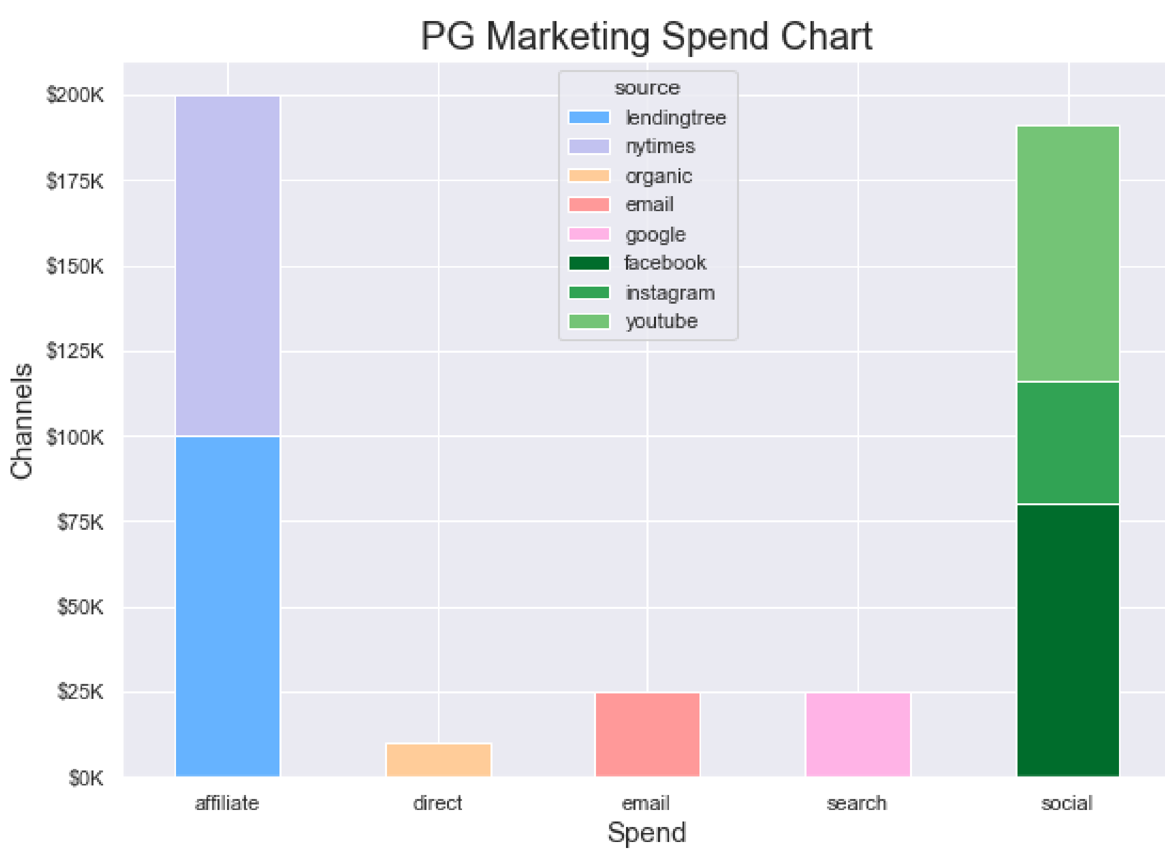
Out[9]:

	source	channel	totalspend	totaltraffic	index	num_conversions	cpt	cr	
0	google	search	25000.0	15000	1	1186	1.666667	0.079067	2
1	facebook	social	80000.0	10000	2	521	8.000000	0.052100	15
2	instagram	social	36000.0	15000	3	1161	2.400000	0.077400	3
3	organic	direct	10000.0	30000	4	2189	0.333333	0.072967	[illegible]
4	youtube	social	75000.0	5000	5	367	15.000000	0.073400	20
5	nytimes	affiliate	100000.0	25000	6	1905	4.000000	0.076200	5
6	email	email	25000.0	12000	7	854	2.083333	0.071167	2
7	lendingtree	affiliate	100000.0	20000	8	1771	5.000000	0.088550	5

In [10]:
```
# Create a Stacked bar chart to show the spend media mix
# Prepare the data: create a pivot for the chart
df_spend= df_spend.loc[~(df_spend['channel']=='none')]
pivot_df = pd.pivot_table(df_spend, index = 'channel', values = 'totalspend',
columns = 'source', aggfunc = 'max')
```

In [11]:
```
# Create a Stacked bar chart to show the spend media mix
# Set the font scale, the colors for the bars, figure size, etc.
def thousands(x, pos):
    return '${:,.0f}K'.format(x*1e-3)

fmtr = mtick.FuncFormatter(thousands)

sns.set(font_scale=1)
colors = ['#66b3ff','#c2c2f0','#ffcc99', '#ff9999','#ffb3e6', "#006D2C", "#31A
354","#74C476", '#99ff99']
pivot_df.loc[:,['lendingtree', 'nytimes', 'organic', 'email', 'google', 'faceb
ook', 'instagram','youtube']].plot.bar(
    stacked=True,
    color=colors,
    figsize=(10,7)).yaxis.set_major_formatter(fmtr)

# Set a chart title
plt.title('Marketing Spend Chart', fontsize = 20)
plt.ylabel("Channels", fontsize = 15)
plt.xlabel("Spend", fontsize=15)
plt.xticks(rotation=0)
#plt.legend(loc="upper right")
#sort the bars
```

Out[11]:
```
(array([0, 1, 2, 3, 4]), <a list of 5 Text xticklabel objects>)
```



From stacked bar chart, we can get a general idea about how much investment each channel / source got. Social and Affiliate were the two channels got the most spend, much more than search and email. Two vendors, Lending Tree and NY Times split the Affiliate spend equally, each of them getting 100k. The social campaign spent on Facebook the most (80k), followed by YouTube (75K), and Instagram the least (35k).

Create Bar Charts about Marketing Effectiveness at Channel Level

In [12]:
```
# Create a dataframe for data visualization at channel level
df_channel = df_full.groupby(['channel']).sum()
# Drop the unnamed column and the other columns won't be used
df_channel = df_channel.drop(df_channel.columns[0], axis=1)
```

In [13]:
```
# Drop the columns we need to recalculate
df_channel = df_channel.drop(columns=['index', 'cpt','cr','cpc'])
df_channel = df_channel.reset_index()
```

In [14]:
```
# Remove the row of source as 'none': since it does not have any spend and it
 contributes to a very small amount of traffic and conversion
df_channel = df_channel.loc[~(df_channel['channel']=='none')]
```

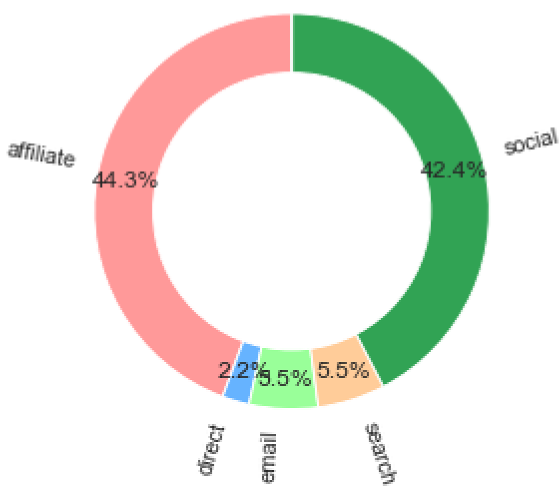
In [15]:
```
# Calculate the CPM, CPC and the Conversion Rate by Channel
df_channel['cpt'] = (df_channel.totalspend / df_channel.totaltraffic)
df_channel['cr'] = (df_channel.num_conversions / df_channel.totaltraffic)
df_channel['cpc'] = (df_channel.totalspend /df_channel.num_conversions)
```

In [16]:
```
df_channel
```

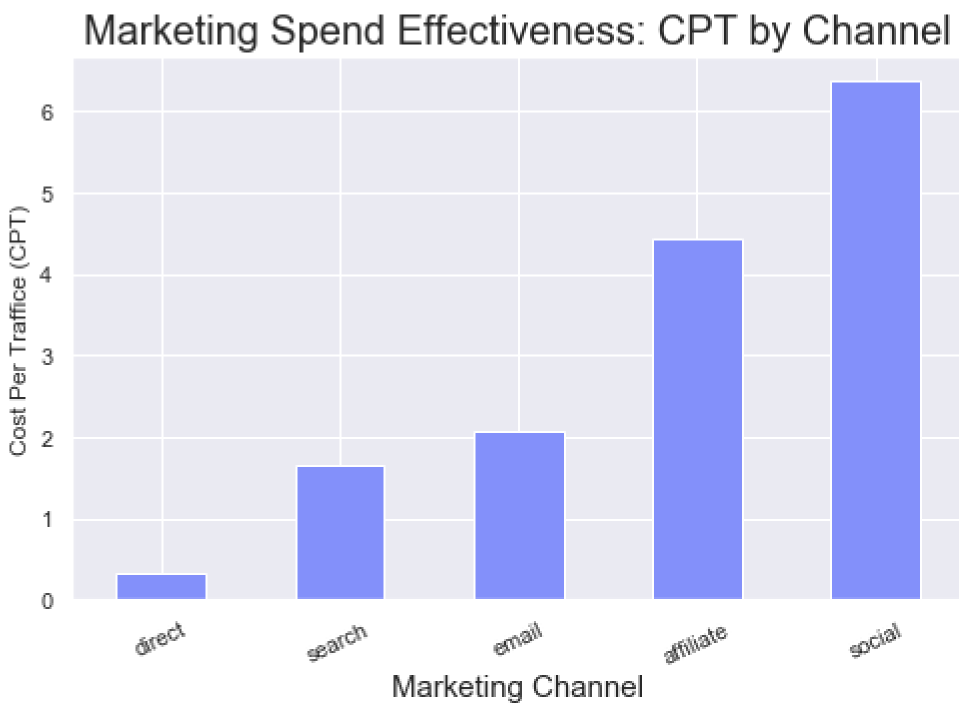
Out[16]:

	channel	totalspend	totaltraffic	num_conversions	cpt	cr	cpc
0	affiliate	200000.0	45000	3676	4.444444	0.081689	54.406964
1	direct	10000.0	30000	2189	0.333333	0.072967	4.568296
2	email	25000.0	12000	854	2.083333	0.071167	29.274005
4	search	25000.0	15000	1186	1.666667	0.079067	21.079258
5	social	191000.0	30000	2049	6.366667	0.068300	93.216203

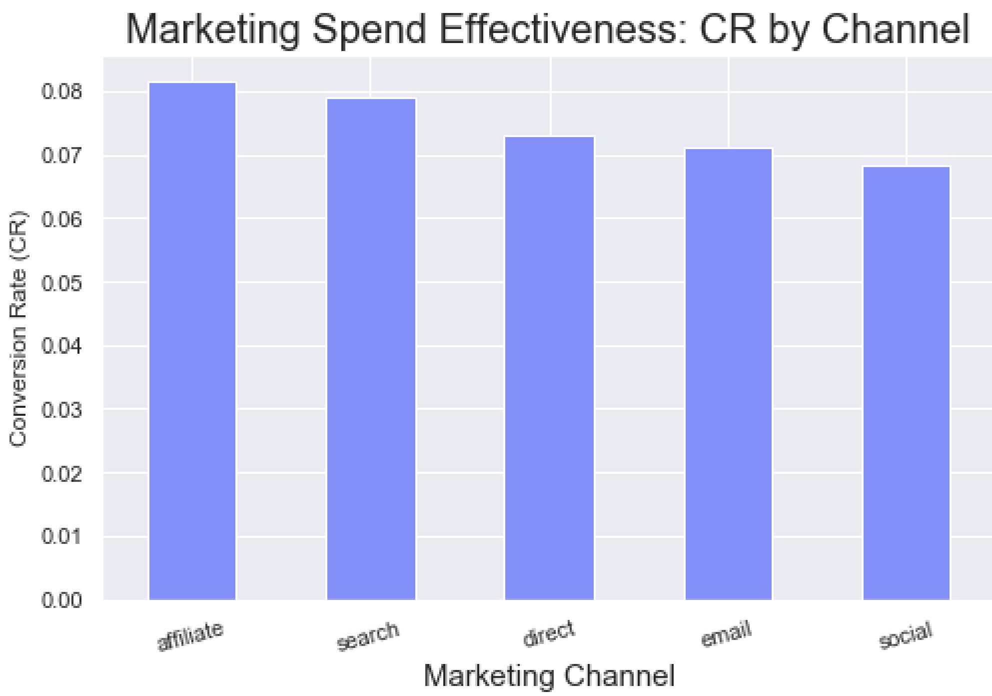
In [17]:
```
# Create a Pie Chart to show the breakdown of marketing spend by channel
labels = ['affiliate','direct','email','search','social']
# Set the colors
colors = ['#ff9999','#66b3ff','#99ff99','#ffcc99',"#31A354"]
#plt.rcParams["figure.figsize"] = (4,4)
fig1, ax1 = plt.subplots()
ax1.pie(df_channel.totalspend, colors = colors,  labels=labels, autopct='%1.1f%%',
        startangle=90, pctdistance=0.85, rotatelabels=True)

# Draw the circle
centre_circle = plt.Circle((0,0),0.70, fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')

plt.show()
```



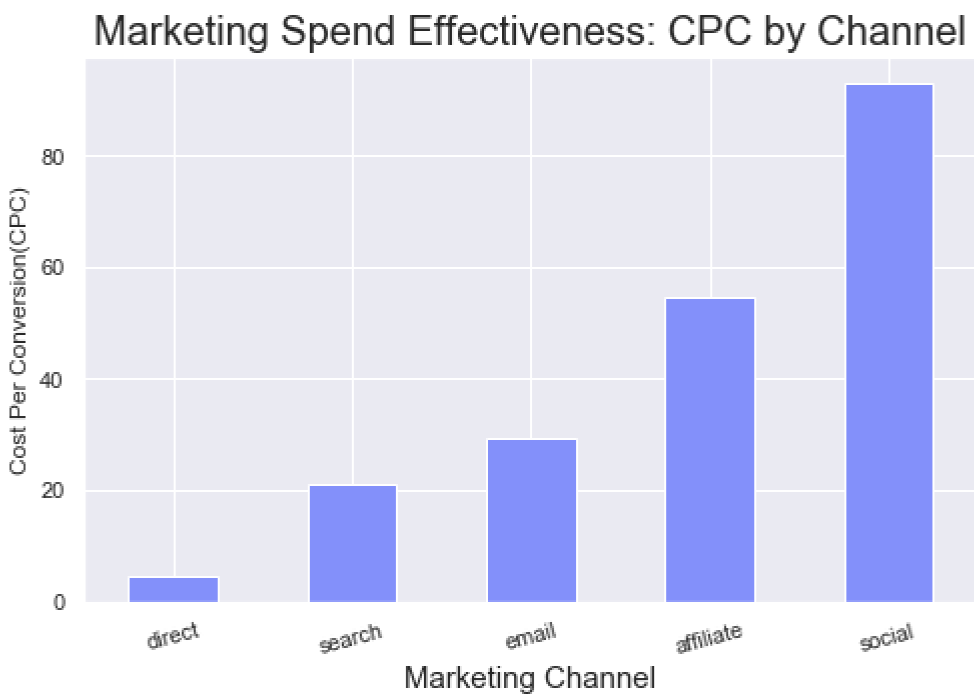
In [18]:
```
# Create Bar Chart at Channel Level
# Get a big picture: we would like to know which channel worked the most or le
ast effectively
# Sort the channels by Cost per Traffic, from lowest to highest
df_channel = df_channel.sort_values('cpt')

# Plot the bar chart for Cost per Traffic: set the colors, figure size, etc.
df_channel.set_index(['channel'])[['cpt']].plot.bar(stacked=True, color="#8390
FA",  figsize=(8,5), legend=None)

# Set a chart title, x-axis name, y-axis name, etc.
plt.title("Marketing Spend Effectiveness: CPT by Channel", fontsize=20)
plt.xlabel("Marketing Channel", fontsize=15)
plt.xticks(rotation=25)
plt.ylabel("Cost Per Traffice (CPT)")

plt.show()
```



In [19]:
```
# Sort the channels by Conversion Rate descendingly
df_channel = df_channel.sort_values('cr', ascending=False)

# Plot the bar chart for Conversion Rate: set the colors, figure size, etc.
df_channel.set_index(['channel'])[['cr']].plot.bar(stacked=True,  color="#8390
FA", figsize=(8,5), legend=None)

plt.title("Marketing Spend Effectiveness: CR by Channel", fontsize=20)
plt.xlabel("Marketing Channel", fontsize=15)
plt.xticks(rotation=15)
plt.ylabel("Conversion Rate (CR)")

plt.show()
```



In [20]:
```
# Sort the channels by Cost per Conversion ascendingly.
df_channel = df_channel.sort_values('cpc')

# Plot the bar chart for Cost per Conversion: set the colors, figure size, etc.
df_channel.set_index(['channel'])[['cpc']].plot.bar( stacked=True,  color="#8390FA",  figsize=(8,5),  legend=None)

plt.title("Marketing Spend Effectiveness: CPC by Channel", fontsize=20)
plt.xlabel("Marketing Channel", fontsize=15)
plt.xticks(rotation=15)
plt.ylabel("Cost Per Conversion(CPC)")

plt.show()
```



Despite Direct, Search was the cheapest channel in getting traffic and conversion, with a cpt of 1.67 and cpc of 21, followed by Email.

Affiliate and Social were the two most expensive channels in both attracting traffic and conversion. However, Affiliate saw the highest CR of 8.2% while Social got the lowest (6.8%).

Direct saw the lowest cost per traffic and conversion, but it was supposed to be "organic," which means it should not have any spend. Probably The Company has a different definition of Direct as a channel and Organic as a channel. I'll need more context to decide if this was a data discrepancy. As a result, I kept the Direct data in the charts and later analysis.

Note about Direct Traffic

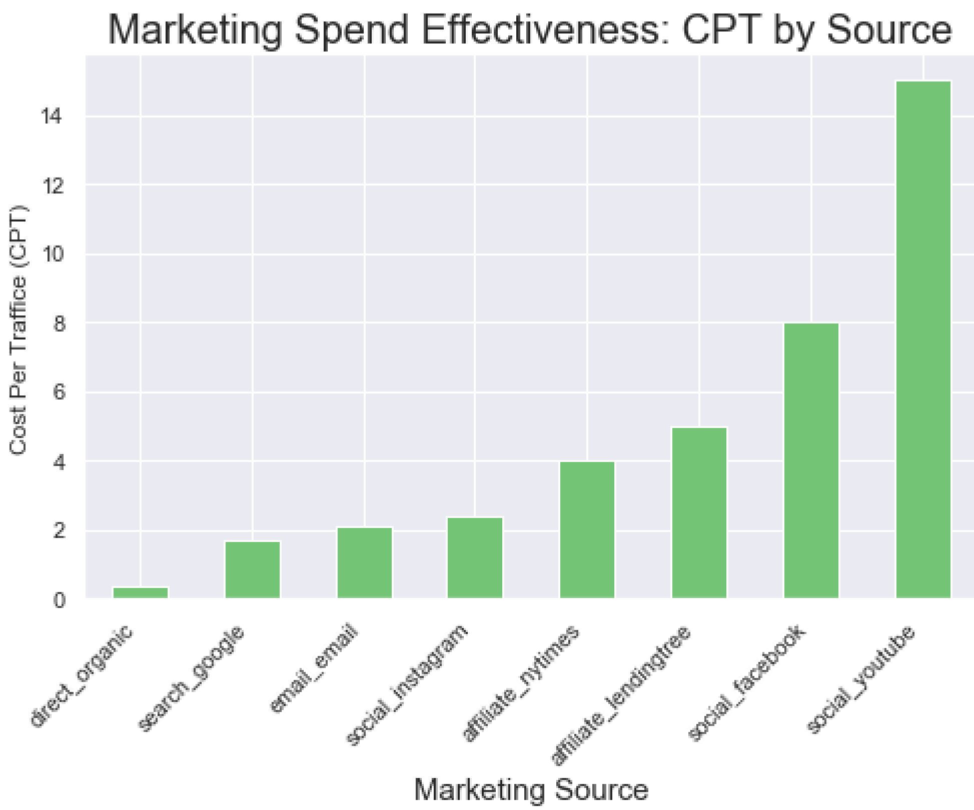
- Any source without attribution, notably anything without a referrer
- Often typing a URL directly or a bookmark.

However, it could also be:

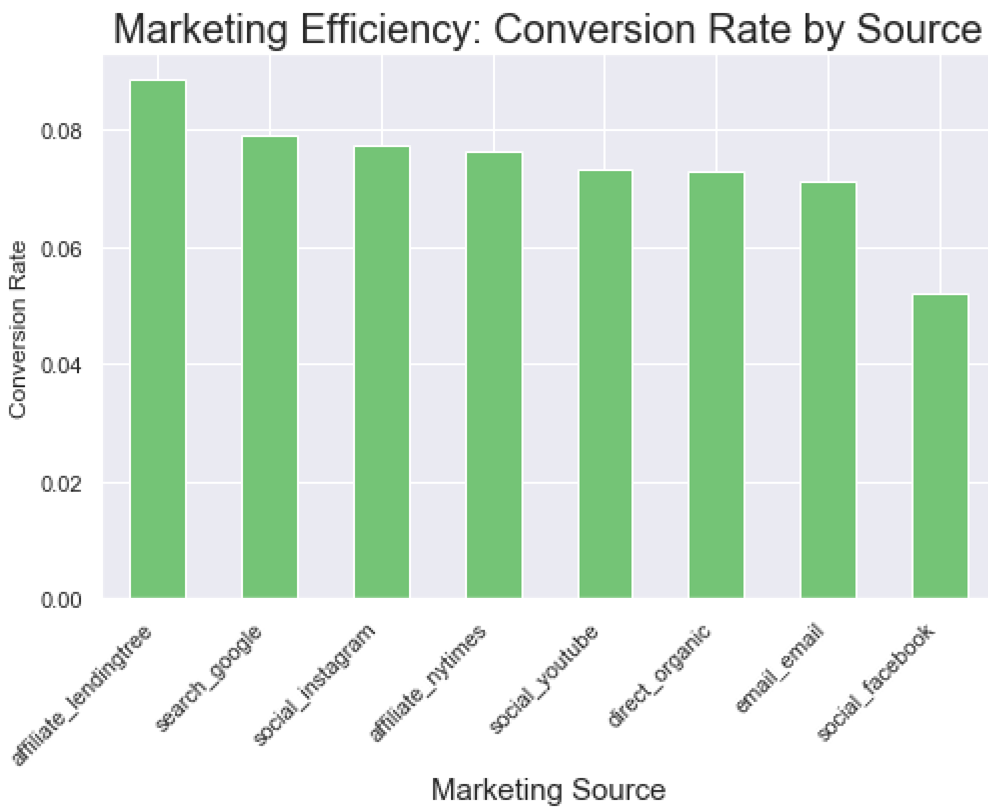
- Link in a PDF
- Link in an App
- Link in Text
- Link in an email program (not webmail)
- Any untagged link in something other than a website

Create Bar Charts about Marketing Effectiveness at Source Level

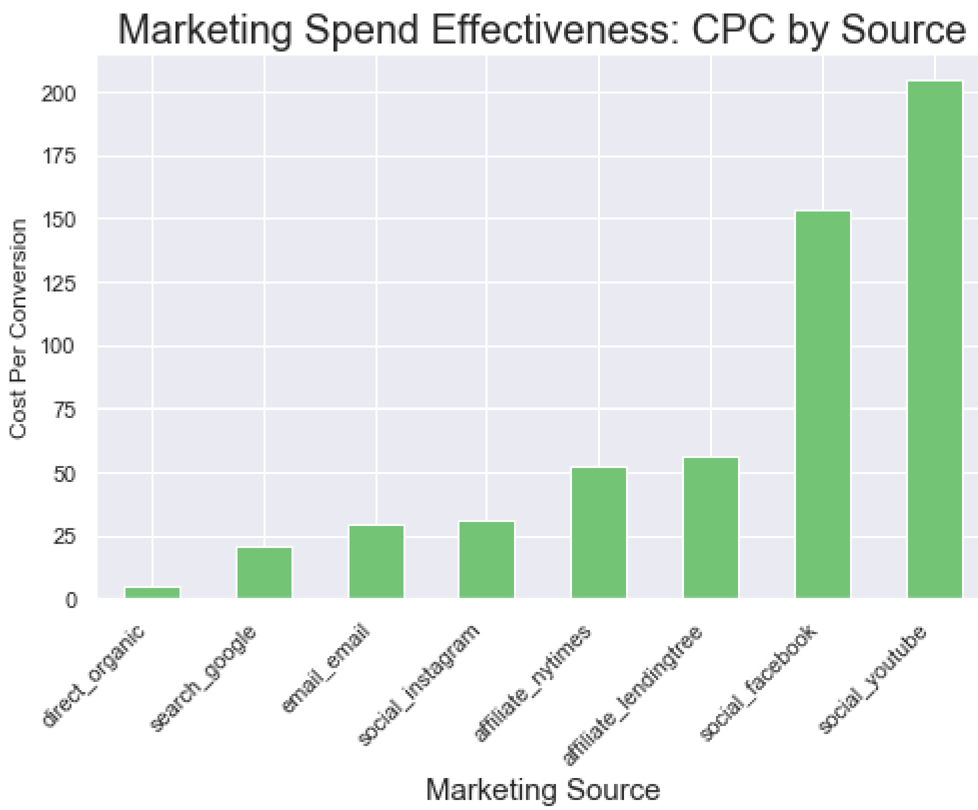
In [21]:
```
# Create Bar Chart at Source Level
# Sort the sources by Cost per Traffic, from lowest to highest
df_full_display = df_full_display.sort_values('cpt')

# Plot the bar chart for Conversion Rate: set the colors, figure size, etc.
df_full_display.set_index(['channel_source'])[['cpt']].plot.bar(stacked=True,
color="#74C476", figsize=(8,5), legend=None)

# Set a chart title, x-axis name, y-axis name, etc.
plt.title("Marketing Spend Effectiveness: CPT by Source", fontsize=20)
plt.xlabel("Marketing Source", fontsize=15)
plt.xticks(rotation=45, horizontalalignment='right')
plt.ylabel("Cost Per Traffice (CPT)")

plt.show()
```



In [22]:
```
# Sort the sources by Conversion Rate descendingly
df_full_display = df_full_display.sort_values('cr', ascending=False)

# Plot the bar chart for Conversion Rate: set the colors, figure size, etc.
df_full_display.set_index(['channel_source'])[['cr']].plot.bar(stacked=True,
color="#74C476",  figsize=(8,5), legend=None)

plt.title("Marketing Efficiency: Conversion Rate by Source", fontsize=20)
plt.xlabel("Marketing Source", fontsize=15)
plt.xticks(rotation=45, horizontalalignment='right')
plt.ylabel("Conversion Rate")

plt.show()
```



In [23]:
```
# Sort the sources by Cost per Conversion ascendingly
df_full_display = df_full_display.sort_values('cpc')

# Plot the bar chart for Cost per Conversion: set the colors, figure size, et
c.
df_full_display.set_index(['channel_source'])[['cpc']].plot.bar(stacked=True,
color="#74C476", figsize=(8,5),legend=None)

plt.title("Marketing Spend Effectiveness: CPC by Source", fontsize=20)
plt.xlabel("Marketing Source", fontsize=15)
plt.xticks(rotation=45, horizontalalignment='right')
plt.ylabel("Cost Per Conversion")

plt.show()
```



Despite Organic (Direct) Google (Search) cost the least to get the traffic and conversions, followed by Email and Instagram.

Lending Tree (Affiliate) was the vendor saw the highest conversion rate, while another Affiliate source, NY Times was at the fourth place.

Social was the channel with highest cost per traffic and conversion, but lowest conversion rate. However, different social media platforms performed variously. Instagram saw a low cost per traffic and conversion, while a relatively high conversion rate (ranked only after Lending Tree and Google). From this perspective, Instagram was more effective and efficient marketing source than Facebook and YouTube.

Question 2

1. Taking spend efficiency out of the equation for now, what channels and/or sources are most likely to lead to a conversion? What channels and/or sources are least likely to lead to a conversion? You should use a model or method to statistically calculate the answers to these questions, look for significant differences and interpret them for your stakeholders. The goal is data objectivity. (Hint: you are trying to infer conversion rate which is a percent aka a binomial variable).

Question 2. Solution 1 - Statistical Significant Difference Test (z-test)

The data visual in Question 1 section shows that Lending Tree (Affiliate) was the most effective in earning conversions, with a conversion rate of 8.86%, while Facebook was the source least likely to lead to a conversion, with a conversion rate of 5.21%. This was a piece of directional insight. To check if there was a statistically significant difference among the marketing effectiveness of these channels and vendors, I designed to use the z-test to check the statistical significance.

z-test at channel level

In [24]:

```
df_channel_con = df_channel[['channel','totaltraffic','num_conversions','cr']]
.sort_values('totaltraffic')
df_channel_con
```

Out[24]:

	channel	totaltraffic	num_conversions	cr
2	email	12000	854	0.071167
4	search	15000	1186	0.079067
1	direct	30000	2189	0.072967
5	social	30000	2049	0.068300
0	affiliate	45000	3676	0.081689

In [25]:
```
# Null Hypothesis: The difference in proportions is = 0.  HA: the difference i
n proportions is ≠ 0.
for x in range(df_channel_con.shape[0]):
    for y in range(x+1, df_channel_con.shape[0]):

        Ho = 0
        # Number of customers converted from each source
        conv_users = np.array([df_channel_con.iloc[x].loc['num_conversions'],
df_channel_con.iloc[y].loc['num_conversions']])
        # Total number of traffic from each source
        t_users = np.array([df_channel_con.iloc[x].loc['totaltraffic'], df_cha
nnel_con.iloc[y].loc['totaltraffic']])
        # Two-sided hypothesis test
        z_stat, p_val = proportions_ztest(conv_users, t_users, value = Ho, alt
ernative='two-sided', prop_var=False)
        # Test Statistic
        print('Test Statistic:','{0:0.4f}'.format(z_stat))
        # P-value
        print('P-value :','{0:0.4f}'.format(p_val))
        # Print the results
        alpha = 0.05
        if p_val < alpha:
            print("Statistically significant, reject null hypothesis that popu
lation proportions are equal between "+df_channel_con.iloc[x].loc['channel']+"
and "+df_channel_con.iloc[y].loc['channel'])
        else:
            print("Not significant, fail to reject the null hypothesis that po
pulation proportions are equal between "+df_channel_con.iloc[x].loc['channel']
+" and "+df_channel_con.iloc[y].loc['channel'])
```

```
Test Statistic: -2.4407
P-value : 0.0147
Statistically significant, reject null hypothesis that population proportions
are equal between email and search
Test Statistic: -0.6428
P-value : 0.5203
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between email and direct
Test Statistic: 1.0463
P-value : 0.2954
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between email and social
Test Statistic: -3.7865
P-value : 0.0002
Statistically significant, reject null hypothesis that population proportions
are equal between email and affiliate
Test Statistic: 2.3159
P-value : 0.0206
Statistically significant, reject null hypothesis that population proportions
are equal between search and direct
Test Statistic: 4.1682
P-value : 0.0000
Statistically significant, reject null hypothesis that population proportions
are equal between search and social
Test Statistic: -1.0192
P-value : 0.3081
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between search and affiliate
Test Statistic: 2.2308
P-value : 0.0257
Statistically significant, reject null hypothesis that population proportions
are equal between direct and social
Test Statistic: -4.3585
P-value : 0.0000
Statistically significant, reject null hypothesis that population proportions
are equal between direct and affiliate
Test Statistic: -6.7650
P-value : 0.0000
Statistically significant, reject null hypothesis that population proportions
are equal between social and affiliate
```

z-test at source level

In [26]:
```
df_full_con = df_full_display[['index','source','totaltraffic','num_conversions','cr%']].sort_values('index')
df_full_con
```

Out[26]:

	index	source	totaltraffic	num_conversions	cr%
0	1	google	15000	1186	7.9%
1	2	facebook	10000	521	5.2%
2	3	instagram	15000	1161	7.7%
3	4	organic	30000	2189	7.3%
4	5	youtube	5000	367	7.3%
5	6	nytimes	25000	1905	7.6%
6	7	email	12000	854	7.1%
7	8	lendingtree	20000	1771	8.9%

In [27]:
```
# Null Hypothesis: The difference in proportions is = 0.  HA: the difference in proportions is ≠ 0.
for xa in range(df_full_con.shape[0]):
    for y in range(x+1, df_full_con.shape[0]):

        Ho = 0
        # Number of customers converted from each source
        conv_users = np.array([df_full_con.iloc[x].loc['num_conversions'], df_full_con.iloc[y].loc['num_conversions']])
        # Total number of traffic from each source
        t_users = np.array([df_full_con.iloc[x].loc['totaltraffic'], df_full_con.iloc[y].loc['totaltraffic']])
        # Two-sided hypothesis test
        z_stat, p_val = proportions_ztest(conv_users, t_users, value = Ho, alternative='two-sided', prop_var=False)
        # Test Statistic
        print('Test Statistic:','{0:0.4f}'.format(z_stat))
        # P-value
        print('P-value :','{0:0.4f}'.format(p_val))
        # Print the results
        alpha = 0.05
        if p_val < alpha:
            print("Statistically significant, reject null hypothesis that population proportions are equal between "+df_full_con.iloc[x].loc['source']+" and "+df_full_con.iloc[y].loc['source'])
        else:
            print("Not significant, fail to reject the null hypothesis that population proportions are equal between "+df_full_con.iloc[x].loc['source']+" and "+df_full_con.iloc[y].loc['source'])
```

```
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
```

```
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
Test Statistic: -0.6831
P-value : 0.4945
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and nytimes
Test Statistic: 0.5139
P-value : 0.6073
Not significant, fail to reject the null hypothesis that population proportio
ns are equal between youtube and email
Test Statistic: -3.4263
P-value : 0.0006
Statistically significant, reject null hypothesis that population proportions
are equal between youtube and lendingtree
```

Question 3. Solution 2 - A Logistic Regression Approach

The goal of this solution was using a statistical model to check if there was a significant difference in conversion rate among different marketing channels and sources. Logistic Regression should be used here since we want to predict 1 of 2 possible outcomes: whether a user will convert or not (predicts a probability between 0 and 1) depending on the channel or source. Although using the Logistic Regression Approach was a little bit over overcomplicated here.

In [28]:

```
# Before we run the Logistic Regression model, we need to generate the proper
 dataset
df_full_non=df_full[['index','source','channel','totaltraffic','num_conversion
s']].sort_values('index')
# Calcuate the number of traffic without conversions
df_full_non['not_converted']=df_full_non['totaltraffic'] - df_full_non['num_co
nversions']
df_full_non
```

Out[28]:

	index	source	channel	totaltraffic	num_conversions	not_converted
0	1	google	search	15000	1186	13814
1	2	facebook	social	10000	521	9479
2	3	instagram	social	15000	1161	13839
3	4	organic	direct	30000	2189	27811
4	5	youtube	social	5000	367	4633
5	6	nytimes	affiliate	25000	1905	23095
6	7	email	email	12000	854	11146
7	8	lendingtree	affiliate	20000	1771	18229
8	9	none	none	5000	370	4630

In [29]:

```
dfs = []
for i, r in df_full_non.iterrows():
    df = pd.DataFrame({'source' :  np.repeat(r['source'] ,r['not_converted']),
                       'channel' :  np.repeat(r['channel'] ,r['not_converted'
]),
                       'index' :  np.repeat(r['index'] ,r['not_converted']),
                       'id' : np.repeat('0' ,r['not_converted'])})
    dfs.append(df)
df_non = pd.concat(dfs)
```

In [30]:

```
# Append the dataframe of traffic without conversion to the conversion datafra
me
df_log= df_conversion.drop(df_conversion.columns[0], axis=1)
df_log= df_log.append(df_non)
df_log = df_log.loc[~(df_log['source']=='none')]
df_log['converted'] = np.where(df_log['id']=='0', 0, 1)
df_log.head()
```

Out[30]:

	source	channel	index	id	converted
0	organic	direct	4	1	1
2	google	search	1	3	1
3	lendingtree	affiliate	8	4	1
4	lendingtree	affiliate	8	5	1
5	lendingtree	affiliate	8	6	1

Solution 3. Logistic Regression at Channel level

In [31]:
```
# We only need the dummy variables created from source and channel, and the de
pendent variable: converted
# Created the dummy variables from source and channel
Source = pd.get_dummies(df_log['source'],drop_first=False)
Channel = pd.get_dummies(df_log['channel'],drop_first=False)
```

In [32]:
```
# Append the dummy variables to the dataframe for modeling
df_log_channel = pd.concat([df_log,Channel],axis=1)
# Add the intercept
df_log_channel['intercept'] = 1
# Drop the variables we don't need in the regression
df_log_channel.drop(['source','channel','index','id'],axis=1,inplace=True)
```

In [33]:
```
# Check the first few rows of the dataset
df_log_channel.head()
```

Out[33]:

	converted	affiliate	direct	email	search	social	intercept
0	1	0	1	0	0	0	1
2	1	0	0	0	1	0	1
3	1	1	0	0	0	0	1
4	1	1	0	0	0	0	1
5	1	1	0	0	0	0	1

In [34]:
```
# Run the logistic model and print the result at the channel level
# Check if Affiliate's conversion rate is significantly higher than other chan
nels
logit_mod = sm.Logit(df_log_channel['converted'], df_log_channel[['intercept',
'direct','email','search','social']])
results = logit_mod.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.267200
         Iterations 7
```

Out[34]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	132000
Model:	Logit	**Df Residuals:**	131995
Method:	MLE	**Df Model:**	4
Date:	Tue, 16 Feb 2021	**Pseudo R-squ.:**	0.0007906
Time:	13:58:23	**Log-Likelihood:**	-35270.
converged:	True	**LL-Null:**	-35298.
		LLR p-value:	2.191e-11

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-2.4196	0.017	-140.582	0.000	-2.453	-2.386
direct	-0.1224	0.028	-4.356	0.000	-0.177	-0.067
email	-0.1493	0.039	-3.783	0.000	-0.227	-0.072
search	-0.0355	0.035	-1.019	0.308	-0.104	0.033
social	-0.1935	0.029	-6.756	0.000	-0.250	-0.137

In [35]:
```
# Logistic interpretation differs from linear: we need to exponentiate the coe
fficients
# The resulting value is the multiplicative change in the odds
# If the coefficient is negative we need to take the reciprocal
# Flip 'increase' to 'decrease' in the interpretation so it's easier to unders
tand
print("Direct is "+ str(round(1/np.exp(-0.1207),2)) + " times LESS likely to c
onvert a user than Affiliate.");
print("Email is "+ str(round(1/np.exp(-0.1476),2)) + " times LESS likely to co
nvert a user than Affiliate.");
print("Social is "+ str(round(1/np.exp(-0.1918),2)) + " times LESS likely to c
onvert a user than Affiliate.");
print("Search saw no statistical significant difference in the efficiency of a
ttracting conversions compared to Affiliate");
```

```
Direct is 1.13 times LESS likely to convert a user than Affiliate.
Email is 1.16 times LESS likely to convert a user than Affiliate.
Social is 1.21 times LESS likely to convert a user than Affiliate.
Search saw no statistical significant difference in the efficiency of attract
ing conversions compared to Affiliate
```

Solution 3. Logistic Regression at Source Level - Affiliate

In [36]:

```
# Create the new dataframe for the modeling at source level, Affiliate only
df_log_affiliate = df_log.loc[(df_log['source'] == 'lendingtree') | (df_log['s
ource'] == 'nytimes')]
df_log_affiliate[['lendingtree', 'nytimes']] = pd.get_dummies(df_log_affiliate
['source'])
df_log_affiliate['intercept'] = 1
df_log_affiliate.drop(['source', 'channel','index','id' ], axis=1, inplace=Tru
e)
df_log_affiliate.head()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py:3140: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  self[k1] = value[k2]
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py:3697: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  errors=errors)
```

Out[36]:

	converted	lendingtree	nytimes	intercept
3	1	1	0	1
4	1	1	0	1
5	1	1	0	1
12	1	1	0	1
15	1	0	1	1

In [37]:
```
logit_mod = sm.Logit(df_log_affiliate['converted'], df_log_affiliate[['intercept','nytimes']])
results = logit_mod.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.282625
         Iterations 6
```

Out[37]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	45000
Model:	Logit	**Df Residuals:**	44998
Method:	MLE	**Df Model:**	1
Date:	Tue, 16 Feb 2021	**Pseudo R-squ.:**	0.0008836
Time:	13:58:24	**Log-Likelihood:**	-12718.
converged:	True	**LL-Null:**	-12729.
		LLR p-value:	2.106e-06

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-2.3315	0.025	-93.671	0.000	-2.380	-2.283
nytimes	-0.1637	0.034	-4.749	0.000	-0.231	-0.096

In [38]:
```
print("New York Times is "+ str(round(1/np.exp(-0.1598),2)) + " times LESS likely to convert a user than Lending Tree.");
```

```
New York Times is 1.17 times LESS likely to convert a user than Lending Tree.
```

Solution 3. Logistic Regression at Source Level - Social

In [39]:

```
# Create the new dataframe for the modeling at source level, Social only
df_log_social = df_log.loc[(df_log['source'] == 'facebook') | (df_log['source'
] == 'youtube')|(df_log['source'] == 'instagram')]
df_log_social[['facebook', 'youtube','instagram']] = pd.get_dummies(df_log_soc
ial['source'])
df_log_social['intercept'] = 1
df_log_social.drop(['source', 'channel','index','id' ], axis=1, inplace=True)
df_log_social.head()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
```

Out[39]:

	converted	facebook	youtube	instagram	intercept
8	1	0	1	0	1
22	1	0	0	1	1
34	1	0	1	0	1
35	1	0	1	0	1
39	1	1	0	0	1

In [40]:
```
logit_mod = sm.Logit(df_log_social['converted'], df_log_social[['intercept','i
nstagram','youtube']])
results = logit_mod.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.248128
         Iterations 7
```

Out[40]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	30000
Model:	Logit	**Df Residuals:**	29997
Method:	MLE	**Df Model:**	2
Date:	Tue, 16 Feb 2021	**Pseudo R-squ.:**	0.004377
Time:	13:58:24	**Log-Likelihood:**	-7443.9
converged:	True	**LL-Null:**	-7476.6
		LLR p-value:	6.136e-15

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-2.9011	0.045	-64.470	0.000	-2.989	-2.813
instagram	0.3655	0.070	5.187	0.000	0.227	0.504
youtube	0.4229	0.054	7.775	0.000	0.316	0.529

In [41]:

```
logit_mod = sm.Logit(df_log_social['converted'], df_log_social[['intercept','f
acebook','youtube']])
results = logit_mod.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.248128
         Iterations 7
```

Out[41]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	30000
Model:	Logit	**Df Residuals:**	29997
Method:	MLE	**Df Model:**	2
Date:	Tue, 16 Feb 2021	**Pseudo R-squ.:**	0.004377
Time:	13:58:24	**Log-Likelihood:**	-7443.9
converged:	True	**LL-Null:**	-7476.6
		LLR p-value:	6.136e-15

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-2.5356	0.054	-46.758	0.000	-2.642	-2.429
facebook	-0.3655	0.070	-5.187	0.000	-0.504	-0.227
youtube	0.0574	0.062	0.922	0.357	-0.065	0.179

In [42]:

```
print("Instagram is "+ str(round(np.exp(0.3655),2)) + " times MORE likely to c
onvert a user than Facebook.");
print("YouTube Times is "+ str(round(np.exp(0.4229),2)) + " times MORE likely
 to convert a user than Facebook.");
print("Instagram saw no statistical significant difference in the efficiency o
f attracting conversions compared to Youtube");
```

```
Instagram is 1.44 times MORE likely to convert a user than Facebook.
YouTube Times is 1.53 times MORE likely to convert a user than Facebook.
Instagram saw no statistical significant difference in the efficiency of attr
acting conversions compared to Youtube
```

Question 2. Solution 3 - uses beta-binomial Bayesian model

In [43]:
```
# Solution 3: uses beta-binomial Bayesian model
# and then Monte Carlo simulation of the best performing campaign

# alphas is the # successes (conversions) of each campaign
alphas = df_full.num_conversions + 1

# betas is the # failures (non-conversion) of each campaign
betas = df_full.totaltraffic - df_full.num_conversions + 1

# Monte Carlo simulation
trials = 10000
simulated_samples = np.array([random.beta(alpha, beta, size=trials) for alpha,
beta in zip(alphas, betas)])

best_idxs = np.argmax(simulated_samples, axis=0)
```

In [44]:
```
# We are very confident that lendtree is best since it's the maximum in ~100%
 of simulations
print(np.mean(best_idxs == 7))
```

```
0.9988
```

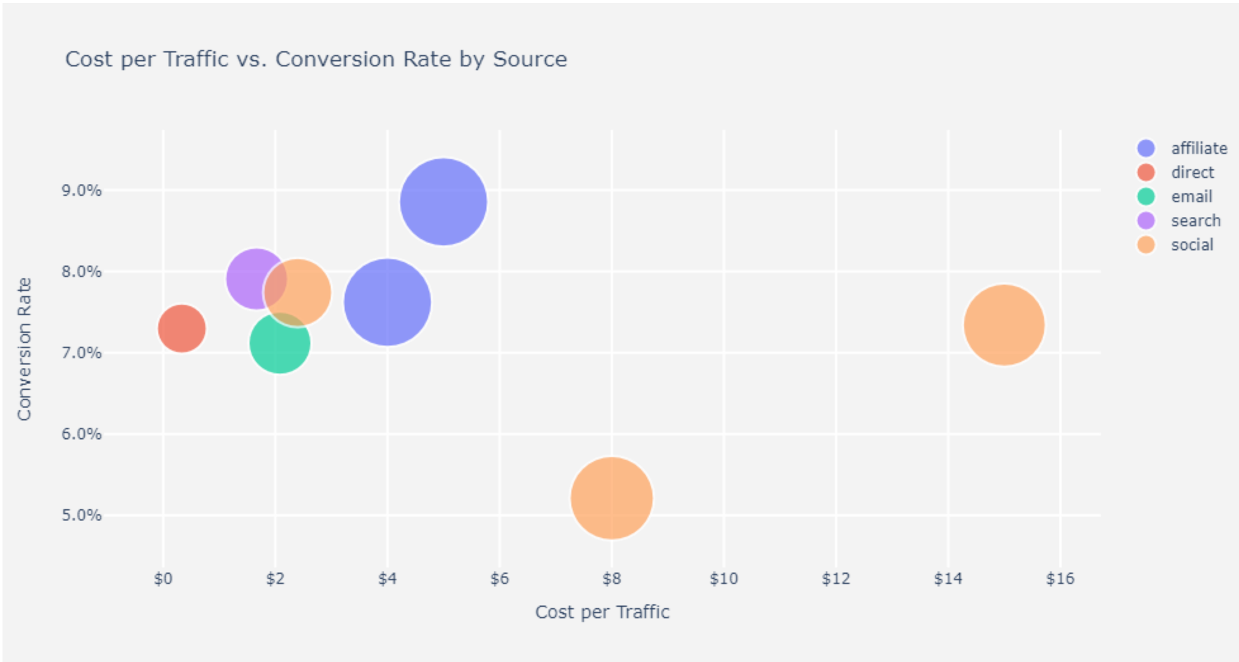
Question 3

1. What advice would you give the marketing team on adjusting their spend patterns and why? How would you explain or handle artifacts such as the performance from none (represents no channel attribution can be detected)? Be sure to take into account conversion rate performance as well as spend efficiency.

Bubble Chart: A Directional and Interactive View of correlation between CPT and CR

The marketing sources at the upper left side came with a relatively high conversion rate and low cost per traffic. Organic, Google, Email and Instagram showed a very close performance. Facebook's conversion rate was much lower than other sources and YouTube's Cost per Traffic was much higher than other source. Generally speaking, the chart also shows a directional trend of the higher cost per traffic with a higher conversion rate (but has not been proven statistically).

In [45]:
```
hover_text = []
bubble_size = []

# Add the hover text
for index, row in df_full_display.iterrows():
    hover_text.append(('Channel: {channel}<br>'+
                       'Source: {source}<br>'+
                       'Cost per Traffic: {cpt}<br>'+
                       'Conversion Rate: {cr}<br>'+
                       'Total Spend: {totalspend}<br>').format(channel=row['cha
nnel'],
                                                               source=row['sour
ce'],
                                                               cpt=row['cpt_for
mat'],
                                                               cr=row['cr%'],
                                                               totalspend=row[
'spend_format']))
    bubble_size.append(math.sqrt(row['totalspend']))

# Add the hover text to the dataframe
df_full_display['text'] = hover_text
df_full_display['size'] = bubble_size
sizeref = 2.*max(df_full_display['size'])/(100**2)*2

# Dictionary with dataframes for each channel
channel_names = ['affiliate', 'direct', 'email', 'search', 'social']
channel_data = {channel:df_full_display.query("channel == '%s'" %channel) for
 channel in channel_names}

# Create the bubble for CPT and CR
fig = go.Figure()

for channel_name, channel in channel_data.items():
    fig.add_trace(go.Scatter(
        x=channel['cpt'], y=channel['cr'],
        name=channel_name, text=channel['text'],
        marker_size=channel['size'],
        ))

# Adjust the appearance and layout
fig.update_traces(mode='markers', marker=dict(sizemode='area', sizeref=sizeref
, line_width=2))

fig.update_layout(
    title='Cost per Traffic vs. Conversion Rate by Source',
    xaxis=dict( title='Cost per Traffic', gridcolor='white', gridwidth=2,  tic
kformat='$,2f',  hoverformat='$,2f'),
    yaxis=dict(title='Conversion Rate', gridcolor='white', gridwidth=2,  tickf
ormat=".1%",  hoverformat=".1%"),
    paper_bgcolor='rgb(243, 243, 243)',
    plot_bgcolor='rgb(243, 243, 243)',
)
fig.show()
```



The advice I would give the marketing team on adjusting their spend patterns and why:

1. I suggest the marketing team to invest more on Search, to lift the overall marketing effectiveness and efficiency: Search (Google) got the second smallest investment (25k, 6% of the overall marketing spend) while it saw a relatively low Cost per Traffic, as well as one of the highest conversion rates. Search's conversion rate was statistical significantly higher than Email and Social. Affiliate, the channel with highest conversion rate did not outperform Search statistical significantly.

2. I suggest the marketing team to reduce the spend on Social, shift the budget to other channels (especially Search), while increase the Affiliate budget generally. Affiliate and Social were the two most expensive channels but they performed differently in getting customers converted. Affiliate was the most effective marketing channel, with the highest conversion rate (8.2%). The conversion rate was statistical significantly higher than Direct, Email and Social. Therefore, when it comes to the Cost per Conversion, Affiliate was much lower than the Social (54 dollars per conversion vs. 93 dollars per conversion). The high conversion rate indicates that Affiliate has a lot of potential in earning new customers. With a higher spend, it might also see a lower cost per traffic in the future.

3. At the source level:

3.1 I suggest shifting some budget from New York Times to Lending Tree. Within Affiliate, Lending Tree saw a slightly higher cost per traffic (5 dollars) and a significantly higher conversion rate, compared to New York Times.

3.2 I suggest minimizing the budget of Facebook and YouTube and relocating the money to Instagram to improve the spend effectiveness and efficiency of the overall social media marketing. Instagram performed better than the other two platforms, with a much lower cost per traffic. Instagram's conversion rate was statistical significantly higher than Facebook too. YouTube was the most expensive source, not only within Social channel, but across all the marketing vendors, in attracting traffic and conversion, though it has a higher conversion rate than Facebook. Currently, Instagram only cost 36k, 8% of the total marketing spend, much lower than the 18% of Facebook and 17% of YouTube.

4. I suggest the team relocating a small amount of Email spend to Search: in the current spending, Email and Search shared the same amount of spend (25k). According to the analysis in Question 2, Email was significantly lease likely to convert a customer, compared to Search. Email's Cost per Traffic was also higher than Search, making it more expensive to gain a user. This might be related to the feature of Email as a marketing channel. The Analytics team could work together with the Data Science team to figure out the threshold of the Email spend, then shift the budget to other channels.

5. Direct (Organic) was the top channel (source) with the lowest spend, cost per traffic and cost per conversion, indicating The Company's organic campaign was effective and efficient. This may be related to how this channel works. As I mentioned in the section of Question 1, in my opinion, an Organic marketing source should not have any spend. Therefore, I probably would suggest the marketing team to keep the current budget under current circumstances.

6. These suggestions were made based on the provided data with a limited number of variables. To make further recommendations, more detailed data will be preferred (for example, a dataset with timestamp, ad formats, targeting strategy, and other metrics, such as customer's spend on The Company).

7. If the marketing team decide to make a budget allocation, the Analytics team need to monitor the campaign performance (spend effectiveness and conversion rate) carefully. Analysts can calculate the benchmarks to see if the channel or source performed better as the spend increases or decreases. If the channel / source had reached the diminishing return point, increasing budget would not help to improve its marketing performance.

8. Different marketing channels and vendors plays different roles in the marketing funnel. Some channels might saw a very high conversion rate, but the customers converted from such channels probably came across the ads from other channels before. Some channels might not contribute much to gaining customer, but they helped to build a brand awareness among the customers. If cutting all the budget from these channels, the overall performance of other channels might get impacted in a negative way.

The Analytics team could work together with the Data Science team, use the historical data to create the Marketing Mix Model, Multitouch Attribution, or other statistical models, to get a deeper understanding of the marketing performance. In this way, we could provide the marketing team better suggestions on budget allocation and optimization The Company's campaigns.

How I would explain or handle artifacts such as the performance from none:

1. I calculated the proportion of the conversion with the 'none' of the total numbers of conversions. It took only 3.6% of them. The share was not very large, so I filtered the 'none' out in previous analysis.

2. The conversions from 'none' might come from word-of-mouth or other offline marketing source, such as radio, TV, Subway ads or print ads (newspaper, magazine, or banners). The Analytics team could work with the Data science team, using modeling (Multitouch Attribution or Marketing Mix Modeling) to figure out the effects of offline marketing (if there is any).

3. If the 'none' source had made a very large portion of the total conversions, or an increasing number of conversions with no source had been seen in a short time, the analytics team should do a trouble shooting to figure out which channel / vendor caused such problem, and then work with the marketing team and data engineer team to see if there was any tagging issues or problems during data ETL process. If the analytics team had got an urgent project which need to be done before the data discrepancies were solve, I suggest using the average of Cost per Traffic across channels to calculate an estimated spend of ‘none’.

4. The Analytics team could also design a quick questionnaire to survey these 'none' customers (since they've already been converted, The Company should have their contact information), figuring out where they came from.

Question 4

4. Last, this data is pretty simplistic, what are some other attributes or factors that may affect marketing spend efficiency and conversion rates for our customers?

These are some factors which may affect the marketing spend efficiency and conversion rate:

1. Campaign objective (KPI): the objective will affect the marketing spend efficiency and conversion rate. If the campaign aims at building a larger audience base and increasing the brand awareness among customers, the campaign optimization strategy and marketing channel choice will be quite different from the campaign for gaining conversions. One of the most important things in marketing analysis is learning about campaign objectives.

2. Targeting audience: Audience's demographic factors, including age, job titles, and their locations, may have an impact on the spend effectiveness, especially the conversation rate. Studying the targeting audience will help to improve the marketing effectiveness. (This may also affect customers spend on the platform after converted. But this is an LTV problem. The Analytics team may start another kind of analysis on it.)

3. Categories of products: The Company provides service on different kinds of insurance products, including auto, health, home, and pet. The product category may affect the spend efficiency and

conversation rate. Some products may be easier to get customers' attention or get customers converted The Analytics team could work with the data science team to figure out a reasonable benchmark for each category.

4. Competitors: The competitors' digital marketing campaign may affect The Company's marketing effectiveness. The Insights team could create competitive analysis when the new competitors appears, or the existing competitors launch huge new campaigns or making significant strategy changes.

5. Creative / Media tactic / Ad format: ad's creative and copy writing play an important role in digital marketing. For instance, if the ad is launched with an attractive picture or video clip, it will be easier to get people's attention, earning impressions and clicks. The call-to-action copy writing probably leads to a higher conversion rate. A/B testing could be designed to test such factors.

6. Targeting tactics: Targeting tactics may affect the conversion rate and spend efficiency too. In digital marketing, many customers will not be converted on the first time they saw the ads or visited the website. Retargeting the customers who have visited The Company's website may be easier to get them converted than targeting the brand-new audience. Targeting at FIRST- or THIRD-party audience may also lead to the different performance. The targeting strategy should be decided by the campaign objective mentioned in 1. The analytics team should also work with the data science team to figure out the benchmarks for the performance to monitor and evaluate the campaign.

7. Multitouch: As I mentioned in Question 3, different marketing channels and vendors works variously in the marketing funnel. In the current dataset, we only got the last touch of the conversion process (the channel directly leads to the conversion got 100% credit). However, the customers might saw ads from other channels before they finally converted. If we tracked customers' conversion path, the Analytics team could work together with the Data Science team to explore the correlations among marketing of different channels.

8. Landing pages: landing page mainly affects the conversion rate. If a customer landing on The Company's website were provided with attractive content and an easy path to register, this customer would be more likely to be converted. To optimize the landing page content and UI/UX design, Analytics could run A/B testing as well as marketing research with questionnaire.

9. Keywords and Matching Type (specific for Paid Search): in Search campaigns, Keywords and Matching Types have a huge impact on the spend efficiency. If the keywords were too broad, it might be too competitive to bid for; if the keywords were too specific then not many people would search them. The analytics team could do some research around keywords and work closely with the Paid Search media team to figure out the best practice of the search campaigns.

10. Time and Events: In the provided dataset, there is no time-related variables. Several time-related factors may affect the campaign effect:

10.1 Weekday/Weekend: Based on the historical data, the analytics team could analyze if the day of week have an impact on the marketing effectiveness.

10.2 Time: the time of the day may have an impact on the customer's behavior.

10.3 Time of the year: some categories of the insurance have a specific renewing time (for example, most of the health insurance will be renewed at the beginning of the year). If the marketing campaign focusing on health insurance were launched at the time, it might lead to a lower cost per traffic and higher conversion rate.

10.4 Holiday Season: Different companies have various marketing strategies during holiday season. Currently the advertising market is ready competitive but also has a large traffic. Campaigning during holiday season may lead to a higher cost per traffic. But together with the discounts, campaigns might also get a higher conversion rate. The analytics team should partner with Data Science team and media agency to figure out how effective the holiday season campaign would be.

11. Device: Device may affect the conversion rate. Users have different behaviors on desktops, smart phones, and tablets. The mobile device may be easier to get a visit or impression while desktop is a better end for people to register (be converted). The conversion rate on different devices may be related to the webpage design (if it is mobile friendly). As mentioned before, the Analytics team could design A/B testing together with the Creative and UI/UX team to optimize the user experience on different devices and improve the conversion rate.