```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df = pd.read_csv('/content/drive/MyDrive/marketing_campaign_dataset.csv')
```

```
df.head()
```

	Campaign_ID	Company	Campaign_Type	Target_Audience	Duration	Channel_Used	Conversion_Rate	Acquisition_Cost	ROI	Location	
0	1	Innovate Industries	Email	Men 18-24	30 days	Google Ads	0.04	$16,174.00	6.29	Chicago	1/1/
1	2	NexGen Systems	Email	Women 35-44	60 days	Google Ads	0.12	$11,566.00	5.61	New York	1/2/
2	3	Alpha Innovations	Influencer	Men 25-34	30 days	YouTube	0.07	$10,200.00	7.18	Los Angeles	1/3/
3	4	DataTech Solutions	Display	All Ages	60 days	YouTube	0.11	$12,724.00	5.55	Miami	1/4/
4	5	NexGen Systems	Email	Men 25-34	15 days	YouTube	0.05	$16,452.00	6.50	Los Angeles	1/5/

```
df.duplicated().sum()
```

```
0
```

```
df.isnull().sum()
```

	0
Campaign_ID	0
Company	0
Campaign_Type	0
Target_Audience	0
Duration	0
Channel_Used	0
Conversion_Rate	0
Acquisition_Cost	0
ROI	0
Location	0
Date	0
Clicks	0
Impressions	0
Engagement_Score	0
Customer_Segment	0

dtype: int64

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200005 entries, 0 to 200004
Data columns (total 15 columns):
```

```
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Campaign_ID       200005 non-null  int64
 1   Company           200005 non-null  object
 2   Campaign_Type     200005 non-null  object
 3   Target_Audience   200005 non-null  object
 4   Duration          200005 non-null  object
 5   Channel_Used      200005 non-null  object
 6   Conversion_Rate   200005 non-null  float64
 7   Acquisition_Cost  200005 non-null  object
 8   ROI               200005 non-null  float64
 9   Location          200005 non-null  object
 10  Date              200005 non-null  object
 11  Clicks            200005 non-null  int64
 12  Impressions       200005 non-null  int64
 13  Engagement_Score  200005 non-null  int64
 14  Customer_Segment  200005 non-null  object
dtypes: float64(2), int64(4), object(9)
memory usage: 22.9+ MB
```

```
for col in df.columns:
    print(col, '\n', len(df[col].unique()))
```

```
Campaign_ID
 200005
Company
 5
Campaign_Type
 5
Target_Audience
 5
Duration
 4
Channel_Used
 6
Conversion_Rate
 15
Acquisition_Cost
 15001
ROI
 601
Location
 5
Date
 365
Clicks
 901
Impressions
 9001
Engagement_Score
 10
Customer_Segment
 5
```

```
df['CTR(%)'] = df['Clicks'] / df['Impressions'] * 100
```

```
df['Acquisition_Cost'] = df['Acquisition_Cost'].replace({'\$': '', ',': ''}, regex=True).astype(float)
```

```
df['Acquisition_Cost']
```

	Acquisition_Cost
0	16174.0
1	11566.0
2	10200.0
3	12724.0
4	16452.0
...	...
200000	18365.0
200001	8168.0
200002	13397.0
200003	18508.0
200004	13835.0

200005 rows × 1 columns

dtype: float64

```
df['CPC'] = df['Acquisition_Cost'] / df['Clicks']
```

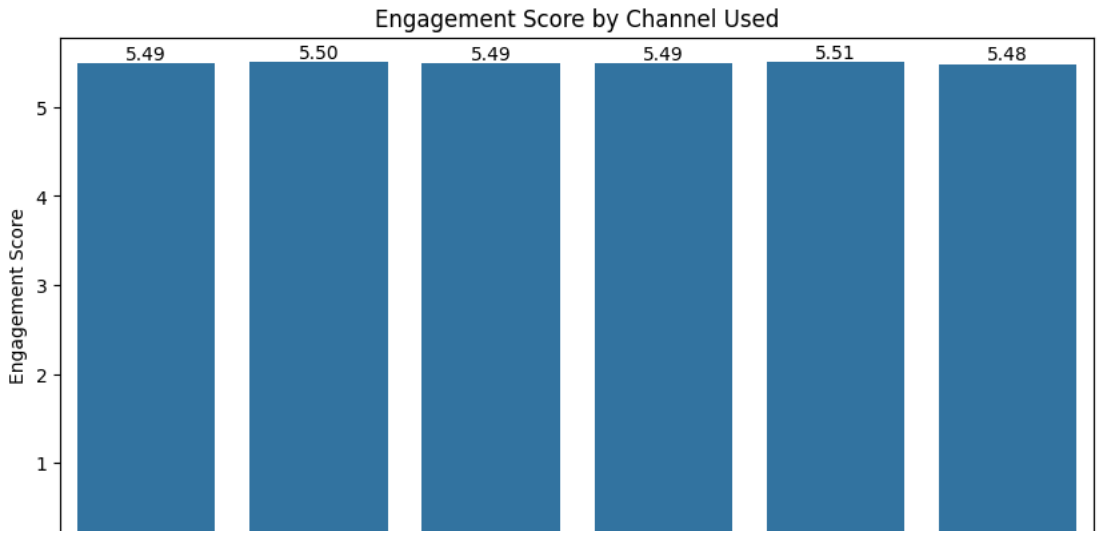
```
df.head()
```

	Campaign_ID	Company	Campaign_Type	Target_Audience	Duration	Channel_Used	Conversion_Rate	Acquisition_Cost	ROI	Location	
0	1	Innovate Industries	Email	Men 18-24	30 days	Google Ads	0.04	16174.0	6.29	Chicago	1/1/
1	2	NexGen Systems	Email	Women 35-44	60 days	Google Ads	0.12	11566.0	5.61	New York	1/2/
2	3	Alpha Innovations	Influencer	Men 25-34	30 days	YouTube	0.07	10200.0	7.18	Los Angeles	1/3/
3	4	DataTech Solutions	Display	All Ages	60 days	YouTube	0.11	12724.0	5.55	Miami	1/4/
4	5	NexGen Systems	Email	Men 25-34	15 days	YouTube	0.05	16452.0	6.50	Los Angeles	1/5/

```
campaign_performance = df.groupby(['Channel_Used']).agg({'Engagement_Score':'mean'}).reset_index()
campaign_performance
```

	Channel_Used	Engagement_Score
0	Email	5.487842
1	Facebook	5.503748
2	Google Ads	5.493989
3	Instagram	5.489039
4	Website	5.508828
5	YouTube	5.484802

```
plt.figure(figsize=(10, 5))
sns.barplot(x='Channel_Used', y='Engagement_Score', data=campaign_performance)
EG = campaign_performance['Engagement_Score'].values
for index, value in enumerate(EG):
    plt.text(index, value + 0.001, f'{value:.2f}', ha='center', va='bottom')
plt.title('Engagement Score by Channel Used')
plt.xlabel('Channel Used')
plt.ylabel('Engagement Score')
plt.show()
```




```
df['Conversion_Rate'] = df['Conversion_Rate']*100
```

```
df2 = df[['CPC','CTR(%)','Conversion_Rate']]
df2
```

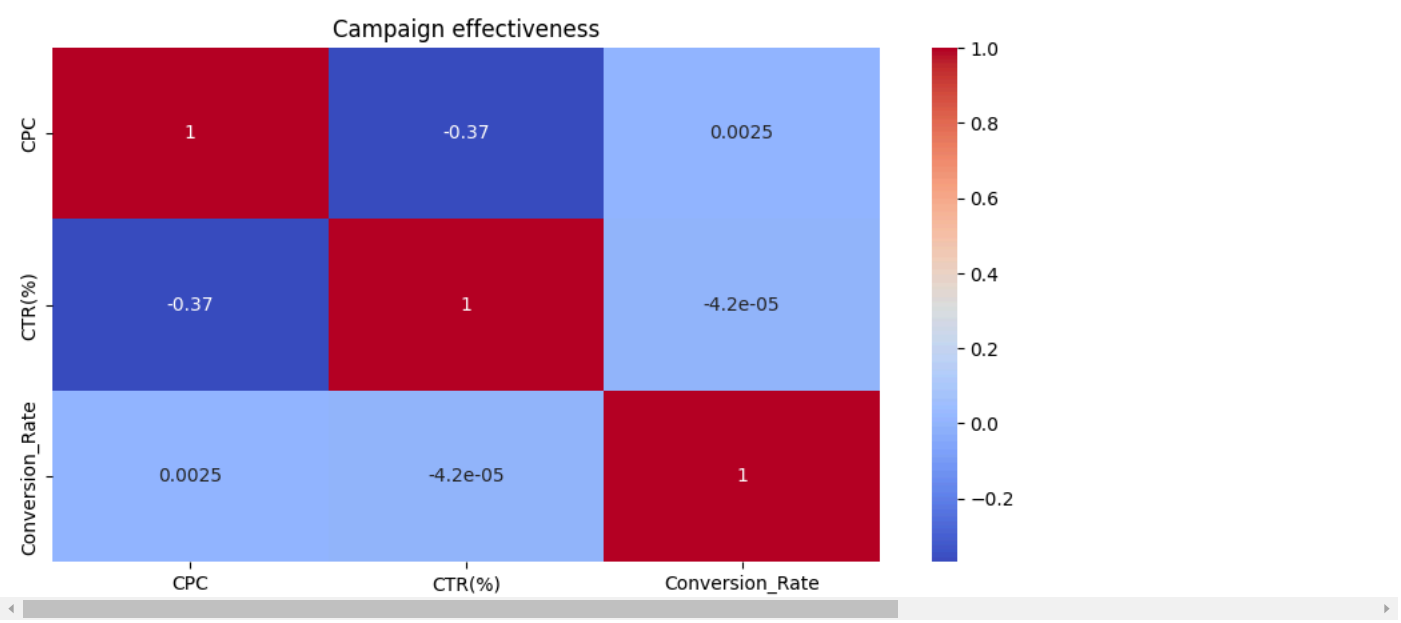
	CPC	CTR(%)	Conversion_Rate
0	31.964427	26.326743	4.0
1	99.706897	1.541938	12.0
2	17.465753	7.586386	7.0
3	58.635945	11.923077	11.0
4	43.408971	9.021662	5.0
...	...	...	...
200000	21.404429	14.328657	6.0
200001	35.824561	7.431551	2.0
200002	18.529737	7.572266	5.0
200003	35.053030	19.109663	10.0
200004	14.972944	12.680115	1.0

200005 rows × 3 columns

```
corr_matrix = df2.corr()
corr_matrix
```

	CPC	CTR(%)	Conversion_Rate
CPC	1.000000	-0.368949	0.002533
CTR(%)	-0.368949	1.000000	-0.000042
Conversion_Rate	0.002533	-0.000042	1.000000

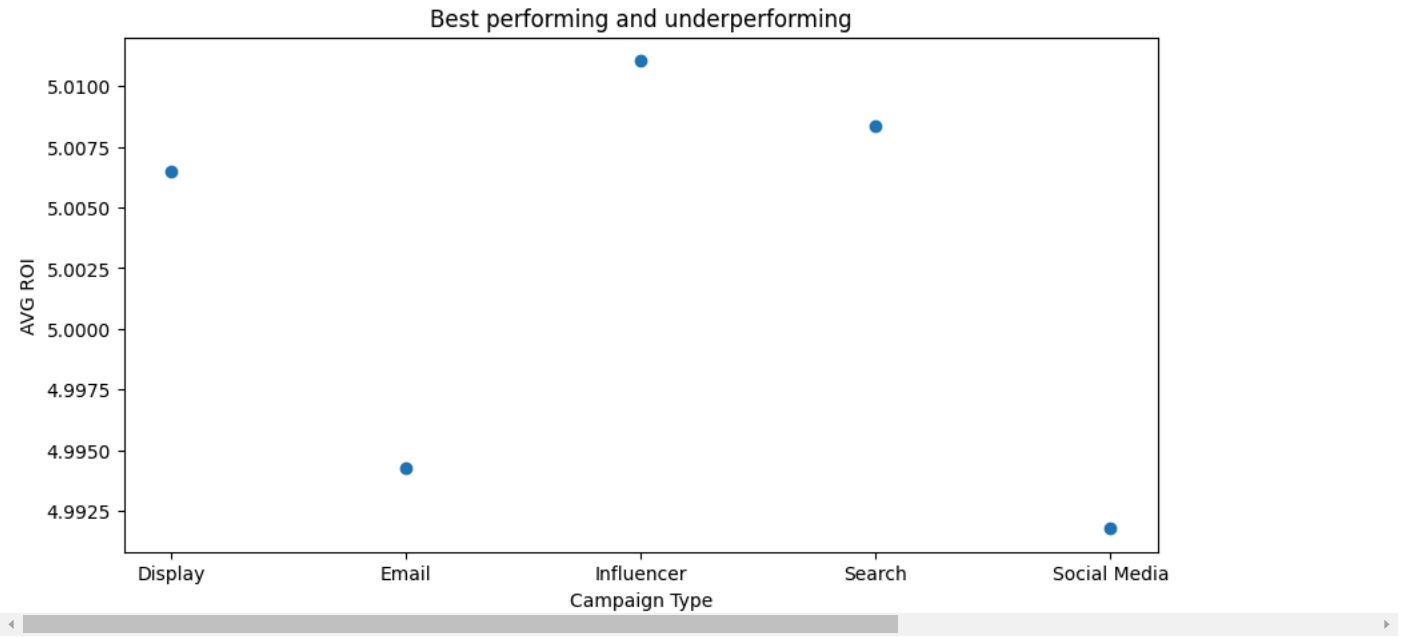
```
plt.figure(figsize=(10, 5))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Campaign effectiveness')
plt.show()
```




```
perf_underperf = df.groupby(['Campaign_Type']).agg({'ROI':'mean'}).reset_index()
perf_underperf
```

	Campaign_Type	ROI
0	Display	5.006497
1	Email	4.994274
2	Influencer	5.011040
3	Search	5.008357
4	Social Media	4.991781

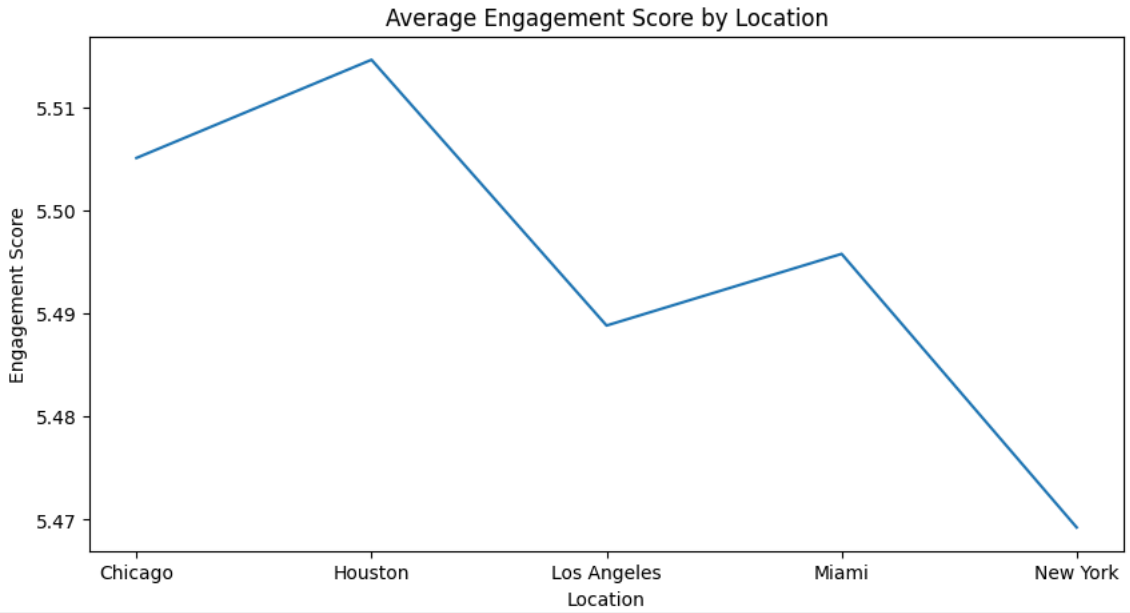
```
plt.figure(figsize=(10, 5))
plt.scatter(perf_underperf['Campaign_Type'], perf_underperf['ROI'])
plt.title('Best performing and underperforming')
plt.xlabel('Campaign Type')
plt.ylabel('AVG ROI')
plt.show()
```




```
location_trend = df.groupby(['Location']).agg({'Engagement_Score':'mean'}).reset_index()
location_trend
```

	Location	Engagement_Score
0	Chicago	5.505061
1	Houston	5.514578
2	Los Angeles	5.488823
3	Miami	5.495766
4	New York	5.469257

```
plt.figure(figsize=(10,5))
sns.lineplot(x='Location', y='Engagement_Score', data=location_trend)
plt.title('Average Engagement Score by Location')
plt.xlabel('Location')
plt.ylabel('Engagement Score')
#plt.xticks(rotation=90)
plt.show()
```




```
df[['Date','Clicks']].head(35)
```

	Date	Clicks
0	1/1/2021	506
1	1/2/2021	116
2	1/3/2021	584
3	1/4/2021	217
4	1/5/2021	379
5	1/6/2021	100
6	1/7/2021	817
7	1/8/2021	624
8	1/9/2021	861
9	1/10/2021	642
10	1/11/2021	321
11	1/12/2021	659
12	13/01/2021	677
13	14/01/2021	994
14	15/01/2021	482
15	16/01/2021	299
16	17/01/2021	931
17	18/01/2021	218
18	19/01/2021	182
19	20/01/2021	193
20	21/01/2021	975
21	22/01/2021	319
22	23/01/2021	646
23	24/01/2021	764
24	25/01/2021	527
25	26/01/2021	809
26	27/01/2021	953
27	28/01/2021	604
28	29/01/2021	384
29	30/01/2021	952
30	31/01/2021	512
31	2/1/2021	309
32	2/2/2021	367
33	2/3/2021	243
34	2/4/2021	360

```
# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)

# Set 'Date' as the index
df.set_index('Date', inplace=True)

# Group by date and resample to monthly frequency, summing clicks for each month
click_trend = df['Clicks'].resample('M').sum()

# Display monthly summed click trend
print(click_trend)
```

```
Date
2021-01-31    9321651
2021-02-28    8433098
2021-03-31    9357568
```