IIT Data Visualisation

1. Import Libraries

In [1]:

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

```
/Users/admin/anaconda3/lib/python3.7/site-packages/pandas/compat/_opti
onal.py:138: UserWarning: Pandas requires version '2.7.0' or newer of
'numexpr' (version '2.6.9' currently installed).
  warnings.warn(msg, UserWarning)
```

2. Load Dataset

In [2]:

```
data = pd.read_csv('Marketing.csv')
data
```

Out[2]:

	Campaign ID	Campaign Name	Audience	Age	Geography	Reach	Impressions	Frequency	Clicks	Uniq Clic
0	Campaign 1	SHU_6(Educators and Principals)	Educators and Principals	25-34	Group 1 (Australia, Canada, United Kingdom, Gh...	11387	23283	2.044700	487	4
1	Campaign 1	SHU_6(Educators and Principals)	Educators and Principals	35-44	Group 1 (Australia, Canada, United Kingdom, Gh...	8761	15683	1.790092	484	3

3. Description of Data

In [3]:

```
data.describe()
```

Out[3]:

	Reach	Impressions	Frequency	Clicks	Unique Clicks	Unique Link Clicks (ULC)	Click Throug Rat (CTF
count	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.00000
mean	5723.272727	8783.636364	1.418234	364.393939	288.000000	159.303030	4.81455
std	7671.821261	11327.981290	0.471318	556.340326	432.381125	257.960933	2.90168
min	91.000000	103.000000	1.042614	9.000000	8.000000	3.000000	1.66889
25%	889.000000	1874.000000	1.131868	49.000000	44.000000	27.000000	2.59259
50%	2557.000000	3146.000000	1.174759	135.000000	111.000000	63.000000	3.98268
75%	6145.000000	12372.000000	1.665733	325.000000	246.000000	129.000000	6.62138
max	30110.000000	39161.000000	3.169081	2593.000000	1994.000000	1095.000000	12.95180

4. Data Type of each Variable

In [4]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33 entries, 0 to 32
Data columns (total 16 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Campaign ID                           33 non-null     object 
 1   Campaign Name                         33 non-null     object 
 2   Audience                              33 non-null     object 
 3   Age                                   33 non-null     object 
 4   Geography                             33 non-null     object 
 5   Reach                                 33 non-null     int64  
 6   Impressions                           33 non-null     int64  
 7   Frequency                             33 non-null     float64
 8   Clicks                                33 non-null     int64  
 9   Unique Clicks                         33 non-null     int64  
 10  Unique Link Clicks (ULC)              33 non-null     int64  
 11  Click-Through Rate (CTR)              33 non-null     float64
 12  Unique Click-Through Rate (Unique CTR)  33 non-null     float64
 13  Amount Spent in INR                   33 non-null     float64
 14  Cost Per Click (CPC)                  33 non-null     float64
 15  Cost Per Result (CPR)                 33 non-null     float64
dtypes: float64(6), int64(5), object(5)
memory usage: 4.2+ KB
```

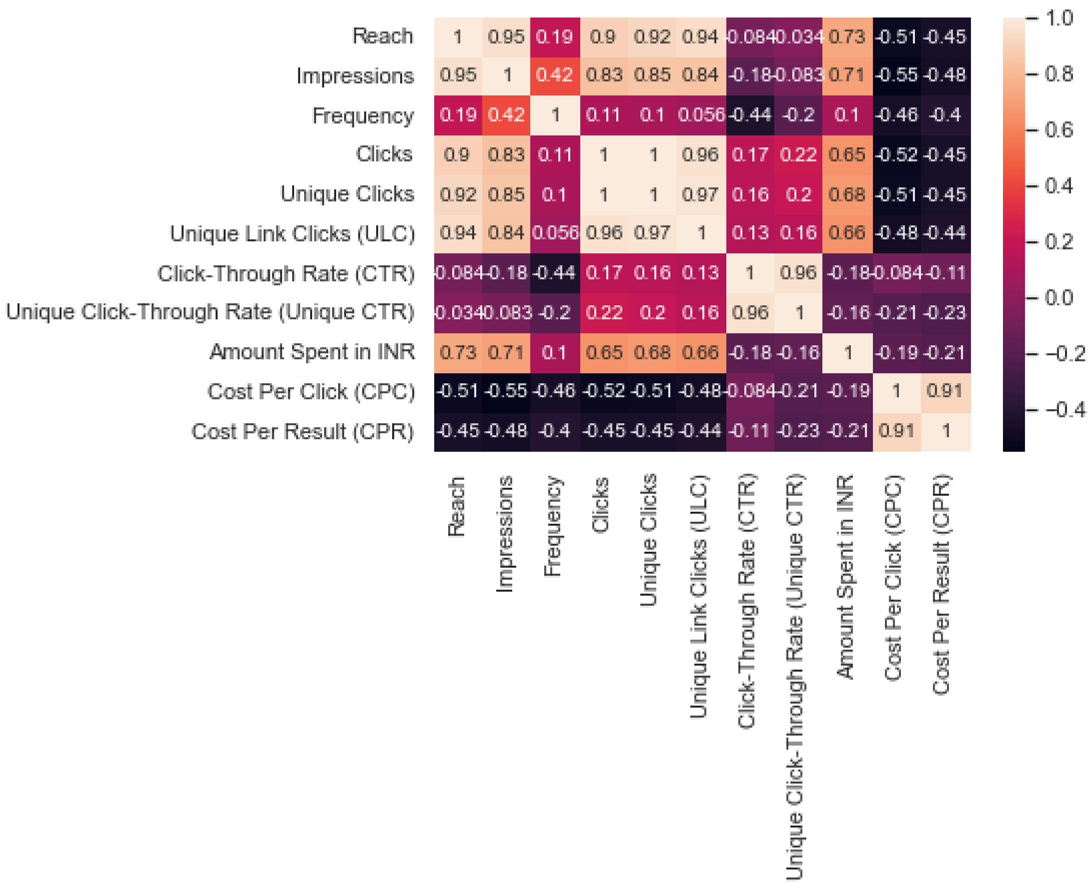
5. Correlation Matrix

In [5]:

```
cor = data.corr()
sns.set(rc = {'figure.figsize':(10,10)})

sns.heatmap(cor, annot = True)
```

Out[5]:

```
<AxesSubplot:>
```

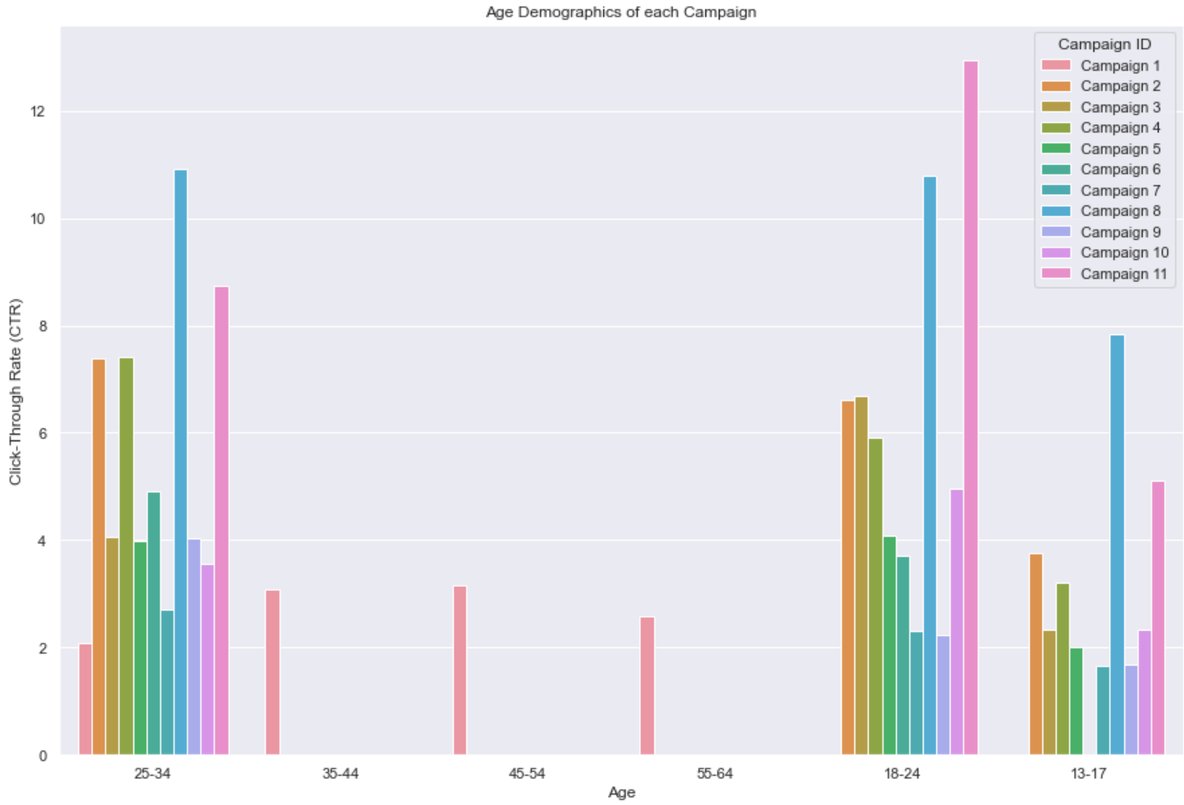


The correlation matrix indicates the correlation between each variable. This applies to only continuous variables. Values closer to positive 1 indicate stronger correlation and values closer to negative 1 indicate weaker correlation.

Age groups

In [6]:

```
sns.set(rc = {'figure.figsize':(15,10)})

plt.title("Age Demographics of each Campaign")
sns.barplot(x = data['Age'],y = data['Click-Through Rate (CTR)'],data = data, hue =
plt.show()
```



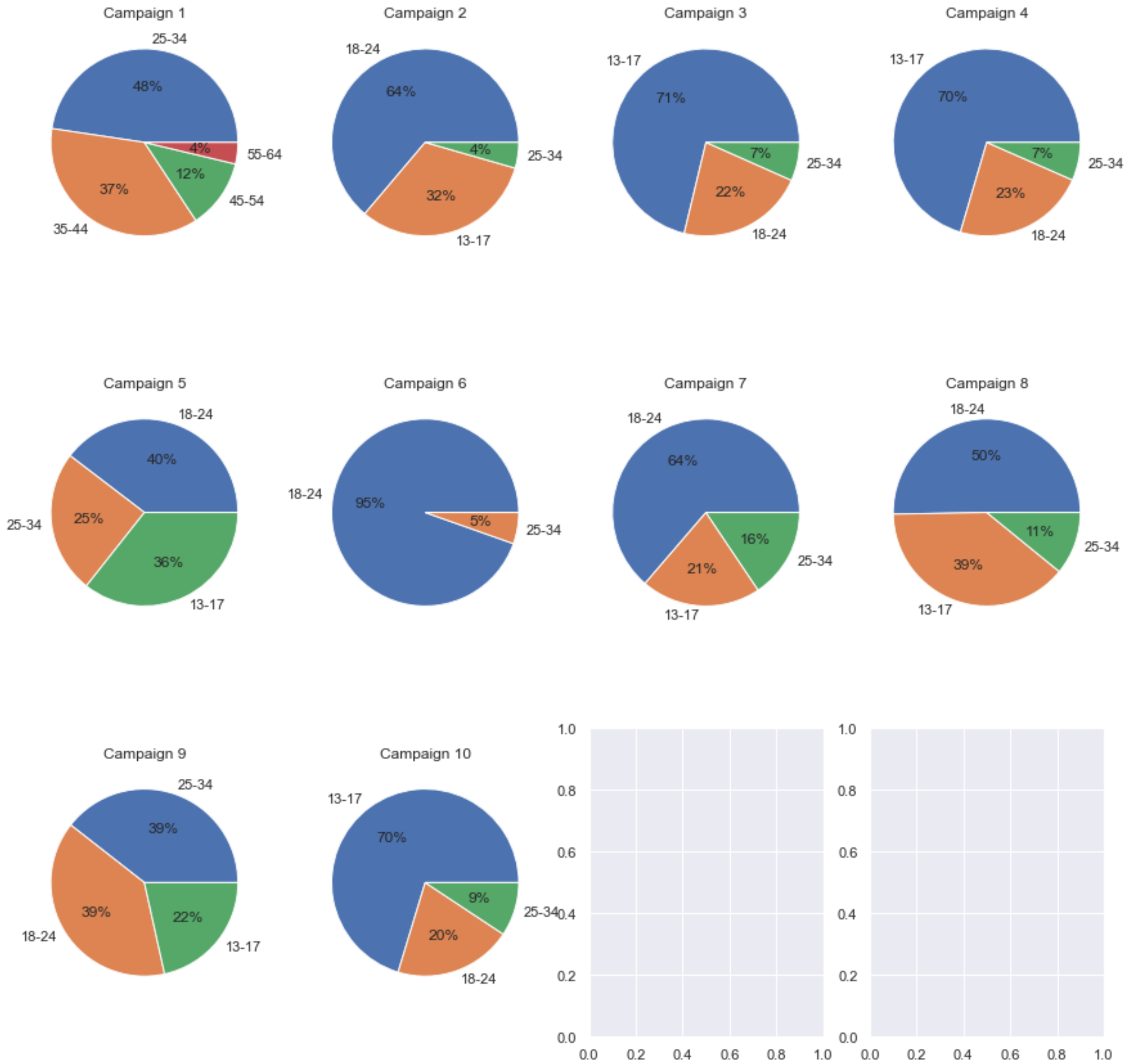
In [7]:

```
c1 = data[data['Campaign ID'] == 'Campaign 1']
c2 = data[data['Campaign ID'] == 'Campaign 2']
c3 = data[data['Campaign ID'] == 'Campaign 3']
c4 = data[data['Campaign ID'] == 'Campaign 4']
c5 = data[data['Campaign ID'] == 'Campaign 5']
c6 = data[data['Campaign ID'] == 'Campaign 6']
c7 = data[data['Campaign ID'] == 'Campaign 7']
c8 = data[data['Campaign ID'] == 'Campaign 8']
c9 = data[data['Campaign ID'] == 'Campaign 9']
c10 = data[data['Campaign ID'] == 'Campaign 10']
c11 = data[data['Campaign ID'] == 'Campaign 11']
```

In [8]:

```
figure, axis = plt.subplots(3, 4, figsize = (15,15))

axis[0,0].pie(c1['Reach'],labels = c1['Age'],autopct='%.0f%%')
axis[0,0].set_title("Campaign 1")

axis[0,1].pie(c2['Reach'],labels = c2['Age'],autopct='%.0f%%')
axis[0,1].set_title("Campaign 2")

axis[0,2].pie(c3['Reach'],labels = c3['Age'],autopct='%.0f%%')
axis[0,2].set_title("Campaign 3")

axis[0,3].pie(c4['Reach'],labels = c4['Age'],autopct='%.0f%%')
axis[0,3].set_title("Campaign 4")


axis[1,0].pie(c5['Reach'],labels = c5['Age'],autopct='%.0f%%')
axis[1,0].set_title("Campaign 5")

axis[1,1].pie(c6['Reach'],labels = c6['Age'],autopct='%.0f%%')
axis[1,1].set_title("Campaign 6")

axis[1,2].pie(c7['Reach'],labels = c7['Age'],autopct='%.0f%%')
axis[1,2].set_title("Campaign 7")

axis[1,3].pie(c8['Reach'],labels = c8['Age'],autopct='%.0f%%')
axis[1,3].set_title("Campaign 8")


axis[2,0].pie(c9['Reach'],labels = c9['Age'],autopct='%.0f%%')
axis[2,0].set_title("Campaign 9")

axis[2,1].pie(c10['Reach'],labels = c10['Age'],autopct='%.0f%%')
axis[2,1].set_title("Campaign 10")

axis[3,2].pie(c11['Reach'],labels = c11['Age'],autopct='%.0f%%')
axis[3,2].set_title("Campaign 11")

plt.show()
```

```
----------------------------------------------------------------------
-----
IndexError                                Traceback (most recent call
 last)
<ipython-input-8-7177a4a584c9> in <module>
     33 axis[2,1].set_title("Campaign 10")
     34 
---> 35 axis[3,2].pie(c11['Reach'],labels = c11['Age'],autopct='%.0f%
%')
     36 axis[3,2].set_title("Campaign 11")
     37 

IndexError: index 3 is out of bounds for axis 0 with size 3
```

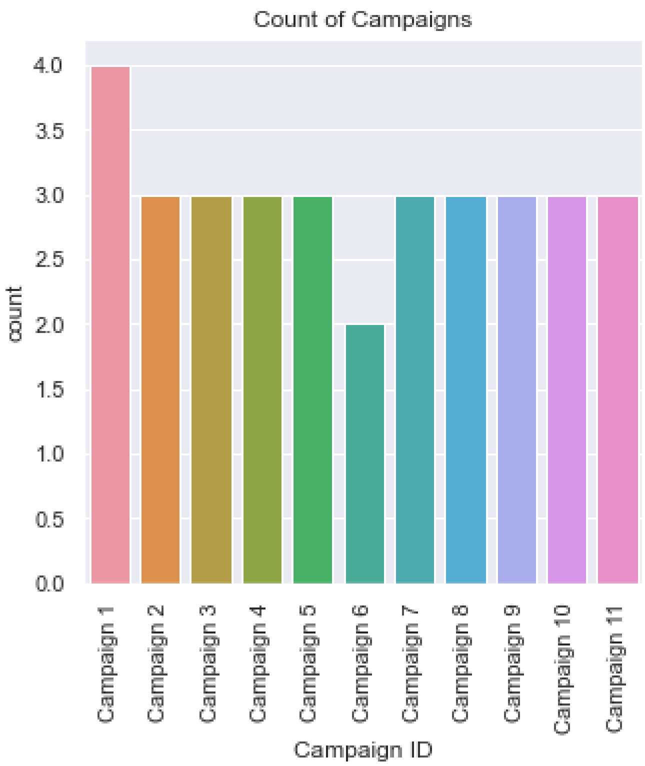


From the above Pie charts, it is clear that majority of the campaigns have an audience in the 18-24 age group.

Number of Campaigns

In [9]:

```
sns.set(rc = {'figure.figsize':(5,5)})

sns.countplot(x = data['Campaign ID'], data = data)
plt.xticks(rotation = 90)
plt.title("Count of Campaigns")
plt.show()
```



Grouping by Campaign

In [10]:

```
campaign_data = data.groupby(["Campaign ID"])['Reach','Impressions','Clicks','Unique
campaign_data
```

```
/Users/admin/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.
py:1: FutureWarning: Indexing with multiple keys (implicitly converted
to a tuple of keys) will be deprecated, use a list instead.
  """Entry point for launching an IPython kernel.
```

Out[10]:

	Campaign ID	Reach	Impressions	Clicks	Unique Clicks	Unique Link Clicks (ULC)	Unique Click-Through Rate (Unique CTR)	Amount Spent in INR
0	Campaign 1	23904	47139	1218	967	420	17.414206	2333.33
1	Campaign 10	3636	4091	121	105	57	10.089332	856.67
2	Campaign 11	2555	2900	178	156	126	26.063625	897.68
3	Campaign 2	46494	67313	3743	2833	1595	18.275484	1579.02
4	Campaign 3	3187	3572	119	109	44	12.729516	850.68
5	Campaign 4	3307	4267	171	146	112	17.659132	923.96
6	Campaign 5	15024	20483	648	552	237	11.179020	837.78
7	Campaign 6	31831	37246	1400	1238	987	8.275220	955.21
8	Campaign 7	29668	65215	1420	1146	518	12.229233	1035.24
9	Campaign 8	21929	28974	2765	2058	1073	28.078127	942.78
10	Campaign 9	7333	8660	242	194	88	7.650006	876.26

In [11]:

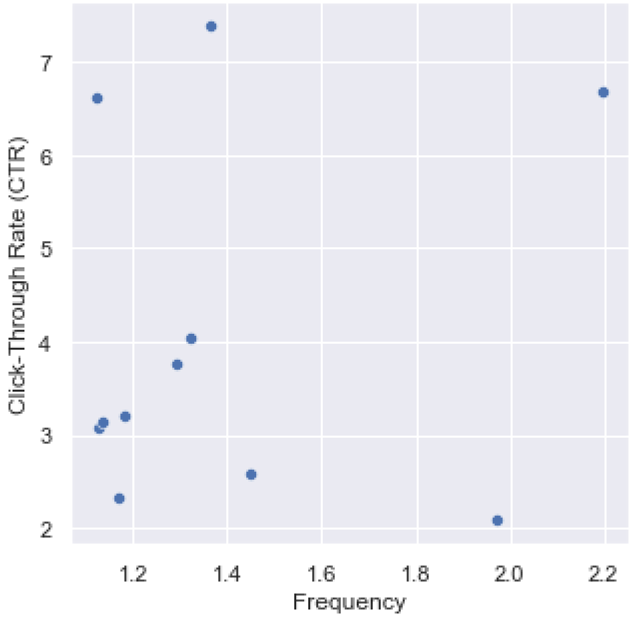
```
campaign_data['Frequency'] = campaign_data['Impressions']/campaign_data['Reach']
campaign_data['Click-Through Rate (CTR)'] = (campaign_data['Clicks']/campaign_data['
campaign_data['Cost Per Click (CPC)'] = campaign_data['Amount Spent in INR']/campaig
campaign_data['Cost Per Result (CPR)'] = campaign_data['Amount Spent in INR']/campai
campaign_data
```

Out[11]:

	Campaign ID	Reach	Impressions	Clicks	Unique Clicks	Unique Link Clicks (ULC)	Unique Click-Through Rate (Unique CTR)	Amount Spent in INR	Frequency	Th
0	Campaign 1	23904	47139	1218	967	420	17.414206	2333.33	1.972013	2.5
1	Campaign 10	3636	4091	121	105	57	10.089332	856.67	1.125138	2.9
2	Campaign 11	2555	2900	178	156	126	26.063625	897.68	1.135029	6.1
3	Campaign 2	46494	67313	3743	2833	1595	18.275484	1579.02	1.447778	5.5
4	Campaign 3	3187	3572	119	109	44	12.729516	850.68	1.120803	3.3
5	Campaign 4	3307	4267	171	146	112	17.659132	923.96	1.290293	4.0
6	Campaign 5	15024	20483	648	552	237	11.179020	837.78	1.363352	3.1
7	Campaign 6	31831	37246	1400	1238	987	8.275220	955.21	1.170117	3.7
8	Campaign 7	29668	65215	1420	1146	518	12.229233	1035.24	2.198160	2.1
9	Campaign 8	21929	28974	2765	2058	1073	28.078127	942.78	1.321264	9.5
10	Campaign 9	7333	8660	242	194	88	7.650006	876.26	1.180963	2.7

In [12]:

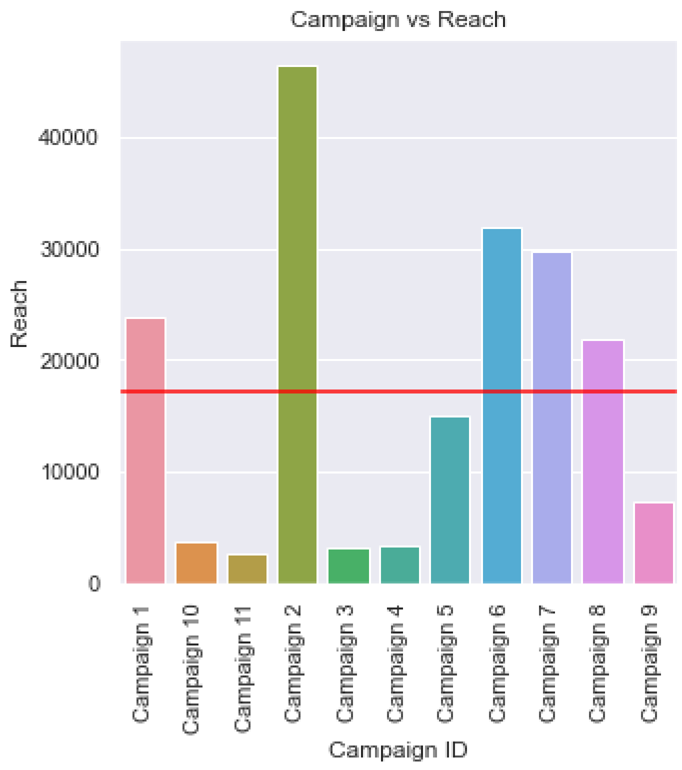
```
sns.set(rc = {'figure.figsize':(5,5)})

sns.scatterplot(x = campaign_data['Frequency'], y = data['Click-Through Rate (CTR)']
plt.show()
```



Campaign vs Reach

In [13]:

```
sns.set(rc = {'figure.figsize':(5,5)})

sns.barplot(x = campaign_data['Campaign ID'],y = campaign_data['Reach'], data = camp
plt.xticks(rotation = 90)
plt.title("Campaign vs Reach")
plt.axhline(y=np.nanmean(campaign_data['Reach']),color = 'red')
plt.show()
```

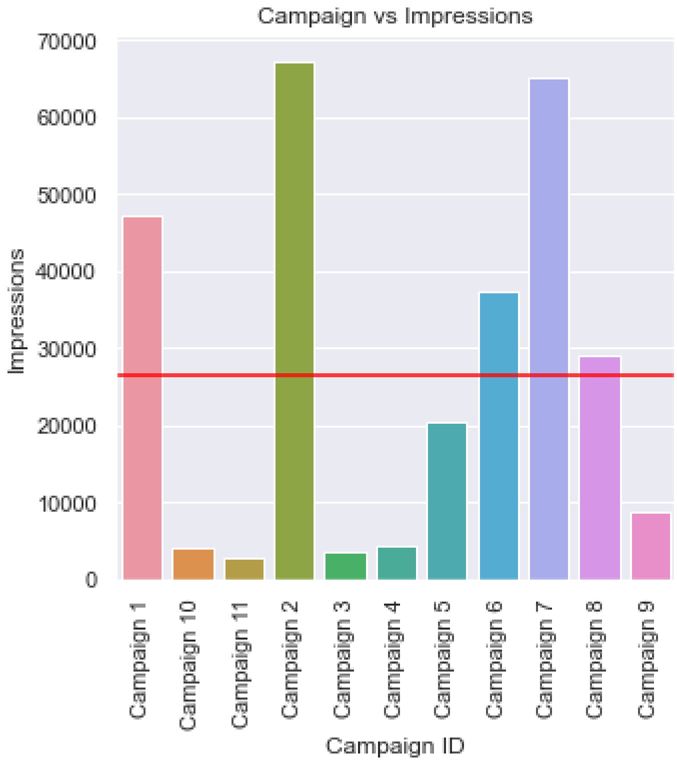


In the above plot, the red line indicates the average reach of all the campaigns. Campaign 3,4,5,9,10 and 11 are below the average reach.

Campaign vs Impressions

In [14]:

```
sns.barplot(x = campaign_data['Campaign ID'],y = campaign_data['Impressions'], data
plt.xticks(rotation = 90)
plt.title("Campaign vs Impressions")
plt.axhline(y=np.nanmean(campaign_data['Impressions']),color = 'red')
plt.show()
```

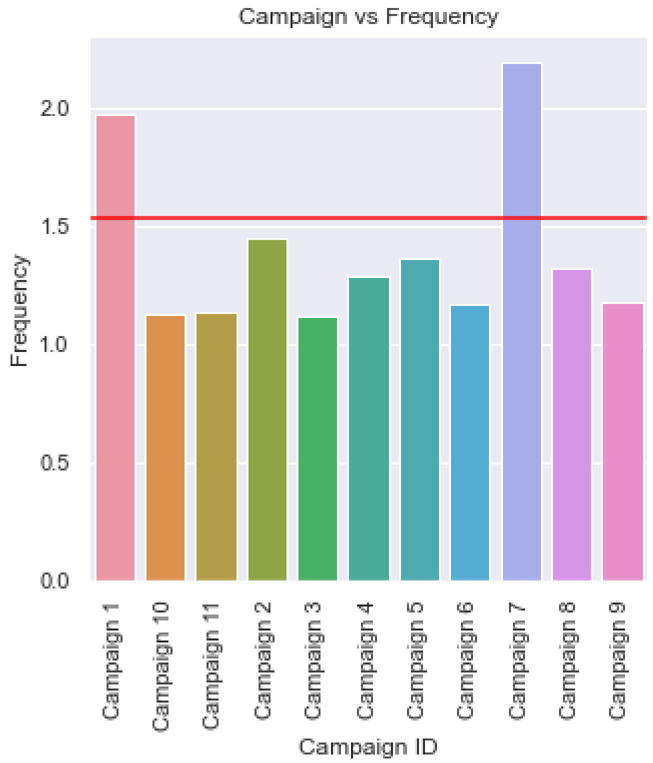


In the above plot, the red line indicates the average impression of all the campaigns. Campaign 3,4,5,9,10 and 11 are below the average impression.

Campaign vs Frequency

In [15]:

```
sns.barplot(x = campaign_data['Campaign ID'],y = campaign_data['Frequency'], data = 
plt.xticks(rotation = 90)
plt.title("Campaign vs Frequency")
plt.axhline(y=(campaign_data['Impressions'].sum()/campaign_data['Reach'].sum()),colo
plt.show()
```

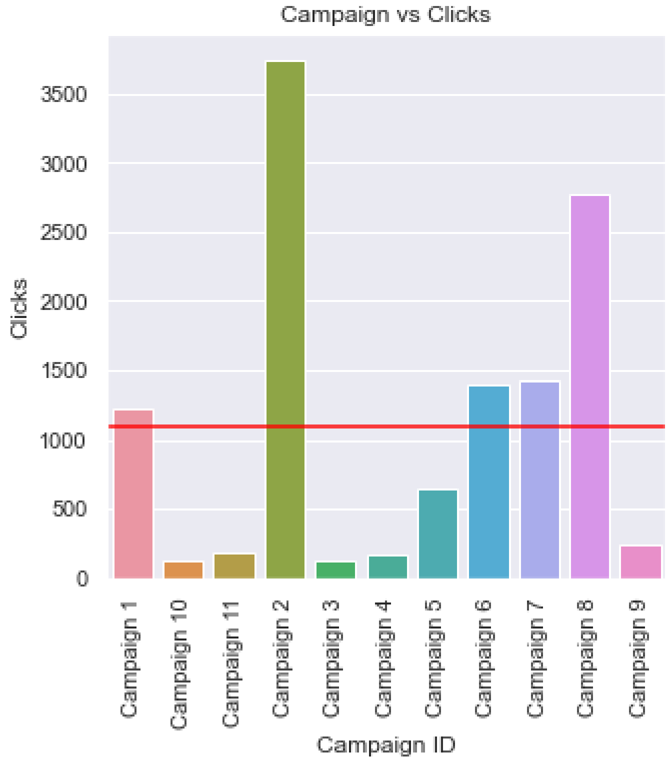


In the above plot, the red line indicates the average frequency of all the campaigns. Campaigns 1 and 7 have a higher frequency. This indicates that the ad appears on the user screen more often than the other campaigns.

Campaign vs Clicks

In [16]:

```
sns.barplot(x = campaign_data['Campaign ID'],y = campaign_data['Clicks'], data = cam
plt.xticks(rotation = 90)
plt.title("Campaign vs Clicks")
plt.axhline(y=np.nanmean(campaign_data['Clicks']),color = 'red')
plt.show()
```

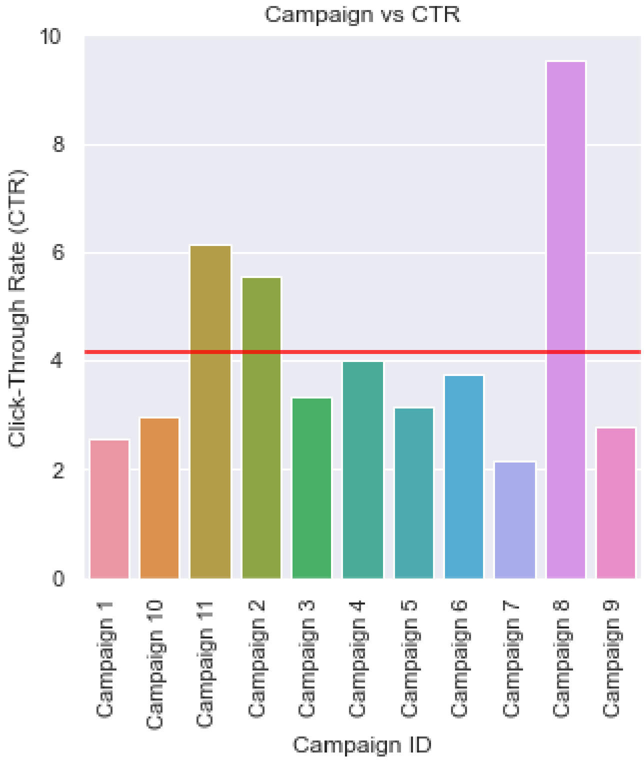


The red line indicates the average number of clicks. Campaign 2 and 8 have far more number of clicks when compared to the average.

Campaign vs CTR

In [17]:

```
sns.barplot(x = campaign_data['Campaign ID'],y = campaign_data['Click-Through Rate (
plt.xticks(rotation = 90)
plt.title("Campaign vs CTR")
plt.axhline(y=(campaign_data['Clicks'].sum()/campaign_data['Impressions'].sum())*100
plt.show()
```

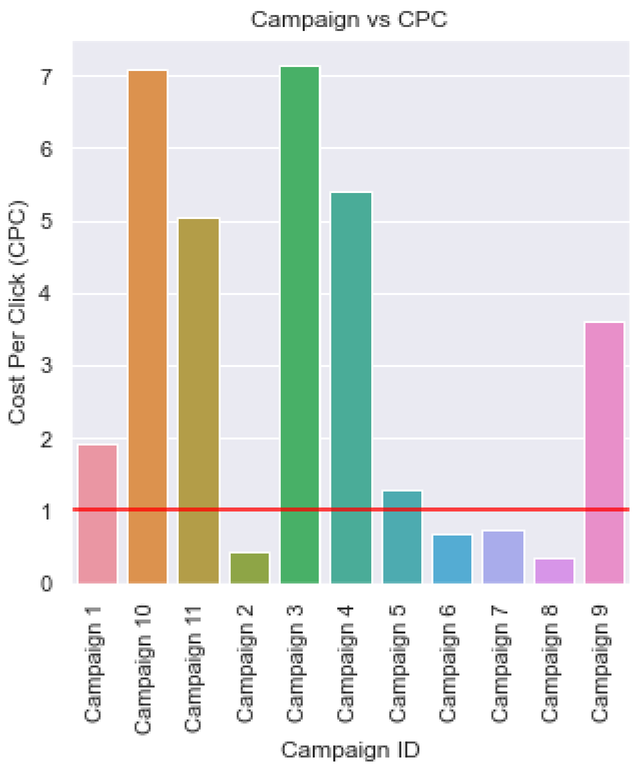


The red line indicates the average click through rate. Campaign 2,8 and 11 have a CTR higher than the average.

Campaign vs CPC

In [18]:

```
sns.barplot(x = campaign_data['Campaign ID'],y = campaign_data['Cost Per Click (CPC)
plt.xticks(rotation = 90)
plt.title("Campaign vs CPC")
plt.axhline(y=(campaign_data['Amount Spent in INR'].sum()/campaign_data['Clicks'].su
plt.show()
```

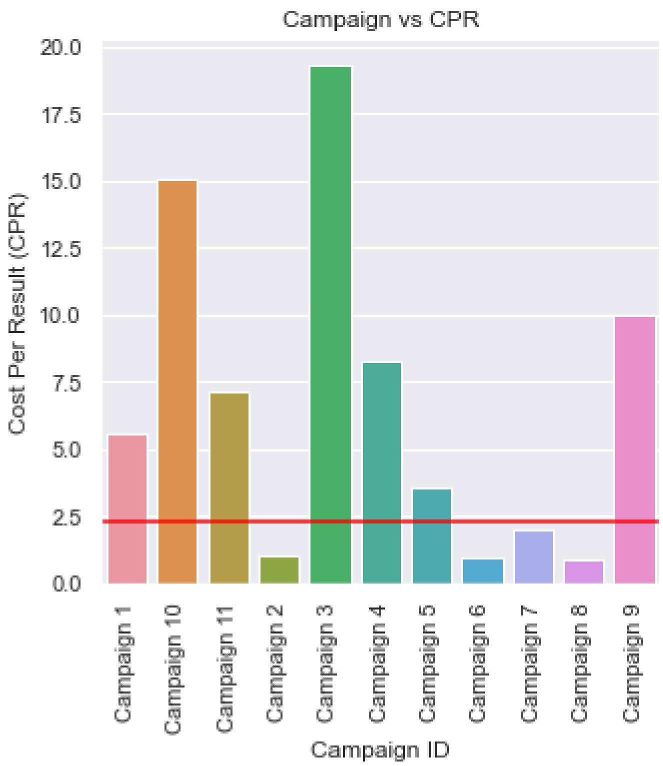


The red line indicates the average CPC. Campaign 3 and 10 have a significantly higher CPC than the average.

Campaign vs CPR

In [19]:

```
sns.barplot(x = campaign_data['Campaign ID'],y = campaign_data['Cost Per Result (CPR
plt.xticks(rotation = 90)
plt.title("Campaign vs CPR")
plt.axhline(y=(campaign_data['Amount Spent in INR'].sum()/campaign_data['Unique Link
plt.show()
```



The red line indicates the average CPR. Campaign 3 and 10 have a higher CPR than the average.

Campaign 3 and 10 have very low reach and impressions. They also have a high CPC and CPR when compared to rest of the campaigns. Therefore, we suggest to discontinue campaign 3 and 10.