Project:- **Ads CONVERSION RATES**:

Dataset : We collected DataSet From the Medeley Website

GOAL:-

- This project aims to analyze the Conversion rates of Online advertising campaigns.
- Conversion rate refers to the percentage of users or visitors who clicks on a ads after clicking the ads.
- CTR refers Click Through Rate is used to measure the percentage of user who click on a ads.
- Conversions refers specific action that a user takes on a ads.
- Impressions means no of times an ad displayed to a user

In [166]:
```
import pandas as pd  # pandas library using for read the dataset file n data validation

import seaborn as sns
import matplotlib.pyplot as plt #these two libraries using for data visulaization purpo


import warnings
warnings.filterwarnings('ignore')  # To ignore  the Warnings we are use warnings librar
```

In [167]:
```
pd.__version__  # pandas Version
```

Out[167]: '1.5.3'

In [168]:
```
#Loading the data set

ads_data = pd.read_csv(r"C:\Users\Mounika Reddy\Downloads\Dataset_Ads.csv")  #reading t
#pandas library
```

In [169]:
```
ads_data.sample(5)
```

Out[169]:

	Age	Gender	Income	Location	Ad Type	Ad Topic	Ad Placement	Clicks	Click Time	Conversion Rate
3957	26	Male	95974.09	Urban	Banner	Technology	Social Media	3	2024-03-12 20:45:56.910350	0.0560
7191	21	Female	28762.38	Rural	Banner	Technology	Search Engine	6	2023-05-24 20:45:56.919400	0.2388
7341	47	Female	35517.37	Suburban	Banner	Finance	Search Engine	2	2023-06-12 20:45:56.920349	0.2595
6107	4	Female	42742.66	Rural	Banner	Fashion	Website	4	2023-04-22 20:45:56.916351	0.1852
6812	47	Female	57768.71	Rural	Banner	Food	Search Engine	5	2023-10-08 20:45:56.918348	0.1649

DATA VALIDATION

used to checking wheather the data is in correct Format or not

In [170]:
```
# Here income and Age column has negative values which are the data entry errors so we
# them with using abs() function to make them positive

ads_data['Age'] = ads_data['Age'].abs()
ads_data['Income'] = ads_data['Income'].abs()
```

in age column had 0-5 ages also these ages are not even able to click the ads hence we are filtering these ages

In [171]:
```
ads_data_filtered = ads_data[ads_data['Age']>5]
```

In [172]:
```
ads_data = ads_data_filtered
```

In [173]:
```
#Checking for duplicates

duplicates = ads_data.duplicated().sum()
```

In [174]:
```
print(f'Duplicates: {duplicates}')
```

```
Duplicates: 0
```

In [175]:
```
ads_data['Click Time'] = pd.to_datetime(ads_data['Click Time'])
```

In [176]:
```

ads_data['Date'] = ads_data['Click Time'].dt.date
ads_data['Time'] = ads_data['Click Time'].dt.time
 # Convert 'time_column' to a datetime object
ads_data['Time'] = pd.to_datetime(ads_data['Time'], format='%H:%M:%S.%f')
 # Remove microseconds by flooring the time to the nearest second
ads_data['Time'] = ads_data['Time'].dt.floor('S')
```

In [177]:
```
ads_data.head()
```

Out[177]:

	Age	Gender	Income	Location	Ad Type	Ad Topic	Ad Placement	Clicks	Click Time	Conversion Rate	CT
0	61	Male	35717.43	Urban	Banner	Travel	Social Media	3	2024-01-18 20:45:56.898459	0.0981	0.073
1	41	Male	47453.25	Rural	Video	Travel	Search Engine	5	2023-04-24 20:45:56.898459	0.0937	0.059
2	49	Female	68126.35	Rural	Text	Food	Social Media	4	2024-02-24 20:45:56.898459	0.1912	0.056
3	68	Female	64585.73	Suburban	Text	Health	Website	6	2023-12-13 20:45:56.898459	0.1122	0.023
4	63	Male	21109.40	Urban	Native	Fashion	Search Engine	5	2023-07-02 20:45:56.898459	0.1426	0.053

In [178]:
```
ads_data = ads_data.drop(columns=['Click Time'])
```

In [179]:
```
#ads_data['Impressions'] = ads_data['Clicks'] / ads_data['CTR']
```

In [180]:
```
ads_data['Conversions'] = ads_data['Clicks'] * (ads_data['Conversion Rate'] /100)
```

In [181]:
```
#conversion Rate = (ads_data['Conversions'] / ads_data['Clicks']) * 100
```

In [182]:
```
column_order = [
    'Age', 'Gender', 'Income', 'Location', 'Ad Type', 'Ad Topic',
    'Ad Placement','Date','Time', 'Clicks',  'Conversions', 'Conversion Rate', 'CTR'
]

ads_data = ads_data[column_order]
```

In [183]:
```
ads_data.sample(5)
```

Out[183]:

	Age	Gender	Income	Location	Ad Type	Ad Topic	Ad Placement	Date	Time	Clicks	Conversions
9140	30	Female	68663.22	Suburban	Native	Health	Website	2023-09-28	1900-01-01 20:45:56	5	0.017170
4824	58	Male	118241.39	Urban	Banner	Food	Social Media	2023-08-28	1900-01-01 20:45:56	6	0.010176
9126	54	Other	18948.64	Suburban	Video	Food	Website	2023-04-24	1900-01-01 20:45:56	7	0.013566
854	19	Male	28861.12	Suburban	Native	Food	Search Engine	2023-12-23	1900-01-01 20:45:56	9	0.037710
2563	15	Female	40341.99	Urban	Text	Finance	Website	2023-08-27	1900-01-01 20:45:56	7	0.001127

In [184]:
```
ads_data.to_csv('ads_data_cleaned.csv' , index=False) #saving without the index
```

DATA - ANALYSIS

In [185]:
```
#display basic information about the data set
ads_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9778 entries, 0 to 9999
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Age              9778 non-null   int64
 1   Gender           9778 non-null   object
 2   Income           9778 non-null   float64
 3   Location         9778 non-null   object
 4   Ad Type          9778 non-null   object
 5   Ad Topic         9778 non-null   object
 6   Ad Placement     9778 non-null   object
 7   Date             9778 non-null   object
 8   Time             9778 non-null   datetime64[ns]
 9   Clicks           9778 non-null   int64
 10  Conversions      9778 non-null   float64
 11  Conversion Rate  9778 non-null   float64
 12  CTR              9778 non-null   float64
dtypes: datetime64[ns](1), float64(4), int64(2), object(6)
memory usage: 1.0+ MB
```

Exploratory DATA ANALYSIS

In [197]:
```
#Desscriptive statistics for numeric columns

ads_data.describe().T
```

Out[197]:

	count	mean	std	min	25%	50%	75%	
Age	9778.0	35.029965	13.966552	6.000	25.000000	35.000000	44.000000	92
Income	9778.0	50162.215033	19765.870171	78.950	36785.025000	50139.810000	63321.182500	126635
Clicks	9778.0	5.023829	2.259874	0.000	3.000000	5.000000	6.000000	17
Conversions	9778.0	0.010208	0.008231	0.000	0.004267	0.008064	0.013899	0
Conversion Rate	9778.0	0.202447	0.121511	0.001	0.109000	0.180800	0.275400	0
CTR	9778.0	0.050422	0.019870	0.000	0.037100	0.050400	0.063800	0

1. The average CTR for all ads during the campagain period is 5%

2. The std CTR for all ads during the campagain period is 1.98%

3. The MAX CTR for all ads during the campagain period is 12.7%

In [186]:
```
#Descriptive Statistics categorical columns
ads_data.describe(include='object')
```

Out[186]:

	Gender	Location	Ad Type	Ad Topic	Ad Placement	Date
count	9778	9778	9778	9778	9778	9778
unique	3	3	4	6	3	364
top	Male	Rural	Banner	Finance	Website	2023-06-12
freq	4861	3317	2514	1691	3270	44

1. The most common gender intrested with the ads is Male, in 4,861 ads.

2. The m0st of ads were intrested to users in Rural areas, in 3,317 ads.

3. The most used ad type is Banner ads, which were used in 2,514 ads.

4. The most used adTopic is Finance ads, which were used in 1691 ads.

5. The most used ad placement is Website, which were used in 2,514 ads.

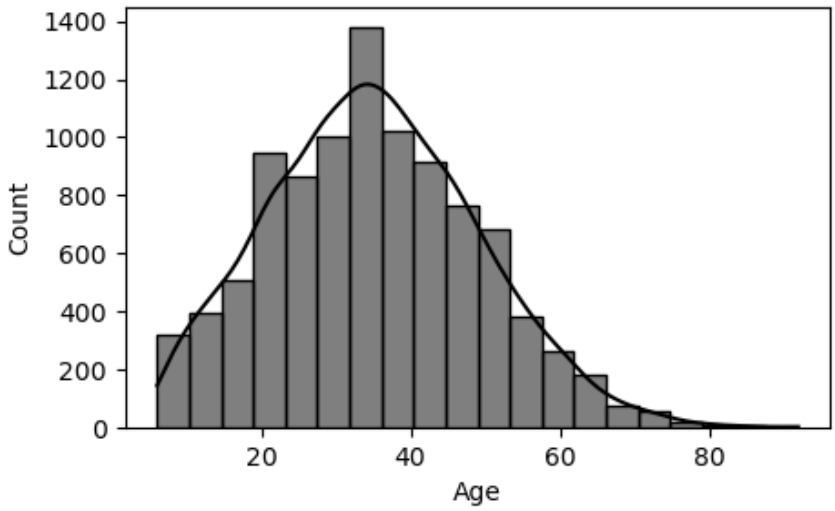
DATA VISUALIZATION

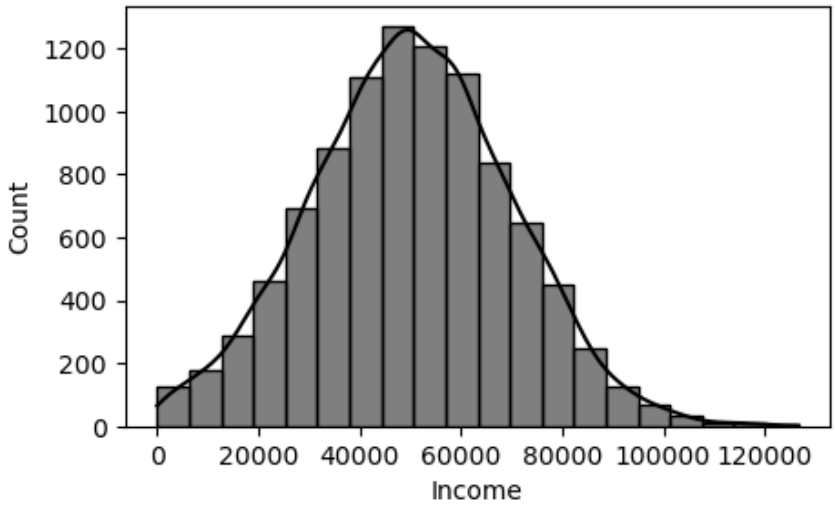
Visualizing the Distribution of Conversion Rate and CTR

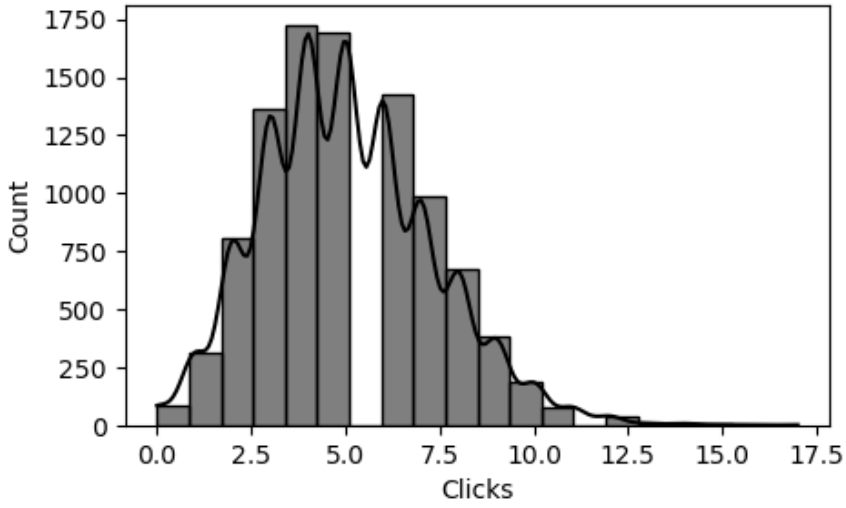
To understand the distribution of Conversion rate and CTR we can plot histograms

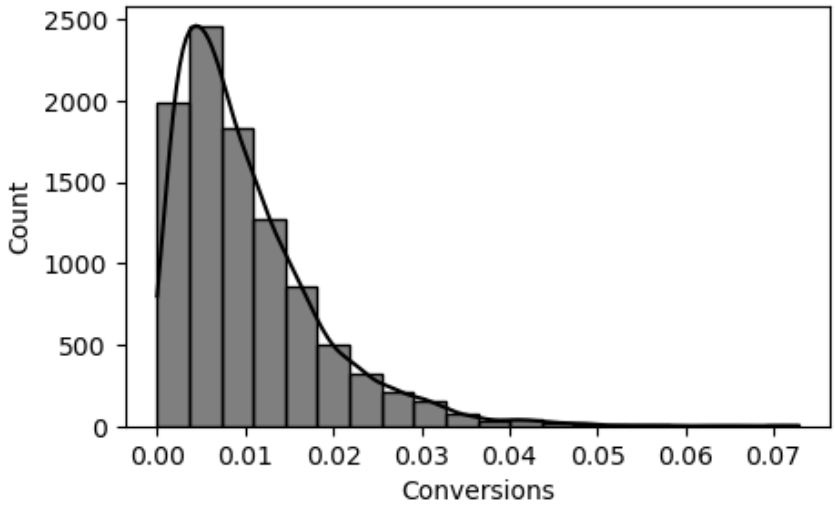
UNIVARIANT ANALYSIS

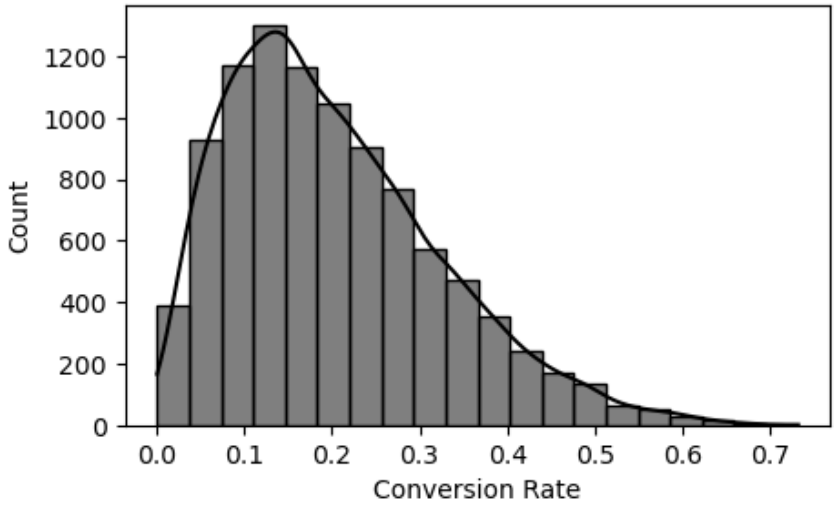
In [189]:
```
#histogram to understand the distribution
import matplotlib.pyplot as plt
import seaborn as sns
for i in ads_data.select_dtypes(include='number').columns:
    plt.figure(figsize=(5,3))
    sns.histplot(data=ads_data,x=i , bins=20 , kde=True,color='Black')
    plt.show()
```

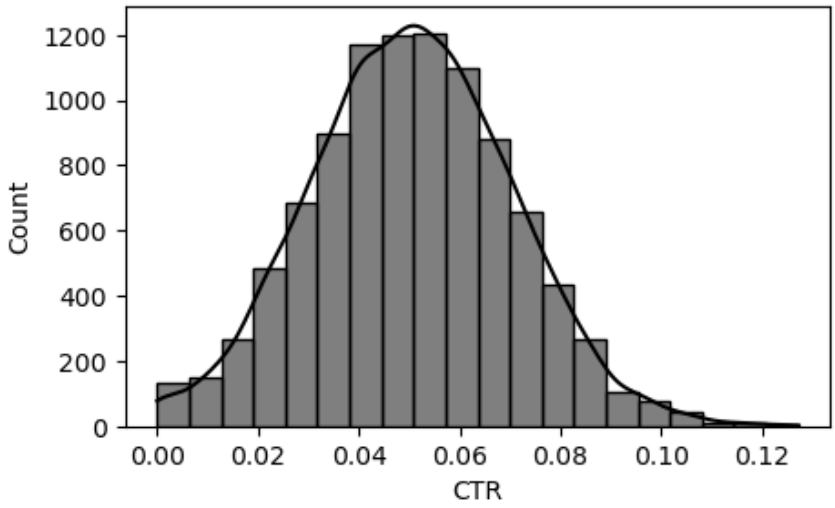




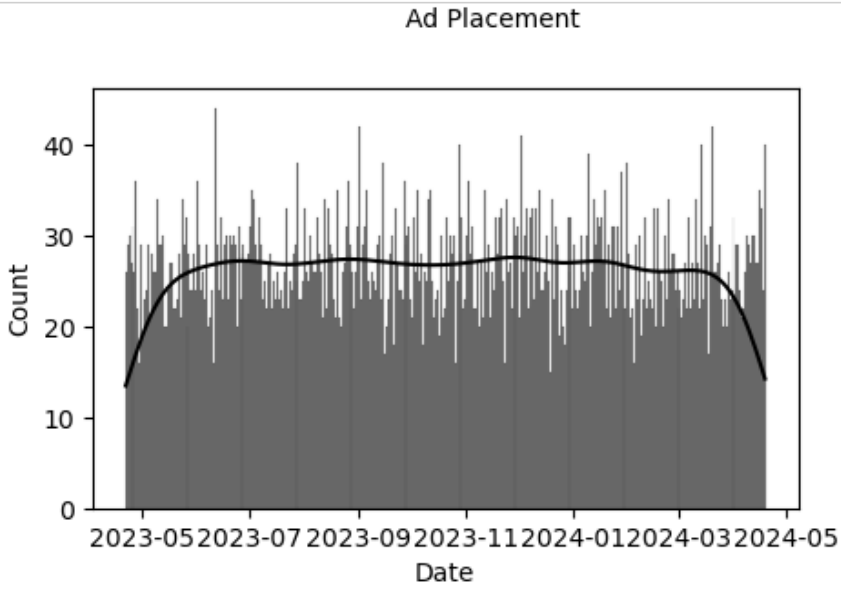



2500
2000
1500
1000
500
0
Count
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Conversions


1200
1000
800
600
400
200
0
Count
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Conversion Rate


1200
1000
800
600
400
200
0
Count
0.00
0.02
0.04
0.06
0.08
0.10
0.12
CTR

In [190]:
```
#histogram to understand the distribution for categorical
import matplotlib.pyplot as plt
import seaborn as sns
for i in ads_data.select_dtypes(include='object').columns:
    plt.figure(figsize=(5,3))
    sns.histplot(data=ads_data,x=i , bins=20,kde=True , color='black')
    plt.show()
```

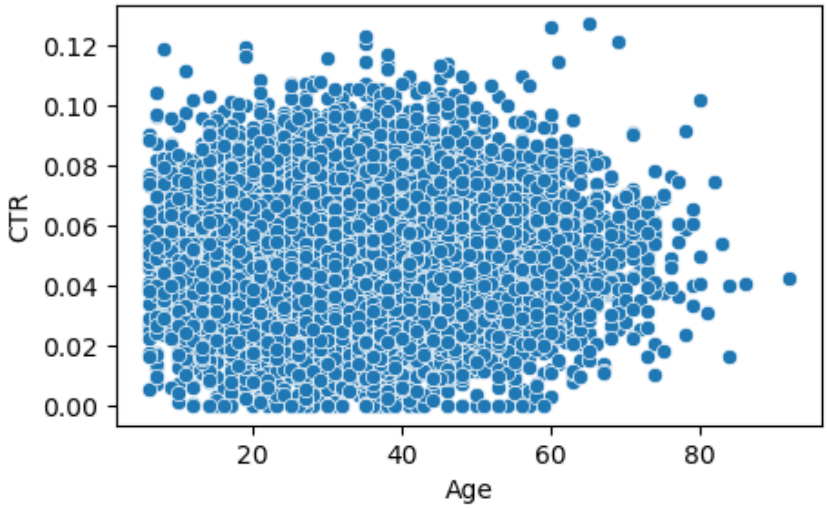


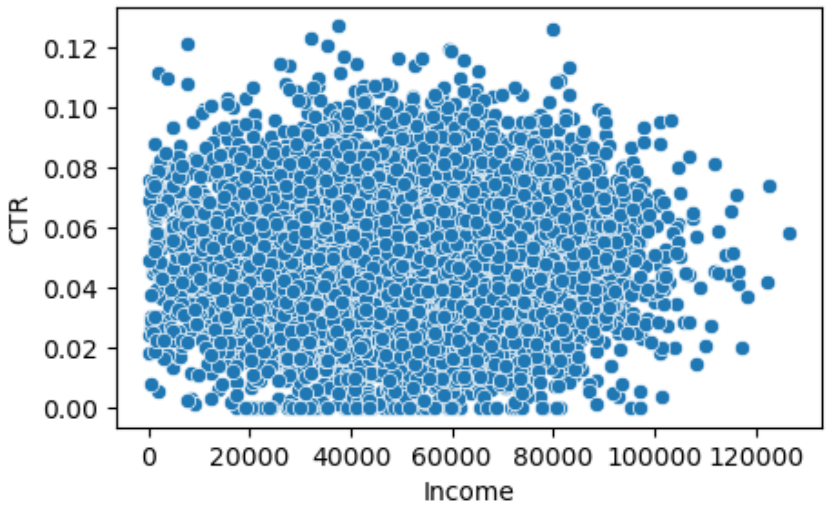
In [210]:
```
ads_data.select_dtypes(include='number').columns
```

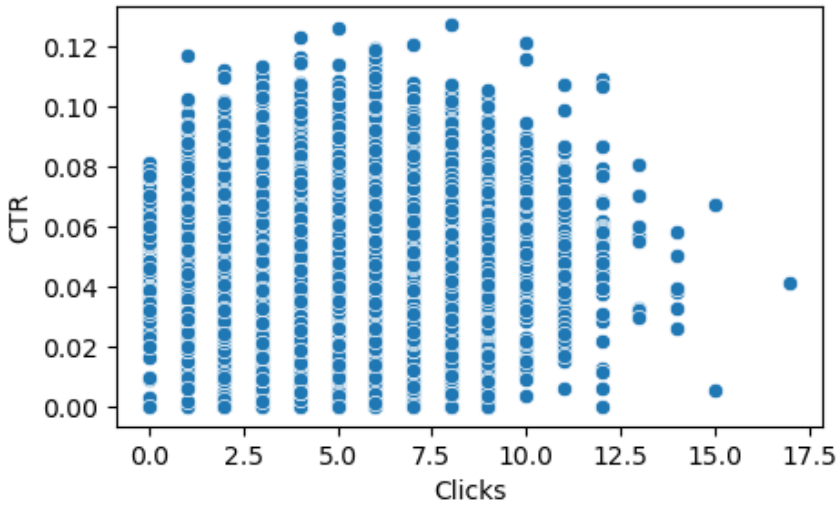
Out[210]:
```
Index(['Age', 'Income', 'Clicks', 'Conversions', 'Conversion Rate', 'CTR'], dtype='obj
ect')
```

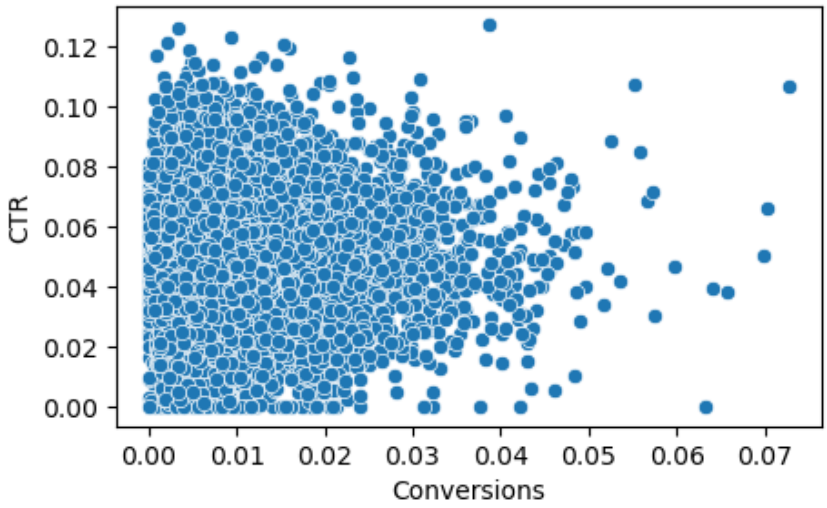
BIVARIANT ANALYSIS

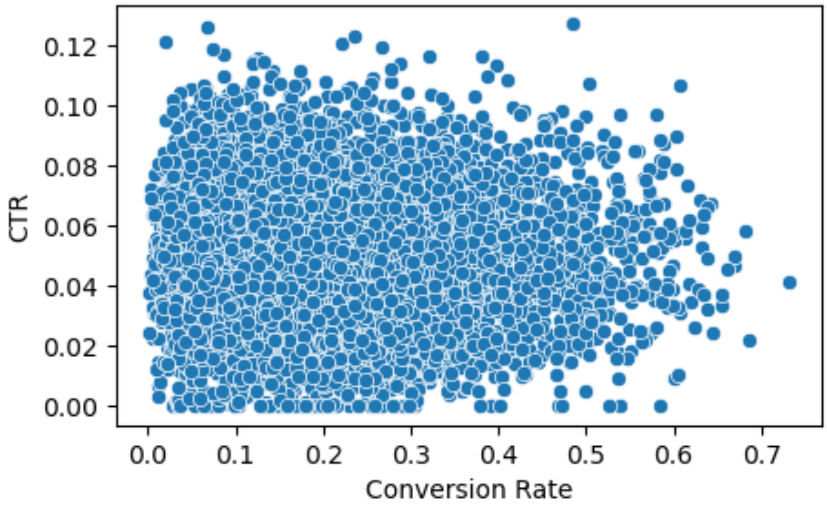
In [214]:
```
#Scatter plot to understand the relationship between num-num

for i in ['Age', 'Income', 'Clicks', 'Conversions', 'Conversion Rate']:
    plt.figure(figsize=(5,3))
    sns.scatterplot(data=ads_data , x=i , y='CTR')
    plt.show()
```



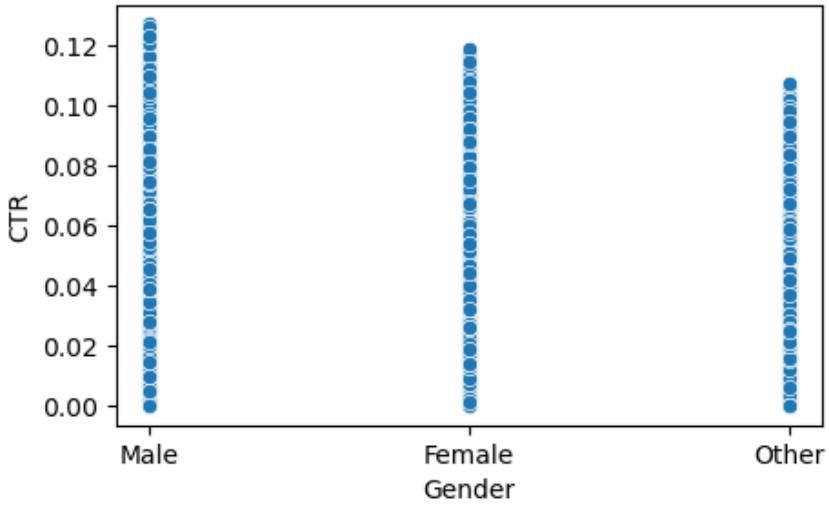


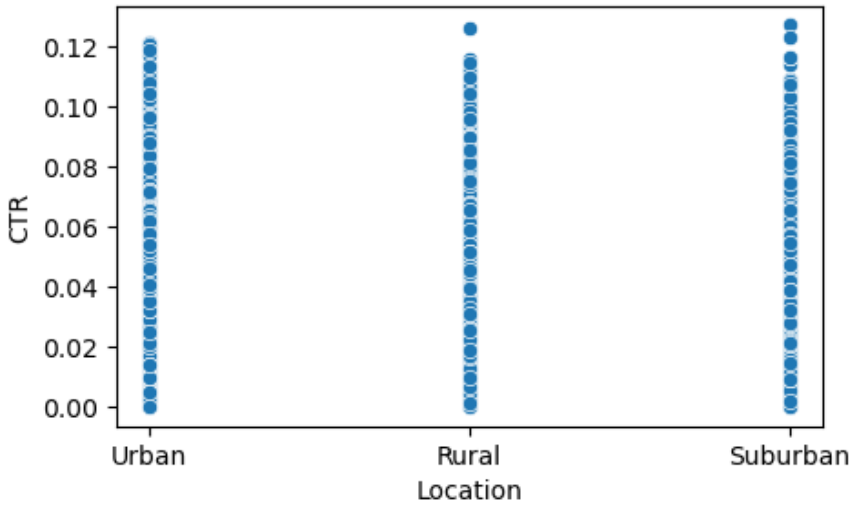


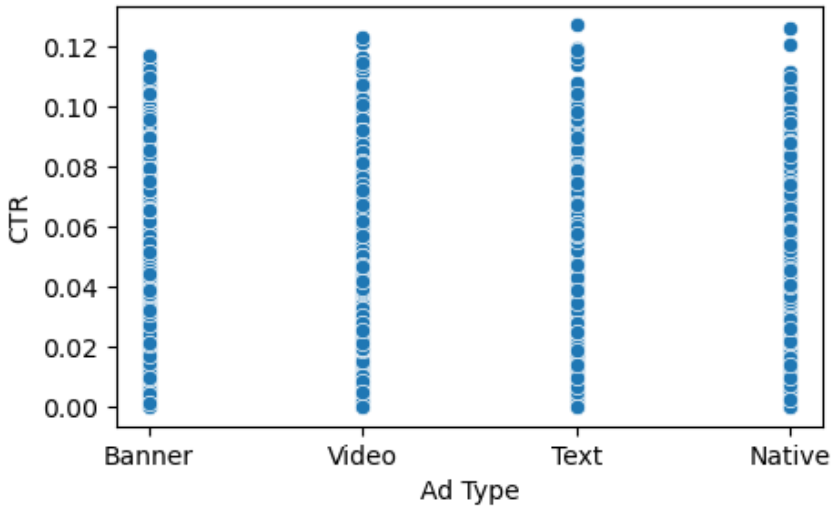

CTR
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Conversions

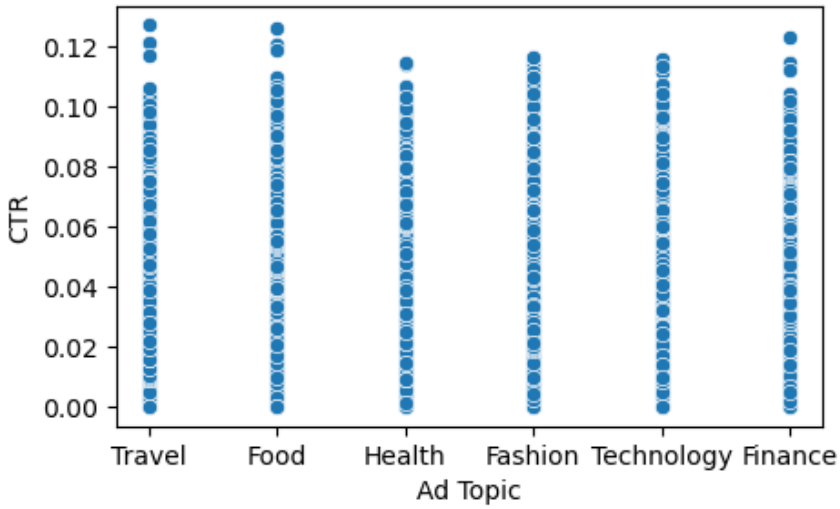

CTR
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Conversion Rate

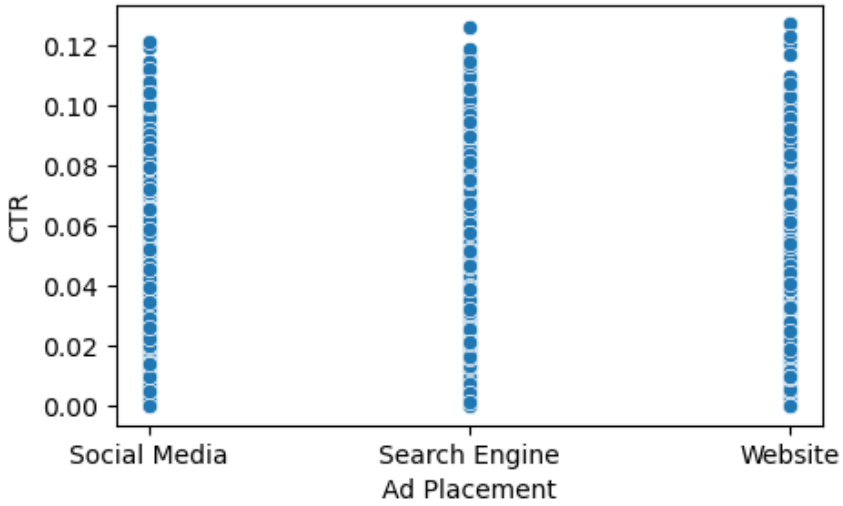
In [191]:
```
# for categorical versus Numerical

for i in ['Gender', 'Location', 'Ad Type', 'Ad Topic', 'Ad Placement', 'Date'
       ]:
    plt.figure(figsize=(5,3))
    sns.scatterplot(data=ads_data , x = i , y='CTR')
    plt.show()
```

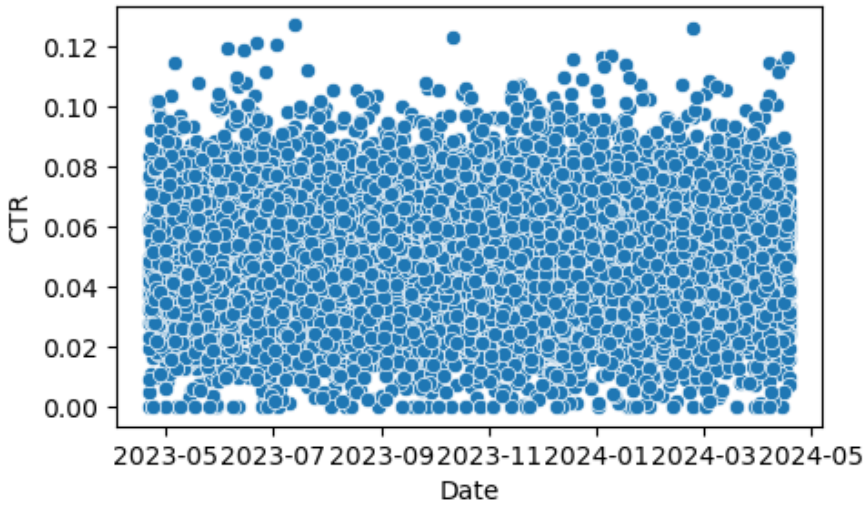


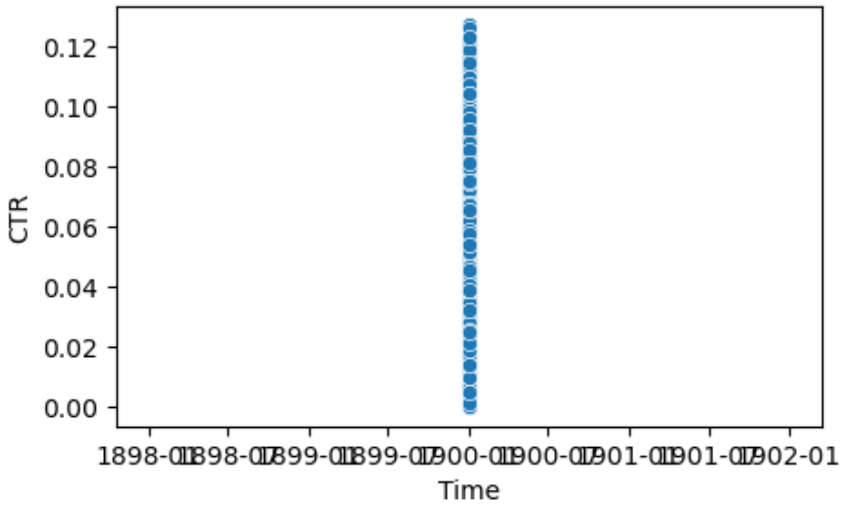




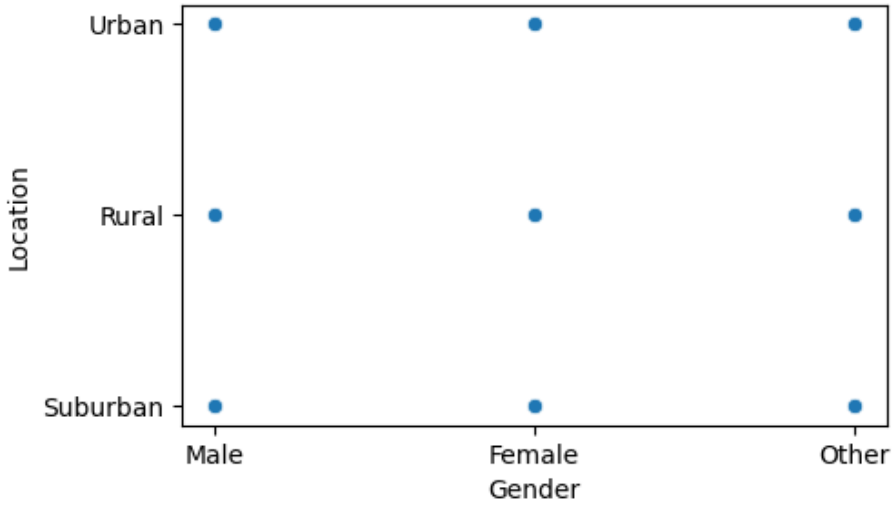

CTR
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Travel
Food
Health
Fashion
Technology
Finance
Ad Topic

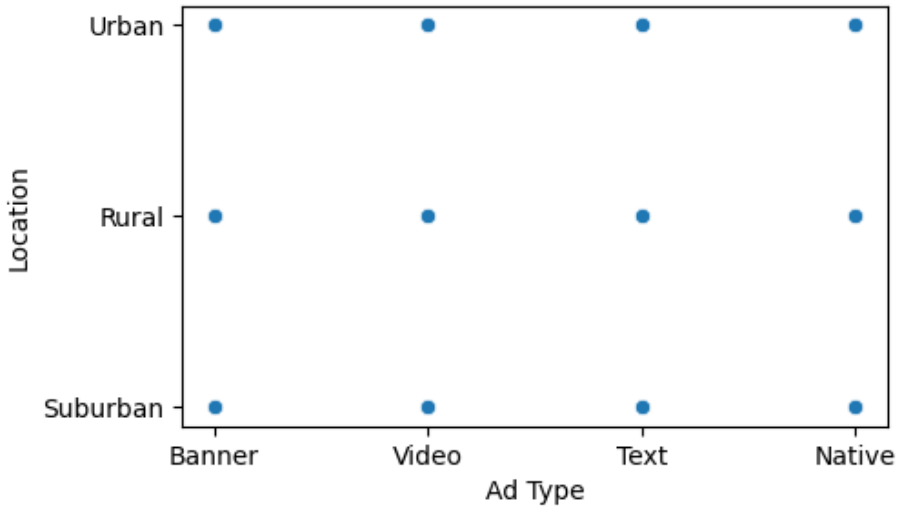

CTR
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Social Media
Search Engine
Website
Ad Placement

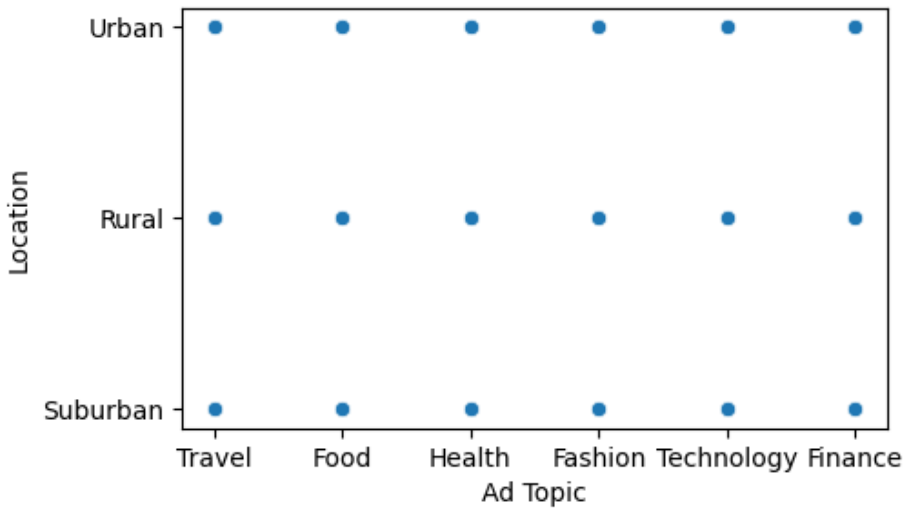

CTR
0.12
0.10
0.08
0.06
0.04
0.02
0.00
2023-05
2023-07
2023-09
2023-11
2024-01
2024-03
2024-05
Date

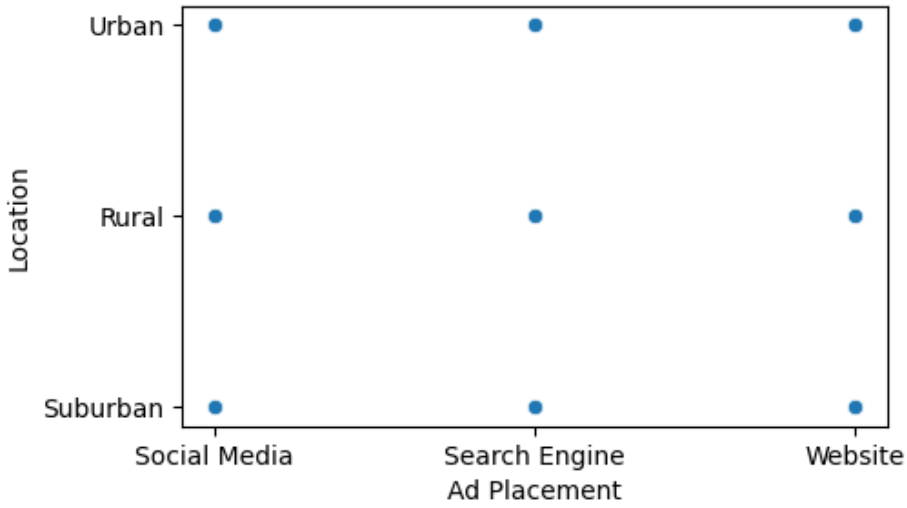

0.12
0.10
0.08
0.06
0.04
0.02
0.00
CTR
Time

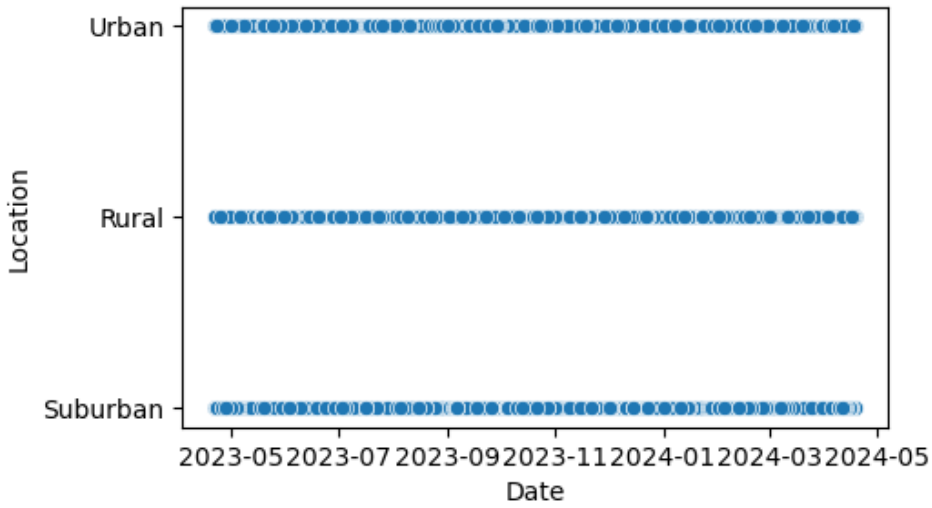
In [192]:
```
# categorical versus categorical
for i in ['Gender',  'Ad Type', 'Ad Topic', 'Ad Placement', 'Date'
       ]:
    plt.figure(figsize=(5,3))
    sns.scatterplot(data=ads_data , x = i , y='Location')
    plt.show()
```











Correlation Analysis

To understand the relationships between Conversion Rate, CTR, and other columns such as Age, Income, and Clicks,conversions we can perform a correlation analysis.

+1: Perfect positive correlation (i.e., as one variable increases, the other also increases).

-1: Perfect negative correlation (i.e., as one variable increases, the other decreases).

0: No correlation (i.e., the two variables are independent).

In [219]:
```
#Calculate the corelation Matrix
corr = ads_data.select_dtypes(include='number').corr()
```

In [220]:
```
# plotting the corelation HeatMap
plt.figure(figsize=(10,3))
sns.heatmap(corr, annot=True , cmap='coolwarm' , fmt = '.2f' ,linewidths=0.5)
plt.title('corelation_matrix')
plt.show()
```