In [ ]:
```
import pandas as pd
import numpy as np

data = pd.read_csv("/Users/jatin/Documents/Finlatics/DsResearch/Digital M
df = pd.DataFrame(data)
print(df)
```

```
       month  day campaign_number user_engagement     banner placement  \
0      April    1          camp 1            High  160 x 600       abc
1      April    1          camp 1            High  160 x 600       def
2      April    1          camp 1            High  160 x 600       ghi
3      April    1          camp 1            High  160 x 600       mno
4      April    1          camp 1             Low  160 x 600       def
...      ...  ...             ...             ...        ...       ...
15403  April    1          camp 1             Low  160 x 600       ghi
15404  April    1          camp 1             Low  160 x 600       mno
15405   June   29          camp 1            High  800 x 250       ghi
15406   June   29          camp 1            High  800 x 250       mno
15407   June   29          camp 3            High  240 x 400       def

       displays      cost  clicks   revenue  post_click_conversions  \
0             4    0.0060       0    0.0000                       0
1         20170   26.7824     158   28.9717                      23
2         14701   27.6304     158   28.9771                      78
3        171259  216.8750    1796  329.4518                     617
4           552    0.0670       1    0.1834                       0
...         ...       ...     ...       ...                     ...
15403        16    0.0249       0    0.0000                       0
15404      2234    0.4044      10    1.8347                       3
15405         1    0.0157       0    0.0000                       0
15406         4    0.0123       0    0.0000                       0
15407      1209    0.3184       2    0.1115                       3

       post_click_sales_amount  Unnamed: 12  Unnamed: 13
0                       0.0000          NaN          NaN
1                    1972.4602          NaN          NaN
2                    2497.2636          NaN          NaN
3                   24625.3234          NaN          NaN
4                       0.0000          NaN          NaN
...                        ...          ...          ...
15403                   0.0000          NaN          NaN
15404                 101.7494          NaN          NaN
15405                   0.0000          NaN          NaN
15406                   0.0000          NaN          NaN
15407                 110.4224          NaN          NaN

[15408 rows x 14 columns]
```

In [ ]:
```
print(df.info(), "\n")
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15408 entries, 0 to 15407
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   month                    15408 non-null  object 
 1   day                      15408 non-null  int64  
 2   campaign_number          15408 non-null  object 
 3   user_engagement          15408 non-null  object 
 4   banner                   15408 non-null  object 
 5   placement                14995 non-null  object 
 6   displays                 15408 non-null  int64  
 7   cost                     15408 non-null  float64
 8   clicks                   15408 non-null  int64  
 9   revenue                  15408 non-null  float64
 10  post_click_conversions   15408 non-null  int64  
 11  post_click_sales_amount  15408 non-null  float64
 12  Unnamed: 12              0 non-null      float64
 13  Unnamed: 13              0 non-null      float64
dtypes: float64(5), int64(4), object(5)
memory usage: 1.6+ MB
None
```

	day	displays	cost	clicks	revenue
count	15408.000000	15408.000000	15408.000000	15408.000000	15408.000000
mean	15.518886	15512.573014	11.370262	161.788487	17.929943
std	8.740909	44392.392890	45.369499	728.276911	96.781834
min	1.000000	0.000000	0.000000	0.000000	0.000000
25%	8.000000	78.000000	0.024000	0.000000	0.000000
50%	15.000000	1182.000000	0.339850	6.000000	0.483950
75%	23.000000	8960.250000	2.536225	53.000000	3.839800
max	31.000000	455986.000000	556.704800	14566.000000	2096.211600

	post_click_conversions	post_click_sales_amount	Unnamed: 12	Unnamed: 13
count	15408.000000	15408.000000	0.0	0.0
mean	42.300623	2123.288058	NaN	NaN
std	213.685660	10523.029607	NaN	NaN
min	0.000000	0.000000	NaN	NaN
25%	0.000000	0.000000	NaN	NaN
50%	0.000000	0.000000	NaN	NaN
75%	3.000000	163.351200	NaN	NaN
max	3369.000000	199930.318000	NaN	NaN

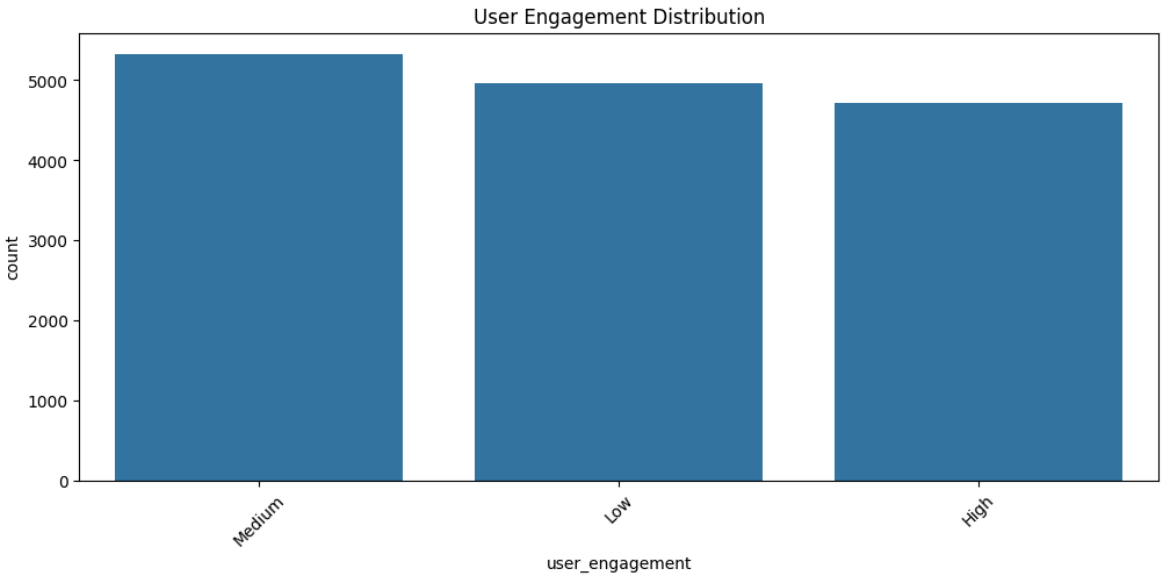
```
75%           NaN
max           NaN
```

In [ ]:
```
engagement_unique_count = df["user_engagement"].value_counts()
print(engagement_unique_count)
```

```
user_engagement
Medium    5489
Low       5035
High      4884
Name: count, dtype: int64
```

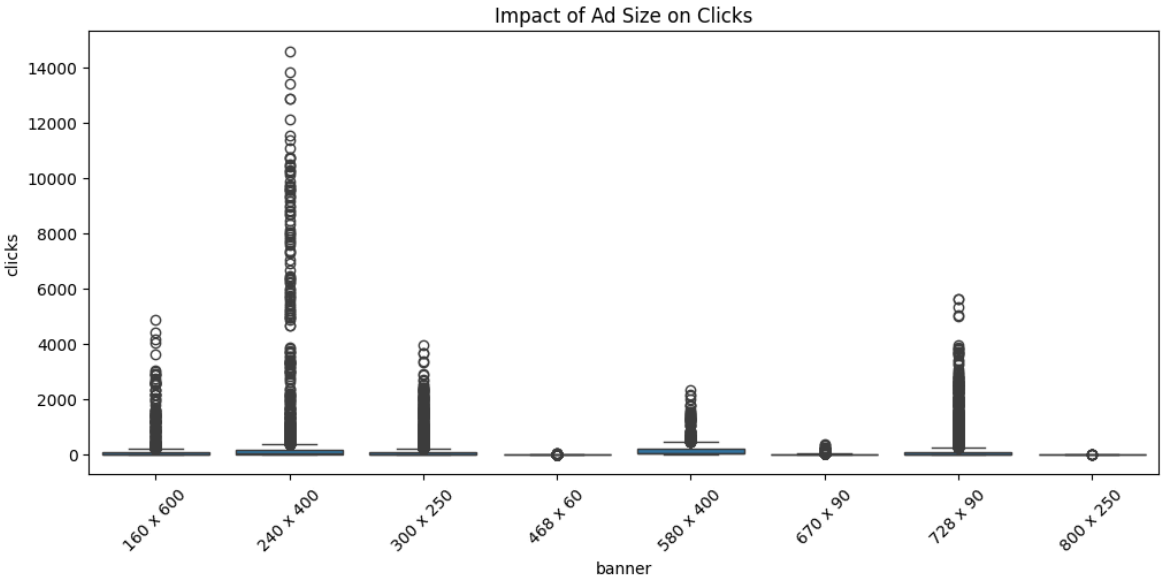
Q1

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

df = df.drop(columns=["Unnamed: 12", "Unnamed: 13"], errors="ignore")

df = df.dropna(subset=["placement"])

df["date"] = pd.to_datetime(df["month"] + " " + df["day"].astype(str) + "

df = df.drop(columns=["month", "day"])

plt.figure(figsize=(12,5))
sns.countplot(data=df, x="user_engagement", order=df["user_engagement"].v
plt.title("User Engagement Distribution")
plt.xticks(rotation=45)
plt.show()
```



Q2

In [ ]:
```
plt.figure(figsize=(12,5))
sns.boxplot(data=df, x="banner", y="clicks")
plt.title("Impact of Ad Size on Clicks")
plt.xticks(rotation=45)
plt.show()
```



Q3

In [ ]:
```
placement_stats = df.groupby("placement")[["displays", "clicks"]].sum().s
print(placement_stats.head())
```

```
           displays   clicks
placement
mno        143161775   993039
ghi         59740415  1247049
def         28177492   176097
jkl          7692732    75063
abc           242142     1584
```

Q4

In [ ]:
```
corr = df[["cost", "revenue"]].corr()
print("Correlation between cost and revenue:\n", corr)
```

```
Correlation between cost and revenue:
              cost   revenue
cost     1.000000  0.760258
revenue  0.760258  1.000000
```

Q5

In [ ]:
```
df["revenue_per_click"] = df["revenue"] / df["clicks"].replace(0, np.nan)
print("Average Revenue per Click:", df["revenue_per_click"].mean())
```

```
Average Revenue per Click: 0.09294329678211666
```

Q6

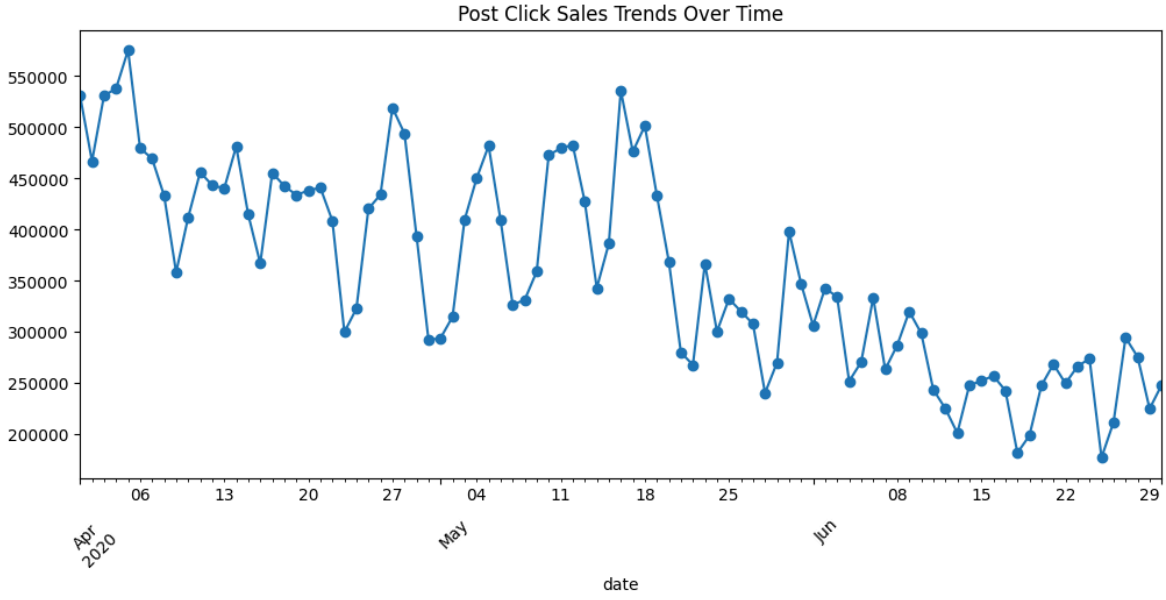
In [ ]:
```
df["conversion_rate"] = df["post_click_conversions"] / df["clicks"].repla
campaign_conversion = df.groupby("campaign_number")["conversion_rate"].me
```

```
print("Top Campaigns by Conversion Rate:\n", campaign_conversion.head())
```

```
Top Campaigns by Conversion Rate:
 campaign_number
camp 1    0.357042
camp 3    0.045473
camp 2    0.020111
Name: conversion_rate, dtype: float64
```

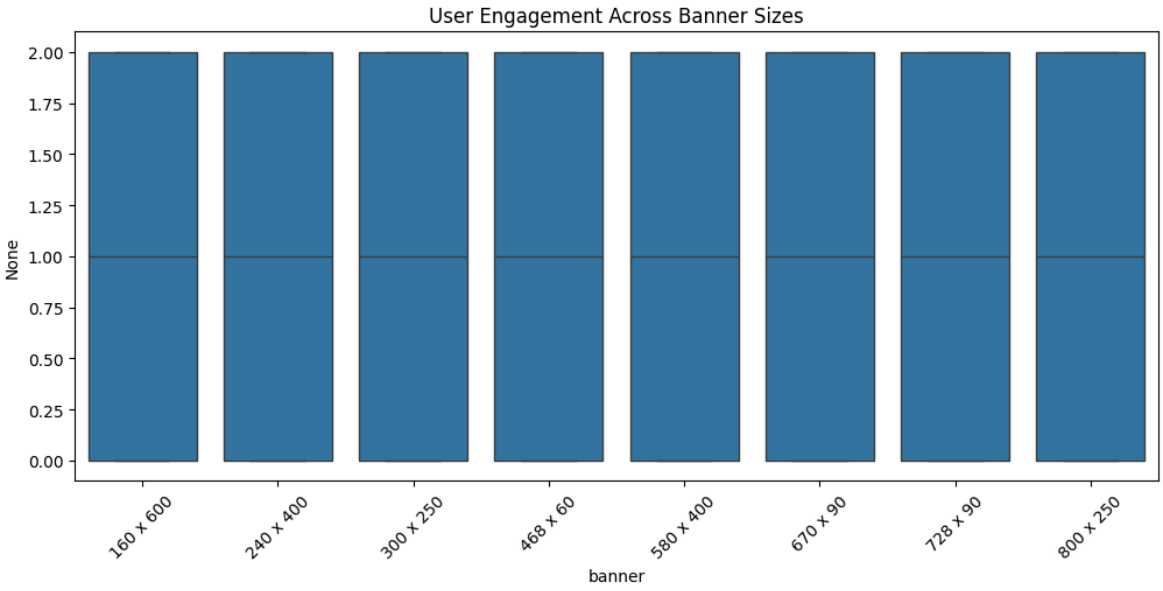
Q7

In [ ]:
```
plt.figure(figsize=(12,5))
df.groupby("date")["post_click_sales_amount"].sum().plot(marker="o")
plt.title("Post Click Sales Trends Over Time")
plt.xticks(rotation=45)
plt.show()
```



Q8

In [ ]:
```
plt.figure(figsize=(12,5))
sns.boxplot(data=df, x="banner", y=df["user_engagement"].astype("category
plt.title("User Engagement Across Banner Sizes")
plt.xticks(rotation=45)
plt.show()
```



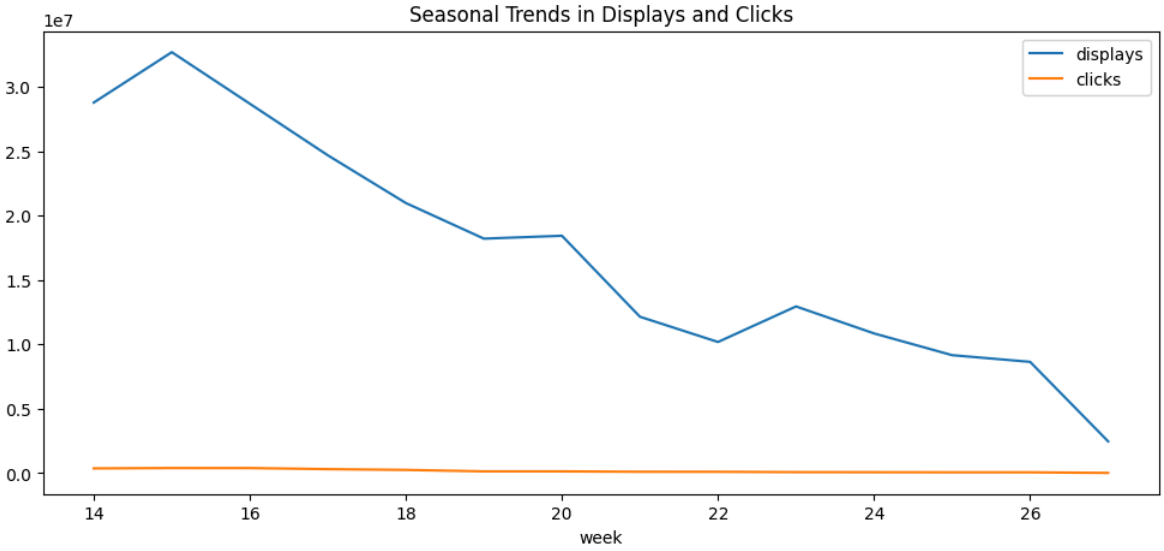
Q9

In [ ]:
```
placement_conversion = df.groupby("placement")["conversion_rate"].mean().
print("Top Placement Types by Conversion Rate:\n", placement_conversion.h
```

```
Top Placement Types by Conversion Rate:
 placement
abc    0.301971
jkl    0.224332
ghi    0.187649
mno    0.182342
def    0.152955
Name: conversion_rate, dtype: float64
```

Q10

In [ ]:
```
df["week"] = df["date"].dt.isocalendar().week
weekly_stats = df.groupby("week")[["displays", "clicks"]].sum()
weekly_stats.plot(kind="line", figsize=(12,5))
plt.title("Seasonal Trends in Displays and Clicks")
plt.show()
```



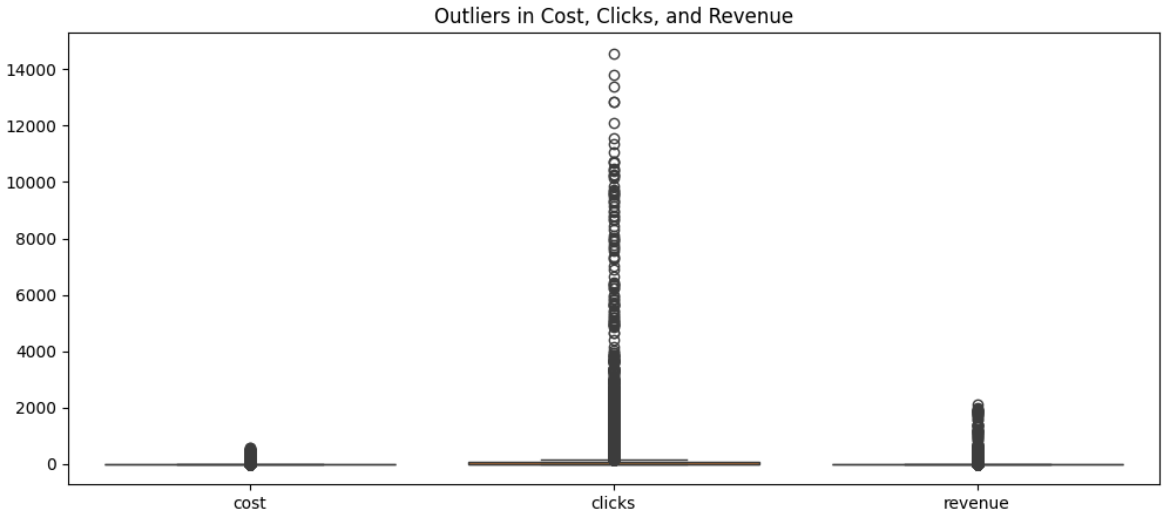
Q11

In [ ]:
```
engagement_revenue_corr = df.groupby("user_engagement")["revenue"].mean()
print("Revenue by User Engagement:\n", engagement_revenue_corr)
```

```
Revenue by User Engagement:
 user_engagement
High      45.076201
Medium    11.048293
Low        1.037745
Name: revenue, dtype: float64
```

Q12

In [ ]:
```
plt.figure(figsize=(12,5))
sns.boxplot(data=df[["cost", "clicks", "revenue"]])
plt.title("Outliers in Cost, Clicks, and Revenue")
plt.show()
```



Q13

In [ ]:
```
effectiveness = df.groupby(["banner", "placement"])[["clicks", "post_clic
print("Campaign Effectiveness:\n", effectiveness.head())
```

```
Campaign Effectiveness:
                      clicks  post_click_conversions
banner    placement
160 x 600 abc              3                       0
          def          20257                    2525
          ghi           9799                    4021
          jkl              0                       0
          mno         209511                   42239
```

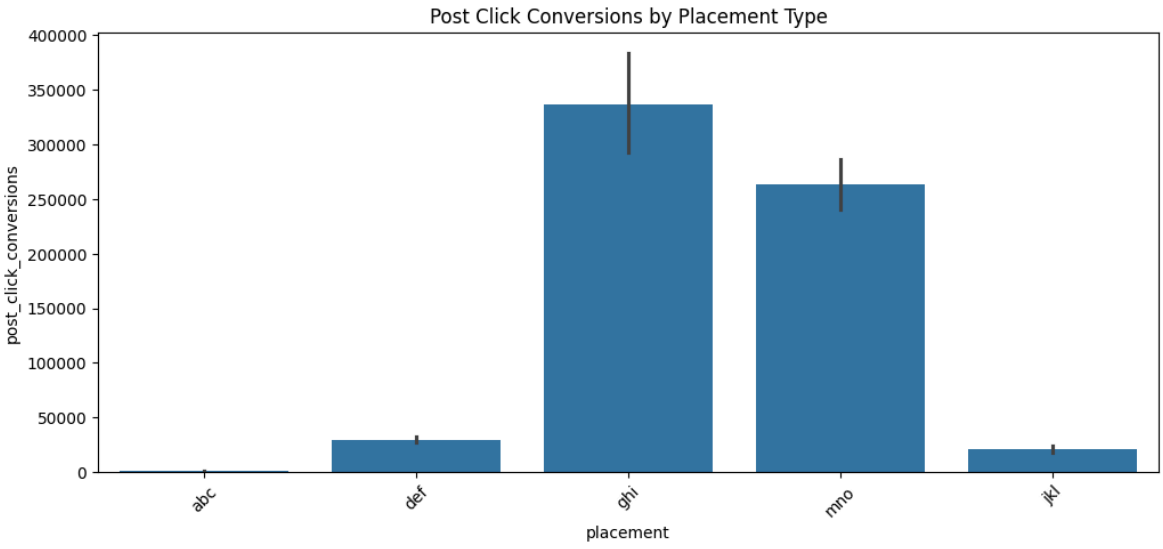
Q14

In [ ]:
```
df["ROI"] = df["revenue"] / df["cost"].replace(0, np.nan)
roi_stats = df.groupby("campaign_number")["ROI"].mean().sort_values(ascen
print("Top Campaigns by ROI:\n", roi_stats.head())
```

```
Top Campaigns by ROI:
 campaign_number
camp 1    2.189996
camp 3    1.822633
camp 2    1.643087
Name: ROI, dtype: float64
```

Q15

In [ ]:
```
plt.figure(figsize=(12,5))
sns.barplot(data=df, x="placement", y="post_click_conversions", estimator
plt.title("Post Click Conversions by Placement Type")
plt.xticks(rotation=45)
plt.show()
```



Q16

In [ ]:
```
df["weekday"] = df["date"].dt.weekday
weekday_engagement = df.groupby("weekday")["user_engagement"].value_count
```

```
print("User Engagement by Weekday:\n", weekday_engagement)
```

```
User Engagement by Weekday:
 user_engagement  High  Low  Medium
weekday
0                 677  704     754
1                 670  703     757
2                 665  731     780
3                 669  711     768
4                 672  694     748
5                 679  703     757
6                 677  713     763
```

Q17

In [ ]:
```
df["CPC"] = df["cost"] / df["clicks"].replace(0, np.nan)
cpc_stats = df.groupby(["campaign_number", "banner"])["CPC"].mean().sort_
print("Campaigns with Lowest CPC:\n", cpc_stats.head())
```

```
Campaigns with Lowest CPC:
 campaign_number  banner
camp 3           800 x 250    0.000500
camp 2           800 x 250    0.007000
camp 3           468 x 60     0.010349
camp 2           580 x 400    0.013092
                 240 x 400    0.023859
Name: CPC, dtype: float64
```

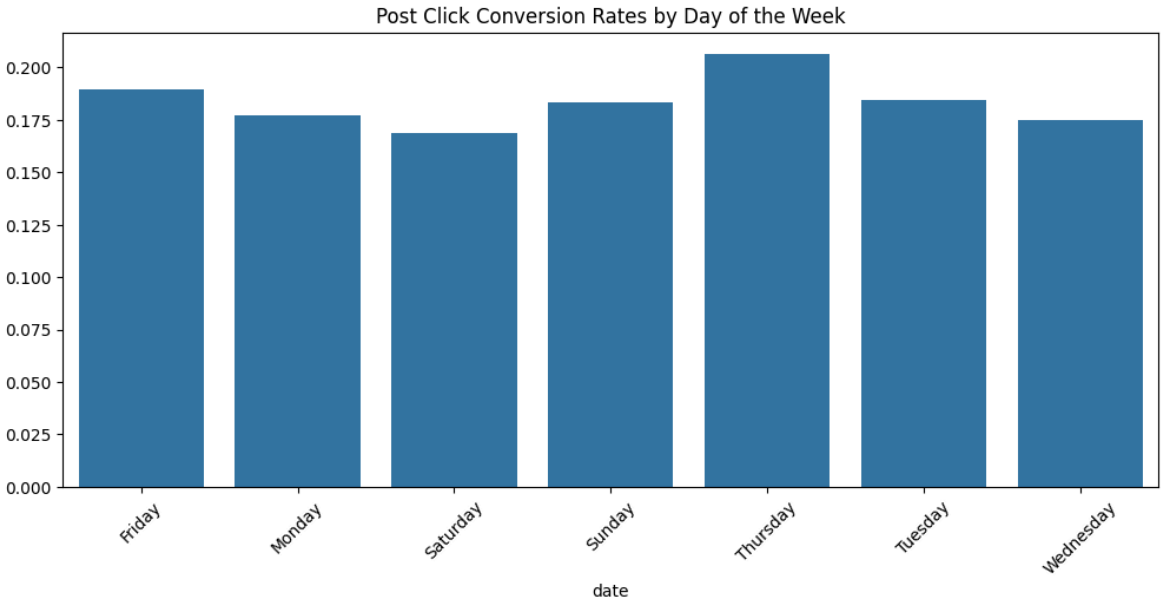
Q18

In [ ]:
```
df["cost_per_conversion"] = df["cost"] / df["post_click_conversions"].rep
cost_effective = df.groupby(["campaign_number", "placement"])["cost_per_c
print("Most Cost-Effective Campaigns for Conversions:\n", cost_effective.
```

```
Most Cost-Effective Campaigns for Conversions:
 campaign_number  placement
camp 1           abc          0.163195
                 jkl          0.263877
camp 3           abc          0.317597
camp 1           ghi          0.325304
                 mno          0.474666
Name: cost_per_conversion, dtype: float64
```

Q19

In [ ]:
```
daily_conversion = df.groupby(df["date"].dt.day_name())["conversion_rate"
plt.figure(figsize=(12,5))
sns.barplot(x=daily_conversion.index, y=daily_conversion.values)
plt.title("Post Click Conversion Rates by Day of the Week")
plt.xticks(rotation=45)
plt.show()
```



Q20

In [ ]:
```
engagement_conversion = df.groupby("user_engagement")["conversion_rate"].
print("Conversion Rates by User Engagement:\n", engagement_conversion)
```

```
Conversion Rates by User Engagement:
 user_engagement
High      0.425424
Medium    0.073980
Low       0.049346
Name: conversion_rate, dtype: float64
```