In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [2]:
```
data = pd.read_csv('online_advertising_performance_data.csv')
data.head(n=10)
```

Out[2]:

| | month | day | campaign_number | user_engagement | banner | placement | displays | cost | clicks | revenue | post_click_conversi |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | April | 1 | camp 1 | High | 160 x 600 | abc | 4 | 0.0060 | 0 | 0.0000 | |
| **1** | April | 1 | camp 1 | High | 160 x 600 | def | 20170 | 26.7824 | 158 | 28.9717 | |
| **2** | April | 1 | camp 1 | High | 160 x 600 | ghi | 14701 | 27.6304 | 158 | 28.9771 | |
| **3** | April | 1 | camp 1 | High | 160 x 600 | mno | 171259 | 216.8750 | 1796 | 329.4518 | |
| **4** | April | 1 | camp 1 | Low | 160 x 600 | def | 552 | 0.0670 | 1 | 0.1834 | |
| **5** | April | 1 | camp 1 | Low | 160 x 600 | ghi | 16 | 0.0249 | 0 | 0.0000 | |
| **6** | April | 1 | camp 1 | Low | 160 x 600 | mno | 2234 | 0.4044 | 10 | 1.8347 | |
| **7** | April | 1 | camp 1 | Medium | 160 x 600 | def | 2963 | 1.8899 | 4 | 0.7338 | |
| **8** | April | 1 | camp 1 | Medium | 160 x 600 | ghi | 580 | 0.9917 | 9 | 1.6512 | |
| **9** | April | 1 | camp 1 | Medium | 160 x 600 | mno | 20152 | 11.1678 | 185 | 33.9397 | |

In [3]:
```
data.columns
```

Out[3]:
```
Index(['month', 'day', 'campaign_number', 'user_engagement', 'banner',
       'placement', 'displays', 'cost', 'clicks', 'revenue',
       'post_click_conversions', 'post_click_sales_amount', 'Unnamed: 12',
       'Unnamed: 13'],
      dtype='object')
```

In [4]:
```
data.info()
```

```
<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
```

In [5]:
```
data.describe()
```

Out[5]:

| | day | displays | cost | clicks | revenue | post_click_conversions | post_click_sales_amount |
|---|---|---|---|---|---|---|---|
| **count** | 15408.000000 | 15408.000000 | 15408.000000 | 15408.000000 | 15408.000000 | 15408.000000 | 15408.000000 |
| **mean** | 15.518886 | 15512.573014 | 11.370262 | 161.788487 | 17.929943 | 42.300623 | 2123.288058 |
| **std** | 8.740909 | 44392.392890 | 45.369499 | 728.276911 | 96.781834 | 213.685660 | 10523.029607 |
| **min** | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 8.000000 | 78.000000 | 0.024000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 15.000000 | 1182.000000 | 0.339850 | 6.000000 | 0.483950 | 0.000000 | 0.000000 |
| **75%** | 23.000000 | 8960.250000 | 2.536225 | 53.000000 | 3.839800 | 3.000000 | 163.351200 |
| **max** | 31.000000 | 455986.000000 | 556.704800 | 14566.000000 | 2096.211600 | 3369.000000 | 199930.318000 |

In [6]:
```
data.isna().any()
```

Out[6]:
```
month                    False
day                      False
campaign_number          False
user_engagement          False
banner                   False
placement                 True
displays                 False
cost                     False
clicks                   False
revenue                  False
post_click_conversions   False
post_click_sales_amount  False
Unnamed: 12               True
Unnamed: 13               True
dtype: bool
```

In [7]:
```
data.drop(['Unnamed: 12','Unnamed: 13'],axis =1,inplace = True)
```

In [8]:
```
data.head(n=6)
```

Out[8]:

| | month | day | campaign_number | user_engagement | banner | placement | displays | cost | clicks | revenue | post_click_conversi |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | April | 1 | camp 1 | High | 160 x 600 | abc | 4 | 0.0060 | 0 | 0.0000 | |
| **1** | April | 1 | camp 1 | High | 160 x 600 | def | 20170 | 26.7824 | 158 | 28.9717 | |
| **2** | April | 1 | camp 1 | High | 160 x 600 | ghi | 14701 | 27.6304 | 158 | 28.9771 | |
| **3** | April | 1 | camp 1 | High | 160 x 600 | mno | 171259 | 216.8750 | 1796 | 329.4518 | |
| **4** | April | 1 | camp 1 | Low | 160 x 600 | def | 552 | 0.0670 | 1 | 0.1834 | |
| **5** | April | 1 | camp 1 | Low | 160 x 600 | ghi | 16 | 0.0249 | 0 | 0.0000 | |

In [9]:
```
data.isna().any()
```

Out[9]:
```
month                    False
day                      False
campaign_number          False
user_engagement          False
banner                   False
placement                 True
displays                 False
cost                     False
clicks                   False
revenue                  False
post_click_conversions   False
post_click_sales_amount  False
dtype: bool
```

In [10]:
```
for col in data:
    if data[col].dtype == object :
        print(f'{col} : {data[col].unique()}')
```

```
month : ['April' 'May' 'June']
campaign_number : ['camp 1' 'camp 2' 'camp 3']
user_engagement : ['High' 'Low' 'Medium']
banner : ['160 x 600' '240 x 400' '300 x 250' '468 x 60' '580 x 400' '670 x 90'
 '728 x 90' '800 x 250']
placement : ['abc' 'def' 'ghi' 'mno' 'jkl' nan]
```

What is the overall trend in user engagement throughout the campaign period?

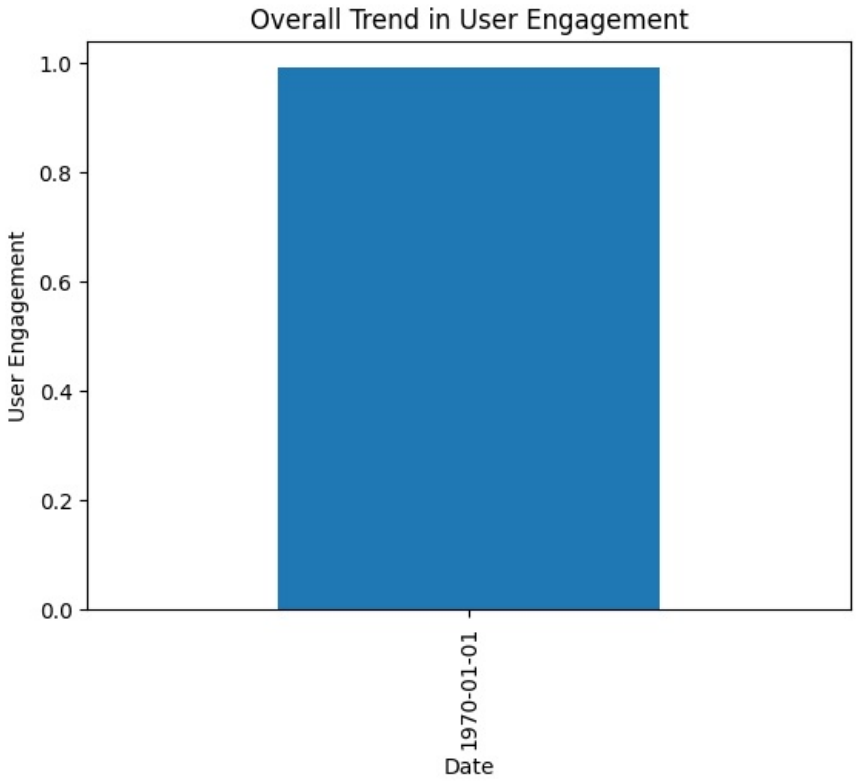
In [11]:
```
data.user_engagement = data.user_engagement.map({'High' :2 , 'Low': 0 , 'Medium':1})
```

In [12]:
```
data.user_engagement.value_counts()
```

Out[12]:
```
user_engagement
1    5489
0    5035
2    4884
Name: count, dtype: int64
```

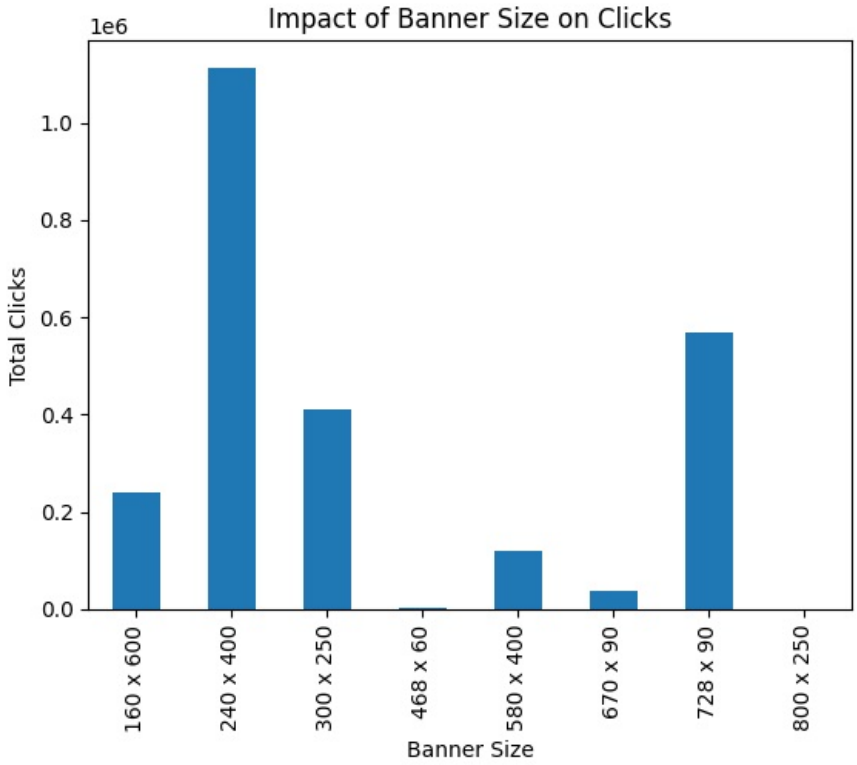
In [13]:
```
data['day'] = pd.to_datetime(data['day'])
engagement_trend = data.groupby(data['day'].dt.date)['user_engagement'].mean()

# Print the first few rows of the engagement trend data
print(engagement_trend.head())

# Plot the engagement trend
engagement_trend.plot(kind='bar')
plt.title('Overall Trend in User Engagement')
plt.xlabel('Date')
plt.ylabel('User Engagement')
plt.show()
```

```
day
1970-01-01    0.9902
Name: user_engagement, dtype: float64
```




How does the size of the ad (banner) impact the number of clicks generated?

In [14]:
```
banner_clicks = data.groupby('banner')['clicks'].sum()
banner_clicks.plot(kind='bar')
plt.title('Impact of Banner Size on Clicks')
plt.xlabel('Banner Size')
plt.ylabel('Total Clicks')
plt.show()
```




Which publisher spaces (placements) yielded the highest number of displays and clicks?

In [15]:
```
top_placements_displays = data.groupby('placement')['displays'].sum().nlargest(5)
top_placements_clicks = data.groupby('placement')['clicks'].sum().nlargest(5)
print('<---------------------------------------------------->')
print("Top Placements by Displays:\n", top_placements_displays)
print('<---------------------------------------------------->')
print("Top Placements by Clicks:\n", top_placements_clicks)
print('<---------------------------------------------------->')
```

```
<---------------------------------------------------->
Top Placements by Displays:
 placement
mno    143161775
ghi     59740415
def     28177492
jkl      7692732
abc       242142
Name: displays, dtype: int64
<---------------------------------------------------->
Top Placements by Clicks:
 placement
ghi    1247049
mno     993039
def     176097
jkl      75063
abc       1584
Name: clicks, dtype: int64
<---------------------------------------------------->
```

Is there a correlation between the cost of serving ads and the revenue generated from clicks?

In [16]:
```
cost_revenue_corr = data['cost'].corr(data['revenue'])
print('<---------------------------------------------------->')
print("Correlation between Cost and Revenue:", cost_revenue_corr)
print('<---------------------------------------------------->')
```

```
<---------------------------------------------------->
Correlation between Cost and Revenue: 0.7605199343382272
<---------------------------------------------------->
```

What is the average revenue generated per click for Company X during the campaign period?

In [17]:
```
avg_revenue_per_click = data['revenue'].sum() / data['clicks'].sum()
print('<---------------------------------------------------->')
print("Average Revenue per Click:", avg_revenue_per_click)
print('<---------------------------------------------------->')
```

```
<---------------------------------------------------->
Average Revenue per Click: 0.1108233559193802
<---------------------------------------------------->
```

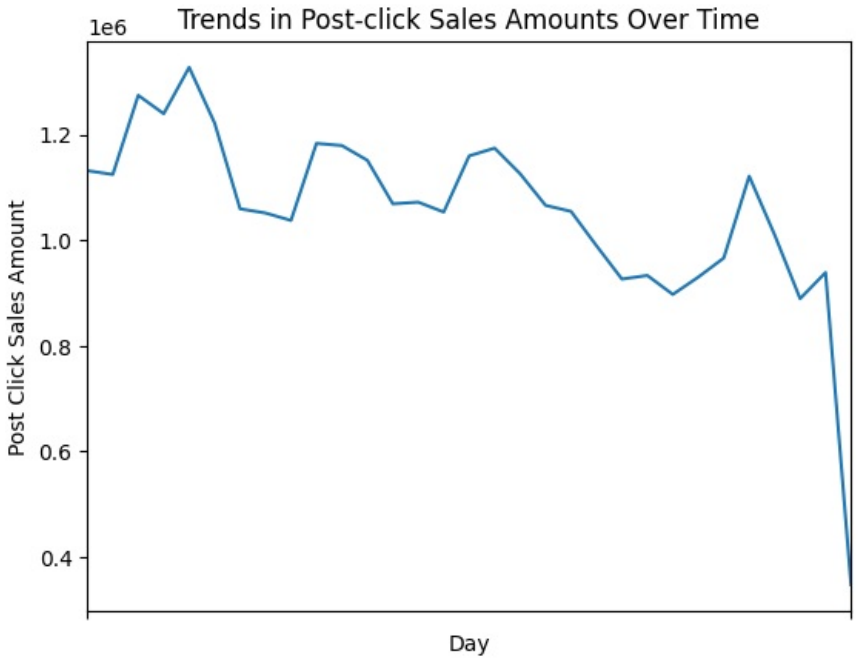
Which campaigns had the highest post-click conversion rates?

In [18]:
```
post_click_conversion_rates = (data.groupby('campaign_number')['post_click_conversions'].sum() /
                               data.groupby('campaign_number')['clicks'].sum())
highest_conversion_campaigns = post_click_conversion_rates.nlargest(5)
print('<-------------------------------------------------------->')
print("Campaigns with Highest Post-click Conversion Rates:\n", highest_conversion_campaigns)
print('<-------------------------------------------------------->')
```

```
<-------------------------------------------------------->
Campaigns with Highest Post-click Conversion Rates:
 campaign_number
camp 1    0.449272
camp 3    0.024271
camp 2    0.015624
dtype: float64
<-------------------------------------------------------->
```

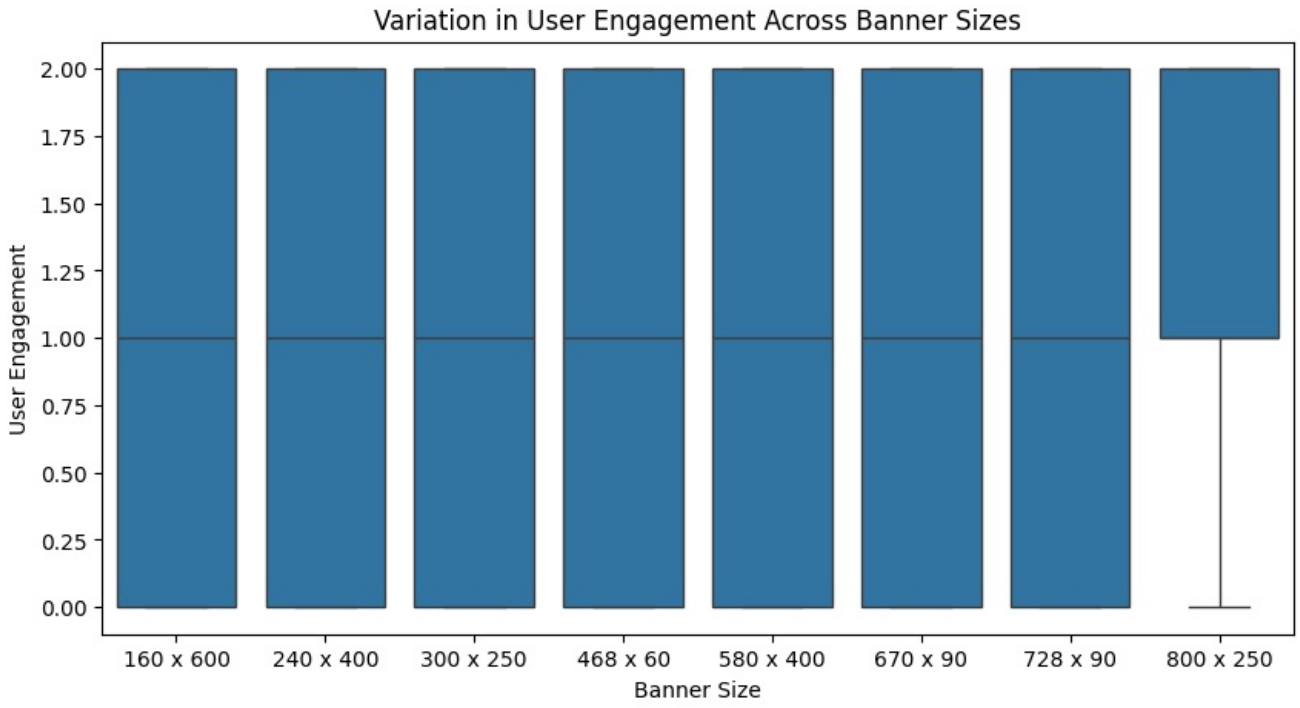
Are there any specific trends or patterns in post-click sales amounts over time?

In [19]:
```
sales_trend = data.groupby('day')['post_click_sales_amount'].sum()
sales_trend.plot(kind='line')
plt.title('Trends in Post-click Sales Amounts Over Time')
plt.xlabel('Day')
plt.ylabel('Post Click Sales Amount')
plt.show()
```




How does the level of user engagement vary across different banner sizes?

In [20]:
```
plt.figure(figsize = (10,5))
sns.boxplot(x='banner', y='user_engagement', data=data)
plt.title('Variation in User Engagement Across Banner Sizes')
plt.xlabel('Banner Size')
plt.ylabel('User Engagement')
plt.show()
```




Which placement types result in the highest post-click conversion rates?

In [21]:
```
placement_conversion_rates = (data.groupby('placement')['post_click_conversions'].sum() /
                              data.groupby('placement')['clicks'].sum())
highest_conversion_placements = placement_conversion_rates.nlargest(5)
print('<-------------------------------------------------------->')
print("Placement Types with Highest Post-click Conversion Rates:\n", highest_conversion_placements)
print('<-------------------------------------------------------->')
```

```
<-------------------------------------------------------->
Placement Types with Highest Post-click Conversion Rates:
 placement
abc    0.520202
jkl    0.277807
ghi    0.270288
mno    0.265015
def    0.169543
dtype: float64
<-------------------------------------------------------->
```

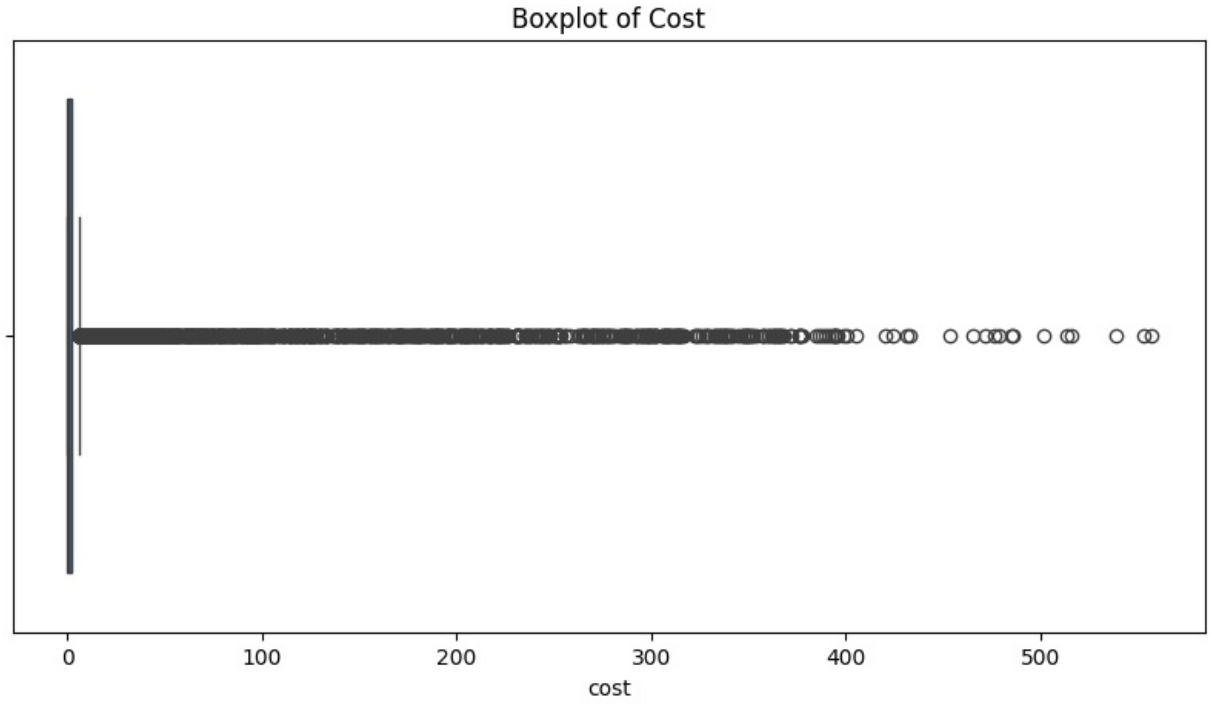
Is there a correlation between user engagement levels and the revenue generated?

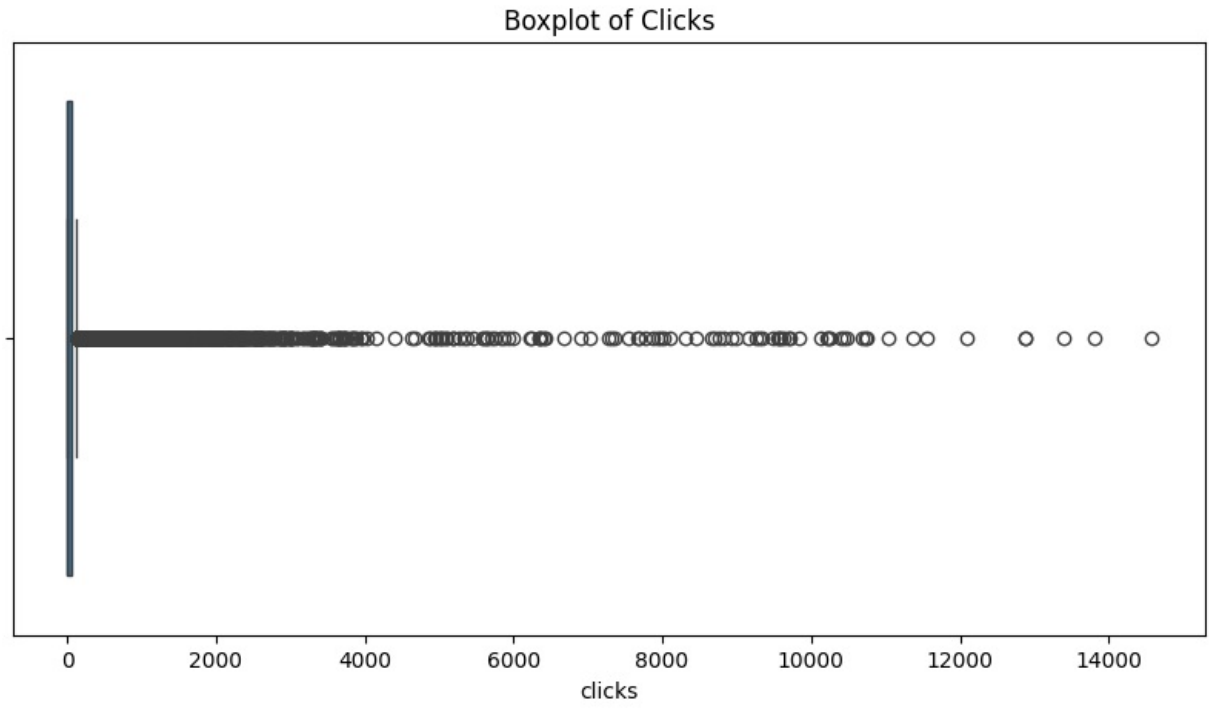
In [22]:
```
engagement_revenue_corr = data['user_engagement'].corr(data['revenue'])
print('<-------------------------------------------------------->')
print("Correlation between User Engagement and Revenue:", engagement_revenue_corr)
print('<-------------------------------------------------------->')
```

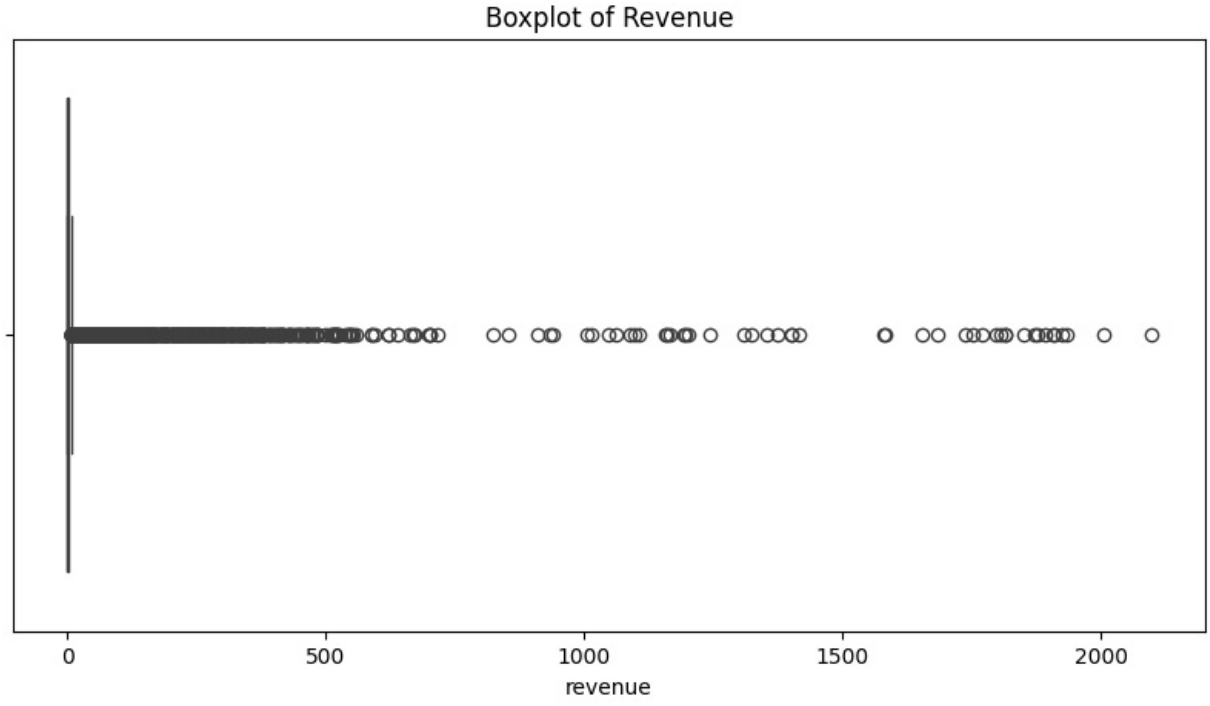
```
<-------------------------------------------------------->
Correlation between User Engagement and Revenue: 0.1753892426950315
<-------------------------------------------------------->
```

Are there any outliers in terms of cost, clicks, or revenue that warrant further investigation?

In [23]:
```
plt.figure(figsize = (10,5))
sns.boxplot(x='cost', data=data)
plt.title('Boxplot of Cost')
plt.show()

plt.figure(figsize = (10,5))
sns.boxplot(x='clicks', data=data)
plt.title('Boxplot of Clicks')
plt.show()

plt.figure(figsize = (10,5))
sns.boxplot(x='revenue', data=data)
plt.title('Boxplot of Revenue')
plt.show()
```


Boxplot of Cost
0
100
200
300
400
500
cost


Boxplot of Clicks
0
2000
4000
6000
8000
10000
12000
14000
clicks


Boxplot of Revenue
0
500
1000
1500
2000
revenue

How does the effectiveness of campaigns vary based on the size of the ad and placement type?

In [24]:
```
campaign_effectiveness = data.groupby(['campaign_number', 'banner', 'placement']).agg({
    'post_click_conversions': 'sum',
    'clicks': 'sum',
    'cost': 'sum',
    'revenue': 'sum'
})


campaign_effectiveness['ROI'] = (campaign_effectiveness['revenue'] - campaign_effectiveness['cost']) / campaign_
campaign_effectiveness['Conversion Rate'] = campaign_effectiveness['post_click_conversions'] / campaign_effecti

campaign_effectiveness.head(n=6)
```

Out[24]:

| | | | post_click_conversions | clicks | cost | revenue | ROI | Conversion Rate |
|---|---|---|---|---|---|---|---|---|
| **campaign_number** | **banner** | **placement** | | | | | | |
| **camp 1** | **160 x 600** | **abc** | 0 | 0 | 1.3917 | 0.0000 | -1.000000 | NaN |
| | | **def** | 2386 | 8612 | 1212.8717 | 1434.2667 | 0.182538 | 0.277055 |
| | | **ghi** | 3930 | 7742 | 1614.9998 | 1291.9142 | -0.200053 | 0.507621 |
| | | **jkl** | 0 | 0 | 0.0015 | 0.0000 | -1.000000 | NaN |
| | | **mno** | 40690 | 89886 | 12238.4652 | 15079.9859 | 0.232179 | 0.452685 |
| | **240 x 400** | **def** | 5001 | 15502 | 2371.1678 | 2626.5298 | 0.107695 | 0.322604 |

Are there any specific campaigns or banner sizes that consistently outperform others in terms of ROI?

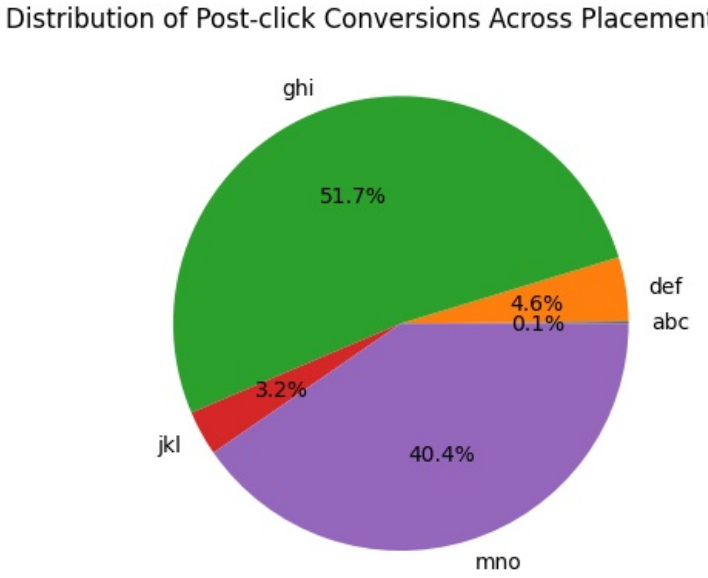
In [25]:
```
campaign_effectiveness.head(n=7)
```

Out[25]:

| | | | post_click_conversions | clicks | cost | revenue | ROI | Conversion Rate |
|---|---|---|---|---|---|---|---|---|
| **campaign_number** | **banner** | **placement** | | | | | | |
| **camp 1** | **160 x 600** | **abc** | 0 | 0 | 1.3917 | 0.0000 | -1.000000 | NaN |
| | | **def** | 2386 | 8612 | 1212.8717 | 1434.2667 | 0.182538 | 0.277055 |
| | | **ghi** | 3930 | 7742 | 1614.9998 | 1291.9142 | -0.200053 | 0.507621 |
| | | **jkl** | 0 | 0 | 0.0015 | 0.0000 | -1.000000 | NaN |
| | | **mno** | 40690 | 89886 | 12238.4652 | 15079.9859 | 0.232179 | 0.452685 |
| | **240 x 400** | **def** | 5001 | 15502 | 2371.1678 | 2626.5298 | 0.107695 | 0.322604 |
| | | **ghi** | 204508 | 557480 | 24884.0791 | 91362.4930 | 2.671524 | 0.366844 |

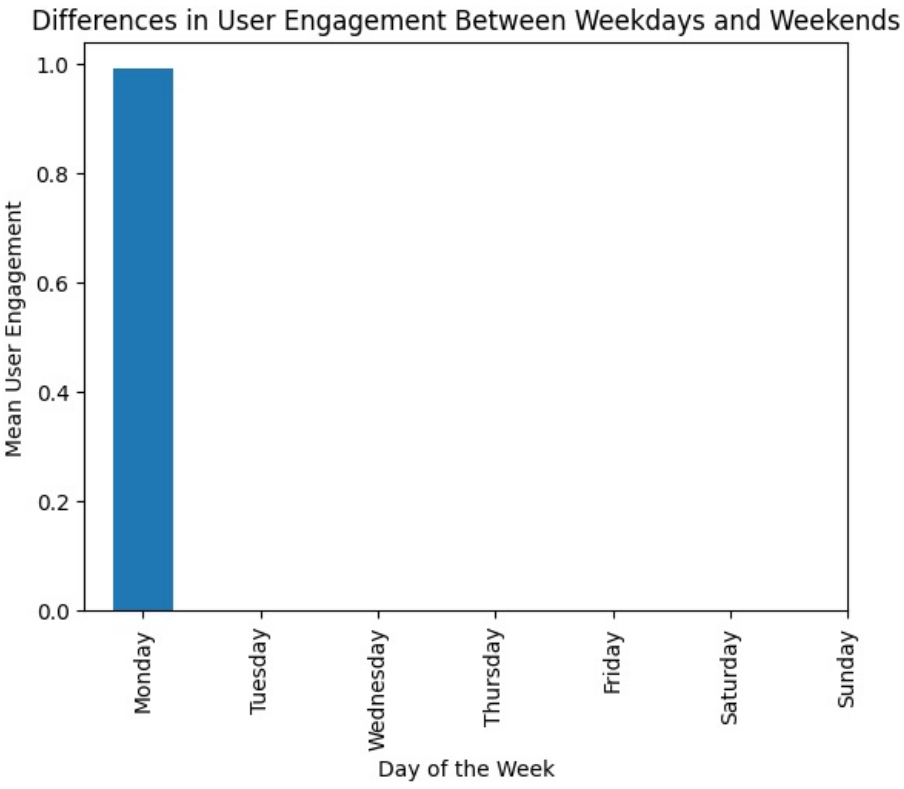
What is the distribution of post-click conversions across different placement types?

In [26]:
```
placement_conversion_distribution = data.groupby('placement')['post_click_conversions'].sum()
placement_conversion_distribution.plot(kind='pie', autopct='%1.1f%%')
plt.title('Distribution of Post-click Conversions Across Placement Types')
plt.ylabel('')
plt.show()
```



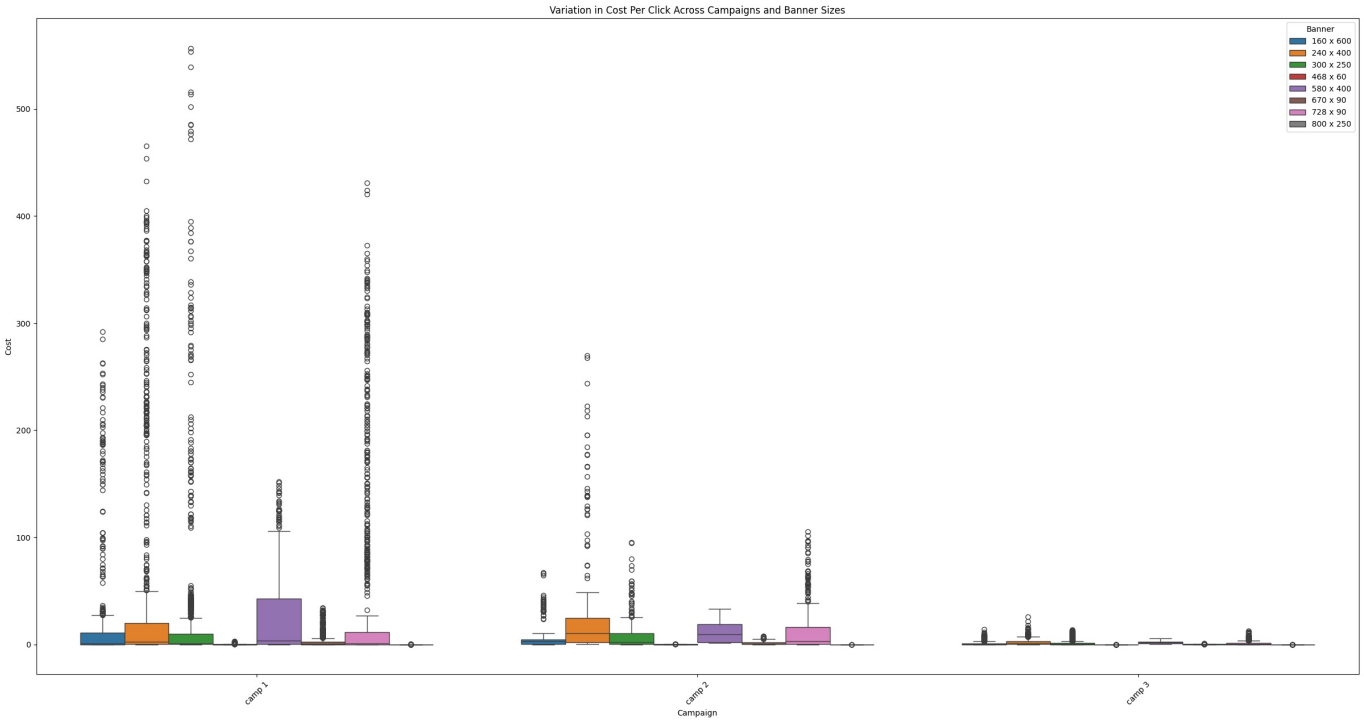

Are there any noticeable differences in user engagement levels between weekdays and weekends?

In [27]:
```
data.reset_index(drop=True, inplace=True)
data['day_of_week'] = data['day'].dt.dayofweek
weekday_engagement = data.groupby('day_of_week')['user_engagement'].mean()
weekday_engagement.plot(kind='bar')
plt.title('Differences in User Engagement Between Weekdays and Weekends')
plt.xlabel('Day of the Week')
plt.ylabel('Mean User Engagement')
plt.xticks(range(7), ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
plt.show()
```




How does the cost per click (CPC) vary across different campaigns and banner sizes?

In [28]:
```
plt.figure(figsize = (30,15))
sns.boxplot(x='campaign_number', y='cost', hue='banner', data=data)
plt.title('Variation in Cost Per Click Across Campaigns and Banner Sizes')
plt.xlabel('Campaign')
plt.ylabel('Cost')
plt.xticks(rotation=45)
plt.legend(title='Banner')
plt.show()
```




Are there any campaigns or placements that are particularly cost-effective in terms of generating post-click conversions?

In [29]:
```
cost_effectiveness = data.groupby(['campaign_number', 'placement']).agg({
    'post_click_conversions': 'sum',
    'cost': 'sum'
})
cost_effectiveness['cost_per_conversion'] = cost_effectiveness['cost'] / cost_effectiveness['post_click_conversi
cost_effectiveness_sorted = cost_effectiveness.sort_values(by='cost_per_conversion')
top_cost_effective = cost_effectiveness_sorted.head(5)
print('<------------------------------------------>')
print("Top 5 Cost-Effective Campaigns or Placements:")
print('<------------------------------------------>')
top_cost_effective
```

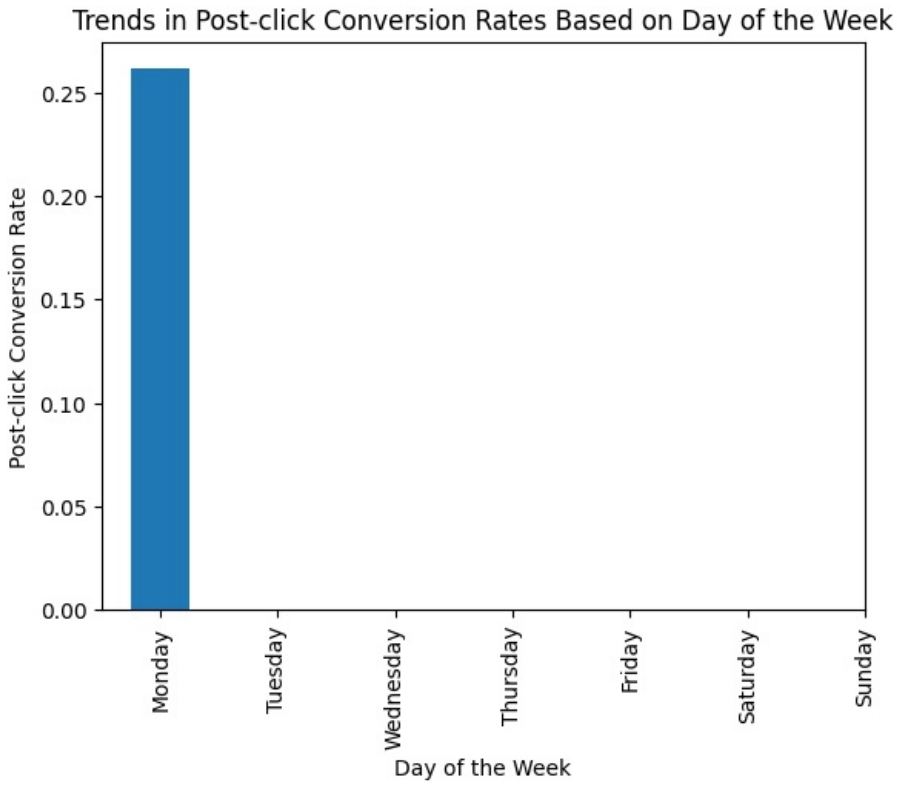
```
<------------------------------------------>
Top 5 Cost-Effective Campaigns or Placements:
<------------------------------------------>
```

Out[29]:

| campaign_number | placement | post_click_conversions | cost | cost_per_conversion |
|---|---|---|---|---|
| camp 1 | abc | 808 | 98.3361 | 0.121703 |
| | jkl | 20109 | 2746.7036 | 0.136591 |
| | ghi | 329024 | 52153.6716 | 0.158510 |
| | mno | 254778 | 83968.8304 | 0.329576 |
| | def | 28364 | 11719.4056 | 0.413179 |

Can we identify any trends or patterns in post-click conversion rates based on the day of the week?

In [30]:
```
day_conversion_rates = (data.groupby('day_of_week')['post_click_conversions'].sum() /
                        data.groupby('day_of_week')['clicks'].sum())
day_conversion_rates.plot(kind='bar')
plt.title('Trends in Post-click Conversion Rates Based on Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Post-click Conversion Rate')
plt.xticks(range(7), ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
plt.show()
```




How does the effectiveness of campaigns vary between new users and returning users in terms of post-click conversions?

In [31]:
```
campaign_effectiveness = data.groupby(['campaign_number', data.index]).agg({
    'post_click_conversions': 'sum',
    'clicks': 'sum'
})

campaign_effectiveness['conversion_rate'] = campaign_effectiveness['post_click_conversions'] / campaign_effecti
campaign_effectiveness.reset_index(inplace=True)
```

In [32]:
```
plt.figure(figsize=(10, 6))
mean_conversion_rate = campaign_effectiveness.groupby(['campaign_number']).agg({'conversion_rate': 'mean'})

mean_conversion_rate.plot(kind='bar', color='skyblue')
plt.title('Mean Conversion Rate by Campaign')
plt.xlabel('Campaign')
plt.ylabel('Mean Conversion Rate')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```