```
Customer Lifetime Value (CLV) analysis is a strategic approach that estimates the total revenue a business can expect from a customer throughout their entire relationship. By understanding CLV, companies can assess customer profitability, inform marketing investments, and enhance retention strategies. This analysis involves calculating both historical and predictive CLV, enabling businesses to tailor their efforts towards high-value customers and optimize resource allocation for sustained growth
```

```
Dataset Description:
The dataset provided comprises customer interaction data from various channels, detailing the cost incurred, conversion rate achieved, and revenue generated for each customer.Below are the features:

customer_id: A unique identifier for each customer.

channel: The marketing channel through which the customer was acquired,
    such as 'referral', 'paid advertising', 'email marketing', or 'social media'.

cost: The expenditure associated with acquiring the customer through the specified channel.

conversion_rate: The percentage of interactions that resulted in a desired outcome, such as a purchase or sign-up.

revenue: The income generated from the customer, presumably as a result of the conversion
```

In [2]:

```python
#Importing necessary libraries:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [42]:

```python
#Data path
data = pd.read_csv(r"C:\Users\Hp\Downloads\customer_acquisition_data.c
print(data.head())
```

```
   customer_id           channel       cost  conversion_rate  revenue
0            1          referral   8.320327         0.123145     4199
1            2  paid advertising  30.450327         0.016341     3410
2            3   email marketing   5.246263         0.043822     3164
3            4      social media   9.546326         0.167592     1520
4            5          referral   8.320327         0.123145     2419
```

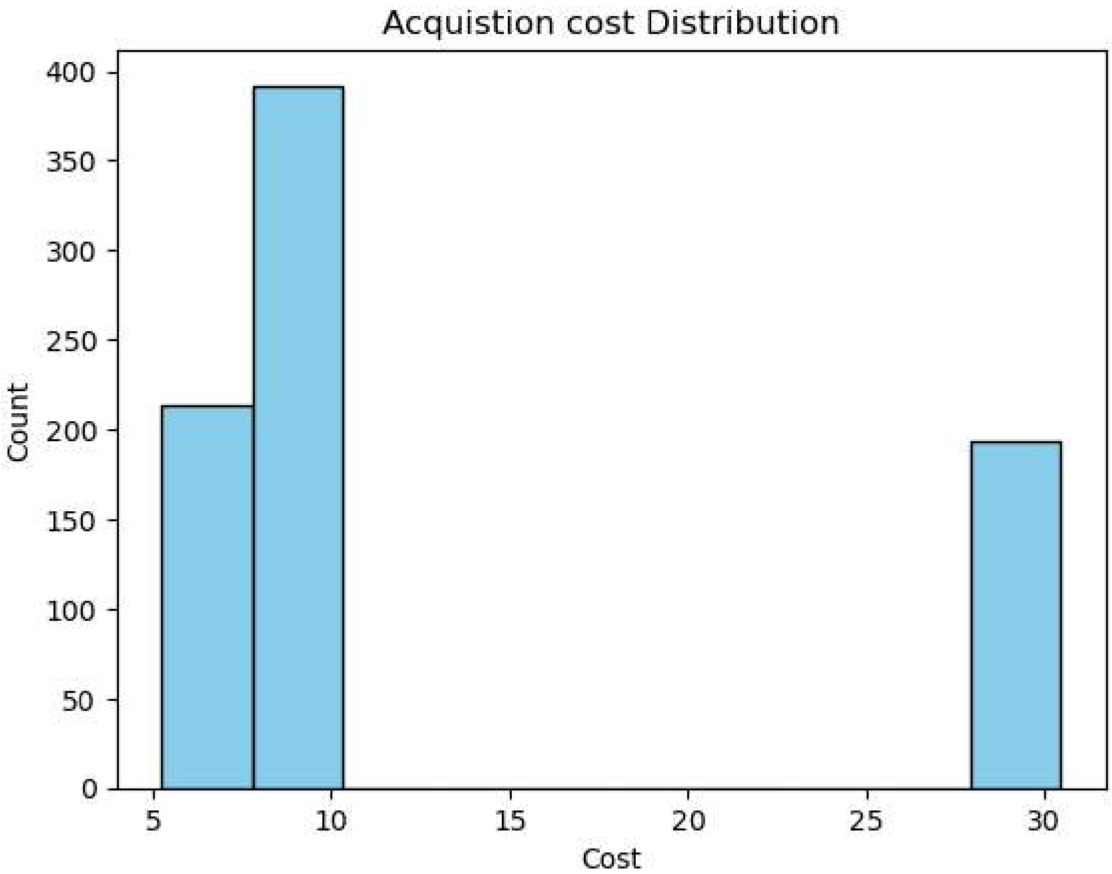
In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   customer_id      800 non-null    int64  
 1   channel          800 non-null    object 
 2   cost             800 non-null    float64
 3   conversion_rate  800 non-null    float64
 4   revenue          800 non-null    int64  
dtypes: float64(2), int64(2), object(1)
memory usage: 31.4+ KB
```

In [8]:
```
#Checking if there is null values:
data.isnull().sum()
```

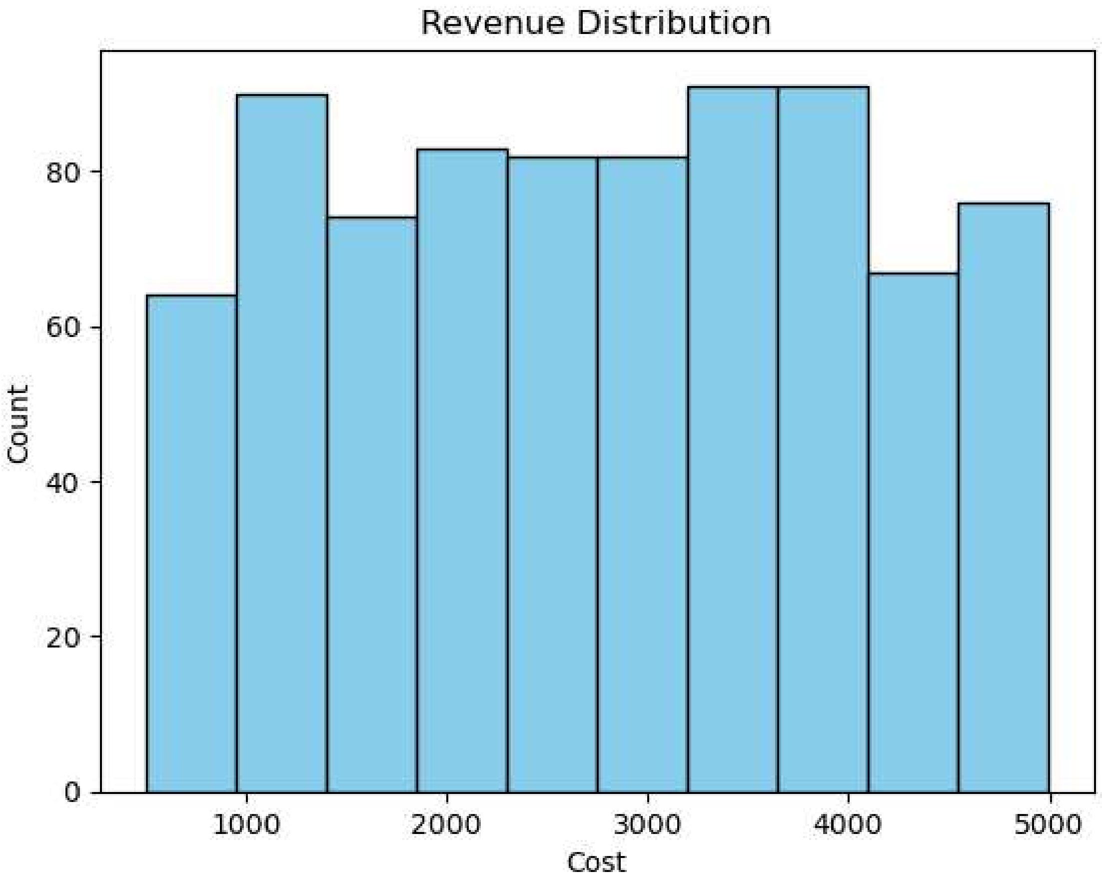
Out[8]:
```
customer_id        0
channel            0
cost               0
conversion_rate    0
revenue            0
dtype: int64
```

In [6]:
```
#ACQUISTION COST DISTRIBUTION

plt.hist(data['cost'], bins=10, color='skyblue', edgecolor='black')

# Add title and labels
plt.title('Acquistion cost Distribution')
plt.xlabel('Cost')
plt.ylabel('Count')

# Show the plot
plt.show()
```

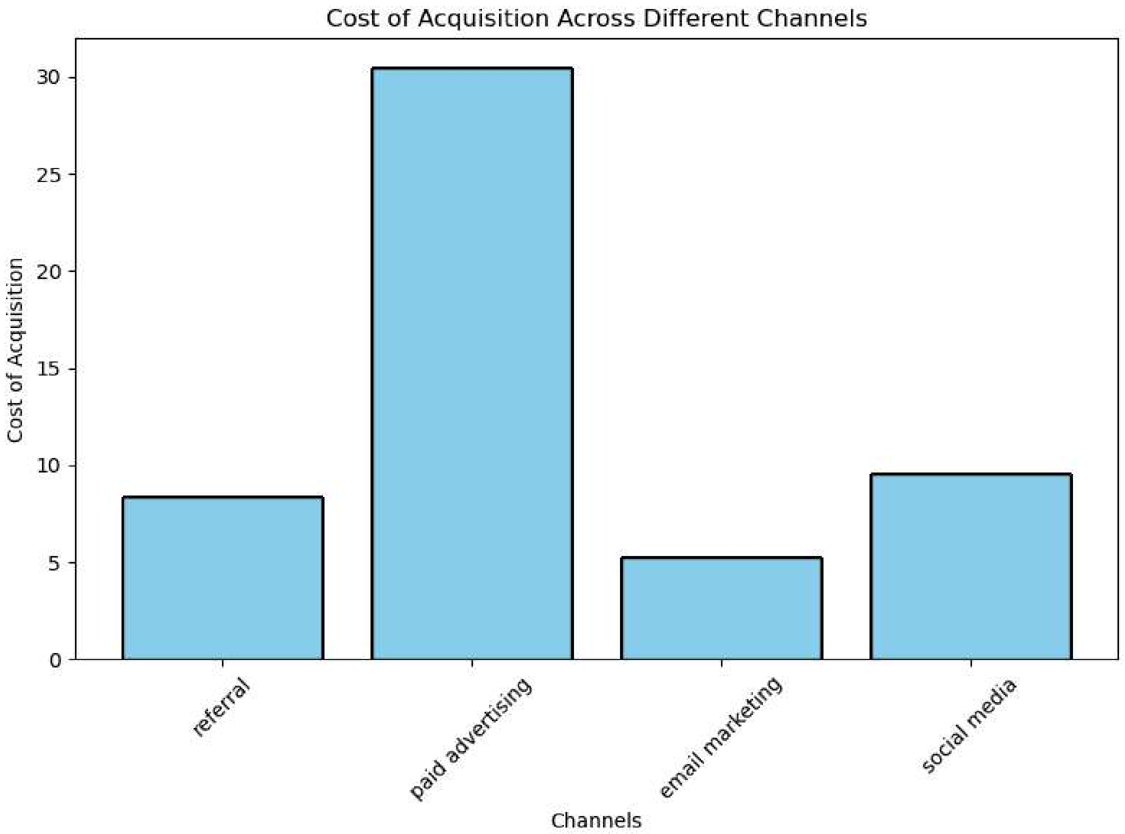


In [11]:

```
#REVENUE DISTRIBUTION

plt.hist(data['revenue'], bins=10, color='skyblue', edgecolor='black')

# Add title and labels
plt.title('Revenue Distribution')
plt.xlabel('Cost')
plt.ylabel('Count')

# Show the plot
plt.show()
```

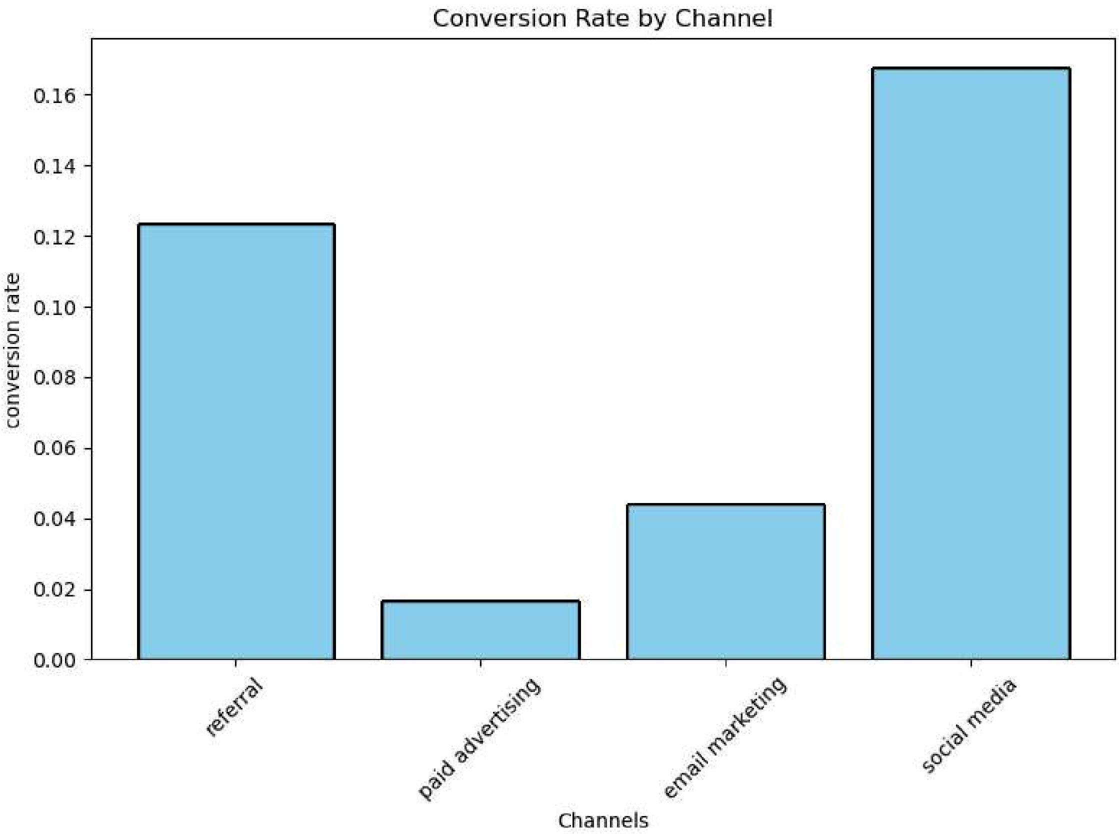


In [12]:
```
# Plotting the Cost of Acquisition across different channels
plt.figure(figsize=(8, 6))
plt.bar(data['channel'], data['cost'], color='skyblue', edgecolor='bla

# Adding titles and labels
plt.title('Cost of Acquisition Across Different Channels')
plt.xlabel('Channels')
plt.ylabel('Cost of Acquisition')
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



So paid advertisement is the most expensive channel, and email marketing is the least expensive channel.

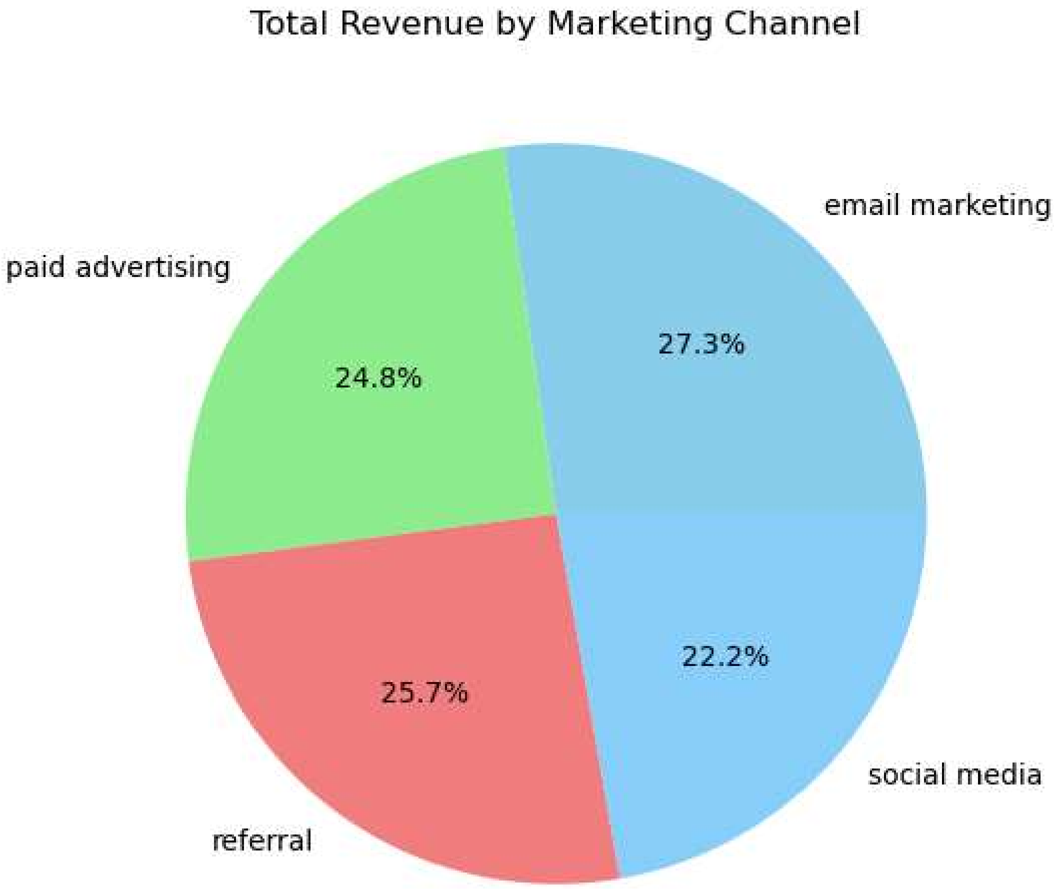
In [14]:
```python
# Plotting the Conversion Rate by Channel
plt.figure(figsize=(8, 6))
plt.bar(data['channel'], data['conversion_rate'], color='skyblue', edg

# Adding titles and labels
plt.title('Conversion Rate by Channel')
plt.xlabel('Channels')
plt.ylabel('conversion rate')
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```




```
Social media is the most effective channel for converting customers,
while paid advertising is the least effective.
```

In [23]:

```
#Total Revenue by Channel
# Group by Channel and sum the Revenue for each channel
channel_revenue = data.groupby('channel')['revenue'].sum()

# Plotting the pie chart for Total Revenue by Channel
plt.figure(figsize=(8, 6))
plt.pie(channel_revenue, labels=channel_revenue.index, autopct='%1.1f%

# Adding a title
plt.title('Total Revenue by Marketing Channel')

# Show the plot
plt.show()
```



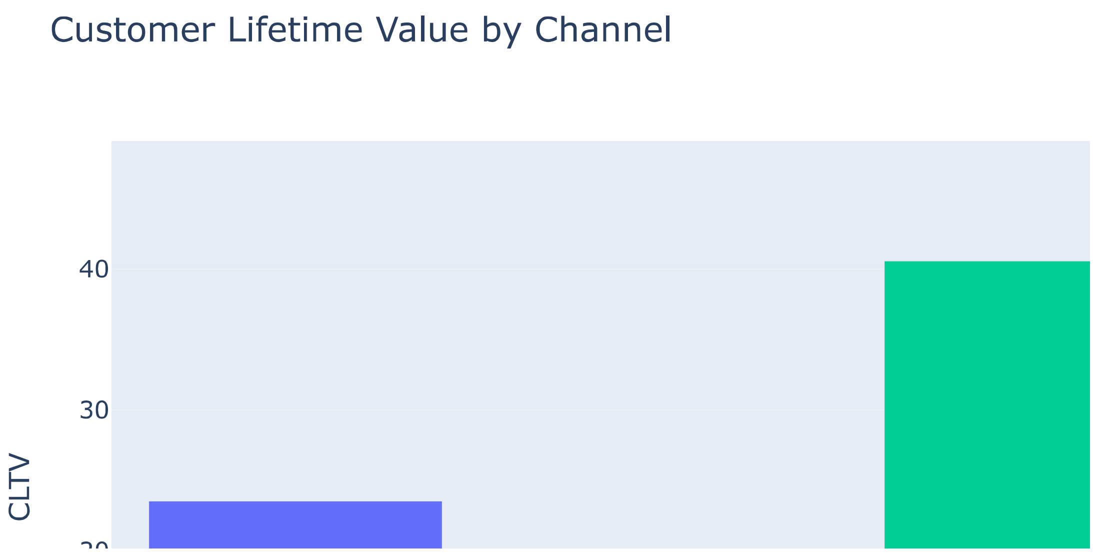
So email marketing is the most profitable channel in terms of generating revenue.But there’s not a huge difference between the percentages of revenue generation from all the channels to call any channel the least profitable channel.

In [35]:

```
import plotly.express as px
data['roi'] = data['revenue'] / data['cost']
roi_by_channel = data.groupby('channel')['roi'].mean().reset_index()

fig = px.bar(roi_by_channel,
             x='channel',
             y='roi', title='Return on Investment (ROI) by Channel')
fig.show()
```


```
The ROI from email marketing is way higher than all other channels,
while the ROI from paid advertising is the lowest.
```

In [37]:
```
#calculate the customer lifetime value from each channel
data['cltv'] = (data['revenue'] - data['cost']) * data['conversion_rat
 
channel_cltv = data.groupby('channel')['cltv'].mean().reset_index()
 
fig = px.bar(channel_cltv, x='channel', y='cltv', color='channel',
             title='Customer Lifetime Value by Channel')
 
fig.update_xaxes(title='Channel')
fig.update_yaxes(title='CLTV')
 
fig.show()
```




```
So the customer lifetime value from Social Media and the referral
channels is the highest.
```

In [43]:

```
import pandas as pd
from sklearn.linear_model import LinearRegression


# Check if any NaN values are present in the features or target variab
print("Missing values in features:")
print(data[['conversion_rate', 'cost', 'revenue']].isna().sum())

# Features (X) and target (y)
X = data[['conversion_rate', 'cost']]  # Features for the model
y = data['revenue']  # Target variable (what we want to predict)

# Initialize the linear regression model
model = LinearRegression()

# Fit the model with the data
model.fit(X, y)

# Make predictions
data['predicted_revenue'] = model.predict(X)

# Calculate predicted CLV (predicted revenue - cost)
data['predicted_clv'] = data['predicted_revenue'] - data['cost']

# Display the updated data with predictions
print("\nUpdated Data with Predicted Revenue and Predicted CLV:")
print(data[['customer_id', 'predicted_revenue', 'predicted_clv']])

```

```
Missing values in features:
conversion_rate    0
cost               0
revenue            0
dtype: int64

Updated Data with Predicted Revenue and Predicted CLV:
     customer_id  predicted_revenue  predicted_clv
0              1        2732.193320    2723.872993
1              2        2827.033562    2796.583235
2              3        2832.144156    2826.897893
3              4        2676.938499    2667.392173
4              5        2732.193320    2723.872993
..           ...                ...            ...
795          796        2676.938499    2667.392173
796          797        2832.144156    2826.897893
797          798        2676.938499    2667.392173
798          799        2827.033562    2796.583235
799          800        2832.144156    2826.897893

[800 rows x 3 columns]
```

```
Interpretation:
For example, for customer_id 1, the predicted revenue is 2732.19.
This is the amount of money the model expects this customer to generate based on the given features such as conversion_rate and cost.
```

In [ ]:

```
1
```