Task 4 : - Analyze and visualize sentiment patterns in social media data to understand public opinion and attitudes towards specific topics or brands

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

In [16]:
```
# Load the Dataset
data = pd.read_csv(r"C:\Users\kunal\Documents\PRODIGY Intership Material\PRODIGY_DS
```

In [17]:
```
df.shape
```

Out[17]:
```
(732, 15)
```

In [36]:
```
# Display descriptive statistics with orange gradient background
df.describe().loc[['min', '50%', 'mean', 'max', 'std']].T.style.background_gradient
```

Out[36]:

	min	50%	mean	max	std
Unnamed: 0.1	0.000000	366.500000	366.464481	732.000000	211.513936
Unnamed: 0	0.000000	370.500000	369.740437	736.000000	212.428936
Retweets	5.000000	22.000000	21.508197	40.000000	7.061286
Likes	10.000000	43.000000	42.901639	80.000000	14.089848
Year	2010.000000	2021.000000	2020.471311	2023.000000	2.802285
Month	1.000000	6.000000	6.122951	12.000000	3.411763
Day	1.000000	15.000000	15.497268	31.000000	8.474553
Hour	0.000000	16.000000	15.521858	23.000000	4.113414

In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 732 entries, 0 to 731
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0.1  732 non-null    int64  
 1   Unnamed: 0    732 non-null    int64  
 2   Text          732 non-null    object 
 3   Sentiment     732 non-null    object 
 4   Timestamp     732 non-null    object 
 5   User          732 non-null    object 
 6   Platform      732 non-null    object 
 7   Hashtags      732 non-null    object 
 8   Retweets      732 non-null    float64
 9   Likes         732 non-null    float64
 10  Country       732 non-null    object 
 11  Year          732 non-null    int64  
 12  Month         732 non-null    int64  
 13  Day           732 non-null    int64  
 14  Hour          732 non-null    int64  
dtypes: float64(2), int64(6), object(7)
memory usage: 85.9+ KB
```

In [20]:
```
df.isna().sum()
```

Out[20]:
```
Unnamed: 0.1    0
Unnamed: 0      0
Text            0
Sentiment       0
Timestamp       0
User            0
Platform        0
Hashtags        0
Retweets        0
Likes           0
Country         0
Year            0
Month           0
Day             0
Hour            0
dtype: int64
```

In [21]:
```
df_columns=df.columns
for col in df.columns:
    print(col)
```

```
Unnamed: 0.1
Unnamed: 0
Text
Sentiment
Timestamp
User
Platform
Hashtags
Retweets
Likes
Country
Year
Month
Day
Hour
```

In [22]:
```
df.head()
```

Out[22]:

	Unnamed: 0.1	Unnamed: 0	Text	Sentiment	Timestamp	User	Platform	Hashtags	R
0	0	0	Enjoying a beautiful day at the park! ...	Positive	2023-01-15 12:30:00	User123	Twitter	#Nature #Park	
1	1	1	Traffic was terrible this morning. ...	Negative	2023-01-15 08:45:00	CommuterX	Twitter	#Traffic #Morning	
2	2	2	Just finished an amazing workout! 💪 ...	Positive	2023-01-15 15:45:00	FitnessFan	Instagram	#Fitness #Workout	
3	3	3	Excited about the upcoming weekend getaway! ...	Positive	2023-01-15 18:20:00	AdventureX	Facebook	#Travel #Adventure	
4	4	4	Trying out a new recipe for dinner tonight. ...	Neutral	2023-01-15 19:55:00	ChefCook	Instagram	#Cooking #Food	

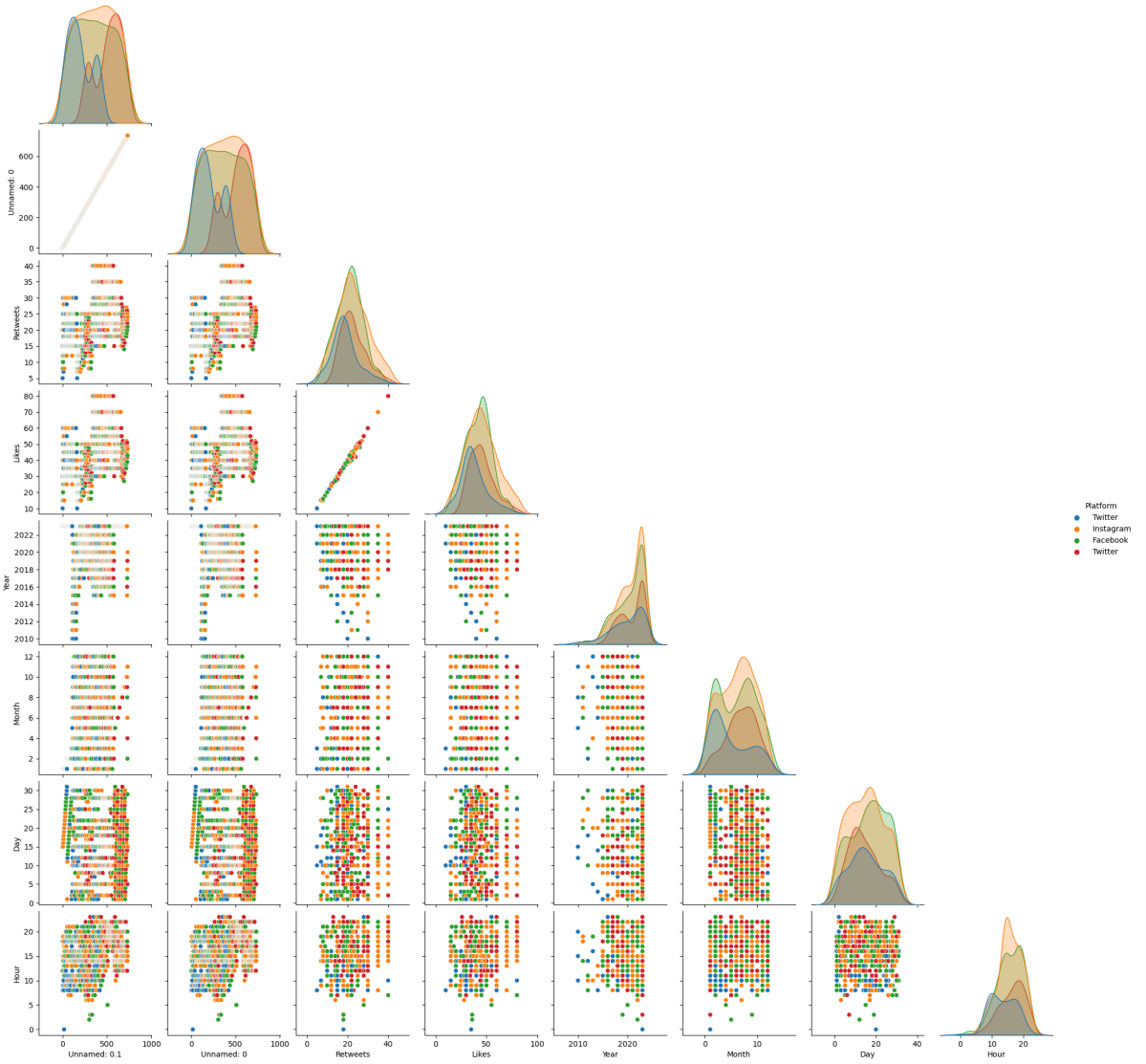
In [23]:
```
df.duplicated().any()
```

Out[23]:
```
False
```

VISUALIZATION ON DATA 📊

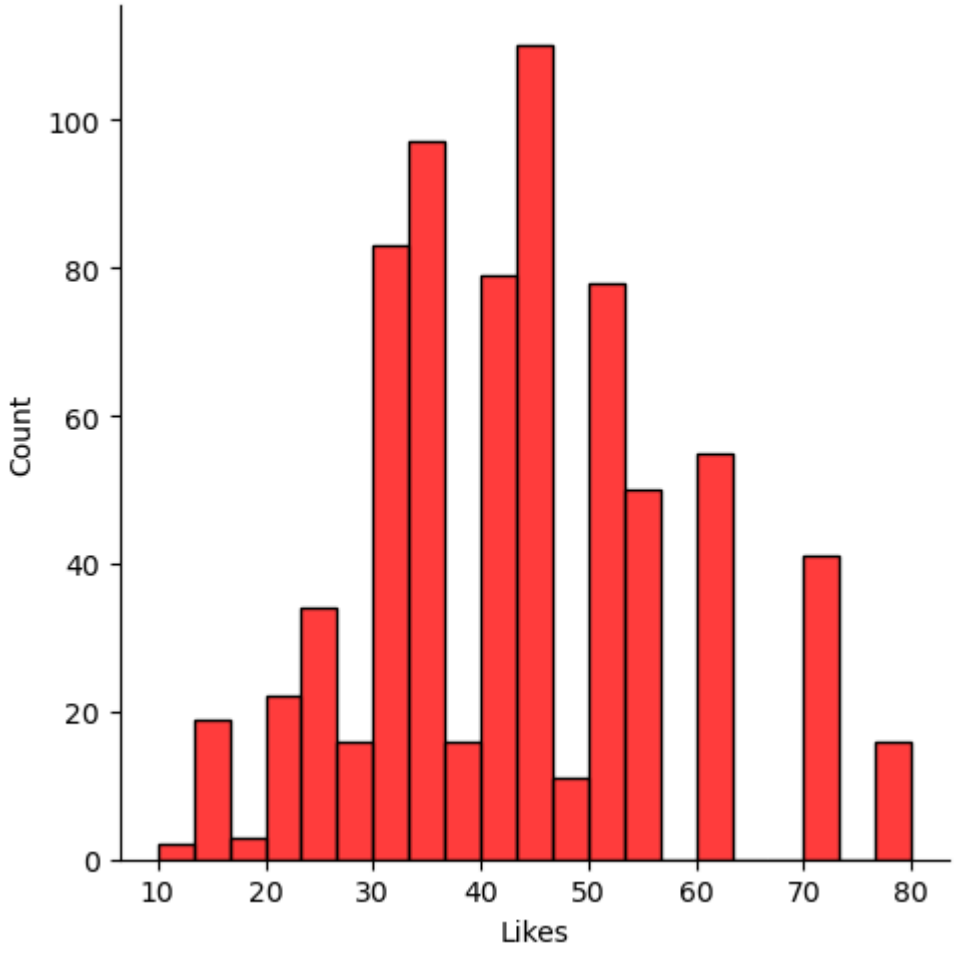
In [24]:
```
sns.pairplot(df, hue = 'Platform',corner=True )
```

Out[24]:
```
<seaborn.axisgrid.PairGrid at 0x14bd5942a90>
```

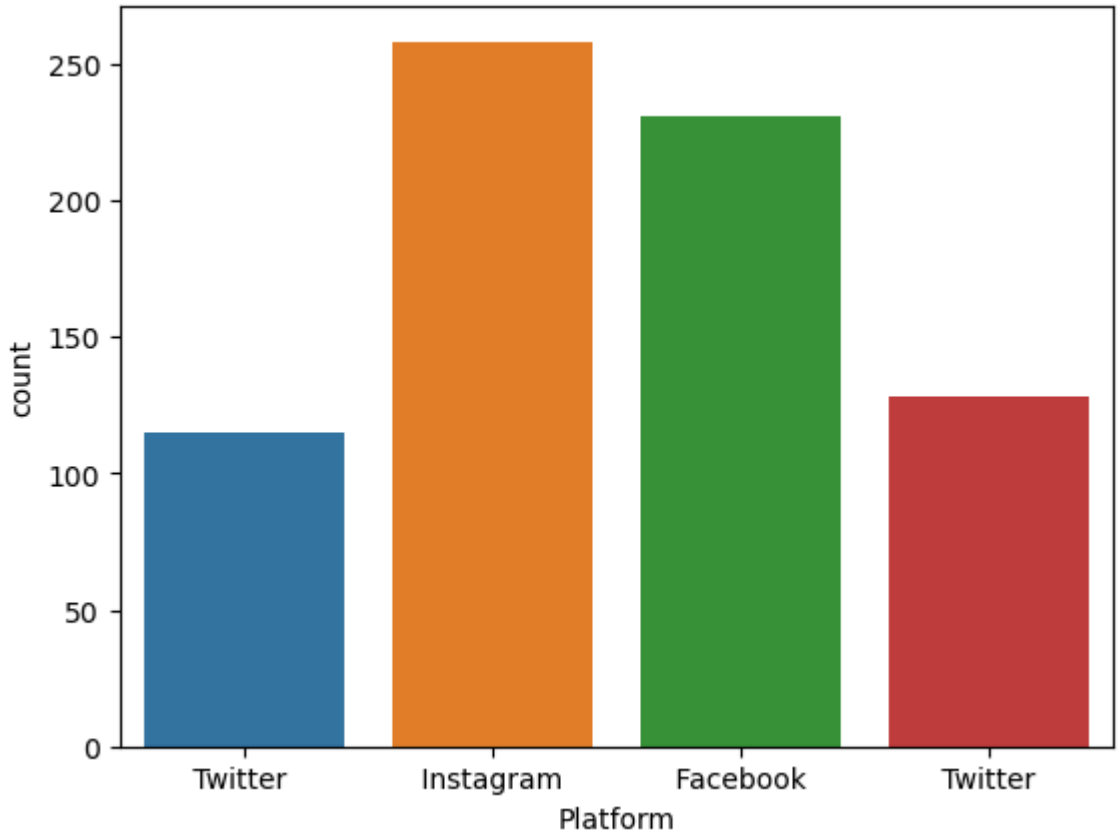


In [45]:
```
sns.displot(data=df, x='Likes', color='Red', fill=True)
```

Out[45]:
```
<seaborn.axisgrid.FacetGrid at 0x14be0f4c650>
```



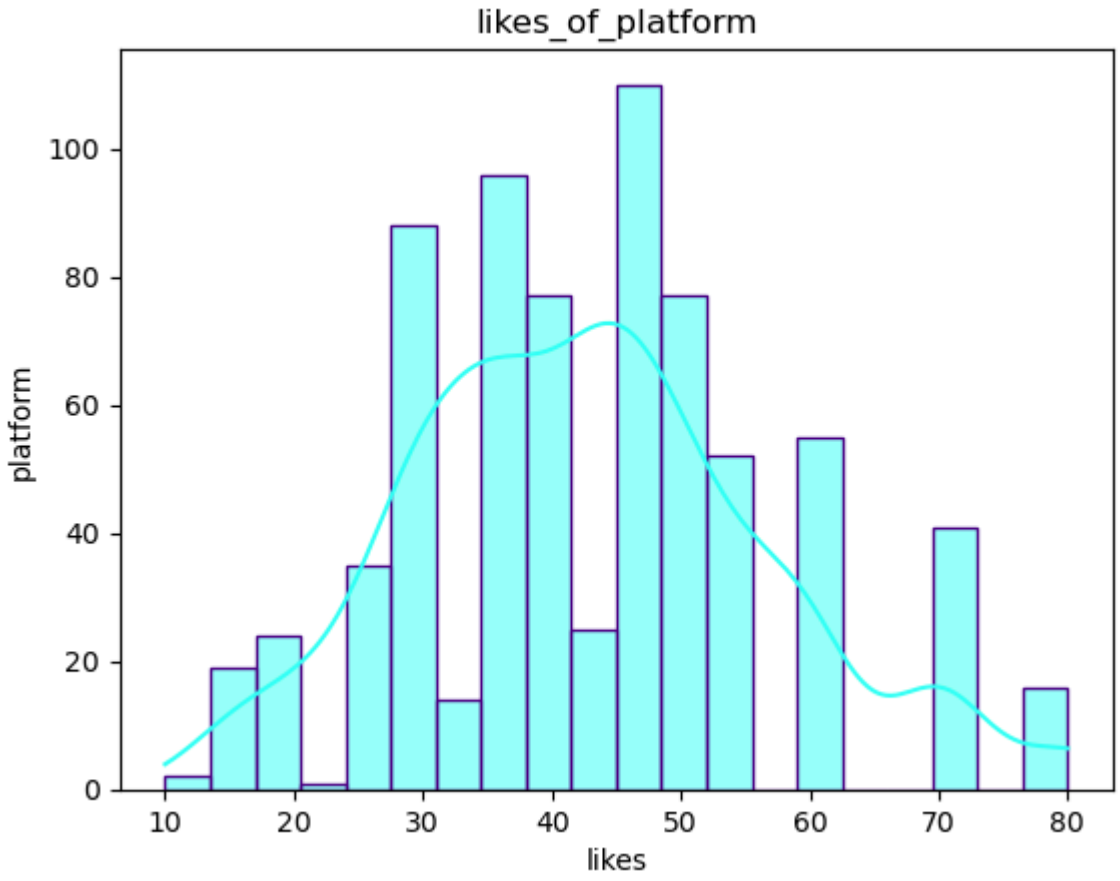
In [27]:
```
sns.countplot(x = 'Platform' , data = df)
plt.show()
```



In [71]:
```
sns.histplot(df['Likes'],bins=20,color='#33FFF6',edgecolor='INDIGO',kde=True)
plt.xlabel("likes")
```

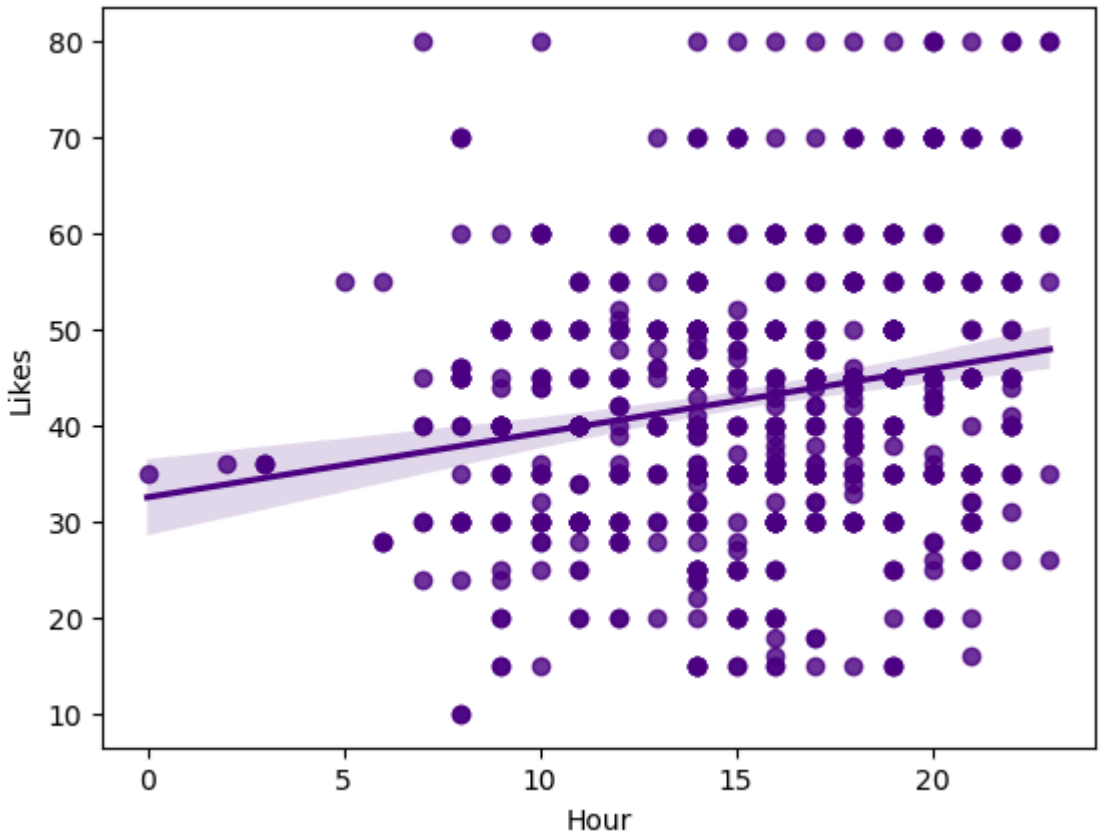
```
plt.ylabel("platform")
plt.title("likes_of_platform")
```

Out[71]:
```
Text(0.5, 1.0, 'likes_of_platform')
```



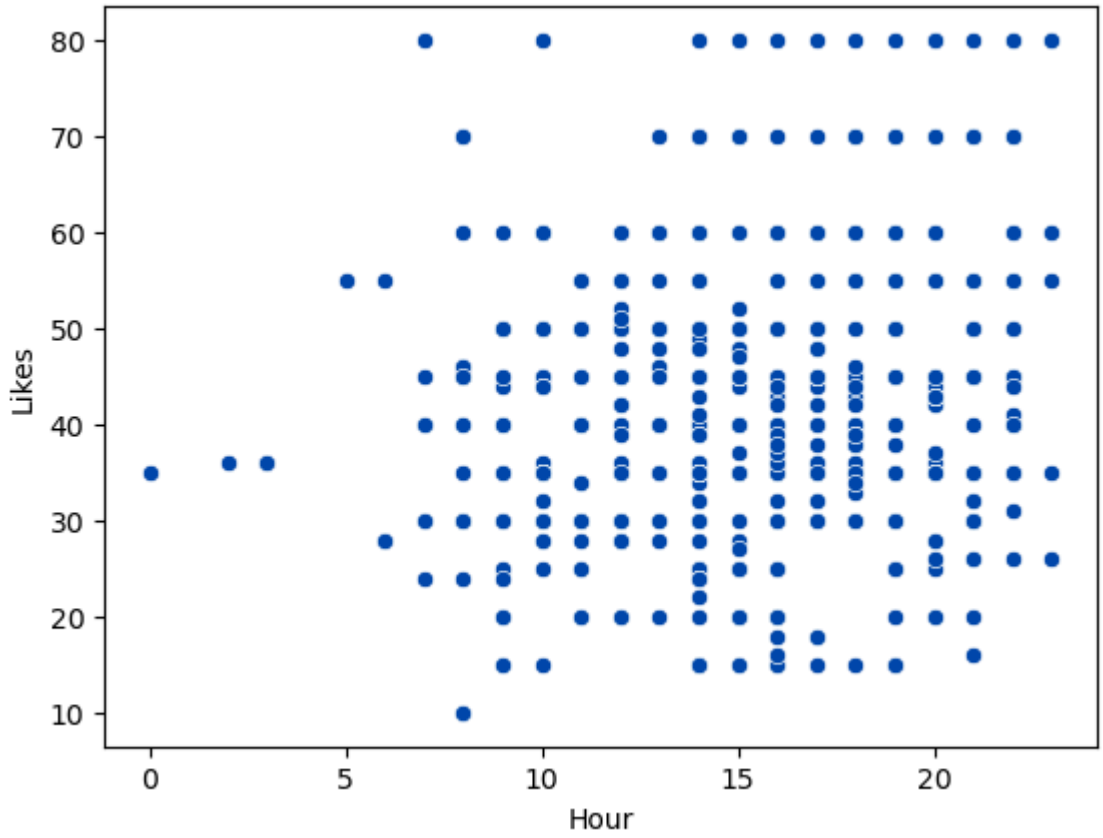
In [54]:
```
sns.regplot(data = df , x ='Hour' , y ='Likes', color='indigo')
```

Out[54]:
```
<Axes: xlabel='Hour', ylabel='Likes'>
```

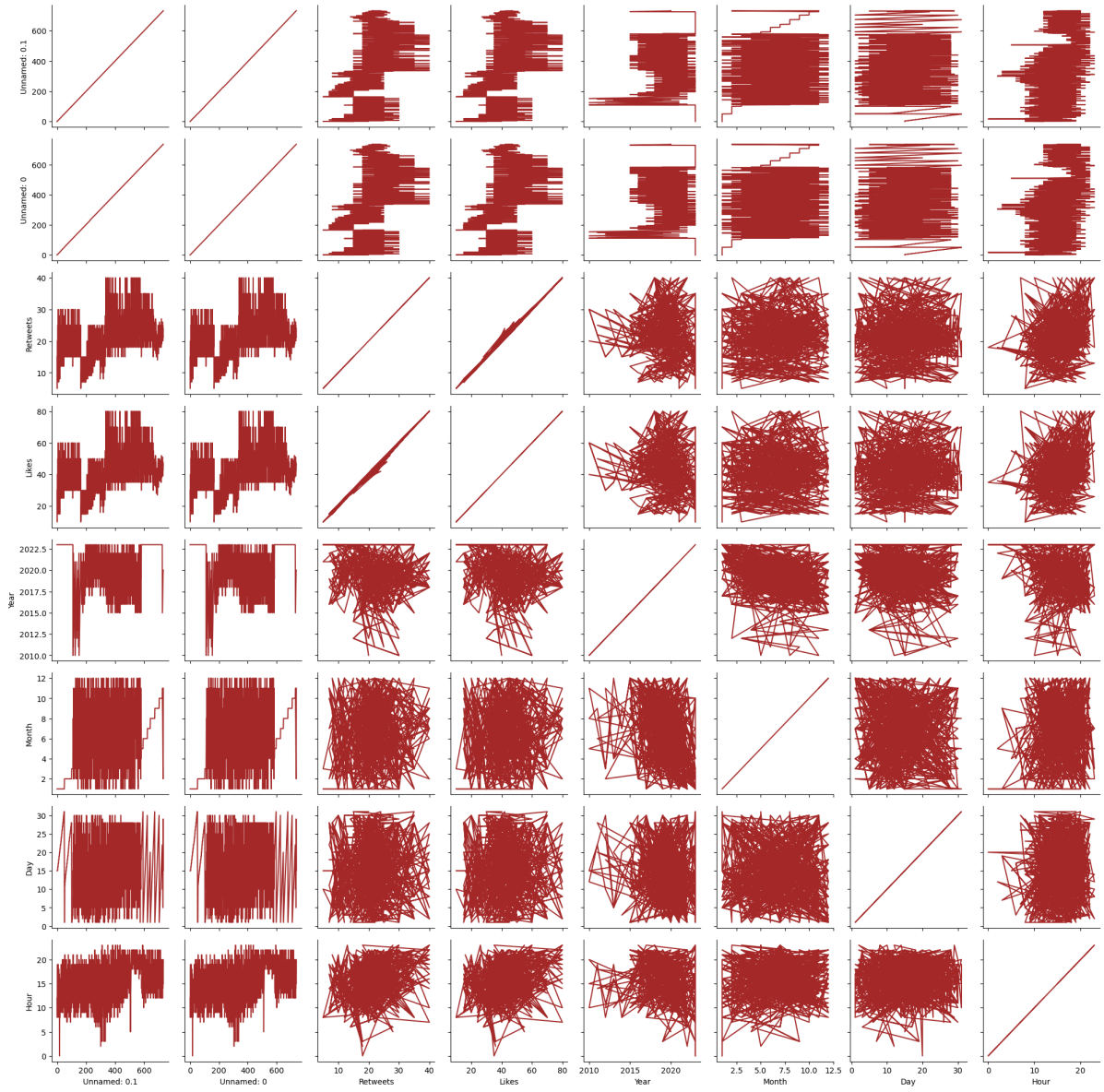


In [59]:
```
sns.scatterplot(data = df , x ='Hour' , y ='Likes',color='#0047AB')
```

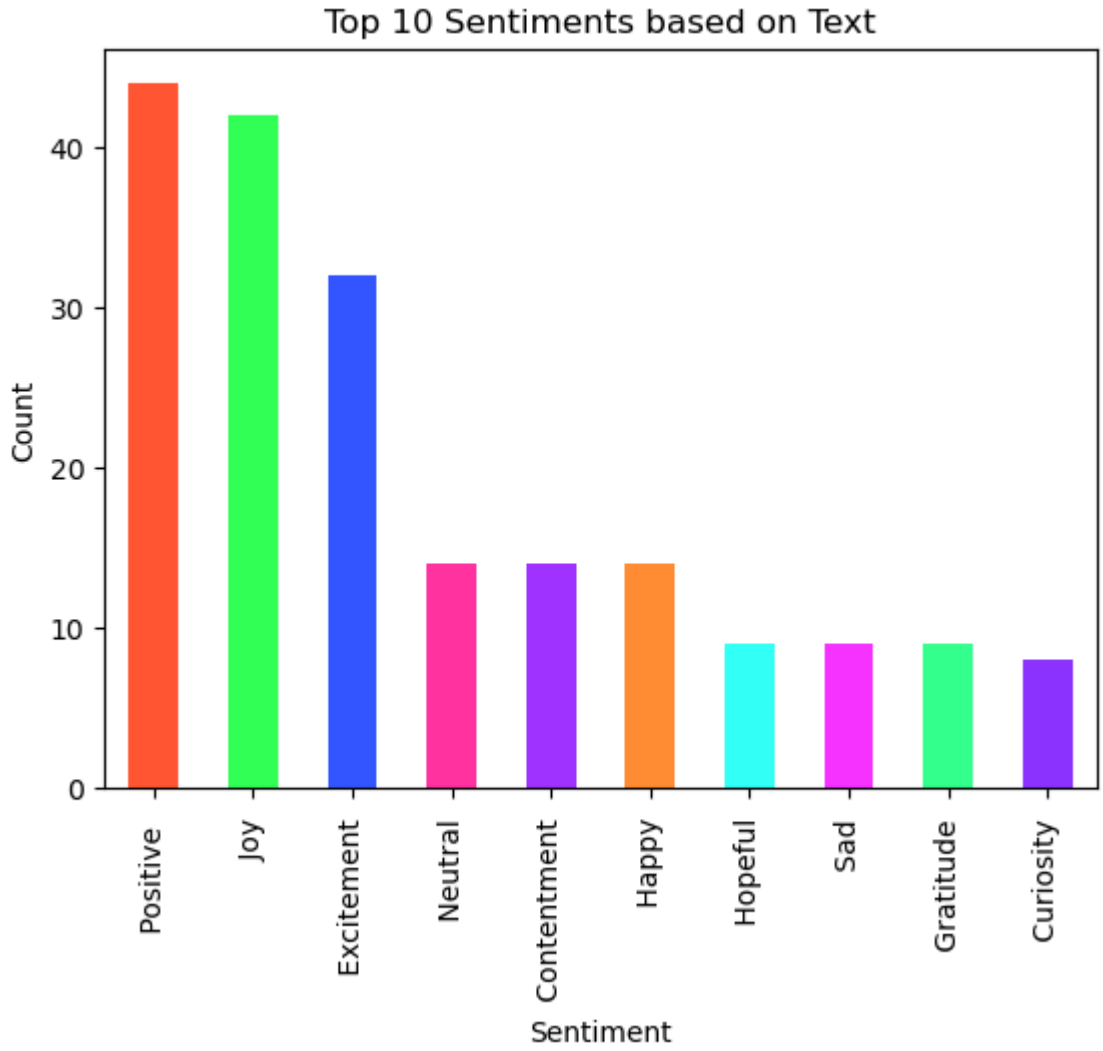
Out[59]:
```
<Axes: xlabel='Hour', ylabel='Likes'>
```



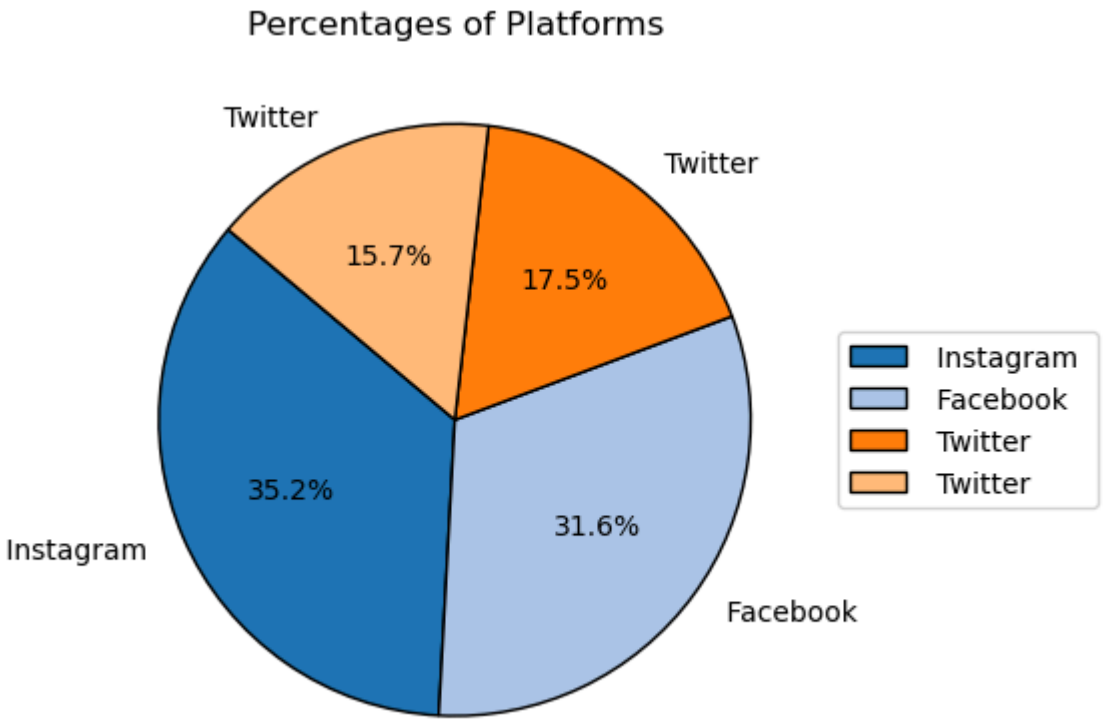
In [64]:
```
plot = sns.PairGrid(df)
plot.map(plt.plot, color= 'brown')
plt.show()
```



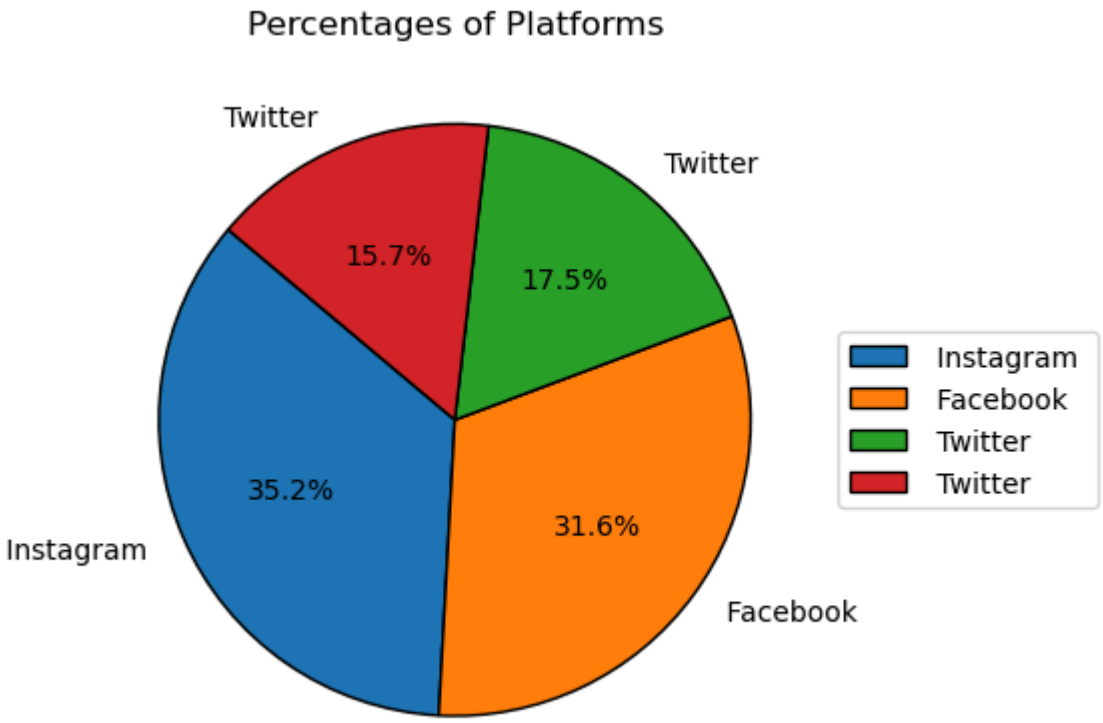
In [65]:
```
# Get the top 10 sentiments based on text
top_10_sentiments = df['Sentiment'].value_counts().nlargest(10)

# Define a list of colors for the top 10 sentiments
colors = ['#FF5733', '#33FF57', '#3357FF', '#FF33A1', '#A133FF', '#FF8C33', '#33FFF

# Plot the bar chart with unique colors for each sentiment
top_10_sentiments.plot(kind='bar', color=colors)
plt.title('Top 10 Sentiments based on Text')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```



In [66]:
```
# Get the value counts of the 'Platform' column
platform_counts = df['Platform'].value_counts()

# Define a list of colors for the platforms
colors = plt.get_cmap('tab20').colors  # Using a colormap with enough distinct colo

# Plot the pie chart
platform_counts.plot(kind='pie', autopct='%1.1f%%', colors=colors, startangle=140,

# Set the title and display the legend outside the pie chart
plt.title('Percentages of Platforms')
plt.legend(platform_counts.index, loc='center left', bbox_to_anchor=(1, 0.5))
plt.ylabel('')  # Remove the y-label to clean up the chart

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



In [67]:
```
# Get the value counts of the 'Platform' column
platform_counts = df['Platform'].value_counts()

# Define a list of colors for the platforms (using a colormap with enough distinct
colors = plt.get_cmap('tab10').colors  # 'tab10' has 10 distinct colors

# Plot the pie chart
platform_counts.plot(kind='pie', autopct='%1.1f%%', colors=colors, startangle=140,

# Set the title
plt.title('Percentages of Platforms')

# Position the legend outside the pie chart
plt.legend(platform_counts.index, loc='center left', bbox_to_anchor=(1, 0.5))

# Remove the y-label to clean up the chart
plt.ylabel('')

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



In [68]:
```
H_retweet= df.groupby('Hashtags')['Retweets'].max().value_counts().nlargest(10).sor
```