```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load Dataset

```
df=pd.read_csv('/content/social_media_performance.csv')
```

checking top 5 rows of dataset

Double-click (or enter) to edit

```
df.head(10)
```

	post_id	platform	content_type	topic	language	region	post_datetime	hashtags	sentiment_score	views	likes	comm
0	1	LinkedIn	article	Technology	UR	BR	2025-04-25 09:47:00	#AI #Innovation #TechTrends #Programming #Codi...	0.76	37781	1202	
1	2	LinkedIn	poll	Health	FR	JP	2025-10-29 09:44:00	#Fitness #Nutrition #Wellness #Health #MentalH...	0.46	23541	1399	
2	3	LinkedIn	article	Travel	HI	FR	2025-02-10 14:12:00	#Travel #Journey #Adventure #Tourism #ExploreM...	-0.01	30714	1663	
3	4	LinkedIn	image	Sports	DE	DE	2025-04-18 22:41:00	#Cricket #Workout #Fitness	0.55	31294	1372	
4	5	LinkedIn	poll	Business	DE	US	2025-04-28 10:17:00	#Entrepreneur #Leadership #StartupLife	0.70	43129	2234	
5	6	LinkedIn	image	Sports	FR	AU	2025-10-03 15:25:00	#Fitness #Training #Cricket #Basketball #Sports	0.23	11016	633	
6	7	LinkedIn	carousel	Food	HI	FR	2025-07-15 15:16:00	#Food #MustTry #Cooking #Recipe #Foodie #FoodBlog	0.49	1212	104	
7	8	LinkedIn	image	Fashion	ES	AU	2025-02-24 09:40:00	#FashionTips #Style #Lookbook #TrendingStyle #...	0.69	8331	737	
8	9	LinkedIn	image	Business	EN	BR	2025-10-18 02:31:00	#Leadership #Entrepreneur #Hustle	0.97	11821	460	
9	10	LinkedIn	poll	Health	FR	AU	2025-12-20 14:42:00	#Fitness #MentalHealth #HealthTips #Health #Nu...	0.41	37301	1170	

Next steps: Generate code with df | New interactive sheet

Understand the data structure

```
shape=df.shape
col=df.columns

print(shape)
print(col)
```

```
(10000, 15)
Index(['post_id', 'platform', 'content_type', 'topic', 'language', 'region',
       'post_datetime', 'hashtags', 'sentiment_score', 'views', 'likes',
       'comments', 'shares', 'engagement_rate', 'is_viral'],
      dtype='object')
```

```
info=df.info()
print(info)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   post_id          10000 non-null  int64  
 1   platform         10000 non-null  object 
 2   content_type     10000 non-null  object 
 3   topic            10000 non-null  object 
 4   language         10000 non-null  object 
 5   region           10000 non-null  object 
 6   post_datetime    10000 non-null  object 
 7   hashtags         10000 non-null  object 
 8   sentiment_score  10000 non-null  float64
 9   views            10000 non-null  int64  
 10  likes            10000 non-null  int64  
 11  comments         10000 non-null  int64  
 12  shares           10000 non-null  int64  
 13  engagement_rate  10000 non-null  float64
 14  is_viral         10000 non-null  int64  
dtypes: float64(2), int64(6), object(7)
memory usage: 1.1+ MB
None
```

```
describe=df.describe()
print(describe)
```

```
           post_id  sentiment_score          views         likes  \
count  10000.00000     10000.000000   10000.000000   10000.00000   
mean    5000.50000         0.440221  212142.158100   15236.87470   
std     2886.89568         0.452148  254526.880805   20450.03532   
min        1.00000        -1.000000     116.000000       3.00000   
25%     2500.75000         0.160000   30936.500000    1623.00000   
50%     5000.50000         0.560000   78193.000000    5504.00000   
75%     7500.25000         0.780000  343206.750000   21467.25000   
max    10000.00000         1.000000  999022.000000  116255.00000   

           comments        shares  engagement_rate      is_viral  
count  10000.000000  10000.000000     10000.000000  10000.000000  
mean    5860.139600   2343.759700         0.112667      0.545700  
std     7865.396933   3146.163176         0.064301      0.497932  
min        1.000000      0.000000         0.008300      0.000000  
25%      624.000000    249.000000         0.061900      0.000000  
50%     2116.500000    846.500000         0.102000      1.000000  
75%     8256.250000   3302.500000         0.151925      1.000000  
max    44713.000000  17885.000000         0.300000      1.000000  
```

Handling Missing Values

```
df.isnull().sum()
```

	0
post_id	0
platform	0
content_type	0
topic	0
language	0
region	0
post_datetime	0
hashtags	0
sentiment_score	0
views	0
likes	0
comments	0
shares	0
engagement_rate	0
is_viral	0

dtype: int64

% of Missing Value

```
df.isnull().sum()/len(df)*100
```

	0
post_id	0.0
platform	0.0
content_type	0.0
topic	0.0
language	0.0
region	0.0
post_datetime	0.0
hashtags	0.0
sentiment_score	0.0
views	0.0
likes	0.0
comments	0.0
shares	0.0
engagement_rate	0.0
is_viral	0.0

dtype: float64

Dropping irrelevant column

```
df.drop(columns='hashtags',inplace=True)
```

```
df.head()
```

	post_id	platform	content_type	topic	language	region	post_datetime	sentiment_score	views	likes	comments	shares
0	1	LinkedIn	article	Technology	UR	BR	2025-04-25 09:47:00	0.76	37781	1202	462	185
1	2	LinkedIn	poll	Health	FR	JP	2025-10-29 09:44:00	0.46	23541	1399	538	215
2	3	LinkedIn	article	Travel	HI	FR	2025-02-10 14:12:00	-0.01	30714	1663	639	255
3	4	LinkedIn	image	Sports	DE	DE	2025-04-18 22:41:00	0.55	31294	1372	528	211
4	5	LinkedIn	poll	Business	DE	US	2025-04-28 10:17:00	0.70	43129	2234	859	343

Next steps: Generate code with df | New interactive sheet

Handling categorical values

```
df.select_dtypes(include=['object']).columns
```

```
Index(['platform', 'content_type', 'topic', 'language', 'region',
       'post_datetime'],
      dtype='object')
```

Value count for categorical data

```
df['region'].value_counts()
```

region	count
FR	1076
MX	1071
US	1044
DE	1021
IN	995
AU	972
CA	969
JP	957
UK	952
BR	943

dtype: int64

```
df['region'].value_counts().head()
```

region	count
FR	1076
MX	1071
US	1044
DE	1021
IN	995

dtype: int64

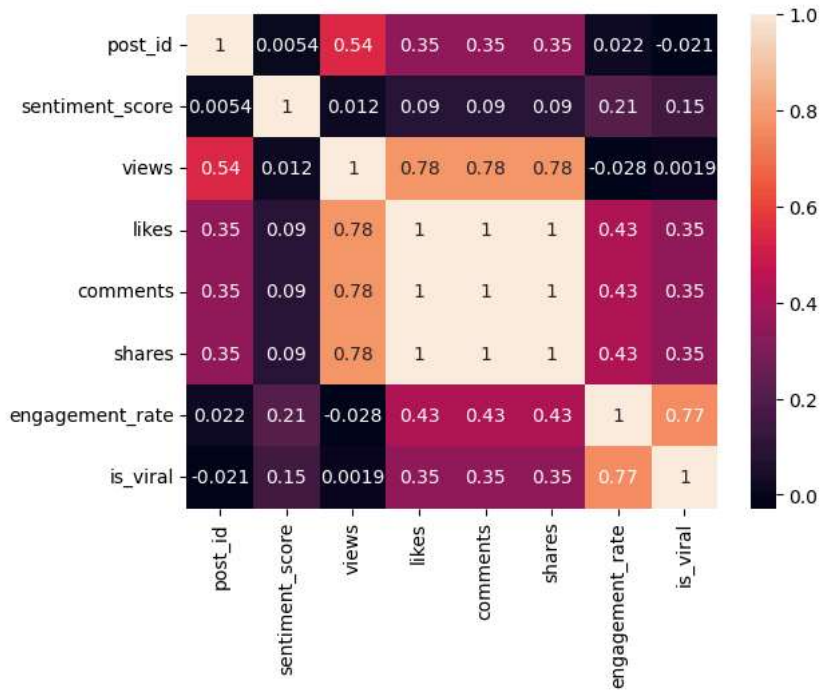
Correlation Analysis

```
df.corr(numeric_only=True)
```

	post_id	sentiment_score	views	likes	comments	shares	engagement_rate	is_viral
post_id	1.000000	0.005354	0.538471	0.350532	0.350532	0.350532	0.021521	-0.021168
sentiment_score	0.005354	1.000000	0.012239	0.090013	0.090013	0.090014	0.205524	0.152154
views	0.538471	0.012239	1.000000	0.778615	0.778615	0.778616	-0.028039	0.001916
likes	0.350532	0.090013	0.778615	1.000000	1.000000	1.000000	0.425684	0.348791
comments	0.350532	0.090013	0.778615	1.000000	1.000000	1.000000	0.425683	0.348791
shares	0.350532	0.090014	0.778616	1.000000	1.000000	1.000000	0.425683	0.348791
engagement_rate	0.021521	0.205524	-0.028039	0.425684	0.425683	0.425683	1.000000	0.766070
is_viral	-0.021168	0.152154	0.001916	0.348791	0.348791	0.348791	0.766070	1.000000

Heatmap Correlation Matrix

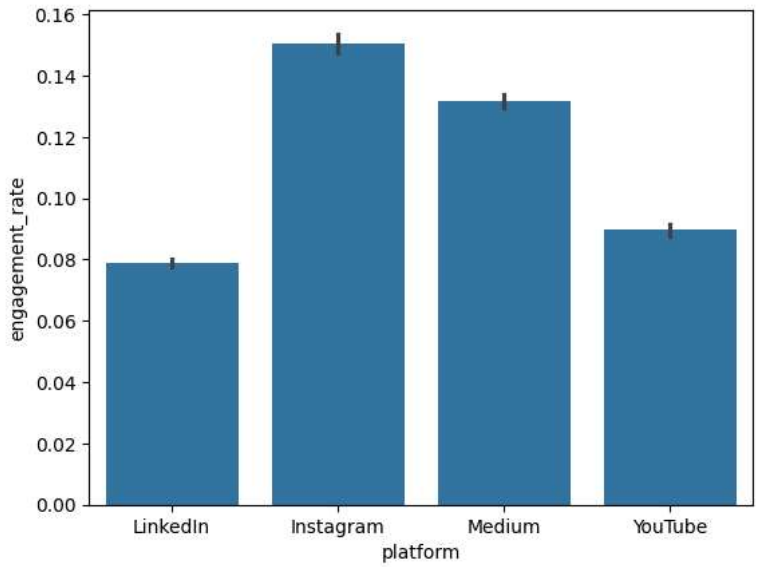
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(df.corr(numeric_only=True),annot=True)
plt.show()
```



Bar Plot

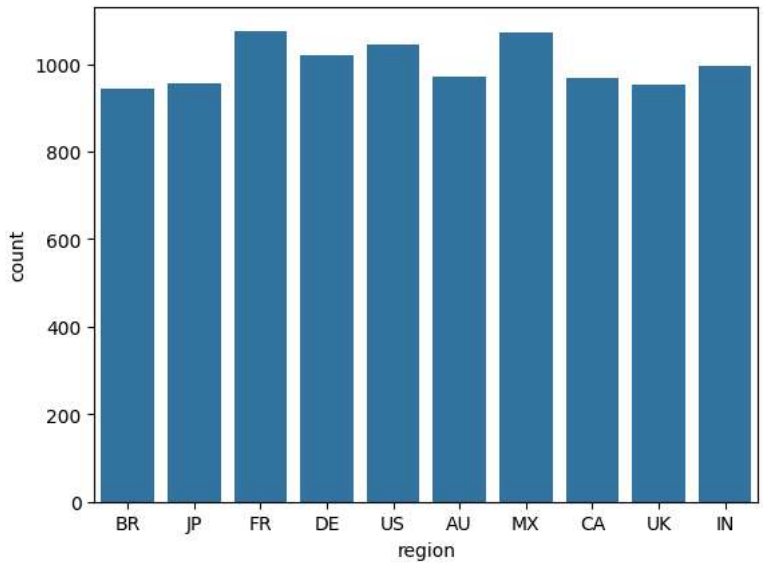
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv('/content/social_media_performance.csv')
```

```
sns.barplot(x='platform',y='engagement_rate',data=df)
plt.show()
```



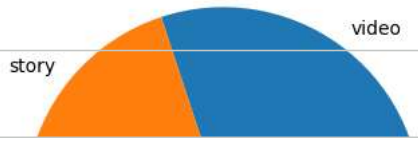
Count Plot

```
sns.countplot(x='region',data=df)
plt.show()
```



Pie Chart

```
df['content_type'].value_counts().plot(kind='pie')
plt.show()
```



Pair Plot

```
sns.pairplot(df)
plt.show()
```