```
import pandas as pd
import numpy as np
import matplotlib as plt

df = pd.read_csv("/content/ai_ghibli_trend_dataset_v2.csv")

df
```

	image_id	user_id	prompt	likes	shares	comments	platform	generation_time	gpu_usage	file_size_kb	resolution	styl
0	77ce5c72-eb45-4651-bcb1-c0677c0fceaf	6a7adf3d	Studio Ghibli-inspired ocean with giant fish	916	410	555	Reddit	4.80	49	1684	1024x1024	
1	7d66c67f-0d11-4ef9-895c-d865ef11fe40	523b8706	Ghibli-style village at sunset	2965	1361	417	Reddit	11.11	81	2808	1024x1024	
2	d7978afd-3932-4cce-9a21-5f9bf2bc1f64	0e02592a	A lone traveler exploring an enchanted ruin	4727	655	785	Instagram	5.56	41	1800	2048x2048	
3	cb34636a-a15c-4b15-999c-759dbb8896fe	9ed78a42	Spirited Away-style bustling market street	1629	1954	212	TikTok	12.45	88	479	2048x2048	
4	7511fbb8-db05-4584-a3a4-e8bb525ed58b	69ec8f02	Magical Ghibli forest with floating lanterns	2573	1281	913	TikTok	4.80	64	1789	512x512	
...	...	...	...	...	...	...	...	...	...	...	...	
495	135267ba-3941-42ae-a421-3be20e3856d1	c80e6866	Anime-style train passing through a fantasy world	1634	1328	624	TikTok	10.56	73	3255	1024x1024	
496	b2ced831-5b08-403d-bfb3-b562e256f359	81669630	Serene meadow with a tiny spirit creature	4198	833	812	Reddit	8.41	89	4710	1024x1024	
497	8c87b8aa-b304-43cf-82b3-3a199367ec17	258613ea	Ghibli-style mountain with floating islands	1237	1703	530	Reddit	12.05	86	1545	2048x2048	
498	04dba0f4-fdb7-4341-8bb6-dca5a595cd81	23b56439	Cozy tea shop in a mystical town, Ghibli style	1852	1158	259	Twitter	5.86	56	2796	512x512	
499	1960cd2d-0e99-4e09-b426-62212dd8b37f	7dea460a	Mysterious temple hidden in a magical forest	3944	1059	149	Twitter	5.81	35	1314	1024x1024	

500 rows × 16 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.isnull()
```

	image_id	user_id	prompt	likes	shares	comments	platform	generation_time	gpu_usage	file_size_kb	resolution	style_accura
0	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
495	False	False	False	False	False	False	False	False	False	False	False	
496	False	False	False	False	False	False	False	False	False	False	False	
497	False	False	False	False	False	False	False	False	False	False	False	
498	False	False	False	False	False	False	False	False	False	False	False	
499	False	False	False	False	False	False	False	False	False	False	False	

500 rows × 16 columns

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

	0
image_id	0
user_id	0
prompt	0
likes	0
shares	0
comments	0
platform	0
generation_time	0
gpu_usage	0
file_size_kb	0
resolution	0
style_accuracy_score	0
is_hand_edited	0
ethical_concerns_flag	0
creation_date	0
top_comment	0

dtype: int64

```
df.drop("image_id",axis =1,inplace = True)
```

```
df.drop("user_id",axis = 1,inplace = True)
```

```
df.drop("prompt",axis=1,inplace = True)
```

```
df.drop("top_comment",axis=1,inplace = True)
```

```
df.drop("creation_date",axis=1,inplace = True)
```

```
df
```

	likes	shares	comments	platform	generation_time	gpu_usage	file_size_kb	resolution	style_accuracy_score	is_hand_edited
0	916	410	555	Reddit	4.80	49	1684	1024x1024	89	Yes
1	2965	1361	417	Reddit	11.11	81	2808	1024x1024	92	Yes
2	4727	655	785	Instagram	5.56	41	1800	2048x2048	61	No
3	1629	1954	212	TikTok	12.45	88	479	2048x2048	76	No
4	2573	1281	913	TikTok	4.80	64	1789	512x512	58	No
...	...	...	...	...	...	...	...	...	...	...
495	1634	1328	624	TikTok	10.56	73	3255	1024x1024	96	Yes
496	4198	833	812	Reddit	8.41	89	4710	1024x1024	83	Yes
497	1237	1703	530	Reddit	12.05	86	1545	2048x2048	81	Yes
498	1852	1158	259	Twitter	5.86	56	2796	512x512	78	Yes
499	3944	1059	149	Twitter	5.81	35	1314	1024x1024	82	Yes

500 rows × 11 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.describe()
```

	likes	shares	comments	generation_time	gpu_usage	file_size_kb	style_accuracy_score
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	2601.262000	1040.182000	506.872000	8.317780	61.124000	2511.822000	74.626000
std	1429.433498	562.668738	283.384066	3.903103	18.151131	1390.178578	14.679001
min	105.000000	13.000000	5.000000	1.540000	30.000000	101.000000	50.000000
25%	1343.500000	587.750000	276.750000	5.027500	45.000000	1374.750000	62.000000
50%	2566.500000	1092.000000	518.000000	8.380000	63.000000	2498.000000	74.000000
75%	3913.250000	1502.000000	744.250000	11.540000	77.000000	3729.000000	87.250000
max	4944.000000	1999.000000	998.000000	14.990000	90.000000	4973.000000	100.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   likes                 500 non-null    int64
 1   shares                500 non-null    int64
 2   comments              500 non-null    int64
 3   platform              500 non-null    object
 4   generation_time       500 non-null    float64
 5   gpu_usage             500 non-null    int64
 6   file_size_kb          500 non-null    int64
 7   resolution            500 non-null    object
 8   style_accuracy_score  500 non-null    int64
 9   is_hand_edited        500 non-null    object
 10  ethical_concerns_flag 500 non-null    object
dtypes: float64(1), int64(6), object(4)
memory usage: 43.1+ KB
```

```
df.shape
```

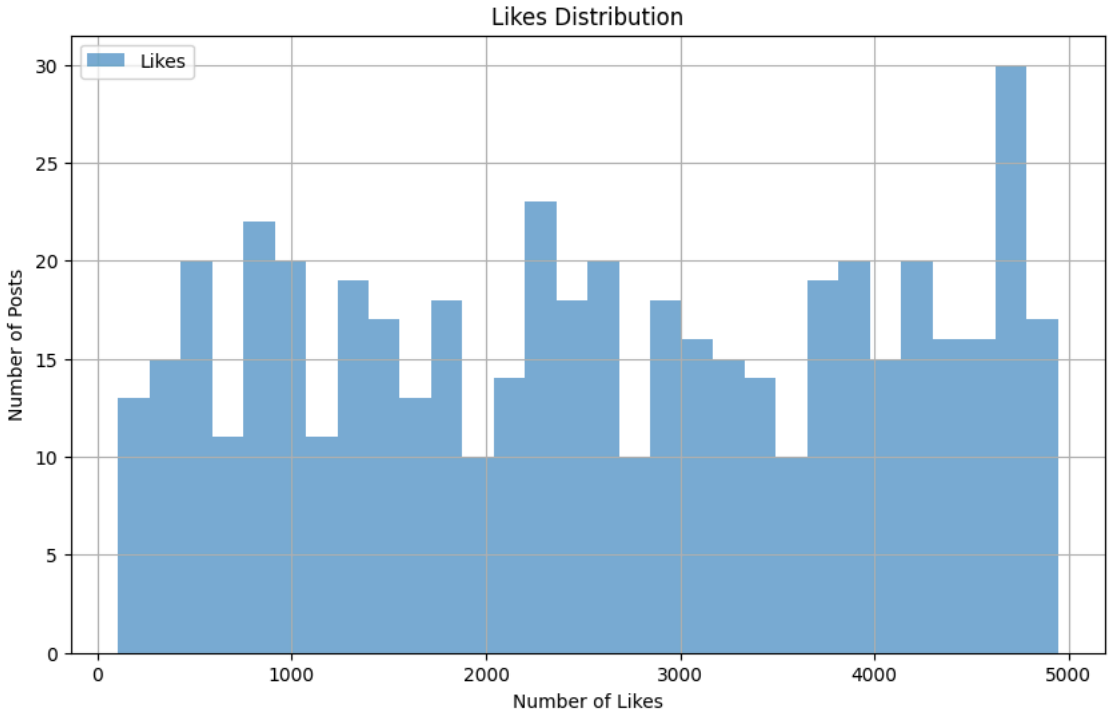
```
(500, 11)
```

```
df.size
```

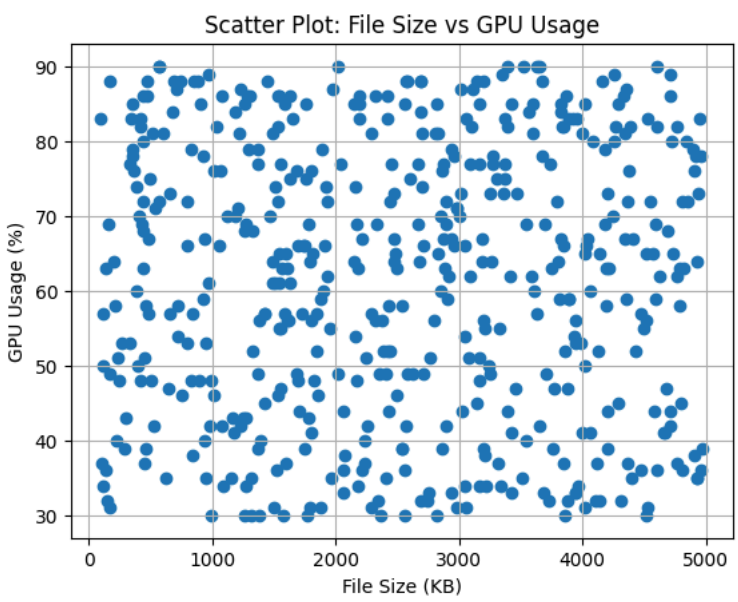
```
5500
```

```
import matplotlib.pyplot as plt



plt.figure(figsize=(10, 6))
plt.hist(df["likes"], bins=30, alpha=0.6, label='Likes')
plt.title("Likes Distribution")
plt.xlabel("Number of Likes")
```

```
plt.ylabel("Number of Posts")
plt.legend()
plt.grid(True)
plt.show()
```

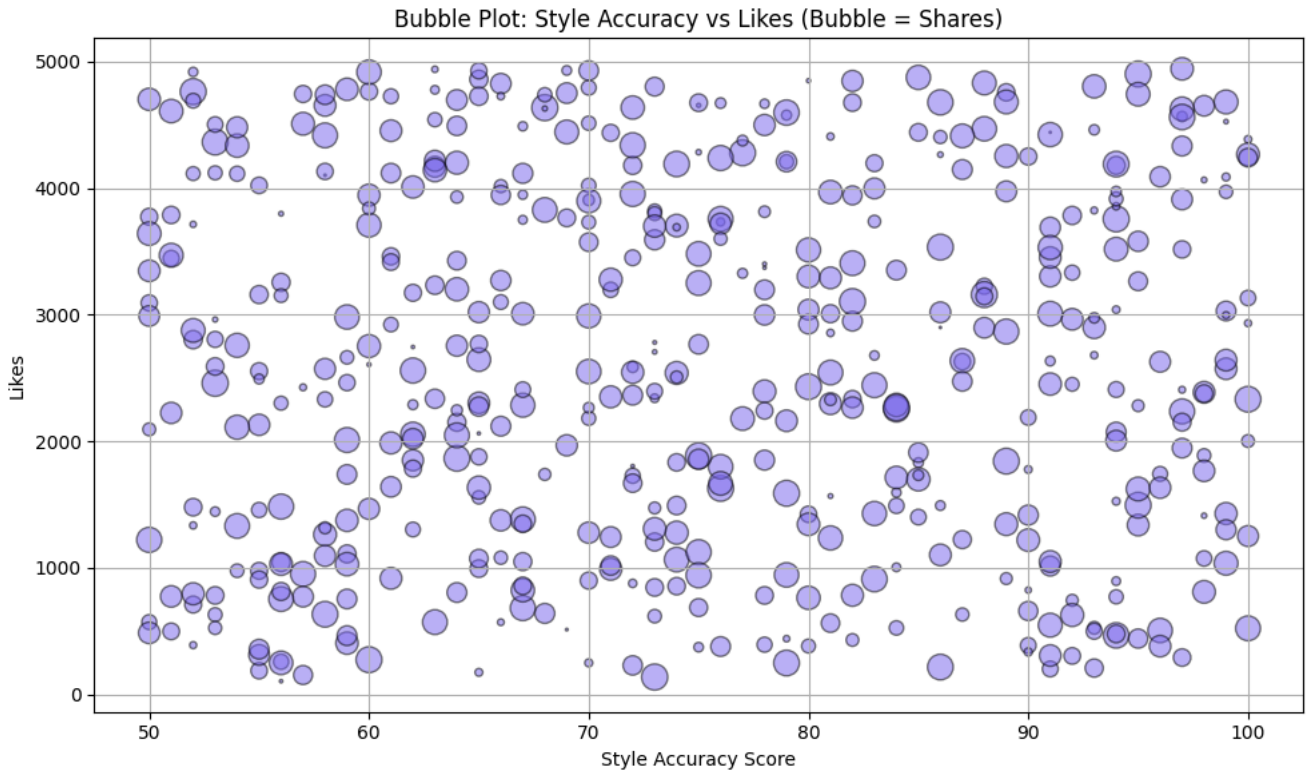



```
plt.scatter(df['file_size_kb'],df['gpu_usage'])
plt.xlabel("File Size (KB)")
plt.ylabel("GPU Usage (%)")
plt.title("Scatter Plot: File Size vs GPU Usage")
plt.grid(True)
plt.show()
```




```
plt.figure(figsize=(10, 6))
plt.scatter(
    df["style_accuracy_score"],
    df["likes"],
    s=df["shares"] / 10,
    alpha=0.5,
    color="mediumslateblue",
    edgecolors="black")
plt.title("Bubble Plot: Style Accuracy vs Likes (Bubble = Shares)")
plt.xlabel("Style Accuracy Score")
plt.ylabel("Likes")
plt.grid(True)
plt.tight_layout()
```

```
plt.show()
```


Bubble Plot: Style Accuracy vs Likes (Bubble = Shares)
Likes
Style Accuracy Score
0
1000
2000
3000
4000
5000
50
60
70
80
90
100