Top Instagram Influencers Analysis

Presented By- Varun Malhotra

CONTENTS

Objective

This project will analyze and predict insights on Instagram influencers using a dataset containing information on influencer ranking, engagement metrics, followers, likes, and more. This project targets a more advanced audience with 5 years of experience, so, it will involve detailed exploratory data analysis (EDA) and visualizations.

Data Overview

```
In [2]: data=pd.read_csv('top_insta_influencers_data.csv')
```

```
In [3]: data.head()
```

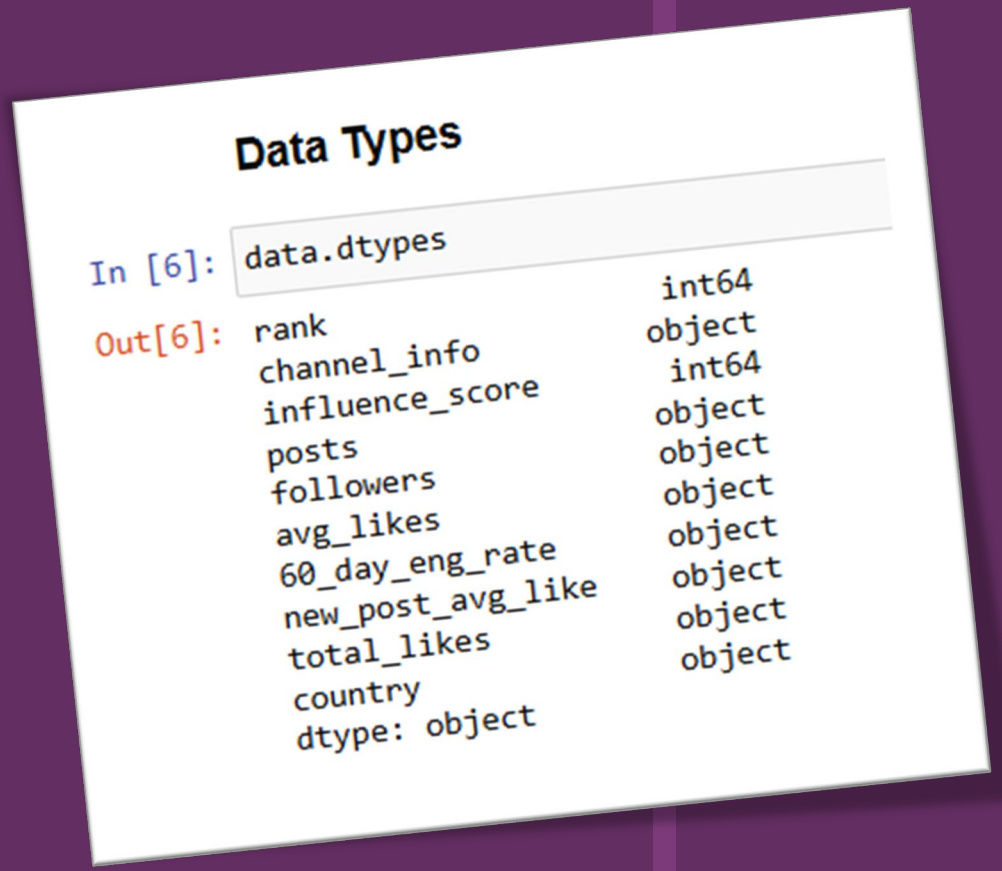
Out[3]:

	rank	channel_info	influence_score	posts	followers	avg_likes	60_day_eng_rate	new_post_avg_like	total_likes	country
0	1	cristiano	92	3.3k	475.8m	8.7m	1.39%	6.5m	29.0b	Spain
1	2	kyliejenner	91	6.9k	366.2m	8.3m	1.62%	5.9m	57.4b	United States
2	3	leomessi	90	0.89k	357.3m	6.8m	1.24%	4.4m	6.0b	NaN
3	4	selenagomez	93	1.8k	342.7m	6.2m	0.97%	3.3m	11.5b	United States
4	5	therock	91	6.8k	334.1m	1.9m	0.20%	665.3k	12.5b	United States

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

Out[4]:

	rank	channel_info	influence_score	posts	followers	avg_likes	60_day_eng_rate	new_post_avg_like	total_likes	country
195	196	iambeckyg	71	2.3k	33.2m	623.8k	1.40%	464.7k	1.4b	United States
196	197	nancyajram	81	3.8k	33.2m	390.4k	0.64%	208.0k	1.5b	France
197	198	luansantana	79	0.77k	33.2m	193.3k	0.26%	82.6k	149.2m	Brazil
198	199	nickjonas	78	2.3k	33.0m	719.6k	1.42%	467.7k	1.7b	United States
199	200	raisa6690	80	4.2k	32.8m	232.2k	0.30%	97.4k	969.1m	Indonesia



Columns Overview

- **rank**: Influencer rank
- **channel_info**: Instagram handle or channel information
- **influence_score**: Calculated influence score based on engagement and followers
- **posts**: Total number of posts made by the influencer
- **followers**: Number of followers
- **avg_likes**: Average likes per post
- **60_day_eng_rate**: Engagement rate over the past 60 days
- **new_post_avg_like**: Average likes on recent posts
- **total_likes**: Cumulative likes on all posts
- **country**: Influencer's country

Data Cleaning

```
In [8]: data.isnull().values.any()

Out[8]: True

In [9]: data = data.dropna(subset = ['country'])

In [10]: data.isnull().sum()

Out[10]: rank                0
         channel_info        0
         influence_score     0
         posts               0
         followers           0
         avg_likes           0
         60_day_eng_rate     0
         new_post_avg_like   0
         total_likes         0
         country             0
         dtype: int64

In [11]: data.isnull().values.any()

Out[11]: False
```

Data Preprocessing

```
In [14]: #Converting posts value into numeric format
posts=[]
for i in data['posts']:
    if 'k' in i:
        posts+=[int(float(i.replace('k',''))*1000)]
    else:
        posts+=[int(i)]
data['posts']=posts
```

```
In [15]: #Converting followers value into numeric format
followers=[]
for i in data['followers']:
    if 'm' in i:
        followers+=[int(float(i.replace('m',''))*1000000)]
    else:
        followers+=[int(i)]
data['followers']=followers
```

```
In [16]: #Converting avg likes value into numeric format
avg_likes=[]
for i in data['avg_likes']:
    if 'm' in i:
        avg_likes+=[int(float(i.replace('m',''))*1000000)]
    elif 'k' in i:
        avg_likes+=[int(float(i.replace('k',''))*1000)]
    else:
        avg_likes+=[int(i)]
data['avg_likes']=avg_likes
```

```
In [17]: #Converting total likes value into numeric format
total_likes=[]
for i in data['total_likes']:
    if 'b' in i:
        total_likes+=[int(float(i.replace('b',''))*1000000000)]
    elif 'm' in i:
        total_likes+=[int(float(i.replace('m',''))*1000000)]
    elif 'k' in i:
        total_likes+=[int(float(i.replace('k',''))*1000)]
    else:
        total_likes+=[int(i)]
data['total_likes']=total_likes
```

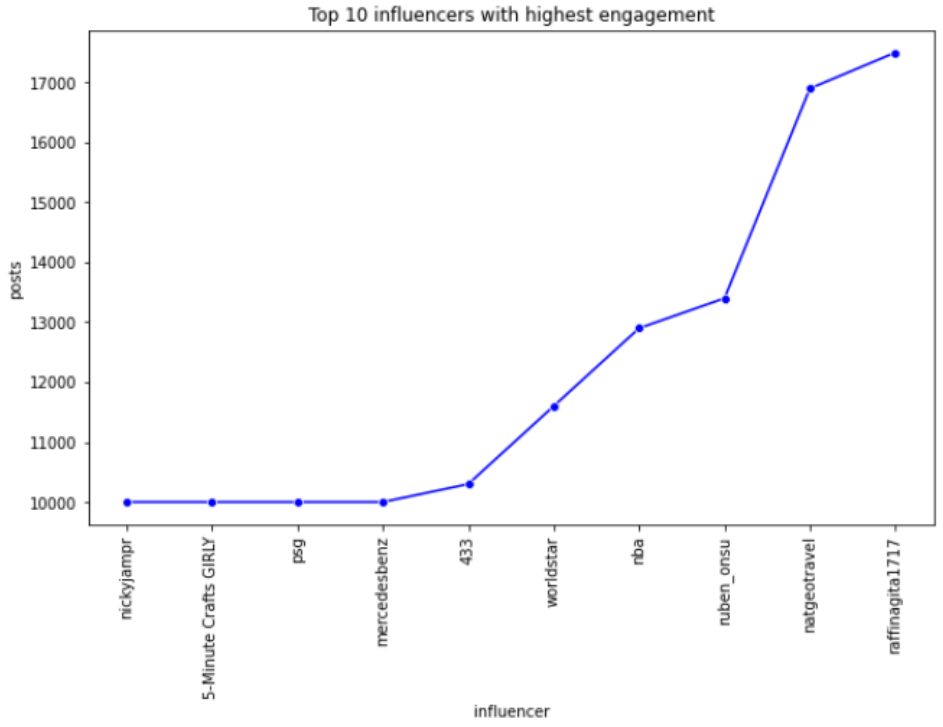
Data Types After typecasting →

```
In [18]: data.dtypes

Out[18]: rank                int64
         channel_info       object
         influence_score     int64
         posts               int64
         followers           int64
         avg_likes           int64
         60_day_eng_rate    object
         new_post_avg_like  object
         total_likes         int64
         country            object
         dtype: object
```

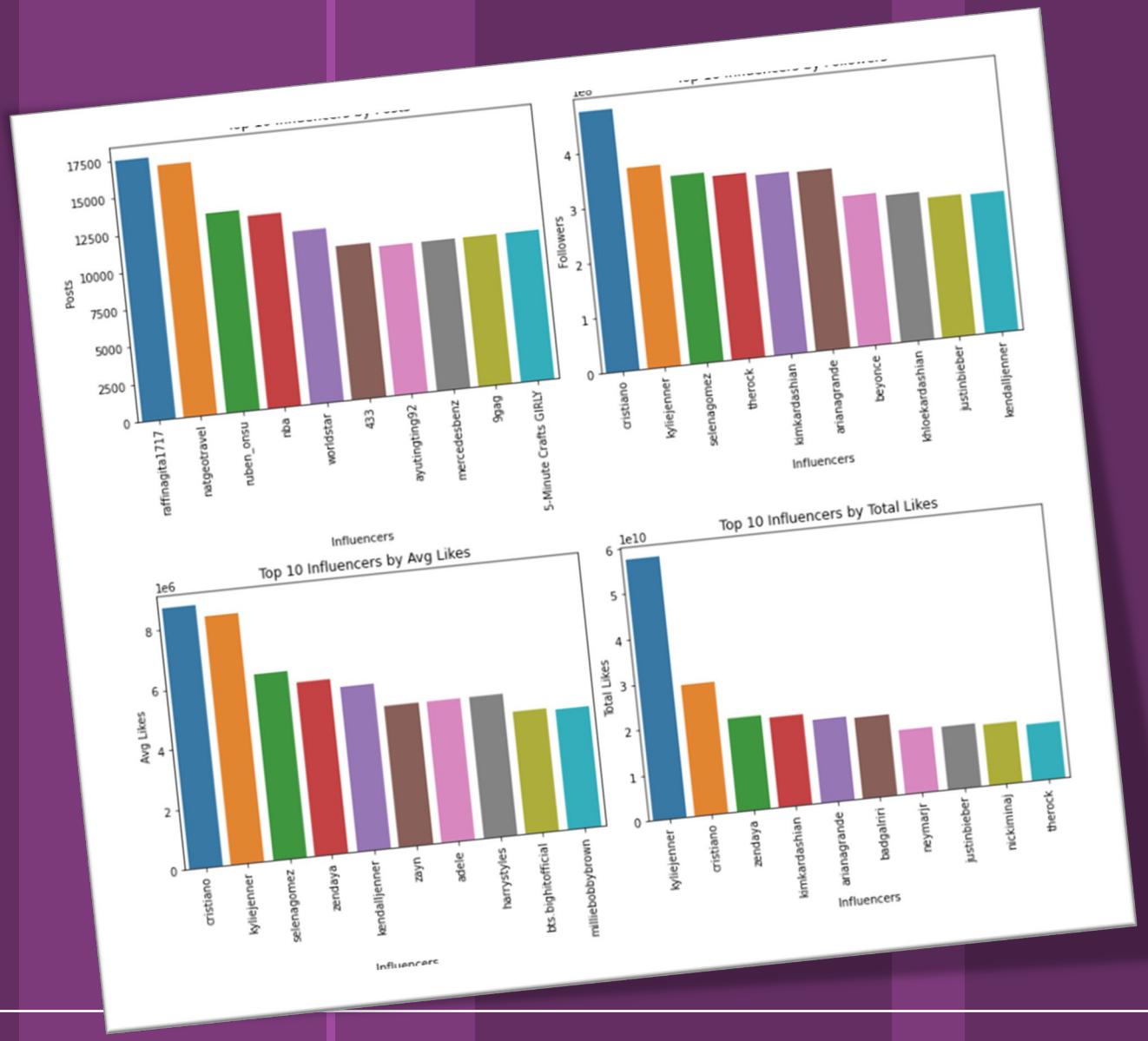
Exploratory Data Analysis

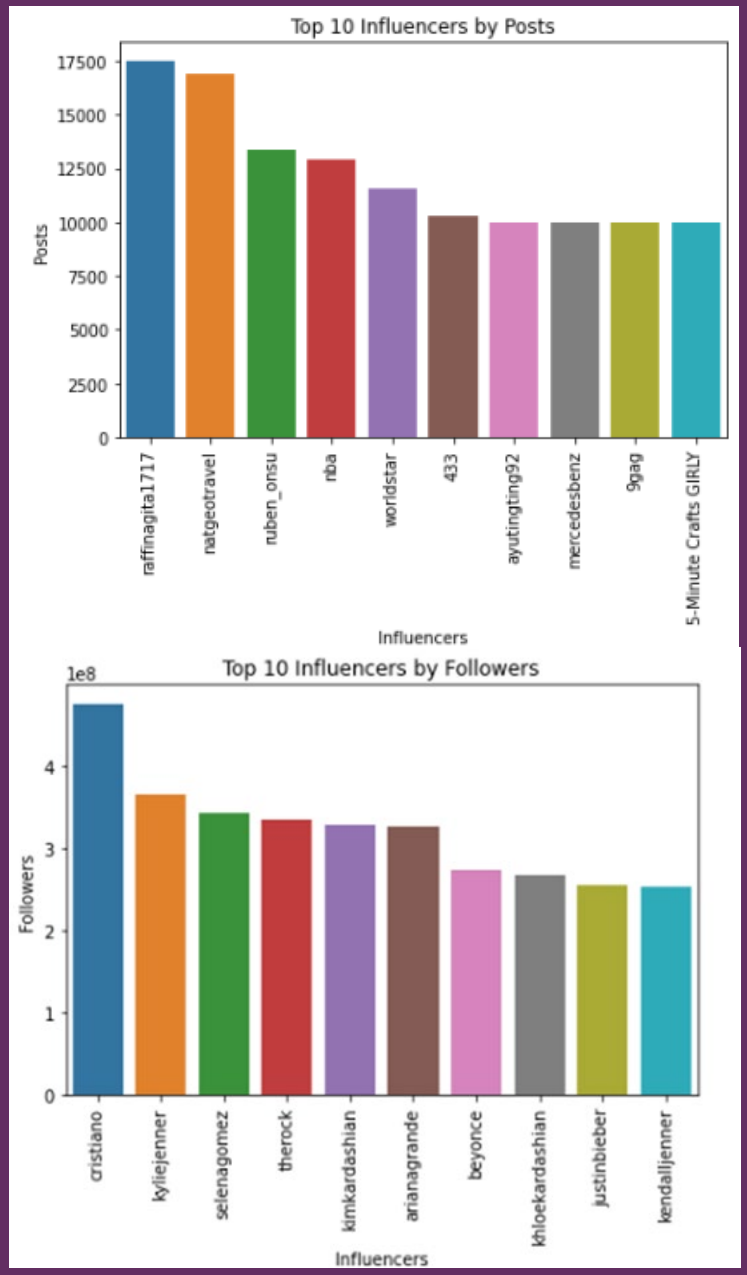
Line Chart

In [23]:
```
posts = data.groupby('channel_info')['posts'].mean().sort_values().tail(10)

# Plot
plt.figure(figsize=(10,6))
sns.lineplot(x=posts.index, y=posts.values, marker='o', color='blue')
plt.title('Top 10 influencers with highest engagement')
plt.xlabel('influencer')
plt.xticks(rotation=90)
plt.ylabel('posts')
plt.show()
```



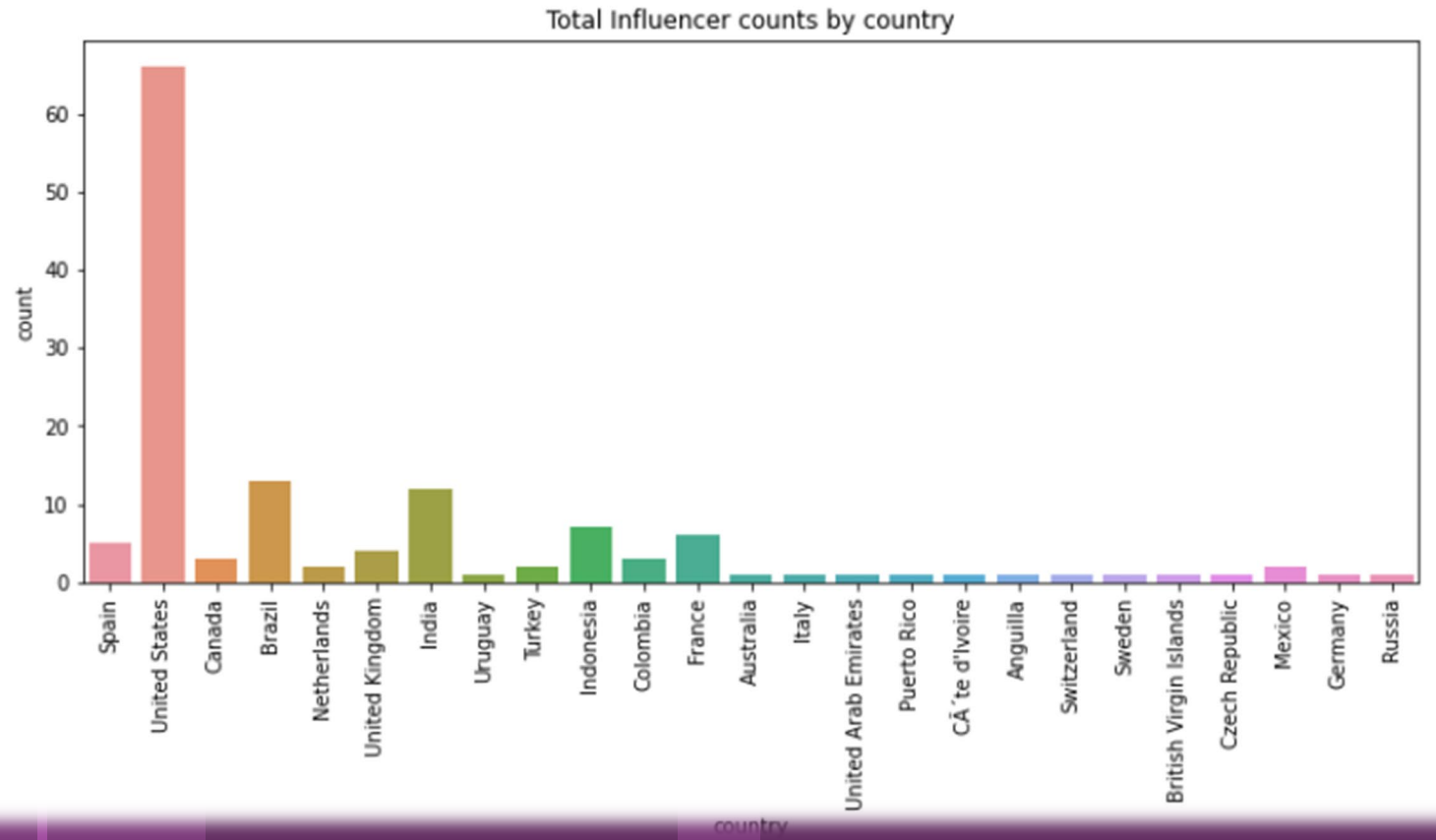
Bar Chart of Top 10 Influencers



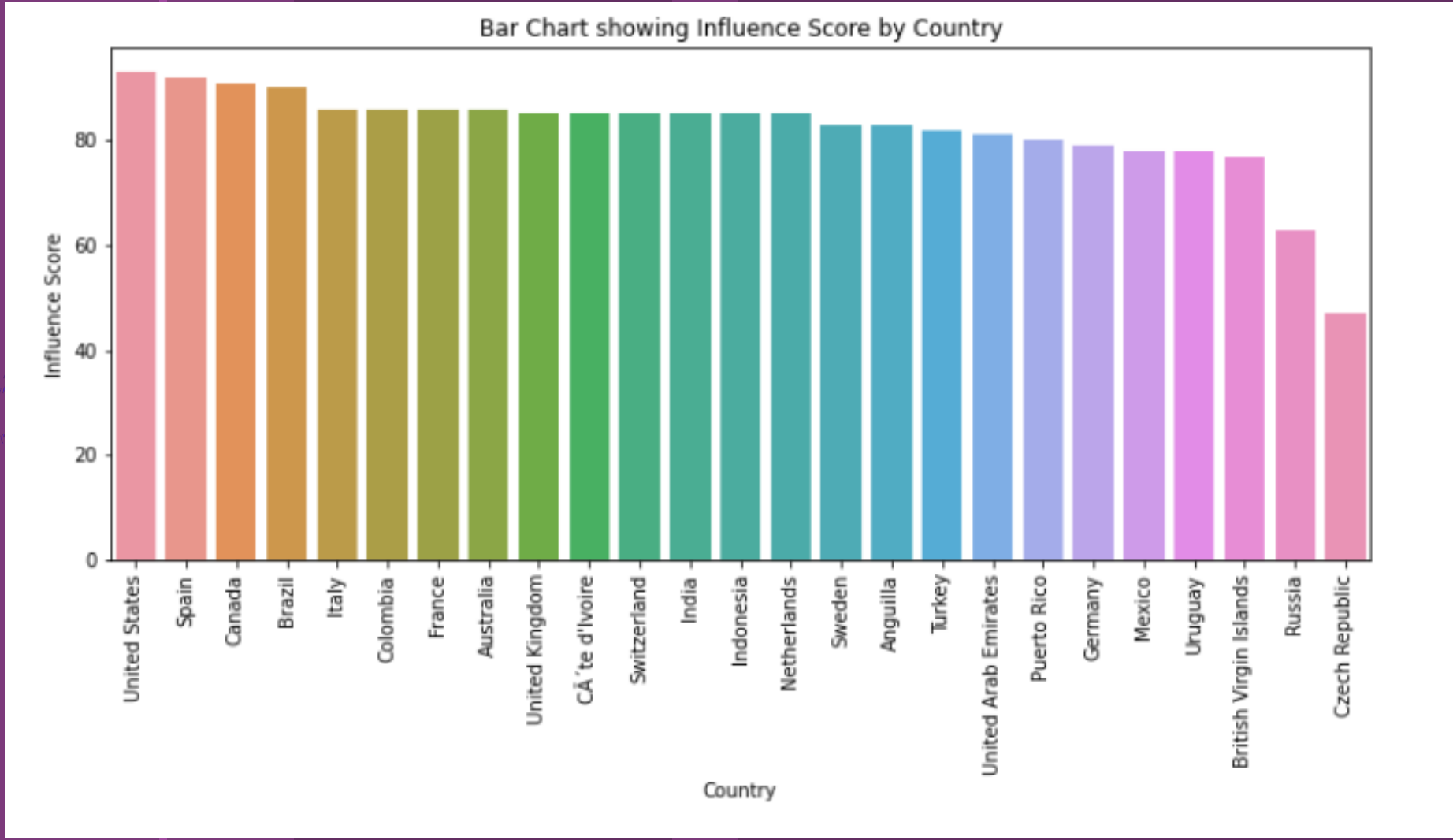


Influencers Count by Country

In [29]:
```
#count plot
plt.figure(figsize=(12, 5))
sns.countplot(data=data, x='country').set(title='Total Influencer counts by country')
plt.xticks(rotation=90)
plt.show()
```



Influence Score by Country



Summary

Influencer Rankings and Distribution:

- Analyzed top influencers, their reach, and engagement patterns.

Micro-Influencers with High Engagement:

- Discovered lower-followed influencers with significant engagement.

Country-based Trends:

- Identified which countries have the most active influencer bases and higher follower counts.

Growth Indicators:

- Showcased influencers with rising engagement rates on recent posts.

THANK YOU

Varun Malhotra

varunvirenmalu@gmail.com

https://github.com/ivarunmalhotra

https://www.linkedin.com/in/i-varunmalhotra/