In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
insta_data = pd.read_csv('instagram.csv')
```

In [3]:

```
insta_data.head()
```

Out[3]:

	Influencer insta name	instagram name	category_1	category_2	Followers	Audience country(mostly)	Auther engagement
0	433	433	Sports with a ball	NaN	48.5M	Spain	383
1	__youngbae__	TAEYANG	Music	NaN	12.7M	Indonesia	4
2	_agentgirl_	НАСТЯ ИВЛЕЕВА	Shows	NaN	18.8M	Russia	310
3	_imyour_joy	Joy	Lifestyle	NaN	13.5M	Indonesia	1
4	_jeongjaehyun	Jaehyun	NaN	NaN	11.1M	Indonesia	2

In [4]:

```
insta_data.tail()
```

Out[4]:

	Influencer insta name	instagram name	category_1	category_2	Followers	Audience country(mostly)	eng
995	zendaya	Zendaya	Cinema & Actors/actresses	Fashion	136.1M	United States	
996	zidane	zidane	Sports with a ball	NaN	31.2M	Spain	
997	zkdlin	KAI	Music	NaN	13.9M	Indonesia	
998	zoeisabellakravitz	Zoë Kravitz	Cinema & Actors/actresses	NaN	8.2M	United States	
999	zoesugg	Zoë Sugg	Lifestyle	Business & Careers	9.4M	United Kingdom	

In [5]:

```
insta_data.shape
```

Out[5]:

```
(1000, 8)
```

In [6]:

```
insta_data.columns
```

Out[6]:

```
Index(['Influencer insta name', 'instagram name', 'category_1', 'category_
2',
       'Followers', 'Audience country(mostly)', 'Authentic engagement\r
\n',
       'Engagement avg\r\n'],
      dtype='object')
```

In [7]:

```
insta_data.duplicated().sum()
```

Out[7]:

```
0
```

In [8]:

```
insta_data.isnull().sum()
```

Out[8]:

```
Influencer insta name         0
instagram name               21
category_1                  108
category_2                  713
Followers                     0
Audience country(mostly)     14
Authentic engagement\r\n      0
Engagement avg\r\n            0
dtype: int64
```

In [9]:

```
insta_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Influencer insta name     1000 non-null   object
 1   instagram name            979 non-null    object
 2   category_1                892 non-null    object
 3   category_2                287 non-null    object
 4   Followers                 1000 non-null   object
 5   Audience country(mostly)  986 non-null    object
 6   Authentic engagement
    1000 non-null   object
 7   Engagement avg
          1000 non-null   object
dtypes: object(8)
memory usage: 62.6+ KB
```

In [10]:

```
insta_data.describe()
```

Out[10]:

	Influencer insta name	instagram name	category_1	category_2	Followers	Audience country(mostly)	enga
count	1000	979	892	287	1000	986	
unique	997	975	31	27	411	32	
top	angelinajolie	Bruno Goes 🎭	Music	Cinema & Actors/actresses	6M	United States	
freq	2	2	235	59	11	279	

In [11]:

```
insta_data.drop_duplicates(subset=['Influencer insta name'],inplace=True)
```

In [12]:

```
insta_data.rename({'category_1':'Category','Audience country(mostly)':'Audience Country'
insta_data.rename({'Subscribers':'Followers',},axis=1,inplace=True)
```

In [13]:

```
insta_data = insta_data.drop(['Influencer insta name','Authentic engagement\r\n'],axis=1
```

In [14]:

```
insta_data.head()
```

Out[14]:

	instagram name	Category	category_2	Followers	Audience Country	Engagement avg\r\n
0	433	Sports with a ball	NaN	48.5M	Spain	637K
1	TAEYANG	Music	NaN	12.7M	Indonesia	542.3K
2	НАСТЯ ИВЛЕЕВА	Shows	NaN	18.8M	Russia	377.9K
3	Joy	Lifestyle	NaN	13.5M	Indonesia	1.4M
4	Jaehyun	NaN	NaN	11.1M	Indonesia	3.1M

In [15]:

```
insta =['Followers','Engagement avg\r\n']
```

In [16]:

```
import re
```

In [17]:

```
def convert(x):
    return re.findall('\d+\.?\d*',x)
```

In [18]:

```
def update(data,data_update):
    for i in data_update:
        data['new'+i]=data[i].apply(convert)
        data['new'+i]=data['new'+i].apply(lambda x: "".join(x))
        data['new'+i]=pd.to_numeric(data['new'+i])
        data['new'+i]=np.where(['M' in j for j in data[i]],data['new'+i]*1000000,
                               np.where(['K' in j1 for j1 in data[i]],data['new'+i]*1000,d
    return data
```

In [20]:

```
update(insta_data,insta)
```

Out[20]:

	instagram name	Category	category_2	Followers	Audience Country	Engagement avg\r\n	newFollowers
0	433	Sports with a ball	NaN	48.5M	Spain	637K	48500000.(
1	TAEYANG	Music	NaN	12.7M	Indonesia	542.3K	12700000.(
2	НАСТЯ ИВЛЕЕВА	Shows	NaN	18.8M	Russia	377.9K	18800000.(
3	Joy	Lifestyle	NaN	13.5M	Indonesia	1.4M	13500000.(
4	Jaehyun	NaN	NaN	11.1M	Indonesia	3.1M	11100000.(
...	...	...	...	...	...	...	..
995	Zendaya	Cinema & Actors/actresses	Fashion	136.1M	United States	8.6M	136100000.(
996	zidane	Sports with a ball	NaN	31.2M	Spain	744.1K	31200000.(
997	KAI	Music	NaN	13.9M	Indonesia	1.6M	13900000.(
998	Zoë Kravitz	Cinema & Actors/actresses	NaN	8.2M	United States	885.5K	8200000.(
999	Zoë Sugg	Lifestyle	Business & Careers	9.4M	United Kingdom	289.3K	9400000.(

997 rows × 8 columns

In [21]:

```
insta_data['Engagement Rate']=np.round((insta_data['newEngagement avg\r\n']/insta_data['
```

In [22]:

```
insta_data['Followers'].unique()
```

Out[22]:

```
array(['48.5M', '12.7M', '18.8M', '13.5M', '11.1M', '7.9M', '25M', '3M',
       '4.9M', '4.1M', '13.2M', '6.5M', '19.8M', '27.9M', '58.3M',
       '17.2M', '7.2M', '8.8M', '28.9M', '8.2M', '13.6M', '40.5M',
       '49.7M', '34.2M', '2.6M', '8.3M', '31.5M', '10M', '12M', '61.1M',
       '5.1M', '10.8M', '22.4M', '13.4M', '15.8M', '21.1M', '61.8M', '8M',
       '9.1M', '6.8M', '17.6M', '27M', '16.2M', '12.4M', '9.2M', '5.6M',
       '8.1M', '24.1M', '15.2M', '6.7M', '30M', '6.3M', '6.1M', '21.9M',
       '5.5M', '7.7M', '39M', '16.3M', '12.6M', '13.9M', '14.9M', '20.3M',
       '17.8M', '22.6M', '35.5M', '19M', '11.4M', '34.5M', '57.5M',
       '10.7M', '8.5M', '27.4M', '13M', '302.3M', '9.7M', '28M', '9.5M',
       '21.5M', '6.6M', '14.6M', '13.3M', '21.4M', '23.1M', '14.1M',
       '25.3M', '30.5M', '38.5M', '125.1M', '10.4M', '16M', '35.4M',
       '20.2M', '7.1M', '50.7M', '50.6M', '14.3M', '25.2M', '16.8M',
       '246.9M', '16.4M', '14.2M', '17.7M', '7.8M', '101.7M', '11.5M',
       '46.2M', '32M', '18.4M', '33.1M', '3.4M', '18.2M', '23.7M',
       '22.1M', '18.3M', '11.9M', '5.8M', '40.2M', '42.6M', '24.9M',
       '7.5M', '61.7M', '16.1M', '9M', '20.5M', '9.4M', '18.6M', '23M',
       '60.9M', '26.4M', '26.8M', '12.8M', '9.3M', '43.3M', '24.8M',
       '15M', '14M', '9.8M', '8.9M', '91.4M', '17M', '47.7M', '15.7M',
       '33.2M', '26.7M', '20.1M', '106M', '14.8M', '54M', '37.4M',
       '31.3M', '3.2M', '7.4M', '6.4M', '15.4M', '5M', '36M', '5.7M',
       '12.1M', '419.6M', '19.2M', '32.9M', '4.3M', '12.9M', '44.9M',
       '20.6M', '32.8M', '35.7M', '4.8M', '33.5M', '13.7M', '71.9M',
       '21.2M', '23.3M', '15.9M', '128.7M', '10.1M', '6M', '16.7M',
       '65.5M', '4.5M', '4.2M', '49.6M', '18.9M', '25.1M', '6.2M',
       '25.8M', '5.9M', '3.7M', '14.7M', '46.4M', '10.2M', '81.6M',
       '7.3M', '11M', '22.7M', '10.3M', '18.1M', '29.6M', '35.6M',
       '15.3M', '29M', '9.6M', '7M', '107.1M', '31.8M', '21M', '25.5M',
       '3.6M', '26.1M', '10.6M', '24.5M', '16.6M', '77M', '46.3M', '3.1M',
       '36.5M', '19.3M', '22.2M', '12.3M', '73.2M', '20.4M', '47.9M',
       '42.9M', '2.7M', '42.2M', '27.3M', '18.7M', '10.5M', '19.7M',
       '32.7M', '41.9M', '11.8M', '11.3M', '31.2M', '127.2M', '28.6M',
       '52.9M', '22.5M', '23.9M', '4M', '29.4M', '487.2M', '17.9M',
       '27.8M', '5.3M', '59.3M', '19.4M', '14.5M', '19.5M', '49.1M',
       '41.2M', '51.9M', '64.8M', '12.5M', '40.1M', '11.7M', '21.3M',
       '200.8M', '15.5M', '19.1M', '21.7M', '45.8M', '3.5M', '4.7M',
       '12.2M', '30.2M', '33.7M', '9.9M', '35.8M', '227M', '64.2M',
       '5.2M', '68.9M', '39.8M', '47.4M', '50.8M', '23.8M', '25.7M',
       '62.9M', '156.6M', '227.4M', '140.2M', '33.3M', '73.9M', '230.2M',
       '33.9M', '296.4M', '11.6M', '116.6M', '8.7M', '166.4M', '46.6M',
       '47.3M', '11.2M', '323.3M', '75.7M', '31.6M', '10.9M', '44.8M',
       '315.4M', '52.8M', '26.5M', '27.5M', '8.4M', '26.9M', '36.7M',
       '19.6M', '16.9M', '24.3M', '28.2M', '31.7M', '43M', '21.6M',
       '8.6M', '24M', '30.6M', '42M', '61.6M', '56.5M', '53M', '19.9M',
       '29.2M', '15.6M', '62.7M', '36.1M', '35.3M', '27.7M', '163.8M',
       '48.2M', '3.3M', '26.6M', '4.6M', '4.4M', '49M', '18M', '32.4M',
       '66.5M', '75.3M', '6.9M', '37.5M', '214.6M', '45.4M', '65.8M',
       '68.5M', '29.5M', '29.9M', '172M', '181.6M', '32.3M', '42.7M',
       '208.9M', '44.3M', '38.6M', '43.5M', '5.4M', '22.9M', '33.6M',
       '35.2M', '7.6M', '22.3M', '55.4M', '58M', '20.8M', '22.8M',
       '60.6M', '30.4M', '38.9M', '17.5M', '113M', '27.6M', '30.3M',
       '13.8M', '65.1M', '37.9M', '27.1M', '20.9M', '39.7M', '30.8M',
       '13.1M', '29.3M', '308.2M', '67.4M', '32.6M', '70.5M', '71.8M',
       '31.9M', '58.6M', '41.1M', '51.5M', '22M', '52.1M', '204.7M',
       '37.7M', '116.8M', '45M', '307M', '38.1M', '33.8M', '17.4M',
       '24.6M', '32.2M', '42.4M', '40.9M', '21.8M', '79M', '188.1M',
       '57.9M', '62.4M', '3.8M', '17.1M', '136.1M'], dtype=object)
```

In [23]:

```
insta_data['Followers'].value_counts()
```

Out[23]:

```
6M        11
9.7M      10
13.9M      9
5.5M       9
8.2M       9
          ..
41.2M      1
64.8M      1
40.1M      1
21.3M      1
136.1M     1
Name: Followers, Length: 411, dtype: int64
```

In [24]:

```
insta_data['Followers'].str[-1].unique()
```

Out[24]:

```
array(['M'], dtype=object)
```

In [25]:

```
insta_data['newFollowers']=insta_data['newFollowers']/1000000
```

In [26]:

```
insta_data = insta_data.drop(['Engagement avg\r\n','newEngagement avg\r\n'],axis=1)
```

In [27]:

```
insta_data = insta_data.drop(['category_2'],axis=1)
```

In [28]:

```
insta_data.head()
```

Out[28]:

	instagram name	Category	Followers	Audience Country	newFollowers	Engagement Rate
0	433	Sports with a ball	48.5M	Spain	48.5	1.313
1	TAEYANG	Music	12.7M	Indonesia	12.7	4.270
2	НАСТЯ ИВЛЕЕВА	Shows	18.8M	Russia	18.8	2.010
3	Joy	Lifestyle	13.5M	Indonesia	13.5	10.370
4	Jaehyun	NaN	11.1M	Indonesia	11.1	27.928

In [29]:

```
insta_data.sort_values(by='newFollowers',ascending=False,ignore_index=True)
```

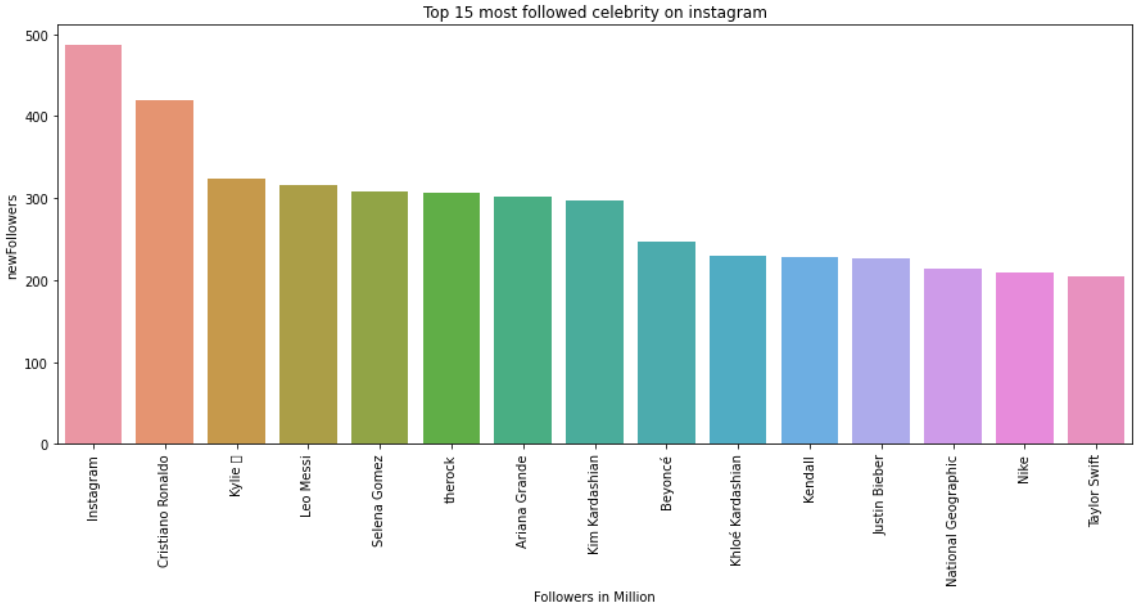
Out[29]:

	instagram name	Category	Followers	Audience Country	newFollowers	Engagement Rate
0	Instagram	Photography	487.2M	India	487.2	0.096
1	Cristiano Ronaldo	Sports with a ball	419.6M	India	419.6	1.668
2	Kylie ♡	Fashion	323.3M	United States	323.3	3.805
3	Leo Messi	Sports with a ball	315.4M	Argentina	315.4	1.680
4	Selena Gomez	Music	308.2M	United States	308.2	1.428
...	...	...	...	...	...	...
992	Drew Starkey	NaN	3.2M	United States	3.2	37.500
993	GeorgeNotFound	NaN	3.1M	United States	3.1	35.484
994	설인아 SEORINA	Lifestyle	3M	South Korea	3.0	33.260
995	HAECHAN	NaN	2.7M	NaN	2.7	77.778
996	NaN	NaN	2.6M	United States	2.6	42.308

997 rows × 6 columns

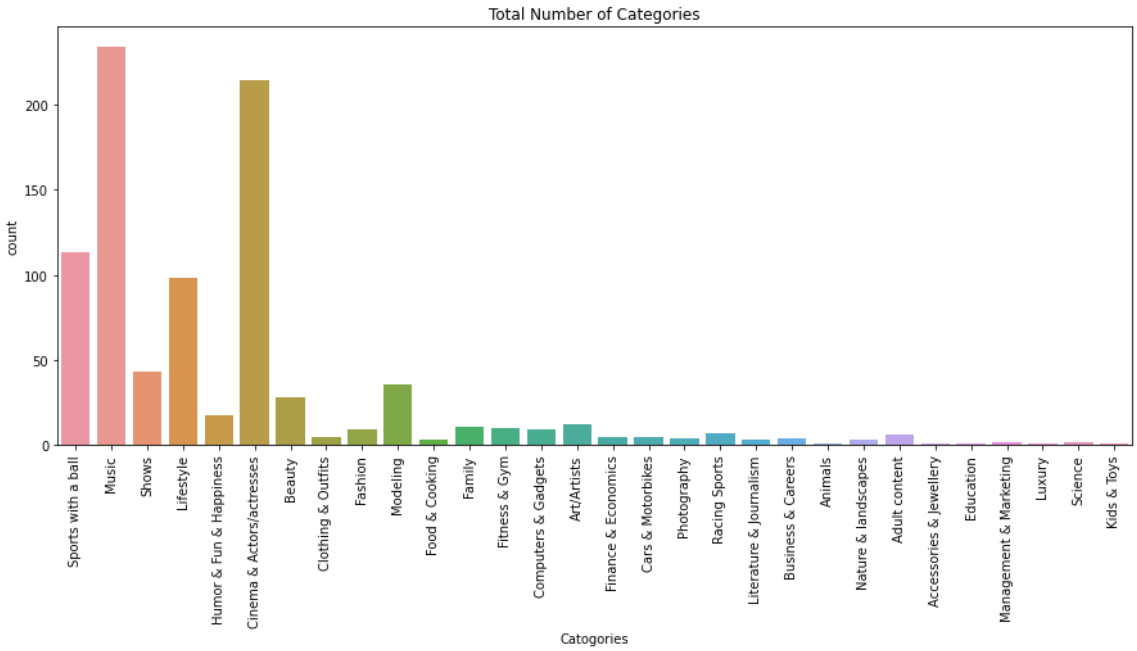
In [30]:

```
plt.figure(figsize=(15,6))
sns.barplot(x='instagram name',y='newFollowers',data=insta_data.sort_values(by='newFollo
plt.xticks(rotation = 90)
plt.title('Top 15 most followed celebrity on instagram')
plt.xlabel('Followers in Million')
plt.show()
```



In [31]:

```
plt.figure(figsize=(15,6))
sns.countplot(x = 'Category', data=insta_data)
plt.xticks(rotation = 90)
plt.title('Total Number of Categories')
plt.xlabel('Catogories')
plt.show()
```



In [32]:

```
insta_data['Audience Country'].unique()
```

Out[32]:

```
array(['Spain', 'Indonesia', 'Russia', 'Brazil', 'Poland', 'South Korea',
       'United States', 'Thailand', 'India', 'Iraq', 'Morocco', 'Turkey',
       nan, 'Mexico', 'Chile', 'Iran', 'Italy', 'Colombia', 'Argentina',
       'Philippines', 'United Kingdom', 'Germany', 'Nigeria', 'Serbia',
       'Albania', 'United Arab Emirates', 'China', 'France', 'Japan',
       'Egypt', 'Syria', 'Algeria', 'Ukraine'], dtype=object)
```

In [33]:

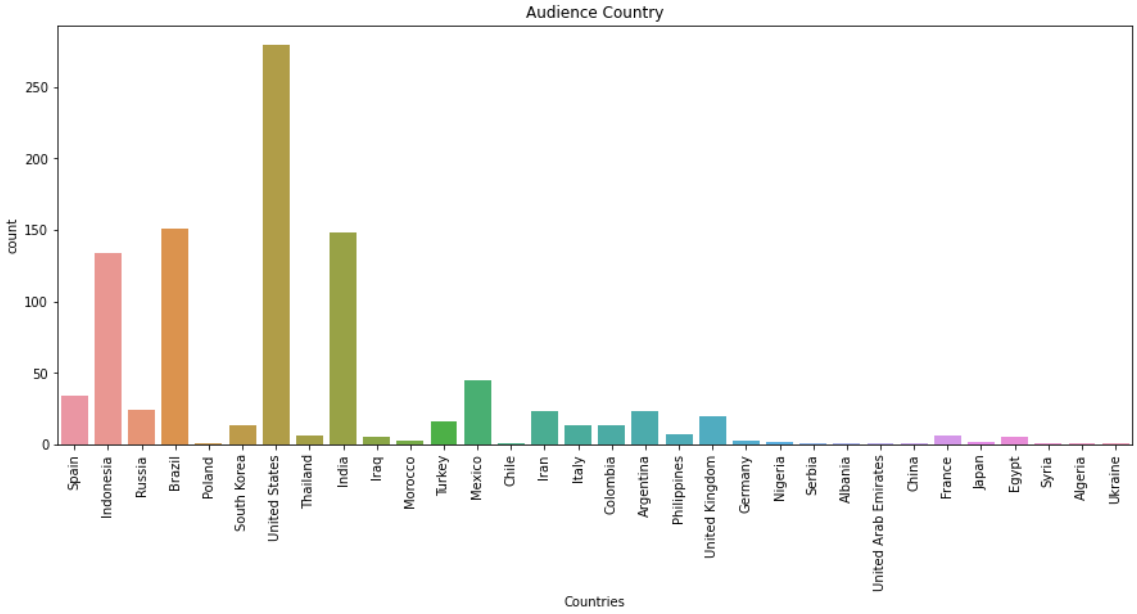
```
insta_data['Audience Country'].value_counts()
```

Out[33]:

```
United States           279
Brazil                  151
India                   148
Indonesia               134
Mexico                   45
Spain                    34
Russia                   24
Argentina                23
Iran                     23
United Kingdom           20
Turkey                   16
Italy                    13
Colombia                 13
South Korea              13
Philippines               7
France                    6
Thailand                  6
Iraq                      5
Egypt                     5
Germany                   3
Morocco                   3
Nigeria                   2
Japan                     2
Chile                     1
Algeria                   1
Syria                     1
Serbia                    1
China                     1
United Arab Emirates      1
Albania                   1
Poland                    1
Ukraine                   1
Name: Audience Country, dtype: int64
```

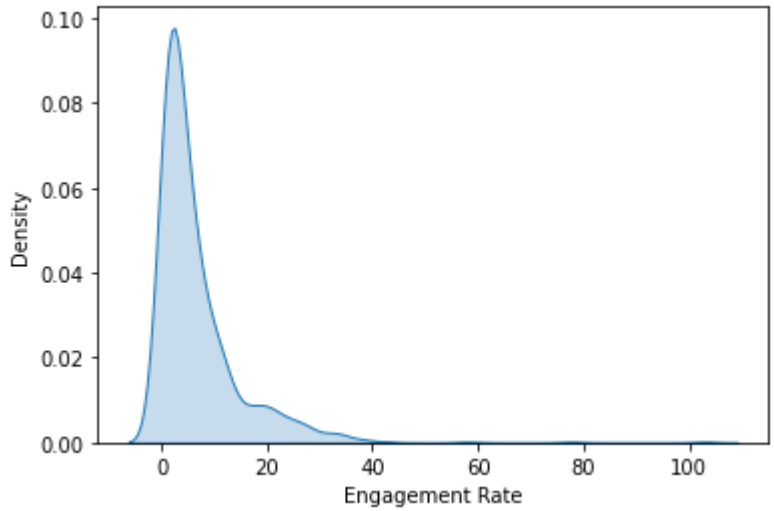
In [34]:

```
plt.figure(figsize=(15,6))
sns.countplot(x = 'Audience Country', data=insta_data)
plt.xticks(rotation = 90)
plt.title('Audience Country')
plt.xlabel('Countries')
plt.show()
```



In [35]:

```
sns.kdeplot(x='Engagement Rate', data=insta_data, shade=True);
```



In [36]:

```
insta_data['newFollowers'].info()
```

```
<class 'pandas.core.series.Series'>
Int64Index: 997 entries, 0 to 999
Series name: newFollowers
Non-Null Count  Dtype
--------------  -----
997 non-null    float64
dtypes: float64(1)
memory usage: 15.6 KB
```

In [37]:

```
insta_data['newFollowers'].describe()
```

Out[37]:

```
count    997.000000
mean      25.539619
std       40.586338
min        2.600000
25%        9.000000
50%       14.600000
75%       26.500000
max      487.200000
Name: newFollowers, dtype: float64
```

In [38]:

```
insta_data['newFollowers'].quantile(0.9)
```

Out[38]:

```
47.7
```

In [39]:

```
def for_mini_followers_instagram(coun,cat):
    data1=insta_data[insta_data['Audience Country']==coun]
    data_mini=data1[data1['newFollowers']<60]
    return data_mini.sort_values(by='Engagement Rate',ascending=False).groupby('Category
```

In [40]:

```
for_mini_followers_instagram('United States','Fashion')
```

Out[40]:

	instagram name	Audience Country	Engagement Rate
558	loren gray	United States	1.01

In [41]:

```
for_mini_followers_instagram('India','Music')
```

Out[41]:

	instagram name	Audience Country	Engagement Rate
705	Olivia Rodrigo	India	15.284
992	Zayn Malik	India	12.918
242	djsnake	India	4.721
118	Bebe Rexha	India	3.028
876	Tamannaah Bhatia	India	2.528
216	Darshan Raval #Goriye	India	1.683
604	marshmello	India	1.392
581	Madhuri Dixit	India	1.121
824	shreyaghoshal	India	0.601
333	Guru Randhawa	India	0.430

In [42]:

```
for_mini_followers_instagram('Russia','Lifestyle')
```

Out[42]:

	instagram name	Audience Country	Engagement Rate
534	Валерия Чекалина	Russia	8.793

In [43]:

```
def for_mega_followers_instagram(coun,cat):
    data1=insta_data[insta_data['Audience Country']==coun]
    data_mega=data1[data1['newFollowers']>60]
    return data_mega.sort_values(by='Engagement Rate',ascending=False).groupby('Category
```

In [44]:

```
for_mega_followers_instagram('United States','Fashion')
```

Out[44]:

	instagram name	Audience Country	Engagement Rate
510	Kylie ♡	United States	3.805
494	Kim Kardashian	United States	0.978

In [45]:

```
for_mega_followers_instagram('India','Music')
```

Out[45]:

	instagram name	Audience Country	Engagement Rate
818	Shakira	India	1.317
670	Neha Kakkar (Mrs. Singh)	India	1.131
458	Justin Bieber	India	0.281