In [1]:
```
#importing the libaries

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

import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
#importing the data:

data = pd.read_excel("C:/Users/Dell/Documents/Protfolio case studies/Global YouTube Statistics 2023/Global YouTube Statistics.xlsx")
```

In [3]:
```
data.head(5)
```

Out[3]:

	rank	Youtuber	subscribers	video views	category	Title	uploads	Country	Abbreviation	channel_type	...	subscribers_for_last_30_days	created_year	created_month	created_date	Gross tertiary education enrollment (%)	Population	Unemployment rate	Urban_population	Lati
0	1	T-Series	245000000	228000000000	Music	T-Series	20082	India	IN	Music	...	2000000.0	2006.0	Mar	13.0	28.1	1.366418e+09	5.36	471031528.0	20.59
1	2	YouTube Movies	170000000	0	Film & Animation	youtubemovies	1	United States	US	Games	...	NaN	2006.0	Mar	5.0	88.2	3.282395e+08	14.70	270663028.0	37.09
2	3	MrBeast	166000000	28368841870	Entertainment	MrBeast	741	United States	US	Entertainment	...	8000000.0	2012.0	Feb	20.0	88.2	3.282395e+08	14.70	270663028.0	37.09
3	4	Cocomelon - Nursery Rhymes	162000000	164000000000	Education	Cocomelon - Nursery Rhymes	966	United States	US	Education	...	1000000.0	2006.0	Sep	1.0	88.2	3.282395e+08	14.70	270663028.0	37.09
4	5	SET India	159000000	148000000000	Shows	SET India	116536	India	IN	Entertainment	...	1000000.0	2006.0	Sep	20.0	28.1	1.366418e+09	5.36	471031528.0	20.59

5 rows × 28 columns

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

Out[4]:

	rank	Youtuber	subscribers	video views	category	Title	uploads	Country	Abbreviation	channel_type	...	subscribers_for_last_30_days	created_year	created_month	created_date	Gross tertiary education enrollment (%)	Population	Unemployment rate	Urban_population	Lati
990	991	Natan por Aï¿	12300000	9029609749	Sports	Natan por Aï¿	1200	Brazil	BR	Entertainment	...	700000.0	2017.0	Feb	12.0	51.3	2.125594e+08	12.08	183241641.0	-14.23
991	992	Free Fire India Official	12300000	1674409945	People & Blogs	Free Fire India Official	1500	India	IN	Games	...	300000.0	2018.0	Sep	14.0	28.1	1.366418e+09	5.36	471031528.0	20.59
992	993	Panda	12300000	2214684303	NaN	HybridPanda	2452	United Kingdom	GB	Games	...	1000.0	2006.0	Sep	11.0	60.0	6.683440e+07	3.85	55908316.0	55.37
993	994	RobTopGames	12300000	374123483	Gaming	RobTopGames	39	Sweden	SE	Games	...	100000.0	2012.0	May	9.0	67.0	1.028545e+07	6.48	9021165.0	60.12
994	995	Make Joke Of	12300000	2129773714	Comedy	Make Joke Of	62	India	IN	Comedy	...	100000.0	2017.0	Aug	1.0	28.1	1.366418e+09	5.36	471031528.0	20.59

5 rows × 28 columns

In [5]:
```
data.shape
```

Out[5]:
```
(995, 28)
```

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 995 entries, 0 to 994
Data columns (total 28 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   rank                                    995 non-null    int64
 1   Youtuber                                995 non-null    object
 2   subscribers                             995 non-null    int64
 3   video views                             995 non-null    int64
 4   category                                949 non-null    object
 5   Title                                   995 non-null    object
 6   uploads                                 995 non-null    int64
 7   Country                                 873 non-null    object
 8   Abbreviation                            873 non-null    object
 9   channel_type                            965 non-null    object
 10  video_views_rank                        994 non-null    float64
 11  country_rank                            879 non-null    float64
 12  channel_type_rank                       962 non-null    float64
 13  video_views_for_the_last_30_days        939 non-null    float64
 14  lowest_monthly_earnings                 995 non-null    float64
 15  highest_monthly_earnings                995 non-null    float64
 16  lowest_yearly_earnings                  995 non-null    float64
 17  highest_yearly_earnings                 995 non-null    float64
 18  subscribers_for_last_30_days            658 non-null    float64
 19  created_year                            990 non-null    float64
 20  created_month                           990 non-null    object
 21  created_date                            990 non-null    float64
 22  Gross tertiary education enrollment (%) 872 non-null    float64
 23  Population                              872 non-null    float64
 24  Unemployment rate                       872 non-null    float64
 25  Urban_population                        872 non-null    float64
 26  Latitude                                872 non-null    float64
 27  Longitude                               872 non-null    float64
dtypes: float64(17), int64(4), object(7)
memory usage: 217.8+ KB
```

In [7]:
```
data.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
rank	995.0	4.980000e+02	2.873761e+02	1.000000e+00	2.495000e+02	4.980000e+02	7.465000e+02	9.950000e+02
subscribers	995.0	2.298241e+07	1.752611e+07	1.230000e+07	1.450000e+07	1.770000e+07	2.460000e+07	2.450000e+08
video views	995.0	1.103954e+10	1.411084e+10	0.000000e+00	4.288145e+09	7.760820e+09	1.355470e+10	2.280000e+11
uploads	995.0	9.187126e+03	3.415135e+04	0.000000e+00	1.945000e+02	7.290000e+02	2.667500e+03	3.013080e+05
video_views_rank	994.0	5.542489e+05	1.362782e+06	1.000000e+00	3.230000e+02	9.155000e+02	3.584500e+03	4.057944e+06
country_rank	879.0	3.860535e+02	1.232245e+03	1.000000e+00	1.100000e+01	5.100000e+01	1.230000e+02	7.741000e+03
channel_type_rank	962.0	7.457193e+02	1.944387e+03	1.000000e+00	2.700000e+01	6.550000e+01	1.397500e+02	7.741000e+03
video_views_for_the_last_30_days	939.0	1.756103e+08	4.163782e+08	1.000000e+00	2.013750e+07	6.408500e+07	1.688265e+08	6.589000e+09
lowest_monthly_earnings	995.0	3.688615e+04	7.185872e+04	0.000000e+00	2.700000e+03	1.330000e+04	3.790000e+04	8.509000e+05
highest_monthly_earnings	995.0	5.898078e+05	1.148622e+06	0.000000e+00	4.350000e+04	2.127000e+05	6.068000e+05	1.360000e+07
lowest_yearly_earnings	995.0	4.422574e+05	8.612161e+05	0.000000e+00	3.265000e+04	1.595000e+05	4.551000e+05	1.020000e+07
highest_yearly_earnings	995.0	7.081814e+06	1.379704e+07	0.000000e+00	5.217500e+05	2.600000e+06	7.300000e+06	1.634000e+08
subscribers_for_last_30_days	658.0	3.490791e+05	6.143554e+05	1.000000e+00	1.000000e+05	2.000000e+05	4.000000e+05	8.000000e+06
created_year	990.0	2.012630e+03	4.512503e+00	1.970000e+03	2.009000e+03	2.013000e+03	2.016000e+03	2.022000e+03
created_date	990.0	1.574646e+01	8.777520e+00	1.000000e+00	8.000000e+00	1.600000e+01	2.300000e+01	3.100000e+01
Gross tertiary education enrollment (%)	872.0	6.362775e+01	2.610689e+01	7.600000e+00	3.630000e+01	6.800000e+01	8.820000e+01	1.131000e+02
Population	872.0	4.303873e+08	4.727947e+08	2.025060e+05	8.335541e+07	3.282395e+08	3.282395e+08	1.397715e+09
Unemployment rate	872.0	9.279278e+00	4.888354e+00	7.500000e-01	5.270000e+00	9.365000e+00	1.470000e+01	1.472000e+01
Urban_population	872.0	2.242150e+08	1.546874e+08	3.558800e+04	5.590832e+07	2.706630e+08	2.706630e+08	8.429340e+08
Latitude	872.0	2.663278e+01	2.056053e+01	-3.841610e+01	2.059368e+01	3.709024e+01	3.709024e+01	6.192411e+01
Longitude	872.0	-1.412815e+01	8.476081e+01	-1.721046e+02	-9.571289e+01	-5.192528e+01	7.896288e+01	1.382529e+02

In [8]:
```
data.columns
```

Out[8]:
```
Index(['rank', 'Youtuber', 'subscribers', 'video views', 'category', 'Title',
       'uploads', 'Country', 'Abbreviation', 'channel_type',
       'video_views_rank', 'country_rank', 'channel_type_rank',
       'video_views_for_the_last_30_days', 'lowest_monthly_earnings',
       'highest_monthly_earnings', 'lowest_yearly_earnings',
       'highest_yearly_earnings', 'subscribers_for_last_30_days',
       'created_year', 'created_month', 'created_date',
       'Gross tertiary education enrollment (%)', 'Population',
       'Unemployment rate', 'Urban_population', 'Latitude', 'Longitude'],
      dtype='object')
```

In [9]:
```
data.duplicated().sum()
```

Out[9]:
```
0
```

Data Cleaning:

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

Out[10]:
```
rank                                  0
Youtuber                              0
subscribers                           0
video views                           0
category                             46
Title                                 0
uploads                               0
Country                             122
Abbreviation                        122
channel_type                         30
video_views_rank                      1
country_rank                        116
channel_type_rank                    33
video_views_for_the_last_30_days     56
lowest_monthly_earnings               0
highest_monthly_earnings              0
lowest_yearly_earnings                0
highest_yearly_earnings               0
subscribers_for_last_30_days        337
created_year                          5
created_month                         5
created_date                          5
Gross tertiary education enrollment (%)    123
Population                          123
Unemployment rate                   123
Urban_population                    123
Latitude                            123
Longitude                           123
dtype: int64
```

In [11]:
```
data.nunique()
```

Out[11]:
```
rank                                  995
Youtuber                              995
subscribers                           289
video views                           988
category                               18
Title                                 992
uploads                               777
Country                                49
Abbreviation                           49
channel_type                           14
video_views_rank                      953
country_rank                          246
channel_type_rank                     286
video_views_for_the_last_30_days      908
lowest_monthly_earnings               557
highest_monthly_earnings              736
lowest_yearly_earnings                757
highest_yearly_earnings               419
subscribers_for_last_30_days           53
created_year                           19
created_month                          12
created_date                           31
Gross tertiary education enrollment (%)    47
Population                             48
Unemployment rate                      47
Urban_population                       48
Latitude                               48
Longitude                              48
dtype: int64
```

In [12]:
```
# subscribers_for_last_30_days has many  null values, hence we are using the mean formula to fill in those null values.

data['subscribers_for_last_30_days'].fillna(data['subscribers_for_last_30_days'].mean(),inplace=True)
```

In [13]:
```
data = data.dropna()
```

In [14]:
```
#checking for null values,

data.isnull().sum()
```

Out[14]:
```
rank                                  0
Youtuber                              0
subscribers                           0
video views                           0
category                              0
Title                                 0
uploads                               0
Country                               0
Abbreviation                          0
channel_type                          0
video_views_rank                      0
country_rank                          0
channel_type_rank                     0
video_views_for_the_last_30_days      0
lowest_monthly_earnings               0
highest_monthly_earnings              0
lowest_yearly_earnings                0
highest_yearly_earnings               0
subscribers_for_last_30_days          0
created_year                          0
created_month                         0
created_date                          0
Gross tertiary education enrollment (%)    0
Population                            0
Unemployment rate                     0
Urban_population                      0
Latitude                              0
Longitude                             0
dtype: int64
```

In [15]:
```
data.dtypes
```

Out[15]:
```
rank                                    int64
Youtuber                               object
subscribers                             int64
video views                             int64
category                               object
Title                                  object
uploads                                 int64
Country                                object
Abbreviation                           object
channel_type                           object
video_views_rank                      float64
country_rank                          float64
channel_type_rank                     float64
video_views_for_the_last_30_days      float64
lowest_monthly_earnings               float64
highest_monthly_earnings              float64
lowest_yearly_earnings                float64
highest_yearly_earnings               float64
subscribers_for_last_30_days          float64
created_year                          float64
created_month                          object
created_date                          float64
Gross tertiary education enrollment (%)    float64
Population                            float64
Unemployment rate                     float64
Urban_population                      float64
Latitude                              float64
Longitude                             float64
dtype: object
```

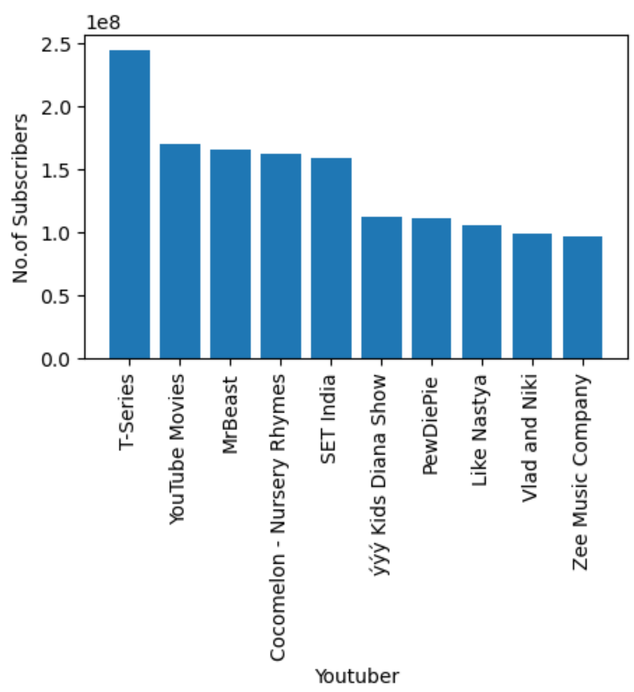
In [ ]:
```

```

EDA:

1. Find the YouTuber with the highest number of subscribers.

In [16]:
```
sorted_data = data.sort_values(by='subscribers', ascending=False).iloc[:10]
```

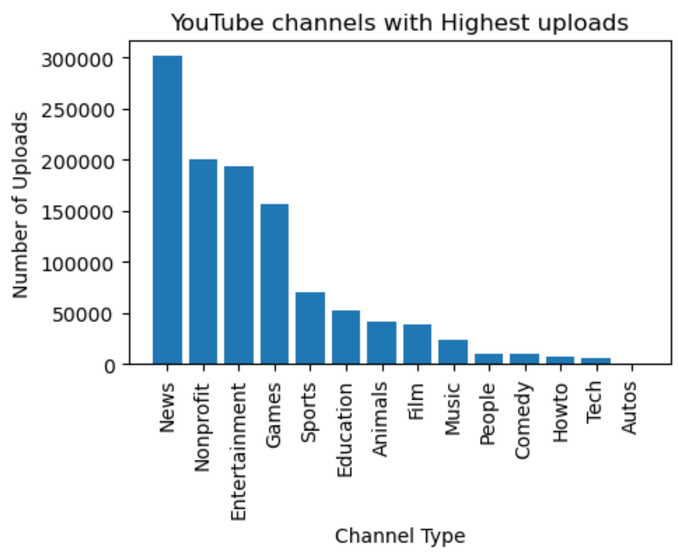
In [17]:
```
plt.figure(figsize=(5,3))

plt.bar(sorted_data['Youtuber'],sorted_data['subscribers'])
plt.xlabel('Youtuber')
plt.ylabel('No.of Subscribers')
plt.xticks(rotation=90)
plt.show()
```



- YouTuber with the highest no.of Subscribers:- is T-Series with Subscribers: 245000000

2. Find the channel with the highest uploads in YouTube.

In [18]:
```
highest_uploads = data.sort_values(by='uploads',ascending=False)

plt.figure(figsize=(5,3))
plt.bar(highest_uploads['channel_type'],highest_uploads['uploads'])
plt.xlabel('Channel Type')
plt.ylabel('Number of Uploads')
plt.title('YouTube channels with Highest uploads')
plt.xticks(rotation=90)
plt.show()
```

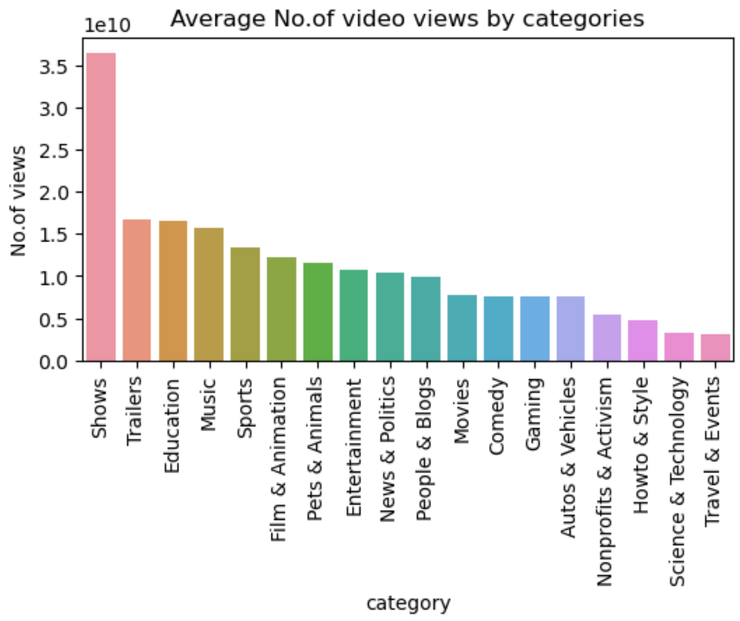


- News channels has highest uploads with count of 300000

3. How does the average number of video views vary across various categories in the dataset?

In [19]:
```
average_views = data.groupby('category')['video views'].mean().reset_index()

category_views = average_views.sort_values(by = 'video views', ascending=False)
```

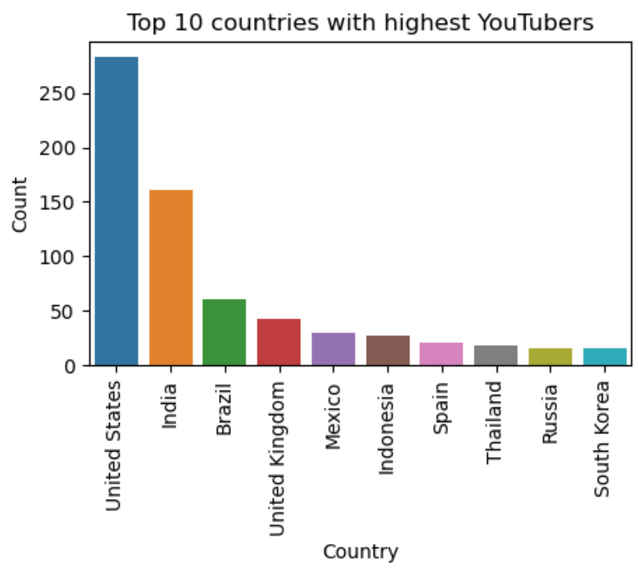
In [20]:
```
plt.figure(figsize=(6,3))
sns.barplot(x='category', y='video views',data=category_views, order=category_views['category'])
plt.title('Average No.of video views by categories')
plt.ylabel('No.of views')
plt.xlabel('category')

plt.xticks(rotation=90)
plt.show()
```



4. Top 10 countries that has the highest number of YouTubers?

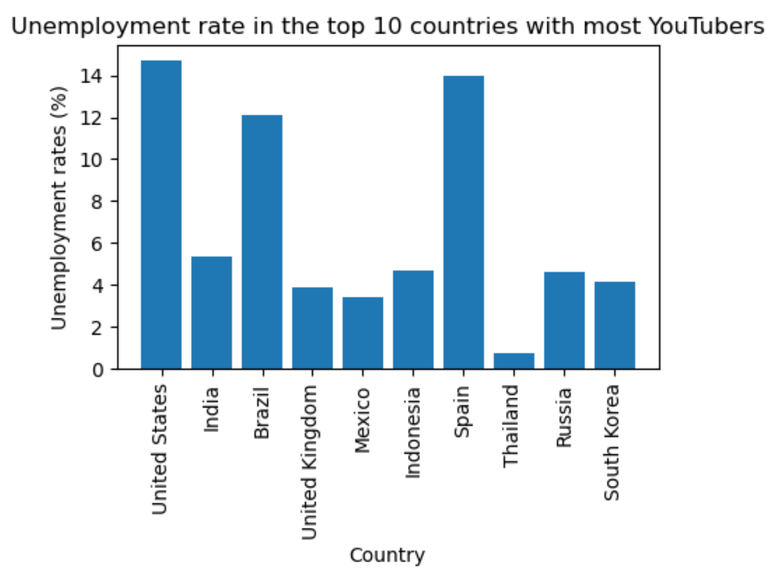
In [21]:
```
top_countries = data['Country'].value_counts().iloc[:10]

plt.figure(figsize=(5,3))
sns.barplot(x=top_countries.index, y=top_countries.values)
plt.title("Top 10 countries with highest YouTubers")
plt.xticks(rotation=90)
plt.xlabel("Country")
plt.ylabel("Count")
```

Out[21]:
```
Text(0, 0.5, 'Count')
```



5. What is the unemployment rate in the country with the highest number of YouTubers?

In [22]:
```
top_10_countries = data['Country'].value_counts().head(10).index

unemployment_rates = [
    data[data['Country']==country]['Unemployment rate'].iloc[0]
    for country in top_10_countries
]

plt.figure(figsize=(5,3))
plt.bar(top_10_countries,unemployment_rates)
plt.title('Unemployment rate in the top 10 countries with most YouTubers')
plt.xlabel('Country')
plt.ylabel('Unemployment rates (%)')
plt.xticks(rotation=90)
plt.show()
```



6. Which YouTuber in the dataset has the lowest and highest monthly earnings?

In [23]:
```
#YouTuber in the dataset with Lowest monthly earnings:

low_monthly_earning =  data.nsmallest(1,'lowest_monthly_earnings')

print('The YouTuber with Lowest monthly earnings is:- ')
print(low_monthly_earning[['Youtuber','lowest_monthly_earnings']])
```

```
The YouTuber with Lowest monthly earnings is:-
         Youtuber  lowest_monthly_earnings
1  YouTube Movies                     0.0
```

In [24]:
```
#YouTuber in the dataset with Highest monthly earnings:

high_monthly_earning = data.nlargest(1, 'highest_monthly_earnings')

print('The YouTuber with Highest monthly earnings is:- ')
print(high_monthly_earning[['Youtuber','highest_monthly_earnings']])
```

```
The YouTuber with Highest monthly earnings is:-
         Youtuber  highest_monthly_earnings
417  DaFuq!?Boom!                9200000.0
```

7. Which YouTuber in the dataset has the lowest and highest Yearly earnings?

In [25]:
```
#YouTuber in the dataset with Lowest Yearly earnings:

low_yearly_earning =  data.nsmallest(1,'lowest_yearly_earnings')

print('The YouTuber with Lowest Yearly earnings is:- ')
print(low_yearly_earning[['Youtuber','lowest_yearly_earnings']])
```

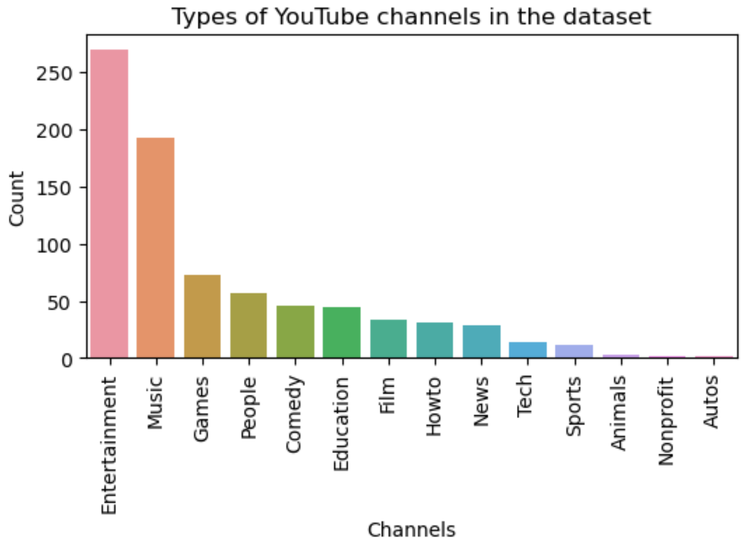
```
The YouTuber with Lowest Yearly earnings is:-
          Youtuber  lowest_yearly_earnings
16  5-Minute Crafts                    0.0
```

In [26]:
```
#YouTuber in the dataset with Highest monthly earnings:

high_monthly_earning = data.nlargest(1, 'highest_yearly_earnings')

print('The YouTuber with Highest monthly earnings is:- ')
print(high_monthly_earning[['Youtuber','highest_yearly_earnings']])
```

```
The YouTuber with Highest monthly earnings is:-
         Youtuber  highest_yearly_earnings
417  DaFuq!?Boom!              110600000.0
```

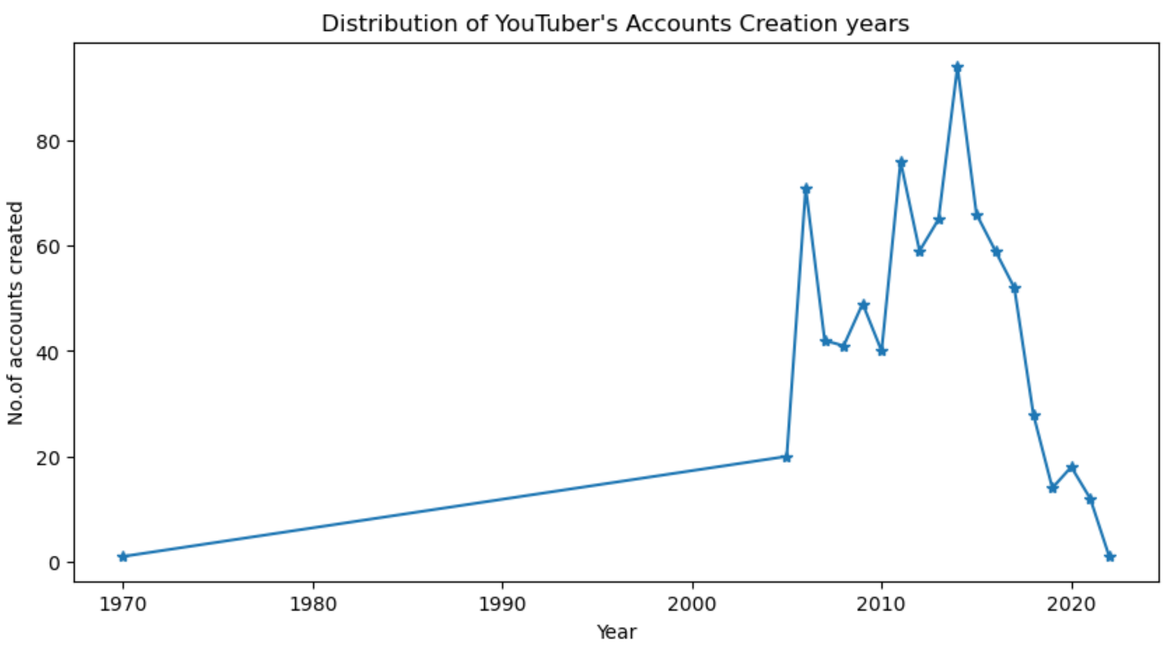
8. What types of YouTube channels are there in the dataset, and how many YouTubers are in each type?

In [27]:
```
data['channel_type'].value_counts()
plt.figure(figsize=(6,3))

sns.countplot(x='channel_type', data=data, order=data['channel_type'].value_counts().index)
plt.title("Types of YouTube channels in the dataset")
plt.xlabel("Channels")
plt.ylabel("Count")
plt.xticks(rotation=90)
plt.show()
```



9. In which year were most YouTubers created in the dataset?

In [28]:
```
counts = data['created_year'].value_counts().sort_index()

plt.figure(figsize=(10,5))
plt.plot(counts.index,counts.values, marker="*")
plt.title("Distribution of YouTuber's Accounts Creation years")
plt.xlabel("Year")
plt.ylabel("No.of accounts created")
plt.show()
```



- The Maximum accounts are created by the YouTubers in the **year 2014**.

10. What is the distribution of YouTuber creation months?

In [29]:
```
creation_months = data['created_month'].value_counts().sort_index()

plt.figure(figsize=(10,5))
plt.plot(creation_months.index, creation_months.values, marker="*")
plt.title("Distribution of YouTuber creations throughout months")
plt.xlabel("Months")
plt.ylabel("No.of accounts created")
plt.show()
```


- The Maximum accounts are created in the Month of **January**.