In [1]:

```
%matplotlib inline
import pandas as pd
```

In [2]:

```
cast = pd.read_csv("C:\\Users\\sujoydutta\\Downloads\\cast.csv")
cast.head()
```

Out[2]:

	title	year	name	type	character	n
0	Suuri illusioni	1985	Homo $	actor	Guests	22.0
1	Gangsta Rap: The Glockumentary	2007	Too $hort	actor	Himself	NaN
2	Menace II Society	1993	Too $hort	actor	Lew-Loc	27.0
3	Porndogs: The Adventures of Sadie	2009	Too $hort	actor	Bosco	3.0
4	Stop Pepper Palmer	2014	Too $hort	actor	Himself	NaN

Define a year as a "Superman year" whose films feature more Superman characters than Batman. How many years in film history have been Superman years?

In [3]:

```
supermanfilm = cast[cast['character'].str.contains('Superman', na=False, case=False)]
batmanfilm = cast[cast['character'].str.contains('Batman', na=False, case=False)]

supermanfilm_count = supermanfilm.groupby('year')['title'].nunique().reset_index(name='s
batmanfilm_count = batmanfilm.groupby('year')['title'].nunique().reset_index(name='batma


film_counts = pd.merge(supermanfilm_count, batmanfilm_count, on='year', how='outer').fil
superman_years = film_counts[film_counts['superman_count'] > film_counts['batman_count']
len(superman_years)
```

Out[3]:

```
13
```

How many years have been "Batman years", with more Batman characters than Superman characters?

In [4]:

```
batman_years = film_counts[film_counts['superman_count'] < film_counts['batman_count']]
len(batman_years)
```

Out[4]:

```
35
```

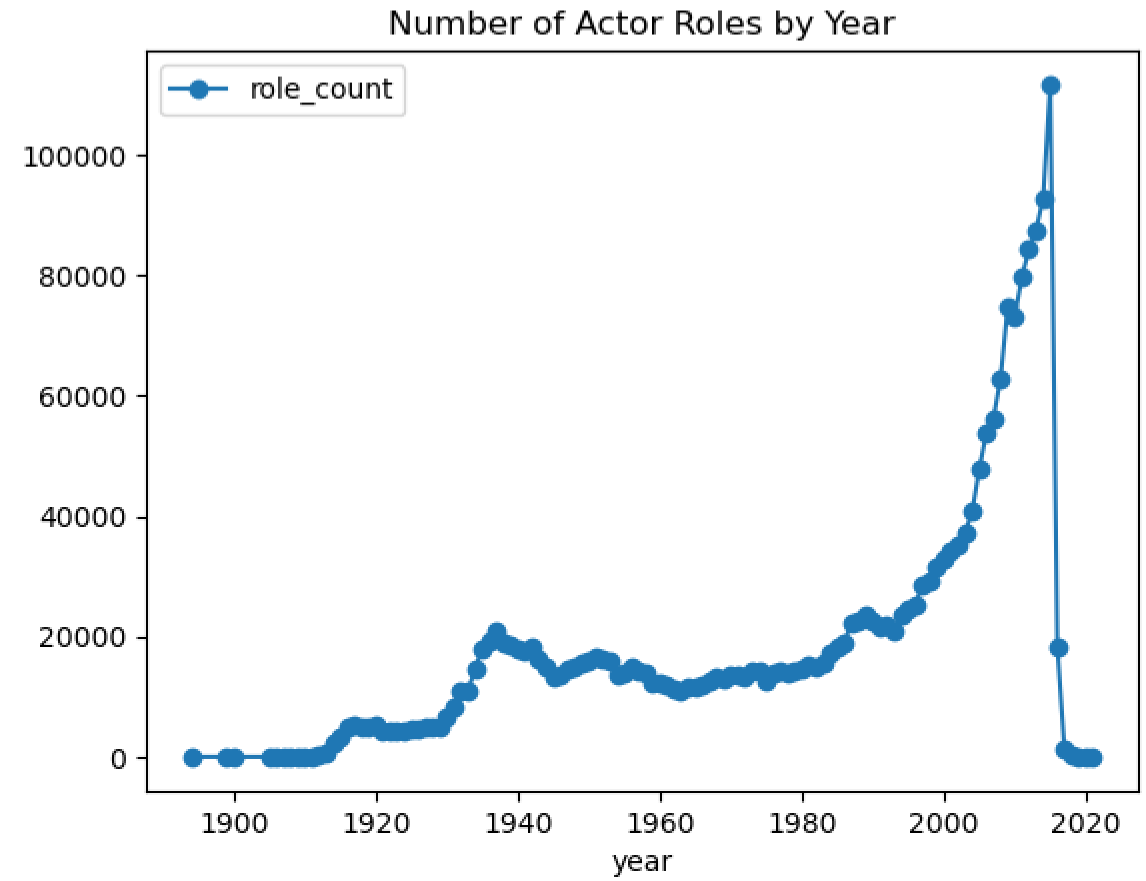
Plot the number of actor roles each year and the number of actress roles each year over the history of film.

In [5]:

```
actorrole = cast[cast['type'].str.contains('actor', na=False, case=False)]
actorrole_count = actorrole.groupby('year').size().reset_index(name='role_count')
actorrole_count.plot(x='year', y='role_count', kind='line', marker='o', title='Number of
```

Out[5]:

```
<Axes: title={'center': 'Number of Actor Roles by Year'}, xlabel='year'>
```

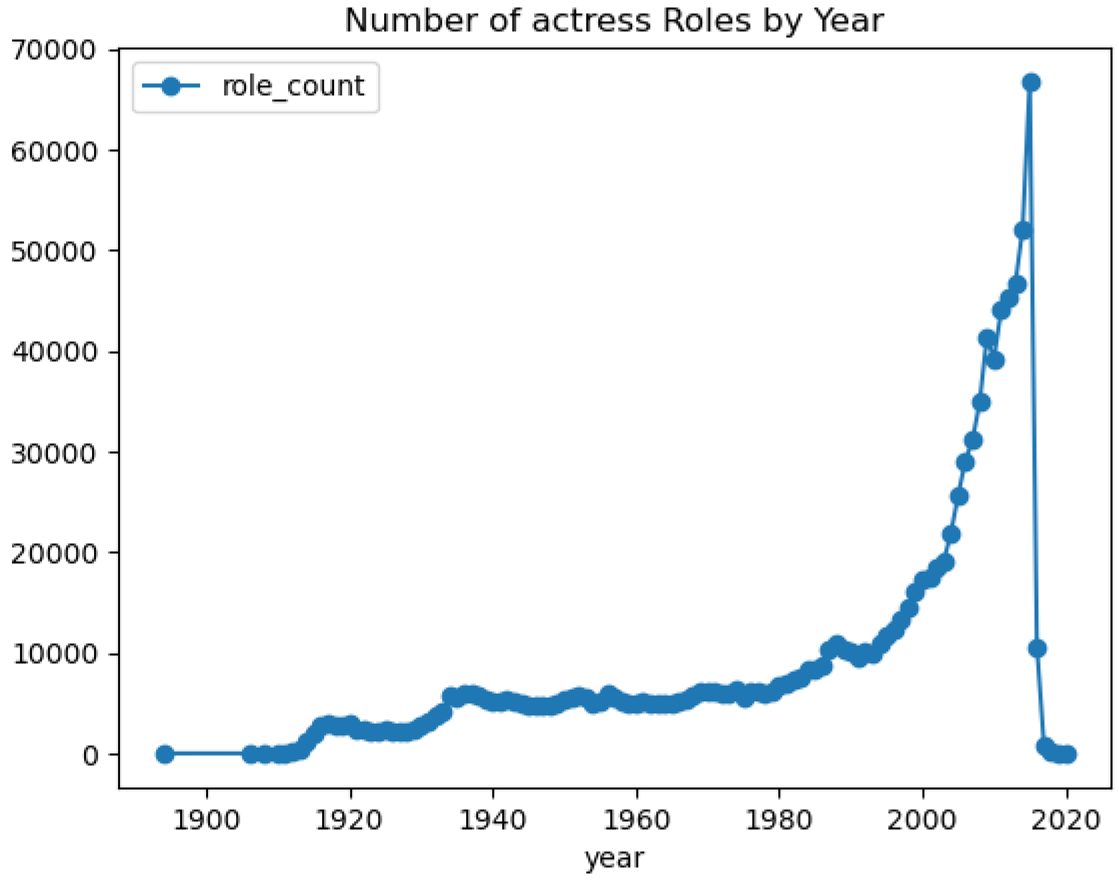


In [6]:

```
actressrole = cast[cast['type'].str.contains('actress', na=False, case=False)]
actressrole_count = actressrole.groupby('year').size().reset_index(name='role_count')
actressrole_count.plot(x='year', y='role_count', kind='line', marker='o', title='Number
```

Out[6]:

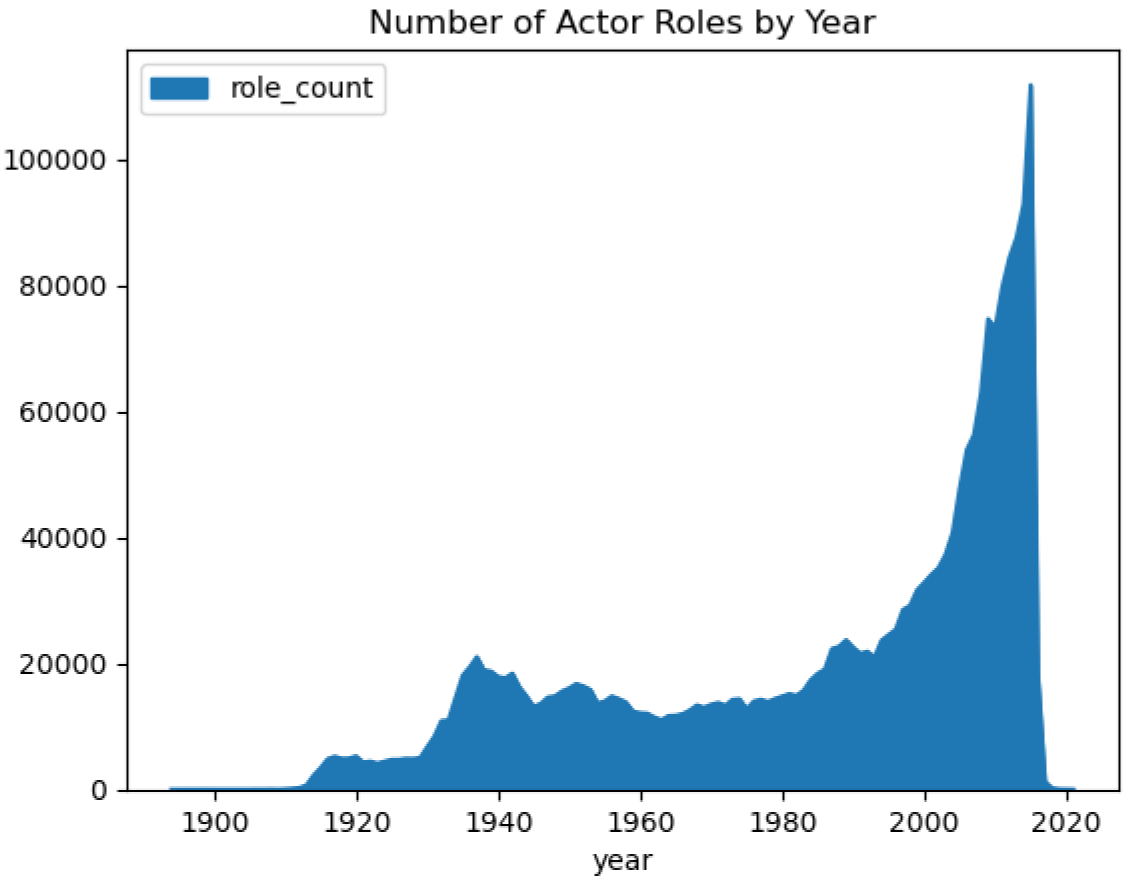
```
<Axes: title={'center': 'Number of actress Roles by Year'}, xlabel='year'>
```



Plot the number of actor roles each year and the number of actress roles each year, but this time as a kind='area' plot.

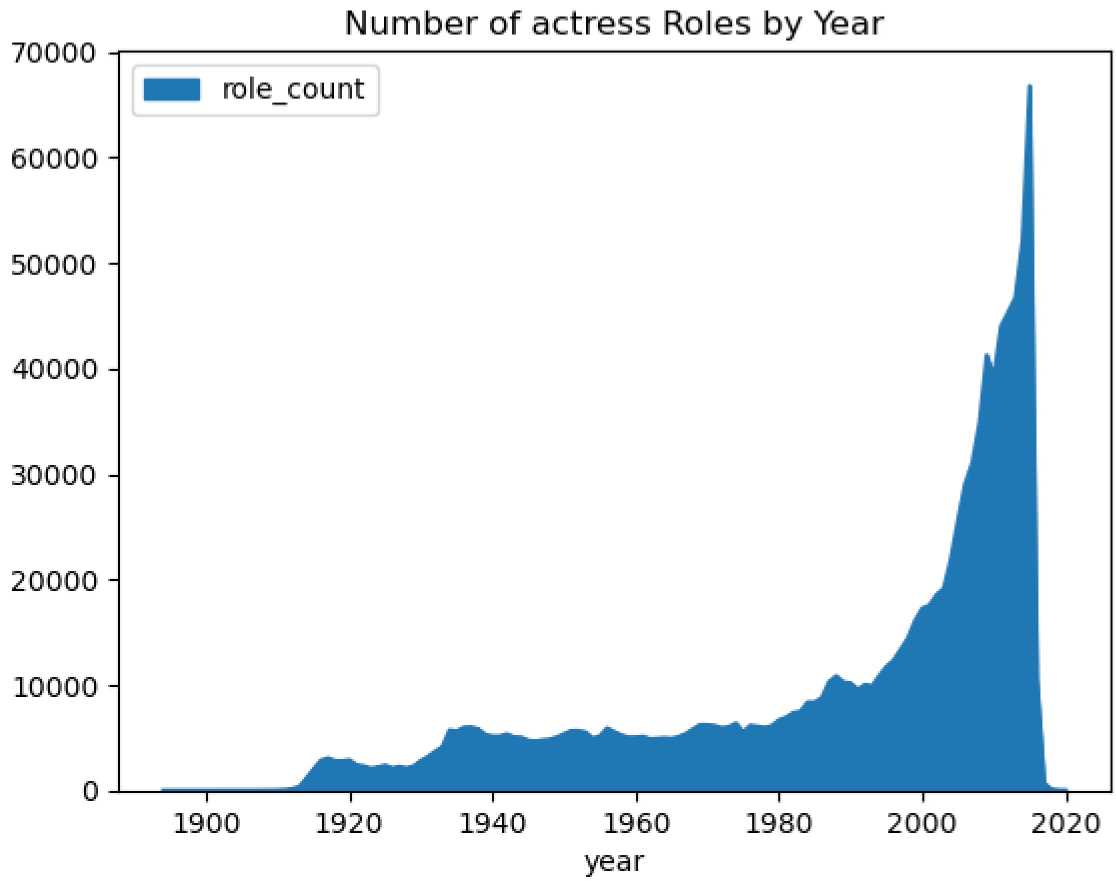
In [7]:
```
actorrole = cast[cast['type'].str.contains('actor', na=False, case=False)]
actorrole_count = actorrole.groupby('year').size().reset_index(name='role_count')
actorrole_count.plot(x='year', y='role_count', kind='area', title='Number of Actor Roles
```

Out[7]:
```
<Axes: title={'center': 'Number of Actor Roles by Year'}, xlabel='year'>
```



In [8]:
```
actressrole = cast[cast['type'].str.contains('actress', na=False, case=False)]
actressrole_count = actressrole.groupby('year').size().reset_index(name='role_count')
actressrole_count.plot(x='year', y='role_count', kind='area', title='Number of actress R
```

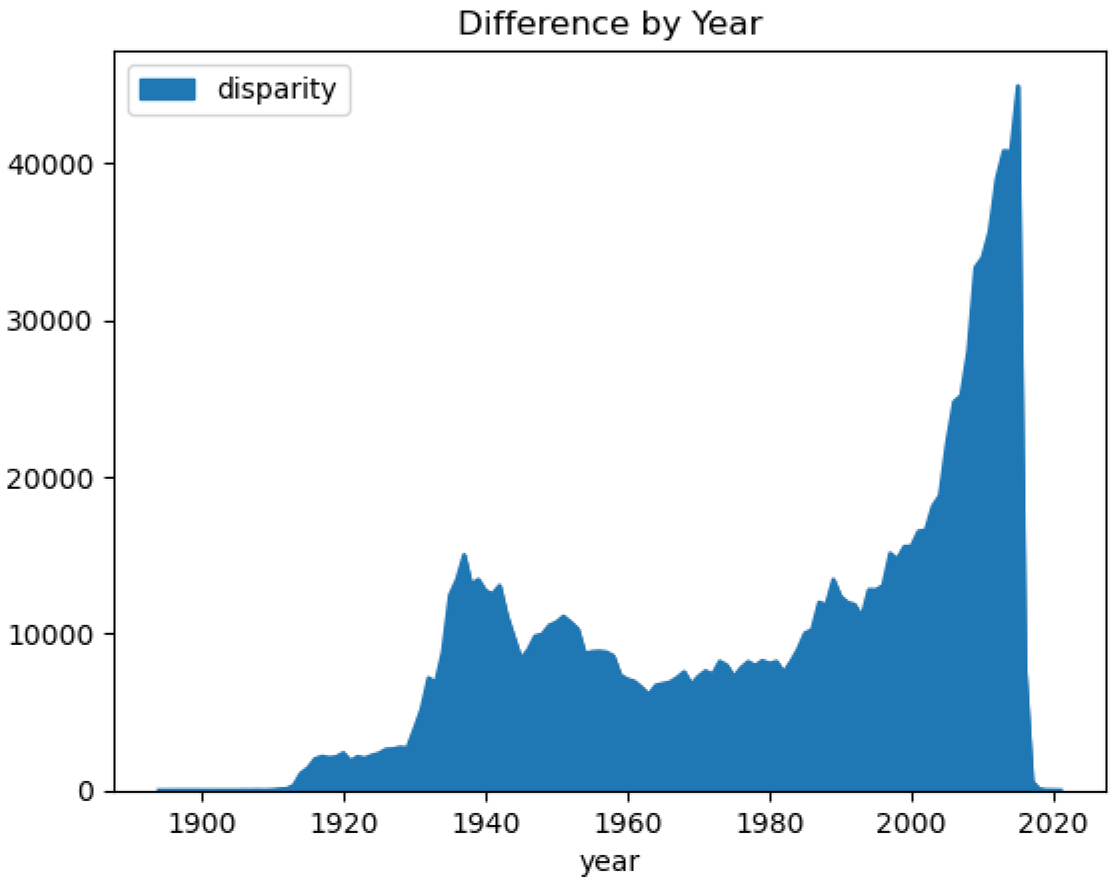
Out[8]:
```
<Axes: title={'center': 'Number of actress Roles by Year'}, xlabel='year'>
```



Plot the difference between the number of actor roles each year and the number of actress roles each year over the history of film.

In [9]:
```
role_difference = pd.merge(actorrole_count, actressrole_count, on='year', how='outer').f
role_difference['disparity']=role_difference.role_count_x-role_difference.role_count_y
role_difference.plot(x='year', y='disparity', kind='area', title='Difference by Year')
```

Out[9]:
```
<Axes: title={'center': 'Difference by Year'}, xlabel='year'>
```



Plot the fraction of roles that have been 'actor' roles each year in the history of film.

In [12]:

```
role_proportion = pd.merge(actorrole_count, actressrole_count, on='year', how='outer').f
role_proportion['totalroles']=role_difference.role_count_x+role_difference.role_count_y
```

In [14]:

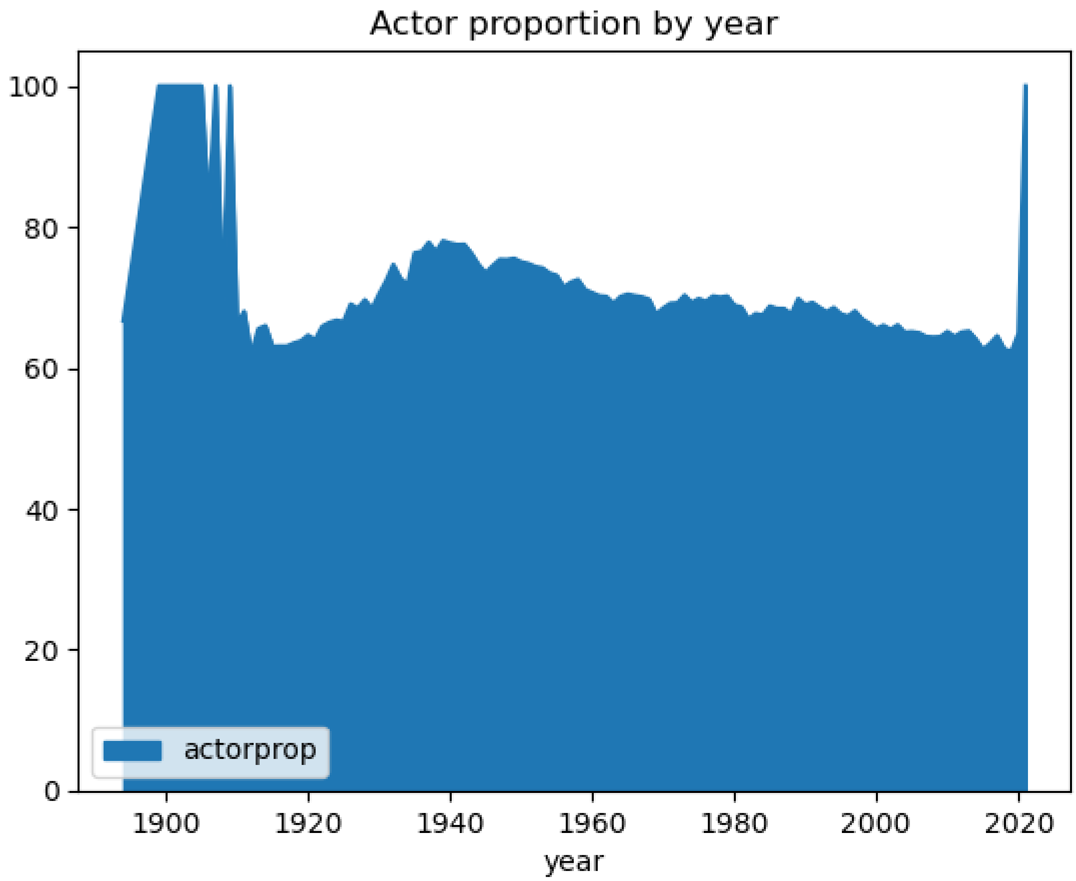
```
role_proportion['actorprop']=(role_proportion.role_count_x/role_proportion.totalroles)*1
role_proportion['actressprop']=100-role_proportion.actorprop
```

In [15]:

```
role_proportion.plot(x='year', y='actorprop', kind='area', title='Actor proportion by ye
```

Out[15]:

```
<Axes: title={'center': 'Actor proportion by year'}, xlabel='year'>
```



Plot the fraction of supporting (n=2) roles that have been 'actor' roles each year in the history of film.

In [20]:

```
supactorrole = cast[(cast['type'].str.contains('actor', na=False, case=False)) & (cast['
supactorrole_count = supactorrole.groupby('year').size().reset_index(name='role_count_M'
supactressrole = cast[(cast['type'].str.contains('actress', na=False, case=False)) & (ca
supactressrole_count = supactressrole.groupby('year').size().reset_index(name='role_coun
suproledf = pd.merge(supactorrole_count, supactressrole_count, on='year', how='outer').f
suproledf['totalroles']=suproledf.role_count_M+suproledf.role_count_F
suproledf['actorprop']=(suproledf.role_count_M/suproledf.totalroles)*100
suproledf['actressprop']=(suproledf.role_count_F/suproledf.totalroles)*100
suproledf.head()
```

Out[20]:

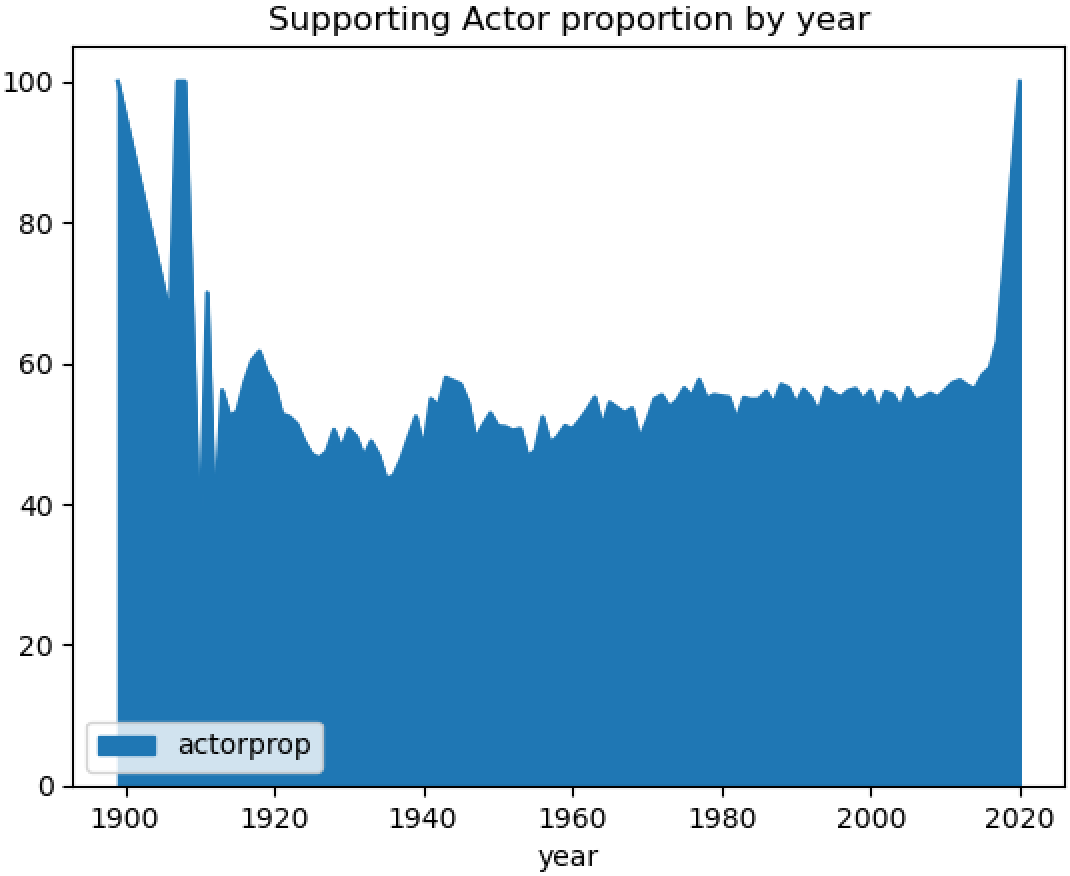
	year	role_count_M	role_count_F	totalroles	actorprop	actressprop
0	1899	2	0.0	2.0	100.000000	0.000000
1	1906	2	1.0	3.0	66.666667	33.333333
2	1907	1	0.0	1.0	100.000000	0.000000
3	1908	2	0.0	2.0	100.000000	0.000000
4	1910	1	2.0	3.0	33.333333	66.666667

In [22]:

```
suproledf.plot(x='year', y='actorprop', kind='area', title='Supporting Actor proportion
```

Out[22]:

```
<Axes: title={'center': 'Supporting Actor proportion by year'}, xlabel='year'>
```



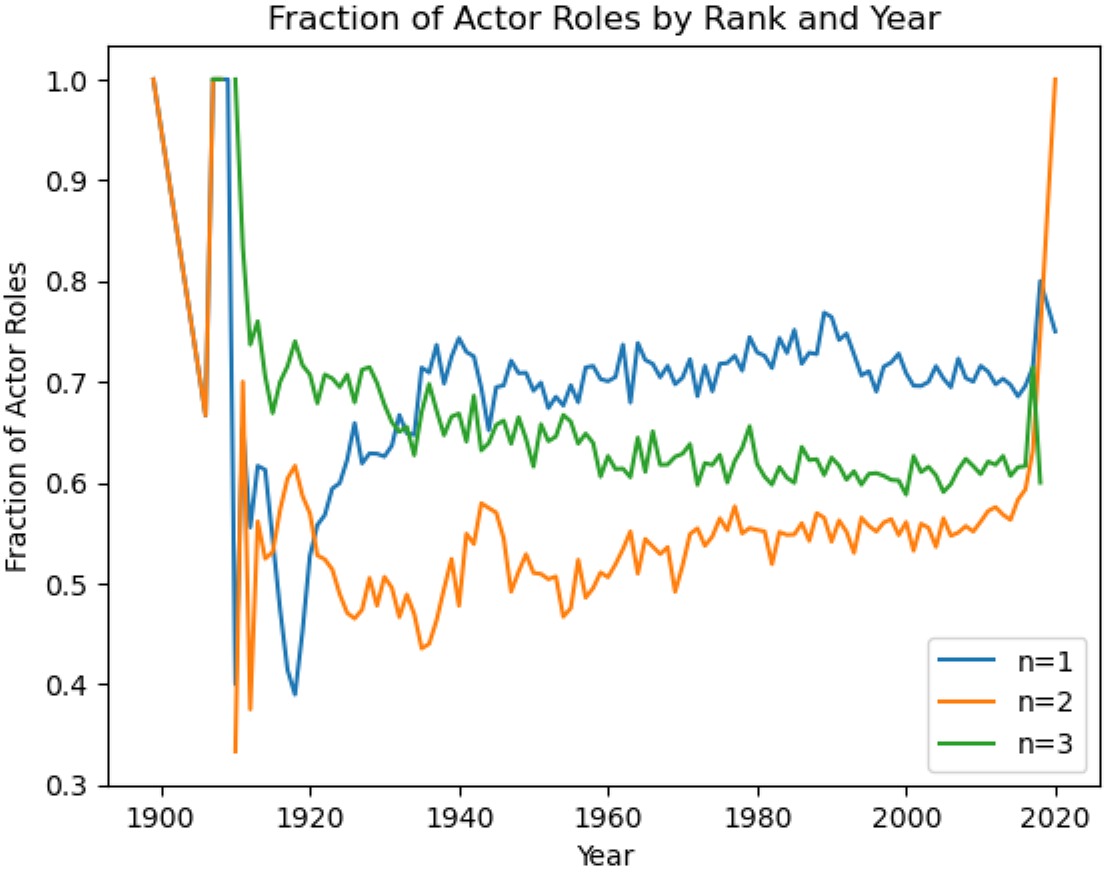
Build a plot with a line for each rank n=1 through n=3, where the line shows what fraction of that rank's roles were 'actor' roles for each year in the history of film.

In [24]:

```python
fractions = {}


for n in range(1, 4):

    roles = cast[cast['n'] == n]


    actor_roles = roles[roles['type'].str.contains('actor', na=False, case=False)]


    total_roles_by_year = roles.groupby('year').size()
    actor_roles_by_year = actor_roles.groupby('year').size()


    fraction = actor_roles_by_year / total_roles_by_year


    fractions[f'n={n}'] = fraction
```

In [26]:

```python
import matplotlib.pyplot as plt
fractions_df = pd.DataFrame(fractions)


fractions_df.plot(title='Fraction of Actor Roles by Rank and Year')
plt.xlabel('Year')
plt.ylabel('Fraction of Actor Roles')
plt.show()
```


Fraction of Actor Roles by Rank and Year
Fraction of Actor Roles
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
1900
1920
1940
1960
1980
2000
2020
Year
n=1
n=2
n=3