In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
movies = pd.read_csv(r"C:\Users\Jan Saida\Downloads\movie.csv.zip")
ratings = pd.read_csv(r"C:\Users\Jan Saida\Downloads\rating.csv.zip")
tags = pd.read_csv(r"C:\Users\Jan Saida\Downloads\tag.csv.zip")
```

In [3]:
```
print(movies.shape)
print(ratings.shape)
print(tags.shape)
```

```
(27278, 3)
(20000263, 4)
(465564, 4)
```

In [4]:
```
movies.head()
```

Out[4]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

In [5]:
```
del ratings['timestamp']
del tags['timestamp']
```

In [6]:
```
ratings.columns
```

Out[6]:
```
Index(['userId', 'movieId', 'rating'], dtype='object')
```

In [7]:
```
tags.columns
```

Out[7]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [8]:
```
row_0 = tags.iloc[0]
```

In [9]:
```
type(row_0)
```

Out[9]:
```
pandas.core.series.Series
```

In [10]:
```
print(row_0)
```

```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [11]:
```
row_4=ratings.iloc[4]
```

In [12]:
```
type(row_4)
```

Out[12]:
```
pandas.core.series.Series
```

In [13]:
```
print(row_4)
```

```
userId      1.0
movieId    50.0
rating      3.5
Name: 4, dtype: float64
```

In [14]:
```
row_0['userId']
```

Out[14]:
```
18
```

In [15]:
```
'rating' in row_0
```

Out[15]:
```
False
```

In [16]:
```
row_0.name
```

Out[16]:
```
0
```

In [17]:
```
row_0=row_0.rename('firstRow')
```

In [18]:
```
row_0.name
```

Out[18]:
```
'firstRow'
```

In [19]:
```
tags.head()
```

Out[19]:

	userId	movieId	tag
0	18	4141	Mark Waters
1	65	208	dark hero
2	65	353	dark hero
3	65	521	noir thriller
4	65	592	dark hero

In [20]:
```
tags.tail()
```

Out[20]:

	userId	movieId	tag
465559	138446	55999	dragged
465560	138446	55999	Jason Bateman
465561	138446	55999	quirky
465562	138446	55999	sad
465563	138472	923	rise to power

In [21]:
```
tags.index
```

Out[21]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [22]:
```
tags.columns
```

Out[22]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [23]:
```
tags.iloc[[0,33,500]]
```

Out[23]:

	userId	movieId	tag
0	18	4141	Mark Waters
33	65	58652	girls who play boys
500	342	55908	entirely dialogue

In [24]:
```
ratings['rating'].describe()
```

Out[24]:
```
count    2.000026e+07
mean     3.525529e+00
std      1.051989e+00
min      5.000000e-01
25%      3.000000e+00
50%      3.500000e+00
75%      4.000000e+00
max      5.000000e+00
Name: rating, dtype: float64
```

In [25]:
```
ratings.describe()
```

Out[25]:

	userId	movieId	rating
count	2.000026e+07	2.000026e+07	2.000026e+07
mean	6.904587e+04	9.041567e+03	3.525529e+00
std	4.003863e+04	1.978948e+04	1.051989e+00
min	1.000000e+00	1.000000e+00	5.000000e-01
25%	3.439500e+04	9.020000e+02	3.000000e+00
50%	6.914100e+04	2.167000e+03	3.500000e+00
75%	1.036370e+05	4.770000e+03	4.000000e+00
max	1.384930e+05	1.312620e+05	5.000000e+00

In [26]:
```
ratings.describe().transpose()
```

Out[26]:

	count	mean	std	min	25%	50%	75%	max
userId	20000263.0	69045.872583	40038.626653	1.0	34395.0	69141.0	103637.0	138493.0
movieId	20000263.0	9041.567330	19789.477445	1.0	902.0	2167.0	4770.0	131262.0
rating	20000263.0	3.525529	1.051989	0.5	3.0	3.5	4.0	5.0

In [27]:
```
ratings.describe()
```

Out[27]:

	userId	movieId	rating
count	2.000026e+07	2.000026e+07	2.000026e+07
mean	6.904587e+04	9.041567e+03	3.525529e+00
std	4.003863e+04	1.978948e+04	1.051989e+00
min	1.000000e+00	1.000000e+00	5.000000e-01
25%	3.439500e+04	9.020000e+02	3.000000e+00
50%	6.914100e+04	2.167000e+03	3.500000e+00
75%	1.036370e+05	4.770000e+03	4.000000e+00
max	1.384930e+05	1.312620e+05	5.000000e+00

In [28]:
```
ratings['rating'].mean()
```

Out[28]:
```
3.5255285642993797
```

In [29]:
```
ratings.mean()
```

Out[29]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [30]:
```
ratings['rating'].min()
```

Out[30]:
```
0.5
```

In [31]:
```
ratings.min()
```

Out[31]:
```
userId     1.0
movieId    1.0
rating     0.5
dtype: float64
```

In [32]:
```
ratings['rating'].max()
```

Out[32]:
```
5.0
```

In [33]:
```
ratings.max()
```

Out[33]:
```
userId     138493.0
movieId    131262.0
rating          5.0
dtype: float64
```

In [34]:
```
ratings['rating'].std()
```

Out[34]:
```
1.051988919275684
```

In [35]:
```
ratings.std()
```

Out[35]:
```
userId     40038.626653
movieId    19789.477445
rating         1.051989
dtype: float64
```

In [36]:
```
ratings['rating'].mode()
```

Out[36]:
```
0    4.0
Name: rating, dtype: float64
```

In [37]:
```
ratings.mode()
```

Out[37]:

	userId	movieId	rating
0	118205	296	4.0

In [38]:
```
ratings.corr()
```

Out[38]:

	userId	movieId	rating
userId	1.000000	-0.000850	0.001175
movieId	-0.000850	1.000000	0.002606
rating	0.001175	0.002606	1.000000

In [39]:
```
filter1=ratings['rating']>10
print(filter1)
filter1.any()
```

```
0           False
1           False
2           False
3           False
4           False
            ...
20000258    False
20000259    False
20000260    False
20000261    False
20000262    False
Name: rating, Length: 20000263, dtype: bool
```

Out[39]:
```
False
```

In [40]:
```
filter2=ratings['rating']>0
filter2.all()
```

Out[40]:
```
True
```

In [41]:
```
movies.shape
```

Out[41]:
```
(27278, 3)
```

In [42]:
```
movies.isnull().any().any()
```

Out[42]:
```
False
```

In [43]:
```
ratings.shape
```

Out[43]:
```
(20000263, 3)
```

In [44]:
```
ratings.isnull().any().any()
```

Out[44]:
```
False
```

In [45]:
```
tags.shape
```

Out[45]:
```
(465564, 3)
```

In [46]:
```
tags.isnull().any().any()
```

Out[46]:
```
True
```

In [47]:
```
tags=tags.dropna()
```

In [48]:
```
tags.isnull().any().any()
```

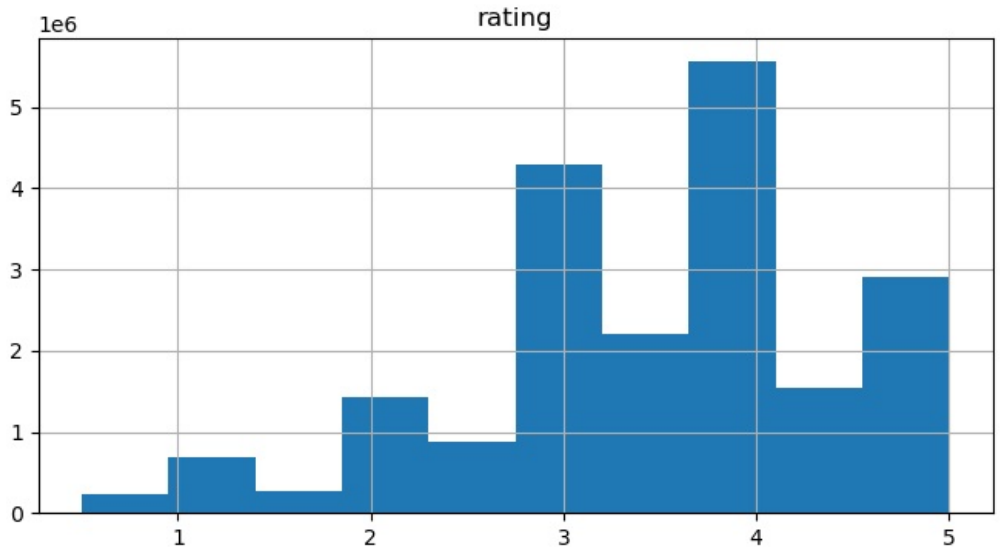
Out[48]:
```
False
```

In [49]:
```
tags.shape
```

Out[49]:
```
(465548, 3)
```

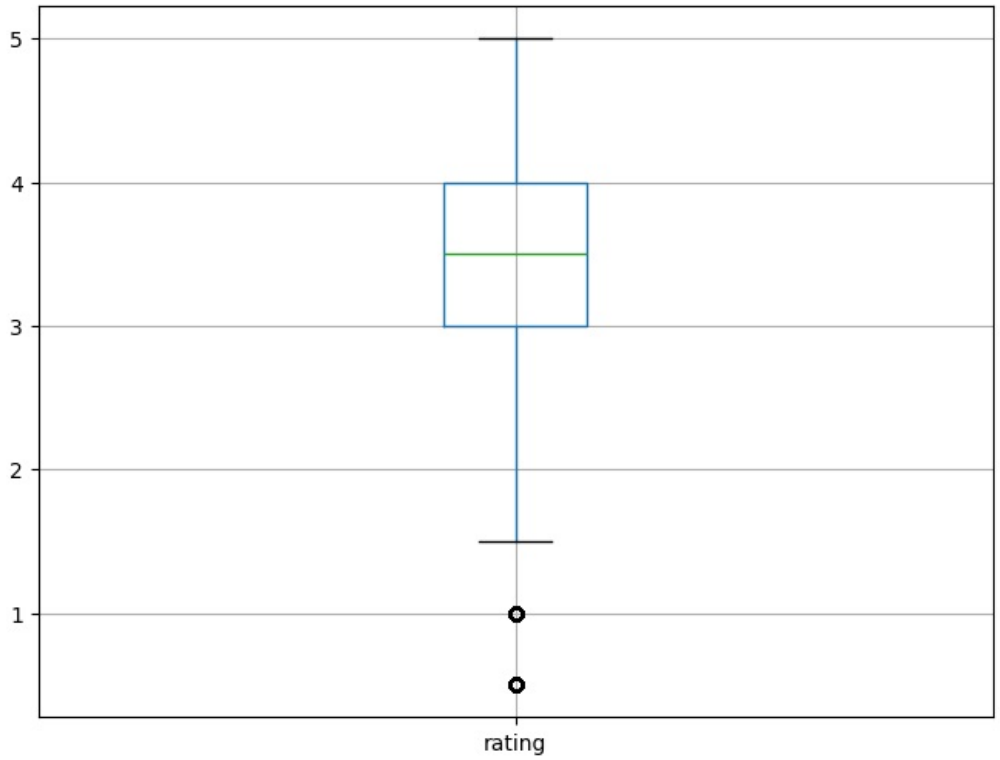
In [50]:
```
%matplotlib inline
ratings.hist(column='rating', figsize=(8,4))
```

Out[50]:
```
array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```



In [51]:
```
ratings.boxplot(column='rating', figsize=(8,6))
```

Out[51]: <Axes: >



In [52]:
```
tags['tag'].head()
```

Out[52]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [53]:
```
movies[['title','genres']].head()
```

Out[53]:

	title	genres
0	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	Jumanji (1995)	Adventure|Children|Fantasy
2	Grumpier Old Men (1995)	Comedy|Romance
3	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	Father of the Bride Part II (1995)	Comedy

In [54]:
```
tags['tag'].tail()
```

Out[54]:
```
465559          dragged
465560    Jason Bateman
465561           quirky
465562              sad
465563    rise to power
Name: tag, dtype: object
```

In [55]:
```
ratings[-19:]
```

Out[55]:

	userId	movieId	rating
20000244	138493	55269	5.0
20000245	138493	55814	5.0
20000246	138493	56757	3.0
20000247	138493	56801	3.0
20000248	138493	58879	4.5
20000249	138493	59315	4.0
20000250	138493	59725	3.0
20000251	138493	59784	5.0
20000252	138493	60069	4.0
20000253	138493	60816	4.5
20000254	138493	61160	4.0
20000255	138493	65682	4.5
20000256	138493	66762	4.5
20000257	138493	68319	4.5
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

In [56]:
```
ratings[20:]
```

Out[56]:

	userId	movieId	rating
20	1	924	3.5
21	1	1009	3.5
22	1	1036	4.0
23	1	1079	4.0
24	1	1080	3.5
...	...	...	...
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

20000243 rows × 3 columns

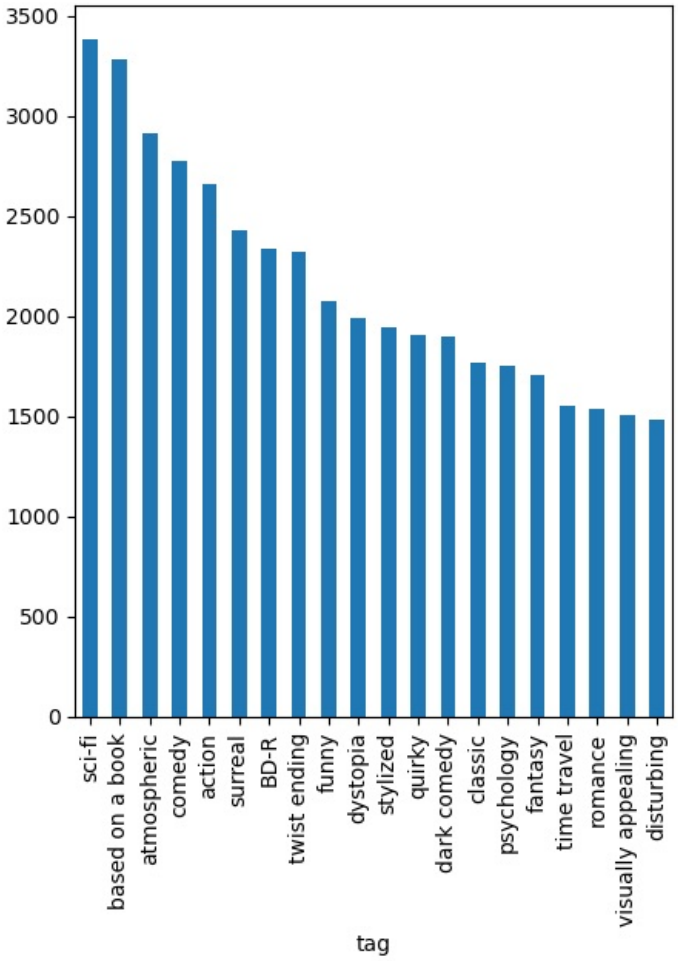
In [57]:
```
tags_counts=tags['tag'].value_counts()
```

In [58]:
```
tags_counts[-10:]
```

Out[58]:
```
tag
missing child                   1
Ron Moore                       1
Citizen Kane                    1
mullet                          1
biker gang                      1
Paul Adelstein                  1
the wig                         1
killer fish                     1
genetically modified monsters   1
topless scene                   1
Name: count, dtype: int64
```

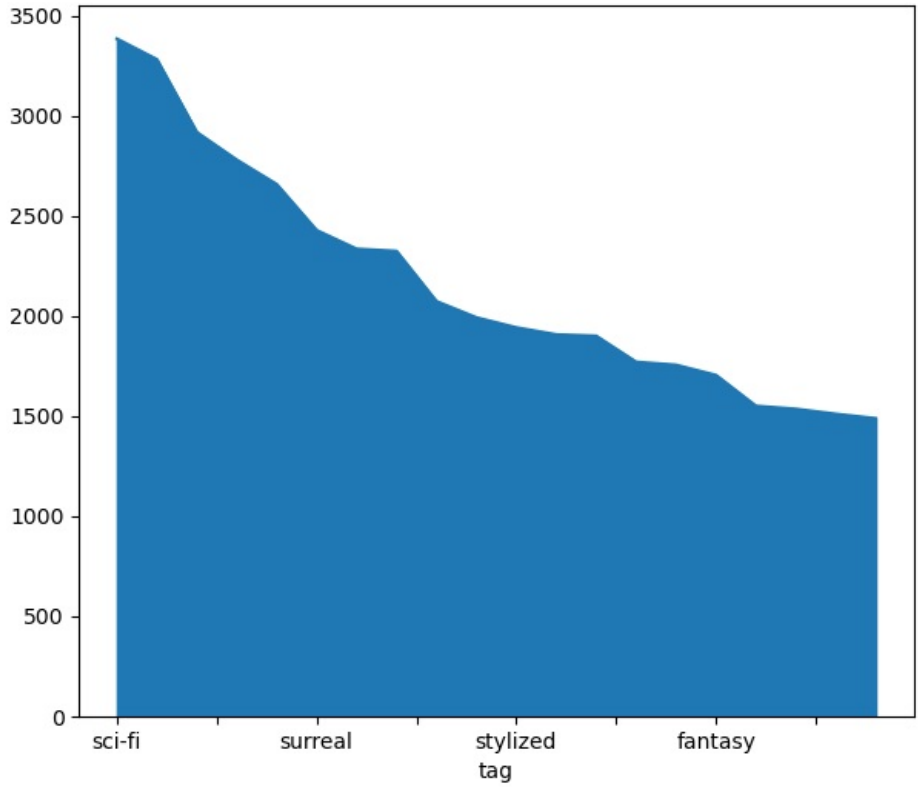
In [59]:
```
tags_counts[:20].plot(kind='bar', figsize=(5,6))
```

Out[59]:
```
<Axes: xlabel='tag'>
```



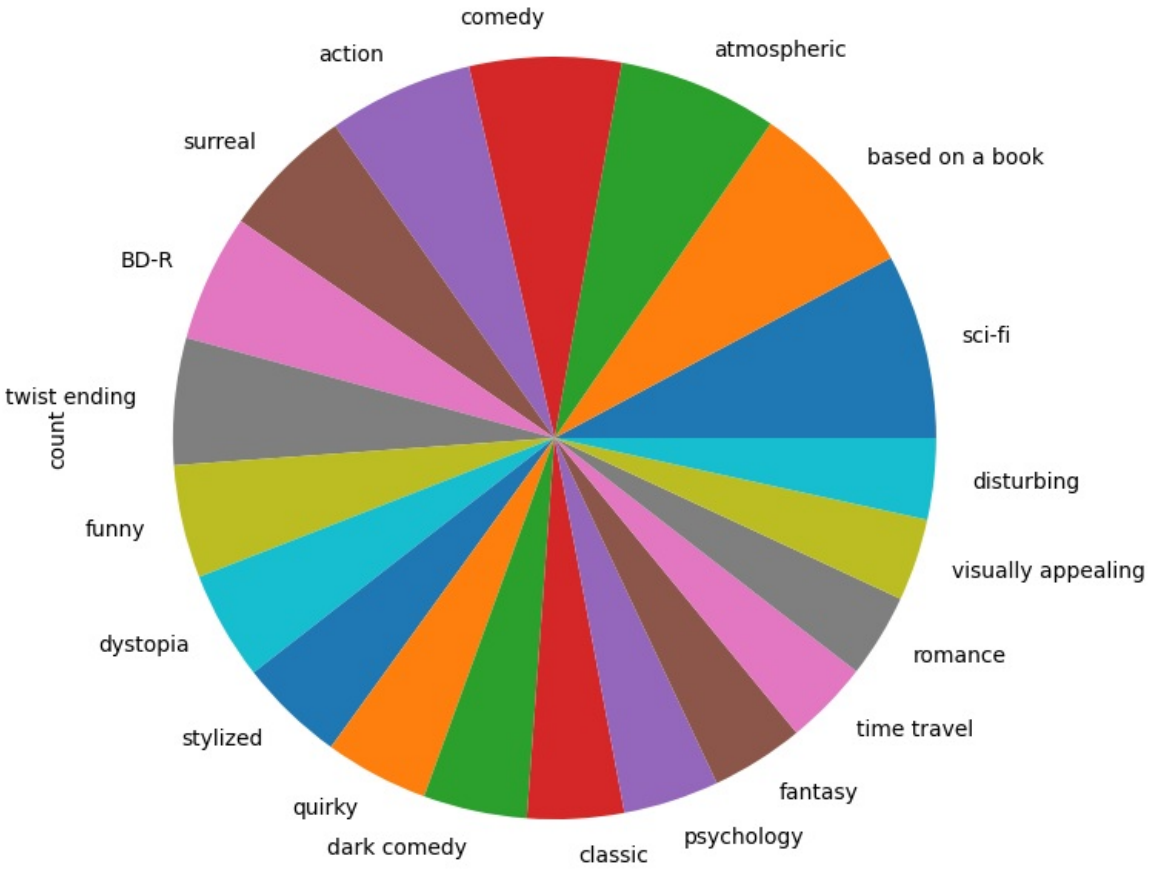
In [60]:
```
tags_counts[:20].plot(kind='area', figsize=(7,6))
```

Out[60]:
```
<Axes: xlabel='tag'>
```



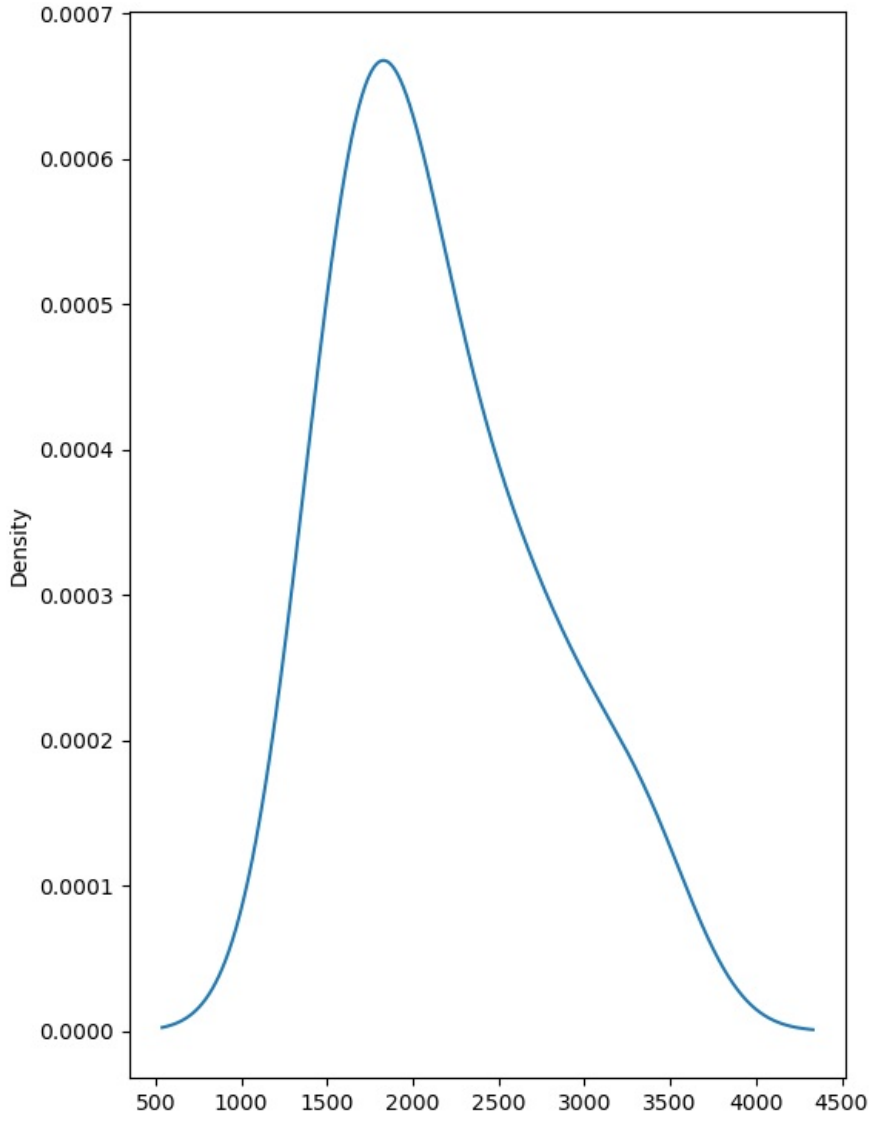
In [61]:
```
tags_counts[:20].plot(kind='pie', figsize=(8,9))
```

Out[61]: <Axes: ylabel='count'>



In [69]:
```
tags_counts[:20].plot(kind='kde', figsize=(6,9))
```

Out[69]:
```
<Axes: ylabel='Density'>
```



In [ ]: