In [1]:
```
# DATASET :- MOVIELENS 20M
# PURPOSE :- ANALYZE THE RATINGS AND TAGS OF MOVIES
# MOVIELENS :- A WEB-BASED RECOMMENDATION SYSTEM & VIRTUAL COMMUNITY THAT MAKES MOVIE RECOMMENDATIONS BASED ON USER
 PREFERENCES AND MOVIE RATINGS
# MOVIELENS 20M :- 20 MILLION RATINGS AND 465564 TAG APPLICATIONS APPLIED TO 27278 MOVIESBY 138000 USERS
```

In [2]:
```
# IMPORT LIBRARIES
```

In [3]:
```
import pandas as pd # IMPORTING PANDAS LIBRARY
```

In [4]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [8]:
```
movies = pd.read_csv(r'C:\Users\Lenovo\Downloads\archive\movie.csv',sep=',')
# IMPORT MOVIE DATASET
# sep => SPECIFYA CUSTOM DELIMETER (CHARACTER LIKE COMMA etc THAT SEPARATES DATA)
```

In [9]:
```
print(type(movies))
```
```
<class 'pandas.core.frame.DataFrame'>
```

In [10]:
```
movies.head(20) # TOP 20 MOVIES DATA PRINTED
```
Out[10]:

	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
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller
10	11	American President, The (1995)	Comedy|Drama|Romance
11	12	Dracula: Dead and Loving It (1995)	Comedy|Horror
12	13	Balto (1995)	Adventure|Animation|Children
13	14	Nixon (1995)	Drama
14	15	Cutthroat Island (1995)	Action|Adventure|Romance
15	16	Casino (1995)	Crime|Drama
16	17	Sense and Sensibility (1995)	Drama|Romance
17	18	Four Rooms (1995)	Comedy
18	19	Ace Ventura: When Nature Calls (1995)	Comedy
19	20	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller

In [11]:
```
tags = pd.read_csv(r'C:\Users\Lenovo\Downloads\archive\tag.csv',sep=',')
# IMPORT TAG DATASET
```

In [12]:
```
tags.head() # TOP 5 TAGS PRINTED
```
Out[12]:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40
1	65	208	dark hero	2013-05-10 01:41:18
2	65	353	dark hero	2013-05-10 01:41:19
3	65	521	noir thriller	2013-05-10 01:39:43
4	65	592	dark hero	2013-05-10 01:41:18

In [13]:
```
ratings = pd.read_csv(r'C:\Users\Lenovo\Downloads\archive\rating.csv',sep=',',parse_dates=['timestamp'])
# IMPORT RATING DATASET
# parse_dates='timestamp' => IF TRUE,PANDAS WILL ATTEMPT TO GUESS THE FORMAT OF DATETIME STRINGS
```

In [16]:
```
ratings.head()
```
Out[16]:

	userId	movieId	rating	timestamp
0	1	2	3.5	2005-04-02 23:53:47
1	1	29	3.5	2005-04-02 23:31:16
2	1	32	3.5	2005-04-02 23:33:39
3	1	47	3.5	2005-04-02 23:32:07
4	1	50	3.5	2005-04-02 23:29:40

In [17]:
```
del ratings['timestamp'] # REMOVE TIMESTAMP COLUMN FROM RATINGS & TAGS
del tags['timestamp']
```

In [18]:
```
# DATA STRUCTURES
```

In [19]:
```
row_0 = tags.iloc[0] # iloc=> HELPS TO SELECT A SPECIFIC ROW OR COLUMN FROM DATASET
type(row_0)
```
Out[19]:
```
pandas.core.series.Series
```

In [20]:
```
print(row_0) # ZEROTH ROW ELEMENTS
```
```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [21]:
```
row_0.index
```
Out[21]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [22]:
```
row_0['userId']
```
Out[22]:
```
18
```

In [23]:
```
'rating' in row_0 # IN row_0 THERE IS NO RATING AS THERE IS NO RATING COLUMN IN TAGS
```
Out[23]:
```
False
```

In [24]:
```
row_0.name # NAME OF ZEROTH ROW
```
Out[24]:
```
0
```

In [25]:
```
row_0 = row_0.rename('firstRow')
row_0.name
```
Out[25]:
```
'firstRow'
```

In [26]:
```
# DATA FRAMES
```

In [27]:
```
tags.head()
```
Out[27]:

	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 [28]:
```
tags.index
```
Out[28]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [29]:
```
tags.columns
```
Out[29]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [32]:
```
tags.iloc[[0,11,500]]
```
Out[32]:

	userId	movieId	tag
0	18	4141	Mark Waters
11	65	1783	noir thriller
500	342	55908	entirely dialogue

In [33]:
```
# DESCRIPTIVE STATISTICS
```

In [34]:
```
ratings['rating']
```
Out[34]:
```
0           3.5
1           3.5
2           3.5
3           3.5
4           3.5
5           3.5
6           4.0
7           4.0
8           4.0
9           4.0
10          4.0
11          4.0
12          4.0
13          3.5
14          3.5
15          4.0
16          3.5
17          3.5
18          3.0
19          3.5
20          3.5
21          3.5
22          4.0
23          4.0
24          3.5
25          3.5
26          4.0
27          4.0
28          3.5
29          3.5
           ...
20000233    3.5
20000234    3.5
20000235    4.5
20000236    4.5
20000237    4.0
20000238    4.0
20000239    4.0
20000240    3.0
20000241    4.0
20000242    4.0
20000243    4.5
20000244    5.0
20000245    5.0
20000246    3.0
20000247    3.0
20000248    4.5
20000249    4.0
20000250    3.0
20000251    5.0
20000252    4.0
20000253    4.5
20000254    4.0
20000255    4.5
20000256    4.5
20000257    4.5
20000258    4.5
20000259    4.5
20000260    3.0
20000261    5.0
20000262    2.5
Name: rating, Length: 20000263, dtype: float64
```

In [35]:
```
ratings['rating'].describe() # DESCRIPTIVE DATA OF RATING COLUMN
```
Out[35]:
```
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 [36]:
```
ratings.describe() # DESCRIPTIVE DATA OF RATINGS DATASET
```
Out[36]:

	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 [37]:
```
ratings['rating'].mean() # MEAN OF RATING COLUMN
```
Out[37]:
```
3.5255285642993797
```

In [38]:
```
ratings.mean() # MEAN OF RATINGS DATASET
```
Out[38]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [39]:
```
ratings['rating'].min() # MINIMUM VALUE OF RATING COLUMN
```
Out[39]:
```
0.5
```

In [40]:
```
ratings['rating'].max() # MAXIMUM VALUE OF RATING COLUMN
```
Out[40]:
```
5.0
```

In [41]:
```
ratings['rating'].std() # STANDARD DEVIATION : HOW DISPERSED THE DATA IS IN RELATION TO MEAN
```
Out[41]:
```
1.051988919275684
```

In [42]:
```
ratings['rating'].mode() # MODE : REPEATED NO.
```
Out[42]:
```
0    4.0
dtype: float64
```

In [43]:
```
ratings.corr() # CALCULATES THE RELATIONSHIP B/W EACH COLUMN IN DATASET

# CORRELATION : EXPRESS THE EXTENT TO WHICH 2 VARIABLES ARE LINEARLY RELATED (-1 TO +1)
```
Out[43]:

	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 [44]:
```
filter1 = ratings['rating'] > 10
print(filter1)
```
```
0           False
1           False
2           False
3           False
4           False
5           False
6           False
7           False
8           False
9           False
10          False
11          False
12          False
13          False
14          False
15          False
16          False
17          False
18          False
19          False
20          False
21          False
22          False
23          False
24          False
25          False
26          False
27          False
28          False
29          False
            ...
20000233    False
20000234    False
20000235    False
20000236    False
20000237    False
20000238    False
20000239    False
20000240    False
20000241    False
20000242    False
20000243    False
20000244    False
20000245    False
20000246    False
20000247    False
20000248    False
20000249    False
20000250    False
20000251    False
20000252    False
20000253    False
20000254    False
20000255    False
20000256    False
20000257    False
20000258    False
20000259    False
20000260    False
20000261    False
20000262    False
Name: rating, Length: 20000263, dtype: bool
```

In [45]:
```
filter1.any() # PRINTS ONE VALUE FOR EACH COLUMN,True IF ANY VALUE IN THAT COLUMN IS TRUE OTHERWISE False
```
Out[45]:
```
False
```

In [46]:
```
filter2 = ratings['rating'] > 0 # PRINTS ONE VALUE FOR EACH COLUMN,True IF ALL VALUE IN THAT COLUMN ARE TRUE,OTHERWI
SE False
filter2.all()
```
Out[46]:
```
True
```

In [47]:
```
# DATA CLEANING : HANDLING MISSING DATA
```

In [48]:
```
movies.shape # TOTAL NO. OF ROWS AND COLUMNS
```
Out[48]:
```
(27278, 3)
```

In [49]:
```
movies.isnull().any().any() # NO NULL VALUES # True-NULL VALUE, False-NO NULL VALUE
```
Out[49]:
```
False
```

In [50]:
```
ratings.shape
```
Out[50]:
```
(20000263, 3)
```

In [51]:
```
ratings.isnull().any().any()
```
Out[51]:
```
False
```

In [52]:
```
tags.shape
```
Out[52]:
```
(465564, 3)
```

In [53]:
```
tags.isnull().any().any() # HAVE NULL VALUE
```
Out[53]:
```
True
```

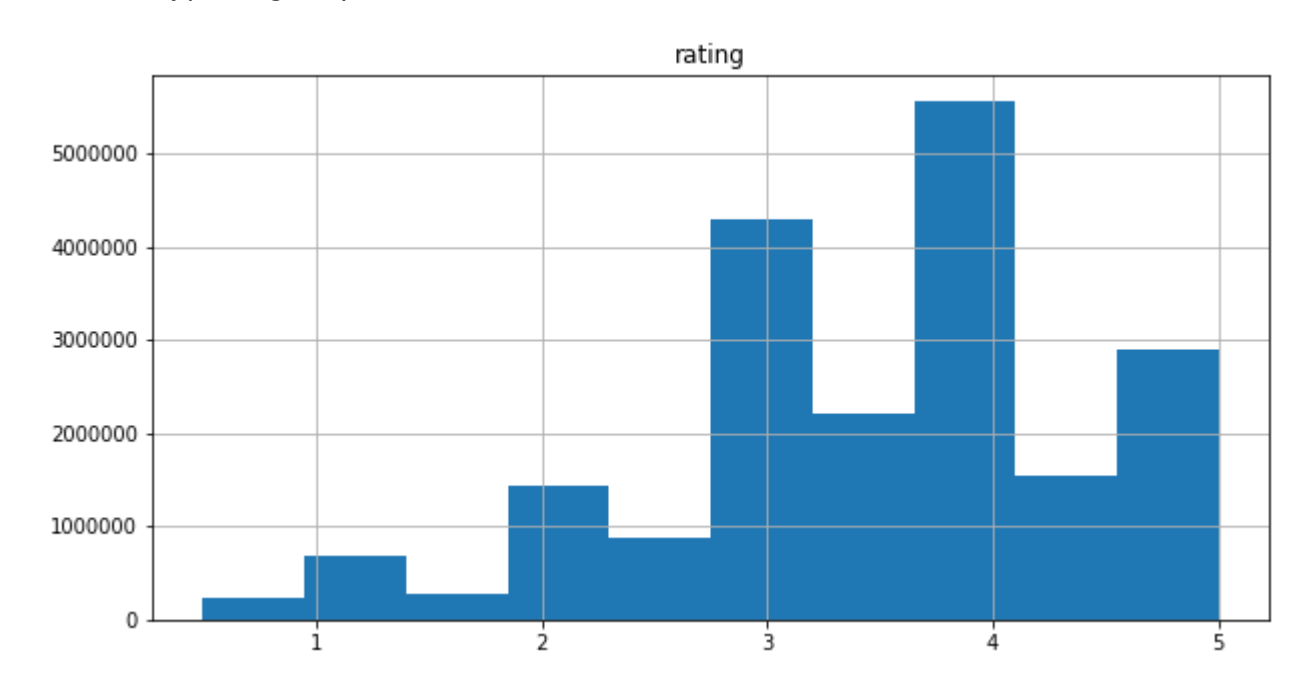
In [54]:
```
tags = tags.dropna() # REMOVE THE NULL VALUES
```

In [55]:
```
tags.isnull().any().any()
```
Out[55]:
```
False
```

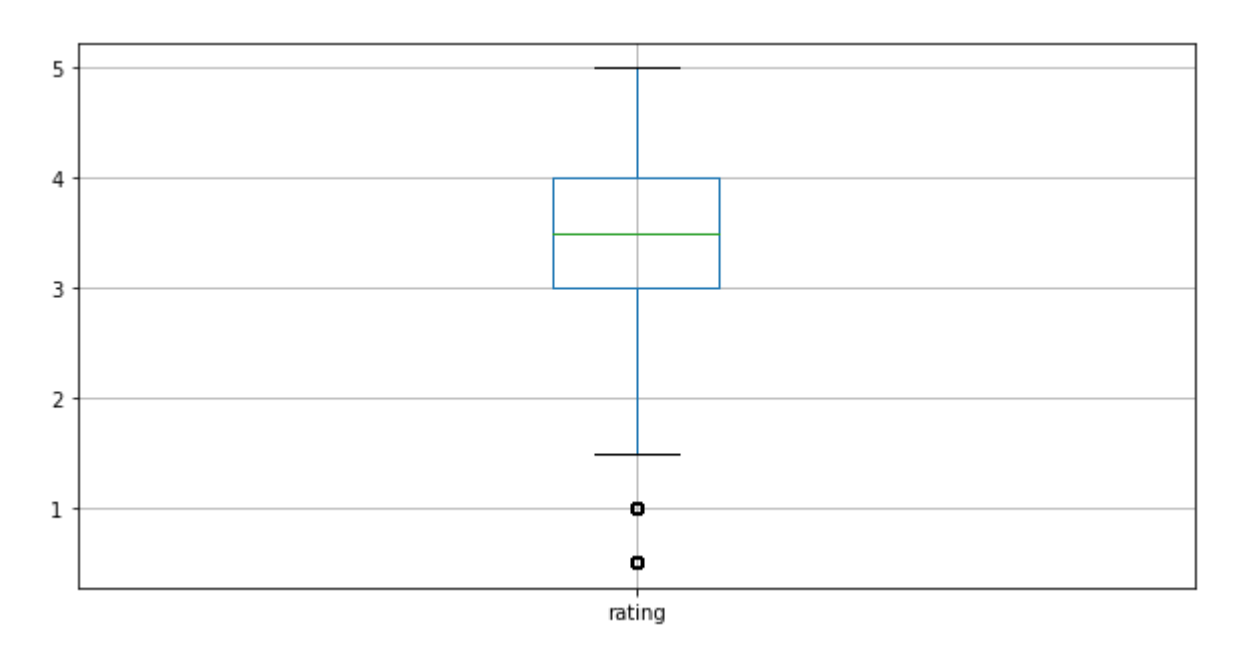
In [56]:
```
tags.shape
```
Out[56]:
```
(465548, 3)
```

In [57]:
```
# DATA VISUALIZATION
```

In [58]:
```
%matplotlib inline
ratings.hist(column='rating',figsize=(10,5)) # PLOTTING HISTOGRAM
```
Out[58]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000000043A87940>]],
      dtype=object)
```



In [59]:
```
ratings.boxplot(column='rating',figsize=(10,5)) # PLOTTING BOXPLOT
```
Out[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x59fc3c88>
```



In [60]:
```
# SLICING OUT COLUMNS
```

In [61]:
```
tags['tag'].head() # TAG COLUMN FIRST 5 VALUES
```
Out[61]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [62]:
```
movies[['title','genres']].head() # TITLE & GENRES COLUMN FIRST 5 VALUES
```
Out[62]:

	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

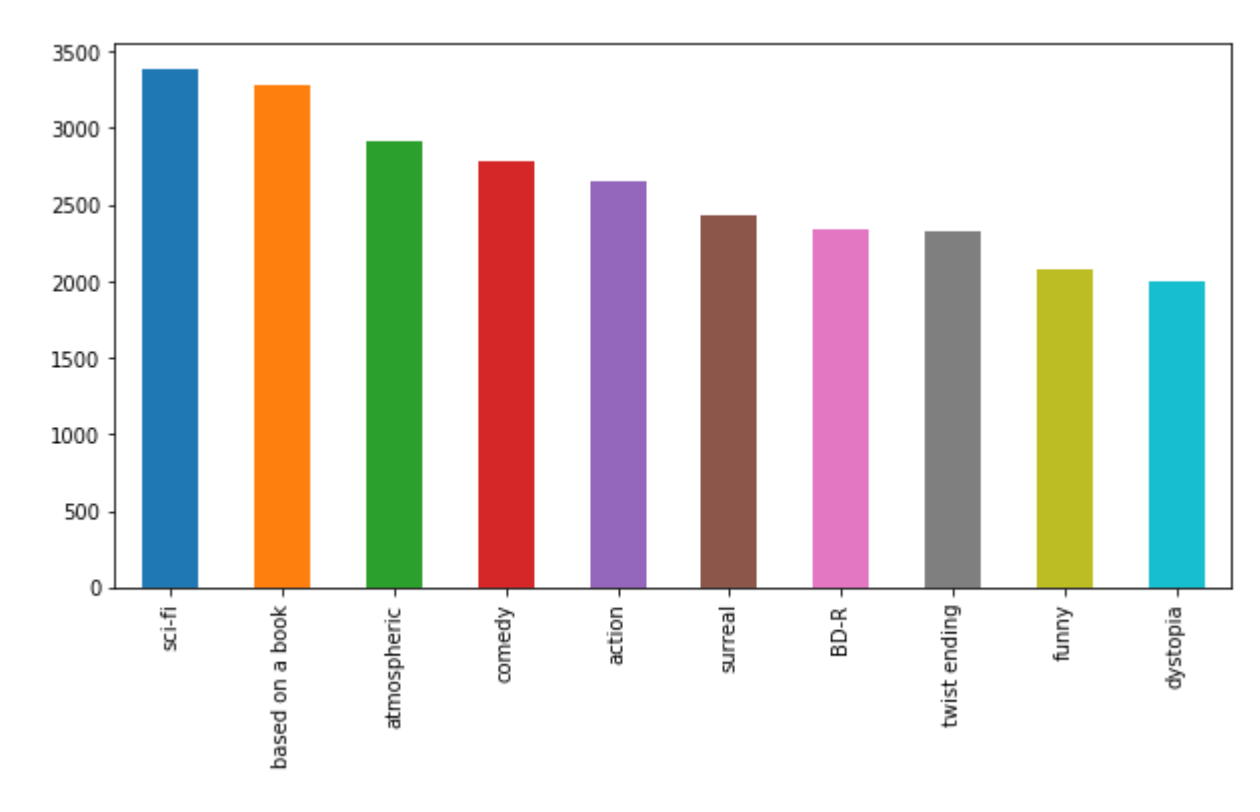
In [63]:
```
ratings[-10:]
```
Out[63]:

	userId	movieId	rating
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 [64]:
```
tag_counts = tags['tag'].value_counts() # SERIES CONTAINING COUNTS OF UNIQUE VALUES
tag_counts
```
Out[64]:
```
sci-fi                                                3384
based on a book                                       3281
atmospheric                                           2917
comedy                                                2779
action                                                2657
surreal                                               2427
BD-R                                                  2334
twist ending                                          2323
funny                                                 2072
dystopia                                              1991
stylized                                              1941
quirky                                                1906
dark comedy                                           1899
classic                                               1769
psychology                                            1754
fantasy                                               1703
time travel                                           1549
romance                                               1534
visually appealing                                    1509
disturbing                                            1487
aliens                                                1428
thought-provoking                                     1422
social commentary                                     1417
Nudity (Topless)                                      1400
violence                                              1336
drugs                                                 1312
Criterion                                             1286
true story                                            1276
nudity (topless)                                      1245
adventure                                             1243
                                                      ...
Sexual                                                   1
silhouettes                                              1
I love the powerglove                                    1
Facing Mortality                                         1
color symbolism                                          1
in the true meaning of the word:awesome                  1
too much sex                                             1
AFI #79                                                  1
Farley Granger                                           1
Morgan Neville                                           1
Henry de Sliva                                           1
Imperialism                                              1
mental illnesses vs. sanity                              1
Richard Rush                                             1
consensual sex                                           1
small budget                                             1
Tim Burton Hits Another One Out Of The Cemetery          1
exploding helicopter                                     1
saliva song rocks                                        1
karrin's favorite                                        1
Fletcher Markle                                          1
Not Another Mingella Stretcher! We Hates Them!           1
jonathan rhys meyers                                     1
perfect entertainment                                    1
brainless fun                                            1
Restitution                                              1
sailor and lula                                          1
Detroit MI                                               1
3-hours of sex and drugs                                 1
rescue mission                                           1
Name: tag, Length: 38643, dtype: int64
```

In [65]:
```
tag_counts[-10:]
```
Out[65]:
```
Fletcher Markle                                   1
Not Another Mingella Stretcher! We Hates Them!    1
jonathan rhys meyers                              1
perfect entertainment                             1
brainless fun                                     1
Restitution                                       1
sailor and lula                                   1
Detroit MI                                        1
3-hours of sex and drugs                          1
rescue mission                                    1
Name: tag, dtype: int64
```

In [66]:
```
tag_counts[:10].plot(kind='bar',figsize=(10,5))
```
Out[66]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x5bc7cf60>
```



In [ ]: