QUESTION 1

In [2]:
```
import pandas as pd
import numpy as np
```

In [12]:
```
df = pd.read_csv('popularity.csv')
```

1.1

In [13]:
```
df.drop('Unnamed: 0', axis=1, inplace=True)
df
```

Out[13]:

	avg_shares	avg_comments	avg_expert	popularity_score
0	147.3	23.9	19.1	14.6
1	28.6	1.5	33.0	7.3
2	17.9	37.6	21.6	8.0
3	94.2	4.9	8.1	9.7
4	293.6	27.7	1.8	20.7
...	...	...	...	...
195	4.1	11.6	5.7	3.2
196	76.4	26.7	22.3	11.8
197	218.5	5.4	27.4	12.2
198	140.3	1.9	9.0	10.3
199	266.9	43.8	5.0	25.4

200 rows × 4 columns

1.2

In [14]:
```
null_filter = df['avg_shares'].isnull()
df[null_filter]
```

Out[14]:

	avg_shares	avg_comments	avg_expert	popularity_score
19	NaN	7.6	7.2	9.7

In [15]:
```
mean = df['avg_shares'].mean()
df['avg_shares'].fillna(mean, inplace=True)
df[19:20]
```

Out[15]:

	avg_shares	avg_comments	avg_expert	popularity_score
19	147.291457	7.6	7.2	9.7

In [16]:
```
null_filter = df['avg_comments'].isnull()
df[null_filter]
```

Out[16]:

	avg_shares	avg_comments	avg_expert	popularity_score
7	168.4	NaN	12.8	11.7
26	202.5	NaN	31.6	16.6
37	163.5	NaN	7.4	18.0
45	70.6	NaN	40.8	10.5

In [17]:
```
mean = df['avg_comments'].mean()
df['avg_comments'].fillna(mean, inplace=True)
df[7:46]
```

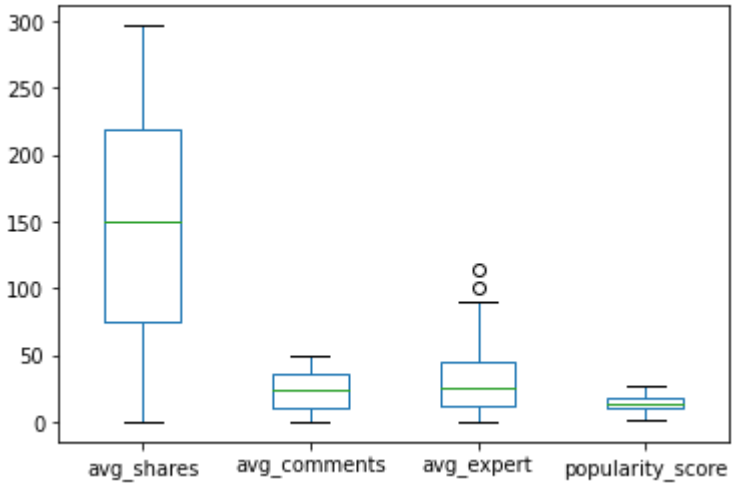
Out[17]:

	avg_shares	avg_comments	avg_expert	popularity_score
7	168.400000	23.319388	12.8	11.7
8	280.200000	10.100000	21.4	14.8
9	19.400000	16.000000	22.3	6.6
10	107.400000	14.000000	10.9	11.5
11	177.000000	9.300000	6.4	12.8
12	296.400000	36.300000	100.9	23.8
13	237.400000	27.500000	11.0	18.9
14	232.100000	8.600000	8.7	13.4
15	206.900000	8.400000	26.4	12.9
16	131.100000	42.800000	28.9	18.0
17	191.100000	28.700000	18.2	17.3
18	151.500000	41.300000	58.5	18.5
19	147.291457	7.600000	7.2	9.7
20	120.200000	19.600000	11.6	13.2
21	43.100000	26.700000	35.1	10.1
22	197.600000	3.500000	5.9	11.7
23	239.300000	15.500000	27.3	15.7
24	74.700000	49.400000	45.7	14.7
25	109.800000	14.300000	31.7	12.4
26	202.500000	23.319388	31.6	16.6
27	141.300000	26.800000	46.2	15.5
28	27.500000	1.600000	20.7	6.9
29	38.200000	3.700000	13.8	7.6
30	95.700000	1.400000	7.4	9.5
31	248.400000	30.200000	20.3	20.2
32	205.000000	45.100000	19.6	22.6
33	67.800000	36.600000	114.0	12.5
34	261.300000	42.700000	54.7	24.2
35	117.200000	14.700000	5.4	11.9
36	171.300000	39.700000	37.7	19.0
37	163.500000	23.319388	7.4	18.0
38	240.100000	7.300000	8.7	13.2
39	240.100000	16.700000	22.9	15.9
40	239.900000	41.500000	18.5	23.2
41	292.900000	28.300000	43.2	21.4
42	104.600000	5.700000	34.4	10.4
43	109.800000	47.800000	51.4	16.7
44	289.700000	42.300000	51.2	25.4
45	70.600000	23.319388	40.8	10.5

1.3

In [18]:
```
df.boxplot(column =['avg_shares','avg_comments','avg_expert','popularity_score'], grid =
```

Out[18]: <AxesSubplot:>



1.4

In [19]:
```
from sklearn import preprocessing
d = preprocessing.normalize(df)
scaled_df = pd.DataFrame(d)
scaled_df.head(200)
```

Out[19]:

	0	1	2	3
0	0.974525	0.158121	0.126364	0.096592
1	0.645597	0.033860	0.744919	0.164785
2	0.376136	0.790096	0.453885	0.168105
3	0.989807	0.051487	0.085111	0.101923
4	0.993117	0.093697	0.006089	0.070019
...	...	...	...	...
195	0.294287	0.832617	0.409131	0.229687
196	0.901235	0.314961	0.263057	0.139196
197	0.990413	0.024477	0.124198	0.055300
198	0.995191	0.013477	0.063840	0.073061
199	0.982311	0.161204	0.018402	0.093483

200 rows × 4 columns

QUESTION 2

2.1

In [20]:
```
df = pd.read_csv('test.csv')
```

In [21]:
```
null_filter = df['Age'].isnull()
df[null_filter]
```

Out[21]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
10	902	3	Ilieff, Mr. Ylio	male	NaN	0	0	349220	7.8958	NaN	S
22	914	1	Flegenheim, Mrs. Alfred (Antoinette)	female	NaN	0	0	PC 17598	31.6833	NaN	S
29	921	3	Samaan, Mr. Elias	male	NaN	2	0	2662	21.6792	NaN	C
33	925	3	Johnston, Mrs. Andrew G (Elizabeth Lily" Watson)"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
36	928	3	Roth, Miss. Sarah A	female	NaN	0	0	342712	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
408	1300	3	Riordan, Miss. Johanna Hannah""	female	NaN	0	0	334915	7.7208	NaN	Q
410	1302	3	Naughton, Miss. Hannah	female	NaN	0	0	365237	7.7500	NaN	Q
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

86 rows × 11 columns

In [22]:
```
mean = df['Age'].mean()
df['Age'].fillna(mean, inplace=True)
df
```

Out[22]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	892	3	Kelly, Mr. James	male	34.50000	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.00000	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.00000	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.00000	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.00000	1	1	3101298	12.2875	NaN	
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	30.27259	0	0	A.5. 3236	8.0500	NaN	
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.00000	0	0	PC 17758	108.9000	C105	
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.50000	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
416	1308	3	Ware, Mr. Frederick	male	30.27259	0	0	359309	8.0500	NaN	
417	1309	3	Peter, Master. Michael J	male	30.27259	1	1	2668	22.3583	NaN	

418 rows × 11 columns

2.2

In [23]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df['Sex'] = le.fit_transform(df["Sex"])
df['Embarked'] = le.fit_transform(df["Embarked"])
df
```

Out[23]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	892	3	Kelly, Mr. James	1	34.50000	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	0	47.00000	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	1	62.00000	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	1	27.00000	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.00000	1	1	3101298	12.2875	NaN	
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	1	30.27259	0	0	A.5. 3236	8.0500	NaN	
414	1306	1	Oliva y Ocana, Dona. Fermina	0	39.00000	0	0	PC 17758	108.9000	C105	
415	1307	3	Saether, Mr. Simon Sivertsen	1	38.50000	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
416	1308	3	Ware, Mr. Frederick	1	30.27259	0	0	359309	8.0500	NaN	
417	1309	3	Peter, Master. Michael J	1	30.27259	1	1	2668	22.3583	NaN	

418 rows × 11 columns

In [28]:
```
drop_data = df.drop(['Pclass','SibSp','Parch','Cabin'], axis = 1)
drop_data.head()
```

Out[28]:

	PassengerId	Name	Sex	Age	Ticket	Fare	Embarked
0	892	Kelly, Mr. James	1	34.5	330911	7.8292	1
1	893	Wilkes, Mrs. James (Ellen Needs)	0	47.0	363272	7.0000	2
2	894	Myles, Mr. Thomas Francis	1	62.0	240276	9.6875	1
3	895	Wirz, Mr. Albert	1	27.0	315154	8.6625	2
4	896	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.0	3101298	12.2875	2

QUESTION 3

In [15]:
```
import PIL
from PIL import Image
from matplotlib import image
from matplotlib import pyplot
```

In [110]:
```
data = image.imread('SampleImage.jpg')
pyplot.imshow(data)
pyplot.show()
```

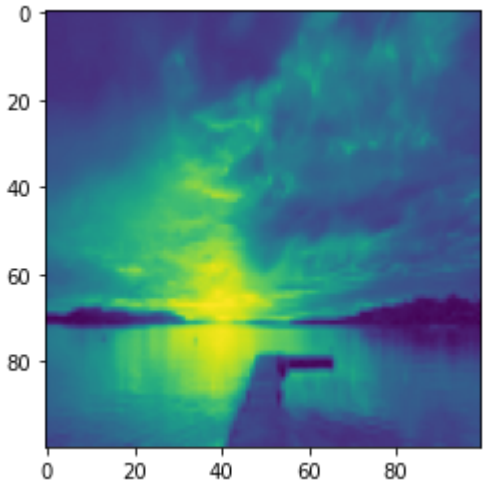
3.1

In [16]:
```
img = Image.open('SampleImage.jpg')
imgGray = img.convert('L')
imgGray.save('SampleImage_gray.jpg')
image2 = Image.open('SampleImage_gray.jpg')
```

3.2

In [17]:
```
image = PIL.Image.open("sampleImage.jpg")
resized_img = image.resize((100, 100))
```

In [18]:
```
pyplot.imshow(resized_img)
pyplot.show()
```



3.3

In [19]:
```
from numpy import asarray
data = asarray(image)
print(data)
```

```
[[  0   0   6 ...   0   5   0]
 [  7   0   0 ...  21   0 255]
 [  0  12 255 ...   0   8   0]
 ...
 [  0   9   0 ...   2 236  16]
 [  0 255   4 ...   2  23   0]
 [ 11   0   4 ...   0 255   3]]
```

QUESTION 4

```
4.1
```

In [3]:
```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
df = pd.read_csv('a3-Q4.csv')
def func(text):
    tokens = word_tokenize(text)
    return tokens
for i in df['content'].index:
    df['content'][i] = func(df['content'][i])
df
```

```
[nltk_data] Downloading package punkt to C:\Users\Nupur
[nltk_data]     goel\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
<ipython-input-3-4c1b679125a4>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df['content'][i] = func(df['content'][i])
```

Out[3]:

	tweet_id	sentiment	author	content
0	1956967341	empty	xoshayzers	[@, tiffanylue, i, know, i, was, listenin, to,...
1	1956967666	sadness	wannamama	[Layin, n, bed, with, a, headache, ughhhh, ...
2	1956967696	sadness	coolfunky	[Funeral, ceremony, ..., gloomy, friday, ...]
3	1956967789	enthusiasm	czareaquino	[wants, to, hang, out, with, friends, SOON, !]
4	1956968416	neutral	xkilljoyx	[@, dannycastillo, We, want, to, trade, with, ...
...	...	...	...	...
39995	1753918954	neutral	showMe_Heaven	[@, JohnLloydTaylor]
39996	1753919001	love	drapeaux	[Happy, Mothers, Day, All, my, love]
39997	1753919005	love	JenniRox	[Happy, Mother, 's, Day, to, all, the, mommies...
39998	1753919043	happiness	ipdaman1	[@, niariley, WASSUP, BEAUTIFUL, !, !, !, FOLL...
39999	1753919049	love	Alpharalpha	[@, mopedronin, bullet, train, from, tokyo, th...

40000 rows × 4 columns

4.2

In [24]:
```
import string
string.punctuation
def remove_punctuation(x):
    x_nopunct = [word for word in x if word not in string.punctuation]
    return x_nopunct
df['content_clean'] = df['content'].apply(lambda x: remove_punctuation(x))
df
```

Out[24]:

	tweet_id	sentiment	author	content	content_clean
0	1956967341	empty	xoshayzers	@tiffanylue i know i was listenin to bad habi...	[t, i, f, f, a, n, y, l, u, e, , i, , k, n, ...
1	1956967666	sadness	wannamama	Layin n bed with a headache ughhhh...waitin o...	[L, a, y, i, n, , n, , b, e, d, , w, i, t, ...
2	1956967696	sadness	coolfunky	Funeral ceremony...gloomy friday...	[F, u, n, e, r, a, l, , c, e, r, e, m, o, n, ...
3	1956967789	enthusiasm	czareaquino	wants to hang out with friends SOON!	[w, a, n, t, s, , t, o, , h, a, n, g, , o, ...
4	1956968416	neutral	xkilljoyx	@dannycastillo We want to trade with someone w...	[d, a, n, n, y, c, a, s, t, i, l, l, o, , W, ...
...	...	...	...	...	...
39995	1753918954	neutral	showMe_Heaven	@JohnLloydTaylor	[J, o, h, n, L, l, o, y, d, T, a, y, l, o, r]
39996	1753919001	love	drapeaux	Happy Mothers Day All my love	[H, a, p, p, y, , M, o, t, h, e, r, s, , D, ...
39997	1753919005	love	JenniRox	Happy Mother's Day to all the mommies out ther...	[H, a, p, p, y, , M, o, t, h, e, r, s, , D, ...
39998	1753919043	happiness	ipdaman1	@niariley WASSUP BEAUTIFUL!!! FOLLOW ME!! PEE...	[n, i, a, r, i, l, e, y, , W, A, S, S, U, P, ...
39999	1753919049	love	Alpharalpha	@mopedronin bullet train from tokyo the gf ...	[m, o, p, e, d, r, o, n, i, n, , b, u, l, l, ...

40000 rows × 5 columns

4.3

In [31]:
```
from nltk.corpus import stopwords
df = pd.read_csv('a3-Q4.csv')
d = df[['content']]
for i in d:
    data = d[i]
    text = data.to_string()
    tokens = word_tokenize(text)
    tokens = [w.lower() for w in tokens]
    words = [word for word in tokens if word.isalpha()]
    stop_words = stopwords.words('english')
    print(stop_words)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

4.4

In [29]:
```
from nltk.stem.porter import PorterStemmer
df = pd.read_csv('a3-Q4.csv')
d = df[['content']]
for i in d:
    data = d[i]
    text = data.to_string()
    tokens = word_tokenize(text)
    print(tokens[:100])
    porter = PorterStemmer()
    stemmed = [porter.stem(word) for word in tokens]
    print(stemmed[:100])
```

```
['0', '@', 'tiffanylue', 'i', 'know', 'i', 'was', 'listenin', 'to', 'bad', 'habi', '...', '1', 'Layin', 'n', 'bed', 'with', 'a', 'headache', 'ughhhh', '...', 'waitin', 'o', '...', '2', 'Funeral', 'ceremony', '...', 'gloomy', 'friday', '...', '3', 'wants', 'to', 'hang', 'out', 'with', 'friends', 'SOON', '!', '4', '@', 'dannycastillo', 'We', 'want', 'to', 'trade', 'with', 'someone', 'w', '...', '5', 'Re-pinging', '@', 'ghostridah14', ':', 'why', 'did', "n't", 'you', 'go', 'to', '...', '6', 'I', 'should', 'be', 'sleep', ',', 'but', 'im', 'not', '!', 'thinking', 'about', '...', '7', 'Hmmm', '.', 'http', ':', '//www.djhero.com/', 'is', 'down', '8', '@', 'charviray', 'Charlene', 'my', 'love', '.', 'I', 'miss', 'you', '9', '@', 'kelcouch', 'I', "'m", 'sorry']
['0', '@', 'tiffanylu', 'i', 'know', 'i', 'wa', 'listenin', 'to', 'bad', 'habi', '...', '1', 'layin', 'n', 'bed', 'with', 'a', 'headach', 'ughhhh', '...', 'waitin', 'o', '...', '2', 'funer', 'ceremoni', '...', 'gloomi', 'friday', '...', '3', 'want', 'to', 'hang', 'out', 'with', 'friend', 'soon', '!', '4', '@', 'dannycastillo', 'we', 'want', 'to', 'trade', 'with', 'someon', 'w', '...', '5', 're-ping', '@', 'ghostridah14', ':', 'whi', 'did', "n't", 'you', 'go', 'to', '...', '6', 'i', 'should', 'be', 'sleep', ',', 'but', 'im', 'not', '!', 'think', 'about', '...', '7', 'hmmm', '.', 'http', ':', '//www.djhero.com/', 'is', 'down', '8', '@', 'charviray', 'charlen', 'my', 'love', '.', 'i', 'miss', 'you', '9', '@', 'kelcouch', 'i', "'m", 'sorri']
```