In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.decomposition import PCA
```

In [12]:
```
# load the dataset, and skip the header/first row
data = np.genfromtxt('movieReplicationSet2.csv', delimiter = ',', skip_header =
# read the csv file as a dataframe in pandas
df = pd.read_csv('movieReplicationSet2.csv')
# get a subset of the dataset for just the movie ratings
movies = df.copy()
movies = movies.iloc[:,0:400]
# get a subset of the dataset for just the personality traits
personality = df.copy()
personality = df.iloc[:,420:474]
# size of the personality data set - 1097 rows × 54 columns
personality
```

Out[12]:

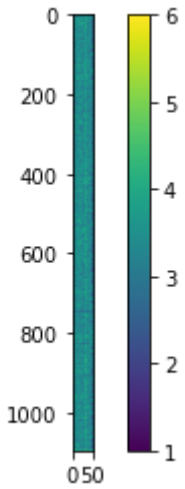
	Is talkative	Tends to find fault with others	Does a thorough job	Is depressed/Blue	Is original/comes up with new ideas	Is reserved	Is helpful and unselfish with others	Can be somewhat careless
0	1.0	2.0	NaN	4.0	4.0	5.0	2.0	3.0
1	2.0	3.0	4.0	1.0	3.0	5.0	3.0	4.0
2	4.0	2.0	4.0	2.0	3.0	3.0	4.0	4.0
3	5.0	3.0	5.0	4.0	5.0	3.0	4.0	1.0
4	4.0	4.0	4.0	4.0	2.0	3.0	4.0	4.0
...	...	...	...	...	...	...	...	...
1092	4.0	4.0	4.0	4.0	3.0	3.0	4.0	4.0
1093	5.0	5.0	5.0	2.0	5.0	4.0	5.0	5.0
1094	4.0	2.0	5.0	4.0	2.0	5.0	5.0	1.0
1095	4.0	2.0	4.0	3.0	5.0	5.0	5.0	5.0
1096	5.0	4.0	5.0	2.0	4.0	3.0	5.0	4.0

1097 rows × 54 columns

In [18]:
```
# first, fill missing values in the data with average of the corresponding colum
for i in range(54):
    personality.iloc[:,i] = personality.iloc[:,i].fillna(personality.iloc[:,i].m
```

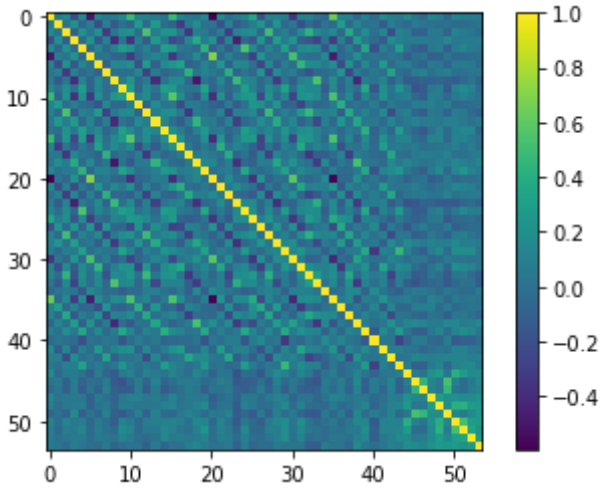
In [19]:
```
plt.imshow(personality) # Display an image, i.e. data, on a 2D regular raster.
plt.colorbar()
```

Out[19]: <matplotlib.colorbar.Colorbar at 0x7f95824f32b0>



In [45]:
```
# Compute correlation between each measure across all courses:
r = np.corrcoef(personality,rowvar=False) # 54x54
# Plot the data:
plt.imshow(r)
plt.colorbar()
```

Out[45]: (54, 54)



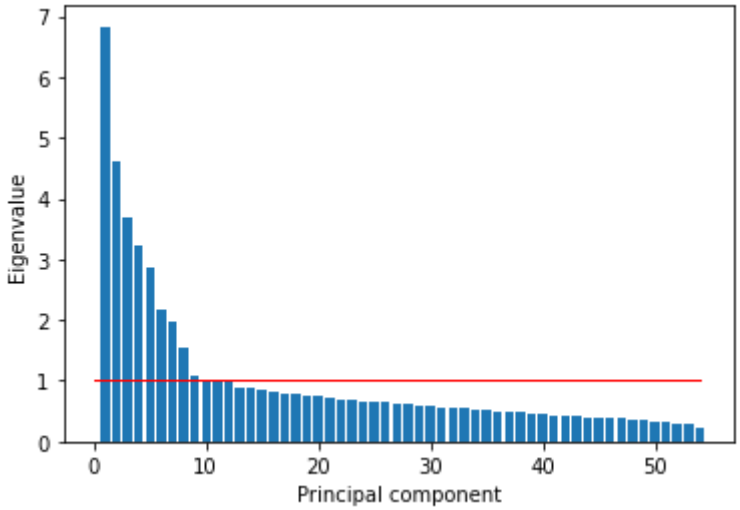
In [21]:
```
# 1. Z-score the data:
zscoredData = stats.zscore(personality)

# 2. Run the PCA:
pca = PCA().fit(zscoredData)
```

In [79]:
```
# Eigenvalues: Single vector of eigenvalues in decreasing order of magnitude
eigVals = pca.explained_variance_ # 54
# Loadings (eigenvectors): Weights per factor in terms of the original data.
loadings = pca.components_ # 54X54
# Rotated Data: Simply the transformed data
rotatedData = pca.fit_transform(zscoredData) # 1097x54
covarExplained = eigVals/sum(eigVals)*100 # 54
```

In [36]:
```
# What a scree plot is: Plotting a bar graph of the sorted Eigenvalues
numClasses = 54
plt.bar(np.linspace(1,54,54),eigVals)
plt.xlabel('Principal component')
plt.ylabel('Eigenvalue')
plt.plot([0,numClasses],[1,1],color='red',linewidth=1) # Kaiser criterion line
```

Out[36]:
```
[<matplotlib.lines.Line2D at 0x7f95825731f0>]
```

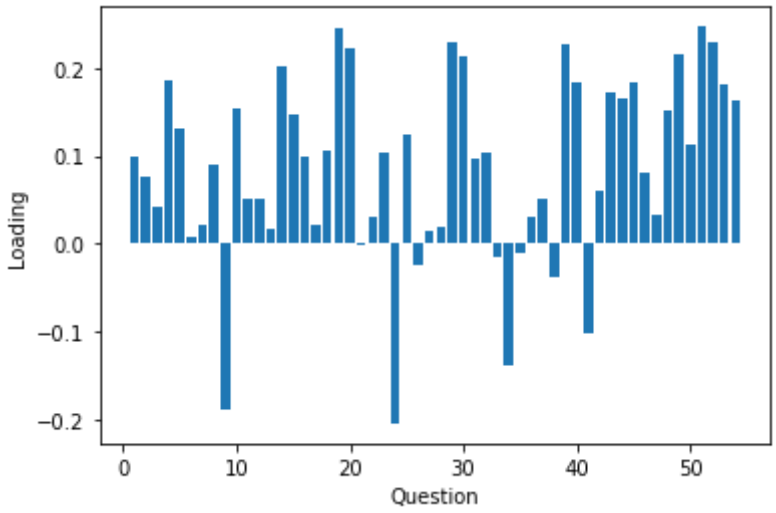


In [ ]:
```
# Kaiser criterion: Keep all factors with an eigenvalue > 1
# Rationale: Each variable adds 1 to the sum of eigenvalues. The eigensum.
# We expect each factor to explain at least as much as it adds to what needs
# to be explained. The factors have to carry their weight.

# By this criterion, we would report 9 meaningful factors.
```

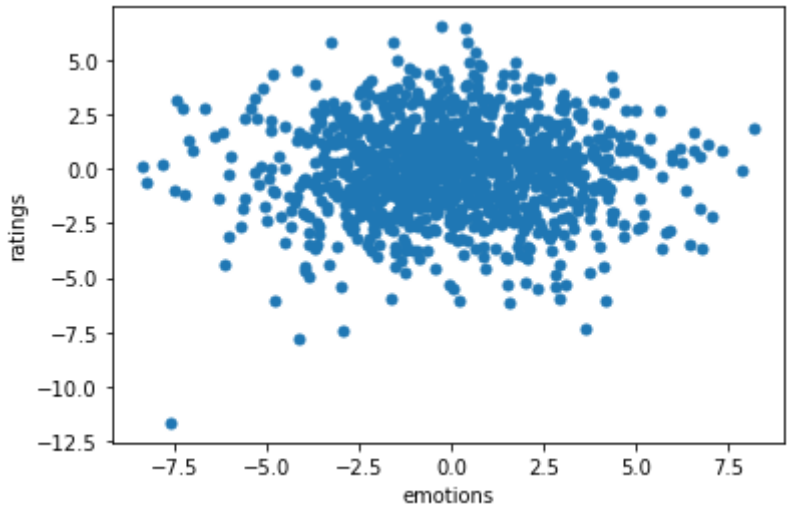
In [49]:
```
whichPrincipalComponent = 1
plt.bar(np.linspace(1,54,54),loadings[whichPrincipalComponent,:]*-1)
plt.xlabel('Question')
plt.ylabel('Loading')
# we can tell that the first factor being:
# The emotions on the screen "rub off" on me - for instance
# if something sad is happening I get sad or
# if something frightening is happening I get scared
```

Out[49]:
```
Text(0, 0.5, 'Loading')
```



In [50]:
```
plt.plot(rotatedData[:,0]*-1,rotatedData[:,1]*-1,'o',markersize=5)
plt.xlabel('emotions')
plt.ylabel('ratings')
```

Out[50]:
```
Text(0, 0.5, 'ratings')
```



In [81]:
```
"""
personality.iloc[:,50] # first factor
personality.iloc[:,18] # second factor
personality.iloc[:,51] # third factor
personality.iloc[:,28] # fourth factor
personality.iloc[:,38] # fifth factor
personality.iloc[:,19] # sixth factor
personality.iloc[:,48] # seventh factor
personality.iloc[:,29] # eighth factor
personality.iloc[:,23] # ninth factor
"""
```

Out[81]:
```
'\npersonality.iloc[:,50] # first factor\npersonality.iloc[:,18] # second factor
\npersonality.iloc[:,51] # third factor\npersonality.iloc[:,28] # fourth factor
\npersonality.iloc[:,38] # fifth factor\npersonality.iloc[:,19] # sixth factor\n
personality.iloc[:,48] # seventh factor\npersonality.iloc[:,29] # eighth factor
\npersonality.iloc[:,23] # ninth factor\n'
```

Vanessa (Ziwei) Xu
Professor Pascal Wallisch & Milan Bradonjic
Introduction to Data Science
Dec. 22nd, 2021

Project 3 Report

For the first question of this project, I've found 9 factors that I'll interpret meaningfully using the Kaiser criterion where I kept all 9 factors with an eigenvalue greater than 1. After inspecting the loadings matrix, I found that the 9 factors I found were *"The emotions on the screen "rub off" on me - for instance if something sad is happening I get sad or if something frightening is happening I get scared"*, "*Worries a lot", "When watching a movie I get completely immersed in the alternative reality of the film", "Can be moody", "Gets nervous easily", "Has an active imagination"," When watching a movie I feel like the things on the screen are happening to me", "Values artistic/aesthetic experiences", "Is emotionally stable/not easily upset".* While the first eight factors all contribute positively, the last one is negatively correlated.