In [1]:

```
from sklearn import datasets
from sklearn import preprocessing
from sklearn.decomposition import PCA
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from matplotlib import pyplot as plt
```

In [2]:

```
df = pd.read_csv('churn_clean.csv')
```

In [3]:

```
df.info
```

Out[3]:

```
<bound method DataFrame.info of       CaseOrder Customer_id
Interaction  \
0             1     K409198  aa90260b-4141-4a24-8e36-b04ce1f4f77b
1             2     S120509  fb76459f-c047-4a9d-8af9-e0f7d4ac2524
2             3     K191035  344d114c-3736-4be5-98f7-c72c281e2d35
3             4      D90850  abfa2b40-2d43-4994-b15a-989b8c79e311
4             5     K662701  68a861fd-0d20-4e51-a587-8a90407ee574
...         ...         ...                                   ...
9995       9996     M324793  45deb5a2-ae04-4518-bf0b-c82db8dbe4a4
9996       9997     D861732  6e96b921-0c09-4993-bbda-a1ac6411061a
9997       9998     I243405  e8307ddf-9a01-4fff-bc59-4742e03fd24f
9998       9999     I641617  3775ccfc-0052-4107-81ae-9657f81ecdf3
9999      10000      T38070  9de5fb6e-bd33-4995-aec8-f01d0172a499

                                   UID          City State  \
0     e885b299883d4f9fb18e39c75155d990   Point Baker    AK
1     f2de8bef964785f41a2959829830fb8a   West Branch    MI
2     f1784cfa9f6d92ae816197eb175d3c71       Yamhill    OR
3     dc8a365077241bb5cd5ccd305136b05e       Del Mar    CA
4     aabb64a116e83fdc4befc1fbab1663f9    Needville    TX
...                                ...           ...   ...
9995  9499fb4de537af195d16d046b79fd20a   Mount Holly    VT
9996  c09a841117fa81b5c8e19afec2760104   Clarksville    TN
9997  9c41f212d1e04dca84445019bbc9b41c      Mobeetie    TX
9998  3e1f269b40c235a1038863ecf6b7a0df    Carrollton    GA
9999  0ea683a03a3cd544aefe8388aab16176  Clarkesville    GA

                    County    Zip       Lat        Lng  ...  MonthlyCharge  \
0     Prince of Wales-Hyder  99927  56.25100 -133.37571  ...     172.455519
1                  Ogemaw  48661  44.32893  -84.24080  ...     242.632554
2                 Yamhill  97148  45.35589 -123.24657  ...     159.947583
3               San Diego  92014  32.96687 -117.24798  ...     119.956840
4               Fort Bend  77461  29.38012  -95.80673  ...     149.948316
...                   ...    ...       ...        ...  ...            ...
9995              Rutland   5758  43.43391  -72.78734  ...     159.979400
9996           Montgomery  37042  36.56907  -87.41694  ...     207.481100
9997              Wheeler  79061  35.52039 -100.44180  ...     169.974100
9998              Carroll  30117  33.58016  -85.13241  ...     252.624000
9999            Habersham  30523  34.70783  -83.53648  ...     217.484000

     Bandwidth_GB_Year Item1 Item2  Item3  Item4  Item5 Item6 Item7 Item8
0           904.536110     5     5      5      3      4     4     3     4
1           800.982766     3     4      3      3      4     3     4     4
2          2054.706961     4     4      2      4      4     3     3     3
3          2164.579412     4     4      4      2      5     4     3     3
4           271.493436     4     4      4      3      4     4     4     5
...                ...   ...   ...    ...    ...    ...   ...   ...   ...
9995       6511.252601     3     2      3      3      4     3     2     3
9996       5695.951810     4     5      5      4      4     5     2     5
9997       4159.305799     4     4      4      4      4     4     4     5
9998       6468.456752     4     4      6      4      3     3     5     4
9999       5857.586167     2     2      3      3      3     3     4     1

[10000 rows x 50 columns]>
```

In [4]:

```
#df.rename dfr = dfr.rename(columns={'Item1': 'Timely response'})
```

In [5]:

```
dfpca = df[['Outage_sec_perweek','Tenure', 'MonthlyCharge', 'Bandwidth_GB_Year', 'Item
dfpcanormalized=(dfpca-dfpca.mean())/dfpca.std()
```

```
pca = PCA(n_components=dfpca.shape[1])
pca.fit(dfpcanormalized)
PCA(n_components=12)
dfpca2 = pd.DataFrame(pca.transform(dfpcanormalized),columns=['PC1','PC2','PC3','PC4',
```

In [6]:

```
pd.DataFrame(dfpcanormalized).to_csv("churn_clean_normalized.csv")
```

In [7]:

```
loadings=pd.DataFrame(pca.components_.T,
columns=['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9','PC10','PC11', 'PC12'],
index=dfpcanormalized.columns)
loadings
```

Out[7]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	
Outage_sec_perweek	-0.017428	0.004010	-0.014271	0.707528	-0.700306	-0.085860	0.017491	-0.009
Tenure	-0.016148	0.702882	-0.061382	-0.044803	-0.040867	0.007139	-0.004091	0.006
MonthlyCharge	0.000901	0.040213	-0.008127	0.700704	0.710047	-0.006815	0.014660	-0.015
Bandwidth_GB_Year	-0.016646	0.704158	-0.061496	0.000301	0.004703	0.009129	-0.002994	0.005
Item1	0.458823	0.030630	0.281317	0.011999	0.010224	-0.070293	-0.119133	-0.045
Item2	0.433927	0.038135	0.282943	0.020531	-0.010460	-0.110173	-0.168688	-0.066
Item3	0.400496	0.035273	0.281071	-0.015650	-0.007066	-0.174392	-0.254671	-0.147
Item4	0.145814	-0.038722	-0.568735	-0.020574	0.012388	-0.170738	-0.483552	-0.442
Item5	-0.175661	0.055808	0.588414	-0.003980	-0.004810	0.137971	0.058711	-0.208
Item6	0.405127	-0.006603	-0.183885	0.000066	0.005222	-0.061882	0.064847	0.758
Item7	0.358286	0.002014	-0.181958	-0.038178	0.005552	-0.177723	0.806397	-0.379
Item8	0.308760	-0.013666	-0.131801	0.060848	-0.056694	0.928091	-0.014057	-0.112

In [8]:

```
print(loadings)
```

```
                         PC1       PC2       PC3       PC4       PC5  \
Outage_sec_perweek -0.017428  0.004010 -0.014271  0.707528 -0.700306
Tenure             -0.016148  0.702882 -0.061382 -0.044803 -0.040867
MonthlyCharge       0.000901  0.040213 -0.008127  0.700704  0.710047
Bandwidth_GB_Year  -0.016646  0.704158 -0.061496  0.000301  0.004703
Item1               0.458823  0.030630  0.281317  0.011999  0.010224
Item2               0.433927  0.038135  0.282943  0.020531 -0.010460
Item3               0.400496  0.035273  0.281071 -0.015650 -0.007066
Item4               0.145814 -0.038722 -0.568735 -0.020574  0.012388
Item5              -0.175661  0.055808  0.588414 -0.003980 -0.004810
Item6               0.405127 -0.006603 -0.183885  0.000066  0.005222
Item7               0.358286  0.002014 -0.181958 -0.038178  0.005552
Item8               0.308760 -0.013666 -0.131801  0.060848 -0.056694

                         PC6       PC7       PC8       PC9      PC10  \
Outage_sec_perweek -0.085860  0.017491 -0.009299  0.012963  0.018676
Tenure              0.007139 -0.004091  0.006191 -0.007068 -0.003703
MonthlyCharge      -0.006815  0.014660 -0.015930 -0.000782  0.021294
Bandwidth_GB_Year   0.009129 -0.002994  0.005920 -0.006864 -0.002301
Item1              -0.070293 -0.119133 -0.045767  0.026016 -0.241054
Item2              -0.110173 -0.168688 -0.066098  0.076924 -0.590979
Item3              -0.174392 -0.254671 -0.147274 -0.398166  0.672440
Item4              -0.170738 -0.483552 -0.442260  0.431741  0.088821
Item5               0.137971  0.058711 -0.208105  0.693275  0.267613
Item6              -0.061882  0.064847  0.758796  0.400822  0.231202
Item7              -0.177723  0.806397 -0.379584  0.069242  0.068276
Item8               0.928091 -0.014057 -0.112587 -0.046051  0.044924

                        PC11      PC12
Outage_sec_perweek  0.013044  0.000114
Tenure              0.006657 -0.705717
MonthlyCharge      -0.011640 -0.045372
Bandwidth_GB_Year   0.002403  0.707032
Item1               0.793182  0.002251
Item2              -0.574045 -0.001056
Item3              -0.177177  0.000132
Item4               0.018565  0.000734
Item5              -0.041482 -0.000553
Item6              -0.063788 -0.000334
Item7              -0.040348  0.000502
Item8              -0.042713 -0.001487
```

In [9]:

```
print("Variance explained by all PC =", sum (pca.explained_variance_ratio_ * 100))
```

```
Variance explained by all PC = 100.0
```

In [10]:

```
varex = pca.explained_variance_ratio_ * 100
vardf = pd.DataFrame(varex.round(2), columns= ["Variance per PC"], index = ['PC1','PC2
vardf
```

Out[10]:

	Variance per PC
PC1	24.57
PC2	16.63
PC3	13.62
PC4	8.52
PC5	8.17
PC6	6.51
PC7	5.77
PC8	4.94
PC9	4.49
PC10	4.02
PC11	2.71
PC12	0.05

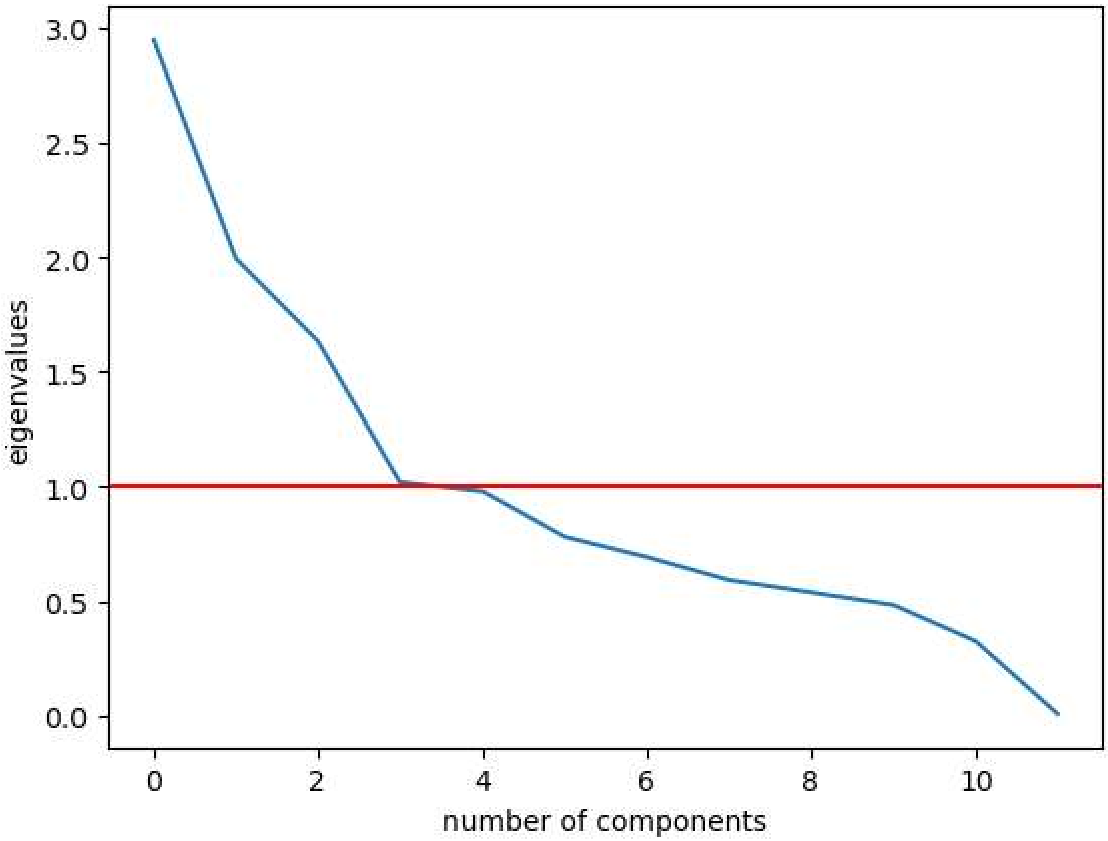
In [11]:

```
print("Variance sum of PC1, PC2 and PC3: ", np.cumsum(pca.explained_variance_ratio_ *
```

```
Variance sum of PC1, PC2 and PC3:  54.82299383765756
```

In [12]:

```
cov_matrix = np.dot(dfpcanormalized.T, dfpcanormalized) / dfpca.shape[0]
eigenvalues = [np.dot(eigenvector.T, np.dot(cov_matrix, eigenvector)) for eigenvector
plt.plot(eigenvalues)
plt.xlabel('number of components')
plt.ylabel('eigenvalues')
plt.axhline(y=1, color='red')
plt.show()
```


3.0
2.5
2.0
1.5
1.0
0.5
0.0
eigenvalues
0
2
4
6
8
10
number of components

```
In [ ]:
```