A

1)

```
Can we simplify the churn data set using principal component analysis and retain the components that
capture the most variance for future use in machine learning models and data visualization?
```

2)

```
One goal of the data analysis is to reduce the 'Churn' data set into principal components and
determine the optimal number of components for future use in statistical analysis.
```

B

1)

```
 PCA analyzes the the data set by creating new components out of a linear combination of the initial
variables. They are constructed such that the components are not correlated with each other and the
amount of variance within each component is greatest in the first component and decreases with each
following component. An expected outcome is that after primary component analysis is complete, we
should have a simplified data set with reduced dimensionality created from the initial data set. The
analysis will also provide information about the explained variance ratio which will detail how much
variance is contained in each principal component.
```

2)

```
 One assumption of primary component analysis is that large variance indicates importance. It is
assumed that principal components with the highest variance are the most significant.
```

C

1)

```
 The continuous variables being used are:

 Lat
 Lng
 Population
```

```
Children
Age
Income
Outage_sec_perweek
Email
Contacts
Yearly_equip_failure
Tenure
MonthlyCharge
Bandwidth_GB_Year
Item1
Item2
Item3
Item4
Item5
Item6
Item7
Item8
```

2)

```
Standardize the data.
```

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
# Assuming your CSV file is named 'data.csv', adjust the file path as needed
file_path = '/home/dj/skewl/D212/2/churn_clean.csv'
pd.set_option('display.max_columns', None)
# Read the data from the CSV file into a DataFrame
df = pd.read_csv(file_path)
#drop index column
df = df.loc[:, ~df.columns.str.contains('Unnamed')]
# get numeric columns
data = df.select_dtypes(include='number')
# remove zip and CaseOrder columns because it is categorical
del data['Zip']
del data['CaseOrder']
df = data
#standardize the data
df=(df-df.mean())/df.std()
#write to csv file.
df.to_csv('standardized-data.csv', index=False)
```

D

1)

    Matrix of principal components:

In [2]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components=21)

#fit pca model to our data
pca.fit(df)
#transform data set to 22 PCA components
data_pca = pd.DataFrame(pca.transform(df),
     columns=['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9',
             'PC10','PC11','PC12','PC13','PC14','PC15','PC16','PC17'
              ,'PC18','PC19','PC20','PC21'])

loadings = pd.DataFrame(pca.components_.T,
     columns=['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9',
             'PC10','PC11','PC12','PC13','PC14','PC15','PC16','PC17'
              ,'PC18','PC19','PC20','PC21'],
     index=df.columns)
print(loadings.head(21))
```

```
                          PC1       PC2       PC3       PC4       PC5  \
Lat                 -0.001112 -0.023121 -0.007380 -0.713407 -0.025042
Lng                  0.008058  0.009447  0.022445  0.177972 -0.338392
Population          -0.002181 -0.000771  0.015616  0.652679  0.173323
Children             0.004128  0.015957  0.028784 -0.016885  0.413388
Age                  0.006509  0.000521 -0.028836  0.055294 -0.426834
Income               0.001022  0.005808  0.025622 -0.055938  0.186964
Outage_sec_perweek  -0.017494  0.003909 -0.014166  0.013937 -0.259856
Email                0.008792 -0.019741 -0.002773  0.149799 -0.088409
Contacts            -0.008725  0.003459 -0.011524  0.029306 -0.438742
Yearly_equip_failure -0.007705  0.017671  0.008043 -0.007244  0.150265
Tenure              -0.016266  0.702098 -0.063693 -0.007696  0.009770
MonthlyCharge        0.000980  0.039884 -0.009138 -0.002964 -0.416994
Bandwidth_GB_Year   -0.016790  0.703617 -0.062724 -0.009177  0.009116
Item1                0.458719  0.031335  0.280924 -0.011199 -0.017378
Item2                0.433834  0.038617  0.281971 -0.018981 -0.020335
Item3                0.400518  0.035598  0.280415 -0.003381  0.000304
Item4                0.145752 -0.039814 -0.568295 -0.005339  0.009238
Item5               -0.175652  0.056530  0.586829 -0.008554 -0.028968
Item6                0.405012 -0.006736 -0.183775  0.012565  0.012014
Item7                0.358211  0.001737 -0.181488 -0.020250  0.019927
Item8                0.308716 -0.013350 -0.131543  0.045283 -0.011427

                          PC6       PC7       PC8       PC9      PC10  \
Lat                  0.112069 -0.098595 -0.028808 -0.010332 -0.022292
Lng                 -0.710967  0.354154 -0.092208 -0.064324 -0.066207
Population           0.307612 -0.122630  0.097508  0.054599  0.067829
Children            -0.493891 -0.097139  0.136314  0.066512 -0.076010
Age                  0.263319  0.423545 -0.075478 -0.178439  0.096758
Income              -0.035440  0.324461  0.092339  0.779760  0.332631
Outage_sec_perweek  -0.115988 -0.457488  0.584093  0.090340 -0.210243
Email               -0.146479 -0.345697 -0.426345  0.036436 -0.135628
Contacts             0.141564  0.020315  0.020926  0.515860 -0.525189
Yearly_equip_failure  0.052174  0.415508  0.581382 -0.254131 -0.248981
Tenure               0.025127  0.009253 -0.036361 -0.004253 -0.035038
MonthlyCharge       -0.107632 -0.228324  0.280072 -0.020344  0.679170
Bandwidth_GB_Year   -0.004363 -0.021363 -0.011151  0.003876  0.003704
Item1               -0.002033 -0.002239  0.015228 -0.022008 -0.010938
Item2                0.018206 -0.016517  0.014141  0.000544 -0.009914
Item3                0.003300 -0.012957 -0.026180 -0.035907 -0.011518
Item4               -0.013591  0.005827 -0.012544 -0.028655 -0.010613
Item5                0.042602  0.003137 -0.014066 -0.002505 -0.003013
Item6                0.015886 -0.004968  0.007999  0.019149 -0.003231
Item7               -0.006088  0.025328 -0.026909  0.069894 -0.012482
Item8                0.016670 -0.004744  0.069500 -0.000909  0.034239

                         PC11      PC12      PC13      PC14      PC15  \
Lat                  0.087520 -0.010790  0.057719  0.095224  0.660205
```

```
Lng                 -0.173572 -0.094972 -0.158149  0.071398  0.360598
Population          -0.025682  0.027218  0.108331  0.167885  0.606033
Children             0.187104  0.176812  0.690935 -0.019480 -0.004987
Age                  0.345449 -0.323264  0.538841  0.035972 -0.043814
Income               0.205332 -0.238138 -0.146506  0.024111  0.018182
Outage_sec_perweek   0.034554 -0.551538 -0.004926  0.081577 -0.049628
Email                0.751640  0.005453 -0.236985 -0.057343  0.041014
Contacts            -0.084467  0.454008  0.160663 -0.045656  0.000769
Yearly_equip_failure  0.420133  0.266144 -0.294977 -0.013081  0.039989
Tenure               0.000451 -0.038848 -0.008380 -0.004235  0.011882
MonthlyCharge        0.111057  0.452563  0.013219  0.004390 -0.009063
Bandwidth_GB_Year    0.002315  0.006581 -0.003180 -0.008830  0.011760
Item1               -0.004500  0.024850 -0.007659  0.071972  0.021536
Item2               -0.002179 -0.000882  0.018278  0.109222 -0.006481
Item3               -0.004230 -0.007590 -0.020047  0.175058 -0.005387
Item4               -0.021718  0.020818 -0.010999  0.180290  0.061364
Item5               -0.007609 -0.013871  0.002466 -0.136959  0.015125
Item6                0.021769  0.017593 -0.001535  0.053518 -0.061373
Item7                0.013871  0.014773 -0.010484  0.159747 -0.124997
Item8               -0.040845 -0.090967  0.020455 -0.903150  0.185797

                         PC16      PC17      PC18      PC19      PC20  \
Lat                  0.087845 -0.044067 -0.005204  0.015805 -0.011682
Lng                  0.059220 -0.038542  0.017837  0.000416 -0.025267
Population           0.090310 -0.012212  0.000593  0.001053 -0.007964
Children            -0.013577  0.015098  0.013894  0.020949 -0.000465
Age                 -0.002093  0.004171 -0.009878  0.005712  0.014211
Income              -0.077328  0.007595 -0.002393  0.005199  0.013404
Outage_sec_perweek   0.012222  0.010283  0.013432  0.017977  0.013847
Email               -0.012751  0.014797  0.005772 -0.016556  0.000869
Contacts            -0.035995  0.004012 -0.026819  0.020297 -0.000501
Yearly_equip_failure  0.010939  0.014188 -0.001251  0.007763 -0.021791
Tenure              -0.002089 -0.007220 -0.007826 -0.004391  0.007360
MonthlyCharge        0.013084  0.017403 -0.000506  0.021466 -0.011578
Bandwidth_GB_Year   -0.002077 -0.006102 -0.006224 -0.001992  0.001790
Item1               -0.113274  0.044657  0.025446 -0.240334  0.792983
Item2               -0.171007  0.068403  0.072172 -0.591234 -0.572810
Item3               -0.249520  0.149958 -0.395794  0.673666 -0.176095
Item4               -0.472789  0.445426  0.430805  0.087188  0.019061
Item5                0.059286  0.208307  0.693579  0.263929 -0.042083
Item6                0.050732 -0.756383  0.402499  0.229705 -0.065203
Item7                0.799107  0.374344  0.070906  0.066331 -0.041194
Item8               -0.004547  0.109457 -0.046218  0.046139 -0.043523

                         PC21
Lat                  0.001011
Lng                  0.000711
Population          -0.000064
```

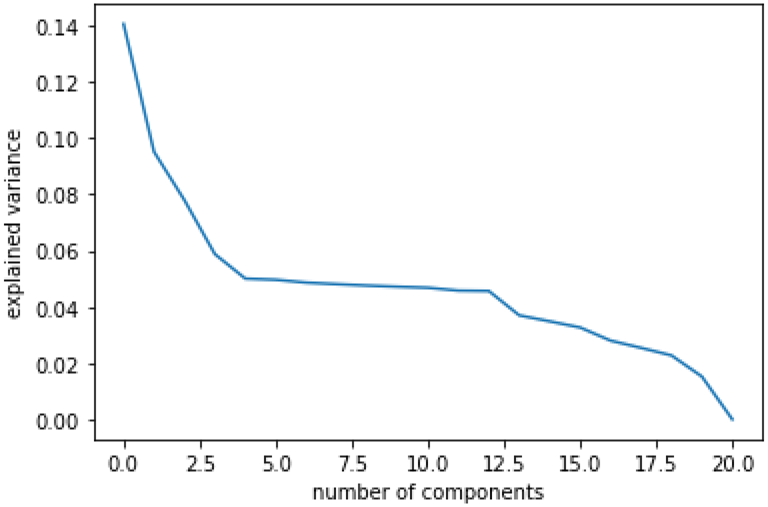
```
Children              -0.021623
Age                    0.022412
Income                -0.000913
Outage_sec_perweek     0.000350
Email                  0.000247
Contacts              -0.000953
Yearly_equip_failure -0.000131
Tenure                -0.705243
MonthlyCharge         -0.045786
Bandwidth_GB_Year      0.706787
Item1                  0.002931
Item2                 -0.001136
Item3                  0.000078
Item4                  0.000089
Item5                 -0.000809
Item6                 -0.000564
Item7                  0.000481
Item8                 -0.001970
```

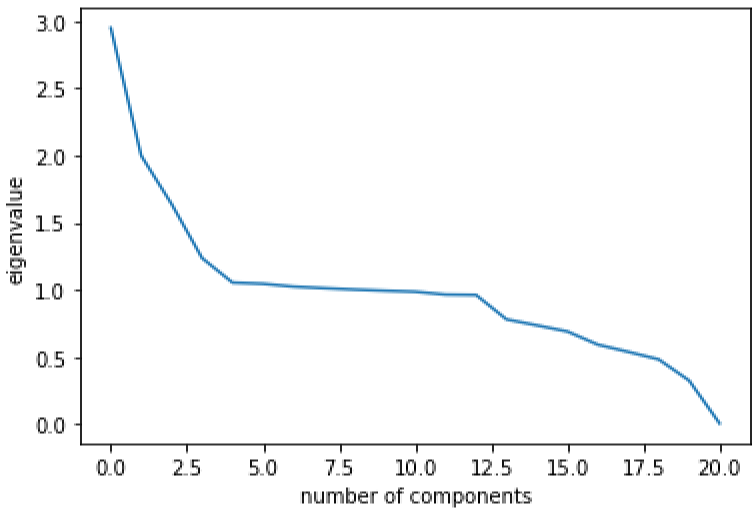
2)

```
Total number of principal components:

Using the plot of explained variance below and the elbow rule we can determine that the total number
of principal components is 4.
```

In [3]:

```
plt.plot(pca.explained_variance_ratio_)
plt.xlabel('number of components')
plt.ylabel('explained variance')
plt.show()

cov_matrix = np.dot(df.T, df) / df.shape[0]
eigenvalues = [np.dot(eigenvector.T, np.dot(cov_matrix, eigenvector)) for
eigenvector in pca.components_]

plt.plot(eigenvalues)
plt.xlabel('number of components')
plt.ylabel('eigenvalue')
plt.show()
```





3)

```
Variance of each principal component:

PC1 0.13414451

PC2 0.12586588

PC3 0.07448371

PC4 0.0561557
```

In [4]:
```
print(pca.explained_variance_ratio_[:4])
```

[0.14041402 0.09511751 0.07794643 0.05882728]

4)

Total variance of the first four principal components is 0.39064979323049326.

In [5]:
```
total_variance_first_four = pca.explained_variance_ratio_[:4].sum()
print(total_variance_first_four)
```

0.3723052340938371

5)

The results of the PCA data analysis show that the 'Churn' data set can be dimensionally reduced to an optimum number of four principal components. The optimum number of components was determined using the elbow method and a scree plot of explained variance. The principal component matrix was also determined. This provides data to help understand how each principal component is loaded by the original variables in the 'Churn' data set.

In [ ]: