Healthy Lifestyle

In [68]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [69]:
```
data= pd.read_excel("C:/Users/darshini/OneDrive/Documents/darshu/Heathy life
data
```

Out[69]:

	Age	Gender	Commitment to healthy lifestyle	Vegetables	Fruits	Millets/ Grains	Junk Foods	Caffeinated beverages	Alcohol	Ph Ex
0	20-30	2	2	1	2	1	2	3	4	
1	20-30	2	2	1	3	1	3	4	4	
2	20-30	1	1	1	3	1	1	2	4	
3	20-30	2	2	1	1	1	3	1	4	
4	20-30	1	2	2	2	1	3	1	3	
...	...	...	...	...	...	...	...	...	...	
386	30-40	1	2	1	2	1	3	1	2	
387	30-40	1	2	1	3	1	3	1	3	
388	30-40	1	2	1	1	1	4	1	4	
389	20-30	2	3	1	1	1	3	2	4	
390	20-30	1	2	1	2	1	2	1	4	

391 rows × 19 columns

In [70]:
```python
print(data.shape)
print(data.info())
```

```
(391, 19)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 391 entries, 0 to 390
Data columns (total 19 columns):
 #   Column                           Non-Null Count  Dtype 
---  ------                           --------------  ----- 
 0   Age                              391 non-null    object
 1   Gender                           391 non-null    int64 
 2   Commitment to healthy lifestyle  391 non-null    int64 
 3   Vegetables                       391 non-null    int64 
 4   Fruits                           391 non-null    int64 
 5   Millets/ Grains                  391 non-null    int64 
 6   Junk Foods                       391 non-null    int64 
 7   Caffeinated beverages            391 non-null    int64 
 8   Alcohol                          391 non-null    int64 
 9   Physical Exercise                391 non-null    int64 
 10  Yoga/ Meditation                 391 non-null    int64 
 11  Sleep hours                      391 non-null    int64 
 12  Work hours                       391 non-null    int64 
 13  stress level                     391 non-null    int64 
 14  Illness                          391 non-null    int64 
 15  Screen time                      391 non-null    int64 
 16  Self grooming/ Hobbies           391 non-null    int64 
 17  Socialize                        391 non-null    int64 
 18  Connection with nature           391 non-null    int64 
dtypes: int64(18), object(1)
memory usage: 58.2+ KB
None
```

The independent variables are-

1. Age
2. Vegetables
3. Fruits
4. Millets/ Grains
5. Junk Foods
6. Caffeinated beverages
7. Alcohol
8. Physical Excercise
9. Yoga/Meditation
10. Sleep hours
11. Work hours
12. stress level
13. Illness
14. Screen time
15. Self grooming/Hobbies
16. Socialize
17. Connection with nature

In [71]:
```python
print(data.describe())
```

```
           Gender  Commitment to healthy lifestyle  Vegetables      Fruits  \
count  391.000000                       391.000000  391.000000  391.000000   
mean     1.580563                         2.207161    1.235294    1.713555   
std      0.494099                         0.709315    0.522207    0.705143   
min      1.000000                         1.000000    1.000000    1.000000   
25%      1.000000                         2.000000    1.000000    1.000000   
50%      2.000000                         2.000000    1.000000    2.000000   
75%      2.000000                         3.000000    1.000000    2.000000   
max      2.000000                         4.000000    4.000000    4.000000   

       Millets/ Grains  Junk Foods  Caffeinated beverages     Alcohol  \
count       391.000000  391.000000             391.000000  391.000000   
mean          1.304348    2.324808               1.805627    3.585678   
std           0.596534    0.726335               0.993863    0.646251   
min           1.000000    1.000000               1.000000    1.000000   
25%           1.000000    2.000000               1.000000    3.000000   
50%           1.000000    2.000000               1.000000    4.000000   
75%           1.000000    3.000000               3.000000    4.000000   
max           4.000000    4.000000               4.000000    4.000000   

       Physical Exercise  Yoga/ Meditation  Sleep hours  Work hours  \
count         391.000000        391.000000   391.000000  391.000000   
mean            1.774936          2.268542     2.358056    2.186701   
std             1.256975          1.456025     0.797226    0.910580   
min             1.000000          1.000000     1.000000    1.000000   
25%             1.000000          1.000000     2.000000    2.000000   
50%             1.000000          1.000000     2.000000    2.000000   
75%             2.000000          4.000000     3.000000    3.000000   
max             4.000000          4.000000     4.000000    4.000000   

       stress level     Illness  Screen time  Self grooming/ Hobbies  \
count    391.000000  391.000000   391.000000              391.000000   
mean       2.058824    2.478261     2.516624                1.820972   
std        0.682467    0.693496     1.097365                1.147425   
min        1.000000    1.000000     1.000000                1.000000   
25%        2.000000    2.000000     2.000000                1.000000   
50%        2.000000    3.000000     2.000000                1.000000   
75%        2.000000    3.000000     4.000000                2.000000   
max        4.000000    4.000000     4.000000                4.000000   

        Socialize  Connection with nature  
count  391.000000              391.000000  
mean     2.171355                1.849105  
std      1.115760                0.716289  
min      1.000000                1.000000  
25%      1.000000                1.000000  
50%      2.000000                2.000000  
75%      3.000000                2.000000  
max      4.000000                4.000000  
```

Data Cleaning

In [72]:
```
#Missing Values
print(data.isnull().sum())
```

```
Age                                0
Gender                             0
Commitment to healthy lifestyle    0
Vegetables                         0
Fruits                             0
Millets/ Grains                    0
Junk Foods                         0
Caffeinated beverages              0
Alcohol                            0
Physical Exercise                  0
Yoga/ Meditation                   0
Sleep hours                        0
Work hours                         0
stress level                       0
Illness                            0
Screen time                        0
Self grooming/ Hobbies             0
Socialize                          0
Connection with nature             0
dtype: int64
```

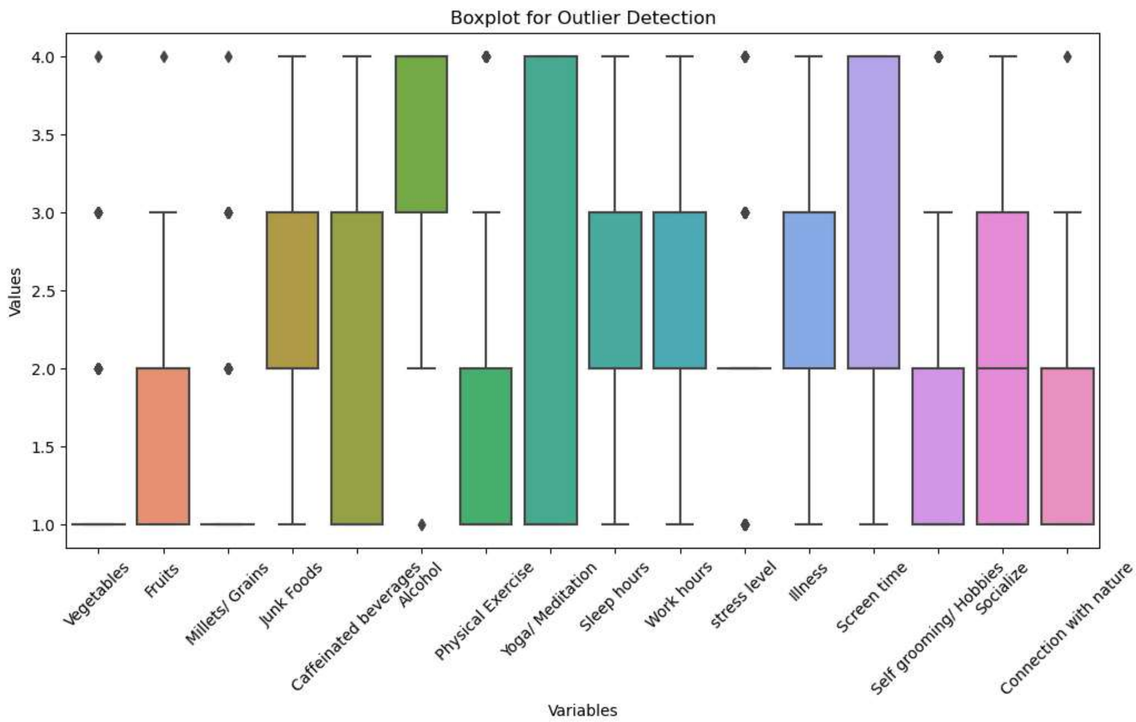
In [73]:

```
# List of columns to remove
columns_to_remove = ['Age','Gender','Commitment to healthy lifestyle']

# Drop the columns from the dataset
data.drop(columns=columns_to_remove, inplace=True)

# Print the modified dataset
print(data)
```

```
     Vegetables  Fruits  Millets/ Grains  Junk Foods  Caffeinated beverage
s  \
0             1       2                1           2
3
1             1       3                1           3
4
2             1       3                1           1
2
3             1       1                1           3
1
4             2       2                1           3
1
..          ...     ...              ...         ...
...
386           1       2                1           3
1
387           1       3                1           3
1
388           1       1                1           4
1
389           1       1                1           3
2
390           1       2                1           2
1

     Alcohol  Physical Exercise  Yoga/ Meditation  Sleep hours  Work hours
\
0          4                  1                 1            3           3
1          4                  1                 1            2           1
2          4                  1                 1            3           1
3          4                  1                 1            2           2
4          3                  1                 1            3           1
..       ...                ...               ...          ...         ...
386        2                  4                 4            2           4
387        3                  4                 4            2           2
388        4                  4                 4            3           4
389        4                  4                 2            2           2
390        4                  1                 4            2           3

     stress level  Illness  Screen time  Self grooming/ Hobbies  Socialize
\
0               2        3            2                       2          3
1               3        3            1                       1          4
2               2        3            3                       2          2
3               2        3            1                       1          3
4               2        3            2                       1          1
..            ...      ...          ...                     ...        ...
386             1        3            4                       4          1
387             2        2            4                       4          4
388             2        3            4                       1          4
389             3        2            2                       1          4
390             2        2            4                       1          1

     Connection with nature
0                         1
1                         3
2                         1
3                         3
4                         1
..                      ...
386                       2
```

```
387                        2
388                        2
389                        1
390                        2

[391 rows x 16 columns]
```

In [74]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(data=data)
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.title('Boxplot for Outlier Detection')
plt.xlabel('Variables')
plt.ylabel('Values')
plt.show()
```



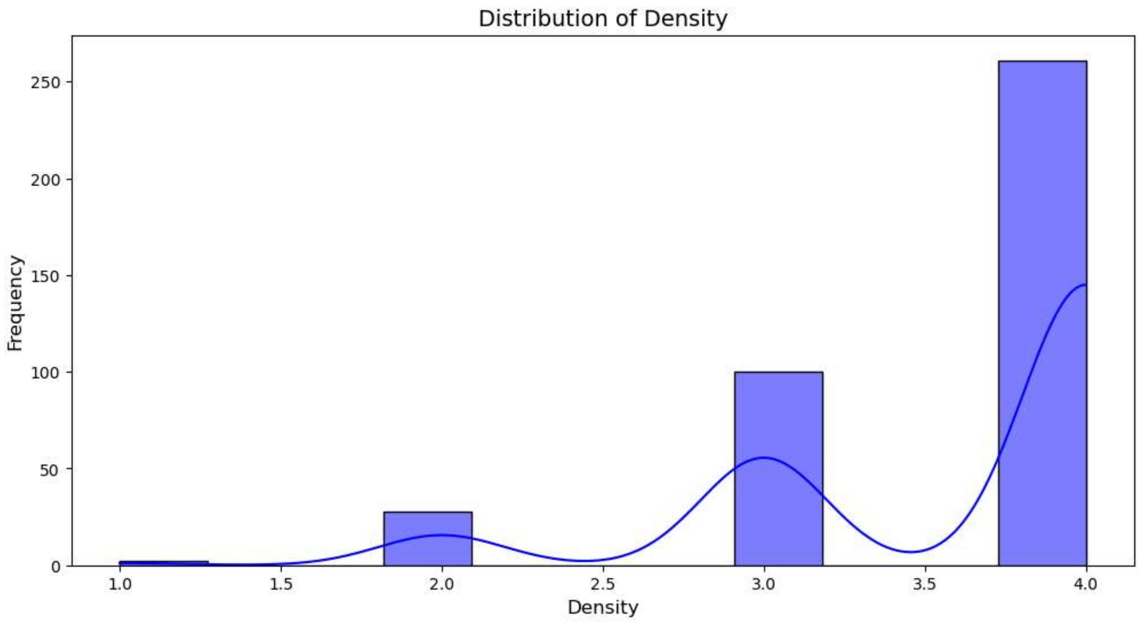
In [75]:
```
from sklearn.preprocessing import StandardScaler
# Initialize StandardScaler

scaler = StandardScaler()

# Fit and transform the ordinal data
ordinal_data_standardized = scaler.fit_transform(data)

# Convert the standardized data back to a DataFrame
ordinal_data_standardized = pd.DataFrame(ordinal_data_standardized, columns=

# Print the standardized ordinal data
print(ordinal_data_standardized)
```

```
     Vegetables    Fruits  Millets/ Grains  Junk Foods  Caffeinated bevera
ges  \
0     -0.451154  0.406743        -0.510847   -0.447761                 1.203
289
1     -0.451154  1.826713        -0.510847    0.930778                 2.210
753
2     -0.451154  1.826713        -0.510847   -1.826300                 0.195
824
3     -0.451154 -1.013227        -0.510847    0.930778                -0.811
640
4      1.466250  0.406743        -0.510847    0.930778                -0.811
640
..          ...       ...              ...         ...
...
386   -0.451154  0.406743        -0.510847    0.930778                -0.811
640
387   -0.451154  1.826713        -0.510847    0.930778                -0.811
640
388   -0.451154 -1.013227        -0.510847    2.309318                -0.811
640
389   -0.451154 -1.013227        -0.510847    0.930778                 0.195
824
390   -0.451154  0.406743        -0.510847   -0.447761                -0.811
640

      Alcohol  Physical Exercise  Yoga/ Meditation  Sleep hours  Work hour
s  \
0    0.641938          -0.617299         -0.872353     0.806253    0.89431
0
1    0.641938          -0.617299         -0.872353    -0.449703   -1.30490
6
2    0.641938          -0.617299         -0.872353     0.806253   -1.30490
6
3    0.641938          -0.617299         -0.872353    -0.449703   -0.20529
8
4   -0.907431          -0.617299         -0.872353     0.806253   -1.30490
6
..        ...                ...               ...          ...
...
386 -2.456799           1.772442          1.190691    -0.449703    1.99391
9
387 -0.907431           1.772442          1.190691    -0.449703   -0.20529
8
388  0.641938           1.772442          1.190691     0.806253    1.99391
9
389  0.641938           1.772442         -0.184672    -0.449703   -0.20529
8
390  0.641938          -0.617299          1.190691    -0.449703    0.89431
0

     stress level   Illness  Screen time  Self grooming/ Hobbies  Socializ
e  \
0       -0.086303  0.753296    -0.471389                0.156226   0.74362
4
1        1.380846  0.753296    -1.383830               -0.716407   1.64102
2
2       -0.086303  0.753296     0.441052                0.156226  -0.15377
4
3       -0.086303  0.753296    -1.383830               -0.716407   0.74362
4
4       -0.086303  0.753296    -0.471389               -0.716407  -1.05117
```

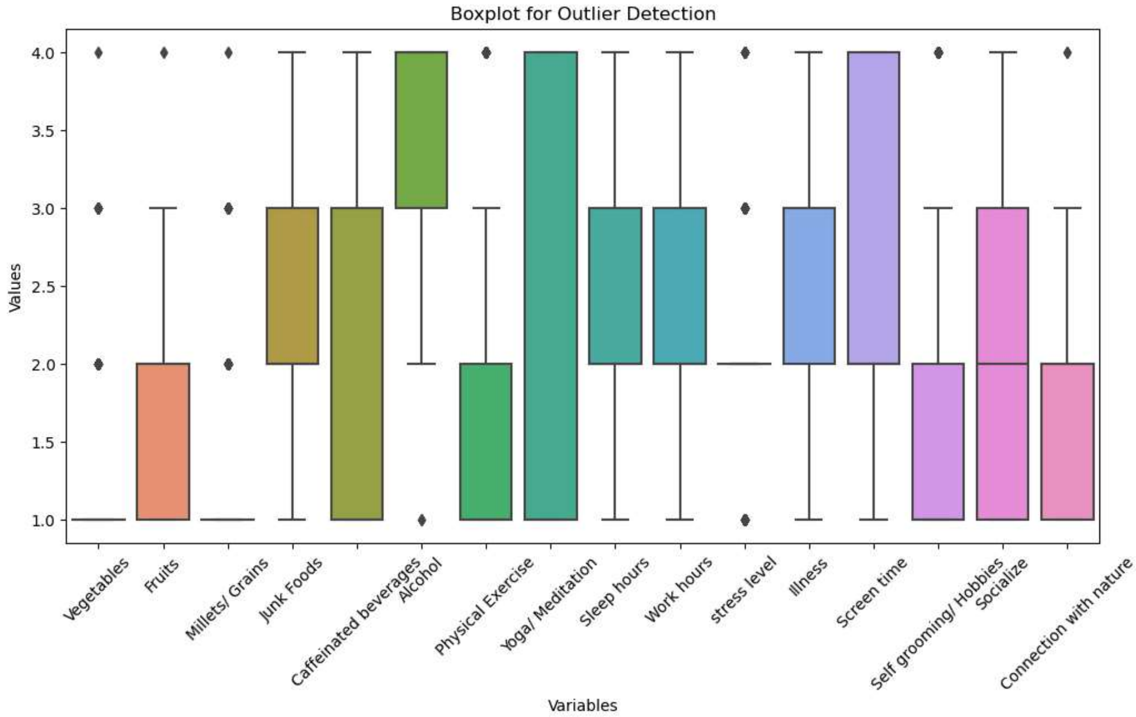
```
2
..          ...        ...          ...                       ...
...
386     -1.553452  0.753296     1.353494                  1.901492  -1.05117
2
387     -0.086303 -0.690521     1.353494                  1.901492   1.64102
2
388     -0.086303  0.753296     1.353494                 -0.716407   1.64102
2
389      1.380846 -0.690521    -0.471389                 -0.716407   1.64102
2
390     -0.086303 -0.690521     1.353494                 -0.716407  -1.05117
2

     Connection with nature
0                 -1.186941
1                  1.608806
2                 -1.186941
3                  1.608806
4                 -1.186941
..                      ...
386                0.210932
387                0.210932
388                0.210932
389               -1.186941
390                0.210932

[391 rows x 16 columns]
```

In [76]:

```python
plt.figure(figsize=(12, 6))
sns.histplot(data=data, x='Alcohol', kde=True, color='blue')
plt.xlabel('Density', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Distribution of Density', fontsize=14)
plt.show()
```



In [77]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(data=data)
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.title('Boxplot for Outlier Detection')
plt.xlabel('Variables')
plt.ylabel('Values')
plt.show()
```

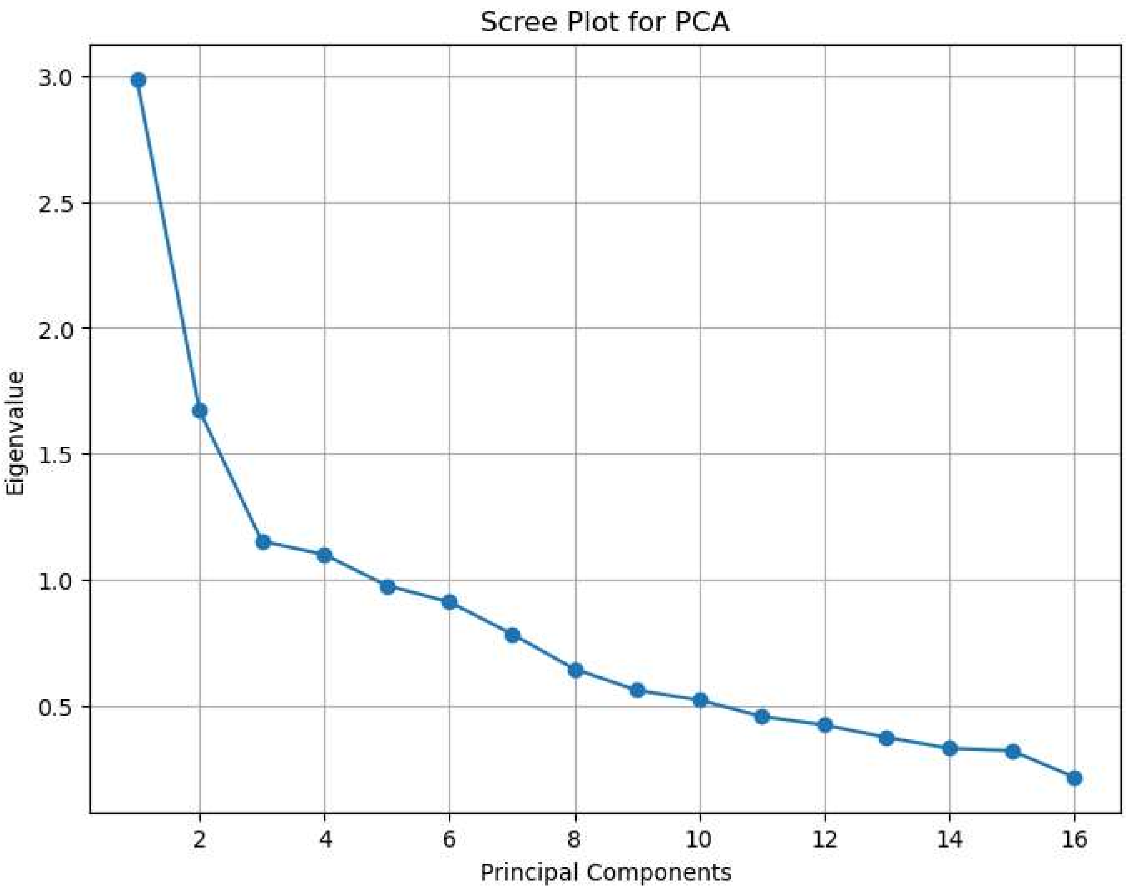


In [79]:
```
from sklearn.decomposition import PCA
import numpy as np
```

In [80]:
```
# Calculate similarity matrix (e.g., using rank correlation)
# Here, Spearman's rank correlation is used as an example
similarity_matrix = np.corrcoef(data, rowvar=False)

# Compute eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(similarity_matrix)

# Print eigenvalues
print("Eigenvalues:", eigenvalues)
```

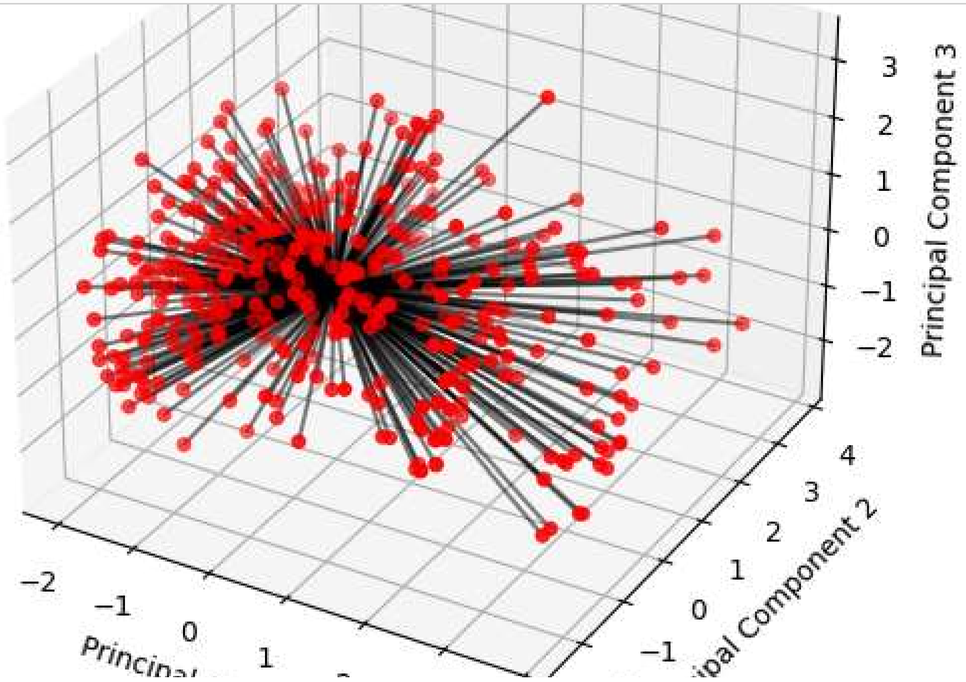
```
Eigenvalues: [2.04440097 1.52582626 1.37706209 1.29342654 0.48752547 1.150
91784
 0.6093436  1.03233519 0.67029317 0.70895603 0.72929996 0.9777568
 0.79882799 0.81491896 0.90609291 0.8730162 ]
```

In [91]:
```
# Standardize the data
scaler = StandardScaler()
ordinal_data_standardized = scaler.fit_transform(data)

# Perform PCA
pca = PCA()
pc=pca.fit(data)

# Get the eigenvalues
eigenvalues = pca.explained_variance_

# Plot the scree plot
plt.figure(figsize=(8, 6))
plt.plot(range(1, len(eigenvalues) + 1), eigenvalues, marker='o', linestyle
plt.title('Scree Plot for PCA')
plt.xlabel('Principal Components')
plt.ylabel('Eigenvalue')
plt.grid(True)
plt.show()
```




```
Therefore, based on this scree plot, it's reasonable to consider retaining
the first 2 or 3 principal components for further analysis.
```

In [105]:
```python
pc_transformed = pc.transform(data)

fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

# Plot points
ax.scatter(pc_transformed[:, 0], pc_transformed[:, 1], pc_transformed[:, 2],

# Plot lines connecting points
for i in range(len(pc_transformed)):
    ax.plot([0, pc_transformed[i, 0]], [0, pc_transformed[i, 1]], [0, pc_tra

ax.set_xlabel('Principal Component 1')
ax.set_ylabel('Principal Component 2')
ax.set_zlabel('Principal Component 3')
ax.set_title('Dimension Reduction Plot with Lines for PCA (3 Components)')

plt.show()
```



In [ ]: