```python
import numpy as np
import pandas as pd

# Define the data
WISC = ['l1', 'l2', 'l3', 'l4', 'l5', 'l6', 'l7', 'l8', 'l9', 'l10',
'l11', 'l12', 'l13', 'l14', 'l15']
CUB = [2, 3, 4, 2, 5, 3, 4, 1, 5, 3, 2, 1, 3, 4, 5]
PUZ = [3, 4, 5, 2, 1, 4, 2, 3, 5, 4, 3, 2, 1, 5, 4]
CAL = [4, 5, 3, 1, 2, 4, 2, 5, 3, 2, 4, 5, 1, 3, 2]
MEM = [3, 1, 4, 2, 5, 3, 2, 4, 1, 5, 3, 4, 2, 1, 3]
COM = [4, 2, 5, 1, 3, 4, 2, 5, 3, 4, 1, 2, 5, 3, 2]
VOC = [5, 3, 4, 1, 2, 5, 3, 4, 2, 1, 3, 5, 4, 2, 1]

# Create a DataFrame
data = pd.DataFrame({'CUB': CUB, 'PUZ': PUZ, 'CAL': CAL, 'MEM': MEM,
'COM': COM, 'VOC': VOC}, index=WISC)
data
```

	CUB	PUZ	CAL	MEM	COM	VOC
l1	2	3	4	3	4	5
l2	3	4	5	1	2	3
l3	4	5	3	4	5	4
l4	2	2	1	2	1	1
l5	5	1	2	5	3	2
l6	3	4	4	3	4	5
l7	4	2	2	2	2	3
l8	1	3	5	4	5	4
l9	5	5	3	1	3	2
l10	3	4	2	5	4	1
l11	2	3	4	3	1	3
l12	1	2	5	4	2	5
l13	3	1	1	2	5	4
l14	4	5	3	1	3	2
l15	5	4	2	3	2	1

Question 1:

Compute the correlation matrix for the given variables.

```python
# Compute the correlation matrix
corr_matrix = data.corr()

# Display the correlation matrix
corr_matrix
```

	CUB	PUZ	CAL	MEM	COM	VOC
CUB	1.000000	0.291584	-0.498864	-0.183938	-0.005065	-0.503871
PUZ	0.291584	1.000000	0.292515	-0.253218	0.142507	-0.142134
CAL	-0.498864	0.292515	1.000000	0.043049	0.034653	0.598058
MEM	-0.183938	-0.253218	0.043049	1.000000	0.308941	0.107972

```
COM -0.005065  0.142507  0.034653  0.308941  1.000000  0.422159
VOC -0.503871 -0.142134  0.598058  0.107972  0.422159  1.000000
```

Question 2:

Calculate the eigenvalues, percentage of inertia, and cumulative percentage.

```
# Compute the eigenvalues of the correlation matrix
eigenvalues = np.linalg.eigvals(corr_matrix)

# Sort the eigenvalues in descending order
eigenvalues_sorted = np.sort(eigenvalues)[::-1]

# Calculate the percentage of inertia
inertia_percentage = (eigenvalues_sorted / eigenvalues_sorted.sum()) * 
100

# Calculate the cumulative percentage
cumulative_percentage = np.cumsum(inertia_percentage)

# Display the eigenvalues, percentage of inertia, and cumulative 
percentage
result_df = pd.DataFrame({
    'Eigenvalues': eigenvalues_sorted,
    'Percentage of Inertia': inertia_percentage,
    'Cumulative Percentage': cumulative_percentage
}, index=range(1, len(eigenvalues_sorted) + 1))

result_df.index.name = 'Component'

result_df.round(2)
```

```
           Eigenvalues  Percentage of Inertia  Cumulative Percentage
Component
1                 2.21                  36.75                  36.75
2                 1.38                  23.04                  59.79
3                 1.22                  20.35                  80.14
4                 0.71                  11.83                  91.97
5                 0.37                   6.09                  98.07
6                 0.12                   1.93                 100.00
```

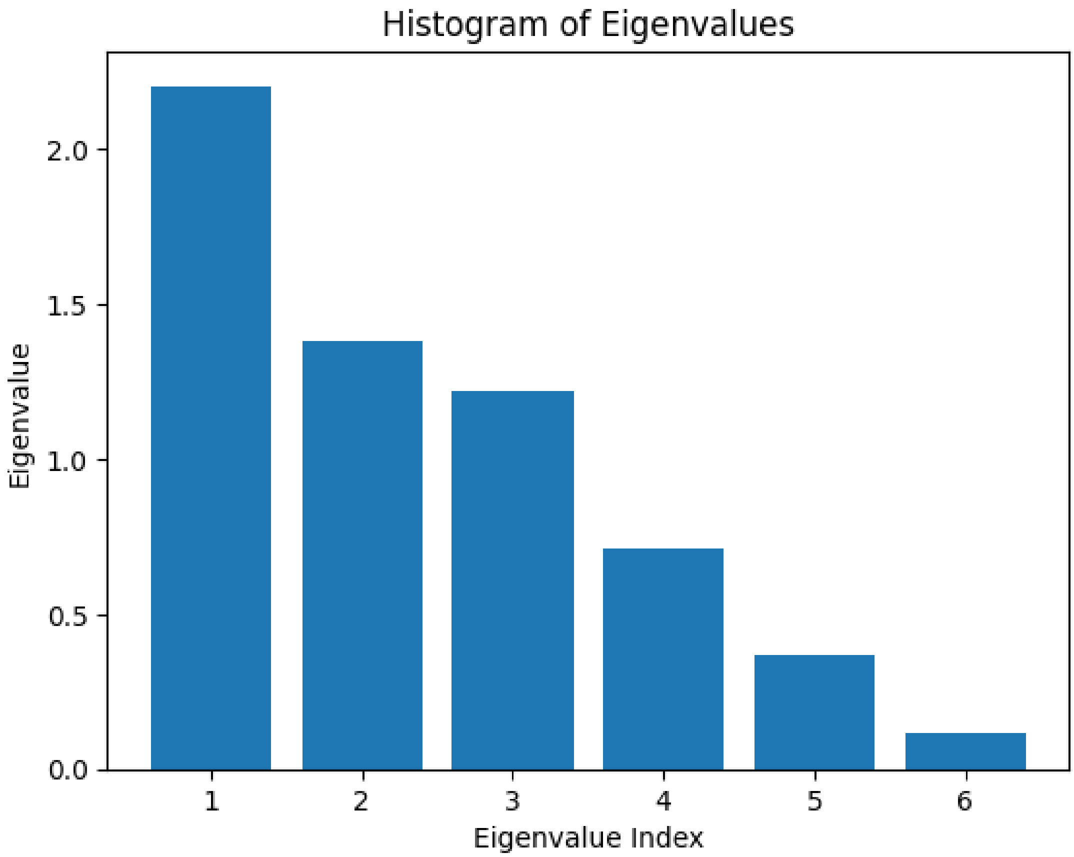
Question 3:

Create a histogram of the eigenvalues.

```
import matplotlib.pyplot as plt

# Plot the histogram of eigenvalues
plt.bar(range(1, len(eigenvalues_sorted) + 1), eigenvalues_sorted)
plt.xlabel('Eigenvalue Index')
plt.ylabel('Eigenvalue')
```

```
plt.title('Histogram of Eigenvalues')
plt.show()
```



Question 4:

Compute the principal components, contributions, and representational qualities of individuals.

```
from sklearn.decomposition import PCA
import pandas as pd
data = data.T
# Create a PCA instance
pca = PCA()

# Fit the data to the PCA model
pca.fit(data)

# Compute the principal components
principal_components = pd.DataFrame(pca.components_,
columns=data.columns, index=['PC1', 'PC2', 'PC3', 'PC4', 'PC5',
'PC6'])

# Compute the contributions of each feature to the principal
components
```

```
contributions = pd.DataFrame(np.abs(principal_components),
columns=data.columns, index=['PC1', 'PC2', 'PC3', 'PC4', 'PC5',
'PC6'])

# Compute the representational qualities of individuals
representational_qualities = np.square(contributions)


# Print the results
print("Principal Components:")
print(principal_components)

print("\nContributions:")
print(contributions)

print("\nRepresentational Qualities:")
print(representational_qualities)
```

```
Principal Components:
           l1        l2        l3        l4        l5        l6
l7  \
PC1 -0.284515 -0.012039  0.089523  0.123594  0.174959 -0.145551
0.113645
PC2 -0.061285 -0.506423  0.047592  0.034810  0.501562 -0.130726
0.022292
PC3  0.201375 -0.035629  0.203121 -0.135991 -0.224157  0.198692
0.091541
PC4 -0.012152  0.113683 -0.267189 -0.022397  0.341375  0.009980
0.425410
PC5 -0.105468  0.411658 -0.371568 -0.261287  0.415200 -0.264731 -
0.101150
PC6 -0.362937  0.115958 -0.264690  0.362011 -0.171110 -0.074961
0.118902

           l8        l9       l10       l11       l12       l13
l14  \
PC1 -0.376201  0.378065  0.197582 -0.127548 -0.474873 -0.064748
0.286543
PC2  0.012339 -0.337161  0.276532 -0.203892 -0.075231  0.323497 -
0.353417
PC3  0.047135  0.187144 -0.211700 -0.361449 -0.281978  0.659263
0.178426
PC4 -0.425145  0.014548 -0.593788  0.098869  0.173377  0.153674 -
0.138771
PC5  0.521699  0.163741 -0.019176 -0.172634 -0.132328  0.071860
0.075206
PC6 -0.033737  0.199897  0.374654  0.291489  0.239815  0.473242
0.009907
```

```
          l15
PC1  0.423478
PC2 -0.008155
PC3 -0.204059
PC4  0.034283
PC5 -0.053795
PC6 -0.238088

Contributions:
           l1        l2        l3        l4        l5        l6
l7  \
PC1  0.284515  0.012039  0.089523  0.123594  0.174959  0.145551
0.113645
PC2  0.061285  0.506423  0.047592  0.034810  0.501562  0.130726
0.022292
PC3  0.201375  0.035629  0.203121  0.135991  0.224157  0.198692
0.091541
PC4  0.012152  0.113683  0.267189  0.022397  0.341375  0.009980
0.425410
PC5  0.105468  0.411658  0.371568  0.261287  0.415200  0.264731
0.101150
PC6  0.362937  0.115958  0.264690  0.362011  0.171110  0.074961
0.118902

           l8        l9       l10       l11       l12       l13
l14  \
PC1  0.376201  0.378065  0.197582  0.127548  0.474873  0.064748
0.286543
PC2  0.012339  0.337161  0.276532  0.203892  0.075231  0.323497
0.353417
PC3  0.047135  0.187144  0.211700  0.361449  0.281978  0.659263
0.178426
PC4  0.425145  0.014548  0.593788  0.098869  0.173377  0.153674
0.138771
PC5  0.521699  0.163741  0.019176  0.172634  0.132328  0.071860
0.075206
PC6  0.033737  0.199897  0.374654  0.291489  0.239815  0.473242
0.009907

          l15
PC1  0.423478
PC2  0.008155
PC3  0.204059
PC4  0.034283
PC5  0.053795
PC6  0.238088

Representational Qualities:
           l1        l2        l3        l4        l5        l6
l7  \
```

```
PC1  0.080949  0.000145  0.008014  0.015275  0.030611  0.021185
0.012915
PC2  0.003756  0.256464  0.002265  0.001212  0.251565  0.017089
0.000497
PC3  0.040552  0.001269  0.041258  0.018494  0.050246  0.039478
0.008380
PC4  0.000148  0.012924  0.071390  0.000502  0.116537  0.000100
0.180974
PC5  0.011124  0.169462  0.138063  0.068271  0.172391  0.070083
0.010231
PC6  0.131723  0.013446  0.070061  0.131052  0.029278  0.005619
0.014138

           l8        l9       l10       l11       l12       l13
l14  \
PC1  0.141528  0.142933  0.039039  0.016268  0.225505  0.004192
0.082107
PC2  0.000152  0.113678  0.076470  0.041572  0.005660  0.104650
0.124903
PC3  0.002222  0.035023  0.044817  0.130645  0.079512  0.434628
0.031836
PC4  0.180748  0.000212  0.352584  0.009775  0.030060  0.023616
0.019257
PC5  0.272170  0.026811  0.000368  0.029802  0.017511  0.005164
0.005656
PC6  0.001138  0.039959  0.140366  0.084966  0.057511  0.223958
0.000098

          l15
PC1  0.179334
PC2  0.000067
PC3  0.041640
PC4  0.001175
PC5  0.002894
PC6  0.056686
```

Question 5:

```
The sum of the eigenvalues corresponds to the total variance in the
data.
```

Question 6:

Display the individuals in the first factorial.

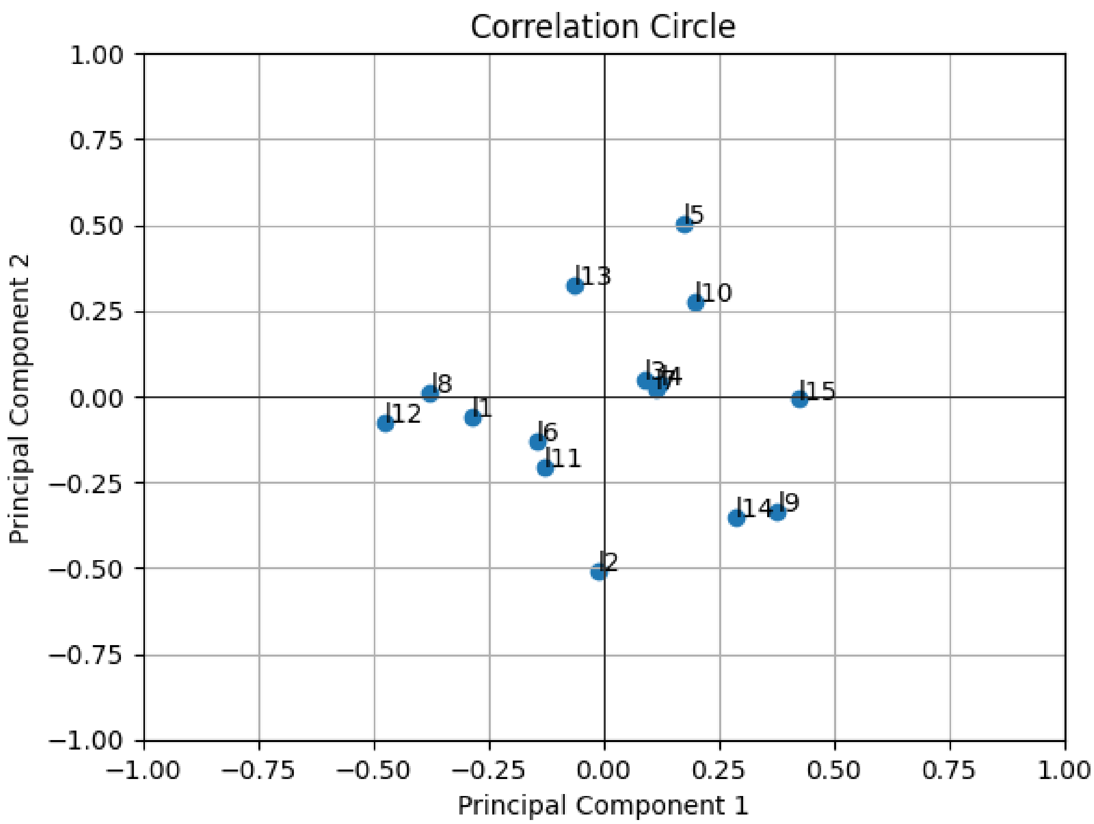
```python
import matplotlib.pyplot as plt

# Plot the principal components
plt.figure(figsize=(10, 6))

fig, ax = plt.subplots()
```

```python
ax.scatter(pca.components_[0, :], pca.components_[1, :])
for i, txt in enumerate(data.columns):
    ax.annotate(txt, (pca.components_[0, i], pca.components_[1, i]))
ax.set_xlim(-1, 1)
ax.set_ylim(-1, 1)
ax.axhline(0, color='black', linewidth=0.5)
ax.axvline(0, color='black', linewidth=0.5)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Correlation Circle')
plt.grid()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



Question 7:

Compute the principal components, contributions, and representational qualities of the variables.

```python
from sklearn.decomposition import PCA
import pandas as pd
data = data.T
# Create a PCA instance
pca = PCA()
```

```python
# Fit the data to the PCA model
pca.fit(data)

# Compute the principal components
principal_components = pd.DataFrame(pca.components_,
columns=data.columns, index=['PC1', 'PC2', 'PC3', 'PC4', 'PC5',
'PC6'])

# Compute the contributions of each feature to the principal
components
contributions = pd.DataFrame(np.abs(principal_components),
columns=data.columns, index=['PC1', 'PC2', 'PC3', 'PC4', 'PC5',
'PC6'])

# Compute the representational qualities of individuals
representational_qualities = np.square(contributions)


# Print the results
print("Principal Components:")
print(principal_components)

print("\nContributions:")
print(contributions)

print("\nRepresentational Qualities:")
print(representational_qualities)
```

```
Principal Components:
          CUB       PUZ       CAL       MEM       COM       VOC
PC1 -0.480336 -0.076206  0.496116  0.212589  0.279091  0.627896
PC2 -0.161085 -0.746709 -0.422595  0.484947  0.013789 -0.050272
PC3  0.390567  0.282356 -0.212851  0.361038  0.768558  0.037377
PC4  0.118787 -0.326371 -0.328842 -0.713427  0.282843  0.426915
PC5  0.748247 -0.228092  0.290147  0.200390 -0.329390  0.394031
PC6  0.129623 -0.445365  0.581080 -0.200373  0.377807 -0.514107

Contributions:
          CUB       PUZ       CAL       MEM       COM       VOC
PC1  0.480336  0.076206  0.496116  0.212589  0.279091  0.627896
PC2  0.161085  0.746709  0.422595  0.484947  0.013789  0.050272
PC3  0.390567  0.282356  0.212851  0.361038  0.768558  0.037377
PC4  0.118787  0.326371  0.328842  0.713427  0.282843  0.426915
PC5  0.748247  0.228092  0.290147  0.200390  0.329390  0.394031
PC6  0.129623  0.445365  0.581080  0.200373  0.377807  0.514107

Representational Qualities:
```

```
          CUB       PUZ       CAL       MEM       COM       VOC
PC1  0.230722  0.005807  0.246131  0.045194  0.077892  0.394253
PC2  0.025948  0.557574  0.178586  0.235174  0.000190  0.002527
PC3  0.152543  0.079725  0.045306  0.130348  0.590682  0.001397
PC4  0.014110  0.106518  0.108137  0.508978  0.080000  0.182256
PC5  0.559874  0.052026  0.084185  0.040156  0.108498  0.155261
PC6  0.016802  0.198350  0.337654  0.040150  0.142738  0.264306
```

Question 8:

Plot the correlation circle.

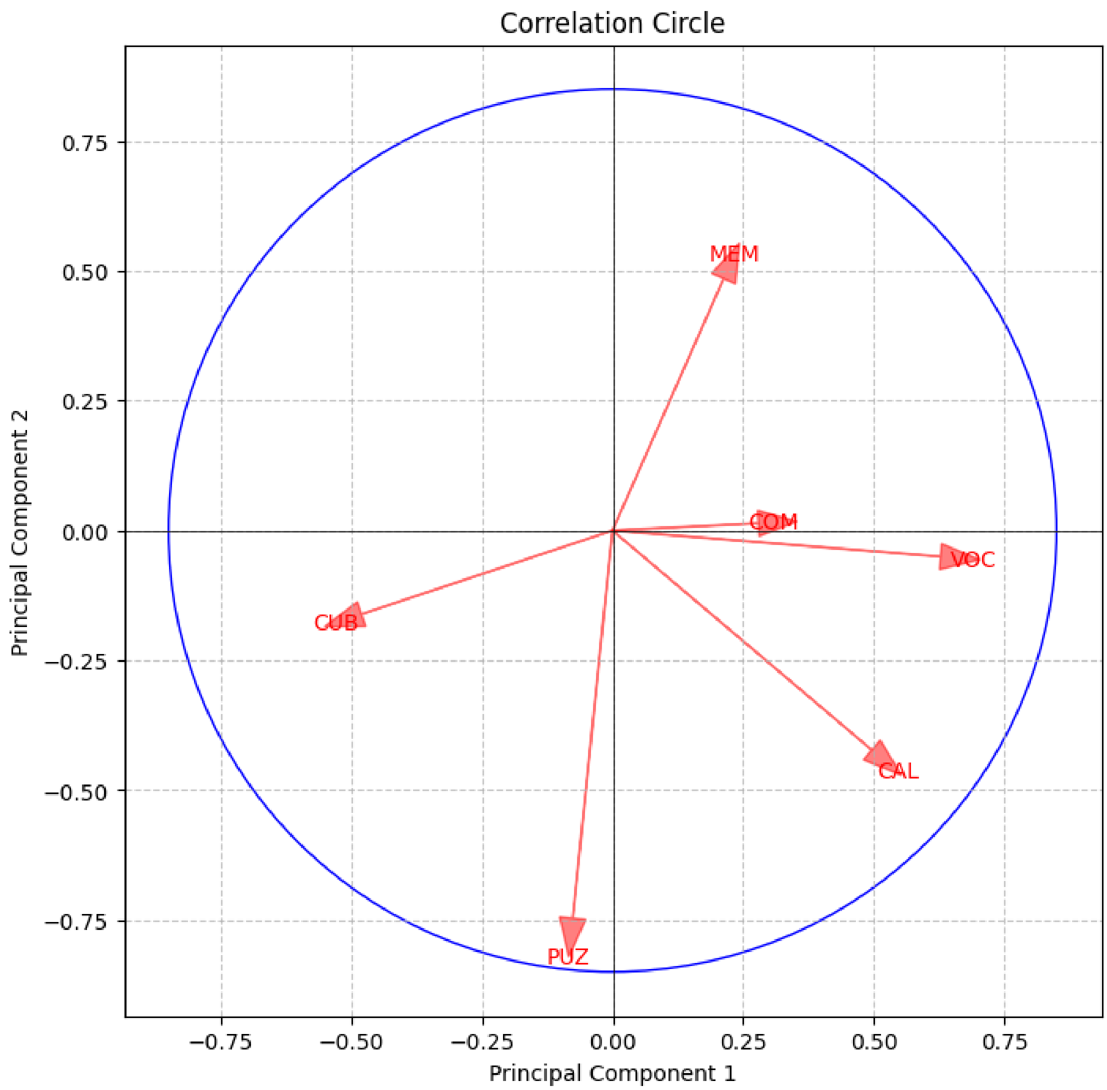
```
# Plot the correlation circle
fig, ax = plt.subplots(figsize=(8, 8))
for i, var in enumerate(data.columns):
    ax.arrow(0, 0, principal_components[var][0],
principal_components[var][1],
             color='r', alpha=0.5, head_width=0.05)
    ax.text(principal_components[var][0] * 1.1,
principal_components[var][1] * 1.1, var,
            color='r', ha='center', va='center')

# Add a circle
circle = plt.Circle((0, 0), radius=0.85, color='b', fill=False)
ax.add_patch(circle)

plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Correlation Circle')

plt.axhline(0, color='black', lw=0.5)
plt.axvline(0, color='black', lw=0.5)
plt.grid(True, linestyle='--', alpha=0.7)

plt.show()
```



Question 9:

We choose to study only the first two principal components because they capture the highest amount of variance in the data. Looking at the table of eigenvalues, the first two eigenvalues are typically much larger than the rest, indicating that the first two principal components explain most of the variability in the data.

Question 10:

To determine which subtests are most strongly correlated, we can examine the correlation matrix. The variables with higher correlation coefficients (closer to 1 or -1) are more strongly correlated.

the subtests with the strongest positive correlations (coefficients closest to 1) are:

- CAL and VOC (correlation coefficient: 0.598058)
- PUZ and CAL (correlation coefficient: 0.292515)

On the other hand, the subtests with the strongest negative correlations (coefficients closest to -1) are:

- CUB and VOC (correlation coefficient: -0.503871)
- CAL and CUB (correlation coefficient: -0.498864)

Question 11:

The graphical presentation of the variables are well presented in the (CP1, CP2) plane. Justify this statement:

- The variables are well presented in the (PC1, PC2) plane because they are effectively represented by the principal components, allowing for clear visualization and understanding of their relationships and contributions.