1. Normalization methods with formula (value ranges of output, etc.)
    a. Min-max normalization
        i. Formula

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \times (new_{max} - new_{min}) + new_{min}$$

            1. x = original data point
            2. min(x) = minimum value of the feature
            3. max(x) = maximum value of the feature
            4. newMin, newMax = desired range of normalized data (typically [0, 1] or [-1, 1])
        ii. Output range
            1. Typically between [0, 1], but can be scaled to any other range (e.g., [-1, 1]).
    b. Z-score normalization
        i. Formula

$$z = \frac{x - \mu}{\sigma}$$

            1. x = original data point
            2. u = mean of the feature
            3. sigma = standard deviation of feature
        ii. Output range
            1. No specific range, but most values lie between [-3, 3] for normally distributed data
    c. Z-score normalization using the mean absolute deviation instead of SD
        i. Formula

$$z_{MAD} = \frac{x - \mu}{MAD}$$

            1. x = original data point
            2. u = mean of feature
            3. MAD = mean absolute deviation

$$MAD = \frac{1}{n} \sum_{i=1}^{n} |x_i - \mu|$$

        ii. Output range
            1. No specific range, similar to regular Z-score normalization.
            2. Scale of the output depends on the distribution of the data and is often smaller compared to using standard deviation
    d. Normalization by decimal scaling

i. Formula

$$x' = \frac{x}{10^j}$$

1. j = smallest integer s.t. maximum absolute value of x' < 1
2. j = ⌈log10(max(|x|))⌉

ii. Output range

1. Depends on the magnitude of the original data, but typically within [-1, 1]

2. Normalization for group of data
   a. Min-max normalization with min = 0 and max = 1
      i. min(x) = 200, max(x) = 1000
      ii. Using above formula
         1. For 200
            a. x' = (200 - 200) / (1000 - 200) = 0
         2. For 300
            a. x' = (300 - 200) / (1000 - 200) = 0.125
         3. For 400
            a. x' = (400 - 200) / (1000 - 200) = 0.25
         4. For 600
            a. x' = (600 - 200) / (1000 - 200) = 0.5
         5. For 1000
            a. x' = (1000 - 200) / (1000 - 200) = 1
         6. Result: **[0, 0.125, 0.25, 0.5, 1]**
   b. Z-score normalization
      i. Calculating mean and standard deviation
         1. Mean = (200 + 300 + 400 + 600 + 1000)/ 5 = 500
         2. Standard deviation
            a. SD = √(200 - 500)^2 + (300 - 500)^2 + (400 - 500)^2 + (600 - 500)^2 + (1000 - 500)^2 / 5
            b. SD = √400000/5
            c. SD = √80000
            d. SD = 282.84
         3. Calculating Z-scores
            a. For 200

$$z = \frac{200 - 500}{282.84} = \frac{-300}{282.84} \approx -1.06$$

            i.
            b. For 300

$$z = \frac{300 - 500}{282.84} = \frac{-200}{282.84} \approx -0.71$$

            i.
            c. For 400

i. $$z = \frac{400 - 500}{282.84} = \frac{-100}{282.84} \approx -0.35$$

d. For 600

i. $$z = \frac{600 - 500}{282.84} = \frac{100}{282.84} \approx 0.35$$

e. For 1000

i. $$z = \frac{1000 - 500}{282.84} = \frac{500}{282.84} \approx 1.77$$

f. Result: **[−1.06, −0.71, −0.35, 0.35, 1.77]**

c. Z-score normalization using the mean absolute deviation instead of SD

i. Calculate the Mean Absolute Deviation (MAD)

1. MAD = (|200−500|+|300−500|+|400−500|+|600−500|+|1000−500|)/ 5
2. MAD = 1200 / 5 = 240

ii. Calculating Z-scores using MAD

1. For 200
   a. zMad = (200 - 500) / 240 = -300 / 240 = 1.25
2. For 300

a. $$z_{MAD} = \frac{300 - 500}{240} = \frac{-200}{240} \approx -0.83$$

3. For 400

a. $$z_{MAD} = \frac{400 - 500}{240} = \frac{-100}{240} \approx -0.42$$

4. For 600

a. $$z_{MAD} = \frac{600 - 500}{240} = \frac{100}{240} \approx 0.42$$

5. For 1000

a. $$z_{MAD} = \frac{1000 - 500}{240} = \frac{500}{240} \approx 2.08$$

6. Result: **[−1.25, −0.83, −0.42, 0.42, 2.08]**

d. Normalization by decimal scaling

i. Using the formula described above for decimal scaling

ii. Maximum value = 1000

iii. j = log10(1000) = 3
iv. Divide each value by 10^3 = 1000
1. For 200
a. x' = 200 / 1000 = 0.2
2. For 300
a. x' = 300 / 1000 = 0.3
3. For 400
a. x' = 400 / 1000 = 0.4
4. For 600
a. x' = 600 / 1000 = 0.6
5. For 1000
a. x' = 1000 / 1000 = 1
6. Result: **[0.2, 0.3, 0.4, 0.6, 1]**

3. **Flowchart** (procedures for attribute subset selection
a. Stepwise forward selection
i. Steps
1. Begin
a. Start of the process
2. Initialize required attribute set
a. Begin with an empty set for selected features
3. Select best attribute
a. Evaluate all available attributes using statistical measures, such as p-value or F-score to determine which attribute contributes the most to the model
4. Check if attribute exceeds stopping threshold
a. Compare selected attribute to predefined threshold (this determines whether an attribute is important or not)
b. If yes, proceed
c. If no, stop
5. Add selected attribute to reduced attribute set
a. Best attribute passing the threshold is added to selected set
6. Check for more attributes
a. If there are more attributes left
i. Loop back to select next best attribute
b. Else stop the process
7. End
a. Process concludes when no more attributes left to select / threshold condition not met

4. Pair similarity
a. **Symmetric attribute**s
i. Under symmetric attributes, the similarity between two individuals is based on the number of traits they share. If two individuals share more traits, they are considered more similar.

b. Asymmetric attributes

i. Under asymmetric attributes, the similarity is based on the number of traits that one individual possesses that the other also possesses. This implies that having more traits in common is more important for one individual than the other.

c. Similarity Calculation

i. We'll calculate the similarity between each pair of individuals using both symmetric and asymmetric attributes.

ii. **Symmetric attributes**:

1. Similarity(A, B) = Number of shared traits between A and B / Total number of traits
2. Most compatible pair under symmetric attributes: ('Kevin', 'Eric')

iii. Asymmetric attributes

1. Similarity(A, B) = Number of traits A possesses that B also possesses / Total number of traits A possesses
2. Most compatible pair under asymmetric attributes: ('Eric', 'Kevin')

5. Data of participants and attributes

a. Step 1

i. Checking data quality

1. Cleaning
    a. Missing values: The column "Social problems" has 4 missing values.
2. Transformation
    a. Handling missing values: We will impute the missing values in the "Social problems" column using the mean, a common approach for small datasets.
    b. Outlier handling: We will examine the row with zero values, and if deemed invalid, it will be dropped.
    c. Scaling: Variables will be standardized (z-score normalization) since PCA is sensitive to variable scales.
3. Missing
    a. Scale differences: Variables have different ranges, so normalization might be needed for further analysis, especially for PCA.
4. Outliers
    a. Outliers: The 5th row contains zero values for all variables, which could indicate a problematic or special case.
5. Scale / Normalization
    a. Normalization: The data (except the "D" identifier column) was standardized using z-score normalization to ensure consistent scales for all variables.

b. Step 2

i. Covariance matrix of data after step **a**

1. **Compute the Covariance Matrix**: We'll compute the 10x10 covariance matrix of the cleaned and normalized data.
2. **Compute the Total Variance**: This will be the sum of the diagonal elements of the covariance matrix.
3. **Compute Pearson's Correlation** between Variable 1 ("Fluid IQ") and Variable 2 ("Crystallized IQ")

ii. Total variance of data

1. **Covariance Matrix** (10x10): The matrix displays covariances among the variables after scaling.
2. **Total Variance**: The sum of diagonal elements is approximately 10.36.

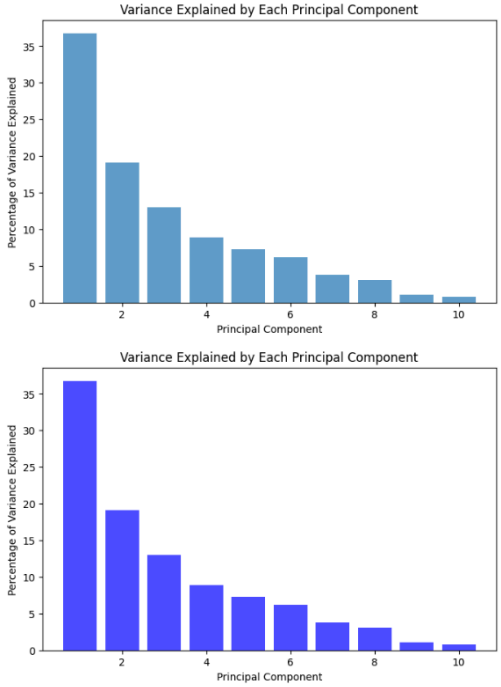
iii. Pearson's correlation between variable 1 and variable 2

1. **Pearson's Correlation** between "Fluid IQ" and "Crystallized IQ" is 0.49, indicating a moderate positive correlation.

c. PCA

i. Plot percentage of variances of each PC in decreasing order

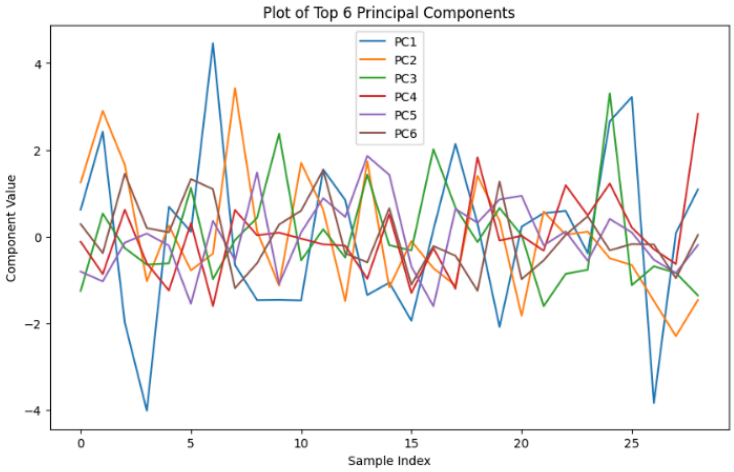
1.



ii. Components needed to capture > 85% total data variance

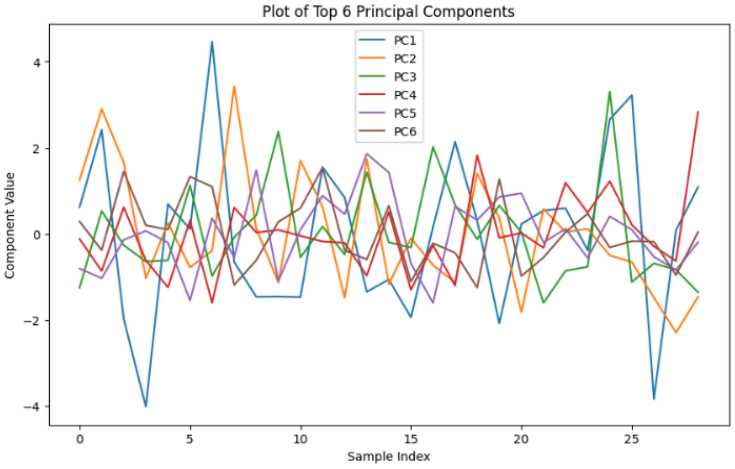
1. Components needed to capture >85% variance: 6

iii. PC plot of N components selected



1.

iv. Plot of generated (NEW) top P PC variables I selected



1.

v. Computing covariance matrix of NEW P PC variables

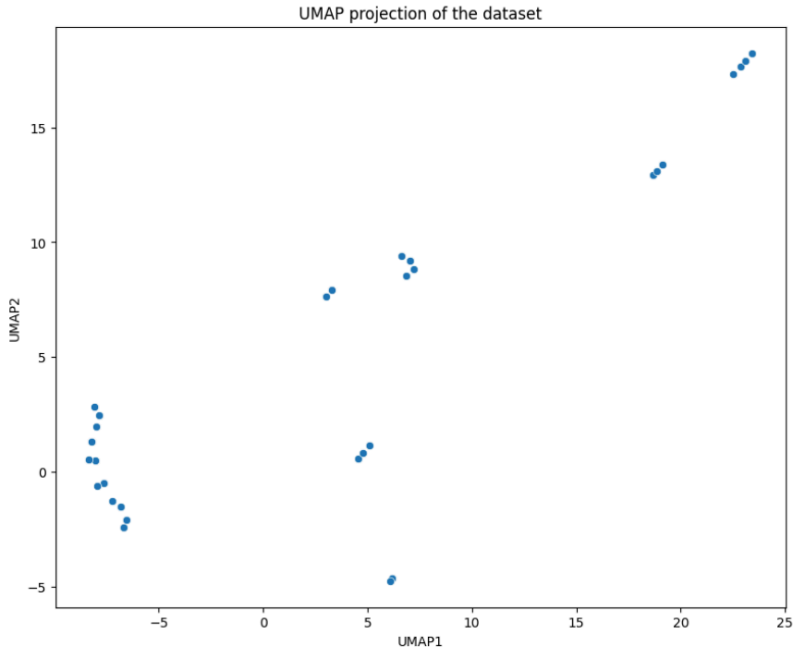
```
Covariance Matrix of the Top P Principal Components:
[[ 3.80573576e+00 -3.66982678e-16 -1.34658430e-17 -3.47474949e-17
   2.39186012e-16 -1.70200090e-16]
 [-3.66982678e-16  1.97585731e+00  1.68530159e-16 -2.90877699e-16
   4.03783812e-17 -3.68910873e-16]
 [-1.34658430e-17  1.68530159e-16  1.34201769e+00 -4.32233800e-16
  -8.10662922e-18 -2.32186527e-16]
 [-3.47474949e-17 -2.90877699e-16 -4.32233800e-16  9.19489491e-01
  -1.07997755e-16  2.21256782e-17]
 [ 2.39186012e-16  4.03783812e-17 -8.10662922e-18 -1.07997755e-16
   7.58785616e-01  3.88122731e-16]
 [-1.70200090e-16 -3.68910873e-16 -2.32186527e-16  2.21256782e-17
   3.88122731e-16  6.45008663e-01]]
```

1.

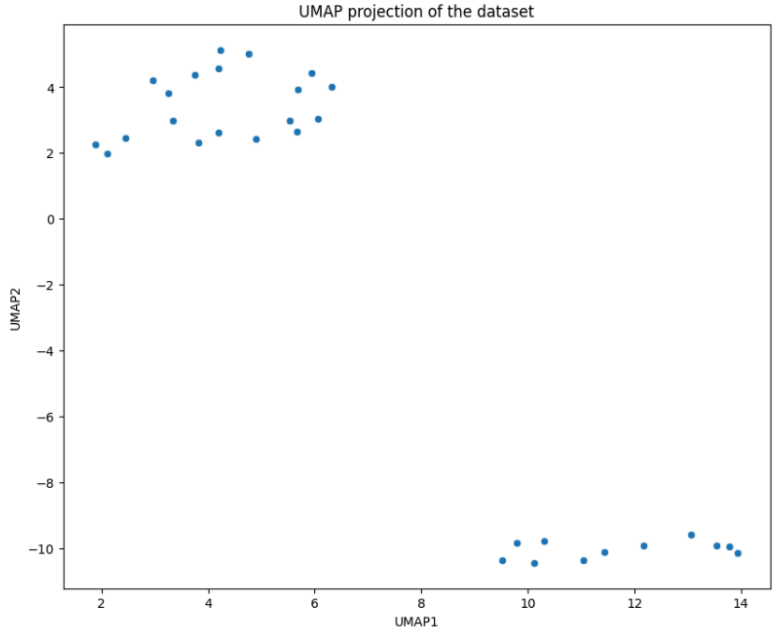
vi. Pearson's correlation between PC1 and PC2

1. Pearson's correlation coefficient: -1.35e-16
2. P-value: 0.99

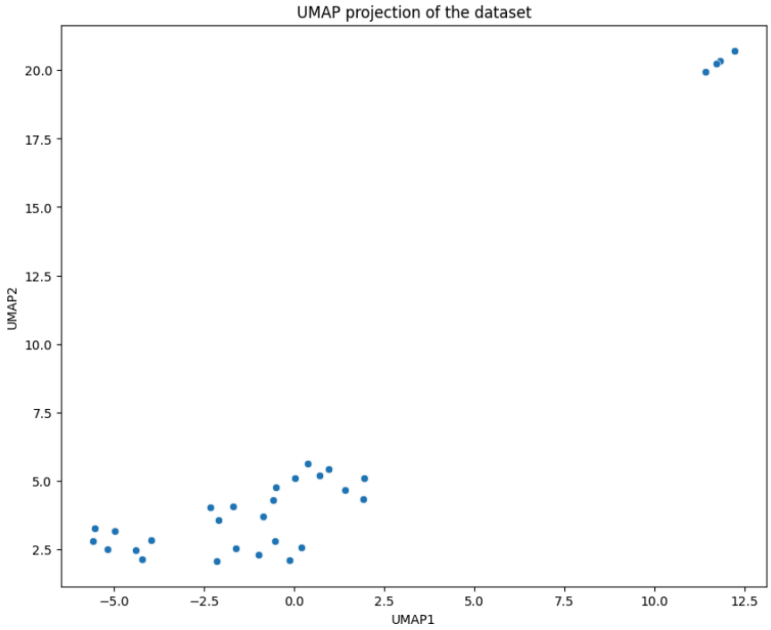
6. UMap or tSNE for visualization of clean data in 2D
   a. Selecting parameters with smaller neighbors
      i. **n_neighbors:** Lower values (e.g., 2–10) focus more on local structures


UMAP projection of the dataset
UMAP2
UMAP1
15
10
5
0
−5
−5
0
5
10
15
20
25

N = 2


UMAP projection of the dataset
UMAP2
UMAP1
4
2
0
−2
−4
−6
−8
−10
2
4
6
8
10
12
14

N = 3


UMAP projection of the dataset
UMAP2
UMAP1
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5

N = 4