# Analysis of categorical sequences

Adriana Clavijo Daza

2023-05-12

## Optimal Matching

Optimal Matching (OM) is a technique used in social sciences for the comparison of sequences of categorical states indexed by time. this method has applications in different areas of social sciences, for instance, life course or career path analysis. OM uses the Needleman-Wunsch algorithm, that was developed to compare biological sequences. This algorithm is an application of dynamic programming, an iterative method that simplifies an optimization problem by breaking it into a recursion of smaller problems that are simpler to solve.

## The OM algorithm

Given a set of $n$ states, say, $S = \{s_1, \ldots, s_n\}$ a sequence of size $t > 0$ can be denoted as $X = (x_1, \ldots, x_t)$, where $x_i \in S$ for $i = 1, \ldots, t$. Additionally, the set of all possible sequences with states belonging to $S$ is denoted by $\mathbf{S}$.

Now, let $X, Y \in \mathbf{S}$ be two sequences of size $t_1$ and $t_2$, respectively. In order to find the optimal way to align these two sequences, we define an empty array, $F$, of size $(t_1 + 1) \times (t_2 + 1)$. The algorithm below explains how the array $F$ is filled.

1: $F(1,1) \leftarrow 0$
2: **for** $j \leftarrow 2, t_2 + 1$ **do**
3: $\quad F(1,j) \leftarrow F(1,j-1) + d$
4: **end for**
5: **for** $i \leftarrow 2, t_1 + 1$ **do**
6: $\quad F(i,1) \leftarrow F(i-1,1) + d$
7: **end for**
8: **for** $i \leftarrow 2, t_1 + 1$ **do**
9: $\quad$ **for** $j \leftarrow 2, t_2 + 1$ **do**
10: $\quad\quad F(i,j) \leftarrow min\{F(i-1,j) + d, F(i,j-1) + d, F(i-1,j-1) + k(y_{i-1}, x_{j-1})\}$
11: $\quad$ **end for**
12: **end for**

Here, $d$ is the cost of inserting a gap, and $k(y_{i-1}, x_{j-1})$ is the cost associated to change from the state $y_{i-1}$ to $x_{j-1}$, which is defined in a matrix $K$ of size $n \times n$, usually known as the cost matrix.

The cost of inserting a gap, $d$, to align the sequences is also known as INDEL (insert or delete) cost and it can also be interpreted as deleting a term of the sequence $X$ or adding a term to the sequence $Y$, which is equivalent.

Lines 1-7 of the algorithm correspond to initialization and equation; starting with a cost of 0 in $F(1,1)$ and with the first row and column representing cumulative costs of successively adding gaps. The remaining lines of the algorithm correspond to the row-wise recursion to fill the array $F$ according to the content of the sequences to be aligned. At any step of the recursion, the algorithm is looking at a specific pair of indexes (location) and calculating if substitution or insertion/deletion is the cheapest operation. Successively adding the costs of the cheapest operations results in the overall optimal cost for aligning the sequences $X$ and $Y$.

When $F$ is completely filled, the value in the last cell, i.e. $F(t_1+1, t_2+1)$ corresponds to the optimal cost of aligning the sequences $X$ and $Y$. It is possible to recover the steps with a traceback from the last cell. However, this is not necessary to perform OM.

### Cost matrix

The `R` package `TraMineR` provides several functions to work with sets of sequences. The package implements OM and offers several methods for computing the cost matrix $K$.

**Transition rates (`TRATE`)** The substitution cost between states $s_i$ and $s_j$, $1 \leq i, j \leq n$, is calculated as:

$$K(s_i, s_j) = c - P(s_i|s_j) - P(s_j|s_i), \tag{1}$$

where $P(s_i|s_j)$ is the probability of transition from state $s_j$ in time $t$ to $s_i$ in time $t+1$ and $c$ is a constant, set to a value such that $0 \leq K(s_i, s_j) \leq 2$).

**Chi-squared distance (`FUTURE`)**

$$K(s_i, s_j) = ChiDist(\mathbf{P_i}, \mathbf{P_j}), \tag{2}$$

where $\mathbf{P}_\cdot = (P(s_1|.), \ldots, P(s_n|.))'$

**Relative frequencies (`INDELS` and `INDELSLOG`)**

$$K(s_i, s_j) = indel_i + indel_j, \tag{3}$$

where $indel_i = 1/f_i$ for method `INDEL`, $indel_i = \log[2/(1+f_i)]$ and $f_i$ is the relative frequency of the state $s_i$ for $i = 1, \ldots, n$.

### Example

Let us suppose that $S$ is the alphabet, let $X = \{S, E, N, D\}$ and $Y = \{A, N, D\}$ be two sequences in $\mathbf{S}$. Supposing that $d = 2$ and

$$K(i,j) = \begin{cases} 0 & \text{if } i = j, \\ 3 & \text{otherwise} \end{cases}$$

The array $F$ is initialized as follows:

| | | S | E | N | D |
|---|---|---|---|---|---|
| | 0 | 2 | 4 | 6 | 8 |
| A | 2 | | | | |
| N | 4 | | | | |
| D | 6 | | | | |

To fill the second row of $F$ we proceed as follows:

- $F(2,2) = min\{F(1,2)+d, F(2,1)+d, F(1,1)+k(y_1,x_1)\} = min\{2+2, 2+2, 0+3\} = 3$
- $F(2,3) = min\{F(1,3)+d, F(2,2)+d, F(1,2)+k(y_1,x_2)\} = min\{4+2, 3+2, 2+3\} = 5$
- $F(2,4) = min\{F(1,4)+d, F(2,3)+d, F(1,3)+k(y_1,x_3)\} = min\{6+2, 5+2, 4+3\} = 7$
- $F(2,5) = min\{F(1,5)+d, F(2,4)+d, F(1,4)+k(y_1,x_4)\} = min\{8+2, 7+2, 6+3\} = 9$
- $F(3,2) = min\{F(2,2)+d, F(3,1)+d, F(2,1)+k(y_2,x_1)\} = min\{3+2, 4+2, 2+3\} = 5$
- $F(3,3) = min\{F(2,3)+d, F(3,2)+d, F(2,2)+k(y_2,x_2)\} = min\{5+2, 5+2, 3+3\} = 6$

- $F(3,4) = min\{F(2,4) + d, F(3,3) + d, F(2,3) + k(y_2, x_3)\} = min\{5+2, 5+2, 5+0\} = 5$
- $F(3,5) = min\{F(2,4) + d, F(3,4) + d, F(2,4) + k(y_2, x_4)\} = min\{7+2, 5+2, 5+3\} = 7$
- $F(4,2) = min\{F(3,2) + d, F(4,1) + d, F(3,1) + k(y_3, x_1)\} = min\{5+2, 6+2, 4+3\} = 7$
- $F(4,3) = min\{F(3,3) + d, F(4,2) + d, F(3,2) + k(y_3, x_2)\} = min\{6+2, 7+2, 5+3\} = 8$
- $F(4,4) = min\{F(3,4) + d, F(4,3) + d, F(3,3) + k(y_3, x_3)\} = min\{5+2, 8+2, 5+3\} = 7$
- $F(4,5) = min\{F(3,5) + d, F(4,4) + d, F(3,4) + k(y_3, x_4)\} = min\{7+2, 7+2, 5+0\} = 5$

| | | S | E | N | D |
|---|---|---|---|---|---|
| | 0 | 2 | 4 | 6 | 8 |
| A | 2 | 3 | 5 | 7 | 9 |
| N | 4 | 5 | 6 | 5 | 7 |
| D | 6 | 7 | 8 | 7 | 5 |

In this simple example, we can easily obtain two optimal (equivalent) alignments:
S E N D with
A - N D or
- A N D

In both cases we have two matches (cost 0), one mismatch (cost 3) and one gap (cost 2), giving a total cost 5 that is exactly what we obtained in the last cell of $F$.

#### Normalization

For cases when we have sequences of different lengths, it can be useful to account for this differences with a normalization factor.

Given a set two sequences $X, Y \in \mathbf{S}$ of length $t_1$ and $t_2$, respectively. Let $d_{X,Y}$ be the distance between the sequences $X$ and $Y$, $t_{max}$ the length of the longest sequence in $\mathbf{S}$ and $d_{max}$ the maximum distance between any pair of sequences in $\mathbf{S}$.

We can find the following options to normalize the distances between sequences:

- `maxlength`: $d_{X,Y}/t_{max}$
- `gmean`: $1 - \frac{d_{max} - d_{X,Y}}{\sqrt{t_1 * t_2}}$
- `maxdist`: $d_{X,Y}/d_{max}$

## Data from the 40+ Healthy Aging Study

#### About the data

As part of the Women 40+ Healthy Aging Study, a large study that was conducted by the Department of Clinical Psychology and Psychotherapy of the University of Zurich, a psychometric instrument was developed in order to obtain information about the history of romantic relationships of women. The study was conducted between June 2017 and February 2018 with women between 40 and 75 years who (self-)reported good, very good or excellent health condition and the absence of acute or chronic somatic disease or mental disorder. The participants who reported psychotherapy or psychopharmacological treatment in the previous 6 months were excluded as well as habitual drinkers. Other exclusion criteria were pregnancy in the last 6 months, premature menopause, surgical menopause, intake of hormonal treatment (including contraceptives), shift-work and recent long-distance flight. The participants were recruited from the general population using online advertisement and flyers.

The questionnaire asked the participants to provide information about relationship phases starting from the age of 15 years until the current age at the time of the data collection. The phases were defined by the start and end age and for each phase and information about civil status, relationship status, living situation, children and quality of the relationship was collected. Before including the data corresponding to their own

history, the participants were prompted to answer some of the questions based on an example. Some of the participants were excluded when the example entries were not correctly filled. In total 250 individuals were considered in the analysis.

In order to create a sequence for each participant the information about civil status, relationship status, living situation and the matenity is taken into account. A yearly sequence is created and the states considered are the following:

- 1 = Single + no children
- 2 = Single + children
- 3 = Changing relationships + no children
- 4 = Changing rel. + children
- 5 = Relationship + living apart + no children
- 6 = Relationship + living together + no children
- 7 = Relationship + living apart + children
- 8 = Relationship + living together + children
- 9 = Married + no children
- 10 = Married + children

Additionally, personality scores for the women included in the study are available. Personality refers to the enduring characteristics and behavior that comprise a person's unique adjustment to life, including major traits, interests, drives, values, self-concept, abilities, and emotional patterns. These scores are obtained via psychometric instruments and evaluate the main personality traits:

- Agreeableness
- Conscientiousness
- Extraversion
- Neuroticism
- Openness

Optimal matching analysis is performed with the aim to obtain clusters of sequences that are similar and characterize the most common relationship history profiles.

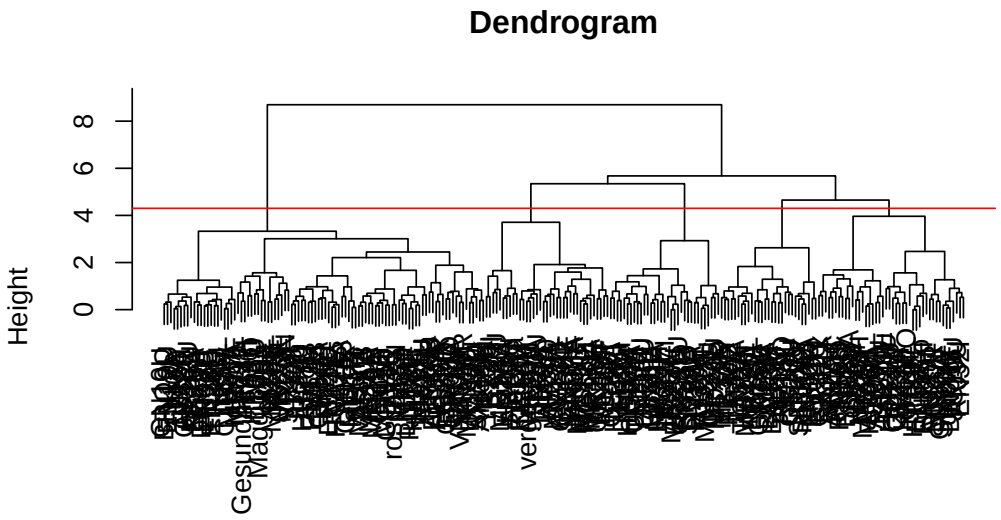
### Application of OM

Using the `R` package `TraMineR` the cost matrix is calculated with transition rates between states.

| Status | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0000 | 2.0000 | 1.9845 | 2.0000 | 1.9246 | 1.9462 | 2.0000 | 1.9950 | 1.9846 | 1.9836 |
| 2 | 2.0000 | 0.0000 | 2.0000 | 2.0000 | 2.0000 | 2.0000 | 1.9552 | 1.9224 | 2.0000 | 1.9985 |
| 3 | 1.9845 | 2.0000 | 0.0000 | 2.0000 | 1.9409 | 1.9198 | 2.0000 | 1.9982 | 1.9736 | 1.9765 |
| 4 | 2.0000 | 2.0000 | 2.0000 | 0.0000 | 1.9878 | 2.0000 | 1.9550 | 1.9512 | 2.0000 | 1.9965 |
| 5 | 1.9246 | 2.0000 | 1.9409 | 1.9878 | 0.0000 | 1.9522 | 1.9772 | 1.9903 | 1.9808 | 1.9745 |
| 6 | 1.9462 | 2.0000 | 1.9198 | 2.0000 | 1.9522 | 0.0000 | 2.0000 | 1.9971 | 1.9766 | 1.9583 |
| 7 | 2.0000 | 1.9552 | 2.0000 | 1.9550 | 1.9772 | 2.0000 | 0.0000 | 1.9708 | 2.0000 | 1.9965 |
| 8 | 1.9950 | 1.9224 | 1.9982 | 1.9512 | 1.9903 | 1.9971 | 1.9708 | 0.0000 | 2.0000 | 1.9836 |
| 9 | 1.9846 | 2.0000 | 1.9736 | 2.0000 | 1.9808 | 1.9766 | 2.0000 | 2.0000 | 0.0000 | 1.9862 |
| 10 | 1.9836 | 1.9985 | 1.9765 | 1.9965 | 1.9745 | 1.9583 | 1.9965 | 1.9836 | 1.9862 | 0.0000 |

From this cost matrix it is possible to calculate pairwise distances between sequences using the algorithm previously described. A correction of the distances is done to account for the differences in size of the sequences. This is done dividing the obtained distance by the length of the longest sequence.

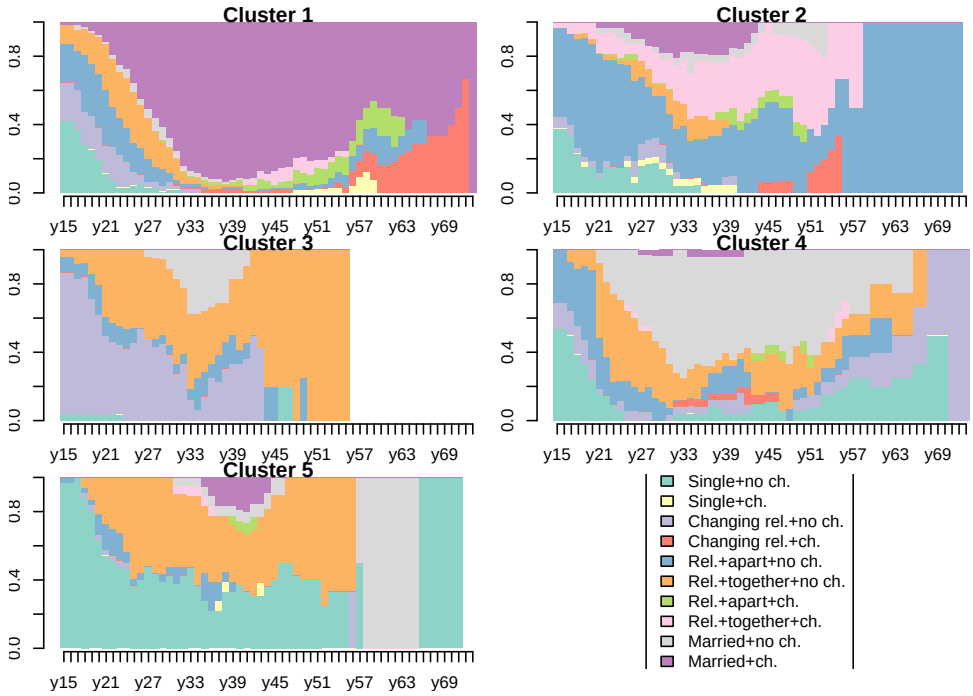
We then use this sequences to apply a hierarchical agglomerative clustering method called AGNES. The following figure shows the dendrogram. In this case, we decided to cut at 5 clusters in order to preserve enough individuals in each cluster.



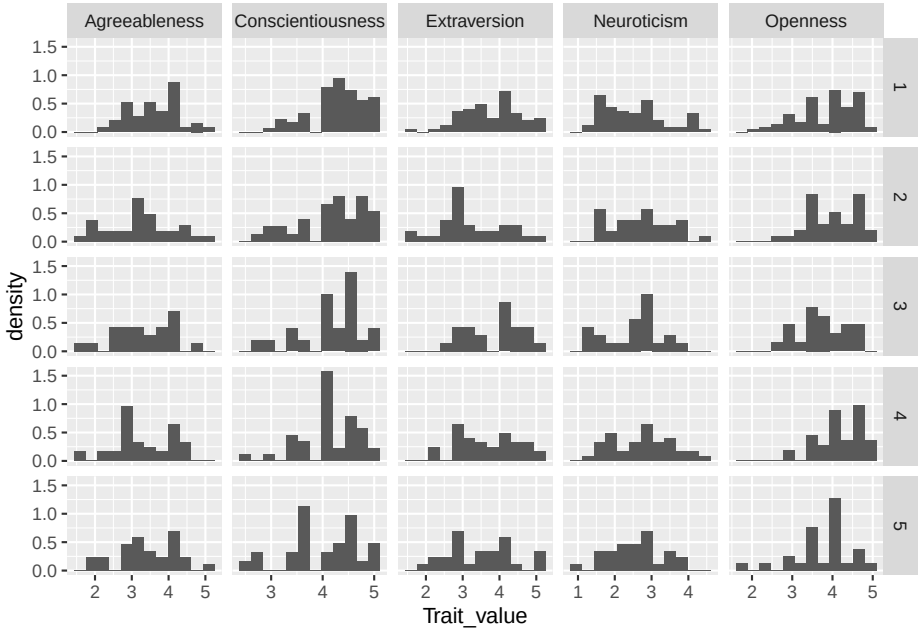

| Cluster | n |
|---|---|
| 1 | 96 |
| 2 | 37 |
| 3 | 29 |
| 4 | 44 |
| 5 | 33 |

We can visualize the clusters of sequences to and try to identify common features to describe them. It is important to consider that this description is subjective but can be useful to characterize the groups.

- Cluster 1: Married with children then divorced/widowed
- Cluster 2: Sequences with more changes (unstable)
- Cluster 3: Younger, mostly not married without children
- Cluster 4: Older, without children
- Cluster 5: Married late then divorced/widowed, without children



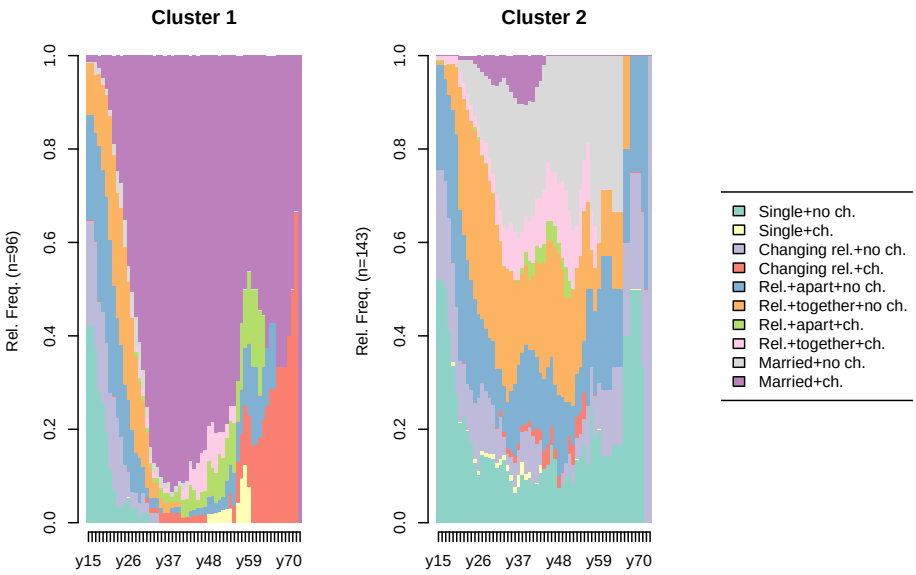

Now, we are interested in predicting the personality scores based on the relationships history of the women. The following figure shows the distribution of the score for each trait by cluster.



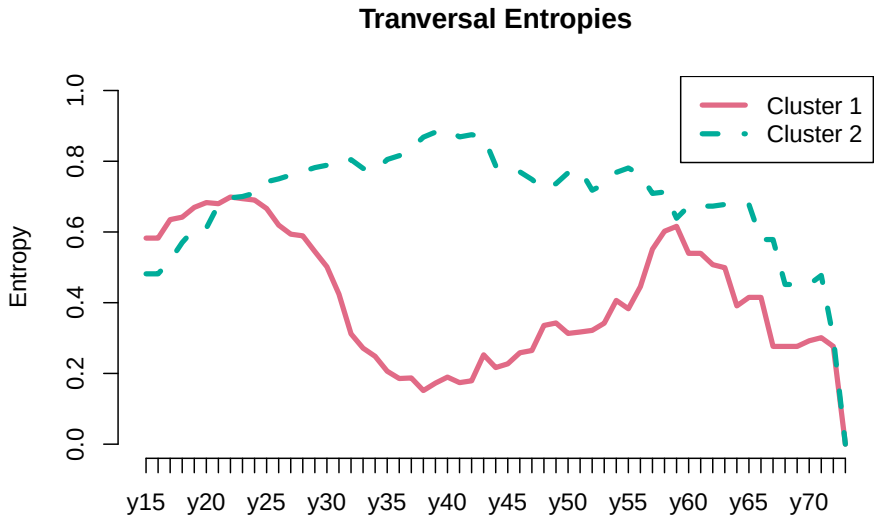

No difference is obvious at a first glance. However, we can also use the distance matrix to obtain predictions of the personality traits using $k$-nearest neighbors ($k$NN); a non-parametric method used for prediction.

We also explore with a lower number of clusters in order to find better defined groups. As we can observe in the distribution plots of the sequences states, the majority of women in cluster 1 have children, while we mostly find women without children in cluster 2.

| Cluster | n |
|---|---|
| 1 | 96 |
| 2 | 143 |



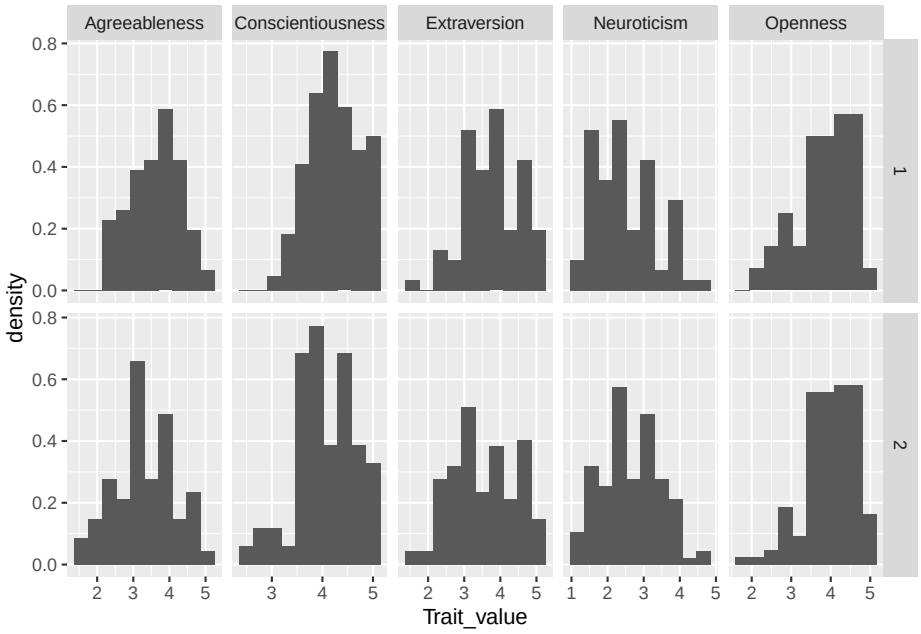

The transversal entropy of the sequences by cluster is displayed in the figure below.




```
## [1] 2
```

We can observe that the entropy decreases significantly for the cluster of women with children as compared to women without children, which means that the variability of the states for the first group. This can be interpreted as a sign of stability.

The following figure shows the distribution of the personality traits for the two clusters.




There seems to be differences in the distributions of agreeableness, conscientiousness and neuroticism between the clusters. In the following section we explore the predictive capacity of the distance matrix obtained via OM.

## k-Nearest Neighbors

Given a training set $\mathcal{D} = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ of $n$ labeled data points, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathcal{Y}$ (a finite set of class labels for classification or a continuous range of values for regression). $k$-NN provides a way to predict the label or value for a new, data point $x_{n+1}$ (for which $Y$ is unknown) by finding the $k$ training data points closest to $x_{n+1}$ and taking a majority vote of their labels (for classification) or averaging the values of $Y$ (for regression).

There are different ways of calculating the distance between the new data point $x_{n+1}$ and the points in $\mathcal{D}$. For instance, the Euclidean o Mahalanobis distances are usually used. In our case we already count with a matrix distance obtained with OM.

The choice of $k$ is a hyperparameter that can be tuned to optimize the performance of the $k$-NN algorithm. A larger $k$ reduces the effect of noise and outliers, but can also lead to overfitting. A smaller $k$ is more sensitive to noise and outliers, but can better capture local structure.

To compare the performance of different values of $k$, we use the mean squared error (MSE).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4}$$

where $y_i$ is the observed value and $\hat{Y}_i$ is the predicted valua via $k$NN.

In this part of the analysis we only consider the individuals who have available personality scores, that leaves us with a sample size of 200 individuals. We also split the data into two subsets: train (70%) and test (30%). We evaluate the MSE of the predictions for the individuals in the test set but only using the data from the nearest neighbors available in the train set.

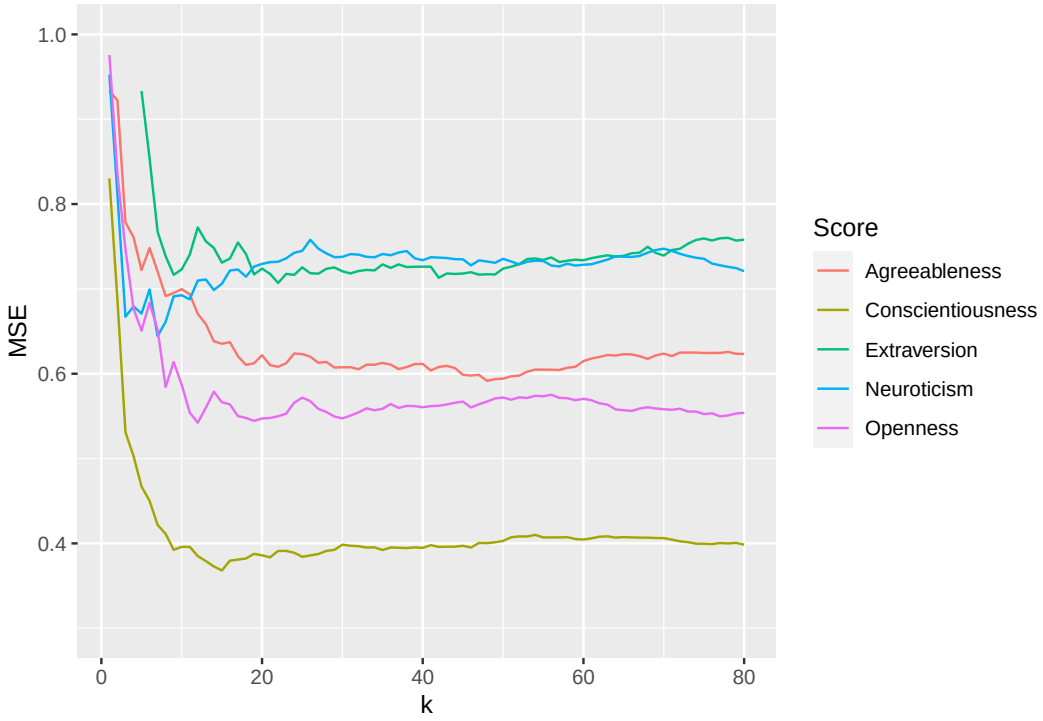
The following figure shows for every personality score and different values of $k$ the MSE, i.e. for $k = 1, \ldots, 80$ we predict values of $Y$ and compare them with the observed values using the MSE. As a reference, a red

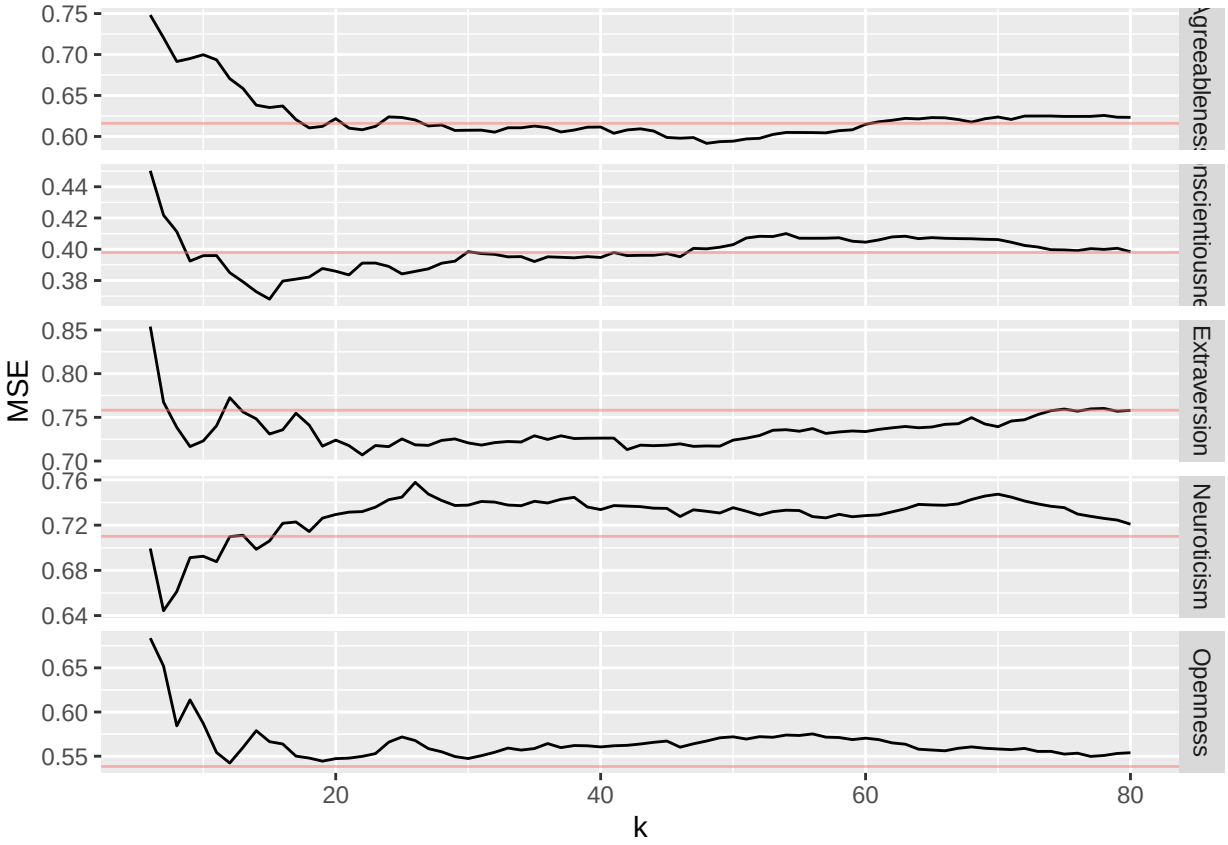
line for every personality trait is added to indicate the MSE of the trivial prediction, i.e. the prediction considering all the sample points in the train set.

For agreeableness, the MSE is minimum around $k = 50$ and increases again. However, this minimum is not considerably lower than the trivial prediction.

For openness, it is not very clear where the minimum MSE is and the prediction with kNN is always worse than the trivial prediction.

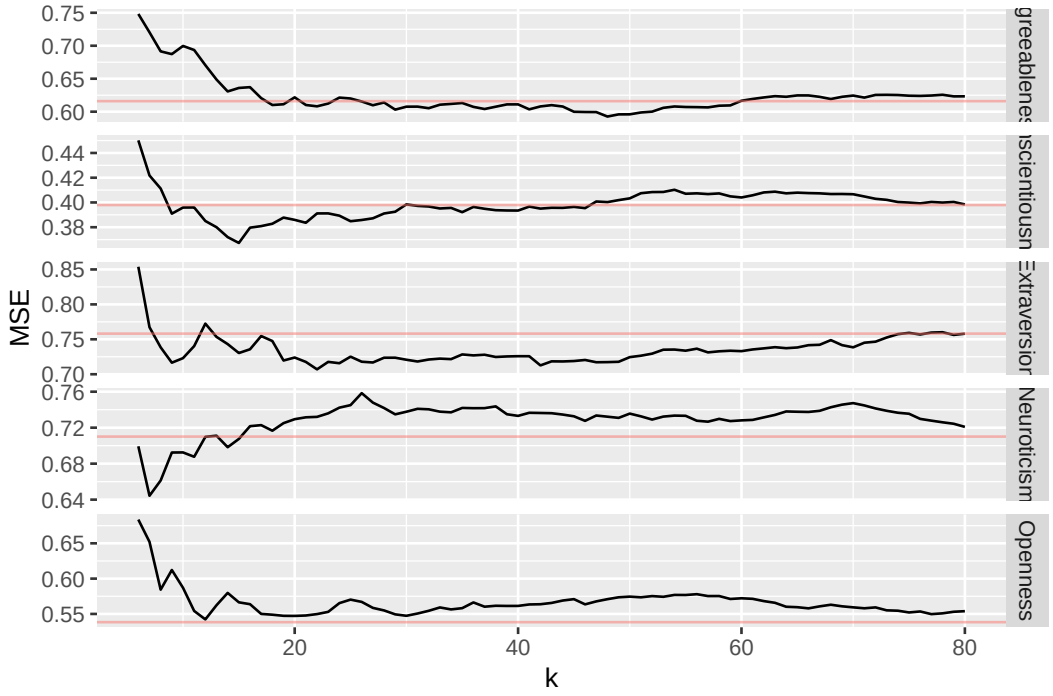
For conscientiousness, extraversion and neuroticism we observe that the MSE decreases as $k$ increases and takes a minimum value (around $k = 15$, $k = 10$, $k = 5$, respectively) thet is considerably lower than the trivial prediction and then the MSE increases again.



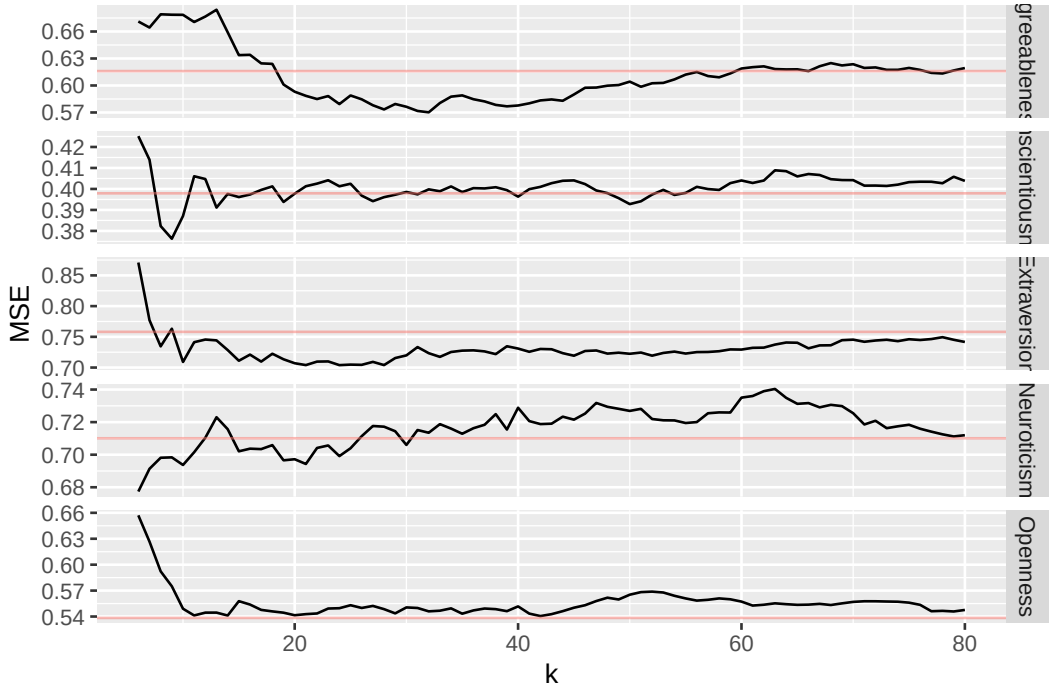



**Other experiments**

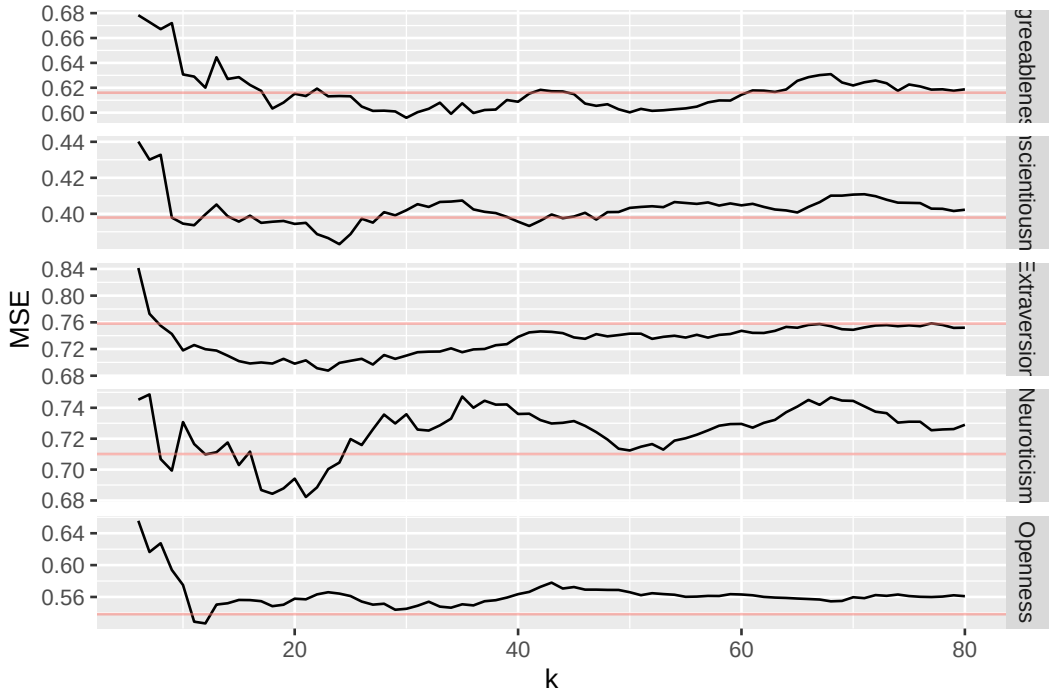
- Setting the constant $c$ as the maximum of $2 - P(s_i|s_j) - P(s_j|s_i)$ for the calculation of the cost matrix.



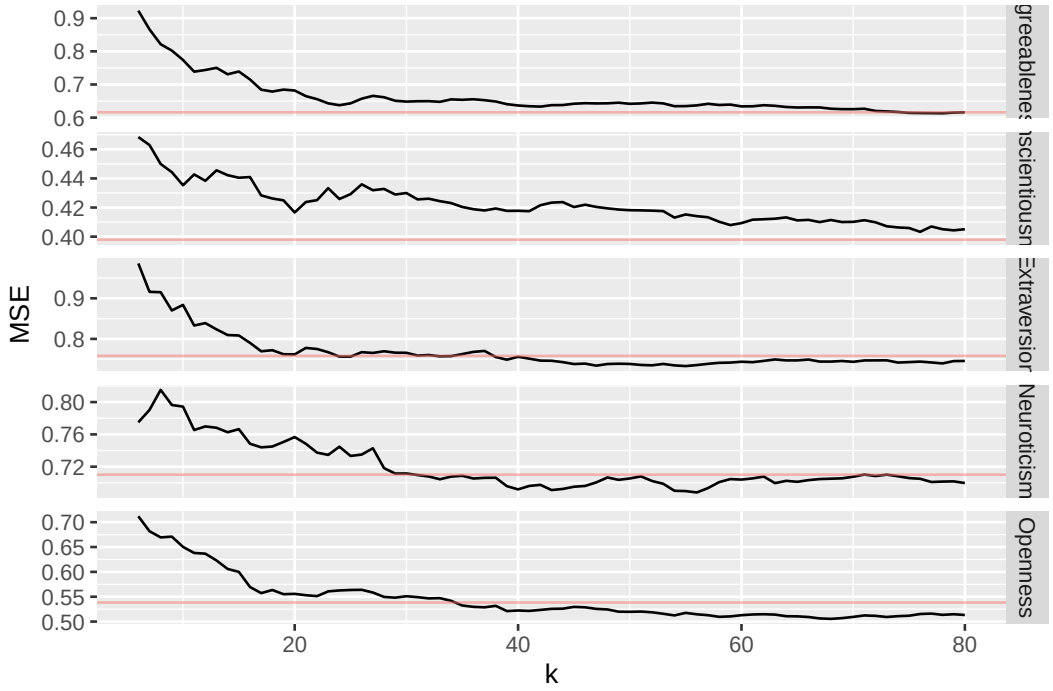
- Using `norm = "gmean"` normalization in `TraMineR::seqdist`.



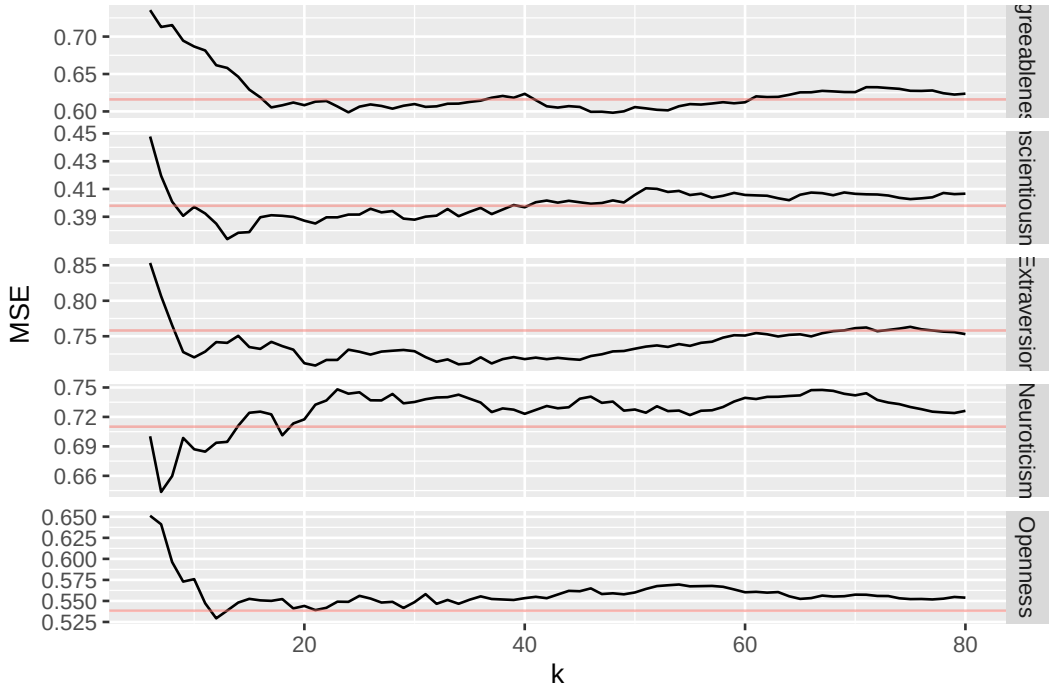

- Using `method = "FUTURE"` for the calculation of the cost matrix in `TraMineR::seqcost`.



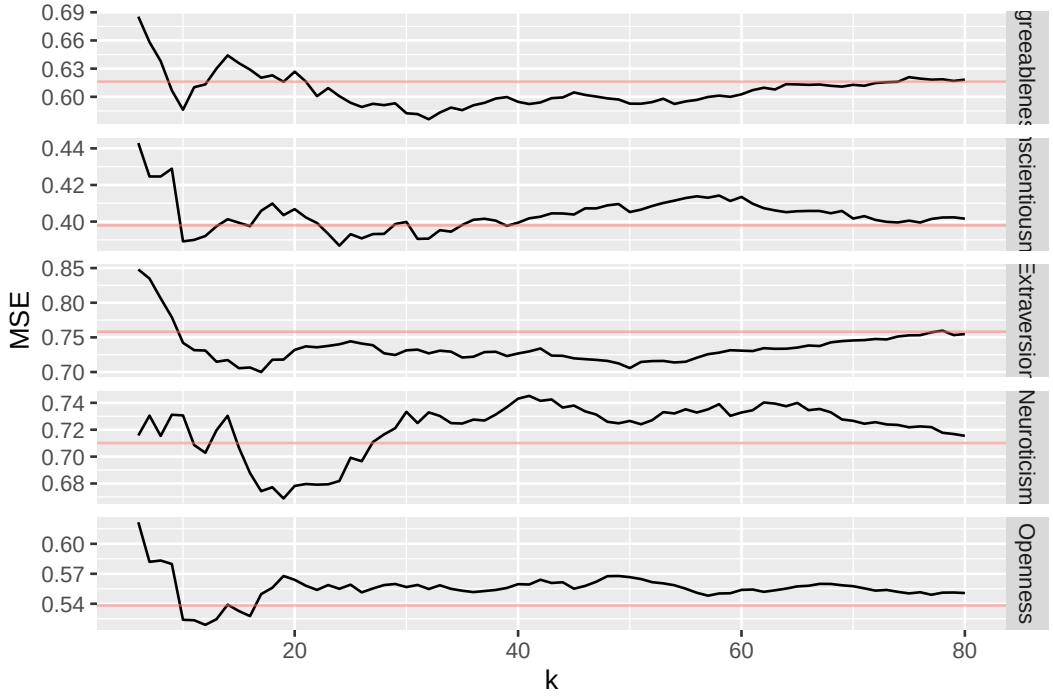

- Using `method = "INDELS"` for the calculation of the cost matrix in `TraMineR::seqcost`.




- Using `method = "INDELSLOG"` for the calculation of the cost matrix in `TraMineR::seqcost`.




- Using `method = "FUTURE"` for the calculation of the cost matrix in `TraMineR::seqcost` and `norm = "gmean"` normalization in `TraMineR::seqdist`.




- Using `method = "FUTURE"` and setting the cost of missing to a fixed value in the cost matrix in `TraMineR::seqcost` and `norm = "gmean"` normalization in `TraMineR::seqdist`.