A numerical analysis of cuisines across Europe

Rishabh Verma

12/3/2021

1 Introduction

I got my hands on a dataset describing the contents of the kitchens in 20 different European countries, published in 1975. It includes pantry items like soup tins, olive oil, and tea bags; frozen items like fish and vegetables; and refrigerated items like yogurt and butter.

The credit for this dataset goes to John Hartigan's database for clustering algorithms, and a proper citation is included at the end of this paper.

Some countries might be similar, and some countries might be not.

I can ask questions about this dataset based on which countries are similar. For example, which countries drink coffee, and which countries drink tea? Which countries keep lots of frozen items stocked? Which countries have access to fresh fruit?

This paper explores the feasibility of answering questions like these using dimensionality reduction and identifying clusters.

The dimensionality reduction algorithm I will use is called multi-dimensional scaling (MDS).

1.1 Data Description

This dataset describes the prevalence of 20 foodstuffs among 16 European countries.

The attributes include the name of the food, a two digit character code for the food, and 16 measurements of the prevalence of that food in a country's households.

All measurements of prevalence are percentage values between 0 and 100.

1.2 Data cleaning

This dataset is not "tidy." A single row corresponds to a food item, but we are interested in the pantry of each individual country. By transposing this dataset, each country and its measurements will occupy a row.

The cross-section below shows the first 5 food rows and first 10 country attributes in the original dataset.

```
head(data, 5)
```

```
## # A tibble: 5 x 18
##   Name   Code     WG    IT    FR    NS    BM    LG    GB    PL    AA    SD    SW
##   <chr>  <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1 groun~ GC       90    82    88    96    94    97    27    72    55    73    97
## 2 insta~ IC       49    10    42    62    38    61    86    26    31    72    13
## 3 tea b~ TB       88    60    63    98    48    86    99    77    61    85    93
## 4 sugar~ SS       19     2     4    32    11    28    22     2    15    25    31
## 5 packa~ BP       57    55    76    62    74    79    91    22    29    31     0
## # ... with 5 more variables: DK <int>, NY <int>, FD <int>, SP <int>, ID <int>
```

The cross-section below shows the first 5 country rows and first 4 food attributes in the tidied, i.e. transposed, dataset.

```
head(tidy_data, 5)
```

```
## # A tibble: 5 x 22
##   code  `ground coffee` `instant coffee` `tea bags` `sugarless sweets`
##   <chr>           <int>            <int>      <int>               <int>
## 1 WG                 90               49         88                  19
## 2 IT                 82               10         60                   2
## 3 FR                 88               42         63                   4
## 4 NS                 96               62         98                  32
## 5 BM                 94               38         48                  11
## # ... with 17 more variables: packaged biscuits <int>, packaged soup <int>,
## #   tinned soup <int>, instant potatoes <int>, frozen fish <int>,
## #   frozen vegetables <int>, fresh apples <int>, fresh oranges <int>,
## #   tinned fruit <int>, shop jam <int>, garlic clove <int>, butter <int>,
## #   margarine <int>, olive oil <int>, yogurt <int>, crispbread <int>,
## #   regions <fct>
```

Additionally, here is a map of country name to 2-digit country code.

WG	West Germany
IT	Italy
FR	France
NS	Netherlands
BM	Belgium
LG	Luxemburg
GB	Great Britain
PL	Portugal
AA	Austria
SD	Switzerland
SW	Sweden
DK	Denmark
NY	Norway
FD	Finland
SP	Spain
ID	Ireland

2 Methods

Since there are 20 measurements of foodstuffs made, each country's kitchen can be represented as a vector in $\mathbb{R}^{20}$.

Multi-dimensional scaling (MDS) is a flexible method of dimensionality reduction that may be useful for this dataset. To understand MDS, suppose you have a set of data-points $A \subset \mathbb{R}^n$. For each distinct pair of data-points $i, j \in A$ such that $i \neq j$, you can compute a real-valued distance between them. This distance could be computed using the norm of the vector space, i.e. $|i - j|$, but it doesn't have to be. It can be any distance function that makes sense to the user.

Given the resulting matrix of pairwise distances and a dimensionality parameter $k \in \mathbb{Z}^+$, the MDS problem is to approximate a set of coordinates in $\mathbb{R}^k$ whose pairwise Euclidean distances match the inputted pairwise distances.

The `stats` package in R provides a routine `cmdscale` which implements an algorithm for this problem.

Since all data values are percentages from 0 to 100, they are all "on the same scale." They do not need to be normalized for a meaningful computation of distance.

3 Building a model

3.1 Multi-Dimensional Scaling

Let's start simple. What happens if you compute the Euclidean distance between each point and pass that into MDS?

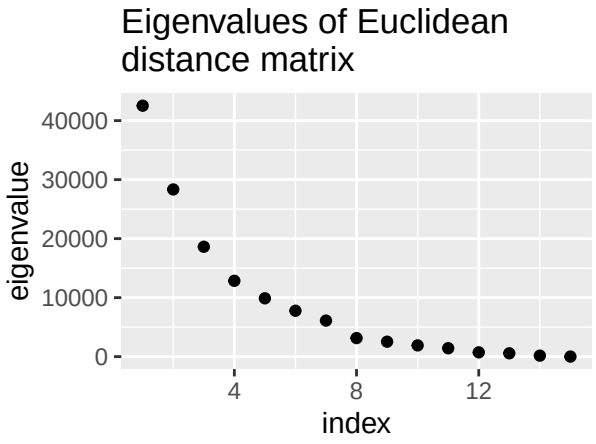
The `cmdscale` routine will compute the eigenvalues of the distance matrix, which I can analyze to decide how many dimensions I want to include in my model.

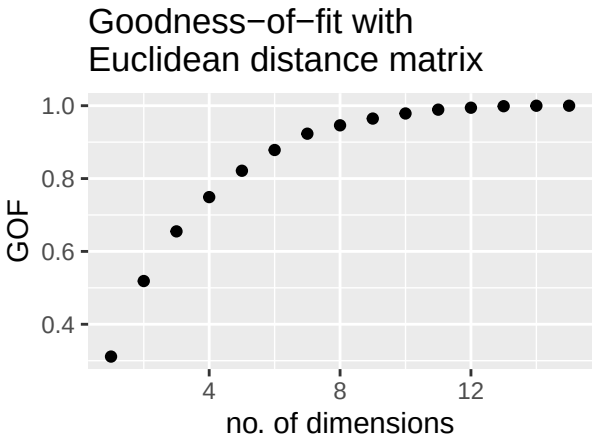
```
# Compute a matrix of Euclidean distances
distances <- dist(entries,
                  method="minkowski", p=2) %>%
  as.matrix()

# Perform MDS
model.1 <- cmdscale(distances, k=1, eig=TRUE)
model.2 <- cmdscale(distances, k=2, eig=TRUE)
model.3 <- cmdscale(distances, k=3, eig=TRUE)
```

In addition to plotting the eigenvalues, I can examine how passing a higher dimensionality argument results in a higher goodness-of-fit.

The output of `cmdscale` includes two values for goodness-of-fit. One value is computed using the absolute values of the eigenvalues of the distance matrix, and the other is computed using only the positive eigenvalues. The distance matrix has positive eigenvalues, so these values always agree.





The eigenvalues have a rather gradual decay, until they drop off sharply after the seventh eigenvalue. This suggests that we would only see diminishing returns after adding seven dimensions to the model, so this model may not perform well.

This is supported by the goodness-of-fit increasing with respect to number of dimensions at a similar sluggish pace. We want a model with a sharper change.

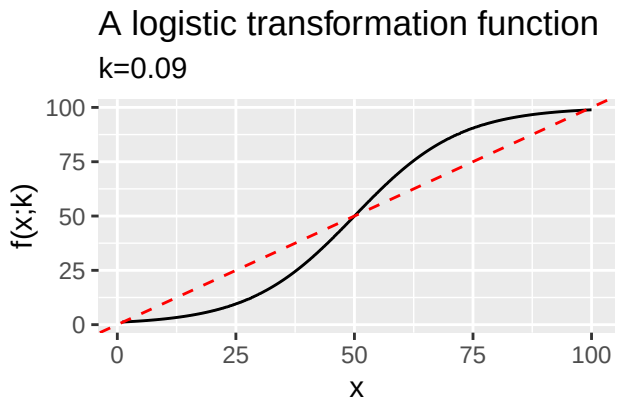
3.2 Building a competing model

What if I try transforming the data so that countries with the same extreme values are regarded as more similar? For example, a country with 80% prevalence of instant coffee will be regarded as closer to a country with 90% prevalence of instant coffee.

I can do this using a logistic function centered at $x = 50$ with a height of 100.

$$f(x;k) = \frac{100}{1 + e^{-k(x-50)}}$$

The value k controls the shape of this logistic curve. $k = 0.09$ yields the following function.



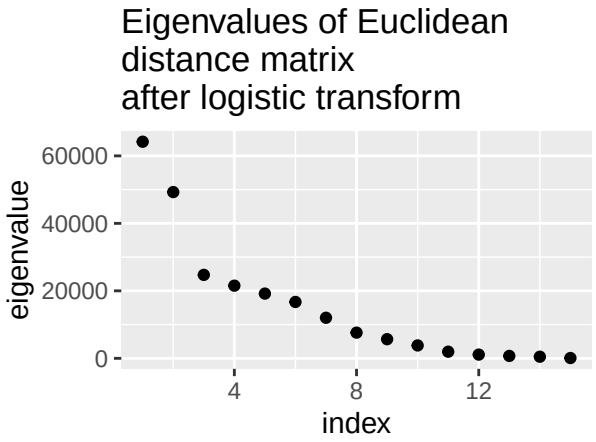
This function pulls the data apart toward the ends.

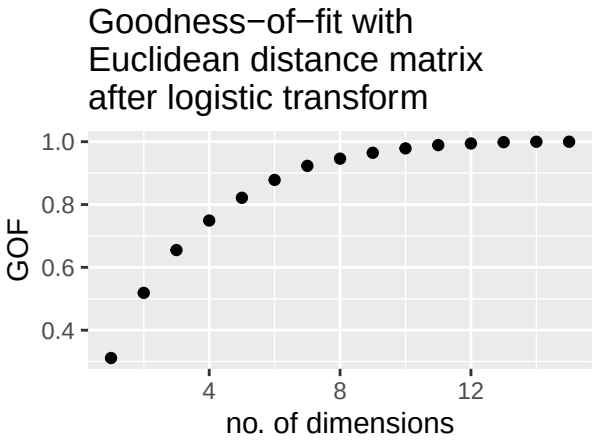
A tightly-clustered attribute at an extreme end will become even more tightly clustered, and contribute less variation to the distance calculation.

A spread-out attribute toward the center will become much more spread out, and will contribute more variation to the distance calculation.

A spread-out attribute that is firmly to one side will not be as significantly altered as one that is closer to the center.

Now let's try applying this function and re-analyzing the data.





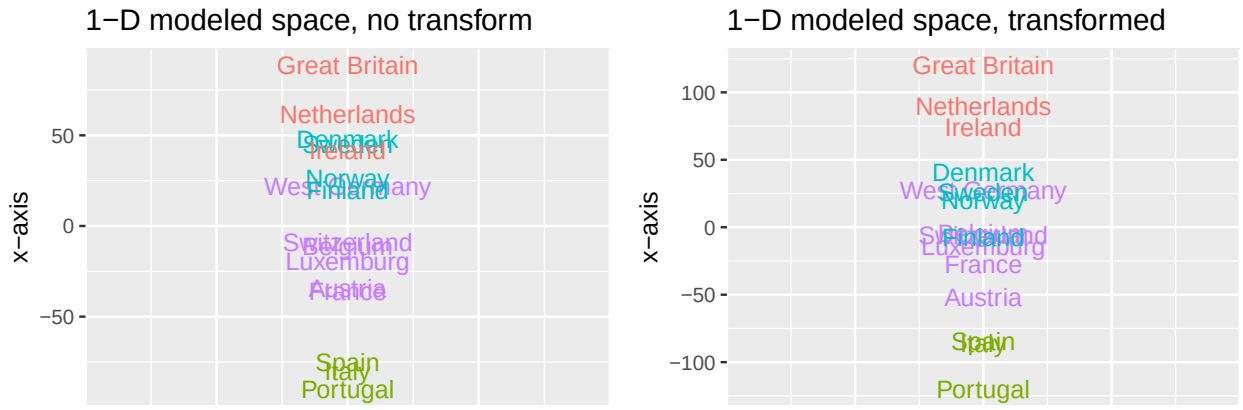
This eigenvalue plot presents two eigenvalues that are much more significant than the rest. This plot is still less than ideal since the eigenvalues still roll off gradually. A low-dimensional model will not capture all of the information in the data, but it's the best we can do.

Transformation brings the GOF of the 2-dimensional model from 0.519 to 0.495. This is not a significant change.

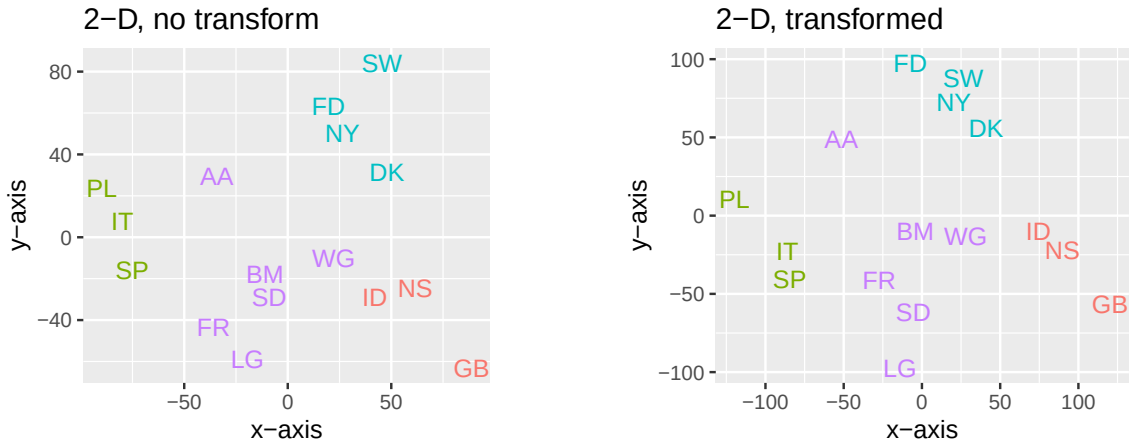
All in all, it seems that the logistic transformation will make for a better 2-D model based off of the eigenvalues.

3.3 The spaces from four different models

The one-dimensional spaces created by both models are:



The two-dimensional spaces created by both models are:



Comparing the entries of the 2-D models and the 1-D models reveals that adding a second dimension does not change the modeled coordinates in the first dimension. This means the x-axis of the 2-D graphs are the exact same as the vertical x-axis in the 1-D graphs.

I am not an expert in European cultures, but I can at least pull out three cultural groups using prior knowledge. The English-adjacent countries are classified together since their local languages are West Germanic cousins: Scots, English, Frisian, and Dutch.

- Scandinavian countries (blue): Norway, Finland, Sweden, Denmark
- Mediterranean countries (green): Spain, Portugal, Italy
- English-adjacent countries (red): Great Britain, Ireland, the Netherlands

I have left the rest unclassified (purple) because again, I do not know enough to make any finer classifications.

In the 1-D model, you can pick out the English countries and the Mediterranean countries. The Scandinavian countries are not distinct.

The 2-D space captures a lot more nuance. Adding a second dimension pulls Scandinavia apart from the other European countries.

The logistic transform does not have a dramatic effect on the spaces formed, but since its eigenvalue plot looks better, I will proceed using the models formed after using the logistic transform.

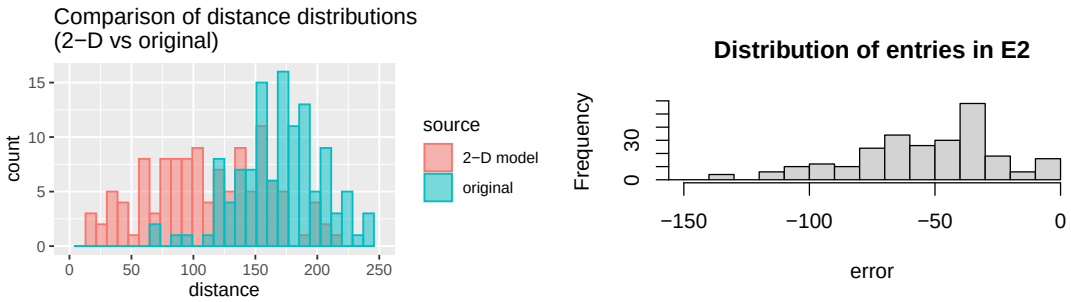
4 Analyzing the model

4.1 2 dimensions vs 3 dimensions

The distance matrix of the model's fitted points is supposed to approximate the inputted distance matrix.

Let D be the original distance matrix. Let D_2 be the modeled distance matrix using $k = 2$ dimensions.

The left histogram below compares the distribution of the non-zero distances in D and in D_2. It shows that the modeled coordinate system yields a distance matrix D_2 that underapproximates the entries of D by quite a lot.



The right histogram above examines the distribution of the entries in the error matrix, $E_2 = D_2 - D$. It shows that all of the errors are in fact non-positive. This is why the red histogram is to the left of the blue one. It then follows that the modeled distances are all too short; the points are too close together.

Thus, the entries in E_2 with the greatest absolute value represent points that are far away in $\mathbb{R}^{20}$, but the dimensionality reduction from MDS squishes them together.

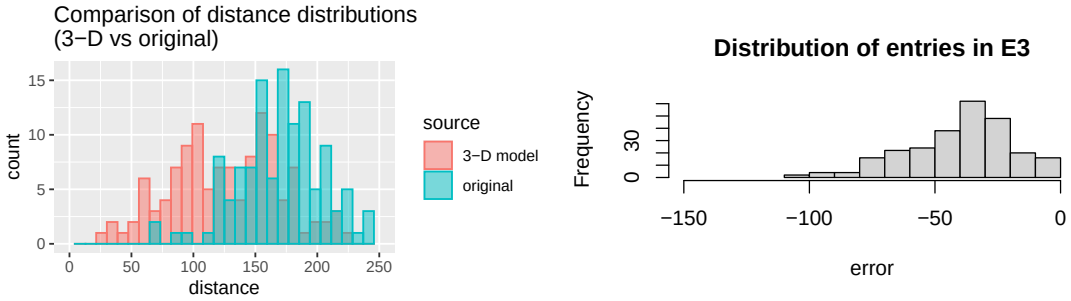
The distribution of the errors in the right figure shows that there is one outlier*, and a few more large values. Analysis of the error values more significant than -90 reveals two things:

1. Ireland should be further from West Germany, Netherlands*, Belgium, Luxemburg, and Denmark.
2. Sweden should be further away from Denmark, Norway, and Finland.

What if we add a third dimension? Let's dive into $E_3 = D_3 - D$.

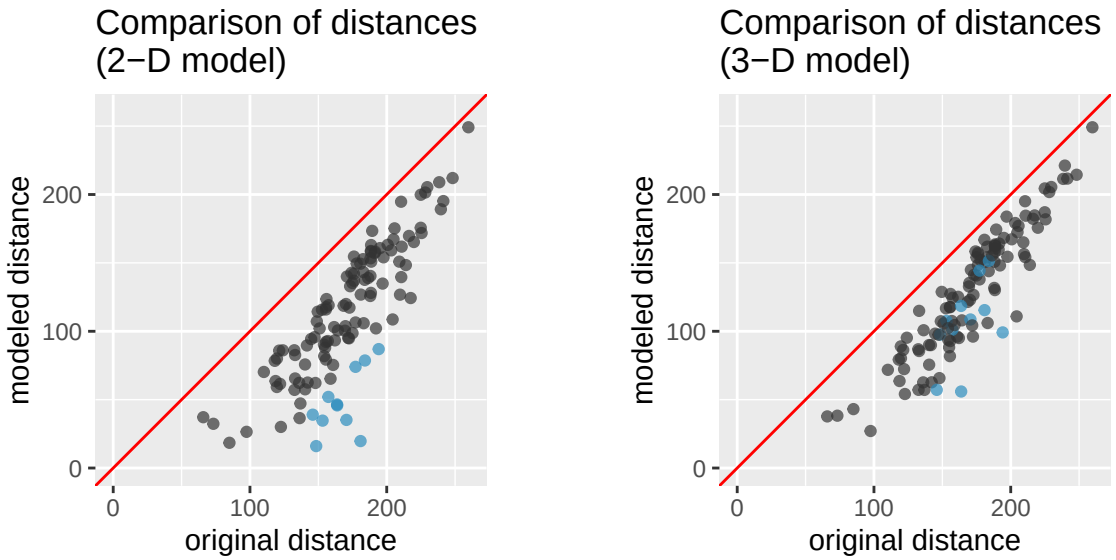
On the left, you can see the modeled distance histogram D_3 shifts a little closer to the original distance histogram D.

On the right, you can see the distribution of errors in E_3 shifts closer to zero.



For dimensions $k = 2, 3$, the previous section operated by computing $E_k = D_k - D$ and analyzing the entries of E_k with a histogram.

I can also create scatterplots directly comparing the entries of D_k and D. In these scatterplots, the vertical distance from any point and the red line represents the corresponding error value in E_k.



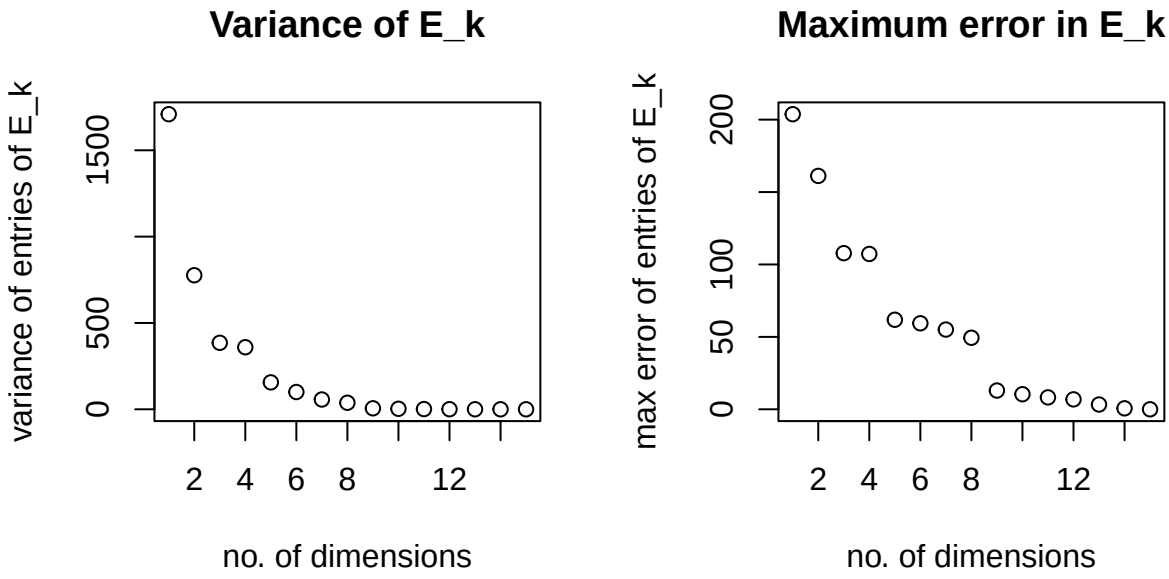
I colored a dozen of the points with the highest errors in E_2 blue. They're all sitting together in the left scatterplot. Speaking roughly, these points all have an original distance around $[150, 190]$ and a 2-D modeled distance of only $[10, 90]$, but adding the third dimension pushes this group way up in the right scatterplot to $[50, 150]$, which is considerably better.

Again, these points represent a small set of pairwise distance relations that really need a third dimension to be captured, mostly regarding Ireland or Sweden.

4.2 What about more dimensions?

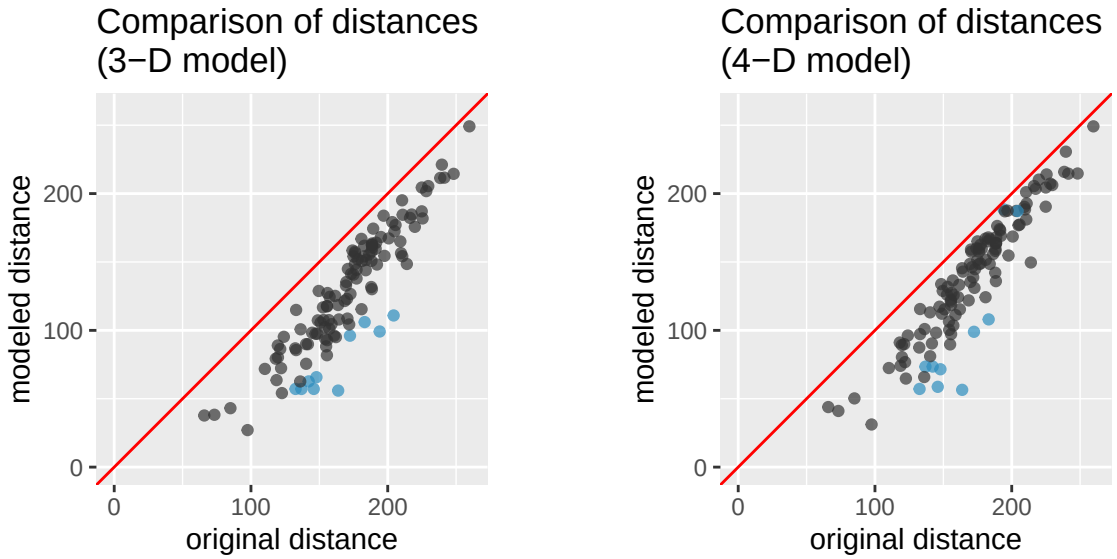
How does this improve with more dimensions? For $k = 1, 2, ..., 15$, let's compute and plot the variance of the entries of $E_k = D_k - D$.

Recall that since we are modeling 16 points in $\mathbb{R}^{20}$, using a number of dimensions anywhere close to 15 is not very meaningful.



There's a bit of a staircase pattern. This plot suggests that going from 2-D to 3-D is very useful, but going from 3-D to 4-D does not seem to affect the error matrix much.

Now I'm just too curious. What if I replicate the scatterplots, but comparing 3-D and 4-D? The 3-D scatterplot is already pretty good, so how much improvement could the 4-D scatterplot yield? This time I've colored in the dozen points with the highest errors in E_3 to see where they go.



Yeah, not very much improvement with 4 dimensions.

5 Using the model to analyze the data

5.1 Investigating the outliers

5.1.1 Ireland and the Netherlands

In section 4.1, I used the histograms of E_2, E_3 and the scatterplots of D_2 against D, D_3 against D to identify some outliers in the 2-D model that were better represented in the 3-D model.

Let's dive into those outliers and see what we can learn.

The most significant outlier in E_2 is due to Ireland and the Netherlands. Just how different can they be? Apparently they differ by 50 percentage points on three different foodstuffs.

```
data %>%
  select(Name, ID, NS) %>%
  mutate(diff = ID - NS) %>%
  filter(abs(diff) > 50)
```

```
## # A tibble: 3 x 4
##   Name             ID    NS  diff
##   <chr>         <int> <int> <int>
## 1 ground coffee    13    96   -83
## 2 butter           97    31    66
## 3 margarine        25    97   -72
```

Ireland uses butter whereas the Netherlands use margarine. The Netherlands also keep a lot more ground coffee and yogurt.

5.1.2 Sweden and Scandinavia

Why should Sweden be further away from the rest of Scandinavia? Let's dive into the entries and see where Sweden differs from the average of its peers by 30 percentage points.

```
data %>%
  select(Name, SW, DK, NY, FD) %>%
  mutate(average = (DK+NY+FD)/3) %>%
  filter(abs(SW-average) > 30) %>%
  select(-average)
```

```
## # A tibble: 4 x 5
##   Name                SW    DK    NY    FD
##   <chr>            <int> <int> <int> <int>
## 1 packaged biscuits     0    66    62    64
## 2 tinned soup          43    17     4    10
## 3 margarine            32    91    94    94
## 4 crispbread           93    34    62    64
```

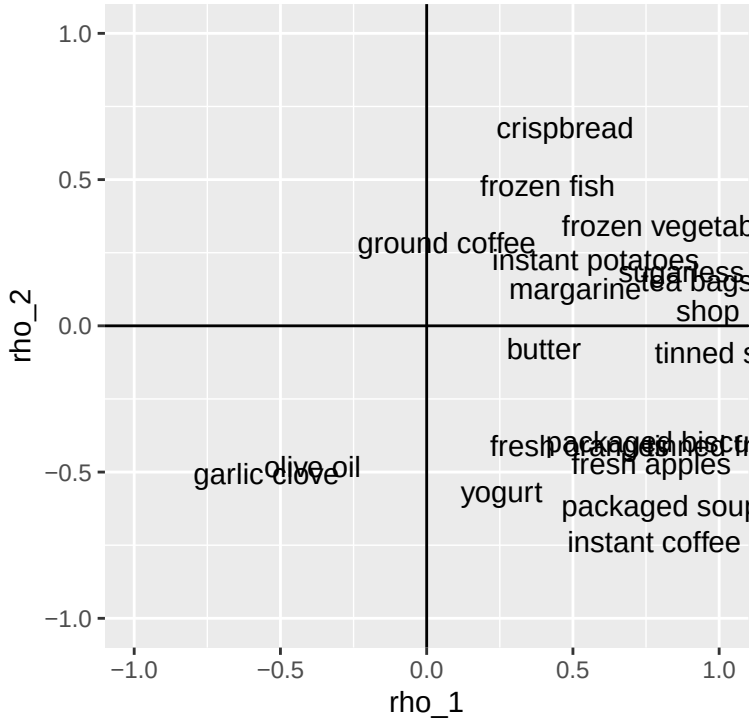
It is reported that 0% of Swedish households have packaged biscuits, but roughly 64% of their Scandinavian peers do. The total absence of packaged biscuits in Sweden is **suspicious**. There may be missing information.

I should also note that Sweden also seems to consume a lot more tinned soup, less margarine, and more crispbread than its peers.

5.2 Finding trends

Let's look at just the 2-D logistic-transformed model. What do the two coordinates mean?

I can investigate this by iterating through each attribute, computing its correlation with the two coordinate systems, and plotting the results on a scatterplot.

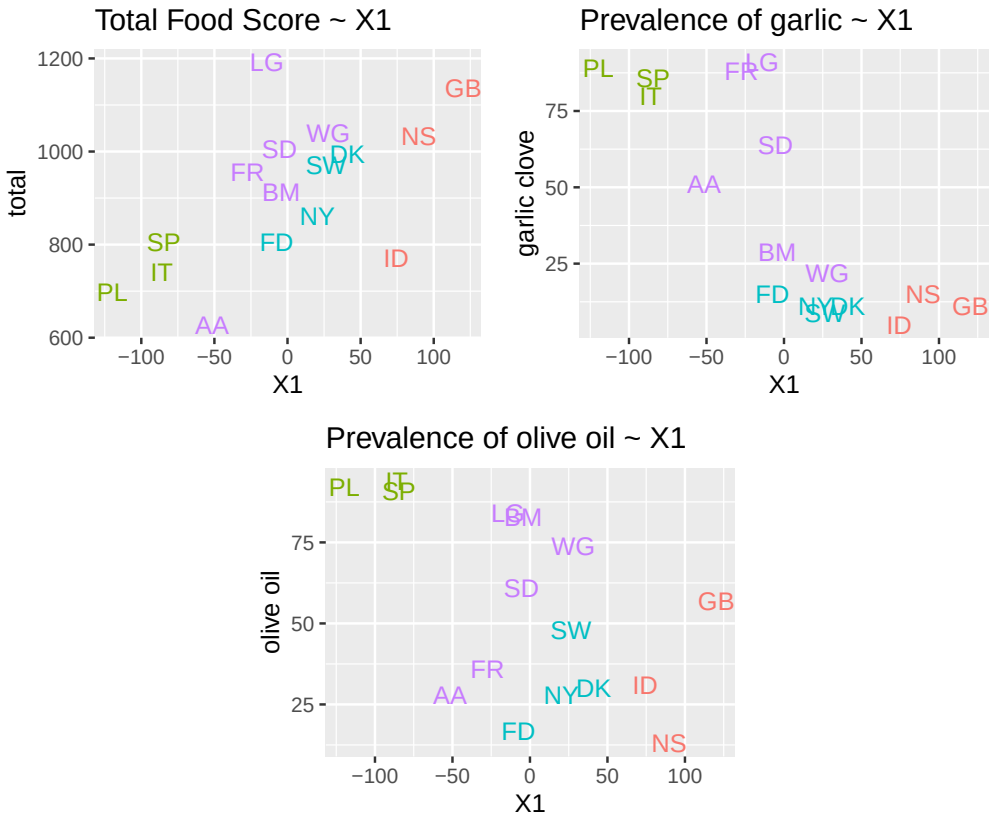


5.2.1 The first coordinate axis

The first coordinate is negatively correlated with the use of olive oil and garlic, and is positively correlated with all other foods. Why is that?

Out of 20 foods, 18 are positively correlated with x_1. Perhaps x_1 just measures the diversity of foods within a pantry in a single country. Suppose I were to iterate through each country and sum up the elements of the corresponding vector in $\mathbb{R}^{20}$. A high sum means that in that country, lots of households have lots of different types of foods. A low sum might mean that most households have a less diverse set of foods stocked.

But what about the olive oil and garlic?



Countries with a low total food score include the Mediterranean countries, Finland, Ireland, and Austria.

Countries with a high prevalence of garlic cloves include the Mediterranean countries, France, and Luxembourg.

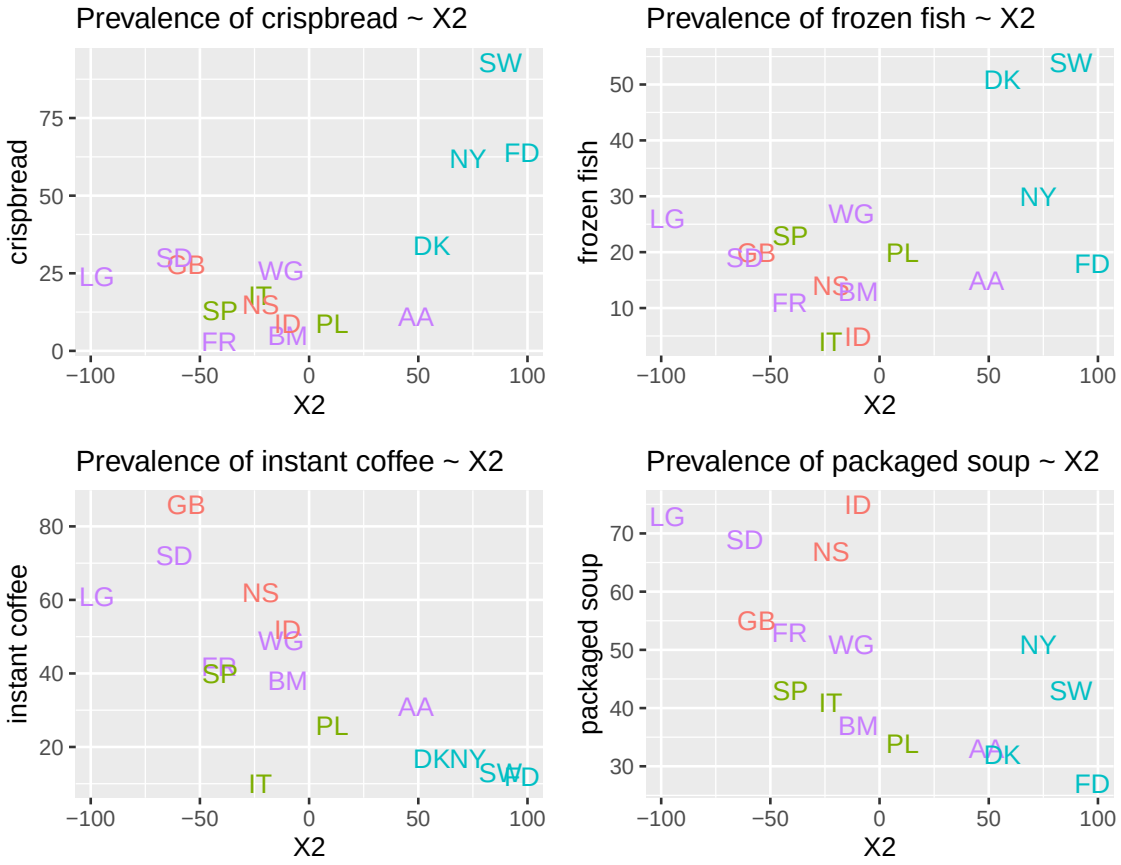
Countries with a high prevalence of olive oil include the Mediterranean countries, Luxembourg, and Belgium.

Notice a pattern? I think x_1 simultaneously measures Mediterranean-ness and diversity of foodstuffs. These two patterns are *confounded* with each other, and the model does its best to represent them both with x_1.

5.2.2 The second coordinate axis

The second coordinate is positively correlated with crispbread and frozen foods, and is negatively correlated with instant coffee and packaged soup.

According to Wikipedia, crispbread originates from Scandinavia. Come on, *come on*, let's put our thinking caps on.



Dare I say the second axis measures Scandinavian-ness? This is consistent with the fact that in the physical spaces modeled by MDS, Scandinavia needed the second dimension to be distinct from the central Europe.

This attests to the cultural distinctness of Scandinavia from the rest of Europe.

6 References

Data credit:

John Hartigan,
Clustering Algorithms,
Wiley, 1975.
ISBN 0-471-35645-X
LC: QA278.H36
Dewey: 519.5'3

Data retrieved from https://people.sc.fsu.edu/ jburkardt/datasets/hartigan/hartigan.html

7 Code

This document was generated from an RMarkdown notebook.

The markdown code and R chunks used to generate this document can be found at https://github.com/vermarish/european-cuisine.