**Multidimensional Scaling (MDS) for Analyzing Perception Data** Ryan Lidster
Workshop at Pronunciation in Second Language Learning and Teaching (PSLLT) September 2018

**Definition** of MDS: a means of *ordinating* (i.e. creating categories and clines) and visualizing data by taking potentially complex information and arranging it into a set of points in *n*-dimensional space.

- 1-dimensional space: a line, e.g. a number line.
- 2-dimensional space: a plane / a surface
- 3-dimensional space: a volume, e.g. a cube
- 4- or higher dimensional space: no physical analog; can still be analyzed mathematically, but interpretation is more challenging, and in practice, a high-dimensional model is used *very* rarely

- ***Not a statistical test***. Alone, it does not test hypotheses, but can be useful for argumentation
- ***Primarily for "distance" data***
    - Concrete, real-world distances → MDS is almost purely a means of visualizing data
    - Abstract, psychological distances → MDS can also aid in interpreting data

**Distances in L2 pronunciation (distances in perceptual space)**

- Psychological closeness → difficulty in discrimination, inaccuracy in identification against competitors, fuzziness in lexical representation
- Psychologically disparate → ease in discrimination, high identification, more exclusive lexical representations

Perceptual space is warped by language experience, so sometimes it is of key concern to find out how target language sounds or utterances are perceived. MDS could also be used on a wide variety of data types:

- L2 learners' perception of the relative salience of segmental, suprasegmental, or even indexical distinctions
- L1 listeners' perception of L2 accented speech as "closer to" or "farther away" from native speakers, or from each other in terms of "groups of accents" (also, e.g. for L2 learners perceiving different dialects)

Eventual data set required is a "**dissimilarity matrix**"

- Pairwise distances between all (or almost all) stimuli. With a larger number of stimuli, some pairs can be missing and it will still be possible to impute distances, but with more than 5% missing in data with large numbers of stimuli (i.e. >25), or with even a small number of missing points in a smaller data set, higher dimensional solutions will not be uniquely computable

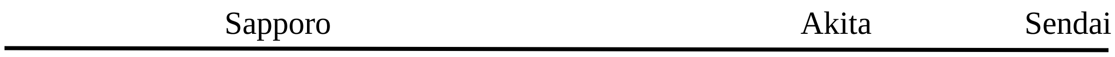
| | Stimulus 1 | Stimulus 2 | Stimulus 3 | Stimulus 4 |
|---|---|---|---|---|
| Stimulus 1 | 0 | Distance between 1 and 2 | Distance between 1 and 3 | Distance between 1 and 4 |
| Stimulus 2 | Distance between 1 and 2 | 0 | Distance between 2 and 3 | Distance between 2 and 4 |
| Stimulus 3 | Distance between 1 and 3 | Distance between 2 and 3 | 0 | Distance between 3 and 4 |
| Stimulus 4 | Distance between 1 and 4 | Distance between 2 and 4 | Distance between 3 and 4 | 0 |

- The diagonal of the dissimilarity matrix will be 0 because the distance of a stimulus to itself is 0.
- Typically, the matrix should be "square symmetric" (i.e. cell [1,2] is equal to [2,1], and so on). Sometimes, however, the perceived distance might depend on order or anchor effects. "A" compared to the standard of "B" might genuinely be different from "B" compared to the standard of "A." That is doable in MDS, but that analysis is much more complex and typically not done in psychology.

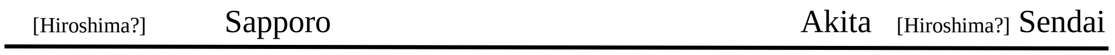
## Warm-up Example: Flight Distances in Japan

| | Sapporo | Akita | Sendai | Tokyo | Nagoya | Kyoto | Osaka | Hiroshima | Kagoshima | Naha |
|---|---|---|---|---|---|---|---|---|---|---|
| Sapporo | 0 | 456.98 | 557.48 | 901.03 | 968.67 | 1011.51 | 1122.15 | 1263.8 | 1640.04 | 2261.27 |
| Akita | 456.98 | 0 | 129.63 | 446.28 | 527.58 | 588.11 | 691.34 | 880.59 | 1242.5 | 1852.75 |
| Sendai | 557.48 | 129.63 | 0 | 349.79 | 486.48 | 564.82 | 656.24 | 877.38 | 1222.87 | 1820.25 |
| Tokyo | 901.03 | 446.28 | 349.79 | 0 | 253.69 | 357.37 | 399.66 | 672.23 | 960.4 | 1522.92 |
| Nagoya | 968.67 | 527.58 | 486.48 | 253.69 | 0 | 103.69 | 170.06 | 422.73 | 740.26 | 1333.92 |
| Kyoto | 1011.51 | 588.11 | 564.82 | 357.37 | 103.69 | 0 | 113.83 | 323.38 | 658.57 | 1264.65 |
| Osaka | 1122.15 | 691.34 | 656.24 | 399.66 | 170.06 | 113.83 | 0 | 280.86 | 571.14 | 1164.8 |
| Hiroshima | 1263.8 | 880.59 | 877.38 | 672.23 | 422.73 | 323.38 | 280.86 | 0 | 378.09 | 1001.07 |
| Kagoshima | 1640.04 | 1242.5 | 1222.87 | 960.4 | 740.26 | 658.57 | 571.14 | 378.09 | 0 | 622.98 |
| Naha | 2261.27 | 1852.75 | 1820.25 | 1522.92 | 1333.92 | 1264.65 | 1164.8 | 1001.07 | 622.98 | 0 |

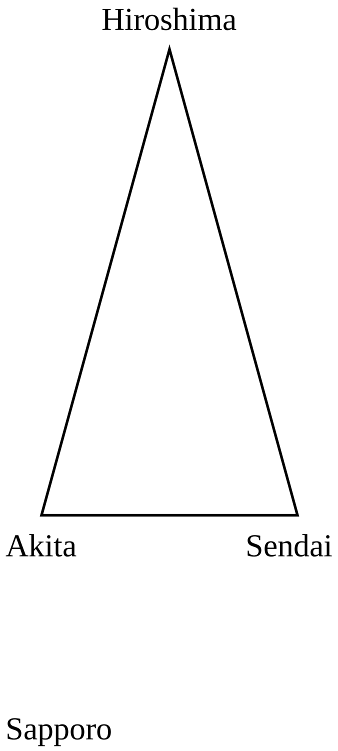
First, we have to decide how many dimensions are necessary for the data. If everything were lined up in a row, then 1 dimension would be enough, but this is rarely the case.



Example: Akita-Sendai is 129km.  But Hiroshima-Akita vs Hiroshima-Sendai is only 3km different. That can’t exist on a 1-dimensional line.

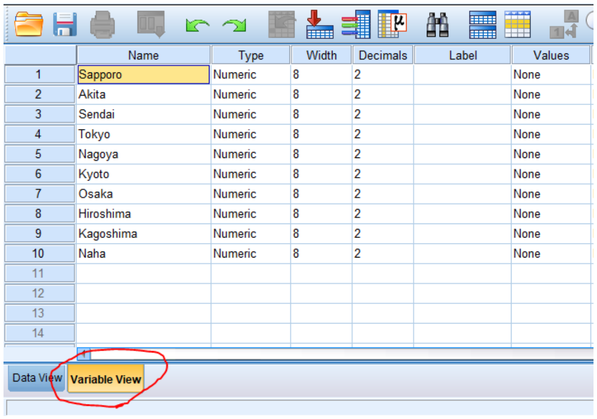


There’s no way to fit all of the points on a single straight line in a way that reproduces the distances from the dissimilarity matrix. If instead of a line, we allowed there to be a 2-dimensional space, then we could create a triangle, and the relationship of distances could be recreated faithfully. E.g.:



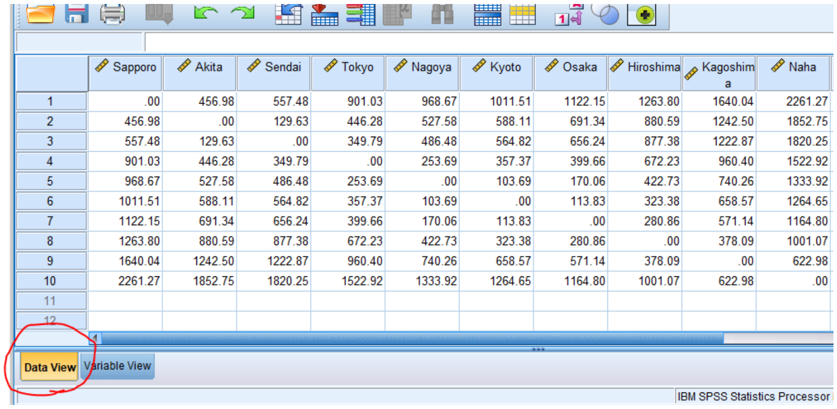
## SPSS Walkthrough:

Stimulus names go in variable names in Variable view



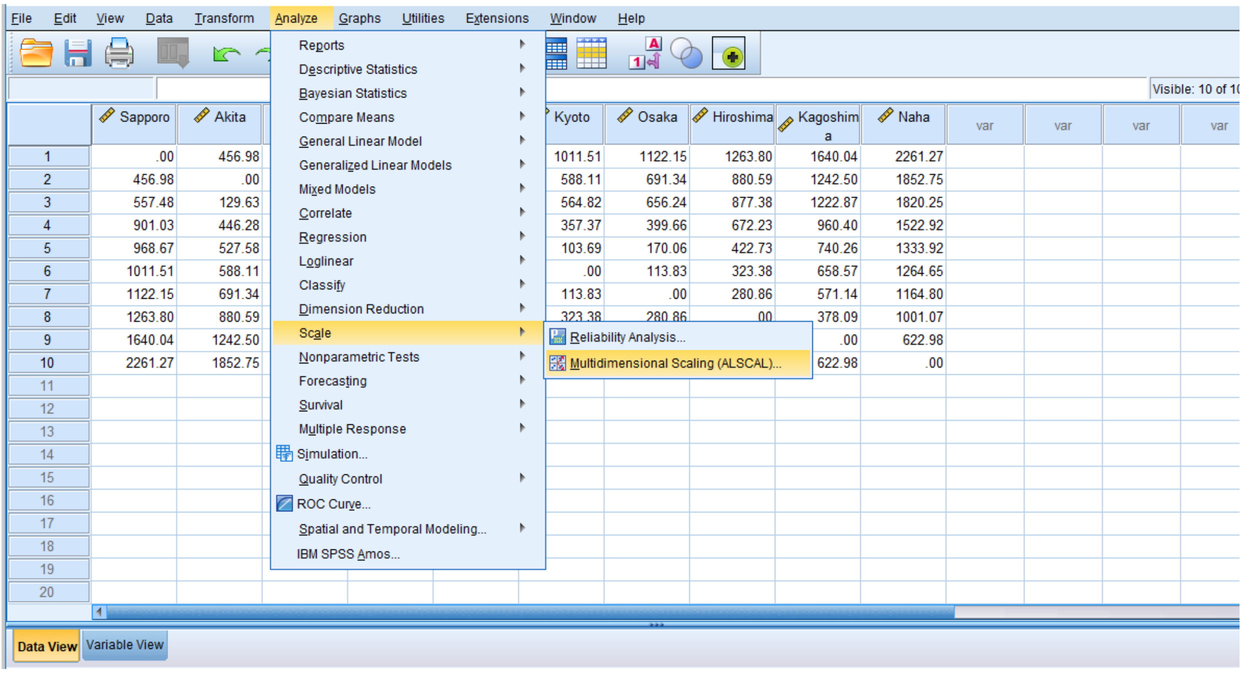
|  | Name | Type | Width | Decimals | Label | Values |
|---|---|---|---|---|---|---|
| 1 | Sapporo | Numeric | 8 | 2 |  | None |
| 2 | Akita | Numeric | 8 | 2 |  | None |
| 3 | Sendai | Numeric | 8 | 2 |  | None |
| 4 | Tokyo | Numeric | 8 | 2 |  | None |
| 5 | Nagoya | Numeric | 8 | 2 |  | None |
| 6 | Kyoto | Numeric | 8 | 2 |  | None |
| 7 | Osaka | Numeric | 8 | 2 |  | None |
| 8 | Hiroshima | Numeric | 8 | 2 |  | None |
| 9 | Kagoshima | Numeric | 8 | 2 |  | None |
| 10 | Naha | Numeric | 8 | 2 |  | None |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |

Dissimilarity matrix goes in Data

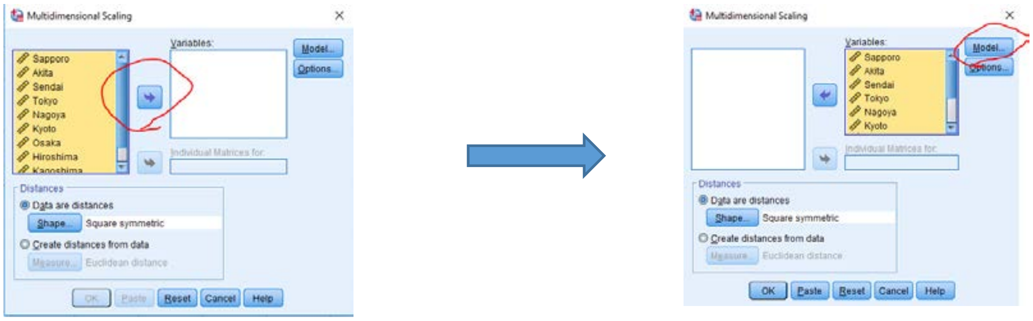


|  | Sapporo | Akita | Sendai | Tokyo | Nagoya | Kyoto | Osaka | Hiroshima | Kagoshima | Naha |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | .00 | 456.98 | 557.48 | 901.03 | 968.67 | 1011.51 | 1122.15 | 1263.80 | 1640.04 | 2261.27 |
| 2 | 456.98 | .00 | 129.63 | 446.28 | 527.58 | 588.11 | 691.34 | 880.59 | 1242.50 | 1852.75 |
| 3 | 557.48 | 129.63 | .00 | 349.79 | 486.48 | 564.82 | 656.24 | 877.38 | 1222.87 | 1820.25 |
| 4 | 901.03 | 446.28 | 349.79 | .00 | 253.69 | 357.37 | 399.66 | 672.23 | 960.40 | 1522.92 |
| 5 | 968.67 | 527.58 | 486.48 | 253.69 | .00 | 103.69 | 170.06 | 422.73 | 740.26 | 1333.92 |
| 6 | 1011.51 | 588.11 | 564.82 | 357.37 | 103.69 | .00 | 113.83 | 323.38 | 658.57 | 1264.65 |
| 7 | 1122.15 | 691.34 | 656.24 | 399.66 | 170.06 | 113.83 | .00 | 280.86 | 571.14 | 1164.80 |
| 8 | 1263.80 | 880.59 | 877.38 | 672.23 | 422.73 | 323.38 | 280.86 | .00 | 378.09 | 1001.07 |
| 9 | 1640.04 | 1242.50 | 1222.87 | 960.40 | 740.26 | 658.57 | 571.14 | 378.09 | .00 | 622.98 |
| 10 | 2261.27 | 1852.75 | 1820.25 | 1522.92 | 1333.92 | 1264.65 | 1164.80 | 1001.07 | 622.98 | .00 |
| 11 |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |

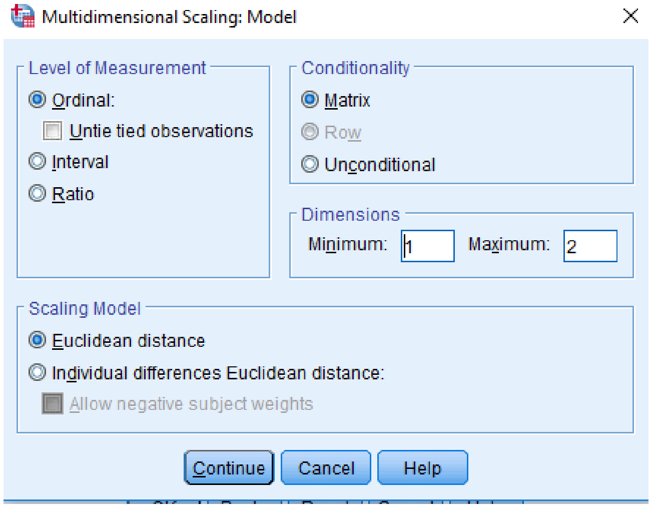
Analyze -> Scale -> Multidimensional Scaling



Select all the variables (stimuli) you want to include in the analysis, and then the model you'll use



**Model options**:


Multidimensional Scaling: Model

Level of Measurement
● Ordinal:
□ Untie tied observations
○ Interval
○ Ratio

Conditionality
● Matrix
○ Row
○ Unconditional

Dimensions
Minimum: 1 Maximum: 2

Scaling Model
● Euclidean distance
○ Individual differences Euclidean distance:
□ Allow negative subject weights

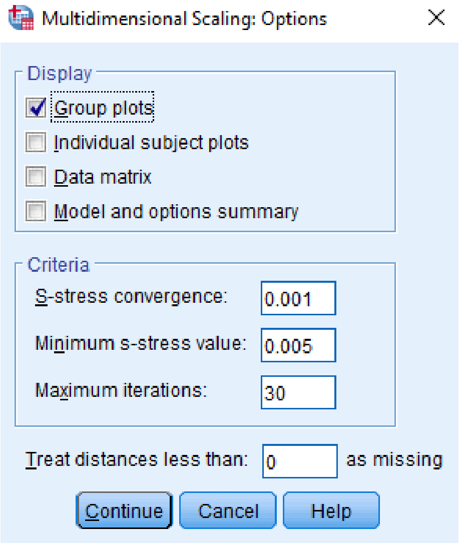
"Level of measurement" → Even if using ratio data, selecting ordinal will not reduce the meaningfulness of the result. There are almost no conceivable situations where L2 researchers would use anything other than ordinal assumptions.

"Conditionality" → Also leave as "Matrix"

"Dimensions" → In this particular case, we have a very good idea of what the dimensionality (2) will be, but when trying to figure out dimensionality, you can try out multiple dimensions and see what the indices of fit are to decide on the most appropriate model

"Scaling Model" → Select "Euclidean distance." An "Individual differences" model requires a very different data setup that is beyond this workshop, but doing so is possible in SPSS

**Output:**


Multidimensional Scaling: Options

Display
☑ Group plots
□ Individual subject plots
□ Data matrix
□ Model and options summary

Criteria
S-stress convergence: 0.001
Minimum s-stress value: 0.005
Maximum iterations: 30

Treat distances less than: 0 as missing

"Group plots" → Visually displays the output in SPSS. This is not always desirable since SPSS is limited in visual displays, but it can help to get a general idea.

"Individual subject plots" → Only useable for individual (INSCAL) analysis, on a different type of data set

"Data matrix" → Displays distances in the scaled matrix or matrices you create. However, **with only an "ordinal" assumption, this will only give relative distances, so particular data set will not often be meaningful, and instead it's often more useful to calculate distances between points.**

"Model and options summary" → Prints the settings you selected in the SPSS Output file

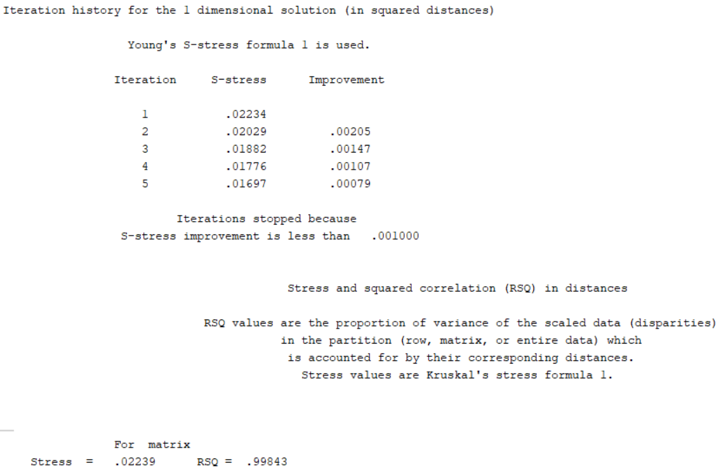
"S-stress convergence" → MDS creates a set of points in space. The distances between those points should match up to the distances in the original input matrix, but that might not be possible in the model-specified number of dimensions. The degree to which the original input distances and the MDS output distances diverge is called "stress," and in general, stress is bad and should be minimized.

MDS algorithms work iteratively. They produce a first pass estimation, then calculate the divergence from the original matrix, make an adjustment, and then recalculate the degree of divergence, adjust again, and so on. The level of s-stress convergence specifies how big of an improvement you need to make at each iteration in order to continue. After a new iteration makes less improvement than the convergence criterion, it will stop. In general, there are not many reasons to change it from 0.001.

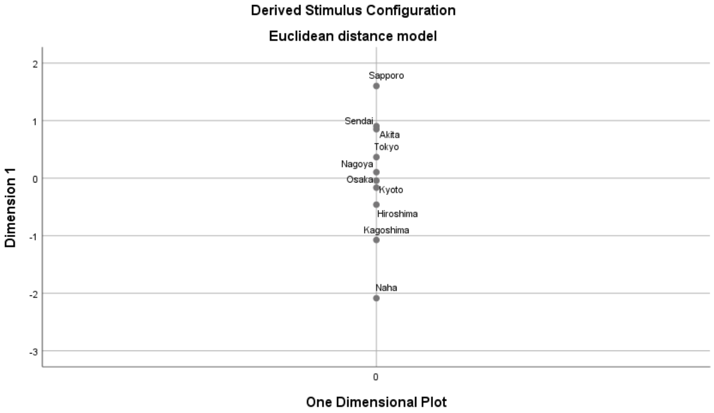
"Minimum s-stress value" → Once the model gets better than a certain amount, then it can stop. Very few reasons to alter this.

"Maximum iterations" → If there is still not a good fit after this many iterations, the model will just give up. If you try 30 and the model is still iterating, then it is perfectly sound to increase this number, but in practice, this hasn't been needed.

SPSS shows each iteration, but for L2 researchers, this data is useless. **Only the final value matters**. “Young’s S-Stress for the matrix is 0.01697, and “matrix stress” is 0.02239, with an $R^2$ value for the matrix of 0.99843. In general, **stress values of less than 0.1 or $R^2$ of higher than 0.9 are considered “good fit”** (Clopper, 2008), but these are very rarely achieved in psychology data. As you can see, even in only 1 dimension, stress is very low. Putting all the cities in Japan on a simple number line doesn’t result in huge discrepancies (assuming ordinal data).

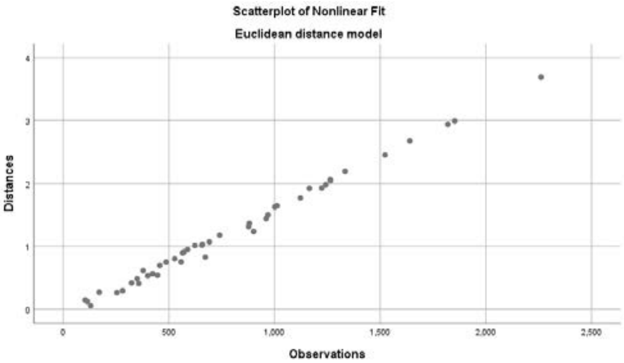


```
Iteration history for the 1 dimensional solution (in squared distances)

                  Young's S-stress formula 1 is used.

                Iteration     S-stress      Improvement

                    1           .02234
                    2           .02029         .00205
                    3           .01882         .00147
                    4           .01776         .00107
                    5           .01697         .00079

                       Iterations stopped because
                 S-stress improvement is less than   .001000


                                      Stress and squared correlation (RSQ) in distances

                              RSQ values are the proportion of variance of the scaled data (disparities)
                                         in the partition (row, matrix, or entire data) which
                                          is accounted for by their corresponding distances.
                                            Stress values are Kruskal's stress formula 1.



                For  matrix
    Stress  =   .02239      RSQ =  .99843
```



**“Scatterplot of Nonlinear Fit”** (also one for Euclidean distances, and in the case of Japan, this makes sense, but for psychological testing data, that makes very strong assumptions)

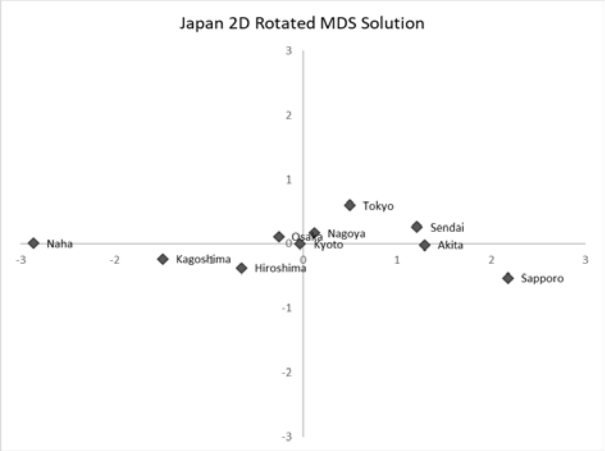
- X-axis shows the original distances between cities in the input, and Y-axis shows the distance in MDS space. The units are arbitrary, but it should be more or less a straight line. With psychological data, this would be a fantastic result, but since we’re using actual physical data, there are some discrepancies that indicate that Japanese cities are not perfectly summarized by putting them on a straight line.

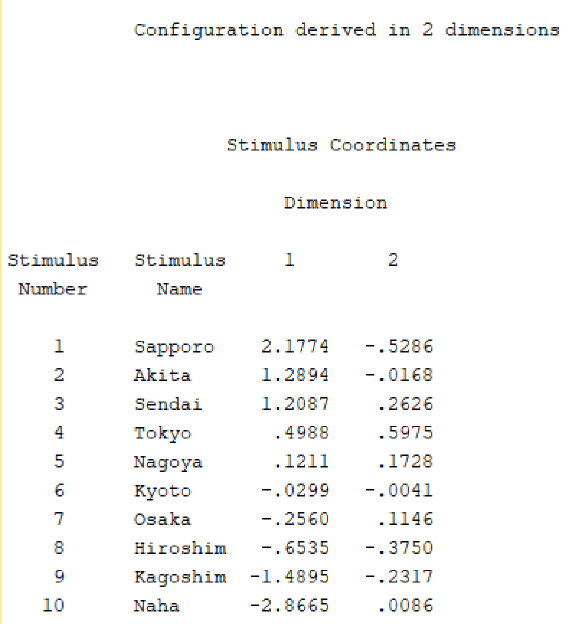


**2-dimensional output**:

The “coordinates” are the points in the combined *n*-dimensional space where the stimuli have been placed.

We can plot these points in whatever software you like. SPSS also has one, but the scales are always off and it’s hard to modify. We get this:



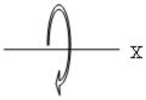


```
                Configuration derived in 2 dimensions


                      Stimulus Coordinates

                           Dimension

Stimulus   Stimulus     1        2
 Number      Name

    1      Sapporo    2.1774   -.5286
    2      Akita      1.2894   -.0168
    3      Sendai     1.2087    .2626
    4      Tokyo       .4988    .5975
    5      Nagoya      .1211    .1728
    6      Kyoto      -.0299   -.0041
    7      Osaka      -.2560    .1146
    8      Hiroshim   -.6535   -.3750
    9      Kagoshim  -1.4895   -.2317
   10      Naha      -2.8665    .0086
```

It doesn’t look much like Japan. Typically, MDS puts the dimension that explains the largest portion of the variance on the **x-axis**. In our case, Japan is long and skinny, so North-South is much more important than East-West, and it looks tilted on its side. We can take the coordinates and multiply that matrix by a rotation matrix in order to spin the points around. This **rotation** is only done to make things more easily interpretable.

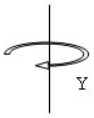
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$R_x$ rotates the points around the z-axis



$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

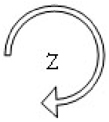
$R_y$ rotates the points around the y-axis



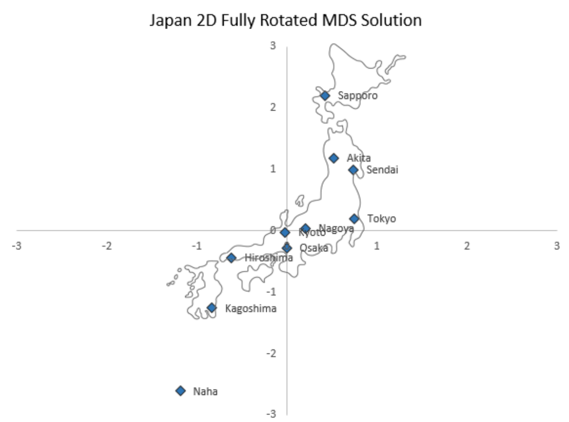
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$R_z$ rotates the points around the x-axis

**[This will be the only one used in 2d solutions]**



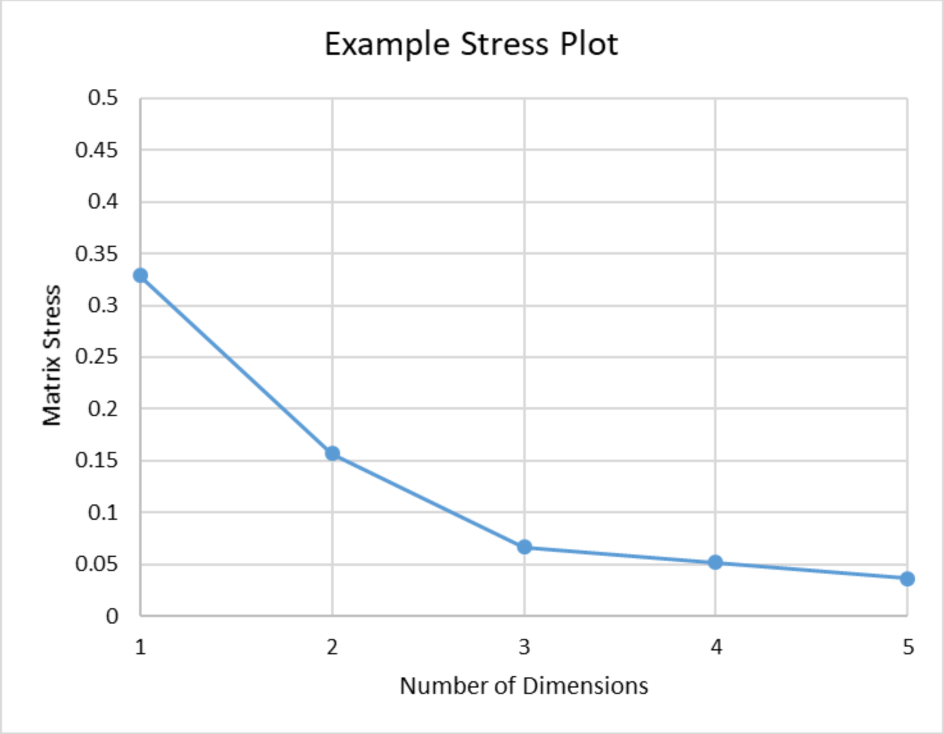
Rotating the points gives us this map:



Summary: the math behind MDS can accurately detect how many dimensions there are, and recreate a map of a space based solely on the distances between the points contained in it.

## Stress Plots

One measure of how much improvement in model fit you make by increasing dimensionality. Unlike a scree plot in Factor Analysis, you will be looking at the point at the ***bottom*** of the elbow. After that point, increasing dimensionality does not drastically increase fit. Alternatively, in SPSS, it's easy to obtain $R^2$ values, and you can choose the number of dimensions beyond which $R^2$ does not dramatically increase. Importantly, though, there is no cut-and-dry, mathematical cutoff for dimensionality, so researchers typically have to make an argument from interpretability of the data (Atagi & Bent, 2014; Clopper, 2008).



(From real data on American English listeners' grouping rates on a Free Classification task for German vowels (Daidone, Kruger, & Lidster, 2015))

This particular example is most likely best analyzed as 3-dimensional, so long as the 3d output is interpretable.

Neither SPSS nor R will make a stress plot for you automatically. You have to actually go in and run the model with $k$ = 1, $k$ = 2, $k$ = 3, and so on, and record the stress values and then plot them in order to get this.

## Summary of Assessing Dimensionality

Essentially, there are four criteria you can use, in decreasing order of importance:

1. Interpretability

2a. Bottom of the elbow on a stress plot

2b. Top of the elbow on an $R^2$ plot

4. Point at which $R^2$ is greater than 0.9 and/or stress is less than 0.1, within limits given the nature of the data

## Applied Example in R: Perception of German Vowels

Possible Data Sources:

**Perceptual assimilation**

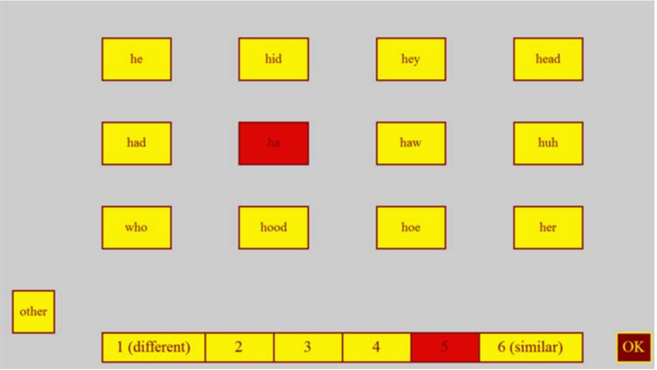
Overlap scores (Levy, 2006) yield pairwise similarities. 1 – overlap scores → dissimilarities.

**Similarity Judgment Task**

Pairs of stimuli, rated on a Likert scale from “identical” to “very different”

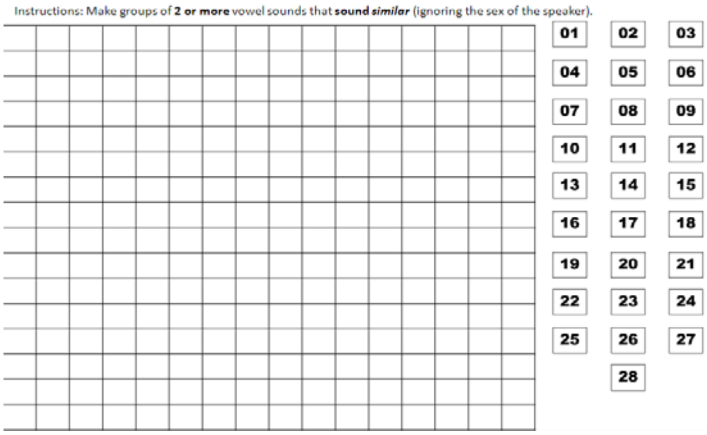
**Distances Calculated from Rater Scores**

Raters gave a “5” to person A, and a “4” to person B, meaning a distance of 1



**Free classification**

How often was stimulus A grouped with stimulus B (%)
Grouped together more often = more similar
93% of the time in a group = distance of 0.07
15% of the time in a group = distance of 0.85


Instructions: Make groups of **2 or more** vowel sounds that **sound *similar*** (ignoring the sex of the speaker).

---

Input dissimilarity matrix (28x28) into SPSS or R

**For SPSS:**

1. Analyze -> Scale -> Multidimensional Scaling

2. In Model, initially choose 1-5 dimensions for min to max in order to evaluate stress. Record “matrix stress” values and plot them against number of dimensions. See what makes sense and seems to fit well enough to examine in more detail

3. Take the “Coordinates” from that output, plot them in any program and see if there is a clear pattern

4. Run correlations with the rotated points and acoustic, phonological, or indexical features to confirm whether distances between stimuli correspond to particular features of interest. (If you test significance of the correlations, make sure to correct for multiple comparisons.)

**Example with Non-Native Language Perception Data** (from Daidone, Kruger, & Lidster, 2015) **in R:**

We'll use the **isoMDS** function in the library called "MASS." The entirety of MDS can be run using 5 lines of code.

1. Set your working directory to where the dissimilarity matrix is stored as a table
   a. setwd("C:/Users/rflidste/Desktop") ← don't forget quotation marks

2. Call the MASS library
   a. library(MASS)

3. Read in your table **while giving it a variable name**, and then convert it to a matrix
   a. [initialtable] <- read.table("[Name of file]") ← don't forget quotation marks

4. Reformat the table as a matrix
   a. [initialmatrix] <- as.matrix([initialtable])

5. Run isoMDS on your matrix and give the result a variable name so you can open it up
   a. [transparent.output.name] <- isoMDS([initialmatrix], k = [number of dimensions, default is 2])

That's it. You can then either view the points by clicking on them directly, or by using the command $points to get the coordinates (i.e. locations of each stimulus).

Note that the name for the **stress is the "final value**" that gets displayed in the console output, but keep in mind that SPSS and R give stress in different units. What SPSS will report as a "matrix stress of 0.125," R will report as a "final value of 12.5." They mean the same thing. Less than 0.1 / 10 is great, but usually unobtainable. More important is to find where in the "stress plot" the elbow occurs.

A full, annotated script is available for download on the PSLLT website

If you use that script, you should only have to replace your working directory and file name.

Highlight the entire script and click on "Run" to obtain the results.

At this workshop, however, will struggle through it together.

1. Set your working directory

2. Read in your table and call it something

```
table <- read.table("aegerman.txt")
```

3. Convert it to a matrix with a name so that the isoMDS function can operate on it

```
matrix <- as.matrix(table)
```

4. Call the MASS library so that you can use MDS functions

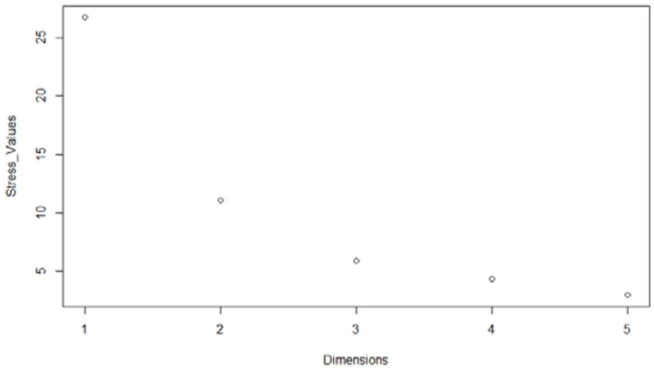
```
library(MASS)
```

5. Use the function isoMDS function, but specify dimensionality to be 1, 2, 3, and so on, and record the stress ("final") values

```
1d.output <- isoMDS(matrix, k=1)
initial  value 31.359689
iter   5 value 26.873858
iter   5 value 26.862128
final  value 26.719928
converged
2d.output <- isoMDS(matrix, k=2)
initial  value 17.698913
iter   5 value 13.675430
iter  10 value 12.074892
iter  15 value 11.173715
iter  15 value 11.163114
final  value 11.067965
converged
3d.output <- isoMDS(matrix, k=3)
initial  value 14.336974
iter   5 value 7.416254
iter  10 value 6.094048
iter  15 value 5.927000
final  value 5.857228
converged
…
```

6. You can create a stress plot by creating a column vector using the "c()" function in R with dimensions and stress values, and then using the "plot(x, y)" function to display it

```
Dimensions <- c(1,2,3,4,5)
Stress_Values < c(1d.output$stress,2d.output$stress,3d.output$stress,
                4d.output$stress,5d.output$stress)
plot(Dimensions, Stress-Values)
```



7. Decide on dimensionality, and then obtain coordinates for the stimuli in that dimensional space using $points

```
> 3d.output$points
             [,1]         [,2]        [,3]
fy   -0.379037146 -0.597544517  0.06256909
my   -0.417584600 -0.552179345  0.12950149
fY   -0.405999112 -0.165208206 -0.31670512
mY   -0.410951826 -0.217585926 -0.37365152
fi    0.801594222 -0.249652696  0.45407903
mi    0.795764086 -0.255615338  0.44386083
fI    0.786813164 -0.233236045 -0.26426827
…
…
```

8. Using your favorite plotting program, examine and rotate the points for interpretability

9. Run correlations with acoustic measurements to confirm whether distances correspond to particular features of the stimuli themselves

**Uses and Cautions**

- $R^2$ is not easy to obtain using R, but there are ways to calculate it "by hand" if reviewers need it. Alternatively, there are other functions for obtaining MDS data, including "cmdscale," which can provide $R^2$ as part of the list of objects it creates. Cmdscale, though, requires transforming your data into a "distgps" object first, which is slightly more involved for the R-uninitiated.

- Even though the actual numbers from R and SPSS may differ, it is only the *relative* positions and distances in multidimensional space that are interpretable. They can be scaled from one to the other

- Because the exact amount things are rotated is effectively arbitrary, it's important not to assume that there is one "correct" rotation amount, and be appropriately modest in interpretations

**Expansions**

- INSCAL is a way of looking at individual variation in the perceptual space. When you are able to create a full dissimilarity matrix per participant, you will get a group plot of distances, and then a set of additional values for each individual:
  - Eigenvectors for each dimension (expand or shrink dimensions in order to convert the group plot to something closer to that individual's pattern of distances)
  - Fit for that individual

**References**

Atagi, E., & Bent, T. (2013). Auditory free classification of nonnative speech. *Journal of Phonetics*, *41*, 509–519. doi: 10.1016/j.wocn.2013.09.003

Clopper, C. G. (2008). Auditory free classification: Methods and analysis. *Behavior Research Methods*, *40*, 575–581. doi: 10.3758/BRM.40.2.575

Daidone, D., Kruger, F., & Lidster, R. (2015). Perceptual assimilation and free classification of German vowels by American English listeners. In The Scottish Consortium for ICPhS 2015 (Ed.), *Proceedings of the 18th International Congress of Phonetic Sciences*. Glasgow, UK: Glasgow University.

**Perceptual Assimilation Data:**

Need a way to convert the response data into a dissimilarity matrix. There are two, somewhat competing methods for this:

Faris, M. M., Best, C. T., & Tyler, M. D. (2018). Discrimination of uncategorised non-native vowel contrasts is modulatedby perceived overlap with native phonological categories. *Journal of Phonetics*, *70*, 1-19. doi:10.1016/j.wocn.2018.05.003

Levy, E. S. (2009). On the assimilation-discrimination relationship in American English adults' French vowel learning. *Journal of the Acoustical Society of America*, *126*, 2670–2682. doi: 10.1121/1.3224715

---

For Levy's model, the overlap between L2 categories *a* and *b* is defined as:

$$Overlap_{(a,b)} = \sum_{k}^{K} \min(a_k, b_k)$$

where "min" means the minimum between two values, $a_k$ is the percent of *a* stimuli that were categorized as L1 category *k*, $b_k$ is the percent of *b* stimuli that were categorized as L1 category *k*, and this is summed across all response options *K*.

Overlap scores seek to find how much the *overall pattern of classification* "overlaps" between any two stimuli.

Example: Percent categorizations of German /i/ and /ɪ/ by American English (AE) listeners into AE categories:

| AE Category: | | i | ɪ | eɪ | ɛ | æ | ʌ | ɝ | ɑ | ɔ | oʊ | ʊ | u |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| German Vowel | i | **92.3** | 5.3 | 1.1 | 0.6 | | | | | | | 0.2 | |
| | ɪ | 5.0 | **77.8** | 8.0 | 8.7 | | 0.2 | | | | | 0.3 | |

Overlap = 5.0 + 5.3 + 1.1 + 0.6 + 0 + 0.2 = **12.2**

In total across all response categories, German /i/ and /ɪ/ were categorized the same way 12.2 percent of the time.

You'd then repeat this calculation for every pair of vowels. Note, though, that these are **similarities**, not "dissimilarities," so you would need to take 1 minus the overlap scores in order to obtain the dissimilarity matrix. The result would be a square symmetric table with German vowels **against German vowels**.