Back in the “olden days” of the 1970s it was apparently not unknown for statisticians to create 3D visualizations using *tinkertoys.*For some inexplicable reason, the advent of the PC led to a decline in this practice, with the result that 3D visualizations are now much less common. However, it is possible to create 3D-like visualizations digitally. Although they only show two dimensions at any one time, the user can understand the third dimension by interacting with the visualization. In this post I use Plotly’s excellent plotting package to create an interactive, 3D visualization of a correspondence analysis.

**The data**

The data table that I use in this example shows perceptions of different cola brands. It is a good example for correspondence analysis as the table is relatively large, and correspondence analysis is thus useful at providing a summary.

**The traditional 2D correspondence analysis map**

The standard 2D “map” from correspondence analysis is shown below. I’ve created this using the Displayr/flipDimensionReduction package , which creates maps that automatically arrange the labels so that they do not overlap. I created it using the following code

library(flipDimensionReduction)

my.ca <- CorrespondenceAnalysis(my.data,

normalization = "Column principal (scaled)")

This visualization explains 86% of the variance from the correspondence analysis. This leads to the question: is the 14% that is not explained interesting?

﻿

**Creating the 3D interactive visualization in Plotly**

The standard visualization plots of correspondence analysis plots the first two dimensions. The code below uses plotly to create a 3D plot of the first three dimensions. In theory you can encode further dimensions (e.g., using color, font size, markers and the like), but I’ve never been smart enough to interpret them myself! You can readily repurpose this code to your own correspondence analysis by replacing my.ca with the name or the object that contains your correspondence analysis. If you use a package other than flipDimensionReduction to create the correspondence analysis you will also need to work out how to extract the coordinates.

rc = my.ca$row.coordinates

cc = my.ca$column.coordinates

library(plotly)

p = plot\_ly()

p = add\_trace(p, x = rc[,1], y = rc[,2], z = rc[,3],

mode = 'text', text = rownames(rc),

textfont = list(color = "red"), showlegend = FALSE)

p = add\_trace(p, x = cc[,1], y = cc[,2], z = cc[,3],

mode = "text", text = rownames(cc),

textfont = list(color = "blue"), showlegend = FALSE)

p <- config(p, displayModeBar = FALSE)

p <- layout(p, scene = list(xaxis = list(title = colnames(rc)[1]),

yaxis = list(title = colnames(rc)[2]),

zaxis = list(title = colnames(rc)[3]),

aspectmode = "data"),

margin = list(l = 0, r = 0, b = 0, t = 0))

p$sizingPolicy$browser$padding <- 0

my.3d.plot = p

If you are reading this on a desktop, you will be able to interact with the visualization below.

In addition to creating it in Plotly, Displayr , It’s free to do this; you just click **Insert > R Output**, paste in the R code, press **CALCULATE**, and then either choose **Export > Public Web Page,** or **Export > Embed**to embed it in another document or page. If you [click here](https://app.displayr.com/Try/3D%20Correspondence%20Analysis) you will go into a Displayr document containing the code used to create the analyses and visualizations in this chart. You can modify it however you like to re-use for your own analyses.