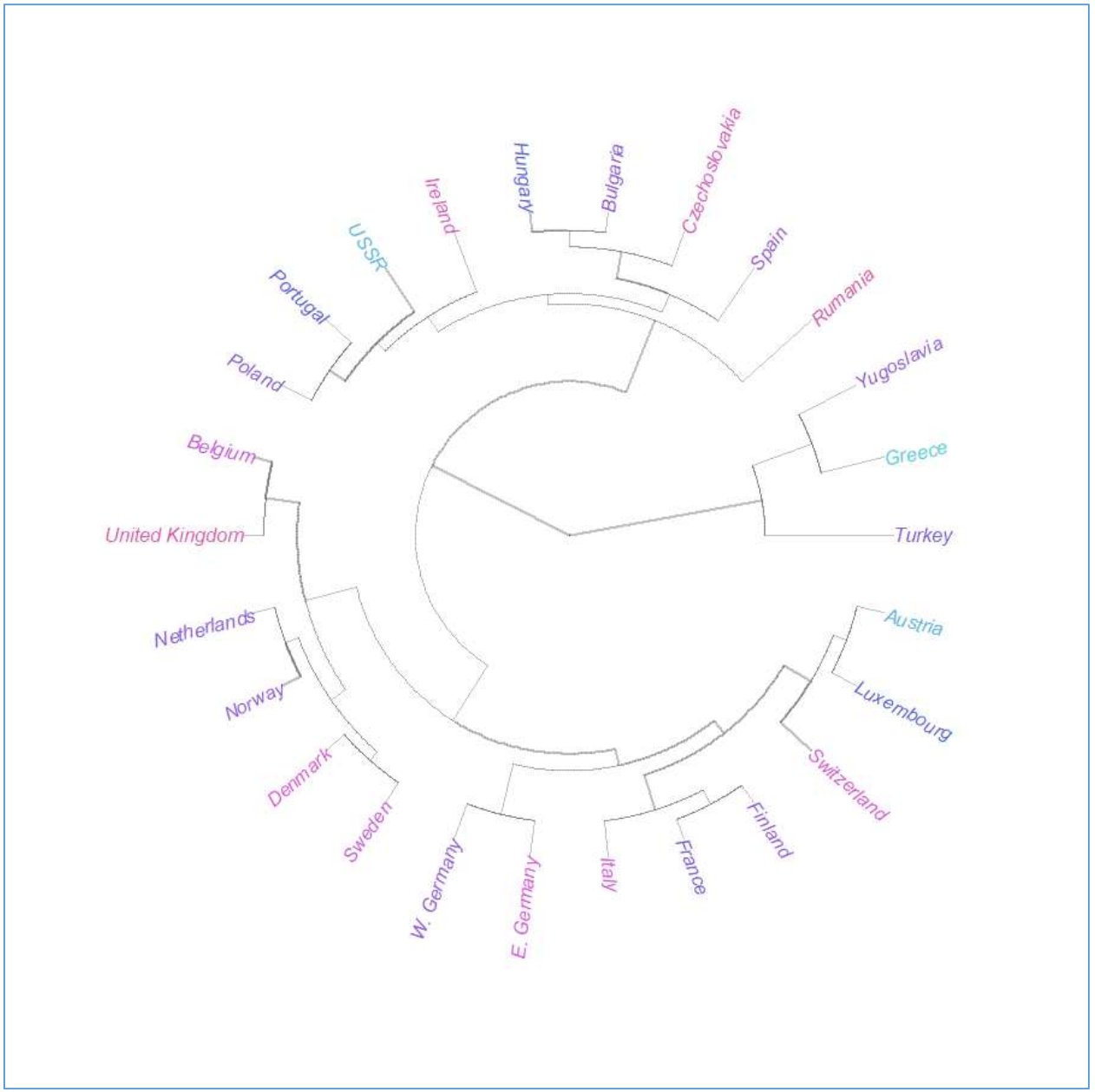
Problem 1

You can find a dataset dealing with European employment in 1979 at http://lib.stat.cmu.edu/DASL/Stories/EuropeanJobs.html. This dataset gives the percentage of people employed in each of a set of areas in 1979 for each of a set of European countries. Notice this dataset contains only 26 data points. That's fine; it's intended to give you some practice in visualization of clustering.

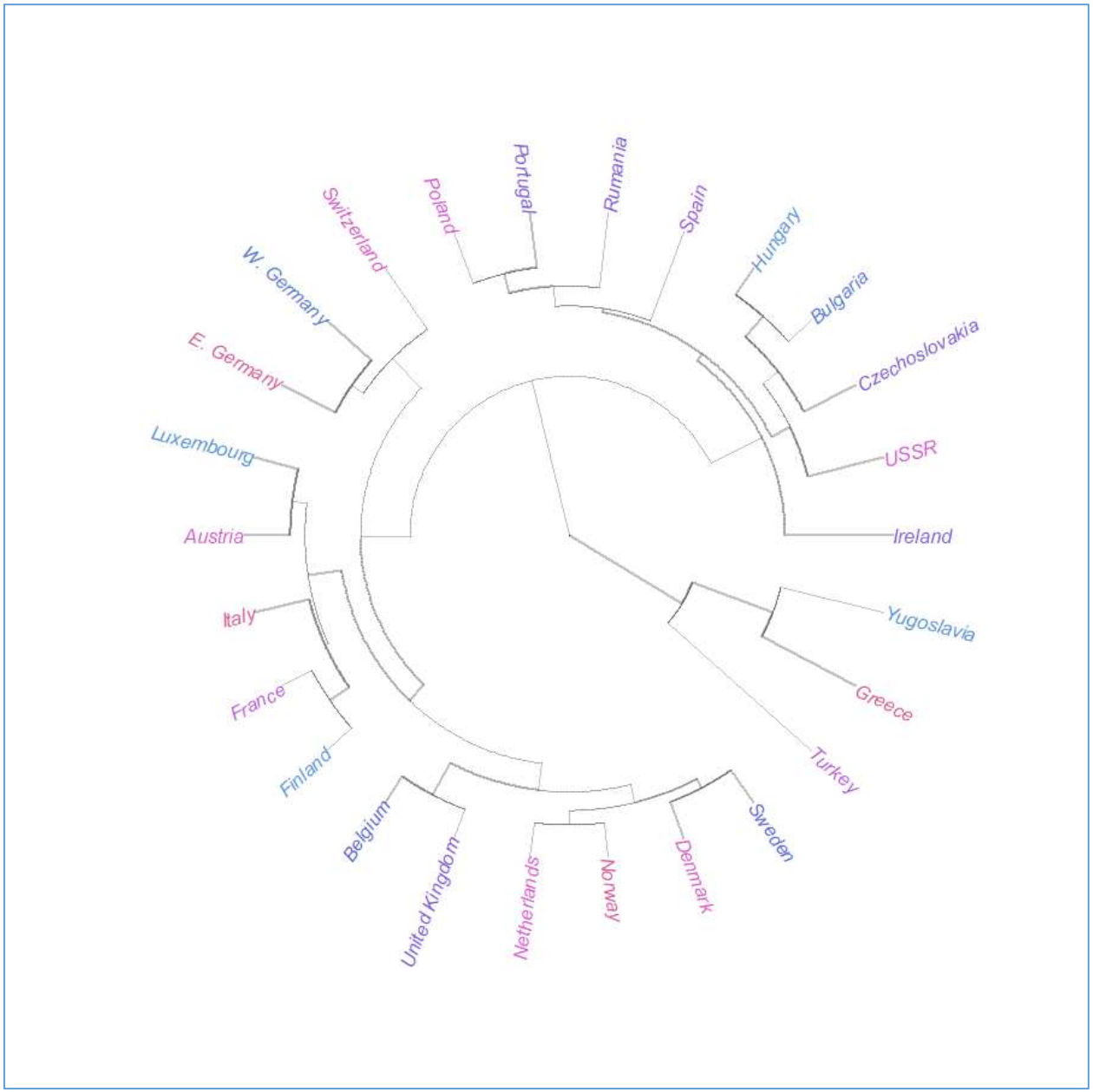
1. Use an agglomerative clusterer to cluster this data. Produce a dendrogram of this data for each of single link, complete link, and group average clustering. You should label the countries on the axis. What structure in the data does each method expose? it's fine to look for code, rather than writing your own. Hint: I made plots I liked a lot using R's hclust clustering function, and then turning the result into a phylogenetic tree and using a fan plot, a trick I found on the web; try plot(as.phylo(hclustresult), type='fan'). You should see dendrograms that "make sense" (at least if you remember some European history), and have interesting differences.

My code is found in hw4_prob1.R. The following three pages depict the dendrograms for single link, complete link and group average clustering respectively. I obtained guidance from https://rpubs.com/gaston/dendrograms on plotting these dendrograms and using the ape package. I believe the complete link clustering make the most sense as the clusters correspond well to the divisions along an eastern and western bloc of nations post-World War II. Interestingly the Nordic nations (Denmark, Norway and Sweden) are group together with the addition of the Netherlands. Also Austria, Luxemborg and Switzerland are group together as the high-income land-locked Alpine nations. Whereas in the Single Link Clustering dendrogram, this grouping does not occur.

Single Link Clustering:



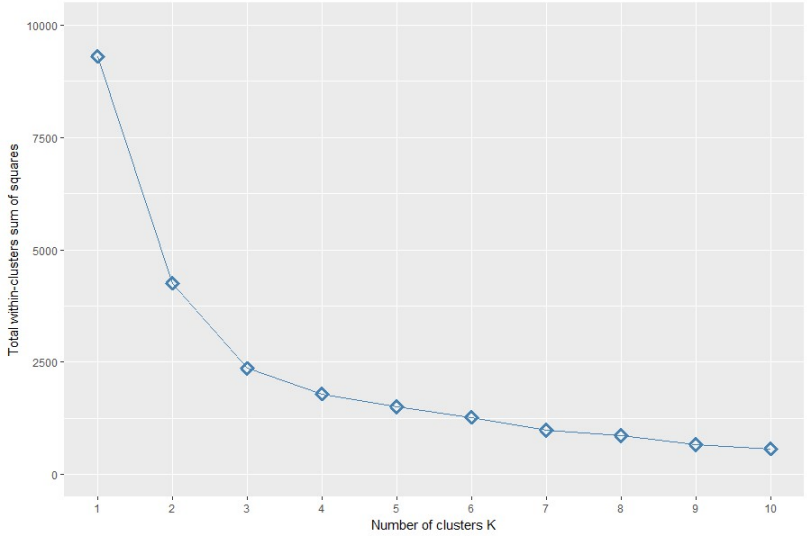
Complete Link Clustering:

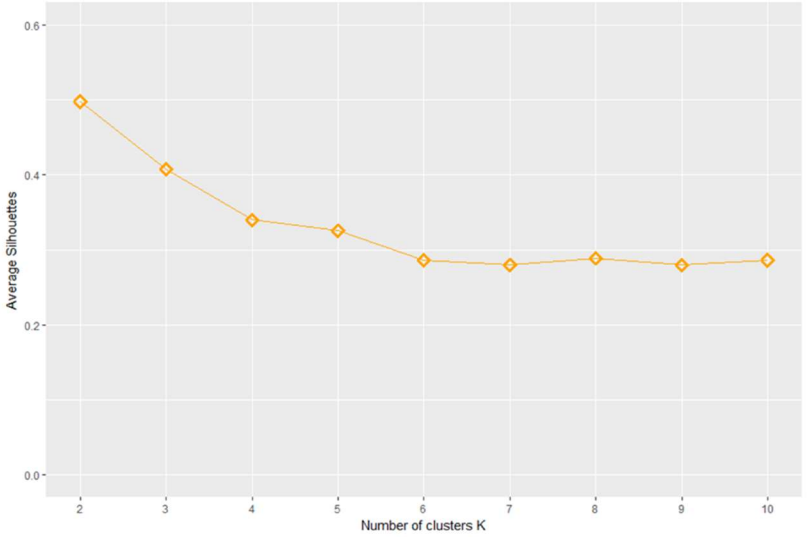


Group Average Clustering:

2. Using k-means, cluster this dataset. What is a good choice of k for this data and why?

My code is found in hw4_prob1.R. A good number for k would be either three or four according to looking at the knee in the total within sum of squares plot or the average silhouettes plot (see below). I used guidance from https://uc-r.github.io/kmeans_clustering.





Problem 2

Do exercise 6.2 in the Jan 15 version of the course text

6.2. Obtain the activities of daily life dataset from the UC Irvine machine learning website: (https://archive.ics.uci.edu/ml/datasets/Dataset+for+ADL+Recognition+with+Wrist-worn+Accelerometer data provided by Barbara Bruno, Fulvio Mastrogiovanni and Antonio Sgorbissa).

(a) Build a classiffier that classifies sequences into one of the 14 activities provided. To make features, you should vector quantize, then use a histogram of cluster centers (as described in the subsection; this gives a pretty explicit set of steps to follow). You will find it helpful to use hierarchical k-means to vector quantize. You may use whatever multi-class classifier you wish, though I'd start with R's decision forest, because it's easy to use and effective. You should report (a) the total error rate and (b) the class confusion matrix of your classifier.

My code is found in hw4_prob2_AB_v2.R. I used a k of 480. I used a vector length (chunk size) of 32 to perform the vector quantization. I used R's randomForest function from the randomForest package. I obtained an accuracy of 73.7% and my confusion matrix is below with the diagonals highlighted in blue.

	Brush_teeth	Climb_stairs	Comb_hair	Descend_stairs	Drink_glass	Eat_meat	Eat_soup	Getup_bed	Liedown_bed	Pour_water	Sitdown_chair	Standup_chair	Use_telephone	Walk	class.error
Brush_teeth	6	0	0	0	0	0	0	0	0	2	1	0	0	0	33.3%
Climb_stairs	0	72	0	1	0	0	0	1	0	1	1	3	0	2	11.1%
Comb_hair	0	0	16	0	4	0	0	1	0	1	1	1	0	0	33.3%
Descend_stairs	0	5	0	22	1	0	0	0	0	0	2	2	0	1	33.3%
Drink_glass	0	0	0	0	76	0	0	0	0	2	1	0	1	0	5.0%
Eat_meat	0	0	0	0	1	0	0	0	0	3	0	0	0	0	100.0%
Eat_soup	0	0	0	0	0	0	0	0	0	2	0	0	0	0	100.0%
Getup_bed	0	0	0	0	3	0	0	44	0	7	6	20	0	0	45.0%
Liedown_bed	0	0	0	0	0	0	0	6	1	2	11	2	0	0	95.5%
Pour_water	0	0	0	0	0	0	0	1	0	75	3	1	0	0	6.3%
Sitdown_chair	0	0	0	0	0	0	0	1	1	2	63	13	0	0	21.3%
Standup_chair	0	0	0	0	0	0	0	5	0	3	11	62	0	0	23.5%
Use_telephone	0	0	0	0	4	0	0	0	0	1	2	0	3	0	70.0%
Walk	0	16	0	0	0	0	0	1	0	1	2	9	0	51	36.3%

(b) Now see if you can improve your classifier by (a) modifying the number of cluster centers in your hierarchical k-means and (b) modifying the size of the fixed length samples that you use.

My code is found in hw4_prob2_AB_v2.R. I tried chunk sizes of 12, 24, 36, 48, 60 and 72. I used k values of 120, 240, 360, 480, 600 and 720. As the chunk size increased, in general, the accuracy decreased. As the number of clusters increased, in general, the accuracy decreased. The best accuracy I obtained in my study depicted below is 78.23% accuracy with a chunk size of 12 and a k value of 120.

		Number of Clusters, k					
		120	240	360	480	600	720
Chunk Size	12	**78.228230%**	78.078080%	76.876880%	77.477480%	75.675680%	76.576580%
	24	76.126130%	78.228228%	76.126126%	75.825826%	75.825826%	76.726727%
	36	75.225230%	70.120120%	71.021021%	70.720721%	70.720721%	70.720721%
	48	75.375380%	71.921922%	73.423423%	71.321321%	68.618619%	67.117117%
	60	70.870870%	71.621622%	69.069069%	68.918919%	66.816817%	66.516517%
	72	71.621620%	69.969970%	69.219219%	68.168168%	64.414414%	64.864865%