Homework 4: Unsupervised Learning

MACS 30100: Perspectives on Computational Modeling
University of Chicago

Overview

For each of the following prompts, produce responses *with* code in-line. While you are encouraged to stage and draft your problem set solutions using any files, code, and data you'd like within the private repo for the assignment, *only the final, rendered PDF with responses and code in-line will be graded.*

Note: take a look at the `hw04.pdf` file to see a better rendering of this problem set (e.g., cleaner looking table, etc.).

Dimension Reduction

Conceptual Problems

1. (5 points) Compute the total variance from the following PCA output.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Standard deviation	3.55	2.41	1.82	1.31	1.05	0.86	0.81	0.79	0.72	0.70
Variance	3.45	3.10	1.75	0.98	0.64	0.33	0.31	0.30	0.09	0.05

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(corrr)
library(amerika)
library(factoextra)
```

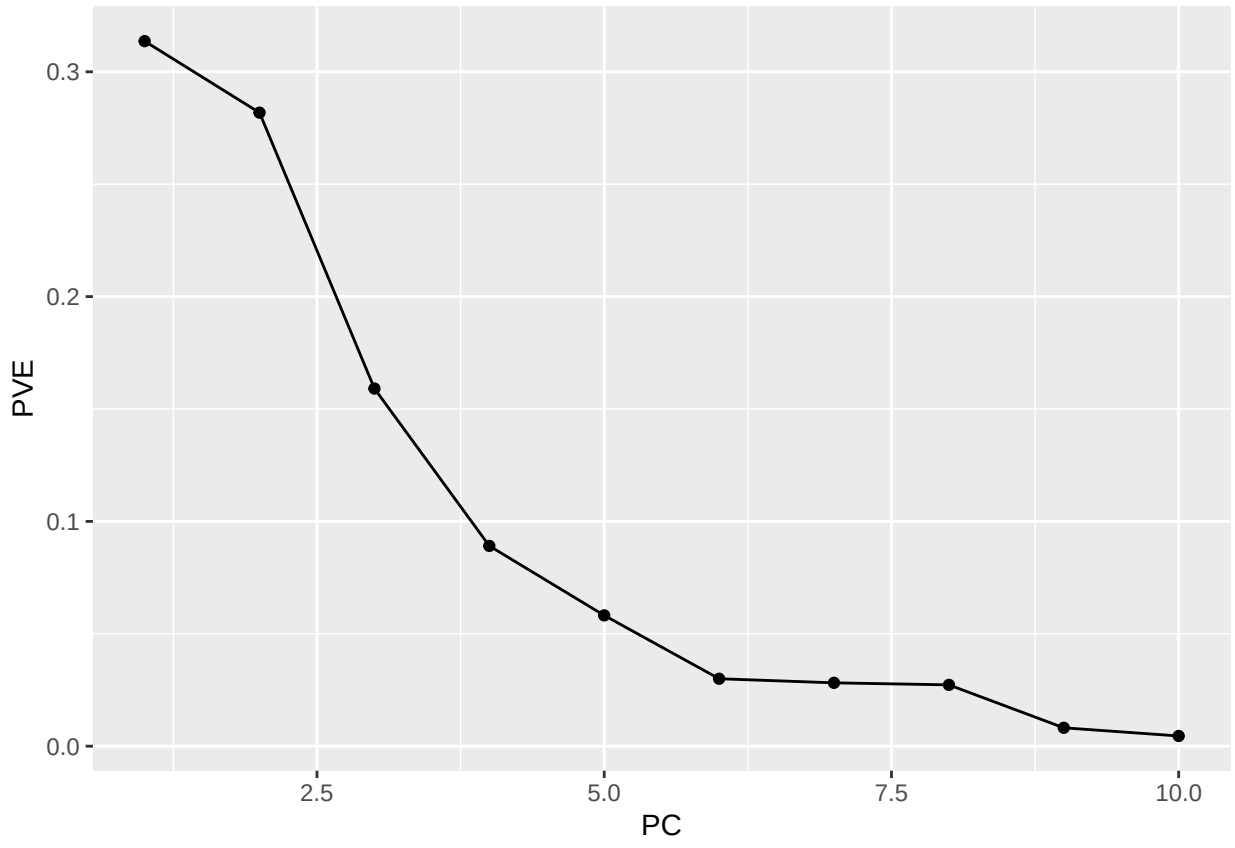
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(patchwork)
library(ggrepel)

var <- c(3.45, 3.10, 1.75, 0.98, 0.64, 0.33, 0.31, 0.3, 0.09, 0.05)
tot_var <- sum(var)
tot_var
```

```
## [1] 11
```

2. (10 points) Make a *manual* scree plot based on these results. That is, *no* canned functions or packages (e.g., `factoextra`).

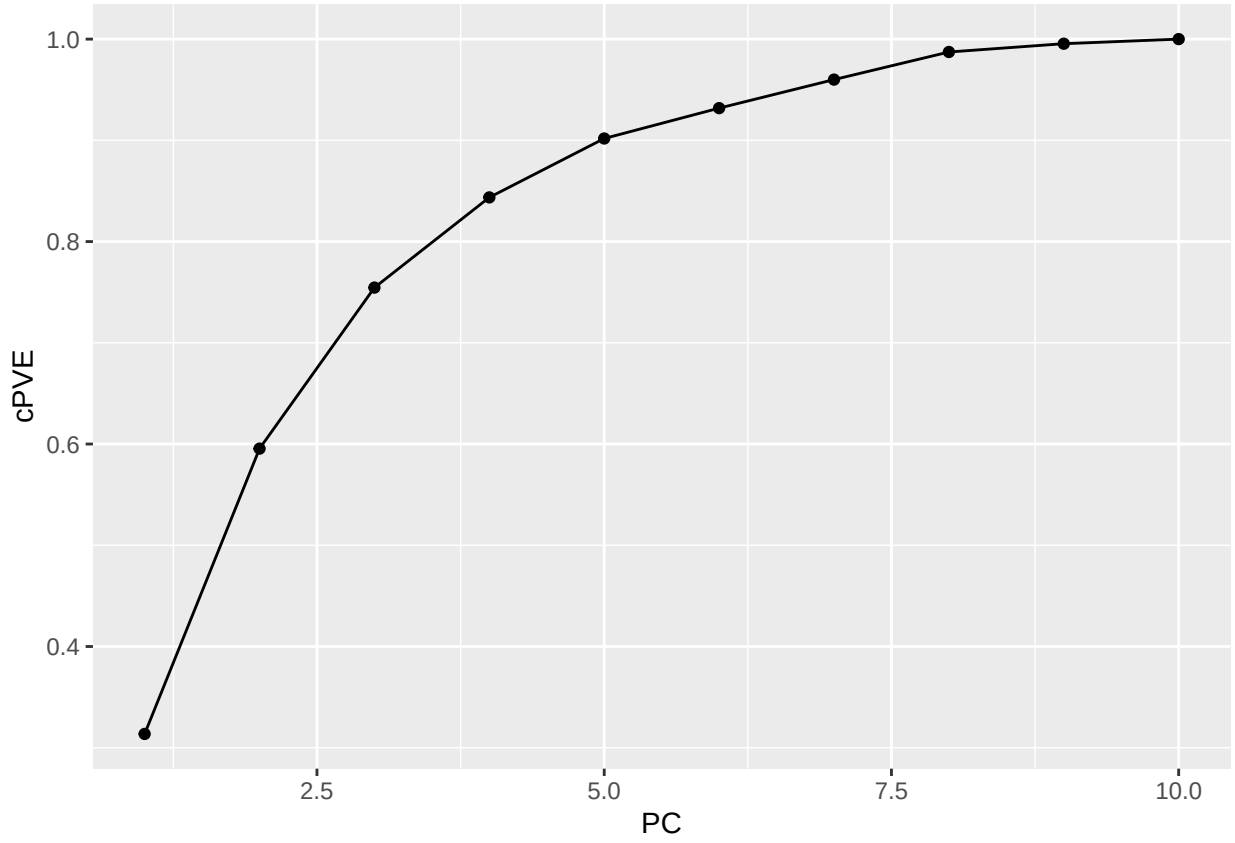
```
scree <- tibble(PC = 1:10, PVE = var / tot_var)
ggplot(data = scree) +
  geom_line(aes(x = PC, y = PVE)) +
  geom_point(aes(x = PC, y = PVE))
```



3. (10 points) Based on your results in the previous question, how many PCs would you suggest characterize these data well? That is, what would the dimensionality of your new reduced data space be?

```
add <- 0
cumPVE <- rep(0, length(scree$PVE))
```

```r
for (i in 1:length(scree$PVE)) {
  cumPVE[i] <- add + scree$PVE[i]
  add = cumPVE[i]
}
scree$cPVE <- cumPVE

ggplot(data = scree) +
  geom_line(aes(x = PC, y = cPVE)) +
  geom_point(aes(x = PC, y = cPVE))
```




```r
scree
```

```
## # A tibble: 10 x 3
##       PC     PVE  cPVE
##    <int>   <dbl> <dbl>
##  1     1 0.314   0.314
##  2     2 0.282   0.595
##  3     3 0.159   0.755
##  4     4 0.0891  0.844
##  5     5 0.0582  0.902
##  6     6 0.03    0.932
##  7     7 0.0282  0.96
##  8     8 0.0273  0.987
##  9     9 0.00818 0.995
## 10    10 0.00455 1
```

Three principal components seems to explain most of the variation in the data set with 75 while also allowing visualization of the data in three dimensional space. Looking at the scree plot,

4. (10 points) Calculate the Euclidean distance between each of the following observations, i, and some observation at 0 (i.e., x_0) in 4-dimensional space $\forall X \in \{1, 2, 3, 4\}$.

i	X_1	X_2	X_3	X_4	Euclidean Distance
1	2	2	3	1	...
2	1	1	-2	2	...
3	1	-2	-2	-1	...
4	3	3	2	2	...
5	-3	2	-1	1	...

```
x1 <- c(2, 2, 3, 1)
x2 <- c(1, 1, -2, 2)
x3 <- c(1, -2, -2, -1)
x4 <- c(3, 3, 2, 2)
x5 <- c(-3, 2, -1, 1)
vectors <- list(x1, x2, x3, x4, x5)

distances <- rep(0, 5)
for (i in 1:length(vectors)) {
   distances[i] <- sqrt(sum(vectors[[i]]^2))
}

data.frame(i = seq(1, 5), distance = distances)
```

```
##   i distance
## 1 1 4.242641
## 2 2 3.162278
## 3 3 3.162278
## 4 4 5.099020
## 5 5 3.872983
```

An Applied Problem

For the following applied problem, use the 2019 American National Election Study (ANES) Pilot survey data. These data include, among many other features, a battery of 35 feeling thermometers, which are questions with answers ranging from 1 to 100 for how respondents "rate" some topic (e.g., *How would you rate Obama?* or *How would you rate Japan?*). See the documentation and more detail here.

To make your lives a bit easier, I have preprocessed the data for you, including: 1) feature engineering (via kNN) for missing data, and 2) reduction of the feature space to include only the 35 feeling thermometers and a feature for the respondent's party affiliation (`democrat`), where 1 = Democrat and 0 = non-Democrat (which could be Republican, Independent, or decline to say).

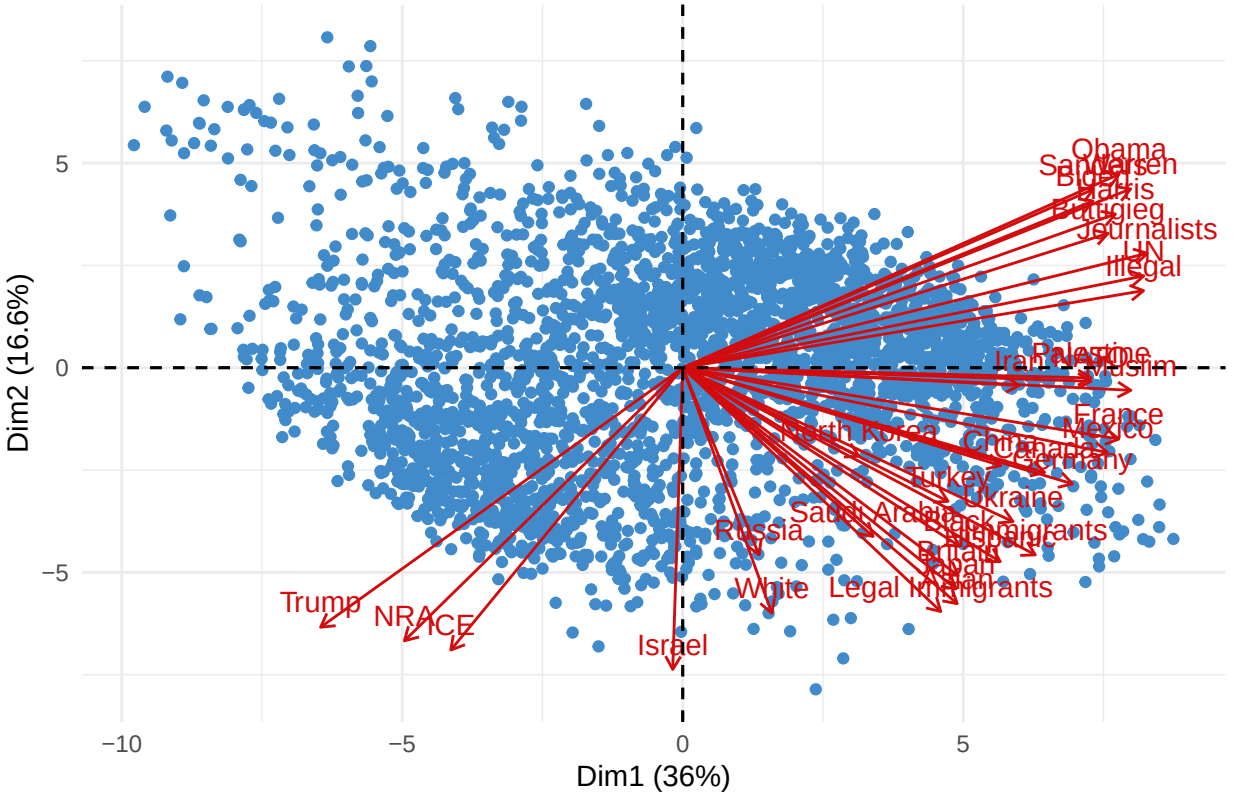
5. (10 points) Fit a PCA model on all 35 feeling thermometers from the 2019 ANES, but be careful to *not* include the party affiliation feature.

```
anes <- read_rds('/Users/egemenpamukcu/Downloads/ps4-egemenpamukcu-main/data/anes.rds')

pca_fit <- anes[,-36] %>%
  scale() %>%
  prcomp()

summary(pca_fit)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.5491 2.4082 1.82408 1.30799 1.04776 0.86327 0.81279
## Proportion of Variance 0.3599 0.1657 0.09506 0.04888 0.03137 0.02129 0.01888
## Cumulative Proportion  0.3599 0.5256 0.62065 0.66953 0.70090 0.72219 0.74107
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.78918 0.72418 0.69914 0.68242 0.66830 0.65582 0.63517
## Proportion of Variance 0.01779 0.01498 0.01397 0.01331 0.01276 0.01229 0.01153
## Cumulative Proportion  0.75886 0.77384 0.78781 0.80112 0.81388 0.82616 0.83769
##                           PC15   PC16    PC17   PC18    PC19    PC20    PC21
## Standard deviation     0.62471 0.6176 0.60486 0.5826 0.57746 0.56879 0.55645
## Proportion of Variance 0.01115 0.0109 0.01045 0.0097 0.00953 0.00924 0.00885
## Cumulative Proportion  0.84884 0.8597 0.87019 0.8799 0.88942 0.89866 0.90751
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.55282 0.54164 0.53058 0.52288 0.51509 0.50415 0.48309
## Proportion of Variance 0.00873 0.00838 0.00804 0.00781 0.00758 0.00726 0.00667
## Cumulative Proportion  0.91624 0.92462 0.93267 0.94048 0.94806 0.95532 0.96199
##                           PC29    PC30    PC31    PC32    PC33    PC34    PC35
## Standard deviation     0.47198 0.45893 0.45325 0.43761 0.42993 0.40210 0.39188
## Proportion of Variance 0.00636 0.00602 0.00587 0.00547 0.00528 0.00462 0.00439
## Cumulative Proportion  0.96835 0.97437 0.98024 0.98571 0.99099 0.99561 1.00000
```

6. (20 points) Plot the feature contributions from each of the feeling thermometers in the first two dimensions (i.e., PC1 and PC2). Describe the patterns, groupings, and structure of the lower-dimensional projections in *substantive* terms.

```
pca_fit %>%
  fviz_pca_biplot(label = "var",
                  col.var = amerika_palettes$Republican[2],
                  col.ind = amerika_palettes$Democrat[3]) +
  labs(title = "") +
  theme_minimal()
```



It seems like the (upper) right and (lower) left corners of the two dimensional representation of our data corresponds to Democrats and Republicans respectively. The observations accumulated near the upper right corner of the biplot have warmer feelings towards politicians affiliated with the Democratic Party, such as Obama, Biden, and Harris. They also happen to be more 'globalist' as they have warmer feelings towards the United Nations and illegal immigrants. The lower left corner corresponds to three features that is positively correlated with Republican party identity, namely Trump, ICE and NSA. The loading vectors for ICE and Illegal Immigrant features are nearly 180 degrees apart, which indicates that, expectedly, these two features are negatively correlated, same logic applies to the Trump and Obama feature pair as well. The lower right corner of the visualization houses relatively more bipartisan features, as most of them are almost orthogonal to the directly partisan features. Nevertheless, some party identification can still be inferred as the features "White", "Israel" and "Russia" are closer to Republican Party identity, whereas "Palestine" and "Muslim" are closer to features affiliated with Democratic Party.

Clustering

A Conceptual Problem

7. (10 points) What are the two properties required for a *hard* partitional solution, and when thus relaxed, give a *soft* partitional clustering solution? Be sure to answer this both formally (with mathematical notation) and substantively (with words). Then, give an example or two of each and how they relate to these two central properties of clustering.

Hard partitioning methods such as k-means assign each observation to the cluster it 'fits' the best. At the end of the clustering process, each observation belongs to one of the kth clusters so that $c_1 \cup c_2 \cup ...c_k$ covers the all the observations in the data set. Hard clustering also does not allow overlapping between clusters which can be shown as $C_k \cap c'_k = \varnothing$. Soft partitional clustering instead allows overlapping clusters

and, instead of assigning each observation to the most likely cluster, it returns a probability distribution indicating the likelihood of an observation belonging to a specific clusters. An example of hard partitioning is k-means where you decide beforehand how many clusters you want (k) and the alogrithm will classify every observation as belonging to one of those groups. An example of soft partitioning is GMM which uses distribution assumptions to weight the probabilities of certain observations belonging to clusters.

An Applied Problem

In this applied problem, you will again use the 2019 ANES data, but this time to explore the clustering solution from fitting a fuzzy c-means (FCM) algorithm to all feeling thermometers. As with the dimension reduction exercise, derive a clustering solution using *only* the feeling thermometers. The idea here is to explore whether attitudes on these issues, countries, and people map onto natural groupings between major American political parties.

8. (5 points) Load and scale the ANES *feeling thermometer* data.

```
anes_scaled <- anes %>%
  scale()
anes_scaled <- as.tibble(anes_scaled)
```

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
anes_scaled$democrat <- anes$democrat
anes_scaled
```

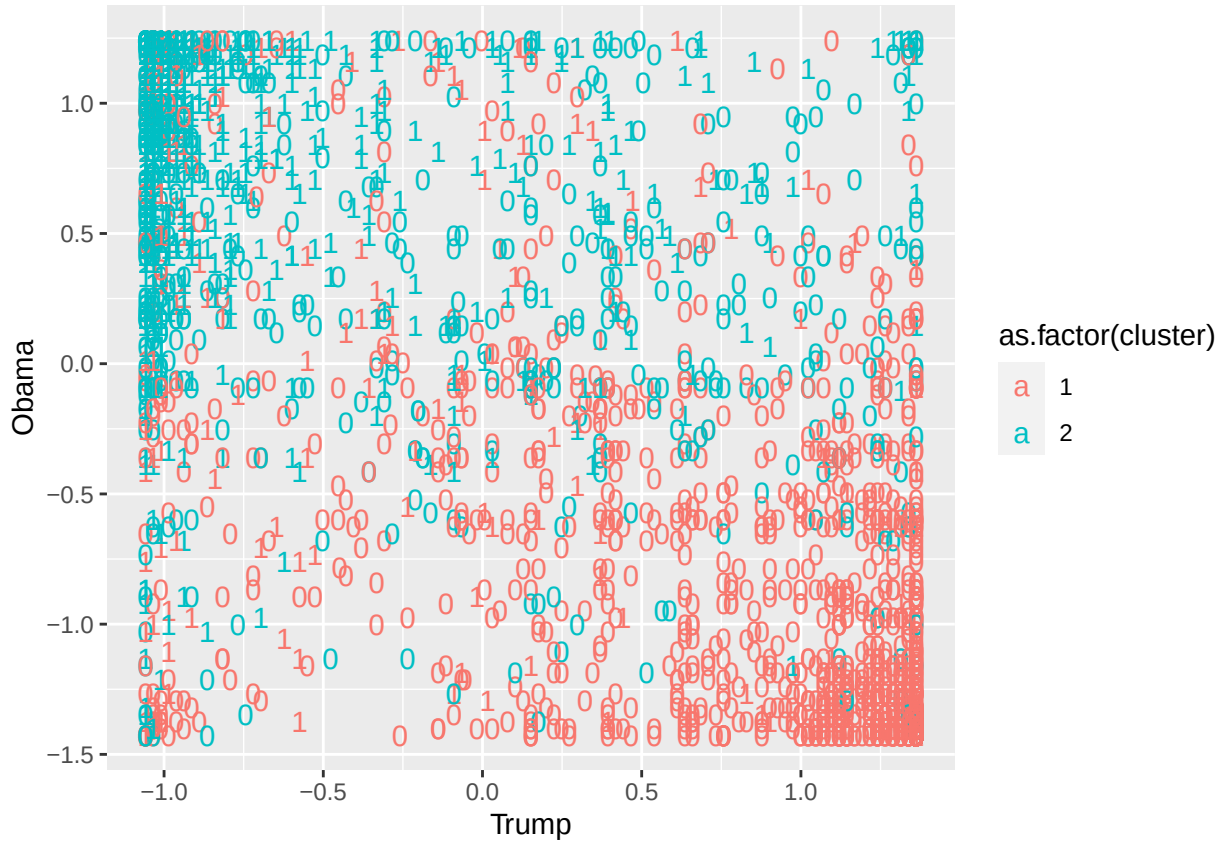
```
## # A tibble: 3,165 x 36
##      Trump   Obama   Biden Warren Sanders Buttigieg Harris   Black   White
##      <dbl>   <dbl>   <dbl>  <dbl>   <dbl>     <dbl>  <dbl>   <dbl>   <dbl>
##  1  0.0794  0.974   0.293   0.332   0.194     0.837  0.486  1.18    1.23
##  2 -0.0658 -0.626  -0.0379 -0.699  -0.324    -0.302 -0.221  0.848   1.10
##  3 -1.06    1.00    1.38   -0.758   0.511     1.05   1.09  -0.962  -0.961
##  4  1.36   -0.0929 -1.27   -1.20   -1.22     -1.31  -1.20  -2.98   -3.11
##  5  1.22   -0.947  -0.519  -1.17   -0.929    -0.774 -0.963  0.932   1.01
##  6  0.0552  0.787   1.13    0.832   1.09      0.675  0.958  0.679  -0.0837
##  7 -1.06    1.24    0.984   1.48    0.280     1.08   1.53   1.14    0.618
##  8 -1.03    0.787   0.0824  1.33    0.827     1.08   1.46   0.0901  0.180
##  9  0.418  -1.40   -0.429  -1.14   -1.22     -1.28  -1.16  -2.98    1.28
## 10 -1.06    1.24    1.44    0.862   1.52      1.38   0.149  0.385   0.443
## # ... with 3,155 more rows, and 27 more variables: Hispanic <dbl>, Asian <dbl>,
## #   Muslim <dbl>, Illegal <dbl>, Immigrants <dbl>, `Legal Immigrants` <dbl>,
## #   Journalists <dbl>, NATO <dbl>, UN <dbl>, ICE <dbl>, NRA <dbl>, China <dbl>,
## #   `North Korea` <dbl>, Mexico <dbl>, `Saudi Arabia` <dbl>, Ukraine <dbl>,
## #   Iran <dbl>, Britain <dbl>, Germany <dbl>, Japan <dbl>, Israel <dbl>,
## #   France <dbl>, Canada <dbl>, Turkey <dbl>, Russia <dbl>, Palestine <dbl>,
## #   democrat <dbl>
```

9. (5 points) Fit an FCM algorithm to the scaled data initialized at $k = 2$, driven by the assumption that party affiliation (Democrat or non-Democrat) underlies these data.

```
library(e1071)
anes_scaled <- anes_scaled[, -36]

fuzzy <- cmeans(x = anes_scaled, centers = 2)
anes_scaled$cluster <- fuzzy$cluster
```

10. (15 points) Visualize the cluster scores from your FCM solution plotted over the range of feelings toward `Trump` and `Obama`, with data points colored by cluster assignment and also labeled by the respondent's true party affiliation (the `democrat` feature). As party wasn't included in your clustering solution, what can you conclude based on these patterns? Is there a grouping pattern among observations along a partisan dimension, or isn't there? Do respondents group in expected ways (e.g., Trump supporters to the right and Obama supporters to the left)? Do cluster assignments align with the true party affiliation or not? How would you evaluate the effectiveness of FCM for this type of task?

```
anes_scaled$democrat = anes$democrat

ggplot(data = anes_scaled) +
  geom_text(aes(x = Trump, y = Obama, label=democrat, col = as.factor(cluster)))
```



Looking at the visualization, the fuzzy c-means clustering method seems to capture most of the party affiliation in the dataset. Roughly the party identification in this two dimensional representation corresponds to the lower right and upper left corners (people who hate and love either Trump or Obama) and the coloring in the graph seems to match that pattern. Most of the 0s (non-Democrats) are colored in blue and most of the democrats (1s) are colored in red. As expected, for the observations laying close to the lower left - upper right diagonal, the clustering results align less with the true party identification which indicates

the effect of the feeling thermometers for Trump and Obama on the clustering predictions. In general, FCM does a good job in understanding party identity in an unsupervised manner.