Homework 4: Unsupervised Learning

MACS 30100: Perspectives on Computational Modeling
University of Chicago

Boseong Yun

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

- Answer: The total variance of the following PCA output is 11!

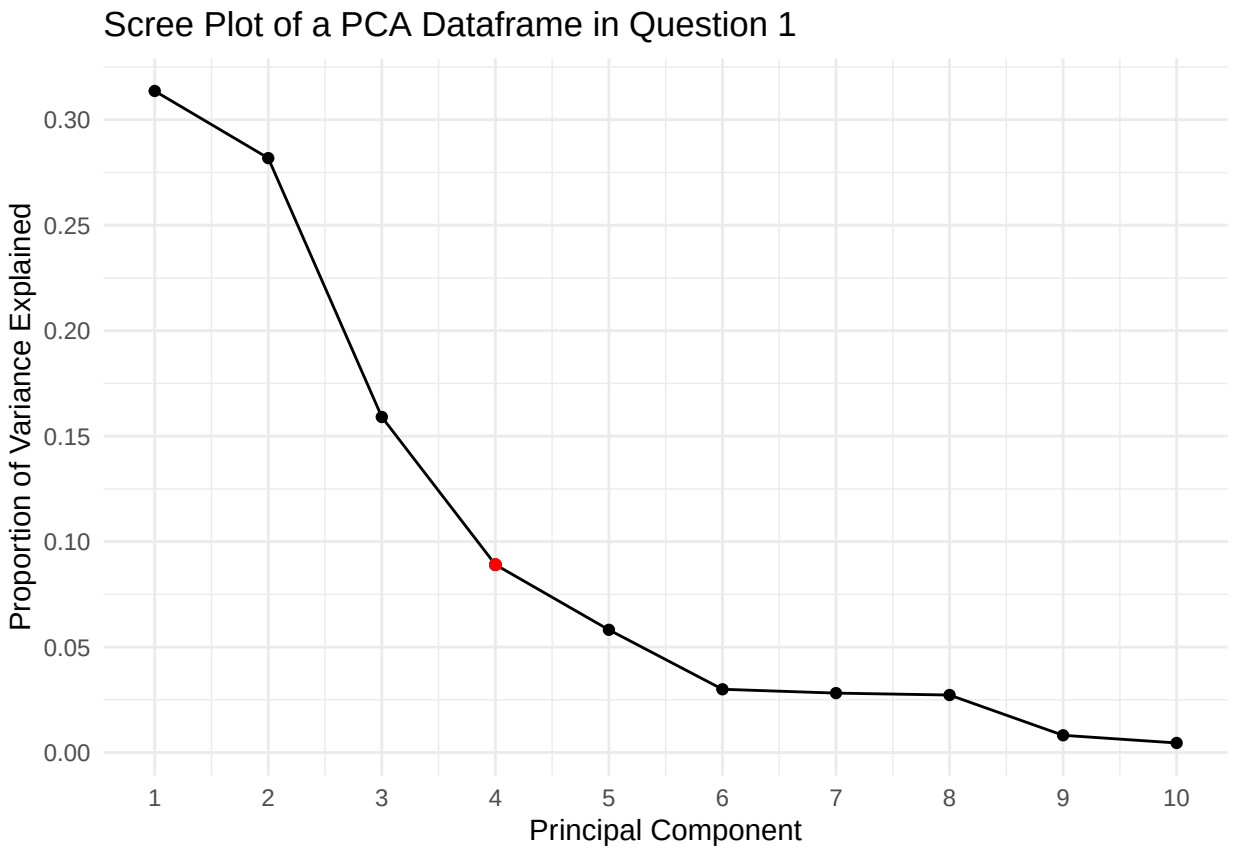
```
# Creating a PCA dataframe in long format
pca_long <- tibble(
  SD = c(3.55, 2.41, 1.82, 1.31, 1.05, 0.86, 0.81, 0.79, 0.72, 0.70),
  VAR = c(3.45, 3.10, 1.75, 0.98, 0.64, 0.33, 0.31, 0.30, 0.09, 0.05),
  PCA = paste0("PCA", 1:10),
  PVE = VAR / sum(VAR)
)

# Creating a PCA dataframe in a wide format
pca_wide <- pca_long %>%
  select(-PVE) %>%
  pivot_longer(SD:VAR) %>%
  pivot_wider(names_from = PCA,
              values_from = value)

# Computing the total variance: the total variance is the sum of all variances of all individual princi
# Source: https://ro-che.info/articles/2017-12-11-pca-explained-variance
cat("The total variance of the following PCA output is", sum(pca_long$VAR))
```

```
## The total variance of the following PCA output is 11
```

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

- *Notes: A Scree Plot is a graphical representation of the percentages of variation that each PC accounts for. Accordingly, I have put PVE on the y-axis per the definition of a scree plot*

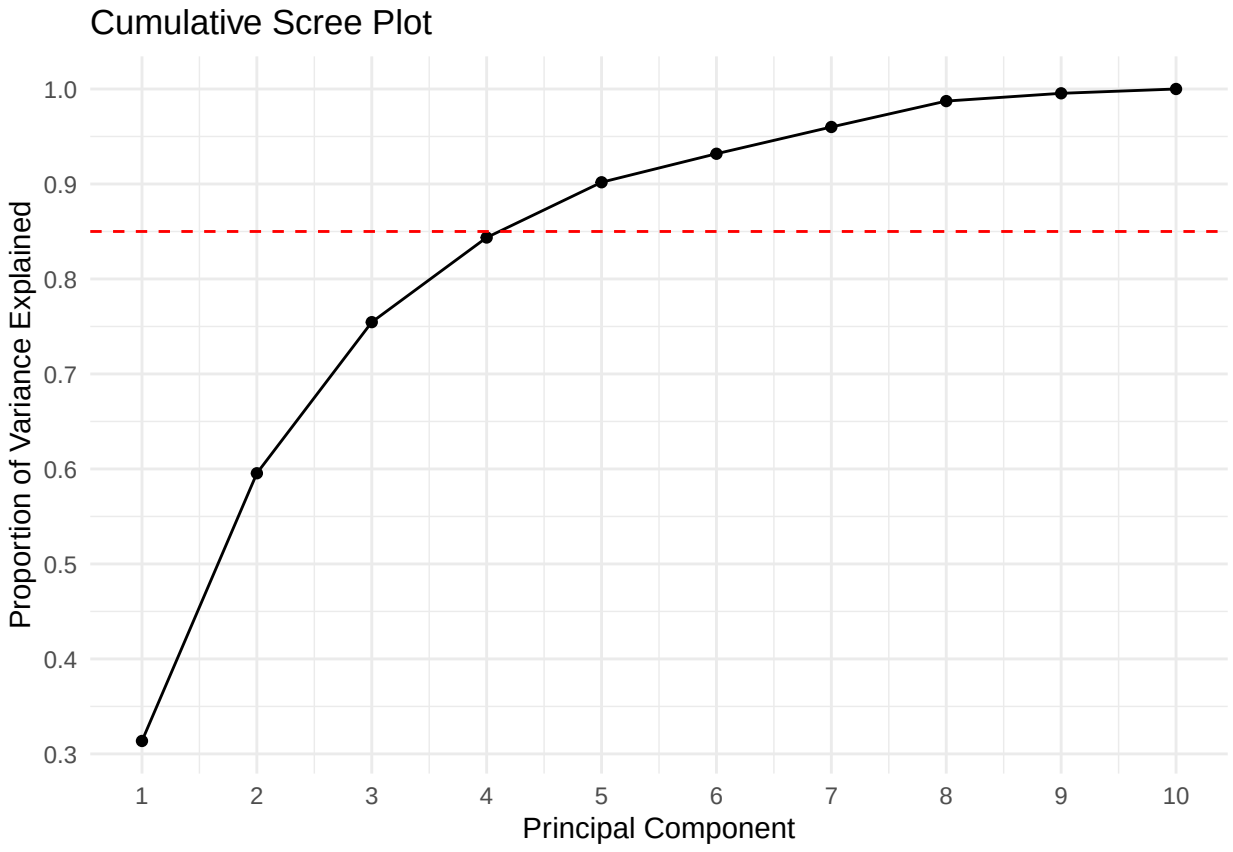
```r
# Source: https://afit-r.github.io/pca

# Scree Plot
pca_long %>%
  mutate(pca_level = as.integer((substr(PCA, 4, 5)))) %>%
  ggplot(aes(x = pca_level, y = PVE)) +
  geom_point() +
  geom_line(group = 1) +
  geom_point(aes(x = 4, y = pca_long$PVE[4]) , color = "red") +
  theme_minimal() +
  labs(
    title = "Scree Plot of a PCA Dataframe in Question 1",
    x = "Principal Component",
    y = "Proportion of Variance Explained"
  ) +
  scale_x_continuous(breaks = 1:10) +
  scale_y_continuous(breaks = seq(0, 0.35, by = 0.05))
```



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?

- **Answer: As mentioned in the ISLR textbook, there is no well-accepted objective way of selecting the appropriate principal components. However, using the eyeball technique where we look at the elbow point, it can be said that the first four principle components characterize these data well and the appropriate dimensionality of my new reduced data space be 4. The first four principle components account for approximately 85% of the total variation and this is desirable. Furhter consideration of adding more principal components require more information about the number of predictiors in the data and some domain knoweldge. (Source: ISLR 384~385).**

```
pca_long %>%
  mutate(pca_level = as.integer((substr(PCA, 4, 5)))) %>%
  ggplot(aes(x = pca_level, y = cumsum(PVE))) +
  geom_point() +
  geom_line(group = 1) +
  theme_minimal() +
  labs(
    title = "Cumulative Scree Plot",
    x = "Principal Component",
    y = "Proportion of Variance Explained"
  ) +
  scale_x_continuous(breaks = 1:10) +
  scale_y_continuous(breaks = seq(0, 1, by = 0.1)) +
  geom_hline(yintercept = 0.85, linetype = 2, color = "red")
```



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 $\forall X \in \{1, 2, 3, 4\}$.

- **Answer: Since the question asks the Euclidean distance between each of the following observation including 0, I have also added the Euclidean from the origin!**

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	...

```
# Creating the Euclidean distance datafrmae
df_euc <- tibble(
  i = 0:5,
  x1 = c(0, 2, 1, 1, 3, -3),
  x2 = c(0, 2, 1, -2 , 3, 2),
  x3 = c(0, 3, -2, -2, 2, -1),
  x4 = c(0, 1, 2, -1, 2, 1)
)

# Calculating the euclidena distance
dist(df_euc[, -1], method = "euclidean") %>%
  as.matrix() %>%
  as_tibble() %>%
  mutate(i = 0:5) %>%
  select(i,everything()) %>%
  set_names(c("i", 0:5)) %>%
  knitr::kable(
    caption = "The Euclidean Distance Between Each of The Observations (0 = Origin)"
  )
```

Table 3: The Euclidean Distance Between Each of The Observations (0 = Origin)

i	0	1	2	3	4	5
0	0.000000	4.242641	3.162278	3.162278	5.099019	3.872983
1	4.242641	0.000000	5.291503	6.782330	2.000000	6.403124
2	3.162278	5.291503	0.000000	4.242641	4.898980	4.358899
3	3.162278	6.782330	4.242641	0.000000	7.348469	6.082763
4	5.099019	2.000000	4.898980	7.348469	0.000000	6.855655
5	3.872983	6.403124	4.358899	6.082763	6.855655	0.000000

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.

```
## Tidy way ##

# Loading the data
data <- readRDS(here("data/anes.rds"))

# Fitting a PCA using a non-tidy way
pca_fit <- data[,-36] %>%
  scale() %>%
  prcomp(); summary(pca_fit)
```

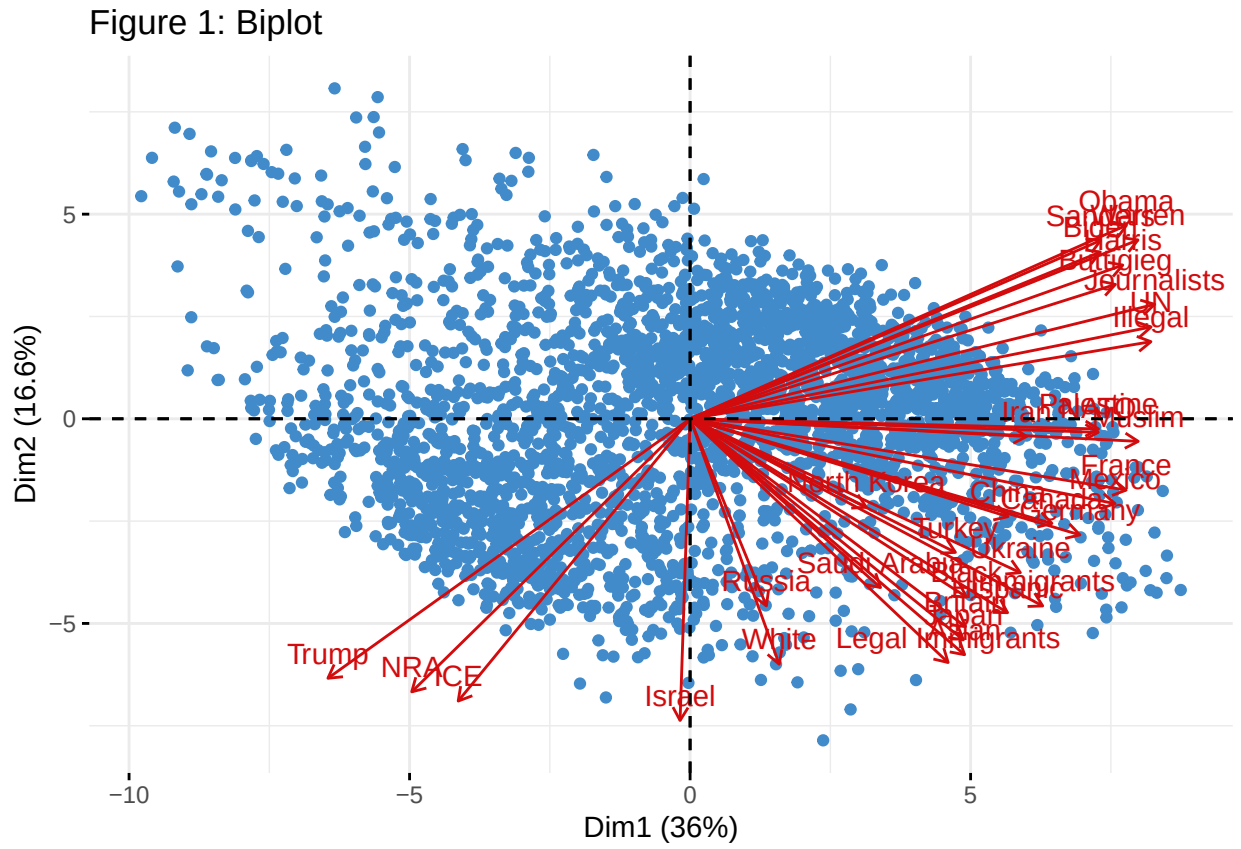
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.5491 2.4082 1.82408 1.30799 1.04776 0.86327 0.81279
## Proportion of Variance 0.3599 0.1657 0.09506 0.04888 0.03137 0.02129 0.01888
## Cumulative Proportion  0.3599 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.

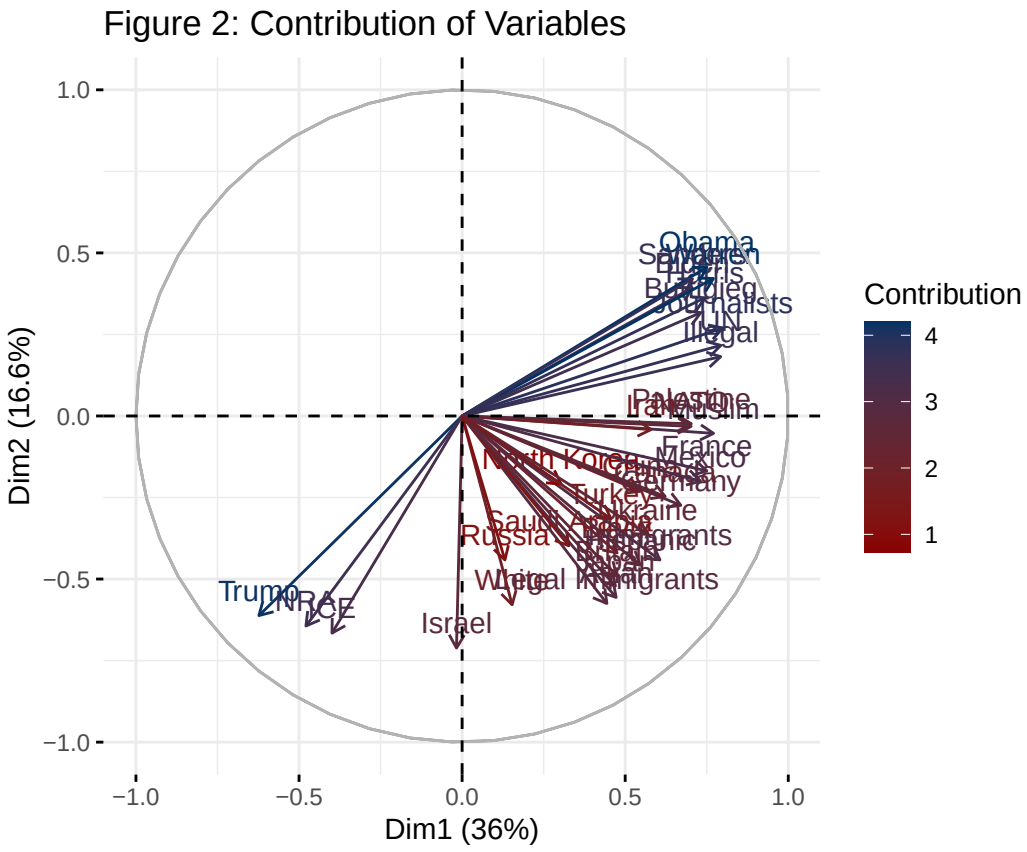
- Anwser: The first biplot figure shows the general results from the principal component analysis. The x-axis represents PC1 and the y-axis represents PC2. The percentages in the parantheses show the percentage of variation explained by the principle components. The blue points represent component scores and the red lines represent the first two principal component loading vectors.
- The second figure shows the importance of variables as indicated by the color. For instance, it is possible to determine that the first loading vector weights heavily on the variables on the far right of

PC1 and the variables on the second loading vectors weights heavily on the variables on the far bottom of PC2 (notice the scale on each axis!). The details are further illustrated in the contribution figures for each principal component. Another interesting to note is that most loading vectors have positive values of PC1 (except for Trump, NRA, ICE, and Israel) and thus indiciate positive some correlation along PC1. Conversely, we can see that most loading vectors have negative values of PC2. In closer examiniation, it is also possible to see that there are largely three sets of arrows pointing in three different directions without any arrow directed to the top-left.

```
# Figure1 : biplot
pca_fit %>%
  fviz_pca_biplot(label = "var",
                  col.var = amerika_palettes$Republican[2],
                  col.ind = amerika_palettes$Democrat[3]) +
  labs(title = "Figure 1: Biplot")
```

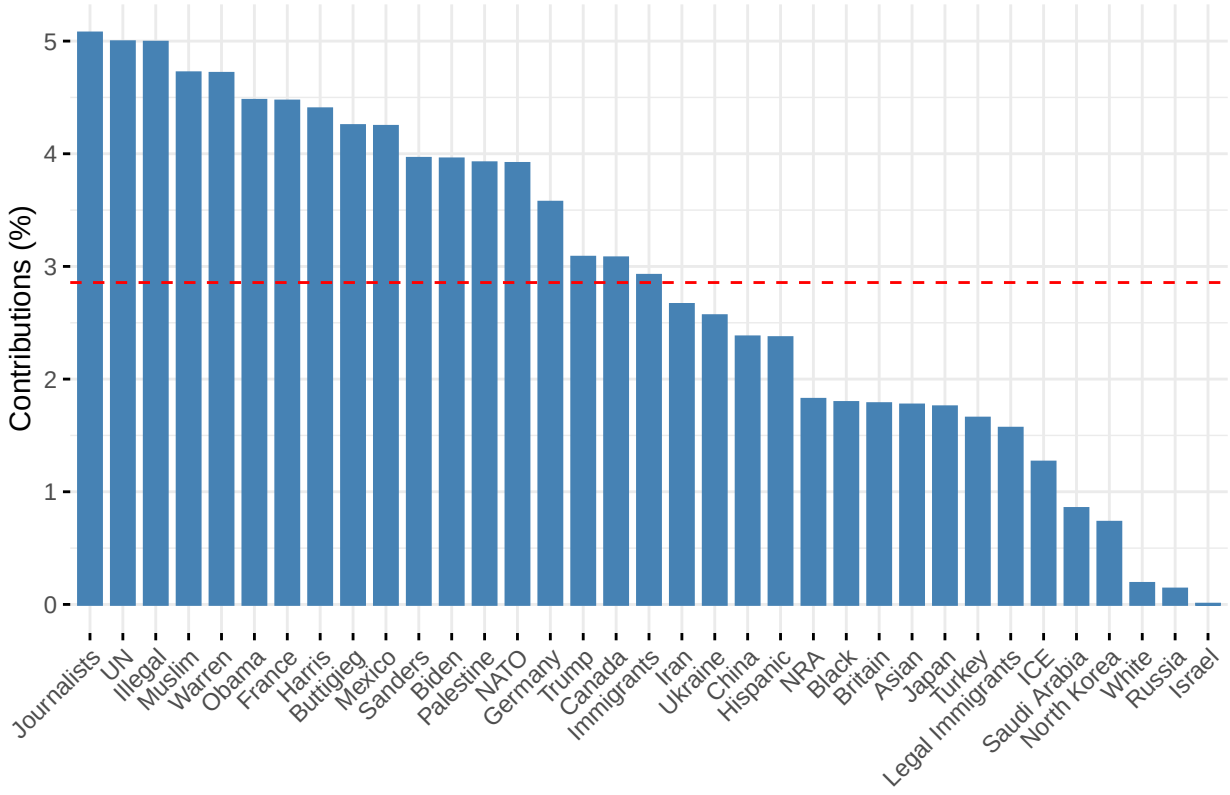



```
# Figure2 : Pca
pca_fit %>%
  fviz_pca_var(col.var = "contrib") +
  scale_color_gradient(high = amerika_palettes$Democrat[1],
                       low = amerika_palettes$Republican[1]) +
  labs(color = "Contribution",
       title = "Figure 2: Contribution of Variables")
```



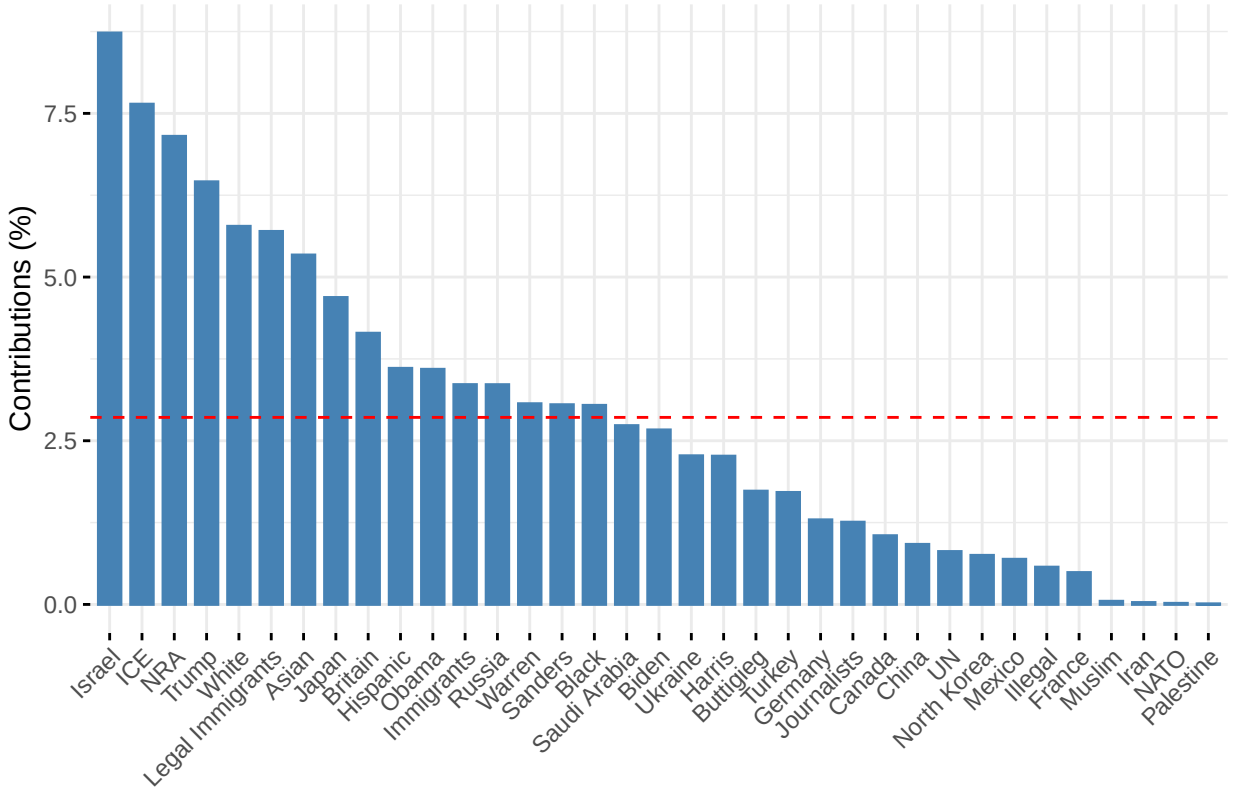

```
# Figure 2: PC1 (closer examination)
fviz_pca_contrib(pca_fit, choice = "var", axes = 1) +
  labs(
    title = "Figure 2: Contribution of Variables for Principal Component 1"
  )
```

Figure 2: Contribution of Variables for Principal Component 1




```
# Figure 2: PC2 (closer examination)
fviz_pca_contrib(pca_fit, choice = "var", axes = 2) +
  labs(
    title = "Figure 2: Contribution of Variables for Principal Component 2"
  )
```

Figure 2: Contribution of Variables for Principal Component 2



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.

- The hard partitional solution requires observations to be mutually exclusive and non-overlapping like in k-means clusters or k-medians. In these methods, the clusteres are assigned by using the distance metrics where the selected distance metric (such as Euclidean distance) within clusteres and between observations and the center of the cluster is minimized to ensure no overlap between clusters. Specifically, the objective function for optimization can be characterized as (Source: Lecture on Clustering):

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

- The soft partitional solution, however, relaxes these assumptions and allow for for observations to be overlap like in Gaussian Mixture Models and fuzzy C-means clustering. Although both GMM and FCM allow observations to belong to multiple clusters, there is a difference in how they assign the clusters. The GMM derive cluster probabilites for all observations across all clusters when FCM is concerned with multiple cluster assignments based on distance calucated between multiple cluster centroids (Source: Professor Waggoner's Text 45-47) In the typical GMM model ,the clusters are assigned based on probability and thus allow for some overlap.

- For instance, the probability distribution $p(x)$ as defined by (Source: Professor Waggoner's Text pg. 37):

$$p(x) = \sum_{k=1}^{K} \alpha_k N(x; \mu_k \sigma_k)$$

- As components are partitioned probablistically given by this probability function (where distribution parameters can change), some obsevatioans could be assinged to multiple clusters. Although both hard and soft partition create clusters, it can be seen that they are different at a substantive level.

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.

```
# Loading and Scaling the ANES data
data_scaled <- data[, -36] %>%
  scale() %>%
  as_tibble()
```

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.

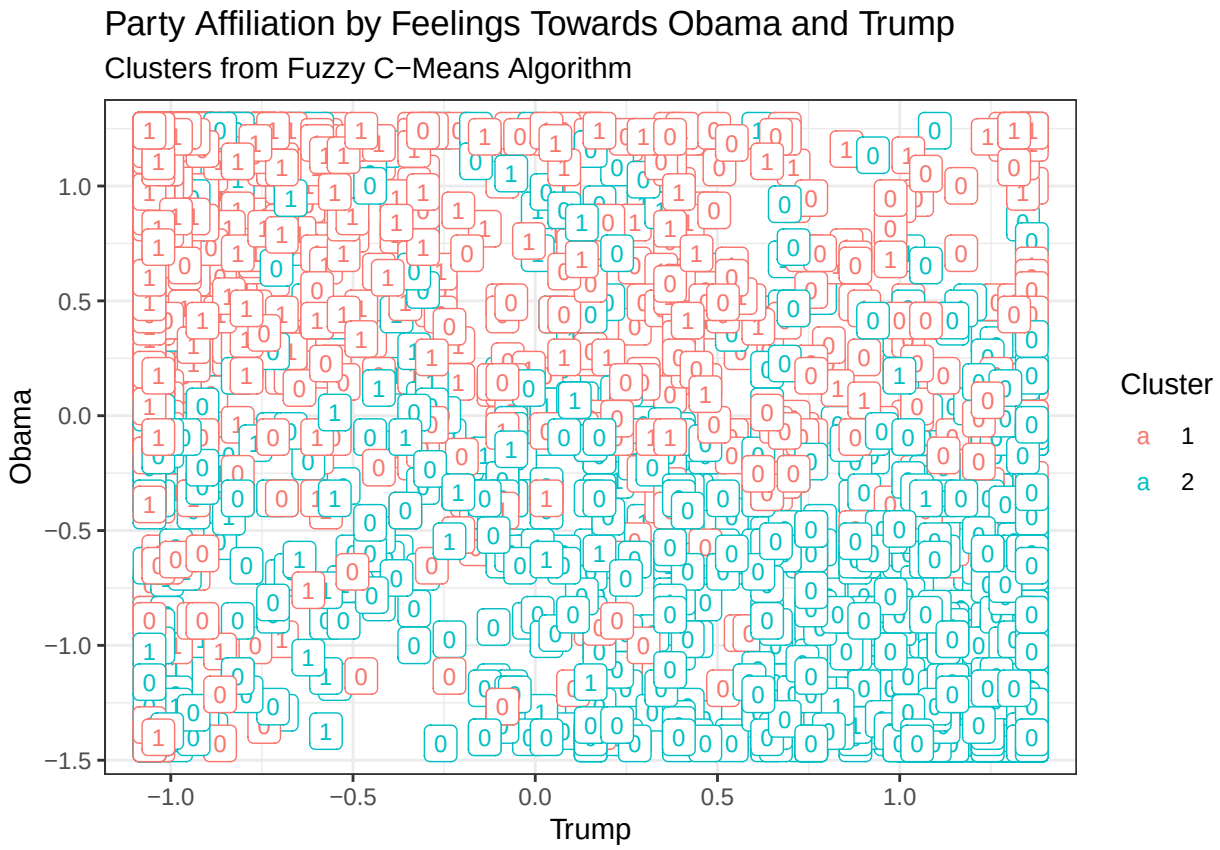
```
# Fitting an FCM algorith: (Source: Professor Waggoner's text pg. 48-49)
cm <- cmeans(data_scaled,
             centers = 2,
             m = 2)
```

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?

- Answer: Based on these patterns, it is possible to see that party affiliation has a strong influence on the clustering solution. Although party wasn't included in my clustering solution, the plot shows that many democrats were assigned to Cluster 1 and non-democrats were assigned to 0. Along the partisan dimension (most noticeably in a diagnoal fashion), those who have positive ratings for Obama and negative ratings were more likely to clustered in accordance with their party affiliation and vice versa. This plot indeed shows that the respondents group in expectd ways and cluster assignments align with the true party affiliation! It is interesting, however, to see that those who have positive feelings for both candidates were more likely to be democrats and those who had negative feelings for both canddiates were more likely to be republicans.

- I can evaluate the effctivenes of FCM for this type of task in several ways. Firstly, I can evaluate the effectiveness of FCM by using the domain knowledge. In this case, I can evaluate its effectiveness by observing how the colustering sorted the party affiliation well. I can also evluate the effectivness of FCM by using validation measures such as Fuuzy V-Measure (Entrophy Based Methods), elbow method, and average silhouette width. Finally, I can evaluate its effectiveness by comparing the outputs across diffrent clustering algorithms.

```
# Source: Professor Waggoner's text pg. 48-49
data_scaled$Cluster <- cm$cluster
data_scaled$Cluster <- as.factor(
  ifelse(data_scaled$Cluster == 1, 2, 1)
)

# Adding the democrat
data_scaled$democrat <- data$democrat

# Creating the ggplot
plot_fcm <- data_scaled %>%
  ggplot(aes(x = Trump, y = Obama, label = democrat, color = Cluster)) +
  geom_jitter() +
  geom_label(size = 3) +
  labs(
    title = "Party Affiliation by Feelings Towards Obama and Trump",
    subtitle = "Clusters from Fuzzy C-Means Algorithm"
  )

# Displaying the plot
plot_fcm
```


Party Affiliation by Feelings Towards Obama and Trump
Clusters from Fuzzy C−Means Algorithm
Obama
Trump
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
Cluster
a 1
a 2