1. Analyzing the ANES 2022 Pilot Study - PCA

The ANES 2022 Pilot Study is a cross-sectional survey conducted to test new questions under consideration for potential inclusion in the ANES 2024 Time Series Study and to provide data about voting and public opinion after the 2022 midterm elections in the United States. Information about this study is available at https://electionstudies.org/data-center/2022-pilot-study/.

The dataset contains information about the respondents profile (e.g. birthyr, gender, race, educ, marstat, ...) and answers to 235 questions from different categories.

The data contain answers to 235 questions (see PDFquestionnaire). The dataset also contains a number of variables that are not questions, but rather contain information about how the survey was conducted (see user's guide and codebook).

Loading Necessary Libraries

```
library(tidymodels)
library(tidyverse)
library(tidyclust)
library(tidymodels)
library(embed)
library(ggrepel)
library(patchwork)
```

Reading in the data

```
df <- read.csv('https://preasmyer.github.io/assets/Data/anes_pilot.csv')
```

Section 1 - PCA Analysis

(1.1) Here, I identify the feeling thermometer questions. These questions ask respondents to rate their feelings toward a number of groups on a scale from 0 to 100. The questions are listed in variables starting with `ft...`. Identify the names of all feeling thermometer questions ignoring the `ftblack` and `ftwhite` questions as these were only asked based on race of the respondent and therefore contain a large number of missing values.

```
df <- df %>%
  select(starts_with('ft'),
         -ends_with('_page_timing'),
         -starts_with('ftblack'),
         -starts_with('ftwhite'))
colnames(df)
```

```
##  [1] "fthisp"   "ftasian"  "ftfbi"    "ftscotus" "fttrump"  "ftbiden"
##  [7] "ftdem"    "ftrep"    "ftteach"  "ftfem"    "ftnfem"   "ftjourn"
## [13] "ftmen"    "ftwomen"  "fttrans"
```

(1.2) If a respondent did not answer a feeling thermometer question, the value is coded as a negative number. Here, I replaced the negative values with `NA` and removed all rows that have NA values for *any of the selected feeling thermometer questions.* (see `drop_na` function). I have about 1560 data points left after this.

```
df <- df %>%
  mutate(across(starts_with("ft"), ~ ifelse(. < 0, NA, .)))
df <- df %>% drop_na(starts_with("ft"))
dim(df)
```

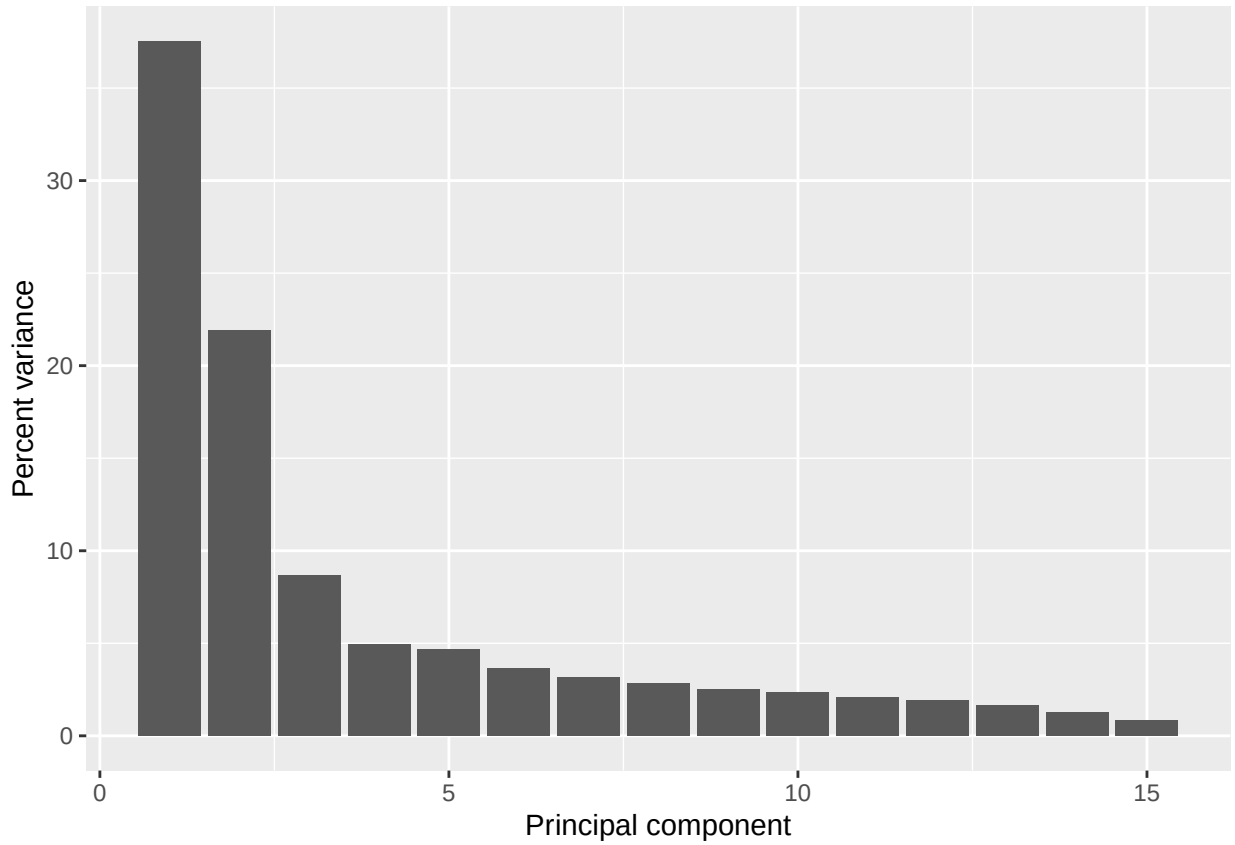
```
## [1] 1565   15
```

(1.3) In subsection 3, I perform a principal component analysis of the feeling thermometer responses using `step_pca`.

```
pca_rec <- recipe(data=df, formula= ~.) %>%
    step_normalize(all_numeric_predictors()) %>%
    step_pca(all_numeric_predictors())
tasks_pca <- pca_rec %>%
    prep() %>%
    bake(new_data=NULL) %>%
    bind_cols(df)
explained_variance <-  pca_rec %>%
    prep() %>%
    pluck("steps", 2) %>%
    tidy(type="variance")
explained_variance %>%
    pivot_wider(id_cols="component", names_from="terms", values_from="value")
```

```
## # A tibble: 15 x 5
##    component variance `cumulative variance` `percent variance`
##        <int>    <dbl>                 <dbl>              <dbl>
##  1         1    5.63                   5.63             37.5
##  2         2    3.28                   8.91             21.9
##  3         3    1.30                  10.2               8.68
##  4         4    0.739                 11.0               4.93
##  5         5    0.698                 11.7               4.65
##  6         6    0.546                 12.2               3.64
##  7         7    0.478                 12.7               3.18
##  8         8    0.424                 13.1               2.83
##  9         9    0.377                 13.5               2.51
## 10        10    0.356                 13.8               2.37
## 11        11    0.311                 14.1               2.07
## 12        12    0.287                 14.4               1.91
## 13        13    0.249                 14.7               1.66
## 14        14    0.194                 14.9               1.29
## 15        15    0.127                 15                 0.845
## # i 1 more variable: `cumulative percent variance` <dbl>
```

```
perc_variance <- explained_variance %>% filter(terms == "percent variance")
cum_perc_variance <- explained_variance %>% filter(terms == "cumulative percent variance")
ggplot(explained_variance, aes(x=component, y=value)) +
    geom_bar(data=perc_variance, stat = "identity") +
    labs(x="Principal component", y="Percent variance")
```



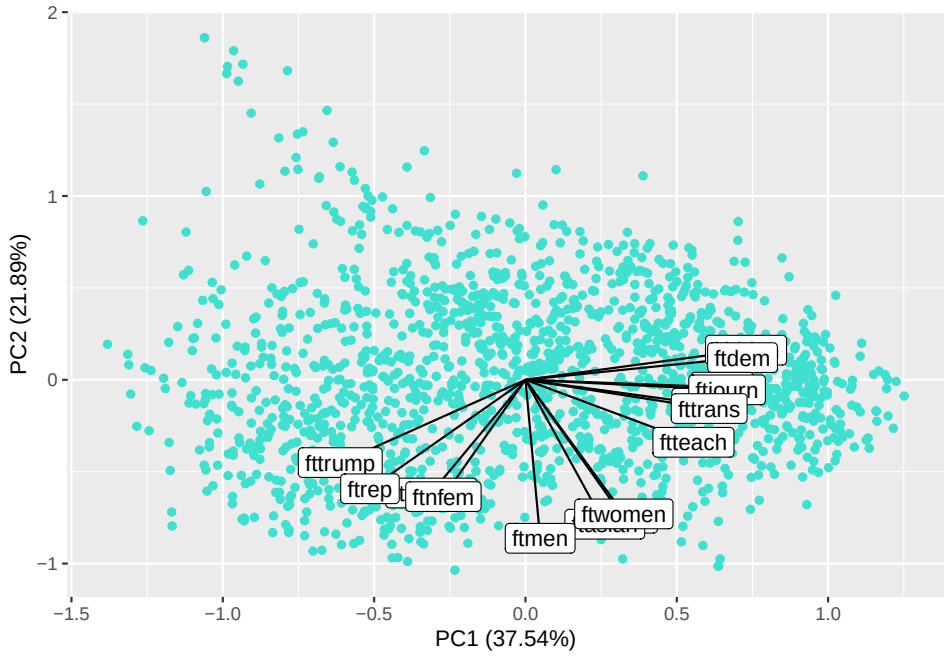
1.3 Comment:

Looking at 'elbow' in the scree plot, the first three principal components should be used. Cumulatively, these principal components explain roughly 68% of the variance, and each additional principal component used gives only small marginal gains to the explained variance (Decending in magnitude from ~5% each). Note, however, that the elbow method is not an exact science, and the amount of principal components to use may depend more heavily on the context of the problem, or the desired results.

(1.4) Here, I created a biplot using the first two components. I needed to multiply the loadings with a factor to get an improved visualization.

```
loadings <- pca_rec %>%
    prep() %>%
    pluck("steps", 2) %>%
    tidy(type="coef")
loadings <- pca_rec %>%
    prep() %>%
    pluck("steps", 2) %>%
    tidy(type="coef") %>%
    pivot_wider(id_cols="terms", names_from="component", values_from="value")
set.seed(124)
scale <- 2
ggplot(tasks_pca, aes(x=PC1/4, y=PC2/4)) +
    geom_point(color="turquoise") +
    geom_segment(data=loadings,
        aes(xend=scale * PC1, yend=scale * PC2, x=0, y=0),
```

```
        arrow = arrow(length=unit(0.15, "cm"))) +
    geom_label(data=loadings,
        aes(x=scale * PC1, y=scale * PC2, label=terms))+
  xlab('PC1 (37.54%)') + ylab('PC2 (21.89%)')
```



1.4 Comments:

Mathematically, the first Principal component represents the direction of most variance within the data in a plane. The second Principal Component represents the direction, perpendicular to the first principal component, which has the second most variance around the first principal component line. Both of these components form an n-1 dimension hyperplane. The values in n-dimensional space are then projected onto this n-1 dimensional hyperplane, and that is what the biplot is showing. The first Principal component will always be the most important in PCA, as it will always capture the most variance. This is followed sequentially by all other principal components. The vectors which represent the variables in the data show their relation to one another. Closely related variables will have smaller angles, typically well under 90 degrees, while negatively correlated variables will have angles over 90 degrees.

(1.5) The ANES 2022 Pilot Study is a rich data set. I can map the respondents profile and responses to other questions onto the principal component scatterplot.

I selected the following profile data:

- gender
- educ (education level)
- marstat (marital status)

```
df2 <- read.csv('https://gedeck.github.io/DS-6030/datasets/anes_pilot_2022_csv_20221214/anes_pilot_2022
df2 <- df2 %>%
  select(gender, educ, marstat)
```

Add steps to convert the columns into factors in your data processing pipeline. See the questionnaire for the meaning of the different factor levels.

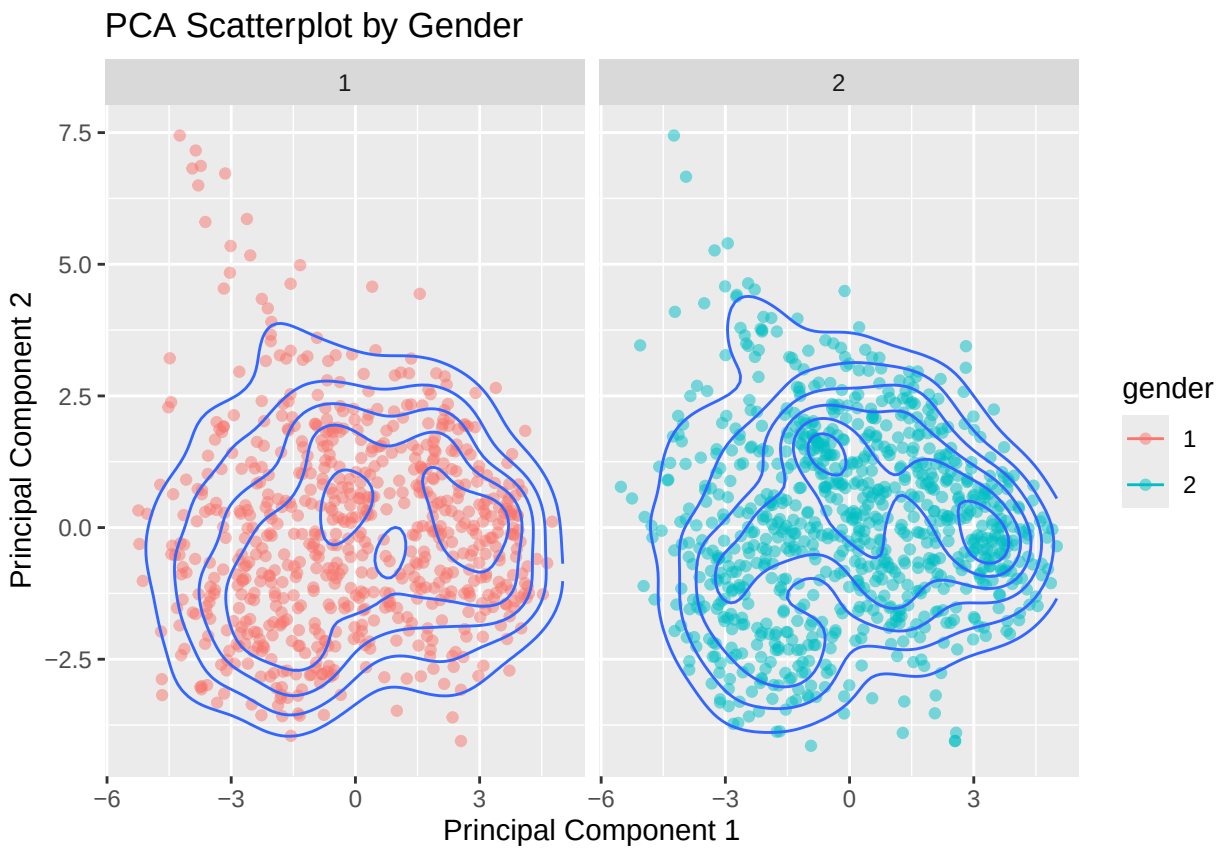
```
df2$gender <- as.factor(df2$gender)
df2$educ <- as.factor(df2$educ)
df2$marstat <- as.factor(df2$marstat)
```

Combine the data set with the transformed PCA values.

```
df2 <- df2[row.names(df2) %in% row.names(df), ]
tasks_pca2 <- cbind(tasks_pca, df2)
```

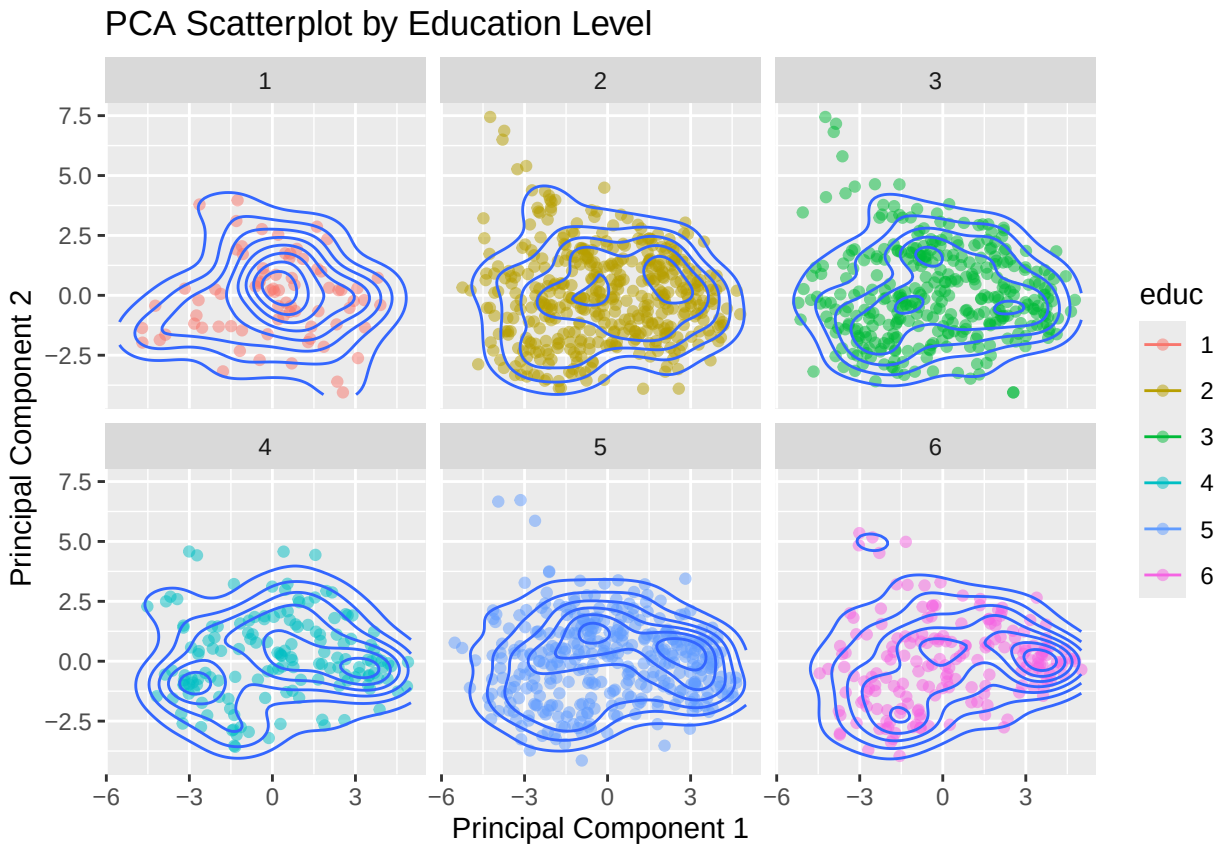
Create scatterplots of the first two components, add a `geom_density2d` layer, and use `facet_wrap` to create a separate plot for each factor level.

```
ggplot(tasks_pca2, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = gender), alpha = 0.5) +
  geom_density2d() +
  facet_wrap(~gender) +
  labs(
    title = "PCA Scatterplot by Gender",
    x = "Principal Component 1",
    y = "Principal Component 2"
  )
```

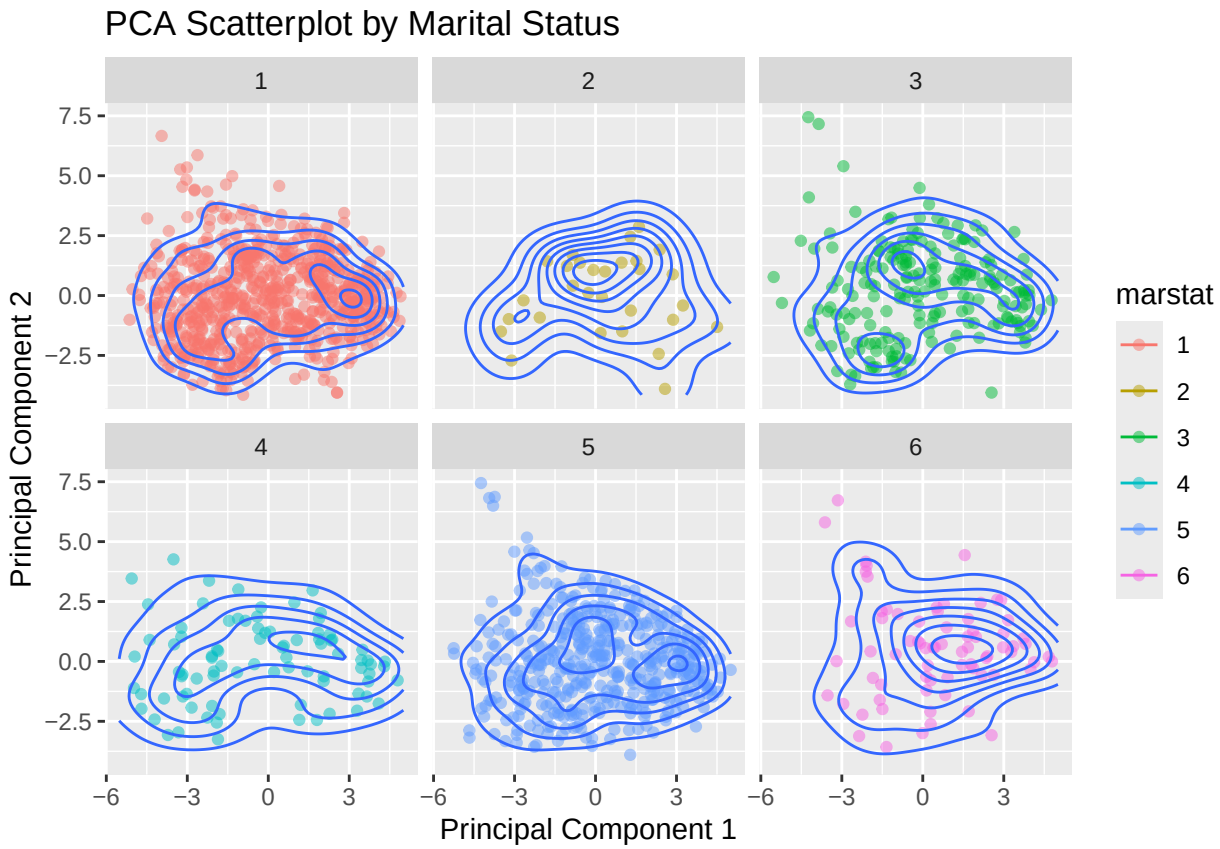



```
ggplot(tasks_pca2, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = educ), alpha = 0.5) +
  geom_density2d() +
```

```
  facet_wrap(~educ) +
  labs(
    title = "PCA Scatterplot by Education Level",
    x = "Principal Component 1",
    y = "Principal Component 2"
  )
```




```
ggplot(tasks_pca2, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = marstat), alpha = 0.5) +
  geom_density2d() +
  facet_wrap(~marstat) +
  labs(
    title = "PCA Scatterplot by Marital Status",
    x = "Principal Component 1",
    y = "Principal Component 2"
  )
```



1.5 Comments:

I notice that the distribution of gender seems very similar between both categories. There may be subtle differences on the inside of the topography, but the general pattern exists. There is also a general pattern within 'education', although the only category that exhibits a substantially different result is for the lowest level of education, though it seems the sample size in this category is smaller. For the marriage class, it seems that never married (level 5), married (level 1) and divorced (level 3) people generally follow a similar trend. While the other classes look slightly different, they appear smaller in sample size.

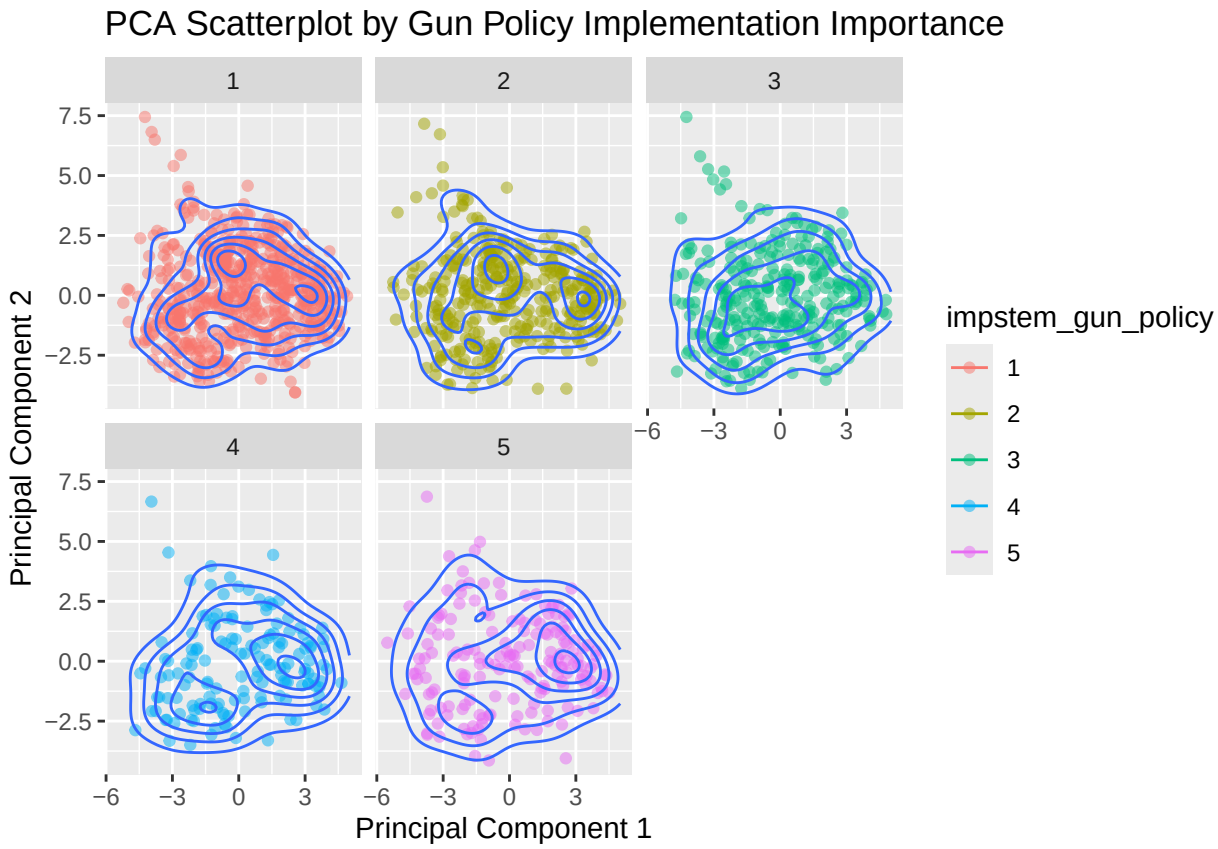
(1.6) As an extension of **(1.5)**, I will now focus on the answers to the actual questions. I formulate a hypothesis to see if I can find a correlation with the PCA analysis.

```
df3 <- read.csv('https://gedeck.github.io/DS-6030/datasets/anes_pilot_2022_csv_20221214/anes_pilot_2022_
df3 <- df3 %>%
  select(impstem_gun_policy, gunown, impstem_crime)
df3 <- df3[row.names(df3) %in% row.names(df), ]
df3$impstem_gun_policy <- as.factor(df3$impstem_gun_policy)
df3$impstem_crime <- as.factor(df3$impstem_crime)
df3$gunown <- as.factor(df3$gunown)
tasks_pca3 <- cbind(tasks_pca2, df3)
```

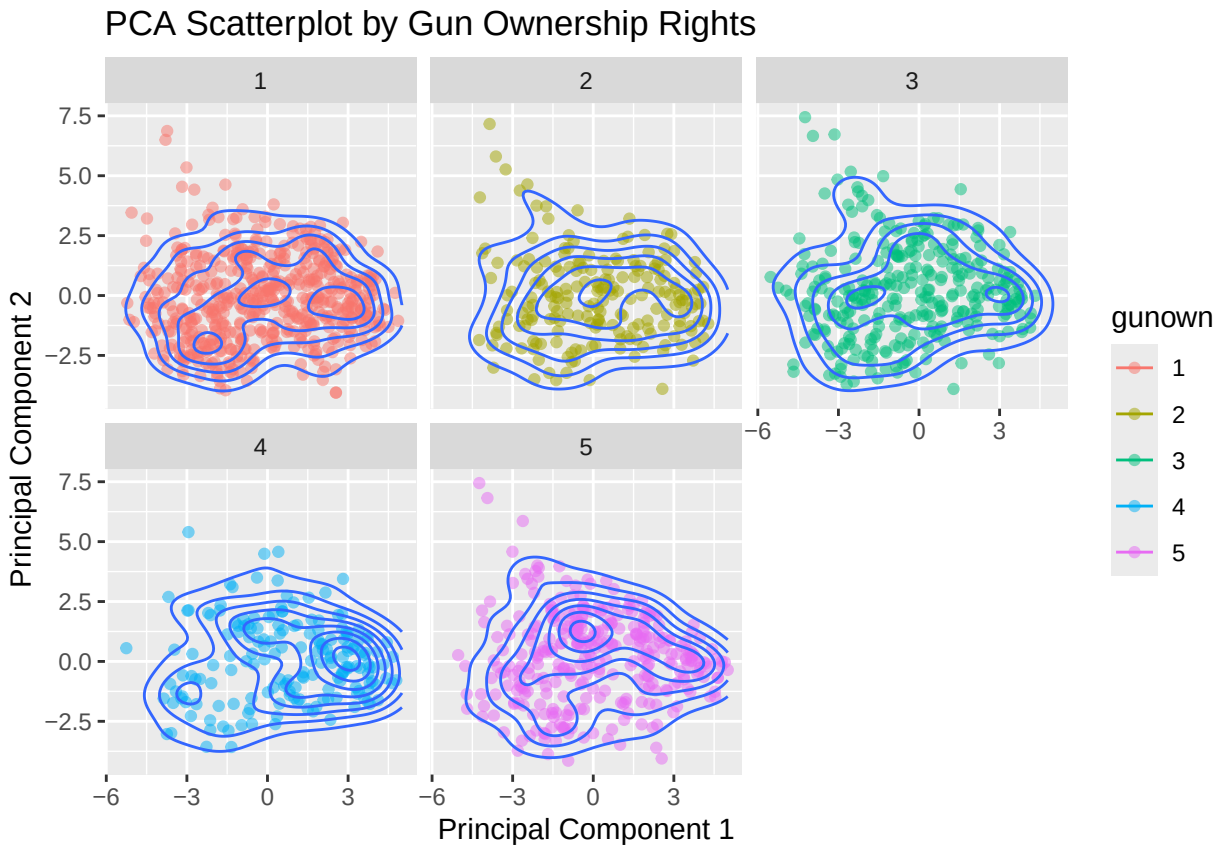
Creating Plots

```
ggplot(tasks_pca3, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = impstem_gun_policy), alpha = 0.5) +
  geom_density2d() +
```

```
  facet_wrap(~impstem_gun_policy) +
  labs(
    title = "PCA Scatterplot by Gun Policy Implementation Importance",
    x = "Principal Component 1",
    y = "Principal Component 2"
  )
```




```
ggplot(tasks_pca3, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = gunown), alpha = 0.5) +
  geom_density2d() +
  facet_wrap(~gunown) +
  labs(
    title = "PCA Scatterplot by Gun Ownership Rights",
    x = "Principal Component 1",
    y = "Principal Component 2"
  )
```




```
ggplot(tasks_pca3, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = impstem_crime), alpha = 0.5) +
  geom_density2d() +
  facet_wrap(~impstem_crime) +
  labs(
    title = "PCA Scatterplot by Importance of Adressing Crime",
    x = "Principal Component 1",
    y = "Principal Component 2"
  )
```

1.6 Comments:

I chose the 'Guns and Crime' topic, and specifically wanted to see the relationship between gun ownership rights, gun policy implementation importance, and importance for policies to address crime. Per the data-sheet, each of these variables have the exact same structure within their levels - 1-5, extremely important-Not at all Important, respectively.

The first three levels of each of these plots look extremely similar with respect to their topology, barring the third level of Gun Policy Implementation Importance. It does appear that each level of gun ownership rights tends to correlate highly with each level of addressing crime more-so than other side-by-side comparisons of the variables. Though it's worth mentioning that level 5 of the crime variable seems to have a small relative sample size and it's distribution may not be representative of population truths.

Based on the plots, I can generalize that the level of importance that people assign gun ownership rights roughly correlate to the level of importance they give toward crime as an issue in america. Of these groups, it seems that the first three levels of importance in gun policy mimic these choices in levels of importance, but not as much as the initial two variables discussed.

Section 2: Clustering

In this section, I will continue with the analysis of the ANES 2022 Pilot study and cluster the respondents based on their answers to the feeling thermometer questions.

(A) Hierarchical clustering

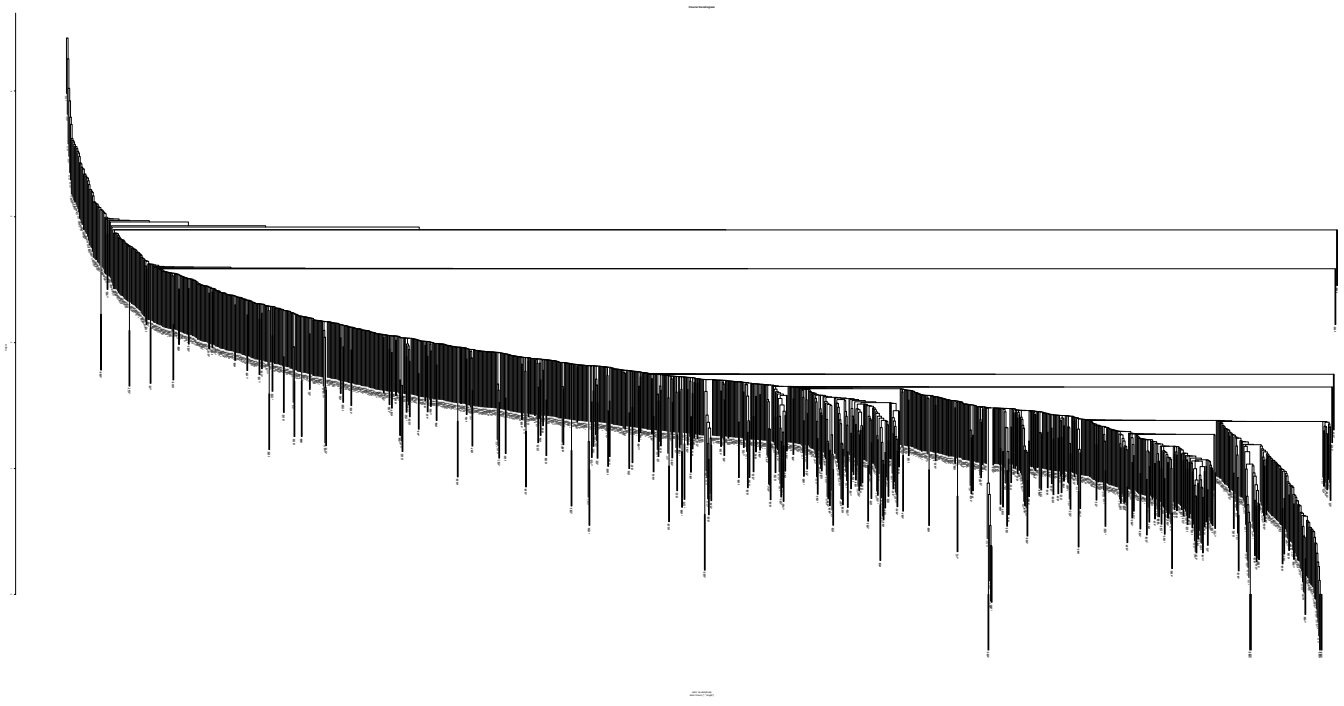
(2.1) I created a hierarchical clustering using the feeling thermometer data with the `tidyclust` package.

Creating the models

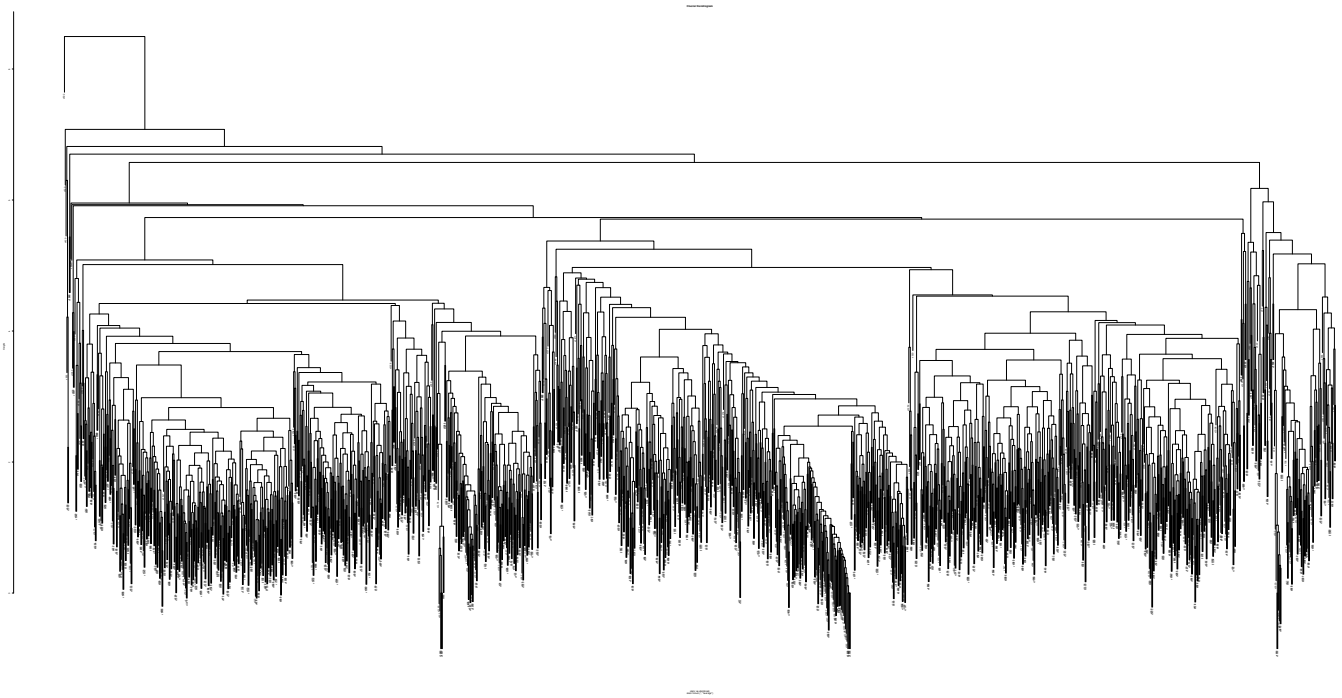
```
formula <- ~ .
clust_rec <- recipe(data=df, formula=formula) %>%
    step_normalize(all_numeric_predictors())
datafit <- clust_rec %>%
    prep() %>%
    bake(new_data=df)

complete_hier <- hier_clust(mode='partition', engine='stats', linkage_method='complete')
avg_hier <- hier_clust(mode='partition', engine='stats', linkage_method='average')
single_hier <-hier_clust(mode='partition', engine='stats', linkage_method='single')

comp_model <- complete_hier %>% fit(formula, data=datafit)
avg_model <- avg_hier %>% fit(formula, data=datafit)
single_model <- single_hier %>% fit(formula, data=datafit)
```

Creating Dendrograms of the different models

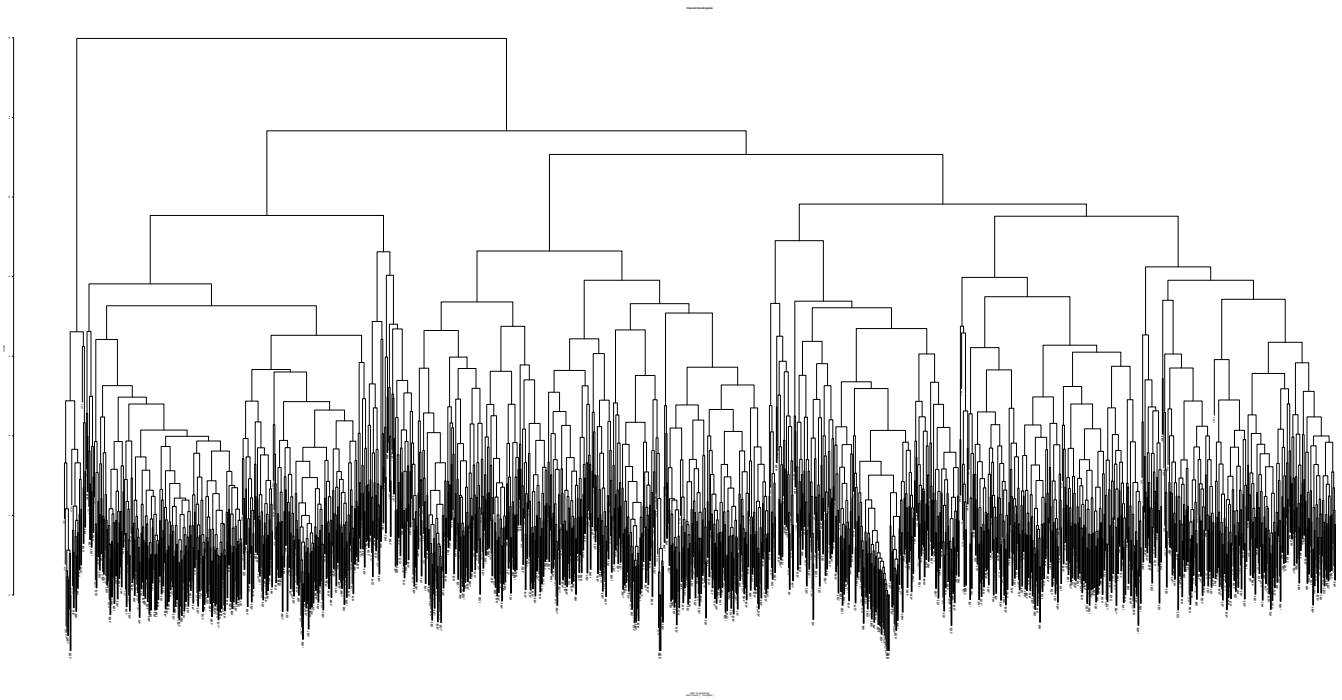
```
plot(single_model$fit)
```



```
plot(avg_model$fit)
```



```
plot(comp_model$fit)
```



The number of clusters to use depends on a few criteria and usually it is domain specific. Looking at a dendrogram is a good start and will allow us to see the separation between classes. Ideally, you want to create enough clusters to have meaningful separation between classes, but not so many clusters that every observation is its own cluster. For example, in the complete model, perhaps creating 7 classes would be preferable, as there exists a value along the vertical axis where this split would create that many classes.

Going a very short distance below this point would result in many more classes being legitimized, and going any further up would result in even few classes and more generalization (less accuracy). On the other hand, the single model might be difficult to determine this number, as any horizontal line collecting clusters for the model would result in many clusters until the much higher vertical levels. In sum, The number of clusters considered depends on the model, the dendrogram, the data, and the context. I would choose the complete model with 7 clusters in this hypothetical example, as it appears to separate classes nicely.

(B) k-means clustering

(2.2) Now, I will use k-means clustering to cluster the respondents based on their answers to the feeling thermometer questions using the `tidyclust` package.

Creating a k-means clustering with 5 clusters.

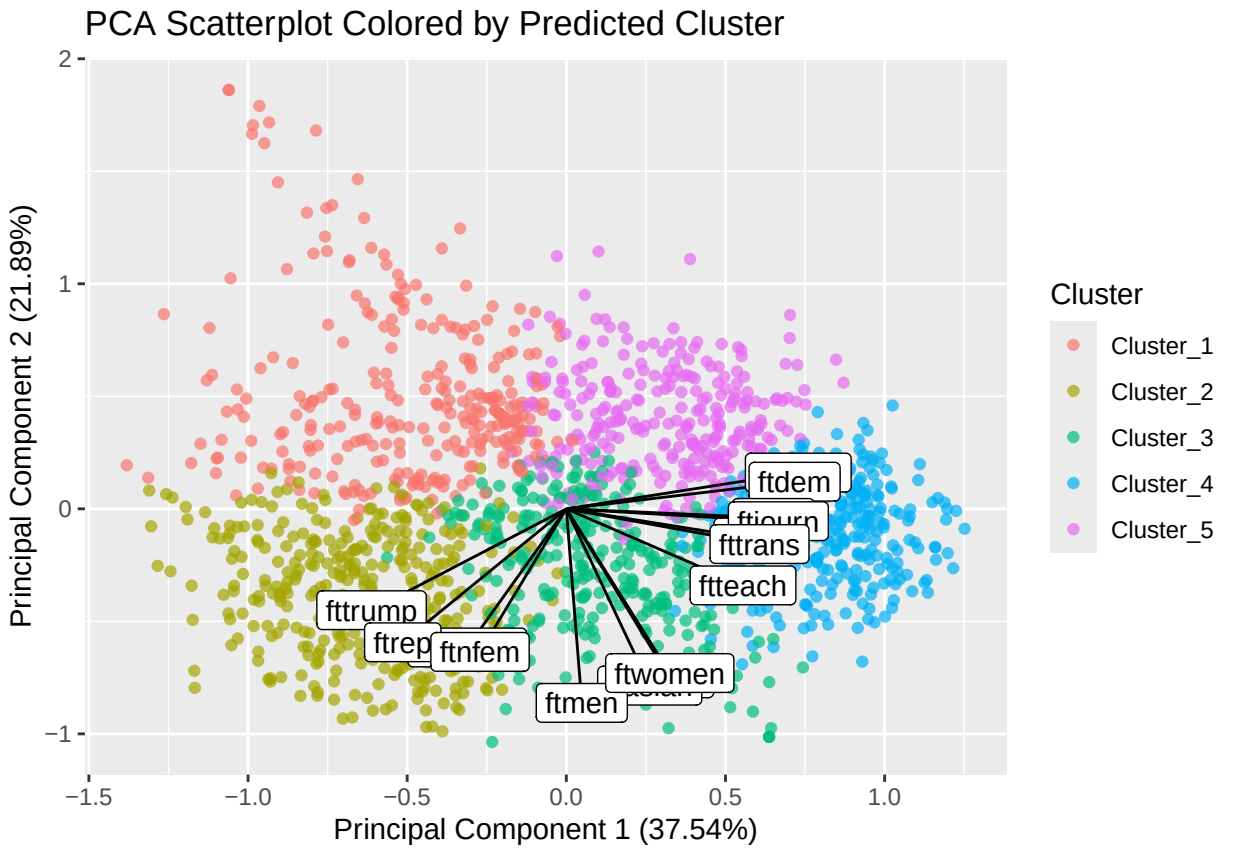
```
kmeans_rec <- recipe(~ ., data=df) %>%
    step_normalize(all_predictors())
kmeans <- k_means(num_clusters=5) %>%
    set_engine("stats") %>%
    set_mode("partition")
kmeans_wf <- workflow() %>%
    add_recipe(kmeans_rec) %>%
    add_model(kmeans)
kmeans_model <- kmeans_wf %>% fit(data=df)
```

Combining the dataset, the results from the PCA, and the k-means clustering in a tibble.

```
df4 <- augment(kmeans_model, new_data=tasks_pca3)
```

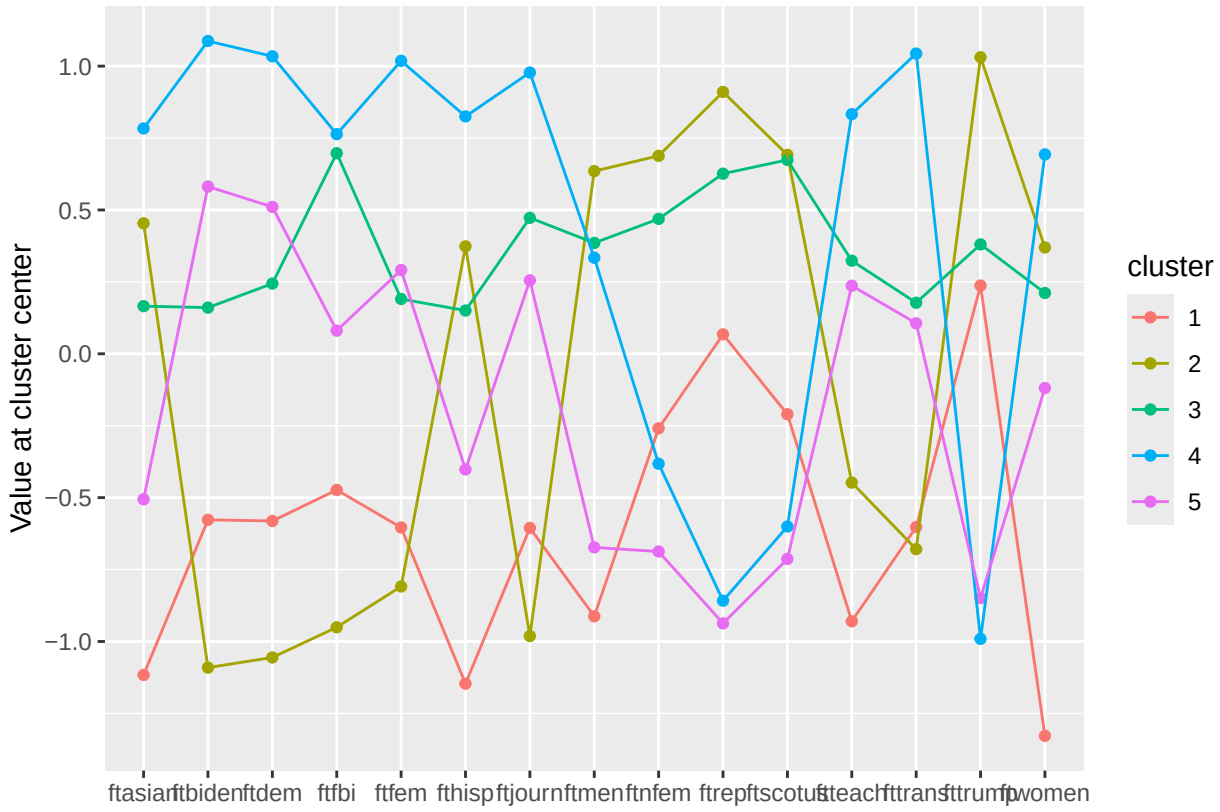
Creating a scatterplot of the first two principal components and color the points by the cluster assignment. Describe your observations.

```
df4$.pred_cluster <- as.factor(df4$.pred_cluster)
a <- ggplot(df4, aes(x = PC1 / 4, y = PC2 / 4, color = .pred_cluster)) +
    geom_point(alpha = 0.7) +
    geom_segment(data = loadings,
        aes(xend = scale * PC1, yend = scale * PC2, x = 0, y = 0),
        arrow = arrow(length = unit(0.15, "cm")), inherit.aes = FALSE) +
    geom_label(data = loadings,
        aes(x = scale * PC1, y = scale * PC2, label = terms),
        inherit.aes = FALSE) +
    labs(
        x = "Principal Component 1 (37.54%)",
        y = "Principal Component 2 (21.89%)",
        title = "PCA Scatterplot Colored by Predicted Cluster",
        color = "Cluster"
    )
a
```



Applying the `tidy` command to the fitted k-means model extracts the cluster centroids. Here, I visualize the cluster centers in a parallel coordinate plot and interpret the different clusters. It can be helpful to order the variables for the visualization (use `scale_x_discrete(limits=c("fttrans", "ftfem", ...))` where the order is defined by the `limits` argument).

```
tidy(kmeans_model) %>%
    pivot_longer(cols=c("fthisp", "ftasian", "ftfbi", "ftscotus", "fttrump", "ftbiden",
  "ftdem", "ftrep", "ftteach", "ftfem", "ftnfem", "ftjourn",
  "ftmen", "ftwomen", "fttrans")
) %>%
ggplot(aes(x=name, y=value, group=cluster, color=cluster)) +
    geom_point() +
    geom_line() +
    labs(x="", y="Value at cluster center")
```



2.2 Comments:

From the scatterplot of the PCA with data points colored by cluster, I can easily see suparation in the data, and can see different clusters aligning with certian loadings. For example, right-leaning feeling thermometer values tend to be in their own cluster, as do left-leaning FT observations. Additioinally, gender specific FT scores also tend to their own cluster in the projection.

From the parallel coordinate plot, most variables seem to take different values at the cluster centroids. This tells me that there is decent separation between clusters and less evidence of overlap between them. There is, however, one exception to this generalization in clusters 1 and 2. These clusters follow the same general trend on most variables, but differ vastly in the fthisp and ftasian variables, which may have had enough of an influence alone to separate the classes.

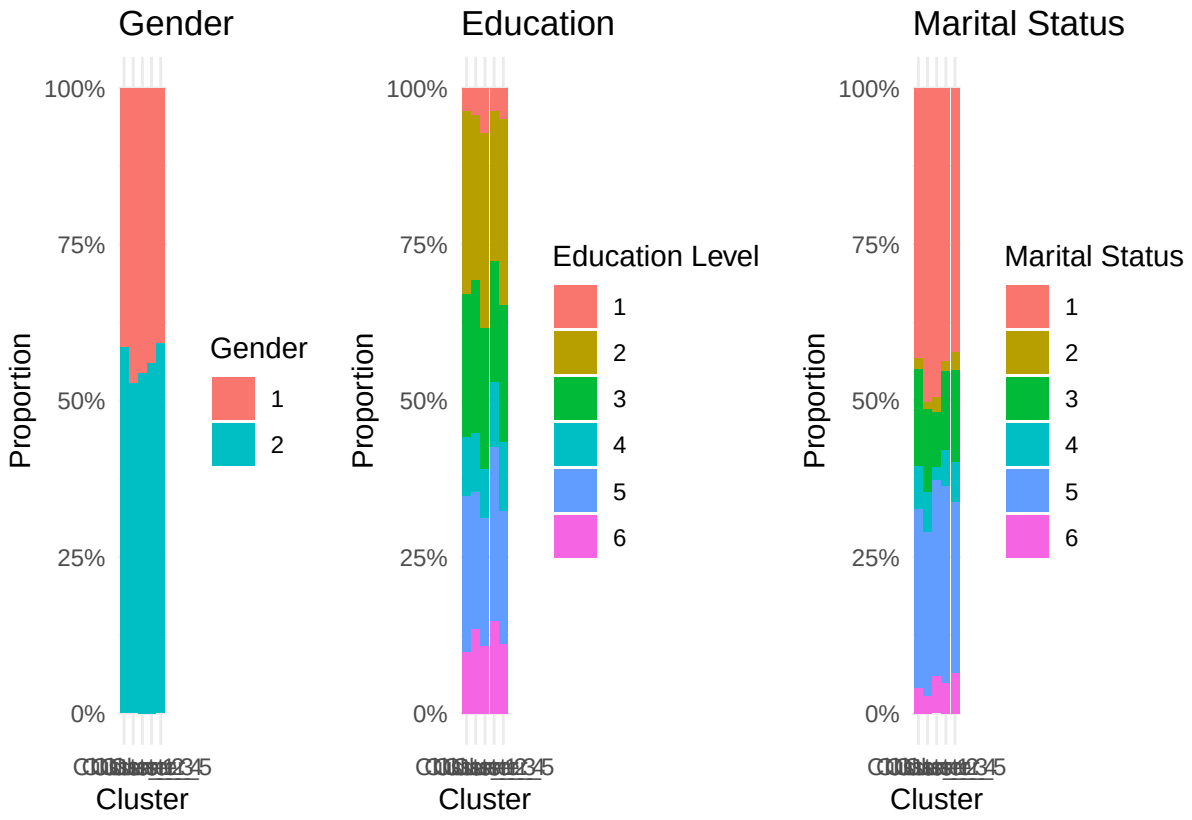
(C) Explore dataset

Characterize the different clusters.

(2.3) I will use the profile data from **(1.5)** to characterize the different clusters.

```
ggplot(df4, aes(x = .pred_cluster, fill = gender)) +
    geom_bar(position = "fill") +
    labs(
        title = "Gender",
        x = "Cluster",
        y = "Proportion",
        fill = "Gender"
    ) +
```

```
    scale_y_continuous(labels = scales::percent) +
    theme_minimal()+
ggplot(df4, aes(x = .pred_cluster, fill = educ)) +
    geom_bar(position = "fill") +
    labs(
        title = "Education",
        x = "Cluster",
        y = "Proportion",
        fill = "Education Level"
    ) +
    scale_y_continuous(labels = scales::percent) +
    theme_minimal()+
ggplot(df4, aes(x = .pred_cluster, fill = marstat)) +
    geom_bar(position = "fill") +
    labs(
        title = "Marital Status",
        x = "Cluster",
        y = "Proportion",
        fill = "Marital Status"
    ) +
    scale_y_continuous(labels = scales::percent) +
    theme_minimal()
```



2.3 Comments:

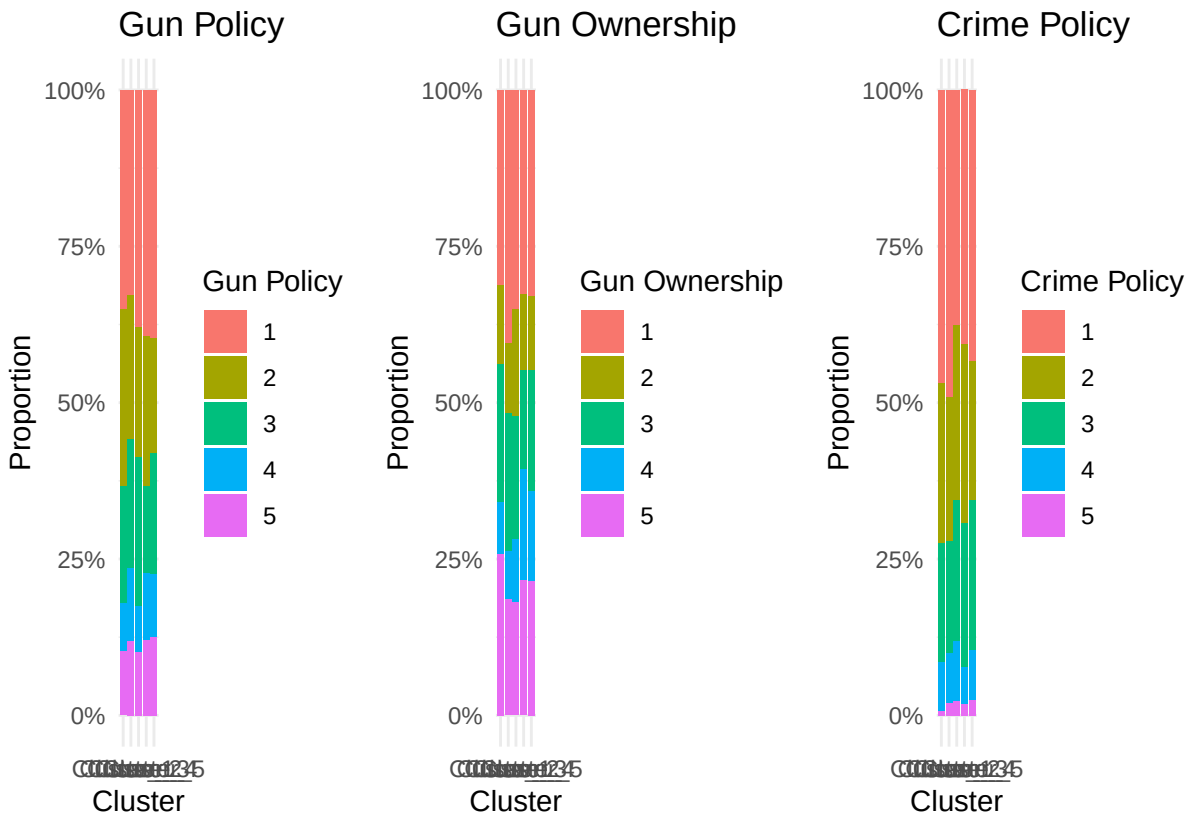
Each plot displays the distribution of the factos in each cluster. Clusters for each plot displays clusters 1 through 5, left to right, respectively.

Points from 1.6 were that 1) gender seemed to be equally distributed per PCA. 2) Education looked similar barring low levels of education (level 1). 3) Marital status was similar for married, never married, and divorced people (levels 1,3 and 5), while other classes were dissimilar.

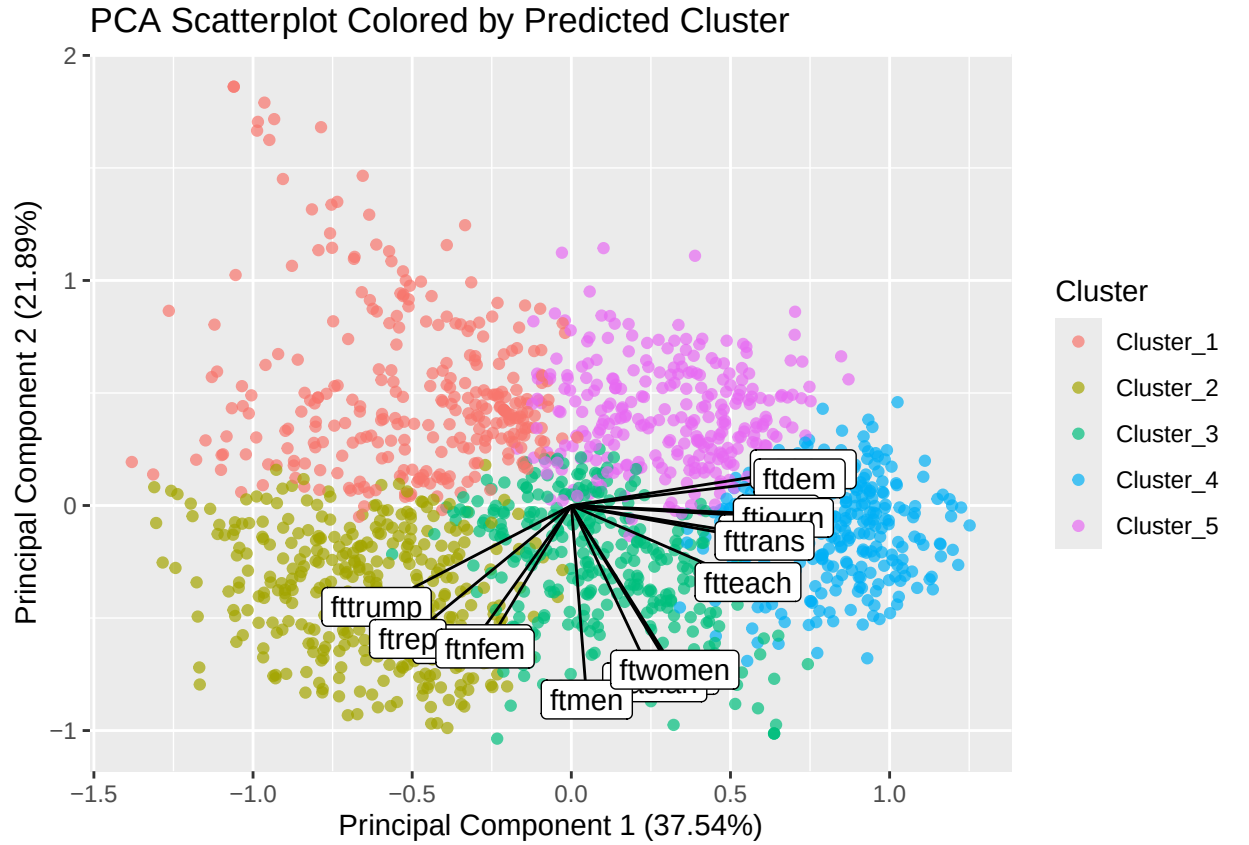
Generally, the trends seem to hold true in the bar charts between clusters. Low educated people seem to fluctuate amounts within the clusters, sometimes by twice the magnitude as other clusters. The same is true for highly educated people, though the magnitude is far less as a factor of itself. For gender, the classes appear to be somewhat even amongst all clusters. For marital status, The clusters appear to be similar for levels 1, 3 and 5. Note that the fluctuation of representation of level 1 affects the plot more than the others, however it is the most represented subgroup within the variable and small fluctuations make large proportionate impacts between levels. Interestingly with this variable, level 6 and level 4 tend to fluctuate the most, which were the non-conforming classes as stated in 1.6. Level 2 appears stable but it is a very underrepresented level.

(2.4) Now I will use the questions I had from from **(1.6)** to characterize the different clusters.

```
ggplot(df4, aes(x = .pred_cluster, fill = impstem_gun_policy)) +
    geom_bar(position = "fill") +
    labs(
        title = "Gun Policy",
        x = "Cluster",
        y = "Proportion",
        fill = "Gun Policy"
    ) +
    scale_y_continuous(labels = scales::percent) +
    theme_minimal()+
ggplot(df4, aes(x = .pred_cluster, fill = gunown)) +
    geom_bar(position = "fill") +
    labs(
        title = "Gun Ownership",
        x = "Cluster",
        y = "Proportion",
        fill = "Gun Ownership"
    ) +
    scale_y_continuous(labels = scales::percent) +
    theme_minimal()+
ggplot(df4, aes(x = .pred_cluster, fill = impstem_crime)) +
    geom_bar(position = "fill") +
    labs(
        title = "Crime Policy",
        x = "Cluster",
        y = "Proportion",
        fill = "Crime Policy"
    ) +
    scale_y_continuous(labels = scales::percent) +
    theme_minimal()
```


Gun Policy
Gun Ownership
Crime Policy
100%
75%
50%
25%
0%
Proportion
Cluster
1
2
3
4
5

a


PCA Scatterplot Colored by Predicted Cluster
Principal Component 2 (21.89%)
Principal Component 1 (37.54%)
2
1
0
−1
−1.5
−1.0
−0.5
0.0
0.5
1.0
ftdem
fttrans
ftteach
fttrump
ftrep
ftnfem
ftwomen
ftmen
Cluster
Cluster_1
Cluster_2
Cluster_3
Cluster_4
Cluster_5

2.4 Comments:

There are certain clusters that have noticeably higher and lower values at the extremes, indicating separation between clusters based on certain variable criteria. For example, Cluster 3 shows low amounts of people who think gun ownership rights are important (relative), High amounts of people who think gun ownership rights are least important, as well as a large proportion of people who think gun policies are very important. Conversely, clusters 1 and 2 see an opposite trend with lowest levels of Gun Policy Importance, and the highest average level of importance with Gun Ownership rights. These two clusters parrot some of the common conventional views of the political idologies they're comprised of. With fttrump and ftrep being correlated highly with cluster 1, and moderately with cluster 2, and ftbiden and ftdem being highly correlated with clutser 4 and moderately with part of cluster 3.

These findings generally coincided with my answer to 1.7, with Gun Ownership and Gun Policy representation correlating inversely, and Gun ownership and crime policy roughly correlating, though there are some discrepancies between the two in the most recent plot.

Thank you for reviewing my work!