Appendix: Full Code

```r
library(tidyverse)
library(cluster)
library(factoextra)  # fviz_nbclust helpers
library(ggrepel)
library(ggplot2)
library(dplyr)
library(caret)
library(tidyr)
set.seed(params$seed)
knitr::opts_chunk$set(comment = NA)
knitr::opts_chunk$set(echo = FALSE, include = FALSE)

data <- read.csv("HS119_members.csv")

#Age And Ideology Setup
# Create age column
data$age <- 2025 - data$born

# Remove rows where age cannot be calculated
data <- data %>% filter(!is.na(age))


# Convert party_code to factor with labels
data$party <- factor(data$party_code,
                     levels = c(100, 200, 328),
                     labels = c("Democrat", "Republican", "Independent"))
#Party Loyalty Setup
S118_votes <- read.csv("S118_votes.csv")
S118_members <- read.csv("S118_members.csv")
S118_rollcalls <- read.csv("S118_rollcalls.csv")
#Intra-Party Factions
infile <- params$infile
which_chamber <- params$which_chamber
k_grid <- params$k_min:params$k_max

df_raw <- readr::read_csv(infile, show_col_types = FALSE)

# Basic sanity check for expected columns
needed <- c("chamber","party_code","bioname","nominate_dim1","nominate_dim2")
missing <- setdiff(needed, names(df_raw))
if (length(missing) > 0) {
  stop(paste("Missing required columns:", paste(missing, collapse=", ")))
}

df <- df_raw %>%
  filter(chamber == which_chamber) %>%
  select(bioname, party_code, chamber, nominate_dim1, nominate_dim2) %>%
  filter(!is.na(nominate_dim1), !is.na(nominate_dim2))

df_dem <- df %>% filter(party_code == 100)   # Democrats
df_gop <- df %>% filter(party_code == 200)   # Republicans
```

```r
tibble(
  party = c("Democrats","Republicans"),
  N = c(nrow(df_dem), nrow(df_gop))
)
fit_party_clusters <- function(df_party, party_label, k_grid = 2:5) {
  if (nrow(df_party) < min(k_grid)) {
    warning(paste0("Too few members in ", party_label, " for k>=2. Skipping."))
    return(NULL)
  }
  X <- df_party %>% select(nominate_dim1, nominate_dim2)

  # Elbow
  wss <- sapply(k_grid, function(k){
    km <- kmeans(X, centers = k, nstart = 50)
    km$tot.withinss
  })

  # Silhouette
  sil_avgs <- sapply(k_grid, function(k) {
    km <- kmeans(X, centers = k, nstart = 50)
    sil <- silhouette(km$cluster, dist(X))
    mean(sil[, "sil_width"])
  })

  #k_best <- k_grid[which.max(sil_avgs)]
  k_best <- 3
  km <- kmeans(X, centers = k_best, nstart = 200)
  df_out <- df_party %>% mutate(cluster = factor(km$cluster))

  centers <- as_tibble(km$centers) %>% mutate(cluster = factor(1:n()))

  p_elbow <- tibble(k = k_grid, wss = wss) %>%
    ggplot(aes(k, wss)) +
    geom_line() + geom_point() +
    labs(title = paste0(party_label, " - Elbow (WSS)"),
         x = "k", y = "Total within-cluster SS") +
    theme_minimal()

  p_sil <- tibble(k = k_grid, silhouette = sil_avgs) %>%
    ggplot(aes(k, silhouette)) +
    geom_line() + geom_point() +
    labs(title = paste0(party_label, " - Average Silhouette by k"),
         x = "k", y = "Average silhouette width") +
    theme_minimal()

  p_map <- df_out %>%
  ggplot(aes(nominate_dim1, nominate_dim2, color = cluster)) +
  geom_point(size = 3, alpha = 0.9) +  # larger, clear points
  # Mark cluster centers with an X
  geom_point(data = centers, aes(nominate_dim1, nominate_dim2, color = cluster),
             size = 6, shape = 4, stroke = 1.5, inherit.aes = FALSE) +
  labs(
    title = paste0(party_label, " - NOMINATE Cluster Map (k=", k_best, ")"),
    subtitle = "Each point = Member of Congress; Color = Faction cluster center (X)",
    x = "NOMINATE dim1 (Liberal - Conservative)",
    y = "NOMINATE dim2 (Populist - Establishment)",
    color = "Faction Cluster"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )


  list(
    df = df_out,
    centers = centers,
    k_best = k_best,
    silhouette_table = tibble(k = k_grid, avg_silhouette = sil_avgs),
    elbow_plot = p_elbow,
    silhouette_plot = p_sil,
    map_plot = p_map
  )
}

# Heuristic labeling rules - tweak after inspecting centers
label_dem_cluster <- function(center_dim1, center_dim2) {
  if (center_dim1 <= -0.55) return("Progressive")
  if (center_dim1 >= -0.30 && center_dim2 >= 0.05) return("Blue Dog / Moderate-Establishment")
  return("Mainline Moderate")
}

label_gop_cluster <- function(center_dim1, center_dim2) {
  if (center_dim1 >= 0.60 && center_dim2 <= 0.00) return("Trump-aligned / Populist")
  if (center_dim1 <= 0.35 && center_dim2 >= 0.05) return("Moderate / Establishment")
  return("Traditional Conservative")
}

apply_labels <- function(res, party = c("D","R")) {
  party <- match.arg(party)
  if (is.null(res)) return(NULL)
  centers_labeled <- res$centers %>%
    rowwise() %>%
    mutate(
      faction = if (party == "D")
        label_dem_cluster(nominate_dim1, nominate_dim2)
      else
        label_gop_cluster(nominate_dim1, nominate_dim2)
    ) %>%
    ungroup()
  df_labeled <- res$df %>%
    left_join(centers_labeled %>% select(cluster, faction), by = "cluster")
  list(members = df_labeled, centers = centers_labeled)
```

```
}
res_dem <- fit_party_clusters(df_dem, "Democrats", k_grid)
if (!is.null(res_dem)) {
  print(res_dem$elbow_plot)
}
if (!is.null(res_dem)) {
  print(res_dem$silhouette_plot)
}
if (!is.null(res_dem)) {
  print(res_dem$map_plot)
}
if (!is.null(res_dem)) {
  out_dem <- apply_labels(res_dem, "D")

  cat("## Cluster Centers (Democrats)
")
  print(out_dem$centers %>% arrange(faction))
  "~/Documents/GitHub/Stat_380_Final_Proj/HS119_members.csv"
  cat("## Counts by Faction (Democrats)
")
  print(out_dem$members %>% count(faction, sort = TRUE))
}
res_gop <- fit_party_clusters(df_gop, "Republicans", k_grid)
if (!is.null(res_gop)) {
  print(res_gop$elbow_plot)
}
if (!is.null(res_gop)) {
  print(res_gop$silhouette_plot)
}
if (!is.null(res_gop)) {
  print(res_gop$map_plot)
}
if (!is.null(res_gop)) {
  out_gop <- apply_labels(res_gop, "R")

  cat("## Cluster Centers (Republicans)
")
  print(out_gop$centers %>% arrange(faction))

  cat("## Counts by Faction (Republicans)
")
  print(out_gop$members %>% count(faction, sort = TRUE))
}
closest_to_centers <- function(df_members, centers_tbl, top_n = 8) {
  if (is.null(df_members)) return(invisible(NULL))
  out <- list()
  for (cl in centers_tbl$cluster) {
    cen <- centers_tbl %>% filter(cluster == cl) %>%
      select(nominate_dim1, nominate_dim2) %>% as.numeric()
    tmp <- df_members %>%
      filter(cluster == cl) %>%
      mutate(dist_to_center = sqrt((nominate_dim1 - cen[1])^2 + (nominate_dim2 - cen[2])^2)) %>%
      arrange(dist_to_center) %>%
      slice_head(n = top_n) %>%
      select(bioname, nominate_dim1, nominate_dim2, faction, dist_to_center)
    out[[as.character(cl)]] <- tmp
  }
  out
}

if (exists("out_dem") && !is.null(out_dem)) {
  cat("## Democrats - Examples Closest to Centers
")
  print(closest_to_centers(out_dem$members, out_dem$centers, 10))
}
if (exists("out_gop") && !is.null(out_gop)) {
  cat("## Republicans - Examples Closest to Centers
")
  print(closest_to_centers(out_gop$members, out_gop$centers, 10))
}
#Age and Ideology Question

# Compare with chamber
ggplot(data, aes(x = chamber, y = age, color = party)) +
  geom_point(size = 1) +
  labs(title = "Age and Chamber by Party",
       x = "Chamber",
       y = "Age",
       color = "Party")

# Compare with State
ggplot(data, aes(x = state_abbrev, y = age, color=party)) +
  geom_point(size = 1) +
  labs(title = "Age and State by Party",
       x = "State",
       y = "Age")

# Compare with Ideological Score
ggplot(data, aes(x = nominate_dim1, y = age, color = party)) +
  geom_point(size = 1) +
  labs(title = "Age and Ideology by Party",
       x = "Ideology Score",
       y = "Age")

# Compare with Ideological Radicality Score
ggplot(data, aes(x = nominate_dim2, y = age,color = party)) +
  geom_point(size = 1) +
  labs(title = "Age and Ideology Radicality by Party",
       x = "Ideology Radicality Score",
       y = "Age")


# Compare with Ideological Score over Time
ggplot(data, aes(x = nokken_poole_dim1, y = age, color = party)) +
  geom_point(size = 1) +
  labs(title = "Age and Ideological Drift by Party",
       x = "Ideology Change over Time",
       y = "Age")


#Party Loyalty
data <- S118_votes
votes <- subset(data, select = -c(congress, chamber, prob))
votes$cast_code = votes$cast_code == 1

mat <- xtabs( cast_code ~ rollnumber + icpsr , data = votes)
mat

decomp <- svd(mat)
decomp$u
decomp$d
decomp$v

k = 3

d <- decomp$d[1:2]
U2 <- decomp$u[, 1:2, drop = FALSE]
V2 <- decomp$v[, 1:2, drop = FALSE]

U_scaled <- U2 %*% diag(sqrt(d))
V_scaled <- V2 %*% diag(sqrt(d))

icpsr_coords <- data.frame(
  icpsr = as.character(colnames(mat)),
  X = V_scaled[,1],
  Y = V_scaled[,2]
)
colnames(mat)


S118_members$icpsr <- as.character(S118_members$icpsr)
head(S118_members)
head(icpsr_coords)
named_icpsr <- left_join(icpsr_coords, S118_members, by = 'icpsr')

named_icpsr$party_color <- ifelse(
  named_icpsr$party_code == 200, "red",
  ifelse(named_icpsr$party_code == 100, "blue", "purple")
)

plot(
  named_icpsr$X, named_icpsr$Y,
  pch = 19,
  col = named_icpsr$party_color,
  xlab = "Component 1",
  ylab = "Component 2",
  main = "Rank-2 Embedding of Legislators (icpsr)"
)
```

```
text(named_icpsr$X, named_icpsr$Y,
     labels = named_icpsr$bioname,
     pos = 3, cex = 0.7)


rows <- grep("CORR", named_icpsr$bioname)
named_icpsr[rows, ]



# testing model strength
d <- decomp$d
pve <- d^2 / sum(d^2)
tot_pve <- cumsum(pve)


# PC scree matrix
options(scipen = 999)
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))
plot(pve, type = "b", pch = 19,
     xlab = "Component number",
     ylab = "Proportion of Variance Explained",
     main = "Variance Explained by Custom Components")

plot(pve, type = "b", pch = 19,
     xlab = "Component number",
     ylab = "Proportion of Variance Explained ( log )",
     main = "Variance Explained by Custom Components",
     log = "y")

plot(1-tot_pve, type = "b", pch = 19,
     xlab = "Component number",
     ylab = "Cumulative Variance Unexplained",
     main = "Cumulative Var Unexplained by Components")

plot(1-tot_pve, type = "b", pch = 19,
     xlab = "Component number",
     ylab = "Cumulative Variance Unexplained ( log )",
     main = "Cumulative Var Unexplained by Components",
     log = "y")


# corrilation scores
#S118_rollcalls$icpsr <- as.character(S118_rollcalls$icpsr)

score_data <- named_icpsr[, c("X", "Y", "nominate_dim1", "nominate_dim2")]
cor_matrix <- cor(score_data, use = "complete.obs")
print(cor_matrix)

custom_dims <- c("X", "Y")
nom_dims    <- c("nominate_dim1", "nominate_dim2")

for (c_dim in custom_dims) {
```

```r
  for (n_dim in nom_dims) {

    plot(
      named_icpsr[[c_dim]],
      named_icpsr[[n_dim]],
      pch = 19,
      col = named_icpsr$party_color,
      xlab = paste("Custom", c_dim),
      ylab = paste("NOMINATE", n_dim),
      main = paste(c_dim, "vs", n_dim)
    )

    text(
      named_icpsr[[c_dim]],
      named_icpsr[[n_dim]],
      labels = named_icpsr$bioname,
      pos = 3,
      cex = 0.6
    )
  }
}


# ploting scores
plot(named_icpsr$X, named_icpsr$Y,
     pch = 19,
     col = named_icpsr$party_color,
     xlab = "X",
     ylab = "Y",
     main = "Custom Scores")

text(named_icpsr$X, named_icpsr$Y,
     labels = named_icpsr$bioname,
     pos = 3, cex = 0.6)

plot(named_icpsr$nominate_dim1, named_icpsr$nominate_dim2,
     pch = 19,
     col = named_icpsr$party_color,
     xlab = "NOMINATE Dimension 1",
     ylab = "NOMINATE Dimension 2",
     main = "NOMINATE Scores")

text(named_icpsr$nominate_dim1, named_icpsr$nominate_dim2,
     labels = named_icpsr$bioname,
     pos = 3, cex = 0.6)

# getting corrilation
cor(named_icpsr$X, named_icpsr$nominate_dim1, use = "complete.obs")
cor(named_icpsr$X, named_icpsr$nominate_dim2, use = "complete.obs")
cor(named_icpsr$Y, named_icpsr$nominate_dim2, use = "complete.obs")
cor(named_icpsr$Y, named_icpsr$nominate_dim1, use = "complete.obs")

# finding acc and conf
```

```r
S118_votes$icpsr <- as.character(S118_votes$icpsr)
named_icpsr$icpsr <- as.character(named_icpsr$icpsr)

results <- list()

for (roll in unique(S118_votes$rollnumber)) {

  votes_sub <- S118_votes |>
    filter(rollnumber == roll) |>
    left_join(named_icpsr, by = "icpsr") |>
    filter(!is.na(X), !is.na(nominate_dim1))

  if (nrow(votes_sub) == 0) next

  votes_sub$cast_bin <- ifelse(votes_sub$cast_code == 1, 1, 0)

  glm_custom <- glm(cast_bin ~ X + Y, data = votes_sub, family = binomial)
  prob_custom <- predict(glm_custom, type = "response")
  pred_custom <- ifelse(prob_custom > 0.5, 1, 0)
  acc_custom <- mean(pred_custom == votes_sub$cast_bin, na.rm = TRUE)
  conf_custom <- mean(abs(prob_custom - 0.5), na.rm = TRUE)

  glm_nom <- glm(cast_bin ~ nominate_dim1 + nominate_dim2, data = votes_sub, family = binomial)
  prob_nom <- predict(glm_nom, type = "response")
  pred_nom <- ifelse(prob_nom > 0.5, 1, 0)
  acc_nom <- mean(pred_nom == votes_sub$cast_bin, na.rm = TRUE)
  conf_nom <- mean(abs(prob_nom - 0.5), na.rm = TRUE)

  results[[as.character(roll)]] <- data.frame(
    rollnumber = roll,
    acc_custom, acc_nom,
    conf_custom, conf_nom
  )
}

results_df <- do.call(rbind, results)
head(results_df)
summary(results_df)

# acc and conf matrix
results_long <- results_df %>%
  select(acc_custom, acc_nom, conf_custom, conf_nom) %>%
  pivot_longer(
    cols = everything(),
    names_to = c("metric", "system"),
    names_sep = "_",
    values_to = "value"
  )

results_df <- results_df[, !(names(results_df) %in% "rollnumber")]
col_means <- sapply(results_df, mean, na.rm = TRUE)
col_vars <- sapply(results_df, var, na.rm = TRUE)
summary_stats <- data.frame(
  Mean = col_means,
  Variance = col_vars
)

summary_stats

#t-test
t.test(results_df$acc_custom, results_df$acc_nom, paired = TRUE)
t.test(results_df$conf_custom, results_df$conf_nom, paired = TRUE)

# plots for acc and conf
ggplot(results_long, aes(x = value, color = system)) +
  geom_density(size = 1) +
  facet_wrap(~metric, scales = "free") +
  scale_color_manual(values = c("blue", "red"), labels = c("Custom", "NOMINATE")) +
  labs(
    x = "Value",
    y = "Density",
    color = "Scoring System",
    title = "Density of Vote Prediction Accuracy and Confidence"
  ) +
  theme_minimal()
```