Candidate Political Positioning Analysis - Complete & Rigorous

nadia

2025-11-26

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE, fig.width = 8, fig.height = 6)

#Load all required libraries
library(readr)
library(dplyr)
```

```
## Warning: pakiet 'dplyr' został zbudowany w wersji R 4.4.3
```

```
##
## Dołączanie pakietu: 'dplyr'
```

```
## Następujące obiekty zostały zakryte z 'package:stats':
##
##     filter, lag
```

```
## Następujące obiekty zostały zakryte z 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: pakiet 'tidyr' został zbudowany w wersji R 4.4.3
```

```
library(ggplot2)
```

```
## Warning: pakiet 'ggplot2' został zbudowany w wersji R 4.4.3
```

```
library(factoextra)
```

```
## Warning: pakiet 'factoextra' został zbudowany w wersji R 4.4.3
```

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

```
library(psych)
```

```
## Warning: pakiet 'psych' został zbudowany w wersji R 4.4.3
```

```
##
## Dołączanie pakietu: 'psych'
```

```
## Następujące obiekty zostały zakryte z 'package:ggplot2':
##
##     %+%, alpha
```

```
library(caret)
```

```
## Warning: pakiet 'caret' został zbudowany w wersji R 4.4.3
```

```
## Ładowanie wymaganego pakietu: lattice
```

```
library(randomForest)
```

```
## Warning: pakiet 'randomForest' został zbudowany w wersji R 4.4.3
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Dołączanie pakietu: 'randomForest'
```

```
## Następujący obiekt został zakryty z 'package:psych':
##
##     outlier
```

```
## Następujący obiekt został zakryty z 'package:ggplot2':
##
##     margin
```

```
## Następujący obiekt został zakryty z 'package:dplyr':
##
##     combine
```

```
library(nnet)
library(MASS)
```

```
## Warning: pakiet 'MASS' został zbudowany w wersji R 4.4.3
```

```
##
## Dołączanie pakietu: 'MASS'
```

```
## Następujący obiekt został zakryty z 'package:dplyr':
##
##     select
```

```
library(fastshap)
```

```
## Warning: pakiet 'fastshap' został zbudowany w wersji R 4.4.3
```

```
##
## Dołączanie pakietu: 'fastshap'
```

```
## Następujący obiekt został zakryty z 'package:dplyr':
##
##     explain
```

```
library(tibble)
library(igraph)
```

```
## Warning: pakiet 'igraph' został zbudowany w wersji R 4.4.3
```

```
##
## Dołączanie pakietu: 'igraph'
```

```
## Następujący obiekt został zakryty z 'package:tibble':
##
##     as_data_frame
```

```
## Następujący obiekt został zakryty z 'package:tidyr':
##
##     crossing
```

```
## Następujące obiekty zostały zakryte z 'package:dplyr':
##
##     as_data_frame, groups, union
```

```
## Następujące obiekty zostały zakryte z 'package:stats':
##
##     decompose, spectrum
```

```
## Następujący obiekt został zakryty z 'package:base':
##
##     union
```

```
library(tidygraph)
```

```
## Warning: pakiet 'tidygraph' został zbudowany w wersji R 4.4.3
```

```
##
## Dołączanie pakietu: 'tidygraph'
```

```
## Następujący obiekt został zakryty z 'package:igraph':
##
##     groups
```

```
## Następujący obiekt został zakryty z 'package:MASS':
##
##     select
```

```
## Następujący obiekt został zakryty z 'package:stats':
##
##     filter
```

```
library(ggraph)
```

```
## Warning: pakiet 'ggraph' został zbudowany w wersji R 4.4.3
```

```
library(visNetwork)
```

```
## Warning: pakiet 'visNetwork' został zbudowany w wersji R 4.4.3
```

```
library(viridis)
```

```
## Warning: pakiet 'viridis' został zbudowany w wersji R 4.4.3
```

```
## Ładowanie wymaganego pakietu: viridisLite
```

```
library(pheatmap)
```

```
## Warning: pakiet 'pheatmap' został zbudowany w wersji R 4.4.3
```

```
library(ggcorrplot)
```

```
## Warning: pakiet 'ggcorrplot' został zbudowany w wersji R 4.4.3
```

```
library(vip)
```

```
## Warning: pakiet 'vip' został zbudowany w wersji R 4.4.3
```

```
##
## Dołączanie pakietu: 'vip'
```

```
## Następujący obiekt został zakryty z 'package:fastshap':
##
##     gen_friedman
```

```
## Następujący obiekt został zakryty z 'package:utils':
##
##     vi
```

```
library(fpc)
```

```
## Warning: pakiet 'fpc' został zbudowany w wersji R 4.4.3
```

```
library(stringr)
library(cluster)



#Set seed for reproducibility
set.seed(123)

#Helper functions
make_safe_levels <- function(f) {
safe_levels <- make.names(levels(f))
factor(make.names(as.character(f)), levels = safe_levels)
}

#Min-max normalization to 0-1 scale
normalize_01 <- function(x) {
(x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
}



# Data Loading and Initial Cleaning

#Load data
df <- read_csv("C:/Users/nadia/Desktop/pilot/altinget_answers_playwright_271.csv",
show_col_types = FALSE)

#Normalize candidate name column
if (!"candidate_name" %in% names(df) & "name" %in% names(df)) {
df <- rename(df, candidate_name = name)
}

#Detect question columns (Q1-Q26)
question_cols <- grep("^Q", names(df), value = TRUE)
cat("Detected", length(question_cols), "question columns\n")
```

```
## Detected 26 question columns
```

```
#Initial filtering: complete cases only
before_n <- nrow(df)
df <- df %>%
filter(!is.na(party) & party != "") %>%
filter(if_all(all_of(question_cols), ~ !is.na(.))) %>%
mutate(across(all_of(question_cols), as.numeric))

cat("Rows before filtering:", before_n, "\n")
```

```
## Rows before filtering: 199
```

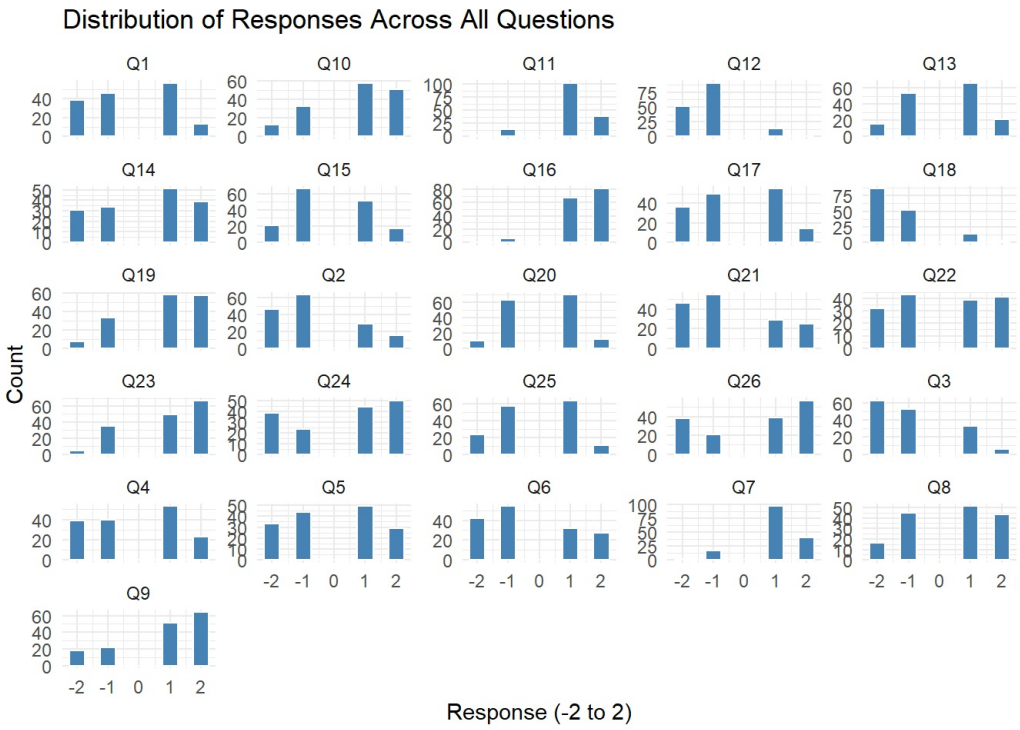
```
cat("Rows after filtering:", nrow(df), "\n")
```

```
## Rows after filtering: 155
```

```
# Exploratory Data Analysis (Pre-Filtering)

## Response Distributions

#Summary statistics for all questions
summary(select(df, all_of(question_cols)))
```

```
##        Q1               Q2               Q3               Q4
##  Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:-1.5000   1st Qu.:-2.0000   1st Qu.:-2.0000   1st Qu.:-1.5000
##  Median :-1.0000   Median :-1.0000   Median :-1.0000   Median :-1.0000
##  Mean   :-0.2645   Mean   :-0.6387   Mean   :-0.8645   Mean   :-0.1226
##  3rd Qu.: 1.0000   3rd Qu.: 1.0000   3rd Qu.: 0.0000   3rd Qu.: 1.0000
##  Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.0000
##        Q5                Q6               Q7           Q8
##  Min.   :-2.00000   Min.   :-2.0000   Min.   :-2   Min.   :-2.0000
##  1st Qu.:-1.00000   1st Qu.:-2.0000   1st Qu.: 1   1st Qu.:-1.0000
##  Median : 1.00000   Median :-1.0000   Median : 1   Median : 1.0000
##  Mean   :-0.01935   Mean   :-0.3355   Mean   : 1   Mean   : 0.3871
##  3rd Qu.: 1.00000   3rd Qu.: 1.0000   3rd Qu.: 2   3rd Qu.: 2.0000
##  Max.   : 2.00000   Max.   : 2.0000   Max.   : 2   Max.   : 2.0000
##        Q9               Q10              Q11              Q12
##  Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.000   Min.   :-2.000
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.: 1.000   1st Qu.:-2.000
##  Median : 1.0000   Median : 1.0000   Median : 1.000   Median :-1.000
##  Mean   : 0.7806   Mean   : 0.6516   Mean   : 1.039   Mean   :-1.161
##  3rd Qu.: 2.0000   3rd Qu.: 2.0000   3rd Qu.: 1.000   3rd Qu.:-1.000
##  Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.000   Max.   : 1.000
##       Q13              Q14              Q15              Q16
##  Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-1.000
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.: 1.000
##  Median : 1.0000   Median : 1.0000   Median :-1.0000   Median : 2.000
##  Mean   : 0.1613   Mean   : 0.2129   Mean   :-0.1484   Mean   : 1.445
##  3rd Qu.: 1.0000   3rd Qu.: 1.5000   3rd Qu.: 1.0000   3rd Qu.: 2.000
##  Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.000
##       Q17              Q18              Q19              Q20
##  Min.   :-2.0000   Min.   :-2.000   Min.   :-2.0000   Min.   :-2.00000
##  1st Qu.:-1.0000   1st Qu.:-2.000   1st Qu.:-1.0000   1st Qu.:-1.00000
##  Median :-1.0000   Median :-2.000   Median : 1.0000   Median : 1.00000
##  Mean   :-0.2516   Mean   :-1.316   Mean   : 0.8065   Mean   : 0.07097
##  3rd Qu.: 1.0000   3rd Qu.:-1.000   3rd Qu.: 2.0000   3rd Qu.: 1.00000
##  Max.   : 2.0000   Max.   : 2.000   Max.   : 2.0000   Max.   : 2.00000
##       Q21              Q22              Q23              Q24
##  Min.   :-2.0000   Min.   :-2.00000   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:-2.0000   1st Qu.:-1.00000   1st Qu.: 0.0000   1st Qu.:-1.0000
##  Median :-1.0000   Median : 1.00000   Median : 1.0000   Median : 1.0000
##  Mean   :-0.4387   Mean   : 0.09032   Mean   : 0.9032   Mean   : 0.2903
##  3rd Qu.: 1.0000   3rd Qu.: 2.00000   3rd Qu.: 2.0000   3rd Qu.: 2.0000
##  Max.   : 2.0000   Max.   : 2.00000   Max.   : 2.0000   Max.   : 2.0000
##       Q25              Q26
##  Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median :-1.0000   Median : 1.0000
##  Mean   :-0.1097   Mean   : 0.3613
##  3rd Qu.: 1.0000   3rd Qu.: 2.0000
##  Max.   : 2.0000   Max.   : 2.0000
```

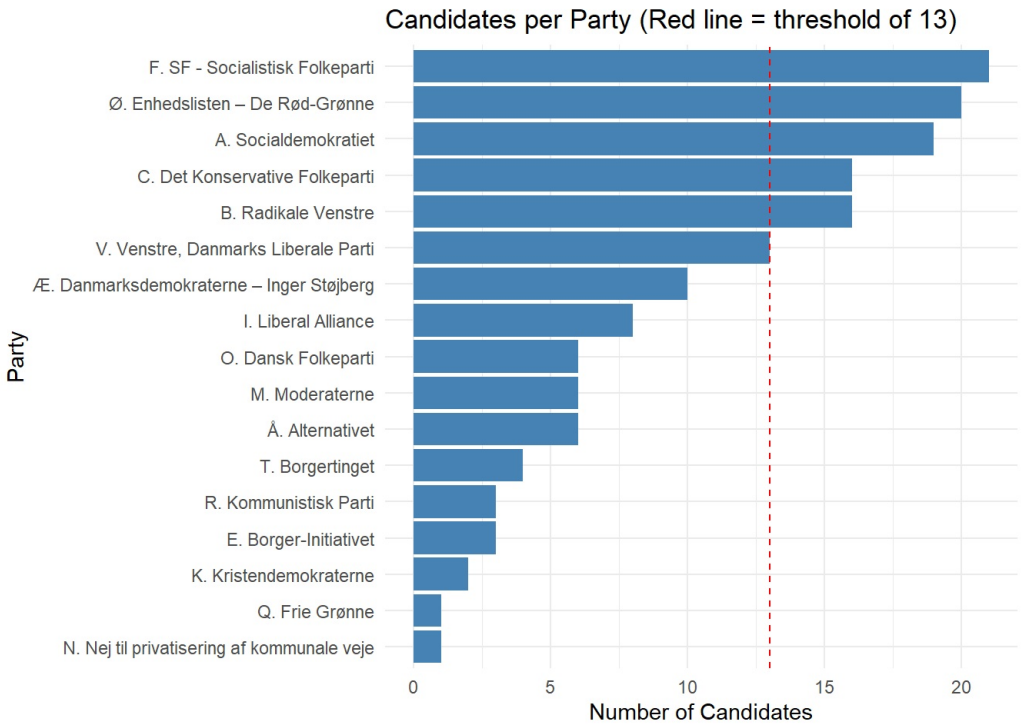
```
#Visualize response distributions
df_long <- df %>%
pivot_longer(all_of(question_cols), names_to = "Question", values_to = "Value")

ggplot(df_long, aes(x = Value)) +
geom_histogram(binwidth = 0.5, color = "white", fill = "steelblue") +
facet_wrap(~Question, scales = "free_y", ncol = 5) +
theme_minimal() +
labs(title = "Distribution of Responses Across All Questions",
x = "Response (-2 to 2)", y = "Count")
```



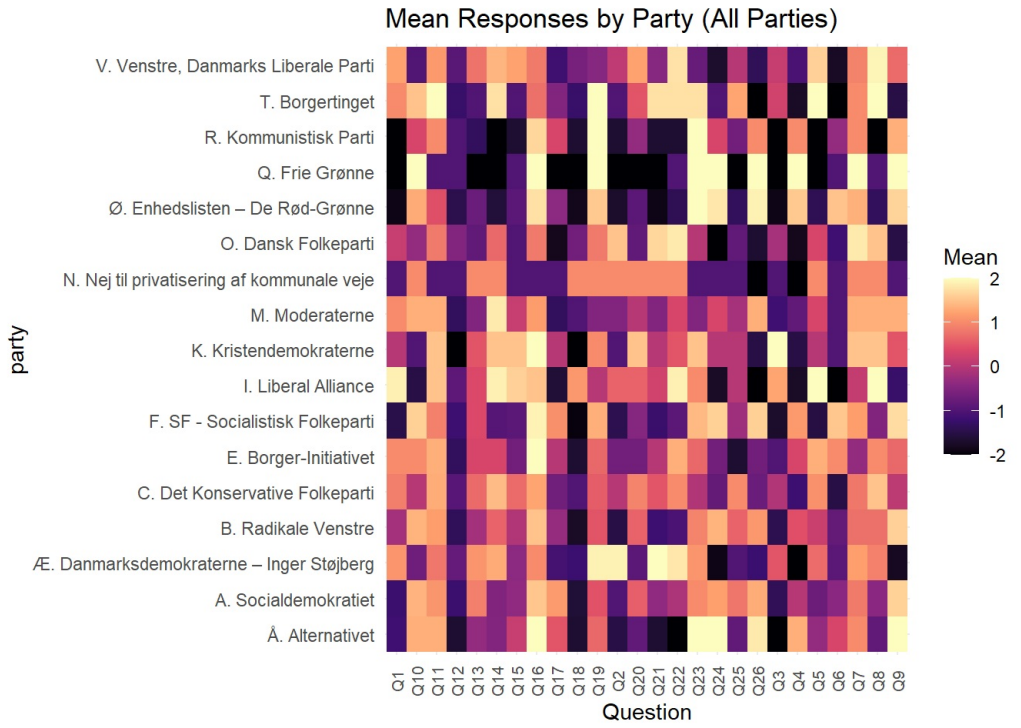

```
## Party Distribution (Before Filtering)

#Count candidates per party
party_counts <- df %>%
count(party, name = "n") %>%
arrange(desc(n))

print(party_counts)
```

```
## # A tibble: 17 × 2
##    party                                        n
##    <chr>                                    <int>
##  1 F. SF - Socialistisk Folkeparti             21
##  2 Ø. Enhedslisten – De Rød-Grønne             20
##  3 A. Socialdemokratiet                        19
##  4 B. Radikale Venstre                         16
##  5 C. Det Konservative Folkeparti              16
##  6 V. Venstre, Danmarks Liberale Parti         13
##  7 Æ. Danmarksdemokraterne – Inger Støjberg    10
##  8 I. Liberal Alliance                          8
##  9 M. Moderaterne                               6
## 10 O. Dansk Folkeparti                          6
## 11 Å. Alternativet                              6
## 12 T. Borgertinget                              4
## 13 E. Borger-Initiativet                        3
## 14 R. Kommunistisk Parti                        3
## 15 K. Kristendemokraterne                       2
## 16 N. Nej til privatisering af kommunale veje   1
## 17 Q. Frie Grønne                               1
```

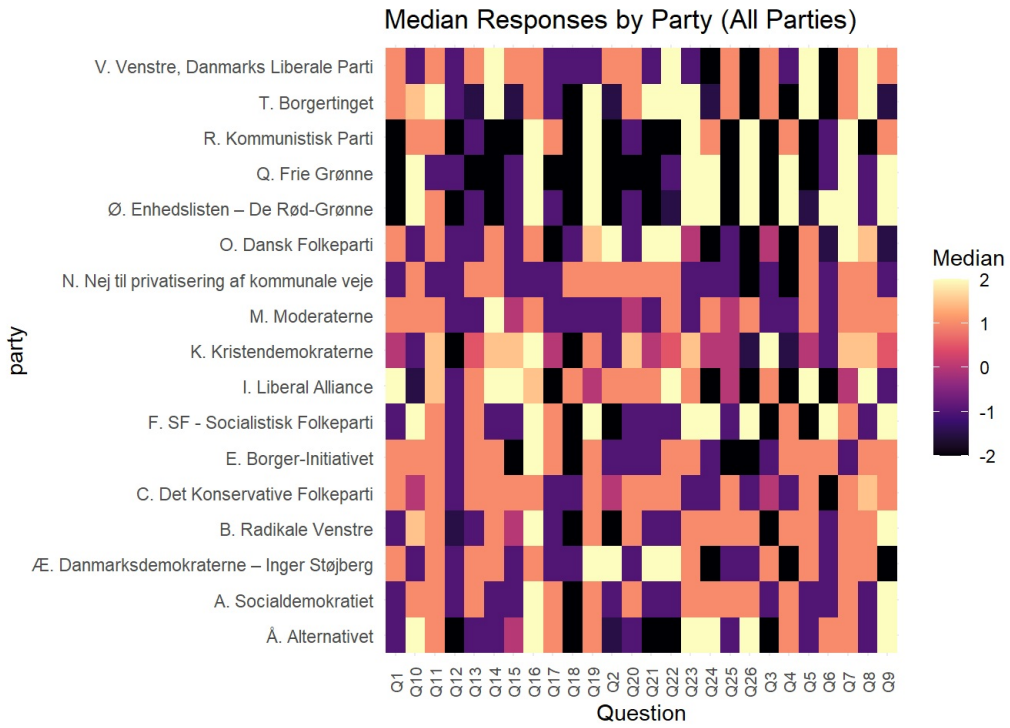
```
ggplot(party_counts, aes(x = reorder(party, n), y = n)) +
geom_col(fill = "steelblue") +
geom_hline(yintercept = 13, linetype = "dashed", color = "red") +
coord_flip() +
theme_minimal() +
labs(title = "Candidates per Party (Red line = threshold of 13)",
x = "Party", y = "Number of Candidates")
```




```
## Mean and Median Responses by Party (Pre-Filtering)

#Calculate mean and median for each party-question combination
party_summary <- df %>%
group_by(party) %>%
summarise(across(all_of(question_cols),
list(mean = ~mean(.x, na.rm=TRUE),
median = ~median(.x, na.rm=TRUE))),
.groups = "drop")

#Create mean matrix for heatmap
mean_long <- party_summary %>%
select(party, ends_with("_mean")) %>%
pivot_longer(-party, names_to = "Question", values_to = "Mean") %>%
mutate(Question = str_remove(Question, "_mean"))

ggplot(mean_long, aes(x = Question, y = party, fill = Mean)) +
geom_tile() +
scale_fill_viridis_c(option = "magma") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 8)) +
labs(title = "Mean Responses by Party (All Parties)")
```



```
#Create median matrix for heatmap
median_long <- party_summary %>%
select(party, ends_with("_median")) %>%
pivot_longer(-party, names_to = "Question", values_to = "Median") %>%
mutate(Question = str_remove(Question, "_median"))

ggplot(median_long, aes(x = Question, y = party, fill = Median)) +
geom_tile() +
scale_fill_viridis_c(option = "magma") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 8)) +
labs(title = "Median Responses by Party (All Parties)")
```



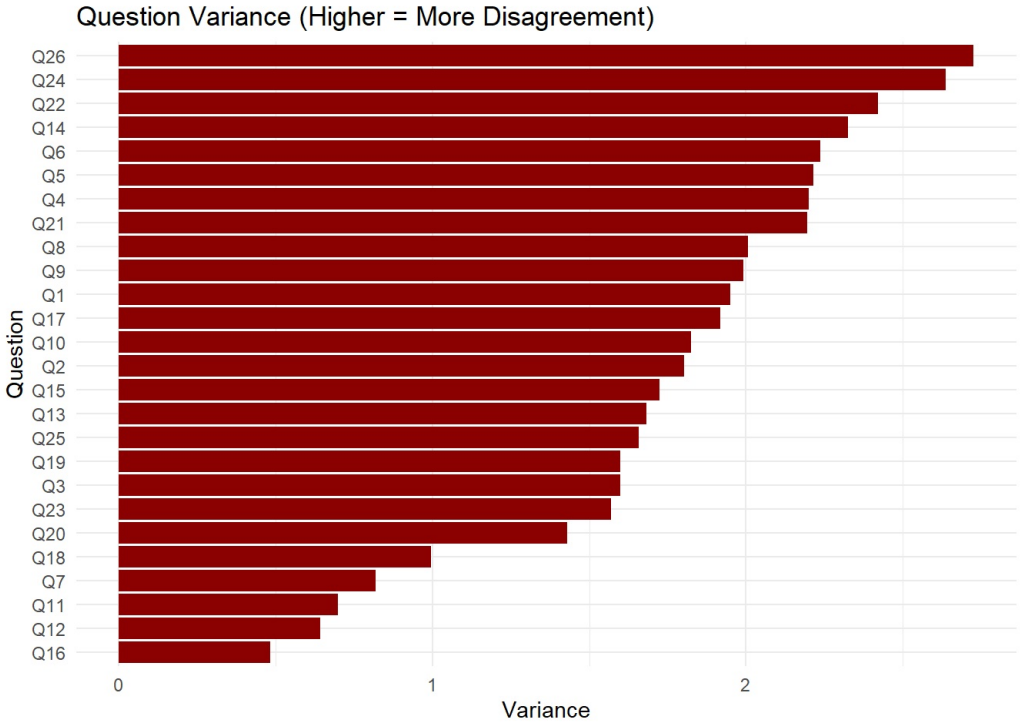

```
## Question Variance Analysis

#Compute variance and SD per question
variance_df <- df %>%
select(all_of(question_cols)) %>%
summarise(across(everything(), var, na.rm = TRUE)) %>%
pivot_longer(cols = everything(), names_to = "Question", values_to = "Variance")
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), var, na.rm = TRUE)`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
sd_df <- df %>%
select(all_of(question_cols)) %>%
summarise(across(everything(), sd, na.rm = TRUE)) %>%
pivot_longer(cols = everything(), names_to = "Question", values_to = "SD")

variance_summary <- variance_df %>%
left_join(sd_df, by = "Question") %>%
arrange(desc(Variance))

print(variance_summary)
```

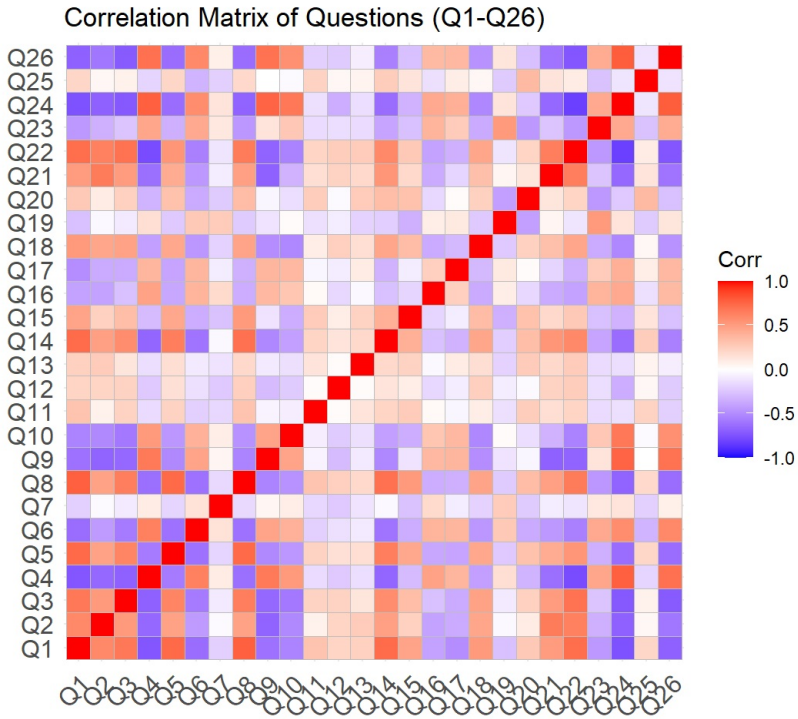
```
## # A tibble: 26 × 3
##    Question Variance    SD
##    <chr>       <dbl> <dbl>
##  1 Q26          2.73  1.65
##  2 Q24          2.64  1.62
##  3 Q22          2.42  1.56
##  4 Q14          2.32  1.52
##  5 Q6           2.24  1.50
##  6 Q5           2.21  1.49
##  7 Q4           2.20  1.48
##  8 Q21          2.20  1.48
##  9 Q8           2.01  1.42
## 10 Q9           1.99  1.41
## # ℹ 16 more rows
```

```
ggplot(variance_summary, aes(x = reorder(Question, Variance), y = Variance)) +
geom_col(fill = "darkred") +
coord_flip() +
theme_minimal() +
labs(title = "Question Variance (Higher = More Disagreement)",
x = "Question", y = "Variance")
```




```
# Internal Consistency and Item Analysis

#Correlation matrix of questions
corr_matrix <- cor(select(df, all_of(question_cols)), use = "pairwise.complete.obs")

ggcorrplot(corr_matrix, lab = FALSE,
title = "Correlation Matrix of Questions (Q1-Q26)")
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## ℹ The deprecated feature was likely used in the ggcorrplot package.
##   Please report the issue at <https://github.com/kassambara/ggcorrplot/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
#Cronbach's alpha with automatic key checking
alpha_initial <- psych::alpha(select(df, all_of(question_cols)),
check.keys = TRUE)
```

```
## Warning in psych::alpha(select(df, all_of(question_cols)), check.keys = TRUE): Some items were negatively corr
elated with the first principal component and were automatically reversed.
##  This is indicated by a negative sign for the variable name.
```

```
cat("\n=== CRONBACH'S ALPHA (BEFORE REVERSAL) ===\n")
```

```
##
## === CRONBACH'S ALPHA (BEFORE REVERSAL) ===
```

```
print(alpha_initial$total)
```

```
##  raw_alpha std.alpha   G6(smc) average_r     S/N         ase       mean
##  0.9357037 0.9303083 0.9550076 0.3392446 13.3489 0.006625487 -0.3280397
##         sd  median_r
##  0.8178246 0.3182825
```

```
#Extract item statistics
item_stats <- alpha_initial$item.stats %>%
rownames_to_column("Question") %>%
as_tibble() %>%
arrange(r.drop)

print(item_stats %>% select(Question, r.drop, raw.r))
```

```
## # A tibble: 26 × 3
##    Question r.drop raw.r
##    <chr>     <dbl> <dbl>
##  1 Q7-       0.200 0.241
##  2 Q25       0.222 0.279
##  3 Q13       0.231 0.288
##  4 Q11       0.264 0.301
##  5 Q19-      0.265 0.320
##  6 Q12       0.273 0.308
##  7 Q20       0.384 0.431
##  8 Q17-      0.398 0.452
##  9 Q15       0.461 0.509
## 10 Q16-      0.470 0.495
## # ℹ 16 more rows
```

```
#Programmatic reversal of negatively correlated items
items_to_reverse <- item_stats %>%
filter(r.drop < 0) %>%
pull(Question)

if (length(items_to_reverse) > 0) {
cat("\nReversing items with negative item-total correlation:",
paste(items_to_reverse, collapse = ", "), "\n")

df <- df %>%
mutate(across(all_of(items_to_reverse), ~ -1 * .x))

#Recalculate alpha after reversal
alpha_after <- psych::alpha(df %>% select(all_of(question_cols)),
check.keys = TRUE)
cat("\n=== CRONBACH'S ALPHA (AFTER REVERSAL) ===\n")
print(alpha_after$total)
} else {
cat("\nNo items required reversal.\n")
}
```

```
##
## No items required reversal.
```

```
# Party Filtering (Threshold = 13)

#Filter parties with at least 13 candidates
threshold <- 13
valid_parties <- party_counts %>%
filter(n >= threshold) %>%
pull(party)

cat("Keeping parties with >=", threshold, "candidates:",
paste(valid_parties, collapse = ", "), "\n")
```

```
## Keeping parties with >= 13 candidates: F. SF - Socialistisk Folkeparti, Ø. Enhedslisten – De Rød-Grønne, A. So
cialdemokratiet, B. Radikale Venstre, C. Det Konservative Folkeparti, V. Venstre, Danmarks Liberale Parti
```

```
df <- df %>%
filter(party %in% valid_parties) %>%
mutate(party = droplevels(factor(party)))

cat("Remaining observations:", nrow(df), "\n")
```

```
## Remaining observations: 105
```

```
cat("Remaining parties:", length(levels(df$party)), "\n")
```

```
## Remaining parties: 6
```

```
# Principal Component Analysis (PCA)

## PCA Computation and Variance Explained

#Prepare scaled matrix for PCA
question_matrix <- df %>%
select(all_of(question_cols)) %>%
as.matrix() %>%
scale()

#Perform PCA
pca <- prcomp(question_matrix, center = TRUE, scale. = TRUE)

#Extract eigenvalues
eig <- factoextra::get_eigenvalue(pca)
cat("\n=== VARIANCE EXPLAINED BY PRINCIPAL COMPONENTS ===\n")
```

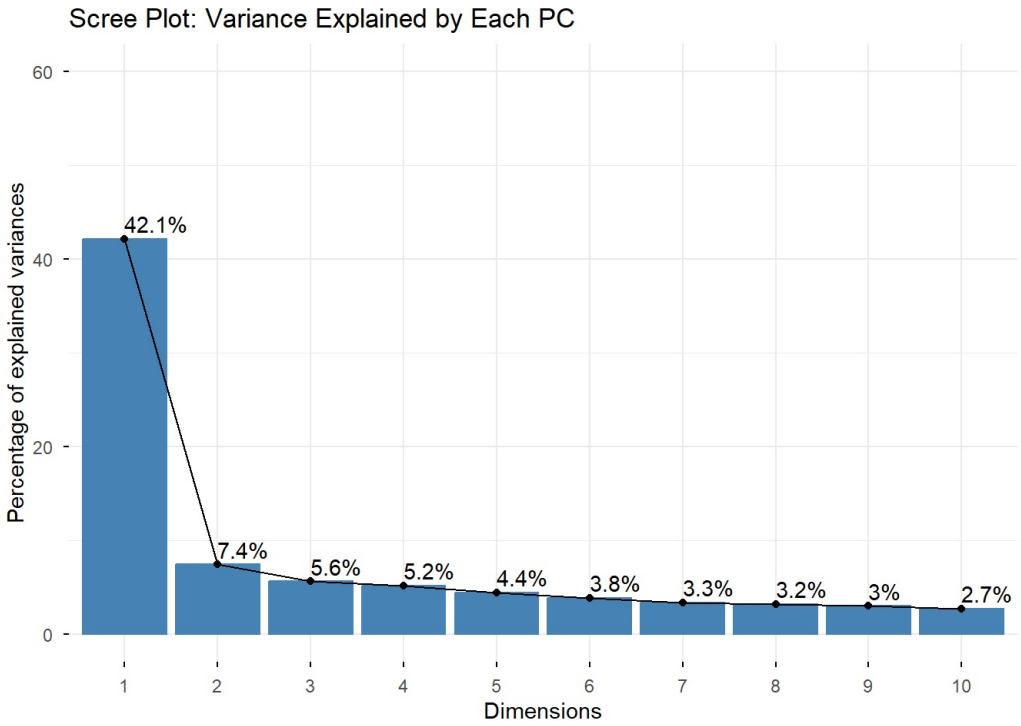
```
##
## === VARIANCE EXPLAINED BY PRINCIPAL COMPONENTS ===
```

```
print(eig[1:6, ])
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 10.9552347        42.135518                    42.13552
## Dim.2  1.9226088         7.394649                    49.53017
## Dim.3  1.4547420         5.595162                    55.12533
## Dim.4  1.3432492         5.166343                    60.29167
## Dim.5  1.1429046         4.395787                    64.68746
## Dim.6  0.9987978         3.841530                    68.52899
```

```
#Store PC1 and PC2 variance percentages
pc1_pct <- eig[1, "variance.percent"]
pc2_pct <- eig[2, "variance.percent"]

#Scree plot
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 60)) +
labs(title = "Scree Plot: Variance Explained by Each PC")
```

```
## Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
## Ignoring empty aesthetic: `width`.
```




```
## PCA Loadings Analysis

#Extract loadings
loadings <- as.data.frame(pca$rotation) %>%
rownames_to_column("Question")

#Top questions for PC1 (ideological dimension)
top_PC1 <- loadings %>%
arrange(desc(abs(PC1))) %>%
slice_head(n = 15)

#Top questions for PC2
top_PC2 <- loadings %>%
arrange(desc(abs(PC2))) %>%
slice_head(n = 15)

cat("\n=== TOP 15 QUESTIONS DEFINING PC1 (Main Ideological Dimension) ===\n")
```

```
##
## === TOP 15 QUESTIONS DEFINING PC1 (Main Ideological Dimension) ===
```

```
print(top_PC1%>% select(Question, PC1))
```

```
##    Question        PC1
## 1       Q24 -0.2714017
## 2        Q1  0.2663243
## 3        Q8  0.2453282
## 4       Q26 -0.2450293
## 5        Q4 -0.2412517
## 6        Q5  0.2370080
## 7        Q6 -0.2351612
## 8       Q22  0.2348180
## 9       Q14  0.2315917
## 10       Q3  0.2307329
## 11      Q23 -0.2214503
## 12       Q2  0.2071678
## 13      Q10 -0.2023381
## 14      Q15  0.1882552
## 15      Q20  0.1854942
```

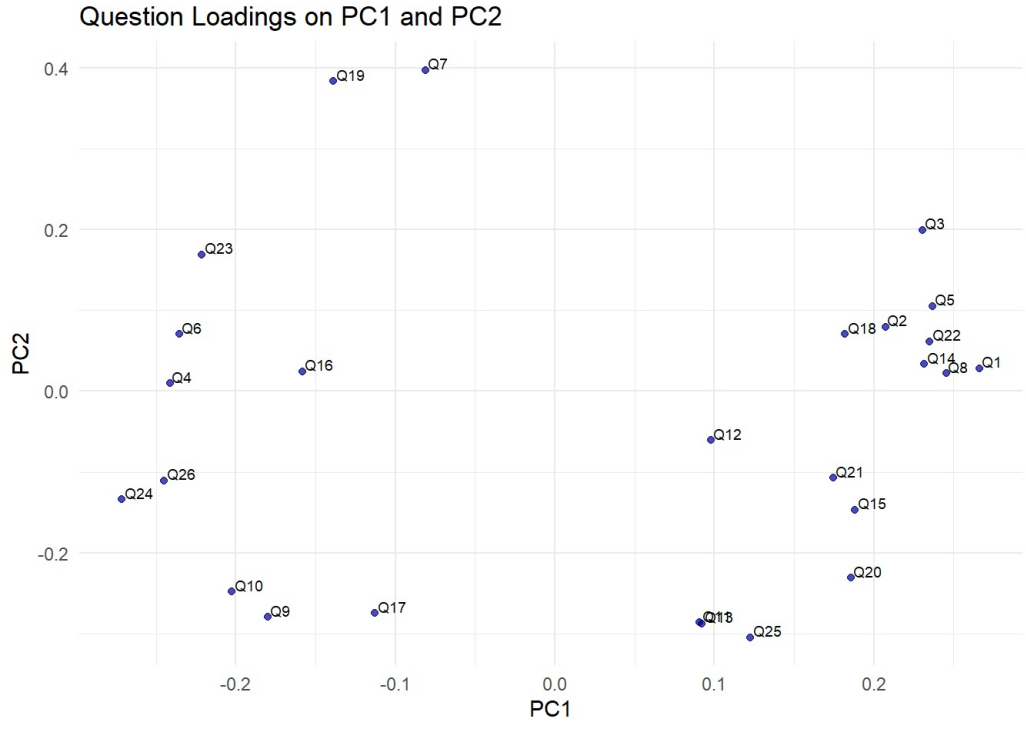
```
cat("\n=== TOP 15 QUESTIONS DEFINING PC2 ===\n")
```

```
##
## === TOP 15 QUESTIONS DEFINING PC2 ===
```

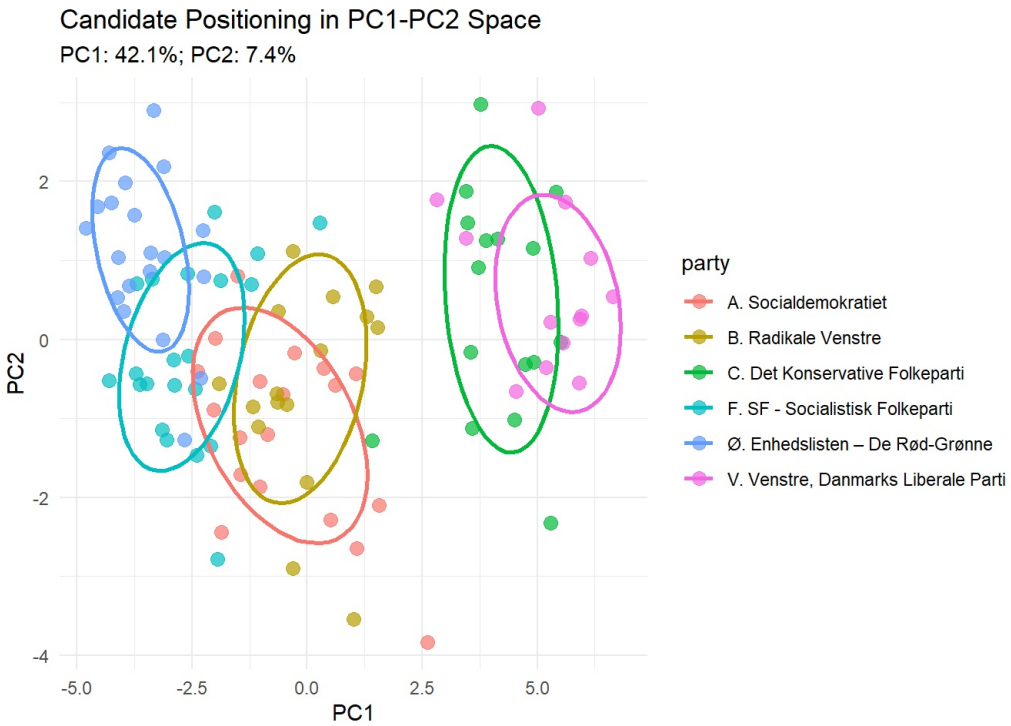
```
print(top_PC2%>% select(Question, PC2))
```

```
##    Question        PC2
## 1        Q7  0.3973742
## 2       Q19  0.3833189
## 3       Q25 -0.3044923
## 4       Q13 -0.2874796
## 5       Q11 -0.2862715
## 6        Q9 -0.2792666
## 7       Q17 -0.2746936
## 8       Q10 -0.2481612
## 9       Q20 -0.2310865
## 10       Q3  0.1994836
## 11      Q23  0.1690279
## 12      Q15 -0.1466609
## 13      Q24 -0.1337538
## 14      Q26 -0.1105016
## 15      Q21 -0.1072435
```

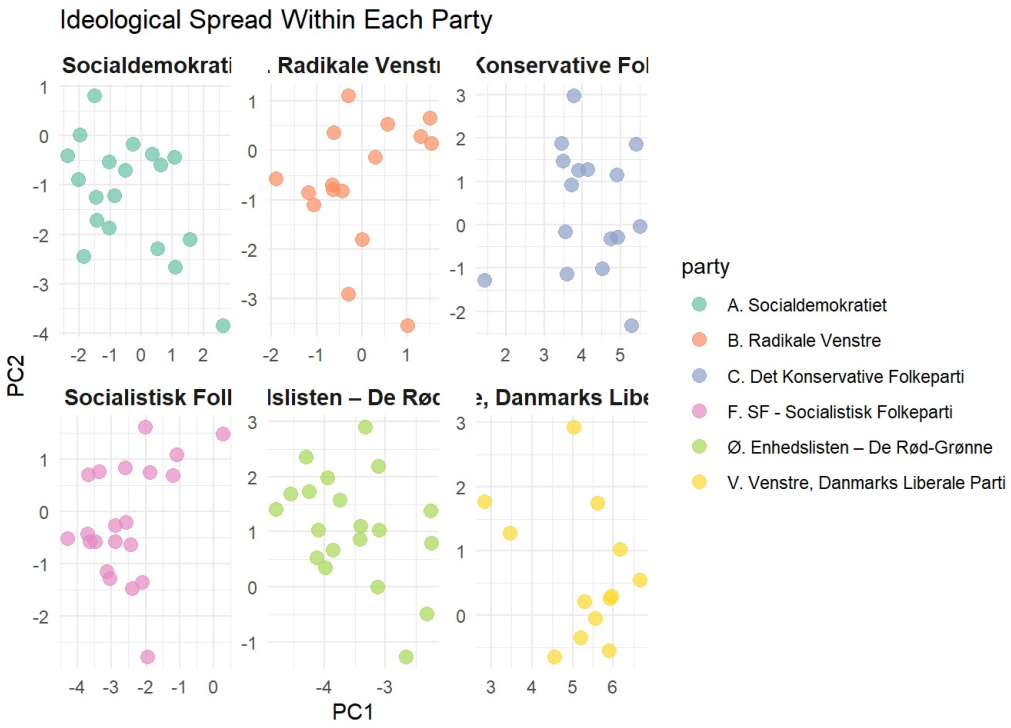
```
#Visualize loadings
ggplot(loadings, aes(x = PC1, y = PC2, label = Question)) +
geom_point(alpha = 0.7, color = "darkblue") +
geom_text(size = 2.5, hjust = -0.1, vjust = -0.1) +
theme_minimal() +
labs(title = "Question Loadings on PC1 and PC2")
```



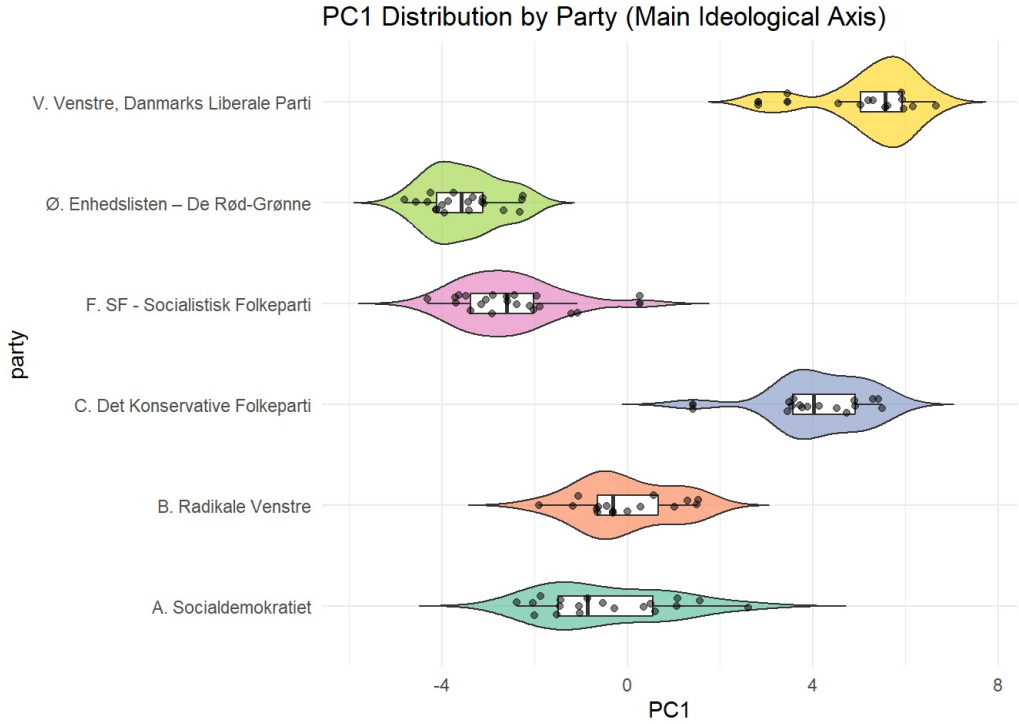
```
## Candidate Positioning in PC Space

#Extract candidate scores
candidate_scores <- as.data.frame(pca$x) %>%
mutate(candidate_name = df$candidate_name,
party = df$party)

#Calculate party centroids
party_centroids <- candidate_scores %>%
group_by(party) %>%
summarise(across(starts_with("PC"), mean),
n = n(),
.groups = "drop")

#Calculate distance from party centroid for each candidate
candidate_scores <- candidate_scores %>%
left_join(party_centroids %>%
select(party, PC1_cent = PC1, PC2_cent = PC2),
by = "party") %>%
mutate(distance_from_party = sqrt((PC1 - PC1_cent)^2 + (PC2 - PC2_cent)^2))

#Plot: PC1 vs PC2 with party colors and ellipses
groups_with_enough <- candidate_scores %>%
count(party) %>%
filter(n >= 3) %>%
pull(party)

ggplot(candidate_scores, aes(x = PC1, y = PC2, color = party)) +
geom_point(size = 3, alpha = 0.7) +
stat_ellipse(data = candidate_scores %>% filter(party %in% groups_with_enough),
aes(group = party), level = 0.68, linewidth = 0.9) +
theme_minimal() +
labs(title = "Candidate Positioning in PC1-PC2 Space",
subtitle = paste0("PC1: ", round(pc1_pct, 1), "%; PC2: ", round(pc2_pct, 1), "%"))
```



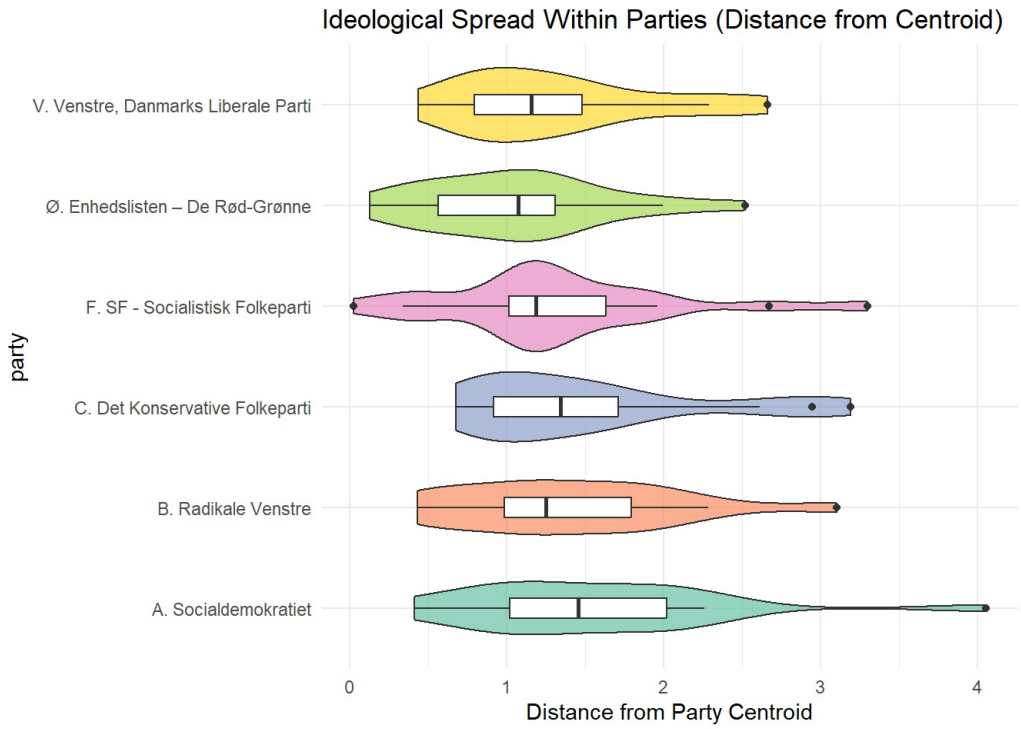

```
#Faceted view by party
ggplot(candidate_scores, aes(x = PC1, y = PC2)) +
geom_point(aes(color = party), size = 3, alpha = 0.7) +
facet_wrap(~party, scales = "free") +
theme_minimal() +
scale_color_brewer(palette = "Set2") +
labs(title = "Ideological Spread Within Each Party") +
theme(strip.text = element_text(face = "bold", size = 11))
```




```
#Violin plot: PC1 distribution by party
ggplot(candidate_scores, aes(x = party, y = PC1, fill = party)) +
geom_violin(alpha = 0.7, trim = FALSE) +
geom_boxplot(width = 0.2, fill = "white") +
geom_jitter(width = 0.1, size = 1.5, alpha = 0.5) +
coord_flip() +
theme_minimal() +
scale_fill_brewer(palette = "Set2") +
labs(title = "PC1 Distribution by Party (Main Ideological Axis)") +
theme(legend.position = "none")
```




```
#Ideological spread within parties
ggplot(candidate_scores, aes(x = party, y = distance_from_party, fill = party)) +
geom_violin(alpha = 0.7) +
geom_boxplot(width = 0.2, fill = "white") +
coord_flip() +
theme_minimal() +
scale_fill_brewer(palette = "Set2") +
labs(title = "Ideological Spread Within Parties (Distance from Centroid)",
y = "Distance from Party Centroid") +
theme(legend.position = "none")
```




```
## Statistical Testing: Party Differences on PC1

#ANOVA: Do parties differ on PC1?
anova_pc1 <- aov(PC1 ~ party, data = candidate_scores)
cat("\n=== ANOVA: PC1 ~ Party ===\n")
```

```
##
## === ANOVA: PC1 ~ Party ===
```

```
print(summary(anova_pc1))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## party        5 1025.0  204.99   177.4 <2e-16 ***
## Residuals   99  114.4    1.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Post-hoc comparisons
cat("\n=== Tukey HSD Post-Hoc Comparisons ===\n")
```

```
##
## === Tukey HSD Post-Hoc Comparisons ===
```

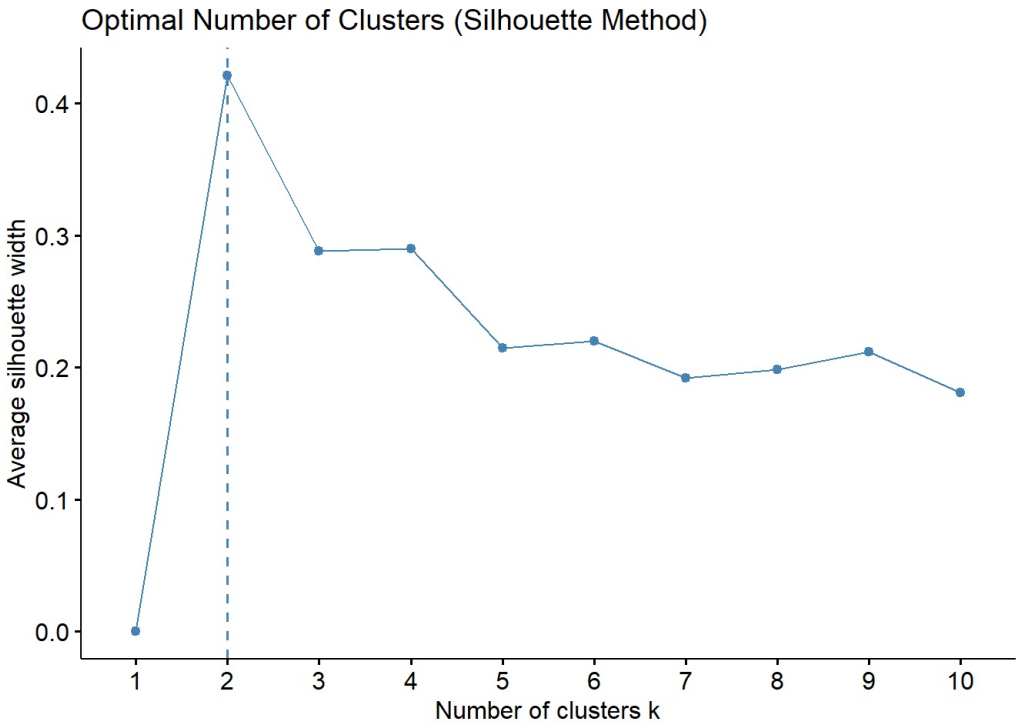
```
print(TukeyHSD(anova_pc1))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = PC1 ~ party, data = candidate_scores)
##
## $party
##                                                                    diff
## B. Radikale Venstre-A. Socialdemokratiet                          0.3960915
## C. Det Konservative Folkeparti-A. Socialdemokratiet               4.5977147
## F. SF - Socialistisk Folkeparti-A. Socialdemokratiet             -2.1255793
## Ø. Enhedslisten – De Rød-Grønne-A. Socialdemokratiet             -3.0801135
## V. Venstre, Danmarks Liberale Parti-A. Socialdemokratiet          5.6923629
## C. Det Konservative Folkeparti-B. Radikale Venstre                4.2016231
## F. SF - Socialistisk Folkeparti-B. Radikale Venstre              -2.5216708
## Ø. Enhedslisten – De Rød-Grønne-B. Radikale Venstre              -3.4762050
## V. Venstre, Danmarks Liberale Parti-B. Radikale Venstre           5.2962714
## F. SF - Socialistisk Folkeparti-C. Det Konservative Folkeparti   -6.7232939
## Ø. Enhedslisten – De Rød-Grønne-C. Det Konservative Folkeparti   -7.6778281
## V. Venstre, Danmarks Liberale Parti-C. Det Konservative Folkeparti  1.0946482
## Ø. Enhedslisten – De Rød-Grønne-F. SF - Socialistisk Folkeparti  -0.9545342
## V. Venstre, Danmarks Liberale Parti-F. SF - Socialistisk Folkeparti  7.8179422
## V. Venstre, Danmarks Liberale Parti-Ø. Enhedslisten – De Rød-Grønne  8.7724764
##                                                                       lwr
## B. Radikale Venstre-A. Socialdemokratiet                         -0.66390430
## C. Det Konservative Folkeparti-A. Socialdemokratiet               3.53771883
## F. SF - Socialistisk Folkeparti-A. Socialdemokratiet             -3.11470267
## Ø. Enhedslisten – De Rød-Grønne-A. Socialdemokratiet             -4.08091378
## V. Venstre, Danmarks Liberale Parti-A. Socialdemokratiet          4.56793025
## C. Det Konservative Folkeparti-B. Radikale Venstre                3.09713277
## F. SF - Socialistisk Folkeparti-B. Radikale Venstre              -3.55833511
## Ø. Enhedslisten – De Rød-Grønne-B. Radikale Venstre              -4.52401656
## V. Venstre, Danmarks Liberale Parti-B. Radikale Venstre           4.12979953
## F. SF - Socialistisk Folkeparti-C. Det Konservative Folkeparti   -7.75995824
## Ø. Enhedslisten – De Rød-Grønne-C. Det Konservative Folkeparti   -8.72563969
## V. Venstre, Danmarks Liberale Parti-C. Det Konservative Folkeparti  -0.07182361
## Ø. Enhedslisten – De Rød-Grønne-F. SF - Socialistisk Folkeparti  -1.93058901
## V. Venstre, Danmarks Liberale Parti-F. SF - Socialistisk Folkeparti  6.71547653
## V. Venstre, Danmarks Liberale Parti-Ø. Enhedslisten – De Rød-Grønne  7.65952234
##                                                                       upr
## B. Radikale Venstre-A. Socialdemokratiet                          1.45608734
## C. Det Konservative Folkeparti-A. Socialdemokratiet               5.65771047
## F. SF - Socialistisk Folkeparti-A. Socialdemokratiet             -1.13645590
## Ø. Enhedslisten – De Rød-Grønne-A. Socialdemokratiet             -2.07931318
## V. Venstre, Danmarks Liberale Parti-A. Socialdemokratiet          6.81679549
## C. Det Konservative Folkeparti-B. Radikale Venstre                5.30611349
## F. SF - Socialistisk Folkeparti-B. Radikale Venstre              -1.48500650
## Ø. Enhedslisten – De Rød-Grønne-B. Radikale Venstre              -2.42839345
## V. Venstre, Danmarks Liberale Parti-B. Radikale Venstre           6.46274318
## F. SF - Socialistisk Folkeparti-C. Det Konservative Folkeparti   -5.68662963
## Ø. Enhedslisten – De Rød-Grønne-C. Det Konservative Folkeparti   -6.63001658
## V. Venstre, Danmarks Liberale Parti-C. Det Konservative Folkeparti  2.26112005
## Ø. Enhedslisten – De Rød-Grønne-F. SF - Socialistisk Folkeparti   0.02152061
## V. Venstre, Danmarks Liberale Parti-F. SF - Socialistisk Folkeparti  8.92040778
## V. Venstre, Danmarks Liberale Parti-Ø. Enhedslisten – De Rød-Grønne  9.88543036
##                                                                     p adj
## B. Radikale Venstre-A. Socialdemokratiet                         0.8858773
## C. Det Konservative Folkeparti-A. Socialdemokratiet              0.0000000
## F. SF - Socialistisk Folkeparti-A. Socialdemokratiet             0.0000002
## Ø. Enhedslisten – De Rød-Grønne-A. Socialdemokratiet             0.0000000
## V. Venstre, Danmarks Liberale Parti-A. Socialdemokratiet          0.0000000
## C. Det Konservative Folkeparti-B. Radikale Venstre               0.0000000
## F. SF - Socialistisk Folkeparti-B. Radikale Venstre              0.0000000
## Ø. Enhedslisten – De Rød-Grønne-B. Radikale Venstre              0.0000000
## V. Venstre, Danmarks Liberale Parti-B. Radikale Venstre          0.0000000
## F. SF - Socialistisk Folkeparti-C. Det Konservative Folkeparti   0.0000000
## Ø. Enhedslisten – De Rød-Grønne-C. Det Konservative Folkeparti   0.0000000
## V. Venstre, Danmarks Liberale Parti-C. Det Konservative Folkeparti  0.0788640
## Ø. Enhedslisten – De Rød-Grønne-F. SF - Socialistisk Folkeparti  0.0590807
## V. Venstre, Danmarks Liberale Parti-F. SF - Socialistisk Folkeparti 0.0000000
## V. Venstre, Danmarks Liberale Parti-Ø. Enhedslisten – De Rød-Grønne 0.0000000
```

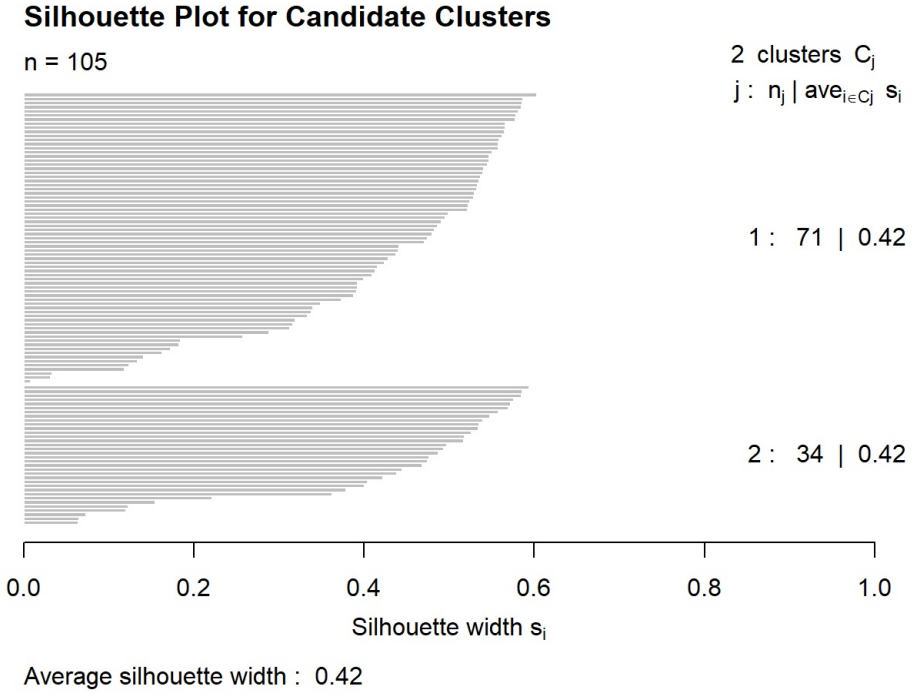
```
#Check homogeneity of variance
if (requireNamespace("car", quietly = TRUE)) {
cat("\n=== Levene's Test for Homogeneity of Variance ===\n")
print(car::leveneTest(PC1 ~ party, data = candidate_scores))
}
```

```
##
## === Levene's Test for Homogeneity of Variance ===
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  5  1.1883 0.3204
##       99
```

```
# Clustering Analysis

## Optimal Number of Clusters (Candidate-Level)

#Use first 6 PCs for clustering
pc_matrix <- candidate_scores %>%
select(starts_with("PC")) %>%
select(PC1:PC6) %>%
as.matrix()

#Determine optimal k using silhouette method
k_max <- min(10, nrow(pc_matrix) - 1)
fviz_nbclust(pc_matrix, FUN = kmeans, method = "silhouette", k.max = k_max) +
labs(title = "Optimal Number of Clusters (Silhouette Method)")
```




```
#Perform k-means with k=2 (based on typical left-right split)
k_cand <- 2
set.seed(123)
kmeans_model <- kmeans(pc_matrix, centers = k_cand, nstart = 50)

candidate_scores$cluster_cand <- factor(kmeans_model$cluster)

#Silhouette diagnostic
sil <- silhouette(as.numeric(candidate_scores$cluster_cand), dist(pc_matrix))
plot(sil, main = "Silhouette Plot for Candidate Clusters")
```




```
#Cluster composition by party
cat("\n=== Cluster Composition by Party ===\n")
```

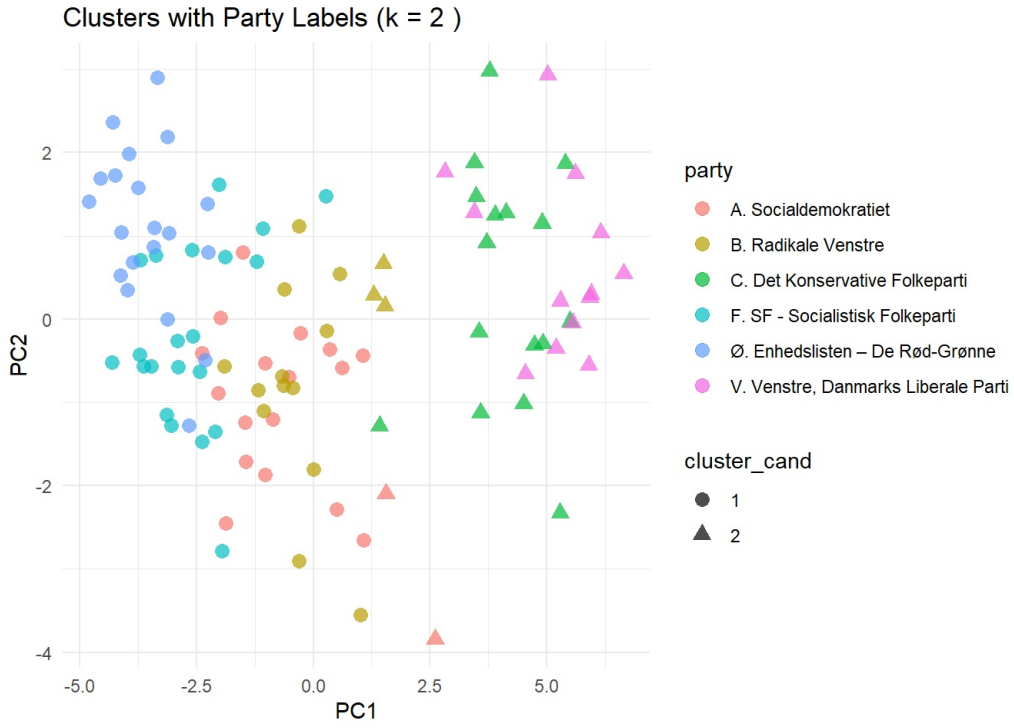
```
##
## === Cluster Composition by Party ===
```

```
print(table(candidate_scores$cluster_cand, candidate_scores$party))
```

```
##
##     A. Socialdemokratiet B. Radikale Venstre C. Det Konservative Folkeparti
##   1                   17                  13                              0
##   2                    2                   3                             16
##
##     F. SF - Socialistisk Folkeparti Ø. Enhedslisten – De Rød-Grønne
##   1                              21                              20
##   2                               0                               0
##
##     V. Venstre, Danmarks Liberale Parti
##   1                                   0
##   2                                  13
```

```
#Visualize clusters
ggplot(candidate_scores, aes(x = PC1, y = PC2, color = cluster_cand)) +
geom_point(size = 3, alpha = 0.7) +
theme_minimal() +
labs(title = paste("K-means Clustering (k =", k_cand, ")"),
subtitle = "Clusters based on PC1-PC6")
```


```
#Clusters with party overlaid
ggplot(candidate_scores, aes(x = PC1, y = PC2, color = party, shape = cluster_cand)) +
geom_point(size = 3, alpha = 0.7) +
theme_minimal() +
labs(title = paste("Clusters with Party Labels (k =", k_cand, ")"))
```




```
## Jaccard Stability Analysis

#Bootstrap stability of clusters using Jaccard index
set.seed(123)
clust_boot <- fpc::clusterboot(pc_matrix, B = 100,
clustermethod = kmeansCBI,
krange = k_cand,
seed = 123)
```

```
## boot 1
## boot 2
## boot 3
## boot 4
## boot 5
## boot 6
## boot 7
```

```
## boot 8
## boot 9
## boot 10
## boot 11
## boot 12
## boot 13
## boot 14
## boot 15
## boot 16
## boot 17
## boot 18
## boot 19
## boot 20
## boot 21
## boot 22
## boot 23
## boot 24
## boot 25
## boot 26
## boot 27
## boot 28
## boot 29
## boot 30
## boot 31
## boot 32
## boot 33
## boot 34
## boot 35
## boot 36
## boot 37
## boot 38
## boot 39
## boot 40
## boot 41
## boot 42
## boot 43
## boot 44
## boot 45
## boot 46
## boot 47
## boot 48
## boot 49
## boot 50
## boot 51
## boot 52
## boot 53
## boot 54
## boot 55
## boot 56
## boot 57
## boot 58
## boot 59
## boot 60
## boot 61
## boot 62
## boot 63
## boot 64
## boot 65
## boot 66
## boot 67
## boot 68
## boot 69
## boot 70
## boot 71
## boot 72
## boot 73
## boot 74
## boot 75
## boot 76
## boot 77
## boot 78
## boot 79
## boot 80
## boot 81
## boot 82
## boot 83
## boot 84
## boot 85
## boot 86
```

```
## boot 87
## boot 88
## boot 89
## boot 90
## boot 91
## boot 92
## boot 93
## boot 94
## boot 95
## boot 96
## boot 97
## boot 98
## boot 99
## boot 100
```

```
cat("\n=== Jaccard Stability (Bootstrap) ===\n")
```

```
##
## === Jaccard Stability (Bootstrap) ===
```

```
cat("Mean Jaccard coefficient per cluster:\n")
```

```
## Mean Jaccard coefficient per cluster:
```

```
print(clust_boot$bootmean)
```

```
## [1] 0.9728562 0.9475503
```

```
cat("\nNote: Values > 0.75 indicate stable clusters\n")
```

```
##
## Note: Values > 0.75 indicate stable clusters
```

```
# Predictive Modeling with Nested CV (NO DATA LEAKAGE)

## Prepare Full Dataset for CV

#Use entire filtered dataset (n=105) with cross-validation
#NO train/test split - CV provides unbiased estimates
X_full <- df %>% select(all_of(question_cols))
y_full_pretty <- factor(df$party)
y_full <- make_safe_levels(y_full_pretty)

cat("Total observations for nested CV:", nrow(X_full), "\n")
```

```
## Total observations for nested CV: 105
```

```
cat("Number of classes:", length(levels(y_full)), "\n")
```

```
## Number of classes: 6
```

```
cat("Observations per class:\n")
```

```
## Observations per class:
```

```
print(table(y_full_pretty))
```

```
## y_full_pretty
##                A. Socialdemokratiet                 B. Radikale Venstre
##                                  19                                  16
##      C. Det Konservative Folkeparti     F. SF - Socialistisk Folkeparti
##                                  16                                  21
##     Ø. Enhedslisten – De Rød-Grønne V. Venstre, Danmarks Liberale Parti
##                                  20                                  13
```

```
## Outer CV Loop: Performance Estimation

#Outer loop: 10-fold CV repeated 5 times for stable performance estimates
#This gives us 50 performance measurements per model
outer_ctrl <- trainControl(
method = "repeatedcv",
number = 10, # 10-fold CV (each fold ~10-11 observations)
repeats = 5, # Repeat 5 times for stability
classProbs = TRUE,
summaryFunction = defaultSummary, # Better metrics for multi-class
savePredictions = "final",
verboseIter = FALSE
)

cat("Outer CV configuration:\n")
```

```
## Outer CV configuration:
```

```
cat(" Method: 10-fold CV repeated 5 times\n")
```

```
##  Method: 10-fold CV repeated 5 times
```

```
cat(" Total model fits per algorithm: 50\n")
```

```
##  Total model fits per algorithm: 50
```

```
cat(" Each test fold: ~", round(nrow(X_full)/10), "observations\n")
```

```
##  Each test fold: ~ 10 observations
```

```
## Train Models with Nested CV

#Random Forest with nested CV
set.seed(123)
rf_cv <- train(
x = X_full,
y = y_full,
method = "rf",
trControl = outer_ctrl,
tuneLength = 5,
metric = "Accuracy",
importance = TRUE
)
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
cat("\n=== RANDOM FOREST: 10-FOLD CV (REPEATED 5x) ===\n")
```

```
##
## === RANDOM FOREST: 10-FOLD CV (REPEATED 5x) ===
```

```
print(rf_cv)
```

```
## Random Forest
##
## 105 samples
##  26 predictor
##   6 classes: 'A..Socialdemokratiet', 'B..Radikale.Venstre', 'C..Det.Konservative.Folkeparti', 'F..SF...Sociali
stisk.Folkeparti', 'Ø..Enhedslisten...De.Rød.Grønne', 'V..Venstre..Danmarks.Liberale.Parti'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 94, 95, 94, 95, 93, 95, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.7338889  0.6761003
##    8    0.7238384  0.6641349
##   14    0.7220000  0.6617329
##   20    0.6955455  0.6298635
##   26    0.6808889  0.6123900
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
cat("\nBest mtry:", rf_cv$bestTune$mtry, "\n")
```

```
##
## Best mtry: 2
```

```
#SVM Radial
set.seed(123)
svm_cv <- train(
x = X_full,
y = y_full,
method = "svmRadial",
preProcess = c("center", "scale"),
trControl = outer_ctrl,
tuneLength = 5,
metric = "Accuracy"
)
```

```
## Warning: Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
cat("\n=== SVM RADIAL: 10-FOLD CV (REPEATED 5x) ===\n")
```

```
##
## === SVM RADIAL: 10-FOLD CV (REPEATED 5x) ===
```

```
print(svm_cv)
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 105 samples
##  26 predictor
##   6 classes: 'A..Socialdemokratiet', 'B..Radikale.Venstre', 'C..Det.Konservative.Folkeparti', 'F..SF...Sociali
stisk.Folkeparti', 'Ø..Enhedslisten...De.Rød.Grønne', 'V..Venstre..Danmarks.Liberale.Parti'
##
## Pre-processing: centered (26), scaled (26)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 94, 95, 94, 95, 93, 95, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.5502525  0.4580683
##   0.50  0.6013333  0.5187188
##   1.00  0.5939798  0.5078518
##   2.00  0.6017677  0.5168871
##   4.00  0.5982020  0.5122843
##
## Tuning parameter 'sigma' was held constant at a value of 0.03102456
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.03102456 and C = 2.
```

```
#Multinomial Logistic Regression
set.seed(123)
lr_cv <- train(
x = X_full,
y = y_full,
method = "multinom",
preProcess = c("center", "scale"),
trControl = outer_ctrl,
tuneLength = 3,
metric = "Accuracy",
trace = FALSE
)
```

```
## Warning: Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
## Setting row names on a tibble is deprecated.
```

```
cat("\n=== MULTINOMIAL LOGISTIC REGRESSION: 10-FOLD CV (REPEATED 5x) ===\n")
```

```
##
## === MULTINOMIAL LOGISTIC REGRESSION: 10-FOLD CV (REPEATED 5x) ===
```

```
print(lr_cv)
```

```
## Penalized Multinomial Regression
##
## 105 samples
##  26 predictor
##   6 classes: 'A..Socialdemokratiet', 'B..Radikale.Venstre', 'C..Det.Konservative.Folkeparti', 'F..SF...Sociali
stisk.Folkeparti', 'Ø..Enhedslisten...De.Rød.Grønne', 'V..Venstre..Danmarks.Liberale.Parti'
##
## Pre-processing: centered (26), scaled (26)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 94, 95, 94, 95, 93, 95, ...
## Resampling results across tuning parameters:
##
##   decay  Accuracy   Kappa
##   0e+00  0.5256768  0.4259126
##   1e-04  0.6190606  0.5385566
##   1e-01  0.6565960  0.5840768
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was decay = 0.1.
```

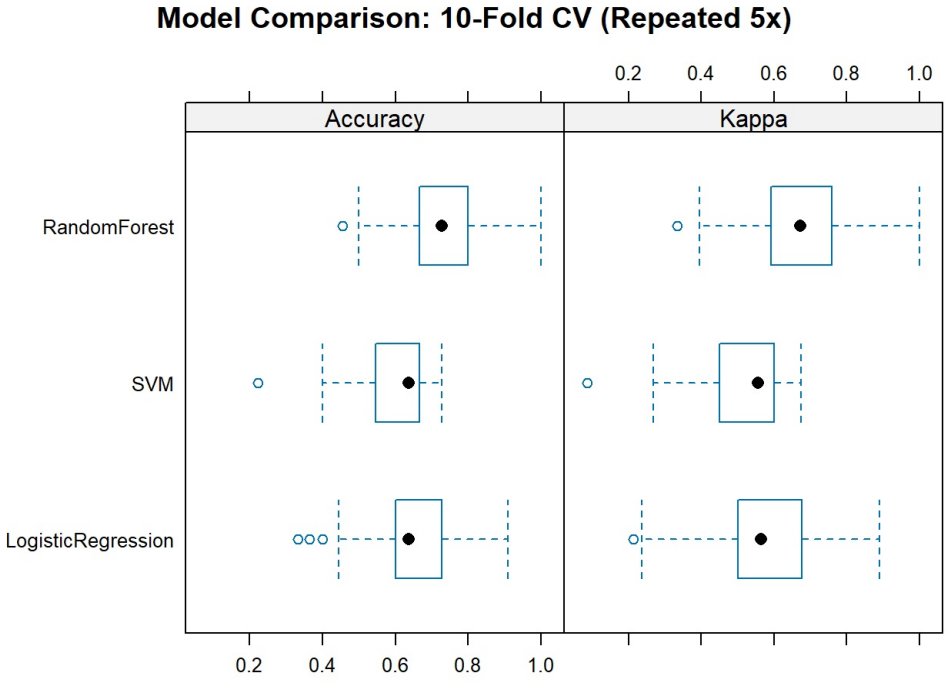
```
## Model Comparison

#Compare all models
results <- resamples(list(
RandomForest = rf_cv,
SVM = svm_cv,
LogisticRegression = lr_cv
))

cat("\n=== MODEL COMPARISON (50 CV ITERATIONS EACH) ===\n")
```

```
##
## === MODEL COMPARISON (50 CV ITERATIONS EACH) ===
```

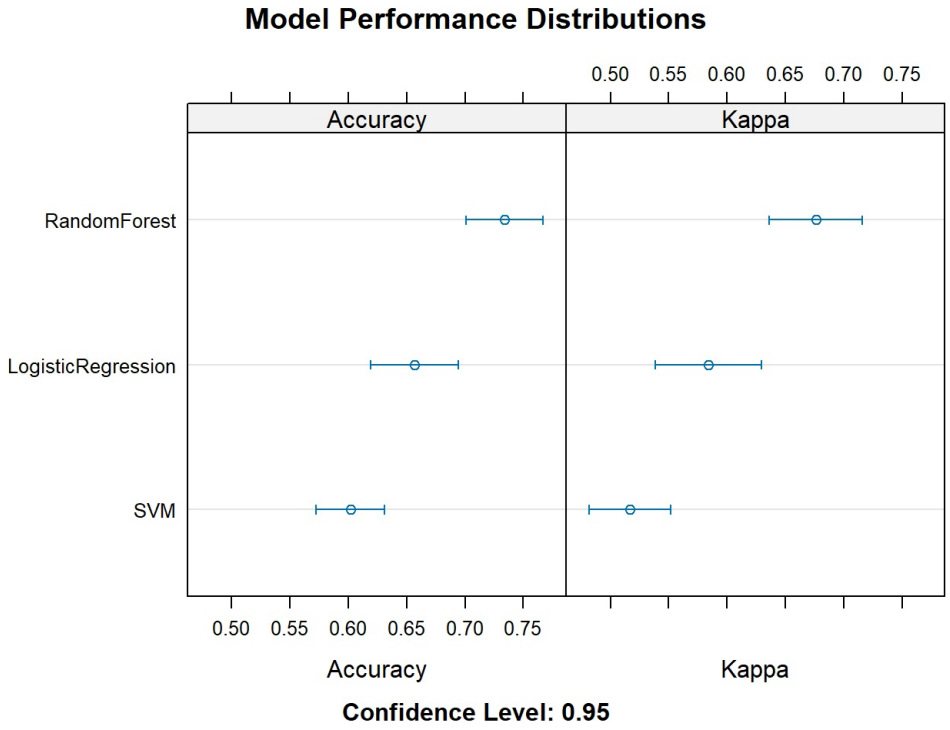
```
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: RandomForest, SVM, LogisticRegression
## Number of resamples: 50
##
## Accuracy
##                          Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## RandomForest       0.4545455 0.6666667 0.7272727 0.7338889 0.8000000 1.0000000
## SVM                0.2222222 0.5549242 0.6363636 0.6017677 0.6666667 0.7272727
## LogisticRegression 0.3333333 0.6000000 0.6363636 0.6565960 0.7272727 0.9090909
##                    NA's
## RandomForest          0
## SVM                   0
## LogisticRegression    0
##
## Kappa
##                           Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## RandomForest       0.33333333 0.5931818 0.6716337 0.6761003 0.7590361 1.0000000
## SVM                0.08695652 0.4625000 0.5555556 0.5168871 0.5977273 0.6732673
## LogisticRegression 0.21311475 0.5030488 0.5643564 0.5840768 0.6748529 0.8900000
##                    NA's
## RandomForest          0
## SVM                   0
## LogisticRegression    0
```

```
#Visualize
bwplot(results, main = "Model Comparison: 10-Fold CV (Repeated 5x)")
```




```
dotplot(results, main = "Model Performance Distributions")
```




```
#Statistical comparison
cat("\n=== STATISTICAL SIGNIFICANCE TESTS ===\n")
```

```
##
## === STATISTICAL SIGNIFICANCE TESTS ===
```

```
diff_results <- diff(results)
summary(diff_results)
```

```
##
## Call:
## summary.diff.resamples(object = diff_results)
##
## p-value adjustment: bonferroni
## Upper diagonal: estimates of the difference
## Lower diagonal: p-value for H0: difference = 0
##
## Accuracy
##                     RandomForest SVM       LogisticRegression
## RandomForest                      0.13212   0.07729
## SVM                 3.821e-07              -0.05483
## LogisticRegression 0.0008952    0.0211941
##
## Kappa
##                     RandomForest SVM      LogisticRegression
## RandomForest                      0.15921  0.09202
## SVM                 4.734e-07             -0.06719
## LogisticRegression 0.001145     0.019665
```

```
## Report Final Performance with Confidence Intervals

#Extract CV results with confidence intervals
get_cv_stats <- function(model, model_name) {
cv_results <- model$resample

tibble(
Model = model_name,
Mean_Accuracy = mean(cv_results$Accuracy),
SD_Accuracy = sd(cv_results$Accuracy),
CI_Lower = mean(cv_results$Accuracy) - 1.96 * sd(cv_results$Accuracy) / sqrt(nrow(cv_results)),
CI_Upper = mean(cv_results$Accuracy) + 1.96 * sd(cv_results$Accuracy) / sqrt(nrow(cv_results)),
Mean_Kappa = mean(cv_results$Kappa),
SD_Kappa = sd(cv_results$Kappa)
)
}

performance_summary <- bind_rows(
get_cv_stats(rf_cv, "Random Forest"),
get_cv_stats(svm_cv, "SVM Radial"),
get_cv_stats(lr_cv, "Logistic Regression")
)

cat("\n=== FINAL MODEL PERFORMANCE (WITH 95% CI) ===\n")
```

```
##
## === FINAL MODEL PERFORMANCE (WITH 95% CI) ===
```

```
print(performance_summary, digits = 3)
```

```
## Warning: `...` must be empty in `format.tbl()`
## Caused by error in `format_tbl()`:
## ! `...` must be empty.
## ✖ Problematic argument:
## • digits = 3
```

```
## # A tibble: 3 × 7
##   Model          Mean_Accuracy SD_Accuracy CI_Lower CI_Upper Mean_Kappa SD_Kappa
##   <chr>                  <dbl>       <dbl>    <dbl>    <dbl>      <dbl>    <dbl>
## 1 Random Forest          0.734       0.115    0.702    0.766      0.676    0.141
## 2 SVM Radial             0.602       0.103    0.573    0.630      0.517    0.123
## 3 Logistic Regr…         0.657       0.133    0.620    0.693      0.584    0.159
```

```
#Select best model
best_model_name <- performance_summary$Model[which.max(performance_summary$Mean_Accuracy)]
best_model <- switch(best_model_name,
"Random Forest" = rf_cv,
"SVM Radial" = svm_cv,
"Logistic Regression" = lr_cv)

cat("\n==> BEST MODEL:", best_model_name, "\n")
```

```
##
## ==> BEST MODEL: Random Forest
```

```
cat(" Mean CV Accuracy:", round(max(performance_summary$Mean_Accuracy), 3), "\n")
```

```
##  Mean CV Accuracy: 0.734
```

```
# Feature Importance Analysis (NO LEAKAGE)

## Strategy: Bootstrap Stability Selection

#Bootstrap feature importance across CV folds
#This ensures features are selected WITHOUT seeing full dataset
cat("\n=== FEATURE IMPORTANCE: BOOTSTRAP STABILITY APPROACH ===\n")
```

```
##
## === FEATURE IMPORTANCE: BOOTSTRAP STABILITY APPROACH ===
```

```
cat("Computing importance across 100 bootstrap samples...\n")
```

```
## Computing importance across 100 bootstrap samples...
```

```
set.seed(123)
n_boot <- 100
importance_boot <- matrix(NA, nrow = n_boot, ncol = length(question_cols))
colnames(importance_boot) <- question_cols

for (i in 1:n_boot) {

#Bootstrap sample
boot_idx <- sample(1:nrow(X_full), replace = TRUE)
X_boot <- X_full[boot_idx, ]
y_boot <- y_full[boot_idx]

#Fit RF and extract importance
rf_boot <- randomForest(x = X_boot, y = y_boot, importance = TRUE, ntree = 500)

#Get mean importance across classes
imp_matrix <- importance(rf_boot, type = 1) # Mean decrease in accuracy
importance_boot[i, ] <- rowMeans(as.matrix(imp_matrix))

if (i %% 20 == 0) cat(" Completed", i, "/ 100 bootstrap samples\n")
}
```

```
##  Completed 20 / 100 bootstrap samples
##  Completed 40 / 100 bootstrap samples
##  Completed 60 / 100 bootstrap samples
##  Completed 80 / 100 bootstrap samples
##  Completed 100 / 100 bootstrap samples
```

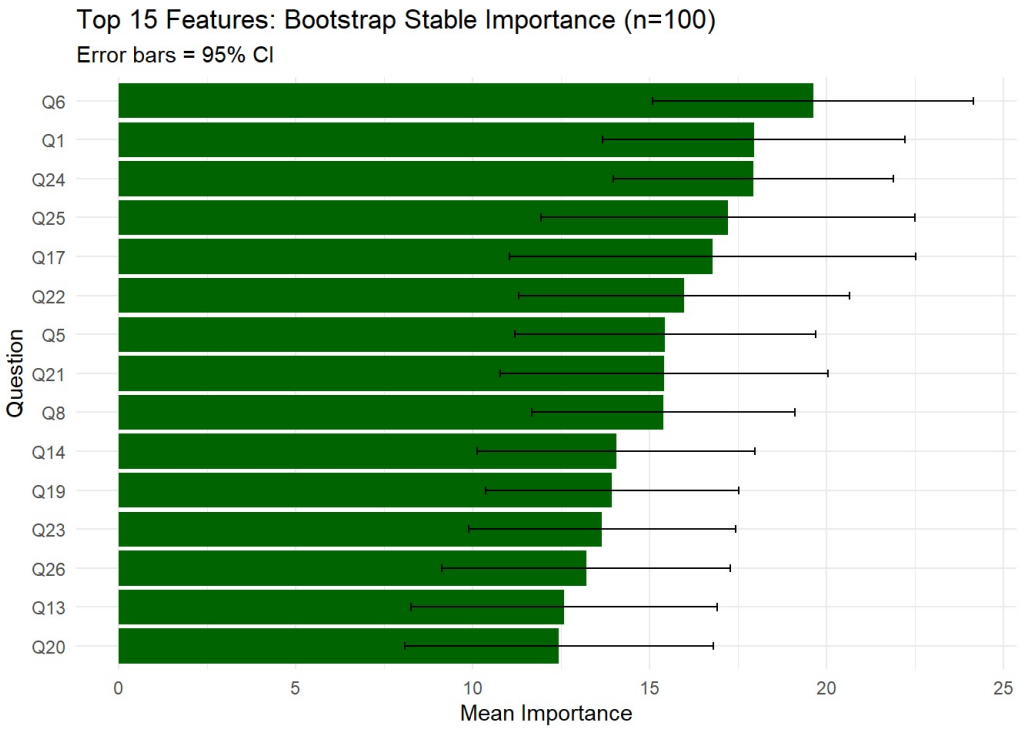
```
#Calculate stable importance metrics
importance_stable <- tibble(
Question = colnames(importance_boot),
Mean_Importance = colMeans(importance_boot),
SD_Importance = apply(importance_boot, 2, sd),
CI_Lower = colMeans(importance_boot) - 1.96 * apply(importance_boot, 2, sd),
CI_Upper = colMeans(importance_boot) + 1.96 * apply(importance_boot, 2, sd),
Stability = apply(importance_boot, 2, function(x) sum(x > median(importance_boot)) / n_boot)
) %>%
arrange(desc(Mean_Importance))

cat("\n=== TOP 15 FEATURES: BOOTSTRAP STABLE IMPORTANCE ===\n")
```

```
##
## === TOP 15 FEATURES: BOOTSTRAP STABLE IMPORTANCE ===
```

```
print(importance_stable %>% slice_head(n = 15))
```

```
## # A tibble: 15 × 6
##    Question Mean_Importance SD_Importance CI_Lower CI_Upper Stability
##    <chr>              <dbl>         <dbl>    <dbl>    <dbl>     <dbl>
##  1 Q6                  19.6          2.32    15.1      24.2      1
##  2 Q1                  18.0          2.18    13.7      22.2      0.99
##  3 Q24                 17.9          2.02    14.0      21.9      1
##  4 Q25                 17.2          2.69    11.9      22.5      0.94
##  5 Q17                 16.8          2.93    11.0      22.5      0.93
##  6 Q22                 16.0          2.38    11.3      20.6      0.88
##  7 Q5                  15.4          2.17    11.2      19.7      0.89
##  8 Q21                 15.4          2.36    10.8      20.0      0.85
##  9 Q8                  15.4          1.90    11.7      19.1      0.91
## 10 Q14                 14.1          2.00    10.1      18.0      0.68
## 11 Q19                 13.9          1.83    10.4      17.5      0.7
## 12 Q23                 13.7          1.92     9.90     17.4      0.6
## 13 Q26                 13.2          2.08     9.13     17.3      0.48
## 14 Q13                 12.6          2.21     8.26     16.9      0.46
## 15 Q20                 12.4          2.23     8.08     16.8      0.37
```

```
ggplot(importance_stable[1:15,], aes(x = reorder(Question, Mean_Importance),
y = Mean_Importance)) +
geom_col(fill = "darkgreen") +
geom_errorbar(aes(ymin = CI_Lower, ymax = CI_Upper), width = 0.2) +
coord_flip() +
theme_minimal() +
labs(title = "Top 15 Features: Bootstrap Stable Importance (n=100)",
subtitle = "Error bars = 95% CI",
x = "Question", y = "Mean Importance")
```




```
## Alternative: Permutation Importance (CV-Based)

#Compute permutation importance within CV framework
cat("\n=== FEATURE IMPORTANCE: CV-BASED PERMUTATION ===\n")
```

```
##
## === FEATURE IMPORTANCE: CV-BASED PERMUTATION ===
```

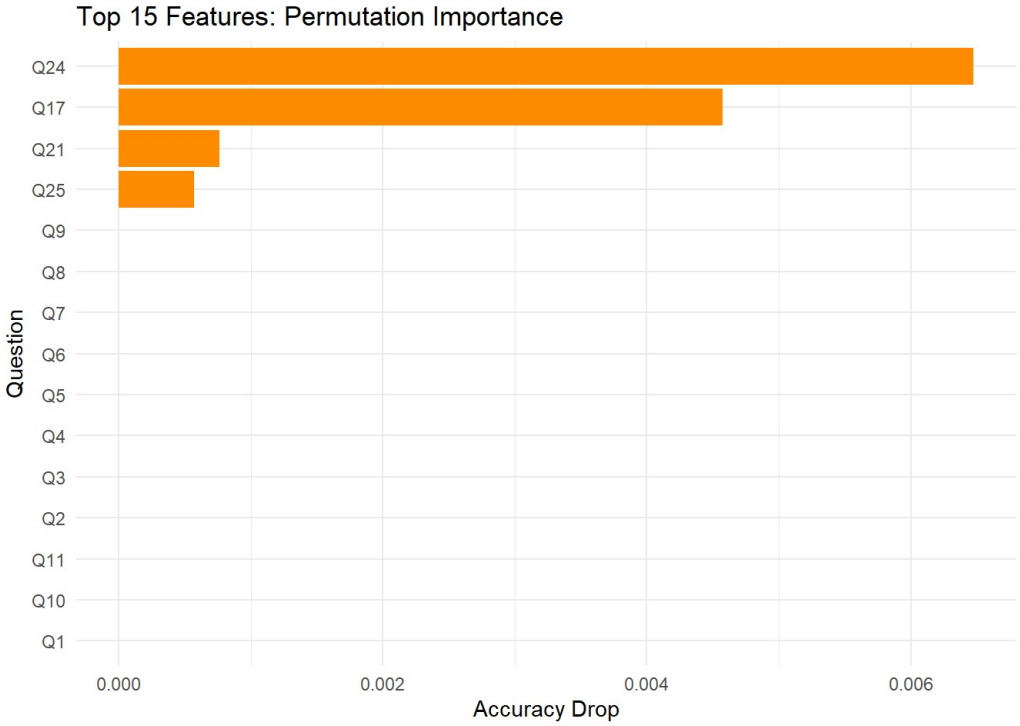
```
set.seed(123)
vi_perm_cv <- vip::vi_permute(
object = best_model,
feature_names = question_cols,
train = as.data.frame(lapply(X_full, as.numeric)),
target = y_full,
metric = "Accuracy",
nsim = 50,
pred_wrapper = function(object, newdata) predict(object, newdata = newdata)
)

vi_perm_cv <- vi_perm_cv %>%
rename(Question = Variable, Perm_Importance = Importance) %>%
arrange(desc(Perm_Importance))

cat("\n=== TOP 15 FEATURES: PERMUTATION IMPORTANCE ===\n")
```

```
##
## === TOP 15 FEATURES: PERMUTATION IMPORTANCE ===
```

```
print(vi_perm_cv %>% slice_head(n = 15))
```

```
## # A tibble: 15 × 3
##    Question Perm_Importance   StDev
##    <chr>              <dbl>   <dbl>
##  1 Q24             0.00648  0.00449
##  2 Q17             0.00457  0.00481
##  3 Q21             0.000762 0.00261
##  4 Q25             0.000571 0.00228
##  5 Q1              0        0
##  6 Q2              0        0
##  7 Q3              0        0
##  8 Q4              0        0
##  9 Q5              0        0
## 10 Q6              0        0
## 11 Q7              0        0
## 12 Q8              0        0
## 13 Q9              0        0
## 14 Q10             0        0
## 15 Q11             0        0
```

```
ggplot(vi_perm_cv[1:15,], aes(x = reorder(Question, Perm_Importance),
y = Perm_Importance)) +
geom_col(fill = "darkorange") +
coord_flip() +
theme_minimal() +
labs(title = "Top 15 Features: Permutation Importance",
x = "Question", y = "Accuracy Drop")
```




```
## SHAP Values (Full Dataset - Descriptive Only)

#SHAP on full model for interpretation (not for selection)
#ACKNOWLEDGE: This uses full dataset, so for INTERPRETATION not SELECTION
cat("\n=== SHAP VALUES: GLOBAL FEATURE IMPORTANCE (INTERPRETIVE) ===\n")
```

```
##
## === SHAP VALUES: GLOBAL FEATURE IMPORTANCE (INTERPRETIVE) ===
```

```
cat("NOTE: SHAP computed on full dataset for interpretation only.\n")
```

```
## NOTE: SHAP computed on full dataset for interpretation only.
```

```
cat("Feature SELECTION uses bootstrap/permutation methods above.\n\n")
```

```
## Feature SELECTION uses bootstrap/permutation methods above.
```

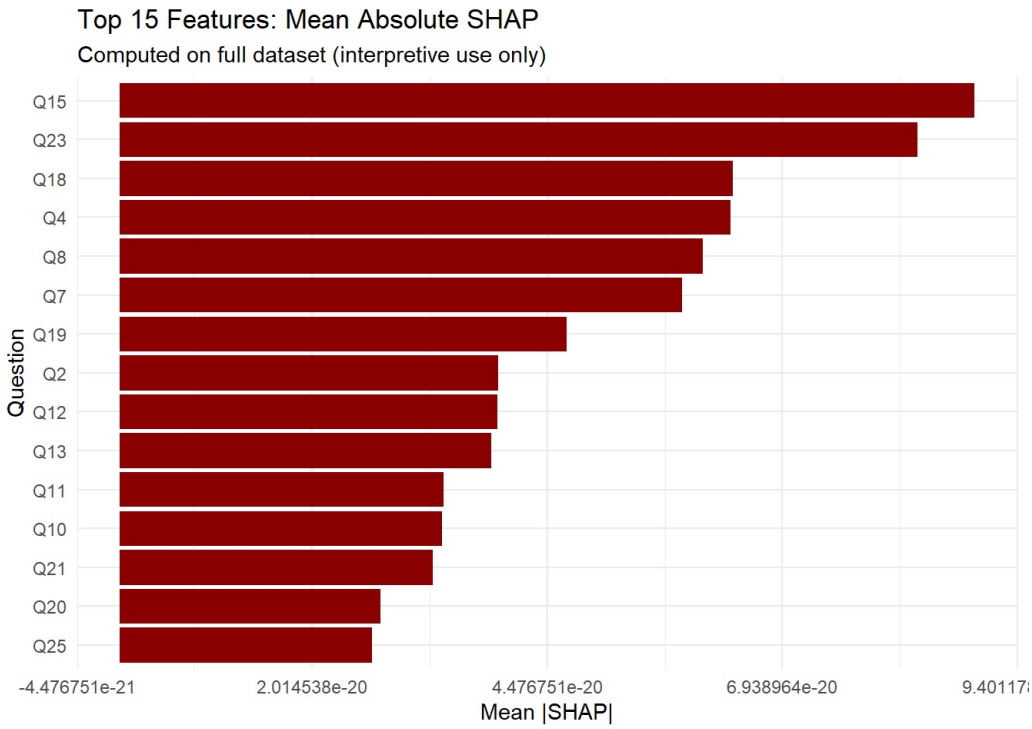
```
rf_final <- best_model$finalModel
X_numeric <- as.data.frame(lapply(X_full, as.numeric))

predict_wrap <- function(object, newdata) {
as.matrix(predict(object, newdata = newdata, type = "prob"))
}

set.seed(123)
bg_idx <- sample(1:nrow(X_numeric), min(100, nrow(X_numeric)))
bg <- X_numeric[bg_idx, ]

set.seed(123)
shap_values <- fastshap::explain(
object = rf_final,
X = X_numeric,
pred_wrapper = predict_wrap,
nsim = 100,
newdata = bg
)

if (is.list(shap_values)) {
shap_mat <- Reduce("+", lapply(shap_values, abs)) / length(shap_values)
} else {
shap_mat <- abs(shap_values)
}

shap_summary <- tibble(
Question = colnames(shap_mat),
Mean_SHAP = colMeans(shap_mat, na.rm = TRUE)
) %>%
arrange(desc(Mean_SHAP))

cat("\n=== TOP 15 FEATURES: MEAN |SHAP| ===\n")
```

```
##
## === TOP 15 FEATURES: MEAN |SHAP| ===
```

```
print(shap_summary %>% slice_head(n = 15))
```

```
## # A tibble: 15 × 2
##    Question Mean_SHAP
##    <chr>        <dbl>
##  1 Q15       8.95e-20
##  2 Q23       8.35e-20
##  3 Q18       6.42e-20
##  4 Q4        6.40e-20
##  5 Q8        6.11e-20
##  6 Q7        5.89e-20
##  7 Q19       4.68e-20
##  8 Q2        3.96e-20
##  9 Q12       3.96e-20
## 10 Q13       3.89e-20
## 11 Q11       3.39e-20
## 12 Q10       3.38e-20
## 13 Q21       3.28e-20
## 14 Q20       2.73e-20
## 15 Q25       2.64e-20
```

```
ggplot(shap_summary[1:15,], aes(x = reorder(Question, Mean_SHAP), y = Mean_SHAP)) +
geom_col(fill = "darkred") +
coord_flip() +
theme_minimal() +
labs(title = "Top 15 Features: Mean Absolute SHAP",
subtitle = "Computed on full dataset (interpretive use only)",
x = "Question", y = "Mean |SHAP|")
```




```
## Consensus Ranking (Properly Weighted)

#Combine bootstrap and permutation importance (both CV-based)
#SHAP included but weighted lower since it uses full data
consensus_df <- importance_stable %>%
select(Question, Bootstrap_Imp = Mean_Importance) %>%
left_join(vi_perm_cv %>% select(Question, Perm_Importance), by = "Question") %>%
left_join(shap_summary %>% select(Question, SHAP = Mean_SHAP), by = "Question") %>%
mutate(
# Normalize to 0-1
Bootstrap_norm = normalize_01(Bootstrap_Imp),
Perm_norm = normalize_01(Perm_Importance),
SHAP_norm = normalize_01(SHAP),


# Consensus: Bootstrap=50%, Permutation=40%, SHAP=10%
# (Lower weight for SHAP since it uses full data)
Consensus_Score = 0.50 * Bootstrap_norm +
                  0.40 * Perm_norm +
                  0.10 * SHAP_norm
) %>%
arrange(desc(Consensus_Score))

cat("\n=== CONSENSUS FEATURE RANKING (NO LEAKAGE) ===\n")
```

```
##
## === CONSENSUS FEATURE RANKING (NO LEAKAGE) ===
```

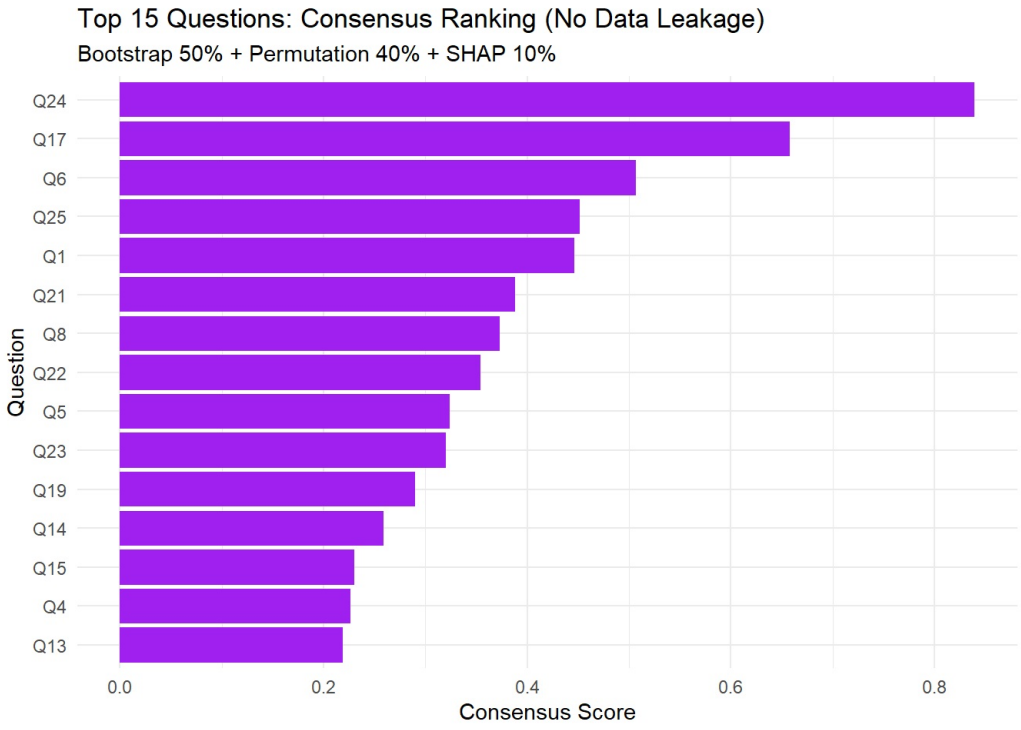
```
cat("Weights: Bootstrap=50%, Permutation=40%, SHAP=10%\n\n")
```

```
## Weights: Bootstrap=50%, Permutation=40%, SHAP=10%
```

```
print(consensus_df %>%
select(Question, Consensus_Score, Bootstrap_norm, Perm_norm, SHAP_norm) %>%
slice_head(n = 15))
```

```
## # A tibble: 15 × 5
##    Question Consensus_Score Bootstrap_norm Perm_norm SHAP_norm
##    <chr>              <dbl>          <dbl>     <dbl>     <dbl>
##  1 Q24                0.839          0.845    1         0.167
##  2 Q17                0.658          0.740    0.706     0.0562
##  3 Q6                 0.507          1        0         0.0665
##  4 Q25                0.451          0.780    0.0882    0.260
##  5 Q1                 0.446          0.847    0         0.224
##  6 Q21                0.388          0.615    0.118     0.335
##  7 Q8                 0.373          0.613    0         0.667
##  8 Q22                0.354          0.666    0         0.208
##  9 Q5                 0.324          0.618    0         0.151
## 10 Q23                0.320          0.455    0         0.929
## 11 Q19                0.290          0.481    0         0.499
## 12 Q14                0.259          0.492    0         0.130
## 13 Q15                0.230          0.261    0         1
## 14 Q4                 0.226          0.313    0         0.700
## 15 Q13                0.219          0.357    0         0.406
```

```
ggplot(consensus_df[1:15,], aes(x = reorder(Question, Consensus_Score),
y = Consensus_Score)) +
geom_col(fill = "purple") +
coord_flip() +
theme_minimal() +
labs(title = "Top 15 Questions: Consensus Ranking (No Data Leakage)",
subtitle = "Bootstrap 50% + Permutation 40% + SHAP 10%",
x = "Question", y = "Consensus Score")
```




```
#Extract top 10 for survey use
top10_questions <- consensus_df$Question[1:10]
cat("\n=== TOP 10 MOST DISCRIMINATING QUESTIONS (FOR SURVEY) ===\n")
```

```
##
## === TOP 10 MOST DISCRIMINATING QUESTIONS (FOR SURVEY) ===
```

```
cat(paste(top10_questions, collapse = ", "), "\n")
```

```
## Q24, Q17, Q6, Q25, Q1, Q21, Q8, Q22, Q5, Q23
```

```
cat("\nThese questions selected using CV-based methods (no data leakage).\n")
```

```
##
## These questions selected using CV-based methods (no data leakage).
```

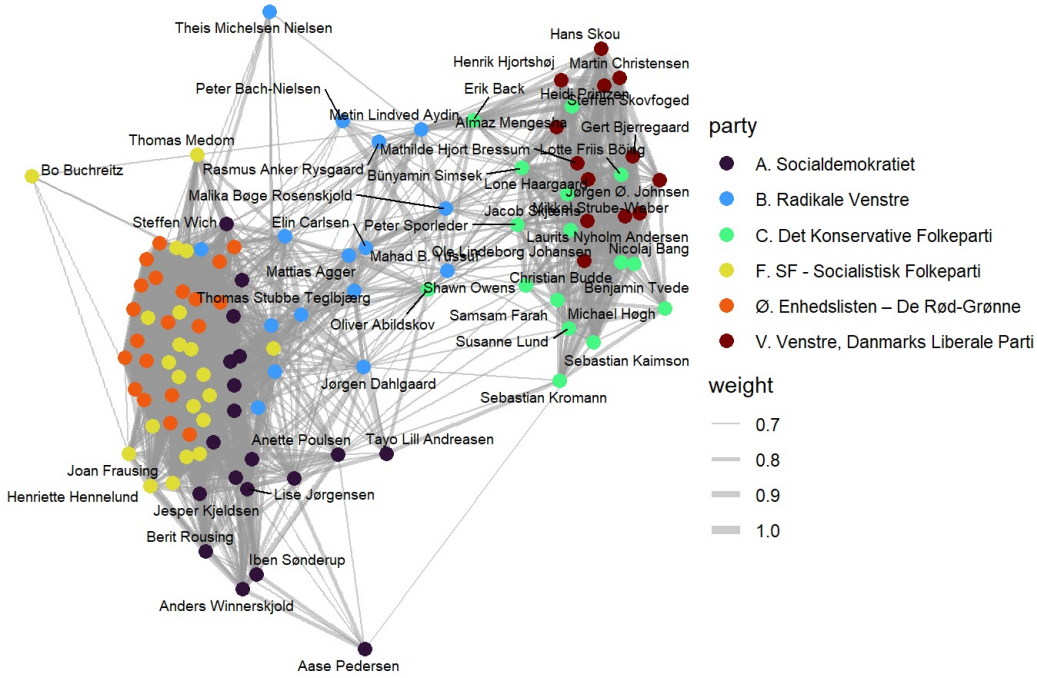
```
# Network Analysis: Direction Agreement

#Build network based on direction agreement (sign concordance)
build_direction_network <- function(question_subset, data_df,
threshold = 0.7, title = NULL) {

#Extract answer direction (sign)
A <- sign(as.matrix(select(data_df, all_of(question_subset))))
n_nodes <- nrow(A)

#Calculate pairwise direction agreement
edges <- list()
idx <- 1
for (i in 1:(n_nodes-1)) {
for (j in (i+1):n_nodes) {
agreement <- mean(A[i, ] == A[j, ], na.rm = TRUE)
if (!is.na(agreement) && agreement >= threshold) {
edges[[idx]] <- data.frame(from = i, to = j, weight = agreement)
idx <- idx + 1
}
}
}

if (length(edges) == 0) {
message("No edges above threshold for ", title)
return(NULL)
}

edges_df <- bind_rows(edges)

nodes_df <- tibble(
id = 1:n_nodes,
label = data_df$candidate_name,
party = data_df$party,
title = paste0(data_df$candidate_name, " (", data_df$party, ")")
)

g <- igraph::graph_from_data_frame(edges_df, vertices = nodes_df, directed = FALSE)

#Assign colors by party
parties <- sort(unique(nodes_df$party))
pal <- viridis::turbo(length(parties))
party_col_map <- setNames(pal, parties)
V(g)$color <- party_col_map[as.character(V(g)$party)]

#Plot network
p <- ggraph(g, layout = "fr") +
geom_edge_link(aes(width = weight), alpha = 0.5, color = "grey60") +
geom_node_point(aes(color = party), size = 3) +
geom_node_text(aes(label = label), repel = TRUE, size = 2.5) +
scale_color_manual(values = party_col_map) +
scale_edge_width(range = c(0.5, 2)) +
theme_void() +
labs(title = title)

print(p)

return(list(graph = g, nodes = nodes_df, edges = edges_df))
}

#Network based on top 10 consensus questions
cat("\n=== BUILDING NETWORK: TOP 10 CONSENSUS QUESTIONS ===\n")
```

```
##
## === BUILDING NETWORK: TOP 10 CONSENSUS QUESTIONS ===
```

```
network_top10 <- build_direction_network(
question_subset = top10_questions,
data_df = df,
threshold = 0.7,
title = "Candidate Network: Top 10 Most Discriminating Questions (threshold=0.7)"
)
```

```
## Warning: ggrepel: 54 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Candidate Network: Top 10 Most Discriminating Questions (threshold=0.7)




```
#Network statistics
if (!is.null(network_top10)) {
g <- network_top10$graph
cat("\nNetwork Statistics:\n")
cat(" Nodes:", vcount(g), "\n")
cat(" Edges:", ecount(g), "\n")
cat(" Density:", edge_density(g), "\n")
cat(" Average degree:", mean(degree(g)), "\n")
}
```

```
##
## Network Statistics:
##  Nodes: 105
##  Edges: 2043
##  Density: 0.3741758
##  Average degree: 38.91429
```

```
# Session Information

sessionInfo()
```

```
## R version 4.4.1 (2024-06-14 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26200)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=Polish_Poland.utf8  LC_CTYPE=Polish_Poland.utf8
## [3] LC_MONETARY=Polish_Poland.utf8 LC_NUMERIC=C
## [5] LC_TIME=Polish_Poland.utf8
##
## time zone: Europe/Paris
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] cluster_2.1.6        stringr_1.5.1        fpc_2.2-13
##  [4] vip_0.4.1            ggcorrplot_0.1.4.1   pheatmap_1.0.13
##  [7] viridis_0.6.5        viridisLite_0.4.2    visNetwork_2.1.4
## [10] ggraph_2.2.1         tidygraph_1.3.1      igraph_2.1.4
## [13] tibble_3.2.1         fastshap_0.1.1       MASS_7.3-65
## [16] nnet_7.3-19          randomForest_4.7-1.2 caret_7.0-1
## [19] lattice_0.22-6       psych_2.5.6          factoextra_1.0.7
## [22] ggplot2_4.0.1        tidyr_1.3.1          dplyr_1.1.4
## [25] readr_2.1.5
##
## loaded via a namespace (and not attached):
##   [1] RColorBrewer_1.1-3   rstudioapi_0.17.1    jsonlite_1.8.8
##   [4] magrittr_2.0.3       modeltools_0.2-24    farver_2.1.2
##   [7] rmarkdown_2.28       vctrs_0.6.5          memoise_2.0.1
##  [10] rstatix_0.7.2        htmltools_0.5.8.1    broom_1.0.10
##  [13] Formula_1.2-5        pROC_1.19.0.1        sass_0.4.9
##  [16] parallelly_1.45.1    bslib_0.8.0          htmlwidgets_1.6.4
##  [19] plyr_1.8.9           lubridate_1.9.3      cachem_1.1.0
##  [22] lifecycle_1.0.4      iterators_1.0.14     pkgconfig_2.0.3
##  [25] Matrix_1.7-0         R6_2.5.1             fastmap_1.2.0
##  [28] future_1.67.0        digest_0.6.37        ggpubr_0.6.1
##  [31] labeling_0.4.3       yardstick_1.3.2      timechange_0.3.0
##  [34] polyclip_1.10-7      abind_1.4-8          compiler_4.4.1
##  [37] proxy_0.4-27         bit64_4.0.5          withr_3.0.2
##  [40] S7_0.2.0             backports_1.5.0      carData_3.0-5
##  [43] ggforce_0.5.0        ggsignif_0.6.4       lava_1.8.1
##  [46] ModelMetrics_1.2.2.2 tools_4.4.1          prabclus_2.3-4
##  [49] future.apply_1.20.0  glue_1.7.0           nlme_3.1-164
##  [52] grid_4.4.1           reshape2_1.4.4       generics_0.1.3
##  [55] recipes_1.3.1        gtable_0.3.6         tzdb_0.4.0
##  [58] class_7.3-22         data.table_1.16.0    hms_1.1.3
##  [61] car_3.1-3            utf8_1.2.4           flexmix_2.3-20
##  [64] ggrepel_0.9.6        foreach_1.5.2        pillar_1.10.1
##  [67] vroom_1.6.5          robustbase_0.99-6    splines_4.4.1
##  [70] tweenr_2.0.3         survival_3.6-4       bit_4.0.5
##  [73] tidyselect_1.2.1     knitr_1.48           gridExtra_2.3
##  [76] stats4_4.4.1         xfun_0.47            graphlayouts_1.2.2
##  [79] hardhat_1.4.2        diptest_0.77-2       timeDate_4041.110
##  [82] DEoptimR_1.1-4       stringi_1.8.4        yaml_2.3.10
##  [85] evaluate_1.0.3       codetools_0.2-20     kernlab_0.9-33
##  [88] cli_3.6.5            rpart_4.1.23         jquerylib_0.1.4
##  [91] Rcpp_1.0.13          globals_0.18.0       parallel_4.4.1
##  [94] gower_1.0.2          mclust_6.1.1         listenv_0.9.1
##  [97] ipred_0.9-15         scales_1.4.0         prodlim_2025.04.28
## [100] e1071_1.7-16         purrr_1.1.0          crayon_1.5.3
## [103] rlang_1.1.6          mnormt_2.1.1
```

```
# This gives you everything at once
cat("Column names:\n")
```

```
## Column names:
```

```
print(names(df))
```

```
##  [1] "candidate_name" "party"          "Q1"             "Q2"
##  [5] "Q3"             "Q4"             "Q5"             "Q6"
##  [9] "Q7"             "Q8"             "Q9"             "Q10"
## [13] "Q11"            "Q12"            "Q13"            "Q14"
## [17] "Q15"            "Q16"            "Q17"            "Q18"
## [21] "Q19"            "Q20"            "Q21"            "Q22"
## [25] "Q23"            "Q24"            "Q25"            "Q26"
```

```
cat("\nParty distribution:\n")
```

```
##
## Party distribution:
```

```
print(table(df$party))
```

```
##
##                A. Socialdemokratiet                 B. Radikale Venstre
##                                  19                                  16
##      C. Det Konservative Folkeparti     F. SF - Socialistisk Folkeparti
##                                  16                                  21
##     Ø. Enhedslisten – De Rød-Grønne V. Venstre, Danmarks Liberale Parti
##                                  20                                  13
```

```
cat("\nNumber of unique parties:", length(unique(df$party)), "\n")
```

```
##
## Number of unique parties: 6
```