Group 7 Data Science I Final Project

Rosalie Daniels, Martina Radoslavov, Ayaka Sanui

December 3, 2025

```
# Set-Up
library(tidyr)
library(tidycensus)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
library(here)
```

```
## here() starts at /Users/martinaradoslavov/Library/CloudStorage/OneDrive-WeillCornellMedicine/DataSci
```

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.5.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.1     v readr     2.1.5
## v ggplot2   4.0.1     v stringr   1.6.0
## v lubridate 1.9.4     v tibble    3.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(ggplot2)
library(cluster)
```

```r
library(GGally)
library(factoextra)
```

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

```r
library(ggcorrplot)
library(gtsummary)
library(gt)
library(clustertend)
```

```
## Package `clustertend` is deprecated.  Use package `hopkins` instead.
```

```r
library(NbClust)
library(clValid)
library(mclust)
```

```
## Package 'mclust' version 6.1.2
## Type 'citation("mclust")' for citing this R package in publications.
##
## Attaching package: 'mclust'
##
## The following object is masked from 'package:psych':
##
##     sim
##
## The following object is masked from 'package:purrr':
##
##     map
##
## The following object is masked from 'package:dplyr':
##
##     count
```

```r
library(sf)
```

```
## Linking to GEOS 3.13.0, GDAL 3.8.5, PROJ 9.5.1; sf_use_s2() is TRUE
```

```r
library(cluster)
library(patchwork)

# Variable list for ACS pull with explanation of variable from ACS Tables
variables <-
  c(
    # Race
    'B02001_001', # Total Population for Races
    'B02001_002', # White Alone
    'B02001_003', # Black or African American Alone
    'B02001_005', # Asian Alone
    'B02001_009', # Two or more races

    # Educational Attainment
    'B15003_001', # Total Education Population
    'B15003_017', # High School Diploma
    'B15003_022', # Bachelors Degree
    'B15003_025', # Doctorate Degree
```

```
    # Housing Status
    'B25003_001', # Total Housing Units
    'B25003_002', # Owner Occupied
    'B25003_003', # Renter Occupied

    # Geographic Mobility by Citizenship
    'B07007_001', # Total Population for Mobility
    'B07007_002', # Native Population
    'B07007_003', # Foreign Born Population

    # Poverty Status
    "B17001_001", # Total
    "B17001_002", # Income in the past 12 months below poverty level

    # Transportation to Work
    "B08006_001",  # Total workers
    "B08006_008",  # Public transportation

    # Travel time to work
    "B08303_001", # Total workers
    "B08303_013", # Workers with commute time 90 or more minutes

    # Family Characteristics/Structure
    "B09002_001",  # Total population in families
    "B09002_002",  # In married-couple families

    # Employment Status
    "B23025_001", # Total Population 16+
    "B23025_005",  # Unemployed

    # Internet
    "B28002_001", # Total
    "B28002_002" # With Internet Subscription in Household
  )

# Retrieves ACS estimates for counties in TX (2019-2023 5-year ACS)
tx_counties <- get_acs(
  geography = "county",
  state = "TX",
  variables = variables,
  year = 2023,
  survey = "acs5"
)
```

```
## Getting data from the 2019-2023 5-year ACS
```

```
# Pivot to Wide Format
tx_counties <- tx_counties %>%
  select(GEOID, NAME, variable, estimate) %>%
  pivot_wider(names_from = variable, values_from = estimate)

# Converting variables to percentages
acs_clean <- tx_counties %>%
  mutate(
```

```
    # Race
    percent_white = B02001_002 / B02001_001 * 100,
    percent_black = B02001_003 / B02001_001 * 100,
    percent_asian = B02001_005 / B02001_001 * 100,
    percent_two_or_more  = B02001_009 / B02001_001 * 100,

    # Education
    percent_hs_diploma = B15003_017 / B15003_001 * 100,
    percent_bachelors = B15003_022 / B15003_001 * 100,
    percent_doctorate = B15003_025 / B15003_001 * 100,

    # Housing
    percent_owner_occupied = B25003_002 / B25003_001 * 100,
    percent_renter = B25003_003 / B25003_001 * 100,

    # Mobility
    percent_native_us = B07007_002 / B07007_001 * 100,
    percent_foreign_born = B07007_003 / B07007_001 * 100,

    # Poverty
    percent_poverty = B17001_002 / B17001_001 * 100,

    # Public Transit
    percent_public_transit = B08006_008 / B08006_001 * 100,

    # Long Commute to Work
    percent_long_commute = B08303_013 / B08303_001 * 100,

    # Family Structure
    percent_married_family = B09002_002 / B09002_001 * 100,

    # Unemployment
    percent_unemployed = B23025_005 / B23025_001 * 100,

    # Internet
    percent_internet = B28002_002 / B28002_001 * 100
  )

# Dataset With the Percentage Variables/Data
# Rename to cleaner names
data <- acs_clean %>%
  select(NAME, percent_white, percent_black, percent_asian, percent_two_or_more,
         percent_hs_diploma, percent_bachelors, percent_doctorate, percent_owner_occupied,
         percent_renter, percent_native_us, percent_foreign_born,
         percent_poverty, percent_public_transit, percent_long_commute,
         percent_married_family, percent_unemployed, percent_internet) %>%
  rename(
    White = percent_white,
    Black = percent_black,
    Asian = percent_asian,
    Two_Races = percent_two_or_more,
    HS_Diploma = percent_hs_diploma,
    Bachelors = percent_bachelors,
```

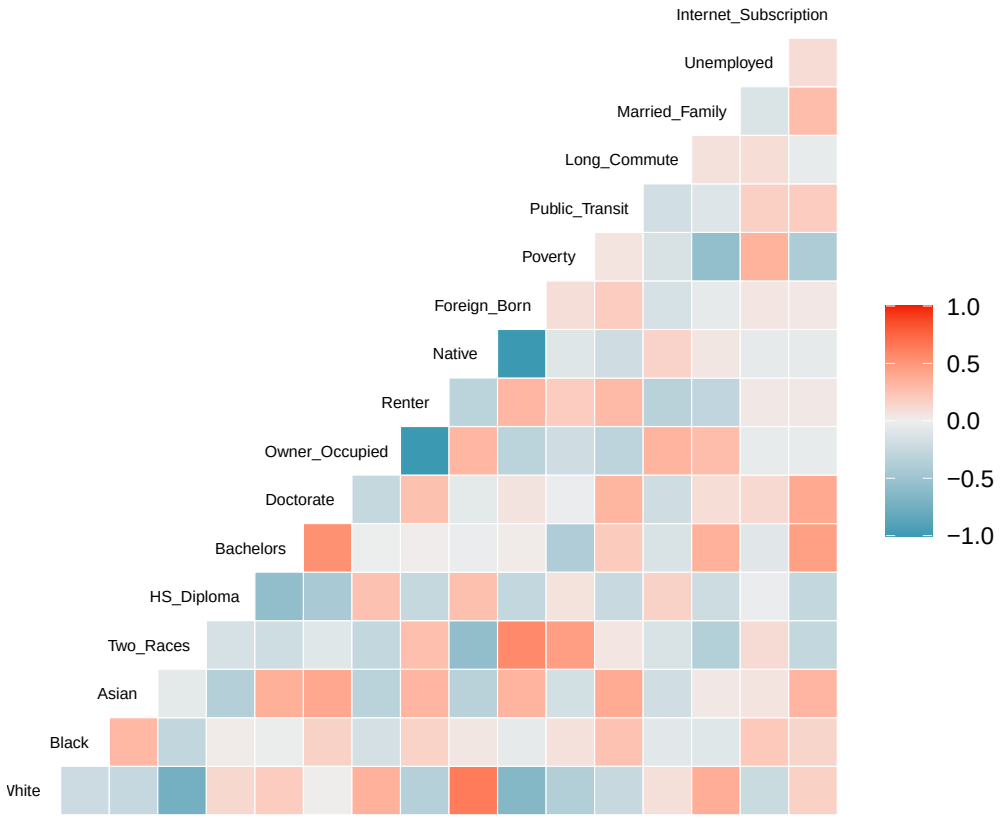
```
    Doctorate = percent_doctorate,
    Owner_Occupied = percent_owner_occupied,
    Renter = percent_renter,
    Native = percent_native_us,
    Foreign_Born = percent_foreign_born,
    Poverty = percent_poverty,
    Public_Transit = percent_public_transit,
    Long_Commute = percent_long_commute,
    Married_Family = percent_married_family,
    Unemployed = percent_unemployed,
    Internet_Subscription = percent_internet
  )

# EDA
# Create summary table with tbl_summary
summary_table <- data %>%
  select(-NAME) %>%
  tbl_summary(
    statistic = all_continuous() ~
      "{mean} ({sd}); {median} [{p25}, {p75}]",
    digits = all_continuous() ~ 2,
    label = list(
      White ~ "White (%)",
      Black ~ "Black (%)",
      Asian ~ "Asian (%)",
      Two_Races ~ "Two or More Races (%)",
      HS_Diploma ~ "High School Diploma (%)",
      Bachelors ~ "Bachelors Degree (%)",
      Doctorate ~ "Doctorate Degree (%)",
      Owner_Occupied ~ "Owner-Occupied Housing (%)",
      Renter ~ "Renter-Occupied Housing (%)",
      Native ~ "Native Population (%)",
      Foreign_Born ~ "Foreign-Born Population (%)",
      Poverty ~ "Below Poverty Line (%)",
      Public_Transit ~ "Public Transit Use (%)",
      Long_Commute ~ "Long Commute (90+ min) (%)",
      Married_Family ~ "Married-Couple Family (%)",
      Unemployed ~ "Unemployment Rate (%)",
      Internet_Subscription ~ "Home Internet Subscription (%)"
    )
  ) %>%
  modify_caption("Table 1: Socioeconomic Indicators at County Level")

# Output
summary_table
```

```
# Correlation Heatmap
ggcorr(data %>%
         select(-NAME),
       hjust = .9,
       size = 2)
```

Table 1: Table 1: Socioeconomic Indicators at County Level

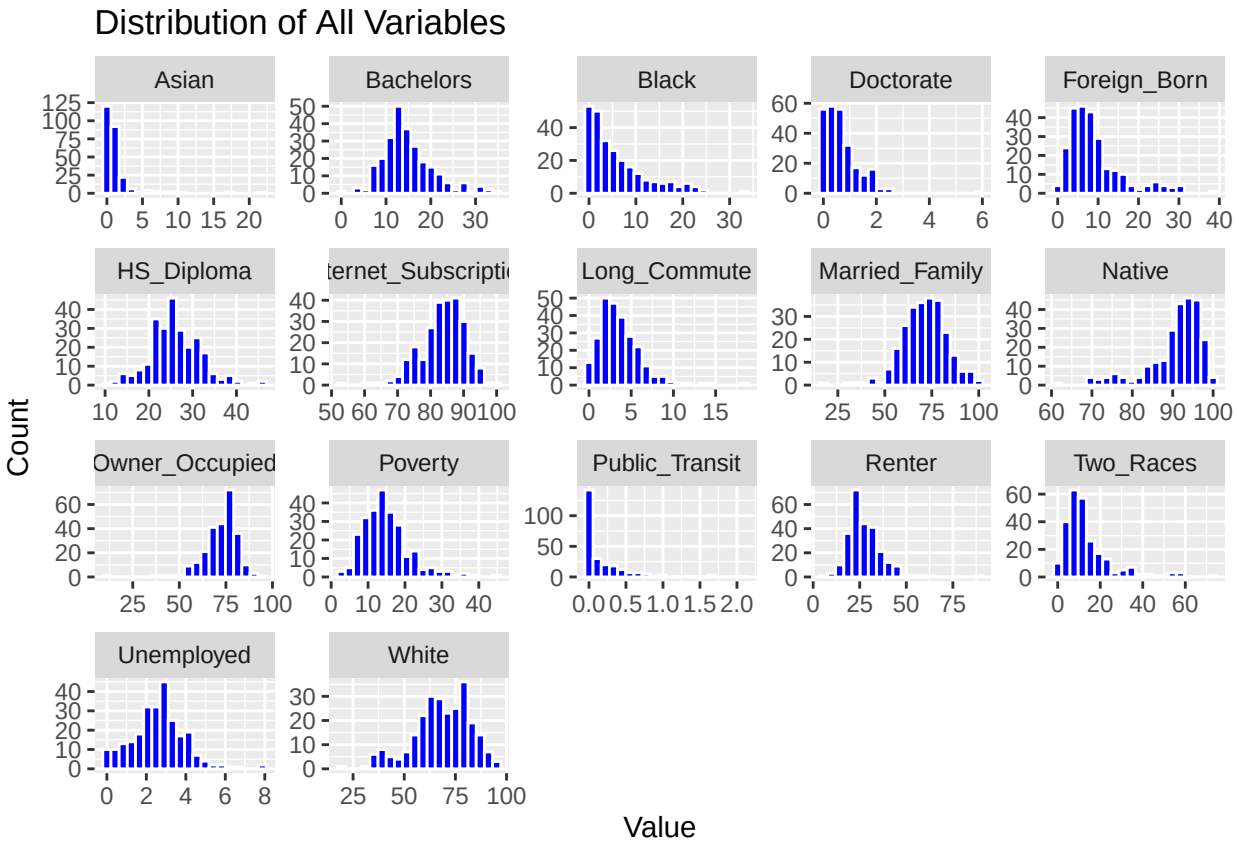
Characteristic	**N = 254**[1]
White (%)	68.36 (13.98); 69.24 [60.89, 78.89]
Black (%)	6.06 (6.23); 4.02 [1.22, 8.34]
Asian (%)	1.23 (2.29); 0.63 [0.20, 1.24]
Two or More Races (%)	13.63 (11.61); 10.19 [6.82, 16.62]
High School Diploma (%)	26.07 (5.76); 25.58 [22.38, 29.97]
Bachelors Degree (%)	14.83 (5.72); 13.78 [11.22, 17.71]
Doctorate Degree (%)	0.67 (0.67); 0.51 [0.21, 0.94]
Owner-Occupied Housing (%)	72.53 (9.25); 74.47 [67.59, 78.43]
Renter-Occupied Housing (%)	27.47 (9.25); 25.53 [21.57, 32.41]
Native Population (%)	90.79 (6.81); 92.54 [89.03, 95.32]
Foreign-Born Population (%)	9.21 (6.81); 7.46 [4.68, 10.97]
Below Poverty Line (%)	14.93 (6.39); 14.20 [10.63, 17.68]
Public Transit Use (%)	0.19 (0.34); 0.02 [0.00, 0.24]
Long Commute (90+ min) (%)	3.64 (2.53); 3.17 [1.95, 4.82]
Married-Couple Family (%)	71.16 (11.79); 71.59 [64.34, 78.22]
Unknown	2
Unemployment Rate (%)	2.62 (1.34); 2.65 [1.83, 3.37]
Home Internet Subscription (%)	83.61 (6.76); 84.34 [80.27, 87.81]

[1] Mean (SD); Median [Q1, Q3]



```
# Histogram Plot of Everything
data %>%
  select(-NAME) %>%
  pivot_longer(everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 20, fill = "blue", color = "white") +
  facet_wrap(~ variable, scales = "free") +
  labs(title = "Distribution of All Variables", x = "Value", y = "Count")
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

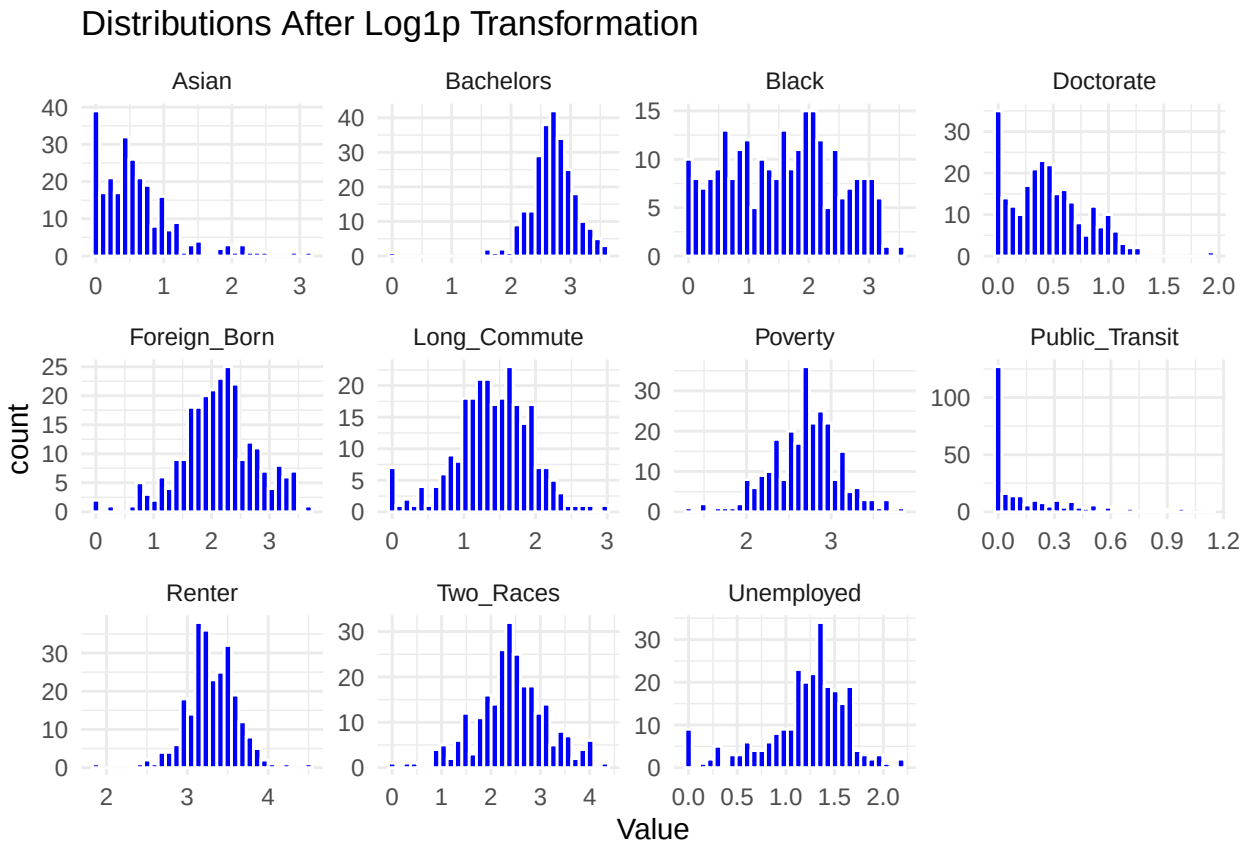



```
# Have to perform Log Transform on some variables
skewed_vars <- c(
  "Black", "Asian", "Two_Races", "Foreign_Born", "Public_Transit",
  "Poverty", "Long_Commute", "Renter", "Unemployed",
  "Doctorate", "Bachelors"
)

# Maps Set-Up
# Extract Shape File
tx_shapes <- st_read("tl_rd22_us_county/tl_rd22_us_county.shp")
```

```
## Reading layer `tl_rd22_us_county' from data source
##   `/Users/martinaradoslavov/Library/CloudStorage/OneDrive-WeillCornellMedicine/DataScienceI/Final Pr
##   using driver `ESRI Shapefile'
## Simple feature collection with 3234 features and 17 fields
```

```
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -179.2311 ymin: -14.60181 xmax: 179.8597 ymax: 71.43979
## Geodetic CRS:  NAD83
```

```r
# Read Texas Only
tx_shapes <- tx_shapes %>%
  filter(STATEFP == "48")

# Join to acs_clean
map_data <- tx_shapes %>%
  left_join(acs_clean, by = "GEOID")

# Log Transform
skewed_vars <- c(
  "Black", "Asian", "Two_Races", "Foreign_Born", "Public_Transit",
  "Poverty", "Long_Commute", "Renter", "Unemployed",
  "Doctorate", "Bachelors"
)

data_transformed <- data %>%
  mutate(across(all_of(skewed_vars), ~log1p(.x)))

data_transformed %>%
  select(all_of(skewed_vars)) %>%
  pivot_longer(everything(), names_to = "Variable", values_to = "Value") %>%
  ggplot(aes(x = Value)) +
  geom_histogram(bins = 30, fill = "blue", color = "white") +
  facet_wrap(~ Variable, scales = "free", ncol = 4) +
  theme_minimal() +
  labs(title = "Distributions After Log1p Transformation")
```




```
clean_data <- data_transformed %>%
  select(-NAME) %>%
  mutate(across(everything(), as.numeric))


# PCA
# Check missing values (excluding NAME)
colSums(is.na(data %>% select(-NAME)))
```

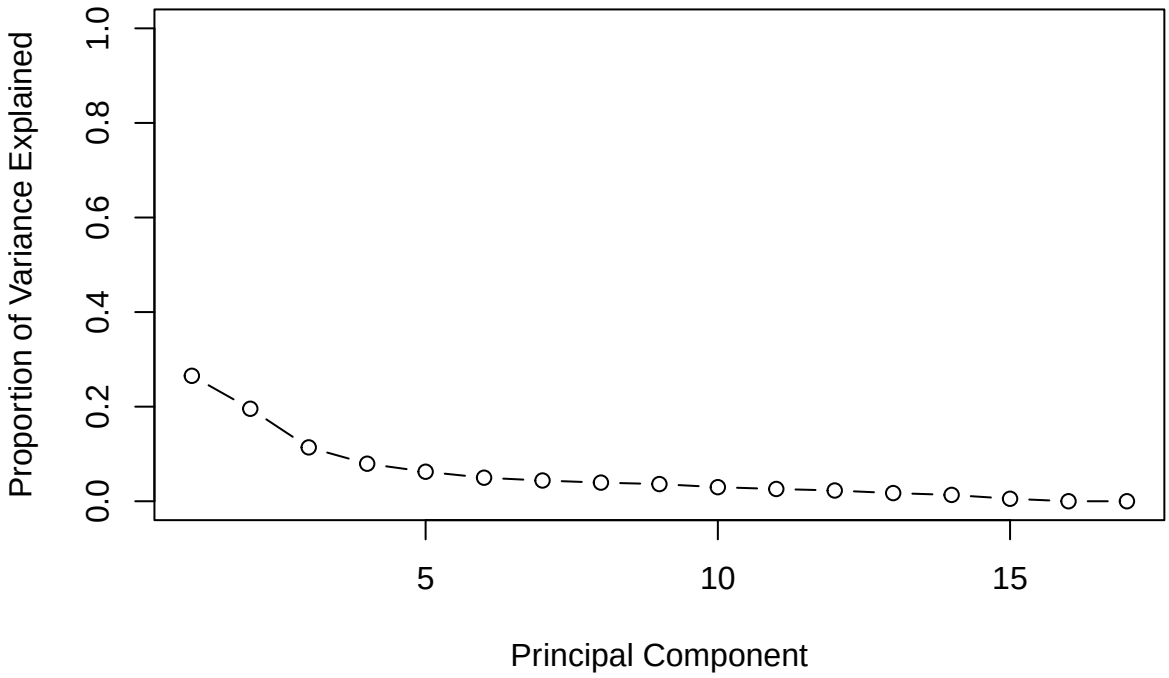
```
##                White                Black                Asian
##                    0                    0                    0
##            Two_Races            HS_Diploma            Bachelors
##                    0                    0                    0
##            Doctorate        Owner_Occupied               Renter
##                    0                    0                    0
##               Native          Foreign_Born              Poverty
##                    0                    0                    0
##        Public_Transit         Long_Commute       Married_Family
##                    0                    0                    2
##           Unemployed Internet_Subscription
##                    0                    0
```

```
# Remove NAs and scale for PCA
clean_data <- data %>%
  select(-NAME) %>%
  na.omit()
```

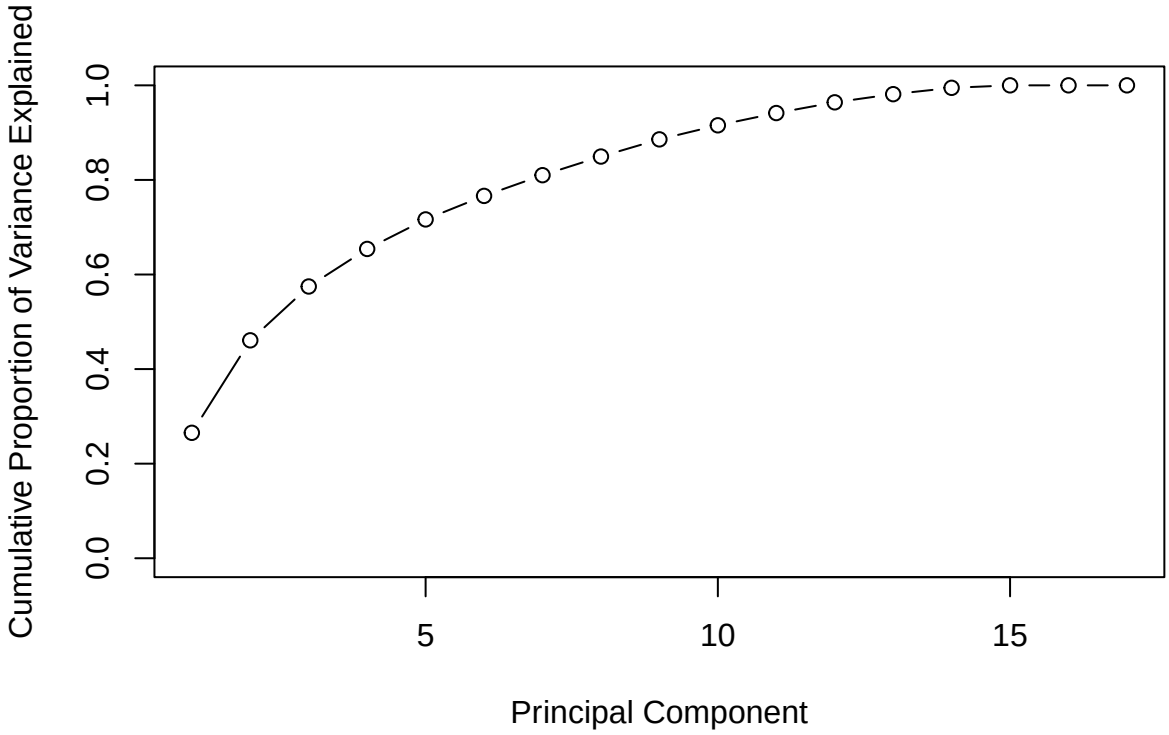
```r
# Run PCA
pca_result <- prcomp(clean_data, center = TRUE, scale. = TRUE)

# Convert PCA scores to dataframe
pc_scores <- as.data.frame(pca_result$x)

# PCA Summary
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.1235 1.8233 1.3915 1.16150 1.03015 0.91980 0.86320
## Proportion of Variance 0.2652 0.1956 0.1139 0.07936 0.06242 0.04977 0.04383
## Cumulative Proportion  0.2652 0.4608 0.5747 0.65404 0.71647 0.76623 0.81006
##                           PC8     PC9    PC10    PC11   PC12    PC13    PC14
## Standard deviation     0.81827 0.78561 0.71121 0.66364 0.6212 0.54151 0.47695
## Proportion of Variance 0.03939 0.03631 0.02975 0.02591 0.0227 0.01725 0.01338
## Cumulative Proportion  0.84945 0.88575 0.91551 0.94141 0.9641 0.98136 0.99475
##                           PC15      PC16      PC17
## Standard deviation     0.29888 9.801e-15 5.469e-16
## Proportion of Variance 0.00525 0.000e+00 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00 1.000e+00
```

```r
# Proportion of Variance Explained (PVE)
explained_variance <- summary(pca_result)$importance[2,]

# PVE Plot
plot(explained_variance, type = "b",
     xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1))
```

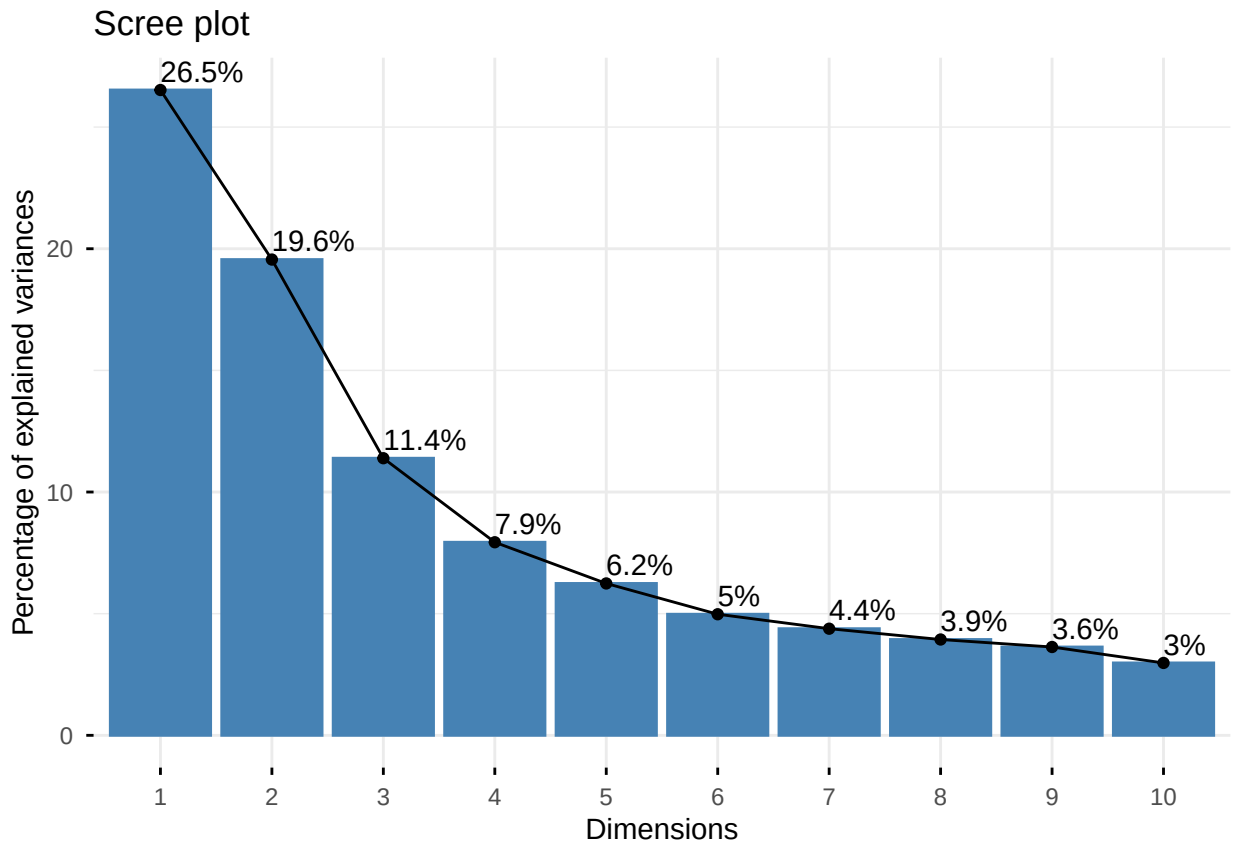



```
# Cumulative PVE
plot(cumsum(explained_variance), type = "b",
     xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained",
     ylim = c(0, 1))
```



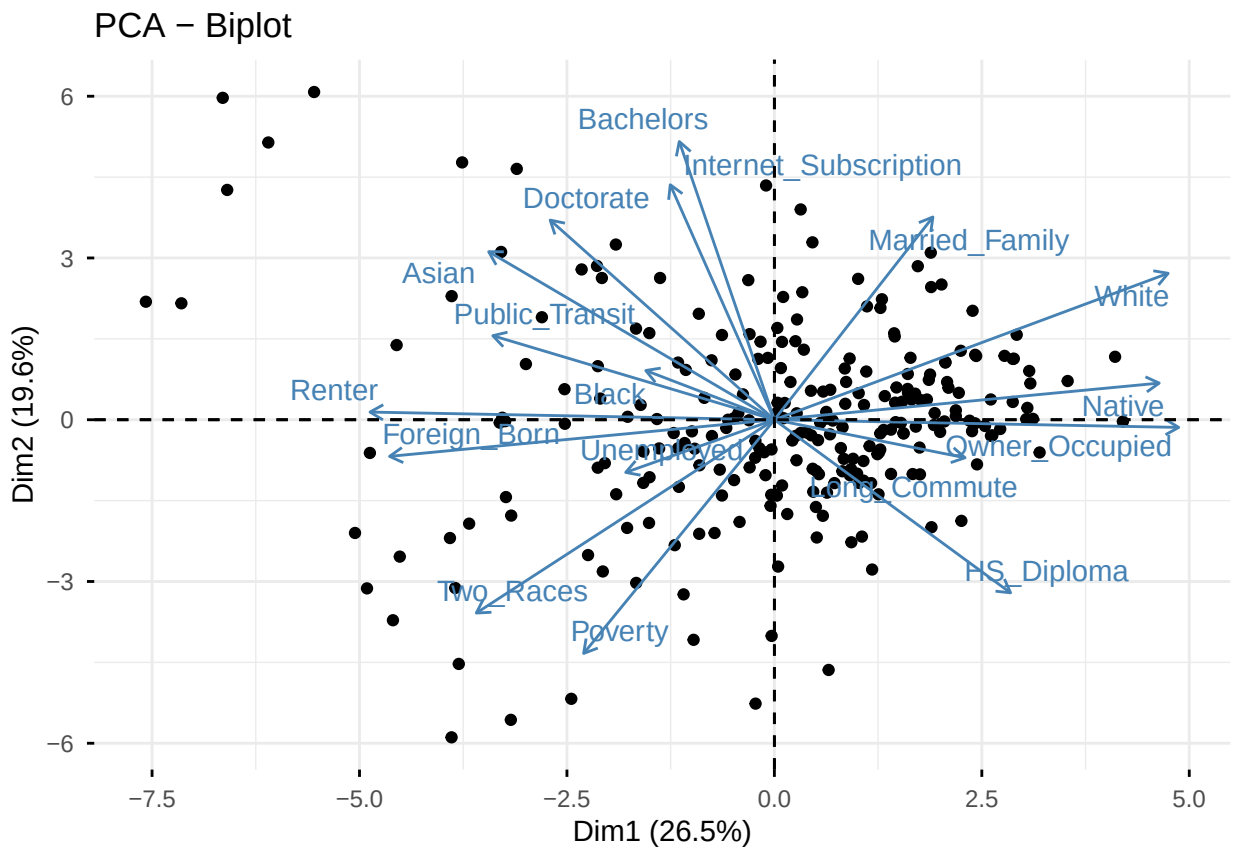

```
# Scree Plot
fviz_eig(pca_result, addlabels = TRUE)
```

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

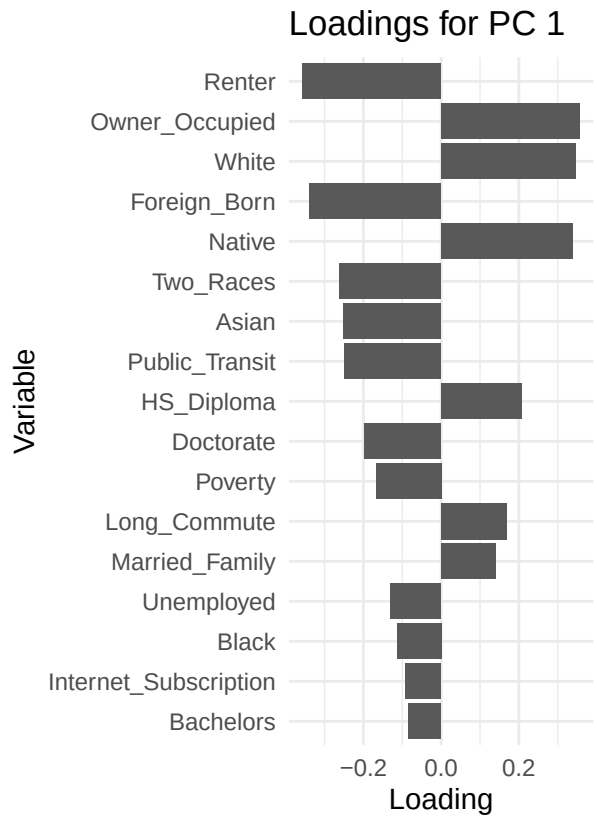


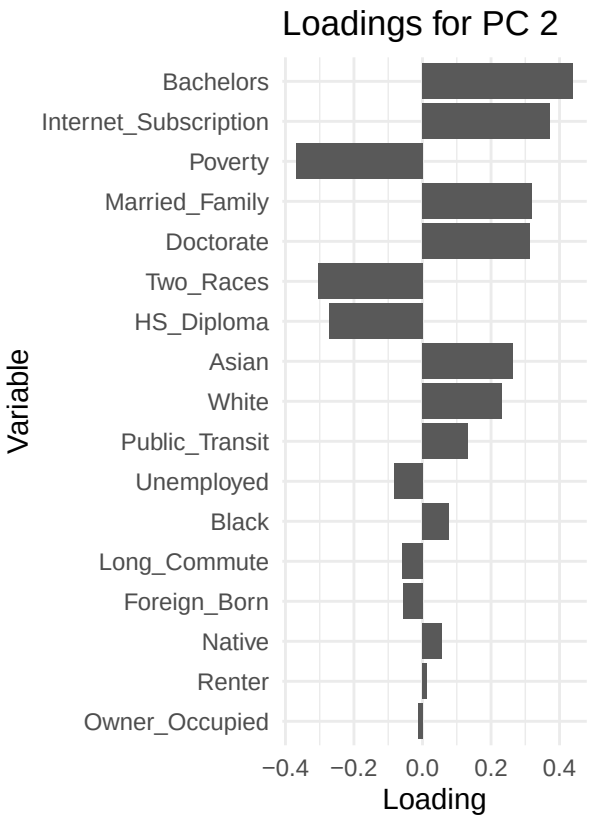

```
# PCA Biplot
fviz_pca_biplot(pca_result, geom.ind = "point", repel = TRUE)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the ggpubr package.
##   Please report the issue at <https://github.com/kassambara/ggpubr/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

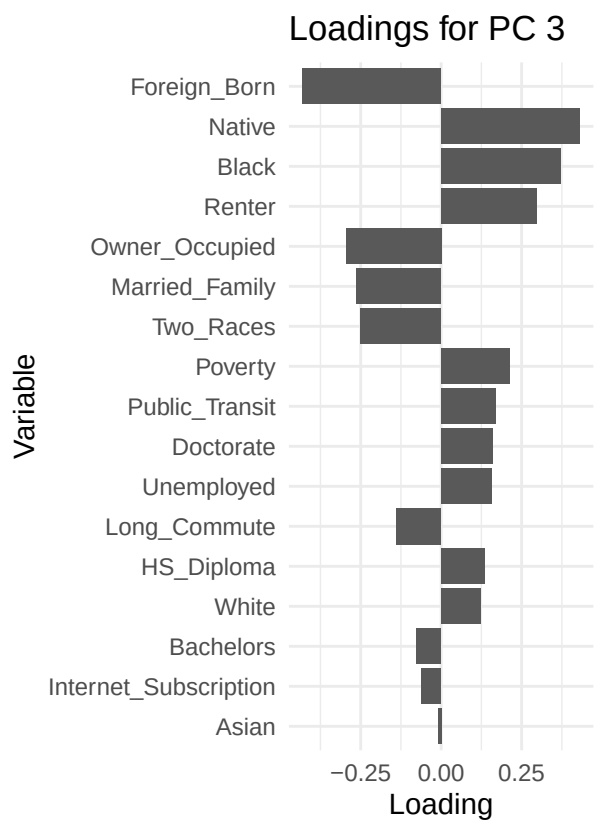


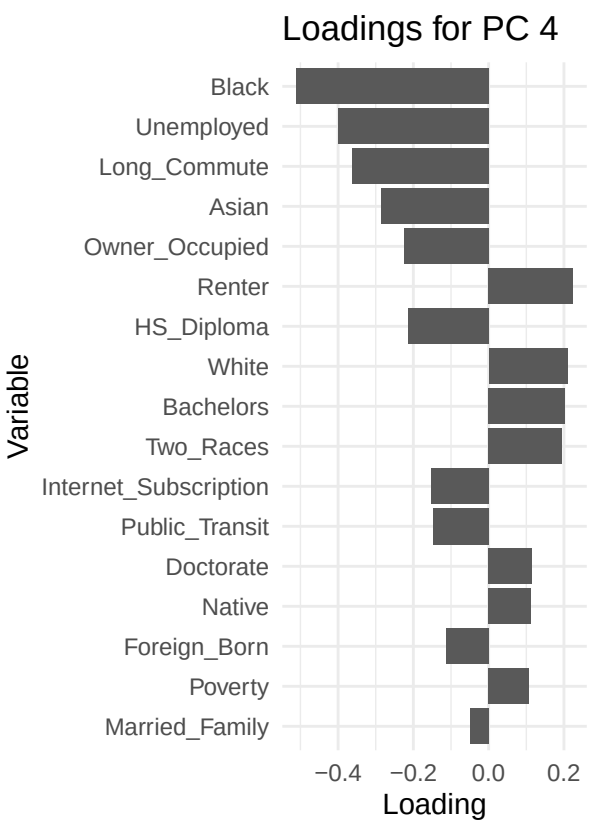

```r
# Loading Plots
plot_loading <- function(loading_vector, pc_number) {
  df <- data.frame(
    variable = names(loading_vector),
    loading  = as.numeric(loading_vector)
  )

  ggplot(df, aes(
    x = reorder(variable, abs(loading)),
    y = loading
  )) +
    geom_col() +
    coord_flip() +
    labs(
      title = paste("Loadings for PC", pc_number),
      x = "Variable",
      y = "Loading"
    ) +
    theme_minimal(base_size = 11)
}

plots <- lapply(1:5, function(i) {
  plot_loading(pca_result$rotation[, i], i)
})

plots[[1]] | plots[[2]]
```



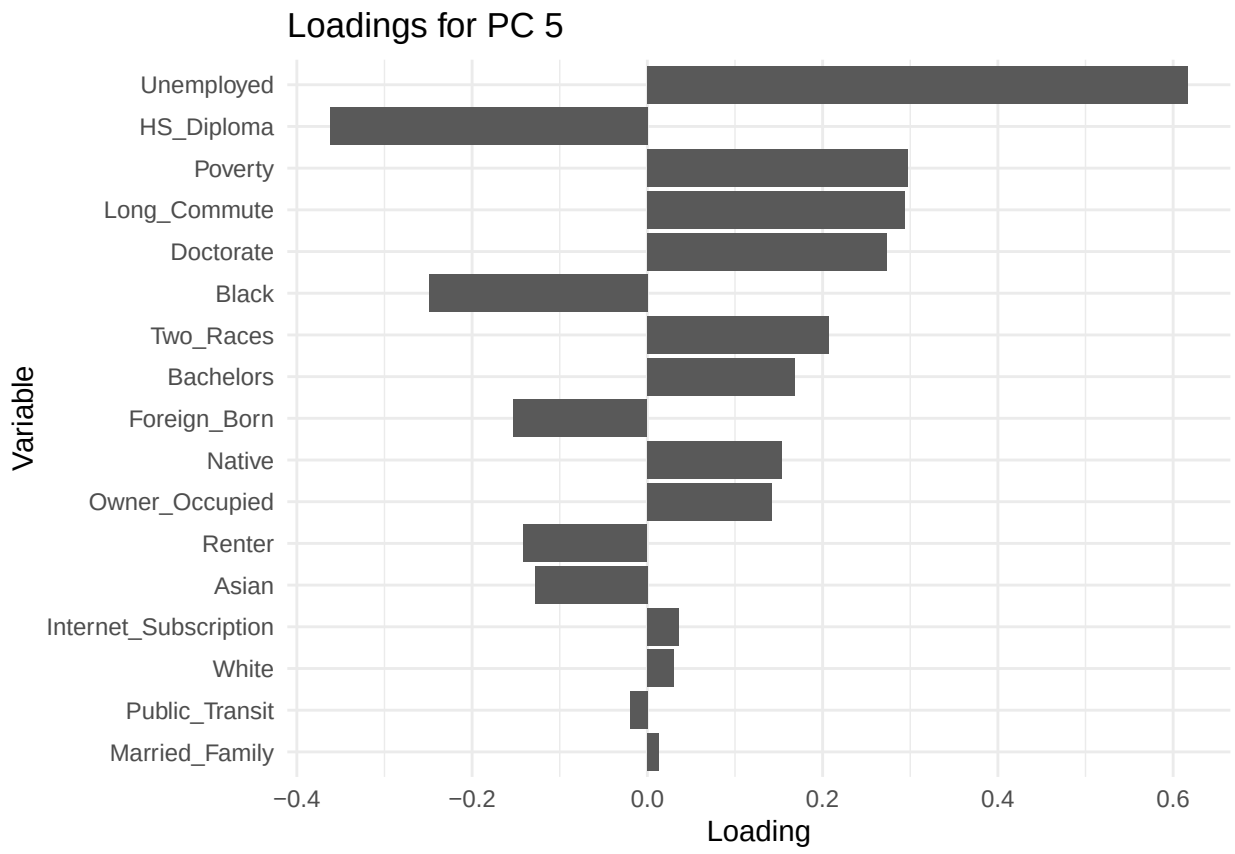



```
plots[[3]] | plots[[4]]
```


Loadings for PC 3
Foreign_Born
Native
Black
Renter
Owner_Occupied
Married_Family
Two_Races
Poverty
Public_Transit
Doctorate
Unemployed
Long_Commute
HS_Diploma
White
Bachelors
Internet_Subscription
Asian
Variable
−0.25
0.00
0.25
Loading

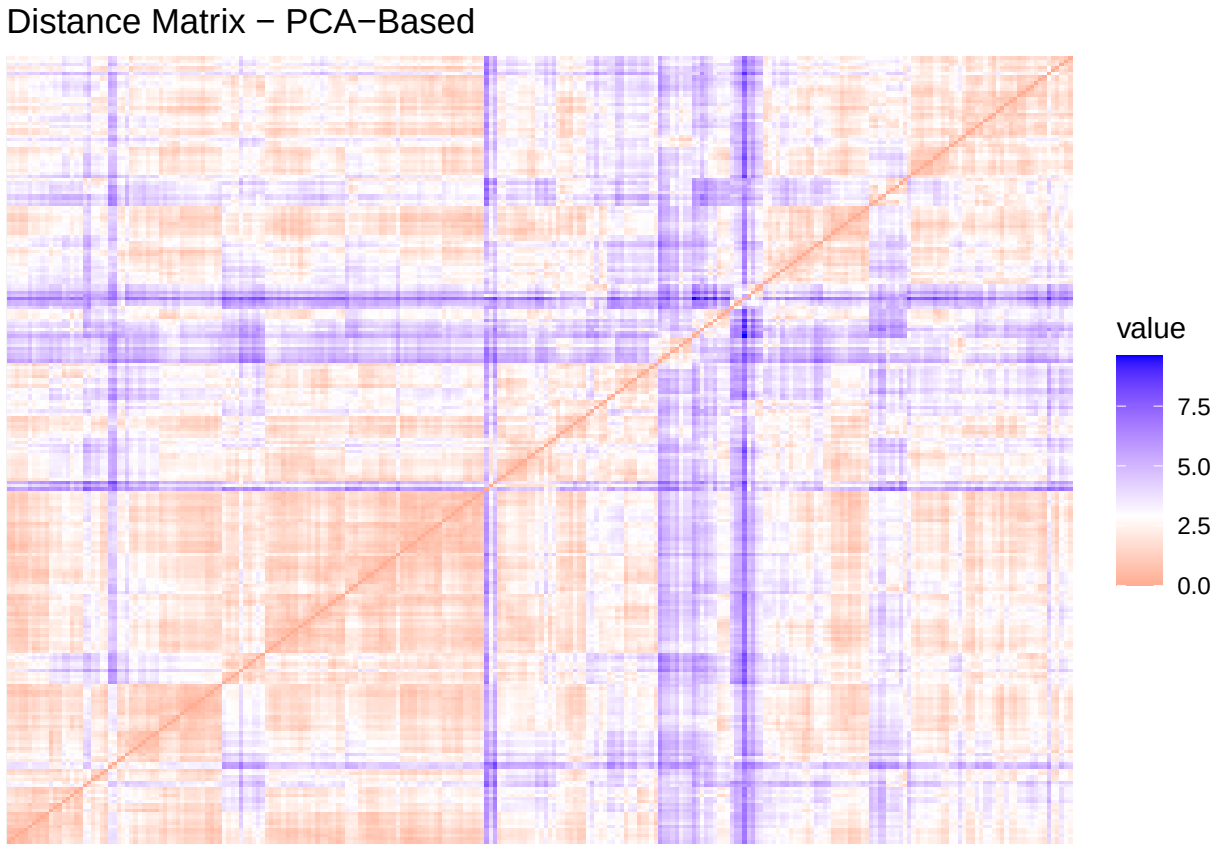

Loadings for PC 4
Black
Unemployed
Long_Commute
Asian
Owner_Occupied
Renter
HS_Diploma
White
Bachelors
Two_Races
Internet_Subscription
Public_Transit
Doctorate
Native
Foreign_Born
Poverty
Married_Family
Variable
−0.4
−0.2
0.0
0.2
Loading

```
plots[[5]]
```




```
# Clustering
# Use first 5 PCs
pc_for_clustering <- pc_scores[, 1:5]

# Scale PC scores for clustering
pc_for_clustering <- scale(pc_for_clustering)

# Distance visualization using PCs
fviz_dist(dist(pc_for_clustering), show_labels = FALSE) +
  labs(title = "Distance Matrix - PCA-Based")
```

```
## 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 factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# Compare Clustering Methods
set.seed(2025)
cl_methods <- c("hierarchical", "kmeans", "pam", "model")

internal_valid <- clValid(
  pc_for_clustering,
  nClust = 2:6,
  clMethods = cl_methods,
  validation = "internal"
)
```

```
## Warning in clValid(pc_for_clustering, nClust = 2:6, clMethods = cl_methods, :
## rownames for data not specified, using 1:nrow(data)
```

```
summary(internal_valid)
```

```
##
## Clustering Methods:
##  hierarchical kmeans pam model
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                                  2        3        4        5        6
##
## hierarchical Connectivity    5.9829  16.0881  20.0571  20.0571  25.9560
```

```
##              Dunn            0.2253   0.2195   0.2195   0.2195   0.1604
##              Silhouette      0.4505   0.4195   0.3391   0.3126   0.2920
## kmeans       Connectivity   45.1298 104.4512 150.3274 149.4714 154.7258
##              Dunn            0.0724   0.0463   0.0368   0.0480   0.0523
##              Silhouette      0.2446   0.1921   0.1315   0.1866   0.1887
## pam          Connectivity  102.5063 168.5917 178.9179 204.2361 187.3135
##              Dunn            0.0401   0.0394   0.0457   0.0481   0.0493
##              Silhouette      0.1439   0.1125   0.1384   0.1400   0.1644
## model        Connectivity   87.8139 175.5762 162.4052 102.5337 175.2548
##              Dunn            0.0292   0.0445   0.0438   0.0691   0.0699
##              Silhouette      0.2161   0.0641  -0.0086   0.2301   0.1213
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 5.9829 hierarchical 2
## Dunn         0.2253 hierarchical 2
## Silhouette   0.4505 hierarchical 2
```

```
stab_valid <- clValid(
  pc_for_clustering,
  nClust = 2:6,
  clMethods = cl_methods,
  validation = "stability"
)
```

```
## Warning in clValid(pc_for_clustering, nClust = 2:6, clMethods = cl_methods, :
## rownames for data not specified, using 1:nrow(data)
```

```
optimalScores(stab_valid)
```

```
##          Score       Method Clusters
## APN 0.01331429 hierarchical        2
## AD  2.43394441       kmeans        6
## ADM 0.09934420 hierarchical        2
## FOM 0.96994612        model        6
```

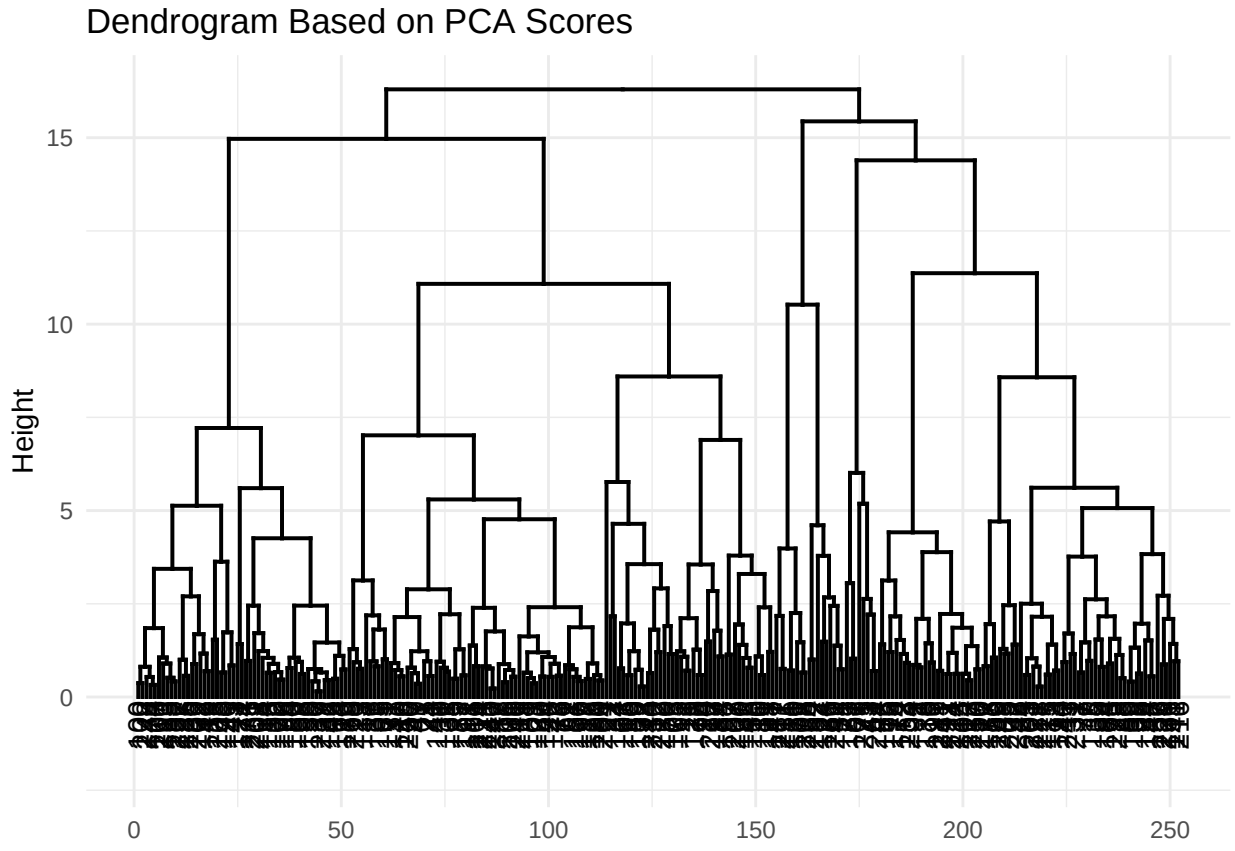
```
# Use the PCA data for clustering
Xsc <- scale(pc_for_clustering)

set.seed(2025)

# Methods to compare
m <- c("average", "single", "complete", "ward")
names(m) <- m

# Function to compute agglomerative coefficient
ac <- function(method) {
  agnes(Xsc, method = method)$ac
}

# Compute agglomerative coefficients for each method
agnes_results <- purrr::map_dbl(m, ac)
agnes_results
```

```
##   average    single  complete      ward
```

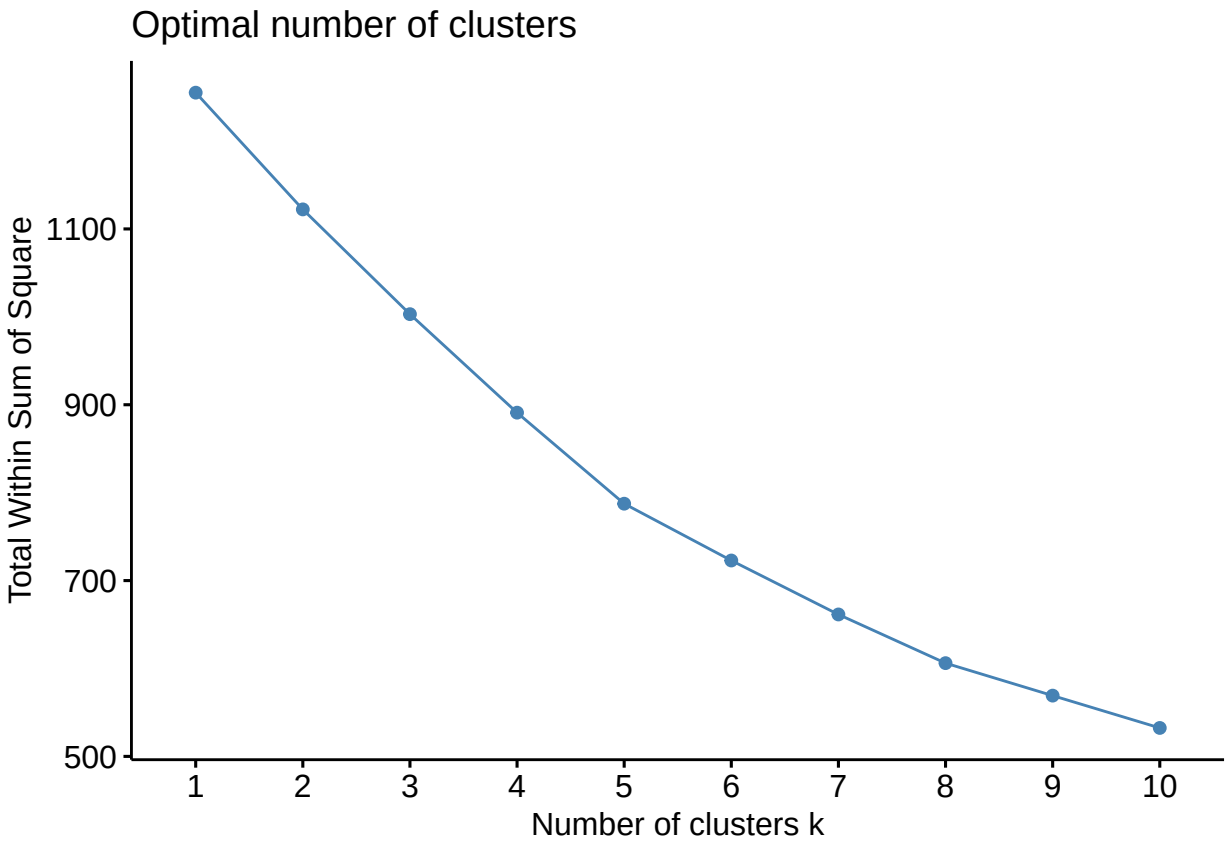
```
## 0.8287190 0.7035395 0.9045182 0.9428435
```

```r
# Hierarchical Clustering
set.seed(2025)

# Compute distance on PCs
dist_pc <- dist(pc_for_clustering, method = "euclidean")

# Hierarchical clustering (Ward's method)
hc_complete <- hclust(dist_pc, method = 'ward.D2')

# Plot Dendrogram
fviz_dend(hc_complete) +
  theme_minimal() +
  labs(title = "Dendrogram Based on PCA Scores")
```

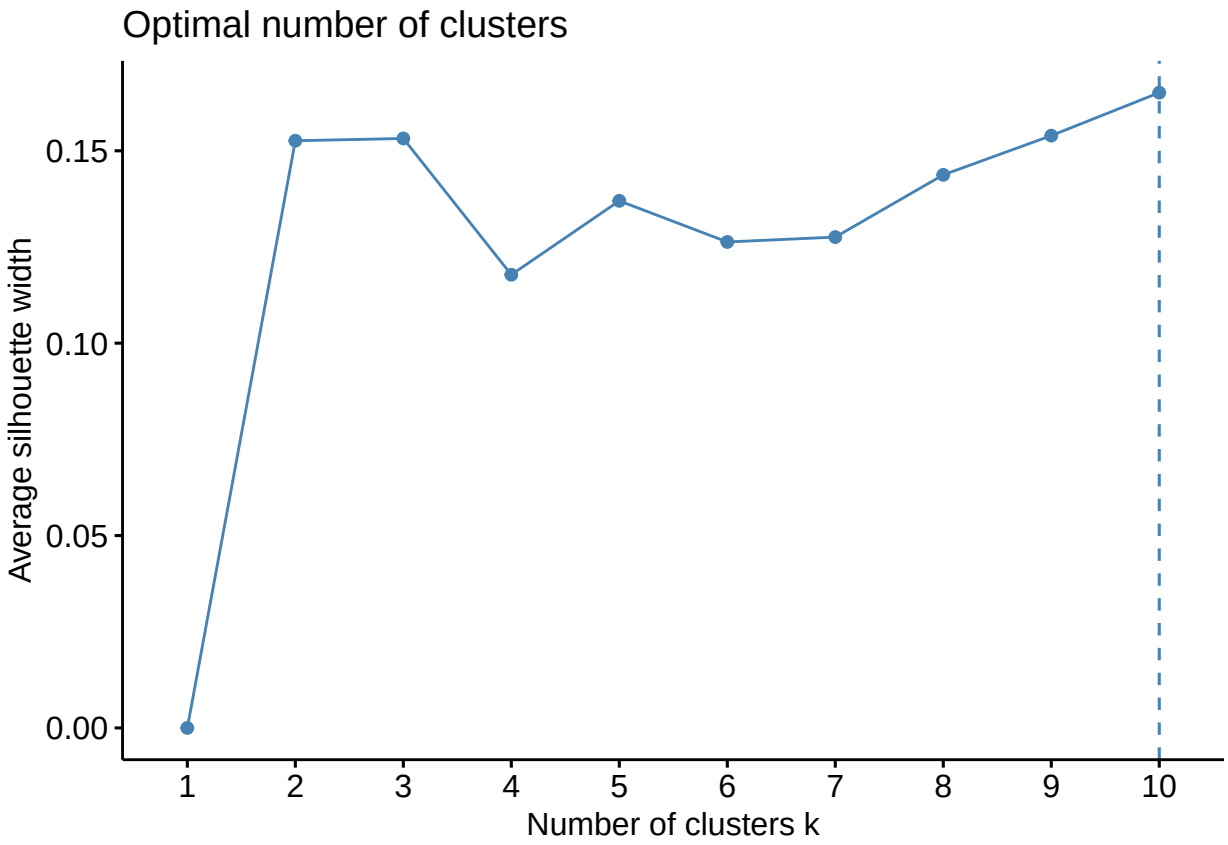
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



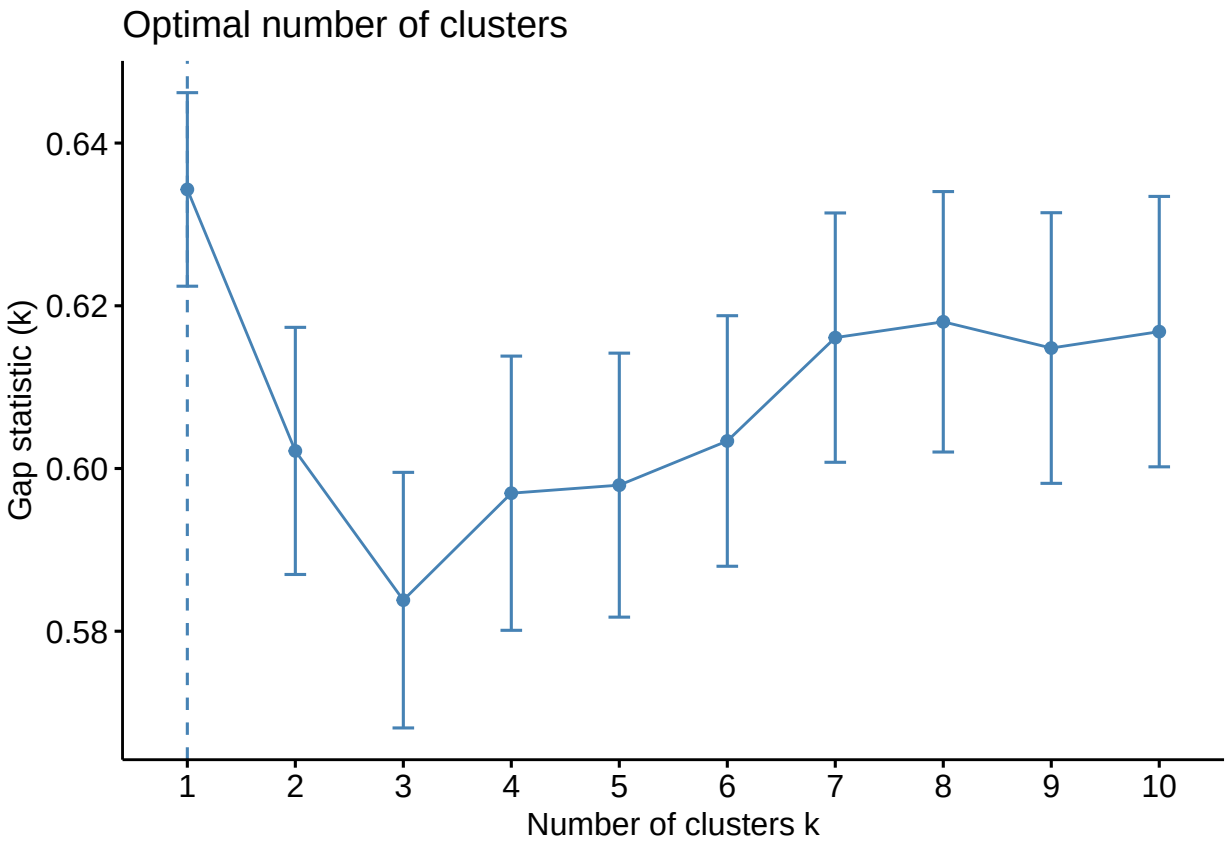

```r
# WSS, Silhouette, Gap Statistic
set.seed(2025)
fviz_nbclust(pc_for_clustering, FUN = hcut, method = "wss", k.max = 10)
```



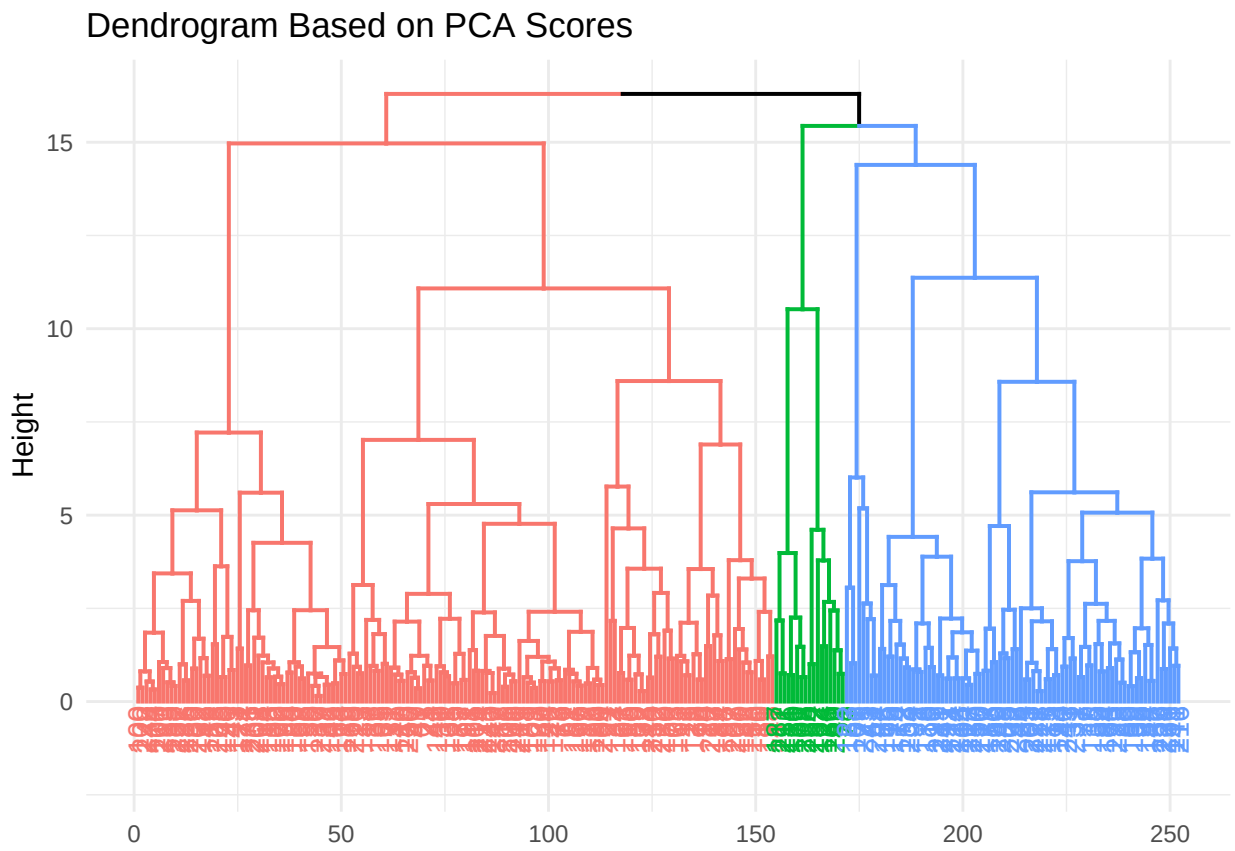

```
fviz_nbclust(pc_for_clustering, FUN = hcut, method = "silhouette", k.max = 10)
```




```
fviz_nbclust(pc_for_clustering, FUN = hcut, method = "gap_stat", k.max = 10)
```

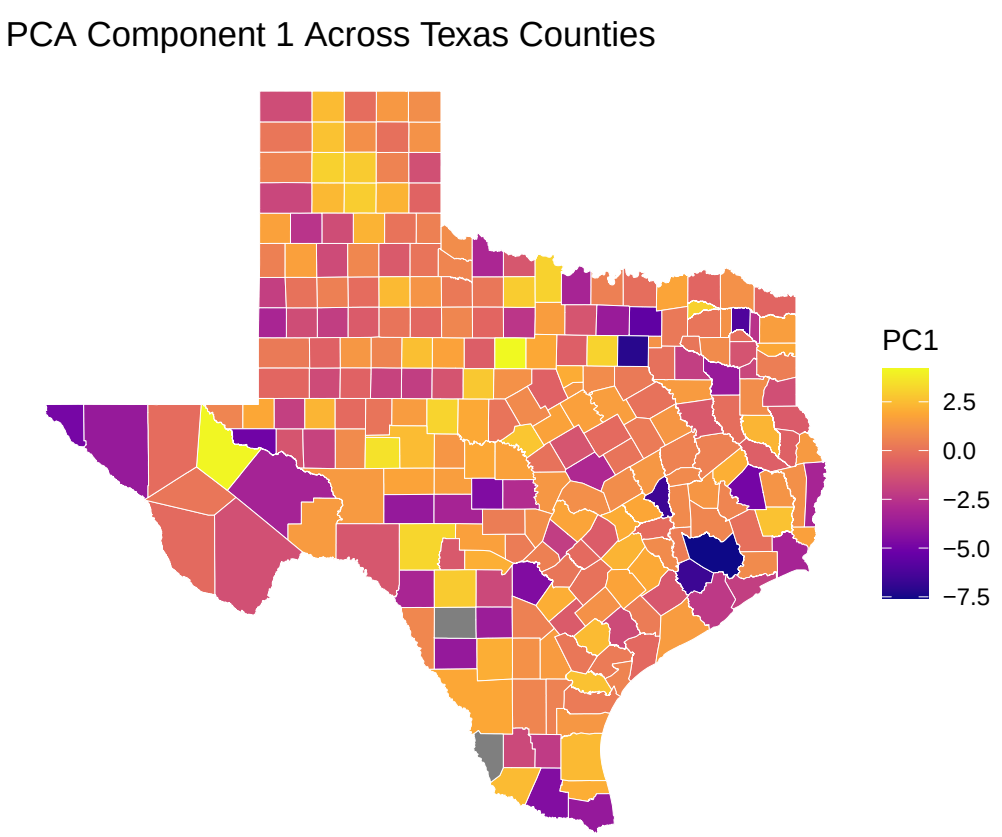



```
# Cut into the number of clusters
clusters_pca <- cutree(hc_complete, k = 3)
fviz_dend(hc_complete, k = 3) +
  theme_minimal() +
  labs(title = "Dendrogram Based on PCA Scores")
```

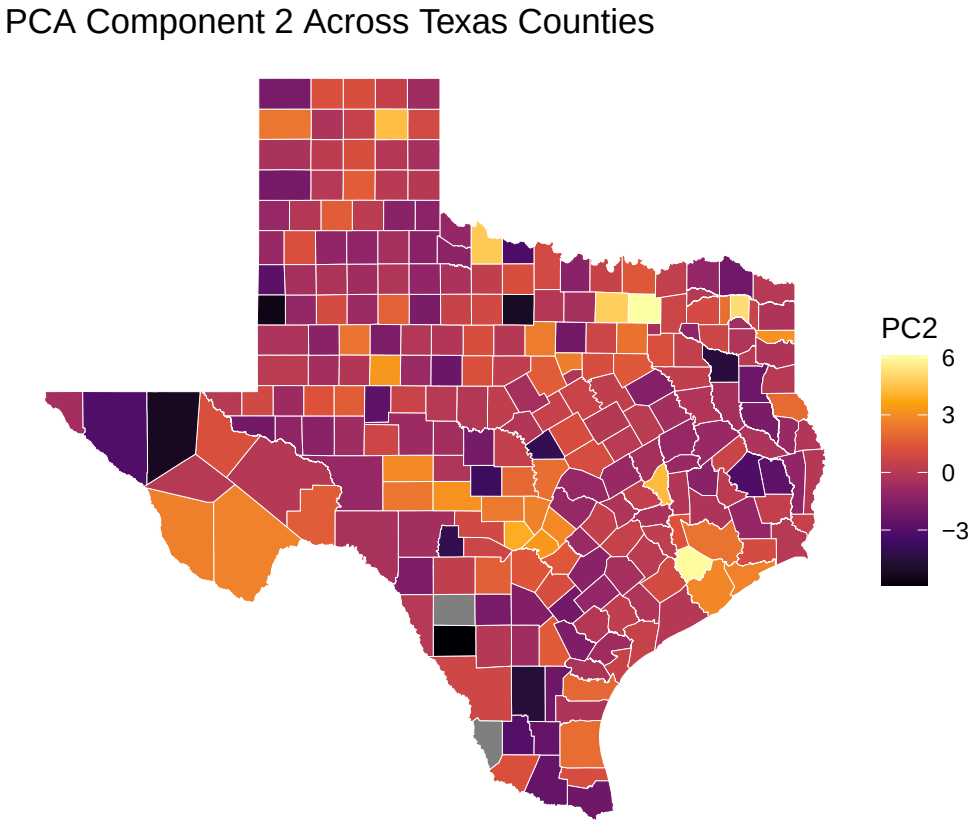



```
# Maps for PCA and Clustering
# PCA
rows_used <- rownames(clean_data)

# Create full PC vectors for mapping
PC1_full <- rep(NA, nrow(acs_clean))
PC2_full <- rep(NA, nrow(acs_clean))

PC1_full[rownames(acs_clean) %in% rows_used] <- pc_scores$PC1
PC2_full[rownames(acs_clean) %in% rows_used] <- pc_scores$PC2

pca_df <- acs_clean %>%
  mutate(
    PC1 = PC1_full,
    PC2 = PC2_full
  )

# Join to shapefile
map_data_pca <- tx_shapes %>%
  left_join(pca_df, by = "GEOID")

# Map PC1
ggplot(map_data_pca) +
  geom_sf(aes(fill = PC1), color = "white", size = 0.15) +
  scale_fill_viridis_c(option = "plasma") +
  labs(title = "PCA Component 1 Across Texas Counties", fill = "PC1") +
```

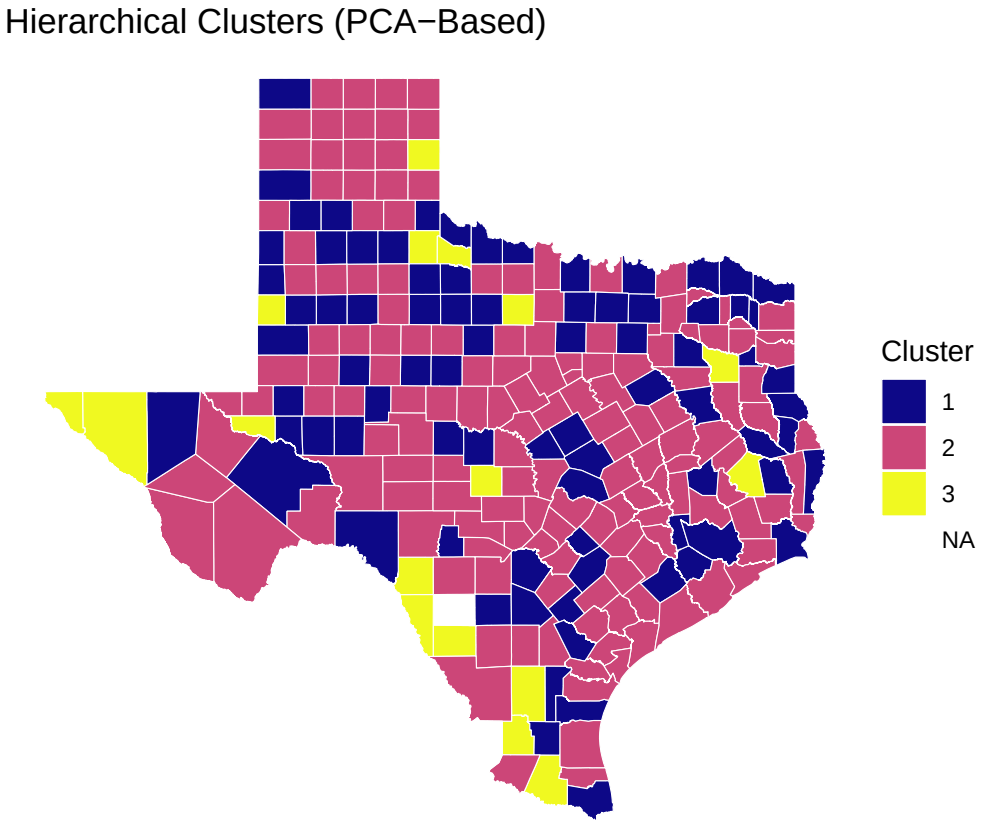
```
  theme_void()
```




```
# Map PC2
ggplot(map_data_pca) +
  geom_sf(aes(fill = PC2), color = "white", size = 0.15) +
  scale_fill_viridis_c(option = "inferno") +
  labs(title = "PCA Component 2 Across Texas Counties", fill = "PC2") +
  theme_void()
```




```
# Clustering Map
# Full cluster vector
cluster_full <- rep(NA, nrow(acs_clean))
cluster_full[rownames(acs_clean) %in% rows_used] <- clusters_pca

cluster_df <- acs_clean %>%
  mutate(cluster = cluster_full)

map_data_cluster <- tx_shapes %>%
  left_join(cluster_df, by = "GEOID")

# Plot clusters
ggplot(map_data_cluster) +
  geom_sf(aes(fill = factor(cluster)), color = "white") +
  scale_fill_viridis_d(option = "plasma") +
  labs(title = "Hierarchical Clusters (PCA-Based)", fill = "Cluster") +
  theme_void()
```


Hierarchical Clusters (PCA−Based)
Cluster
1
2
3
NA