OSNAHW4

2025-10-29

```r
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```r
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(ggraph)
```

```
## Loading required package: ggplot2
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v lubridate 1.9.4     v tibble    3.3.0
## v purrr     1.1.0     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::%--%()      masks igraph::%--%()
## x dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## x purrr::compose()       masks igraph::compose()
## x tidyr::crossing()      masks igraph::crossing()
## x dplyr::filter()        masks stats::filter()
## x dplyr::lag()           masks stats::lag()
## x purrr::simplify()      masks igraph::simplify()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(tidycensus)
library(sf)
```

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

```r
library(tigris)
```

```
## To enable caching of data, set `options(tigris_use_cache = TRUE)`
## in your R script or .Rprofile.
##
## Attaching package: 'tigris'
##
## The following object is masked from 'package:igraph':
##
##     blocks
```

```r
library(ggplot2)
library(dplyr)
library(viridis)
```

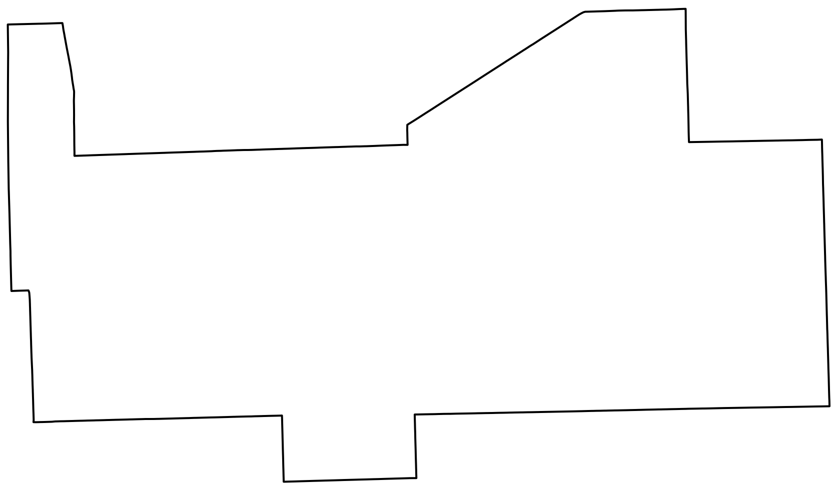
```
## Loading required package: viridisLite
```

```r
library(patchwork)
library(ggcorrplot)

# Load community area boundaries and filter for Dunning
ca_sf <- st_read("/Users/mohammedfawwazuddin/Desktop/dunning_ca/data/raw/community_areas/Boundaries - Co
```

```
## Reading layer `geo_export_b4e7b613-ef4c-47a9-b31c-b51bd1c24b7a' from data source `/Users/mohammedfaww
##   using driver `ESRI Shapefile'
## Simple feature collection with 77 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -87.94011 ymin: 41.64454 xmax: -87.52414 ymax: 42.02304
## Geodetic CRS:  WGS 84
```

```r
dunning_sf <- ca_sf %>% filter(community == "DUNNING" | area_num_1 == 17)
plot(st_geometry(dunning_sf), main = "Dunning Community Area")
```

Dunning Community Area



```
# Tract mapping from official Chicago data
ca_tract_mapping <- data.frame(
  tract = c("170100", "170200", "170300", "170400", "170500",
            "170600", "170700", "170800", "170900", "171000", "171100"),
  community_area = "Dunning",
  area_num_1 = 17
) %>%
  mutate(
    GEOID_tract = paste0("17031", tract),
    GEOID_bg_prefix = paste0("17031", tract)
  )

cat("Dunning Community Area contains", nrow(ca_tract_mapping), "census tracts\n")
```

```
## Dunning Community Area contains 11 census tracts
```

```
# Define census variables for analysis
my_vars_2020 <- c(
  total_pop = "P1_001N",
  race_white = "P2_003N",
  race_black = "P2_004N",
  hispanic = "P2_010N"
)

my_vars_2010 <- c(
```

```
  total_pop = "P001001",
  race_white = "P003002",
  race_black = "P003003",
  hispanic = "P004003"
)

# Get decennial census data for Cook County
cook_bg_2020 <- get_decennial(
  geography = "block group",
  variables = my_vars_2020,
  year = 2020,
  state = "IL",
  county = "Cook",
  geometry = TRUE,
  output = "wide"
)
```

```
## Getting data from the 2020 decennial Census
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future sessions
## Using the PL 94-171 Redistricting Data Summary File

##   |                                                                              |
```

```
cook_bg_2010 <- get_decennial(
  geography = "block group",
  variables = my_vars_2010,
  year = 2010,
  state = "IL",
  county = "Cook",
  geometry = TRUE,
  output = "wide"
)
```

```
## Getting data from the 2010 decennial Census
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future sessions
## Using Census Summary File 1

##   |                                                                              |
```

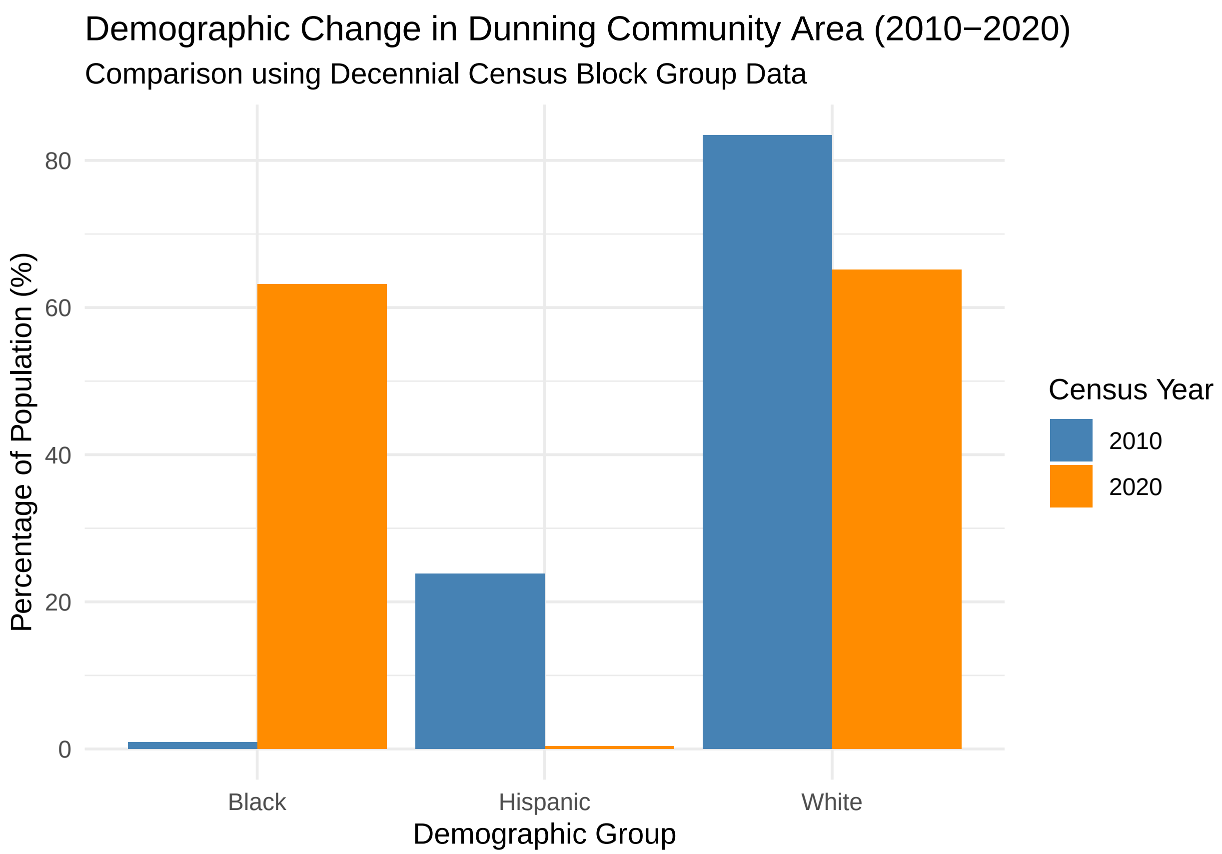
```
# Transform CRS to match Dunning boundaries
cook_bg_2020 <- st_transform(cook_bg_2020, st_crs(dunning_sf))
cook_bg_2010 <- st_transform(cook_bg_2010, st_crs(dunning_sf))

# Filter for Dunning block groups using tract mapping
dunning_bg_2020 <- cook_bg_2020 %>%
```

```r
  mutate(tract_code = substr(GEOID, 1, 11)) %>%
  filter(tract_code %in% ca_tract_mapping$GEOID_tract)

dunning_bg_2010 <- cook_bg_2010 %>%
  mutate(tract_code = substr(GEOID, 1, 11)) %>%
  filter(tract_code %in% ca_tract_mapping$GEOID_tract)

# Calculate summary statistics for Dunning
dunning_summary_2020 <- dunning_bg_2020 %>%
  st_drop_geometry() %>%
  summarize(
    total_pop = sum(total_pop, na.rm = TRUE),
    pct_white = (sum(race_white, na.rm = TRUE) / sum(total_pop, na.rm = TRUE)) * 100,
    pct_black = (sum(race_black, na.rm = TRUE) / sum(total_pop, na.rm = TRUE)) * 100,
    pct_hispanic = (sum(hispanic, na.rm = TRUE) / sum(total_pop, na.rm = TRUE)) * 100
  ) %>%
  mutate(year = "2020")

dunning_summary_2010 <- dunning_bg_2010 %>%
  st_drop_geometry() %>%
  summarize(
    total_pop = sum(total_pop, na.rm = TRUE),
    pct_white = (sum(race_white, na.rm = TRUE) / sum(total_pop, na.rm = TRUE)) * 100,
    pct_black = (sum(race_black, na.rm = TRUE) / sum(total_pop, na.rm = TRUE)) * 100,
    pct_hispanic = (sum(hispanic, na.rm = TRUE) / sum(total_pop, na.rm = TRUE)) * 100
  ) %>%
  mutate(year = "2010")

# 2010 vs 2020 demographics
dunning_comparison <- bind_rows(dunning_summary_2010, dunning_summary_2020)
print(dunning_comparison)
```

```
## # A tibble: 2 x 5
##   total_pop pct_white pct_black pct_hispanic year
##       <dbl>     <dbl>     <dbl>        <dbl> <chr>
## 1     41932      83.4     0.937       23.8   2010
## 2     43147      65.1    63.2          0.350 2020
```

```r
# Demographic comparison plot
dunning_plot_data <- dunning_comparison %>%
  select(year, pct_white, pct_black, pct_hispanic) %>%
  pivot_longer(cols = -year, names_to = "demographic", values_to = "percentage") %>%
  mutate(demographic = case_when(
    demographic == "pct_white" ~ "White",
    demographic == "pct_black" ~ "Black",
    demographic == "pct_hispanic" ~ "Hispanic"
  ))

ggplot(dunning_plot_data, aes(x = demographic, y = percentage, fill = year)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  scale_fill_manual(values = c("2010" = "steelblue", "2020" = "darkorange")) +
  labs(title = "Demographic Change in Dunning Community Area (2010-2020)",
       subtitle = "Comparison using Decennial Census Block Group Data",
```

```
      x = "Demographic Group",
      y = "Percentage of Population (%)",
      fill = "Census Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5))
```

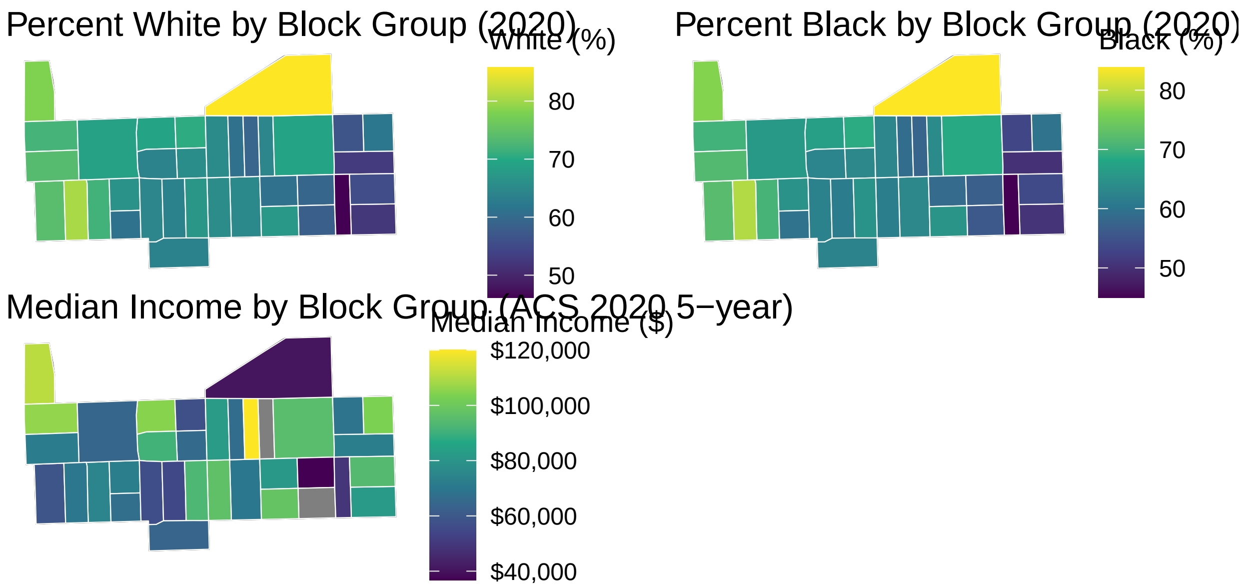



```
# Get income data from ACS to supplement decennial data
acs_income_2020 <- get_acs(
  geography = "block group",
  variables = c(med_income = "B19013_001"),
  year = 2020,
  state = "IL",
  county = "Cook",
  geometry = TRUE,
  output = "wide"
)
```

```
## Getting data from the 2016-2020 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future session
```

```
acs_income_2020 <- st_transform(acs_income_2020, st_crs(dunning_sf))

dunning_income_2020 <- acs_income_2020 %>%
  mutate(tract_code = substr(GEOID, 1, 11)) %>%
```

```r
  filter(tract_code %in% ca_tract_mapping$GEOID_tract)

# Join income data with demographic data for mapping
dunning_bg_2020_joined <- dunning_bg_2020 %>%
  left_join(st_drop_geometry(dunning_income_2020)[, c("GEOID", "med_incomeE")], by = "GEOID")

bg_plot_data <- dunning_bg_2020_joined %>%
  mutate(
    pct_white = (race_white / total_pop) * 100,
    pct_black = (race_black / total_pop) * 100,
    pct_hispanic = (hispanic / total_pop) * 100
  )

# Map showing white population distribution
p1 <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black") +
  geom_sf(data = bg_plot_data, aes(fill = pct_white), color = "white") +
  scale_fill_viridis_c(name = "White (%)") +
  labs(title = "Percent White by Block Group (2020)") +
  theme_void()

# Maps for black population
p2 <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black") +
  geom_sf(data = bg_plot_data, aes(fill = pct_black), color = "white") +
  scale_fill_viridis_c(name = "Black (%)") +
  labs(title = "Percent Black by Block Group (2020)") +
  theme_void()

#Map for income distribution
p3 <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black") +
  geom_sf(data = bg_plot_data, aes(fill = med_incomeE), color = "white") +
  scale_fill_viridis_c(name = "Median Income ($)", labels = scales::dollar) +
  labs(title = "Median Income by Block Group (ACS 2020 5-year)") +
  theme_void()

# Display all three maps together for comparison
p1 + p2 + p3 + plot_layout(ncol = 2)
```

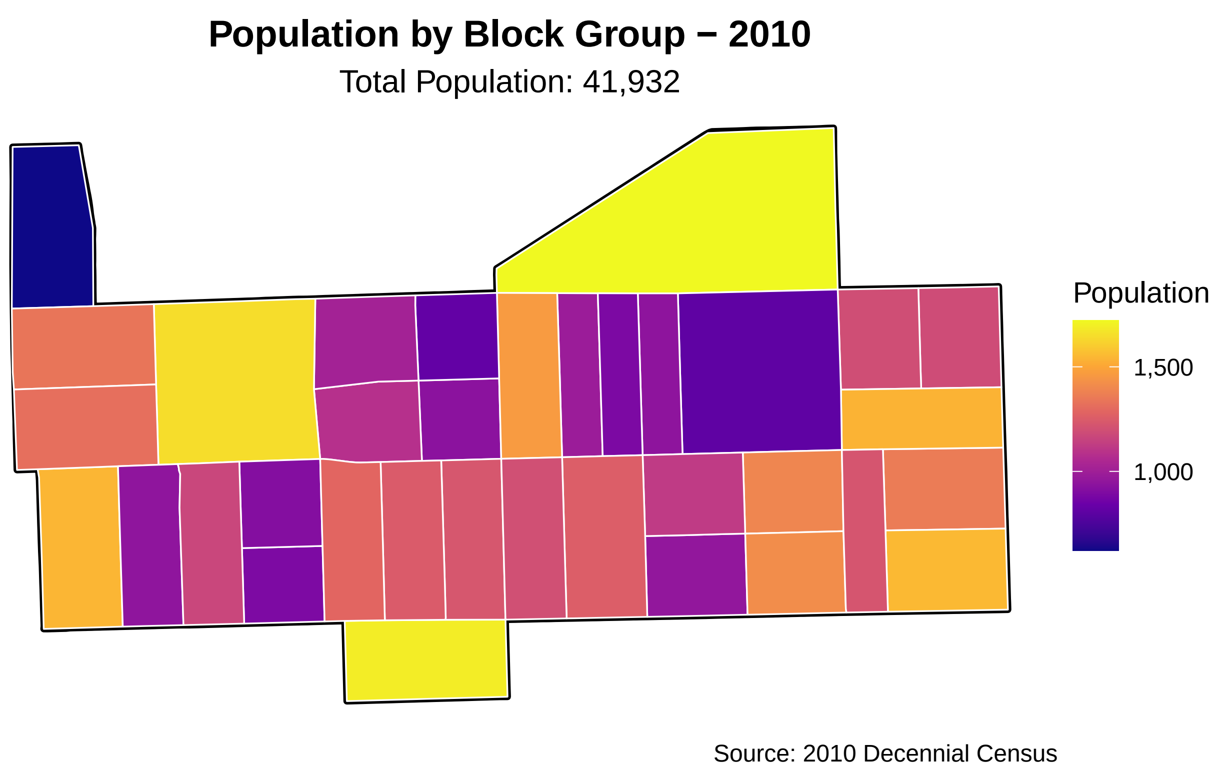



```r
# Prepare data for population comparison between 2010 and 2020
pop_2010_data <- dunning_bg_2010 %>% mutate(year = "2010")
pop_2020_data <- dunning_bg_2020 %>% mutate(year = "2020")

all_pop_data <- bind_rows(pop_2010_data, pop_2020_data)

# Population maps for both years
pop_map_2010 <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black", linewidth = 1.2) +
  geom_sf(data = pop_2010_data, aes(fill = total_pop), color = "white", linewidth = 0.3) +
  scale_fill_viridis_c(
    name = "Population",
    option = "plasma",
    breaks = c(500, 1000, 1500, 2000, 2500, 3000),
    labels = scales::comma,
    trans = "sqrt"
  ) +
  labs(title = "Population by Block Group - 2010",
       subtitle = paste("Total Population:", scales::comma(sum(pop_2010_data$total_pop, na.rm = TRUE)))
       caption = "Source: 2010 Decennial Census") +
  theme_void() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    legend.position = "right"
  )
```

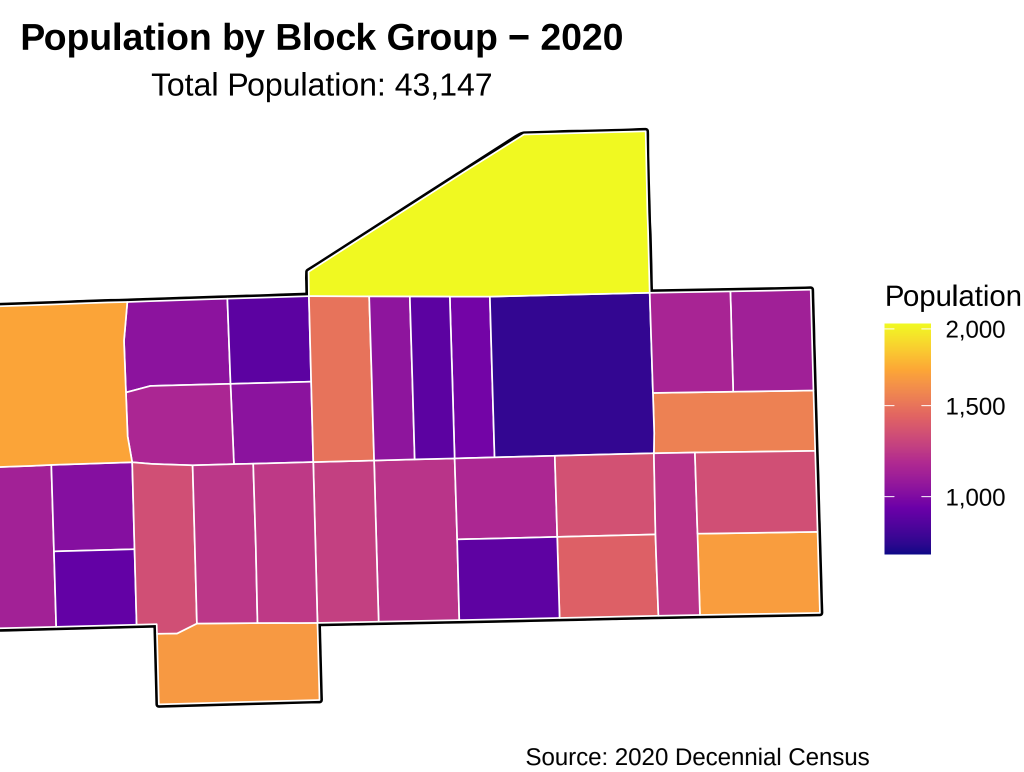
```r
pop_map_2020 <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black", linewidth = 1.2) +
  geom_sf(data = pop_2020_data, aes(fill = total_pop), color = "white", linewidth = 0.3) +
  scale_fill_viridis_c(
    name = "Population",
    option = "plasma",
    breaks = c(500, 1000, 1500, 2000, 2500, 3000),
    labels = scales::comma,
    trans = "sqrt"
  ) +
  labs(title = "Population by Block Group - 2020",
       subtitle = paste("Total Population:", scales::comma(sum(pop_2020_data$total_pop, na.rm = TRUE))),
       caption = "Source: 2020 Decennial Census") +
  theme_void() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    legend.position = "right"
  )

cat("POPULATION DISTRIBUTION COMPARISON: 2010 vs 2020\n")
```

```
## POPULATION DISTRIBUTION COMPARISON: 2010 vs 2020
```

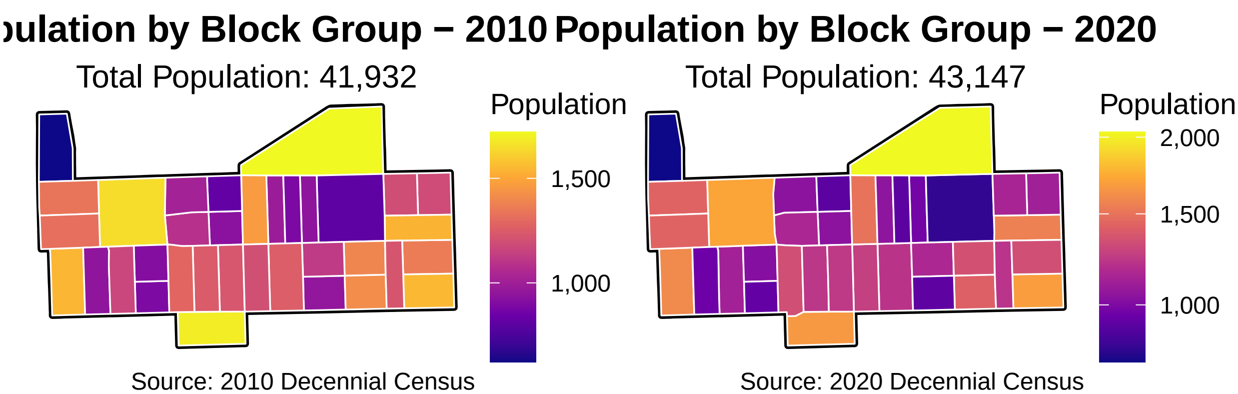
```r
print(pop_map_2010)
```



```r
print(pop_map_2020)
```




```r
# Side-by-side comparison of population maps
population_comparison <- pop_map_2010 + pop_map_2020 +
  plot_annotation(
    theme = theme(
      plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
      plot.subtitle = element_text(hjust = 0.5, size = 12)
    )
  )

print(population_comparison)
```




```r
# Analyze population changes between 2010 and 2020
population_change_analysis <- dunning_bg_2020 %>%
  st_drop_geometry() %>%
  select(GEOID, total_pop_2020 = total_pop) %>%
  left_join(
    dunning_bg_2010 %>%
      st_drop_geometry() %>%
      select(GEOID, total_pop_2010 = total_pop),
    by = "GEOID"
  ) %>%
  mutate(
    population_change = total_pop_2020 - total_pop_2010,
    percent_change = (population_change / total_pop_2010) * 100
  )

print("Population Change Summary by Block Group:")
```

```
## [1] "Population Change Summary by Block Group:"
```

```r
print(population_change_analysis)
```

```
## # A tibble: 35 x 5
##    GEOID        total_pop_2020 total_pop_2010 population_change percent_change
##    <chr>                 <dbl>          <dbl>             <dbl>          <dbl>
##  1 170311702002           1556           1538                18           1.17
```

```
##   2 170311710005           1218           1242               -24           -1.93
##   3 170311703002            968            949                19            2.00
##   4 170311703001            797            835               -38           -4.55
##   5 170311707003           1122           1152               -30           -2.60
##   6 170311704004           1043           1012                31            3.06
##   7 170311704001            903            844                59            6.99
##   8 170311701001           2034           1752               282           16.1
##   9 170311711002           1683           1553               130            8.37
##  10 170311710004           1158           1109                49            4.42
## # i 25 more rows
```

```r
# Calculate and display population change summary
cat("Total Population 2010:", sum(dunning_bg_2010$total_pop, na.rm = TRUE), "\n")
```

```
## Total Population 2010: 41932
```

```r
cat("Total Population 2020:", sum(dunning_bg_2020$total_pop, na.rm = TRUE), "\n")
```

```
## Total Population 2020: 43147
```

```r
cat("Net Population Change:", sum(dunning_bg_2020$total_pop, na.rm = TRUE) - sum(dunning_bg_2010$total_p
```

```
## Net Population Change: 1215
```

```r
cat("Percent Change:", round(((sum(dunning_bg_2020$total_pop, na.rm = TRUE) - sum(dunning_bg_2010$total_
```

```
## Percent Change: 2.9 %
```

```r
# Compare block group counts between years
cat("Number of block groups in 2010:", nrow(dunning_bg_2010), "\n")
```

```
## Number of block groups in 2010: 35
```

```r
cat("Number of block groups in 2020:", nrow(dunning_bg_2020), "\n")
```

```
## Number of block groups in 2020: 35
```

```r
cat("Difference:", nrow(dunning_bg_2020) - nrow(dunning_bg_2010), "block groups\n")
```

```
## Difference: 0 block groups
```

```r
# Identify block groups with significant population changes
significant_changes <- population_change_analysis %>%
  filter(abs(percent_change) > 10)

# Part 2: ACS Variable Selection and Justification

# Define 7 ACS variables for detailed community analysis
```

```r
acs_vars_2022 <- c(
  total_pop = "B01003_001",
  race_white = "B02001_002",
  race_black = "B02001_003",
  hispanic = "B03002_012",
  med_income = "B19013_001",
  hh_size = "B25010_001",
  owner_occupied = "B25003_002",
  renter_occupied = "B25003_003"
)

# Get latest ACS data for Cook County block groups
cook_bg_acs_2022 <- get_acs(
  geography = "block group",
  variables = acs_vars_2022,
  year = 2022,
  survey = "acs5",
  state = "IL",
  county = "Cook",
  geometry = TRUE,
  output = "wide"
)
```

```
## Getting data from the 2018-2022 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future session

##   |                                                                              |
```

```r
# Transform and filter for Dunning block groups
cook_bg_acs_2022 <- st_transform(cook_bg_acs_2022, st_crs(dunning_sf))
dunning_bg_acs_2022 <- cook_bg_acs_2022 %>%
  mutate(tract_code = substr(GEOID, 1, 11)) %>%
  filter(tract_code %in% ca_tract_mapping$GEOID_tract)

# Calculate housing tenure percentages
dunning_bg_acs_2022 <- dunning_bg_acs_2022 %>%
  mutate(
    total_housing = owner_occupiedE + renter_occupiedE,
    pct_owner_occupied = ifelse(total_housing > 0, (owner_occupiedE / total_housing) * 100, NA)
  )

# Display geographic composition summary
cat("Dunning Community Area (CA 17) geographic composition:\n")
```

```
## Dunning Community Area (CA 17) geographic composition:
```

```r
cat("•", nrow(ca_tract_mapping), "census tracts\n")
```

```
## • 11 census tracts
```

```r
cat("•", nrow(dunning_bg_2020), "census block groups (2020 Decennial)\n")
```

```
## • 35 census block groups (2020 Decennial)
```

```r
cat("•", nrow(dunning_bg_2010), "census block groups (2010 Decennial)\n")
```

```
## • 35 census block groups (2010 Decennial)
```

```r
cat("•", nrow(dunning_bg_acs_2022), "census block groups (2022 ACS)\n\n")
```

```
## • 35 census block groups (2022 ACS)
```

```r
# Document selected variables and their justifications
selected_variables <- data.frame(
  Variable_Name = c("total_pop", "race_white", "race_black", "hispanic", "med_income", "hh_size", "owne
  Census_Code = c("B01003_001", "B02001_002", "B02001_003", "B03002_012", "B19013_001", "B25010_001", "B
  Description = c(
    "Total population - Count of all people in geographic area",
    "White alone population - People identifying as White alone",
    "Black or African American alone population - People identifying as Black alone",
    "Hispanic or Latino population - People of Hispanic or Latino origin",
    "Median household income - Middle value of household income distribution",
    "Average household size - Mean number of people per household",
    "Owner-occupied housing units - Homes owned by residents"
  ),
  Data_Source = rep("ACS 5-Year 2022", 7),
  Geography_Level = rep("Block Group", 7)
)

print(selected_variables)
```

```
##    Variable_Name Census_Code
## 1      total_pop  B01003_001
## 2     race_white  B02001_002
## 3     race_black  B02001_003
## 4       hispanic  B03002_012
## 5     med_income  B19013_001
## 6        hh_size  B25010_001
## 7 owner_occupied  B25003_002
##                                                                      Description
## 1                      Total population - Count of all people in geographic area
## 2                     White alone population - People identifying as White alone
## 3 Black or African American alone population - People identifying as Black alone
## 4            Hispanic or Latino population - People of Hispanic or Latino origin
## 5        Median household income - Middle value of household income distribution
## 6                  Average household size - Mean number of people per household
## 7                        Owner-occupied housing units - Homes owned by residents
##       Data_Source Geography_Level
## 1 ACS 5-Year 2022     Block Group
## 2 ACS 5-Year 2022     Block Group
## 3 ACS 5-Year 2022     Block Group
```

```
## 4 ACS 5-Year 2022     Block Group
## 5 ACS 5-Year 2022     Block Group
## 6 ACS 5-Year 2022     Block Group
## 7 ACS 5-Year 2022     Block Group
```

```r
write_csv(selected_variables, "dunning_acs_variable_descriptions.csv")

# Analyze data quality and completeness
cat("1. DATA AVAILABILITY ANALYSIS:\n")
```

```
## 1. DATA AVAILABILITY ANALYSIS:
```

```r
cat("-----------------------------\n")
```

```
## -----------------------------
```

```r
missing_analysis <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  summarise(across(c(total_popE, race_whiteE, race_blackE, hispanicE, med_incomeE, hh_sizeE, owner_occu

print("Missing value analysis for 2022 ACS data:")
```

```
## [1] "Missing value analysis for 2022 ACS data:"
```

```r
print(missing_analysis)
```

```
##   total_popE_missing total_popE_complete total_popE_pct_missing
## 1                  0                  35                      0
##   race_whiteE_missing race_whiteE_complete race_whiteE_pct_missing
## 1                   0                   35                       0
##   race_blackE_missing race_blackE_complete race_blackE_pct_missing
## 1                   0                   35                       0
##   hispanicE_missing hispanicE_complete hispanicE_pct_missing
## 1                 0                 35                     0
##   med_incomeE_missing med_incomeE_complete med_incomeE_pct_missing
## 1                   3                   32                8.571429
##   hh_sizeE_missing hh_sizeE_complete hh_sizeE_pct_missing
## 1                0                35                    0
##   owner_occupiedE_missing owner_occupiedE_complete owner_occupiedE_pct_missing
## 1                       0                       35                           0
```

```r
# Analyze statistical distributions of ACS variables
cat("\nVARIABLE DISTRIBUTION ANALYSIS:\n")
```

```
##
## VARIABLE DISTRIBUTION ANALYSIS:
```

```
distribution_summary <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  select(total_popE, race_whiteE, race_blackE, hispanicE, med_incomeE, hh_sizeE, owner_occupiedE) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  group_by(variable) %>%
  summarise(
    n = n(),
    mean = mean(value, na.rm = TRUE),
    median = median(value, na.rm = TRUE),
    sd = sd(value, na.rm = TRUE),
    min = min(value, na.rm = TRUE),
    max = max(value, na.rm = TRUE),
    cv = sd/mean * 100
  )

print("Distribution summary for ACS variables:")
```

```
## [1] "Distribution summary for ACS variables:"
```

```
print(distribution_summary)
```

```
## # A tibble: 7 x 8
##   variable            n     mean   median        sd      min       max    cv
##   <chr>           <int>    <dbl>    <dbl>     <dbl>    <dbl>     <dbl> <dbl>
## 1 hh_sizeE           35     2.69     2.67     0.542     1.64      4.36  20.2
## 2 hispanicE          35   418.      385       262.        36      1131    62.8
## 3 med_incomeE        35 90748.    85404     25814.     46571    153640    28.4
## 4 owner_occupiedE    35   362.      364       149.       127       901    41.2
## 5 race_blackE        35    36.4       3        70.7        0       337   194.
## 6 race_whiteE        35   814.      851       305.       307      1715    37.5
## 7 total_popE         35  1216.     1188       428.       459      2109    35.2
```

```
# Measure spatial variation across block groups
cat("\nSPATIAL VARIATION ANALYSIS:\n")
```

```
##
## SPATIAL VARIATION ANALYSIS:
```

```
spatial_variation <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  mutate(
    pct_white = ifelse(total_popE > 0, (race_whiteE / total_popE) * 100, NA),
    pct_black = ifelse(total_popE > 0, (race_blackE / total_popE) * 100, NA),
    pct_hispanic = ifelse(total_popE > 0, (hispanicE / total_popE) * 100, NA),
    pct_owner = ifelse(total_housing > 0, (owner_occupiedE / total_housing) * 100, NA)
  ) %>%
  summarise(
    pop_range = max(total_popE, na.rm = TRUE) - min(total_popE, na.rm = TRUE),
    income_range = max(med_incomeE, na.rm = TRUE) - min(med_incomeE, na.rm = TRUE),
    white_pct_range = max(pct_white, na.rm = TRUE) - min(pct_white, na.rm = TRUE),
    black_pct_range = max(pct_black, na.rm = TRUE) - min(pct_black, na.rm = TRUE),
```

```r
    hispanic_pct_range = max(pct_hispanic, na.rm = TRUE) - min(pct_hispanic, na.rm = TRUE),
    owner_pct_range = max(pct_owner, na.rm = TRUE) - min(pct_owner, na.rm = TRUE)
  )

print("Spatial variation across Dunning block groups:")
```

```
## [1] "Spatial variation across Dunning block groups:"
```

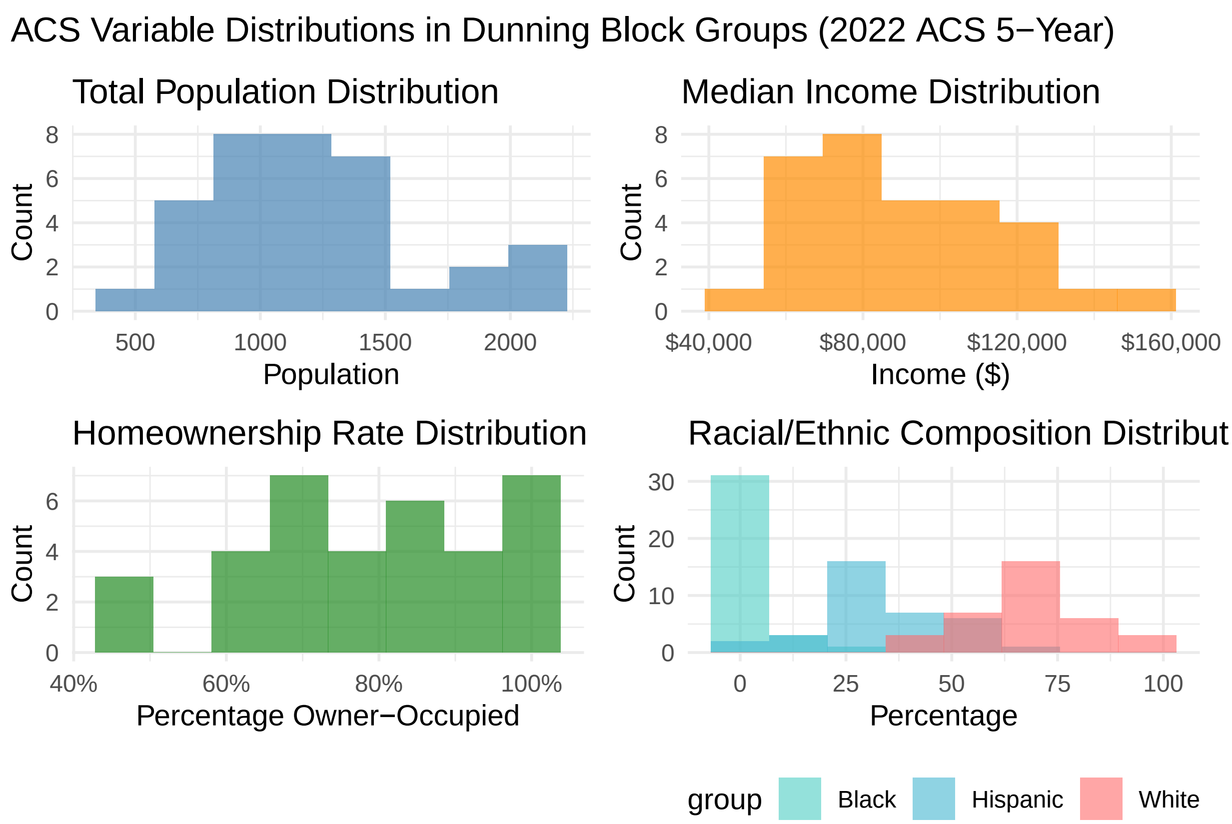
```r
print(spatial_variation)
```

```
##   pop_range income_range white_pct_range black_pct_range hispanic_pct_range
## 1      1650       107069         56.8732        32.43503           62.27693
##   owner_pct_range
## 1        53.36323
```

```r
# Visualizations of variable distributions

p1 <- ggplot(dunning_bg_acs_2022, aes(x = total_popE)) +
  geom_histogram(fill = "steelblue", alpha = 0.7, bins = 8) +
  labs(title = "Total Population Distribution", x = "Population", y = "Count") +
  theme_minimal()

p2 <- ggplot(dunning_bg_acs_2022, aes(x = med_incomeE)) +
  geom_histogram(fill = "darkorange", alpha = 0.7, bins = 8) +
  labs(title = "Median Income Distribution", x = "Income ($)", y = "Count") +
  scale_x_continuous(labels = scales::dollar) +
  theme_minimal()

p3 <- ggplot(dunning_bg_acs_2022, aes(x = pct_owner_occupied)) +
  geom_histogram(fill = "forestgreen", alpha = 0.7, bins = 8) +
  labs(title = "Homeownership Rate Distribution", x = "Percentage Owner-Occupied", y = "Count") +
  scale_x_continuous(labels = scales::percent_format(scale = 1)) +
  theme_minimal()

racial_composition <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  mutate(
    pct_white = ifelse(total_popE > 0, (race_whiteE / total_popE) * 100, NA),
    pct_black = ifelse(total_popE > 0, (race_blackE / total_popE) * 100, NA),
    pct_hispanic = ifelse(total_popE > 0, (hispanicE / total_popE) * 100, NA)
  ) %>%
  select(pct_white, pct_black, pct_hispanic) %>%
  pivot_longer(cols = everything(), names_to = "group", values_to = "percentage")

p4 <- ggplot(racial_composition, aes(x = percentage, fill = group)) +
  geom_histogram(alpha = 0.6, position = "identity", bins = 8) +
  labs(title = "Racial/Ethnic Composition Distribution", x = "Percentage", y = "Count") +
  scale_fill_manual(
    values = c("pct_white" = "#FF6B6B", "pct_black" = "#4ECDC4", "pct_hispanic" = "#45B7D1"),
    labels = c("pct_white" = "White", "pct_black" = "Black", "pct_hispanic" = "Hispanic")
  ) +
  theme_minimal() +
```

```r
  theme(legend.position = "bottom")

variable_distributions <- (p1 + p2) / (p3 + p4) +
  plot_annotation(title = "ACS Variable Distributions in Dunning Block Groups (2022 ACS 5-Year)")

print(variable_distributions)
```

```
## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```r
ggsave("acs_variable_distributions.png", variable_distributions, width = 12, height = 10, dpi = 300)
```

```
## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

```r
# Part 3: Network Analysis for Community Structure

n_bg_dunning <- nrow(dunning_bg_acs_2022)
cat("Number of block groups in Dunning:", n_bg_dunning, "\n")
```

```
## Number of block groups in Dunning: 35
```

```r
# Identify communities that border Dunning
all_cas_sf <- ca_sf
dunning_neighbors <- all_cas_sf %>%
  st_filter(dunning_sf, .predicate = st_touches)

cat("Dunning's neighboring community areas:\n")
```

```
## Dunning's neighboring community areas:
```

```r
print(dunning_neighbors$community)
```

```
## [1] "PORTAGE PARK"   "MONTCLARE"      "BELMONT CRAGIN" "OHARE"
```

```r
# Get ACS data for neighboring areas
neighbor_bg_2022 <- get_acs(
  geography = "block group",
  variables = acs_vars_2022,
  year = 2022,
  survey = "acs5",
  state = "IL",
  county = "Cook",
  geometry = TRUE,
  output = "wide"
)
```

```
## Getting data from the 2018-2022 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future session
```

```r
neighbor_bg_2022 <- st_transform(neighbor_bg_2022, st_crs(dunning_sf))

# Combine Dunning and neighbor block groups
study_area <- bind_rows(dunning_sf, dunning_neighbors)
study_bg_2022 <- st_intersection(neighbor_bg_2022, study_area)
```

```
## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
```

```r
study_bg_combined <- bind_rows(
  dunning_bg_acs_2022 %>% mutate(area_type = "Dunning"),
  study_bg_2022 %>%
    filter(!GEOID %in% dunning_bg_acs_2022$GEOID) %>%
    mutate(area_type = "Neighbor")
)

cat("Total block groups in study area (Dunning + neighbors):", nrow(study_bg_combined), "\n")
```

```
## Total block groups in study area (Dunning + neighbors): 242
```

```r
# Prepare data for network analysis
analysis_data <- study_bg_combined %>%
  st_drop_geometry() %>%
  mutate(
    pct_white = ifelse(total_popE > 0, (race_whiteE / total_popE) * 100, NA),
    pct_black = ifelse(total_popE > 0, (race_blackE / total_popE) * 100, NA),
    pct_hispanic = ifelse(total_popE > 0, (hispanicE / total_popE) * 100, NA),
    pct_owner = ifelse(total_housing > 0, (owner_occupiedE / total_housing) * 100, NA),
    income_scaled = scale(med_incomeE)
  ) %>%
  select(GEOID, total_popE, pct_white, pct_black, pct_hispanic, income_scaled, hh_sizeE, pct_owner, area
  drop_na()

# Standardize variables for similarity calculation
scaled_data <- analysis_data %>%
  select(total_popE, pct_white, pct_black, pct_hispanic, income_scaled, hh_sizeE, pct_owner) %>%
  scale() %>%
  as.data.frame()

rownames(scaled_data) <- analysis_data$GEOID

cat("Final analysis dataset dimensions:", dim(scaled_data), "\n")
```

```
## Final analysis dataset dimensions: 32 7
```

```r
cat("Variables used for similarity:", names(scaled_data), "\n")
```

```
## Variables used for similarity: total_popE pct_white pct_black pct_hispanic income_scaled hh_sizeE pct
```

```r
# Block group similarity network. Calculate similarity between block groups

bg_distances <- dist(scaled_data, method = "euclidean")
bg_similarity <- 1 / (1 + as.matrix(bg_distances))
k <- 5
adjacency_matrix <- matrix(0, nrow = nrow(bg_similarity), ncol = ncol(bg_similarity))

for(i in 1:nrow(bg_similarity)) {
  similar_indices <- order(bg_similarity[i,], decreasing = TRUE)[2:(k+1)]
  adjacency_matrix[i, similar_indices] <- bg_similarity[i, similar_indices]
  adjacency_matrix[similar_indices, i] <- bg_similarity[similar_indices, i]
}

bg_graph <- graph_from_adjacency_matrix(adjacency_matrix, mode = "undirected", weighted = TRUE)

V(bg_graph)$GEOID <- analysis_data$GEOID
V(bg_graph)$pct_white <- analysis_data$pct_white
V(bg_graph)$pct_black <- analysis_data$pct_black
V(bg_graph)$pct_hispanic <- analysis_data$pct_hispanic
V(bg_graph)$income <- analysis_data$income_scaled
V(bg_graph)$area_type <- analysis_data$area_type

cat("Network created with", vcount(bg_graph), "vertices and", ecount(bg_graph), "edges\n")
```

```
## Network created with 32 vertices and 120 edges
```

```
louvain_communities <- cluster_louvain(bg_graph)
V(bg_graph)$louvain_community <- louvain_communities$membership

cat("Detected", length(unique(louvain_communities$membership)), "communities via Louvain algorithm\n")
```

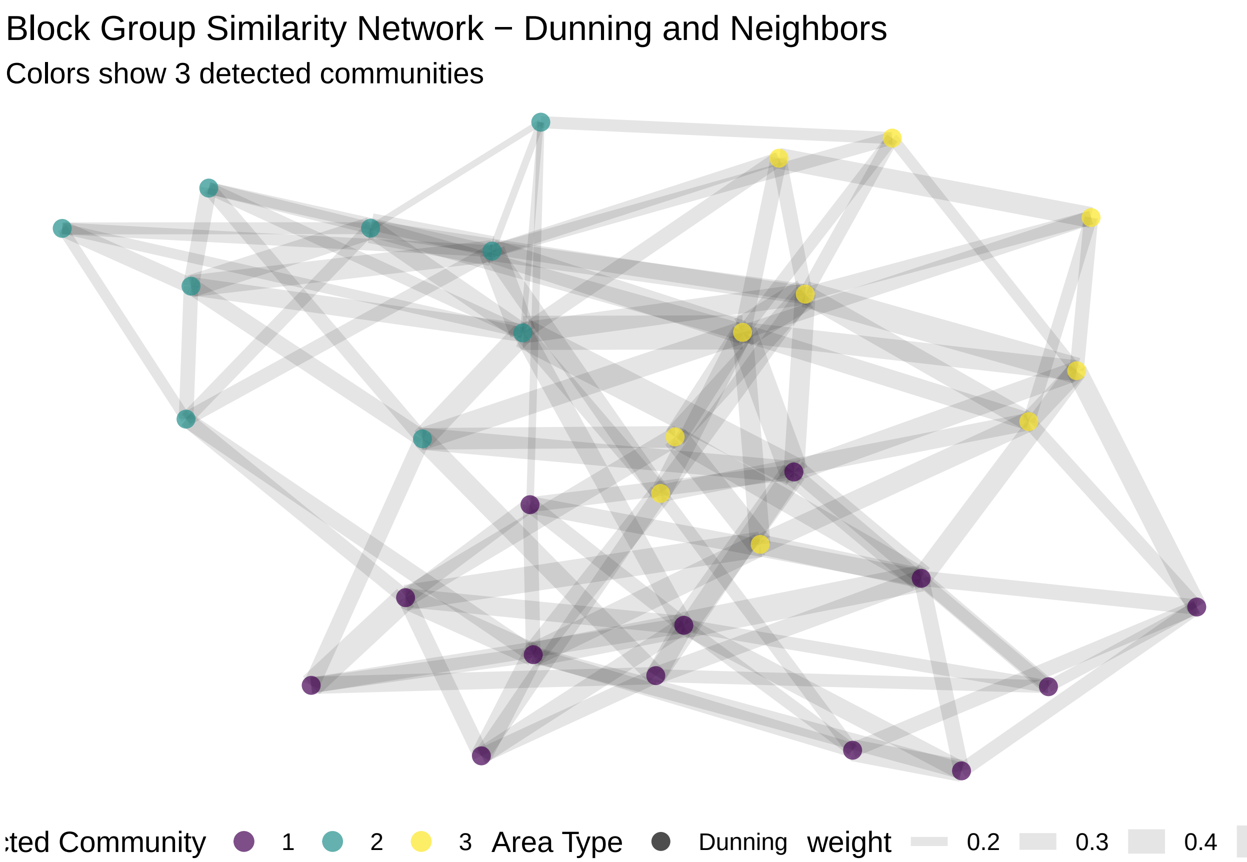
```
## Detected 3 communities via Louvain algorithm
```

```
modularity_score <- modularity(louvain_communities)
cat("Network modularity:", round(modularity_score, 3), "\n")
```

```
## Network modularity: 0.34
```

```
set.seed(123)
layout <- create_layout(bg_graph, layout = "fr")

network_plot <- ggraph(layout) +
  geom_edge_link(alpha = 0.1, aes(width = weight)) +
  geom_node_point(aes(color = as.factor(louvain_community), shape = area_type), size = 3, alpha = 0.7)
  scale_color_viridis_d(name = "Detected Community") +
  scale_shape_manual(name = "Area Type", values = c(16, 17)) +
  labs(title = "Block Group Similarity Network - Dunning and Neighbors",
       subtitle = paste("Colors show", length(unique(louvain_communities$membership)), "detected commun
  theme_void() +
  theme(legend.position = "bottom")

print(network_plot)
```

```
## Warning: The `trans` argument of `continuous_scale()` is deprecated as of ggplot2 3.5.0.
## i Please use the `transform` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
ggsave("network_communities.png", network_plot, width = 12, height = 10, dpi = 300)

community_comparison <- data.frame(
  GEOID = V(bg_graph)$GEOID,
  Detected_Community = V(bg_graph)$louvain_community,
  Official_Area = V(bg_graph)$area_type
) %>%
  group_by(Detected_Community, Official_Area) %>%
  summarise(Count = n(), .groups = "drop")

print("Comparison between detected communities and official areas:")
```

```
## [1] "Comparison between detected communities and official areas:"
```

```r
print(community_comparison)
```

```
## # A tibble: 3 x 3
##   Detected_Community Official_Area Count
##                <dbl> <chr>         <int>
## 1                  1 Dunning          13
## 2                  2 Dunning           9
## 3                  3 Dunning          10
```

```r
community_profiles <- analysis_data %>%
  left_join(data.frame(GEOID = V(bg_graph)$GEOID,
                       Detected_Community = V(bg_graph)$louvain_community), by = "GEOID") %>%
  group_by(Detected_Community) %>%
  summarise(
    n_block_groups = n(),
    avg_pct_white = mean(pct_white, na.rm = TRUE),
    avg_pct_black = mean(pct_black, na.rm = TRUE),
    avg_pct_hispanic = mean(pct_hispanic, na.rm = TRUE),
    avg_income = mean(income_scaled, na.rm = TRUE),
    contains_dunning = any(area_type == "Dunning")
  )

print("Profiles of detected communities:")
```

```
## [1] "Profiles of detected communities:"
```

```r
print(community_profiles)
```

```
## # A tibble: 3 x 7
##   Detected_Community n_block_groups avg_pct_white avg_pct_black avg_pct_hispanic
##                <dbl>          <int>         <dbl>         <dbl>            <dbl>
## 1                  1             13          68.8          3.01             29.2
## 2                  2              9          53.0          4.86             46.1
## 3                  3             10          77.7          1.81             30.2
## # i 2 more variables: avg_income <dbl>, contains_dunning <lgl>
```

```r
cat("\nARGUMENTS FOR DUNNING AS A COMMUNITY\n")
```

```
##
## ARGUMENTS FOR DUNNING AS A COMMUNITY
```

```r
dunning_vertices <- V(bg_graph)[area_type == "Dunning"]
dunning_internal_edges <- E(bg_graph)[dunning_vertices %--% dunning_vertices]
dunning_external_edges <- E(bg_graph)[dunning_vertices %--% V(bg_graph)[area_type != "Dunning"]]

internal_density <- ifelse(length(dunning_internal_edges) > 0,
                           mean(dunning_internal_edges$weight, na.rm = TRUE), 0)
external_density <- ifelse(length(dunning_external_edges) > 0,
                           mean(dunning_external_edges$weight, na.rm = TRUE), 0)

cat("Internal similarity within Dunning:", round(internal_density, 3), "\n")
```

```
## Internal similarity within Dunning: 0.314
```

```r
cat("External similarity from Dunning to neighbors:", round(external_density, 3), "\n")
```

```
## External similarity from Dunning to neighbors: 0
```

```r
cat("Cohesion ratio (internal/external):",
    ifelse(external_density > 0, round(internal_density/external_density, 2), "Infinite"), "\n")
```

```
## Cohesion ratio (internal/external): Infinite
```

```r
dunning_demographics <- analysis_data %>%
  filter(area_type == "Dunning") %>%
  select(pct_white, pct_black, pct_hispanic)

dunning_variance <- apply(dunning_demographics, 2, var, na.rm = TRUE)
neighbor_variance <- analysis_data %>%
  filter(area_type != "Dunning") %>%
  select(pct_white, pct_black, pct_hispanic) %>%
  apply(2, var, na.rm = TRUE)

variance_comparison <- data.frame(
  Variable = names(dunning_variance),
  Dunning_Variance = dunning_variance,
  Neighbor_Variance = neighbor_variance,
  Ratio = dunning_variance / neighbor_variance
)

print("Variance comparison (lower = more homogeneous):")
```

```
## [1] "Variance comparison (lower = more homogeneous):"
```

```r
print(variance_comparison)
```

```
##                  Variable Dunning_Variance Neighbor_Variance Ratio
## pct_white       pct_white        184.77955                NA    NA
## pct_black       pct_black         44.63837                NA    NA
## pct_hispanic pct_hispanic        236.45264                NA    NA
```

```r
cat("\nARGUMENTS AGAINST DUNNING AS A COMMUNITY\n")
```

```
##
## ARGUMENTS AGAINST DUNNING AS A COMMUNITY
```

```r
dunning_communities <- community_comparison %>%
  filter(Official_Area == "Dunning") %>%
  group_by(Detected_Community) %>%
  summarise(Count = sum(Count))

cat("Dunning block groups are split across", nrow(dunning_communities), "detected communities\n")
```

```
## Dunning block groups are split across 3 detected communities
```

```r
if(nrow(dunning_communities) > 1) {
  cat("Dunning lacks internal cohesion as a single community\n")
}
```

```
## Dunning lacks internal cohesion as a single community
```

```r
boundary_strength <- ifelse(internal_density > 0, external_density / internal_density, Inf)
cat("Boundary permeability index:", round(boundary_strength, 3), "\n")
```

```
## Boundary permeability index: 0
```

```r
cat("Values closer to 1 indicate weak boundaries between Dunning and neighbors\n")
```

```
## Values closer to 1 indicate weak boundaries between Dunning and neighbors
```

```r
cat("\nPROPOSAL FOR ALTERNATIVE COMMUNITY ORGANIZATION\n")
```

```
##
## PROPOSAL FOR ALTERNATIVE COMMUNITY ORGANIZATION
```

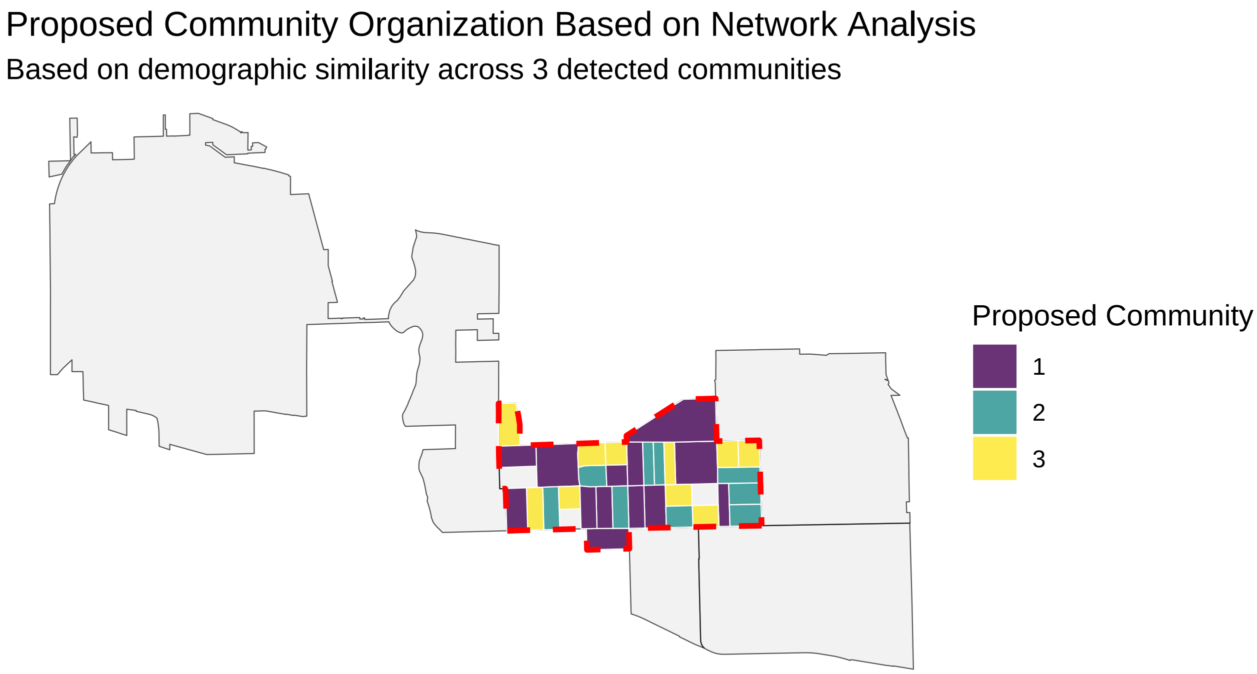
```r
proposed_communities <- community_profiles %>%
  filter(n_block_groups >= 3) %>%
  arrange(desc(n_block_groups))

print("Proposed community organization based on network analysis:")
```

```
## [1] "Proposed community organization based on network analysis:"
```

```r
print(proposed_communities %>% select(Detected_Community, n_block_groups, avg_pct_white, avg_pct_black,
```

```
## # A tibble: 3 x 5
##   Detected_Community n_block_groups avg_pct_white avg_pct_black avg_pct_hispanic
##                <dbl>          <int>         <dbl>         <dbl>            <dbl>
## 1                  1             13          68.8          3.01             29.2
## 2                  3             10          77.7          1.81             30.2
## 3                  2              9          53.0          4.86             46.1
```

```r
proposal_data <- study_bg_combined %>%
  left_join(data.frame(GEOID = V(bg_graph)$GEOID,
                       Community = V(bg_graph)$louvain_community), by = "GEOID") %>%
  filter(Community %in% proposed_communities$Detected_Community)

proposal_plot <- ggplot() +
  geom_sf(data = study_area, fill = "lightgray", alpha = 0.3) +
  geom_sf(data = proposal_data, aes(fill = as.factor(Community)), color = "white", alpha = 0.8) +
  geom_sf(data = dunning_sf, fill = NA, color = "red", size = 1, linetype = "dashed") +
  scale_fill_viridis_d(name = "Proposed Community") +
  labs(title = "Proposed Community Organization Based on Network Analysis",
       subtitle = paste("Based on demographic similarity across", nrow(proposed_communities), "detected
       caption = "Dashed red line shows current Dunning boundary") +
  theme_void()

print(proposal_plot)
```



Dashed red line shows current Dunning boundary

```r
ggsave("proposed_communities.png", proposal_plot, width = 10, height = 8, dpi = 300)

cat("\nSUMMARY AND CONCLUSIONS\n")
```

```
##
## SUMMARY AND CONCLUSIONS
```

```r
dunning_cohesion <- ifelse(internal_density > external_density, "High", "Low")
boundary_strength_assessment <- ifelse(boundary_strength < 0.5, "Strong", "Weak")
community_unity <- ifelse(nrow(dunning_communities) == 1, "Unified", "Fragmented")

cat("FINAL ASSESSMENT OF DUNNING AS A COMMUNITY:\n")
```

```
## FINAL ASSESSMENT OF DUNNING AS A COMMUNITY:
```

```r
cat("• Internal cohesion:", dunning_cohesion, "\n")
```

```
## • Internal cohesion: High
```

```r
cat("• Boundary strength:", boundary_strength_assessment, "\n")
```

```
## • Boundary strength: Strong
```

```r
cat("• Community unity:", community_unity, "\n")
```

```
## • Community unity: Fragmented
```

```r
cat("• Network modularity:", round(modularity_score, 3), "\n")
```

```
## • Network modularity: 0.34
```

```r
cat("• Proposed communities:", nrow(proposed_communities), "\n\n")
```

```
## • Proposed communities: 3
```

```r
cat("DATA-BASED RECOMMENDATION:\n")
```

```
## DATA-BASED RECOMMENDATION:
```

```r
if (dunning_cohesion == "High" && boundary_strength_assessment == "Strong" && community_unity == "Unifi
  cat("Dunning functions well as a single community\n")
} else {
  cat("The proposed", nrow(proposed_communities), "communities better reflect demographic patterns\n")
}
```

```
## The proposed 3 communities better reflect demographic patterns
```

```r
# Part 4: Airport Proximity Analysis

# Load and prepare airport data
# O'Hare International Airport coordinates (from public data)
ohare_airport <- data.frame(
  name = "O'Hare International Airport",
  longitude = -87.9048,
  latitude = 41.9786
) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326) %>%
  st_transform(st_crs(dunning_sf))

# Calculate distance from each block group centroid to O'Hare
dunning_bg_acs_2022 <- dunning_bg_acs_2022 %>%
  mutate(
    centroid = st_centroid(geometry),
    distance_to_ohare = as.numeric(st_distance(centroid, ohare_airport)) * 0.000621371, # Convert to mi
    distance_km = as.numeric(st_distance(centroid, ohare_airport)) / 1000
  )

# Also calculate for the broader study area
study_bg_combined <- study_bg_combined %>%
  mutate(
    centroid = st_centroid(geometry),
    distance_to_ohare = as.numeric(st_distance(centroid, ohare_airport)) * 0.000621371,
    distance_km = as.numeric(st_distance(centroid, ohare_airport)) / 1000
```

```
  )

# Analyze relationship between airport proximity and demographics
distance_analysis <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  select(
    GEOID,
    distance_to_ohare,
    total_popE,
    med_incomeE,
    pct_owner_occupied,
    race_whiteE,
    race_blackE,
    hispanicE
  ) %>%
  mutate(
    pct_white    = (race_whiteE / total_popE) * 100,
    pct_black    = (race_blackE / total_popE) * 100,
    pct_hispanic = (hispanicE / total_popE) * 100
  ) %>%
  select(
    GEOID, distance_to_ohare, total_popE, med_incomeE,
    pct_owner_occupied, pct_white, pct_black, pct_hispanic
  )

# Correlation analysis
correlation_matrix <- distance_analysis %>%
  select(distance_to_ohare, med_incomeE, pct_owner_occupied, pct_white, pct_black, pct_hispanic) %>%
  cor(use = "complete.obs")

print("Correlation between airport distance and demographic variables:")
```

```
## [1] "Correlation between airport distance and demographic variables:"
```

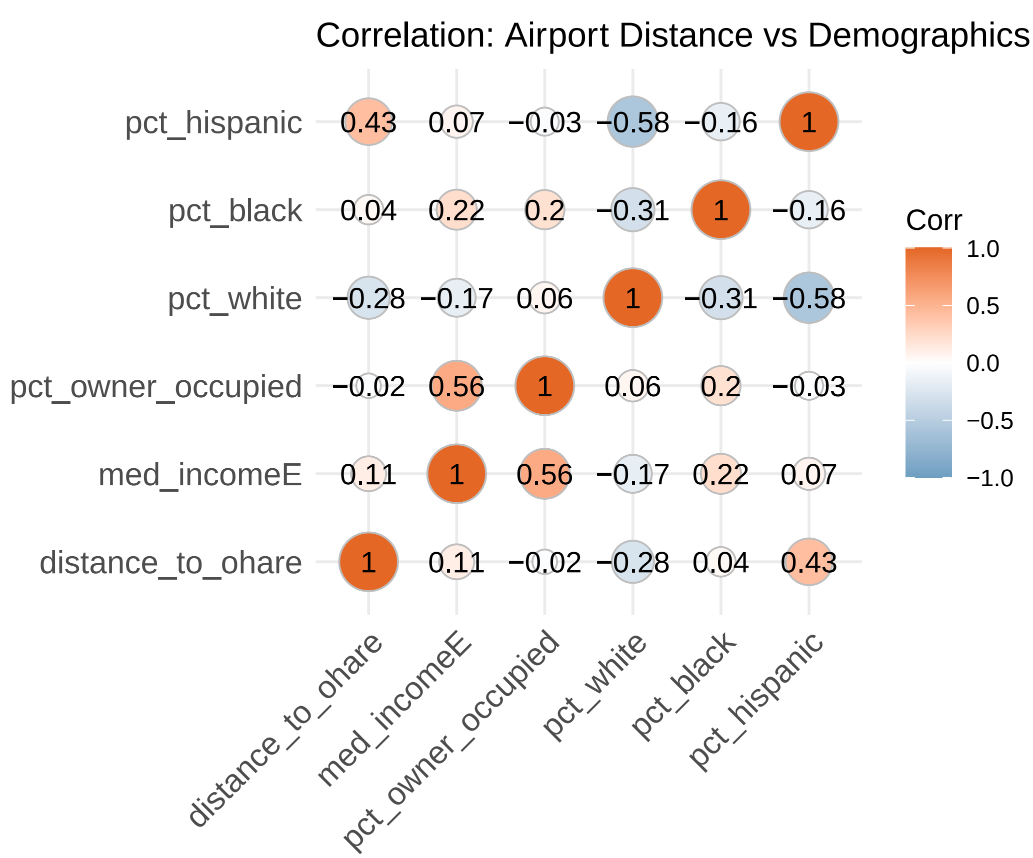
```
print(correlation_matrix)
```

```
##                    distance_to_ohare med_incomeE pct_owner_occupied   pct_white
## distance_to_ohare         1.00000000  0.10985714        -0.01794237 -0.28421020
## med_incomeE               0.10985714  1.00000000         0.56486278 -0.16855829
## pct_owner_occupied       -0.01794237  0.56486278         1.00000000  0.05980201
## pct_white                -0.28421020 -0.16855829         0.05980201  1.00000000
## pct_black                 0.04029326  0.21927154         0.20080315 -0.31381874
## pct_hispanic              0.43282802  0.07387146        -0.02887666 -0.58024435
##                      pct_black pct_hispanic
## distance_to_ohare   0.04029326   0.43282802
## med_incomeE         0.21927154   0.07387146
## pct_owner_occupied  0.20080315  -0.02887666
## pct_white          -0.31381874  -0.58024435
## pct_black           1.00000000  -0.16217103
## pct_hispanic       -0.16217103   1.00000000
```

```r
# Visualize correlation matrix
corr_plot <- ggcorrplot(correlation_matrix,
                        method = "circle",
                        colors = c("#6D9EC1", "white", "#E46726"),
                        title = "Correlation: Airport Distance vs Demographics",
                        lab = TRUE)
```

```
## 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.
```

```r
# make sure it actually draws
print(corr_plot)
```




```r
# Distance zones analysis
distance_zones <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  mutate(
    distance_zone = case_when(
```

```r
      distance_to_ohare < 3 ~ "0-3 miles",
      distance_to_ohare < 5 ~ "3-5 miles",
      distance_to_ohare < 7 ~ "5-7 miles",
      TRUE ~ "7+ miles"
    ),
    distance_zone = factor(distance_zone, levels = c("0-3 miles", "3-5 miles", "5-7 miles", "7+ miles"))
  ) %>%
  group_by(distance_zone) %>%
  summarise(
    n_block_groups = n(),
    avg_distance = mean(distance_to_ohare),
    avg_income = mean(med_incomeE, na.rm = TRUE),
    avg_owner_occupied = mean(pct_owner_occupied, na.rm = TRUE),
    avg_white = mean((race_whiteE / total_popE) * 100, na.rm = TRUE),
    avg_black = mean((race_blackE / total_popE) * 100, na.rm = TRUE),
    avg_hispanic = mean((hispanicE / total_popE) * 100, na.rm = TRUE),
    total_population = sum(total_popE, na.rm = TRUE)
  )

print("Demographic patterns by distance from O'Hare:")
```

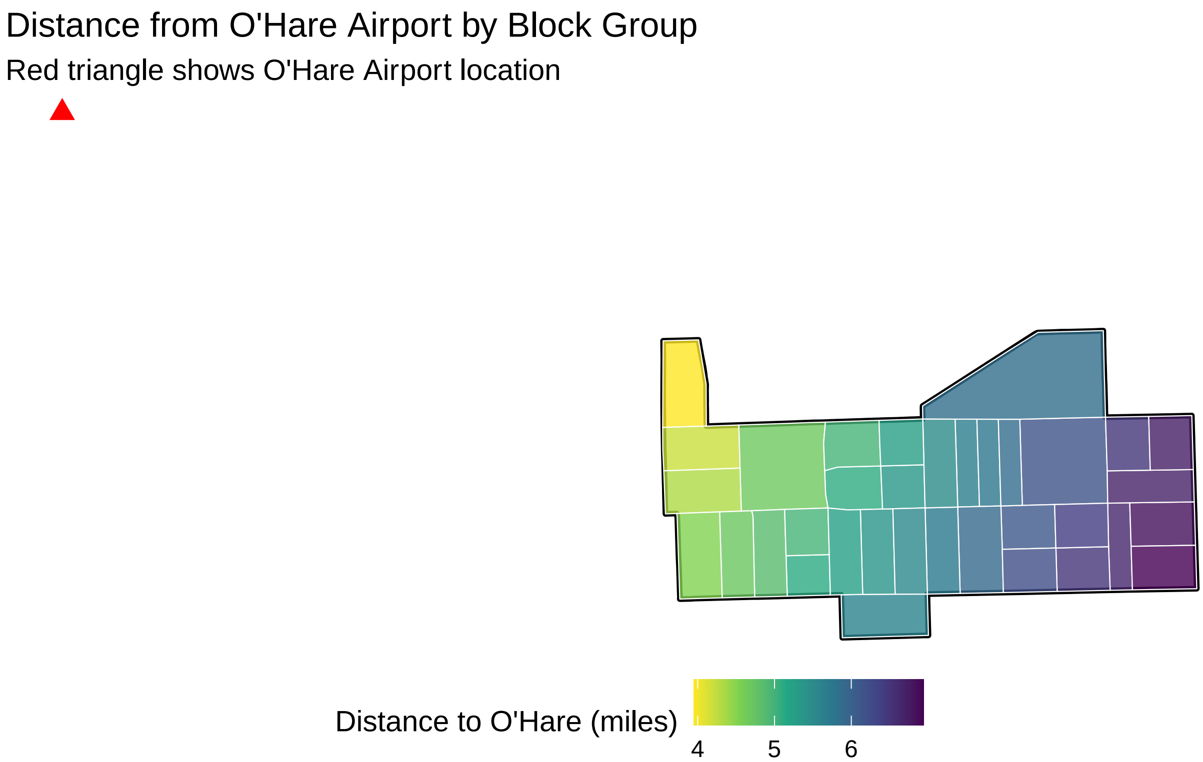
```
## [1] "Demographic patterns by distance from O'Hare:"
```

```r
print(distance_zones)
```

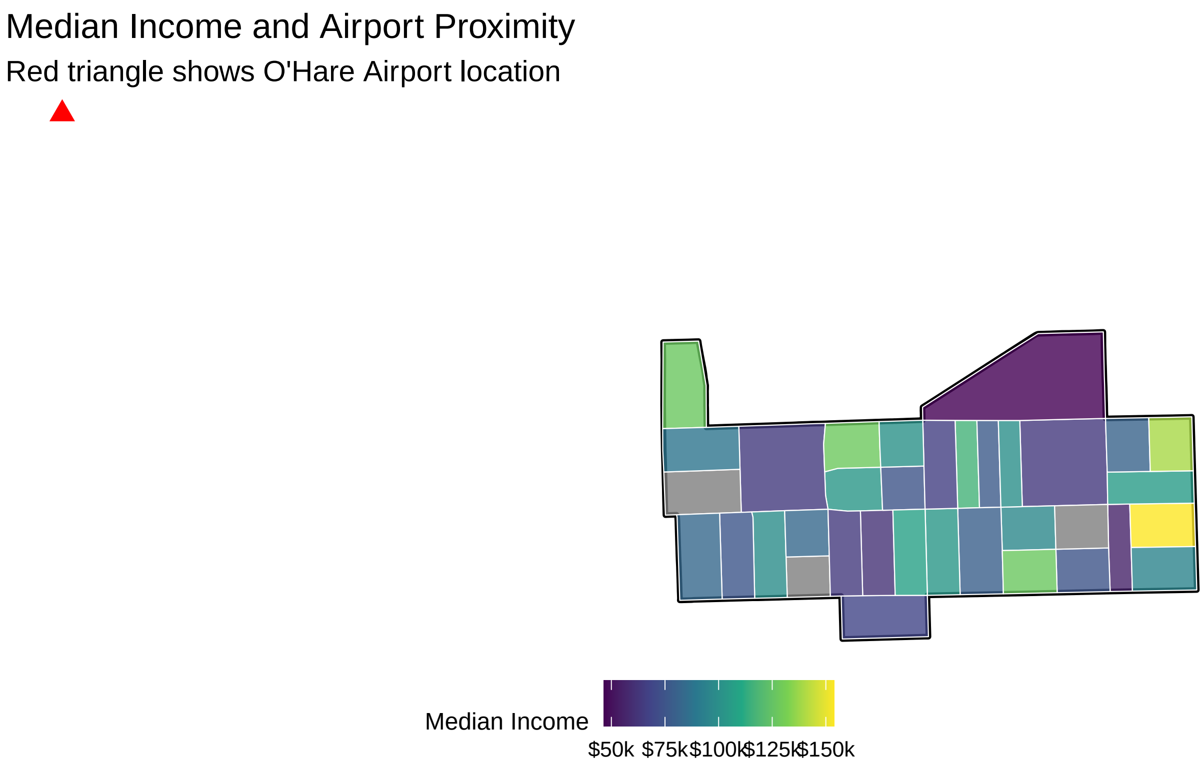
```
## # A tibble: 2 x 9
##   distance_zone n_block_groups avg_distance avg_income avg_owner_occupied
##   <fct>                  <int>        <dbl>      <dbl>              <dbl>
## 1 3-5 miles                  9         4.58     93329.               83.7
## 2 5-7 miles                 26         5.95     89888.               77.8
## # i 4 more variables: avg_white <dbl>, avg_black <dbl>, avg_hispanic <dbl>,
## #   total_population <dbl>
```

```r
# Visualization: Airport proximity maps

# Map 1: Distance to O'Hare
p_distance <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black", size = 1) +
  geom_sf(data = dunning_bg_acs_2022, aes(fill = distance_to_ohare), color = "white", alpha = 0.8) +
  geom_sf(data = ohare_airport, color = "red", size = 3, shape = 17) +
  scale_fill_viridis_c(name = "Distance to O'Hare (miles)", direction = -1) +
  labs(title = "Distance from O'Hare Airport by Block Group",
       subtitle = "Red triangle shows O'Hare Airport location") +
  theme_void() +
  theme(legend.position = "bottom")

# Map 2: Income vs Distance
p_income_distance <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black", size = 1) +
  geom_sf(data = dunning_bg_acs_2022, aes(fill = med_incomeE), color = "white", alpha = 0.8) +
  geom_sf(data = ohare_airport, color = "red", size = 3, shape = 17) +
  scale_fill_viridis_c(
```

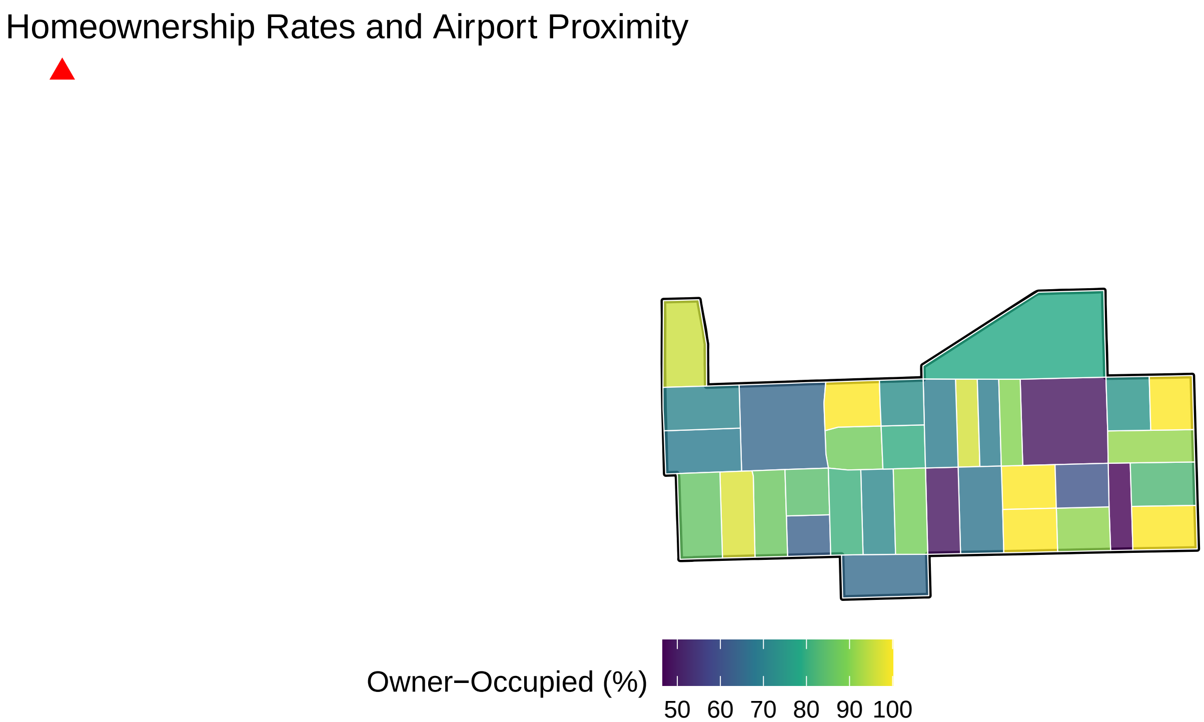
```r
    name   = "Median Income",
    labels = scales::label_dollar(scale = 0.001, suffix = "k")  # e.g., 60k, 80k
  ) +
  labs(
    title    = "Median Income and Airport Proximity",
    subtitle = "Red triangle shows O'Hare Airport location"
  ) +
  theme_void() +
  theme(
    legend.position = "bottom",
    legend.title    = element_text(size = 9),
    legend.text     = element_text(size = 8)
  )

# Map 3: Homeownership vs Distance
p_owner_distance <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black", size = 1) +
  geom_sf(data = dunning_bg_acs_2022, aes(fill = pct_owner_occupied), color = "white", alpha = 0.8) +
  geom_sf(data = ohare_airport, color = "red", size = 3, shape = 17) +
  scale_fill_viridis_c(name = "Owner-Occupied (%)") +
  labs(title = "Homeownership Rates and Airport Proximity") +
  theme_void() +
  theme(legend.position = "bottom")

# Display airport proximity maps separately (no patchwork)
print(p_distance)
```




```r
print(p_income_distance)
```




```r
print(p_owner_distance)
```




```r
# Statistical models for airport proximity effects
library(broom)

# Model 1: Income vs Distance
model_income <- lm(med_incomeE ~ distance_to_ohare, data = dunning_bg_acs_2022)
print("Income vs Distance model:")
```

```
## [1] "Income vs Distance model:"
```

```r
print(summary(model_income))
```

```
##
## Call:
## lm(formula = med_incomeE ~ distance_to_ohare, data = dunning_bg_acs_2022)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45024 -20290  -2439  12538  58537
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          70532      33712   2.092    0.045 *
## distance_to_ohare     3592       5933   0.605    0.549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 26080 on 30 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.01207,    Adjusted R-squared:  -0.02086
## F-statistic: 0.3665 on 1 and 30 DF,  p-value: 0.5495
```

```r
# Model 2: Homeownership vs Distance
model_owner <- lm(pct_owner_occupied ~ distance_to_ohare, data = dunning_bg_acs_2022)
print("Homeownership vs Distance model:")
```

```
## [1] "Homeownership vs Distance model:"
```

```r
print(summary(model_owner))
```

```
##
## Call:
## lm(formula = pct_owner_occupied ~ distance_to_ohare, data = dunning_bg_acs_2022)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.677 -10.662   1.373  11.654  20.741
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       79.09585   19.21336   4.117 0.000241 ***
## distance_to_ohare  0.03268    3.39772   0.010 0.992385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.88 on 33 degrees of freedom
## Multiple R-squared:  2.803e-06,  Adjusted R-squared:  -0.0303
## F-statistic: 9.249e-05 on 1 and 33 DF,  p-value: 0.9924
```

```r
# Model 3: Racial composition vs Distance
model_white <- lm((race_whiteE/total_popE)*100 ~ distance_to_ohare, data = dunning_bg_acs_2022)
print("White population % vs Distance model:")
```

```
## [1] "White population % vs Distance model:"
```

```r
print(summary(model_white))
```

```
##
## Call:
## lm(formula = (race_whiteE/total_popE) * 100 ~ distance_to_ohare,
##     data = dunning_bg_acs_2022)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.494  -6.751  -1.347   6.907  27.394
##
## Coefficients:
```

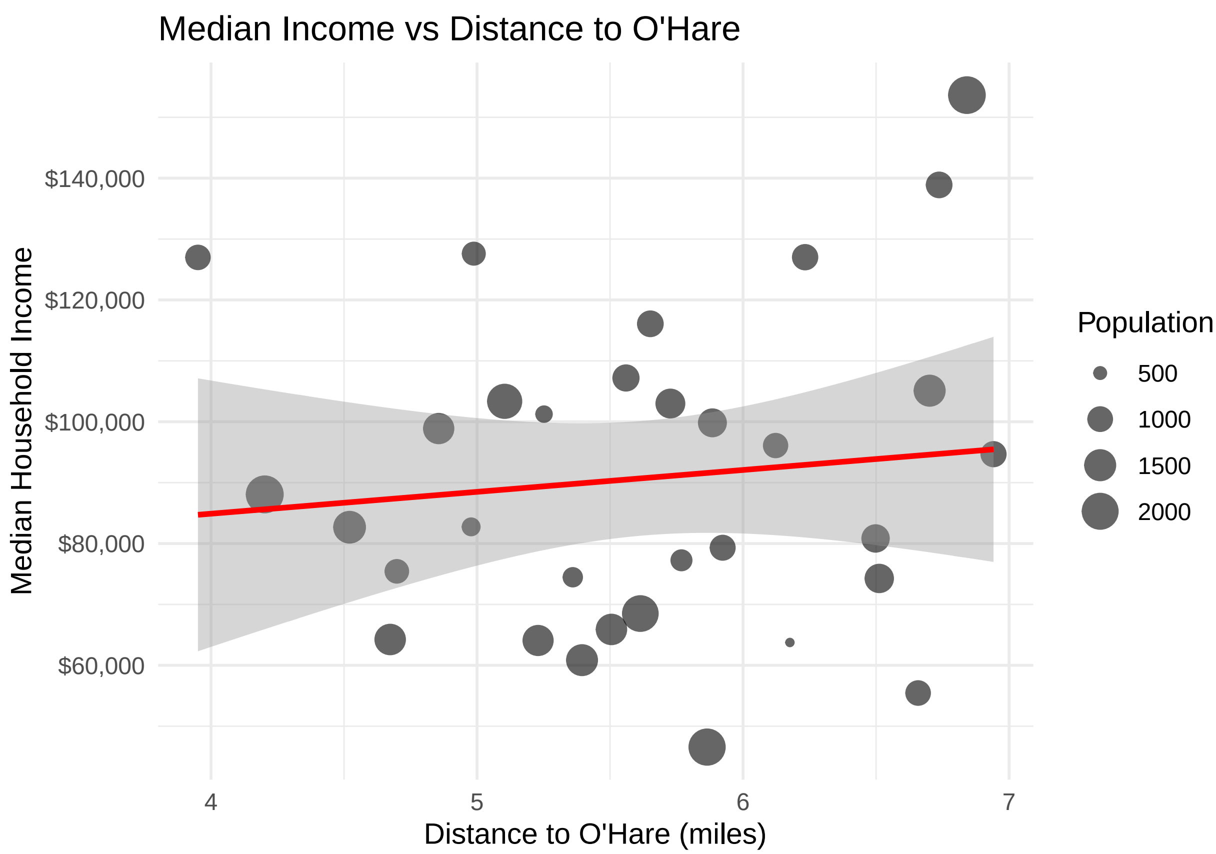
```
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         93.043     15.483   6.010 9.38e-07 ***
## distance_to_ohare   -4.540      2.738  -1.658    0.107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.8 on 33 degrees of freedom
## Multiple R-squared:  0.07692,    Adjusted R-squared:  0.04895
## F-statistic:  2.75 on 1 and 33 DF,  p-value: 0.1067
```

```
# Scatter plots with regression lines
scatter_data <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  mutate(
    pct_white = (race_whiteE / total_popE) * 100,
    pct_black = (race_blackE / total_popE) * 100,
    pct_hispanic = (hispanicE / total_popE) * 100
  )

# Income vs Distance scatter
p_scatter_income <- ggplot(scatter_data, aes(x = distance_to_ohare, y = med_incomeE)) +
  geom_point(aes(size = total_popE), alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  scale_y_continuous(labels = scales::dollar) +
  labs(title = "Median Income vs Distance to O'Hare",
       x = "Distance to O'Hare (miles)",
       y = "Median Household Income",
       size = "Population") +
  theme_minimal()

# Homeownership vs Distance scatter
p_scatter_owner <- ggplot(scatter_data, aes(x = distance_to_ohare, y = pct_owner_occupied)) +
  geom_point(aes(size = total_popE), alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  labs(title = "Homeownership vs Distance to O'Hare",
       x = "Distance to O'Hare (miles)",
       y = "Percentage Owner-Occupied (%)",
       size = "Population") +
  theme_minimal()

# White population vs Distance scatter
p_scatter_white <- ggplot(scatter_data, aes(x = distance_to_ohare, y = pct_white)) +
  geom_point(aes(size = total_popE), alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  labs(title = "White Population % vs Distance to O'Hare",
       x = "Distance to O'Hare (miles)",
       y = "Percentage White (%)",
       size = "Population") +
  theme_minimal()

# Print scatter plots individually (no combining)
print(p_scatter_income)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

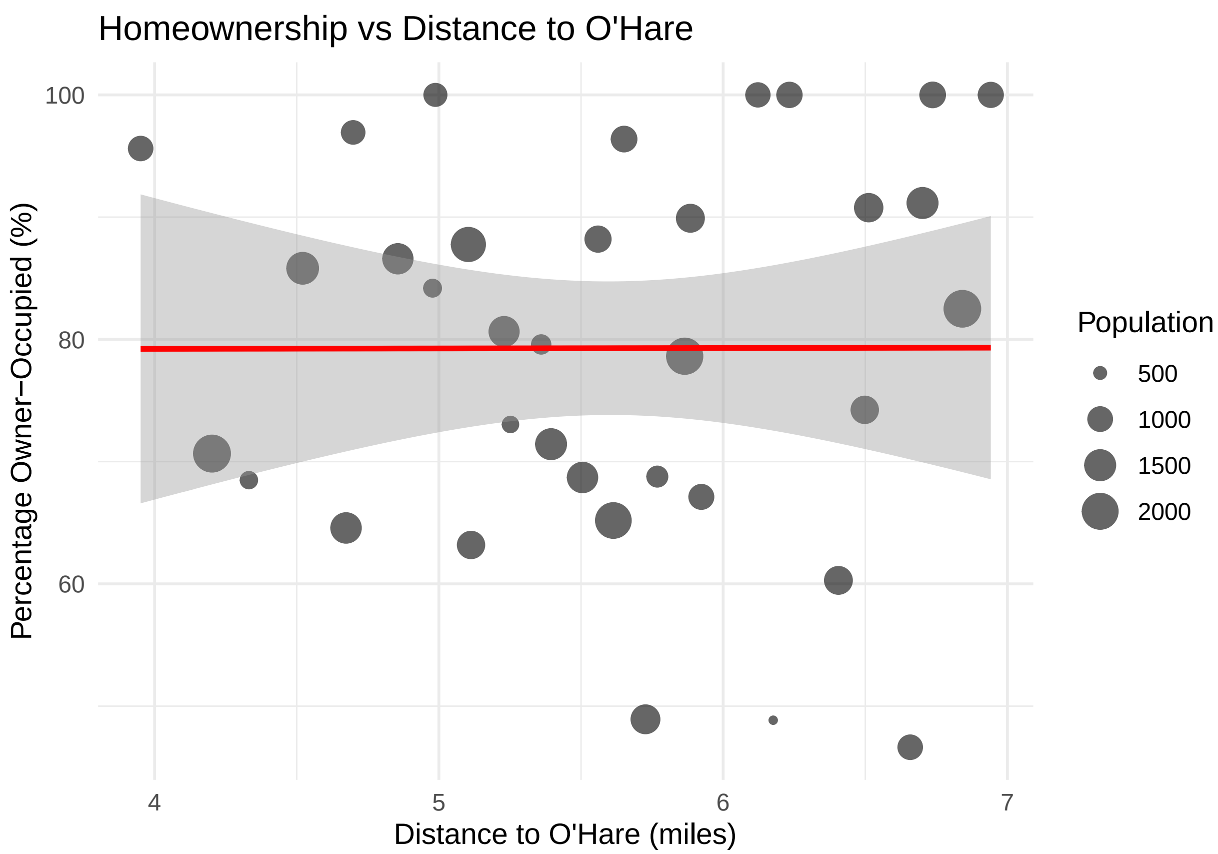
```
## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



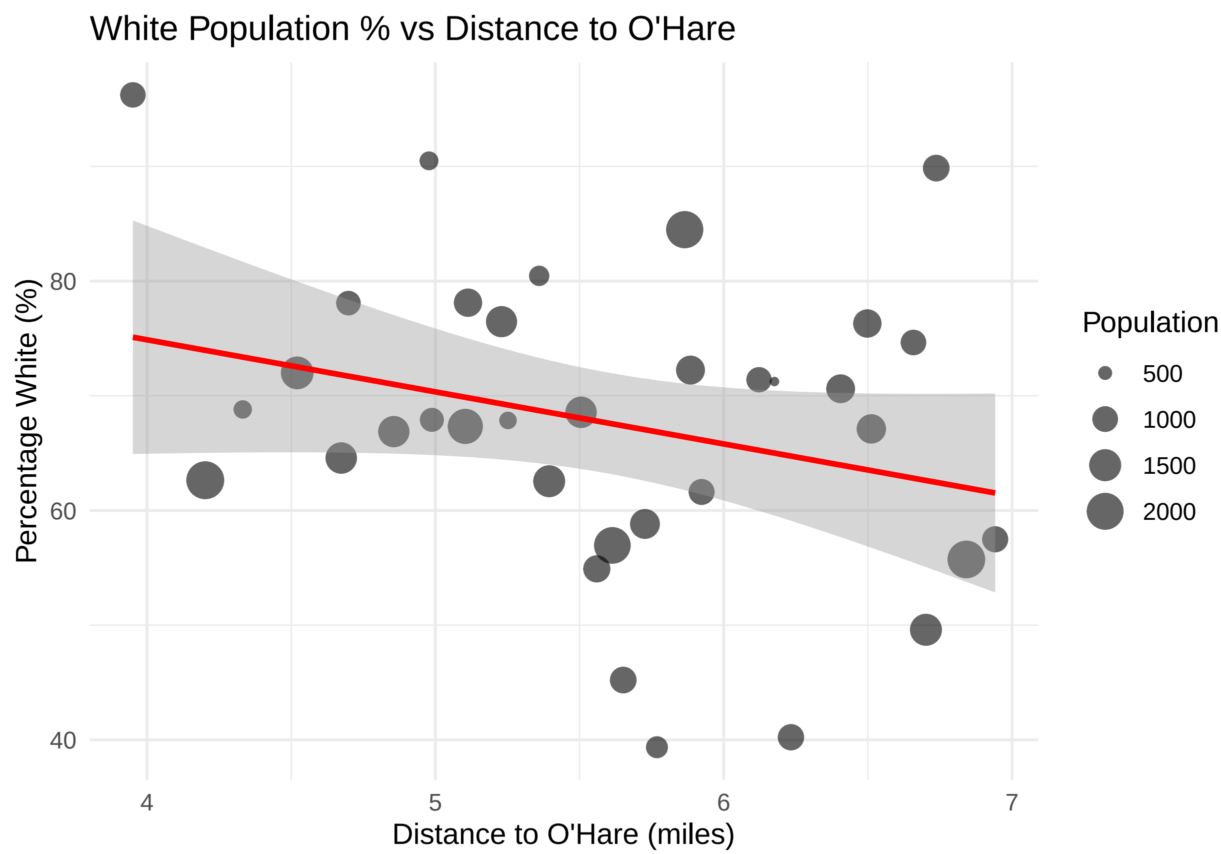

```
print(p_scatter_owner)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
print(p_scatter_white)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Integrate airport proximity into network analysis
analysis_data_with_distance <- study_bg_combined %>%
  st_drop_geometry() %>%
  mutate(
    pct_white = ifelse(total_popE > 0, (race_whiteE / total_popE) * 100, NA),
    pct_black = ifelse(total_popE > 0, (race_blackE / total_popE) * 100, NA),
    pct_hispanic = ifelse(total_popE > 0, (hispanicE / total_popE) * 100, NA),
    pct_owner = ifelse(total_housing > 0, (owner_occupiedE / total_housing) * 100, NA),
    income_scaled = scale(med_incomeE),
    distance_scaled = scale(distance_to_ohare)
  ) %>%
  select(GEOID, total_popE, pct_white, pct_black, pct_hispanic, income_scaled,
         hh_sizeE, pct_owner, distance_scaled, area_type) %>%
  drop_na()

# Update similarity network with airport proximity
scaled_data_with_distance <- analysis_data_with_distance %>%
  select(total_popE, pct_white, pct_black, pct_hispanic, income_scaled,
         hh_sizeE, pct_owner, distance_scaled) %>%
  scale() %>%
  as.data.frame()

rownames(scaled_data_with_distance) <- analysis_data_with_distance$GEOID

# Create updated network with airport proximity
bg_distances_updated <- dist(scaled_data_with_distance, method = "euclidean")
```

```r
bg_similarity_updated <- 1 / (1 + as.matrix(bg_distances_updated))

adjacency_matrix_updated <- matrix(0, nrow = nrow(bg_similarity_updated),
                                   ncol = ncol(bg_similarity_updated))

for(i in 1:nrow(bg_similarity_updated)) {
  similar_indices <- order(bg_similarity_updated[i,], decreasing = TRUE)[2:(k+1)]
  adjacency_matrix_updated[i, similar_indices] <- bg_similarity_updated[i, similar_indices]
  adjacency_matrix_updated[similar_indices, i] <- bg_similarity_updated[similar_indices, i]
}

bg_graph_updated <- graph_from_adjacency_matrix(adjacency_matrix_updated,
                                                mode = "undirected", weighted = TRUE)

V(bg_graph_updated)$GEOID <- analysis_data_with_distance$GEOID
V(bg_graph_updated)$pct_white <- analysis_data_with_distance$pct_white
V(bg_graph_updated)$pct_black <- analysis_data_with_distance$pct_black
V(bg_graph_updated)$pct_hispanic <- analysis_data_with_distance$pct_hispanic
V(bg_graph_updated)$income <- analysis_data_with_distance$income_scaled
V(bg_graph_updated)$distance <- analysis_data_with_distance$distance_scaled
V(bg_graph_updated)$area_type <- analysis_data_with_distance$area_type

# Detect communities in updated network
louvain_communities_updated <- cluster_louvain(bg_graph_updated)
V(bg_graph_updated)$louvain_community <- louvain_communities_updated$membership

cat("Updated network with airport proximity:\n")
```

```
## Updated network with airport proximity:
```

```r
cat("Vertices:", vcount(bg_graph_updated), "Edges:", ecount(bg_graph_updated), "\n")
```

```
## Vertices: 32 Edges: 123
```

```r
cat("Detected communities:", length(unique(louvain_communities_updated$membership)), "\n")
```

```
## Detected communities: 4
```

```r
cat("Modularity:", round(modularity(louvain_communities_updated), 3), "\n")
```

```
## Modularity: 0.32
```

```r
# Compare community detection with and without airport proximity
community_comparison_updated <- data.frame(
  GEOID = V(bg_graph_updated)$GEOID,
  Detected_Community = V(bg_graph_updated)$louvain_community,
  Official_Area = V(bg_graph_updated)$area_type
) %>%
  group_by(Detected_Community, Official_Area) %>%
  summarise(Count = n(), .groups = "drop")

print("Updated community detection with airport proximity:")
```

```
## [1] "Updated community detection with airport proximity:"
```

```r
print(community_comparison_updated)
```

```
## # A tibble: 4 x 3
##   Detected_Community Official_Area Count
##                <dbl> <chr>         <int>
## 1                  1 Dunning           9
## 2                  2 Dunning          10
## 3                  3 Dunning           7
## 4                  4 Dunning           6
```

```r
# Airport proximity impact summary
cat("\nAIRPORT PROXIMITY IMPACT ANALYSIS\n")
```

```
##
## AIRPORT PROXIMITY IMPACT ANALYSIS
```

```r
cat("===================================\n")
```

```
## ===================================
```

```r
# Key findings
distance_stats <- dunning_bg_acs_2022 %>%
  st_drop_geometry() %>%
  summarise(
    min_distance = min(distance_to_ohare, na.rm = TRUE),
    max_distance = max(distance_to_ohare, na.rm = TRUE),
    mean_distance = mean(distance_to_ohare, na.rm = TRUE),
    median_distance = median(distance_to_ohare, na.rm = TRUE)
  )

print("Distance statistics for Dunning block groups:")
```

```
## [1] "Distance statistics for Dunning block groups:"
```

```r
print(distance_stats)
```

```
##   min_distance max_distance mean_distance median_distance
## 1     3.950809     6.941489      5.599308        5.613795
```

```r
# Airport proximity conclusions
cat("\nCONCLUSIONS ABOUT AIRPORT PROXIMITY EFFECTS:\n")
```

```
##
## CONCLUSIONS ABOUT AIRPORT PROXIMITY EFFECTS:
```

```r
if(correlation_matrix["distance_to_ohare", "med_incomeE"] > 0.1) {
  cat("• Positive correlation found: Income increases with distance from airport\n")
} else if(correlation_matrix["distance_to_ohare", "med_incomeE"] < -0.1) {
  cat("• Negative correlation found: Income decreases with distance from airport\n")
} else {
  cat("• Weak correlation between income and airport distance\n")
}
```

```
## • Positive correlation found: Income increases with distance from airport
```

```r
if(correlation_matrix["distance_to_ohare", "pct_owner_occupied"] > 0.1) {
  cat("• Homeownership rates tend to be higher further from the airport\n")
} else if(correlation_matrix["distance_to_ohare", "pct_owner_occupied"] < -0.1) {
  cat("• Homeownership rates tend to be higher closer to the airport\n")
} else {
  cat("• Weak relationship between homeownership and airport distance\n")
}
```

```
## • Weak relationship between homeownership and airport distance
```

```r
# Save airport proximity data for further analysis
write_csv(distance_analysis, "dunning_airport_proximity_analysis.csv")

cat("\nAirport proximity analysis completed and saved.\n")
```

```
##
## Airport proximity analysis completed and saved.
```

```r
library(scales)  # for rescale()
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:viridis':
##
##     viridis_pal
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
# Ensure required variables exist:
# dunning_bg_acs_2022 must have:
#  - distance_to_ohare (miles)
#  - med_incomeE (median income)
#  - pct_owner_occupied (%)
```

```r
# 1. Create standardized (0-1) versions of the three variables
library(dplyr)
library(ggplot2)
library(sf)
library(scales)

# 1. Create SAFE scaled variables (no NAs, no out-of-range values)
dunning_composite <- dunning_bg_acs_2022 %>%
  mutate(
    income_scaled = rescale(med_incomeE, to = c(0, 1), na.rm = TRUE),
    owner_scaled  = rescale(pct_owner_occupied, to = c(0, 1), na.rm = TRUE),

    # distance: flip so closer = more intense
    dist_scaled_raw = rescale(distance_to_ohare, to = c(0, 1), na.rm = TRUE),
    dist_scaled     = 1 - dist_scaled_raw,

    # Replace all NA values with 0 (or a neutral mid value if you prefer)
    income_scaled = ifelse(is.na(income_scaled), 0, income_scaled),
    owner_scaled  = ifelse(is.na(owner_scaled), 0, owner_scaled),
    dist_scaled   = ifelse(is.na(dist_scaled), 0, dist_scaled),

    # Ensure all values are within 0-1 to avoid rgb() crash
    income_scaled = pmin(pmax(income_scaled, 0), 1),
    owner_scaled  = pmin(pmax(owner_scaled, 0), 1),
    dist_scaled   = pmin(pmax(dist_scaled, 0), 1),

    # Composite RGB color
    composite_col = rgb(income_scaled, owner_scaled, dist_scaled)
  )

# 2. Single composite map for all 3 variables
p_composite <- ggplot() +
  geom_sf(data = dunning_sf, fill = NA, color = "black", size = 0.8) +
  geom_sf(data = dunning_composite,
          aes(fill = composite_col),
          color = "white",
          alpha = 0.95,
          size = 0.1) +
  geom_sf(data = ohare_airport,
          color = "yellow",
          fill  = "yellow",
          size = 3,
          shape = 23) +
  scale_fill_identity(
    name   = "Composite (R,G,B)",
    guide  = "none"
  ) +
  labs(
    title    = "Composite Map: Income, Homeownership, and Airport Proximity (Dunning)",
    subtitle = "Red = higher income, Green = higher homeownership, Blue = closer to O'Hare"
  ) +
  theme_void() +
  theme(
```

```r
    plot.title    = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5)
  )

print(p_composite)
```