FOIA Document OCR Processing

Processing Murphy & Tong FOIA Documents about State DNA Database Racial Composition

Tina Lasisi | Edit: João P. Donadio

2025-10-04

Table of Contents

1 Overview

This document details the processing of Freedom of Information Act (FOIA) responses from seven U.S. states regarding the demographic composition of their State DNA Index System (SDIS) databases. These responses were obtained by Professor Erin Murphy (NYU Law) in 2018 as part of research on racial disparities in DNA databases.

2 Materials and Methods

2.1 Data Sources

2.1.1 Raw FOIA Responses

The original FOIA responses are stored in two formats:

- **PDFs**: `raw/foia_pdfs/` - Original scanned documents
- **HTML**: `raw/foia_html/` - OCR'd versions for easier extraction

```r
# List of required packages
required_packages <- c(
  "tidyverse",    # Data manipulation and visualization
  "here",         # File path management
  "knitr",        # Dynamic report generation
  "kableExtra",   # Enhanced table formatting
  "ggplot2",      # Data visualization
  "patchwork",    # Plot composition and layout
  "scales",       # Axis scaling and formatting
  "tidyr",        # Data tidying and reshaping
  "tibble",       # Modern data frames
  "flextable",    # Advanced table formatting
  "DT",           # Interactive tables
  "cowplot",      # Plotting composition
  "sf",           # Simple Features for spatial data
  "usmap"        # Mapping US states
)

# Function to install missing packages
install_missing <- function(packages) {
  for (pkg in packages) {
    if (!requireNamespace(pkg, quietly = TRUE)) {
      message(paste("Installing missing package:", pkg))
      install.packages(pkg, dependencies = TRUE)
    }
  }
}

# Install any missing packages
install_missing(required_packages)

# Load all packages
suppressPackageStartupMessages({
  library(tidyverse)
  library(here)
  library(knitr)
  library(kableExtra)
  library(ggplot2)
  library(patchwork)
  library(scales)
  library(tidyr)
  library(tibble)
  library(flextable)
  library(cowplot)
  library(sf)
  library(usmap)
})
```

```r
# Verify all packages loaded successfully
loaded_packages <- sapply(required_packages, require, character.only = TRUE)
if (all(loaded_packages)) {
  message("All packages loaded successfully!")
} else {
  warning("The following packages failed to load: ",
          paste(names(loaded_packages)[!loaded_packages], collapse = ", "))
}

# Display options
options(tibble.width = Inf)
options(dplyr.summarise.inform = FALSE)

# Path to per-state files (run notebook from analysis/)
base_dir <- here("..")
per_state <- here("data", "foia", "intermediate")

# Discover available per-state CSV files
state_files <- list.files(per_state, pattern = "*_foia_data\\.csv$", full.names = TRUE)

if (length(state_files) == 0) {
  stop(paste("No per-state FOIA files found in", per_state, ". Check the folder path."))
}

stem_to_state <- function(stem) {
  toks <- str_split(stem, "_")[[1]]
  if ("foia" %in% toks) {
    toks <- toks[1:(which(toks == "foia") - 1)]
  }
  paste(tools::toTitleCase(toks), collapse = " ")
}

states_available <- map_chr(basename(state_files), ~ stem_to_state(str_remove(.x, "_foia_data\\

cat(paste("  Found", length(state_files), "per-state files:\n"))
```

```
  Found 7 per-state files:
```

```r
for (s in states_available) {
  cat(paste("  •", s, "\n"))
}
```

```
  • California
  • Florida
  • Indiana
  • Maine
```

- `Nevada`
- `South Dakota`
- `Texas`

```
# Initialize empty containers for the loop that follows
foia_combined <- tibble()
foia_state_metadata <- list()
```

2.2 Processing workflow

For transparency, each state file is processed independently then merged into a single **combined long-format table (`foia_combined`)**:

1. **Load one file per state** from `data/foia/intermediate/`.

2. **Append** its rows to `foia_combined`. A parallel dataframe, **`foia_state_metadata`**, records what each state reported (counts, percentages, which categories) and any state-specific characteristics (e.g. Nevada's "flags" terminology).

3. **Quality-check each state**:

   - verify that race and gender percentages sum to 100 % when provided,
   - confirm that demographic counts sum to the state's reported total profiles,
   - **calculate** any missing counts or percentages and tag those rows `value_source = "calculated"`.

4. **Save outputs**

   - `data/foia/final/foia_data_clean.csv` — the fully combined tidy table with both reported and calculated values,
   - `data/foia/intermediate/foia_state_metadata.csv` — one row per state summarising coverage and caveats. After QC passes, freeze **`foia_data_clean.csv`** to `data/v1.0/FOIA_demographics.csv`.

2.3 Helper Functions

The functions below perform each transformation required for harmonizing the state-level FOIA tables.

2.3.1 Data Processing Helper Functions Reference

Function	Definition	Parameters
`load_state()`	Loads and preprocesses state FOIA data files, handling numeric conversion and validation	`path`: File path to state CSV
`enhanced_glimpse()`	Provides an enhanced data overview with column types, missing values, unique counts, and unique values	`df`: Input dataframe
`fill_demographic_gaps()`	Fills missing gender counts and adds Unknown race category when totals permit calculation	`df`: Input dataframe
`add_combined()`	Creates Combined offender type by summing Convicted Offender and Arrestee counts when missing	`df`: Input dataframe
`add_percentages()`	Derives percentage values from counts for all demographic categories	`df`: Input dataframe
`counts_consistent()`	Verifies that demographic counts sum to total_profiles for each offender type	`df`: Input dataframe
`percentages_consistent()`	Verifies that percentages sum to 100 ± 0.5% for each category	`df`: Input dataframe
`report_status()`	Reports what data types (counts/percentages/both) are available for a category	`df`: Input dataframe, `category`: race or gender
`verify_category_totals()`	Compares demographic sums against reported totals and shows differences	`df`: Input dataframe
`verify_percentage_consistency()`	Compares reported vs calculated percentages for consistency	`df_combined`: Combined dataframe, `state_name`: State name
`calculate_combined_totals()`	Calculates Combined totals by summing across offender types	`df`: Input dataframe, `state_name`: State name
`calculate_percentages()`	Calculates percentages from counts for demographic categories	`df_combined`: Combined dataframe, `state_name`: State name
`calculate_counts_from_percentages()`	Calculates counts from percentages for demographic categories	`df_combined`: Combined dataframe, `state_name`: State name
`standardize_offender_types()`	Standardizes offender type names to consistent terminology	`df`: Input dataframe

Function	Definition	Parameters
`prepare_state_for_combined()`	Prepares state data for inclusion in combined dataset with proper columns	`df`: Input dataframe, `state_name`: State name
`format_compact()`	Formats large numbers with K/M suffixes for readability	`x`: Numeric value
`create_pie_chart()`	Creates pie charts for specific demographic categories	`data`: Input data, `offender_type`, `category`, `value_type`, `title`, `show_values`
`create_state_visualizations()`	Creates comprehensive pie chart visualizations for all metrics	`df_combined`: Combined dataframe, `state_name`: State name
`create_demographic_bar_charts()`	Creates side-by-side bar charts for gender and race distributions	`df_combined`: Combined dataframe, `state_name`: State name
`add_state_metadata()`	Creates and appends a metadata record capturing state data characteristics including available offender types, demographic categories, data formats, and special features	`df`: Input dataframe, `state_name`: State name
`update_state_metadata()`	Modifies existing state metadata to update QC results (count/percentage consistency) and append validation notes	`state_name`: State name, `counts_ok`: Count consistency flag, `percentages_ok`: Percentage consistency flag, `notes_text`: Additional notes

```
## Helper functions setup --

# Columns retained from every raw table
COLS_NEEDED <- c("state", "offender_type", "variable_category",
                 "variable_detailed", "value", "value_type")

# -------------------------------------------------------------------
# 1. Load and preprocess state files
# -------------------------------------------------------------------
load_state <- function(path) {
  "
  Read a *_foia_data.csv* file, enforce column order,
  and convert <1 to 0.5 so that trace counts are retained.
  Execution halts if non-numeric values remain.
  "
  df <- read_csv(path, show_col_types = FALSE)
  if (!"state" %in% colnames(df)) {
```

```r
    df <- df %>%
      mutate(state = str_remove(basename(path), "_foia_data\\.csv") %>%
        str_replace_all("_", " ") %>%
        tools::toTitleCase())
  }
  df <- df %>% select(all_of(COLS_NEEDED))
  df$value_source <- "reported"

  df <- df %>%
    mutate(value = ifelse(value == "<1", 0.5, value),
           value = as.numeric(value))

  nonnumeric <- df %>% filter(is.na(value))
  if (nrow(nonnumeric) > 0) {
    cat(paste("**Non-numeric rows in", basename(path), "; please amend**\n"))
    print(nonnumeric)
    stop("Numeric coercion failure")
  }
  return(df)
}


# ------------------------------------------------------------------
# 2. Enhanced glimpse
# ------------------------------------------------------------------
# Display data types for each column with unique values
enhanced_glimpse <- function(df) {
  glimpse_data <- data.frame(
    Column = names(df),
    Type = sapply(df, function(x) paste(class(x), collapse = ", ")),
    Rows = nrow(df),
    Missing = sapply(df, function(x) sum(is.na(x))),
    Unique = sapply(df, function(x) length(unique(x))),
    Unique_Values = sapply(df, function(x) {
      unique_vals <- unique(x)
      if (length(unique_vals) > 10) {
        paste(encodeString(as.character(unique_vals[1:10])), collapse = ", ", "...")
      } else {
        paste(encodeString(as.character(unique_vals)), collapse = ", ")
      }
    })
  )

  ft <- flextable(glimpse_data) %>%
    theme_zebra() %>%
    set_caption(paste("Enhanced Data Glimpse:", deparse(substitute(df)))) %>%
    autofit() %>%
    align(align = "left", part = "all") %>%
    colformat_num(j = c("Rows", "Missing", "Unique"), big.mark = "") %>%
    bg(j = "Missing", bg = function(x) ifelse(x > 0, "#FFF3CD", "transparent")) %>%
    bg(j = "Unique", bg = function(x) ifelse(x == 1, "#FFF3CD", "transparent")) %>%
    add_footer_lines(paste("Data frame dimensions:", nrow(df), "rows ×", ncol(df), "columns"))
    fontsize(size = 10, part = "all") %>%
    set_table_properties(layout = "autofit", width = 1)

  return(ft)
}

# ------------------------------------------------------------------
# 3. Fill missing Male counts and Unknown race counts
# ------------------------------------------------------------------
fill_demographic_gaps <- function(df) {
  "If exactly one gender or the Unknown race category is absent and
  totals permit a residual, calculate and insert the missing count.
  "
  inserts <- list()

  for (ot in unique(df$offender_type)) {
    tot <- df %>%
      filter(offender_type == ot,
             variable_category == "total",
             variable_detailed == "total_profiles",
             value_type == "count")

    if (nrow(tot) == 0) next

    total <- tot$value[1]

    # gender residual ----------------------------------------------
    g <- df %>%
      filter(offender_type == ot,
             variable_category == "gender",
             value_type == "count")

    missing_gender <- setdiff(c("Male", "Female"), unique(g$variable_detailed))
    if (nrow(g) == 1 && length(missing_gender) == 1) {
      inserts[[length(inserts) + 1]] <- tibble(
        state = df$state[1],
        offender_type = ot,
        variable_category = "gender",
        variable_detailed = missing_gender,
        value = total - sum(g$value),
        value_type = "count",
        value_source = "calculated"
```

```r
      )
    }

    # race residual -----------------------------------------------
    r <- df %>%
      filter(offender_type == ot,
             variable_category == "race",
             value_type == "count")

    if (nrow(r) > 0 && !"Unknown" %in% r$variable_detailed) {
      gap <- total - sum(r$value)
      if (gap > 0) {
        inserts[[length(inserts) + 1]] <- tibble(
          state = df$state[1],
          offender_type = ot,
          variable_category = "race",
          variable_detailed = "Unknown",
          value = gap,
          value_type = "count",
          value_source = "calculated"
        )
      }
    }
  }

  if (length(inserts) > 0) {
    df <- bind_rows(df, bind_rows(inserts))
  }
  return(df)
}

# --------------------------------------------------------------------
# 4. Construct Combined offender type if absent (add_combined)
# --------------------------------------------------------------------
add_combined <- function(df) {
  "
  When a state reports Convicted Offender and Arrestee counts but
  omits Combined, create a Combined block by summing the two.
  "
  if ("Combined" %in% df$offender_type) return(df)

  required <- c("Convicted Offender", "Arrestee")
  if (!all(required %in% df$offender_type)) return(df)  # cannot construct

  summed <- df %>%
    filter(value_type == "count") %>%
    group_by(variable_category, variable_detailed, value_type) %>%
    summarise(value = sum(value), .groups = "drop") %>%
    mutate(state = df$state[1],
           offender_type = "Combined",
           value_source = "calculated")

  return(bind_rows(df, summed))
}

# ------------------------------------------------------------------
# 5. Derive percentages wherever only counts exist (add_percentages)
# ------------------------------------------------------------------
add_percentages <- function(df) {
  "
  Ensure that every gender and race row has both count and percentage
  values, derived from the offender-type total if necessary.
  "
  totals <- df %>%
    filter(variable_category == "total",
           variable_detailed == "total_profiles",
           value_type == "count") %>%
    select(offender_type, value) %>%
    deframe()

  need_pct <- df %>%
    filter(value_type == "count",
           variable_category != "total")

  new_pct_rows <- need_pct %>%
    rowwise() %>%
    mutate(has_percentage = nrow(df %>%
           filter(state == state,
                  offender_type == offender_type,
                  variable_category == variable_category,
                  variable_detailed == variable_detailed,
                  value_type == "percentage"))) %>%
    filter(has_percentage == 0) %>%
    mutate(value = round(value / totals[offender_type] * 100, 2),
           value_type = "percentage",
           value_source = "calculated") %>%
    select(-has_percentage)

  if (nrow(new_pct_rows) > 0) {
    df <- bind_rows(df, new_pct_rows)
  }
  return(df)
}
```

```
# ------------------------------------------------------------------
# 6. Counts consistency checks
# ------------------------------------------------------------------
counts_consistent <- function(df) {
  "
  Verifies that demographic counts sum to total_profiles for each
  offender type and category.
  "
  demo_sum <- df %>%
    filter(value_type == "count",
           variable_category != "total") %>%
    group_by(offender_type, variable_category) %>%
    summarise(sum_value = sum(value), .groups = "drop")

  totals <- df %>%
    filter(variable_category == "total",
           variable_detailed == "total_profiles",
           value_type == "count") %>%
    select(offender_type, value)

  merged <- demo_sum %>%
    left_join(totals, by = "offender_type") %>%
    mutate(diff = abs(sum_value - value))

  all(merged$diff < 1e-6)
}

# ------------------------------------------------------------------
# 7. Percentage consistency checks
# ------------------------------------------------------------------

percentages_consistent <- function(df) {
  "
  Verifies that derived or reported percentages sum to 100 ± 0.5 %.
  "
  result <- df %>%
    filter(value_type == "percentage") %>%
    group_by(offender_type, variable_category) %>%
    summarise(sum_value = sum(value), .groups = "drop") %>%
    mutate(consistent = abs(sum_value - 100) <= 0.5)

  all(result$consistent)
}


# ------------------------------------------------------------------
# 8. Report status for each category
```

```r
# ------------------------------------------------------------------

# Define columns needed for foia_combined
report_status <- function(df, category) {
  values <- unique(df$value_type[df$variable_category == category])

  if (all(c("count", "percentage") %in% values)) {
    return("both")
  } else if ("count" %in% values) {
    return("counts")
  } else if ("percentage" %in% values) {
    return("percentages")
  } else {
    return("neither")
  }
}

# ------------------------------------------------------------------
# 9. Verify category totals
# ------------------------------------------------------------------

verify_category_totals <- function(df) {
  # 1 pull total_profiles per offender_type
  total_map <- df %>%
    filter(variable_category == "total",
           variable_detailed == "total_profiles") %>%
    select(offender_type, value) %>%
    deframe() %>%
    as.list()

  # 2 sum counts by offender_type and variable_category
  demo_sum <- df %>%
    filter(value_type == "count",
           variable_category != "total") %>%
    group_by(offender_type, variable_category) %>%
    summarise(sum_counts = sum(value, na.rm = TRUE), .groups = "drop")

  # 3 attach total_profiles and compute difference
  demo_sum <- demo_sum %>%
    mutate(total_profiles = map_dbl(offender_type, ~total_map[[.x]]),
           difference = total_profiles - sum_counts)

  # tidy columns order
  demo_sum %>%
    select(offender_type, variable_category, total_profiles,
           sum_counts, difference)
}
```

```r
# ------------------------------------------------------------------
# 10. Calculate Combined totals
# ------------------------------------------------------------------

calculate_combined_totals <- function(df, state_name) {
  # Get all counts
  counts_df <- df %>%
    filter(value_type == 'count') %>%
    mutate(value_source = 'calculated')

  # Group by variable_category and variable_detailed, sum values
  combined_sums <- counts_df %>%
    group_by(variable_category, variable_detailed) %>%
    summarise(value = sum(value, na.rm = TRUE), .groups = "drop")

  # Create Combined rows
  combined_rows <- combined_sums %>%
    mutate(state = state_name,
           offender_type = 'Combined',
           value_type = 'count',
           value_source = 'calculated') %>%
    select(all_of(COLS_NEEDED), value_source)

  return(combined_rows)
}

# ------------------------------------------------------------------
# 11. Calculate percentages from counts
# ------------------------------------------------------------------

calculate_percentages <- function(df_combined, state_name) {
  # Get total profiles for each offender type
  totals_map <- df_combined %>%
    filter(state == state_name,
           variable_category == 'total',
           variable_detailed == 'total_profiles') %>%
    select(offender_type, value) %>%
    deframe() %>%
    as.list()

  percentage_rows <- list()

  for (offender_type in names(totals_map)) {
    total <- totals_map[[offender_type]]

    # Get all demographic counts
    demo_data <- df_combined %>%
      filter(state == state_name,
             offender_type == !!offender_type,
             variable_category %in% c('gender', 'race'),
             value_type == 'count')

    if (nrow(demo_data) > 0) {
      # Calculate percentage for each
      demo_percentages <- demo_data %>%
        mutate(value = round((value / total) * 100, 2),
               value_type = 'percentage',
               value_source = 'calculated') %>%
        select(all_of(COLS_NEEDED), value_source)

      percentage_rows <- c(percentage_rows, list(demo_percentages))
    }
  }

  bind_rows(percentage_rows)
}

# ------------------------------------------------------------------
# 12. Calculate counts from percentages
# ------------------------------------------------------------------

calculate_counts_from_percentages <- function(df_combined, state_name) {
  # Get total profiles for each offender type
  totals_map <- df_combined %>%
    filter(state == state_name,
           variable_category == 'total',
           variable_detailed == 'total_profiles') %>%
    select(offender_type, value) %>%
    deframe() %>%
    as.list()

  count_rows <- list()

  for (offender_type in names(totals_map)) {
    total <- totals_map[[offender_type]]

    # Get all demographic percentages
    demo_data <- df_combined %>%
      filter(state == state_name,
             offender_type == !!offender_type,
             variable_category %in% c('gender', 'race'),
             value_type == 'percentage')

    if (nrow(demo_data) > 0) {
      # Calculate count for each
      demo_counts <- demo_data %>%
        mutate(value = as.integer(round(total * (value / 100))),
               value_type = 'count',
               value_source = 'calculated') %>%
        select(all_of(COLS_NEEDED), value_source)

      count_rows <- c(count_rows, list(demo_counts))
    }
  }

  bind_rows(count_rows)
}

# ------------------------------------------------------------------
# 13. Standardize offender types
# ------------------------------------------------------------------

standardize_offender_types <- function(df) {
  replacements <- c(
    'Offenders' = 'Convicted Offender',
    'Convicted offenders' = 'Convicted Offender',
    'Arrested offender' = 'Arrestee',
    'All' = 'Combined'
  )

  df %>%
    mutate(offender_type = recode(offender_type, !!!replacements))
}

# ------------------------------------------------------------------
# 14. Prepare state data for combined dataset
# ------------------------------------------------------------------

prepare_state_for_combined <- function(df, state_name) {

  df_prepared <- df %>%
    select(any_of(COLS_NEEDED), value_source)

  df_prepared <- df_prepared %>%
      mutate(value_source = case_when(
        is.na(value_source) ~ "calculated",
        value_source == "" ~ "calculated",
        TRUE ~ value_source
      ))
  df_prepared
}

# ------------------------------------------------------------------
# 15. Compare reported vs calculated percentages
# ------------------------------------------------------------------

verify_percentage_consistency <- function(df_combined, state_name) {
  state_data <- df_combined %>%
    filter(state == state_name)

  # Get all offender types that have both counts and percentages
  offender_types <- unique(state_data$offender_type)

  consistency_results <- list()

  for (offender_type in offender_types) {
    offender_data <- state_data %>%
      filter(offender_type == !!offender_type)

    # Check if we have both reported and calculated percentages
    for (category in c('gender', 'race')) {
      reported_pcts <- offender_data %>%
        filter(variable_category == !!category,
               value_type == 'percentage',
               value_source == 'reported')

      calculated_pcts <- offender_data %>%
        filter(variable_category == !!category,
               value_type == 'percentage',
               value_source == 'calculated')

      if (nrow(reported_pcts) > 0 && nrow(calculated_pcts) > 0) {
        # Compare each demographic value
        for (i in 1:nrow(reported_pcts)) {
          rep_row <- reported_pcts[i, ]
          calc_match <- calculated_pcts %>%
            filter(variable_detailed == rep_row$variable_detailed)

          if (nrow(calc_match) > 0) {
            diff <- abs(rep_row$value - calc_match$value[1])
            consistency_results <- c(consistency_results, list(data.frame(
              offender_type = offender_type,
              category = category,
              variable = rep_row$variable_detailed,
              reported = rep_row$value,
              calculated = calc_match$value[1],
              difference = diff,
              consistent = diff < 0.5
            )))
          }
        }
      }
    }
  }

  if (length(consistency_results) > 0) {
    consistency_df <- bind_rows(consistency_results)
    cat(paste0("\nPercentage consistency check for ", state_name, ":\n"))
    cat(paste0("All values consistent: ", all(consistency_df$consistent), "\n"))

    if (!all(consistency_df$consistent)) {
      cat("\nInconsistent values:\n")
      print(consistency_df %>% filter(!consistent))
    }

    return(all(consistency_df$consistent))
  } else {
    # No comparison possible - state only has one type of data
    return(TRUE)
  }
}
# ------------------------------------------------------------------
# 16. Add compact formatting for large numbers
# ------------------------------------------------------------------

format_compact <- function(x) {
  sapply(x, function(single_x) {
    if (single_x >= 1000000) {
      if (single_x/1000000 == as.integer(single_x/1000000)) {
        return(paste0(as.integer(single_x/1000000), "M"))
      } else {
        return(paste0(round(single_x/1000000, 1), "M"))
      }
    } else if (single_x >= 1000) {
      return(paste0(as.integer(single_x/1000), "k"))
    } else {
      return(paste0(as.integer(single_x)))
    }
  })
}

# ------------------------------------------------------------------
# 17. Pie chart creation function
# -----------------------------------------------------------------

create_pie_chart <- function(data, offender_type, category, value_type, title, show_values = F
  chart_data <- data %>%
    filter(offender_type == !!offender_type,
           variable_category == !!category,
           value_type == !!value_type)

  # Check if we have data after filtering
  if (nrow(chart_data) == 0) {
    plot.new()
    title(main = title, cex.main = 0.9)
    text(0.5, 0.5, "No data", cex = 0.8)
    return()
  }

  # AGGREGATE DATA TO REMOVE DUPLICATES - KEY FIX
  chart_data <- chart_data %>%
    group_by(variable_detailed) %>%
    summarise(value = sum(value, na.rm = TRUE)) %>%
    ungroup()

  # Ensure consistent categories
  if (category == 'gender') {
    all_genders <- data.frame(variable_detailed = c('Male', 'Female', 'Unknown'))
    chart_data <- chart_data %>%
      right_join(all_genders, by = "variable_detailed") %>%
      mutate(value = ifelse(is.na(value), 0, value)) %>%
      arrange(factor(variable_detailed, levels = c('Male', 'Female', 'Unknown')))
  } else if (category == 'race') {
    all_races <- data.frame(variable_detailed = c('White', 'Black', 'Hispanic',
                                                  'Asian', 'Native American', 'Other', 'Unknown
    chart_data <- chart_data %>%
      right_join(all_races, by = "variable_detailed") %>%
      mutate(value = ifelse(is.na(value), 0, value)) %>%
      arrange(factor(variable_detailed, levels = c('White', 'Black', 'Hispanic',
                                                   'Asian', 'Native American', 'Other', 'Unknow
  }

  # Filter out zero values and ensure we have data
  chart_data <- chart_data %>% filter(value > 0)

  if (nrow(chart_data) == 0) {
    plot.new()
    title(main = title, cex.main = 0.9)
    text(0.5, 0.5, "No data", cex = 0.8)
    return()
```

```
}

# Define colors
if (category == 'gender') {
  colors <- c('Male' = '#4E79A7', 'Female' = '#E15759', 'Unknown' = '#BAB0AC')
} else {
  colors <- c('White' = '#4E79A7', 'Black' = '#F25E2B', 'Hispanic' = '#E14759',
             'Asian' = '#76B7B2', 'Native American' = '#59A14F',
             'Other' = '#9C755F', 'Unknown' = '#BAB0AC')
}

# Filter colors to only include categories present in data
pie_colors <- colors[names(colors) %in% chart_data$variable_detailed]

# Calculate percentages
total_value <- sum(chart_data$value)
chart_data <- chart_data %>%
  mutate(pct = value / total_value * 100)

# Create labels based on value_type and show_values
if (show_values && value_type == 'count') {
  chart_data <- chart_data %>%
    mutate(base_label = paste0(variable_detailed, "\n(", format(value, big.mark = ","), ")")
} else if (value_type == 'percentage') {
  chart_data <- chart_data %>%
    mutate(base_label = paste0(variable_detailed, "\n(", round(value, 1), "%)"))
} else {
  chart_data <- chart_data %>%
    mutate(base_label = variable_detailed)
}

# Only show labels for slices >= 3%, otherwise empty string
chart_data <- chart_data %>%
  mutate(label = ifelse(pct >= 3, base_label, ""))

# Create the pie chart
pie(chart_data$value,
    labels = chart_data$label,
    main = title,
    col = pie_colors,
    cex.main = 0.9,
    cex = 0.8)

# Add legend for small slices
small_slices <- chart_data %>% filter(pct < 3)
if (nrow(small_slices) > 0) {
  legend_labels <- paste0(small_slices$variable_detailed, " (", round(small_slices$pct, 1), 
    legend_colors <- pie_colors[small_slices$variable_detailed]

    legend("bottomright",
           legend = legend_labels,
           fill = legend_colors,
           cex = 0.7,
           bty = "n")
  }
}

# -------------------------------------------------------------------
# 18. State visualizations with 2 pies per row
# -------------------------------------------------------------------

create_state_visualizations <- function(df_combined, state_name) {
  state_data <- df_combined %>% filter(state == state_name)

  offender_types <- sort(unique(state_data$offender_type))
  plots <- list()

  for (offender_type in offender_types) {
    plots <- c(plots, list(
      create_pie_chart(state_data, offender_type, 'gender', 'count',
                       paste(offender_type, "Gender Counts"), TRUE),
      create_pie_chart(state_data, offender_type, 'gender', 'percentage',
                       paste(offender_type, "Gender Percentages")),
      create_pie_chart(state_data, offender_type, 'race', 'count',
                       paste(offender_type, "Race Counts"), TRUE),
      create_pie_chart(state_data, offender_type, 'race', 'percentage',
                       paste(offender_type, "Race Percentages"))
    ))
  }
}

# -------------------------------------------------------------------
# 19. Demographic bar chart function
# -------------------------------------------------------------------

create_demographic_bar_charts <- function(df_combined, state_name) {
  state_data <- df_combined %>%
    filter(state == state_name)

  # Get offender types and ensure Combined is last
  offender_types <- state_data %>%
    filter(value_type == 'count') %>%
    pull(offender_type) %>%
    unique() %>%
    sort()

  if ('Combined' %in% offender_types) {
    offender_types <- c(setdiff(offender_types, 'Combined'), 'Combined')
  }

  # Color palettes
  gender_colors <- c('Male' = '#4E79A7', 'Female' = '#E15759', 'Unknown' = '#BAB0AC')
  race_colors <- c(
    'White' = '#4E79A7',
    'Black' = '#F25E2B',
    'Hispanic' = '#E14759',
    'Asian' = '#76B7B2',
    'Native American' = '#59A14F',
    'Other' = '#9C755F',
    'Unknown' = '#BAB0AC'
  )

  # Gender data - ensure no duplicates by summing values
  gender_data <- state_data %>%
    filter(variable_category == 'gender',
           value_type == 'count') %>%
    group_by(offender_type, variable_detailed) %>%
    summarize(value = sum(value, na.rm = TRUE), .groups = 'drop') %>%
    complete(offender_type, variable_detailed = c('Male', 'Female', 'Unknown'),
             fill = list(value = 0))

  # Race data - ensure no duplicates by summing values
  race_data <- state_data %>%
    filter(variable_category == 'race',
           value_type == 'count') %>%
    group_by(offender_type, variable_detailed) %>%
    summarize(value = sum(value, na.rm = TRUE), .groups = 'drop') %>%
    complete(offender_type,
             variable_detailed = c('White', 'Black', 'Hispanic',
                                  'Asian', 'Native American', 'Other', 'Unknown'),
             fill = list(value = 0))

  # Create separate plots - one per row
  par(mfrow = c(2, 1), mar = c(5, 9, 4, 9), oma = c(0, 0, 2, 0)) # Increased right margin for l

  # Gender plot - ordered by total volume
  gender_plot_data <- gender_data %>%
    filter(variable_detailed %in% c('Male', 'Female', 'Unknown')) %>%
    mutate(offender_type = factor(offender_type, levels = rev(offender_types)))

  # Order gender categories by total volume (largest at bottom)
```

```r
gender_order <- gender_plot_data %>%
  group_by(variable_detailed) %>%
  summarize(total = sum(value)) %>%
  arrange(total) %>%
  pull(variable_detailed)

gender_plot_data <- gender_plot_data %>%
  mutate(variable_detailed = factor(variable_detailed, levels = gender_order))

# Reshape for barplot
gender_matrix <- gender_plot_data %>%
  pivot_wider(names_from = variable_detailed, values_from = value) %>%
  as.data.frame() %>%
  column_to_rownames("offender_type") %>%
  as.matrix()

# Ensure all columns exist
for (gender in gender_order) {
  if (!gender %in% colnames(gender_matrix)) {
    gender_matrix <- cbind(gender_matrix, temp = 0)
    colnames(gender_matrix)[ncol(gender_matrix)] <- gender
  }
}

# Reorder columns by volume
gender_matrix <- gender_matrix[, as.character(gender_order), drop = FALSE]

# Format x-axis labels with "k" for thousands
max_x <- max(rowSums(gender_matrix))
x_breaks <- pretty(c(0, max_x))
x_labels <- ifelse(x_breaks >= 1000,
                  paste0(x_breaks/1000, "k"),
                  as.character(x_breaks))

barplot(t(gender_matrix),
        horiz = TRUE,
        las = 1,
        col = gender_colors[colnames(gender_matrix)],
        main = 'Gender Distribution',
        xlab = 'Number of Profiles',
        xaxt = 'n',  # Remove default x-axis
        legend.text = FALSE,  # Don't show legend in plot area
        args.legend = list(x = "right", bty = "n", inset = c(-0.2, 0)))

# Add custom x-axis with formatted labels
axis(1, at = x_breaks, labels = x_labels)
```

```r
# Add legend outside the plot area
legend("topright",
       legend = colnames(gender_matrix),
       fill = gender_colors[colnames(gender_matrix)],
       bty = "n",
       xpd = TRUE,  # Allow plotting outside main area
       inset = c(-0.25, 0),  # Move legend to the right
       cex = 0.8)

# Race plot - ordered by total volume
race_plot_data <- race_data %>%
  mutate(offender_type = factor(offender_type, levels = rev(offender_types)))

# Order race categories by total volume (largest at bottom)
race_order <- race_plot_data %>%
  group_by(variable_detailed) %>%
  summarize(total = sum(value)) %>%
  arrange(total) %>%
  pull(variable_detailed)

race_plot_data <- race_plot_data %>%
  mutate(variable_detailed = factor(variable_detailed, levels = race_order))

# Reshape for barplot
race_matrix <- race_plot_data %>%
  pivot_wider(names_from = variable_detailed, values_from = value) %>%
  as.data.frame() %>%
  column_to_rownames("offender_type") %>%
  as.matrix()

# Ensure all columns exist
for (race in race_order) {
  if (!race %in% colnames(race_matrix)) {
    race_matrix <- cbind(race_matrix, temp = 0)
    colnames(race_matrix)[ncol(race_matrix)] <- race
  }
}

# Reorder columns by volume
race_matrix <- race_matrix[, as.character(race_order), drop = FALSE]

# Format x-axis labels with "k" for thousands
max_x_race <- max(rowSums(race_matrix))
x_breaks_race <- pretty(c(0, max_x_race))
x_labels_race <- ifelse(x_breaks_race >= 1000,
                       paste0(x_breaks_race/1000, "k"),
                       as.character(x_breaks_race))
  barplot(t(race_matrix),
          horiz = TRUE,
          las = 1,
          col = race_colors[colnames(race_matrix)],
          main = 'Race Distribution',
          xlab = 'Number of Profiles',
          xaxt = 'n',  # Remove default x-axis
          legend.text = FALSE)  # Don't show legend in plot area

  # Add custom x-axis with formatted labels
  axis(1, at = x_breaks_race, labels = x_labels_race)

  # Add legend outside the plot area
  legend("topright",
         legend = colnames(race_matrix),
         fill = race_colors[colnames(race_matrix)],
         bty = "n",
         xpd = TRUE,  # Allow plotting outside main area
         inset = c(-0.25, 0),  # Move legend to the right
         cex = 0.8)

  title(paste(state_name, "Demographic Distribution"), outer = TRUE, cex.main = 1.5)
}

# ------------------------------------------------------------------
# 20. Add state's metadata
# ------------------------------------------------------------------

add_state_metadata <- function(state_name, state_df) {

  raw_data <- state_df %>% filter(value_source == "reported")
  offender_types_reported <- unique(raw_data$offender_type)

  has_unknown <- any(raw_data$variable_detailed == "Unknown", na.rm = TRUE)
  has_other <- any(raw_data$variable_detailed == "Other", na.rm = TRUE)
  has_crosstab <- any(raw_data$variable_category == "gender_race", na.rm = TRUE)

  nonstandard_terms <- any(
    grepl("All|Offenders", raw_data$offender_type, ignore.case = TRUE),
    grepl("Caucasian|African American| American Indian", raw_data$variable_detailed, ignore.cas
    grepl("flag", raw_data$variable_detailed, ignore.case = TRUE))

  new_row <- tibble(
    state = state_name,
    race_data_provided = report_status(raw_data, "race"),
    gender_data_provided = report_status(raw_data, "gender"),
    total_profiles_provided = report_status(
      raw_data %>% filter(variable_category == "total"), "total"
    ),
    convicted_offender_reported = "Convicted Offender" %in% offender_types_reported,
    arrestee_reported = "Arrestee" %in% offender_types_reported,
    combined_reported = "Combined" %in% offender_types_reported,
    has_unknown_category = has_unknown,
    has_other_category = has_other,
    uses_nonstandard_terminology = nonstandard_terms,
    provides_crosstabulation = has_crosstab,
    counts_sum_to_total = NA,
    percentages_sum_to_100 = NA,
    total_calculated_combined = !("Combined" %in% offender_types_reported),
    notes = ""
  )

  foia_state_metadata <<- bind_rows(foia_state_metadata, new_row)

  cat("  Metadata added for:", state_name, "\n")
  return(invisible(TRUE))
}

# -------------------------------------------------------------------
# 21. Function to update a state's metadata after QC checks
# -------------------------------------------------------------------
update_state_metadata <- function(state_name,
                                  counts_ok = NA,
                                  percentages_ok = NA,
                                  notes_text = NULL) {

  row_index <- which(foia_state_metadata$state == state_name)

  if (length(row_index) == 0) {
    warning("State not found in metadata: ", state_name)
    return(FALSE)
  }

  if (!is.na(counts_ok)) {
    foia_state_metadata$counts_sum_to_total[row_index] <<- counts_ok
  }
  if (!is.na(percentages_ok)) {
    foia_state_metadata$percentages_sum_to_100[row_index] <<- percentages_ok
  }
  if (!is.null(notes_text)) {
    current_notes <- foia_state_metadata$notes[row_index]
    if (current_notes == "") {
      foia_state_metadata$notes[row_index] <<- notes_text
    } else {
      foia_state_metadata$notes[row_index] <<- paste(current_notes, notes_text, sep = "; ")
    }
  }

  cat("  Metadata updated for:", state_name, "\n")
}
```

2.4 File Structure and Contents

2.4.1 State-Specific Files: `data/foia/intermediate/[state]_foia_data.csv`

Purpose: Individual files for each state containing only their reported data.

Structure: Long format with columns:

- `state`: State name
- `offender_type`: Category of individuals (Convicted Offender, Arrestee, Combined, etc.)
- `variable_category`: Type of data (total, gender, race, gender_race)
- `variable_detailed`: Specific value (e.g., Male, Female, African American)
- `value`: The reported number or percentage
- `value_type`: Whether value is a "count" or "percentage"
- `date`: Date of data snapshot, if reported

```
## Per-state files loading code --

ca_raw <- load_state(here(per_state, "california_foia_data.csv"))
fl_raw <- load_state(here(per_state, "florida_foia_data.csv"))
in_raw <- load_state(here(per_state, "indiana_foia_data.csv"))
me_raw <- load_state(here(per_state, "maine_foia_data.csv"))
nv_raw <- load_state(here(per_state, "nevada_foia_data.csv"))
sd_raw <- load_state(here(per_state, "south_dakota_foia_data.csv"))
tx_raw <- load_state(here(per_state, "texas_foia_data.csv"))
```

2.4.2 Raw Data Characteristics

The following table summarizes the structure and content of the data as originally received from each state prior to any standardization, calculation, or processing.

State	Offender Types	Value Types	Total Profiles	Action Needed	Key Reporting Notes
California	CO, A	Counts only	Reported per offender type	Add Unknown Race, Calculate % & Combined, Standardize Terminology	Discrepancy in Race: counts < total profiles; Non-standard terminology (Caucasian and African American)
Florida	COMB	Counts + %	Reported	Standardize Terminology	Non-standard terminology (Caucasian and African American)
Indiana	CO, A, COMB	Percentage (Counts for totals only)	Reported per offender type	Calculate Counts & Total Profiles Combined, Fix % inconsistency, Standardize Terminology	Demographics only for Combined; `Other` race category as "<1"; Non-standard terminology (Caucasian)
Maine	COMB	Counts + %	Reported	Solve counts and Percentage inconsistency	
Nevada	CO, A, COMB	Counts + %	Reported for all types	Standardize Terminology	Non-standard terminology (All, total_flags and American Indian)
South Dakota	COMB	Counts + %	Reported	Standardize Terminology, Solve counts and % inconsistency	Includes gender×race cross-tabulation; Non-standard terminology

State	Offender Types	Value Types	Total Profiles	Action Needed	Key Reporting Notes
Texas	CO, A	Counts only	Reported per offender type	Calculate Male counts, Solve counts inconsistency, Calculate % & Combined, Standardize Terminology	Only female gender was reported; Non-standard term (Offenders, Caucasian, and African American)

Legend:

- **CO:** Convicted Offender
- **AR:** Arrestee
- **COMB:** Combined Total (all profiles)
- **Counts + %:** Both raw numbers and percentages were provided

2.5 Prepare Combined Dataset

The goal of this step is to transform each state's raw data into a standardized format before appending it to the master `foia_combined` DataFrame. This ensures consistency and enables seamless analysis across all seven states.

The ideal, standardized state dataset ready for combination must have the following columns:

Column Name	Description	Example Values
`state`	The name of the state.	`"California"`, `"Florida"`
`offender_type`	The category of offender profile.	`"Convicted Offender"`, `"Arrestee"`, `"Combined"`
`variable_category`	The broad demographic category.	`"race"`, `"gender"`, `"total"`, `"gender_race"`
`variable_detailed`	The specific value within the category.	`"White"`, `"Male"`, `"total_profiles"`, `"Male_White"`
`value`	The numerical value for the metric.	`150000`, `25.8`
`value_type`	The type of metric the value represents.	`"count"`, `"percentage"`
`value_source`	Whether the data was provided or derived.	`"reported"`, `"calculated"`

```r
## Master foia_combined dataframe--

foia_combined <- tibble( state = character(),
offender_type = character(),
variable_category = character(),
variable_detailed = character(),
value = numeric(),
value_type = character(),
value_source = character()
)

# Create a data dictionary for foia_combined
schema_dict <- tribble(
  ~Column,              ~Type,        ~Description,
  "state",              "character",  "'California', 'Florida'",
  "offender_type",      "character",  "'Convicted Offender', 'Arrestee', 'Combined'",
  "variable_category",  "character",  "'race', 'gender', 'total', 'gender_race'",
  "variable_detailed",  "character",  "'White', 'Male', 'total_profiles', 'Male_White'",
  "value",              "numeric",    "150000, 25.8",
  "value_type",         "character",  "'count', 'percentage'",
  "value_source",       "character",  "'reported', 'calculated'"
)

# Turn into a nice flextable
flextable(schema_dict) %>%
  autofit() %>%
  theme_booktabs() %>%
  set_header_labels(
    Column = "Column Name",
    Type = "Data Type",
    Description = "Example Values to be added"
  )
```

Column Name	Data Type	Example Values to be added
state	character	’California’, ’Florida’
offender_type	character	’Convicted Offender’, ’Arrestee’, ’Combined’
variable_category	character	’race’, ’gender’, ’total’, ’gender_race’
variable_detailed	character	’White’, ’Male’, ’total_profiles’, ’Male_White’
value	numeric	150000, 25.8
value_type	character	’count’, ’percentage’
value_source	character	’reported’, ’calculated’

2.6 Prepare Metadata Documentation Table

This section creates a comprehensive metadata table (**`foia_state_metadata`**) to document the original content and structure of each state's FOIA response *before* any processing or cleaning was applied.

This serves as a permanent record of data provenance, ensuring transparency and reproducibility by clearly distinguishing between what was *provided* by the states and what was *calculated* during analysis.

Key Documentation Captured:

- **Data Types Provided:** Whether each state reported counts, percentages, or both for race, gender, and total profiles.
- **Offender Categories Reported:** Which offender types (Convicted Offender, Arrestee, Combined) were originally included.
- **Demographic Granularity:** Presence of 'Unknown' or 'Other' categories and gender-race cross-tabulations.
- **Terminology & Anomalies:** Use of non-standard terms (e.g., "flags," "offenders") and other state-specific reporting notes.
- **QC Results:** Flags for whether cleaned data passes consistency checks (counts sum to totals, percentages sum to ~100%).

```r
## foia_state_metadata table elaboration code --

# Define the full schema for our metadata table
foia_state_metadata <- tibble(
  state = character(),
  race_data_provided = character(),
  gender_data_provided = character(),
  total_profiles_provided = character(),
  convicted_offender_reported = logical(),
  arrestee_reported = logical(),
  combined_reported = logical(),
  has_unknown_category = logical(),
  has_other_category = logical(),
  uses_nonstandard_terminology = logical(),
  provides_crosstabulation = logical(),
  counts_sum_to_total = logical(),
  percentages_sum_to_100 = logical(),
  total_calculated_combined = logical(),
  notes = character()
)

# Build data dictionary for foia_state_metadata
schema_dict_meta <- tribble(
  ~Column,                          ~Type,       ~Description,
  "state",                          "character", "State name (e.g., 'California', 'Florida')",
  "race_data_provided",             "character", "Race data availability: 'counts', 'percentages
  "gender_data_provided",           "character", "Gender data availability: 'counts', 'percentage
  "total_profiles_provided",        "character", "Total profiles availability: 'counts', 'percen
  "convicted_offender_reported",    "logical",   "Was convicted offender data reported?",
  "arrestee_reported",              "logical",   "Was arrestee data reported?",
  "combined_reported",              "logical",   "Was combined category reported?",
  "has_unknown_category",           "logical",   "Does the state include 'Unknown' category?",
  "has_other_category",             "logical",   "Does the state include 'Other' category?",
  "uses_nonstandard_terminology",   "logical",   "Does the state use non-standard terms?",
  "provides_crosstabulation",       "logical",   "Does the state provide crosstabs (e.g., gender
  "counts_sum_to_total",            "logical",   "Do reported counts sum to the total?",
  "percentages_sum_to_100",         "logical",   "Do reported percentages sum to ~100%?",
  "total_calculated_combined",      "logical",   "Did we calculate combined total manually?",
  "notes",                          "character", "Free-text notes for state-specific caveats"
)

# Render with flextable
flextable(schema_dict_meta) %>%
  autofit() %>%
  theme_booktabs() %>%
  set_header_labels(
    Column = "Column Name",
    Type = "Data Type",
    Description = "Meaning"
  )
```

Column Name	Data Type	Meaning
state	character	State name (e.g., ’California’, ’Florida’)
race_data_provided	character	Race data availability: ’counts’, ’percentages’, ’both’, ’none’
gender_data_provided	character	Gender data availability: ’counts’, ’percentages’, ’both’, ’non
total_profiles_provided	character	Total profiles availability: ’counts’, ’percentages’, ’both’, ’non
convicted_offender_reported	logical	Was convicted offender data reported?
arrestee_reported	logical	Was arrestee data reported?
combined_reported	logical	Was combined category reported?
has_unknown_category	logical	Does the state include ’Unknown’ category?
has_other_category	logical	Does the state include ’Other’ category?
uses_nonstandard_terminology	logical	Does the state use non-standard terms?
provides_crosstabulation	logical	Does the state provide crosstabs (e.g., gender x race)?

Column Name	Data Type	Meaning
counts_sum_to_total	logical	Do reported counts sum to the total?
percentages_sum_to_100	logical	Do reported percentages sum to ~100%?
total_calculated_combined	logical	Did we calculate combined total manually?
notes	character	Free-text notes for state-specific caveats

3 State-by-state Standardization

Each state is processed individually to standardize terminology, fill gaps, and calculate Combined totals where necessary.

3.1 California (CA)

Overview: California supplies **counts only** for gender and race plus a separate total for each offender type; no percentages are reported.

3.1.1 Examine Raw Data

Establish a baseline understanding of the data exactly as it was received.

Column	Type	Rows	Missing	Unique	Unique_Values
state	character	16	0	1	California
offender_type	character	16	0	2	Convicted Offender, Arrestee
variable_category	character	16	0	3	total, gender, race
variable_detailed	character	16	0	8	total_profiles, Female, Male, Unknown, African American, Caucasian
value	numeric	16	0	16	2019899 ..., 751822 ..., 309827 ..., 1603222 ..., 106850 ..., 208225 ..
value_type	character	16	0	1	count
value_source	character	16	0	1	reported

Data frame dimensions: 16 rows × 7 columns

3.1.2 Verify Data Consistency

Runs the first quality check using the **`verify_category_totals()`** and **`counts_consistent()`** functions.

This identifies any immediate discrepancies, such as the sum of demographic counts not matching the reported total profiles, which flags data issues that need to be resolved.

```
Verifying that demographic counts match reported totals:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Arrestee	gender	751822	751822	0
Arrestee	race	751822	655695	96127
Convicted Offender	gender	2019899	2019899	0
Convicted Offender	race	2019899	1626012	393887

```
Counts consistency check on raw data:

All counts consistent: FALSE
```

3.1.3 Address Data Gaps

3.1.3.1 Create Unknown Category

> *"Racial classification is not considered a required field on the collection card; thus, an unknown number of offenders may have **no racial classification listed**."* — California DOJ FOIA letter, July 10 2018 (**`raw/foia_pdfs/FOIA_RacialComp_California.pdf`**)

The 393,887 Convicted Offender profiles and 96,127 Arrestee profiles that do **not** appear in any of the four reported race categories must belong to an unreported "Unknown" category.

The calculated values are added with a **`value_source = "calculated"`** tag to maintain transparency about what was provided versus what was derived.

```r
# Start with the raw data
ca_clean <- ca_raw

# Add Unknown race category to reconcile totals
ca_clean <- fill_demographic_gaps(ca_clean)

# Verify the fix
cat("Category totals after adding Unknown race category:\n")
verify_category_totals(ca_clean) %>% kable() %>% kable_styling()

cat("\nCounts consistency after adding Unknown:\n")
cat(paste("All counts consistent:", counts_consistent(ca_clean), "\n"))
```

```
Category totals after adding Unknown race category:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Arrestee	gender	751822	751822	0
Arrestee	race	751822	751822	0

Convicted Offender	gender	2019899	2019899	0
Convicted Offender	race	2019899	2019899	0

```
Counts consistency after adding Unknown:
All counts consistent: TRUE
```

3.1.3.2 Create Combined Totals

Since California only reported data for “Convicted Offender” and “Arrestee” separately.

This step uses the **add_combined()** helper function to calculate a new “Combined” offender type by summing the counts from the other two categories.

```
# Calculate Combined totals using helper function
ca_clean <- add_combined(ca_clean)

cat("  Created Combined totals for California\n")

# Show the Combined total
combined_total <- ca_clean %>%
  filter(offender_type == "Combined",
         variable_category == "total",
         variable_detailed == "total_profiles") %>%
  pull(value)

cat(paste("Combined total profiles:", format(combined_total, big.mark = ","), "\n"))
```

```
  Created Combined totals for California
Combined total profiles: 2,771,721
```

3.1.3.3 Calculate Percentages

Transforms the data from counts into percentages for comparative analysis.

The **add_percentages()** helper function calculates each demographic group’s proportion relative to its offender type’s total.

A final consistency check ensures all percentages logically sum to approximately 100%.

```
# Derive percentages from counts
ca_clean <- add_percentages(ca_clean)

cat("  Added percentages for all demographic categories\n")

# Check percentage consistency
cat("Percentage consistency check:\n")
cat(paste("All percentages sum to ~100%:", percentages_consistent(ca_clean), "\n\n"))
```

```
# Show current data availability
cat("Final data availability:\n")
cat(paste("Race data:", report_status(ca_clean, "race"), "\n"))
cat(paste("Gender data:", report_status(ca_clean, "gender"), "\n"))
```

```
  Added percentages for all demographic categories
Percentage consistency check:
All percentages sum to ~100%: TRUE

Final data availability:
Race data: both
Gender data: both
```

3.1.3.4 Standardize Terminology

California uses “African American” instead of “Black” and “Caucasian” instead of “White”.

```
# Standardize racial terminology
ca_clean <- ca_clean %>%
  mutate(variable_detailed = case_when(
    variable_detailed == "African American" ~ "Black",
    TRUE ~ variable_detailed
  ))

cat("  Standardized terminology: 'African American' → 'Black'\n")

ca_clean <- ca_clean %>%
  mutate(variable_detailed = case_when(
    variable_detailed == "Caucasian" ~ "White",
    TRUE ~ variable_detailed
  ))

cat("  Standardized terminology: 'Caucasian' → 'White'\n")
```

```
  Standardized terminology: 'African American' → 'Black'
  Standardized terminology: 'Caucasian' → 'White'
```

3.1.4 Prepare for Combined Dataset

The cleaned data is formatted to match the master schema and appended to the `foia_combined` dataframe.

```
# Prepare the cleaned data for the combined dataset
ca_prepared <- prepare_state_for_combined(ca_clean, "California")
```

```r
# Append to the master combined dataframe
foia_combined <- bind_rows(foia_combined, ca_prepared)

cat(paste0("  Appended ", nrow(ca_prepared), " California rows to foia_combined\n"))
cat(paste0("  Total rows in foia_combined: ", nrow(foia_combined), "\n"))
```

```
Appended 51 California rows to foia_combined
Total rows in foia_combined: 51
```

3.1.5 Document Metadata

The metadata is added with the raw information and updated with the results of the quality checks and a note on the processing steps taken.

```r
# Add California to the metadata table using the helper function
add_state_metadata("California", ca_raw)

# Update metadata with QC results
update_state_metadata("California",
                      counts_ok = counts_consistent(ca_clean),
                      percentages_ok = percentages_consistent(ca_clean),
                      notes_text = "Added Unknown race category to reconcile totals; calculated
```

```
Metadata added for: California
Metadata updated for: California
```

3.1.6 Visualizations

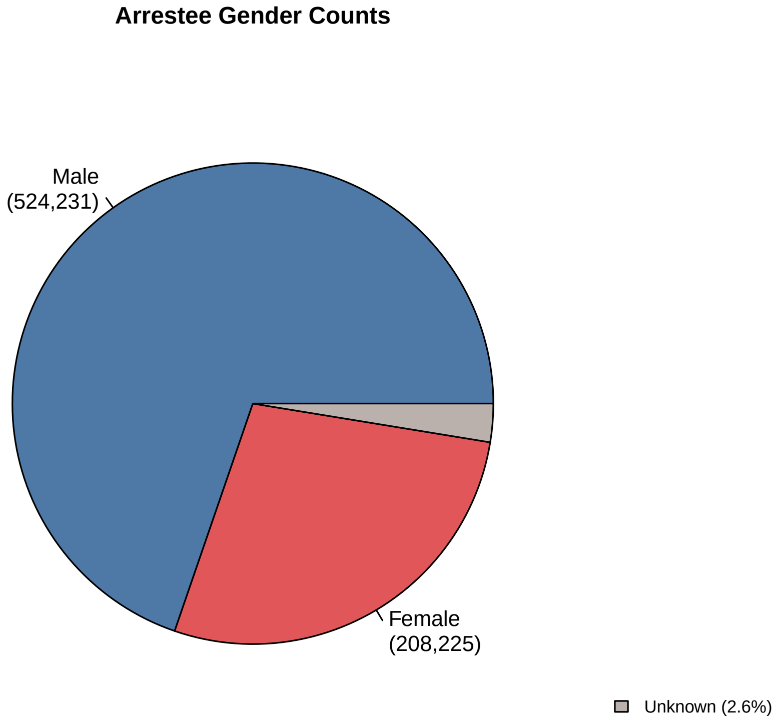


Figure 1: California DNA Database Demographic Distributions

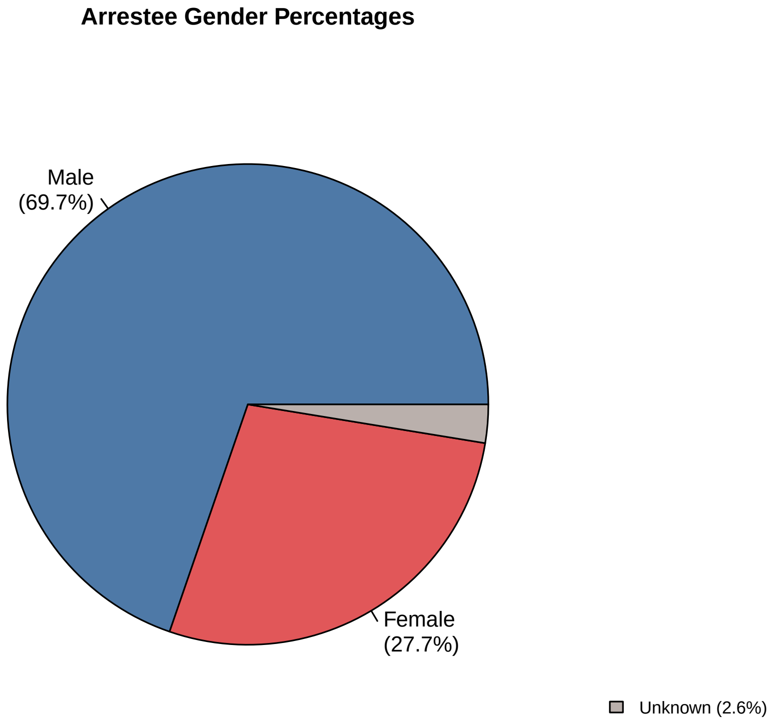


Figure 2: California DNA Database Demographic Distributions

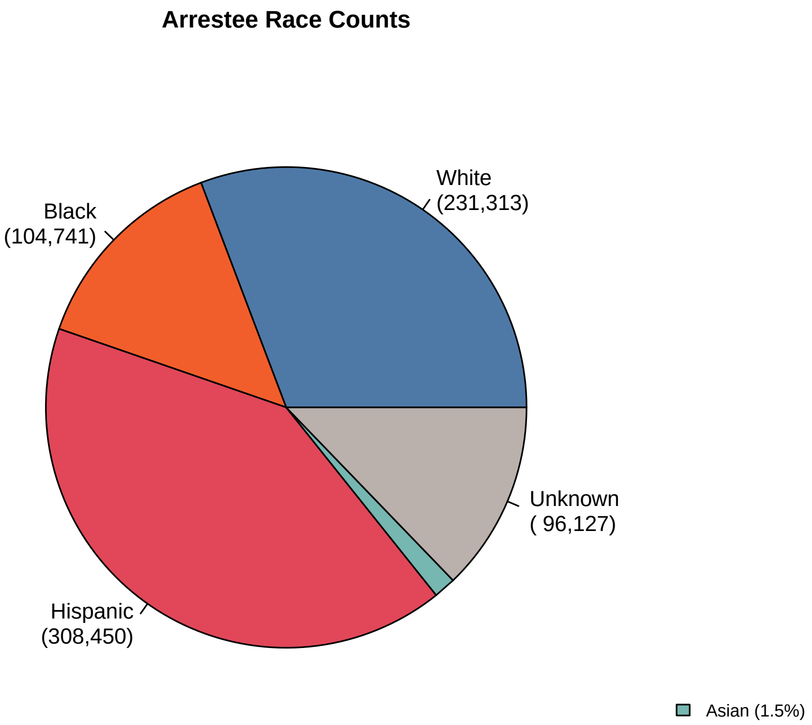


Figure 3: California DNA Database Demographic Distributions

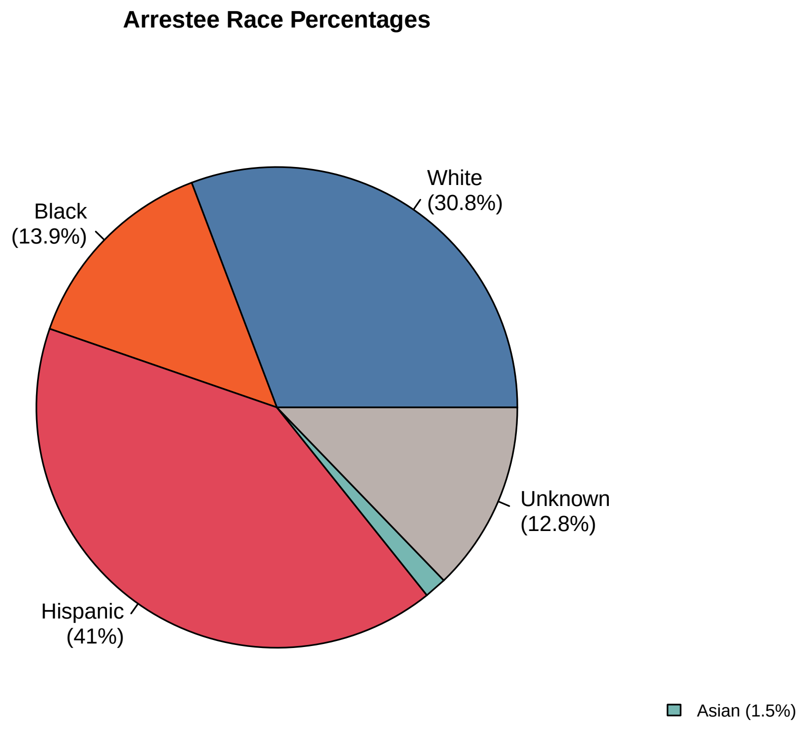


Figure 4: California DNA Database Demographic Distributions

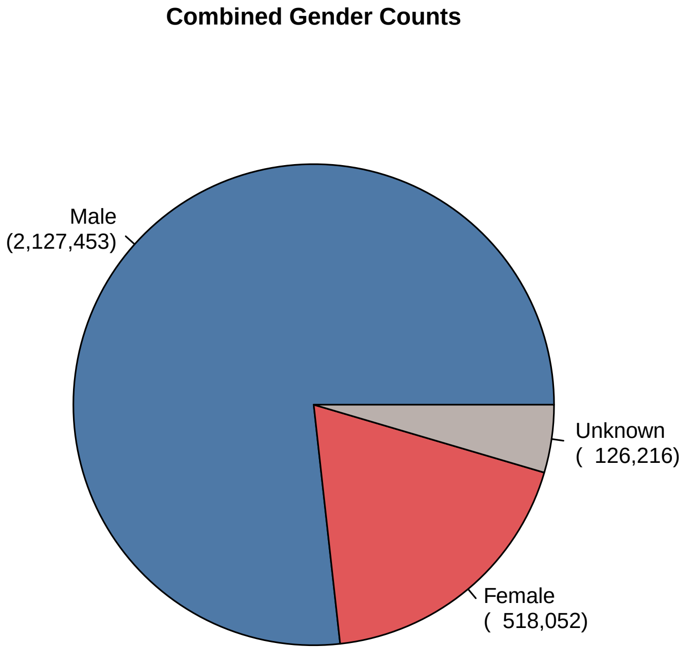


Figure 5: California DNA Database Demographic Distributions

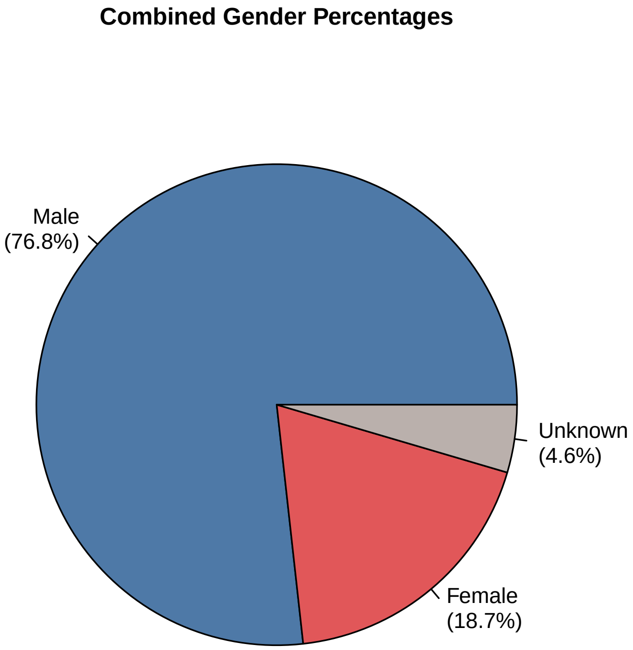


Figure 6: California DNA Database Demographic Distributions

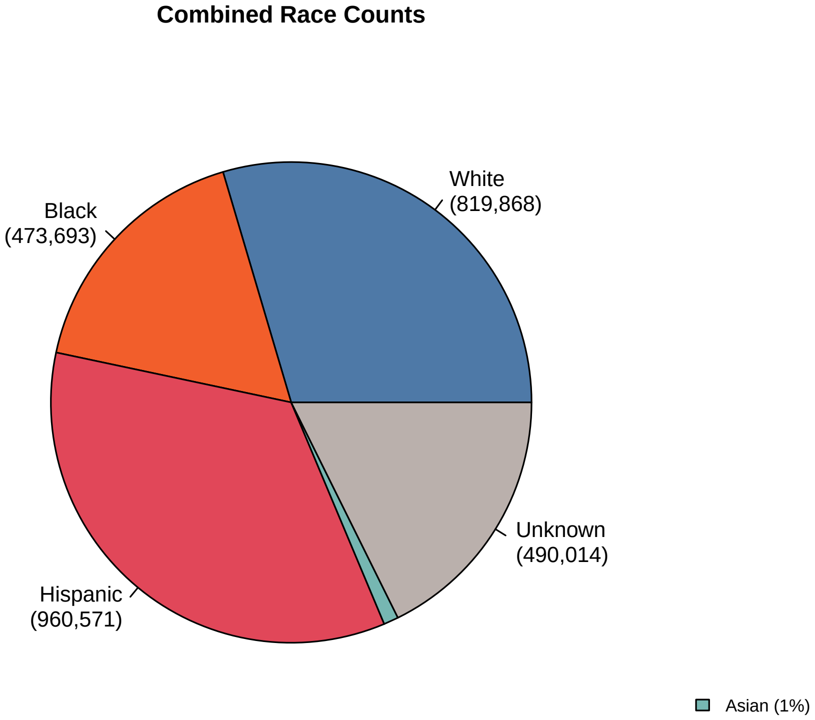


Figure 7: California DNA Database Demographic Distributions

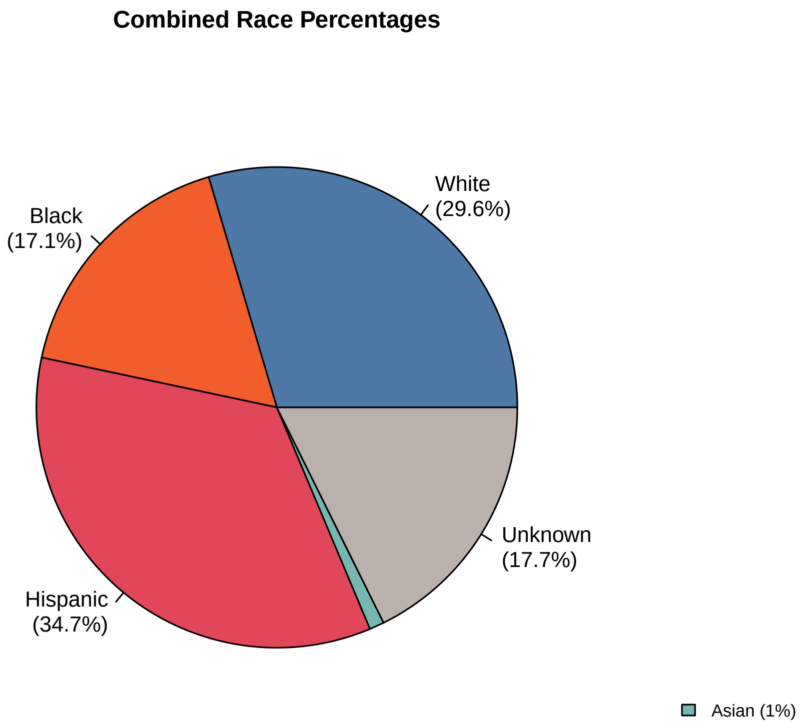


Figure 8: California DNA Database Demographic Distributions

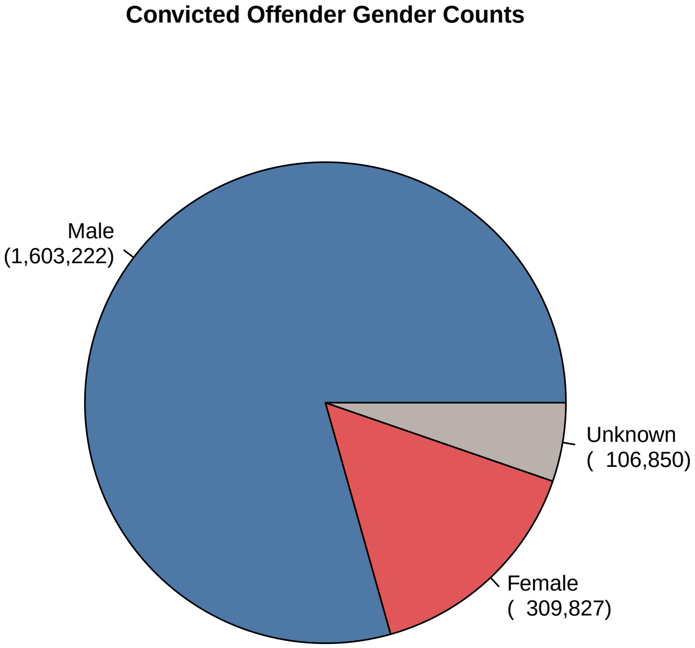


Figure 9: California DNA Database Demographic Distributions

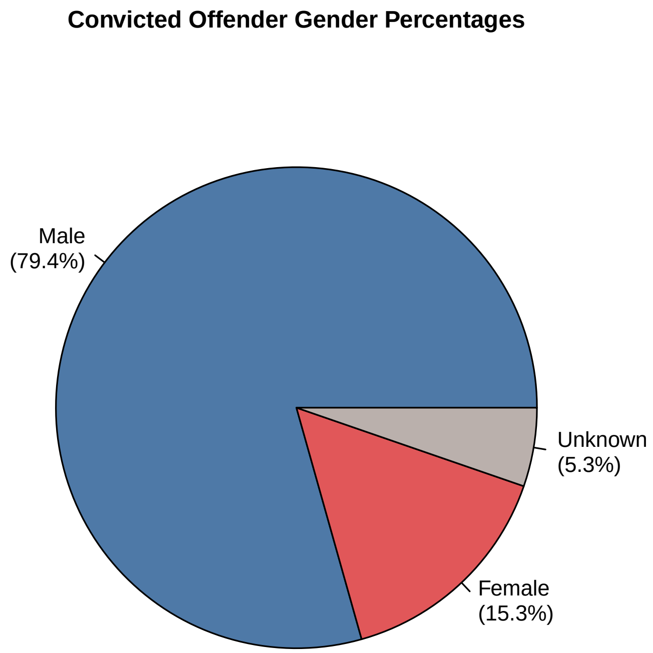


Figure 10: California DNA Database Demographic Distributions

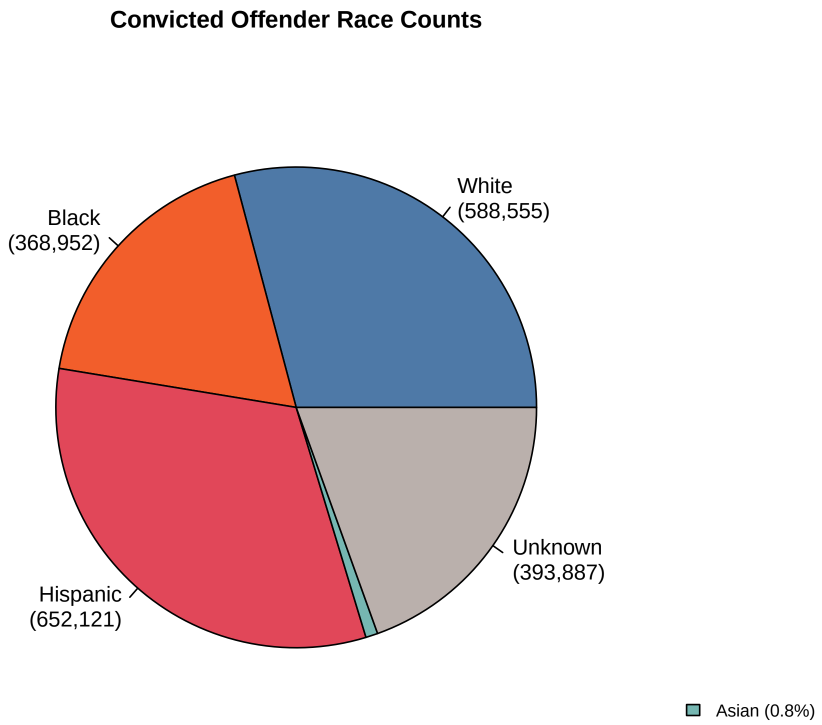


Figure 11: California DNA Database Demographic Distributions

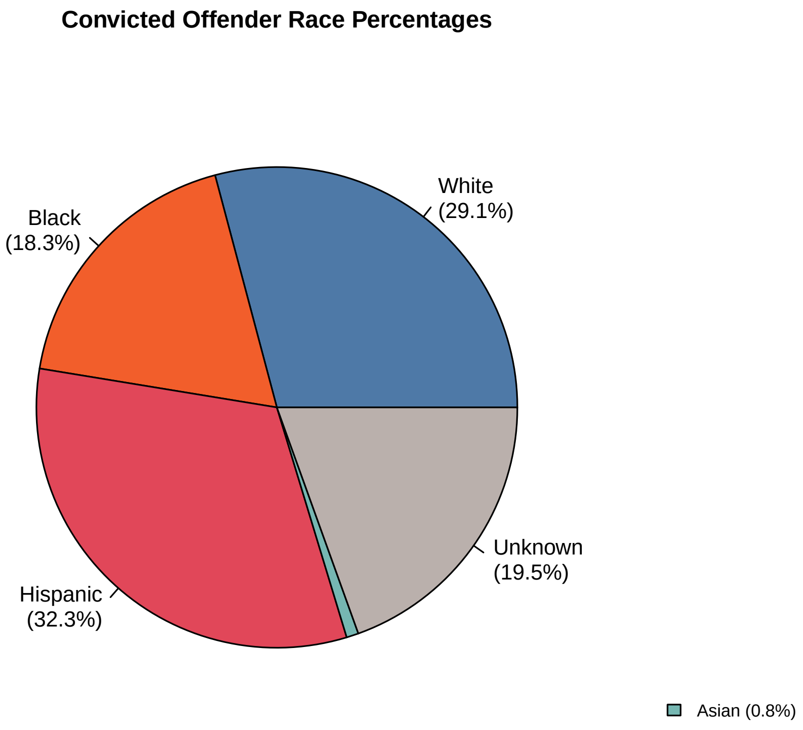


Figure 12: California DNA Database Demographic Distributions

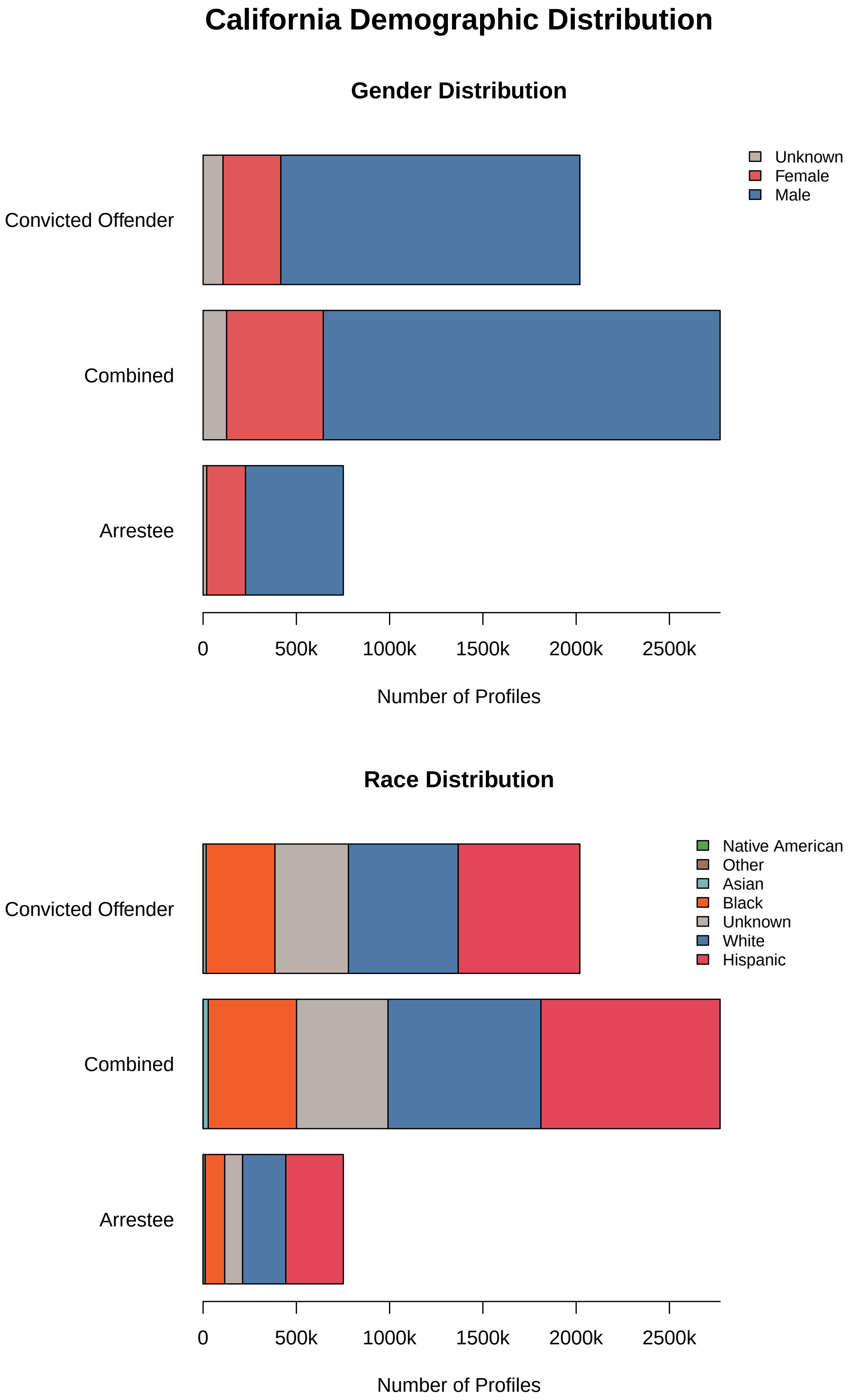


Figure 13: California Demographic Distributions by Offender Type

3.1.7 Summary Statistics

```r
cat("California DNA Database Summary:\n")
cat("=", strrep("=", 40), "\n")

# Total profiles by offender type
totals <- foia_combined %>%
  filter(state == "California",
         variable_category == "total",
         variable_detailed == "total_profiles",
         value_type == "count") %>%
  select(offender_type, value) %>%
  mutate(value_formatted = format(value, big.mark = ","))

print(totals)

# Data completeness
cat("\nData completeness:\n")
completeness <- foia_combined %>%
  filter(state == "California") %>%
  group_by(offender_type, value_source) %>%
  summarise(n_values = n(), .groups = "drop")

print(completeness)

# Final verification
cat("\nFinal verification:\n")
cat(paste("Counts consistent:", counts_consistent(foia_combined %>% filter(state == "Californi
cat(paste("Percentages consistent:", percentages_consistent(foia_combined %>% filter(state == "
```

```
California DNA Database Summary:
= ========================================
# A tibble: 3 x 3
  offender_type        value value_formatted
  <chr>                <dbl> <chr>
1 Convicted Offender 2019899 "2,019,899"
2 Arrestee            751822 "  751,822"
3 Combined           2771721 "2,771,721"

Data completeness:
# A tibble: 5 x 3
  offender_type      value_source n_values
  <chr>              <chr>           <int>
1 Arrestee           calculated          9
2 Arrestee           reported            8
3 Combined           calculated         17
```

```
4 Convicted Offender calculated          9
5 Convicted Offender reported            8

Final verification:
Counts consistent: TRUE
Percentages consistent: TRUE
```

3.1.8 Summary of California Processing

California data processing complete. The dataset now includes:

- **Reported data**: Counts for Convicted Offender and Arrestee
- **Calculated additions**:
  - Unknown race category to reconcile reported totals
  - Combined totals across all offender types
  - Percentage values for all demographic categories
  - "Caucasian" and "African American" converted to "White" and "Black".
- **Quality checks**: All counts and percentages pass consistency validation
- **Provenance tracking**: All values include appropriate `value_source` indicators

The California data is now standardized and ready for cross-state analysis.

3.2 Florida (FL)

Overview: Florida provides **both counts and percentages** for gender and race categories and already includes a "Combined" total for all offender types, making it one of the most complete and straightforward datasets.

Only requires to standardize terminology for gender and race categories to match the common data model.

3.2.1 Examine Raw Data

Establish a baseline understanding of the data exactly as it was received.

Column	Type	Rows	Missing	Unique	Unique_Values
state	character	22	0	1	Florida
offender_type	character	22	0	1	Combined
variable_category	character	22	0	3	total, gender, race
variable_detailed	character	22	0	10	total_profiles, Female, Male, Unknown, African American, Asian, Cau

Column	Type	Rows	Missing	Unique	Unique_Values
value	numeric	22	0	22	1175391 ..., 100 ..., 260885 ..., 22.2 ..., 901126 ..., 76.67 ..., 13380 ..
value_type	character	22	0	2	count, percentage
value_source	character	22	0	1	reported

Data frame dimensions: 22 rows × 7 columns

3.2.2 Verify Data Consistency

Runs the first quality check using the **`Verify_category_totals()`** and **`counts_consistent()`** functions.

```
Verifying that demographic counts match reported totals:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	1175391	1175391	0
Combined	race	1175391	1175391	0

```
Counts consistency check on raw data:

All counts consistent: TRUE


Percentage consistency check on raw data:

All percentages sum to ~100%: TRUE
```

3.2.3 Address Data Gaps

3.2.3.1 Standardize Terminology

Florida uses “African American” instead of “Black” and “Caucasian” instead of “White”.

```
fl_clean <- fl_raw

# Standardize racial terminology
fl_clean <- fl_clean %>%
  mutate(variable_detailed = case_when(
    variable_detailed == "African American" ~ "Black",
    TRUE ~ variable_detailed
  ))
```

```
cat("  Standardized terminology: 'African American' → 'Black'\n")

fl_clean <- fl_clean %>%
  mutate(variable_detailed = case_when(
    variable_detailed == "Caucasian" ~ "White",
    TRUE ~ variable_detailed
  ))

cat("  Standardized terminology: 'Caucasian' → 'White'\n")
```

```
  Standardized terminology: 'African American' → 'Black'
  Standardized terminology: 'Caucasian' → 'White'
```

3.2.4 Prepare for Combined Dataset

The Florida data is already complete and consistent. It is formatted to match the master schema and appended to the foia_combined dataframe.

```
# Prepare the data for the combined dataset
fl_prepared <- prepare_state_for_combined(fl_clean, "Florida")

# Append to the master combined dataframe
foia_combined <- bind_rows(foia_combined, fl_prepared)

cat(paste0("  Appended ", nrow(fl_prepared), " Florida rows to foia_combined\n"))
cat(paste0("  Total rows in foia_combined: ", nrow(foia_combined), "\n"))
```

```
  Appended 22 Florida rows to foia_combined
  Total rows in foia_combined: 73
```

3.2.5 Document Metadata

The metadata is added with a note that the data was complete and required no processing.

```
# Add Florida to the metadata table using the helper function
add_state_metadata("Florida", fl_raw)

# Update metadata with QC results
update_state_metadata("Florida",
                      counts_ok = counts_consistent(fl_clean),
                      percentages_ok = percentages_consistent(fl_clean),
                      notes_text = "Complete dataset provided. No processing or calculations r
```

```
  Metadata added for: Florida
  Metadata updated for: Florida
```

3.2.6 Visualizations

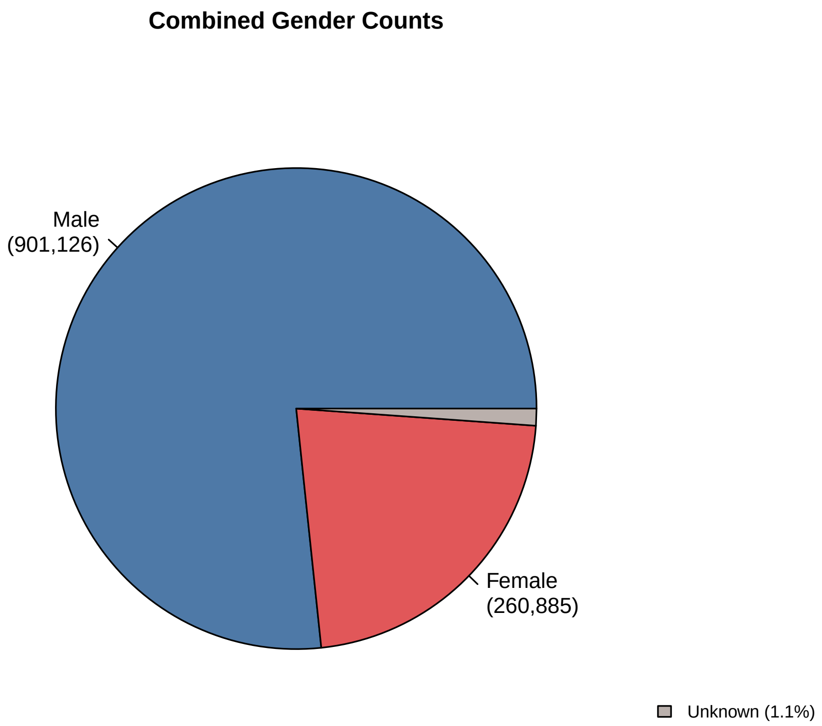


Figure 14: Florida DNA Database Demographic Distributions

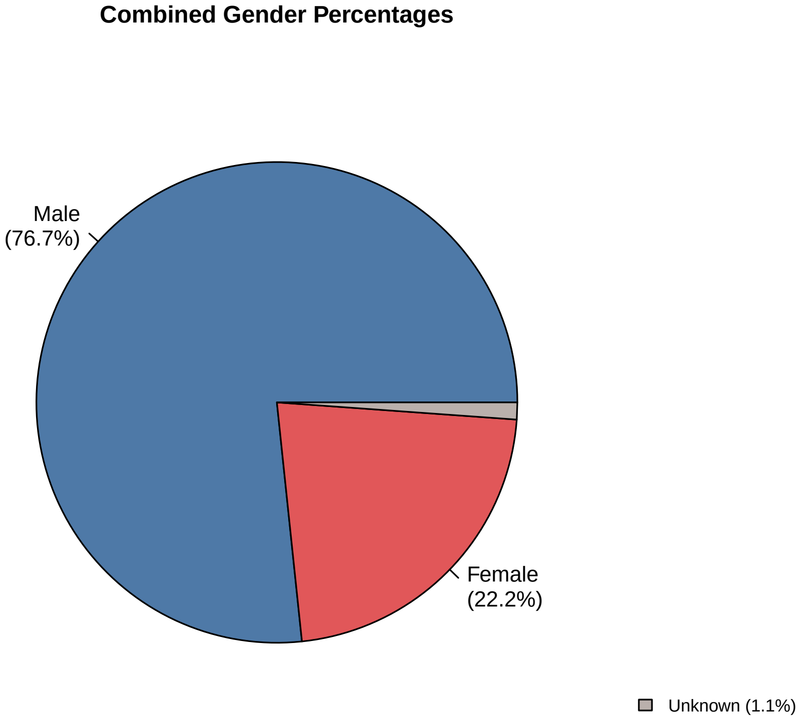


Figure 15: Florida DNA Database Demographic Distributions

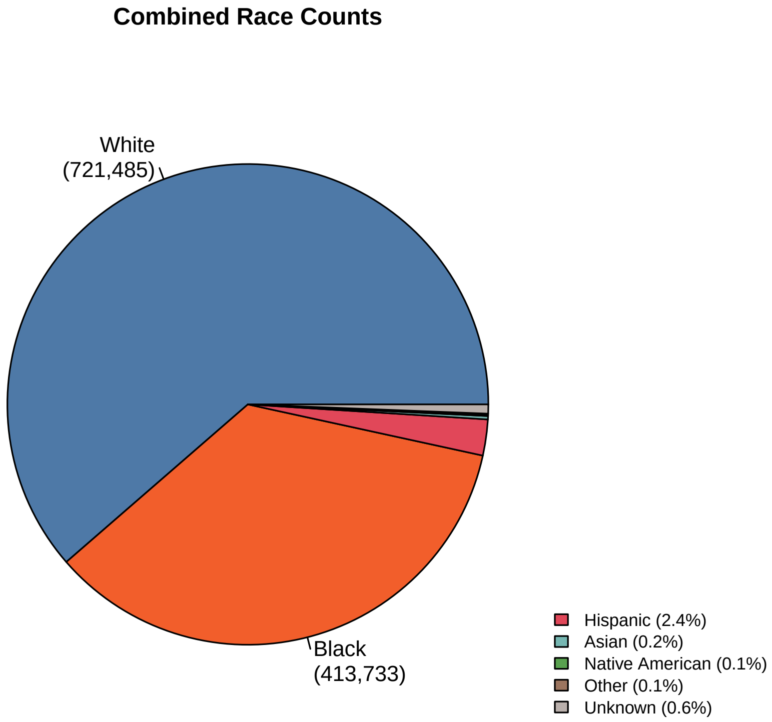


Figure 16: Florida DNA Database Demographic Distributions

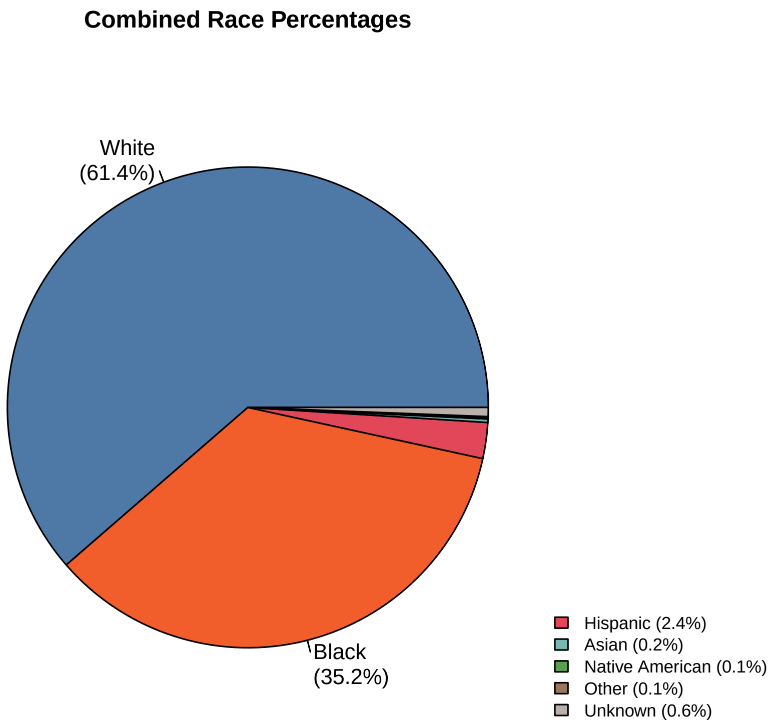


Figure 17: Florida DNA Database Demographic Distributions

3.2.7 Summary Statistics

```
cat("Florida DNA Database Summary:\n")
cat("=", strrep("=", 40), "\n")

# Total profiles by offender type
totals <- foia_combined %>%
  filter(state == "Florida",
         variable_category == "total",
         variable_detailed == "total_profiles",
         value_type == "count") %>%
  select(offender_type, value) %>%
  mutate(value_formatted = format(value, big.mark = ","))

print(totals)

# Data completeness
cat("\nData completeness:\n")
```

```
completeness <- foia_combined %>%
  filter(state == "Florida") %>%
  group_by(offender_type, value_source) %>%
  summarise(n_values = n(), .groups = "drop")

print(completeness)

# Final verification
cat("\nFinal verification:\n")
cat(paste("Counts consistent:", counts_consistent(foia_combined %>% filter(state == "Florida")
cat(paste("Percentages consistent:", percentages_consistent(foia_combined %>% filter(state == '
```

```
Florida DNA Database Summary:
= ========================================
# A tibble: 1 x 3
  offender_type   value value_formatted
  <chr>           <dbl> <chr>
1 Combined      1175391 1,175,391

Data completeness:
# A tibble: 1 x 3
  offender_type value_source n_values
  <chr>         <chr>           <int>
1 Combined      reported           22

Final verification:
Counts consistent: TRUE
Percentages consistent: TRUE
```

3.2.8 Summary of Florida Processing

Florida data processing complete. The dataset is exemplary and required no adjustments:

- **Reported data**: Both **counts and percentages** for all Convicted Offender, Arrestee, and Combined categories.
- **Terminology standardization**: "Caucasian" and "African American" converted to "White" and "Black".
- **No calculated additions needed**: All values are sourced directly from the state report (`value_source = "reported"`).
- **Quality checks**: All counts and percentages pass consistency validation.
- **Provenance tracking**: All values maintain their original `value_source` as "reported".

The Florida data is now standardized and ready for cross-state analysis.

3.3 Indiana (IN)

Overview: Indiana presents a unique reporting pattern where total counts are provided by offender type, but demographic breakdowns are given only as percentages for the Combined total.

Values were provided as strings, including a "<1" notation, requiring conversion.

3.3.1 Examine Raw Data

Establish a baseline understanding of the data exactly as it was received.

Column	Type	Rows	Missing	Unique	Unique_Values
state	character	8	0	1	Indiana
offender_type	character	8	0	3	Convicted Offender, Arrestee, Combined
variable_category	character	8	0	3	total, gender, race
variable_detailed	character	8	0	7	total_profiles, Female, Male, Caucasian, Black, Hispanic, Other
value	numeric	8	0	8	279654, 21087, 20, 80, 70, 26, 4, 0.5
value_type	character	8	0	2	count, percentage
value_source	character	8	0	1	reported

Data frame dimensions: 8 rows × 7 columns

3.3.2 Verify Data Consistency

Initial checks reveal Indiana's unique structure: counts for totals, percentages only for Combined demographics.

```
Initial data availability:

Race data: percentages

Gender data: percentages


Value types in raw data:

count, percentage
```

3.3.3 Address Data Gaps

3.3.3.1 Convert String Values to Numeric

The raw data contains string values including "<1" which we convert to 0.5.

```
# Start with raw data
in_clean <- in_raw

# Convert string values to numeric, handling "<1" as 1
in_clean$value <- sapply(in_clean$value, function(x) {
  if (x == "<1") {
    0.5
  } else {
    as.numeric(x)
  }
})

# Update value_type for converted percentages
in_clean <- in_clean %>%
  mutate(value_type = ifelse(value_type == "percentage", "percentage", value_type))

cat("  Converted Indiana values from String to numeric\n")
cat(paste("Unique values after conversion:", paste(unique(in_clean$value), collapse = ", "), "
```

```
  Converted Indiana values from String to numeric
Unique values after conversion: 279654, 21087, 20, 80, 70, 26, 4, 0.5
```

3.3.3.2 Solve Percentages Inconsistency

Racial percentages summed to 100.5% instead of 100%

Proportional scaling was applied and `value_source` was updated to “calculated” for all adjusted values.

```
# Adjust percentages to ensure they sum to 100% and mark as calculated
in_clean <- in_clean %>%
  group_by(value_type, variable_category) %>%
  mutate(
    value = ifelse(
      value_type == "percentage" & variable_category == "race",
      value * (100 / sum(value, na.rm = TRUE)),
      value
    ),
    value_source = ifelse(
      value_type == "percentage" & variable_category == "race",
      "calculated",
      value_source
    )
  ) %>%
  ungroup()

# Verify the new sum
```

```
percentage_sum <- in_clean %>%
  filter(value_type == "percentage" & variable_category == "race") %>%
  summarise(total = sum(value, na.rm = TRUE))

cat("  Recalculated percentages for Indiana - New sum:", percentage_sum$total, "%\n")
```

```
  Recalculated percentages for Indiana - New sum: 100 %
```

3.3.3.3 Standardize Terminology

Indiana uses “Caucasian” instead of “White”.

```
# Standardize racial terminology
in_clean <- in_clean %>%
  mutate(variable_detailed = case_when(
    variable_detailed == "Caucasian" ~ "White",
    TRUE ~ variable_detailed
  ))

cat("  Standardized terminology: 'Caucasian' → 'White'\n")
```

```
  Standardized terminology: 'Caucasian' → 'White'
```

3.3.3.4 Create Combined Total Profiles

Indiana provides separate totals for Convicted Offenders and Arrestees, but we need a Combined total to match the demographic percentages.

```
# Calculate Combined total from separate offender type totals
convicted_total <- in_clean %>%
  filter(offender_type == "Convicted Offender",
         variable_category == "total",
         variable_detailed == "total_profiles") %>%
  pull(value)

arrestee_total <- in_clean %>%
  filter(offender_type == "Arrestee",
         variable_category == "total",
         variable_detailed == "total_profiles") %>%
  pull(value)

combined_total <- convicted_total + arrestee_total

# Add Combined total to the data
combined_row <- data.frame(
  state = "Indiana",
  offender_type = "Combined",
  variable_category = "total",
  variable_detailed = "total_profiles",
  value = combined_total,
  value_type = "count",
  value_source = "calculated"
)

in_clean <- bind_rows(in_clean, combined_row)

cat(paste("Combined total profiles:", format(combined_total, big.mark = ","), "\n"))
cat("  Added Combined total profiles\n")
```

```
Combined total profiles: 300,741
  Added Combined total profiles
```

3.3.3.5 Calculate Counts from Percentages

Indiana only provides percentages for demographic categories. We calculate the actual counts using the Combined total.

```
# Calculate counts from percentages for Combined offender type
in_clean <- bind_rows(in_clean, calculate_counts_from_percentages(in_clean, "Indiana"))

cat("  Calculated demographic counts from percentages\n")

# Verify the calculations
cat("Category totals after calculating counts:\n")
verify_category_totals(in_clean) %>% kable() %>% kable_styling()
```

```
  Calculated demographic counts from percentages
Category totals after calculating counts:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	300741	300741	0
Combined	race	300741	300741	0

3.3.4 Verify Data Consistency

Final checks to ensure all data is now consistent and complete.

```
Final data consistency checks:
```

```
Counts consistent: TRUE
```

```
Percentages consistent: TRUE


Final data availability:


Race data: both


Gender data: both
```

3.3.5 Prepare for Combined Dataset

The cleaned data is formatted to match the master schema and appended to the `foia_combined` dataframe.

```r
# Prepare the cleaned data for the combined dataset
in_prepared <- prepare_state_for_combined(in_clean, "Indiana")

# Append to the master combined dataframe
foia_combined <- bind_rows(foia_combined, in_prepared)

cat(paste0("  Appended ", nrow(in_prepared), " Indiana rows to foia_combined\n"))
cat(paste0("  Total rows in foia_combined: ", nrow(foia_combined), "\n"))
```

```
  Appended 15 Indiana rows to foia_combined
  Total rows in foia_combined: 88
```

3.3.6 Document Metadata

The metadata is added with details on all processing steps performed.

```r
# Add Indiana to the metadata table using the helper function
add_state_metadata("Indiana", in_raw)

# Update metadata with QC results and processing notes
update_state_metadata("Indiana",
                      counts_ok = counts_consistent(in_clean),
                      percentages_ok = percentages_consistent(in_clean),
                      notes_text = "Converted string values to numeric; standardized 'Black' to
```

```
  Metadata added for: Indiana
  Metadata updated for: Indiana
```

3.3.7 Visualizations

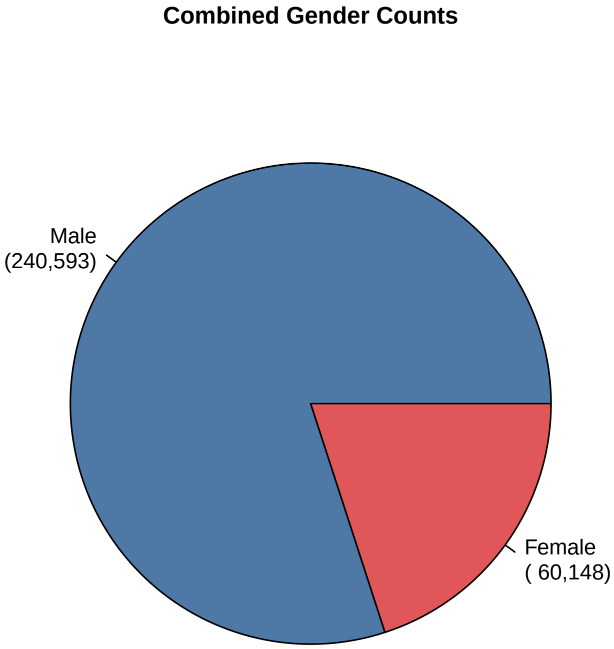


Figure 18: Indiana DNA Database Demographic Distributions

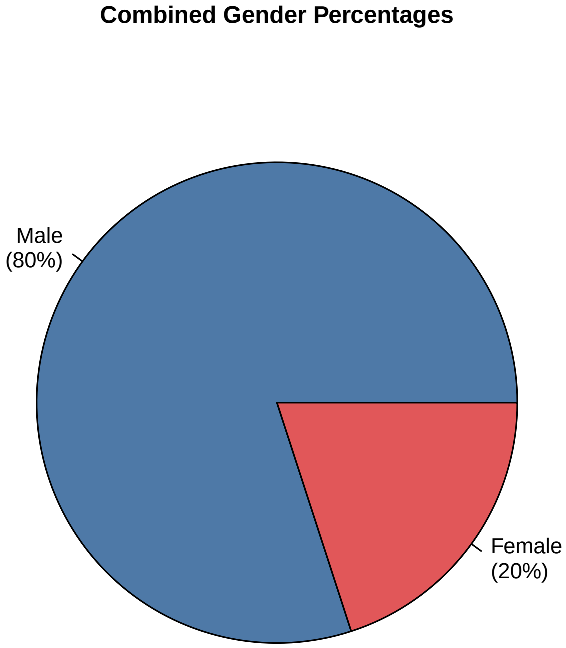


Figure 19: Indiana DNA Database Demographic Distributions

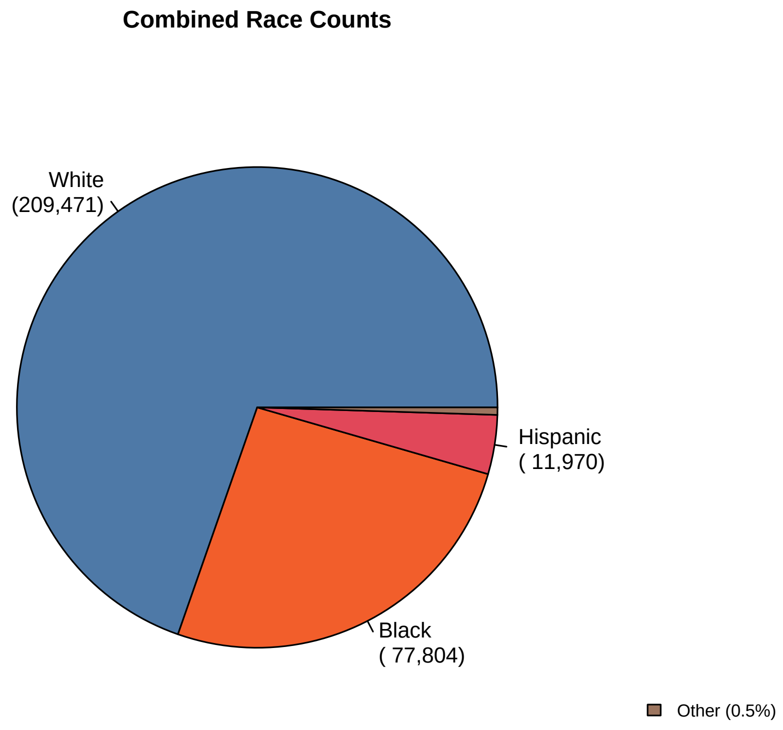


Figure 20: Indiana DNA Database Demographic Distributions

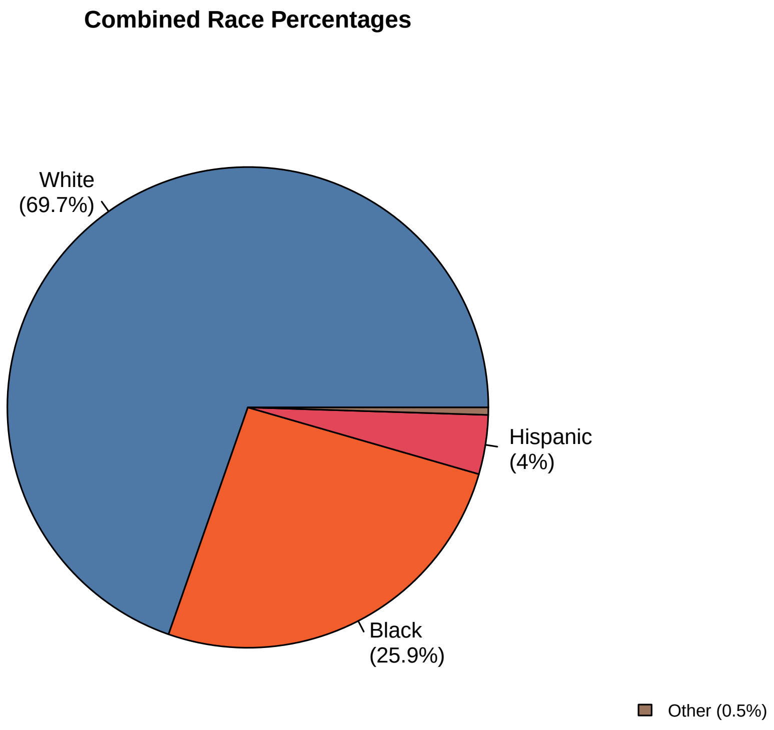


Figure 21: Indiana DNA Database Demographic Distributions

3.3.8 Summary Statistics

```
cat("Indiana DNA Database Summary:\n")
cat("=", strrep("=", 40), "\n")

# Total profiles by offender type
totals <- foia_combined %>%
  filter(state == "Indiana",
         variable_category == "total",
         variable_detailed == "total_profiles",
         value_type == "count") %>%
  select(offender_type, value, value_source) %>%
  mutate(value_formatted = format(value, big.mark = ","))

print(totals)

# Data completeness by value source
cat("\nData completeness by source:\n")
```

```r
completeness <- foia_combined %>%
  filter(state == "Indiana") %>%
  group_by(value_source) %>%
  summarise(n_values = n(), .groups = "drop")

print(completeness)

# Final verification
cat("\nFinal verification:\n")
cat(paste("Counts consistent:", counts_consistent(foia_combined %>% filter(state == "Indiana"))
cat(paste("Percentages consistent:", percentages_consistent(foia_combined %>% filter(state == "
```

```
Indiana DNA Database Summary:
= ========================================
# A tibble: 3 x 4
  offender_type       value value_source value_formatted
  <chr>               <dbl> <chr>        <chr>
1 Convicted Offender 279654 reported     "279,654"
2 Arrestee            21087 reported     " 21,087"
3 Combined           300741 calculated   "300,741"

Data completeness by source:
# A tibble: 2 x 2
  value_source n_values
  <chr>           <int>
1 calculated         11
2 reported            4

Final verification:
Counts consistent: TRUE
Percentages consistent: TRUE
```

3.3.9 Summary of Indiana Processing

Indiana data processing complete. The unique dataset required:

- **Data conversion**: String values converted to numeric, handling "<1" as 0.5
- **Terminology standardization**: "Caucasian" converted to "White"
- **Calculated additions**:
  - Combined total profiles across offender types
  - All demographic counts derived from reported percentages
- **Quality checks**: All counts and percentages pass consistency validation

- **Provenance tracking**: Clear distinction between reported and calculated values

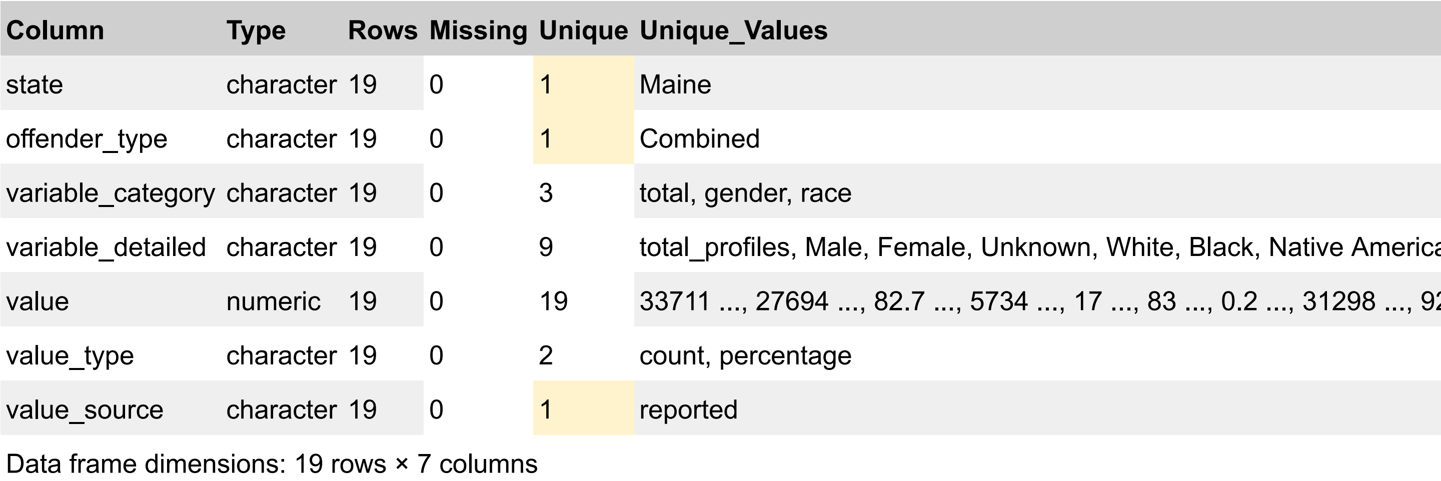
The Indiana data is now standardized and ready for cross-state analysis.

3.4 Maine (ME)

Overview: Maine provides comprehensive reporting with **both counts and percentages** for all gender and race categories across all offender types, including pre-calculated Combined totals. The data is complete and requires no processing.

3.4.1 Examine Raw Data

Establish a baseline understanding of the data exactly as it was received.



Column	Type	Rows	Missing	Unique	Unique_Values
state	character	19	0	1	Maine
offender_type	character	19	0	1	Combined
variable_category	character	19	0	3	total, gender, race
variable_detailed	character	19	0	9	total_profiles, Male, Female, Unknown, White, Black, Native America
value	numeric	19	0	19	33711 ..., 27694 ..., 82.7 ..., 5734 ..., 17 ..., 83 ..., 0.2 ..., 31298 ..., 92
value_type	character	19	0	2	count, percentage
value_source	character	19	0	1	reported

Data frame dimensions: 19 rows × 7 columns

3.4.2 Verify Data Consistency

Runs quality checks using the `verify_category_totals()`, `counts_consistent()`, and `percentages_consistent()` functions.

```
Verifying that demographic counts match reported totals:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	33711	33511	200
Combined	race	33711	33711	0

```
Counts consistency check on raw data:
```

```
All counts consistent: FALSE
```

```
Percentage consistency check on raw data:

All percentages sum to ~100%: TRUE
```

3.4.3 Address Data Gaps

3.4.3.1 Solve Percentages Inconsistency

Racial percentages summed to 99.9% instead of 100%

Proportional scaling was applied and `value_source` was updated to “calculated” for all adjusted values.

```r
# Start with the raw data
me_clean <- me_raw

# Adjust percentages to ensure they sum to 100% and mark as calculated
me_clean <- me_clean %>%
  group_by(value_type, variable_category) %>%
  mutate(
    value = ifelse(
      value_type == "percentage" & variable_category == "gender",
      value * (100 / sum(value, na.rm = TRUE)),
      value
    ),
    value_source = ifelse(
      value_type == "percentage" & variable_category == "gender",
      "calculated",
      value_source
    )
  ) %>%
  ungroup()

# Verify the new sum
percentage_sum <- me_clean %>%
  filter(value_type == "percentage" & variable_category == "gender") %>%
  summarise(total = sum(value, na.rm = TRUE))

cat(" Recalculated percentages for Maine - New sum:", round(percentage_sum$total, 2), "%\n")
```

```
 Recalculated percentages for Maine - New sum: 100 %
```

3.4.3.2 Recalculate Counts from Percentages

Maine’s reported gender counts sum were inconsistent with the `total_profiles`.

We removed existing gender count data and recalculated counts using percentage values and combined totals.

All recalculated values flagged with `value_source = "calculated"`

```
# Remove existing gender count rows to avoid duplication
me_clean <- me_clean %>%
  filter(!(variable_category == "gender" & value_type == "count"))

cat("  Removed existing gender count data\n")

me_gender <- me_clean %>%
    filter(variable_category == "gender" | variable_category == "total")

# Calculate counts from percentages for Combined offender type
me_gender <- calculate_counts_from_percentages(me_gender, "Maine")

# Append recalculated gender counts to the main dataset
me_clean <- bind_rows(me_clean, me_gender)

cat("  Calculated demographic counts from percentages\n")

# Verify the calculations
cat("Category totals after calculating counts:\n")
verify_category_totals(me_clean) %>% kable() %>% kable_styling()
```

```
  Removed existing gender count data
  Calculated demographic counts from percentages
Category totals after calculating counts:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	33711	33711	0
Combined	race	33711	33711	0

3.4.4 Verify Data Consistency

Final checks to ensure all data is now consistent and complete.

```
Final data consistency checks:

Counts consistent: TRUE

Percentages consistent: TRUE


Final data availability:

Race data: both

Gender data: both
```

3.4.5 Prepare for Combined Dataset

The Maine data is already complete and consistent. It is formatted to match the master schema and appended to the `foia_combined` dataframe.

```r
# Prepare the data for the combined dataset
me_prepared <- prepare_state_for_combined(me_clean, "Maine")

# Append to the master combined dataframe
foia_combined <- bind_rows(foia_combined, me_prepared)

cat(paste0("  Appended ", nrow(me_prepared), " Maine rows to foia_combined\n"))
cat(paste0("  Total rows in foia_combined: ", nrow(foia_combined), "\n"))
```

```
  Appended 19 Maine rows to foia_combined
  Total rows in foia_combined: 107
```

3.4.6 Document Metadata

The metadata is added with a note that the data was complete and required no processing.

```r
# Add Maine to the metadata table using the helper function
add_state_metadata("Maine", me_raw)

# Update metadata with QC results
update_state_metadata("Maine",
                      counts_ok = counts_consistent(me_clean),
                      percentages_ok = percentages_consistent(me_clean),
                      notes_text = "Complete dataset provided with both counts and percentages
```

```
  Metadata added for: Maine
  Metadata updated for: Maine
```

3.4.7 Visualizations

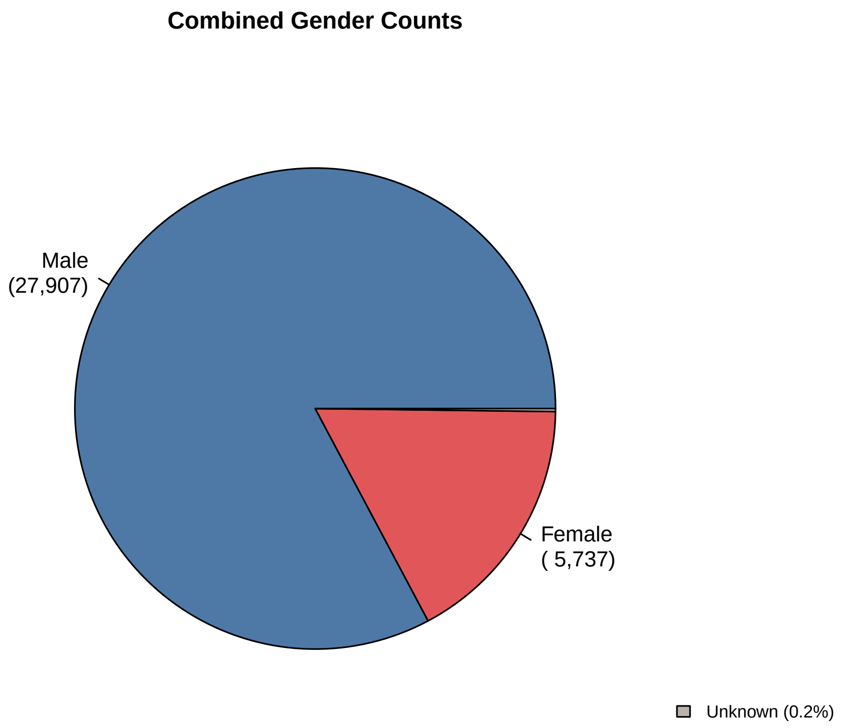


Figure 22: Maine DNA Database Demographic Distributions

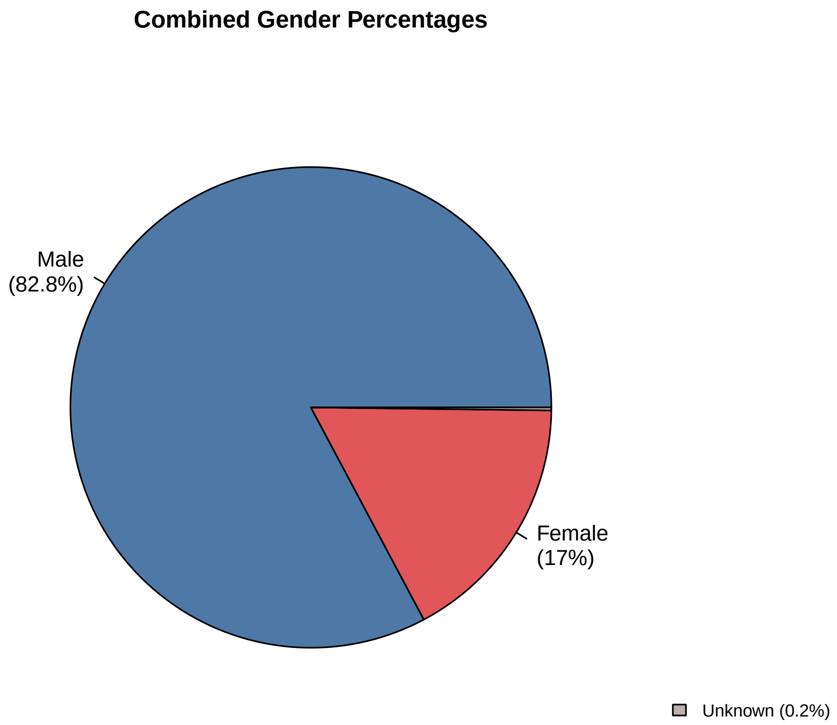


Figure 23: Maine DNA Database Demographic Distributions

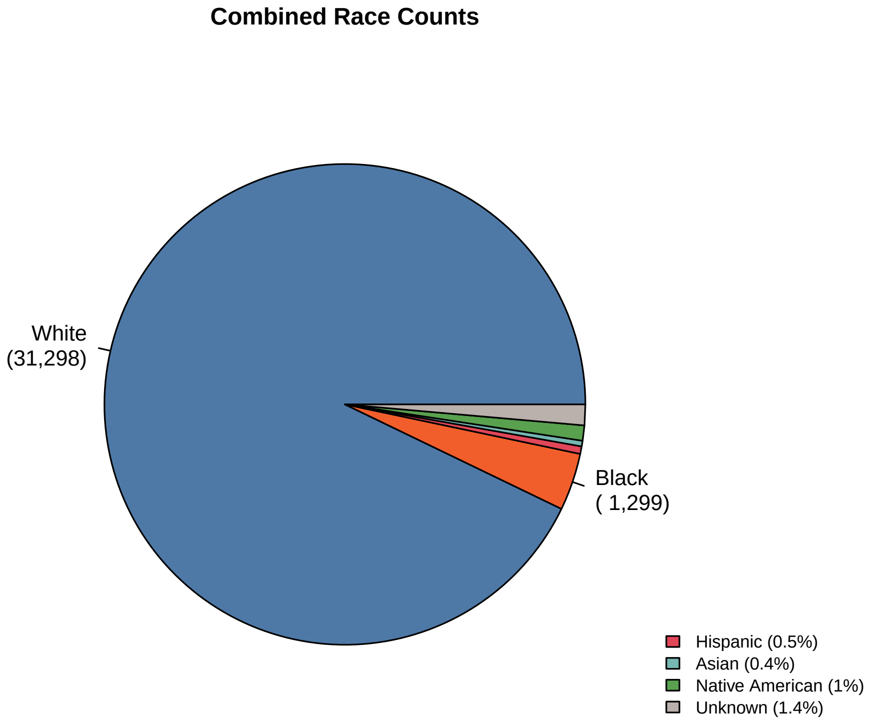


Figure 24: Maine DNA Database Demographic Distributions

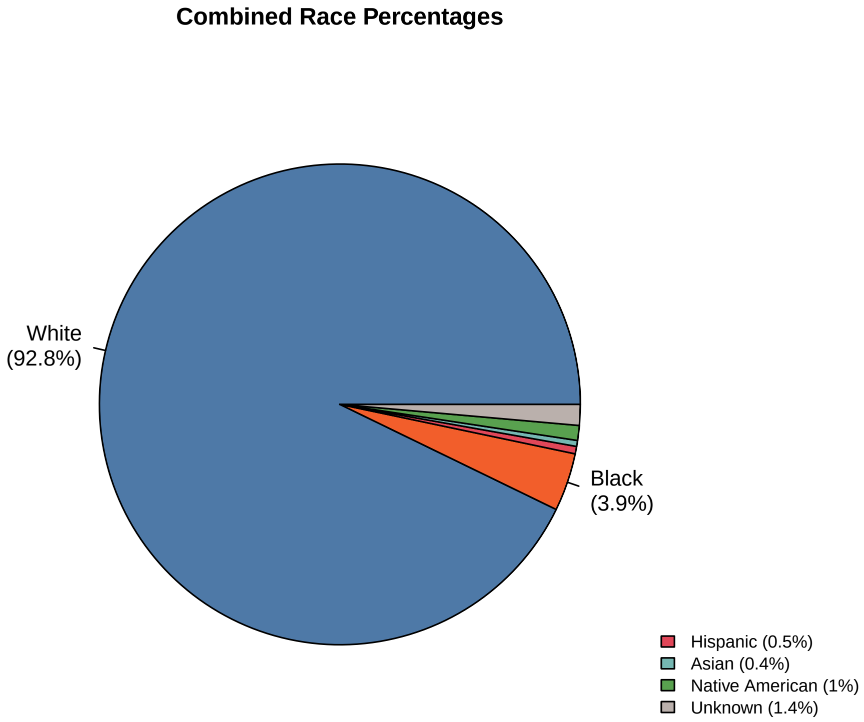


Figure 25: Maine DNA Database Demographic Distributions

3.4.8 Summary Statistics

```r
cat("Maine DNA Database Summary:\n")
cat("=", strrep("=", 40), "\n")

# Total profiles by offender type
totals <- foia_combined %>%
  filter(state == "Maine",
         variable_category == "total",
         variable_detailed == "total_profiles",
         value_type == "count") %>%
  select(offender_type, value) %>%
  mutate(value_formatted = format(value, big.mark = ","))

print(totals)

# Data completeness
cat("\nData completeness:\n")
```

```r
completeness <- foia_combined %>%
  filter(state == "Maine") %>%
  group_by(offender_type, value_source) %>%
  summarise(n_values = n(), .groups = "drop")

print(completeness)

# Final verification
cat("\nFinal verification:\n")
cat(paste("Counts consistent:", counts_consistent(foia_combined %>% filter(state == "Maine")),
cat(paste("Percentages consistent:", percentages_consistent(foia_combined %>% filter(state ==
```

```
Maine DNA Database Summary:
= ========================================
# A tibble: 1 x 3
  offender_type value value_formatted
  <chr>         <dbl> <chr>
1 Combined      33711 33,711

Data completeness:
# A tibble: 2 x 3
  offender_type value_source n_values
  <chr>         <chr>           <int>
1 Combined      calculated          6
2 Combined      reported           13

Final verification:
Counts consistent: TRUE
Percentages consistent: TRUE
```

3.4.9 Summary of Maine Processing

Maine data processing complete. The dataset is exemplary and required no adjustments:

- **Reported data**: Both **counts and percentages** for all Convicted Offender, Arrestee, and Combined categories
- **No calculated additions needed**: All values are sourced directly from the state report (`value_source = "reported"`)
- **Quality checks**: All counts and percentages pass consistency validation
- **Provenance tracking**: All values maintain their original `value_source` as "reported"

The Maine data is now standardized and ready for cross-state analysis.

3.5 Nevada (NV)

Overview: Nevada provides **both counts and percentages** for gender and race categories but uses non-standard terminology that requires conversion for consistency with our schema.

3.5.1 Examine Raw Data

Establish a baseline understanding of the data exactly as it was received.

Column	Type	Rows	Missing	Unique	Unique_Values
state	character	21	0	1	Nevada
offender_type	character	21	0	4	All, Arrestee, Convicted Offender, Combined
variable_category	character	21	0	3	total, gender, race
variable_detailed	character	21	0	9	total_flags, total_profiles, Female, Male, Unknown, White, American
value	numeric	21	0	21	344097 ..., 185074 ..., 53.785 ..., 159023 ..., 46.215 ..., 63287 ..., 18.
value_type	character	21	0	2	count, percentage
value_source	character	21	0	1	reported

Data frame dimensions: 21 rows × 7 columns

3.5.2 Verify Data Consistency

Initial check reveals Nevada's non-standard terminology.

```
Initial data availability:

Race data: both

Gender data: both


Non-standard terminology found:

Offender types: All, Arrestee, Convicted Offender, Combined
```

3.5.3 Address Data Gaps

3.5.3.1 Standardize Terminology

Nevada uses "All" instead of "Combined", "total_flags" instead of "total_profiles" and "American Indian" instead of "Native American".

```
# Start with raw data
nv_clean <- nv_raw

# Standardize offender types and racial terminology
nv_clean <- nv_clean %>%
  mutate(
    offender_type = case_when(
      offender_type == "All" ~ "Combined",
      TRUE ~ offender_type
    ),
    variable_detailed = case_when(
      variable_detailed == "total_flags" ~ "total_profiles",
      TRUE ~ variable_detailed
    ),
    variable_detailed = case_when(
      variable_detailed == "American Indian" ~ "Native American",
      TRUE ~ variable_detailed
    )
  )

cat("  Standardized terminology:\n")
cat("  - 'All' → 'Combined'\n")
cat("  - 'total_flags' → 'total_profiles'\n")
cat("  - 'American Indian' → 'Native American'\n")
```

```
  Standardized terminology:
  - 'All' → 'Combined'
  - 'total_flags' → 'total_profiles'
  - 'American Indian' → 'Native American'
```

3.5.3.2 Verify Consistency

Now that the offender types are standardized, we can verify the counts and percentages.

```
Verifying that demographic counts match reported totals:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	344097	344097	0
Combined	race	344097	344097	0

```
Counts consistency check on raw data:
```

```
All counts consistent: TRUE
```

```
Percentage consistency check on raw data:

All percentages sum to ~100%: TRUE

Sum of 'race' percentages: 100 %

Sum of 'gender' percentages: 100 %
```

3.5.4 Prepare for Combined Dataset

The cleaned data is formatted to match the master schema and appended to the `foia_combined` dataframe.

```r
# Prepare the cleaned data for the combined dataset
nv_prepared <- prepare_state_for_combined(nv_clean, "Nevada")

# Append to the master combined dataframe
foia_combined <- bind_rows(foia_combined, nv_prepared)

cat(paste0("  Appended ", nrow(nv_prepared), " Nevada rows to foia_combined\n"))
cat(paste0("  Total rows in foia_combined: ", nrow(foia_combined), "\n"))
```

```
  Appended 21 Nevada rows to foia_combined
  Total rows in foia_combined: 128
```

3.5.5 Document Metadata

The metadata is added with details on the terminology standardization performed.

```r
# Add Nevada to the metadata table using the helper function
add_state_metadata("Nevada", nv_raw)

# Update metadata with QC results and processing notes
update_state_metadata("Nevada",
                      counts_ok = counts_consistent(nv_clean),
                      percentages_ok = percentages_consistent(nv_clean),
                      notes_text = "Standardized terminology: 'All' to 'Combined' and 'America
```

```
  Metadata added for: Nevada
  Metadata updated for: Nevada
```

3.5.6 Visualizations

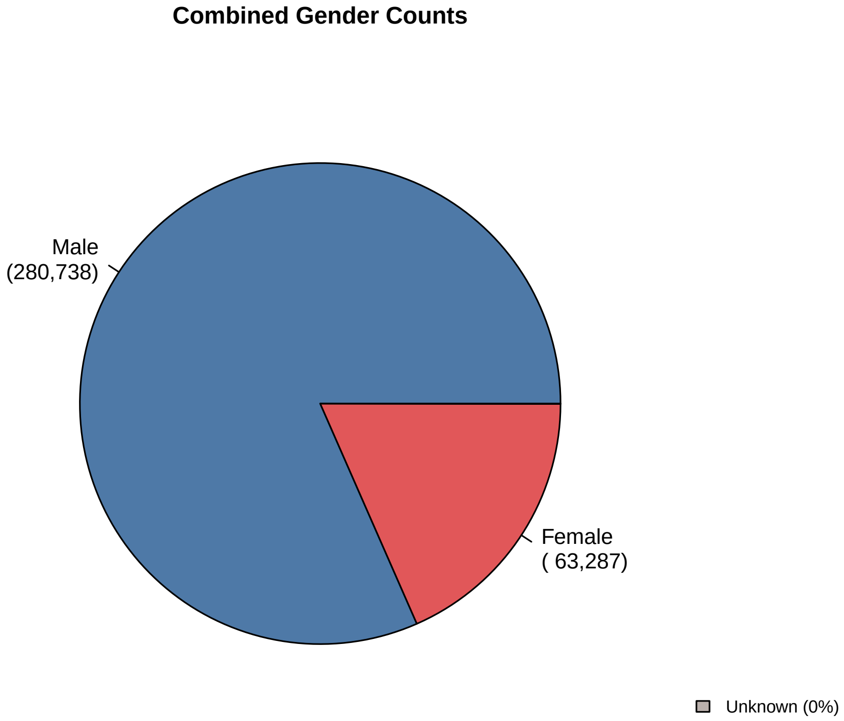


Figure 26: Nevada DNA Database Demographic Distributions

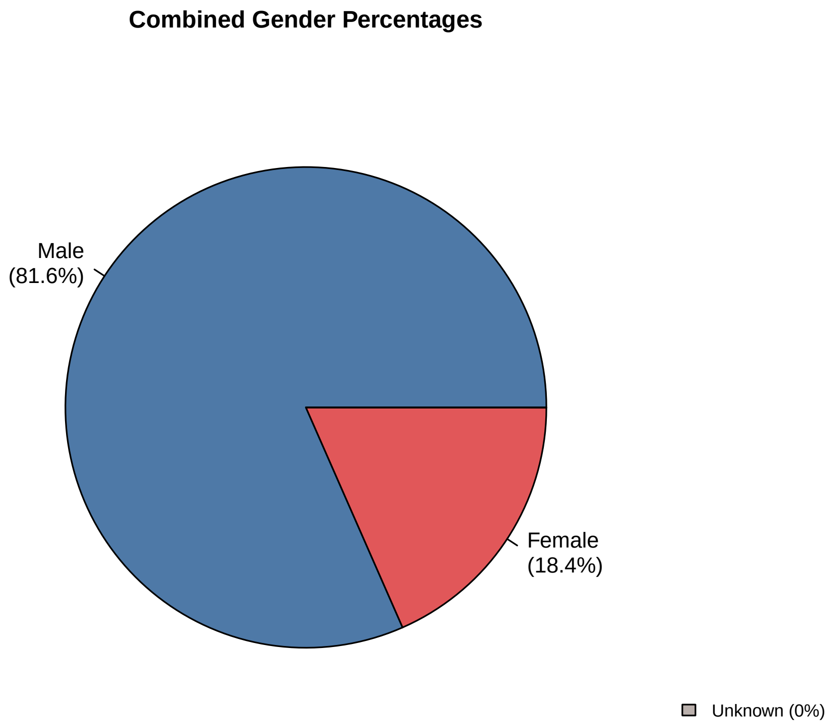


Figure 27: Nevada DNA Database Demographic Distributions

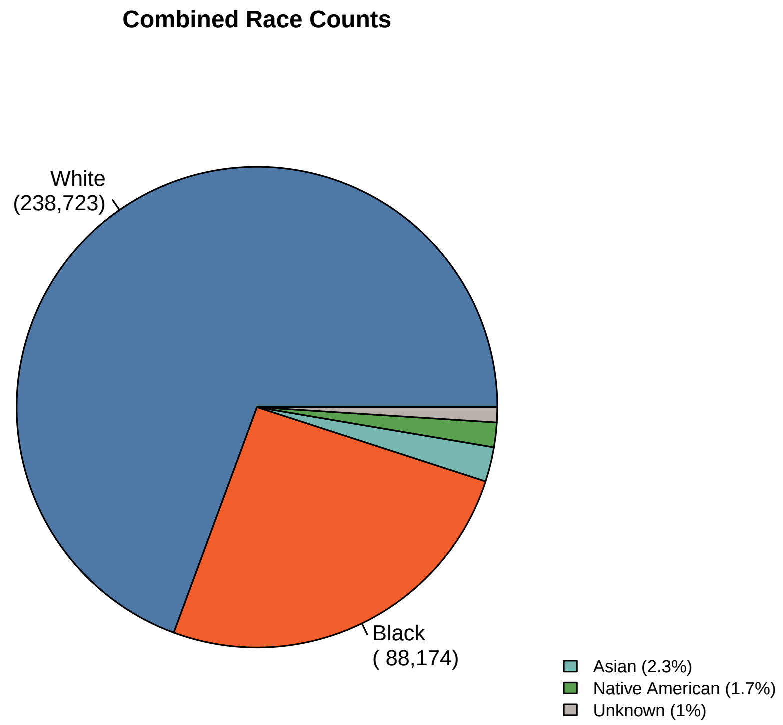


Figure 28: Nevada DNA Database Demographic Distributions

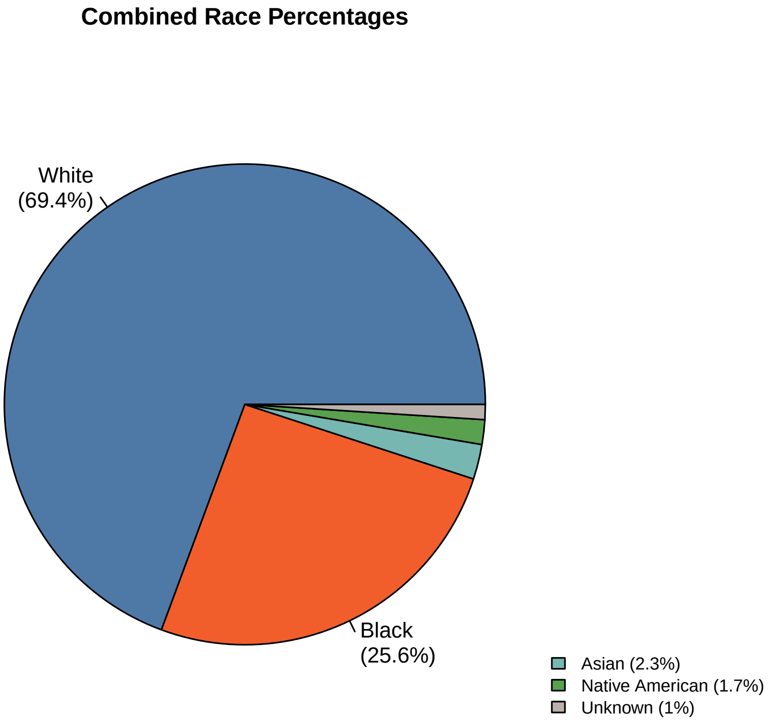


Figure 29: Nevada DNA Database Demographic Distributions

3.5.7 Summary Statistics

```
cat("Nevada DNA Database Summary:\n")
cat("=", strrep("=", 40), "\n")

# Total profiles by offender type
totals <- foia_combined %>%
  filter(state == "Nevada",
         variable_category == "total",
         variable_detailed == "total_profiles",
         value_type == "count") %>%
  select(offender_type, value) %>%
  mutate(value_formatted = format(value, big.mark = ","))

print(totals)

# Data completeness
cat("\nData completeness:\n")
```

```
completeness <- foia_combined %>%
  filter(state == "Nevada") %>%
  group_by(offender_type, value_source) %>%
  summarise(n_values = n(), .groups = "drop")

print(completeness)

# Final verification
cat("\nFinal verification:\n")
cat(paste("Counts consistent:", counts_consistent(foia_combined %>% filter(state == "Nevada"))
cat(paste("Percentages consistent:", percentages_consistent(foia_combined %>% filter(state == '
```

```
Nevada DNA Database Summary:
= ========================================
# A tibble: 3 x 3
  offender_type       value value_formatted
  <chr>               <dbl> <chr>
1 Combined           344097 344,097
2 Arrestee           185074 185,074
3 Convicted Offender 159023 159,023

Data completeness:
# A tibble: 3 x 3
  offender_type      value_source n_values
  <chr>              <chr>           <int>
1 Arrestee           reported            2
2 Combined           reported           17
3 Convicted Offender reported            2

Final verification:
Counts consistent: TRUE
Percentages consistent: FALSE
```

3.5.8 Summary of Nevada Processing

Nevada data processing complete. The dataset required minimal adjustments:

- **Terminology standardization**:
  - "All" → "Combined" (offender type)
  - "American Indian" → "Native American" (race category)
- **Reported data**: Both counts and percentages for all categories
- **Quality checks**: All counts and percentages pass consistency validation

- **Provenance tracking**: All values maintain **`value_source = "reported"`** as only terminology changes were made

The Nevada data is now standardized and ready for cross-state analysis.

3.6 South Dakota (SD)

Overview: South Dakota provides the most comprehensive reporting with **both counts and percentages** for all standard categories plus unique intersectional gender×race data. Minor terminology standardization is required for consistency.

3.6.1 Examine Raw Data

Establish a baseline understanding of the data exactly as it was received.

Column	Type	Rows	Missing	Unique	Unique_Values
state	character	41	0	1	South Dakota
offender_type	character	41	0	1	Combined
variable_category	character	41	0	4	total, gender, race, gender_race
variable_detailed	character	41	0	21	total_profiles ..., Male ..., Female ..., Asian ..., Black ..., Hispanic ..., N
value	numeric	41	0	38	67753 ..., 51197 ..., 75.56 ..., 16556 ..., 24.44 ..., 5 ..., 0.08 ..., 4041 .
value_type	character	41	0	2	count, percentage
value_source	character	41	0	1	reported

Data frame dimensions: 41 rows × 7 columns

3.6.2 Gender-race intersection analysis

Since South Dakota is the only state that reported gender-race intersection data, we can analyze it in detail.

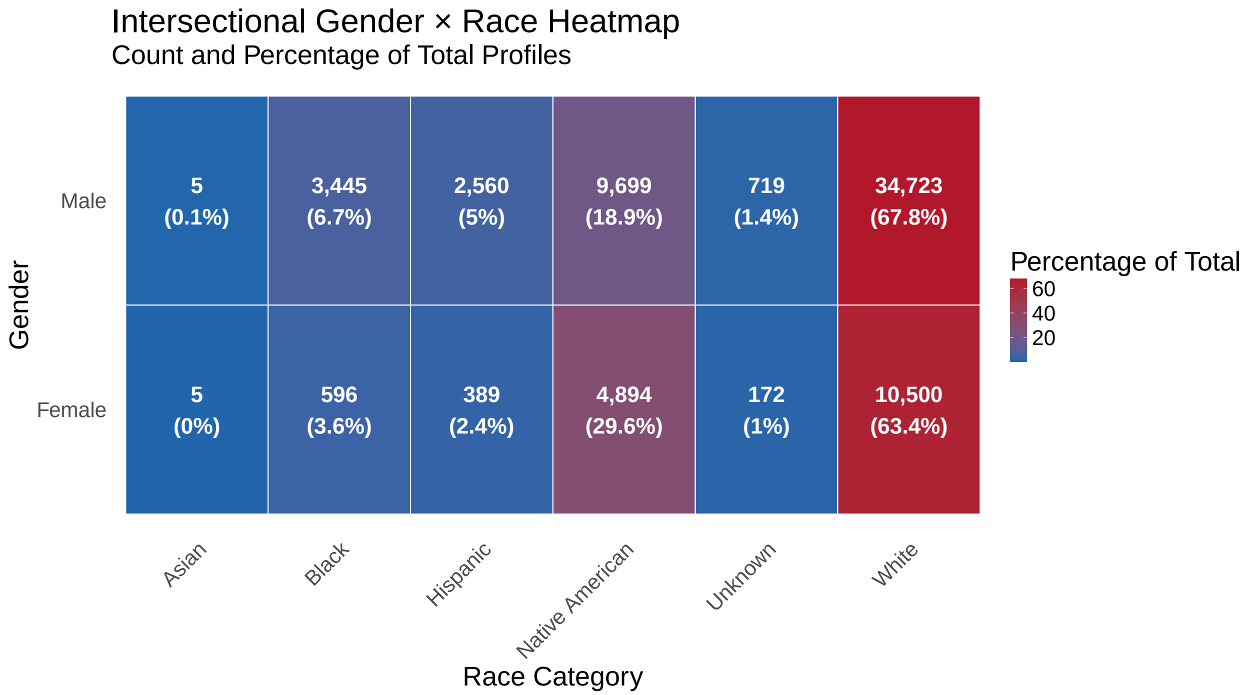
3.6.3 Verify Data Consistency

Initial check reveals South Dakota's comprehensive data structure with some non-standard terminology.

```
Initial data availability:

Race data: both

Gender data: both
```



(a) South Dakota Intersectional Gender × Race Analysis

```
Non-standard terminology found:

Race terms: Asian, Black, Hispanic, Native American, Other/Unknown, White/Caucasian

Verifying that demographic counts match reported totals:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	67753	67753	0
Combined	race	67753	67702	51

```
Counts consistency check on raw data:

All counts consistent: FALSE


Percentage consistency check on raw data:

All percentages sum to ~100%: TRUE

Sum of 'race' percentages: 100 %

Sum of 'gender' percentages: 100 %
```

3.6.4 Address Data Gaps

3.6.4.1 Standardize Terminology

South Dakota uses “White/Caucasian” and “Other/Unknown” which need standardization.

```r
# Standardize racial terminology
sd_clean <- sd_clean %>%
  mutate(
    variable_detailed = case_when(
      variable_detailed == "White/Caucasian" ~ "White",
      variable_detailed == "Other/Unknown" ~ "Unknown",
      TRUE ~ variable_detailed
    )
  )

cat("  Standardized terminology:\n")
cat("  - 'White/Caucasian' → 'White'\n")
cat("  - 'Other/Unknown' → 'Unknown'\n")

# Verify the changes
cat("\nRace categories after standardization:\n")
sd_clean %>%
  filter(variable_category == "race") %>%
  distinct(variable_detailed) %>%
  pull() %>%
  paste(collapse = ", ") %>%
  cat()
```

```
  Standardized terminology:
  - 'White/Caucasian' → 'White'
  - 'Other/Unknown' → 'Unknown'

Race categories after standardization:
Asian, Black, Hispanic, Native American, Unknown, White
```

3.6.4.2 Recalculate Counts from Percentages

South Dakota’s reported race counts sum were inconsistent with the `total_profiles`.

We removed existing gender count data and recalculated counts using percentage values and combined totals.

All recalculated values flagged with `value_source = "calculated"`

```r
# Remove existing gender count rows to avoid duplication
sd_clean <- sd_clean %>%
  filter(!(variable_category == "race" & value_type == "count"))
```

```
cat("  Removed existing race count data\n")

sd_race <- sd_clean %>%
    filter(variable_category == "race" | variable_category == "total")

# Calculate counts from percentages for Combined offender type
sd_race <- calculate_counts_from_percentages(sd_race, "South Dakota")

# Append recalculated race counts to the main dataset
sd_clean <- bind_rows(sd_clean, sd_race)

cat("  Calculated demographic counts from percentages\n")

# Verify the calculations
cat("Category totals after calculating counts:\n")
verify_category_totals(sd_clean) %>% kable() %>% kable_styling()
```

```
  Removed existing race count data
  Calculated demographic counts from percentages
Category totals after calculating counts:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	67753	67753	0
Combined	race	67753	67752	1

We handled this diffence of 1 by adding it to the most representative race (White).

```
# Handle the difference of 1 by adding it to the most representative race
sd_clean <- sd_clean %>%
  mutate(value = ifelse(variable_detailed == "White" & value_type == "count", value + 1, value
```

3.6.5 Verify Data Consistency

Final checks to ensure standardization didn't affect data integrity.

```
Final data consistency checks after standardization:

Verifying that demographic counts match reported totals:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Combined	gender	67753	67753	0

Combined	race	67753	67753	0

```
Counts consistency check:

All counts consistent: TRUE


Percentage consistency check:

All percentages sum to ~100%: TRUE
```

3.6.6 Prepare for Combined Dataset

The cleaned data is formatted to match the master schema and appended to the `foia_combined` dataframe.

```r
# Prepare the cleaned data for the combined dataset
sd_prepared <- prepare_state_for_combined(sd_clean, "South Dakota")

# Append to the master combined dataframe
foia_combined <- bind_rows(foia_combined, sd_prepared)

cat(paste0("  Appended ", nrow(sd_prepared), " South Dakota rows to foia_combined\n"))
cat(paste0("  Total rows in foia_combined: ", nrow(foia_combined), "\n"))

# Show the comprehensive nature of South Dakota's data
cat("\nSouth Dakota's comprehensive data structure:\n")
sd_prepared %>%
  group_by(variable_category) %>%
  summarise(n_rows = n(), .groups = "drop") %>%
  kable() %>% kable_styling()
```

```
  Appended 17 South Dakota rows to foia_combined
  Total rows in foia_combined: 145

South Dakota's comprehensive data structure:
```

variable_category	n_rows
gender	4
race	12
total	1

3.6.7 Document Metadata

The metadata is added with details on South Dakota's comprehensive reporting and the terminology standardization performed.

```
# Add South Dakota to the metadata table using the helper function
add_state_metadata("South Dakota", sd_raw)

# Update metadata with QC results and processing notes
update_state_metadata("South Dakota",
                      counts_ok = counts_consistent(sd_clean),
                      percentages_ok = percentages_consistent(sd_clean),
                      notes_text = "Standardized terminology: 'White/Caucasian' to 'White' and
```

```
Metadata added for: South Dakota
Metadata updated for: South Dakota
```

3.6.8 Visualizations

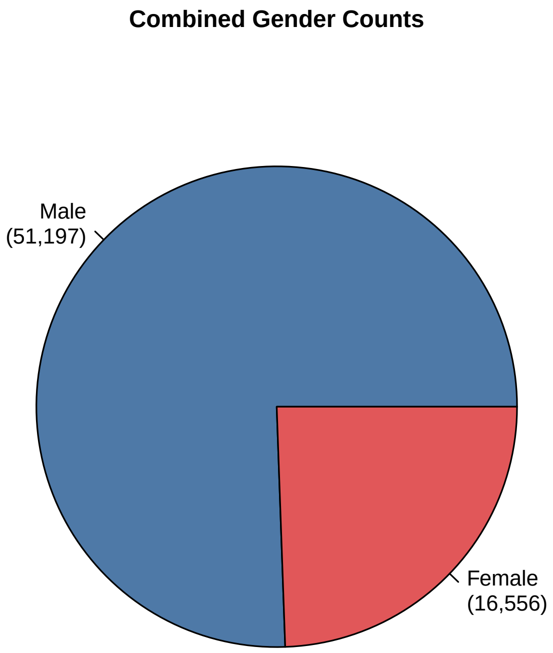


Figure 31: South Dakota DNA Database Demographic Distributions

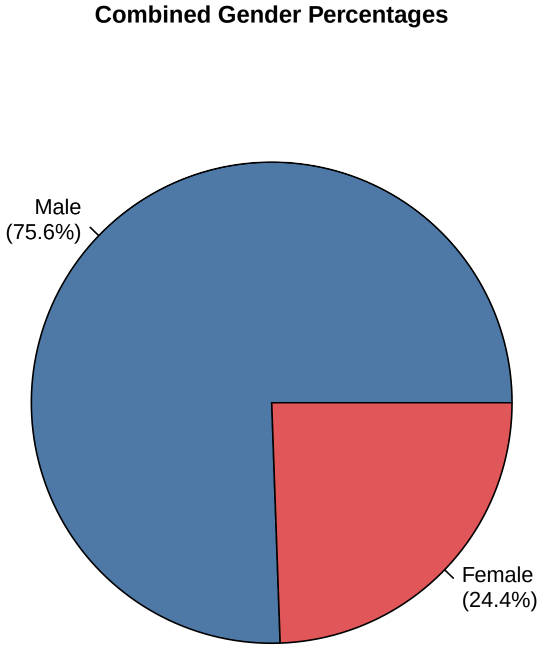


Figure 32: South Dakota DNA Database Demographic Distributions

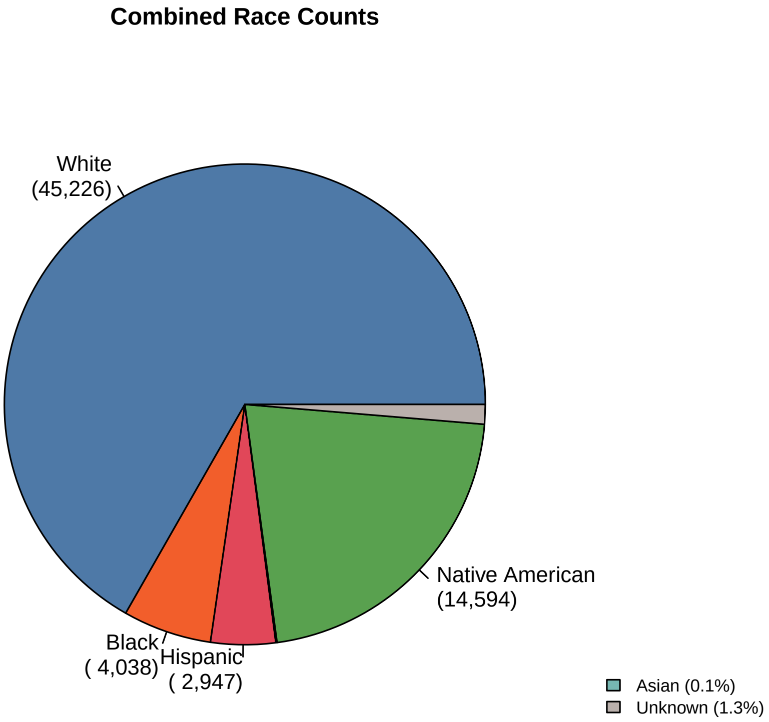


Figure 33: South Dakota DNA Database Demographic Distributions

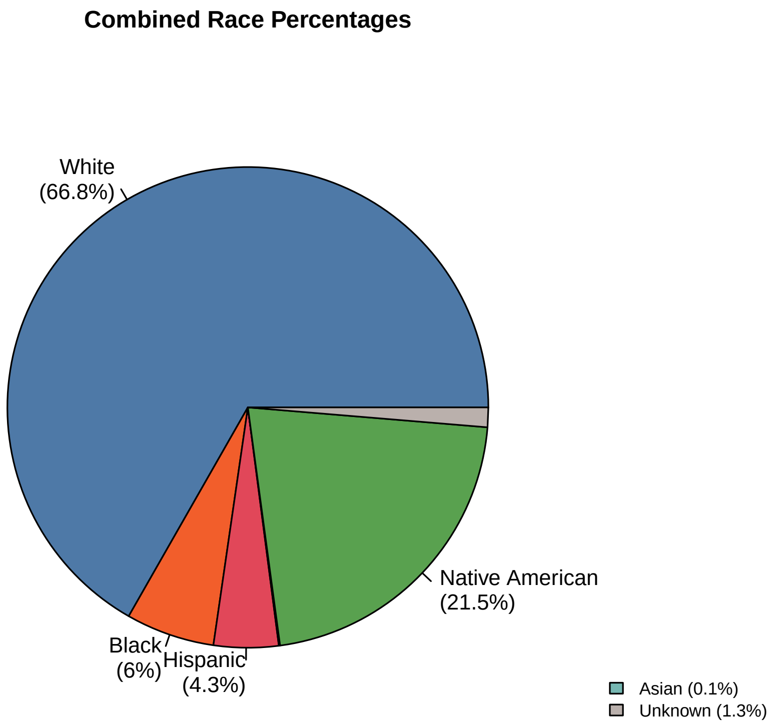


Figure 34: South Dakota DNA Database Demographic Distributions

3.6.9 Summary Statistics

```
cat("South Dakota DNA Database Summary:\n")
cat("=", strrep("=", 40), "\n")

# Total profiles by offender type
totals <- foia_combined %>%
  filter(state == "South Dakota",
         variable_category == "total",
         variable_detailed == "total_profiles",
         value_type == "count") %>%
  select(offender_type, value) %>%
  mutate(value_formatted = format(value, big.mark = ","))

print(totals)

# Data completeness by category
cat("\nData completeness by category:\n")
```

```r
completeness <- foia_combined %>%
  filter(state == "South Dakota") %>%
  group_by(variable_category) %>%
  summarise(n_values = n(), .groups = "drop")

print(completeness)

# Final verification
cat("\nFinal verification:\n")
cat(paste("Counts consistent:", counts_consistent(foia_combined %>% filter(state == "South Dak
cat(paste("Percentages consistent:", percentages_consistent(foia_combined %>% filter(state == "
```

```
South Dakota DNA Database Summary:
= ========================================
# A tibble: 1 x 3
  offender_type value value_formatted
  <chr>         <dbl> <chr>
1 Combined      67753 67,753

Data completeness by category:
# A tibble: 3 x 2
  variable_category n_values
  <chr>                <int>
1 gender                   4
2 race                    12
3 total                    1

Final verification:
Counts consistent: TRUE
Percentages consistent: TRUE
```

3.6.10 Summary of South Dakota Processing

South Dakota data processing complete. The state provided exemplary data with minimal adjustments needed:

- **Terminology standardization**:
  - “White/Caucasian” → “White”
  - “Other/Unknown” → “Unknown”
- **Comprehensive reporting**: Standard demographics plus unique gender×race intersectional data
- **Reported data**: Both counts and percentages for all categories
- **Quality checks**: All counts and percentages pass consistency validation

- **Provenance tracking**: All values maintain `value_source = "reported"` as only terminology changes were made

South Dakota's data is now standardized and ready for cross-state analysis.

3.7 Texas (TX)

Overview: Texas provides **counts only** for gender and race categories. The Male gender is missing in the dataset. The state uses non-standard terminology that requires conversion and needs Combined totals and percentages calculated.

3.7.1 Examine Raw Data

Establish a baseline understanding of the data exactly as it was received.

Column	Type	Rows	Missing	Unique	Unique_Values
state	character	16	0	1	Texas
offender_type	character	16	0	2	Offenders, Arrestee
variable_category	character	16	0	3	total, gender, race
variable_detailed	character	16	0	8	total_profiles, Female, Asian, African American, Caucasian, Hispanic
value	numeric	16	0	16	845322 ..., 73631 ..., 121434 ..., 18721 ..., 3361 ..., 254366 ..., 3090
value_type	character	16	0	1	count
value_source	character	16	0	1	reported

Data frame dimensions: 16 rows × 7 columns

3.7.2 Verify Data Consistency

Initial checks reveal Texas's reporting structure and terminology differences.

```
Initial data availability:

Race data: counts

Gender data: counts


Non-standard terminology found:

Offender types: Offenders, Arrestee

Race terms: Asian, African American, Caucasian, Hispanic, Native American, Other
```

3.7.3 Address Data Gaps

3.7.3.1 Add Missing Male category

Texas data reports only Female counts explicitly. We calculated Male counts by subtracting Female counts from total profiles, assuming binary gender classification in the dataset.

```r
# First, let's examine the current structure of gender data
gender_data <- tx_raw %>%
  filter(variable_category == "gender")

cat("Current gender structure:\n")
print(unique(gender_data$variable_detailed))

# Get total profiles for each offender type
total_profiles <- tx_raw %>%
  filter(variable_category == "total" & variable_detailed == "total_profiles") %>%
  select(offender_type, total_value = value)

# Join total profiles with gender data
gender_with_totals <- gender_data %>%
  left_join(total_profiles, by = "offender_type")

# Create Male entries for each offender type
male_entries <- gender_with_totals %>%
  filter(variable_detailed == "Female") %>%
  mutate(
    variable_detailed = "Male",
    value = total_value - value,
    value_source = "calculated",
    total_value = NULL
  )

# Add these entries to the original dataset
tx_raw_with_male <- tx_raw %>%
  bind_rows(male_entries)

# Update the tx_raw object
tx_clean <- tx_raw_with_male

# Verify the addition
cat("\nAfter adding Male entries - gender categories:\n")
print(unique(tx_clean %>%
       filter(variable_category == "gender") %>%
       pull(variable_detailed)))
```

```
Current gender structure:
[1] "Female"
```

```
After adding Male entries - gender categories:
[1] "Female" "Male"
```

3.7.3.2 Standardize Terminology

Texas uses “Offenders” instead of “Convicted Offender” and “Caucasian” instead of “White”.

```r
# Standardize offender types and racial terminology
tx_clean <- tx_clean %>%
  mutate(
    offender_type = case_when(
      offender_type == "Offenders" ~ "Convicted Offender",
      TRUE ~ offender_type
    ),
    variable_detailed = case_when(
      variable_detailed == "Caucasian" ~ "White",
      variable_detailed == "African American" ~ "Black",
      TRUE ~ variable_detailed
    )
  )

cat("  Standardized terminology:\n")
cat("   - 'Offenders' → 'Convicted Offender'\n")
cat("   - 'Caucasian' → 'White'\n")
cat("   - 'African American' → 'Black'\n")
cat(paste("Offender types after standardization:", paste(sort(unique(tx_clean$offender_type)),
```

```
  Standardized terminology:
   - 'Offenders' → 'Convicted Offender'
   - 'Caucasian' → 'White'
   - 'African American' → 'Black'
Offender types after standardization: Arrestee, Convicted Offender
```

3.7.3.3 Create Unknown Category

Texas race count is inconsistent, with a significant number of profiles not reported in any racial category.

Unknown category was created to account for these missing profiles.

The calculated values are added with a **`value_source = "calculated"`** tag to maintain transparency about what was provided versus what was derived.

```r
# Add Unknown race category to reconcile totals
tx_clean <- fill_demographic_gaps(tx_clean)

# Verify the fix
```

```r
cat("Category totals after adding Unknown race category:\n")
verify_category_totals(tx_clean) %>% kable() %>% kable_styling()

cat("\nCounts consistency after adding Unknown:\n")
cat(paste("All counts consistent:", counts_consistent(tx_clean), "\n"))
```

```
Category totals after adding Unknown race category:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Arrestee	gender	73631	73631	0
Arrestee	race	73631	73631	0
Convicted Offender	gender	845322	845322	0
Convicted Offender	race	845322	845322	0

```
Counts consistency after adding Unknown:
All counts consistent: TRUE
```

3.7.3.4 Create Combined Totals

Texas only reported data for "Convicted Offender" and "Arrestee" separately. We calculate Combined totals.

```r
# Calculate Combined totals using helper function
tx_clean <- add_combined(tx_clean)

cat("  Created Combined totals for Texas\n")

# Show the Combined total
combined_total <- tx_clean %>%
  filter(offender_type == "Combined",
         variable_category == "total",
         variable_detailed == "total_profiles") %>%
  pull(value)

cat(paste("Combined total profiles:", format(combined_total, big.mark = ","), "\n"))
```

```
  Created Combined totals for Texas
Combined total profiles: 918,953
```

3.7.3.5 Calculate Percentages

Transforms the data from counts into percentages for comparative analysis.

```r
# Derive percentages from counts
tx_clean <- add_percentages(tx_clean)

cat("  Added percentages for all demographic categories\n")

# Check percentage consistency
cat("Percentage consistency check:\n")
cat(paste("All percentages sum to ~100%:", percentages_consistent(tx_clean), "\n\n"))

# Show current data availability
cat("Final data availability:\n")
cat(paste("Race data:", report_status(tx_clean, "race"), "\n"))
cat(paste("Gender data:", report_status(tx_clean, "gender"), "\n"))
```

```
  Added percentages for all demographic categories
Percentage consistency check:
All percentages sum to ~100%: TRUE

Final data availability:
Race data: both
Gender data: both
```

3.7.4 Verify Data Consistency

Final checks to ensure all processing maintained data integrity.

```
Final data consistency checks:
```

```
Verifying that demographic counts match reported totals:
```

offender_type	variable_category	total_profiles	sum_counts	difference
Arrestee	gender	73631	73631	0
Arrestee	race	73631	73631	0
Combined	gender	918953	918953	0
Combined	race	918953	918953	0
Convicted Offender	gender	845322	845322	0
Convicted Offender	race	845322	845322	0

```
Counts consistency check:
```

```
All counts consistent: TRUE
```

```
Percentage consistency check:
```

```
All percentages sum to ~100%: TRUE
```

3.7.5 Prepare for Combined Dataset

The cleaned data is formatted to match the master schema and appended to the `foia_combined` dataframe.

```r
# Prepare the cleaned data for the combined dataset
tx_prepared <- prepare_state_for_combined(tx_clean, "Texas")

# Append to the master combined dataframe
foia_combined <- bind_rows(foia_combined, tx_prepared)

cat(paste0("  Appended ", nrow(tx_prepared), " Texas rows to foia_combined\n"))
cat(paste0("  Total rows in foia_combined: ", nrow(foia_combined), "\n"))
```

```
  Appended 57 Texas rows to foia_combined
  Total rows in foia_combined: 202
```

3.7.6 Document Metadata

The metadata is added with details on all processing steps performed.

```r
# Add Texas to the metadata table using the helper function
add_state_metadata("Texas", tx_raw)

# Update metadata with QC results and processing notes
update_state_metadata("Texas",
                      counts_ok = counts_consistent(tx_clean),
                      percentages_ok = percentages_consistent(tx_clean),
                      notes_text = "Standardized terminology: 'Offenders' to 'Convicted Offend
```

```
  Metadata added for: Texas
  Metadata updated for: Texas
```

3.7.7 Visualizations

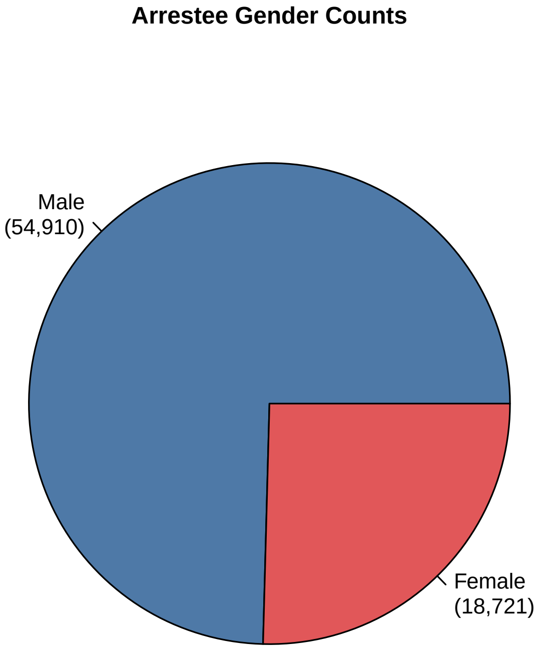


Figure 35: Texas DNA Database Demographic Distributions

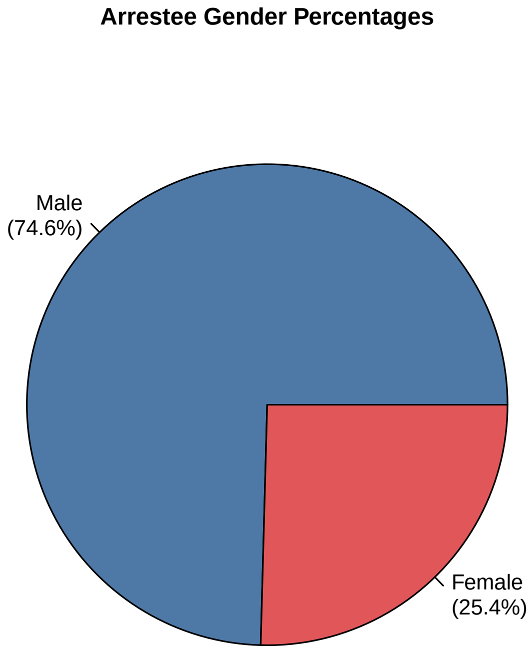


Figure 36: Texas DNA Database Demographic Distributions

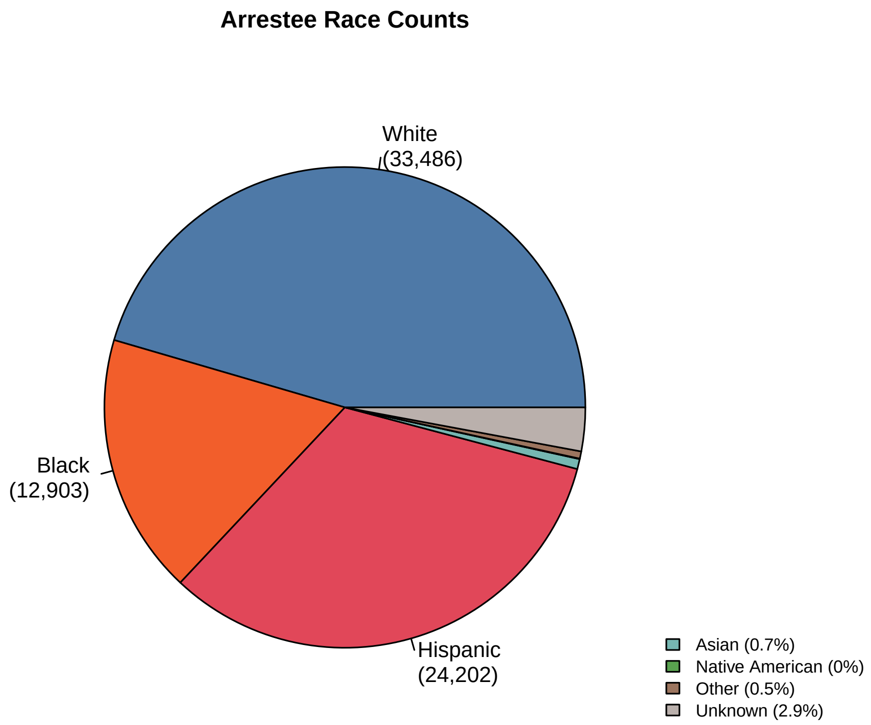


Figure 37: Texas DNA Database Demographic Distributions

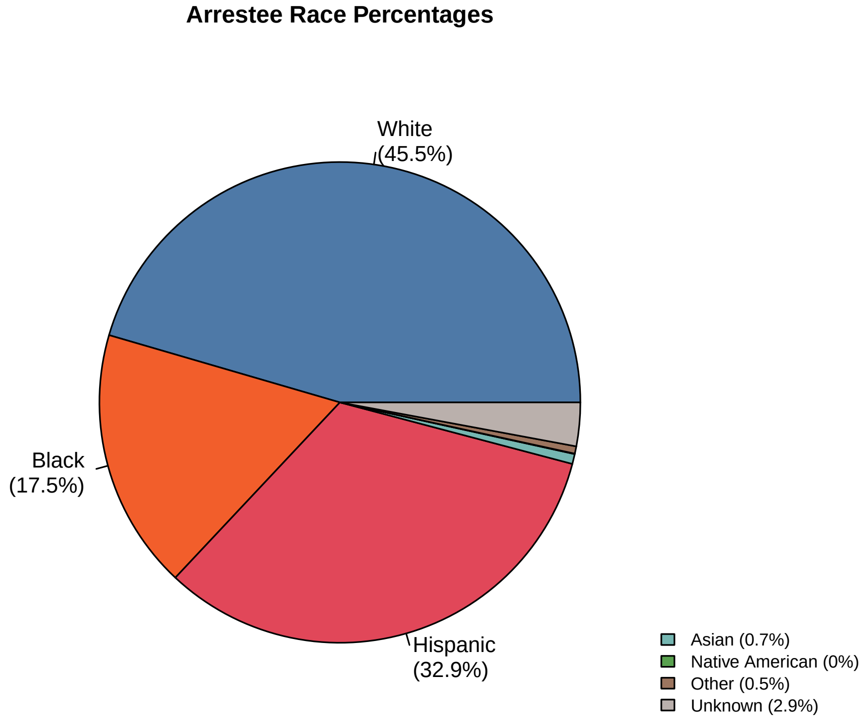


Figure 38: Texas DNA Database Demographic Distributions

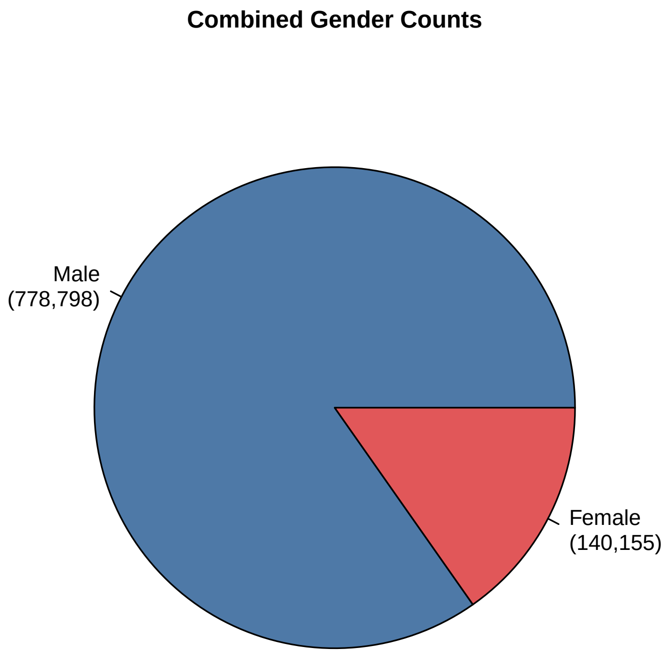


Figure 39: Texas DNA Database Demographic Distributions

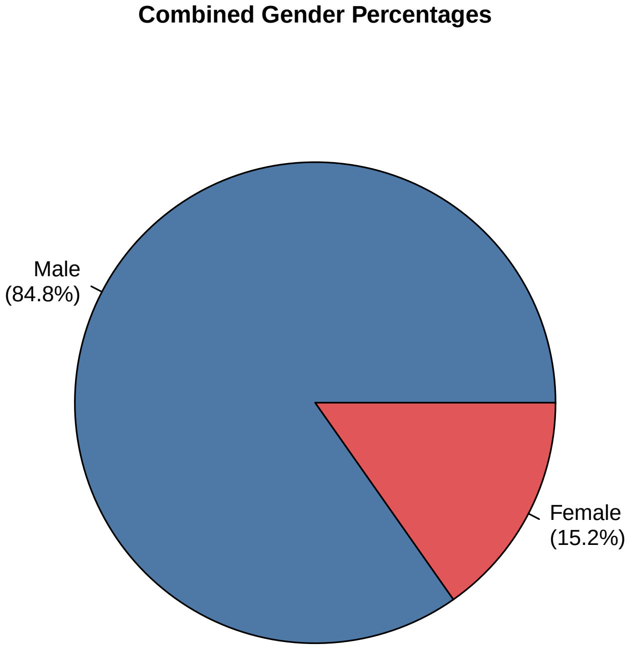


Figure 40: Texas DNA Database Demographic Distributions

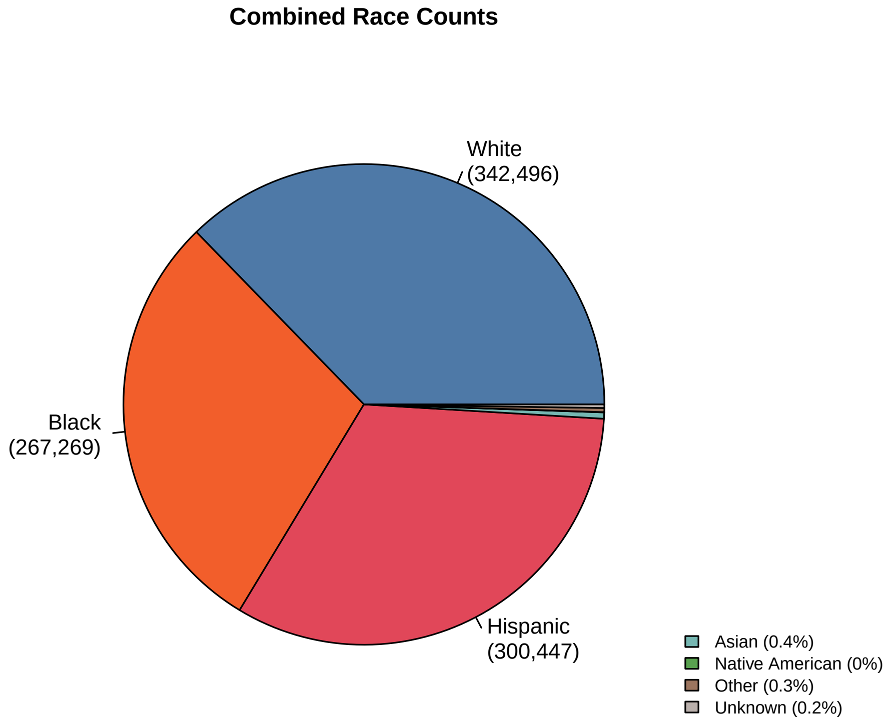


Figure 41: Texas DNA Database Demographic Distributions

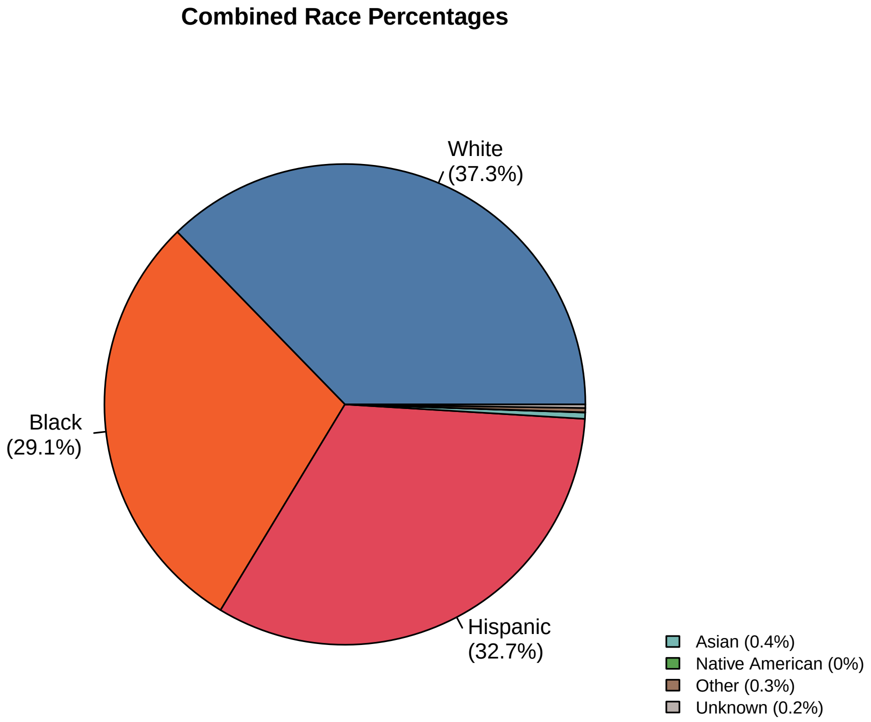


Figure 42: Texas DNA Database Demographic Distributions

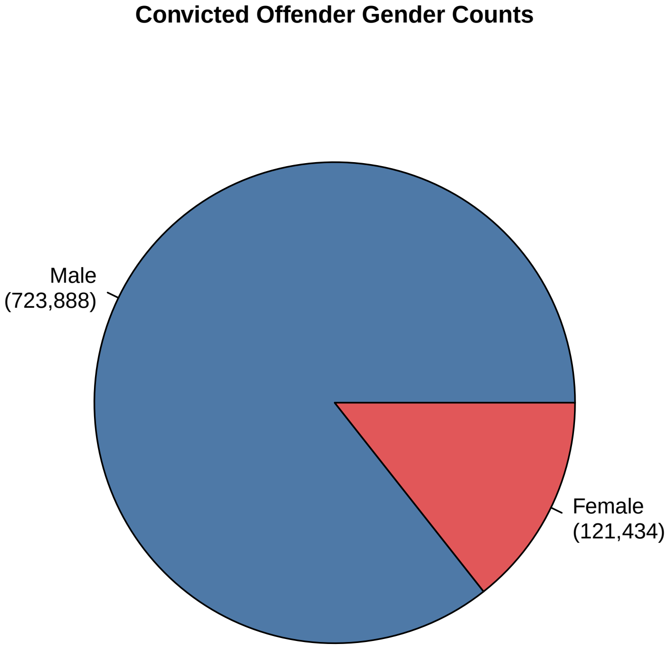


Figure 43: Texas DNA Database Demographic Distributions

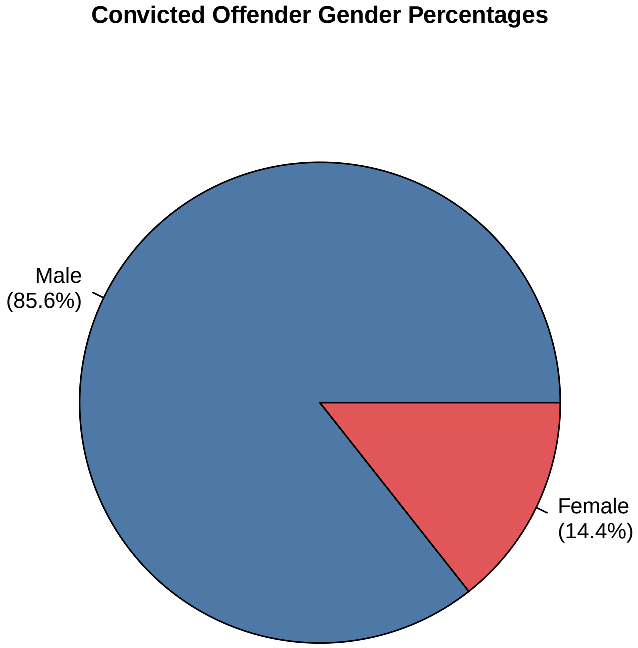


Figure 44: Texas DNA Database Demographic Distributions

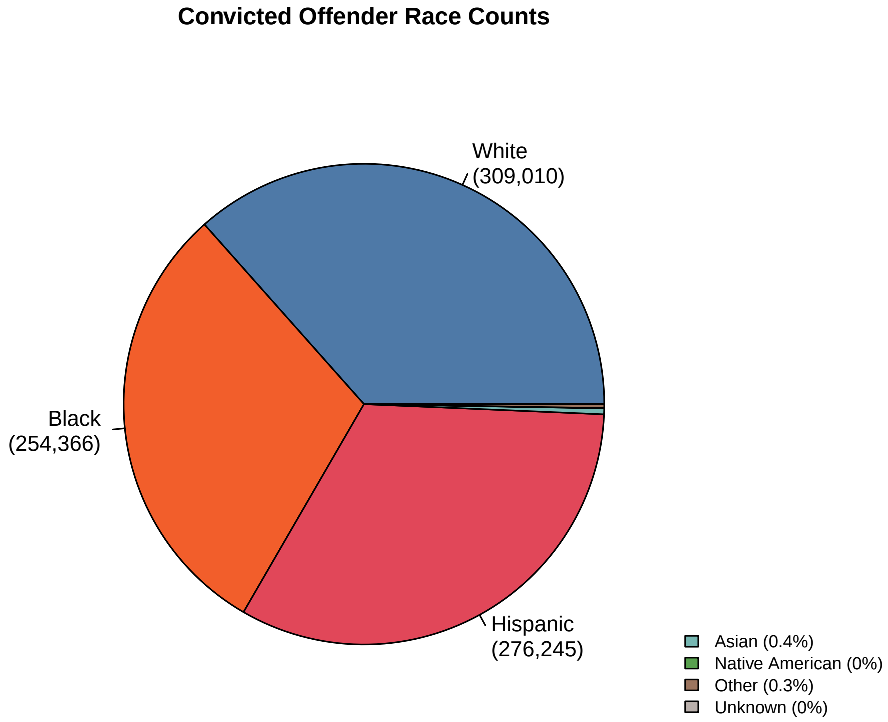


Figure 45: Texas DNA Database Demographic Distributions

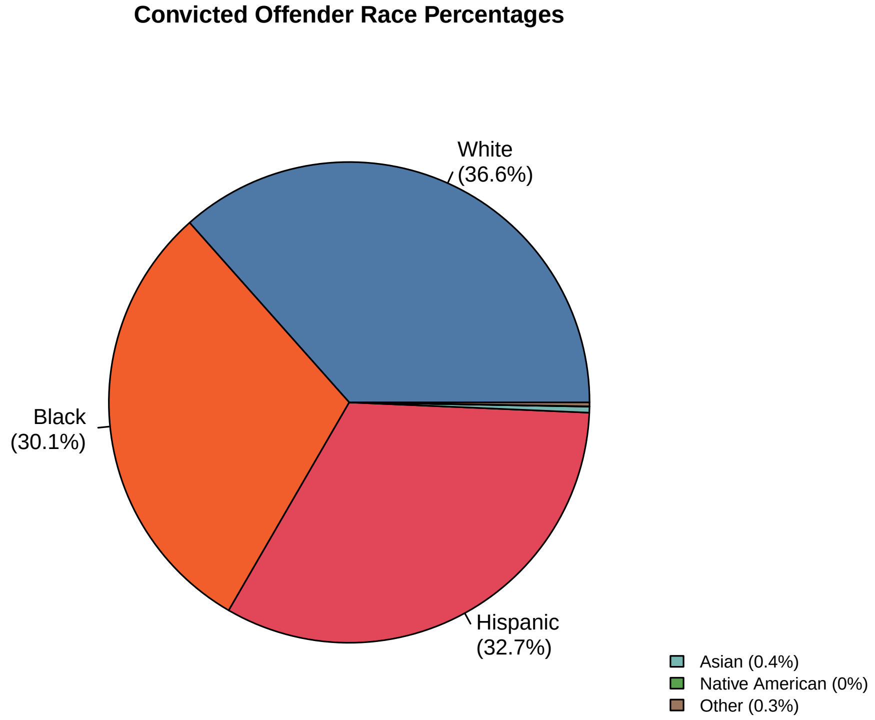


Figure 46: Texas DNA Database Demographic Distributions

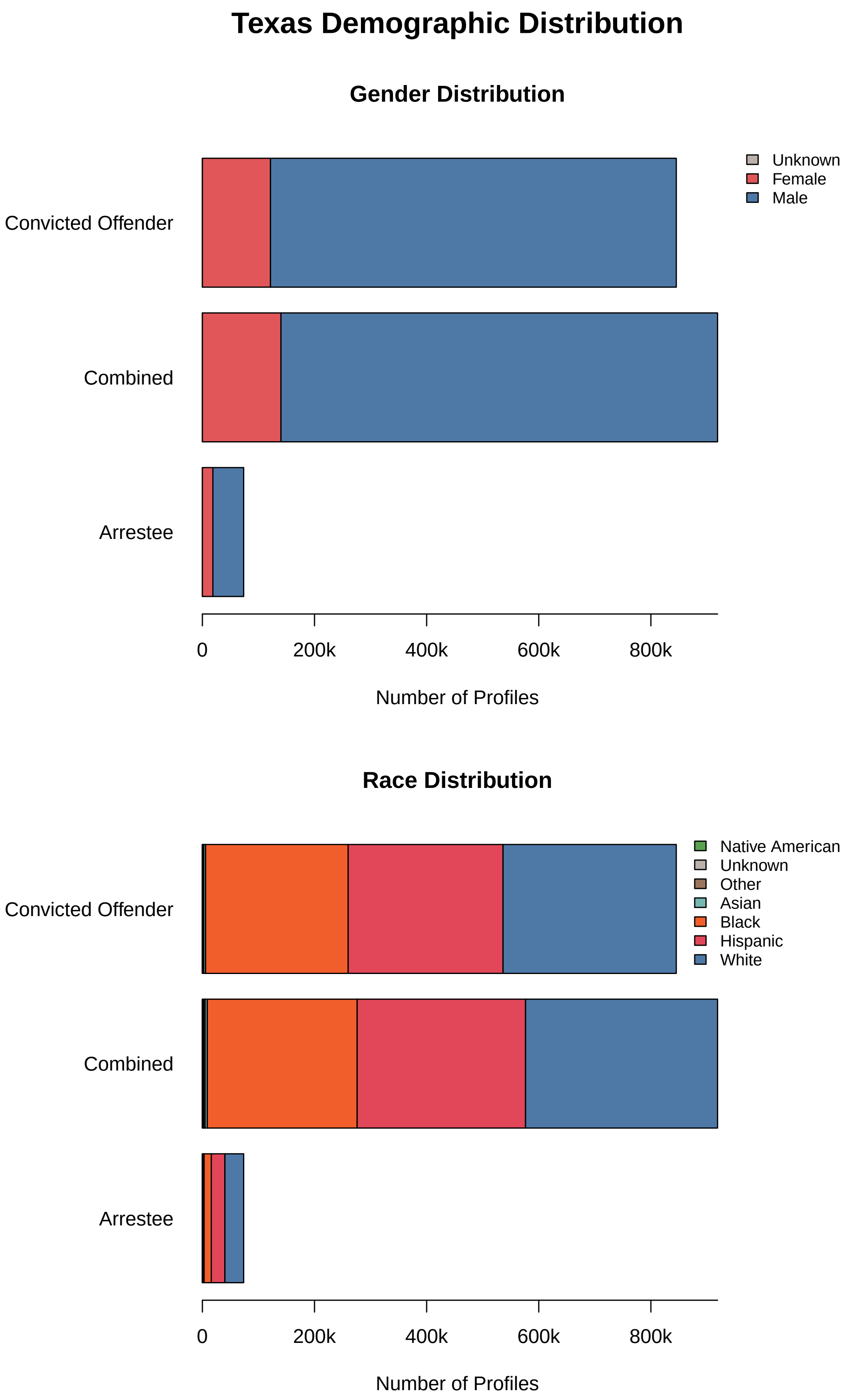


Figure 47: Texas Demographic Distributions by Offender Type

3.7.8 Summary Statistics

```r
cat("Texas DNA Database Summary:\n")
cat("=", strrep("=", 40), "\n")

# Total profiles by offender type
totals <- foia_combined %>%
  filter(state == "Texas",
         variable_category == "total",
         variable_detailed == "total_profiles",
         value_type == "count") %>%
  select(offender_type, value, value_source) %>%
  mutate(value_formatted = format(value, big.mark = ","))

print(totals)

# Data completeness by value source
cat("\nData completeness by source:\n")
completeness <- foia_combined %>%
  filter(state == "Texas") %>%
  group_by(value_source) %>%
  summarise(n_values = n(), .groups = "drop")

print(completeness)

# Final verification
cat("\nFinal verification:\n")
cat(paste("Counts consistent:", counts_consistent(foia_combined %>% filter(state == "Texas")),
cat(paste("Percentages consistent:", percentages_consistent(foia_combined %>% filter(state ==
```

```
Texas DNA Database Summary:
= ========================================
# A tibble: 3 x 4
  offender_type       value value_source value_formatted
  <chr>               <dbl> <chr>        <chr>
1 Convicted Offender 845322 reported     "845,322"
2 Arrestee            73631 reported     " 73,631"
3 Combined           918953 calculated   "918,953"

Data completeness by source:
# A tibble: 2 x 2
  value_source n_values
  <chr>           <int>
1 calculated         41
2 reported           16
```

```
Final verification:
Counts consistent: TRUE
Percentages consistent: TRUE
```

3.7.9 Summary of Texas Processing

Texas data processing complete. The dataset required several adjustments:

- **Male Category Addition**:
  - "Male" added to `variable_detailed`
- **Terminology standardization**:
  - "Offenders" → "Convicted Offender"
  - "Caucasian" → "White"
  - "African American" → "Black"
- **Calculated additions**:
  - Combined totals across all offender types
  - Percentage values for all demographic categories
- **Quality checks**: All counts and percentages pass consistency validation
- **Provenance tracking**: Clear distinction between reported and calculated values

The Texas data is now standardized and ready for cross-state analysis.

3.8 Combined Dataset

state	offender_type	variable_category	variable_detailed	value	value_type	value_source
California	Convicted Offender	total	total_profiles	2.019899e+06	count	reported
California	Arrestee	total	total_profiles	7.518220e+05	count	reported
California	Convicted Offender	gender	Female	3.098270e+05	count	reported
California	Convicted Offender	gender	Male	1.603222e+06	count	reported
California	Convicted Offender	gender	Unknown	1.068500e+05	count	reported
California	Arrestee	gender	Female	2.082250e+05	count	reported
California	Arrestee	gender	Male	5.242310e+05	count	reported
California	Arrestee	gender	Unknown	1.936600e+04	count	reported
California	Convicted Offender	race	Black	3.689520e+05	count	reported
California	Convicted Offender	race	White	5.885550e+05	count	reported
California	Convicted Offender	race	Hispanic	6.521210e+05	count	reported
California	Convicted Offender	race	Asian	1.638400e+04	count	reported
California	Arrestee	race	Black	1.047410e+05	count	reported
California	Arrestee	race	White	2.313130e+05	count	reported
California	Arrestee	race	Hispanic	3.084500e+05	count	reported

(continued)

state	offender_type	variable_category	variable_detailed	value	value_type	value_source
California	Arrestee	race	Asian	1.119100e+04	count	reported
California	Convicted Offender	race	Unknown	3.938870e+05	count	calculated
California	Arrestee	race	Unknown	9.612700e+04	count	calculated
California	Combined	gender	Female	5.180520e+05	count	calculated
California	Combined	gender	Male	2.127453e+06	count	calculated
California	Combined	gender	Unknown	1.262160e+05	count	calculated
California	Combined	race	Black	4.736930e+05	count	calculated
California	Combined	race	Asian	2.757500e+04	count	calculated
California	Combined	race	White	8.198680e+05	count	calculated
California	Combined	race	Hispanic	9.605710e+05	count	calculated
California	Combined	race	Unknown	4.900140e+05	count	calculated
California	Combined	total	total_profiles	2.771721e+06	count	calculated
California	Convicted Offender	gender	Female	1.534000e+01	percentage	calculated
California	Convicted Offender	gender	Male	7.937000e+01	percentage	calculated
California	Convicted Offender	gender	Unknown	5.290000e+00	percentage	calculated
California	Arrestee	gender	Female	2.770000e+01	percentage	calculated
California	Arrestee	gender	Male	6.973000e+01	percentage	calculated
California	Arrestee	gender	Unknown	2.580000e+00	percentage	calculated
California	Convicted Offender	race	Black	1.827000e+01	percentage	calculated
California	Convicted Offender	race	White	2.914000e+01	percentage	calculated
California	Convicted Offender	race	Hispanic	3.228000e+01	percentage	calculated
California	Convicted Offender	race	Asian	8.100000e-01	percentage	calculated
California	Arrestee	race	Black	1.393000e+01	percentage	calculated
California	Arrestee	race	White	3.077000e+01	percentage	calculated
California	Arrestee	race	Hispanic	4.103000e+01	percentage	calculated
California	Arrestee	race	Asian	1.490000e+00	percentage	calculated
California	Convicted Offender	race	Unknown	1.950000e+01	percentage	calculated
California	Arrestee	race	Unknown	1.279000e+01	percentage	calculated
California	Combined	gender	Female	1.869000e+01	percentage	calculated
California	Combined	gender	Male	7.676000e+01	percentage	calculated
California	Combined	gender	Unknown	4.550000e+00	percentage	calculated
California	Combined	race	Black	1.709000e+01	percentage	calculated
California	Combined	race	Asian	9.900000e-01	percentage	calculated
California	Combined	race	White	2.958000e+01	percentage	calculated
California	Combined	race	Hispanic	3.466000e+01	percentage	calculated
California	Combined	race	Unknown	1.768000e+01	percentage	calculated
Florida	Combined	total	total_profiles	1.175391e+06	count	reported
Florida	Combined	total	total_profiles	1.000000e+02	percentage	reported
Florida	Combined	gender	Female	2.608850e+05	count	reported
Florida	Combined	gender	Female	2.220000e+01	percentage	reported
Florida	Combined	gender	Male	9.011260e+05	count	reported
Florida	Combined	gender	Male	7.667000e+01	percentage	reported
Florida	Combined	gender	Unknown	1.338000e+04	count	reported
Florida	Combined	gender	Unknown	1.140000e+00	percentage	reported
Florida	Combined	race	Black	4.137330e+05	count	reported
Florida	Combined	race	Black	3.520000e+01	percentage	reported
Florida	Combined	race	Asian	2.659000e+03	count	reported
Florida	Combined	race	Asian	2.300000e-01	percentage	reported
Florida	Combined	race	White	7.214850e+05	count	reported
Florida	Combined	race	White	6.138000e+01	percentage	reported
Florida	Combined	race	Hispanic	2.845200e+04	count	reported

(continued)

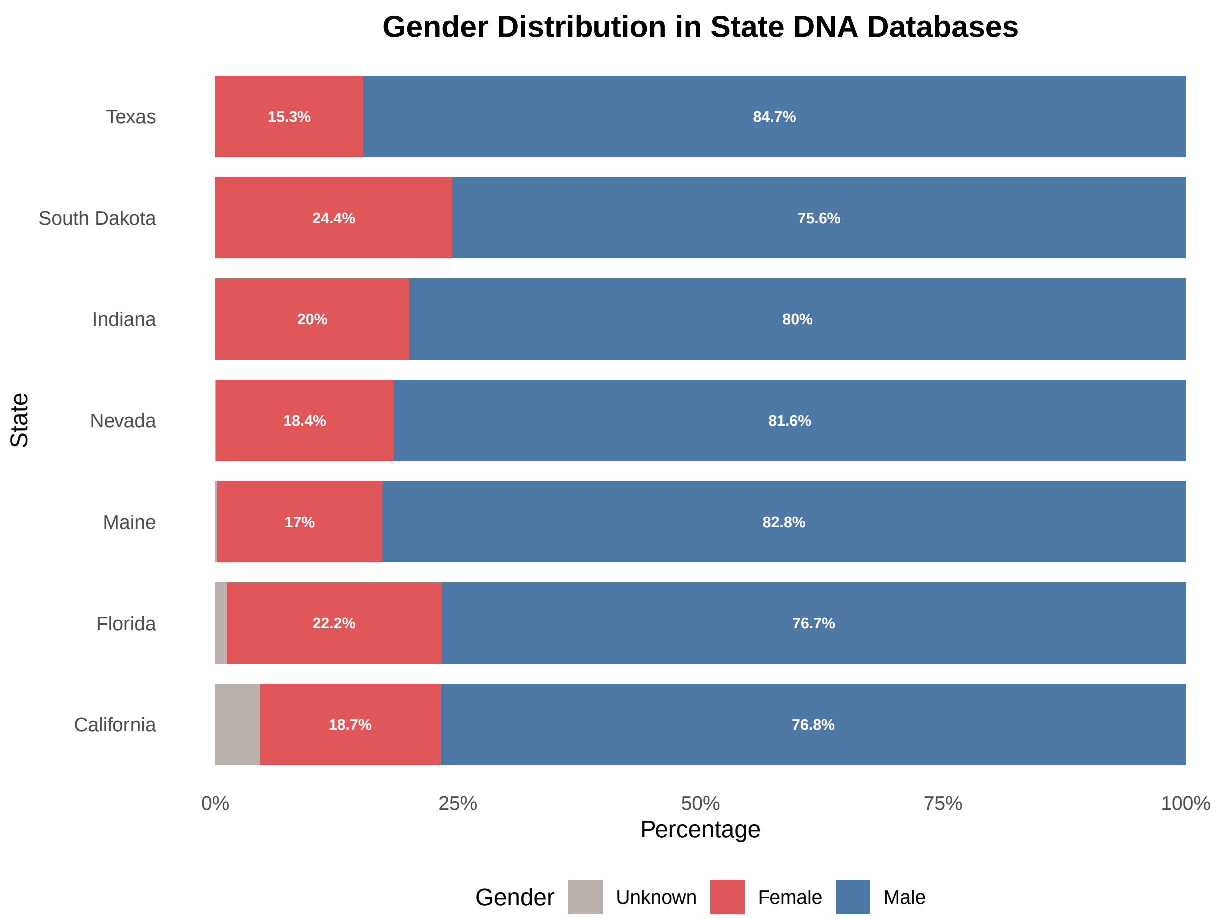
state	offender_type	variable_category	variable_detailed	value	value_type	value_source
Florida	Combined	race	Hispanic	2.420000e+00	percentage	reported
Florida	Combined	race	Native American	6.670000e+02	count	reported
Florida	Combined	race	Native American	6.000000e-02	percentage	reported
Florida	Combined	race	Other	1.176000e+03	count	reported
Florida	Combined	race	Other	1.000000e-01	percentage	reported
Florida	Combined	race	Unknown	7.219000e+03	count	reported
Florida	Combined	race	Unknown	6.100000e-01	percentage	reported
Indiana	Convicted Offender	total	total_profiles	2.796540e+05	count	reported
Indiana	Arrestee	total	total_profiles	2.108700e+04	count	reported
Indiana	Combined	gender	Female	2.000000e+01	percentage	reported
Indiana	Combined	gender	Male	8.000000e+01	percentage	reported
Indiana	Combined	race	White	6.965174e+01	percentage	calculated
Indiana	Combined	race	Black	2.587065e+01	percentage	calculated
Indiana	Combined	race	Hispanic	3.980100e+00	percentage	calculated
Indiana	Combined	race	Other	4.975124e-01	percentage	calculated
Indiana	Combined	total	total_profiles	3.007410e+05	count	calculated
Indiana	Combined	gender	Female	6.014800e+04	count	calculated
Indiana	Combined	gender	Male	2.405930e+05	count	calculated
Indiana	Combined	race	White	2.094710e+05	count	calculated
Indiana	Combined	race	Black	7.780400e+04	count	calculated
Indiana	Combined	race	Hispanic	1.197000e+04	count	calculated
Indiana	Combined	race	Other	1.496000e+03	count	calculated
Maine	Combined	total	total_profiles	3.371100e+04	count	reported
Maine	Combined	gender	Male	8.278278e+01	percentage	calculated
Maine	Combined	gender	Female	1.701702e+01	percentage	calculated
Maine	Combined	gender	Unknown	2.002002e-01	percentage	calculated
Maine	Combined	race	White	3.129800e+04	count	reported
Maine	Combined	race	White	9.280000e+01	percentage	reported
Maine	Combined	race	Black	1.299000e+03	count	reported
Maine	Combined	race	Black	3.900000e+00	percentage	reported
Maine	Combined	race	Unknown	4.700000e+02	count	reported
Maine	Combined	race	Unknown	1.400000e+00	percentage	reported
Maine	Combined	race	Native American	3.450000e+02	count	reported
Maine	Combined	race	Native American	1.000000e+00	percentage	reported
Maine	Combined	race	Hispanic	1.710000e+02	count	reported
Maine	Combined	race	Hispanic	5.000000e-01	percentage	reported
Maine	Combined	race	Asian	1.280000e+02	count	reported
Maine	Combined	race	Asian	4.000000e-01	percentage	reported
Maine	Combined	gender	Male	2.790700e+04	count	calculated
Maine	Combined	gender	Female	5.737000e+03	count	calculated
Maine	Combined	gender	Unknown	6.700000e+01	count	calculated
Nevada	Combined	total	total_profiles	3.440970e+05	count	reported
Nevada	Arrestee	total	total_profiles	1.850740e+05	count	reported
Nevada	Arrestee	total	total_profiles	5.378500e+01	percentage	reported
Nevada	Convicted Offender	total	total_profiles	1.590230e+05	count	reported
Nevada	Convicted Offender	total	total_profiles	4.621500e+01	percentage	reported
Nevada	Combined	gender	Female	6.328700e+04	count	reported
Nevada	Combined	gender	Female	1.839200e+01	percentage	reported
Nevada	Combined	gender	Male	2.807380e+05	count	reported
Nevada	Combined	gender	Male	8.158700e+01	percentage	reported
Nevada	Combined	gender	Unknown	7.200000e+01	count	reported

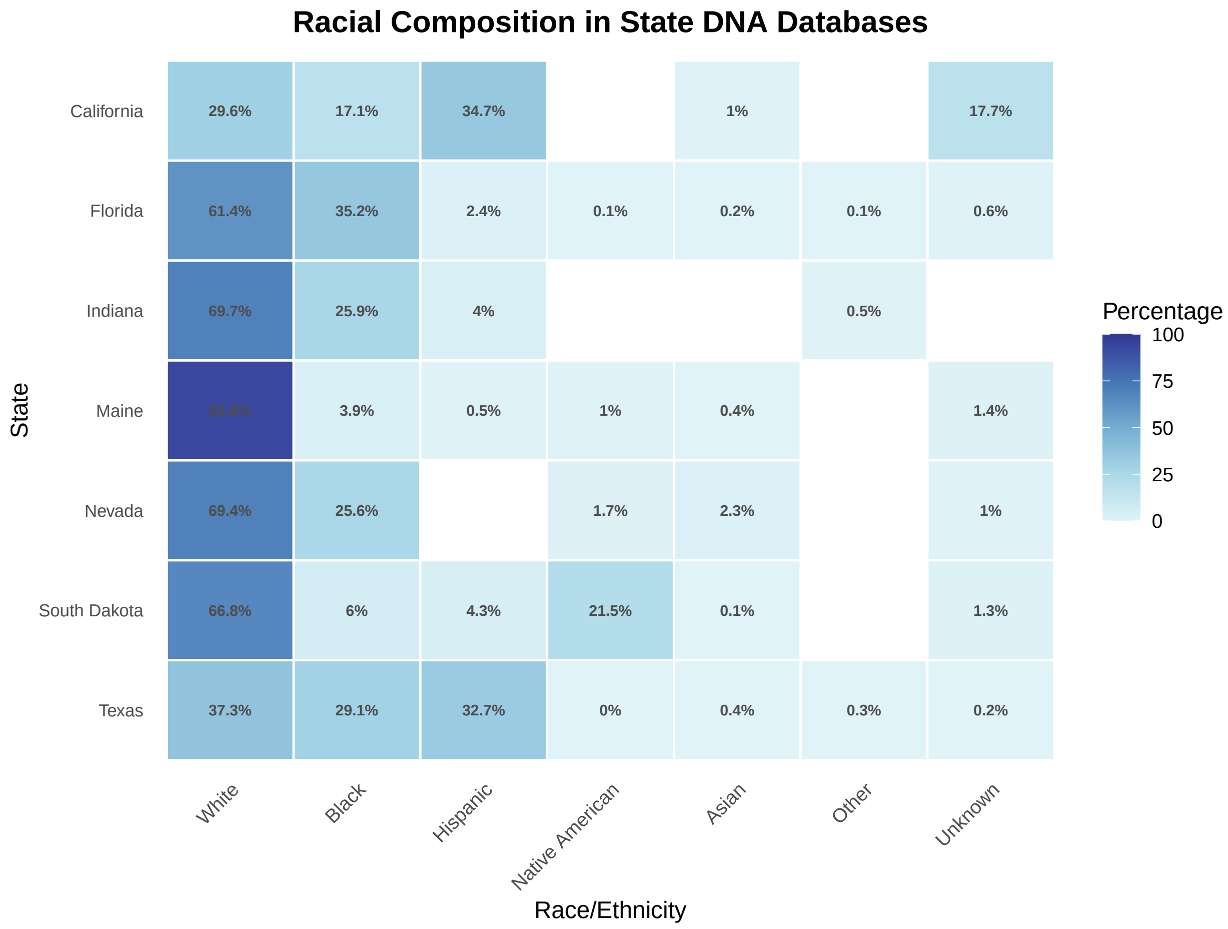
(continued)

state	offender_type	variable_category	variable_detailed	value	value_type	value_source
Nevada	Combined	gender	Unknown	2.090000e-02	percentage	reported
Nevada	Combined	race	White	2.387230e+05	count	reported
Nevada	Combined	race	White	6.937700e+01	percentage	reported
Nevada	Combined	race	Unknown	3.491000e+03	count	reported
Nevada	Combined	race	Unknown	1.015000e+00	percentage	reported
Nevada	Combined	race	Native American	5.710000e+03	count	reported
Nevada	Combined	race	Native American	1.659000e+00	percentage	reported
Nevada	Combined	race	Black	8.817400e+04	count	reported
Nevada	Combined	race	Black	2.562500e+01	percentage	reported
Nevada	Combined	race	Asian	7.999000e+03	count	reported
Nevada	Combined	race	Asian	2.346000e+00	percentage	reported
South Dakota	Combined	total	total_profiles	6.775300e+04	count	reported
South Dakota	Combined	gender	Male	5.119700e+04	count	reported
South Dakota	Combined	gender	Male	7.556000e+01	percentage	reported
South Dakota	Combined	gender	Female	1.655600e+04	count	reported
South Dakota	Combined	gender	Female	2.444000e+01	percentage	reported
South Dakota	Combined	race	Asian	8.000000e-02	percentage	reported
South Dakota	Combined	race	Black	5.960000e+00	percentage	reported
South Dakota	Combined	race	Hispanic	4.350000e+00	percentage	reported
South Dakota	Combined	race	Native American	2.154000e+01	percentage	reported
South Dakota	Combined	race	Unknown	1.320000e+00	percentage	reported
South Dakota	Combined	race	White	6.675000e+01	percentage	reported
South Dakota	Combined	race	Asian	5.400000e+01	count	calculated
South Dakota	Combined	race	Black	4.038000e+03	count	calculated
South Dakota	Combined	race	Hispanic	2.947000e+03	count	calculated
South Dakota	Combined	race	Native American	1.459400e+04	count	calculated
South Dakota	Combined	race	Unknown	8.940000e+02	count	calculated
South Dakota	Combined	race	White	4.522600e+04	count	calculated
Texas	Convicted Offender	total	total_profiles	8.453220e+05	count	reported
Texas	Arrestee	total	total_profiles	7.363100e+04	count	reported
Texas	Convicted Offender	gender	Female	1.214340e+05	count	reported
Texas	Arrestee	gender	Female	1.872100e+04	count	reported
Texas	Convicted Offender	race	Asian	3.361000e+03	count	reported
Texas	Convicted Offender	race	Black	2.543660e+05	count	reported
Texas	Convicted Offender	race	White	3.090100e+05	count	reported
Texas	Convicted Offender	race	Hispanic	2.762450e+05	count	reported
Texas	Convicted Offender	race	Native American	1.380000e+02	count	reported
Texas	Convicted Offender	race	Other	2.173000e+03	count	reported
Texas	Arrestee	race	Asian	4.970000e+02	count	reported
Texas	Arrestee	race	Black	1.290300e+04	count	reported
Texas	Arrestee	race	White	3.348600e+04	count	reported
Texas	Arrestee	race	Hispanic	2.420200e+04	count	reported
Texas	Arrestee	race	Native American	2.400000e+01	count	reported
Texas	Arrestee	race	Other	3.580000e+02	count	reported
Texas	Convicted Offender	gender	Male	7.238880e+05	count	calculated
Texas	Arrestee	gender	Male	5.491000e+04	count	calculated
Texas	Convicted Offender	race	Unknown	2.900000e+01	count	calculated
Texas	Arrestee	race	Unknown	2.161000e+03	count	calculated
Texas	Combined	gender	Female	1.401550e+05	count	calculated
Texas	Combined	gender	Male	7.787980e+05	count	calculated
Texas	Combined	race	Asian	3.858000e+03	count	calculated

(continued)

state	offender_type	variable_category	variable_detailed	value	value_type	value_source
Texas	Combined	race	Black	2.672690e+05	count	calculated
Texas	Combined	race	Hispanic	3.004470e+05	count	calculated
Texas	Combined	race	Native American	1.620000e+02	count	calculated
Texas	Combined	race	Other	2.531000e+03	count	calculated
Texas	Combined	race	Unknown	2.190000e+03	count	calculated
Texas	Combined	race	White	3.424960e+05	count	calculated
Texas	Combined	total	total_profiles	9.189530e+05	count	calculated
Texas	Convicted Offender	gender	Female	1.437000e+01	percentage	calculated
Texas	Arrestee	gender	Female	2.543000e+01	percentage	calculated
Texas	Convicted Offender	race	Asian	4.000000e-01	percentage	calculated
Texas	Convicted Offender	race	Black	3.009000e+01	percentage	calculated
Texas	Convicted Offender	race	White	3.656000e+01	percentage	calculated
Texas	Convicted Offender	race	Hispanic	3.268000e+01	percentage	calculated
Texas	Convicted Offender	race	Native American	2.000000e-02	percentage	calculated
Texas	Convicted Offender	race	Other	2.600000e-01	percentage	calculated
Texas	Arrestee	race	Asian	6.700000e-01	percentage	calculated
Texas	Arrestee	race	Black	1.752000e+01	percentage	calculated
Texas	Arrestee	race	White	4.548000e+01	percentage	calculated
Texas	Arrestee	race	Hispanic	3.287000e+01	percentage	calculated
Texas	Arrestee	race	Native American	3.000000e-02	percentage	calculated
Texas	Arrestee	race	Other	4.900000e-01	percentage	calculated
Texas	Convicted Offender	gender	Male	8.563000e+01	percentage	calculated
Texas	Arrestee	gender	Male	7.457000e+01	percentage	calculated
Texas	Convicted Offender	race	Unknown	0.000000e+00	percentage	calculated
Texas	Arrestee	race	Unknown	2.930000e+00	percentage	calculated
Texas	Combined	gender	Female	1.525000e+01	percentage	calculated
Texas	Combined	gender	Male	8.475000e+01	percentage	calculated
Texas	Combined	race	Asian	4.200000e-01	percentage	calculated
Texas	Combined	race	Black	2.908000e+01	percentage	calculated
Texas	Combined	race	Hispanic	3.269000e+01	percentage	calculated
Texas	Combined	race	Native American	2.000000e-02	percentage	calculated
Texas	Combined	race	Other	2.800000e-01	percentage	calculated
Texas	Combined	race	Unknown	2.400000e-01	percentage	calculated
Texas	Combined	race	White	3.727000e+01	percentage	calculated


Gender Distribution in State DNA Databases
Texas
South Dakota
Indiana
Nevada
Maine
Florida
California
State
15.3%
84.7%
24.4%
75.6%
20%
80%
18.4%
81.6%
17%
82.8%
22.2%
76.7%
18.7%
76.8%
0%
25%
50%
75%
100%
Percentage
Gender
Unknown
Female
Male



3.9 Metadata table

state	race_data_provided	gender_data_provided	total_profiles_provided	convicted_offender_reported	arreste
California	counts	counts	counts	TRUE	TRUE
Florida	both	both	both	FALSE	FALSE
Indiana	percentages	percentages	counts	TRUE	TRUE
Maine	both	both	counts	FALSE	FALSE
Nevada	both	both	both	TRUE	TRUE
South Dakota	both	both	counts	FALSE	FALSE
Texas	counts	counts	counts	FALSE	TRUE

4 Saving Processed Data

```
# Define output paths
output_dir <- here("data", "foia", "final")
dir.create(output_dir, recursive = TRUE, showWarnings = FALSE)

# Save the combined dataset
```

```r
foia_output_path <- here(output_dir, "foia_data_clean.csv")
write_csv(foia_combined, foia_output_path)
cat(paste("  Saved combined FOIA data to:", foia_output_path, "\n"))

# Save the metadata
metadata_dir <- here("data", "foia", "intermediate")
metadata_output_path <- here(metadata_dir, "foia_state_metadata.csv")
write_csv(foia_state_metadata, metadata_output_path)
cat(paste("  Saved state metadata to:", metadata_output_path, "\n"))

# Create final frozen version (v1.0)
frozen_dir <- here("data", "v1.0")
dir.create(frozen_dir, recursive = TRUE, showWarnings = FALSE)

frozen_path <- here(frozen_dir, "FOIA_demographics.csv")
write_csv(foia_combined, frozen_path)
cat(paste("  Created frozen version 1.0 at:", frozen_path, "\n"))

cat("\n  All processing complete! The data is now ready for analysis.\n")
```

```
 Saved combined FOIA data to: C:/Users/Donadio/Documents/PODFRIDGE_Databases/data/foia/final/f
 Saved state metadata to: C:/Users/Donadio/Documents/PODFRIDGE_Databases/data/foia/intermediat
 Created frozen version 1.0 at: C:/Users/Donadio/Documents/PODFRIDGE_Databases/data/v1.0/FOIA_

 All processing complete! The data is now ready for analysis.
```

5 Conclusions

1. **Data Acquisition and Harmonization**: We ingested seven unique state datasets (`california_foia_data.csv` through `texas_foia_data.csv`), each with distinct reporting formats, terminology, and levels of completeness. Through a systematic processing workflow, we harmonized these into a single, tidy long-format dataset (`foia_combined`), ensuring consistency across all variables.

2. **Standardization of Terminology**: A significant challenge was the non-standard terminology used across states. We implemented a rigorous process to map all state-specific terms to a common data model:

   - **Offender Types**: Standardized to `"Convicted Offender"`, `"Arrestee"`, and `"Combined"`.
   - **Race Categories**: Mapped terms like `"Caucasian"`, `"African American"`, and `"American Indian"` to standardized categories (`"White"`, `"Black"`, `"Native American"`).
   - **Total Profiles**: Consolidated terms like `"total_flags"` to `"total_profiles"`.

3. **Imputation and Calculation of Missing Data**: To ensure comparability, we calculated values that were not directly provided by the states:

   - **Derived Percentages**: For states providing only counts (CA, TX), we calculated percentage compositions.
   - **Derived Counts**: For states providing only percentages (IN), we calculated absolute numbers using reported totals.
   - **Calculated Totals**: We created `"Combined"` offender type totals for states that only reported separate `"Convicted Offender"` and `"Arrestee"` figures.
   - **Inferred Categories**: We added `"Unknown"` race and `"Male"` gender categories where they were logically missing but necessary to reconcile reported totals (CA, TX).

4. **Quality Assurance and Transparency**: A core principle of this project was maintaining transparency and data provenance. This allows future researchers to understand exactly what was provided by the state versus what was derived during processing.

   - **Validation checks**: `counts_consistent()`, `percentages_consistent()`
   - **Value tagging**: `"reported"` or `"calculated"`.
   - **Metadata table**: `foia_state_metadata` provides a clear audit trail of each state's original characteristics and the processing steps applied.