PSID Panel Data Preparation (2015–2023)

Elena Ajayi

2025-07-04

PSID Panel Data Processing Pipeline

Purpose

Clean and reshape PSID panel data across 5 waves (2015, 2017, 2019, 2021, 2023) to create a longitudinal dataset suitable for panel analysis.

1. Load Required Packages

```
library(dplyr)      # For data manipulation
library(tidyr)      # For reshaping data (long <-> wide)
library(readxl)     # For reading Excel files
library(stringr)    # For string operations on ER codes
library(purrr)      # For map functions
library(scales)     # For formatting numbers

cat("All required packages loaded successfully\n")
```

```
## All required packages loaded successfully
```

2. Define Variable Selection and Labels

```
# PSID Variable Mapping for 2015, 2017, 2019, 2021, and 2023
# Using actual PSID variable codes from the labels.txt file

# Year-specific variable mappings using actual PSID codes
variable_mappings <- list(
  "2015" = c(
    "ER60002" = "ID",                     # Family ID
    "ER64810" = "Race",                   # Race of head (L40)
    "ER64809" = "Hispanic",               # Hispanic ethnicity (L39)
    "ER61721" = "OtherRealEstateValue",   # Other real estate (W1)
    "ER61735" = "FarmBusinessIndicator",  # Farm/business indicator (W10)
    "ER61736" = "FarmBusinessValue",      # Farm/business value (W11A)
    "ER61740" = "FarmBusinessDebt",       # Farm/business debt (W11B)
    "ER61744" = "StocksValue",            # Stocks indicator (W15)
    "ER61745" = "StocksDebt",             # Stocks value (W16)
    "ER61764" = "CheckingSavingsValue",   # Checking/savings value (W21)
    "ER61766" = "CheckingSavingsDebt",    # Checking/savings debt (W22)
    "ER61771" = "VehicleValue",           # Vehicle value (W27)
    "ER61772" = "VehicleDebt",            # Vehicle debt (W28)
```

```r
    "ER61792" = "OtherAssetsValue",         # Other assets value (W33)
    "ER61793" = "OtherAssetsDebt",          # Other assets debt (W34)
    "ER61797" = "CreditCardDebt",           # Credit card debt (W38A)
    "ER61808" = "StudentLoanDebt",          # Student loan debt (W39B1)
    "ER61813" = "CDsValue",                 # CDs value (W39B2)
    "ER61818" = "SavingsBondsValue",        # Savings bonds value (W39B3)
    "ER61829" = "OtherDebtsValue",          # Other debt (W39B7)
    "ER65406" = "WealthExclEquity",         # Wealth w/o equity (WEALTH1)
    "ER65408" = "WealthInclEquity",         # Wealth w/ equity (WEALTH2)
    "ER60017" = "Age",                      # Age of head (BC21)
    "ER60018" = "Gender",                   # Sex of head (F1F)
    "ER60021" = "NumChildren",              # Number of children
    "ER60024" = "MaritalStatus",            # Marital status
    "ER65452" = "MetroArea",                # Metropolitan area (METRO)
    "ER65453" = "BealeCode"                 # Beale rural-urban code (BEALE)
  ),

  "2017" = c(
    "ER66002" = "ID",                       # Family ID
    "ER70882" = "Race",                     # Race of reference person (L40)
    "ER70881" = "Hispanic",                 # Hispanic ethnicity (L39)
    "ER67774" = "OtherRealEstateValue",     # Other real estate (W1)
    "ER67788" = "FarmBusinessIndicator",    # Farm/business indicator (W10)
    "ER67789" = "FarmBusinessValue",        # Farm/business value (W11A)
    "ER67793" = "FarmBusinessDebt",         # Farm/business debt (W11B)
    "ER67797" = "StocksValue",              # Stocks indicator (W15)
    "ER67798" = "StocksDebt",               # Stocks value (W16)
    "ER67817" = "CheckingSavingsValue",     # Checking/savings value (W21)
    "ER67819" = "CheckingSavingsDebt",      # Checking/savings debt (W22)
    "ER67825" = "VehicleValue",             # Vehicle value (W27A)
    "ER67826" = "VehicleDebt",              # Vehicle debt (W28)
    "ER67846" = "OtherAssetsValue",         # Other assets value (W33)
    "ER67847" = "OtherAssetsDebt",          # Other assets debt (W34)
    "ER67851" = "CreditCardDebt",           # Credit card debt (W38A)
    "ER67862" = "StudentLoanDebt",          # Student loan debt (W39B1)
    "ER67867" = "CDsValue",                 # CDs value (W39B2)
    "ER67872" = "SavingsBondsValue",        # Savings bonds value (W39B3)
    "ER67883" = "OtherDebtsValue",          # Other debt (W39B7)
    "ER71483" = "WealthExclEquity",         # Wealth w/o equity (WEALTH1)
    "ER71485" = "WealthInclEquity",         # Wealth w/ equity (WEALTH2)
    "ER66017" = "Age",                      # Age of reference person (BC21)
    "ER66018" = "Gender",                   # Sex of reference person (F1F)
    "ER66021" = "NumChildren",              # Number of children
    "ER66024" = "MaritalStatus",            # Marital status
    "ER71531" = "MetroArea",                # Metropolitan area (METRO)
    "ER71532" = "BealeCode"                 # Beale rural-urban code (BEALE)
  ),

  "2019" = c(
    "ER72002" = "ID",                       # Family ID
    "ER76897" = "Race",                     # Race of reference person (L40)
    "ER76896" = "Hispanic",                 # Hispanic ethnicity (L39)
    "ER73797" = "OtherRealEstateValue",     # Other real estate (W1)
```

```r
    "ER73811" = "FarmBusinessIndicator",   # Farm/business indicator (W10)
    "ER73812" = "FarmBusinessValue",       # Farm/business value (W11A)
    "ER73816" = "FarmBusinessDebt",        # Farm/business debt (W11B)
    "ER73820" = "StocksValue",             # Stocks indicator (W15)
    "ER73821" = "StocksDebt",              # Stocks value (W16)
    "ER73840" = "CheckingSavingsValue",    # Checking/savings value (W21)
    "ER73842" = "CheckingSavingsDebt",     # Checking/savings debt (W22)
    "ER73847" = "VehicleValue",            # Vehicle value (W27A)
    "ER73848" = "VehicleDebt",             # Vehicle debt (W28A)
    "ER73853" = "CDsValue",                # CDs value (W27)
    "ER73854" = "CDsDebt",                 # CDs debt (W28)
    "ER73874" = "OtherAssetsValue",        # Other assets value (W33)
    "ER73875" = "OtherAssetsDebt",         # Other assets debt (W34)
    "ER73879" = "CreditCardDebt",          # Credit card debt (W38A)
    "ER73890" = "StudentLoanDebt",         # Student loan debt (W39B1)
    "ER73895" = "SavingsBondsValue",       # Savings bonds value (W39B2)
    "ER73900" = "OtherDebtsValue",         # Other debt (W39B3)
    "ER77509" = "WealthExclEquity",        # Wealth w/o equity (WEALTH1)
    "ER77511" = "WealthInclEquity",        # Wealth w/ equity (WEALTH2)
    "ER72017" = "Age",                     # Age of reference person (BC21)
    "ER72018" = "Gender",                  # Sex of reference person (F1F)
    "ER72021" = "NumChildren",             # Number of children
    "ER72024" = "MaritalStatus",           # Marital status
    "ER77592" = "MetroArea",               # Metropolitan area (METRO)
    "ER77593" = "BealeCode"                # Beale rural-urban code (BEALE)
  ),

  "2021" = c(
    "ER78002" = "ID",                      # Family ID
    "ER81144" = "Race",                    # Race of reference person (L40)
    "ER81143" = "Hispanic",                # Hispanic ethnicity (L39)
    "ER79919" = "OtherRealEstateValue",    # Other real estate (W1)
    "ER79933" = "FarmBusinessIndicator",   # Farm/business indicator (W10)
    "ER79934" = "FarmBusinessValue",       # Farm/business value (W11A)
    "ER79938" = "FarmBusinessDebt",        # Farm/business debt (W11B)
    "ER79942" = "StocksValue",             # Stocks indicator (W15)
    "ER79943" = "StocksDebt",              # Stocks value (W16)
    "ER79962" = "CheckingSavingsValue",    # Checking/savings value (W21)
    "ER79964" = "CheckingSavingsDebt",     # Checking/savings debt (W22)
    "ER79969" = "VehicleValue",            # Vehicle value (W27A)
    "ER79970" = "VehicleDebt",             # Vehicle debt (W28A)
    "ER79975" = "CDsValue",                # CDs value (W27)
    "ER79976" = "CDsDebt",                 # CDs debt (W28)
    "ER79996" = "OtherAssetsValue",        # Other assets value (W33)
    "ER79997" = "OtherAssetsDebt",         # Other assets debt (W34)
    "ER80001" = "CreditCardDebt",          # Credit card debt (W38A)
    "ER80012" = "StudentLoanDebt",         # Student loan debt (W39B1)
    "ER80017" = "SavingsBondsValue",       # Savings bonds value (W39B2)
    "ER80022" = "OtherDebtsValue",         # Other debt (W39B3)
    "ER81836" = "WealthExclEquity",        # Wealth w/o equity (WEALTH1)
    "ER81838" = "WealthInclEquity",        # Wealth w/ equity (WEALTH2)
    "ER78017" = "Age",                     # Age of reference person (BC21)
    "ER78018" = "Gender",                  # Sex of reference person (F1F)
```

```r
    "ER78021" = "NumChildren",              # Number of children
    "ER78025" = "MaritalStatus",            # Marital status
    "ER81919" = "MetroArea",                # Metropolitan area (METRO)
    "ER81920" = "BealeCode"                 # Beale rural-urban code (BEALE)
  ),

  "2023" = c(
    "ER82002" = "ID",                       # Family ID
    "ER85121" = "Race",                     # Race of reference person (L40)
    "ER85120" = "Hispanic",                 # Hispanic ethnicity (L39)
    "ER83888" = "OtherRealEstateValue",     # Other real estate (W1)
    "ER83902" = "FarmBusinessIndicator",    # Farm/business indicator (W10)
    "ER83903" = "FarmBusinessValue",        # Farm/business value (W11A)
    "ER83907" = "FarmBusinessDebt",         # Farm/business debt (W11B)
    "ER83911" = "StocksValue",              # Stocks indicator (W15)
    "ER83912" = "StocksDebt",               # Stocks value (W16)
    "ER83931" = "CheckingSavingsValue",     # Checking/savings value (W21)
    "ER83934" = "CheckingSavingsDebt",      # Checking/savings debt (W22)
    "ER83939" = "VehicleValue",             # Vehicle value (W27A)
    "ER83940" = "VehicleDebt",              # Vehicle debt (W28A)
    "ER83945" = "CDsValue",                 # CDs value (W27)
    "ER83946" = "CDsDebt",                  # CDs debt (W28)
    "ER83966" = "OtherAssetsValue",         # Other assets value (W33)
    "ER83967" = "OtherAssetsDebt",          # Other assets debt (W34)
    "ER83971" = "CreditCardDebt",           # Credit card debt (W38A)
    "ER83982" = "StudentLoanDebt",          # Student loan debt (W39B1)
    "ER83987" = "SavingsBondsValue",        # Savings bonds value (W39B2)
    "ER83992" = "OtherDebtsValue",          # Other debt (W39B3)
    "ER85690" = "WealthExclEquity",         # Wealth w/o equity (WEALTH1)
    "ER85692" = "WealthInclEquity",         # Wealth w/ equity (WEALTH2)
    "ER82018" = "Age",                      # Age of reference person (BC21)
    "ER82019" = "Gender",                   # Sex of reference person (F1F)
    "ER82022" = "NumChildren",              # Number of children
    "ER82026" = "MaritalStatus",            # Marital status
    "ER85773" = "MetroArea",                # Metropolitan area (METRO)
    "ER85774" = "BealeCode"                 # Beale rural-urban code (BEALE)
  )
)

# Helper functions
get_all_variables <- function() {
  all_vars <- unique(unlist(lapply(variable_mappings, names)))
  return(all_vars)
}

cat("Variable selection defined for", length(variable_mappings), "waves\n")
```

```
## Variable selection defined for 5 waves
```

```r
cat("Total unique variables across all years:", length(get_all_variables()), "\n")
```

```
## Total unique variables across all years: 143
```

3. Load Excel Data

```r
cat("Loading Excel Data...\n")
```

```
## Loading Excel Data...
```

```r
# Get all unique ER codes across all years
all_er_codes <- get_all_variables()

# Load the main panel Excel file
psid_raw <- readxl::read_excel("2015_2017_2019_2021_2023.xlsx",
                               sheet = "Data",
                               col_types = "guess")

# Check available columns and extract only those that exist
available_cols <- intersect(all_er_codes, names(psid_raw))
missing_cols <- setdiff(all_er_codes, names(psid_raw))

if(length(missing_cols) > 0) {
  cat("Warning: Missing columns:", paste(missing_cols, collapse = ", "), "\n")
}

# Extract only the columns we need
psid_selected <- psid_raw %>%
  select(all_of(available_cols))

cat("Data loaded successfully:", nrow(psid_selected), "rows,", ncol(psid_selected), "columns\n")
```

```
## Data loaded successfully: 13225 rows, 143 columns
```

4. Reshape to Long Format

```r
cat("Reshaping to Long Format...\n")
```

```
## Reshaping to Long Format...
```

```r
# Function to extract year from ER code
extract_year <- function(er_codes) {
  # Create a look-up table from all variable_mappings
  lookup <- character()
  for(year in names(variable_mappings)) {
    year_codes <- names(variable_mappings[[year]])
    lookup[year_codes] <- year
  }

  # Look up each ER code
  result <- lookup[er_codes]
  result[is.na(result)] <- "Unknown"
  return(result)
}

# Function to get variable name from ER code and year
get_var_name <- function(er_code, year) {
  if(year %in% names(variable_mappings)) {
    mapping <- variable_mappings[[year]]
```

```
    var_name <- mapping[er_code]
    return(ifelse(is.na(var_name), paste0("Unknown_", er_code), var_name))
  } else {
    return(paste0("Unknown_", er_code))
  }
}

# Reshape to long format
psid_long <- psid_selected %>%
  # Convert to long format
  pivot_longer(
    cols = everything(),
    names_to = "ER_CODE",
    values_to = "VALUE"
  ) %>%
  # Extract year from ER code
  mutate(
    YEAR = extract_year(ER_CODE),
    # Create row identifier to maintain family relationships
    row_id = rep(1:nrow(psid_selected), ncol(psid_selected))
  ) %>%
  # Get variable short names
  mutate(
    var_short = purrr::map2_chr(ER_CODE, YEAR, get_var_name)
  ) %>%
  # Filter out unknown years and variables
  filter(YEAR != "Unknown", !str_detect(var_short, "Unknown_"))

cat("Long format created:", nrow(psid_long), "observations\n")
```

```
## Long format created: 1891175 observations
```

5. Reshape to Wide Format

```
cat("Reshaping to Wide Format...\n")
```

```
## Reshaping to Wide Format...
```

```
# First, we need to identify the ID variable for each year and create a consistent ID
psid_with_id <- psid_long %>%
  # Extract ID values for each row
  filter(var_short == "ID") %>%
  select(row_id, YEAR, ID = VALUE) %>%
  # Join back to get ID for each observation
  right_join(psid_long, by = c("row_id", "YEAR")) %>%
  # Remove the ID variable rows since we now have it as a column
  filter(var_short != "ID") %>%
  # Clean up
  select(-row_id, -ER_CODE)

# Create the wide format: one row per ID × YEAR
psid_wide <- psid_with_id %>%
  # Remove any remaining missing values in critical columns
  filter(!is.na(ID), !is.na(YEAR), !is.na(var_short)) %>%
```

```r
  # Create the wide format
  pivot_wider(
    id_cols = c(ID, YEAR),
    names_from = var_short,
    values_from = VALUE,
    values_fn = function(x) x[!is.na(x)][1]  # Take first non-NA value if duplicates
  ) %>%
  # Convert YEAR to numeric for easier analysis
  mutate(YEAR = as.numeric(YEAR)) %>%
  # Arrange by ID and YEAR
  arrange(ID, YEAR)

cat("Wide format created:", nrow(psid_wide), "rows (ID × YEAR combinations)\n")
```

```
## Wide format created: 46583 rows (ID × YEAR combinations)
```

6. Data Cleaning and Labeling

```r
cat("Cleaning and Labeling Data...\n")
```

```
## Cleaning and Labeling Data...
```

```r
# Convert relevant columns to numeric for analysis
numeric_vars <- c("WealthInclEquity", "WealthExclEquity", "CreditCardDebt",
                  "StudentLoanDebt", "Age", "NumChildren", "FarmBusinessValue",
                  "StocksValue", "CheckingSavingsValue", "VehicleValue",
                  "OtherAssetsValue", "CDsValue", "SavingsBondsValue")

# Apply categorical labels
race_labels <- c(
  "1" = "White",
  "2" = "Black/African American",
  "3" = "American Indian/Alaska Native",
  "4" = "Asian",
  "5" = "Native Hawaiian/Pacific Islander",
  "7" = "Other",
  "9" = "Multiple Races"
)

gender_labels <- c(
  "1" = "Male",
  "2" = "Female"
)

marital_labels <- c(
  "1" = "Married",
  "2" = "Never married",
  "3" = "Widowed",
  "4" = "Divorced",
  "5" = "Separated",
  "8" = "Other",
  "9" = "NA/Don't know"
)
```

```r
hispanic_labels <- c(
  "1" = "Hispanic",
  "5" = "Not Hispanic"
)

farm_labels <- c(
  "1" = "Yes",
  "5" = "No"
)

# Create final analysis dataset
psid_analysis <- psid_wide %>%
  # Convert numeric variables
  mutate(across(all_of(intersect(numeric_vars, names(.))), as.numeric)) %>%
  # Apply categorical labels
  mutate(
    Race = race_labels[as.character(as.numeric(Race))],
    Gender = gender_labels[as.character(as.numeric(Gender))],
    MaritalStatus = marital_labels[as.character(as.numeric(MaritalStatus))],
    Hispanic = hispanic_labels[as.character(as.numeric(Hispanic))],
    FarmBusinessIndicator = farm_labels[as.character(as.numeric(FarmBusinessIndicator))]
  )

cat("Data cleaned and labeled successfully!\n")
```

```
## Data cleaned and labeled successfully!
```

7. Generate Summary Statistics

```r
cat("Generating Summary Statistics...\n")
```

```
## Generating Summary Statistics...
```

```r
# Summary 1: Wealth statistics by year
wealth_summary <- psid_analysis %>%
  group_by(YEAR) %>%
  summarise(
    `Sample Size` = scales::comma(n()),
    `Mean Wealth (incl. equity)` = scales::dollar(mean(WealthInclEquity, na.rm = TRUE)),
    `Median Wealth (incl. equity)` = scales::dollar(median(WealthInclEquity, na.rm = TRUE)),
    `Mean Wealth (excl. equity)` = scales::dollar(mean(WealthExclEquity, na.rm = TRUE)),
    `Median Wealth (excl. equity)` = scales::dollar(median(WealthExclEquity, na.rm = TRUE)),
    .groups = "drop"
  )

print("Wealth Statistics by Year:")
```

```
## [1] "Wealth Statistics by Year:"
```

```r
print(wealth_summary)
```

```
## # A tibble: 5 x 6
##    YEAR `Sample Size` `Mean Wealth (incl. equity)` Median Wealth (incl. equity~1
##   <dbl> <chr>         <chr>                        <chr>
## 1  2015 9,048         $225,898                     $23,000
```

```
## 2  2017 9,607         $246,790                    $27,850
## 3  2019 9,569         $255,710                    $33,800
## 4  2021 9,207         $320,305                    $54,000
## 5  2023 9,152         $390,614                    $64,000
## # i abbreviated name: 1: `Median Wealth (incl. equity)`
## # i 2 more variables: `Mean Wealth (excl. equity)` <chr>,
## #   `Median Wealth (excl. equity)` <chr>
```

```
# Summary 2: Debt statistics by year
debt_summary <- psid_analysis %>%
  group_by(YEAR) %>%
  summarise(
    `Sample Size` = scales::comma(n()),
    `Mean Credit Card Debt` = scales::dollar(mean(CreditCardDebt, na.rm = TRUE)),
    `Families with Credit Card Debt` = scales::comma(sum(CreditCardDebt > 0, na.rm = TRUE)),
    `% with Credit Card Debt` = paste0(round(100 * sum(CreditCardDebt > 0, na.rm = TRUE) / sum(!is.na(C
    `Mean Student Loan Debt` = scales::dollar(mean(StudentLoanDebt, na.rm = TRUE)),
    `Families with Student Loan Debt` = scales::comma(sum(StudentLoanDebt > 0, na.rm = TRUE)),
    `% with Student Loan Debt` = paste0(round(100 * sum(StudentLoanDebt > 0, na.rm = TRUE) / sum(!is.na
    .groups = "drop"
  )

print("Debt Statistics by Year:")
```

```
## [1] "Debt Statistics by Year:"
```

```
print(debt_summary)
```

```
## # A tibble: 5 x 8
##    YEAR `Sample Size` `Mean Credit Card Debt` `Families with Credit Card Debt`
##   <dbl> <chr>         <chr>                   <chr>
## 1  2015 9,048         $3.78                   6,195
## 2  2017 9,607         $3.75                   6,988
## 3  2019 9,569         $3.75                   6,907
## 4  2021 9,207         $3.90                   6,436
## 5  2023 9,152         $3.83                   6,360
## # i 4 more variables: `% with Credit Card Debt` <chr>,
## #   `Mean Student Loan Debt` <chr>, `Families with Student Loan Debt` <chr>,
## #   `% with Student Loan Debt` <chr>
```

```
# Summary 3: Demographic statistics by year
demo_summary <- psid_analysis %>%
  group_by(YEAR) %>%
  summarise(
    `Sample Size` = scales::comma(n()),
    `Mean Age` = paste0(round(mean(Age, na.rm = TRUE), 1), " years"),
    `Mean Number of Children` = paste0(round(mean(NumChildren, na.rm = TRUE), 1), " children"),
    .groups = "drop"
  )

print("Demographic Statistics by Year:")
```

```
## [1] "Demographic Statistics by Year:"
```

```
print(demo_summary)
```

```
## # A tibble: 5 x 4
```

```
##    YEAR `Sample Size` `Mean Age` `Mean Number of Children`
##   <dbl> <chr>         <chr>      <chr>
## 1  2015 9,048         45.5 years 0.8 children
## 2  2017 9,607         46.1 years 0.8 children
## 3  2019 9,569         46.6 years 0.8 children
## 4  2021 9,207         47.4 years 0.8 children
## 5  2023 9,152         47.8 years 0.7 children
```

```
# Summary 4: Race distribution by year
if("Race" %in% names(psid_analysis)) {
  race_summary <- psid_analysis %>%
    filter(!is.na(Race)) %>%
    group_by(YEAR, Race) %>%
    summarise(count = n(), .groups = "drop") %>%
    group_by(YEAR) %>%
    mutate(
      total = sum(count),
      percentage = round(100 * count / total, 1)
    ) %>%
    select(YEAR, Race, count, percentage) %>%
    pivot_wider(names_from = YEAR, values_from = percentage, names_prefix = "")

  print("Race Distribution by Year (%):")
  print(race_summary)
}
```

```
## [1] "Race Distribution by Year (%):"
## # A tibble: 32 x 7
##    Race                             count `2015` `2017` `2019` `2021` `2023`
##    <chr>                            <int>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 American Indian/Alaska Native       46    0.7    0.7     NA     NA     NA
##  2 Asian                               68    1.1   NA       NA     NA     NA
##  3 Black/African American            2469   39.7   NA       NA     NA     NA
##  4 Multiple Races                      51    0.8   NA       NA     NA     NA
##  5 Native Hawaiian/Pacific Islander     4    0.1   NA       NA     NA     NA
##  6 Other                              192    3.1   NA       NA     NA     NA
##  7 White                             3384   54.5   NA       NA     NA     NA
##  8 Asian                              105   NA      1.5     NA     NA     NA
##  9 Black/African American            2617   NA     37.4     NA     NA     NA
## 10 Multiple Races                      80   NA      1.1     NA     NA     NA
## # i 22 more rows
```

8. Save Output Files

```
cat("Saving Output Files...\n")
```

```
## Saving Output Files...
```

```
# Save the main analysis dataset
saveRDS(psid_analysis, "psid_panel_2015_2023.rds")
write.csv(psid_analysis, "psid_panel_2015_2023.csv", row.names = FALSE)

# Save summary statistics
write.csv(wealth_summary, "wealth_summary.csv", row.names = FALSE)
write.csv(debt_summary, "debt_summary.csv", row.names = FALSE)
```

```r
write.csv(demo_summary, "demographic_summary.csv", row.names = FALSE)

cat("All files saved successfully!\n")
```

```
## All files saved successfully!
```

```r
cat("Final dataset dimensions:", nrow(psid_analysis), "rows ×", ncol(psid_analysis), "columns\n")
```

```
## Final dataset dimensions: 46583 rows × 30 columns
```

```r
cat("Years covered:", paste(sort(unique(psid_analysis$YEAR)), collapse = ", "), "\n")
```

```
## Years covered: 2015, 2017, 2019, 2021, 2023
```

Summary

This script successfully:

1. **Loaded and processed** PSID panel data for 2015-2023
2. **Mapped variables** consistently across survey waves using ER codes
3. **Reshaped data** from wide to long to wide format for panel analysis
4. **Applied categorical labels** to improve data interpretability
5. **Generated summary statistics** showing wealth, debt, and demographic trends
6. **Saved clean datasets** ready for further analysis

The final dataset contains 46583 family-year observations across 5 survey waves, with variables covering wealth, debt, demographics, and family characteristics.