Fertility analysis in the US (2010-2024)

Gianfranco Gervasoni Morandi

2026-01-14

Contents

1. Executive Summary

This project leverages U.S. Census Bureau (ACS) data to explore how socioeconomic factors—such as education, poverty levels, and ethnicity—influence fertility rates across the United States. The analysis spans over a decade, handling complex hierarchical data and structural gaps (2020 Pandemic).

Technical Challenges & Solutions

Census Data Wrangling: Built a robust R pipeline to clean domain-specific symbols (**, (X), N) and normalize inconsistent column schemas across multi-year datasets.

Hierarchical Label Processing: Developed custom stringr functions to handle indented categorical labels, enabling precise filtering of demographic sub-groups.

Advanced Modeling: Implemented a dual-model approach using tidymodels:

Random Forest: To capture non-linear interactions and maximize predictive power.

Lasso Regression: To ensure statistical interpretability and feature selection.

Key Findings

Predictive Performance: The Random Forest model achieved an R-Squared of 0.934, indicating that demographic features explain the vast majority of fertility variance.

Socioeconomic Drivers: Lasso regularization identified Higher Education (Bachelor's/Graduate degrees) and High Income (>200% poverty level) as the most significant negative predictors of fertility rates.

Adolescent Trends: The model accurately isolated age-specific dynamics, particularly within the 15-19 and 20-34 cohorts.

Model Metrics

R-Squared: 0.934 RMSE: 5.93 MAE: 3.26

2. Data Ingestion & Cleaning

```
file_paths <- list.files(path = "C:/Users/giang/OneDrive/Desktop/Fertility-US-data/", pattern = "*.csv"
read_acs_data <- function(filename) {
     # Extract name from the name (ej. "...2010...")
     year_extracted <- str_extract(filename, "20[0-9]{2}")

     read_csv(filename, show_col_types = FALSE) %>%
         mutate(Year = as.numeric(year_extracted)) %>%
         # Deletion of second row if descriptive
         filter(row_number() > 1)
}
fertility_raw <- map_dfr(file_paths, read_acs_data)
glimpse(fertility_raw)
```

```
## Rows: 532
## Columns: 18
## $ `Label (Grouping)`
## $ `United States!!Total!!Estimate`
## $ `United States!!Total!!Margin of Error`
## $ `United States!!Women with births in the past 12 months !!Number!!Estimate`
## $ `United States!!Women with births in the past 12 months !!Number!!Margin of Error`
## $ `United States!!Women with births in the past 12 months !!Percent Distribution!!Estimate`
## $ `United States!!Women with births in the past 12 months !!Percent Distribution!!Margin of Error`
## $ `United States!!Women with births in the past 12 months !!Rate per 1,000 women!!Estimate`
## $ `United States!!Women with births in the past 12 months !!Rate per 1,000 women!!Margin of Error`
## $ `United States!!Percent of women who had a birth in the past 12 months who were unmarried!!Estimat
## $ `United States!!Percent of women who had a birth in the past 12 months who were unmarried!!Margin
## $ Year
## $ `United States!!Women with births in the past 12 months!!Number!!Estimate`
## $ `United States!!Women with births in the past 12 months!!Number!!Margin of Error`
## $ `United States!!Women with births in the past 12 months!!Percent Distribution!!Estimate`
## $ `United States!!Women with births in the past 12 months!!Percent Distribution!!Margin of Error`
## $ `United States!!Women with births in the past 12 months!!Rate per 1,000 women!!Estimate`
## $ `United States!!Women with births in the past 12 months!!Rate per 1,000 women!!Margin of Error`
```

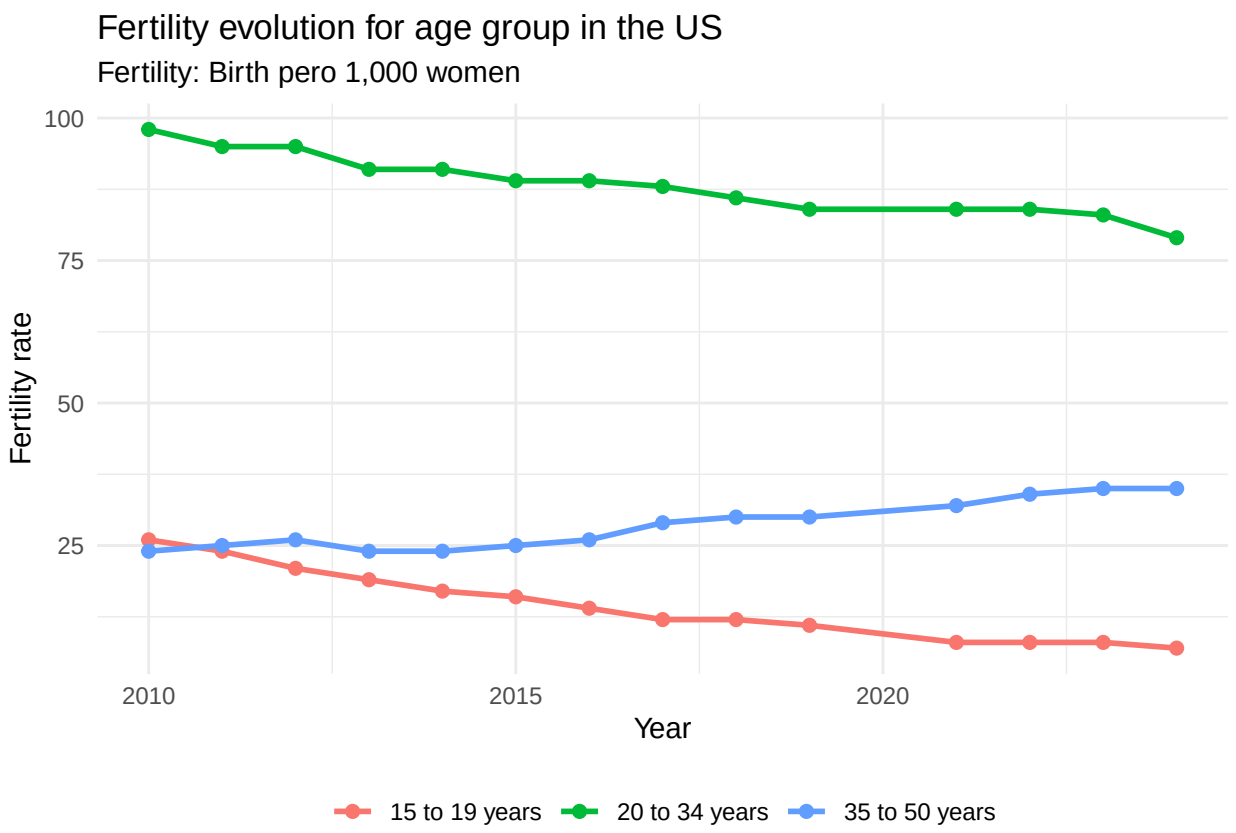
```
census_na_strings <- c("**", "***", "*****", "-", "N", "(X)")

fertility_clean <- fertility_raw %>%
  # 1. Standardization "snake_case"
  clean_names() %>%
  # 2. Fusion of columns with same name
  # Using coalesce() to take data from the original column or the duplicated one (_2) if the first one
  mutate(
    # Fusion for "Women with births... Number"
      women_with_births_combined = coalesce(
        united_states_women_with_births_in_the_past_12_months_number_estimate,
        united_states_women_with_births_in_the_past_12_months_number_estimate_2
        ),
    # Fusion for "Rate per 1,000 women"
      fertility_rate_combined = coalesce(
        united_states_women_with_births_in_the_past_12_months_rate_per_1_000_women_estimate,
```

```
        united_states_women_with_births_in_the_past_12_months_rate_per_1_000_women_estimate_2
        )
    ) %>%

  # 3. Renaming
  select(
    year,
    grouping = label_grouping,
    total_women = united_states_total_estimate,
    women_with_births = women_with_births_combined,
    fertility_rate_per_1000 = fertility_rate_combined
    ) %>%

  # 4. Cleaning
  mutate(across(c(total_women, women_with_births, fertility_rate_per_1000),
                \(x) replace(x, x %in% census_na_strings, NA))) %>%

  # Converting to numbers
  mutate(across(c(total_women, women_with_births, fertility_rate_per_1000),
                \(x) as.numeric(str_remove_all(x, ",")))) %>%

  # Filtering
  filter(!is.na(year))

# Visualization function
plot_demographic_group <- function(data, target_labels, plot_title) {

  data %>%
    # 1. Cleaning
    mutate(grouping_clean = str_trim(grouping)) %>%

    # 2. Filtering categories of interest
    filter(grouping_clean %in% target_labels) %>%

    # 3. Visualization
    ggplot(aes(x = year, y = fertility_rate_per_1000, color = grouping_clean)) +
    geom_line(linewidth = 1) +
    geom_point(size = 2) +

    # Aesthetics
    theme_minimal() +
    labs(
      title = plot_title,
      subtitle = "Fertility: Birth pero 1,000 women",
      y = "Fertility rate",
      x = "Year",
      color = NULL
    ) +
    theme(legend.position = "bottom")
}

# Setting the categories of interest
labels_age <- c("15 to 19 years", "20 to 34 years", "35 to 50 years")
```

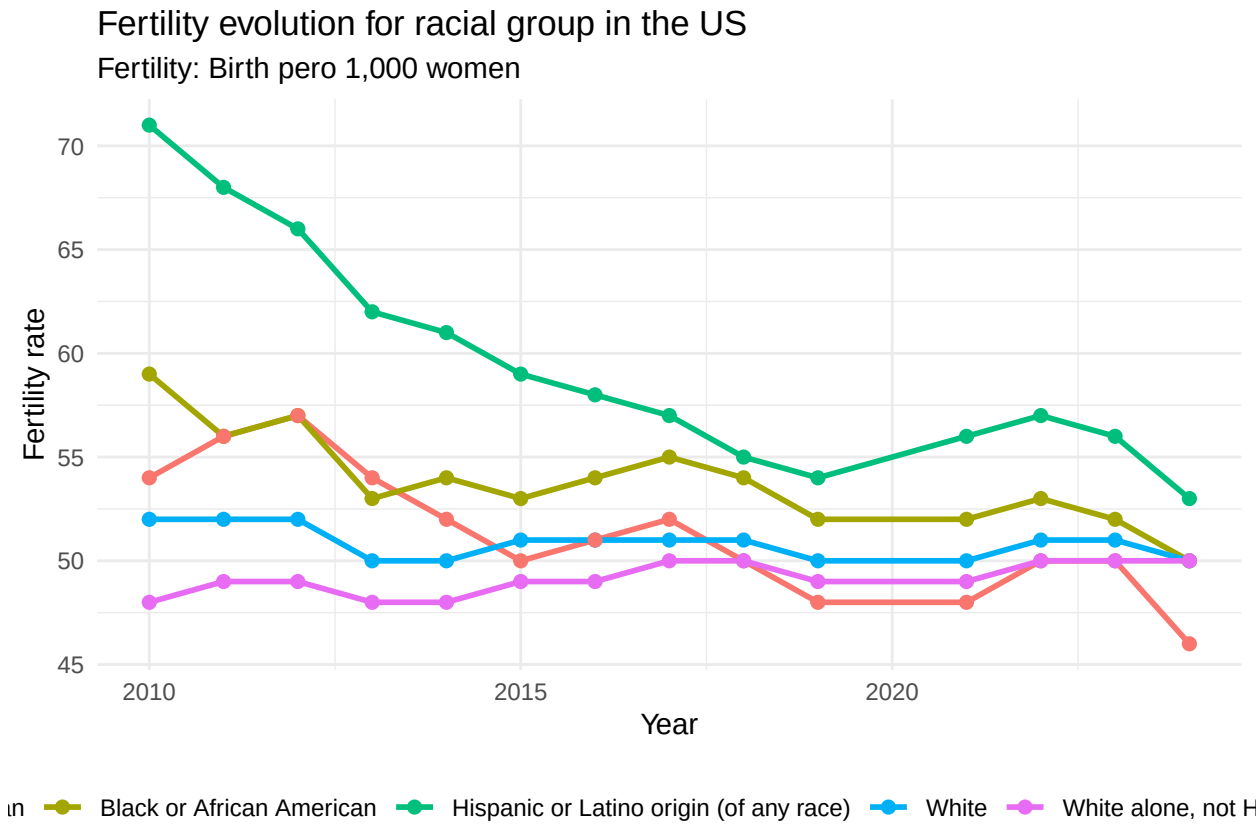
```r
labels_race <- c(
  "White",
  "Black or African American",
  "Asian",
  "Hispanic or Latino origin (of any race)",
  "White alone, not Hispanic or Latino"
)
labels_education <- c(
  "Less than high school graduate",
  "High school graduate (includes equivalency)",
  "Some college or associate's degree",
  "Bachelor's degree",
  "Graduate or professional degree"
)
labels_nativity <- c("Native", "Foreign born")
labels_poverty <- c(
  "Below 100 percent of poverty level",
  "100 to 199 percent of poverty level",
  "200 percent or more above poverty level"
)
labels_labor_force <- c(
  "Women 16 to 50 years",
  "In labor force"
)
labels_public_ass <- c(
  "Women 15 to 50 years",
  "Received public assistance income",
  "Did not receive public assistance income"
  )
```

3. Exploratory Data Analysis (EDA)

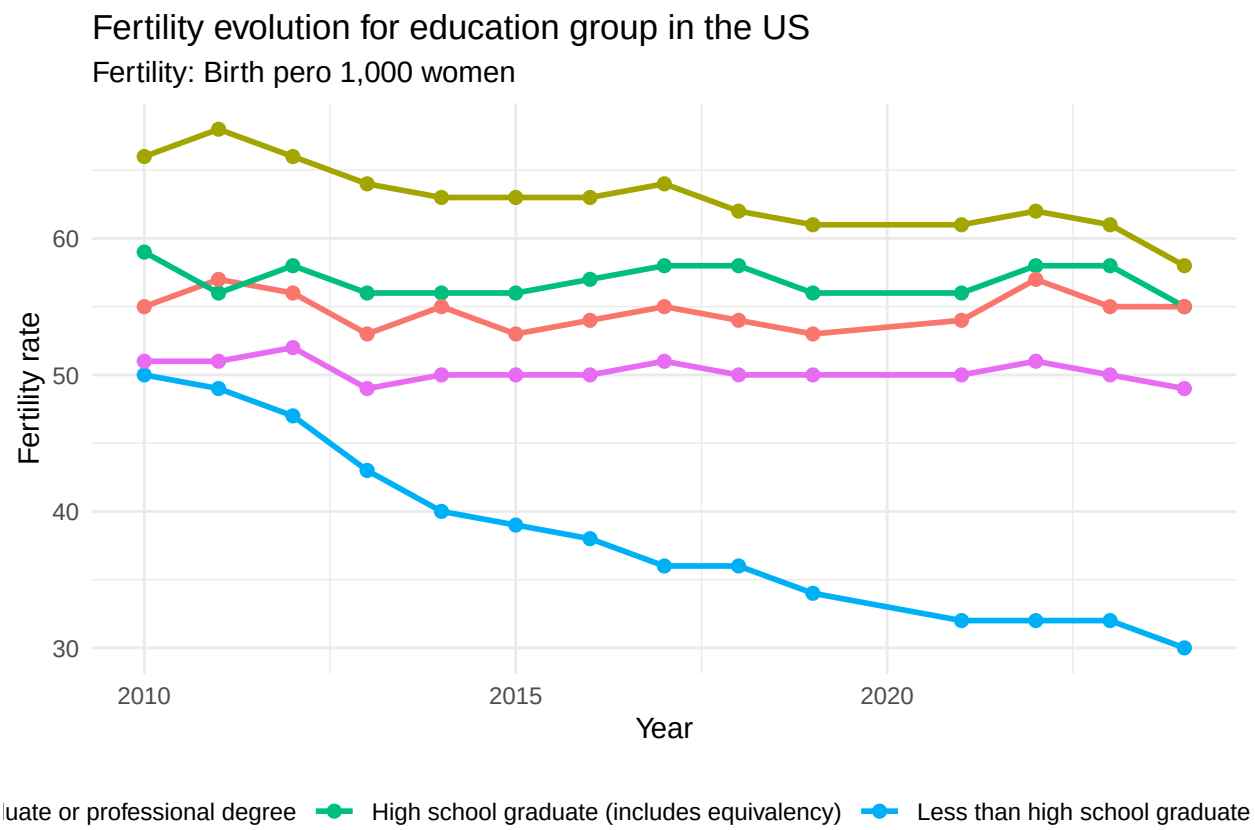
```r
# Plotting
plot_demographic_group(fertility_clean, labels_age, "Fertility evolution for age group in the US")
```



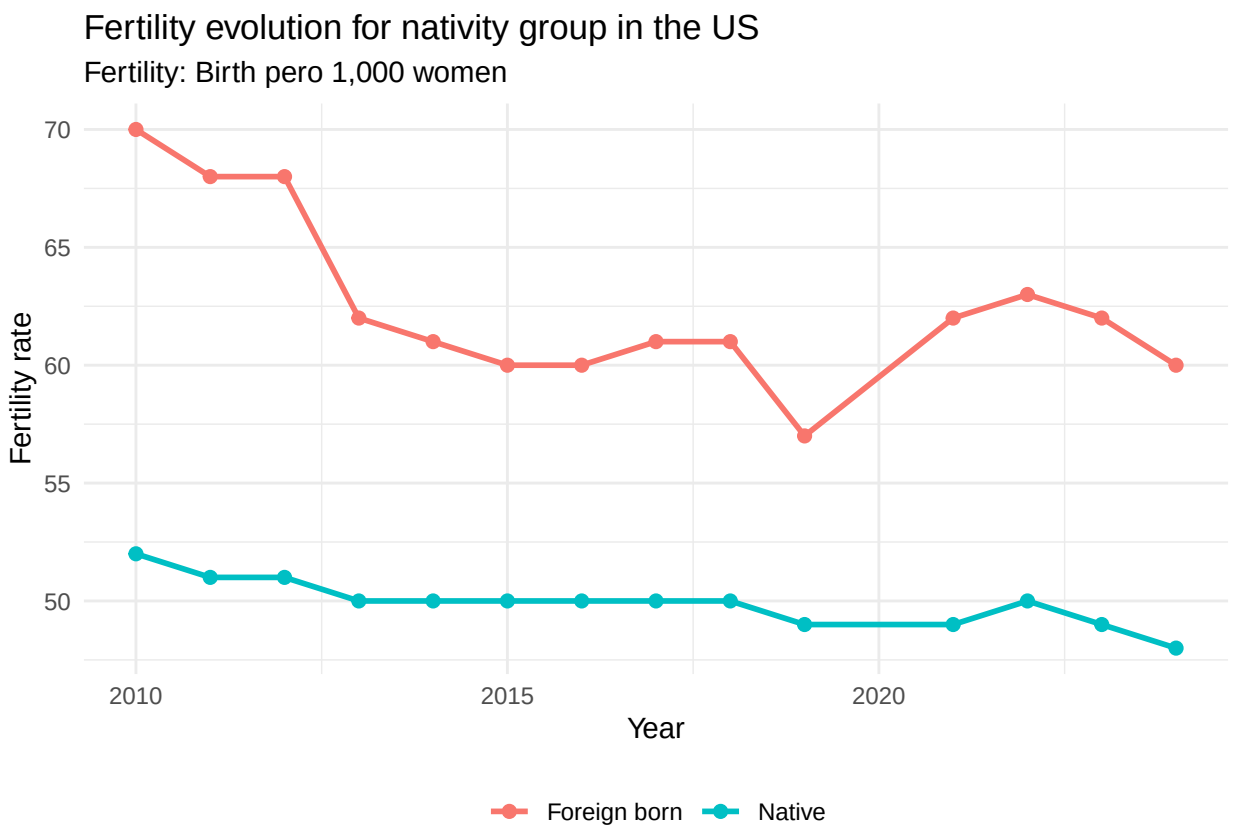

```
plot_demographic_group(fertility_clean, labels_race, "Fertility evolution for racial group in the US")
```



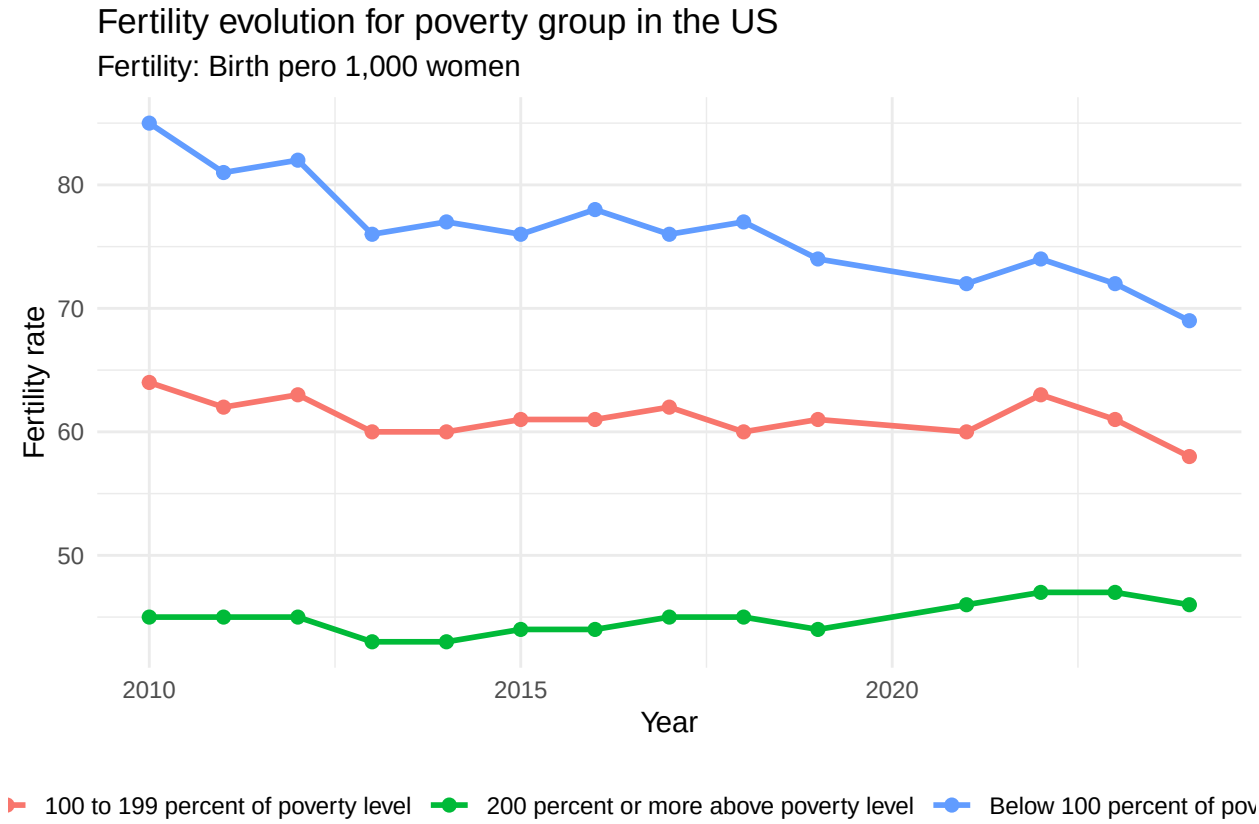
```
plot_demographic_group(fertility_clean, labels_education, "Fertility evolution for education group in th
```



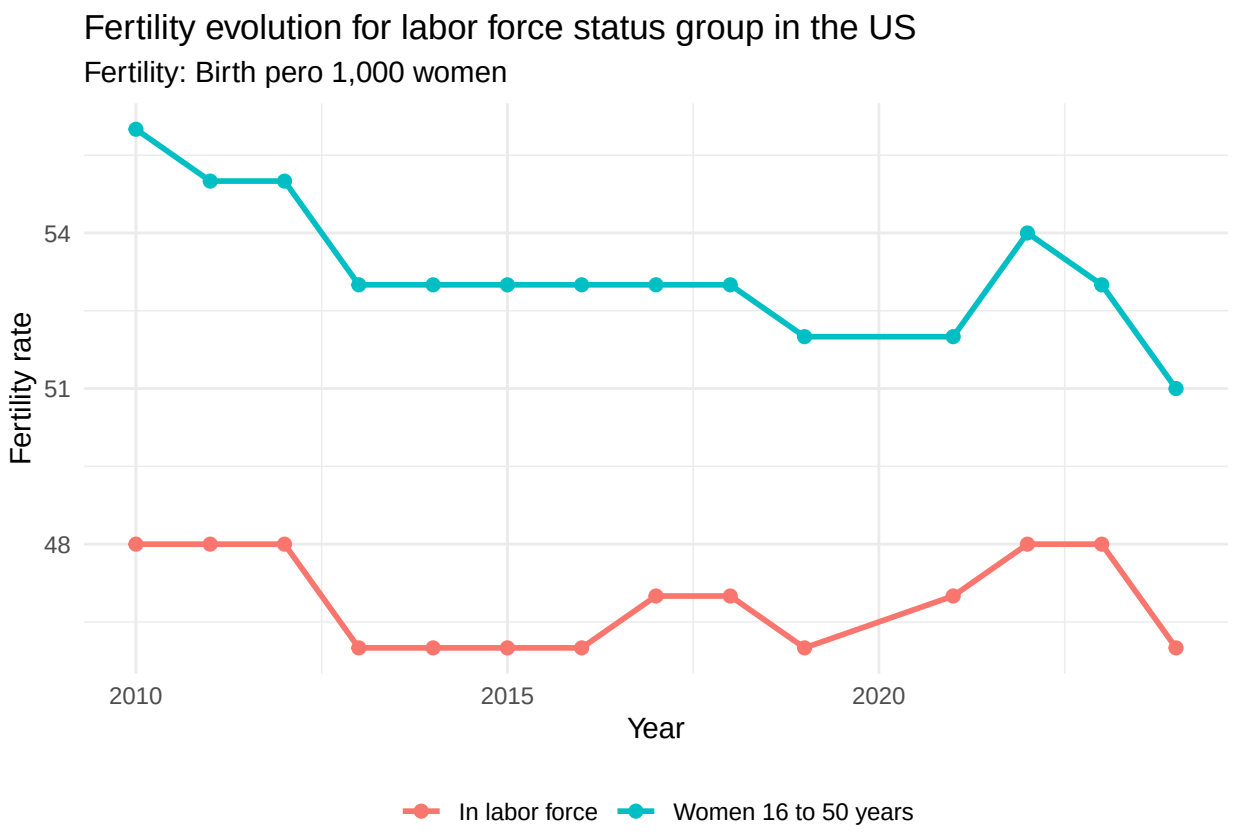

```
plot_demographic_group(fertility_clean, labels_nativity, "Fertility evolution for nativity group in the
```



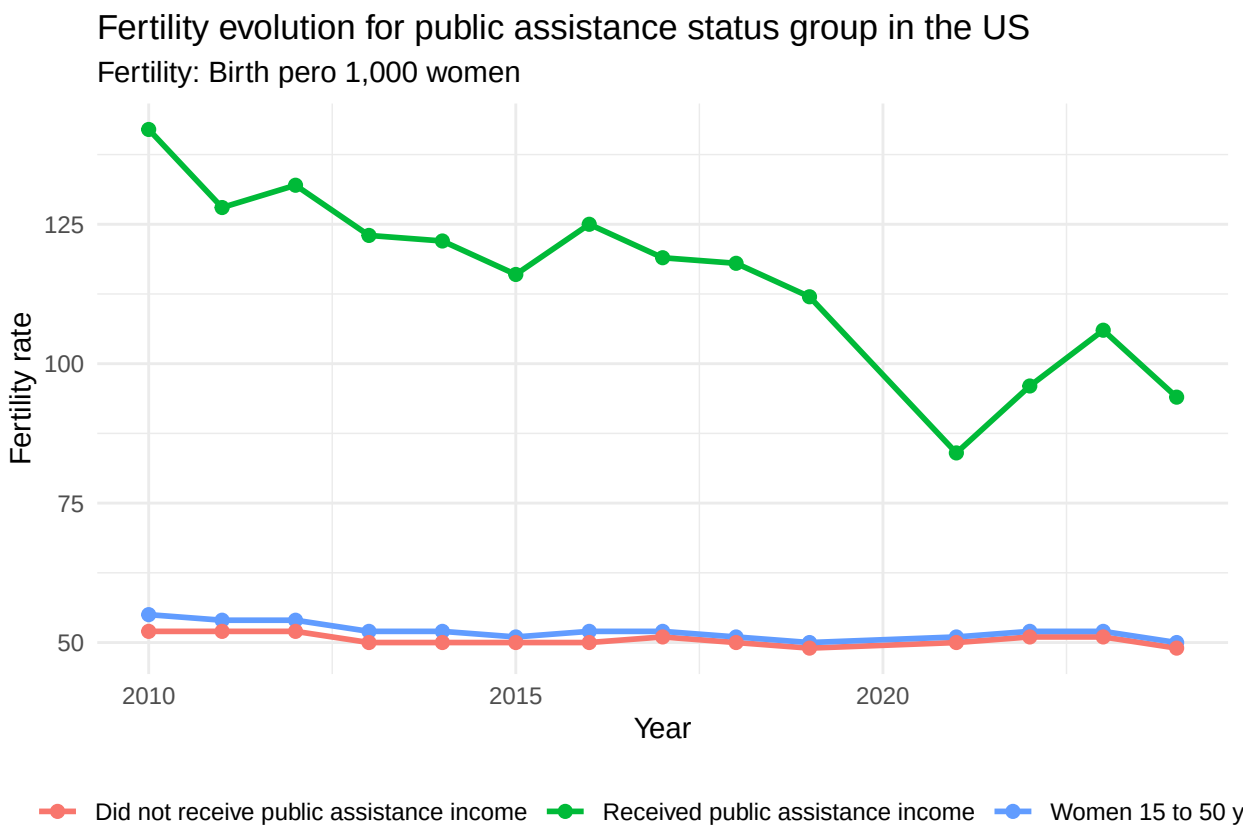

```
plot_demographic_group(fertility_clean, labels_poverty, "Fertility evolution for poverty group in the US
```



```
plot_demographic_group(fertility_clean, labels_labor_force, "Fertility evolution for labor force status
```



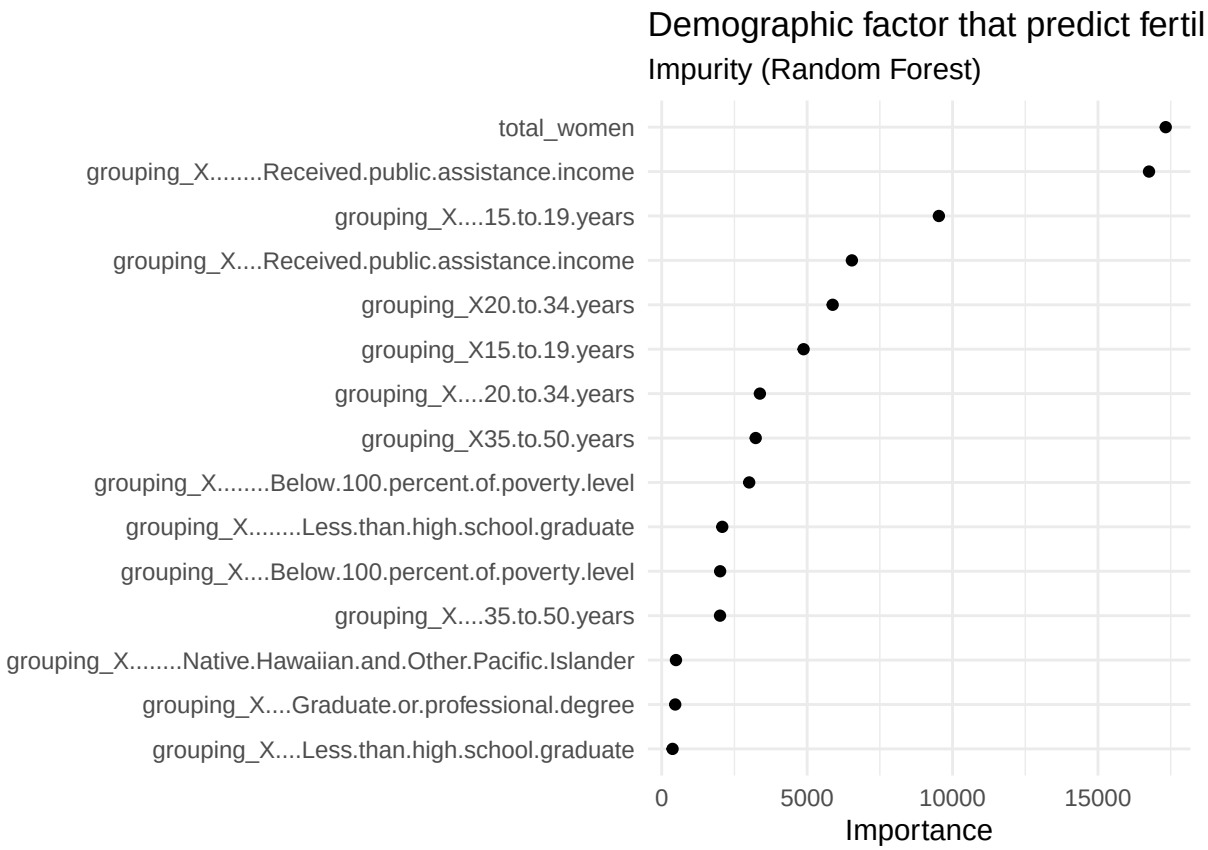

```
plot_demographic_group(fertility_clean, labels_public_ass, "Fertility evolution for public assistance s
```



4. Machine Learning Pipeline

```
# 1. Cleaning
fertility_model_df <- fertility_clean %>%
  filter(!is.na(fertility_rate_per_1000)) %>%
  filter(!grouping %in% c("Women 15 to 50 years", "NATIVITY", "EDUCATIONAL ATTAINMENT",
                          "POVERTY STATUS IN THE PAST 12 MONTHS", "LABOR FORCE STATUS"))

# 2. Split (80% training, 20% testing)
set.seed(123)
data_split <- initial_split(fertility_model_df, prop = 0.8)
train_data <- training(data_split)
test_data  <- testing(data_split)

# 3. Preprossessing
fertility_recipe <- recipe(fertility_rate_per_1000 ~ ., data = train_data) %>%
  step_rm(year, women_with_births) %>%
  step_dummy(all_nominal_predictors()) %>%
  # Normalizing y imputing predictors
  step_normalize(all_numeric_predictors()) %>%
  step_impute_median(all_numeric_predictors())

# 4. Model (Random Forest)
rf_spec <- rand_forest(trees = 1000) %>%
  set_engine("ranger", importance = "impurity") %>%
```

```r
  set_mode("regression")

# 5. Workflow
rf_workflow <- workflow() %>%
  add_recipe(fertility_recipe) %>%
  add_model(rf_spec)

rf_fit <- rf_workflow %>% fit(data = train_data)

# 6. Variable importance
rf_fit %>%
  extract_fit_parsnip() %>%
  vip(geom = "point", num_features = 15) +
  theme_minimal() +
  labs(title = "Demographic factor that predict fertility",
       subtitle = "Impurity (Random Forest)")
```




```r
# 7. Results
results <- test_data %>%
  bind_cols(predict(rf_fit, test_data)) %>%
  metrics(truth = fertility_rate_per_1000, estimate = .pred)

print(results)
```

```
## # A tibble: 3 x 3
```

```
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       5.93
## 2 rsq     standard       0.934
## 3 mae     standard       3.26
```

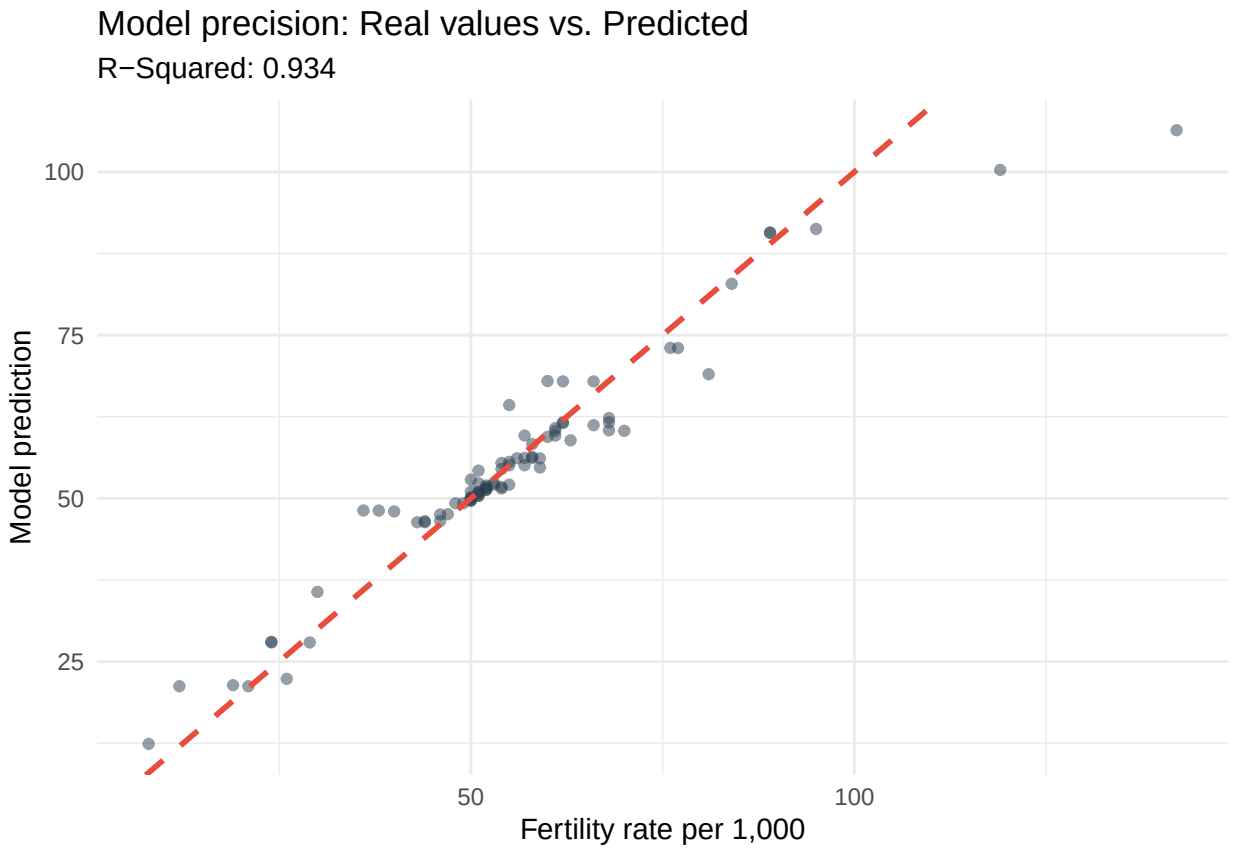
```
# Extraction of feature importance
importance_scores <- rf_fit %>%
  extract_fit_parsnip() %>%
  vi() %>%
  slice_max(Importance, n = 5) %>%
  mutate(
    Variable = str_replace_all(Variable, "grouping_", ""),
    Variable = str_replace_all(Variable, "_", " "),
    # Rounding
    Importance = round(Importance, 2)
  )

print("--- TOP 5 FERTILITY PREDICTORS ---")
```

```
## [1] "--- TOP 5 FERTILITY PREDICTORS ---"
```

```
print(importance_scores)
```

```
## # A tibble: 5 x 2
##   Variable                                    Importance
##   <chr>                                            <dbl>
## 1 total women                                     17329.
## 2 X........Received.public.assistance.income      16754.
## 3 X....15.to.19.years                              9529.
## 4 X....Received.public.assistance.income           6539.
## 5 X20.to.34.years                                  5878.
```

```
# Total women, public assistance, year (younger women, 15 to 19 years old,
# are more fertile than women 20 to 34 years old)

# Predicting on the test data
test_predictions <- test_data %>%
  bind_cols(predict(rf_fit, test_data))

# Model precision
ggplot(test_predictions, aes(x = fertility_rate_per_1000, y = .pred)) +
  geom_point(alpha = 0.5, color = "#2c3e50") +
  geom_abline(lty = 2, color = "#e74c3c", linewidth = 1) + # Perfect line
  theme_minimal() +
  labs(
    title = "Model precision: Real values vs. Predicted",
    subtitle = paste0("R-Squared: ", round(results$.estimate[2], 3)),
    x = "Fertility rate per 1,000",
    y = "Model prediction"
  )
```

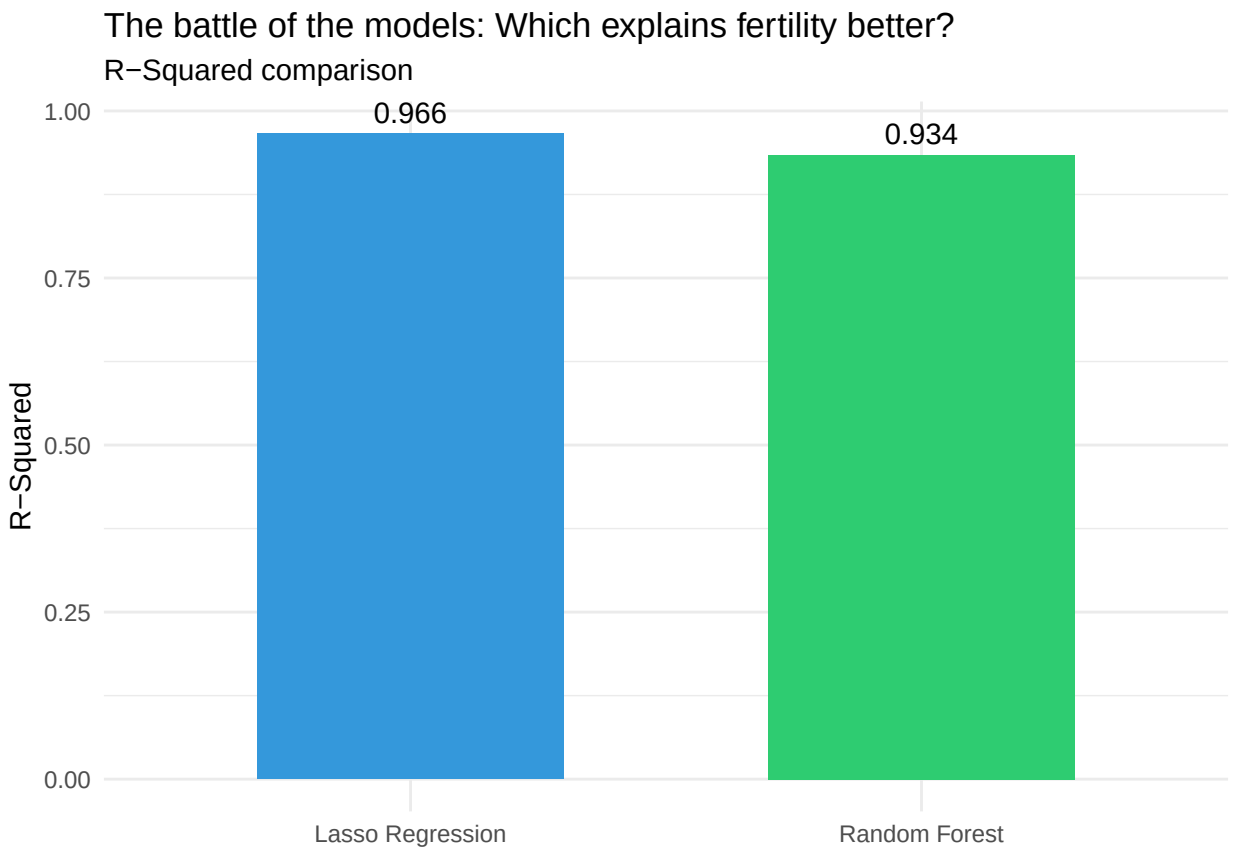



```
# Generalized Linear Model (GLM) - Lasso regularization to select variables

lasso_spec <- linear_reg(penalty = 0.1, mixture = 1) %>% # mixture = 1 is Lasso
  set_engine("glmnet")

lasso_workflow <- workflow() %>%
  add_recipe(fertility_recipe) %>%
  add_model(lasso_spec)

lasso_fit <- lasso_workflow %>% fit(data = train_data)

# Variables
lasso_fit %>% extract_fit_parsnip() %>% tidy() %>% filter(estimate != 0)
```

```
## # A tibble: 51 x 3
##    term                                                  estimate penalty
##    <chr>                                                    <dbl>   <dbl>
##  1 (Intercept)                                             55.9        0.1
##  2 total_women                                             -1.37       0.1
##  3 grouping_X.............Asian                            -0.748      0.1
##  4 grouping_X.............Black.or.African.American        -0.157      0.1
##  5 grouping_X.............Native.Hawaiian.and.Other.Pacific.Isl~   1.31       0.1
##  6 grouping_X.............Some.other.race                   0.0375     0.1
##  7 grouping_X.............White                           -0.135      0.1
##  8 grouping_X........100.to.199.percent.of.poverty.level   0.873      0.1
##  9 grouping_X........200.percent.or.more.above.poverty.level  -0.971      0.1
```

```
## 10 grouping_X........American.Indian.and.Alaska.Native            1.19       0.1
## # i 41 more rows
```

```r
# Lasso found the Intercept at 55.9, meaning that 56 is the base fertility rate
# per 1,000 women. With no other information, the model predicts 56 births/1,000 women.

# Asian and high-income families decrease fertility.
# Poverty level and native Hawaiian, American Indian, and other islanders
# increase fertility.

# 1. Evaluating Lasso in the test dataset
lasso_results <- test_data %>%
  bind_cols(predict(lasso_fit, test_data)) %>%
  metrics(truth = fertility_rate_per_1000, estimate = .pred) %>%
  mutate(model = "Lasso Regression")

# 2. Comparing the results with the Random Forest results
comparison_table <- results %>%
  mutate(model = "Random Forest") %>%
  bind_rows(lasso_results) %>%
  filter(.metric == "rsq") # Comparison of R-squared

# 3. Visualization of the results
ggplot(comparison_table, aes(x = model, y = .estimate, fill = model)) +
  geom_col(width = 0.6) +
  geom_text(aes(label = round(.estimate, 3)), vjust = -0.5) +
  scale_fill_manual(values = c("#3498db", "#2ecc71")) +
  theme_minimal() +
  labs(
    title = "The battle of the models: Which explains fertility better?",
    subtitle = "R-Squared comparison",
    y = "R-Squared",
    x = NULL
  ) +
  theme(legend.position = "none")
```



5. Model Evaluation & Final Insights

After evaluating two machine learning approaches, Random Forest demonstrated superior predictive power (R^2 = 0.93), capturing complex interactions between age and education. Lasso Regression, on the other hand, simplified the phenomenon, identifying poverty status above 200% as the greatest socioeconomic constraint on the fertility rate in the US, while specific ethnic groups and moderate poverty levels are the main drivers. This duality of models offers both an accurate predictive tool and a roadmap for understanding the underlying causes of the demographic transition.